

si η es chico ($\Delta \bar{W} = -\eta \frac{\partial E}{\partial \bar{W}}$) E disminuye

algoritmo de propagación de errores

- inicializan los pesos \bar{W} "ÉPOCAS"
- loop en pasos de aprendizaje (τ fijo o hasta que pare algo)

- $\Delta \bar{W} = 0$
- loop entre $\mu = 1, \dots, P$
 - $\bar{x}_1 \leftarrow \bar{x}^\mu$ (elimo x^μ con \bar{x}_1)
 - loop sobre capas $m = 1, \dots, M-1$
 - $\bar{h}_m = \bar{W} \cdot \bar{x}_m$ \rightarrow input de las neuronas en la capa m
 - $\bar{x}_{m+1} = g(\bar{h}_m)$ \rightarrow salida de la capa m
 - $\bar{o}^\mu \leftarrow \bar{x}_M$
 - $\delta_m = -\frac{\partial E}{\partial \bar{o}^\mu} g'(\bar{h}_m)$
 - loop sobre capas ($m = m, \dots, 2 \rightarrow$ para atrás)
 - $\delta_{m-1} = [\bar{W}^T \delta_m] \otimes g'(\bar{h}_{m-1})$
 - loop sobre capas $m = 1, \dots, M-1$
 - $(*) \Delta \bar{W} += \eta \delta_{m+1} \cdot \bar{x}_m$ para complicada \hookrightarrow hay que guardar los x

hay que guardar x, h, δ, W y ΔW

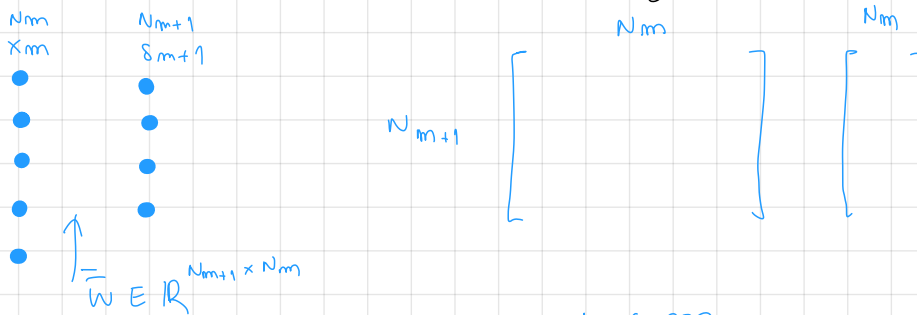
$$\bar{W} = \bar{W} + \Delta \bar{W}$$

(se puede hacer que si

CONTROL DE DETERMINACION

$\Delta W < w_0 \rightarrow$ break)

Por componentes está en el Hertz



para cada espacio entre capas hay una matriz

$$\Delta W_{ij} += \eta \delta_{m+1,i} x_{m,j}$$

$j = 1, \dots, N_m$
 $i = 1, \dots, N_{m+1}$

uno puede alterar los umbrales poniendo una neurona adicional

$$x_{m,i} = g \left(\sum_{j=1}^m w_{ij} x_{m,j} - \theta_i \right)$$

podemos poner una neurona con una salida constante, que nos de estos parámetros
↓
esta neurona estaría en la capa de entrada y le mandaría señales a todos)

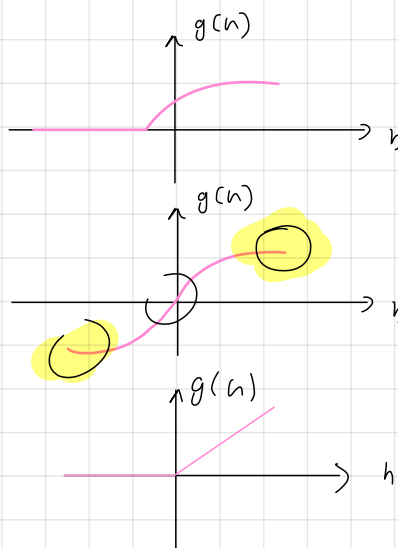
no hay un teorema que nos garantice la convergencia

Heurística

inicialización de los pesos

si los pesos son muy grandes

↓
derivada pequeña, g' logio
 $\Delta \bar{w}$ pequeño ($|\Delta \bar{w}| < 1$)
↓
modificaciones muy pequeñas



otro típico

sigmoide

tgh

- se quiere que $|\bar{w} \cdot \bar{x}_0| \sim \underbrace{O(1)}_{\text{orden 1}}$

↓
normalizar los inputs

(muchos de pto, se puede poner con media 0 y varianza $1/N \rightarrow$ no se pierde info)

↓
esto por defecto en varios paquetes)

N : dim de la capa de entrada

h va a tener varianza

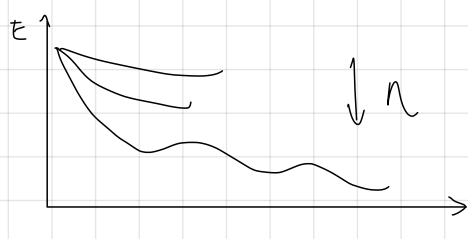
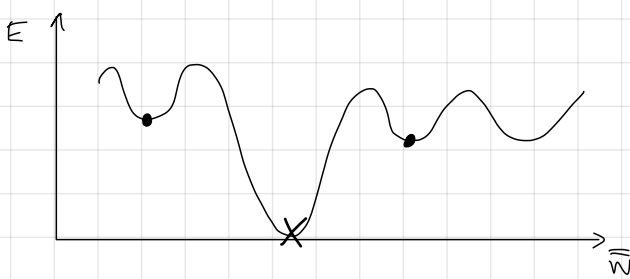


grafico lo que se quiere minimizar para los distintos valores de N

tende a fallar por ser un método local

↓
el método puede quedar atrapado en mínimos locales o pto silla no deseado)



Para ver si nos quedamos en un min local:

- se pueden probar muchos CF distintos

- o bien modifican el método de aprendizaje

Podemos pensar que se trata de una pelota \Rightarrow si tiene impulso atravesará los mínimos locales

$$E(\bar{w}) \quad \bar{g} = \bar{\nabla}_{\bar{w}} E(\bar{w}) \rightarrow \text{MOMENTO}$$

$$\bar{V}_T = \beta \bar{V}_{T-1} + (1-\beta) \bar{g}_T$$

\downarrow
promedio del gradiente
con una corte de tiempo
de $1/\beta$

o

o

\leftarrow

$$\Delta W = -\eta \bar{V}_T$$

guarda en memoria
la dirección en la que
es el movimiento

otra forma

NESTERON

$$\bar{V}_{T+1} = \beta \bar{V}_{T-1} + \eta \nabla E(\bar{w} - \beta \bar{V}_{T-1})$$

$$\Delta \bar{w} = -\bar{V}_T$$

ADAM

$$\bar{m}_t = \beta \bar{m}_{t-1} + (1-\beta) \bar{g}_t$$

$$V_t = \beta V_t + (1-\beta) |g_t|^2$$

$$\Delta \bar{w} = -\eta \frac{\bar{m}_t}{\sqrt{V_t} + \epsilon}$$

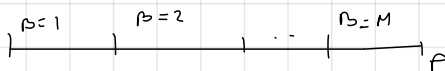
stochastic gradient descent

$\Delta \bar{w} \rightarrow$ suma sobre $\mathcal{H} = 1, \dots, P$

SGD (parámetros)

nº de
elementos
de un
Batch)

Podemos dividir el conjunto en subconjuntos



$$\Delta W = \sum_{\mathcal{H}=1}^P \Delta \bar{w}^{\mathcal{H}}$$

$$= \sum_{b=1}^B \left[\sum_{\mathcal{H}_b=1}^{P/B} \Delta \bar{w}^{\mathcal{H}_b} B \right] \frac{1}{B}$$

cada uno de estos valores
es una variable aleatoria
que alterna entre el vector ΔW