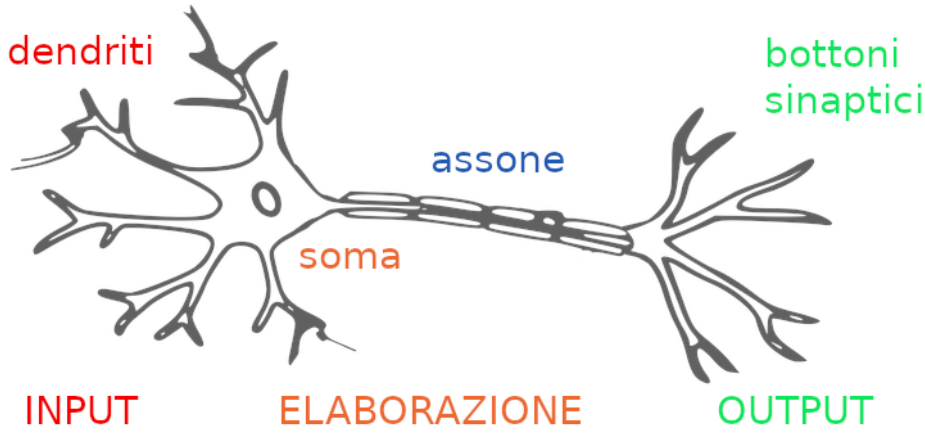
# IL METODO DEL GRADIENTE PER L'ADDESTRAMENTO DI RETI NEURALI

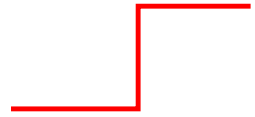
# Neuroni




Un **neurone** è una cellula che

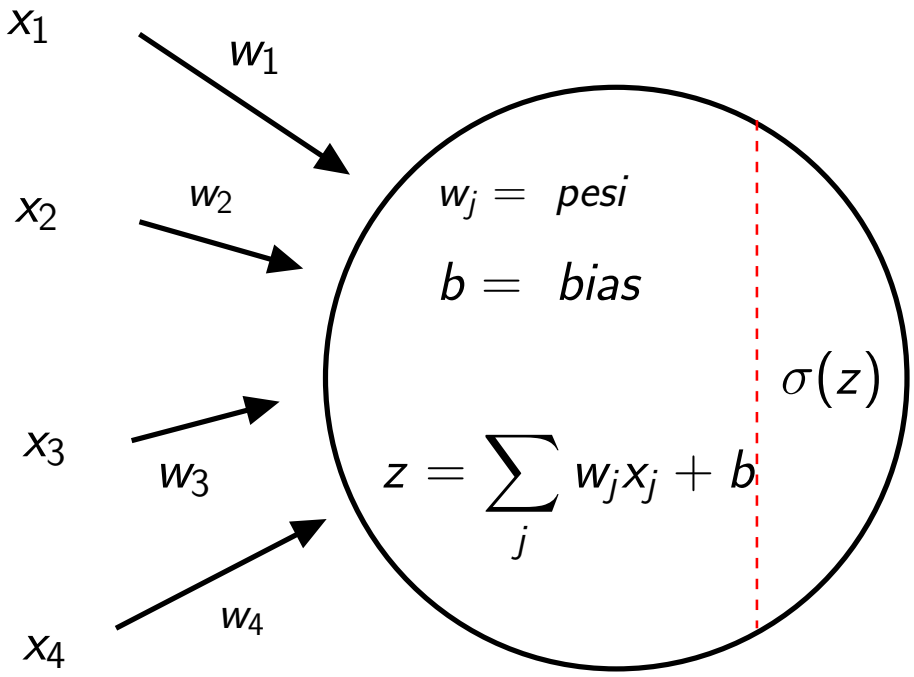
- riceve
- elabora
- trasmette

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).

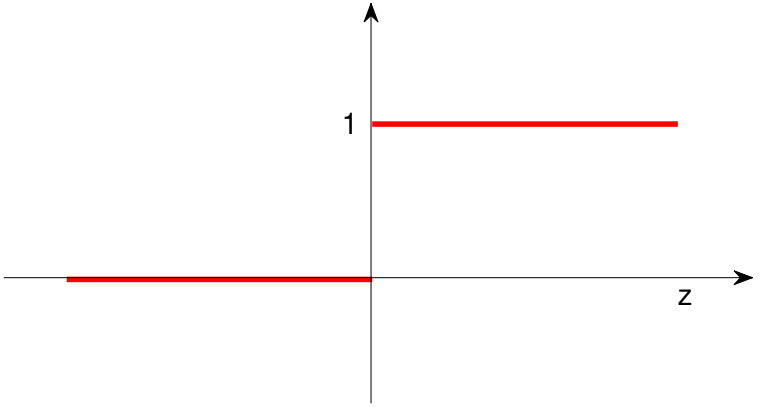


Nel cervello umano è stimato vi siano 100 milardi di neuroni.

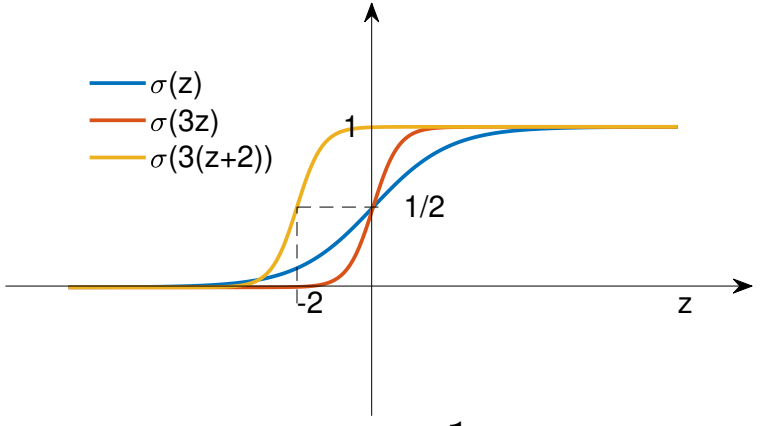
# Neurone artificiale




$$y = \sigma(z) = G(W, b; x)$$



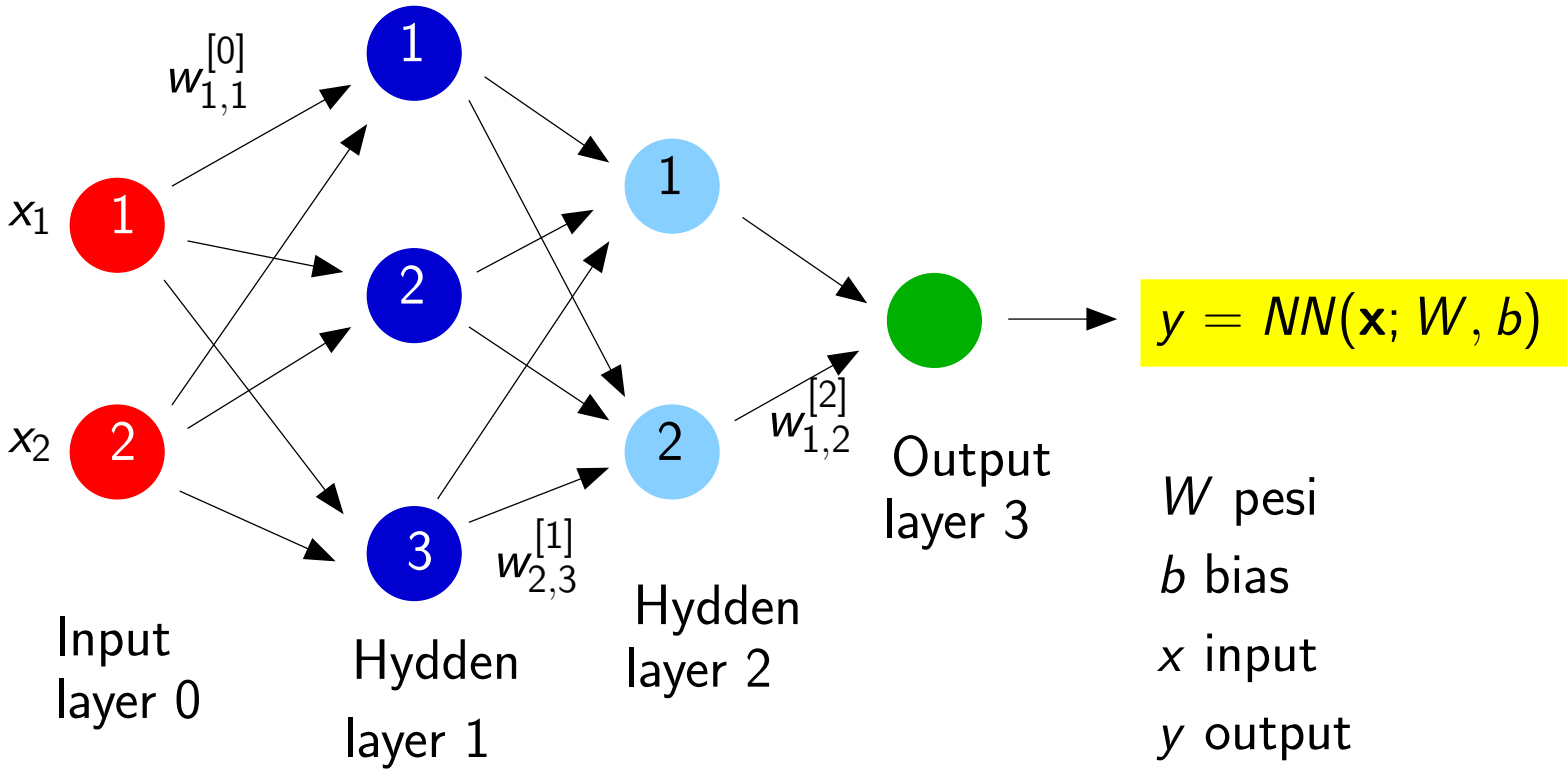

gradino $\sigma(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$




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

# Reti neurali – Neural Networks (NN)

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

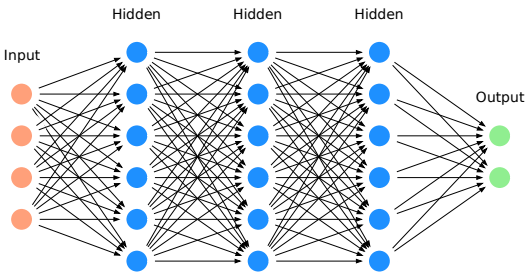



Ogni arco ha associato un peso, ogni nodo ha associato un bias. Numero di layer e numero di neuroni per ogni layer sono fissati a priori. Incognite: pesi e bias

# 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 $i = 1, \ldots, N$ ($N$ molto grande),
- si raccolgono tutte le incognite $W$ e $b$ in un vettore $\mathbf{s}$,
- si definisce la **loss function**




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

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

- si cerca $\mathbf{s}^*$ punto di minimo di $\mathcal{L}(\mathbf{s})$, cioè:

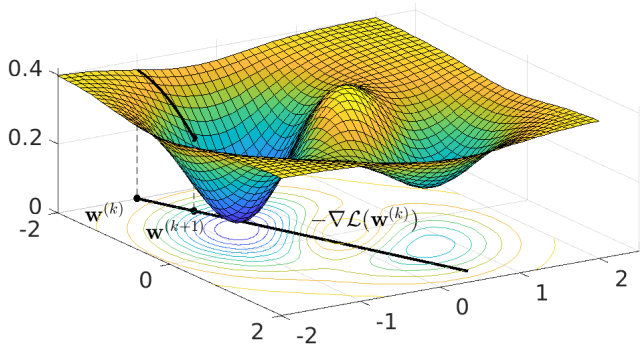
$$\mathcal{L}(\mathbf{s}^*) = \min_{\mathbf{s}} \mathcal{L}(\mathbf{s})$$

- $\mathbf{s}^*$ 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 $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ generica (non necessariamente del tipo $\mathcal{L}(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T A\mathbf{s} - \mathbf{s}^T\mathbf{b}$ 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*

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

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

- Esistono strategie per calcolare $\alpha_k$, ma non ricette ottimali,
- spesso si prende $\alpha_k = \textit{costante}$ (è detto learning rate),
- quando $N$ è molto grande, calcolare

$$\nabla \mathcal{L}(\mathbf{s}^{(k)}) = \nabla \left( \frac{1}{2} \sum_{i=1}^{N} \| y^{\{i\}} - NN(\mathbf{x}^{\{i\}}; \mathbf{s}^{(k)}) \|^2 \right)$$

è molto costoso, allora si fa la seguente approssimazione:

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

con $i$ scelto casualmente nell'insieme $\{1, \ldots, N\}$ (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.