

Le reti neurali

Alberto Borghese

Università degli Studi di Milano
Laboratory of Applied Intelligent Systems (AIS-Lab)
Dipartimento di Scienze dell'Informazione
borghese@dsi.unimi.it



A.A. 2009-2010

1/46

<http://homes.dsi.unimi.it/~borghese>



Sommario



Dal neurone artificiale alle reti neurali

L'apprendimento in reti di perceptroni

Esempio con unità lineari ed accenno ad unità non-lineari

A.A. 2009-2010

2/46

<http://homes.dsi.unimi.it/~borghese>



Brains cause minds (J. Searle)



Le reti neurali

Se il neurone biologico consente l'intelligenza, perché non dovrebbe consentire l'intelligenza artificiale un neurone sintetico?

“.. a neural network is a system composed of *many simple processing elements* operating in *parallel* whose function is determined by *network structure, connection strengths*, and the *processing performed at computing elements* or nodes. ... Neural network architectures are inspired by the architecture of biological nervous systems, which use many simple processing elements operating in parallel to obtain high computation rates”. (DARPA, 1988)....



A cosa servono?



Le reti neurali offrono i seguenti specifici vantaggi nell'elaborazione dell'informazione:

- Apprendimento basato su esempi (non è richiesta l'elaborazione di un modello aderente alla realtà)
- Autoorganizzazione dell'informazione nella rete
- Robustezza ai guasti (codifica ridondante dell'informazione)
- Funzionamento in tempo reale (realizzazione HW)
- Basso consumo (0.5nW ÷ 4nW per neurone, 20W per il SN).



Cosa sono le reti neurali artificiali?

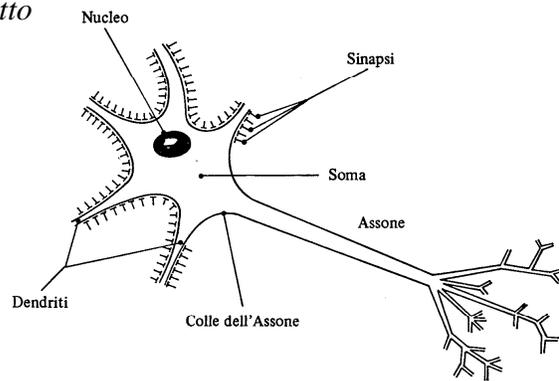


- Le reti neurali sono algoritmi non lineari per l'**approssimazione** di soluzioni di problemi dei quali non esiste un modello preciso (o se esiste è troppo oneroso computazionalmente), mediante l'utilizzo di esempi (dati e uscite) oppure per classificazioni. Connessioni con il dominio della statistica.
- Sono un capitolo importante negli argomenti di intelligenza artificiale.
- Da un altro punto di vista possono essere utilizzate per lo studio delle reti neurali naturali, ovvero dei processi cognitivi.



Il neurone artificiale

- *Potenziale di azione (tutto o nulla).*
- *Integrazione nel soma.*
- *Soglia di attivazione.*



Neurone come elemento di calcolo universale: in grado di calcolare qualsiasi funzione logica (cioè implementabile in un computer).

A.A. 2009-2010

7/46

<http://homes.dsi.unimi.it/~borghese>



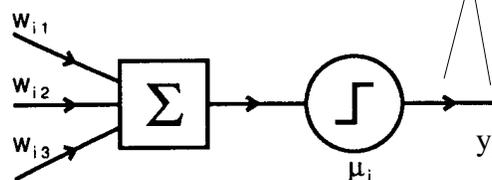
Il modello di McCulloch-Pitts

- Il tempo di propagazione lungo i dendriti non viene considerato.
- La variazione delle forma d'onda del potenziale di membrana lungo il dendrita non viene considerata.
- Gli input non sono sincroni.
- Le interazioni tra input non sono lineari.
- I pesi sono supposti costanti.

$$y_i(t+1) = \Theta(w_{ij}u_j(t) - \mu_i)$$

$$\Theta(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{altrimenti} \end{cases}$$

$$-1 < x < +1$$



A.A. 2009-2010

8

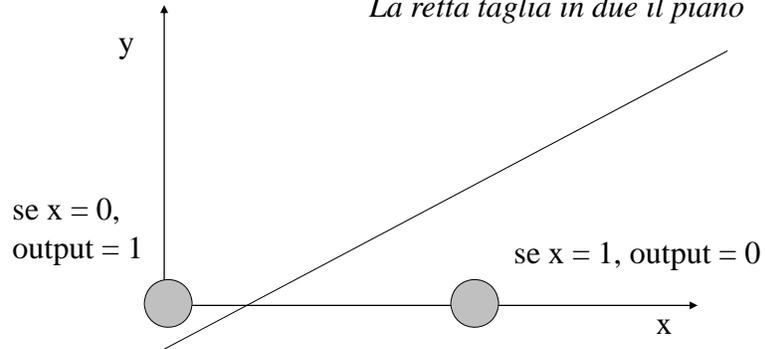
McCulloch-Pitts (1943)



Rappresentazione della retta



La retta taglia in due il piano



$$y = \mathbf{w}\mathbf{u} - \mu$$

$$y = mx + q \quad \text{singularità se la retta è // } y$$

$$x_2 = mx_1 + q \quad 1 = w_2 \quad m = -w_1$$

$$w_1x_1 + w_2x_2 - q = 0$$

$$w_0 + w_1x_1 + w_2x_2 = 0 \quad \Leftrightarrow \quad \mathbf{w} \cdot \mathbf{x} = 0 \quad \text{con } x_0 \equiv 1; w_0 = -q$$

A.A. 2009-2010

9/46

<http://homes.dsi.unimi.it/~borgnese>

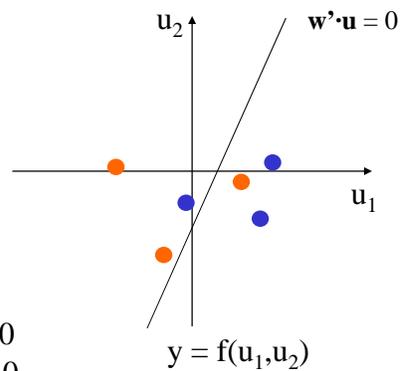
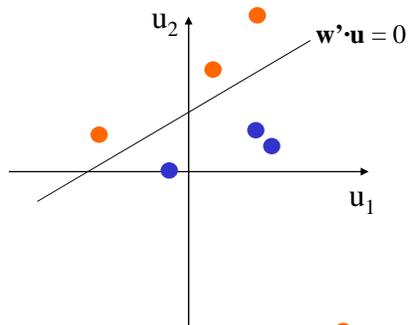


Funzioni linearmente separabili (classificatore!)



Linearmente separabile

Non linearmente separabile



- $y > 0$
- $y < 0$

A.A. 2009-2010

10/46

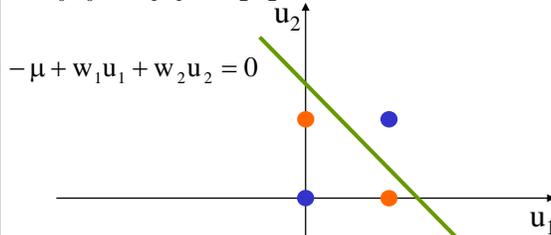
<http://homes.dsi.unimi.it/~borgnese>



La "morte" del neurone di McCulloch-Pitts (Minsky, 1969): XOR



$$w_0u_0 + w_1u_1 + w_2u_2 = 0 \quad \mathbf{w}' \cdot \mathbf{u} = 0 \quad u_0 = 1$$



$$-\mu + w_1u_1 + w_2u_2 = 0$$

u_1	u_2	y
0	0	-1
0	1	1
1	0	1
1	1	-1

a

b

c

d

● $y(u_1, u_2, 1) = 1$

● $y(u_1, u_2, 1) = -1$

d: $w_1 + w_2 < \mu$

b: $w_2 > \mu$

c: $w_1 > \mu$

a: $\mu > 0$

Il sistema di 4 equazioni non è risolvibile.

$w_1, w_2 > \mu$ e $w_1 + w_2 < \mu$ Impossibile!!

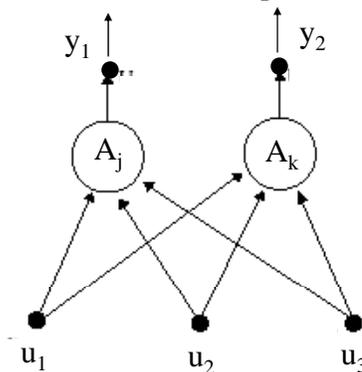
A.A. 2009 Si possono imparare solamente funzioni linearmente separabili ghese



La rete neurale ad un livello



La rete opera una trasformazione dallo spazio di input allo spazio di output.



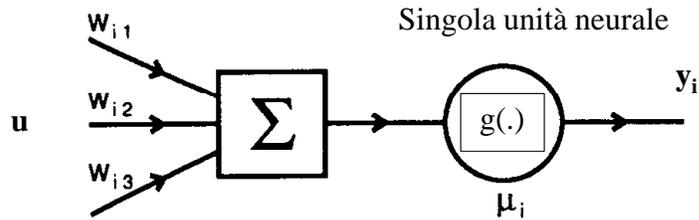
$$y_i = g(w_{ij}u_j - \mu_i)$$

La trasformazione o mappatura dipende dai parametri $\{w_{ij}\}$ e $\{\mu_i\}$ in modo tale che la rete neurale approssimi la trasformazione tra i pattern di input e di output.

Se $g(\cdot) = 1$, la rete diventa un modello lineare: $y_i = w_{ij}u_j - \mu_i$

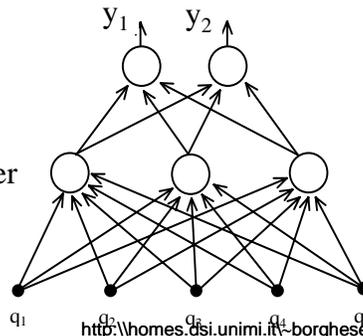


Una rete neurale a più livelli



$$y_i = g(w_{ij}u_j - \mu_i)$$

Unità nascoste – Hidden layer



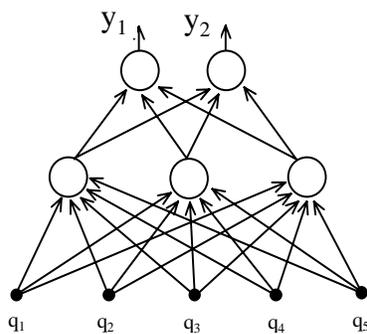
A.A. 2009-2010

13/46

<http://homes.dsi.unimi.it/~borgese>



Spiking neurons



Spiking neurons. Sono neuroni la cui uscita è il singolo spike. Modellazione realistica (e.g. McCullochPitts).

Connessionismo classico. Uscita compresa tra min – Max. Frequenza di scarica.

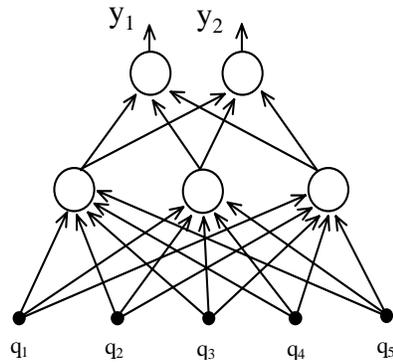
A.A. 2009-2010

14/46

<http://homes.dsi.unimi.it/~borgese>



Caratteristiche



Livelli di unità di attivazione

Collegamento in cascata

Input convergenti, output divergenti.

Capacità di approssimazione universale

Perceptrone: layered networks, flusso unidirezionale dell'elaborazione.

L'output viene interpretato come frequenza di scarica del neurone d'uscita della rete.

A.A. 2009-2010

15/46

<http://homes.dsi.unimi.it/~borgese>



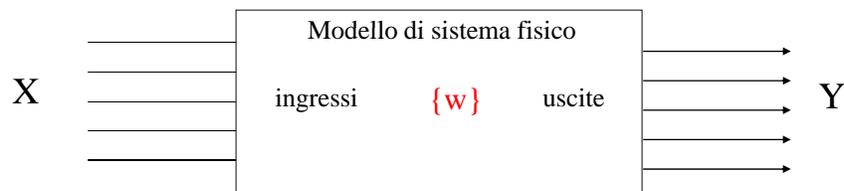
Complessità della funzione realizzabile

Quanti più neuroni artificiali vengono connessi tanto più la funzione complessiva approssimabile diviene più complessa

$$Y = |y_1, y_2, y_3, \dots, y_n|^T$$

$$y_i = g(X)$$

$$X = |x_1, x_2, x_3, \dots, x_m|^T$$

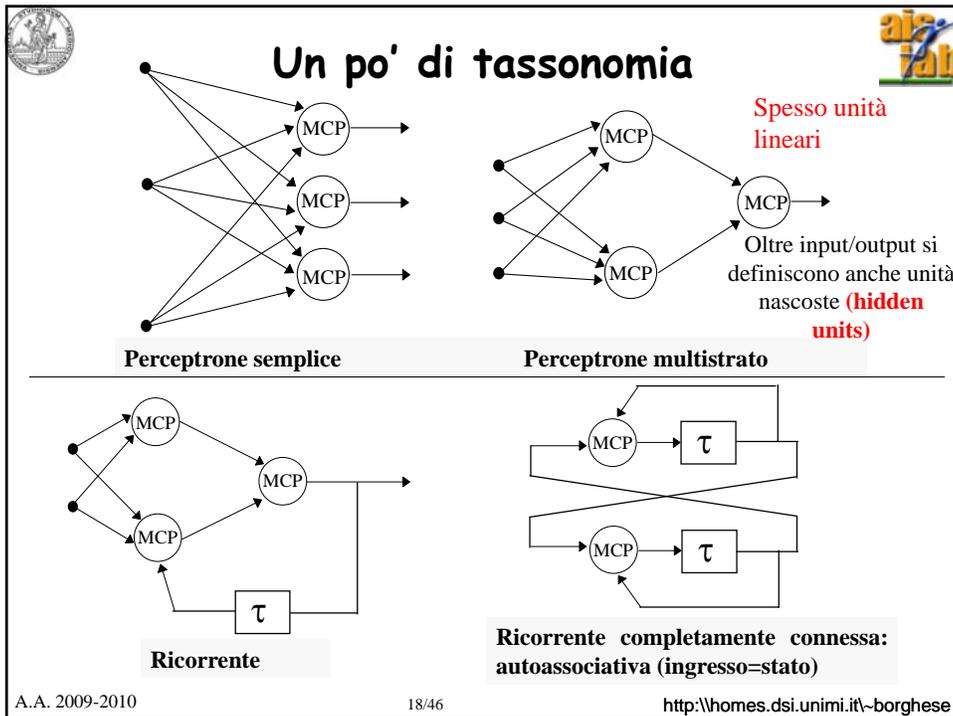
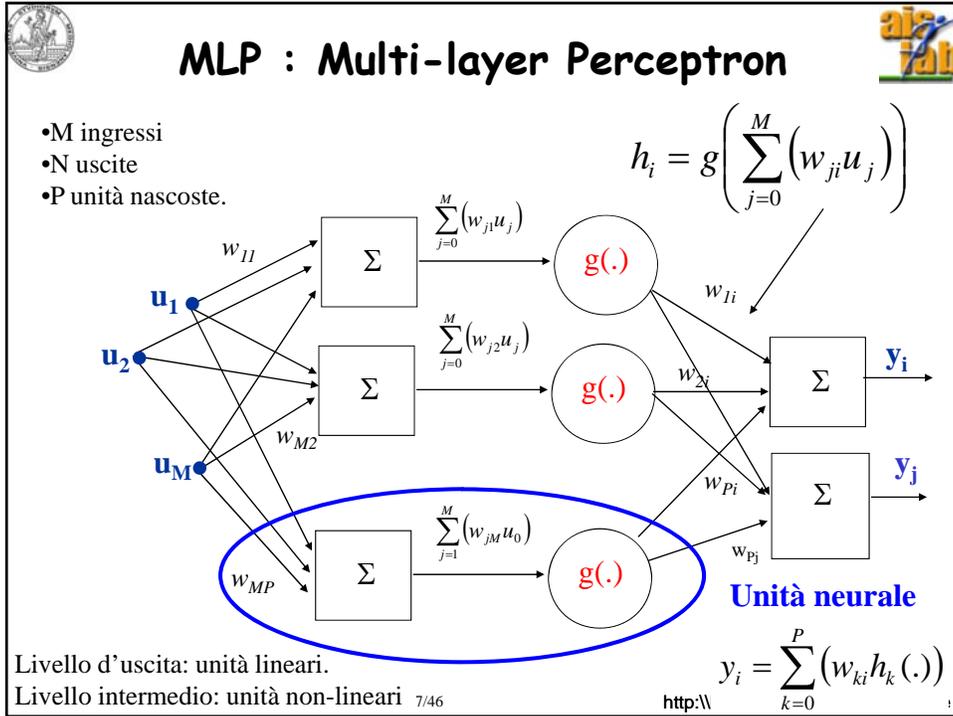


Reti neurali = approssimatori universali.

A.A. 2009-2010

16/46

<http://homes.dsi.unimi.it/~borgese>





Costituenti delle reti neurali

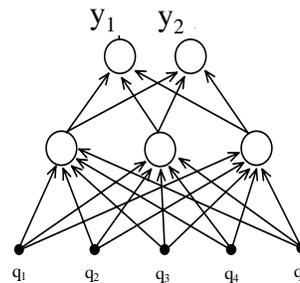


Un neurone artificiale è costituito da:

- Un insieme di input (provenienti da altri neuroni)
- Un peso che rappresenta l'efficacia ed il segno della sinapsi.
- Una funzione di somma (pesata) degli input.
- Una funzione di attivazione che trasforma gli input nell'output del neurone.

Una rete neurale è costituita da:

- Un insieme di neuroni artificiali.
- La connettività tra neuroni.



Sommario



Dal neurone artificiale alle reti neurali

L'apprendimento in reti di perceptroni

Esempio con unità lineari ed accenno ad unità non-lineari



I vari tipi di apprendimento



Supervisionato (learning with a teacher). Viene specificato per ogni coppia di pattern di input/output, il pattern desiderato di output.

Non-supervisionato (learning without a teacher). I neuroni identificano pattern di ingresso simili. Clustering. Mappe neurali.

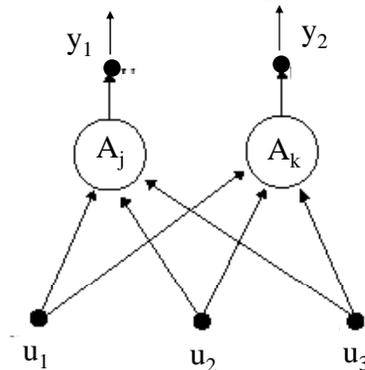
Apprendimento con rinforzo (reinforcement learning, learning with a distal teacher). L'ambiente fornisce un'informazione del tipo success or fail.



Lo spirito dell'apprendimento supervisionato



La rete opera una trasformazione dallo spazio di input allo spazio di output.



Apprendimento è la modifica dei parametri $\{w_{ij}\}$ e $\{\mu_j\}$ in modo tale che la rete neurale approssimi la trasformazione tra i pattern di input e di output.

$$y_i = g(w_{ij}u_j - \mu_i)$$



Funzione costo per unità di attivazione continue



Possiamo derivare una regola di apprendimento di spirito **Hebbiano** per una qualsiasi funzione di attivazione continua. Consideriamo un perceptrone ad un livello.

$$y = g\left(\sum_{j=1} w_{ij} u_j - \mu_i\right) = g\left(\sum_{j=0} (w_{ij} u_j)\right)$$

Si tratta di un problema di minimizzazione di una cifra di merito, J , sullo spazio di parametri W :

$$E(w) = \left\| \underbrace{y^D - g(W^{nuovo}U)}_{\text{Errore}} \right\| \leq \left\| y^D - g(W^{vecchio}U) \right\|$$

Errore

Devo trovare $\{w\}$: $E(w)$ è minimo.

$$E(w) = \frac{1}{2} \sum_p \left[\sum_j (y_{jp}^D - y_{jp})^2 \right] = \frac{1}{2} \sum_p \left[\sum_j \left(y_{jp}^D - g\left(\sum_i w_{ij} u_{ip}\right) \right)^2 \right]$$



Apprendimento supervisionato



$$\min_{\{w\}} J(\cdot) \quad J = \left\| Y^D - g(W^{nuovo}U) \right\| \leq \left\| Y^D - g(W^{vecchio}U) \right\|$$

Y^D è l'uscita desiderata nota.

- Si tratta di un problema di minimizzazione di una cifra di merito (J) sullo spazio di parametri W .

Soluzione iterativa (gradiente):

Obiettivo: se esiste una soluzione, trovare ΔW in modo iterativo tale che l'insieme dei pesi W^{nuovo} ottenuto come:

$$W^{nuovo} = W^{vecchio} + \Delta W$$

dia luogo a un errore sulle uscite di norma minore che con $W^{vecchio}$



Minimizzazione di funzioni di più variabili



$\min(J\{\mathbf{w}\} | \dots)$ funzione costo od errore

Gradiente:
$$\nabla \mathbf{J}(\mathbf{w}) = \frac{\partial J(\{\mathbf{w}\} | \dots)}{\partial w_1} \frac{\mathbf{w}_1}{|\mathbf{w}_1|} + \frac{\partial J(\{\mathbf{w}\} | \dots)}{\partial w_2} \frac{\mathbf{w}_2}{|\mathbf{w}_2|} + \frac{\partial J(\{\mathbf{w}\} | \dots)}{\partial w_3} \frac{\mathbf{w}_3}{|\mathbf{w}_3|} + \frac{\partial J(\{\mathbf{w}\} | \dots)}{\partial w_4} \frac{\mathbf{w}_4}{|\mathbf{w}_4|} + \dots$$

Modifico il valore dei pesi di una quantità proporzionale alla pendenza della funzione costo rispetto a quel parametro.

Estensione della tecnica del gradiente a più variabili.

$$\Delta \mathbf{w} = -\eta \nabla \mathbf{J}(\mathbf{w}) \Leftrightarrow \Delta w_{ij} = -\eta \frac{\partial J(\{\mathbf{w}\} | \dots)}{\partial w_{ij}}$$

Serve un' **approssimazione iniziale** per i pesi $\mathbf{W}_{ini} = \{w_j\}_{ini}$.



La pratica dell'apprendimento supervisionato



Fino a quando l'apprendimento non è stato completato:

1. Presentazione di un pattern di input / output.
2. Calcolo dell'output della rete con il pattern corrente.
3. Calcolo dell'errore
4. Calcolo dei gradienti
5. Calcolo dell'incremento dei pesi.

Aggiornamento dei pesi:

- Per trial (ogni pattern)
- Per epoca (ogni insieme di pattern).



Apprendimento supervisionato tramite gradiente



Coppie input/output note.

Definizione di una funzione costo che misuri l'errore sull'uscita.

Modifica dei valori dei pesi in modo tale che la funzione costo sia minimizzata.

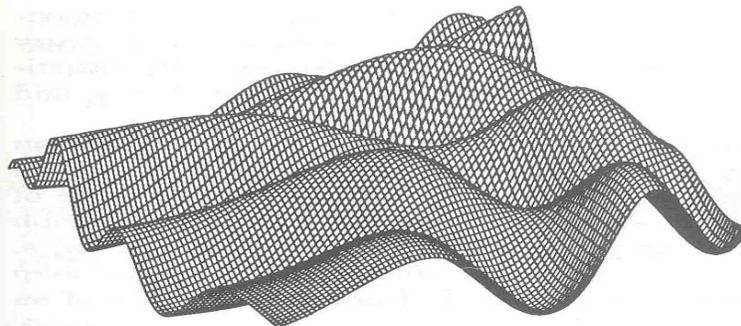
Reti multi-strato hanno elevata capacità computazionale, ma anche elevata complessità.



Problemi nell'apprendimento supervisionato tramite gradiente



- Nota: W_{ini} è generalmente casuale e può condizionare la convergenza degli algoritmi iterativi.
- I problemi di convergenza sono legati all'esistenza di minimi locali del funzionale $J(w | \dots)$





Problemi



Quando si termina l'algoritmo di apprendimento?

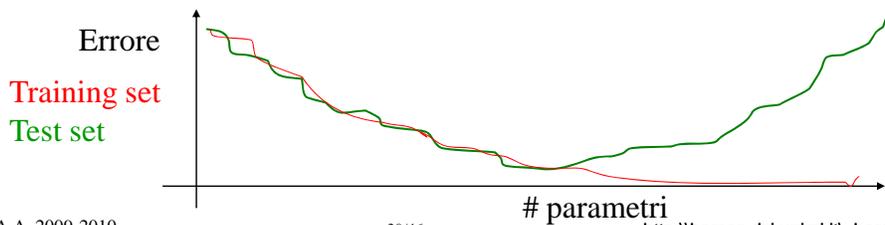
Bootstrap – Vengono estratti pattern con ripetizioni.

Cross-Validation - Errore sull'insieme di training =

Errore sull'insieme di test.

Utilizzare lo “structural risk” invece dell’”empirical risk”.

Si vuole evitare che la rete si specializzi troppo sui pattern di training e non sia in grado di interpolare.



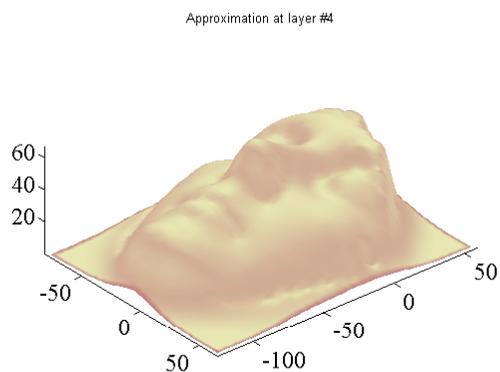
A.A. 2009-2010

29/46

<http://homes.dsi.unimi.it/~borghese>



Problema dell'overfitting dovuto a sovrapparametrizzazione



A.A. 2009-2010

30/46

<http://homes.dsi.unimi.it/~borghese>



Sommario



Dal neurone artificiale alle reti neurali

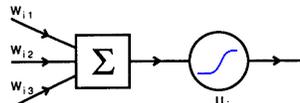
L'apprendimento in reti di perceptroni

Esempio con unità lineari ed accenno ad unità non-lineari



Unità di attivazione lineari



$$y_j = g\left(\sum_{i=1}^M w_{ij}u_i - \mu_j\right) = g\left(\sum_{i=0}^M (w_{ij}u_i)\right)$$


Caso lineare ($g(\cdot) = I$):

$$y_j = \sum_{i=1} w_{ij}u_i - \mu_j = \sum_{i=0} (w_{ij}u_i) \quad \implies \quad \mathbf{Y} = \mathbf{W} \mathbf{U}$$

Soluzione di un sistema lineare nei pesi!!

Condizione di risolubilità: \mathbf{W} di rango massimo \rightarrow
 $\{w\}$ sono linearmente indipendenti.



Unità lineari, soluzione iterativa



$$J = E(\mathbf{w}) = \frac{1}{2} \sum_p \left[\sum_j (y_{jp}^D - y_{jp})^2 \right] = \frac{1}{2} \sum_p \left[\sum_j \left(y_{jp}^D - \left(\sum_i w_{ij} u_{ip} \right) \right)^2 \right]$$

$$\Delta w_{ij} = -\eta \frac{\partial}{\partial w_{ij}} \frac{1}{2} \sum_j \left(y_j^D - \left(\sum_i w_{ij} u_i \right) \right)^2$$

$$\Delta w_{ij} = +\eta \sum_j \left(y_j^D - \left(\sum_i w_{ij} u_i \right) \right) u_i = +\eta \sum_j (y_j^D - y_j) u_i$$

Hebbian learning

δ rule (Hoff, 1960)

A.A. 2009-2010

33/46

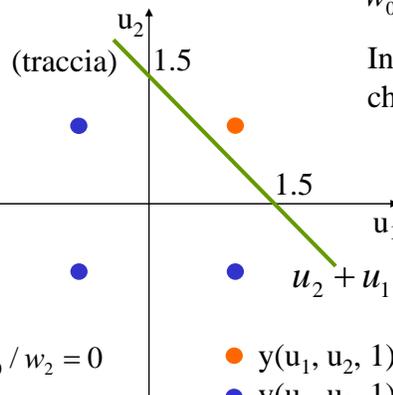
<http://homes.dsi.unimi.it/~borghese>



Esempio - AND (grafica)



Troviamo la soluzione graficamente



$$w_0 u_0 + w_1 u_1 + w_2 u_2 = 0$$

In verde la retta $\mathbf{w} \cdot \mathbf{u} = 0$ che taglia il piano $u_1 u_2$.

$$u_2 + (w_1 / w_2) u_1 + w_0 / w_2 = 0$$

$$u_2 + u_1 - 1.5 = 0$$

↓

$$w_1 / w_2 = 1 \quad w_0 / w_2 = -1.5 \quad \Rightarrow \quad w_2 = k \quad w_1 = k \quad w_0 = -1.5 * k$$

$$w_0 = -1.5$$

$$q = +1.5$$

$$w_1 = 1$$

$$w_2 = 1$$

$$\bullet \quad u_2 + u_1 - 1.5 = 0$$

$$\bullet \quad y(u_1, u_2, 1) = 1$$

$$\bullet \quad y(u_1, u_2, 1) = -1$$

Esistono più soluzioni
Separabilità lineare.

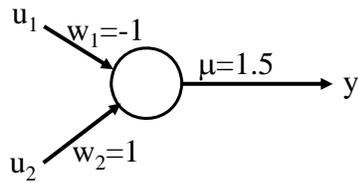
A.A. 2009-2010

34/46

<http://homes.dsi.unimi.it/~borghese>



Esempio di delta rule - I



$$U = [-1, 1] \quad y^D = -1$$

u_1	u_2	y	y^D
-1	-1	-1	-1
-1	1	+1	-1
1	-1	-1	-1
1	1	-1	+1

$$y = \sum_{i=1} w_i u_i - \mu = \sum_{i=0} (w_i u_i) = (-1)(-1) + (1)(1) - 1.5 = 0.5 \gg -1$$

$$u_0 = 1 \quad w_0 = -\mu$$

A.A. 2009-2010

35/46

<http://homes.dsi.unimi.it/~borgnese>

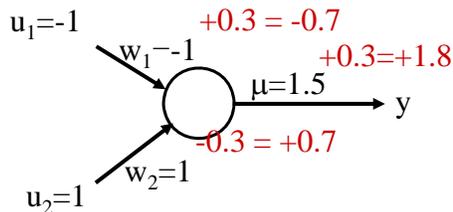


Esempio di delta rule - II



$$U = [-1, 1] \quad y^D = -1$$

$$\eta = 0.2$$



$$y = \sum_{i=1} (w_i u_i - \mu) = \sum_{i=0} (w_i u_i) = -0.4 > -1$$

$$\Delta w_{ij} = +\eta (y_j^D - y_j) u_i$$

$$\Delta \mu = \Delta w_0 = \eta (y_0^D - y_0) u_0 = \eta (-1 - 0.5)(1) = +0.30$$

$$\Delta w_1 = \eta (y_1^D - y_1) u_1 = \eta (-1 - 0.5)(-1) = +0.30$$

$$\Delta w_2 = \eta (y_2^D - y_2) u_2 = \eta (-1 - 0.5)(1) = -0.30$$

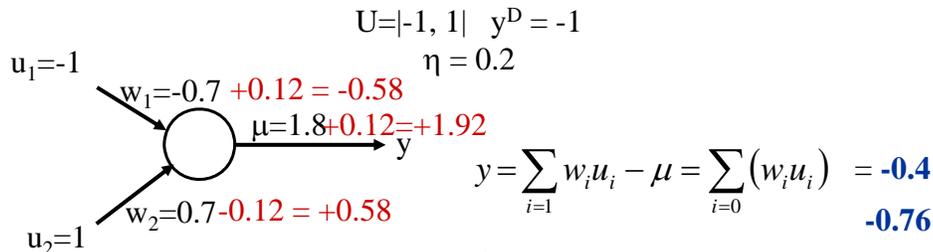
A.A. 2009-2010

36/46

<http://homes.dsi.unimi.it/~borgnese>



Esempio di delta rule - III



$$\Delta w_{ij} = +\eta (y_i^D - y_i) u_j$$

$$\Delta \mu = \Delta w_0 = \eta (y_i^D - y_i) u_0 = \eta (-1 - (-0.4)) (1) = +0.12$$

$$\Delta w_1 = \eta (y_i^D - y_i) u_1 = \eta (-1 - (-0.4)) (-1) = +0.12$$

$$\Delta w_2 = \eta (y_i^D - y_i) u_2 = \eta (-1 - (-0.4)) (1) = -0.12$$

Che relazione c'è tra i pesi e la retta che separa le uscite positive da quelle negative?

A.A. 20

ase



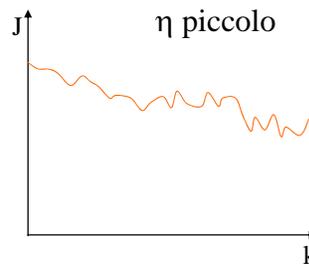
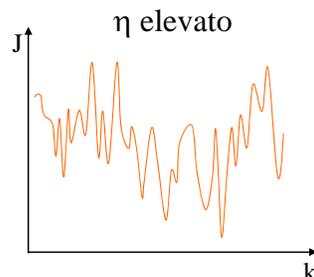
Ruolo di η - learning rate



$$\Delta w_{ij} = +\eta (y_j^D - y_j) u_i$$

Calmiera il Δw_{ij} per evitare che :

- Un peso sia specifico di un'unità ingresso-uscita.
- Oscillazioni durante l'apprendimento senza convergenza.

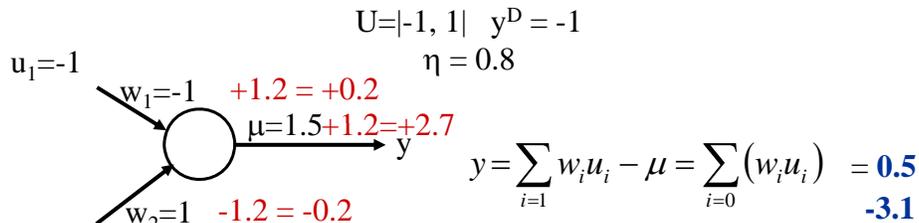


A.A. 2009-2010

η può variare durante l'addestramento. www.homes.dsi.unimi.it/~borgnese



Esempio di delta rule - Cattiva scelta di η



$$\Delta w_{ij} = +\eta (y_j^D - y_j) u_i$$

$$\Delta \mu = \Delta w_0 = \eta (y_i^D - y_i) u_0 = \eta (-1 - 0.5)(1) = +1.2$$

$$\Delta w_1 = \eta (y_i^D - y_i) u_1 = \eta (-1 - 0.5)(-1) = +1.2$$

$$\Delta w_2 = \eta (y_i^D - y_i) u_2 = \eta (-1 - 0.5)(1) = -1.2$$

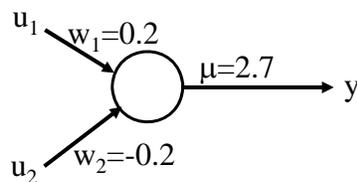
A.A. 2009-2010

39/46

<http://homes.dsi.unimi.it/~borghese>



Esempio di specializzazione sui pattern a, b, c



u_1	u_2	y^D	
-1	-1	-1	a
-1	1	-1	b
1	-1	-1	c
1	1	1	d

a $y = \sum_{i=1} w_i u_i - \mu = \sum_{i=0} (w_i u_i) = (0.2)(-1) + (-0.2)(1) - 2.7 = -3.1$

b $y = \sum_{i=1} w_i u_i - \mu = \sum_{i=0} (w_i u_i) = (0.2)(-1) + (-0.2)(1) - 2.7 = -2.9$

c $y = \sum_{i=1} w_i u_i - \mu = \sum_{i=0} (w_i u_i) = (0.2)(1) + (-0.2)(-1) - 2.7 = -2.3$

d $y = \sum_{i=1} w_i u_i - \mu = \sum_{i=0} (w_i u_i) = (0.2)(1) + (-0.2)(1) - 2.7 = -2.7$

Errato su d. Specializzazione su a, b, c

A.A. 2009-2010

40/46

<http://homes.dsi.unimi.it/~borghese>



Unità non-lineari, soluzione iterativa



$$J = E(\mathbf{w}) = \frac{1}{2} \sum_p \left[\sum_j (y_{jp}^D - y_{jp})^2 \right] = \frac{1}{2} \sum_p \left[\sum_j \left(y_{jp}^D - g \left(\sum_i w_{ij} u_{ip} \right) \right)^2 \right]$$

$$\Delta w_{ijp} = -\eta \frac{\partial}{\partial w_{ij}} \frac{1}{2} \sum_j \left(y_{jp}^D - g \left(\sum_i w_{ij} u_{ip} \right) \right)^2 =$$

$$\eta \sum_j \left(y_{jp}^D - g \left(\sum_i w_{ij} u_{ip} \right) \right) g' \left(\sum_i w_{ij} u_{ip} \right) u_i = +\eta \underbrace{\left(y_{jp}^D - y_{jp} \right) u_{ip} g' \left(\sum_i w_{ij} u_{ip} \right)}_{\delta \text{ rule}}$$

δ rule

A.A. 2009-2010

41/46

<http://homes.dsi.unimi.it/~borgese>

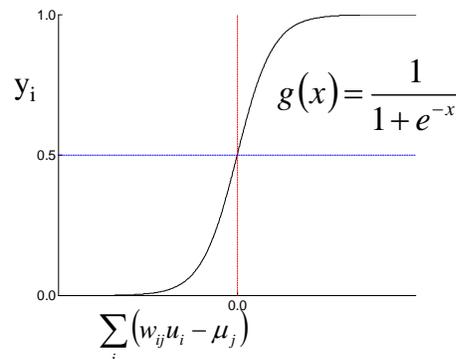


Perceptrone con unità di attivazione logistiche



$$g'(x) = g(x) \cdot (1 - g(x)) \quad y_j = g \left(\sum_i w_{ij} u_i - \mu_j \right)$$

$$\begin{aligned} g'(x) &= \frac{e^{-x}}{(1 + e^{-x})^2} = \\ &= \frac{1}{1 + e^{-x}} \left(1 - \frac{1}{1 + e^{-x}} \right) = \\ &= g(x)(1 - g(x)) \end{aligned}$$



A.A. 2009-2010

42/46

<http://homes.dsi.unimi.it/~borgese>



Update dei pesi per funzione logistica



$$J = E(\mathbf{w}) = \frac{1}{2} \sum_p \left[\sum_j (y_{jp}^D - y_{jp})^2 = \frac{1}{2} \sum_j \left(y_{jp}^D - g\left(\sum_i w_{ij} u_{ip}\right) \right)^2 \right]$$

$$\Delta w_{ijp} = +\eta \sum_j (y_{jp}^D - g(\cdot)) g'(\cdot) u_i = +\eta (y_{jp}^D - y_j) \underbrace{y_j (1 - y_j)}_{\delta \text{ rule}} u_{ip}$$

 δ rule

NB $y_i \in [0, 1]$. Per $y_i = 0$ o $y_i = 1$ non c'è apprendimento anche se l'uscita è sbagliata. Quando si verifica questa situazione?

Si cerca di mantenere le unità lontane della saturazione.



Riassunto - topologia



I neuroni connessionisti sono basati su:

- Ricevere una somma pesata degli ingressi.
- Trasformarla secondo una funzione non-lineare (scalino o logistica)
- Inviare il risultato di questa funzione all'uscita o ad altre unità.

Le reti neurali sono topologie ottenute connettendo tra loro i neuroni in modo opportuno e riescono a calcolare funzioni molto complesse.



Riassunto - Apprendimento



Algoritmi iterativi per adattare il valore dei parametri (pesi).

Definizione di una funzione costo che misura la differenza tra valore fornito e quello desiderato.

Algoritmo (gradiente) che consente di aggiornare i pesi in modo da minimizzare la funzione costo.

Training per pattern (specializzazione) o per epoche.



Sommario



Dal neurone artificiale alle reti neurali

L'apprendimento in reti di perceptroni

Esempio con unità lineari ed accenno ad unità non-lineari