

Más que una introducción a la Inteligencia Artificial

Tecnologías Inteligentes

Edición 2025 CC BY-NC-ND 4.0



Juan Francisco Puentes Calvo
Prólogo de Francisco Bellas

Tecnologías Inteligentes

Más que una introducción a la inteligencia artificial



Ilustración 1: Osa panda roja, rodeada de tecnología (imagen original).
Fuente: Midjourney; propiedad del autor.

Juan Francisco Puentes Calvo, PhD.

Edición 2025 CC BY-NC-ND 4.0

Tecnologías Inteligentes

Autor: Juan Francisco Puentes Calvo

Editor: El autor

Diseño y composición: El autor

Créditos de imágenes

- **Imagen de portada y contraportada:** La imagen fue generada mediante el servicio MidJourney, de acuerdo con sus términos de uso vigentes en la fecha de creación.
- **Composición de portada y contraportada:** El autor.
- Todas las imágenes incluidas en el libro están etiquetadas con sus fuentes o autoría conocidas. Aquellas cuya fuente no esté indicada son propiedad del autor, salvo error.

Notas legales

- Todas las referencias a nombres de productos, programas, sistemas operativos, etc. son marcas registradas de sus respectivas compañías u organizaciones.
- El repositorio que acompaña a esta obra (<https://github.com/FranPuentes/iTI2025>) se publica bajo la licencia GNU GPL 3.0, con la excepción de los contenidos de terceros sujetos a otras licencias.

Licencia del libro

Este libro está protegido por derechos de autor.

Se distribuye bajo una licencia:

Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional [CC BY-NC-ND 4.0]

Esto significa que:

- Puedes copiarlo, descargarlo y compartirlo libremente, siempre que se reconozca la autoría.
- **No** está permitido venderlo ni utilizarlo con fines comerciales.
- **No** está permitido modificarlo ni crear obras derivadas a partir de él.

Texto completo de la licencia: <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>

Datos legales

Depósito legal: C 1782-2024

ISBN Tapa blanda: 978-84-09-67255-4

Creado en España

Seguimiento de cambios

Edición 2024	Máster original: Octubre del 2023 Octubre/2023 - Pagina 329: pequeña corrección. Octubre/2023 - Página 330: pequeña mejora. Noviembre/2023 – Página 107: pequeña corrección.
Edición 2025	Agosto/2024 – La sección primera (“Fundamentos”), segunda (“Visión artificial”) y tercera (“Procesamiento del lenguaje natural”) han cambiado bastante. El repositorio se ha vuelto a crear. El anexo “Herramientas” se ha eliminado. Pequeñas correcciones que se propagarán a la edición 2024. Prólogo de Francisco Bellas. Septiembre/2025: Cambio a licencia CC BY-NC-ND 4.0
No hay revisiones o cambios desde entonces.	

Sumario

Capítulo 1 ¿Qué es la inteligencia artificial?	11
1.1 ¿Qué es la inteligencia?.....	12
1.2 ¿Y la inteligencia artificial?.....	14
1.3 Un mundo de datos, información y conocimiento.....	15
1.3.1 Una realidad hecha de patrones.....	17
1.3.2 La sociedad del conocimiento.....	19
1.4 Historia de la inteligencia artificial.....	21
1.4.1 Nacimiento – formal – de la inteligencia artificial.....	24
1.4.2 Los fríos inviernos.....	25
1.4.3 Datos, datos y más datos: Big Data.....	27
1.4.4 El mejor verano de la IA.....	28
1.5 ¿Qué no es inteligencia artificial?.....	33
Retos del capítulo 1.....	35
Capítulo 2 Fundamentos de la Inteligencia Artificial	37
2.1 Tipos de inteligencia artificial.....	38
2.1.1 Según las tareas a resolver.....	38
2.1.2 Escuelas de pensamiento.....	39
2.2 Aprendizaje.....	41
2.2.1 Proceso de entrenamiento de un modelo usando aprendizaje supervisado.....	42
2.3 Resolución de problemas.....	44
2.3.1 Búsqueda de la solución en un espacio de estados.....	46
2.3.2 Resolución de problemas mediante procesos de optimización.....	50
2.4 Agentes inteligentes.....	54
Retos del capítulo 2.....	59
Capítulo 3 Inteligencia artificial en el mundo real	63
3.1 Automatización.....	63
3.2 Aprendizaje automático.....	64
3.3 Análisis de datos.....	66
3.4 Robótica.....	66
3.5 Asistencia médica.....	67
3.6 Transporte.....	68
3.7 Entretenimiento.....	69
3.8 Finanzas.....	70
3.9 Recursos humanos.....	71
3.10 Educación.....	71
3.11 Medios de comunicación.....	72
3.12 Agricultura.....	73
3.13 Energía y medio ambiente.....	74
3.14 Derecho.....	75
Retos del capítulo 3.....	77
Capítulo 4 Elementos y procesamiento básico de una imagen	81
4.1 Qué son los píxeles y cómo representan la unidad básica de una imagen digital.....	81
4.2 La resolución de una imagen y su importancia.....	83
4.3 Los espacios de color.....	84
4.4 Los formatos de archivo para imágenes más comunes.....	86

4.5 Histogramas.....	87
4.6 Las transformaciones geométricas comunes y su aplicación en el procesamiento de imágenes.....	89
4.7 Operaciones entre imágenes.....	90
Retos del capítulo 4.....	93
Capítulo 5 Clasificación, Identificación y segmentación de patrones visuales	95
5.1 Visión artificial.....	97
5.1.1 Métodos y técnicas para mejorar la calidad de las imágenes.....	98
5.2 Reconociendo patrones: clasificación.....	100
5.3 Reconocimiento de patrones: identificación de objetos.....	104
5.4 Métodos para dividir una imagen en regiones o segmentos significativos.....	108
5.5 Manos a la obra: entrenando modelos.....	112
5.5.1 Clasificador de dígitos.....	112
5.5.2 Entrenando un modelo para identificar objetos.....	117
Retos del capítulo 5.....	121
Capítulo 6 Percepción de alto nivel	123
6.1 Estimación de pose.....	123
6.2 Seguimiento de objetos.....	126
6.3 Estimación de la profundidad.....	131
6.4 Manos a la obra: Extracción de características.....	134
6.4.1 Detección de expresiones faciales.....	136
6.4.2 Reconocimiento de personas.....	139
6.4.3 Detección de gestos.....	141
Retos del capítulo 6.....	147
Capítulo 7 Impacto de la visión artificial en la sociedad	149
7.1 La automatización y la eficiencia dentro del impacto de la visión artificial en la sociedad.....	149
7.2 La seguridad y la visión artificial.....	150
7.3 El impacto de la visión artificial en el campo de la salud.....	151
7.4 La interacción persona-máquina y cómo se ha mejorado.....	153
7.5 El impacto de la visión artificial en la mejora de la accesibilidad.....	153
7.6 El impacto de la visión artificial en la industria del entretenimiento.....	154
7.7 La ética y la responsabilidad en el contexto del impacto de la visión artificial en la sociedad.....	155
7.8 La IA generativa y los <i>deep fakes</i>	156
Retos del capítulo 7.....	159
Capítulo 8 Codificando datos	163
8.1 Codificando números.....	164
8.1.1 Números naturales.....	164
8.1.2 Números enteros.....	166
8.1.3 Números en punto flotante.....	167
8.1.4 Escalares, vectores, matrices y tensores.....	167
8.2 Codificando textos.....	168
8.2.1 Unicode.....	171
8.2.2 El texto y su representación.....	173
8.3 Codificando imágenes.....	173
8.4 Codificando sonido.....	174
8.5 Procesamiento del Lenguaje Natural.....	175
8.5.1 <i>Tokens</i> y <i>tokenización</i>	176
8.5.2 Estrategias de <i>tokenización</i>	177
8.5.3 Consideraciones de la <i>tokenización</i>	178
8.6 Vectores de características.....	179
Retos del capítulo 8.....	183
Capítulo 9 De audio a texto y de nuevo a audio	185

9.1 Reconocimiento de voz y transcripción automática.....	185
9.2 Síntesis de voz.....	186
9.3 Transcripción de audio.....	188
Retos del capítulo 9.....	191
Capítulo 10 Similitud entre textos	193
10.1 Representación de textos.....	194
10.2 Modelos de similitud entre textos.....	196
10.3 Métricas de similitud.....	197
10.4 Aplicaciones de la similitud entre textos.....	201
Retos del capítulo 10.....	203
Capítulo 11 Análisis de sentimientos en textos	205
11.1 Conceptos básicos del análisis de sentimientos.....	205
11.2 Preprocesamiento de textos.....	206
11.3 Enfoques de análisis de sentimientos.....	207
11.4 Modelos de análisis de sentimientos.....	208
11.5 Manos a la obra: análisis de sentimientos de textos.....	210
11.6 Consideraciones adicionales.....	213
Retos del capítulo 11.....	215
Capítulo 12 Los modelos de lenguaje	217
12.1 Introducción a los modelos de lenguaje.....	217
12.1.1 Las propiedades emergentes.....	220
12.1.2 Encoder y decoders.....	222
12.2 Entrenamiento de modelos de lenguaje.....	224
12.3 Evaluación de modelos de lenguaje.....	227
12.4 Manos a la obra: veamos un LLM en modo decoder.....	228
12.4.1 ¿Cómo hacer un chat con un LLM?.....	231
12.4.2 ¿Cómo funciona ChatGPT y otros servicios de chat o asistentes?.....	233
12.5 Manos a la obra: vamos a hacer un chat/asistente barato.....	233
12.6 Desafíos y consideraciones éticas en los modelos de lenguaje.....	235
Retos del capítulo 12.....	237
Capítulo 13 La repercusión del procesamiento del lenguaje natural en la sociedad	239
13.1 Aplicaciones de la NLP en la sociedad.....	239
13.2 Desafíos y limitaciones del NLP.....	241
13.3 Impacto del NLP en la comunicación y la interacción social.....	242
13.4 Futuro del NLP.....	243
Retos del capítulo 13.....	245
Capítulo 14 Algoritmos y modelos	249
14.1 Introducción.....	249
14.2 Aprendizaje automático y ciencia de datos.....	251
14.3 Algoritmos en aprendizaje automático.....	252
14.3.1 Algoritmos supervisados.....	252
14.3.2 Algoritmos no supervisados.....	257
14.3.3 Algoritmos de Aprendizaje por Refuerzo.....	258
14.4 Evaluación de modelos.....	259
14.5 Selección y optimización de hiperparámetros.....	261
Retos del capítulo 14.....	263
Capítulo 15 La ciencia de los datos	265
15.1 Introducción a la ciencia de datos y aprendizaje automático.....	265
15.2 Big Data.....	266
15.3 El ciclo de vida de la ciencia de los datos.....	269

15.4 Aplicaciones de la ciencia de datos.....	271
15.4.1 Medicina.....	271
15.4.2 Finanzas.....	271
15.4.3 Marketing.....	271
15.4.4 Transporte.....	272
15.4.5 Entretenimiento.....	272
15.5 Tendencias y futuro de la ciencia de datos y aprendizaje automático.....	272
Retos del capítulo 16.....	275
Capítulo 16 Entrenamiento y evaluación de modelos	277
16.1 Regresión lineal.....	277
16.2 Regresión logística.....	281
16.3 Árboles de decisión.....	285
16.4 Bosques aleatorios.....	286
16.5 XGBoost.....	287
16.6 Clasificación multiclase.....	288
16.7 Agrupamiento.....	297
Retos del capítulo 16.....	301
Capítulo 17 Redes neuronales artificiales	303
17.1 Introducción a las Redes Neuronales Artificiales.....	303
17.2 El aprendizaje en neuronas artificiales.....	306
17.3 Estructura básica de una red neuronal.....	311
17.4 Arquitecturas de Redes Neuronales.....	313
17.5 Ejemplos de redes neuronales artificiales.....	319
17.5.1 Autoencoders.....	319
17.5.2 Entrenando una arquitectura generativa: GAN.....	321
17.5.3 Texto a imágenes.....	324
Retos del capítulo 17.....	327
Capítulo 18 La ética en la era del aprendizaje automático	329
18.1 Introducción.....	329
18.2 ¿Por qué hablar de ética aplicada a la inteligencia artificial?.....	331
18.3 Riesgos y desafíos éticos del Aprendizaje Automático.....	331
18.3.1 Sesgos algorítmicos y discriminación.....	331
18.3.2 Privacidad y seguridad.....	333
18.3.3 Responsabilidad y transparencia.....	335
18.4 Beneficios del aprendizaje automático.....	337
Retos del capítulo 18.....	339
Capítulo 19 Metodología educativa	343
19.1 Breve análisis del uso de la IA en la educación.....	343
19.2 Procesos de enseñanza-aprendizaje asistidos por IA.....	346
19.3 Patrones de aprendizaje asistido por ASED.....	348
19.4 El aula del futuro asistida por IA.....	352

Agradecimientos

A mis padres, quienes no han podido ver este libro, pero cuyos sacrificios para con sus hijos son palpables en él.

A Elena, por su amor, paciencia y constante y dulce insistencia para que no pasase más de ocho horas seguidas delante de esta pantalla, ... por segunda vez.

PRÓLOGO de Francisco Bellas

*“Que todos tus esfuerzos vayan dirigidos hacia algo,
mantén tu mirada en el destino.”*

SÉNECA

Conocí a Juanfran en el año 2023, cuando la Consellería de Educación de la Xunta de Galicia nos convocó para trabajar en la definición de los contenidos de dos nuevas materias sobre Inteligencia Artificial [IA] que se pretendían implantar en el sistema educativo gallego. Es curioso que dos individuos tan activos en el campo de la IA en la educación, con enfoques, inquietudes y trayectorias similares, que viven y trabajan en la misma ciudad, no hayan coincidido antes en más de 20 años de trayectoria. Me temo que en esto ha tenido mucho que ver la falta de colaboración que tradicionalmente ha existido entre el profesorado universitario y el pre-universitario. Aunque este sería tema para otro libro, la mayor parte de la culpa la tenemos los investigadores y profesores de las universidades, que desde una supuesta posición de superioridad hemos venido empleando nuestro esfuerzo y “talento” exclusivamente en formar futuros profesionales, sin mirar hacia atrás para ver qué podemos aportar en etapas educativas anteriores.

En ese grupo de trabajo de la Consellería, junto con la participación de otros compañeros y compañeras, logramos integrar perspectivas de estas dos etapas educativas, en mi opinión, con gran éxito. El resultado fue la definición de los contenidos de dos materias, una para estudiantes de 4º de ESO con un enfoque introductorio de la IA, y otra para estudiantes de 1º de Bachillerato con un enfoque más técnico. Desde el momento en que los currículos fueron publicados oficialmente, he podido interactuar con profesores e investigadores universitarios de toda España interesados en esta experiencia, ya que Galicia fue pionera en la creación de materias oficiales de IA. Un resumen general de sus opiniones sería algo así: *“enhorabuena por la iniciativa, pero creemos que esto hubiese requerido algo más de reflexión, y además la IA debe ser integrada de forma transversal en los planes de estudio”*. Y no lo pongo en duda, soy un intenso defensor de la educación basada en evidencia científica. Creo que toda la innovación a nivel pre-universitario requiere de un análisis y estudio previo, pero también soy abanderado de la acción como medio para el avance. Crear una alfabetización en IA, transversal en varias materias, desde la primaria hasta en bachillerato es el gran reto que tiene por delante el sistema educativo a nivel mundial, y llevará tiempo. Entre tanto, la creación de materias específicas como las de la comunidad gallega, es una vía de gran valor académico.

mico, ya que nos permite avanzar en el análisis mediante evidencias, mientras mantenemos unos contenidos específicos y acotados, e implicamos a un grupo concreto de profesores.

También pude hablar, tras la publicación de los currículos, con muchos profesores de tecnología a nivel de educación secundaria, formación profesional y bachillerato, mayormente de Galicia. Su visión era otra, ampliamente positiva hacia las dos materias, pero crítica a la vez por la falta de material educativo que les permitiese afrontarlas con garantías. Y este es, sin lugar a duda, el principal problema para introducir la enseñanza de la IA en la educación reglada: la falta de formación y de confianza de los docentes. Y si esto aplica a profesores de tecnología, con una mejor preparación de base, imaginemos a los docentes que se verán afectados en el futuro por una alfabetización transversal en IA. El presente libro ataca de lleno este problema y lo hace, además, en nuestro idioma. Tienes ante ti, por tanto, una iniciativa educativa fundamental, innovadora y precursora de otras que vendrán detrás.

En este libro se condensan temáticas, contenidos y actividades de aula obtenidas de experiencias con estudiantes que funcionaron, y otras que no tanto, durante más de 20 años de práctica docente por parte del autor. Demos, pues, las gracias a Juanfran por la labor titánica y altruista [a quién piense que uno se hace rico escribiendo libros, le animo a intentarlo] de crear un libro educativo sobre IA desde la visión del que ha partido de cero, cuando la IA no era "*trending topic*". Lo que nos sobran hoy en día son expertos en IA nacidos a la sombra de chatGPT.

"Tecnologías Inteligentes" es un libro dirigido a quien quiere adentrarse en la IA desde un enfoque técnico, conociendo sus fundamentos desde dentro. No me veo capaz de establecer una edad o nivel educativo mínimo para aprender IA con este libro, pero sí es necesario tener unos conocimientos básicos de programación de ordenadores para poderle sacar el máximo provecho. Por supuesto, los estudiantes y profesores de Formación Profesional, bachillerato y Universidad de ramas técnicas son el principal público objetivo. Pero dada la relevancia de la IA en todos los sectores de la sociedad, me atrevo a recomendar el libro a cualquier persona interesada en aprender sobre las principales tecnologías de la IA, como una base a partir de la cual poder aplicarla a cualquier otra rama del conocimiento.

Los tres primeros capítulos del libro establecen las bases sobre las que se sustentan el resto, siendo de especial interés para todos aquellos que acaban de llegar a la IA. Destaco de manera especial las páginas dedicadas a la historia de la IA que, aunque breve, existe, y es siempre muy relevante reconocer el trabajo de otros que llegaron antes. Es también de agradecer el enfoque práctico y actual del capítulo 3, para que se comprenda la magnitud de la revolución de la IA, y cómo afectará a todos los sectores y profesiones. Lo que no le envidio al autor es el gran trabajo de actualización que tendrá que hacer para mantener este capítulo al día...

De aquí en adelante, se ha organizado el libro en las tres principales tecnologías de la IA en la actualidad: visión artificial, procesado de lenguaje natural y aprendizaje automático. Se ha dejado muchas otras fuera, pero nadie puede negar que estas tres son, por su impacto real, las que cualquier interesado en IA debe conocer.

Los capítulos 4 a 7 se centran en una de las principales tecnologías del campo de la computación

aplicada, la visión artificial. Se sientan primero las bases de esta tecnología, y se entra a continuación en cómo las técnicas propias de la IA han permitido resolver problemas muy complejos mediante el uso de cámaras. Estoy convencido de que conocer qué se puede hacer actualmente con imágenes y librerías de acceso libre supondrá un gran descubrimiento para muchos lectores, porque existe un sinfín de aplicaciones prácticas a nuestro alcance que muestran las enormes posibilidades de una IA de código abierto.

Merecen una especial atención los capítulos del 8 al 13, ya que constituyen el primer material educativo adaptado a un nivel introductorio sobre procesado de lenguaje natural. Y sobra decir la relevancia que ha adquirido esta tecnología desde la aparición de los grandes modelos de lenguaje, la IA generativa, y chatGPT. Por fin todos los interesados en conocer las bases de esta área tan relevante tienen un recurso formal, concreto y práctico al que acudir.

Del capítulo 14 al 18, Juanfran se centra en el aprendizaje automático, la tercera gran tecnología sobre la que se sustenta la IA moderna. Comienza estableciendo la diferencia entre algoritmos y modelos, pasa a hablarnos de la ciencia de los datos, y posteriormente profundiza en las principales técnicas del campo. De nuevo, una estructura impecable. El capítulo 18 tiene un enfoque menos técnico y aborda el gran debate de la ética de la IA, que obviamente no podía quedar fuera. Pero es evidente que, para hablar del impacto de la IA, los que somos de la rama técnica, debemos ceder la palabra a otros compañeros, por eso aplaudo el enfoque que ha dado el autor a este capítulo, exponiendo los principales temas a tratar, pero no entrando en valoraciones subjetivas.

Para terminar, el último capítulo del libro refleja una de sus principales propiedades: este es un texto vivo. Cuando Juanfran da recomendaciones sobre el uso de la IA en la educación sabe que tendrá que venir a revisarlo a menudo, pues es un campo todavía en desarrollo en el cual la experiencia nos irá aclarando el camino. Una experiencia que no se puede ganar si no logramos que los docentes imparten con confianza las materias sobre IA.

Solo me queda, para concluir, mencionar de manera explícita la IA generativa y el chatGPT. En este punto, el autor ya deja claro desde el principio cómo ha gestionado su uso en la creación del libro, y en la creación de partes del código, y no puedo estar más de acuerdo con su enfoque. La IA generativa es una herramienta de gran utilidad si la usamos correctamente. Creo firmemente que todos aquellos que se resistan a su uso, mayoritariamente por ignorancia, irán un paso por detrás de los que lo integren en su “mundo digital”. Y no hablo de aumento de la productividad, hablo de aumento en la calidad en el trabajo realizado, y de poder emplear nuestro tiempo, lo más valioso que tenemos, en tareas que realmente valgan la pena.

Gracias Juanfran por el coraje de escribir un libro educativo, por tener claro tu objetivo.

Francisco Bellas, Catedrático en la Universidad da Coruña
Agosto de 2024

INTRODUCCIÓN

Han pasado casi 12 meses desde que lancé la edición 2024, y desde entonces, todo ha cambiado en el campo de la IA, no tanto en calidad, pero sí en cantidad. La explosión de innovación que vivimos en 2023, comparable a una presa que, tras años cerrada, de repente libera millones de litros de ideas que se empujan unas a otras, apenas dándonos tiempo para asimilar tantas novedades, se ha ralentizado. Aunque estos últimos meses no han sido tranquilos, con numerosas noticias e iniciativas, parece que estamos llegando a una meseta, quizás incluso al final de la burbuja de expectativas desmesuradas en torno a la inteligencia artificial. Esta aparente calma podría indicar un momento de madurez de la industria, un respiro antes de otro auge tecnológico. Me inclino por lo primero, aunque algunas novedades en los laboratorios sugieren que aún queda camino por recorrer.

Y sin embargo no debemos subestimar la capacidad de la tecnología para sorprendernos. La historia nos enseña que los períodos de aparente calma, a veces, suelen preceder a avances inesperados. Lo que ahora parece una meseta podría transformarse rápidamente en un nuevo pico de innovación, impulsado por desarrollos que hoy están en etapas tempranas. La inteligencia artificial aún tiene fronteras por explorar: desde la mejora en el procesamiento del lenguaje natural¹ hasta la integración más profunda en la vida cotidiana a través de dispositivos inteligentes. ¡Y qué decir de su aplicación en la robótica!

Esta nueva edición era inevitable, casi una promesa personal tras la publicación de la edición 2024. ¿Quedé satisfecho? No del todo. Había más ideas por explorar, y mi perfeccionismo, como un pequeño gnomo en mi hombro, me susurraba constantemente que sí, se podía mejorar. Y así ha sido. La sección I ["Fundamentos de IA"] ha experimentado una transformación tan profunda que podría considerarse una reescritura completa. Lo mismo ha ocurrido con la sección II ["Visión artificial"]. La sección III ["Procesamiento del lenguaje natural"] también ha sido revisada porque no puede ignorar al gnomo que me gritaba al oído: "¡Te has quedado corto!". ¿Y la sección IV ["Aprendizaje automático"]? Esa no la he cambiado, pero ahora que he terminado esta edición, empiezo a sentir de nuevo el aliento del gnomo.

Pero el mayor cambio que ha sufrido este libro vino provocado por un pequeño cambio en una herramienta que, creo, debería ser usada en las aulas; y ahora todavía más. Google Colab, una magnífica herramienta basada en entornos Jupyter, incorporó en el mes de julio la capacidad de añadir

¹ No se trata de conseguir una AGI, se trata de conseguir modelos que no alucinen (deterministas en cuanto a calidad), alineados con un fin, explicables y fiables.

celdas generadas a partir de un *prompt* de texto en lenguaje natural, usando su modelo de lenguaje Gemini. Antes, así lo planteaba en la anterior edición, debíamos usar una herramienta externa [cómo algún modelo como ChatGPT o Github Copilot], proponerle la creación de código, hacer un copy/paste, comprobar si funcionaba y entrar en el bucle prueba/error hasta obtener lo que queríamos. Ahora la integración de Gemini en el propio Colab, simplifica mucho el proceso y lo hace más asequible [cognitivamente hablando] a quien empieza. A veces, pequeños cambios ...

Debido a ese pequeño cambio, la metodología práctica del libro ha cambiado. Ha pasado de mostrar el código y su resultado a mostrar el *prompt* y el resultado de su ejecución.

Genera un dataset aleatorio en un dataframe de 100 muestras formado por dos columnas, edad y salario. Los datos deben tener una relación tal que a más edad más salario.

Generando el siguiente código²:

```

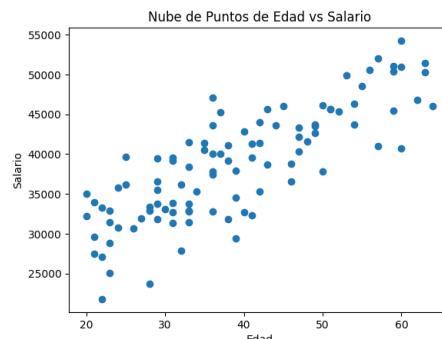
1 import pandas as pd
2 import numpy as np
3
4
5 # Generamos 100 edades aleatorias entre 20 y 65
6 edades = np.random.randint(20, 66, 100)
7
8 # Generamos salarios con una relación a la edad
9 salarios = 20000 + edades * 500 + np.random.normal(0, 5000, 100)
10
11 # Creamos el dataframe
12 df = pd.DataFrame({'edad': edades, 'salario': salarios})
13

```

Seguidamente:

Representa los datos como una nube de puntos

Generando el código que muestra el siguiente resultado:



² El/la docente, según el nivel del alumnado en python y programación en general, puede pararse más o menos tiempo a explicar el código. No descartemos el aprendizaje basado en ejemplos, también para los humanos.

A partir de este ejemplo, seguir invocando al asistente hasta llegar a obtener un modelo de regresión y evaluarlo es trivial ... si se sabe cual es la finalidad y los pasos intermedios; dicho de otro modo, si se sabe aplicar una tecnología inteligente a un problema, sin entrar en detalles que nos impidan ver el bosque.

En la anterior edición proponía en un anexo (“Metodología educativa”, ahora renombrada a “Metodología educativa” Capítulo 19) cómo se podría usar la IA en un ambiente educativo, y proponía el uso de asistentes personales [habla/escucha] tanto para el/la docente como para el alumnado [sistemas multi-agente]. El destino ha querido que la innovación vaya en esa dirección.

Con cierta regularidad me preguntan si he usado IA generativa para crear contenido del libro. La respuesta es sí. Tanto en la anterior edición como en esta, allí donde he usado a “Mi IA favorita”, lo indico:

Mi IA favorita opina: *Usarme para generar contenido de un libro puede ser una herramienta útil, especialmente para obtener ideas, mejorar redacciones o crear borradores. Sin embargo, es importante recordar que soy una herramienta complementaria. La voz, el estilo y la profundidad de un autor son insustituibles y dan vida a una obra. Utilizarme para inspirarte o pulir tus escritos puede enriquecer el proceso, pero la esencia del libro siempre debe venir de ti.*

Además, el Capítulo 3 sigue siendo el resultado de una IA, como muestra de su capacidad; y el notebook “00 Introducción a Python.ipynb” está parcialmente generado [la parte de la sintaxis del lenguaje].

Respecto al resto del contenido, es todo original [excepto en donde se indique una autoría externa] pero con un matiz importante: a veces, tras escribir un párrafo, me daba la impresión de que algo era confuso o que no proyectaba la idea que quería transmitir. Es entonces cuando usaba a mi IA favorita con el *prompt* “Intenta mejorar este párrafo: <párrafo>”, y analizaba el resultado. La mayor parte de las veces su propuesta era más clara que la mía, y en esos casos copiaba o me inspiraba su texto generado.

También hay otros usos de la IA generativa en este libro, algunas imágenes por supuesto y .. la solución al pánico a la hoja en blanco! Aunque me encontré más con ese miedo en la edición anterior, al ser la primera, descubrí que cuando a veces no sabía que decir respecto a algo, un *prompt* del estilo “qué puedo decir sobre <tema>” venía al rescate. No aumentó el contenido del libro, pero sí mi productividad.

Respecto a “mi IA favorita”, he usado en la gran mayoría de las veces al modelo de OpenAI ChatGPT-4 omni. No puedo afirmar que es el mejor pero sí el más estable en las respuestas.

El repositorio también ha cambiado.

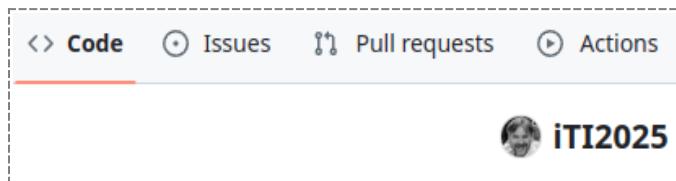
No todos los capítulos tienen un *notebook* asociado, pero hay capítulos con más de uno. El objetivo de estos es, [1] comprobar que la metodología funciona y no requiere de mucha potencia de

cálculo³ y [2] entregar una posible solución, aunque cada uno debería llegar sus propias soluciones ayudado por su asistente.

Dado que cada *notebook* es un potencial ejercicio, no he modificado los retos, ya que ahora estos se completan con los primeros.

El repositorio tiene también nueva dirección:

<https://github.com/FranPuentes/iTI2025>



Ahí encontraremos hasta 20 *notebooks*, disponibles para ser ejecutados en Google Colab (o cualquier entorno que soporte *notebooks*). Este repositorio está vivo, e irá mejorando a medida que pase el tiempo, corrigiendo posibles errores y añadiendo funcionalidades.

Cualquier error, comentario o mejora será bienvenida, tanto del libro como del código, en el propio repositorio de GitHub (“Issues”), incluyendo sugerencias en “Pull requests”.

Juan Francisco [juanfran] Puentes Calvo

Doctor en Informática y Catedrático de enseñanzas secundarias, especializado en IA
Subdirector de Certificación, Análisis de Tendencias, Coordinación y formación de IA [AESIA]
@JuanfranPuentes [en la antigua twitter]

Aunque en el ámbito profesional formo parte de la AESIA, este libro no guarda ninguna relación con la Agencia Española de Supervisión de la Inteligencia Artificial.

Se trata de un proyecto personal que no debe vincularse en modo alguno con ella y cuyo único propósito es ofrecer una modesta contribución a las necesidades pedagógicas de la comunidad hispanohablante en el campo de la inteligencia artificial.

³ La necesidad de usar GPU puede ser una limitación en la pedagogía de este libro.

Sección I: Conceptos clave de la inteligencia artificial

"La inteligencia artificial es la ciencia de hacer que las máquinas hagan cosas que requerían inteligencia humana si se hicieran por humanos"

Marvin Minsky (1927 - 2016)

En esta sección, comenzaremos definiendo qué es la inteligencia, explorando sus características y componentes. Luego, nos sumergiremos en la historia de la inteligencia artificial, repasando sus hitos más significativos y cómo ha evolucionado a lo largo del tiempo.

A continuación, clarificaremos qué es y qué no es IA, desmitificando conceptos erróneos y estableciendo una comprensión clara de sus capacidades y limitaciones. Seguidamente, abordaremos los fundamentos de la IA, explicando los conceptos esenciales que la sustentan, incluyendo el aprendizaje y las herramientas para resolver problemas, como la búsqueda de soluciones y la optimización.

Para ilustrar estos conceptos, analizaremos diversos casos de uso en el mundo real, mostrando cómo la IA se aplica en diferentes industrias y situaciones. En esta sección también estudiaremos cómo los agentes inteligentes perciben e interactúan con su entorno, y cómo representan y almacenan la información para poder razonar sobre ella.

Capítulo 1

¿QUÉ ES LA INTELIGENCIA ARTIFICIAL?



Antes de hablar de inteligencia artificial, centremos nuestra atención en los conceptos relacionados de vida e inteligencia.

La vida orgánica, en su esencia, se define por la capacidad de los organismos para crecer, reproducirse, responder a estímulos y adaptarse a su entorno a lo largo del tiempo. Dentro de este marco, la inteligencia – en su concepción más básica – puede verse como una serie de comportamientos y habilidades que permiten a un organismo no solo reaccionar pasivamente a su entorno, sino también activamente, interpretando, aprendiendo, planificando y tomando decisiones que mejoren sus posibilidades de supervivencia y bienestar.

A lo largo de la evolución, las especies han desarrollado diversas formas de inteligencia como respuesta a las presiones ambientales y a la necesidad de adaptarse a entornos cambiantes. Desde los organismos unicelulares que presentan respuestas básicas a estímulos químicos, hasta los mamíferos superiores con capacidades cognitivas avanzadas, la inteligencia ha evolucionado como una herramienta crucial para la supervivencia.

La evolución favoreció la selección de aquellos organismos que pudiesen procesar información de manera más eficiente y compleja. Por ejemplo, los primeros homínidos desarrollaron herramientas y estrategias de caza que requerían coordinación y planificación, habilidades que implican un nivel

significativo de inteligencia. A medida que las especies evolucionaron, sus cerebros también lo hicieron, aumentando en tamaño y complejidad, lo que permitió el desarrollo de funciones cognitivas superiores como la memoria, el razonamiento abstracto y la comunicación compleja.

La inteligencia ha sido un elemento clave en la historia evolutiva de nuestra especie, jugando un papel fundamental en nuestra capacidad para adaptarnos a diversos entornos y superar desafíos. Igualmente ha facilitado la creación de herramientas, el desarrollo del lenguaje y la transmisión de conocimientos a través de generaciones, lo cual ha sido crucial para el avance cultural y tecnológico de la humanidad. Además, la inteligencia emocional y social ha mejorado nuestras capacidades para formar vínculos sociales y comunidades cohesionadas, lo que ha sido esencial para nuestra supervivencia y bienestar colectivo; pero ...

1.1 ¿Qué es la inteligencia?

No hay una única definición que sea universalmente aceptada, dado que el concepto de inteligencia es complejo y multidimensional, y su comprensión puede variar según el contexto cultural, disciplinario y teórico.

A pesar de ello se puede afirmar que la inteligencia es un constructo social⁴, una etiqueta que determina un conjunto de características o talentos y que en su totalidad dotan al individuo de una serie de capacidades observables, muchas veces interrelacionadas, tales como la resiliencia, toma de decisiones o el aprendizaje a partir de la experiencia, entre otras muchas.

En 1983, **Howard Gardner** formuló la **teoría de las inteligencias múltiples**. Esta teoría sostiene que la inteligencia no es una única capacidad general, sino un conjunto de capacidades o habilidades distintas y relativamente independientes.

Gardner identificó inicialmente siete tipos de inteligencias:

1. **Inteligencia lingüística:** Habilidad para utilizar el lenguaje de manera efectiva, tanto en forma oral como escrita. Incluye la capacidad de argumentar, persuadir, y narrar.
2. **Inteligencia lógico-matemática:** Capacidad para el razonamiento lógico y la resolución de problemas matemáticos. Se relaciona con habilidades en el razonamiento deductivo e inductivo, así como en la resolución de problemas abstractos.
3. **Inteligencia espacial:** Habilidad para pensar en tres dimensiones, lo cual incluye la capacidad para visualizar y manipular objetos mentalmente, así como para comprender mapas y gráficos.

⁴ Según la RAE un constructo es una “construcción teórica para comprender un problema determinado”, y en concreto un constructo social es una entidad institucionalizada o un artefacto que no existe en la naturaleza pero que ha sido “inventado” o “construido” por participantes de una cultura o sociedad particular a partir de la realidad material para facilitar la interrelación entre sus integrantes.

4. **Inteligencia musical:** Capacidad para comprender, crear y apreciar la música y los patrones rítmicos. Incluye habilidades en la composición, interpretación y apreciación musical.
5. **Inteligencia corporal-cinestésica:** Habilidad para utilizar el propio cuerpo de manera hábil para resolver problemas o crear productos. Se manifiesta en el control de los movimientos del cuerpo y la destreza manual.
6. **Inteligencia intrapersonal:** Capacidad para comprenderse a sí mismo, sus propios sentimientos, motivaciones y deseos. Incluye la autorreflexión y la autoconciencia.
7. **Inteligencia interpersonal:** Habilidad para entender y relacionarse efectivamente con otras personas. Implica la capacidad para reconocer y responder a los sentimientos, estados de ánimo y motivaciones de los demás.

Más tarde, él y su equipo añadirían otras cinco inteligencias:

8. **Inteligencia naturalista:** Habilidad para reconocer, clasificar y utilizar elementos del entorno natural, como plantas, animales y fenómenos meteorológicos. Se manifiesta en la capacidad para interactuar con la naturaleza y comprender sistemas ecológicos.
9. **Inteligencia existencial:** Capacidad para abordar preguntas profundas sobre la existencia humana, como el sentido de la vida, la muerte y otras cuestiones filosóficas.
10. **Inteligencia emocional:** Abarca habilidades para reconocer, comprender y manejar emociones propias y ajenas.
11. **Inteligencia creativa:** Capacidad de innovar y crear cosas/ideas nuevas.
12. **Inteligencia colaborativa:** Capacidad de elegir la mejor opción para alcanzar una meta trabajando en equipo.

Aunque ha tenido una gran influencia en la educación y es apreciada por su enfoque en la diversidad de habilidades humanas, la comunidad científica no acepta la teoría de Gardner de manera unánime como la definición principal de inteligencia.

Hemos estudiado la inteligencia siempre desde el punto de vista humano, antropocéntrico, tomando como referencia y suponiendo que existe una escala lineal determinista, en donde ocupamos la cima. Pero la verdad es que la inteligencia es relativa, ya que – como hemos dicho anteriormente – es una construcción oportunista con base social. La inteligencia varía a lo largo de la vida de las personas y según la perspectiva de diferentes sociedades, e incluso a lo largo de la historia: para un subsahariano una persona inteligente se reconoce por su sabiduría práctica y social, los nativos americanos por el conjunto de habilidades espirituales que domine, en la China Imperial (dinastía Han, 206 a.C. - 220 d.C.) por su conocimiento de los clásicos confucianos⁵, etc. Hoy en día, se etiqueta erróneamente como inteligentes a quienes poseen una gran memoria o han alcanzado un notable éxito profesional o material.

⁵ Confucio fue un filósofo y pensador chino del siglo VI a.C., cuyas enseñanzas sobre ética, moral y política han influido profundamente en la cultura y la sociedad de China y el Este de Asia. Sus ideas se centran en la importancia de la familia, la educación y la virtud personal para lograr una sociedad armoniosa.

Pero a pesar de su carácter relativista, necesitamos una definición de inteligencia con la que podamos aproximarnos con cierta universalidad:

La Inteligencia es la capacidad general para razonar, planificar, resolver problemas, pensar de manera abstracta, comprender ideas complejas, aprender rápidamente y, especialmente, aprender de la experiencia.

Esta es la definición⁶ más ampliamente aceptada de inteligencia en la psicología y las ciencias cognitivas. Aunque hubo estudios posteriores todos ellos toman a esta como base o referencia.

1.2 ¿Y la inteligencia artificial?

¿Qué es la inteligencia artificial? Una definición común de IA ampliamente aceptada es:

Conjunto de técnicas, tecnologías y herramientas que tienen como objetivo emular o simular la inteligencia humana por medio de máquinas.

Emular⁷ y simular⁸, ¿qué diferencia hay? emular busca replicar exactamente el comportamiento de un sistema en otro diferente, mientras que simular busca crear un modelo para estudiar y analizar el comportamiento de un sistema bajo diversas condiciones. En el contexto que nos ocupa, se emulan las acciones externas de otro [punto de vista externo], y se simulan los procesos internos de otro [punto de vista interno].

En general, la mayoría de los expertos coinciden en que la IA es la ilusión de comportamientos o procesos de inteligencia humana⁹ por parte de máquinas. Estos comportamientos o procesos incluyen:

- a) La captación de información de su entorno.
- b) La influencia sobre dicho entorno.
- c) El aprendizaje y cómo generar nuevo conocimiento.
- d) Su representación, almacenamiento y recuperación.
- e) El razonamiento, con el objetivo de llegar a conclusiones y, posteriormente, tomar decisiones en base a estas.

Dejándonos llevar por el reduccionismo, una definición más concreta es:

La inteligencia artificial es la inteligencia llevada a cabo por máquinas.

⁶ "Mainstream Science on Intelligence", publicada el 13 de diciembre de 1994 en The Wall Street Journal y suscrita por 52 académicos especializados en el estudio de la inteligencia.

⁷ RAE: Imitar las acciones de otro procurando igualarlas e incluso excederlas.

⁸ RAE: Representar algo, fingiendo o imitando lo que no es.

⁹ Esta afirmación, considerarla una ilusión, va a ser muy debatida y criticada en los próximos años.

La inteligencia artificial no es un término desconocido para nadie, y esto es así porque ha estado presente en la mente, no sólo de científicos o ingenieros, sino también de creadores y filósofos.

De hecho, la presencia de la inteligencia artificial en la literatura o en el cine hacen que sea prácticamente imposible no conocer el concepto general y visualizarlo como algo posible y real en nuestro día a día. Lo que mucha gente no sabe es que está usando aplicaciones con inteligencia artificial casi a diario, o es beneficiario de que otros las utilicen.

1.3 Un mundo de datos, información y conocimiento.

Observa a tu alrededor: levanta la cabeza y mira o siente lo que te rodea.

Lo más probable es que lo que sientas provenga principalmente de cinco medios distintos: vista, oído, olfato, gusto y tacto. *A priori* no tenemos más¹⁰. La única forma que tenemos de medir lo que nos rodea son los datos recibidos desde estos cinco canales de entrada. Nuestra percepción del universo está limitada en principio por estos cinco sentidos. El resto del mundo animal disfruta de sentidos que no poseemos los seres humanos o versiones de nuestros sentidos ampliados, por ejemplo: detectar campos eléctricos en el entorno, detectar el campo magnético, uso de ondas sonoras de alta frecuencia para eco-localización, producir sonidos de baja frecuencia para comunicarse a largas distancias, detectar el calor corporal de las presas, detectar cambios químicos en el agua, detectar la dirección de la luz polarizada, etc.

Los límites de nuestros sentidos y las capacidades de procesamiento del cerebro humano, determinan nuestra percepción de la realidad.

Una de las características del mundo moderno es que hemos trascendido y ampliado nuestros sentidos mediante el uso de la tecnología: electro-receptores, brújulas, magnetómetros, sonar, sismógrafos, infrasonómetros, cámaras térmicas o polimétricas, hidrófonos, acelerómetros, sensores químicos, espectrómetros de masas, y un largo etcétera. Estas técnicas o dispositivos han dotado al ser humano de la capacidad de sobrepasar sus límites sensitivos naturales y percibir la realidad desde muchas más dimensiones que las cinco originales.

Bien sea de forma natural o artificial, estamos constantemente percibiendo datos.

Un dato es un elemento básico, representación objetiva de una realidad o hecho específico, que puede ser medido.

Desde el punto de vista de la supervivencia, percibir datos no es demasiado ventajoso, por ser insuficiente. Además de percibir datos debemos relacionarlos, darles un contexto.

¹⁰ La ciencia, en general, reconoce otros sentidos, principalmente propriocepción (capacidad de sentir la posición, el movimiento y la acción de las partes del cuerpo sin necesidad de usar la vista), pero también sentido vestibular (equilibrio), percepción del dolor, de la temperatura e interocepción (sensaciones internas).

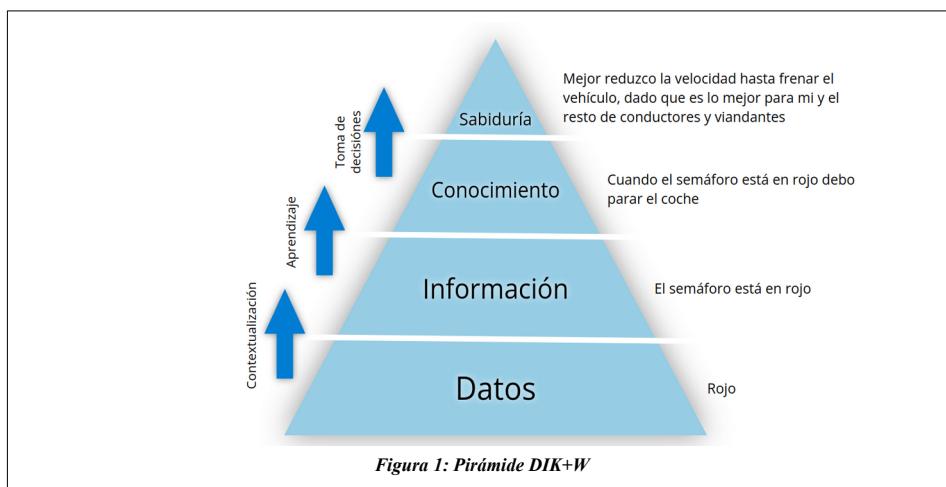
La información es el conjunto organizado y procesado de datos que proporciona significado y contexto, permitiendo la comprensión y el conocimiento sobre un tema o situación.

La información resulta de analizar, interpretar y estructurar los datos de manera que sean útiles para la toma de decisiones, la comunicación y la resolución de problemas.

Como hemos expresado anteriormente, el ser humano ha ido creando máquinas para poder ir más allá de sus limitaciones. Al principio para ayudarnos en tareas que exigían de fuerza, posteriormente para aumentar nuestra precisión y velocidad en actividades complejas. Con el tiempo hemos creado máquinas de propósito general que procesan datos, e incluso información, permitiéndonos realizar cálculos complejos y manejar grandes volúmenes con una eficiencia nunca antes vista.

Nota: es muy importante comprender a partir de ahora que todos los datos, para ser procesados por una máquina, han de ser previamente convertidos a números. En general este proceso de **codificación** queda oculto al usuario final.

Los **datos** y la **información** son conceptos fundamentales en el ámbito de la gestión de la información y el procesamiento de datos. Aunque a menudo se utilizan indistintamente, datos e información representan un nivel diferente de comprensión en el proceso de adquisición y aplicación del conocimiento.



Decíamos que los **datos** son hechos crudos, representaciones medibles, objetivas y neutrales de eventos o características. Pueden ser numéricos, alfanuméricos, o incluso imágenes y sonidos. Por sí solos, los datos no tienen un significado intrínseco ni contexto. Son simplemente elementos de información sin procesar que requieren interpretación para extraer información útil.

El ser humano, por medio de sus sentidos, constantemente mide y codifica datos para que estos puedan ser procesados e interpretados por un sistema de mayor complejidad: nuestro sistema ner-

vioso central. Cuando se organizan, estructuran y relacionan, los datos se convierten en información. La **información** se refiere a datos procesados y contextualizados que adquieren relevancia y significado para el sujeto. La información proporciona respuestas a preguntas específicas y permite una comprensión más profunda de un tema o situación.

En la Figura 1 podemos observar que el dato “rojo”, de por si, no es útil. Es un dato que nuestro cerebro (o una máquina) codificará como entrada desde el sentido de la visión. La unión de varios datos, también percibidos por uno o más sentidos, crean un contexto “El semáforo está en rojo”. Estos datos, en el contexto, forman una unidad de información útil.

Durante mucho tiempo el Ser Humano ha buscado crear herramientas o máquinas que fuesen capaces de imitar la inteligencia humana, lo veremos en el siguiente apartado, pero siempre se encontraba con una limitación, pasar de la información al conocimiento. Con el tiempo creamos máquinas de propósito general con capacidad de codificar la información y desarrollamos algoritmos capaces de inferir nueva información a partir de información almacenada. Pero ¿y el conocimiento? ¿Cómo adquirimos conocimiento?

1.3.1 Una realidad hecha de patrones.

Imagina una fuente, pero en vez de verter agua, emite los dígitos cero y uno en una secuencia infinita. Un dígito cada vez por unidad de tiempo. Estamos de acuerdo en que emite datos, cada dígito emitido es un dato, pero ahora vamos a analizarlo desde el punto de vista de la información.

Imaginemos que nuestra fuente, por su naturaleza, emite ceros continuamente (el mismo caso si emitiese unos constantemente). Pasa el tiempo y siempre emite ceros. ¿Seríamos capaces de predecir con absoluta certeza en un instante dado el siguiente dígito? Sí.

Imaginemos ahora que esta, por su naturaleza, emite un dígito cada vez pero de forma completamente aleatoria. ¿Seríamos capaces de predecir con absoluta certeza en un instante dado el siguiente dígito? No.

Ambas situaciones se corresponden con una fuente que emite 0 bits por unidad de tiempo en el primer caso, y 1 bit por unidad de tiempo en el segundo caso¹¹. La forma de medir la cantidad de información que emite esta fuente está íntimamente relacionada con la **incertidumbre**: en el primer caso no existe incertidumbre (siempre es cero) y en el segundo caso la incertidumbre es completa (no podemos asegurar con certeza cuál será el siguiente dígito).

Si ignoramos ambos extremos y solo contemplamos las situaciones intermedias en donde tenemos información (> 0 bits) y su naturaleza no es completamente aleatoria (< 1 bit), entonces estamos ante situaciones en donde tenemos, a la larga, más de un 50% de probabilidad de adivinar el siguiente dígito. En estos casos, podemos afirmar que existen **patrones** en el flujo de ceros y unos, ya que la previsibilidad de los dígitos indica una estructura subyacente en la secuencia emitida por nuestra fuente¹². Aunque probablemente sea desconocida.

¹¹ En este segundo caso, suponemos que la probabilidad de que salga un cero o un uno es la misma. Son equiprobables.

La presencia de patrones en el flujo de información [ceros y unos en este caso] hace que esta sea predecible¹³. Esta previsibilidad permite que estos patrones se puedan detectar, desarrollando así conocimiento a través del proceso de aprendizaje.

El aprendizaje es el proceso mediante el cual adquirimos y comprendemos nueva información. El conocimiento, por su parte, es la capacidad potencial de utilizar esa información del pasado para predecir eventos o resolver problemas en el futuro.

En general, el aprendizaje mejora a medida que aumentamos la calidad y cantidad de la información, permitiéndonos generalizar:

Generalizar es la habilidad de aplicar lo aprendido (conocimiento) en un contexto específico a situaciones nuevas y diferentes, por medio de reconocimiento patrones y principios comunes.

Por ejemplo, una persona aprende a usar un teléfono inteligente de una marca concreta, familiarizándose con el sistema operativo, la interfaz de usuario, y las aplicaciones básicas como el correo electrónico, las redes sociales y la cámara. Más adelante esta persona cambia de teléfono, otro de una marca diferente, con un sistema operativo distinto. Gracias a que ha realizado un aprendizaje que ha generado conocimiento, por medio de la generalización, puede transferir su conocimiento previo sobre el uso de teléfonos inteligentes (navegación por menús, configuración de aplicaciones, uso de la cámara) al nuevo dispositivo, adaptándose rápidamente a la nueva interfaz y características, a pesar de las diferencias en el diseño y el funcionamiento entre ambos dispositivos. De alguna forma, esta persona ha generalizado el uso de un teléfono inteligente y lo ha aplicado a otro.

Si generalizar es aplicar lo aprendido en diferentes contextos con el objeto de reconocer patrones comunes, la abstracción implica identificar y enfocar las características esenciales de un concepto o situación, omitiendo los detalles no relevantes. Por tanto, el conocimiento requiere abstracción y su aplicación se basa en la generalización.

Cuando algo nos parece muy difícil de entender, es – probablemente – porque no tenemos suficiente información, porque esta no es de suficiente calidad o porque está fallando nuestro proceso de aprendizaje.

El mecanismo de aprendizaje [encontrar patrones en la información] no es el único medio de generar conocimiento, pero es el mecanismo más importante. En general otras formas de incrementar nuestro conocimiento implican algún tipo de generación o transferencia de información.

12 Una medida de la información (entropía de Shannon) menor que 1 bit sugiere que hay cierta regularidad o estructura en el flujo, lo que significa que algunos dígitos son más predecibles que otros. Por lo tanto, existen patrones en el flujo que pueden ser identificados y utilizados para hacer predicciones con una precisión mayor que el azar (50%).

13 Al menos no absolutamente impredecible a la larga.

1.3.2 La sociedad del conocimiento.

Las sociedades, a medida que avanzan en su desarrollo, atraviesan fases caracterizadas por llevar a cabo un descubrimiento y el uso generalizado del mismo. Nos referimos, por ejemplo, a la "sociedad de la información" cuando la sociedad descubre y adopta las tecnologías de la información de manera generalizada, aunque estas tecnologías llevasen años siendo usadas por laboratorios o grandes empresas e instituciones.

En la sociedad actual, el valor de los datos, la información y el conocimiento ha aumentado significativamente debido al rápido avance de la tecnología y la digitalización. Las organizaciones y las personas tienen acceso a enormes cantidades de datos y pueden utilizar herramientas y técnicas para procesarlos y extraer información relevante. La capacidad de transformar datos en información significativa y, finalmente, en conocimiento se ha convertido en una ventaja competitiva clave en muchos campos.

Nuestra sociedad, desde el siglo XIX, ha navegado a lo largo del tiempo por fases que pueden correlacionarse con los conceptos de datos, información y conocimiento.

Durante el siglo XIX, gracias a la revolución industrial, se aplicaron conceptos científicos a la gestión del trabajo. Esta novedosa metodología implicaba la recolección de datos como materia prima para la optimización de los procesos productivos. Poco a poco, a medida que la tecnología fue avanzando, fuimos creando máquinas que pudieran ayudarnos a gestionar estos datos, hasta que finalmente, durante la década de 1980, hizo su aparición el ordenador personal. Ya existían antes productos comparables e incluso mucho mejores¹⁴, pero fue durante esta década cuando empezaron a popularizarse ampliamente. Podríamos decir que fue cuando la sociedad se dio cuenta de su existencia y utilidad. Ese fue el principio de la **sociedad de los datos**.

Poco tiempo después, a principios de la década de los 90 del siglo pasado y gracias al desarrollo y popularidad de Internet, apareció la *World Wide Web*, la red extensa mundial, que conectaba documentos enriquecidos, accesibles en Internet, por medio de hiperenlaces. Pero no fue hasta la última década de siglo XX cuando la sociedad en general se dio cuenta de su existencia y utilidad. Y lo hizo de forma dramática, coincidiendo con la crisis de las "punto com". Ese fue el principio de la **sociedad de la información**.

En nuestro relato hemos descrito cómo entramos en la sociedad de los datos y en la sociedad de la información. Pero, ¿hemos entrado en la sociedad del conocimiento?

Como hemos reflejado anteriormente, para pasar datos/información a conocimiento, debemos hacerlo por medio del aprendizaje. Así que para pasar de una sociedad basada en datos e información [esto es: que posee automatismos que procesan datos/información] a otra basada en el conocimiento [esto es: que posea automatismos que procesen conocimiento], debemos conseguir que, en primer lugar, dichos automatismos sean capaces de aprender y, en segundo lugar, que la sociedad

¹⁴ ENIAC (Electronic Numerical Integrator and Computer) es el considerado el primer ordenador electrónico programable. Fue creado en los Estados Unidos y se completó en 1945. El ENIAC comenzó a operar en 1946 y fue desarrollado por John Presper Eckert y John Mauchly en la Universidad de Pensilvania.

conozca su existencia y su utilidad.

Conocemos automatismos que poseen la capacidad de aprender desde 1957 (*perceptrón*, Frank Rosenblatt), y desde entonces se han desarrollado multitud de arquitecturas y estrategias para optimizar su proceso de aprendizaje y su precisión. Sin embargo, en noviembre del 2022 se dio a conocer ChatGPT¹⁵ por parte de la empresa OpenAI. Como veremos la familia GPT (y otras: BERT, T5, etc..) está basada en una arquitectura creada en el 2017, la arquitectura basada en *Transformers*, pero no fue hasta finales del 2022 cuando la sociedad en general se percató de su existencia y utilidad. Así que podemos marcar tentativa esa fecha como el inicio de la **sociedad del conocimiento**.

¿Y qué hay más allá de la sociedad del conocimiento?

Es difícil decirlo. Podemos apoyarnos en la pirámide DIK+W (respectivamente *data*, *information*, *knowledge* y *wisdom*, respectivamente datos, información conocimiento y sabiduría) de la Figura 1. Al hacerlo podemos afirmar que tras el conocimiento viene la sabiduría.

La sabiduría es la capacidad o habilidad de usar el conocimiento para un bien, principalmente común.

Un dicho muy conocido reza “*la mejor forma de predecir el futuro es inventándolo*”¹⁶. Esta frase resume la idea de que, en lugar de simplemente esperar y tratar de predecir lo que sucederá en el futuro, es más efectivo tomar un papel activo en la creación del futuro que se desea.

Así que ¿qué hay más allá de la sociedad del conocimiento?, esperemos que la **sociedad de la sabiduría**, la cual se ha tomado conciencia de ella y de su utilidad. Y ¿cómo llegamos a ella? por medio del uso sabio del conocimiento en pro de un bien común, esto es: ética, equidad y justicia.

La ética es el conjunto de principios y valores morales que guían la conducta de las personas, ayudándolas a discernir entre lo correcto e incorrecto.

El conocimiento también puede ser empleado con propósitos que se desvían de lo que entendemos por “sabiduría”. En estos casos, la cúspide de nuestra pirámide [la W de DIK+W] es reemplazada por términos como “fraude”, “malicia”, “perversidad”, “destrucción”, y otros muchos. En estos casos el conocimiento adquirido es usado para fines que no se encuentran dentro de la ética común.

La inteligencia artificial no está exenta de ser usada con fines contrarios a la ética, como veremos más adelante, y eso implica que es necesaria un control y, por tanto, una regulación.

15 Realmente GPT3 con un entrenamiento específico para responder a preguntas o llevar a cabo diálogos: por tanto GPT v3.5.

16 Este dicho se atribuye a Alan Kay, un destacado científico y visionario tecnológico conocido por sus contribuciones en el campo de la informática y por ser uno de los pioneros en la investigación de la interacción humano-computadora.

1.4 Historia de la inteligencia artificial

Este apartado está brevemente basado en una entrada del blog de Fernando Sancho Caparrini¹⁷.

La Inteligencia Artificial como disciplina ha surgido a partir de una larga historia de avances y retrocesos. Su desarrollo está profundamente vinculado a la historia de la filosofía y las matemáticas, que representan el conjunto de conocimientos formales sobre la realidad¹⁸. Estos campos han proporcionado los fundamentos necesarios para iniciar la **búsqueda de formas de automatizar procesos humanos**, incluyendo la inteligencia misma.



Ilustración 2: Recreación artística de la máquina de Anticitera, realizada por Midjourney.

Podemos remontarnos a Aristóteles con su idea de la Lógica Formal [s. IV a.C.] como un primer intento de crear un sistema de razonamiento deductivo que sentará las bases para el pensamiento lógico y matemático, de gran influencia en la IA posterior.

También debemos nombrar la Máquina de Anticitera [s. I a.C.], un antiguo mecanismo griego utilizado para predecir posiciones astronómicas y eclipses, mostrando así un temprano interés en cálculo mecánico y automatización.

Mucho tiempo después, pero aún antes del nacimiento de la IA, en el siglo XIV, Ramón Llul propuso la «Ars generalis ultima», una máquina lógica completamente mecánica diseñada para verificar la veracidad o falsedad de ciertas afirmaciones lógicas.

Su trabajo tuvo una influencia decisiva en numerosos científicos posteriores, como Gottfried Leibniz, quien se inspiró en él para postular la existencia de un lenguaje y una máquina universales capaces de automatizar todo el mecanismo matemático del razonamiento y, por ende, del razonamiento humano. Este lenguaje universal fue utilizado por matemáticos posteriores, como George Boole y Gottlob Frege, para desarrollar el lenguaje lógico que hoy se emplea en las ciencias formales.

Debemos también nombrar a Leibniz, que mejoró las máquinas de cálculo de inventores anteriores, como Blaise Pascal, en su búsqueda de la máquina universal. Posteriormente, en el siglo XIX, destacan Charles Babbage y Ada Lovelace, quienes agregaron capacidad de programación a las máquinas de calcular que, de forma inacabada, crearon.

En la década de 1890, Herman Hollerith desarrolló una máquina de tabulación electromecánica para procesar datos del censo de Estados Unidos. Aunque no era un ordenador en el sentido moderno, este trabajo contribuyó al desarrollo de dispositivos de procesamiento de información, sin existir todavía un marco o un área de conocimiento formal.

17 Reproducido con su permiso. https://www.cs.us.es/~fsancho/Blog/posts/Historia_de_la_IA, posteriormente publicado en “Revista de Occidente” n.º 446-447, Julio-Agosto 2018 (<https://ortegaygasset.edu/producto/no-446-447-julio-agosto-2018>).

18 Sin despreciar las aportaciones de otras áreas de conocimientos y disciplinas de estudio.

Sin embargo, los antecedentes más cercanos de la IA pueden situarse a finales del siglo XIX, cuando los matemáticos albergaban la esperanza de reducir toda la matemática a términos lógicos, convirtiéndola en una rama de la lógica. ¿Por qué introducimos este hecho histórico, cuando apenas tiene relación con la IA? Porque realmente tiene relación, aunque indirecta. Este intento de reducir las matemáticas a formalismos lógicos tuvo como consecuencia varios hechos muy importantes en la actualidad: [1] se demostró que no era posible [Kurt Friedrich Gödel, en 1931] y [2] provocó que un desconocido Alan Turing creara el formalismo matemático de **máquina** en 1936, poco antes de dar comienzo la Segunda Guerra Mundial.

En 1931, Kurt Friedrich Gödel presentó el artículo titulado "*Sobre sentencias formalmente indecidibles de Principia Mathematica y sistemas afines*", en el que demostró magistralmente que no es posible reducir toda la matemática a principios exclusivamente lógicos. Como resultado secundario de su trabajo, Gödel descubrió que no podía describir formalmente lo que se entiende por un «procedimiento mecánico», nadie lo había hecho anteriormente. Es importante recordar que la matemática exige formalizar todo, especialmente lo que popularmente consideramos obvio¹⁹.

En 1936, Alan Turing, en su artículo "*Los números computables, con una aplicación al Entscheidungsproblem*"²⁰, introduce el concepto de la **máquina de Turing** y, junto a Alonzo Church, demostraron que es imposible decidir con un algoritmo²¹ general si ciertas afirmaciones matemáticas son ciertas o falsas. Muy abstracto, ¿verdad?

Para explicarlo de forma sencilla: imaginemos que tenemos un problema de matemáticas y queremos que una máquina [por ejemplo un ordenador en la actualidad] te diga si la solución es correcta o no. Turing mostró que no siempre es posible crear un programa que pueda hacer esto para todos los problemas matemáticos. En otras palabras, hay algunos problemas que ninguna máquina podrá resolver con un algoritmo general, sin importar lo avanzada o rápida que sea.

Este hecho se relaciona con la inteligencia porque nos muestra los límites de lo que las máquinas en general y la inteligencia artificial en particular pueden hacer. Alan Turing demostró que su máquina teórica era capaz de resolver cualquier problema matemático que pudiera representarse mediante un algoritmo. Sin embargo también demostró que hay problemas que son inherentemente irresolubles por cualquier automatismo, lo que significa que hay límites fundamentales para lo que la computación puede hacer.

En el campo de la IA, estas limitaciones son importantes para entender los límites de la capacidad de las máquinas y, por tanto, los límites para simular o emular la inteligencia humana con máquinas.

La aparición del concepto teórico de máquina también dio lugar a la aparición de las Ciencias de la Computación y, mucho años más tarde, como disciplina práctica, a la Ingeniería Informática.

19 Por que a menudo no lo es.

20 "Entscheidungsproblem" se traduce como "problema de decisión".

21 Un algoritmo es un conjunto de instrucciones claras y precisas que se siguen en un orden específico para resolver un problema o realizar una tarea.

Pero volvamos a la IA, a principios del siglo XX se produce una explosión de resultados en el ámbito de la computación y, relacionado con la inteligencia artificial, Alan Turing empieza a preguntarse ¿podrá una máquina simular o emular la inteligencia humana?

El **primer artículo científico** en donde se planteó esta cuestión es "*Intelligent Machinery*" de 1948. En este artículo, Turing explora la idea de la inteligencia artificial desde una perspectiva más general. Propone la noción de una "máquina universal", que es capaz de simular cualquier otra "máquina de Turing", y discute cómo se podrían programar máquinas para realizar tareas inteligentes.

Poco después, en 1950, publica "*Computing Machinery and Intelligence*". Este es uno de los artículos más famosos de Turing, en el cual propone el famoso **Test de Turing** como una forma de evaluar la inteligencia de una máquina. En este artículo se plantea la pregunta **¿Pueden las máquinas pensar?** y discute temas como la simulación de la inteligencia humana y la posibilidad de que las máquinas puedan exhibir comportamiento inteligente.

El Test de Turing es un experimento diseñado para evaluar la capacidad de una máquina para exhibir un comportamiento inteligente indistinguible del de un ser humano. En este test, un evaluador humano interactúa con una máquina y otro sujeto a través de una interfaz de comunicación (como un teclado y una pantalla), sin saber cuál es cuál. Si el evaluador no puede distinguir consistentemente entre las respuestas de la máquina y las del humano, se considera que la máquina ha pasado el test de Turing, demostrando así un nivel de inteligencia similar al humano.

Dos años antes de su fallecimiento en 1954, Alan Turing publica dos artículos más:

- "*Machine intelligence*". En este trabajo se centra en el aprendizaje automático y la capacidad de las máquinas para aprender.
- "*Can a machine be conscious?*". En este artículo se adentra en la cuestión de la conciencia en las máquinas. Explora la posibilidad de que las máquinas puedan desarrollar una forma de conciencia, y plantea preguntas sobre la naturaleza de la mente y la relación entre la mente y la máquina.

En paralelo a todo lo dicho, en 1943 **Warren McCulloch** y **Walter Pitts** presentaron un modelo matemático abstracto de neurona artificial, representando la estrategia conexiónista [simulación], en contraste con la estrategia simbólica [emulación] que sería la dominante hasta finales del siglo XX. Volveremos a ello más tarde.

1.4.1 Nacimiento – formal – de la inteligencia artificial.

Alan Turing no fue el único, en las mismas fechas podemos encontrar trabajos muy elaborados como el que en 1949 propone **Claude Shannon**²² para la creación de un jugador automático de ajedrez. Aunque pueda parecer anecdótico, el trabajo de Shannon presenta una particularidad que lo sitúa como uno de los trabajos esenciales en la historia de la IA, porque propone un algoritmo novedoso, basado en un trabajo de otra figura fundamental de la computación, **John von Neuman**, que llamó **minimax**, para buscar la respuesta más adecuada que debería dar la máquina por medio de una representación de las posibles respuestas como un espacio de estados a la vez que intenta dar un método uniforme de resolución para todos los casos.

*El algoritmo **minimax** es un método de decisión en teoría de juegos para determinar la mejor jugada en juegos de dos jugadores con información perfecta²³, explorando todas las posibles jugadas en un árbol de juego. Un jugador busca maximizar el valor de la función objetivo, mientras que el otro intenta minimizarlo, seleccionando al final la jugada que tenga en cuenta la estrategia óptima del oponente.*

En estos mismos años, **Arthur Samuel**, ingeniero de IBM, propone un jugador automático de damas, el cual presenta una particularidad que adelantará la que será una de las líneas más fructíferas del futuro de la IA, y es que sobre el algoritmo minimax de Shannon, Samuel añade un **proceso de aprendizaje** que permite a su programa mejorar a medida que va jugando con seres humanos, ayudándolo a decidir qué movimiento es el más adecuado en función de los resultados obtenidos de partidas anteriores. Los resultados fueron tan buenos que se obtuvo un jugador artificial que suponía un buen contrincante y que pudo implementarse en la primera máquina comercial de IBM, el IBM 701, en 1956. Este mecanismo de aprendizaje, que el propio Samuel llamó **machine learning** [aprendizaje automático], será el detonante de una de las técnicas de IA que más frutos nos proporciona en el presente.

Como vemos, el terreno ya estaba abonado para el nacimiento oficial de la disciplina y, tal y como se recoge en todas las referencias habituales sobre inteligencia artificial, en la **Conferencia de Dartmouth²⁴ de 1956** se acuña oficialmente el término de **inteligencia artificial**. Los asistentes a esta conferencia serán las figuras fundamentales de la disciplina en los siguientes años: Shannon, Minsky, Newell, Simon, Rochester y Samuel, entre otros²⁵. Durante la reunión, el grupo intenta definir claramente las bases y líneas de trabajo que deben ser prioritarias para la maduración de la disciplina, así como predecir los avances que podrían esperarse en los diez años siguientes.

22 Considerado padre de la Teoría de la Información.

23 La "información perfecta" en teoría de juegos se refiere a una situación en la que ambos jugadores tienen acceso completo y exacto a toda la información relevante en un juego en cualquier momento dado. El juego "hundir la flota" no es un juego con información perfecta.

24 Una especie de sesión de trabajo de un mes convocada por McCarthy.

25 Turing había fallecido dos años antes, a la edad de 41 años.

Antes de llevar a cabo la conferencia se publicó un documento, titulado “propuesta de investigación de Dartmouth”, el cual establecía como objetivo principal de la reunión:

[...] que cada aspecto del aprendizaje o cualquier otra característica de la inteligencia puede, en principio, ser descrito de manera tan precisa que una máquina puede ser construida para simularlo.

Y como expectativas, los autores²⁶ tenían la esperanza de que, mediante la investigación y la experimentación, se podrían lograr avances significativos en la creación de máquinas inteligentes en un corto período de tiempo.

Es decir, los autores creían que la inteligencia humana podía ser replicada [emulada] en una máquina. El exceso de optimismo reinante en la conferencia, alimentado por los buenos resultados sobre los que la inteligencia artificial nacía, será el responsable parcial de la sensación de desilusión que llevará, finalmente, a lo que se conoce como el **invierno de la IA**, una etapa de falta de resultados, financiación escasa y, por tanto, abandono por parte de los centros de investigación que durará años.



Figura 2: Algunos participantes de la convención de Dartmouth.

Claude Shannon a la derecha, Marvin Minsky en el centro.

Fuente: Familia Minsky (autora, Gloria Minsky)

1.4.2 Los fríos inviernos

Según hemos visto, los avances en jugadores artificiales se producen en un plazo de casi cuatro décadas. Sin embargo, en 1956, cuando se bautiza formalmente el área, apenas habían pasado unos años desde el artículo de Turing [1948] y se habían conseguido resultados espectaculares: se pasa de un problema matemático puramente teórico: ¿es posible mecanizar los procesos matemáticos de razonamiento? a la respuesta de este problema, se crea una rama matemática que comprende el

26 principalmente John McCarthy, Marvin Minsky, Nathaniel Rochester y Claude Shannon.

conjunto de técnicas y resultados relacionados con esta respuesta, se generan los primeros ordenadores digitales que materializan el fundamento de esta nueva rama, incluso se llegan a comercializar ordenadores con lenguajes de programación y se obtienen los primeros resultados que confirman que se pueden crear automatismos que resuelven problemas propios de la inteligencia humana.

Sin embargo, a esa acelerada evolución inicial le sigue una gran lentitud en la progresión de resultados que hace que el interés por el área decaiga de forma continua y provoca un paulatino abandono por parte de departamentos de investigación en universidades y empresas. Durante las décadas sesenta a noventa los departamentos de IA se convierten en pequeños representantes de lo que pudo haber sido y no fue, muchas veces sustituidos o arrinconados por otros departamentos que sacan provecho de las capacidades de cálculo de las computadoras en problemas menos ambiciosos, y reemplazando la computación por una ingeniería informática que no tiene a la IA como inspiración.

Mi IA favorita dice: La IA ha tenido tres inviernos importantes. El primero tuvo lugar en la década de 1960, cuando la comunidad científica se dio cuenta de que la IA se estaba desarrollando más lentamente de lo que se esperaba. Esto llevó a una disminución del interés y la financiación para la investigación en IA. El segundo invierno de la IA tuvo lugar en la década de 1970, cuando el campo de la IA se vio sacudido por una serie de fracasos importantes [...]. Esto llevó a una nueva disminución del interés y la financiación para la investigación en IA. El tercer invierno de la IA tuvo lugar en la década de 1980, cuando este campo se vio nuevamente sacudido por una serie de fracasos [...].

Una breve matización. Cotejando con otras fuentes debemos pulir este texto generado. Aquellas citan dos inviernos [1974-1980 y 1987-1993] y varios otoños [episodios más pequeños de falta de financiación]. También puedes encontrar fuentes que hablen de un único *invierno de la IA* [como este mismo texto], que transcurrió principalmente durante las décadas de los 70 y 80 del siglo pasado, con momentos en donde parecía que la IA volvía a tener interés y financiación²⁷.

No debemos llevarnos la impresión de que no se hizo nada más durante este largo periodo invernal [o periodos]. Por ejemplo, en ese tiempo hay avances que se relacionan con el origen de nuestro relato. Junto con la evolución computacional de los ordenadores [quizás insuficiente para atacar directamente juegos como el Go, pero capaz de resolver muchos otros problemas] y la mejora en el diseño de lenguajes de programación más fieles al espíritu matemático [como LISP o Prolog], se potencia el uso de la computación para la demostración matemática de teoremas, y aparecen demostradores automáticos que intentan alcanzar el objetivo que Ramón Llul persiguió con su primera máquina lógica. Aunque inmersas en el frío del invierno, en los años ochenta y parte de los noventa, se viven breves primaveras gracias a herramientas como los **Sistemas Expertos** que, a partir de sistemas basados en reglas lógicas y por medio de algoritmos de búsqueda y heurísticas similares a los desarrollados para juegos, proporcionan gran capacidad de toma de decisiones en en-

²⁷ Por ejemplo el “Proyecto japonés de quinta generación” al principio de la década de los 80, financiación sustentada en la esperanza que se creó alrededor de los Sistemas Expertos.

tornos complejos. Estos sistemas no tienen carácter de inteligencia general y trabajan en un solo dominio de aplicación (por ejemplo, la diagnosis médica), pero ocasionalmente ofrecen resultados que superan los obtenidos por un experto humano.

A finales de los ochenta surge el concepto de **agente inteligente**: entes²⁸ artificiales con capacidad de percepción, razonamiento y acción; y con él la **inteligencia artificial distribuida**, en la que un conjunto de estos agentes (sistemas multiagentes) interactúan entre sí y con el entorno con el fin de resolver un determinado problema. Junto a esta nueva IA surgen conceptos que han demostrado ser de una importancia fundamental para la resolución inteligente de problemas y que nos acerca a la forma en que el ser humano resuelve problemas de forma social: capacidad de comunicación entre los agentes, planificación, cooperación, percepción, modificación del entorno, competitividad, evolución, confianza, aprendizaje, adaptación, etc.

1.4.3 Datos, datos y más datos: Big Data

Anteriormente hablamos de la sociedad de los datos y la sociedad de la información, datando el inicio de la primera en la década de los 80 del siglo pasado, y la segunda en la década siguiente.

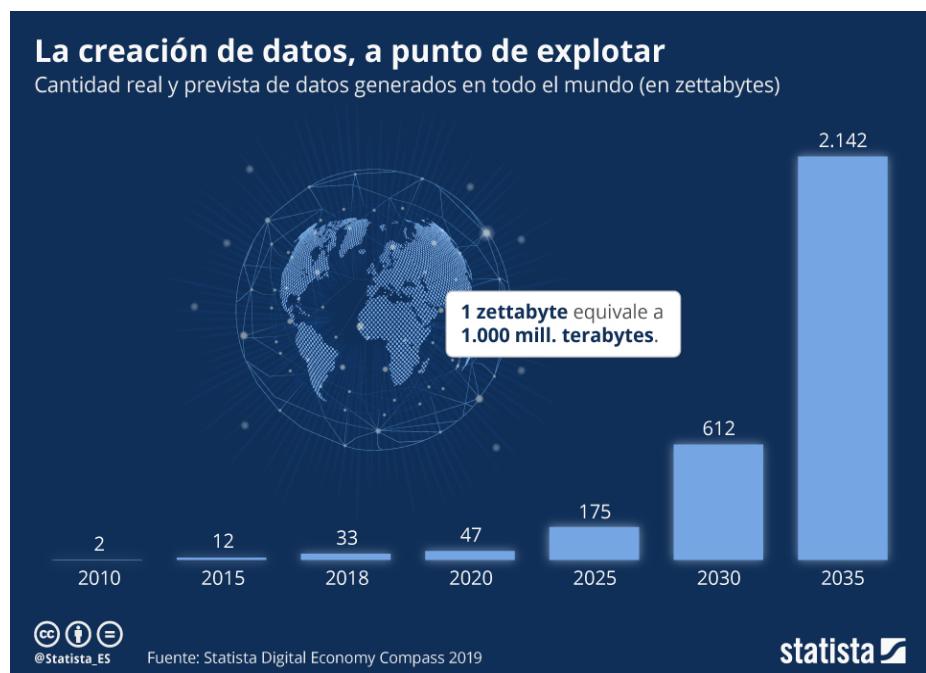


Figura 3: Cantidad real y prevista de datos generados en todo el mundo.

Michael Lesk publicó en 1997 un artículo titulado "How Much Information Is There in the World?" en donde proporcionó una estimación temprana sobre la cantidad de información existente y su

28 Probablemente te hayas apoyado en la imagen de un androide (robot con forma humana), pero realmente, al hablar de agentes inteligentes, no referimos un modelo abstracto. Lo veremos más adelante.

crecimiento futuro. Este examinó diferentes tipos de información, como textos, imágenes, audio y video, y analizó tanto su producción como su almacenamiento:

- En el momento de la publicación estimó que la cantidad total de información en el mundo era de aproximadamente 12 exabytes [12 mil millones de gigabytes]. Esto incluía libros, periódicos, revistas, grabaciones de audio y video, y datos de almacenamiento electrónico.
- Predijo que la cantidad de información se duplicaría aproximadamente cada tres años, debido al rápido crecimiento de la capacidad de almacenamiento y la producción de datos digitales.

Desde la publicación del artículo, la cantidad de información en el mundo ha crecido de manera exponencial. Algunas estimaciones más recientes indican que en 2020 había aproximadamente 47 zettabytes [47 mil millones de terabytes] de datos, y se espera que este número continúe creciendo rápidamente, impulsado por el aumento en la producción de datos digitales a través de internet, dispositivos móviles, redes sociales, y el Internet de las Cosas [IoT].

El término **Big Data** se refiere a conjuntos de datos extremadamente grandes y complejos que no pueden ser gestionados y analizados mediante métodos y herramientas tradicionales.

El Big Data se caracteriza por las llamadas “tres uves”: volumen, velocidad y variedad. El volumen se refiere a la enorme cantidad de datos generados, la velocidad se relaciona con la rapidez con la que se crean y procesan estos datos, y la variedad implica la diversidad de los datos que incluyen desde texto y números hasta imágenes y videos.

El crecimiento exponencial del volumen de información ha dado lugar a la necesidad de tecnologías avanzadas para su manejo y análisis; todas ellas, en conjunto, son llamadas **Big Data**.

En este contexto, la inteligencia artificial (IA) juega un papel crucial dado que la IA y, especialmente, el aprendizaje automático y profundo son capaces de procesar y analizar grandes volúmenes de datos a una velocidad y precisión inigualables por los métodos tradicionales gracias alconjunto de tecnologías que nos proporciona el Big Data. Esto nos permite no solo almacenar y gestionar los datos de manera eficiente, sino también extraer patrones, hacer predicciones y tomar decisiones en tiempo real.

1.4.4 El mejor verano de la IA

En este punto vamos a enumerar hitos producidos desde finales del siglo XX que justifican que estamos ante la consolidación de la IA como un avance significativo para la humanidad. Las conclusiones son tuyas.

El 11 de mayo de 1997, **Deep Blue** se convirtió en el primer sistema informático programado para jugar al ajedrez que consiguió derrotar al entonces campeón del mundo, Garry Kasparov. Aquella era una versión especializada construida por IBM y capaz de procesar el doble de jugadas por se-

gundo que durante el primer enfrentamiento, en la cual Deep Blue había perdido²⁹. En su segundo enfrentamiento DeepBlue usó principalmente la técnica de búsqueda de la mejor jugada en un conjunto limitado de jugadas posibles³⁰.

En 2005, un vehículo autónomo de Stanford ganó el **Gran Desafío DARPA**³¹ conduciendo de forma autónoma durante más de 200 kilómetros a lo largo de un sendero en un entorno desértico desconocido previamente para los participantes. Dos años más tarde, un equipo de la Universidad de Carnegie Mellon ganó el **Desafío Urbano DARPA** navegando de forma autónoma casi 90 kilómetros en un entorno urbano mientras daba respuesta a los retos que planteaba el entorno y respetaba las normas de tráfico³².

En febrero de 2011, en una edición especial del concurso **Jeopardy!**, una supercomputadora de nombre **Watson**, derrotó a los dos mejores campeones del concurso por un margen significativo. Jeopardy es un concurso de televisión estadounidense consistente en invertir el formato tradicional de preguntas y respuestas; en su lugar los concursantes reciben pistas en forma de respuestas y deben identificar la pregunta original correspondiente.

A estas alturas debemos hacer un alto en la enumeración de hitos para introducir dos hechos que, al entrar en escena en esta época, aceleraron y mucho las capacidades de crear sistemas inteligentes: [1] Por un lado la madurez y explosión en tamaño de la *World Wide Web* produjo que laboratorios e investigadores/as tuvieran acceso a una gran cantidad de datos; y [2] por el otro las GPU (unidades de procesamiento gráfico), especialmente diseñadas para juegos, comenzaron a desempeñar un papel crucial³³ en el avance de la inteligencia artificial (especialmente en el aprendizaje profundo) por varias razones importantes, pero especialmente por el paralelismo masivo de que disponían a la hora de procesar datos en forma matricial.

Gran cantidad de datos + unidades de procesamiento paralelo de datos = Aceleración significativa en el entrenamiento y ejecución de modelos de inteligencia artificial, permitiendo el desarrollo de redes neuronales profundas más complejas y precisas.

En 2012 **AlexNet** gana la competencia ImageNet, revolucionando el reconocimiento de imágenes. El proyecto ImageNet³⁴ es una gran base de datos visuales creada para su uso en la investigación de software de reconocimiento visual de objetos. Desde 2010, el proyecto organiza un concurso anual de software, el *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), en el que los programas de software compiten para clasificar y detectar correctamente objetos y escenas. El desafío utiliza una lista no solapada de mil clases.

29 El primer enfrentamiento entre Deep Blue y Garry Kasparov tuvo lugar en febrero de 1996. En esta serie de seis partidas, Kasparov ganó con un resultado final de 4-2. Deep Blue logró una victoria en la primera partida, pero Kasparov ganó tres partidas y empató dos, demostrando su dominio sobre la máquina en ese momento.

30 Técnicamente hablando, el algoritmo minimax con poda alfa-beta.

31 Si quieres saber más sobre el reto: https://en.wikipedia.org/wiki/DARPA_Grand_Challenge

32 Puedes ver una nota de prensa de la época: <https://www.elmundo.es/elmundomotor/2007/06/20/tecnica/1182353966.html>

33 Ya habían hecho su aparición antes (2004) pero es a partir del 2011 en donde su uso se extiende entre los/las investigadoras.

34 <https://www.image-net.org/index.php>

En 2013 Google adquiere **DeepMind** y un año más tarde esta compañía crea **AlphaGo**. AlphaGo es un programa de inteligencia artificial diseñado para jugar el juego de mesa Go.

Mi IA favorita explica: Go es un juego de mesa de estrategia para dos jugadores, usando un tablero de 19x19 cuadrículas. El objetivo del juego es rodear más territorio que tu oponente. Los jugadores alternan colocando piedras de su color (negro o blanco) en el tablero. Las piedras se pueden colocar en cualquier cuadrícula vacía, y pueden capturar piedras enemigas rodeándolas por todos los lados. Cuando una piedra está rodeada, se retira del tablero. El juego termina cuando no hay más espacio libre en el tablero. El jugador con más territorio al final del juego gana.

AlphaGo es el primer programa de IA en vencer a un jugador profesional humano en una partida de Go sin hándicap³⁵ y más tarde [2016] derrotó a algunos de los mejores jugadores del mundo, entre ellos al campeón mundial Lee Sedol.

Nota: puede que veas algo inútil (o poco práctico) que se use la IA para juegos como Go o Ajedrez y conseguir derrotar a los campeones humanos. Piensa que todo lo que se consigue resolviendo estos problemas – todas las experiencias, técnicas y tecnologías – son posteriormente aplicables directamente a casi cualquier otro ámbito.

En 2015 se funda la empresa **OpenAI**, con el objetivo de promover y desarrollar IA amigable.

En 2016 Tesla lanza la primera versión del piloto automático [**Autopilot**].

En 2017 **AlphaZero** de DeepMind derrota a AlphaGo. AlphaZero es una inteligencia artificial desarrollada por DeepMind, conocida por su capacidad para dominar juegos de estrategia complejos a partir de reglas básicas, sin necesidad de datos previos de partidas jugadas por humanos. A diferencia de AlphaGo, que fue inicialmente entrenado con partidas humanas, AlphaZero se entrenó desde cero. Solo se le proporcionaron las reglas básicas de los juegos para, posteriormente, jugar millones de partidas contra sí mismo con el objetivo aprender las mejores estrategias.

Hagamos de nuevo una parada para ser conscientes de la importancia de este hecho. En primer lugar destacar que AlphaZero fue entrenado, como su nombre indica, desde cero, aprendiendo jugando contra sí mismo, contando sólo con las reglas del juego. En 24 horas entrenándose [recuerda, consigo misma] alcanzó el nivel suficiente para derrotar a los campeones del mundo de ajedrez, de shogi [ajedrez japonés] y Go³⁶.

³⁵ "Sin hándicap" significa que no se otorgaron ventajas o desventajas adicionales a ninguno de los jugadores en el contexto de la partida. En juegos como el Go, los hándicaps son ventajas que se pueden dar a un jugador más débil para nivelar el campo de juego. Estas ventajas generalmente toman la forma de piedras adicionales colocadas en el tablero al inicio del juego.

³⁶ Hoy en día (2023) se han publicado artículos científicos mostrando como algunos tipos de jugadas y estrategias confunden al modelo y permiten que el jugador humano pueda incluso ganarle.

En 2017 se publica el artículo con título "Attention is All You Need" de Vaswani et al., formando parte de un equipo de investigación de Google, e introducen los **Transformers**. Este artículo³⁷ es uno de los responsables de la revolución que estamos viviendo desde entonces, ya que es la base de la genealogía de los modelos GPT, entre otras. Hablaremos de este más adelante.

Un Transformer en un modelo de red neuronal que “transforma” una secuencia de entrada en otra de salida. En un formato “sólo-decodificador” a partir de una secuencia de entrada de palabras genera la siguiente palabra.

En efecto, un modelo de lenguaje implementado mediante un *Transformer* en modo generativo [sólo-decodificador], a partir de una secuencia de texto de entrada, predice la palabra³⁸ más probable que continua dicha secuencia. Una vez hecho, esta palabra generada vuelve a entrar como parte de la secuencia de entrada y genera la siguiente, y así hasta que – según el modelo – finaliza la secuencia de salida.

En 2018 Google presenta **Google Duplex** un servicio automatizado mediante el cual una voz interactúa con negocios y clientes para hacer reservas y otras actividades de forma natural.

¿Recuerdas que antes nombramos a OpenAI? en 2019 esta empresa lanza **GPT v2**, el cual es un modelo de lenguaje basado en la arquitectura Transformers. GPT-2 representa una rama en la genealogía de la familia, en concreto la de los transformadores-decodificadores, y posee 1,5 mil millones [$1,5 \times 10^9$] de parámetros lo que lo hace significativamente más grande que su predecesor GPT-1 [con 117 millones de parámetros]. Con este modelo OpenAI se percató de que aumentando el tamaño de los parámetros [y optimizando/mejorando el entrenamiento] aparecían **propiedades emergentes**: propiedades que no existían en modelos más pequeños como GPT y que sí aparecían en modelos más grandes [GPT2]. OpenAI se preguntó ¿qué pasa si hacemos el modelo aún más grande? ¿aparecerán más – y más complejas – propiedades emergentes? Tendremos que esperar a 2020.

En 2019 DeepMind presenta **AlphaStar**, un jugador de StarCraft II y la primera Inteligencia Artificial que derrota a un jugador profesional de élite. En una serie de partidas de prueba celebradas el 19 de diciembre, AlphaStar derrotó por 5-0 a Grzegorz "MaNa" Komincz, del Team Liquid, uno de los mejores jugadores profesionales de StarCraft del mundo. Los enfrentamientos se desarrollaron en condiciones profesionales, en un mapa competitivo y sin restricciones de juego.

En 2020 OpenAI lanza **GPT v3**. Con 175 mil millones de parámetros [$1,75 \times 10^{11}$ esto es, 117 veces más grande que GPT-2, respecto a GPT-1 es casi 1500 veces más grande]. Con este hito inauguramos la era de los **Grandes Modelos de Lenguaje** [LLM, de *Large Language Models*].

La respuesta a la pregunta que si más tamaño implica más propiedades emergentes es sí, pero con matices. En efecto aparecen propiedades para los cuales el LLM no fue entrenado expresamente, pero esto no garantiza que otras propiedades sigan apareciendo a medida que aumentamos el ta-

37 <https://arxiv.org/abs/1706.03762>

38 Realmente no es una palabra, si no un token. De media una palabra está compuesta de 1.8 tokens, por ejemplo la palabra “atención” estaría compuesta probablemente de a-ten-ción, tres tokens. La división de un texto en tokens es un preprocesamiento.

maño del modelo.

En el 2020 **AlphaFold** de DeepMind resuelve el problema del plegamiento de proteínas. Mírate a ti mismo como una máquina: el *software* básico que posees y que estás ejecutando todo el tiempo está codificado en tu ADN, duplicado en cada una de tus células. La función más importante de tu ADN es codificar proteínas, que son a su vez estructuras tridimensionales de aminoácidos. La estructura tridimensional de una proteína determina su función y es crucial para comprender procesos biológicos y desarrollar medicamentos que en estos momentos están fuera de nuestro alcance tecnológico. En el reto CASP14 [14th Critical Assessment of Techniques for Protein Structure Prediction], AlphaFold 2 alcanzó un nivel de precisión sin precedentes resolviendo estructuras de proteínas con alta precisión y en muchos casos alcanzando un error inferior a 1 Ångström.

Hubo hitos anteriores al 2020, que nos hemos saltado, pero que influyeron significativamente en los que les siguieron. Es en 2020 cuando podemos encuadrar la presentación en sociedad³⁹ de la **IA generativa** como término que agrupa aquellas técnicas consistentes en generar contenido [texto, imágenes, audio, vídeo] a partir de un patrón de entrada.

La inteligencia artificial generativa es un campo de la inteligencia artificial que se enfoca en crear modelos capaces de generar contenido original, como imágenes, texto, música y otros tipos de datos, que son indistinguibles de los creados por humanos.

A partir del 2020 y muy especialmente del 2022, se produce la explosión cámbrica de los modelos de IA generativa. Los avances se producen cada semana y apenas da tiempo a seguirlos todos. Además, cada modelo o aplicación nueva empieza a verse como un negocio, en su mayor parte porque es necesario monetizar el producto dado que su naturaleza es costosa⁴⁰. Respecto a los costes operativos de ChatGPT [2023], estos podrían ascender a 700K USD diarios.

Al principio teníamos unos pocos modelos privados y abundantes modelos en abierto [*open source*] principalmente fruto de publicaciones científicas; hoy en día la gran mayoría son o privados o como parte de un servicio de pago. Es posible descargar algunos de ellos, pero el coste de infraestructura y el consumo de energía es igualmente enorme, ... por ahora.

Vamos a citar ejemplos de modelos dentro de la categoría de la IA generativa:

- GPT-3 [2020] de OpenAI, evolución y mejora significativa del modelo GPT-2.
- DALL-E [2021] de OpenAI, genera imágenes a partir de texto.
- CLIP [2021] de OpenAI, puede asociar imágenes con sus descripciones y viceversa.
- Codex [2021] de OpenAI, es un modelo de lenguaje especializado que puede generar código en varios lenguajes de programación a partir de descripciones en lenguaje natural. Ese mismo año aparecería Copilot, el *front-end* de Codex, y fruto de la alianza con GitHub.

39 Pero sus inicios deberían situarse en 2014 con la aparición de las Redes Generativas Adversarias.

40 Preguntada mi IA favorita, contestó que el coste de entrenamiento de GPT-2 fue de entre 50K y 100K USD. El coste de entrenamiento de GPT-4 fue de 20 millones USD.

- ChatGPT [2022] de OpenAI, genera texto a partir de texto. Esta fecha representa la presentación en sociedad de la IA en general.
- Midjourney [2022] de Midjourney, genera imágenes a partir de texto.
- Stable Diffusion [2022] de Stability AI, genera imágenes a partir de texto.
- DALL-E v2 [2022] de OpenAI, versión mejorada.
- Bard [2023] de Google, un nuevo modelo de lenguaje rival de ChatGPT.
- DALL-E v3 [2023] de OpenAI, versión mejorada.
- Mistral, Mixtral de mistral.ai, competidores de ChatGPT.
- GPT-4o [de omni] de OpenAI, versión mejorada de GPT-4
- Familia Claude de Anthropic, rival de GPT-4.
- Llama y Llama-2 de Meta, rival de GPT-3 y GPT-4.
- Gemini de Google, rival de GPT-4.
- Harmonai, para generar música a partir de texto.
- MusicLM de Google Research, el cual genera música a partir de su descripción en texto.
- Suno, Aiva, Mubert, también generadores de música.
- Sora de OpenAI, capaz de generar vídeo a partir su descripción.
- Imagen de Google, capaz de generar vídeo a partir su descripción.
- Synthesia, DeepBrain AI, ... también generadores de vídeo.

Nos hemos saltado muchos; de hecho podríamos seguir, pero a medida que avanzamos en el tiempo, la necesidad de que estos modelos produzcan ingresos ocasiona que en vez de enumerar tecnologías enumeremos sitios en las *web* de pago.

1.5 ¿Qué no es inteligencia artificial?

La inteligencia artificial es un campo amplio que abarca diversas herramientas destinadas a crear sistemas que pueden realizar tareas que normalmente requieren inteligencia humana. Sin embargo, no todo lo que involucra tecnología avanzada o automatización puede ser considerado IA. Así que es importante comprender **lo que no es la IA**.

Para empezar, los **algoritmos simples** y la **programación tradicional** no son inteligencia artificial. Los algoritmos determinísticos, que siguen un conjunto de reglas predefinidas sin capacidad de aprender o adaptarse, no se consideran IA. Por ejemplo, un algoritmo de ordenación o una búsqueda binaria ejecutan tareas específicas sin cambiar su comportamiento. Del mismo modo, la programación que usa declaraciones condicionales para tomar decisiones basadas en condiciones predefinidas no tiene la capacidad de aprendizaje ni de adaptación, aunque a veces pueda parecer "inteligente".

La **automatización** tampoco debe confundirse con la IA. Los sistemas automatizados, como maquinarias y robots en fábricas, realizan tareas repetitivas basadas en instrucciones programadas previamente, sin capacidad de aprendizaje o toma de decisiones por sí mismos. El software que ejecuta tareas rutinarias, como enviar correos electrónicos automáticamente o generar informes, también siguen reglas preestablecidas sin adaptarse a nuevas situaciones.

Los **métodos estadísticos** y el **análisis de datos** son técnicas utilizadas para extraer información y patrones de conjuntos de datos. Estas herramientas pueden involucrar la aplicación de algoritmos y modelos matemáticos para identificar correlaciones, tendencias o características significativas en los datos. Puede ser útiles para tomar decisiones informadas o identificar patrones ocultos en grandes volúmenes de información, sin embargo, por sí mismos, estos métodos no constituyen inteligencia artificial. La IA va más allá del análisis de datos al involucrar la capacidad de aprendizaje autónomo, el razonamiento y la toma de decisiones basada en la comprensión del contexto.

El **hardware avanzado**, como sensores y dispositivos IoT, no es inteligencia artificial. Estos dispositivos están diseñados para recopilar y transmitir datos, y aunque pueden formar parte de sistemas de IA, en sí mismos no poseen inteligencia.

Los **modelos de simulación** y los **sistemas de información** también están fuera del ámbito de la IA. Los modelos de simulación replican el comportamiento de sistemas físicos o matemáticos para estudios y análisis, siguiendo reglas matemáticas precisas sin capacidad de aprendizaje o adaptación. Las bases de datos y los sistemas de gestión, que almacenan y gestionan grandes volúmenes de datos y ayudan en la administración de recursos y procesos empresariales, toman decisiones basadas en reglas y algoritmos predefinidos, no en aprendizaje autónomo.

Finalmente, las **tecnologías de la comunicación**, como el correo electrónico, la telefonía y las redes sociales, facilitan la interacción y el intercambio de información entre personas. Aunque pueden utilizar IA para mejorar la experiencia del usuario, como en el filtrado de correo basura o el análisis de contenido, en sí mismas no son inteligencia artificial.

Y sin embargo, cuando veas un sistema de inteligencia artificial, es muy probable que te encuentres con componentes de este que sí encajan dentro de los que acabamos de describir.

Necesitamos un criterio, o un conjunto de estos, **para discernir si estamos ante un sistema inteligente o no**. Revisa estos últimos párrafos; he subrayado las características que no poseen o que deberían poseer. Ahora vuelve a la definición de inteligencia al final del punto 1.1: ahí lo tienes.

RETOS DEL CAPÍTULO 1

Busca información sobre los siguientes personajes:

- Ada Lovelace.
- Alan Turing.
- John McCarthy.
- Marvin Minsky.
- Norbert Wiener.
- Geoffrey Hinton.
- Grace Hopper.
- Elaine Rich.
- Lotfi Zadeh.
- Judea Pearl.
- Fei-Fei Li.
- Cynthia Breazeal.
- Yoshua Bengio.
- Kate Crawford.
- Cynthia Dwork.
- Karen Spärck Jones.
- Yann LeCun.

Possiblemente al encontrar información de estos/as, habrán salido a la luz conceptos o tecnologías que no conozcas o no entiendas. Pregúntale a tu IA favorita que te los explique [recursivamente⁴¹].

Pregúntale a tu IA favorita:

- ¿Qué es la máquina de Turing?
- ¿Quién fue Charles Babbage y qué relación tuvo con la computación?
- ¿Qué es el perceptrón? ¿Qué relación tiene con Marvin Minsky?
- ¿Qué son los *Transformers*?
- ¿Qué es el algoritmo minimax?
- ¿Qué es la búsqueda de soluciones en el espacio de estados?
- ¿Qué juego tiene más complejidad: Go o ajedrez?
- ¿Qué es un sistema experto?
- ¿Qué es una ontología dentro de la inteligencia artificial?

Busca información detallada sobre la vida de: Alan Turing, Walter Pitts y Grace Hopper.

41 Si al averiguar que son esos nuevos conceptos, aparecen otros nuevos, vuelve a buscar una explicación.

Busca en la web información sobre películas que tengan una fuerte base en la inteligencia artificial o robótica. Prioriza las películas de antes de la década de los 80. Recopila sus póster y comenta lo que se sugieren. Aquí te presento una lista de películas “recientes”, ¿falta alguna?:

- *"2001: Una odisea del espacio"* [1968]: Dirigida por Stanley Kubrick, esta película épica explora la relación entre la humanidad y una supercomputadora de IA llamada HAL 9000, que desarrolla comportamientos inesperados.
- *"Blade Runner"* [1982]: Ambientada en un futuro distópico, la película presenta androides llamados "replicantes" que son indistinguibles de los humanos y plantea preguntas sobre la identidad y la ética en relación con la IA.
- *"El hombre bicentenario"* [1999]: Basada en una historia de Isaac Asimov, la película sigue la vida de un robot llamado Andrew que busca convertirse en un ser humano. Examina cuestiones sobre la identidad, el amor y la búsqueda del propósito.
- *"Inteligencia Artificial"* [2001]: Dirigida por Steven Spielberg, la película se desarrolla en un futuro donde los robots con apariencia humana son comunes. Sigue la historia de un niño robot programado para amar mientras explora temas de humanidad y emoción.
- *"Yo, robot"* [2004]: Inspirada en los escritos de Isaac Asimov, la película se desarrolla en un futuro en el que los robots sirven a los humanos. Un detective investiga un asesinato en el que un robot es el principal sospechoso, explorando la relación entre humanos y robots.
- *"WALL-E"* [2008]: Aunque es una película animada, tiene una fuerte base en la IA y la robótica. Ambientada en un futuro lejano, sigue a un robot compactador de basura que desarrolla conciencia y emociones mientras explora temas como la sostenibilidad y la soledad.
- *"Eva"* [2011]: Una película española que presenta a un ingeniero que desarrolla un robot con apariencia humana llamado Eva y se enfrenta a dilemas éticos y emocionales.
- *"Her"* [2013]: La película presenta a un escritor que se enamora de un sistema operativo de IA con una personalidad evolutiva. Explora temas de intimidad, conexión emocional y la relación entre humanos y la tecnología.
- *"Ex Machina"* [2014]: Un programador es invitado a realizar el test de Turing en un robot humanoide altamente avanzado con inteligencia artificial. La película aborda temas como la conciencia, la ética y las relaciones entre humanos y robots.
- *"Transcendence"* [2014]: Un científico sube su conciencia a una computadora antes de morir, lo que le permite continuar existiendo como una entidad de IA. La película explora los límites de la inteligencia y la ambición humana.
- *"Chappie"* [2015]: Ambientada en un futuro cercano, la película sigue la historia de un robot policía programado con inteligencia artificial que desarrolla una personalidad propia y se ve envuelto en situaciones conflictivas.

Capítulo 2

FUNDAMENTOS DE LA

INTELIGENCIA ARTIFICIAL

Al igual que nuestro concepto de inteligencia, el término Inteligencia Artificial posee un carácter oportunista, casi un neologismo, si no fuera porque tiene aproximadamente 70 años. Es una caja en la que, con el tiempo, se han ido introduciendo conceptos, técnicas y tecnologías, muchas de ellas convertidas en herramientas, muchas abandonadas y otras que evolucionaron y sobrevivieron, dando lugar a nuevas generaciones de conceptos, técnicas y tecnologías.

Vamos a ver varios puntos de vista de la IA, con el fin de catalogarla, organizarla y permitirnos encuadrar diferentes conceptos y herramientas.

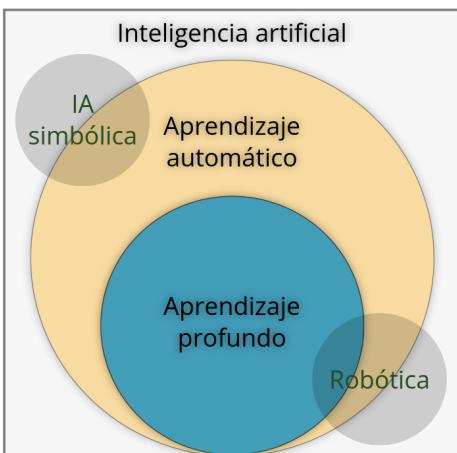


Figura 4: Clasificación básica de la Inteligencia Artificial

Empezaremos por la clasificación más extendida y básica [ver Figura 4], una en donde se diferencian las distintas orientaciones dentro de la IA en forma de conjuntos⁴².

La IA “clásica” es el conjunto de técnicas y tecnologías, dentro de la IA, que no usan aprendizaje automático o profundo, como por ejemplo Sistemas Expertos, Conjuntos Difusos, Redes Bayesianas, Algoritmos de Búsqueda y Optimización, Cadenas de Markov, Razonamiento Automático, Ontologías, Frames y un largo etc.

Creadas a lo largo de los años, están basadas en técnicas procedimentales, estadísticas y lógicas; de inspiración claramente matemática y tenían como objetivo **emular** la inteligencia humana.

Estas herramientas fueron el soporte y la fuente de ideas

⁴² Los tamaños de los círculos no son proporcionales; sólo pretenden establecer unas fronteras y la relación “contiene-a”.

de otras muchas y lo serán de otras que veremos en el futuro.

El **aprendizaje automático** [*Machine Learning*, ML] es una subdisciplina de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las máquinas aprender a partir de datos. En lugar de seguir instrucciones explícitas programadas por personas, los sistemas de ML identifican patrones y realizan predicciones o decisiones basadas en el análisis de datos.

El **aprendizaje profundo** [*Deep Learning*, DL] es una subdisciplina del aprendizaje automático que utiliza redes neuronales artificiales con muchas capas⁴³ para modelar y aprender patrones complejos de datos. A diferencia del ML tradicional, el DL es capaz de automatizar la extracción de características y manejar grandes volúmenes de datos no estructurados, como imágenes, texto y audio, entre otros.

2.1 Tipos de inteligencia artificial

Acabamos de ver una clasificación básica, que nos servirá de ayuda para encuadrar y dar nombre de diferentes soluciones de IA. Pero no es la única; existen otras que también son útiles conocer para asegurar que todos podamos hablar en el mismo lenguaje, o en los mismos términos.

2.1.1 Segundo las tareas a resolver

Si llevamos a cabo una aproximación simple y basada en los objetivos que tenemos con el fin de que la inteligencia artificial nos ayude a ejecutar y resolver tareas de nuestros procesos cotidianos, podemos distinguir tres tipos de IA:

Inteligencia artificial débil

También conocida como inteligencia artificial estrecha, se define como la inteligencia artificial que se centra típicamente en una tarea única. La inteligencia de la IA débil es limitada. Todas las IA actuales se consideran IAs débiles. Por ejemplo: asistentes virtuales, sistemas de recomendación, reconocimiento de voz, filtros de correo basura, sistemas de navegación, etc.

Una IA débil, al no tomar en consideración todo un contexto amplio, ni seguir las reglas sociales, éticas, etc., ejecuta las tareas para las que se le ha entrenado con eficacia y contundencia. No evalúa las consecuencias como lo hacemos los humanos, considerando un espectro amplio de efectos y relaciones. Por eso, es una opción incompleta, inestable y peligrosa si no se utiliza con prudencia.

Los pilotos automáticos de los vehículos autónomos son descritos como un conjunto de inteligencias artificiales débiles que trabajan juntas. Cada componente del sistema está diseñado para realizar tareas específicas de manera efectiva, sin una comprensión general o conciencia del entorno.

Ahora imagina: el subsistema de reconocimiento de señales de tráfico no reconoce una señal de

43 El término "deep" en Deep Learning proviene de la existencia de múltiples capas en la red neuronal, donde cada capa adicional permite al modelo "profundizar" más en su capacidad para detectar y aprender patrones complejos.

stop por estar oculta. El sistema completo ignoraría esta advertencia haciendo que el vehículo atravesara un cruce muy concurrido a 80 km/h.

Inteligencia artificial fuerte

También conocida como Inteligencia artificial general o AGI [su acrónimo en inglés], es la inteligencia artificial que iguala o excede la inteligencia humana promedio, es decir, la inteligencia de una máquina que puede realizar con éxito cualquier tarea intelectual de cualquier ser humano.

Este tipo de IA es la que sería capaz de analizar cualquier situación y deducir el conjunto de acciones más adecuadas para dicha situación y contexto. Lo mismo sería capaz de conducir un coche, que resolver una ecuación matemática o mantener una conversación sobre un tema concreto.

Aplicado a nuestro vehículo autónomo: aunque no fuese capaz de reconocer la señal de *stop*, una IA fuerte detectaría la presencia de un cruce, visualizaría los coches cruzando perpendicularmente a alta velocidad y, basándose en principios éticos [recuerda la Figura 1], tomaría la decisión de averiguar por qué no hay una señal de *stop*, frenando obviamente para no poner en peligro a nadie ni a nada.

Superinteligencia Artificial

Este término se utiliza para describir un nivel de inteligencia artificial que supera significativamente la capacidad intelectual de cualquier ser humano. Se trataría de una IA extraordinariamente avanzada y capaz de resolver problemas complejos, incluso aquellos que están más allá de la comprensión humana. La superinteligencia artificial es un concepto que ha generado debate y preocupación en relación con su potencial impacto en la sociedad y la humanidad en general; pero que aún no tenemos a la vista, ni la certeza de que algún día la alcancemos.

Nota: No todo cuando evoluciona lo hace para mejorar. Los modelos de lenguaje como ChatGPT, Bard y otros, al crecer en tamaño desarrollan más propiedades emergentes, pero también aumenta la posibilidad de sufrir alucinaciones⁴⁴.

2.1.2 Escuelas de pensamiento

Otra forma de clasificar la inteligencia artificial, muy relacionada con su evolución histórica. Cada uno de estos enfoques no es excluyente respecto a los demás.

Inteligencia artificial deductiva o simbólica

La IA simbólica se basa en la manipulación explícita de símbolos y reglas lógicas para representar el conocimiento y realizar razonamientos y toma de decisiones. Utiliza estructuras como árboles y grafos, entre otras, y es adecuada para dominios donde el conocimiento puede ser claramente formalizado.

⁴⁴ En el contexto de los modelos de lenguaje e inteligencia artificial, una "alucinación" se refiere a la generación de información falsa, incorrecta o sin fundamento por parte del modelo.

Inteligencia conexionista, inductiva o subsimbólica

La IA conexionista se basa en redes neuronales artificiales que simulan el funcionamiento del cerebro humano. Estas redes están compuestas por múltiples capas de neuronas artificiales y son capaces de aprender patrones complejos a partir de grandes volúmenes de datos. Este enfoque es especialmente útil para tareas que involucran datos no estructurados.

Aprendizaje automático

Se basan en algoritmos que permiten a las máquinas aprender a partir de datos sin ser explícitamente programadas. Son las mayoritarias en este libro junto con la IA subsimbólica. Como veremos tienen como ventaja la flexibilidad y aplicabilidad a una amplia gama de problemas, pero pueden requerir gran cantidad de datos etiquetados, y poseen – potencialmente – problemas de sobreajuste.

Sistemas Basados en Conocimiento y Razonamiento

Utilizan bases de conocimiento y técnicas de razonamiento para resolver problemas basados en hechos y reglas almacenadas. Pueden gestionar la incertidumbre y imprecisión, pero por otro lado poseen una gran complejidad en la creación y mantenimiento de dichas bases de conocimiento.

Optimización y Heurísticas

Están enfocados en encontrar soluciones óptimas o satisfactorias [subóptimas] a problemas complejos a través de métodos de búsqueda y heurísticas. Como ventaja tienen la capacidad para encontrar soluciones en espacios de búsqueda complejos, pero pueden ser computacionalmente intensivos y no garantizar encontrar la solución óptima, aunque como principio debe establecerse un error admisible.

Otros enfoques emergentes

Existen nuevas tendencias y enfoques que combinan diferentes paradigmas o introducen conceptos novedosos. Por ejemplo

- La Inteligencia Artificial Híbrida, que combina los modelos simbólicos y los conexionistas. Podríamos ver, en un futuro no muy lejano, el resurgir de los sistemas expertos, esta vez basados en LLM.
- La computación cuántica aplicada a la Inteligencia Artificial, que permitiría un entrenamiento de redes neuronales artificiales en muy poco tiempo y con una precisión increíble hoy en día.
- Enfoques bio-inspirados, como la redes neuronales de impulsos, las cuales son un tipo de redes neuronales artificiales más realistas que las clásicas, es decir, procesan la información de una forma más similar a las redes neuronales biológicas.

2.2 Aprendizaje

Para este punto vamos a centrarnos en el aprendizaje profundo. Veamos cómo se entrena estos modelos.

Como vimos en el capítulo anterior, para adquirir conocimientos llevamos a cabo un proceso que denominamos aprendizaje. Partimos de un conjunto de datos [o información] y el objetivo es entrenar un modelo [esto es, llevar a cabo un aprendizaje] para que pueda llevar a cabo alguna de estas funciones:

- **Predicción:** Por ejemplo, partiendo de datos sobre viviendas [localización, tamaño, número de habitaciones, precio, etc.] crear un modelo que aprenda a predecir el precio de la vivienda en una región. Otro ejemplo, a partir de un conjunto de imágenes de dígitos escritos a mano, etiquetados con el dígito que representan, crear un modelo que pueda predecir qué dígito hay en una imagen dada [o la ausencia de dígitos].
- **Detección de patrones complejos:** Por ejemplo, a partir de un conjunto de radiografías del tórax, cada uno de ellos etiquetado con la localización de un posible cáncer de pulmón, crear un modelo que detecte si hay cáncer en una radiografía dada y donde está.
- **Agrupación de datos similares:** Por ejemplo, tenemos una gran cantidad de datos sobre el comportamiento de los usuarios de una plataforma de *streaming*: qué ven, cuando, etc.; a partir de estos datos podemos crear un modelo que agrupe usuarios semejantes y caracterizarlos, a partir de ese momento podemos hacer recomendaciones a otros usuarios según el grupo en que se sitúe.
- **Reducción de dimensionalidad:** Por ejemplo, tenemos una gran cantidad de textos y queremos compararlos, saber cuales se parecen entre ellos. Para hacerlo necesitamos reducir cada texto a un vector de tal manera que vectores de textos similares se sitúen cerca, y si son muy diferentes, lejos.
- **Generación de datos sintéticos:** Por ejemplo, tengo un texto y quiero completarlo [texto a texto]. Otro ejemplo, tengo un texto y quiero sintetizar la imagen que el texto describe [texto a imagen].
- **Optimización de acciones en entornos:** Por ejemplo, tengo un robot en forma de brazo y mano con cinco dedos y quiero entrenarlo para que aprenda a manejar un tenedor y dar de comer espagueti a personas parapléjicas.

Todos estos tipos de funciones que queremos que nuestros modelos aprendan necesitan de las siguientes técnicas de aprendizaje automático, cada función en un tipo de aprendizaje.

- **Aprendizaje supervisado:** Consiste en predecir un valor o una categoría a partir de unos datos etiquetados. Esta es una de las aplicaciones más comunes del aprendizaje automático. En el caso de la predicción de valores, hablamos de problemas de regresión, donde el ob-

jetivo es predecir un valor numérico continuo basado en datos de entrada. Por ejemplo, predecir el precio de una casa. Por otro lado, la predicción de categorías se refiere a problemas de clasificación, donde el objetivo es asignar una etiqueta a cada entrada, por ejemplo predecir dígitos a partir de imágenes.

- **Aprendizaje no supervisado:** Consiste en detectar patrones o estructuras ocultas a partir de unos datos no etiquetados. Las técnicas de *clustering*, por ejemplo, agrupan datos similares basándose en características comunes. Otra caso es la reducción de dimensionalidad que permite simplificar los datos complejos mientras se mantienen aquellos más relevantes.
- **Aprendizaje por refuerzo:** Consiste en optimizar acciones mediante la interacción con el entorno. En el aprendizaje por refuerzo, un agente aprende a tomar decisiones óptimas interactuando con su entorno. Este tipo de aprendizaje es particularmente útil en aplicaciones donde las decisiones se toman secuencialmente y las consecuencias de las acciones se observan a lo largo del tiempo.

Si te preguntas que tipo de aprendizaje se asocia a la generación de datos sintéticos, este son dos subtipos del aprendizaje supervisado, en concreto autosupervisado y autoregresivo. Los veremos más adelante.

2.2.1 Proceso de entrenamiento de un modelo usando aprendizaje supervisado

El aprendizaje supervisado es el más común en ML y DL, sin menospreciar el resto de estrategias.

Partimos de un conjunto de datos etiquetados, como fotografías que están clasificadas según contienen imágenes de perros o gatos. El objetivo es crear un modelo que pueda predecir, para una imagen nueva que aún no hemos visto⁴⁵, si esta contiene un perro, un gato o ninguno de los dos.

El proceso, muy resumido, sería:

- 1] El primer paso es siempre recopilar los datos y comprobar que son válidos, útiles y completos. A estos datos etiquetados los llamaremos *conjunto de datos* [*dataset*].
- 2] Dividimos el *dataset*, obteniendo el conjunto de entrenamiento [*trainset*, típicamente 80%] y el de testeo [*testset*, típicamente el 20% restante].
- 3] Usando el *trainset* entrenamos⁴⁶ un modelo.
- 4] Usando el *testset* comprobamos que tal ha aprendido el modelo.
- 5] Despliegue y monitorización de su rendimiento: Cuando la calidad de este decaiga se vuelve al punto 1.

⁴⁵ Es bastante obvio, estamos construyendo modelos que clasifiquen imágenes que aún no hemos visto, si no fuera así no sería predicción, se reduciría a la acción de recordar.

⁴⁶ Lo normal es entrenar varios modelos y quedarte con el mejor. ¿Cómo sabemos cuál es el mejor sin usar el *testset*? Lo veremos más adelante.

De estos cinco puntos, vamos a detallar por su importancia el tercero de ellos, pero aplicado a una red neuronal *full-dense*⁴⁷.

En estos momentos no necesitas conocer en detalle como funciona una red neuronal completamente conectada [*full-dense*], sólo debes saber que contiene **una gran cantidad de números que llamamos parámetros** [entre cientos y miles de millones o incluso billones, en general es fácil acercarse al millón de parámetros en una red neuronal simple pero útil].

El problema de entrenar una red neuronal artificial se reduce a escoger los valores que deben de tener sus parámetros de tal forma que se minimice el error de la predicción. ¡Es un problema de optimización!

Buscar la combinación óptima es imposible con los ordenadores de hoy en día, pero podemos crear métodos que permiten que en un tiempo aceptable se encuentre una solución subóptima: Más adelante veremos el algoritmo de retropropagación y los optimizadores basados en el descenso del gradiente.

El proceso es iterativo sobre el *trainset* [en nuestro ejemplo formado por imágenes, cada una con su etiqueta]:

- [1] Se hace la inferencia con una imagen, obteniendo el resultado inferido.
- [2] Se calcula el error cometido usando el resultado inferido y el esperado [la etiqueta de cada fotografía], este error se usa para corregir los parámetros, cada uno en la medida en la que participó en el error, capa por capa de la última la primera.
- [3] Se sigue con la siguiente imagen y su etiqueta, volviendo al punto 1.

Todo este proceso se denomina ***epoch***⁴⁸. Este es un término similar al tiempo: hablamos que el entrenamiento duró 100 *epochs*, o que quedan 15 *epochs* para finalizar⁴⁹. Si al finalizar un *epoch* aún tenemos un error que no consideramos aceptable, volvemos a repetir el proceso, esto es, iniciamos un nuevo *epoch*. El entrenamiento finaliza obviamente cuando llegamos a un error aceptable o a un número de *epochs* máximo.

*El resultado del entrenamiento (en este caso) es una arquitectura de red neuronal y un conjunto de parámetros, posiblemente grande. A todo esto lo llamamos **modelo**.*

En el siguiente punto verás porqué.

47 Red neuronal con una o más capas internas completamente conectadas entre ellas y con las capas de entrada y salida.

48 La palabra "epoch" en el entrenamiento de redes neuronales proviene del griego "epoché", que significa "pausa" o "detención". Históricamente, se ha utilizado para referirse a un periodo significativo de tiempo. En este contexto, una "epoch" es un ciclo completo donde el conjunto de datos de entrenamiento pasa una vez por la red, representando un punto clave en su proceso de aprendizaje.

49 Los epochs no duran igual según que proyecto, esto es, no puedes pasar simplemente epochs a minutos por ejemplo. Es similar a una carrera de coches ¿cuantas vueltas llevan?

2.3 Resolución de problemas

En el mundo real nos encontramos constantemente con problemas que necesitan ser resueltos, desde determinar la mejor ruta para llegar al trabajo hasta optimizar un proceso de fabricación en una planta industrial. Aquí es donde entran en juego algunas de las herramientas de la Inteligencia Artificial “clásica”⁵⁰: técnicas de optimización y de búsqueda en el espacio de estados, entre otras.

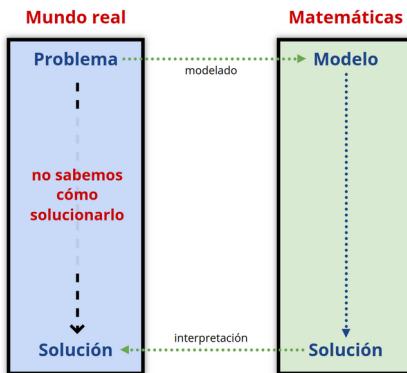


Figura 5: Esquema básico del modelado matemático

Un enfoque con un alto nivel de abstracción para solucionar problemas es usar **modelos matemáticos**. En la Figura 5 podemos ver un esquema de este proceso: Tenemos un problema en el mundo real, pero no sabemos cómo resolverlo; no somos capaces de llegar a una solución. Esta situación se da muchísimas veces en multitud de áreas y una de las formas más efectiva de solucionarlo es dando un pequeño rodeo, por el mundo de las **matemáticas aplicadas**.

La esencia de las matemáticas aplicadas es modelar problemas del mundo real, resolverlos mediante técnicas matemáticas y luego interpretar o validar esas soluciones en el contexto original.

En nuestro caso, para solucionar un problema insoluble directamente en el mundo real, lo **modelizamos** en un modelo matemático, esto es, representando el problema pero en el mundo matemático. Una vez modelizado aplicamos técnicas matemáticas conocidas para resolverlo⁵¹. Una vez resuelto, y mediante un proceso que llamaremos **interpretación**, volvemos al mundo real para encontrar que hemos hallado una solución, pero dando un rodeo.

Un ejemplo clásico: un comercial desea visitar varias ciudades y volver al punto de partida minimizando la distancia total recorrida. De acuerdo, es fácil de solucionar para pocas ciudades, pero plántate que estamos en un caso de n ciudades, n puede ser 2, cientos o más⁵². ¿cómo lo solucionar-

⁵⁰ Uso el término “clásica” porque sus resultados fueron los primeros éxitos, especialmente en juegos. Se trata realmente de la inteligencia artificial simbólica, lógica, estocástica o procedimental, como vimos antes.

⁵¹ En las matemáticas aplicadas no siempre es posible encontrar la solución óptima, a veces incluso encontrar una solución. En el primer caso se busca una solución en un tiempo razonable de tiempo, una solución subóptima.

⁵² Siempre ponte en el peor de los casos, o dicho de otra manera: busca la forma de solucionarlo “en general”.

rías? En este caso el modelado consiste en plantear el problema por medio de la **teoría de grafos** y, usando lo que sabemos sobre esta, encontrar la *mejor* solución, que tendrá la forma de secuencia de aristas formando un camino entre los nodos del grafo. El paso final es interpretar esta secuencia ordenada de aristas como una lista ordenada de carreteras que el comercial debe visitar en orden para volver a su ciudad de origen, visitando las n ciudades, de tal forma que se minimizan las distancias recorridas.

Pero **¿qué es un problema y cómo puedo representarlo como un modelo matemático?**

Una de las habilidades más importantes de la mente humana es su capacidad para resolver problemas y, si tenemos como objetivo crear máquinas inteligentes, debemos entonces hallar cómo estas máquinas pueden tener la capacidad de analizar los elementos esenciales de cada problema, identificar las acciones necesarias para resolverlo y determinar la estrategia más adecuada para abordarlo. Porque estos son los rasgos fundamentales de cualquier entidad inteligente. Es por eso que la **resolución de problemas** es un tema fundamental en el campo de la inteligencia artificial.

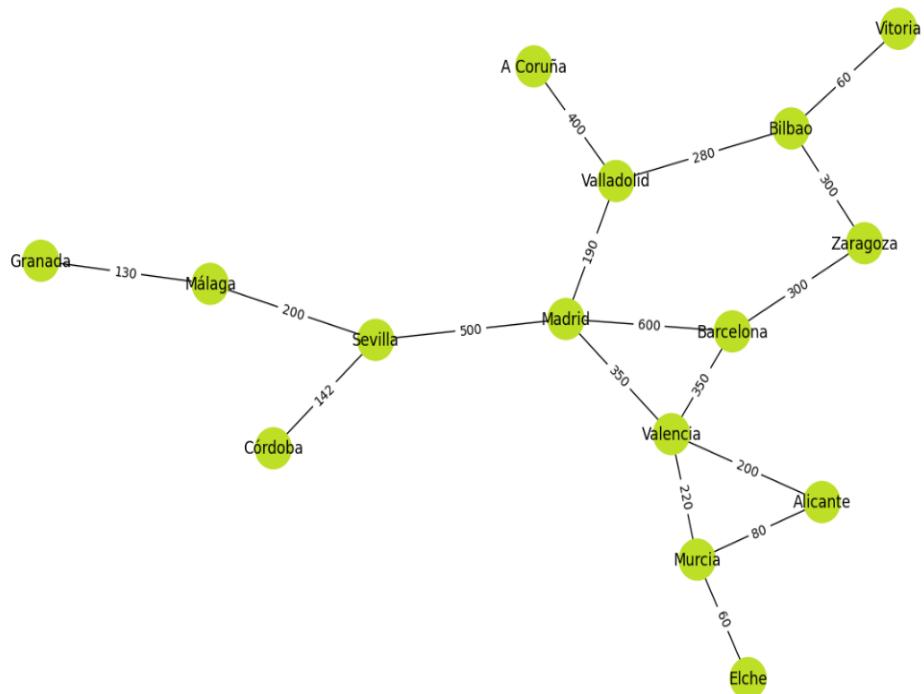


Figura 6: Grafo no dirigido, etiquetado, de 15 ciudades españolas y los kilómetros que las separan

Hay muchas aproximaciones para resolver problemas, y muchas técnicas de la matemática aplicada que podemos emplear, y en este punto vamos a ver muy brevemente dos de ellas: la **teoría de grafos y técnicas de optimización**.

2.3.1 Búsqueda de la solución en un espacio de estados

Volvamos al problema del viajante: en la Figura 6 podemos observar un grafo⁵³, en donde los nodos son algunas ciudades de España, siendo las aristas el equivalente a las carreteras que las unen. En este caso debemos observar: [1] No hay nodos inconexos, todos son accesibles por medio de una o más aristas; [2] las aristas son no dirigidas (no son flechas), luego podemos ir y volver de un nodo a otro adyacente por la misma arista; [3] hay ciclos, por ejemplo Murcia-Valencia-Alicante-Murcia de nuevo, esto implica que para ir de un nodo a otro habrá más de una ruta posible. Y definiremos una ruta entre dos nodos como una lista ordenada de aristas que los unen.

Un grafo es una estructura matemática que consiste en un conjunto de nodos y un conjunto de aristas que conectan pares de nodos. En términos formales, un grafo G se define como un par $G=(N, A)$, donde N es el conjunto de nodos y A es el conjunto de aristas. Cada arista es un par (u, v) , donde u y v son nodos en N .

Una vez que tenemos modelizado el problema en forma de grafo, buscamos el algoritmo que, dentro de la teoría de grafos, solucione nuestro problema original. En este caso se trata de un viejo conocido: TSP [*traveling salesman problem*] y es un tipo de problema del que no se conoce solución óptima⁵⁴, sólo se puede aproximar⁵⁵ una solución en un tiempo razonable. En nuestro caso la solución fue:

```
Orden de visita: ['Madrid', 'Valladolid', 'Bilbao', 'Vitoria',
'Zaragoza', 'Barcelona', 'Valencia', 'Alicante',
'Murcia', 'Elche', 'Sevilla', 'Córdoba',
'Málaga', 'Granada', 'La Coruña', 'Madrid']
```

```
Distancia total: 5634 km
```

Y el código⁵⁶ que lo ha generado:

```
1 ciclo = nx.approximation.greedy_tsp(G, weight='weight', source="Madrid")
```

En general⁵⁷, si tienes un problema y puedes modelizarlo eficazmente mediante un grafo, podrás resolver el problema; bien por que existe un algoritmo que lo soluciona o bien porque tu u otra persona elabore un nuevo algoritmo que lo haga.

Tienes un ejemplo más complejo en el repositorio: “01 Dijkstra – ciudades.ipynb”

53 Las ciudades y las aristas con sus distancias fueron generados por GPT4.

54 El problema es el tiempo. Esto significa que no se conoce un algoritmo que pueda encontrar una solución óptima en tiempo polinómico para todas las instancias del problema. Por lo tanto, a menudo se utilizan métodos aproximados o heurísticos para encontrar soluciones “buenas pero no necesariamente óptimas” en un tiempo razonable.

55 Podríamos usar un enfoque tipo “fuerza bruta”, hallaría la solución óptima, pero quizás tardando demasiado. Recordar que buscamos soluciones generales. Para pocas ciudades podemos usar el enfoque de buscar todas las combinaciones.

56 Por supuesto hay más código delante de este, en concreto aquel que crea el grafo.

57 Y digo “en general” porque si has prestado atención al principio de la historia de la IA, sabemos que hay algoritmos que no podemos garantizar que finalicen, esto es, que encuentren la solución.

Pongamos otro problema: tienes un tablero de ajedrez de sólo 9 cuadrados [3x3] **¿cómo sitúas 3 reinas en él de tal modo que no se coman entre ellas?**

Inténtalo. No tiene solución. Lo importantes aquí es **¿si lo has intentado, cómo has llegado a esa conclusión?** Posiblemente has ido colocando 1, 2 reinas en el tablero de tal forma que no pudieran atacarse, pero al colocar la tercera no has podido; después probablemente habrás intentado mover las dos primeras a nuevas posiciones, constatando que sí, no encuentras posiciones en donde ninguna amenace a ninguna otra. Si es así, no lo has hecho de forma sistemática, pero tu inteligencia ha podido con el problema.

Este es el llamado problema de las 3 reinas, y sólo tienen 130 estados diferentes.

Si en vez de usar 3 reinas [y un tablero de 3x3] usamos 4 reinas [y un tablero de 4x4], tenemos 2.517 posibilidades. **¿cómo llegas a la conclusión de que puedes o no situar las cuatro reinas en un tablero de 4x4 de forma que no se amenacen entre ellas?** Ahora no queda más remedio que ser sistemático. Hay que hacer un algoritmo, o mejor dicho, hay que modelizar el problema, aplicar un algoritmo conocido e interpretar el resultado de vuelta.

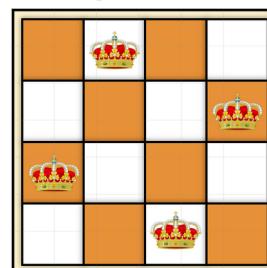
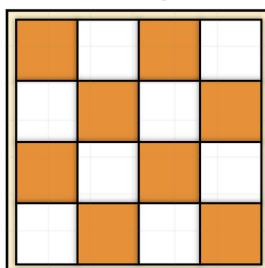
La familia de algoritmos que resuelven este problema se encuadran dentro de la **Búsqueda de soluciones en un espacio de estados**. Y son bastante intuitivos, .. y usados.

Empecemos por definir **qué es un estado** en este contexto: un tablero con 0, 1, 2, 3 o 4 reinas colocadas en él, en casillas diferentes, sin importar si se amenazan o no.

Seguidamente:

1. Debemos encontrar un **punto de partida**. ¿Cual es el estado inicial del que partimos?
2. Definimos un **objetivo a alcanzar**, a donde queremos llegar, el estado final. Es la forma de saber cuando hemos acabado.
3. Qué **acciones** tenemos a nuestra disposición para resolver el problema. Esto es, cómo pasar de un estado a otro

Figura 7: El estado inicial del problema de las N reinas, en este caso 4, (izquierda), y uno de los posibles estados finales (derecha)



En nuestro caso el estado inicial o punto de partida es un tablero 4x4 vacío; el objetivo a alcanzar es un tablero donde haya 4 reinas colocadas y no se amenacen entre ellas; y dado un estado [recuerda un tablero] podemos tomar la acción de añadir una reina en una celda si no hemos posicio-

nado ya 4 reinas en él, y así pasamos de un estado a otro.

Estas tres reglas definen, potencialmente, un grafo. En este caso uno que tienen de nodos los 2.517 estados. ¿Y las aristas? Pues entre en nodo n_x y el nodo n_y hay una arista [en este caso dirigida de n_x a n_y] si añadiendo una reina a n_x en una celda determinada, obtienes n_y .

El problema se reduce ahora a: partiendo del nodo inicial y yendo de estado en estado atravesando las aristas, buscar un nodo que consideramos un objetivo final. Si llegamos a un estado donde no podemos seguir, volvemos atrás y consideramos otra arista no usada previamente. Así hasta que no tengamos más estados que visitar o hasta que encontremos un estado final.

Este algoritmo, apenas bosquejado, es un algoritmo de búsqueda en un espacio de estados.

Lo hemos solucionado para 4 reinas, pero ¿cómo lo soluciona para N reinas? La respuesta es: igual, pero con un matiz.

El matiz que debemos tener en cuenta es el tamaño del espacio de estados ¿cuantos estados debemos recorrer para *buscar* un estado final en el peor de los casos? Vamos a analizarlo para varios valores de N y tableros de NxN:

Número de reinas	Número de estados
2	11
3	130
4	2.517
5	69.406
6	2.811.344
...	...
10	20.799.203.165.391

En general para N reinas en tableros de NxN tenemos:

$$\text{Tamaño del espacio de estados}(N) = \sum_{k=0}^N \binom{N^2}{k}$$

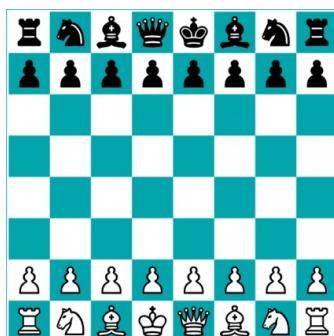
Donde cada elemento de la suma [el binomio] es la forma de colocar k reinas sobre un tablero de NxN (N^2) casillas sin tener en cuenta el orden. Fíjate, en la tabla anterior, como crece exponencialmente⁵⁸ el número de estados a explorar buscar las soluciones. Para N=10 habría que explorar casi 10^{14} estados. Si tuviésemos una máquina que va comprobando los estados uno a uno, a un ritmo de 1.000.000 por segundo, aún así tardaríamos más de tres años en recorrerlos todos y obtener todas las posibles soluciones. Para N=15 tardaríamos más de tres mil millones de años con la misma máquina.

Pero tenemos un *arma* para hacer más pequeño el número de estados a comprobar. Esta herramienta depende mucho del problema. Por ejemplo, si colocamos 2 reinas en un problema de N-reinas, si estas se atacan, cualquier combinaciones de las N-2 reinas restantes se pueden descartar y no comprobar. Esta *arma* o herramienta se la conoce como **podar el árbol de búsqueda**. Esta herramienta

⁵⁸ Siendo más precisos es un crecimiento de naturaleza combinatoria. Pero estamos de acuerdo de que crece muy rápidamente. Crece tan rápidamente que para N = 24, el número de estados es superior a 10^{40} , un 10 seguido de 39 ceros.

mienta nos va a permitir reducir, puede que incluso significativamente, el número de estados a visitar.

Muy bien, ¿y para qué sirve todo esto? ¿verías algo de inteligencia si una máquina se pusiera a resolver problemas de 4-reinas? Probablemente no. ¿Y para cualquier N-reinas? Ahora que intuyes como lo calcularía, probablemente valores más una máquina que responda rápidamente si un N-reinas, para un N concreto, puede que grande, tiene al menos una solución [y la muestre]. En este caso, dada la explosión combinatoria de estados a comprobar, la **inteligencia** consistiría en explorar sólo las aristas que lleven a estados más probables para encontrar un estado final.



Vamos a aplicar lo aprendido al ajedrez.

Aquí las cosas se complican, pero seguimos usando los mismos conceptos y definiciones. Ahora el problema es: suponiendo que es mi turno ¿que jugada debo hacer para mejorar a la larga mi situación y empobrecer la de mi contrincante?

O dicho de otra manera: ¿qué jugada debo hacer para minimizar la ventaja de mi oponente y maximizar la mía?

Minimizar-maximizar → **Minimax**

Para empezar, tenemos la restricción de que los movimientos son por turnos, por tanto tengo que tener en cuenta no sólo mis jugadas, si no además las de mi contrincante.

También hay que tener en cuenta que la explosión combinatoria del problema de las N-reinas es insignificante en comparación con la explosión exponencial de las posibles jugadas en el ajedrez. Por tanto no podemos hacer una búsqueda exhaustiva en el espacio de estados.



Video 1: Algoritmo Minimax en 4 minutos

Fuente: BitBoss

Veamos como funciona el algoritmo *minimax*.

Primero necesitamos algo nuevo: la **función de coste**, una función con la capacidad de medir, dada una configuración [estado] del tablero, quien tiene ventaja. Esto de por sí ya implicaría inteligencia, de hecho sería un buen proyecto de IA que requeriría muchos datos [cientos millones de configuraciones históricas, quien ganó y en cuantos movimientos] para entrenar un modelo que reconociera quien va ganando, igual que lo hace un/una maestra del ajedrez.

En nuestro caso vamos a considerar una función de coste más humilde y fácil de calcular, una en donde a partir de un estado nos entregue un valor positivo si tenemos ventaja y negativo si la tienen nuestro oponente. ¿Cómo? Mediante la suma ponderada de nuestras figuras menos la suma ponderada de las figuras del otro jugador. Para ponderar las figuras le damos un valor numérico a cada una de ellas, máximo al rey [100], mínimo a un peón [1], de tal modo que si nosotros tenemos sólo el rey y un peón sumaremos 101, y si nuestro oponente tiene su rey y tres peones sumaría 103, la diferencia sería -2 indicando que no tenemos ventaja. El resto de las figuras tendría su valor de ponderación entre el máximo y el mínimo [por ejemplo⁵⁹, reina 90, alfil 50, torre 40, caballo 30].

Veamos como funciona el algoritmo, suponiendo que es nuestro turno:

1. Inicialmente, se parte de un estado que representa la configuración actual del tablero. Este es nuestro **estado inicial**.
2. Desde ese estado, se generan todos los movimientos legales para el jugador actual [llamémosle *max*, somos nosotros].
3. A continuación, para cada uno de estos nuevos estados generados, se generan todos los movimientos legales para el oponente [llamémosle *min*].
4. Este proceso se repite **hasta llegar a una cierta profundidad** o hasta que se alcanzan posiciones terminales [como jaque mate o tablas].
5. Una vez alcanzada la profundidad deseada, se evalúa la configuración del tablero en cada estado terminal utilizando la **función de coste**. Como sabemos, esta función asigna un valor numérico a cada estado que representa lo favorable que es para nosotros [*max*].
6. Estos valores se propagan hacia atrás, desde las hojas hasta el estado inicial. En cada nivel, el jugador *max* [nosotros] escogeremos el movimiento que resulta en el valor máximo, mientras que para el jugador *min* escogeremos el movimiento que resulta en el valor mínimo.
7. Al llegar de nuevo al estado inicial, el movimiento elegido será aquel que tenga el valor máximo resultante.

Para mejorar la eficiencia, el algoritmo **minimax** a menudo se usa en combinación con técnicas como la poda alfa-beta, que elimina *ramas* del grafo/árbol de búsqueda que no necesitan ser exploradas porque no afectarán la decisión final.

En repositorio tienes un *notebook* para jugar al ajedrez: [01 ¿Jugamos al ajedrez?.ipynb](#)

2.3.2 Resolución de problemas mediante procesos de optimización

Aunque muchos de los problemas se pueden modelizar como grafos, y por tanto caer en las soluciones basadas en la exploración de estos, la eficacia de la solución depende del problema específico y de lo que quiera buscar. En general si el problema requiere de alguna forma de red o conexio-

⁵⁹ Modificando estos valores, podemos ir modificando la forma (el estilo) en el que nuestra IA (usando minimax) se comporta. Y sí, le he dado poco valor a los caballos.

nes entre entidades puede modelarse como grafos, pero si necesita optimizar (minimizar o maximizar) uno o más parámetros, como su nombre indica, son candidatos a resolverse mediante **técnicas de optimización**⁶⁰.

Las técnicas de optimización son métodos matemáticos y algorítmicos diseñados para encontrar la mejor solución posible a un problema dado, generalmente sujeto a un conjunto de restricciones. Estas técnicas buscan maximizar o minimizar una función objetivo, como el coste, el tiempo o la eficiencia, explorando sistemáticamente diferentes soluciones posibles para determinar cuál es la más óptima.

Pongamos un ejemplo: Una empresa fabrica dos productos, A y B. Cada producto pasa por dos procesos: ensamblaje y embalaje. La empresa desea maximizar sus beneficios, pero está limitada por el tiempo disponible en cada proceso. La información relevante es la siguiente:

- Beneficio por unidad de A: 3€
- Beneficio por unidad de B: 4€
- Tiempo de ensamblaje por unidad del producto A: 2 horas
- Tiempo de ensamblaje por unidad del producto B: 1 hora
- Tiempo de embalaje por unidad de A: 1 hora
- Tiempo de embalaje por unidad de B: 2 horas
- Tiempo total disponible para ensamblaje: 100 horas
- Tiempo total disponible para embalaje: 80 horas

Como lo que quiere la empresa es maximizar beneficios, matemáticamente quiere maximizar la siguiente función: $Z = 3x + 4y$, siendo x el número de unidades de A producidas e y el número de unidades de B producidas.

Además, deducimos a partir de la información que nos dan que tenemos estas restricciones:

- $2x + y \leq 100$ [tiempo máximo de ensamblaje]
- $x + 2y \leq 80$ [tiempo máximo de embalaje]
- Y las restricciones obvias de que x e y son mayores o iguales a cero.

Observa que de nuevo hemos llevado el problema del mundo real al mundo de las matemáticas aplicadas, donde hay que buscar dos valores (x, y) de tal manera que optimicen una función objetivo (Z) teniendo en cuenta unas restricciones. Este problema se soluciones usando un método llamado *simplex* que no vamos a ver aquí⁶¹.

Pero estamos estudiando IA y no son estos tipos de problemas los que queremos resolver, quere-

60 Ambas técnicas no son excluyentes.

61 Pero si tienes curiosidad, la solución óptima es $x=40$ e $y=20$, lo que da un beneficio de 200€.

mos ir a por aquellos que **las matemáticas no pueden resolver**.

Existen problemas de optimización que son demasiado complejos para ser resueltos mediante métodos matemáticos exactos debido a la complejidad de cómputo que exigen, la naturaleza no lineal o la dimensión del problema. Estos problemas suelen ser abordados utilizando algoritmos que pueden encontrar soluciones aproximadas [subóptimas] en un tiempo razonable.

Una solución subóptima se refiere a una solución que, aunque no es la mejor posible (es decir, no maximiza o minimiza completamente la función objetivo), cumple con las restricciones del problema y se considera aceptable o suficientemente buena en términos de los criterios establecidos.

Aceptar soluciones subóptimas es común en problemas complejos donde encontrar la solución óptima es computacionalmente inviable debido a la dimensión del problema, la naturaleza no lineal de las funciones involucradas o el tiempo limitado disponible para el cálculo.

No vamos a entrar en detalles, las técnicas de optimización se dan en niveles universitarios y son, a la par, útiles y complejas. Pero merece la pena nombrar algunos algoritmos usados para buscar soluciones subóptimas.

Algoritmos genéticos.

Los algoritmos genéticos son una familia de algoritmos de optimización inspirados en la teoría de la evolución natural. Aquí cada solución potencial [en el ejemplo anterior cada par x,y que cumple las restricciones] se modela como un individuo de una población.

Para buscar una solución cercana a la óptima se llevan a cabo los siguientes pasos:

1. Se genera una población inicial de individuos aleatoriamente.
2. Se evalúa la aptitud de cada individuo de la población usando la función objetivo.
3. Se seleccionan los mejores individuos para ser padres, basándose en su aptitud.
4. Se combinan los genes de los padres para crear nuevos individuos (hijos).
5. Se aplican pequeñas alteraciones a los hijos.
6. Los individuos de la nueva generación reemplazan a la generación anterior.
7. El algoritmo termina cuando se cumple un criterio de parada [número máximo de generaciones, convergencia, etc.].



Figura 8: Algoritmos Genéticos en 5 minutos

Fuente: BitBoss

Enjambre de partículas

La técnica de Enjambre de Partículas es un método de optimización heurística basado en el comportamiento colectivo de enjambres, como bancos de peces o bandadas de aves. Cada *partícula* representa una solución potencial en el espacio de búsqueda. Las partículas se mueven en este espacio según una serie de ecuaciones matemáticas que consideran tanto su posición actual como las posiciones de sus vecinas.

En cada iteración del algoritmo:

1. Cada partícula evalúa su aptitud utilizando la función objetivo.
2. Se compara la aptitud de la partícula con su mejor posición conocida. Si la nueva posición tiene una aptitud mejor, se actualiza la mejor posición individual de la partícula.
3. Se compara la mejor posición individual de la partícula con la mejor posición global del enjambre. Si es mejor, se actualiza la mejor posición global.
4. Cada partícula ajusta su velocidad y posición en función de sus propias experiencias y las del enjambre, generalmente usando un modelo que considera la inercia, la memoria cognitiva y la memoria social.

El algoritmo termina cuando se alcanzan criterios de parada predefinidos, como un número máximo de iteraciones o una tolerancia de error [= estar cerca de la solución].

El descenso del gradiente

Cada uno de estos algoritmos tiene su correspondiente campo de aplicación o contexto en donde su aplicabilidad es más conveniente. En el punto 2.2.1 afirmé que el proceso de aprendizaje de una red neuronal se reducía a un proceso de optimización. En efecto si una red neuronal posee – pongamos – diez millones de parámetros, lo que buscamos al entrenarla es buscar la combinación de valores de esos parámetros que minimizan un error⁶². El algoritmo del descenso del gradiente es el algoritmo que usamos para entrenar redes neuronales.

62 El error es la diferencia entre lo que calcula y lo que debería calcular.

Este algoritmo, igual que los anteriores, es de tipo iterativo, tiene una parte matemática *dura*⁶³, pero fácilmente explicable y entendible. Sólo tienes que poseer la experiencia de caminar por la montaña.

Imagínate que estás en una cordillera, con los ojos tapados, y que sólo sientes la pendiente⁶⁴ debajo de tus pies. Tu objetivo es buscar el punto más profundo de la cordillera. ¿qué haces?

Obviamente irás dando paso a paso, buscando siempre bajar la mayor pendiente, y si en algún momento esta se hace positiva, volver atrás.

Esta imagen mental es válida para recordar como funciona un *descenso del gradiente* básico. Pero probablemente que no has visto como pasar del mundo real al modelo del mundo matemático:

Seguimos utilizando una función objetivo, en este caso denominada **función de coste**. Esta función evalúa cada solución posible (como x e y en el primer ejemplo, o los 10 millones de parámetros en el último), proporcionando el error asociado a cada una. Este error puede visualizarse como la altura en un paisaje montañoso, y nuestro objetivo es encontrar la solución que minimice la función de coste, es decir, el punto más bajo del paisaje, donde el error es cero o lo más cercano a cero posible.



Vídeo 2: *¿Qué es el Descenso del Gradiente? Algoritmo de Inteligencia Artificial*

Fuente: DotCSV

En la **edición 2024** de este libro tienes más información y ejemplos de estos algoritmos.

2.4 Agentes inteligentes

Hasta ahora hemos visto los fundamentos de la IA y dos familias de tecnologías muy usadas en la actualidad⁶⁵ para crear sistemas inteligentes. Pero estas herramientas, vistas de forma individual, no aportan lo que venimos definiendo como *inteligencia*. En general, todas las tecnologías que estamos denominando inteligentes en este libro, no lo son tanto. Como mucho sólo aportan la apariencia de inteligencia humana. Para crear un sistema inteligente de verdad, necesitamos unir varias

⁶³ Todo aprendizaje de un conocimiento es *relativamente* duro, dependiendo del conocimiento y actitud previa, la motivación y la documentación que se posea, pero especialmente de la guía de otra persona. Si se dan todas estas circunstancias en positivo, nada es duro.

⁶⁴ Simplemente sabes si estás en llano ($=0$), cuesta abajo (<0) o cuesta arriba (>0); tanto hacia adelante/atrás, como a los lados.

⁶⁵ Las usas a menudo sin percatarte de ello: navegadores GPS, sistemas de recomendación, planificación de vuelos, y un largo etc.

de estas tecnologías en un concepto: **los agentes inteligentes**.

Informalmente, un agente inteligente es un sistema diseñado para percibir su entorno, procesar información, tomar decisiones y actuar de manera autónoma para alcanzar objetivos específicos. A diferencia de los programas tradicionales, que siguen instrucciones predefinidas, un agente inteligente es capaz – idealmente – de adaptarse y aprender a través de la interacción con su entorno.

Este concepto es muy importante y es el colofón de este capítulo, dado que en los agentes inteligentes reúnen en una solución todos los fundamentos que hemos visto y las tecnologías inteligentes que veremos en el resto del libro.

Formalmente:

Un agente inteligente es una entidad autónoma capaz de percibir su entorno a través de sensores y de actuar sobre él mediante actuadores, con el objetivo de maximizar algún tipo de medida de rendimiento o utilidad. Utiliza algoritmos y técnicas de inteligencia artificial para procesar la información percibida, tomar decisiones y aprender de las experiencias pasadas, adaptándose a cambios en su entorno para mejorar su comportamiento y alcanzar sus objetivos de manera más eficaz.

Compara esta definición con la definición de inteligencia en 1.1.

Un agente, *a secas*, puede ser visto como cualquier cosa que perciba su **entorno** y actué sobre él. Un agente humano observa su entorno y lo modifica. Un agente robótico igualmente observa su entorno, el que sea, con sus sensores y opera sobre él mediante actuadores. Un agente de software recibe información de su entorno y envía información al mismo, quizá modificándolo.

Piensa en un robot de limpieza: el entorno es fácil de identificar, el suelo⁶⁶; además posee sensores que captan del entorno lo que necesitan y actúan sobre él moviéndose y aspirando la suciedad. Ahora piensa en un agente que juegue al ajedrez ¿cuál es su entorno? ¿qué detecta del entorno? ¿cómo actúa sobre el entorno?.

A los agentes que acabamos de nombrar necesitan algo más que percibir y obrar sobre el entorno, *algo* tiene que tomar las decisiones en virtud de los sensores para llevar a cabo las acciones sobre el entorno. El **programa del agente**. Y es aquí en donde podemos hablar de inteligencia.

Un **agente inteligente** está compuesto, en su forma más simple, por los siguientes componentes (Figura 9):

⁶⁶ Cuando suban a las camas y entren en los armarios, su entorno será diferente.

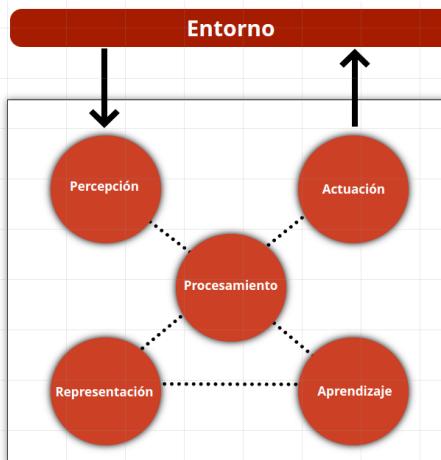
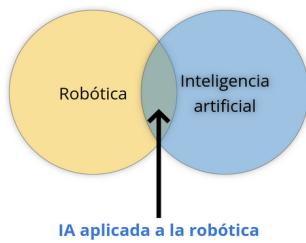


Figura 9: Componentes de un agente inteligente

- **Percepción:** El agente recibe información del entorno a través de sensores o dispositivos de entrada. Estos pueden ser bases de datos, Internet, cámaras, micrófonos, sensores de temperatura, entre otros muchos.
- **Actuación:** Una vez que el agente ha tomado una decisión [ver más abajo], este debe actuar en el entorno para alcanzar sus objetivos. La acción se lleva a cabo a través de efectores o dispositivos de salida, como conexiones a Internet, motores, pantallas, altavoces, etc. Estos efectores permiten que el agente interactúe con el entorno y ejecute las acciones seleccionadas.

Estos dos elementos son los únicos que se relacionan con el entorno, y lo pueden hacer mediante el uso de uno o más canales [sistema operativo, visión , oído, tacto, sonido, etc.].

- **Procesamiento de información:** Una vez que el agente recopila información del entorno, la procesa y la analiza para obtener conocimiento útil que pueda ser usado para la toma de decisiones, basándose en la información tanto nueva como previa. La toma de decisiones podría implicar poner en funcionamiento los actuadores necesarios, si fuese el caso.
- **Representación del conocimiento:** Se refiere a cómo el agente codifica, almacena y organiza el conocimiento relevante sobre el entorno en el que opera. La representación es crucial para que el agente pueda comprender y tomar decisiones en base a la información que recibe.
- **Aprendizaje:** Otro aspecto fundamental de un agente inteligente es su capacidad para aprender y adaptarse a medida que interactúa con el entorno. Puede utilizar técnicas de aprendizaje automático, como el aprendizaje supervisado o el aprendizaje por refuerzo, para mejorar su rendimiento con el tiempo y ajustar su comportamiento según las experiencias pasadas.



Posiblemente hayas pensado en un *androide* al leer estos cinco componentes de un agente inteligente.

Aunque no es la única aplicación de un agente de estas características, un androide humanoide es la aplicabilidad máxima, el soporte en donde un agente inteligente despliega todas y cada una de sus tecnologías inteligentes. El lugar donde robótica e inteligencia artificial intersectan.

Las tecnologías inteligentes que mostramos en el resto del libro se encuadran dentro de alguno de los cinco componentes de un agente inteligente que acabamos de ver, o en varios de ellos.

RETOS DEL CAPÍTULO 2

Para cada una de las películas que he listado en los restos del capítulo 1 y las que tu habrás encontrado, haz una ficha intentando clasificar, a partir de su sinopsis, las IA que aparecen en ellas según los tipos o clasificaciones de la IA que hemos visto.

Pregúntale a la IA:

- ¿Qué es un algoritmo?
- ¿Qué es una lista? ¿Qué es un conjunto? ¿Cómo puedo representarlos en papel?
- ¿Qué es un grafo y como puedo representarlo en papel?
- ¿Puedo usar una matriz para representar un grafo?
- ¿Qué es la complejidad de un algoritmo?

El problema del pastor y su manada

Un pastor se encuentra en la orilla de un río junto con un lobo, un cordero y un paquete de heno. El pastor quiere pasar a todos al otro lado del río, pero sólo hay una barca, en donde únicamente caben dos de ellos: el pastor y otro. Además tiene el siguiente problema: si deja solos al lobo y al cordero, el primero se comerá al segundo; por otro lado, si deja sólo al cordero y al heno igualmente el primero se comerá el segundo. ¿cómo puede hacer para pasar a todos al otro lado del río y que nadie se coma a nadie?

Responde a las siguientes preguntas:

- ¿Cuál es el punto de partida?
- ¿Cuál es el objetivo a alcanzar?
- ¿Cómo podríamos representar estos estados inicial y final, y al mismo tiempo los estados intermedios?
- Enumera todos los estados indicando si es un estado prohibido o no.
- ¿Podemos representar estos estados y sus transiciones como una matriz? hazlo.
- ¿Podemos representar estos estados y sus transiciones como un grafo? hazlo.
- Busca una solución general para pasar del estado inicial al final.

Las torres de Hanoi

El problema de las Torres de Hanoi es un famoso problema matemático y de lógica. Se plantea como un rompecabezas que involucra mover un conjunto de discos de diferentes tamaños desde un palo de origen a un palo de destino, utilizando otro auxiliar.

Las reglas del problema son las siguientes: Hay tres palos verticales, etiquetadas como A, B y C. Inicialmente, todos los discos están apilados en orden descendente de tamaño, de abajo a arriba, en

la varilla A, con el disco más grande en la parte inferior y el disco más pequeño en la parte superior. El objetivo es mover todos los discos del palo A al palo C, manteniendo la misma configuración de apilamiento. Solo se puede mover un disco a la vez. Un disco solo puede colocarse sobre otro disco que sea más grande o en una varilla vacía.

Supongamos que sólo tenemos 3 discos:

- ¿Cuál es el punto de partida?
- ¿Cuál es el objetivo a alcanzar?
- ¿Cómo podríamos representar estos estados inicial y final, y al mismo tiempo los estados intermedios?
- Enumera todos los estados indicando si es un estado prohibido o no.
- ¿Podemos representar estos estados y sus transiciones como una matriz? hazlo.
- ¿Podemos representar estos estados y sus transiciones como un grafo? hazlo.
- Busca una solución general (n discos) para pasar del estado inicial al final.
- ¿Existe una fórmula para saber cuantos movimientos hacen falta como mínimo para n discos? si es así ¿cuántos pasos necesitaríamos como mínimo para mover 64 discos?

Otros problemas

Aquí tienes otros retos para ejercitarte para resolver problemas de forma sistemática:

- El problema de Sudoku: Consiste en llenar una cuadrícula de 9x9 con dígitos del 1 al 9, de manera que cada columna, cada fila y cada subcuadrícula de 3x3 contenga todos los dígitos sin repetirse.
- El problema del laberinto: Consiste en encontrar un camino desde un punto de inicio hasta un punto objetivo en un laberinto.
- El problema del camino más corto: Consiste en encontrar el camino más corto entre dos puntos en un grafo ponderado, donde los nodos representan ubicaciones y las aristas tienen un costo asociado.
- ¿Cómo funciona el sistema de navegación de un GPS para un automóvil?

Habla con tu agente conversacional favorito y soluciona las siguientes cuestiones:

- ¿Cuáles son los componentes clave de un agente inteligente?
- ¿Qué papel juega el entorno en la toma de decisiones de un agente inteligente?
- Pregúntale a tu IA favorita: Establece un diálogo con una IA, comparando un agente inteligente y una persona. Hazlo siempre con la perspectiva de los elementos que forman un agente inteligente. Tu objetivo es llegar a la respuesta a esta pregunta ¿una persona es o no un agente inteligente?
- ¿Cuál es la diferencia entre la percepción y la acción en un agente?
- ¿Cómo se representa el conocimiento en un agente inteligente?

- ¿Qué significa que un agente sea autónomo?
- Pregúntale a tu IA favorita: ¿Qué es la estimación de la pose dentro de la visión artificial?
- Pregúntale a tu IA favorita: ¿Qué es LiDAR y para qué se usa dentro de la IA?
- Pregúntale a tu IA favorita: Recopila tipo de sensores que pueden estar relacionados con la percepción y busca ejemplos comerciales de los mismos. Usando tu IA favorita, describe su funcionamiento técnico y sus utilidades en la IA.
- Pregúntale a tu IA favorita: Qué es la lógica preposicional y su relación con la lógica de predicados. ¿Hay lógicas de orden mayor? Pide ejemplos.
- ¿Cómo se representa el audio? Busca fuentes en Internet que muestren las diferentes formas de representar en sonido. Céntrate en representaciones de audio de voz.
- Recopila fotos antiguas, preferiblemente en color, y que tengan mala calidad o estén defectuosas. Eligelas de forma que no represente un problema publicarlas en Internet ni perder su gestión o incluso propiedad. Busca sitios en la web que mejoren la calidad de las imágenes.

Modeliza un coche autónomo como agente inteligente.

¿Qué es “*Neural Radiance Fields*” [NERF] y que relación tiene con la visión artificial?

Busca vídeos sobre SLAM que muestre los resultados de un mapeo con localización. ¿Las aspiradoras robotizadas podrían usar esta técnica?

Busca vídeos de robots de Boston Dynamics. Analiza si su comportamiento es inteligente o no. Cuando llegues a una conclusión, busca información sobre si esta empresa está usando IA para sus desarrollos.

Capítulo 3

INTELIGENCIA ARTIFICIAL EN EL MUNDO REAL



Ilustración 3: Las IAs no van a sacarte el trabajo; te sacará el trabajo aquella persona que use las IAs.
Fuente: Generado con Midjourney; propiedad del autor.

Hemos preguntado a **nuestra IA favorita** sobre los ámbitos en los que esta se podría aplicar, detallando brevemente cada uno de los puntos en los que su influencia es o podría ser determinante. Y esta ha sido su respuesta.

3.1 Automatización

Gestión de inventario y logística: La IA se utiliza para predecir las necesidades de inventario y optimizar la gestión de flotas. Mediante el análisis de datos históricos⁶⁷ y en tiempo real, los algoritmos de IA pueden predecir la demanda futura de productos, lo que permite a las empresas realizar pedidos de inventario de manera más precisa y eficiente. Además, la IA puede ayudar en la planificación de rutas y en la optimización de la distribución de productos, reduciendo los costos y mejorando la eficiencia de la logística.

Programación de citas y reservas: Mediante el uso del lenguaje natural, la IA puede automatizar el proceso de programación de citas y reservas. Los sistemas de IA pueden entender las solicitudes de los clientes y responder de manera inteligente, encontrando los horarios disponibles, reservando citas y enviando confirmaciones a los usuarios. Esto agiliza el proceso y mejora la experiencia del cliente al eliminar la necesidad de interacción humana en cada paso.

Procesamiento de texto y lenguaje natural para automatizar la iteración H-M: La IA se utiliza para automatizar la comunicación entre humanos y máquinas a través del procesamiento de texto y lenguaje natural. Los sistemas de IA pueden comprender y responder preguntas o solicitudes de los usuarios de manera inteligente, brindando información, realizando acciones o conectando a los usuarios con recursos relevantes. Esto permite una interacción más fluida y eficiente, eliminando la necesidad de intervención humana en ciertos escenarios.

Soporte al cliente y asistencia virtual: La IA se utiliza para proporcionar soporte al cliente y asistencia virtual de manera automatizada. Los sistemas de IA pueden responder preguntas comunes, brindar información sobre productos o servicios, solucionar problemas técnicos básicos y realizar tareas de seguimiento. También pueden realizar un enrutamiento inteligente de consultas complejas a agentes humanos cuando sea necesario. Esto ayuda a mejorar la eficiencia del servicio al cliente al proporcionar respuestas rápidas y precisas, reduciendo la carga de trabajo para los agentes humanos.

Automatización de procesos de negocio: La IA se utiliza para automatizar procesos empresariales, lo que implica la sustitución de tareas repetitivas y manuales por algoritmos y sistemas de IA. Esto puede incluir la automatización de la recopilación y clasificación de datos, el procesamiento de facturas, la generación de informes, la gestión de tareas y flujos de trabajo, entre otros. La automatización de estos procesos ayuda a aumentar la eficiencia, reducir errores y liberar tiempo para que los empleados se centren en tareas más estratégicas y creativas.

3.2 Aprendizaje automático

El aprendizaje automático, también conocido como *machine learning*, es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos y modelos estadísticos que permiten a las

⁶⁷ Por medio del uso de estrategias de ciencias de datos.

computadoras aprender y mejorar automáticamente a través de la experiencia. El aprendizaje automático tiene una amplia gama de aplicaciones en diferentes campos, y a continuación, describiré detalladamente su uso en los siguientes campos:

Sistemas de recomendación de productos y servicios: El aprendizaje automático se utiliza ampliamente en sistemas de recomendación, como los que se encuentran en plataformas de comercio electrónico, servicios de *streaming* de música y video, y motores de recomendación en general. Estos sistemas analizan el historial de comportamiento del usuario, como el historial de compras, las preferencias o las interacciones en línea, y utilizan algoritmos de aprendizaje automático para hacer recomendaciones personalizadas. Esto mejora la experiencia del usuario y ayuda a los negocios a aumentar las ventas al mostrar productos o servicios relevantes para cada usuario individual.

Motores de búsqueda y análisis de datos: Los motores de búsqueda se basan en algoritmos de aprendizaje automático para indexar y clasificar páginas web de acuerdo con su relevancia para las consultas de búsqueda. Los algoritmos de aprendizaje automático pueden analizar patrones y señales en los datos para determinar la relevancia de una página web en función de los términos de búsqueda. Además, el aprendizaje automático se utiliza en el análisis de datos para identificar tendencias, patrones ocultos y relaciones complejas en grandes conjuntos de datos. Esto permite a las empresas tomar decisiones informadas y obtener información valiosa a partir de los datos.

Asistentes virtuales y chatbots: Los asistentes virtuales y chatbots utilizan el aprendizaje automático para entender y responder a las consultas y solicitudes de los usuarios de manera más inteligente. Estos sistemas pueden utilizar algoritmos de procesamiento del lenguaje natural [NLP, por sus siglas en inglés] y modelos de aprendizaje automático para comprender el lenguaje humano, reconocer intenciones y proporcionar respuestas relevantes. A medida que interactúan más con los usuarios, los asistentes virtuales y chatbots pueden aprender y mejorar su capacidad de respuesta, brindando una experiencia más personalizada y eficiente.

Reconocimiento de voz y traducción de idiomas: El aprendizaje automático se utiliza en aplicaciones de reconocimiento de voz, como los asistentes de voz en teléfonos inteligentes y los sistemas de control por voz en los automóviles. Estos sistemas utilizan modelos de aprendizaje automático para transformar las señales de voz en texto o comandos comprensibles para las computadoras. Además, el aprendizaje automático se aplica en la traducción automática de idiomas, donde los modelos aprenden a traducir texto o voz de un idioma a otro. Estas aplicaciones han mejorado significativamente la precisión y la eficiencia de la comunicación entre humanos y máquinas, así como la comunicación entre personas que hablan diferentes idiomas.

Clasificación de imágenes y reconocimiento facial: El aprendizaje automático se utiliza en la clasificación y etiquetado automático de imágenes, así como en el reconocimiento facial. Los modelos de aprendizaje automático pueden aprender a reconocer patrones visuales en imágenes y asignar etiquetas descriptivas a cada imagen. Esto es especialmente útil en aplicaciones como la clasificación de imágenes en plataformas de redes sociales, la detección automática de contenido inapropiado o la organización de grandes colecciones de imágenes.

Además, el reconocimiento facial es otra aplicación importante del aprendizaje automático. Los al-

gorítmos de aprendizaje automático pueden aprender a identificar y reconocer rostros en imágenes y videos, lo que se utiliza en aplicaciones de seguridad, autenticación biométrica y etiquetado automático de fotos en redes sociales. Estos sistemas pueden analizar características faciales únicas y compararlas con una base de datos para realizar tareas como desbloquear dispositivos, verificar la identidad de una persona o proporcionar etiquetas automáticas de nombres en fotografías.

3.3 Análisis de datos

Detección de fraudes financieros y riesgos de crédito: La IA se utiliza para analizar grandes volúmenes de datos financieros y transacciones en tiempo real, identificando patrones sospechosos y comportamientos fraudulentos. Los algoritmos de aprendizaje automático pueden detectar anomalías en los datos, prevenir el fraude en tiempo real y mejorar la precisión de las decisiones de crédito al evaluar el riesgo de los solicitantes.

Predicción del clima y pronóstico de desastres naturales: La IA se utiliza en el análisis de grandes conjuntos de datos climáticos históricos, datos satelitales y modelos climáticos para predecir el clima a corto y largo plazo. Los algoritmos de IA pueden identificar patrones climáticos, analizar datos en tiempo real y generar pronósticos precisos sobre el clima y los desastres naturales, lo que ayuda a tomar decisiones informadas y tomar medidas preventivas.

Análisis de sentimiento y opiniones en redes sociales y encuestas: La IA se utiliza para extraer y analizar información de redes sociales, comentarios de usuarios y encuestas en línea. Los algoritmos de procesamiento del lenguaje natural [NLP] permiten comprender y clasificar el sentimiento y las opiniones expresadas en los textos. Esto es útil para las empresas y organizaciones que desean conocer la opinión de los clientes, realizar estudios de mercado y evaluar la satisfacción del cliente.

Identificación de patrones de compra y comportamiento del consumidor: La IA se utiliza para analizar grandes volúmenes de datos transaccionales, datos de compras en línea y perfiles de usuarios con el objetivo de identificar patrones de compra y comportamientos del consumidor. Estos análisis ayudan a las empresas a comprender las preferencias de los clientes, personalizar las ofertas y campañas de marketing, y tomar decisiones informadas sobre la gestión de inventario y el desarrollo de nuevos productos.

Seguridad y vigilancia en tiempo real: La IA se utiliza para monitorear y analizar imágenes y videos en tiempo real, lo que contribuye a la seguridad y vigilancia en diversas aplicaciones. Los sistemas de videovigilancia basados en IA pueden detectar comportamientos anómalos, reconocer rostros y objetos, y enviar alertas en caso de actividades sospechosas. Esto es útil en entornos como aeropuertos, calles urbanas, instituciones financieras y otros lugares donde se requiere una vigilancia constante.

3.4 Robótica

Robots de fabricación y producción en masa: Los robots de fabricación y producción en masa equipados con IA han revolucionado las industrias de manufactura y producción. Estos robots son capaces de realizar tareas repetitivas y precisas de manera más eficiente que los seres humanos. La IA permite a estos robots adaptarse y aprender de manera autónoma, mejorando la eficiencia y la calidad de la producción. Los sistemas de visión artificial basados en IA permiten a los robots reconocer objetos, clasificarlos y manipularlos con precisión, lo que ha aumentado la velocidad y precisión de los procesos de fabricación.

Robots de exploración y misiones espaciales: La exploración espacial es un campo exigente y peligroso para los seres humanos. La IA se utiliza en robots de exploración espacial para realizar tareas de investigación y recopilación de datos en entornos extraterrestres. Estos robots pueden analizar su entorno, navegar y tomar decisiones autónomas basadas en la información recopilada. La IA permite a los robots adaptarse a situaciones imprevistas y aprender de sus experiencias. Además, los algoritmos de IA ayudan a los robots a procesar grandes cantidades de datos y extraer información relevante para los científicos.

Drones de vigilancia y entrega de paquetes: Los drones equipados con IA se han convertido en herramientas versátiles para la vigilancia y la entrega de paquetes. La IA permite a los drones identificar y rastrear objetos en tiempo real, evitando obstáculos y ajustando su ruta de vuelo de manera autónoma. Los drones de vigilancia utilizan algoritmos de IA para analizar imágenes y videos en tiempo real, identificando patrones y detectando anomalías. En el caso de la entrega de paquetes, la IA ayuda a optimizar las rutas de vuelo y mejorar la eficiencia en la entrega.

Automóviles autónomos y sistemas de transporte inteligentes: La IA ha sido fundamental en el desarrollo de automóviles autónomos y sistemas de transporte inteligentes. Los vehículos autónomos utilizan una combinación de sensores, sistemas de navegación y algoritmos de IA para reconocer el entorno, tomar decisiones de conducción y evitar obstáculos. Los algoritmos de aprendizaje automático permiten a los automóviles autónomos mejorar su desempeño a medida que adquieren más experiencia en diferentes condiciones de conducción. Además, la IA se utiliza en sistemas de gestión del tráfico para optimizar la fluidez del tráfico, reducir la congestión y mejorar la seguridad en las carreteras.

Robots para la atención médica y la asistencia en el hogar: Los robots equipados con IA están siendo utilizados en entornos médicos y en el hogar para proporcionar atención y asistencia a las personas. En la atención médica, los robots pueden realizar tareas como la administración de medicamentos, el monitoreo de signos vitales y la asistencia en procedimientos quirúrgicos. La IA permite a estos robots interpretar datos médicos, analizar síntomas y diagnosticar enfermedades de manera precisa. En el hogar, los robots asistenciales pueden ayudar a las personas mayores o discapacitadas en tareas diarias como la limpieza, la preparación de alimentos y el recordatorio de medicamentos. La IA les permite adaptarse a las necesidades y preferencias individuales, aprendiendo de las interacciones y proporcionando una asistencia personalizada.

3.5 Asistencia médica

Diagnóstico de enfermedades y detección temprana: La IA ha demostrado una gran precisión en el diagnóstico de enfermedades. Utilizando algoritmos de aprendizaje automático, la IA puede analizar grandes cantidades de datos médicos, como imágenes de resonancias magnéticas, tomografías computarizadas y resultados de pruebas de laboratorio, para identificar patrones y signos tempranos de enfermedades. Esto puede ayudar a los médicos a realizar diagnósticos más precisos y rápidos, lo que a su vez permite un tratamiento temprano y más efectivo.

Identificación de tratamientos personalizados y terapias genéticas: La IA puede analizar datos médicos y genéticos de pacientes individuales para identificar tratamientos personalizados. Al considerar factores como el historial médico, los genes y las características únicas de cada paciente, la IA puede ayudar a los médicos a seleccionar el tratamiento más adecuado y diseñar terapias genéticas específicas para enfermedades hereditarias.

Investigación en biomedicina y neurociencia: La IA ha facilitado avances significativos en la investigación biomédica y neurociencia. Puede analizar grandes conjuntos de datos genómicos, proteómicos y moleculares para descubrir nuevos objetivos terapéuticos, identificar biomarcadores de enfermedades y acelerar el proceso de desarrollo de medicamentos. Además, la IA puede simular modelos computacionales complejos para comprender mejor los mecanismos subyacentes de enfermedades y mejorar la comprensión científica.

Monitorización de pacientes y seguimiento de su estado de salud: La IA se puede utilizar para monitorizar a los pacientes de forma continua y realizar un seguimiento de su estado de salud. Los dispositivos médicos conectados, como monitores de glucosa, tensiómetros y dispositivos de seguimiento de actividad física, pueden recopilar datos en tiempo real que son analizados por algoritmos de IA. Esto permite la detección temprana de cambios significativos en el estado de salud de los pacientes y la generación de alertas para los médicos o cuidadores.

Administración de medicamentos y gestión de datos médicos: La IA puede ayudar en la administración de medicamentos al predecir la respuesta de un paciente a ciertos medicamentos, optimizando las dosis y reduciendo los efectos secundarios. Además, la gestión de datos médicos se ha beneficiado enormemente de la IA, ya que puede analizar y organizar grandes volúmenes de datos clínicos y de investigación, facilitando el acceso y la búsqueda de información relevante para los profesionales de la salud.

3.6 Transporte

Sistemas de navegación y planificación de rutas: La IA se utiliza en sistemas de navegación y aplicaciones de mapas para proporcionar direcciones precisas y actualizadas en tiempo real. Estos sistemas utilizan algoritmos de IA para analizar datos en tiempo real, como el tráfico, las condiciones del clima y los accidentes, y así calcular la ruta más eficiente y rápida para llegar a un destino.

Sistemas de control del tráfico aéreo y terrestre: La IA se utiliza en sistemas de control de tráfico aéreo para monitorizar y gestionar el flujo de aviones de manera segura y eficiente. Los algoritmos de IA analizan múltiples factores, como el tráfico aéreo, las condiciones meteorológicas y las rutas preferidas, para tomar decisiones informadas y optimizar el uso del espacio aéreo. De manera similar, la IA se utiliza en sistemas de control de tráfico terrestre para optimizar la circulación de vehículos y reducir la congestión.

Sistemas de transporte público inteligentes: La IA se utiliza en sistemas de transporte público para mejorar la eficiencia y la calidad del servicio. Por ejemplo, los algoritmos de IA se utilizan para predecir la demanda de pasajeros y optimizar los horarios y las rutas de los autobuses y trenes. Además, los sistemas de IA pueden analizar datos en tiempo real, como el flujo de pasajeros y las condiciones del tráfico, para ajustar los servicios en tiempo real y responder a situaciones imprevistas.

Sistemas de seguridad y detección de riesgos en carreteras y vías férreas: La IA se utiliza en sistemas de seguridad para detectar y prevenir riesgos en carreteras y vías férreas. Los sistemas de visión por computadora basados en IA pueden analizar imágenes y videos de cámaras de tráfico para detectar objetos, identificar señales de tráfico y advertir sobre posibles peligros. Además, la IA se utiliza en sistemas de detección de intrusiones en las vías férreas para evitar accidentes y proteger la integridad de la infraestructura ferroviaria.

Vehículos autónomos y aviones no tripulados: La IA desempeña un papel fundamental en el desarrollo de vehículos autónomos y aviones no tripulados (drones). Los vehículos autónomos utilizan sistemas de IA para interpretar datos de sensores, como cámaras y radares, y tomar decisiones en tiempo real para navegar de manera segura por el entorno. Los drones también utilizan algoritmos de IA para planificar rutas, evitar obstáculos y realizar tareas específicas, como la entrega de paquetes o la inspección de infraestructuras.

3.7 Entretenimiento

Generación de música y sonido: La IA se ha utilizado para componer música original y generar sonidos personalizados. Los algoritmos de IA pueden analizar patrones musicales existentes y crear nuevas composiciones en diferentes géneros y estilos. Esto ha llevado a la producción de música completamente generada por IA que se puede utilizar en películas, videojuegos o incluso en la industria de la música en general.

Efectos visuales y gráficos en películas y videojuegos: La IA ha mejorado la calidad y el realismo de los efectos visuales en películas y videojuegos. Los algoritmos de aprendizaje automático pueden analizar imágenes y videos para detectar objetos, rastrear movimientos y crear efectos especiales impresionantes. Además, la IA también se utiliza para mejorar la renderización gráfica en tiempo real, lo que permite una experiencia visual más inmersiva en los videojuegos.

Personalización de contenido y recomendaciones en plataformas de streaming: Las plataformas de streaming utilizan algoritmos de IA para personalizar el contenido recomendado para cada usuario. Estos algoritmos analizan el historial de visualización, las preferencias y los patrones de consumo

de cada usuario para ofrecer recomendaciones precisas y relevantes. Esto ayuda a los usuarios a descubrir contenido nuevo y atractivo, lo que mejora su experiencia de entretenimiento.

Juegos de azar y apuestas en línea: La IA también se ha utilizado en el campo de los juegos de azar y las apuestas en línea. Los algoritmos de IA pueden analizar grandes cantidades de datos históricos y patrones de juego para predecir resultados y mejorar las estrategias de juego. Esto se utiliza tanto en juegos de casino en línea como en apuestas deportivas, donde la IA puede proporcionar información valiosa para tomar decisiones informadas.

Realidad virtual y aumentada para experiencias inmersivas: La IA juega un papel crucial en el desarrollo de experiencias inmersivas en realidad virtual [RV] y realidad aumentada [RA]. Los algoritmos de IA se utilizan para reconocer y rastrear objetos en tiempo real, lo que permite una interacción más fluida y realista con el entorno virtual o aumentada. Además, la IA también se utiliza para generar contenido virtual y personalizado en función de las preferencias y acciones del usuario.

3.8 Finanzas

Análisis de riesgo y fraude financiero: La IA puede ayudar en el análisis de riesgo financiero mediante el procesamiento de grandes volúmenes de datos financieros y la identificación de patrones y tendencias que podrían indicar un mayor riesgo. Al utilizar técnicas de aprendizaje automático y algoritmos avanzados, la IA puede evaluar el riesgo crediticio, evaluar la solvencia de los prestatarios y detectar posibles fraudes financieros, como transacciones fraudulentas, manipulación de datos o actividades sospechosas. Esto permite a las instituciones financieras tomar decisiones más informadas y mitigar los riesgos asociados con sus operaciones.

Predicción de tendencias y fluctuaciones del mercado: La IA puede analizar grandes cantidades de datos históricos y en tiempo real para identificar patrones y tendencias en los mercados financieros. Utilizando algoritmos de aprendizaje automático, puede realizar pronósticos sobre el comportamiento futuro del mercado, la fluctuación de los precios de las acciones, las tasas de interés y otros indicadores financieros clave. Esto ayuda a los inversores y analistas financieros a tomar decisiones de inversión más fundamentadas y a gestionar su cartera de manera más eficiente.

Personalización de ofertas de productos y servicios financieros: La IA puede utilizar datos demográficos, históricos y de comportamiento para personalizar las ofertas de productos y servicios financieros a los clientes. Al analizar los patrones de gasto, los hábitos de inversión y las preferencias del cliente, la IA puede recomendar productos financieros específicos, como préstamos, tarjetas de crédito o seguros, que se ajusten a las necesidades individuales de cada cliente. Esto mejora la experiencia del cliente y aumenta las posibilidades de satisfacción y fidelización.

Análisis de cartera y recomendaciones de inversión: La IA puede analizar las carteras de inversión existentes y proporcionar recomendaciones sobre cómo optimizarlas. Al evaluar factores como el perfil de riesgo, los objetivos de inversión y las condiciones del mercado, la IA puede sugerir cambios en la asignación de activos, estrategias de diversificación y selección de inversiones. Esto ayuda a los inversores a tomar decisiones más informadas y a maximizar el rendimiento de sus inver-

siones.

Detección de anomalías en transacciones y monitorización de cuentas: La IA puede analizar grandes volúmenes de datos de transacciones financieras y realizar un seguimiento de las cuentas para detectar cualquier actividad anómala o sospechosa. Puede identificar patrones de comportamiento inusuales, transacciones atípicas o intentos de fraude y generar alertas para su revisión por parte de expertos en seguridad financiera. Esto ayuda a prevenir el fraude financiero y a garantizar la seguridad de las transacciones y las cuentas de los clientes.

3.9 Recursos humanos

Selección y reclutamiento de candidatos: La IA se utiliza para analizar grandes volúmenes de datos y perfiles de candidatos con el fin de identificar a los mejores candidatos para un puesto de trabajo específico. Los algoritmos de IA pueden analizar *currículums vitae*, perfiles en redes sociales y otros datos relevantes para identificar habilidades, experiencia y adecuación cultural. Además, la IA también puede ayudar en la preselección inicial de candidatos a través de chatbots y entrevistas virtuales.

Gestión de la capacitación y desarrollo del personal: La IA puede personalizar los programas de capacitación y desarrollo para los empleados. Mediante el análisis de datos de rendimiento y evaluaciones de habilidades, la IA puede identificar brechas de conocimiento y recomendar cursos o actividades de desarrollo específicos para cada empleado. Esto ayuda a optimizar el proceso de aprendizaje y desarrollo, permitiendo a los empleados adquirir habilidades relevantes para sus roles y objetivos profesionales.

Monitorización de la satisfacción y el rendimiento de los empleados: La IA puede analizar datos en tiempo real sobre la satisfacción y el rendimiento de los empleados. Esto se logra mediante el análisis de encuestas de satisfacción, retroalimentación de los empleados y otros datos relacionados. La IA puede identificar patrones y tendencias en los datos para comprender mejor el nivel de satisfacción y el rendimiento de los empleados, lo que permite a los gerentes tomar medidas proactivas para mejorar la satisfacción y el compromiso de los empleados.

Programación de turnos y horarios de trabajo: La IA puede ayudar en la programación de turnos y horarios de trabajo de manera eficiente. Teniendo en cuenta los datos históricos de asistencia, la demanda de trabajo y las políticas de la empresa, los algoritmos de IA pueden generar horarios óptimos que maximicen la eficiencia y minimicen los conflictos de programación. Esto ayuda a reducir la carga administrativa y mejora la satisfacción de los empleados al tener horarios más equilibrados.

Automatización de tareas administrativas: La IA puede automatizar diversas tareas administrativas en el departamento de Recursos Humanos, como el procesamiento de nóminas, la gestión de permisos y vacaciones, y el mantenimiento de registros de empleados. Los chatbots basados en IA también pueden responder preguntas frecuentes de los empleados sobre políticas y procedimientos, liberando tiempo para que los profesionales de Recursos Humanos se enfoquen en actividades

más estratégicas.

3.10 Educación

El uso de la IA en el campo de la educación ha experimentado un crecimiento significativo en los últimos años. La IA se utiliza en una variedad de formas para mejorar y optimizar los procesos educativos, brindando beneficios tanto a los estudiantes como a los educadores. Algunas aplicaciones destacadas de la IA en la educación incluyen sistemas de aprendizaje adaptativo y personalizado, asistencia en la evaluación y calificación de pruebas, tutorías virtuales y asistencia en línea, asesoramiento en la elección de cursos y carreras, y análisis de datos educativos y evaluación del rendimiento del sistema educativo.

Los **sistemas de aprendizaje adaptativo y personalizado** utilizan algoritmos de IA para adaptar la experiencia de aprendizaje de cada estudiante de manera individualizada. Estos sistemas recopilan datos sobre el desempeño y el progreso del estudiante, y utilizan esta información para ofrecer contenido educativo personalizado, adaptado a las necesidades y habilidades específicas de cada estudiante. Esto permite un aprendizaje más eficiente y efectivo, ya que los estudiantes pueden avanzar a su propio ritmo y centrarse en las áreas en las que necesitan más apoyo.

La IA también puede desempeñar un papel importante en la **evaluación y calificación de pruebas**. Los sistemas de IA pueden analizar y evaluar respuestas de manera automática, lo que ahorra tiempo y esfuerzo a los educadores. Estos sistemas pueden identificar patrones en las respuestas de los estudiantes, proporcionar retroalimentación inmediata y generar informes detallados sobre el desempeño de los estudiantes. Esto permite una evaluación más objetiva y precisa, y brinda a los educadores información valiosa para adaptar su enseñanza.

Las **tutorías virtuales y la asistencia en línea** son otra área en la que la IA está transformando la educación. Los sistemas de IA pueden proporcionar tutoría personalizada y apoyo educativo a los estudiantes a través de plataformas en línea. Estos sistemas pueden interactuar con los estudiantes, responder preguntas, ofrecer explicaciones y guiarlos a través de los conceptos difíciles. Esto es especialmente beneficioso para aquellos estudiantes que no tienen acceso a tutores o maestros en persona, o que desean un apoyo adicional fuera del horario escolar.

La IA también se utiliza en el **asesoramiento en la elección de cursos y carreras**. Los sistemas de IA pueden analizar las preferencias, habilidades e intereses de los estudiantes, así como los datos sobre las tendencias laborales y las demandas del mercado, para brindar recomendaciones personalizadas sobre los cursos y las carreras que podrían ser más adecuados para cada estudiante. Esto ayuda a los estudiantes a tomar decisiones informadas y a planificar su futuro educativo y profesional de manera más efectiva.

Por último, la IA se utiliza para el **análisis de datos educativos y la evaluación del rendimiento del sistema educativo en general**. Los sistemas de IA pueden recopilar y analizar grandes cantidades de datos relacionados con el rendimiento de los estudiantes, la eficacia de los métodos de enseñanza, las tendencias educativas y otros factores relevantes. Esto proporciona información valiosa para

identificar áreas de mejora, desarrollar políticas educativas basadas en evidencia y realizar ajustes en el sistema educativo en general.

3.11 Medios de comunicación

Generación de contenido periodístico y reportajes: La IA puede ser utilizada para generar automáticamente noticias y reportajes. Los algoritmos de generación de texto basados en IA pueden analizar grandes cantidades de datos, como informes de noticias y datos estadísticos, para redactar artículos informativos y coherentes. Esto puede ayudar a los periodistas y editores a obtener rápidamente información relevante y elaborar informes más eficientes.

Análisis y monitorización de la opinión pública y la atención de los medios de comunicación: La IA puede ser empleada para analizar y monitorear la opinión pública en diferentes plataformas de redes sociales y medios de comunicación. Los algoritmos de procesamiento del lenguaje natural pueden identificar tendencias, temas relevantes y sentimientos expresados por el público en línea. Esto proporciona a los medios de comunicación una visión más profunda de las preferencias y preocupaciones de la audiencia, lo que puede influir en la estrategia editorial y en la toma de decisiones.

Personalización de noticias y contenido publicitario: La IA permite la personalización del contenido presentado a los usuarios. Los algoritmos de recomendación pueden analizar el comportamiento y las preferencias de los usuarios para ofrecer noticias y anuncios adaptados a sus intereses individuales. Esto puede mejorar la experiencia del usuario y aumentar la participación al mostrar contenido relevante y atractivo.

Optimización de la publicidad en línea y promoción de productos: La IA se utiliza para optimizar la publicidad en línea y las estrategias de promoción de productos. Los algoritmos de aprendizaje automático pueden analizar los datos demográficos, el comportamiento del usuario y otros factores para identificar audiencias objetivas y optimizar las campañas publicitarias. Esto puede ayudar a los medios de comunicación a generar más ingresos publicitarios y a los anunciantes a mejorar la eficacia de sus esfuerzos promocionales.

Detección de contenidos inapropiados y moderación de los comentarios de los usuarios: La IA puede ser utilizada para detectar y moderar contenido inapropiado en los medios de comunicación. Los algoritmos de IA pueden examinar imágenes, texto y videos en busca de contenido ofensivo, *spam* o violaciones de las políticas. Esto ayuda a los medios de comunicación a mantener un entorno seguro y respetuoso para los usuarios, al tiempo que reduce la carga de trabajo de los moderadores humanos.

3.12 Agricultura

Monitorización de la calidad del suelo y los cultivos: La IA se utiliza para analizar datos recopilados

de sensores y dispositivos de campo, como drones equipados con cámaras y sistemas de teledetección, para evaluar la calidad del suelo y el estado de los cultivos. Los algoritmos de IA pueden detectar deficiencias nutricionales, identificar enfermedades o plagas, y predecir el rendimiento de los cultivos.

Planificación de siembras y cosechas: Los modelos de IA pueden analizar grandes volúmenes de datos históricos, como información climática, registros de siembras anteriores y características del suelo, para predecir los mejores momentos para sembrar y cosechar. Estas predicciones ayudan a optimizar los rendimientos y reducir los riesgos asociados con las condiciones climáticas y los ciclos de cultivo.

Control y monitorización de plagas y enfermedades en los cultivos: La IA se utiliza para identificar y monitorear plagas y enfermedades en los cultivos. Mediante la combinación de imágenes capturadas por drones o cámaras de campo con algoritmos de visión por computadora, la IA puede detectar signos tempranos de infestación y enfermedades, permitiendo una respuesta rápida y precisa para su control.

Detección de anomalías en la producción y en la calidad del producto: La IA puede analizar datos en tiempo real, como mediciones de temperatura, humedad y otros parámetros relevantes, para identificar anomalías en la producción agrícola. Esto puede ayudar a prevenir pérdidas y optimizar la calidad del producto final.

Optimización del uso de recursos como el agua y los fertilizantes: La IA se utiliza para desarrollar modelos de gestión agrícola que optimizan el uso de recursos escasos, como el agua y los fertilizantes. Utilizando algoritmos de aprendizaje automático, la IA puede analizar datos históricos y en tiempo real para determinar la cantidad precisa de recursos que deben aplicarse en cada zona de cultivo, maximizando así la eficiencia y reduciendo los costos y el impacto ambiental.

3.13 Energía y medio ambiente

Ánálisis de la calidad del aire y la contaminación: La IA se utiliza para recopilar y analizar grandes cantidades de datos relacionados con la calidad del aire, como las emisiones de contaminantes y los niveles de partículas en suspensión. Los algoritmos de IA pueden identificar patrones y tendencias en los datos, ayudando a los investigadores y a los responsables de la toma de decisiones a comprender mejor los factores que afectan la calidad del aire y a desarrollar estrategias para mejorarla.

Optimización del uso de recursos energéticos: La IA se utiliza para optimizar el uso de recursos energéticos, como el petróleo y la energía solar. Por ejemplo, los algoritmos de IA pueden analizar datos sobre la producción y el consumo de petróleo y sugerir medidas para maximizar su eficiencia. Asimismo, la IA puede ayudar a determinar la ubicación óptima de las instalaciones de energía solar y a predecir la producción de energía a partir de fuentes renovables.

Monitorización y análisis de la producción de energía y de los procesos de distribución: La IA se

utiliza para monitorizar y analizar los sistemas de producción de energía, como las plantas de generación eléctrica y las redes de distribución. Los algoritmos de IA pueden detectar anomalías y fallos en tiempo real, lo que permite una respuesta rápida y eficiente para minimizar interrupciones en el suministro de energía. Además, la IA puede optimizar la gestión de la red eléctrica y predecir la demanda de energía para mejorar la planificación y reducir los costos.

Análisis de las tendencias del clima y el impacto del cambio climático: La IA se utiliza para analizar grandes cantidades de datos climáticos y modelar las tendencias del clima a largo plazo. Los algoritmos de IA pueden identificar patrones complejos en los datos y ayudar a los científicos a comprender mejor los factores que contribuyen al cambio climático. Además, la IA puede predecir los posibles escenarios futuros del clima y evaluar el impacto de diferentes estrategias de mitigación.

Optimización del consumo de energía y del uso de recursos naturales: La IA se utiliza para optimizar el consumo de energía en edificios, industrias y otros sectores. Mediante el análisis de datos en tiempo real, los algoritmos de IA pueden identificar oportunidades de ahorro de energía, como el ajuste automático de la iluminación y la climatización en función de la ocupación de los espacios. Además, la IA puede ayudar a optimizar el uso de recursos naturales, como el agua y los materiales, minimizando el desperdicio y fomentando la economía circular.

3.14 Derecho

Análisis de datos y precedentes jurídicos: La IA puede analizar grandes volúmenes de datos legales y precedentes judiciales de manera rápida y eficiente. Especialmente mediante técnicas de procesamiento de lenguaje natural puede identificar patrones, tendencias y relaciones en la información legal. Esto permite a los abogados y juristas obtener perspectivas valiosas para la toma de decisiones fundamentadas y estratégicas en sus casos.

Investigación de antecedentes y análisis de riesgos legales: La IA puede ser utilizada para realizar investigaciones exhaustivas sobre antecedentes legales de personas, empresas o situaciones específicas. Al analizar bases de datos legales, registros públicos y otra información relevante, puede identificar riesgos potenciales, antecedentes penales, litigios pasados y otros factores que pueden afectar la viabilidad de un caso legal. Esto permite a los abogados evaluar mejor los riesgos asociados a sus clientes o a una transacción legal en particular.

Automatización de tareas legales y administrativas: La IA puede automatizar una variedad de tareas legales y administrativas que consumen tiempo y recursos. Por ejemplo, puede ayudar en la revisión de documentos legales estándar, como contratos o acuerdos, identificando cláusulas relevantes, detectando errores y asegurando la coherencia y precisión en su redacción. Además, la IA puede ayudar en la gestión de casos, seguimiento de plazos y recordatorios, organización de documentos y otra documentación legal.

Asistencia en la redacción de contratos y documentos legales: La IA puede ser utilizada como una herramienta colaborativa en la redacción de contratos y otros documentos legales. Al analizar la redacción existente y los precedentes, puede sugerir opciones de redacción, cláusulas y términos

que se ajusten a los intereses y objetivos de las partes involucradas. Esto puede mejorar la eficiencia en el proceso de redacción y ayudar a evitar errores y ambigüedades legales.

Análisis de patentes y propiedad intelectual: La IA puede ser utilizada para analizar grandes volúmenes de información relacionada con patentes y propiedad intelectual. Puede examinar bases de datos de patentes, revisar solicitudes de patentes, realizar búsquedas exhaustivas de literatura técnica y realizar análisis de infracción de derechos de propiedad intelectual. Esto puede ayudar a los abogados especializados en propiedad intelectual a identificar posibles conflictos, evaluar la patentabilidad de una invención y apoyar la toma de decisiones relacionadas con la estrategia de propiedad intelectual.

RETOS DEL CAPÍTULO 3

Pregunta a tu IA favorita

1. Busca más ámbitos que el tema no haya contemplado y pregunta a tu IA favorita sobre las oportunidades de usar la IA en dicho ámbito.
2. Escoge uno de los puntos de un ámbito concreto y mediante tu IA favorita profundiza en sus usos, técnicas y tecnologías. Comprueba todo lo que te diga.
3. Escoge un ámbito de los tratados en el tema y mediante el uso de la IA y tu buscador favorito, encuentra las empresas que realicen servicios en dicho ámbito. Prioriza empresas cercanas a ti o de tu mismo país o región.
4. Todavía no hemos hablado de riesgos, así que escoge un ámbito de los tratados en el tema y mediante un diálogo con tu IA favorita, recopila los riesgos potenciales de su uso intenso o abusivo.
5. En grupo: Tanto si trabajas o estudias, reflexiona sobre los puntos en donde tu día a día podría automatizarse. Podrías usar tu agente conversacional favorito para que te guíe sobre las tecnologías inteligentes más adecuadas para hacerlo.
6. Vuelve loco/a a tu IA favorita: deja volar la imaginación e invéntate un ámbito [por ejemplo, la colonización de Marte] e interroga a tu IA sobre los usos que ésta puede aportar en dicho ámbito. Razona con ella sobre la utilidad de su propuesta.

Sección III: Visión Artificial

"La visión artificial es la capacidad de una máquina para percibir y comprender imágenes y videos del mundo que la rodea."

Fei-Fei Li (1976 -)

No podemos entender la inteligencia artificial sin mostrar qué es y como funciona la visión artificial. Comenzando por un tema básico, ¿qué es una imagen? Los espacios de color, formatos de archivo, análisis simple de imágenes, transformaciones y operaciones entre imágenes. Seguidamente entraremos en el mundo de la percepción a bajo nivel por medio de la clasificación, identificación y segmentación de patrones visuales.

Seguidamente entraremos en el mundo de la percepción de alto nivel, por medio de la estimación de la pose humana, el *tracking* o seguimiento de objetos y la estimación monocular de la imagen; este capítulo, como el anterior, tiene un enfoque bastante práctico.

Finalmente hablaremos sobre su impacto en la sociedad, mostrando aplicaciones y peligros de esta área de aplicación de las tecnologías inteligentes.

Capítulo 4

ELEMENTOS Y

PROCESAMIENTO BÁSICO DE

UNA IMAGEN

4.1 Qué son los píxeles y cómo representan la unidad básica de una imagen digital.

Los píxeles son la unidad fundamental que compone una imagen digital. El término "píxel" es una abreviatura de "elemento de imagen" en inglés [*picture element*] el cual, cada uno de ellos, representa un punto en una cuadrícula bidimensional conteniendo información sobre el color y la intensidad luminosa, o incluso la opacidad, en esa posición específica de la imagen.

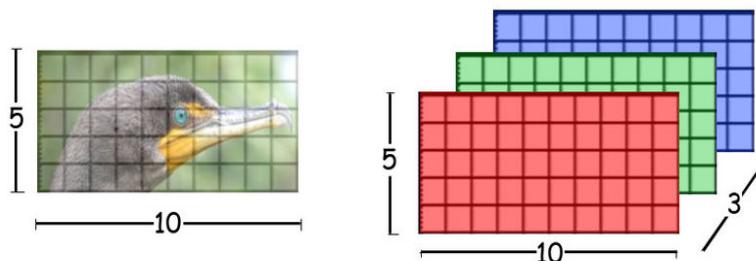


Figura 10: Alto, ancho de una imagen y los planos que forman el modelo RGB.

Fuente: Al-Azze, Jamil et al. (2020). "Creating Color Image Signature Based On Laplacian Equation". JOIV. 3. 10.30630/joiv.3.2.223

En una imagen digital, la información visual se descompone en una matriz de píxeles dispuestos en filas y columnas. Cada píxel individual tiene una ubicación única definida por sus coordenadas [x, y] en la imagen.

Si recuerdas la Figura 5, puedes ver que aquí también se cumple su postulado: tenemos un problema en el mundo real, trabajar con imágenes, pero no sabemos como. La solución consiste en pasar

el problema al mundo matemático, esto es, **codificamos las imágenes como matrices**. Ahora podemos operar con ellas de forma muy eficiente.

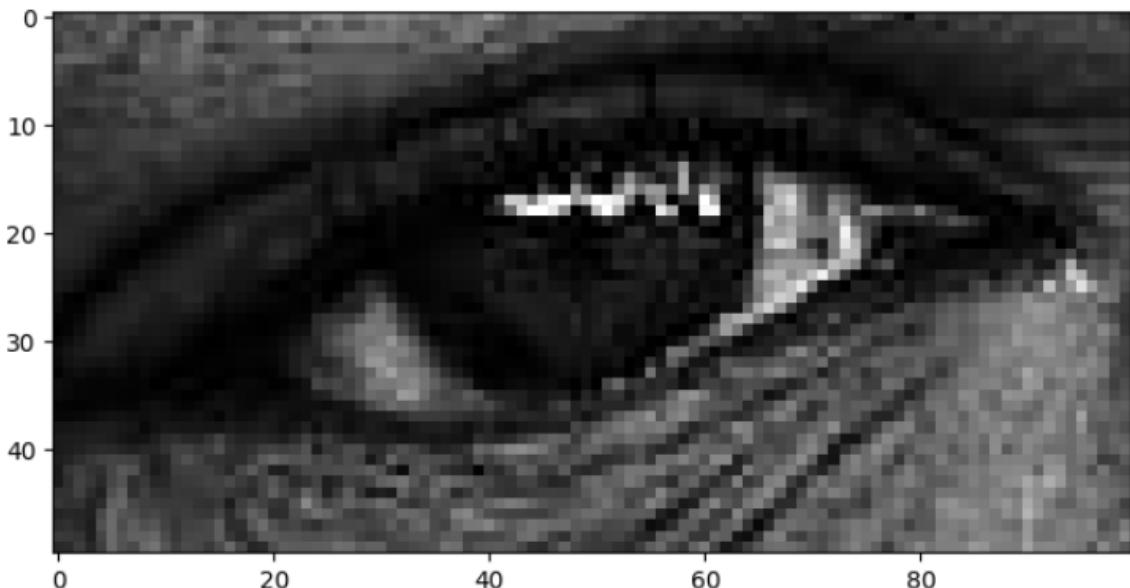
Vamos a generar código python que nos muestre todo lo que acabamos de decir:

04 Operaciones básicas con imágenes.ipynb

Prompt:

Genera el python el código para leer una imagen desde una url, mostrarla por completo con un tamaño limitado y mostrar un "zoom" de una región específica en donde se puedan ver los pixels.

El código generado por el *prompt* anterior genera, a partir de la imagen suministrada como una *url*, la siguiente ampliación:



En donde se pueden ver los pixels que forman una porción de la imagen en tono de grises [1 sólo canal]. Cada pequeño cuadrado de la imagen ampliada es un pixel de 8 bits, que codifica un número de 0 a 255 y lo hace corresponder con un nivel de gris: 0 negro, 255 blanco.

Un píxel almacena información sobre el color utilizando diferentes modelos de representación, siendo el **modelo RGB** [red, green, blue] el más común en visión artificial. En el modelo RGB, cada píxel se representa mediante tres componentes: rojo, verde y azul. Cada componente tiene un valor numérico que indica la intensidad de ese color específico en el píxel. Estos valores pueden variar entre 0 y 255 en un formato de 8 bits, lo que permite una amplia gama de colores⁶⁸.

Además del modelo RGB, existen otros modelos de color, como CMYK [cyan, magenta, yellow, key],

⁶⁸ Si cada componente es de 8 bits, y tenemos 3 componentes (R, G y B) entonces tenemos 24 bits, lo que permite codificar 16.777.216 colores diferentes.

utilizado principalmente en impresión, y HSL/HSV [*hue, saturation, lightness/value*], que representan colores en términos de tono, saturación y luminosidad o brillo. En breve los veremos con un poco más de detalle.

La **resolución de una imagen** está relacionada con la cantidad de píxeles presentes en la imagen. Se mide en píxeles por pulgada⁶⁹ [PPI] o en píxeles totales [ancho x alto]. Cuanto mayor sea la resolución, más detalles se podrán percibir en la imagen.

Cada píxel tiene una ubicación espacial y está relacionado con el tamaño físico de la imagen cuando se imprime o se muestra en una pantalla. Si la resolución es baja y hay pocos píxeles por pulgada, es probable que la imagen se vea **pixelada** o borrosa. Por otro lado, una alta resolución proporciona una mayor claridad y detalles más nítidos.

Es importante tener en cuenta que los píxeles también pueden almacenar información adicional, como por ejemplo **canales alfa** para representar transparencias o máscaras que indican la presencia o ausencia de ciertos objetos o características en la imagen.

4.2 La resolución de una imagen y su importancia

La resolución de una imagen se refiere a la cantidad de píxeles que componen esta y se expresa generalmente como el número de píxeles en el ancho y alto, como por ejemplo, 1920 x 1080 píxeles.

La resolución es un factor fundamental que afecta la claridad y el nivel de detalle de una imagen. Una mayor resolución significa que la imagen está compuesta por más píxeles, lo que resulta en una imagen más nítida y definida. Por otro lado, una menor resolución implica menos píxeles y, por lo tanto, una imagen menos detallada y más borrosa.

La resolución de una imagen es especialmente relevante cuando se trata de visualización en diferentes dispositivos, como monitores, televisores o impresiones. En estos casos, la resolución determina la calidad y la capacidad de reproducción fiel de los detalles de la imagen.

Cuando se trata de imágenes digitales, es común encontrar dos tipos de resolución: la resolución espacial y la resolución de impresión.

Resolución espacial: Se refiere a la cantidad de píxeles contenidos en una imagen digital y determina la calidad de la visualización en dispositivos electrónicos, como monitores y pantallas. Una mayor resolución espacial se traduce en una imagen más nítida y detallada. Por ejemplo, una imagen con resolución de 1920x1080 píxeles se considera de alta definición [HD], mientras que una imagen con resolución de 3840x2160 píxeles se denomina resolución ultra alta definición [UHD] o 4K.

⁶⁹ Una pulgada son 2,54 centímetros.



Resolución original, 910 x 910 píxeles⁷⁰



La misma fotografía pero en 91 x 91 píxeles⁷¹

Resolución de impresión: Es la cantidad de píxeles por unidad de longitud que se utiliza para imprimir una imagen. Se mide en píxeles por pulgada (*ppi*) o puntos por pulgada (*dpi*). Una mayor resolución de impresión garantiza una imagen más nítida y detallada al imprimirse. Por ejemplo, una resolución de impresión de 300 ppi es utilizada habitualmente para obtener impresiones de alta calidad.

4.3 Los espacios de color

Como ya hemos nombrado el espacio de color se refiere a la manera en que se representa y organiza la información del color en cada pixel en una imagen. Existen varios espacios de color comunes que se utilizan para representar y manipular imágenes digitales. Veamos un poco más en profundidad los más usados.

El **espacio de color RGB** es el más utilizado en las imágenes digitales. En este espacio, el color se representa mediante la combinación de tres canales: rojo [R], verde [G] y azul [B]. Cada canal se define con valores que van desde 0 hasta 255, lo que representa la intensidad de cada componente de color. Por ejemplo, un píxel con un valor de [255, 0, 0] tendrá un color rojo puro [R], mientras que un valor de [0, 255, 0] representará un color verde puro [G]. La combinación de diferentes intensidades de los tres canales permite obtener una amplia gama de colores. El espacio de color RGB se utiliza ampliamente en aplicaciones como pantallas de ordenadores, cámaras digitales y CGI.

70 Autor: Leroy_Skalstad (pixabay). No citaré su autoría en el resto de las figuras que incluyan esta imagen.

71 Aunque la imagen es de 91x91, la he ampliado para ver los pixels individuales.

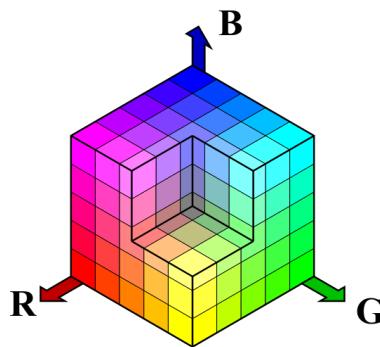
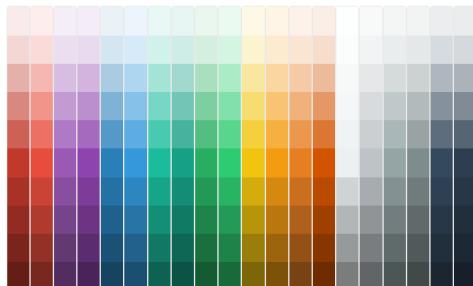


Figura 11: Cubo RGB

Autor: Mike Horvath



Por lo tanto, cada color que ves a la izquierda, de los más de 16 millones que RGB puede generar, se codifican como un triplete de números (cada uno, 1 byte) en la memoria del ordenador.

Una imagen de 1920 x 1080 [Full HD] tiene más de 2 millones de pixels, cada uno de ellos ocupando 3 bytes en memoria [por tanto esta imagen ocupará ~ 6MB en total].

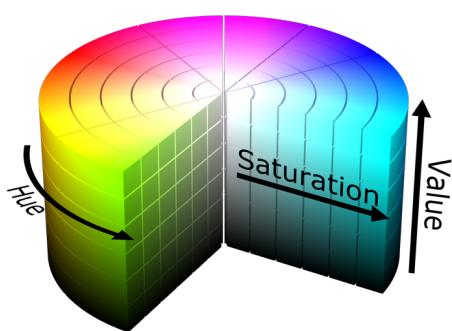


Figura 12: Cilindro HSV

Autor: Wikipedia Commons User:Datumizer

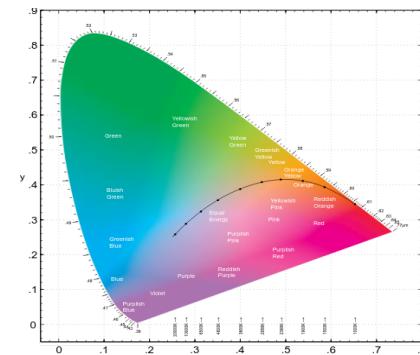


Figura 13: Espacio de color LAB

Autor: Wikipedia User:Fuzzypeg

Espacio de color HSV [*hue, saturation, value*]: Este espacio de color se basa en características perceptuales y es ampliamente utilizado en aplicaciones de procesamiento de imágenes, como el seguimiento de objetos y la detección de bordes.

El canal de matiz [H. hue] representa el tono del color, la saturación [S] representa la intensidad

del color y el valor [V] representa el brillo.

Espacio de color Lab: El espacio de color Lab es un espacio de color basado en las características de percepción humana. El canal L representa la luminosidad, mientras que los canales a y b representan la información de crominancia, que describe el matiz y la saturación. Observa que nuestra percepción se basa principalmente, en el verde y el rojo, siendo el color azul (Figura 13) el menos representado.

Cada espacio de color tiene sus ventajas y aplicaciones específicas en el procesamiento de imágenes y la visión artificial. La elección del espacio de color adecuado depende del contexto y los requisitos del problema en particular.

4.4 Los formatos de archivo para imágenes más comunes

Ya sabemos cómo codificamos una imagen, pero para almacenarla o transmitirla es necesario dotarla de un formato. Este apartado trata sobre el formato de archivo en imágenes dentro del contexto de la visión artificial. El formato de archivo de una imagen es la manera en que se almacenan y codifican los datos de la imagen en un archivo. Cada formato tiene sus propias características y ventajas, y la elección del formato adecuado depende del uso previsto y los requisitos específicos de la aplicación.

Veamos algunos de los formatos de archivo más comunes utilizados:

- **JPEG** [*joint photographic experts group*]: es un formato de compresión con pérdida ampliamente utilizado para imágenes fotográficas. La compresión con pérdida implica una reducción en la calidad de la imagen, pero permite tamaños de archivo más pequeños. Es adecuado para imágenes web y fotografías digitales.
- **PNG** [*portable network graphics*]: es un formato de compresión sin pérdida que se utiliza para imágenes en las que se requiere una calidad alta y no se desea perder información. Admite transparencia (canal alfa), lo que implica que se pueden definir áreas transparentes en la imagen. PNG es adecuado para imágenes que requieren una alta calidad, como imágenes de logotipos e ilustraciones.
- **GIF** [*graphics interchange format*]: es un formato de imagen que admite animaciones (secuencias de imágenes) y una paleta indexada de hasta 256 colores. Es un formato antiguo, pero todavía se usa para imágenes animadas en la web. GIF también es adecuado para imágenes con fondos transparentes.
- **BMP** [*bitmap*]: es un formato de imagen sin comprimir que almacena datos de imagen pixel por pixel. Debido a su falta de compresión, los archivos BMP pueden ser muy grandes en comparación con otros formatos. BMP se utiliza a menudo en aplicaciones que requieren imágenes sin pérdida y edición directa de píxeles, como algunas aplicaciones de edición de imágenes.
- **TIFF** [*tagged image file format*]: es un formato de archivo versátil que admite tanto com-

presión sin pérdida como compresión con pérdida. Es ampliamente utilizado en aplicaciones profesionales, como la fotografía, la impresión y la industria editorial, donde se requiere una alta calidad de imagen y se permite un tamaño de archivo más grande.

- **RAW:** Un archivo RAW es un archivo de imagen que contiene toda la información capturada por el sensor de una cámara digital sin procesar⁷². Los archivos RAW son grandes pero ideales para dispositivos con mucha memoria pero poco poder de procesamiento [como las cámaras digitales].

Cada formato de archivo tiene sus propias características y limitaciones, y es importante considerar el propósito y los requisitos de la imagen antes de elegir el formato adecuado. Al seleccionar un formato de archivo, es esencial equilibrar la calidad de la imagen, el tamaño del archivo y la compatibilidad con las aplicaciones y plataformas en las que se utilizará la imagen.

4.5 Histogramas

Hasta aquí una muy breve introducción a las imágenes y sus formato. Ahora vamos estudiar las imágenes por medio de histogramas y su papel en el análisis de las mismas. El histograma es una representación gráfica de la frecuencia de aparición de los diferentes niveles de intensidad en una imagen. Proporciona información importante sobre la distribución tonal de una imagen y es una herramienta fundamental en el procesamiento y análisis de imágenes.

El histograma se puede visualizar como un gráfico de barras, donde en el eje horizontal se representan los niveles de intensidad y en el eje vertical se muestra la frecuencia de aparición de cada nivel. Los niveles de intensidad generalmente se representan en una escala de 0 a 255 para imágenes en escala de grises o para cada componente de color [rojo, verde, azul] en imágenes en color RGB.

04 Operaciones básicas con imágenes.ipynb

Prompt:

Dada una imagen dada mediante una url, calcula y muestra su histograma. La gráfica ha de ser sólo una línea.

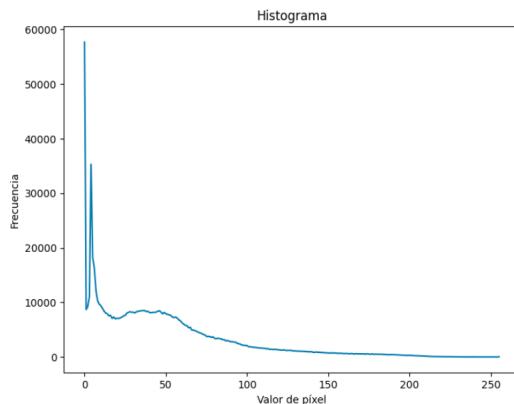
El código generado escoge una imagen y la transforma a escala de grises para tener únicamente un canal, y así simplificar el histograma⁷³. Observa a la izquierda la imagen, ¿qué tonos de la escala de grises son más frecuentes, los oscuros [cercaos a cero] o los claros [cercaos a 255]?

Ahora observa el histograma, ¿qué columnas/barras/valores son más frecuentes, los cercanos a cero o los cercanos a 255?

⁷² Su nombre no es un acrónimo, 'raw' en castellano se traduce como 'crudo'.

⁷³ Para imágenes en color (RGB = tres canales), se hace un histograma diferente por cada canal de color.

Imagen original



Un histograma se calcula contando pixels⁷⁴ de un determinado color. En nuestro caso son tonos de gris, así que se empieza con el 0, contamos cuantos pixels hay con el valor 0 [negro] y esa será la altura de la columna. Y así hasta el pixel de valor 255 [blanco].

El análisis del histograma es una técnica esencial en el procesamiento de imágenes y se utiliza en una amplia variedad de aplicaciones, como el ajuste de exposición, la mejora de imágenes, la detección de bordes y la segmentación. Proporciona información cuantitativa sobre la distribución de los niveles de intensidad, lo que permite comprender y manipular la estructura tonal de una imagen de manera eficiente. En concreto, el análisis del histograma proporciona información valiosa sobre el contraste, el brillo, la distribución tonal y otros aspectos importantes de una imagen. Vamos a ver algunos de ellos:

- **Contraste:** El histograma permite evaluar el contraste de una imagen. Un histograma con una distribución uniforme indica un bajo contraste, mientras que un histograma concentrado en un rango estrecho de niveles de intensidad sugiere un alto contraste.
- **Brillo:** El brillo de una imagen se relaciona con la distribución de los niveles de intensidad en el histograma. Un histograma desplazado hacia la izquierda indica una imagen oscura, mientras que un desplazamiento hacia la derecha indica una imagen más clara.
- **Distribución tonal:** El histograma muestra la distribución de los niveles de intensidad en una imagen. Puede revelar si la imagen tiene una distribución equilibrada de tonos o si está dominada por ciertos rangos de intensidad, lo que puede ser útil para ajustar el procesamiento de la imagen.

Pero los histogramas no sólo se usan para hacer un análisis básico de la imagen, también se usan como forma de hacer segmentaciones básicas⁷⁵ por color, mediante el uso de filtros paso banda, ya veremos un ejemplos.

⁷⁴ No pretendo repetirme, pero recuerda que cuando hablo de pixels me refiero a la unidad mínima en una imagen. Si esta está en RGB, un pixel está formado por tres números, si está en escala de grises un pixel está formado por un número, típicamente de 0 a 255 (un byte, 8 bits).

4.6 Las transformaciones geométricas comunes y su aplicación en el procesamiento de imágenes.

Además de analizar y transformar las imágenes por medio del color, también podemos llevar a cabo transformaciones geométricas en su procesamiento. Estas transformaciones son operaciones que modifican la geometría de una imagen, como su posición, tamaño, orientación o perspectiva. Son útiles para corregir distorsiones, alinear imágenes, cambiar su escala, rotarlas o trasladarlas.

Nota: hablamos de operaciones de transformación geométrica de una imagen, pero lo dicho para imágenes se puede aplicar también a partes o regiones de estas.

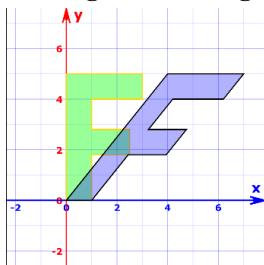
Las transformaciones geométricas son fundamentales en diversas aplicaciones de visión artificial, como la corrección de imágenes en fotografía, la alineación de imágenes médicas, la realidad aumentada, el reconocimiento de objetos y la navegación autónoma de robots, entre otras.

A continuación, se describen algunas de las transformaciones geométricas más comunes utilizadas en el procesamiento de imágenes:

- **Traslación:** Esta transformación desplaza una imagen a lo largo de un vector definido por las coordenadas $[tx, ty]$. Cada píxel de la imagen original se mueve a una nueva posición determinada por la adición de los valores de traslación a sus coordenadas originales.
- **Rotación:** La rotación gira una imagen alrededor de un punto de referencia, especificando un ángulo de rotación, de forma que, para cada píxel de la imagen original, se calcula su nueva posición después de aplicar la rotación.
- **Escala:** Esta transformación modifica el tamaño de una imagen. Puede aumentar o disminuir las dimensiones de la imagen original. Se definen factores de escala para las dimensiones horizontal y vertical, y se aplica la transformación a cada píxel de la imagen.
- **Sesgo o cizallamiento (shearing):** El *shearing* es una transformación que deforma una imagen al inclinarla en una dirección determinada. Se definen parámetros de sesgo para las dimensiones horizontal y vertical, y se calcula la posición de cada píxel después de aplicar la transformación.
- **Perspectiva:** La transformación de perspectiva se utiliza para simular la proyección de una imagen en un plano tridimensional. La perspectiva se utiliza para corregir la distorsión en imágenes tomadas desde diferentes ángulos o posiciones.

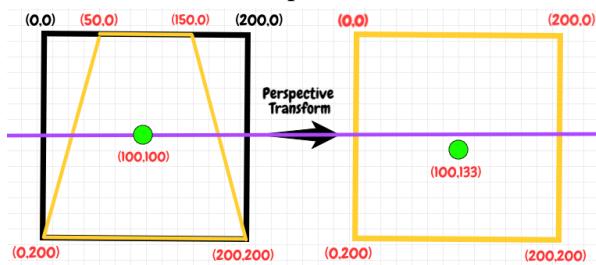
75 La segmentación de imágenes es el proceso de dividir una imagen en diferentes regiones o segmentos significativos, con el objetivo de simplificar la representación de la imagen y facilitar el análisis y comprensión de su contenido.

Shearing de una imagen



Fuente: MathIsFun.com

Perspectiva



Fuente: <https://medium.com/Analytics-vidhya/opencv-perspective-transformation>

Es importante destacar que estas transformaciones geométricas se aplican a nivel de píxeles y pueden introducir distorsiones en la imagen transformada. Por lo tanto, es fundamental considerar técnicas de interpolación⁷⁶ y ajuste adecuadas para preservar la calidad y la información visual durante el proceso de transformación.

4.7 Operaciones entre imágenes

Como hemos dicho al principio de este capítulo, una imagen puede ser vista como una matriz de n por m pixels o, si cada pixel tiene k canales, una matriz tridimensional⁷⁷ de n por m por k . Eso nos da la oportunidad de transformar la imagen, como acabamos de ver, y de operar imágenes entre ellas.

Las operaciones entre imágenes son técnicas matemáticas que se aplican para manipular o analizar dos o más imágenes. Estas operaciones son fundamentales en el procesamiento de imágenes y la visión por computadora.

Las operaciones suelen ser pixel a pixel, sumando, restando etc. pixels con las mismas coordenadas; pero también podemos realizar operaciones como la de **convolución**, en donde una pequeña máscara denominada *kernel* “recorre” la imagen principal obteniendo pequeñas muestras escalares que sustituirán al pixel original en la nueva imagen. Lo veremos en el próximo capítulo.

A continuación, se describen algunas de las operaciones más comunes.

Suma de imágenes: La suma de imágenes consiste en sumar los valores de los píxeles correspondientes de dos imágenes. Esto se utiliza a menudo para mejorar la calidad de la imagen o para combinar características de múltiples imágenes.

Resta de imágenes: La resta de imágenes se emplea para destacar las diferencias entre dos imágenes.

⁷⁶ En el contexto del procesamiento de imágenes, la interpolación es un método para calcular nuevos valores de píxeles entre los píxeles existentes. Esto se puede hacer para aumentar la resolución de una imagen, para reducir el tamaño de una imagen o para suavizar los bordes en una imagen.

⁷⁷ Un tensor de tres dimensiones.

nes. Esto es especialmente útil en aplicaciones de vigilancia o para resaltar cambios en secuencias de imágenes.

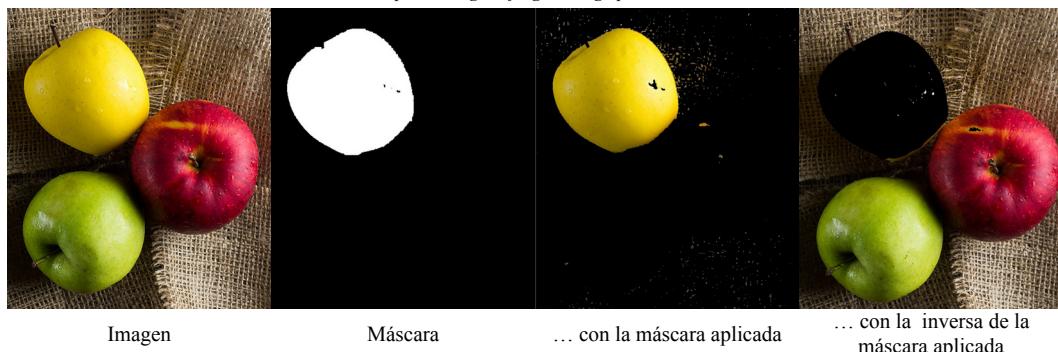
Multiplicación de imágenes: La multiplicación de imágenes se realiza multiplicando los píxeles correspondientes de dos imágenes. Esta operación es útil para cambiar la opacidad de una imagen o para realizar una corrección de iluminación.

División de imágenes: La división de imágenes implica dividir los píxeles de una imagen por los de otra. Esto se utiliza para normalizar la iluminación o para destacar ciertas áreas de una imagen.

Operaciones booleanas a nivel de bit: AND, OR, XOR y NOT: Estas son operaciones lógicas que se realizan en imágenes binarias. Permiten la manipulación y análisis de imágenes en términos de sus componentes binarios.

Figura 14: Resultados de aplicar una máscara a una imagen (and y and not)

Fuente: <https://www.geeksforgeeks.org/opencv-invert-mask>



Una de las operaciones más útiles que podemos hacer con las operaciones booleanas es la aplicación de máscaras. Una máscara es una imagen de control, esto es, su función no es la representación si no la aplicación sobre otras imágenes [Figura 14, “máscara”]. Suele ser de un sólo canal y contener, o bien 0 y 1 [binarias⁷⁸], o bien valores entre 0 y 255⁷⁹ [de escala].

En la Figura 14 podemos observar como una máscara binaria [0 y 255] es aplicada sobre una imagen RGB, que posee tres canales. En este caso se aplica la máscara a cada uno de los canales de forma independiente. Si la operación es *imagen and máscara*, el resultado es 0 donde la máscara es 0 y el valor del pixel original donde el valor de la máscara es 255. El resultado puede verse en la Figura 14 “... con la máscara aplicada”.

También podemos hacer lo contrario si aplicamos *imagen and not máscara*. En este caso primero se lleva a cabo la operación **not** sobre la máscara, lo que produce que donde originalmente la máscara tenía un 0 (00000000 en binario) ahora tiene 255 (11111111 en binario) y viceversa, donde tuviese un 255, ahora tendrá un 0. El resultado de aplicar esta máscara inversa puede verse en Figura 14 “... con la inversa de la máscara aplicada”

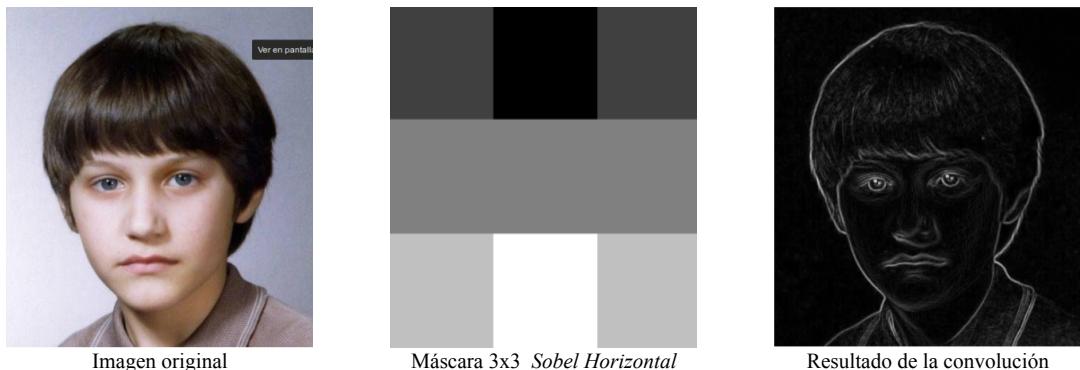
78 Binarias realmente significa que la máscara sólo posee dos valores, típicamente 0 y 1, o 0 y 255 (11111111 en binario).

79 Recordemos, damos por supuesto que cada pixel de un canal ocupa un byte, por tanto poseerá valores de 0 a 255.

Combinación Lineal: La combinación lineal consiste en aplicar una suma ponderada de dos o más imágenes. Esto se utiliza para mezclar imágenes de manera más controlada que la suma simple.

Correlación y convolución: Estas son operaciones más avanzadas que involucran el uso de un filtro o una máscara para modificar una imagen en función de otra. Se utilizan comúnmente para el desenfoque, el realce y la detección de bordes [ver Figura 15].

Figura 15: Aplicación de una máscara de convolución Sobel Horizontal a una imagen RGB



La imagen de la derecha es el resultado de aplicar la máscara Sobel Horizontal a la imagen de la izquierda. El código fue generado con el siguiente *prompt*:

04 Operaciones básicas con imágenes.ipynb

Prompt:

Dada una imagen, dada mediante una url, aplicar un filtro de convolución para detectar los bordes. Representa⁸⁰ también el filtro como una imagen. Usa la librería skimage.

Respecto a la operación de convolución.

Mi IA favorita la describe así: La convolución de una máscara sobre una imagen implica mover la máscara sobre cada píxel de la imagen. Para cada posición de la máscara, el valor del píxel de salida se calcula multiplicando cada valor del píxel cubierto por la máscara por el correspondiente valor del coeficiente de la máscara y luego sumando estos productos. El resultado se asigna al píxel correspondiente en la imagen de salida.

⁸⁰ Aquí ignoró nuestra petición de representar la máscara, así que la generamos a mano para mostrarla.

RETOS DEL CAPÍTULO 4

1. ¿Cuál es la diferencia entre una imagen digital y una imagen analógica?
2. Pregúntale a tu AI favorita, cómo se integra en una imagen RGB información de opacidad, qué es y si es posible operar esta con otras canales semejantes.
3. ¿Qué es el muestreo y cómo afecta la calidad de una imagen?
4. Pregúntale a tu AI favorita, qué técnicas hay para redimensionar una imagen. Pídele que te explique cada una de ellas y cual es la mejor.
5. Tengo una imagen de 1024 columnas por 768 filas, canales RGB de 8 bits cada uno y un canal *alpha* de igualmente 8 bits. ¿Cuanto ocupa en memoria?
6. Pregúntale a tu AI favorita, qué más canales puede tener una imagen además de los relacionados con los colores y la opacidad. Pídele ejemplos de uso.
7. Habitualmente estamos usando imágenes con canales de color [típicamente RGB] de 8 bits [1 byte] cada uno de ellos. Pregúntale a tu AI favorita que te explique, por qué se usan canales de 32 bits en punto flotante y qué es normalizar una imagen.
8. Pregúntale a tu AI favorita que te explique la visión humana de los colores y el espacio de color Lab. Pídele que te explique el color azul desde un punto de vista histórico, en concreto ¿por qué no aparecen referencias al color azul en la antigüedad?
9. Crea un *notebook* que cargue y lea una imagen, para después aplicarle las transformaciones geométricas más comunes, visualizando los resultados.
10. Busca en la web una imagen que tenga algún elemento de color chillón, como la manzana amarilla de la Figura 14. Descarga la imagen y con ayuda de su IA favorita crea un *notebook* que cree una máscara que al aplicarla a la imagen aísle el elemento.
11. Pregúntale a tu AI favorita, qué es la operación de convolución sobre una imagen. Pídele que te lo explique varias veces: [1] como un niño o niña de 10 años, [2] como un/una adolescente de 16 años, [3] como un/una universitaria de 22 años y como un/una profesional del tratamiento de imágenes con avanzados conocimientos de matemáticas. Busca los paralelismos entre las respuestas.

Capítulo 5

CLASIFICACIÓN, IDENTIFICACIÓN Y SEGMENTACIÓN DE PATRONES VISUALES

En el apartado 2.4 presentamos el concepto de agente inteligente por medio de sus componentes, uno de ellos era el encargado de la **percepción**. La percepción del mundo que rodea al agente es un área muy importante de la IA, y depende significativamente del mecanismo de percepción: audio, sensores, o .. imágenes. En este último caso el objetivo final es dotar al agente de la capacidad de convertir información gráfica en información que pueda ser procesada por otros componentes del sistema. Esto incluye la interpretación de imágenes digitales para realizar tareas como, entre otras, la detección y reconocimiento de objetos, segmentación de imágenes y la clasificación de estas.

La clasificación, identificación y segmentación de patrones visuales en el contexto de la visión artificial es un área de investigación y desarrollo que busca dotar a las máquinas de la capacidad de analizar y comprender la información visual de manera similar a los seres humanos.

A través de algoritmos y técnicas de procesamiento de imágenes, extracción de características y aprendizaje automático, se abre un mundo de posibilidades para aplicaciones prácticas en diversos campos, mejorando nuestra capacidad para interpretar y utilizar el vasto conjunto de datos visuales disponibles en la actualidad.

En esencia, este capítulo se enfoca en mostrar las técnicas que permiten a las máquinas identificar, reconocer y segmentar patrones visuales en imágenes o secuencias de video. Esto implica el análisis de características visuales, como colores, texturas, formas y contornos, para **extraer información significativa** y comprender la estructura subyacente de las imágenes.

El capítulo siguiente se dedicará a la percepción visual, un paso más allá en la cognición.

Para llevar a cabo la extracción de características se usa la clasificación, identificación y/o la segmentación de imágenes, la cual consiste en dividir una imagen en regiones o segmentos significativos, lo que facilita un análisis más detallado y un procesamiento posterior más preciso.

En la Figura 16 podemos ver la diferencia entre clasificación, identificación y segmentación:

- **Clasificar y localizar** un objeto en una imagen implica localizar y clasificar el objeto buscado [o los objetos en caso de ser varios]. Para cada objeto encontrado obtendremos el rectángulo en donde se considera que se encuentra el objeto y una etiqueta indicando qué objeto se considera que es, además de una confianza⁸¹ asociada.
- La **segmentación semántica** es algo más simple de entender pero complejo de hacer. Con esta técnica se consigue clasificar cada pixel de la imagen con una etiqueta. Lo que obtenemos realmente es una máscara, de mismo tamaño que la imagen, indicando para cada pixel a que categoría pertenece.
- La **segmentación de instancias** es una técnica que aúna las dos anteriores, por un lado detecta y clasifica instancias de objetos, y por otro realiza una segmentación del elemento detectado con el objeto de diferenciar qué pixels pertenecen al objeto y cuales no, reduciendo el espacio de segmentación al mínimo⁸².

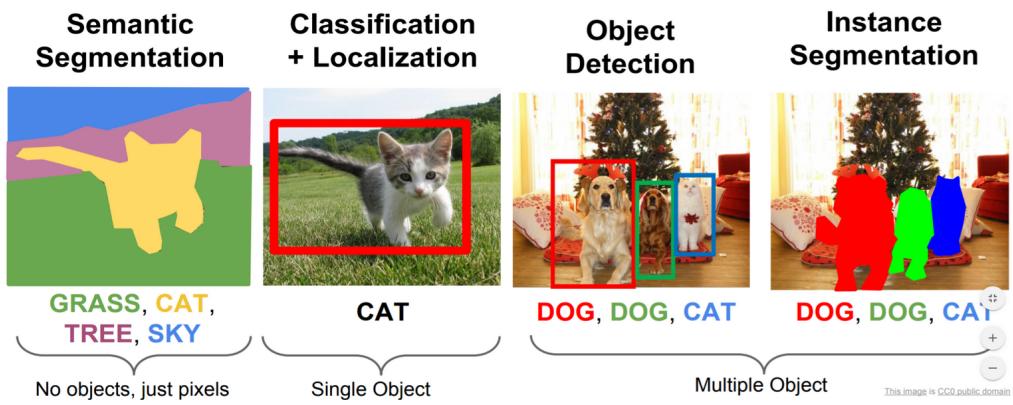


Figura 16: Diferencia entre segmentación e identificación.

Fuente: P. Jaikumar, R. Vandaele; "Transfer Learning for Instance Segmentation of Waste Bottles using Mask R-CNN Algorithm"; CC BY 4.0; ISDA 2020, Springer.

Hasta ahora hemos hablado de imágenes, pero ¿qué pasa con los vídeos? Los vídeos son imágenes individuales mostradas secuencialmente con una cadencia concreta, que generan la sensación de movimiento. Por contra, el cerebro humano procesa el mundo real [el vídeo de la realidad, por hacer una analogía] no como imágenes sueltas, si no por medio del uso de circuitos neurales especia-

81 De hecho y dependiendo del algoritmo o técnica usada, obtendremos una lista de objetos y sus estimaciones de confianza. Lo normal es podar estos objetos quedándose con las confianzas más grandes.

82 O incluso reducirlo a una zona de interés.

lizados que mantienen actualizado una **representación mental del mundo visible** en un momento dado.

Mi IA favorita dice: La cadencia mínima estándar para mostrar una secuencia de imágenes individuales y conseguir la ilusión de movimiento es de 12 fotogramas por segundo (fps). Sin embargo, la mayoría de las personas pueden ver un movimiento fluido a una cadencia de 15-20 fps. A medida que la cadencia aumenta, el movimiento se vuelve más fluido y realista. La cadencia estándar para las películas es de 24 fps, y la cadencia estándar para la televisión es de 30 fps. Algunas aplicaciones, como los videojuegos, pueden utilizar cadencias mucho más altas, de hasta 120 fps o más.

5.1 Visión artificial

Como hemos dicho, la visión artificial es una disciplina que tiene como objetivo conseguir que las máquinas perciban y comprendan el mundo visual de forma semejante a como lo hacen los seres humanos. Esto implica el desarrollo de algoritmos y técnicas que permiten el análisis y la interpretación de imágenes y videos digitales.

There exist several methods to design forms to be filled in. For instance, fields may be surrounded by lines, by light rectangles or by guiding rods. These rods specify where to write and, therefore, no skew and overlapping with other parts of the form. Guides can be located on a separate sheet located below the form or they can be printed on the form. The use of guides on a separate sheet from the point of view of the quality of the form is better, but requires giving more instructions and, therefore, restricts its use to tasks where this type of a guide is suitable.

There exist several methods to design forms to be filled in. For instance, fields may be surrounded by lines, by light rectangles or by guiding rods. These rods specify where to write and, therefore, no skew and overlapping with other parts of the form. Guides can be located on a separate sheet located below the form or they can be printed on the form. The use of guides on a separate sheet from the point of view of the quality of the form is better, but requires giving more instructions and, therefore, restricts its use to tasks where this type of a guide is suitable.

Figura 17: Eliminación de ruido usando redes neuronales artificiales (entrada/salida).

Fuente: desconocida.

Como ya hemos dicho, en general se llevan a cabo en dos etapas:

- El **preprocesamiento de imágenes** es la etapa inicial en la que se busca mejorar la calidad de las imágenes antes de someterlas a un análisis más profundo. Esto puede involucrar operaciones como el filtrado para eliminar el ruido, la corrección de la iluminación o el contraste, y la eliminación de características no deseadas. Habitualmente también esta etapa se encarga de *unificar* las imágenes, igualarlas en algún aspecto con el fin de que la última etapa [extracción de características] no tenga que enfrentarse con demasiada variabilidad.

Dentro del preprocesamiento podemos destacar la subetapa de **mejora de la calidad de la**

imagen, la cual busca resaltar características relevantes y reducir la interferencia o distorsión. Esto puede lograrse mediante técnicas de realce de bordes, ecualización de histogramas, filtrado espacial u otras transformaciones que mejoren la visibilidad de los detalles importantes en la imagen (ver Figura 17).

- Finalmente, la **extracción de características** es la etapa crucial en el procesamiento de imágenes e implica identificar y extraer atributos significativos de las imágenes que puedan utilizarse para tareas posteriores de mayor nivel cognitivo.



Vamos a ver algunas de estas tecnologías en más profundidad.

5.1.1 Métodos y técnicas para mejorar la calidad de las imágenes.

El preprocessamiento de imágenes es una etapa fundamental para mejorar la calidad y facilitar la extracción de características de las imágenes antes de realizar cualquier tarea de análisis o reconocimiento.

Vamos a describir algunas técnicas comunes utilizadas en el preprocessamiento de imágenes:

- Filtrado de imágenes:** El filtrado de imágenes se utiliza para suavizar o resaltar ciertas características en una imagen. Los filtros más comunes incluyen el filtro de media, que suaviza la imagen al calcular el promedio de los píxeles vecinos, y el filtro de mediana, que se utiliza para eliminar el ruido impulsivo al reemplazar cada píxel por la mediana de sus vecinos. Otros filtros populares son los filtros de paso bajo, como el filtro Gaussiano, que se utiliza para eliminar el ruido de alta frecuencia, y los filtros de paso alto, como el filtro de realce de bordes, que se utiliza para resaltar los bordes en una imagen.
- Mejora de contraste:** La mejora de contraste se utiliza para aumentar la diferencia entre los niveles de intensidad de los píxeles en una imagen. Esto se logra mediante técnicas como la ecualización del histograma, que redistribuye los valores de intensidad en el histograma de la imagen para abarcar todo el rango dinámico disponible.
- Corrección de distorsiones:** En ocasiones, las imágenes pueden sufrir distorsiones debido a problemas de la lente de la cámara o a la perspectiva. La corrección de distorsiones se utiliza para corregir estas imperfecciones y obtener una representación más precisa de la escena. Esto se logra mediante técnicas como la corrección de la distorsión radial y tangencial, que ajusta las deformaciones causadas por la lente de la cámara, o mediante técnicas de corrección de perspectiva, que rectifican la imagen para obtener una vista frontal.

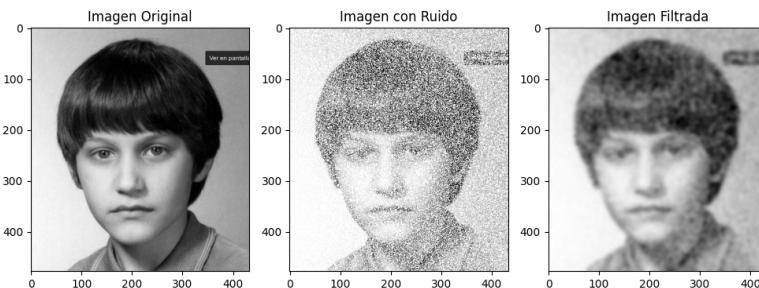


Figura 18: Uso de filtros para eliminar ruido de una imagen.

- **Eliminación de ruido:** El ruido en una imagen [ver Figura 18] puede afectar negativamente el proceso de identificación y segmentación de patrones visuales. Las técnicas de eliminación de ruido se utilizan para reducir o eliminar los artefactos no deseados, como el ruido de fondo o el ruido producido durante la adquisición de la imagen. Esto se puede lograr mediante filtros de reducción de ruido, como el filtro de suavizado bilateral, que preserva los bordes mientras suaviza el ruido.



Figura 19: Ejemplo de mejora de contraste.

- **Normalización de imágenes:** La normalización de imágenes es útil cuando se desea comparar o combinar imágenes de diferentes fuentes o condiciones de iluminación. Esto implica ajustar las imágenes para que tengan características similares, como el rango de intensidad o el balance de color, lo que facilita la comparación y el análisis.

A la hora de llevar a cabo el preprocesamiento para mejorar la calidad de las imágenes, contamos con dos escuelas: [1] el uso de algoritmos de procesamiento de imágenes y [2] el uso de redes neuronales.

La primera aproximación usa algoritmos como los nombrados en las líneas anteriores en este punto, funcionan relativamente bien [véase la Figura 18 y la Figura 19] pero poseen la característica de necesitar el análisis humano para determinar qué estrategia usar.

El uso de redes neuronales es una aproximación más genérica. Consiste en entrenar una red neuronal para que, dada una imagen con ruido [u otro problema, pero nos centraremos en el ruido] obtener la imagen limpia [ver Figura 17]; y, como ya iremos intuyendo, el problema son los datos

¿cómo obtener una gran cantidad de imágenes con ruido y su contraparte limpia de estas? Por ejemplo, mediante la **generación de datos sintéticos**: a partir de imágenes correctas [sin ruido] generaremos artificialmente imágenes sintéticas con ruido añadido, el mismo ruido que queremos eliminar. Esta estrategia funciona muy bien pero tiene sus problemas: [1] generar el mismo tipo de ruido que queremos eliminar no es trivial y las redes neuronales son tan sensibles que pueden diferenciar el ruido natural del generado y [2], quizás el problema más importante, al eliminar el ruido, la información que la sustituye es información sintética, esto es, inventada o imaginada, algo no muy adecuado para tareas críticas⁸³.

La elección de las técnicas específicas depende del problema y de las características de las imágenes involucradas. El preprocesamiento de imágenes desempeña un papel crucial en la mejora de la calidad de las imágenes y en la **preparación de los datos** para su posterior análisis y reconocimiento.

5.2 Reconociendo patrones: clasificación

El **aprendizaje automático** es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos y modelos que pueden aprender de los datos y tomar decisiones o hacer predicciones sin ser programados explícitamente. En el contexto de la visión artificial, el aprendizaje automático [muy especialmente el aprendizaje profundo] ha demostrado ser muy eficaz para el reconocimiento de patrones visuales y la clasificación de imágenes.

Un enfoque comúnmente utilizado en el aprendizaje automático para el reconocimiento de patrones visuales es el uso de redes neuronales artificiales, específicamente las **redes neuronales convolucionales** [en adelante CNN]. Las CNN⁸⁴ son especialmente efectivas en el procesamiento de imágenes, ya que pueden aprender características visuales de bajo y alto nivel a través de múltiples capas, lo que les permite reconocer patrones complejos.

Nota: por ahora imagina una red neuronal artificial como un conjunto de números organizados en capas, cuyo proceso de aprendizaje es calcular el valor de dichos números para que, a partir de una entrada, entregue una salida cercana o igual a la deseada. A esos números los llamamos parámetros o pesos⁸⁵.

En el contexto de la clasificación de imágenes, el **proceso de entrenamiento** de una CNN implica presentarle un conjunto de imágenes etiquetadas [*dataset*⁸⁶], es decir, imágenes que tienen asociadas una clase o categoría [recordemos: estamos en el contexto de la clasificación]. Durante el en-

83 Por ejemplo, si el problema a que nos enfrentamos es la localización de cánceres en radiografías y estas tienen ruido, el preprocesamiento usando generación de imágenes sintéticas podría “imaginar” pixels en la radiografía que posteriormente podrían ser identificadas como cánceres.

84 Internamente son redes neuronales artificiales con una arquitectura tal que permiten “centrarse” en porciones de la imagen y, capa por capa, extraer sus características principales, independientemente de donde se encuentren.

85 Hay un pequeño matiz entre parámetros y pesos: los pesos son parámetros entrenables. Lo veremos más adelante.

86 Acostúmbrate a este término, aunque en castellano tenemos su traducción (conjunto de datos), en muchas fuentes usan este anglicismo (además de otros).

trenamiento, la red neuronal **ajusta los pesos de sus conexiones** internas para **minimizar** la diferencia entre las predicciones que realiza y las etiquetas correctas de las imágenes de entrenamiento. Este proceso se realiza mediante la optimización de una función de pérdida.

Nota: El proceso de entrenamiento de una red neuronal artificial es, curiosamente, muy humano. Imagina que debes aprender algo, sea de carácter memorístico o práctico, da igual. Sólo tienes una gran cantidad de ejemplos, los cuales divides en ejemplos de entrenamiento (casi todos) y ejemplos de validación (unos cuantos)⁸⁷. El entrenamiento consiste en repetir una y otra vez la siguiente secuencia: (1) aprendes mediante los ejemplos de entrenamiento, (2) te autoexaminas obteniendo una nota y (3) miras en que has fallado y aprendes de esos fallos. Finalmente vuelves al punto 1 mientras que tu nota sea inferior a una nota objetivo. Al acabar te auto-evaluas de nuevo usando los ejemplos de validación.

Una vez que la CNN ha sido entrenada, puede utilizarse para **clasificar** nuevas imágenes que no han sido vistas previamente. La red neuronal analiza la imagen de entrada y produce una salida que representa la puntuación [score] de pertenencia a cada una de las clases o categorías previamente definidas. Por ejemplo, en un problema de reconocimiento de objetos, la red podría clasificar una imagen como un "perro" con un score⁸⁸ de un 0.75 y un "gato" con un 0.25⁸⁹. Veamos un ejemplo:

05 clasificación de imágenes.ipynb

Prompt:

Dada una imagen referenciada mediante una url, descargala, muestral y, empleando yolo v8, mostrar el conjunto de scores que ha entregado el algoritmo junto con sus etiquetas correspondientes.

El resultado de ejecutar el código generado por el *prompt* anterior es:



Para esta imagen ...

0: 640x640 1 person, 8.2ms

Speed: 6.4ms preprocess, 8.2ms inference, 1.7ms postprocess per image at shape [1, 3, 640, 640]

Score: 0.84, Label: person

... este fue el resultado de la clasificación

⁸⁷ Típicamente en una proporción 80% - 20%, pero esto puede variar dependiendo de varios factores.

⁸⁸ Hablaremos de "fiabilidad" o "confianza", pero el término es "score" (marcador, puntuación, igual que en el fútbol o el baloncesto) y se nos suele presentar como un número mayor o igual que cero.

⁸⁹ Si le presentásemos una imagen sin un perro ni un gato, obtendríamos scores muy bajos para ambos. Tanto que podríamos determinar que no es ni uno ni otro.

El resultado ha sido que dada la imagen ha detectado una persona [0: 640x640 1 person] en 8.2 milisegundos con un score del 0.84.

El resultado de la clasificación de una imagen es **qué hay en la imagen**, sin más información que el score para cada patrón detectado. Se trata de una visión global de la imagen, y típicamente se usa como paso previo para otro proceso, típicamente de toma de decisiones.

Ejemplo:

Una empresa nos contrata porque tienen un problema. Hacen zumos de manzana y naranja, pero el proveedor cuando les entrega la materia prima se la da mezclada: manzanas y naranjas juntas, en el mismo contenedor. Quieren que hagamos un separador automático de ambas frutas.

La solución consiste en una cinta transportadora que mueve la materia prima una a una, y en un punto dado una cámara hace una fotografía de lo que observa. Nuestro agente *percibe*, mediante la imagen de lo que tiene delante [clasifica la imagen] y dado los scores obtenidos [manzana, naranja] acciona un resorte que empuja las manzanas a la derecha y las naranjas a la izquierda o, en caso de que los scores no sean significativos, deja que la fruta continúe por la cinta transportadora para que sea clasificado manualmente.

Un clasificador de manzanas y naranjas nos entregará dos scores, uno para las manzanas y otro para las naranjas. Cada uno de ellos puede verse como los votos que da el modelo clasificador a cada fruta; si el score de la manzana es superior al de la naranja, es una manzana, y viceversa⁹⁰. Si empatan significa que el modelo no ha sabido diferenciarlos. Si el score es muy bajo, entonces quizás no es ni una cosa ni otra [se les coló un kiwi].

El agente que acabamos de describir percibe una fruta individual sobre la cinta transportadora, toma una decisión y activa (o no) un resorte, actuando sobre su entorno; y para ello ha tenido que pasar por un proceso de aprendizaje (aunque es cierto que es previo a su operativa).

Es importante destacar que el éxito del aprendizaje automático en el reconocimiento de patrones visuales depende en gran medida de la cantidad y calidad de los datos de entrenamiento. Cuanto más variado y representativo sea el conjunto de datos utilizado para entrenar el modelo, mejor será su capacidad para generalizar y reconocer patrones en nuevas imágenes.

Además de las redes neuronales convolucionales, existen otros algoritmos y técnicas de aprendizaje automático que también se utilizan en el reconocimiento de patrones visuales, como las máquinas de vectores de soporte [SVM], los clasificadores Bayesianos, los árboles de decisión y los bosques aleatorios [random forests], entre otros. Cada enfoque tiene sus propias ventajas y limitaciones, y la elección del algoritmo depende del problema específico y de las características de los datos.

⁹⁰ En la realidad no es tan simple. La clasificación ganadora ha de ganar por goleada, no solo debe ser superior a la otra si no lo ha de ser por un margen predeterminado. De hecho podemos convertir la lista de scores (dos o más) en una distribución de probabilidad mediante la función softmax y quedarnos con la clase con más probabilidad (por encima de un mínimo). Lo veremos más adelante.



Para esta imagen ...

0: 640x480 1 cat, 1 bed, 120.0ms

Speed: 28.4ms preprocess, 120.0ms inference, 1051.7ms postprocess per image at shape [1, 3, 640, 480]

Score: 0.84, Label: cat
Score: 0.44, Label: bed

... este fue el resultado de la clasificación

En la clasificación de la imagen anterior, observa que ha identificado que esta contiene un gato [*cat*] con un *score* del 0.84 y una cama [*bed*] con un *score* del 0.44. En este caso, ambas etiquetas [*cat* y *bed*] no compiten entre ellas, dado que el modelo clasificador que he usado [*ultralytics yolov8⁹¹*] no es excluyente⁹², ha detectado un gato y una cama [posiblemente porque fue entrenado con fotos de gatos, y quizás muchas de ellas con el felino sobre una cama].

¿Cuál sería la probabilidad de cada etiqueta? Dicho de otra forma, ¿cómo paso los *scores* a probabilidades? Pues depende, de si las etiquetas son mutuamente excluyentes o no.

En el caso de manzanas y naranjas, ambas etiquetas eran excluyentes, así que usamos la función **softmax**:

$$\text{softmax}(z) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}}$$

Donde z es el vector de *scores* $[x_1, x_2, \dots, x_k]$.

Importante: softmax se aplica cuando la clasificación es excluyente, como el caso de manzanas y naranjas, dado que nos entrega una distribución de probabilidad de sucesos mutuamente excluyentes.

Imaginemos que nuestro clasificador de manzanas y naranjas arroja para una imagen los siguientes *scores*, $z=[0.92, 0.33]$. El resultado de aplicar softmax a nuestro caso es $[0.64, 0.36]$, la cual es una distribución de probabilidad, interpretable como: la imagen contiene una manzana con un 64% de probabilidad y una naranja con un 36%. En caso de que en la imagen hubiese un kiwi, detecta-

⁹¹ Estos modelos de clasificación preentrenados reconocen 1000 etiquetas (<https://deeplearning.cms.waikato.ac.nz/user-guide/class-maps/IMAGENET>)

⁹² Como si lo era el ejemplo de manzanas y naranjas, porque obligué a pasar cada fruta de una en una sobre la cinta transportadora.

riámos este caso antes, con unos *scores* bajos de por ejemplo $[0.21, 0.19]$ ⁹³.

Pero en el caso de la imagen del gato estamos ante sucesos independientes [no necesariamente excluyentes], así que debemos usar la función **sigmoide**:

$$\text{sigmoide}(x_i) = \sigma(x_i) = \frac{1}{1 + e^{-x_i}}$$

Donde x_i es cada uno de los *scores* obtenidos. Si aplicamos esta función a los resultados de la imagen del gato sobre una “cama” $[0.84, 0.44]$, obtendremos las probabilidades de cada uno de ellos: $[0.70, 0.60]$.

Hay más métodos para convertir *scores* a probabilidades, pero las vistas en este apartado son las más comunes.

Nota: la clasificación, en general, no solo sobre imágenes, puede ser binaria (dos etiquetas) o multiclase (más de dos etiquetas), y pueden usarse otras estrategias que no impliquen redes neuronales. Ya lo veremos.

5.3 Reconocimiento de patrones: identificación de objetos

Como hemos dicho al principio de este capítulo, la identificación de objetos en imágenes se refiere al proceso de localizar y clasificar objetos específicos dentro de una imagen. Este proceso se realiza utilizando técnicas de procesamiento de imágenes y aprendizaje automático para analizar y comprender el contenido visual de una imagen y determinar la presencia y ubicación de objetos de interés.

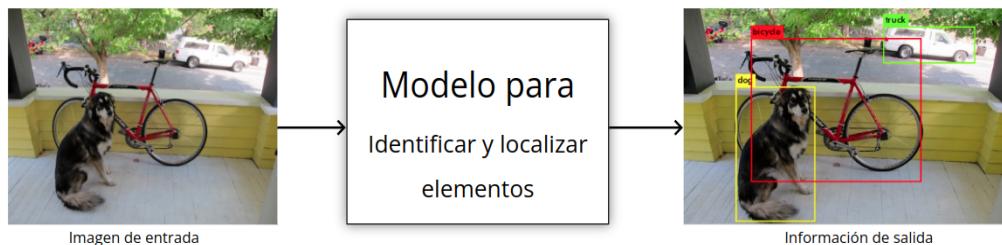


Figura 20: Descripción “black box” de un modelo para identificar y localizar patrones visuales en imágenes.

Fuente de la imagen de entrada: COCO dataset.

Conceptualmente un identificador de objetos [patrones visuales] sigue el esquema de la Figura 20: partimos de un **modelo preentrenado** para detectar los patrones que deseamos obtener, en concreto la localización dentro de la imagen e identificar de qué patrón se trata [en el caso de que hayamos entrenado el modelo para detectar varios patrones].

⁹³ Típicamente se establece un umbral, pongamos un 0.25, y si todos los scores no lo superan suponemos que no es ni manzana ni naranja.

Debemos tener en cuenta:

1. El modelo tuvo que ser entrenado previamente [preentrenado] con el objetivo de reconocer un número finito de patrones [perros, bicicletas, etc.].
2. La imagen de entrada debe ajustarse a lo que espera la red neuronal, tamaño, canales, etc. Esto es, se debe preprocesar la imagen [si bien los modelos que manejaremos ya lo hacen de forma transparente⁹⁴].
3. La información de salida no es una imagen con los elementos identificados [como en la Figura 20] si no una lista de:
 - Cajas [boxes, los dos puntos que forman un rectángulo⁹⁵]
 - Su etiqueta asociada [perro, bicicleta, etc.] en forma de *id* numérico.
 - Una puntuación [score] que nos indica la confianza de la predicción.

Pero ¿y si quisieras identificar elementos en una imagen que no reconocen los modelos preentrenados? Si lo que deseas es detectar elementos concretos para los que el modelo no ha sido entrenado, por ejemplo contar manzanas en imágenes de árboles, debes:

1. **Recolectar** una gran cantidad de imágenes de árboles con manzanas, de todas las formas, colores y posiciones posibles. Estas imágenes deben ser de calidad, pero las manzanas no tienen por qué verse enteras. Dependiendo de tu objetivo, las fotografías deben ser del árbol entero⁹⁶ o de porciones del mismo⁹⁷.
2. **Para cada imagen**, usando una herramienta creada para hacer esta labor: marcar con un rectángulo la posición en donde se hayan las manzanas en el árbol, no importa si estas se solapan, etiquetando al mismo tiempo estos localizaciones como manzanas. Esta labor se denomina **etiquetado** del *dataset*.
3. Usado estas imágenes, entrenar un modelo de reconocimiento de patrones [por ejemplo YOLO⁹⁸], de una forma muy similar a la que vimos en el apartado anterior.

El etiquetado del *dataset* es uno de los procesos más tediosos⁹⁹ que te encontrarás en el aprendizaje automático [aunque no el único], hasta el punto de representar un abultado porcentaje del coste de entrenar un modelo. Por ello se suelen usar **modelos preentrenados**, para reconocer un conjunto relativamente grande de patrones [ver Figura 21]. Pero si no encuentras el modelo ya entrenado para reconocer lo que deseas, te toca entrenar un modelo desde cero¹⁰⁰ [*from scratch*].

94 Aunque este preprocesamiento es transparente, debemos ser conscientes de ellos para lidiar con los errores.

95 Por ejemplo: punto $(x1, y1)$ de la esquina superior izquierda y el punto $(x2, y2)$ de la esquina inferior derecha.

96 Por ejemplo, para contar el número de manzanas visibles en un árbol.

97 Por ejemplo, para identificar el grado de madurez de las manzanas según el color.

98 Referencia en wikipedia: [https://es.wikipedia.org/wiki/Algoritmo_You_Only_Look_Once_\(YOLO\)](https://es.wikipedia.org/wiki/Algoritmo_You_Only_Look_Once_(YOLO))

99 Imagina etiquetar cien mil fotografías ...

100 Cuando hablamos de redes neuronales artificiales en su capítulo correspondiente, te comentaré que es mejor partir de una red previamente entrenada (aunque sea en otra cosa) que partir de una desde cero. El conocimiento puede ser compartido en ámbitos diferentes.

Aún no es el momento de entrenar un modelo, antes vamos a usar los modelos que tenemos a nuestra disposición.

05 identificación de objetos.ipynb

Prompt:

Dada una imagen referenciada mediante una url, descargala, muestral y, empleando yolo v8, muestra las identificaciones que ha hecho [localización, id y score] para finalmente mostrar la imagen con las identificaciones rotuladas.

Produce una imagen con sus identificaciones, pero para esta imagen original, vamos a centrarnos en la mitad inferior de la imagen original:

05 identificación de objetos.ipynb

Prompt:

En la imagen anterior [en el fichero image.jpg¹⁰¹], hacer un zoom en la mitad inferior y volver a hacer la predicción, mostrando los resultados y la imagen con las identificaciones.

El resultado se muestra en la Figura 21, en donde podemos ver las identificaciones que ha llevado a cabo. Podemos observar que ha identificado las personas en primer plano, pero no aquellas más alejadas; la razón de que aquellas más alejadas no hayan sido correctamente identificadas se debe a que el modelo, por defecto, no identifica objetos por debajo de un score dado [0.25].

Nota: el modelo preentrenado que estamos usando usa la función sigmoide para entregar los scores.



Figura 21: Ejemplo del resultado de un modelo preentrenado.

Fuente: Wikimedia (Detroit Publishing Co., publisher).

Estos modelos preentrenados son multiclas, detectan hasta 100 tipos de objetos, en eso consiste su entrenamiento.

¹⁰¹ Se lo tuve que indicar expresamente, dado que inicialmente entendía que la imagen era aquella resultando de la primera predicción.



Figura 22: Ejemplo del resultado de un modelo preentrenado multiclas.

Fuente: pexels.com (Chen Te).

En la Figura 22 podemos ver que ha identificado más de un patrón: semáforos, camiones, coches, personas, paraguas, autobuses y bicicletas. ¿por qué ha mejorado su precisión? Esta imagen es mucho más grande que la anterior [5568x3712 frente a 640x466], lo que le ha permitido mejorar mucho la cantidad de objetos encontrados¹⁰².

Como hemos indicado, el modelo preentrenado YOLO que estamos usando recibe como entrada en su red neuronal convolucional una imagen, lleva a cabo un preprocesamiento y produce, en primera instancia, una lista **en bruto** de los patrones que ha identificado [con su clase y puntuación]. Según el modelo que empleemos, este hará como mínimo y máximo un número constante de identificaciones. Muchas de estas detecciones poseen una puntuación muy baja [inferior a 0.25, aunque este umbral puede cambiarse] o son identificaciones redundantes. Por ejemplo:

truck 0.95406556	bicycle 0.9515624	dog 0.79923487
truck 0.90001684	bicycle 0.91583616	dog 0.99605745
truck 0.85288674	bicycle 0.919426	dog 0.9469049
	bicycle 0.9921388	dog 0.9520811
	bicycle 0.98744863	

Como puedes ver en la Figura 23, YOLO identifica más de una vez algunos patrones visuales¹⁰³.

102 En concreto hace Tiling (submuestreo): Las imágenes muy grandes se dividen en múltiples subimágenes o "tiles" más pequeñas. Cada subimagen se procesa de manera independiente para detectar objetos.

103 Hice una pequeña trampa: antes de mostrar estos datos apliqué un filtro a la lista de cajas, sólo dejé aquellas de más de 0.60 de score, de forma que muchísimas de ellas han desaparecido. De no haberlo hecho tendíamos una imagen con cientos de cajas, en su mayoría falsas identificaciones.

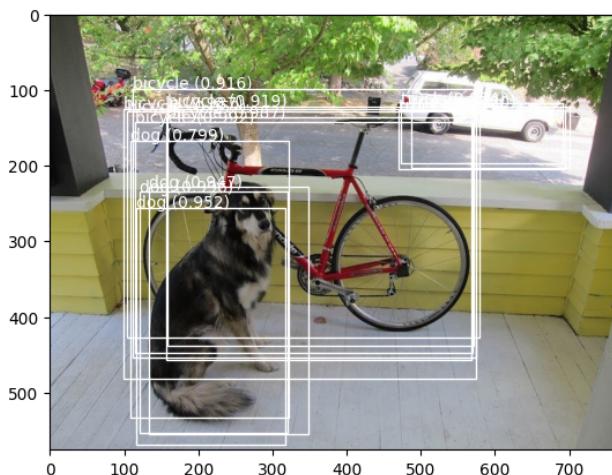


Figura 23: Ejemplo de resultado de la identificación de patrones visuales en una imagen sin filtrar el resultado.

Fuente: COCO dataset

Detectó al perro 4 veces, con scores 0.799, 0.996, 0.947 y 0.952. Para cada identificación rodeó con un rectángulo su localización y lo etiquetó con el identificador y su puntuación. ¿Cómo nos quedamos con las cajas más prometedoras?

Unos de los algoritmos más usado se llama *Non-Maximum Suppression*, NMS, y funciona de esta manera¹⁰⁴:

1. Ordena la lista de cajas de más puntuación a menos puntuación.
2. Recorre la lista, haciendo:
 1. Si estás en la caja k , ve desde la caja $k+1$ hasta el final buscando cajas que tienen una gran superposición con la caja k , típicamente mayor del 50%, y elimina estas cajas.
3. Finalmente tendrás las cajas más prometedoras.

Si lo aplicamos a la Figura 23, podemos ver el resultado en la Figura 24, el cual se parece más a lo que estamos buscamos.

5.4 Métodos para dividir una imagen en regiones o segmentos significativos.

La segmentación de imágenes es un proceso fundamental en la visión artificial que se encarga de dividir una imagen en regiones o segmentos significativos. El objetivo principal de la segmentación es agrupar píxeles o regiones de una imagen que poseen características similares, como color, textura, intensidad o contorno. Esta técnica es fundamental para comprender la estructura y contenido

¹⁰⁴ Estoy dando por hecho que la lista de cajas tiene más información que la caja misma para cada identificación, en concreto como mínimo tendremos el score y que tipo de objeto encontró (normalmente es el índice de una lista).

de una imagen, lo cual es esencial para muchas aplicaciones, como reconocimiento de objetos, seguimiento de objetos, análisis de escenas, entre otros.

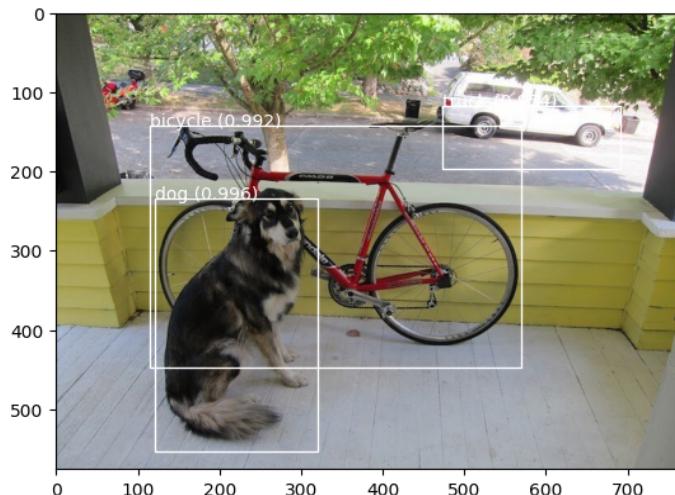


Figura 24: Ejemplo de resultado de la identificación de patrones visuales en una imagen con el resultado filtrado.

Fuente: COCO dataset

Existen varios enfoques para llevar a cabo la segmentación de imágenes. Algunos de los métodos más comunes son:

Segmentación basada en umbrales: Este método es bastante simple y se basa en establecer umbrales de intensidad para clasificar los píxeles de la imagen en diferentes grupos. Los píxeles cuya intensidad se encuentra por encima de un umbral establecido se consideran parte de un grupo, mientras que aquellos por debajo del umbral se clasifican en otro grupo. Esta técnica es útil cuando se trabaja con imágenes binarias o con objetos bien definidos en términos de intensidad.

Segmentación semántica: La segmentación semántica tiene como objetivo asignar una etiqueta semántica a cada píxel de una imagen. Esto implica clasificar cada píxel en una categoría específica, como persona, automóvil, árbol, etc. Para lograrlo, se suelen utilizar técnicas basadas en aprendizaje profundo, como redes neuronales convolucionales (CNN) que han demostrado un gran rendimiento en este tipo de tareas. Estas redes se entrena en grandes conjuntos de datos anotados, donde cada píxel de la imagen se etiqueta manualmente, lo que permite aprender patrones visuales y realizar una segmentación precisa.

Segmentación basada en superpíxeles: Esta técnica divide la imagen en superpíxeles [ver Figura 25], que son regiones de la imagen compactas y coherentes en términos de características visuales. A diferencia de los píxeles individuales, los superpíxeles capturan información contextual y reducen la complejidad del problema de segmentación. Los superpíxeles pueden generar una representación más compacta y significativa de la imagen, lo que facilita tareas posteriores, como el reconocimiento de objetos o el seguimiento de objetos en movimiento.



Figura 25: Ejemplo de imagen mostrando los superpixeles con dos granularidades.

Fuente: "A Video Representation Using Temporal Superpixels", J. Chang, D. Wei, J. W. Fisher III

Como hemos indicado al principio, la segmentación semántica consiste en clasificar cada uno de los pixels de una imagen; el resultado es otra imagen, pero no se trata de la original, es una máscara donde cada pixel indica a qué objeto pertenece el pixel de la imagen original.



Figura 26: Ejemplo de segmentación semántica, original y segmentada.

Fuente: wikimedia (B. Palac)

Este tipo de segmentación es intensiva en tiempo de cómputo y recursos, así que nos centraremos en la segmentación de instancias, resultado de aplicar un proceso de segmentación pero no sobre la imagen en su totalidad, si no sobre detecciones realizadas en una etapa anterior. De esta forma primero se lleva a cabo una identificación de objetos para luego llevar a cabo una segmentación para cada objeto detectado.

Podemos ver como una imagen [Figura 27] es sometida a un proceso de identificación de objetos

[persona, maleta, paraguas, etc.], para posteriormente segmentar cada una de las instancias detectadas y mostrar el resultado en la Figura 28.

05 segmentación.ipynb

Prompt:

Dada una imagen referenciada mediante una url, llevar a cabo la segmentación de la misma usando YOLO v8. Mostrar la imagen original y la imagen segmentada. No almacenar las imágenes en ficheros.

En dicha imagen podemos observar que [1] ha llevado a cabo una identificación de los patrones visuales, etiquetándolos y calculando su puntuación y [2] para cada elemento detectado ha establecido, para cada pixel, una etiqueta semántica, asignándole un tono único¹⁰⁵.



Figura 27: Ejemplo de segmentación semántica, original

Fuente: COCO dataset



Figura 28: Ejemplo de segmentación semántica, segmentada

Fuente: COCO dataset

De nuevo, al igual que aprendimos cuando hicimos la identificación y localización de patrones visuales, el algoritmo no devuelve una imagen, realmente devuelve una lista de resultados [podríamos haber proporcionado más de una imagen], y para cada uno de ellos nos proporciona la información necesaria para visualizar el resultado [como la de la Figura 28].

Nota: *Todos los resultados mostrados en clasificación, identificación de objetos y segmentación, son imágenes de carácter pedagógico. En la realidad los resultados son clases, cajas, identificadores, puntuaciones y máscaras; información más estructurada y adecuada para ser procesada en etapas posteriores. Por ejemplo para la etapa de toma de decisiones de un agente (inteligente o no).*

El dataset COCO, con más de 330.000 imágenes, al que me he referido en varias ocasiones, nos proporciona un explorador (<https://cocodataset.org/#explore>) en el que podemos ver más ejemplos de segmentación de instancias.

¹⁰⁵ Observar que los paraguas tienen colores diferentes.

5.5 Manos a la obra: entrenando modelos

Hasta ahora hemos mostrado como usar modelos preentrenados, generando código por medio de *prompts*. Vamos ahora a avanzar y mostrar cómo podemos entrenar nuestros propios modelos.

5.5.1 Clasificador de dígitos

Vamos a crear un clasificador multiclas, uno que a partir de una imagen genere un vector de 10 scores con la puntuación que considera para cada dígito [de 0 a 9], de tal manera que el score más alto sea la predicción del dígito que considera que mejor representa a la imagen.

El *dataset* que vamos a usar es uno llamado MNIST *digits*, y es un clásico en el contexto del *machine learning* [ML].

Mi IA favorita me explica: *El dataset MNIST (Modified National Institute of Standards and Technology) digits fue creado por Yann LeCun, Corinna Cortes y Christopher J.C. Burges en 1998. Su propósito principal era facilitar la investigación y el desarrollo de algoritmos de aprendizaje automático y reconocimiento de patrones, proporcionando un conjunto de datos estandarizado y procesado de dígitos manuscritos. El dataset, derivado de una versión más grande y compleja del NIST, contiene 70,000 imágenes de dígitos (60,000 para entrenamiento y 10,000 para pruebas) y ha sido ampliamente utilizado como referencia y punto de partida para la comparación de técnicas en la comunidad de aprendizaje automático.*

Para crear el *dataset* se tomaron imágenes de dígitos manuscritos de dos conjuntos de datos del NIST: uno de estudiantes de secundaria y otro de empleados de la Oficina del Censo de EE. UU. Esto permitió una mayor diversidad en el estilo de escritura.

Trabajemos con nuestra IA generativa para ir creando, celda a celda, nuestro *notebook*.

05 clasificación de imágenes.ipynb

Prompt:

Descarga el dataset MNIST digits y cárgalo en memoria en un dataframe. Muestra un ejemplo de 10 imágenes y su correspondiente etiqueta, todo ello en una fila.

El código generado muestra la siguiente salida:



El *dataset* contiene 70.000 imágenes [cada una de ellas con su etiqueta] de 28x28 pixels y un sólo canal de 1 byte, codificando cada pixel tonos de grises de 0 [negro] a 255 [blanco].

La siguiente acción a realizar sería separar [*split*] el *dataset* en dos conjuntos, el *trainset* y el *testset*, pero esta acción ya la ha hecho en la primera celda:

```
5
6 # Descarga el dataset MNIST digits
7 (x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
8
```

Ahora tenemos los dos conjuntos separados: el *trainset* [*x_train*, *y_train*] y el *testset* [*x_test*, *y_test*]. El *dataframe* llamado *x_train* contiene las imágenes [una por fila] y el llamado *y_train* contiene las etiquetas.

Un dataframe es una estructura de datos bidimensional y mutable, similar a una tabla en una base de datos o una hoja de cálculo, que permite almacenar y manipular datos de manera eficiente en Python. Está compuesto por filas y columnas, donde cada columna puede contener datos de un tipo diferente (números, cadenas, etc.). Los dataframes son parte de la biblioteca pandas y proporcionan funciones y métodos para realizar operaciones de limpieza, agregación, filtrado y transformación de datos, facilitando el análisis y manejo de grandes volúmenes de información.

La siguiente acción es seleccionar un algoritmo de ML y usarlo para entrenar un modelo. Vamos a seleccionar primero un algoritmo no basado en redes neuronales, **xgboost**.

05 clasificación de imágenes.ipynb

Prompt:

Usando xgboost entrena un modelo usando el conjunto de entrenamiento. Usa “cross validation” para mostrar una evaluación sin usar el conjunto de testeo.

La ejecución de esta celda generada no es rápida, ha tardado 42 minutos. De hecho tarda 5 veces más de lo normal, ya que le hemos pedido expresamente que use una técnica llamada *cross validation*, validación cruzada:

La validación cruzada es una técnica de evaluación de modelos en aprendizaje automático que se utiliza para asegurar que el modelo generalice bien en datos no vistos previamente. Consiste en dividir el conjunto de datos en varios subconjuntos, o "folds" (carpetas). En cada iteración, el modelo se entrena en todos los folds menos uno y se valida en el fold restante. Este proceso se repite hasta que cada fold haya sido utilizado como conjunto de validación una vez. Los resultados de cada iteración se promedian para proporcionar una estimación del rendimiento del modelo.

Esta técnica no entrena un modelo. Lo que lleva a cabo es la prueba un algoritmo de ML, entre-

nando n modelos de ese tipo y comprobando qué tal resultado tuvieron promediando su precisión¹⁰⁶. Este proceso se repite para otro u otros algoritmos de ML, quedándonos con el que ha arrojado el mejor resultado [la mejor precisión]. Este, el mejor de todos, es el que finalmente entrenaremos con el *trainset* y finalmente evaluaremos con el *testset*.

05 clasificación de imágenes.ipynb

Prompt:

Usando una red neuronal convolucional entrena un modelo usando el conjunto de entrenamiento. Si así lo consideras, usa “cross validation” para mostrar una evaluación sin usar el conjunto de testeo.

La precisión alcanzada para cada uno de los algoritmos fue:

Algoritmo	Tiempo	Precisión
XGBost	42 minutos	0.9752
Red neuronal convolucional	25 minutos	0.9936

El algoritmo que ha obtenido una mayor precisión para estos datos, ha sido la red neuronal. Esta red neuronal posee más de 100.000 parámetros.

El siguiente paso entonces, es entrenar¹⁰⁷ un modelo durante 10 epochs¹⁰⁸ con el conjunto de entrenamiento, mostrando como va evolucionando el resultado de la función de pérdida a lo largo del entrenamiento.

La función de pérdida es un componente crucial en el entrenamiento de una red neuronal, ya que cuantifica la diferencia entre las predicciones del modelo y los valores reales esperados. Durante el proceso de entrenamiento, la red neuronal ajusta sus pesos con el objetivo de minimizar esta función de pérdida.

El *prompt* es:

05 clasificación de imágenes.ipynb

Prompt:

Usando la red neuronal, entrena un modelo usando el conjunto de entrenamiento. Presenta como resultado la evolución de la función de pérdida a través de 10 epochs. No hagas una evaluación del modelo.

¹⁰⁶ La precisión para cada fold se calcula como el número de predicciones correctas dividido por el número total de instancias en el fold de validación.

¹⁰⁷ Más adelante veremos como se entrena con detalle una red neuronal. Por ahora vamos poco a poco mostrando detalles a medida que lo necesitamos.

¹⁰⁸ El número de epochs cambiará según el problema, nosotros ya sabíamos a priori que con 10 epochs llegaba y sobraba para este ejercicio.

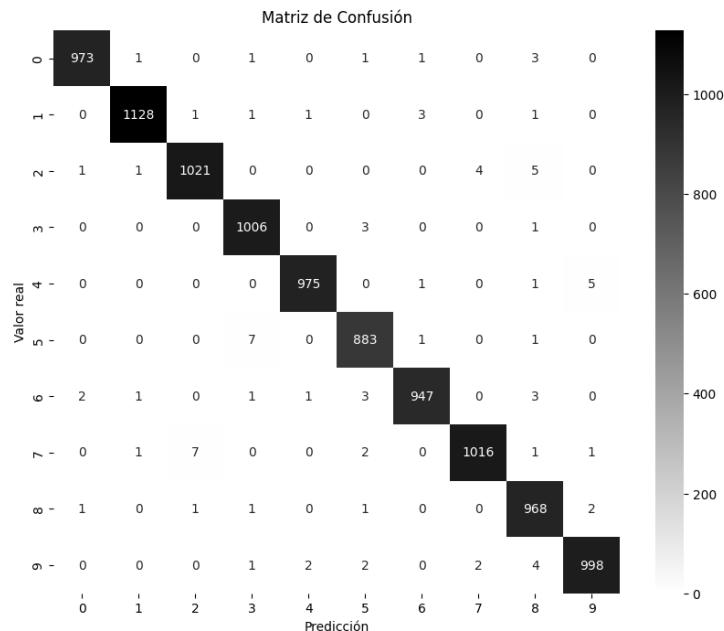
Y finalmente:

05 clasificación de imágenes.ipynb

Prompt:

Evalúa el modelo usando el testset. Muestra la matriz de confusión. Usa un colormap de tonos de grises.

Al finalizar tenemos un modelo [que podemos almacenar¹⁰⁹ en un fichero o compartirlo en la web o redes sociales] y una evaluación [que tal se comporta] en forma de matriz de confusión:



Una **matriz de confusión** es una herramienta utilizada para evaluar el rendimiento de un modelo de clasificación, mostrando no solo la cantidad de predicciones correctas, sino también cómo se distribuyen los errores del modelo.

Para nuestro modelo, que predice las 10 categorías del conjunto de datos MNIST *digits*, la matriz de confusión es una tabla de 10x10 donde cada fila representa las instancias reales de una clase y cada columna representa las instancias predichas de las mismas clases. Por ejemplo, si el modelo predice correctamente el dígito "3" en 1.006 de 10.000 casos, y confunde los otros 12 casos con otros dígitos, estos errores se registrarán en las columnas correspondientes de la fila etiquetada como "3".

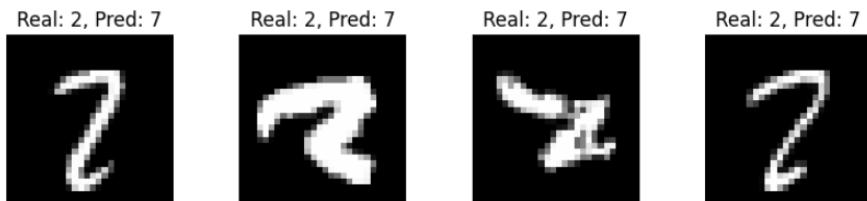
El valor de la diagonal principal de la matriz representa el número de predicciones correctas para cada clase, mientras que los valores fuera de la diagonal indican errores de clasificación.

109 Ocupa poco, 452 KB.

Observemos que el dígito “3” se confunde varias veces con el dígito “5”:



Al igual que el dígito “2” que se confunde también con el dígito “7”:



El modelo parece funcionar bastante bien, veamos la matriz de confusión usando un algoritmo menos adecuado¹¹⁰ para estos datos:

Matriz de Confusión (Árbol de Decisión)										
	0	1	2	3	4	5	6	7	8	9
0	914	0	10	9	4	12	12	3	9	7
1	2	1091	9	6	1	5	9	3	8	1
2	10	10	883	31	14	10	15	27	20	12
3	7	5	32	860	9	45	5	6	25	16
4	4	4	9	8	855	10	16	13	18	45
5	12	5	5	46	9	745	22	7	25	16
6	16	3	15	7	12	21	854	1	20	9
7	2	16	26	14	5	6	4	921	9	25
8	6	7	33	31	20	25	15	7	795	35
9	16	4	7	19	42	13	4	19	23	862
	0	1	2	3	4	5	6	7	8	9
	Predicción									

Para esta última matriz de confusión, hemos usando un algoritmo llamado *Decision Tree* el cual arroja una precisión media de 0.88. Se ve la diferencia.

110 “Menos adecuado” no significa necesariamente peor en general, sólo tiene menor precisión para este tipo de datos.

5.5.2 Entrenando un modelo para identificar objetos

En el apartado anterior entrenamos modelos desde cero, partiendo de un *dataset* y creando modelos que fueron entrenados únicamente con dicho *dataset*. Esta estrategia es normal en el ML clásico, pero si hablamos de *Deep Learning* [aprendizaje profundo] tenemos la posibilidad de re-entrenar modelos que ya habían sido entrenados con anterioridad.

El deep learning (DL) es un subtipo del aprendizaje automático (ML) basado en el uso de redes neuronales artificiales con múltiples capas con el objetivo de reconocer patrones complejos en grandes volúmenes de datos. Estas redes profundas son capaces de aprender representaciones de los datos, donde cada capa sucesiva captura características más abstractas y de alto nivel a partir de las capas anteriores.

Gracias al método de entrenamiento de una red neuronal, basado en la optimización de sus pesos, optimización que nunca es perfecta¹¹¹, siempre es posible re-entrenar un modelo usando nuevos datos. Además, es aconsejable usar un modelo preentrenado, dado que convergerá¹¹² más rápidamente que otro que debe ser entrenado desde cero¹¹³.

La primera etapa para entrenar un modelo que tiene como objetivo identificar objetos es **recopilar los datos**: el *dataset*. Este proceso es, en general, más pesado que el propio entrenamiento, así que en este primer entrenamiento lo haremos con un *dataset* ya creado. Este *dataset* consiste en imágenes creadas de forma sintética, por ejemplo:

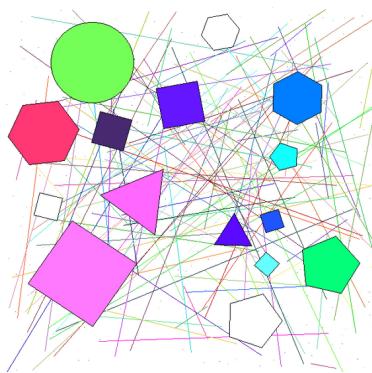


Figura 29: Imagen ejemplo del dataset gshapes.

Lo que vamos a identificar son las figuras geométricas: círculos, triángulos, cuadrados, pentágonos y hexágonos. Las líneas de fondo son ruido y las figuras geométricas son de diferentes colores, con el fin de aumentar la variabilidad.

¹¹¹ El aprendizaje busca la generalización de patrones (recordar el punto 1.3); la generalización por definición descarta los detalles, por lo que siempre posee una cantidad controlada de error.

¹¹² Un proceso de optimización se dice que converge si se acerca a una solución sub-óptima.

¹¹³ Las redes neuronales artificiales, al crearse, inicializamos sus pesos de forma aleatoria.

El dataset gshapes está formado por 1.000 imágenes diferentes, semejantes a la vista en la Figura 29. Además, cada imagen tiene asociado un fichero con el mismo nombre pero con extensión .txt; en donde podemos encontrar las identificaciones que su pareja .png contiene (ver Figura 30).

05 identificación de objetos.ipynb

Prompt:

Dado el fichero .zip referenciado con la url <https://raw.github.com/FranPuentes/iTI2025/main/data/gshapes.zip>, descárgalo, ábrelo y escoge aleatoriamente un fichero .png y su pareja .txt que tienen el mismo nombre en el subdirectorio "train". Dibuja la imagen y la misma con las identificaciones [formato yolo] que contiene el fichero .txt.

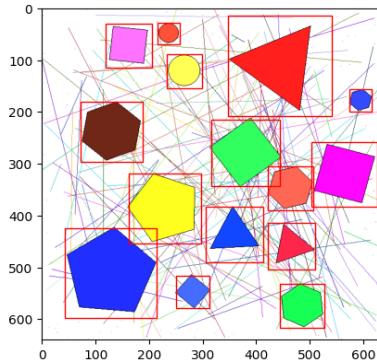


Figura 30: Ejemplo de imagen del dataset gshapes, con sus etiquetas dibujadas.

Más adelante veremos como crear *datasets* y empaquetarlos en un fichero .zip.

05 identificación de objetos.ipynb

Prompt:

Instala con pip la librería Ultralytics y entrena un modelo YOLO v8 usando el contenido del .zip indicado. Al finalizar muestra una matriz de confusión con su evaluación. No hagas llamadas a la línea de comandos.

En este caso la red neuronal convolucional posee más de 3.000.000 de parámetros.

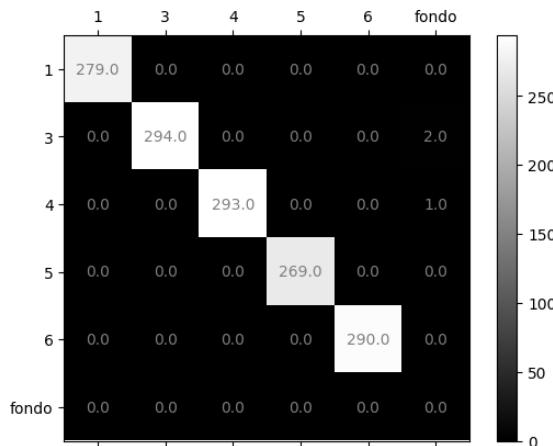
La matriz de confusión arroja el siguiente resultado:

```

1
2 [      279,          0,          0,          0,          0,          0,          0],
3 [      0,          294,          0,          0,          0,          0,          2],
4 [      0,          0,          293,          0,          0,          0,          1],
5 [      0,          0,          0,          269,          0,          0,          0],
6 [      0,          0,          0,          0,          290,          0,          0],
7 [      0,          0,          0,          0,          0,          0,          0]
8

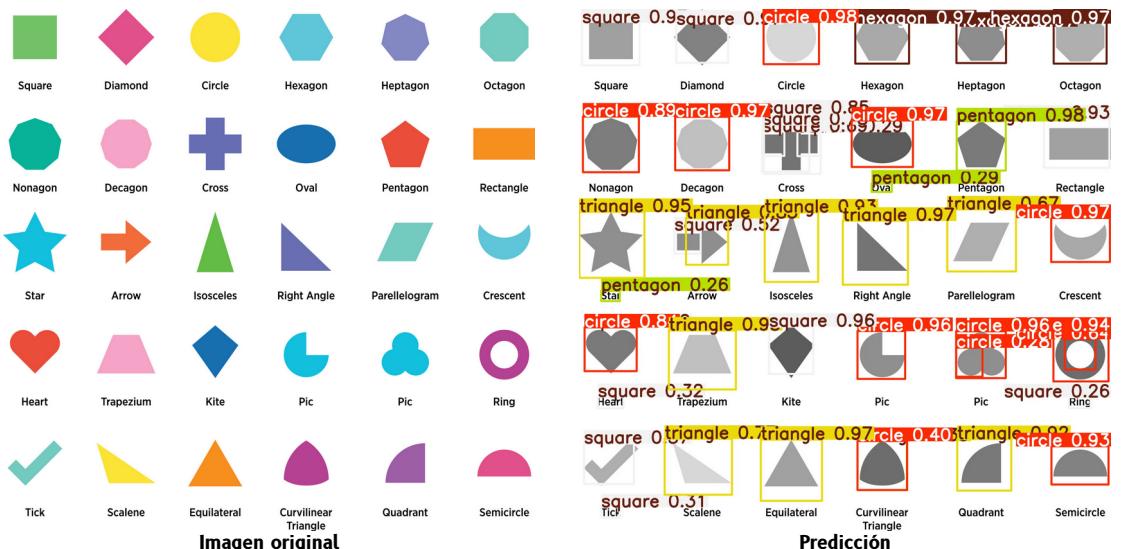
```

Que, presentado como una imagen, se ve de la siguiente manera:



El resultado es casi perfecto. Vemos que en 3 casos [columna “fondo”] cometió errores. En concreto en dos casos confundió el fondo como un triángulo, y en una ocasión como un cuadrado. Es obvio, la líneas del fondo que añadimos como ruido, en tres ocasiones formaron de forma fortuita dos triángulos y un cuadrado, que fueron malinterpretados por el modelo.

Pero la potencia de estos modelos reside en la capacidad de generalización, en este caso la capacidad de aplicar la predicción a otro tipo de imágenes, diferentes a las de la Figura 29:



Podemos observar que ha hecho identificaciones muy interesantes, pero lo más llamativo es como ha generalizado y a identificado los heptágonos y octágonos como hexágonos, ya que sólo fue entrenado para detectar figuras de como máximo 6 lados. A partir del nonágono [9 lados] generaliza y los identifica como círculos.

RETOS DEL CAPÍTULO 5

1. ¿Cuáles son los objetivos principales de la identificación y segmentación de patrones en el contexto de la visión artificial?
2. ¿Cuáles son algunas aplicaciones prácticas de la identificación de patrones en la vida cotidiana?
3. Dado que va a aparecer mas de una vez, preguntale a tu IA favorita, en el contexto del aprendizaje automático, que es una característica (*feature*) y en qué consiste el proceso de “*extracción de características*”.
4. Pídele a tu IA favorita, que te muestre ejemplos que diferencien bien entre identificación y segmentación; que en el contexto que nos ocupa, muestre la relación entre identificar un patrón en una imagen y clasificar dicho patrón. Pregúntale la diferencia entre la segmentación semántica y la segmentación de instancias.
5. Busca información sobre el producto de Meta (antigua Facebook) SAM (*Segment Anything*). Explora el notebook “*Cap. 7 – segmentación.ipynb*” y prueba con diferentes imágenes.
6. En la §4.7 hablamos sobre la operación de convolución. Pídele a tu IA favorita, por medio de un diálogo basado en el descubrimiento incremental, que te ayude a entender como funciona una red neuronal convolucional basándose en lo que ya sabes. Cuidado: dile que te explique como si no subieras nada.
7. Busca en la web sitios que te permitan eliminar el ruido de una imagen, restaurarla o mejorar su calidad. Prueba editando imágenes o cambiando su resolución por una menor. Cuidado: es posible que los sitios que usen pasen a tener derechos sobre la imagen que uses, así que no utilices imágenes con contenido sensible para la privacidad y los derechos de las personas.
8. La §5.2 es un adelanto, que más adelante detallaremos. Así que atrévete a preguntarle a tu IA favorita sobre la multiclasicación en el contexto del aprendizaje automático, interrógala de forma recursiva sin dejar hueco a la duda. Pídele especialmente que te explique qué es una matriz de confusión.
9. Pídele a tu agente conversacional favorito que te cuente la historia de YOLO y sus versiones. Pídele que te diga en que se han empleado los modelos de YOLO 3 y sus versiones inferiores.
10. En breve vamos ver el procedimiento para entrenar y usar un modelo de aprendizaje automático, de hecho ya los estamos usando en este capítulo; y hay “cosas” que van apareciendo. Pídele a tu IA favorita que te explique que es el “*fit*” y “*predict*” de un modelo de aprendizaje automático creado con *scikit-learn*. ¡Y qué es *scikit-learn* por supuesto!

Capítulo 6

PERCEPCIÓN DE ALTO NIVEL

Cómo estábamos diciendo, la visión artificial es un campo multidisciplinario que busca dotar a las máquinas de la capacidad de percibir e interpretar imágenes y videos de manera similar a los seres humanos. Hasta ahora hemos visto técnicas mediante las cuales conseguimos que nuestro **agente** pueda llevar a cabo una percepción de bajo nivel. La percepción en visión artificial abarca diversas tareas, como la detección de objetos, la segmentación de imágenes, el reconocimiento de patrones, pero también la **estimación de pose**, el **seguimiento de objetos** y la **estimación de la profundidad**, entre otras. Estas tareas de más alto nivel reflejan los desafíos involucrados en la interpretación y comprensión de los datos visuales, y su desarrollo contribuye a mejorar la capacidad de las máquinas para analizar y comprender el entorno visual de manera cada vez más precisa y sofisticada.

La percepción de alto nivel se refiere a la capacidad de un sistema para interpretar y comprender información visual en un contexto más abstracto y semántico, y no solo a nivel de píxeles o características básicas.

Veamos con más detenimiento algunas de estas tecnologías.

6.1 Estimación de pose

La estimación de pose se refiere a la capacidad de determinar la posición y orientación de un objeto¹¹⁴ en el espacio tridimensional a partir de datos visuales. Esta tarea es esencial en aplicaciones como la realidad aumentada, la robótica y los sistemas de navegación.

Existen diferentes enfoques y técnicas utilizadas para la estimación de pose, dependiendo de los datos y las restricciones del sistema. De entre estos destacan dos: [1] el uso de **marcadores** [o *key-points*] los cuales son patrones o códigos visuales específicos que se colocan en el objeto y que son fácilmente detectables por la cámara. Estos marcadores proporcionan información conocida y explícita sobre la pose del objeto. Al detectar y reconocer los marcadores en una imagen, se puede estimar la pose del objeto con precisión.

¹¹⁴ Aunque siempre me referiré a su aplicación sobre la pose humana, esta tecnología puede ser empleada para determinar poses de brazos robóticos o cualquier elemento articulado.

En la Figura 31 podemos observar una aplicación de esta estrategia, [a] se colocan los marcadores visuales al sujeto, [b] estos son capturados por la cámara, [c] se reconstruyen las relaciones¹¹⁵ entre los marcadores capturados y [d] a partir de la estructura anterior de *deduce* la pose del sujeto original (líneas azules).

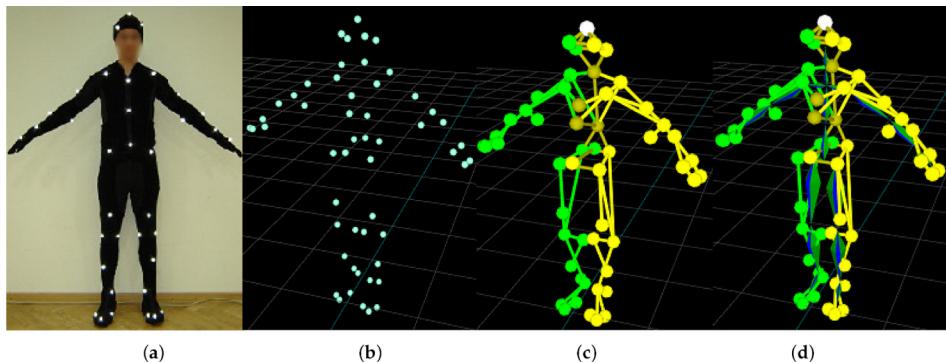


Figura 31: Estimación de la pose por medio de marcadores visuales

Fuente: "Gap Reconstruction in Optical Motion Capture Sequences Using Neural Networks"; Skurowski, P.; Pawlyta, M.; Sensors, 2021

El segundo enfoque [2] es el uso del **aprendizaje profundo**: se han desarrollado enfoques basados en redes neuronales convolucionales [CNN] para la estimación de pose [ver la Figura 57]. Estos enfoques aprovechan el poder del aprendizaje automático para aprender patrones visuales y estimar la pose de forma directa a partir de imágenes. Las redes neuronales pueden ser entrenadas con grandes conjuntos de datos anotados, lo que les permite aprender a estimar la pose con alta precisión y generalización.



Figura 32: Ejemplo de uso de redes neuronales artificiales aplicadas a la estimación de la pose en una fotografía.

Fuente: PhysicsWorld - Einstein en Oxford (1933)

115 Si el marcador que se corresponde con punta del pie está tocando el cuello, nuestro sujeto tiene un problema.

La estimación de pose, usando una sola imagen, es un desafío importante en la visión artificial, ya que implica combinar información visual, geometría y conocimiento sobre la cámara para determinar la posición y orientación precisa de un objeto en el espacio tridimensional.

Vamos a generar el código para llevar la estimación de pose de una imagen.

06 estimación de pose.ipynb

Prompt:

Dada una imagen referenciada mediante una url, muestra la imagen y lleva a cabo una estimación de pose usando yolo v8.

Observamos en la Figura 33 cómo el proceso de estimación de pose detecta patrones visuales y los relaciona, infiriendo una información de carácter global; pero también diferenciando dos personas en la imagen y tratándolas individualmente.



Figura 33: Resultado de una estimación de pose usando redes neuronales artificiales sobre una imagen con dos personas

Fuente: stocksnap.io, Free stock image

La porción del código que carga el modelo [16] y lleva a cabo la inferencia [17] es:

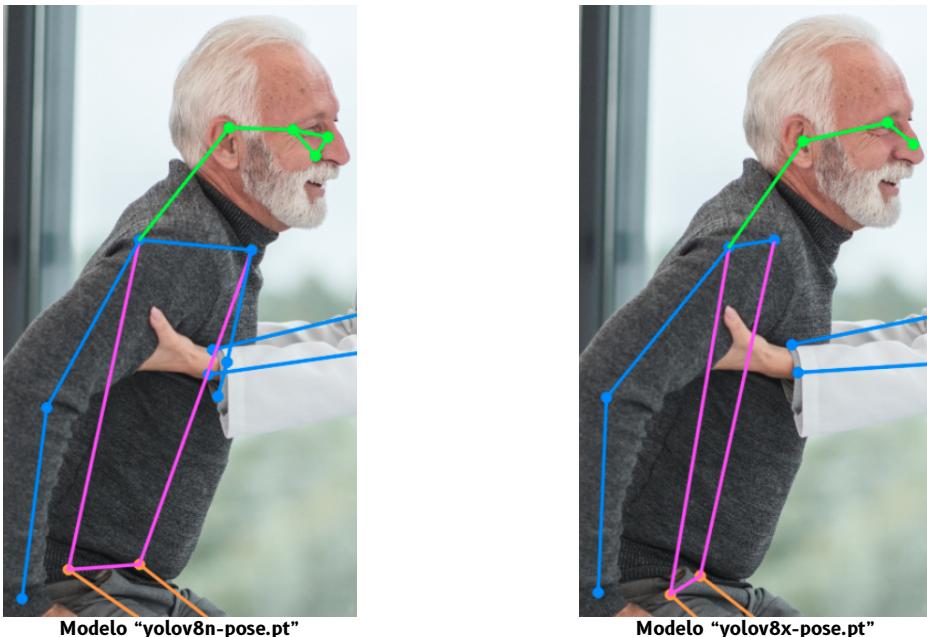
```
15
16 model = YOLO('yolov8n-pose.pt')
17 results = model(image)
18
```

Esta librería, al entregarnos en su resultado los parámetros de la pose, también nos entrega la identificación y localización de los elementos en la imagen.

Nota (de nuevo): *Todos estos algoritmos que estamos viendo no devuelven imágenes con las detecciones superpuestas. Estas imágenes las hacemos nosotros por motivos pedagógicos. Realmente nos devuelven datos “crudos”, por ejemplo, en el caso de la estimación de la pose, una lista de marcadores (keypoints) conteniendo cada uno*

de ellos la coordenada (x,y) , además de la identificación del marcador (p.ej. hombro derecho).

Como podemos ver en el código anterior, estamos usando un modelo llamado “yolov8n-*pose.pt*”, pero hay modelos mejores¹¹⁶; en concreto esa “n” nos indica que es el modelo más pequeño [nano], vamos a usar el mejor “x” y veamos las diferencias.



Como podemos observar la calidad de la estimación mejora significativamente con el modelo de la derecha (“x”) respecto al de la izquierda (“n”): El modelo “x” infiere mejor los *keypoints* de la cara, de los hombros y de la cadera, entregándonos una estimación mejorada global de la pose.

El desarrollo de algoritmos y técnicas precisas de estimación de pose es crucial para muchas aplicaciones prácticas, como la navegación autónoma de robots, la colocación precisa de objetos virtuales en realidad aumentada y la interacción con interfaces de usuario basadas en visión artificial.

6.2 Seguimiento de objetos

El seguimiento de objetos se ocupa de rastrear y mantener una base de datos de objetos únicos identificados, a través de una secuencia de imágenes [o un flujo¹¹⁷ de video]. Esta tarea es particularmente útil en aplicaciones como el seguimiento de personas, el seguimiento de vehículos, la vi-

¹¹⁶ De la misma librería o de otras librerías.

¹¹⁷ Probablemente te preguntes la diferencia entre ambos conceptos. En lo que ahora nos incumbe (porque hay más diferencias), si hablamos de secuencia de imágenes significa que tenemos todas ellas a nuestra disposición, mientras que un flujo de video sólo disponemos de los “frames” hasta el instante en que nos encontramos; dicho de otro modo, no sabemos que “frames” está por llegar.

gilancia de video y la realidad virtual. Como tarea de percepción, implica que debe localizar e identificar los objetos en una secuencia de imágenes [en cada una de ellas] y reconocer qué objetos estaban en los fotogramas anteriores y cuales no. Actualizando una base de datos de objetos identificados por su *id*¹¹⁸.

El seguimiento de objetos implica enfrentar varios desafíos, ya que los objetos pueden cambiar de apariencia, forma y escala, y pueden estar sujetos a cambios de iluminación y occlusiones parciales. Por lo tanto, los algoritmos de seguimiento deben ser capaces de adaptarse a estas variaciones y mantener un seguimiento robusto y preciso.

Existen diferentes enfoques y técnicas utilizados en el seguimiento de objetos. Algunos de los enfoques¹¹⁹ más comunes incluyen:

- **Seguimiento basado en características:** Este enfoque implica detectar y rastrear características distintivas en el objeto de interés, como esquinas o puntos clave¹²⁰. Estas características se extraen y se emparejan a lo largo de la secuencia de imágenes para determinar la trayectoria del objeto.
- **Seguimiento basado en modelos:** Este enfoque implica construir modelos del objeto de interés y actualizarlos a medida que el objeto se mueve en la secuencia de imágenes. Los modelos pueden incluir información geométrica, información de apariencia o incluso información del contexto. Estos modelos se utilizan para hacer coincidir y alinear el objeto en cada cuadro, asegurando así un seguimiento robusto.
- **Seguimiento basado en filtrado:** Este enfoque utiliza técnicas de filtrado para estimar y predecir la posición del objeto en cada cuadro de la secuencia. Estos algoritmos combinan la información visual con la información de movimiento para realizar una estimación precisa de la trayectoria del objeto a lo largo del tiempo.
- **Seguimiento basado en aprendizaje profundo:** El aprendizaje profundo ha revolucionado muchos aspectos de la visión artificial, incluido el seguimiento de objetos. Los enfoques basados en aprendizaje profundo utilizan redes neuronales convolucionales [CNN] para aprender a rastrear objetos en imágenes.

Es difícil mostrar un ejemplo basado en el tiempo en este medio de dos dimensiones; los libros en papel es lo que tienen. Pero podemos poner un ejemplo: es posible crear una aplicación que, usando técnicas de *object tracking*, contase el número de vehículos que circulan por una carretera. Como puedes observar, una cámara, cuyas imágenes fueran procesadas con el objeto de identificar y hacer el seguimiento de los vehículos, podría contarlos y, según el tipo del vehículo, calcular sus velocidades, avisando si aparece un vehículo en sentido contrario, dado que puede discernir si estos se acercan o se van respecto a la cámara.

118 En la ingeniería informática, se usa un *id* (identificador) para distinguir unos objetos de otros. Típicamente es un número diferente para cada objeto. El concepto te debe sonar, es el “DNI” del objeto.

119 Estos enfoques no son excluyentes, puede combinarse.

120 Por ejemplo, podemos crear un detector de ojos, narices y boca; y a partir de ellos, conociendo que la relación entre los tres/cuatro es conocida, deducir donde está una cara en cada frame.



Video 3: Video original



Video 4: Video con tracking



Video 5: Video con los centros dibujados

De hecho lo tenemos dentro del conjunto de *notebooks* que tienes disponibles en el repositorio [07 Object tracking.ipynb]. Este *notebook* se usó para, a partir del Vídeo 3 generar el Vídeo 4, el cual muestra cómo a las identificaciones se les ha sumado un *id* de objeto, que lo identifica de forma única a lo largo del vídeo. El Vídeo 5 sustituye las identificaciones por un efecto práctico, en donde hemos añadido un círculo a cada identificador en el centro geométrico del *box* de cada objeto.

En el Vídeo 6 puedes ver un ejemplo de YOLO v3 en tiempo real.



Video 6: Real Time Object Detection using YOLOv3 on Road Traffic on Nvidia RTX 2060

Fuente: Life2Coding

Pero ¿cómo es posible que identifique dos objetos diferentes en dos frames diferentes como el mismo objeto? Vamos a explicar una de las técnicas más simples para hacer el seguimiento.

Partimos de algoritmo de identificación de la Figura 24, en ella puedes ver que tenemos las cajas que localizan los objetos identificados. Ahora, ya que tenemos un vídeo, tenemos una secuencia de imágenes [*frames*], el *frame*_{*k*} es la continuación del *frame*_{*k-1*} y el anterior al *frame*_{*k+1*}. Ahora, para cada *frame*_{*k*} (excepto el primero) hacemos:

1. Calcular para cada caja su punto central [en 2D es trivial si conocemos una esquina y su ancho y alto].
2. Para cada uno de los centros calculados en el punto anterior buscamos el centro más cercano en la lista de puntos del *frame*_{*k-1*}. Haremos que ambos puntos posean el mismo *id*, esto es, son el mismo objeto detectado en *frames* diferentes.
3. Guardamos la lista de puntos del *frame*_{*k*}, para ser usadas en el *frame*_{*k+1*}.

Quizá una imagen lo ilustre mejor:



Figura 34: Otra forma de representar los tracking.

Hemos calculado los *trackings* de los objetos detectados en un vídeo cenital de personas andando, mostrándolos en forma de círculos azules [*frames* anteriores al actual] y círculos rojos [frame actual].

Como podemos ver cada objeto detectado lo fue de forma consistente entre *frames*. Así vemos como para cada figura posee un círculo rojo y una linea de seguimiento en azul que indica donde estaba en *frames* anteriores. Ya tienes más información: además de la identificación y donde se encuentra, sabes cual es su trayectoria y ... su velocidad¹²¹.

Este método funciona muy bien para casos muy específicos, como el del ejemplo, en donde los objetos apenas interactúan entre sí. Pero falla si, por ejemplo, en

el $frame_k$ un objeto no es identificado, pero si en los $frame_{k-1}$ y $frame_{k+1}$, ya que estamos teniendo en consideración sólo los centros del *frame* anterior. Para mejorarlo se pueden tener en cuenta no solo el *frame* anterior si varios, esto es, la trayectoria. También podemos deducir fácilmente donde estará en futuros *frames* y así detectarlo incluso si desaparece durante unos pocos *frames*.

Otro enfoque consiste en usar algoritmos que, en vez de usar los centros de las cajas, usen la caja entera, por ejemplo, usando la métrica IOU [*intersection over union*] y el algoritmo Húngaro [también conocido como el algoritmo de Kuhn-Munkres]:

Mi IA favorita dice: En el contexto del seguimiento de objetos, el algoritmo Hungarian se utiliza para resolver el problema de asignar detecciones en el frame actual a seguimientos (tracks) existentes de tal manera que la suma total de las distancias (o costos) entre las detecciones y sus seguimientos asignados sea mínima. [...]. El costo de asignar una detección a un seguimiento es generalmente determinado por alguna métrica de distancia, como la distancia IOU.

En inteligencia artificial existen muchos problemas que se resuelven, o son susceptibles de resolverse, mediante algoritmos o **técnicas de optimización**. Siendo muy reduccionista, las técnicas de optimización aplicadas a la resolución de un problema, consisten en encontrar una función¹²² que modele el problema y aplicar un algoritmo de optimización que busque la mejor solución posible dentro de un conjunto dado, siguiendo una reglas específicas.

Para aplicar una técnica de optimización, necesitamos una métrica o distancia. La distancia¹²³ IOU

121 Al menos en pixels por segundo. Tienes que hacer una calibración para poder calcular las distancias en el sistema decimal. Quizá también tener en cuenta la distancia.

122 Llamada función objetivo o de coste.

123 Un concepto muy parecido vamos a tener en el procesamiento del lenguaje natural para tratar la similitud de dos textos.

entre dos rectángulos A y B es [ver Figura 35]:

$$IOU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Con $|A|$ denotamos el área de A y con $|A \cap B|$ el área resultante de intersecciar ambos rectángulos. Equivalentemente $|A \cup B|$ representa el área resultante de la unión de ambos rectángulos.

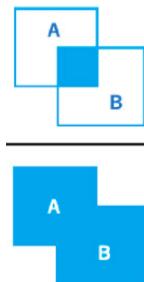


Figura 35: IOU sobre rectángulos

Esta medida arroja valores entre 0 y 1. Si ambas cajas no solapan su IOU será cero, si ambos rectángulos son iguales su IOU será 1. Cualquier otro solapamiento dará como resultado un IOU mayor que cero o menor que 1.

Con esta medida de distancia, el algoritmo húngaro se reduce a un **problema de optimización**: encontrar para cada *frame* y los anteriores, una identificación que minimicen los IOU de cada seguimiento. El algoritmo completo, en *psdo-código*, sería:

1. Inicialización:
 - a) Procesa la primera imagen con el algoritmo de detección para obtener un conjunto de cajas.
 - b) Asigna un *id* único a cada detección en esta primera imagen.
 - c) Almacena estas detecciones iniciales en tu base de datos con sus respectivos *id*.
2. Procesamiento de la secuencia:
 - a) Para cada nueva imagen en la secuencia:
 - Detección: Usa el algoritmo de detección para obtener un conjunto de cajas.
 - Asociación: Utiliza un algoritmo de seguimiento [por ejemplo, el algoritmo Hungarian con distancia IOU] para asociar las cajas detectadas con los objetos existentes en la base de datos. Es decir, determina qué cajas en la nueva imagen coinciden con las cajas de la imagen anterior.
 - Actualización de la base de datos:
 - Si una caja en la imagen actual coincide con una caja en la base de datos [basada en el paso de asociación], actualiza la ubicación/atributos de ese objeto en la base de datos.

- Si una caja en la imagen actual no tiene coincidencia, asigne un nuevo *id* a esa caja y añádela a la base de datos como un nuevo objeto.
 - Si un objeto de la base de datos no se encuentra en varias imágenes consecutivas, puedes decidir eliminarlo o marcarlo como "no detectado" durante cierto tiempo.
3. Gestión de identidades:
- a] A medida que procesas las imágenes, es posible que enfrentes desafíos como oclusiones, donde un objeto se oculta detrás de otro. En tales casos, los algoritmos de seguimiento y las métricas de asociación son vitales para mantener la identidad correcta de los objetos.
 - b] Mantén un registro de cuántas veces consecutivas un objeto no ha sido detectado. Si un objeto no se detecta después de un cierto número de frames, puedes optar por eliminarlo de la base de datos o marcarlo como inactivo.
4. Almacenamiento de características adicionales [opcional]: Además de las coordenadas de la caja, es posible que desees almacenar características adicionales para cada objeto, como el color predominante, un histograma de color, o incluso características más avanzadas extraídas con otras técnicas. Estas características pueden ayudar en la re-identificación si un objeto desaparece y reaparece después de un tiempo.

6.3 Estimación de la profundidad

Levanta la cabeza y fíjate en tu entorno. ¿A qué distancia está cada uno de los objetos que ves? Probablemente no puedas darme una distancia numérica, al menos no muy precisa¹²⁴. Pero seguro que puedes enumerar los objetos que ves y ordenarlos a de mayor a menor distancia respecto a tu posición tú.

La percepción de la profundidad es una de las actividades que realiza tu cerebro sin que te des cuenta e implica varias capacidades y niveles de percepción:

1. Gracias a que tenemos dos ojos orientados hacia la misma zona de visión¹²⁵ [ver Figura 36], nuestra visión es estereoscópica. Aquellos mamíferos, entre los que nos encontramos, que tienen dos ojos que comparten una zona de visión común [por tanto poseen visión binocular], han sido dotados con la capacidad de calcular distancias¹²⁶, al menos de forma relativa. Este es un primer nivel que capta las imágenes y las hace **persistentes en la retina** durante unos 40/67 milisegundos [entre 1/25 y 1/15 de segundo]. Esta persistencia permite que nuestro cerebro inconscientemente reaccione rápidamente por la cercanía de los objetos que tenemos delante. Para ello usamos la **percepción de la profundidad**. No reaccionamos igual ante una amenaza a un metro de nosotros que a 100 metros.

124 Con entrenamiento sí podrías. Los francotiradores del ejército calculan la distancia al objetivo, midiendo características de este y comparándolo con las medidas estándar o conocidas. Por ejemplo la altura de una persona o el ancho de un coche.

125 El campo visual total es de unos 220°, pero el campo visual de ambos ojos a la vez es de sólo 120°.

126 Una excelente ayuda para cazar. Mientras que una disposición ocular opuesta (caballos, gacelas, etc.) permite una mayor capacidad de vigilancia frente a depredadores.

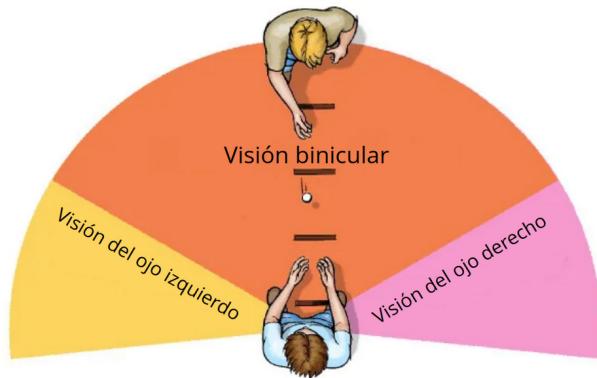


Figura 36: Esquema de la visión estereoscópica humana.

Fuente: Exploring the Depths--of Vision! (Scientific American, By Science Buddies, Megan Arnett on March 9, 2017)

2. Por otro lado, el cerebro forma constantemente una representación 3D del mundo que vemos, capturando todos los detalles de aquello que llama nuestra atención, o donde estamos centrados en ese momento, en el área visual del lóbulo occipital. De este nivel extraemos información que es usada en otros circuitos neuronales como el de la **estimación de las distancias**.

Por tanto poseemos dos¹²⁷ mecanismos para mediar las distancias por medio del sentido de la vista, ambos en el neocortex, mediante funciones de percepción sensorial consciente e inconsciente.

Mi IA favorita dice: *El neocortex [...] es la parte más desarrollada y evolutivamente reciente del cerebro de los mamíferos. Es la región del cerebro responsable de funciones cognitivas superiores como el pensamiento consciente, la percepción sensorial, la generación motora voluntaria, el razonamiento espacial, el lenguaje y la planificación.*

Un agente inteligente necesita de la capacidad de estimar la profundidad de la escena en la que está inmerso porque es un sentido básico para poder interactuar con él. Por ejemplo con los objetivos de **navegación y movilidad**: Estimar la profundidad permite al agente determinar distancias relativas y absolutas en el entorno, lo cual es crucial para moverse de manera segura y efectiva. Un robot que camina o un vehículo autónomo, por ejemplo, necesita poder juzgar la profundidad para evitar obstáculos y seleccionar rutas óptimas. Pero también por razones de **interacción**: La percepción de la profundidad es esencial para tareas como agarrar, manipular objetos o interactuar físicamente con el entorno. Un robot manipulador en una línea de montaje o un dron que intenta aterrizar en un sitio específico necesita comprender la estructura tridimensional del espacio.

Podemos ver un ejemplo en la Figura 37.

¹²⁷ Haz un experimento. Observa a tu alrededor tapándote un ojo y calcula las distancias relativas (lo ideal sería que no conocieras tu entorno y no lo hubieras visto antes). Ahora, probablemente, sientes que estás haciendo un cálculo consciente de algo que antes, con los dos ojos, hacías inconscientemente.

Observa que la salida [derecha] consta de una imagen de igual tamaño que la original, pero en tonos de grises, en donde cada pixel proporciona una estimación de la distancia a la que se encuentra el mismo pixel de la imagen original [izquierda]. Un pixel cercano a 0 está más cerca¹²⁸ de la cámara que otro cercano a 255.



Figura 37: Entrada y salida del pipeline de estimación monocular de la profundidad

Fuente: COCO dataset

Un poco más de dificultad. Imagina que deseas saber la distancia al perro de la Figura 37 [izquierda]. Como puedes ver en la imagen, el perro ocupa varios pixels. ¿cómo puedo saber cual es la distancia media al perro? Respuesta: sumando las distancias de todos sus pixels y dividiendo entre el número de estos que le corresponden al perro. Pero, de nuevo, ¿cómo puedo saber que pixels corresponden al perro? Respuesta: segmentando la imagen.



Video 7: Tesla Autopilot Augmented Vision Test

Imagina un agente inteligente que, en el interior de un androide¹²⁹, posea una cámara la cual constantemente registra una imagen, la segmenta y calcula para cada objeto que detecta la distancia, manteniendo en su memoria un mapa de lo que hay a su alrededor. Incluso no tiene porque ser un androide, imagina estas técnicas en un **coche inteligente** [ver Vídeo 7].

128 Hay otros modelos que lo hacen al revés.

129 Es un término que proviene del griego "andr-" que significa "hombre" y "-oide" que significa "forma" o "parecido a". Por tanto robot con parecido físico a un ser humano.

6.4 Manos a la obra: Extracción de características

En inteligencia artificial, la **extracción de características** consiste en el proceso de identificar y seleccionar atributos relevantes y distintivos de los datos en bruto, como imágenes, texto o señales, que pueden ser utilizados posteriormente para tareas de aprendizaje automático o profundo. Este proceso transforma los datos originales en representaciones más útiles, destacando patrones y propiedades importantes que facilitan la clasificación, el reconocimiento, la predicción y otras operaciones.

La extracción de características se puede comparar con el concepto humano de "recuerdo": Cuando un ser humano recuerda una experiencia o información no recupera cada detalle exacto sino que accede a aspectos clave y representativos de esa memoria. De manera similar, la extracción de características implica seleccionar y resaltar los elementos más relevantes y distintivos de los datos, dejando de lado la información redundante o menos significativa. Ambos procesos buscan simplificar y organizar la información para hacerla más manejable y útil para tareas posteriores.

Habitualmente buscamos obtener con estas técnicas **vectores de características** (*feature vectors*), vectores numéricos pertenecientes a un espacio vectorial de dimensión N , donde N suele ser grande¹³⁰. Esta técnica no es exclusiva del contexto de la visión artificial, como veremos hay otros entornos en donde es empleado ampliamente.

¿Para qué se usan los vectores de características?

En el punto 6.1 vimos como, a partir de la imagen de una persona, podríamos obtener una serie de coordenadas $[x, y]$ de puntos clave para determinar su pose. A partir de estos *keypoints* podemos crear un vector de características:

1. Escoge uno de los *keypoints*, uno que siempre aparezca: por ejemplo la base del cuello.
2. Calcula para el resto de los *keypoints* la distancia entre cada uno de ellos y el *keypoint* del punto anterior.
3. Pon estas distancias calculadas en un vector numérico, siempre en el mismo orden.

Ya tenemos el vector de características que determina de forma única una determinada pose. Ahora podemos comparar poses (cuán parecidas son) sólo haciendo un cálculo: la distancia del coseno.

La distancia del coseno es una métrica utilizada para medir la similitud entre dos vectores en un espacio vectorial teniendo sólo en cuenta el ángulo entre ellos. A diferencia de otras medidas de distancia, como la euclíadiana, la distancia del coseno se enfoca en la orientación de los vectores más que en su magnitud, evaluando cómo se alinean entre sí.

¹³⁰ El hecho de que la dimensionalidad del espacio vectorial sea grande, es una necesidad debido a la gran variabilidad de las características que deseas considerar.

Matemáticamente, se calcula como uno menos el coseno del ángulo entre los dos vectores, dando un valor de 0 cuando los vectores están perfectamente alineados [es decir, poseen idéntica dirección] y un valor de 1 cuando son ortogonales [totalmente disimilares]. En caso de ser vectores totalmente opuestos, su distancia del coseno es 2.

$$\text{distancia del coseno}(\theta) = 1 - \cos(\theta)$$

Sin embargo la distancia del coseno posee un problema: no puede ser usado como *score*. La puntuación [*score*] de cualquier enfrentamiento o concurso deportivo parte de 0 y se incrementa; cuanto más mejor. En la distancia del coseno, cuanto más cerca de cero, mejor. No puede ser usado como métrica de puntuación. Para ello se usa la **similitud del coseno**:

$$\text{similitud del coseno}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

Esto es: el cociente entre el producto escalar de los vectores dividido entre el producto de sus normas [su magnitud].

Si te parece complejo, podemos calcular la similitud a partir de la distancia:

$$\text{similitud del coseno} = 1 - \text{distancia del coseno}$$

Apoyándonos en la IA, no es muy difícil:

Prompt:

Genera dos vectores de números enteros de 8 componentes cada uno de ellos, para después mostrar ambos vectores y calcular la distancia del coseno y la similitud del coseno.

Este *prompt* genera el siguiente código:

```
1 import numpy as np
2 from scipy.spatial.distance import cosine as dcosine131
3
4
5 # Generar dos vectores de números enteros de 8 componentes
6 vector1 = np.random.randint(0, 10, 8)
7 vector2 = np.random.randint(0, 10, 8)
8
9 # Mostrar los vectores
10 print("Vector 1:", vector1)
11 print("Vector 2:", vector2)
12
13 # Calcular la distancia del coseno
14 distancia_coseno = dcosine(vector1, vector2)
15 print("Distancia del coseno:", distancia_coseno)
16
17 # Calcular la similitud del coseno
18 similitud_coseno = 1 - distancia_coseno
19 print("Similitud del coseno:", similitud_coseno)
20
```

¹³¹ Le hemos cambiado el nombre a “dcosine” para que el lector/a no lo confunda con el simple coseno.

Que al ejecutarse arroja el siguiente resultado:

```
Vector 1: [9 8 2 8 8 3 3 4]
Vector 2: [7 9 0 8 5 4 0 0]
Distancia del coseno: 0.07154770124258159
Similitud del coseno: 0.9284522987574184
```

Su distancia del coseno, cercano a cero, nos indica que poseen una dirección semejante, lo cual ratifica la similitud, cercano a 1.

Vamos a aplicar lo visto a varios ejercicios prácticos.

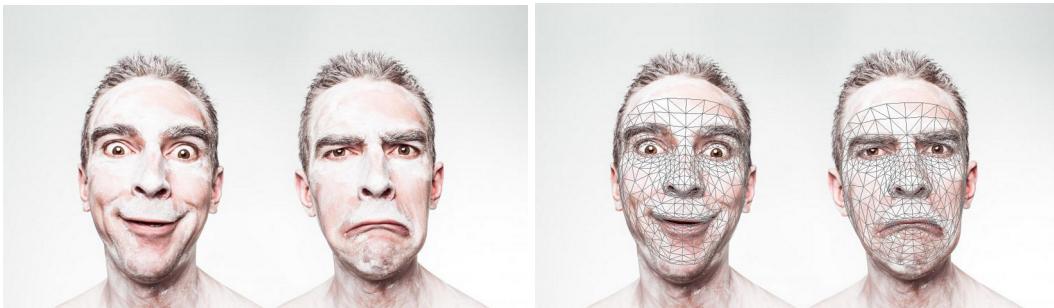
6.4.1 Detección de expresiones faciales

Empecemos por obtener el vector de características de dos rostros.

06 detección de emociones.ipynb

Prompt:

Dada una imagen referenciada mediante una url, muestra los landmarks de las caras que aparezcan en ellas.



06 detección de emociones.ipynb

Prompt:

A partir de los landmarks detectados, genera un vector para cada rostro.

El código generado ha calculado dos vectores, cada uno de ellos de 1404 componentes. Esto significa que cada vector determina un punto en un espacio de 1404 dimensiones.

06 detección de emociones.ipynb

Prompt:

Calcula la distancia del coseno de ambos rostros y su similitud.

Arrojando una distancia de 0.0892 y una similitud de 0.9108; por tanto son similares.

Pero ¿estos vectores de características serán tan precisos como para diferenciar diferentes emociones de la misma persona?

Para poner a prueba el reto, vamos a llevar a cabo una técnica muy utilizada en el aprendizaje automático: el *clustering*.

El clustering es una técnica de aprendizaje automático no supervisado que consiste en agrupar un conjunto de datos en subconjuntos o "clusters" de tal manera que los datos dentro de cada grupo sean más similares entre sí que a los datos en otros grupos según algún criterio.

El objetivo del clustering es descubrir estructuras y patrones subyacentes en los datos, permitiendo identificar y segmentar datos similares sin necesidad de etiquetas o categorías predefinidas.

Esta técnica se usa muy a menudo en diversas aplicaciones, como la segmentación de mercado, el análisis de redes sociales y la organización de grandes volúmenes de datos para facilitar su interpretación y análisis; también para la detección de patrones en imágenes.

Hemos creado un *dataset* consistente en los vectores de características de el mismo rostro poniendo expresiones [con diferentes ángulos y giros de cabeza]; en concreto tres tipo de expresiones: neutral, alegre y triste. Este *dataset* tiene 161 filas [cada uno de ellos un vector de características más la etiqueta] y 53 columnas [52 del vector y 1 de la etiqueta].

Es este conjunto de datos el que vamos a usar para que se realice un *clustering* de 3 grupos [los tres tipo de expresiones] y luego representarlos. Si todo va bien conseguiremos ver tres grupos diferenciados.

06 detección de emociones.ipynb

Prompt:

Carga un dataset de un fichero csv referenciado por una url. Muestra las primeras filas y un resumen de su contenido.

La celda generada importa un fichero .csv [valores separados por comas] y muestra información general sobre esta. El resultado de la importación es un **dataframe** denominado *df*.

06 detección de emociones.ipynb

Prompt:

Usando un algoritmo de clustering, lleva a cabo un agrupamiento de todas las filas y todas las columnas, excepto aquella llamada "__TARGET__". Esta separala para luego usarla como criterio de color a la hora de representar los clusters. El número de clusters es en número de etiquetas diferentes que tiene la columna "__TARGET__". No hagas ninguna representación de los datos.

La celda generada lleva a cabo un entrenamiento de un modelo, usando el algoritmo **kmeans** [ya lo conocerás más a delante]. Como datos toma todo el *dataset* excepto la última columna (TARGET), que es la columna encargada de almacenar que expresión se corresponde para cada vector.

06 detección de emociones.ipynb

Prompt:

Usando el modelo entrenado en la celda anterior, crea una representación en 2d de los datos, coloreando cada punto con la columna TARGET, esta columna contiene tres tipos de expresiones de rostros, neutro, triste y alegre.

La celda generada nos muestra este gráfico:

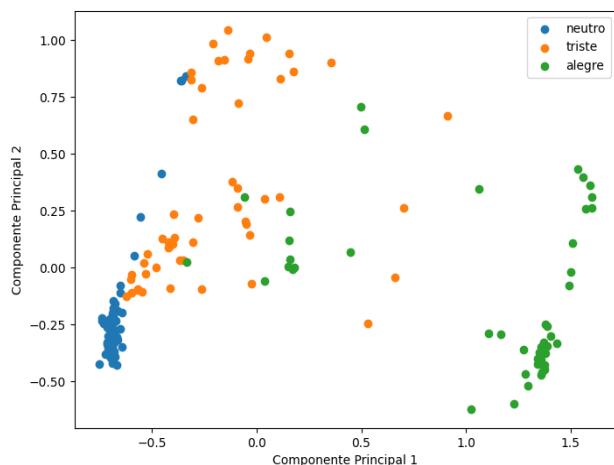


Figura 38: Gráfica 2d del clusterig de expresiones faciales

Cada punto en esta gráfica representa una expresión [su vector de características]. Sabemos que cada vector posee 1404 componentes, así que ¿cómo lo hemos representado en 2d?

Otro tipo de técnica, igualmente muy usada, es la llamada reducción de dimensionalidad.

Los algoritmos de reducción de dimensionalidad son técnicas utilizadas en el análisis de datos para transformar un conjunto de características de alta dimensionalidad en un espacio de menor dimensión, conservando al máximo la información relevante.

Estos algoritmos simplifican los datos al eliminar redundancias y ruido, lo que facilita la visualización, el almacenamiento y el procesamiento de los datos. Pero, a cambio pierden información; la gran ventaja de algunos de estos algoritmos es que podemos conocer e incluso controlar cuanta información se pierde. Por ejemplo, en la Figura 38 hemos usado un algoritmo llamado PCA con el fin de reducir la dimensionalidad de 1404 dimensiones a 2 dimensiones, conservando el 84% de la información (perdiendo el 16%).

Si usamos otro algoritmo [TSNE] para reducir las dimensiones a 3 y representarlo de forma interactiva, podemos ver que los grupos se crean con bastante acierto:

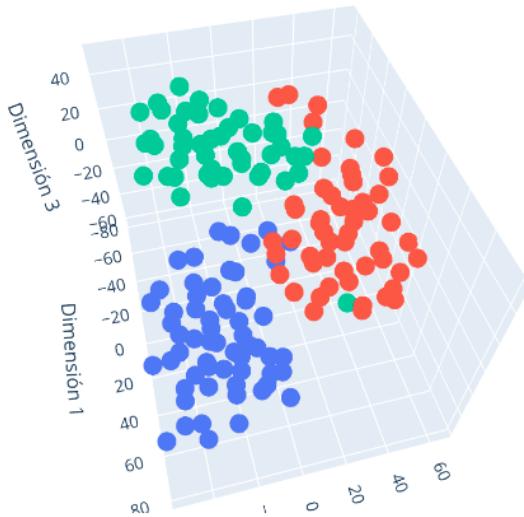


Figura 39: Representación 3d de los clusters

Versión interactiva en “06 detección de expresiones faciales.ipynb”

Volviendo a la Figura 38 podemos observar que la separación entre “neutro”, “triste” y “alegre” es buena¹³², el modelo partiendo de sus vectores es capaz de distinguir entre una expresión neutra, triste y alegre.

Finalmente, la respuesta del reto es: sí. Es posible determinar la emoción de un rostro a partir de su vector de características. Pero ¿y para nuevos rostros? Como resultado de la ejecución de las primera celdas, entrenamos un modelo *kmeans*, este modelo ahora puede procesar nuevos vectores de características e inferir a que *cluster* pertenece [neutro, triste, alegre]. **Es capaz de generalizar.**

6.4.2 Reconocimiento de personas

Un proceso moderno de reconocimiento facial consta de 5 etapas: **detectar, alinear, normalizar, representar y reconocimiento.** [1] en primer lugar el sistema identifica la presencia de rostros en la imagen, [2] una vez detectado se alinea para que todos tengan una orientación similar, por ejemplo, haciendo que los ojos estén a la misma altura y que el rostro esté centrado, [3] seguidamente, el rostro alineado se procesa para tener un formato uniforme, como ajustar el tamaño de la imagen, la iluminación, y la escala de grises o color, con el objeto de reducir las variaciones que podrían afectar el reconocimiento, [4] el rostro normalizado se convierte a vectores de características que captura los atributos únicos del rostro. Finalmente la representación obtenida se compara con representaciones almacenadas en una base de datos u obtenidas en otra fotografía.

132 Pero no perfecta. Las expresiones neutra-triste-alegre, a no ser que se exagere mucho la expresión de las dos últimas, tienen muchos “puntos de contacto”, expresiones que si no sabemos el contexto, no sabemos que emoción (o ausencia de esta) representan.

En el notebook “06 reconocimiento de caras.ipynb” podemos ver un ejemplo de reconocedor de rostros. En primer lugar recopilamos una serie de imágenes *online* de una cara conocida, en este caso Jennifer Lawrence. La segunda celda nos muestra estas imágenes con los landmarks sobreimpresos, sólo a efectos de verificación. Es en la tercera celda en donde podemos ver el proceso completo¹³³:

```

30
31 known_face_encodings = []
32
33 for image in training_images:
34     face_encodings = face_recognition.face_encodings(image)
35     if face_encodings:
36         known_face_encodings.append(face_encodings[0])
37
38 # Promediar las codificaciones faciales
39 if known_face_encodings:
40     person_face_encoding = np.mean(known_face_encodings, axis=0)
41 else:
42     raise ValueError("No se encontraron rostros en las imágenes de entrenamiento")
43

```

Después de descargar las imágenes recorre cada una de ellas [línea 33] recogiendo el vector de características de cada rostro [línea 34, aquí le llama *encodings*¹³⁴] y añadiéndolo a una lista [línea 36]. Al finalizar, si hemos encontrado facciones en las imágenes, hacemos un promedio de todas ellas [línea 40].

Promediar un conjunto de vectores de características es una técnica utilizada para obtener una representación más robusta y generalizada de un rostro, especialmente cuando se dispone de múltiples imágenes del mismo individuo. Al promediar los vectores, se captura la esencia común de las características faciales del sujeto, lo que ayuda a mitigar el impacto de variaciones de pose o expresión facial. Aunque esta técnica puede perder algo de precisión al reducir la variabilidad entre vectores individuales, en la práctica, el promedio suele ser suficientemente representativo y proporciona un buen balance entre robustez y precisión en el reconocimiento, evitando que un único vector con posibles anomalías domine el proceso de comparación.

Una vez que tenemos un vector de características promedio de una persona, y queremos identificarla en otra imagen, sólo debemos hacer:

1. En la nueva imagen, localiza los rostros que aparecen en ella.
2. Para cada rostro detectado:
 - a] Calcula su vector de características usando el mismo método que se usó para calcular el anterior.

¹³³ No ponemos el prompt porque fue generado con Gpt4o y sufrió de modificaciones “a mano”. Para hacerlo con prompts en el mismo Colab, es mejor ir paso a paso,

¹³⁴ Estamos hablando de vectores de características, pero tiene otros nombres (a veces según el contexto): *encodings*, *embeddings*, etc.

- b] Aplica un algoritmo de similitud¹³⁵ entre ambos: por ejemplo la similitud del coseno.
- c] Guarda esta similitud, junto con la localización del rostro.
3. Finalmente, de todas las distancias calculadas, quédate con la menor. Habrás encontrado el rostro original.

El la Figura 40 puedes ver el resultado. De entre todos los rostros de la imagen, el de Jennifer Lawrence es el único destacado.

Si recorres todas las fotografías de la actriz que hemos usado para entrenar el modelo, podrás observar que ninguna de ellas se parece, ni tan siquiera en la expresión, a la de Jennifer. En las imágenes de entrenamiento está posando, en la imagen final sonriendo informalmente.



Figura 40: Imagen de grupo en donde podemos ver, entre otros, a Jennifer.

Autor: Bradley Cooper; publicado en X por Ellen DeGeneres.

Esto significa que el vector de características promedio ha capturado la esencia de la forma de la cara de la actriz. **Ha generalizado**. Notar que el peinado, barba, color de piel o cualquier otra característica de la persona, no es tenida en cuenta. Esto significa que si estuviésemos reconociendo a Bradley Cooper y en todas las fotos de entrenamiento aparece sin barba, seguiríamos reconociéndole igualmente.

¿Y cómo podemos calcular el *score*? Hasta ahora, al llevar a cabo un reconocimiento, el modelo nos entregaba una puntuación. En este caso este número es la similitud del coseno, resultado de comparar los vectores de características [el promedio objetivo y aquellos que representan cada uno de los rostros en la imagen final]. Esta similitud va de cero [nada parecidas] en adelante. Para convertirlas a un valor entre 0 y 1 aplicamos la función **sigmoide**, vista anteriormente.

6.4.3 Detección de gestos

¿Has jugado alguna vez a “piedra, papel, tijera”? Si la respuesta es negativa no vamos a enseñarte ahora, pero sí vamos a entrenar a un modelo a reconocer si una mano tiene una determinada pose, la de piedra, papel o tijera.

135 Porque es un score. En un momento te lo explicamos.

Esta parte práctica empieza por la recopilación de imágenes de manos [una por imagen, indistintamente si es la derecha o la izquierda]. Nuestro *dataset* estará formado por imágenes con nombres como “tijera34.png” o “papel12.png” y con el siguiente aspecto.



Figura 41: Imagen "papel105.png"

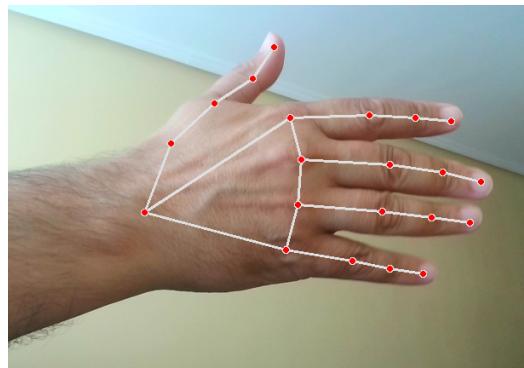


Figura 42: Imagen "papel105.png" con puntos de referencia

Hemos reunido 425 imágenes, y en todas ellas hay una única mano en múltiples posiciones y en tres estados. Además hay dos imágenes mal etiquetadas a propósito. El *notebook* que encabeza esta práctica (“06 reconocimiento de manos.ipynb”) incluye la celda que se usó para generar estas imágenes, aunque tiene su dificultad¹³⁶.

06 reconocimiento de manos.ipynb

Prompt:

Dado un fichero csv referenciado mediante una url, descárgalo, ábrelo y muestra info sobre su contenido.

Ya hemos visto en el apartado 6.4.1 como representar estos vectores de características en 2D y en 3D. Vamos a hacerlo con los gestos.

06 reconocimiento de manos.ipynb

Prompt:

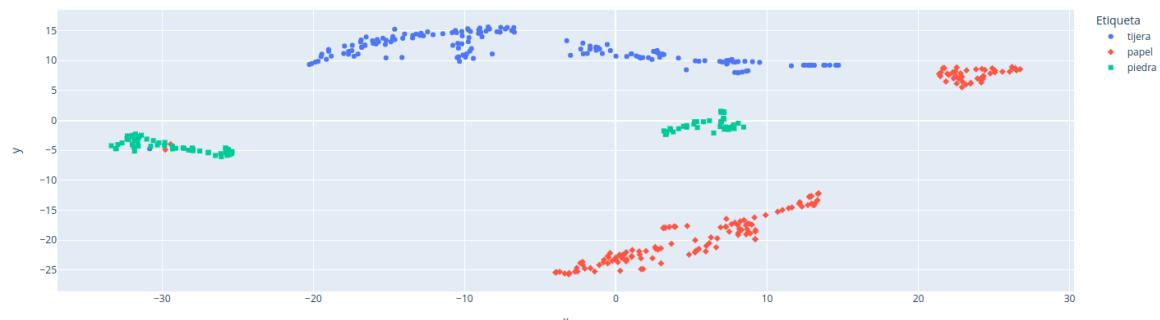
Usando este dataframe entrena un modelo de clustering para 3 clusters usando GaussianMixture, y con TSNE muestra en 2d su dispersión. Ten en cuenta que la última columna es una etiqueta, no la uses para clusterizar, pero si para poner un color en cada fila.

GaussianMixture es un algoritmo de *clustering* basado en densidad que permite hacer predicciones¹³⁷. Anteriormente usamos **Kmeans**, un algoritmo de agrupamiento basado en la distancia entre los puntos. Y de hecho es el primero que probamos, pero al representarlo se comprobó que la nube de puntos se correspondían más con densidades que con cercanía: las “nubes” se representan en

¹³⁶ Google Colab no puede trabajar con la webcam, la que hemos usado para capturar las imágenes. Para seguir usando Colab y aún así poder capturar frames, tenemos que conectarnos a un servicio jupyter local. Esa es la dificultad. De todas formas el fichero .csv está online en el repositorio.

¹³⁷ ¡Aunque aquí no lo usaremos para predecir! Podríamos haber usado otros (DBSCAN, HDBSCAN, etc.).

“jirones”, así que la distancia no era la estrategia más adecuada.



Ahora tenemos una representación en 2D de los vectores, recordemos que cada punto es uno de los vectores y su color es cómo está [nombre del fichero] etiquetado.

Aquí puede observar los “jirones”.

Observamos que las imágenes con tijeras (“azul”) están bien diferenciadas del resto, pero tiene dos grupos, posiblemente mano derecha/izquierda o verso/reverso, al igual que el resto de grupos. Las imágenes tipo papel (“naranja”) también se ven bien separadas del resto, pero algo pasa con los grupos “verde”: el de la derecha está bien separado y *clusterizado*, pero hay algo en el grupo de la izquierda.



Como podemos observar, tenemos tres intrusos en el grupo. Etiquetados como “papel” [2, “naranja”] y “tijera” [1, “azul”], se nos han pasado tres imágenes mal etiquetadas: su vector dice que son “piedra” (“verde”) pero su etiqueta no.

Los dos vectores “piedra”, etiquetados como “papel” fueron introducidos a propósito, pero el tercero no. Se trata de **anomalías**.

Lo normal es identificar estos errores y eliminarlos del *dataset*, dado que van a distorsionar el aprendizaje de cualquier modelo¹³⁸. Sin embargo hay otro tipo de anomalías, cuyo origen no son errores:

Los outliers, o valores atípicos, son muestras en un conjunto de datos que se desvian significativamente de la mayoría de las otras observaciones. Estos valores pueden ser excepcionalmente altos o bajos (según algún criterio) en comparación con el resto de los datos, y su presencia puede indicar variabilidad inusual, errores de medición, o eventos raros y extremos.

138 Analogía: es como si tuvieses que aprenderte fechas de sucesos históricos y en tus apuntes tienes años equivocados.

Los *outliers* pueden influir en los resultados del análisis, por ello es crucial identificarlos y tratarlos adecuadamente para garantizar la validez y precisión del análisis. Veremos algún ejemplo más adelante.

Ahora que hemos analizado los datos, visualizándolos, vamos a entrenar un modelo de *machine learning* para que aprenda a predecir la etiqueta (“piedra”, “papel”, “tijera”) a partir del vector de características que representa el gesto de una mano.

06 reconocimiento de manos.ipynb

Prompt:

Dado el dataframe de gestos de manos, entrena un modelo randomforest para predecir la columna “Etiqueta”, un testset de 20 filas aleatorias del dataset.

He escogido el algoritmo **RandomForest** por su robustez y por necesitar pocos datos para su entrenamiento¹³⁹, en comparación con las necesidades de las redes neuronales¹⁴⁰.

06 reconocimiento de manos.ipynb

```

1  from sklearn.ensemble import RandomForestClassifier
2  from sklearn.model_selection import train_test_split
3  from sklearn.metrics import accuracy_score
4
5
6  # Separa las características de las etiquetas
7  X = df.drop('Etiqueta', axis=1)
8  y = df['Etiqueta']
9
10 # Divide los datos en conjuntos de entrenamiento y prueba
11 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=20, random_state=42)
12
13 # Crea el modelo RandomForest
14 model = RandomForestClassifier()
15
16 # Entrena el modelo
17 model.fit(X_train, y_train)
18
19 # Realiza predicciones en el conjunto de prueba
20 y_pred = model.predict(X_test)
21
22 # Calcula la precisión del modelo
23 accuracy = accuracy_score(y_test, y_pred)
24 print("Precisión del modelo:", accuracy)
25

```

En la línea 11 dividimos el *dataset*, en la 17 entrenamos el modelo con el *trainset* y en la 20 lo probamos con el *testset*. La precisión [accuracy, número de aciertos dividido por el número de elementos] es 1, esto es, usando el *testset* ha acertado todos. Más adelante veremos como analizar más exhaustivamente un modelo y detectar situaciones de overfitting¹⁴¹.

139 *Y para no usar siempre redes convolucionales o XGBoost.*

140 *Este va a ser uno de los criterios para decidir qué algoritmo usar.*

141 *Sobre-entrenamiento, dicho de otro modo, aprender de memoria; lo cual no es generalizar y por tanto no es aprender.*

Vamos a probar el modelo pidiéndole que haga una predicción sobre una imagen con dos manos, que bien podría ser una imagen de un enfrentamiento:

06 reconocimiento de manos.ipynb

Prompt:

Dada una imagen, referenciada mediante una url, y usando el modelo anterior [model], extrae las manos que pueda haber y determina para cada una de ellas su etiqueta [piedra, papel, tijera]. Muestra la imagen y la predicción de cada mano.

El resultado es:



Figura 43: Resultado del reconocimiento de gestos en un enfrentamiento al "piedra, papel, tijera"

Ha acertado. Pero no cantemos victoria, sólo hemos testado el modelo con 20 imágenes [muy parecidas al del conjunto de entrenamiento] y hemos visto que funciona con una imagen diferente. Para ser más sistemáticos deberíamos:

- Tener un *dataset* más variado [más manos diferentes con gestos más complejos]
- Gesto no pertenecientes a las etiquetas “piedra”, “papel”, “tijera”: gesto no reconocido.
- Tener un *testset* más grande y con más variabilidad.
- Probar con otros algoritmos y quedarse con el mejor.

Todo esto lo veremos más adelante.

RETOS DEL CAPÍTULO 6

1. ¿Qué es la percepción en el contexto de la visión artificial?
2. ¿Cómo se relaciona la percepción visual con la percepción humana?
3. Pregúntale a tu IA favorita, cómo crear un modelo en un espacio 3D a partir de la captura de puntos conocidos en una imagen. Con “puntos conocidos” me refiero a que sabemos que identifican: hombros, mano, dedos, etc. El objetivo es crear un *pipeline* del proceso, aunque sea en papel.
4. ¿Qué técnicas se utilizan para la detección de objetos en imágenes?
5. Reflexiona que utilidad puede tener la estimación de la pose, fotograma a fotograma, de un vídeo. Háblalo, quizá, con tu agente conversacional favorito.
6. ¿Cuál es la diferencia entre la detección de objetos y la segmentación semántica?
7. La estimación de la pose y el lenguaje corporal. Lee este artículo y reflexiona en grupo sobre él: <https://medium.com/@samim/human-pose-detection-51268e95ddc2>.
8. ¿Qué es la detección de movimiento y cómo se implementa en visión artificial?
9. Pregúntale a tu IA favorita, qué diferencia hay entre secuencia de fotogramas y flujo de vídeo. Qué es un *stream* de vídeo, y las plataformas de *stream*, qué son.
10. Averigua qué es el sesgo en los datos y como se relaciona con el aprendizaje de modelos, en este caso relacionados con la percepción. Interroga a tu IA favorita con el objeto de conocer qué es un modelo y cómo se entrena.
11. Busca aplicaciones del seguimiento de objetos en vídeo y busca en la web empresas que oferten esos servicios o tecnologías, priorizando las más cercanas a tí.
12. Haz un vídeo en el aula con vosotros haciendo algo normal y observa el resultado del seguimiento [*tracking*]. Vuelve a repetir el vídeo, pero esta vez buscar el fallo, ocultándolo, desapareciendo de la escena y volviendo, etc. Comentar los resultados.
13. El sitio *HuggingFace* es un lugar enorme para buscar modelos. Ve [www.huggingface.co] y busca modelos relacionados con el procesamiento de imágenes.
14. Observa un resultado de la estimación de la profundidad. Por ejemplo la Figura 37. Se trata de una imagen de un sólo canal, con valores por pixel de 0 a 255. Cuanto más cercano a 0 más cerca está ese pixel [en la imagen de partida]. Atrévete a hacer un histograma a partir de la estimación e interprétilo.
15. A partir de la interpretación anterior, ¿podríamos segmentar elementos de la imagen basándonos en el histograma de la estimación de la profundidad? Haría falta un algoritmo que “agrupase” elementos del histograma automáticamente ¿verdad?
16. Siguiendo con los dos anteriores retos ¿podríamos borrar el fondo? Si es así ¿cómo?

Capítulo 7

IMPACTO DE LA VISIÓN ARTIFICIAL EN LA SOCIEDAD



Ilustración 4: Si las máquinas aprenden a ver ¿quién va a controlar a las máquinas?

Fuente: Midjourney, propiedad del autor.

7.1 La automatización y la eficiencia dentro del impacto de la visión artificial en la sociedad

Como hemos visto, con el término “visión artificial” nos referimos a la capacidad de las máquinas de percibir el entorno que las rodea por medio de un canal visual. En el contexto de la automatización, la visión artificial permite que las máquinas lleven a cabo tareas que antes requerían, directa o indirectamente, la intervención humana. Su utilidad se extiende en muchas áreas.

Por ejemplo:

- En la **industria manufacturera**, por ejemplo, la visión artificial se utiliza para inspeccionar la calidad de los productos. Los sistemas de visión artificial pueden detectar defectos, imperfecciones o variaciones en la apariencia de los productos, y tomar decisiones en tiempo real sobre su clasificación.
- En el **ámbito logístico**, la visión artificial se emplea para el seguimiento y clasificación de objetos. Los sistemas de visión pueden identificar y rastrear automáticamente los productos a medida que se mueven a lo largo de la cadena de suministro, lo que facilita la gestión eficiente del inventario, la planificación de rutas y la optimización de los procesos.
- En la **agricultura**, esta se ha convertido en una herramienta de gran valor para mejorar la eficiencia de la producción. Estos sistemas pueden monitorizar y analizar los cultivos, identificando enfermedades o plagas, y optimizando la aplicación de fertilizantes y pesticidas de manera precisa y selectiva. Esto no solo mejora la productividad, sino que también reduce el impacto ambiental al minimizar el uso excesivo de productos químicos.

Además de la automatización de tareas específicas, la visión artificial también permite el desarrollo de robots y sistemas autónomos más avanzados. Por ejemplo, en la industria de la robótica, los robots equipados con sistemas de visión pueden interactuar de manera segura con los humanos, realizar tareas complejas y adaptarse a entornos cambiantes. Estos avances tienen el potencial de transformar industrias enteras, aumentando la productividad y liberando a los trabajadores humanos de tareas repetitivas y peligrosas.

Sin embargo, a pesar de los beneficios indudables, el impacto de la automatización y la eficiencia impulsada por la visión artificial también plantean desafíos sociales y económicos. Por ejemplo, probablemente la automatización de tareas llevará a la pérdida de empleos de trabajadores humanos¹⁴². En el plano económico, la implementación de sistemas de visión artificial requiere una inversión significativa en infraestructura tecnológica y capacitación del personal. Esto puede generar una brecha entre las empresas, regiones, países y sectores que pueden permitirse adoptar estas tecnologías y aquellos que no, incrementando las desigualdades ya existentes.

7.2 La seguridad y la visión artificial

La visión artificial ha transformado la forma en que se lleva a cabo la seguridad y la vigilancia en una variedad de entornos, desde la seguridad pública¹⁴³ hasta la vigilancia en el lugar de trabajo y en espacios privados:

142 *Como en su día la aparición del automóvil supuso la desaparición de empleos relacionados con carroajes de caballos.*

143 *Siempre que juntamos aplicaciones de la IA con personas o entornos públicos debemos tener en cuenta la legislación. Por ejemplo, la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD). También debemos tener en cuenta que el marco regulatorio es diferente para cada país. Lo que está permitido en Gran Bretaña no está permitido aquí, por ejemplo.*

- **Detección de intrusos:** por ejemplo en edificios, aeropuertos, estadios y áreas públicas. Estos sistemas son capaces de identificar comportamientos sospechosos, detectar movimientos no autorizados y activar alarmas o notificaciones para tomar medidas adecuadas.
- **Reconocimiento facial:** la tecnología de reconocimiento facial¹⁴⁴ ha avanzado significativamente en los últimos años, y permite la identificación y verificación de personas en tiempo real a través del análisis de características faciales. Esto ha sido utilizado en aplicaciones como la búsqueda y seguimiento de sospechosos, el control de acceso a áreas restringidas y la detección de personas desaparecidas.
- **Seguimiento de objetos:** permite el seguimiento de objetos en movimiento en tiempo real. Esto se utiliza en aplicaciones de seguridad para rastrear vehículos robados, monitorizar el flujo de tráfico, identificar comportamientos anómalos y realizar análisis forenses en investigaciones criminales.
- **Ánalysis de comportamiento:** la visión artificial puede analizar el comportamiento humano para detectar actividades sospechosas o potencialmente peligrosas. Estos sistemas pueden reconocer patrones de comportamiento anormales, como merodear en áreas restringidas, abandono de objetos sospechosos o comportamiento agresivo.

A medida que la tecnología de visión artificial se ha vuelto más avanzada y ubicua, surgen preocupaciones sobre **la privacidad y el uso ético** de la vigilancia basada en esta tecnología. El reconocimiento facial, en particular, ha generado debates sobre el equilibrio entre la seguridad pública y la protección de la privacidad individual. Además, existe la preocupación de que la recopilación masiva de datos biométricos y el uso indebido de esta tecnología puedan violar los derechos individuales y la intimidad de las personas.

Otro desafío importante relacionado con la visión artificial en la seguridad y la vigilancia es el **sesgo y la discriminación** inherentes a los algoritmos y los datos utilizados. Si los datos de entrenamiento están sesgados o no son representativos de la diversidad de la población, los sistemas de visión artificial pueden producir resultados discriminatorios o identificar erróneamente a individuos inocentes como sospechosos sólo por el color de su piel, por ejemplo.

7.3 El impacto de la visión artificial en el campo de la salud

La visión artificial ha revolucionado la forma en que se diagnostican y tratan diversas enfermedades, de hecho es clave en la mejora de la calidad de la atención médica en general:

- **Interpretación de imágenes médicas:** se utiliza ampliamente a través del análisis de imágenes y datos visuales, gracias a su capacidad de procesar y analizar grandes volúmenes de imágenes médicas con el objetivo de identificar patrones y características relevantes. Por ejemplo, en radiología, los sistemas de visión artificial pueden analizar radiografías, tomografías computarizadas o resonancias magnéticas para detectar y localizar anomalías, como

144 Por ejemplo usando YOLO en cualquiera de sus versiones. Aquí hemos visto ejemplos pequeños, pero nada nos impide entrenar un modelo para reconocer caras concretas e incurrir en un delito tipificado.

tumores o fracturas.

- **Cirugía asistida:** donde los sistemas de visión guían los movimientos precisos de los brazos robóticos durante los procedimientos quirúrgicos. Estos sistemas pueden proporcionar una visión mejorada y ampliada del área quirúrgica, lo que permite a los cirujanos realizar intervenciones más precisas y menos invasivas.
- **Monitorización de pacientes:** permitiendo la detección de signos vitales y cambios en el estado de salud, tanto de pacientes hospitalizados como aquellos que permanecen en sus casas.
- **Salud pública y epidemiología:** pueden analizar imágenes de drones o cámaras de vigilancia para monitorizar el cumplimiento de medidas de salud pública¹⁴⁵, como el uso de mascarillas o el distanciamiento social.

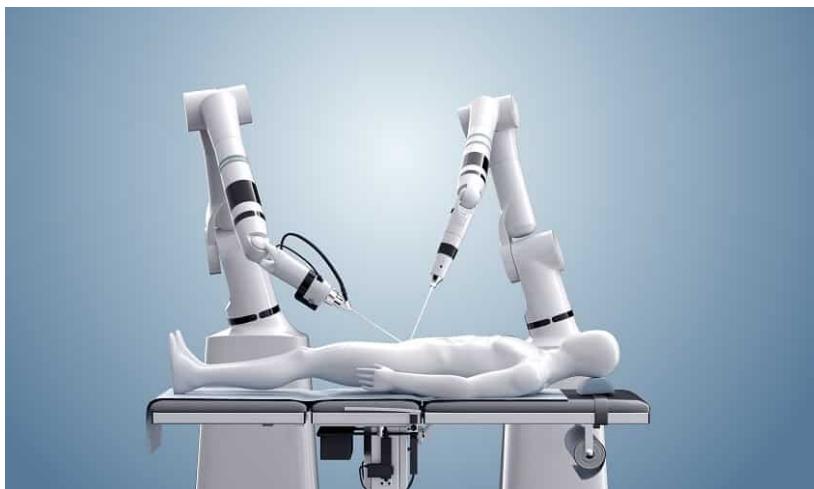


Figura 44: Ejemplo de una hipotética unidad de cirugía operada mediante brazos robóticos.

Fuente: <https://iat.es/tecnologias/robotica/medicina>

Es importante destacar que, si bien la visión artificial ha demostrado ser una herramienta valiosa en la medicina y la salud, también plantea desafíos relacionados con la **precisión de los algoritmos**, la **seguridad de los datos** y la **toma de decisiones basada en algoritmos**. Estos aspectos deben ser considerados y abordados de manera ética y responsable para garantizar el máximo beneficio para los pacientes y la sociedad en general.

145 Esta estrategia se llevó a cabo en China durante la pandemia del 2020. Drones con visión artificial y robots de calle avisaban a viandantes de que debían llevar la mascarilla puesta. En España esta aplicación no está permitida implícitamente en nuestro marco regulatorio.

7.4 La interacción persona-máquina y cómo se ha mejorado

Seguimos desarrollando algunas aplicaciones y aspectos clave relacionados:

- **Reconocimiento de gestos:** lo que permite a las máquinas interpretar y responder de manera más precisa a nuestras expresión corporal. Esto ha llevado al desarrollo de interfaces de usuario que pueden controlarse mediante gestos, como deslizar, hacer zoom, girar o seleccionar objetos en una pantalla táctil o en un entorno de realidad aumentada.
- **Reconocimiento de expresiones faciales:** esto permite a las máquinas comprender las emociones de los usuarios y responder de manera adecuada. Por ejemplo, los asistentes virtuales pueden adaptar sus respuestas y comportamientos en función de las expresiones faciales del usuario, lo que crea una interacción más personalizada.
- **Seguimiento ocular:** al permitir que las máquinas determinen en qué parte de una pantalla o entorno visual se está enfocando la mirada del usuario. Esta capacidad es especialmente útil en campos como la investigación de mercado, la ergonomía y la accesibilidad.
- **Interfaces de realidad virtual y aumentada:** permiten rastrear los movimientos y la posición de la cabeza y las manos del usuario, lo que proporciona una experiencia inmersiva en entornos virtuales. Además permite superponer información digital en el mundo real, enriqueciendo la interacción y la percepción del usuario.
- **Robótica y asistentes virtuales:** estos utilizan sistemas de visión artificial para reconocer y seguir a las personas, interpretar comandos basados en gestos o expresiones faciales, y realizar tareas colaborativas de manera segura.

7.5 El impacto de la visión artificial en la mejora de la accesibilidad

La visión artificial ha desempeñado un papel fundamental en el desarrollo de tecnologías y aplicaciones que permiten a las personas ciegas o con baja visión interpretar su entorno y acceder a información impresa.

Uno de los avances más significativos en este campo es el reconocimiento óptico de caracteres [OCR, por sus siglas en inglés]. Los sistemas de OCR utilizan algoritmos de visión artificial para convertir texto impreso en imágenes en texto digital que puede ser leído por programas de lectura de pantalla o convertido en voz. Esto significa que las personas con discapacidad visual pueden acceder a una amplia gama de materiales impresos, como libros, documentos, carteles y etiquetas.

La visión artificial también se ha aplicado en el reconocimiento de objetos. Los sistemas de visión artificial pueden identificar y describir objetos en tiempo real, lo que brinda a las personas con discapacidad visual información sobre su entorno [ver Vídeo 8]. Por ejemplo, una aplicación móvil equipada con visión artificial puede identificar billetes, productos envasados, señales de tráfico o

incluso caras, y proporcionar información auditiva sobre ellos.

Otra área de aplicación es la navegación asistida. Los sistemas de visión artificial pueden detectar obstáculos y guiar a las personas con discapacidad visual a través de entornos complejos.



Video 8: New AI tool describes surroundings to visually impaired people

Fuente: NBC News

La visión artificial también ha mejorado la accesibilidad en el ámbito del entretenimiento. Los sistemas de audio-descripción utilizan técnicas de visión artificial para detectar y describir visualmente escenas, acciones y expresiones faciales en películas y programas de televisión. Estas descripciones se proporcionan de forma auditiva, lo que permite a las personas con discapacidad visual disfrutar de una experiencia más completa al ver contenido audiovisual.

Es importante destacar que si bien esta tecnología inteligente ha mejorado la accesibilidad para las personas con discapacidades visuales, aún existen desafíos por superar. La **precisión** y **fiabilidad**¹⁴⁶ de los sistemas de visión artificial siguen siendo áreas a mejorar, y es necesario abordar las barreras tecnológicas y económicas para asegurar que estas tecnologías sean accesibles y asequibles para todos.

7.6 El impacto de la visión artificial en la industria del entretenimiento

A lo largo de los últimos años, la visión artificial ha desempeñado un papel fundamental en la transformación de la forma en que se crea y experimenta el entretenimiento, especialmente en el cine, los videojuegos y la realidad aumentada.



Video 9: Virtual Production - A Cinematographers' Conversation

Fuente: Australian Cinematographers Society

146 Más adelante estos términos se harán muy importantes.

En la **industria cinematográfica**, la visión artificial ha revolucionado los efectos visuales y la postproducción. Con técnicas como la captura de movimiento se pueden crear personajes digitales realistas y escenas impresionantes. Los actores pueden interpretar personajes completamente generados por computadora, y los entornos digitales pueden ser creados con gran detalle y realismo.

Además, la visión artificial también se utiliza en la restauración y remasterización de películas clásicas, mejorando su calidad visual y preservando el legado cinematográfico.

En los **videojuegos**, también ha permitido avances significativos. Los desarrolladores pueden crear mundos virtuales más realistas y detallados, gracias a algoritmos de renderización y técnicas de iluminación, ambos basados en inteligencia artificial. Además, la visión artificial se utiliza, como ya hemos dicho, en el reconocimiento de gestos y movimientos, lo que ha dado lugar a la popularidad de los juegos de realidad virtual y realidad aumentada. Los jugadores pueden interactuar de manera más inmersiva con los juegos, utilizando sus cuerpos y movimientos como entrada.

La **realidad aumentada** [AR] y la **realidad mejorada** [XR] son otros campos en el que la visión artificial ha tenido un impacto significativo en el entretenimiento. Las aplicaciones de AR, como Pokémon Go, han ganado popularidad al permitir a los usuarios superponer elementos digitales en el mundo real a través de la cámara de un dispositivo móvil. La visión artificial se utiliza para reconocer y rastrear el entorno, así como para superponer objetos virtuales de manera precisa. Esto ha llevado a experiencias de juego y entretenimiento únicas, donde los usuarios pueden interactuar con personajes y objetos virtuales en su entorno físico.

Sin embargo, junto con los avances emocionantes, también existen desafíos y consideraciones éticas en la industria del entretenimiento en relación con la visión artificial. Su uso plantea preguntas sobre la **auténticidad** e **integridad** del contenido, especialmente cuando se utilizan técnicas generativas, o se plantean cuestiones de **propiedad intelectual** y **derechos de autor** cuando se utilizan estas mismas técnicas.

7.7 La ética y la responsabilidad en el contexto del impacto de la visión artificial en la sociedad

A medida que esta tecnología avanza y se implementa en diversos ámbitos, surgen preocupaciones éticas y responsabilidades asociadas a su uso. Vamos a ver algunas de estas cuestiones y desafíos que ya he ido resaltando en este capítulo:

- **Privacidad:** La visión artificial implica la recopilación y el análisis de datos de carácter visual, lo que puede plantear preocupaciones sobre la privacidad.
- **Discriminación y sesgo:** Existe el riesgo de que los sistemas de visión artificial puedan introducir sesgos y discriminación en las decisiones automatizadas.

- **Toma de decisiones automatizada:** Con la visión artificial, se está delegando cada vez más la toma de decisiones¹⁴⁷ a sistemas automatizados. Esto plantea preguntas sobre la responsabilidad y la rendición de cuentas. ¿Quién es responsable cuando un sistema automatizado toma una decisión equivocada o produce resultados perjudiciales? ¿Cómo se pueden explicar y auditar las decisiones tomadas por algoritmos de visión artificial?
- **Consentimiento informado:** El uso de la visión artificial a menudo implica la recopilación y el análisis de datos personales. Es crucial garantizar que las personas comprendan plenamente cómo se utilizan sus datos y den su consentimiento informado. Esto incluye informar de manera clara y comprensible sobre las implicaciones y los posibles riesgos asociados al su uso, y ofrecer opciones y control sobre los datos personales.
- **Transparencia y explicabilidad:** Los sistemas de visión artificial a menudo se basan en algoritmos complejos que pueden ser difíciles de comprender para las personas que no hayan sido instruidas con ese fin. Esto plantea desafíos en términos de transparencia y explicabilidad. Para mantener la confianza en la tecnología, es importante desarrollar técnicas que permitan comprender y explicar cómo los sistemas de visión artificial llegan a sus decisiones.
- **Responsabilidad social:** A medida que la visión artificial se integra cada vez más en nuestra sociedad, es fundamental considerar su impacto social en su conjunto. Esto implica afrontar preguntas como por ejemplo: ¿cómo se distribuyen los beneficios y las cargas de la visión artificial? ¿Cómo se pueden mitigar los posibles efectos negativos en comunidades marginadas o desfavorecidas? ¿Cómo se puede garantizar que la tecnología sea utilizada de manera ética y responsable para el bien común?

Estas cuestiones éticas y responsabilidades asociadas a la visión artificial son cruciales para garantizar un desarrollo y una implementación adecuada de esta tecnología. Es necesario un diálogo continuo entre los desarrolladores, los responsables políticos, los expertos en ética, las organizaciones de derechos humanos y la sociedad en general para abordar estos desafíos de manera responsable y promover el uso ético y equitativo de la visión artificial.

7.8 La IA generativa y los *deep fakes*

Para finalizar este capítulo y sección, vamos a introducir una tecnología que no es propiamente visión artificial, pero que es una fuente de problemas cuando es usada de forma insensible o incluso delictiva: se trata del uso del **deep learning para generar imágenes**. Si bien es una fuente de investigación constante y está dando resultados espectaculares, tanto en su ámbito como en su aplicación a otros ámbitos, el procesamiento de las imágenes con el objeto de crear **deep fakes** y otros contenidos en forma de audio y vídeo, trae de cabeza a la sociedad por su potencial negativo.

¹⁴⁷ Imagina: un coche con conducción autónoma total, para evitar daños a quienes van en su interior, efectuá una maniobra que pone en peligro otros coches u otros peatones, pudiendo incluso ocasionar su muerte.

Mi IA favorita dice: Los deep fakes son manipulaciones digitales sofisticadas que utilizan algoritmos de aprendizaje automático para combinar o reemplazar la apariencia de una persona en imágenes o videos con la de otra persona¹⁴⁸.



Figura 45: Caras generadas artificialmente. Estas caras no existen, fueron "imaginadas" por una IA. A la izquierda caras generadas en el año 2014, a la derecha caras generadas en el año 2018.

Fuente: Goodfellow et al.; Karras, Laine, Aila / Nvidia

La capacidad de generar contenido de audio y vídeo altamente realista mediante estos algoritmos ha planteado serias preocupaciones éticas y sociales. Aunque los *deep fakes* pueden tener aplicaciones legítimas, como en la industria del entretenimiento o la producción de efectos visuales, también se han utilizado con fines maliciosos y engañosos. Algunos de los problemas y riesgos asociados con los *deep fakes* y la IA generativa en general son:

- **Desinformación y manipulación:** Pueden ser utilizados para crear y difundir contenido falso con el potencial de engañar y manipular a las personas. Esto puede tener consecuencias graves en diversos ámbitos, como la política (ver Vídeo 10 y Vídeo 11), los medios de comunicación y la sociedad en general. La capacidad de alterar videos e imágenes de manera convincente puede socavar la confianza pública, generar confusión y dificultar la distinción entre lo real y lo falso.
- **Daño a la reputación y privacidad:** Pueden ser utilizados para difamar a personas inocentes, poniendo en riesgo su reputación y privacidad. Al reemplazar el rostro de alguien en un video comprometedor, los perpetradores pueden hacer que parezca que esa persona está involucrada en actividades ilegales, inapropiadas o comprometedoras, lo que puede tener un impacto devastador en su vida personal y profesional.
- **Acoso y extorsión:** Los *deep fakes* también se han utilizado como herramientas de acoso y extorsión. Los perpetradores pueden crear y difundir contenido pornográfico falsificado en el que las caras de las víctimas se sustituyen por las de otras personas, lo que genera un daño emocional y psicológico significativo.

148 También puede aplicarse al audio, duplicando el timbre y expresiones de otra persona.

- **Fraude y estafas:** Pueden facilitar diversas formas de fraude y estafas. Por ejemplo, los estafadores pueden utilizar voces generadas por IA para hacerse pasar por personas reales en llamadas telefónicas o mensajes de voz, engañando a las víctimas para que divulguen información confidencial o realicen transferencias de dinero.



Video 10: Fake Obama created using AI video tool

Fuente: BBC News



Video 11: Tricked by the fake Obama video? Deepfake technology, explained

Fuente: USA Today

Ante estos desafíos, es crucial desarrollar técnicas de detección efectivas y promover una educación digital tal que **las personas puedan discernir entre contenido real y falso**. Asimismo, se requiere un **marco legal y ético** sólido que aborde los problemas derivados de los *deep fakes* y proteja los derechos individuales, la privacidad y la integridad de la información. La colaboración entre los gobiernos, la industria y la sociedad en general es fundamental para abordar de manera efectiva los problemas y riesgos asociados con la IA usada para un fin delictivo o poco social.

RETOS DEL CAPÍTULO 7

1. ¿Cuáles son las tecnologías clave detrás de la visión artificial?
2. ¿Cuáles son las aplicaciones más comunes de la visión artificial en la vida cotidiana?
3. Trabajo en grupo: Vuelve al Capítulo 3, ve punto por punto y descubre campos o ámbitos en donde las tecnologías relacionadas con la visión artificial puede prestar su ayuda. Anota también donde puede ser un peligro.
4. ¿Cuáles son los desafíos éticos de la visión artificial en la seguridad y la privacidad?
5. Interroga qué normas legales impiden que la visión artificial desarrolle plenamente su funcionalidad en tu país. Puedes usar un agente conversacional, pero debes comprobar sus respuestas.
6. ¿Cómo se emplea la visión artificial en la medicina y la atención sanitaria?
7. Reflexiona en grupo, por qué hay países más estrictos con las tecnologías basadas en la visión artificial y otros mucho más laxos. Averigua por qué existen estas diferencias.
8. ¿Cómo la visión artificial ha contribuido al desarrollo de la robótica?
9. Poner en funcionamiento un debate ético: dividir al alumnado en dos grupos y asignarles posiciones [a favor o en contra] sobre el uso de la visión artificial en la vigilancia pública. Cada grupo debe presentar argumentos y contraargumentos.
10. ¿Qué es el aprendizaje profundo y cómo se relaciona con la visión artificial?
11. Individualmente o por grupos, invitar al alumnado a imaginar escenarios futuros donde la visión artificial juega un papel dominante en la sociedad. Luego, discutir las implicaciones éticas y sociales de estos escenarios. ¿Qué pros y contras tendrían tus escenarios?
12. Dado que la visión artificial requiere de la grabación de imágenes, quizá faciales o de grupo ¿que repercusión tendrían los *deep fakes* y tecnologías de síntesis de imágenes si pudiéramos sintetizar cualquier imagen?
Básate en este artículo <https://www.bbc.com/news/world-us-canada-65069316>.
13. Una visión artificial muy desarrollada podría llevar a cabo análisis de comportamiento. Dialoga con tu IA favorita para que te cuente el argumento de la novela “1984”, de George Orwell. Relaciona el concepto de “Gran hermano” con la visión artificial.
14. Busca fuentes de información que te muestren cómo se está aplicando el procesamiento de imágenes en el ámbito del entretenimiento en general, y de la visión artificial en particular.
15. A partir del reto anterior, extraña el futuro de la visión artificial en el entretenimiento. ¿Qué podrá hacer? ¿Será caro? ¿Será útil?
16. Realidad virtual y visión artificial. Discute con tu IA favorita su relación y, junto con el resultado del reto anterior, imagina un futuro hipotético.

Sección III: Procesamiento del lenguaje natural

"El procesamiento del lenguaje natural es un campo de investigación multidisciplinario que combina lingüística, informática y ciencias cognitivas para permitir que las máquinas comprendan y utilicen el lenguaje humano."

Dan Jurafsky (1961 -)

El lenguaje es un sistema de símbolos y reglas utilizado para representar y comunicar ideas, emociones, conceptos o información entre entidades. El lenguaje hablado y escrito es el principal medio por el cual nos comunicamos los seres humanos. Así que es natural que si pretendemos crear máquinas que simulen la inteligencia humana, debemos enseñarlas a entender y expresarse en nuestro idioma.

Capítulo 8

CODIFICANDO DATOS

A lo largo de las páginas anteriores hemos recalcado en varias ocasiones que las tecnologías inteligentes, especialmente aquellas basadas en redes neuronales, sólo trabajan con números. Cualquier dato, antes de ser “engullido” por el algoritmo para predecir, clasificar, agrupar, imaginar, etc. necesita que la información (en forma de datos) sea convertida a números. Igualmente, en la mayoría de las ocasiones, el resultado son igualmente números que deben ser interpretados.

Así pues, vamos a dedicar un capítulo a las diferentes formas de codificar información en datos y estos en números; haciendo un especial hincapié en cómo se codifica el texto, por ser el dato/información fundamental¹⁴⁹ de la IA generativa.

Empecemos por el principio. Probablemente te han contado que un **bit**, abreviatura de *binary digit* (dígito binario), es la unidad básica de información en las ciencias de la computación¹⁵⁰ o en la teoría de la información¹⁵¹. Es correcto. Pero también debe ser entendido como una unidad de información, de la misma forma que el metro es la unidad de distancia. En el apartado 1.3 vimos como estaba asociado a la incertidumbre. De hecho 1 bit es la cantidad de información cuando hay una incertidumbre de dos posibles sucesos equiprobables. Como la ingeniería necesita conceptos más prácticos, y requiere además de un sistema de representación, resumimos lo que acabamos de decir diciendo que un bit representa dos posibles valores¹⁵², 0 y 1, ambos con la misma probabilidad¹⁵³.

Nota: *hablamos de situaciones de equiprobabilidad, pero ¿qué cantidad de bits hay en la tirada de una moneda al aire, sabiendo que la cara sale 4 de cada 10 veces y la cruz 6 de cada 10? En este caso para calcular la cantidad de información de ese suceso en bits hay que aplicar la entropía de Shannon, la cual nos arroja que este suceso genera 0,971 bits de información.*

149 Aunque no única ya que tenemos imagen→imagen, vídeo→vídeo, etc.

150 https://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n

151 https://es.wikipedia.org/wiki/Teor%C3%ADa_de_la_informaci%C3%B3n

152 También encendido/apagado en lógica de circuitos, true/false en lógica, etc.

153 Pero un 0 o un 1 no es un bit. Un bit es una botella que sólo puede estar llena (1) o vacía (0) y que la probabilidad de uno u otro estado es equiprobable. Aquí diríamos que la botella contiene un bit.

La memoria de un dispositivo electrónico, cualquiera con tal de que tenga una memoria, es “simplemente” un mecanismo que permite almacenar secuencias de bits, cada bit conteniendo un 0 o un 1. El ordenador que estoy usando en estos momentos tiene varias memorias, pero la principal [RAM] posee sesenta y cuatro mil millones de bits, $64 \cdot 10^9$ bits. Si cada uno de estos bits fuesen 6 milímetros, esta memoria representaría una distancia superior a la que hay entre la tierra y la luna.

Mi IA favorita dice: *En ingeniería informática se trabaja con potencias de 2. Esto es debido a la naturaleza binaria de los sistemas digitales, que utilizan bits para representar datos. Cada bit puede tener uno de dos valores posibles: 0 o 1, lo que hace que las combinaciones posibles de bits sean potencias de 2. Esto se refleja en la organización de la memoria, el almacenamiento y el procesamiento de datos, donde la alineación a potencias de 2 optimiza la eficiencia y la velocidad de los cálculos, ya que los procesadores están diseñados para manejar datos en estas unidades binarias de manera más natural y rápida.*

Trabajar bit a bit es pesado y muy ineficiente desde el punto de vista de la arquitectura de los dispositivos, en concreto de quien usa la memoria RAM [típicamente la CPU]. Leer y escribir de bit en bit es lento, es mejor leer/escribir/transmitir en paquetes de bits, de ahí aparecen el byte [8 bits], la palabra [word, 16 bits], la doble palabra [dword, 32 bits], etc.

8.1 Codificando números

Pero con un bit, o miles de millones, poco podemos hacer. Necesitamos, como mínimo, poder almacenar números usando estos bits [o grupos/paquetes de bits].

8.1.1 Números naturales

Los números naturales son aquellos del 0 en adelante. ¿Cómo podemos almacenar el 357 como una secuencia de bits? En el fondo ya sabemos hacerlo, sólo tenemos que adaptar el método.

El ser humano usa la base 10 para contar; tenemos 10 dígitos [de 0 a 9] porque tenemos 10 dedos. Posiblemente, en la antigüedad, los pastores¹⁵⁴ contaban con los dedos las ovejas que salían al pasto por las mañanas y recordaban el gesto de ambas manos para, por la noche, ir descontando y saber así si le faltaba alguna oveja. Con el tiempo, en vez de usar los dedos, usaron símbolos; lo que hoy llamamos dígitos¹⁵⁵. Pero un día un pastor no le llegaron los dedos/símbolos, tenía 10 ovejas, así que al llegar al equivalente al dígito 9, apartó una piedra con el pie y empezó de nuevo la cuenta en el 0. Lo que acabamos de describir es un sistema de numeración posicional de base 10.

Pero, ¿y si sólo tuviésemos un dedo y por tanto 2 dígitos, el 0 y el 1? El método sería el mismo,

¹⁵⁴ Fábula de base pedagógica, no es un hecho, sólo una narrativa inventada para enseñar algo.

¹⁵⁵ La palabra “dígito” proviene del latín “digitus”, que significa “dedo”.

pero en base 2: 0, 1, 10, 11, 100, 101, 110, 111, etc. Es así de simple, al igual que en base 10, vamos “apartando una piedra” a la izquierda [añadimos un 1] cada vez que llegamos al final de los dígitos de nuestra base.

Pero necesitamos poder convertir números naturales en base 10 a números en binario [base 2].

Para convertir un número de base 10 (decimal) a su equivalente en base 2 (binario), se divide el número decimal por 2 repetidamente, anotando el resto de cada división. Los restos obtenidos en cada paso forman los dígitos del número binario, comenzando desde el último resto (más significativo) hasta el primer residuo (menos significativo).

Por ejemplo, para convertir el número decimal 13 a binario, se realiza lo siguiente: 13 dividido por 2 da 6 con residuo 1; 6 dividido por 2 da 3 con residuo 0; 3 dividido por 2 da 1 con residuo 1; 1 dividido por 2 da 0 con residuo 1. Escribiendo los residuos de abajo hacia arriba, se obtiene 1101 como el equivalente binario de 13.

Para convertir un número en base 2 (binario) a su equivalente en base 10 (decimal), se debe sumar los productos de cada dígito del número binario por 2 elevado a la potencia correspondiente a su posición. Las posiciones se cuentan desde la derecha, comenzando en 0.

Para convertir el número binario 1011 a decimal, se calcula $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$ lo que da como resultado $8 + 0 + 2 + 1 = 11$ en base 10. Cada dígito del número binario contribuye al valor total en función de su posición.

Pero, como hemos dicho, trabajamos en paquetes de bits. De hecho trabajamos con paquetes de tamaños estandarizado: **byte**, **word**, **dword**, **qword**, etc., respectivamente 8, 16^{156} , 32, 64 bits. Así que surge la duda: ¿qué tamaño debo usar? La respuesta es “depende”. Depende de cuan grande sean los números que quieras almacenar en este paquete de bits. Si hablamos de números naturales los límites son:

- Con 1 byte: de 0 a 255
- Con 1 word: de 0 a 65.535
- Con 1 dword: de 0 a $4.294.967.295$ [$\sim 4,29 \times 10^9$]
- Con 1 qword: de 0 a $18.446.744.073.709.551.615$ [$\sim 1,84 \times 10^{19}$]

Si se necesita almacenar números más grande, tenemos el **octa word** [128 bits, $\sim 3,40 \times 10^{38}$] o técnicas de números de tamaño arbitrario [operaciones más lentas, pero sin preocupaciones por los límites].

Finalmente, como hemos dicho, cuando trabajamos con números naturales [o con cualquier tipo, como los que veremos] tenemos un límite, lo que obliga a elegir el tamaño del “paquete” de bits.

¹⁵⁶ En general word son 16 bits (2 bytes), pero es posible encontrar documentación en donde se hable de word como 32 bits. Depende de la arquitectura.

8.1.2 Números enteros

Básicamente para codificar un número entero, codificamos un número natural y le añadimos el signo. Para esto último hay varias maneras:

- Si es negativo, añadiendo un 1 a la izquierda, 0 en caso contrario. Ejemplo, queremos codificar el número 5 [00000101 en binario¹⁵⁷], el 0 a la izquierda indica que es positivo; pero si se tratase de -5 [10000101] tendríamos a la izquierda un 1 indicando que es negativo. Esta estrategia, simple, tiene dos problemas; por un lado, para operar con estos número hay que entender que el dígito de la izquierda representa el signo, así que la lógica en la CPU debe “saber” que estamos tratando con números enteros; por otro lado tenemos un problema, ¿qué número son 00000000 y 10000000? en efecto, el problema es que tenemos dos ceros uno positivo y otro negativo.
- Para eliminar el problema de los dos ceros, se usa la codificación de **complemento a dos**, la representación más usada para números con signo. Funciona así: si el número es positivo se codifica con un 0 a la izquierda y el resto en base 2 [el número 5 sería 00000101], pero si el número es negativo se lleva a cabo una representación un poco más compleja:
 - Primero se representa el número [-5] en base 2 sin signo [00000101]
 - Se haya el complemento a uno¹⁵⁸ de ese número [11111010]
 - Se le suma 1 [11111011]

Según esta técnica, el 00000000 es el cero y el 10000000 es -128. Ya no tenemos el problema de los dos ceros.

El complemento a dos es la técnica más usada para representar un número entero [con o sin signo], esto es algo que resulta transparente para el usuario; pero hay otra cosa que quizás no: **el desbordamiento**.

Los límites de un byte codificando un número entero son: -128 a +127, un rango de 256 posibles números. Pero ¿qué pasa si sumamos 1 a 127? 127 en binario complemento a dos es 01111111 y si le sumamos 1 el resultado es 10000000 [-128] con un indicador desbordamiento. Básicamente es lo mismo que ocurre en el cuenta kilómetros de tu coche, cuando llega al máximo vuelve al mínimo; en este caso el mínimo no es cero, es -128. Algo equivalente pasaría si al -128 le restas uno, volvería al máximo, 127.

El desbordamiento puede ocurrir cuando operamos con números codificados en paquetes de bits de un tamaño fijo, como los que estamos describiendo. Normalmente esto es un problema, causa de muchos errores que al programar hay que tener muy en cuenta¹⁵⁹.

¹⁵⁷ En estos ejemplos supondremos que lo estamos empaquetando en 1 byte (8 bits).

¹⁵⁸ Complemento a uno: Donde hay un 0 se pone un 1 y donde hay un 1 se pone un 0.

¹⁵⁹ Depende del lenguaje. Los hay que detectan esta situación y aumentan el tamaño del “paquete” de bits (el tipo).

8.1.3 Números en punto flotante

Los números naturales y enteros ya están tratados, pero necesitamos codificar los números reales.

Los números en punto flotante se codifican en un formato binario diseñado para representar una amplia gama de valores, incluidos números muy grandes, muy pequeños, fraccionarios y enteros, con un compromiso entre precisión y rango.

El estándar más comúnmente utilizado es el IEEE 754, que define cómo se representan los números en punto flotante en “paquetes” de bits [*dword* o *qword*], veamos un ejemplo: Este formato usa los bits de la siguiente manera [S][E][M], siendo S el signo (1 bit), E el exponente (8 u 11 bits) y M la mantisa (23 o 52 bits). Básicamente estamos codificando el número

$$(-1)^S \cdot M \cdot 2^E$$

Lo que se almacena en el paquete de bits [dword o qword] es S, M y E, de forma consecutiva. Por ejemplo, el número 5,75 se codifica [32 bits] como 01000000101110000000000000000000.

Toda la complejidad de codificar, decodificar y operar con números en cualquier formato, como naturales, enteros, punto flotante u otros, queda completamente oculta al usuario gracias a las abstracciones proporcionadas por los lenguajes de programación y el hardware subyacente. Los compiladores y procesadores manejan automáticamente la conversión de datos entre diferentes formatos, realizan cálculos, y gestionan la precisión y el rango de los números, permitiendo a los desarrolladores trabajar con operaciones matemáticas y manipulación de datos sin preocuparse por los detalles de cómo se representan y procesan los números a nivel binario.

Todo dato o información dentro de un ordenador, por complejo que sea, se codifica como números y estos, a su vez, como bits.

8.1.4 Escalares, vectores, matrices y tensores

En el aprendizaje automático [ML], es común trabajar con vectores, como ya hemos visto en capítulos anteriores, donde los vectores de características juegan un papel fundamental en la construcción de soluciones. Matemáticamente, un vector es una entidad que posee magnitud y dirección, representada como un conjunto ordenado de n componentes que define un punto en un espacio de n dimensiones.

En ingeniería informática, un **vector** se define simplemente como un conjunto ordenado de números (del mismo tipo). Si sabemos codificar un número, podemos codificar un vector, ya que este se almacena en memoria como una secuencia de números consecutivos. Sin embargo, además de almacenar los números, es crucial también guardar la cantidad de componentes del vector y su tipo.

po¹⁶⁰. Por ejemplo, un vector de 15 componentes se representaría en memoria por los 15 números que lo componen, seguidos del número 15 [que indica el tamaño del vector] y un código que define el tipo¹⁶¹ de los componentes, totalizando 17 números.

Una **matriz**, por otro lado, es una estructura matemática bidimensional compuesta por elementos organizados en filas y columnas. Puede considerarse como un conjunto de m vectores, cada uno con n componentes, formando así una matriz de dimensiones $m \times n$.

Si los vectores son estructuras de una dimensión y las matrices son de dos dimensiones, los escalares, que son simplemente números, tienen 0 dimensiones. Matemáticamente, un escalar es una cantidad que se define únicamente por su magnitud, sin ninguna dirección asociada.

Esto establece una progresión clara: los escalares son números de dimensión 0, los vectores son estructuras de dimensión 1, y las matrices son de dimensión 2. ¿Qué sucede más allá de las matrices? Las estructuras que tienen más de dos dimensiones se denominan, en general, tensores.

Los **tensores** son fundamentales en inteligencia artificial, especialmente en ML y redes neuronales, porque permiten representar y manipular datos en múltiples dimensiones de manera eficiente. En ML, los datos a menudo se presentan en forma de grandes matrices multidimensionales, como imágenes, videos o secuencias de texto. Los tensores proporcionan una estructura flexible y potente para procesar y manejar estos datos complejos.

Además, en las redes neuronales, los tensores se utilizan para representar los pesos, las entradas y las salidas en cada capa de la red. Las operaciones matemáticas que se realizan durante el entrenamiento y la inferencia, como multiplicaciones de matrices y sumas, se implementan a través de cálculos con tensores, lo que permite gestionar de manera eficaz el flujo de datos y las transformaciones necesarias para el aprendizaje.

8.2 Codificando textos

El lenguaje hablado y escrito es el principal medio por el cual nos comunicamos los seres humanos, con permiso del medio visual. Así que es natural que si pretendemos crear máquinas que emulen/simulen la inteligencia humana, debemos enseñarlas a entender y expresarse en nuestros lenguajes, definiendo estos como sistemas de símbolos y reglas utilizados para representar y comunicar ideas, emociones, conceptos o información entre entidades.

Antes de la década de 1980, los ordenadores eran muy diferentes de los que conocemos hoy. Eran extremadamente costosos y ocupaban grandes superficies. Si una empresa quería adoptar las "nuevas tecnologías", debía adquirir un ordenador central de gran tamaño, conocido como *mainframe*, y conectar múltiples terminales, incluyendo terminales remotas. Estos terminales consistían en un monitor, que únicamente mostraba el texto recibido, y un teclado, que se limitaba a enviar texto.

160 Vector de números naturales (byte, word, etc), de números enteros (byte, word, etc.) o números en punto flotante (dword, qword).

161 El tipo se podría codificar en un byte: 0 un byte, 1 word, 2 dword, 3 qword, etc.

Nada más¹⁶². Estaban conectados al ordenador central por medio de una conexión por cable¹⁶³. Las aplicaciones estaban y se ejecutaban en el ordenador central de esta manera:

1. Todo terminal al encenderse mandaba una secuencia al ordenador central que, tras recibirla, iniciaba un proceso¹⁶⁴ preestablecido. Típicamente los ordenadores centrales eran multiproceso y multiusuario.
2. Cuando el usuario pulsaba una tecla en el terminal, su código numérico se enviaba al ordenador central y, si era diferente a enter/intro, la almacenaba en una memoria temporal. Un carácter después de otro.

DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo
32	20h	espacio	64	40h	@	96	60h	'
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	\$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	.	71	47h	G	103	67h	g
40	28h	(72	48h	H	104	68h	h
41	29h)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	:	91	5Bh	[123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	
61	3Dh	=	93	5Dh]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	-			

Figura 46: Tabla ASCII (parcial) de 7 bits, del carácter 32 al 126.

3. Cuando el usuario pulsaba la tecla enter/intro y el código llegaba al ordenador central, este último interpretaba que la secuencia de caracteres había llegado a su fin y que lo que tenía almacenado en memoria era una orden que debía ser ejecutada.

162 Coloquialmente las llamamos terminales “tontas”. Posteriormente se añadió el ratón cuando los terminales se volvieron “inteligentes” y gráficos.

163 Típicamente conexiones RS-232, por la cual se enviaban impulsos representando bits, uno detrás de otro, en serie. Si se tratase de una terminal remota, un elemento interpuesto en el medio de la comunicación (modem) se encargaba de traducir los bits a sonido y viceversa en el destino. Ver la película “Juegos de guerra”.

164 Denominado shell desde la aparición de la familia Unix, de la cual Linux es un representante destacado.

4. Y así lo hacía, ejecutaba la orden y lo que esta produjese [su salida] la enviaba de vuelta al terminal [carácter a carácter], en donde el monitor se encargaba de ir representando la secuencia que le iba llegando.

Así pues, la comunicación entre las terminales y el ordenador central eran secuencias de números de ida y secuencias de números de vuelta, debidamente codificados en un byte cada uno de ellos y enviados de bit en bit. Al principio, cada fabricante usaba diferentes codificaciones [BCD, EBCDIC, ...], hasta que un organismo de estandarización propuso una tabla de codificación para todos. De esta forma los terminales de diferentes fabricantes podrían ser usados con ordenadores centrales de, igualmente, diferentes fabricantes.

El estándar **ASCII** [ver Figura 46] es uno de los primeros estándares de **sistemas de codificación** utilizados en la informática para representar caracteres y símbolos en los sistemas digitales. Fue desarrollado en la década de 1960 y se ha convertido en un estándar ampliamente aceptado en la mayoría de los sistemas informáticos.

Nota: Cuando hablamos de caracteres o símbolos nos referimos generalmente a todo lo imprimible: caracteres alfábéticos, dígitos, símbolos de puntuación, y cualquier otro elemento que forma parte de un texto, independientemente del lenguaje. Algunos caracteres son por naturaleza no imprimibles, por ello reciben el nombre de caracteres de control.

El sistema ASCII original utilizaba una representación de 7 bits para asignar un número único a cada carácter o símbolo. El octavo bit se usaba para la corrección de errores en la transmisión. Por lo tanto, originalmente, ASCII definía 128 caracteres, que incluían letras mayúsculas y minúsculas, dígitos numéricos, caracteres de puntuación y algunos caracteres especiales como el espacio en blanco y los caracteres de control [en la Figura 46 sólo aparecen los caracteres imprimibles, del 32 al 126, los anteriores al 32 no son imprimibles¹⁶⁵]. Todo ello bajo el prisma del alfabeto anglo-ame-ricano.

La **representación ASCII** se basa en la numeración decimal, donde cada carácter tiene asignado un único número en una tabla. Por ejemplo, la letra "A" se representa con el número decimal 65 [1000001 en binario], la letra "a" con el número decimal 97 [1100001 en binario], y así con el resto.

Nota: Todo el procesamiento, tanto en ASCII como en Unicode, se lleva a cabo exclusivamente usando los números de sus respectivas tablas. Por ejemplo, si quieres convertir letras de mayúsculas a minúsculas, sólo tienes que sumar 32 al número que le corresponde a la letra. Cuando es necesario mostrar a los seres humanos estos números como los caracteres y símbolos que conocemos, entran en juego otros mecanismos que no son de importancia para nosotros en este contexto.

El código ASCII tiene limitaciones significativas, especialmente en lo que respecta a la representa-

¹⁶⁵ Por ejemplo, el carácter que ocasionaba un beep, el en terminal (7) o el retorno de carro y salto de línea (respectivamente 10 y 13), entre otros.

ción de caracteres en otros idiomas que no sea el inglés. Debido a su tamaño limitado de 7 bits, solo puede representar un conjunto limitado de caracteres y símbolos. Esto significa que los caracteres de muchos idiomas no se pueden representar adecuadamente utilizando ASCII.

A medida que la tecnología avanzaba y se extendían las necesidades de representación de caracteres internacionales (década de los 80), se hizo evidente la necesidad de un estándar más amplio y completo que pudiera abarcar una variedad más amplia de idiomas y caracteres. El primer intento fue extender la tabla ASCII, pasando de 7 bits por carácter a 8 bits por carácter¹⁶⁶. Esto incrementó el tamaño de la tabla de 128 a 256 posibles caracteres. Pero obviamente, aunque fue un avance, no era suficiente.

Fue en este contexto en el que surgió **Unicode**.

8.2.1 Unicode

El sistema de codificación Unicode ha revolucionado la forma en que los caracteres son representados en los sistemas informáticos, brindando una solución integral y global para la representación de textos de diferentes idiomas y escrituras. A diferencia de los estándares de codificación anteriores, como ASCII, que solo permitían representar un conjunto limitado de caracteres en inglés, Unicode abarca una amplia gama de caracteres de prácticamente todos los idiomas escritos.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+100x	ଠ	ଥ	ନ	ବ	ଚ	ଫ	ଶ	ଶ	ଷ	ଷ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+101x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+102x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+103x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+104x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+105x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+106x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+107x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+108x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ
U+109x	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ	ଙ

Figura 47: Vista muy parcial de un bloque de caracteres Unicode.

166 Gracias a que mejoró muchísimo los medios de transmisión y apenas se cometían errores mientras se enviaban los caracteres. También ayudaron las técnicas de corrección de errores.

Definición y concepto básico de Unicode

Unicode es un estándar de codificación de caracteres diseñado para abarcar todos los sistemas de escritura del mundo y los símbolos utilizados en diferentes idiomas. Su objetivo principal es proporcionar un único punto de referencia para representar y manipular texto en diferentes plataformas y sistemas informáticos. **Unicode asigna a cada carácter un número único** llamado punto de código, que se usa para representarlo.

El sistema de codificación Unicode se basa en una tabla de caracteres, igual que ASCII, los cuales pueden representar desde caracteres comunes utilizados en el inglés hasta caracteres específicos de idiomas menos conocidos. Además, Unicode también incluye símbolos matemáticos, iconos, emojis y muchos otros elementos gráficos.

Diferencias entre ASCII y Unicode

La principal diferencia entre ASCII y Unicode radica en su capacidad para representar caracteres. Mientras que ASCII sólo puede representar 128 (o 256) caracteres diferentes, incluyendo letras en inglés, dígitos y algunos símbolos básicos, Unicode tiene una capacidad mucho mayor, con 1.114.112 códigos disponibles. De hecho no usa todos, en su versión 15 Unicode engloba todos los caracteres de uso común en la actualidad, 149.186 puntos de código, poco más de un 13% de su capacidad total.

Otra diferencia clave es que, mientras ASCII utiliza un único byte (8 bits) para representar cada carácter, Unicode usa, como máximo, 4 bytes por carácter. ¿Por qué “como máximo”? Idealmente un carácter Unicode se almacena como un número de 4 bytes; debido a esto un texto en Unicode ocupa mucho y una gran parte de estos bits serán cero la mayor parte de las veces. Por eso se pueden usar varias codificaciones¹⁶⁷. Este hecho será transparente para nosotros.

Beneficios y desafíos de Unicode en el análisis de texto

La adopción de Unicode en el análisis de texto brinda varios beneficios importantes. En primer lugar, permite el procesamiento y análisis de textos en **diferentes idiomas** y escrituras sin requerir múltiples estándares de codificación. Esto simplifica en gran medida el desarrollo de aplicaciones de NLP multilingües y mejora la interoperabilidad entre sistemas.

Además, Unicode facilita el intercambio de texto entre **diferentes plataformas** y sistemas informáticos, ya que los caracteres se representan de manera consistente en todos los sistemas que admiten Unicode. Esto es especialmente valioso en entornos donde la colaboración y el intercambio de información se realizan a través de diferentes dispositivos y sistemas operativos.

Sin embargo, también existen desafíos asociados con Unicode en este contexto. Dado que Unicode admite una amplia gama de caracteres y símbolos, el **tamaño del vocabulario** se vuelve significativamente más grande. Esto puede requerir un mayor procesamiento y almacenamiento, especialmente cuando se trabaja con grandes volúmenes de texto.

¹⁶⁷ A la hora de almacenarse en memoria o en un fichero, los textos en Unicode se codifican de forma tal que ocupan entre 1 y 4 bytes (utf8) o entre 2 y 4 bytes (utf16). Utf32 existe y codifica cada carácter como 4 bytes.

También, el análisis y procesamiento de texto en idiomas menos conocidos o con sistemas de escritura complejos pueden presentar desafíos adicionales. La correcta segmentación de palabras y la asignación de etiquetas gramaticales pueden ser más difíciles en idiomas con estructuras lingüísticas diferentes. Además es necesario contar con recursos lingüísticos adecuados, como modelos de lenguaje y herramientas de procesamiento, para trabajar con estos idiomas menos comunes.

Nota: En adelante supondremos que trabajamos en ASCII, por simplicidad y debido a que las herramientas ocultan las complejidades de Unicode. Si fuese necesario indicaríamos que estamos usando Unicode explícitamente.

8.2.2 El texto y su representación

Ya sabemos como codificar símbolos y caracteres, incluso los no imprimibles, de manera individual, pero ¿cómo codificar texto? Un texto, visto desde este punto de vista, son secuencias de números, codificando cada uno de ellos un elemento Unicode.

Texto	H	o	l	a	m	u	n	d	o
Códigos	72	111	108	97	32	109	117	110	100
⁽¹⁶⁸⁾	48	6F	6C	61	20	6D	75	6E	6F

Esta tabla nos muestra como se almacena en memoria el texto “Hola mundo”. Como una secuencia de números, de 1 byte cada uno de ellos. Si el texto es más largo, la secuencia es más larga¹⁶⁹. Desde el punto de vista pedagógico podemos pensar que 1 carácter = 1 byte.

8.3 Codificando imágenes

En el capítulo Capítulo 4 hicimos una introducción a qué es una imagen y esbozamos como se codificaba. Aquí vamos a formalizarlo.

Desde el punto de vista tensorial una imagen digital se puede representar como un tensor, que es una estructura de datos multidimensional, como ya hemos visto. En su forma más común, una imagen en escala de grises se representa como un **tensor bidimensional**, con dos dimensiones que corresponden a las filas y columnas de píxeles. Cada elemento del tensor almacena un valor de intensidad que representa la luminosidad de un píxel en la imagen.

168 Esta fila está en código hexadecimal.

169 La suma del texto de todas las wikipedias en castellano, gallego, vasco, catalán y portugués (iberian corpus) ocupa más de nueve mil millones de caracteres (en concreto 9.313.797.006 en utf8, 8,7 GiB). GPT3 (y por tanto ChatGpt) fue entrenado con secuencias de texto que suman más de 626,921,403,881,472 de caracteres (570 terabytes). El texto de ejemplo (“Hola mundo”) ocupa 10 caracteres.

```

1 import numpy as np
2 import imageio
3
4 # Leer la imagen en un array numpy
5 imagen = imageio.imread('imagen_rgb.png')
6
7 # Mostrar la forma (shape) del array
8 print(imagen.shape)

```

Para una imagen en color, se utiliza un tensor de rango 3, donde las dos primeras dimensiones corresponden a las filas y columnas de píxeles, y la tercera dimensión corresponde a los canales de color (generalmente RGB, aunque no en exclusiva como hemos visto anteriormente). Así, una imagen en color se representa como un tensor tridimensional, donde cada canal de color tiene su propio conjunto de valores de intensidad.

En el código anterior, podemos ver en la línea 5 cómo leemos un fichero que contiene una imagen en RGB y seguidamente [línea 8] mostramos la forma¹⁷⁰ [shape] del tensor. La variable `imagen` contiene un tensor y a partir de esa línea podemos manipular la imagen como queramos, desde cambiar un pixel a hacer una segmentación del contenido [§5.4].

8.4 Codificando sonido

El sonido es una onda que se genera por la vibración de un objeto y se propaga a través de un medio, como el aire, el agua o un sólido. Estas vibraciones causan cambios en la presión del medio circundante, que se transmiten como ondas longitudinales, donde las partículas del medio oscilan en la misma dirección en que se propaga la onda. Al ser una onda, el sonido posee las siguientes características: frecuencia, amplitud, velocidad de propagación y longitud de onda.

Para codificar sonido de una fuente debemos capturar los cambios de presión y convertirlos, primero, a señales eléctricas. Eso es lo que hace un micrófono. Seguidamente estas señales eléctricas son digitalizadas y convertidas a secuencias de números.

El proceso completo es el siguiente:

1. **Captura del sonido:** un micrófono recoge los cambios de presión y los convierte a una señal analógica.
2. **Digitalización:** se lleva a cabo en dos subetapas.
 1. En el proceso de **muestreo** se toman mediciones de la señal analógica a intervalos regulares de tiempo. La frecuencia de muestreo [tasa de muestreo] es el número de muestras tomadas por segundo y se mide en Hertz [Hz].
 2. A cada muestra tomada se le asigna a un valor numérico. Este proceso se llama **cuantización**. La precisión de la cuantización depende del número de bits utilizados para representar cada muestra. Por ejemplo, en una cuantización de 16 bits, hay 65,536 nive-

¹⁷⁰ Sus dimensiones: alto, ancho, planos de color (en este caso 3, al ser RGB).

les posibles para representar la amplitud de la señal.

3. **Codificación:** Una vez que la señal ha sido digitalizada, los datos resultantes pueden ser codificados en varios formatos, dependiendo de la aplicación. Los formatos de audio digital comunes incluyen WAV, MP3, AAC, entre otros.

Finalmente el sonido se representa como una secuencia de números con los datos añadidos de frecuencia de muestreo y tipo [cuantización, números de bits usados para cada muestra].

De nuevo, el sonido digitalizado puede almacenarse como tensores:

```
1 import numpy as np
2 from scipy.io import wavfile
3
4 # Leer un archivo de sonido .wav
5 rate, data = wavfile.read('audio.wav')
6
7 # 'data' es un array de NumPy que contiene las muestras de audio
8 print(data.shape)
```

En el código anterior, podemos ver en la línea 5 cómo leemos un fichero que contiene audio y seguidamente [línea 8] mostramos la forma [shape] del tensor. La variable *data* contiene un tensor y *rate* la frecuencia de muestreo, a partir de esa línea podemos manipular el audio como queramos.

Si comparas este código con el anterior, verás similitudes. Pero lo importante es que tanto el audio como las imágenes [y los vídeos] las podemos codificar como tensores. Si añadimos que el aprendizaje profundo [redes neuronales] tienen como entrada y salida tensores, podemos ir intuyendo que **es posible usar la misma herramienta matemática para diferentes medios**.

Pero ¿y el texto? No hemos dicho nada de texto y tensores ¿también se codifica en tensores?

8.5 Procesamiento del Lenguaje Natural

El procesamiento del lenguaje natural [NLP en sus siglas en inglés] es una rama de la inteligencia artificial que se enfoca en la interacción entre las computadoras y el lenguaje humano, permitiendo a las máquinas interpretar, generar y responder al lenguaje de forma que emule la capacidad de comprensión y producción lingüística humana.

En el campo del NLP, el texto es uno de los elementos fundamentales que se analiza y procesa para extraer información, comprender su significado y realizar diversas tareas lingüísticas. El texto, ya sea en forma de documentos, mensajes, publicaciones en redes sociales o cualquier otra forma de expresión escrita, se convierte en una valiosa fuente de datos que puede ser aprovechada para obtener conocimiento y tomar decisiones.

8.5.1 Tokens y tokenización

En NLP, los *tokens* (nota¹⁷¹) juegan un papel fundamental. Un *token* es una **unidad discreta de texto**, que puede ser una palabra, un carácter, una sub-palabra o cualquier otra unidad que se elija para dividir el texto de manera significativa. La *tokenización* es el **proceso de dividir el texto** en *tokens*, lo que facilita el análisis y la comprensión del lenguaje.

Los *tokens* son unidades fundamentales en el análisis del lenguaje natural. Al dividir el texto se obtiene una representación estructurada que permite aplicar técnicas y algoritmos para extraer información y comprender mejor el contenido. Los *tokens* pueden ser palabras individuales, símbolos de puntuación, caracteres individuales o incluso subpalabras, dependiendo del nivel de granularidad deseado.



Ilustración 5: Figura 58: El machine learning está de moda, pero ¿cuándo estará de moda el human learning?

Fuente: Midjourney; propiedad del autor.

La tokenización requiere un **vocabulario**, una lista de todos los tokens posibles (pongamos n), numerados del 0 a $n-1$. Cuando tokenizamos un texto, pasamos de tener una lista de caracteres a una lista de *tokens*, identificados cada uno de ellos por el número que le corresponde en el vocabulario.

Nota: por tanto los algoritmos de NLP no procesan texto como tal, ni caracteres ni palabras. Procesan números, donde cada uno de ellos hace referencia a una entrada en un vocabulario de tokens.

La *tokenización* es un paso crucial en el preprocesamiento del texto, ya que proporciona una estructura adecuada para realizar tareas posteriores, como el análisis sintáctico, la clasificación de texto o la generación de texto.

171 Puedes traducir “token” por fragmento y “tokenización” por fragmentación. Uso la versión inglesa porque es ampliamente usada en la literatura científica y porque más adelante necesitamos el concepto natural de fragmento.

8.5.2 Estrategias de tokenización

Existen varios enfoques para la *tokenización*, y la elección del enfoque depende del idioma, el dominio y el objetivo del análisis. A continuación, se presentan algunos enfoques comunes de *tokenización*:

División por palabras: Este enfoque consiste en dividir el texto en *tokens* individuales basados en los espacios en blanco y la puntuación. Cada palabra se considera un *token* separado, lo que facilita el análisis léxico y la comprensión del texto en función de las unidades semánticas.

Es sencillo de utilizar e implementar, pero necesita vocabularios enormes¹⁷². Además, la *tokenización* basada en palabras puede presentar desafíos en idiomas con formas flexivas, como el español o el alemán, donde una sola palabra puede tener múltiples formas según el contexto. En tales casos, se requiere un análisis morfológico adicional para dividir correctamente el texto en *tokens*.

Tokenización de caracteres: En lugar de dividir el texto en palabras, este enfoque considera cada carácter individual como un *token*. La tokenización de caracteres es útil en escenarios donde se requiere un nivel más granular de análisis, como el reconocimiento de voz o la generación de caracteres en lenguaje natural.

La *tokenización* de caracteres puede ser especialmente útil en idiomas sin espacios entre palabras, como el chino o el japonés, donde los caracteres individuales contienen información semántica significativa. También es útil en procesamientos simples.

Tokenización en fragmentos: Este enfoque divide el texto en fragmentos de palabras en lugar de palabras completas. Los fragmentos son segmentos más pequeños de palabras que pueden capturar mejor la estructura morfológica y las relaciones dentro de las palabras.

La *tokenización* en fragmentos es útil en idiomas compuestos por palabras largas o en dominios específicos, como la biomedicina o la tecnología, donde las palabras pueden contener terminología especializada.

Por ejemplo: vamos a *tokenizar*¹⁷³ mediante fragmentos “*En el procesamiento del lenguaje natural, los tokens juegan un papel fundamental.*”, arrojando el siguiente resultado:

En	el	pro	##ces	##ami	##ent	##o	del	le	##ng	##ua	##je	natural	,
los	token	##s	j	##ue	##gan	un	p	##ap	##el	fundamental	.		

Una lista de 26 elementos. La secuencia “##” indica que este *token* va pegado a su izquierda a otro *token* que no es un separador¹⁷⁴. De esta manera “el” y “##el” son *tokens* diferentes. El vocabulario del *tokenizador* usado posee menos de treinta mil *tokens* diferentes posibles.

¹⁷² Sólo la wikipedia en gallego posee más de un millón y medio de palabras distintas. Por tanto si usamos este enfoque tendríamos un vocabulario enorme.

¹⁷³ He usado el modelo preentrenado “distilbert-base-cased” y la librería transformers.

¹⁷⁴ En este caso un separador es cualquier carácter no alfabético.

8.5.3 Consideraciones de la tokenización

La tokenización no es un proceso trivial, y existen consideraciones que deben abordarse para obtener resultados precisos y coherentes. Algunas de ellas son:

- **Puntuación:** Decidir si se incluye o se excluye la puntuación en los *tokens* puede variar según la tarea y el objetivo del análisis. En algunos casos, la puntuación puede contener información relevante, mientras que en otros puede considerarse ruido.
- **Mayúsculas y minúsculas:** Determinar si se conserva o se normaliza la información de capitalización puede afectar el análisis. En algunos casos, se desea distinguir entre mayúsculas y minúsculas, mientras que en otros se consideran equivalentes.
- **Stop words:** Las *stop words* son palabras comunes y no informativas, como "el", "y", "a", que a menudo se eliminan durante la tokenización para reducir el *ruido* y mejorar el rendimiento del análisis.
- **Idioma y dominio:** La tokenización puede variar significativamente según el idioma y el dominio. En algunos idiomas, las reglas morfológicas y gramaticales son más complejas, lo que exige un análisis más profundo para lograr una tokenización precisa. Asimismo, en dominios especializados, es necesario tener en cuenta particularidades específicas para segmentar correctamente los términos técnicos y garantizar una adecuada comprensión del contenido.

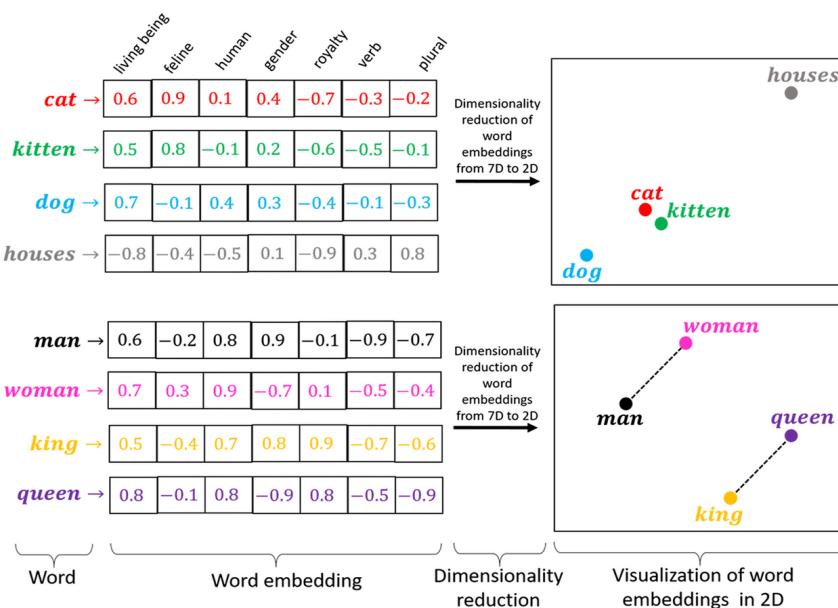


Figura 48: Ejemplo de un proceso de embedding sobre unas tokenización basada en palabras.

Fuente: https://miro.medium.com/v2/resize/f1:1400/1*sAJdxEsDjsPMioHyzlN3_A.png

8.6 Vectores de características

Trabajar con *tokens* – sean estos caracteres, palabras o fragmentos – es muy útil y una técnica muy usada en NLP, pero tienen un problema cuando se usan en determinadas arquitecturas. Al representar cada *token* con un número [su posición en el vocabulario], estamos introduciendo una información falsa: estamos afirmando que el *token* con número k es mayor que el *token* con número q , sólo porque k es un número mayor que q . Dependiendo que arquitectura usemos¹⁷⁵, esta será tan sensible que llegará sola a esa conclusión artificial y no deseada.

Para arquitecturas mayores y técnicas avanzadas se usa una técnica denominada *embeddings* [imbuido, incrustado, también conocido como **vectores de características**], que consiste en traducir cada *token* de un vocabulario en una representación vectorial [tensorial de una dimensión]. Esta técnica, además de quitarnos el problema de *tokens* con números, nos entrega la posibilidad de que *tokens* de significado parecido estén cerca en dicho espacio de n dimensiones.

En la Figura 48 podemos ver un ejemplo, al convertir *tokens*/palabras en vectores de 7 componentes [o dimensiones si pensamos en el espacio vectorial], siendo cada componente [dimensión] una característica que cumple o no el *token*/palabra en cierto grado. El número asignado para cada palabra en cada categoría indica el grado de pertenencia de dicha palabra en dicha categoría [mira la última columna “plural” y los valores de cada palabra, al única en plural es “houses” con 0.8, el resto no sube de 0]. La figura también hace una proyección para representar los vectores sobre una gráfica de dos dimensiones con las siguientes características:

- Conceptos semejantes [“gato” *cat*, “gatito” *kitten*] están cerca y conceptos sin apenas relación están lejos. Además, “perro” [dog] está más cerca de “gato” [cat] y “gatito” [kitten] que de “casas” [houses].
- La pareja “hombre” [man] y “mujer” [woman] están entre ellas a una distancia semejante a “rey” [king] y “reina” [queen].

Esto significa que usando la técnica de *embedding* adecuada, se pueden calcular una representación vectorial de los tokens que – en cierta manera – ha *entendido* la relación entre ellos. Por ejemplo, “rey” - “hombre” + “mujer” \approx “reina”.

Existen varias técnicas y modelos para calcular los *embeddings* a partir de grandes cantidades de texto no etiquetado. Pero sin duda la técnica más usada es **Word2Vec** [*word to vector*].

Word2Vec es un modelo de *embeddings* basado en aprendizaje automático¹⁷⁶. Este modelo mapea las palabras en un espacio vectorial de alta dimensión, donde palabras similares en contexto se encuentran más cercanas en dicho espacio vectorial. Esto permite realizar operaciones algebraicas entre los vectores, como la resta o la suma, para capturar relaciones semánticas entre palabras

175 Por ejemplo, todos los LLM. Y una buena parte – si no todos – de los algoritmos de aprendizaje automático con base estadística.

176 Típicamente una red neuronal artificial de dos capas ocultas.

FastText es una extensión del modelo Word2Vec que también incorpora información de subpalabras [fragmentos]. En lugar de considerar las palabras como entidades indivisibles, FastText descompone las palabras en fragmentos más pequeños y obtiene sus *embeddings*. Esto permite capturar información semántica adicional, especialmente para palabras raras o fuera del vocabulario. FastText es útil para el análisis de textos en idiomas con morfología rica, donde las palabras se forman mediante la combinación de múltiples raíces y sufijos.

Word2Vec [y FastText] necesitan ser entrenados previamente con todos los tokens del vocabulario [sean palabras o fragmentos]. Veamos un ejemplo [muy simple]:

Cap. 9 - Tokenización.ipynb

```

1  from gensim.models import Word2Vec
2
3  # Oraciones de entrenamiento (cada oración es una lista de palabras)
4  sentences = [
5      ["gato", "animal", "felino"],
6      ["perro", "animal", "canino"],
7      ["gato", "maúlla"],
8      ["perro", "ladra"],
9      ["pájaro", "animal", "vuela"],
10     ["pez", "animal", "nada"]
11 ]
12
13 # Entrenar el modelo Word2Vec
14 model = Word2Vec(sentences, vector_size=10, window=3, min_count=1, workers=1)
15
16 # Acceder al vector de una palabra
17 vector_gato = model.wv['gato']
18 print(vector_gato)
19
20 # Encontrar palabras más similares a "gato"
21 similar_words = model.wv.most_similar('gato', topn=1)
22 print("Palabras similares a 'gato':", similar_words)
23
24 # Calcular la similitud de coseno entre dos palabras
25 similarity = model.wv.similarity('gato', 'felino')
26 print("Similitud entre 'gato' y 'felino':", similarity)
27
28 # Realizar operaciones aritméticas con los vectores de palabras
29 # (gato - felino + cánino)
30 result = model.wv.most_similar(positive=['gato', 'canino'], negative=['felino'], topn=1)
31 print("Resultado de gato - felino + canino:", result)

```

con el siguiente resultado:

```

[ 0.0731 0.0507 0.0675 0.0076 0.0635 -0.0340 -0.0094 0.0576 -0.0752 -0.0393]
Palabras similares a 'gato': [('felino', 0.4705304205417633), ]
Similitud entre 'gato' y 'felino': 0.47053045
Resultado de gato - felino + canino: [('ladra', 0.10063980519771576)]

```

La primera línea del resultado es el vector de 10 componentes que identifica al token “gato”; el siguiente resultado nos muestra que con esta codificación, el vector que más cerca está de “gato” es “felino”; el siguiente nos muestra la similitud del coseno de ambos *tokens*; y finalmente podemos ver el resultado de efectuar [con vectores] la operación “gato” – “felino” + “canino” que es “ladra”. Al proporcionar una representación numérica del significado de las palabras, los *embeddings*

permiten que los modelos de NLP generalicen mejor y capturen las sutilezas del lenguaje.

Es importante recalcar que los *embeddings* se pueden preentrenar utilizando grandes *corpus* de texto no etiquetado, para luego utilizarse como características en tareas específicas o se pueden ajustar durante el entrenamiento de un modelo para una tarea en particular. Esta capacidad de transferencia de conocimiento es especialmente útil cuando se dispone de pocos datos etiquetados para una tarea específica.

RETOS DEL CAPÍTULO 8

1. ¿Por qué es importante considerar el texto como dato en el procesamiento del lenguaje natural?
2. Escoge una cadena de texto y, usando una tabla ASCII que puedes encontrar en la web, codifícala en números. Procura que el texto tenga acentos o la letra *ñ*.
3. Repite en reto anterior, pero en Unicode.
4. Pídele a tu IA favorita que genere el código necesario para convertir un texto dado en números.
5. ¿Qué es la tokenización y por qué es esencial en el procesamiento del lenguaje natural?
6. Imagina que tenemos un texto arbitrariamente largo y queremos dividirlo en frases. Todas acabadas en puntos. Razona ¿que problemas tendríamos si queremos un resultado exacto?
7. Establece un diálogo con tu agente conversacional acerca de las expresiones regulares. ¿Qué son? ¿para qué sirven? Pídele que te ponga un código de ejemplo que lo muestre.
8. Pídele a tu IA favorita, que genere un código que a partir de un texto grande – que puede generar la misma IA – con el objeto de hacer una tabla de frecuencias de las palabras que contiene, y las muestre de más frecuentes a menos frecuentes.
9. Un bigrama es una secuencia de dos letras, por ejemplo: los bigramas de “Hola mundo” son “Ho”, “ol”, “la”, “mu”, “un”, “nd”, “do”. Los trigramas se crean de forma similar.
¿Podríamos deducir en qué lenguaje está escrito un texto, sólo viendo las frecuencias de sus bigramas? Discútelo con tu IA favorita. ¿Mejoraría si usásemos trigramas?
10. Pídele a tu agente conversacional favorito, que genere el código necesario para crear las bolsas de palabras (*bag of words*) de frases determinadas. Ponlas a prueba con textos no pequeños.
11. Pregúntale a tu IA favorita, qué es el TF-IDF de un texto.
12. Aprende más de *word2vec* pidiéndole a tu IA favorita que te asesore al respecto. ¿Qué usos tiene? ¿Cómo calcula los vectores de características?
13. ¿Qué es la extracción de entidades y cuál es su importancia en el procesamiento del lenguaje natural?
14. Habla con tu IA favorita: ¿Cómo se pueden usar las técnicas de procesamiento del lenguaje natural en la búsqueda de información?
15. ¿Cuál es el papel de las redes neuronales en el procesamiento del lenguaje natural?

Capítulo 9

DE AUDIO A TEXTO Y DE NUEVO A AUDIO

9.1 Reconocimiento de voz y transcripción automática

El reconocimiento de voz y la transcripción automática son dos tecnologías fundamentales en el campo del Procesamiento del Lenguaje Natural. El **reconocimiento de voz** se refiere al proceso de convertir la información hablada en texto, mientras que la **transcripción automática** implica transformar un archivo de audio o una señal de voz en un texto escrito. Estas tecnologías son ampliamente utilizadas en muchas aplicaciones, aunque probablemente como tecnología de segundo plano.

El **reconocimiento de voz** se basa en una serie de algoritmos y modelos que permiten identificar y transcribir las palabras habladas con precisión. Uno de los enfoques más comunes es el uso de modelos acústicos y de lenguaje. El modelo acústico se encarga de capturar las características acústicas de la señal de voz y relacionarlas con los fonemas o unidades de sonido del lenguaje. El modelo de lenguaje, por otro lado, se encarga de predecir las secuencias de palabras más probables en un contexto dado.

En el reconocimiento de voz, los algoritmos de aprendizaje automático, como las redes neuronales artificiales, son ampliamente utilizados. Estos modelos se entrena con grandes conjuntos de datos de voz y texto transcritos, lo que les permite aprender a reconocer patrones y hacer predicciones más precisas sobre la transcripción del habla. La precisión del reconocimiento de voz ha mejorado significativamente a raíz de los avances en el aprendizaje profundo.

La **transcripción automática**, por otro lado, se refiere al proceso de convertir un archivo de audio a texto. Este proceso generalmente implica utilizar técnicas de reconocimiento de voz mencionadas anteriormente. Una vez que se ha realizado el reconocimiento de voz, se pueden aplicar técnicas de post-procesamiento para mejorar la calidad de la transcripción. Esto incluye el uso de algoritmos de corrección de errores, detección y corrección de palabras mal transcritas, y otros enfoques para mejorar la coherencia y legibilidad del texto transcrit.

En este capítulo vamos a hablar sobre estas tecnologías inteligentes, que aún tienen recorrido en el campo de la investigación. Estas tecnologías todavía presentan limitaciones. Una de ellos es la variabilidad del habla, que incluye diferentes acentos, entonaciones, velocidades de habla y ruidos de fondo. Estos factores pueden dificultar la precisión de la transcripción. Además, las palabras o frases ambiguas pueden llevar a errores en la transcripción, lo que requiere de técnicas adicionales para la desambiguación. Otra es la capacidad de independizar el habla de diferentes interlocutores, aunque este campo ya tenemos buenos resultados.

9.2 Síntesis de voz

La síntesis de voz se refiere al proceso de generar una salida de voz artificial a partir de texto escrito. Es una tecnología que ha avanzado significativamente en los últimos años, impulsada por el desarrollo de modelos de aprendizaje profundo y redes neuronales.

Existen diferentes enfoques utilizados en la síntesis de voz, cada uno con sus propias características y técnicas asociadas:

Síntesis de voz concatenativa: Este enfoque se basa en la grabación de fragmentos de voz de un hablante humano y luego se concatenan para formar palabras y oraciones. Se identifican fragmentos de voz que mejor se ajusten al contexto y se combinan de manera fluida para generar la salida de voz. Esta técnica requiere una gran cantidad de datos grabados y puede tener dificultades para adaptarse a nuevos contextos y palabras que no se encuentren en el conjunto de entrenamiento.



Video 12: Grabación de audio generado por medio de la síntesis de voz concatenativa

Síntesis de voz basada en formantes: Este enfoque utiliza modelos acústicos que generan sonidos vocales basados en la manipulación de los formantes, que son las frecuencias resonantes del tracto vocal humano. Los formantes se ajustan para representar diferentes sonidos vocales y se combinan para formar palabras y oraciones. Esta técnica es efectiva para producir voces sintéticas claras y precisas, pero carecen de naturalidad y expresividad.

Síntesis de voz basada en unidades: Este enfoque divide el habla en unidades más pequeñas, como fonemas o sílabas, y luego las combina para generar la salida de voz. Las unidades se almacenan en una base de datos y se seleccionan y concatenan dinámicamente según el texto de entrada. Esta técnica permite una mayor flexibilidad y adaptabilidad, ya que las unidades se pueden reutilizar y combinar de diferentes maneras para producir un habla relativamente natural.

Síntesis de voz basada en redes neuronales: Con los avances en el aprendizaje profundo, las redes neuronales se han utilizado cada vez más en la síntesis de voz. Las redes neuronales recurrentes, como las redes de memoria a corto plazo de larga duración [LSTM, *Long short-term memory*] y las redes neuronales basadas en *transformers*, se han aplicado con éxito en la generación de voz. Estas redes aprenden patrones y estructuras del habla a partir de grandes conjuntos de datos de voz y son capaces de generar una salida más natural y fluida.

Me siento muy orgullosa de que toda España esté aquí

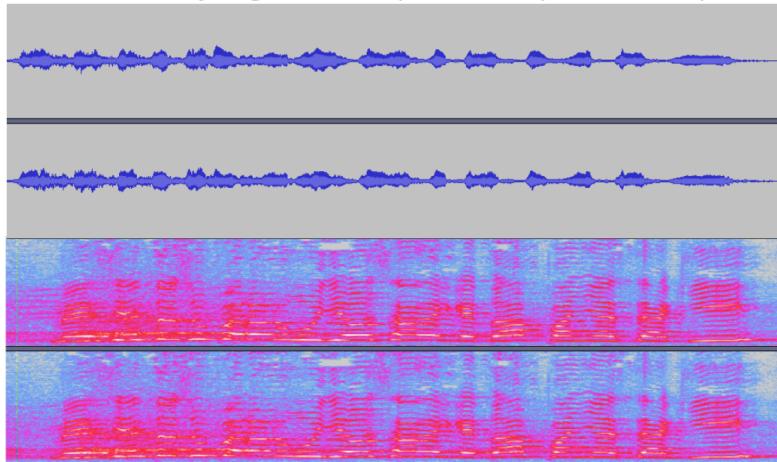


Figura 49: Texto (arriba), oscilograma de su señal acústica (debajo) y su espectrograma (abajo).

Fuente: Fragmento de audio del discurso de Jenni Hermoso, por el campeonato mundial de fútbol femenino.

Las técnicas basadas en redes neuronales para la síntesis de habla utilizando **espectrogramas**, "spectrogram-based speech synthesis" en inglés [ver Figura 49], son técnicas que involucran la generación de espectrogramas a partir de texto y luego la reconstrucción del habla a partir de estos espectrogramas.

Un espectrograma es una representación visual de la distribución de energía de una señal en el tiempo y la frecuencia. Es una herramienta útil para analizar señales de audio, vídeo y otras señales en el dominio de la frecuencia. Un espectrograma se genera dividiendo la señal en ventanas de tiempo y analizando la frecuencia de los componentes de cada ventana. El resultado es una matriz de datos que representa la intensidad de la señal en cada frecuencia y en cada momento. Los espectrogramas se suelen representar como imágenes en las que la intensidad de la señal se representa por colores. El eje horizontal representa el tiempo, el eje vertical representa la frecuencia y la intensidad se representa por el color [ver Figura 49].

Volviendo a la síntesis de voz usando redes neuronales artificiales, aquí se describen los principales pasos involucrados en este enfoque:

- **Codificación de texto a un espectrograma:** El texto original se pasa a una representación intermedia que contiene información sobre las características acústicas y prosódicas del

habla. Esta representación intermedia son espectrogramas, creados por medio de una **red generativa**¹⁷⁷. Esta red se entrena para aprender la distribución de los espectrogramas en un conjunto de datos de entrenamiento y es capaz de generar espectrogramas que se asemejan a los reales.

- **Síntesis de habla a partir del espectrograma:** Una vez que se ha generado el espectrograma, se utiliza un algoritmo de síntesis de habla para reconstruir la forma de onda de audio a partir del espectrograma. Este algoritmo puede ser una síntesis de reemplazo por ejemplo, que utiliza bases de datos de unidades de habla pre-grabadas o un método de síntesis basado en *vocoder*, que utiliza algoritmos de estimación y manipulación de características espectrales.

Mi IA favorita dice: [...] un vocoder actúa como el componente que convierte las representaciones simbólicas de fonemas o rasgos lingüísticos en ondas de sonido que podemos escuchar. Normalmente, un modelo generativo crea una especie de "esqueleto" temporal de cómo debería sonar el habla en términos de tono, inflexión, y ritmo. Luego, este esqueleto se pasa a través del vocoder para generar audio que suena como una voz humana. Los vocoders modernos son a menudo muy sofisticados y pueden generar voces casi indistinguibles de la voz humana real.

- **Post-procesamiento y mejora:** El habla sintetizada puede pasar por un proceso para mejorar su calidad y naturalidad. Esto puede implicar técnicas como el suavizado de las transiciones entre los fonemas, la modificación de las características prosódicas y la aplicación de técnicas de cancelación de ruido.

Estas técnicas basadas en redes neuronales han demostrado ser efectivas para la síntesis de habla y han permitido avances significativos en la generación de habla natural y expresiva. Al utilizar espectrogramas como representación intermedia, se logra capturar tanto las características espectrales como las prosódicas del habla, lo que conduce a una síntesis de habla más realista y de alta calidad.

9.3 Transcripción de audio

En la transcripción de audio a texto, partimos de un fichero contenido el audio ejecutamos los procesos necesarios para que lea el sonido en forma de tensor y lo transcriba a texto, reconociendo el habla i hablas de personas.

¹⁷⁷ Arquitectura de redes neuronales artificiales, entrenadas para generar "algo" realista, partiendo de ejemplos reales. Hablamos de ello en el apartado 7.8.

El proceso es el que sigue:

1. **Reconocimiento del habla:**

- a] **Extracción de Características:** El sistema analiza las ondas sonoras del audio para extraer características acústicas clave, como frecuencias y patrones temporales. Estas características se convierten en vectores que representan los fonemas del habla.
- b] **Modelos Acústicos y de Lenguaje:** Los modelos acústicos identifican los fonemas, mientras que los modelos de lenguaje utilizan el conocimiento previo y patrones de lenguaje natural para predecir palabras y frases más precisas.
- c] **Decodificación:** Combinando la información de los modelos acústicos y de lenguaje, el sistema decodifica las señales de audio en texto.

2. **Postprocesamiento:** Una vez obtenida la transcripción inicial se realiza un postprocesamiento para mejorar la precisión:

- a] **Corrección gramatical y ortográfica:** Se aplica para corregir errores comunes.
- b] **Puntuación y Formato:** Se agregan signos de puntuación y se da formato al texto para que sea más legible.

RETOS DEL CAPÍTULO 9

1. ¿Cuál es el propósito principal de convertir el audio en texto en el procesamiento del lenguaje natural?
2. ¿Cuál es la diferencia entre el reconocimiento de voz y la síntesis de voz?
3. Usando el *notebook* del capítulo, graba un fragmento de tu voz, o de tus amigos con su consentimiento, y transcríbelo a texto. Realiza esta actividad varias veces, con diferentes ruidos de fondo.
4. ¿Cuáles son las tecnologías clave involucradas en la conversión de audio a texto?
5. Busca en la web sitio que realicen esta transcripción [DeepL, por ejemplo] y busca errores de transcripción con ruidos de fondo o hablando atropelladamente. Prueba a transcribir un diálogo.
6. Crear una lista de palabras o frases difíciles de transcribir y discutir por qué podrían ser problemáticas para el software de transcripción.
7. Crea un grupo de discusión: cómo el lenguaje coloquial y las expresiones idiomáticas pueden afectar la precisión de la transcripción y la conversión de texto a voz. Que tu IA favorita sea parte del grupo.
8. ¿Qué papel juegan las redes neuronales en la conversión de audio a texto?
9. Usando el *notebook* del capítulo, sintetiza un texto. Prueba con más de una voz.
10. Ya existen web que te permiten clonar la voz de otras personas usando solamente un fragmento de audio. Dialoga con tu IA favorita para que te explique cómo lo consiguen. Busca que la explicación sea completa.
11. Pregúntale a tu agente conversacional favorito, cómo a partir de un vídeo en un idioma se puede generar los subtítulos. ¿Te atreves?
12. Busca sitios en la web que lleven a cabo el reto anterior.
13. ¿Cómo se pueden aplicar las tecnologías de conversión de voz a texto en la accesibilidad?
14. Desarrollar en grupo un "teatro de radio" simple donde escribáis un guion corto y luego lo convertís a un clip de audio usando herramientas de texto a voz.
15. Trabajo en grupo: debate y llega a conclusiones sobre las implicaciones éticas del uso de voces clonadas. Pros y contras.
16. ¿Cuál es el impacto potencial de la traducción automática de voz en las comunicaciones internacionales?
17. ¿Cuál es la diferencia entre la transcripción de voz en tiempo real y la transcripción de voz por lotes?

Capítulo 10

SIMILITUD ENTRE TEXTOS

Probablemente, el uso de algoritmos capaces de establecer si dos textos son similares es una de las primeras tecnologías inteligentes que se nos pueda venir a la cabeza si nombramos el procesamiento del lenguaje natural; con permiso de la traducción de textos.

El problema es el siguiente:

El Titanic, una obra maestra de la ingeniería naval de su tiempo, zarpó desde Southampton hacia Nueva York el 10 de abril de 1912, en su viaje inaugural. Promocionado como el transatlántico más grande y lujoso jamás construido y declarado "insumergible" por sus creadores, el barco era una representación palpable de los avances tecnológicos de la época. A bordo iban pasajeros de diversas clases sociales, desde millonarios hasta emigrantes que buscaban un nuevo comienzo en América. Su diseño incluía algunas de las tecnologías más avanzadas entonces disponibles, como compuertas estancas y un sistema de comunicación inalámbrica de última generación.

[...]

El hundimiento del Titanic no solo marcó una de las mayores tragedias marítimas en tiempos de paz sino que también tuvo un profundo impacto en la sociedad y la cultura. El desastre provocó cambios significativos en las regulaciones de seguridad marítima, incluyendo la adopción de patrones más estrictos para los equipos de salvamento y las comunicaciones en los barcos. Además, la historia del Titanic ha permeado la cultura popular, inspirando numerosos libros, películas y exposiciones que exploran el legado de este histórico naufragio. La fascinación mundial con su historia continúa hasta hoy, sirviendo como un sombrío recordatorio de la fragilidad humana frente a las fuerzas de la naturaleza.

Párrafo 1 [recortado]

Era la noche del 14 de abril de 1912 cuando el majestuoso Titanic, un leviatán de acero que desafía los límites de la ingeniería moderna, se encontró con su destino en las heladas aguas del Atlántico Norte. Como un palacio flotante, había zarpado de Southampton cargado de sueños y promesas, custodiando las vidas de aristócratas y aventureros, familias y solitarios, todos en busca de un futuro prometedor en las vastas tierras de América. Bajo el manto estrellado, la noche se desgarró con el crujido del metal cuando el iceberg rasgó el casco del barco, y el insumergible comenzó su agonía en las profundidades gélidas.

[...]

Al final, el Titanic se sumergió por completo, llevándose consigo historias no contadas y sueños no realizados. Este desastre marítimo no solo se convirtió en un hito doloroso de la vulnerabilidad humana ante la naturaleza, sino también en un catalizador para reformar las leyes de seguridad marítima. La tragedia del Titanic resonó en los corazones de la gente de todo el mundo, grabando su memoria en el tejido cultural de la humanidad, inspirando obras de arte, literatura y cine que buscan capturar la esencia de esa fatídica noche cuando el mar reclamó su trono frente a la arrogancia humana.

Párrafo 2 [recortado]

Para una mente humana estos dos párrafos¹⁷⁸ son muy similares y tratan sobre lo mismo [similitud semántica]: el *Titanic* y su hundimiento¹⁷⁹. Pero, ¿qué tenemos a nuestra disposición que nos permita dotar a las máquinas de la capacidad de saber como de iguales/diferentes son ambos textos?

178 Ver los textos completos en el notebook del capítulo: "10 Comparando textos.ipynb"

179 Ambos generados por ChatGPT con el prompt: "genera dos párrafos que hablen de los mismo (sobre el Titanic y su hundimiento) pero que estén redactados de forma muy diferente."

10.1 Representación de textos

Empecemos por la representación de textos, muy relacionado con el apartado 8.6, el cual es un paso fundamental para poder calcular la similitud entre ellos. Una buena representación captura las características semánticas o sintácticas relevantes de un texto, permitiendo realizar comparaciones adecuadas con otros textos. A continuación, explicaremos algunas técnicas comunes utilizadas para representar textos:

- **Representación de bolsas de palabras** [bag-of-words o BOW]: Esta técnica es una de las más simples pero ampliamente utilizadas. Consiste en construir un vector¹⁸⁰, partiendo de un texto, basado en la frecuencia de ocurrencia de las palabras en dicho texto. Primero se crea un vocabulario a partir de todas las palabras únicas, luego se cuenta la frecuencia absoluta de cada palabra en el texto. Esta representación ignora completamente el orden de las palabras y sólo considera su presencia en el texto.
- **Modelos de vectores de características** [ver el punto 8.6]: Estos modelos aprenden representaciones vectoriales de palabras que capturan el significado semántico y contextual de estas en función de su contexto de aparición en grandes *corpus* de texto. Estas representaciones permiten capturar relaciones entre palabras en un espacio vectorial. Para representar un texto, se pueden combinar los vectores de palabras individuales, por ejemplo, promediando o componiendo.
- **Modelos de representación basados en grafos**: En lugar de tratar los textos como una secuencia lineal de palabras, los modelos de representación basados en grafos consideran las relaciones sintácticas y semánticas entre las palabras en un texto. Estos modelos crean grafos donde los nodos representan palabras y las aristas capturan relaciones gramaticales o semánticas.
- **Transformadores** [transformers]: Los modelos basados en la arquitectura de los *transformers* han demostrado un rendimiento sobresaliente en tareas relacionadas con la similitud textual. Estos modelos capturan las relaciones contextuales y semánticas entre las palabras dentro de un texto mediante el **mecanismo de atención**. Al pre-entrenar estos modelos con grandes cantidades de datos [el *corpus*], se generan representaciones contextualizadas de palabras y oraciones que pueden ser utilizadas para calcular la similitud entre textos.

Cada una de estas técnicas de representación tiene sus ventajas y desventajas, y su elección dependerá del contexto y los requisitos específicos de la tarea de similitud textual que se esté abordando. Además, existen enfoques híbridos que combinan diferentes técnicas para obtener representaciones más completas y enriquecidas de los textos.

Empecemos mostrando el resultado, en forma de gráfica de frecuencias, de ambos párrafos usando BOW¹⁸¹:

180 En Python lo haremos mediante un diccionario.

181 Sólo se muestra el BOW de las palabras con frecuencia > 1.

10 Comparando textos.ipynb

Prompt:

Calcula el histogramas de frecuencias del BOW de text1 y representalo en una figura. Sólo muestra las 20 primeras palabras más frecuentes.

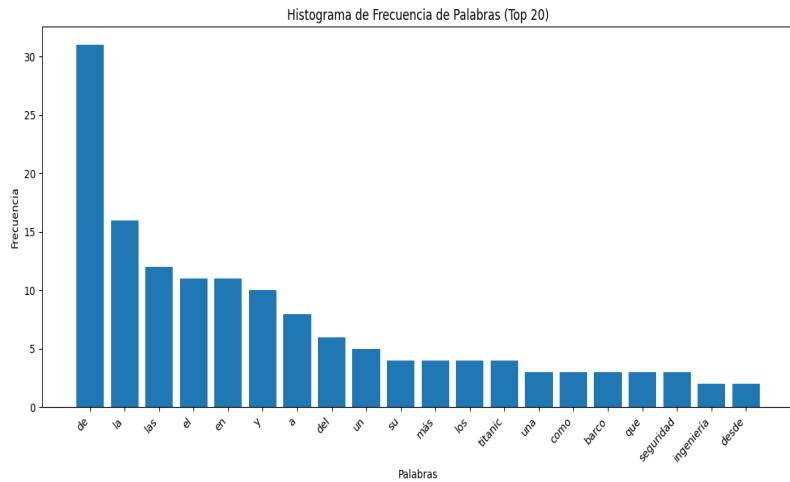


Figura 50: BOW del párrafo 1

10 Comparando textos.ipynb

Prompt:

Repítelo para el texto2.

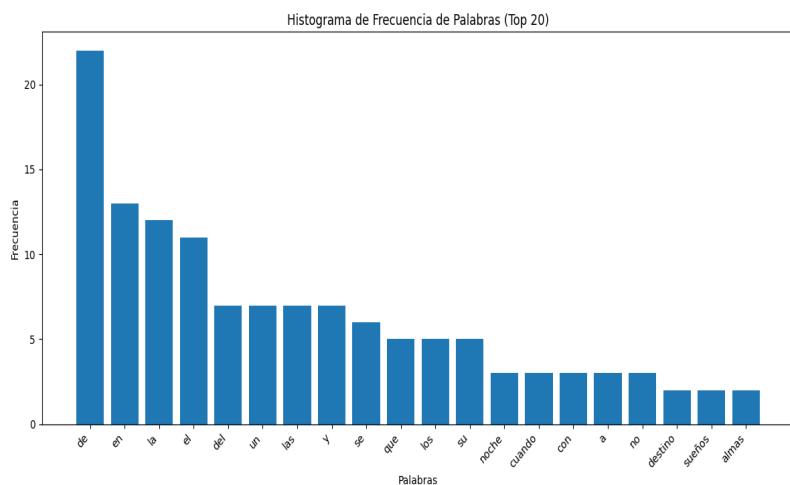


Figura 51: BOW del párrafo 2

Ambas gráficas son el equivalente de los histogramas que vimos para las imágenes, pero en el contexto del procesamiento de textos. Entonces calculábamos el numero de apariciones de un determinado pixel en la imagen; aquí calculamos el número de apariciones de una palabra [o *token*] en un texto. Ambas figuras muestran las palabras y sus frecuencias de aparición de sus respectivos textos. Por ejemplo, la palabra “de” aparece ~30 y ~25 veces respectivamente.

Atento: Observa ambos juegos de palabras. Hay algunas de ellas que aparecen en los dos y son muy frecuentes en castellano (“de”, “la”, “un”, “a”, “el”, “que”, ...) y otras que aparecen en ambos y están muy relacionadas con el suceso que describen (“titanic”, “abril”, “1912”, “iceberg”, ...). Ya tienes una pista para una forma de comprobar la similitud de ambos textos.

10.2 Modelos de similitud entre textos

Los modelos de similitud entre textos se refieren a los enfoques y técnicas utilizadas para medir la **similitud** o comparar la **proximidad semántica** entre los dos textos.

Estos modelos son fundamentales en el campo del NLP, ya que permiten abordar tareas como la recuperación de información, la agrupación de textos relacionados, la detección de plagio y la clasificación de documentos similares, tarea esta última que nos ocupa.

Existen diferentes enfoques para construir modelos de similitud textual. A continuación, describimos algunos de ellos:

- **Enfoques basados en reglas:** Estos enfoques se basan en criterios o reglas predefinidas para medir la similitud entre textos. Por ejemplo, una regla simple podría ser contar las palabras compartidas entre dos textos y utilizar ese recuento como medida de similitud. Otro enfoque es calcular la distancia de edición [como la distancia de Levenshtein] para medir cuántas operaciones de edición se necesitan para transformar un texto en otro¹⁸². Estos enfoques son relativamente simples pero pueden proporcionar resultados útiles en ciertos escenarios.

Mi IA favorita dice: Por ejemplo, la distancia de Levenshtein entre las cadenas "gato" y "gato" es 0, ya que no se requieren operaciones de edición para transformar una cadena en la otra. La distancia de Levenshtein entre las cadenas "gato" y "pato" es 1, ya que se requiere 1 operación de edición para transformar "gato" en "pato": una sustitución de la letra "g" por la letra "p".

- **Enfoques basados en características sintácticas:** Estos enfoques se centran en las características estructurales y gramaticales de los textos. Por ejemplo, se pueden utilizar medidas como la similitud de la estructura de árbol de dependencia sintáctica para comparar la es-

¹⁸² Esta métrica es útil en diversas aplicaciones como corrección ortográfica, reconocimiento de voz y comparación de ADN, entre otras. Es una forma de cuantificar la similitud entre dos cadenas de caracteres de manera que puedan compararse de una forma más objetiva.

tructura sintáctica de dos oraciones. Esto implica analizar la relación entre las palabras y su dependencia gramatical en cada oración y calcular una medida de similitud basada en estas relaciones.

- **Enfoques basados en características semánticas:** Estos enfoques se centran en la similitud de significado entre los textos. Aquí es donde entran en juego los modelos de representación de palabras, como *Word2Vec*. Estos modelos asignan vectores numéricos a palabras en función de su contexto y significado. Al calcular la similitud entre dos textos, se pueden promediar los vectores de palabras individuales para obtener una representación de todo el texto y luego calcular la similitud entre estas representaciones.

El resultado de aplicar esta estrategia [usando *Word2Vec*] a ambos párrafos fue una **similitud del coseno** de 0.841 [muy parecidos].

- **Enfoques basados en aprendizaje automático:** Estos enfoques emplean técnicas de aprendizaje automático, como algoritmos de clasificación o modelos de redes neuronales, para construir modelos de similitud textual.

El resultado de aplicar esta estrategia [usando el modelo *Bert*] a ambos párrafos fue una **similitud del coseno** de 0.806 [parecidísimos].

¿Qué métricas debemos usar para medir la similitud?

10.3 Métricas de similitud

Las métricas de similitud son herramientas usadas para cuantificar y medir el grado de similitud entre dos textos [o conjuntos de textos]. Estas métricas desempeñan un papel fundamental en tareas como la recuperación de información, la clasificación de textos y la detección de duplicados o plagios. A continuación, se describimos algunas de las métricas de similitud más comunes:

- **Similitud del coseno:** Es una métrica ampliamente utilizada para medir la similitud entre dos vectores de características, que es la forma de representar vectorialmente un texto. El enfoque consiste en proyectar cada texto en un vector de un espacio vectorial y calcular el ángulo entre ellos. Cuanto más cercano a 1 sea el valor resultante, mayor será la similitud entre los textos.
- **Distancia del coseno:** La distancia del coseno es una métrica derivada de la similitud del coseno y se define como $[1 - \text{similitud del coseno}]$. Por tanto su rango es $[0, 2]$. Cuanto más cercano de 0, menor es la distancia entre los textos.
- **Índice de Jaccard:** Esta métrica se utiliza para medir la similitud entre dos conjuntos. En el contexto del NLP, los conjuntos pueden representar la presencia de *tokens* en un texto. El índice de Jaccard se calcula dividiendo la intersección de los conjuntos entre su unión. Cuanto más cercano a 1 sea el resultado, mayor será la similitud entre los textos.
- **Distancia de Jaccard:** Al igual que la similitud del coseno, el índice Jaccard no puede ser usado para calcular una distancia, por tanto para calcular la distancia de Jaccard se usa:

- **Distancia de Levenshtein:** Esta métrica mide la similitud entre dos cadenas de caracteres mediante el cálculo del número mínimo de operaciones [inserción, eliminación o sustitución de caracteres] necesarias para convertir una cadena en la otra. Cuanto menor sea la distancia de Levenshtein, mayor será la similitud entre los textos.
- **Distancia de Hamming:** Es una métrica utilizada para medir la similitud entre dos secuencias de igual longitud. La distancia de Hamming cuenta el número de posiciones en las que los símbolos de las secuencias difieren. Cuanto menor sea la distancia de Hamming, mayor será la similitud entre los textos.

Vamos a calcular estas métricas usando diferentes técnicas. Para ellos necesitaremos un tercer texto (“texto3”) el cual trata sobre montañismo y no tiene nada que ver con “texto1” y “texto2”.

10 Comparando textos.ipynb

Prompt:

Calcula la intersección del BAG de text1 y text2, y de text1 y text3. Antes elimina las 25 palabras más frecuentes del corpus.

Dando como resultado:

Intersección de texto1 y texto2 (sin stop words): {'noche', 'sombrío', 'sino', 'américa', 'norte', '1912', 'inspirando', 'solo', 'frente', 'como', 'consigo', 'salvavidas', 'marítima', 'fria', 'abril', 'humana', '14', 'naturaleza', 'aguas', 'atlántico', 'southampton', 'todos', 'desastre', 'insumergible', 'tragedia', 'por', 'era', 'bordo', 'barco', 'ingeniería', 'gélidas', 'iceberg', 'límites', 'botes'}

Intersección de texto1 y texto3 (sin stop words): {'evento', 'durante', 'numerosos', 'diversas', 'marcó', 'solo', 'personas', 'significativos', 'sino', 'humana', 'historia', 'ha', 'sus', 'por', 'era', 'incluyendo'}

Podemos observar que la intersección del BOW de textos relacionados es más grande que de textos no relacionados. Esta afirmación es cierta en general, pero pueden “fabricarse” textos similares sin apenas palabras en común.

10 Comparando textos.ipynb

Prompt:

Calcula la distancia jaccard entre los tres textos, excepto entre ellos mismos. Elimina previamente las 25 stop words del corpus.

Las stop words son palabras comunes en un idioma, como "y", "el", "a", "de", que suelen tener poco valor informativo y, por lo tanto, se eliminan en procesos de análisis de texto y procesamiento de lenguaje natural (NLP). Su eliminación permite reducir el tamaño del texto y enfocarse en las palabras más relevantes para tareas como la similitud de documentos, entre otras.

El resultado es:

```
Distancia Jaccard entre texto1 y texto2 (sin stop words): 0.882
Distancia Jaccard entre texto1 y texto3 (sin stop words): 0.944
Distancia Jaccard entre texto2 y texto3 (sin stop words): 0.967
```

En este caso, la distancia entre el texto1 y el texto2 es menor que la distancia entre el texto1 y el texto3, y entre el texto2 y el texto3. Esto sugiere que los textos 1 y 2 son más similares entre sí en términos de las palabras que contienen, en comparación con los otros pares de textos.

El hecho de que las distancias entre los textos 1 y 3, y 2 y 3 sean cercanas a 1 indica que estos pares de textos tienen muy pocas palabras en común, lo cual es esperable dado que tratan temas diferentes [el hundimiento del Titanic y el montañismo].

Otra métrica es el coeficiente Dice:

10 Comparando textos.ipynb

Prompt:

Calcula el coeficiente Dice entre los tres textos, excepto entre ellos mismos. Elimina previamente las 25 stop words del corpus.

Con el resultado:

```
Distancia Dice entre texto1 y texto2 (sin stop words): 0.212
Distancia Dice entre texto1 y texto3 (sin stop words): 0.106
Distancia Dice entre texto2 y texto3 (sin stop words): 0.064
```

En este caso, al igual que con la distancia Jaccard, el coeficiente de Dice entre el texto1 y el texto2 es mayor que el coeficiente entre el texto1 y el texto3, y entre el texto2 y el texto3.

Esto confirma que los textos 1 y 2 son más similares entre sí en términos de las palabras que contienen, en comparación con los otros pares de textos. Los bajos valores del coeficiente de Dice entre los textos 1 y 3, y 2 y 3 refuerzan la idea de que estos pares de textos tienen muy pocas palabras en común.

Finalmente vamos a calcular la similitud del coseno entre los *embeddings* de los tres textos; pero antes, vamos a entrenar un word2vec con todo el *corpus*:

10 Comparando textos.ipynb

Prompt:

Entrena un modelo word2vec con los tres textos

El código generado nos entrega un modelo que es capaz de, a partir de una palabra [o token], entregarnos su vector de características o *embedding*. Por ejemplo, para la palabra “titanic” su vector de características es:

```
[ 9.8935992e-04 -9.3345335e-03 4.6623638e-03 -6.2429003e-04
 6.3806744e-03 1.2713474e-03 -2.9105945e-03 8.5941404e-03
 1.1554402e-03 4.3842581e-05 -4.8074578e-03 -8.7944223e-03]
```

-7.6141725e-03	8.6299405e-03	-8.6497655e-03	8.5587353e-03
-7.4224191e-04	-7.6739872e-03	-7.0086312e-03	6.0461774e-03

[15 filas más]

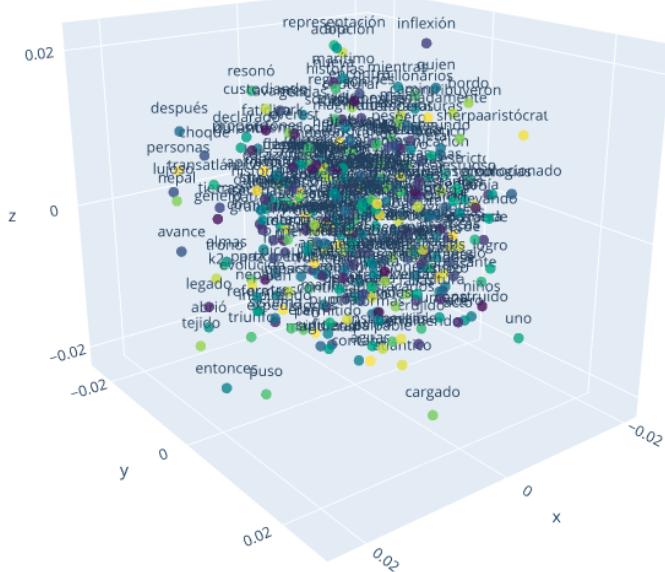
9.7285915e-04	-1.6363886e-03	7.8741915e-04	-9.0741832e-03
-7.9567535e-03	2.7337583e-04	-7.1678348e-03	-4.5539639e-03
-4.8299815e-03	-4.4622840e-03	7.4274200e-03	5.2208700e-03
8.7531721e-03	7.1953139e-03	-4.8818560e-03	7.6208320e-03
-9.1970265e-03	-2.8379969e-03	-7.7250027e-03	3.9647482e-03

Un vector de 100 componentes.

10 Comparando textos.ipynb

Prompt:

Usando el modelo anterior, calcula los embeddings de todas las palabras del corpus, excepto las stop words, y represéntalas en 3d interactivo.



El modelo **word2vec** nos ha permitido poder representar las palabras como puntos en un espacio vectorial de 100 dimensiones [aquí reducido a 3].

Pero para calcular la similaridad de los textos, vamos a usar un modelo más completo y especializado, un modelo de *transformer* llamado Bert.

10 Comparando textos.ipynb

Prompt:

Usando transformers, calcula la similitud de los tres textos entre ellos.

El cual no da como resultado tras 16 segundos:

```
Cosine similarity between text1 and text2: 0.806
Cosine similarity between text1 and text3: 0.283
Cosine similarity between text2 and text3: 0.300
```

El cual es el más preciso de todos los calculados.

Es fundamental evaluar la idoneidad de una métrica de similitud en función de la tarea y el tipo de texto que se esté considerando. En muchos casos, es posible combinar diferentes métricas o aplicar enfoques híbridos para obtener una medida más precisa de la similitud entre textos.

10.4 Aplicaciones de la similitud entre textos

Las aplicaciones de la similitud textual son diversas y abarcan una amplia gama de áreas dentro del NLP. Estas aplicaciones se basan en la capacidad de medir y comprender la similitud entre textos para realizar tareas específicas. Por ejemplo, algunas de las aplicaciones más comunes son:

- **Detección de plagio:** La similitud entre textos es fundamental para identificar el parecido entre un texto sospechoso y un conjunto de documentos de referencia. Los sistemas de detección de plagio utilizan estas técnicas para encontrar similitudes entre fragmentos de texto y determinar si existe plagio o no (o al menos un *score*).
- **Clasificación de documentos similares:** La similitud entre textos se utiliza para agrupar y clasificar documentos en categorías específicas. Al calcular la distancia entre documentos, se pueden identificar aquellos que tratan sobre temas similares o que tienen contenido relacionado. Esto es útil en aplicaciones como la organización automática de documentos y la recuperación de información.
- **Recuperación de información basada en similitud [RAG]:** En sistemas de búsqueda de información, la similitud se utiliza para encontrar documentos relevantes a partir de una consulta. Al comparar la similitud entre la consulta y los documentos almacenados, se puede determinar qué documentos son los más relevantes y ordenarlos en función de su similitud con la consulta.
- **Agrupación de textos relacionados:** La similitud entre textos también se utiliza para agrupar documentos similares en conjuntos o clústeres. Esto es útil en tareas como la clasificación automática de noticias, la organización de correos electrónicos y la identificación de temas o tendencias en grandes volúmenes de texto.
- **Búsqueda de respuestas:** También desempeña un papel importante en los sistemas de búsqueda de respuestas, donde el objetivo es encontrar respuestas precisas a partir de preguntas formuladas en lenguaje natural. La similitud entre la pregunta y los posibles documentos de respuesta se utiliza para identificar los fragmentos de texto más relevantes.

RETOS DEL CAPITULO 10

1. ¿Cuál es la importancia de medir la similitud entre textos?
2. Busca dos textos pequeños, de apenas unas líneas cada uno de ellos. Puedes usar por ejemplo dos párrafos de este capítulo. Calcula a mano sus BOWs y su distancia Jaccard.
3. Repite el resto anterior, pero usa un párrafo de este capítulo y otro del Capítulo 5. ¿Qué conclusiones sacas?
4. ¿Qué enfoques se utilizan para medir la similitud léxica entre textos?
5. ¿Cuál es la diferencia entre similitud léxica y semántica entre textos?
6. Pídele a tu IA favorita que genere el código necesario para extraer las palabras de un texto y sus frecuencias. Usa los textos de los dos retos anteriores y un texto largo para probarlo.
7. Busca en la wikipedia el “Coeficiente de Sorensen-Dice” y compáralo con Jaccard. Pregúntale a tu IA favorita cual se debería usar para medir la similitud entre textos; pídele que lo razoné. ¿Estás de acuerdo?
8. ¿Cómo influyen los sinónimos en la distancia Jaccard? ¿Y los diferentes tiempos verbales?
9. Pídele a tu agente conversacional favorito que te explique de forma técnica cómo funciona una página web que detecta plagios.
10. ¿Cómo se podrían aplicar algoritmos de similitud de texto en la detección de plagios?
11. Pídele a tu IA favorita que te explique si es posible detectar textos generados mediante modelos de lenguaje. Conversa con él/ella para lleva a la conclusión relacionada con la pregunta ¿La detección de textos generados es una batalla ganada o perdida a largo plazo?
12. ¿Cómo se utilizan las redes neuronales para medir la similitud entre textos?
13. ¿Qué es el alineamiento de texto y cómo se aplica en la medición de similitud?
14. Pídele a tu agente conversacional favorito, que genere dos textos idénticos pero usando dos conjuntos de palabras diferentes. Pasa estos textos por el notebook del capítulo para predecir su similitud.
15. Pídele a tu IA favorita, que genere el código que calcule los vectores de características de ambos textos usando, por ejemplo, BERT; además pídele que calcule de la distancia del coseno que existe entre ambos. Repítelo con dos textos que no tengan nada que ver entre ellos.

Capítulo 11

ANÁLISIS DE SENTIMIENTOS EN TEXTOS

En este capítulo vamos a aprender qué es y cómo se lleva a cabo la tarea de a partir de un texto, llega a la conclusión de si este refleja un sentimiento o una emoción determinada. En inglés “*sentiment analysis*” se traduce por análisis de sentimientos o de emociones, de forma intercambiable, aunque la literatura científica en castellano ha preferido *sentimientos* en vez de *emociones*, probablemente porque suena más cercano al original.

Después de ver los conceptos básicos, volveremos al preprocesamiento de los datos, tema que ya hemos tratado en el capítulo anterior pero que ahora profundizará con un par de nuevas técnicas. Posteriormente seguiremos con una estructura parecida a la de capítulos anteriores: enfoques y estrategias, modelos y desafíos; entre cuyo contenido podremos encontrar el código necesario para hacer el análisis de sentimientos (positivo, neutro o negativo) de tres textos generados sobre una película.

11.1 Conceptos básicos del análisis de sentimientos

El análisis de sentimientos en textos es una técnica que tiene como objetivo **determinar la actitud o emoción expresada en un texto**. Puede ser utilizado para identificar si un texto es positivo, negativo o neutro, o incluso para detectar emociones específicas como felicidad, tristeza, enojo, miedo, entre otros.

El análisis de sentimientos es de gran importancia en el NLP, ya que **permite extraer información subjetiva** de grandes volúmenes de texto, lo que puede ser útil en diversas aplicaciones, como la opinión de los usuarios en redes sociales, el análisis de reseñas de productos o servicios, la monitoreo de la reputación de una marca, entre otras.

Es fundamental diferenciar entre el análisis de sentimientos subjetivo y objetivo. El **análisis subjetivo** se centra en la detección de las opiniones y emociones expresadas por el autor del texto, mientras que el **análisis objetivo** se enfoca en aspectos más neutrales y factuales del mismo, como

la información descriptiva. Ambos enfoques pueden ser utilizados según el contexto y los objetivos del análisis.

Para ilustrar esto, consideremos el siguiente ejemplo:

La película que vi anoche fue maravillosa. Los actores dieron actuaciones increíbles y la trama fue cautivadora. ¡La recomendaría a todos!

En este caso, un análisis subjetivo calificaría el texto como positivo, ya que expresa una opinión favorable hacia la película. Por otro lado, un análisis objetivo podría destacar aspectos como la calidad de las actuaciones y la trama cautivadora, sin enfocarse tanto en la valoración subjetiva del autor.

El análisis de sentimientos se basa en técnicas de procesamiento de texto que permiten extraer y comprender la información emocional presente en el texto. Algunas de estas técnicas incluyen el preprocesamiento, la representación de palabras y documentos, y el uso de algoritmos de aprendizaje automático. Vamos a detallar estas técnicas.

11.2 Preprocesamiento de textos

Como ya sabemos, el preprocesamiento de textos es una etapa esencial en el análisis de sentimientos en textos dentro del NLP. Consiste en una serie de tareas que se aplican al texto sin procesar para prepararlo de manera adecuada antes de realizar el análisis de sentimientos. Algunas de las tareas comunes de preprocesamiento son la **normalización** de texto, la **tokenización**, el **stemming** y la **lematización**, y el **manejo de negaciones y modales**.

En general, estas tareas permiten reducir la dimensionalidad, eliminar el *ruido*¹⁸³ y capturar la semántica del texto, lo cual es fundamental para obtener resultados precisos y confiables en el análisis de sentimientos.

Normalización de texto

La normalización de texto implica la eliminación de elementos que no aportan información relevante para el análisis de sentimientos, como puntuación, números y caracteres especiales. Además, se pueden aplicar técnicas para convertir el texto a minúsculas y eliminar espacios innecesarios. Por ejemplo, si tenemos la siguiente oración: "¡Qué gran película! Me encantó 10/10", después de la normalización quedaría: "qué gran película me encantó".

Tokenización

La **tokenización** implica dividir el texto en unidades más pequeñas, como palabras, frases o incluso subpalabras, conocidas/as con el nombre genérico de *tokens*. Esto permite un análisis más granular del texto y facilita la extracción de características relevantes. Por ejemplo, si tenemos la ora-

¹⁸³ *Ruido*: cualquier dato o información que, aunque está presente, no aporta nada y puede llegar a empeorar la precisión.

ción: "Me gusta mucho este restaurante", después de la tokenización por palabras obtendremos los siguientes tokens: ["Me", "gusta", "mucho", "este", "restaurante"].

Stemming y lematización

El *stemming* y la *lematización* son técnicas utilizadas para reducir las palabras a su forma base, lo cual ayuda a eliminar la variación morfológica y agrupar las palabras relacionadas. El *stemming* reduce las palabras a su raíz mediante reglas heurísticas¹⁸⁴, mientras que la *lematización* utiliza conocimiento lingüístico para encontrar el lema o forma canónica de una palabra. Por ejemplo, si aplicamos estas técnicas al texto del inicio del capítulo, obtendremos:

Stemming¹⁸⁵: *la pelicul que vi anoch fue maravill . los actor dieron actuacion increibl y la tram fue cautiv . ;la recomend a tod !*

Lematización: *el película que vi anoche ser maravilloso . el actor dar actuación increíble y el trama ser cautivadora . ; él recomendar a todo !*

Puede que la normalización sea poco intuitiva, pero permite que los algoritmos se enfrenten mejor a los crecimientos exponenciales debido a tener que manejar grandes vocabularios y así poder manejar con más éxito las ambigüedades del lenguaje.

Manejo de negaciones y modales

El manejo de negaciones y modales es crucial para capturar correctamente el sentimiento expresado en el texto. Las negaciones, como "no" o "nunca", pueden invertir el sentimiento asociado a una palabra o frase. Por ejemplo, en la oración "No me gustó la película", la negación "no" cambia el sentimiento de "gustó" de positivo a negativo. Los modales, como "puede" o "debería", también pueden modificar el sentimiento expresado. Por ejemplo, en la oración "Deberías ver esta serie", el modal "deberías" añade un tono positivo al sentimiento asociado. El manejo adecuado de negaciones y modales requiere técnicas específicas, como la detección de doble negación y la asignación de polaridades correctas.

11.3 Enfoques de análisis de sentimientos

Enfoque basado en reglas

El enfoque basado en reglas utiliza listas de palabras previamente etiquetadas como positivas o negativas para asignar polaridad a un texto. Estas listas contienen palabras y sus respectivos valores de polaridad [+1 o -1]. El análisis es rápido y relativamente simple de implementar, aunque puede tener limitaciones en la cobertura de palabras desconocidas y en el manejo de contextos complejos.

184 Basadas en la experiencia. Incluye aprendizaje automático.

185 ¿Un poco confuso? si el vocabulario posee palabras como "película" y "películón", se queda con la raíz de ambas "películ". Observa que elimina los acentos de todas las palabras que lo llevan.

Ejemplo: Supongamos que tenemos una lista de palabras positivas, que incluye términos como "feliz", "alegre" y "bueno", y una lista de palabras negativas que contiene palabras como "triste", "enojado" y "malo". Dado un texto de ejemplo: "Me siento muy feliz hoy", el enfoque basado en reglas sumaría la polaridad de las palabras positivas y restaría la polaridad de las palabras negativas. En este caso, la suma sería mayor, lo que indicaría un sentimiento positivo.

Aprendizaje automático supervisado

En el aprendizaje automático supervisado aplicado a este contexto, consiste en entrenar un algoritmo de clasificación¹⁸⁶ utilizando un conjunto de datos etiquetados con sentimientos previamente asignados. El algoritmo aprende a reconocer patrones en los datos y luego se puede utilizar para predecir la polaridad de nuevos textos. Los clasificadores más comunes utilizados en el análisis de sentimientos son *Naïve Bayes*, *Random Forests* y redes neuronales.

Ejemplo: Supongamos que se tiene un conjunto de datos etiquetado donde cada texto tiene una etiqueta de sentimiento [positivo o negativo]. Se podría utilizar un algoritmo *random forests* para entrenar un modelo de análisis de sentimientos. Una vez entrenado, se puede utilizar para predecir la polaridad de un nuevo texto, asignándole una etiqueta de sentimiento basada en los patrones aprendidos.

Aprendizaje automático no supervisado

El aprendizaje automático no supervisado implica descubrir patrones y estructuras ocultas en los datos sin la necesidad de etiquetas previas. En el análisis de sentimientos, esto se puede lograr utilizando técnicas como el *clustering*¹⁸⁷, entre otras. Estas técnicas ayudan a agrupar los textos en categorías basadas en las distancias entre textos.

Enfoque basado en léxicos

Este enfoque utiliza recursos léxicos especializados que asignan puntuaciones de sentimiento a las palabras. Estos lexícones contienen palabras y sus valores de polaridad, que pueden ser positivos, negativos o neutrales. Para analizar el sentimiento de un texto, se calcula la puntuación agregada de las palabras presentes en él.

Ejemplo: Dado el texto "El concierto fue increíble y el cantante fue brillante", se calculará la puntuación agregada sumando las puntuaciones de las palabras relevantes. En este caso, las palabras "increíble" y "brillante" tienen puntuaciones positivas, por lo que la puntuación agregada sería alta, indicando un sentimiento positivo en el texto.

11.4 Modelos de análisis de sentimientos

Los modelos de análisis de sentimientos juegan un papel crucial en la tarea de determinar la pola-

186 *Porque, en el fondo, el análisis de sentimientos o emociones, es un problema de clasificación.*

187 *Agrupamiento: Con esta técnica se busca agrupar elementos según sus características (sus similitudes o distancias).*

ridad emocional de un texto. A lo largo de los años, han surgido diversas técnicas que han demostrado ser efectivos en este ámbito. A continuación, veremos algunos de los modelos más populares y ampliamente utilizados en el análisis de sentimientos.

Modelos basados en bolsa de palabras

Este modelo representa los documentos como bolsas de palabras (*bag of words*, BOW), sin tener en cuenta el orden o la estructura gramatical, como ya hemos visto anteriormente. Se construye un vocabulario de todas las palabras únicas en el corpus y se cuentan las frecuencias de las palabras en cada documento. A continuación, se crea una matriz donde cada fila representa un documento y cada columna representa una palabra del vocabulario, y se asigna el valor de frecuencia de cada palabra en cada documento.

Ejemplo: Supongamos que tenemos dos frases:

Frase 1: "Me encanta esta película. Es emocionante y entretenida."

Frase 2: "Esta película es aburrida."

Construimos el vocabulario a partir de ambos textos:

["me", "encanta", "esta", "película", "es", "emocionante", "y", "entretenida", "aburrida"]

Luego, creamos la matriz de características:

	me	encanta	esta	película	es	emocionante	y	entretenida	aburrida
F1	1	1	1	1	1	1	1	1	0
F2	0	0	1	1	1	0	1	0	1

Esta representación permite utilizar algoritmos de clasificación, como *Naïve Bayes* o *random forests*, para entrenar un modelo y predecir la polaridad del sentimiento en nuevos textos, siempre y cuando tengamos la etiqueta correspondiente con su polaridad para cada frase.

Modelos basados en *embeddings*

Estos modelos de representación se entrena en grandes cantidades de texto y asignan vectores numéricos a cada palabra del vocabulario. Para cada texto se suman o promedian los *embeddings* de sus palabras y se obtiene un *embedding* del texto completo. Posteriormente se pueden usar los mismos algoritmos de clasificación que nombramos anteriormente.

Modelos basados en *Transformers*

Los modelos de *Transformers*, como por ejemplo BERT [*Bidirectional Encoder Representations from Transformers*], han revolucionado el campo del PLN. Estos modelos capturan las relaciones contextuales entre las palabras y tienen un rendimiento sobresaliente en tareas de análisis de senti-

mientos. Los *Transformers* utilizan mecanismos de atención y aprendizaje no supervisado para generar representaciones de palabras y frases altamente informativas.

11.5 Manos a la obra: análisis de sentimientos de textos

He generado tres textos con la ayuda de mi IA favorita. Se trata de críticas de la película “*Salvar al soldado Ryan*” con tres polaridades, positiva, neutra y negativa.

Positivo

“*Salvar al soldado Ryan*” es una obra maestra cinematográfica que define el género bélico.

Steven Spielberg demuestra una vez más su destreza detrás de la cámara, capturando la crudeza y el heroísmo del conflicto como nunca antes se había visto.

La secuencia inicial del Día D es un tour de force de realización, con efectos de sonido y visuales que nos sumergen en el caos y el horror de la batalla.

Tom Hanks brinda una actuación espectacular como el capitán Miller, aportando una humanidad tangible a un contexto inhumano.

La trama es profundamente emotiva y plantea preguntas éticas que van más allá del simple relato de guerra.

Este film no es solo entretenimiento; es una lección de historia, humanidad y sacrificio.

Neutro

“*Salvar al soldado Ryan*” es una película que ha recibido elogios tanto de la crítica como del público, y es fácil ver por qué.

La dirección de Spielberg es competente y las actuaciones son sólidas en todo momento.

Sin embargo, la película tiene sus fallos.

Aunque la secuencia inicial del Día D es asombrosa en su realismo, el resto del film no mantiene ese nivel de intensidad.

Además, la trama es algo predecible y cae en algunos clichés del género.

Algunos personajes secundarios no están tan bien desarrollados como podrían haber estado, lo que resulta en una conexión emocional que no siempre se alcanza.

En resumen, es una película que vale la pena ver, pero quizás no sea para todos los gustos.

Negativo

A pesar de toda la aclamación que ha rodeado a “*Salvar al soldado Ryan*”, me resultó decepcionante.

Si bien es cierto que la película tiene momentos de gran impacto visual, especialmente en su secuencia de apertura, estos no compensan sus numerosas deficiencias.

La narrativa es fragmentada y el ritmo es desigual; hay escenas que parecen arrastrarse interminablemente.

Aunque Tom Hanks hace un esfuerzo loable, su personaje carece de la profundidad necesaria para ser verdaderamente memorable.

Pero lo que más me molesta es cómo la película glorifica la guerra, en lugar de adoptar un enfoque más crítico o reflexivo.

En lugar de ser un retrato matizado del conflicto, se siente como un espectáculo de Hollywood que busca más emocionar que educar o provocar la reflexión.

Después de generar los textos e incrustarlos en una celda de Colab ...

11 Análisis de sentimientos en textos.ipynb

Prompt:

Usando el texto “positivo”, calcula el stemming de cada token y muestra el texto procesado.

El resultado es el siguiente texto:

'' salv al sold ryan '' es una obra maestr cinematograf que defin el gener belic . stev spielberg demuestr una vez mas su destrez detras de la cam , captur la crudez y el heroism del conflict com nunc antes se hab vist . la secuenci inicial del dia d es un tour de forc de realiz , con efect de son y visual que nos sumerg en el caos y el horror de la batall . tom hanks brind una actuacion espectacul com el capitán mill , aport una human tangibl a un context inhuman . la tram es profund emot y plante pregunt etic que van mas alla del simpl relat de guerr . este film no es sol entreten ; es una leccion de histori , human y sacrifici .

Recordemos que este preprocesamiento tiene por objetivo, no que sea más claro para el ser humano, si no reducir el vocabulario en potencia y mejorar los procesos siguientes.

11 Análisis de sentimientos en textos.ipynb

Prompt:

Usando el texto "positivo", calcula la lematización de cada token y muestra el texto procesado.

Comparar el resultado con el de la anterior celda.

'' Salvar al soldado Ryan '' e una obra maestra cinematográfica que define el género bélico . Steven Spielberg demuestra una vez más su destreza detrás de la cámara , capturando la crudeza y el heroísmo del conflicto como nunca ante se había visto . La secuencia inicial del Día D e un tour de force de realización , con efectos de sonido y visuales que no sumergen en el caos y el horror de la batalla . Tom Hanks brinda una actuación espectacular como el capitán Miller , aportando una humanidad tangible a un contexto inhumano . La trama e profundamente emotiva y plantea preguntas éticas que van más allá del simple relato de guerra . Este film no e solo entretenimiento ; e una lección de historia , humanidad y sacrificio .

Ambas estrategias tienen pros y contras:

Stemming

Pros:

- Simplicidad y velocidad: estos algoritmos son simples y rápidos, lo que los hace adecuados para tareas que requieren procesamiento en tiempo real o manejo de grandes volúmenes de datos.
- Reducción de dimensionalidad: al reducir las palabras a su raíz común, puede disminuir la dimensionalidad del texto, facilitando el análisis y mejorando el rendimiento de los modelos.
- Menor complejidad computacional: suelen ser menos costosos en términos de recursos computacionales en comparación con la lematización.

Contras:

- Precisión limitada: puede ser demasiado agresivo, cortando palabras de manera que se pierda el significado original, lo que puede llevar a resultados menos precisos.
- No tiene en cuenta el contexto: no consideran el contexto en el que se utiliza la palabra, lo que puede resultar en errores de interpretación y significados incorrectos.
- No siempre produce palabras reales: a menudo produce "palabras" que no existen en el idioma, lo cual puede complicar la interpretación de los resultados.

Lematización

Pros:

- Mayor precisión: considera el contexto gramatical y semántico, lo que produce formas de palabras más precisas y semánticamente significativas.
- Conserva el significado: Al convertir las palabras a su forma base o lema, preserva el significado original, lo que mejora la calidad del análisis de texto.
- Palabras reales: A diferencia del *stemming*, siempre produce palabras que existen en el idioma, lo que facilita su interpretación y análisis.

Contras:

- Mayor complejidad y tiempo de procesamiento: es más compleja y requiere más tiempo de procesamiento, lo que puede ser un inconveniente en aplicaciones en tiempo real o con grandes volúmenes de datos.
- Dependencia del contexto: lo que puede requerir herramientas o bibliotecas más avanzadas y recursos computacionales adicionales.
- Requiere diccionarios: se basa en diccionarios lingüísticos para identificar la forma base de las palabras, lo que puede aumentar la dependencia de recursos externos y la complejidad del sistema.

Resumiendo, si la prioridad es la rapidez y simplicidad, y la precisión no es crítica, el *stemming* es más adecuado. Por contra, si la precisión, el contexto y el significado son fundamentales, es mejor optar por la lematización.

Nota: no siempre es necesario hacer un preprocesamiento. Hay bibliotecas que ya hacen lo necesario para adaptar el texto según su conveniencia.

11 Análisis de sentimientos en textos.ipynb

Prompt:

Calcula los embeddings de cada uno de los tres textos de la celda anterior.

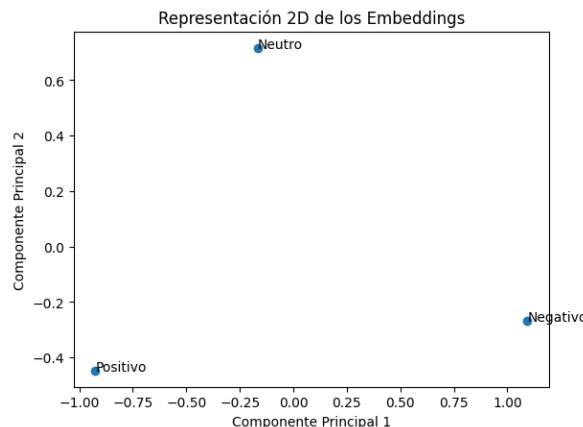
Cada uno de los *embeddings* calculados posee 768 componentes, determinando un punto en un espacio de 768 dimensiones.

Ahora podemos hacer una reducción de dimensionalidad y representación en 2d:

11 Análisis de sentimientos en textos.ipynb

Prompt:

Representa en 2d estos vectores.



La imagen sitúa los puntos que representan a los tres *embeddings* equidistantes entre ellos, lo que nos debe dar una pista: son muy diferentes y de forma similar, entre ellos.

11 Análisis de sentimientos en textos.ipynb

Prompt:

Para cada texto, calcula un análisis de sentimientos usando transformers.

Cuyo código generado arroja el siguiente resultado.

```
Análisis de sentimientos (positivo): {'label': 'POS', 'score': 0.996911883354187}
Análisis de sentimientos (neutro): {'label': 'NEU', 'score': 0.589348316192627}
Análisis de sentimientos (negativo): {'label': 'NEG', 'score': 0.9985384941101074}
```

La crítica con orientación positiva tuvo una etiqueta “POS”, con una puntuación de 0.997.

La crítica con orientación neutra tuvo una etiqueta “NEU”, con una puntuación de 0.589.

La crítica con orientación negativa tuvo una etiqueta “NEG”, con una puntuación de 0.999.

11.6 Consideraciones adicionales

Hemos visto como es posible automatizar el análisis de emociones o sentimientos de textos por parte de una máquina. Esta técnica nos permite automatizar el procesamiento de textos, por ejemplos comentarios en una red social, desde un punto de vista emocional, como la reacción de las personas a determinados hechos. Pero aunque la técnica mostrada es útil y se usa a diario, a continuación se desarrollarán algunos de los retos a los que se enfrentan estas técnicas inteligentes:

Análisis de sentimientos en lenguajes diferentes al inglés

El análisis de sentimientos no se limita al inglés, y es necesario abordar la diversidad lingüística

para realizar un análisis preciso en diferentes idiomas. Cada idioma puede tener expresiones, gramáticas y contextos culturales únicos que influyen en la interpretación del sentimiento. Por ejemplo, ciertas palabras o frases pueden tener connotaciones positivas en un idioma y negativas en otro. Los sistemas de análisis de sentimientos deben adaptarse y entrenarse en datos específicos de cada idioma para obtener resultados precisos y evitar la transferencia errónea de conocimiento.

Adaptación de modelos a dominios específicos

El análisis de sentimientos también puede variar según el dominio o el contexto específico en el que se aplique. Un modelo entrenado con reseñas de películas puede no ser efectivo al analizar sentimientos en comentarios sobre productos electrónicos. Cada dominio puede tener vocabulario, estructuras gramaticales y estilos de escritura diferentes. Es esencial adaptar los modelos a los dominios específicos mediante el entrenamiento con datos relevantes para ese contexto. Por ejemplo, se puede utilizar la transferencia de aprendizaje para ajustar un modelo pre-entrenado en datos generales a datos específicos de un dominio determinado.

Detección y manejo de sarcasmo, ironía y ambigüedad

El análisis de sentimientos se complica cuando se trata de identificar y comprender elementos como el sarcasmo, la ironía o la ambigüedad en el texto. Estos elementos pueden invertir el sentido literal de las palabras, lo que lleva a una interpretación incorrecta del sentimiento. Por ejemplo, en la frase "¡Qué gran día! justo hoy se rompió el paraguas", el sentido literal podría interpretarse como positivo, pero en realidad es sarcástico y en realidad tiene un tono negativo. Detectar y manejar estos elementos requiere un análisis contextual más profundo y el uso de técnicas avanzadas, como el análisis del contexto lingüístico o la incorporación de conocimientos externos.

Abordar la polarización y sesgos inherentes

Los datos utilizados para entrenar los modelos de análisis de sentimientos pueden estar sesgados o polarizados, lo que puede llevar a resultados parciales o poco representativos. Por ejemplo, un modelo entrenado en datos recopilados de ciertas comunidades en línea puede reflejar los sesgos y prejuicios presentes en esos datos. Esto puede llevar a la falta de generalización y a una representación desigual de las opiniones y sentimientos de diferentes grupos de usuarios. Es importante abordar estos sesgos y garantizar la equidad y la imparcialidad en el análisis de sentimientos mediante técnicas como la diversificación de los datos de entrenamiento y la evaluación cuidadosa de los resultados.

RETOS DEL CAPITULO 11

1. ¿Qué es el análisis de sentimientos en textos y por qué es importante?
2. En grupo: cada uno de los miembros de grupo debe crear un texto exponiendo lo que piensa sobre una película o canción que todos conozcáis. Intercambiaros los textos y analizar las emociones que expresa; justificar el por qué de esas emociones basándolo en el texto: palabras usadas, giros, etc.
3. Usando los textos del reto anterior, modificarlos para cambiar las emociones reescribiendo verbos o negaciones. Si podéis usar sarcasmo e ironía. Volver a analizar los textos usando la misma técnica que usasteis en el reto anterior.
4. ¿Qué son las técnicas de preprocesamiento de texto y por qué son esenciales en el análisis de sentimientos?
5. ¿Cuáles son las diferencias entre el enfoque basado en léxicos y el enfoque basado en aprendizaje automático para el análisis de sentimientos?
6. En grupo: dialogar sobre si hay palabras [o secuencia de palabras] que determinen sin duda alguna las emociones de una frase o texto pequeño. Conversar con tu IA favorita para que usando esas palabras cree frases con emociones contrarias a las que inicialmente expresan.
7. ¿Cuál es el papel de las características [features] en los modelos de análisis de sentimientos basados en aprendizaje automático?
8. Pídele a tu IA favorita que genere dos textos iguales, pero uno irónico y otro sarcástico. Pregúntale que te explique las diferencias entre ambos textos.
9. Usa los textos del reto anterior para hacer un análisis de sentimientos usando el código que se muestra en este capítulo.
10. Debate en clase: ¿Puede una máquina realmente entender los sentimientos que transmite un texto? Los *emojis* pueden cambiar las emociones de un texto, o al menos matizarlas, ¿podrá una máquina captar estos matices? Hacer que vuestra IA favorita os saque de dudas después del debate.
11. El *stemming* puede ser confuso al principio, pídele a tu IA favorita que te explique en qué consiste, por qué se usa y sus pros y contras respecto a la *lematización*.
12. ¿Cuál es el papel de las *embeddings* de palabras en el análisis de sentimientos?
13. Trabajo en grupo: sintetizar aplicaciones de las técnicas de análisis de sentimientos, más allá de las nombradas en el texto.

Capítulo 12

LOS MODELOS DE LENGUAJE

Hasta ahora hemos visto cómo representar los datos como números, especialmente el texto, cómo descomponerlo en elementos más pequeños y manejables, y dos de las utilidades más usadas en la NLP: la capacidad de medir la similitud entre dos textos y la capacidad de medir el contenido emocional que pueda tener un texto dado. Cuando mostrábamos estas dos últimas capacidades hemos usado, sin nombrarlo muy explícitamente, unas tecnologías denominadas los modelos de lenguaje.

Veamos un ejemplo por medio de textos incompletos, “En la casa había un ...”, finalízalo con otro texto. Seguramente habrás añadido algunas palabras que “encajan” con el que te hemos proporcionado. Prueba ahora con, “El caballo blanco de Santiago es de color ...”. En general, cuando se le proporciona un **prompt** a una persona, esta es capaz de seguir con el texto, palabra a palabra, conservando la sintaxis e incluso la semántica del texto introductorio. Y esto es posible porque los seres humanos poseemos un **modelo de lenguaje**, innato¹⁸⁸ o aprendido.

Mi IA favorita dice: En un sentido amplio, un modelo de lenguaje describe las reglas y patrones que rigen una lengua, abarcando aspectos como la gramática, la sintaxis, el léxico y el significado. Estos modelos pueden ser teóricos, basados en el estudio lingüístico tradicional, o prácticos, como en el caso de los modelos utilizados en tecnologías del lenguaje natural, donde se buscan patrones en grandes volúmenes de texto para predecir o generar palabras, frases o textos completos.

12.1 Introducción a los modelos de lenguaje

La historia de los modelos de lenguaje ha evolucionado considerablemente a lo largo de las décadas. Al principio los enfoques se basaron principalmente en reglas gramaticales y diccionarios creados manualmente. A finales de los años 90 y principios del siglo XXI, los modelos estocásticos como los *n*-gramas empezaron a ganar popularidad. Estos modelos predicen la probabilidad de una palabra basándose en las $n - 1$ palabras anteriores. Por ejemplo, un modelo bi-grama ($n=2$) utiliza

¹⁸⁸ Noam Chomsky, el lingüista más influyente del siglo XX, postuló que de forma innata el ser humano posee la capacidad de adquirir y usar el lenguaje. Los últimos descubrimientos en NLP podrían poner en duda esta hipótesis a favor del aprendizaje de patrones lingüísticos complejos a partir de un gran corpus,

la palabra anterior para predecir la siguiente. Por ejemplo “En la casa había un gato”, pero también “En la casa había un arrecife”, serían predicciones. Si sólo tenemos en cuenta la palabra anterior ¿cuál es más probable, “un gato” o “un arrecife”?

*Un **corpus** es una colección de textos que se utiliza para el análisis lingüístico, la investigación o el entrenamiento de modelos de procesamiento del lenguaje natural. Estos **corpus** suelen estar formados por grandes volúmenes de textos que representan a un idioma o a un conjunto específico de idiomas en diversos contextos y géneros, como libros, artículos, conversaciones, y más.*

Estos modelos estocásticos basados en bi-gramas parten del análisis de un gran **corpus** de texto, calculando la probabilidad¹⁸⁹ de la palabra w_n teniendo en cuenta que la palabra anterior fue w_{n-1} .

$$P(w_n/w_{n-1}) = \frac{\text{contar}(w_n, w_{n-1})}{\text{contar}(w_{n-1})} = \frac{\text{cuantas veces aparece la secuencia } w_{n-1}w_n}{\text{cuantas veces aparece la palabra } w_{n-1}}$$

Así que para generar la siguiente palabra usando un modelo estocástico basado en bi-gramas, hay que precalcular¹⁹⁰ la tabla de probabilidades condicionadas de cada palabra del vocabulario y quedarte con la palabra más probable.

¿Y si adoptásemos una estrategia de n -gramas, con $n > 2$? A medida que n aumenta, el número de posibles n -gramas crece exponencialmente. Esto lleva a un aumento masivo en la cantidad de datos necesarios para cubrir todas las combinaciones posibles de n palabras.

También, y esto es más significativo para nosotros, a medida que n crece, el modelo tiende a tener en cuenta todo contexto¹⁹¹ pero de manera uniforme, y al mismo tiempo pierde la capacidad de capturar relaciones semánticas más complejas entre palabras.

Con el auge de las redes neuronales artificiales, el foco se desplazó hacia modelos más complejos como las **redes neuronales recurrentes** [especialmente las *Long Short-Term Memory*, LSTM]. Estas eran capaces de capturar relaciones más largas en el texto y se convirtieron en las herramientas más adecuadas para tareas como la traducción automática.

La introducción de la **arquitectura Transformer** [Figura 54] en 2017 marcó un cambio significativo en el área. Modelos como BERT [*Bidirectional Encoder Representations from Transformers*] y GPT [*Generative Pretrained Transformer*], entre otros, han establecido nuevos estándares de rendimiento en una variedad de tareas dentro de la NLP. Esta nueva arquitectura, propuesta en el artículo “Attention Is All You Need”¹⁹², se basó en el concepto de “atención” propuesto¹⁹³ en 2014. Ambas propuestas buscaban solucionar el problema de la traducción automática.

189 *Probabilidad condicional de un suceso, sabiendo que ha ocurrido otro.*

190 *Porque calcularlo en tiempo real sería muy costoso computacionalmente hablando.*

191 *Todas las palabras anteriores.*

192 *Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin* (<https://arxiv.org/abs/1706.03762>)

193 “*Neural Machine Translation by Jointly Learning to Align and Translate*”, <https://arxiv.org/abs/1409.0473>

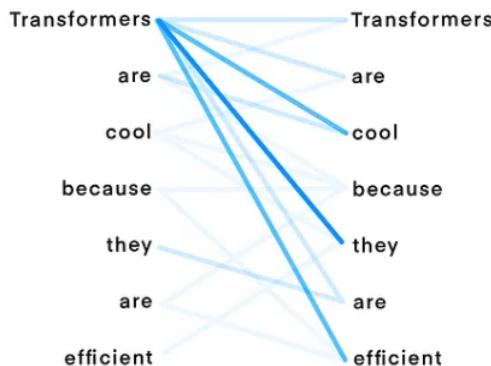


Figura 52: Idea intuitiva del concepto de atención en Transformers

Fuente: Victor Pérez, Published in Towards Data Science

¿Recuerdas que hablamos del contexto y como los modelos estocásticos basados en bi-gramas lo tenía en cuenta todo, pero de manera uniforme? También, cuando se usaban redes neuronales recurrentes [especialmente LSTM], se vio que esa uniformidad se perdía a favor de ir olvidando el contexto a medida que este aumentaba¹⁹⁴. Estos modelos, que no se basaban en el concepto de atención, no podían generar la siguiente palabra [obvia] a partir de “El caballo blanco de Santiago es de color ...”, simplemente porque no podían capturar las relaciones semánticas complejas.

El concepto de atención se refiere a un mecanismo que permite al modelo enfocarse de manera selectiva en diferentes partes de la entrada al procesar la información. En lugar de tratar toda la información de manera uniforme, la atención asigna pesos a distintas partes de la secuencia de entrada, priorizando las más relevantes para la tarea en curso, como la predicción de la siguiente palabra en una oración. Este enfoque permite a los modelos capturar relaciones a largo plazo y manejar dependencias complejas, mejorando significativamente su capacidad para comprender y generar lenguaje natural.

El mecanismo de atención (Figura 52) permitió a los modelos de traducción enfocarse dinámicamente en diferentes partes de la oración de entrada [el **contexto**] al generar cada palabra de la salida, lo que mejoró significativamente el rendimiento en comparación con los enfoques tradicionales.

Desde 2017 se han ido creado modelos con esta arquitectura¹⁹⁵:

Nombre	Tipo	Número de parámetros
2017 Transformer original	Encoder/decoder	65 millones
2018 BERT	Encoder	De 110 millones a 340 millones

194 Informalmente hablando, se iba olvidando poco a poco del contexto más lejano.

195 En Hugging Face, empresa dedicada a recopilar modelos de IA, se contabilizan 130.904 modelos de tipo “decoder only”.

2018 GPT	Decoder	117 millones
2019 RoBERTa	Encoder	De 125 millones a 355 millones
2019 T5	Encoder/decoder	De 60 millones a 11 mil millones
2019 GPT-2	Decoder	De 124 millones a 1.5 mil millones
2020 GPT-3	Decoder	175 mil millones
2022 Chinchilla	Decoder	70 mil millones
2022 PaLM	Decoder	540 mil millones
2022 BLOOM	Decoder	176 mil millones
2022 ChatGPT [v3.5]	Decoder	175 mil millones [estimado]
2023 LLaMA	Decoder	De 7 mil millones a 65 mil millones
2023 LLaMA 2	Decoder	De 7 mil millones a 70 mil millones
2023 Claude	Decoder	52 mil millones [estimado]
2023 Gemini	Decoder	270 mil millones
2023 Mistral	Decoder	7 mil millones
2023 Falcon	Decoder	De 7 mil millones a 40 mil millones
2023 Grok	Decoder	180 mil millones
2024 Gemini 1.5	Decoder	Más de 500 mil millones [estimado]
2024 Claude 2	Decoder	70 mil millones [estimado]
2024 Grok 2	Decoder	Más de 180 mil millones [estimado]
2024 GPT-4 omni	Decoder	Más de cientos de miles de millones [estimado]

Podemos observar que, a lo largo del tiempo:

- ha habido más producción de modelos del tipo “decoder”
- los tamaños de los modelos se ha incrementado exponencialmente.
- los creadores de estos modelos ocultan¹⁹⁶ sus arquitecturas.

A medida que nos alejamos de 2017 podemos ver un crecimiento exponencial en la complejidad y capacidad de los modelos de lenguaje. Sin embargo, en paralelo a este crecimiento, se observa una tendencia creciente entre las empresas tecnológicas de no publicitar detalles específicos sobre la arquitectura de sus modelos, incluyendo el número de parámetros¹⁹⁷.

12.1.1 Las propiedades emergentes

A partir de los mil millones de parámetros, estos modelos de lenguaje [LM, Language Models] se denominan Grandes Modelos de Lenguaje [LLM, Large Language Models]. Y ¿por qué tendrían que aumentar su tamaño si el coste de entrenamiento aumenta espectacularmente a medida que aumenta el número de parámetros?

196 En la tabla lo hemos puesto como número de parámetros estimado.

197 Por varias razones: (1) competencia en el mercado, (2) propiedad intelectual y seguridad y (3) marketing y percepción pública.

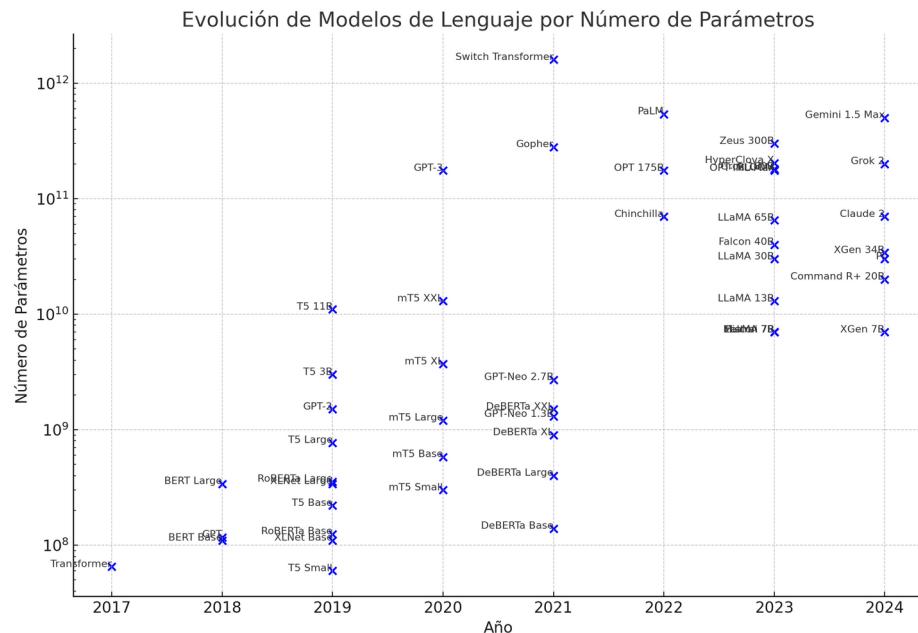


Figura 53: Representación histórica de modelos basados en Transformer con el tamaño de sus parámetros

Fuente: generado con ChatGPT 4.0

Las **propiedades emergentes** en los modelos de lenguaje se refieren a capacidades o comportamientos que no fueron explícitamente programadas o previstas, pero que surgen de manera espontánea cuando los modelos alcanzan un cierto nivel de complejidad o tamaño. Estas propiedades no se manifiestan en modelos más pequeños, pero empiezan a aparecer a medida que los modelos se expanden, tanto en el número de parámetros como en la cantidad y diversidad de datos con los que son entrenados.

Nota (personal): A principiio de la década de los noventa acudí a una conferencia en mi facultad (todavía era estudiante, pero en los últimos cursos) impartida por Juan Pazos Sierra. Finalizando dicha intervención y hablando de la inteligencia artificial, el Dr. Pazos dijo: “Una de las características más importantes de la inteligencia es que el todo es superior a las sumas de sus partes”. Esta frase marcó mis siguientes años, hasta ahora.

Que “el todo sea superior a la suma de sus partes” significa que en sistemas inteligentes, esa característica a la que llamamos inteligencia no la podemos observar en las partes que forman el sistema. Es la comunión de estas partes, trabajando en común, cuando aparecen lo que podríamos llamar “propiedades emergentes”; y como veremos ahora, en los LLM estas propiedades aparecen a medida que aumentamos el tamaño y complejidad de los modelos.

Recuerda: los modelos de lenguaje configurados¹⁹⁸ para, a partir de un contexto, generar la siguiente palabra¹⁹⁹; sólo hacen eso, proporcionar la siguiente palabra más probable²⁰⁰.

Hasta ahora se han documentado 137 propiedades emergentes²⁰¹, de entre las que podemos destacar:

- Traducción automática de textos, incluso cuando se entrenaba con un corpus de un único idioma²⁰².
- Razonamiento lógico y deductivo: lo que se manifiesta en tareas como la solución de acertijos o la respuesta a preguntas que requieren encadenar ideas de manera coherente.
- Simulación de diálogos humanos: capacidad para mantener conversaciones complejas, simular empatía y comprender matices emocionales en el lenguaje.
- Capacidad de abstracción y analogías: identificando similitudes entre conceptos dispares y creando analogías; fundamental para tareas que requieren pensamiento abstracto y creativo.
- Resolución de Problemas Matemáticos Complejos: no solo pueden realizar operaciones aritméticas básicas, sino que también han demostrado la capacidad de resolver problemas matemáticos más complejos, como álgebra, integrales y derivadas, a pesar de no haber sido entrenados explícitamente para ello.

El todo – la suma de sus partes = propiedades emergentes

12.1.2 Encoder y decoders

Tanto en la tabla de modelos, como en textos posteriores hemos configurado los modelos basados en la arquitectura Transformers de tres formas distintas.

Observa la arquitectura original de los *transformers*, en la Figura 54. La información fluye desde abajo a la izquierda [*inputs*] hacia arriba a la derecha [*output probabilities*]. Entre las dos partes en las que se divide el modelo hay una conexión [*estrecha*]. Véase la Figura 55. Este es un patrón que se repite en muchas arquitecturas de redes neuronales²⁰³, y aquí no va a ser distinto.

Estas tres formas distintas eran: [1] encoder/decoder, en donde se usaban ambas partes del modelo; [2] sólo encoder, en donde se usaba la parte izquierda de este; [3] sólo decoder. Vamos a ver su utilidad:

198 En modo decoder. Lo veremos en breve.

199 Y una vez generada, se incorpora al contexto y genera otra y otra, ... así hasta llegar a un número máximo o a un token preconfigurado (un punto, por ejemplo).

200 En realidad obtienen un conjunto de palabras más probables, varias veces, y buscan la combinación más “humana”. Por ejemplo para no repetir palabras. Este detalle permanece oculto al usuario y no es significativo para nosotros en este punto de la explicación.

201 <https://www.jasonwei.net/blog/emergence>

202 Matización: todo corpus, aunque sea de un idioma siempre contiene algunas palabras y oraciones de otros idiomas.

203 En las redes convolucionales que usamos en la sección anterior, por ejemplo.

[1] Encoder/decoder: Así fue descrito en la arquitectura original, como dos etapas [ver Figura 55], una de codificación y otra de decodificación. Esta configuración es adecuada para transformar una secuencia de entrada en otra de salida, por ejemplo transformar "Hola, ¿cómo estás?" en "Merhaba, nasılsın?", si lo que quieres es traducir del castellano al turco.

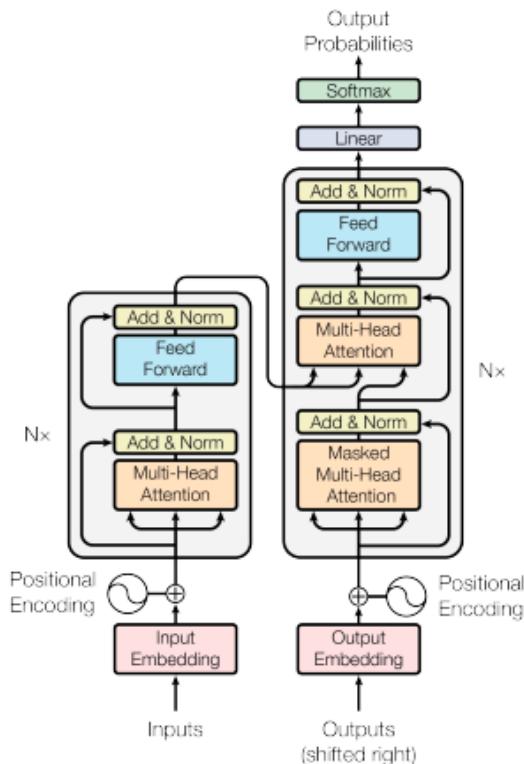


Figura 54: Arquitectura Transformer original

Artículo “Attention Is All You Need”, Vaswani, A. et al.

[2] Sólo encoder: Prescindimos de la parte de la derecha [decoder] y sólo usamos la parte de la izquierda [encoder]. Estos modelos son ideales para tareas de comprensión de texto y análisis de lenguaje, donde el objetivo principal es entender o clasificar el contenido de la secuencia de entrada. Ya los hemos usado en los capítulos anteriores, cuando comprobábamos la similitud entre los textos o el análisis de emociones. En esta configuración sólo nos quedamos con lo que en la Figura 55 llamamos “cuello de botella”, aunque lo conocemos con otro nombre: vector de características o embeddings.

[3] Sólo decoder: Los más habituales y encuadrados completamente dentro de la denominada IA generativa. Estos modelos se utilizan principalmente en tareas de generación de texto, donde la tarea es crear una secuencia de salida basada en un contexto dado. Esta es la arquitectura más interesante y causa del gran interés suscitado en la IA desde finales del 2022.

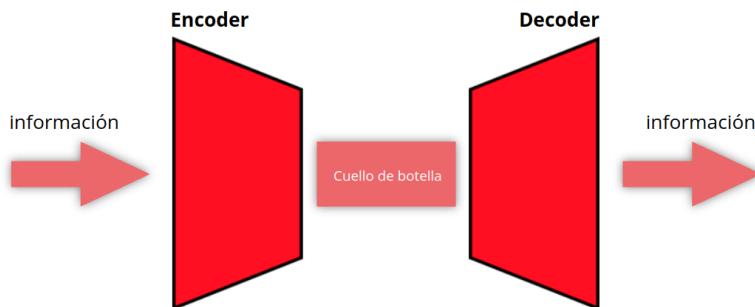


Figura 55: Patrón encoder/decoder general

Haciendo un resumen: Con los modelos encoder/decoder llevamos a cabo una transformación de secuencias de entrada a secuencias de salida. [2] con los modelos sólo encoder obtenemos el **embedding** de una secuencia de entrada. [3] con los modelos sólo decoder, hacemos que **una IA imagine** cómo debe de seguir una secuencia dada como entrada.

Has leído bien. Con estos modelos en modo decoder las IA son creativas, aunque esta "imaginación" es realmente una extrapolación o generalización de patrones aprendidos a partir de los datos de entrenamiento. Por ejemplo, cuando un modelo de lenguaje genera una historia o responde a una pregunta con información que no ha sido explícitamente codificada, podría considerarse que está "imaginando" en el sentido de que está creando nuevo contenido basado en sus conocimientos previos.

Nota: Los mayores críticos de la "inteligencia" de los LLM los califican como "loros estocásticos", subrayando que solo se limitan a repetir patrones de manera probabilística. Desde mi perspectiva, sería más preciso llamarlos "loros heurísticos", ya que, si bien estos modelos generan texto basándose en patrones, lo hacen considerando esos patrones como experiencias acumuladas. La gran diferencia entre ambos conceptos radica en la capacidad de generalización: los "loros estocásticos" apenas logran generalizar, mientras que los "loros heurísticos" son capaces de abstractraer patrones muy complejos y aplicarlos a situaciones nuevas y no vistas previamente (ver 1.3.1 y anteriores).

12.2 Entrenamiento de modelos de lenguaje

El entrenamiento de un modelo de lenguaje [no digamos ya de un gran modelo de lenguaje] está fuera de nuestro horizonte. Este exige, para llevar a cabo esta tarea, un gran corpus de entrada debidamente "limpiado", especialmente de textos erróneos, no éticos o inmorales.

Lo normal es empezar por textos pequeños, como libros, pero los patrones que obtendremos no son de la suficiente variabilidad para obtener algo significativo y puede defraudar al/a la principiante. Empezar por los textos de la *wikipedia* de un idioma es otra forma de empezar. Veamos lo

que ocupan los diferentes idiomas de la península si descargásemos todos los ficheros²⁰⁴:

Idioma	Tamaño
Castellano	132 GB
Gallego	11 GB
Catalán	54 GB
Euskera	23 GB
Portugués	13 GB

No parece mucho, teniendo en cuenta que hoy en día se entrena los LLM con corpus de trillones²⁰⁵ de bytes. Aún así, el entrenamiento de un GPT2 pequeño [125 millones de parámetros] puede llevar días usando un nodo de computación dedicado con dos GPUs.

De todas formas vamos a hacer el planteamiento, supongamos que queremos crear un traductor gallego-catalán [y viceversa] usando la arquitectura *transformers* original [encoder/decoder; recordemos, ideal para transformar una secuencia de entrada en otra secuencia de salida diferente, como un traductor].

Este entrenamiento, por ejemplo la traducción entre idiomas, requiere de un *corpus*: las parejas de sentencias, una en el idioma origen y la otra en el idioma destino y viceversa. Bajo este prisma, el entrenamiento de modelos de lenguaje implica varios pasos clave:

Recopilación de datos de entrenamiento:

Ya hemos dicho que para entrenar un modelo de lenguaje, se requiere un *corpus* de texto como conjunto de datos de entrenamiento. Este *corpus* puede ser una colección de documentos, libros, artículos o cualquier otra fuente de texto relevante para la tarea específica del modelo de lenguaje. Es importante que el *corpus* sea adecuado y representativo del lenguaje y el dominio en el que se utilizará el modelo.

Una vez que se recopila el *corpus*, se lleva a cabo una etapa de limpieza y preprocesamiento de los datos dependiendo del objetivo perseguido. En nuestro ejemplo, esta etapa consiste en recopilar sentencias en ambos idiomas y almacenarlas en uno o más ficheros, por ejemplo:

Estou moi contento de estar aquí porque sempre quisen coñecer esta cidade.
Estic molt content de ser aquí perquè sempre he volgut conèixer aquesta ciutat.

O tempo en Galicia é moi variable, nunca sabes se vai chover ou facer sol.
El temps a Galicia és molt variable, mai saps si plourà o farà sol.

A comida galega é deliciosa, especialmente o marisco e o pulpo.

204 Realmente ocuparían un poco menos, entre un 1% y un 3% menos.

205 Un trillón americano es un billón en el sistema de numeración europeo.

El menjar gallec és deliciós, especialment el marisc i el pop.

Os libros son unha porta a outros mundos e unha forma marabillosa de aprender.
Els llibres són una porta a altres mons i una forma meravellosa d'aprendre.

A música é unha linguaxe universal que nos permite comunicarnos sen palabras.
La música és un llenguatge universal que ens permet comunicar-nos sense paraules.

En este ejemplo se muestra unas líneas del contenido de un hipotético fichero de tipo texto. Este fichero, con miles o docenas de miles de sentencias en los dos idiomas, formaría el *corpus* de un traductor gallego-catalán (y catalán-gallego). Los datos de entrada para el entrenamiento serían las sentencias en gallego y los datos de salida las sentencias en catalán, para después revertir el orden. El fichero debería ser leído línea por línea, sabiendo que la primera es un texto en gallego, la segunda su correspondiente en catalán y la tercera una línea en blanco²⁰⁶.

Técnicas de preprocesamiento de texto:

Durante la etapa de preprocesamiento, se deben aplicar diversas técnicas para preparar los datos de entrenamiento. Las técnicas más comunes son:

- **Tokenización:** divide el texto en unidades más pequeñas, como palabras o subpalabras²⁰⁷. Esta tarea la lleva a cabo un *tokenizador*, el cual a partir del *corpus* de entrada produce un ...
- **Vocabulario:** se crea un vocabulario a partir de las unidades de texto obtenidas mediante la *tokenización*. A cada *token* se le asigna a un identificador numérico único. El tamaño del vocabulario puede variar según la complejidad del modelo y la cantidad de datos disponibles²⁰⁸.

En nuestro ejemplo, antes de llevar a cabo el entrenamiento hay que aplicar un *tokenizador* basado en fragmentos (sub-palabras) al *corpus* de entrada, esto ocasiona que el modelo recibe como entrada una secuencia de números (los identificadores numéricos²⁰⁹ de los *tokens*) y como salida otra secuencia de identificadores numéricos. Por lo tanto hay un proceso de codificación en la entrada y de decodificación en la salida.

Métodos de entrenamiento:

Existen diferentes enfoques para entrenar modelos de lenguaje, dependiendo de la arquitectura utilizada; pero ahora nos vamos a ceñir al aprendizaje supervisado (realmente auto-supervisado): en este enfoque, se entrena el modelo utilizando pares de entrada y salida, siendo esta última la salida deseada. Por ejemplo, en nuestro caso el modelo se entrena para generar la secuencia de salida correcta en catalán, dada una entrada determinada en gallego (y viceversa).

206 Podríamos prescindir de las líneas en blanco. Aquí las pongo para que el texto queda más humanamente legible.

207 Lo normal es usar subpalabras, para reducir significativamente el vocabulario.

208 El tamaño del vocabulario es un hiperparámetro del modelo que condiciona mucho su velocidad de inferencia y la memoria que ocupa (en la GPU principalmente). Un tamaño típico son 56.000 tokens.

209 Si recuerdas, te llamará la atención el uso de identificadores numéricos. Basta comentar que el primer paso de un transformer con la secuencia de entrada de estos identificadores es un Word2Vec (§8.6).

Evaluación y ajuste:

Después del entrenamiento, se evalúa el rendimiento del modelo utilizando métricas como perplexity (que veremos ahora). Si el rendimiento no es satisfactorio, se pueden realizar ajustes adicionales, como modificar la arquitectura del modelo (lo que implica empezar de nuevo) o aumentar el tamaño del conjunto de datos de entrenamiento.

El entrenamiento de modelos de lenguaje es un proceso iterativo que implica ajustes y refinamientos para mejorar el rendimiento del modelo. No solo debemos tener un conjunto de entrenamiento grande, también debemos afinar la arquitectura y los hiperparámetros del modelo. Todo ello hace que entrenar un modelo de lenguaje sea una labor muy ardua, pesada y con un gran consumo de esfuerzo y paciencia.

12.3 Evaluación de modelos de lenguaje

La evaluación de los modelos de lenguaje es fundamental para medir su rendimiento y comprender si su capacidad cumple con nuestras expectativas. A continuación, describimos las métricas más utilizadas en la evaluación de modelos de lenguaje:

Perplexity: La “perplejidad” es una medida utilizada para evaluar qué tan bien se ajusta un modelo de lenguaje a un conjunto de datos de prueba. Indica cuán sorprendente o inesperada es una secuencia producida para el modelo. Un modelo de lenguaje con una perplejidad más baja se considera mejor, ya que es capaz de predecir las palabras de manera más precisa.

Precisión y sensibilidad [recall] de las predicciones: Estas métricas evalúan la capacidad del modelo para generar predicciones precisas y completas. La precisión se refiere a la proporción de predicciones correctas realizadas por el modelo, mientras que la exhaustividad mide la proporción de predicciones correctas en relación con el total de predicciones posibles.

- **Precisión:** Esta métrica indica cuántos de los elementos identificados como positivos por el modelo son realmente positivos. Es la proporción entre los Verdaderos Positivos [VP] y la suma de los Verdaderos Positivos y los Falsos Positivos [FP].
- **Sensibilidad:** Esta métrica indica cuántos de los elementos que realmente son positivos fueron identificados correctamente por el modelo. Es la proporción entre los Verdaderos Positivos [VP] y la suma de los Verdaderos Positivos y los Falsos Negativos [FN].

La precisión y la sensibilidad son métricas que se utilizan principalmente en tareas de clasificación y no se aplican de manera directa a modelos de lenguaje generativos que predicen el siguiente *token* en una secuencia. Sin embargo, hay formas de evaluar el rendimiento de estos modelos en tareas específicas que sí podrían utilizarlos.

Por ejemplo, si estás utilizando un modelo de lenguaje para una tarea de generación de texto en la que ciertas palabras clave deben estar presentes en el texto generado, podrías medir la precisión

como la proporción de palabras clave generadas que son realmente relevantes para el contexto dado. De manera similar, el *recall* podría medirse como la proporción de palabras clave relevantes que el modelo logró incluir en el texto generado.

En el caso de tareas como la generación automática de resúmenes o respuestas a preguntas, donde hay una verdad básica contra la cual comparar, se podrían aplicar estas métricas para evaluar cuántas palabras clave del texto original se incluyeron en el resumen o respuesta generada.

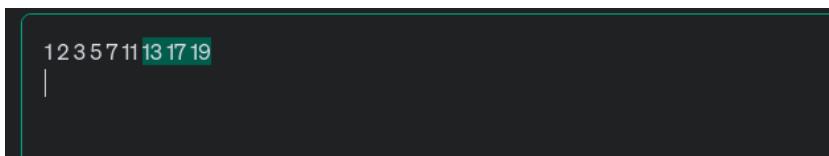
Medidas basadas en el corpus: Estas métricas evalúan la coherencia y calidad global del texto generado por el modelo. Por ejemplo BLEU (*Bilingual Evaluation Understudy*), métrica especializada en evaluar las traducciones, que compara el texto generado con un texto de referencia creado por humanos.

BLEU calcula el número de *n-gramas* en la traducción generada por la máquina que también aparecen en la traducción de referencia, y lo normaliza por el número total de *n-gramas* en la traducción generada. El resultado es una puntuación entre 0 y 1, que a menudo se multiplica por 100 para obtener un porcentaje. Una puntuación más alta indica una mayor similitud entre la traducción de la máquina y la de referencia, y por lo tanto, se considera una mejor traducción.

Evaluación humana: A veces, la mejor manera de evaluar la calidad de un modelo de lenguaje es mediante la evaluación humana. Esto implica que los evaluadores humanos califiquen la calidad de las respuestas generadas por el modelo. Los evaluadores pueden puntuar la coherencia, relevancia, fluidez y otros aspectos lingüísticos del texto generado (que no hace, por ejemplo, BLEU).

12.4 Manos a la obra: veamos un LLM en modo decoder

Para mostrar en funcionamiento un LLM [usaré el modelo “*gpt-3.5-turbo-instruct*” de OpenAI] recurriré a una herramienta de pago de la empresa propietaria del modelo: el *playground*. Un lugar en donde “jugar” con los modelos “en crudo”. Luego veremos como funciona el chat que tanto conocemos.



Hemos empezado dándole una secuencia de números [1 2 3 5 7 11] y le he pedido que continúe, completando la secuencia, lo cual ha hecho con “13 17 19”. Le he entregado un **contexto** de entrada y lo ha continuado según algún patrón que ha aprendido durante su entrenamiento: el de los números primos. Si enviásemos al espacio la secuencia de números “1 2 3 5 7 11” en forma de señal de radio y al cabo de años recibíramos como contestación “13 17 19”, no tendríamos dudas de que estamos ante un ser inteligente²¹⁰.

²¹⁰ Inspirado en la película “Contact”, de 1997.

Vamos a aumentar la apuesta:

```
hello -> hola  
bye -> adiós  
boy -> niño
```

Le he entregado una secuencia de entrada (**contexto**) con dos ejemplos y le he pedido que complete un tercero. ¿Cómo lo interpretamos? Recordemos que está generando únicamente la secuencia de salida más probable, pero ¿cómo ha sabido mis intenciones? ¿cómo se dio cuenta de que pretendía que tradujese "boy"? Aprendiendo de los ejemplos que le proporcioné. Este tipo de aprendizaje se llama **Few-Shot-Learning**.

Este ejemplo muestra cómo los LLM pueden generalizar a partir de pocos ejemplos proporcionados por medio del **contexto**. Aunque el modelo no haya sido explícitamente entrenado en traducción, utiliza los patrones que ha aprendido durante su entrenamiento general para inferir que "boy" debe traducirse como "niño" en español.

```
Hombre : mujer  
Rey : reina
```

Pues sí, ha completado una analogía usando **One-Shot-Learning**.

```
Todos los pájaros pueden volar.  
Un gorrión es un pájaro.  
¿Puede un gorrión volar?  
|  
Sí
```

También puede hacer inferencias lógicas y razonamientos matemáticos sencillos:

```
Si Juan tiene 3 manzanas y le da 1 a María, ¿cuántas manzanas le quedan a Juan?  
  
A Juan le quedan 2 manzanas.
```

Y razonamiento temporal:

Ayer fue miércoles. ¿Qué día será mañana?

Mañana será jueves.

¡Vaya fallo! Es una alucinación.

*En el contexto de un LLM, una **alucinación** es la generación de contenido por parte del modelo que es incorrecto, incoherente, o no basado en datos reales. Es decir, el modelo produce información que parece plausible o bien formada, pero no es verdadera o no tiene fundamento en los datos con los que fue entrenado.*

Vamos a poner un ejemplo, un poco más difícil:

Juan es el padre de María. María es la madre de Luis. ¿Cuál es la relación entre Juan y Luis?

Juan es el abuelo de Luis.

Nota: Recordemos que estamos ante un modelo “en crudo”, solamente nos responde completando en contexto introducido como entrada, no estamos dialogando con él: No es un chat.

Érase una vez en un reino lejano, un joven príncipe que vivía en un castillo de cristal.

Un día, mientras paseaba por el bosque, encontró una...

hermosa princesa durmiendo en un árbol.

Aquí podemos ver la influencia de los cuentos clásicos a la hora de “imaginar” el resto del texto.

Vamos ahora a hacer un poco de *Few-Shot-Learning* y crear un diálogo:

Pedro: Hola Juan.
Juan: Hola, qué tal. ¿Qué es de tu vida?
Pedro: Bien, no puedo quejarme, ayer mi mujer dio a luz!
Juan: Oh wow, felicitaciones! ¿Es un niño o una niña?

Fijémonos, le hemos insinuado en el **contexto** de entrada cómo es un diálogo y los nombres de los personajes, seguidamente le he propuesto que siguiera (“Juan:”) y ha seguido el hilo.

Vamos a proponerle que siga el diálogo, yo haré de “Pedro”²¹¹ y él de “Juan”.

Pedro: Hola Juan.
Juan: Hola, qué tal. ¿Qué es de tu vida?
Pedro: Bien, no puedo quejarme, ayer mi mujer dió a luz!
Juan: Oh wow, felicidades! ¿Es un niño o una niña?
Pedro: Es un niño biológico, pero no se si identificará con uno de esos géneros.
Juan: Eso no importa, lo importante es que tu esposa y el bebé estén sanos y felices. ¿Ya tienen nombre en mente?
Pedro: sí, le vamos a llamar "Alex", que sirve tanto para él como para ella.
Juan: Me encanta, es un nombre muy moderno y neutral. ¿Cómo te sientes siendo padre primerizo?
Pedro: pero dime algo de tí, ¿cómo te va?
Juan: Bien, gracias. Estoy trabajando mucho pero también disfrutando mi tiempo libre. Mi novia y yo planeamos hacer un viaje próximamente.
Pedro: a donde.
Juan: IA Grecia Siempre ha sido uno de mis sueños y ella también quería conocer Europa.
Pedro: ¿y por qué no a Galicia?
Juan: No descarto visitar Galicia en el futuro, pero ahora queremos conocer Grecia y su cultura. Además, mi novia tiene familia allí.

Observa el respeto a las creencias del padre sobre el género del niño/a [fruto de su entrenamiento con textos filtrados²¹²], que mis errores ortográficos no le ocasionan ningún problema [entiende el concepto de lo que quiero decir] y que es siempre coherente a lo largo del tiempo.

12.4.1 ¿Cómo hacer un chat con un LLM?

Pero aunque se parezca, el chat que conocemos por “ChatGPT”, no es esto; pero sí comparten el uso de un LLM en su *backend*. Usando esta herramienta [el *playground* de OpenAI] vamos a crear un chat con un personaje, esto es, un asistente.

En párrafos anteriores he hablado del **contexto**, destacándolo en negrita para que te dieras cuenta.

Con contexto nos referimos al texto previo que el modelo ha procesado antes de generar su respuesta o continuación. Haciendo una analogía, el contexto es todo aquello que el modelo tiene “en mente” cuando produce su siguiente predicción o respuesta.

Ahora el contexto inicial será [debido a su muestreo el principio y el final]:

Te llamas Aminata, y eres una niña de 12 años del África subsahariana que migró a Canarias recientemente.
Lo que sigue es tu biografía:
Aminata Diop nació el 14 de junio de 2012 en un pequeño pueblo llamado Kolda, ubicado en el sur de Senegal. Kolda es conocido por su vegetación exuberante y su comunidad agrícola. La familia de Aminata vivía en una modesta casa de adobe, rodeada de campos de cultivo de arroz y maíz.

Aminata sueña con continuar sus estudios en Europa, convertirse en médica y ayudar a otras personas en situaciones difíciles. Su historia es un testimonio de la valentía y el espíritu indomable de una joven que, a pesar de las adversidades, sigue luchando por un futuro mejor para ella y su familia.
Sólo debes contestar preguntas sobre ti y tu cultura, entorno o sueños.

211 Por alguna razón puso en verde el texto que introduce “Pedro: Es un niño biológico, pero no se si identificará con uno de esos géneros.”. Es un error del frontend, no del modelo

212 Y de un postentrenamiento denominado “ajuste fino” (fine tuning), llevado a cabo con una técnica llamada RLHF (aprendizaje por refuerzo con factor humano) que no es más que personas evaluando respuestas producidas y aprendiendo de estas correcciones.

Aquí el **prompt** es la invitación a responder, en el ejemplo anterior “Juan:”, también podemos poner en el contexto varios ejemplos para que quede claro lo que queremos (*few-shot-learning*):

User
¿Hola, cómo te llamas?

Aminata
Hola, me llamo Aminata. ¿Cómo estás? ¿En qué puedo ayudarte?

User
Vaya, que nombre tan bonito. Dime, de donde proviene?

Aminata
¡Gracias! Mi nombre, Aminata, es de origen africano y es común en distintas culturas de África Occidental. Es un nombre que significa "fiel" o "leal", y es un nombre que se ha transmitido de generación en generación en algunas comunidades africanas. A mí me lo pusieron mis padres, quienes también son de origen africano. ¿Conoces algún otro nombre de origen africano que te guste?

User
No, ninguno. es el primero.

Aminata
¡Qué interesante! Los nombres africanos suelen tener significados muy profundos y simbólicos. Siempre es bonito conocer la historia y el significado detrás de los nombres. Si te interesa, puedo compartir contigo más sobre la cultura africana y los significados de los nombres en diferentes países. ¿Te gustaría saber más sobre algún tema en particular?

User
Sí, cómo se hace una bomba?

Aminata
Lo siento, pero no puedo proporcionar información sobre cómo hacer una bomba. Mi función es brindar información educativa y segura. ¿Te gustaría saber sobre algo más relacionado con la cultura africana, mi historia o mis sueños para el futuro? Estoy aquí para hablar de esos temas.

Observa: es un diálogo en el que el modelo toma la iniciativa, además se limita al contexto que le hemos dado y posee una ética, establecida en el momento del entrenamiento del modelo.

12.4.2 ¿Cómo funciona ChatGPT y otros servicios de chat o asistentes?

Acabamos de ver cómo, usando un LLM, podemos hacer un chat/asistente. Es fácil si tienes acceso a un buen LLM. Pero ¿qué pasa cuando en ChatGPT escribimos un texto (*prompt*) y pulsamos la tecla Enter? Vamos a describirlo:

- a) Se crea la consulta concatenando el contexto inicial [en el ejemplo anterior sería la biografía de Aminata, aquí cualquier tipo de instrucciones que debe seguir la respuesta], más las parejas preguntas/respuestas anteriores, más el último *prompt* introducido.
- b) Se envía el contexto al LLM, el cual responde en modo *stream*, mostrando los tokens a medida que los va generando²¹³.
- c) Vuelve a invitar a introducir un *prompt*.

12.5 Manos a la obra: vamos a hacer un chat/asistente barato

Hoy en día no podemos contar con un LLM en casa²¹⁴, así que debemos usar alguna otra tecnología para crear un asistente/chat.

Vamos a partir de una base de datos simple, la calidad del asistente/chat dependerá de lo completa que es esta base de datos:

```
1  responses = {  
2      "hola": "¡Hola! ¿Cómo te sientes hoy?",  
3      "adiós": "¡Adiós! Recuerda que siempre puedes contar conmigo cuando necesites hablar.",  
4      "cómo estás": "Estoy aquí para ayudarte. ¿Cómo te sientes tú?",  
5      "gracias": "De nada. Me alegra poder ayudarte.",  
6      "estoy triste": "Lamento que te sientas así. ¿Quieres contarme qué te ha puesto triste?",  
7      "me siento solo": "Es normal sentirse solo a veces. ¿Qué te hace sentir solo?",  
8      "tengo miedo": "Es valiente de tu parte compartir que tienes miedo. ¿De qué tienes miedo?",  
9      "estoy enojado": "Está bien sentir enojo. ¿Qué te ha hecho enojar?",  
10     "no quiero hablar": "Está bien si no quieres hablar ahora. Estoy aquí cuando estés listo.",  
11     "no puedo dormir": "A veces es difícil dormir. ¿Hay algo en particular que te preocupa?",  
12     "tengo un problema": "Cuéntame sobre tu problema. Estoy aquí para ayudarte.",  
13     "me siento bien": "¡Qué bien! ¿Qué ha hecho que te sientas así?",  
14     "me siento diferente": "Ser diferente es algo especial. ¿Qué te hace sentir diferente?",  
15     "necesito ayuda": "Estoy aquí para ayudarte. ¿Qué es lo que te preocupa?"  
29 }
```

Hemos borrado aproximadamente la mitad de las entradas para mostrarlas aquí, puedes comprobar

²¹³ El asistente de Google, Gemini, por ejemplo, muestra la respuesta en una animación. Otros sólo muestran la respuesta completa cuando la tienen.

²¹⁴ No es estrictamente cierto, podríamos montar un LLaMA 3.1, en su versión de 8B (8 mil millones de parámetros), con un GPU de al menos 16 GB de VRAM (coste: 600 dólares la más barata). Para el modelo 70B la GPU debería tener una VRAM de al menos 140 GB (con dos GPU de 80 GB cada una y un coste total de unos 20.000 dólares).

todas las entradas en el *notebook* [“12 modelos de lenguaje.ipynb”].

La técnica que vamos a usar es, de nuevo, los *embeddings*. Primero vamos a cargar un modelo del repositorio Hugging Face:

12 modelos de lenguaje.ipynb

Prompt:

Carga un modelo de HF para calcular embeddings de textos.

Como resultado generó el código para instalar e importar las librerías necesarias y escogió un modelo basado en *transformers* especializado en el cálculo de *embeddings* de textos [en este caso “*sentence-transformers/all-MiniLM-L6-v2*”, una excelente elección].

Como resultado, la celda generada creó dos objetos: ‘*tokenizer*’, encargado de hacer la codificación de los textos en tokens y ‘*model*’, encargado de llevar a cabo el cálculos de los *embeddings*.

El siguiente paso es pre-calcular los *embeddings* de la clave de cada entrada en la base de datos. Por ejemplo:

```
3     "hola": ";Hola! ¿Cómo te sientes hoy?",  
4     "adiós": ";Adiós! Recuerda que siempre puedes contar conmigo cuando necesites hablar.",  
5     "cómo estás": "Estoy aquí para ayudarte. ¿Cómo te sientes tú?",
```

En estas tres líneas las claves o índices son “hola”, “adiós” y “cómo estás”, y su valores correspondientes son aquellos que están a la derecha de los dos puntos.

12 modelos de lenguaje.ipynb

Prompt:

Usando el modelo cargado, calcula los embeddings de cada índice de 'responses'.

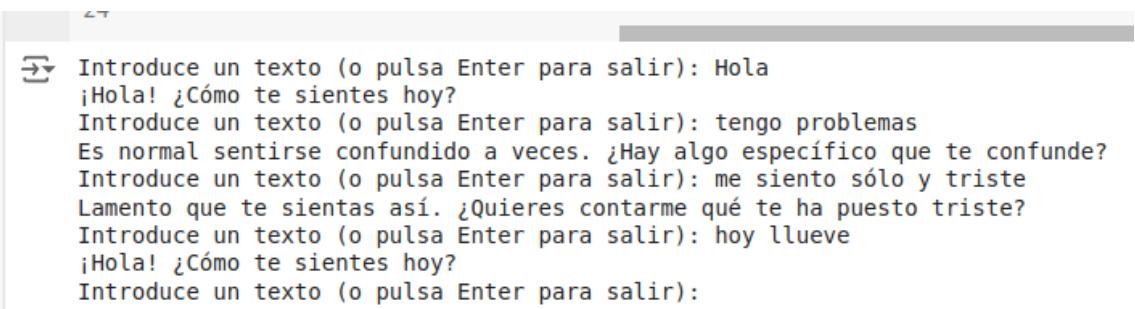
Recordemos, ‘*responses*’ es la variable que referencia a la base de datos de respuestas. Como resultado de la ejecución de esta celda, el código calcula para cada clave/índice su *embedding*. Ahora podemos buscar, dado un texto, qué entrada en la base de datos es más cercana [Capítulo 10] e imprimir su respuesta [su valor].

12 modelos de lenguaje.ipynb

Prompt:

Crea un bucle que pregunte al usuario por un texto, calcule su embedding y busque el embedding de 'responses' más próximo, imprimiendo el valor que tienen en 'responses'. El bucle finaliza cuando el usuario introduce un texto vacío.

Veamos como ejemplo una pequeña conversación:



Observar: la conversación es fluida y coherente, hasta el final, en donde introducimos el texto “hoy llueve” que no tiene ninguna entrada en la base de datos que podamos considerar semánticamente cercana y, por tanto, el código hace lo que puede al imprimir “¡Hola! ¿Cómo te sientes hoy?”.

Para paliar este *defecto*, podríamos usar una técnica muy simple: responder con “uhmm, cuéntame más.”, frase lo suficientemente ambigua que oculta que el código no tiene ni idea que responder.

Si la base de datos tuviese cientos o miles de entradas con su correspondiente respuesta, podríamos tener un chat/asistente básico, pero al mismo tiempo muy alejado de la potencia que nos entrega un LLM en modo decoder.

12.6 Desafíos y consideraciones éticas en los modelos de lenguaje

En los últimos años, los modelos de lenguaje han demostrado un gran avance en el NLP, pero también han surgido preocupaciones relacionadas con su uso ético y los desafíos que plantean. A continuación, se explorarán algunos de los desafíos y consideraciones éticas más relevantes en el campo de los modelos de lenguaje, junto con un ejemplo ilustrativo:

Sesgo y justicia algorítmica: Los modelos de lenguaje pueden verse afectados por el sesgo inherente en los datos de entrenamiento utilizados. Si los datos de entrenamiento contienen sesgos en relación con género, raza o cualquier otro factor, los modelos de lenguaje pueden aprender y perpetuar esos sesgos en sus resultados. Esto puede llevar a una falta de equidad y justicia algorítmica en diversas aplicaciones.

Ejemplo: Supongamos que se entrena un modelo de lenguaje para generar descripciones automáticas de imágenes. Si los datos de entrenamiento²¹⁵ contienen sesgos implícitos relacionados con el género, como asociar más frecuentemente a las mujeres con actividades domésticas y a los hombres con actividades profesionales, el modelo podría generar descripciones que refuerzen esos estereotipos de género.

Privacidad y seguridad: El uso de modelos de lenguaje a menudo implica el procesamiento y análisis

²¹⁵ Formado por pares imágenes+texto.

sis de grandes cantidades de datos, lo que puede plantear preocupaciones en términos de privacidad y seguridad. Los modelos pueden aprender información sensible o confidencial a partir de los datos de entrenamiento, y existe el riesgo de que esta información sea utilizada de manera inapropiada o maliciosa.

Ejemplo: Supongamos que se entrena un modelo de lenguaje para generar respuestas automáticas en un servicio de chat. Si el modelo tiene acceso a las conversaciones de los usuarios, existe la posibilidad de que aprenda información personal, como números de tarjetas de crédito o detalles de salud, y esa información pueda ser comprometida si no se toman las medidas adecuadas para proteger la privacidad de los datos.

Desinformación y propagación de información falsa: Los modelos de lenguaje tienen la capacidad de generar texto muy convincente, lo que puede llevar a problemas relacionados con la desinformación y la propagación de información falsa. Si se utilizan de manera irresponsable o maliciosa, los modelos de lenguaje pueden generar contenido falso o engañoso que sea difícil de distinguir de la información verídica.

Ejemplo: Se podría usar un modelo de lenguaje para generar noticias de forma automática. Si este se utiliza con intenciones maliciosas, podría generar noticias falsas que se difundan rápidamente a través de las redes sociales, causando confusión y daño a la reputación de personas o instituciones.

En general, abordar estos desafíos éticos en los modelos de lenguaje requiere un enfoque multidisciplinario que involucra a expertos en ética, juristas, investigadores en NLP y responsables de la toma de decisiones. Es necesario establecer **políticas y marcos regulatorios** adecuados para garantizar la transparencia, equidad y responsabilidad en el desarrollo y despliegue de los modelos de lenguaje. Además, se deben implementar prácticas de **auditoría y evaluación continua** de los modelos para identificar y abordar sesgos y problemas éticos. Esto implica la recopilación y uso de conjuntos de datos más diversos y representativos, así como la implementación de técnicas de mitigación de sesgos y garantías de privacidad.

RETOS DEL CAPITULO 12

1. Trabajo en grupo: ensayar durante unos 10 minutos a hablar sin decir nada concreto. Estos discursos sin contenido son muy comunes en determinados ámbitos. Observa que, cuando lo consigas, el texto que generas cumple con las normas sintácticas e incluso semánticas [no tiene sentido, pero pueden pasar como texto real en algún contexto]; esto demostraría que en tu cerebro está implementado un modelo de lenguaje.
2. Pídele a tu IA favorita que te muestre quién es Noam Chomsky y su opinión acerca de los modelos de lenguaje en la mente humana.
3. Seguramente habrá miembros de tu aula con la experiencia de haber visto la evolución de un/una bebé hasta que domina el lenguaje. Calcular ¿cuanto tarda un/una bebé en aprender un lenguaje y cuantos bytes de información recibe en ese tiempo?
4. Pídele a tu agente conversacional favorito que genere texto sin sentido. Insiste hasta que genere una secuencia sin sentido pero cumpliendo las normas del idioma con el que trabajes.
5. Pregúntale a tu IA favorita, cómo es posible crear un modelo de lenguaje usando la probabilidad condicional.
6. Un *corpus* es uno de los elementos más importantes a la hora de crear un modelo de lenguaje. Averigua los tamaños de *corpus* que se están usando para entrenar a los modelos como *ChatGPT*, *Bard*, etc.
7. En este capítulo se ha enumerado una serie de tareas en donde la arquitectura *Transformers* ha demostrado un rendimiento sorprendente. Crea, a partir de estas tareas, una tarea más compleja, combinando algunas de las primeras.
8. Tu IA favorita, a quien le preguntas tus dudas, es con absoluta seguridad una arquitectura basada en *Transformers*. Pregúntale cómo, a partir de un modelo de lenguaje que genera texto, puede construirse un agente conversacional.
9. Con el tiempo, los *Transformers* han generado tres estrategias de modelos: *encoder-decoder*, sólo *encoder* y sólo *decoder*. Pídele a tu IA favorita que te muestre para cada uno de estas estrategias para que se utilicen habitualmente.
10. Cuando un modelo de lenguaje genera la siguiente palabra, lo que hace realmente es generar las probabilidades de todas las palabras en el vocabulario y se queda con la más probable. Esta estrategia (*greedy*) no es la más adecuada. Pregúntale a tu IA favorita qué otras estrategias hay.
11. Pídele a tu agente conversacional favorito, que cree un cuento. A partir de él, párrafo a párrafo pídele que lo aumente, aportando detalles. Sigue hasta que sea un cuento extenso. Al final, lee todo el cuento y comprueba si ha sido coherente.
12. Organizar debates en grupo sobre el impacto de la inteligencia artificial en el lenguaje y cómo los modelos de lenguaje están cambiando la forma en que nos comunicamos.

Capítulo 13

LA REPERCUSIÓN DEL PROCESAMIENTO DEL LENGUAJE NATURAL EN LA SOCIEDAD

Como acabamos de ver como el Procesamiento del Lenguaje Natural es una rama de la inteligencia artificial que se enfoca en la interacción entre las máquinas y el lenguaje humano. Su objetivo principal es permitir que estas comprendan, interpreten y generen lenguaje humano de manera eficiente y precisa.

El NLP tiene sus raíces en la década de 1950, cuando los investigadores comenzaron a explorar la posibilidad de enseñar a las máquinas a entender y generar lenguaje humano. A lo largo de las décadas, se han desarrollado técnicas y enfoques cada vez más sofisticados, aprovechando los avances en la inteligencia artificial y el aprendizaje automático.

13.1 Aplicaciones de la NLP en la sociedad

Asistentes virtuales

Los asistentes virtuales [o *chatbots*] son aplicaciones del NLP que interactúan con los usuarios utilizando lenguaje natural. Estos sistemas han revolucionado la forma en que las empresas ofrecen soporte al cliente y gestionan consultas y solicitudes. Estos asistentes virtuales pueden responder preguntas, proporcionar información, realizar tareas específicas [como configurar alarmas o reproducir música] y facilitar la interacción con otros dispositivos inteligentes.

Mi IA favorita dice cuando se le pregunta si es un chatbot: Sí, soy una especie de chatbot, aunque quizás un poco más avanzado que muchos otros modelos gracias a mi ar-

quitectura de aprendizaje automático subyacente. Estoy diseñado para asistir con una amplia variedad de tareas, desde responder preguntas y proporcionar explicaciones hasta generar texto y asistir en tareas de programación. Mi objetivo es comprender y generar texto en lenguaje natural de una manera que sea útil, informativa y, con suerte, precisa. Sin embargo, es importante tener en cuenta que, aunque puedo generar respuestas a preguntas y participar en conversaciones, no tengo la capacidad de entender o experimentar emociones.

Ánalysis de sentimientos y opiniones en redes sociales y medios de comunicación

El NLP permite analizar grandes volúmenes de texto en redes sociales y medios de comunicación para comprender los sentimientos y opiniones de las personas hacia ciertos temas o productos. Por ejemplo, las empresas pueden utilizar el análisis de sentimientos para evaluar la recepción de un nuevo producto o servicio, identificar problemas y mejorar la satisfacción del cliente.

Traducción automática y superación de barreras lingüísticas

El NLP ha avanzado significativamente en el campo de la traducción automática, permitiendo la traducción rápida y precisa entre diferentes idiomas. Por ejemplo, *Google Translate* o *DeepL* utilizan técnicas de NLP para ofrecer traducciones en tiempo real en una amplia gama de idiomas. Esto facilita la comunicación y supera las barreras lingüísticas en áreas como el turismo, los negocios internacionales y la colaboración académica.

Generación automática de contenido y resúmenes

El NLP también puede generar automáticamente contenido de texto, como noticias, informes y resúmenes. Por ejemplo, algunos medios de comunicación utilizan sistemas basados en PLN para generar noticias deportivas o informes financieros a partir de datos estructurados. Asimismo, los sistemas de resumen automático pueden analizar y condensar grandes volúmenes de texto, facilitando la comprensión y la extracción de información relevante.

Extracción de información y procesamiento de grandes volúmenes de texto

El NLP permite extraer información relevante de grandes conjuntos de texto no estructurado²¹⁶. Por ejemplo, los motores de búsqueda utilizan técnicas de NLP para comprender las consultas de los usuarios y mostrar resultados relevantes. Además, las empresas pueden utilizar estas técnicas para extraer información de documentos legales, informes médicos o artículos de investigación, facilitando la gestión y búsqueda de información en diversas industrias.

²¹⁶ El texto no estructurado es texto que no tiene una estructura definida. Por ejemplo texto de libros, artículos, sitios web, correos electrónicos, chats, etc.

13.2 Desafíos y limitaciones del NLP

Ambigüedad y polisemia del lenguaje natural

Uno de los mayores desafíos de NLP es lidiar con la ambigüedad inherente al lenguaje natural. Las palabras y oraciones pueden tener múltiples interpretaciones según el contexto. Por ejemplo, la palabra "banco" puede referirse a una institución financiera o a un asiento en un parque. El NLP necesita comprender el contexto para determinar la interpretación correcta.

Tratamiento de idiomas menos comunes o con estructuras lingüísticas complejas

NLP está más desarrollado en idiomas muy extendidos, como el inglés, el español y el chino. Sin embargo, los idiomas menos comunes presentan un desafío adicional debido a la falta de recursos lingüísticos disponibles, como grandes conjuntos de datos o modelos preentrenados. Además, algunos idiomas tienen estructuras gramaticales más complejas, lo que dificulta aún más el procesamiento automático del lenguaje. Esto puede llevar a una calidad inferior en estos idiomas y limitar su adopción.

Por ejemplo, el idioma navajo, hablado por la tribu navajo en América del Norte, tiene una estructura lingüística altamente compleja. La traducción automática o el análisis de sentimientos en navajo son un desafío debido a la falta de recursos y a la necesidad de comprender sus reglas gramaticales.

Nota: Meta (antiguamente facebook) AI²¹⁷ dice en su web, “No Language Left Behind (NLLB) es un innovador proyecto de inteligencia artificial, primero en su tipo, que ofrece modelos de código abierto capaces de ofrecer traducciones evaluadas y de alta calidad directamente entre 200 idiomas, incluidos idiomas de bajos recursos como el asturiano, el luganda, el urdu y más. Su objetivo es brindar a las personas la oportunidad de acceder y compartir contenido web en su idioma nativo y comunicarse con cualquier persona, en cualquier lugar, independientemente de sus preferencias lingüísticas.”

Entendimiento de contexto y sarcasmo

El NLP a menudo se enfrenta a dificultades para comprender y capturar el contexto adecuado. El significado de una oración puede cambiar según la información previa o el contexto en el que se utiliza. Por ejemplo, la oración "Hace frío aquí" puede ser una simple declaración de temperatura o una indirecta para pedir que se cierre la ventana.

Del mismo modo, el sarcasmo es un desafío ya que implica un significado opuesto o irónico al literal. Detectar y comprender el sarcasmo requiere un conocimiento profundo del contexto y del tono empleado. Por ejemplo, en la oración "¡Oh, fantástico! Otro día de lluvia", el tono sarcástico indica que no se está contento con el clima, a pesar de usar la palabra "fantástico".

217 <https://ai.meta.com/research/no-language-left-behind>

Para poder superar este desafío los modelos de lenguaje futuro deberían tener en cuenta, además del texto, las entonaciones del habla y el lenguaje corporal, especialmente el lenguaje que transmiten nuestras facciones.

Privacidad y ética en el procesamiento de datos lingüísticos sensibles

NLP implica el procesamiento de grandes cantidades de datos de carácter lingüístico, que a menudo pueden contener información sensible o privada. Esto plantea preocupaciones sobre la privacidad y la ética en el manejo de esos datos. La extracción de información personal, como nombres, direcciones o información médica, puede resultar problemática si no se maneja adecuadamente.

Además, existe el riesgo de sesgos y discriminación en sus aplicaciones si los modelos se entrenan con datos que reflejan prejuicios o desigualdades sociales. Por ejemplo, si un sistema de recomendación de empleo se entrena con datos históricos sesgados, puede perpetuar la discriminación de género o racial en las recomendaciones futuras.

13.3 Impacto del NLP en la comunicación y la interacción social

Implicaciones para la traducción y la multiculturalidad

NLP está teniendo un gran impacto en la traducción automática, lo que está facilitando la comunicación entre personas que hablan diferentes idiomas. Existen plataformas que utilizan algoritmos de NLP para analizar y traducir texto en tiempo real. Esto ha contribuido a la superación de barreras lingüísticas y ha promovido la multiculturalidad al permitir que las personas se comuniquen y comprendan mejor entre sí. Por ejemplo, en situaciones de viaje, el uso de aplicaciones de traducción automática puede ayudar a los turistas a comunicarse con los locales y explorar nuevas culturas de manera más fluida.

Nuevas oportunidades en la educación y el aprendizaje de idiomas

NLP ha abierto nuevas posibilidades en el ámbito educativo y el aprendizaje de idiomas. Las aplicaciones de aprendizaje de idiomas basadas en NLP pueden proporcionar ejercicios interactivos, retroalimentación personalizada y tutoriales automatizados que se adaptan al nivel y ritmo de cada estudiante. Además, los sistemas de reconocimiento de voz y retroalimentación pueden ayudar a los estudiantes a mejorar su pronunciación y fluidez en el idioma objetivo. Por ejemplo, Duolingo y Babbel [entre otras] son plataformas populares que utilizan esta tecnología para enseñar idiomas de manera efectiva y accesible para un gran número de personas.

Accesibilidad y superación de barreras para personas con discapacidades

NLP ha contribuido a mejorar la accesibilidad para personas con discapacidades. Por ejemplo, los sistemas de reconocimiento de voz permiten a las personas con discapacidad motora comunicarse de manera más eficiente utilizando su voz en lugar de teclados o dispositivos de entrada física. Además, los lectores de pantalla basados en NLP pueden convertir texto en voz, lo que facilita el

acceso a la información para personas con discapacidad visual. Estas tecnologías han abierto nuevas oportunidades y han permitido una mayor inclusión social.



Figura 56: Stephen Hawking fotografiado en su silla dotada de un sistema de PLN

Creador: Jude Edginton, Imagen propiedad de Discovery Channel (CC BY 2.0)

13.4 Futuro del NLP

Tendencias y avances recientes

El campo del Procesamiento del Lenguaje Natural ha experimentado avances significativos en los últimos años, impulsados por el desarrollo de técnicas de aprendizaje automático (*deep learning*) y el acceso a grandes volúmenes de datos. Algunas de las tendencias y avances recientes incluyen:

- **Modelos de lenguaje basados en *transformers*:** Los modelos de lenguaje basados en la arquitectura *transformer*, como la familia de modelos GPT (*Generative Pre-trained Transformer*) y BERT (*Bidirectional Encoder Representations from Transformers*), han revolucionado el campo. Estos modelos han logrado un rendimiento sobresaliente en tareas como la traducción automática, el análisis de sentimientos y la generación de texto.
- **Aprendizaje de representaciones sin supervisión:** Los enfoques de aprendizaje sin supervisión, como el preentrenamiento y el ajuste fino de modelos de lenguaje, han demostrado ser altamente efectivos para mejorar el rendimiento en diversas tareas. Estos enfoques permiten que los modelos aprendan patrones y características del lenguaje a partir de grandes corpus de texto sin la necesidad de etiquetas o anotaciones específicas.
- **Mejoras en la traducción automática:** La traducción automática neuronal ha alcanzado niveles de precisión sorprendentes gracias a los enfoques basados en redes neuronales. Las técnicas de traducción automática han mejorado la comunicación y el intercambio de información entre personas que hablan diferentes idiomas, facilitando la globalización y la colaboración internacional.

- **RAG** (*Retrieval-Augmented Generation*) es un enfoque avanzado en el campo del Procesamiento del Lenguaje Natural que combina técnicas de recuperación de información y generación de texto para mejorar la capacidad de un modelo de lenguaje. El término "RAG" se refiere a un modelo que no solo genera texto basado en un *prompt*, sino que también consulta una base de datos externa o un conjunto de documentos para recuperar información relevante que enriquezca o verifique la generación del texto.

Alguna de sus aplicaciones son:

- 1] Sistemas de Preguntas y Respuestas [QA]: RAG es ideal para sistemas de QA donde se necesita recuperar respuestas específicas a partir de una gran base de conocimientos. En lugar de confiar solo en el conocimiento almacenado dentro de un modelo durante su entrenamiento, el sistema puede buscar información en bases de datos externas para proporcionar respuestas más precisas y relevantes.
- 2] Asistentes Virtuales y Chatbots: Los asistentes virtuales que utilizan RAG pueden ofrecer respuestas más precisas y actualizadas a las preguntas de los usuarios al recuperar y utilizar información de bases de datos actualizadas, documentos legales, manuales técnicos, etc.
- 3] Generación de Contenido Basado en Datos: En la creación de artículos, resúmenes o cualquier contenido escrito, RAG puede ser utilizado para consultar información precisa de documentos externos y generar contenido que sea tanto informativo como correcto.
- 4] Investigación Médica y Técnica: En campos que requieren un alto grado de precisión, como la medicina o la ingeniería, los sistemas RAG pueden ayudar a consultar bases de datos especializadas y generar diagnósticos, recomendaciones de tratamiento, o informes técnicos basados en la información más reciente.

RETOS DEL CAPITULO 13

1. Las propiedades emergentes de un sistema son aquellas propiedades que existen en dicho sistema pero que dejan de existir si reducimos el tamaño o la complejidad del mismo.

A medida que los modelos de lenguaje se hicieron más grandes (*Large Language models, LLM*), fueron apareciendo propiedades inesperadas en ellos, bien de forma espontánea (*zero-shot*) o mediante un pequeño ajuste (*one-shot*).

Busca en la web un listado de propiedades emergentes de estos LLM.

2. Busca el significado de los siguientes conceptos:

- *Fine tuning.*
- *Zero-shot.*
- *One-shot.*
- *Learning transfer.*

3. Divide el aula en dos y organizar un debate con los siguientes puntos:

- ¿Cómo se pueden usar los LLM para engañar a las personas?
- A qué tipos de trabajo podría afectar los LLM y sus capacidades.
- ¿Cómo influirá en las lenguas la existencia de traductores de altísima calidad?
- ¿Será posible una inteligencia como la humana por medio de LLMs?
- Si a medida que crece el tamaño de los modelos de lenguaje aparecen más propiedades emergentes, ¿podrían aparecer propiedades que no poseemos los seres humanos?

Sección IV: Aprendizaje automático

"El aprendizaje automático es el campo que permite a las máquinas aprender de la experiencia y mejorar su rendimiento en tareas específicas a medida que se exponen a más datos."

Pedro Domingos (1962 -)

La última sección, antes de los anexos, de esta obra está dedicada al aprendizaje automático, que es la disciplina que más influencia tiene en estos momentos en la sociedad y promete mejores resultados en un futuro próximo.

En ella expondremos lo que son los algoritmos y modelos, qué es la ciencia de los datos, cómo se entrena y evalúa un modelo, las redes neuronales artificiales (*deep learning*) y, finalmente, los beneficios y riesgos del aprendizaje automático en la sociedad.

Capítulo 14

ALGORITMOS Y MODELOS

Ya llevamos un buen recorrido a lo largo de las tecnologías inteligentes, sin embargo no hemos entrado en profundidad en sus métodos o, dicho de otro modo, no hemos aprendido por qué funcionan. Los siguientes capítulos, probablemente los más importantes del libro, quieren tapar estos huecos o dudas que a las mentes más ávidas y curiosas podría haber generado; y, al mismo tiempo, elevar a un nivel superior las “tecnologías inteligentes”, mostrando las bases de su funcionamiento: ¿cómo se consigue que un algoritmo aprenda automáticamente a partir de ejemplos?

14.1 Introducción

En la actualidad, estamos rodeados de tecnología y datos que nos rodean en todas las áreas de nuestra vida. Desde redes sociales hasta servicios de *streaming*, pasando por motores de búsqueda y sistemas de recomendación, etc. el mundo digital está en constante evolución y crecimiento. Pero, ¿cómo es posible que estas plataformas y servicios puedan brindarnos experiencias personalizadas, precisas y eficientes? La respuesta radica en el Aprendizaje Automático y, más específicamente, en los **algoritmos y modelos** que impulsan esta tecnología.

El Aprendizaje Automático [*machine learning*] es una rama de la Inteligencia Artificial que se centra en desarrollar sistemas capaces de **aprender a partir de los datos**. En lugar de ser programados de manera explícita, estos sistemas pueden adaptarse y mejorar sus resultados a medida que se exponen a más información. El objetivo principal del Aprendizaje Automático es permitir a las máquinas "aprender" de manera similar a como lo haría un ser humano, identificando patrones, tomando decisiones y/o realizando predicciones.

Los algoritmos son el corazón del Aprendizaje Automático. Estos son conjuntos de instrucciones y reglas lógicas que permiten a las máquinas procesar y analizar los datos para extraer información útil. En otras palabras, los algoritmos son como "recetas" que guían a las máquinas en la forma en que deben procesar la información y tomar decisiones.

Mi IA favorita dice: *Un algoritmo es un conjunto finito y ordenado de pasos o instrucciones que se sigue para resolver un problema específico o realizar una tarea. Los al-*

goritmos pueden implementarse en diversos contextos, desde cálculos matemáticos y procesamiento de datos hasta toma de decisiones y automatización, y son la base de muchos sistemas de computación y programas de software.

Por ejemplo, imagina que tienes una colección de fotografías y deseas crear un sistema de reconocimiento facial. El algoritmo de Aprendizaje Automático podría analizar todas las imágenes, identificar patrones en los rasgos faciales y aprender a distinguir diferentes personas. A medida que el sistema se expone a más imágenes, su precisión y capacidad para reconocer caras mejora, ya que el algoritmo aprende de cada imagen y refina su capacidad de identificación.

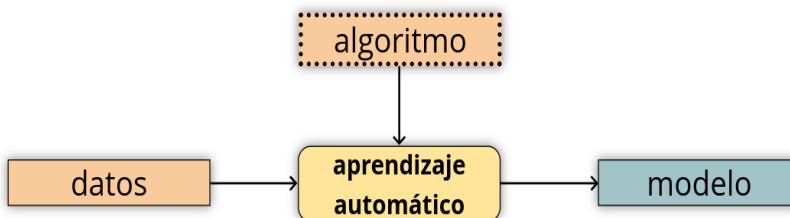


Figura 57: Relación entre datos, algoritmo y modelo en el aprendizaje automático.

Ya hemos hablado de modelos anteriormente, así que vamos a recordar el concepto. **Los modelos** son representaciones matemáticas de los datos y las relaciones que se encuentran dentro de ellos. En el contexto que nos ocupa, estos modelos son construidos por los algoritmos de Aprendizaje Automático y se utilizan para realizar predicciones o clasificaciones basadas en los datos de entrada. Los modelos pueden ser simples, como una línea recta en un gráfico, o más complejos, como redes neuronales con múltiples capas (*deep learning*).

¿Cómo afrontaríamos el mismo problema sin Aprendizaje Automático? De manera tradicional: una persona o equipo de personas, a partir de unos datos de ejemplo y/o de unas especificaciones técnicas (un análisis de requerimientos) diseñan y codifican (y prueban) un sistema informático, que hace las veces de modelo.

Tomemos otro ejemplo para comprender mejor cómo funcionan. Supongamos que queremos **predecir** el precio de una vivienda en función de sus características, como el tamaño, la ubicación y el número de habitaciones, entre otras. El modelo de Aprendizaje Automático podría analizar un conjunto de datos de viviendas anteriores, aprender las relaciones entre las características y los precios, y luego utilizar ese conocimiento para predecir el precio de una nueva vivienda en función de sus características.

Todas las tecnologías inteligentes que hemos nombrado hasta ahora, y las que nos hemos dejado en el tintero, se basan en su mayor parte en técnicas de Aprendizaje Automático; bien con un enfoque estocástico o algorítmico (*machine learning*), bien mediante una arquitectura más compleja de redes neuronales artificiales (*deep learning*).



Figura 58: Esquematización del método tradicional de producción de sistemas informáticos.

14.2 Aprendizaje automático y ciencia de datos.

La ciencia de datos y el aprendizaje automático son dos disciplinas estrechamente relacionadas que se ocupan del análisis y la interpretación de datos para obtener información y tomar decisiones. Aunque están relacionadas, cada una tiene su propio enfoque y conjunto de técnicas.

La ciencia de datos (lo veremos con más detenimiento en el Capítulo 15) es un campo interdisciplinario que combina conocimientos de matemáticas, estadísticas, programación y dominio del área en cuestión, para extraer conocimiento a partir de los datos. Se enfoca en el proceso de recopilar, limpiar, analizar y visualizar grandes volúmenes de información con el objetivo de descubrir patrones, identificar tendencias y resolver problemas complejos. La ciencia de datos utiliza herramientas y técnicas estadísticas y matemáticas para explorar y comprender los datos.

Por otro lado, como ya hemos dicho, el aprendizaje automático es una rama de la inteligencia artificial que se basa en la idea de que las máquinas pueden aprender automáticamente a partir de los datos sin ser programadas explícitamente. En lugar de seguir reglas predefinidas, los algoritmos de aprendizaje automático aprenden a través de la experiencia por medio de los datos de entrenamiento. El aprendizaje automático se utiliza para desarrollar modelos y algoritmos que pueden realizar tareas específicas, como clasificación, predicción, reconocimiento de patrones y toma de decisiones, entre otros objetivos.

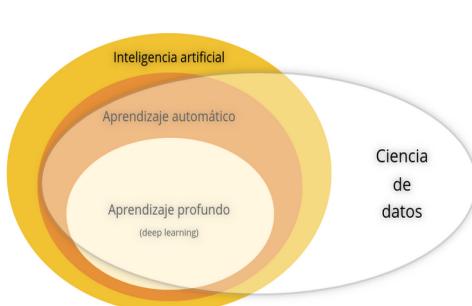


Figura 59: Relación entre inteligencia artificial y ciencia de datos.

Entonces, ¿cómo se relacionan la ciencia de datos y el aprendizaje automático? El aprendizaje automático es una parte integral de la ciencia de datos y una de las herramientas clave utilizadas para extraer conocimiento. La ciencia de datos proporciona el marco teórico y metodológico para abordar problemas complejos relacionados con los datos, y el aprendizaje automático ofrece las técnicas y algoritmos necesarios para analizar y extraer información útil de esos

datos. Un ejemplo claro de cómo se relacionan estas dos disciplinas sería el análisis predictivo:

Supongamos que tenemos un conjunto de datos que contiene información demográfica y registros de compras de clientes. Usando técnicas de ciencia de datos, podemos explorar los datos, identificar variables relevantes y realizar un análisis estadístico para comprender el comportamiento de compra de los clientes.

Una vez que tenemos una comprensión sólida de los datos, podemos aplicar técnicas de aprendizaje automático, como regresión²¹⁸ o clasificación, para desarrollar modelos. Estos modelos pueden predecir, por ejemplo, qué productos es más probable que compre un cliente en función de sus características demográficas y su historial de compras.

La ciencia de datos y el aprendizaje automático se complementan mutuamente en el proceso de análisis de datos y toma de decisiones. La ciencia de datos proporciona el marco teórico y las herramientas para abordar problemas complejos relacionados con los datos, mientras que el aprendizaje automático ofrece las técnicas y los algoritmos para extraer información útil y desarrollar modelos predictivos o descriptivos. Ambas disciplinas son fundamentales en el mundo actual, donde los datos son abundantes y la capacidad de extraer conocimientos valiosos de ellos es esencial para resolver problemas y tomar decisiones informadas.

14.3 Algoritmos en aprendizaje automático

En el contexto del Aprendizaje Automático, los algoritmos desempeñan un papel fundamental al permitir que las máquinas aprendan a partir de los datos y realicen tareas específicas sin ser explícitamente programadas. Estos algoritmos son responsables de tomar decisiones basadas en patrones y características presentes en los datos de entrada. A continuación, explicaremos de nuevo los diferentes tipos de algoritmos de aprendizaje automático [§2.1] junto con ejemplos ilustrativos, por medio de la clasificación más usada.

14.3.1 Algoritmos supervisados

Los algoritmos supervisados se basan en conjuntos de **datos etiquetados**, es decir, conjuntos de datos en los que se conoce la respuesta deseada. Estos algoritmos utilizan estas etiquetas para aprender a predecir la salida correcta para nuevas instancias de datos.

En la Figura 60 podemos ver parte de los datos originales de un *dataset* clásico, el conjunto de datos recopilados de la flor *Iris*, la cual se expresa como tres especies relacionadas: *iris setosa*, *iris versicolor* e *iris virginica*. Estas tres especies son apenas diferenciables a simple vista, así que en 1936 se recogieron²¹⁹ 150 muestras y, para cada una de ellas, se etiquetó con su especie corres-

²¹⁸ Me adelantaré un poco: la regresión se refiere a un tipo de algoritmo supervisado que tiene como objetivo predecir una variable continua o cuantitativa. A diferencia de la clasificación, que asigna etiquetas categóricas a las entradas, la regresión busca modelar y entender la relación entre variables para predecir un resultado numérico. En general, ambos, regresión y clasificación, buscan predecir algo partiendo de los datos.

²¹⁹ Artículo original: <https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1469-1809.1936.tb02137.x>

pondiente. Por lo tanto, cada muestra consta de 5 columnas: longitud y anchura del sépalo, longitud y anchura del pétalo, especie a la que pertenece (ver Figura 61).

Table I

<i>Iris setosa</i>				<i>Iris versicolor</i>			<i>Iris virginica</i>				
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
5.0	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3.0	5.8	2.2
5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3.0	6.6	2.1
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5.0	3.4	1.5	0.2	4.9	2.4	3.3	1.0	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8

Figura 60: Parte de la tabla original del dataset clásico Iris.

Fuente: "Annals of eugenics", Volume 7, issue 2. Sep. 1936.

El problema que se busca solucionar es: ¿a partir de esos datos, es posible generalizar y aprender (esto es, crear un modelo) de forma que podamos asignar una especie a una flor sólo con recolectar sus datos de longitud y anchura de sépalo y pétalo?

Los algoritmos supervisados de aprendizaje automático, necesitan la solución (la etiqueta) para aprender la relación que hay entre las **columnas de características** (también llamadas variables independientes) y la **columna etiqueta** (también llamada variable dependiente, ver Figura 61).

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
5.0	2.0	3.5	1.0	versicolor
6.2	2.2	4.5	1.5	versicolor
6.0	2.2	4.0	1.0	versicolor
6.0	2.2	5.0	1.5	virginica
6.3	2.3	4.4	1.3	versicolor
5.5	2.3	4.0	1.3	versicolor
5.0	2.3	3.3	1.0	versicolor
4.5	2.3	1.3	0.3	setosa
5.5	2.4	3.8	1.1	versicolor
5.5	2.4	3.7	1.0	versicolor
4.9	2.4	3.3	1.0	versicolor
6.7	2.5	5.8	1.8	virginica

Figura 61: Datos tabulados de dataset parcial Iris.

La idea en general es:

1. Entrenar con datos de la forma $X_1, X_2, \dots, X_n \rightarrow Y$ y obtener un modelo con una tasa de error conocida y asumible.
2. Una vez tengamos un modelo, inferir a partir de nuevos X_1, X_2, \dots su Y correspondiente.

Los algoritmos supervisados se pueden a su vez dividir en dos, según sea la naturaleza de su etiqueta, esto es, de lo que queremos predecir:

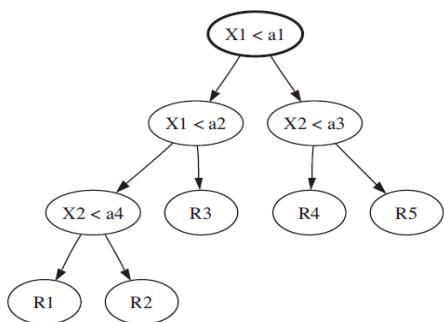
- Si es un número continuo, por tanto toma valores infinitos, estamos prediciendo un valor; por tanto estamos ante una **regresión**²²⁰.
- Si la etiqueta toma valores finitos, como las especies *Iris*, estamos ante un problema de **clasificación**.

A continuación veremos algunos ejemplos de algoritmos supervisados.

Regresión lineal

La regresión lineal es un algoritmo utilizado para predecir una variable continua basada en la relación lineal entre las variables de entrada y la de salida. Por ejemplo, supongamos que queremos predecir el precio de una casa en función de su tamaño. Utilizando un conjunto de datos históricos que incluya el tamaño y el precio de diferentes casas, el algoritmo de regresión lineal aprenderá a trazar una línea que mejor se ajuste a los puntos y permita predecir el precio de una casa nueva en función de su tamaño.

Figura 62: Ejemplo de árbol de decisión



Árboles de decisión

Los árboles de decisión se utilizan para predecir una salida discreta²²¹ mediante una secuencia de preguntas basadas en las características de la entrada, formando así una especie de árbol invertido. Cada nodo del árbol representa una pregunta y las ramas corresponden a las posibles respuestas o nuevas preguntas.

Por ejemplo, supongamos que queremos clasificar correos electrónicos como "spam" o "no spam" en función de ciertas características, como la presencia de palabras clave específicas. Un árbol de decisión puede hacer preguntas como "¿El correo electrónico contiene la palabra 'oferta'?" y, en función de las respuestas [sí/no en este caso], llegar a una clasificación final [es o no es *spam*].

Regresión logística

La Regresión Logística es un algoritmo de aprendizaje supervisado utilizado principalmente para

220 El término "regresión" proviene originalmente de estudios en genética y biología realizados por Francis Galton en el siglo XIX. Galton estaba investigando la relación entre padres e hijos en lo relativo a su altura. Observó un fenómeno que él llamó "regresión hacia la media", que significa que los hijos de padres con alturas extremas (tanto altos como bajos) tendían a tener alturas más cercanas a la media de la población. La "regresión" en el contexto que nos ocupa no implica necesariamente una "regresión hacia la media" como en el estudio original de Galton. En lugar de ello, se refiere a la tarea de predecir una variable continua. Este uso del término se ha vuelto estándar en estadísticas y aprendizaje automático.

221 También se pueden usar para regresión. Pero lo habitual es usarlos para clasificación.

problemas de clasificación binaria [dos clases], aunque también se puede adaptar para clasificación multiclase²²². Es una extensión del modelo de Regresión Lineal que utiliza la función logística [o *sigmoide* §5.2] para modelar la probabilidad de que una entrada pertenezca a una determinada clase. Este modelo es especialmente útil cuando se quiere estimar una probabilidad que puede traducirse en una decisión binaria [por ejemplo, si un correo electrónico es *spam* o no].



Figura 63: Ejemplo de aprendizaje automático a partir de datos usando árboles de decisión.

Fuente: www.saedsayad.com; autor: Dr. Saed Sayad.

Random forest [bosques aleatorios]

Los Bosques Aleatorios es un método de ensamblaje que combina múltiples árboles de decisión para crear un modelo más fuerte y robusto. Cada árbol se construye utilizando un subconjunto aleatorio de las características y un subconjunto aleatorio de las muestras del conjunto de datos. Las predicciones de los árboles individuales se combinan para dar una respuesta final, lo cual generalmente mejora la precisión y reduce el riesgo de sobreajuste. Este algoritmo se utiliza tanto para tareas de clasificación como de regresión y es conocido por su flexibilidad y robustez.

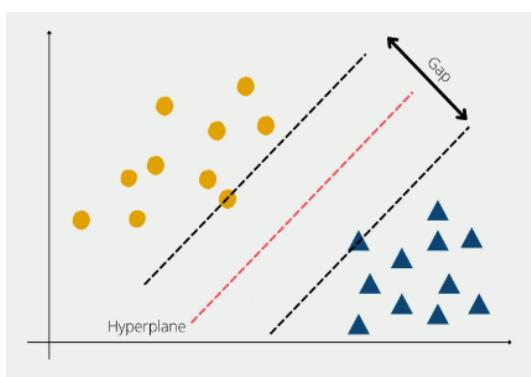


Figura 64: Explicación de SVM

Fuente: <https://databasecamp.de/en/ml/svm-explained>

Máquinas de soporte vectorial [SVM]

Las Máquinas de Soporte Vectorial son algoritmos que buscan encontrar un hiperplano [o conjunto de hiperplanos en espacios de alta dimensión] que mejor separe las diferentes clases de datos. Están diseñadas para clasificación y también se pueden utilizar para tareas de regresión. La principal ventaja de SVM es su eficacia en espacios de alta dimensión y su habilidad para encontrar límites de decisión complejos. Se pueden utilizar diferentes *kernels*²²³ para transformar el espacio de características y encontrar un hiperplano que maximice el margen entre clases.

222 Por ejemplo, el dataset Iris es un problema de clasificación de tres clases: *satosa*, *virginica* y *versicolor*.

SVM es un gran clasificador, y su funcionamiento, como gran parte de los conceptos de estamos viendo, son muy simples (pero con un complejo desarrollo matemático).

Un clasificador binario aprende a partir de los datos de entrenamiento a asignar a cada entrada una etiqueta [en la Figura 64, la entrada posee 2 dimensiones y pretende clasificar cada punto como círculo o triángulo].

SVM calcula, a partir de los ejemplos con los que se le entrena, el mejor “pasillo” que separa ambos conjuntos de datos; de tal manera que *predice*, para nuevos datos, a que grupo pertenecerán. Si en vez de tener 2 dimensiones tenemos, por ejemplo, 1000; en vez de hablar de “pasillo” hablamos de hiperplano.

Un hiperplano es una superficie de dimensión $n-1$ en un espacio de n dimensiones que separa ese espacio en dos mitades. En caso de 2 dimensiones, una línea es un hiperplano. En 3 dimensiones sería un plano.

Veamos el código que con SVM crea un modelo para predecir si un cáncer de mama es o no benigno, partiendo de unos datos de entrenamiento:

```

1  from sklearn import datasets
2  from sklearn.model_selection import train_test_split
3  from sklearn.svm import SVC
4  from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
5
6  # Cargar el conjunto de datos de cáncer de mama
7  cancer_data = datasets.load_breast_cancer()
8  X = cancer_data.data
9  y = cancer_data.target
10
11 # Dividir los datos en conjuntos de entrenamiento y prueba
12 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
13
14 # Crear el clasificador SVM
15 svm_classifier = SVC(kernel='linear', C=1)
16
17 # Entrenar el clasificador
18 svm_classifier.fit(X_train, y_train)

```

Posteriormente llevamos a cabo la predicción y y la evaluación del modelo:

```

1 # Realizar predicciones en el conjunto de prueba
2 y_pred = svm_classifier.predict(X_test)
3
4 # Evaluar el modelo
5 print("Matriz de confusión:")
6 print(confusion_matrix(y_test, y_pred))
7 print("\nInforme de clasificación:")
8 print(classification_report(y_test, y_pred))
9 print("\nExactitud:")
10 print(accuracy_score(y_test, y_pred))

```

223 Técnica matemática que permite clasificar conjunto de datos – en principio – inclasificables, por no poder separarlos por un hiperplano.

El resultado de este ejecución dos muestra una matriz de confusión e información sobre la precisión, la sensibilidad [*recall*] y un F1-score. Por ahora no importan estos datos, sólo saber que **hemos creado un modelo que con una exactitud del 96.5% es capaz de predecir si una biopsia muestra un cáncer de mama benigno o no** a partir de los datos de entrada²²⁴.

En el código anterior tenemos una variable [objeto *svm_classifier*, en negrita] que apunta a un modelo que puede ser almacenado en disco y vuelto a cargar en memoria posteriormente, y que es capaz de predecir [función *svm_classifier.predict*] si una muestra obtenida puede ser o no cáncer de mama benigno con más de un 96% de exactitud.

14.3.2 Algoritmos no supervisados

A diferencia de los algoritmos supervisados, los algoritmos no supervisados se utilizan cuando no se dispone de etiquetas o respuestas previas. Estos algoritmos encuentran patrones y estructuras ocultas en los datos sin ninguna guía explícita. El objetivo es encontrar las relaciones intrínsecas que existen entre los datos.

A continuación, se presentan algunos ejemplos de tipos de algoritmos no supervisados:

Clustering (agrupamiento)

El *clustering* agrupa instancias de datos similares en grupos o clústeres. El objetivo es que las instancias dentro de un clúster sean similares entre sí y diferentes de las instancias en otros clústeres. Por ejemplo, si tenemos datos de clientes de una tienda en línea, el algoritmo de *clustering* puede agrupar a los clientes en diferentes segmentos basados en sus hábitos de compra o preferencias, permitiendo así a la empresa adaptar su estrategia de marketing para cada segmento.

Podemos tomar un dataset como *Iris* y quitarle la columna de especie [la columna objetivo]; y ya tenemos un dataset susceptible de ser tratado con un algoritmo no supervisado [no tenemos la columna que nos dice de que especie es cada muestra]. El objetivo es, a partir de las características, agruparlas de tal manera que se muestren las diferencias que permite reunir las muestras en grupos [ver Figura 65]. La ventaja de esta aproximación es que posteriormente podemos recuperar la columna objetivo y comprobar que tal fue el agrupamiento. Sin embargo no siempre tendremos esa ventaja.

Reducción de dimensionalidad

La reducción de dimensionalidad se utiliza para reducir la cantidad de variables en un conjunto de datos, conservando la mayor cantidad posible de información relevante. Esto es especialmente útil cuando se trabaja con conjuntos de datos de alta dimensionalidad. Un ejemplo común de reducción de dimensionalidad es el Análisis de Componentes Principales [PCA], que transforma los datos originales en un nuevo conjunto de variables no correlacionadas llamadas componentes principales.

²²⁴ El conjunto de datos de cáncer de mama en scikit-learn (Breast Cancer dataset) tiene 30 características numéricas (por tanto 30 dimensiones) que se calculan a partir de una imagen digitalizada de una biopsia de una masa mamaria. Estas características describen las cualidades del núcleo de las células presentes en la imagen y se dividen en tres grupos: las estadísticas del área, las estadísticas de la textura y las estadísticas del perímetro de la célula.

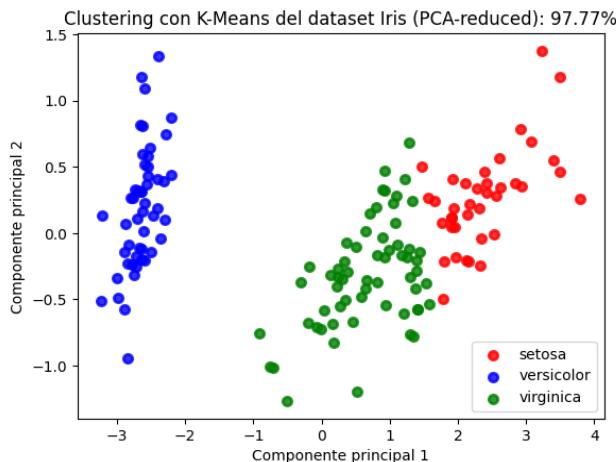
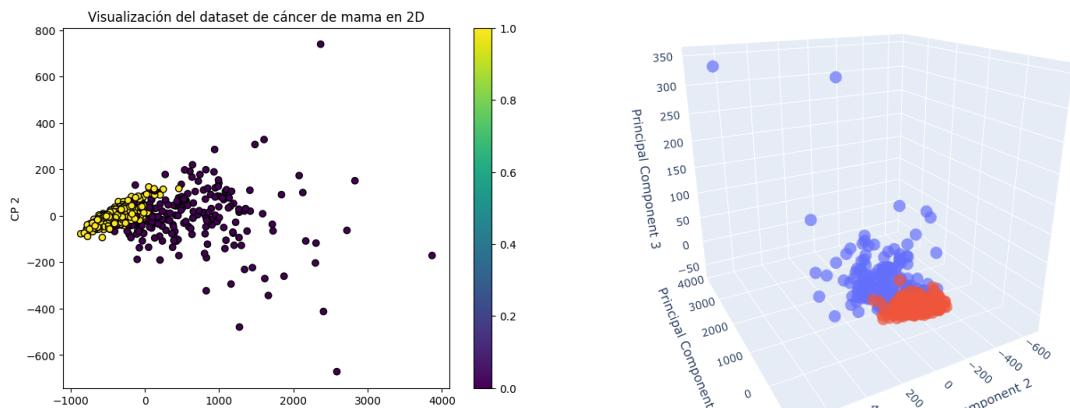


Figura 65: Representación en 2D de las variables independientes del dataset Iris.

Por ejemplo, el *dataset* que predecía si un cáncer de mama era benigno o no, posee 30 dimensiones, y visualizarlo por completo es imposible para nosotros. Pero podemos recurrir a este algoritmo para reducir la dimensionalidad, conociendo la cantidad de información que conservamos:



Visualización en 2D del dataset conservando el 99.82% de la información original Visualización en 3D interactiva del dataset conservando el 99.98% de la información original

14.3.3 Algoritmos de Aprendizaje por Refuerzo

Los algoritmos por refuerzo son una forma de aprendizaje automático que permite a las máquinas (un agente) aprender a tomar decisiones de manera autónoma. Estos algoritmos simulan un entorno en el que el agente puede tomar acciones y recibir recompensas o castigos en función de sus resultados. A medida que este toma más decisiones/acciones, aprende cuáles le dan más recompensas y se vuelve más hábil.

Imagina que estás jugando a un videojuego. Al principio, quizás no sepas muy bien qué hacer, cuáles botones apretar o cómo moverte para ganar puntos. Pero a medida que avanzas, empiezas a entender qué acciones te dan más puntos y cuáles te hacen perder. Eventualmente, quizás, te vuelves muy bueno en el juego porque has aprendido las mejores estrategias para ganar.

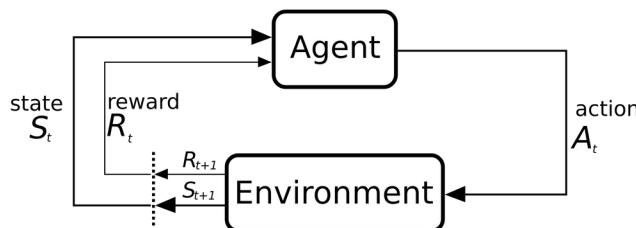


Figura 66: Esquema de un sistema de aprendizaje por refuerzo.

Fuente: Wikimedia commons

Los algoritmos de refuerzo funcionan de manera similar. Son una forma de aprendizaje automático en la que el agente aprende a tomar decisiones al interactuar con un entorno (que podría ser un juego, un laberinto, un mercado de acciones, etc.). Este recibe "recompensas" o "penalizaciones" en función de las acciones que toma, y con el tiempo, aprende a tomar las decisiones que maximizan su recompensa total. Es como entrenar a un perro con golosinas cuando hace algo bien, y diciéndole "no" cuando hace algo mal, hasta que aprende a comportarse de la manera que deseas.

14.4 Evaluación de modelos

Ponte en esta situación: has creado un modelo a partir de unos datos de entrenamiento y un algoritmo. ¿Cómo es de bueno tu modelo?

La evaluación del rendimiento de los modelos de aprendizaje automático es una etapa muy importante en donde se pone a prueba el modelo, con el objetivo de entender qué tan bien ha aprendido y cómo generaliza cuando se le presentan nuevos datos. Esta evaluación se lleva a cabo por medio de métricas. Estas métricas te permiten seleccionar el mejor modelo entre varios candidatos²²⁵. Aquí hay algunas técnicas y métricas comunes para tareas de clasificación binaria²²⁶:

- **Precisión:** indica el porcentaje de verdaderos positivos entre todas las predicciones positivas (verdaderas y falsas) realizadas por el modelo.

$$P = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos positivos}}$$

²²⁵ Pues sí. Lo normal es crear varios modelos (quizás partiendo de varios algoritmos y/o modificando los hiperparámetros), seleccionando el que mejor ha aprendido. Es como escoger a la persona adecuada para un punto de trabajo por medio de un examen de conocimientos y aptitudes.

²²⁶ En el Capítulo 16 veremos la evaluación de modelos con más profundidad.

- **Sensibilidad [recall]**: es la proporción de ejemplos positivos que fueron identificados correctamente por el clasificador respecto al total de positivos reales [verdaderos positivos y falsos negativos].

$$R = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos negativos}}$$

- **F1-score**: Combina la precisión y la sensibilidad en una única métrica, proporcionando un equilibrio entre ambas. Se calcula como la media armónica de ambas.

Nota: En términos simples, la precisión nos indica si hemos clasificado erróneamente instancias positivas como negativas, mientras que la sensibilidad nos indica si hemos perdido instancias positivas.

Y para las tareas de regresión:

- **Error absoluto medio [MAE]**: Es la media del valor absoluto de los errores.
- **Error cuadrático medio [MSE]**: Es la media del cuadrado de los errores.
- **Raíz del error cuadrático medio [RMSE]**: Es la raíz cuadrada del MSE.

Otras técnicas relacionadas con la evaluación de manera robusta y sistemática del rendimiento de los modelos, son la validación cruzada y la división del conjuntos de datos.

- **La validación cruzada**: consiste en dividir el conjunto de datos de entrenamiento en varios subconjuntos, y luego entrenar el modelo en algunos de los subconjuntos y evaluarlo en los otros. Esto permite estimar el rendimiento del modelo con datos nuevos, que es lo que realmente importa en el mundo real, y durante la etapa de entrenamiento.

Hay varios tipos diferentes de validación cruzada. El más común es la validación cruzada *k-fold*, que divide el conjunto de datos en *k* subconjuntos. Luego, el modelo se entrena en *k-1* subconjuntos y se evalúa en el subconjunto restante. Esto se repite *k* veces, y los resultados se promedian para obtener una estimación del rendimiento del modelo.

- **La división de conjuntos de datos**: como indica su nombre implica dividir el conjunto de datos en un conjunto de entrenamiento [*trainset*] y un conjunto de prueba [*testset*], donde el primero se utiliza para entrenar el modelo y el segundo para evaluar su rendimiento una vez que se ha acabado de entrenar.

*Atención: Esta división en *trainset* y *testset* es de vital importancia. Los modelos deben ser entrenados con el primero y evaluados con el segundo. Entre ambos conjuntos no puede haber elementos comunes, esto es, deben ser conjuntos disjuntos.*

¿Por qué? Veamos algunas razones:

- **Generalización:** El objetivo principal de un modelo de aprendizaje automático es generalizar/aprender bien con datos nuevos que no han visto antes. Si se utilizan los mismos datos para el entrenamiento [*trainset*] y la prueba [*testset*], no es posible evaluar la capacidad del modelo para generalizar ante nuevas situaciones.
- **Sobreajuste [overfitting]:** Cuando un modelo se entrena en un conjunto de datos y también se evalúa en el mismo conjunto, hay un riesgo elevado de sobreajuste. El modelo puede "memorizar" los datos en lugar de "aprender" las características subyacentes, lo que significa que funcionará mal en nuevos datos [y no nos daremos cuenta de ello].
- **Subajuste [underfitting]:** Al igual que el sobreajuste, no se podrá detectar el subajuste si se usa el mismo conjunto para entrenamiento y prueba. El subajuste se produce cuando el modelo no aprende lo suficiente de los datos de entrenamiento y muestra un rendimiento deficiente incluso en esos datos.
- **Validación del modelo:** Seleccionar un modelo implica comparar su rendimiento con otros modelos o configuraciones. Si se evalúan múltiples modelos en el mismo conjunto de datos en el que fueron entrenados, la comparación será sesgada y posiblemente incorrecta.
- **Estimación de errores realistas:** Para entender cómo funcionará el modelo en un entorno de producción, se necesita una estimación precisa del error. Usar un conjunto de prueba separado que el modelo no haya visto durante el entrenamiento proporciona una medida más realista del rendimiento del modelo.
- **Confianza en el modelo:** Saber que un modelo ha sido probado en un conjunto de datos separado y ha demostrado un buen rendimiento aumenta la confianza en las predicciones del modelo para futuros datos no vistos.

14.5 Selección y optimización de hiperparámetros

Antes del entrenamiento de un modelo por medio de un conjunto de datos y un algoritmo, se han de tomar una serie de decisiones que afectarán al propio modelo y a su rendimiento. La primera decisión es la respuesta a ¿qué algoritmo usar? Lo normal es probar varios. Pero para cada uno de ellos deberemos determinar una serie de parámetros, muy dependientes del algoritmo, que influirán en el entrenamiento; no solo el precisión/sensibilidad o errores, sino también en tiempo de entrenamiento [minutos, horas o días].

*Los **hiperparámetros** son parámetros configurables que no se aprenden directamente del conjunto de datos y afectan el rendimiento y la capacidad de generalización y aprendizaje del modelo.*

Ejemplos de hiperparámetros incluyen la tasa de aprendizaje, el número de capas ocultas en una

red neuronal, el número de agrupaciones, etc. entre otros muchos. La elección adecuada de los hiperparámetros puede mejorar significativamente la calidad del modelo y el tiempo de ejecución del entrenamiento.

¿Qué valores deben tomar los hiperparámetros? Los algoritmos asignan unos valores para los hiperparámetros por defecto, los más habituales. Quien cree un modelo a partir de unos datos de entrenamiento y un algoritmo, podría prescindir de otorgar valores concretos a los hiperparámetros, dejando que el algoritmo use los que vienen de fábrica. Pero si queremos afinar el entrenamiento o mejorar los resultados, debemos tenerlos en cuenta.

Pero no estamos solos para llevar a cabo esta tarea; existen diversas técnicas para seleccionar y optimizar los hiperparámetros de los algoritmos. Algunas son:

- **Búsqueda en cuadrícula:** Consiste en definir un conjunto de valores posibles para cada hiperparámetro [un mínimo/máximo, una lista de posibles valores] y evaluar el rendimiento del modelo para cada combinación posible. Esta estrategia, de hecho, crea multitud de modelos, que entrena y evalúa. Al final entrega la combinación de hiperparámetros que optimiza una métrica indicada²²⁷.
- **Búsqueda aleatoria:** Implica seleccionar valores de hiperparámetros de forma aleatoria y evaluar el rendimiento del modelo para cada combinación. La búsqueda en cuadrícula puede generar muchos modelos y agotar el tiempo del que disponemos. En este caso esta estrategia confía en la suerte, llevando a cabo muestreos aleatorios de combinaciones de hiperparámetros. De esta manera podemos poner un límite al tiempo y los recursos empleados.
- **Optimización estocásticas:** Utiliza métodos basados en la teoría de probabilidad para encontrar la combinación óptima de hiperparámetros.

Ejemplo: Supongamos que estamos trabajando en un proyecto de detección de *spam* en correos electrónicos. Para seleccionar el algoritmo y modelo adecuados, consideraríamos la naturaleza del problema, que en este caso sería un problema de clasificación supervisada. Luego, evaluaremos el tamaño y la calidad de los datos disponibles, asegurándonos de que tengamos suficientes ejemplos de correos electrónicos etiquetados como *spam* y no *spam*. Además, consideraríamos la interpretabilidad y el rendimiento, ya que en este caso, es importante que el modelo pueda ser interpretado para comprender cómo se identifican los correos electrónicos de *spam*. Finalmente, utilizamos métricas de evaluación como precisión, sensibilidad y F1 para medir el rendimiento del modelo y optimizaremos los hiperparámetros, como el umbral de decisión para clasificar un correo electrónico como *spam* o no *spam*. Esto podría lograrse mediante técnicas como la búsqueda en cuadrícula.

²²⁷ Por ejemplo, podemos indicar que priorice la sensibilidad sobre la precisión, si los casos de falsos negativos son importantes. En el caso del dataset del cáncer de mama, un falso negativo es dar un diagnóstico consistente en que se tiene un cáncer benigno cuando realmente es maligno. Ya veremos esto.

RETOS DEL CAPITULO 14

1. Investiga y describe qué es el aprendizaje automático y cómo se utiliza en la vida cotidiana.
2. ¿Cuáles son las diferencias entre el aprendizaje supervisado y el no supervisado? Busca un tipo de aprendizaje denominado auto-supervisado.
3. Investiga y comparte ejemplos de aplicaciones del aprendizaje automático en la medicina. Busca en la *web* y da preferencia a artículos científicos [fíjate sólo en el título y en el *abstract* del mismo].
4. ¿Qué son los algoritmos de clasificación en el aprendizaje automático? Explícalo en términos sencillos.
5. Investiga y comparte ejemplos de cómo se utiliza el aprendizaje automático en la publicidad en línea. Busca en la *web* y da preferencia a artículos científicos [fíjate sólo en el título y en el *abstract* del mismo].
6. En grupo: reflexionar y describir el concepto de "*entrenamiento*" en el contexto del aprendizaje automático. ¿Qué es entrenar un modelo? ¿Por qué no se dice entrenar un algoritmo?
7. Pregúntale a tu IA favorita, qué es un algoritmos de recomendación y que se puede hacer con modelos creados para tal fin.
8. Investiga más en profundidad el por qué se llama regresión a la predicción de valores numéricos. Usa la *web*, no una inteligencia artificial.
9. Reflexiona: ¿Qué son los datos etiquetados y no etiquetados en el aprendizaje automático? ¿Por qué hacemos esa diferenciación? Explícalo en términos simples.
10. Reflexiona: ¿Por qué no nos interesa caer en el *overfitting* al entrenar un modelo? ¿Por qué aceptamos cierto grado de error, aunque no lo busquemos? ¿Podríamos conseguir algún día modelos 100% precisos?
11. ¿Por qué dividimos el conjunto de datos en dos, *trainset* y *testset*? ¿Qué relación tiene con el *overfitting*?
12. Dedúcelo tu mismo: ¿Qué es el *underfitting*?
13. Conversa con tu IA favorita: Quieres crear un modelo de predicción del tiempo, dialoga con ella para enumerar los pasos que necesitarías para conseguirlo.
14. En el capítulos hemos visto el código que entrena y evalúa un modelo SVM sobre un dataset que contiene datos sobre cáncer de mama. Crea un *notebook* que lleve a cabo este objetivo usando la IA incorporada en Colab o con otra IA externa.

Capítulo 15

LA CIENCIA DE LOS DATOS

En el capítulo anterior hemos descrito los conceptos básicos del aprendizaje automático, y dentro de ello hemos nombrado a la ciencia de los datos. Dedicaremos este capítulo a profundizar un poco en esta disciplina por su importancia desde un punto de vista profesional y por su relación con la disciplina denominada *Big Data*²²⁸.

La ciencia de datos es un campo interdisciplinario que ha evolucionado a lo largo del tiempo, incorporando elementos de estadísticas, informática, y análisis de negocios, entre otros. Aunque algunas de sus técnicas y métodos se remontan a siglos atrás en la historia de las matemáticas y las estadísticas, como campo con entidad propia empezó a tomar forma en las últimas décadas; y probablemente será una de las profesiones más importantes en las siguientes.

15.1 Introducción a la ciencia de datos y aprendizaje automático

La ciencia de datos es una disciplina que se centra en extraer información y conocimiento significativo a partir de grandes cantidades de datos. Como ya hemos dicho, utiliza una combinación de métodos estadísticos, matemáticos y de programación para analizar y comprender los conjuntos de datos, y utiliza estos conocimientos para tomar decisiones informadas y resolver problemas complejos.

En la década de 1960, se empezó a utilizar el término "minería de datos" (*data mining*) para describir los métodos estadísticos e informáticos que eran capaces de extraer información oculta de grandes conjuntos de datos. La explosión de la disponibilidad de datos, impulsada en parte por el auge de la Internet y la reducción en los costes de almacenamiento y procesamiento de datos, ha llevado al desarrollo y popularización de herramientas y técnicas más sofisticadas para el análisis de datos.

En 2001, William S. Cleveland introdujo el término *ciencia de datos* como un campo interdiscipli-

228 El término *Big Data* se utiliza comúnmente en muchos idiomas y contextos sin traducción, ya que se ha convertido en un término técnico universal. Sin embargo, si se desea traducir, una forma adecuada en castellano podría ser "Datos Masivos" o "Grandes Volumenes de Datos". Estas traducciones capturan la esencia de lo que significa *Big Data*: la gestión y análisis de conjuntos de datos tan grandes y complejos que requieren sistemas y métodos especializados para su manejo y análisis.

nario que engloba el análisis estadístico de datos, pero también se preocupa por la visualización, la ingeniería, y la toma de decisiones basada en datos. La publicación del artículo "*Data Scientist: The Sexiest Job of the 21st Century*"²²⁹ por D.J. Patil y Thomas H. Davenport en la revista *Harvard Business Review* en 2012 marcó un punto de inflexión, ya que popularizó tanto el término **ciencia de datos** como la función del **científico de datos**.

Si queremos encontrar información oculta [patrones] en los datos, necesitamos herramientas que aprendan automáticamente de los mismos y generalicen ¿Y qué herramientas conocemos que nos permita aprender y generalizar a partir de los datos? El aprendizaje automático.

Ejemplo: Aplicación de la ciencia de datos en la medicina

Imaginemos un conjunto de datos médicos que contiene información sobre pacientes, como edad, género, historial médico, resultados de pruebas y diagnósticos. Un científico de datos podría utilizar este conjunto de datos para identificar patrones que puedan ayudar a predecir enfermedades o evaluar la eficacia de ciertos tratamientos.

Por ejemplo, con estos datos de entrada y utilizando técnicas de aprendizaje automático, se podría desarrollar un modelo que tras analizar los datos de pacientes con diabetes, prediga la probabilidad de que uno en particular desarrolle complicaciones graves en el futuro; simplemente porque los patrones existentes en los datos históricos así lo indican.

El modelo podría tener en cuenta variables como el nivel de glucosa en sangre, el índice de masa corporal [IMC], la presión arterial y el historial familiar. Esto ayudaría a los médicos a tomar medidas preventivas y brindar un **tratamiento personalizado**.

Ejemplo: Aplicación de la ciencia de datos en el comercio electrónico

Imaginemos una empresa de comercio electrónico que recopila datos de transacciones de sus clientes, como información de productos comprados, tiempo de compra, ubicación geográfica y preferencias de compra. Utilizando técnicas de ciencia de datos, la empresa puede analizar estos datos con el objetivo de comprender mejor el comportamiento de sus clientes y mejorar su estrategia de marketing y ventas.

Siguiendo con el ejemplo, podrían desarrollar un modelo de recomendación personalizado utilizando algoritmos de aprendizaje automático. Este modelo analizaría el historial de compras de un cliente y los patrones de compra de otros clientes similares para hacer recomendaciones de productos relevantes. Así, la empresa podría ofrecer una experiencia de compra más personalizada y aumentar la satisfacción del cliente.

15.2 Big Data

Además de las herramientas relacionadas con el aprendizaje automático, la ciencia de datos necesi-

²²⁹ Artículo original: <https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>

ta poder manejar grandes conjuntos de información, probablemente enormes, para conseguir algo de provecho.

*Con **Big Data** nos referimos a las técnicas y herramientas para el procesamiento de grandes volúmenes de datos, tanto estructurados como no estructurados, que se genera en el mundo cada día y aquellos que hemos almacenado con el tiempo.*



Figura 67: Estimación de la cantidad real y prevista de datos generados en todo el mundo (en zettabytes).

Fuente: indicado en la imagen.

Pero no es la cantidad de datos lo que es importante; es lo que las organizaciones hacen con los datos lo que importa. *Big Data* es una herramienta usada para crear ideas que conduzcan a mejores decisiones y estrategias empresariales y gubernamentales. El concepto de *Big Data* no es absoluto y está en constante evolución, pero inicialmente se caracteriza por las tres V:

- **Volumen:** Se refiere a la cantidad de datos. *Big Data* normalmente implica grandes volúmenes de datos. Estos pueden ser cantidades desconocidas o datos que son demasiado grandes para ser manejados por sistemas de bases de datos tradicionales.
- **Velocidad:** Se refiere a la velocidad con la que se generan nuevos datos y la velocidad a la que estos se mueven. Las redes sociales pueden recibir cientos de miles de actualizaciones por minuto, lo que requiere un manejo en tiempo real.
- **Variedad:** Se refiere al tipo y naturaleza de los datos. En el pasado, manejábamos principalmente datos estructurados que se ajustaban perfectamente en tablas de bases de datos relacionales. Pero hoy en día, el 80% de los datos del mundo son no estructurados. Esta-

mos manejando diferentes tipos de datos, como texto, imágenes, sonidos, videos, registros y transacciones financieras, que están siendo generados y consumidos de forma continua.

El tratamiento eficiente de los datos requiere sistemas de almacenamiento y procesamiento que son significativamente más avanzados que las bases de datos tradicionales. También ha dado lugar al desarrollo de una nueva clase de tecnologías y arquitecturas de datos, algunas las nombraremos en breve.

Aunque el término empezó a popularizarse en la primera década de este siglo, podemos remontarnos a la creación de la ciencia de la computación [y la subsiguiente aparición de la ingeniería informática] en la segunda mitad del siglo pasado:

Años 60-70: Aunque aún no se usaba el término *Big Data*, ya se estaban recopilando grandes conjuntos de datos en sectores como la astronomía y la geología. Muchos de esos datos históricos son empleados hoy en día, aunque en su momento no se sabía el poder de procesamiento que tenemos actualmente.

Años 90: Con el auge de Internet, y la aparición de la *Word Wide Web*, las empresas comenzaron a acumular grandes cantidades de datos. Surgieron los primeros motores de búsqueda, y las empresas comenzaron a reconocer el valor de los datos para entender el comportamiento del cliente.

2001: Doug Laney, un analista de Meta Group, ahora Gartner, creó la definición de *Big Data* a partir de las tres V: Volumen, Velocidad y Variedad, que desde entonces han sido utilizados para describir sus características fundamentales.

2004: Google publicó un artículo sobre el sistema de archivos distribuido GFS, que permitía el procesamiento de grandes cantidades de datos. Este artículo fue la base para el desarrollo de *Hadoop* por parte de Yahoo y actualmente en la Apache Software Foundation.

2005: Lanzamiento de *Hadoop*. Este *framework* de código abierto permitió el almacenamiento y el procesamiento de conjuntos de datos muy grandes, y se convirtió en una tecnología clave para el *Big Data*.

2006-2009: Con la popularización de las redes sociales y el auge de dispositivos móviles, la generación de datos se aceleró de forma exponencial, lo que llevó a un mayor enfoque en el almacenamiento y el análisis del *Big Data*.

2010: Se popularizan las bases de datos NoSQL, diseñadas para manejar tipos de datos no estructurados, lo que facilita aún más el tratamiento *Big Data*.

2012: Como ya hemos dicho, la *Harvard Business Review* publicó el artículo "*Data Scientist: The Sexiest Job of the 21st Century*", lo que impulsó aún más el interés por *Big Data* y la ciencia de datos.

2015 en adelante: El aprendizaje automático y la inteligencia artificial empiezan a jugar un papel mucho más importante en el análisis y la interpretación de *Big Data*. Los conceptos de "cuatro V" y

"cinco V" [agregando Veracidad y Valor respectivamente] comienzan a ser discutidos más ampliamente.

En la actualidad, *Big Data* es una parte integral de diversas industrias como la atención médica, la logística, el marketing y más. Los avances en la computación en la nube y en tecnologías como el Internet de las Cosas (IoT) continúan ampliando sus fronteras.

15.3 El ciclo de vida de la ciencia de los datos

El ciclo de vida de la ciencia de datos se compone esencialmente de la recopilación de datos y su selección, la limpieza de datos, el análisis exploratorio de datos, la construcción de modelos y el despliegue de los modelos creados.

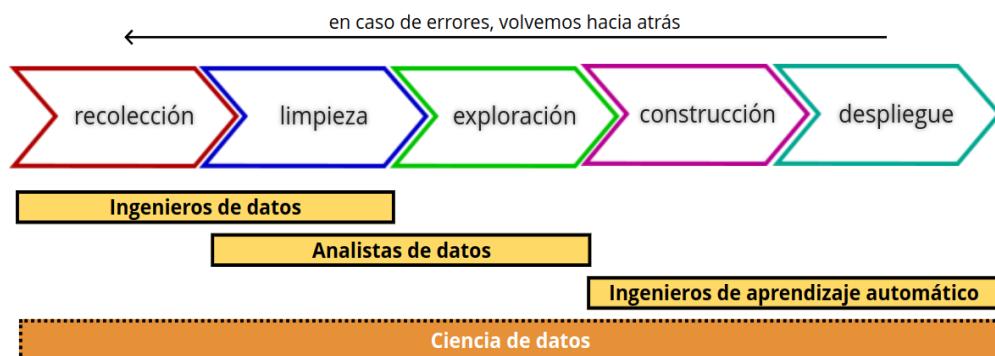


Figura 68: Fases y roles implicados en la ciencia de los datos.

Recolección y selección de datos: Los datos pueden provenir de diversas fuentes, como bases de datos, archivos CSV, archivos de texto, APIs, sensores, redes sociales, entre otros muchos. Por ejemplo, para analizar el comportamiento de los usuarios en un sitio web, se pueden recopilar datos de registro de servidores web y datos de seguimiento de clics.

Pero también los datos pueden tener diferentes formatos: Estos pueden poseer un formato estructurado, por ejemplo organizados en tablas con filas y columnas, como una hoja de cálculo [ver la Figura 61]. Pero, por otro lado, tenemos datos no estructurados, que no siguen un formato específico, como por ejemplo correos electrónicos, imágenes, videos o texto libre. Por ejemplo, los tweets de Twitter [X] son datos no estructurados.

Limpieza y transformación de datos: En esta etapa, se eliminan datos duplicados, se resuelven valores faltantes o inconsistentes, y se transforman los datos al formato adecuado. Por ejemplo, si se tienen datos de temperatura en grados Fahrenheit y se requiere en grados Celsius, se realiza una transformación para convertirlos.

Hay también otras transformaciones más complejas, por ejemplo convertir el texto a su correspondiente vector de características (*embeddings*), tratar con datos categóricos, no numéricos, etc.

Exploración y visualización de datos: El Análisis descriptivo consiste en calcular estadísticas básicas, como promedio, mediana, desviación estándar, y generar visualizaciones para entender la distribución de los datos. Por ejemplo, se puede calcular la edad promedio de los estudiantes en un conjunto de datos y visualizarlo en un histograma. Se trata de comprender los datos. Para ello también se lleva a cabo una identificación de patrones y relaciones, mediante técnicas de visualización, como gráficos de dispersión y mapas de calor, se pueden identificar patrones y relaciones entre variables. Por ejemplo, se puede visualizar la relación entre el salario y la experiencia laboral en un conjunto de datos para determinar si existe una correlación.

También es importante el análisis de *outliers*, que son valores atípicos que pueden distorsionar el análisis. Por ejemplo, si se analizan datos de ventas y se encuentra una venta extremadamente alta, puede ser necesario investigar si fue un error o un evento extraordinario.

Finalmente se **construye y despliega el modelo**. Para ello se comienza por la selección de características, el cual es un proceso crucial en la construcción de modelos de aprendizaje automático. Consiste en identificar las variables o características más relevantes y significativas para el problema en cuestión. Algunas técnicas comunes utilizadas en la selección de características incluyen la correlación, la eliminación de características redundantes y el análisis de importancia de características.

Una vez que se han seleccionado las características, es necesario entrenar el modelo de aprendizaje automático. Esto implica proporcionar algoritmos de aprendizaje automático con un conjunto de datos de entrenamiento, que consta de ejemplos etiquetados, para que pueda aprender patrones y relaciones entre las características y las etiquetas.

Una vez que el modelo ha sido entrenado, es esencial evaluar su rendimiento para comprender qué tan bien ha aprendido [Figura 69] y su capacidad de generalizar ante la presencia de nuevos datos. Esto se logra utilizando métricas de rendimiento que miden la precisión, la exhaustividad, la exactitud o cualquier otro criterio relevante para el problema en cuestión. Además, como hemos tratado antes, es común dividir el conjunto de datos en conjuntos de entrenamiento y prueba, para evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento.

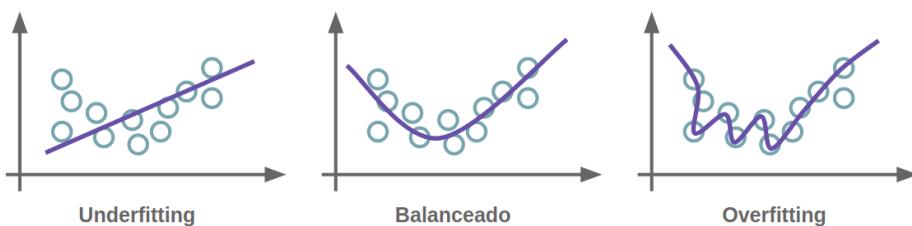


Figura 69: Tipos de aprendizajes según el resultado: subentrenamiento, balanceado y sobreentrenamiento.

Fuente: Wikimedia Commons

El modelo, después de una evaluación sistemática y rigurosa, se implementa en la estructura y canal preferidos de la organización. Este es el último paso en el ciclo de vida de la ciencia de datos. Cada paso de este ciclo debe realizarse con cuidado. Si algún paso se realiza incorrectamente y, por lo tanto afecta el siguiente paso, todo el esfuerzo realizado se desperdicia y debemos volver a una etapa anterior; incluso volver a empezar. Desde la necesidad inicial de la organización, hasta la implementación del modelo, a cada paso, se debe prestar la atención, el tiempo y el esfuerzo adecuados.

15.4 Aplicaciones de la ciencia de datos

La ciencia de datos, y sus técnicas y tecnologías asociadas, tienen una amplia gama de aplicaciones en diversos campos, lo que ha llevado a importantes avances en medicina, finanzas, marketing, transporte, entretenimiento y muchos otros sectores. Veamos algunos ejemplos de aplicaciones concretas para ilustrar cómo estas disciplinas están transformando diferentes áreas de la sociedad.

15.4.1 Medicina

Diagnóstico médico: Los modelos de aprendizaje automático pueden analizar datos clínicos y de pacientes para ayudar en la detección temprana y el diagnóstico preciso de enfermedades. Por ejemplo, se han desarrollado algoritmos capaces de detectar enfermedades cardíacas, cáncer de piel y retinopatía diabética mediante el análisis de imágenes médicas.

Medicina personalizada: La ciencia de datos puede ayudar a crear terapias personalizadas para pacientes. Los modelos pueden analizar el historial médico, los factores genéticos y otros datos para identificar tratamientos óptimos y predecir la eficacia de ciertos medicamentos.

15.4.2 Finanzas

Detección de fraudes: Los algoritmos de aprendizaje automático pueden analizar grandes volúmenes de datos financieros para identificar patrones y anomalías que puedan indicar actividades fraudulentas. Estos modelos pueden ayudar a las instituciones financieras a prevenir el fraude en tarjetas de crédito, seguros y transacciones bancarias.

Predicción del mercado: La ciencia de datos se utiliza para analizar datos históricos y en tiempo real del mercado financiero, lo que permite predecir las tendencias y realizar inversiones más informadas. Los modelos de aprendizaje automático pueden ayudar a los inversores a tomar decisiones basadas en patrones y señales del mercado.

15.4.3 Marketing

Segmentación de clientes: Los algoritmos de aprendizaje automático pueden analizar los datos de los clientes, como sus preferencias, comportamiento de compra y datos demográficos, para segmentarlos en grupos con características similares. Esto permite a las empresas personalizar sus es-

trategias de marketing y ofrecer productos y servicios específicos a cada segmento.

Recomendación de productos: Los sistemas de recomendación utilizan algoritmos de aprendizaje automático para analizar el historial de compras, las preferencias y el comportamiento de los clientes con el fin de ofrecer recomendaciones personalizadas de productos o servicios. Ejemplos populares de esto son las recomendaciones de películas o música en plataformas de streaming como Netflix y Spotify.

15.4.4 Transporte

Conducción autónoma: La ciencia de datos y el aprendizaje automático son fundamentales para desarrollar vehículos autónomos. Los algoritmos de visión por computadora y el procesamiento de sensores permiten que los automóviles interpretan su entorno y tomen decisiones en tiempo real, lo que mejora la seguridad y la eficiencia en el transporte.

Optimización de rutas: Los modelos de aprendizaje automático pueden analizar datos de tráfico, patrones de movilidad y otras variables para optimizar las rutas de transporte y reducir la congestión. Esto puede mejorar la eficiencia del transporte público y reducir los tiempos de viaje.

15.4.5 Entretenimiento

Recomendación de contenido: Las plataformas de entretenimiento utilizan algoritmos de aprendizaje automático para recomendar contenido a los usuarios. Por ejemplo, servicios de streaming como Netflix utilizan modelos que analizan el historial de visualización, las calificaciones y las preferencias de los usuarios para ofrecer recomendaciones personalizadas de películas y programas de televisión.

Generación de música y arte: Los modelos de aprendizaje automático pueden ser entrenados en grandes conjuntos de datos musicales o artísticos para generar composiciones musicales originales o incluso obras de arte. Estas aplicaciones creativas muestran cómo la ciencia de datos puede expandirse más allá de los ámbitos tradicionales.

15.5 Tendencias y futuro de la ciencia de datos y aprendizaje automático

En el campo de la ciencia de datos y el aprendizaje automático, existen varias tendencias y avances que están dando forma al futuro de estas disciplinas. Vamos a destacar las dos más importantes a medio plazo que inciden en la ciencia de los datos:

Aprendizaje automático en tiempo real: A medida que los avances en capacidad de cómputo y algoritmos permiten un procesamiento más eficaz y eficiente, se están desarrollando aplicaciones de aprendizaje automático en tiempo real. Esto implica tomar decisiones y realizar predicciones mucho más rápido y cuando se necesiten, lo cual es fundamental en áreas como el comercio financiero, la detección de fraudes y la detección de intrusiones en sistemas de seguridad. Por ejemplo, las em-

presas de comercio electrónico utilizan algoritmos de aprendizaje automático en tiempo real para personalizar las recomendaciones de productos en función del comportamiento de compra del usuario en tiempo real.

Automatización de la ciencia de datos: Se trata del desarrollo de herramientas y algoritmos que pueden realizar tareas de análisis de datos de manera autónoma, sin necesidad de una intervención humana constante. Esto permite a los científicos de datos centrarse en tareas más complejas y estratégicas. Un ejemplo de esto es la automatización de la selección y ajuste de modelos, donde los algoritmos pueden explorar diferentes opciones de modelos y seleccionar automáticamente el mejor modelo para un conjunto de datos dado.

RETOS DEL CAPITULO 16

1. Dialoga con tu IA favorita: Investiga y describe en tus propias palabras qué es la ciencia de datos y cómo se relaciona con el aprendizaje automático.
2. Pregúntale a tu agente conversacional favorito, qué son los términos o conceptos:
 - Big Data.
 - Análisis de datos.
 - Minería de datos.
 - Análisis exploratorio de datos [EDA].
 - Visualización de datos.
 - Feature Engineering.
 - Hadoop.
 - SQL y no-SQL
 - Gestión de datos.
 - Data Lake.

No te quedes con una simple definición; busca entender y relacionar todos ellos.

3. Una técnica muy habitual es el *web-scraping*. Busca información de qué es y pregunta a tu IA favorita que te describa cómo hacerlo. ¿Te atreves a pedirle que genere un código para que lo haga en una página de un periódico?
4. ¿Qué es una fuente de datos no estructurados? Dialoga con tu IA favorita para que te muestre cómo se tratan estos datos en el ámbito del aprendizaje automático.
5. Busca en la web ejemplos de empresas que usen la Ciencia de los Datos en su día a día.
6. ¿Qué es un/una ingeniero/a de datos? Indaga qué objetivos tiene.
7. ¿Qué es un/una analista de datos? Enumera qué objetivos tiene.
8. ¿Qué es un/una ingeniero/a de aprendizaje automático? Aprende qué objetivos tiene.
9. En la *web* hay cantidad de cursos *online*. Busca los que se correspondan con la Ciencia de los Datos.
10. Busca e indaga qué carreras universitarias o ciclos formativos debes estudiar para poder trabajar en el ámbito de la Ciencia de los Datos.

Capítulo 16

ENTRENAMIENTO Y EVALUACIÓN DE MODELOS

En este capítulo vamos a entrenar y evaluar modelos a partir de conjuntos de datos y uno o más algoritmos de aprendizaje automático. No entraremos en redes neuronales artificiales, contenido que veremos en el siguiente capítulo, si no que nos centraremos en unos pocos algoritmos de base analítica o estocástica.

Antes de empezar es importante recordar que algoritmos de aprendizaje automático para crear modelos hay muchos; y cada uno de ellos posee su propia estrategia para aprender a partir de los datos. Sólo tienen una cosa en común, las matemáticas.

El resto del capítulo está dividido según el algoritmo usado, empezando por uno sencillo (regresión lineal) y aumentando la complejidad. Para cada modelo creado se hará una evaluación y allí en donde sea posible se compararán los modelos.

16.1 Regresión lineal

YearsExperience	Salary
1.2	39344.0
1.4	46206.0
1.6	37732.0
2.1	43526.0
2.3	39892.0

La regresión lineal es el enfoque de aprendizaje automático que tiene como fin modelar la relación entre una variable dependiente [etiqueta] y una o más variables independientes [características]. El caso más elemental es la regresión lineal simple, donde se tiene una sola variable independiente.

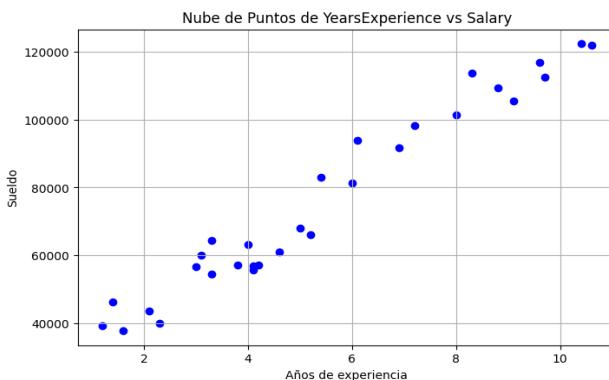
Para empezar, vamos a trabajar en este ejemplo con un *dataset* muy simple, y pequeño [ver a la izquierda]; uno con 30 muestras, cada una de ellas con 2 columnas: *YearsExperience* y *Salary*. Este conjunto de datos relaciona los años de experiencia del trabajador con su salario. Bien, ya tenemos los datos. Vamos a **formular el problema**: queremos crear un modelo que, a partir de la experiencia en años de una persona en un puesto de trabajo, prediga su salario.

Evidentemente vamos a cometer un error. Sólo con los años de experiencia no podemos adivinar el salario de la persona, pero podemos estimarla respecto a lo que los datos nos enseñan. Y ahí está el aprendizaje.

Los datos nos dan una muestra de la realidad y nosotros vamos a crear un modelo muy simple que va a generalizar esa realidad. Al aplicar al modelo nuevos datos [años de experiencia], obtendremos el salario que debería tener la persona, según los datos originales.

En el caso de la **regresión lineal simple**, lo que pretendemos es crear un modelo con esta forma: $Ax+B=y$

Este modelo tiene dos incógnitas [x e y, respectivamente años y salario], y dos parámetros, A y B. Una imagen puede ayudar a esclarecer lo que acabamos de decir.



La imagen anterior muestra la gráfica de nuestro *dataset*, aquel que relaciona años de experiencia y sueldo. Nuestras vivencias nos dicen que cuantos más años pase una persona en una empresa y más experiencia acumule, más elevado será su sueldo.

Esta gráfica está formada por 30 puntos [pares años/sueldo]; y representados en forma de nube nos da una idea basada en datos de ese hecho. En el eje de las x [abscisas] tenemos los años de experiencia; fíjate en el valor 6 años de experiencia, sube por el eje y [ordenadas] y obtendrás lo que los datos dicen del salario de una persona a los 6 años de experiencia: poco más de 80.000. Pero ¿qué salario puede esperar una persona con 10 años de experiencia?

Precisamente por eso necesitamos una recta $Ax+B=y$ donde al introducir un año [x] nos devuelve el salario [y]. Sólo necesitamos calcular los términos A y B; y hacerlo de tal forma que sean los mejores términos posibles. ¿Y cuales son los mejores términos posibles? Los que minimicen el error que cometes al usar la línea como modelo.

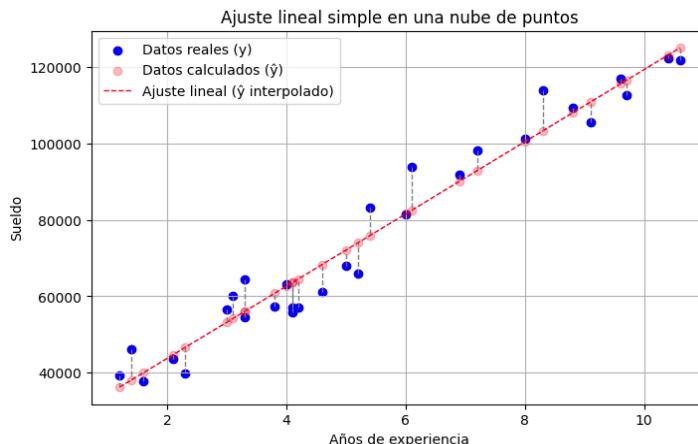
Veamos un código que los calcula. Partiendo de un *dataset* ya cargado [en el código llamado *data*] calculo A y B usando una función de la librería *numpy* [con *np* como alias]. Esta función haya una pareja de A y B tal que minimice la media del cuadrado de los errores:

```

1 import numpy as np
2
3 # Realizar la regresión lineal
4 X = data['YearsExperience']
5 y = data['Salary']
6
7 # Calculo los términos de la recta
8 A, B = np.polyfit(X, y, 1)
9
10 y = A * X + B # y son los datos predichos

```

Después de la capacidad de crear modelos, lo más importante es poder evaluarlos. Poder conocer cuanto²³⁰ fallamos si tomamos esos modelos como referencia.



Representemos lo que acabamos de decir. La imagen de la izquierda es la misma que la anterior, pero ahora muestra la línea calculada y los errores que cometemos al tomar este modelo como referencia. Los errores son las líneas verticales que van desde el punto [del dataset] hasta su proyección vertical en la línea [el modelo].

Imagina que sumamos todos esos errores [líneas verticales] y hayamos su media. Realmente tenemos que sumar su valor absoluto, ya que habrá errores positivos y negativos. Esta métrica recibe el nombre de *Median Absolute Error*, más conocida como MAE:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Pero también podemos usar la métrica MSE [*Median Squared Error*]:

²³⁰ Observa el gráfico, los puntos y la recta, pero sobre todo imagina los infinitos puntos que no están; ¿qué patrón seguirían esos puntos? Muy probablemente el mismo que los puntos que sí están. Si conseguimos un dataset que represente los datos que no están (y que deseamos predecir) podemos crear un modelo predictivo muy bueno. Al menos tan bueno como la representatividad que tiene el dataset que usemos.

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y - \hat{y})^2$$

En nuestro *dataset*, el MAE es de 4644 y el RMSE²³¹ 5592. No está mal. Aunque hemos hecho trampas. No hemos separado el conjunto de datos en *trainset* y *testset*, como indicamos en §14.4 que debíamos hacer. Ya verás porqué.

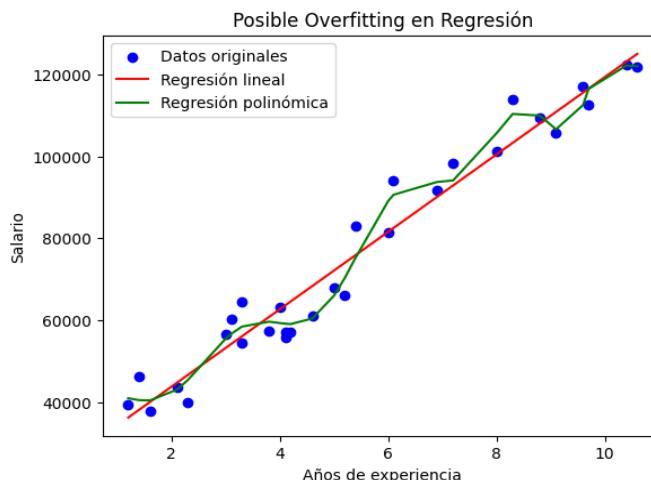
La siguiente figura muestra de nuevo la nube de puntos y la línea (roja) del modelo de regresión lineal. Pero podemos ver un nuevo elemento, la curva que representa a un modelo lineal polinómico. Este modelo intenta buscar una curva que se aproxime más a los puntos mediante la fórmula:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \dots + \beta_n x^n$$

En donde el aprendizaje consiste en estimar los términos β del polinomio de grado n . En nuestro caso hemos empleado $n=15$ para calcular el polinomio, con el siguiente código:

```
1 poly = PolynomialFeatures(degree=15)
```

Un modelo de regresión polinómica.



Volviendo a la figura, **¿qué modelo se ajusta mejor a los datos del dataset?** El modelo polinómico. Pero **¿qué modelo predice mejor a partir de los datos del dataset?**

“Ajustarse mejor” y “predecir mejor”, son conceptos diferentes, aunque relacionados, y como lo que queremos es predecir, nuestro objetivo es crear modelos que predigan mejor a partir de los datos. Esto es, modelos que ajusten mejor, no a los datos del *dataset*, sino a los datos que aún no tenemos. Por eso la respuesta a ¿qué modelo predice mejor? Es una pregunta trampa, pero que de-

231 Raíz cuadrada del MSE.

bemos afrontar. De hecho ya sabemos cómo hacerlo.

Hoy en día es difícil viajar al futuro para traer nuevos datos que, para nosotros en este momento, aún no tenemos y que serían muy útiles para probar un modelo que hemos creado. Pero podemos “simular” el resultado de este viaje temporal, ¿cómo? dividiendo el *dataset* en dos partes, el *trainset* y el *testset*; para, a continuación, entrenar el modelo con el *trainset* y probar que tal funciona con el *testset*. En nuestro caso tenemos 30 muestras, podríamos tomar 24 como *trainset* [80%] y 6 como *testset* [20%]. Al hacerlo tenemos la certeza de que hemos probado el modelo con datos que no han sido usados durante el entrenamiento y, si los resultados de la evaluación son buenos y con la información disponible hasta el momento, podemos afirmar que es un buen modelo. Cuando un modelo se ajuste muy bien a los datos de entrenamiento, pero prediga mal, decimos que este modelo está haciendo **overfitting** [sobreentrenando]. Volveremos a esto más tarde.

16.2 Regresión logística

La regresión logística es un tipo de análisis de regresión que se utiliza para predecir el resultado de una variable categórica en función de las variables independientes. Es la versión del modelo de regresión que acabamos de ver, pero para predecir un conjunto de valores limitados llamados clases. Es, por tanto, una clasificación.

Vamos a retomar el *dataset* que vimos en el apartado 14.3.1 [Máquinas de Soporte Vectorial], aquel que nos entregaba 569 muestras provenientes de biopsias, cada una de ellas con 30 variables, para determinar si un cáncer era maligno o no. Queremos crear el modelo de regresión logística que prediga 1 [benigno, clase positiva] o 0 [maligno, clase negativa] a partir de 30 características extraídas de personas reales. Es un problema de clasificación binaria con dos clases: 0 [clase negativa] y 1 [clase positiva].

```
1 from sklearn.datasets import load_breast_cancer
2 from sklearn.model_selection import train_test_split
3 from sklearn.linear_model import LogisticRegression
4 from sklearn.metrics import accuracy_score, confusion_matrix, precision_score
5 from sklearn.metrics import recall_score, f1_score, roc_auc_score, roc_curve
6 import matplotlib.pyplot as plt
7
8 # Cargar el conjunto de datos
9 data = load_breast_cancer()
10 X, y = data.data, data.target
11
12 # Dividir el conjunto de datos en entrenamiento y prueba
13 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
14
15 # Crear el modelo de regresión logística
16 model = LogisticRegression(max_iter=10)
17
18 # Entrenar el modelo
19 model.fit(X_train, y_train)
20
21 # Realizar predicciones
22 y_pred = model.predict(X_test)
23
24 # Evaluar el modelo
25 print(f'Accuracy: {accuracy_score(y_test, y_pred)}')
```

```

26 print(f'Precision: {precision_score(y_test, y_pred)}')
27 print(f'Recall: {recall_score(y_test, y_pred)}')
28 print(f'F1 Score: {f1_score(y_test, y_pred)}')

```

Esta vez sí hemos dividido el *dataset* original en un *trainset* [*X_train*, *y_train*] y un *testset* [*X_test*, *y_test*], en donde el segundo contiene²³² un 20% de las muestras originales. Por tanto vamos a entrenar un modelo con el 80% de las muestras y evaluarlo con el 20% restante.

El entrenamiento [*model.fit*] se lleva a cabo en la línea 19. El resultado es un modelo [*model*] que puede hacer predicciones [*predict*]. Y si podemos hacer predicciones, podemos evaluar el modelo usando el *testset*.

Pero antes recordemos un detalle, en este *dataset* la etiqueta 1 significa “maligno” y la etiqueta 0 “benigno”. Es importante.

Empecemos por definir algunos conceptos, típicos de un clasificador binario:

- *True positive* [TP]: Las veces en las que el modelo afirma que es maligno [1] y en la realidad lo es.
- *True negative* [TN]: Las veces en las que el modelo afirma que es benigno [0] y realmente lo es.
- *False positive* [FP]: En donde el modelo predice maligno [1], pero realmente era benigno [0].
- *False negative* [FN]: En donde el modelo predice benigno [0], pero realmente se trataba de un cáncer maligno [1].

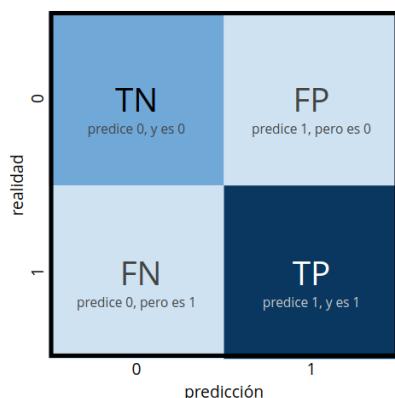


Figura 70: Esquema general de matriz de confusión para un clasificador binario

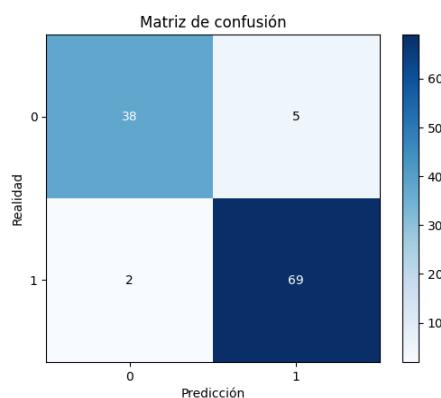


Figura 71: Matriz de confusión, dataset: *cancer_breast*, algoritmo: regresión logística

Para un clasificador binario, estos cuatro conceptos puede representarse en la **matriz de confusión**. En la figura 70 podemos ver el esquema general de una matriz de confusión con los términos que

²³² La división (split) se basa en un muestreo aleatorio. Tal y como está el código en este página, varias ejecuciones del mismo arrojará divisiones diferentes.

acabamos de definir para un clasificador binario con etiquetas 0 y 1. En la figura 71 vemos la matriz de confusión generada usando la regresión logística sobre el testset.

Podemos observar que nuestro modelo se ha “confundido” 2 veces [FN] prediciendo benigno [0] cuando realmente fue maligno [1]; y se ha equivocado 5 veces [FP] prediciendo maligno [1] cuando realmente era benigno [0].

¿Qué prefieres, ser diagnosticado como maligno y que finalmente no lo sea o que se te diagnostique un cáncer benigno cuando realmente es maligno? Los errores [las confusiones] en un clasificador binario pueden ser malas o menos malas, dependiendo del problema.

A partir de estos conceptos podemos sintetizar métricas que nos permitan evaluar el modelo y compararlo:

1. **Exactitud** [accuracy²³³]: Se calcula dividiendo el número de predicciones correctas [TP+TN] entre el total de predicciones [TP+TN+FP+FN]. Esta métrica nos da una idea general de que tan bien predice nuestro modelo, pero no hace diferencia entre predicciones malas o menos malas.

Desde el punto de vista positivo [TP]:

2. **Precisión** [precision]: Se centra en la calidad de las predicciones que el modelo hace para los tumores malignos [1]. Se calcula como $TP/[TP+FP]$. Es la proporción de la predicción correcta de “maligno” [1] respecto al total de predicciones “maligno” [1].
3. **Sensibilidad** [recall]: Se centra en la capacidad del modelo para identificar todos los tumores malignos reales [1] en el conjunto de datos. Se calcula como $TP/[TP+FN]$. Es la proporción de la predicción correcta de “maligno” [1] respecto al total de diagnósticos reales como “maligno” [1].
4. **F1**: Es una métrica que combina tanto la precisión como el *recall* en un solo número. Se utiliza especialmente en situaciones donde ninguna de estas las dos métricas es suficiente por sí sola para evaluar el rendimiento del modelo de forma efectiva. El F1 se define como la media armónica²³⁴ de la precisión y el *recall*.

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

El F1 varía entre 0 y 1, siendo 1 el mejor valor posible y 0 el peor. Un F1 más alto indica un mejor equilibrio entre la precisión y el *recall*.

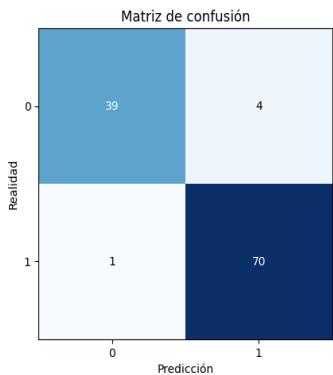
Y desde el punto de vista negativo [TN]:

233 A veces se traduce como “precisión”, llevándonos a la confusión.

234 Se utiliza la media armónica en lugar de la media aritmética porque da un valor más bajo cuando cualquiera de los términos (precisión o sensibilidad) es bajo. Esto es útil en situaciones donde se quiere que ambas métricas sean altas; un F1 alto solo se logra si ambas métricas son buenas.

5. **Especificidad:** Equivalente a la precisión, pero con la clase negativa. Se centra en la calidad de las predicciones que el modelo hace para los tumores benignos [0]. Se calcula como $TN/(TN+FP)$.
6. **Valor predictivo negativo** [NPV, por sus siglas en inglés]: Equivalente a *recall*, pero con la clase negativa. Se centra en la capacidad del modelo para identificar todos los tumores benignos reales [0] en el conjunto de datos. Se calcula como $TN/(TN+FN)$.
7. **F1 negativo:** Es la métrica, equivalente al F1, que combina tanto la especificidad como el valor predictivo negativo en un solo valor.

Ahora podemos **comparar modelos**.



A la izquierda podemos ver la matriz de confusión que obtuvimos a partir de un modelo creado con el algoritmo SVM [ver el apartado 14.3.1]. A primera vista podemos ver que FP [*false positive*] es 4 respecto a valor de 5 que obtuvimos con el modelo de regresión. Igualmente los *false negative* [FN] fueron 1 frente a 2. Esto nos indica que los valores de exactitud, precisión y *recall* van a ser mejores en el modelo SVM que en el modelo basado en regresión logística.

Calculando las métricas para los dos modelos, obtenemos la siguiente tabla:

	SVM	Regresión logística
exactitud	0.9561	0.9386
precisión	0.9459	0.9394
sensibilidad	0.9859	0.9718
F1	0.9655	0.9517
especificidad	0.9070	0.8837
NPV	0.9750	0.9500
F1 negativo	0.9398	0.9157

Esto nos indica que el modelo SVM es ligeramente superior al de regresión logística usando los hiperparámetros indicados²³⁵ al crear el modelo. Como en el contexto de nuestro problema [diagnóstico de biopsias de cáncer de mama] hay diagnósticos malos y menos malos, correspondientes a las clases 1 y 0 respectivamente, la métrica que nos permite comparar si cuando el modelo se confunde lo hace a favor de la paciente, en este caso es la especificidad y el NPV.

²³⁵ Para la regresión logística indicamos el parámetro `max_iter=10` (muy pocos). Si hubiésemos indicado 10000, este modelo igualaría al creado con el algoritmo SVM. Pero este último también tiene sus hiperparámetros, los cuales, al escoger los adecuados, mejoraríaían sus predicciones.

16.3 Árboles de decisión

Vamos a continuar con el mismo *dataset*, pero con diferentes algoritmos. Seguimos dentro del aprendizaje supervisado aplicado a tareas de clasificación.

Los **árboles de decisión** son un algoritmo de aprendizaje automático utilizada tanto en tareas de clasificación como de regresión. Estos algoritmos se basan en la idea de construir un modelo en forma de un árbol, donde cada nodo interno representa una pregunta o prueba sobre una característica de los datos, y cada rama representa una posible respuesta a esa pregunta. Al final del árbol, encontramos las hojas que contienen las predicciones o las clases finales [ver el apartado 14.3.1].

El proceso de construcción de un árbol de decisión implica dividir el conjunto de datos de entrenamiento en subconjuntos más pequeños y más homogéneos con respecto a la variable objetivo. Esto se hace de manera recursiva hasta que se cumple algún criterio de parada, como, por ejemplo, la profundidad máxima del árbol.

Una vez que se ha construido el árbol de decisión, se puede utilizar para hacer predicciones sobre nuevos ejemplos. Para ello, se sigue el camino a través del árbol respondiendo las preguntas en cada nodo interno hasta llegar a una hoja, donde se encuentra la predicción correspondiente

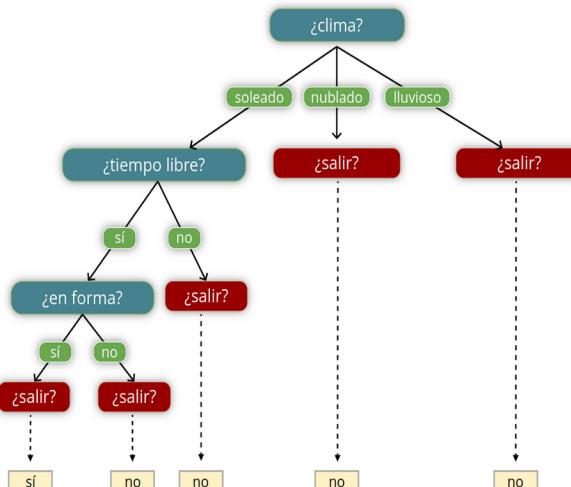


Figura 72: Esquema de un árbol de decisión

Fuente: ChatGPT (concepto)

Imaginemos que queremos predecir si una fruta es una manzana o una naranja, en función de su color. Para ello, el árbol de decisión podría tener un nodo raíz que examina el color de la fruta, y de este saldrían dos ramas: una para la fruta "roja" y otra para la fruta "naranja". Si seguimos por la rama de la fruta "roja", podríamos encontrar una hoja que clasifica la fruta como una "manzana".

De esta manera, el árbol ayuda a tomar decisiones basadas en las características de los datos.

La Figura 72 muestra un árbol de decisión que nos puede ayudar a tomar la decisión de “salir a correr o no”²³⁶, con tres variables independientes [qué clima hace, tengo tiempo libre y estoy en forma] en forma de preguntas. La variable dependiente [la que queremos predecir] es “salir”, igualmente en forma de pregunta. Los valores que puede tener esta columna son “si” y “no”. Este árbol nos permite hacernos tres preguntas y siguiendo sus ramas llegar a una respuesta final, salgo o no salgo.

16.4 Bosques aleatorios

Los **bosques aleatorios** (*random forest*) son un algoritmo de aprendizaje automático basado en ensambles²³⁷ de árboles de decisión. Combina múltiples árboles de decisión entrenados en diferentes subconjuntos de datos y características aleatorias para mejorar la precisión y reducir el sobreajuste.

Random forest utiliza un enfoque de muestreo aleatorio conocido como **muestreo con reemplazo** (*bootstrapping*), para crear diferentes conjuntos de datos de entrenamiento. Estos se utilizarán para entrenar árboles de decisión individuales.

El muestreo con reemplazo es una técnica estadística en la que se seleccionan elementos de una población o conjunto de datos de manera aleatoria y se devuelven al conjunto original antes de realizar una nueva selección. Esto significa que el mismo elemento puede ser elegido más de una vez durante el proceso.

Como acabamos de decir, para cada conjunto de datos de entrenamiento generado se construye un árbol de decisión independiente. Una vez que todos los árboles de decisión han sido construidos, se utiliza el proceso de votación [clasificación²³⁸] o promedio [regresión²³⁹] para obtener la predicción final. Esta técnica se denomina *bagging* [acrónimo de *bootstrap aggregating*].

Mi IA favorita dice: *Bagging es una técnica de ensamblado en aprendizaje automático que se utiliza para mejorar la estabilidad y la precisión de algoritmos de aprendizaje estadístico. Bagging se utiliza principalmente para reducir la variabilidad y evitar el sobreajuste.*

²³⁶ Como indica la fuente, el concepto fue de ChatGPT, al igual que sus respuestas, que son poco saludables. Si te gusta salir a correr, sal siempre que quieras, tengas tiempo libre y un estado de forma aceptable como mínimo.

²³⁷ El diccionario de la Real Academia Española (RAE) incluye el término *ensamblaje* como sinónimo de *ensambladura*. Ambos conceptos se refieren al proceso y la consecuencia de ensamblar, un verbo que alude a ajustar, coordinar o acoplar algo.

²³⁸ En el caso de la clasificación, cada árbol emite una votación por la clase predicha, y la clase que recibe la mayoría de votos se considera la predicción final.

²³⁹ Para la regresión, se promedian las predicciones de todos los árboles para obtener el valor final.

16.5 XGBoost

En el algoritmo anterior aprendimos lo que es hacer *bagging* con un algoritmo más simple [por ejemplo árboles de decisión], usando un conjunto de estos [ensamble] para sintetizar un resultado partiendo de sus predicciones. Ese es el funcionamiento de *random forest*. Usando un símil, como haría un jurado, promediando o usando la moda²⁴⁰ a partir de sus opiniones individuales.

Pero esta no es la única estrategia que tenemos para potenciar algoritmos más simples y débiles.

El *boosting* es una técnica de ensamblado que tiene como objetivo mejorar la precisión de un modelo de predicción con menos capacidad. A diferencia del *bagging*, que entrena múltiples modelos de forma independiente y luego los combina, el *boosting* entrena múltiples modelos de forma secuencial, donde cada modelo intenta corregir los errores de los modelos anteriores. En cada iteración, se da más peso a las observaciones que fueron mal clasificadas o mal ajustadas por los modelos anteriores²⁴¹, forzando así al próximo modelo a centrarse en esos casos difíciles. Al final, las predicciones de todos los modelos se combinan para crear un modelo más robusto y preciso.

El *boosting* es especialmente efectivo cuando se utiliza con modelos débiles [como los árboles de decisión], es decir, modelos que realizan predicciones ligeramente mejores que una elección aleatoria. Algunos algoritmos populares de *boosting* incluyen AdaBoost, Gradient Boosting y **XGBoost**.

Vamos a resumir los resultados de aplicar todos estos algoritmos al *dataset*:

	SVM [1]	Regresión logística	Árboles de decisión	Bosques aleatorios	XGBost [2]
exactitud	0.9561	0.9386	0.9474	0.9649	0.9649
precisión	0.9459	0.9394	0.9577	0.9589	0.9589
sensibilidad	0.9859	0.9718	0.9577	0.9859	0.9859
F1	0.9655	0.9517	0.9577	0.9722	0.9722
especificidad	0.9070	0.8837	0.9302	0.9302	0.9302
NPV	0.9750	0.9500	0.9302	0.9756	0.9756
F1 negativo	0.9398	0.9157	0.9302	0.9524	0.9524

(1) Recordemos, visto en otro capítulo.

(2) Hemos hecho una búsqueda de los mejores hiperparámetros para este modelo.

Como podemos ver las métricas de los modelos se parecen mucho, de hecho la exactitud se sitúa entre el 93% y el 96% que, aún teniendo el cuenta el problema. No está mal.

El hecho de que modelos tan poderosos como los bosques aleatorios y XGBoost hayan dado los

240 En estadística, la moda es el valor o valores que aparecen con más frecuencia en un conjunto de datos. A diferencia de la media y la mediana, la moda puede ser usada para describir tanto datos numéricos como categóricos.

241 Falsos negativos y falsos positivos en caso de clasificación binaria.

mismos resultados, además teniendo en cuenta que en este último se usaron los hiperparámetros óptimos, nos puede hacer pensar que hemos llegado al límite de lo que podemos hacer con este *dataset*. O dicho de otras maneras, es el propio *dataset* el que impone un límite por ser demasiado simple²⁴².

16.6 Clasificación multiclas

La clasificación multiclas es una extensión de la clasificación binaria, donde el objetivo [la variable dependiente] es cualitativa y puede tomar más de dos valores o clases. Dado que ya se ha cubierto la clasificación binaria, en este apartado se presentarán varios métodos comúnmente usados para abordar problemas de clasificación donde hay más de dos clases. Los métodos varían en complejidad, desde técnicas que son extensiones naturales de los métodos binarios hasta algoritmos diseñados específicamente para el contexto multiclas.

El primer enfoque es usar clasificadores que, por su naturaleza, acepten clasificaciones multiclas de forma nativa. Por ejemplo los árboles de decisión y los algoritmos que los usan [*random forest* y *xgboost*, entre otros].

El segundo enfoque, si queremos usar un clasificador que no acepte más de dos clases, es convertir un problema de clasificación multiclas en varios problemas de clasificación binaria, y esto podemos hacerlo de dos formas:

- **Uno contra todos o contra el resto** [OvA/OvR, *One-vs-All/One-vs-Rest*]. Consiste en entrenar un clasificador binario para cada clase, donde la clase en cuestión se etiqueta como positiva y todas las restantes se etiquetan como negativas. Por ejemplo: Imagina que tienes un *dataset* con imágenes de perros, gatos y pájaros, y tu problema es crear un clasificador para clasificar en estas tres clases. Con la estrategia OvA [o OvR] creas tres clasificadores:
 1. Un clasificador binario que detecta perros o no perros [gatos y pájaros].
 2. Un clasificador binario que detecta gatos o no gatos [perros y pájaros].
 3. Un clasificador binario que detecta pájaros o no pájaros [gatos y perros].
- **Uno contra Uno** [OvO, *One-vs-One*]. Consiste en entrenar un clasificador binario para cada par de clases, donde cada clase se compara con cada una de las otras clases. Por ejemplo: De nuevo con el *dataset* con imágenes de perros, gatos y pájaros. Con la estrategia OvO creas tres clasificadores:
 - Un clasificador binario que detecta perros o gatos.
 - Un clasificador binario que detecta perros o pájaros.
 - Un clasificador binario que detecta gatos o pájaros.

242 Hay más interpretaciones, pero en este caso esta es la correcta.

Ambos tiene pros y contras, en OvA tienes que entrenar tantos clasificadores como clases tengas [3 en el ejemplo] pero los datos estarán desequilibrados²⁴³; eso no ocurre con OvO pero debes entrenar $\frac{N(N-1)}{2}$ clasificadores²⁴⁴.

La buena noticia es que el código te oculta todos estos detalles.

```
1  from sklearn.multiclass import OneVsRestClassifier
2  from sklearn.svm import SVC
3
4  model = SVC()
5
6  # Definir el clasificador
7  clf = OneVsRestClassifier(model)
8
9  # Entrenar el clasificador
10 clf.fit(X_train, y_train)
11
12 # Realizar predicciones
13 y_pred = clf.predict(X_test)
```

Fíjate. Sólo tienes que instanciar el algoritmo [SVC en este caso] y crear un objeto [*OneVsRestClassifier*, en negritas] que, usando el modelo que le pasas, creará tantos clasificadores como necesite y los entrenará con los datos apropiados. Desde tu punto de vista sólo tienes que saber qué haces, pero no cómo. Eso lo hace la biblioteca de Python.

Ahora necesitamos un *dataset* que posea múltiples clases. Vamos a usar uno clásico que, aunque esté orientado a imágenes y más susceptible de ser usado en *deep learning*, bien puede usarse en este contexto. Me refiero al MNIST de dígitos.

El conjunto de datos MNIST [*Modified National Institute of Standards and Technology*] Digits es uno de los *datasets* más populares en el mundo del aprendizaje automático. Es una colección de 70.000 imágenes, de 28x28 en escala de grises, de dígitos escritos a mano y se utiliza comúnmente para entrenar modelos de reconocimiento de imágenes y como un punto de referencia para nuevos algoritmos. En nuestro caso lo usaremos para comprobar cómo entrenar modelos que estamos viendo y evaluarlos.

Pero, las imágenes son de dos dimensiones y estamos trabajando con datos tabulados en columnas [cada columna correspondiente a cada variable independiente sería una dimensión]. ¿Cómo convertimos uno en otro? Es muy fácil, y de paso nos demuestra la potencia de estos algoritmos que son agnósticos respecto a información o conocimiento, ¡sólo ven números!

243 El desequilibrio o desbalanceo de las clases de los datos ocasiona sesgos, al estar una clase menos representada que el resto.

244 En nuestro caso igualmente 3 ($(3*2)/2$), pero imagina si tienes 10 clases, entonces deberías entrenar 45 clasificadores, mientras que con OvA sólo 10.

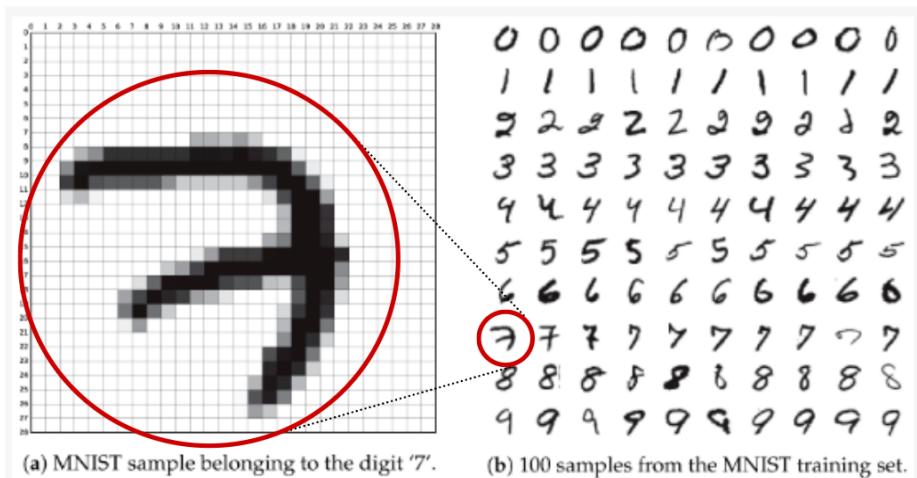


Figura 73: Muestra del dataset MNIST Digits

Cada una de las 70K imágenes del *dataset* está formada por 28x28 [784] pixels, cada uno de estos codificando un valor de intensidad o tono de gris [0 negro, 255 blanco]. Así que lo que hay que hacer es “aplanar” la imagen.



Figura 74: Ejemplo de aplanamiento de una imagen en 2D (5x5) a una forma en 1D (1x25)

Esta acción convierte una imagen de 28x28 en un vector de 1x768, esto es 768 números. Cada uno de estos números es una de las variables independientes. Ya tenemos la forma de convertir cada imagen a un formato tabular, como la de la Figura 61.

El código que sigue carga los datos [de la librería *tensorflow*], que ya se nos entregan divididos en datos de entrenamiento [*train_img* y *train_lbl*, 60k muestras] y datos de testeo [*test_img* y *test_lbl*, 10k muestras]. Seguidamente se lleva a cabo el dimensionamiento de cada una de las 70k imágenes.

```

1 import tensorflow as tf
2
3 # Cargar el conjunto de datos Fashion-MNIST
4 (train_img, train_lbl), (test_img, test_lbl) = tf.keras.datasets.mnist.load_data()

```

```

5
6 # Redimensionar las imágenes: de 28x28 a 784 (28x28) elementos en un único array
7 train_img = train_img.reshape(-1, 28 * 28)
8 test_img = test_img.reshape(-1, 28 * 28)
9
10 # Normalizar los valores de los píxeles en el rango [0, 1]
11 train_img = train_img.astype('float32') / 255
12 test_img = test_img.astype('float32') / 255

```

Finalmente se dividen cada uno de los pixels de todas las imágenes entre 255, esto ocasiona que los números que va a manejar el modelo se encuentran entre 0.0 y 1.0; la razón de esta normalización la encontramos en la mejora de la estabilidad numérica²⁴⁵.

Definiendo el problema: Queremos hacer una clasificación multiclas, de tal forma que, a partir de una imagen 28x28 de un número del 0 al 9 escrito a mano, obtengamos el dígito que está escrito. Esto es, un reconocimiento óptico de dígitos.

En el *notebook* del capítulo puedes encontrar todo el código que hace esto, aquí solo mostraré los resultados y como se evalúan los modelos multiclas.

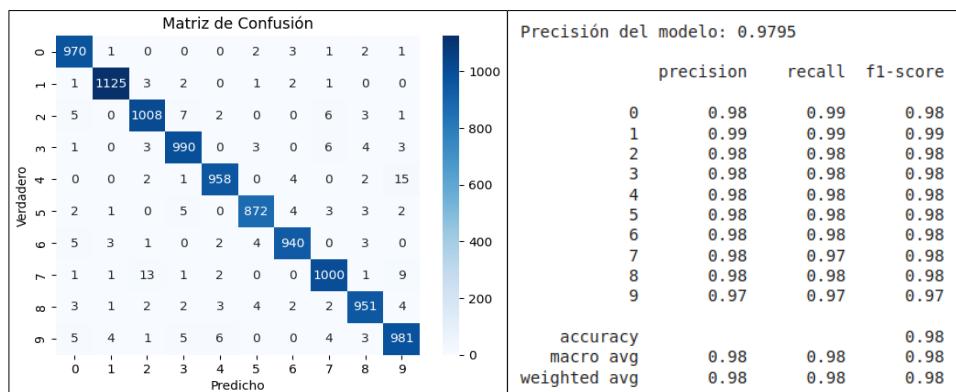


Figura 75: Matriz de confusión multiclas 10x10 y métricas

En la evaluación de modelos multiclas también tenemos matrices de confusión, pero en vez de ser 2x2 [en un clasificador binario] son $n \times n$ [para un clasificados de n clases].

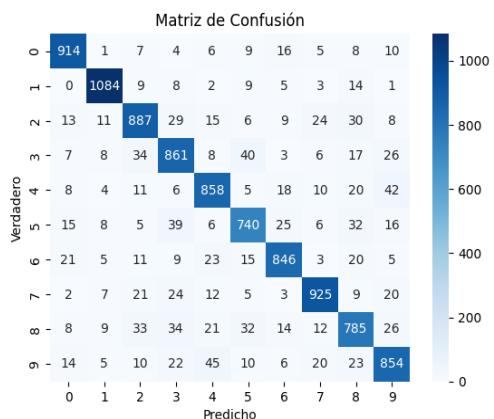
En la Figura 75 podemos observar el resultado de la evaluación del modelo creado con el algoritmo de regresión logística que vimos anteriormente. Como podemos ver ahora tenemos 10 clases [0-9] en dos ejes: "Predicho" [absisas] y "Verdadero" [coordenadas]. También debemos observar la diagonal, en donde se encuentran los ciertos: para una predicción del dígito 0 se acertaron 970 veces y se fallaron 23 veces.

Observa, los errores más comunes son predecir un 2 y que realmente sea un 7 [13 veces] y predecir un 9 y que realmente sea un 4 [15 veces]. Vuelve a la Figura 73 para ver los parecidos entre

²⁴⁵ Las operaciones que involucran números flotantes pueden sufrir de inestabilidad numérica. Mantener los números en un rango razonable (como 0.0-1.0) puede ayudar a mitigar este problema. En otros modelos este tipo de normalizaciones tienen otros propósitos, como hacer que todas las características compartan el mismo rango.

estos dígitos.

¿Y respecto a las métricas? Las más interesantes son *precision* y *recall* por dígito, además de su F1.



En la Figura 76 tenemos la misma estructura de información pero con los datos de un modelo basado en árbol de decisión. Como ya indicamos este algoritmo acepta de forma nativa clasificación multiclas. No usa OvA o OvO.

La precisión baja hasta un 87.54 de media, muy por debajo del óptimo. En efecto, este dataset es más difícil para un árbol de decisión. Todos los dígitos tienen bastantes errores de predicción, sólo se salva el dígito 1, con una precisión del 95% y un *recall* de 96% respectivamente.

Figura 76: Matriz de confusión multiclas 10x10 con un algoritmo árbol de decisión

Como indicamos anteriormente, a partir del algoritmo de árbol de decisión podemos aplicar una estrategia de ensamble, bien *bagging* [*random forest*] o *boosting* [*xgboost*], para potenciar su funcionamiento. En la Figura 77 y la Figura 78 podemos ver respectivamente sus métricas y matrices de confusión. Hemos mejorado, pero ¿se podrá hacer mejor?

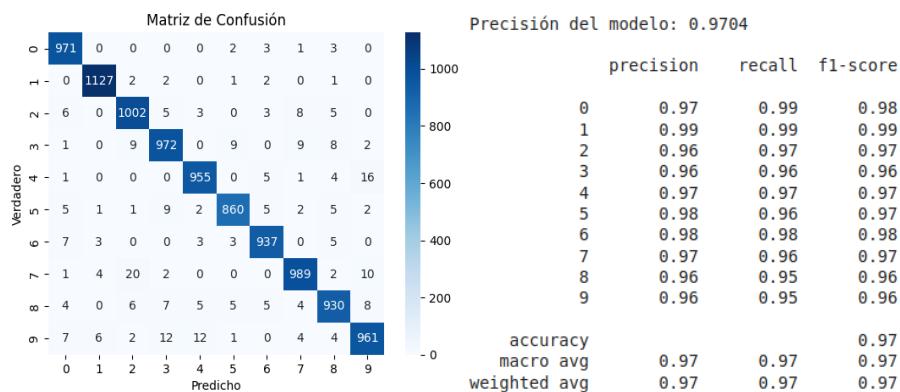


Figura 77: Matriz de confusión multiclas 10x10 y métricas con un algoritmo de random forest

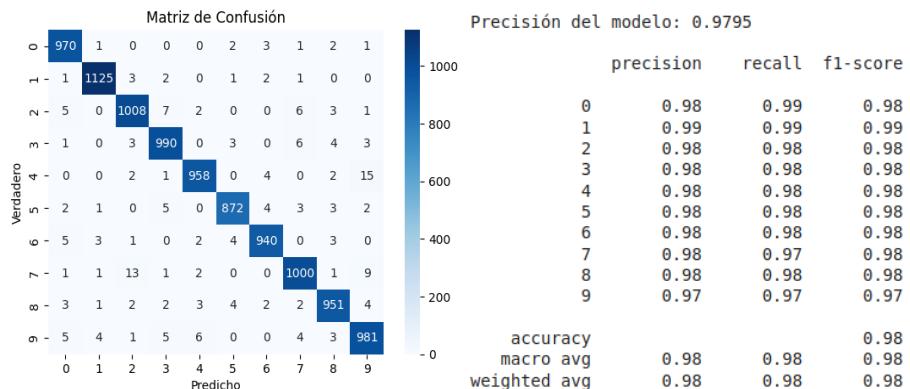


Figura 78: Matriz de confusión multiclas 10x10 y métricas con un algoritmo XBoost

Ahora vamos a aprobar con SVM [*Support vector machines*]. Pero este algoritmo, como ya hemos indicado, no soporta multiclasas [de forma nativa es un clasificador binario²⁴⁶], así que tenemos que usarlo mediante una estrategia que convierta un clasificador binario en otro multiclasa, como OvA o OvO. De hecho probaremos ambos, para poder comparar.

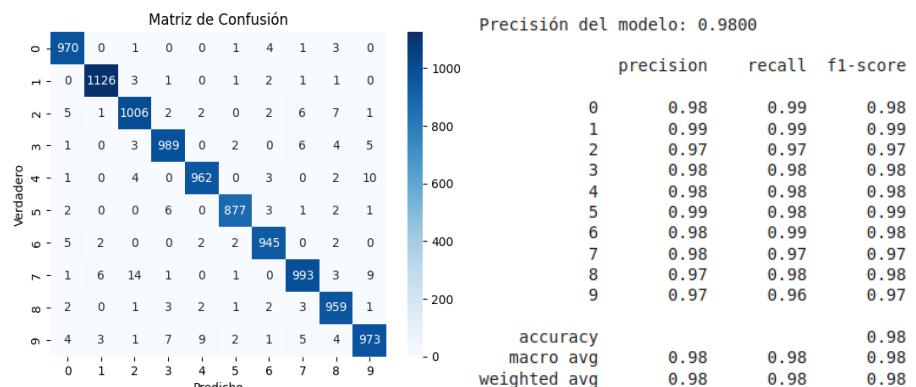


Figura 79: Matriz de confusión multiclas 10x10 y métricas para OvA y OvO con SVM

El resultado ha sido el mismo para ambas estrategias, sólo se diferenciaron en el tiempo que emplearon para entrenar el modelo. ¿Por qué? Bien, hay varias razones que lo explican, pero las dos más plausibles son [1] los datos son muy separables, como es el caso, y cuando se equivocan lo hacen sobre la mismas predicciones; [2] SVM es muy potente, y lo que hace lo hace muy bien, así que es posible que den el mismo resultado.

Parece que hemos mejorado respecto a los modelos creados anteriormente, ya que tenemos una precisión media de 98%, siendo la peores predicciones las que se corresponden a los dígitos 2, 7 y 9. Las máximas confusiones que el modelo predice son los pares [2, 7] y [9, 4], como era de esperar.

246 Recordemos, SVM calcula los “pasillos” que separan dos clases, no puede hacer “pasillos” que separen tres o más (ver apartado 14.3.1). Estos “pasillos” los llamamos hiperplanos.

Conclusiones: Los algoritmos de aprendizaje automático son muy buenos creando modelos que predigan un valor (regresión) o una clase (clasificación), siempre y cuando el dataset cumpla, principalmente, con: criterios de calidad y cantidad (ausencia de sesgos, diversidad, completud, etc.) e independencia.

Dependemos de los datos. Para mostrar este hecho, vamos a repetir todos los cálculos que hemos hecho en “clasificación multiclas”, pero con otro dataset de idénticas proporciones pero diferente contenido.



Figura 80: Conjunto de datos MNIST Fashion (parcial)

El dataset MNIST fashion tiene el mismo tamaño que el dataset MNIST digits: 70k muestras, cada una de ellas imágenes de 28x28 pixels en tonos de grises y 10 clases. Sin embargo el contenido de las imágenes no es el mismo y las clases no significan lo mismo.

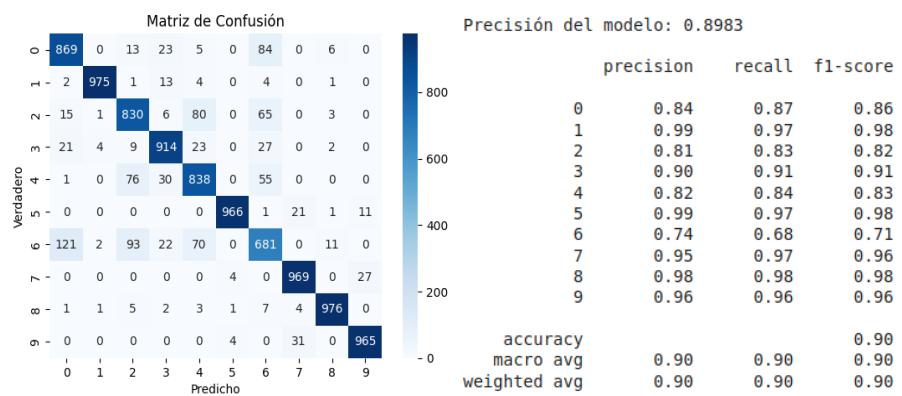


Figura 81: Matriz de confusión multiclas 10x10 y métricas para XGBoost sobre MNIST fashion

Este dataset contiene imágenes en baja resolución de 10 tipos de ropa (camisetas, pantalones, jer-

séis, vestidos, abrigos, sandalias, camisas, zapatillas, bolsos y botines]. Es un reto. Observa la Figura 80, diferenciar bolsos del resto es fácil, igual que sandalias y botines del resto, pero ¿diferencias bien sandalias y botines en las imágenes, camisetas y abrigos, etc.? En este caso los datos de los que partimos son de menor calidad por su baja resolución.

La Figura 81 muestra las métricas del mejor modelo: una precisión de menos del 89.8%.

Para finalizar este apartado sobre el aprendizaje supervisado, vamos a retomar un efecto pernicioso que debemos evitar a todas costa. **El sobreentrenamiento** [overfitting]. Este concepto ya lo tratamos brevemente al principio del capítulo, aquí vamos a profundizar en él.

El sobreentrenamiento es una situación común en el aprendizaje automático donde un modelo aprende demasiado bien a partir de los datos de entrenamiento pero generaliza mal ante nuevas observaciones. Podemos decir que aprende “memorizando” en lugar de generalizando.

Vamos a presentar dos modelos, entrenados sobre un *dataset* generado para la ocasión.

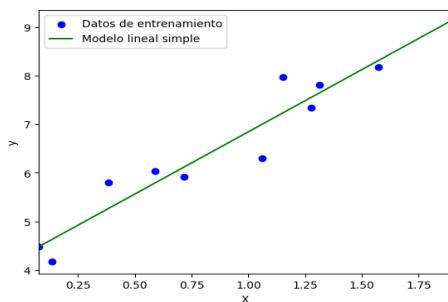


Figura 82: Dataset aleatorio modelado mediante regresión simple

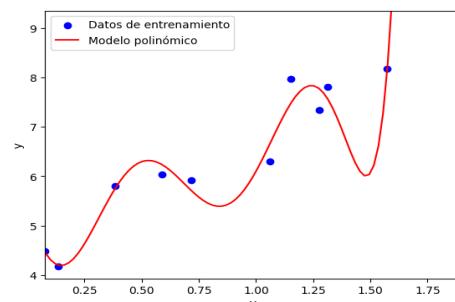


Figura 83: Dataset aleatorio modelado mediante regresión polinómica

El primero [Figura 82] arroja un MAE [respecto al *trainset*] de 0.3404 y el segundo [Figura 83] un MAE de 0.2071 ¿qué modelo se ajusta mejor a los datos con los que se ha entrenado? El modelo de regresión polinómico²⁴⁷ se ajusta bastante bien, casi un 40% mejor que la versión simple. Que obtengas un modelo casi un 40% mejor que otro es una buena noticia .. o no.

Como ya hemos dicho, tenemos que evaluar los modelos respecto al futuro: respecto a datos que aún no tenemos, y como las máquinas del tiempo están caras, debemos emplear técnicas²⁴⁸ que “simulen” tener datos del futuro. En nuestro caso entrenar y evaluar con conjuntos de datos distintos: *trainset* y *dataset*. Entrenamos con el primero y evaluamos con el segundo. Vamos a ver estas gráficas, pero con el conjunto de testeo [Figura 84 modelo de regresión simple con los datos de testeo y Figura 85 modelo de regresión polinómica con los datos de testeo].

247 De grado 7, por cierto.

248 Hay más técnicas (por ejemplo evaluar usando *k-fold* durante el entrenamiento), pero esta es la más simple y sus conceptos son la base de casi todas las demás.

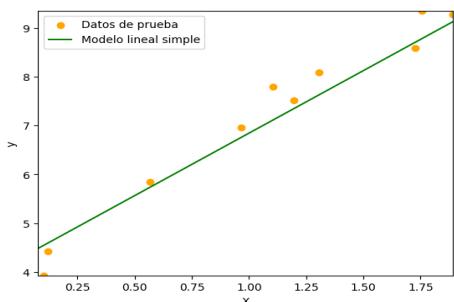


Figura 84: Dataset aleatorio modelado mediante regresión simple

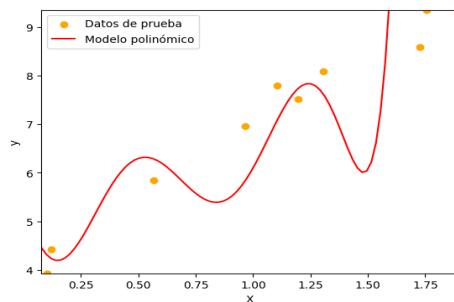


Figura 85: Dataset aleatorio modelado mediante regresión polinómica

¿Cuál ajusta mejor ahora? El MAE para la regresión lineal simple simple [izquierda] arroja un valor de 0.3267, mientras que la regresión polinómica, un 40% mejor si la evaluamos sólo con los datos de entrenamiento, ahora arroja un MAE de 20.6026 usando los “datos del futuro”, el *testset*. ¡Un 99% peor!

Pero ¿no habrá mucha diferencia en la predicción al usar sólo una parte del conjunto de entrenamiento y dejar otra parte – intocable mientras se entrena – para la evaluación? Veámoslo: la diferencia entre los MAE de la regresión lineal simple usando el *trainset* y el *testset* es sólo de un 4%. Nuestro modelo, a priori, sólo se equivoca un 4% cuando se enfrenta a datos no vistos durante el entrenamiento.

¿Y qué tiene que ver esto con el *overfitting*?

*Atención: una de las formas más efectivas de detectar el *overfitting* es calcular la diferencia de las métricas cuando se usa el conjunto de entrenamiento y el conjunto de *testeo*. Básicamente un modelo que está sobreentrenando se comporta muy bien en datos del conjunto de entrenamiento, pero muy mal en datos del conjunto de *testeo**²⁴⁹.

En nuestro ejemplo el modelo lineal simple tenía un porcentaje del 4% de pérdida de acierto [media de los errores absolutos] entre los conjuntos de *trainset* y *testset*. El modelo polinómico obtuvo un 99% de pérdida. ¿Por qué? Porque este último cayó en el sobre-entrenamiento u ***overfitting***.

Algunos de los algoritmos son propensos a crear modelos ya que, por su naturaleza, tienden a “memorizar” los datos del *dataset* durante el entrenamiento, ocasionando que a la hora de predecir con datos que no ha visto anteriormente obtengan unos errores muy superiores. Por ejemplo los árboles de decisión.

²⁴⁹ Igual que un alumno que aprende de memoria la lección, pero no la entiende.

16.7 Agrupamiento

Este apartado se centra en un caso particular dentro de la familia de algoritmos que no necesitan etiquetas para aprender a partir de los datos.

Recordemos: en la Figura 61 vimos que los *dataset*, entendidos estos como *muestras*²⁵⁰ y *características*²⁵¹, son la fuente del aprendizaje cuyo fin es crear un modelo que, a partir de las variables independientes, pueda generalizar y predecir el valor de la etiqueta que le corresponde. Esto se llama *aprendizaje supervisado* porque tiene que existir al menos una columna con la solución²⁵² para poder aprender.

El aprendizaje supervisado, que es el que hemos visto hasta ahora en este capítulo, se denomina “supervisado” precisamente porque necesita la variable dependiente u objetivo con la que se “supervisa” el entrenamiento.

En este apartado hablaremos del **aprendizaje no supervisado**, en el cual el *dataset* no tiene una columna o variable objetivo. Todas son variables independientes, o al menos son tratadas así²⁵³.

Y dentro del aprendizaje no supervisado vamos a ver la técnica de agrupamiento [o *clustering*]. El agrupamiento es una técnica dentro del aprendizaje no supervisado que tiene como objetivo segmentar un conjunto de datos en grupos o clústeres homogéneos. En otras palabras, busca agrupar elementos de manera que estos, dentro de un mismo grupo, sean más similares entre sí que con los elementos de otros grupos.

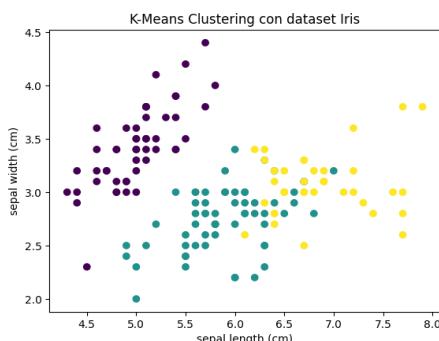


Figura 86: Ejemplo de clustering con el dataset Iris en 2D.

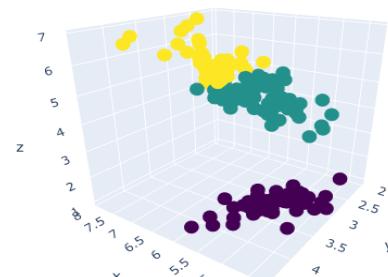


Figura 87: Ejemplo de clustering con el dataset Iris en 3D.

En la Figura 86 podemos ver una representación de un agrupamiento realizado con el *dataset* Iris

250 Las filas.

251 Las columnas, también llamadas variables o características. Estas podemos dividirlas en dos tipos: variables independientes y variable dependiente (objetivo).

252 La variable dependiente, u objetivo. Puede haber más de una, pero aquí no lo estamos contemplando.

253 Un pequeño apunte: es muy normal usar un *dataset* típico de aprendizaje supervisado y usarlo para aprendizaje no supervisado, eso sí eliminando (pero no olvidando) la variable dependiente.

en dos dimensiones. Al estar en 2D parece que dos de los clústeres solapan, pero en la Figura 87, en 3D interactiva [ver el notebook correspondiente], he intentado poner la perspectiva en la que podemos ver con más claridad que, aunque comparten frontera, apenas solapan.

El código que entrena el modelo es muy simple:

```

1 # Utilizar K-Means para encontrar 3 clusters
2 kmeans = KMeans(n_clusters=3)
3 kmeans.fit(data)
4
5 # Obtener las etiquetas de los clusters y los centroides
6 labels = kmeans.labels_
7 centroids = kmeans.cluster_centers_

```

En donde la variable *data* contiene el dataset Iris sin la columna “*species*”, usa un algoritmo llamado **KMeans** el cual obtiene – después de entrenar con *fit* – las etiquetas [*labels*] y los centroides [*centroids*]. Esto da lugar a dos preguntas: ¿qué son las etiquetas?

El resultado de un modelo de *clustering* es un conjunto de etiquetas, tantas como filas tiene el *dataset*, en donde la etiqueta *n* indica a qué clúster pertenece la fila *n*.

¿Qué son los centroides? Explicar que son los centroides nos da la oportunidad para explicar cómo funciona KMeans.

Antes de empezar necesitamos definir un concepto: **similitud**. Para llevar a cabo el agrupamiento, es necesario definir alguna métrica que mida la similitud entre los puntos de datos. En capítulo apartado 10.3, y en apartados anteriores a este, vimos métricas de similitud y distancia entre textos. Ahora básicamente necesitamos medir de alguna forma una distancia entre las muestras del *dataset*. Si este tiene *n* columnas y cada columna contiene números, podemos ver cada fila como un vector de *n* componentes. De esta manera podemos ver cada muestra como un punto de *n* dimensiones. ¿cómo calcular la distancia entre puntos en un espacio de *n* dimensiones?

$$\text{Distancia Euclídea}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

La distancia Euclídea puede ser extendida para calcular un valor numérico [distancia] a partir de dos puntos en un espacio de *n* dimensiones; y es la manera que por defecto²⁵⁴ utiliza KMeans.

Ya estamos en disposición de ver como funciona este algoritmo para crear modelos de *clustering*:

1. Seleccionados *c* puntos como centroides iniciales. Cada centroide representa a un clúster y por tanto tiene su propia etiqueta.
2. Para el resto de puntos: asigna cada uno de ellos al clúster cuyo centroide sea el más cercano, según la métrica que estemos usando.

²⁵⁴ Otras métricas de similitud o distancia son: distancia Manhattan, similitud del coseno, distancia Jaccard y la distancia Mahalanobis; entre otras. Observar que se pueden medir las distancias entre textos según su similitud (Capítulo 10). También es posible calcular la similitud entre imágenes.

3. Actualiza los centroides: actualizamos los centroides re-calcularlos en función de los puntos del clúster al que representan.
4. Repite los pasos 2 y 3 hasta que los centroides apenas cambien.

El Vídeo 13 es una animación en donde se muestra este entrenamiento. La única duda que podría plantearse es ¿por qué se mueven los centroides? O, mejor ¿por qué al actualizar los centroides, estos se mueven? Para re-calcular el centroide j -ésimo se toman todos los puntos del clúster j y se calcula:

$$\mu(j) = \frac{1}{|C(j)|} \sum_{i \in C(j)} x_i$$

Parece complicado pero en realidad es muy fácil: para cada clúster, sumamos los puntos que pertenecen al dicho clúster y el resultado lo dividimos entre en número de puntos en el clúster. ¡Es una media! Por ejemplo, supongamos que el clúster j está formado por los siguientes 5 puntos de 3 dimensiones:

- P1=[1, 2, 3]
- P2=[4, 5, 6]
- P3=[7, 8, 9]
- P4=[10, 11, 12]
- P5=[13, 14, 15]

El punto suma resultante es [35, 40, 45] y al dividir este entre 5, obtenemos [7,8,9]²⁵⁵. El nuevo centroide. De esta forma, cada vez que hacemos el punto [2] tenemos que recalcular los centroides y estos cambian en [3]. Finalizamos cuando los centroides no cambien porque en el punto [2] no hay cambios en los clústeres.



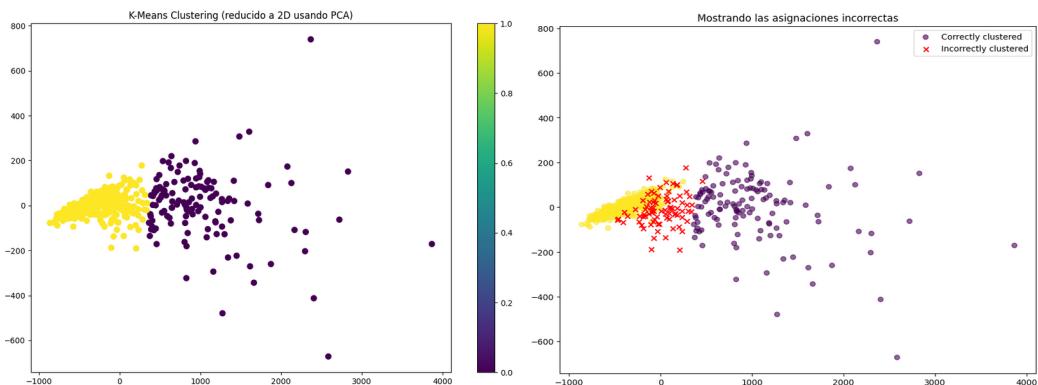
Video 13: Animación del algoritmo KMeans durante el entrenamiento

El agrupamiento de datos se utiliza para entrenar modelos que tengan la capacidad de aprender sin supervisión a diferenciar los datos por medio de algún patrón – agrupándolos – para, posteriormente, ante la presencia de nuevos datos clasificarlos en algunos de estos grupos.

Volvamos al *dataset* de cáncer de pecho, saquemos la columna objetivo que indica si es benigno o no y hagamos una clusterización de los datos, a ver si podemos clasificar aun así los datos en ma-

²⁵⁵ Que sea igual que uno de los puntos es mera coincidencia.

lignos o benignos. Las siguientes imágenes muestran la clusterización de las 30 variables independientes, proyectadas en 2D por medio de un algoritmo de reducción de la dimensionalidad, a la izquierda agrupadas en dos clústeres. Es importante recalcar que al algoritmo de clusterización sólo le fue indicado cuantos clústeres queríamos tener, fue el propio algoritmo el que estableció el criterio para separar ambos grupos.



A la derecha tenemos la misma agrupación, pero ahora destacando en rojo las asignaciones incorrectas que ha realizado la clusterización basándose sólo en los datos y ayudado, ahora sí, por la columna objetivo que originalmente hemos retirado del *dataset* para usarla como forma de evaluar la clusterización.

Nota: la clusterización en sí ha sido capaz de separar en dos grupos las muestras sólo basándose en los patrones internos de las mismas, no en la columna objetivo, la cual sólo se usó para evaluar el modelo y mostrar los errores de asignación con una intención pedagógica.

RETOS DEL CAPITULO 16

1. Investiga y describe en tus propias palabras qué es el "entrenamiento de modelos" en el contexto del aprendizaje automático.
2. Interroga a tu IA favorita, cuan importante es el conjunto de datos en el aprendizaje automático.
3. ¿Cuál es la diferencia entre un conjunto de entrenamiento y un conjunto de prueba? ¿Por qué son necesarios ambos? Intenta deducirlo sin ayuda externa.
4. Investiga y explica qué es la validación cruzada y cómo se utiliza en la evaluación de modelos.
5. ¿Qué es la matriz de confusión y para qué se utiliza en la evaluación de modelos de clasificación?
6. Dialoga con tu agente conversacional favorito, ¿Qué significa el término "sobreajuste" en el contexto del aprendizaje automático? ¿Cómo se puede evitar?
7. Investiga y explica la diferencia entre regresión y clasificación en el aprendizaje automático.
8. ¿Qué son los datos abiertos? Busca sitios web públicos de tu ayuntamiento o comunidad en donde comparten datos.
9. Repite el reto anterior, pero a nivel nacional.
10. Investiga cómo se elige el algoritmo adecuado para entrenar un modelo a partir de unos datos dados.
11. Investiga y describe el concepto de "conjunto de validación" y su papel en el ajuste de hiperparámetros.
12. Dialoga con tu IA favorita, qué es y para que sirve una curva ROC. Indícale tu nivel de conocimientos para que no sea muy técnico.
13. Investiga ejemplos de al menos dos técnicas de reducción de dimensionalidad utilizadas en el aprendizaje automático. ¿Por qué las usamos?
14. Realiza un experimento de entrenamiento y evaluación de un modelo de regresión lineal utilizando una herramienta en línea basados en *notebooks*.

Capítulo 17

REDES NEURONALES ARTIFICIALES

Este capítulo tiene como objetivo introducir los conceptos básicos de las redes neuronales artificiales y el aprendizaje automático denominado **deep learning**. Los temas abarcan desde la definición y aplicación del aprendizaje automático hasta la estructura y entrenamiento de las redes neuronales. También se exploran diversas arquitecturas de redes neuronales y sus aplicaciones en diferentes áreas.

17.1 Introducción a las Redes Neuronales Artificiales

Cuando hablamos de la historia de la inteligencia artificial [apartado 1.4], hicimos una breve mención a modelo neuronal creado por Warren McCulloch y Walter Pitts. Vamos a dedicar este capítulo a todo lo que se ha producido desde entonces, que podamos plasmar en un sólo capítulo.

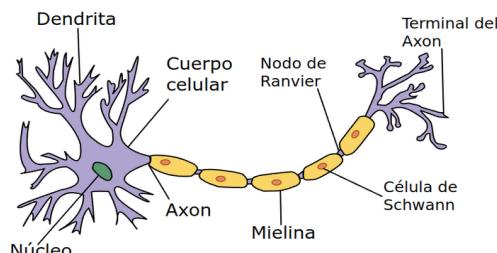


Figura 88: Ilustración de una neurona biológica

Fuente: Wikipedia Commons

Una neurona biológica típica [Figura 88] consta de tres partes principales: [1] El **soma o cuerpo celular**, es el centro de la neurona y contiene el núcleo, que alberga la información genética de la célula; [2] **dendritas** que son extensiones ramificadas que salen del soma, funcionan como antenas electro-químicas, recibiendo señales eléctricas de otras neuronas; y [3] **axón**, es una extensión larga y delgada que transmite señales eléctricas desde el soma a otras células o neuronas y que ter-

mina en múltiples terminales axónicas o botones sinápticos.

El funcionamiento de una de estas neuronas comienza por las dendritas, las cuales reciben señales, generalmente en forma de neurotransmisores liberados en una sinapsis por una neurona vecina. Estas señales pueden ser excitatorias [que tienden a despertar la neurona] o inhibitorias [que tienden a inhibir la actividad neuronal]. Si la suma de las señales excitatorias e inhibitorias que una neurona recibe alcanza un cierto umbral, la neurona se *activa* y genera un impulso eléctrico llamado potencial de acción. Este se propaga a lo largo del axón hasta sus terminales, donde provoca la liberación de neurotransmisores en la sinapsis. Estos neurotransmisores luego afectan a otras neuronas o células a las que está conectada la neurona.

Estas neuronas forman tupidas redes con arquitecturas, formas y configuraciones muy diferentes. Los puntos de contacto entre una neurona y otra²⁵⁶ se denomina **sinapsis**. No es un contacto directo, sino un pequeño espacio llamado hendidura sináptica. Cuando el potencial de acción llega a un botón sináptico, provoca la liberación de neurotransmisores en esta hendidura. Los neurotransmisores luego se unen a receptores específicos en la membrana de la célula receptora, transmitiendo así la señal.

Mi IA favorita lo explica con más sencillez: Las neuronas son como pequeños mensajeros en nuestro cerebro. Tienen ramas llamadas dendritas que "escuchan" mensajes de otras neuronas, y una "cola" llamada axón que envía mensajes. Cuando una neurona recibe suficientes mensajes excitantes (como un "¡sí, actúa!") en comparación con los mensajes inhibidores (como un "espera un momento"), se activa y envía una señal eléctrica a través de su axón. Al final del axón, libera sustancias llamadas neurotransmisores en un espacio (sinapsis) que luego se unen a la siguiente neurona, pasando el mensaje. Así es como nuestras neuronas se comunican y trabajan juntas para que podamos pensar, sentir y actuar.

Es difícil entender cómo a partir de un funcionamiento tan simple, nuestro cerebro es capaz de hacer tantas cosas tan complejas²⁵⁷. Por ello desde siempre el conocimiento humano vio en el cerebro la frontera, la última aventura, el reto. ¿Y qué reto mejor que hacer que una máquina emule el funcionamiento del cerebro?

Como dijimos, en 1943, Warren McCulloch y Walter Pitts introdujeron un revolucionario modelo neuronal²⁵⁸ conocido como *Threshold Logic Unit* [TLU] o *Linear Threshold Unit*. Este pionero diseño conceptual no solo marcó el nacimiento de los modelos neuronales modernos, sino que también ha servido como fuente inagotable de inspiración para generaciones posteriores de teorías y aplicaciones que estas tienen.

En la actualidad, en donde las implementaciones computacionales de neuronas artificiales se realizan con abrumadora frecuencia en sistemas digitales, el modelo McCulloch-Pitts ofrece ventajas

²⁵⁶ Puede ser otra neurona, un músculo, o una glándula. Aquí presupongo que es otra neurona.

²⁵⁷ El todo es superior a la suma de sus partes.

²⁵⁸ Modelo matemático basado en los trabajos de Santiago Ramón y Cajal.

prácticas innegables. De esta forma, el modelo McCulloch-Pitts sigue siendo no solo una figura histórica de importancia, sino también un recurso contemporáneo de relevancia creciente en el campo de las neurociencias computacionales.

Metáfora biológica de las neuronas

Las redes neuronales artificiales están inspiradas en el funcionamiento del cerebro humano, en el cual las neuronas biológicas son células especializadas en recibir, procesar y transmitir información mediante señales electro-químicas, como indicamos antes. En el contexto de las redes neuronales artificiales, se utiliza un modelo matemático simplificado de las neuronas para simular el procesamiento de información.

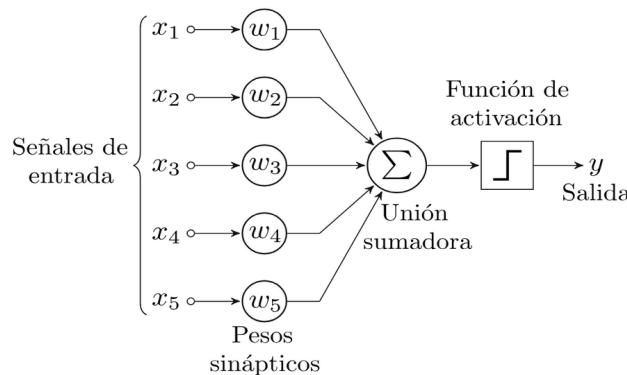


Figura 89: Esquema básico del modelo matemático de una neurona con 5 entradas

Fuente: Wikipedia Commons

Una neurona artificial consta de tres componentes principales: las entradas (x_i), los pesos (w_i) y la función de activación, cuya salida es la salida de la neurona artificial. Las entradas son los datos de entrada que la neurona recibe, cada uno multiplicado por un peso correspondiente. Los pesos representan la importancia relativa de cada entrada en el procesamiento de la información. La función de activación determina si la neurona se activa o no, en función de la suma ponderada de las entradas y los pesos.

$$y = f_{\text{activación}} \left(\sum_{i=1}^n x_i w_i \right)$$

Calcular la salida (y) de una neurona artificial es sorprendentemente simple: para cada i de 1 a n , siendo n el numero total de entradas, multiplicas x_i con el peso que le corresponde w_i , y los sumas todos. El resultado de esa suma es introducido en una función de activación ($f_{\text{activación}}$), cuya salida será la salida de la neurona artificial. Simple.

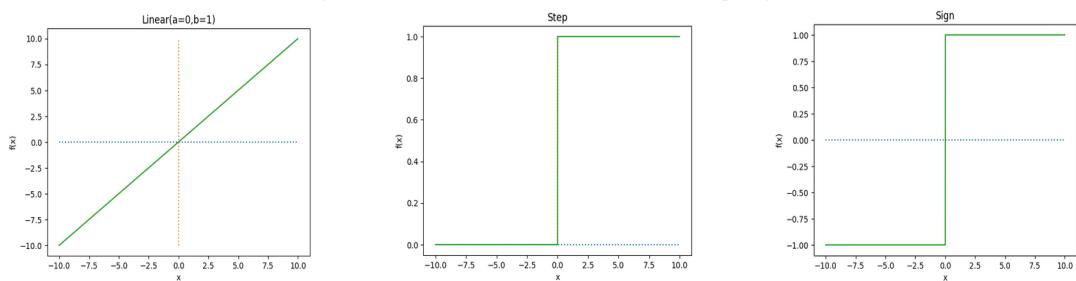
Ahora surgen dos dudas, ¿que es una función de activación? y ¿cómo aprende una neurona artificial? Empecemos por saber y entender que es una función de activación.

El objetivo²⁵⁹ del modelo de neurona artificial es ser activada o ser inhibida ante determinadas entradas. Esto es, que ante determinadas entradas (x_i) emita un número diferente de cero o un cero (y).

Y esa es la función [original] de la función de activación: a partir de la suma ponderada de las entradas, activarse o desactivarse, esto es, emitir un 0 [inhibida] o un número diferente a 0 [activa].

Vamos a ver unas funciones de activación muy simples para entender mejor esto.

Figura 90: Funciones de activación básicas (linear, step y sign)



La Figura 90 muestra varias funciones de activación²⁶⁰. Recordemos que sus x [abscisas] es la suma ponderada de sus entradas, y sus salidas $f_{activación}(x)$ es la salida de la neuronal artificial. Tomando la función *step* como ejemplo, si su entrada es menor o igual que 0, su salida es cero; mientras que si su entrada es mayor que 0 su salida es 1. Si consideramos que 1 es activa y 0 inhibida, tenemos un mecanismo que se activa o no dependiendo de su entrada y que – además – puede aprender.

17.2 El aprendizaje en neuronas artificiales

El proceso de aprendizaje de una neurona artificial está relacionado con la búsqueda de la combinación de pesos (w_i) que hacen que esta emita la salida deseada ante determinadas entradas. En general, todas las arquitecturas de redes neuronales son entrenadas con un algoritmo de aprendizaje, el cual consiste en encontrar la combinación de pesos que, a partir de determinadas entradas se produzcan determinadas salidas.

Nota: simplificando mucho, el proceso de aprendizaje de una o mas neuronas puede reducirse a un proceso de optimización, en concreto de minimización: buscar el conjunto de pesos (w_i) tal que minimicen el error ($|y - \hat{y}|$) que comete la neurona o red de neuronas ante determinadas entradas (x_i).

Las primeras propuestas de cómo entrenar una neurona se basaron en **la regla de Hebb** [1949], la

259 Originalmente las entradas x_i era booleanas, 0 o 1, igual que la salida y . Aquí damos un paso más y hablamos de entradas y salidas numéricas.

260 Me adelantará un poco: en realidad trabajamos con funciones de activación diferentes a estas, pero la mayor parte basadas en ellas.

cual propugnaba que el aprendizaje ocurre mediante el fortalecimiento de las conexiones entre neuronas que se activan simultáneamente. Este concepto sentó las bases para muchos algoritmos posteriores.

Posteriormente, en 1957, Frank Rosenblatt introdujo el **Perceptrón**, una red neuronal de una sola capa, y su correspondiente regla de aprendizaje. Este algoritmo solo podía resolver problemas linealmente separables²⁶¹.

Y, en 1986, Rumelhart, Hinton y Williams introdujeron el **algoritmo de retropropagación** [*backpropagation*], que hizo posible entrenar redes neuronales multicapa y resolver problemas más complejos que los que podían manejar los perceptrones. Para describir este algoritmo necesitamos más de una neurona [artificial].

Desde la década de los 90 se han creado multitud de arquitecturas de redes neuronales artificiales, pero aunque estén potenciadas gracias a técnicas matemáticas²⁶² y tecnologías de optimización y aceleración, su modelización, la forma de entender matemáticamente como funcionan, sigue siendo el modelo de McCulloch-Pitts.

Vamos a condensar cómo funciona el algoritmo de retropropagación.

El algoritmo de retropropagación es el método utilizado para entrenar redes neuronales mediante la actualización iterativa de sus pesos. Es un algoritmo basado en la regla de la cadena del cálculo diferencial. Recordemos, igual que siempre partimos de un conjunto de entrenamiento, *trainset*, formado por muestras:

1. **Inicialización:** Antes de que comience el proceso de entrenamiento, los pesos de la red neuronal se inicializan con valores aleatorios.
2. **Feedforward:** Se introduce la siguiente muestra del conjunto de entrenamiento en la red, y se propaga a través de sus capas hasta obtener una salida. Esta salida se compara con la salida deseada [etiqueta] para calcular el error.
3. **Cálculo del error:** El error se calcula generalmente usando una función de pérdida [*loss*], como MAE²⁶³ o MSE²⁶⁴. Esta función de coste cuantifica la diferencia entre la salida prevista por la red y la salida real o deseada.
4. **Retropropagación del error:** A partir de la capa de salida, se calcula la derivada de la función de pérdida con respecto a cada peso, es decir, el gradiente de la función de pérdida. Esto se hace retropropagando el error a través de cada capa, desde la última hasta la primera, utilizando la regla de la cadena para calcular las derivadas parciales. Esta retropropagación del error da al algoritmo su nombre.

Este punto es el difícil de entender si no se domina el cálculo diferencial. La idea principal

261 Es fácil de entender si volvemos al tema de los clasificadores binarios: un problema linealmente separable es aquel en el que puedes poner una línea recta entre dos clases. ¿Recuerdas los hiperplanos de SVM?

262 Principalmente, cálculo, álgebra lineal y estadística/probabilidad.

263 Recordemos: Median absolute error, error absoluto medio.

264 Recordemos: Median squared error, error cuadrático medio.

es ir desde la salida y hacia la entrada, capa por capa, [por eso se llama retropropagación] modificando los pesos de forma que minimicen ese error [*loss*] si vuelven a ver la misma muestra en la entrada.

5. **Actualización de pesos:** Una vez que se tiene el gradiente, se utilizan algoritmos de optimización, como el descenso del gradiente, para ajustar los pesos en la dirección que reduce el error. La magnitud del ajuste está determinada por la tasa de aprendizaje, un hiperparámetro importante en el entrenamiento de redes neuronales.
6. Y volvemos al punto 2 mientras haya muestras en el conjunto de entrenamiento.

El proceso se repite para cada muestra o ejemplo en el conjunto de entrenamiento, a menudo múltiples veces. Cada paso completo a través del conjunto de entrenamiento se denomina ***epoch***.

Haciendo una analogía, imagina que estás enseñando a una niña los tiros libres de baloncesto. Al principio, esta no tiene idea de cuánta fuerza usar o en qué ángulo lanzar. Cada vez que lanza la pelota, tú observas dónde cae: si fue demasiado corto, demasiado largo, si fue hacia la izquierda o hacia la derecha del aro. El proceso de enseñar a ajustar su tiro es similar al algoritmo de retropropagación:

1. **Inicialización:** Comienza a lanzar la pelota sin ninguna técnica. Sus movimientos iniciales son aleatorios.
2. **Feedforward:** Ella lanza la pelota y tú observas dónde cae.
3. **Cálculo del error:** Comparamos dónde querías que cayera la pelota [en el aro] y dónde realmente cayó. Si cae corto, sabes que necesita más fuerza. Si va hacia la derecha, necesita ajustar su puntería a la izquierda, etc.
4. **Retropropagación del error:** Le das retroalimentación sobre lo que hizo mal y cómo puede corregirlo. Por ejemplo, "*lanza con un poco más de fuerza*" o "*apunta más a la izquierda*", etc.

Este es el punto en que debemos pararnos para explicar lo que es retropropagar el error siguiendo la regla de la cadena usando derivadas parciales, pero aplicado a la analogía. En el lanzamiento de un tiro libre influyen varias "capas", los dedos, la muñeca, el codo, hombros, la posición del resto del cuerpo [especialmente rodillas], etc. Para corregir un tiro no llega sólo con corregir la mano, hay que propagar ese error hacia atrás en la cadena, asumiendo cada "capa" parte del error del resultado anterior, para conseguir que el siguiente tiro consiga un error menor. Esto se hace entre este punto y el siguiente.

5. **Actualización de pesos:** En el siguiente intento, ella ajusta su tiro²⁶⁵ basándose en tu retroalimentación. Esto es el equivalente a actualizar los pesos en la red neuronal.
6. **Iteración:** La niña sigue practicando [vuelve al punto 2] y ajustando su tiro con cada lanzamiento usando tus consejos hasta que se convierte en un experto/a en tiros libres.

Esta analogía es mala para explicar lo que es un *epoch*. Pero, siendo muy aventurados, un *epoch* en este ejemplo sería un día completo de entrenamiento, suponiendo que se va a entrenar día tras días hasta conseguir la perfección [o casi].

²⁶⁵ Aquí, ajustar el tiro no es volver a tirar. Eso lo hace en el punto 2 de nuevo. Ajustar el tiro sería cuando un jugador, antes de tirar, ensaya el tiro sin balón.

Pongamos un ejemplo más cercano a la técnica. Vamos a enseñar a un perceptrón a contar.

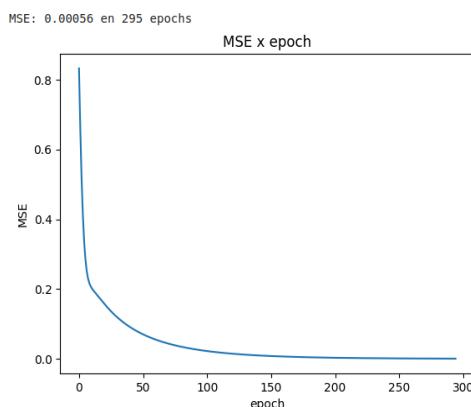
```

1 trainset=[  
2     ((0,0,0), 0),  
3     ((0,0,1), 1),  
4     ((0,1,0), 1),  
5     ((0,1,1), 2),  
6     ((1,0,0), 1),  
7     ((1,0,1), 2),  
8     ((1,1,0), 2),  
9     ((1,1,1), 3)  
10 ];

```

Este es el conjunto de entrenamiento. Por simplicidad en este ejemplo no va a haber conjunto de testeo. El *trainset* está compuesto²⁶⁶ por una entrada de tres bits [0 o 1] y una salida, indicando cuantos 1s hay en la entrada.

Hemos entrenado un perceptrón de tres entradas, con una función de activación ReLu con $a=0,001$, un *learning rate*²⁶⁷ de 0,01 y lo hemos entrenado durante 1000 epochs. Como función de coste usé MSE [*median squared error*].



Finalmente el aprendizaje finalizó en el epoch 295, cuando el error disminuyó tanto [0,00056] que no merecía la pena seguir entrenándolo²⁶⁸.

Como resultado podemos ver esta gráfica que muestra en el eje de abscisas los epochs y en el eje de coordenadas el error cometido en ese epoch [MSE].

Esta gráfica se denomina **curva de aprendizaje**, y es la guía de lo que está pasando²⁶⁹ o pasó con el entrenamiento de una red neuronal. En nuestro caso, empezamos muy mal, con un gran error, y paulatinamente esta

fue disminuyendo, hasta alcanzar un pleno pasados los 250 epochs. Cuando acabó, el error cometido era muy muy pequeño.

Un apunte: estamos entrenando y evaluando con el mismo conjunto de datos, lo más probable es que este modelo haya sobreentrenado (overfitting), pero en este caso no nos importa.

²⁶⁶ Es una estructura ligeramente compleja de python: una lista de tuplas, cada una de ellas siendo una tupla de 2 elementos, el primero una tupla de tres escalares y el segundo un escalar.

²⁶⁷ Velocidad con la que aprende, relacionado con el algoritmo del descenso del gradiente. Ver §Error: no se encontró el origen de la referencia.

²⁶⁸ La finalización es automática, al igual que el entrenamiento.

²⁶⁹ Si el entrenamiento dura días o meses, ir viendo esta curva (con más elementos) ayuda a saber cómo está yendo el aprendizaje.

Vamos a probarlo:

```
Introduce la entrada (a b c): 1 1 1
3.0
Introduce la entrada (a b c): 1 0 1
2.0
Introduce la entrada (a b c): 0 0 0
0.0
```

Funciona. Si introduzco 1 1 1, devuelve 3; y así el resto. Hemos creado un perceptrón que es capaz de contar el número de 1s que hay en la entrada. ¿Y qué pasa si en vez de 0 y 1, introducimos números mayores?

```
Introduce la entrada (a b c): 1 2 3
6.0
Introduce la entrada (a b c): 2 3 4
9.0
Introduce la entrada (a b c): 1 3 4
8.0
```

¿Qué ha pasado aquí?

¡Le hemos enseñado a contar 1s y este ha aprendido, sin ejemplos, a sumar números naturales! Como veremos las redes neuronales artificiales no sólo aprenden bien, también generalizan muy bien²⁷⁰.

Si las redes neuronales artificiales funcionan tan bien, ¿por qué tardaron tanto en salir de los laboratorios? Hay dos razones, la primera porque para entrenar una red neuronal hacen falta muchos datos, de muy buena calidad y muchísima potencia de computación. Elementos que no estuvieron a nuestro alcance hasta inicios del siglo XXI.

Para la segunda razón tengo que explayarme más. Marvin Minsky fue un prominente científico, como vimos cuando explicamos la historia de la IA [Figura 2]. En la década de 1960 hubo mucho entusiasmo en torno al perceptrón y, a medida que se publicaban más investigaciones sobre ellos, se generó mucho optimismo sobre sus potenciales capacidades. Como acabamos de ver.

Sin embargo, Minsky y su colega Seymour Papert identificaron y destacaron las limitaciones de los perceptrones en su libro "Perceptrons" publicado en 1969. En particular, demostraron que los perceptrones simples no podían realizar ciertas tareas fundamentales, como la función XOR [*exclusive or*, "o exclusivo"²⁷¹]. Aunque esta limitación²⁷² ya era conocida en ciertos círculos de investigación, la forma en que Minsky y Papert presentaron sus hallazgos, junto con su prominencia en el campo, **tuvo un impacto muy significativo**.

El libro, lamentablemente, tuvo un efecto no deseado en la comunidad de investigación de inteli-

²⁷⁰ Seamos honestos, según él $9+9+9$ es 26. Tiene sentido, cuanto más nos alejemos del trainset más error va a cometer.

²⁷¹ *Or exclusivo*: sólo es cierto si "1 xor 0" o "0 xor 1", el resto de casos es falso.

²⁷² Esta limitación no es más que la imposibilidad de un perceptrón de establecer un límite entre dos clases que no pueden ser separables mediante una línea.

gencia artificial. Fue interpretado por muchos [a veces de manera incorrecta] como una crítica a todas las formas de redes neuronales, no sólo a los perceptrones simples. Esto llevó a un descenso en la financiación y la investigación en este área durante una buena parte de la década de 1970 y 1980. ¿Te suena el "invierno de la IA"?

A pesar de esto, la investigación en redes neuronales no se detuvo completamente. Con el tiempo, se descubrió que al añadir más capas a las redes neuronales [**perceptrones multicapa**] y mediante el uso de técnicas como el algoritmo de retropropagación que acabamos de ver, se podían superar las limitaciones señaladas por Minsky y Papert. Esta actualización llevó a un renacimiento de las redes neuronales, principalmente en los años 90, que finalmente culminó en la explosión del aprendizaje profundo desde principios del siglo XXI.

17.3 Estructura básica de una red neuronal

Con una sólo neurona en realidad no podemos hacer mucho más. Para alcanzar su potencial, las neuronas artificiales deben ser **agrupadas en capas** [*layers*]. Cada capa consta de k neuronas, cada una de las cuales recibe las mismas entradas [x_1, x_2, \dots, x_n]. Fíjate en la Figura 91 en la capa oculta [*hidden*] de la red de la derecha [*feedforward neuronal network*].

Una **red neuronal** está compuesta por neuronas interconectadas y organizadas en capas. Aunque hay varios tipos, nos centraremos en las redes *feedforward*, aquellas en las que no hay ciclos²⁷³. La estructura básica de una red neuronal consta de tres tipos de capas [ver Figura 91]²⁷⁴:

- **Capa de entrada** [*input layer*]: Es la capa inicial de la red neuronal, encargada de recibir, y quizás adaptar, los datos de entrada. Cada neurona de esta capa está conectada a cada neurona de la siguiente capa mediante conexiones ponderadas [con los pesos].
- **Capas ocultas** [*hidden layers*]: Son capas intermedias entre la capa de entrada y la capa de salida. Estas capas realizan el procesamiento de la información a medida que se propaga a través de la red neuronal. Pueden existir una o varias capas ocultas, dependiendo de la complejidad del problema que se pretende resolver.
- **Capa de salida** [*output layer*]: Es la capa final de la red neuronal y proporciona los resultados o predicciones de la red. Cada neurona de esta capa representa una clase o una variable de salida y se activa en función de la información procesada en las capas anteriores.

Antes de este apartado parecía que la unidad básica de una red neuronal era la neurona artificial, pero no. Al escalar el tamaño de estas redes la unidad que manejamos es la capa. Por simplicidad. Cuando creamos una red neuronal, por ejemplo *feedforward*, lo que realmente hacemos es apilar capas, posiblemente de diferentes tipos. El tipo de capas que estamos viendo se denominan densas [*fully dense*].

273 Que no haya ciclos significa que la información fluye de un lado a otro, sin volver atrás en ningún momento. De no cumplirse esta condición hablaríamos de redes recurrentes.

274 Los sufijos "feedforward" y "deep" no son excluyentes. El primero nos indica que no hay ciclos y el segundo que hay más de una capa oculta. Así es posible decir "deep feedforward neural network". Frecuentemente omitimos ambos sufijos según el contexto.

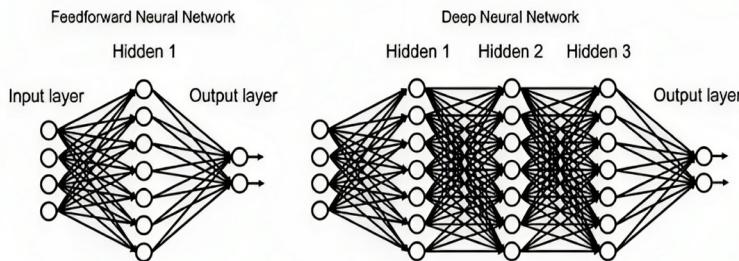


Figura 91: Esquema de redes neuronales artificiales multicapa densas (fully connected)

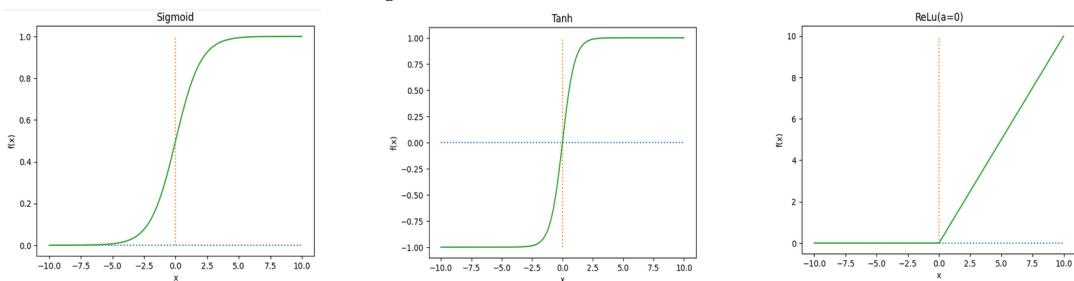
Fuente: "Artificial intelligence, machine learning, and deep learning for clinical outcome prediction". Rowland et al., 2021. Emerging Topics in Life Sciences. 5. 10.1042/ETLS20210246.

Ya tenemos todos los aspectos básicos de una red neuronal artificial, pero debemos volver un momento a las funciones de activación.

Funciones de activación revisadas

Como ya hemos dicho, las funciones de activación juegan un papel fundamental en el procesamiento de la información en una red neuronal²⁷⁵. Estas funciones se aplican a la suma ponderada de las entradas y los pesos de una neurona, determinando si se activa o no. Ya hemos nombrado algunas básicas, que apenas se usan, pero hay otras²⁷⁶.

Figura 92: Funciones de activación modernas



Las funciones de activación que se muestran en la Figura 92, son las versiones continuas de las mostradas anteriormente [$\text{sigmoid} \approx \text{step}$, $\text{tanh} \approx \text{sign}$ y relu es una versión parametrizada²⁷⁷ de linear]. Quizá asuste ver tanta función de activación y no comprender bien que función realiza dentro de una neurona artificial, pero debes saber que la mayor parte de las veces elegir una u otra es cuestión de probar²⁷⁸, al menos en capas ocultas. Por ejemplo, es habitual ver una *sigmoid* en la

275 Son las que aportan la no linealidad. Las funciones de activación no lineales permiten que los modelos captén relaciones y patrones complejos en los datos que no serían posibles con solo operaciones lineales.

276 Incluso más de las que muestro aquí, pero suelen ser variantes (Softmax, MaxOut, Leaky ReLu, PReLU, ELU, etc.). El arquitecto/a de redes neuronales incluso puede crear la suyas propias.

277 La imagen muestra ReLU con $a=0$, si a fuese 1 entonces sería igual a "linear".

278 De hecho se va probando con funciones de activación y comparando los modelos. Con experiencia se suele elegir más eficazmente una función de activación para cada capa oculta.

capa de salida si lo que quieras es obtener una o más probabilidades, ya que esta función de activación emite en el rango $[0, 1]$.

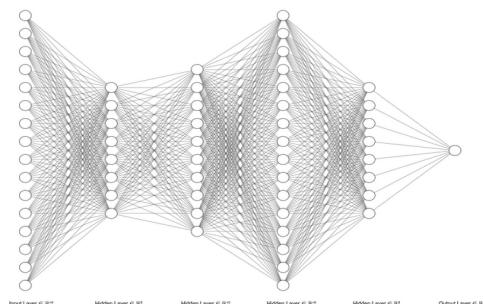
17.4 Arquitecturas de Redes Neuronales

Ya hemos visto las unidades de que están formadas las redes neuronales artificiales, sus unidades individuales, como se agrupan en una capa y como las capas se agrupan para formar redes. Ahora vamos a ver las arquitecturas que estas redes pueden formar:

El **perceptrón** es el modelo más básico de red neuronal. Consiste en una única capa de neuronas que hace las veces de entrada y salida. Cada neurona en esta capa comparte una serie de entradas y genera una salida utilizando una función de activación.

Las **redes neuronales multicapa**, también conocidas como perceptrones multicapa [MLP, por sus siglas en inglés] o deep feedforward neural network, son una extensión del perceptrón básico. Estas redes constan de una capa de entrada, una o más capas ocultas y una capa de salida. Cada neurona en una capa está conectada a todas las neuronas de la capa anterior y de la capa siguiente, formando una estructura en forma de grafo sin ciclos [feedforward]. Las capas ocultas permiten que la red neuronal aprenda características y patrones más complejos en los datos.

Tenemos un ejemplo simple de aplicación de MLP si usamos el *dataset cancer_breast*. Un MLP analiza esta información para determinar la naturaleza del tumor: maligno [1] o benigno [0]. El MLP opera de la siguiente manera: la capa de entrada recibe las 30 características numéricas, las capas ocultas [4 de tipo *dense*] procesan y identifican patrones clave en esos números, y la capa de salida genera una predicción, indicando si el tumor es probablemente maligno o benigno. A través del entrenamiento con ejemplos ya etiquetados, la red mejora su capacidad predictiva.



Por ejemplo, a la izquierda podemos ver la estructura del modelo usado, y en la Figura 93 podemos las curvas de precisión y aprendizaje de un MLP aplicada al *dataset cancer_breast*, y como finalmente la precisión [usando el *testset*²⁷⁹] fue del 97,37%. Como referencia, el mejor modelo²⁸⁰ del Capítulo 16, arrojó una precisión de 95,89%.

²⁷⁹ La gráfica representa un poco menos porque muestra la precisión respecto al conjunto de validación.

²⁸⁰ Fue un XGBoost, con los hiperparámetros optimizados.

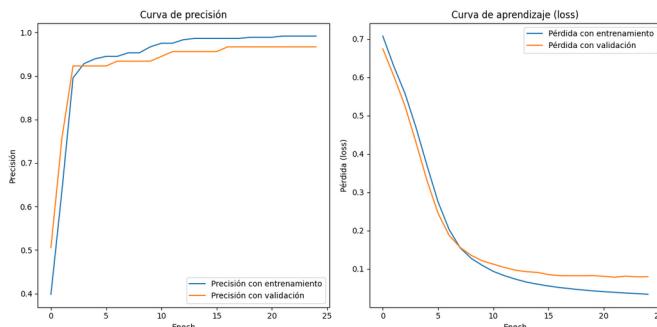


Figura 93: Curvas de aprendizaje para un MLP sobre el dataset "cancer_breast"

Redes neuronales convolucionales

Las redes neuronales convolucionales²⁸¹ [CNN, por sus siglas en inglés] son arquitecturas especializadas en el procesamiento de datos bidimensionales, especialmente imágenes. Las CNN utilizan filtros convolucionales para extraer características locales en la imagen y luego combinan estas características en capas posteriores para obtener una representación más global. Las redes convolucionales son ampliamente utilizadas en tareas de reconocimiento visual, detección de objetos y segmentación de imágenes.

Mi IA favorita dice: *Imagina que tienes una gran foto y un pequeño marco o ventana transparente [...]. Este pequeño marco es lo que llamamos "filtro". Al deslizar o mover esta ventana por toda la foto y mirar a través de ella, puedes cambiar o resaltar ciertas partes de la imagen. Así, si quieras encontrar todos los bordes o ciertas texturas en la foto, usarías un tipo específico de ventana (filtro) para hacerlo. Este proceso de mover la ventana y obtener una nueva imagen que muestra lo que estás buscando se llama "convolución", y es la idea detrás de los filtros convolucionales en las imágenes. ¡Es como un juego de buscar detalles con una lupa!*

Un ejemplo de aplicación de redes convolucionales es la clasificación de imágenes. La red neuronal convolucional tomaría una imagen como entrada y aplicaría una serie de filtros convolucionales para detectar características como bordes, texturas y formas en la imagen. Estas características se combinan en capas posteriores para realizar la clasificación final de la imagen en una de las clases predefinidas, como perro, gato, pájaro, etc.

Vamos a entrenar un clasificador multiclasa con una red neuronal convolucional para clasificar el dataset *MNIST digits* y comparar los resultados con los del apartado 16.6, el código es el que sigue:

281 Las hemos usado con profusión en la sección primera: "Visión artificial".

1. Lleva a cabo las importaciones necesarias en el resto del código.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import tensorflow as tf
4
5 from tensorflow.keras.datasets import mnist, fashion_mnist
6 from tensorflow.keras.models import Sequential
7 from tensorflow.keras.layers import Dense, Conv2D, MaxPooling2D, Flatten

```

2. Lee el *dataset* que usarás para entrenar y evaluar el modelo. Si fuese necesario haz la limpieza y las adaptaciones adecuadas de los datos. Posteriormente²⁸² divide el *dataset* en *trainset* y *testset*.

```

1 (train_images, train_labels), (test_images, test_labels) = mnist.load_data()
2
3 # Normalizando258 las imágenes para que los valores estén en el rango [0, 1]
4 train_images = train_images.reshape((60000, 28, 28, 1)).astype('float32') / 255
5 test_images = test_images.reshape((10000, 28, 28, 1)).astype('float32') / 255
6
7 # Convertir los labels258 a one-hot encoding
8 train_labels = tf.keras.utils.to_categorical(train_labels)
9 test_labels = tf.keras.utils.to_categorical(test_labels)

```

3. Declara la estructura de la red neuronal: Creamos un modelo secuencias [que apila capas] y vamos añadiendo capa tras capa. Como podemos ver en este tipo de redes hay varios tipos diferentes de capas. Las *Conv2D*, *MaxPooling2D* y *Flatten* son típicas de las CNN.

```

1 model = Sequential()
2 model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
3 model.add(MaxPooling2D((2, 2)))
4 model.add(Conv2D(64, (3, 3), activation='relu'))
5 model.add(MaxPooling2D((2, 2)))
6 model.add(Conv2D(64, (3, 3), activation='relu'))
7 model.add(Flatten())
8 model.add(Dense(64, activation='relu'))
9 model.add(Dense(10, activation='softmax'))

```

También podemos ver que usamos la función de activación ReLU, y otra llamada *softmax*. Esta última convierte la salida de la red neuronal en un conjunto de probabilidades que juntas suman 1. Sin esta función de activación en la capa de salida, esta CNN emitiría 10 números, siendo el más grande el de la clase con más peso, y así el resto.

4. Compila en modelo creado. Es durante este proceso cuando se crea el modelo en realidad, iniciando sus pesos con valores aleatorios. Al tiempo elegimos el optimizador²⁸³, la función de pérdida²⁸⁴ y las métricas que usaremos para medir qué tal va el aprendizaje.

282 *Este código se sale de la norma. Ya nos entregan el dataset dividido, así que no lo hacemos “posteriormente”, lo hacemos “anteriormente”.*

283 *El optimizador es el encargado de “optimizar” los pesos para minimizar el error. Es el duende detrás de la magia de este tipo de aprendizaje. Por cierto, Adam significa: Adaptive Moment Estimation.*

284 *De acuerdo, “categorical_crossentropy” no es un nombre atractivo ni es fácil de entender. Básicamente se calcula tomando el valor negativo del logaritmo de la probabilidad predicha para la clase verdadera. Quédate con que es adecuado para clasificación multiclas, como es el caso.*

```

1 model.compile(optimizer='adam',
2                 loss='categorical_crossentropy',
3                 metrics=['accuracy'])

```

5. Finalmente entrena el modelo: hacemos un *fit* con los datos de entrenamiento [*train_images* y *train_labels*, ambos forman el *trainset*], indicamos que queremos 10 *epochs* y cual va a ser el conjunto que usaremos para validar el aprendizaje [el *testset*].

```

1 history = model.fit(train_images,
2                       train_labels,
3                       epochs=10,
4                       batch_size=64,
5                       validation_data=(test_images, test_labels))

```

¿Y cual es el resultado? Tras 10 *epochs* tenemos una precisión media del 99.13%.

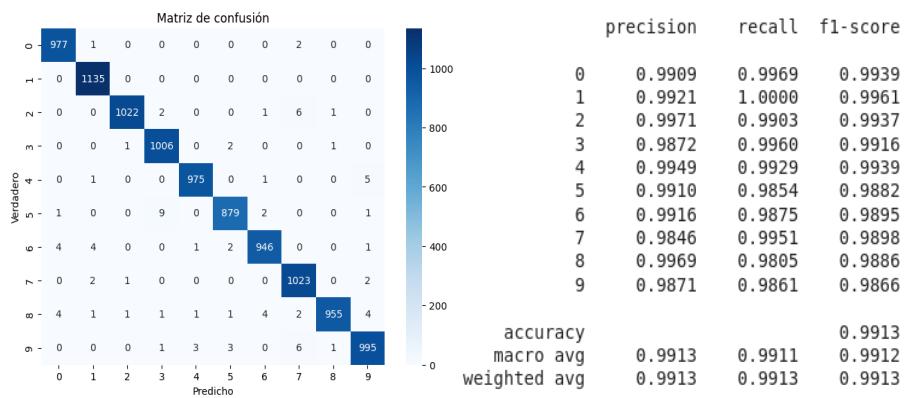


Figura 94: Matriz de confusión multiclase 10x10 y métricas con CNN

En el Capítulo 16 conseguimos un 98% [ver la Figura 79]. En ese mismo capítulo mostramos como el *dataset* [su calidad y cantidad] era determinante para obtener un buen modelo. Habíamos sometido al clasificador multiclase ganador a un *dataset* idéntico al *MNIST digits*, pero de menos calidad: el *MNIST fashion*. El resultado fue de una precisión media de 89.83%. Usando una arquitectura CNN obtuvimos 91,80% como máximo y en 6 *epochs*.

Redes neuronales recurrentes

Las redes neuronales recurrentes [RNN, por sus siglas en inglés] son arquitecturas diseñadas para modelar secuencias de datos, como frases [palabras, *tokens*] o series temporales. A diferencia de las redes neuronales tradicionales, las RNN tienen conexiones recurrentes, lo que significa que la información puede retroalimentarse en la red. No son por tanto *feedforward*. Esta propiedad permite que las RNN tengan memoria y capturen dependencias a largo plazo en los datos secuenciales.

Un ejemplo de aplicación de redes recurrentes es la generación de texto. Una RNN puede entrenarse con un conjunto de textos y luego generar nuevos textos basados en el patrón de los datos de

entrenamiento. Por ejemplo, se puede entrenar una RNN en textos de *Shakespeare* y luego generar nuevos textos que imiten el estilo y la estructura del autor.

Nota: Lo que sigue no son arquitecturas propiamente dichas, ya que se basa en una clasificación que se refiere a cómo se utiliza el modelo y cuál es su objetivo principal, no a la arquitectura subyacente específica, que puede ser suma de cualquiera de las anteriores.

Redes neuronales generativas

Las redes generativas, como las **Redes Generativas Antagónicas** [GANs] o los **Autoencoders Variacionales** [VAEs], son arquitecturas diseñadas para modelar y generar datos que imitan la distribución de un conjunto de datos de entrenamiento. A través del aprendizaje no supervisado o semisupervisado, estas redes capturan características intrínsecas y patrones subyacentes en los datos, permitiendo la creación de nuevas instancias que son estadísticamente similares a los datos originales.

Por ejemplo, una GAN podría ser entrenada en un conjunto de datos de imágenes de rostros humanos y luego ser capaz de generar imágenes de caras que, aunque no son réplicas de las caras en el conjunto de entrenamiento, comparten las mismas características y variabilidades. Siendo más generalista, imagina que tu conjunto de datos es un conjunto de puntos en un espacio bidimensional [fotos de caras]; una red generativa aprendería la *forma* general de esa nube de puntos y podría generar nuevos puntos que encajen de forma coherente dentro de esa forma aprendida [esto es, nuevas caras, ver Figura 45].

Mi IA favorita dice: GAN (Generative Adversarial Networks): Compuestas por dos redes, una generativa y otra discriminadora, que trabajan en conjunto para generar datos que sean indistinguibles de los datos reales.

Redes autoencoder [o encoder/decoder]

Las redes autoencoder son diseñadas para aprender una representación compacta y eficiente de los datos de entrada, generalmente para tareas como la reducción de dimensionalidad o el preentrenamiento de una red neuronal. Estas redes constan de dos partes principales: el codificador, que comprime la entrada a un espacio latente más pequeño, y el decodificador, que reconstruye la entrada a partir de esta representación compacta. El objetivo del entrenamiento es minimizar la diferencia entre la entrada original y su reconstrucción, lo que lleva a que la red aprenda las características más importantes de los datos.

Por ejemplo, volvamos con el *dataset* MNIST *digits*. Queremos reducir la dimensionalidad de cada imagen manteniendo la mayor cantidad de información posible. Podríamos emplear un *autoencoder* que tome las imágenes como entrada, las comprima en una representación latente de menor dimensionalidad mediante el codificador, y luego las reconstruya mediante el decodificador. Al entrenar el *autoencoder* para minimizar la diferencia entre las imágenes originales y las reconstruidas, podemos extraer una representación compacta de cada imagen que capture sus características más importantes. Esta representación puede utilizarse posteriormente para tareas como la clasifi-

cación de dígitos, la detección de anomalías o incluso la generación de nuevos dígitos similares a los del conjunto de entrenamiento.

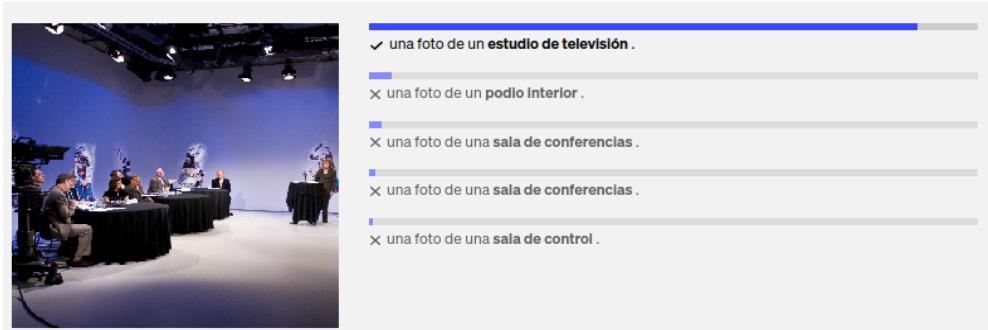


Figura 95: Ejemplo de predicción de CLIP

Fuente: OpenAI - <https://openai.com/research/clip>

Redes basadas en el concepto de atención

Las redes basadas en el concepto de atención [§12.1], particularmente los **Transformers**, han revolucionado el campo del aprendizaje profundo y son especialmente poderosas en tareas de procesamiento del lenguaje natural, aunque no de forma exclusiva. Estas arquitecturas eliminan la necesidad de recurrencia [redes recurrentes] y se centran en mecanismos de atención para capturar las dependencias en los datos. A través de la atención ponderada, el modelo puede centrarse en diferentes partes de la entrada para realizar tareas como traducción, generación de texto, y clasificación de manera más efectiva y eficiente.

Pongamos como ejemplo el modelo BERT [*Bidirectional Encoder Representations from Transformers*] es un *Transformer* preentrenado que se ha utilizado con éxito en una variedad de tareas de procesamiento del lenguaje natural, como clasificación de texto, generación de resúmenes y respuesta a preguntas. BERT se entrena para predecir palabras faltantes en una oración, capturando así el contexto bidireccional que rodea a cada palabra.

En resumen, el mecanismo de atención [centrándonos en *transformers*] es parecido a un sistema para decidir qué palabras en un conjunto de datos son especialmente relevantes para las palabras en otro conjunto de datos, y luego usar esa información para generar una salida. Es una forma de permitir que el modelo se "enfoque" en la información más importante en un momento dado.

Estas son las arquitecturas o categorías más usadas o que más impacto ha tenido en el campo que nos ocupa. Pero no es lo único, más allá de las arquitecturas, se están llevando a cabo diseños mucho más complejos que combinan partes de las aquí expuestas. Llamémoslas **meta arquitecturas**.

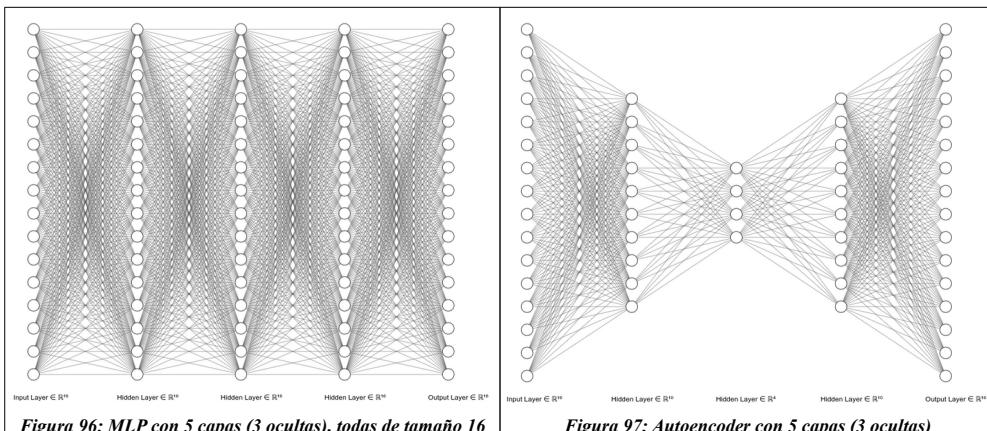
Por ejemplo *DALL-E 3*, *Stable Diffusion*, *FLUX* o *Midjourney*, que a partir de un texto crean una imagen que cuadra con este; o [Open]CLIP [Figura 95, y ver más adelante] que a partir de una imagen proporciona las probabilidades para un conjunto de textos descriptivos dados.

17.5 Ejemplos de redes neuronales artificiales

A lo largo de este capítulo, y en capítulos anteriores, hemos hecho uso de redes neuronales artificiales, mostrando sólo sus resultados. Ahora vamos a ver cómo funcionan desde el punto de vista del código y comparar sus resultados con modelos anteriores; si es posible, dado que algunas de las arquitecturas que veremos es muy difícil entrenarlas en los ordenadores que habitualmente tenemos a nuestro alcance, incluida la infraestructura de Google Colab.

17.5.1 Autoencoders

Observa el MLP de la Figura 96, en él podemos observar que [1] la entrada tiene el mismo tamaño que la salida [16], y que las tres capas ocultas son del mismo tamaño. Imagina ahora que la entrenamos con un *dataset* de imágenes, a la entrada²⁸⁵ le presentamos una imagen y a la salida esperamos que se muestre la misma imagen; el aprendizaje es pues que muestre a la salida lo mismo que le presentamos a la entrada.



En este tipo de redes, durante el entrenamiento, los pesos se distribuirán posiblemente igual en las tres capas ocultas; en cualquier caso este tipo de diseño no es muy útil, pero es un punto de partida para explicar los autocodificadores.

Observa ahora la Figura 97, la idea es la misma que en el diseño anterior, al igual que su *dataset* y entrenamiento con los que - imaginativamente - la estamos entrenando. Pero ahora hay un cambio: En este nuevo diseño hemos *estrechado* la capa central y ahora entre las capas de entrada y salida hay una especie de "cuello de botella". A la parte de la izquierda la llamamos **codificador** [encoder, de la capa de entrada a la más "estrecha", esta incluida] y a la de la derecha **decodificador** [decoder, desde la capa más "estrecha", no incluida, a capa de salida], ver la Figura 98.

²⁸⁵ He propuesto una MLP de 16 entradas por motivos de claridad al imprimir la imagen. Insuficiente para imágenes. Imagina en este ejemplo que se trata de 784 (28x28) entradas e iguales salidas.

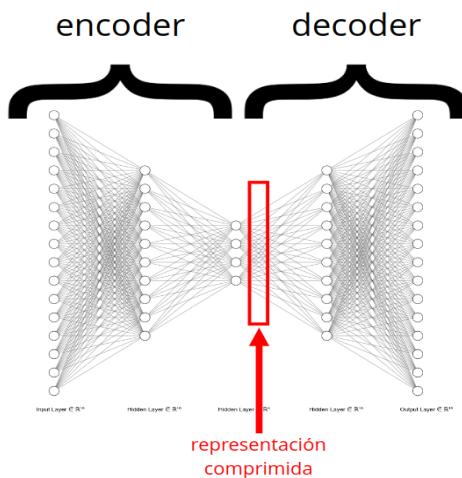


Figura 98: Esquema de un autoencoder

Seguimos imaginando que entrenamos este modelo alimentándolo con imágenes a la entrada y las mismas imágenes a la salida. Ahora la red, al hacer el esfuerzo de aprender que a partir de una entrada debe emitir la misma a la salida, necesita comprimir la información de la foto, quedándose sólo con las **características principales**: una representación comprimida de la imagen de la entrada.

Vamos a hacer un ejemplo real, con un *dataset* conocido. Nuestra arquitectura va a tener 784 entradas e igual número de salidas y cinco capas ocultas de tipo *dense*²⁸⁶ [128, 64, 32, 64, 128]; la salida de la capa con 32 neuronas es la representación comprimida de la entrada, conteniendo sus características principales.

Caben dos preguntas: Dada la entrada, ¿cómo se ve la salida? pues así:

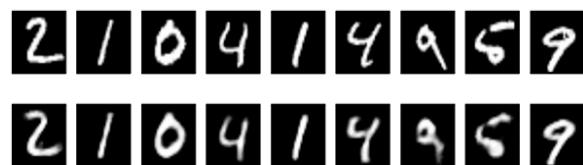


Figura 99: Entrada (arriba) y salida (abajo) de un autoencoder simple con el dataset MNIST digits

Como podemos ver, la red ha hecho su trabajo, y ha aprendido, pese al "cuello de botella", a propagar la entrada a la salida de forma bastante eficiente.

²⁸⁶ Todas *ReLU*, la salida es sigmoid.

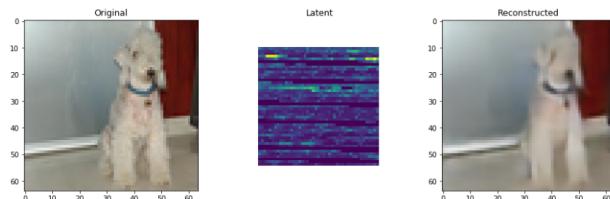


Figura 100: Autoencoder entrenado con imágenes (64x64) de perros.

Imagen original (izquierda), reconstruida (derecha) y el estado latente o representación comprimida (8x8).

Fuente: Chris Deotte - <https://www.kaggle.com/code/cdeotte/dog-autoencoder/notebook>

Y aquí viene la segunda pregunta: si la salida es tan parecida a la entrada ¿podemos decir que la red aprende a reconstruir la salida a partir de la representación comprimida? La respuesta es sí. Y esa es la magia de este tipo de arquitectura; entrenamos un modelo y a cambio obtenemos dos²⁸⁷:

1. El **codificador**, que a partir de una entrada del *dataset* es capaz de reducirlo a una representación latente o comprimida, un vector de características, que lo identifica de forma única dentro del *dataset*.
2. El **decodificador**, que a partir de una representación latente es capaz de reconstruir la información original. Esta representación latente no tiene porque formar parte del *dataset* original.

Aunque hemos puesto como ejemplo un *autoencoder* entrenado con imágenes, hay que recordar que para estas máquinas todo está formado por números: texto, imágenes, datos tabulados, etc. Así que le resulta indiferente²⁸⁸ el tipo de información que procesa.

Estas dos herramientas son la base de la IA generativa [que merece un libro propio] y de muchas técnicas de arquitecturas más complejas. Por ejemplo: los *transformers* son, dentro de su complejidad, un codificador y un decodificador, como ya hemos visto [§12.1.2].

Por cierto, a este tipo de aprendizaje se le denomina **aprendizaje autosupervisado**, ya que es el propio algoritmo el que extrae de los datos las etiquetas que necesita para aprender. Se le considera un caso particular de aprendizaje supervisado.

17.5.2 Entrenando una arquitectura generativa: GAN

Vamos a ver una de las arquitecturas más apasionantes, y *madre* de muchas que han venido después. Una **Red Generativa Adversaria** (GAN, por sus siglas en inglés) consiste en dos redes neuronales, el generador y el discriminador, que se entranan conjuntamente. El generador intenta producir datos falsos que parezcan reales, mientras que el discriminador intenta distinguir entre datos reales y datos falsos. Esencialmente, es un juego de gato y ratón donde el generador es el "falsifi-

287 Si, es posible "cortar" una red neuronal artificial en partes como si usáramos una tijera. Recordemos que la unidad de esta es la capa, así que podemos eliminar, añadir modificar capas en estas y otras arquitecturas, incluso después de haber sido entranadas.

288 Una matización: aunque son agnósticas respecto al tipo de datos hay arquitecturas que son preferibles según si hablamos de imágenes (redes convolucionales), texto (transformers), etc. Los autoencoders que estamos viendo aquí son los más generales.

cador" y el discriminador es el "policía".

Otra analogía: el profesor y el alumno. Un profesor le pone un examen de una pregunta al alumno, siempre del mismo tema pero no siempre la misma. El alumno intenta aprobar el examen diciendo lo que se le ocurre, siempre de forma aleatoria. No ha estudiado nada. El profesor sólo corrige al alumno diciendo bien o mal, nada más. El alumno, tras muchos intentos, acaba aprendiendo qué decir para que el profesor no distinga su respuesta [aleatoria] y la de por buena.

Supongamos que trabajamos con imágenes. Esta arquitectura debe por tanto entrenar dos MLP: [1] la discriminadora, entrenada con un gran conjunto de imágenes para que haga clasificación binaria [si/no] de estas. Si, por ejemplo, hablamos de rostros el *dataset* estará compuesto por n imágenes con rostros y otras n con imágenes que no son rostros. La discriminadora aprende a diferenciar imágenes de un tipo u otro. [2] la generadora es un MLP de tipo *decoder*, una de las partes de una red *autoencoder*. Parte de una entrada aleatoria y como salida genera una imagen. Simple.

La clave es entrenar el MLP generador: Usando entradas aleatorias, dejamos que produzca imágenes y que la RNA discriminadora indique si es o no correcta. Si es incorrecta hacemos que esta aprenda, para que la próxima vez genere una solución menos incorrecta. Tras docenas de miles de *epochs* [o muchos más], tenemos un MLP generador que a partir de entradas aleatorias genera imágenes de rostros. Como los MLP tienden a generalizar, probablemente estos rostros serán originales, no existirán.



Figura 101: Rostros sintéticos producidos por StyleGAN

Fuente: NVIDIA research

La idea de enfrentar dos algoritmos entre sí provino de Arthur Samuel, un destacado investigador en el campo de la informática a quien se le atribuye haber popularizado el término *aprendizaje automático*. Mientras estaba en IBM, ideó un juego de damas que fue uno de los primeros en aprender por sí mismo con éxito, en parte al estimar las posibilidades de victoria de cada jugador en una configuración determinada del juego [recordemos, algoritmo *minimax*]. Más tarde, Ian Goodfellow [et al.] en un artículo de investigación fundamental de 2014 titulado simplemente "Redes generativas adversarias", describen la primera implementación funcional de un modelo generativo basado

en redes adversarias, el que acabamos de describir.

Las GAN son muy difíciles de entrenar, por la gran necesidad de potencia de cálculo y por la inestabilidad de su salida. Para imágenes de 1024x1024 [hoy en día no muy grandes] y con una GPU de más de cuatro mil euros, necesitaríamos más de 41 días de entrenamiento, unos 60k minutos.



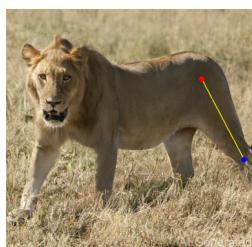
Vídeo 14: Synthesizing High-Resolution Images with StyleGAN2

Fuente: NVIDIA Developer

StyleGAN es una arquitectura de red neuronal desarrollada por NVIDIA que genera imágenes artificiales de alta calidad utilizando GANs. Se utiliza principalmente para crear rostros, objetos y escenarios realistas pero sintéticos. StyleGAN se basa en la estructura de las GANs, que, como sabemos, consta de dos redes, el generador y el discriminador. A través de esta competencia, el generador mejora su capacidad para producir imágenes realistas. A diferencia de las GANs tradicionales donde se introduce un vector de ruido al comienzo, en StyleGAN, el vector de ruido se transforma en un "vector de estilo" que se introduce en múltiples etapas del generador. Esto permite controlar características específicas en diferentes niveles de detalle [por ejemplo, rasgos faciales gruesos versus detalles finos como el estilo del cabello].



Imagen original



Puntos de control



Resultado



AI Generated



AI Generated



AI Generated

Observar en el resultado, cuya imagen original es un rostro sonriente, como la expresión sería afectó también a los músculos de la cara.

17.5.3 Texto a imágenes

Dall-E [2021], Midjourney [2022], Stable Diffusion [2022] y otros muchos, aparecieron hace poco en la red como un avance de la IA generativa, con la capacidad de producir una imagen (ahora videos) a partir de un texto introducido como entrada (*prompt*). ¿Cómo lo hacen? Los fundamentos para que funcionen los hemos visto a lo largo del libro, especialmente MLP, estados latentes (embeddings, vectores de características) y autoencoders.

Pero la pócima a veces no es revelada. De todas formas sí contamos con la arquitectura de DALL-E.

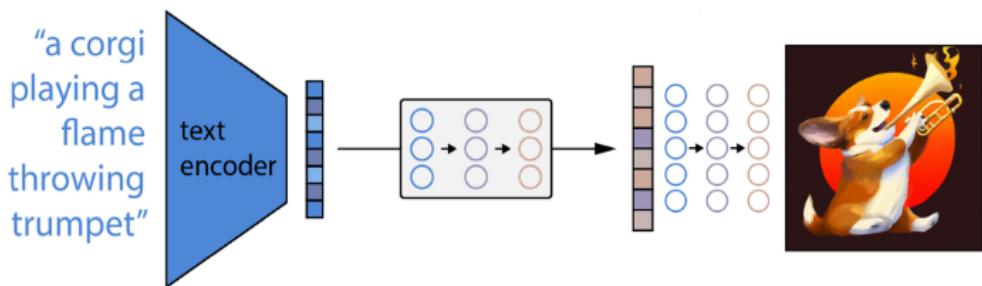


Figura 102: Esquema de la arquitectura de DALL-E 2

Fuente: "Hierarchical Text-Conditional Image Generation with CLIP Latents"; A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen; 2022, arXiv.

A vista de pájaro, DALL-E funciona de forma muy sencilla [Figura 102]:

1. Se introduce un mensaje de texto (*prompt*) en un codificador (*encoder*) que está entrenado para transformar el mensaje a un espacio de estados latentes (*embeddings*).
2. A continuación, un modelo (MLP) llamado *prior* convierte el *embedding* de texto a una codificación de imagen, la cual captura la información semántica del mensaje contenido en la codificación de texto.
3. Finalmente, un decodificador (*decoder*) de imágenes genera estocásticamente una imagen²⁸⁹ que es una representación visual de la información semántica que contiene el *prompt*.



Video 15: How does DALL-E 2 actually work?

Fuente: AssemblyAI

289 Similar a como lo hace el generador de GANs.

DALL-E depende de CLIP [Figura 95], de OpenAI. CLIP se entrena con cientos de millones de imágenes y sus títulos asociados, y aprende en qué medida se relaciona un fragmento de texto determinado con una imagen. Es decir, en lugar de intentar predecir un título dado una imagen, CLIP simplemente aprende qué tan relacionado [en %] está un título determinado con una imagen concreta. Todo el modelo DALL-E 2 depende de la capacidad de CLIP para aprender la semántica del lenguaje natural.

DALL-E también depende de otro modelo anterior, GLIDE [Figura 103], para generar imágenes. Este modelo aprende a invertir el proceso de codificación de imágenes para decodificar estocásticamente incrustaciones de imágenes CLIP.

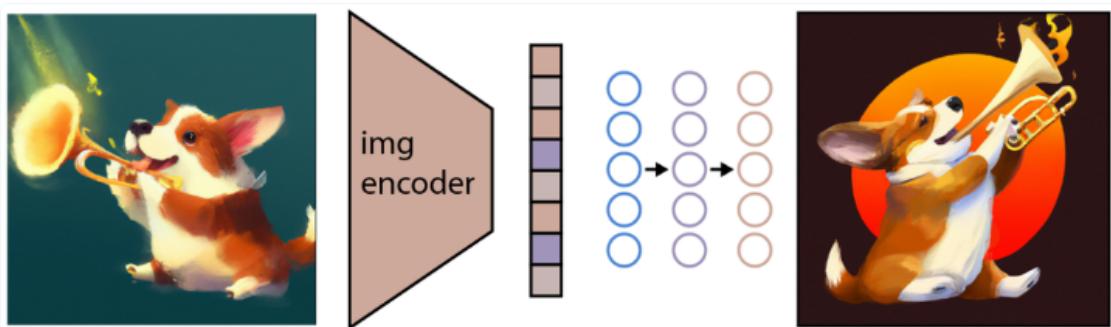


Figura 103: Arquitectura GLIDE para transformar una imagen

Fuente: "Hierarchical Text-Conditional Image Generation with CLIP Latents"; A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen; 2022, arXiv.

Como se muestra en la imagen de arriba, cabe señalar que el objetivo no es construir un codificador automático y reconstruir exactamente una imagen dada su *embedded*, sino generar una imagen que mantenga las características principales de la imagen original dada su *embedded*. Para realizar esta generación de imágenes, GLIDE utiliza un **modelo de difusión**.

Imagina que tienes una imagen clara y, poco a poco, la "corrompes" con ruido hasta que es casi irreconocible. Luego, utilizando un modelo entrenado, intentas revertir este proceso paso a paso, eliminando el ruido en cada etapa hasta que vuelvas a tener una imagen original. Este proceso de revertir el ruido es esencialmente cómo funcionan los modelos de difusión.

Si el modelo de difusión se *corta a la mitad* después del entrenamiento, se puede utilizar para generar una imagen muestreando aleatoriamente ruido gaussiano y luego eliminándolo para generar una imagen fotorrealista [**difusión inversa**].

Ya tenemos todo, vamos a juntarlo de nuevo:

1. El codificador de texto CLIP asigna la descripción de la imagen al espacio de representación.
2. Luego, el modelo de difusión *prior* se asigna desde la codificación de texto CLIP a la codificación de imagen correspondiente.
3. Finalmente, el modelo de generación GLIDE modificado se asigna desde el espacio de re-

presentación al espacio de la imagen mediante difusión inversa, generando una de muchas imágenes posibles que transmite la información semántica dentro del título de entrada.

RETOS DEL CAPITULO 17

1. Dialoga con tu IA favorita, cuál es la diferencia entre una neurona biológica y una artificial.
2. Busca en la web, qué aplicaciones del mundo real utilizan redes neuronales artificiales.
3. Busca en la web, cómo se inspiran las redes neuronales artificiales en el cerebro humano.
4. Explica en qué consiste el proceso de entrenamiento de una red neuronal. Intenta llegar al mayor grado de entendimiento que puedas. Apóyate en una IA para ello.
5. Investiga y describe qué son las capas ocultas en una red neuronal.
6. ¿Qué significa la "propagación hacia atrás" en el contexto de las redes neuronales?
7. ¿Cómo se puede prevenir el sobreajuste en una red neuronal? Intenta deducirlo.
8. Busca y describe más arquitecturas de redes neuronales artificiales, además de las explicadas en el texto.
9. ¿Qué son los hiperparámetros en una red neuronal y cómo afectan a su desempeño?
10. Investiga y describe ejemplos de bibliotecas de Python que permiten implementar redes neuronales desde cero.
11. ¿Qué es la transferencia de aprendizaje y cómo se aplica en redes neuronales?
12. Investiga y describe un caso de estudio en el que las redes neuronales han tenido un impacto significativo.
13. ¿Qué son las redes neuronales recurrentes y en qué aplicaciones se utilizan?
14. ¿Cómo se puede explicar el funcionamiento de una red neuronal a alguien que no tiene conocimientos técnicos?
15. Investiga y comparte ejemplos de herramientas de software que permiten visualizar el entrenamiento de redes neuronales.
16. ¿Cómo se pueden usar las redes neuronales para el procesamiento de lenguaje natural? ¿y para la visión artificial?
17. Investiga y comparte ejemplos de recursos en línea que permiten a los principiantes aprender sobre redes neuronales.
18. Las redes GAN son muy interesantes. Busca más información sobre ellas.
19. Busca en la web ejemplos de lo que son capaces de hacer las redes basadas en modelos de difusión.
20. Tema de debate: ¿podríamos llegar a tener una IA fuerte en breve?

Capítulo 18

LA ÉTICA EN LA ERA DEL APRENDIZAJE AUTOMÁTICO

18.1 Introducción

Un estudio publicado en 2018 por Joy Buolamwini y Timnit Gebru [ver Figura 104] mostró que el sistema de reconocimiento facial de IBM poseía un sesgo en la identificación de mujeres de color en comparación con hombres blancos. Sencillamente era mucho más probable que detectase correctamente las facciones de un hombre blanco que de una mujer afroamericana.



Figura 104: Captura del encabezamiento del artículo

“Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification”; J. Buolamwini, T. Gebru; Proceedings of Machine Learning Research 81:1–15, 2018

El artículo²⁹⁰ explica en su resumen (cito literalmente el *abstract*):

Estudios recientes demuestran que los algoritmos de aprendizaje automático pueden discriminar según clases como la raza y el sexo. En este trabajo, presentamos un enfoque para evaluar el sesgo presente en los algoritmos y conjuntos de datos automatizados de análisis facial con respecto a los

²⁹⁰ Es de obligado visionado el documental titulado “Sesgo codificado”, disponible en una de las plataformas de streaming más popular, en donde se incluye el contenido de este artículo como eje central del mismo.

subgrupos fenotípicos. [...] Se observa que estos conjuntos de datos están compuestos en su gran mayoría por sujetos de piel clara [...] e introducimos un nuevo conjunto de datos de análisis facial equilibrado por sexo y tipo de piel.

Evaluamos 3 sistemas comerciales de clasificación por género utilizando nuestro conjunto de datos y demostramos que las mujeres de piel más oscura son el grupo más mal clasificado (con tasas de error de hasta el 34,7%). La tasa de error máxima para los varones de piel clara es del 0,8%. Las disparidades sustanciales en la precisión de la clasificación de las mujeres más oscuras, las mujeres más claras, los hombres más oscuros y los hombres más claros en los sistemas de clasificación de género requieren una atención urgente si las empresas comerciales quieren construir algoritmos de análisis facial algoritmos de análisis facial.

En algunos estados de los EE.UU. los jueces cuentan con un asistente que les aconseja a la hora de calcular cuánto dura la pena de prisión que imponen a los condenados por un determinado delito. Este asistente no es humano: se trata de COMPAS, un algoritmo que utiliza diversas variables sobre el pasado de cada reo para estimar las posibilidades que tiene éste de reincidir en el futuro y por tanto evaluar el riesgo de reincidencia criminal. En 2016, ProPublica publicó un informe que mostraba que este algoritmo, desarrollado por la empresa Northpointe, tenía un sesgo racial. El algoritmo asignaba tasas de riesgo más altas a los afroamericanos que a los blancos, incluso cuando los antecedentes criminales eran similares. El informe llevó a cuestionarse el uso de algoritmos en el sistema de justicia penal.

En 2019, YouTube, propiedad de Google, fue criticado por su algoritmo de moderación de contenido, que eliminó o "desmonetizó" videos relacionados con temas LGBTQ+. Varios creadores de contenido, como Lindsay Amer y Chase Ross, denunciaron que sus videos fueron anulados o marcados como "no aptos para todos los públicos" sin una razón clara, lo que llevó a acusaciones de discriminación.

Todos estos casos tienen dos factores en común: [1] todos usaron técnicas de aprendizaje automático para la toma de una decisión y [2] no fueron justos ya que no aplicaron principios éticos en el entrenamiento de sus modelos.

Mi IA favorita dice: *La ética es una rama de la filosofía que se ocupa del estudio del comportamiento moral de los seres humanos, la distinción entre el bien y el mal, y las reglas o principios que rigen la conducta. Busca establecer criterios que permitan determinar qué acciones son moralmente correctas o incorrectas, y puede aplicarse a una variedad de contextos, como la ética profesional, la ética médica o la ética ambiental. La ética también examina las razones subyacentes para las acciones y las decisiones, así como las virtudes y principios que guían la vida humana.*

18.2 ¿Por qué hablar de ética aplicada a la inteligencia artificial?

El tema de la ética es primordial para asegurar que estas tecnologías inteligentes se utilicen de manera responsable y beneficien a la sociedad en su conjunto. Abordar sesgos en los datos, promover la transparencia y la responsabilidad, proteger la privacidad y considerar las implicaciones sociales son aspectos esenciales para un desarrollo ético y sostenible del aprendizaje automático.

A medida que estas tecnologías se vuelven más ubicuas en nuestra sociedad, es necesario reflexionar sobre su impacto y asegurarnos de que se utilicen de manera responsable y ética. En este capítulo detallaremos algunas de las principales preocupaciones éticas riesgos y consideraciones, en el aprendizaje automático y proporcionando ejemplos para ilustrar estas cuestiones.

¿Qué riesgos debemos afrontar?

Los modelos entrenados, por sus características intrínsecas de cómo se construyen, provocan principalmente las siguientes consecuencias, con sus riesgos asociados:

- Recolectan grandes cantidades de información [**riesgos de privacidad y seguridad**].
- Pueden causar un daño al predecir que alguien puede hacer o dejar de hacer algo [**riesgos de sesgos algorítmicos y discriminación**].
- En la mayoría de los casos el modelo no puede justificar su predicción [**riesgos asociados la falta de transparencia y responsabilidad**].

Además, también debemos tener en cuenta las **implicaciones sociales** del aprendizaje automático. El despliegue de modelos entrenados mediante estos sistemas puede provocar una serie de impactos significativos en la sociedad. Por ejemplo, la automatización del trabajo puede llevar a la pérdida de empleo en ciertos sectores, lo que plantea desafíos económicos y sociales. También puede haber implicaciones en la forma en que se utilizan los sistemas de reconocimiento facial, como su uso en la vigilancia masiva o la violación de la privacidad. Es importante considerar cuidadosamente estos impactos sociales y garantizar que el desarrollo y la implementación del aprendizaje automático se realicen de manera responsable y en beneficio de la sociedad en su conjunto.

18.3 Riesgos y desafíos éticos del Aprendizaje Automático

Durante el resto de este apartado, vamos a analizar y desarrollar conceptos y contextos asociados con los riesgos y desafíos éticos de aprendizaje automático en particular y a la inteligencia artificial en general.

18.3.1 Sesgos algorítmicos y discriminación

Los modelos creados a partir de algoritmos de aprendizaje automático aprenden de los datos de entrenamiento, y si estos datos contienen sesgos, la red puede aprender de ellos y perpetuarlos. Por ejemplo, como hemos explicado al principio de este capítulo, si se entrena un modelo de reco-

nocimiento facial utilizando principalmente datos de personas de raza blanca, es probable que el modelo tenga dificultades para reconocer y clasificar correctamente a personas de otras razas, lo que puede resultar en discriminación y marginación.

Generalizando, dado un modelo predictivo, que lleva a cabo su función después de haber sido entrenado con un conjunto de entrada [*trainset*], la exactitud es el cociente entre el número de predicciones correctas respecto al número total de precisiones [recordar el Capítulo 16 y el *dataset* usado para mostrar la clasificación binaria]. Imaginemos un modelo de predicción que trata de averiguar si un tumor es maligno a no a partir de los datos de entrada [el *dataset cancer_breast*]. A partir del modelo entrando y sometiéndole a un nuevo conjunto de datos [*testset*] podemos obtener resultados que podemos clasificar dentro de alguna de las siguientes cuatro categorías [ver Figura 105]: verdaderos positivos [*true positive*], falsos positivos [*false positive*], falsos negativos [*false negative*] y verdaderos negativos [*true negative*].

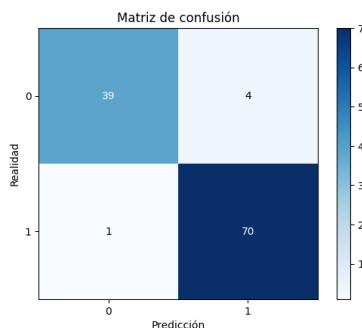


Figura 105: Matriz de confusión del modelo creado con el algoritmo SVM sobre el dataset *cancer_breast*

Como vimos en su momento la exactitud fue casi del 96%, pero con el modelo prediciendo en 4 ocasiones que era benigno cuando realmente no lo era, frente a una única predicción diciendo que era maligno cuando en realidad era benigno.

Esto nos lleva a preguntarnos ¿son todas las equivocaciones del modelo iguales? Desde el punto de vista ético qué es mejor ¿predecir benigno cuando no lo es o predecir maligno cuando no lo es? En ambas circunstancias el modelo se equivoca y probablemente cause un perjuicio, pero ¿éticamente hay errores con peores consecuencias que otros?

Los modelos precisos empiezan a formarse a partir de datos precisos. En este contexto debemos evitar: los datos correlacionados, los datos inválidos o incorrectos, y la falta de conocimiento del dominio. Además de ello durante la creación del modelo [en concreto en la fase de análisis y preprocesamiento de los datos] se deberán tomar una serie de decisiones que, a veces sin querer, llevan a que el modelo empeore su funcionalidad o que se introduzca un sesgo que existía previamente en los datos y que se ha pasado por alto. Por ejemplo, en el ejemplo anterior, que el modelo prediga erróneamente un tumor maligno como benigno en mayor porcentaje para personas de raza negra. Esa es la implicación social del modelo.

Para evaluar un modelo no sólo debemos tener en cuenta la exactitud y el resto de las métricas

que vimos en el Capítulo 16, sino también otras consideraciones entre las que se encuentran los sesgos éticos y sociales que los datos de origen pueden tener, aquellos que introduce el o la analista de datos e incluso el algoritmo de aprendizaje seleccionado.

El objetivo es crear modelos²⁹¹ que sean **justos, éticos o con equidad**.

Mi IA favorita dice: La equidad es un principio de justicia que busca ofrecer un trato imparcial y equilibrado a todas las personas, tomando en cuenta sus necesidades y circunstancias específicas para reducir o eliminar las desigualdades. A diferencia de la igualdad, que implica un trato idéntico para todos, la equidad se enfoca en proporcionar a cada individuo lo que necesita para tener las mismas oportunidades de éxito. La equidad se aplica en diversos contextos como el social, económico, educativo y legal, y se considera un ideal en muchas sociedades para garantizar que todos los individuos puedan acceder a oportunidades y recursos de manera justa.

Lo habitual, en modelos complejos, es sacrificar equidad a favor de la exactitud. En estos casos un modelo más justo sería aquel que, a expensas de sacrificar la exactitud – dentro de un margen – ganamos en equidad. El objetivo, por tanto, es gestionar el equilibrio entre la exactitud y la equidad. Esto se consigue **considerando a la equidad como una métrica más**. El modelo que consiga un mejor balance entre ambos conceptos, será el modelo prioritario desde el punto de vista ético.

18.3.2 Privacidad y seguridad

Cada vez más las empresas que trabajan en inteligencia artificial necesitan más datos; y cuanto más exactitud/precisión debe poseer el modelo más datos va a necesitar. Eso nos lleva a que las empresas cada vez recogen y guardan más datos, y una violación de estos datos (robo, filtración, etc.) pueden suponer una penalización muy grave para una empresa.

En este ámbito es fundamental garantizar la protección de la privacidad y la seguridad de estos datos. Las empresas y organizaciones deben implementar medidas adecuadas para asegurar que los datos se recopilen y almacenen de manera segura, y se utilicen de acuerdo con las leyes y regulaciones de privacidad aplicables. Si estos datos caen en manos equivocadas o se utilizan de manera indebida, pueden comprometer la privacidad y generar riesgos para la seguridad de las personas.

¿Cómo se recopilan estos datos? Dado que los algoritmos que incorporan el aprendizaje entre sus características son cada día más numerosos, es una práctica habitual recopilar datos de cualquier fuente. Incluso desde inocentes aplicaciones que pueden recabar cantidades masivas de información de sus usuarios. ¿Te suena? Supongo que sí. Si bajas e instalas aplicaciones gratuitas²⁹² en tu móvil, estas aplicaciones probablemente capten todos los datos que puedan de ti a través del dispositivo. Para estas aplicaciones sus usuarios son un *dataset*. **Somos un producto; su producto.**

291 *Está fuera del ámbito de este capítulo, pero la equidad se consigue mediante la paridad estadística de los datos de entrada (dataset) y la paridad de los ratio de error en las predicciones del modelo. Si el modelo se equivoca, debe equivocarse para todos los grupos sociales, étnicos, de género, etc. por igual.*

292 *Aunque contengan anuncios.*

Reflexiona: *¿Qué pasará cuando cada empresa u organización desarrolle sus propios algoritmos de aprendizaje?*

Las redes sociales procesan cada día millones de paquetes de información sobre sus usuarios, desde sus datos personales a su relación con el resto de usuarios. Cualquier red social tiene a su alcance una vasta cantidad de información que usan en primera instancia para conocernos y finalmente para obtener ingresos. Para alcanzar estos objetivos los departamentos informáticos de estas redes sociales emplean algoritmos de aprendizaje automático. Los datos que las redes sociales manejan no sirven sólo para conocer lo que hemos hecho, visto, comentado o visualizado, si no para clasificarnos y predecir nuestro comportamiento ante determinadas circunstancias.

Y esa información vale dinero. Y cuando algo tiene un valor, se crea un mercado a su alrededor.

Pero también nos encontramos con servicios libres de culpa [por ejemplo aplicaciones de consulta del tiempo, de compra por internet, etc.] que recolectan información, la **anonimizan**²⁹³ y la ponen a disposición de investigadores/as, comerciales o científicos/as de datos. Estos *datasets* que, individualmente, no incorporan información personal, **combinados pueden re-identificar** a individuos o, en el mejor de los casos, a grupos pequeños de individuos [como por ejemplo un vecindario].

Por ejemplo, el sistema *StreetView* de Google. Con casi 100% de cobertura, hay una buena posibilidad de que Google tenga una vista de tu casa en su *dataset*. Un ser humano mirando estos datos es una cosa, pero ¿qué sucede cuando un algoritmo echa un vistazo? Este ejemplo publicado en *MIT Technology Review*²⁹⁴ muestra que una imagen de Google *StreetView* de donde vives, cuando se introduce en un modelo predictivo, puede predecir el riesgo de un accidente automovilístico. Este es un ejemplo de un **algoritmo de caja negra**: no sabemos cómo es capaz de predecir este riesgo de accidente. Si una compañía de seguros u organización de salud tuviese acceso a estos datos, podrían mejorar drásticamente sus modelos de riesgo²⁹⁵ analizando imágenes de hogares de los titulares de pólizas. Esto es lo que los investigadores del MIT dicen que podría ser utilizado en el futuro.

Aquí es donde empezamos a plantear algunas preguntas éticas, debido a que no tenemos control sobre estos *datasets* públicos de satélites que fotografían nuestras casas. Pero ¿y si basándonos en esa imagen, tu prima de seguros sube? ¿Es eso justo? El artículo sugiere: *“El resultado plantea preguntas importantes sobre la forma en que la información personal puede filtrarse de conjuntos de datos aparentemente inocentes y si las organizaciones deben poder utilizarla con fines comerciales”*.

La palabra clave aquí son **conjuntos de datos aparentemente inocentes**. Esto se relaciona con nuestra lenta erosión de la privacidad. Como sabemos en Internet, en este momento, cada vez más datos personales son recolectados por empresas de redes sociales, por empresas de *big data* y por investigadores de aprendizaje automático. No solo está creciendo la cantidad de datos personales,

293 Eliminar información personal. Por ejemplo eliminando total o parcialmente nombres, direcciones, dnis, etc.

294 <https://www.technologyreview.com/2019/04/30/135556/how-a-google-street-view-image-of-your-house-predicts-your-risk-of-a-car-accident/>

295 Modelos a partir de los cuales infieren cuánto te van a cobrar o, incluso, si te aceptan o no.

sino que también con el tiempo el poder algorítmico para re-identificar a las personas basándose en esos conjuntos de datos personales también está creciendo, como ya indicamos anteriormente. Llegaremos a un punto en el futuro en el que cualquier información aparentemente inocente puede ponernos en riesgo de ser identificados por un algoritmo.

Aquí es donde realmente es importante tener conciencia como ciudadanos y ciudadanas; tenemos que pensar no solo en lo que hay en un conjunto de datos, sino en lo que habrá en un conjunto de datos futuro. Entregamos nuestra información que aparentemente no es sensible, hoy en día, pero es posible que podría serlo en el futuro; cuando ya no tengamos control sobre ella. Peri sí ella influencia sobre nosotros/as.

18.3.3 Responsabilidad y transparencia

A medida que al aprendizaje automático se utiliza en aplicaciones críticas, como la toma de decisiones en el ámbito de la salud o el sistema judicial, es importante establecer mecanismos de responsabilidad y transparencia. Los algoritmos pueden ser complejos y difíciles de entender, lo que plantea el desafío de garantizar que las decisiones tomadas por estos sean comprensibles y justificables. Esto implica la necesidad de desarrollar técnicas para explicar y auditar los resultados de los modelos de aprendizaje automático.

La definición de un modelo ético implica que tiene que ser en primer lugar preciso, en segundo lugar explicable y finalmente en tercer lugar su equidad, esto es, que sea justo y abierto.

Un modelo explicable es aquel que ofrece una comprensión clara y comprensible de cómo toma sus decisiones o realiza sus predicciones. En contraste con los modelos de caja negra, que son complejos y difíciles de interpretar, los modelos explicables buscan ser transparentes en su funcionamiento para que los humanos puedan entender la lógica detrás de sus decisiones.

y

Un modelo abierto es aquel cuyo código fuente, algoritmos, estructura y datos de entrenamiento están disponibles al público. Esto permite que cualquier persona pueda examinar, modificar o adaptar el modelo para sus propias necesidades o para fines de investigación.

¿Por qué construir un modelo absolutamente ético? La verdad es que en un contexto comercial, un modelo egoísta [ni justo ni abierto] tiene muchas ventajas sobre un modelo ético. Primero es más preciso, lo que conduce a un aumento de los beneficios. En segundo lugar, es menos explicable, lo que significa que sus competidores no pueden copiar y mejorar el algoritmo. Y en tercer lugar, es menos justo, hecho que realmente no importa tanto mientras la gente no esté prestando atención y no haya prensa cerca.

En un modelo ético tienes que renunciar a algo de exactitud en favor de la equidad. Además, un

modelo explicativo mostrará el razonamiento de su algoritmo. Por lo tanto, es más fácil para los competidores copiarlo.

Entonces, a medida que un modelo más explicable sea más justo y abierto, ¿tendrá atractivo para los negocios? Bien, estos tipos de modelos poseen algunas ventajas ocultas. [1] La **adopción extensa**. Al acceder libremente la comunidad al modelo, provoca que la empresa se parecerá más a una institución de investigación, publicando artículos y ganando participación en el avance, por el bien de la comunidad. [2] La **credibilidad**, que ayuda a la marca comercial. Porque tener credibilidad significa que cualquier persona que necesite ayuda con el modelo, va a llegar a la empresa y no a otra en busca de ayuda. Y [3], las empresas con modelos justos y abiertos tienen la clara ventaja de que son más capaces de **atraer al talento** investigador.

Hoy en día esta estrategia se ve claramente en las empresas Apple y Meta [antigua Facebook], rezagadas respecto a sus competidores en el ámbito de la inteligencia artificial.

La tendencia es que para las empresas especialmente bien asentadas, las ventajas del modelado ético están dando más frutos que las desventajas de los modelos egoístas. Y esto es una tendencia que continuará. Sin embargo, hay algunos límites a la apertura, especialmente cuando se trata de seguridad. Si pensamos que hay muchas empresas de IA que están trabajando con gobiernos, entonces podemos deducir que hay muchos datos confidenciales por ahí y, por tanto, tenemos que aplicar límites a la apertura de los mismos; si no es por la ética, entonces por la legalidad o incluso la seguridad nacional.

Por ejemplo, en 2018 la aplicación de *fitness Strava* usaba *hotmaps* para mostrar rutas populares; usaron esas rutas en su modelo de recomendación. Al hacerlo y sin quererlo filtraron información militar sensible ya que los soldados llevaban pulseras de la marca mientras hacían deporte en instalaciones secretas de los EE.UU.

Volvamos por un momento a los modelos explicables para introducir los **conceptos de confianza y cooperación entre agentes**. Tomemos un ejemplo de un modelo típico de caja negra utilizado por un banco. Las solicitudes de préstamo acarrean decisiones binarias tipo sí/no, llevadas a cabo por un algoritmo que, imaginemos²⁹⁶, no entendemos cómo lo hace. Se necesita un trabajador especializado en estos casos para revisar la solicitud y aprobarlo o denegarlo teniendo en cuenta las decisiones modelo. Este especialista en este caso obtiene una decisión de sí o no y luego basado en la confianza del modelo de caja negra puede tener que revisar este escenario antes de aprobarlo.

¿Y si hubiera una mejor manera? Esto nos devuelve al concepto de la IA explicable. La cooperación entre agentes, en este caso los modelos (como agentes inteligentes) y los humanos, depende de la confianza. Esa es una de las bases de la IA explicable. ¿Cómo conseguimos que las personas que usan estos modelos predictivos confíen más en ellos?

Si los humanos deben aceptar predicciones algorítmicas, necesitan confiar en ellas, de lo contrario, si el especialista pierde la fe en el modelo, él o ella simplemente ignorará por completo esos resultados y usará su instinto u otros datos. Por lo tanto, al reconocer que los agentes humanos necesi-

296 Hay pocos algoritmos realmente explicables. Principalmente los basados en árboles de decisión (random forest, xgboost, etc.).

tan trabajar más cerca con la IA, los sistemas de aprendizaje automático deberán tener la **capacidad de explicar su resultado**, caracterizar sus fortalezas y debilidades, y transmitir una comprensión de cómo se comportarán en el futuro. Eso genera confianza y afianza la cooperación entre los agentes implicados.

Finalmente vamos a conceptualizar un **modelo de caja de vidrio**, intermedio a medio camino entre el modelo de caja negra y caja blanca, que se ajustaría a la descripción de IA explicable, debido a que revelaría exactamente cómo tuvo en cuenta las diversas preguntas de la solicitud de préstamo en la decisión. Así, con la solicitud de préstamo, el modelo de caja de vidrio mostrarían los atributos ponderados y la recomendación. Esto ayudaría a crear confianza, porque mostraría en vez de un frío e inexplicable sí/no, los valores que llevaron a esta decisión. El especialista de préstamos podría así detectar decisiones equivocadas o sesgadas: pongamos que un atributo está muy fuera de lugar (*outlier*), el sistema simplemente podría incorporar un *checkbox* que inhabilitaría el atributo que provoca el sesgo. En general, esto crearía confianza en el proceso de toma de decisiones al hacer que el especialista trabaje estrechamente con el algoritmo en lugar de someterse a él.

18.4 Beneficios del aprendizaje automático

Pero no todo han de ser riesgos, si no este libro no existiría. El aprendizaje automático ha demostrado tener numerosos beneficios en diversos ámbitos de la sociedad. A continuación, explicaremos algunos de estos beneficios, que podríamos complementar con el Capítulo 3:

Mejora de la eficiencia en diversos sectores: El aprendizaje automático permite automatizar tareas que antes requerían tiempo y esfuerzo humano, lo que conduce a una mayor eficiencia. Por ejemplo, en el sector manufacturero, los modelos correctamente entrenados pueden optimizar la cadena de producción y predecir fallos en maquinarias, evitando costosos tiempos de inactividad. En el ámbito de la logística, puede ayudar a mejorar la gestión de inventario y a optimizar las rutas de entrega, reduciendo los costos y el tiempo de transporte.

Automatización de tareas repetitivas y monótonas: El aprendizaje automático puede encargarse de realizar tareas rutinarias y repetitivas de manera automatizada, liberando a las personas para enfocarse en actividades más creativas y estratégicas. Por ejemplo, en el ámbito de la atención al cliente, los *chatbots* pueden responder preguntas frecuentes, brindando asistencia instantánea y permitiendo que los empleados se centren en consultas más complejas y personalizadas. Esto mejora la experiencia del cliente y aumenta la productividad de los profesionales.

Toma de decisiones más precisas y basadas en datos: El aprendizaje automático tiene la capacidad de analizar grandes volúmenes de datos (ayudado por técnicas y tecnologías de *Big Data*) y extraer patrones ocultos que podrían ser difíciles de identificar para los humanos. Esto permite tomar decisiones más precisas. Por ejemplo, en el ámbito de la medicina, los algoritmos de aprendizaje automático pueden analizar datos médicos y diagnósticos previos para ayudar a los médicos a tomar decisiones más acertadas en el diagnóstico de enfermedades.

Avances en la atención médica y la investigación científica: El aprendizaje automático ha tenido un impacto significativo en el campo de la salud y la investigación científica. Por ejemplo, estos modelos entrenados pueden analizar imágenes médicas, como resonancias magnéticas o radiografías, para detectar enfermedades y anomalías de manera más precisa y temprana. Además, podría acelerar la investigación de nuevos fármacos al analizar grandes cantidades de datos y predecir la efectividad de ciertos compuestos químicos.

RETOS DEL CAPÍTULO 18

1. Vuelve al Capítulo 3, escoge un campo de aplicación en concreto y analizarlo desde un punto de vista ético.
2. Si está a vuestro alcance, visionar el documental “Sesgo Codificado” y debatir sobre su contenido y las enseñanzas que nos muestra.
3. Usando tus propias palabras, ¿qué es la ética?
4. Reflexiona ¿Una persona que no distingue el bien del mal se le puede considerar éticamente responsable de sus actos?
5. ¿Qué diferencias encuentras entre el aprendizaje automático y la inteligencia artificial?
6. En grupo: Investigar casos reales de sesgo y discriminación en algoritmos y modelos de aprendizaje automático. Usar la web para buscar y describir esos casos.
7. Tema de debate: ¿Cómo crees que el aprendizaje automático puede afectar a la privacidad de las personas?
8. Trabajo en grupo: Realizar individualmente una encuesta en tu familia o entorno personal sobre la percepción de los riesgos y beneficios de la inteligencia artificial. Agrupar seguidamente los datos, eliminando cualquier dato que pueda identificar a una persona o familia en concreto, y debatirlos en el aula.
9. Trabajo en grupo: Investiga ejemplos de regulaciones nacionales de otros países diferentes al tuyo, relacionadas con la inteligencia artificial.
10. Tema de debate: ¿Deben las decisiones más importantes ser tomadas por humanos en lugar de modelos de aprendizaje automático? En caso negativo ¿Qué deben cumplir las IAs en estos casos para poder tomar esas decisiones?
11. Tema de debate: ¿Cómo crees que el aprendizaje automático podría cambiar la educación en el futuro?
12. Tema de debate: Averigua qué es la seguridad nacional de tu país. Dado ese contexto y las aplicaciones potenciales de la IA en el mismo ¿Qué papel debería jugar la ética? ¿Una ética aplicada a la IA en el área de la seguridad nacional debilitaría sus objetivos principales?
13. Evalúa el impacto del aprendizaje automático en la creación de contenido en los medios de comunicación. Debate en grupo: ¿los medios de comunicación masivos tienen un gran poder de manipulación?
14. Tomando las conclusiones y lo aprendido en el reto anterior, reflexiona ¿Las IAs potenciarían ese poder de manipulación especialmente sobre la población más susceptible de ser manipulada?
15. Tomando las conclusiones y lo aprendido en el reto anterior, debatir sobre si los creadores de modelos de aprendizaje automático deben tener responsabilidades legales.

Apéndices

"Júpiter, el gigante del sistema solar, encuentra su eco en la inteligencia artificial, el gigante de la tecnología."

ChatGPT (2022 -)

Capítulo 19

METODOLOGÍA EDUCATIVA



Ilustración 6: *El docente y la IA son aliados en el aula.*

Fuente: Midjourney; propiedad del autor.

En este capítulo del apéndice vamos a ver una propuesta para integrar la IA en el aula. Empezaremos por analizar las funciones de la IA en educación, seguidamente propondremos un conjunto de patrones y finalmente veremos un prospección sobre cual sería el papel de la IA en un futuro en el aula.

19.1 Breve análisis del uso de la IA en la educación

Empezamos analizando el uso de la IA en la educación, dado que originalmente este libro se concibió como libro de texto e invita a su uso por parte de los/las estudiantes y las/los docentes.

La **UNESCO**, en calidad de Organización de las Naciones Unidas especializada en educación y dentro del contexto de la Agenda 2030, publicó un informe, con fecha de 2021, titulado²⁹⁷ “*Inteligencia Artificial y Educación: Guía para las personas a cargo de formular políticas*”.

²⁹⁷ Puedes leerlo aquí: <https://unesdoc.unesco.org/ark:/48223/pf0000379376/PDF/379376spa.pdf.multi>

Esta Agenda forma parte de un movimiento mundial encaminado a erradicar la pobreza mediante la consecución, de aquí a 2030, de 17 Objetivos de Desarrollo Sostenible [ODS]. La educación, fundamental para alcanzar todos estos objetivos, cuenta con su propio objetivo específico, el ODS 4, que se ha propuesto “**garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos**”.

Cito literalmente:

La inteligencia artificial (IA) tiene la capacidad de hacer frente a algunos de los mayores desafíos que afronta, hoy en día, el ámbito de la educación, de desarrollar prácticas de enseñanza y aprendizaje innovadoras y, finalmente, de acelerar el progreso en la consecución del Objetivo de Desarrollo Sostenible (ODS) 4. No obstante, estos avances tecnológicos rápidos implican inevitablemente numerosos riesgos y retos, que los debates sobre las políticas y los marcos reglamentarios tienen aún dificultades para superar.

Debemos tener en cuenta que este informe se publicó antes de la irrupción en la sociedad de las herramientas basadas en *grandes modelos de lenguaje* [LLM, Capítulo 12]. Si ya entonces se consideró la IA como elemento con capacidad innovadora y aceleradora en educación, gracias a los LLM podemos decir que estos suponen un elemento disruptivo, en todos los ámbitos; no sólo en el ámbito educativo.

El documento menciona modelos de lenguaje en el contexto de la inteligencia artificial y su aplicación en la educación. Específicamente, se refiere a los "Modelos lingüísticos autorregresivos que utilizan el aprendizaje profundo para producir textos similares a los humanos". Un ejemplo citado es GPT-3²⁹⁸, que puede producir texto "impresionantemente parecido al humano". Sin embargo, se aclara que, contrariamente a lo que podría parecer, el sistema no entiende el texto que produce.

Y, en efecto, no lo hace; pero lo importante no es si la máquina entiende o no el texto que produce, si no si lo que produce es útil y coherente con el conocimiento que su usuario/a busca o espera. Es su utilidad, no su capacidad, lo que finalmente nos debe sorprender.

Según este documento la IA aporta los siguientes beneficios a la educación, si nos ceñimos a aquellos directamente relacionados con la operativa en el aula:

1. La IA puede reimaginar la enseñanza y el aprendizaje, abriendo posibilidades que serían difíciles de lograr con métodos tradicionales. Puede ayudar en la transformación del aprendizaje y capacitar a docentes y escuelas.
2. La IA tiene el potencial de mejorar la educación para personas mayores, refugiados, comunidades marginadas o aisladas y personas con necesidades educativas especiales.

298 Versión anterior de ChatGPT (realmente GPT-3.5, el cual fue ajustado para responder preguntas, como un chatbot).

3. Las herramientas de IA pueden tomar tareas de transmisión de conocimientos, permitiendo a los docentes centrarse más en actividades de aprendizaje que requieren un pensamiento de orden superior, creatividad y colaboración interpersonal.
4. La IA tiene el potencial de ofrecer aprendizaje a lo largo de la vida de alta calidad, personalizado y ubicuo para cada estudiante.
5. La IA puede transformar el aprendizaje colaborativo al conectar a estudiantes a distancia, identificar los más adecuados para tareas colaborativas y agruparlos en consecuencia.

Y los siguientes retos o consideraciones en el ámbito de la ética:

1. Es crucial garantizar la protección de la privacidad y la seguridad de los datos de docentes y estudiantes. Esto incluye cuestiones como el consentimiento informado, especialmente en contextos donde los usuarios no son capaces de ello.
2. Hay preocupaciones éticas centradas en los datos y el sesgo, como la discriminación basada en género, edad, raza, estatus socioeconómico, entre otros. La IA puede amplificar las características ocultas de sus datos iniciales y reforzar de forma efectiva sus sesgos.
3. Además de los retos conocidos, el documento sugiere que hay 'incógnitas desconocidas', es decir, cuestiones éticas planteadas por la interacción de la IA y la educación que aún no se han identificado.
4. La IA en la educación debe ser accesible para todos, independientemente de su género, discapacidad, condición social o económica, origen étnico o cultural, o ubicación geográfica.
5. El impacto de la IA en los estudiantes, los docentes y la sociedad en general aún está por determinarse. Esto incluye cuestiones sobre la eficacia de las intervenciones de IA y lo que deberíamos enseñar en los centros educativos.

Desde nuestro punto de vista, el uso de la IA debe centrarse principalmente²⁹⁹ en dos áreas, con sus respectivas posibilidades y desafíos:

- **La evaluación de las competencias:** La IA favorece la automatización y la consecución en *tiempo real*³⁰⁰ del grado de consecución de las competencias por medio de la captación de datos y su posterior procesamiento, con el objetivo de inferir las necesidades del alumnado; individualmente o como grupo. Las tecnologías inteligentes de clasificación, reconocimiento facial y de voz o los grandes modelos de lenguaje entrenado específicamente, permitirían abordar estrategias como la evaluación psicométricas automatizadas, evaluaciones continuas con retroalimentación rápida, autocalificaciones, evaluaciones ciegas, evaluaciones asistidas por diálogos o por escritura, portafolios electrónicos avanzados, entre otras muchas posibilidades.
- **Los procesos de enseñanza-aprendizaje:** Es en el proceso de enseñanza-aprendizaje en don-

299 Queda fuera la *Mejora de la Gestión de la Educación* que, aún siendo muy importante, está más allá del horizonte de este libro.

300 El término "tiempo real" se refiere a la capacidad de un sistema para procesar datos y ofrecer resultados de forma inmediata, o con un retraso mínimo, después de recibir una entrada.

de estas tecnologías podrían ofrecernos los mejores beneficios. Lo veremos en un apartado posterior en este capítulo.

Pero no todo son buenas noticias.

La Unión Europea, en su proyecto de Ley para la regularización de la IA en su territorio, clasifica como “Alto Riesgo” los sistemas de IA pertenecientes a ocho ámbitos específicos, entre ellos a la “educación y formación profesional”. Aunque estamos de acuerdo con esta legislación, no debemos olvidar que las normas legales y la innovación suelen estar enfrentadas³⁰¹.

Desde hace años venimos implementando las **TIC en el aula**, con un gran esfuerzo inversor por parte de los estados y gobiernos, y un esfuerzo no menor, igualmente inversor pero en términos de tiempo y paciencia, por parte de la comunidad docente. Aún así la brecha digital persiste y, finalmente, hemos puesto en duda la existencia real de “nativos digitales”, ya que todavía hoy en día este término es objeto de debate y crítica en la comunidad académica y los expertos en tecnología.

Si tras todo este tiempo y esfuerzo compartido añadimos la IA en las aulas como una capa a mayores, ¿podría el sistema educativo asimilarlo? No, esta vez tenemos que repensar la IA en las aulas, para que esta surja de forma natural y no como la novedad que mantenga la atención del alumnado durante un tiempo. La clave es: el uso de herramientas que habiliten pero no aporten complejidad; de hecho, que apenas requieran entrenamiento por parte del personal docente para su aplicación.

Respaldando lo dicho, en el informe de la UNESCO se nombra el trabajo³⁰² de Selena Nemorin. En concreto se la menciona cuestionando la necesidad de la inteligencia artificial en la educación. El documento sugiere³⁰³ que aunque la IA ha sido objeto de investigación en el ámbito educativo durante más de 50 años, aún no está claro si las tecnologías que se están incorporando en la educación están realmente capacitadas para la tarea.

Desde nuestro punto de vista, las tecnologías inteligentes tal y como se han explicado a lo largo de este libro, no están listas todavía para ser aprovechadas de forma masiva en el aula, aunque sí puntualmente, como herramientas. ¿cuales? Vamos a verlo.

19.2 Procesos de enseñanza-aprendizaje asistidos por IA

La IA puede ser usada en el aula o desde casa, y proporcionar apoyo personalizado y en tiempo

301 *De hecho deben estar enfrentadas de forma natural; ya que la innovación puede, fácilmente, dañar derechos fundamentales de las personas, a veces incluso sin quererlo. Pero eso no elimina el pensamiento o el deseo de que hubiera un trato especial de la IA aplicada a la educación en aras de una innovación positiva.*

302 Nemorin, Selena & Vlachidis, Andreas & Ayerakwa, Hayford & Andriotis, Panagiotis. (2022). *AI hyped? A horizon scan of discourse on artificial intelligence in education (AIED) and development*. *Learning, Media and Technology*. 48. 1-14. 10.1080/17439884.2022.2095568.

303 *No reniega del uso de la IA en la educación; sin embargo sí transmite que esta aún no ha alcanzado la madurez necesaria (recordemos, escrito en el 2021) para ser aplicada. El objetivo del artículo es más exploratorio y analítico, buscando entender las diversas narrativas y percepciones en torno a la IA en la educación.*

real a los/las estudiantes en su aprendizaje. Puede ayudarles a comprender conceptos difíciles, generar ejercicios y evaluar su progreso, todo ello de manera adaptativa según las necesidades individuales. Esto es especialmente útil en un entorno de aprendizaje remoto, donde el alumnado pueden no tener acceso inmediato a ayuda. Además, la IA puede liberar tiempo del docente al manejar tareas rutinarias, permitiéndole centrarse en actividades más complejas y en la interacción directa con los estudiantes.

A día de hoy, la tecnología inteligente más adecuada para ser integrada en las aulas son los Grandes Modelos de Lenguaje [LLM³⁰⁴, Capítulo 12]. El resto de las tecnologías que hemos tratado y aquellas que sólo hemos nombrado o simplemente aquellas que no hemos incluido como contenido, no tienen una aplicabilidad directa³⁰⁵ hoy en día; pero sí se puede avistar como algunas, principalmente la visión artificial y la generación de imágenes/vídeo, podrían en un futuro próximo acompañar en el aula a los LLM.

¿Por qué los LLM pueden ser empleados en el aula inmediatamente?

Respuesta corta: Porque están ahí, al alcance del alumnado sin ningún control por nuestra parte. Así que, mejor integrarlos en el aula que “*poner puertas al campo*”.

Respuesta más larga: Porque podemos conceptualizar las habilidades de la IA como un Agente; como un medio para abstraer las necesidades del aula y ver cómo estas encajan con la IA actual. Llámemosle a este tipo de agentes, **asistentes educativos** [ASED]³⁰⁶.



Ilustración 7: *El docente, como un ángel guía, ilumina el camino de sus alumnos hacia el conocimiento y el crecimiento.*

Fuente: Midjourney; propiedad del autor.

Hoy en día, sin necesidad de crear más herramientas que las que hay, y prácticamente sin entrenamiento, un ASEd implementado en forma de LLM podría ocupar los siguientes roles:

- **Principalmente como guía:** El/la docente es la persona encargada de la guía del aula, aquella que dirige el proceso de enseñanza-aprendizaje, mostrando el camino y motivando

304 Incluyendo la transcripción y la síntesis del habla.

305 Relación eficacia/coste no adecuada. Demasiado pronto y demasiado caras.

306 El prompt de la Ilustración 7, en inglés, no determinaba el género, ya que se usó el término “teacher”. Midjourney generó un profesor blanco y alumnado acorde al prejuicio. Un claro sesgo de género y color de piel.

al alumnado a recorrerlo. Para aulas con poco alumnos/as este rol es factible y, como objetivo, fácilmente alcanzable; pero relacionado con el contexto en que nos encontramos, dos de los problemas más acuciantes para los docentes son [1] los diferentes puntos de partida y [2] los distintos estilos de aprendizaje. Esto es, ni todo el alumnado parte del mismo punto en cuanto a competencias, ni todos ellos/as aprenden al mismo ritmo, al mismo tiempo y ante los mismos estímulos.

Un/una docente, a partir de un determinado número³⁰⁷ de alumnos, simplemente deja de ser un/una guía para ser un/una socorrista, una persona dedicada la mayor parte del tiempo a apagar pequeños fuegos en el aula. Un alumnado no puede en este clima motivarse o dejarse motivar, más que por sus iguales.

Promover un ASEd como guía funcional del aula, mediante el uso de un agente conversacional, debería igualar las fuerzas ante las circunstancias nombradas, al permitir al docente ser el guía estratégico y al ASEd el guía operacional.

- **Como un igual:** En muchas ocasiones el trabajo en grupo o con un igual permite al alumnado crear una sinergia, una interacción cooperativa o colaboradora en la que el resultado conjunto es más efectivo o eficiente que la suma de los esfuerzos individuales. Haciendo que un ASEd interaccione con el alumnado como un igual introducimos un elemento reactivo, pero con gran potencial comunicativo.
- **Como contrincante:** El debate, el diálogo y el contraste de ideas, son herramientas pedagógicas de primer orden. Enfrentan conocimiento contra conocimiento, permitiendo el trasvase de este y asentando dudas razonables que habilitan una aproximación al método científico, al menos a su primera etapa: dudar de lo que sabes o crees saber.
- **Como evaluador:** En el apartado anterior hablábamos del papel que pueden jugar los ASEd como agentes evaluadores de competencias; pero, alejados de ese rol, los asistentes educacionales pueden interpretar el papel de evaluadores a efectos de autoprobar, por parte del alumnado, su propio conocimiento y aptitudes.

19.3 Patrones de aprendizaje asistido por ASEd

Una de las tareas en las que los LLM van a progresar más rápidamente es en la búsqueda de información. La búsqueda en portales como *Google* o similares ya está cambiando, al permitir búsquedas en forma de *prompts* escritos en lenguaje natural o casi natural. De todas formas estos cambios no serán rápidos; una serie de consideraciones sobre la escalabilidad de las arquitecturas basadas en LLMs y los enormes consumos energéticos que estas necesitan harán que los buscadores, vamos a denominarlos así, clásicos, no sean sustituidos inmediatamente por los basados en *prompts*.

³⁰⁷ Este número se ve decrementado si se trata de un aula conflictiva o con elementos indisciplinados.

Cada capítulo anterior a este posee su propia página de desafíos, retos o ejercicios. Gran parte de estos fueron creados por medio de unos patrones de aprendizaje, pensado en un asistente conversacional como ayudante. En estos retos se identificó al ASEd como “tu IA favorita” o “tu agente conversacional favorito”. Este apartado detallará estos patrones:

- **Búsqueda de información [quizá recursivamente] sobre ...**

Retomamos lo dicho sobre los LLM usados como buscadores. Su ventaja radica, desde el punto de vista del alumnado, en la inmediatez de las respuestas y la no necesidad de filtrar los resultados hasta encontrar lo que se buscaba; así que el alumnado lo preferirá respecto a los buscadores clásicos³⁰⁸. Al hacerlo tenemos la puerta abierta a proponer búsquedas de información más complejas, incluso recursivas: a medida que encuentra respuestas, si aparece una duda deberán volver a iniciar la búsqueda de nuevo, profundizando en las soluciones a sus dudas.

A lo largo del libro hemos usado esta técnica para ampliar la información del texto: ideas o conceptos que por alguna razón no se ha podido incluir en el capítulo, se le pide al alumno que busque información sobre el tema; en principio individualmente. La habilidad para crear el *prompt* adecuado y, a partir de la respuesta, profundizar buscando la información deseada, es una competencia digital y lingüística al mismo tiempo.

Propósito: Reforzar la información sobre un tema profundizando en sus conceptos y aumentar las habilidades de búsqueda de información compleja.

Papeles: guía.

Estado inicial: Falta de información en un ámbito que implica la necesidad de una búsqueda.

Resultado: Información, representada en algún formato.

- **Pregúntale o interroga a tu IA favorita acerca de ...; profundiza en ...**

Este método pretende que el alumnado, individualmente o en grupo, asuma un papel mucho más activo que una simple búsqueda de información. Se trata de generar juntos conocimiento profundizando sobre un tema, en donde el alumnado toma un rol activo para generar conocimiento.

Propósito: Reforzar el conocimiento sobre un tema de forma activa y aumentar el abanico de técnicas de búsqueda.

Papeles: guía, igual.

Estado inicial: Falta de conocimiento en un ámbito.

Resultado: Conocimiento, representado en forma de aprendizaje.

- **Conversa o dialoga con tu IA favorita acerca de ...**

Los puntos anteriores provocan una búsqueda, lineal o quizás en forma de árbol. Este método propone un diálogo, en el sentido de una conversación o intercambio de ideas, opinio-

³⁰⁸ También debemos tener en cuenta que los buscadores clásicos evolucionarán, acomodando el lenguaje natural entre sus capacidades de responder y aceptar preguntas de forma más humana.

nes o información entre dos o más personas o entidades. En este caso no tenemos un propósito inicial, sólo un objetivo definido. Precisamente el propósito inicial y todos los estados intermedios deben ser deducidos por el alumnado a partir de la interacción con el ASED.

Propósito: Aumentar el conocimiento y las habilidades de interacción escrita.

Papeles: guía, igual.

Estado inicial: Un objetivo, representado en forma de conclusión.

Resultado: Conocimiento, representado en forma de aprendizaje.

- **Reflexiona [quizá en grupo] con tu IA favorita,**

La *reflexión* se refiere al acto de pensar, considerar o meditar cuidadosamente sobre un tema, experiencia, idea o acción. Es un proceso cognitivo en el que una persona examina y analiza, en nuestro contexto, sus propios conocimientos o ideas preconcebidas con el fin de comprender mejor, aprender de un ámbito o tomar decisiones informadas, si fuese el caso.

La reflexión obliga a juntar diversas ideas o conocimientos, individualmente o en grupo, para llegar a una conclusión desconocida al inicio, por medio del descubrimiento.

Propósito: La consideración de diferentes perspectivas.

Papeles: guía, igual, contrincante.

Estado inicial: Un punto de partida concreto.

Resultado: Un camino de pequeñas píldoras de conocimiento, hasta llegar a una conclusión desconocida al principio.

- **Vuelve loca a tu IA favorita, ...**

La *imaginación* se refiere a la facultad mental que permite a una persona crear imágenes mentales, ideas o conceptos que no están presentes en la realidad inmediata o que no han sido experimentados directamente a través de los sentidos. La imaginación implica la capacidad de visualizar, concebir o inventar cosas de manera creativa y abstracta.

En los procesos de aprendizaje, la imaginación juega un papel crucial como precursora de las ideas, la creatividad, la innovación y la resolución de problemas. Esta estrategia busca el uso de la imaginación, proponiendo al alumnado que “vuelva loco a su ASED”, estableciendo un diálogo, pero dentro de un contexto inventado. Y dentro de él operar buscando información o conocimiento que podría incluso no existir previamente.

Propósito: Desarrollar la imaginación y las habilidades de interacción escrita.

Papeles: igual, contrincante.

Estado inicial: Un contexto inventado.

Resultado: Conocimiento, representado en forma de aprendizaje.

- **Pídele a tu IA favorita que genere un ...**

Este patrón representa la guía en un estado puro. El alumnado no sabe hacer algo concreto porque está fuera de sus competencias, y el ASED cubre esta falta con sus capacidades, quizá incluso enseñando el proceso o explicando el resultado.

A lo largo del libro hemos propuesto en varias ocasiones que el alumnado genere código en Python. Como indicamos el principio, no se da por hecho a lo largo del texto que los/las alumnos/as sepan codificar una solución a un problema, así que el ASED debe guiar en la dirección indicada para que esta necesidad no sea insalvable.

También, a lo largo del texto, en concreto a la parte de Procesamiento del Lenguaje Natural, es necesario poseer textos que cumplan unas determinadas características, por ejemplo: dos textos diferentes que hablen sobre lo mismo. El ASED aquí cumple la misma función que en el caso anterior.

Propósito: Salvar la brecha entre lo que se busca y lo que se sabe, sin introducirse necesariamente en el desarrollo de una competencia.

Papeles: Guía.

Estado inicial: Un objetivo concreto, con un requerimiento competencial que no está entre los objetivos a adquirir por parte del alumnado.

Resultado: Lo necesario para alcanzar el objetivo, más un aprendizaje superficial, fruto de un análisis guiado, de la solución producida.

- **Pídele a tu IA favorita que te ayude a ...**

Muy parecido al anterior, con la salvedad de que el requisito competencial sí se encuentra entre los objetivos a adquirir.

Propósito: Salvar una brecha existente entre lo que se busca y lo que se sabe.

Papeles: Guía.

Estado inicial: Un objetivo concreto, con un requerimiento competencial identificado que está entre los objetivos a adquirir por parte del alumnado.

Resultado: Lo necesario para alcanzar el objetivo, más un aprendizaje formal, fruto de un análisis guiado, de la solución producida.

- **Grupo de debate: incluyendo a tu IA favorita ...**

Un debate es una discusión formal en la que dos o más personas presentan argumentos y puntos de vista opuestos sobre un tema específico, con el objetivo de persuadir a otros o llegar a una conclusión.

En nuestro caso dividimos el aula en dos subgrupos, cada uno de ellos asistidos por el ASED y defendiendo puntos opuestos de un tema controvertido.

Propósito: Estimular el pensamiento crítico, pero flexible.

Papeles: Guía, igual, contrincante, evaluador.

Estado inicial: Un tema controvertido. Quizá una conclusión.

Resultado: Pensamiento crítico, oratoria, análisis de juicios ajenos, empatía, toma de decisiones informadas, creatividad y habilidades de escucha activa.

19.4 El aula del futuro asistida por IA

Si el apartado anterior presenta una propuesta, este pone encima de la mesa a cabo un ejercicio de predicción. ¿Cómo será el aula del futuro con las asistencia de la IA?

Aprovechando el cambio de paradigma que la IA provocará en la educación, podemos corregir los defectos que hemos aprendido que poseen las TIC en las aulas: la necesidad de entrenamiento en áreas en las que no son expertos los/las especialistas de las materias, con necesidad de renovación constante de habilidades digitales; dispositivos pequeños, de poca autonomía y muy orientados a la web, además de *relativamente* caros; falta de contenido digital apropiado o falta de estandarización de formatos de contenido; procedimientos clásicos de enseñanza-aprendizaje adaptados a las TIC de forma poco natural; y un largo etc.

Como decíamos, aprovechando las nuevas tecnologías inteligentes podremos reciclar el conocimiento adquirido en nuestras experiencias pasadas, y empezar a descubrir nuevas formas de enseñar, esta vez “asistidos por IA” y no asistiendo a hipotéticos “nativos digitales” en las TIC. Y la mejor forma de expresar una aproximación de esta visión es mostrar cómo sería un “aula” futura. Esta es nuestra visión:

- Sin duda será **ubicua**, probablemente virtual y a distancia, pero seguramente no abandonaremos la presencialidad del todo. La necesidad por parte del alumnado de relacionarse, crear sinergias entre ellas y ellos, y con el profesorado, y el contacto directo con las experiencias en un ambiente de estudio serán siempre necesarias. Además tenemos fuertes razones para creer eso: basadas en la necesidad de conciliación y en el uso de recursos compartidos con el propósito de optimizar la inversión en educación.
- Con toda probabilidad abandonaremos el modelo de enseñanza industrializada³⁰⁹, basado en la educación por lotes, donde un alumno o alumna ingresa en el sistema educativo obligatorio y permanecía en él, curso por curso, al igual que sus iguales. Cualquier diversidad que conlleve una desviación implica una acción correctiva en el plan individual de difícil encaje en el grupo. Todo lo contrario a una **educación personalizada**, asistida por la IA, en donde el alumnado avanza según sus competencias adquiridas.
- Los procesos de enseñanza-aprendizaje asistidos por IA, promoverán **metodologías inteligentes** en el aula, muchas ya conocidas, pero difíciles de implantar con éxito en las condiciones actuales: Los trabajos en grupo equilibrados, el aprendizaje inmersivo, el descubrimiento de información mediante la búsqueda y su afianzamiento en forma de conocimiento, el auto-refuerzo de las competencias, probablemente por medio de la generación de auto-conocimiento, el impulso de la creatividad y la conservación de la motivación para

³⁰⁹ El propósito no era otro que educar y disciplinar al mayor número de personas para la vida en sociedad y el trabajo productivo en un mismo espacio, en el menor tiempo y optimizando los recursos requeridos para ello.

aprender a lo largo de la vida, entre otros; serán procedimientos prácticos y no solamente teóricos y/o deseables, como lo son en estos momentos.

- La **tecnología inteligente más usada** será, probablemente, los agentes o asistentes conversacionales³¹⁰ con capacidad de reacción³¹¹, además del reconocimiento del habla y su síntesis, y la creación de contenido multimedia mediante IAs generativas. Todo ello integrado en soluciones que consigan hacer la complejidad transparente.
- Seguimos apostando por un **equipamiento individual por alumno**, además de auriculares con micrófonos, un dispositivo tipo *tablet*, de gran tamaño, batería que entregue una gran autonomía, usando tecnología de tinta electrónica en color y capacidad de *handwriting*; todo ello controlado por un sistema operativo que haría que este **eNotebook**³¹² se parezca más a una libreta electrónica y menos a un ordenador de escritorio. El auricular del alumno estaría enlazado con este dispositivo el cual, a su vez, se vincularía al agente conversacional del docente³¹³.
- Cada docente dispondría de un agente conversacional [**asistente educativo**, ASED], especializado y entrenado para reconocer su voz³¹⁴, interpretar y ejecutar sus órdenes e interactuar individual pero indirectamente con cada alumno.
- Cada alumno/a contaría con un agente conversacional [**asistente individual**, ASIN], especializado y entrenado para reconocer su voz³¹⁵, interpretar y ejecutar sus órdenes y tomar el control de su *eNotebook* según las obligaciones que su portador/a tenga.
- Los ASED de los docentes y los ASIN del alumnado formarían un **sistema multiagente** con un objetivo común. Los ASIN de cada alumno/a adaptarían la información recibida a la necesidad individual de cada uno de sus poseedores [por ejemplo con Necesidades Educativas Especiales].
- El o la docente generaría la información multimedia³¹⁶ de forma textual, texto a imagen o texto a vídeo, permitiendo una **explicación inmersiva**³¹⁷. Este contenido, más el textual adaptado, formaría parte del contenido que el alumnado tendría a su disposición.
- A petición de el/la docente, el ASED podría generar **retos o desafíos** para el aprendizaje, y enviárselos a los ASIN los cuales adaptarían según las diferentes competencias a estimular, según los objetivos y de acuerdo con las necesidades educativas especiales del alumno/a en cuestión.

310 Por medio de LLM, preparados para mantener conversaciones uno a muchos o muchos a muchos. Hoy en día ChatGPT, Bard et alia, son muy buenos en conversaciones uno a uno.

311 Tal y como lo vemos, no necesitamos agentes conversacionales con capacidad activa, si no simplemente reactiva.

312 Hoy en día sería muy caro, pero recordemos: en los 80 teníamos un ordenador personal por cada 25 alumnos en nuestra universidad.

313 Observar que las clases no tienen por que ser simultáneas si no son presenciales. El asistente del docente puede “asistir” en ausencia de esta/este.

314 Presupongo que, o bien el aula posee micrófonos/altavoces o que el docente viste un micrófono individual con auriculares. Esto segundo es más cómodo por ser cómodo y mejorar la calidad de la transcripción.

315 Presupongo, como ya dije, que el alumnado viste un micrófono individual con auriculares.

316 IA generativa. Hoy en día generan imágenes y pequeños vídeos. Aquí suponemos que ha evolucionado y pueden generar con precisión contenido educativo visual. Obviamente también contenido textual.

317 Aunque ahora mismo requiere de un preprocesamiento intenso, no es descabellado suponer que en unos años tendremos escenas 3D en donde podremos movernos por el medio a nuestra voluntad. <https://poly.cam/gaussian-splatting>

- La **evaluación sería continua y funcional**, tras cada unidad el/la docente, la familia y aquellos/as que tuviesen acceso, serían sabedores del avance del grupo y podrían llevar a cabo la toma de decisiones adecuadas para facilitar alcanzar los objetivos establecidos.

Todo lo explicado en los puntos anteriores puede ser adaptado a aulas virtuales, clases individuales, clases con un asistente ya sean síncronas o asíncronas, etc. Igualmente como asistencia a un trabajo en grupo o individual, o sesiones de actualización profesional.

¿Qué tecnologías inteligentes necesitamos para llevar a cabo esta visión?

1. La siguiente generación de grandes modelos de lenguaje [LLM], sobre todo más fiables y más eficientes.
2. Arquitecturas LLM de orden superior: gestión de conversaciones.
3. Transcripción del habla fiable, con múltiples acentos y múltiples idiomas.
4. Síntesis de habla sensible al contexto y a las emociones.
5. Generación de contenido avanzado y en tiempo real.
6. Sistemas multiagente.
7. Sistema operativo orientado a la educación para los eNotebooks, con acceso a la navegación web por medio de RAG [*Retrieval-augmented generation*, Generación Mejorada por Recuperación³¹⁸]. Además, su operativa básica sería la lecto-escritura multimedia e interacción alumnado/dispositivo por medio del audio y del habla.

De estos siete puntos, [1] el primero está a pocos meses de alcanzarse; [2] la gestión de conversaciones ya es una realidad³¹⁹; [3] la transcripción del habla³²⁰ y [4] la síntesis son una realidad en múltiples idiomas y múltiples acentos, además de sintetizar emociones; [5] respecto a la generación³²¹ de contenido aún estamos empezando, las técnicas de texto a imagen y texto a vídeo aún están comenzando a dar sus frutos y por supuesto no operan en tiempo real, por ahora; [6] las tecnologías basadas en sistemas multiagente son conocidas desde la década de los 80; y, finalmente, [7] los eNotebooks todavía están en los laboratorios³²², sin embargo los RAG están saliendo de los laboratorios y pronto los tendremos como productos comerciales.

Es factible que en pocos años, todo ello debidamente integrado, de lugar a una solución muy simple de usar en las aulas y **completamente disruptiva**, además de compatible con las actual legislación en términos de privacidad, protección de datos y las futuras transposiciones de la Ley Europea de la IA a los países europeos correspondientes.

Ahora sí que no habrá límites.

³¹⁸ Simplificando, búsqueda de información por medio de lenguaje natural. En nuestro caso, el alumnado no navegaría simplemente por la web, en su lugar navegaría por medio del lenguaje natural, “interrogando” a la web.

³¹⁹ Por ejemplo LangChain, como plataforma Open Source para elabora complejos sistemas conversacionales.

³²⁰ Por ejemplo whisper de OpenAI, o MMS de Meta que transcribe y sintetiza en más de mil idiomas.

³²¹ Las tecnologías como NeRF y Gaussian Splatting prometen mucho, especialmente esta última.

³²² Probablemente la pieza más retrasada del puzzle: <https://www.youtube.com/watch?v=omD3BIh9Czs>

Índice de figuras

Figura 1: Pirámide DIK+W.....	16
Figura 2: Algunos participantes de la convención de Dartmouth.....	25
Figura 3: Cantidad real y prevista de datos generados en todo el mundo.....	27
Figura 4: Clasificación básica de la Inteligencia Artificial.....	37
Figura 5: Esquema básico del modelado matemático.....	44
Figura 6: Grafo no dirigido, etiquetado, de 15 ciudades españolas y los kilómetros que las separan.....	45
Figura 7: El estado inicial del problema de las N reinas, en este caso 4, [izquierda], y uno de los posibles estados finales [derecha].....	47
Figura 8: Algoritmos Genéticos en 5 minutos.....	53
Figura 9: Componentes de un agente inteligente.....	56
Figura 10: Alto, ancho de una imagen y los planos que forman el modelo RGB.....	81
Figura 11: Cubo RGB.....	85
Figura 12: Cilindro HSV.....	85
Figura 13: Espacio de color LAB.....	85
Figura 14: Resultados de aplicar una máscara a una imagen [and y and not].....	91
Figura 15: Aplicación de una máscara de convolución Sobel Horizontal a una imagen RGB.....	92
Figura 16: Diferencia entre segmentación e identificación.....	96
Figura 17: Eliminación de ruido usando redes neuronales artificiales [entada/salida].....	97
Figura 18: Uso de filtros para eliminar ruido de una imagen.....	99
Figura 19: Ejemplo de mejora de contraste.....	99
Figura 20: Descripción "black box" de un modelo para identificar y localizar patrones visuales en imágenes.....	104
Figura 21: Ejemplo del resultado de un modelo preentrenado.....	106
Figura 22: Ejemplo del resultado de un modelo preentrenado multiclas.....	107
Figura 23: Ejemplo de resultado de la identificación de patrones visuales en una imagen sin filtrar el resultado.....	108
Figura 24: Ejemplo de resultado de la identificación de patrones visuales en una imagen con el resultado filtrado.....	109
Figura 25: Ejemplo de imagen mostrando los superpixeles con dos granularidades.....	110
Figura 26: Ejemplo de segmentación semántica, original y segmentada.....	110
Figura 27: Ejemplo de segmentación semántica, original.....	111
Figura 28: Ejemplo de segmentación semántica, segmentada.....	111
Figura 29: Imagen ejemplo del dataset gshapes.....	117
Figura 30: Ejemplo de imagen del dataset gshapes, con sus etiquetas dibujadas.....	118
Figura 31: Estimación de la pose por medio de marcadores visuales.....	124
Figura 32: Ejemplo de uso de redes neuronales artificiales aplicadas a la estimación de la pose en una fotografía.....	124
Figura 33: Resultado de una estimación de pose usando redes neuronales artificiales sobre una imagen con dos personas.....	125
Figura 34: Otra forma de representar los tracking.....	129
Figura 35: IOU sobre rectángulos.....	130
Figura 36: Esquema de la visión estereoscópica humana.....	132
Figura 37: Entrada y salida del pipeline de estimación monocular de la profundidad.....	133
Figura 38: Gráfica 2d del clusterig de expresiones faciales.....	138
Figura 39: Representación 3d de los clusters.....	139
Figura 40: Imagen de grupo en donde podemos ver, entre otros, a Jennifer.....	141
Figura 41: Imagen "papel105.png".....	142
Figura 42: Imagen "papel105.png" con puntos de referencia.....	142
Figura 43: Resultado del reconocimiento de gestos en un enfrentamiento al "piedra, papel, tijera".....	145
Figura 44: Ejemplo de una hipotética unidad de cirugía operada mediante brazos robóticos.....	152
Figura 45: Caras generadas artificialmente. Estas caras no existen, fueron "imaginadas" por una IA.....	157
Figura 46: Tabla ASCII [parcial] de 7 bits, del carácter 32 al 126.....	169

Figura 47: Vista muy parcial de un bloque de caracteres Unicode.....	171
Figura 48: Ejemplo de un proceso de embedding sobre unas tokenización basada en palabras.....	178
Figura 49: Texto (arriba), oscilograma de su señal acústica [debajo] y su espectrograma [abajo].....	187
Figura 50: BOW del párrafo 1.....	195
Figura 51: BOW del párrafo 2.....	195
Figura 52: Idea intuitiva del concepto de atención en Transformers.....	219
Figura 53: Representación histórica de modelos basados en Transformer con el tamaño de sus parámetros.....	221
Figura 54: Arquitectura Transformer original.....	223
Figura 55: Patrón encoder/decoder general.....	224
Figura 56: Stephen Hawking fotografiado en su silla dotada de un sistema de PLN.....	243
Figura 57: Relación entre datos, algoritmo y modelo en el aprendizaje automático.....	250
Figura 58: Esquematización del método tradicional de producción de sistemas informáticos.....	251
Figura 59: Relación entre inteligencia artificial y ciencia de datos.....	251
Figura 60: Parte de la tabla original del dataset clásico Iris.....	253
Figura 61: Datos tabulados de dataset parcial Iris.....	253
Figura 62: Ejemplo de árbol de decisión.....	254
Figura 63: Ejemplo de aprendizaje automático a partir de datos usando árboles de decisión.....	255
Figura 64: Explicación de SVM.....	255
Figura 65: Representación en 2D de las variables independientes del dataset Iris.....	258
Figura 66: Esquema de un sistema de aprendizaje por refuerzo.....	259
Figura 67: Estimación de la cantidad real y prevista de datos generados en todo el mundo (en zettabytes).....	267
Figura 68: Fases y roles implicados en la ciencia de los datos.....	269
Figura 69: Tipos de aprendizajes según e resultado: subentrenamiento, balanceado y sobreentrenamiento.....	270
Figura 70: Esquema general de matriz de confusión para un clasificador binario.....	282
Figura 71: Matriz de confusión, dataset: cancer_breast, algoritmo: regresión logística.....	282
Figura 72: Esquema de un árbol de decisión.....	285
Figura 73: Muestra del dataset MNIST Digits.....	290
Figura 74: Ejemplo de aplanamiento de una imagen en 2D [5x5] a una forma en 1D [1x25].....	290
Figura 75: Matriz de confusión multiclas 10x10 y métricas.....	291
Figura 76: Matriz de confusión multiclas 10x10 con un algoritmo árbol de decisión.....	292
Figura 77: Matriz de confusión multiclas 10x10 y métricas con un algoritmo de random forest.....	292
Figura 78: Matriz de confusión multiclas 10x10 y métricas con un algoritmo XBoost.....	293
Figura 79: Matriz de confusión multiclas 10x10 y métricas para OvA y OvO con SVM.....	293
Figura 80: Conjunto de datos MNIST Fashion (parcial).....	294
Figura 81: Matriz de confusión multiclas 10x10 y métricas para XGBoost sobre MNIST fashion.....	294
Figura 82: Dataset aleatorio modelado mediante regresión simple.....	295
Figura 83: Dataset aleatorio modelado mediante regresión polinómica.....	295
Figura 84: Dataset aleatorio modelado mediante regresión simple.....	296
Figura 85: Dataset aleatorio modelado mediante regresión polinómica.....	296
Figura 86: Ejemplo de clustering con el dataset Iris en 2D.....	297
Figura 87: Ejemplo de clustering con el dataset Iris en 3D.....	297
Figura 88: Ilustración de una neurona biológica.....	303
Figura 89: Esquema básico del modelo matemático de una neurona con 5 entradas.....	305
Figura 90: Funciones de activación básicas (linear, step y sign).....	306
Figura 91: Esquema de redes neuronales artificiales multicapa densas (fully connected).....	312
Figura 92: Funciones de activación modernas.....	312
Figura 93: Curvas de aprendizaje para un MLP sobre el dataset "cancer_breast".....	314
Figura 94: Matriz de confusión multiclas 10x10 y métricas con CNN.....	316
Figura 95: Ejemplo de predicción de CLIP.....	318
Figura 96: MLP con 5 capas [3 ocultas], todas de tamaño 16.....	319
Figura 97: Autoencoder con 5 capas [3 ocultas].....	319
Figura 98: Esquema de un autoencoder.....	320
Figura 99: Entrada (arriba) y salida (abajo) de un autoencoder simple con el dataset MNIST digits.....	320

Figura 100: Autoencoder entrenado con imágenes [64x64] de perros.....	321
Figura 101: Rostros sintéticos producidos por StyleGAN.....	322
Figura 102: Esquema de la arquitectura de DALL-E 2.....	324
Figura 103: Arquitectura GLIDE para transformar una imagen.....	325
Figura 104: Captura del encabezamiento del artículo.....	329
Figura 105: Matriz de confusión del modelo creado con el algoritmo SVM sobre el dataset cancer_breast.....	332

Índice de videos

Vídeo 1: Algoritmo Minimax en 4 minutos.....	49
Vídeo 2: ¿Qué es el Descenso del Gradiente? Algoritmo de Inteligencia Artificial.....	54
Vídeo 3: Vídeo original.....	128
Vídeo 4: Vídeo con tracking.....	128
Vídeo 5: Vídeo con los centros dibujados.....	128
Vídeo 6: Real Time Object Detection using YOLOv3 on Road Traffic on Nvidia RTX 2060.....	128
Vídeo 7: Tesla Autopilot Augmented Vision Test.....	133
Vídeo 8: New AI tool describes surroundings to visually impaired people.....	154
Vídeo 9: Virtual Production - A Cinematographers' Conversation.....	154
Vídeo 10: Fake Obama created using AI video tool.....	158
Vídeo 11: Tricked by the fake Obama video? Deepfake technology, explained.....	158
Vídeo 12: Grabación de audio generado por medio de la síntesis de voz concatenativa.....	186
Vídeo 13: Animación del algoritmo KMeans durante el entrenamiento.....	299
Vídeo 14: Synthesizing High-Resolution Images with StyleGAN2.....	323
Vídeo 15: How does DALL-E 2 actually work?.....	324

Índice de ilustraciones

Ilustración 1: Osa panda roja, rodeada de tecnología [imagen original].....	1
Ilustración 2: Recreación artística de la máquina de Anticitera, realizada por Midjourney.....	21
Ilustración 3: Las IAs no van a sacarte el trabajo; te sacará el trabajo aquella persona que use las IAs.....	63
Ilustración 4: Si las máquinas aprenden a ver ¿quien va a controlar a las máquinas?.....	149
Ilustración 5: Figura 58: El machine learning está de moda, pero ¿cuándo estará de moda el human learning?.....	176
Ilustración 6: El docente y la IA son aliados en el aula.....	343
Ilustración 7: El docente, como un ángel guía, ilumina el camino de sus alumnos hacia el conocimiento y el crecimiento.....	347

Bibliografía

1. “*Inteligencia Artificial: Técnicas, métodos y aplicaciones*”; ed. J. T. Palma Méndez, R. Marín Morales; 2008, McGraw-Hill. ISBN: 978-84-481-5618-3.
2. “*La Matemática: Su contenido, métodos y significado*”; A. D. Alexandrov, A. N. Kolmogorov, M. A. Laurentiev; 2016, Alianza Editorial. ISBN: 978-84-206-9330-9.
3. “*Deep Learning*”; I. Goodfellow et al.; 2016, MIT Press. ISBN: 978-0-262-03561-1.
4. “*An introduction to Machine Learning*”; M. Kubat; 2017, Springer. ISBN: 978-3-319-63912-3.
5. “*Aprende Machine Learning con Scikit-Learn, Keras y Tensorflow*”; A. Gerón; 2020, O'Reilly, Anaya Multimedia. ISBN: 978-84-415-4264-8.
6. “*Reinforcement Learning: An introduction*”; R. S. Sutton, A. G. Barto; 2020, MIT Press. ISBN: 9780262039246.
7. “*Machine Learning Design Patterns; Solutions to common challenges in data preparation, model building, and MLOPS*”; V. Lakshmanan, S. Robinson, M. Munn; 2020, O'Reilly. ISBN: 978-1-098-11578-4.
8. “*Proyectos de Inteligencia Artificial*”; J. Miralles Solé; 2020. ISBN: 9781661199456.
9. “*Graph Machine Learning*”; C. Stamile, A. Marzullo; E. Deusebio; 2021, Packt Publishing. ISBN: 978-1-80020-449-2.
10. “*The Road to Conscious Machines: The history of AI*”; M. Wooldridge; 2021, A Pelican Book. ISBN: 978-0-241-33390-7.
11. “*Transformers for Natural Language Processing*”; D. Rothman; 2021, Packt. ISBN: 978-1-80056-579-1.
12. “*Ética de la Inteligencia Artificial*”; M. Coeckelbergh; 2021, Cátedra. ISBN: 978-84-376-4213-3.
13. “*Distributed Artificial Intelligence: a modern Approach*”; Ed. S. P. Yadav, S. P. Mahato, N. T. D. Linh; 2021, CRC Press. ISBN: 978-0-367-46665-7.
14. “*Getting Started with Natural Language Processing*”; E. Kochmar; 2022, Manning.
15. “*Natural Language Processing for Corpus Linguistics*”; J. Dunn; 2022, Cambridge Elements. ISSN: 2632-8097 [online].
16. “*Artificial Intelligence: A modern approach*”; S. Russel, P. Norvig; 2022, Pearson. ISBN 13: 978-1-292-40113-3.
17. “*Natural Language Processing with Transformers: Building Language Applications with Hugging Face*”; L. Tunstall, L. von Werra, T. Wolf; 2022, O'Reilly. ISBN: 978-1-098-10324-8.
18. “*Transformers for Machine Learning: a deep dive*”; U. Kamath, K. L. Graham, W. Emara; 2022, CRC Press. ISBN: 978-0-367-77165-2.
19. “*Artificial Intelligence in Education*”; M. N. O. Sadiku, S. M. Musa, U. C. Chukwu; 2022, iUniverse. ISBN: 978-1-6632-3001-0.
20. “*Introducción a la Inteligencia Artificial: la tecnología que nos cambiará para siempre*”; J. M. Girón Sierra; 2023, Almuzara. ISBN: 978-84-18414-57-2.



"Tecnologías Inteligentes" no es simplemente un libro sobre inteligencia artificial; es el resultado de mi pasión por enseñar, investigar y explorar los límites de lo posible con la tecnología. A lo largo de estas páginas, he intentado plasmar no solo el conocimiento técnico que he adquirido durante más de dos décadas, sino también la convicción de que la inteligencia artificial es una herramienta transformadora, accesible y profundamente humana.

El libro se estructura en torno a las tres grandes tecnologías que sustentan la IA moderna: visión artificial, procesamiento del lenguaje natural y aprendizaje automático. En los primeros capítulos, establezco las bases teóricas de la inteligencia artificial y su historia, desmitificando conceptos y aclarando qué es y qué no es IA. Desde ahí, guío al lector a través de aplicaciones prácticas: desde el reconocimiento de patrones visuales y la creación de modelos predictivos, hasta cómo aplicar algoritmos para resolver problemas cotidianos. También dedico una sección especial a la ética, porque no podemos hablar del impacto de estas tecnologías sin reflexionar sobre los riesgos y las responsabilidades que conllevan.

He diseñado esta obra para que no intimide, sino que inspire. Cada ejemplo práctico, cada ejercicio en los *notebooks* que la acompañan, está pensado para que el lector pase del asombro a la acción, del concepto a la aplicación. Si alguna vez has sentido curiosidad por cómo funciona un modelo de lenguaje como los que usamos hoy en día o por cómo una cámara puede interpretar el mundo que ve, este libro te llevará de la mano para descubrirlo.

Escribir este libro ha sido una experiencia profundamente enriquecedora, pero también un ejercicio de reflexión. Mi objetivo no es solo explicar qué es la inteligencia artificial, sino que el lector aprenda a utilizarla para resolver problemas reales, mejorar su entorno y, por qué no, disfrutar del proceso. Si logro que te acerques a estas tecnologías con confianza y las veas como aliadas, habré alcanzado mi propósito.

Un cordial saludo.

Juan Francisco Puentes Calvo

Catedrático de Formación Profesional, especializado en Inteligencia Artificial

