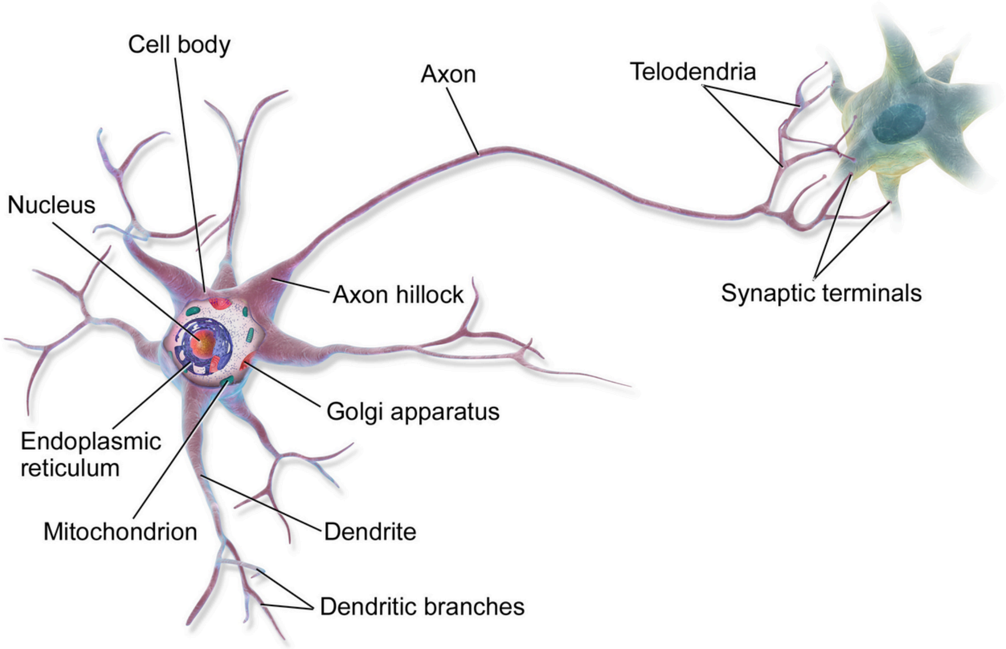
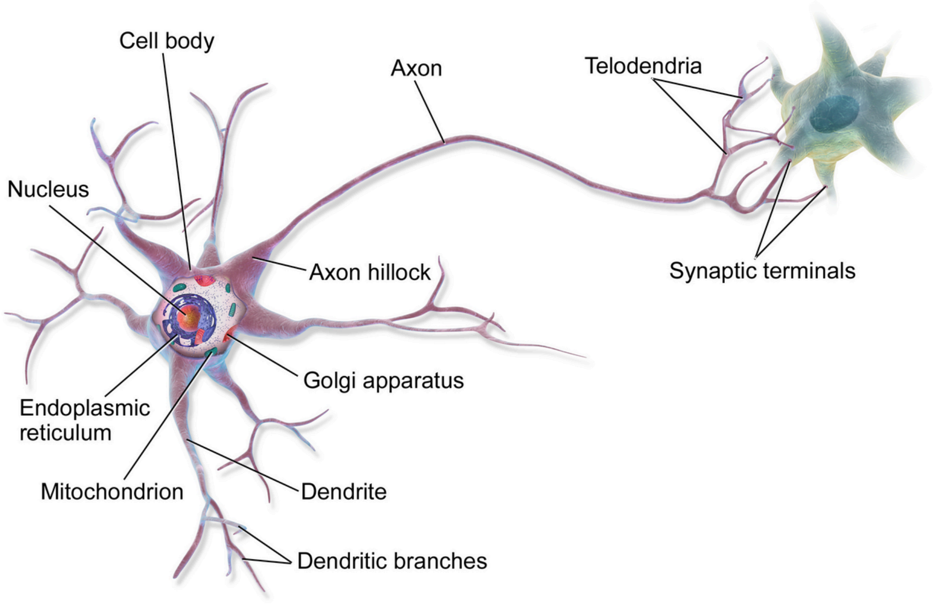
Il Neurone Biologico



Nota: Il soma (cell body) opera un'integrazione dinamica dei segnali: ogni input (eccitatorio o inibitorio) viene pesato in base alla sua intensità, frequenza e posizione anatomica per determinare l'attivazione del neurone.

Dendriti (input) → **Soma** (elaborazione) → **Assone** (output)

Plasticità Sinaptica



Integrazione del segnale

Nel soma, ogni input eccitatorio o inibitorio viene pesato in base alla sua intensità, frequenza e posizione anatomica per determinare l'attivazione del neurone.

Plasticità delle connessioni

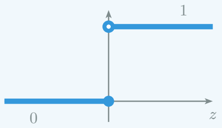
La plasticità sinaptica modula nel tempo l'efficacia delle connessioni attraverso meccanismi di potenziamento e depressione.

Uso ripetuto → Potenziamento **Disuso → Depressione**

Funzioni di Attivazione

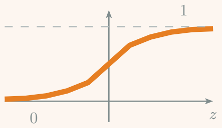
Introducono la **non linearità** che rende le reti neurali potenti

Step (gradino)



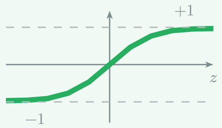
$$\sigma(z) = \begin{cases} 1 \text{ se } z > 0 \\ 0 \text{ altrimenti} \end{cases}$$

Sigmoid



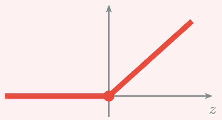
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Tanh



$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

ReLU

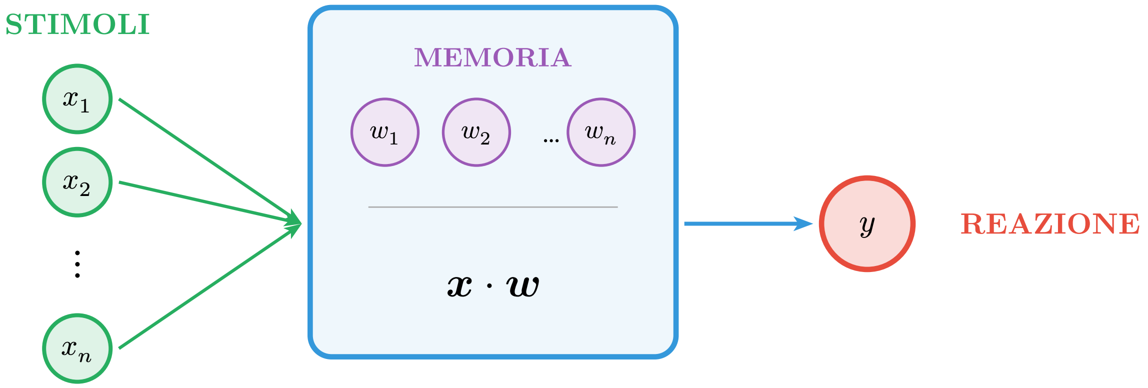


$$\mathrm{ReLU}(z) = \max(0, z)$$

Perché servono? Senza funzioni di attivazione, la composizione di più strati lineari/affini collassa in un'unica trasformazione affine $\boldsymbol{Ax} + \boldsymbol{c}$. La non linearità rende la rete capace di rappresentare funzioni molto più complesse e, in condizioni opportune, di approssimare **qualsiasi funzione** (**approssimatore universale**).

Il Neurone Artificiale

Attenzione: versione incompleta



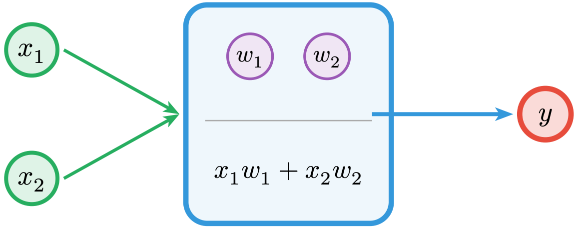
$$y = x_1 w_1 + x_2 w_2 + \ldots + x_n w_n$$

$$\boxed{y = \boldsymbol{x} \cdot \boldsymbol{w}}$$

Devo ripassare fisica: domani verifica o domani interrogazione

Ripassare Fisica: $x_1 \lor x_2$

x_1 = domani verifica, x_2 = domani interrogazione



x_1	x_2	Calcolo	Valore	y
0	0	$0 \cdot 1 + 0 \cdot 1$	$\mathbf{0} <= 0$	0 ✗
0	1	$0 \cdot 1 + 1 \cdot 1$	$\mathbf{+1} > 0$	1 ✓
1	0	$1 \cdot 1 + 0 \cdot 1$	$\mathbf{+1} > 0$	1 ✓
1	1	$1 \cdot 1 + 1 \cdot 1$	$\mathbf{+2} > 0$	1 ✓

Regola di decisione:

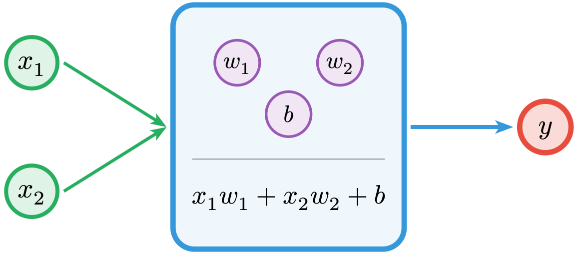
$$y = \begin{cases} 1 & \text{se} \quad x_1 w_1 + x_2 w_2 > 0 \\ 0 & \text{altrimenti} \end{cases}$$

Pesi: $w_1 = +1, \quad w_2 = +1$

Posso uscire a giocare: ho finito i compiti e ho riordinato la camera

Uscire a Giocare: $x_1 \wedge x_2$

x_1 = compiti finiti, x_2 = camera riordinata



x_1	x_2	Calcolo	Valore	y
0	0	$0 \cdot 1 + 0 \cdot 1 - 1.5$	$\mathbf{-1.5} < 0$	0 ✗
0	1	$0 \cdot 1 + 1 \cdot 1 - 1.5$	$\mathbf{-0.5} < 0$	0 ✗
1	0	$1 \cdot 1 + 0 \cdot 1 - 1.5$	$\mathbf{-0.5} < 0$	0 ✗
1	1	$1 \cdot 1 + 1 \cdot 1 - 1.5$	$\mathbf{+0.5} > 0$	1 ✓

Regola di decisione:

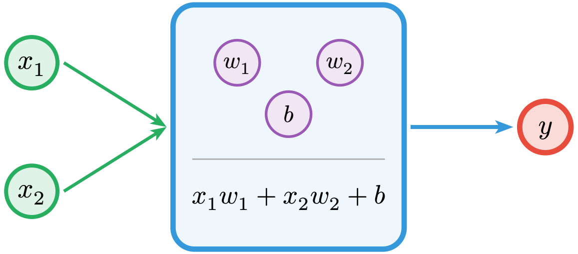
$$y = \begin{cases} 1 & \text{se} \quad x_1 w_1 + x_2 w_2 + b > 0 \\ 0 & \text{altrimenti} \end{cases}$$

Pesi: $w_1 = +1, \quad w_2 = +1, \quad b = -1.5$

Regola d'uso della congiunzione avversativa: devi usare «ma» o «però» ma non entrambi

Ma / Però: $x_1 \oplus x_2$ (XOR)

x_1 = usa «ma», x_2 = usa «però»



x_1	x_2	y atteso
0	0	0
0	1	1
1	0	1
1	1	0

Regola di decisione:

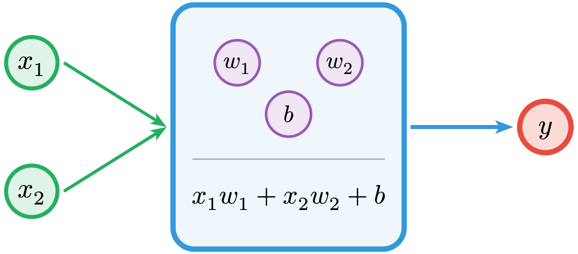
$$y = \begin{cases} 1 & \text{se} \quad x_1 w_1 + x_2 w_2 + b > 0 \\ 0 & \text{altrimenti} \end{cases}$$

Pesi: $w_1 = ?$, $w_2 = ?$, $b = ?$

Regola d'uso della congiunzione avversativa: devi usare «ma» o «però» ma non entrambi

Ma / Però: $x_1 \oplus x_2$ (XOR)

$x_1 =$ usa «ma», $x_2 =$ usa «però»



Tentativo: $w_1 = 1, w_2 = 1, b = -0.5$

x_1	x_2	Calcolo	Valore	y	Atteso
0	0	$0 + 0 - 0.5$	$\mathbf{-0.5} < 0$	0	0 ✓
0	1	$0 + 1 - 0.5$	$\mathbf{+0.5} > 0$	1	1 ✓
1	0	$1 + 0 - 0.5$	$\mathbf{+0.5} > 0$	1	1 ✓
1	1	$1 + 1 - 0.5$	$\mathbf{+1.5} > 0$	1	0 ✗

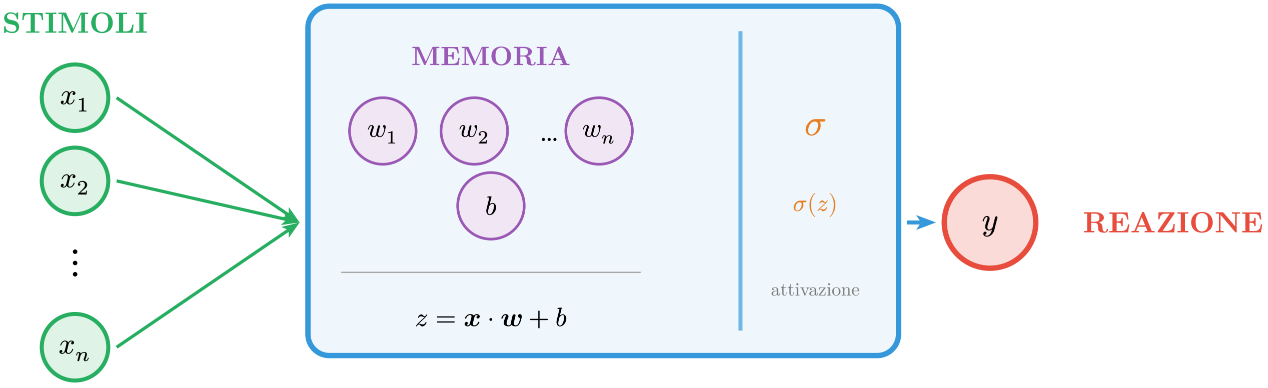
Un singolo neurone non basta!

Non esiste **nessuna** scelta di w_1, w_2, b che risolva XOR. Il problema **non è linearmente separabile**.

Servono più neuroni

Per risolvere XOR servono **almeno 2 strati** di neuroni: nasce la **rete neurale**.

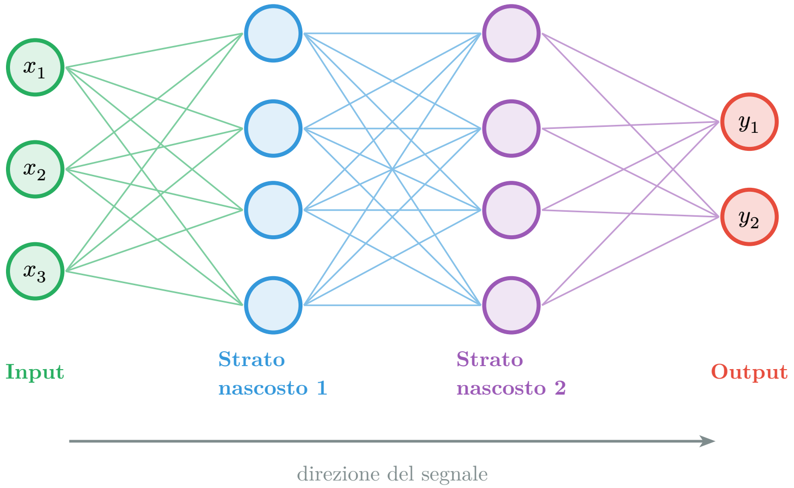
Il Neurone Artificiale



$$z = x_1 w_1 + x_2 w_2 + \ldots + x_n w_n + b$$

$$\boxed{y = \sigma(\boldsymbol{x} \cdot \boldsymbol{w} + b)}$$

La Rete Neurale



Come leggere lo schema:

Cerchi = neuroni, organizzati in **strati** verticali

Linee = connessioni, ognuna con un **peso** w

Ogni neurone calcola $y = \sigma(\boldsymbol{x} \cdot \boldsymbol{w} + b)$

Il segnale scorre da sinistra a destra, strato per strato

Strati nascosti = elaborazione intermedia
Più strati → problemi più complessi (es. XOR)

Ogni Strato è un'operazione Matriciale

$$\overset{3\times 1}{\overset{\boldsymbol{x}}{\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}}} \times \overset{4\times 3}{\overset{\boldsymbol{W}}{\begin{bmatrix}w_{11} & w_{12} & w_{13}\\ w_{21} & w_{22} & w_{23}\\ w_{31} & w_{32} & w_{33}\\ w_{41} & w_{42} & w_{43}\end{bmatrix}}} + \overset{4\times 1}{\overset{\boldsymbol{b}}{\begin{bmatrix}b_1\\b_2\\b_3\\b_4\end{bmatrix}}} \longrightarrow \sigma \longrightarrow \overset{4\times 1}{\overset{\boldsymbol{y}}{\begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix}}}$$

ogni **riga** di $\boldsymbol{W}$ contiene i pesi di **un neurone**

Uno strato = una moltiplicazione

$$\boldsymbol{y} = \sigma(\boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b})$$

Dimensioni (n input, m neuroni):

$\boldsymbol{x}$: $n \times 1$ $\boldsymbol{W}$: $m \times n$ $\boldsymbol{b}$: $m \times 1$ $\boldsymbol{y}$: $m \times 1$

Rete completa = strati in sequenza:

$$\boldsymbol{y} = \sigma_3(\boldsymbol{W}_3 \cdot \sigma_2(\boldsymbol{W}_2 \cdot \sigma_1(\boldsymbol{W}_1 \cdot \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2) + \boldsymbol{b}_3)$$

Calcoli di uno strato **in parallelo** → GPU

Ogni Strato è un'operazione Matriciale

$$\overset{3\times 1}{\boldsymbol{x}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \times \overset{4\times 3}{\boldsymbol{W}} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix} + \overset{4\times 1}{\boldsymbol{b}} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \rightarrow \sigma \rightarrow \overset{4\times 1}{\boldsymbol{y}} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

ogni **riga** di $\boldsymbol{W}$ contiene i pesi di **un neurone**

Ogni strato è una funzione: $f_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$

$$f_i(\boldsymbol{x}) = \sigma_i(\boldsymbol{W}_i \cdot \boldsymbol{x} + \boldsymbol{b}_i)$$

La rete è una composizione di funzioni:

$$F = f_n \circ f_{n-1} \circ \cdots \circ f_1$$

$$\boldsymbol{y} = (f_n \circ f_{n-1} \circ \cdots \circ f_1)(\boldsymbol{x})$$