

$n_{\text{input}} + 1 \rightarrow$ è il bias del layer dopo
 n- $\text{input}_i + 1$ è che in matrice lui
 w_{j2i} passi come:

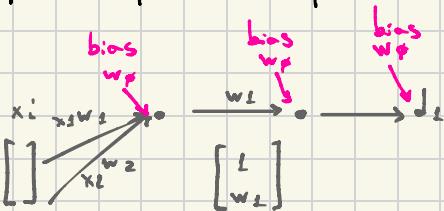
BIAS
NEL
FORWARD

$w_{j2i} \Rightarrow n_{\text{input}} + \text{bias} \times n_{\text{hidden}}$
 quindi avrai sempre un valore in più.

Vi ricordate di **BIAS**, in tutti gli hidden layer

tip. $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \times \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} + \text{bias} = \begin{bmatrix} o_1 & o_2 & o_3 \end{bmatrix}$

poi moltiplico in forward il vettore x_i



, ma noti che il bias =

$$(\sum_i w_{ji} + x_i) + w_{ip}$$

quindi un po' e poi il
 solo numero bias



allora ho tutte le matrici da passare come:

$$\begin{bmatrix} w_{j2i} \\ \vdots \end{bmatrix} \quad \begin{bmatrix} w_{j2j2} \\ \vdots \end{bmatrix} \quad \begin{bmatrix} w_{kj2} \\ \vdots \end{bmatrix}$$

che sono considerate nel forward e nell'backward pass,
 perciò vorrei modificare le questioni

forward

$$\underbrace{\begin{bmatrix} \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix}}_{k\text{-unit}} \Rightarrow \text{np. dot} \left(\text{vettore } x_i, w_{ji} [i:, j\text{-unit}] \right) + w_{ji} [i] [j\text{-unit}]$$

che mi restituisce il vettore risultato relativo al layer.
Dopo questa fase ho il risultato della rete che conosco
trovando l'errore e propagandolo all'interno, come si
comporta il bias in questo caso?

BIDS NELLA BACKPROP.

procedimento simile
per δ_j

tranne per
una cosa *

$$\delta_k$$

Questo è semplice.

il vettore k-th unit

$$(a_k - o_k)$$

$$\Rightarrow w_{kj}^{\text{new}} = w_{kj} + \eta \cdot \delta_k \cdot o_j$$

Pero' w_{kj} ha come prima
riga i bias = il primo elemento

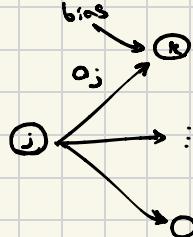
il vettore w_{kj}^{new} , secondo k, non

ha un collegamento con un uniti nel layer precedente.

Quindi quando calcolo Δw_{kj} , che contiene i bias dell'output layer
ho: $\Delta p w_{kj} = \delta_k \cdot o_j$ ma per $\Delta p w_{kj}$?

k-unit

$$\begin{bmatrix} \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix}$$



$$o_j = f_j(\text{net}_j)$$

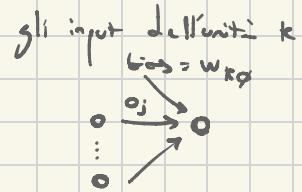
Vediamo la formula:

Il gradiente rispetto a $\Delta_p w_{kp} =$

$$-\frac{\partial E_p}{\partial \text{net}_k} \cdot \frac{\partial \text{net}_k}{\partial w_{kp}} = \dots$$

\downarrow

$$\frac{\partial (\sum_j w_{kj} o_j) + w_{kp}}{\partial w_{kp}} = 1$$



BiAS NELL'UPDATE

Quindi nel caso δ_k ho:

WEIGHTS

$$\text{non bias} \rightarrow \Delta_p w_{kj} = \delta_k \cdot o_j \Rightarrow w_{kj}^{\text{new}} = w_{kj}^{\text{old}} + \eta \cdot \delta_k \cdot o_j$$
$$\text{bias} \rightarrow \Delta_p w_{kp} = \delta_k \cdot 1 \Rightarrow w_{kp}^{\text{new}} = w_{kp}^{\text{old}} + \eta \cdot \delta_k \cdot 1$$

e sempre per similitudine con δ_j ho:

$$\text{non bias} \rightarrow \dots \Rightarrow w_{ji}^{\text{new}} = w_{ji}^{\text{old}} + \eta \cdot \delta_j \cdot o_i$$

$$\text{bias} \rightarrow \dots \Rightarrow w_{j\phi}^{\text{new}} = w_{j\phi}^{\text{old}} + \eta \cdot \delta_j \cdot 1$$

*

ATTENZIONE

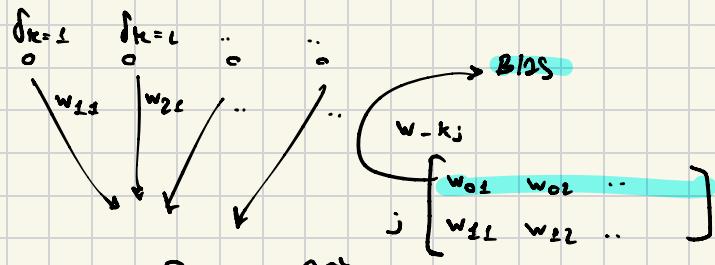
Nel calcolo del δ_j ho:

$$\delta_j = \left(\sum_{k=1}^K \delta_k \cdot w_{kj} \right) \cdot f'_j(\text{net}_j)$$

non c'è il bias

c'è il bias

In quanto fatto ho:



- a. qui δ_j B10S non è incluso, quindi nel calcolo
 b. δ_j ha bisogno di sottrarre la prima w_{0j}
 c. poi prodotto scalare tra δ_k e $w_{kj} [2, :]$

Quando δ_j si calcola:

$$\begin{aligned}
 n\text{-junits} &= x\text{-j.size} \\
 d_j &= \text{np. zeros (n-junits)}
 \end{aligned}$$

Per unità j ho bisogno di calcolare il suo δ_j :

$$" \delta_j = \left[\left(\sum_{k=1}^K \delta_k \cdot w_{kj} \right) \right] \left[f'_j (net_j) \right]^{(1)} \left[f'_j (net_j) \right]^{(2)}$$

Per unità in n-junits:

$$(1) \rightarrow \text{np. dot} (x\text{-i}, w\text{-ji} [1:junit]) + w\text{-jL} [\emptyset] [junit]$$

$$(2) \rightarrow \text{np. dot} (z\text{-k}, w\text{-kj} [junits+1, :])$$

$$d_j [junit] = (1) + f'_{act} (1)$$