Intelligenza Artificiale II

F. Sciarrone

Febbraio 2007

Schema presentazione

- Obiettivi generali
- Reti neurali
- Memoria Associativa
- Percettrone Multistrato

Obiettivi generali

- Caratterizzazione storica
- Caratteristiche principali di una Rete Neurale
- Metodologia di selezione di una rete
- Problemi di Pattern Recognition: il Percettrone

Bibliografia

- *Manuale sulle reti neurali.* D. Floreano - ed. Il Mulino
- *Fondamenti di reti neurali.* G. Martinelli - ed. Siderea
- *L'Officina Neurale.* G. Carrella - ed. FrancoAngeli
- *Le Reti Neuronali.* S. Patarnello - ed. FrancoAngeli
- *Sistemi Complessi e Processi Cognitivi .* R. Serra, G. Zannarini - ed. Calderini
- *Reti Neuronali e Processi Cognitivi* .E. Pessa ed. Di Renzo editore
- *Neural Networks A Comprehensive Foundation.* S. Haykin. Prentice Hall
- *Parallel Distributing Processing.* Rumelhart-McClelland.
- *Neural Networks for pattern recognition.* C. M. Bishop. Oxford Press
- *Obiect Oriented Neural Networks in C++.* J. Rogers. Academic Press
- *Neural and Adaptive Systems.* Principe, Euliano, Lefebre. Wiley ed.

Un pò di storia - 1

- 3000 a.c. Ippocrate: primi tentativi di studio del cervello umano. Vengono individuate le posizioni di certe aree di controllo, sia motorie che sensorie, all'interno del cervello.
- 1812. Sintesi del primo composto organico. Si pensava che in pochi anni sarebbe stato svelato il codice segreto della vita per riprodurre esseri viventi.
- 1890: Williams James (psicologo). Tentativo fatto per comprendere il modo di funzionamento del cervello umano.
- 1920. Nascita della biologia molecolare. Minaccia di imminente clonazione (uomo invisibile nella fantascienza).
- 1936. A. Turing. Proposta di analogia tra cervello umano e computer

Un pò di storia - 2

- 1943: Warren Mc Culloch e Walter Pitts riprodussero una semplice rete neurale impiegando circuiti elettrici collegati tra loro in base a considerazioni sul funzionamento del singolo neurone e dimostrarono che le reti neurali sono analoghe ad una macchina di Turing, per cui qualsiasi operazione eseguita da una rete neurale poteva essere eseguita anche da un computer.

Un pò di storia - 3

- 1949. Wiener. Cybernetics. Visti in termini cibernetici, gli esseri umani e le macchine cominciano ad assomigliarsi sensibilmente. Processo di regolazione come trasferimento di informazioni. E' l'informazione che regola gli esseri viventi.
- 1949 Donald Hebb. The Organization of Behaviour. Correlazione tra psicologia e fisiologia. Esposizione della teoria del rafforzamento delle connessioni. Questa teoria è alla base della legge di apprendimento per le reti neurali (legge di Hebb).
- Decennio 50-60. Era della simulazione su computer. Simulazioni condotte dal gruppo di ricerca IBM (Rochester et al.) sulle funzionalità del cervello. Lavoro basato sulla legge di Hebb.
- 1951. Minsky and Edmond. Synthetic Brain SNARK. Test della legge di Hebb. 300 valvole e 40 resistori. Apprendimento per percorrere un labirinto. Test positivo nonostante errori di connessione.

Un pò di storia - 4

- 1956. Minsky. Nascita dell'intelligenza artificiale. Impulso al campo dell' intelligenza artificiale
- Si delineano due scuole di pensiero in contrapposizione tra loro:
  - Approccio High-Level: programmi intelligenti indipendenti dalla macchina (Minsky)
  - Approccio Low-level: la macchina stessa ha una sua importanza fondamentale e l'intelligenza dipende fortemente dalla macchina e dai suoi componenti elementari. => Connessionismo (Rosemblatt)
- 1957: Perceptron. Rosemblatt. Modello di sistema neurale assemblato in hardware. E' il più datato tra i sistemi neurali tuttavia ancora offi viene utilizzato in varie applicazioni. MARK I
- 1958: The Computer and the Brain. J. Von Neumann. Si introducono suggerimenti sull'imitazione di funzioni neurali semplici tramite l'utilizzo di ripetitori telegrafici e valvole.

Un pò di storia - 5

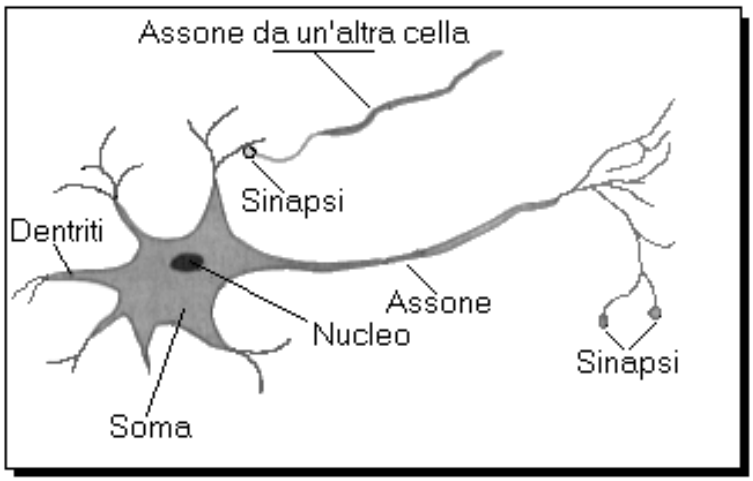
- 1959: Adaline e Madaline. Widrow-Hoff. (Adaptive Linear Neuron, Multiple ……). Primo esempio di rete neurale applicata ad un problema concreto: l'eliminazione di echi presenti su linee telefoniche. Sfruttata commercialmente per alcuni decenni.
- 1960. Avalanche. S. Grossberg. Trasmissione di comandi a braccia meccaniche.
- 1963: *Perceptrons:an Essay in Computational Geometry.* Minsky e Papert. Si critica il perceptron di Rosemblatt. Problema dello XOR. Problemi linearmente separabili. Si interrompono i finanziamenti al connessionismo.
- BSB:J. Anderson. Modello commerciale.
- 1982: J. Hopfield: saggio sulle reti neurali. Introdusse il concetto di energia di una rete neurale. Analogia con il modello di Ising.

Un pò di storia - 6

- 1987: prima conferenza sui sistemi neurali. 1800 persone. 19 società.
- 1987: costituzione della Società internazionale sui sistemi neurali. Grossberg e Kohonen.

Il sistema nervoso

- Il sistema nervoso centrale è costituito da circa 10^{11} *neuroni*



Reti Neurali - Motivazioni

- Architettura di *von Neumann* per i computer tradizionali.
- Istruzioni organizzate in modo gerarchico e ed eseguite sequenzialmente.
- Capacità di effettuare moltissime operazioni al secondo.
- Difficoltà ad eseguire determinati compiti

Reti Neurali - Motivazioni

- Esempio: riconoscimento di oggetti (soluzione difficile da descrivere mediante un insieme di azioni ben definite).

- Soluzione facile per il cervello umano per il modo di elaborare le informazioni.

Il sistema nervoso

- Ogni neurone è elemento di elaborazione.
- Invia un segnale in base a:
  - quelli ricevuti da tutti i neuroni a cui è connesso
  - alla propria *soglia di attivazione*.

⇩

elaborazione delle informazioni parallela e distribuita.

Il sistema nervoso

Altra caratteristica: capacità di apprendere tramite l'esperienza.

⇨sviluppo di **reti neurali artificiali**.

Reti neurali artificiali

Modelli neurali artificiali: semplificazione del sistema nervoso, con una approssimazione più o meno marcata a seconda del tipo di modello.

Una rete neurale artificiale è costituita da:

- ***neuroni*** (unità di elaborazione)
- ***sinapsi*** (collegano i neuroni e "pesano" i segnali che vi transitano).

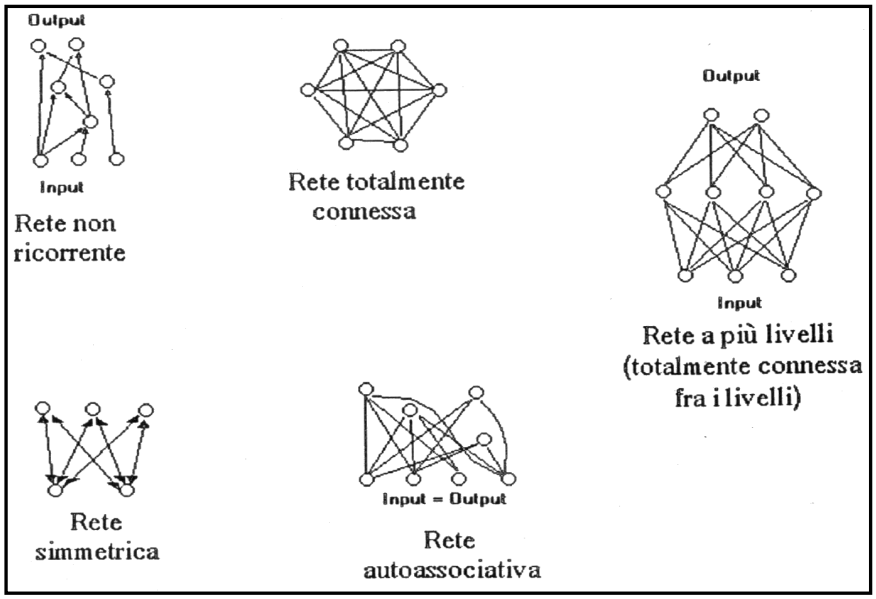
Che cos'è una rete neurale

- Una nuova forma di elaborazione basata sul modello biologico
- Un modello matematico composto da un gran numero di elementi organizzati in livelli
- Un sistema di elaborazione costituito da un elevato numero di semplici ed interconnessi elementi che elaborano le informazioni modificando la risposta dinamica a sollecitazioni esterne
- Sistemi dinamici non lineari con molti gradi di libertà che possono essere impiegati per risolvere problemi computazionali
- Scienza del giusto peso

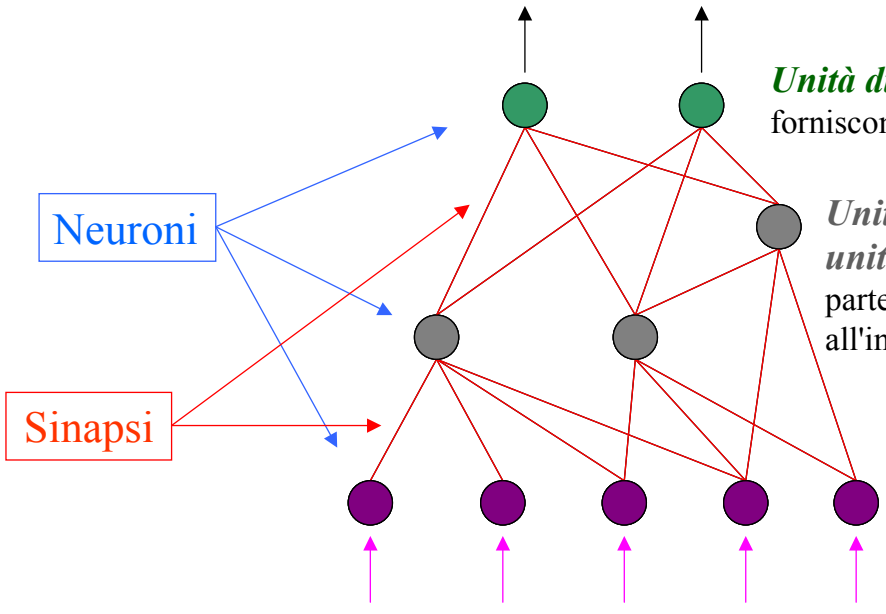
Hecht Nielsen

Un sistema dinamico avente la topologia di un grafo orientato; può elaborare informazioni producendo uno stato a fronte di input iniziale o continuo; i nodi sono gli elementi di elaborazione e le connessioni i canali di informazioni; ogni elemento produce un solo segnale di output che può viaggiare su più canali.

Modelli di reti neurali



Reti neurali artificiali - Architettura



Unità di uscita (***output units***): forniscono la risposta della rete

Unità nascoste (***hidden units***): sono opzionali e partecipano alle comunicazioni all'interno della rete

Unità di ingresso (***input units***): ricevono i segnali dall'ambiente esterno

Reti neurali artificiali-Architettura

Architettura di una rete neurale artificiale - caratterizzata dal:

- numero di neuroni di input
- numero di neuroni di uscita
- strati di sinapsi (o dagli strati di neuroni nascosti).

Sinapsi: sono la *memoria a lungo/breve termine* di una rete in quanto i pesi corrispondenti sono determinati mediante l'apprendimento.

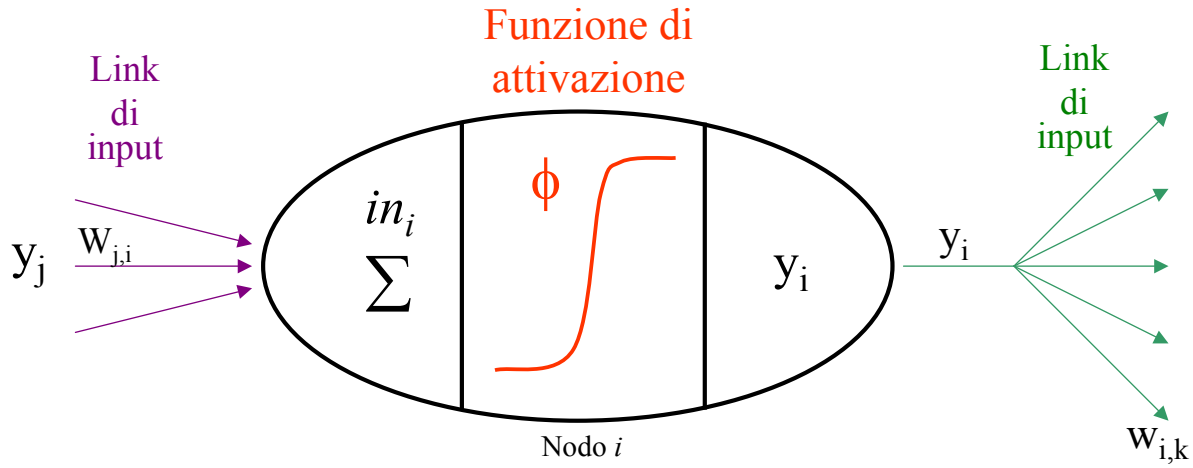
Neurone artificiale

Obiettivo: simulare il ruolo di un neurone biologico.

Si attiva quando la quantità totale del segnale ricevuto supera la propria *soglia di attivazione* ⇨ emette un segnale che raggiunge tutte le unità connesse.

Le sinapsi "pesano" il segnale moltiplicandolo per il valore loro associato.

Neurone artificiale



Ingresso al nodo *i*:

$$in_i = \sum_{j=1}^{n} w_{j,i} y_j$$

Neurone artificiale - Risposta

Input netto A_i del neurone n_i: $A_i = \sum_{j=1}^{n} w_{j,i} y_j - \vartheta_i$

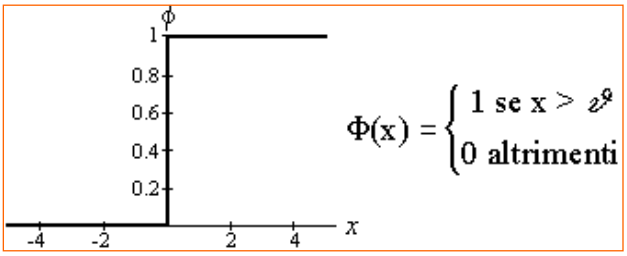
- y_j - segnale proveniente dal neurone n_j
- $w_{j,i}$ - peso della sinapsi n_j-n_i
- ϑ_i - soglia del nodo n_i

Risposta del neurone n_i:

$$y_i = \Phi(A_i) = \Phi\left(\sum_{j=1}^{n} w_{j,i} y_j - \vartheta_i\right)$$

La *funzione di attivazione* ϕ determina il tipo di risposta di un neurone.

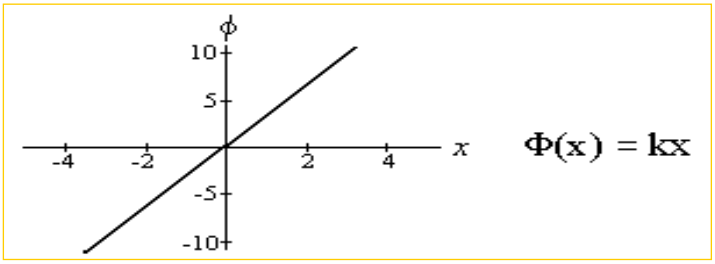
Alcune funzioni di attivazione



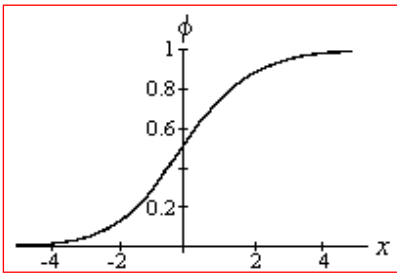
$$\Phi(x) = \begin{cases} 1 \text{ se } x > \vartheta \\ 0 \text{ altrimenti} \end{cases}$$

Funzione a gradino

Funzione lineare continua



$$\Phi(x) = kx$$



$$\Phi(x) = \frac{1}{1 + e^{-kx}} \quad k \text{ costante}$$

Funzione sigmoide

Funzioni di attivazione

- Altra funzione molto utilizzata: tangente iperbolica.

Attivazione dei nodi: binaria o limitata in un intervallo.

Funzioni di attivazione continue: trasmissione di segnali con intensità variabile (come neuroni biologici).

L'attivazione di un neurone dipende solo dall'informazione ricevuta ed è indipendente dalle operazioni svolte dagli altri nodi.

Elaborazione parallela: i segnali in input raggiungono i nodi di uscita passando per eventuali neuroni nascosti.

Classificazione delle reti neurali

- Supervised Models: modelli di rete neurale in cui l'apprendimento viene guidato dall'esterno attraverso un insieme di pattern di esempio forniti alla rete (training set). Rientrano in tale categoria:
  - MultiLayer Perceptron
  - Radial Basi Functions
  - Neurofuzzy Models
  - Ridge Polynomial Network
- Unsupervised Models: modelli di rete neurale in cui la rete apprende autonomamente dagli input forniti:
  - Self-Organizing Map
  - Generative Topographic Mapping

Alcuni Vantaggi e Svantaggi

Vantaggi	Svantaggi
High Accuracy:sono in grado di approssimare mapping complessi	**Poca trasparenza:** operano come *black box*
Indipendenza da assunzioni a priori: non fanno distinzioni a priori sulle distribuzioni dei dati e sulla forma di interazione tra componenti	**Progettazione Trial and Error:** la scelta dei nodi e dei parametri è di tipo euristico
Tolleranza al rumore: risultano molto flessibili rispetto a dati incompleti, rumorosi o mancanti	**Quantità di dati:** per calcolare correttamente le sinapsi occorre una grande quantità di dati. Molto Computer Intensive.
Facilità di manutenzione: possono essere aggiornate con nuovi dati rendendole adatti per ambienti dinamici	**Over-fitting:** bisogna fare molta attenzione alla fase di apprendimento per evitare che la rete generalizzi male
Superano alcune limitazioni di modelli statistici: capacità di generalizzazione	**Mancanza di regole** per selezionare l'algoritmo più appropriato per l'apprendimento
Possono essere implementate in HW parallelo	**Problema della Convergenza**: possono convergere a minimi locali nella superficie dell'errore

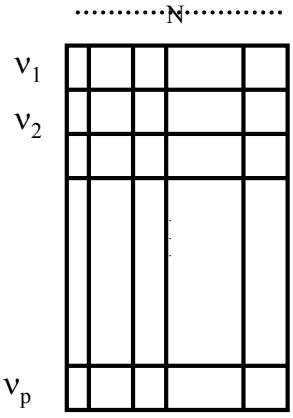
La Memoria Associativa

Memoria associativa

- Memorizzazione e recupero di informazioni attraverso l'associazione con altra informazione. Essa rappresenta la più semplice applicazione di "collective computation" di una rete neurale.
- Un dispositivo di memorizzazione si definisce memoria associativa se consente il recupero dell'informazione sulle basi di una conoscenza parziale del suo contenuto senza conoscerne la locazione di memoria. Si parla quindi memoria indirizzabile per contenuto.
- I computer tradizionali recuperano informazione attraverso una precisa conoscenza dell'indirizzo di memoria.

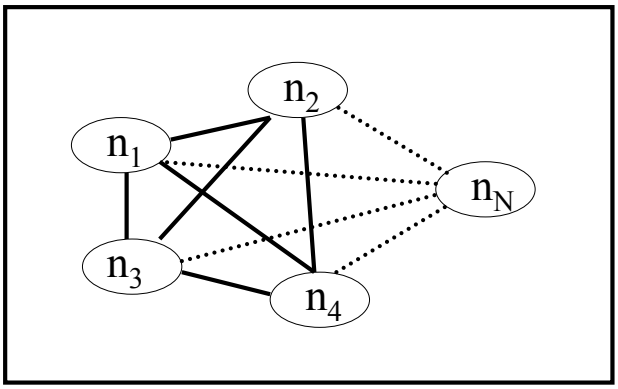
Memoria associativa: costruzione formale

Si supponga di avere p patterns binari contenenti N bit di informazione ciascuno:



$$\nu_i^{\mu}(i = 1,\ldots, N; \mu = 1,\ldots, p)$$

Memoria associativa: topologia



- **Ogni nodo è connesso a tutti gli altri**
- **L'insieme (n_1...n_N) rappresenta un pattern**

Memoria associativa: costruzione formale

- Dato un nuovo pattern n≡(n_1, n_2, ... n_N), la rete deve recuperare il pattern ν che più si avvicina a n => n e ν devono differire nel minor numero di posti possibile. Si utilizza la distanza di Hamming tra patterns:

$$\Downarrow$$

$$H_\mu = \sum_{i=1}^{N} (n_i - \nu_i^\mu)^2$$

Deve essere minima per μ=λ

Memoria associativa: costruzione formale

In via di principio, questo problema risulta facilmente risolvibile su un moderno computer calcolando tutti i valori H_μ e successivamente ricercandone il più piccolo.

Tuttavia

Il metodo diventa troppo pesante per grandi e numerosi pattern

Quindi

Si cerca di costruire una rete neurale di N elementi la quale, data la configurazione iniziale n_i corrispondente al pattern in input, n, evolva autonomamente verso la configurazione desiderata ν_i^λ

Formulazione del problema

Si sfrutta l'analogia tra neuroni e spin di Ising, sostituendo le quantità n_i e ν_i con le nuove variabili σ_i e s_i definite come:

$s_i = 2n_i - 1$, $\sigma_i = 2\nu_i - 1$

tali variabili prendono i valori ± 1 invece che 0 e 1. La distanza di Hamming allora, ricalcolata diviene:

$$(n_i - \nu_i^\mu)^2 = \tfrac{1}{4}(s_i - \sigma_i^\mu)^2 = ... = \frac{1}{2}(1 - s_i\sigma_i^\mu)$$

Il modello evolutivo di Hopfield

- La ricerca del minimo per H_μ si riduce alla ricerca del massimo della funzione:

$$A_\mu(s_i) = \sum_{i=1}^{N} \sigma_i^\mu s_i \qquad \mu = 1 \ldots p$$

Per un dato pattern s_i. Il modello di Hopfield propone il seguente schema evolutivo:

$$s_i(t+1) = \mathrm{sgn}\left(\sum_{j=1}^{N} w_{ij} s_j - \vartheta_i\right)$$

- $w_{ij} = w_{ji}$
- $w_{ii} = 0$
- $\theta_i = 0$

L'evoluzione temporale procede a passi discreti.

La regola di Hebb (apprendimento hebbiano)

Adesso, si tratta di scegliere i pesi w_{ij} in funzione dei pattern memorizzati σ_i^μ tale che la rete evolva autonomamente dal pattern presentato in input s_i al più vicino pattern memorizzato

- Ogni pattern memorizzato corrisponde ad una configurazione stabile
- La più piccola deviazione da esso sarà automaticamente corretta dalla dinamica della rete.

$$\Delta w_{ij} = w_{ij} + \eta \sigma_i \sigma_j$$

Addestramento di una memoria associativa

- Fase di Storage di pattern o attrattori
  - Modello evolutivo di Hopfield
  - Regola di apprendimento Hebbiano
  - Si memorizzano i pattern nella rete
- Fase di Recall dei pattern memorizzati

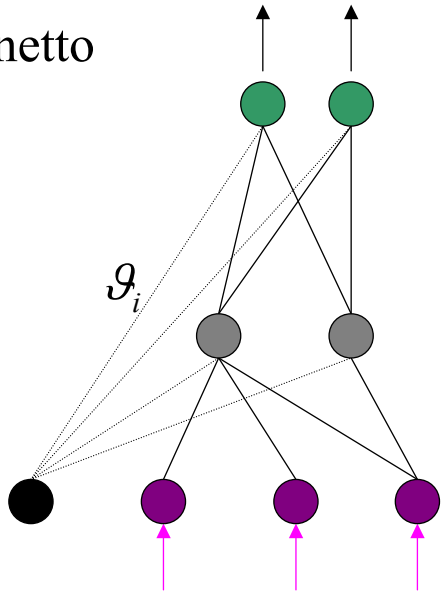
IL PERCETTRONE

Hornik-Stinchombe-White

Neurone artificiale - Bias

Si uniforma il calcolo dell'input netto (soglia come peso sinaptico):

- si inserisce l'*unità di bias*
- *si usa* il peso sinaptico $w_{0,i} = \vartheta_i$
- attivazione costante e pari a -1
  ⇨ $y_0 = -1$



$$y_i = \Phi\left(\sum_{j=0}^{n} w_{j,i} y_j\right)$$

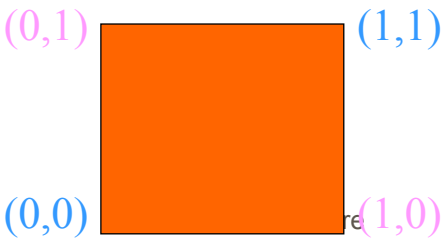
Il problema dello XOR

Percettrone elementare (senza strati nascosti): non può classificare pattern che non sono linearmente separabili.

Questi casi però sono frequenti: ad esempio problema dello XOR.

Caso particolare della classificazione di punti *nell'ipercubo unitario*: ogni punto è in classe 0 o in classe 1.

Per lo XOR si considerano gli angoli del quadrato unitario (i pattern (0,0), (0,1), (1,0) e (1,1))



Classe 0:
0 XOR 0 = 0
1 XOR 1 = 0

Classe 1:
1 XOR 0 = 1
0 XOR 1 = 1

Il problema dello XOR

Uso di un solo neurone (due ingressi e una uscita)

⇩

retta che divide lo spazio di input: i punti nello stesso lato forniscono lo stesso output (0 o 1).

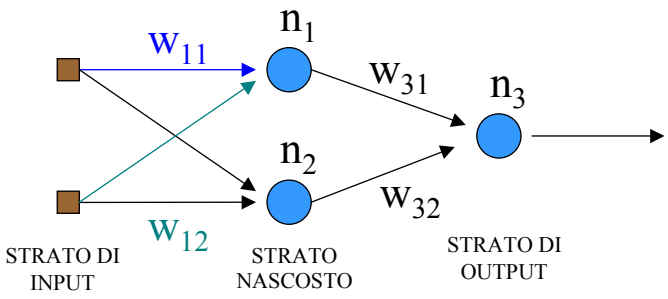
Posizione e orientamento della retta dipendono dai pesi sinaptci.

MA (0,0) e (1,1) [(1,0) e (0,1)] sono su angoli opposti del quadrato unitario ⇨non posso costruire una retta che separi i punti lasciando (0,0) e (1,1) da un lato, (1,0) e (0,1) dall'altro

⇨**un percettrone elementare non risolve il problema dello XOR**

Il problema dello XOR

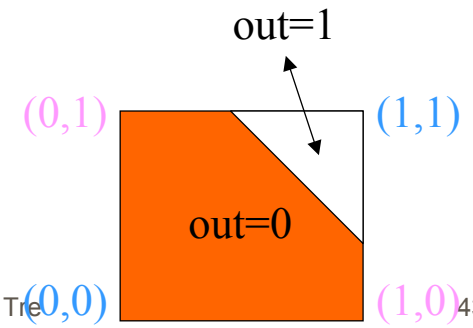
Si risolve introducendo uno strato nascosto con due neuroni (modello McCulloch-Pitts)



Pesi sinaptici n_1 : $w_{11} = w_{12} = 1$

Soglia $\theta_1 = \frac{3}{2}$

⇨

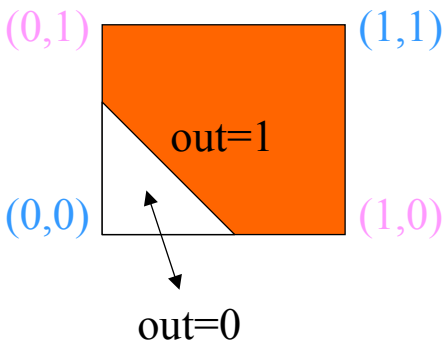


Il problema dello XOR

Pesi sinaptici n_2 : $w_{21} = w_{22} = 1$

Soglia $\theta_2 = \frac{1}{2}$

⇨

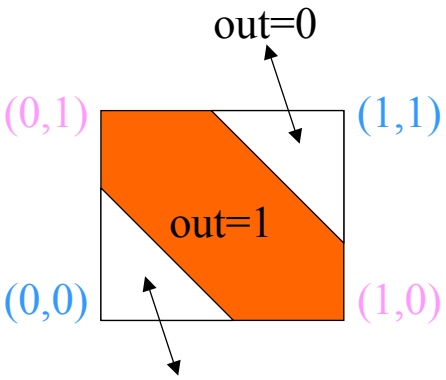


Il problema dello XOR

Pesi sinaptici n_3 : w_{31}= -2 w_{32}=1

Soglia $\theta_3 = \frac{1}{2}$

Regione di decisione della rete completa:



Superficie di decisione

Struttura	Tipo di regione	Problema dell'OR esclusivo	Classi con regioni incuneate	Forme più generali
Strato singolo	Semipiano limitato da un iperpiano			
Due strati	Regioni coinverse, aperte o chiuse			
Tre strati	Regioni la cui complessità è determinata dal numero di nodi			

Rappresentazione degli ingressi

Codifica delle informazione in input alla rete:

- *locale* - ogni unità di ingresso corrisponde ad un oggetto
  - ☹ necessità di un numero elevato di neuroni di ingresso
  - ☹ mancanza di flessibilità rispetto alla necessità di gestire nuovi oggetti
  - ☹ scarsa resistenza al rumore.
- *distribuita* - la rappresentazione di un oggetto utilizza più nodi di ingresso (ogni nodo di ingresso è relativo ad una caratteristica specifica).

Rappresentazione degli ingressi

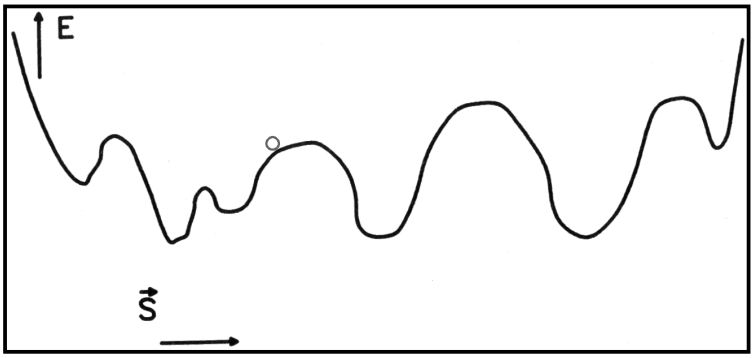
La rappresentazione dei pattern in ingresso ad una rete neurale è molto importante.

Può essere necessaria una normalizzazione dei dati che li costituiscono (ad esempio perché provenienti da "sorgenti" diverse).

È possibile normalizzare il vettore di ingresso x, dividendo ogni componente x_i per la norma del vettore stesso:

$$x_i = \frac{x_i}{\sqrt{\sum_{i=1}^{N} x_i^2}}$$

Superficie dell'errore



- Minimi Locali
- Minimi assoluti

Addestramento

Necessario per determinare preventivamente i valori sinaptici delle connessioni fra i nodi.

Effettuato tramite la modifica graduale dei valori sinaptici iniziali, sulla base di pattern di addestramento.

I valori iniziali delle connessioni sinaptiche sono assegnati in modo casuale all'interno di un intervallo o sono fissati tutti allo stesso valore.

Addestramento

Modalità per eseguire l'addestramento:

•*apprendimento supervisionato* - si utilizzano coppie di pattern

<vettore di ingresso,risposta desiderata>

L'aggiornamento dei pesi sinaptci è effettuato secondo una misura di errore tra la risposta della rete e quella desiderata (controlla la durata).

•*apprendimento per auto-organizzazione* - non si specifica la risposta desiderata corrispondente agli esempi di addestramento, ma si definiscono delle regole tramite cui la rete si auto-organizza.

Addestramento

Modifica dei pesi:

•*modalità on-line* – l'aggiornamento avviene per ogni pattern in ingresso

•*apprendimento per epoche* - effettuato dopo la presentazione di tutti i pattern di addestramento.

Le modifiche calcolate sono sommate ai valori già presenti

I pattern di addestramento sono presentati più volte in ingresso alla rete, per non "dimenticare" quanto già appreso.

Addestramento

Non esiste una regola esatta per determinare con esattezza il numero di esempi da utilizzare per il training, ma lo si può stimare; ad esempio per un MLP

$$numero\ record = 2 * numero\_totale\_connessioni$$

Per il numero di connessioni serve il numero di unità nascoste ⇨ si stima:

$$hu = \sqrt{numero\_ingressi * numero\_uscite}$$

Addestramento

Il valore minimo dell'errore, raggiunto il quale si può arrestare l'addestramento, dipende da più fattori quali la precisione che si vuole ottenere dalla rete, dal tipo di funzione di attivazione, dal numero di uscite della rete, ecc...

Terminata la fase di addestramento si può procedere con una *fase di test* con cui osservare il comportamento della rete tramite appositi vettori, senza modificare i pesi sinaptici.

Back propagation

È l'algoritmo più utilizzato per la modifica dei valori sinaptci.

Usato per reti con un numero qualsiasi di strati intermedi e architetture molto diverse.

Considera la distanza tra *y(t) e d(t)*

- *y(t)* - risposta della rete nell'istante *t*
- *d(t)* - risposta

Back propagation

Considera la somma dei quadrati delle distanze Euclidee:

$$E = \frac{1}{2}\sum_j (d_j(t) - y_j(t))^2$$

Obiettivo: minimizzare l'errore ⇨ i pesi sinaptci vengono modificati nella direzione opposta al gradiente della funzione di errore.

Back-propagation ⇨ gli aggiornamenti sono propagati all'indietro, da uno strato di unità verso quello inferiore.

Problemi: riconoscimento dei *minimi locali* della funzione di errore durante la ricerca del minimo assoluto dell'errore ⇨ varianti di questo algoritmo.

Back propagation

Possibilità: controllare la modifica delle connessioni mediante due parametri.

- *tasso di apprendimento* η - costante tradizionalmente usata per regolare la velocità di apprendimento (percentuale di modifica da applicare ai valori sinaptci).
- *momentum* α - "inerzia" nello spostamento sulla superficie dell'errore durante la ricerca del minimo.

Generalizzazione

Capacità di fornire le risposte appropriate a pattern di input che non sono mai stati incontrati.

Punto di forza delle reti neurali.

Attenzione in fase di costruzione per:

- numero di unità nascoste
- pesi sinaptici
- numero di training record

Problema dell'*overfitting*.

Overfitting

La rete apprende “troppo bene”" i pattern di addestramento.

Può fornire risposte sbagliate a causa di piccole distorsioni che possono agire sull'input.

Compromesso sull'errore in fase di addestramento, in modo che le prestazioni risultino accettabili anche per i nuovi pattern.

Overfitting

Metodo per determinare quando arrestare l'addestramento:

- un insieme di *pattern per l'addestramento* – usato per modificare i pesi sinaptici
- un insieme di pattern di *validazione* (*cross validation*) - per valutare l'errore raggiunto il quale si può arrestare l'addestramento
- un insieme di pattern di *test*

Overfitting

Vettori provenienti dalla stessa distribuzione ⇨

- record di addestramento - l'errore diminuisce con il passare delle epoche
- per gli altri due insiemi decresce inizialmente. L'inversione indica l'inizio dell'overfitting.