

# Reti neurali feedforward

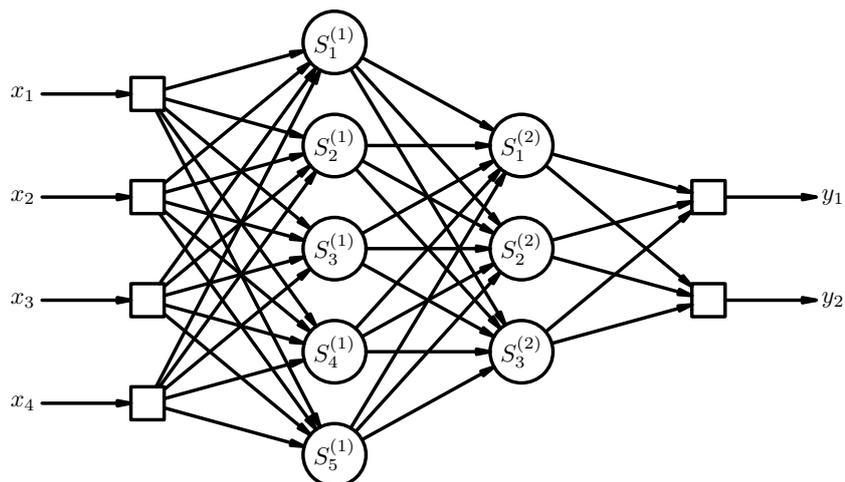
Stefano Ferrari

Università degli Studi di Milano  
stefano.ferrari@unimi.it

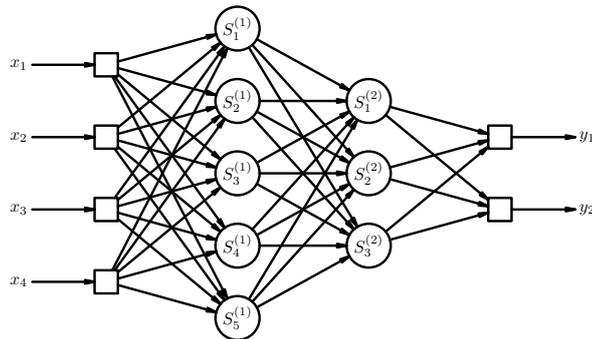
## Reti Neurali

Stage 2016

### Rete neurale feedforward



## Reti feedforward

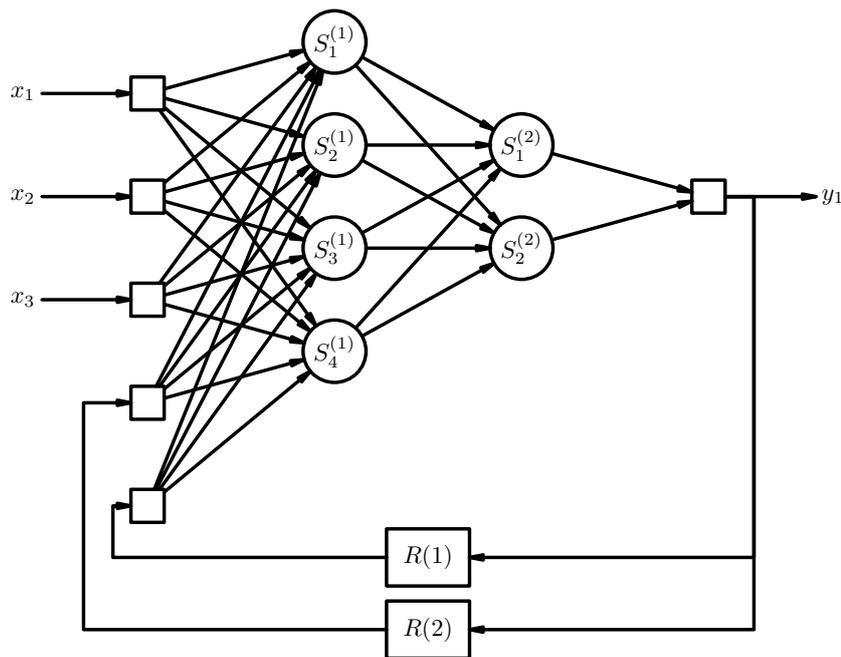


- ▶ Nelle reti feedforward, i neuroni sono organizzati in strati.
- ▶ Ogni strato passa i segnali allo strato seguente.

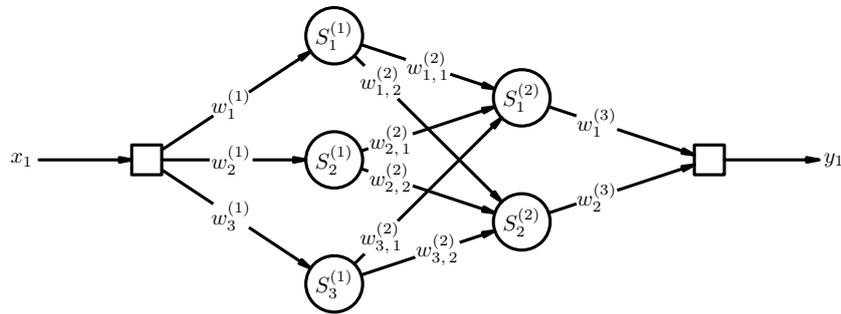
Una rete feedforward è composta da:

- ▶ uno strato di ingresso;
  - ▶ distribuisce i segnali di input ai neuroni del primo strato;
- ▶ uno o più strati intermedi (detti *strati nascosti*, *hidden layers*);
  - ▶ effettuano una trasformazione (generalmente) non lineare;
- ▶ uno strato di uscita;
  - ▶ la cui funzione di attivazione caratterizza il compito realizzato dalla rete.

## Rete neurale ricorrente



## Parametri di una rete feedforward



## Backpropagation

- ▶ Le reti multistrato con funzione di attivazione derivabile consentono l'utilizzo della tecnica di retropropagazione (*backpropagation*) dell'errore.
- ▶ L'adattamento dei parametri interni della rete richiede:
  - ▶ la valutazione dell'errore commesso dalla rete;
  - ▶ la stima della modifica da apportare ai parametri per diminuire l'errore;
  - ▶ la ripetizione di queste operazioni per ogni strato.
- ▶ L'idea alla base del metodo è il calcolo del gradiente dell'errore rispetto a ciascuno parametro, in modo da individuare la direzione lungo la quale modificare il parametro per muovere la rete verso un minimo (locale) dell'errore.

## Backpropagation (2)

Esempio:

- ▶ Ipotezziamo una rete composta da:
  - ▶ strato di ingresso lineare:  $S_1^{(0)} = x$ ;
  - ▶ due strati nascosti:  $S_j^{(1)} = f^{(1)}\left(\sum_k w_j^{(1)} x - \theta_j^{(1)}\right)$  e  $S_i^{(2)} = f^{(2)}\left(\sum_j w_{i,j}^{(2)} S_j^{(1)} - \theta_i^{(2)}\right)$ ;
  - ▶ strato finale lineare:  $\tilde{y} = \sum_i w_i^{(3)} S_i^{(2)}$ .
- ▶ La presentazione alla rete dell'esempio  $(x, y)$  comporta un errore,  $E$ :
  - ▶  $E = (y - \tilde{y})^2$ , che dipende dal valore dei parametri;
  - ▶ questa definizione dell'errore è indipendente dal segno e derivabile;
  - ▶ la derivata dell'errore rispetto ad uno dei parametri della rete fornisce una stima della variazione dell'errore a fronte di una piccola modifica del valore del parametro.

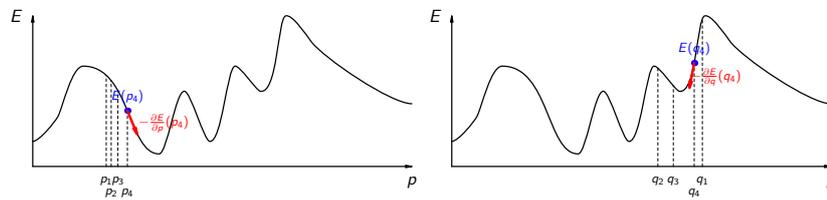
## Backpropagation (3)

Esempio di derivata:



- ▶ in senso orario fa più freddo ...
- ▶ in senso anti-orario fa più caldo ...
- ▶ ma di quanto?

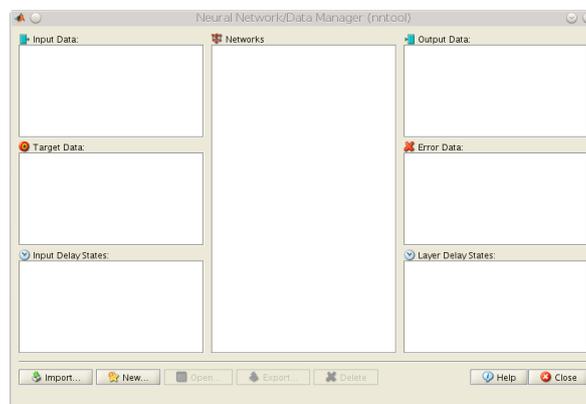
## Addestramento



- ▶ Si conosce il valore attuale dell'errore e anche in che direzione diminuisce.
- ▶ Non si sa però quanto possa diminuire (minimo globale).
- ▶ Né si sa di quando cambia l'errore se si modificano i parametri in un certo modo (lunghezza del passo di addestramento).

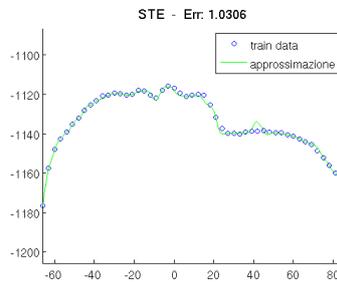
## Reti feed-forward per approssimare una funzione

1. Scaricare  
[http://homes.di.unimi.it/ferrari/RetiNeurali\\_2016/reti\\_neurali\\_03.zip](http://homes.di.unimi.it/ferrari/RetiNeurali_2016/reti_neurali_03.zip)
2. Scompattare sul desktop
3. Eseguire *nntool*



## Reti feed-forward per approssimare una funzione $\mathbb{R} \rightarrow \mathbb{R}$

1. Caricare i dati da *profilof.mat*
2. Usando *nnTool*, creare e addestrare una rete
  - ▶ importare *x* e *xr* come "Input data"
  - ▶ importare *y* come "Target data"
  - ▶ creare una rete feed-forward
  - ▶ usare *x* e *y* per il training
  - ▶ impostare *y\_out* come output della rete
  - ▶ simulare la rete usando *xr* come input e *yr\_out* come output
  - ▶ esportare *y\_out* e *yr\_out*
3. Aprire *plot\_risultati.m* e modificare la variabile *NOME*
4. Da Matlab, eseguire *plot\_risultati*



## Reti feed-forward per approssimare una funzione $\mathbb{R}^2 \rightarrow \mathbb{R}$

1. Caricare i dati da *ciccio\_3d.mat*
2. Usando *nnTool*, creare e addestrare una rete
  - ▶ importare *P* e *Pr* come "Input data"
  - ▶ importare *T* come "Target data"
  - ▶ creare una rete feed-forward
  - ▶ usare *P* e *T* per il training
  - ▶ impostare *T\_out* come output della rete
  - ▶ simulare la rete usando *Pr* come input e *Tr\_out* come output
  - ▶ esportare *T\_out* e *Tr\_out*
3. Aprire *plot\_risultati\_3d.m* e modificare la variabile *NOME*
4. Da Matlab, eseguire *plot\_risultati\_3d*

