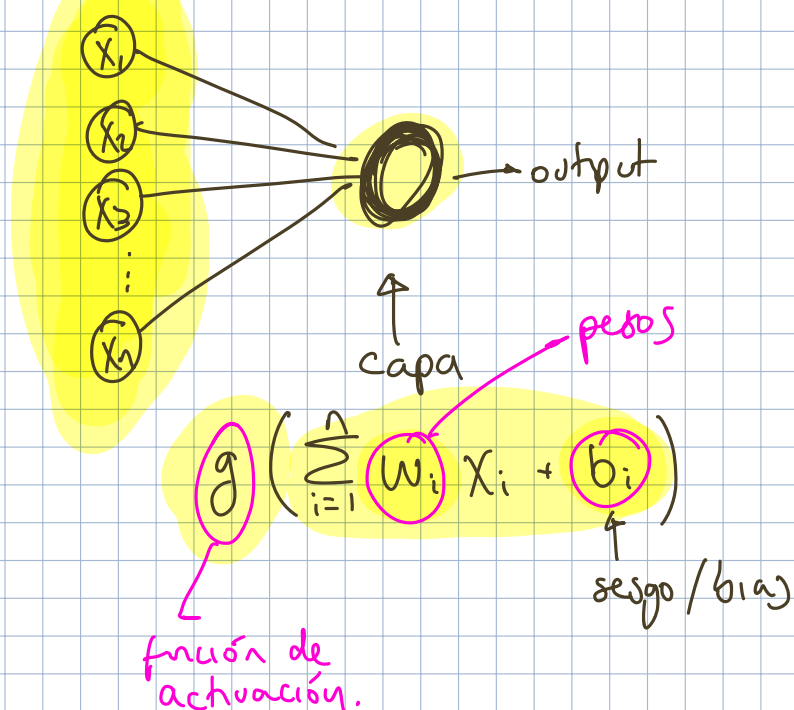


Redes neuronales:

Perceptrón: Red neuronal de 1 capa:

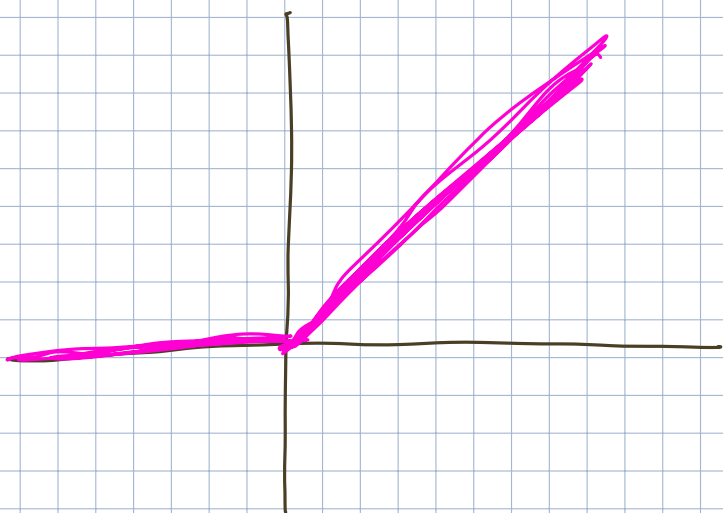
n features:



clasificación binaria / regresión

funciones de activación:

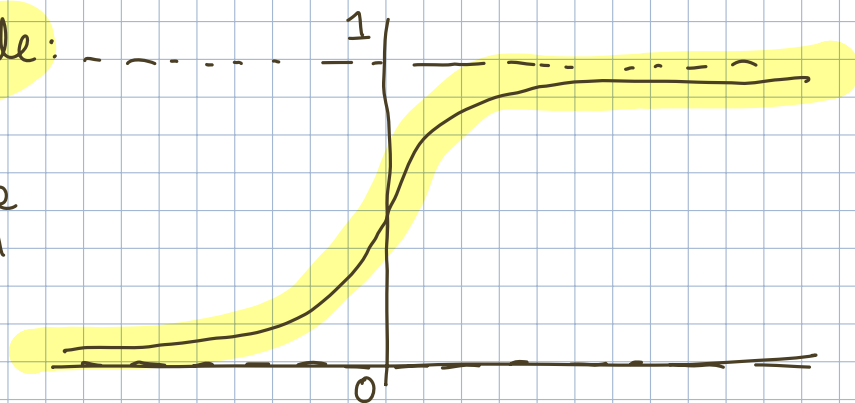
- ReLU: rectified linear unit



$$\text{Relu}(x) = \begin{cases} x & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

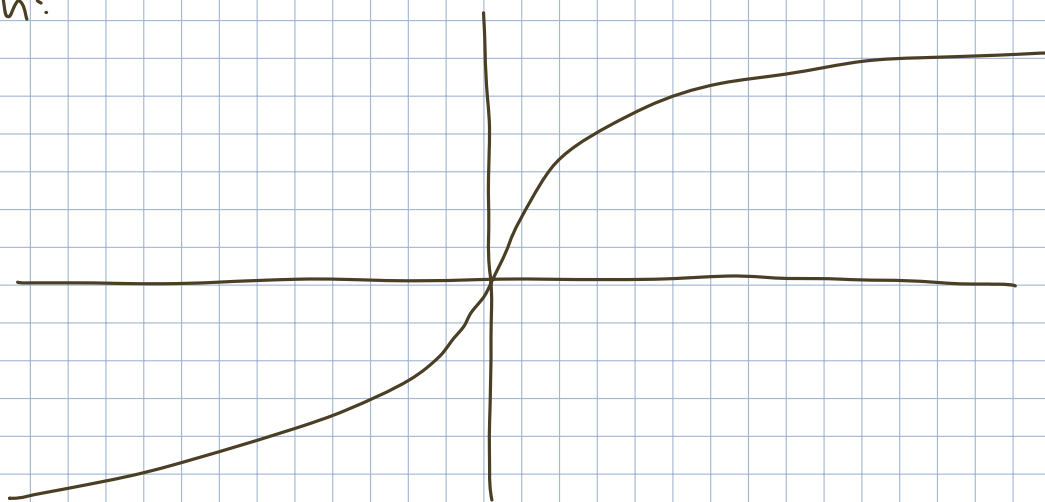
• Sigmoide:

↓
problemas de clasificación



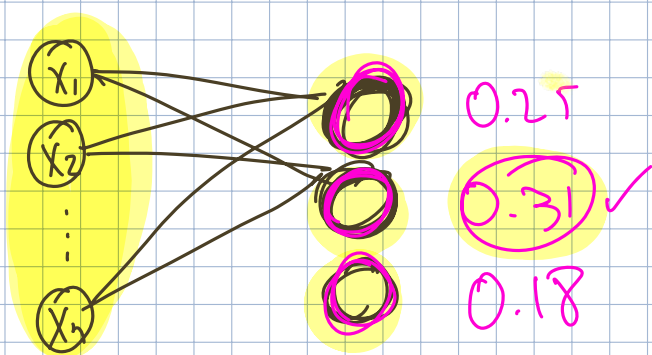
$$f(x) = \frac{1}{1 + e^{-x}}$$

• Tanh:



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

Clasificación multiclase:



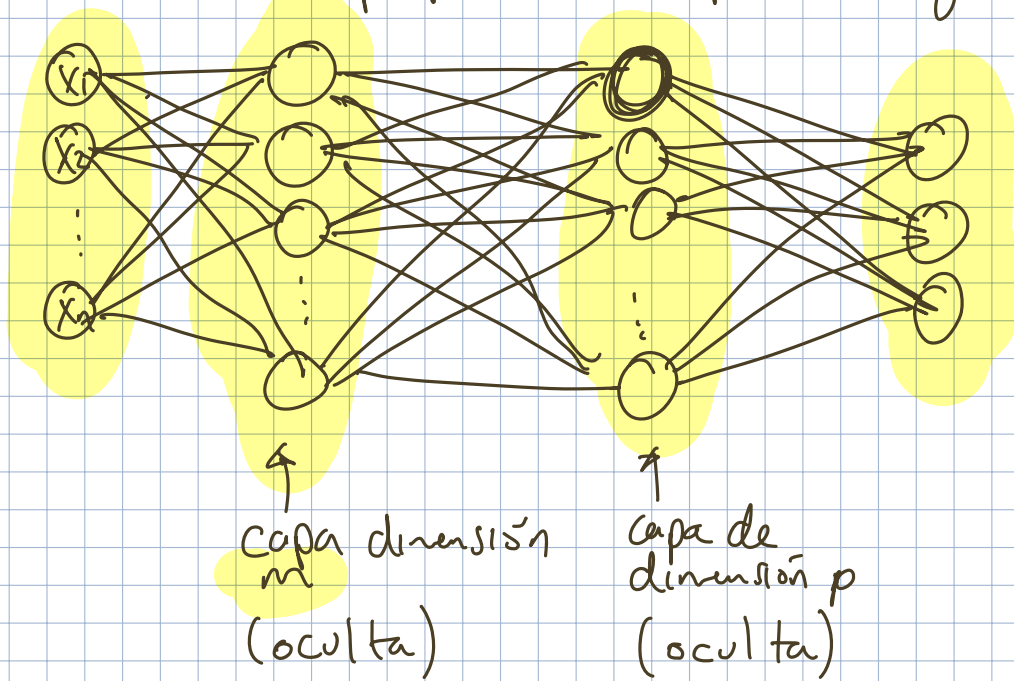
$$\begin{aligned} g\left(\sum_{i=1}^n w_i^1 x_i + b^1\right) \\ g\left(\sum_{i=1}^n w_i^2 x_i + b^2\right) \\ g\left(\sum_{i=1}^n w_i^3 x_i + b^3\right) \end{aligned}$$

$$(b^1 \quad w_1^1 \quad w_2^1 \quad \dots \quad w_n^1) \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = x^t w^1$$

$$(b^2 \quad w_1^2 \quad w_2^2 \quad \dots \quad w_n^2) \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = x^t w^2$$

$$g\left(\begin{pmatrix} b^1 & w_1^1 & w_2^1 & \dots & w_n^1 \\ b^2 & w_1^2 & w_2^2 & \dots & w_n^2 \\ b^3 & w_1^3 & w_2^3 & \dots & w_n^3 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}\right) = g(x^t w)$$

Red neuronal profunda (deep learning):



densa: todas las outputs de las neuronas de una capa se conectan con las neuronas de la siguiente capa.

$$\begin{aligned} g' \left(\sum_{i=1}^n w_i^1 x_i + b^1 \right) \\ g' \left(\sum_{i=1}^n w_i^2 x_i + b^2 \right) \\ \vdots \\ g' \left(\sum_{i=1}^n w_i^m x_i + b^m \right) \end{aligned}$$

$$a^{(0)} = X$$

$$z^{(1)} = a^{(0)} + w^{(1)}$$

$$a^{(1)} = g^1(a^{(0)} + w^{(1)})$$

$a^{(i)}$ es el output de la capa i

$$a^{(2)} = g^2(a^{(1)} + w^{(2)})$$

$$a^{(3)} = g^3(a^{(2)} + w^{(3)})$$

$$\hat{y} = a^{(3)} = g^3(a^{(2)} + w^{(3)}) = g^3(g^2(a^{(1)} + w^{(2)}) + w^{(3)})$$

$$\hat{y} = g^3(g^2(g^1(x^T w^{(1)}) + w^{(2)}) + w^{(3)})$$

2 etapas:

- feed forward
- back propagation.

$$(x^1, y^1), (x^2, y^2) \dots (x^s, y^s)$$

\uparrow
n-dimensional.

Inicializamos los pesos $W^{(1)}, W^{(2)}, W^{(3)}$ aleatoriamente.

Tomamos cada observación: $1, \dots, s$ y las pasamos por la red neuronal con pesos aleatorios para obtener una predicción:

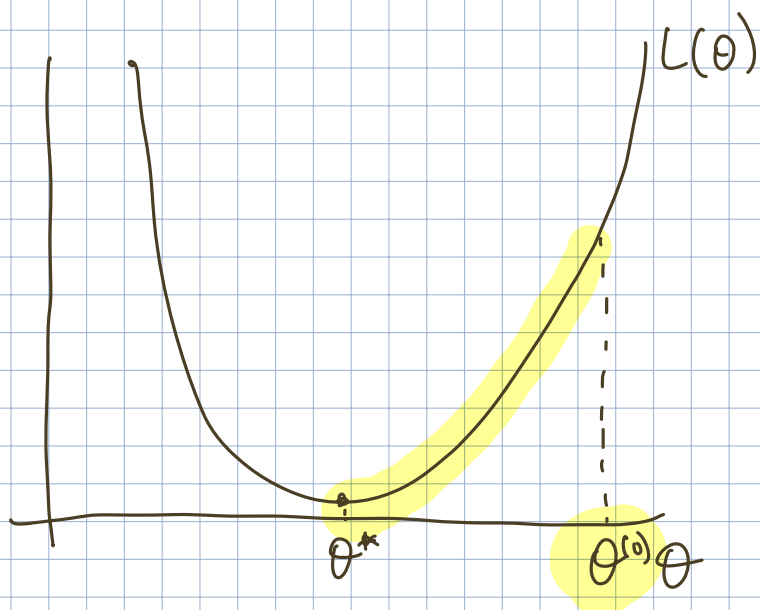
$$\hat{y}_1, \hat{y}_2, \dots, \hat{y}_s$$

función de pérdida:

• Regresión: $\sum_{i=1}^s (y_i - \hat{y}_i)^2 \rightarrow \text{MSE}$

• Clasificación binaria: $\sum_{i=1}^s y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$

\hookrightarrow binary cross entropy.



learning rate.

$$\underline{\theta^{(n)}} = \theta^{(n-1)} - \rho \left(\frac{\partial L}{\partial \theta} \right)$$

$$L = \sum_{i=1}^s (y_i - \hat{y}_i)^2 = y - \hat{y}$$

$$L = y - g^3(g^2(g'(x^t \underline{w^{(1)}}) \underline{w^{(2)}}) \underline{w^{(3)}})$$

$$\frac{\partial L}{\partial w^{(1)}} = g^{3'}(g^2(g'(x^t \underline{w^{(1)}}) \underline{w^{(2)}}) \underline{w^{(3)}}) \cdot g^2(g'(x^t w^{(1)}) w^{(2)})$$

$$\left(\frac{\partial L}{\partial w^{(2)}}\right) = g^{3'}(g^2(g'(x^t \underline{w^{(1)}}) \underline{w^{(2)}}) \underline{w^{(3)}}) \cdot g^{2'}(g'(x^t w^{(1)}) w^{(2)}) \cdot g'(x^t w^{(1)})$$

Regla cadena:

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

$$f'(x) = g'(h(x)) \cdot h'(x)$$

$$L = (y - \hat{y})^2 = (y - g^3(w^{(3)} a^{(2)}))^2 = (y - g^3(w^{(3)} g^2(a^{(1)} w^{(2)})))^2$$

$$\left(\frac{\partial L}{\partial w^{(3)}}\right) = 2 g^{3'}(w^{(3)} a^{(2)}) \cdot a^{(2)}$$

$$\left(\frac{\partial L}{\partial w^{(2)}}\right) = 2 g^{3'}(w^{(3)} g^2(a^{(1)} w^{(2)})) \cdot w^{(3)} g_2'(a^{(1)} w^{(2)}) a^{(1)}$$

función de pérdida con regularización:

• Regresión: $\sum_{i=1}^n (y_i - \hat{y}_i)^2$

→ reg L1

→ reg L2

• Clasificación binaria: $\sum_{i=1}^n y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i) + \lambda \sum_i |w_i|$
 $+ \lambda \sum_i w_i^2$