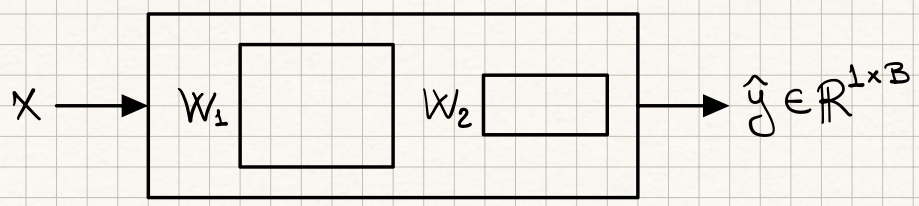
BACK PROPAGATION

L'idea di questo algoritmo è quella di aggiornare tutti i pesi della rete propagando all'indietro l'errore che fa la rete usandola in forward:

1) FORWARD ACTIVATION
2) CALCOLO DELL'ERRORE DELL'OUTPUT
3) PROPAGARE L'ERRORE ALL'INDIETRO

L'idea è aggiustare i pesi per far si di migliorare l'errore in uscita, in proporzione al contributo che dà quel livello all'uscita.

Prendiamo il nostro modello:



Dove W_1 è la matrice dei pesi degli hidden layer e W_2 la matrice dei pesi del livello d'uscita.

Per allenare la rete, usiamo la loss MSE:

$$\mathcal{L}^{MSE} = \frac{1}{2B}(\hat{y}-y)(\hat{y}-y)^T$$

Con B numero di elementi dei vettori $\hat{y}$ e y

Per aggiornare i pesi usiamo:

$$W_1 = W_1 - \gamma \frac{\partial \mathcal{L}^{MSE}}{\partial W_1}$$

$$W_2 = W_2 - \gamma \frac{\partial \mathcal{L}^{MSE}}{\partial W_2}$$

Possiamo scrivere l'uscita in relazione all'ingresso:

$$\hat{y} = W_2 \cdot \underbrace{Sxt\left(\sigma(W_1 \cdot x_e)\right)}_{\longrightarrow A_1}$$

Dove quindi ingresso e uscita sono legate da una relazione non lineare.

Per allenare il modello dobbiamo minimizzare la loss, e lo facciamo sia per aggiustare W_1 che W_2.
Calcoliamo quindi:

$$1)\ \frac{\partial \mathcal{L}^{MSE}}{\partial W_1} \qquad 2)\ \frac{\partial \mathcal{L}^{MSE}}{\partial W_2}$$

Derivata rispetto a W_2

$$\frac{\partial \mathcal{L}^{MSE}}{\partial W_2} = \frac{\partial\left(\frac{1}{2B}(\hat{y}-y)(\hat{y}-y)^T\right)}{\partial W_2} = \frac{\partial}{\partial W_2}\left(\frac{1}{2B}(W_2 A_1 - y)(W_2 A_1 - y)^T\right)$$

Per semplificare possiamo usare la chain rule:

$$\frac{\partial \mathcal{L}^{MSE}}{\partial W_2} = \frac{\partial \mathcal{L}^{MSE}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W_2}$$

Dove:

1) $\frac{\partial \mathcal{L}^{MSE}}{\partial \hat{y}} = \frac{\partial \mathcal{L}^{MSE}}{\partial z a_2} = \frac{1}{B}(\hat{y} - y)$

2) $\frac{\partial \hat{y}}{\partial W_2} = A_1^T$

Mettendo insieme:

$$\frac{\partial \mathcal{L}^{MSE}}{\partial W_2} = \frac{1}{B}(\hat{y} - y)\ A_1^T$$

Derivata rispetto a W_1

Per calcolare questa derivata, facciamo un'altra chain rule, ricordando lo schema della rete:

$z a_0 = X$
$a_0 = \delta x tented(z a_0)$
$z z_1 = W_1 \cdot a_0$
$z a_1 = \sigma(z z_1)$
$a_1 = \delta x tented(z a_1)$
$z z_2 = W_2 \cdot a_1$
$z A_2 = lin(z z_2) = \hat{y}$

$\Rightarrow$

$$\frac{\partial \mathcal{L}}{\partial W_1} = \frac{\partial \mathcal{L}}{\partial z z_1} \frac{\partial z z_1}{\partial W_1} \quad A_0^T$$

$$\frac{\partial \mathcal{L}}{\partial z z_1} = \frac{\partial \mathcal{L}}{\partial A_1} \frac{\partial A_1}{\partial z z_1} \quad \sigma'(z z_1)$$

$$\frac{\partial \mathcal{L}}{\partial A_1} = \frac{\partial \mathcal{L}}{\partial z z_2} \frac{\partial z z_2}{\partial A_1} \quad W_2^T$$

$\frac{\partial \mathcal{L}}{\partial z z_2} = \frac{1}{B}(\hat{y} - y)$

$$\frac{\partial \mathcal{L}}{\partial W_1} = A_0^T\ \sigma'(z z_1)\ \frac{1}{B}(\hat{y} - y)\ W_2^T$$