

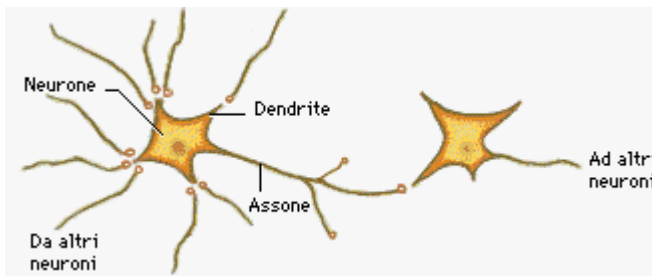


# Data Mining

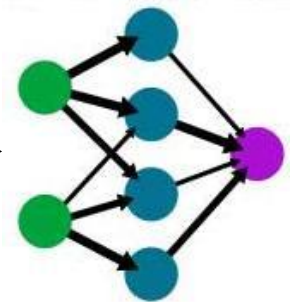
Artificial Neural Networks

# ANN: Definizione

- Una rete neurale artificiale definisce un **modello matematico** per la simulazione di una rete di neuroni biologici (es. sistema nervoso umano)
- Simula i diversi aspetti legati al comportamento e alle capacità del cervello umano:
  - Elaborazione intelligente delle informazioni
  - Elaborazione distribuita
  - Elevato livello di parallelismo
  - Facoltà di apprendimento, di generalizzazione e di adattamento
  - Alta tolleranza ad informazioni poco precise (o sbagliate)



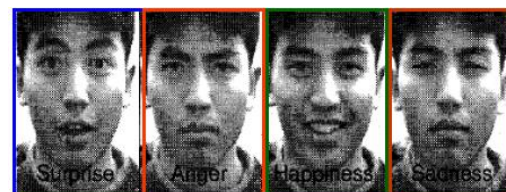
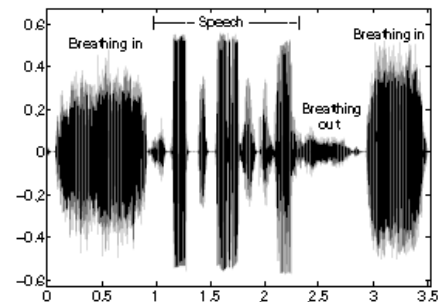
Rete neurale biologica



Rete neurale artificiale<sup>2</sup>

# Applicazioni (1)

- Sistemi di filtraggio del segnale (es. eliminare il rumore da un segnale)
- Sistemi di classificazione
- Previsione di serie temporali (Time Series Forecasting)
- Riconoscimento (es. riconoscimento volti o delle espressioni facciali)



# Applicazioni (2)

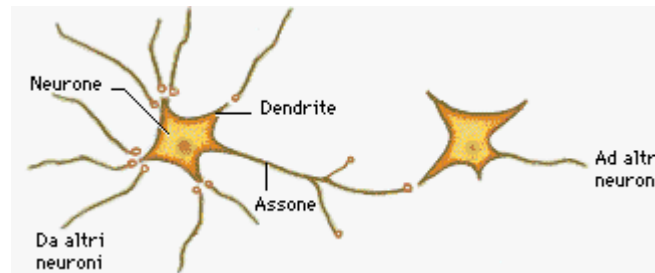
- Problemi di inversione (es. rete per l'inversione di dati satellitari per estrarre diversi tipi di profili atmosferici)
- Sistemi di allarme e prevenzione (es. sistema per il monitoraggio del manto autostradale per allertare con sufficiente anticipo i mezzi di spargimento delle misture saline in caso di pericolo)
- Diagnosi di guasto



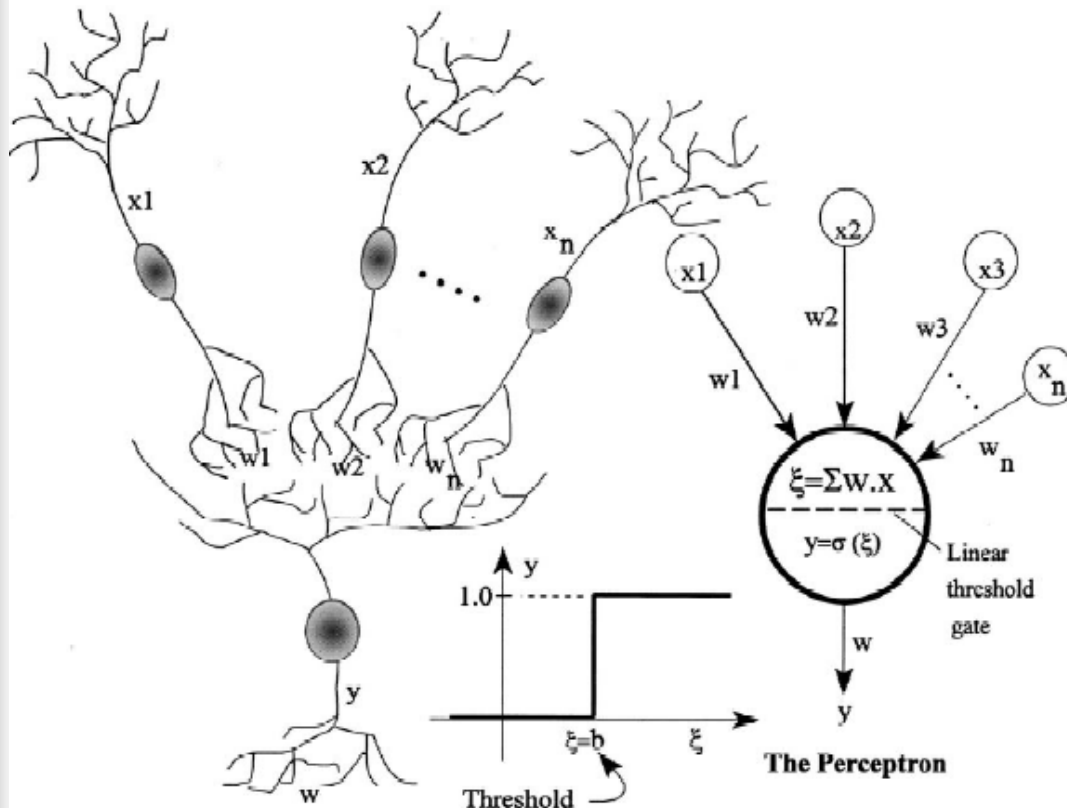
# Trasmissione informazioni (2)

La quantità di segnale (informazione) trasmessa verso il neurone postsinaptico dipende da diversi fattori:

- **Intensità del segnale in entrata** (anche ricezione di più segnali da neuroni diversi)
- **Forza delle sinapsi dei neuroni in entrata** (rilascio dei neurotrasmettitori)
- **Soglia di attivazione del neurone ricevente** (superata una certa soglia di potenziale elettrico, il neurone ricevente si attiva e genera a sua volta un segnale che può essere propagato verso gli altri neuroni)



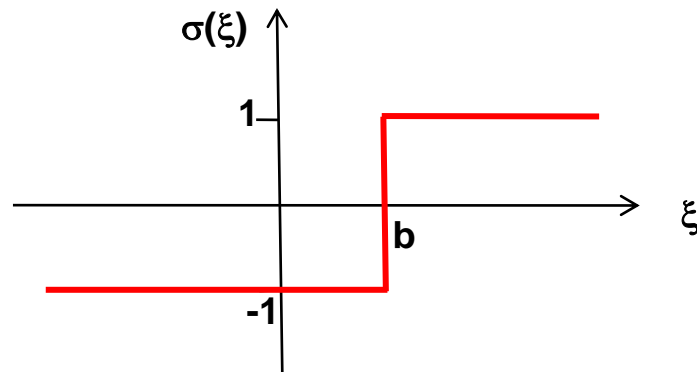
# Rete neurale biologica vs artificiale



- $n$ : neuroni in entrata;
- $x_i$ : intensità del segnale;
- $w_i$ : forza sinapsi;
- $b$ : soglia del neurone postsinaptico
- $\xi$ : combinazione segnali in entrata
- $\sigma$ : funzione di attivazione
- $y$ : output

# Il percettore (1)

- Il percettore è la ANN più semplice, introdotta agli inizi degli anni '50
  - ✓ Classificatore binario
  - ✓ Funzione di attivazione: funzione segno.

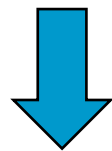


- ✓ Combinazione feature in ingresso: combinazione **lineare** delle feature in ingresso ( $x_i$ ), con i pesi ( $w_i$ ) come **coefficienti**.

# Il perceptrone (2)

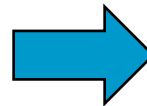
Funzione di attivazione:

$$y' = \begin{cases} 1, & \text{se } \sum_{j=1}^n w_j x_j > b \\ -1, & \text{se } \sum_{j=1}^n w_j x_j < b \end{cases}$$



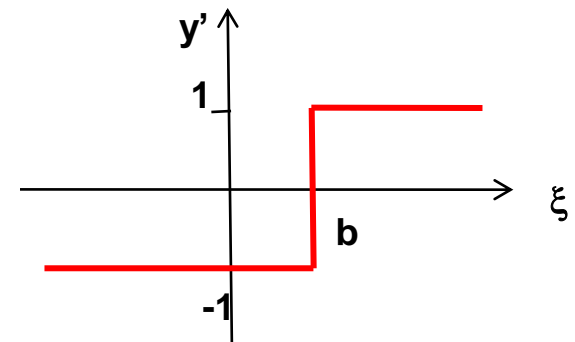
Si considera la **soglia** come **input** con coefficiente 1

$$y' = \begin{cases} 1, & \text{se } \sum_{j=1}^n w_j x_j - b > 0 \\ -1, & \text{se } \sum_{j=1}^n w_j x_j - b < 0 \end{cases}$$



FORMA  
VETTORIALE

$$y' = \text{sign}(W \cdot X)$$



Funzione segno



# Esempio

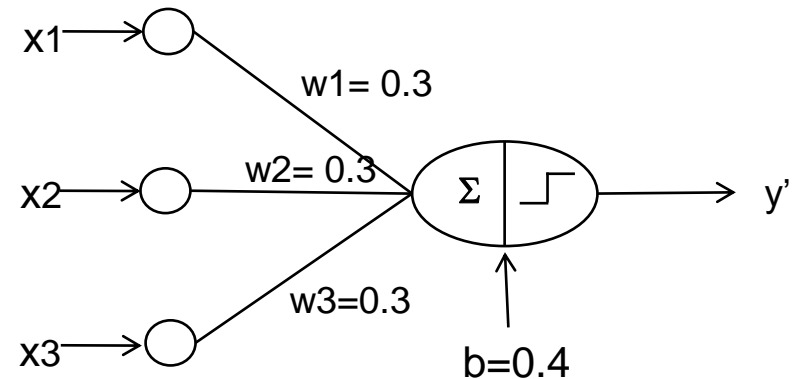
Data set

$x_1$	$x_2$	$x_3$	$y$	$y'$
1	0	0	-1	-1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1
0	0	1	-1	-1
0	1	0	-1	-1
0	1	1	1	1
0	0	0	-1	-1

$y$  assume valore 1 se almeno due input sono maggiori di 0, altrimenti assume valore -1

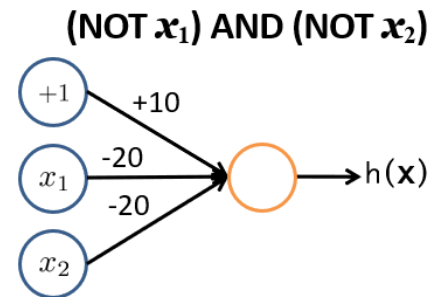
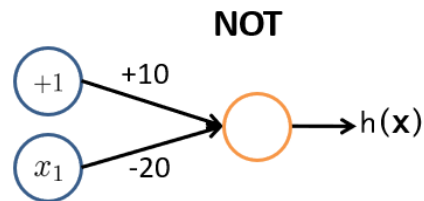
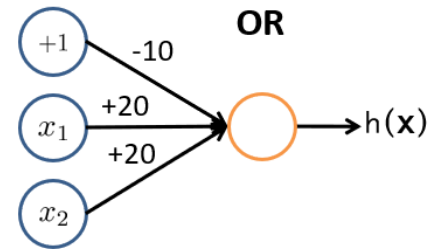
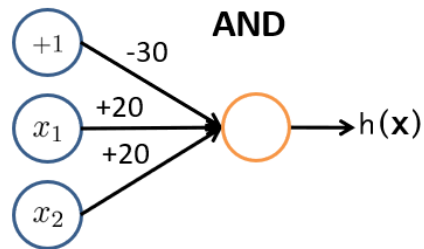
Percettrone

*Classificatore binario*



$$y' = \begin{cases} 1, & \text{se } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0 \\ -1, & \text{se } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0 \end{cases}$$

# Boolean functions



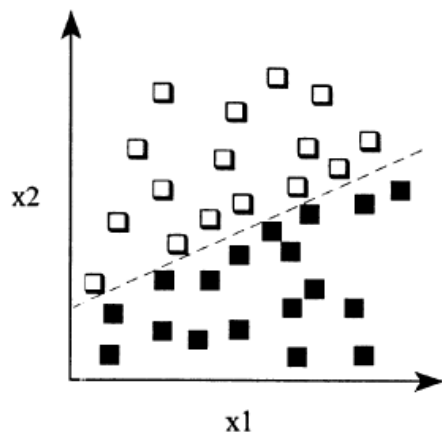


# Apprendimento

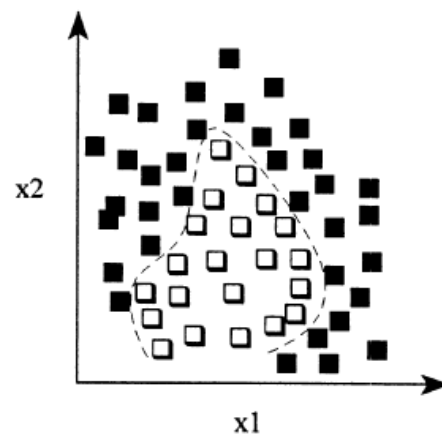
- Concetto generale: processo di aggiornamento della struttura interna di un sistema in risposta a stimoli esterni, al fine di eseguire una specifica attività.
- Reti neurali:
  - ✓ Aggiustare i pesi delle sinapsi al fine di ridurre l'errore fra il valore di output generato dalla rete neurale e l'output corretto
  - ✓ È un processo iterativo: analogamente all'apprendimento per esperienza
  - ✓ Si dice che una rete neurale ha appreso quando:
    - È in grado di memorizzare informazioni imprecise o che contengono rumore, senza che venga alterata la qualità della risposta della rete neurale
    - È in grado di generalizzare da concetti appresi a concetti sconosciuti

# Limiti del perceptrone

- L'algoritmo di apprendimento del perceptrone termina (quindi si raggiunge la soluzione ottima) se il problema è linearmente separabile



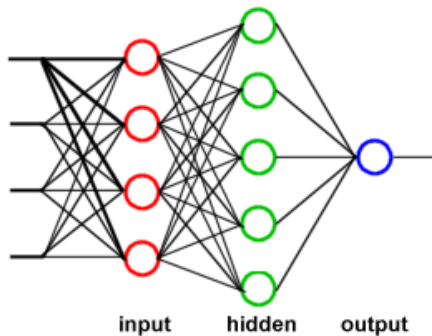
Linearly-separable classes



Nonlinearly-separable classes

# Reti neurali complesse

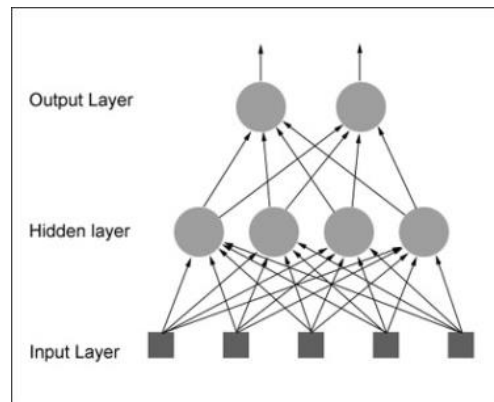
## ■ Percetttrone multilivello



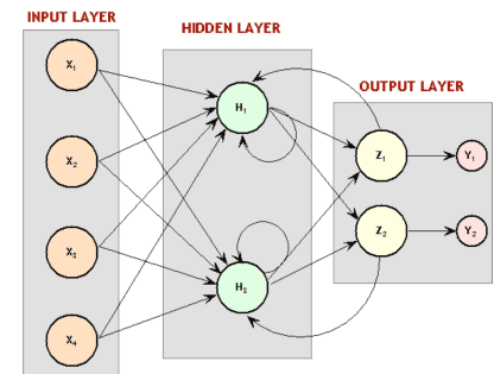
## ■ Funzione di attivazione

## ■ Pattern di connessione

- Feed-forward
- Recurrent



Feed-forward

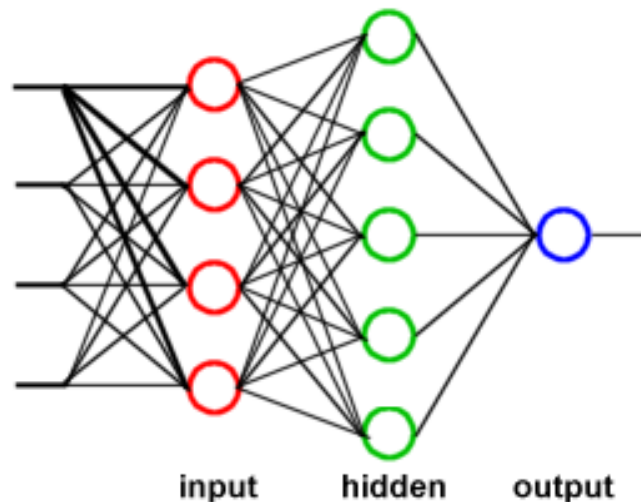


Recurrent

# Percettrone multilivello (1)

La rete è caratterizzata da uno o più livelli intermedi (**livelli nascosti**) fra il livello di input e quello di output.

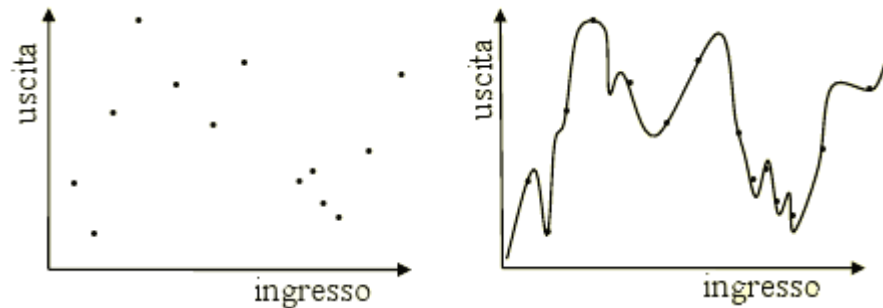
- ✓ Gli output dei nodi di un livello diventano gli input per i nodi del livello successivo.
- ✓ Ogni nodo nascosto si comporta come un singolo percettrone



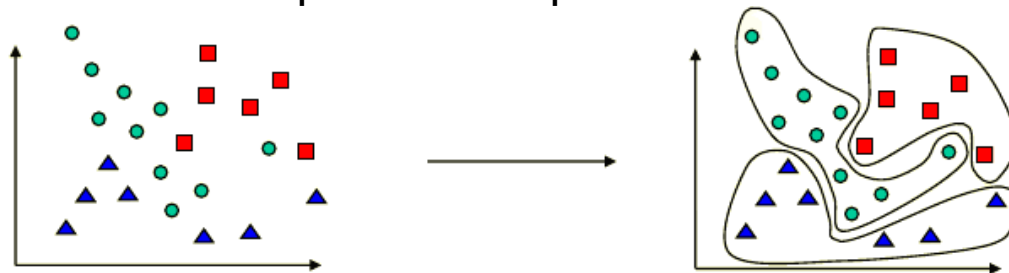
- Quanti livelli nascosti?
  - ✓ 1 livello: **universal approximator** (vedi Teorema Cybenko)
    - Può approssimare in modo arbitrariamente preciso ogni funzione continua (avendo un numero sufficiente di neuroni)

# Vantaggi di una rete neurale (1)

- Capacità di generalizzazione
  - ✓ Utile in problemi di riconoscimento caratteri digitalizzati
- Comportamento dinamico
  - ✓ Capacità di rispondere a stimoli esterni in maniera diversa rispetto a ciò che inizialmente è stato definito all'interno del sistema
- Capacità di riprodurre o modellare comportamenti non lineari di un sistema



- Capacità di classificare pattern complessi





# Vantaggi di una rete neurale (2)

- Tolleranza agli errori
  - ✓ scostamenti non significativi dei valori di input vengono assorbiti dalla rete, diminuendo così il rumore e risultando tolleranti agli errori o al decadimento del segnale
- Tolleranza ai guasti
  - ✓ Il mal funzionamento di una unità di elaborazione può essere ovviato dalla presenza di molteplici neuroni che operano in parallelo
- Capacità di memoria





# Svantaggi di una rete neurale

- Ai fini dell'apprendimento è necessario fornire alla rete neurale tanti più esempi quanto più fortemente non lineare è il sistema da modellare.
  - ✓ Il processo di apprendimento potrebbe risultare lungo
  - ✓ Il processo di apprendimento potrebbe portare a soluzione subottime
- Non è possibile sapere a priori quale sia la miglior topologia della rete (numero di nodi, di strati, collegamenti fra nodi e tipo di funzioni che devono implementare i nodi di ciascun strato) da assegnare alla rete neurale affinché modelli al meglio il nostro sistema.
- La rete neurale è come una scatola nera, quindi non fornisce spiegazioni sul funzionamento del sistema in esame.



# Deep Learning

- Con il termine DNN (**Deep Neural Network**) si denotano reti «profonde» composte da molti livelli (almeno 2 hidden) organizzati gerarchicamente.
- Le implicazioni di universal approximation theorem e la difficoltà di addestrare reti con molti livelli hanno portato per lungo tempo a focalizzarsi su reti con un solo livello hidden.
- L'esistenza di soluzioni non implica efficienza: esistono funzioni computabili con complessità polinomiale operando su  $k$  livelli, che richiedono una complessità esponenziale se si opera su  $k-1$  livelli (Hastad, 1986).
- L'organizzazione gerarchica consente di condividere e riusare informazioni (un po' come la programmazione strutturata). Lungo la gerarchia è possibile selezionare feature specifiche e scartare dettagli inutili (al fine di massimizzare l'invarianza).



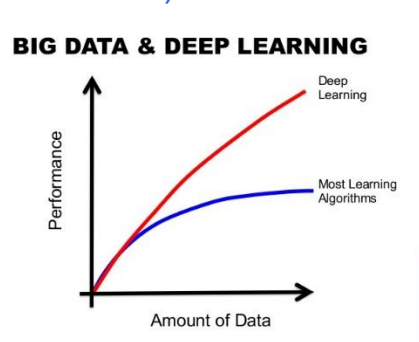
# Ma quanto Deep?

- Le DNN oggi maggiormente utilizzate consistono di un numero di livelli compreso tra 7 e 50.
- Reti più profonde (100 livelli e oltre) hanno dimostrato di poter garantire prestazioni leggermente migliori, a discapito però dell'efficienza.
- «solo» una decina di livelli tra la retina e i muscoli attuatori (altrimenti saremmo troppo lenti a reagire agli stimoli).

# Fattori abilitanti del Deep Learning

- Alcune reti deep ottengono già nel 1998 buone prestazioni in problemi di piccole dimensioni (es. riconoscimento caratteri, riconoscimento oggetti a bassa risoluzione), ma bisogna attendere il 2012 (AlexNet) per un radicale cambio di passo. AlexNet non introduce rilevanti innovazioni rispetto alle CNN di LeCun del 1998, ma alcune condizioni al contorno sono nel frattempo cambiate:

- ✓ **BigData:** disponibilità di dataset etichettati di grandi dimensioni (es. ImageNet: milioni di immagini, decine di migliaia di classi).



- ✓ **GPU computing:** il training di modelli complessi (profondi e con molti pesi e connessioni) richiede elevate potenze computazionali. La disponibilità di GPU con migliaia di core e GB di memoria interna ha consentito di ridurre drasticamente i tempi di training: da mesi a giorni

# Play with it!

- A questo [link](https://playground.tensorflow.org) trovate un playground interattivo per testare il task di classificazione di oggetti variando le caratteristiche della rete neurale
  - ✓ <https://playground.tensorflow.org>
  - ✓ <https://cs.stanford.edu/people/karpathy/convnetjs/>

