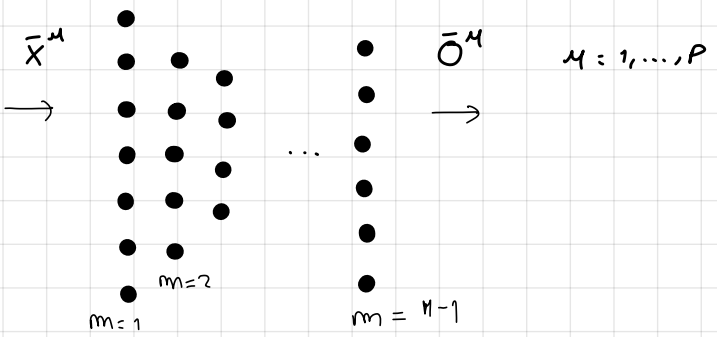


Si η es chico $\left(\Delta \bar{w} = -\eta \frac{\partial E}{\partial \bar{w}}\right)$ E disminuye

algoritmo de propagación de errores

- inicializar los pesos $\bar{\bar{w}}$ — "ÉPOCAS"
- loop en pasos de aprendizaje (τ fijo o hasta que pase algo)
  - $\Delta \bar{\bar{w}} = 0$
  - loop entre $\mu = 1, \dots, P$
    - $\bar{x}_1 \leftarrow \bar{x}^\mu$ (lleno x^μ con $\bar{x}_1$)
    - loop sobre capas $m = 1, \dots, M-1$
      - $\bar{h}_m = \bar{\bar{w}} \cdot \bar{x}_m$ → input de las neuronas en la capa m
      - $x_{m+1} = g(\bar{h}_m)$ → 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 ($m = m, \dots, 2$ → para atrás)
      - $\bar{\delta}_{m-1} = [\bar{\bar{w}}^T \bar{\delta}_m] \otimes g'(\bar{h}_{m-1})$
    - loop sobre capas $m = 1, \dots, m-1$
      - (*) $\Delta \bar{\bar{w}} \mathrel{+}= \eta \, \bar{\delta}_{m+1} \cdot \bar{x}_m$

Hay que guardar x, h, δ, w y Δw

para complicados sí hay que guardar las x

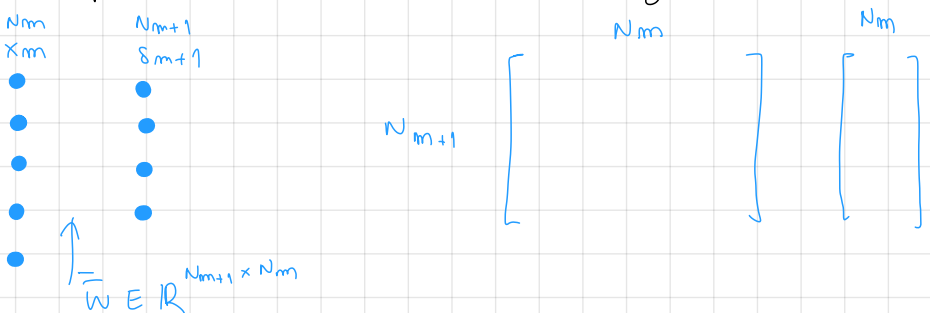
- $\bar{\bar{w}} = \bar{\bar{w}} + \Delta \bar{\bar{w}}$

CONTROL DE DETERMINACIÓN

(se puede hacer que si $\Delta w < w_0$ → break)

¿Cuál es el objetivo de w0?

Por componentes está en el Hertz



para cada espacio entre capas hay una matriz

$$\Delta w_{ij} \mathrel{+}= \eta \, \delta_{m+1,i} \, x_{mj}$$

$j = 1, \dots, N_m$

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

uno puede absorber los umbrales poniendo una neurona adicional

$$x_{m,i} = g\left(\sum_{j=1}^{m} W_{ij}\, x_{m,j} - b_0\right)$$

podemos poner una neurona con una salida cte, que nos de este parámetro

esta neurona estaría en la capa de entrada y le mandaría señales a todas)

no hay un teorema que nos garantice la convergencia

ntos típicos

sigmoide

tg h

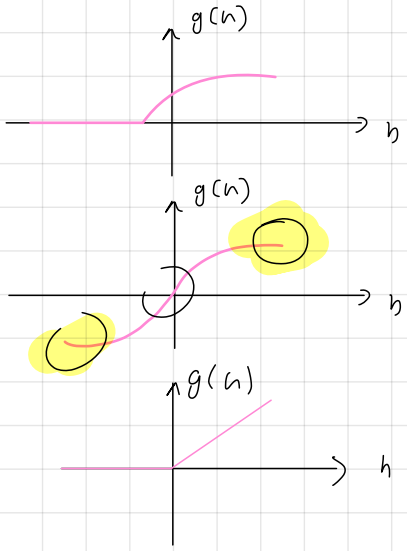
Heurística

inicialización de los pesos

Si los pesos son muy grandes

↓ derivadas pequeñas, g' baja $\Delta \bar{W}$ pequeño ($|\Delta \bar{W}| << 1$)

↓ modificaciones muy pequeñas



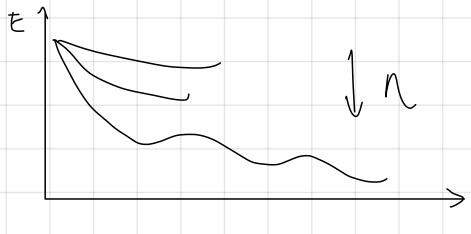
- se quiere que $|\bar{W} \cdot \bar{x}_0| \sim \underbrace{O(1)}_{\text{orden 1}}$

↓ normalizar los inputs (nube de ptos, se puede poner con media 0 y varianza $1/N$ → no se pierde info)

N: dim de la capa de entrada

h va a tener varianza

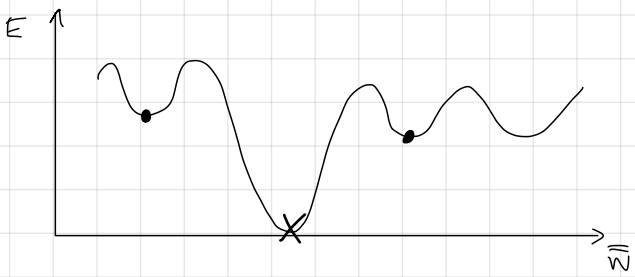
está por defecto en varios paquetes)



grafican lo que se quiere minimizar para distintos valores de η

tiende a fallar por ser un método local

↓ el método puede quedar atrapado en mínimos locales o puntos silla no deseados



Para ver si nos quedamos en un mín local:

- se prueban muchos CI distintos
- o bien modifican el método de aprendizaje

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

$E(\bar{w})$

$\bar{g} = \bar{\nabla}_w E(\bar{w})$ → MOMENTO

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

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

ACOL (K)

$\Delta w = -\eta \bar{V}_\tau$

guarda en memoria la dirección en la que es el movimiento

otra forma

NESTEROV

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

$\Delta \bar{w} = -\bar{V}_\tau$

ADAM

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

$V_t = \beta V_t + (1-\beta_t)|g_\tau|^2$

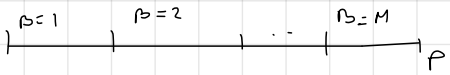
$\Delta \bar{w} = -\eta \frac{\bar{m}_t}{\sqrt{V_\tau} + \varepsilon}$

Stocastic gradient descent

SGD (parámetro. nº de elementos de un Batch)

$\Delta \bar{w}$ → suma sobre $\mu = 1, \ldots, P$

Podemos dividir el conjunto en subconjuntos



$$\Delta w = \sum_{\mu=1}^{P} \Delta \bar{w}^\mu$$

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

cada uno de estos valores es una variable aleatoria que oscila entre el verdadero Δw