

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. 2012-2013

1/53

<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. 2012-2013

2/53

<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)....

Now, this is called learning with Kernels



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).



A.A. 2012-2013

5/53

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



Cosa sono le reti neurali artificiali?



- Le reti neurali sono modelli non lineari per l'**approssimazione** della soluzione di problemi dei quali non esiste un modello preciso (o se esiste è troppo oneroso computazionalmente). I parametri dei modelli risultanti (semiparametrici) vengono calcolati mediante l'utilizzo di esempi (dati di ingresso e uscita desiderata). Connessioni con il dominio della statistica.
- Vengono utilizzate soprattutto per la classificazione e la regressione.
- 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.
- Sono state incorporate nel "machine learning".

A.A. 2012-2013

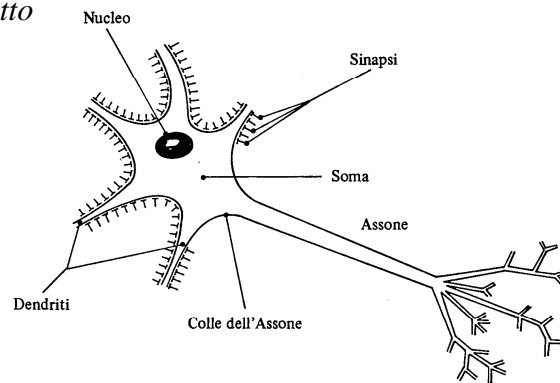
6/53

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



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. 2012-2013

7/53

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



Il modello di McCulloch-Pitts

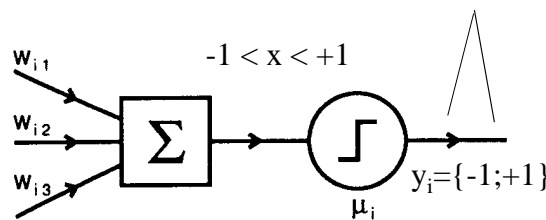
• La variazione delle forma d'onda del potenziale di membrana lungo il dendrita non viene considerata.

• Gli input non sono sincroni.

• Le interazioni tra input non sono lineari. $y_i(t+1) = \Theta(w_{ij}u_j(t) - \mu_i)$

• I pesi sono supposti costanti. $\Theta(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ -1 & \text{altrimenti} \end{cases}$



Sono state pensate per calcolare **funzioni logiche (V o F)**.



A.A. 2012-2013

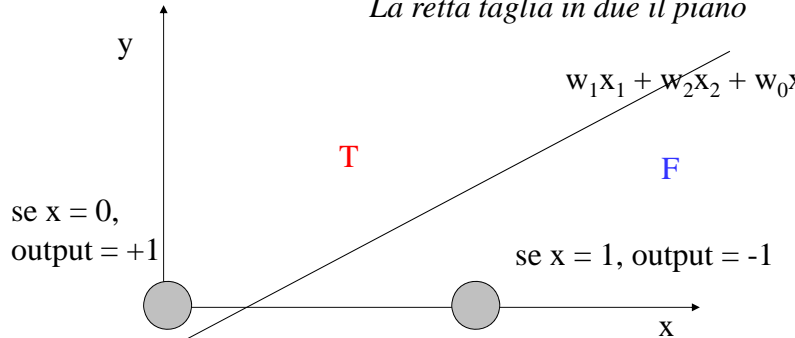
McCulloch-Pitts (1943)

e

Rappresentazione della retta

La retta taglia in due il piano



$$w_1x_1 + w_2x_2 + w_0x_0 = 0$$

$y = mx + q$
 $m = 1, q = -1$

$y = wx - \mu$
 $x_2 = w_1x_1 - \mu$



$w_1x_1 + w_2x_2 + w_0x_0 = 0$
 $\mathbf{w} \cdot \mathbf{x} = 0$
 $w_2 = -1; x_0 = 1$
 $w_1 = m; w_0 = q$

$$z = \Theta(w_j x_j(t))$$

A.A. 2012-2013

9/53

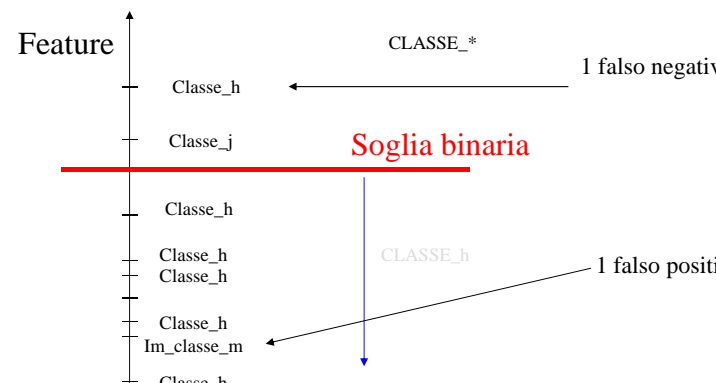
<http://homes.dsi.unimi.it/~borghese>

Classificatore binario

Classificatore binario. Si seleziona una feature, f , e si sceglie la soglia ottimale, w .

$$h_i = \Theta(f_i - \mu_i)$$



A.A. 2012-2013

10/53

<http://ais-lab.dsi.unimi.it>



ADA(ptive) Boosting

L'intuizione ci ha portato a definire delle feature associate alle cinque classi rimaste: regolari, irregolari, allungati, fili e insetti. Ma non ci sono feature singole che consentono la classificazione corretta. Sono feature "deboli".

Il boosting consiste in un metodo incrementale che produce un classificatore composto da più classificatori elementari, binari, in grado di minimizzare l'errore sul training set.

Combina più classificatori "deboli" in un classificatore performante.

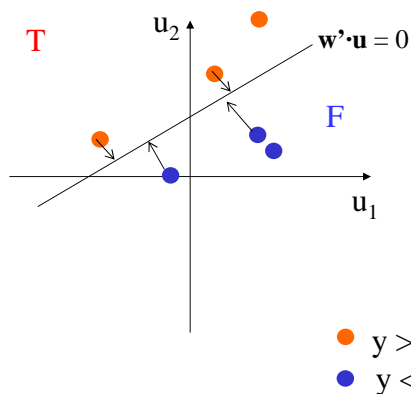
Il risultato del booster è dato dal voto di maggioranza dei risultati pesati ottenuti da classificatori binari elementari che sono stati selezionati durante l'esecuzione del boosting.

Si pu estendere il boosting anche ad una classificazione multi-classe

$$H = \text{sign} \left(\sum_{i=1}^N \alpha_i h_i(\cdot) \right) \quad h_i = \Theta(f_i - \mu_i)$$



Support vector Machines



Definizione dei parametri w , tale per cui sia massimo il **margin**.

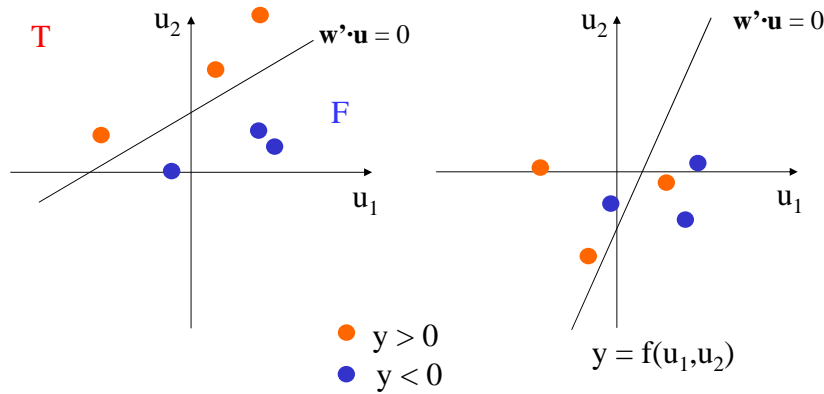


Funzioni linearmente separabili



Linearmente separabile

Non linearmente separabile



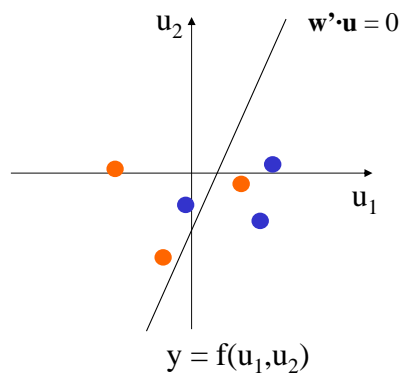
A.A. 2012-2013

13/53

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



Mappatura dello spazio di input



Se voglio mantenere il separatore lineare, devo modificare lo spazio.

Mappatura dello spazio u_1 u_2 in uno spazio in cui i dati siano linearmente separabili.

Introduzione di una funzione **Kernel** (non lineare) che consente di ottenere una curva di separazione non-lineare.

A.A. 2012-2013

14/53

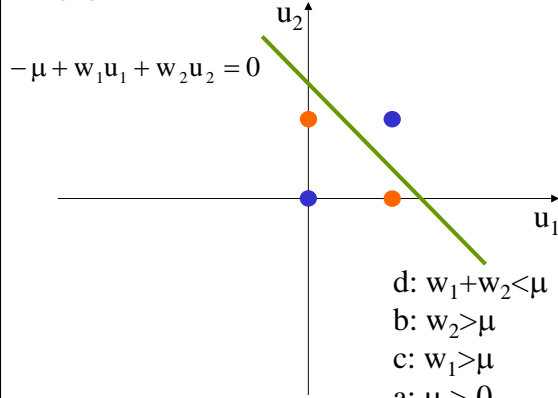
<http://homes.dsi.unimi.it/~borghese>



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



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



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

a

b

c

d

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.

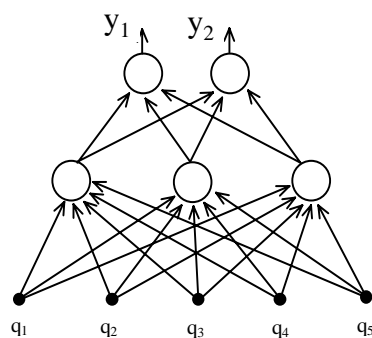
● $y(u_1, u_2, 1) = 1$ ● $y(u_1, u_2, 1) = -1$

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

A.A. 201: Si possono imparare solamente funzioni linearmente separabili jhese



Spiking neurons



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

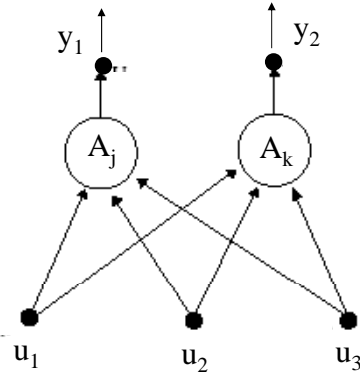
Connessionismo classico. Uscita compresa tra min – Max. **Frequenza di scarica.**



La rete neurale ad un livello



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



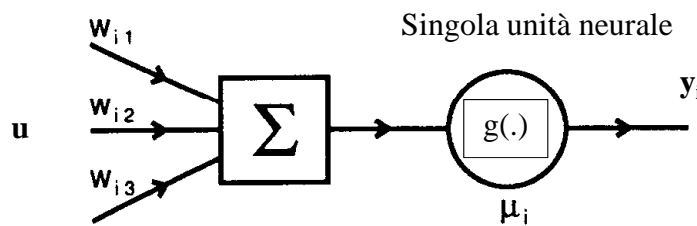
$$y_i = g(w_{ij}u_j - \mu_i)$$

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: $y_i = w_{ij}u_j - \mu_i$

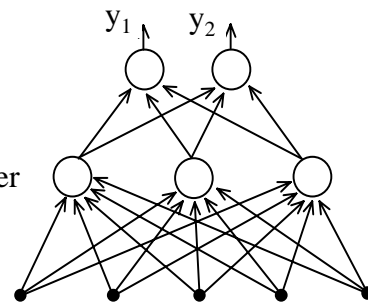


Una rete neurale a più livelli



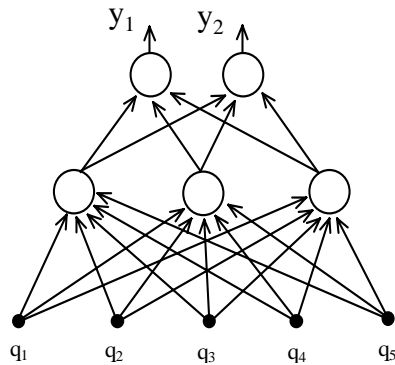
$$y_i = g(w_{ij}u_j - \mu_i)$$

Unità nascoste – Hidden layer





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. 2012-2013

19/53

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



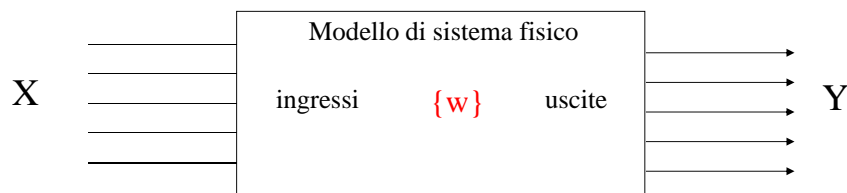
Complessità della funzione realizzabile

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

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

$$y_i = g(X)$$

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

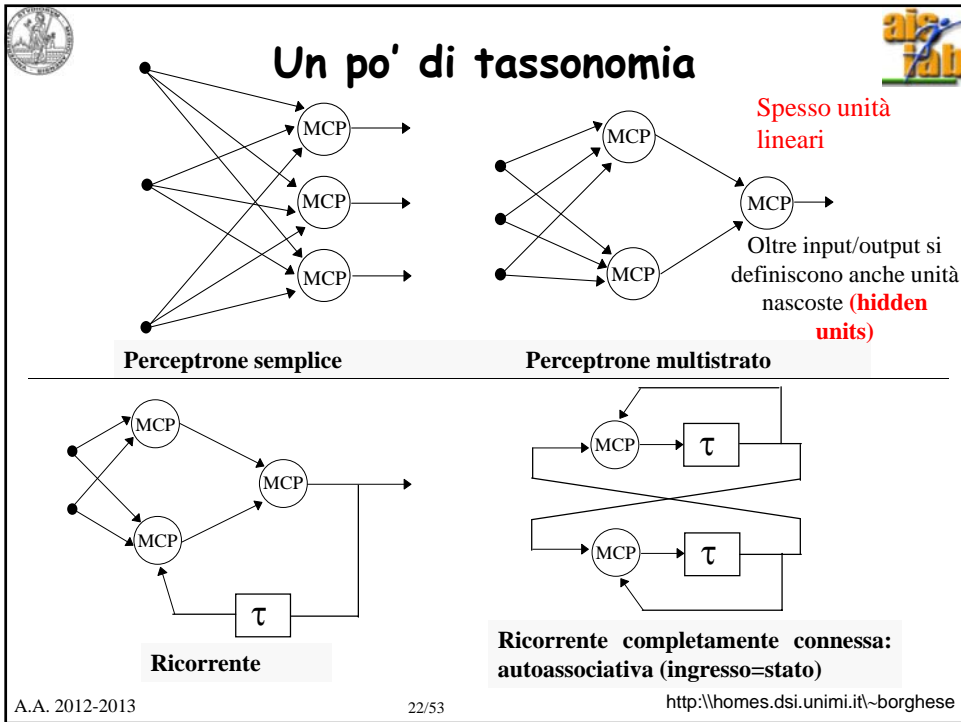
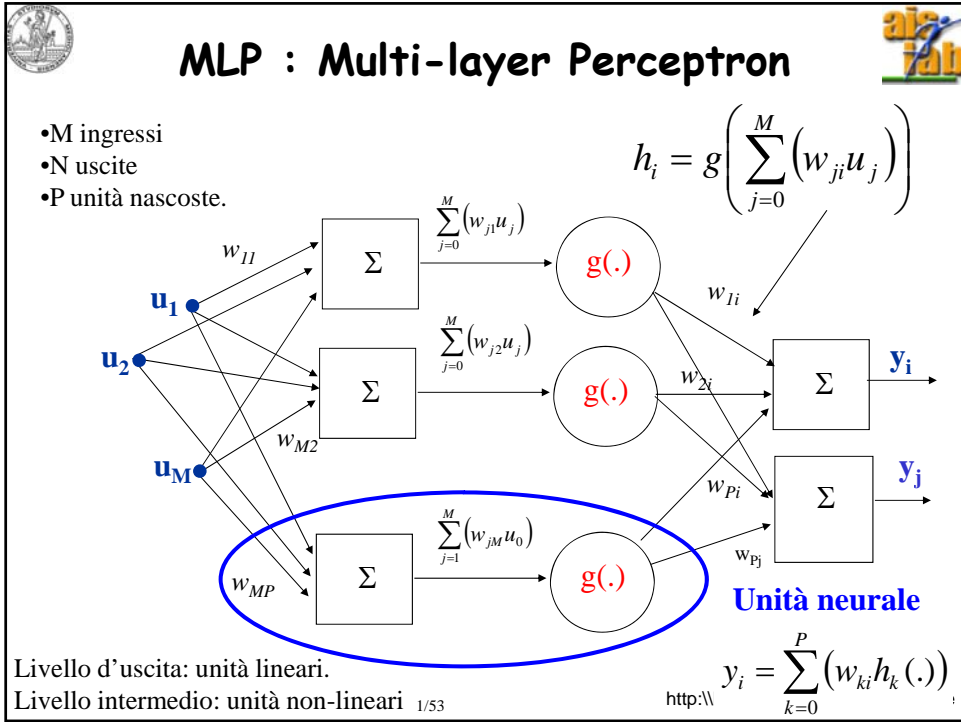


Reti neurali = approssimatori universali.

A.A. 2012-2013

20/53

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





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.

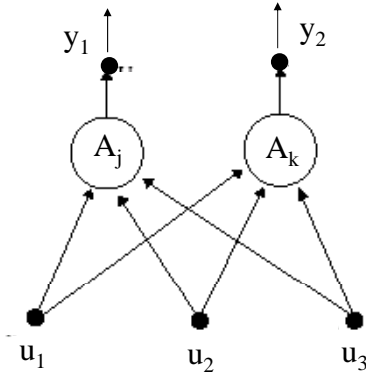
Apprendimento con rinforzo (reinforcement learning, learning with a distal teacher). L'ambiente fornisce un'informazione del tipo success or fail.



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.

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

A.A. 2012-2013

25/53

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



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.

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

A.A. 2012-2013

26/53

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



Apprendimento supervisionato

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

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

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

$$\text{Gradiente: } \nabla J(w) = \frac{\partial J(\{w\} | \dots)}{\partial w_1} \frac{w_1}{|w_1|} + \frac{\partial J(\{w\} | \dots)}{\partial w_2} \frac{w_2}{|w_2|} + \frac{\partial J(\{w\} | \dots)}{\partial w_3} \frac{w_3}{|w_3|} + \frac{\partial J(\{w\} | \dots)}{\partial w_4} \frac{w_4}{|w_4|} + \dots$$

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.

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

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



La pratica dell'apprendimento supervisionato



Fino a quando l'apprendimento non è stato completato:

1. Presentazione di un pattern di input / output (**dati**).
2. Calcolo dell'output della rete con il pattern corrente (**modello**)..
3. Calcolo dell'errore (**distanza**)..
4. Calcolo dei gradienti (**apprendimento**).
5. Calcolo dell'incremento dei pesi (**apprendimento**).

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.

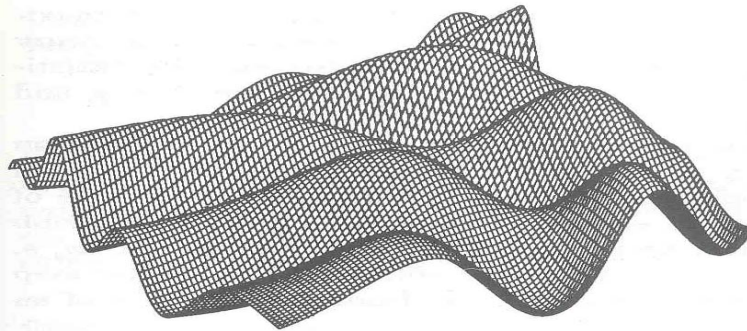
Reti multi-strato hanno elevata capacità computazionale, ma anche elevata complessità'.



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 $J(w | \dots)$



A.A. 2012-2013

31/53

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



Problemi



Quando si termina l'algoritmo di apprendimento?

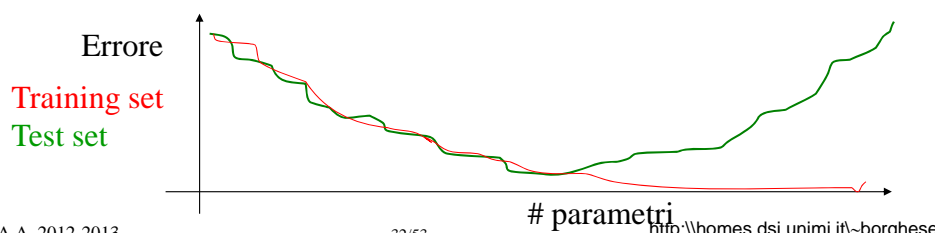
Bootstrap – Vengono estratti pattern con ripetizioni.

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.



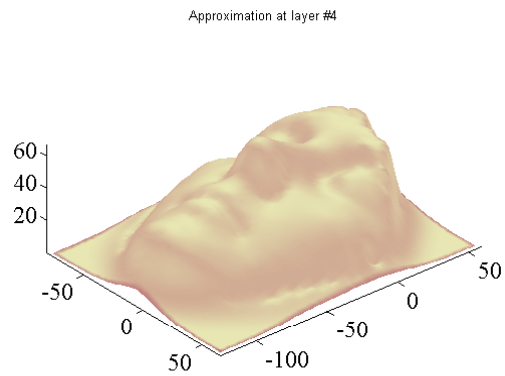
A.A. 2012-2013

32/53

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



Problema dell'overfitting dovuto a sovrapparametrizzazione



A.A. 2012-2013

33/53

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



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. 2012-2013

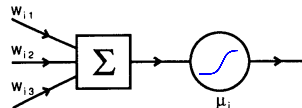
34/53

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



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 ($g(\cdot) = 1$):

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

Soluzione di un sistema lineare nei pesi!!

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

A.A. 2012-2013

35/53

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



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. 2012-2013

36/53

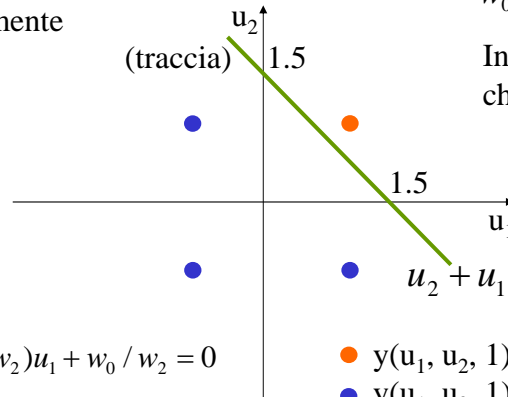
<http://homes.dsi.unimi.it/~borghese>



Esempio - AND



Troviamo la soluzione graficamente



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

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

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

$$u_2 + u_1 - 1.5 = 0$$

⇓

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

$$u_2 + u_1 - 1.5 = 0$$

$$y(u_1, u_2, 1) = 1$$

$$y(u_1, u_2, 1) = -1$$

$$w_0 = -1.5$$

$$w_1 = 1$$

$$w_2 = 1$$

Esistono più soluzioni
Separabilità lineare.

A.A. 2012-2013

37/53

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



Problema lineare



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

Matrice dei termini noti: $b = y^D$

$$b = \begin{bmatrix} -1 \\ -1 \\ -1 \\ +1 \end{bmatrix}$$

Matrice dei coefficienti: A

$$A = \begin{bmatrix} +1 & -1 & -1 \\ +1 & +1 & -1 \\ +1 & -1 & +1 \\ +1 & +1 & +1 \end{bmatrix}$$

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

Vettore delle incognite: $x = w$

$$Ax = b$$

$$x = (A^* A)^{-1} A^* b \rightarrow w = \begin{bmatrix} +0.5 \\ +0.5 \\ -0.5 \end{bmatrix}$$

Minimizzo implicitamente la distanza tra la retta ed i 4 punti.

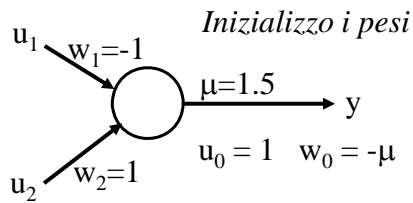
A.A. 2012-2013

38/53

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



Delta rule I: calcolo dell'uscita



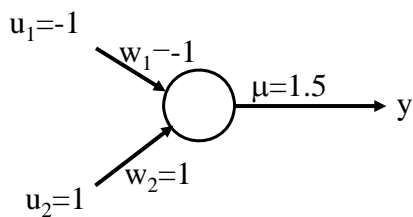
$$U = \{-1, 1\} \quad y^D = -1$$

u_1	u_2	y	y^D
-1	-1		-1
-1	1	0,5	-1
1	-1		-1
1	1		+1

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



Delta rule II: Calcolo dell'errore



u_1	u_2	y	y^D
-1	-1		-1
-1	1	0,5	-1
1	-1		-1
1	1		+1

$$y = \sum_{i=0} (w_i u_i) = 0.5$$

$$\text{Errore} = (y^D - y)^2 = (-1 - (0.5))^2 = -1.5^2$$



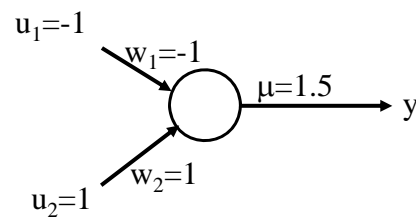
Delta rule III: calcolo del gradiente



$$\frac{d\text{Errore}}{dw_1} = (y_i^D - y_i)u_1 = (-1 - 0.5)(-1) = 1.50$$

$$\frac{d\text{Errore}}{dw_2} = (y_i^D - y_i)u_2 = (-1 - 0.5)(+1) = -1.50$$

$$\frac{d\text{Errore}}{d\mu} = -\frac{d\text{Errore}}{dw_0} = +1.50$$



A.A. 2012-2013

41/53

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



Delta rule IV: aggiornamento pesi



$$\Delta w_{ij} = +\eta(y_j^D - y_j)u_i \quad U = \{-1, 1\} \quad y^D = -1$$

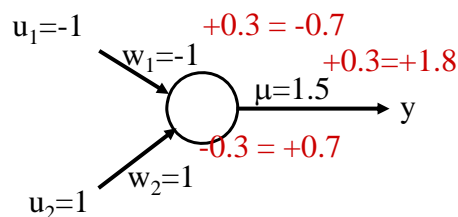
$$\eta = 0.2$$

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

$$\Delta w_2 = \eta(y_i^D - y_i)u_2 = \eta(-1 - 0.5)(1) = -0.3$$

$$\Delta w_0 = \eta(y_i^D - y_i)u_0 = \eta(-1 - 0.5)(1) = -0.30$$

$$\Delta \mu = -\Delta w_0 = +0.30$$



A.A. 2012-2013

42/53

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



Delta rule V: Nuovo valore di uscita

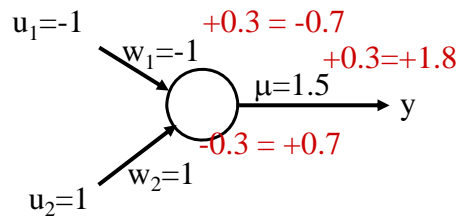


$$U = \{-1, 1\} \quad y^D = -1$$

$$\eta = 0.2$$

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

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



A.A. 2012-2013

43/53

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

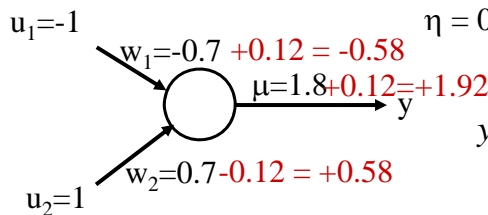


Delta rule VI- Nuovo aggiornamento



$$U = \{-1, 1\} \quad y^D = -1$$

$$\eta = 0.2$$



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

$$-0.76$$

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

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

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

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

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

A.A. 20

3se

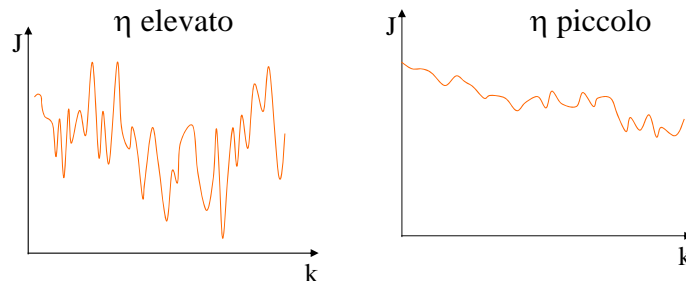


Ruolo di η – learning rate

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

Calmiera il Δw_{ij} per evitare che :

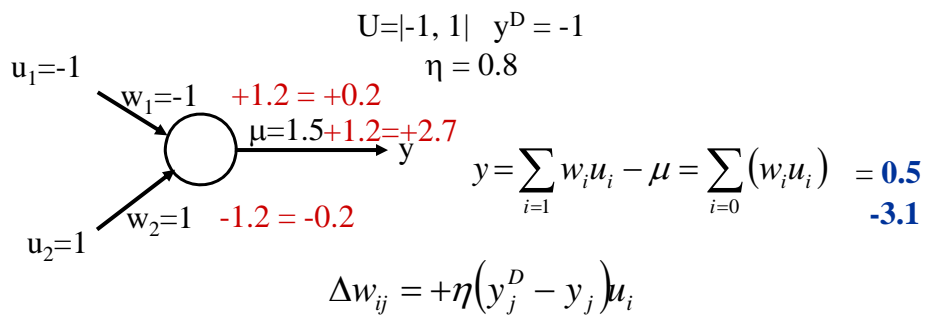
- Un peso sia specifico di un'unità ingresso-uscita.
- Oscillazioni durante l'apprendimento senza convergenza.



A.A. 2012-2013 η può variare durante l'addestramento. <http://homes.dsi.unimi.it/~borgnese>



Esempio di delta rule - Cattiva scelta di η



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

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

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

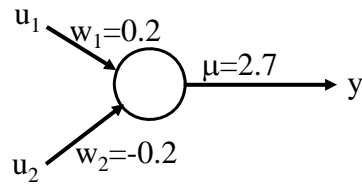
A.A. 2012-2013

46/53

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



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 $y = \sum w_i u_i - \mu = \sum (w_i u_i) = (0.2)(-1) + (-0.2)(1) - 2.7 = -3.1$

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. 2012-2013

47/53

<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. 2012-2013

48/53

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



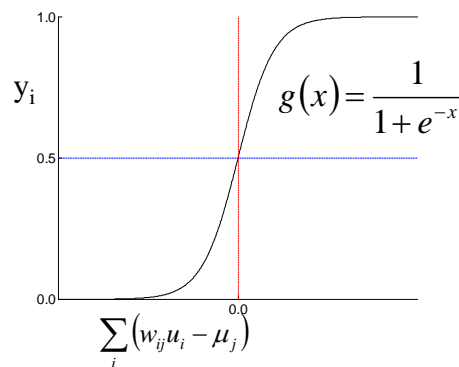
Perceptrone con unità di attivazione logistiche



$$g'(x) = g(x) \cdot (1 - g(x))$$

$$y_j = g\left(\sum_i w_{ij} u_i - \mu_j\right)$$

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



A.A. 2012-2013

49/53

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



Update dei pesi per funzione logistica



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

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

δ rule

derivata

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.

A.A. 2012-2013

50/53

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



Riassunto - topologia



I neuroni connessioneisti 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 unita'.

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