# Intel·ligència Artificial TR

**Jiajun Xia**

**Rui Chen**

**Tutor: Fernando García Vílchez**

**Departament de Matemàtiques**

**Institut Sants**

**29 d'agost de 2025**

# Índex de continguts

# 1. Answer

Dwar Ev va soldar cerimoniosament la connexió final amb or. Els ulls d'una dotzena de càmeres de televisió l'observaven i el subeter transmetia a tot l'univers una dotzena d'imatges del que ell feia.

Es va redreçar i va assentir amb el cap a Dwar Reyn, després es va col·locar al costat de l'interruptor que completaria el contacte quan el llancés. L'interruptor que connectaria, tot d'una, totes les monstruoses màquines de computació de tots els planetes habitats de l'univers —noranta-sis mil milions de planetes— en el supercircuit que les enllaçaria totes en una sola supercalculadora, una màquina cibernètica que combinaria tot el coneixement de totes les galàxies.

Dwar Reyn va parlar breument als bilions d'oients i espectadors. Després d'un moment de silenci, va dir:

—Ara, Dwar Ev.

Dwar Ev va activar l'interruptor. Hi va haver un brunzit poderós, l'alliberament d'energia procedent de noranta-sis mil milions de planetes. Les llums van centellejar i després es van apagar al llarg del tauler de quilòmetres de llargada.

Dwar Ev es va fer enrere i va respirar profundament.

—L'honor de fer la primera pregunta és teu, Dwar Reyn.

—Gràcies —va dir Dwar Reyn—. Serà una pregunta que cap màquina cibernètica per si sola no ha estat capaç de respondre.

Es va girar per mirar la màquina.

—Hi ha un Déu?

La veu poderosa va respondre sense vacil·lar, sense que fes clic cap relé:

—Sí, ara hi ha un Déu.

Una por sobtada va aparèixer al rostre de Dwar Ev. Va saltar per agafar l'interruptor.

Un llamp, vingut d'un cel sense núvols, el va fulminar i va fondre l'interruptor tancat.

*Brown, Fredric. "Answer."Angels and Spaceships, E. P. Dutton & Co., 1954.*

# 2. Introducció

Amb el pas del temps, la visio futurista del gran escritor *Fredric Brown* en el relat *Answer* cada vegada resulta menys ficticia. És probable de que algun dia nosaltres els humans puguem arribar a tal nivell. Malgrat els avenços que estem aconseguint en l'ambit IA encara ens falta un llarg cami per recorrer, que ens separa d'aquell somni esperada.

Actualment la IA ja no és un concepte ciencia-ficcio, sino ja hi es en el nostre dia quotidiana, des d'assistents virtuals fins a sistemes de diagnòstic mèdic avançat.

Tanmateix, les inteligencia artificials com chatgpt o deepseek ha provocat problemes etics i morals. Han sorgit molts problemes estil: Qui controla aquesta tecnologia? Quins treballs seran substituït? Com afronten els riscos de desevolupament descontrolat?

En aquest context voldriem construir una xarxa neuronal, no una IA per la seva complexibilitat i dificultat que esta fora del nostre nivell.

## 2.1 Motivacions:

Un cop explicat el que voliem fer, voldriem aclarir les Motivacions i els raons que ens empuja cap a endavant en l'elaboració d'aquest treball:

- La principal raó que ens va portar a aquest tema es per la nostra afició cap la informatica, i en un futur voldriem a profundir el tema i continuar treballant-hi en un grau o master o doctorat , per tant, ens agradaria anar-hi ja preparat aprofitant el temps per fer el TR.

- Una altre rao que ens van permetre fer el treball es gracies a la gran quantitat i diversitat de recursos que ens oferia a internet, també el nostre tutor Fernando Garcia era un matematic e informatic per tant ens facilitaba la feina, a la vegada un amic nostre ens va oferir recursos externes per elaborar la part practica xarxa neuronal del full de calculs.

- El nostre tutor ens va plantejar diversos temes per treballar-hi i finalment ens vam quedar amb aquest tema tant interessant.

- El nostre interès per la programació també ha sigut una motivació en aquet treball.
- La voluntat de voler construir una eina futuristic amb els coneixements que estem adquirint i disposat a fer un gran treball per afrontar-lo.

## 2.2 Estructura de la memoria

El primer pas que vam donar va ser donar-nos uns Goals (Objectius) per fixar i orientar-nos en el nostre treball, presentada en el capitol 2 Objectius.
Després d'haver fixat el nostre treball, per tant, saber que fer, ens toca adquirir coneixements previs i coneixements actualitzats per assabentar-nos com funciona les xarxes neuronals i poder donar lloc les creacions de les xarxes neuronals. Tot aquesta recerca s'explica en el capitol 3 Recerca prèvia
Una vegada que hem finalitzat amb la recerca prèvia, ens caldra iniciar un altre recerca per saber quins tipus d'eines farem utilitzar per guanyar temps i aumentar la calidad del treball, ja que encara que el treball no sigui de complexibilitat excessiva, és un treball elaborada i exigeix un gran esforç. Tota aquesta recerca es mostra en el capitol 4 Metodologia.
Tot i que el treball constant que hem de invertir i el gran esforç que farem el treball és mayoritariament positiva per part nostra. Graciés aquest treball ens podrem enrequirnos de coneixements que no ens donara a classe i podrem treure una gran advantatge al començar un grau, es mes ens ajudara en millorar la nostra redactacio i pujar el nivell de la llengua que es el nostre punt feble.
Dels assoliments que aconseguim, ilustrarem un apartat on registra tots els resultats que aconseguirem, explicada en el capitol 5 Resultats
Finalment per finalitzar el treball en el capitol 6 Conclusions hem donat les conclusions que hem arribat a elaborar i els futurs plans que podem tenir.

# 3. Objectius

Tal com podem veure en la Introducció, el nostre objectiu del TR (Treball de Recerca) és construir una xarxa neuronal, però no ens val qualsevol xarxa neuronal, ja que n'hi han d'infinites models, per dur a terme aquest estudi, vam seleccionar tres models de xarxes neuronals diferents: una implementada en Python, una altra basada en fulls de càlcul i una tercera aplicada a un exemple real del joc Mobile Legends: Bang Bang. A partir d'aquí desglossem 4 apartats amb diferents objectius: Xarxa neuronal amb llenguatge de programació Xarxa neuronal amb fulls de calculs Xarxa neuronal amb un cas real Objectius personals

## 3.1 Xarxa neuronal amb llenguatge de programació

En aquest apartat fixem els objectius, o més ben dit requisits de la xarxa neuronal que farem amb llenguatge de programació.

a) Fer ús de Python com a llenguatge principal

b) Utilitzar llibreries bàsiques com `NumPy` i `TensorFlow` per facilitar els càlculs

c) Crear una xarxa capaç de reconèixer patrons senzills (per exemple, classificació de dades)

d) Documentar el procés de disseny i d'implementació del codi

## 3.2 Xarxa neuronal amb fulls de calculs

Els objectius d'aquest apartat són:

a) Implementar manualment els càlculs bàsics d'una xarxa neuronal en un full de càlcul (propagació cap endavant i retropropagació)

b) Mostrar de forma visual com funcionen les operacions matemàtiques internes

c) Comparar l'eficiència i la dificultat respecte a la implementació en Python

## 3.3 Xarxa neuronal amb un cas real

Aquí volem aplicar els coneixements anteriors a un context més proper i pràctic:

a) Escollir un cas concret relacionat amb el joc *Mobile Legends: Bang Bang*

b) Recrear el cas real e intentar apropar-nos al màxim a la seva lògica

c) Comunicar-nos amb els treballadors del joc com a font de recursos

## 3.4 Objectius personals

A més dels objectius tècnics, també hi ha objectius d'aprenentatge personal:

a) Dominar les funcionalitats basiques del LaTeX, Vim, Github i Git

b) Millorar el nostre forma de redactar

c) enriquir de coneixements i gaudir del treball

# 4. Recerca prèvia

## 4.1 La Història de la Intel·ligència Artificial: Des dels Orígens fins Avui

### 4.1.1 El Naixement d'una Idea Revo lucionaria (1956)

Tot va començar amb una pregunta provocadora d'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): "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 "intel·ligència artificial" durant la famosa conferència de Dartmouth, marcant l'inici d'una nova era tecnològica.

### 4.1.2 El Joc que ho va canviar tot: The Imitation Game/El test de Turing

El nucli de la IA es basa en un experiment molt senzill, però profund: El joc d'imitació (The imitation game), proposat per Alan Turing. Davant de 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 en el fet que un avaluador havia de començar una conversa en forma de textos escrits amb una persona i una màquina durant 5 minuts, aquest avaluador no sabia qui era qui i el seu objectiu era esbrinar qui era l'humà. Si la màquina aconseguia enganyar a l'avaluador passava el test i es reconeixia que la màquina havia aconseguit un nivell de comportament lingüístic equivalent a la d'un humà, i donava resposta a la pregunta d'Alan Turing. Al cap dels anys, el joc ha estat evolucionant i moderat pels prodigis de la humanitat fins que avui en dia és conegut com el test de Turing. Tot això va ser clau en donar lloc al naixement de la IA i per l'avanç de la tecnologia.

### 4.1.3 ELs grans fites de la IA

#### 1997: La maquina que va vencer un campio

La supercomputadora Deep Blue desnvolupada per IBM va derrotar el campió mundial d'escacs, Garry Kasparov, 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 ambits

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: [3], [2], ChatGPT (2022), [4] )

## 4.2 Historia de les xarxes neuronals artificials

La història de les xarxes neuronals és molt extensa, per tant, resumirem molt aquest apartat, parlant molt breument de les creacions de xarxes més importants.

### 4.2.1 Perceptron (1958)

En la dècada de 1950 a 1960 el psicòleg i informàtic Frank Rosenblatt va crear el Percecptrò, la primera xarxa neuronal creada, a partir d'aquest moment, es potenciarien les xarxes neuronals.
Aquest model pren varies entrades binaries $x_1$, $x_2$, fins les que calguin, i produeix una sola sortida binaria. Per calcular la sortida, Rosenblatt va introduir els "pesos", que está explict a l'apartat 4.8, els seus principals usos són decsions binàries senziles, o per crear funcions lògiques com OR o AND.

### 4.2.2 Multiplayer Perceptron

El multiplayer 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, (tot això explicat a l'apartat 4.6) però amb valors d'entrada i rtida binàries. No hem d'ignorar que els científics asignaban el valor del pes i del umbral manualment en cada neurona, quan més perceptrons havien en les capes, era molt més difícil establir els pesos desitjats.

### 4.2.3 Neurones sigmoide

Per aconseguir que les xarxes neuronals aprenguin per elles mateixes, es a dir, aprenentatge automàtic **??**, va ser necessari introduir un nou tipus de neurones, que són les Neurones Sigmoides, que són similar al perceptró, aquestes neurones en comptes de que les entrades siguin 1 o 0, puguin tenir valors com 0.5, o 0.374 o qualsevol altre valor real. Ara les sorides en lloc de ser 0 o 1, serà d(w . x + b), on d serà la funció sigmoide, explicat en l'apartat 4.7.1. Aquesta va ser la primera funció d'activació **??**.

### 4.2.4 Xarxa neuronal prealmentada (Feedforward)

Les xarxes neuronals prealimentades són les que les sortides d'una sola capa són utilitzades com entrades en la pròxima, com explicarem més endevant en l'apartat 4.6 capa, on les conexions entre les unitats no dornen un cicle.

## 4.3 Que és la Ia?

Hem estat parlant molt durant aquest treball sobre la IA, i ara que em après quina és la seva història, hem de saber que és. Doncs bé, podem definir la IA com sistemes de software i de hardware disenyats per humans que actúen en un la dimensió física o digital, es a dir, raonar sobre el coneixemen, processant la informació derivada de dades i prendre les millors decisions per assolir l'objectiu donat. O dit d'un altre manera, és un camp de la informàtica que consisteix en un conjunt de capacitats interectuals i cognitives expresades per un sistema informàtic creat pels humans que té com a

proposit imitar la intel·ligència humana, com escriure poemes, reconeixer imatges, fer prediccions basades en dades i més funcions. Un exemple de IA i que tot el món coneix i utilitza és el ChatGPT, un chatbot impulsada per un model d'intel·ligència artificial generativa de la empresa OpenAI. Utilitza técniques 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. Aquesta IA ha avançat molt desde el seu llençament, ara pot generar o editar imatges, procesar audios, llegir arxius, comrendre imatges i molt més.

## 4.4 Com funciona la IA?

Una vegada que ja sabem que és una IA, ens toca entendre com funciona. Les intel·ligències arficials 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 l'aprenentatge profund. Per tant per funcionar necessitara: Dades Algorismes Potència computacional Software i Frameworks Mètodes d'Optimització en Intel·ligència Artificial:Fonaments i Aplicacions Ètica i Regulació

### 4.4.1 Dades

Les dades son fonamentals per la IA ja que es la base de l'aprennetatge del model, per poder raonar, prendre decisions, i millorar la presició. Aquí esdevenen uns exemples que pot haver:

**Basé d'aprenentage**

- Les algoritmes de la IA necessiten a base de dades i una gran diversitat de dades per poder identificar patrons i construir prediccion.

**Qualitat vs Quantitat**

- Una gran quantitat de dades ajudaran a la IA a obtendendre major precisió, però la qualitat es encara més important per la complexibilitat de dades que aporta, això evitara que la IA cometes errors per informacio incompleta. **Per exemple:**

En l'ambit medic si vols que la IA fagi una predicció i tan sol li dones una quantitat important de persones sanes i no d'altres exemplars, la IA simplement descartara d'altres possibilitats que podrian haver i només agafar la sana.

**Exemples Reals**

- Les sistemes dels cotxes o assistents virtuals necessiten dades en temps reals per adaptar-se de l'entorn, també plataformes com Netflix o Spotify necessiten dades personalitzades per poder generar recomanacions amb presició.

**Legisme i Etica**

- A la IA s'ha d'aplicar dades com protecció de dades per favorir l'etica i moral, i se li ha d'entrenar amb fonts legitima, valides per tal d'evitar erros legals o tecnics

### 4.4.2 Algorismes

**Algoritme Gredient 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 cambii 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 tracta de determinar el error entre el valor estimat i el valor real, amb la finalitat d'optimitzar els paràmetres de la xarxa neuronal, això està explicat en l'apartat 2.

L' objectiu de l'algoritme gradiant descendent és minimitzar la funció d'error, o també l'error entre la predicció, es a dir, trobar la funció de pèrdua mínima, per aconseguir-ho s'ha de trobar el valor mínim de la funció. La funció de pèrdua mínima significa que els errors dels paràmetres (pesos i biaixos) de la xarxa han de ser molt aproximades al valor real, on s'ajusten aquest paràmetres ja que no són exactes.

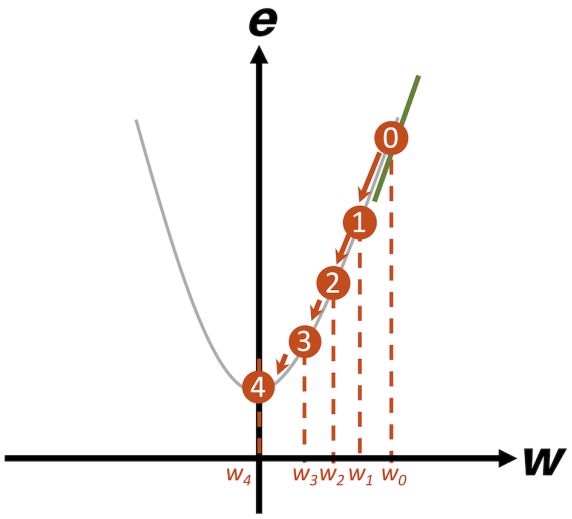


Figura 4.1: Gràfica de gradient descendent

En la figura d'adalt, on podem apreciar quatre iteracions, el valor inicial és un punt qualsevol de la funció. Al inici de la funció, en el punt 0, els paràmetres de la funció s'asignen aleatoriament, i la pèrdua és alta, es a dir, que els errors són grans. Desde aquell punt, trobarem la derivada parcial en cada una de les iteracions, i el gradiet serà l'encarregat de guiar els canvis als paràmetres. Al principi de tot, el gradient serà molt pronunciat, però a mesura que es van generant nous paràmetres durant l'entrenament, s'anirà reduint fins al put més baix d'aquesta corba, on els errors són molt petits, aquest punt se li diu punt de convergència. Aquí és on la xarxa ha après i ha ajustat millor les dades.

La fòrmula del gradient ascendent és la seguent:

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

Aquesta és la formula més ràpida per arribar al punt màxim dels gradients. Per tant la fòrmula 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 fòrmula:**

$\Delta w_{ij}$: Representa quant s'ajusta el pes en una iteració de l'entrenamen.

$a$: 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 incapaç 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 converència, però calcularà moltes iterai això pot ser ineficient. Per tant aquest valor ha de estar en un punt mig, en que no sigui molt gran ni molt petit.

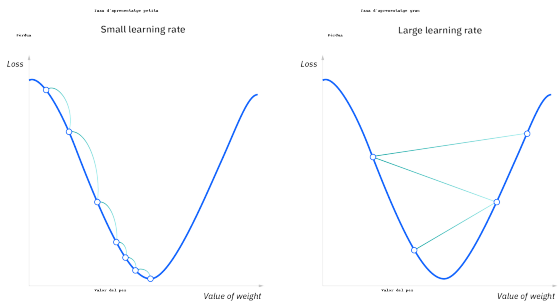


Figura 4.2: Gràfiques de valors gran i petit de la constant d'aprenentatge

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

A la figura 3.2, és una representació en 2 d, però en situacions reals el gradient descendent es representa en la dimensió 3 d.

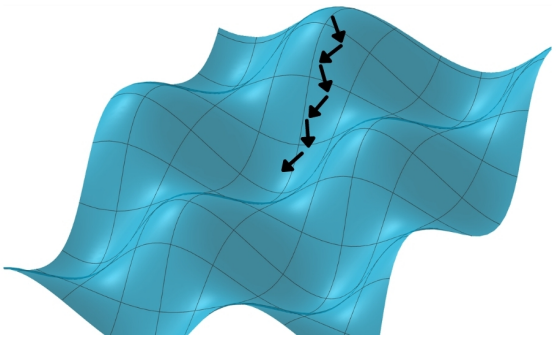

Figura 4.3: Repressentació del gradient descendent en 3d

Fonts: ibm.com i YouTube.

### L'aprenentage automàtic(Machine learning)

- L'aprenentatge automàtic és una branca crucial de la intel·ligencia artificial, consisteix en cobrar vida a la maquina, donar-li el poder d'apendre com els humans,

realitzar tasques de manera autonoma i finalment, les infinites possibilitats d'evolucionar a traves de l'experiencia i molt més. Segons la UC Berkeley el procés de l'aprenentatge automatic es pot dividir en tres parts:

1. **Mecanisme de predicció**

   Un conjunt de regles o operacions matemàtics que analitza les dades d'entrada i intenta identificar els patrons que busca el model.

2. **Funció de pèrdua (o error)**

   Un sistema per avaluar l'encert de les prediccions, comparant-les amb resultats reals (quan es disposa d'ells). Si la predicció és incorrecta, aquesta funció quantifica la magnitud de l'error.

3. **Algorisme d'optimització**

   El procés que ajusta automàticament el model per minimitzar l'error, modificant els paràmetres interns per millorar les prediccions futures. [1]

Segons Nvidia hi han molts tipus d'apranentage automatics:

Aprenentatge supervisat

Aprenentatge semi-supervisat

Aprenentatge no supervisat

Aprenentatge per reforç

### 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 aprén 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.

[1] Llengua original: Anglessa; Traduit per el chat bot Deepseek

**Avantatges i desavanantatge:** Aquest mètode destaca per la seva alta precisió en problemes ben definits, ja que al treballar amb dades prèviament etiquetades pot assolir bons resultats en tasques de classificació i regressió. Una altra avantatge important és la facilitat per avaluar el rendiment dels models. A més, es tracta d'una àmplia àrea d'estudi amb una gran varietat d'algoritmes ben desenvolupats i optimitzats, com ara els arbres de decisió, els random forests, les màquines de vectors de suport (SVM) o les xarxes neuronals. Finalment, un cop entrenat adequadament, el model pot generalitzar el seu aprenentatge i fer prediccions útils sobre dades noves.

No obstant això, aquest enfocament també presenta alguns inconvenients significatius. El principal desavantatge és la seva forta dependència de conjunts de dades etiquetades, que sovint són costosos d'obtenir i preparar com las de medicines. Un altre problema freqüent és el sobreajustament (overfitting), que ocorre quan el model memoritza les dades d'entrenament en lloc d'aprendre patrons generals, la qual cosa provoca una perdua de raonament logica. Finalment, aquest mètode pot tenir dificultats per manejar certs tipus de dades no estructurades o problemes complexos, que podrien requerir quantitats molt grans de dades etiquetades per assolir un bon rendiment.

### Aprenentatge semi-supervisat

L'aprenentatge semi-supervisat representa un punt intermig 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'obtencio de les dades etiquetades son molt costoses i l'extraccio de les característiques son molt complexes.

### 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'algorisme ha de descobrir per si mateix l'estructura i els patrons, fent una diagnosticació agrupant les característiques similars que poden haber 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.
- **Autoencoders:** Els autoencoders són un tipus de xarxa neuronal artificial que aprèn a comprimir i reconstruir dades.

### Aprenentatge per reforç

L'aprenentatge per reforç és una branca de l'aprenentatge automàtic inspirada en la manera com els éssers vius aprenen mitjançant la interacció amb el seu entorn. Com per exemple quan començem a jugar un videojoc, en els jocs rebem senyals de reforços, de si completem un nivell ens ortoga un trofeu, de si matem certs enemics guanyem bonificacions. Amb aquest sistema de penalitzacio i recompenses guia al jugador a millorar les seves tecniques de videojocs, i això el podem aplicar perfectament en la IA. Totes les aprenantatges de reforç segueixen pragmaticament aquest esquema per l'aprenentage:

1. **Acció**
2. **Observació**
3. **Recompensa**
4. **Ajust d'estrategia**

### Aprenentatge profund

- L'aprenentatge profund és una branca de l'aprenentatge automàtic que utilitza Que és una xarxa neuronal artificial/biologica? amb múltiples capes per processar dades complexes i extreure'n característiques rellevants. Inspirat en el funcionament del cervell humà, aquest enfocament permet identificar patrons i analitzar

dades de alta complexitat. Durant la fase d'identificació, empra un aprenentatge jeràrquic, és a dir, progressa gradualment des de característiques simples fins a patrons complexes.

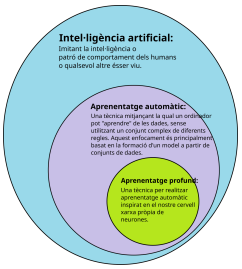


Figura 4.4: Les capes que te la IA per funcionar

### 4.4.3 Potència computacional

La intel·ligència artificial (IA) i la potència de càlcul estan profundament connectades. Sense maquinari potent, les aplicacions d'IA no podrien processar algoritmes complexos ni gestionar les enormes quantitats de dades que requereixen els models actuals.

Ultimament em vist el gran avanç que esta fent la IA, per tant la potencia computacional també creix consecutivament. D'aquesta manera la IA cada vegada es pareguera més a un huma i podrà ser més original alhora de crear contingut.

Una **GPUs** té un papel molt important en la IA de tal manera que pot lograr a accelelar els calculs que fa. Per la cual cosa fa que el seu mercat puguess arribar fins a 53 milions de dolars en 2023 i s'estima a que arribara fins al 473 milions en 2033.

Tot això provoca un gran comerç i invertiment en la creació d'un chip exclusiu solament per la IA, donant llavor una rapideza encara més fascinant alhora de fer calculs.

Uns dels principals protagonistes que treballen en el hardware de la IA, son mostrat per la següent imatge:

# GPUs

Segons Microsoft una GPUs és:


Figura 4.5: Les capes que te la IA per funcionar

La unitat de processament de gràfics (GPU) d'un dispositiu Windows controla el treball relacionat amb els gràfics. Les GPU també es coneixen com a targetes de vídeo o targetes gràfiques.
El treball relacionat amb els gràfics que manegen les GPU inclou els següents elements:

- El grafisme
- Efectes
- Video
- Videojocs

Hi ha dos tipus bàsics de GPU: GPU integrada i GPU discreta. Coneix els diferents tipus de GPU i troba el que s'ajusta a les teves necessitats:
**GPU integrada**

- Les GPU integrades estan integrades directament en el processador principal (CPU).
- Les GPU integrades normalment no són tan potents com les GPU discretes, però són més eficients energèticament.
- Les GPU integrades sovint són menys costoses que les GPU discretes.
- Les GPU integrades permeten que els portàtils siguin més prims, lleugers i eficients.
- Les GPU Integrades són excel·lents per a alguns jocs, edició de vídeo lleuger o per treballar amb fotos.

**GPU discreta**

- Les GPU discretes són més grans que les GPU integrades i utilitzen més potència, però són les més adequades per a tasques de processament intensiu, com ara edició intensa de fotos i vídeos, treball de disseny i jocs.

- En els ordinadors, les GPU discretes són una targeta independent de la CPU, i que es troba en la seva pròpia targeta, fora de la CPU.

- Els ordinadors portàtils i tauletes també poden tenir GPU discrets directament a la placa base, però sovint s'afegeixen a la GPU incorporada. La GPU integrada s'utilitza per a tasques gràfiques més lleugeres, mentre que la GPU discreta s'utilitza per a tasques gràfiques més pesades. El canvi entre GPUs a partir de la tasca que es fa permet un equilibri entre rendiment i eficiència energètica.

### 4.4.4 Software i Frameworks

**Frameworks**

Segons INESDI:
"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. En altres paraules, són marcs de treball que funcionen com un esquelet predefinit, i sobre els quals es pot construir una aplicació o 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."

Un Frameworks, en diferencia, de la biblioteca per el control total que pot tenir el usuari en tema estructura, i organitzacio dels codis, en canvi, una biblioteca està en mans del desenvolupador i només tens accés en els codis que et convenin de la biblioteca. Malgrat la utilitat que propociona no satisfeix la IA de la actualitat i han hagut de desarollar uns Frameworks especifics per la IA, de tal manera en que facilita el desarollament i l'entrenament dels models de la IA.

#### Software

El software és el conjunt de programes, instruccions i regles informàtiques que permeten executar tasques específiques en un ordinador o sistema.
La relació entre el software i la IA funciona de la següent manera: d'una banda, el software dóna utilitat pràctica a la IA en àmbits reals, permetent codificar les seves tècniques i processar les grans quantitats de dades que necessita. D'altra banda, la IA aporta automatització a les tasques repetitives del software, estalviant temps i recursos. Aquesta interacció crea un benefici mutu que forma un cicle perfecte, on cadascú millora i potencia l'altre.

### 4.4.5 Mètodes d'Optimització en Intel·ligència Artificial:Fonaments i Aplicacions

#### Introducció

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

#### Retropropagació en Aprenentatge 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:

- **Fase de propagació endavant**: Les dades d'entrada es transmeten a través de les capes de la xarxa, generant una predicció. Durant aquest procés, cada neurona aplica una transformació lineal seguida d'una funció d'activació no lineal, com *Funció ReLU(Funció Uniat Rectificada Uniforme)* o *Funció sigmoide.*

- **Fase de propagació enrere**: Es calcula l'error entre la predicció 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.

- **Actualització de paràmetres**: Els pesos s'ajusten en la direcció oposada al gradient, utilitzant variants del descens de gradient, com SGD *Algoritme Gredient descendent:*, que aplica updates 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.

Per més informació també podeu accedir aquest aparatat 4.8.1 on també explica la retropropagació de la xarxa neuronal.

**Suport teòric**:

*Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). "Learning representations by back-propagating errors." Nature, 323(6088), 533-536.*

Aquest article seminal estableix els fonaments matemàtics de la retropropagació, demostrant la seva eficàcia per a l'entrenament de xarxes multicapa.

### Optimització en Models Generatius

Els models generatius, com les xarxes generatives adversàries (GANs) i els autoencoders variacionals (VAEs), empren tècniques d'optimització especialitzades:

**Xarxes Generatives Adversàries (GANs)** El marc adversarial de les GANs implica la competició entre dos models:

- **Generador (*Generator*)**: Transforma un vector de soroll/aleatori(*noise vector*) en dades sintètiques, intentant enganyar el discriminador.
- **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.

**Suport teòric**:

*Goodfellow, I., et al. (2014). "Generative Adversarial Networks." Advances in Neural Information Processing Systems 27 (NeurIPS).*

Aquesta publicació introdueix el concepte de GANs, destacant el seu potencial per a la generació de dades realistes mitjançant aprenentatge adversarial.

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

- **Terme de reconstrucció**: Minimitza l'error entre les dades originals i les reconstruïdes.
- **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.

**Suport teòric**:

*Kingma, D. P., & Welling, M. (2013). "Auto-Encoding Variational Bayes." arXiv preprint arXiv:1312.6114.*

Els autors presenten el marc teòric dels VAEs, introduint tècniques clau com el *reparameterization trick* per a l'entrenament eficient de models generatius probabilístics.

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

**Suport teòric**:

*Sutton, R. S., et al. (2000). "Policy Gradient Methods for Reinforcement Learning with Function Approximation." Advances in Neural Information Processing Systems 12 (NeurIPS).*

Aquest treball estableix les bases teòriques dels mètodes de gradient de polítiques, demostrant la seva aplicabilitat en entorns complexos.

**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:

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

**Suport teòric**:

*Holland, J. H. (1975). "Adaptation in Natural and Artificial Systems." University of Michigan Press.*

Holland formalitza els algorismes genètics, inspirant-se en els mecanismes de selecció natural per a resoldre problemes d'optimització complexos.

### 4.4.6 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 per els humans, i per tant cal establir-hi 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, supervisió humana i 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'IA que utilitza 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 d'IA siguin transparents, responsables, segurs i que promoguin el benestar de la societat, ajudant a establir marcs internacionals de bones pràctiques.

Fonts:Bloc IA EAPC Wiki IBM ML La relació que estableix entre la IA i el maquinari [1] AI ACT IA UNESCO

## 4.5 Que és una xarxa neuronal artificial/biologica?

Una xarxa neuronal artificial és un model computacional inspirat en el funcionament del cervell humà, utilitzat en el camp de la intel·ligència artificial (IA) i l'aprenentatge automàtic (machine learning). Està dissenyada per reconèixer patrons, prendre decisions i aprendre a partir de dades, sense ser programada explícitament per a cada tasca 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. Aquí hi és una taula de comparació entre una biologica i artificial:

Fons: https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-basics-neural

| Aspecte | Xarxa neuronal biològica | Xarxa neuronal artificial |
|---|---|---|
| **Base** | Cèl·lules vives (neurones). | Algoritmes matemàtics. |
| **Energia** | Baix consum (~20 watts). | Alt consum (GPUs/TPUs). |
| **Aprenentatge** | Plasticitat sinàptica. | Backpropagation + dades. |
| **Velocitat** | Lent (mil·lisegons). | Ràpid (nanosegons). |

Taula 4.1: Comparativa entre xarxes neuronals biològiques i artificials

## 4.6 Estructura d'una xarxa neuronal

Una xarxa neurnal combina diverses capes de procesament i utilitza elements simples que operen en paral·lel, simulen i estan inspirades en els sistemes nerviosos biològics com hem explican en l'apartat 4.5. Consta d'una capa d'entrada, seguit d'una o varies capes ocultes i finalment una capa de sortida. Les capes estan interconectades mitjançant nodes o neurones; cada capa utilitza la sortida de la capa anterior com a entrada.

- **Capa d'entrada:**La capa d'entrada és la primera capa que reb directament la informació d'entrada que es prcessarà. Aquesta capa no realitzarà càlculs complexos, simplement transmet les dades a les capes seguents per fer el processament.
- **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. Aquestes capes són els que realitzan els càlculs i permet que la xarxa aprengui relacions no lineals i representacions abstracts de les dades, això és molt important per fer tasques complexes com el reconeixement de patrons. 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 produïr la predicció 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 clasificació o qualsevol altre resultat. Les neurones d'aquesta capa estan connectades amb totes les neurones de la capa anterior.

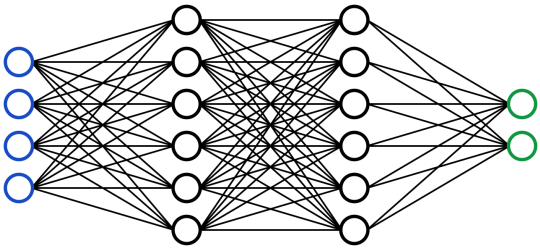

Figura 4.6: Estructura d'una xarxa neuronal

Fons: https://msmk.university/hidden-layer/
Article Linkedin

## 4.7 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, introdueixen la no linealitat i produeixen la sortida de la neurona.

### 4.7.1 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 0 i 1. La seva forma característica és una curva en forma de "S". Si el valor de $x$ que introduïm a la funció és molt gran o fins infinit ($\infty$), llavors $\sigma$ serà 1; en canvi si és molt petit o menys infinit ($-\infty$), $\sigma$ serà 0, i si x = 0, $\sigma$ serà 0,5.

**Avantatges:** La avantatge principal de la funció sigmoide és la seva suavitat i la facilitat de derivació. La funció és diferenciable en tots els punts, cosa que facilita el càlcul de gradients, això vol dir que permet als algoritmes d'optimització treballar amb la funció de manera eficient. A més, la funció és monòtona creixent, ho que significa que

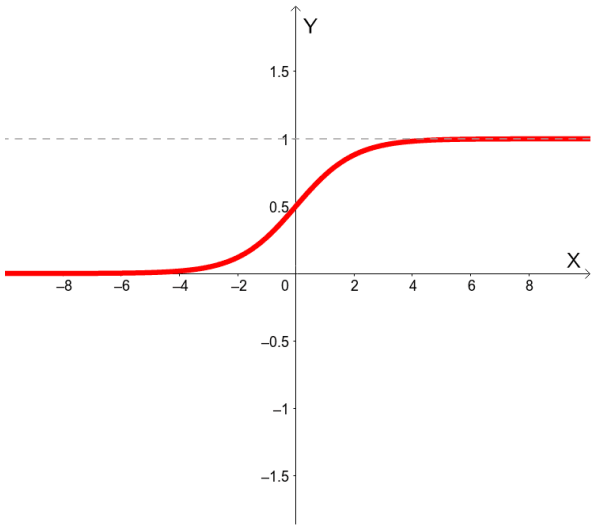


Figura 4.7: Gràfica de la funció sigmoide

una entrada major sempre produirà una sortida major, fent que sigui útil per modelar relacions de causa i efecte.

**Desavantatges:** Tanmateix, la funció sigmoide també té algunes desavantatges. Un dels seus problemes és que la funció es satura els gradients. Quan els valors d'entrades són grans, ho que significa que la derivada de la funció s'apropa a 0 i l'aprenentatge es relentitza. Un altre problema és que aquesta funció no és simètrica, causant a les entrades negatives i positives es processin de mandera diferent, això pot afectaral rendiment de la xarxa. Tot això dificulta el procès d'entrenament de la xarxa en la ràpida minimització de la funció d'error utilitzant l'algoritme de Gradient Descendent.

### 4.7.2 Funció ReLU(Funció Uniat Rectificada Uniforme)

La funció Unitat Rectificada Uniforme té la fòrmula seguent:

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

Aquesta funció té l'algoritme seguent: Si el valor d'entrada es menor que 0, mostra 0, si el valor d'entrada es major o igual que 0, mostrarà el valor d'entrada. Això vol dir que la funció és lineal si la entrada és més gran que 0 perquè la pendent és 1. Encara que la funció ReLU és lineal per a la mitat del seu espai d'entrada, tècnicament és una funció no lineal perquè té un punt no diferenciable en x = 0, on canvia bruscament respecte a x. Aquesta no linealitat permet a les xarxes neuronals apendre patrons complexos.

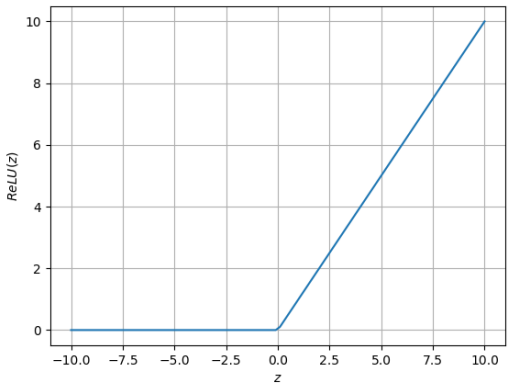


Figura 4.8: Gràfica de la funció ReLU

Tot i això, aquesta funció té una desavanantatge; si utilitzen la funció ReLU com a funció d'activació pasarà una cosa, i es que tots els valors negatius són 0, per tant en el procès de retropropagació, explicat a l'apartat 4.9, no es produeix els valors d'ajust en les neurones negatives. Per solucionar aquest problema s'ha inventat una nova funció que es diu Leaky ReLU. Funciona igual com l funció ReLU, però té un valor determinat per les neurones negatives.

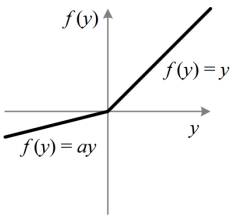


Figura 4.9: Gràfica de la funció Leaky ReLU

### 4.7.3 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 clasificació multiclase. Aquesta funció opera sobre un vector que representa les previsions de cada clase, calculades per les capes anterior.

Per a un vector d'entrada x amb elements x1, x2,..., xC, la funció Softmax es defineix

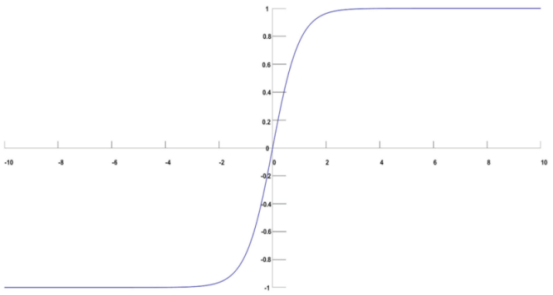

Figura 4.10: Gràfica de la funció Softmax

com:

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

El resultat de la funció Softmax és una distribució de probabilitat de la quàl la suma és 1. Cada element del resultat representa la probailitat de que l'entrada pertanyi a una clase determinada. L'ús d'aquesta funció garantitza de que tots els valors de la sortida siguin positius. Això és molt important perquè les porbabilitats no poden ser negatives.

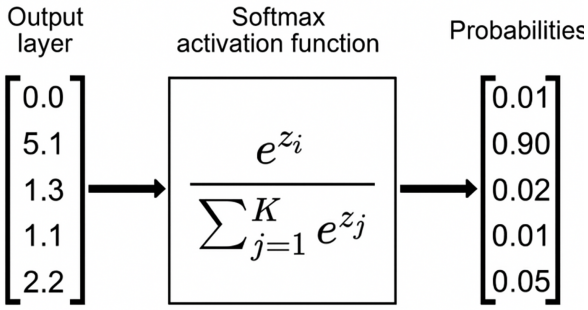


Figura 4.11: Representació de la funció Softmax

Fons: https://msmk.university/hidden-layer/

https://jacar.es/la-funcion-sigmoide-una-herramienta-clave-en-redes-neuronales/

## 4.8 Com funciona una xarxa neuronal?

Ara que sabem quina estructura forma una xarxa neuronal artificial, toca entendre com funciona. Les neurones o nodes son els pilars més importants d'una xarxa neuronal. Cada neurona utilitza la entrada, la processa fent una suma ponderada entre els pesos i les entrades, després una funció d'activació, i pasa la sortida a altres neurones.

Les conexions(pesos i biaixos) son la força de conexió entre dos neurones representades per un pes. Els pesos: Son valors que determinen cuánta influència té la producció d'una neurona sobre un altre, marca la importància que té cada neurona. Els biaixos: Són paràmetres adicionals que ajuda a ajustar els valors de les neurones. És un valor capaç d'apendre com el pes, això vol dir que el model pot utilitzar l'algoritme de retropropagació per a millorar els valors com per exemple els pesos i biaixos. El umbral: Si la sortida de qualsevol node individual és més gran que el valor del umbral, aquell node s'acivarà i envia dades a la seguent capa de la xarxa. En cas contrari, no es pasa ninguna dada a la seguent capa de la xarxa.

Hem de pensar en que cada node individual com el seu propi model de regressió lineal, compost per dades d'entrada, ponderacions, un umbral o un biaix, i una sortida. La fórmula per calcular els valors d'una xarxa neuronal és la seguent:

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

$w_i$: Pesos

$x_i$: Entrades

Per entendre-ho millor, donarem un exemple: Imagina que vols anar a la platja a fer surf i la teva decisió dependrà d'aquestes 3 preguntes:

Hi ha oles?

Hi ha poca gent?

Hi ha tauros?

Analitzarem en detall com funciona un sol node de la xarxa neronal, utilitzant únicament valors binaris (0 y 1) com entrades, 0 voldrà dir "No" y 1 "sí". LLavors suposent el següent cas:

$x_1 = 1$, les oles són bones

$x_2 = 0$, hi ha molta gent a la platja

$x_3 = 1$, Sí que tinc temps

Ara, hem d'asignar algunes ponderacions per determinar la importància. Unes ponderacions majors significan que determinades variabbles son més importants per la decisió.

$w_1 = 5$, hi ha d'haveri moltes oles per poder surfejar

$w_2 = 2$, no t'importa que hi hagi molta gent

$w_3 = 4$, tens pors als taurons

Per últim, ambé suposem un valor umbral de 3, ho que es traduiria en un valor de biaix de -3. Amb totes aquestes entrades, ja podem substituir els valors en la nostre fòrmula per obtenir la sortida.

Fonts: IBM i Amazon Web Services

Les xarxes neuronals operen a través d'un procès de dos pasos: La propagació directa i la retropropagació.

### 4.8.1 Propagació cap a davant

Durant la propagació cap a davan, les dades ingresen en la xarxa a través de la capa d'entrada i flueixen secuencialment a través de les capes ocultes fins a la capa de sortida. En cada neurona, els valors d'entrada del model es multiplican per els seus pesos corresponents i es sumen. Aquesta suma ponderada es pasa a través d'una funció d'activació, explicada prèviament a l'apartat 4.7. Aquest procès continua capa per capa, això acaba conduïnt cap a la predicció final en la capa de sortida.

Aquest procès és important per les seguents raons:

- **Base per l'aprenentatge:** No es pot comprendre com aprenen les xarxes neu-

ronals sense primer entendre com fan prediccions. La programació cap a davan és el requisit prèvi que s'ha de coneixer per comprendre la **??**, l'algoritme que permet l'aprenentatge.

- **Optimització:** En el cas de que una xarxa neuronal no funcioni bé, saber com flueixen les dades per la xarxa t'ajudarà a identificar i solucionar problemes.

- **Diseny del model:** Un diseny eficaç de la xarxa requereix comprendre com es distribueix la informació a través de les configuracions de capes.

## 4.9 Retropropagació en les xarxes neuronals

Mentre que la programació directa fa prediccions, la retroporgramació és la forma en que la xarxa apren d'errors. Implica comparar la predicció de la xarxa amb el valor objectiu real i calcular un terme d'eror mitjançant una funció d'error. Aquest error es propaga enrere a través de la xarxa, començant des de la capa de sortida. Durant aquest procès, 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 procès iteratiu de càlcul d'erros i ajustament de pes permet a la xarxa d'aprenentatge profund millorar gradualment les seves prediccions. El procès de retropropagació es basa en el principi d'optimització del gradient descendent, explicat anteriorment en l'apartat 4.4.2, 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 (4.9.3) per propagar l'error cap enderrere desde la sortida fins l'entrada.

### 4.9.1 Com funcona la retropropagació?

El funcinament de la retropropagació el podem dividir en 5 fases. Cada un d'es possibilita a la xarxa neuronal apendre de manera eficient a partir de dades proporcionats. Les fases són les seguents:

- **Propagació cap endevant:** Aquesta fase inicial és on s'introdueixen les dades de prova en la xarxa neuronal desde la capa d'entrada fins a la sortida.

- **Càlcul d'error:** Una vegada s'obté la sortida de la xarxa, es compara amb el valor desitjat mitjantçant una funció d'error. Aquesta funció quantifica la discrepança que s'ha produït entre la predicció i el valor real.
- **Retropropagació de l'error:** En aquesta fase els gradients de l'error es calculan respecte a cada paràmetre en la xarxa. Com em dit abans, aquest procès comença de manera inversa a la propagació, desde la sortida fins l'entrada.
- **Actualització dels paràmetres:** Una vegada que s'ha calculat els gradients de l'error en tots els paràmetres, aquestos s'actualitzan. D'aquesta manera, els errors en la predicció es minimitzan 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 proba. Després de tot això, es busca garantizar els resultats esperats.

### 4.9.2 Avantatges de la retropropagació

La retropropagació ofereix unes avantatges que l'han convertit en una eina fonamental en l'entrenament de les xarxes neuronals. Les seves avantatges són les seguents:

- **Eficiència en l'optimització de parámetres:** Gracies al càlcul de gradient mitjantçant la regla de la cadena, explicada a l'apartat 4.9.3, la retropropagació facilita l'ajust dels paràmetres d'una xarxa neuronal de manera eficient. Això s'aconsegueix determinant la contribució de cada variable al error total de la xarxa, ho que augmenta notablement a l'hora de fer prediccions precises.
- **Flexibilitat en l'arquitectura:** La retropropagació és compatible en moltes varietats d'arquitectura en les xarxes neuronals, com l'aprenentatge profund amb múltiples capes ocultes. Aquesta característica permet disenys que s'adapten millor a la complexitat de dades i les necesitats de cada tasca.
- **Escalabilitat:** Es posible augmentar la retropropagació perquè traballi amb moltes dades i xarxes neuronas molt grans, ja que hi haurà vegades en que una xarxa pot tindre fins a milions de paràmetres, això exigeix una gestió eficient.

- **Optimització continua:** La retropropagació ofereix la actualització contínua de les variables durant l'entrenament. És un ajust dinàmic important per millorar poc a poc el rendiment de la xarxa amb cada iteració i nous cicles d'aprenentatges.

Fonts: Universitat internacional de València i Universitat de ciències i tecnologia

### 4.9.3 La regla de la cadena

En l'aprenentatge automàtic, sovint es treballa amb funcions compostes complexes que depenen de diversos paràmetres i capes. Per tal de facilitar el càlcul dels gradients i optimitzar el rendiment computacional, s'utilitza la **regla de la cadena**.
La regla de la cadena és una tècnica de derivació matemàtica que permet descompondre la derivada d'una funció composta en el producte de derivades de les seves funcions internes. D'aquesta manera, es poden calcular les derivades de manera escalonada i eficient, especialment en el context de l'algorisme de propagació enrere (backpropagation).
Per exemple, si tenim una funció composta de la forma:

$$L = f(g(h(x)))$$

La derivada de $L$ respecte a $x$ s'obté aplicant la regla de la cadena:

$$\frac{dL}{dx} = \frac{df}{dg} \cdot \frac{dg}{dh} \cdot \frac{dh}{dx}$$

Aquest procediment és fonamental per entrenar xarxes neuronals, ja que permet calcular com cada pes de la xarxa contribueix a l'error global, i així ajustar-los mitjançant tècniques com el descens de gradient.
fonts:Medium deep machine-learning  Greek machine-learning

# 5. Metodologia

Després d'haver fet la recerca previa, i d'haver decidit fer una xarxa neuronal "model" que compleixi tots els requesits que esperavem, ara cal pensar amb quina metodologia podrem aconseguir el nostre objectiu. Per tant vol dir que hem de fer un altre recerca en el que utilizarem les millors eines per fer el Treball de Recerca de manera ràpida i eficient.

Primer ens em fet la pregunta de sobre quins àmbits hem de recorre, una vegada feta la pregunta ja buscarem les millors eines o mètode que ens ajudarà en cadascún dels àmbits. A continuació farem unes presentacions dels àmbits que vam tractar i les diferents solucions que vam donar i el definitiu.

Per començar tractem l'apartat 5.1, que consisteix en com vam estructurar la via de comunicació entre nosaltres i el tutor. A l'apartat 5.2, tracta de quin editor de text vam escollir, que es un punt crucial per el treball ja que les eines que ens aporta l'editor en si mateix ens pot estalviar molt de temps i convertirlo visualment més accessible. A continuació hi es l'apartat 5.3, on t'explica la plataforma que hem escollit per l'entorn col·laboratiu. Finalment per poder crear una xarxa neuronal hem d'utilizar un llenguatge de programació i això s'explicarà a l'apartat 5.4.

## 5.1 Comunicació

Com que som dos persones fent el treball amb un tutor, ens és fonamental tenir una bona via comunicativa. Per tant, hem acollit diferents maneres de comunicació. Ara bé no em pogut registrar tots els registres que hem tingut però la gran majoria s'ha pogut conservar.

D'aquí cap avall explicarem les diferents maneres de comunicació utilizades en l'elaboracio d'aquest treball:

### 5.1.1 Full de Calculs

Hem realitzat diverses reunions amb el tutor, i la via que vam utilitzar per configurar l'horari de les quedades és a través d'un full de càlcul. En aquestes quedades ens em organitzat el treball i ens em quedat amb els diferents mètodes que utilizariem per el trebal, el tutor ens arregla qualsevol tipus de problema, i la instalació dels programes necessaris.

### 5.1.2 Correu Electrònic

L'eina que vam utilizar a distància és el servei de correu electrònic Gmail. Vam fer ús d'aquest aparell per el fet de que és simple i eficient, i té una gran memòria d'emmagatzament per conservar tots els correus que ens hem fet. Les converses fetes per el correu electrònic ha sigut la principal via de comunicació que em establert, per la facilitat i l'eficiència que ens aporta. Per tant, el tema ensenyament i dubtes simples o feiem tot per gmail.

### 5.1.3 Git

També amb l'ajuda del Git, un software que ens va ajudar a guardar tots els canvis que hem anat fent durant el treball de recerca. Vam establir una comunicació diària amb el progrés del treball, deixant comentaris dels canvis i adaptacions que vam anar fet.

## 5.2 Editor de text i processador de text

Per aconseguir una bona presentació de TR, haviem d'escollir o un editor de text o un processador. Tant l'editor com el processador són per crear documents, pero cadascún te les seves funcions.

L'editor de text, és tal com el seu nom, serveix per editar els documents. És molt simple i això genera inconvenients, com per exemple no poder utilizar formats avançat o una personificació. Però, les avantatge d'aquest és que pots tenir la capacitat de navegar rapidament de fitxer en fitxer, facilita la codificació i té compabilitat amb tots

els tipus de llenguatge. És l'ideal per programadors.
Un processador de text són tipus de programes més elaborats i més complexos. Com que són programes amb tanta complexibilitat, et dona accès a formats avançat, utilitzat per a documents professionals com currículums, llibres i d'altres més, també et donen tot tipus d'opcions col·laboratives com disseny de pàgina, correció gràfica i molt més. Per a la part teòrica del nostre projecte sobre la Intel·ligència Artificial, vam seleccionar el format PDF degut a:

1. Connexió segura i estable: Permet l'accés sense la dependència d'una connexió a Internet constant.

2. Baixos requisits de recursos: Pot obrir-se en gairebé qualsevol dispositiu sense necessitat de programes específiques.

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 evita problemes de compatibilitat entre fórmules matemàtiques, imatges i el format del document. En la nostra recerca d'eines que satisfacin tots els requisits, vam identificar dos tipus principals de processadors: Una que es deia WYSIWYG (What You See Is What You Get) i l'altre el WYSIWYM (What You See Is What You Mean). Ara donarem una explicació de les advantatges i desavantatges que tenen:

### 5.2.1 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 la necessitat de coneixer el codi o llenguatges de marcatges. Aquests tipus d'editors poden ser tant com el microsoft word o Google drive.
**Característiques**:

- **Interfície visual:** Els canvis de format (negretes, taules, imatges) es fan mitjançant eines gràfiques, no mitjançant codi.

- **Generació automàtica de codi:** El programa crea el codi subjacent (HTML, CSS, etc.) sense que l'usuari hi intervingui directament.
- **Ús generalitzat:** S'utilitza en àmbits com l'edició web, el disseny gràfic i la publicació digital.

Avantatges:

- **Edició visual intuïtiva:** Permet ajustar el format directament (arrossegar imatges, canviar fonts amb clics).
- Ideal per a usuaris sense coneixements tècnics.
- **Resultat immediat:** No cal compilar, els canvis es veuen al moment.
- **Eines integrades:** Funcions com correcció ortogràfica, taules gràfiques, o opcions de disseny accessibles des del menú
- **Millor per a maquetació complexa:** Documents amb molts elements gràfics (pòsters, fulls informatius)

Desavantatges:

- **Poca precisió en contingut tècnic:** Fórmules matemàtiques o codis es poden deformar-se a l'hora de canviar el format.
- **Errors en documents llargs:** El format manual pot provocar errors (salt de pàgina, numeració desorganitzada).
- **Dependència del programa:** Si s'obre en un altre software, el disseny pot variar.
- **Control de versions complicat:** Difícil treballar en equip sense conflictes de format.

### 5.2.2 What You See Is What You Mean(WYSIWYM)

WSYSIWYM, es refereix a un tipus de processador que es centra en l'estructura del contigut, no en la seva aparença visual immediata, l'usuari marca el text segons la seva

funció (títol, secció, cita i d'altres més) i el format final s'aplica mitjançant un full d'estil com css o Latex. Avantatges:

- **Precisió en elements tècnics:** Fórmules matemàtiques, codi o referències es gestionen amb sintaxi clara.
- **Consistència automàtica:** L'estil s'aplica globalment.
- **Lleuger i portable:** Els fitxers són de text pla.
- **Ideal per a treball en entorns col·laboratius:** Es pot actualizar amb git i fusionar sense haver de canviar el format.

Desavantatges:

- **Corba d'aprenentatge:** Cal aprendre una sintaxi específica.
- **Previsualització no immediata:** En LaTeX, cal compilar; en Markdown, cal un renderitzador.
- **Limitacions en disseny gràfic:** Personalització avançada (ex: posicionament exacte d'imatges) requereix codi addicional.

| **Aspecte** | **WYSIWYG** | **WYSIWYM** |
|---|---|---|
| **Enfocament** | Aparença visual immediata | Estructura semàntica del contingut |
| **Exemples** | Microsoft Word, WordPress | LaTeX, LyX |
| **Usuaris** | No tècnics, dissenyadors | Acadèmics, desenvolupadors tècnics |
| **Control** | Limitada (codi generat automàtic) | Alt (definició manual de l'estructura) |

Taula 5.1: Comparativa entre WYSIWYG i WYSIWYM

font(WYSIWYG): (Wikipedia Arimetrics Wix). font(WYSIWYM): (Wikipedia Ryte WIKI).

### 5.2.3 LaTeX

LaTeX, és un tipus de WYSIWYG, on s'enfoca en l'estructura lògica, no la visual, tal com vam mencionar en l'apartat 5.2.2. LaTeX és un sistema de composició tipogràfica

d'alta qualitat, especialment dissenyat per a la creació de documents acadèmics, tècnics i científics. A diferència dels processadors de text tradicionals com Microsoft Word, LaTeX es basa en codi per estructurar el document, separant el contingut de la seva presentació visual. Va ser desenvolupat per Leslie Lamport en 1984 com un conjunt de macros per al sistema TeX de Donald Knuth. La raó principal de la seva creació va ser la impotència dels processadors de text de l'època per compilar fórmules matemàtiques avançades i gestionar documents complexos amb precisió tipogràfica. És per aquesta raó la qual el vam escollir com a processador per satisfer els requeriments que demanem. Per més informació o dubte podeu consultar el manual de latex: Manual de LaTeX

### 5.2.4 Kile

Kile es un editor especific del LaTeXque funciona en els sistemes operatius tant Linux, Windows com Apple Macintosh. L'editor va ser desenvolupat per la comunitat KDE en que ofereix diverses eines avannçades per a l'edicio de LaTeX. No és l'únic editor que existeix, però és el que més ens convé per el fet de que sigui creat per documents acadèmics i científics.

Avantatges:

- **Entorn integrat complet:** Kile ofereix tot ho necessari per treballar amb LaTeX en una sola interfície, incloent editor, compilador, visualitzador i gestor de referències
- **Altament configurable:** Permet afegir botons a la barra d'eines per executar scripts personalitzats i automatitzar tasques complexes.
- **Feedback WYSWYG:** Et dona una sensació quasi WYSWYG, gracies als visualitzadors que els té integrades en que pots veure els canvis en temps reals.
- **Eines avançades d'edició:** Té autocompletador de comandes, accés ràpid a símbols matemàtics, navegacions ràpides per seccions.

Desavantatges:

- **Nivell d'apranentatge:** Costa molt d'aprendre l'interfície si no estas familialitzat amb KDE.

- **Configuració inicial:** Requereix molts tipus de configuracions externs per tal d'arrenca.

Ara mostrarem una taula de comparació respecte els altres editors: LyX és més adequat per a principiants gràcies a la seva interfície WYSIWYG. No obstant, Kile es més flexible per als usuaris avançats i te més control en els codis i funcions.

| LaTeX | **Kile** | **LyX** |
|---|---|---|
| **Facilitat d'ús** | | X |
| **Control i Flexibilitat** | X | |
| **Visualització en temps real** | | X |

TeXmaker i TeXstudio comparteixen moltes similituds amb Kile en termes de funcionalitat i d'interfície.

| LaTeX | **Kile** | **TeXmaker i TeXstudio** |
|---|---|---|
| **similitud de funcionalitat** | - | - |
| **Integració KDE** | X | |
| **Disponibilitat dels sistemes operatius** | X | |

Gummi és un editor molt més simple que Kile, amb previsualització en temps real però amb moltes menys funcionalitats. És bo per a documents simples però per a projectes complexos no.

| LaTeX | **Kile** | **Gummi** |
|---|---|---|
| **complexibilitat:** | X | |
| **Previsualització** | | X |
| **funcionalitat** | X | |

Per més informació en qüestió de comparació podeu fer una consulta a les següents pagines: 5 editors de LaTeX Analisí detallada de Kile i d'altres editors Fonts: Experiencia Personal de l'us Kile Descripcio de Característiques Kile

### 5.2.5 KDE

El nom "KDE"originalment sí que era un acrònim de "K Desktop Environment"(des de la seva creació el 1996), però a partir del 2009, la comunitat va decidir deixar de considerar-lo un acrònim i utilitzar-lo simplement com un nom propi per representar:

- L'entorn d'escriptori.
- La comunitat global que desenvolupa programari lliure.
- Tots els projectes relacionats.

Font:Wikipedia

#### Programari lliure

El programari (o software en anglès) és el conjunt de programes informàtics, procediments i documentació que permeten a un ordinador realitzar tasques específiques. És la part interna d'una computadora ,a difèrencia del hardware que és l'extern, el software és part física.
Un programari es considera Free software quan respecta les quatres llibertats fonamentals definides per la Free Software Foundation (FSF):

1. **Llibertat 0:** Executar el programa amb qualsevol propòsit.
2. **Llibertat 1:** Estudiar i modificar el codi font (requereix accés al codi)
3. **Llibertat 2:** Redistribuir còpies per ajudar altres usuaris.
4. **Llibertat 3:** Millorar el programa i compartir les modificacions.

Mapa conceptual del Free software:

Font:Free Software Foundation

**L'entorn d'escriptori**

Un entorn d'escriptori en sistemes operatius és la interfície gràfica que permet als usuaris interactuar amb l'ordinador. Inclou elements com finestres, icones, panells, fons de pantalla i eines de gestió d'arxius.A continuació, mostrarem els principals entorns que n'hi ha:

1. KDE Plasma

    - **Característiques:** Altament personalitzable, amb temes globals, widgets i configuració avançada.
    - **Recomanat per:** Usuaris que volen control total sobre l'aparença i el flux de treball.

2. GNOME

    - **Característiques:** Disseny minimalista i enfocat en la productivitat. Compatible amb extensions per ampliar funcionalitats.
    - **Recomanat per:** Usuaris que prefereixen una experiència neta i senzilla.

3. XFCE

    - **Característiques:** Lleuger i ràpid, ideal per a maquinari antic o limitat.
    - **Recomanat per:** Usuaris que busquen eficiència sense sacrificar funcionalitats bàsiques .

4. Cinnamon

    - **Característiques:** Interfície tradicional similar a Windows, amb personalització mitjana.
    - **Recomanat per:** Usuaris que provenen de Windows i busquen una transició suau.

Fonts:Dev Planet

## 5.3 Entorn col·laboratiu

A l'hora de crear un projecte en equip, pot arribar a ser complicat treballar amb 2 o més membres del grup a la vegada en els seus respectius dispositius si no s'escull una eina col·laborativa adequada, sobretot si s'està treballant amb molts fitxers. Pot haver problemes alhora de intercanviar o que sol pugui estar un membre traballant en un fitxer fa que sigui un absolutament caos.

### 5.3.1 Google Drive

Google Drive és una plataforma de Google que ofereix eines gratuïtes com un processador de text (Docs) i un entorn col·laboratiu. Tot i ser senzill i eficient, té algunes limitacions importants que hem de considerar:

- **Poc control en el procés col·laboratiu:** És difícil gestionar canvis simultanis o conflicts entre versions.
- **Limitacions en el control de versions:** No permet controlar detalladament els canvis que es fan i registrar.
- **Falta de privadesa:** Google té accés als teus arxius, cosa que pot ser un problema per a dades sensibles.

Aquestes limitacions fan que Google Drive no sigui l'opció ideal per el nostre TR la qual cosa ens fara escollir l'altre alternativa que teniem, el Git+GitHub+Vim.

### 5.3.2 Git+GitHub+Vim

Git és un sistema de control de versions que ofereix:

- Historial dels canvis.
- Branques per desenvolupar diferents funcionalitat sense afectar la versió principal.
- Control total dels canvis
- Control i revisió de codi

El principal desavantatge que té el Git és que, per si sol, no està optimitzat per a la col·laboració en equips grans. Pero aquí entra en acció el GitHub, que complementa Git amb:

- **Repositoris**
- **Git pull,commit,push:** Dona la possibilitat de poder discutir les versions abans de pujarlo a la original.
- Solucions als conflictes de versions amb l'ajuda del Vim.

**Avantatges:**

- **Copia de seguretat:** Cada usuari té una copia de seguretat del servidor principal, en el cas d'errors o corrupció, sempre el pots reemplaçar-lo per el servidor principal.
- **Garantia de dades:** El model de dades que utilitza Git garantitza la integritat ciptográfica de cada bit del teu projecte. Cada arxiu es recupera mitjançant la seva suma de comprobació quan es torna a desgarregar-se. Es imposible obtenir res de Git que no sigui els bits exactes que s'ha integrat.
- **Àrea de preparació:** Una cosa que diferència a Git d'altres eines es que es posible preparar alguns dels teus arxius i confirmar-los sense confirmar tots els altres canvis que s'han fet en els altres arxius, això et permet preparar només algunes parts d'un arxiu modificat i es realitza amb el commandament git commit "nom de l'arxiu". Git també et dona la opció d'ignorar aquesta característica i confirmar tots els canvis que s'han fet, afegint un "-a" al final del commandament.
- **Eficiència de Git:** Git no depèn d'internet per la majoria de les operacions, totes les accions principals (veure l'historial, fer commits) es fan en localment. Git està programat en llenguatge C, això permet a Git realitzar les tasques amb molta rapidesa encara que es treballi amb molts fichers.

Més informació en: Pagina oficial del Git

**Desavantatges:**

- **Corba d'aprenentatge pronunciada**: Git té un model conceptual complex (working directory, staging area, repositori local/remot) que requereix temps per assimilar. Operacions bàsiques com merge/resolucions de conflictes poden resultar difícils per a principiants.

  1. **Working directory:**
     - La carpeta local del projecte on treballes i fas canvis als fitxers.
     - Modifiques, elimines o crees nous arxius
     - Git analiza aquest aquest directori per fer la comparació dels canvis respecte l'ultima versió
  2. **Staging Area:**
     - És el cache del Git
     - Una zona intermedia on inclous els canvis propers que vols fer
  3. **Repositori Local:**
     - És la base de dades completa de Git que es troba a la teva màquina (dins de la carpeta .git)
     - Conté tot l'historial de commits, branques, tags del Git
     - Els canvis del Staging Area, es guarden permanent aqui
  4. **Repositori Remot:**
     - És una copia del repositori en un servidor (GitHub, GitLab, Bitbucket)
     - Serveix per compartir codis i funciona com a entorn col·laboratiu
     - És utilizen comandes git push(Pujar canvis) i git pull(Baixar canvis)

- **Problemes amb fitxers molt grans**: Git no està optimitzat per a fitxers binaries de gran mida (vídeos, imatges, bases de dades). El rendiment es degrada significativament i el repositori pot inflar-se massa.

- **Gestió complicada de credencials**: Git no gestiona de forma òptima les credencials.

- **Problemes de rendiment en repositoris molt grans**: En repositoris amb milers de commits o arxius, certes operacions (com git status o git checkout) poden tornar-se lentes.

- **Comandes**: Els comandes reben nom contraintuïtiu io la sintaxi varia segons l'operació.

Fonts:Comparacions entre control de versions bloc de critica d'un desenvolupador

**Vim**

Vim (Vi IMproved) és un editor de text avançat, lliure i de codi obert, basat en l'editor clàssic vi. És conegut per ser multiplataforma i també la seva eficiència, personalització i ús sense ratoli, sent popular entre programadors i usuaris tècnics. Aquest editor de text consta de 4 modes amb diferents funcionalitats:

- **Mode Estandard:** Per navegar i executar ordres
- **Mode Insert:** Per escriure text
- **Mode Visual:** Per seleccionar bloc de text
- **Mode Command** Per executar comandes complexes

Fonts:Pagina oficial del Vim Documentacio del Vim

## 5.4 Llenguatge de programació

El llenguatge de programació que vam decidir per fer la part pràctia del treball va ser Python. Python és un llenguatge de programació que va ser creat en 1989 per Guido van Rossum , un informàtic neerlandès. Aquest llenguatge és conegut per la seva simple sintaxi, comparada amb altres llenguatges com JavaScript o C++ que són més complicats, python facilita l'aprenentatge pels usuaris que comencen a apendre a programar desde 0.

### 5.4.1 Python

La raó principal per la que vam escollir aquest llenguatgera per les avantatges seguents:
**Avantatges**

- **No cal compilar:** No és necesari compilar en Python, un cop que tens el codi, s'executa directament.

- **Llibreries** Python té una gran quantitat de llibreries molt útils com "TensorFlow" o numpy, aquestes llibreries estalvien moltes línies de codi a l'hora de programar una xarxa neuronal.
- **Identificació d'errors** Encara que altres llenguatges també poden identificar-te els erros, Python t'ho explica de manera més clara i precisa, estalviant-te el temps d'anar línea per línea buscant l'error.
- **Sintaxi:**La seva sintaxi clara i llegible que està explicada a l'apartat **??** és ho que fa que aquest llenguatge sigui fàcil d'apendre quan un comença a programar desde 0.

Malgrat totes aquestes avantatges, Python també té unes porques desaventatges que hem de tenir present.

**Desavantatges:**

- **Limitacions de rendiment:** La naturalesa interpretada de Python significa que és més lent que altres llenguatges. La causa d'això és que Python es tradueix a codi màquina línea per línea durant l'execució del codi, cosa que afegeix un temps de processament.
- **Consum de memoria:** Gràcies a la bona fexibilitat d'aquest llenguate fa que l'ordinador requereixi un major consum de memòria, un consum excesiu de la memoria pot causar relentització en el programa i bloquejar-lo. Per tant es recomendable tenir un ordinador o portatil decent que pugui executar correctament Python.

### 5.4.2 Sintaxi

Acabem d'explicar que la sintaxi de Python és simple, però què diferencia aquesta sintaxi d'altres llenguatges? La principal raó és que el seu format de codi és visualment ordenat i utilitza paraules clau de l'anglès. A diferència d'altres llenguatges, no utilitza claudàtors per determinar blocs, com per exemple les sentències "if, else":

En aquesta taula podem veurem la diferència de sintaxi que té cada llenguatge, en aquest cas podem apreciar que C és mès llarg que python i que requereis més commandaments i línies de codi. A continuació mostrarem una taula resumida d'aquestes

| **Condicions en C** | **Condicions en Python** |
|---|---|
| `#include <stdio.h>`<br><br>`int main() {`<br>`  int numero = 5;`<br><br>`  if (numero > 0) {`<br>`  printf("És >0.\n")`<br>`  ;`<br>`  } else {`<br>`  printf("És <0.\n")`<br>`  ;`<br>`  }`<br>`  return 0;`<br>`}` | `numero = 5`<br><br>`if numero > 0:`<br>`  print("És positiu.")`<br>`else:`<br>`  print("És negatiu.")` |

Taula 5.2: Comparació de condicions en C i Python

diferències.

**Resum de diferències:**

| **Concepte** | **C** | **Python** |
|---|---|---|
| Terminador de línia | ; obligatori | No és obligatori ; |
| Agrupació de blocs | { } | Espai en blanc |
| Condició | `if (condició)` | `if condició:` |
| Estructura | Requereix el script `main()` | Script directe |
| Librerías | `#include <stdio.h>` | No requereix per `print()` |

Taula 5.3: Resum de les diferències

### 5.4.3 Editor de Text

Per començar a programar, és molt important escollir bé l'editor de text. Per raons tècniques, ara farem una llistat de quines opcions teníem i quina vam escollir finalment.

**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 en 2025. Es caracteritza per ser multiplataforma, lleuger, altament personalitzable i suporta tots els llenguatges de programació gràcies a les seves extensions. Caracteristiques:

- És caracteritza per tenir un control de versions integrat del Git.
- Inclou un depurador de codis pas per pas.
- Interfície intuïtiva modernitzat.

**La nostra experiència utilitzant VS Code:** Vam començar a utilitzar VS Code perquè ja el dominàvem abans del Treball de Recerca (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,el Eclipse.

**Eclipse:** Eclipse és un entorn integrat de desenvolupament (IDE) de codi obert, escrit principalment en Java, però dissenyat per treballar amb múltiples llenguatges de programació. originalment fou creada per international Business Machines (IBM). Actualement cau en mans del Fundació Eclipse. Eclipse te les mateixes característiques que el de VS Code,no obstant, té una funcionalitat de més que es el enfocament en JAVA i ofereix diverses extensions per ell i té una gestio de fitxers grans millors que el de VS Code.

**La nostra experiència utilitzant Eclipse:** Nosaltres el vam escollir per la seva funcionalitat i també per el sugeriment pel nostre tutor, consta d'un rendiment i estabilitatg superior a la de VS Code, de fet te uns extensions més solides que la de VS Code, però al final l'hem hagut de deixar per el fet de que vam decidir treballar amb

Python i no ens convenia massa el seu punt d'enfoc ja que pot funcionar molt bé amb arxius molts grans de JAVA, però d'altres no. Aquesta vegada hem estat fent una gran recerca d'ús d'editor de text i finalment vam trobar el Pycharm.

**Pycharm:** PyCharm, desenvolupat per JetBrains, és l'entorn integrat de desenvolupament (IDE) més popular per a Python en l'actualitat. Empreses com Twitter, Facebook, Amazon i Pinterest el fan servir per al desenvolupament de les seves aplicacions en Python, gràcies a les seves eines avançades i la seva gran adaptabilitat.
Característiques:

- Diverses extensions per Python
- Integració amb frameworks populars
- Suport per a bases de dades
- Depuració avançada

**La nostra experiència utilitzant Pycharm:** Després de fer 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 les avantatges que ens aporta al treball amb Python ens va convecer en utilitzar-ho com a l'editor de text definitiu que treballarem al llarg del treball.
Fonts:VS Code i les seves avantatges Fundació Eclipse Tot l'ho que has de saber del PycharmTotes les novetats de Pycharm (Pagina oficial)

## 5.5 Sistema Operatiu

Els programes i programaris d'alt nivell moltes vegades no són compatibles amb tots tipus de sistemes operatius, o que no tenen el mateix rendiment un respecte a l'altre. Per tant això ens porta a fer una recerca.

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. "

Tal com està escrita en la Universitat Europea, llavors es classifiquen en diverses sistemes operatius elaborada per diferents companya i les més destacades són les següents: Windows Linux macOS

### 5.5.1 Windows

Windows és un sistema operatiu desenvolupat per Microsoft a principis dels anys 80. La seva popularitat va ser tan gran que va arribar a aconseguir fins a un 90% de la quota de mercat en el seu moment més glorios.
Característiques principals:

- Interfície gràfica basada en finestres, que simplifica l'ús i redueix la necessitat de comandes de text.
- Aplicacions integrades amb funcionalitats com la barra de tasques i el menú d'inici.
- Amplia disponibilitat amb programari i facilitat per instal·lar aplicacions.

### 5.5.2 Linux

Linux és un sistema operatiu de codi obert creat per Linus Torvalds el 1991. És conegut per la seva versabilitat.
Característiques principals:

- Llibertat i personalització, gràcies al seu model de codi obert.

- Gran seguretat i estabilitat.
- Gran varietat de recursos, ideal per a ordinadors antics o tasques especialitzades.

### 5.5.3 macOS

macOS és el sistema operatiu exclusiu dels ordinadors Apple, desenvolupat des dels anys 70. Destaca pel seu disseny pulit i la integració amb l'ecosistema de dispositius de la marca. Característiques principals:

- Interfície elegant i intuïtiva
- Optimització per al maquinari d'Apple

**Decisió Final: Windows i Linux**

Hem optat per utilitzar Windows i Linux pels següents motius:

- Linux és ideal per a tasques avançades, terminals i automatitzacions, a més de ser compatible amb el processador Kile.
- Windows ofereix una major facilitat d'ús i compatibilitat amb programari general, sent perfecte per a tasques diàries.

No obstant, en un ordinador rarament hi han dos sistemes operatius convivint, això ens porta ha fer un altre recerca, i n'hem trobat dues:

1. Dual boot: Instal·lar amb dues sistemes en particions separades i triar-ne una en l'arrencada.
2. Màquina virtual: Executar un sistema dins de l'altre.

Finalment, hem escollit la màquina virtual per la seva facilitat d'ús i baixos requisits, encara que el rendiment no sigui tan fluid com en dual boot. Aquesta opció ens permet provar Linux sense afectar la instal·lació principal de Windows.

### 5.5.4 Dos maquines virtual en un sol equip

1. Dual Boot:

- Requereix particionar el disc dur.
- Permet un rendiment complet de cada sistema.
- Menys flexible per canviar entre sistemes sense reiniciar.

2. Maquina virtual:

- Menys eficient en recursos (comparteix CPU, RAM i emmagatzematge).
- Més fàcil de configurar i gestionar.
- Ideal per proves o ús casual sense modificar l'entorn principal.

# 6. Xarxa Neuronal

## 6.1 Introducció

En iniciar la part pràctica del treball vam decidir crear tres tipus de xarxes neuronals: una Xarxa Neuronal amb llenguatge de programació, una altra Xarxa Neuronal amb fulls de calculs i, finalment, una Xarxa Neuronal amb un cas real.
En aquest capítol explicarem com vam elaborar cadascuna d'aquestes xarxes neuronals i, a més, compararem les dues formes de desenvolupament en una Xarxa neuronal de regressió.
Per facilitar l'organització del treball, ens hem repartit les tasques: un membre de l'equip s'ha encarregat de la xarxa neuronal implementada amb llenguatge de programació Python, mentre que l'altre ha treballat una xarxa neuronal feta amb fulls de càlcul. Finalment, hem elaborat conjuntament la taula de comparació, intercanviant perspectives, i també hem treballat plegats la xarxa neuronal del cas real.
Les nostres pràctiques consisteixen en recopilar una sèrie de dades d'alumnes mitjançant una enquesta feta per nosaltres amb la finalitat de crear una Xarxa neuronal que pugues predir la nota final de cada alumne. Les preguntes fetes en el formulari són sobre la matèria de matemàtiques i són les següents:

- Realització de deures
- Hores d'estudis (semanal)
- Hores de section
- Interès en la matèria
- Nota del segon trimestre
- Nota del tercer trimestre
- Nota final

Com es pot veure, les preguntes d'aquest formulari estan relacionades amb el rendiment d'estudis dels alumnes, i s'envia als estudiants que tingues l'assignatura matematiques.

Després de les prediccions, compararem la xarxa neuronal feta amb python y la del full de càlcul i veurem quina de les dos té més precisió.

## 6.2 Xarxa neuronal de regressió

Una intel·ligència artificial és un camp molt extens i, dins d'aquest, les xarxes neuronals també representen una branca àmplia i complexa. En el nostre context utilitzarem un tipus concret de xarxa neuronal per a la pràctica: les **xarxes neuronals de regressió**.

Una xarxa neuronal de regressió, a diferència d'una de classificació, té una sortida de tipus lineal que permet predir un valor numèric continu. Aquest tipus de xarxes requereixen un conjunt de dades prou extens i ben estructurat per poder aprendre les relacions entre les variables d'entrada i generar una estimació fiable.

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

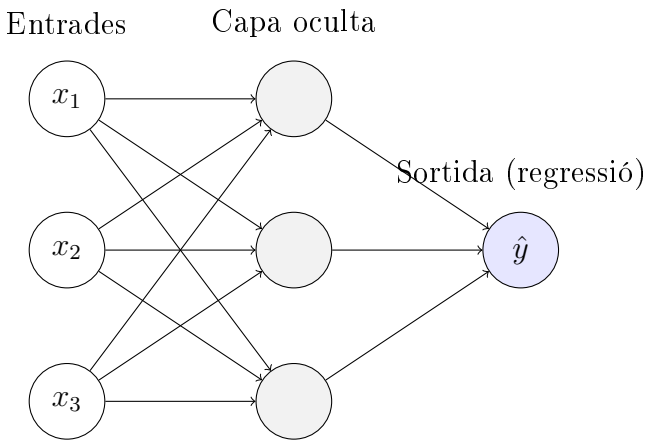


Figura 6.1: Esquema d'una xarxa neuronal de regressió

## 6.3 Xarxa Neuronal amb llenguatge de programació

En aquest apartat explicarem pas a pas de com vaig crear el nostre xarxa neuronal de regressió amb un llenguatge de programació.

### 6.3.1 De celcius a fahrenheit

Abans de començar a treballar amb la xarxa neuronal definitiu, vaig voler fer una de més simple per tal de coneixer amb més profunditat de com funciona una xarxa neuronal de regressió; en aquest cas he escollit un transformador d'unitats de temperatura, de celcius (**Unitat de temperatura basada en el punt de fusió de l'aigua**) a fahrenheit (**Sistema d'unitat basada en el punt d'ebullició i solidificació de l'aigua**). Per crear aquesta xarxa vaig escollir el Python com a llenguatge de programació, i el motiu esta explicat en l'apartat 5.4.

Per començar, importo el framework i la biblioteca de Python que són les bases de dades necessàries per a la creació de la xarxa neuronal: TensorFlow (**framework**) i numpy (**biblioteca**). Com podeu veure, he creat uns *abrevacions* per a poder-los cridar més fàcilment quan els necessiti.

- **TensorFlow:** TensorFlow és un tipus de framework en que ofereix eines per crear i entrenar models de Ml (Machine learning) i Dl (Deep learning), 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, APIs (Application Programming Interface) en diversos llenguatges.
- **numpy:** numpy és una biblioteca científica per treballar amb vectors i matrius.

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

Figura 6.2: Framework i biblioteca del Python

Agafo les paraules **celsius** i **fahrenheit** i les defineixo com a variables amb un signe $=$. Després poso $np$ per indicar la biblioteca, seguit de $.array()$ per especificar que és una llista. Dins dels parèntesis hi poso $[]$ i a dins, els diferents valors de temperatura. Finalment, afegeixo el paràmetre $dtype = float$ per indicar que es tracta de dades numèriques decimals.

Repeteixo el mateix procés amb **fahrenheit**, però en aquest cas les dades no poden ser arbitràries, sinó que han de ser les temperatures corresponents a les de **celsius** convertides a Fahrenheit. D'aquesta manera, la xarxa neuronal pot entendre la relació entre les dues 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 6.3: Agrupació de dades amb numpy

Per començar, defineixo les capes de la xarxa neuronal. Utilitzant $\boxed{tf.keras.layers.Dense}$ (**funció per a capes denses** (Keras és un tipus de API)), creo la primera capa oculta amb el paràmetre $\boxed{units = 3}$ per especificar que tingui **3 neurones**, i $\boxed{input_shape = [1]}$ per indicar que l'entrada és un sol valor. Li assigno el nom $\boxed{oculte_1}$.

Repeteixo el procés per a la segona capa oculta, $\boxed{oculte_2}$, amb $\boxed{units = 3}$ neurones també, però sense especificar $\boxed{input_shape}$ ja que la xarxa ho dedueix automàticament de la capa anterior.

Finalment, defineixo la capa de $\boxed{sortida}$ amb $\boxed{units = 1}$ perquè retorni un sol valor (la predicció en Fahrenheit).

Amb totes les capes creades, les integro en un model seqüencial amb $\boxed{tf.keras.Sequential}$, passant-les dins d'una llista $\boxed{[]}$ en l'ordre correcte: $\boxed{[oculte_1, oculte_2, sortida]}$. Això crea l'estructura bàsica de la nostra 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])S
```

Figura 6.4: Capes ocultes i la sortida d'una xarxa neuronal

Font: (Xarxa neuronal amb Python)

## 6.4 Xarxa Neuronal amb fulls de calculs

En aquest apartat continuarem amb la xarxa neuranal de regressió però aquesta vegada utilizarem un full de calculs per fer-ho. L'estructura que utilitzarem per aquesta

pràctica serà la del perceptró.

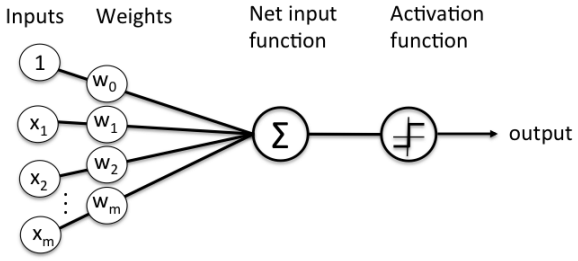


Figura 6.5: Estructura del perceptró

aaa

Un cop sabem quina estructura utilitzarem, 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 6.6: 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, anomenantlos $xi$

- **Realització de deures:** $x1$
- **Hores d'estudis:** $x2$
- **Hores de son:** $x3$
- **Interès en la matèria:** $x4$
- **Notes del segon trimestre:** $x5$
- **Notes del tercer trimestre:** $x6$

- **Nota final:**´ $y$

Aquestra representació queda així:

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 6.7: Taula resumida

L'entrada de "Realització de deures" és una data binaria que només pot prendre valors 0 o 1.

### 6.4.1 Normalització de dades

Abans de continuar, és necessari explicar que és la normalització de dades. La normalització de dades és una tècnica de procesament que consisteix en transformar dades de diferents escales a una escala comú, com per exemple del 0 al 1, això facilita la comparació i l'anàlisi de la xarxa neuronal i millora el seu rendiment. En el nostre cas, tenim dades binaries i dades ordinaries qe poden prendre qualsevol valor, aquest desequilibri afecta els càlculs posteriors si no es solucionen d'alguna manera.
Per aquesta raó, convertirem totes les dades en valors d'entre 0 i 1. Aquest procès implica calcular la miitjana de les dades i la desviació estàndar de cada variable. Per això utilitzant la fòrmula seguent:
$z = \frac{x-\mu}{\sigma}$

On:
$z$ És el valor normalitzat
$x$ És el valor original
$\mu$ És la mitjana
$\sigma$ És la desviació estàndar

Aquests càlculs són fàcils d'obtenir amb les funcions que ens proporciona el full de càlcul. Les dades normalitzades els representarem amb una X.


Figura 6.8: Dades normalitzades

### 6.4.2 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 predicció final. Al començament de l'entrenament, assignarem a tots els valors d'enrada 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.

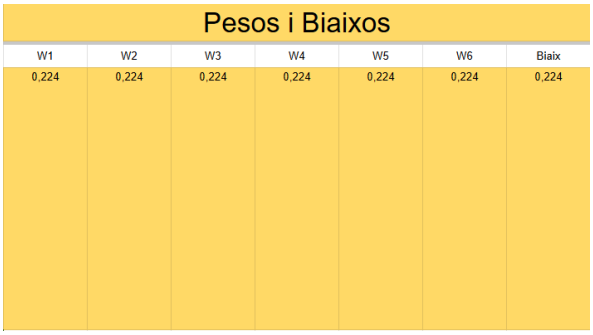


Figura 6.9: Taula dels pesos

En la figura 6.9 podem observar com ha quedat la taula després d'afegir els pesos als valors d'entrada, he afegit 6 columnes de pesos respecte a les 6 entrades i una columna més pel biaix.

### 6.4.3 Funció d'error del model

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

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

Després d'aplicar la fòrmula, obtindrem la predicció de la nota final, tal com es veu en la figura 6.10.

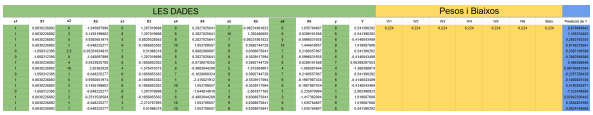


Figura 6.10: Prediccions del model

Hem de recordar que els valors de la predicció són erronis, a que els pesos que hem assignat són aleatoris.

Per questa raò, el següent pas de la pràtcia és entrenar el nostre model per millorar els paràmetres. Per dur això, haurem d'aplicar un procès d'optimització per ajustar aquest paràmetres. En aquest cas utilitzarem l'algoritme gradient descendent, explicat anteriorment en l'apartat 4.4.2.

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

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

Figura 6.11: Erros de la predicció

Es pot veure en la imatge 6.11 dues columnes adicionals, la columna "error" emmagatzema els errors del model i l'altre conté els mateixos errors pero en valor absolut, es 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 aqeustes 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, quan més baix sigui aquest valor, la predicció serà més precisa.

### 6.4.4 Canvis dels paràmetres

Un cop tenim la funció d'error de la primera predicció, 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 tindre en compte les dades seguents: Quan la diferència de $Y$ respecte a $y$ és més gran, més luny 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 l predicció.

Gràcies a la fòrmula 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. Per tant hem de tindre en compte dues coses: La diferènccia de la predicció final ($Y$) i del valor real ($y$) i el valor d'entrada ($X$) respecte al seu pes ($W$). Per tant fòrmula que utilitzarem per calcular els canvis dels paràmetres serà:

## 6.5 Comparació entre una xarxa neurnal creada per un llenguatge de programació netre una de fulls de calculs

## 6.6 Xarxa Neuronal amb un cas real

# 7. Resultats

HelloWorld

# 8. Conclusions

# A. GNU Free Documentation License

Version 1.3, 3 November 2008



<http://fsf.org/>



### 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 non-commercially. 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 noncommercially, 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] 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.

[2] IBM. Pàgina web d'IBM. https://www.ibm.com/history/deep-blue. [Online; consultada el 15/07/2025].

[3] 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.

[4] A. M. TURING. I.-computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950. https://doi.org/10.1093/mind/LIX.236.433.