Le reti neurali: una metodologia per l'apprendimento automatico

Giancarlo Ferrari-Trecate

ferrari@aut.ee.ethz.ch
ferrari@conpro.unipv.it

Sommario

- Modelli matematici
- Modelli black box: il problema dell'apprendimento
- Mal posizione dei problemi di apprendimento e concetto di generalizzazione
- Il percettrone di Rosenblatt
- Le reti neurali Multilayer Perceptron
- L'algoritmo di backpropagation

Modelli matematici

Il concetto di modello matematico di un sistema e' fondamentale nella scienza moderna

Esempi di sistemi:

Meccanica:
dato il moto delle pale di un elicottero studiare le vibrazioni a cui e' sottoposto il seggiolino del guidatore

Metereologia:
date temperatura, pressione atmosferica e livello di nuvolosita' predire che tempo fara' domani

Economia
date delle misurazioni sullo "stato si salute" dell'economia predire l'indice MIB

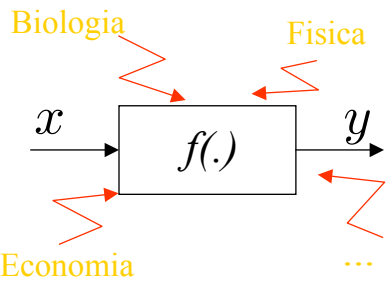
Biologia:
Date delle misurazioni della concentrazione di un ormone nel sangue, ricostruire il profilo secretorio della ghiandola che lo genera

Giochi:
Backgammon / scacchi: data una posizione di gioco scegliere la mossa che piu' probabilmente portera' alla vittoria

Utilita' dei modelli

- Descrizione sintetica di un sistema
- Possibilita' di fare predizioni

Astrazione: modello matematico



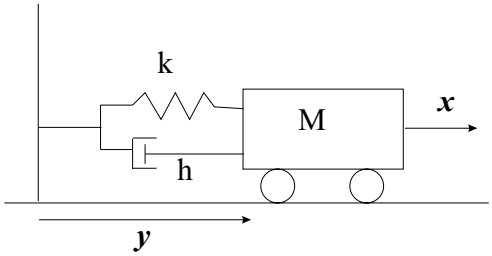
x: ingresso
y: uscita

Meccanica:
dato il moto delle pale di un elicottero studiare le vibrazioni a cui e' sottoposto il seggiolino di del guidatore

Metereologia:
date temperatura, pressione atmosferica e livello di nuvolosita' predire che tempo fara' domani

L'approccio classico: modellistica white box

- Il sistema e' una "scatola" trasparente di cui si conoscono le componenti interne ed il loro funzionamento
- Utilizzare le leggi costitutive per inferire il modello di un sistema



Vantaggi:

- Non servono dati sperimentali e spesso neppure il sistema in esame !

Svantaggi:

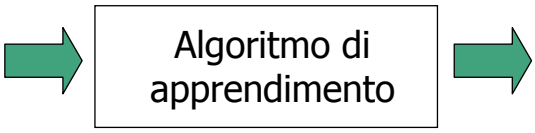
- Ipotesi semplificative
- Inadatto per sistemi complessi
- Inadatto per sistemi senza precise leggi costitutive

Un approccio alternativo: modellistica black box

- Il sistema e' una "scatola" nera: cio' che contiene e' inaccessibile o non e' noto
- Utilizzare dati sperimentali per derivare il modello del sistema

Dati sperimentali:

$(\bar{x}^j, y^j) \quad j = 1, \ldots, n$



Modello matematico !

Vantaggi:

- Non servono leggi costitutive !
- Adatto a descrivere un sistema nelle condizioni operative rispecchiate dai dati sperimentali

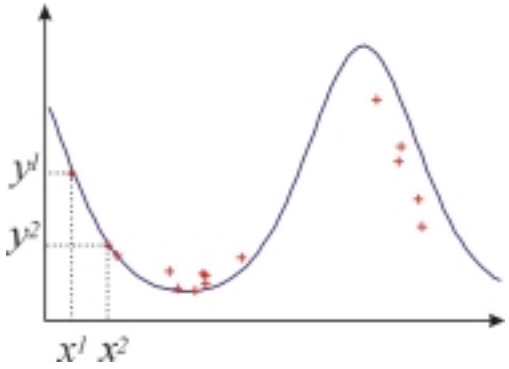
Svantaggi:

- Il modello e' specifico per il sistema in esame.
- Perdita del significato fisico dei parametri

Modellistica black box = Apprendimento

Problema: Ricostruire una funzione incognita $f(\cdot) : X \subseteq \mathbf{R}^d \mapsto \mathbf{R}$
da un numero finito di campioni (che possono essere affetti da rumore)

$$y^j = f(\bar{x}^j) + \epsilon^j \qquad j = 1, \ldots, n$$



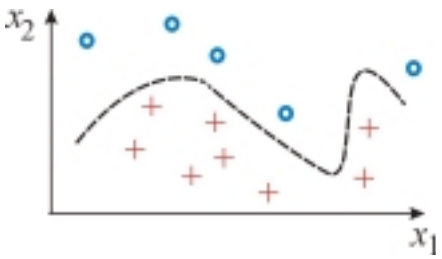
Dati sperimentali	=	Esempi
Inferire il modello matematico	=	Apprendimento basato sugli esempi

Le reti neurali (ma anche procedure piu' classiche come i minimi quadrati) sono algoritmi per risolvere problemi di modellistica black box

Problemi di apprendimento: tassonomia

Dati sperimentali non rumorosi

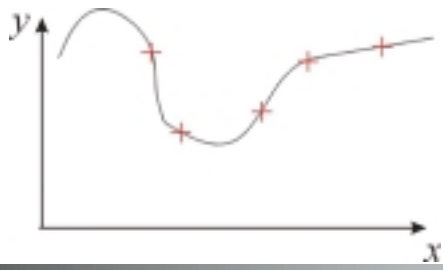
- $f(\bar{x}) \in \{0, 1, \ldots, q\}$ problema di classificazione



cerchi: $y=1$

croci: $y=-1$

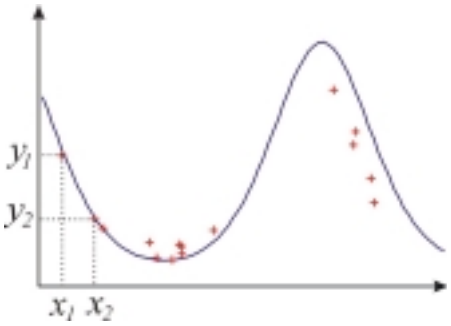
- $f(\bar{x}) \in \mathbf{R}$ problema di interpolazione



Problemi di apprendimento: tassonomia

Dati rumorosi e funzione a valori reali: regressione

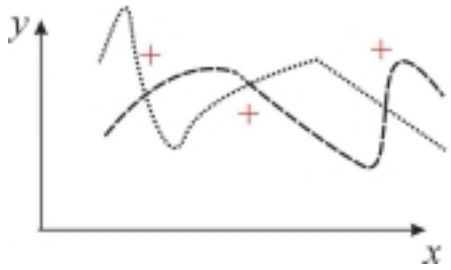
$$y^j = f(\bar{x}^j) + \epsilon^j \qquad j = 1, \ldots, n$$

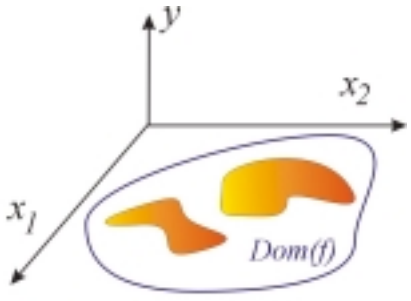


- La funzione $f(\cdot) : X \subseteq \mathbf{R}^d \mapsto \mathbf{R}$ è, in generale, non lineare
- I dati $\{y^j\}_{j=1}^n$ sono rumorosi. ⇒ No interpolazione; Filtraggio del rumore

Difficolta' dei problemi di apprendimento

- L' insieme $\{\bar{x}^j\}_{j=1}^n$ costituisce un campionamento non uniforme ed incompleto del dominio



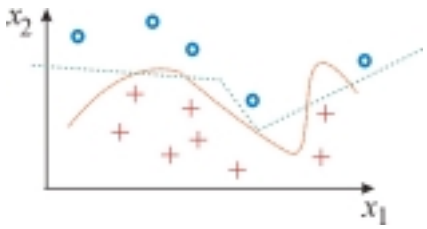


- La dimensione d del dominio della funzione puo' essere molto elevata !

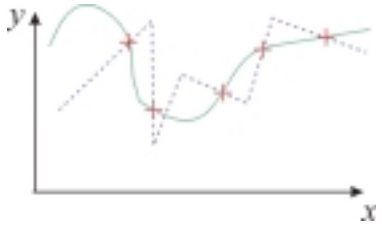
I problemi di apprendimento sono mal posti

Non si ha unicita' della soluzione !

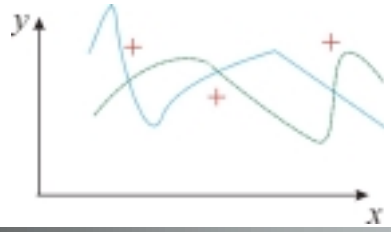
Classificazione



Interpolazione



Regressione



Scelta del modello migliore

Un modello matematico e' "buono" se consente di ottenere predizioni attendibili

Test di cross-validazione

Considero un insieme di dati sperimentali NON utilizzati per addestrare il modello $(\tilde{x}^j, \tilde{y}^j) \quad j = 1, \ldots, n_v$

Il modello e' "buono" se, alimentato con i campioni di ingresso $\tilde{x}^j$ predice in modo soddisfacente i campioni di uscita $\tilde{y}^j$

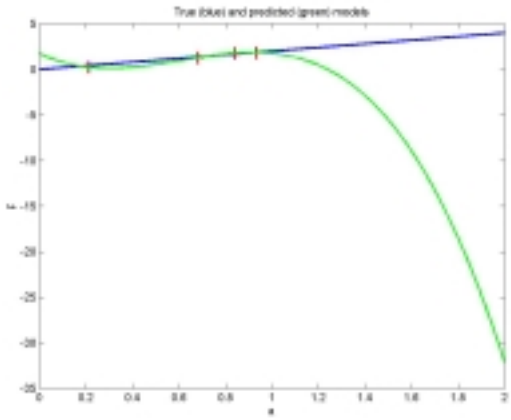
Quando un modello da predizioni attendibili per qualunque dato sperimentale ammissibile, si dice che ha buone capacita' di generalizzazione !

Apprendere la legge di Newton ...

Esperimento: si imprime ad una massa un'accelerazione (a) e si misura la forza corrispondente (F). Si ripete l'esperimento 4 volte e si misurano 4 dati sperimentali (che sono debolmente rumorosi ...)

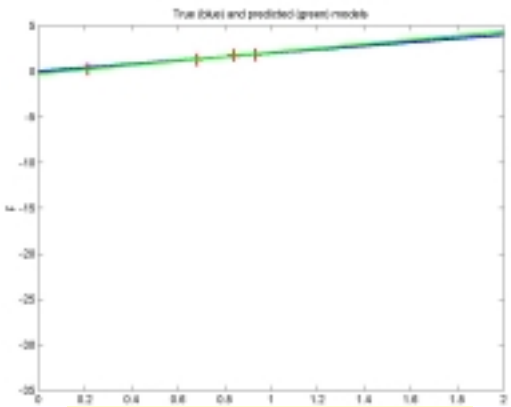
Problema: come dipende la forza dall'accelerazione ?

Interpolazione



Pessima generalizzazione !

Regressione



Buona generalizzazione

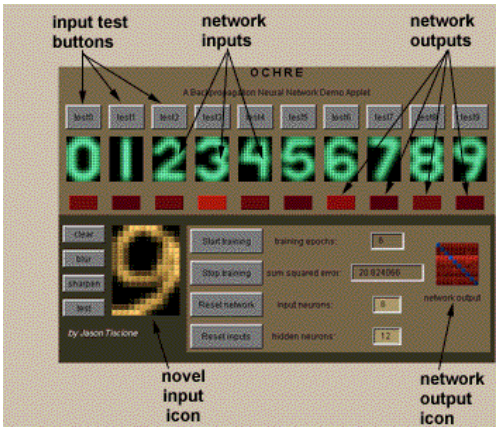
Vantaggio delle reti neurali

Le reti neurali si differenziano da altri algoritmi di apprendimento (come i minimi quadrati) per le loro migliori capacita' di generalizzazione !!

Modelli difficili :

- Riconoscimento di caratteri digitalizzati (reti neurali sono usate dalle poste americane per riconoscere il C.A.P.)

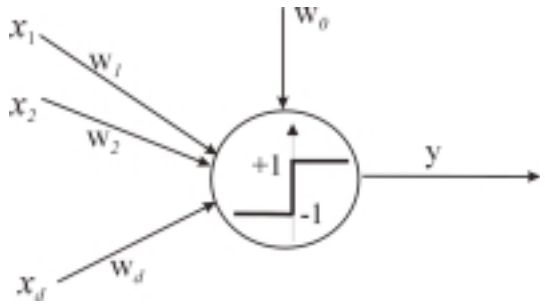
http://www.geocities.com/SiliconValley/2548/ochre.html



- Backgammon: Bill Robertie (campione del mondo per due volte) e' stato battuto 13 volte su 31 incontri da una rete neurale !!

Il Percettrone

Modello del neurone di McCullough e Pitts (1943)

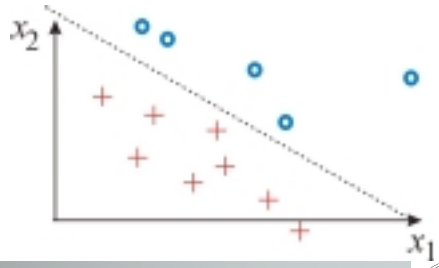


$$\bar{w}' = [\, w_0 \quad w_1 \quad \cdots \quad w_d \,]$$

$$\bar{x}' = [\, 1 \quad x_1 \quad \cdots \quad x_d \,]$$

Rosenblatt (1962): perceptron. Rosenblatt propose un algoritmo per tarare i *pesi* $\bar{w}$ in base ai dati sperimentali in modo da risolvere un problema di classificazione (in due classi)

$$\bar{x}' \cdot \bar{w} = 0$$

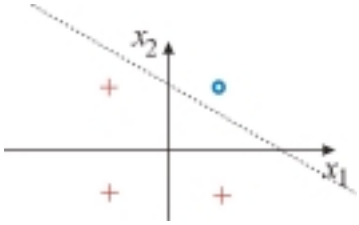


Limite del percettrone

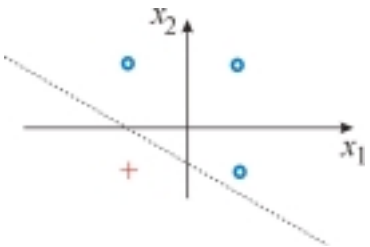
Se gli ingressi sono binari, il percettrone implementa una funzione logica !!

y = valore di verita' { cerchi: $y=1$; croci: $y=-1$

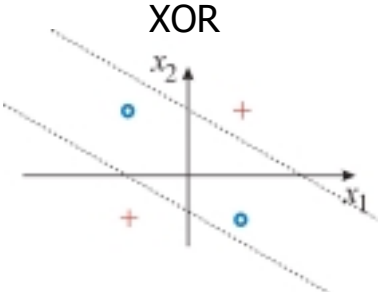
AND



OR

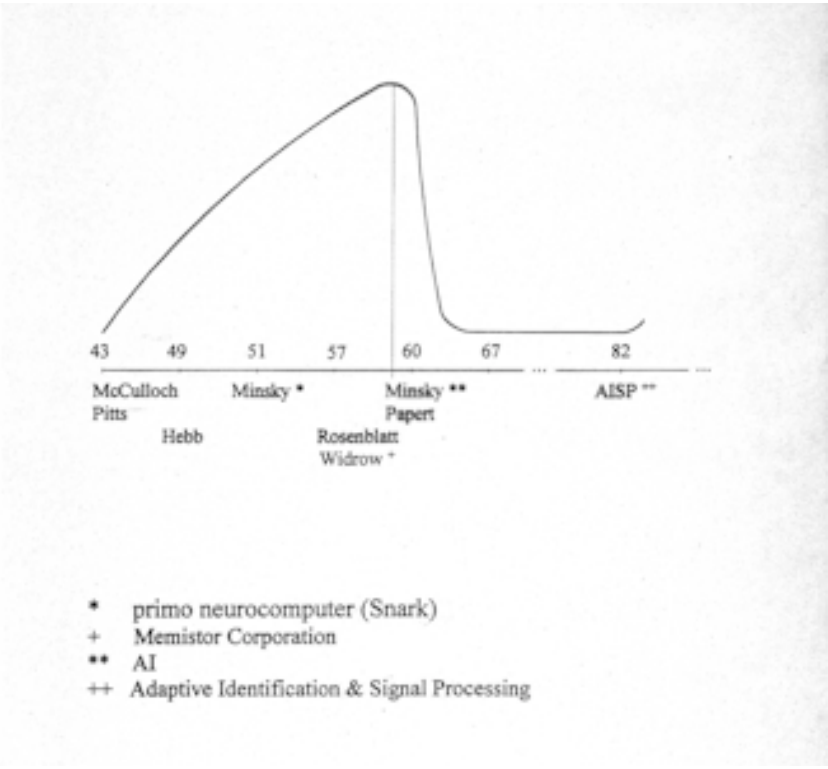


XOR

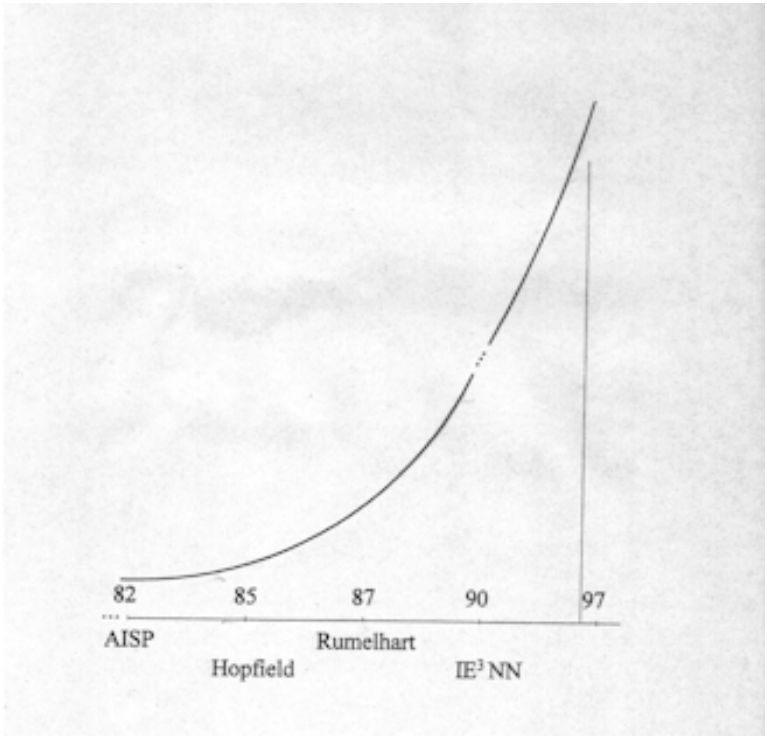


Servono 2 rette per la funzione XOR: non puo' essere implementata dal percettrone !!

Storia delle Reti Neurali - 1



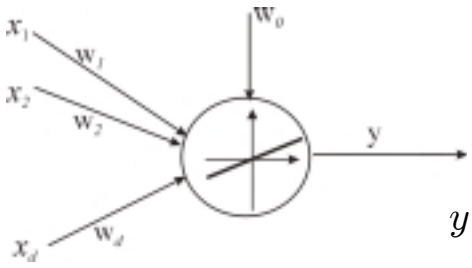
Storia delle reti neurali - 2



Percettroni per interpolazione e regressione

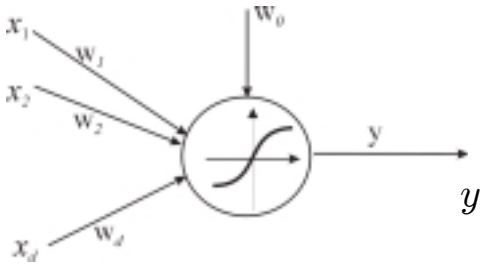
Percettrone lineare:

- implementa solo una funzione lineare



$$y = (\bar{x}' \cdot \bar{w} - w_0)\beta$$

Percettrone sigmoidale:



$$y = \sigma(\bar{x}' \cdot \bar{w} - w_0)$$

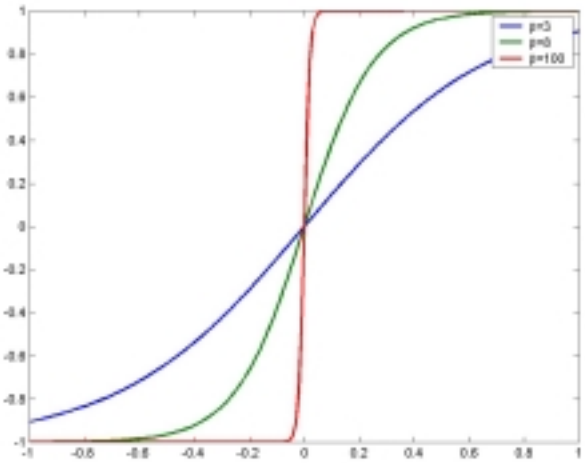
$\sigma(\cdot)$: funzione di attivazione sigmoidale, cioe' monotona, non decrescente tale che

$$\lim_{z \to +\infty} \sigma(z) = 1 \qquad \lim_{z \to -\infty} \sigma(z) = -1$$

Vantaggi della sigmoide

Esempi di sigmoidi:

$$\sigma(z) = \begin{cases} tanh(\beta z) \\ \frac{2}{1+exp(-\beta z)} - 1 \end{cases}$$

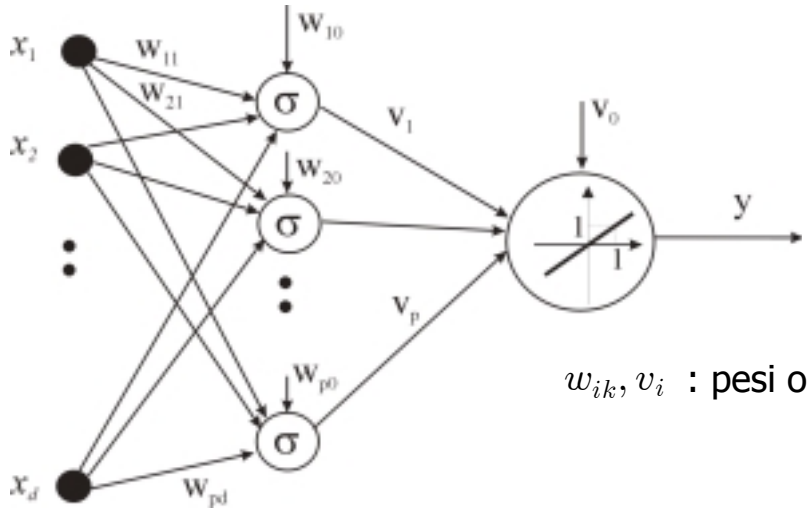


- E' una funzione nonlineare
- Per opportuni valori del parametro assomiglia alla funzione segno (funzioni "quasi logiche")
- A differenza della funzione segno e' differenziabile ovunque

Tuttavia con un solo percettrone non posso implementare una classe di funzioni abbastanza ricca ...

Rete neurale MultiLayer Perceptron (MLP)

Rete MLP a due strati, completamente connessa

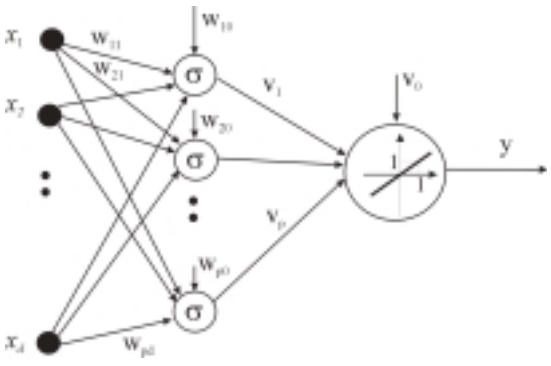


w_{ik}, v_i : pesi o coefficienti

p percettroni nello strato nascosto (hidden layer)
1 percettrone lineare nello strato d'uscita (output layer)

Proprieta' delle reti MLP

Si possono adottare anche reti a piu' strati ma MLP con due strati implementano classi di funzioni molto generali !



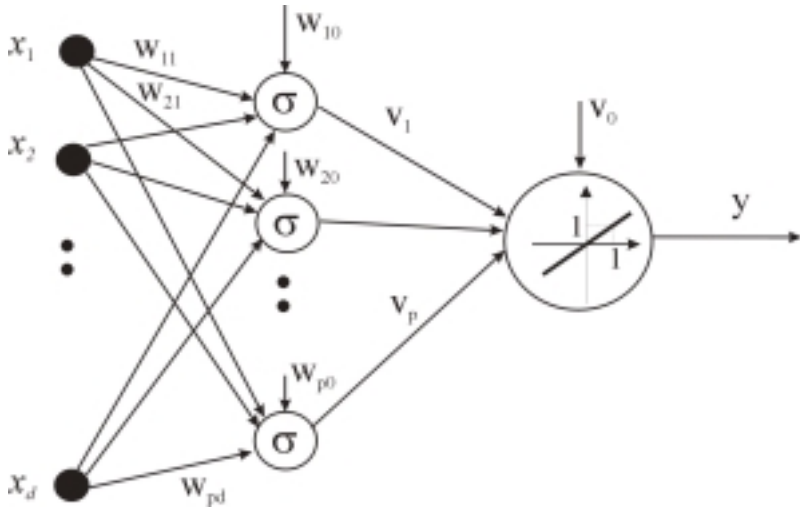
Teorema: se la sigmoide e' la funzione segno, le reti MLP a due strati possono implementare qualunque funzione logica

Teorema (Cybenko): per qualunque funzione continua su un compatto, esiste una rete MLP che (per una opportuna scelta dei pesi) la approssima con precisione arbitraria

$$\sup_{x \in \mathcal{X}} |MLP(x) - f(x)| < \epsilon$$

N.B. Entrambi i teoremi funzionano a patto di scegliere un numero di percettroni nello strato nascosto sufficientemente grande ...

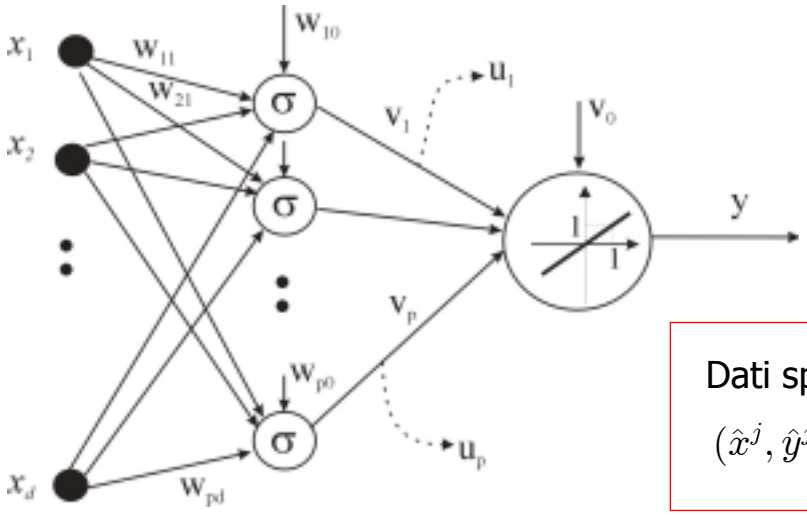
Addestramento delle reti MLP



Addestrare la rete significa assegnare i pesi w_{ik}, v_i in base ai dati sperimentali

Algoritmo di apprendimento

Backpropagation



Dati sperimentali:

$(\hat{x}^j, \hat{y}^j)\ \ j = 1, \ldots, n$

$$\bar{x}' = [\,1 \quad x_1 \quad \cdots \quad x_d\,] \qquad \bar{v}' = [\,v_0 \quad v_1 \quad \cdots \quad v_p\,]$$

$$\bar{w}_i' = [\,w_{i0} \quad w_{i1} \quad \cdots \quad w_{id}\,] \qquad \bar{u}' = [\,1 \quad u_1 \quad \cdots \quad u_p\,]$$

Scopo: trovare i pesi che minimizzano $R(\bar{w}_1, \ldots, \bar{w}_p, \bar{v}) = \frac{1}{2n}\sum_{j=1}^{n}(\hat{y}^j - y^j)^2$

Backpropagation

Scopo: trovare i pesi che minimizzano $R(\bar{w}_1, \ldots, \bar{w}_p, \bar{v}) = \frac{1}{2n}\sum_{j=1}^{n}(\hat{y}^j - y^j)^2$

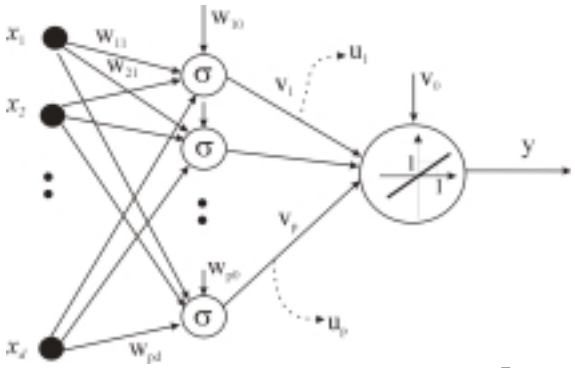
- Vincoli: $y^j - \bar{v}'\bar{u}^j = 0 \qquad j = 1, \ldots, n$
  $u_i^j - \sigma(\bar{w}_i'\bar{x}^j) = 0 \qquad i = 1, \ldots, p$

Lagrangiana:

$$L = \frac{1}{2n}\sum_{j=1}^{n}(\hat{y}^j - y^j)^2 + \frac{1}{n}\sum_{j=1}^{n}\left(\beta^j(y^j - \bar{v}'\bar{u}^j) + \sum_{i=1}^{p}\alpha_i^j(u_i^j - \sigma(\bar{w}_i'\bar{x}^j))\right)$$

- Condizioni necessarie per un punto di minimo:

$$\frac{\partial L}{\partial \alpha_i^j} = 0 \qquad \frac{\partial L}{\partial \beta^j} = 0$$
$$\frac{\partial L}{\partial y^j} = 0 \qquad \frac{\partial L}{\partial u_i^j} = 0$$
$$\frac{\partial L}{\partial \bar{w}_i} = 0 \qquad \frac{\partial L}{\partial \bar{v}} = 0$$



Backpropagation

Lagrangiana:

$$L = \frac{1}{2n}\sum_{j=1}^{n}(\hat{y}^j - y^j)^2 + \frac{1}{n}\sum_{j=1}^{n}\left(\beta^j(y^j - \bar{v}'\bar{u}^j) + \sum_{i=1}^{p}\alpha_i^j(u_i^j - \sigma(\bar{w}_i'\bar{x}^j))\right)$$

$\frac{\partial L}{\partial \alpha_i^j} = 0 \qquad \frac{\partial L}{\partial \beta^j} = 0$ Restituiscono i vincoli

$\frac{\partial L}{\partial y^j} = 0 \qquad \beta^j = \hat{y}^j - y^j$ Errori commessi sui dati sperimentali

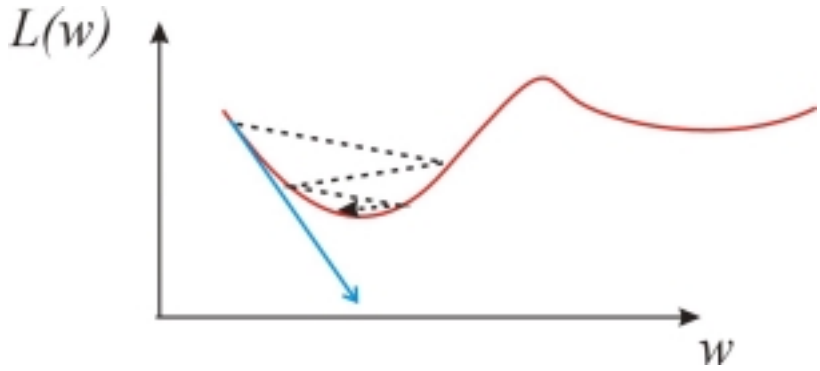
$\frac{\partial L}{\partial u_i^j} = 0 \qquad \alpha_i^j = \beta^j v_i$

$\frac{\partial L}{\partial \bar{w}_i} = 0 \qquad \sum_{j=1}^{n}\beta^j\bar{u}^j = 0$

$\frac{\partial L}{\partial \bar{v}} = 0 \qquad \sum_{j=1}^{n}\alpha_i^j\dot{\sigma}(\bar{w}_i'\bar{x}^j)\bar{x}^j = 0$

} Non sono immediatamente utili ...

Metodo del gradiente



- Update: $w(k+1) = w(k) - \eta \left.\frac{\partial L(w)}{\partial w}\right|_{w=w(k)} \qquad \eta > 0$
- Condizione di terminazione (Es. errore relativo $\frac{|L(w(k+1)) - L(w(k))|}{|L(w(k))|} \leq \epsilon$)

Backpropagation - algoritmo

1. Inizializzazione: w_{ik}, v_i scelti casualmente
2. Forward pass: per tutti gli esempi $(\hat{x}^j, \hat{y}^j)$ $j = 1, \ldots, n$ calcolo u_i^j, y^j $i = 1, \ldots, p$
3. Backward pass:

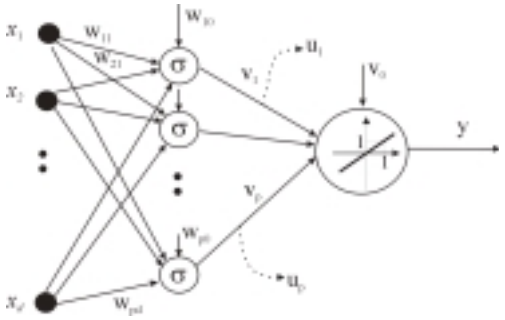
   calcolo $\beta^j = \hat{y}^j - y^j$ (errori d'uscita)

   calcolo $\alpha_i^j = \beta^j v_i$
4. Update dei pesi: $\bar{v} \leftarrow \bar{v} + \eta \sum_{j=1}^{n} \beta^j \bar{u}^j$

   $\bar{w}_i \leftarrow \bar{w}_i + \eta \sum_{j=1}^{n} \alpha_i^j \dot{\sigma}(\bar{w}_i' \bar{x}^j) \bar{x}^j$
5. Terminazione: se la condizione di terminazione del metodo del gradiente e' soddisfatta, esco, altrimenti riprendo dal passo 2

Problemi della backpropagation

- L'algoritmo discusso e' "by epoch" (propago *tutti i dati* in avanti e poi tutti gli errori all'indietro). Puo' essere computazionalmente oneroso se ho tanti dati !
  - Esiste una versione subottima "by pattern" (considero un dato alla volta e ne propago l'errore all'indietro)

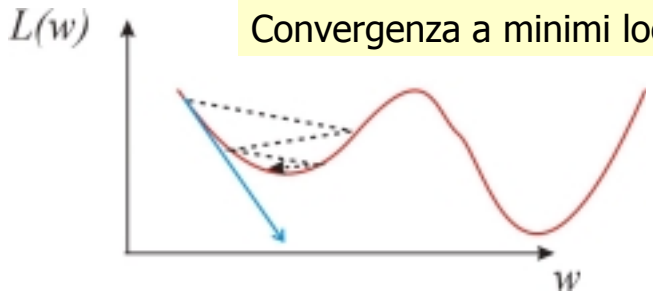


- Utilizza il metodo del gradiente che e' un algoritmo di ottimizzazione nonlineare
  - Possibile convergenza in minimi locali
  - Possibile convergenza lenta verso il minimo

Metodo del gradiente: problemi

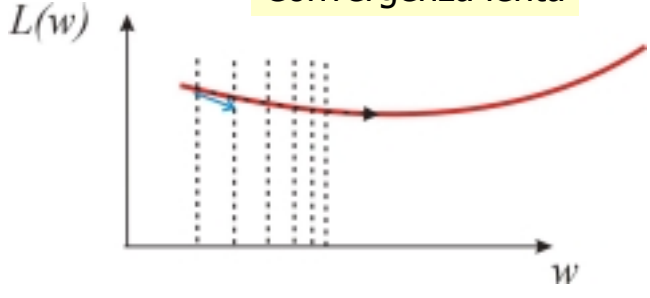
$$w(k+1) = w(k) - \eta \left.\frac{\partial L(w)}{\partial w}\right|_{w=w(k)}$$

Convergenza a minimi locali

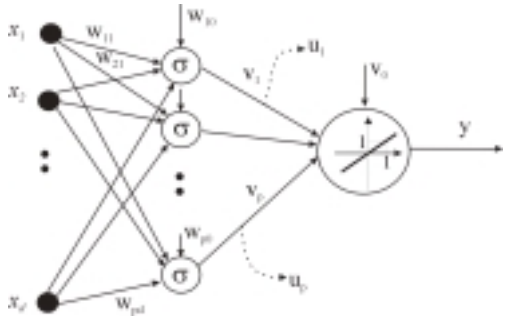


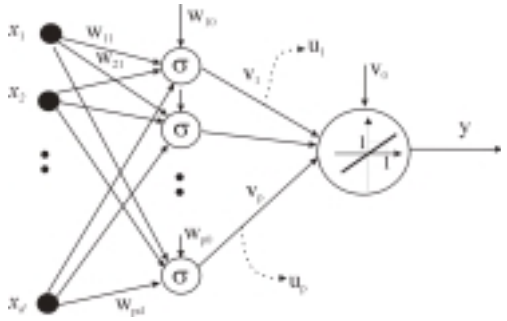
Rimedio parziale:
provare con diverse inizializzazioni

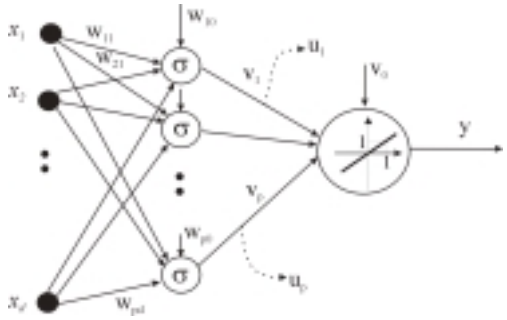
Convergenza lenta



Se la derivata e' piccola, l'algoritmo si muove a piccoli passi


x1
x2
xd
W10
W11
W21
Wp0
Wp1
σ
σ
σ
u1
up
v1
v0
vp
l
l
y


x1
x2
xd
W10
W11
W21
Wp0
Wp1
σ
σ
σ
u1
up
v1
v0
vp
l
l
y


σ
σ
σ
y