

Xarxes neuronals

Anàlisi comparativa de xarxa neuronal en entorns de programació i full de càlcul

Jiajun Xia
Rui Chen

Tutor: Fernando García Vílchez
Departament de Matemàtiques

Institut Sants

10 d'octubre de 2025

Avís legal

Copyright © Jiajun Xia i Rui Chen. Es garanteix permís per copiar, distribuir i modificar aquest document segons els termes de la GNU Free Documentation License, versió 1.3 o qualsevol posterior publicada per la Free Software Foundation. Es disposa d'una còpia d'aquesta llicència a <http://www.fsf.org> i a l'annex C.

Agraïments

Volem agrair al nostre professor de matemàtiques, **Fernando Garcia**, pel suport constant que ens ha portat durant aquest treball, per la seva claredat en la resolució de dubtes i problemes, i sobretot per haver-se ofert voluntàriament a ser el nostre tutor. La seva ajuda diària, malgrat la gran càrrega de treball amb altres projectes de recerca. Ha estat fonamental per a nosaltres.

També volem expressar el nostre agraïment al professor **Gerard Finol**, de la Universitat Rovira i Virgili, i a **David Martínez**, de la Universitat de Barcelona, que ens han facilitat diversos recursos relacionats amb la intel·ligència artificial, incloent-hi un curs d'iniciació molt valuós.

Donem les gràcies a **Jia Zheng**, un amic que ens ha ofert suport extern en la construcció de la nostra xarxa neuronal.

Agraïm profundament a les persones que contribueixen a Softcatalà per oferir-nos aquest corrector tan sofisticat.

Finalment, volem agrair a **Eric Garcia**, company nostre, per haver-se ofert a revisar i examinar el nostre treball de recerca, així com a totes les persones que ens han donat suport al llarg del procés i han fet possible la realització d'aquest projecte.

Resum

El 1956, un petit grup de científics es va reunir a Dartmouth College amb una idea que semblava de ciència-ficció: crear màquines capaces de pensar. Aquella idea va ser l'inici del qual avui coneixem com a intel·ligència artificial, un camp que impulsa el progrés tecnològic. En unes dècades, la IA ha passat de ser una idea impensable a convertir-se en una eina que ens envolta diàriament, des dels assistents虚拟 fins als sistemes de diagnòstic mèdic o els vehicles autònoms.

Dins aquest univers, destaquen les xarxes neuronals artificials, inspirades en el funcionament del cervell humà. Aquestes estructures matemàtiques i computacionals han demostrat una gran capacitat per aprendre de les dades, reconèixer patrons i resoldre problemes complexos. La seva influència creix de manera exponencial, generant avenços impressionants.

En aquest treball s'explorarà el món de la intel·ligència artificial i, especialment, el de les xarxes neuronals. Se n'explicaran els fonaments i les aplicacions en diferents àmbits, tant tècnics com ètics.

Abstract

In 1956, a small group of scientists gathered at Dartmouth College with an idea that seemed like science fiction: creating machines capable of thinking. That idea marked the beginning of what we now know as artificial intelligence, a field that drives technological progress. In just a few decades, AI has evolved from an unthinkable concept to a tool that surrounds us daily, from virtual assistants to medical diagnostic systems and autonomous vehicles.

Within this universe, artificial neural networks stand out, inspired by the functioning of the human brain. These mathematical and computational structures have demonstrated an extraordinary ability to learn from data, recognize patterns, and solve complex problems. Their influence continues to grow exponentially, leading to impressive advances.

This paper explores the world of artificial intelligence, and especially neural networks, explaining their foundations and applications across different fields, from both technical and ethical perspectives.

Índex de continguts

Capítols:

1	Introducció	1
2	Objectius	3
3	Recerca prèvia	5
4	Metodologia	19
5	Resultats	27
6	Conclusions	47

Apèndixs:

A	Optimització i Ajustos	49
B	De Celsius a Fahrenheit	55
C	GNU Free Documentation License	59
	Bibliografia	78

1. Introducció

Al nostre dia a dia consumim continguts digitals constantment: veiem vídeos curts, treiem fotos, realitzem revisions corporals, juguem videojocs, naveguem per Internet, entre moltes altres activitats. Darrere de totes aquestes activitats hi trobem la presència de la Intel·ligència Artificial (IA), que juga un paper important en àmbits tan diversos com els continguts audiovisuals, el diagnòstic mèdic, la gestió i distribució de la informació o l'optimització de la qualitat d'imatges.

Aquest protagonisme fa que la IA sigui cada vegada més demandada en entorns industrials i professionals. Tot i això, la manca de formació específica i la complexitat del camp dificulen el seu accés.

Partint d'aquesta situació, volem endinsar-nos en el món de la intel·ligència artificial, aprofitant l'oportunitat que ens ofereix el treball de recerca i amb la voluntat de treure'n el màxim profit. La nostra motivació principal és adquirir coneixements que ens ajudin a preparar-nos per als reptes del futur i que siguin útils tant acadèmicament com professionalment.

En aquest treball ens centrarem en les xarxes neuronals, que són essencials en la intel·ligència artificial, ja que permeten processar grans quantitats de dades i generar resultats com ara prediccions i la identificació de patrons. Analitzarem com funcionen els algoritmes d'entrenament i algunes aplicacions pràctiques, amb la finalitat de determinar si un full de càcul té la capacitat de competir amb una xarxa neuronal construïda amb un llenguatge de programació de precisió.

1.1 Motivació

Ara que ja hem presentat la temàtica del nostre Treball de Recerca (TR), expliquem quines són les motivacions que ens empenyen a dur-lo a terme:

- La principal raó que ens va portar a aquest tema és la nostra afició per la informàtica. En un futur ens agradaria aprofundir-hi i continuar treballant-hi en un grau, màster o doctorat. Per això aprofitem l'oportunitat del TR per començar a preparar-nos.

- Una altra raó que ens va animar a triar aquest tema són els recursos que tenim a la nostra disposició. D'una banda, el nostre tutor va contactar amb el Gerard Finol [38], professor de la Universitat Rovira i Virgili, i el David Martínez [8], professor de la Universitat de Barcelona. Ells ens van donar una llista de recursos per començar a investigar en aquest camp. A més, un amic nostre ens va oferir recursos externs per elaborar la part pràctica de la xarxa neuronal amb el full de càcul.
- Conjuntament amb el tutor vam decidir que el TR havia de ser una oportunitat per aprendre a fer servir recursos útils per als nostres estudis universitaris: un entorn de treball col·laboratiu, dominar llenguatges de programació i utilitzar un editor de text apropiat per a textos científics.
- La voluntat de construir una eina amb mirada cap al futur, aplicant els coneixements adquirits i amb la determinació de fer un treball rigorós i ambicions.

1.2 Estructura de la memòria

Després d'aquesta breu introducció, al capítol 2: Objectius, es presenten els objectius del nostre TR. A continuació, al capítol 3: Recerca prèvia, es presenten els coneixements previs necessaris per a comprendre el funcionament de les xarxes neuronals. Abans de començar la part pràctica, vam iniciar una nova recerca per decidir quines eines utilitzaríem i de quina manera, aquesta part es mostra en el capítol 4: Metodologia. Un cop hem presentat com treballarem, toca presentar els resultats obtinguts al 5: Resultats. La memòria es tanca amb el capítol 6: Conclusions on exposem les conclusions i les línies de recerca futura que són fruit del nostre TR. Volem fer notar que algunes parts de la recerca que s'han hagut de resumir, a causa de les limitacions del format del TR, s'expliquen de manera detallada en els apèndixs.

2. Objectius

Tal com s'ha explicat a la introducció, l'objectiu principal del nostre **Treball de Recerca (TR)** és construir i analitzar el funcionament d'una xarxa neuronal. No ens conformem amb qualsevol implementació, ja que n'existeixen molts models i variants. Per donar més coherència i solidesa a l'estudi, hem decidit treballar amb **dos models diferents**:

- Una xarxa neuronal implementada amb llenguatge de programació (Python),
- Una xarxa neuronal reproduïda en un full de càcul.

A continuació detallarem els requisits específics de la nostra recerca, dividits segons qui els proposa.

1. Requisits proposats pel tutor:

(a) **Entorn de treball professional**

El tutor ens ha demanat reproduir la manera de treballar dels centres d'investigació en aquest camp. En coordinació amb dos professors universitaris, ha determinat quines eines eren més adequades per portar endavant aquest projecte: sistema operatiu Linux, entorn col·laboratiu Git, editor LaTeX i altres aspectes. Tots els elements relacionats amb la metodologia es detallen al capítol 4: Metodologia.

(b) **Escalabilitat**

La xarxa neuronal ha de poder ampliar-se fàcilment, ja sigui afegint més dades, modificant paràmetres o incrementant-ne la complexitat, sense necessitat de redissenyar-la completament.

(c) **Llibertat**

El nostre projecte és programari lliure [33], accessible, modifiable i compatible per a tothom, complint l'exigència del tutor. Hem fet realitat aquest compromís gràcies a GitHub [43], que ens ha proporcionat les eines per treballar de manera oberta, col·laborativa i segura. Així, el nostre treball esdevé un recurs que qualsevol pot utilitzar, adaptar i millorar, demostrant els beneficis i la viabilitat de la llibertat digital.

(d) Simplicitat i eficàcia

La xarxa neuronal ha de ser tan simple com sigui possible, però alhora amb la màxima eficàcia.

(e) Traçabilitat

El nostre projecte ha estat creat de manera transparent. Qualsevol persona pot descarregar el repositori i reproduir la nostra recerca, així com visualitzar tot el procés de creació del TR.

2. Requisits proposats per nosaltres:

(a) Xarxa neuronal pròpia

No ens hem conformat a fer només la feina que ens va demanar el tutor. Un cop vam assolir els objectius inicials, vam decidir ampliar la recerca. Vam escollir una xarxa neuronal específica, concretament de regressió. Els valors introduïts havien de ser fruit de la nostra pròpia recerca, per exemple mitjançant un formulari. El valor a predir era la nostra nota final de matemàtiques, i la xarxa havia d'assolir almenys un 65% de precisió.

(b) Difusió i didàctica

A més de la implementació en llenguatge de programació, hem desenvolupat una variant de la xarxa de regressió utilitzant un full de càlcul, amb l'objectiu de mostrar de manera visual i entenedora el seu funcionament.

3. Recerca prèvia

3.1 La Història de la IA: Des dels orígens fins avui

1. El Naixement d'una Idea Revolucionària (1956)

L'Alan Turing, el geni matemàtic que va desxifrar Enigma, una màquina emprada pels nazis per codificar els seus missatges durant la Segona Guerra Mundial (1939-1945), va fer una pregunta provocadora a la comunitat científica: "Podran les màquines pensar alguna vegada?". Aquesta qüestió va obrir les portes a un nou camp d'estudi. En 1956 John McCarthy, Marvin Minsky i d'altres especialistes van nominar oficialment el terme "inteligència artificial" durant la conferència de Dartmouth, marcant l'inici d'una nova era tecnològica.

2. El Joc que ho va canviar tot: The Imitation Game/El test de Turing

L'origen de la IA es basa en un experiment molt senzill, però profund: El joc d'imitació (The imitation game), proposat per Alan Turing. Per respondre a la pregunta "Podran les màquines pensar alguna vegada?", Turing va dissenyar un joc que funcionava com a test per les màquines anomenat "The Imitation Game". Aquest test consistia a fer que un avaluador havia d'intercanviar textos escrits amb una persona i una màquina durant 5 minuts, aquest avaluador no sabia qui era qui i havia d'esbrinar qui era l'humà. Si la màquina aconseguia enganyar a l'avaluador passava el test i es considerava que la màquina havia aconseguit un nivell de comportament lingüístic equivalent al d'un humà, per tant, podríem considerar que les màquines poden pensar. Aquest joc ha estat evolucionant gràcies als avenços de la ciència i de la tècnica, i encara avui dia és conegut com el test de Turing.

3. Les grans fites de la IA

1997: La màquina que va vèncer un campió

La supercomputadora Deep Blue desenvolupada per IBM va derrotar el campió mundial d'escacs, Garri Kaspàrov, demostrant que la IA podia superar els humans en jocs d'estratègia complexos.

2022: L'explosió de la IA

Milions d'usuaris van descobrir models com ChatGPT que podien escriure, traduir i programar amb un llenguatge gairebé humà, obrint nous horitzons en la interacció home-màquina.

2025: La IA en tots els àmbits

Avui, la IA està present en dibuix, contingut audiovisual, cotxes autònoms, medicina i molt més, amb models cada vegada més especialitzats i avançats.

Fonts: [54], [83], [62] i [72]

3.2 Què és la IA?

Podem definir la IA com sistemes de software i de hardware dissenyats per humans que actuen en la dimensió física o digital, és a dir, raonar sobre el coneixement, processant la informació derivada de dades i prendre les millors decisions per assolir l'objectiu donat. Explicat amb unes altres paraules: és un camp de la informàtica que consisteix en un conjunt de capacitats intel·lectuals i cognitives expressades per un sistema informàtic creat pels humans, que té com a propòsit imitar la intel·ligència humana.

Un exemple de IA que tot el món coneix i utilitza és el ChatGPT, un xatbot impulsat per un model d'intel·ligència artificial generativa de l'empresa OpenAI. Fa servir tècniques de processament de llenguatge natural per comprendre preguntes fetes per l'usuari i generar respostes coherents en converses, simulant una interacció similar a la d'un humà. També pot generar o editar imatges, processar àudios, llegir arxius i molt més.

Font: [?]

3.3 Com funciona la IA?

Una vegada que ja sabem que és una IA, ens toca entendre com funciona. Les intel·ligències artificials utilitzen algoritmes i models matemàtics per processar grans quantitats de dades i prendre accions basades en patrons i regles establertes a través de l'aprenentatge automàtic o de l'aprenentatge profund. Per tant, per funcionar necessitarà:

1. Dades

Les dades són fonamentals per la IA, ja que és la base de l'aprenentatge del model, per poder raonar, prendre decisions i millorar la precisió. Aquí es mostren alguns dels aspectes que hem de tenir en compte:

- **Base d'aprenentatge**

Els algoritmes de la IA necessiten una base de dades i una gran diversitat de dades per poder identificar patrons i construir prediccions.

- **Qualitat vs Quantitat**

Una gran quantitat de dades ajudaran la IA a obtenir major precisió, però la qualitat és encara més important per la complexitat de dades que aporta, això evitara que la IA cometi errors per informació incompleta. Per exemple, en l'àmbit mèdic si vols que la IA faci una predicción i tan sols li dones una quantitat important de pacients sans i no d'altre tipus de pacients, la IA simplement contemplarà la predicción de persona sana i descartarà la resta de possibilitats.

- **Exemples reals**

Els sistemes dels cotxes o assistents virtuals necessiten dades en temps reals per adaptar-se de l'entorn i poder reaccionar de manera correcta. Les plataformes com Netflix o Spotify necessiten dades personalitzades reals i actualitzades de cada usuari per poder generar recomanacions amb precisió.

2. Algoritmes

Totes les aplicacions de la IA tenen el propòsit de capacitar la màquina perquè pugui operar com un humà. Tot això està basat en combinacions de diferents tipus d'algoritmes. A continuació, explicarem els diferents tipus d'algoritmes d'aprenentatge.

- **L'aprenentatge automàtic (Machine learning):** És una branca crucial de la intel·ligència artificial, consisteix a dotar de vida a la màquina, donar-li el poder d'aprendre com els humans, executar tasques de manera autònoma i finalment, les infinites possibilitats d'evolucionar a través de l'experiència i molt més. Segons la UC Berkeley [76] el procés de l'aprenentatge automàtic consta de:

- (a) **Mecanisme de predicción:** *Un conjunt de regles o operacions matemàtiques que analitza les dades d'entrada i identifica els patrons que busca el model.*

- (b) **Algoritme d'optimització:** El procés que ajusta automàticament el model per minimitzar l'error, modificant els paràmetres interns per millorar les prediccions futures.

Segons Nvidia [61], hi ha molts tipus d'aprenentatge automàtics:

- (a) **Aprenentatge supervisat:** L'aprenentatge supervisat és un tipus d'aprenentatge automàtic que treballa amb dades etiquetades, és a dir, dades que ja inclouen la solució o resultat desitjat. En aquest mètode, la intel·ligència artificial aprèn a associar les dades d'entrada amb les seves etiquetes corresponents, mitjançant l'anàlisi d'exemples prèviament resolts. Això li permet desenvolupar la capacitat de resoldre problemes nous aplicant la lògica i els patrons identificats a partir de dades reals.
- (b) **Aprenentatge no supervisat:** L'aprenentatge no supervisat és una branca de l'aprenentatge automàtic que s'utilitza quan no es disposa de dades etiquetades. A diferència de l'aprenentatge supervisat, on el model rep exemples amb les seves solucions correctes, en aquest cas l'algoritme ha de descobrir per si mateix l'estructura i els patrons, fent un diagnòstic agrupant les característiques similars que poden haver-hi entre les dades. Depenen dels tipus de problemes, les dades s'organitzen de diferents maneres.
- **Clustering:** Tècnica que agrupa les dades en funció de les seves similituds.
 - **Anomaly detection:** Cerca patrons que no encaixen amb el comportament normal.
 - **Association:** Cerca relacions i correlacions entre variables en grans conjunts de dades.
 - **Autocodificador:** Els autocodificadors són un tipus de xarxa neuronal artificial que aprèn a comprimir i reconstruir dades.
- (c) **Aprenentatge semisupervisat:** L'aprenentatge semisupervisat representa un punt intermedi entre l'aprenentatge supervisat i el no supervisat, aprofitant tant dades etiquetades com no etiquetades per millorar l'eficiència dels models d'aprenentatge automàtic. Això funciona quan l'obtenció de les dades etiquetades són molt costoses i l'extracció de les característiques són molt complexes.

- (d) **Aprenentatge per reforç:** L'aprenentatge per reforç és una altra branca de l'aprenentatge automàtic inspirada en la manera com els éssers vius aprenen mitjançant la interacció amb el seu entorn.
- (e) **Aprenentatge profund:** L'aprenentatge profund és una branca de l'aprenentatge automàtic que utilitza una xarxa neuronal amb múltiples capes per processar dades complexes i extreure'n característiques rellevants. Aquesta és inspirat en el funcionament del cervell humà, aquest enfocament permet identificar patrons i analitzar dades d'alta complexitat. Durant la fase d'identificació, s'empra un aprenentatge jeràrquic, és a dir, progressa gradualment des de característiques simples fins a patrons complexes.

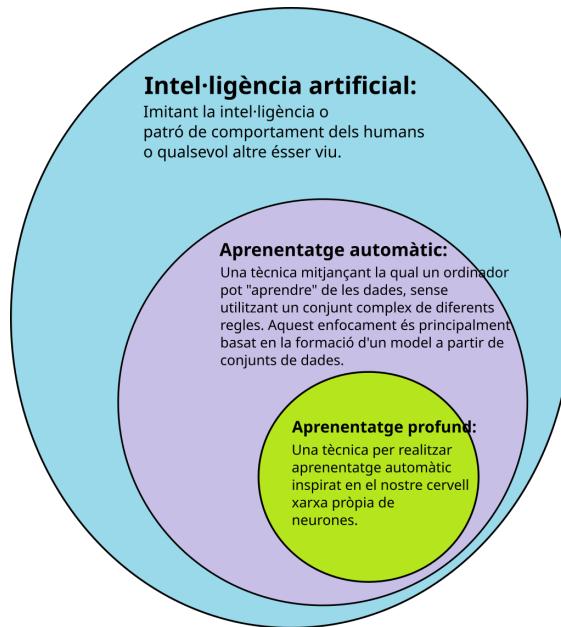


Figura 3.1: Relació entre intel·ligència artificial, aprenentatge automàtic i aprenentatge profund[3]

3. Frameworks

Segons INESDI [23]:

“Un Frameworks és, en el camp de la informàtica, una estructura conceptual que proporciona un conjunt d'eines, biblioteques i patrons de disseny per facilitar el desenvolupament de programari.

Pel que fa als seus components, inclouen biblioteques de codi reutilitzables, mòduls predefinits, regles d'arxiu i directori, patrons de disseny i convencions de codificació.”

4. Ètica i Regulació

Un cop enteses totes les funcionalitats i capacitats de la intel·ligència artificial (IA), és fonamental aplicar-hi principis d'ètica i moral. Encara que la IA no sigui un ésser viu, és una eina creada pels humans, per tant, cal establir limitacions per evitar-ne els mals usos i garantir el respecte als drets humans i a la privacitat. Aquesta part és crucial en el desenvolupament d'una IA responsable. En aquest sentit, molts estats i organismes ja han publicat diverses lleis legítimes i regulacions. Algunes de les més rellevants són:

- **Llei d'IA de la Unió Europea (AI Act):** És la primera regulació integral sobre la IA. Classifica els sistemes d'IA segons el seu nivell de risc (inacceptable, alt, limitat i mínim) i estableix requisits estrictes per als usos d'alt risc, com la transparència, la supervisió humana i la seguretat.
- **Reglament General de Protecció de Dades (GDPR):** Tot i no ser exclusiu per a la IA, aquest reglament europeu protegeix la privacitat i les dades personals dels ciutadans. És clau en el desenvolupament d'una IA que utilitzi dades personals.
- **Principis ètics de la UNESCO sobre la IA (2021):** Proposen una base global per al desenvolupament ètic de la IA, centrant-se en el respecte pels drets humans, la igualtat, la sostenibilitat, la no discriminació i la supervisió humana.
- **Guies de l'OCDE sobre la IA:** Recomanen que els sistemes de IA siguin transparents, responsables, segurs i que promoguin el benestar de la societat, ajudant a establir marcs internacionals de bones pràctiques.

Fonts: [57], [16], [40], [74], [6], [12], [39] i [75].

3.4 Què és una xarxa neuronal artificial/biològica?

Una xarxa neuronal artificial és un model computacional inspirat en el funcionament del cervell humà, utilitzat en el camp de la IA i de l'aprenentatge automàtic (machine learning). Està dissenyada per a reconèixer patrons, prendre decisions i aprendre a partir de dades, sense ser programada explícitament per a cada tasca de manera específica.

Si tenim una artificial també tindrem una biològica. Una xarxa neuronal biològica es refereix al sistema interconnectat de neurones (cèl·lules nervioses) en el cervell i el sistema nerviós

dels éssers vius. Aquestes xarxes són la base de la cognició, l'aprenentatge i les funcions biològiques en humans i animals.

Fonts: [37] i [26]

3.5 Estructura d'una xarxa neuronal

Una xarxa neuronal combina diverses capes de processament i utilitza elements simples que operen en paral·lel, simulen i estan inspirades en els sistemes nerviosos biològics com hem explicat en l'apartat 3.4. Consta d'una capa d'entrada, seguit d'una o diverses capes ocultes i finalment una capa de sortida. Les capes estan interconnectades mitjançant nodes o neurones; cada capa utilitza la sortida de la capa anterior com a entrada.

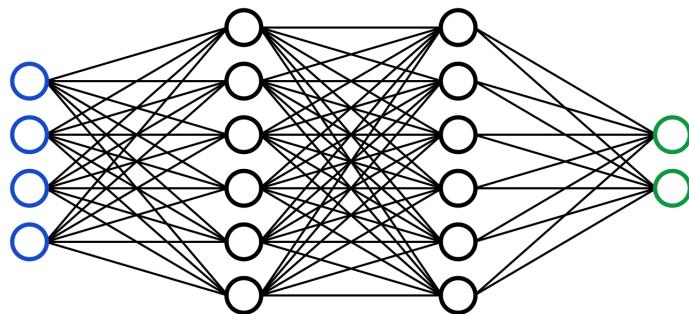


Figura 3.2: Estructura d'una xarxa neuronal [1]

- **Capa d'entrada:** La capa d'entrada és la primera capa que rep directament la informació que es processarà.
- **Capes ocultes:** Les capes ocultes són les capes que estan entre la capa d'entrada i la de sortida, aquestes capes contenen unitats no observables. La seva funció principal és processar les dades de la capa d'entrada per extraure característiques i patrons complexos. La quantitat de neurones que hi ha en les capes ocultes és un factor determinant per la capacitat que tingui la xarxa per capturar dades complexes.
- **Capa de sortida:** La capa de sortida és l'última capa que forma una xarxa neuronal i és l'encarregada de produir la predicció o el resultat final del model. Aquesta capa utilitza la informació que ha processat la o les capes ocultes i la transforma a través d'una funció activa per generar una sortida, que pot ser una predicció numèrica, una

classificació o qualsevol altre resultat. Les neurones d'aquesta capa estan connectades amb totes les neurones de la capa anterior.

Fonts: [21] i [67]

3.6 Exemples de xarxes neuronals

Hi ha molts tipus de xarxes neuronals, però com que el treball té una limitació de pàgines farem un resum de les xarxes més rellevants.

1. Perceptron (1958)

La primera xarxa neuronal, el Perceptró, va ser creada en la dècada de 1950 a 1960 pel psicòleg i informàtic Frank Rosenblatt. Aquesta màquina permet prendre decisions binàries, per exemple respondre sí o no, de manera autònoma.

2. Multilayer Perceptron

El multilayer perceptron és una ampliació de la percepció d'una única neurona a més d'una. A més, apareix el concepte de capes d'entrades, capes ocultes i capes de sortida, però amb valors d'entrada i sortida binàries.

3. Neurones sigmoide

Per aconseguir que les xarxes neuronals aprenguin per elles mateixes, és a dir, aprenentatge automàtic, va ser necessari introduir un nou tipus de neurones: Neurones Sigmoides, que són similars al perceptró. Aquestes neurones en comptes de què les entrades siguin binàries (0 o 1), poden treballar amb qualsevol altre valor real, per exemple, 0.5 o 0.374.

4. Xarxa neuronal prealimentada (Feedforward)

Les xarxes neuronals prealimentades són les que les sortides d'una sola capa són utilitzades com entrades en la pròxima.

Fonts: [36], [7], [50] i [28]

3.7 Funció d'activació

Les funcions d'activació són un component integral de les xarxes neuronals que els permeten aprendre patrons complexos en les dades. Transformen el senyal d'entrada d'una neurona en un senyal de sortida que passa a la capa següent. Sense funcions d'activació, les xarxes neuronals es limitarien a modelar únicament relacions lineals entre entrades i sortides, és a dir, introduceixen la no-linealitat i produueixen la sortida de la neurona.

1. Funció sigmoide

Una funció d'activació molt coneguda és la funció sigmoide. La seva fórmula és:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Aquesta funció matemàtica transforma qualsevol valor d'entrada real en un valor que està entre 0 i 1. La seva forma característica és una corba en forma de “S”. Si el valor de x que introduïm a la funció és molt gran o proper a infinit (∞), llavors σ serà 1; en canvi, si és molt petit o proper a menys infinit ($-\infty$), σ serà 0, i si $x = 0$, σ serà 0,5.

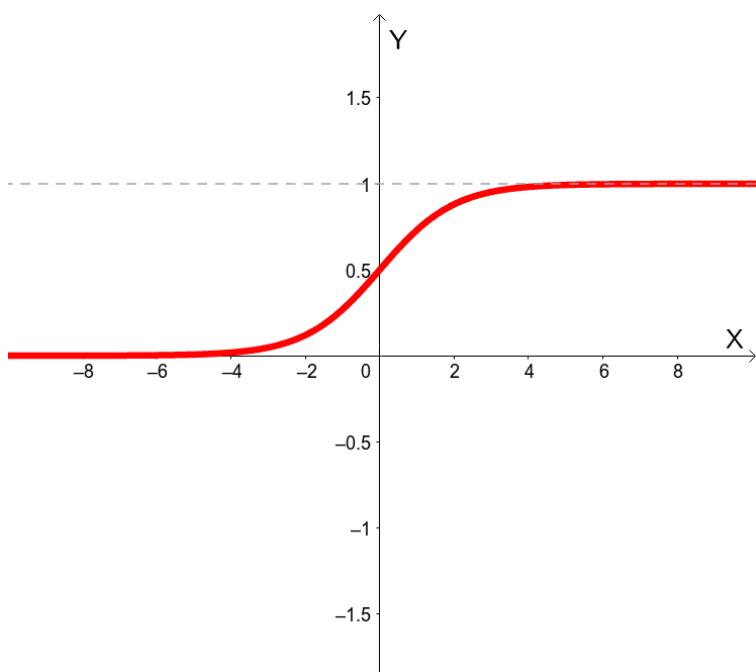


Figura 3.3: Gràfica de la funció sigmoide. [71]

2. Funció ReLU (Funció Unitat Rectificada Uniforme)

La funció Unitat Rectificada Uniforme és:

$$f(x) = \max(0, x)$$

Aquesta funció té aplica el següent algoritme:

- si el valor d'entrada és menor que 0, la sortida és 0.
- si el valor d'entrada és major o igual que 0, mostrerà el valor d'entrada.

Aquest algoritme és lineal si l'entrada és més gran que 0 perquè el pendent és 1. Encara que la funció ReLU és lineal per a la meitat del seu espai d'entrada, tècnicament és una funció no lineal perquè té un punt no derivable en $x = 0$, on canvia bruscament respecte a x . Aquesta no-linealitat permet a les xarxes neuronals aprendre patrons complexos.

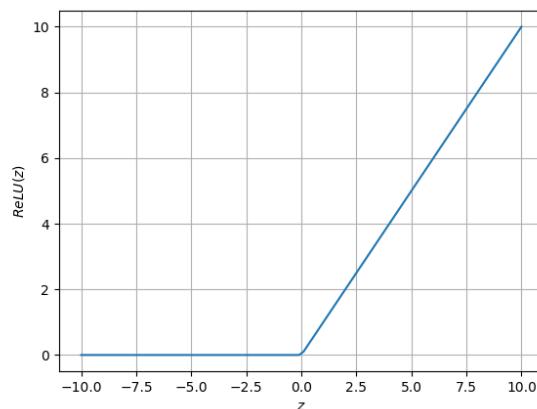


Figura 3.4: Gràfica de la funció ReLU. [11]

3. Funció Softmax

La funció Softmax és una de les funcions que més s'utilitzen en xarxes neuronals i és especialment útil en el context dels problemes de classificació multiclasse. Aquesta funció opera sobre un vector que representa les previsions de cada classe, calculades per les capes anteriors.

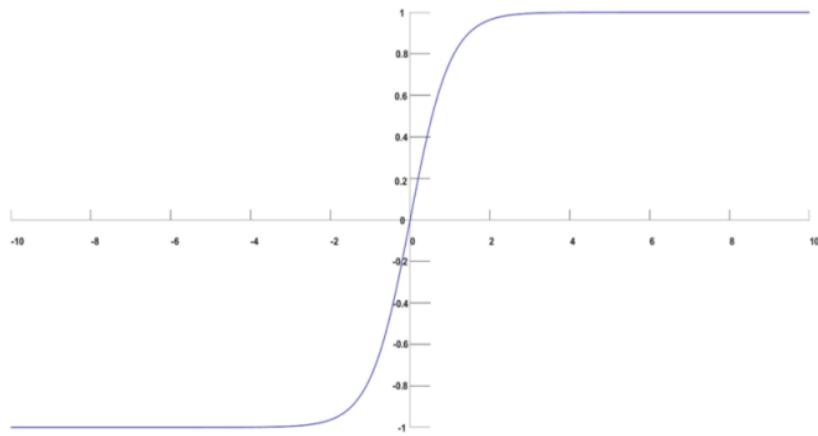


Figura 3.5: Gràfica de la funció Softmax.[52]

Per a un vector d'entrada $x = (x_1, x_2, \dots, x_C)$, la funció Softmax es defineix:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

El resultat de la funció Softmax és una distribució de probabilitat. Cada element del resultat representa la probabilitat que l'entrada pertanyi a una classe determinada. L'ús d'aquesta funció garanteix que tots els valors de la sortida siguin positius i que la seva suma és 1.

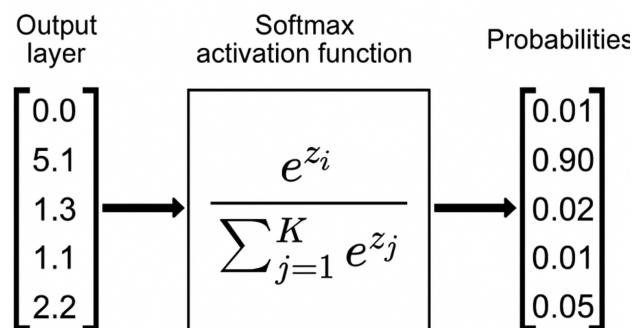


Figura 3.6: Representació de la funció Softmax. [63]

Fonts: [21], [46], [11], [73], [17]

3.8 Com funciona una xarxa neuronal?

Ara coneixem l'estructura d'una xarxa neuronal artificial, podem entendre com funciona. Les neurones o nodes són els pilars més importants d'una xarxa neuronal. Cada neurona utilitza l'entrada, processa aquesta informació fent operacions matemàtiques. A continuació aplica una funció d'activació, i finalment envia el resultat com a sortida per a que sigui la informació d'entrada d'altres neurones.

Les connexions (pesos i biaixos) són la força de connexió entre dues neurones representades per un pes.

Els pesos: Són valors que determinen quanta influència té la producció d'una neurona sobre un altre, marca la importància que té cada neurona.

Els biaixos: Són paràmetres addicionals que ajuden a ajustar els valors de les neurones.

El llindar: Si la sortida de qualsevol node individual és més gran que el valor del llindar, aquell node s'activarà i envia dades a la següent capa de la xarxa. En cas contrari, no passarà cap dada a la següent capa de la xarxa.

Cal entendre que cada node individual, com el seu propi model de regressió lineal, està compost per dades d'entrada, ponderacions, un llindar o un biaix i una sortida. La fórmula per a calcular els valors d'una xarxa neuronal és la següent:

$$\sum w_i x_i + \text{biaix} = w_1 x_1 + w_2 x_2 + w_3 x_3 + \text{biaix}$$

on w_i són els pesos i x_i són les entrades

Font: [42]

3.8.1 Propagació cap a davant

Durant la propagació cap a davant, les dades ingressen en la xarxa a través de la capa d'entrada i flueixen seqüencialment a través de les capes ocultes fins a la capa de sortida. En cada neurona, els valors d'entrada del model es multipliquen pels seus pesos corresponents i se sumen. Aquesta suma ponderada passarà a través d'una funció d'activació, explicada prèviament a l'apartat 3.7. Aquest procés continua capa per capa, això acaba conduint cap a la predicció final en la capa de sortida.

Aquest procés és important per les següents raons:

- **Base per l'aprenentatge:** No es pot comprendre com aprenen les xarxes neuronals sense primer entendre com fan prediccions. La propagació cap a davant és el requisit previ que s'ha de conèixer per comprendre la retropropagació, l'algoritme que permet l'aprenentatge.
- **Optimització:** En el cas que una xarxa neuronal no funcioni bé, saber com flueixen les dades per la xarxa t'ajudarà a identificar i solucionar problemes.
- **Disseny del model:** Un disseny eficaç de la xarxa requereix comprendre com es distribueix la informació a través de les configuracions de capes.

Font: [2]

3.9 Retropropagació en les xarxes neuronals

Mentre que la programació directa fa prediccions, la retropropagació és la forma en què la xarxa aprèn dels errors. Això implica comparar la predicció de la xarxa amb el valor objectiu real i calcular un terme d'error mitjançant una funció d'error.

Aquest error es propaga enrere a través de la xarxa, començant des de la capa de sortida. Durant aquest procés, la xarxa ajusta els pesos i els biaixos de cada connexió en funció de la seva contribució a l'error, amb l'objectiu de minimitzar-lo.

Aquest procés iteratiu de càlcul d'erros i ajustament de pes permet a la xarxa d'aprenentatge profund millorar gradualment les seves prediccions.

El procés de retropropagació es basa en el principi d'optimització del gradient descendent, explicat amb més detall a l'apèndix A: Optimització i Ajustos, on es calculen els gradients d'error respecte als paràmetres de la xarxa en sentit contrari i en cada capa. Per aquesta tasca, s'utilitza l'algoritme de la cadena per propagar l'error cap endarrere des de la sortida fins a l'entrada.

3.9.1 Com funciona la retropropagació?

El funcionament de la retropropagació el podem dividir en 5 fases. Cada una d'aquestes possibilita a la xarxa neuronal aprendre de manera eficient a partir de dades proporcionades.

Les fases són les següents:

- **Propagació cap endavant:** Aquesta fase inicial és on s'introdueixen les dades de prova en la xarxa neuronal des de la capa d'entrada fins a la sortida.
- **Càcul d'error:** Una vegada s'obté la sortida de la xarxa, es compara amb el valor desitjat mitjançant una funció d'error. Aquesta funció quantifica la discrepància que s'ha produït entre la predicció i el valor real.
- **Retropropagació de l'error:** En aquesta fase els gradients de l'error es calculen respecte a cada paràmetre en la xarxa. Com hem dit abans, aquest procés comença de manera inversa a la propagació, des de la sortida fins a l'entrada.
- **Actualització dels paràmetres:** Una vegada que s'ha calculat els gradients de l'error en tots els paràmetres, aquests s'actualitzen. D'aquesta manera, els errors en la predicció es minimitzen de manera gradual en cada iteració de l'entrenament.
- **Configuració de la predicció:** Després de totes les optimitzacions, el mètode de càlcul torna a revisar les entrades de prova. Després de tot això, es busca garantir els resultats esperats.

Fonts: [44] i [22]

4. Metodologia

Després d'haver fet la recerca prèvia i d'haver decidit fer una xarxa neuronal de regressió que compleixi tots els requisits que esperàvem, ara toca escollir quina és la millor manera d'aconseguir el nostre objectiu, és a dir, definir la nostra metodologia. Per tant, també hem de fer recerca sobre quines eines utilitzarem per dur a terme el TR de la millor manera.

Primer, ens hem plantejat la qüestió de sobre quins àmbits hem de recórrer. Buscarem les millors eines i mètodes que ens ajudin en cadascun dels àmbits. A continuació, presentarem els àmbits que hem tractat i les diferents solucions que hem considerat.

Hem estructurat la recerca en els següents apartats:

1. Entorn col·laboratiu.
2. Comunicació.
3. Sistema Operatiu.
4. Llenguatge de programació.
5. Editor de text.

4.1 Entorn col·laboratiu

A l'hora de crear un projecte en equip, pot ser complicat treballar amb dos o més membres alhora en els seus respectius dispositius si no s'escull una eina col·laborativa adequada, especialment quan es treballa amb molts fitxers. Poden sorgir problemes en l'intercanvi d'arxius, o que només un membre pugui editar un fitxer a la vegada, cosa que pot convertir el procés en un autèntic caos.

1. **Google Drive:** és una plataforma de Google que ofereix eines gratuïtes com un processador de text (Docs) i un entorn col·laboratiu. Tot i que és senzill i eficient, presenta diverses limitacions importants: és difícil gestionar canvis simultanis o resoldre conflictes entre versions, no permet un control detallat de les modificacions fetes i el registre de canvis és molt bàsic. A més, Google té accés als arxius, cosa que pot ser un problema per a dades sensibles. Aquestes limitacions fan que Google Drive no sigui

l'opció ideal per al nostre TR, fet que ens va portar a escollir l'altra alternativa que teníem: Git + GitHub + Vim.

2. Git + GitHub + Vim

Git és un sistema de control de versions [5] que permet portar un historial complet dels canvis realitzats en un projecte, crear branques per desenvolupar noves funcionalitats sense afectar la versió principal i oferir un control total de les modificacions. També facilita la revisió de codi i la col·laboració entre membres de l'equip.

Tot i així, Git per si sol no està del tot optimitzat per a equips grans, i aquí és on entra GitHub. Aquesta plataforma complementa Git amb repositoris remots, la possibilitat d'utilitzar les comandes git pull, commit i push per compartir canvis i revisar-los abans d'integrar-los a la versió principal, i eines per resoldre conflictes de manera col·laborativa. L'editor *Vim* [77] també resulta útil en aquest entorn per gestionar conflictes de fusió directament des de la línia d'ordres.

Per a més informació, es pot consultar la documentació oficial de Git [31].

Fonts:[20] i [68]

4.2 Comunicació

Atès que som dues persones realitzant el treball amb un tutor, ens resulta fonamental disposar d'un bon canal de comunicació. Per això, hem adoptat diferents mitjans de comunicació. A continuació, expliquem les diferents formes de comunicació utilitzades en l'elaboració d'aquest treball.

1. **Correu electrònic:** El principal mitjà de comunicació que hem utilitzat ha estat el correu institucional del centre (Gmail [32]). Aquest servei ens ha permès intercanviar informació de manera ràpida i eficient, així com emmagatzemar els missatges per a futures consultes. L'hem fet servir sobretot per a la resolució de dubtes senzills.
2. **Git:** Git [13] és un sistema de control de versions que facilita el treball col·laboratiu, especialment en projectes de programes. En el nostre treball de recerca hem creat un repositori(TR-RJ-2025 [9]) a GitHub [43], que ens ha ofert un entorn adequat per al desenvolupament i la gestió del projecte.

3. **Fulls de càlcul:** Per organitzar les reunions amb el tutor i planificar el nostre treball, hem utilitzat fulls de càlcul compartits [?]. A través d'aquestes eines hem pogut establir el calendari de trobades, definir les metodologies de treball i rebre suport en la instal·lació del programari necessari.

4.3 Sistema Operatiu

Els programes i programaris d'alt nivell moltes vegades no són compatibles amb tots tipus de sistemes operatius, o no tenen el mateix rendiment en un respecte a l'altre. Un sistema operatiu és:

“Para entender qué es un sistema operativo, podemos pensar en él como en un administrador central que gestiona todos los recursos del sistema. Actúa como intermediario entre el hardware de un dispositivo informático (como una computadora, un teléfono inteligente o una tableta) y las aplicaciones de software que se ejecutan en él. Controla las operaciones del hardware y facilita la ejecución de múltiples tareas y procesos a través de una interfaz sencilla. “

Segons la Universitat Europea [27], els sistemes operatius es classifiquen segons l'empresa que els desenvolupa, i els més destacats són els següents: **Microsoft Windows** i **Linux**

4.3.1 Windows

Windows és un sistema operatiu desenvolupat per Microsoft a principis dels anys 80. La seva popularitat va ser tan gran que, en el seu moment més àlgid, va arribar a tenir fins a un 90% de quota de mercat. Entre les seves característiques principals, destaca la interfície gràfica basada en finestres, que facilita l'ús i redueix la necessitat de comandes de text; les aplicacions integrades amb funcionalitats com la barra de tasques i el menú d'inici; i la gran disponibilitat de programari, amb una instal·lació d'aplicacions senzilla i accessible.

4.3.2 Linux

Linux és un sistema operatiu de codi obert creat per Linus Torvalds el 1991 i és conegut per la seva gran versatilitat. Entre les seves característiques principals hi ha la llibertat i la possibilitat de personalitzar-lo completament gràcies al seu model de codi obert, l'eleva-

da seguretat i estabilitat, i la disponibilitat de molts recursos que el fan ideal tant per a ordinadors antics com per a tasques especialitzades.

4.3.3 macOS

Va quedar descartar ja que està limitat a aparells de la marca Apple i nosaltres no en disposem de cap aparell d'aquest tipus.

Decisió Final: Windows i Linux

Hem optat per utilitzar **Windows** i **Linux** pels següents motius. Linux és ideal per a tasques avançades, terminal i automatitzacions, a més de ser compatible amb l'editor Kile. Windows, en canvi, ofereix una major facilitat d'ús i compatibilitat amb programari general, sent perfecte per a tasques diàries.

No obstant això, en un ordinador és poc habitual tenir dos sistemes operatius funcionant alhora, cosa que ens va portar a fer una altra recerca. Vam identificar dues solucions possibles: la primera és el **dual boot** [14], que consisteix a instal·lar els dos sistemes per separat i triar-ne un a l'arrencada. La segona és la **màquina virtual** [?], que permet executar un sistema operatiu dins de l'altre.

Finalment, vam optar per la màquina virtual per la seva facilitat d'ús i baixos requisits, tot i que el rendiment no és tan elevat com en un dual boot. Aquesta opció ens permet provar Linux sense afectar la instal·lació principal de Windows.

4.4 Llenguatge de programació

El llenguatge de programació [30] triat per a la part pràctica del treball va ser **Python** [65]. Creat el 1989 per Guido van Rossum, informàtic neerlandès, Python és conegut per la seva sintaxi senzilla i llegible. Va ser escollit pel tutor i els professors universitaris.

4.4.1 Python

La raó principal per la qual vam escollir aquest llenguatge són els seus avantatges. Python no necessita compilació: un cop escrit el codi, es pot executar directament. Disposa d'una gran quantitat de llibreries útils, com TensorFlow o NumPy, que permeten estalviar moltes línies de codi, especialment en projectes com xarxes neuronals. A més, la identificació d'errors és

clara i precisa, cosa que facilita la resolució de problemes sense haver de revisar el codi línia per línia. Finalment, la seva sintaxi clara i llegible, el converteix en un llenguatge ideal per a principiants.

4.4.2 Sintaxi

Tot i que ja hem explicat que la sintaxi de Python és simple, val la pena destacar què la diferencia d'altres llenguatges. La principal característica és que el seu format de codi és visualment ordenat i utilitza paraules clau senzilles en anglès. A diferència d'altres llenguatges com C++ o Java, no utilitza claudàtors per marcar blocs de codi; en el seu lloc, fa servir espais i tabulacions per organitzar les sentències, la qual cosa obliga a mantenir el codi net i llegible.

4.4.3 Editor de codi

Per començar a programar és molt important escollir un bon editor de codi. Per aquest motiu, vam analitzar les opcions disponibles i vam decidir quina s'adaptava millor a les necessitats del nostre projecte.

VS Code(Visual Studio Code): VS Code és un editor de codi gratuït i de codi obert desenvolupat per Microsoft, i continua sent una de les eines més populars el 2025. És multiplataforma, lleuger i altament personalizable gràcies a les seves extensions, que permeten treballar amb qualsevol llenguatge de programació. A més, inclou control de versions integrat amb Git, un depurador pas a pas i una interfície intuïtiva i moderna que el fa molt còmode d'utilitzar.

La nostra experiència utilitzant VS Code: Vam començar a utilitzar VS Code perquè ja el dominàvem abans del TR, cosa que representava un gran avantatge inicial. Tanmateix, ens vam trobar amb diversos problemes a l'hora d'executar certs codis. Aquests inconvenients, vam haber de buscar una alternativa, l'Eclipse.

Eclipse: És un entorn integrat de desenvolupament (IDE) [18] de codi obert, escrit bàsicament en Java però dissenyat per treballar amb diversos llenguatges de programació. Va ser creat originalment per IBM [59] i actualment és mantingut per la Fundació Eclipse [78]. Té característiques semblants a VS Code, però està enfocat a Java [45], amb eines i extensions específiques per a aquest llenguatge, i ofereix una millor gestió de projectes i fitxers grans.

La nostra experiència utilitzant Eclipse: El vam escollir per la seva funcionalitat i també per l'experiència nostre tutor. Eclipse mostra d'un rendiment i estabilitatg superior a la de VS Code, de ofereix unes extensions més solides que la de VS Code, però finalment l'hem deixat de banda, ja que Eclipse està pensat per treballar amb projectes grans de JAVA, i nosaltres no gaudiem d'aquestes funcionalitats amb Python. Finalment vam trobar el Pycharm.

Pycharm: PyCharm, desenvolupat per JetBrains [79], és actualment l'entorn integrat de desenvolupament (IDE) més popular per a Python. Empreses com Twitter, Facebook, Amazon i Pinterest l'utilitzen gràcies a les seves eines avançades i la seva gran adaptabilitat. Ofereix extensions específiques per a Python, integració amb frameworks populars, suport per a bases de dades i un sistema de depuració avançat que facilita molt el desenvolupament i la resolució d'errors.

La nostra experiència utilitzant Pycharm: Un cop haver fet una recerca molt amplia, finalment vam optar Pycharm. Després d'estar utilitzant molt de temps ens vam donar compte la excepcionalitat que conte en el enfocament en Python, un dels aspectes que més ens va sorprendre va ser la seva capacitat per estalviar-nos temps significatiu en el desenvolupament. La integració amb els principals frameworks de Python ens va permetre accelerar moltíssim tasques que abans requerien una configuració manual extensa. Totes els avantatges que ens aporta al treball amb Python ens va convecer en utilitzar-ho com a l'editor de codi definitiu.

Fonts: [4], [25], [19] i [64]

4.5 Editor de text i processador de text

Per tal d'aconseguir una bona presentació en el TR, havíem d'escollir entre un editor de text o un processador de text. Tot i que ambdós serveixen per crear documents, cadascun té característiques i funcions pròpies.

L'**editor de text** permet crear i modificar documents de manera senzilla. La seva simplicitat pot ser un inconvenient, ja que no permet utilitzar formats avançats ni opcions de personalització complexes. Tanmateix, ofereix avantatges com la possibilitat de navegar ràpidament entre fitxers, facilita la codificació i és compatible amb gairebé tots els llenguatges de programació, cosa que el fa ideal per a programadors.

El **processador de text** és un programa més elaborat i complex. Aquesta complexitat permet utilitzar formats avançats, cosa que el fa adequat per a documents professionals com currículums, llibres i altres publicacions. També ofereix opcions col·laboratives, com el disseny de pàgina, correcció ortogràfica i moltes altres funcionalitats.

Per a la part teòrica del nostre projecte sobre IA, vam seleccionar el format PDF per les següents raons:

1. **Connexió segura i estable:** Permet l'accés als documents sense dependre d'una connexió a Internet constant.
2. **Baixos requisits de recursos:** Pot obrir-se en gairebé qualsevol dispositiu sense necessitat de programes específics.
3. **Portabilitat:** Manté el format independentment del sistema operatiu utilitzat.
4. **Multifuncionalitat:** Admet text, fórmules matemàtiques, imatges i codi de programació integrat.

L'estalvi de temps va ser un factor decisiu, ja que permet evitar problemes de compatibilitat entre fórmules matemàtiques, imatges i el format del document. En la nostra recerca d'eines que complissin tots els requisits, vam identificar dos tipus principals de processadors de text: el **WYSIWYG** (What You See Is What You Get) i el **WYSIWYM** (What You See Is What You Mean).

1. **What You See Is What You Get (WYSIWYG):** WYSIWYG es refereix a un tipus de processador que permet als usuaris veure en temps real el resultat final del document o disseny mentre editen, sense necessitat de conèixer el codi o llenguatges de marcatge. Entre els avantatges hi ha la facilitat d'edició visual, ideal per a usuaris sense coneixements tècnics, el resultat immediat dels canvis i eines integrades com correcció ortogràfica i opcions de disseny. Tot i això, poden presentar poca precisió en contingut tècnic (com fórmules o codis), errors en documents llargs, dependència del programa i dificultats en el control de versions quan es treballa en equip.
2. **What You See Is What You Mean (WYSIWYM):** WYSIWYM es refereix a un tipus de processador que es centra en l'estructura del contingut més que en la seva aparença visual immediata. L'usuari marca el text segons la seva funció (títol, secció,

cita) i el format final s'aplica mitjançant un full d'estil com CSS o LaTeX. Aquest enfocament permet una gran precisió en elements tècnics com fórmules matemàtiques, codi o referències, també fa que els fitxers siguin lleugers i portables, i és ideal per al treball col·laboratiu, ja que es poden actualitzar amb sistemes com git sense afectar el format. Tot i això, pot resultar difícil de començar a utilitzar per la necessitat d'aprendre sintaxi específica, la previsualització no és immediata (cal compilar).

El LaTeX, la nostra elecció, és un sistema de composició tipogràfica de tipus WYSIWYM. Vam explorar les diferents distribucions i entorns de desenvolupament disponibles dins de l'ecosistema LaTeX. Per tal d'editar els fitxers vam triar Kile [48], un entorn integrat de desenvolupament per a LaTeX desenvolupat per la comunitat KDE [47].

LaTeX: es centra en l'estructura lògica del document més que en l'aparença visual. Dissenyat especialment per a la creació de documents acadèmics, tècnics i científics, LaTeX permet separar el contingut de la seva presentació. A diferència dels processadors de text tradicionals com Microsoft Word, es basa en codi per estructurar els documents. A més, LaTeX és programari lliure [33], distribuït sota la LaTeX Project Public License (LPPL) [58], cosa que permet utilitzar-lo, modificar-lo i redistribuir-lo lliurement. La seva creació va néixer de la necessitat de gestionar fórmules matemàtiques avançades i documents complexos amb una precisió tipogràfica que els processadors de text de l'època no podien oferir.

Fonts: [51], [80], [82] i [81]

5. Resultats

Un cop finalitzada la recerca, passem a la part pràctica. Tal com vam indicar als objectius, el nostre principal propòsit era construir una xarxa neuronal de tipus regressió, que analitza les dades disponibles i genera una predicció numèrica contínua. La nostra ambició ens va fer anar una mica més enllà i construir dues xarxes neuronals diferents com a part de la nostra pràctica. En aquest capítol mostrarem els resultats que hem aconseguit en aquest TR.

5.1 Xarxa neuronal de regressió

Una xarxa neuronal de regressió, a diferència d'altres tipus de xarxes, té una sortida lineal que permet predir un valor numèric continu. Aquestes xarxes necessiten un conjunt de dades ampli i ben estructurat perquè puguin aprendre les relacions entre les variables d'entrada i generar prediccions fiables.

A continuació es mostra un esquema representatiu d'una xarxa neuronal de regressió:

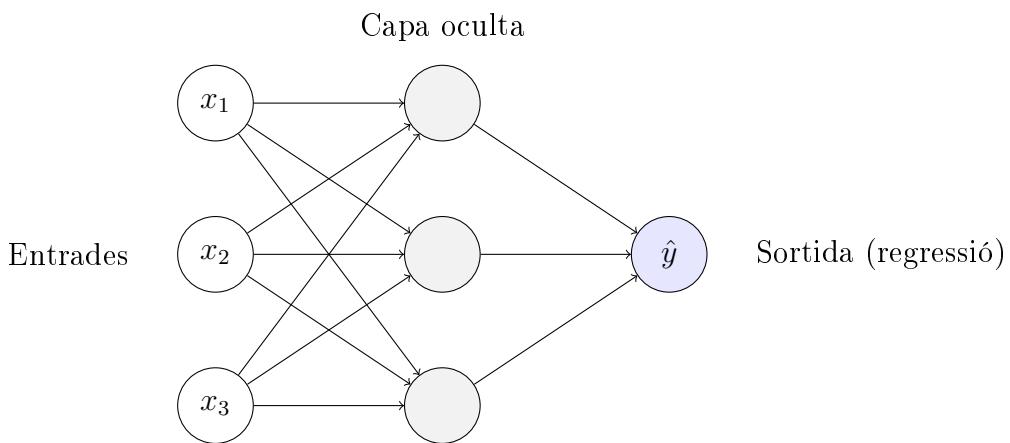


Figura 5.1: Esquema d'una xarxa neuronal de regressió

Abans de crear la xarxa neuronal, vam definir que prediria exactament: les notes finals de matemàtiques dels alumnes. Les dades es recolliren mitjançant un formulari amb les següents preguntes:

1. Realització de deures
2. Hores d'estudi setmanals

3. Hores de secció
4. Interès en la matèria
5. Nota del segon trimestre
6. Nota del tercer trimestre
7. Nota final

D'aquesta manera, la xarxa neuronal aprendrà a predir les notes dels alumnes comprendent la relació entre les dades d'entrada i el resultat final.

Per recollir les respostes vam utilitzar Google Drive per crear el formulari i el vam compartir en diversos grups de WhatsApp de batxillerat. Finalment, vam obtenir 18 respostes, tal com es mostra a la imatge següent:



Figura 5.2: Resposta a una de les preguntes del formulari

Un cop recopilades les dades necessàries, vam començar la construcció de la xarxa neuronal.

5.2 Xarxa Neuronal amb llenguatge de programació

En aquest apartat explicarem pas a pas la creació de la nostra xarxa neuronal de regressió amb un llenguatge de programació.

5.2.1 De Celcius a Fahrenheit

Abans de començar a treballar amb la xarxa neuronal definitiva, vàrem crear una de més simple per entendre millor com funciona una xarxa neuronal de regressió. En aquest cas, vaig triar un transformador d'unitats de temperatura, de Celsius (unitat basada en el punt de fusió de l'aigua) a Fahrenheit (sistema basat en els punts d'ebullició i congelació de l'aigua). Aquesta xarxa es va desenvolupar amb el llenguatge de programació Python. Una explicació detallada es pot trobar a l'apèndix B: De Celsius a Fahrenheit.

5.2.2 Predicció de les notes finals de matemàtiques

Un cop desenvolupada la xarxa neuronal de Celsius a Fahrenheit, ja podem crear la xarxa neuronal que ens havíem proposat. Per començar hem d'importar els frameworks i les biblioteques necessàries: `numpy`, `TensorFlow`, `matplotlib`, `MinMaxScaler`, `random` i `os`.

- **MinMaxScaler:** eina de la llibreria `sklearn` que normalitza les dades i les converteix dins del rang [0, 1], facilitant que la xarxa neuronal entengui les relacions numèriques.
- **random:** llibreria bàsica de Python que genera i selecciona nombres aleatoris.
- **os:** llibreria estàndard de Python que permet treballar amb el sistema operatiu.

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
import random
import os
```

Figura 5.3: Biblioteques i frameworks utilitzats

A diferència de la xarxa anterior, cal fixar els resultats de la predicció per garantir que les notes finals siguin reproducibles. Amb `random` podríem seleccionar qualsevol nombre, però

per assegurar la coherència assignem la llavor (`seed`) amb el valor 42. Així, la xarxa sempre s'inicialitzarà amb els mateixos pesos i biaixos.

També es controla l'aleatorietat de les diferents llibreries: `NumPy`, `TensorFlow`, `random`, i s'estableix el valor del hash de Python amb `os`.

Control de l'aleatorietat:

- **Llavor:** `seed = 42`
- **NumPy:** `np.random.seed(seed)`
- **TensorFlow:** `tf.random.set_seed(seed)`
- **random:** `random.seed(seed)`
- **Python:** `os.environ["PYTHONHASHSEED"] = str(seed)`

```
seed = 42
np.random.seed(seed)
tf.random.set_seed(seed)
random.seed(seed)
os.environ["PYTHONHASHSEED"] = str(seed)
```

Figura 5.4: Definició dels resultats de les prediccions

Un cop fixats els pesos i biaixos, s'emmagatzemen les dades per a l'entrenament. Les dades d'entrada es representen amb `X`, amb l'ordre de variables següent:

1. Realització de deures
2. Hores d'estudi setmanals
3. Hores de secció
4. Interès en la matèria
5. Nota del segon trimestre
6. Nota del tercer trimestre

Definim `X = np.array([])`, creant així una llista que anirem omplint amb les dades del formulari. L'estructura ha de ser sempre `[x1, x2, x3, x4, x5, x6]` amb `dtype=float` ja que hem de treballar amb nombres decimals.

```
X = np.array([
    [1, 0, 8, 8, 7, 9],
    [1, 3, 8, 8, 10, 8],
    [1, 5, 5, 8, 7, 0],
    [1, 1, 6, 10, 8, 10],
    [0, 2.5, 7, 9, 9, 7],
    [0, 0, 8, 8, 6, 5],
    [1, 4, 6, 5, 8, 8],
    [1, 10, 4, 6, 5, 4],
    [0, 1, 6, 7, 8, 7],
    [1, 5, 6, 1, 6, 6],
    [1, 3, 6, 10, 6, 6],
    [0, 1, 8, 3, 1, 1],
    [1, 2, 6, 6, 9, 3],
    [1, 1, 3, 10, 9, 9],
    [1, 1, 7, 10, 9, 9],
    [1, 1, 7, 8, 10, 9],
    [0, 0, 7, 10, 10, 10],
    [1, 0, 7, 10, 9, 10]
], dtype=float)

y = np.array([8,8,6,9,8,6,7,4,8,6,6,1,9,9,8,9,10,10], dtype=float)
```

Figura 5.5: Emmagatzematge de les dades

La sortida, y, tindrà una sola columna amb els valors de les notes recollides al formulari.

La normalització és crucial perquè la xarxa entengui millor les relacions entre les dades, ja que els valors poden variar molt (hores d'estudi, notes, etc.). Per això utilitzem `MinMaxScaler` tant per X com per y.

- Creem l'escalador amb `scaler = MinMaxScaler()`
- Normalitzem X amb `X_scaled = scaler.fit_transform(X)`
- Normalitzem y amb `scaler_y` i `y_scaled = scaler_y.fit_transform(y.reshape(-1,1))`

```
scaler = MinMaxScaler()
X_scaled = scaler.fit_transform(X)

scaler_y = MinMaxScaler()
y_scaled = scaler_y.fit_transform(y.reshape(-1, 1))
```

Figura 5.6: Normalització de les dades

$$x' = \frac{x - m_j}{M_j - m_j}$$

Figura 5.7: Normalització Min-Max[53]

A continuació, definim l'estructura de la xarxa neuronal amb un model seqüencial:

- **Primera capa oculta:**

`tf.keras.layers.Dense(32, activation="relu", input_shape=[6])` 32 neurones amb funció d'activació ReLU. L'entrada té 6 característiques corresponents a \mathbf{X} .

- **Segona capa oculta:**

`tf.keras.layers.Dense(16, activation="relu")` 16 neurones amb ReLU. L'entrada es dedueix automàticament de la capa anterior.

- **Capa de sortida:**

`tf.keras.layers.Dense(1)` Una sola neurona per obtenir la predicció de la nota final. No s'aplica l'activació per mantenir la funció lineal.

```
model = tf.keras.Sequential([
    tf.keras.layers.Dense(32, activation="relu", input_shape=[6]),
    tf.keras.layers.Dense(16, activation="relu"),
    tf.keras.layers.Dense(1)
])
```

Figura 5.8: Estructura de la xarxa neuronal

Compilació del model amb `.compile()`:

- **Optimitzador:** `tf.keras.optimizers.Adam(0.01)` Ajusta automàticament la taxa d'aprenentatge dels pesos.
- **Funció de pèrdua:** `"mean_squared_error"` Penalitza més fortament els errors grans, millorant la precisió de la predicció.
- **Mètrica:** `["mae"]` Monitoritza l'error absolut mitjà entre prediccions i valors reals.

```
model.compile(
    optimizer=tf.keras.optimizers.Adam(0.01),
    loss="mean_squared_error",
    metrics = ["mae"]
)
```

Figura 5.9: Configuració de la xarxa

Entrenament del model amb `.fit()` utilitzant `X_scaled` i `y_scaled` durant 400 èpoques, amb `verbose=1` per mostrar el progrés i l'error per època.

```

historial = model.fit(X_scaled, y_scaled, epochs=400, verbose=1)
plt.plot(historial.history["loss"])
plt.xlabel("Època")
plt.ylabel("Pèrdua (loss)")
plt.show()

```

Figura 5.10: Entrenament de la xarxa i corba de pèrdua

Per tal predir una nova mostra:

- Normalització de l'entrada: `nou_entrada_scaled = scaler.transform(nou_entrada)`
- Predicció amb la xarxa: `prediccio = model.predict(nou_entrada_scaled)`
- Transformació inversa: `prediccio_real = scaler_y.inverse_transform(prediccio)`
- Mostra del resultat: `print("Nota prevista:", round(prediccio_real[0][0],2))`

```

nou_entrada = np.array([[1, 0, 7, 10, 9, 10]])
nou_entrada_scaled = scaler.transform(nou_entrada)

prediccio = model.predict(nou_entrada_scaled)
prediccio_real = scaler_y.inverse_transform(prediccio_scaled)
print("Nota prevista:", round(prediccio_real[0][0],2))

```

Figura 5.11: Resultat final i ajustaments

5.3 Xarxa Neuronal amb full de càlcul

En aquest apartat continuarem amb la xarxa neuronal de regressió però aquesta vegada utilitzarem un full de càlcul. L'estructura que utilitzarem per a aquesta pràctica serà la del perceptró.

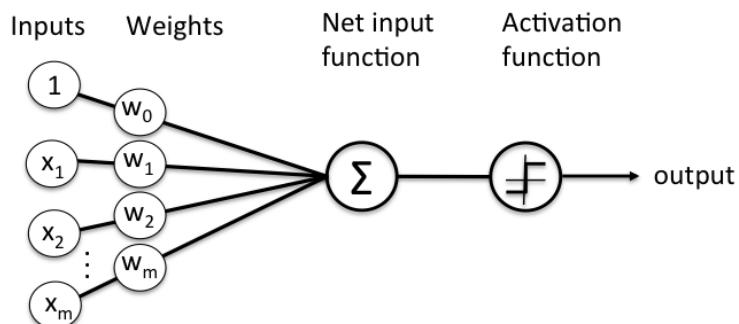


Figura 5.12: Estructura del perceptró. [60]

Ara començarem la pràctica ordenant les dades de cada alumne del formulari en el full de càlculs.

LES DADES							
Alumnes	Realització de deures	Hores d'estudis (semanals)	Hores de son	Interès en la matèria	Notes del segon trimestre	Notes del tercer trimestre	Nota final
1	1	0	8	8	7	9	8
2	1	3	8	8	10	8	8
3	1	5	5	8	7	0	6
4	1	1	6	10	8	10	9
5	0	2,5	7	9	9	7	8
6	0	0	8	8	6	5	6
7	1	4	6	5	8	8	7
8	1	10	4	6	5	4	4
9	0	1	6	7	8	7	8
10	1	5	6	1	6	6	6
11	1	3	6	10	6	6	6
12	0	1	8	3	1	1	1
13	1	2	6	6	9	3	9
14	1	1	3	10	9	9	9
15	1	1	7	10	9	9	8

Figura 5.13: Dades dels alumnes en el full de càlcul

Una vegada he ordenat tota la informació, he decidit representar els valors d'entrada d'una forma més senzilla d'entendre i curta, anomenant-los x_i .

- **Realització dels deures:** X1
- **Hores d'estudi:** X2
- **Hores de son:** X3
- **Interès en la matèria:** X4
- **Notes del segon trimestre:** X5
- **Notes del tercer trimestre:** X6
- **Nota final:** Y

Aquesta representació queda així:

L'entrada “Realització de deures” és una dada binària que només pot prendre valors 0 o 1.

LES DADES							
Alumnes	X1	X2	X3	X4	X5	X6	Y
1	1	0	8	8	7	9	8
2	1	3	8	8	10	8	8
3	1	5	5	8	7	0	6
4	1	1	6	10	8	10	9
5	0	2,5	7	9	9	7	8
6	0	0	8	8	6	5	6
7	1	4	6	5	8	8	7
8	1	10	4	6	5	4	4
9	0	1	6	7	8	7	8
10	1	5	6	1	6	6	6
11	1	3	6	10	6	6	6
12	0	1	8	3	1	1	1
13	1	2	6	6	9	3	9
14	1	1	3	10	9	9	9
15	1	1	7	10	9	9	8

Figura 5.14: Taula resumida

5.3.1 Normalització de dades

Abans de continuar, recordar què és la normalització de dades. La normalització de dades és una tècnica de processament que consisteix a transformar dades de diferents escales a una escala comuna, això facilita la comparació i l'anàlisi de la xarxa neuronal i millora el seu rendiment. En el nostre cas, tenim dades binàries i dades ordinàries que poden prendre qualsevol valor, aquest desequilibri afecta els càlculs posteriors si no es resol.

Per aquesta raó, convertirem totes les dades en valors d'entre 0 i 1. Aquest procés implica calcular la mitjana de les dades i la desviació estàndard de cada variable. Per això utilitzant la fórmula següent:

$$z = \frac{x - \mu}{\sigma}$$

z és el valor normalitzat

x és el valor original

μ és la mitjana

σ és la desviació típica

Aquests càlculs són fàcils d'obtenir amb les funcions que ens proporciona el full de càlcul.

LES DADES														
	x1	X1	x2	X2	x3	X3	x4	X4	x5	X5	x6	X6	y	Y
1	0,6030226892	0	-1,045097896	8	1,207019698	8	0,2827025641	7	-0,09231861823	9	1,035764607	8	0,541390292	
1	0,6030226892	3	0,1455199602	8	1,207019698	8	0,2827025641	10	1,292460655	8	0,6269101568	8	0,541390292	
1	0,6030226892	5	0,9392651974	5	-0,8820528563	8	0,2827025641	7	-0,09231861823	5	-0,5998531935	6	-0,4140043409	
1	0,6030226892	1	-0,648225277	6	-0,1856953382	10	1,053709557	8	0,3692744729	10	1,444619057	9	1,019087608	
0	-1,658312395	2,5	-0,05291634915	7	0,51066218	9	0,6682060607	9	0,8308675641	7	0,2180557067	8	0,541390292	
0	-1,658312395	0	-1,045097896	8	1,207019698	8	0,2827025641	6	-0,5539117094	5	-0,5998531935	6	-0,4140043409	
1	0,6030226892	4	0,5423925788	6	-0,1856953382	5	-0,8738079255	8	0,3692744729	8	0,6269101568	7	0,06369297553	
1	0,6030226892	10	2,92362829	4	-1,578410375	6	-0,4883044289	5	-1,015504801	4	-1,008507644	4	-1,369398974	
0	-1,658312395	1	-0,648225277	6	-0,1856953382	7	-0,1028009324	8	0,3692744729	7	0,2180557067	8	0,541390292	
1	0,6030226892	5	0,9392651974	6	-0,1856953382	1	-0,415821912	6	-0,5539117094	6	-0,1907987434	6	-0,4140043409	
1	0,6030226892	3	0,1455199602	6	-0,1856953382	10	1,053709557	6	-0,5539117094	6	-0,1907987434	6	-0,4140043409	
0	-1,658312395	1	-0,648225277	8	1,207019698	3	-1,644814919	1	-2,861877165	1	-2,23507094	1	-2,802490923	
1	0,6030226892	2	-0,2513526584	6	-0,1856953382	6	-0,4883044289	9	0,8308675641	3	-1,417362094	9	1,019087608	
1	0,6030226892	1	-0,648225277	3	-2,274767893	10	1,053709557	9	0,8308675641	9	1,035764607	9	1,019087608	
1	0,6030226892	1	-0,648225277	7	0,51066218	10	1,053709557	9	0,8308675641	9	1,035764607	8	0,541390292	
Mitjana	0,7333333333	0	2,633333333	0	6,266666667	0	7,266666667	0	7,2	0	6,466666667	0	6,866666667	
Desviació estàndar	0,4422166387	1	2,519700159	1	1,436043949	1	2,594010194	1	2,166410241	1	2,445858177	1	2,093375796	1

Figura 5.15: Taula resumida

5.3.2 Els paràmetres del model

Ara que ja tenim totes les dades preparades, hem d'assignar a cada variable X el seu pes per determinar la seva importància en la predicción final. Al començament de l'entrenament, assignarem a tots els valors d'entrada el mateix pes, ja que es corregiran lentament durant l'entrenament. També hem d'afegir el biaix, que és la constant que ajuda a millorar l'ajust de les prediccions.

Pesos i Biaixos						
W1	W2	W3	W4	W5	W6	Biaix
0,224	0,224	0,224	0,224	0,224	0,224	0,224

Figura 5.16: Taula dels pesos

A la figura 5.16, podem observar com ha quedat la taula després d'afegir els pesos als valors d'entrada, cal afegir 6 columnes de pesos respecte a les 6 entrades i una columna més pel biaix.

5.3.3 Funció d'error del model

Fins ara, ja tenim les dades d'entrada normalitzades i els seus respectius pesos inicials assignats, ara podem aplicar la fórmula que s'utilitza en les xarxes neuronals per calcular la predicción temporal de la nota final. Hem de recordar que la fórmula és:

$$\sum w_i x_i + \text{biaix}$$

Després d'aplicar la fórmula, obtindrem la predicción de la nota final, tal com es veu a la figura 5.17.

Predicció de Y
0,670004842
1,155310402
0,2802163353
0,8146219561
0,3397100598
-0,3062646566
0,466389646
0,09764691594
-0,2257256429
-0,1800825188
0,4192935971
-1,532446956
0,02042336442
0,3584831595
0,9824194958

Figura 5.17: Prediccions del model

Hem de recordar que els valors de la predicción són erronis, ja que els pesos que hem assignat són aleatoris. Per aquesta raó, el següent pas de la pràctica és entrenar el nostre model per millorar els paràmetres. Per dur això, haurem d'aplicar un procés d'optimització per ajustar aquests paràmetres. En aquest cas utilitzarem l'algoritme gradient descendent.

Començarem afegint més columnes en el nostre full en la nostra taula. Si restem els valors de la predicción (Y) amb els valors reals (y) obtindrem la diferència entre la nota dels alumnes i les notes predictives.

Es pot veure a 5.18 dues columnes addicionals, la columna “error” emmagatzema els errors del model i l’altre conté els mateixos errors, però en valor absolut, és a dir, que tots els valors estan en positiu, d’aquesta manera serà més fàcil apreciar la diferència dels errors en les prediccions i els càlculs dels paràmetres.

Després de crear aquestes taules, calcularem la mitjana dels valors que estan en la taula dels errors en positiu, el resultat d’aquesta mitjana ens ajudarà a orientar-nos i determinar si en cada iteració l’error està augmentant o disminuint, com més baix sigui aquest valor, la predicción serà més precisa.

Predictió de Y	Error	Errors en positiu
0,670004842	0,12861455	0,12861455
1,155310402	0,6139201101	0,6139201101
0,2802163353	0,6942206762	0,6942206762
0,8146219561	-0,2044656524	0,2044656524
0,3397100598	-0,2016802322	0,2016802322
-0,3062646566	0,1077396843	0,1077396843
0,466389646	0,4026966705	0,4026966705
0,09764691594	1,46704589	1,46704589
-0,2257256429	-0,7671159349	0,7671159349
-0,1800825188	0,2339218221	0,2339218221
0,4192935971	0,833297938	0,833297938
-1,532446956	1,270043968	1,270043968
0,02042336442	-0,9986642441	0,9986642441
0,3584831595	-0,660604449	0,660604449
0,9824194958	0,4410292037	0,4410292037

Figura 5.18: Erros de la predicción

5.3.4 Canvis dels paràmetres

Un cop tenim la funció d’error de la primera predicción, el pas següent és entrenar el model per ajustar els valors dels pesos i del biaix adequadament perquè les següents prediccions siguin més precises. Per aconseguir això, hem de tenir en compte les dades següents: Quan la diferència de Y respecte a y és més gran, més lluny estaran els pesos dels seus valors ideals. Per tant, hem de trobar la forma de calcular de forma correcta els pesos respecte de la diferència dels valors de la predicción.

Gràcies a la fórmula de la xarxa neuronal sabem que si el valor d’entrada (X) és petit, el seu pes (W) també ho serà, ja que aquests es multipliquen. Hem de tenir en compte dues coses: La diferència de la predicción final (Y) i del valor real (y) i el valor d’entrada (X) respecte al seu pes (W). Per tant, fórmula que utilitzarem per calcular els canvis dels paràmetres serà:

$$\Delta W = (Y_{\text{original}} - Y_{\text{predicció}}) \times X$$

Aquesta fórmula serà la que ens ajudarà a ajustar els paràmetres per tal que la predicción sigui cada cop més precisa. Aplicant aquesta fórmula a cada una de les dades, obtindrem 7 columnes més.

ENTRENAMENT DEL MODEL									
Predictió de Y	Error	Errors en positiu	Canvi W1	Canvi W2	Canvi W3	Canvi W4	Canvi W5	Canvi W6	Canvi de Biaix
0,670004842	0,12861455	0,12861455 -0,07755749178	0,1344147955	-0,1552402953	-0,03635966306	0,01187351754	-0,1332143988	-0,12861455	
1,155310402	0,6139201101	0,6139201101 -0,3702077557	-0,08933762996	-0,741013666	-0,1735567893	-0,7934675878	-0,3848727525	-0,6139201101	
0,2802163533	0,6942206762	0,6942206762 -0,4186308191	-0,6520573205	0,6123393304	-0,1962579652	0,06408949358	0,4162916455	-0,6942206762	
0,8146219561	-0,2044656524	0,2044656524 -0,1232974275	-0,1325398042	-0,03796831847	0,215447412	0,07550394602	0,295374978	0,2044656524	
0,3397100588	-0,2016802322	0,2016802322 -0,3344488289	-0,01067218158	0,102990467	0,1347639534	0,1675695632	0,04397752556	0,2016802322	
-0,3062646566	0,1077396843	0,1077396843 -0,1786660539	0,1125985173	-0,1300439212	-0,03045828501	0,0596782727	0,06460644576	-0,1077396843	
0,466389646	0,4026966705	0,4026966705 -0,2428352292	-0,2184196856	0,07477889441	0,3518795422	-0,1487056007	-0,2524546328	-0,4026966705	
0,09764691594	1,46704589	1,46704589 -0,8846619576	-4,289096867	2,315600452	0,7163650055	1,489792144	1,479526993	-1,46704589	
-0,2257256429	-0,7671159349	0,7671159349 -1,272117863	-0,4972639395	-0,142449853	-0,07886023338	0,2832763326	0,1672740073	0,7671159349	
-0,1800825188	0,2339218221	0,2339218221 -0,1410601662	-0,2197146264	0,04343819187	0,5651134635	0,1295720364	0,04463198971	-0,2339218221	
0,4192935971	0,833297938	0,833297938 -0,5024975635	-0,1212614827	0,1547395424	-0,8780540013	0,4615734853	0,1589921994	-0,833297938	
-1,532446956	1,270043968	1,270043968 -2,106129654	0,8232746028	-1,5329668087	2,088987265	3,63470983	2,838638433	-1,270043968	
0,02042336442	-0,9986642441	0,9986642441 -0,602217198	-0,2510169126	-0,1854472945	-0,4876521734	0,8297577278	-1,415468844	0,9986642441	
0,3584831595	-0,660604449	0,660604449 -0,3983594713	-0,428220502	-1,50272179	0,6960852214	0,5488748094	0,6842307075	0,660604449	
0,9824194958	0,4410292037	0,4410292037 -0,2659506164	0,2858862778	-0,2252169346	-0,464716687	-0,3664368602	-0,4568024399	-0,4410292037	
Mitjana		0,601670735 -0,07341989913	-0,3702284506	-0,0899455521	0,1615150711	0,429844074	0,2367154572		-0,224

Figura 5.19: Canvis en els paràmetres

A la figura 5.19 es pot veure aquestes taules, els canvis del biaix es calculen amb la mateixa fórmula que els pesos, però sense restar-li cap entrada, ja que actua com una entrada constant. Ara és necessari calcular la mitjana dels canvis dels pesos, les mitjanes obtingudes a partir dels canvis seran uns dels valors que utilitzarem per ajustar els paràmetres.

5.3.5 La taxa d'aprenentatge

Abans de començar a entrenar el model, hem de parlar d'una variable molt important: la taxa d'aprenentatge. Aquesta taxa és un paràmetre que ajusta l'ample dels passos que fem per actualitzar els pesos i el biaix del model, el valor d'aquesta taxa no pot ser molt gran perquè podria dificultar trobar el punt de convergència, ni molt petit perquè ralentitzaria la xarxa.

Normalment, el valor de la taxa d'aprenentatge és 1, però de vegades aquest valor és massa gran i augmentaria l'error del model, si això passa hauríem de reduir el seu valor progressivament fins a un punt on l'error de la xarxa disminueixi significativament.

Finalment, per trobar el valor final dels canvis ajustats dels paràmetres haurem de multiplicar a mitjana dels canvis dels pesos per la taxa d'aprenentatge.

5.3.6 Époques d'entrenement

Ja podem començar a entrenar el model, ho farem copiant la mateixa taula que tenim a sota, com es veu en una de les següents figures 5.20.

Pesos i Biaixos							ENTRENAMENT DEL MODEL												
W1	W2	W3	W4	W5	W6	Biaix	Predictió de Y	Error	Error en posició	Canvi W1	Canvi W2	Canvi W3	Canvi W4	Canvi W5	Canvi W6	Canvi de Biaix			
0,15050801009	-0,1482284506	0,1340544479	0,3855150711	0,653844074	0,4607154572	0	0,9312495046	0,3898591216	0,3898591216	-0,2350939508	-0,4074410247	-0,1759577491	-0,1021419908	-0,0539126381	-0,4038237441	-0,3848370674	-0,932811652		
							1,474211457	0,9328211652	0,9328211652	-0,5625123275	-0,1357449098	-0,1579335251	-0,2637103553	-1,0263710353	-0,1263710353	-0,1579355253	-0,1579355253		
							-0,3924325633	0,02157177783	0,02157177783	-0,013008271386	-0,02035161989	0,019027448808	-0,00869389885	-0,01991476704	-0,0123558535	-0,0123558535	-0,0123558535		
							1,473926079	0,4548384706	0,4548384706	-0,2742779177	0,2948377936	0,0844613862	-0,4792676453	-0,1679602365	-0,6570832525	-0,4548384706	-0,4548384706		
							0,72735085392	0,1864182472	0,1864182472	0,391139693	0,008954573051	-0,09519674851	-0,1245585025	-0,15488875	-0,04064956234	-0,1864182472	-0,1864182472		
							-0,4645347292	0,05053038638	0,05053038638	-0,037951920504	-0,02789020048	0,06099117172	-0,027893727	-0,0303070776	0,05053038638	-0,04957739476	-0,04957739476		
							0,1600057058	0,1163127302	0,1163127302	-0,0701932157	-0,06306716169	0,01659873178	0,1016349855	-0,04935132215	-0,02781751359	-0,1163127302	-0,1163127302		
							-1,865172921	-0,4957739476	-0,4957739476	0,2898629433	0,1449485731	-0,2409681144	-0,199991856	-0,14588875	-0,04064956234	-0,1865172921	-0,1865172921		
							0,1224569957	-0,4189251963	-0,4189251963	-0,694708455	-0,2715579014	-0,07779245959	-0,043069075	0,1546983811	0,0913492974	0,4189251963	-0,4189251963		
							-1,4528489395	-0,138844597	-0,138844597	0,6498446862	0,9757505755	0,198265888	-0,50396354	-0,755248186	-0,1852102437	-0,1852102437	-0,1852102437		
							0,00077586884	0,4147802098	0,4147802098	-0,2501218775	-0,0638587996	0,0772275133	-0,4370767112	0,227951615	0,07913954281	-0,4147802098	-0,4147802098	-0,4147802098	
							-3,5261876374	-0,7256767072	-0,7256767072	0,1303868778	0,8759060891	-0,19330869	-0,162788959	-0,162788959	-0,162788959	-0,162788959	-0,162788959		
							-0,1952366874	-1,2144142476	-1,2144142476	0,7323193624	-0,0352462567	-0,2255110696	-0,5930308695	0,009017431	-1,721247671	1,2144142476	-1,2144142476	-1,2144142476	
							1,307320954	0,2822333451	0,2822333451	-0,1738124688	0,1868403199	0,0565639389	-0,3037142304	-0,2394837373	-0,2985418973	-0,2822333451	-0,2822333451	-0,2822333451	
							1,68072044	1,139329952	1,139329952	-0,6870418115	0,3785424738	-0,5818127171	-1,20025895	-0,946832302	-1,1807784	-1,1807784	-1,1807784	-1,1807784	
								Màquina											
										Taxa d'aprenentatge							0,5		
										Causa anul·lat	-0,07603468182	0,05944227714	-0,05858620257	-0,2643552561	-0,1503525898	-0,2375372872		0	

Figura 5.20: Èpoques d'entrenament (Etapa 1)

Figura 5.21: Èpoques d'entrenament (Etapa 2)

Pesos i Biaixos							ENTRENAMENT DEL MODEL											
W1	W2	W3	W4	W5	W6	Biaix	Predictió de Y	Error	Error en posició	Canvi W1	Canvi W2	Canvi W3	Canvi W4	Canvi W5	Canvi W6	Canvi de Biaix		
0.06201162389	-0.07732381683	0.05037635752	0.1397739379	0.5433419007	0.235578269	0	0.42424265944	-0.1169616976	0.1169616976	0.07053055742	-0.122236424	0.1411750729	0.03306537182	-0.01079774231	0.1211447867	0.1169616976		
							0.9884585853	0.4470655843	-0.2659912818	-0.0655071082	-0.5396181495	-0.5771594461	-0.263868641	-0.280705899	-0.4470655843			
							-0.219518943	0.1944856466	0.1944856466	-0.117275276	-0.1726375993	0.1715466201	-0.0459815090	0.01795464616	0.116239391	0.1944856466		
							0.7784677528	0.145092322	-0.1557578276	-0.0448619857	0.1545778543	0.088747703	0.347640249	0.2447855726	0.1944856466			
							0.490028242	-0.05136204996	0.05136204996	-0.0517742097	0.02262386501	0.0343243037	0.04672506134	0.011997881	0.05136204996			
							-0.39709883	0.1690541092	0.1690541092	0.0203454247	0.0176671936	-0.039872030	0.0093440506	0.0101373385	-0.0160605042			
							0.2243534797	0.1606605042	0.1606605042	0.1606605042	0.096881979	0.0714106515	0.0298339065	0.1403864213	-0.059278229	-0.1007701979		
							-1.1137268	0.2556721538	0.2556721538	-0.1541761097	-0.074904319	0.40355558	0.154384545	0.258636299	0.257747313	0.2556721538		
							0.1424102942	-0.369879978	-0.6161834758	0.3897897967	-0.061834758	-0.04070825219	-0.0410151573	0.147331284	0.0669896539	0.2556721538		
							-0.7161077753	-0.3021034344	0.3021034344	0.182175254	0.2837552419	-0.056991941	-0.729828964	0.167386288	0.0574049565	0.3021034344		
							-0.169732388	0.2442220529	0.2442220529	-0.1473714391	-0.03553918341	0.03553918341	0.2573973113	0.1352774548	0.0465972801	-0.242220529		
							-2.335465577	0.4660043467	0.4660043467	0.7727708784	0.3020757968	-0.5624764259	0.7664900616	1.333647199	1.045152798	-0.4660043467		
							0.108629822	0.9102536263	0.9102536263	0.4509660467	-0.228799503	-0.4444444444	-0.1290160672	-0.1563904367	-0.4444444444			
							0.8271735508	0.1913740777	0.1510420910	-0.140535145	-0.4353316074	0.2016529495	0.159065137	0.1962184963	0.1913740777			
							0.96803352	0.42664306	-0.2577545453	0.2765608158	-0.2178704751	-0.4495576989	-0.35443838	-0.44190717813	-0.42664306			
							Mitjana	0.2948787653	0.004909767	-0.073500122	-0.08679411886	-0.03693777403	0.1106859684	0.00448177251	0			
							Taxa d'aprenentatge						0.225					

Figura 5.22: Èpoques d'entrenament (Etapa 3)

Figura 5.23: Èpoques d'entrenament (Etapa 4)

Pesos i Biaixos							ENTRENAMENT DEL MODEL									
W1	W2	W3	W4	W5	W6	Biaix	Predictió de Y	Error	Error en posició	Canvi W1	Canvi W2	Canvi W3	Canvi W4	Canvi W5	Canvi W6	Canvi de Biaix
0.07374237465	-0.1605057322	-0.04169637827	0.0799461923	0.6599848444	0.2199800883	0	0.351475629	-0.189972791	0.11557666	0.229300621	0.0537077763	0.015730301985	0.1967670291	0.189972791		
							0.9843100974	0.4249197754	0.4429197754	-0.267095741	-0.0645366807	-0.5346128937	-0.125214562	-0.5724563832	-0.2776709059	-0.4429197754
							-0.239736861	0.17426766	-0.10582678	-0.16385354	0.1537132872	-0.04925691432	0.160814957	0.1045030158	-0.17426766	
							0.8020949989	-0.2170385095	0.2170385095	0.13087791457	-0.140688944	0.0403030394	0.228695518	0.08614678122	0.135375976	0.2170385095
							0.5146935232	0.02669678901	0.02669678901	-0.0421185296	0.0136330302	0.0178330302	0.022181747944	0.0354203284	0.02669678901	
							-0.4797424527	-0.06573811175	0.06573811175	-0.1090125455	-0.0670727622	0.0739741958	0.1858433275	-0.0364130985	0.0354203284	0.06573811175
							0.2768774051	0.2131844295	0.2131844295	-0.16555048	0.156296525	0.095374544	0.1872322441	-0.1336474841	-0.2131844295	
							-1.290107833	0.079210904	0.079210904	-0.047314256	-0.231617652	0.15353787	0.0378170064	0.080520483	0.0799657079	-0.079210904
							0.2729587498	-0.2664315424	0.2664315424	-0.4544314544	-0.17404411	0.0498464568	-0.0275936575	0.0991249135	0.056530296	0.2664315424
							0.6993441318	-0.335399708	0.335399708	0.1720644765	0.268009994	0.2598630238	-0.883305538	-0.15805151	0.044525978	0.2683393708
							-0.2943998026	0.1196045384	0.1196045384	-0.0712425037	-0.01740494766	0.0222100952	-0.136234452	0.0662503543	0.0282309398	-0.1196045384
							-2.580615384	0.2217875539	0.2217875539	0.3679389565	0.1438253327	-0.267081461	0.364941668	0.534905385	0.4950705814	-0.2217875539
							0.29060165252	-0.72906200832	0.72906200832	-0.439612992	-0.183246441	0.135374671	-0.3598966653	0.065724126	-0.130329936	0.72906200832
							1.103660465	0.08477285653	0.08477285653	-0.0511195579	0.045190682	0.0743501664	-0.0879807422	-0.08477285653		
							0.987718188	0.4463278268	0.4463278268	-0.2691489868	0.2893209792	-0.227922741	-0.4072939868	-0.3708393143	-0.4622095661	-0.4463278268
							Mitjana									
							0.2376324955	-0.020956366419	-0.04023156183	-0.03020481352	-0.0682851813	0.02030721771	-0.05404779849			
							Taxa d'aprenentatge						0.225			
							Canvis ajustats	0.00471457445	0.00905210141	0.00769803042	-0.01536416655	0.04508373985	-0.01216075466			

Figura 5.24: Èpoques d'entrenament (Etapa 5)

Figura 5.25: Èpoques d'entrenament (Etapa 6)

Pesos i Biaixos							ENTRENAMENT DEL MODEL									
W1	W2	W3	W4	W5	W6	Biaix	Predictió de Y	Error	Error en positiu	Canvi W1	Canvi W2	Canvi W3	Canvi W4	Canvi W5	Canvi W6	Canvi de Biaix
0.06654360632	-0.1787592704	-0.05564020441	0.05300043999	0.6715062344	0.1993422202	0	0.319251668	-0.222138424	0.222138424	0.139545098	-0.232156399	0.2681254535	0.06279910205	-0.02650751236	0.2300831174	0.22138424
							0.954803504	0.134135584	0.4134135584	-0.249297757	-0.061959245	-0.498998305	-0.116873073	-0.534320758	-0.2591731587	-0.4134135584
							-0.2452428221	0.1687615188	0.1687615188	-0.1017670249	-0.0585116213	0.148856797	-0.0477093141	0.01557983023	0.1011983837	-0.1687615188
							0.7581264598	-0.2609611487	0.2609611487	0.1573654937	-0.1691616129	0.0485928676	0.274972565	0.09636629065	0.3768994486	0.2609611487
							0.5075115259	-0.03387786609	0.03387786609	-0.05618157775	-0.00179274061	0.01730060455	0.02263799683	0.02614876786	0.00738745828	0.03387786609
							-0.4671959775	-0.05319163651	0.05319163651	-0.0820835014	-0.05589406738	0.06420335304	0.01503741203	-0.0234634703	-0.0319865347	0.05319163651
							0.2801295982	0.2164363226	0.2164363226	-0.1305160133	-0.1173193452	0.04016121613	0.1891237741	-0.0792440897	-0.135680129	-0.2164363226
							-1.303515187	0.06588738704	0.06588738704	-0.03973158931	-0.1926302287	0.1039973352	0.0321731029	0.0669095783	0.0664479344	0.06588738704
							0.3018474414	-0.2395428502	0.2395428502	-0.397236546	-0.1552770089	0.04446195343	-0.2462520779	0.08845698589	0.0522336418	0.2395428502
							-0.329685952	0.0843083888	0.0843083888	-0.05083987139	-0.0122885533	0.0158567479	-0.08883655513	0.04669940381	0.0160859468	0.0843083888
							-0.251612092	0.2863700037	0.2863700037	0.474899267	0.185832275	-0.3458542354	0.4710256542	0.8195557743	0.6400572887	-0.2863700037
							0.3449033419	-0.6741842668	0.6741842668	0.4065484094	-0.1694580077	-0.1251923754	0.5610578393	0.9555632236	0.6741842668	
							1.102323568	0.08373595989	0.08373595989	-0.05049468371	0.0542797658	0.190479873	-0.08823338121	-0.06957349302	-0.08673074358	-0.08373595989
							0.9478416698	0.4064513778	0.4064513778	0.2450994028	0.2634720569	-0.2075593466	-0.4262817013	-0.3377072662	-0.4209678515	-0.4064513778
							Migra	0.2300486623	0.00606686792	-0.03961430519	-0.03109168736	-0.04262232511	0.03444167971	-0.02970841892	0	
							Taxa d'aprenentatge	0.225								
							Canvis ajustats	0.00158504528	0.008915216866	-0.00699562965	-0.00959002315	0.007749577934	0.00688439380	0		

Figura 5.26: Èpoques d'entrenament (Etapa 7)

Pesos i Biaixos							ENTRENAMENT DEL MODEL									
W1	W2	W3	W4	W5	W6	Biaix	Predictió de Y	Error	Error en positiu	Canvi W1	Canvi W2	Canvi W3	Canvi W4	Canvi W5	Canvi W6	Canvi de Biaix
0.06517856104	-0.1876724891	-0.06263583406	0.04341041684	0.6792556123	0.1926578264	0	0.3089500435	-0.2324402485	0.2324402485	0.2324402485	0.2324402485	0.2324402485	0.2324402485	0.2324402485	0.2324402485	0.2324402485
							0.947353107	0.4059636187	0.4059636187	-0.244905273	-0.05075080961	-0.05075080961	-0.117689559	-0.524620046	-0.2545021758	-0.4059636187
							-0.247885544	0.1683187867	0.1683187867	-0.100294002	-0.156217444	0.1457019609	-0.04701874746	0.01535432057	0.0973359157	-0.1653187867
							0.7474802817	-0.2716073265	0.2716073265	0.1637853206	-0.1760627347	-0.054362144	0.1020276525	0.3923891203	0.2716073268	
							0.5052474739	-0.03614281812	0.03614281812	-0.05939608328	-0.00191254549	0.01845670029	0.02415085012	0.03002889525	0.00788114774	0.03614281812
							-0.4670562599	0.05305191894	0.05305191894	-0.08797685476	-0.05454444884	0.0640347118	0.0149973152	-0.02838607311	-0.03181275261	0.05305191894
							0.2826217084	0.2191287328	0.2191287328	-0.1321395977	-0.1185537985	0.04069118415	0.1914764235	-0.089091684733	-0.1377402833	-0.2191287328
							-1.315797121	0.05360185246	0.05360185246	-0.0323231322	-0.1567118923	0.08460572006	0.02617402196	0.0544329385	0.0540578772	-0.05360185246
							0.3135780833	-0.2278122007	0.2278122007	-0.3777838094	-0.1476736321	-0.0423036513	-0.02341930746	0.06412523328	0.04967575126	0.2278122007
							-0.6432172487	-0.2292129077	0.2292129077	0.138220584	-0.2152917079	-0.04256378842	-0.553737585	-0.1299637135	-0.0473353476	0.2292129077
							-0.343638297	0.07036504392	0.07036504392	-0.04243171801	-0.0120395183	0.0130664063	0.07444319127	0.03897602178	0.01342551986	0.07036504392
							-0.507987197	0.2945037261	0.2945037261	0.4883791798	0.090947598	-0.3554717986	0.4844041223	0.8428334888	0.6582367358	0.2945037261
							0.3682153698	-0.6506722389	0.6506722389	0.8524907278	-0.1635984676	-0.120639405	0.3178237989	0.5470866317	0.9225216392	0.6506722389
							1.113101772	0.08401416317	0.08401416317	-0.0566926735	-0.0609423569	0.1238603998	-0.09906362225	-0.07811331875	-0.09737654276	-0.08401416317
							0.9386340358	0.3972437438	0.3972437438	-0.2395469906	0.5275034359	-0.2028573562	-0.415795294	-0.3300569416	-0.4114510102	-0.3972437438
							0.226816224	-0.00339248799	-0.003760472341	-0.02950171081	-0.03702958813	0.03434992765	-0.0255027023	0		
							Canvis ajustats	0.000763309799	0.008461062767	0.00663784939	-0.00833165733	0.00772873372	-0.00573958580	0		
							Migra	0.225								
							Taxa d'aprenentatge	0.225								
							Canvis ajustats	0.000763309799	0.008461062767	0.00663784939	-0.00833165733	0.00772873372	-0.00573958580	0		

Figura 5.27: Èpoques d'entrenament (Etapa 8)

Cada època d'entrenament representa una iteració, podem veure que els paràmetres de la primera època han canviat respecte els de la primera taula, que anomenarem època 0, on aquests paràmetres eren inicialment aleatoris, el valor del primer pes de l'època 1 (W_1) es calcula sumant el pes de l'època 0 pel valor del canvi ajustat.

Per saber si el model ha millorat, la millor forma de comprovar-ho és fixant-se en la taula dels errors en positiu. Podem veure que aquest valor ha disminuït respecte a l'època anterior, això significa que està millorant.

A partir d'ara hem de fer-ho mateix repetidament fins que el canvi de l'error entre les èpoques siguin molt petits, quan passi això significarà que estem pràcticament en el punt de convergència.

5.3.7 Resultats a la vuitena iteració

En aquest cas, han calgut 8 iteracions per arribar al punt de convergència. Hem organitzat els valors de l'error de cada època en una gràfica perquè sigui més fàcil visualitzar els canvis.

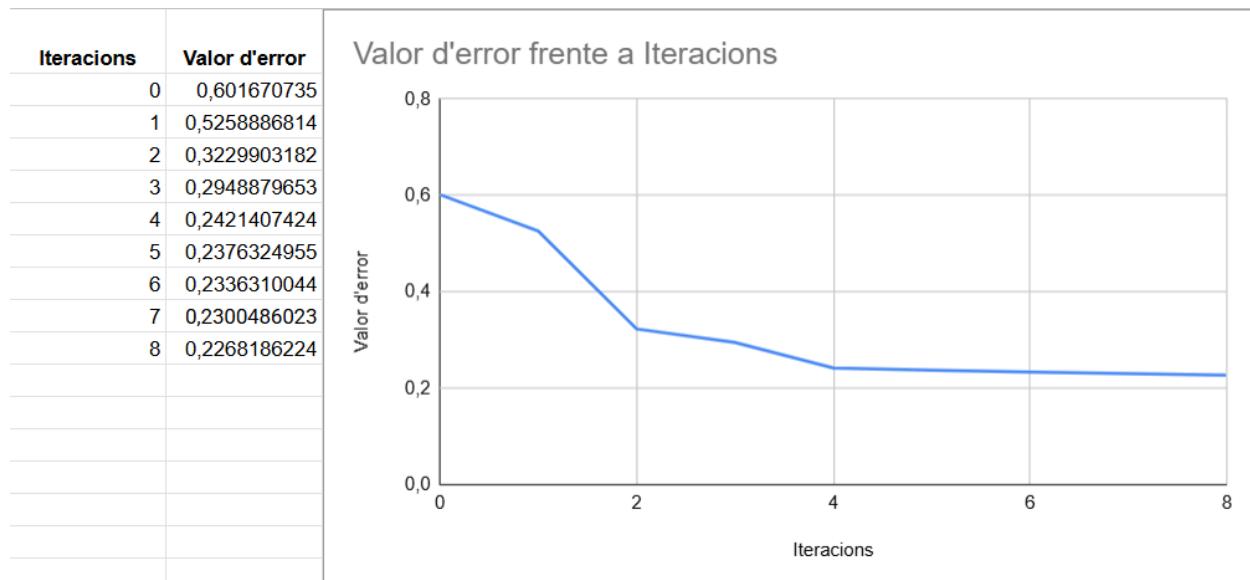


Figura 5.28: Gràfica dels valors d'error en cada època

A la figura 5.28, podem apreciar que l'error ha disminuït dràsticament entre l'època 0 fins a l'època 4, però després de la cinquena època l'error pràcticament no s'ha mogut, vaig ajustar la taxa d'aprenentatge moltes vegades, però l'error encara baixava molt lentament, fins que va arribar a l'etapa 8. Això vol dir que el model no podia ajustar més els pesos.

5.4 Resultats de la xarxa neuronal amb Python

Un cop finalitzada la construcció de la xarxa neuronal, vam avaluar-ne el rendiment, que va assolir una precisió del 93,75%. Aquest valor l'hem tret considerant correcte totes les prediccions amb un marge d'error menor a 0,5 punts, ja que les notes dels alumnes estan arrodonides. Mitjançant l'ús d'un full de càcul vam poder organitzar les dades obtingudes estadísticament.

NOTES ORIGINALS	NOTES PREDICTIVES DE LA XNA	MARGE D'ERROR	HA ENCERTAT LA NOTA?	TOTAL ENCERTES
8	8,01	0,01	Sí	15/16
8	7,89	0,11	Sí	
6	6,04	0,04	Sí	
9	8,55	0,45	Sí	
8	7,97	0,03	Sí	
6	6	0	Sí	
7	7,04	0,04	Sí	
4	3,98	0,02	Sí	
8	7,96	0,04	Sí	
6	6	0	Sí	
6	6,01	0,01	Sí	
1	1	0	Sí	
9	8,71	0,29	Sí	
9	9,09	0,09	Sí	
8	8,77	0,77	No	

Figura 5.29: Resultats obtinguts per la xarxa neuronal amb Python

5.5 Resultats de la xarxa neuronal en full de càlcul

Després de finalitzar totes les etapes, els pesos s'han ajustat correctament i, encara que l'error no s'hagi pogut aproximar-se molt al zero absolut, ja és un valor molt petit que podem donar per bo. A la següent figura podem veure els valors dels pesos finals del model.

Pesos i Biaixos						
W1	W2	W3	W4	W5	W6	Biaix
0,06517856104	-0,1876724891	-0,06263583406	0,04341041684	0,6792556123	0,1926578264	0

Figura 5.30: Pesos finals del model

Amb aquests pesos, el model hauria de ser capaç d'obtenir unes prediccions precises respecte a la nota final dels alumnes, per obtenir els valors finals de la predicció, hem de convertir els valors normalitzats de la columna de la predicció de Y de l'última etapa en valors normals, aïllant en la mateixa fórmula que vam emprar per normalitzar les dades.

$$z = \frac{x - \mu}{\sigma}$$

NOTES ORIGINALS	NOTES PREDICTIVES DE LA XNA	MARGE D'ERROR	HA ENCERTAT LA NOTA?
8	7,5135	0,4865	Sí
8	8,8503	0,8503	No
6	6,3477	0,3477	Sí
9	8,4319	0,5681	No
8	7,9247	0,0753	Sí
6	5,8891	0,1109	Sí
7	7,4583	0,4583	Sí
4	4,1113	0,1113	Sí
8	7,523	0,477	Sí
6	5,5199	0,4801	Sí
6	6,1471	0,1471	Sí
1	1,6175	0,6175	No
9	7,6367	1,3633	No
9	9,1971	0,1971	Sí
8	8,8307	0,8307	No

Figura 5.31: Prediccions finals de la pràctica

A la figura 5.31, després de desnormalitzar els valors, he organitzat les notes inicials i les prediccions finals en unes taules per visualitzar-los millor, a un costat tenim les notes originals i per l'altre les predictives. Això ens dona que la xarxa neuronal ha encertat 10 notes de 15, per tant té una precisió aproximada del 66,7%.

5.6 Comparació resultats Python i Full de càlcul

A continuació compararem les dues xarxes neuronals determinar quina de les dues és millor:

- **Precisió:** Un dels factors més importants a l'hora de crear una xarxa neuronal predictiva és el seu percentatge d'encert, en aquest cas, la precisió de la xarxa neuronal feta per Python (93,75%) ha sigut molt més alta que la del full de càlcul (66,7%)
- **Coneixements previs:** Fer una xarxa neuronal amb Python requereix coneixements de programació, aprendre el llenguatge és indispensable per crear una xarxa neuronal, en canvi, en la xarxa feta amb full de càlcul no ha calgut cap mena de coneixements de programació.
- **Dificultat:** Segons la experiència de cadascun de nosaltres, fer una xarxa neuronal amb full de càlcul és molt més senzill que fer-ho amb Python. No només cal saber programar, a més cal codificar tot el procés de manera manual. En canvi amb el full de càlcul el procés és molt més intuïtiu i mecànic.
- **Temps d'elaboració:** El temps que es requereix en fer una xarxa amb Python depèn de l'habilitat que tinguis en programar, si la persona és àgil i coneix bé el llenguatge pot fer la feina en molt poc temps, mentre que en un full de càlcul, malgrat que les dades i els càlculs s'obtenen automàticament, les etapes es fan manualment, cosa que es torna repetitiva i requereix molt de temps.
- **Visualització:** En una xarxa construïda amb un llenguatge de programació està tot automatitzat, la qual cosa, no mostra res de tot el procés que ha elaborat fins al resultat, en canvi, al full de càlcul, passa tot el contrari, es construeix tot manualment, això vol dir que et mostra tots els càlculs intermedis que porten al resultat final, per tant, resulta molt més didàctica.

6. Conclusions

Després dels resultats obtinguts, podem conoure que el treball està completat satisfactoriament. En aquest capítol presentem els assoliments dels objectius inicialment plantejats.

1. **Entorn de treball professional:** S'ha aconseguit reproduir de manera notable el funcionament dels centres d'investigació en xarxes neuronals, amb l'ajuda de Linux, Git, GitHub i d'altres eines.
2. **Escalabilitat:** La xarxa neuronal ha complert amb èxit aquest requisit, ja que és fàcilment ampliable mitjançant modificacions de paràmetres, incorporació de noves dades o augment de la seva complexitat.
3. **Llibertat:** La publicació del projecte a GitHub ha permès convertir-lo en programari lliure, fent-lo accessible, modificable i millorables per qualsevol persona.
4. **Simplicitat i eficiència:** El model s'ha simplificat al màxim sense perdre eficàcia, assolint un equilibri entre senzillesa i rendiment.
5. **Traçabilitat:** La publicació del projecte a GitHub dota al projecte de traçabilitat. El nostre repositori permet fer un seguiment pas per pas de l'elaboració del treball de recerca.
6. **Xarxa neuronal:** En conclusió, tal com s'ha exposat en l'apartat comparatiu, la xarxa neuronal és superior al full de càcul. Tanmateix, per a models simples continua sent recomanable l'ús del full de càcul.

Tot i que el treball ha estat dur i que ens hagut d'esforçar força en alguns moments, l'experiència ha estat majoritàriament positiva. Hem gaudit molt en l'elaboració d'aquest projecte i hem tingut una gran motivació al llarg del procés. Gràcies a aquesta experiència hem pogut adquirir coneixements que mai ens havíem plantejat, cosa que ens permetrà tenir un cert avantatge en els inicis dels estudis universitaris. A més, aquest treball ens ha ajudat a millorar en la redacció, que fins ara havia estat un dels nostres punts febles.

6.1 Recerca futura

En el camp de la recerca, el final d'un treball pot ser l'inici d'un nou projecte de recerca. Per això creiem convenient fer propostes de com es podria continuar la nostra recerca.

- Ampliar les sortides de la xarxa neuronal, ja que actualment només retorna un únic valor.
- Millorar la velocitat de resposta, perquè en alguns casos el temps d'espera per obtenir un resultat precís era excessiu.
- Desenvolupar nous models de la xarxa neuronal aplicats a casos reals, com el reconeixement de díigits, d'imatges o d'altres àmbits que hem considerat especialment interessants. Tanmateix, les limitacions en el temps i en l'extensió del treball no ens han permès dur-los a terme.
- Crear una plataforma amb HTML[56] i CSS[55], on poder publicar les noves actualitzacions i oferir guies a la comunitat per a principiants, tal com nosaltres ho érem en començar el TR.

A. Optimització i Ajustos

A.1 Mètodes d'Optimització en Intel·ligència Artificial: Fonaments i Aplicacions

Introducció

La intel·ligència artificial integra múltiples estratègies d'optimització, cadascuna adaptada als diferents models d'IA. A continuació, s'exploren els mètodes més destacats, amb una anàlisi dels mecanismes i les referències acadèmiques que hem utilitzat.

1. **Retropropagació en Aprendentatge Profund:** La retropropagació (*backpropagation*) és com el punt i coma en el text per les xarxes neuronals artificials. Aquest algorisme es basa en un procés iteratiu en dues fases:
 - (a) **Fase de propagació endavant:** Les dades d'entrada es transmeten a través de les capes de la xarxa, generant una predició. Durant aquest procés, cada neurona aplica una transformació lineal seguida d'una funció d'activació no lineal, com la *funció unitat rectificada uniforme* o la *funció sigmoide*.
 - (b) **Fase de propagació enrere:** Es calcula l'error entre la predició i el valor real utilitzant una funció de pèrdua (*loss function*), com l'error quadràtic mitjà (*Mean Squared Error*) per a problemes de regressió o l'entropia creuada (*Cross-Entropy Loss*) per a classificació. Mitjançant la regla de la cadena (*chain rule*), es determina la contribució de cada paràmetre a l'error global, obtenint els gradients $\partial L / \partial W$, on L representa la pèrdua i W els pesos de la xarxa.
 - (c) **Actualització de paràmetres:** Els pesos s'ajusten en la direcció oposada al gradient utilitzant variables del descens de gradient que s'actualitzen amb un subconjunt aleatori de dades, o optimitzadors més sofisticats com Adam (*Adaptive Moment Estimation*), que adapta la taxa d'aprenentatge per a cada paràmetre.

Podeu trobar més informació a 3.8.1 on s'explica la retropropagació de la xarxa neuronal.

2. **Optimització en Models Generatius:** Els models generatius, com les xarxes generatives adversàries (GANs) i els autocidificadors variacionals (VAEs), empren tècniques d'optimització especialitzades. **Exemples:**

(a) **Xarxes Generatives Adversàries (GANs):** Les GANs implica la competició entre dos models:

- i. **Generador (*Generator*):** Transforma un vector de soroll/aleatori (*noise vector*) en dades sintètiques, intentant enganyar el discriminador.
- ii. **Discriminador (*Discriminator*):** Distingeix entre dades reals i generades, actuant com un classificador binari.

L'entrenament es formula com un joc minimax, on la funció objectiu és:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]$$

La retropropagació s'aplica alternativament amb dues xarxes, ajustant els seus paràmetres per millorar les seves funcions respectives.

(b) **Autoencoders Variacionals (VAEs):** Els VAEs optimitzen l'*Evidence Lower Bound* (ELBO), que combina dos termes:

- i. **Terme de reconstrucció:** Minimitza l'error entre les dades originals i les reconstruïdes.
- ii. **Terme de regularització:** Minimitza la divergència de Kullback-Leibler (*KL divergence*) entre la distribució latent i una distribució prior (normalment una normal estàndard).

L'optimització es realitza mitjançant *gradient descent* sobre l'ELBO, amb l'ajut del *reparameterization trick* per a un càlcul eficient dels gradients.

(c) **Aprenentatge per Reforç i Optimització basada en Polítiques:** En l'aprenentatge per reforç (*Reinforcement Learning, RL*), l'optimització es centra en maximizar la recompensa acumulada. Un enfocament prominent és el *Policy Gradient*, que ajusta directament la política (*policy*) mitjançant:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot R(\tau) \right]$$

on τ representa una trajectòria i $R(\tau)$ la recompensa acumulada.

Alternatives a la Retropropagació: Algorismes Evolutius: Per a problemes on el càlcul de gradients no és factible, els algorismes genètics (*Genetic Algorithms, GA*) ofereixen una solució viable. Aquests algorismes emulen l'evolució natural mitjançant:

- i. **Selecció:** Els individus més aptes (*fitness*) tenen major probabilitat de reproduir-se.
- ii. **Creuament (*Crossover*):** Combina característiques de dos individus per generar descendència.
- iii. **Mutació:** Introduceix variabilitat genètica aleatòria.

Fonts: [66], [34], [49], [69] i [35]

A.2 Algoritme Gradient Descendent

L'algoritme de Gradient Descendent és un algoritme imprescindible en l'entrenament de les xarxes neuronals, i de la intel·ligència artificial. Primer de tot hem d'entendre que és un gradient i una funció d'error.

- **Gradient:** En les xarxes neuronals, un gradient és un vector que indica la direcció en la que es mou i amb quina intensitat per que la funció d'error cambiï més ràpid respecte als pesos de la xarxa neuronal. Per calcular el gradient, s'utilitza la derivada parcial de la funció de pèrdua respecte als pesos.
- **Funció d'error:** La funció d'error determina la diferència entre el valor estimat i el valor real, amb la finalitat d'optimitzar els paràmetres de la xarxa neuronal.

L'objectiu de l'algoritme gradient descendent és minimitzar la funció d'error.

A la figura A.1 podem apreciar quatre iteracions, el valor inicial és un punt qualsevol de la funció. En el punt 0 els paràmetres de la funció s'asiguen aleatoriament, per tant els errors poden ser grans. A cada itereació trobarem la derivada parcial, i el gradient serà l'encarregat de guiar els canvis als paràmetres.

A l'inici del procés el gradient acostuma a ser molt pronunciat, però a mesura que es van generant nous paràmetres durant l'entrenament, s'anirà reduint fins al punt més baix d'aquesta

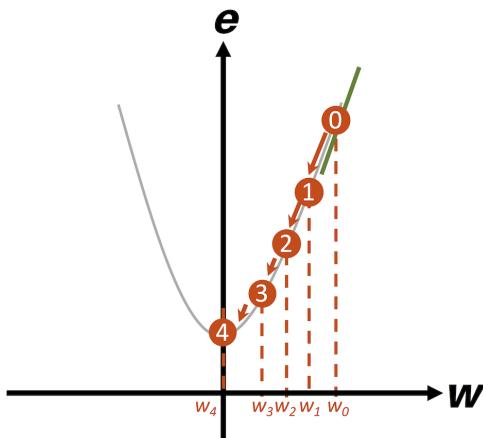


Figura A.1: Gràfica de gradient descendent. [10]

corba, on els errors són mínims, aquest rep el nom de punt de convergència. En aquest punt direm que la xarxa ha après i ha ajustat millor els pesos i els biaixos.

La fòrmula del gradient ascendent és la següent: $\Delta w_{ij} = a \left(\frac{\partial E}{\partial w_{ij}} \right)$

Aquesta és la fórmula més ràpida per arribar al punt màxim dels gradients. Per tant la fórmula del gradient descendent seria aquesta mateixa en negatiu, perquè és la forma més ràpida de trobar el punt mínim.

$$\Delta w_{ij} = -a \left(\frac{\partial E}{\partial w_{ij}} \right)$$

Explicació de la fórmula:

Δw_{ij} : Representa quant s'ajusta el pes en una iteració de l'entrenament.

a : És la constant d'aprenentatge, aquesta constant defineix quant afecta el gradiant en l'actualització dels nostres paràmetres en cada iteració. Si aquest valor és gran, cada iteració és molt gran, i el punt serà incapàc d'introduir-se al punt de convergència, causant que el procès d'optimització acabi en un bucle infinit. Tanmateix, si és petit, el punt s'aproxima poc a poc al punt de convergència, però calcularà moltes iteracions i això pot ser ineficient.

$\left(\frac{\partial E}{\partial w_{ij}} \right)$: És la derivada parcial de E respecte a Δw_{ij} .

La figura A.2 és una representació 2D del gradient descendent, però en situacions reals el gradient descendent acostuma a representar-se en 3D tal com mostra la figura A.3.

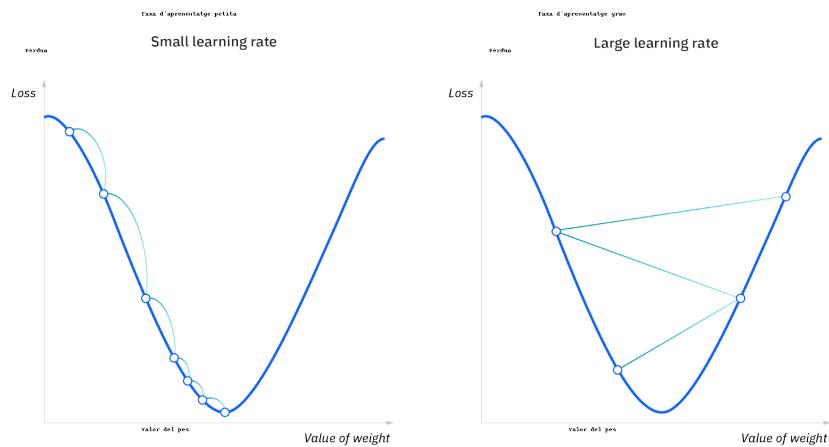


Figura A.2: Gràfiques de valors gran i petit de la constant d'aprenentatge. [29]

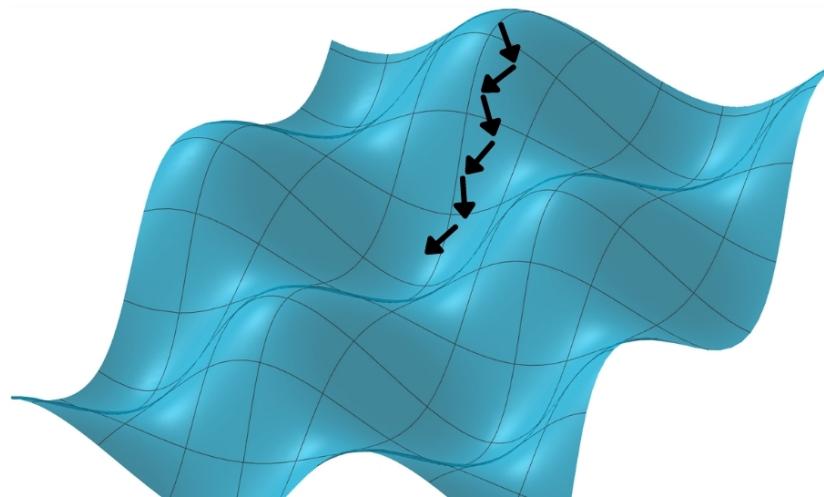


Figura A.3: Representació del gradient descendent en 3D. [24]

Fonts: [41] i [15].

B. De Celsius a Fahrenheit

En aquest capítol mostrarem amb detall la primera xarxa neuronal que vam crear.

Per començar, importem el framework i la biblioteca de Python que són necessàries per a la creació de la xarxa neuronal: **TensorFlow** (framework) i **numpy** (biblioteca). Hem creat unes *abbreviatures* per poder cridar-les més fàcilment quan es necessitin.

- **TensorFlow:** TensorFlow és un framework que ofereix eines per crear i entrenar models de Machine Learning (ML) i Deep Learning (DL), biblioteques per programar xarxes neuronals i altres algoritmes, compatibilitat amb GPU i TPU per accelerar càlculs, TensorBoard per visualitzar i monitorar entrenaments, models preentrenats i repositoris, i APIs en diversos llenguatges.
- **numpy:** numpy és una biblioteca científica per treballar amb vectors i matrius.

```
import tensorflow as tf  
import numpy as np
```

Figura B.1: Framework i biblioteca de Python

Definim les variables **celsius** i **fahrenheit** de tipus `np.array()`, ja que són llistes de valors, i el paràmetre `dtype=float` per especificar que els valors de la llista són nombres decimals. Per **fahrenheit**, les dades no poden ser arbitràries: han de ser les temperatures corresponents a les de **celsius** convertides a Fahrenheit, de manera que la xarxa neuronal pugui entendre la relació entre ambdues sèries de dades.

```
celsius = np.array([-40,-10,0,8,15,22,38], dtype=float)  
fahrenheit = np.array([-40,14,32,46,59,72,100], dtype=float)
```

Figura B.2: Agrupació de dades amb numpy

A continuació definim les capes de la xarxa neuronal. Utilitzant `tf.keras.layers.Dense` (funció per a capes denses), es crea la primera capa oculta amb `units=3` (3 neurones) i `input_shape=[1]` per indicar que l'entrada és un sol valor. La capa rep el nom `oculte_1`. La segona capa oculta, `oculte_2`, també té 3 neurones, però no cal especificar `input_shape`, ja que la xarxa ho dedueix automàticament de la capa anterior.

Finalment, la capa de sortida té `units=1` perquè retorna un sol valor (la predicció en Fahrenheit). Amb totes les capes creades, les integrem en un model seqüencial amb `tf.keras.Sequential`, passant-les dins d'una llista en l'ordre correcte: `[oculte_1, oculte_2, sortida]`. Això crea l'estructura bàsica de la xarxa neuronal.

```
oculte_1 = tf.keras.layers.Dense(units=3, input_shape=[1])
oculte_2 = tf.keras.layers.Dense(units=3)
sortida = tf.keras.layers.Dense(units=1)
model = tf.keras.Sequential([oculte_1, oculte_2, sortida])$
```

Figura B.3: Capes ocultes i capa de sortida d'una xarxa neuronal

Un cop definida l'estructura, cal ensenyar al model a aprendre i entrenar-se. Utilitzem la variable `model`, a la qual ja hem assignat les capes. A continuació apliquem `.compile()` per configurar el model abans de començar l'entrenament.

Dins de `.compile()` especifiquem `optimizer`, que indica com el model ajustarà els pesos. En aquest cas utilitzem "`adam`", un optimitzador adaptatiu que redueix la velocitat d'aprenentatge quan detecta canvis bruscos en un pes, l'augmenta quan el pes és estable i recorda la direcció correcta per evitar oscil·lacions innecessàries.

Definim la funció de pèrdua (que mesura l'error) amb `loss="mean_squared_error"`. La funció de pèrdua (*loss*) mesura l'error del model i guia l'aprenentatge. L'opció MSE (Mean Squared Error) penalitza més fortemet els errors grans i permet aconseguir prediccions més precises.

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$$

```
model.compile(
    optimizer = tf.keras.optimizers.Adam(0.1),
    loss='mean_squared_error'
)
```

Figura B.4: Optimització de la xarxa neuronal

Per entrenar la xarxa neuronal, utilitzem la variable `historial` per guardar l'evolució de l'entrenament, aplicant `.fit()` sobre el model. Dins `.fit()` indiquem les dades d'entrada (`celsius`), les dades de sortida (`fahrenheit`), el nombre de cicles d'entrenament (`epochs=100`) i si volem mostrar el procés per pantalla (`verbose=1`).

Podem utilitzar `print()` per escriure missatges al terminal, com "`Comencem a entrenar...`" abans de començar i "`Model entrenat!`" un cop finalitzat.

```
print('Començem a entrenar...')
historial = model.fit(celsius, fahrenheit, epochs=100, verbose=1)
print('Model entrenat!')
```

Figura B.5: L'entrenament de la xarxa neuronal

Després d'entrenar la xarxa, podem visualitzar la pèrdua per època amb `matplotlib.pyplot`. Primer importem la biblioteca amb `import matplotlib.pyplot as plt` i etiquetem els eixos amb `plt.xlabel("# Època")` i `plt.ylabel("Magnitud de pèrdua")`. La corba es dibuixa amb `plt.plot(historial.history["loss"])` i es mostra amb `plt.show()`. Això permet observar si el model aprèn correctament i com disminueix la pèrdua al llarg de les èpoques.

```
import matplotlib.pyplot as plt
plt.xlabel("# Època")
plt.ylabel("Magnitud de perdúa")
plt.plot(historial.history["loss"])
plt.show()
```

Figura B.6: Corba de pèrdua

Per fer prediccions amb nous valors, utilitzem `model.predict()`. Per exemple, amb 100 graus Celsius: `np.array([[100.0]], dtype=float)`. El resultat s'emmagatzema a la variable `resultat` i es mostra amb `print("El resultado es " + str(resultat) + "fahrenheit")`.

```
print("Fem una predicción!")
resultat = model.predict(np.array([[100.0]], dtype=float))
print("El resultado es" + str(resultat) + "fahrenheit")
```

Figura B.7: Resultat de la prediccó

Per inspeccionar els pesos interns de la xarxa, utilitzem `.get_weights()` en cada capa (`oculte_1.get_weights()`, `oculte_2.get_weights()`, `sortida.get_weights()`) i els mostrem amb `print()`.

```
print('Variables interns del model')
print(oculte_1.get_weights())
print(oculte_2.get_weights())
print(sortida.get_weights())
```

Figura B.8: Pesos i biaixos assignats

Finalment, es mostren exemples de l'inici de l'entrenament, del model entrenat, de la gràfica de pèrdua i del resultat de la prediccó amb 100 Celsius:

```
Començem a entrenar...
Epoch 1/100
1/1    1s 911ms/step - loss: 1951.4182
Epoch 2/100
1/1    0s 39ms/step - loss: 1301.1171
Epoch 3/100
1/1    0s 37ms/step - loss: 947.0582
Epoch 4/100
1/1    0s 40ms/step - loss: 1007.8066
Epoch 5/100
1/1    0s 38ms/step - loss: 1131.7876
Epoch 6/100
1/1    0s 46ms/step - loss: 990.7920
...
Epoch 100/100
1/1    0s 46ms/step - loss: 990.7920
Model entrenat!
```

Figura B.9: Inici de l'entrenament

```
1/1    0s 41ms/step - loss: 0.0664
Epoch 95/100
1/1    0s 39ms/step - loss: 0.0872
Epoch 96/100
1/1    0s 42ms/step - loss: 0.1259
Epoch 97/100
1/1    0s 42ms/step - loss: 0.1340
Epoch 98/100
1/1    0s 41ms/step - loss: 0.1507
Epoch 99/100
1/1    0s 41ms/step - loss: 0.1854
Epoch 100/100
1/1    0s 45ms/step - loss: 0.1929
Model entrenat!
```

Figura B.10: Model entrenat

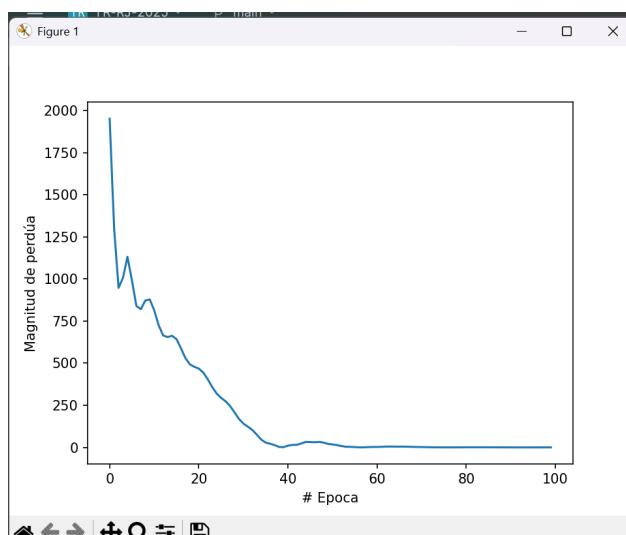


Figura B.11: Gràfica de la corba de pèrdua

```
Model entrenat!
Fem una prediccio!
1/1    0s 88ms/step
El resultat es[[211.72131]]fahrenheit
Variables interns del model
[array([[ 0.22799791, -0.54153144,  0.42276192]], dtype=float32), array([-3.2576356, -3.6861875,  3.8087313], dtype=float32)]
[array([[-0.9587477,  0.37814832,  0.15565477],
       [-1.2250756, -0.74850464, -0.3017928],
       [ 1.5424869, -0.17411143, -0.48074266]], dtype=float32), array([ 3.7836144, -0.6647431,  2.0843108], dtype=float32)]
[array([[1.6083411],
       [0.08999917],
       [0.15081894]], dtype=float32), array([3.6333761], dtype=float32)]

Process finished with exit code 0
```

Figura B.12: Resultat de la predicció i pesos utilitzats per la xarxa neuronal

Podeu trobar el codi complet al repositori del nostre TR [9]

Font: [70]

C. GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept

the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD,

and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute.

However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given

on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review

or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement

between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified

version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliografia

- [1] Ajuntament d'Olot. Entrevista a Pau Bramon, enginyer. <https://www.olotcultura.cat/entrevista-a-pau-bramon-enginyer/>. [Online; consultada el 03/10/2025].
- [2] Raül Sánchez Albaladejo. Les xarxes neuronals de propagació cap endavant una aproximació matemàtica. https://deposit.ub.edu/dspace/bitstream/2445/185653/2/tfgsanchez_albaladejo_raul.pdf. [Online; consultada el 27/09/2025].
- [3] Anònim. Fitxer:ai-ml-dl.svg. <https://ca.m.wikipedia.org/wiki/Fitxer:AI-ML-DL.svg>. [Online; consultada el 07/10/2025].
- [4] arsys. ¿Qué es Visual Studio Code y cuáles son sus ventajas? <https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>. [Online; consultada el 29/08/2025].
- [5] Atlassian. Qué es el control de versiones. <https://www.atlassian.com/es/git/tutorials/what-is-version-control>. [Online; consultada el 28/09/2025].
- [6] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. Dins Grégoire Montavon, Geneviève B. Orr i Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, pàgines 437–478. Springer, segona edició, 2012. Online; consultat el 09/08/2025 https://doi.org/10.1007/978-3-642-35289-8_26.
- [7] Carolina Bento. Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis. <https://medium.com/data-science/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment> [Online; consultada el 27/09/2025].
- [8] David Martínez Carpena. Dades personals de David Martínez. <https://dvmcarpena.com/>. [Online; consultada el 1/10/2025].
- [9] Rui Chen i Jiajun Xia. Repositori web del nostre Treball de Recerca. <https://github.com/Rui564/TR-RJ-2025>. [Online; consultada el 10/7/2025].
- [10] Codificando Bits. ¿Qué es el Gradiente Descendente? <https://codificandobits.com/blog/el-gradiente-descendente/>. [Online; consultada el 06/10/2025].

- [11] codificandobits. La Función de Activación. <https://codificandobits.com/blog/funcion-de-activacion/>. [Online; consultada el 27/09/2025].
- [12] European Comission. AI Act. https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai?utm_source=chatgpt.com. [Online; consultada el 12/08/2025].
- [13] Git community. Pàgina web dels creadors del Git. <https://git-scm.com/>. [Online; consultada el 25/07/2025].
- [14] PC Componentes. Arranque Dual: Qué es, cómo y por qué ejecutarlo. <https://www.pcccomponentes.com/arranque-dual-que-es-como-por-que-ejecutarlo?srsltid=AfmB0ooupDl3JP00iF9Cy5rlG0knqGdRIdKPQ2nuKstbcTwaiHkSnf5x>. [Online; consultada el 28/09/2025].
- [15] Dot CSV. ¿Qué es el Descenso del Gradiente? Algoritmo de Inteligencia Artificial | DotCSV. https://www.youtube.com/watch?v=A6FiCDoz8_4. [Online; consultada el 9/08/2025].
- [16] Escola d'administració pública de Catalunya. Intel·ligència artificial i protecció de dades. https://formaciooberta.eapc.gencat.cat/contingutsdelscursos/tdp/080_int_artificial/inici.html. [Online; consultada el 12/08/2025].
- [17] DataCamp. Introducción a las funciones de activación en las redes neuronales. <https://www.datacamp.com/es/tutorial/introduction-to-activation-functions-in-neural-networks>. [Online; consultada el 27/09/2025].
- [18] DataScientest. Ide : ¿Qué es un Entorno de Desarrollo Integrado? <https://datascientest.com/es/ide-que-es>. [Online; consultada el 28/09/2025].
- [19] DataScientest. PyCharm : Todo sobre el IDE de Python más popular. <https://datascientest.com/es/pycharm>. [Online; consultada el 1/09/2025].
- [20] Institut Obert de Catalunya. Sistemes de control de versions. https://ioc.xtec.cat/materials/FP/Recursos/fp_daw_m08_/web/fp_daw_m08_htmlindex/WebContent/u4/a2/continguts.html. [Online; consultada el 29/08/2025].

- [21] Universitat de ciències i tecnologia. Hidden layer, howpublished = "<https://msmk.university/hidden-layer/>", note = "[online; consultada el 16/08/2025]".
- [22] Universitat de ciències i tecnologia. Retropropagación: Aprendizaje Supervisado. <https://msmk.university/backpropagation/>. [Online; consultada el 19/08/2025].
- [23] Instituto de innovación digital de las profesiones. Tipos de frameworks de IA que debes conocer. <https://www.inesdi.com/blog/Frameworks-de-IA-que-debes-conocer/>. [Online; consultada el 12/08/2025].
- [24] Didàctica de la tecnologia. Gradiente Descendente Estocástico. <https://didatica.tech/gradiente-descendente-estocastico/>. [Online; consultada el 03/10/2025].
- [25] Eclipse. Eclipse IDE. <https://eclipseide.org/>. [Online; consultada el 29/08/2025].
- [26] Jesús Porta Etessam. Redes Neuronales: Biológicas vs. Artificiales. <https://ineurociencias.org/ann-in-neuroscience/>. [Online; consultada el 27/09/2025].
- [27] Universidad Europea. Qué es un sistema operativo y cómo funciona. <https://universidadeuropea.com/blog/que-es-sistema-operativo/>. [Online; consultada el 23/07/2025].
- [28] Geeks for Geeks. Feedforward Neural Network. <https://www.geeksforgeeks.org/nlp/feedforward-neural-network/>. [Online; consultada el 27/09/2025].
- [29] Gabriel Garau. Simple perceptron: Mathematical definition and properties. <https://blog.damavis.com/en/simple-perceptron-mathematical-definition-and-properties/>. [Online; consultada el 06/10/2025].
- [30] C Garcia. Què són els llenguatges de programació i com els podem classificar? <https://codelearn.cat/blog/que-son-els-llenguatges-de-programacio-i-com-els-podem-classificar/>. [Online; consultada el 28/09/2025].
- [31] Git. About. <https://git-scm.com/about/branching-and-merging>. [Online; consultada el 25/08/2025].

- [32] Ayuda Gmail. Usa Gmail con un lector de pantalla. <https://support.google.com/mail/answer/90559?hl=es-419>. [Online; consultada el 28/09/2025].
- [33] El Sistema Operatiu GNU. La definició de programari lliure. <https://www.gnu.org/philosophy/free-sw.ca.html>. [Online; consultada el 24/09/2025].
- [34] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville i Yoshua Bengio. Generative adversarial nets. Dins Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence i K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volum 27. Curran Associates, Inc., 2014.
- [35] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975. second edition, 1992.
- [36] Andrea Navarro i López. Estudi sobre les xarxes neuronals artificials. <https://deposit.ub.edu/dspace/bitstream/2445/98246/3/memoria.pdf>. [Online; consultada el 27/09/2025].
- [37] Ramon Sangüesa i Solé. Classificació: xarxes neuronals. <https://openaccess.uoc.edu/server/api/core/bitstreams/212854cd-d90e-4bb0-854a-df9803661e56/content>. [Online; consultada el 27/09/2025].
- [38] Universitat Rovira i Virgili. Dades personals de Gerard Finol. <https://www.urv.cat/html/grupsrecerca/investigadors/general-I78272.php>. [Online; consultada el 1/10/2025].
- [39] Iberdrola. Tipos de algoritmos de la inteligencia artificial. <https://www.iberdrola.com/conocenos/nuestro-modelo-innovacion/algoritmos-ia>. [Online; consultada el 24/09/2025].
- [40] IBM. What is machine learning. <https://www.ibm.com/think/topics/machine-learning>. [Online; consultada el 12/08/2025].
- [41] IBM. ¿Què és el descenso del gradiente? <https://www.ibm.com/es-es/think/topics/gradient-descent>. [Online; consultada el 9/08/2025].

- [42] IBM. ¿Qué son las redes neuronales? <https://www.ibm.com/es-es/think/topics/neural-networks>. [Online; consultada el 18/08/2025].
- [43] GitHub Inc. Pàgina web de Github. <https://github.com/>. [Online; consultada el 01/06/2025].
- [44] Universidad internacional de Valencia. ¿cómo funciona un algoritmo de backpropagation? [\{Universitat%20internacional%20de%20val%C3%A8ncia](https://www.universidadviu.com/es/actualidad/nuestros-expertos/como-funciona-un-algoritmo-de-backpropagation#que-es-backpropagation). [Online; consultada el 19/08/2025].
- [45] JAVA. ¿Qué es la tecnología Java y por qué la necesito? https://www.java.com/es/download/help/whatis_java.html. [Online; consultada el 28/09/2025].
- [46] Javi. La Función Sísmoide: Una Herramienta Clave en Redes Neuronales. <https://jacar.es/la-funcion-sigmoide-una-herramienta-clave-en-redes-neuronales/>. [Online; consultada el 18/08/2025].
- [47] KDE. La comunitat de KDE. <https://kde.org/ca/>. [Online; consultada el 23/07/2025].
- [48] Kile. Pàgina web del projecte de Kile a Sourceforge. <http://kile.sourceforge.net/>. [Online; consultada el 10/07/2025].
- [49] Diederik P Kingma i Max Welling. Auto-Encoding Variational Bayes, 2022.
- [50] Niranjan Kumar. Sigmoid Neuron — Building Block of Deep Neural Networks. <https://medium.com/data-science/sigmoid-neuron-deep-neural-networks-a4cd35b629d7>. [Online; consultada el 27/09/2025].
- [51] Latex. Latex. <https://manualdelatex.com/>. [Online; consultada el 19/08/2025].
- [52] Li Yin. Commonly used activation functions. <https://liyin2015.medium.com/commonly-used-activation-functions-9821ed348217>. [Online; consultada el 03/10/2025].

- [53] Abdul Moiz Malik. The questionable math behind Trump's reciprocal tariffs. <https://www.dawn.com/news/1901946>. [Online; consultada el 07/10/2025].
- [54] John McCarthy, Marvin L. Minsky, Nathaniel Rochester i Claude E. Shannon. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine*, 27(4):12, Dec. 2006. <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1904>.
- [55] M mdm_. CSS. <https://developer.mozilla.org/es/docs/Web/CSS>. [Online; consultada el 03/10/2025].
- [56] M mdn_. HTML: Lenguaje de etiquetas de hipertexto. <https://developer.mozilla.org/es/docs/Web/HTML>. [Online; consultada el 03/10/2025].
- [57] Wellington Migliari. Explicando la Propuesta Normativa para la Regulación del Uso de la Inteligencia Artificial en Unión Europea. <https://blogs.uoc.edu/digitapia/the-european-unions-artificial-intelligence-act-explained/>. [Online; consultada el 12/08/2025].
- [58] Frank Mittelbach. Reflections on the history of the LATEX Project Public License (LPPL) — A software license for LATEX and more. <https://www.latex-project.org/publications/2011-FMi-TUB-tb100mittbach-lppl-history.pdf>. [Online; consultada el 28/09/2025].
- [59] Netec. ¿Qué es IBM en Informática? <https://www.netec.com/que-es-ibm-en-informatica>. [Online; consultada el 28/09/2025].
- [60] Ninad Lunge. A Deep Architecture: Multi-Layer Perceptron. <https://medium.com/@nlunge786/a-deep-architecture-multi-layer-perceptron-164bc5ff3842>. [Online; consultada el 03/10/2025].
- [61] Nvidia. Supervize Me: What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning? <https://blogs.nvidia.com/blog/supervised-unsupervised-learning/>. [Online; consultada el 10/08/2025].
- [62] OpenAI. Introducing ChatGPT. <https://openai.com/index/chatgpt/>. [Online; consultada el 25/07/2025].

- [63] Pavan Belagatti. Understanding the Softmax Activation Function: A Comprehensive Guide. <https://www.singlestore.com/blog/a-guide-to-softmax-activation-function/>. [Online; consultada el 03/10/2025].
- [64] Pycharm. What's new in pycharm 2025.2. <https://www.jetbrains.com/pycharm/whatsnew/#>. [Online; consultada el 1/09/2025].
- [65] Python. El tutorial de Python. <https://docs.python.org/es/3/tutorial/>. [Online; consultada el 28/09/2025].
- [66] David E. Rumelhart, Geoffrey E. Hinton i Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct 1986.
- [67] Raghurami Etukuru Sergei Plis, Constantine Shulyak. Hidden layer. <https://es.linkedin.com/advice/0/what-some-examples-linear-nonlinear-models-real-world?lang=es>. [Online; consultada el 16/08/2025].
- [68] steveko. Steve Bennett blogs. <https://stevebennett.me/2012/02/24/10-things-i-hate-about-git/>. [Online; consultada el 29/08/2025].
- [69] Richard S Sutton, David McAllester, Satinder Singh i Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. Dins S. Solla, T. Leen i K. Müller, editors, *Advances in Neural Information Processing Systems*, volum 12. MIT Press, 1999.
- [70] Ringa Tech. Tu primera red neuronal en Python y Tensorflow. https://www.youtube.com/watch?v=iX_on3VxZzk. [Online; consultada el 10/08/2025].
- [71] Telefónica Tech. Las matemáticas del Machine Learning: Funciones de activación. <https://telefonicatech.com/blog/las-matematicas-del-machine-learning-funciones-de-activacion>. [Online; consultada el 03/10/2025].
- [72] A. M. Turing. I.-COMPUTING MACHINERY AND INTELLIGENCE. *Mind*, LIX(236):433–460, 10 1950. <https://doi.org/10.1093/mind/LIX.236.433>.

- [73] Ultralytics. Softmax. <https://www.ultralytics.com/es/glossary/softmax>. [Online; consultada el 27/09/2025].
- [74] Ultralytics INC. Comprender el impacto de la potencia de cálculo en las innovaciones de la IA. <https://www.ultralytics.com/es/blog/understanding-the-impact-of-compute-power-on-ai-innovations>. [Online; consultada el 12/08/2025].
- [75] Scientific United Nations Educational i Cultural Organization. Recommendation on the Ethics of Artificial Intelligence. https://www.unesco.org/en/legal-affairs/recommendation-ethics-artificial-intelligence?utm_source=chatgpt.com. [Online; consultada el 12/08/2025].
- [76] Berkeley University of California. What is machine learning (ml)? <https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/>. [Online; consultada el 10/08/2025].
- [77] Vim. Vim online. <https://www.vim.org/>. [Online; consultada el 25/07/2025].
- [78] Viquipèdia. Fundació Eclipse. https://ca.wikipedia.org/wiki/Fundaci%C3%B3_Eclipse. [Online; consultada el 28/09/2025].
- [79] Viquipèdia. JetBrains. <https://ca.wikipedia.org/wiki/JetBrains>. [Online; consultada el 28/09/2025].
- [80] Viquipèdia. WYSIWYG. <https://es.wikipedia.org/wiki/WYSIWYG>. [Online; consultada el 20/08/2025].
- [81] Viquipèdia. WYSIWYM. <https://en.wikipedia.org/wiki/WYSIWYM>. [Online; consultada el 25/08/2025].
- [82] Wix. WYSIWYG. <https://www.wix.com/encyclopedia/definition/wysiwyg>. [Online; consultada el 20/08/2025].
- [83] Ajedrez y Garry Kasparov. El día que Kasparov perdió contra Deep Blue. <https://ethic.es/el-dia-que-kasparov-perdio-contra-deep-blue>. [Online; consultada el 15/06/2025].

Índex de figures

Índex de taules