



Xarxes neuronals

Oscar Galera i Alfaro

Mineria de dades

1er MEINF

Abril 2018

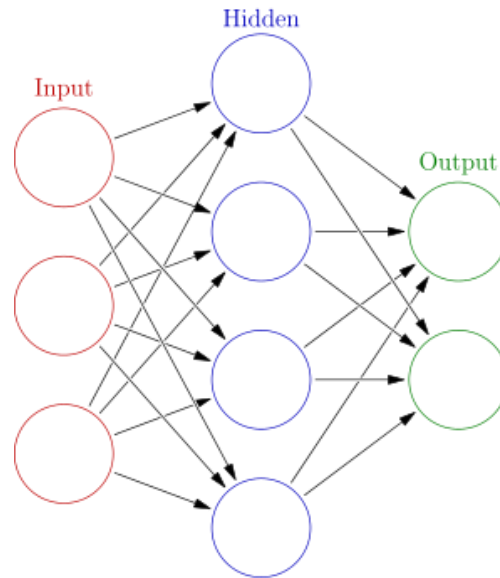
Índex

1. Què és una xarxa neuronal?
2. Què és una neurona?
3. Funcions d'activació
4. Com funcionen?
5. Com aprenen?
6. Descens del gradient
7. Descens del gradient estocàstic
8. *Backpropagation*
9. Fases en l'entrenament d'una xarxa
10. *Demostració*



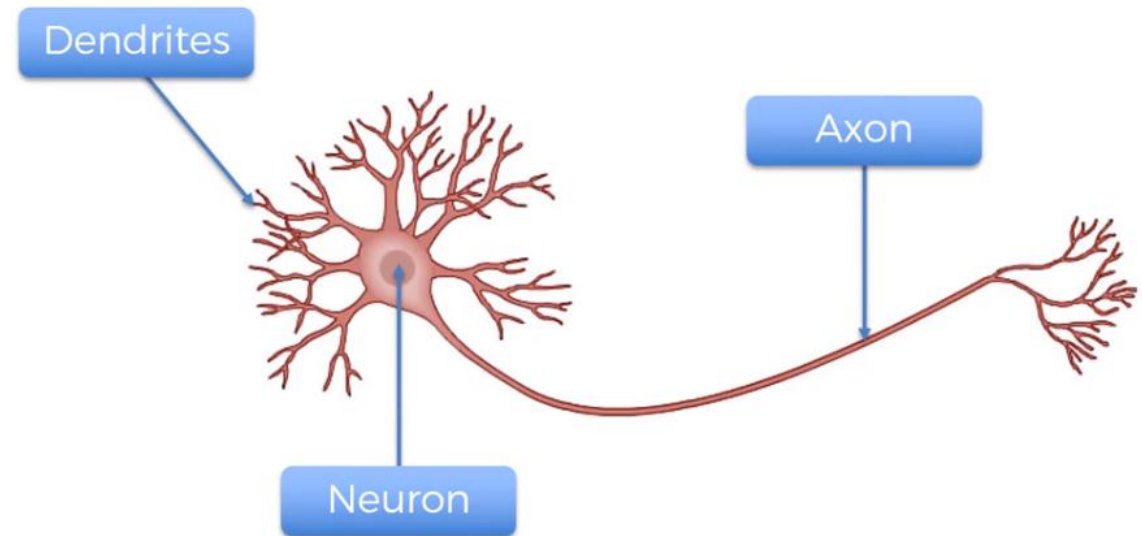
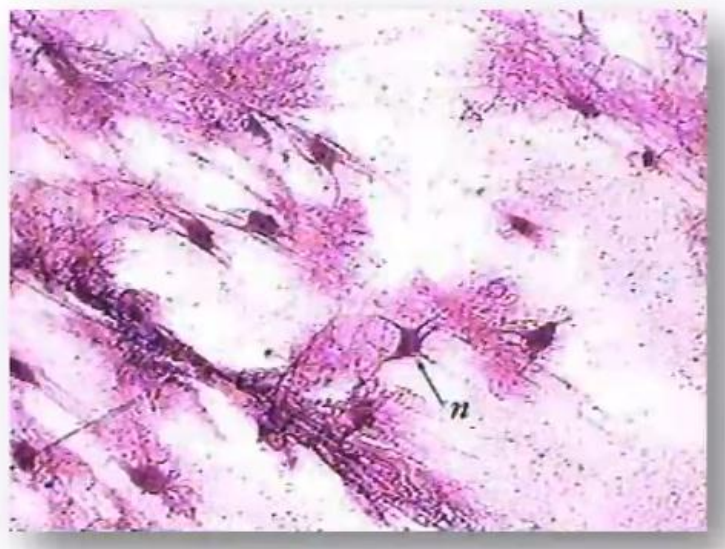
1. Què és una xarxa neuronal?

Una xarxa neuronal està composta per capes de neurones, on sempre hi ha una única capa de neurones d'entrada, n capes de neurones internes (el nombre de capes depèn de la complexitat del problema a resoldre) i una capa de neurones de sortida.



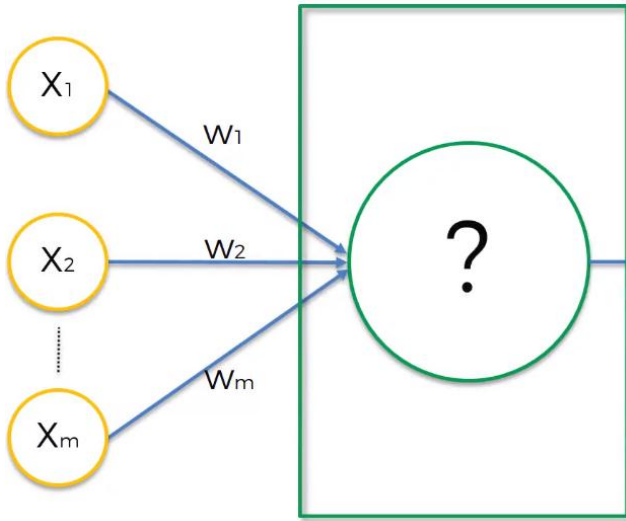
2. Què és una neurona?

Les neurones són els blocs bàsics en què es recolzen les xarxes neuronals. La seva funcionalitat treballant de forma individual no serveix de gaire, però sí que serveix quan treballen de forma estructurada grans quantitats d'aquestes.



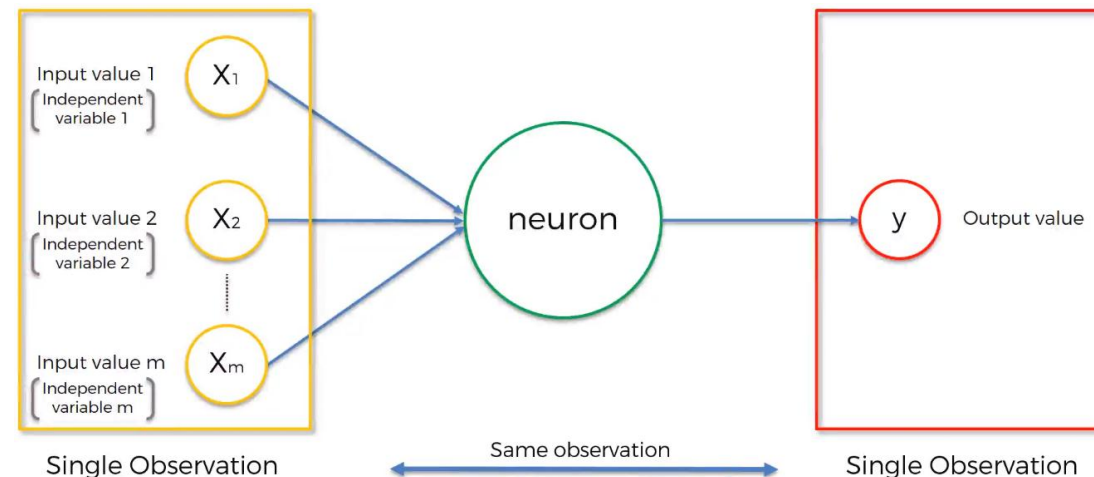
2. Què és una neurona?

Una xarxa neuronal es pot representar a través d'una caixa negra on hi ha un conjunt d'entrades (neurones d'entrada) i una o varies sortides (neurones de sortida). Dins d'aquesta caixa, hi han diverses capes de neurones interconnectades que amaguen la complexitat de l'estructura.



2. Què és una neurona?

Les neurones tenen la capacitat d'aprendre, això s'aconsegueix gràcies als pesos (*weights*) assignats a les connexions entre neurones, que determinen el grau d'importància que té cada una de les neurones de la capa anterior. Aquests valors s'actualitzen a través de la tècnica del *backpropagation* que utilitza el mètode matemàtic del descens del gradient.

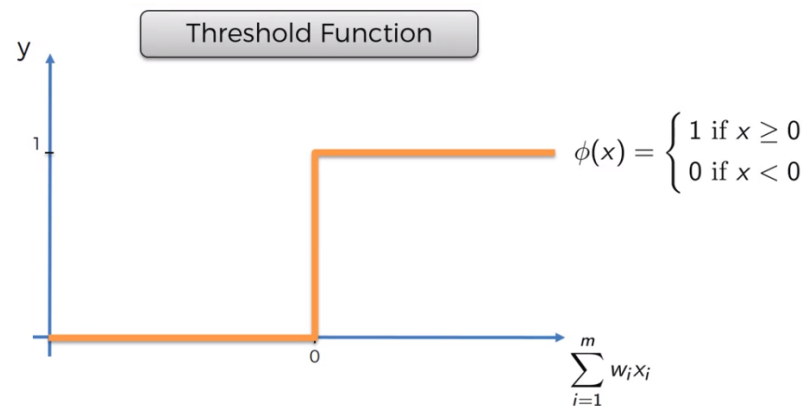


3. Funcions d'activació

Serveixen per calcular el valor que emetrà una neurona com a sortida. Les funcions d'activació més típics són:

- **Threshold function**

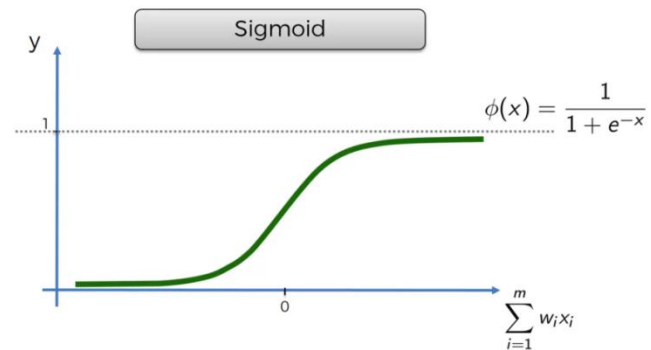
El rang de possibles valors per aquesta funció és 0 o 1, això fa que sigui una funció molt rígida i que s'adapti perfectament a casos on es vol una sortida binària.



3. Funcions d'activació

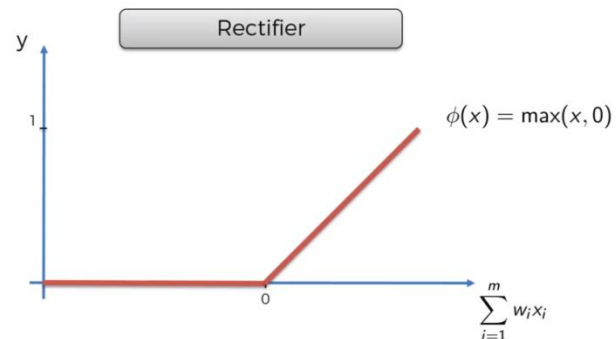
- **Sigmoid**

El rang de valors per aquesta funció va de (0, 1) i es sol utilitzar molt com a funció d'activació en l'última capa de neurones, per calcular probabilitats



- **Rectifier function**

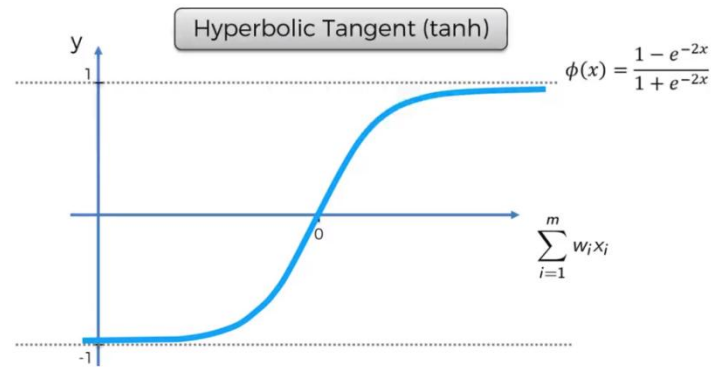
El rang d'aquesta funció va de [0, x], on x correspon a la suma ponderada dels valors d'entrada de la capa de neurones anterior.



3. Funcions d'activació

- **Hyperbolic tangent**

El rang d'aquesta funció va de $(-1, 1)$ i és molt similar a la funció *sigmoid*.

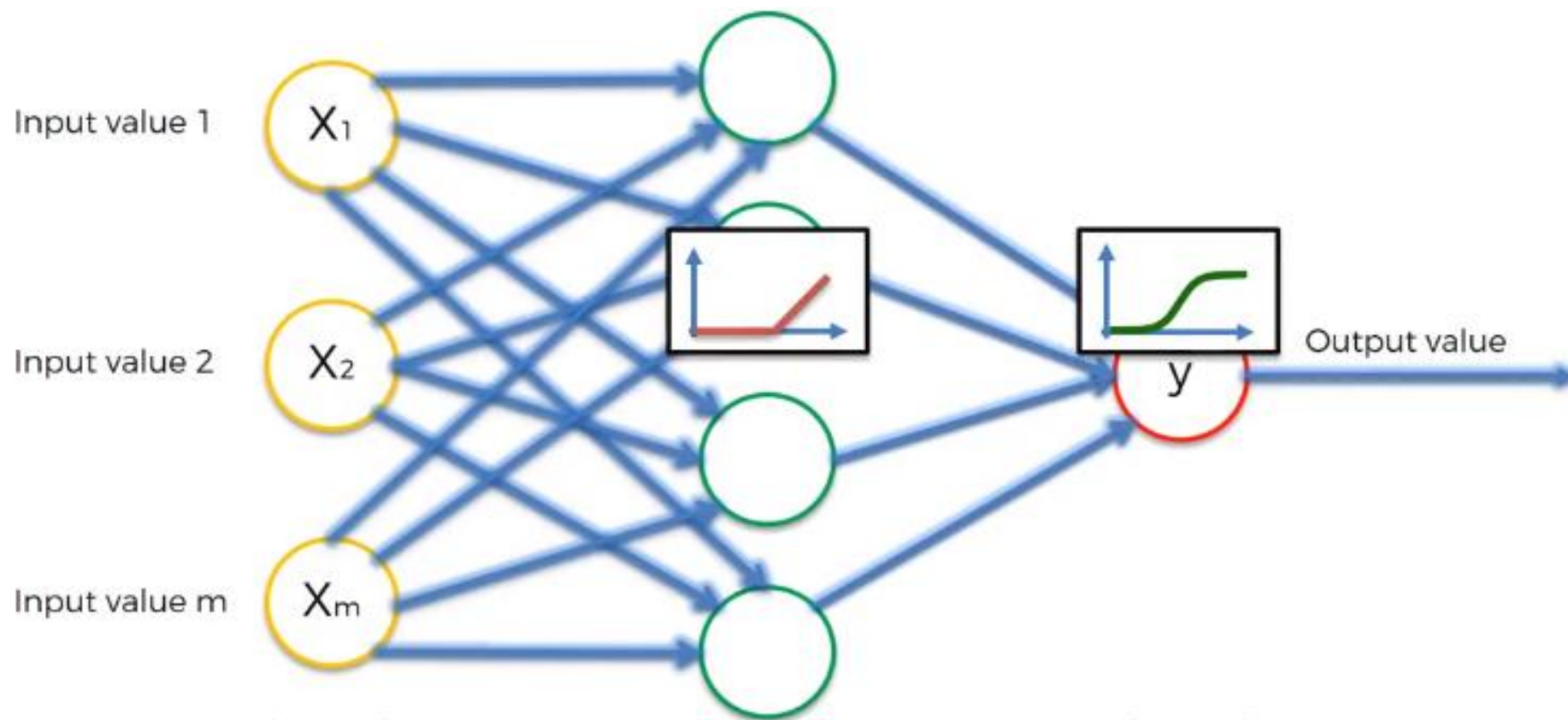


3. Funcions d'activació

Per a cada neurona interna, s'ha de definir una funció d'activació}, per això es poden seguir diferents tècniques.

- Assignar una **mateixa funció d'activació a tota la xarxa neuronal** (rígid però senzill de desenvolupar i mantenir).
- Assignar una **mateixa funció d'activació a nivell de capa de neurones**.
- Assignar una **funció d'activació diferent per a cada neurona** (complex però potent).

3. Funcions d'activació



4. Com funcionen?

El funcionament de les xarxes neuronals es basa en la configuració que s'ha de fer sobre les capes de neurones internes.

A nivell de neurona, s'ha d'elegir quins paràmetres actuen sobre el seu càlcul i crear una connexió que enllaçi les neurones.

Sobre les connexions que uneixen les neurones, s'ha d'assignar un pes que reflexi la importància de l'entrada.

4. Com funcionen?

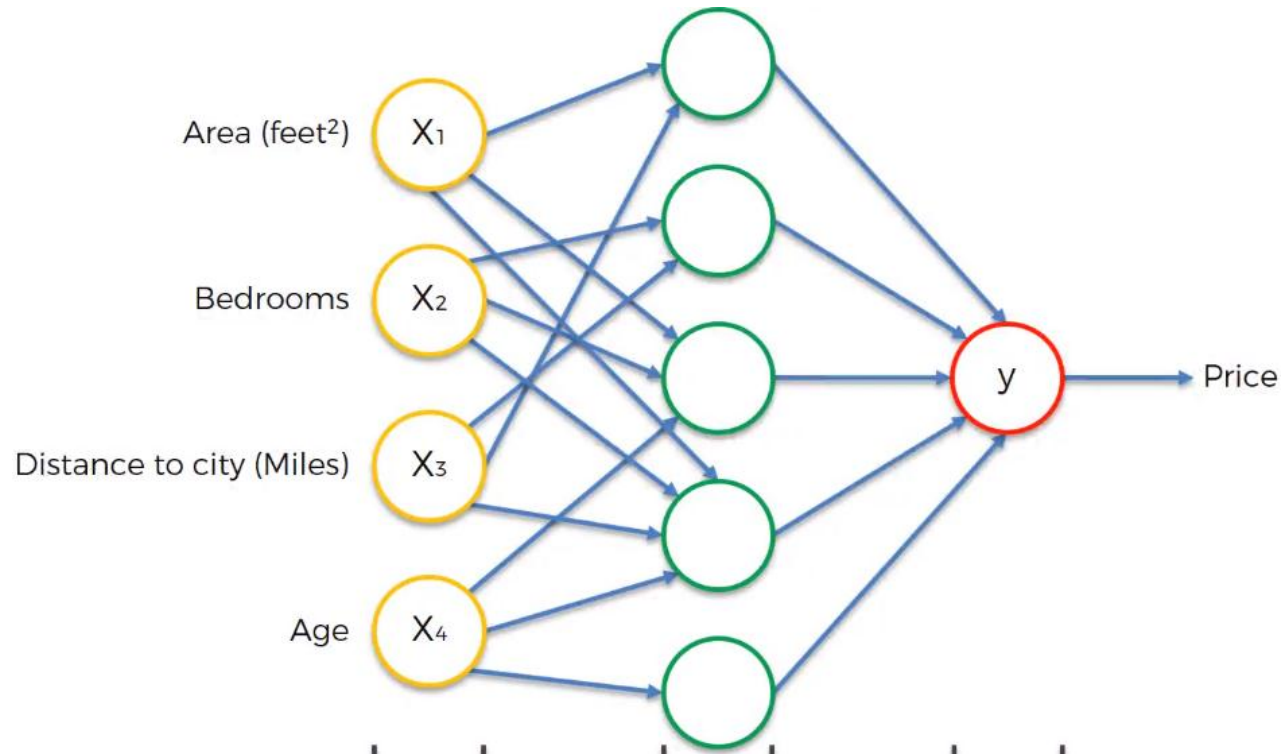
Exemple.

un exemple d'una possible aplicació real, en la que es disposa d'un conjunt dades en base a diferents habitatges que tenen els següents paràmetres:

- Dimensions.
- Nombre de lavabos.
- Distància del centre de la ciutat.
- Antiguitat.

Sobre aquestes dades, es vol especular el preu que pot tenir un habitatge en base a les seves propietats

4. Com funcionen?



Les neurones que no tenen impacte en el càlcul no es representen utilitzant connexions, això es pot interpretar com que tenen un impacte nul sobre aquell càlcul.

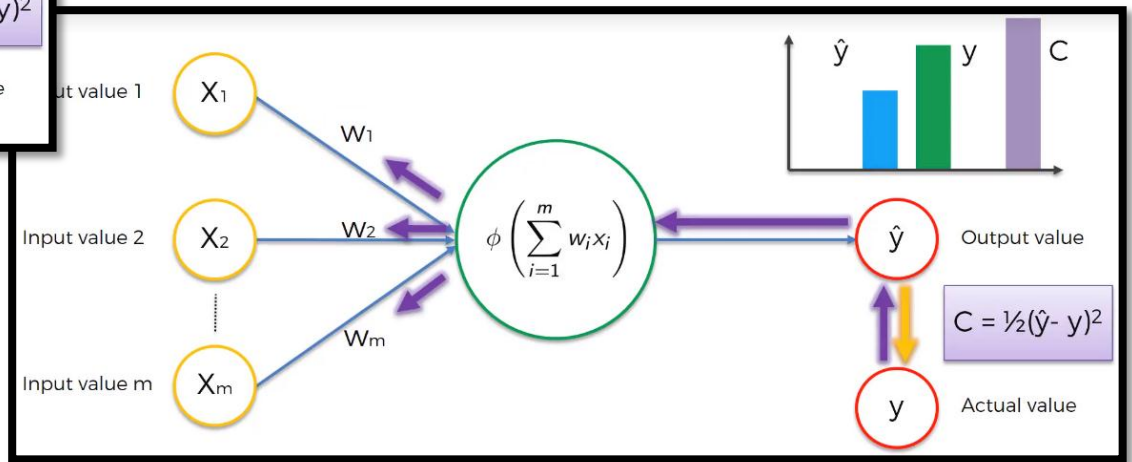
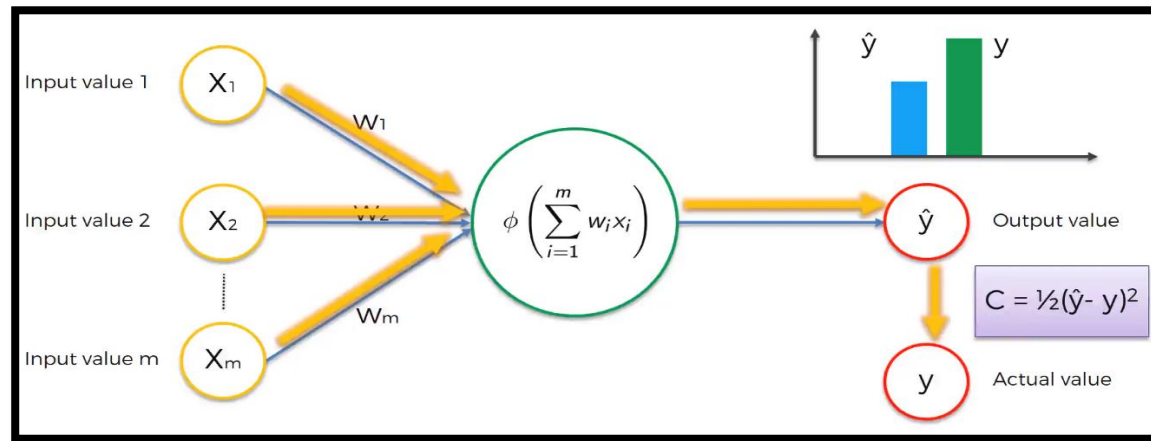
5. Com aprenen?

les xarxes neuronals es basen en un mecanisme d'aprenentatge iteratiu amb el qual van adquirint coneixement a mesura que guanyen experiència. Per aquest motiu, és necessari dividir les etapes de funcionament d'una xarxa neuronal en:

- **Etapa d'entrenament:** en aquesta etapa es proporcionen múltiples exemples a partir dels quals s'adquireix la gran majoria del coneixement. S'ha de tenir en compte, que durant aquesta etapa cal saber el resultat correcte per obtenir una xarxa ben entrenada.
- **Etapa de test:** en aquesta etapa es posa a prova la xarxa, proporcionant noves dades i analitzant els resultats. En cas de comptar amb el resultat correcte per aquestes dades, la xarxa pot continuar aprenent.

5. Com aprenen?

Durant l'etapa d'entrenament i un cop s'obté el resultat proporcionat de la xarxa, cal comparar-lo amb el resultat real i extreure un valor que representi la distància que hi ha entre el valor predit i el correcte. Aquesta magnitud s'extreu a partir de la funció de cost.



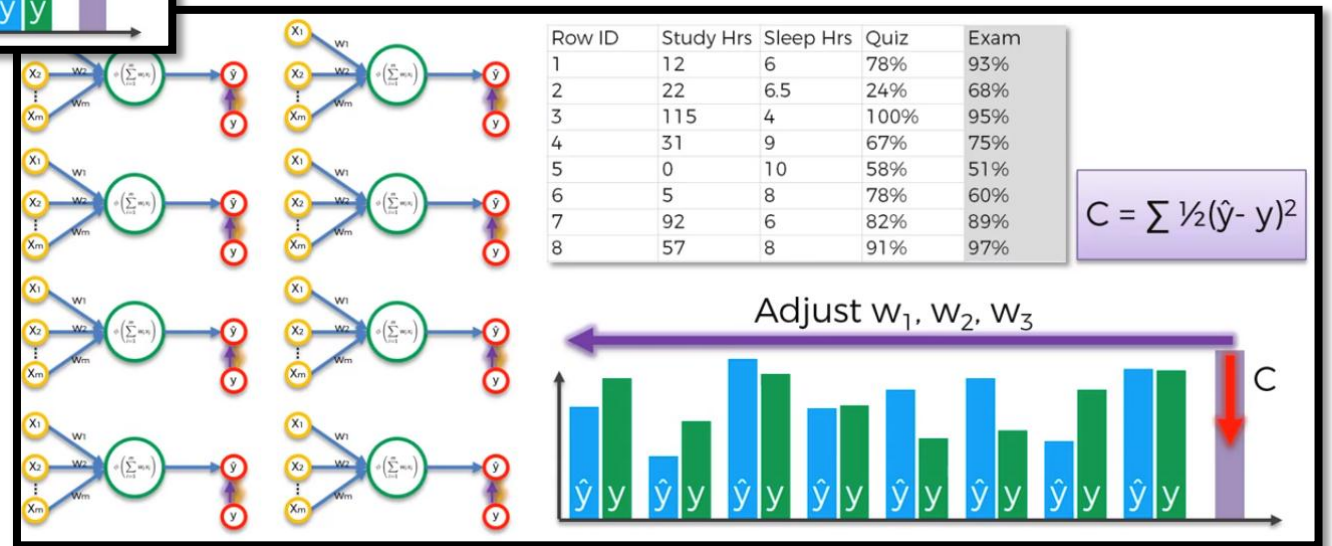
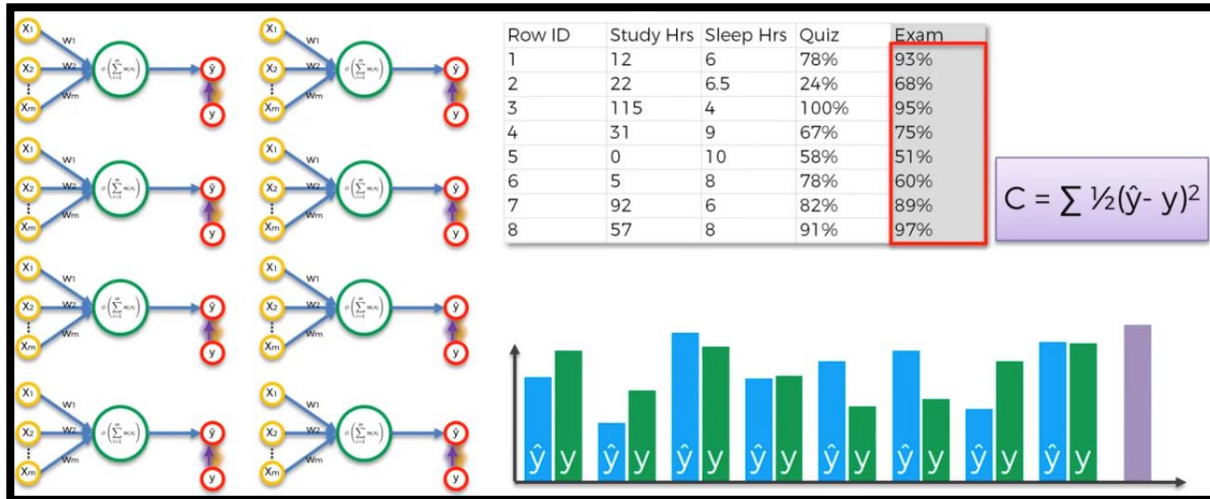
5. Com aprenen?

Exemple.

Un possible exemple, seria la utilització d'una xarxa neuronal per a desenvolupar una regressió que calculés una aproximació de la nota a treure en un examen d'acord amb:

- Nombre d'hores emprades a estudiar.
- Nombre d'hores dormides.
- Última nota obtinguda en el test.

5. Com aprenen?



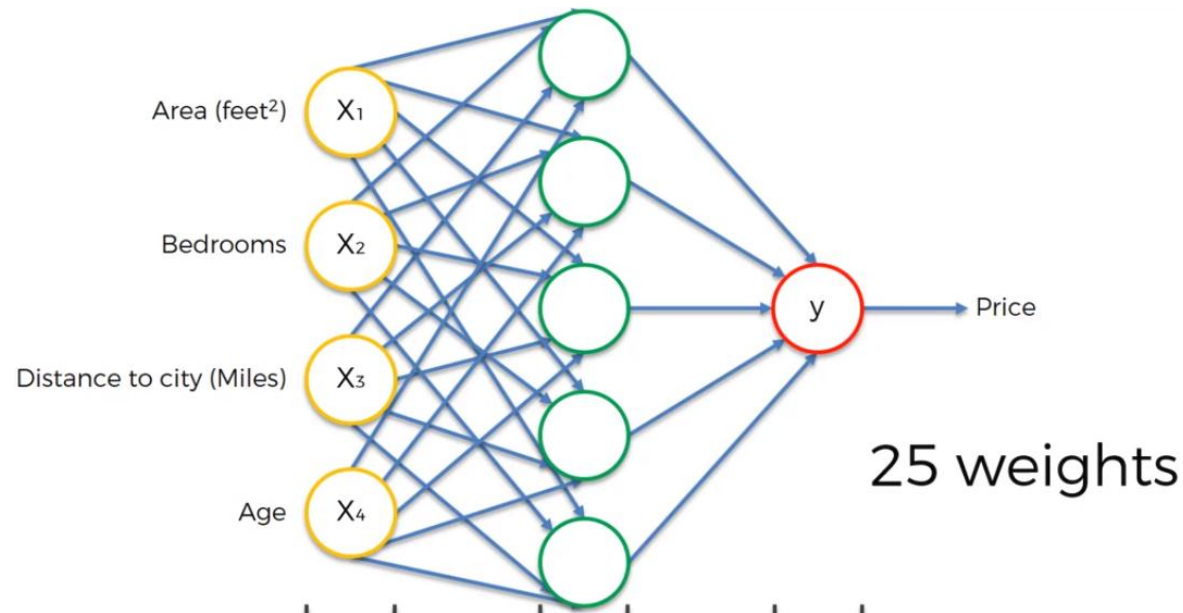
6. Descens del gradient

Per aconseguir que la xarxa neuronal aprengui, cal modificar els pesos de les *synapses* que connecten les diferents capes de neurones.

Per veure la importància d'aquest mètode, assumint que el rang de valors que es poden assignar als pesos de les connexions és $[1, 1000]$, en aquesta xarxa tan senzilla hi ha vint-i-cinc connexions i per tant **hi ha 1000^{25} combinacions possibles.**

6. Descens del gradient

Tenint en compte que el supercomputador més potent a nivell mundial *sunway taihulight* te una potencia de càlcul d'aproximadament **93 PFLOPs**, Caldrien **$1,08 \times 10^{58}$ segons**, es a dir **$3,42 \times 10^{50}$ anys** per configurar la xarxa, tot això **tenint en compte que es pot fer una assignació de pesos per operació, cosa que no pot ser.**



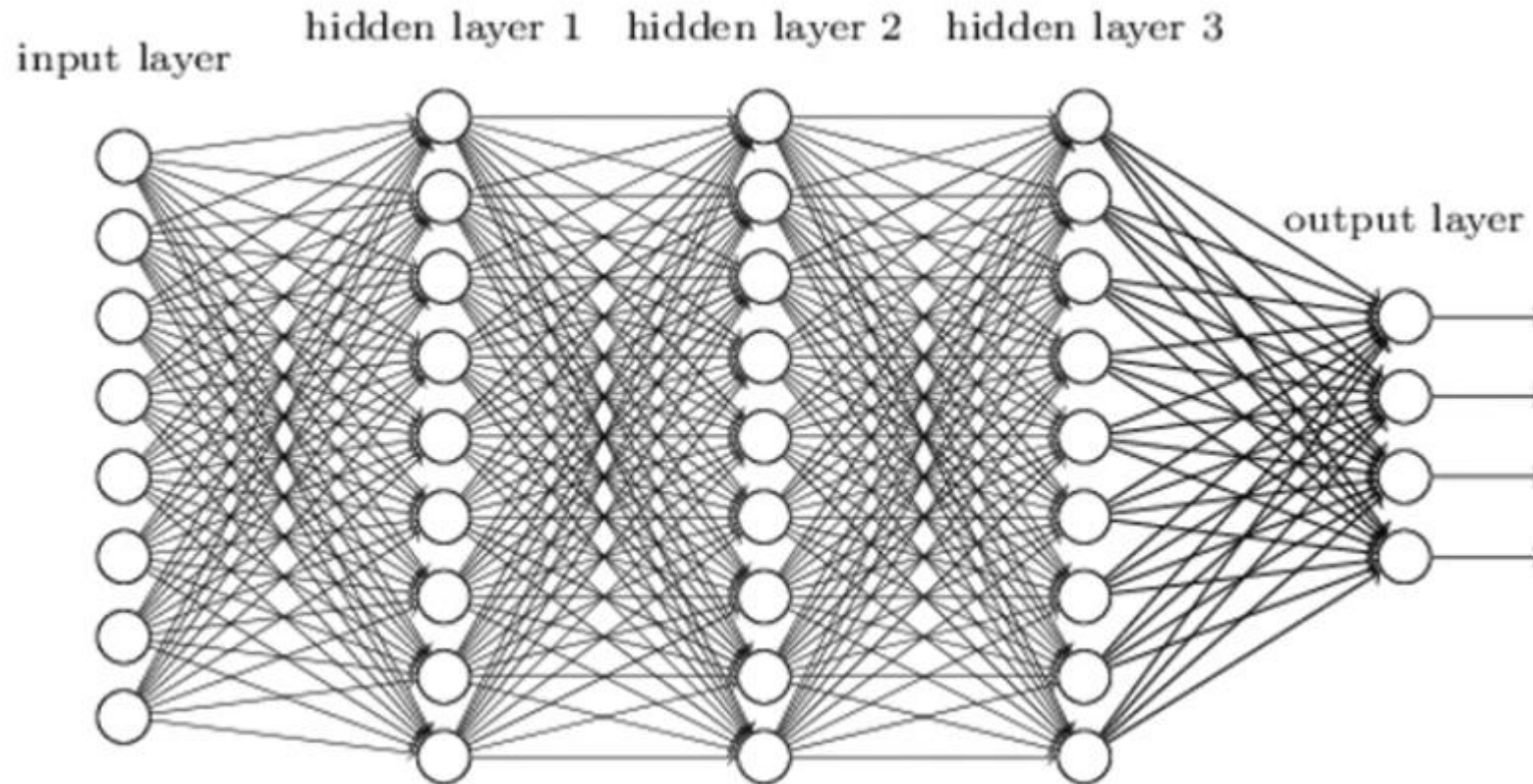
6. Descens del gradient



The sunway taihulight

6. Descens del gradient

I ara que passaria si tenim la següent xarxa???



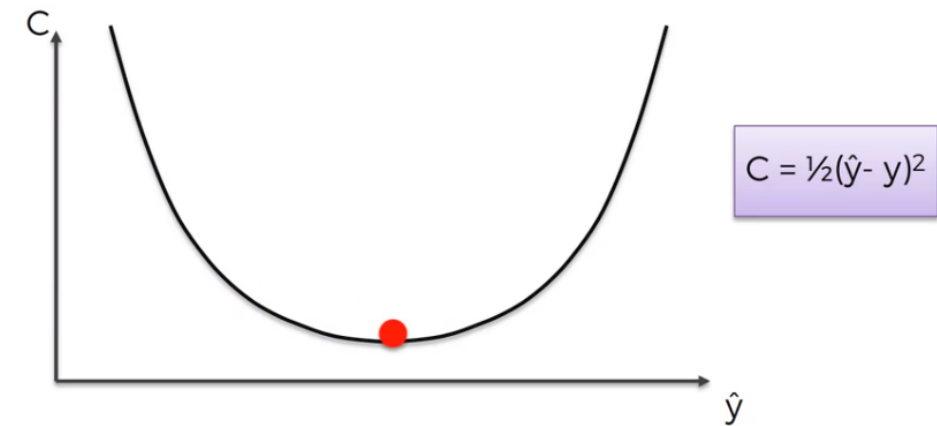
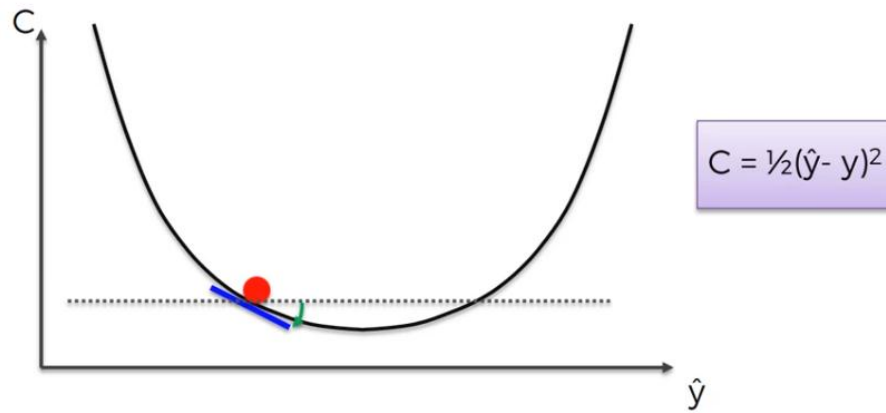
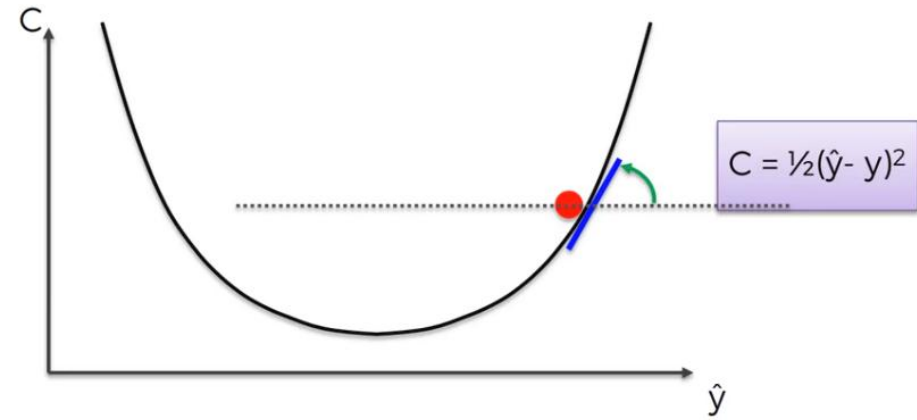
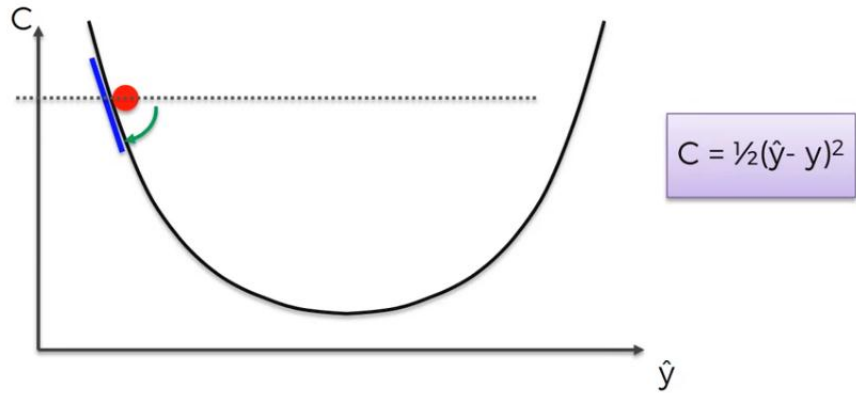
6. Descens del gradient

Vist que utilitzar força bruta no serveix en aquest problema, cal utilitzar un altre mètode que en aquest cas és el descens del gradient.

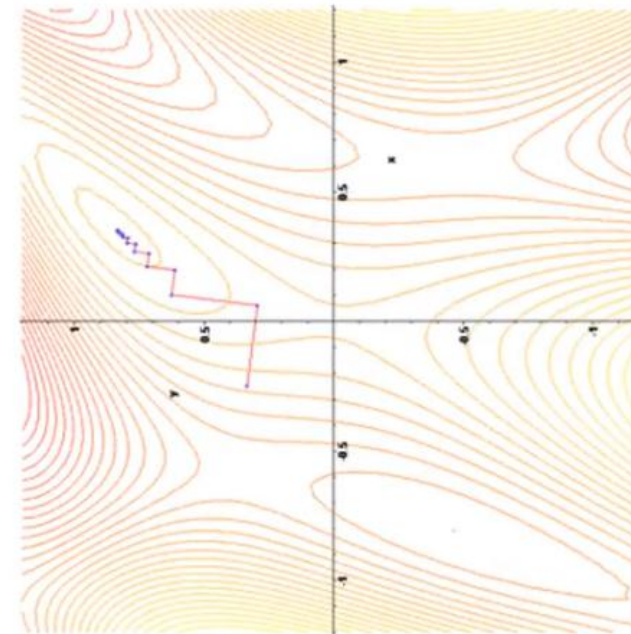
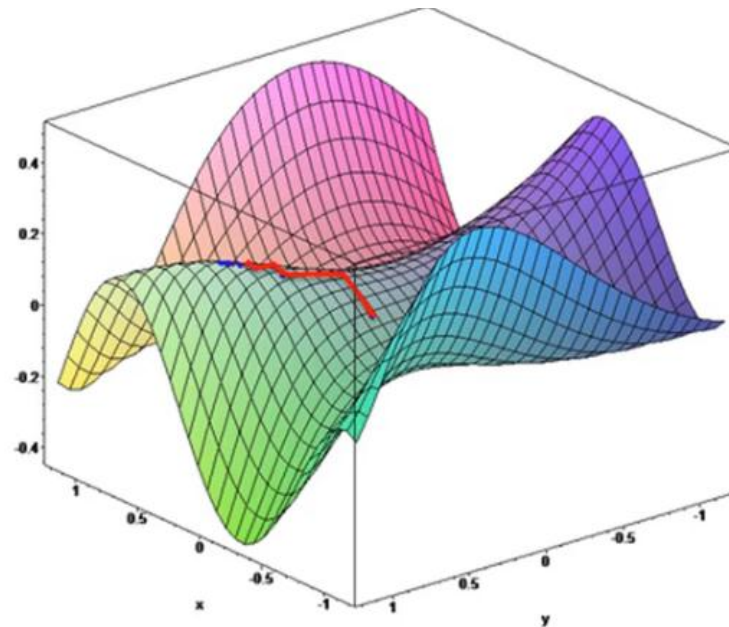
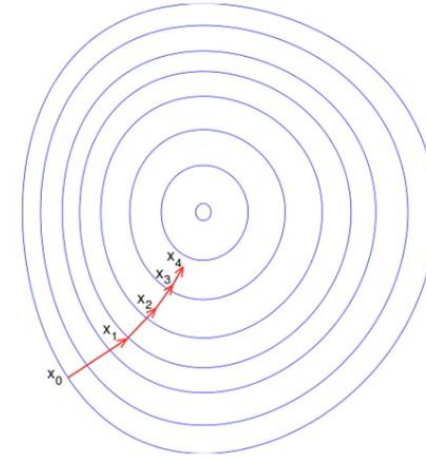
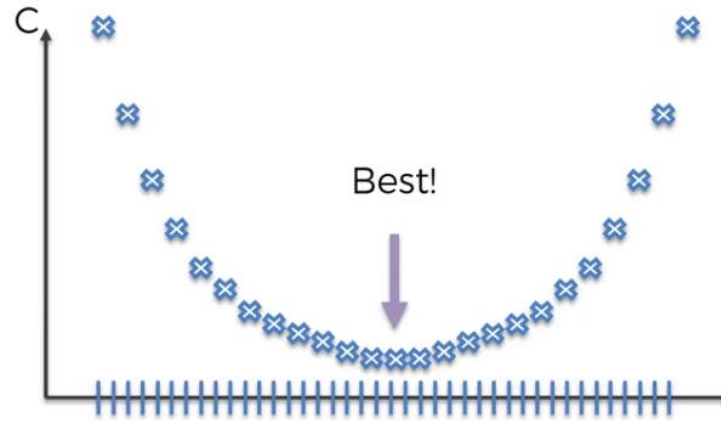
El descens del gradient és el mètode matemàtic en què es basa la tècnica *backpropagation* per aconseguir absorbir el coneixement adquirit durant la fase d'entrenament.

Aquesta tècnica radica en buscar el camí més curt per reduir la funció de cost, això es fa a través de la derivada de la funció de cost que determina el pendent en un punt donat.

6. Descens del gradient



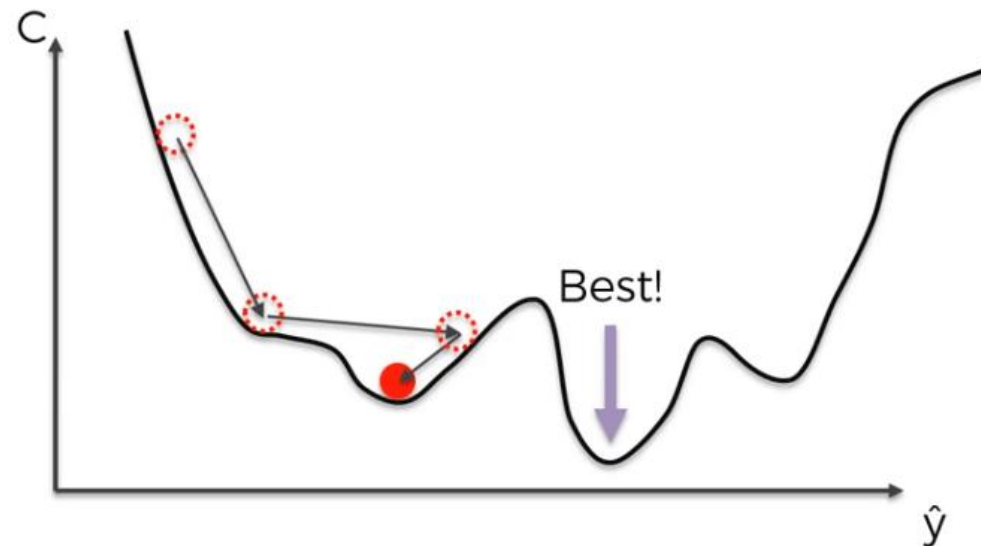
6. Descens del gradient



7. Descens del gradient estocàstic

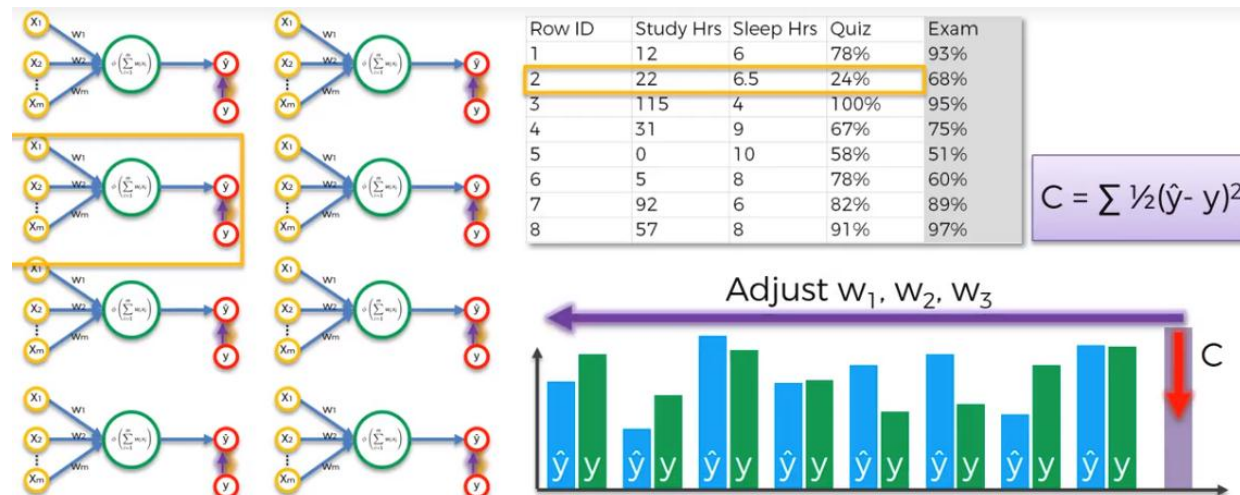
El típic problema que es pot trobar a l'hora d'utilitzar el mètode del descens del gradient, és el de no arribar a trobar l'òptim global, a causa de que s'ha 'caigut' a un òptim local.

Això vol dir que no s'arriba a entrenar la xarxa tant com es voldria. Per resoldre aquest problema, es pot utilitzar el mètode del descens del gradient estocàstic.



7. Descens del gradient estocàstic

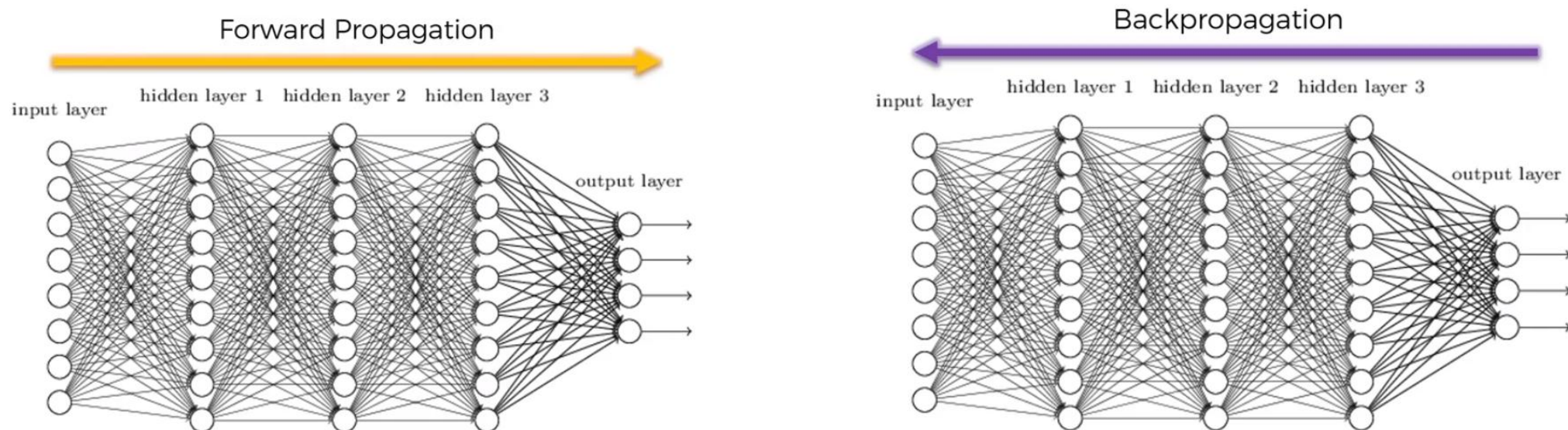
El descens del gradient estocàstic s'aconsegueix aplicant la funció de *backpropagation* per cada registre processat per la xarxa, d'aquesta manera s'obté una major fluctuació i així hi ha més probabilitat de no 'caure' en un òptim local.



És important veure que aquest mètode és **indeterminista**, ja que el resultat dependrà de l'ordre en què es processin els registres.

8. Backpropagation

El Backpropagation és un algorisme complex a nivell matemàtic que permet ajustar tots els pesos a l'hora, és a dir, permet identificar quins són els pesos que estan causant més error i en quina mesura s'han de modificar. Durant l'entrenament d'una xarxa neuronal, es poden destacar dues fases.



9. Fases en l'entrenament d'una xarxa

Les fases que es segueixen durant l'entrenament d'una xarxa són:

1. Iniciar aleatòriament (amb un valor pròxim a 0) els pesos de les connexions.
2. Assignar un valor (de la mateixa observació) a cada una de les neurones de la capa d'entrada.
3. *Forwardpropagation* activant les neurones.
4. Calcular l'error generat a través de la funció de cost.
5. *Backpropagation*, corregint l'error comès.
6. Repetir passos 1-5 fins a arribar a processar totes les observacions.
7. S'ha completat una *epoch*, fer més *epochs* repetint passos 1-6.

10. Demonstráció



Bibliografia

- Curs de *Machine Learning* (Udemy)
 - <https://www.udemy.com/machinelearning/learn/v4/overview>
- Xarxa neuronal
 - https://en.wikipedia.org/wiki/Artificial_neural_network
- *Deep sparse rectifier neural networks*
 - <http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf>
- Funcions de cost més típiques
 - <https://stats.stackexchange.com/questions/154879/a-list-of-cost-functions-used-in-neural-networks-alongside-applications>
- Top 500 supercomputers
 - <https://www.top500.org/system/178764>
- Descendent del gradient
 - <http://iamtrask.github.io/2015/07/27/python-network-part2/>
- Descendent del gradient i Backpropagation
 - <http://neuralnetworksanddeeplearning.com/chap2.html>

Gràcies!
Preguntes?