UNIVERSIDAD
DE GRANADA

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación y Facultad de Ciencias

Doble Grado en Ingeniería Informática y Matemáticas

TRABAJO DE FIN DE GRADO

Análisis teórico y empírico del Deep Double Descent

Presentado por:
Juan Antonio Ruiz Arévalo

Tutores:
Francisco Javier Merí de la Maza
Pablo Mesejo Santiago

Curso académico 2024-2025

Análisis teórico y empírico del Deep Double Descent

Juan Antonio Ruiz Arévalo

Juan Antonio Ruiz Arévalo *Análisis teórico y empírico del Deep Double Descent*.
Trabajo de fin de Grado. Curso académico 2024-2025.

Responsable de tutorización

Francisco Javier Merí de la Maza
Departamento de Análisis Matemático

Pablo Mesejo Santiago
Departamento de Ciencias de la Computación e Inteligencia Artificial

Doble Grado en Ingeniería Informática y Matemáticas

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación y Facultad de Ciencias

Universidad de Granada

Declaración de originalidad

D. Juan Antonio Ruiz Arévalo

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2024-2025, es original, entendido esto en el sentido de que no he utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 23 de febrero de 2025

Fdo: Juan Antonio Ruiz Arévalo

Dedicatoria

Agradecimientos

Agradecimientos

Resumen

Summary

Índice general

Índice de figuras

Índice de tablas

Introducción

En los últimos años, el fenómeno del *Deep Double Descent* ha surgido como un importante campo de interés en el aprendizaje automático. Mientras que la sabiduría tradicional del aprendizaje sugiere que, a medida que aumenta la complejidad del modelo, el error fuera de la muestra disminuye inicialmente hasta alcanzar un mínimo y, a continuación, aumenta de manera monótona debido al sobreajuste, formando la tradicional curva con forma de "U". Este concepto tradicional del aprendizaje automático, conocido como equilibrio entre sesgo y varianza difiere de las recientes observaciones obtenidas, que se cuestionan este punto de vista, especialmente en modelos de aprendizaje profundo, en los que puede producirse una segunda disminución del error fuera de la muestra y alcanzar un nuevo mínimo, formando una nueva curva en la gráfica del error de generalización que presenta dos descensos. Este novedoso descubrimiento pone en tela de juicio la sabiduría clásica sobre el tema y proporciona nuevas perspectivas a la hora de crear y entrenar los modelos.

Este trabajo de fin de grado se centra en explorar el concepto del *Deep Double Descent*, sus fundamentos teóricos y sus implicaciones para el aprendizaje automático moderno. Un área clave de interés es cómo este fenómeno se manifiesta en redes neuronales profundas, conocidas por su enorme complejidad en cuanto a número de párametros se refiere y su potencial de sobreajuste. A pesar de que los modelos de aprendizaje profundo han tenido gran éxito en diversas aplicaciones, como la visión por computador o el procesamiento de lenguaje natural, la curva del doble descenso aporta nuevos conocimientos sobre cómo estos modelos pueden llegar a generalizar en un régimen que ha sido vagamente estudiado y explorado: el régimen sobreparametrizado.

La idea clave detras del fénomeno es la existencia de dos zonas de actuación del modelo claramente diferenciadas, la zona infraparametrizada y la zona sobreparametrizada. Sin embargo, formalizar esta idea es significativamente complejo, dado que los modelos profundos funcionan como "cajas negras", lo que hace que, a medida que aumenta su complejidad, resulte cada vez más difícil interpretar y obtener resultados precisos sobre su funcionamiento interno.

Aunque cada vez hay más estudios abordando el fenómeno, muchos de ellos se centran en una perspectiva empírica del mismo, sin ofrecer una base teórica suficiente, mientras que otros sistematizan conceptos del fenómeno sin llegar a conclusiones prácticas. Con este trabajo buscamos cerrar la brecha entre la teoría y la práctica unificando explicaciones teóricas suficientemente rigurosas con ejemplos empíricos del mundo real con el objetivo de ofrecer una comprensión lo más completa posible del fenómeno.

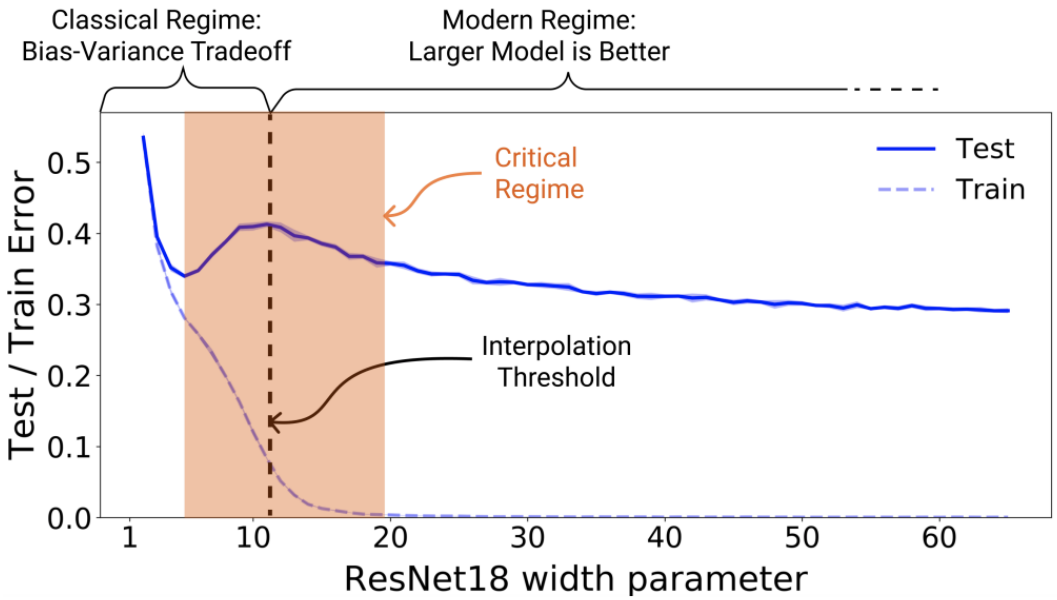

Figura 1.: Ejemplo de doble descenso en ResNet18 [NKB+19]. La imagen muestra el error de entrenamiento (train error) y de generalización (test error) para la arquitectura ResNet18 con diferente capacidad (número de parámetros). En ella, observamos las tres regiones claramente diferenciadas, así como el máximo del error de generalización correspondiente al umbral de interpolación y los dos descensos de la curva de dicho error.

1. Definición del problema

El término *doble descenso profundo (deep double descent, [BHMM19])* describe la forma que toma la curva del error de generalización (error fuera de la muestra) de un modelo de aprendizaje como función de la capacidad del mismo. De manera intuitiva, podemos distinguir 3 zonas o regiones diferenciadas en dicha curva:

- **Región clásica (infraparametrizada):** En esta región, el modelo no es capaz de capturar toda la complejidad subyacente de la distribución de los datos debido a su baja capacidad. Como resultado, el error de generalización disminuirá inicialmente, ligado al hecho de que el modelo aprende de los datos de entrenamiento. Sin embargo, llegado un momento, el error de generalización comenzará a aumentar de manera progresiva (véase Figura 1), dando lugar a la clásica curva en forma de "U" y ligado al hecho (tradicional) de que el modelo, en lugar de seguir aprendiendo de los datos y obtener patrones de los mismos, memoriza los datos de entrenamiento.
- **Región moderna (sobreparametrizada):** En esta región, el modelo tiene una capacidad mayor que la necesaria para ajustar los datos de entrenamiento, es decir, dispone de suficientes herramientas (parámetros) para ajustar cada uno de los datos de entrenamiento. Contrariamente a lo que se esperaría de forma clásica, el error de generalización no necesariamente aumenta en este región y, bajo ciertas condiciones, dicho error puede reducirse nuevamente, produciendo un nuevo descenso de la curva del error de

generalización que se conoce por *doble descenso*.

- **Región crítica:** Esta región marca la transición entre la región infraparametrizada y la región sobreparametrizada y engloba zonas de ambas regiones. Dentro de ella, se encuentra el llamado **umbral de interpolación**, que corresponde al punto donde el modelo tiene justo la capacidad suficiente para ajustar perfectamente los datos de entrenamiento (mismos parámetros que datos). En este punto crítico, el error de generalización alcanzará su máximo.

Este doble descenso puede llegar a suponer la obtención de un nuevo mínimo en la curva del error de generalización, es decir, la obtención de modelos cuyas predicciones sean aún mejores. Sin embargo, se abre la puerta a la investigación del por qué ocurre este novedoso fenómeno, además de que tendremos que replantearnos algunas respuestas, tradicionalmente correctas, ante preguntas clásicas del aprendizaje profundo.

2. Motivación

En el ámbito del aprendizaje automático, es de cultura general conocer la existencia de una brecha entre el desarrollo empírico y la fundamentación teórica subyacente. Los modelos modernos, en particular las redes neuronales profundas, han demostrado resultados sorprendentes [Sej20, HT20] desde la generación de imágenes realistas mediante redes generativas hasta el procesamiento de lenguaje natural (*Natural Language Processing, NLP*). Sin embargo, estos resultados se logran sin una comprensión teórica completa [BD09], especialmente debido al hecho de la facilidad de obtener resultados suficientemente decentes a través de la experimentación, sin la preocupación de preguntarnos si lo que realmente estamos probando es verdaderamente lo más eficiente o adecuado para alcanzar los mejores resultados posibles.

Esta capacidad para obtener resultados que consideramos satisfactorios ha llevado a un enfoque predominantemente práctico, donde los avances se producen de manera más rápida a través del ensayo y error que mediante modelos matemáticos bien fundamentados, lo que impide comprender, desde una perspectiva teórica, tanto la eficacia como las verdaderas limitaciones que estos modelos podrían alcanzar. Este fenómeno plantea cuestiones sobre la sostenibilidad de este enfoque práctico y la necesidad de desarrollar un marco teórico que permita anticipar y guiar estos avances en lugar de simplemente reaccionar ante ellos.

Debido a esta desconexión entre los avances teóricos y empíricos, han comenzado a surgir discrepancias y fenómenos no esperados en los resultados modernos, en particular en los que a modelos profundos se refiere. Es aquí donde se enmarca el concepto del *Deep Double Descent*, fenómeno por el cual, a través de resultados prácticos, se ha corroborado como la práctica continúa superando las expectativas teóricas tradicionales y por el que hoy en día surgen numerosas líneas de estudio tratando de dar una explicación al respecto, resaltando como la teoría subyacente aún está vagamente desarrollada.

De igual manera, este fenómeno tiene una relevancia crucial, ya que podría transformar la forma en que se diseñan y optimizan los modelos profundos. Tradicionalmente, el enfoque

clásico sugiere que, a medida que se aumenta la complejidad del modelo, este tendía a sobreajustarse a los datos, lo que limitaba su capacidad de generalización, lo que impulsaba a optar por modelos más simples. Sin embargo, con este fenómeno, se ha demostrado que, más allá del sobreajuste, los modelos no solo dejan de aumentar su capacidad de generalización, sino que incluso mejoran, alcanzando una generalización aún mejor a la inicial. Este descubrimiento sugiere que, siguiendo este enfoque, podríamos centrarnos en desarrollar modelos más complejos sin necesidad de aplicar técnicas de regularización para evitar el sobreajuste, lo que nos permitiría obtener mejores resultados.

En conclusión, el fenómeno del *Deep Double Descent* representa un cambio de paradigma, digno de estudio, en el aprendizaje automático. Comprender los principios subyacentes no solo nos permitiría avanzar en nuestra comprensión teórica del aprendizaje automático, sino que también ayudaría a ir cerrando la brecha entre la teoría y la práctica. De hecho, si se logra desarrollar un marco teórico sólido, sería posible guiar la creación y optimización de modelos, haciendo que los avances prácticos sean más predecibles.

3. Objetivos

El objetivo principal de este TFG radica en tratar de ofrecer una explicación detallada y estructurada del novedoso fenómeno *Deep Double Descent*. Este estudio se centrará en proporcionar una visión actualizada y rigurosa de sus fundamentos teóricos, implicaciones prácticas y relevancia en el desarrollo de modelos modernos, asegurando que el contenido se mantenga en concordancia con los avances más recientes en la investigación.

Para alcanzar este objetivo, se han definido dos líneas de trabajo interrelacionadas: una orientada al desarrollo **matemático** y otra enfocada en la parte **informática**. Ambas líneas se desarrollan de manera conjunta y complementaria, de modo que los avances teóricos guían y fundamentan la implementación práctica, mientras que los resultados experimentales permiten validar y enriquecer la comprensión teórica. A su vez, cada una de estas líneas de trabajo se descompone en una serie de objetivos parciales que, en conjunto, guían el desarrollo del proyecto.

3.1. Objetivo matemático

El objetivo fundamental para la parte matemática consiste en profundizar en la comprensión teórica del *Deep Double Descent* a través del estudio detallado de sus fundamentos matemáticos, explorando las relaciones con conceptos clásicos como el equilibrio sesgo-varianza y su posible conexión con la teoría de la aproximación no lineal. Con el fin de abordar de forma sistemática las distintas fases de este análisis, el presente objetivo se descompondrá en los siguientes objetivos parciales:

- Realizar un análisis exhaustivo y detallado del estado del arte, revisando las principales teorías, descubrimientos y avances matemáticos relacionados con el fenómeno.
- Explicar de manera detallada la sabiduría clásica relacionada con el tema, analizando las teorías y enfoques tradicionales que prevalecen en la literatura, estableciendo las

bases necesarias para entender el fenómeno.

- Investigar y analizar el fenómeno del *Deep Double Descent*, proporcionando una explicación detallada de sus fundamentos y explorando en profundidad los hallazgos más relevantes de la literatura científica sobre el tema.
- Adentrarnos en la teoría de la aproximación no lineal con el propósito de identificar y analizar posibles analogías con el fenómeno, explorando cómo los enfoques no lineales pueden ofrecer una comprensión más profunda y enriquecedora de las dinámicas que subyacen a este fenómeno.

3.2. Objetivo informático

El objetivo esencial para la parte informática consiste en llevar a cabo la constatación experimental del fenómeno mediante la implementación y análisis de diversas arquitecturas que permitan validar y estudiar empíricamente su comportamiento. Esta parte experimental busca no solo ilustrar la aparición del fenómeno en distintos escenarios y arquitecturas, sino también corroborar y complementar los resultados obtenidos en la parte matemática, estableciendo así una conexión sólida entre la teoría y la práctica. Con el fin de abordar de forma sistemática las distintas fases de este análisis, el presente objetivo se descompondrá, al igual que el objetivo de la parte matemática, en los siguientes objetivos parciales:

- Realizar un análisis exhaustivo y detallado de los casos prácticos y representaciones experimentales en los que se ha manifestado el fenómeno, revisando los principales comportamientos y patrones.
- Presentar resultados experimentales que validen los desarrollos teóricos realizados en la parte matemática, demostrando la coherencia con las predicciones teóricas.
- Desarrollar un análisis experimental que respalde la aparición y las características del fenómeno, proporcionando evidencias prácticas que contribuyan a una comprensión más profunda de su comportamiento en diferentes modelos y escenarios en los que puede aparecer.

4. Planificación del proyecto

Parte I.

Fundamentos Teóricos

1. Teoría de la Probabilidad

En este capítulo se presentarán definiciones y resultados fundamentales de la teoría de la probabilidad y la estadística, con el propósito de introducir conceptos clave que faciliten la comprensión del fenómeno y que utilizaremos a lo largo del desarrollo de gran parte del trabajo. Las fuentes principales utilizadas a lo largo de este capítulo son extractos de [Dem14] y [Kni09].

1.1. Espacios de probabilidad y σ-álgebras

Para establecer la base teórica, consideraremos un conjunto arbitrario Ω, al que nos referiremos como *espacio muestral* y que representa el conjunto de todos los posibles resultados al realizar un experimento. Asimismo, llamaremos *suceso* a cualquier subconjunto de Ω.

Definición 1.1 (*σ-álgebra*)**.** Un conjunto $\mathcal{A}$ de subconjuntos de Ω ($\mathcal{A} \subseteq \mathcal{P}(\mathcal{A})$) se dirá que es una *$\sigma$-álgebra* si verifica las siguientes propiedades:

- $\Omega \in \mathcal{A}$,
- Si $A \in \mathcal{A}$, entonces $A^c = \Omega \setminus \mathcal{A} \in \mathcal{A}$ (A es cerrado bajo complementarios),
- Si $A_n \in \mathcal{A}$, entonces $\bigcup_{n\in\mathbb{N}} A_n \in \mathcal{A}$ (A es cerrado bajo uniones finitas)

Es fácil comprobar que $\mathcal{A}$ es cerrado bajo intersecciones finitas y que, además, la intersección de σ-álgebras es una σ-álgebra.

Definición 1.2 (*σ-álgebra de Borel*)**.** Para cada conjunto $\mathcal{C}$ de subconjuntos de Ω, se define $\sigma(\mathcal{C})$ como la menor σ-álgebra $\mathcal{A}$ que contiene a C. La σ-álgebra $\mathcal{A}$ es la intersección de todas las σ-álgebras que contienen a C y, por tanto, es una σ-álgebra.

Si $(E, \mathcal{O})$ es un espacio topológico, donde O es el conjunto formado por los conjuntos abiertos en E, entonces $\sigma(\mathcal{O})$ es llamada la **σ-álgebra de Borel** del espacio topológico.

Llamaremos *espacio de medida* al conjunto $(\Omega, \mathcal{A})$ donde $\mathcal{A}$ es una σ-álgebra en Ω.

Definición 1.3 (*Medida de probabilidad*)**.** Dado un espacio de medida $(\Omega, \mathcal{A})$, una función $P : \mathcal{A} \to \mathbb{R}$ se llamará *medida de probabilidad* si cumple las siguientes tres propiedades (conocidas como **axiomas de Kolmogorov** [Kol56]):

- $P[A] \geq 0$ para todo $A \in \mathcal{A}$,
- $P[\Omega] = 1$,

- P es σ-aditiva, es decir, si $A_n \in \mathcal{A}$ con $n \in \mathbb{N}$ son conjuntos disjuntos dos a dos, entonces
$$P\left[\bigcup_{n\in\mathbb{N}} A_n\right] = \sum_{n\in\mathbb{N}} P[A_n].$$

La primera condición nos asegura la no negatividad de la probabilidad, es decir, la probabilidad nunca será inferior a 0. A su vez, la segunda condición nos establece que la probabilidad del espacio muestral completo (Ω) debe ser igual a 1, es decir, refleja que uno de los eventos posibles siempre ocurrirá, conocido como *suceso seguro*.

Observación 1.4. Dado que una medida de probabilidad es, por definición, una *medida*, se sigue de manera inmediata que $P(\emptyset) = 0$. Además, al conjunto $\emptyset$ se le suele denotar como *evento imposible*.

Corolario 1.5. *Algunas propiedades básicas de la medida de probabilidad (*P*) que se siguen de la propia definición son las siguientes:*

1. *Si* $A, B \in \mathcal{A}$ *y* $A \subset B$, *entonces* $P[A] \leq P[B]$.
2. $P[A^c] = 1 - P[A]$, *para todo* $A \in \mathcal{A}$.
3. $0 \leq P[A] \leq 1$ *para todo* $A \in \mathcal{A}$.

Observación 1.6. Existen distintas formas de construir los axiomas para un espacio de probabilidad. Por ejemplo, se podrían sustituir las primeras dos propiedades de la definición de medida de probabilidad por las últimas dos propiedades enunciadas en el corolario anterior.

Definición 1.7 (*Espacio de probabilidad*)**.** Dado un espacio de medida $(\Omega, \mathcal{A})$, llamaremos *espacio de probabilidad* a la tripleta $(\Omega, \mathcal{A}, P)$, donde P es una medida de probabilidad.

1.2. Variables aleatorias y esperanza

Las variables aleatorias son funciones numéricas que asignan un valor numérico a cada posible resultado de un experimento aleatorio $w \in \Omega$. De manera intuitiva, una variable aleatoria puede verse como una cantidad numérica cuyo valor no es fijo y que puede tomar distintos valores, por lo que es necesario definir una distribución de probabilidad que asocie probabilidades a los distintos valores que pueda tomar la variable aleatoria.

Definición 1.8 (*Función medible*)**.** Una función $X : (\Omega_1, \mathcal{A}) \to (\Omega_2, \mathcal{B})$ se dice *medible* si
$$X^{-1}(B) \in \mathcal{A} \quad \forall B \in \mathcal{B}$$
donde el conjunto $X^{-1}(B)$ se encuentra formado por todos los puntos $x \in \Omega$ para los cuales $X(x) \in B$.

Definición 1.9 (*Variable aleatoria*)**.** Dado un espacio de probabilidad $(\Omega_1, \mathcal{A}, \mathrm{P})$ y un espacio medible $(\Omega_2, \mathcal{B})$, decimos que $X : (\Omega_1, \mathcal{A}, \mathrm{P}) \to (\Omega_2, \mathcal{B})$ es una *variable aleatoria* si X es una función medible.

Además, si el espacio medible de llegada es n-dimensional, entonces la variable aleatoria X es llamada *vector aleatorio* y lo denotaremos por $X = (X_1, X_2, \ldots, X_n)$ donde cada componente X_i con $i \in \{1, 2, \ldots, n\}$ es una variable aleatoria.

Observación 1.10. En la mayoría de usos prácticos se tiene que el espacio medible de llegada más común es $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, donde $\mathcal{B}(\mathbb{R})$ denota la σ-álgebra de Borel en $\mathbb{R}$.

Definición 1.11. Diremos que una variable aleatoria es *discreta* si esta toma un número finito o numerable de valores en el espacio de llegada. Por otra parte, si la variable aleatoria toma un número infinito o no numerable de valores, diremos que es una variable aleatoria *continua*.

Ejemplo 1.12. Como ejemplo sencillo podemos considerar los posibles resultados obtenidos al lanzar un dado de seis caras, es decir, $w \in \{1, 2, 3, 4, 5, 6\}$ y podemos definir la variable aleatoria discreta que asigna el valor de la cara superior del dado cuando se lanza. En este caso, la variable aleatoria se define como:

$$X(w) = w \quad \textit{para } w \in \{1, 2, 3, 4, 5, 6\}.$$

Definición 1.13 (Función de distribución)**.** La función de distribución (acumulada) de una variable aleatoria X es una función $F_X : \mathbb{R} \to [0, 1]$ dada por

$$F_X(\alpha) = P[\{w : X(w) \leq \alpha\}] \quad \forall \alpha \in \mathbb{R}.$$

Proposición 1.14. *La función de distribución F de una variable aleatoria X cumple las siguientes propiedades:*

1. *F es monótona no decreciente.*
2. $\lim_{x \to \infty} F(x) = 1$ *y* $\lim_{x \to -\infty} F(x) = 0$.
3. *F es continua por la derecha, es decir,* $\lim_{y \to x^+} F(y) = F(x)$.

Definición 1.15 (Función de probabilidad)**.** Sea X una variable aleatoria discreta, llamaremos *función (masa) de probabilidad*, a la función que asigna la probabilidad de que la variable aleatoria tome un valor en particular, es decir:

$$p_X(x) = P[X = x] \quad \textit{donde } x \in \{x_1, \ldots, x_n\}.$$

Las probabilidades asociadas con todos los posibles resultados del experimento deben ser no negativas y sumar 1, es decir $\sum_x p_X(x) = 1$ y, además, $p_X(x) \geq 0$.

Notemos que, el concepto de función de probabilidad, solo tiene sentido al hablar de variables aleatorias discretas. Para variables aleatorias continuas, el concepto análogo es el de función de densidad, donde deberemos integrar para obtener la probabilidad, pues la probabilidad asociada a un único punto en un intervalo es cero.

Definición 1.16 (Función de densidad)**.** Se dice que una función f_X integrable de Lebesgue, no negativa en casi todas partes es la *función de densidad* de una variable aleatoria continua X si su función de distribución puede ser expresada como

$$F_X(\alpha) = \int_{-\infty}^{\alpha} f_X(x)\, dx \quad \forall \alpha \in \mathbb{R}.$$

Notemos que la función de densidad, de manera análoga a la función de probabilidad, cumple $f_X(x) \geq 0$ y $\int_{-\infty}^{\infty} f_X(x) = 1$.

Definición 1.17 (Esperanza de una variable aleatoria)**.** Sea X una variable aleatoria en el espacio de probabilidad $(\Omega, \mathcal{A}, \mathrm{P})$. Definimos la *esperanza o valor esperado* de X, denotado por $E[X]$ como la integral de Lebesgue siguiente:

$$\mathbb{E}[X] = \int_{\Omega} X(w) dP[w].$$

Para vectores aleatorios, su esperanza viene definida componente a componente:

$$\mathbb{E}[(X_1, \ldots, X_n)] = (\mathbb{E}[X_1], \ldots, \mathbb{E}[X_n]).$$

Observación 1.18. Si X es una variable aleatoria discreta con función de probabilidad $\mathrm{P}[X = x_i]$ con $i \in \{1, 2, \ldots, n\}$, su esperanza viene definida por:

$$\mathbb{E}[X] = \sum_{i=1}^{n} x_i \, \mathrm{P}[X = x_i].$$

donde x_i denota cada posible resultado del experimento.

Observación 1.19. Si X es una variable aleatoria continua con función de densidad $f_X(x)$, su esperanza viene definida por:

$$\mathbb{E}[X] = \int_{\mathbb{R}} x \, f_X(x)\, dx.$$

1.2.1. Probabilidad condicional

En esta sección introduciremos la noción clásica de *probabilidad condicional* de sucesos, ligada al supuesto de conocer la probabilidad de un cierto suceso bajo la condición de que ocurra

otro suceso. De igual manera, se introducirán resultados necesarios para el desarrollo del trabajo, tales como el teorema de Bayes.

Definición 1.20. Dados dos sucesos $A, B \in \mathcal{A}$ con $P[B] > 0$, definimos la *probabilidad condicional* de A con respecto a B de la siguiente forma:

$$P[A|B] = \frac{P[A \cap B]}{P[B]}.$$

Definición 1.21. Un conjunto finito $\{A_1, \ldots, A_n\} \subset \mathcal{A}$ se denominará *partición finita* de Ω si cumple:

1. $\bigcup_{i=1}^{n} A_i = \Omega$.
2. $A_i \cap A_j = \emptyset$ para todo $i \neq j$.

Una partición finita cubre todo el espacio con un número finito de conjuntos (sucesos) disjuntos dos a dos.

Teorema 1.22 (Probabilidad total)**.** *Sean $\{A_1, \ldots, A_n\}$ una partición finita de $\mathcal{A}$ y $B \in \mathcal{A}$ un suceso cualquiera del que se conocen las probabilidades condicionales $P[B|A_i]$ $\forall i \in \{1, \ldots, n\}$, entonces la probabilidad del suceso B viene dada por la siguiente expresión:*

$$P[B] = \sum_{i=1}^{n} P[B|A_i]P[A_i].$$

Demostración. Partimos de una partición finita $\{A_1, \ldots, A_n\}$ de $\mathcal{A}$ y de un suceso $B \in \mathcal{A}$. Usando la primera propiedad de la Definición 1.21, podemos expresar el suceso B de la siguiente forma:

$$B = (B \cap A_1) \cup (B \cap A_2) \cup \ldots \cup (B \cap A_n).$$

Usando la segunda propiedad de la Definición 1.21, sabemos que $A_i \cap A_j = \emptyset, i \neq j$. Por tanto, obtenemos que los conjuntos $(B \cap A_i)$, $i \in \{1, \ldots, n\}$ son, también, disjuntos dos a dos.

Por consiguiente, podemos expresar la probabilidad del suceso B como sigue:

$$P[B] = P[B \cap A_1] + P[B \cap A_2] + \cdots + P[B \cap A_n].$$

Finalmente, usando la Definición 1.20, obtenemos que

$$P[A_i \cap B] = P[A_i|B]P[B], \ \forall\, i \in \{1, \ldots, n\}.$$

Por tanto, obtenemos la expresión buscada:

$$\begin{aligned} P[B] &= P[B \cap A_1] + P[B \cap A_2] + \cdots + P[B \cap A_n] \\ &= P[B|A_1]P[A_1] + P[B|A_2]P[A_2] + \cdots + P[B|A_n]P[A_n] \\ &= \sum_{i=1}^{n} P[B|A_i]P[A_i]. \end{aligned}$$

□

Llegados a este punto estamos en las condiciones necesarias de introducir un teorema fundamental que nos permite calcular probabilidades condicionales. Este resultado vincula la probabilidad de un suceso A dado otro suceso B ($P[A|B]$) con la probabilidad del suceso B dado el suceso A ($P[B|A]$).

Teorema 1.23 (Regla de Bayes)**.** *Dada una partición finita $\{A_1, \ldots, A_n\}$ de $\mathcal{A}$ y un suceso $B \in \mathcal{A}$ con $P[B] > 0$, se verifica:*

$$P[A_i|B] = \frac{P[B|A_i]P[A_i]}{\sum_{i=1}^{n} P[B|A_i]P[A_i]}.$$

Demostración. En primer lugar, de la Definición 1.20, sabemos que $P[A_i|B] = \frac{P[A_i \cap B]}{P[B]}$. Además, del Teorema 1.22, conocemos que $P[B] = \sum_{i=1}^{n} P[B|A_i]P[A_i]$.

Por otra parte, aplicando la Definición 1.20 de la siguiente manera:

$$P[B|A_i] = \frac{P[B \cap A_i]}{P[A_i]}.$$

Ahora, despejando, obtenemos $P[A_i \cap B] = P[B \cap A_i] = P[B|A_i]P[A_i]$. Finalmente, combinado ambos resultados alcanzamos la conclusión buscada:

$$P[A_i|B] = \frac{P[A_i \cap B]}{P[B]} = \frac{P[B|A_i]P[A_i]}{\sum_{i=1}^{n} P[B|A_i]P[A_i]}.$$

□

1.2.2. Independencia de variables aleatorias

En esta sección nos centraremos en explicar el concepto de independencia en el contexto de las variables aleatorias. De manera intuitiva, el concepto de independencia, como su nombre indica, va ligado al hecho de que el conocimiento que poseamos de una de las variables no proporciona información adicional sobre el conocimiento de la otra. Para ello, también expandiremos el concepto de función de distribución para el caso de más de una variable

aleatoria (vector aleatorio).

Definición 1.24 (Función de distribución conjunta)**.** Sean $X_1, \ldots, X_n$ variables aleatorias definidas sobre el mismo espacio de probabilidad $(\Omega, \mathcal{A}, \mathrm{P})$, la *función de distribución conjunta* es una función $F_{X_1,\ldots,X_n} : \mathbb{R} \to [0,1]$ dada por

$$F_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = P[X_1 \leq x_1, \ldots, X_n \leq x_n], \quad x_1, \ldots, x_n \in \mathbb{R}.$$

Si interpretamos las n variables aleatorias como un vector aleatorio $X = (X_1, \ldots, X_n)$, podemos simplificar la notación de la siguiente forma:

$$F_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = F_X(x_1, \ldots, x_n).$$

De la misma forma, podemos extender las definiciones de *función de probabilidad* y *función de densidad* para el caso en el que dispongamos de más de una variable aleatoria.

En primer lugar, comenzaremos definiendo y demostrando el resultado de la *regla de la cadena* para probabilidades que nos será de gran utilidad para trabajar con las siguientes definiciones.

Teorema 1.25 (Regla de la cadena)**.** *Sea $(\Omega, \mathcal{A}, P)$ un espacio de probabilidad y $\{A_1, \ldots, A_n\} \in \mathcal{A}$ una serie de sucesos. Entonces, se verifica*

$$P[A_1 \cap A_2 \cap \ \ldots \ \cap A_n] = P[A_1]P[A_2|A_1] \cdots \ P[A_1] \prod_{i=2}^{n} P[A_i|A_1 \ \cap \cdots \ \cap A_{i-1}].$$

Demostración. La demostración se realiza de manera sencilla por recursión teniendo en cuenta que en el primer caso se hace uso de la Definición 1.20:

$$P[A_1 \cap A_2] = P[A_1]P[A_2|A_1].$$

□

Definición 1.26 (Función de probabilidad conjunta)**.** Sean $X_1, \ldots, X_n$ variables aleatorias discretas. La *función (masa) de probabilidad conjunta* de dichas variables viene dada por

$$p_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = P(X_1 = x_1, \ldots, X_n = x_n), \quad (x_1, \ldots, x_n) \in \mathbb{R}^n.$$

Equivalentemente, la función de probabilidad conjunta puede ser expresada de la siguiente forma:

$$p_{X_1,\ldots,X_n}(x_1,\ldots,x_n) = P(X_1 = x_1)\cdot P(X_2 = x_2 \mid X_1 = x_1)\cdot \\ \cdot P(X_3 = x_3 \mid X_1 = x_1, X_2 = x_2)\cdots \\ \cdot P(X_n = x_n \mid X_1 = x_1, X_2 = x_2, \ldots, X_{n-1} = x_{n-1})$$

donde se utiliza la definición de probabilidad condicionada y la regla de la cadena comentada en el teorema anterior.

Además, dado que estamos trabajando con probabilidades, se verifica que la suma total debe ser igual a 1, es decir

$$\sum_i \sum_j \ldots \sum_k P[X_1 = x_{1i}, X_2 = x_{2j}, \ldots, X_n = x_{nk}]$$

donde los distintos índices $(i, j, \ldots, k)$ recorren todos los posibles resultados de cada variable aleatoria.

Definición 1.27 (Función de densidad conjunta)**.** Sean $X_1, \ldots, X_n$ variables aleatorias continuas. Se dice que una función $f_{X_1,\ldots,X_n}$ integrable, no negativa de Lebesgue en casi todas partes es la *función de densidad conjunta* de las variables aleatorias continuas $X_1, \ldots, X_n$ si la función de distibución conjunta puede ser expresada como

$$F_{X_1,\ldots,X_n}(x_1,\ldots,x_n) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f_{X_1,\ldots,X_n}(x_1,\ldots,x_n)\, dx_1 \cdots dx_n \\ \forall (x_1,\ldots,x_n) \in \mathbb{R}^n.$$

Una manera análoga de expresar la función de densidad conjunta es la siguiente:

$$f_{X1,\ldots,X_n}(x_1,\ldots,x_n) = \frac{\partial^n F_{X_1,\ldots,X_n}(x_1,\ldots,x_n)}{\partial x_1 \ldots \partial x_n}.$$

De manera análoga a la definición anterior y dado que estamos trabajando con distribuciones de probabilidad, se verifica que

$$\int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f_{X1,\ldots,X_n}(x_1,\ldots,x_n)\, dx_1, \ldots, dx_n = 1.$$

Una vez conocemos las funciones conjuntas de una serie de variables aleatorias es posible determinar la probabilidad de un suceso sin considerar la influencia de otras variables por medio de las funciones marginales.

Definición 1.28 (Función de distribución marginal)**.** Sean $X_1, \ldots, X_n$ variables aleatorias. Se define la *función de probabilidad marginal* de la variable aleatoria $X_i,\ i \in \{1, \ldots, n\}$ de la siguiente manera:

$$\forall i = 1, \dots, n \qquad F_{X_i}(x_i) = F_{X_1,\dots,X_n}(+\infty, \dots, x_i, \dots, +\infty) \quad \forall x_i \in \mathbb{R}.$$

Definición 1.29 (Función de probabilidad marginal)**.** Sean $X_1, \dots, X_n$ variables aleatorias discretas. Se define la *función (masa) de probabilidad marginal* de la variable aleatoria X_i, $i \in \{1, \dots, n\}$ de la siguiente manera:

$$p_{X_i}(x_i) = \sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_n} p_{X_1,\dots,X_n}(x_1, \dots, x_n), \quad (x_1, \dots, x_n) \in \mathbb{R}.$$

Definición 1.30 (Función de densidad marginal)**.** Sean $X_1, \dots, X_n$ variables aleatorias continuas. Se define la *función de densidad marginal* de la variable aleatoria X_i, $i \in \{1, \dots, n\}$ de la siguiente manera:

$$f_{X_i}(x_i) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f_{X_1,\dots,X_n}(x_1, \dots, x_n)\, dx_1, \dots, dx_{i-1}, dx_{i+1}, \dots, dx_n \quad \forall x_i \in \mathbb{R}.$$

Definición 1.31. Sean $X_1, \dots, X_n$ variables aleatorias definidas sobre el mismo espacio de probabilidad $(\Omega, \mathcal{A}, \mathrm{P})$, con funciones de distribución $F_{X_1}, \dots, F_{X_n}$ respectivamente. Decimos que las variables aleatorias son idénticamente distribuidas (id) si

$$F_{X_1}(x) = F_{X_j}(x), \ \forall j \in \{1, \dots, n\} \ y \ \forall x \in \mathbb{R}.$$

Definición 1.32. Sean $X_1, \dots, X_n$ variables aleatorias definidas sobre el mismo espacio de probabilidad $(\Omega, \mathcal{A}, \mathrm{P})$, con funciones de distribución $F_{X_1}, \dots, F_{X_n}$ respectivamente y función de distribución conjunta F_X. Decimos que las variables aleatorias son independientes si

$$F_X(x_1, \dots, x_n) = F_{X_1}(x_1) \cdots F_{X_n}(x_n), \ \forall x_1, \dots, x_n \in \mathbb{R}.$$

Observación 1.33. De la definición anterior se deduce que dos variables aleatorias son independientes cuando:

- En el caso discreto, su función (masa) de probabilidad conjunta es igual al producto de las funciones (masa) de probabilidad marginales de cada variable aleatoria. Esto es, si X_1, X_2 son dos variables aleatorias discretas, entonces $p_{X_1,X_2}(x_1, x_2) = p_{X_1}(x_1) p_{X_2}(x_2)$, $x_1, x_2 \in \mathbb{R}$.
- En el caso continuo, su función de densidad conjunta es igual al producto de las funciones de densidad marginales de cada variable aleatoria. Esto es, si X_1, X_2 son dos

variables aleatorias continuas, entonces $f_{X_1,X_2}(x_1,x_2) = f_{X_1}(x_1)f_{X_2}(x_2),\ x_1,x_2 \in \mathbb{R}$.

Definición 1.34. Las variables aleatorias que cumplan, de manera simultánea, la Definición 1.31 y la Definición 1.32 las llamaremos *variables aleatorias independientes e idénticamente distribuidas* y se denotarán como *variables aleatorias (i.d.d.)*.

1.2.3. Propiedades de la esperanza y varianza

A continuación, se detallarán algunas de las propiedades fundamentales acerca de la esperanza de una variable aleatoria, que serán de gran utilidad de cara a realizar simplificaciones cuando trabajemos con variables aleatorias. Asimismo, se introduce el concepto de varianza, que está estrechamente relacionado con la esperanza y que usaremos en el devenir del trabajo.

Proposición 1.35. *La esperanza matemática de una constante ($k \in \mathbb{R}$) para una variable aleatoria X es la propia constante.*

Demostración.

$$\mathbb{E}[k] = \int_{-\infty}^{+\infty} k f_X(x)\,dx = k \cdot \int_{-\infty}^{+\infty} f_X(x) = k$$

pues $f_X(x)$ es función de densidad de la variable aleatoria X y, por tanto, el valor de su integral es 1.

□

Proposición 1.36 (Linealidad de la esperanza)**.** *Sean X, Y dos variables aleatorias y $\alpha, \beta \in \mathbb{R}$. Se tiene que $\mathbb{E}[X]$ es un operador lineal, es decir:*

$$\mathbb{E}[\alpha X + \beta Y] = \alpha\mathbb{E}[X] + \beta\mathbb{E}[Y].$$

Demostración. La demostración es consecuencia trivial de la linealidad de la integral de Lebesgue.

□

Teorema 1.37. *Sean $X_1, \ldots, X_n$ variables aleatorias independientes definidas sobre el mismo espacio de probabilidad $(\Omega, \mathcal{A}, \mathrm{P})$, tales que existe la esperanza de cada una de ellas, es decir, $\exists\ \mathbb{E}[X_i]\ \forall i \in \{1,\ldots,n\}$. Entonces existe $\mathbb{E}[X_1 \cdots X_n]$ y, además, se verifica*

$$\mathbb{E}[X_1 \cdots X_n] = \mathbb{E}[X_1] \cdots \mathbb{E}[X_n] = \prod_{i=1}^{n} \mathbb{E}[X_i].$$

Demostración. La demostración se sigue de manera sencilla utilizando el concepto de independencia. Para ello y por simplificar, consideramos el caso en el que tenemos dos variables aleatorias continuas, pues el resultado para el caso discreto y teniendo n variables aleatorias

es análogo.

Por tanto, sean X_1, X_2 dos variables aleatorias continuas con función de densidad asociada f_{X_1} y f_{X_2} respectivamente. La esperazna de la multiplicación de dichas variables viene dada por

$$\mathbb{E}[X_1 X_2] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1 x_2 f_{X_1,X_2}(x_1, x_2)\, dx_1 dx_2$$

donde $f_{X_1,X_2}(x_1, x_2)$ denota la función de densidad conjunta de las variables aleatorias. Dado que las variables aleatorias son independientes, la función de densidad conjunta se puede expresar como el producto de las funciones marginales de cada variable aleatoria, es decir

$$f_{X_1,X_2}(x_1, x_2) = f_{X_1}(x_1) f_{X_2}(x_2).$$

Finalmente, sustituyendo este resultado en la expresión anterior obtenemos el desenlace buscado:

$$\begin{aligned}
\mathbb{E}[X_1 X_2] &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1 x_2 f_{X_1,X_2}(x_1, x_2)\, dx_1 dx_2 \\
&= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1 x_2 f_{X_1}(x_1) f_{X_2}(x_2)\, dx_1 dx_2 \\
&= \left(\int_{-\infty}^{+\infty} x_1 f_{X_1}(x_1)\, dx_1 \right) \left(\int_{-\infty}^{+\infty} x_2 f_{X_2}(x_2)\, dx_2 \right) \\
&= \mathbb{E}[X_1]\mathbb{E}[X_2].
\end{aligned}$$

□

Proposición 1.38. *Sean $X : (\Omega, \mathcal{A}, P) \to (\mathbb{R}, \mathcal{B})$ una variable aleatoria y $g : (\mathbb{R}, \mathcal{B}) \to (\mathbb{R}, \mathcal{B})$ una función (medible) integrable de Lebesgue, entonces $g(X)$ es una variable aleatoria.*

Demostración. La demostración se basa en el hecho de que X y g son funciones medibles y la composición de funciones medibles es una función medible. Es decir $g(X) : (\Omega, \mathcal{A}, P) \to (\mathbb{R}, \mathcal{B})$ definida por $g(X)(w) = g(X(w))$ con $w \in \Omega$ es una función medible desde el espacio de probabilidad $(\Omega, \mathcal{A}, P)$ hasta el espacio de medida $(\mathbb{R}, \mathcal{B})$.

□

Observación 1.39. Sea X una variable aleatoria continua con función de densidad f_X y g una función (medible) de Lebesgue. La esperanza de la variable aleatoria continua $g(X)$ viene dada por

$$\mathbb{E}[g(X)] = \int_{-\infty}^{+\infty} g(x) f_X(x)\, dx.$$

Un resultado análogo se observaría para una variable aleatoria discreta, con la salvedad de que tendrías la suma en lugar de la integral y la función de probabilidad en lugar de la

función de densidad.

Es conveniente remarcar la observación anterior, pues es posible conocer la esperanza de la variable aleatoria $g(X)$ conociendo la distribución de probabilidad de la variable X, sin la necesidad de conocer la propia distribución de $g(X)$. De igual manera, destacamos que estas propiedades de la esperanza se pueden generalizar para vectores aleatorios.

A partir de la definición de esperanza de una variable aleatoria se puede construir el concepto de varianza de la variable aleatoria. A modo intuitivo, la varianza es una medida estadística que nos ayudará a cuantificar la dispersión de un conjunto de datos en relación con su media (esperanza).

Definición 1.40 (Varianza de una variable aleatoria)**.** Sea X una variable aleatoria. Llamamos *varianza* de la variable aleatoria X ($Var(X)$) al valor esperado dado por

$$Var(X) = \mathbb{E}[(X - \mathbb{E}[X])^2].$$

Además, denominaremos *desviación típica* (σ) a $+\sqrt{Var(X)}$.

Observación 1.41. La expresión de la varianza de una variable aleatoria puede expandirse de la siguiente manera:

$$\begin{aligned} \mathrm{Var}(X) &= \mathbb{E}[(X - \mathbb{E}[X])^2] \\ &= \mathbb{E}[X^2 - 2X\mathbb{E}[X] + \mathbb{E}[X]^2] \\ &= \mathbb{E}[X^2] - 2\mathbb{E}[X]\mathbb{E}[X] + \mathbb{E}[X]^2 \\ &= \mathbb{E}[X^2] - \mathbb{E}[X]^2. \end{aligned}$$

donde se han aplicado algunas de las propiedades de la esperanza comentadas anteriormente.

1.3. Distribuciones de probabilidad

Las distribuciones de probabilidad son funciones que describen el comportamiento de una variable aleatoria, indicando la probabilidad de que esta tome ciertos valores. Estas distribuciones pueden ser, como se ha comentado en la sección anterior, discretas o continuas, dependiendo de la naturaleza de la variable aleatoria. En esta sección trataremos algunas de las distribuciones más usuales al trabajar con datos del mundo real y que, además, serán utilizadas durante el transcurso del trabajo.

1.3.1. Distribución Normal

La distribución normal o distribución de Gauss es la distribución de probabilidad más estudiada y utilizada en el ámbito de la inferencia estadística [Bry95], dadas sus propiedades

matemáticas, como su simetría, la concentración de probabilidades alrededor de la media y su relación con otras distribuciones.

Definición 1.42 (Distribución normal)**.** Sea X una variable aleatoria continua. Decimos que X sigue una *distribución normal* si su función de densidad f_X viene dada por

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

donde $\mu, \sigma^2 \in \mathbb{R}$ denotan, respectivamente, la esperanza y varianza de la variable aleatoria X.

Además, si X sigue una distribución normal lo denotaremos como $X \sim \mathcal{N}(\mu, \sigma)$. De igual modo, si $\mu = 0$ y $\sigma^2 = 1$, entonces diremos que la variable aleatoria $X \sim \mathcal{N}(0,1)$ sigue una *distribución normal estándar*.

Proposición 1.43. *Si $X \sim \mathcal{N}(\mu, \sigma)$, entonces la distribución es simétrica con respecto a μ.*

Como consecuencia de la proposición anterior, se pueden relacionar todas las variables aleatorias normales con la distribución $\mathcal{N}(0,1)$.

Proposición 1.44. *Sea $X \sim \mathcal{N}(\mu, \sigma)$. Entonces*

$$Z = \frac{X - \mu}{\sigma}$$

es una variable aleatoria que sigue una distribución normal estándar $Z \sim \mathcal{N}(0,1)$.

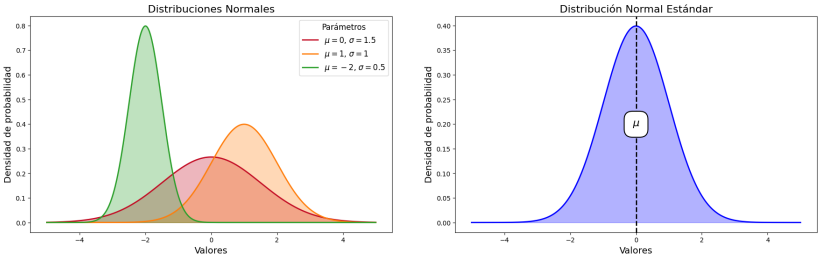

Figura 1.1.: Ejemplos de distribuciones normales. A la izquierda observamos distintas distribuciones normales, donde podemos notar como el parámetro μ determina el centro de la distribución y el parámetro σ controla la dispersión de los datos. A la derecha observamos una distribución normal estándar, donde se puede apreciar con claridad que la distribución es simétrica con respecto a μ. Imagen original del autor.

2. Álgebra Lineal: Matrices

En este capítulo se realizará una introducción al álgebra lineal de matrices, con el objetivo de presentar dos conceptos fundamentales de cara al desarrollo del trabajo: la descomposición en valores singulares y la pseudoinversa de una matriz. Estos conceptos serán presentados de manera precisa y estructurada, proporcionando las bases teóricas necesarias para su comprensión y aplicación en contextos posteriores.

En cuanto a las referencias utilizadas a lo largo de este capítulo, debemos destacar [FIS14], [Str23] y [Poo11]. Estos libros nos ayudaron a presentar los conceptos básicos más importantes, así como las demostraciones de los resultados más relevantes del capítulo.

2.1. Vectores y espacios vectoriales

En primer lugar, se realizará una presentación de los conceptos básicos necesarios, incluyendo los vectores y la estructura de los espacios vectoriales. Estos elementos serán fundamentales para establecer una base sólida para comprender las operaciones y propiedades esenciales en el álgebra lineal de matrices.

Definición 2.1. Un cuerpo F es un conjunto en el que dos operaciones $+$ y $\bullet$, llamadas, respectivamente, suma y multiplicación, están definidas de manera que, para cada par de elementos $x, y \in F$, existe únicos elementos $x + y, x \bullet y \in F$ para los que se cumplen las siguientes condiciones para todos los elementos $a, b, c \in F$:

1. $a + b = b + a$ y $a \bullet b = b \bullet a$.
2. $(a + b) + c = a + (b + c)$ y $(a \bullet b) \bullet c = a \bullet (b \bullet c)$.
3. Existen elementos distintos 0 y 1 en F de manera que $0 + a = a$ y $1 \bullet a = a$.
4. Para cada elemento $a \in F$ y cada elemento distinto de cero $b \in F$, existen elementos $c, d \in F$ verificando $a + c = 0$ y $b \bullet d = 1$.
5. $a \bullet (b + c) = a \bullet b + a \bullet c$.

Denominaremos *escalares* a los elementos de un cuerpo F. En nuestro caso, centraremos nuestra atención en el cuerpo de los números reales ($\mathbb{R}$) con las definiciones usuales de suma y multiplicación.

Definición 2.2. Un objeto de la forma $(a_1, a_2, \ldots, a_n)$ cuyas entradas $a_1, a_2, \ldots, a_n$ son elementos de un cuerpo F, se denomina una n-tupla con entradas o componentes en F. Además, dos n-tuplas $(a_1, a_2, \ldots, a_n)$ y $(b_1, b_2, \ldots, b_n)$ cuyas componentes pertenecen al cuerpo F serán iguales si $a_i = b_i$, $\forall i \in \{1, 2, \ldots, n\}$.

Una vez expuestas las definiciones de cuerpo y n-tupla, podemos introducir el concepto clave de esta sección: el espacio vectorial. A partir de esta noción fundamental, será posible definir de manera precisa el concepto de vector, que será utilizado en la sección siguiente para definir el concepto de matriz.

Definición 2.3. Un espacio vectorial V sobre un cuerpo F consiste en un conjunto en el que dos operaciones (denominadas suma y multiplicación por escalar, respectivamente) están definidas de manera que, para cada par de elementos $x, y \in V$ existe un único elemento $x + y \in V$ y para cada elemento $a \in F$ y cada elemento $x \in V$ existe un único elemento $ax \in V$ cumpliendo las siguientes propiedades:

1. $\forall x, y \in V,\ x + y = y + x$.
2. $\forall x, y, z \in V,\ (x + y) + z = x + (y + z)$.
3. Existe un elemento en V, denotado por 0 de manera que $x + 0 = x,\ \forall x \in V$.
4. Para cada elemento $x \in V$ existe un elemento $y \in V$ de manera que $x + y = 0$.
5. $\forall x \in V,\ 1x = x$.
6. Para cada par de elementos $a, b \in F$ y cada elemento $x \in V,\quad (ab)x = a(bx)$.
7. Para cada elemento $a \in F$ y cada par de elementos $x, y \in V,\quad a(x + y) = ax + ay$.
8. Para cada par de elementos $a, b \in F$ y cada elemento $x \in V,\quad (a + b)x = ax + bx$.

Denominaremos *vectores* a los elementos, generalmente tuplas de un determinado tamaño, de un espacio vectorial V.

Observación 2.4. El conjunto de todas las n-tuplas con entradas en un cuerpo F se denota por F^n (formado por el producto cartesiano de n veces el cuerpo F). Este conjunto es un espacio vectorial sobre F con las operaciones de suma y multiplicación por coordenadas. Es decir, sean $u = (u_1, u_2, \ldots, u_n) \in F^n$, $v = (v_1, v_2, \ldots, v_n) \in F^n$ y $c \in F$, entonces:

- $u + v = (u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n)$.
- $cu = (cu_1, cu_2, \ldots, cu_n)$.

Además, los vectores de F^n se escribirán como vectores columna en lugar de vectores fila. Es decir, si $v \in F^n$ entonces

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = (v_1, v_2, \ldots, v_n)^T$$

De esta manera, $\mathbb{R}^n$ es un espacio vectorial sobre $\mathbb{R}$. Asimismo, dado que nuestro estudio se restringe al cuerpo de los números reales, un vector (real) de dimensión n hará referencia a una n-tupla de números reales, es decir $v \in \mathbb{R}^n,\ v = (v_1, v_2, \ldots, v_n)^T$, donde $v_i \in \mathbb{R}$ para cada $i \in \{1, 2, \ldots, n\}$.

Finalmente, se introducen los conceptos de norma y ortonormalidad de vectores, que serán de gran utilidad en el desarrollo de resultados posteriores..

Definición 2.5. La *norma (o longitud)* de un vector $v \in \mathbb{R}^n$ es el escalar no negativo $||v||$ definido por

$$||v|| = \sqrt{v \cdot v} = \sqrt{v_1^2, v_2^2 + \cdots + v_n^2}.$$

Definición 2.6. Diremos que dos vectores $u, v \in \mathbb{R}^n$ son *ortogonales* si $u \cdot v = 0$. A su vez, diremos que dos vectores $u, v \in V$ son *ortonormales* si son ortogonales y unitarios (su norma es igual a 1).

2.2. Introducción a las matrices

A continuación, se introducirá el concepto de matriz, así como las operaciones y propiedades básicas asociadas a las mismas, que serán las herramientas fundamentales para entender los conceptos clave expuestos en la próxima sección.

Definición 2.7. Una matriz A de tamaño $m \times n$ ($m, n \in \mathbb{N}$) con entradas en un cuerpo F es un conjunto bidimensional de la forma

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

donde cada entrada a_{ij} $(1 \leq i \leq m, 1 \leq j \leq n)$ es un elemento de F. Además, llamamos a las entradas a_{ij} con $i = j$ las entradas diagonales de la matriz. Asimismo, las entradas de la forma $a_{i1}, a_{i2}, \ldots, a_{in}$ forman la i-ésima fila de la matriz y las entradas de la forma $a_{1j}, a_{2j}, \ldots, a_{mj}$ forman la j-ésima columna de la matriz. De esta manera, las filas de la matriz anterior se consideran vectores (n-tuplas) en F^n y las columnas se consideran vectores (m-tuplas) en F^m. Finalmente, la matriz cuyas entradas son todas iguales a cero es llamada la matriz cero y se denota por O.

Definición 2.8. Denotaremos por a_{ij} a la entrada de la matriz A correspondiente a la i-ésima fila y a la j-ésima columna. De esta manera, dadas dos matrices A y B de tamaño $m \times n$, diremos que son **iguales** si todas sus correspondientes entradas son iguales, es decir, $a_{ij} = b_{ij}$ donde $1 \leq i \leq m$ y $1 \leq j \leq n$.

Observación 2.9. El conjunto de todas las matrices de tamaño $m \times n$ con entradas en un cuerpo F es un espacio vectorial que denotaremos por $\mathcal{M}_{m \times n}(F)$ con las operaciones de suma y multiplicación escalar siguientes. Sean $A, B \in \mathcal{M}_{m \times n}(F)$ y $c \in F$, entonces:

- $(a+b)_{ij} = a_{ij} + b_{ij}$ donde $1 \leq i \leq m$ y $1 \leq j \leq n$.
- $(ca)_{ij} = ca_{ij}$ donde $1 \leq i \leq m$ y $1 \leq j \leq n$.

Una vez conocidas la suma y multiplicación por escalares de matrices, la operación utilizada con más frecuencia es su producto, el cual se definirá a continuación.

Definición 2.10. Sea A una matriz de tamaño $m \times n$ y B una matriz de tamaño $n \times p$. Definimos el producto de las matrices A y B, denotado por AB, como la matriz de tamaño $m \times p$ cuyas entradas se corresponden con

$$(ab)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}, \quad \text{donde } 1 \leq i \leq m \ \text{ y } \ 1 \leq j \leq p.$$

Seguidamente, se presentan definiciones de algunos tipos de matrices que serán de gran utilidad a lo largo del trabajo.

Definición 2.11. Llamaremos matriz **cuadrada** a toda matriz A con el mismo número de filas y columnas, es decir, A es de la forma $m \times m$ con $m \in \mathbb{N}$ y el espacio vectorial asociado lo denotaremos, simplemente, como $A \in \mathcal{M}_m(F)$.

Definición 2.12. La matriz **traspuesta** A^T de una matriz $A \in \mathcal{M}_{m\times n}(F)$ es la matriz de tamaño $n \times m$ que se obtiene de la matriz A al intercambiar las filas por las columnas, es decir $(a^T)_{ij} = a_{ji}$. A su vez, decimos que una matriz A es **simétrica** si cumple $A^T = A$. De la propia definición se deduce que una matriz simétrica debe ser cuadrada.

Definición 2.13. Una matriz cuadrada A de tamaño n se llama **ortogonal** si sus columnas están formadas por vectores ortonormales dos a dos.

Dado que el producto de matrices es una operación no convencional, es necesario identificar el elemento neutro de dicho producto, que se corresponderá con lo que denominaremos como matriz identidad. Su nombre proviene de su relación con la aplicación identidad, dado que representa una función de un espacio vectorial sobre sí mismo.

Definición 2.14. Definimos la delta de Kronecker δ_{ij} como la función dada por

$$\delta_{ij} = \begin{cases} 1 & \text{si } i = j \leq r, \\ 0 & \text{si } i \neq j. \end{cases}$$

De esta manera, definimos la **matriz identidad** de tamaño $n \times n$ (I_n) como $(i)_{ij} = \delta_{ij}$. Destacamos de la definición que la matriz identidad es una matriz cuadrada y simétrica.

Ejemplo 2.15. A continuación, se presentan las matrices identidad hasta tamaño n:

$$I_1 = (1), \quad I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \cdots, \quad I_n = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

2.2.1. Rango de una matriz

En esta sección, se abordará el concepto de rango de una matriz, una propiedad fundamental en álgebra lineal, ya que proporciona información crucial sobre la dimensión del espacio generado por sus filas o columnas, y tiene aplicaciones clave en la determinación de la invertibilidad de una matriz.

Comenzaremos con unas definiciones previas, realtivas a los vectores, que nos ayudarán a lo largo de la sección.

Definición 2.16. Sea V un espacio vectorial y S un subconjunto no vacío de V. Un vector $v \in V$ se dice que es una **combinación lineal** de vectores de S si existe un número finito de vectores $u_1, u_2, \ldots, u_n \in S$ y escalares $a_1, a_2, \ldots, a_n \in F$ de manera que $v = a_1u_1, a_2u_2, \ldots, a_nu_n$. En este caso, diremos que v es una combinación lineal de $u_1, u_2, \ldots, u_n$ y llamaremos a $a_1, a_2, \ldots, a_n$ los coeficientes de la combinación lineal.

Observación 2.17. En cualquier espacio vectorial V, el vector cero (todas sus entradas se corresponden con el 0) es una combinación lineal de cualquier subconjunto no vacío de V, dado que $0v = 0$.

Definición 2.18. Un subconjunto S de un espacio vectorial V es llamado **linealmente dependiente** si existe un número finito de vectores distintos $u_1, u_2, \ldots, u_n \in S$ y escalares $a_1, a_2, \ldots, a_n$, donde no todos pueden ser 0, de manera que

$$a_1u_1 + a_2u_2 + \cdots + a_nu_n = 0.$$

En este caso, también decimos que los vectores de S son linealmente dependientes.

Definición 2.19. Un subconjunto S de un espacio vectorial que no es linealmente dependiente es llamado **linealmente independiente**. De igual manera que en la definición anterior, decimos que los vectores de S son linealmente independientes.

Observación 2.20. De las definiciones anteriores se extrae que un vector $v \in V$ es linealmente independiente si no puede ser expresado como combinación lineal de otros vectores del espacio (a excepción del vector cero).

En lo que sigue, se presentan nuevas definiciones sobre matrices, que serán de utilidad para calcular el rango de una matriz.

Definición 2.21. Sea A una matriz de tamaño $m \times n$. Se dice que A es una matriz **escalonada** si es O o satisface las tres condiciones siguientes:

1. El primer elemento no nulo de cada fila, si existe, es un 1.
2. El primer 1 de la segunda y sucesivas filas está a la derecha del primer 1 de la fila anterior.
3. Si tiene filas nulas (compuestas únicamente por ceros), estas aparecen en la parte inferior de la matriz, justo debajo de las filas no nulas.

Además, las operaciones elementales que se pueden realizar a una matriz para obtener su forma escalonada son las siguientes:

- Intercambiar dos filas (columnas).
- Multiplicar una fila (columna) por un múltiplo distinto de cero.
- Sumar un múltiplo de una fila (columna) a otra fila (columna).

Definición 2.22. Sean A y B dos matrices de tamaño $m \times n$. Se dice que la matriz A es **equivalente** por filas a la matriz B (o simplemente equivalente) si B se obtiene de A por medio de la aplicación sucesiva de operaciones elementales.

Definición 2.23. Una matriz A de tamaño $m \times n$ es **escalonada reducida** si es escalonada y además todo elemento en una columna que esté encima del primer uno de cualquier fila es cero.

Ejemplo 2.24. Para matrices cuadradas de tamaño 2, las posibles matrices escalonadas reducidas son las siguientes:

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & x \\ 0 & 0 \end{pmatrix} \quad y \quad \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

donde $x \in F$ puede ser cualquier escalar.

Una vez introducidas las definiciones anteriores, disponemos de todas las herramientas necesarias para introducir el concepto de rango de una matriz.

Definición 2.25 (Rango de una matriz)**.** Sea A una matriz de tamaño $m \times n$. Se denomina **rango** de A ($rang(A)$) al número de filas no nulas de la matriz en la forma escalonada reducida equivalente a A. De manera equivalente, el rango de una matriz se puede definir como la dimensión del espacio generado por sus vectores fila o columna. De esta manera, el rango de una matriz será igual al número de vectores fila o columna que sean linealmente independientes entre sí.

2.2.2. Matriz invertible

En esta sección, se introducirá el concepto de matriz invertible, que se encontrará ampliamente ligado al rango de la matriz. Además, se presentarán algunas propiedades básicas de las matrices invertibles, necesarias para el desarrollo del trabajo.

Definición 2.26. Sea A una matriz cuadrada de tamaño n. Entonces la matriz A es **invertible** (tiene inversa) si existe una matriz B de tamaño $n \times n$ de manera que $AB = BA = I$, donde I denota la matriz identidad de tamaño n.

Observación 2.27. La formulación realizada en la Definición 2.26 de invertibilidad de matrices sólo es válida para matrices cuadradas.

Corolario 2.28. *Sea A una matriz cuadrada de tamaño n. Si A es invertible, entonces la matriz B que verifica $AB = BA = I$ es única. Además, a la matriz B se le denomina la matriz **inversa** de A y se le denota por A^{-1}.*

Demostración. Sea A una matriz cuadrada de tamaño n y sean B y C dos matrices verificando $AB = BA = I$ y $AC = CA = I$. Entonces, se verifica

$$C = CI = C(AB) = (CA)B = IB = B$$

donde se ha usado que el producto de matrices es asociativo.

□

Tras haber presentado la definición de matriz invertible procedemos a exponer un resultado que nos indicará cuando una matriz admite inversa.

Teorema 2.29. *Sea A una matriz cuadrada de tamaño n, entonces equivalen:*

1. *A es invertible.*
2. *A es equivalente a I_n.*
3. *El rango de A es n.*

Demostración. El teorema quedará probado al demostrar la cadena de implicaciones $(1) \implies (2) \implies (3) \implies (1)$.

$(1) \implies (2)$. Dado que A es invertible, existe A^{-1} verificando $AA^{-1} = I_n$. Además A^{-1} puede expresarse como producto de matrices elementales, es decir, $A^{-1} = E_k \cdots E_2 E_1$. Esto implica que A puede transformarse mediante operaciones elementales de filas a la matriz I_n.

$(2) \implies (3)$. Si A es equivalente a la matriz I_n, significa que mediante operaciones elementales podemos reducir A a la matriz identidad. Por tanto, dado que I_n está formado por n vectores fila linealmente independientes, se deduce que el rango es n.

$(3) \implies (1)$ Sabemos que A es equivalente a su forma escalonada reducida RA, donde R es la matriz resultante de multiplicar las distintas operaciones elementales utilizadas, que es invertible al ser producto de matrices elementales. Dado que $rang(A) = n$ todas las filas de

la matriz RA son distintas del vector cero y, como RA es una matriz cuadrada en la forma escalonada reducida, se tiene que $RA = I_n$. Como R es invertible, todas las inversas por la izquierda de R son las mismas que las inversas por la derecha de R, luego se tiene que $RA = AR = I_n$ y, por tanto, A es invertible con inversa R.

□

Finalmente, se detallarán algunas de las propiedades más importantes y relevantes a lo largo del desarrollo del trabajo de las matrices invertibles.

Teorema 2.30. *Sean A y B matrices cuadradas e invertibles del mismo tamaño. Entonces se cumple:*

1. *A^{-1} es invertible y $\left(A^{-1}\right)^{-1} = A$ (la inversa de A^{-1} es la propia matriz A).*
2. *A^T es invertible y $\left(A^T\right)^{-1} = \left(A^{-1}\right)^T$.*
3. *La matriz AB es invertible y $(AB)^{-1} = B^{-1}A^{-1}$.*

Demostración. (1). Dado que A es invertible, se verifica que $A^{-1}A = AA^{-1} = I$, luego también A^{-1} es invertible y su inversa (que sabemos que es única por el Corolario 2.28) es la propia matriz A.

(2). Sabemos que la matriz A es invertible, luego verifica $A^{-1}A = AA^{-1} = I$. Tomamos ahora la traspuesta en ambos lados de la ecuación, obteniendo $\left(A^{-1}A\right)^T = \left(AA^{-1}\right)^T = I^T$. Usando que la traspuesta de un producto de matrices es $(AB)^T = B^TA^T$, obtenemos el resultado deseado.

(3). Basta comprobar que $(AB)B^{-1}A^{-1} = A(BB^{-1})A^{-1} = AIA^{-1} = AA^{-1} = I$ y que $B^{-1}A^{-1}(AB) = B^{-1}(A^{-1}A)B = B^{-1}IB = B^{-1}B = I$, donde se utiliza que A y B son matrices invertibles y la asociatividad del producto de matrices.

□

2.3. Determinantes, vectores propios y valores propios

Definición 2.31. Sea A una matriz cuadrada de tamaño n con $n \geq 2$. Definimos el **determinante** de la matriz A ($\det(A)$) como el escalar dado por la función $\det : \mathcal{M}_{n\times n}(F) \to F$, definida como sigue

$$\det(A) = a_{11}\det(A_{11}) - a_{12}\det(A_{12}) + \cdots + (-1)^{n+1}a_{1n}\det(A_{1n}) = \sum_{j=1}^{n}(-1)^{j+1}a_{1j}\det(A_{1j})$$

donde A_{ij} hace referencia a la submatriz que se obtiene al eliminar de la matriz A la fila i-ésima y la columna j-ésima. Para el caso $n = 1$, el determinante de A es la propia entrada de la matriz A.

El siguiente resultado nos ayuda a conocer cuando una matriz es invertible a través de su determinante, sin necesidad de tener que encontrar la matriz escalonada reducida.

Corolario 2.32. *Una matriz $A \in \mathcal{M}_n(F)$ es invertible si y solo si* $\det(A) \neq 0$. *Además, si A es invertible, entonces* $\det(A^{-1}) = \frac{1}{\det(A)}$.

Demostración. Si $A \in \mathcal{M}_n(F)$ no es invertible, entonces $rang(A)$ es menor a n. Esto significa que A tiene filas (o columnas) que son linealmente dependientes entre sí y, a través de operaciones elementales, podemos obtener una matriz B equivalente a la matriz A con alguna de sus filas formada por un vector nulo, lo que implica que $\det(B) = 0$, pero como A es equivalente a B, se verifica que $\det(A) = \det(B) = 0$.

Por otra parte, si $A \in \mathcal{M}_n(F)$ es invertible, entonces se cumple

$$\det(A) \cdot \det(A^{-1}) = \det(AA^{-1}) = \det(I_n) = 1$$

donde se hace uso de la propiedad de que el producto de determinantes de matrices cuadradas es igual al determinante del producto de las propias matrices.

□

Definición 2.33. Sea A una matriz cuadrada de tamaño n. Un vector no nulo $v \in F^n$ se dice que es un **vector propio** (o autovector) de la matriz A si $Av = \lambda v$ para algún escalar $\lambda \in F$. Además, el escalar λ se llama **valor propio** (o autovalor) de la matriz A correspondiente al vector propio v.

El siguiente teorema nos proporciona cómo calcular, de manera práctica, los valores propios de una determinada matriz.

Teorema 2.34. *Sea $A \in \mathcal{M}_n(F)$. Entonces un escalar $\lambda \in F$ es un valor propio de la matriz A si y solo si* $\det(A - \lambda I_n) = 0$.

Demostración. Un escalar λ es un valor propio de una matriz A si y solo si existe un vector no nulo $v \in F^n$ de manera que $Av = \lambda v$, es decir, $(A - \lambda I_n)(v) = 0$. Esto es cierto si y solo si la matriz $A - \lambda I_n$ no es invertible. Sin embargo, este resultado es equivalente (haciendo uso del Corolario 2.32) al hecho de que $\det(A - \lambda I_n) = 0$.

□

Corolario 2.35. *Para cualquier matriz $A \in \mathcal{M}_{m \times n}(\mathbb{R})$, la matriz $A^T A$ es simétrica y, en consecuencia, puede ser diagonalizable ortogonalmente (teorema espectral real [Blu21]). De esta forma, todos los valores propios de la matriz $A^T A$ son no negativos.*

Demostración. Sea λ un valor propio de la matriz $A^T A$ con su correspondiente vector propio unitario v asociado. Entonces, se verifica

$$0 \leq ||Av||^2 = (Av) \cdot (Av) = (Av)^T Av = v^T A^T Av = v^T \lambda v = \lambda(v \cdot v) = \lambda ||v||^2 = \lambda.$$

□

2.4. Descomposición en valores singulares y pseudoinversa

En esta sección se presentan los principales resultados sobre matrices abordados en este trabajo: la descomposición en valores singulares y la pseudoinversa. Cabe destacar que estos resultados se enuncian para matrices con escalares en el cuerpo $\mathcal{R}$, aunque su generalización a otros cuerpos es posible.

Definición 2.36. Si A es una matriz de tamaño $m \times n$, los **valores singulares** de A son las raíces cuadradas (positivas) de los valores propios de la matriz $A^T A$ y se denotan mediante $\sigma_1, \sigma_2, \ldots, \sigma_n$. Además, es convencional ordenar los valores singulares de manera que $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$.

Observación 2.37. Tiene sentido hablar de las raíces cuadradas positivas de los valores propios de la matriz $A^T A$ por el resultado obtenido en el Corolario 2.35.

El siguiente resultado nos indica que toda matriz, independientemente de su estructura, puede ser factorizada como producto de tres matrices, dos de las cuales serán ortogonales. Este resultado se conoce como *descomposición en valores singulares* y es una de las factorizaciones más importantes de todas las matrices.

Teorema 2.38 (Descomposición en valores singulares)**.** *Sea $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ una matriz cuyo rango es r y con valores singulares positivos $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$ y sea Σ la matriz de tamaño $m \times n$ definida por*

$$\Sigma_{ij} = \begin{cases} \sigma_i & \text{si } i = j \leq r, \\ 0 & \text{en otro caso.} \end{cases}$$

Entonces existen una matriz ortogonal de tamaño $m \times m$ U y una matriz ortogonal de tamaño $n \times n$ V de manera que

$$A = U\Sigma V^T.$$

A esta factorización la llamaremos descomposición en valores singulares (SVD) *de A.*

Demostración. La demostración se fundamenta en la construcción directa de las matrices V y U, verificando posteriormente que se satisface el resultado buscado.

Para construir la matriz ortogonal V, hay que encontrar una base ortonormal $\{v_1, v_2, \ldots, v_n\}$ de F^n formada por vectores propios de la matriz simétrica y cuadrada $A^T A$ de tamaño n. Entonces se tiene que $V = [v_1, v_2, \ldots, v_n]$ es una matriz ortogonal y cuadrada de tamaño n.

Para construir la matriz ortogonal U, primero notemos que $\{Av_1, Av_2, \ldots, Av_n\}$ es un conjunto ortogonal de vectores de F^m. Para demostrar esto, basta suponer que v_i es el vector propio de la matriz $A^T A$ correspondiente al valor propio λ_i. Entonces, para $i \neq j$, se cumple

$$(Av_i) \cdot (Av_j) = (Av_i)^T Av_j = v_i^T A^T Av_j = v_i^T \lambda_j v_j = \lambda_j (v_i \cdot v_j) = 0$$

dado que los vectores propios v_i son ortogonales. Recordamos ahora que los valores singulares satisfacen $\sigma_i = ||Av_i||$ y que los primeros r valores singulares son distintos de cero. Por tanto, podemos normalizar $Av_1, \ldots, Av_r$ de la siugiente forma

$$u_i = \frac{1}{\sigma_i} Av_i \quad \text{para } i = 1, \ldots, r. \tag{2.1}$$

Esto garantiza que $\{u_1, \ldots, u_r\}$ es un conjunto ortonormal de F^m, pero si $r < m$ no será una base para F^m. En este caso, se extiende el conjunto $\{u_1, \ldots, u_r\}$ a una base ortonormal $\{u_1, \ldots, u_m\}$ para F^m. Entonces se tiene $U = [u_1, u_2, \ldots, u_m]$. Ahora, falta comprobar que, con la construcción realizada, se satisface el resultado $A = U\Sigma V^T$. Dado que $V^T = V^{-1}$ (al ser la matriz V ortogonal), esto equivale a demostrar que $AV = U\Sigma$.

En primer lugar, sabemos, a partir de la Ecuación (2.1), que $Av_i = \sigma_i u_i$ para $i = 1, \ldots, r$ y que $||Av_i|| = \sigma_i = 0$ para $i = r+1, \ldots, n$. En consecuencia, $Av_i = 0$ para $i = r+1, \ldots, n$. Por tanto,

$$\begin{aligned}
AV &= A\begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix} \\
&= \begin{bmatrix} A\mathbf{v}_1 & \cdots & A\mathbf{v}_n \end{bmatrix} \\
&= \begin{bmatrix} A\mathbf{v}_1 & \cdots & A\mathbf{v}_r & 0 & \cdots & 0 \end{bmatrix} \\
&= \begin{bmatrix} \sigma_1\mathbf{u}_1 & \cdots & \sigma_r\mathbf{u}_r & 0 & \cdots & 0 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_m \end{bmatrix} \begin{bmatrix} \sigma_1 & \cdots & 0 & O \\ \vdots & \ddots & \vdots & O \\ 0 & \cdots & \sigma_r & O \\ O & O & O & O \end{bmatrix} \\
&= U\Sigma
\end{aligned}$$

quedando probada la igualdad $AV = U\Sigma$.

□

Conocemos, de la Subsección 2.2.2, cuando una matriz era invertible. Sin embargo, el resultado mostrado únicamente era válido para matrices cuadradas. El siguiente resultado generaliza el concepto de matriz inversa cuando la matriz no es cuadrada.

Definición 2.39 (Pseudoinversa)**.** Sea $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ con $m > n$ y donde las columnas de A son linealmente independientes. Se define la **pseudoinversa** *(o inversa de Moore-Penrose)* de la matriz A como la matriz $A^\dagger$ dada por

$$A^\dagger = (A^T A)^{-1} A^T$$

donde se puede comprobar que $A^\dagger \in \mathcal{M}_{n\times m}(\mathbb{R})$.

No obstante, dado que toda matriz se puede factorizar en su descomposición en valores singulares, podemos definir la pseudoinversa de una matriz a partir de dicha factorización.

Definición 2.40. Sea A una matriz de tamaño $m \times n$ de rango r con descomposición en valores singulares $A = U\Sigma V^T$ y con valores singulares distintos de cero $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. Sea $\Sigma^\dagger$ la matriz de tamaño $n \times m$ definida por

$$\Sigma^\dagger_{ij} = \begin{cases} \frac{1}{\sigma_i} & \text{si } i = j \leq r, \\ 0 & \text{en otro caso.} \end{cases}$$

Entonces la factorización $A^\dagger = V\Sigma^\dagger U^T$ es una descomposición en valores singulares de $A^\dagger$, donde $\Sigma^\dagger$ es la pseudoinversa de Σ. Además, $A^\dagger$ es la pseudoinversa de A.

El siguiente resultado presenta una forma análoga de definir la pseudoinversa de una matriz, basándose en las propiedades que debe cumplir la pseudoinversa, que serán de gran utilidad en el desarrollo del trabajo.

Definición 2.41 (Condiciones de Moore-Penrose)**.** Sea $A \in \mathcal{M}_{m\times n}(\mathbb{R})$, la pseudoinversa de A, $A^\dagger \in \mathcal{M}_{n\times m}(\mathbb{R})$, es la única matriz que satisface las siguientes propiedades, conocidas como las condiciones de Moore-Penrose:

1. $AA^\dagger A = A$.
2. $A^\dagger AA^\dagger = A^\dagger$.
3. $(AA^\dagger)^T$ es simétrica, es decir, $(AA^\dagger)^T = AA^\dagger$.
4. $(A^\dagger A)^T$ es simétrica, es decir, $(A^\dagger A)^T = A^\dagger A$.

Además, si A es de rango completo, es decir, $rango(A) = r = \min\{m, n\}$, entonces $A^\dagger$ puede expresarse de forma sencilla como sigue

- Si $r = m = n$, entonces la matriz A es invertible y $A^\dagger = A^{-1}$.
- Si $r = m < n$, entonces A tiene filas linealmente independientes (A es sobreyectiva y AA^T es invertible) y $A^\dagger = A^T(AA^T)^{-1}$.
- Si $r = n < m$, entonces A tiene columnas linealmente independientes (A es inyectiva y A^TA es invertible) y $A^\dagger = (A^TA)^{-1}A^T$.

Finalmente, la solución de norma mínima para un problema de mínimos cuadrados puede definirse mediante la pseudoinversa de una matriz, como se establece en el siguiente teorema.

Teorema 2.42. *El problema de mínimos cuadrados $A\mathbf{x} = \mathbf{b}$, con $A \in \mathcal{M}_{m\times n}(\mathbb{R})$, $\mathbf{x} \in \mathbb{R}^n$ y $\mathbf{b} \in \mathbb{R}^m$, tiene una solución única $\bar{\mathbf{x}}$ de mínimos cuadrados de norma mínima dada por*

$$\bar{\mathbf{x}} = A^\dagger \mathbf{b}$$

Demostración. Sea $A \in \mathcal{M}_{m\times n}(\mathbb{R})$ con $rang(A) = r$ y sea $U\Sigma V^T$ su descomposición en valores singulares. De este modo, se tiene que $A^\dagger = V\Sigma^\dagger U^T$. Sean $\mathbf{y} = V^T\mathbf{x}$ y $\mathbf{c} = U^T\mathbf{b}$, expresados de la siguiente forma

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}$$

donde $\mathbf{y_1}, \mathbf{c_1} \in \mathbb{R}^r$.

Se busca minimizar $\|\mathbf{b} - A\mathbf{x}\|$ o, de manera equivalente, $\|\mathbf{b} - A\mathbf{x}\|^2$. Usando que U^T es ortogonal (dado que U es ortogonal), se tiene

$$\|\mathbf{b} - A\mathbf{x}\|^2 = \|U^T(\mathbf{b} - A\mathbf{x})\|^2 = \|U^T(\mathbf{b} - U\Sigma V^T\mathbf{x})\|^2 = \|U^T\mathbf{b} - U^TU\Sigma V^T\mathbf{x}\|^2$$

$$= \|\mathbf{c} - \Sigma\mathbf{y}\|^2 = \left\| \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} - \begin{bmatrix} D & O \\ O & O \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} \mathbf{c}_1 - D\mathbf{y}_1 \\ \mathbf{c}_2 \end{bmatrix} \right\|^2.$$

Dado que sólo disponemos de control sobre $\mathbf{y}_1$, el valor mínimo ocurre cuando $\mathbf{c}_1 - D\mathbf{y}_1 = 0$ o, de manera equivalente, cuando $\mathbf{y}_1 = D^{-1}\mathbf{c}_1$. De modo que todas las soluciones $\mathbf{x}$ de mínimos cuadrados son de la forma

$$\mathbf{x} = V\mathbf{y} = V \begin{bmatrix} D^{-1}\mathbf{c}_1 \\ \mathbf{y}_2 \end{bmatrix}$$

Definimos $\bar{\mathbf{x}} = V\bar{\mathbf{y}} = V \begin{bmatrix} D^{-1}\mathbf{c}_1 \\ 0 \end{bmatrix}$ y afirmamos que $\bar{\mathbf{x}}$ es la solución de mínimos cuadrados de norma mínima. Para demostrarlo, supongamos que $\mathbf{x}' = V\mathbf{y}' = V \begin{bmatrix} D^{-1}\mathbf{c}_1 \\ y_2 \end{bmatrix}$ es otra solución diferente al problema de mínimos cuadrados (por tanto, $\mathbf{y}_2 \neq 0$). Entonces, se verifica

$$\|\bar{\mathbf{x}}\| = \|V\bar{\mathbf{y}}\| = \|\bar{\mathbf{y}}\| < \|\mathbf{y}'\| = \|V\mathbf{y}'\| = \|\mathbf{x}'\|$$

como se quería probar. Por último, falta demostrar que $\bar{\mathbf{x}}$ es igual a $A^\dagger\mathbf{b}$. Para ello, basta calcular

$$\bar{\mathbf{x}} = V\bar{\mathbf{y}} = V \begin{bmatrix} D^{-1}\mathbf{c}_1 \\ 0 \end{bmatrix} = V \begin{bmatrix} D^{-1} & O \\ O & O \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = V\Sigma^\dagger\mathbf{c} = V\Sigma^\dagger U^T\mathbf{b} = A^\dagger\mathbf{b}.$$

□

3. Aprendizaje Automático y Aprendizaje Profundo

3.1. Fundamentos

El aprendizaje automático o *machine learning* (ML) ([Biso6], [Mur22] y [Mur23]) es una rama de la inteligencia artificial que, mediante el uso de datos y algoritmos de aprendizaje, proporciona a las máquinas la capacidad de aprender de manera automática de los datos. En otras palabras, el aprendizaje automático se encarga de utilizar un conjunto de observaciones para descubrir un proceso subyacente en los mismos ([AMMIL12]), con el objetivo de imitar el comportamiento humano, identificando patrones y relaciones en los datos.

En términos generales, el aprendizaje automático se divide en tres tipos: el aprendizaje supervisado, que actúa sobre datos etiquetados (cada ejemplo de entrada se encuentra asociado a una salida conocida), el aprendizaje no supervisado, que trabaja sobre datos no etiquetados, donde es el propio sistema el que debe ser capaz de reconocer los patrones subyacentes de los datos mediante el uso exclusivo de los ejemplos de entrada, y el aprendizaje por refuerzo, que actúa en un entorno de ensayo-error, donde el modelo aprende a través de recompensas y penalizaciones que se le otorgan a medida que realiza las acciones. Para nuestro trabajo, nos limitaremos a trabajar con el *aprendizaje supervisado*.

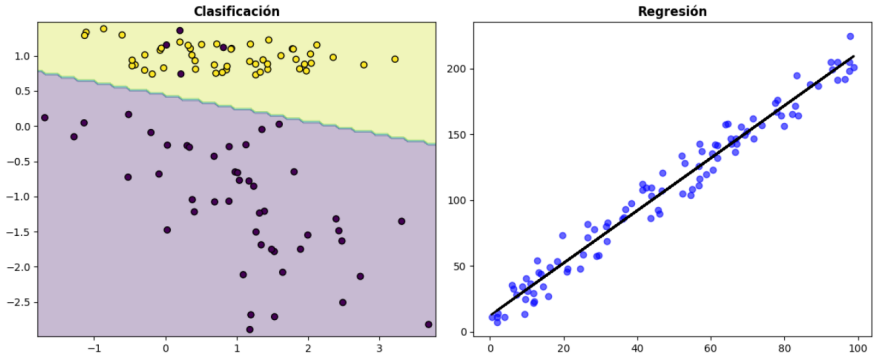

Figura 3.1.: Ejemplos de problemas de clasificación y regresión. A la izquierda, se muestra un problema de clasificación binario, donde se busca encontrar el hiperplano (línea verde) que separe ambos conjuntos de datos etiquetados. A la derecha, se muestra un problema de regresión, donde se busca encontrar la mejor aproximación (línea negra) al conjunto de datos. Imagen original del autor.

Dentro del marco del aprendizaje supervisado, donde el principal objetivo del modelo es aprender a predecir la salida correcta para nuevos ejemplos no conocidos, basándose en las relaciones y patrones extraídos al trabajar con los datos etiquetados, podemos dividir los problemas en dos categorías principales: problemas de clasificación en los que se asigna una salida o clase (discreta) a cada entrada y problemas de regresión en los que se predice un

valor continuo para cada entrada (véase Figura 3.1). De cara al desarrollo de nuestro trabajo, abordaremos ambos tipos de problemas. En particular, en los problemas de clasificación nos centraremos en la clasificación de imágenes.

El aprendizaje profundo o *deep learning* (DL) ([BB23], [Pri23] y [LBH15]) representa un área del aprendizaje automático que utiliza redes neuronales artificiales, inspiradas en la estructura y función del cerebro humano, con múltiples capas, conocidas como redes neuronales profundas ([GBC16] y [Sch15]), con el propósito de identificar y modelar patrones complejos y extraer representaciones jerárquicas en grandes volúmenes de datos.

3.2. Redes neuronales artificiales

Una red neuronal artificial o *artificial neural network* (ANN) ([Bis95], [Rip96]) es un modelo de aprendizaje automático que toma decisiones de manera similar al funcionamiento del cerebro humano, a partir de las interconexiones que presentan las neuronas biológicas, que se organizan en diferentes capas interconectadas (véase Figura 3.2). Estas conexiones simulan las interacciones entre las neuronas biológicas, permitiendo que la red procese información y aprenda de manera similar al propio cerebro humano.

De manera similar al cerebro humano, una red neuronal artificial está formada por neuronas artificiales, también llamadas unidades. Estas unidades se agrupan en diferentes capas formando la arquitectura global de las red neuronal. Cada capa puede contener un número variable de unidades, lo que permite adaptar la red a la complejidad del problema a resolver.

Figura 3.2.: Ejemplos de neurona biológica y neurona artificial extraídos de [BB23]. A la izquierda, se muestran dos neuronas biológicas conectadas mediante sinapsis. A la derecha, se presenta una neurona artificial, que consta de una única capa oculta, donde las conexiones entre unidades están representadas por los pesos y donde cada círculo representa una unidad.

A alto nivel, el funcionamiento de una red neuronal artificial se divide en, al menos, tres capas principales, constituidas por una capa de entrada, una capa oculta y una capa de salida (véase Figura 3.2). En la primera capa o capa de entrada, la información del mundo exterior entra en la red neuronal. Dicha información es procesada y propagada al resto de capas mediante un proceso conocido como propagación hacia delante o *forward propagation*, permitiendo que la información fluya desde la capa de entrada hacia las capas sucesivas.

En la capa oculta o capa de procesamiento, las unidades reciben las salidas de la capa anterior y se encargan de procesar la información mediante conexiones ponderadas (llamadas pesos) y funciones de activación (encargadas de introducir no linealidad en el modelo, permitiendo que la red aprenda y represente relaciones complejas entre las entradas y las salidas), extrayendo características y patrones relevantes de los datos. Los pesos obtenidos controlan la influencia que cada neurona de la capa anterior tiene sobre la neurona actual. Las redes neuronales artificiales pueden tener una gran cantidad de capas ocultas, lo que permite un procesamiento más profundo y detallado de la información. Cada capa oculta analiza la salida de la capa anterior, la procesa aún más y la pasa a la siguiente capa, de modo que, a medida que se avanza por la red, se generan representaciones internas cada vez más abstactas de la entrada original.

Finalmente, en la última capa o capa de salida, la red produce el resultado final en función del problema que se trate y de la predicción calculada haciendo uso de los pesos ajustados en las capas ocultas. La naturaleza de la salida varía según el tipo de tarea que se realice: en un problema de clasificación, la salida es un valor discreto que indica la clase a la que pertenece la entrada, mientras que en un problema de regresión, la salida es un valor continuo que representa una predicción numérica. Por tanto, esta capa es la que traduce la información procesada por la red en un resultado interpretable y acorde con el objetivo del problema.

Por tanto, el proceso de entrenamiento de una red neuronal artificial es un proceso iterativo en el que la red ajusta sus pesos para aprender a realizar tareas específicas. Para llevar a cabo este proceso, es fundamental un conjunto de datos o ejemplos de entrenamiento que sea lo suficientemente representativo, ya que de este conjunto se extraerán los patrones relevantes que la red necesitará aprender.

3.2.1. Redes neuronales convolucionales

Las redes neuronales convolucionales o *convolutional neural networks* (CNN) ([LBD+89], [LBBH98]) son un tipo especial de red neuronal artificial que se utiliza principalmente en procesamiento de imágenes, reconocimiento visual y tareas relacionadas con datos que tienen una estructura de rejilla (matriz multidimensional), como imágenes, vídeos o señales de audio. Una de las características más destacadas de las redes neuronales convolucionales es su capacidad para realizar la extracción automática y jerárquica de características de los datos de entrada, lo que las hace especialmente poderosas para tareas que requieren reconocer patrones complejos en los datos.

Esta capacidad de aprendizaje jerárquico de características es una de las principales razones por las que las CNN son tan eficaces, ya que permiten a la red identificar patrones relevantes sin necesidad de intervención humana para diseñar características específicas, lo que las vuelve especialmente interesantes en áreas como la visión por computador, donde han impulsado avances significativos en aplicaciones como el reconocimiento de imágenes, la segmentación semántica y la detección de objetos, entre otras.

Las redes neuronales convolucionales incluyen varias capas especializadas que las distinguen de las redes neuronales artificiales tradicionales: las capas de convolución, encargadas de la extracción de características de la entrada; las capas de agrupación o *poooling*, respon-

sables de la reducción de la dimensionalidad de la entrada sin perder las características importantes y las capas totalmente conectadas o *fully connected*, que se encuentran en la parte final de la red y que permiten combinar de manera efectiva las características extraídas por las capas anteriores para realizar la predicción final (véase Figura 3.3). Estas capas operan de manera conjunta, transformando la entrada y extrayendo progresivamente las características más relevantes, las cuales se van refinando y volviéndose más abstractas conforme se avanza por la red.

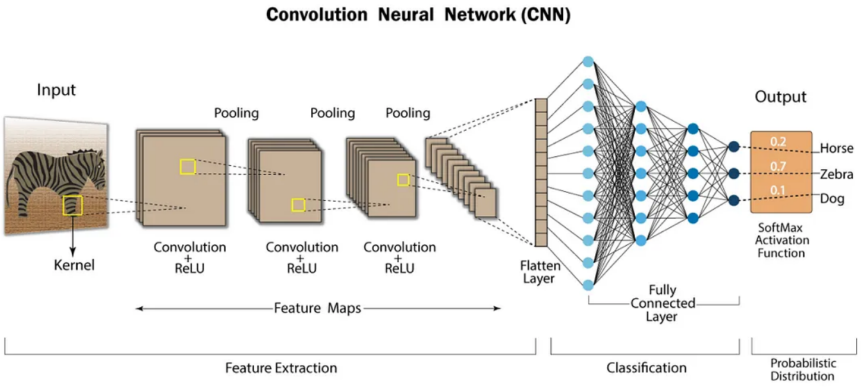

Figura 3.3.: Ejemplo de red neuronal convolucional (CNN) utilizada para clasificación de imágenes [Swa20]. La entrada a la red es una imagen tridimensional. La extracción de características se realiza mediante varias capas convolucionales, seguidas de funciones de activación y capas de pooling, las cuales ayudan a reducir la dimensionalidad. Posteriormente, las características extraídas se aplanan (flattening) y se envían a las capas totalmente conectadas. Finalmente, la salida pasa por una función de activación, en este caso softmax, que genera una distribución de probabilidad sobre las posibles clases de salida.

A continuación se describen las principales capas que conforman una red neuronal convolucional. Se incluyen tanto las capas exclusivas de este tipo de red como aquellas que comparte con las redes neuronales tradicionales:

3.2.1.1. Capa de convolución

La capa convolucional es un componentes fundamental y exclusivo de las redes neuronales convolucionales (CNN), diseñada para extraer características locales de datos estructurados en en forma de matrices multidimensionales. Su funcionamiento se basa en utilizar matrices de valores, conocidas como filtros o *kernels*, que se deslizan sobre la entrada aplicando la operación matemática de convolución.

La convolución es una operación lineal que consiste en desplazar un filtro sobre la entrada, realizando en cada posición una multiplicación elemento a elemento entre los valores del filtro y los de la entrada (diferente de una multiplicación matricial convencional). Posteriormente, se suman estos productos para obtener un único valor de salida. Este proceso se repite hasta deslizar el filtro a lo largo de toda la entrada, obteniendo una nueva matriz denominada mapa de características.

Los valores de los filtros actúan como los pesos que la propia red aprende y optimiza de forma iterativa para maximizar la extracción de características relevantes de la entrada. Además, el número de filtros aplicados sobre cada entrada influye directamente en el mapa de características de salida resultante. Así, a mayor cantidad de filtros, la profundidad del mapa de características resultantes también es mayor.

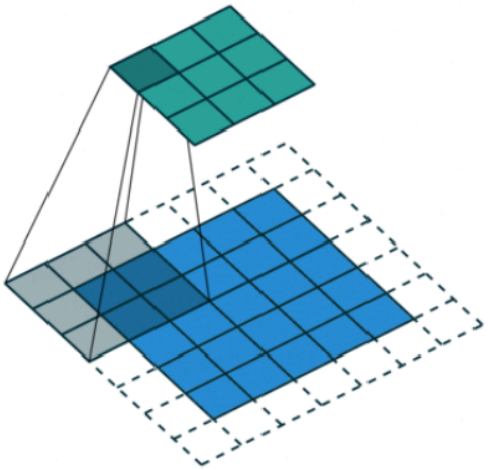

Figura 3.4.: Ejemplo de convolución con padding [Sah18]. El color azul hace referencia a la entrada, mientras que el color verde denota el resultado de la convolución. En particular, el color verde oscuro destaca el resultado de la convolución que se realiza sobre la zona grisácea de la imagen, utilizando, en este caso, un filtro de tamaño 3x3. Por otra parte, las cuadrículas punteadas hacen referencia al padding agregado.

Como se puede observar en la Figura 3.4, la propia naturaleza de la operación de convolución modifica la dimensionalidad de la entrada. Sin embargo, esto no siempre es deseable, ya que en determinadas ocasiones preferiremos mantener la dimensión original de la entrada. Para solucionar este problema, se introduce el concepto de relleno o *padding*, consistente en agregar información adicional alrededor de la entrada, con el fin de preservar su dimensionalidad.

Por otra parte, la elección de la siguiente zona sobre la que se realizará la convolución viene determinada por el tamaño de paso o *stride*. Generalmente, se utiliza un stride de 1, lo que significa que desplazamos el filtro de manera adyacente una posición en cada paso. Sin embargo, también es posible reducir la dimensionalidad de la entrada modificando el stride, como puede observarse en la Figura 3.4, donde, a pesar de utilizar un relleno de una posición para mantener la dimensionalidad, el tamaño del mapa de activación resultante es inferior al tamaño original de la entrada, pues se está utilizando un stride de 2.

En conlusión, el objetivo principal de la capa de convolución es extraer características rele-

vantes de la entrada. Para ello, se utilizan filtros cuyos pesos son aprendidos y optimizados por la propia red. En las primeras capas convolucionales, dado que el tamaño de la entrada es mayor, se detectan principalmente características de bajo nivel, como bordes y texturas. A medida que la información avanza por la red, las capas posteriores capturan patrones más complejos y abstractos. De este modo, mediante la combinación de múltiples capas convolucionales, la red logra desarrollar una representación jerárquica de la entrada.

3.2.1.2. Capa de pooling

Las capas de pooling son otro componente esencial y exclusivo en las redes neuronales convolucionales, ya que su función principal es reducir la dimensionalidad de las representaciones producidas por las capas de convolución, simplificando los mapas de características mientras se preservan las características más relevantes, lo que conlleva una reducción en la cantidad de parámetros y en la complejidad computacional de la red.

El pooling es una operación que toma un conjunto de valores de un mapa de características y lo reduce a un solo valor, con el propósito de submuestrear la información, introduciendo cierta invarianza espacial frente a pequeñas variaciones espaciales de la entrada, lo que permite detectar patrones aunque se encuentren ligeramente desplazados en la imagen, aumentando la robustez de la red.

El tipo de pooling más comúnmente utilizado es el denominado *max pooling* (véase Figura 3.4), que selecciona el valor máximo de un conjunto de valores dentro de una región del mapa de características. Otras alternativas incluyen el *average pooling*, que selecciona el valor promedio del conjunto de valores de la región del mapa de características utilizada y el *global pooling*, que reduce el mapa de características a un único valor. En nuestro proyecto se hará uso del max pooling, integrado en algunas de las arquitecturas que utilizamos para los experimentos.

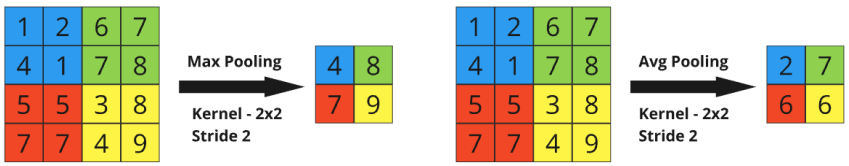

Figura 3.5.: Ejemplos de pooling utilizados en CNN. A la izquierda, se muestra el max pooling, donde se selecciona el valor máximo de una determinada región (en este caso 2x2). A la derecha, se muestra el average pooling, donde se selecciona el valor promedio de la región influida por el filtro. Imagen original del autor.

3.2.1.3. Capa totalmente conectada

Las capas totalmente conectadas o densas (*fully connected*) son un componente fundamental en las redes neuronales, presentes tanto en las tradicionales, formando la arquitectura básica de una ANN, como en las redes convolucionales. Estas capas se encuentran normalmente en

las etapas finales de la red, encargándose de realizar la interpretación final de las características extraídas por las capas anteriores (véase Figura 3.3).

Antes de ingresar a una capa densa, la salida de las capas anteriores debe aplanarse (flattening) en un vector unidimensional. Esto se debe a que, por lo general, dichas salidas tienen forma matricial o tensorial, lo que impide que puedan ser procesadas directamente por las capas densas. Al realizar el aplanamiento, se reorganizan todos los valores en un solo vector, cuyo número de componentes coincide con la cantidad total de elementos de la matriz original. De este modo, el número de unidades en la primera capa densa se corresponde directamente con la longitud de este vector.

Antes de entrar a una capa densa, la salida de las capas anteriores debe aplanarse (*flattening*) en un vector unidimensional puesto que, generalmente, la salida de las capas anteriores tendrán forma matricial o tensorial, lo que impide que puedan ser procesadas directamente por las capas densas. De esta manera, el número de unidades de la primera capa densa se corresponde con el número de componentes del vector unidimensional.

Estas capas combinan y procesan las características extraídas de las capas anteriores y tienen la posibilidad de capturar relaciones globales al conectar cada unidad de una capa con todas las unidades de la siguiente capa por medio de conexiones con pesos entrenables, lo que produce que la mayor parte de los pesos entrenables de una red suelan concentrarse en estas capas debido al elevado número de conexiones que presentan.

3.2.1.4. Capa de activación

Las capas o funciones de activación son las responsables de introducir no linealidad en el modelo, transformando la combinación lineal de las entradas mediante una función matemática. Esta transformación es esencial para que la red pueda aprender y representar patrones complejos en los datos.

Sin funciones de activación, una red neuronal se reduciría simplemente a una combinación lineal de las entradas, sin importar cuántas capas tuviera. Incluso si solo se usaran funciones de activación lineal, la red neuronal se comportaría como una función lineal. Como menciona Bishop en su libro: "Si se considera que las funciones de activación de todas las unidades ocultas de una red son lineales, entonces para cualquier red de este tipo siempre podemos encontrar una red equivalente sin unidades ocultas" ([Biso6]), limitando la capacidad de la red para modelar relaciones complejas.

Por tanto, las capas de activación se colocan después de cada capa lineal (como las capas convolucionales o las capas densas) en una red neuronal, con el objetivo de permitir que dicha capa pueda aprender también relaciones no lineales. A lo largo de este trabajo, se utilizarán algunas de las funciones de activación más comunes y ampliamente empleadas en el campo del aprendizaje profundo, entre las que se incluyen:

- **ReLU (Rectified Linear Unit)**: Es una de las funciones de activación más utilizadas, especialmente en las capas ocultas de las redes neuronales profundas, debido a su

simplicidad computacional, lo que permite acelerar el proceso de entrenamiento. Su expresión matemática es la siguiente

$$\text{ReLU}(x) = \max(0, x)$$

donde x representa la entrada a la función, que corresponde con la salida lineal de la capa anterior de la red neuronal. Sin embargo, esta función de activación puede provocar el problema de "neuronas muertas", donde algunas unidades dejan de activarse permanentemente, cuando su entrada es negativa o 0 (véase Figura 3.6).

- **Softmax:** Se utiliza principalmente en la capa de salida para tareas de clasificación multiclase. Convierte un vector de valores reales en un vector de probabilidades que suman 1, facilitando la interpretación de las salidas como probabilidades de pertenencia a cada clase. Su expresión matemática es la siguiente

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

donde x_i representa la entrada correspondiente a la clase i-ésima antes de aplicar la activación y K el número de clases. Esta función de activación es una generalización de la función sigmoide para clasificación multiclase. Por tanto, para el caso de $K = 2$ (clasificación binaria), esta función se reduce a la función sigmoide.

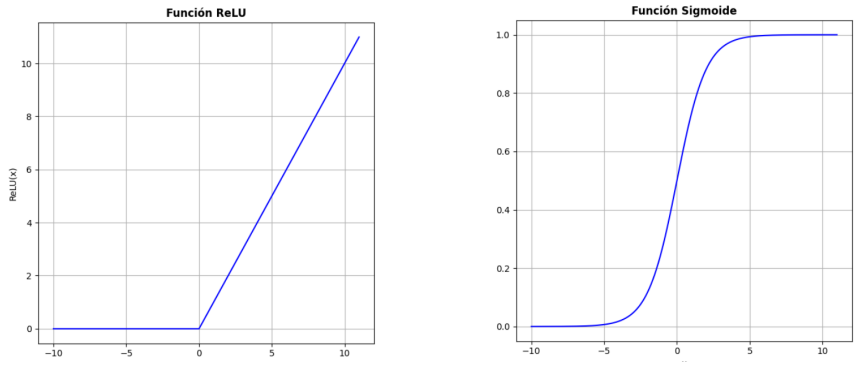

Figura 3.6.: Ejemplos de funciones de activación utilizadas en CNN. A la izquierda, se muestra la función ReLU, donde se observa como deja invariante los valores positivos, estableciendo a cero los valores negativos. A la derecha, se muestra la función sigmoide, que mapea los valores de entrada a un rango entre 0 y 1. Imagen original del autor.

4. El dilema del aprendizaje

En este capítulo, como preámbulo antes de adentrarnos en el fenómeno del Deep Double Descent, presentaremos algunos conceptos básicos de la sabiduría clásica del aprendizaje automático que nos harán entender de manera más precisa el citado fenómeno. Las fuentes principales utilizadas a lo largo de este capítulo son [AMMIL12] y [Bis06].

4.1. Concepto de aprendizaje

El aprendizaje, dentro del marco del aprendizaje automático, puede considerarse como un proceso en el que se busca encontrar una función g que aproxime lo máximo posible a la función objetivo f, que describe las relaciones y patrones subyacentes entre las entradas y salidas de los datos. Dado que la función objetivo es siempre desconocida, pues, en otro caso, no habría que aprender nada al conocer directamente la función objetivo, serán los propios datos etiquetados los que nos ayuden, mediante el entrenamiento de modelos, a obtener una función aproximadora de dicha función objetivo.

En nuestro caso, de cara a trabajar con el aprendizaje supervisado, consideraremos los siguientes componentes del mismo:

- Espacio muestral $\mathcal{X}$: representa el conjunto de todas las posibles entradas x que el modelo puede recibir, tomadas de manera independiente siguiendo alguna (sin restricción) distribución de probabilidad P en $\mathcal{X}$.
- Conjunto $\mathcal{Y}$: compuesto por todas las posibles salidas (etiquetas) que el modelo debe predecir.
- Función objetivo $f : \mathcal{X} \to \mathcal{Y}$, que representa la función objetivo y desconocida que asigna cada entrada $x \in \mathcal{X}$ a una salida $y \in \mathcal{Y}$.
- Un conjunto de datos de entrenamiento $\mathcal{D}$, formado por pares (x, y) con $x \in \mathcal{X}$ y $y \in \mathcal{Y}$, donde $f(x) = y$.
- Un conjunto de hipótesis $\mathcal{H}$, donde se encontrarán todas las funciones candidatas que el modelo puede aprender para aproximar la función objetivo. Es decir, $\mathcal{H} = \{h : X \to Y / X \subseteq \mathcal{X}, Y \subseteq \mathcal{Y}\}$.
- Un algoritmo de aprendizaje $\mathcal{A}$, que es el encargado de elegir una función candidata $h \in \mathcal{H}$ que aproxime a la función objetivo f.

De este modo, el modelo de aprendizaje automático, por medio del algoritmo de aprendizaje, será el encargado de seleccionar la función candidata $g \in \mathcal{H}$ que mejor aproxime a la función objetivo f utilizando el conjunto de datos de entrenamiento disponible, con el

propósito de que la función candidata g siga replicando a la función objetivo f ante nuevos datos no disponibles.

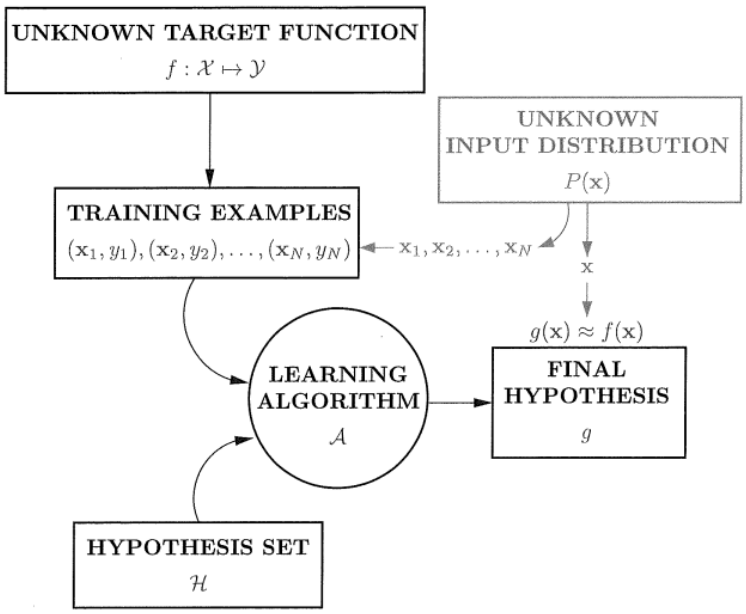

Figura 4.1.: Diagrama representando el concepto básico de aprendizaje [AMMIL12].

4.1.1. Descenso de gradiente y aprendizaje

El descenso de gradiente o *gradient descent* (GD) es la columna vertebral del aprendizaje en redes neuronales y, en general, sienta las bases para las técnicas de aprendizaje automático y aprendizaje profundo. Se trata de un algoritmo de optimización sin restricciones cuyo objetivo principal es minimizar la función de pérdida o error del modelo. Dicha función de pérdida mide qué tan lejos están las predicciones realizadas por el modelo de los valores reales, y minimizarla conllevará asociada una mejora en la precisión y capacidad de generalización del modelo.

La idea subyacente del descenso de gradiente se basa en calcular de forma iterativa el gradiente, es decir, la derivada parcial de la función de pérdida con respecto a los parámetros. A partir de este gradiente, nos desplazamos en la dirección opuesta al mismo (véase Ecuación (4.1)), ya que esta zona indica la dirección del descenso más pronunciado en la función de pérdida. De este modo, se garantiza que los parámetros se ajusten para minimizar progresivamente la pérdida, mejorando así el rendimiento del modelo.

A continuación se muestra la expresión del descenso de gradiente:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}) \tag{4.1}$$

donde $\mathbf{w}^{(\tau+1)}$ representa los valores de los parámetros actualizados, $\mathbf{w}^{(\tau)}$ representa los

valores de los parámetros antes de la actualización, η es un hiperparámetro, denominado tasa de aprendizaje o *learning rate*, que controla el tamaño del paso en cada actualización y $\nabla E(\mathbf{w}^{(\tau)})$ indica el gradiente de la función de pérdida con respecto a los parámetros, es decir, la dirección y magnitud en la que la función de pérdida aumenta de manera más rápida.

Es importante destacar que la tasa de aprendizaje (η) desempeña un papel esencial en el proceso de optimización. Si se elige un valor demasiado pequeño de la misma, el modelo podría tardar mucho en converger (véase Figura 4.2), requiriendo un número elevado de épocas o iteraciones para alcanzar un mínimo adecuado de la función de pérdida. Por el contrario, si se selecciona un valor demasiado grande, el modelo podría no converger e incluso divergir, oscilando alrededor del mínimo sin lograr estabilizarse o incluso aumentando la pérdida. Es por esto que, la mejor estrategia suele consistir en utilizar un valor grande al inicio del entrenamiento, para acelerar el entrenamiento, y reducirlo progresivamente a medida que avanza, con el objetivo de no divergir y estabilizarnos en el mínimo.

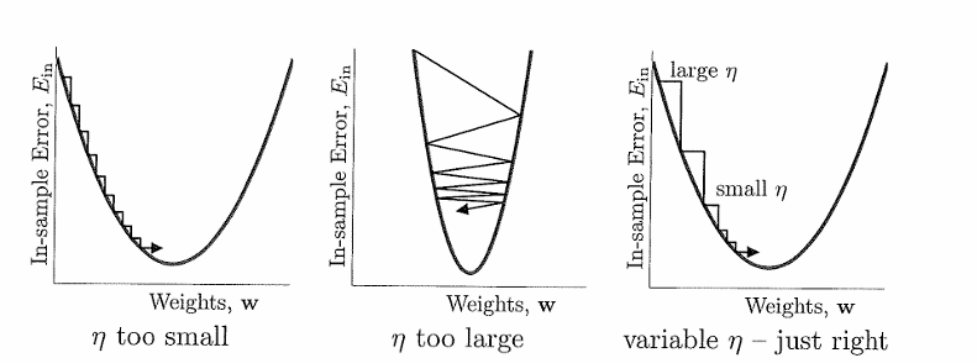

Figura 4.2.: Distintas tasas de aprendizaje para el descenso de gradiente [AMMIL12]. En la primera imagen, una tasa de aprendizaje pequeña lleva a una convergencia lenta con muchas actualizaciones. En la imagen central, una tasa demasiado grande provoca saltos bruscos que pueden impedir la convergencia. En la última imagen, una tasa variable comienza con un valor grande para avanzar rápido y disminuye progresivamente, logrando una convergencia rápida y estable.

Asimismo, existen variantes del descenso de gradiente, como el descenso de gradiente estocástico (SGD), que actualiza los parámetros utilizando cada ejemplo de entrenamiento, lo que provoca que sea muy lento cuando trabajamos con grandes volúmenes de datos. Por otro lado, nos encontramos el descenso de gradiente por lotes (Batch Gradient Descent), que utiliza el conjunto completo de datos de entrenamiento para calcular el gradiente y actualizar los parámetros, lo que permite realizar una convergencia más estable y precisa. Sin embargo, suele ser más costoso tanto computacionalmente como en términos de tiempo. Es por esto que, en el desarrollo de este proyecto, se utilizará el descenso de gradiente por mini-lotes (Mini-batch Gradient Descent), que combina lo mejor de ambos métodos, ya que utiliza subconjuntos de datos de entrenamiento para realizar la actualización de parámetros.

En resumen, el descenso de gradiente permite que la red neuronal mejore sus predicciones a lo largo de múltiples iteraciones o épocas. Cada vez que el modelo procesa un conjunto

de datos, calcula el error y ajusta sus parámetros para aprender de los errores cometidos, repitiéndose este proceso hasta que la función de pérdida alcanza un valor mínimo aceptable.

4.1.1.1. Aprendizaje en una red neuronal

Como se comentó en la Sección 3.2, la primera fase del aprendizaje consiste en la transmisión de la información desde la capa de entrada hacia la capa de salida, pasando por las capas ocultas, proceso conocido como propagación hacia delante o *forward pass*. Durante este proceso, las entradas se multiplican por los pesos de la red, se suman los sesgos o *biases*, consistente en un parámetro adicional que se suma al resultado de la combinación lineal antes de pasar por la función de activación, cuya función principal es permitir que el modelo ajuste su salida de manera más flexible, sin estar forzado a pasar por el origen de coordenadas y, finalmente, se aplican las funciones de activación para introducir no linealidad. Este flujo de datos permite que el modelo genere una predicción para cada entrada.

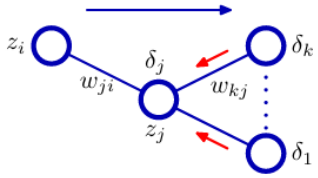

Figura 4.3.: Proceso de aprendizaje mediante descenso de gradiente [Biso6]. Dado un lote de entrenamiento, se propraga la información hacia delante, se calculan las activaciones y el error de salida. Seguidamente, se realiza el paso hacia atrás, calculando el error δ_j de cada unidad j en cada capa. Finalmente, se actualizan los parámetros utilizando el gradiente calculado.

Una vez obtenida la predicción, se calcula la función de pérdida para evaluar la discrepancia entre la predicción y el valor real. De esta manera, podemos describir la salida resultante de cualquier neurona mediante la siguiente expresión, extraída de [Pri23]:

$$h_d = \phi\left(w_{d0} + \sum_{i=1}^{D_i} w_{di} x_i\right)$$

donde d hace referencia a la neurona en dicha posición, ϕ representa una función de activación no lineal, $x_i \in \mathbb{R}^{D_i}$ representa la entrada multidimensional, donde D_i es el número de características de entrada (en este caso consideramos nuestro conjunto de datos de entrenamiento $\mathcal{D}$ como un subconjunto de $\mathbb{R}^{D_i}$) y w_{di} con $i \in \{0, 1, \ldots, D_i\}$ representa los pesos que conectan la entrada x_i con la neurona d, donde para $i = 0$ se obtiene el término del sesgo.

Seguidamente, se inicia la segunda fase del aprendizaje, cuyo objetivo es ajustar los parámetros de la red para minimizar ese error. Para ello, se utiliza retropropagación del error o *backpropagation*, que calcula el gradiente de la función de pérdida con respecto a los pesos y sesgos, mediante la regla de la cadena del cálculo diferencial. Posteriormente, el algoritmo

de descenso de gradiente actualiza los pesos en la dirección opuesta al gradiente.

En conlusión, el forward pass y la backpropagation trabajan de manera conjunta para permitir que la red neuronal aprenda de manera efectiva. El forward pass genera las predicciones para los datos de entrada, la función de pérdida evalúa el error cometido en dichas predicciones, la backpropagation calcula la contribución de cada unidad a dicho error (véase Figura 4.3) y, por último, el descenso de gradiente ajusta los parámetros para mejorar las predicciones futuras, repitiéndose este ciclo a lo largo de múltiples iteraciones.

4.2. Bias-variance tradeoff

El objetivo del aprendizaje radica en obtener un bajo error de prueba o generalización (E_{out}) sobre datos desconocidos, lo que implicará que hemos conseguido aproximar de buena manera la función objetivo f. La capacidad para conseguir un bajo error de generalización está ligada directamente al conjunto de hipótesis ($\mathcal{H}$), donde si nuestro conjunto es suficientemente grande tendremos una mayor probabilidad de aproximar la función objetivo f, al disponer de un mayor número de funciones candidatas. Sin embargo, si nuestro conjunto $\mathcal{H}$ es demasiado grande, puede darse el caso de que, al elegir una de las funciones candidatas (usando nuestro conjunto de entrenamiento), dicha función no sea la que mejor aproxime a la función objetivo, lo que provoque un mayor error de generalización. A este problema se le conoce como el problema del *equilibrio entre aproximación y generalización*. Adicionalmente, el conjunto de hipótesis ideal sería el formado únicamente por la función objetivo, es decir, $\mathcal{H} = \{f\}$.

El análisis sesgo-varianza o *bias-variance analysis* busca descomponer el error de generalización en dos términos principales:

1. Cómo de bien puede $\mathcal{H}$ aproximar a la función objetivo f en general, no solo en la muestra.
2. Hasta qué punto podemos acercanos a una buena función candidata $g \in \mathcal{H}$.

Adicionalmente, el error de generalización E_{out} incluye un término adicional conocido como *ruido*. Este término hace referencia al error irreducible que se encuentra de manera natural en los datos y que, generalmente, es debido a factores fuera del control del modelo, tales como mediciones imprecisas o variables no modeladas. Dado que este tipo de error es inevitable y no puede ser reducido, no se considera relevante en la descomposición del error. Sin embargo, cabe destacar que este ruido suele ser una limitación fundamental de la generalización del modelo.

4.3. Formulación matemática del E_{out}

A continuación, detallaremos matemáticamente las componentes del error fuera de la muestra o de generalización. Para ello y con objeto de simplificar la descomposición del E_{out} de

manera limpia en los dos términos principales citados anteriormente, vamos a considerar un problema de regresión que no presenta ruido en los datos, utilizando el error cuadrático como medida de evaluación del error.

Sea $g^{(\mathcal{D})} \in \mathcal{H}$ nuestra función candidata elegida (hipótesis final), que dependerá del conjunto de datos utilizado ($\mathcal{D}$). Destacamos que, dado cualquier otro conjunto de datos, encontraremos una hipótesis final distinta.

El *error de generalización o error fuera de la muestra*, definido como la diferencia entre la predicción del modelo y los valores reales en un conjunto de datos no visto durante el entrenamiento, quedaría expresado de la siguiente forma:

$$E_{out}(g^{(\mathcal{D})}) = \mathbb{E}_x[(g^{(\mathcal{D})}(x) - f(x))^2], \quad x \notin \mathcal{D} \tag{4.2}$$

donde $\mathbb{E}_x$ denota el valor esperado con respecto a x (basado en la distribución de probabilidad del espacio de entrada $\mathcal{X}$), con el objetivo de obtener el valor esperado del error en todo el espacio.

Dado que la ecuación (4.2) depende de un conjunto de datos en particular, podemos eliminar esta dependencia del conjunto de datos utilizado tomando el valor esperado de dicho error con respecto a todos los conjuntos de datos:

$$\mathbb{E}_{\mathcal{D}}[E_{out}(g^{(\mathcal{D})})] = \mathbb{E}_{\mathcal{D}}[\mathbb{E}_x[(g^{(\mathcal{D})}(x) - f(x))^2]].$$

Cambiamos ahora el orden de las esperanzas dado que, en realidad, estamos integrando y cambiando el orden de integración, que podemos realizarlo dado que el integrando $(g^{(\mathcal{D})}(x) - f(x))^2$ es estrictamente no negativo:

$$\mathbb{E}_{\mathcal{D}}[E_{out}(g^{(\mathcal{D})})] = \mathbb{E}_x[\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - f(x))^2]]. \tag{4.3}$$

Nos centramos en calcular $\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - f(x))^2]$, olvidándonos por ahora del valor esperado sobre x, dado que nos interesa calcular la esperanza con respecto a $(\mathcal{D})$ hasta obtener una descomposición limpia del error. Para ello, vamos a definir el concepto de hipótesis promedio $\bar{g}(x)$ de la siguiente manera:

$$\bar{g}(x) = \mathbb{E}_{\mathcal{D}}[g^{(\mathcal{D})}(x)] \tag{4.4}$$

que corresponde al valor esperado de todas las hipótesis que podemos obtener al aprender de los distintos conjuntos de datos que utilicemos. Destacamos que, en la ecuación (4.4), tenemos x (punto de prueba) fijo, por lo que $g^{(\mathcal{D})}(x)$ es una variable aleatoria determinada por la elección de nuestros datos, donde si tomamos distintos conjuntos de datos, obtendremos distintos valores para la hipótesis en el punto x fijado. Sin embargo, en la realidad nunca dispondremos de esta hipótesis promedio, pues tendríamos un número infinito de conjuntos de datos distintos. Además, cabe destacar que la hipótesis promedio no tiene asegurada su pertenencia al conjunto de hipótesis $(\mathcal{H})$, aunque sea el promedio de hipótesis pertenecientes

a $\mathcal{H}$.

Continuando con la descomposición, tenemos el siguiente resultado:

$$\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - f(x))^2] = \mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - \bar{g}(x) + \bar{g}(x) - f(x))^2] \tag{4.5}$$

donde se ha sumado y restado la misma cantidad ($\bar{g}(x)$) para simplificar la descomposición y, por las propiedades de la esperanza, esto no afecta al valor esperado buscado.

Seguidamente, continuamos desarrollando el término cuadrático de la ecuación (4.5):

$$\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - \bar{g}(x))^2 + (\bar{g}(x) - f(x))^2 + 2(g^{(\mathcal{D})}(x) - \bar{g}(x))(\bar{g}(x) - f(x))].$$

Dado que estamos realizando el valor esperado con respecto a $\mathcal{D}$, de la ecuación anterior tenemos que $(\bar{g}(x) - f(x))$ es una constante. Por tanto, dado que el valor esperado de una constante es la propia constante, para obtener el valor esperado del término cruzado solo necesitamos conocer el valor esperado de $(g^{(\mathcal{D})}(x) - \bar{g}(x))$:

$$\mathbb{E}_{\mathcal{D}}[g^{(\mathcal{D})}(x) - \bar{g}(x)] = \mathbb{E}_{\mathcal{D}}[g^{(\mathcal{D})}(x)] - \mathbb{E}_{\mathcal{D}}[\bar{g}(x)] = \bar{g}(x) - \bar{g}(x) = 0.$$

Finalmente, de la ecuación (4.5) nos queda la siguiente expresión:

$$\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - f(x))^2] = \mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - \bar{g}(x))^2] + (\bar{g}(x) - f(x))^2$$

donde hemos usado que el valor esperado de una constante $(\bar{g}(x) - f(x))$ es igual a dicha constante.

Por consiguiente, hemos obtenido una descomposición de la ecuación (4.5) en dos términos que serán los asociados a los términos de varianza (*variance*) y sesgo (*bias*) respectivamente.

1. $\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - \bar{g}(x))^2]$: nos indica cómo de lejos se encuentra nuestra hipótesis, $g^{(\mathcal{D})}(x)$, obtenida de un conjunto de datos particular de la mejor (promedio) hipótesis, $\bar{g}(x)$, que podemos obtener utilizando nuestro conjunto de hipóstesis $\mathcal{H}$. Este es el término asociado a la varianza de x (**var(x)**).
2. $(\bar{g}(x) - f(x))^2$: nos indica cómo de lejos se encuentra dicha hipótesis ideal, $\bar{g}(x)$, de la función objetivo $f(x)$. Este es el término asociado al sesgo de x (**bias(x)**).

Por tanto, volviendo a la ecuación (4.3) nos queda:

$$\mathbb{E}_{\mathcal{D}}[E_{out}(g^{(\mathcal{D})})] = \mathbb{E}_x[\mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - f(x))^2]] = \mathbb{E}_x[\textbf{var(x)} + \textbf{bias(x)}]$$

donde denotaremos por **sesgo** a $\mathbb{E}_x[\textbf{bias(x)}]$ y por **varianza** a $\mathbb{E}_x[\textbf{var(x)}]$.

Observación 4.1. Cuando hay ruido presente en los datos de entrenamiento, es decir, $y(x) = f(x) + \epsilon(x)$ y, además, ϵ es una variable aleatoria con media 0 y varianza σ^2, entonces

$$E_{out}(g^{(\mathcal{D})}) = \mathbb{E}_{x,y}[g^{(\mathcal{D})}(x) - y(x)^2]$$

y, en este caso, se verifica

$$\mathbb{E}_{\mathcal{D}}[E_{out}(g^{(\mathcal{D})})] = \sigma^2 + bias + var.$$

Demostración. Dado que asumimos que ϵ es una variable aleatoria con media 0, obtenemos que $\mathbb{E}[\epsilon(x)] = 0$.

$$\mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(x) - y(x))^2] = \mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(x) - f(x)) - \epsilon(x)^2] \tag{4.6}$$

donde dado que y depende del ruido, consideramos también el valor esperado con respecto a ϵ que afectará únicamente a y. Procediendo de manera similiar a la Ecuación (4.5), obtenemos

$$\mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(x) - \bar{g}(x) + \bar{g}(x) - f(x) - \epsilon(x))^2]$$

y, tras realizar operaciones, llegamos a

$$\mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(x) - f(x))^2] + 2\mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(x) - f(x))\epsilon(x)] + \mathbb{E}_{\mathcal{D},\epsilon}[\epsilon^2(x)]$$

donde el primer sumando no depende de ϵ, el tercer sumando no depende de $\mathcal{D}$ y el segundo sumando puede expresarse de la siguiente forma:

$$2\mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(x) - f(x))\epsilon(x)] = 2\mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(x) - f(x))]\mathbb{E}_{\mathcal{D},\epsilon}[\epsilon(x)].$$

donde el valor esperado con respecto a ϵ no depende de $\mathcal{D}$. De esta manera y conociendo que $\mathbb{E}_{\epsilon}[\epsilon(x)] = 0$ y $\mathbb{E}_{\epsilon}[\epsilon(x)^2] = \sigma^2$, obtenemos que la Ecuación (4.6) puede expresarse de la siguiente forma

$$\mathbb{E}_{\mathcal{D},\epsilon}[(g^{(\mathcal{D})}(x) - y(x))^2] = \mathbb{E}_{\mathcal{D}}[(g^{(\mathcal{D})}(x) - f(x))^2] + \sigma^2.$$

Finalmente, aplicando el valor esperado sobre x a la expresión anterior, obtenemos el resultado buscado.

□

Como conclusión de esta sección, detallaremos algunos aspectos clave del análisis del sesgo y de la varianza:

- Aunque el análisis de sesgo-varianza se basa en la medida del error cuadrático, el algoritmo de aprendizaje utilizado por el modelo no tiene que basarse en minimizar el error cuadrático. Es decir, se puede usar cualquier otro criterio (función de pérdida) para producir $g^{(\mathcal{D})}$ basado en el conjunto de datos $\mathcal{D}$ y, una vez que tenemos $g^{(\mathcal{D})}$, calculamos su sesgo y varianza utilizando el error cuadrático.

- El sesgo y la varianza no pueden ser calculados en la práctica, ya que dependen de la función objetivo y de la distribución de probabilidad de la entrada (ambas desconocidas), por lo que su descomposición resulta de utilidad como una herramienta conceptual a la hora de entender y desarrollar un modelo.

4.4. Equilibrio clásico entre sesgo y varianza

Siguiendo con los resultados obtenidos en la Sección 4.3, nuestro objetivo ahora es minimizar el error fuera de la muestra, inducido por tres componentes: ruido, sesgo y varianza. Dado que, como se ha comentado previamente, el rudio es irreducible y no hay nada que podamos hacer para evitarlo, nos centraremos en intentar reducir los dos términos principales reducibles del error: el sesgo y la varianza.

Conocemos que el sesgo viene definido por la medida en la que la predicción ideal obtenida de todos los conjuntos de datos difiere de la función objetivo, o expresado de manera similar, el sesgo es el resultado de la incapacidad del modelo para describir la función objetivo. Este resultado sugiere que podemos reducir este término de error haciendo que nuestro modelo sea más flexible, es decir, aumentando nuestro conjunto de hipótesis $\mathcal{H}$ (haciéndolo más complejo), con el objetivo de disponer de un mayor número de funciones candidatas para forzar que la hipótesis promedio se aproxime lo máximo posible a la función objetivo.

Por otro lado, la varianza viene a ofrecernos la medida en la que varían las hipótesis para distintos conjuntos de datos con respecto a la hipótesis ideal, o expresado de manera similar, la varianza evalúa cómo de sensible es una hipótesis a la selección específica del conjunto de datos $\mathcal{D}$. Este análisis revela que podemos reducir este término de error disminuyendo el conjunto de hipótesis ya que, en un espacio de hipótesis restringido, al haber menos hipótesis, las hipótesis son menos sensibles a las variaciones en el conjunto de datos. Como resultado, al cambiar el conjunto de datos para seleccionar una hipótesis, es más probable que se elijan hipótesis similares.

En consecuencia, observamos una dependencia de ambos términos de error con respecto a la complejidad del conjunto de hipótesis ($\mathcal{H}$) utilizado. Esta dependencia viene dada por:

- Al aumentar la complejidad del conjunto de hipótesis, es decir, al incrementar su tamaño, el sesgo disminuirá, pero la varianza irá aumentado.
- Si reducimos la complejidad del conjunto de hipótesis, es decir, disminuimos su tamaño, el sesgo irá aumentando, pero la varianza disminuirá.

De este modo, el equilibrio buscado en esta sección con objeto de minimizar el error fuera de la muestra viene ligado a la elección de un conjunto de hipótesis ($\mathcal{H}$) suficientemente complejo como para acercanos a la función objetivo y suficientemente simple como para no disponer de una cantidad excesiva de hipótesis que induzcan un alto grado de varianza. Este equilibrio se podrá conseguir mediante diversas técnicas, como la regularización, que se explicarán en secciones posteriores.

4.4.1. Curva de aprendizaje

Para finalizar este capítulo, introduciremos el concepto de curva de aprendizaje o *learning curve*, que será la principal herramienta que utilizaremos a lo largo de todo el proyecto para analizar el rendimiento de los distintos modelos que utilicemos.

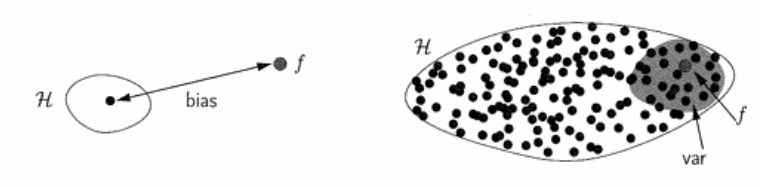

Figura 4.4.: Distintos casos del conjunto de hipótesis y de la función objetivo [AMMIL12]. A la izquierda observamos como nuestro conjunto de hipótesis presenta únicamente una función candidata, alejada de la función objetivo f, lo que implica un alto sesgo y una varianza nula. A la derecha observamos un conjunto de hipótesis con numerosas funciones candidatas que incluye a la función objetivo f, por lo que el sesgo es muy cercano a 0 y la varianza es grande.

En primer lugar y de manera análoga a la ecuación (4.2), definimos el *error de entrenamiento o error dentro de la muestra (E_{in})* como la diferencia entre la predicción del modelo y los valores reales del conjunto de datos de entrenamiento, es decir

$$E_{in}(g^{(\mathcal{D})}) = \mathbb{E}_x[(g^{(\mathcal{D})}(x) - f(x))^2], \quad x \in \mathcal{D}.$$

Por consiguiente, después de aprender de un conjunto particular de datos $\mathcal{D}$ de tamaño N, la hipótesis final elegida $g^{(\mathcal{D})}$ tendrá error de entrenamiento ($E_{in}(g^{(\mathcal{D})})$) y error de generalización ($E_{out}(g^{(\mathcal{D})})$), ambos dependiendo del conjunto de datos utilizado. Como se comentó en la Sección 4.3, al realizar la esperanza con respecto a todos los conjuntos de datos de dichos errores obtenemos los errores esperados $\mathbb{E}_{\mathcal{D}}[E_{in}(g^{(\mathcal{D})})]$ y $\mathbb{E}_{\mathcal{D}}[E_{out}(g^{(\mathcal{D})})]$, donde ambos errores son funciones del tamaño del conjunto de datos (N).

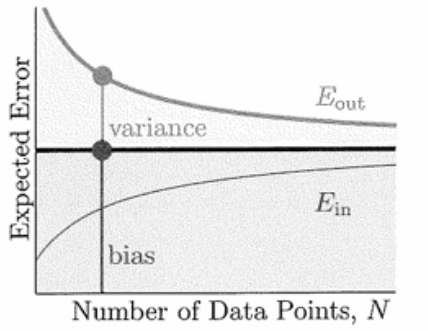

Figura 4.5.: Ejemplo de curva de aprendizaje tradicional [AMMIL12]. Observamos la curva de aprendizaje de un modelo con respecto al tamaño del conjunto de datos. Se puede comprobar como el E_{out} decrece, mientras que el E_{in} crece, ambos de manera monótona. Además, se puede observar la descomposición sesgo-varianza, donde la línea central en negrita denota la hipótesis promedio.

Denominaremos *curva de aprendizaje* a la representación gráfica que muestra la relación entre el rendimiento de un modelo y el tamaño del conjunto de datos utilizado para su entrena-

miento, es decir, a la gráfica que incluye los errores esperados $\mathbb{E}_{\mathcal{D}}[E_{in}(g^{(\mathcal{D})})]$ y $\mathbb{E}_{\mathcal{D}}[E_{out}(g^{(\mathcal{D})})]$ como función de N.

Además, dado que la curva de aprendizaje está ligada a los errores esperados del modelo tanto dentro como fuera de la muestra, podríamos analizar el equilibrio sesgo-varianza directamente sobre dicha gráfica. Sin embargo, para llevar a cabo dicho análisis, necesitaríamos conocer la hipótesis promedio $\bar{g}$ que, como sabemos de secciones anteriores, es imposible de calcular. No obstante, si tuvieramos dicha hipótesis promedio, podríamos realizar el análisis (véase Figura 4.5), teniendo en cuenta que el E_{out} es suma de sesgo y varianza (obviando el término de ruido).

Sin embargo, de cara a analizar el doble descenso a lo largo del proyecto, **no utilizaremos directamente la curva de aprendizaje** tal y como se ha definido, dado que nuestro objetivo es analizar el error con respecto a la capacidad del modelo y no en función del número de datos utilizados. Es por esto que, nuestra curva de aprendizaje será una modificación de la curva de aprendizaje original, donde en el eje X de la gráfica aparecerá indistintamente la capacidad o el número de épocas de entrenamiento del modelo y en el eje Y el error esperado (véase Figura 4.6), tanto dentro como fuera de la muestra, para un número fijo de datos de entrenamiento.

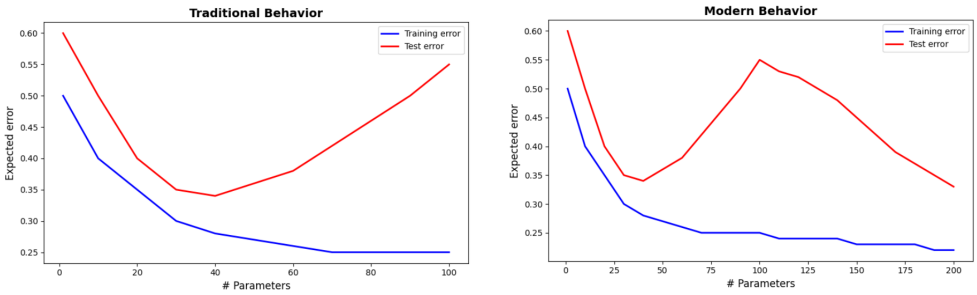

Figura 4.6.: Ejemplos de curvas de aprendizaje modificadas para este proyecto. A la izquierda se muestra la curva clásica de aprendizaje, donde el error de test aumenta llegado a un cierto punto. A la derecha, la curva moderna refleja que, al aumentar la complejidad del modelo, el error de test vuelve a disminuir. Imagen original del autor.

4.5. Underfitting y overfitting

Consideremos nuevamente el problema del aprendizaje supervisado que estamos tratando, consistente en encontrar una "buena" función aproximadora $g \in \mathcal{H}$ basada en los datos de entrenamiento $\mathcal{D}$. Además, se asume que estos datos provienen de una determinada distribución de probabilidad, es decir, $\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ es una colección de n copias idénticas e idénticamente distribuidas de las variables aleatorias (X, Y) que toman valores en $\mathcal{X} \times \mathcal{Y}$ y que siguen una distribución de probabilidad conjunta $P[X, Y]$, permitiendo modelar incertidumbre en las predicciones. De igual manera, se asume la existencia de una función real no negativa $L(g(X), Y)$, denominada *función de pérdida*, que es la encargada de evaluar

la diferencia entre la predicción realizada por la función candidata g y el verdadero valor y.

En lo que prosigue a lo largo de esta sección restringiremos nuestro conjunto de hipótesis $\mathcal{H}$ a una determinada clase de funciones, es decir, $\mathcal{H}$ quedaría fijo.

Definición 4.2 (Riesgo real)**.** El riesgo real asociado a la función candidata $g \in \mathcal{H}$ viene definido como el valor esperado de la función de pérdida, esto es

$$L(g) = \mathbb{E}[L(g(X), Y)] = \int_{\mathcal{X}} L(g(X), Y)\, dP[X, Y] = P[g(X) \neq Y].$$

El riesgo real es también denominado como *error de generalización*.

Por tanto, el objetivo final de un algoritmo de aprendizaje, como se comentó en secciones anteriores, es encontrar la función candidata g^* entre una clase fija de funciones del conjunto de hipótesis $\mathcal{H}$ para la cual el riesgo real sea mínimo, es decir

$$g^* = \arg\min_{g \in \mathcal{H}} L(g).$$

No obstante, en la práctica y por lo general, el riesgo real no puede ser calculado porque la distibución conjunta P[X, Y] es desconocida por el algoritmo de aprendizaje. Es por esto que tenemos que recurrir a un cálculo estimado del mismo, denominado *riesgo empírico*, calculado haciendo uso de la media de la función de pérdida sobre el conjunto de entrenamiento.

Definición 4.3 (Riesgo empírico)**.** El riesgo empírico asociado a la función candidata $g \in \mathcal{H}$ sobre el conjunto de entrenamiento $\mathcal{D}$ viene definido por

$$L_{emp}(g) = \frac{1}{n} \sum_{i=1}^{n} L(g(X_i), Y_i).$$

Ligado a este concepto surge el *principio de minimización del riesgo empírico* [Vap91], que establece que el algoritmo de aprendizaje $\mathcal{A}$ debe elegir una función candidata $\hat{g}$ sobre el conjunto de hipótesis $\mathcal{H}$ que minimice el riesgo empírico sobre el conjunto de entrenamiento $\mathcal{D}$, es decir

$$\hat{g} = \arg\min_{g \in \mathcal{H}} L_{emp}(g). \tag{4.7}$$

Finalmente, la diferencia entre cualquier función candidata $g \in \mathcal{H}$ y la mejor función candidata g^* puede descomponerse de la siguiente manera [LT24]:

$$L(g) - L(g^*) = \underbrace{L(g) - \inf_{g \in \mathcal{H}} L(g)}_{\text{error de estimación}} + \underbrace{\inf_{g \in \mathcal{H}} L(g) - L(g^*)}_{\text{error de aproximación}}$$

donde, aparte del error de estimación y del error de aproximación, existe otra fuente de error conocida como *error de optimización* que indica la diferencia entre el riesgo de la función candidata, devuelto por el procedimiento de optimización (en nuestro caso el descenso de gradiente), y un minimizador del riesgo empírico. De esta manera, el algoritmo de aprendizaje $\mathcal{A}$ definido por el principio de minimización del riesgo empírico consiste en resolver el problema de optimización dado por la ecuación (4.7).

Proposición 4.4. *Para cualquier minimizador del riesgo empírico $\hat{g}$, el error de estimación verifica*

$$L(\hat{g}) - \inf_{g \in \mathcal{H}} L(g) \leq 2 \sup_{g \in \mathcal{H}} |L_{emp}(g) - L(g)|.$$

Demostración. Partiendo de la siguiente desigualdad

$$L(\hat{g}) - \inf_{g \in \mathcal{H}} L(g) \leq |L(\hat{g}) - L_{emp}(\hat{g})| + |L_{emp}(\hat{g}) - \inf_{g \in \mathcal{H}} L(g)|$$

con el primer sumando verificando

$$|L(\hat{g}) - L_{emp}(\hat{g})| \leq \sup_{g \in \mathcal{H}} |L_{emp}(g) - L(g)|$$

dado que $\hat{g} \in \mathcal{H}$. Por otra parte, el segundo sumando verifica

$$|L_{emp}(\hat{g}) - \inf_{g \in \mathcal{H}} L(g)| = |\inf_{g \in \mathcal{H}} L_{emp}(g) - \inf_{g \in \mathcal{H}} L(g)| \leq \sup_{g \in \mathcal{H}} |L_{emp}(g) - L(g)|$$

y, sumando ambas desigualdades, se obtiene el resultado deseado.

□

De este modo, la estrategia a seguir en el aprendizaje automático es la de encontrar un correcto conjunto de hipótesis $\mathcal{H}$ para mantener ambos errores lo más pequeños posible. Es por esto que, dependiendo del conjunto $\mathcal{H}$ elegido, podemos encontrarnos las siguientes situaciones:

1. Si el conjunto $\mathcal{H}$ es muy "pequeño", ninguna función candidata será capaz de capturar la complejidad de los datos de entrenamiento y no será capaz de aproximarse a g^*.
2. Si el conjunto $\mathcal{H}$ es muy "grande", el límite de la Proposición 4.4 (máxima brecha de generalización sobre $\mathcal{H}$) aumentará y la función candidata $\hat{g}$ elegida como minimizadora del riesgo empírico puede generalizar de forma no adecuada aún teniendo un error de entrenamiento bajo.

donde se pone de manifiesto la fuerte dependencia de la minimización empírica del riesgo (error del modelo) del conjunto de hipótesis elegido.

Parte II.

Estado del Arte

5. Evolución del Deep Double Descent

El *Deep Double Descent*, al ser un fenómeno estrechamente relacionado al avance tecnológico y al crecimiento en la capacidad y el tamaño de los modelos de aprendizaje profundo, ha despertado un mayor interés en los últimos años, como puede observarse en la Figura 5.1. No obstante, aunque se percibe, durante los últimos años, una tendencia al alza del número de publicaciones que hacen referencia al citado fenómeno, únicamente encontramos un total 143 publicaciones en Scopus[1]. Esta reciente relevancia del fenómeno ilustra el carácter **innovador** y **pionero** de este proyecto.

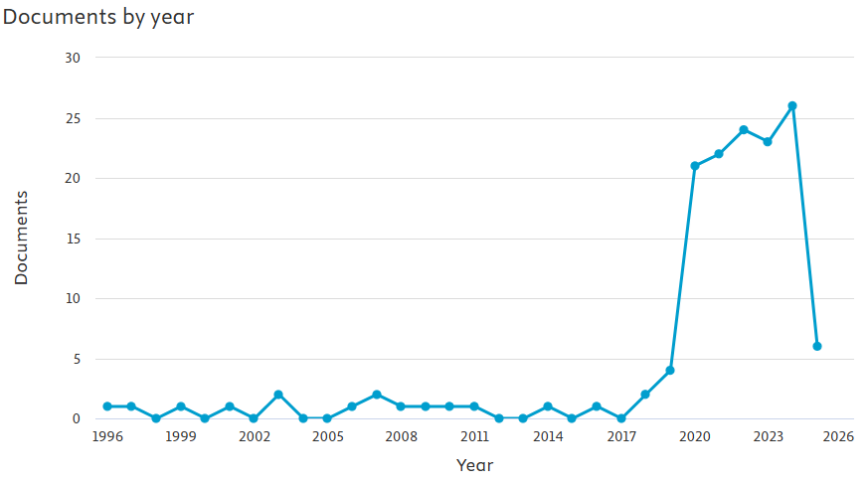

Figura 5.1.: Número de publicaciones relativas al Deep Double Descent en función del año de publicación.

Sin embargo, aunque la cantidad de artículos científicos sobre este fenómeno aún sea limitada, el interés por parte de investigadores y científicos está creciendo rápidamente. Incluso en ausencia de publicaciones científicas formales, se continúan obteniendo nuevos resultados, tanto teóricos como prácticos, que continúan enriqueciendo nuestra comprensión del problema. De hecho, en Scopus[2], podemos encontrar 105 preimpresiones relacionadas con el fenómeno.

A pesar de la disponibilidad de artículos que abordan este fenómeno, la mayoría de ellos no proporcionan una explicación detallada del mismo, centrándose generalmente en casos prácticos y dejando de lado el análisis teórico subyacente, que se encuentra detrás de los resultados obtenidos, lo que limita la comprensión completa de ciertos aspectos del fenómeno.

[1]Encontradas 143 publicaciones a fecha 10 de febrero de 2025 usando la consulta: TITLE-ABS-KEY (deep AND double AND descent).
[2]Encontradas 105 preimpresiones a fecha 10 de febrero de 2025 usando la consulta: TITLE-ABS-KEY (deep AND double AND descent).

5.1. Origen y primeras manifestaciones del fenómeno

Aunque la primera publicación formal sobre el *Deep Double Descent* data de 1996, como se observa en la Figura 5.1, el fenómeno parece haber sido descrito por primera vez en la literatura de la física teórica en 1989, aunque de manera superficial y sin la denominación actual. Por ello, nos remontamos hasta 1995 cuando Opper presenta este resultado en su artículo "Statistical Mechanics of Generalization" [Opp95], a través del uso de una red neuronal formada por un perceptrón de una sola capa con una función de activación lineal conocida como ADALINE [WH60], y, más tarde, en su revisión del artículo en 2001 "Learning to Generalize" [Opp01].

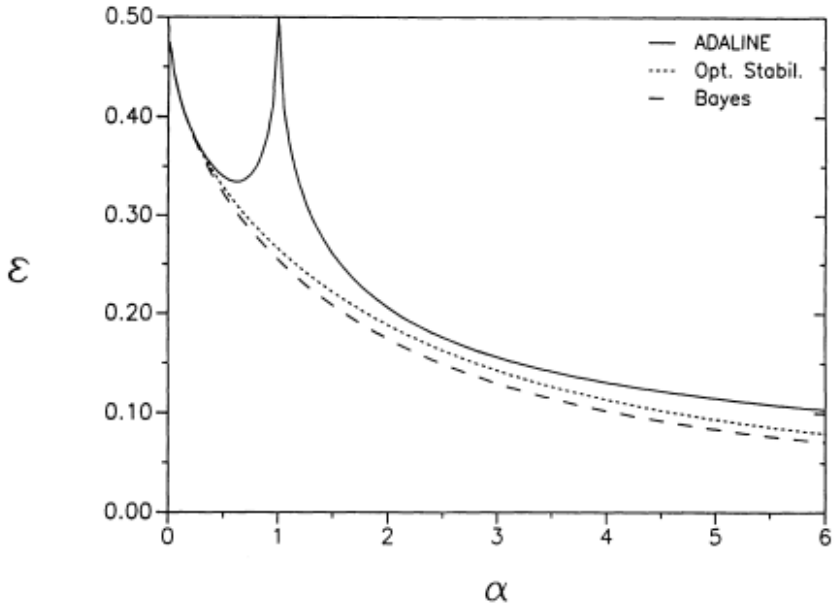

Figura 5.2.: Comparación del error de generalización (ϵ) para distintos modelos en función de la fracción entre el número de ejemplos aprendido y el número de parámetros (α). Se observa la curva del doble descenso para el modelo ADALINE [Opp95].

En ambos artículos, Opper presenta de manera gráfica el fenómeno del doble descenso utilizando un modelo neuronal lineal (denominado así por el uso de funciones de activación lineales), aunque sin profundizar demasiado en este resultado (véase Figura 5.2). En sus representaciones, el error de generalización (ϵ) se muestra en función del número de ejemplos de entrenamiento (n) y el número de parámetros (P), donde $\alpha = \frac{n}{P}$. Se observa que el error de generalización alcanza su pico cuando α se aproxima a 1, es decir, cuando el número de ejemplos de entrenamiento es similar al número de parámetros del modelo.

Comportamientos similares a los obtenidos por Opper también han sido reportados por Advani & Saxe en [AS17], Spigler et al. en [SGd+19] y Geiger et al. en [GSd+19]. Estos trabajos, anteriores a la formalización del fenómeno tal como se conoce hoy en día, trabajan con

redes neuronales profundas con un gran número de parámetros y estudian el comportamiento del error de generalización utilizando herramientas de física estadística inspiradas en el trabajo de Opper. Además, [SGd+19] y [GSd+19] proponen una conexión entre el fenómeno y la transición de jamming propia de la física estadística, mostrando por qué los modelos pueden llegar a generalizar mejor después del pico del error de prueba. Sin embargo, fue Duin (2000) en [Dui00] (véanse Figuras 6 y 7) el primero en mostrar curvas de generalización, utilizando datos del mundo real, bastante similares a las curvas del doble descenso que tenemos hoy en día.

5.2. El nacimiento del Deep Double Descent

Iniciamos esta sección abordando el equilibrio clásico entre sesgo y varianza, un concepto fundamental en la teoría del aprendizaje automático ([GBD92], [HTF01] y [GB10]). Esta teoría sostiene que, a medida que la complejidad de un modelo aumenta, su sesgo disminuye, pero su varianza se incrementa. Como resultado, llega un punto en el que el modelo comienza a sobreajustar, lo que provoca un aumento en el error de generalización, dominado por la varianza y la tradicional curva en forma de "U" del error de generalización. De acuerdo con esta visión tradicional, una vez superado cierto umbral de complejidad, los modelos más grandes son cada vez peores y, por tanto, se busca encontrar un equilibrio en el modelo.

Sin embargo, los resultados prácticos modernos no comparten esta teoría. En la actualidad, la visión moderna entre los profesionales es que los *modelos grandes son mejores* ([KSH12], [NMB+19], [HCB+19], [SLJ+14]). Estos estudios muestran que el uso de redes neuronales con un gran número de parámetros conduce a un mejor rendimiento, evidenciando que los modelos más complejos pueden obtener resultados superiores a los modelos simples. Por ello, este trabajo se enmarca dentro del estudio de la generalización en redes neuronales profundas. En particular, se enlaza con investigaciones previas, como la de Zhang et al. (2021) en [ZBH+21], quienes argumentan que comprender el aprendizaje profundo aún requiere replantearse los paradigmas tradicionales de generalización, desafiando la noción clásica de sesgo-varianza.

El fenómeno del *Deep Double Descent* fue nombrado así, por primera vez, por Belkin et al. en [BHMM19], haciendo referencia a los dos descensos que presenta la curva del error de generalización. En este artículo, se busca unificar la teoría clásica del equilibrio sesgo-varianza con los resultados prácticos obtenidos por la teoría moderna, mediante una curva de error unificada (véase Figura 5.3).

Belkin et al. muestra la aparición del fenómeno en diversos conjuntos de datos y con distintos tipos de modelos, entre los que se incluyen los árboles de decisión y redes neuronales con dos capas ocultas. Además, ofrece una primera intuición sobre su causa, argumentando que, en la región sobreparametrizada, el modelo dispone de un mayor número de funciones candidatas compatibles con los datos. A esto se suma el efecto de la regularización implícita inducida por ciertos algoritmos de optimización, como el gradiente descendente ([SHN+24]).

Posteriormente, Nakkiran et al. ([NKB+19]) observaron que el fenómeno del doble descenso no solo dependía del tamaño del modelo, sino también del número de épocas de

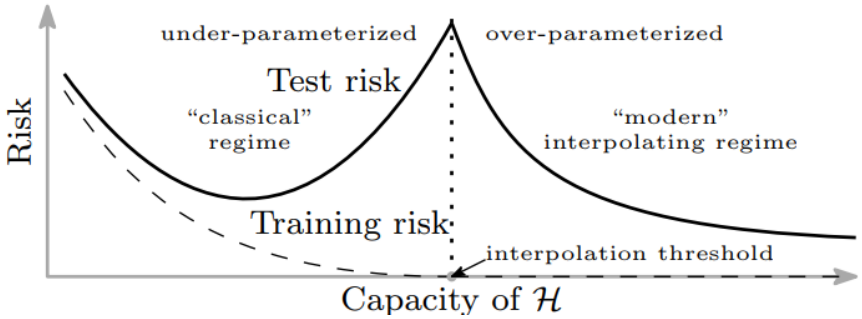

Figura 5.3.: Curvas para el error de entrenamiento (línea discontinua) y el error de generalización (línea continua). Antes del umbral de interpolación, se muestra la curva clásica en "U". Después de dicho umbral, se observa la curva del error moderna, induciendo el doble descenso. Imagen obtenida de [BHMM19].

entrenamiento. Además, unificaron estos resultados mediante la introducción de una nueva medida de complejidad para un modelo: la **complejidad efectiva del modelo**, y conjeturaron bajo qué condiciones este fenómeno podía ocurrir en función de dicha medida. En este mismo trabajo, también formalizaron los distintos tipos de doble descenso que pueden manifestarse: en función del número de parámetros, del número de épocas y del tamaño del conjunto de entrenamiento y que será la base conceptual sobre la que desarrollaremos este proyecto.

5.3. Avances recientes del Deep Double Descent

En los últimos años, la comprensión del fenómeno del *Deep Double Descent* ha avanzado significativamente, con nuevas investigaciones que han refinado su caracterización y explorado sus implicaciones en redes neuronales profundas.

Uno de los principales avances en el campo del aprendizaje estadístico ha sido la reconsideración de los límites de la sabiduría clásica sobre el sesgo y la varianza, especialmente al analizar el impacto del uso de un gran número de parámetros en el aprendizaje ([ZBH+21], [CJvdS23]). Por otro lado, Schaeffer et al. (2023) ([SKR+23]) realizaron los primeros estudios teóricos y experimentales enfocados en identificar y analizar las posibles causas y factores que pueden desencadenar el fenómeno.

Otras líneas de investigación han explorado cómo ciertas técnicas pueden mitigar el doble descenso. Por ejemplo, Yang y Suzuki (2023) en [YS24] analizaron el impacto de la regularización mediante el uso de dropout, demostrando que dicha técnica puede reducir la magnitud del segundo descenso en el error de generalización. Asimismo, Heckel y Yilmaz (2021) en [HY20] investigaron cómo el uso de la parada anticipada *o early stopping* puede influir en el fenómeno.

Desde una perspectiva más empírica, varios estudios han analizado la manifestación del fenómeno en escenarios de aprendizaje adversario. En particular, Min et al. (2021) en [MCK20]

demostraron que, en algunos casos, un aumento de los datos de entrenamiento puede mejorar la robustez del modelo, aunque también puede provocar un descenso adverso en la generalización. Asimismo, Singh et al. (2022) en [SLHS22] presentaron un análisis teórico del fenómeno en redes neuronales de tamaño finito (con un número finito de neuronas por capa y que no son capaces de memorizar el conjunto de entrenamiento), proporcionando una caracterización matemática básica que ayuda a entender los mecanismos subyacentes. Por último, Somepalli et al. (2022) en [SFB+22] investiga la reproducibilidad del aprendizaje en redes neuronales y su relación con el doble descenso.

Finalmente, investigaciones recientes han mostrado extensiones del fenómeno, pues el fenómeno del doble descenso no está necesariamente limitado a dos descensos, sino que, bajo ciertas circunstancias, pueden observarse más de dos descensos ([dSB21], [CMBK21]). Además, se ha estudiado la relación del doble descenso con otros fenómenos emergentes en el aprendizaje profundo, como es el caso del *grokking*. En este contexto, Davies et al. (2023) en [DLK23] propusieron una conexión entre ambos, sugiriendo que el aprendizaje prolongado puede llevar a una mejora abrupta de la generalización, similar a lo que ocurre en zona sobreparametrizada del fenómeno.

Parte III.

Desarrollo Teórico y Empírico

6. Análisis teórico del Deep Double Descent

6.1. Planteamiento teórico

6.1.1. Análisis intuitivo en un problema OLS

6.2. Resto de desarrollos a realizar

6.3. Aproximación no lineal

6.3.1. Analogía con el deep double descent

6.4. Conclusión

7. Análisis empírico del Deep Double Descent

7.1. Materiales y métodos

7.2. Implementación e infraestructura

7.3. Experimentos

7.4. Conclusión

Parte IV.

Conclusiones y Trabajos Futuros

8. Conclusiones

9. Trabajos futuros

Glosario

$\mathcal{H}$ Conjunto de hipótesis de un modelo.

$\mathcal{D}$ Conjunto de datos de entrenamiento.

Bibliografía

[AMMIL12] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, y Hsuan-Tien Lin. *Learning From Data*. AML-Book, 2012.

[AS17] Madhu S. Advani y Andrew M. Saxe. High-dimensional dynamics of generalization error in neural networks, 2017.

[BB23] Christopher Michael Bishop y Hugh Bishop. *Deep Learning - Foundations and Concepts*. 1 edición, 2023.

[BD09] Shai Ben-David. *Theory-Practice Interplay in Machine Learning - Emerging Theoretical Challenges*, volumen 5781 de *Lecture Notes in Computer Science*. Springer, 2009.

[BHMM19] Mikhail Belkin, Daniel Hsu, Siyuan Ma, y Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, July 2019.

[Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[Bis06] Christopher M Bishop. *Pattern Recognition and Machine Learning*, volumen 4 de *Information science and statistics*. Springer, 2006.

[Blu21] Avrim Blum. Mathematical toolkit spring 2021, lecture 5, April 13 2021. Notes based on notes from Madhur Tulsiani.

[Bry95] Włodzimierz Bryc. *The Normal Distribution: Characterizations with Applications*. Springer New York, NY, 1995.

[CJvdS23] Alicia Curth, Alan Jeffares, y Mihaela van der Schaar. A u-turn on double descent: Rethinking parameter counting in statistical learning, 2023.

[CMBK21] Lin Chen, Yifei Min, Mikhail Belkin, y Amin Karbasi. Multiple descent: Design your own generalization curve, 2021.

[Dem14] Amir Dembo. *Probability Theory: STAT310/MATH230*. CreateSpace Independent Publishing Platform, 2014.

[DLK23] Xander Davies, Lauro Langosco, y David Krueger. Unifying grokking and double descent, 2023.

[dSB21] Stéphane d'Ascoli, Levent Sagun, y Giulio Biroli. Triple descent and the two kinds of overfitting: where and why do they appear?*. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124002, December 2021.

[Dui00] R.P.W. Duin. Classifiers in almost empty spaces. En *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volumen 2, páginas 1–7 vol.2, 2000.

[FIS14] S.H. Friedberg, A.J. Insel, y L.E. Spence. *Linear Algebra*. Pearson Education, 2014.

[GB10] Xavier Glorot y Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. En *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volumen 9 de *Proceedings of Machine Learning Research*, páginas 249–256. PMLR, 13–15 May 2010.

[GBC16] Ian Goodfellow, Yoshua Bengio, y Aaron Courville. *Deep Learning*. MIT Press, 2016. Book in preparation for MIT Press.

[GBD92] Stuart Geman, Elie Bienenstock, y René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[GSd+19] Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, y Matthieu Wyart. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Phys. Rev. E*, 100:012115, Jul 2019.

[HCB+19] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, y Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism, 2019.

[HT20] Fengxiang He y Dacheng Tao. Recent advances in deep learning theory. *ArXiv*, abs/2012.10931, 2020.

[HTF01] Trevor Hastie, Robert Tibshirani, y Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[HY20] Reinhard Heckel y Fatih Furkan Yilmaz. Early stopping in deep networks: Double descent and how to eliminate it, 2020.

[Kni09] Oliver Knill. *Probability Theory and Stochastic Processes with Applications*. Overseas Press, 2009.

[Kol56] A.N. Kolmogorov. *Foundations Of The Theory Of Probability*. Chelsea Publising Company, 1956.

[KSH12] Alex Krizhevsky, Ilya Sutskever, y Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. En F. Pereira, C.J. Burges, L. Bottou, y K.Q. Weinberger, editores, *Advances in Neural Information Processing Systems*, volumen 25. Curran Associates, Inc., 2012.

[LBBH98] Yann LeCun, L'eon Bottou, Yoshua Bengio, y Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBD+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, y L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

[LBH15] Yann LeCun, Yoshua Bengio, y Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[LT24] Marc Lafon y Alexandre Thomas. Understanding the Double Descent Phenomenon in Deep Learning, March 2024. arXiv:2403.10459.

[MCK20] Yifei Min, Lin Chen, y Amin Karbasi. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization, 2020.

[Mur22] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[Mur23] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[NKB+19] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, y Ilya Sutskever. Deep double descent: Where bigger models and more data hurt, 2019.

[NMB+19] Brady Neal, Sarthak Mittal, Aristide Baratin, Vinayak Tantia, Matthew Scicluna, Simon Lacoste-Julien, y Ioannis Mitliagkas. A modern take on the bias-variance tradeoff in neural networks, 2019.

[Opp95] Manfred Opper. Statistical mechanics of learning: Generalization. En *The Handbook of Brain Theory and Neural Networks*, páginas 922–925. Springer-Verlag, 1995.

[Opp01] Manfred Opper. Learning to generalize. En *Frontiers of Life*, volumen 3, Part 2, páginas 763–775. Academic Press, 2001.

[Poo11] David Poole. *Álgebra lineal. Una introducción moderna*. Cengage Learning, 3 edición, 2011.

[Pri23] Simon J.D. Prince. *Understanding Deep Learning*. MIT Press, 2023.

[Rip96] Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[Sah18] Sumit Saha. A comprehensive guide to convolutional neural networks — the eli5 way. https://medium.com/towards-data-science/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, 2018. [Recurso online, accedido el 19 de febrero de 2025].

[Sch15] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, January 2015.

[Sej20] Terrence J. Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117:30033 – 30038, 2020.

[SFB+22] Gowthami Somepalli, Liam Fowl, Arpit Bansal, Ping Yeh-Chiang, Yehuda Dar, Richard Baraniuk, Micah Goldblum, y Tom Goldstein. Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective, 2022.

[SGd+19] S Spigler, M Geiger, S d'Ascoli, L Sagun, G Biroli, y M Wyart. A jamming transition from under- to over-parametrization affects generalization in deep learning. *Journal of Physics A: Mathematical and Theoretical*, 52(47):474001, October 2019.

[SHN+24] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, y Nathan Srebro. The implicit bias of gradient descent on separable data, 2024.

[SKR+23] Rylan Schaeffer, Mikail Khona, Zachary Robertson, Akhilan Boopathy, Kateryna Pistunova, Jason W. Rocks, Ila Rani Fiete, y Oluwasanmi Koyejo. Double descent demystified: Identifying, interpreting & ablating the sources of a deep learning puzzle, 2023.

[SLHS22] Sidak Pal Singh, Aurelien Lucchi, Thomas Hofmann, y Bernhard Schölkopf. Phenomenology of double descent in finite-width neural networks, 2022.

[SLJ+14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, y Andrew Rabinovich. Going deeper with convolutions, 2014.

[Str23] Gilbert Strang. *Introduction to Linear Algebra*. CUP, 6 edición, 2023.

[Swa20] Swapna. Convolutional Neural Network | Deep Learning. https://developersbreach.com/convolution-neural-network-deep-learning/, 2020. [Recurso online, accedido el 19 de febrero de 2025].

[Vap91] V. Vapnik. Principles of risk minimization for learning theory. En *Proceedings of the 5th International Conference on Neural Information Processing Systems*, NIPS'91, página 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[WH60] Bernard Widrow y Ted Hoff. Adaptive switching circuits. En *1960 IRE WESCON Convention Record, Part 4*, páginas 96–104. Institute of Radio Engineers, 1960.

[YS24] Tian-Le Yang y Joe Suzuki. Dropout drops double descent, 2024.

[ZBH+21] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, y Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, February 2021.