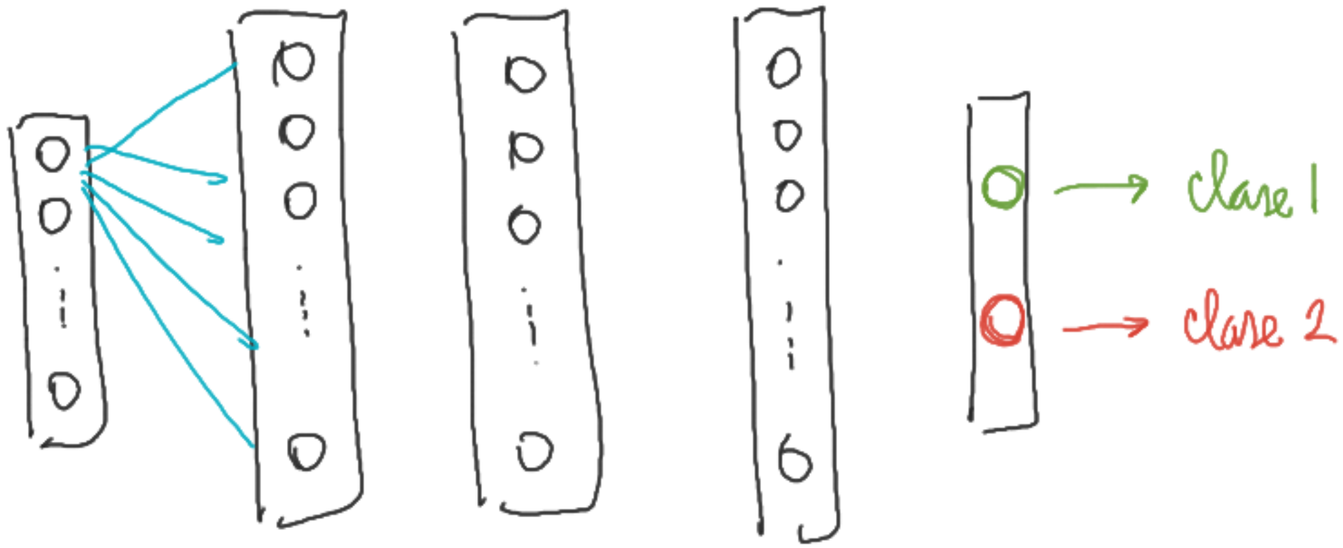
$$\text{ReLU}: \mathbb{R} \longrightarrow \mathbb{R}$$

$$\text{ReLU}(x) = \begin{cases} 0; & x < 0 \\ x; & x \geq 0 \end{cases} = \max(0, x).$$



Clasificación: Última capa #neuronas = # categorías

activación = softmax.

$$\text{Softmax}(x_1, x_2, \cdots, x_k)$$

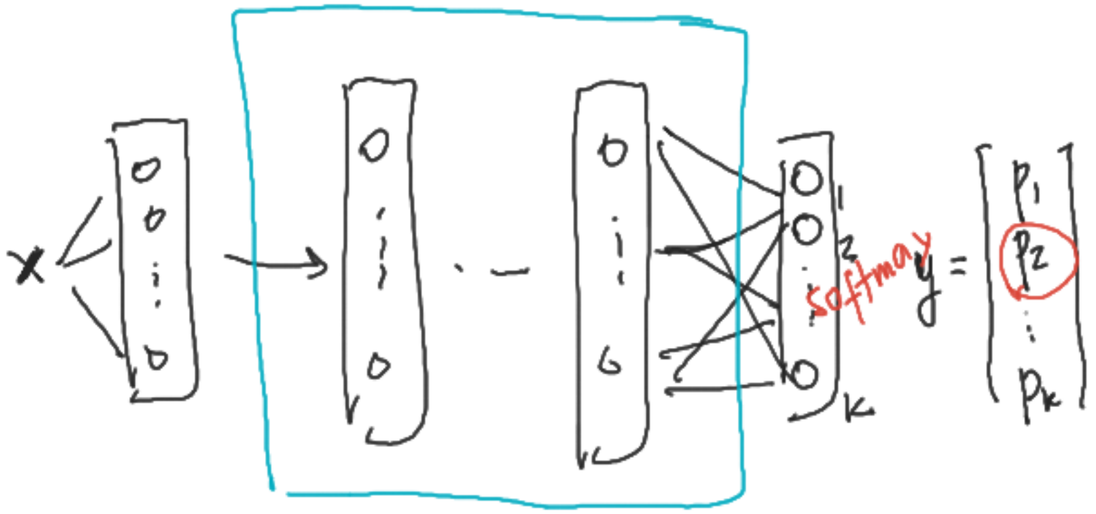
$$= \left(\frac{e^{x_1}}{\sum_{i=1}^{k} e^{x_i}}, \frac{e^{x_2}}{\sum_{i=1}^{k} e^{x_i}}, \cdots, \frac{e^{x_k}}{\sum_{i=1}^{n} e^{x_i}} \right)$$

$$\Rightarrow \sum_{i=1}^{k} y_i = 1$$

y_1 y_2 $\cdots$ y_k

$P(x \in C_1)$ $P(x \in C_2)$ $\cdots$ $P(x \in C_k)$

max

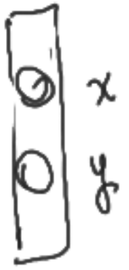


* se clasifica como 2

Regresión:

neuronas = dimensión del dato a estimar

activación = identidad



$y \in \mathbb{R}^1$

$(x,y) \in \mathbb{R}^2$

Descenso Gradiente:

$f: \mathbb{R}^n \longrightarrow \mathbb{R}$

Queremos hallar $x^* \in \mathbb{R}^n$ que minimiza $f(x)$.

x_0 cualquiera

$$x_{n+1} = x_n - \alpha \nabla f(x_n)$$

$W^{(k)}$ = matriz de pesos en la capa k de la red

$= [w_{ij}^{(k)}]$

fijos

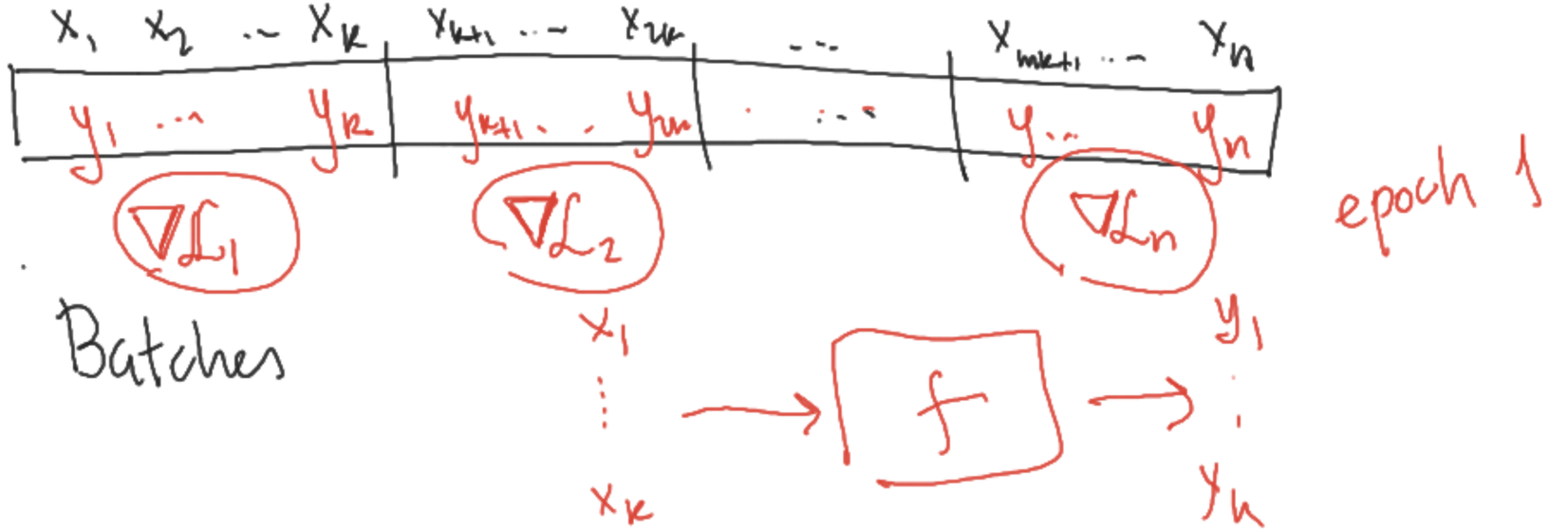
loss function

$$L(y_i) = L(x_i, w_{ij}^{(k)})$$

$$y = \varphi_n\left(W_2 \varphi_1\left(W_1 x + b_1 \right) + b_2 \right) \dots$$

Backtracking

$$w_{ij}^{(k),n+1} = w_{ij}^{(k),n} - \alpha \nabla_{w_{ij}} L\left(x_i, w_{ij}^{(k),n}\right)$$



Batches ó el batch_size

hasta que el descenso gradiente converge.

algoritmo

α tamaño paso = learning rate # epochs