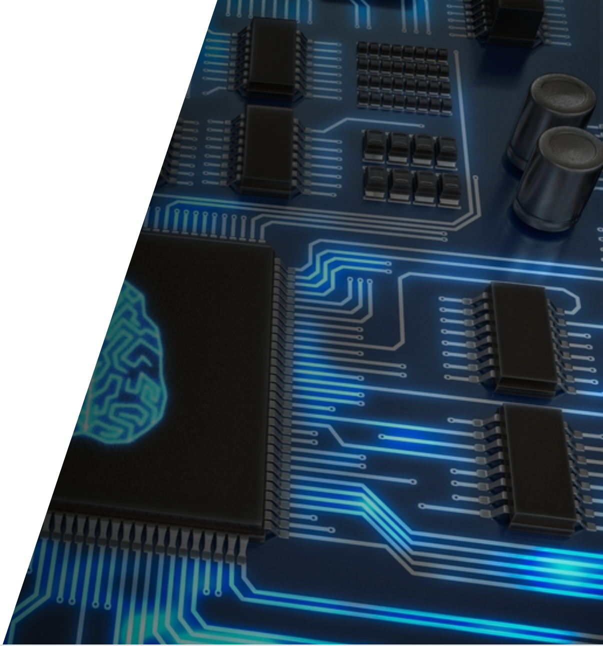
Le reti neurali



Samuele Donzelli
Zaccaria Eliseo Carrettoni

Politecnico di Milano
July 9, 2020

Sommario:

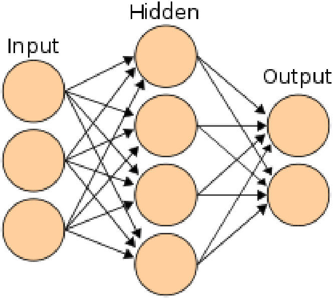
- Obiettivo: analizzare la struttura ed il funzionamento delle reti neurali artificiali. Qui sarà presentato principalmente elemento che le compone ovvero il **neurone artificiale**.
- Tecnica usata per addestrare le reti: **back propagation**, e spiegazione dei modelli matematici alla base.
- Rete neurale implementata su Matlab.

Indice (provvisorio)

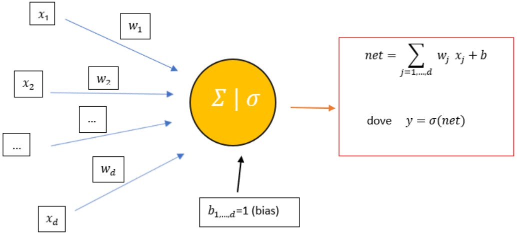
- introduzione
- generalità sulle reti neurali
- algoritmo di forewar propagation
- algoritmo di back propagation
- commento sulle parti principali del codice prodotto

Introduzione

Le reti neurali sono formate da gruppi di neuroni artificiali organizzati in **livelli**; tipicamente sono presenti un livello di **input** un livello di **output** e uno o più livelli intermedi o nascosti (**hidden**). Ogni livello contiene uno o più neuroni. La tipologia di reti che andremo ad analizzare è quella delle reti **Feedforward** (alimentazione in avanti).



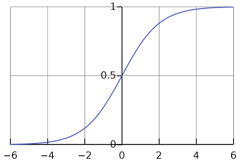
Cos’è un neurone artificiale?



- $x_1, \dots, x_d$ sono i **d** ingressi del neurone.
- $w_1, \dots, w_d$ sono i pesi (weight) che determinano l’efficacia delle connessioni tra neuroni.
- $b_1, \dots, b_d$ è il **bias** è un ulteriore peso che si considera collegato a un input fittizio con valore sempre 1.
- $net_1, \dots, net_d$ è il livello di eccitazione globale del neurone.
- σ: è la funzione di attivazione.

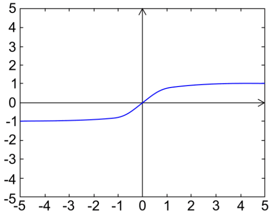
Funzione di attivazione (σ, τ)

- **standard logistic function,** valori in [0,1] con $\sigma(net) = 1/(1 + e^{-net})$.



La derivata: $\sigma'(net) = \sigma(net)(1 - \sigma(net))$

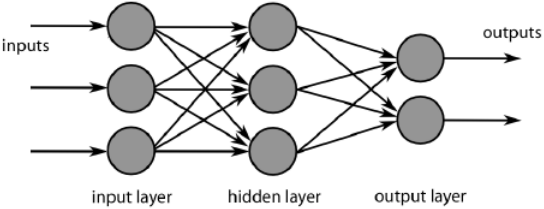
- **tangente iperbolica,** (valori in [-1,1]) con $\tau(net) = tanh(net)$.



La derivata: $\tau'(net) = 1 - \tau(net)^2$

Le reti feedforward

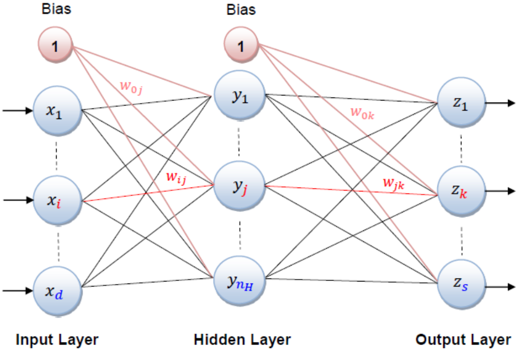
In tali reti le connessioni collegano i neuroni di un livello con i neuroni di un livello successivo. Non sono consentite connessioni all'indietro o connessioni verso lo stesso livello.



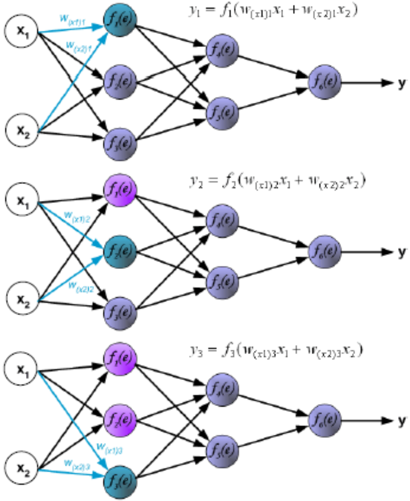
Forward propagation

Si intende la propagazione delle informazioni in avanti dal livello di input a quello di output.

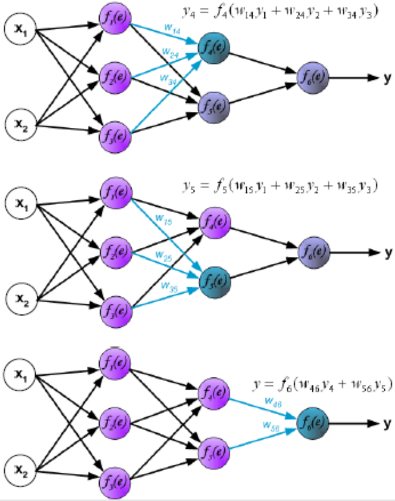
$$z_k = \sigma\left(\sum_{j=1\ldots n_H} w_{jk}\, y_j + b_k\right) = \sigma\left(\sum_{j=1\ldots n_H} w_{jk}\, \sigma\left(\sum_{i=1\ldots d} w_{ij}\, x_i + b_j\right) + b_k\right)$$



Forward propagation: esempio grafico

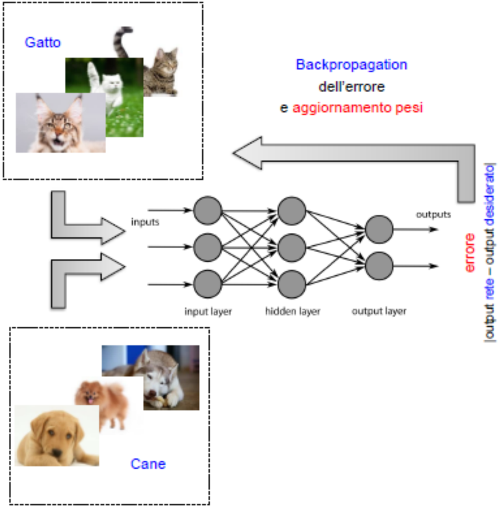


e continua..



$y_4 = f_4(w_{14}y_1 + w_{24}y_2 + w_{34}y_3)$
$y_5 = f_5(w_{15}y_1 + w_{25}y_2 + w_{35}y_3)$
$y = f_6(w_{46}y_4 + w_{56}y_5)$

Training


Gatto
Backpropagation
dell'errore
e aggiornamento pesi
inputs
outputs
input layer
hidden layer
output layer
errore
|output rete – output desiderato|
Cane

Training supervisionato di un classificatore

Problema di classificazione supervisionata:

- **s** classi con dimensione dei pattern **d** $\longrightarrow$ rete con **d**:n_H:**s** neuroni in tre livelli.
- La **prenormalizzazione** dei pattern di input favorisce la convergenza durante l'addestramento.
- vettore di **output** desiderato (usando σ) assume la forma: $y^L = [0, 0, ..\mathbf{1}..0]$
- **obiettivo:** modificare i pesi della rete in modo da minimizzare l'errore medio sui pattern del training set.
- **n.b.:** Prima dell'inizio dell'addestramento i pesi sono tipicamente inizializzati con valori **random**:
  - input hidden in $\pm 1/\sqrt{d}$.
  - output hidden in $\pm 1/\sqrt{n_H}$.

Error backpropagation

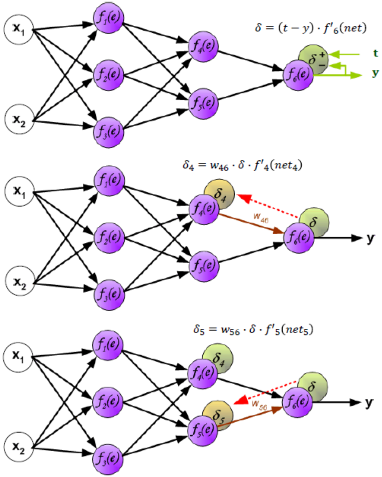
- **z=[**$z_1, ..., z_s$**]** l'output prodotto dalla rete, **x=[**$x_1, ..., x_d$**]** gli ingressi in input e **t=[**$t_1, ..., t_d$**]** è l'output desiderato.
- **Loss function:** somma dei quadrati degli errori (per il pattern "x") definita come:

$$J(\mathbf{w}, \mathbf{x}) \equiv \frac{1}{2} \sum_{c=1...s} (t_c - z_c)^2$$

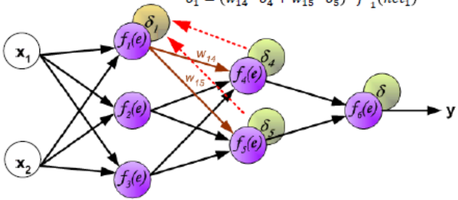
- J(w) può essere **ridotto** modificando i pesi **w** in direzione **opposta al gradiente di J.**

Backpropagation: esempio grafico

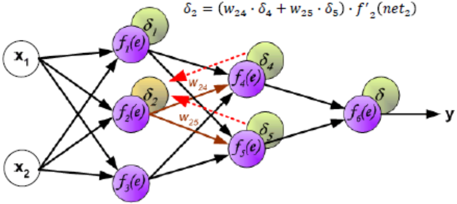
Stessa rete a 4 livelli su cui abbiamo eseguito forward progpagation:



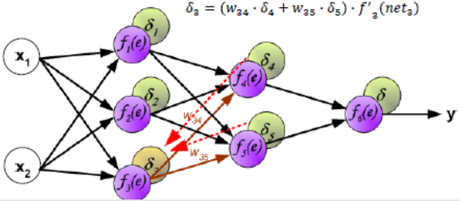
$$\delta_1 = (w_{14} \cdot \delta_4 + w_{15} \cdot \delta_5) \cdot f'_1(net_1)$$



$$\delta_2 = (w_{24} \cdot \delta_4 + w_{25} \cdot \delta_5) \cdot f'_2(net_2)$$



$$\delta_3 = (w_{34} \cdot \delta_4 + w_{35} \cdot \delta_5) \cdot f'_3(net_3)$$



e infine: aggiornamento pesi

$$w'_{(x1)1} = w_{(x1)1} + \eta \cdot \delta_1 \cdot x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta \cdot \delta_1 \cdot x_2$$