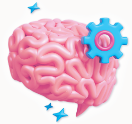
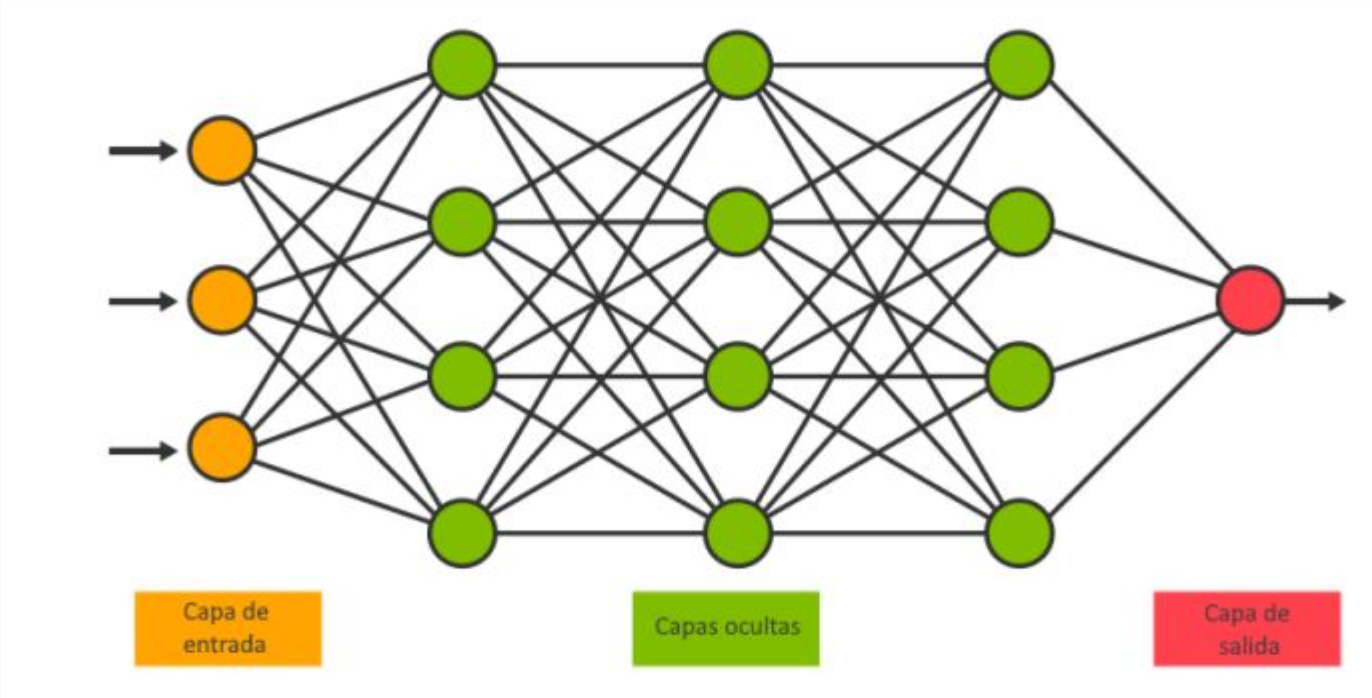


Ayudantía 12

Redes Neuronales

Por Josefina Israel y Felipe Vidal

24 de noviembre 2023


Capa de
entrada
Capas ocultas
Capa de
salida

La Neurona o Perceptrón

¿Pollo frito o perrito?

→ Tiene algo que parecen ojos

→ Lleva algo que parece un collar

→ No está en un plato

→ No está junto a otros alimentos

x1 Ojos : 1

x2 Collar : 1

x3 No está en un plato: 1

x4 No está junto a otros alimentos: 1

x1+x2+x3+x4 > 3 → perrito

→ **1**

Entradas

x1 Ojos : 1

x2 Collar : 1

x3 No está en un plato: 1

x4 No está junto a otros alimentos: 1

x1+x2+x3+x4 >3
→ perrito

Salida

→ **1**

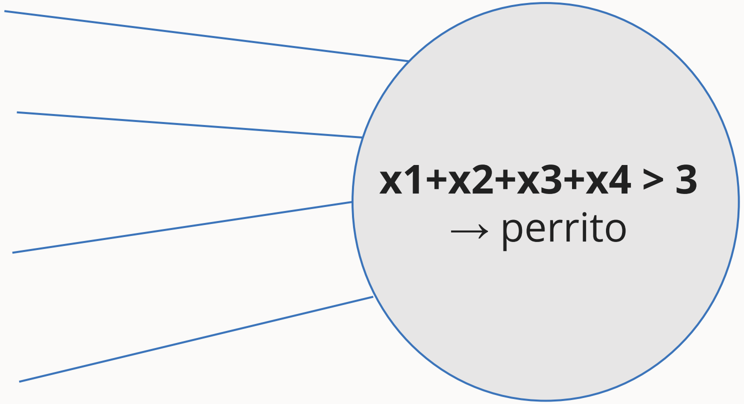
Entradas

x1 Ojos : 1

x2 Collar : 1

x3 No está en un plato: 0

x4 No está junto a otros alimentos: 1



Salida

→ 0

Entradas

x1 Ojos : 1

x2 Collar : 1

x3 No está en un plato: 0

x4 No está junto a otros alimentos: 1

3

2

1

2

x1w1+x2w2+x3w3 +x4 w4 > 6 → perrito

Salida

→ 1

Entradas

x1 Ojos : 1

x2 Collar : 1

x3 Está en un plato: 1

x4 Está junto a otros alimentos: 0

3

2

-1

-2

x1w1+x2w2+x3w3 +x4 w4 > 4

→ perrito

Salida

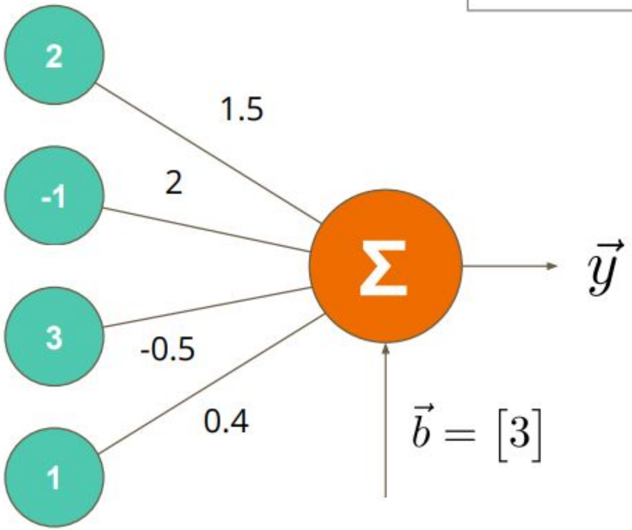
→ **1**

BIAS = - UMBRAL

$$WX + b \leq 0 \rightarrow Y = 0$$

$$WX + b > 0 \rightarrow Y = 1$$

$$\boxed{W\vec{x}+\vec{b}=\sum_{i=1}^{m}(w_i x_i)+\vec{b}=\vec{y}}$$



$$\vec{x}=\begin{bmatrix}2\\-1\\3\\1\end{bmatrix}\qquad W=\begin{bmatrix}1.5\\2\\-0.5\\0.4\end{bmatrix}$$

$$\vec{b}=[3]$$

Quiz en Menti! Pregunta 1

Teniendo en cuenta los vectores de input y del perceptrón, ¿cuál es el output?

$$X = \begin{pmatrix} 3 \\ -5 \\ 2 \end{pmatrix}, \quad W = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \quad b = -3$$

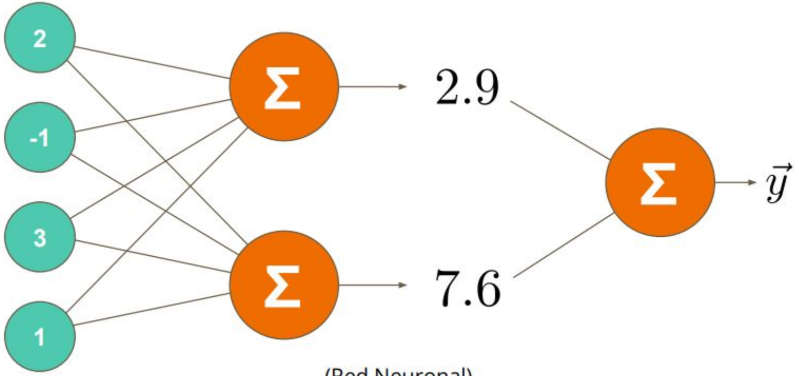
Combinando perceptrones

Una red neuronal es una combinación de perceptrones.

La salida de cada perceptrón de una capa actúa como entrada para el perceptrón de la siguiente capa, hasta llegar a la capa final.

La gracia está en que la red neuronal se entrena hasta encontrar los parámetros (pesos y bias) que permiten generar todas las combinaciones cuyo output será la clasificación correcta.

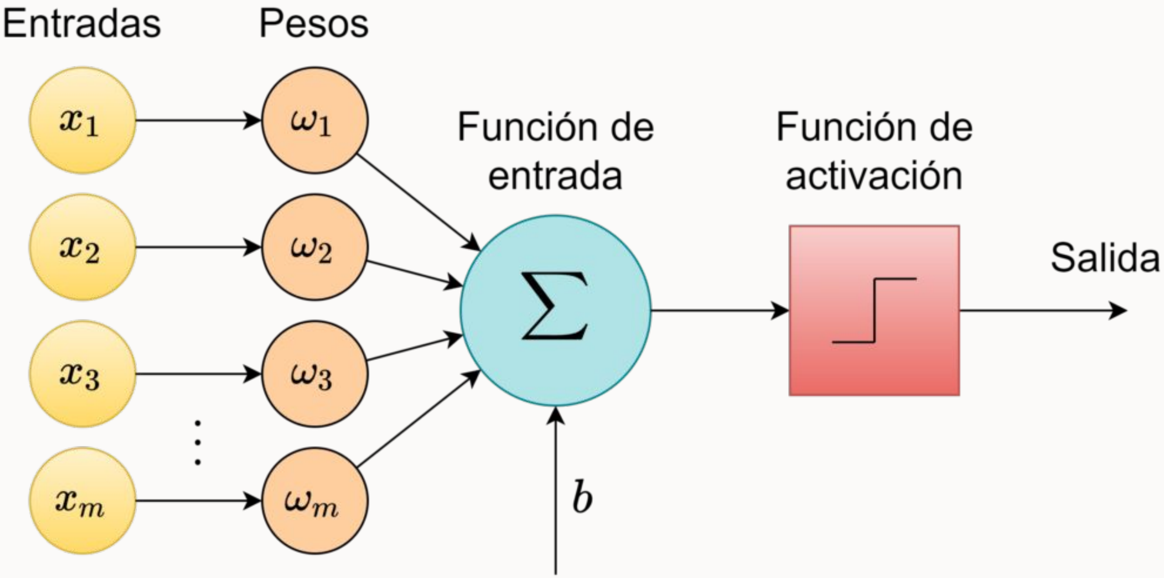
RED NEURONAL



(Red Neuronal)

Función de activación: f(x1w1+x2w2+x3w3+x4 w4...)

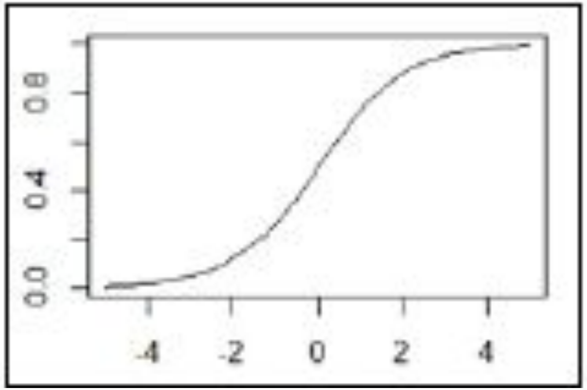
Hasta ahora simplemente estábamos haciendo una función escalonada sin embargo es preferible el uso de otras con el fin de no producir cambios de valores instantáneos si no que sean de manera gradual.

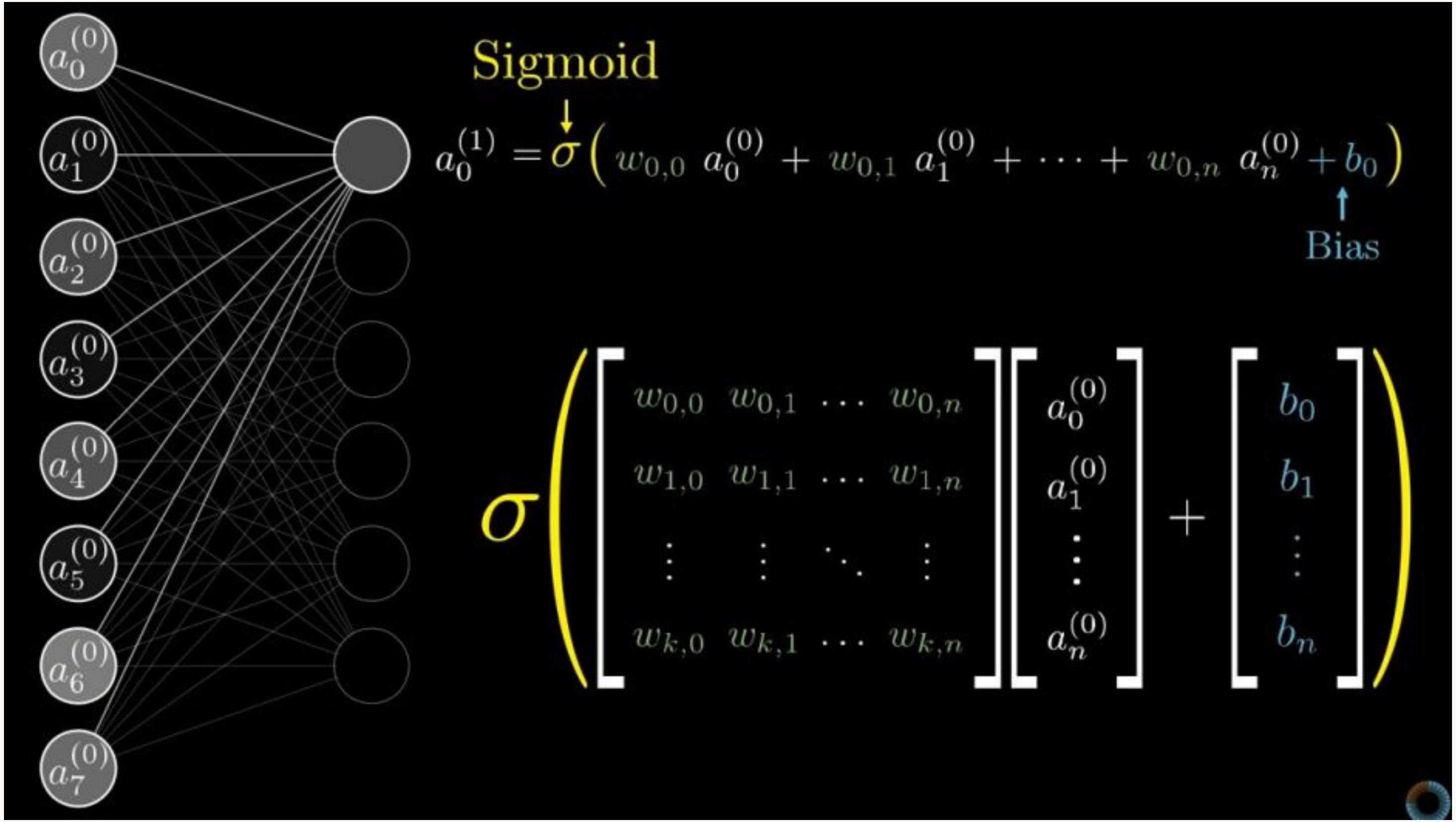


Quiz en Menti! Pregunta 2

¿Qué sucede si un MLP no incorpora funciones de activación **no-lineales** entre capas?

Función de activación: Sigmoidal

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


Sigmoid
$a_0^{(1)} = \sigma\left(w_{0,0}\, a_0^{(0)} + w_{0,1}\, a_1^{(0)} + \cdots + w_{0,n}\, a_n^{(0)} + b_0\right)$
Bias
$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}\right)$
$a_0^{(0)}$ $a_1^{(0)}$ $a_2^{(0)}$ $a_3^{(0)}$ $a_4^{(0)}$ $a_5^{(0)}$ $a_6^{(0)}$ $a_7^{(0)}$

Quiz en Menti! Pregunta 3

Siguiendo la P1. Si la última capa es una función sigmoidal, y el **resultado** corresponde a la **probabilidad** de ser un perrito…

¿Cuál es la probabilidad entregada por el modelo?

Encontrar los parámetros adecuados.

¿Cómo autoajusta sus parámetros la red neuronal?

Entrenando una red neuronal

Necesitamos un set de datos etiquetados, con las características pasadas a datos numéricos. Luego separamos entre datos de entrenamiento y de prueba. Los parámetros se irán ajustando con el set de entrenamiento hasta obtener los resultados correctos.

Tiene Collar	**Tiene ojos**	**Está en un plato**	**Está junto a otros alimentos**
1	1	0	0
0	0	1	1
1	1	1	0

Perrito
1
0
1

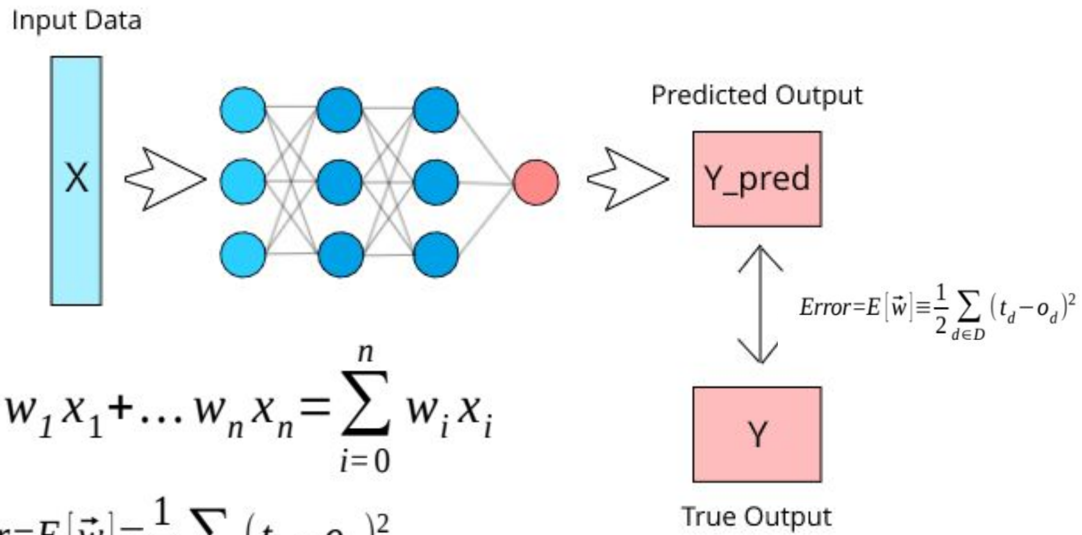
Quiz en Menti! Pregunta 4

“Deben usarse los mismos datos para **entrenar** y para **testear**”

¿Es cierto lo anterior?

Función de Costo o Loss

Partimos con parámetros al azar y usamos la Función que cuantifica la diferencia promedio entre los valores predecidos y los valores reales.



Training set: (x_i, t_i)

- x_i: input features.
- t_i: target output.
- o_i: predicted output.

$$O = w_0 + w_1 x_1 + \ldots w_n x_n = \sum_{i=0}^{n} w_i x_i$$

$$Error = E[\vec{w}] \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

Gradient Descent

Queremos calcular el gradiente sobre la función de Costo, con el objetivo de movernos en la dirección contraria en que este aumenta y así disminuirlo de forma iterativa.

Incremental mode SDG: rápido y efectivo

1. Elige un subset aleatorio del conjunto de entrenamiento
2. Estimar la dirección del gradiente de $Error = E[\vec{w}] \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$
3. Actualizar los pesos usando el negativo de la dirección del gradiente

Batch mode Gradient Descent:
Do until satisfied

1. Compute the gradient $\nabla E_D[\vec{w}]$
2. $\vec{w} \leftarrow \vec{w} - \eta \nabla E_D[\vec{w}]$

Incremental mode Gradient Descent:
Do until satisfied

- For each training example d in D
  1. Compute the gradient $\nabla E_d[\vec{w}]$
  2. $\vec{w} \leftarrow \vec{w} - \eta \nabla E_d[\vec{w}]$

$$E_D[\vec{w}] \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

$$E_d[\vec{w}] \equiv \frac{1}{2} (t_d - o_d)^2$$

Quiz en Menti! Pregunta 5

¿Por qué nos interesa que las funciones de activación sean **diferenciables**?

Gradient Descent: Learning Rate η

El learning rate es un parámetro que usamos como ponderador para el gradiente, en el SGD. En general usamos valores cercanos a 1 cuando estamos lejos del óptimo y cercanos a 0 cuando estamos cerca.

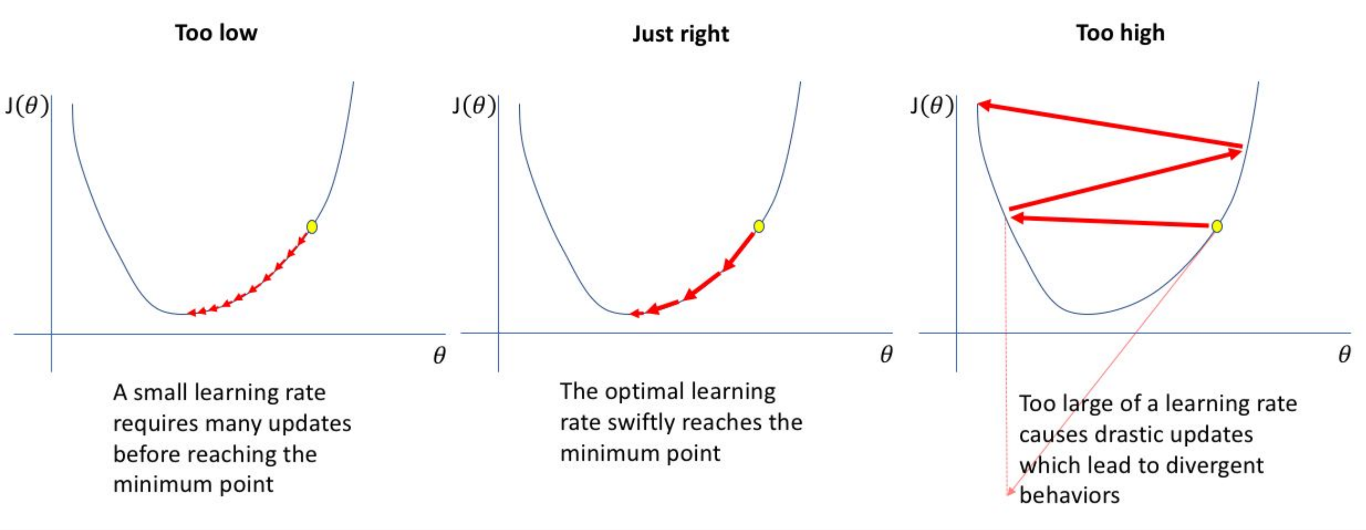
Su función es hacernos descender por la función de Costo (Loss) de la forma más “eficiente” posible

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Learning rate

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

Gradient Descent: Learning Rate η



Quiz en Menti! Pregunta 6

¿Cuáles de las siguientes afirmaciones son correctas respecto al **learning rate**?

En resumen hasta ahora:

La NN predice → calcula el Loss → Obtiene su gradiente → actualiza los parámetros → repite

Backpropagation

Aplicar la regla de la cadena sobre el error en la salida, pasando por cada una de las capas con el fin de encontrar cómo varía el error al variar los pesos

$$\frac{\partial Loss(y, y_{true})}{\partial w_3} = \frac{\partial Loss(y, y_{true})}{\partial y} \frac{\partial y}{\partial w_3}$$

$$\frac{\partial Loss(y, y_{true})}{\partial w_2} = \frac{\partial Loss(y, y_{true})}{\partial y} \frac{\partial y}{\partial w_3} \frac{\partial x_3}{\partial w_2}$$

$$\frac{\partial Loss(y, y_{true})}{\partial w_1} = \frac{\partial Loss(y, y_{true})}{\partial w_2} \frac{\partial x_2}{\partial w_1}$$

...

```
Vamos al jupyter
```