Intelligenza Artificiale

Reti Neurali

Alberto Broggi

Reti Neurali: storia

- Per costruire un sistema intelligente possiamo cercare di simulare l'unico modello esistente in natura:

il cervello

- Le prime idee risalgono a McCulloch e Pitts (1943)
- Approfondimenti dovuti a molti studiosi (1950-1960)
- Furono abbandonate negli anni settanta a causa di problemi computazionali
- Interesse rinnovato alla fine degli anni '80

Reti Neurali: storia

- Negli anni '80 la tecnologia VLSI consente di produrre architetture di elaborazione innovative

- I calcolatori a parallelismo massivo hanno le caratteristiche adatte per le reti neurali:
  - un elevato numero di elementi di elaborazione indipendenti e semplici
  - gli elementi di elaborazione sono collegati tra loro
  - gli elementi di elaborazione lavorano in modo parallelo in modalità SIMD

Reti Neurali

- L'obiettivo delle reti neurali è di simulare il cervello in modo da:
  - **riconoscere certe situazioni**:
    ricercare nella propria memoria un evento uguale all'evento in ingresso
  - **gestire le situazioni incerte**:
    ricercare nella propria memoria un evento simile all'evento in ingresso: *generalizzazione*

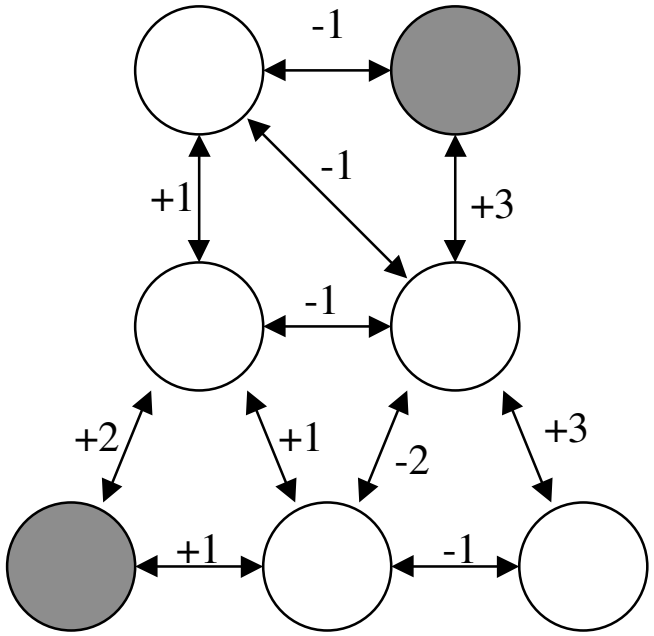
Reti di Hopfield

- Hopfield (1982) propone un sistema per memorizzare certi tipi di informazioni;
  questo sistema ha le seguenti caratteristiche:
  - **Rappresentazione distribuita:**
    informazioni diverse possono condividere lo stesso spazio fisico
  - **Controllo asincrono distribuito:**
    ogni elemento prende decisioni locali che si sommano per formare il quadro globale
  - **Memoria indirizzabile per contenuto:**
    per recuperare una informazione è sufficiente produrne solo una porzione
  - **Tolleranza ai guasti:**
    se alcuni elementi si comportano in modo scorretto o si guastano, il sistema deve continuare a funzionare

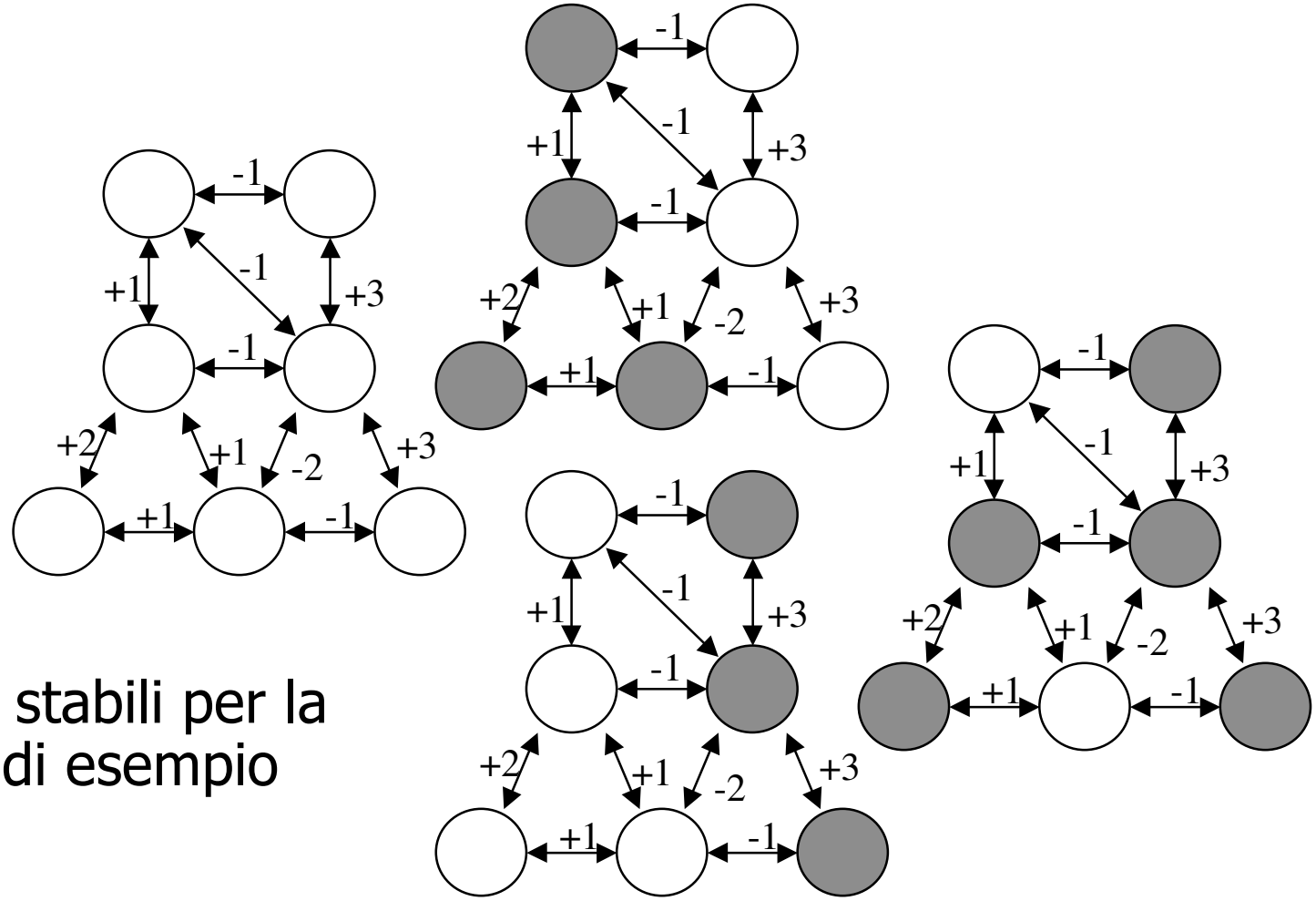
Reti di Hopfield

Rilassamento parallelo

- si sceglie un elemento a caso;
- si calcola la somma dei pesi provenienti da connessioni con vicini attivi;
- se la somma è positva l'elemento viene attivato, altrimenti si disattiva.



Reti di Hopfield



- Stati stabili per la rete di esempio

Reti di Hopfield

- Hopfield ha dimostrato che partendo da qualunque stato casuale l'algoritmo di rilassamento parallelo porta il sistema ad uno stato stabile, senza divergenze e senza oscillazioni

- Inoltre, partendo da una configurazione non stabile, il sistema si stabilizza verso lo schema più vicino

- Se un'unità di elaborazione si guasta non genera problemi; i problemi si hanno se si guasta un cluster di unità adiacenti

Reti di Hopfield

- Cosa codificano i pesi e le connessioni ?

  La *conoscenza* del sistema

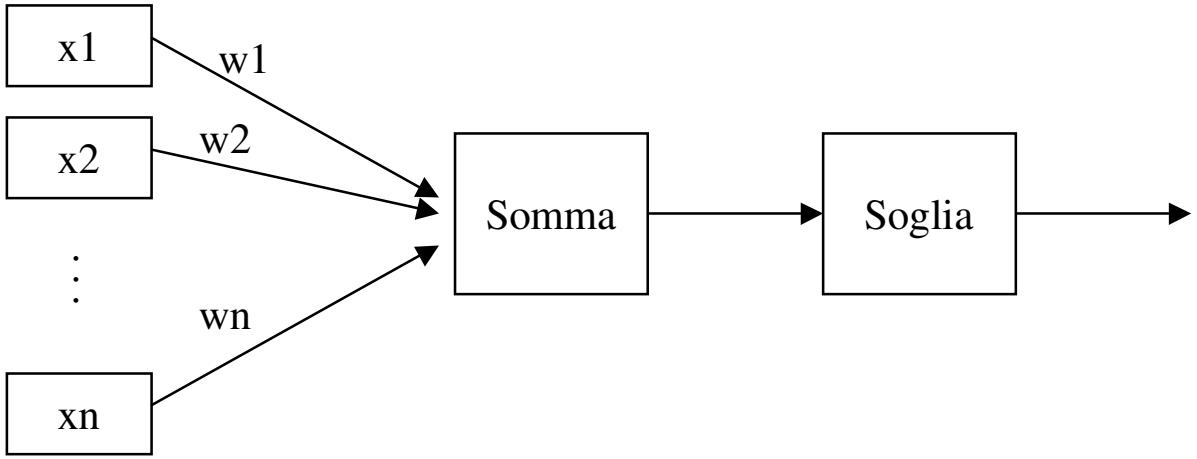
- Come viene acquisita questa conoscenza ?

  Mediante opportuno *training*

Il Percettrone

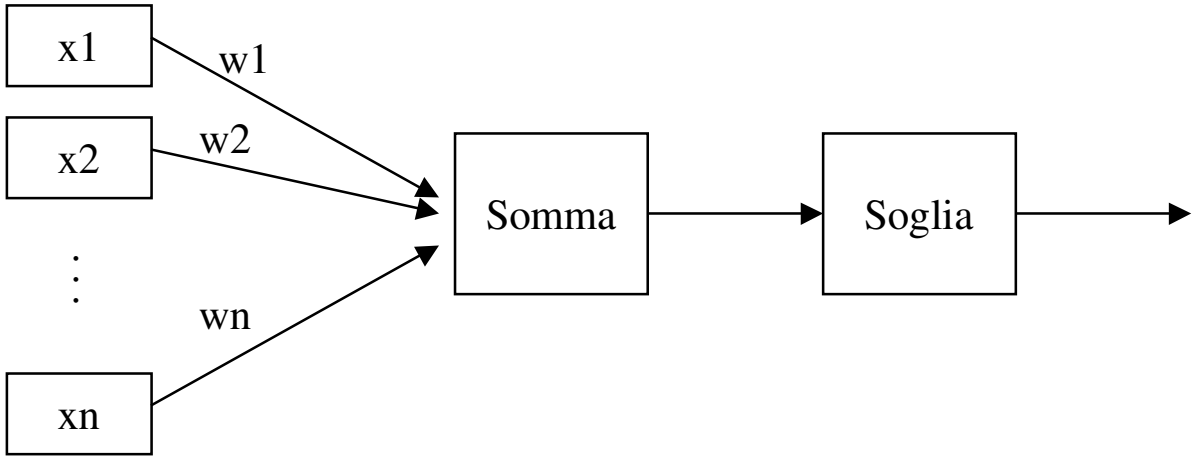
- Il percettrone è il mattone base delle reti neurali
- Nasce da un'idea di Rosenblatt (1962)

- Cerca di simulare il funzionamento del neurone



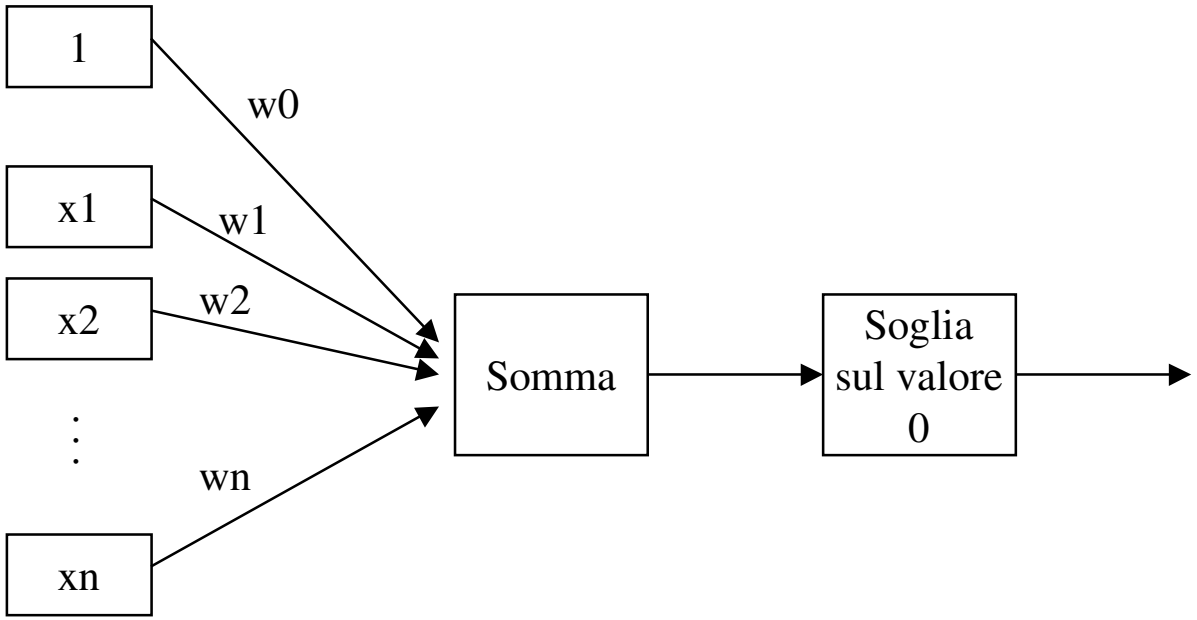
Il Percettrone

- I valori di uscita sono digitali: 0 oppure 1
- Gli ingressi *xi* e i pesi *wi* sono valori reali positivi o negativi
- Ingressi, somma, soglia:
- L'apprendimento consiste nel modificare pesi e soglia



Il Percettrone

- Talvolta si elimina il valore di soglia inserendo un nuovo ingresso (ad 1) con un proprio peso



Il Percettrone con più uscite

- E' possibile generalizzare e realizzare un percettrone dotato di molti ingressi e molte uscite
- Tutti gli ingressi sono collegati a tutti i sommatori
- I percettroni sono tutti indipendenti e pertanto possono essere addestrati singolarmente

Il Percettrone

- Problema:
  - dati insiemi di punti su uno spazio 2D, classificarli in due gruppi
  - inoltre dato un nuovo punto (x,y) decidere a quale gruppo appartiene

- Il primo problema è di classificazione, mentre per risolvere il secondo è richiesta capacità di generalizzazione, simile all'apprendimento di concetti;
- non vi è una regola, si deve imparare da esempi

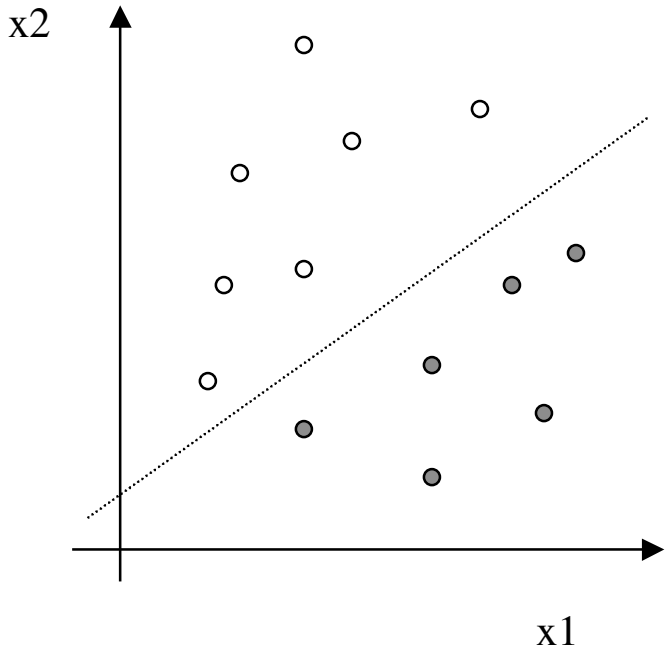
Problema di classificazione

- Con (x1, x2) in ingresso, si ha:

  $g(x) = w0 + w1\ x1 + w2\ x2$

  mentre l'uscita è data da:

  1 se $g(x) > 0$
  0 se $g(x) < 0$



- La linea di separazione è data da:
  $x2 = - (w1/w2)\ x1 - (w0/w2)$

Problema di classificazione

- La retta è completamente determinata da w0, w1, e w2
- La retta si chiama *superficie di decisione*
- Il problema è di identificare una superficie di decisione opportuna
- Il segno di g(x) è usato per la classificazione, ma il suo modulo è importante per capire quanto può essere giusta la classificazione (quanto si è distanti dalla superficie di decisione)

Problema di classificazione

- Il problema è quindi ridotto alla determinazione dell'insieme dei pesi (w0, w1,… wn) migliore per minimizzare gli errori di classificazione
- E' un problema di ottimizzazione
- Viene utilizzata una tecnica di discesa nella direzione del gradiente

Il Percettrone

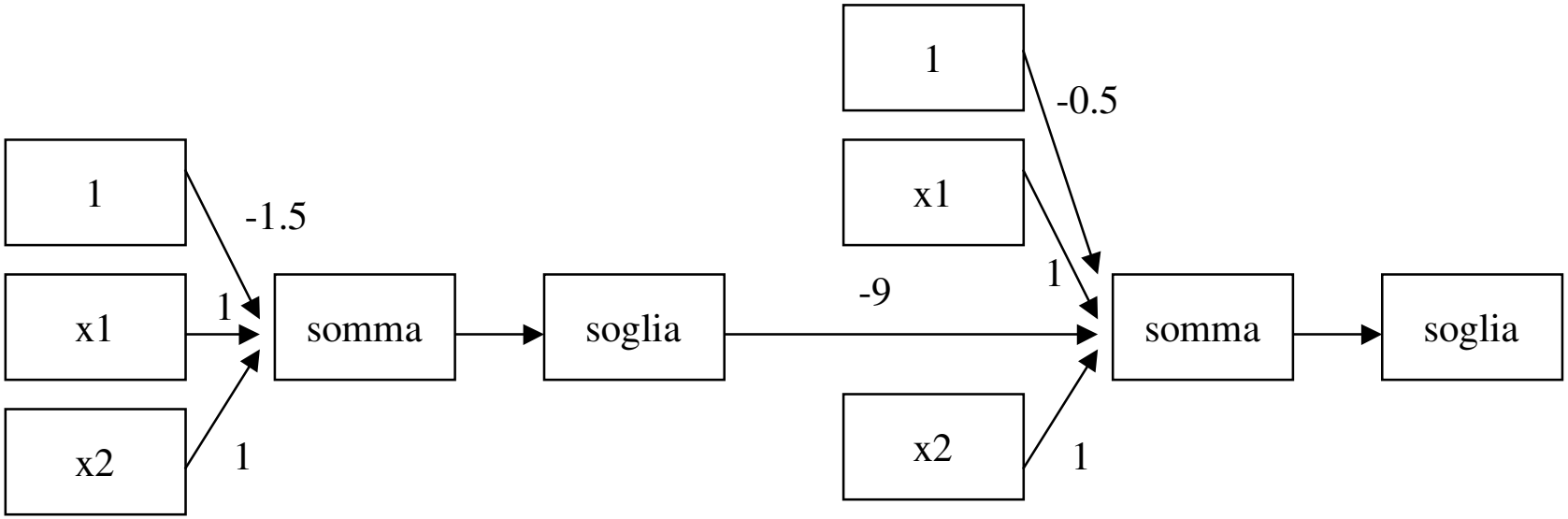
- Il *teorema di convergenza del percettrone* (Rosemblatt, 1962) assicura che il percettrone riuscirà a delimitare le 2 classi in un sistema *linearmente separabile*
- In altre parole, nell'ottimizzazione non esistono minimi locali

- Problema: come ci si comporta in caso di situazioni non linearmente separabili?
  Ad esempio il problema della disgiunzione (XOR)

Percettrone multistrato

- Nei casi non linearmente separabili si usano più percettroni in cascata (o multistrato)

- Equivale ad avere due rette per isolare i due punti



Percettrone multistrato

- Il percettrone multistrato risolve il problema della rappresentazione della conoscenza ma introduce il problema dell'apprendimento:

  il teorema di convergenza non si estende ai percettroni multistrato

- E' necessario determinare un metodo per far apprendere anche i percettroni multistrato

Reti con propagazione all'indietro (back propagation)

- Obiettivi:
  - partire da un insieme di percettroni con pesi casuali
  - insegnare in modo veloce
  - avere capacità di generalizzazione
  - avere insiemi di percettroni su larga scala

- Reti a 2 strati, a 3 strati. A 4 strati?
- Strati di ingresso, di uscita, e nascosto

Reti multistrato

- Per l'apprendimento non c'è bisogno di passi iterativi, ma è sufficiente un procedimento a ritroso, partendo dalle uscite, fino a giungere agli ingressi

- La rete viene addestrata con un insieme di coppie ingresso-uscita. E' auspicabile che riesca ad avere anche buona capacità di generalizzazione

Reti multistrato

- Per poter procedere all'indietro, tutte le funzioni devono essere invertibili
- Purtroppo la funzione *soglia* non lo è
- E' necessario quindi sostituirla con una funzione simile, ma invertibile

- La scelta è caduta sulla sigmoide, che ha buone caratteristiche di derivabilità e continuità:

  uscita = 1 / (1+e ^(-somma))

Reti multistrato

- Purtroppo non esiste un equivalente del teorema di convergenza valido per i percettroni
- E' necessario trovare dei modi per eludere i minimi locali.

- Fortunatamente per reti di grandi dimensioni i minimi locali non sono un problema perché il numero di nodi (e quindi il numero di gradi di libertà) è elevato.
- Caratteristica scoperta solo di recente grazie ai calcolatori veloci

Generalizzazione

- Alla rete si forniscono delle coppie ingresso-uscita
- Per molti problemi, però, è impossibile fornire tutti i possibili ingressi (riconoscimento dello scritto, dei volti,...)
- E' richiesta quindi alla rete una capacità di generalizzazione

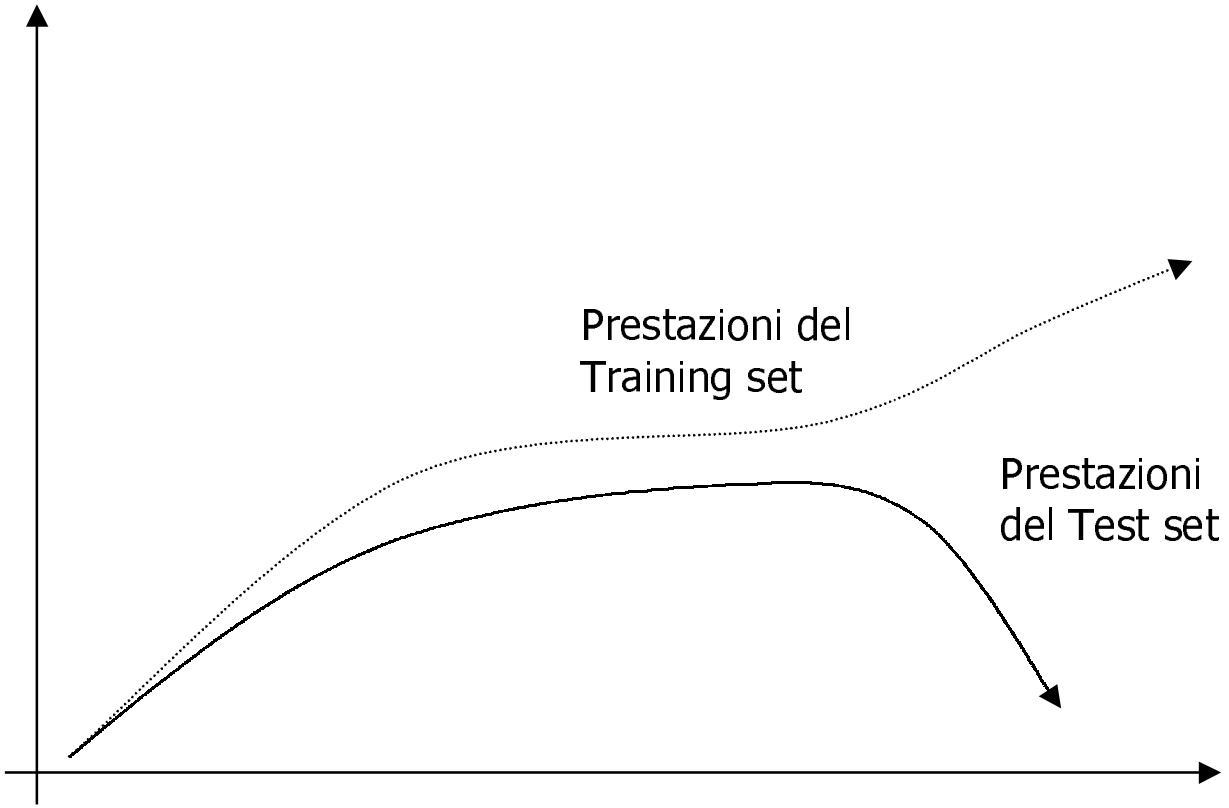
- Esistono però svariati problemi

Generalizzazione

- Con un addestramento troppo lungo la rete non **impara**, ma **memorizza**

- Ci sono diversi metodi per impedire la memorizzazione:
  - non effettuare addestramento troppo lungo
  - aggiungere rumore agli ingressi (ma non troppo da confondere il classificatore)
  - ridurre il numero di unità di elaborazione del sistema (e quindi i pesi): si può agire sul numero di nodi nascosti: la rete dovrà trovare un modo più compatto di rappresentazione

Reti ricorrenti

- Introducono l'effetto memoria
- Hanno complessità elevata
- Difficile convergenza

Problemi del Training