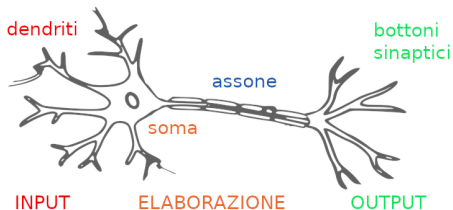


IL METODO DEL GRADIENTE PER L'ADDESTRAMENTO DI RETI NEURALI

Neuroni



Un **neurone** è una cellula che

- riceve
- elabora
- trasmette

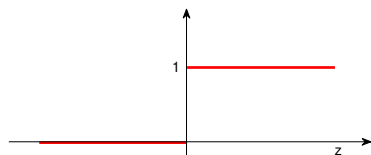
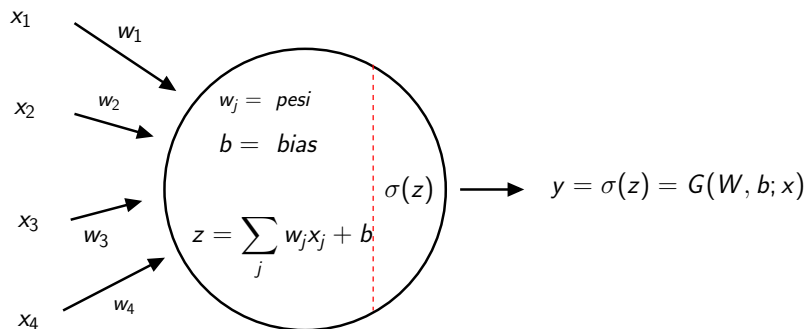
informazioni da/ad altre cellule attraverso segnali elettrici e chimici.

Se la somma degli segnali in ingresso supera una certa soglia, allora il neurone trasmette il segnale (è come un sensore).

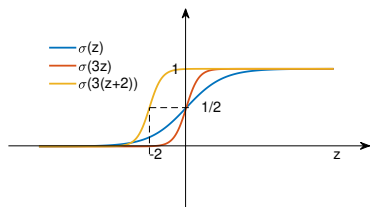


Nel cervello umano è stimato vi siano 100 miliardi di neuroni.

Neurone artificiale



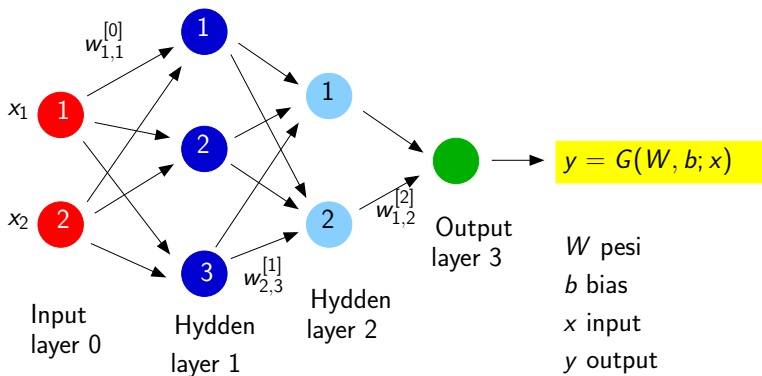
$$\text{gradino } \sigma(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$



$$\text{sigmoide } \sigma(z) = \frac{1}{1 + e^{-z}}$$

Reti neurali

Una rete neurale è modellata da un grafo. I nodi sono i neuroni e gli archi sono le sinapsi.



Ogni arco ha associato un **peso**, ogni nodo ha associato un **bias**.
Numero di layer e numero di neuroni per ogni layer sono fissati a priori. **Incognite:** pesi e bias

Addestramento di una rete neurale

Serve per determinare pesi e bias della rete neurale.

- Si prendono molti set di input $\mathbf{x}^{\{i\}} = [x_1^{\{i\}}, x_2^{\{i\}}]$ ed i corrispondenti output (noti) $y^{\{i\}}$, per $i = 1, \dots, N$ (N molto grande),
- si raccolgono tutte le incognite W e b in un vettore \mathbf{s} ,
- si definisce la **loss function**

$$\Phi(\mathbf{s}) = \frac{1}{2} \sum_{i=1}^N \|y^{\{i\}} - G(\mathbf{s}; \mathbf{x}^{\{i\}})\|^2,$$

$y^{\{i\}}$ sono i valori noti (target), $G(\mathbf{s}; \mathbf{x}^{\{i\}})$ sono i valori calcolati con la rete neurale a partire dai dati $\mathbf{x}^{\{i\}}$ associati a $y^{\{i\}}$;

- si calcola \mathbf{s}^* punto di minimo di $\Phi(\mathbf{s})$, cioè:

$$\Phi(\mathbf{s}^*) = \min_{\mathbf{s}} \Phi(\mathbf{s})$$

- \mathbf{s}^* contiene i pesi ed i bias ottimali della rete neurale

Metodo del gradiente per calcolare \mathbf{s}^*

Il metodo del gradiente può essere utilizzato per calcolare il punto di minimo di una funzione $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ generica (non necessariamente del tipo $\Phi(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T \mathbf{A} \mathbf{s} - \mathbf{s}^T \mathbf{b}$ come per risolvere un sistema lineare).

Si costruisce una successione $\{\mathbf{s}^{(k)}\}_{k \geq 0}$:

$$\begin{cases} \mathbf{s}^{(0)} \text{ dato} \\ \mathbf{s}^{(k+1)} = \mathbf{s}^{(k)} + \alpha_k \mathbf{d}^{(k)}. \end{cases}$$

- Si sceglie la *direzione di discesa del gradiente*

$$\mathbf{d}^{(k)} = -\nabla \Phi(\mathbf{s}^{(k)}).$$

- La scelta di $\mathbf{s}^{(0)}$ ora è cruciale, se Φ non è globalmente convessa, non è garantita la convergenza al punto di minimo assoluto di Φ per un dato iniziale $\mathbf{s}^{(0)}$ qualsiasi (potrebbe convergere ad un punto di minimo locale o di sella).

- Esistono strategie per calcolare α_k , ma non ricette ottimali,
- spesso si prende $\alpha_k = \text{costante}$ (è detto **learning rate**),
- quando N è molto grande, calcolare $\Phi(\mathbf{s}^{(k)})$ è molto costoso, si fa la seguente approssimazione:

$$\nabla\Phi(\mathbf{s}^{(k)}) \approx \frac{1}{2}\nabla(\|y^{\{i\}} - G(\mathbf{s}^{(k)}; \mathbf{x}^{\{i\}})\|^2)$$

con i scelto casualmente nell'insieme $\{1, \dots, N\}$ (in tal caso si parla di **metodo del gradiente stocastico**).

Osservazione: (k) è il passo del metodo del gradiente, $\{i\}$ è l'indice del set di input.