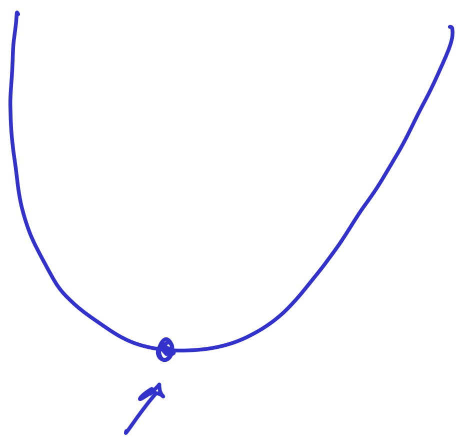
REGRESIÓN LINEAL 1D:

$(x_1, t_1), (x_2, t_2) \dots (x_N, t_N)$ $\qquad y = \underline{w}x + \underline{b}$

$$y_i = w x_i + b$$

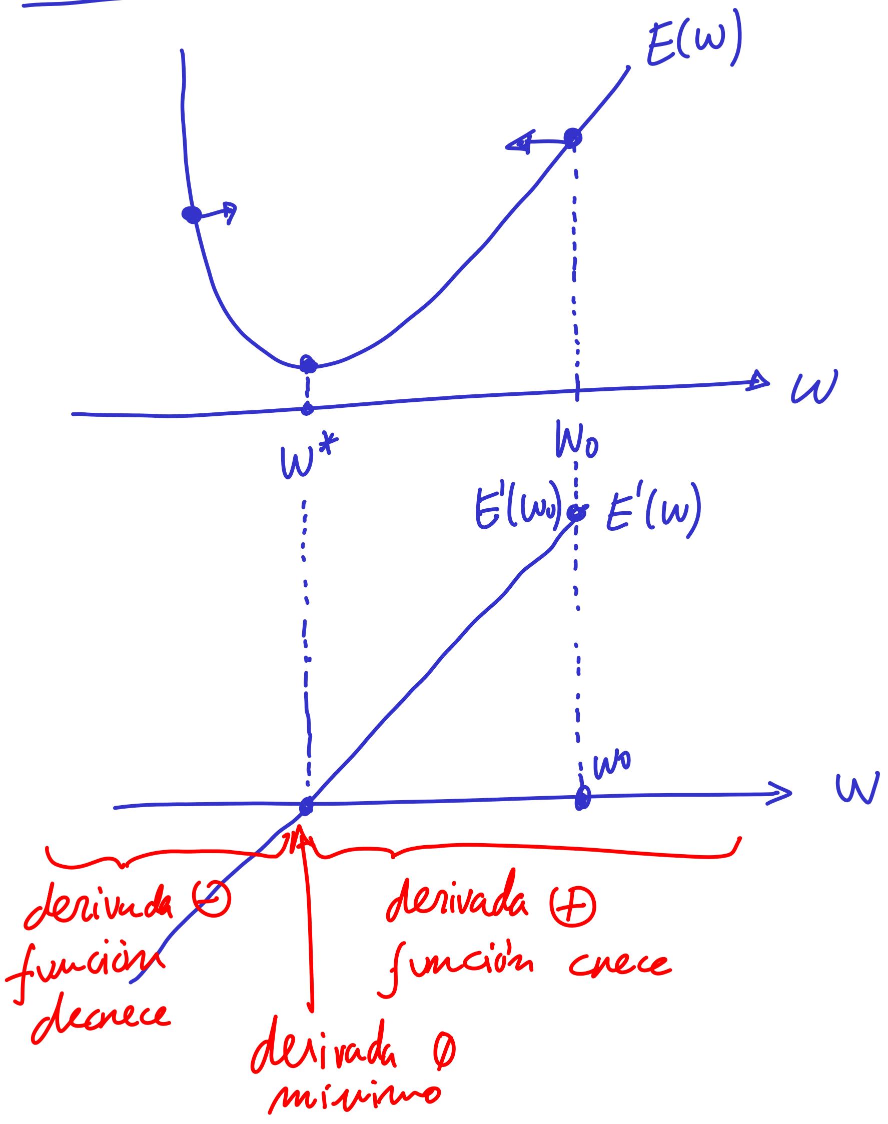
$$E = \frac{1}{2} \sum_{i=1}^{N} (y_i - t_i)^2$$



Planteamiento habitual:

$$\left.\begin{aligned} \frac{\partial E}{\partial w} &= 0 \\ \frac{\partial E}{\partial b} &= 0 \end{aligned}\right\} \text{resuelvo} \rightarrow \boxed{\text{Solución}}$$

DESCENSO POR GRADIENTE:



Si función crece $\leftarrow$

si función decrece $\rightarrow$

siempre > 0

Si $E'(w_0) > 0 \Rightarrow w_1 = w_0 - \Delta$

Si $E'(w_0) < 0 \Rightarrow w_1 = w_0 + \Delta$

W_0

> 0, learning rate

$$\boxed{w_1 = w_0 - \eta \cdot E'(w_0)}$$

DESCENSO POR GRADIENTE

$$\boxed{w_t = w_{t-1} - \eta \cdot E'(w_{t-1})}$$

PARA R. LINEAL:

$$\boxed{\begin{aligned} w_t &= w_{t-1} - \eta \cdot \left.\frac{\partial E}{\partial w}\right|_{w_{t-1}} \\ b_t &= b_{t-1} - \eta \cdot \left.\frac{\partial E}{\partial b}\right|_{b_{t-1}} \end{aligned}}$$

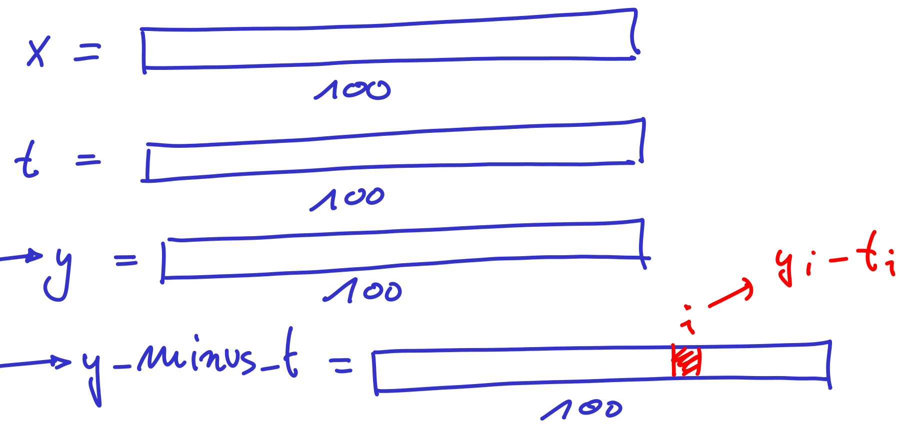
$$\boxed{\begin{aligned} w_t &= w_{t-1} - \eta \cdot \sum_{i=1}^{N} (y_i - t_i) \cdot x_i \\ b_t &= b_{t-1} - \eta \cdot \sum_{i=1}^{N} (y_i - t_i) \end{aligned}}$$

D.GRAD. REGR. LINEAL

$$E = \frac{1}{2} \sum_{i=1}^{N} (y_i - t_i)^2 \qquad y_i = w x_i + b$$

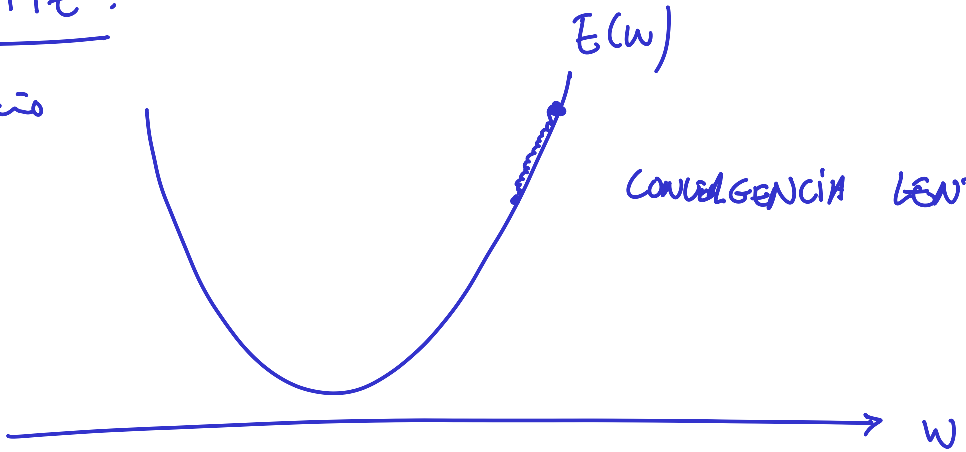
$$\frac{\partial E}{\partial b} = \sum_{i=1}^{N} (y_i - t_i) \cdot \underbrace{\frac{\partial y_i}{\partial b}}_{1} \qquad \frac{\partial E}{\partial w} = \sum_{i=1}^{N} (y_i - t_i) \underbrace{\frac{\partial y_i}{\partial w}}_{x_i} = \sum_{i=1}^{N} (y_i - t_i) x_i$$

```python
num_iters = 8 # number of iterations
eta = 0.0001  # learning rate
for i in range(num_iters):
    y = w*x + b
    y_minus_t = y - t
    dw = np.sum(y_minus_t*x)
    db = np.sum(y_minus_t)
    w -= eta*dw
    b -= eta*db
```



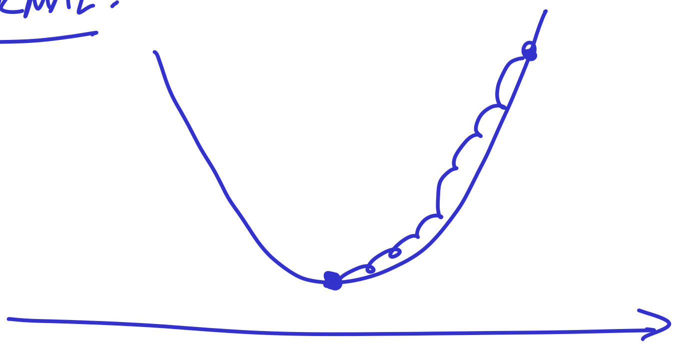
LEARNING RATE:

η pequeño

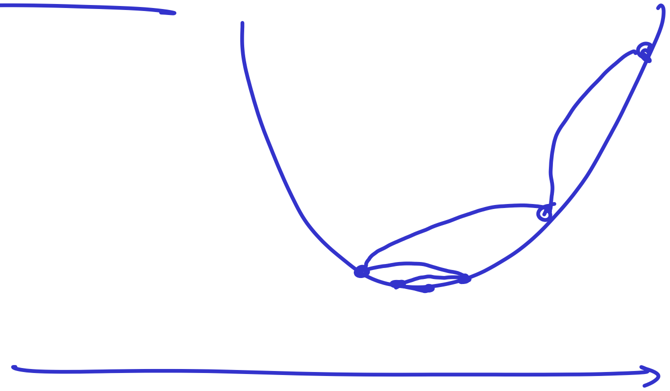


CONVERGENCIA LENTA

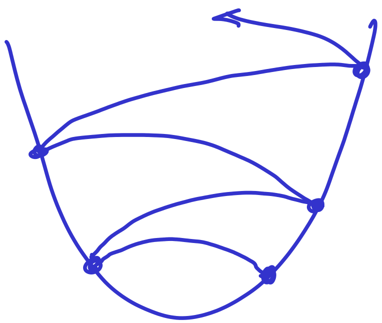
η NORMAL:



η ALTO



η MUY ALTO



DIVERGE ...

REGR. LINEAL d DIMENSIONES:

$$\{(\vec{x}_1, t_1), (\vec{x}_2, t_2), \ldots, (\vec{x}_N, t_N)\}$$

$\vec{x}_i \in \mathbb{R}^d \quad \vec{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \qquad \vec{w} = (w_1, w_2, \ldots, w_d)^T$

$t_i \in \mathbb{R}$

$$y_i = f(\vec{x}_i) = \vec{w}^T \cdot \vec{x}_i + b = \sum_{j=1}^{d} \underline{w_j} x_{ij} + \underline{b} \qquad \Big\| \qquad E = \frac{1}{2} \sum_{i=1}^{N} (y_i - t_i)^2$$

$$b \leftarrow b - \eta \cdot \frac{\partial E}{\partial b} \rightsquigarrow \frac{\partial E}{\partial b} = \sum_{i=1}^{N} (y_i - t_i)$$

$$w_j \leftarrow w_j - \eta \frac{\partial E}{\partial w_j} \longrightarrow \frac{\partial E}{\partial w_j} = \sum_{i=1}^{N} (y_i - t_i) \cdot x_{ij}$$

$$\vec{\nabla}_{\vec{w}} E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_d} \right)^T$$

$$\underbrace{\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}}_{\vec{w}} \leftarrow \underbrace{\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}}_{\vec{w}} - \eta \cdot \underbrace{\begin{pmatrix} \partial E / \partial w_1 \\ \partial E / \partial w_2 \\ \vdots \\ \partial E / \partial w_d \end{pmatrix}}_{\nabla_{\vec{w}} E}$$

$$\boxed{\vec{w} \leftarrow \vec{w} - \eta \cdot \vec{\nabla}_{\vec{w}} E}$$

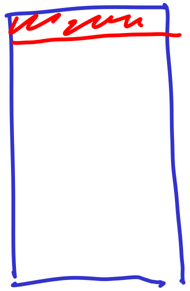
$$\downarrow$$

$$\boxed{\vec{w} \leftarrow \vec{w} - \eta \cdot \sum_{i=1}^{N} (y_i - t_i) \cdot \vec{x}_i}$$

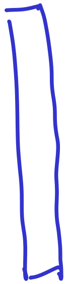
$$\vec{\nabla}_{\vec{w}} E = \sum_{i=1}^{N} (y_i - t_i) \cdot \underbrace{\begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{pmatrix}}_{\vec{x}_i}$$

$$\rightarrow \sum_{i=1}^{N} (y_i - t_i)\, \vec{x}_i$$

PROGRAMACIÓN:

$X =$ 

$N \times d$

$t =$ 

$N \times 1$

$w =$ 

$1 \times d$

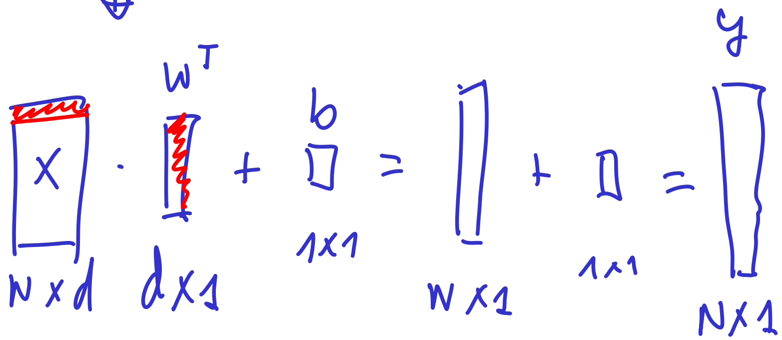
$b =$ 

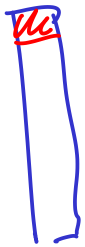
1×1

$$y = \underline{\vec{w}^t \cdot \vec{x} + b}$$

$$y = np.dot(X, w^T) + b$$

$$\downarrow$$



$y - t =$ 

$N \times 1$

```
b -= η * np.sum(y - t)

w -= η * np.sum((y - t) * x, axis = 0)    //    w -= η · (y - t)ᵀ @ x
```

keepdims = True

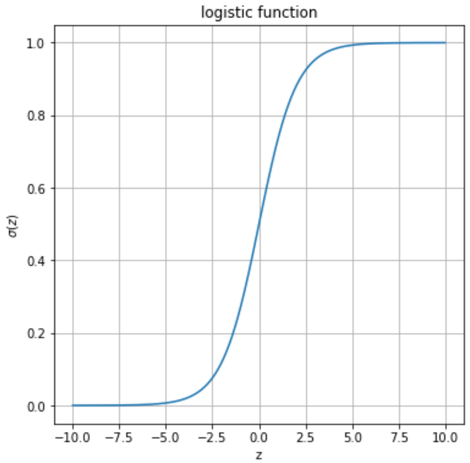
REGRESIÓN LOGÍSTICA:

$$\{(\vec{x}_1, t_1), (\vec{x}_2, t_2), \dots, (\vec{x}_N, t_N)\}$$

$$\vec{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^T \in \mathbb{R}^d \qquad \vec{w} = (w_1, w_2, \dots, w_d)^T$$

$$t_i = \{0, 1\} \qquad b$$

$$y_i = \sigma(\underbrace{\vec{w}^T \cdot \vec{x}_i + b}_{z_i})$$



$$y_i = PROB(t=1 \mid \vec{x}_i)$$

$$1 - y_i = PROB(t=0 \mid \vec{x}_i)$$

MAXIMIZAR LA PROB. QUE EL MODELO ASIGNA A LAS CLASES REALES

VEROSIMILITUD / LIKELIHOOD

$$\boxed{l = \prod_{i=1}^{N} P(t=t_i \mid \vec{x}_i)} \xrightarrow{\log} \boxed{L = \log l = \sum_{i=1}^{N} \log P(t=t_i \mid \vec{x}_i)}$$

MAXIMIZAR

VEROSIMILITUD LOGARÍTMICA

$$P(t=t_i|\overline{X}_i) = \begin{cases} y_i & \text{si } t_i = 1 \\ 1-y_i & \text{si } t_i = \emptyset \end{cases}$$

PROB. DEL MODELO
PARA LA CLASE REAL t_i

$$L = \sum_{i=1}^{N} \log \begin{bmatrix} y_i & \text{si } t_i = 1 \\ 1-y_i & \text{si } t_i = \emptyset \end{bmatrix} = \dots$$

$$\dots = \sum_{i=1}^{N} \begin{bmatrix} \log y_i & \text{si } t_i = 1 \\ \log(1-y_i) & \text{si } t_i = 0 \end{bmatrix} = \sum_{i=1}^{N} \left(t_i \cdot \log y_i + (1-t_i) \log (1-y_i) \right)$$

MAXIMIZAR

$$\boxed{XE = -L = -\sum_{i=1}^{N} \left(t_i \cdot \log y_i + (1-t_i) \log (1-y_i) \right)}$$

CROSS-ENTROPY $\rightarrow$ MINIMIZAMOS

DESCENSO POR GRADIENTE:

$$\boxed{\begin{aligned} b &\leftarrow b - \eta \cdot \frac{\partial XE}{\partial b} \\ \vec{w} &\leftarrow \overline{w} - \eta \cdot \overrightarrow{\nabla_{\overline{w}} XE} \end{aligned}}$$

$$z_i = \vec{w}^T \vec{x}_i + b = \sum_{j=1}^{d} w_j x_{ij} + b$$

$$y_i = \sigma(z_i)$$

$$\frac{\partial XE}{\partial b} = \frac{\partial XE}{\partial y_i} \cdot \frac{\partial y_i}{\partial b} = \boxed{\frac{\partial XE}{\partial y_i}} \cdot \boxed{\frac{\partial y_i}{\partial z_i}} \cdot \underbrace{\boxed{\frac{\partial z_i}{\partial b}}}_{=1}$$

$$\frac{\partial XE}{\partial w_j} = \frac{\partial XE}{\partial y_i} \cdot \frac{\partial y_i}{\partial w_j} = \boxed{\frac{\partial XE}{\partial y_i}} \cdot \boxed{\frac{\partial y_i}{\partial z_i}} \cdot \underbrace{\boxed{\frac{\partial z_i}{\partial w_j}}}_{=x_{ij}}$$

$$XE = -\sum_{i=1}^{N} \left(t_i \log y_i + (1-t_i) \log (1-y_i) \right)$$

$$\frac{\partial XE}{\partial y_i} = -\sum_{i=1}^{N} \left[\underbrace{\frac{t_i}{y_i} - \frac{1-t_i}{1-y_i}} \right] = \boxed{\sum_{i=1}^{N} \frac{y_i - t_i}{y_i(1-y_i)}}$$

$$\frac{t_i}{y_i} - \frac{1-t_i}{1-y_i} = \frac{t_i - \cancel{t_i y_i} - y_i + \cancel{t_i y_i}}{y_i(1-y_i)} = \frac{t_i - y_i}{y_i(1-y_i)}$$

$$y_i = \sigma(z_i) = \frac{1}{1+e^{-z_i}}$$

$$\frac{\partial y_i}{\partial z_i} = \frac{+1}{(1+e^{-z_i})^2} \cdot e^{-z_i} = \frac{1 + e^{-z_i} - 1}{(1+e^{-z_i})(1+e^{-z_i})} =$$

$$= \underbrace{\frac{1}{1+e^{-z_i}}}_{\sigma(z_i)} - \underbrace{\frac{1}{(1+e^{-z_i})(1+e^{-z_i})}}_{\sigma(z_i)^2} = \sigma(z_i) - \sigma(z_i)^2 = \sigma(z_i)\left(1-\sigma(z_i)\right) = \boxed{y_i(1-y_i)}$$

$$\boxed{\frac{\partial XE}{\partial b} = \sum_{i=1}^{N} (y_i - t_i)}$$

$$\frac{\partial XE}{\partial w_j} = \sum_{i=1}^{N} (y_i - t_i) \cdot x_{ij} \implies \boxed{\nabla_{\vec{w}} XE = \sum_{i=1}^{N} (y_i - t_i) \cdot \vec{x}_i}$$

$$\boxed{\begin{aligned} b &\leftarrow b - \eta \cdot \sum_{i=1}^{N} (y_i - t_i) \\ \vec{w} &\leftarrow \vec{w} - \eta \cdot \sum_{i=1}^{N} (y_i - t_i) \cdot \vec{x}_i \end{aligned}}$$

DESCENSO POR GRADIENTE
PARA REGR. LOGISTICA

REGR. LINEAL

```
num_iters = 8 # number of iterations
eta = 0.0001  # learning rate
for i in range(num_iters):
    y = w*x + b
    y_minus_t = y - t
    dw = np.sum(y_minus_t*x)
    db = np.sum(y_minus_t)
    w -= eta*dw
    b -= eta*db
```

REGR. LOGISTICA

```
num_iters = 1000 # number of iterations
eta = 0.0002     # learning rate
for i in range(num_iters):
    z = w*x + b
    y = 1.0/(1.0 + np.exp(-z))
    y_minus_t = y - t
    dw = np.sum(y_minus_t*x)
    db = np.sum(y_minus_t)
    w -= eta*dw
    b -= eta*db
```