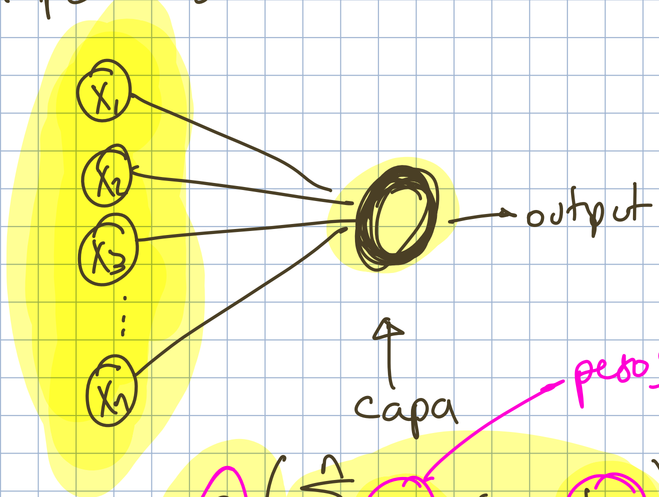
Redes neuronales:

Perceptrón: Red neuronal de 1 capa:

n features:



clasificación binaria / regresión

pesos

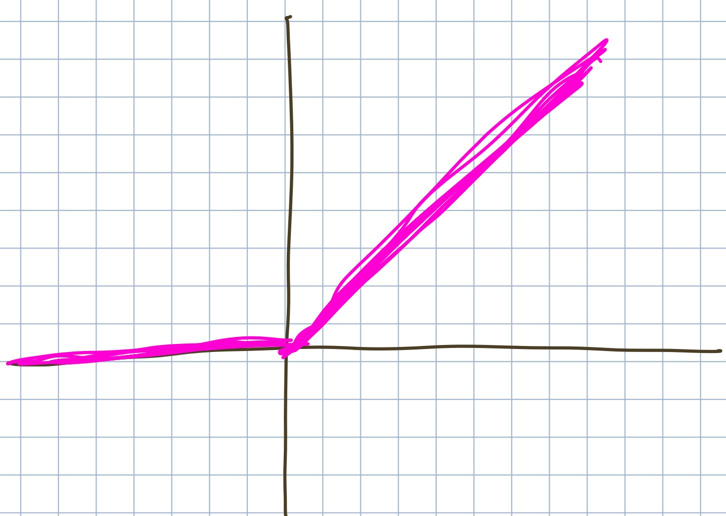
$$g\left(\sum_{i=1}^{n} w_i x_i + b_i\right)$$

sesgo / bias

función de activación.

funciones de actuación:

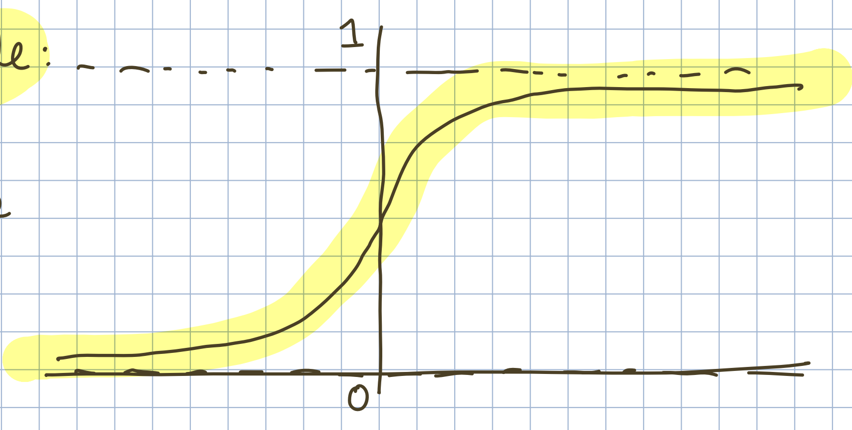
- ReLU: rectified linear unit



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

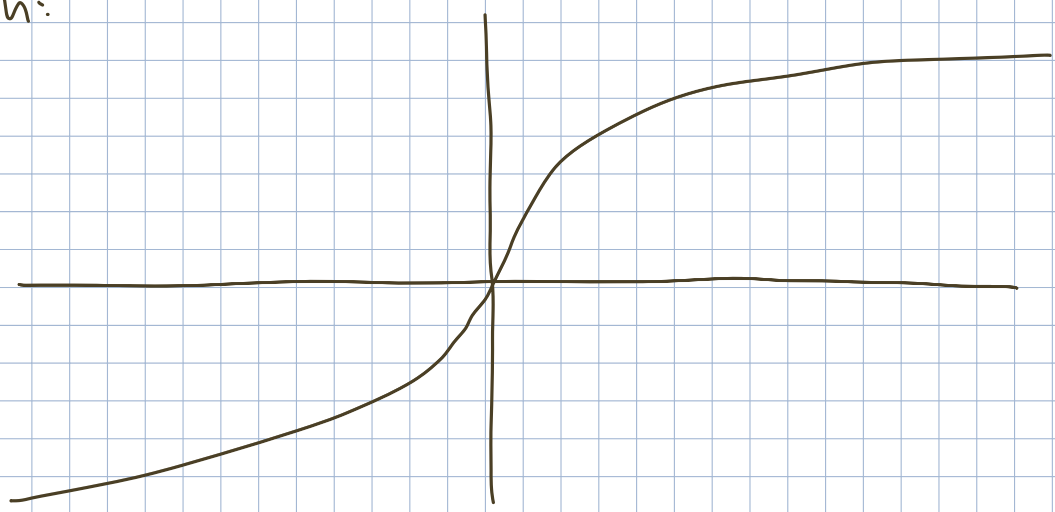
- Sigmoide:

problemas de clasificación



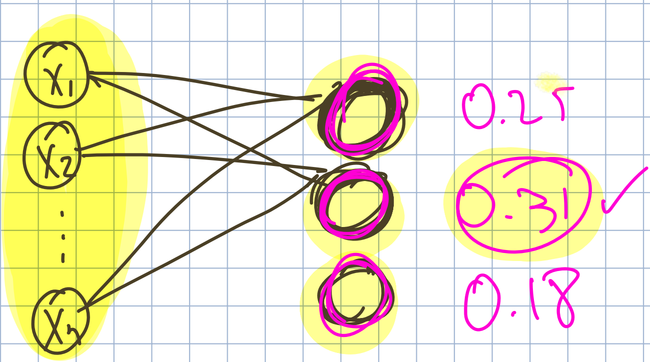
$$f(x) = \frac{1}{1+e^{-x}}$$

• Tanh:



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

Clasificación multiclase:



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

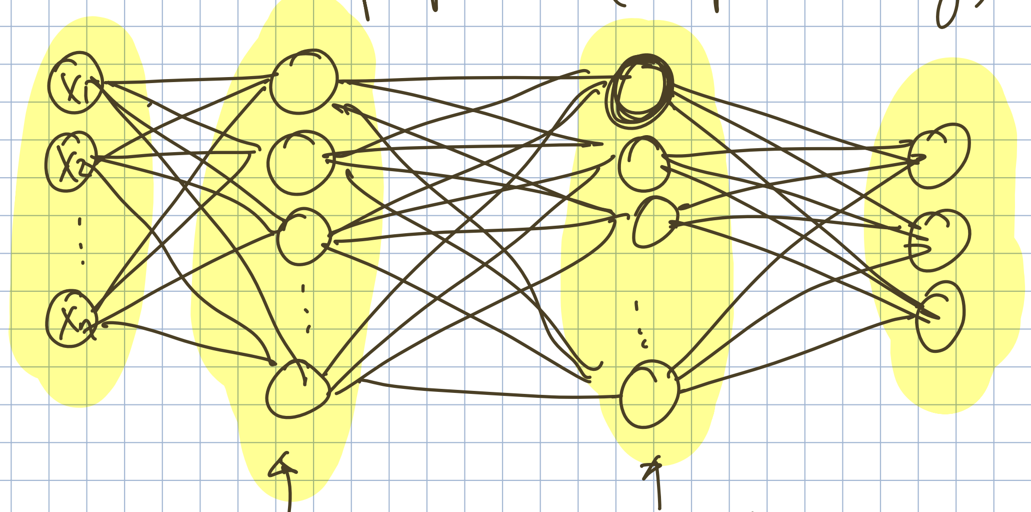
$$\begin{pmatrix} b^1 & w_1^1 & w_2^1 & \cdots & w_n^1 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = X^t W^1$$

$$\begin{pmatrix} b^2 & w_1^2 & w_2^2 & \cdots & w_n^2 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} = X_2^t W^2$$

$$g\begin{pmatrix} b^1 & w_1^1 & w_2^1 & \cdots & w_n^1 \\ b^2 & w_1^2 & w_2^2 & \cdots & w_n^2 \\ b^3 & w_1^3 & w_2^3 & & w_n^3 \end{pmatrix} \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = g(X^t W)$$

Red neuronal profunda (deep learning):

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



capa dimensión m (oculta)

capa de dimensión p (oculta)

$$g^1\left(\sum_{i=1}^{n} w_i^1 x_i + b^1\right)$$

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

$$\vdots$$

$$g^1\left(\sum_{i=1}^{n} w_i^m x_i + b^m\right)$$

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

$$z^{(1)} = a^{(0)t} W^{(1)}$$

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

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

$$a^{(2)} = g^2(a^{(1)t} W^{(2)})$$

$$a^{(3)} = g^3(a^{(2)t} W^{(3)})$$

$$\hat{y} = a^{(3)} = g^3(a^{(2)t} W^{(3)}) = g^3(g^2(a^{(1)t} W^{(2)}) W^{(3)})$$

$$\hat{y} = g^3(g^2(g^1(x^t W^{(1)}) W^{(2)}) W^{(3)})$$

2 etapas:
- feed forward
- back propagation.

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

n-dimensional.

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

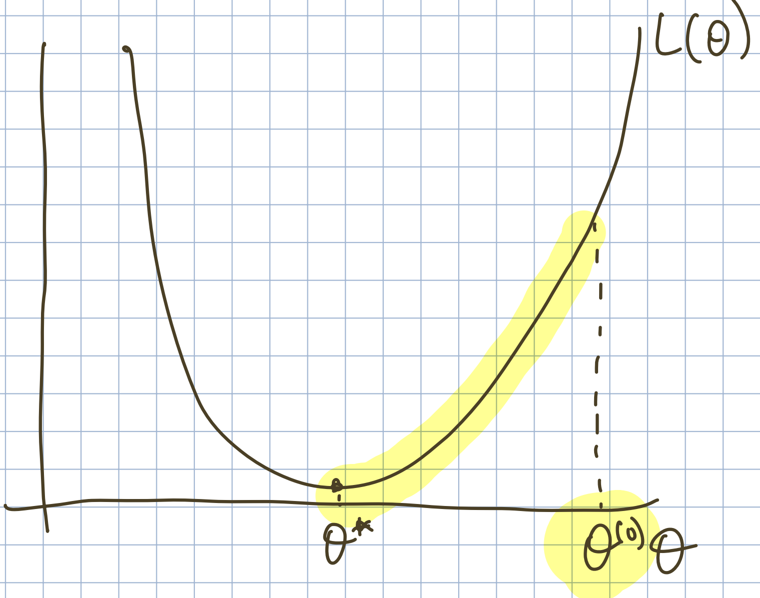
Tomamos cada observación: $1, \dots, s$ y las pasamos por la red neuronal con pesos aleatorios para obtener unas predicciones:

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

Función de pérdida:

- Regresión: $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ → MSE
- Clasificación binaria: $\sum_{i=1}^{n} y_i \ln(\hat{y}_i) + (1-y_i) \ln(1-\hat{y}_i)$

  └ binary cross entropy.



learning rate.

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

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

$$L = y - g^3(g^2(g^1(x^t \underline{w^{(1)}})\underline{w^{(2)}})\underline{w^{(3)}})$$

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

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

$$g^{2'}(g^1(x^t w^{(1)})w^{(2)}) \cdot g^1(x^t w^{(1)})$$

Regla cadena:

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

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

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

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

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

Fucción de pérdida con regularización:

• Regresión: $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ $+\lambda \sum_j |w_j|$ $\longrightarrow$ reg L1

$+\lambda \sum_j w_j^2$ $\longrightarrow$ reg L2

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

$+\lambda \sum_j w_j^2$