Introduzione alle Reti Neurali

Senigallia 13-15 Ottobre 2006

Revised: 26 settembre 2006 12h : 50min

Indice

La natura è fonte di ispirazione per i Modelli matematici.

Ad esempio:

- La teoria geometrica dei frattali
- Gli algoritmi genetici
- Le reti neurali artificiali (ANN)

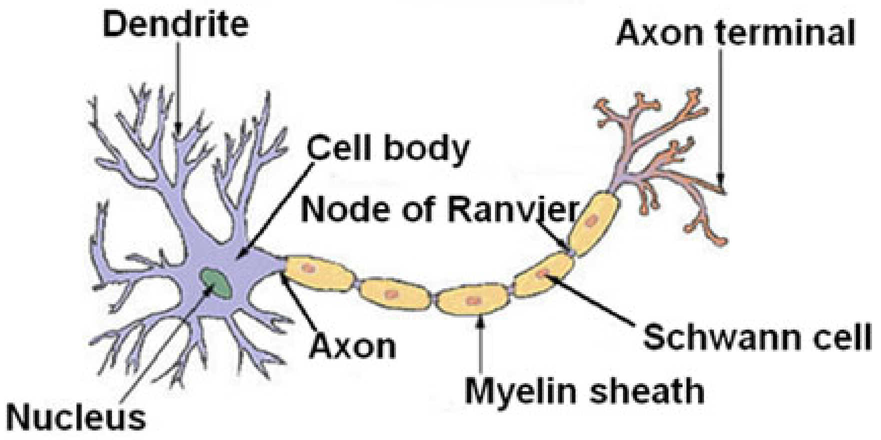
Cronobibliografia sulle Reti Neurali.

- 1942 - McCulloch e Pitts
- 1949 - 1969 - Hebb, Rosenblatt, Widrow, Hoff, Minsky e Papert
- 1969 - 1982 assenza di contributi rilevanti
- 1982 - in poi Nuovi importanti contributi: Hopfield, Kohonen, Rumelhart, Minsky, Chua e Yang.

Attualmente l'interesse per le reti neurali è molto alto.

Le reti neurali artificiali tentano di riprodurre il funzionamento delle reti neurali biologiche.

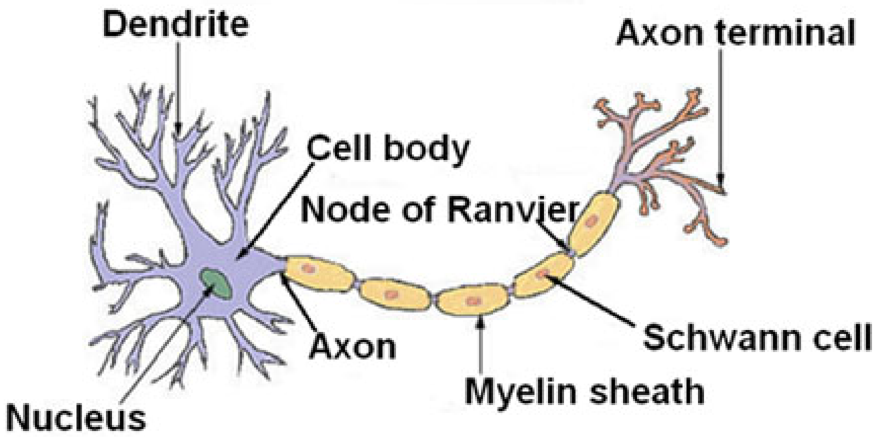
Structure of a Typical Neuron



Soma: è la parte centrale della cellula dove è collocato il nucleo

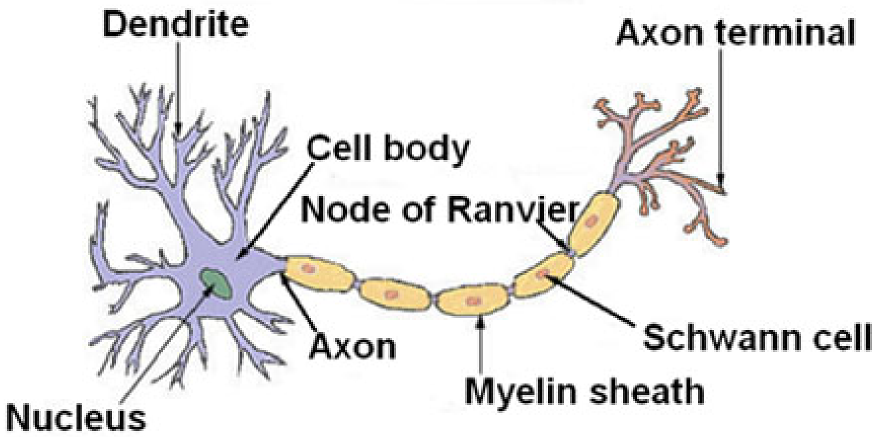
$$5\mu m \rightarrow 1mm$$

Structure of a Typical Neuron



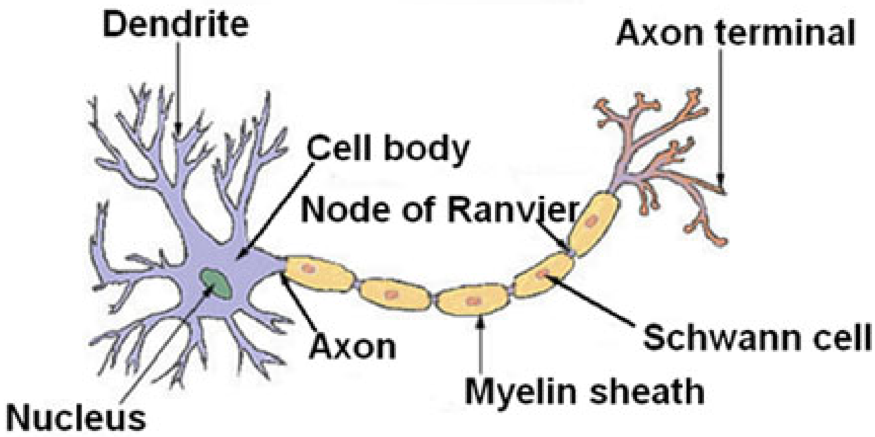
Dendriti: costituiscono la struttura attraverso la quale il neurone acquisisce informazioni.

Structure of a Typical Neuron

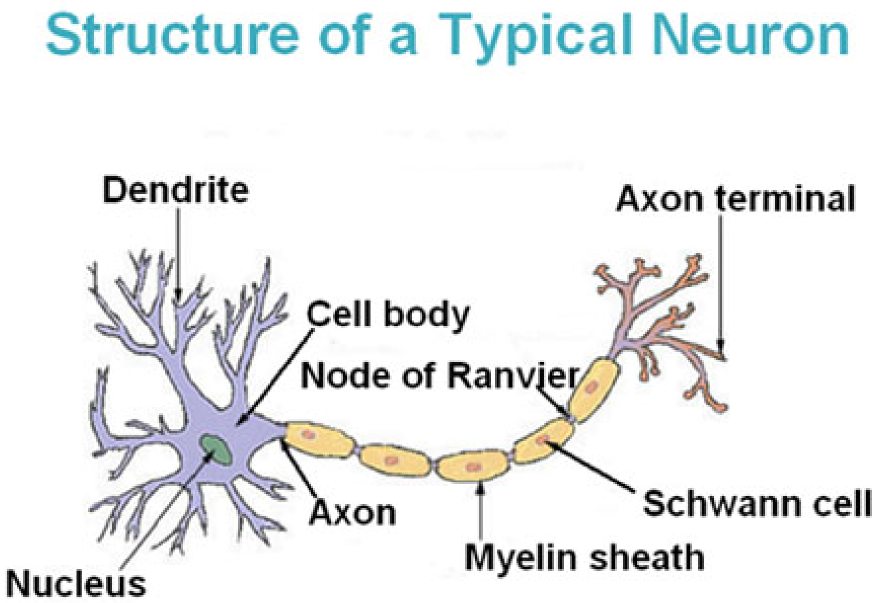


Assone: estensione del neurone; può essere lungo millimetri o metri. l'assone più lungo è il nervo sciatico che parte dalla base della colonna vertebrale e termina all' alluce di ciascun piede.

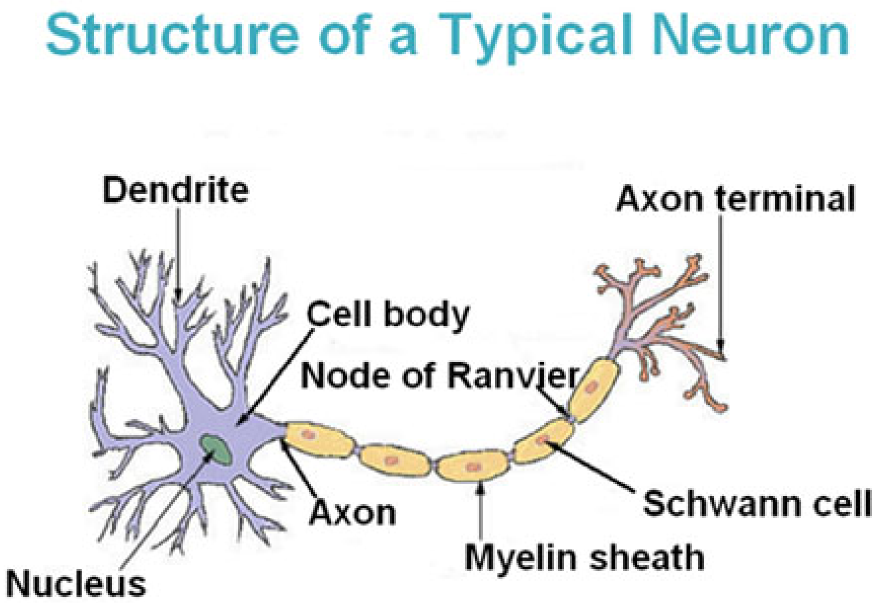
Structure of a Typical Neuron



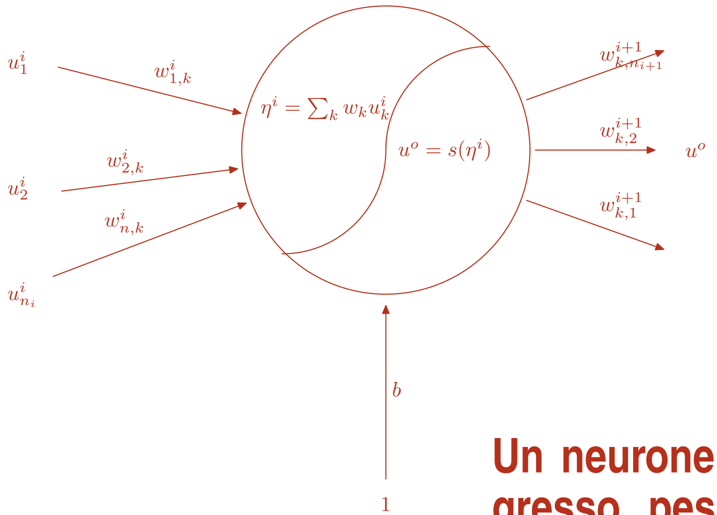
L'assone è rivestito di mielina. Le cellule di Schwann producono la mielina.



I nodi di Ranvier interrompono il rivestimento dell’assone a intervalli regolari. Hanno importanti funzioni nella trasmissione degli impulsi lungo l’assone.

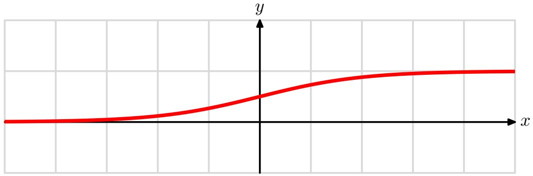


L'assone termina con una struttura che mediante il rilascio di neurostrasmettitori comunica con gli altri neuroni.

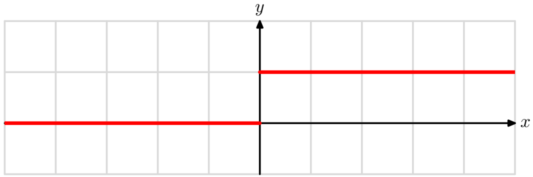


Un neurone riceve segnali u^i in ingresso, pesati mediante i valori w_k^i delle connessioni, li somma e li elabora mediante la funzione s di attivazione.
Restituisce in uscita un valore u^{i+1}

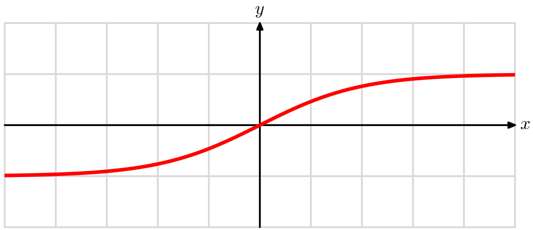
Possibili funzioni di attivazione sono date da



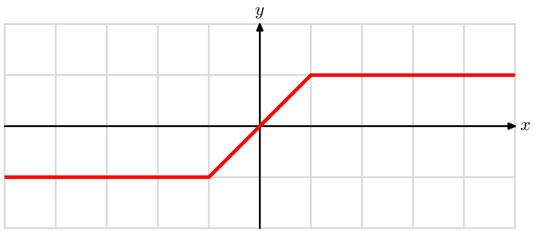
(1) $$\sigma(x) = \frac{1}{1+e^{-x}}$$



(2) $$\sigma(x) = \begin{cases} 0 & x \le a \\ 1 & x > a \end{cases}$$

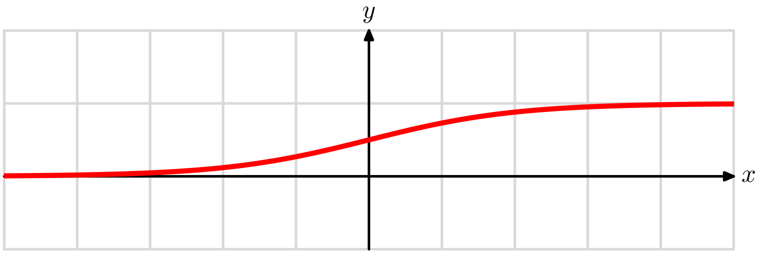


(3) $$\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}}$$



(4) $$\sigma(x) = \begin{cases} -1 & x \le -a \\ x & -a < x \le a \\ 1 & x > a \end{cases}$$

Solitamente è vantaggioso considerare la funzione

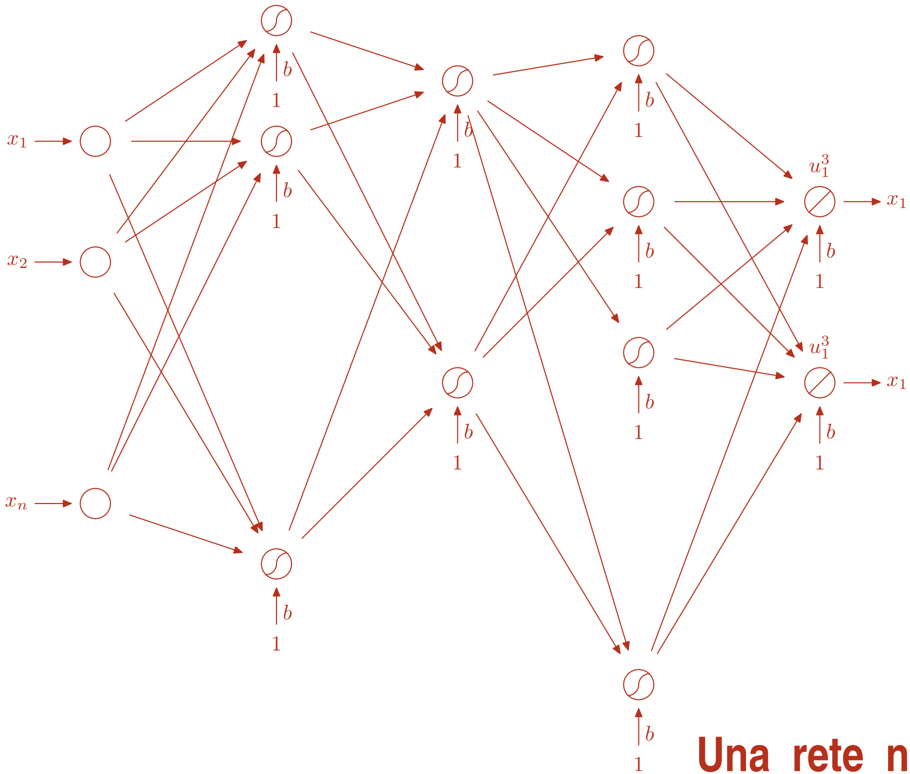


$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

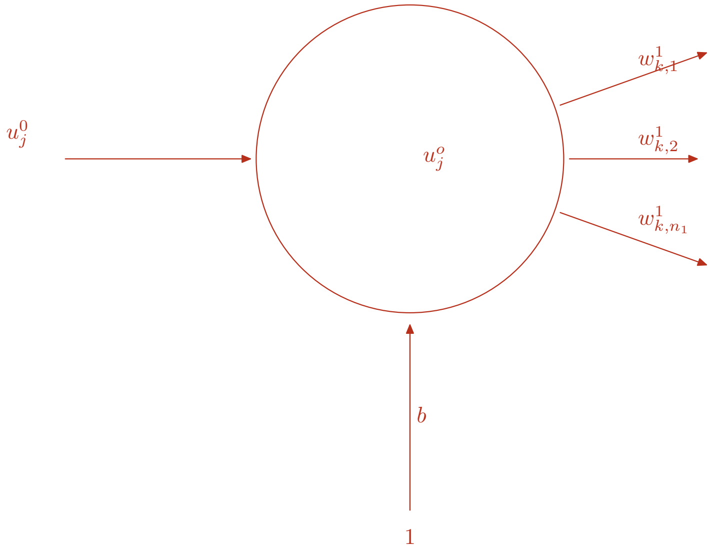
si ha infatti

$$\sigma' = \sigma(1 - \sigma)$$

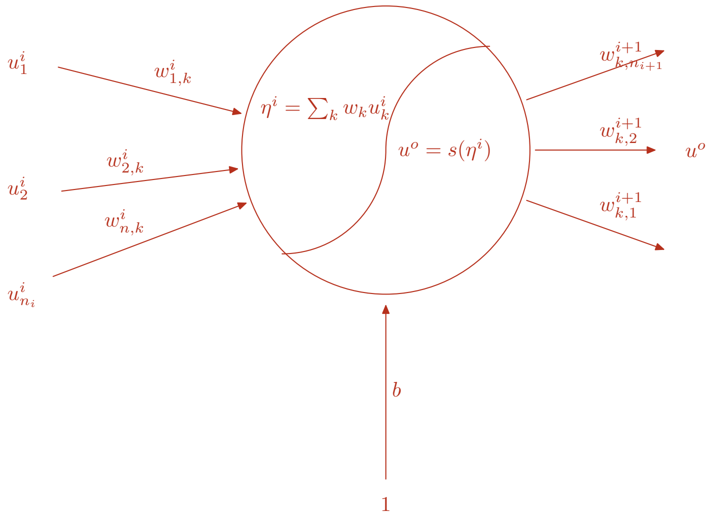
questo semplifica l'algoritmo di backpropagation che consente di adattare la rete



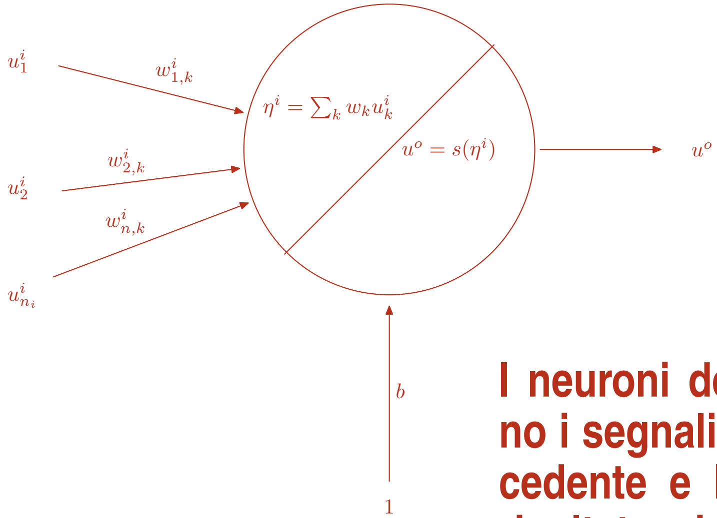
Una rete neurale feedfoward è costituita da un certo numero di neuroni disposti su strati interconnessi tra loro mediante collegamenti che trasportano segnali con diversa rilevanza.



I neuroni del primo strato ricevono segnali dall'esterno e li passano allo strato successivo

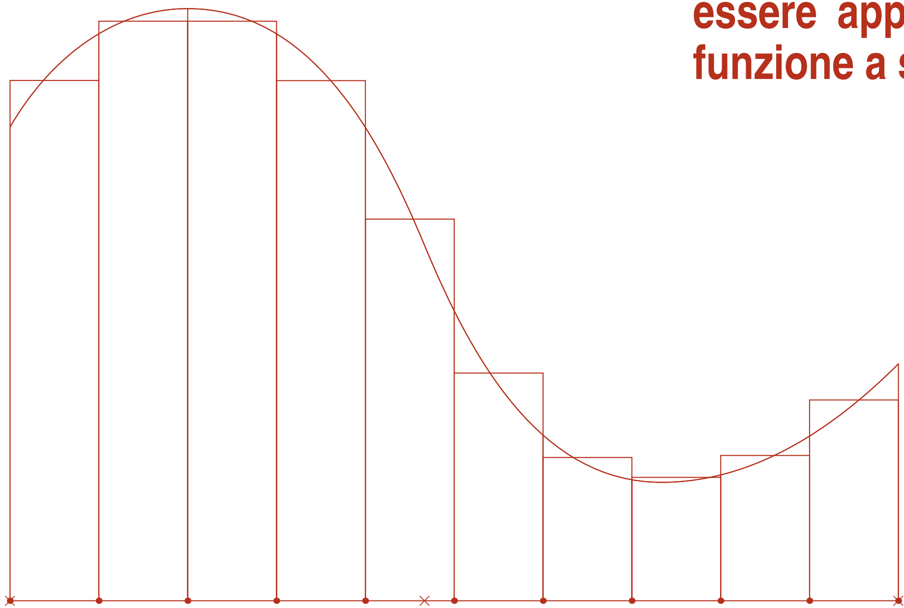


I neuroni degli strati interni elaborano i segnali

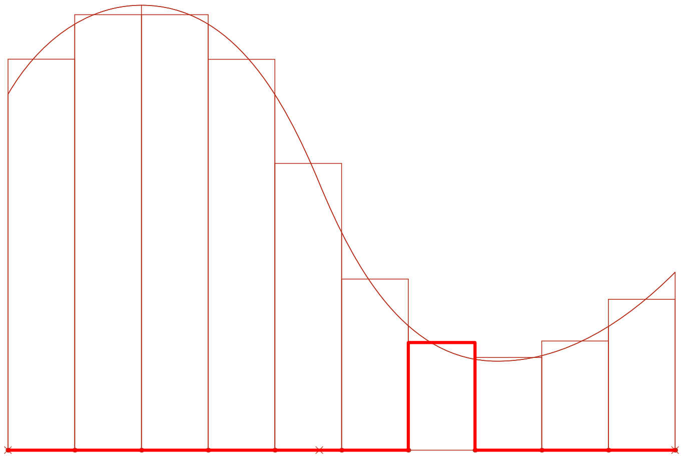


I neuroni dell'ultimo strato sommano i segnali ricevuti dallo strato precedente e li passano all'esterno il risultato; in altre parole la funzione di attivazione dell'ultimo strato è semplicemente l'identità.
(Ciò consente di avere in uscita un qualunque numero reale (e non soltanto un valore compreso in $[0,1]$)

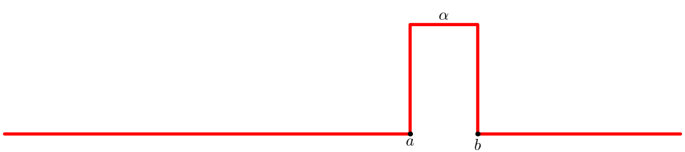
Una rete neurale feedforward è in grado di approssimare ogni funzione continua in modo molto semplice. Infatti ogni funzione continua può essere approssimata mediante una funzione a scalini.

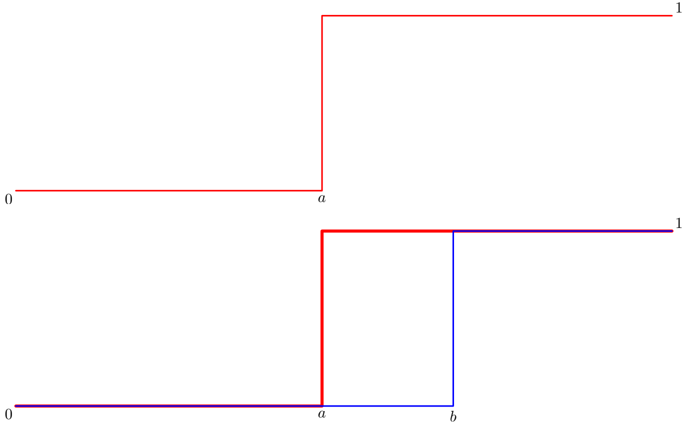


Una funzione a scalini può essere ottenuta come somma pesata di funzioni del tipo



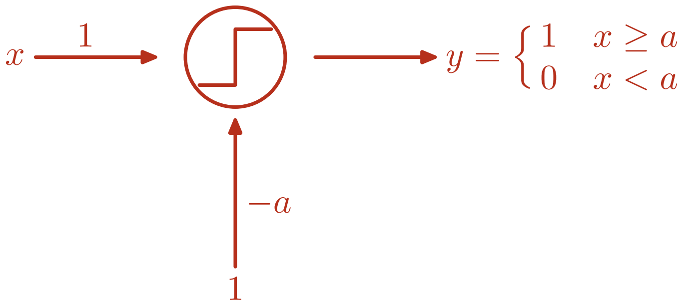
$$\begin{cases} 0 & x < a \\ \alpha & a \leq x \leq b \\ 0 & x > b \end{cases}$$

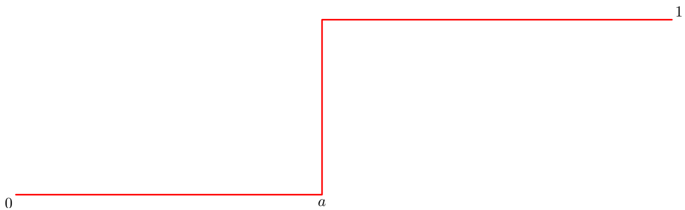




e queste si possono ottenere come differenza di due funzioni del tipo

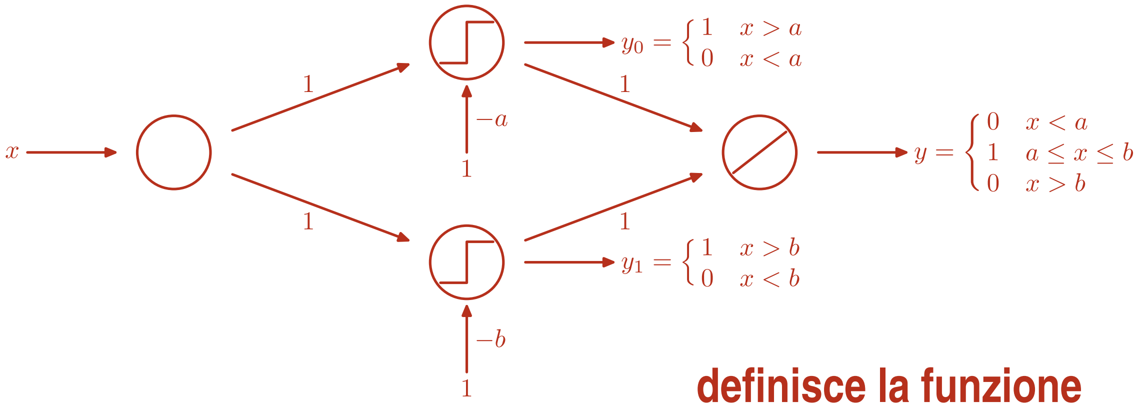
$$\begin{cases} 0 & x < a \\ 1 & x \geq a \end{cases} \quad , \quad \begin{cases} 0 & x < b \\ 1 & x \geq b \end{cases}$$



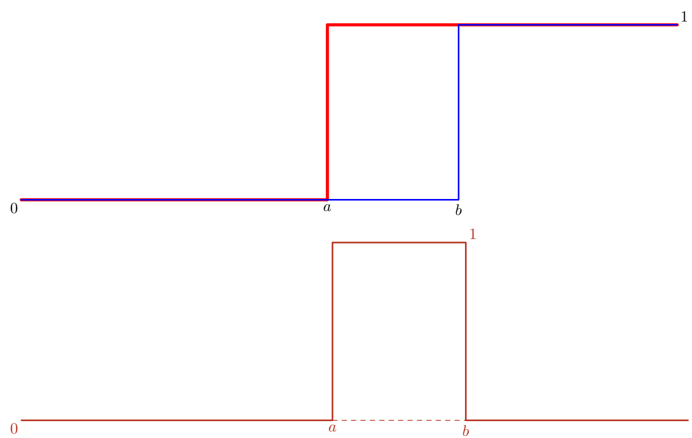


Ciascuna delle quali pu'o essere ottenuta come uscita di un singolo neurone.

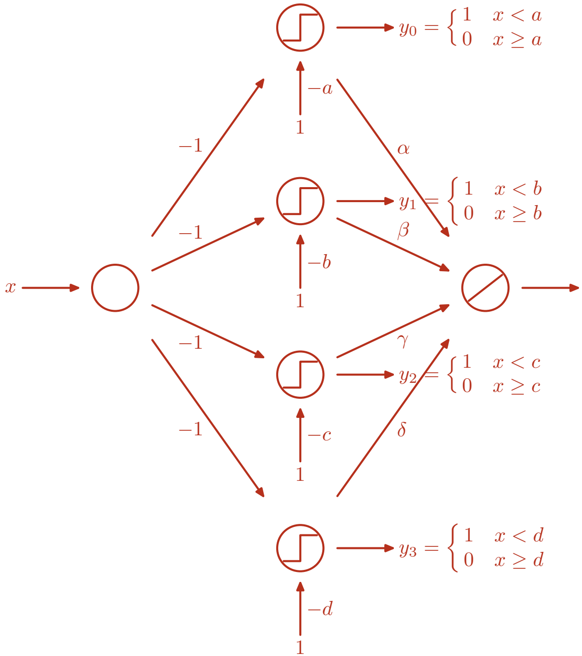
La rete neurale



definisce la funzione

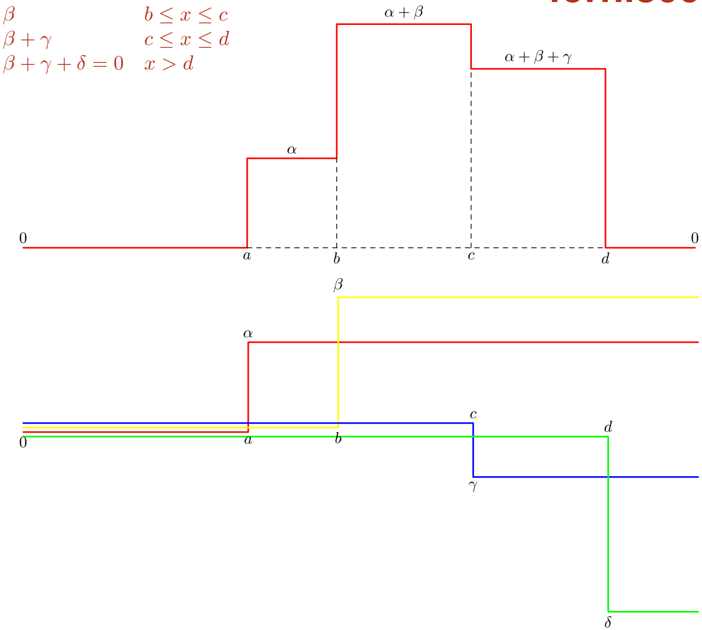


Mentre



$$\bar{q} = \begin{cases} 0 & x < a \\ \alpha & a \leq x \leq b \\ \alpha + \beta & b \leq x \leq c \\ \alpha + \beta + \gamma & c \leq x \leq d \\ \alpha + \beta + \gamma + \delta = 0 & x > d \end{cases}$$

fornisce



Consideriamo

$$(x_j, t_j) \qquad , \qquad j = 1..N$$

Problema: determinare i coefficienti W^i e b^i di una rete neurale in modo che detti o_j i risultati in uscita dalla rete che corrispondono agli ingressi x_j, sia minimo lo scarto quadratico tra o_j e t_j

$$E = \sum_{j=1}^{N} (o_j - t_j)^2$$

$\mathcal{E}$ è funzione dei coefficienti della rete

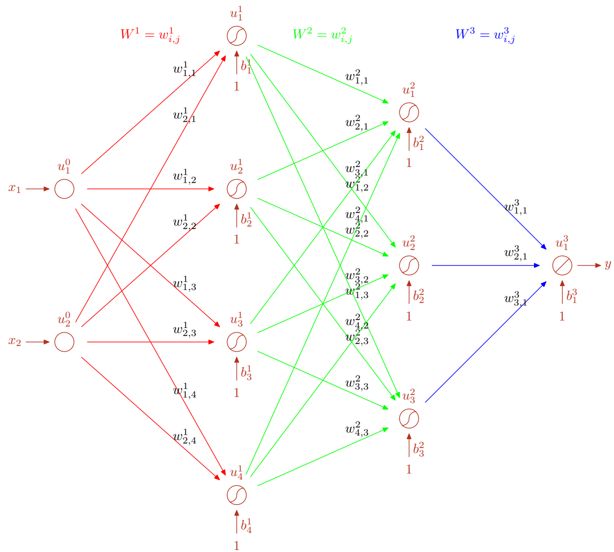
Si può minimizzare usando il metodo del gradiente

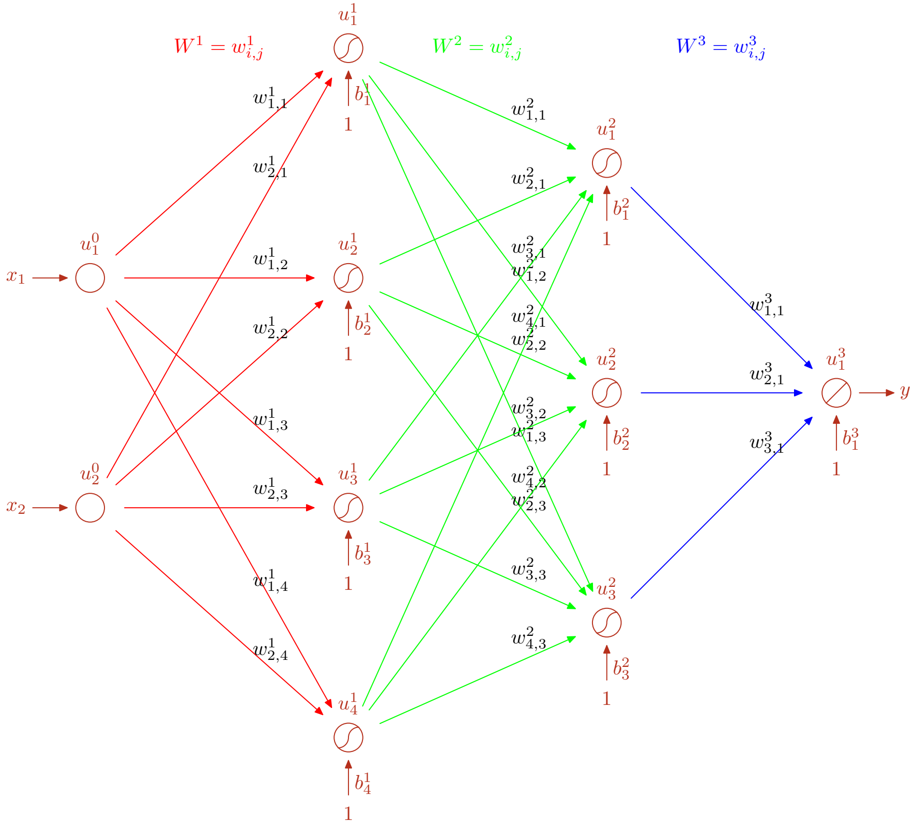
nonostante la apparente complessità dei calcoli necessari per ottenere le derivate di $\mathcal{E}$ rispetto ad ognuno dei parametri

esistono semplici formule localizzate per descriverle.

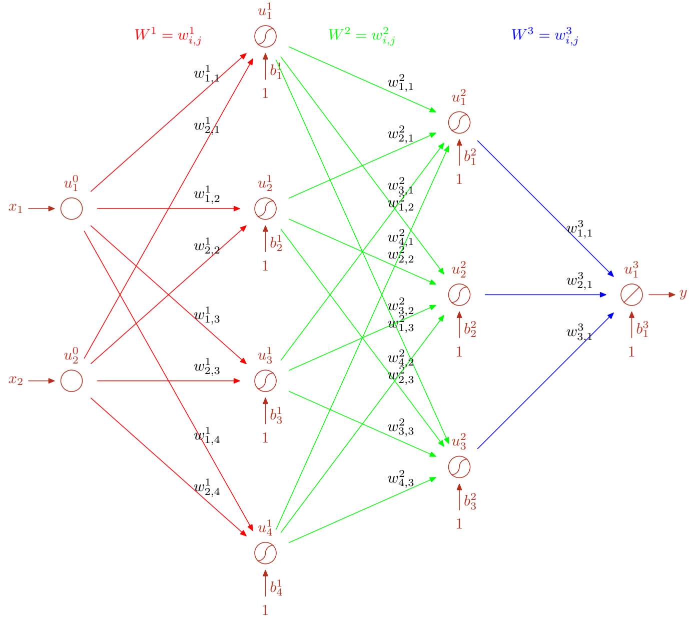
Supponiamo ad esempio di considerare una rete neurale, a due strati nascosti, con

- uno strato di input costituito da 2 neuroni
- uno primo strato nascosto costituito da 4 neuroni
- uno secondo strato nascosto costituito da 3 neuroni
- uno strato di output costituito da 1 neurone





Definisce una funzione reale di due variabili reali

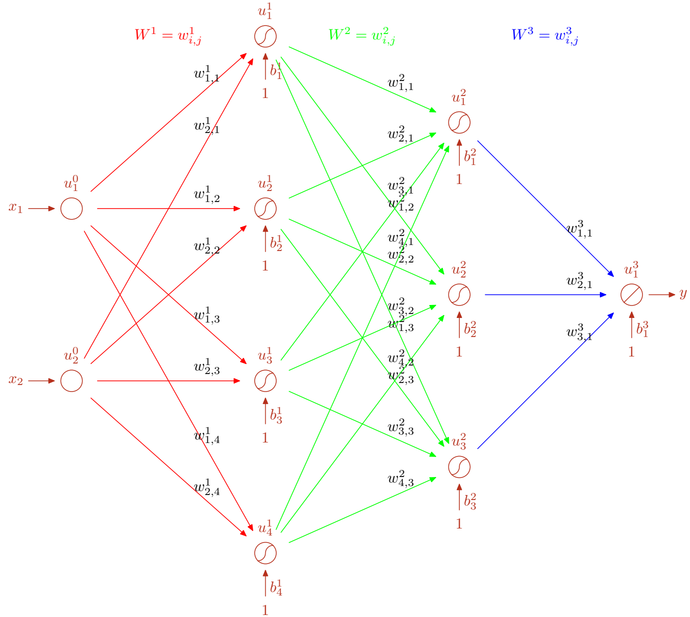


È identificata dalle matrici

$$W^1 = \{w^1_{k,j}\} \quad , \quad W^2 = \{w^2_{k,j}\} \quad , \quad W^3 = \{w^3_{k,j}\}$$

e dai vettori (bias)

$$b^1 = \{b^1_k\} \quad , \quad b^2 = \{b^2_k\} \quad , \quad b^3 = \{b^3_k\}$$



Lo stato della rete può essere descritto mediante i vettori

u_0 **il vettore di input** $u^0 = (u^0_1, u^0_2)$;

u^1 **il vettore di uscita del primo strato** $u^2 = (u^0_1, u^0_2, u^0_2, u^0_2)$

u^2 **il vettore di uscita del secondo strato** $u^2 = (u^0_1, u^0_2, u^0_2)$

u^3 **il vettore di output** $u^3 = o^3$

Le rete neurale produce il corrispondente del segnale di input propagandolo in avanti

$$u^0 = (x_1, x_2) \quad , \quad \bar{u}^0 = (u^0, 1)$$

$$W_1 = \begin{pmatrix} w^1_{1,1} & w^1_{1,2} & w^1_{1,3} & w^1_{1,4} \\ w^1_{2,1} & w^1_{2,2} & w^1_{2,3} & w^1_{2,4} \end{pmatrix} \qquad \bar{W}_1 = \begin{pmatrix} w^1_{1,1} & w^1_{1,2} & w^1_{1,3} & w^1_{1,4} \\ w^1_{2,1} & w^1_{2,2} & w^1_{2,3} & w^1_{2,4} \\ b^1_1 & b^1_2 & b^1_3 & b^1_4 \end{pmatrix}$$

$$\eta^1 = \bar{u}^0 \bar{W}^1 = (u^0_1, u^0_2, 1) \begin{pmatrix} w^1_{1,1} & w^1_{1,2} & w^1_{1,3} & w^1_{1,4} \\ w^1_{2,1} & w^1_{2,2} & w^1_{2,3} & w^1_{2,4} \\ b^1_1 & b^1_2 & b^1_3 & b^1_4 \end{pmatrix}$$

$$u^1 = s(\eta^1) = (u^1_1, u^1_3, u^1_3, u^1_4)$$

La funzione di attivazione s è una sigmoide o l'identità

$$\mathbf{u}^1 = (u_1^1, u_3^1, u_3^1, u_4^1) \quad , \quad \bar{\mathbf{u}}^1 = (\mathbf{u}^1, 1)$$

$$W_2 = \begin{pmatrix} w_{1,1}^2 & w_{1,2}^2 & w_{1,3}^2 \\ w_{2,1}^2 & w_{2,2}^2 & w_{2,3}^2 \\ w_{3,1}^2 & w_{3,2}^2 & w_{3,3}^2 \\ w_{4,1}^2 & w_{4,2}^2 & w_{4,3}^2 \end{pmatrix} \qquad \bar{W}_2 = \begin{pmatrix} w_{1,1}^2 & w_{1,2}^2 & w_{1,3}^2 \\ w_{2,1}^2 & w_{2,2}^2 & w_{2,3}^2 \\ w_{3,1}^2 & w_{3,2}^2 & w_{3,3}^2 \\ w_{4,1}^2 & w_{4,2}^2 & w_{4,3}^2 \\ b_1^2 & b_2^2 & b_3^2 \end{pmatrix}$$

$$\eta^2 = \bar{\mathbf{u}}^1 \bar{W}^2 = (u_1^1, u_3^1, u_3^1, u_4^1, 1) \begin{pmatrix} w_{1,1}^2 & w_{1,2}^2 & w_{1,3}^2 \\ w_{2,1}^2 & w_{2,2}^2 & w_{2,3}^2 \\ w_{3,1}^2 & w_{3,2}^2 & w_{3,3}^2 \\ w_{4,1}^2 & w_{4,2}^2 & w_{4,3}^2 \\ b_1^2 & b_2^2 & b_3^2 \end{pmatrix}$$

$$\mathbf{u}^2 = s(\eta^2) = (u_1^2, u_2^2, u_3^2)$$

$$u^2 = s(\eta^2) = (u_1^2, u_2^2, u_3^2), \quad , \quad \bar{u}^2 = s(\eta^2) = (u_1^2, u_2^2, u_3^2, 1)$$

$$W_3 = \begin{pmatrix} w_{1,1}^2 \\ w_{1,2}^2 \\ w_{1,3}^2 \end{pmatrix} \qquad \bar{W}_3 = \begin{pmatrix} w_{1,1}^3 \\ w_{1,2}^2 \\ w_{1,3}^3 \\ b_1^3 \end{pmatrix}$$

$$\eta^3 = \bar{u}^2 \bar{W}^3 = (u_1^2, u_2^2, u_3^2, 1) \begin{pmatrix} w_{1,1}^3 \\ w_{1,2}^2 \\ w_{1,3}^3 \\ b_1^3 \end{pmatrix}$$

$$u^3 = s(\eta^3) = (u_1^3) = y$$

La rete viene adattata ai dati mediante Le formule di aggiornamento all'indietro.

$$\bar{W}^3 = \bar{W}^3 - l_r(\delta^3 \bar{u}^2)^T$$

$$\bar{W}^2 = \bar{W}^2 - l_r(\delta^2 \delta^3 \bar{u}^1)^T$$

$$\bar{W}^1 = \bar{W}^1 - l_r(\delta^1 \delta^2 \delta^3 \bar{u}^0)^T$$

dove

l_r è il learning rate, il passo con cui si muove il metodo del gradiente.

$$D_2 = \begin{pmatrix} u_1^2(1-u_1^2) & 0 & 0 \\ 0 & u_2^2(1-u_2^2) & 0 \\ 0 & 0 & u_3^2(1-u_3^2) \end{pmatrix}$$

$$D_1 = \begin{pmatrix} u_1^1(1-u_1^1) & 0 & 0 & 0 \\ 0 & u_2^1(1-u_2^1) & 0 & 0 \\ 0 & 0 & u_3^1(1-u_3^1) & 0 \\ 0 & 0 & 0 & u_4^1(1-u_4^1) \end{pmatrix}$$

e

$$\delta^3 = (u3 - t)$$

$$\delta^2 = D_2 W^3 = \begin{pmatrix} u_1^2(1-u_1^1) & 0 & 0 \\ 0 & u_2^2(1-u_2^1) & 0 \\ 0 & u_3^2(1-u_3^2) & 0 \end{pmatrix} \begin{pmatrix} w_{1,1}^2 \\ w_{1,2}^2 \\ w_{1,3}^2 \end{pmatrix}$$

$$\delta^1 = D_1 W^2 = \begin{pmatrix} u_1^1(1-u_1^1) & 0 & 0 & 0 \\ 0 & u_2^1(1-u_2^1) & 0 & 0 \\ 0 & 0 & u_3^1(1-u_3^1) & 0 \\ 0 & 0 & 0 & u_4^1(1-u_4^1) \end{pmatrix} \begin{pmatrix} w_{1,1}^2 & w_{1,2}^2 & w_{1,3}^2 \\ w_{2,1}^2 & w_{2,2}^2 & w_{2,3}^2 \\ w_{3,1}^2 & w_{3,2}^2 & w_{3,3}^2 \\ w_{4,1}^2 & w_{4,2}^2 & w_{4,3}^2 \end{pmatrix}$$