

Adesso :

la matrice k_i per : $w_{kj} =$

MATRICE DI
 $\delta_k \cdot o_j$
 PER GESTIONE DEL
 BATCH

$$\begin{bmatrix} [1 \ 1 \ 1 \ \dots \ 1] \\ [] \\ \vdots \\ [] \end{bmatrix}$$
 inizializzata con

$$, \text{bins_row} = \text{np.ones} (1, x.\text{shape}[L])$$

aggiungiamo matrice k_i $\Delta_p w_{kj} = \delta_k \cdot o_j :$

dove $w_{kj}^{\text{new}} = w_{kj}^{\text{old}} + \eta \cdot \Delta_p w_{kj}$

e **batch** $w_{kj}^{\text{new}} = w_{kj}^{\text{old}} + \eta \cdot \sum_p \Delta_p w_{kj}$

per poi passare nella classe trainer :

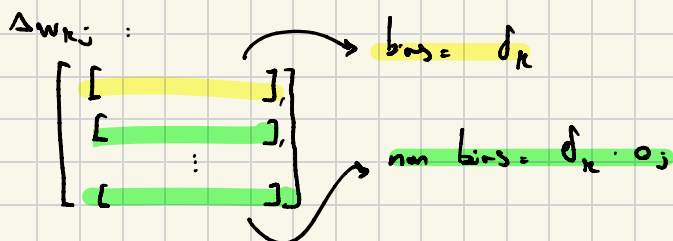
Questa matrice :
 - viene reinizializzata
 ad ogni epoca
 deve avere la
 stessa shape di w_{kj}

online $\begin{bmatrix} w_{kj} \end{bmatrix} += \eta \cdot \begin{bmatrix} \Delta_p w_{kj} \end{bmatrix}$

batch $\begin{bmatrix} w_{kj} \end{bmatrix} += \eta \cdot \begin{bmatrix} \sum_p \Delta_p w_{kj} \end{bmatrix}$

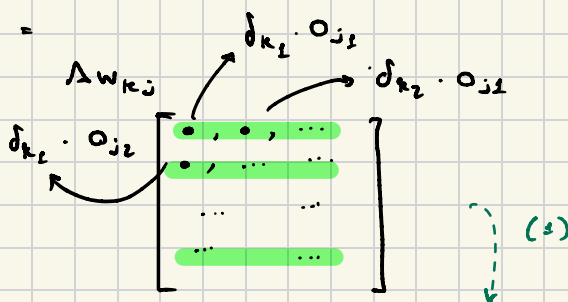
mini b ?

IL BIAS IN
MATRICE Δw_{kj}



in codice la creazione di Δw_{kj} non bias:

$$\Delta w_{kj_non_b} = data_w =$$



quindi ogni elemento di
x-pur moltiplica ogni elem.
di δ_k , uso np.outer

(2)

quindi esempio

$$\text{np.outer}(x_j = [a, b], \delta_k = [1, 2]) =$$

$$= \begin{bmatrix} [a \cdot 1, a \cdot 2] \\ [b \cdot 1, b \cdot 2] \end{bmatrix}$$

Per ogni j il bias posto aggiunto in testa x_j a x_{j-1} :

$$\text{np. outer} (x_j = [1, a, b], \delta_k = [1, 2]) =$$

$$x_j[\emptyset] \cdot \delta_k[\emptyset] = \delta_k[\emptyset]$$

$$\begin{bmatrix} [*, \dots, \dots], \\ [\dots], \\ [\dots] \end{bmatrix}$$

Una volta calcolato Δw_{kj} posto aggiunto w_{kj}