Le reti neurali: una metodologia per l'apprendimento automatico

(seconda parte)

Giancarlo Ferrari-Trecate

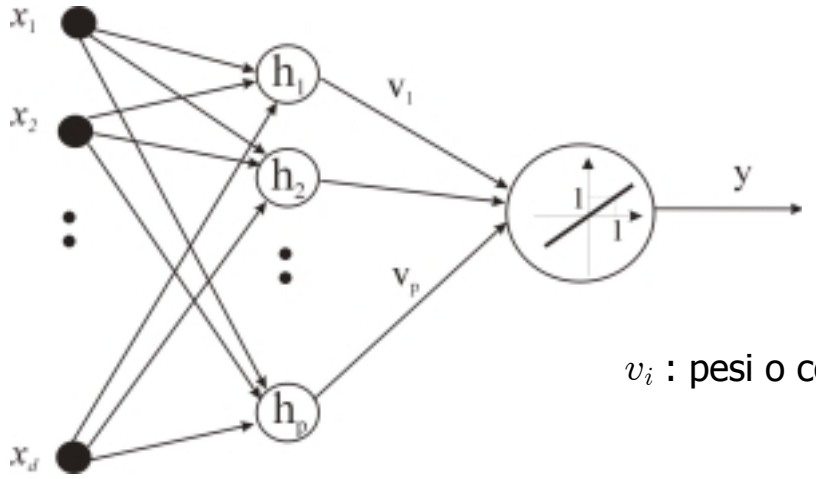
ferrari@aut.ee.ethz.ch
ferrari@conpro.unipv.it

Sommario

- Le reti neurali Radial Basis Function (RBF)
- Scelta della rete neurale ottimale:
  - Misura empirica di generalizzazione: la cross-validazione
  - Errore di generalizzazione
    - > Errore di approssimazione: il caso dei problemi di regressione
    - > Errore di stima: il caso dei problemi di classificazione
  - Metodi empirici per la taratura di reti MLP
- Conclusioni e referenze bibliografiche

Le reti neurali Radial Basis Functions (RBF)

Rete RBF:



v_i : pesi o coefficienti

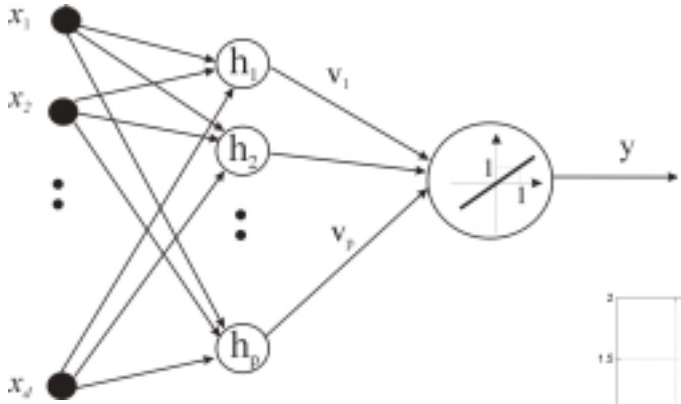
- p percettroni nello strato nascosto con funzioni di attivazione

$$h_i(\bar{x}) = h(\|\bar{x} - \bar{\xi}_i\|), \; h(\cdot) \text{ funzione continua}$$

$\bar{\xi}_i$: centri, $\bar{x}' = [\, x_1 \quad \cdots \quad x_d \,]$

$h_i(\cdot)$ e' simmetria radiale rispetto al centro $\bar{\xi}_i$

Reti neurali RBF: proprieta'



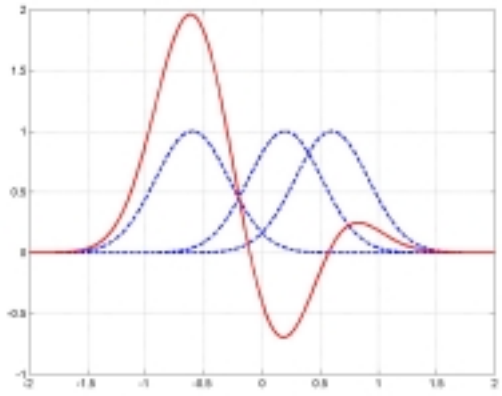
$$y = \sum_{i=1}^{p} v_i h(\|\bar{x} - \bar{\xi}_i\|)$$

Esempio:

$$h(z) = exp(\tfrac{-z^2}{0.2})$$

$$\bar{\xi}_1 = -0.6 \quad \bar{\xi}_2 = 0.2 \quad \bar{\xi}_3 = 0.6$$

$$v_1 = 2 \quad v_2 = -1 \quad v_3 = 0.5$$

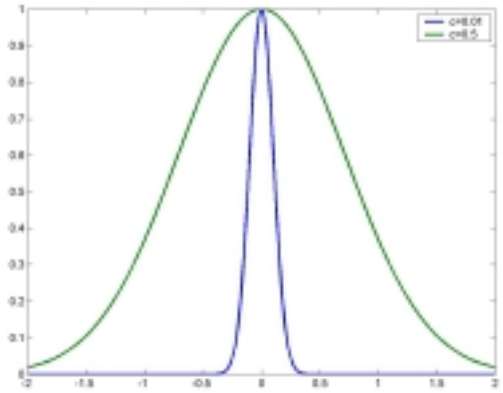


Se i centri sono fissati le reti neurali RBF sono lineari nei parametri

Funzioni di attivazione per reti neurali RBF

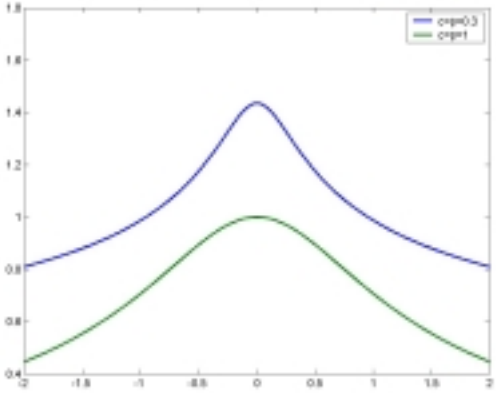
Gaussiana

$$h(z) = exp(\frac{-z^2}{2c^2})$$



Multiquadrica inversa

$$h(z) = (z^2 + c^2)^{-\beta} \quad \beta > 0$$



Se le funzioni di base sono "a campana" esse implementano dei campi ricettivi la cui ampiezza dipende dai parametri delle funzioni di attivazione

Algoritmi di addestramento (regressione)

Scopo: Abbiamo collezionato i dati sperimentali $(\hat{x}^j, \hat{y}^j)\ j = 1, \ldots, n$
Trovare i pesi e i centri che minimizzano

$$R(\bar{\xi}_1, \ldots, \bar{\xi}_p, \bar{v}) = \frac{1}{2n} \sum_{j=1}^{n} (\hat{y}^j - y^j)^2$$

$$R(\bar{\xi}_1, \ldots, \bar{\xi}_p, \bar{v}) = \frac{1}{2n} \sum_{j=1}^{n} \left(\hat{y}^j - \sum_{i=1}^{p} v_i h(\|\hat{x}^j - \bar{\xi}_i\|) \right)^2$$

- Il funzionale e' non lineare rispetto ai centri e quadratico rispetto ai pesi
  - Dobbiamo risolvere un problema di ottimizzazione non-lineare

Problema: Qualunque metodo di ottimizzazione (come il metodo del gradiente)
- e' computazionalmente costoso se il numero di neuroni e' alto
- e' possibile che termini in un minimo locale ...

Algoritmi di addestramento sub-ottimi

Idea: Spezzare l'algoritmo di addestramento in due parti:

1) Posizionare i centri

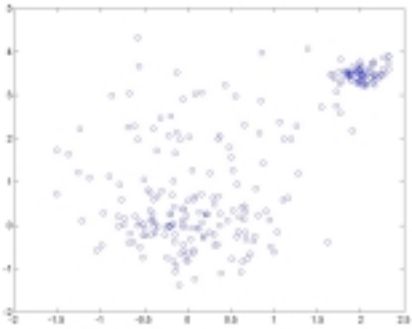
2) Trovare i pesi che minimizzano

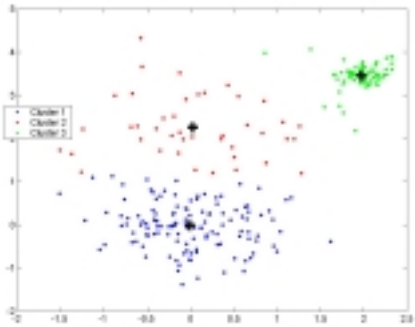
$$R(\bar{v}) = \frac{1}{2n}\sum_{j=1}^{n}\left(\hat{y}^j - \sum_{i=1}^{p} v_i h(\|\hat{x}^j - \bar{\xi}_i\|)\right)^2$$

1) Metodi per posizionare i centri:

- Griglia uniforme. Problema: inutile mettere dei centri dove ho pochi dati

- Algoritmi di clustering che dividono i dati $\hat{x}^j$ in p sottoinsiemi e ne trovano i "baricentri". Sono algoritmi computazionalmente efficienti !





Algoritmi di addestramento sub-ottimi

2) Trovare i pesi che minimizzano

$$R(\bar{v}) = \frac{1}{2n}\sum_{j=1}^{n}\left(\hat{y}^j - \sum_{i=1}^{p} v_i h(\|\hat{x}^j - \bar{\xi}_i\|)\right)^2$$

Risolvere un problema ai minimi quadrati per i quali esiste la formula esplicita

$$\Phi \in \mathbf{R}^{n\times p} \qquad \Phi_{ij} = h(\|\hat{x}^i - \bar{\xi}_j\|)$$

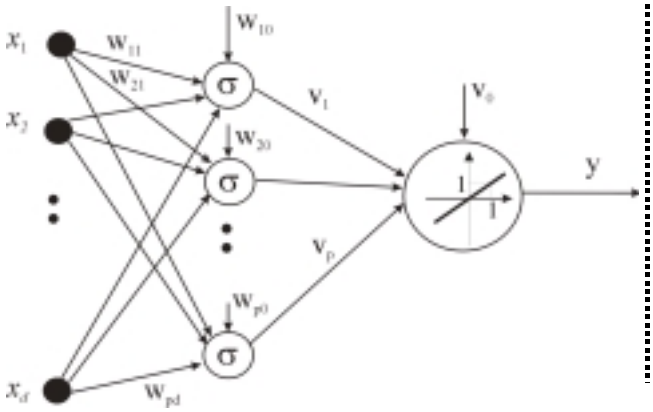
$$\hat{y}' = [\,\hat{y}^1 \quad \cdots \quad \hat{y}^n\,]$$

$$\bar{v}^o = \arg\min R(\bar{v}) = (\Phi'\Phi)^{-1}\Phi'\hat{y}$$

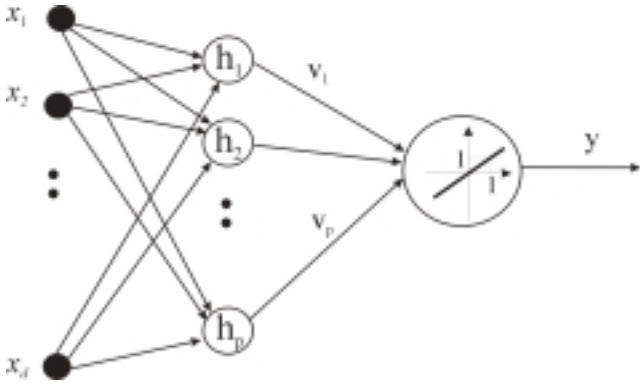
L'algoritmo proposto e' complessivamente sub-ottimo ma computazionalmente efficiente !

Problemi aperti

Reti neurali MLP



Reti neurali RBF



- Dato un insieme di dati sperimentali, qual e' il numero di neuroni ottimo ?
- Reti neurali MLP: e' necessario mantenere tutte le connessioni tra lo strato di ingresso o lo strato nascosto ?
- In generale, la funzione di attivazione dei neuroni dipende da parametri. Come li scelgo ?

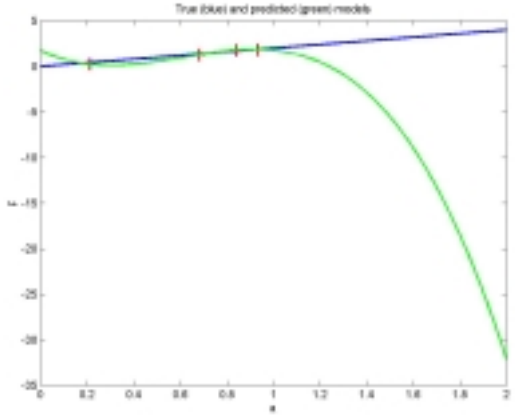
La risposta comune

L'architettura della rete neurale e i parametri delle funzioni di attivazione vanno scelti in modo da massimizzare la capacita' di generalizzazione !

Generalizzazione = capacita' di ottenere predizioni attendibili su nuovi dati (che NON sono stati usati per addestrare la rete neurale)

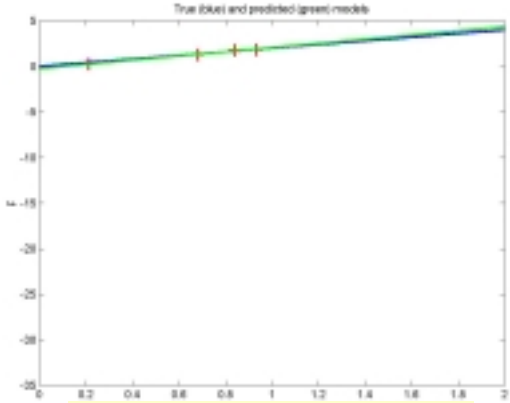
Ricordate i modelli della legge di Newton ?

Interpolazione



Pessima generalizzazione !

Regressione



Buona generalizzazione

Valutare la capacita' di generalizzazione: il metodo della cross-validazione

Test di cross-validazione

Si hanno a disposizione:

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

- Per confrontare la capacita' di generalizzazione di due reti neurali R1 e R2 gia' addestrate si calcolano

$$V_1 = \frac{1}{n_v} \sum_{j=1}^{n_v} \left(R1(\tilde{x}^j) - \tilde{y}^j\right)^2 \quad V_2 = \frac{1}{n_v} \sum_{j=1}^{n_v} \left(R2(\tilde{x}^j) - \tilde{y}^j\right)^2$$

Se $V_1 > V_2$ concludo che la rete neurale migliore e' R2 !

Il metodo di cross validazione consente di modificare l'architettura e/o i parametri di una rete neurale gia' addestrata e di stabilire se ho ottenuto un modello migliore del precendente.

Difetti della cross-validazione

- Si hanno complessivamente pochi dati e non si puo' sprecarne per la cross-validazione.

- I dati di validazione sono poco rappresentativi. Per esempio, se fossero molto "simili" a quelli di addestramento il test di cross-validazione indurrebbe a scegliere il modello che rappresenta meglio i dati di addestramento.

- La cross-validazione consente solo una procedura di *trial and error*
  1) modificare pesi e parametri della rete neurale e addestrarla
  2) valutare se la capacita' di generalizzazione e' migliorata. Se no ritornare al punto 1)

Non ho indizi su quali cambiamenti possano portare ad un miglioramento !

La procedura di trial and error puo' richiedere tempi lunghissimi prima di arrivare ad un modello soddisfacente

Necessita' di avere linee guida per la scelta del modello

Astrazione del problema di apprendimento

Problemi di apprendimento e generazione dei dati:

- Regressione $\hat{y}^j = f(\hat{x}^j) + \epsilon^j \quad j = 1, \ldots, n \quad f(\bar{x}) \in \mathbf{R}$
- Classificazione (2 classi) $\hat{y}^j = f(\hat{x}^j) \quad j = 1, \ldots, n \quad f(\bar{x}) \in \{-1, 1\}$

Ipotesi a priori

- $f(\cdot)$ appartiene ad uno spazio di funzioni $\mathcal{F}$. Esempio: $\mathcal{F} = \mathcal{C}(\mathbf{R}^d)$

- I dati $\{\hat{x}^j\}_{j=1}^n$ e i campioni di rumore $\{\epsilon^j\}_{j=1}^n$ sono realizzazioni di variabili casuali.

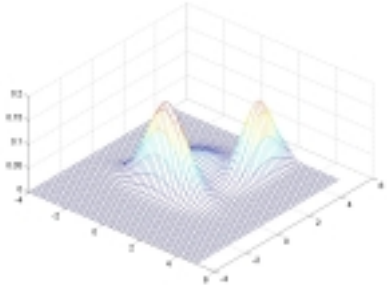
Breve introduzione alle variabili casuali

Una variabile casuale e' uno scalare o un vettore di cui e' impossibile predire esattamente il valore che assumera' ma e' possibile dire con che probabilita' il valore apparterra' ad un dato insieme

La probabilita' che una variabile casuale $\bar{x} \in \mathbf{R}^d$ assuma un valore in un insieme $A \subseteq \mathbf{R}^d$ (*la probabilita' dell'evento* $\bar{x} \in A$) viene descritta tramite una funzione $\nu(\bar{\xi})$ (*densita' di probabilita'*) e calcolata come

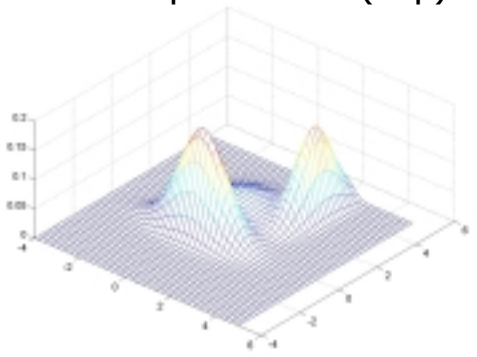
$$Prob(\bar{x} \in A) = \int_A \nu(\bar{\xi}) d\bar{\xi}$$

- $\nu(\bar{x}) \geq 0$
- L'evento certo $\bar{x} \in A = \mathbf{R}^d$ ha probabilita' pari ad uno $\Rightarrow \int_{\mathbf{R}^d} \nu(\bar{\xi}) d\bar{\xi} = 1$

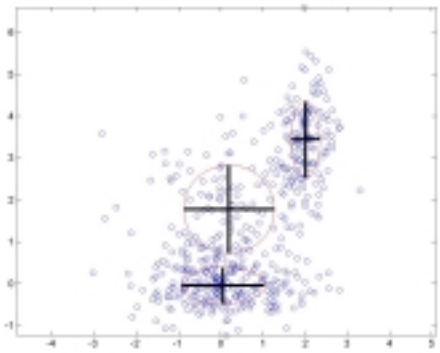


Breve introduzione alle variabili casuali

Densita' di probabilita' (ddp)



Estrazione di campioni (realizzazioni)



Media: $E[\bar{x}] = \int_{\mathbf{R}^d} \bar{\xi}\nu(\bar{\xi})d\bar{\xi}$

Media empirica: $E_{emp}[\bar{x}] = \frac{1}{n}\sum_{j=1}^{n} \hat{x}^j$

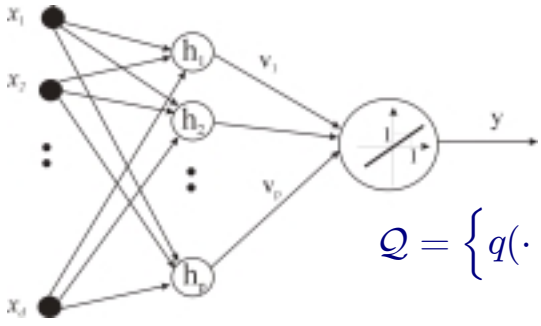
Ipotesi: - I dati $\{\hat{x}^j\}_{j=1}^n$ sono estratti secondo una ddp $\nu(\bar{x})$
- I campioni di rumore (regressione) $\{\epsilon^j\}_{j=1}^n$ sono estratti secondo una ddp $\nu_\epsilon(\bar{x})$
- L'estrazione di ciascun campione non e' influenzata dall'estrazione degli altri (indipendenza probabilistica)

Astrazione del problema di apprendimento

Caratterizziamo ora la rete neurale e l'algoritmo di addestramento

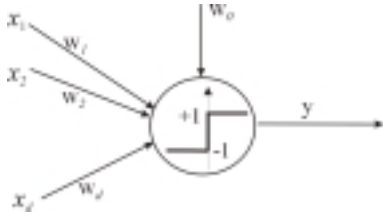
Una rete neurale implementa, al variare dei pesi una classe di funzioni $\mathcal{Q}$

Esempio: reti neurali RBF



$$\mathcal{Q} = \left\{ q(\cdot) : q(\bar{x}) = \sum_{i=1}^{p} v_i h(\|\bar{x} - \bar{\xi}_i\|), \forall \bar{v} \in \mathbf{R}^p \right\}$$

Esempio: percettrone a soglia

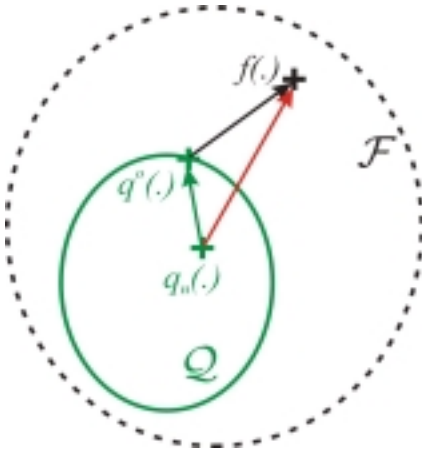


$$\bar{w}' = [\, w_0 \quad w_1 \quad \cdots \quad w_d \,]$$
$$\bar{x}' = [\, 1 \quad x_1 \quad \cdots \quad x_d \,]$$
$$\mathcal{Q} = \left\{ q(\cdot) : q(\bar{x}) = sign(\bar{w}'\bar{x}), \forall \bar{w} \in \mathbf{R}^{d+1} \right\}$$

Algoritmo di addestramento: la minimizzazione del rischio empirico

- Tramite l'algoritmo di addestramento si vorrebbe ricostruire la funzione $f(\cdot)$

  Pero' la classe di funzioni implementabili da una rete neurale e' limitata e usualmente $\mathcal{Q} \in \mathcal{F}$



- Allora si vorrebbe calcolare

  (supponiamo che esista...) $q^o(\bar{x}) = \operatorname{argmin}_{q \in \mathcal{Q}} R(q)$

  Rischio atteso: $R(q) = E_{\bar{x},\epsilon}[(y - q(\bar{x}))^2]$

- *Pero' le densita' di probabilita' non sono note a priori: si hanno solo i dati sperimentali !* Allora si minimizza il rischio empirico

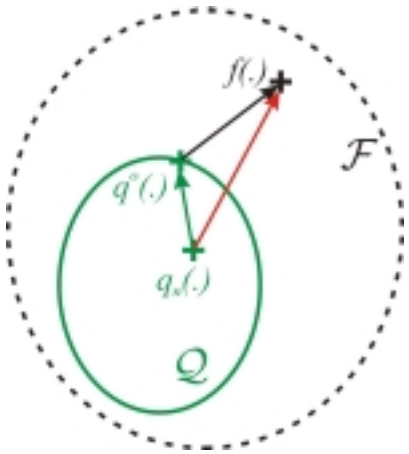
$$q_n(\bar{x}) = \operatorname{argmin}_{q \in \mathcal{Q}} R_{emp}(q)$$

Rischio empirico: $R_{emp}(q) = \frac{1}{n} \sum_{j=1}^{n} (\hat{y}^j - q(\hat{x}^j))^2$

Errore di generalizzazione

Definizione: l'errore di generalizzazione e' il rischio atteso nell'usare $q_n(\cdot)$ *come stima di* $f(\cdot)$

$$R(q_n) = E_{\bar{x},\epsilon}[(y - q_n(\bar{x}))^2]$$



- E' causato da due sorgenti di errore:

  - Errore di approssimazione

    $R(q^o) = E_{\bar{x},\epsilon}[(y - q^o(\bar{x}))^2]$

    Dipende criticamente da $\mathcal{Q}$ (cioe' dalla struttura delle rete neurale). Quanto piu' $\mathcal{Q}$ e' "grande", (quanti piu' neuroni sono usati) tanto piu' decresce.

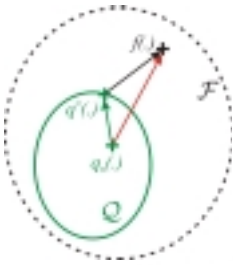
  - Errore di stima $R(q_n) - R(q^o)$

    Dipende criticamente dal numero dei dati di addestramento e dalla "grandezza" di $\mathcal{Q}$

Scomposizione: $R(q_n) = R(q^o) + (R(q_n) - R(q^o))$

Esempio: il modello della legge di Newton

$\nu(\bar{x})$: uniforme su [0,2], $\nu_\epsilon(\bar{x})$: gaussiana (varianza=.004), $\mathcal{F}$: funzioni lineari



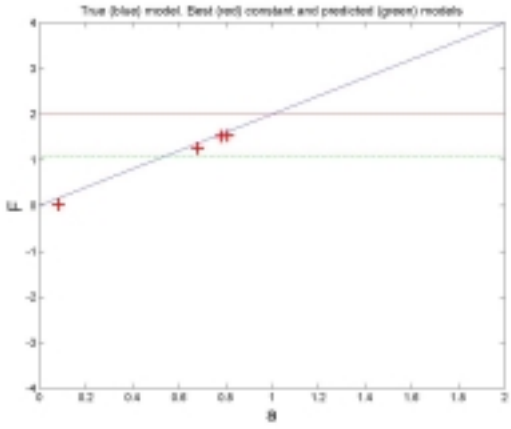
$\mathcal{Q}$: funzioni costanti

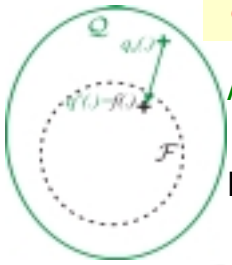
Basso errore di stima

$R(q_n) - R(q^o) = 0.7$

Alto errore di appross.

$R(q^o) = 1.37$





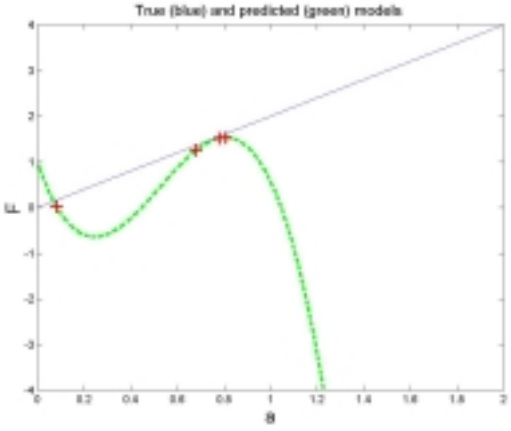
$\mathcal{Q}$: polinomi 3 ordine

Alto errore di stima

$R(q_n) - R(q^o) = 553$

Basso errore di appross.

$R(q^o) = 0$



Errore di approssimazione per reti neurali RBF e MLP

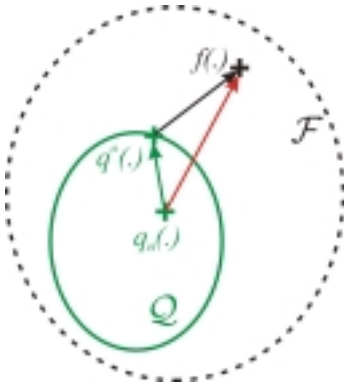
Consideriamo reti neurali MLP e RBF per problemi di regressione

Teorema (Cybenko): *per qualunque funzione in* $\mathcal{F} = C(\mathcal{X})$ *, ove* $\mathcal{X} \in \mathbf{R}^d$ *e' un insieme compatto, esiste una rete MLP che (per una opportuna scelta dei pesi) la approssima con precisione arbitraria*

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

- Concetto: se $\mathcal{F}$ e' una classe di funzioni *sufficientemente regolari* e se si usa un numero di neuroni *sufficientemente grande*, l'errore di approssimazione diminuisce arbitrariamente

- Esistono teoremi analoghi per reti neurali RBF
  - *teoremi di approssimazione universale*



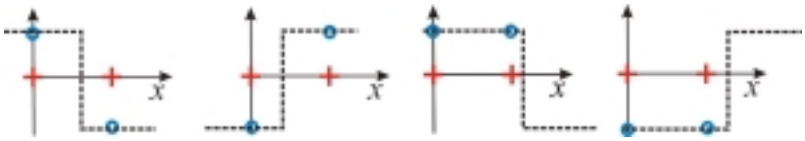
Errore di stima per problemi di classificazione: definizioni di base

Definizione: *Sia $\mathcal{Q}$ una classe di funzioni di dominio $A \in \mathbf{R}^d$ e a valori in $\{-1,1\}$ Un insieme di punti $\{\bar{x}^j\}_{j=1}^t$ nel dominio A e' completamente suddiviso dalla classe $\mathcal{Q}$ se per qualunque vettore $b \in \{-1,1\}^t$ esiste una funzione $q \in \mathcal{Q}$ tale che*

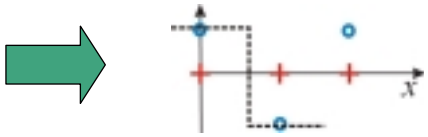
$$[\, q(\bar{x}^1) \quad \cdots \quad q(\bar{x}^t) \,] = b$$

Esempio: $\mathcal{Q} = \{\alpha \cdot sign(x - \beta), \alpha \in \{-1, 1\}, \beta \in \mathbf{R}\}$

- L'insieme di punti $\{x^1 = 0, x^2 = 0.5\}$ e' completamente suddiviso da $\mathcal{Q}$



- L'insieme di punti $\{x^1 = 0, x^2 = 0.5, x^3 = 1\}$ non e' completamente suddiviso da $\mathcal{Q}$



Errore di stima per problemi di classificazione: la VC-dimension

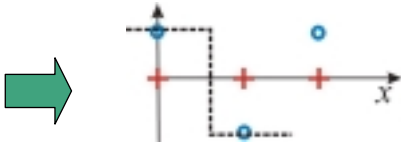
Definizione: *la dimensione di Vapnik e Chervonenkis di $\mathcal{Q}$ ($VC(\mathcal{Q})$) e' il piu' grande intero d tale per cui esiste un insieme di punti $\{\bar{x}^j\}_{j=1}^d$ completamente suddiviso da $\mathcal{Q}$. Se tale intero non esiste, $VC(\mathcal{Q}) = +\infty$.*

- Se $VC(\mathcal{Q}) = d$, nessun insieme di $d+1$ punti puo' essere completamente suddiviso da $\mathcal{Q}$

La dimensione di Vapnik e Chervonenkis e' una misura della complessita' della classe di funzioni $\mathcal{Q}$

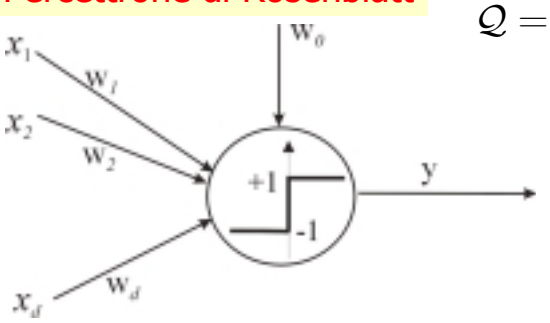
Esempio: $\mathcal{Q} = \{\alpha \cdot sign(x - \beta), \alpha \in \{-1, 1\}, \beta \in \mathbf{R}\}$

- Si e' trovato un insieme di 2 punti completamente suddiviso da $\mathcal{Q}$
- E' facile vedere che nessun insieme di 3 punti puo' essere completamente suddiviso da $\mathcal{Q}$
- Deduciamo che $VC(\mathcal{Q}) = 2$



VC-dimension: esempi

Percettrone di Rosenblatt



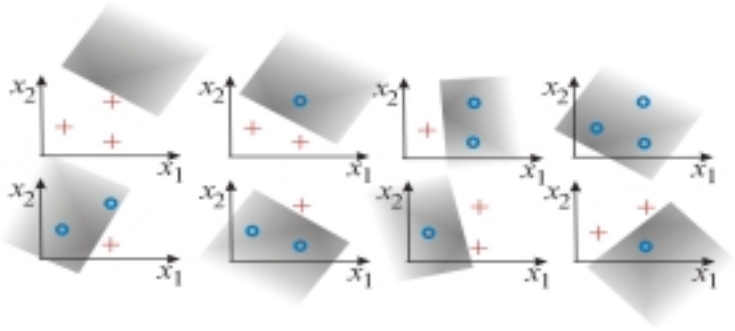
$$\mathcal{Q} = \left\{ q(\cdot) : q(\bar{x}) = sign(\bar{w}'\bar{x}), \forall \bar{w} \in \mathbf{R}^{d+1} \right\}$$

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

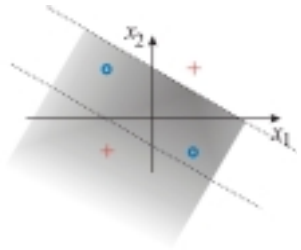
$$\bar{x}' = [\, 1 \quad x_1 \quad \cdots \quad x_d \,]$$

Teorema: $VC(\mathcal{Q}) = d + 1$

Esempio: $d = 2$

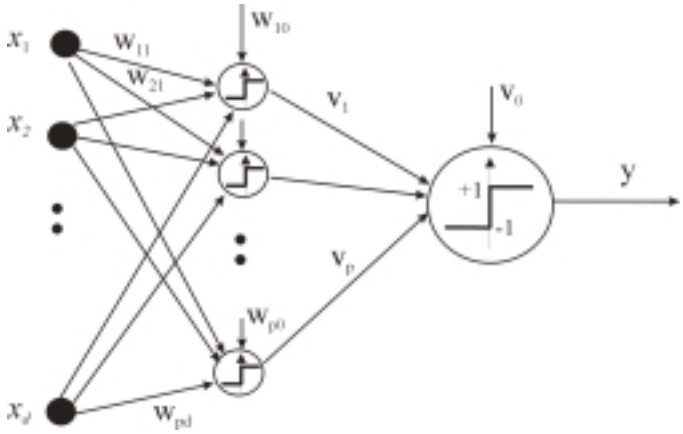


Questo insieme di 4 punti non puo' essere descritto dal percettrone ! (XOR)



VC-dimension: esempi

Rete neurale MLP con funzioni di attivazione $sign(\cdot)$



Teorema: *sia W il numero totale dei pesi (inclusi i bias) in una rete neurale MLP con funzioni di attivazione $sign(\cdot)$ e un numero di neuroni $N > 2$. Sia $\mathcal{Q}$ la classe di funzioni implementate dalla rete. Allora*

$$VC(\mathcal{Q}) \leq 2W \ln(eN)$$

Errore di stima: problemi di classificazione

Problema: si puo' ridurre l'errore di stima minimizzando il rischio empirico ?

- Se il rischio empirico ed il rischio atteso fossero simili per $\forall q \in \mathcal{Q}$ simultaneamente, minimizzando l'uno o l'altro non si otterrebbero risultati molto differenti.
  - Ci si chiede se, data una precisione $\eta > 0$, vale la disuguaglianza

$$\sup_{q \in \mathcal{Q}} |R(q) - R_{emp}(q)| > \eta$$

- Il rischio empirico dipende dagli specifici dati sperimentali

$$R_{emp}(q) = \frac{1}{n} \sum_{j=1}^{n} (\hat{y}^j - q(\hat{x}^j))^2$$

  - Per evitare questa dipendenza, si modellizzano i dati sperimentali come variabili casuali e si vorrebbe calcolare

$$Prob(\sup_{q \in \mathcal{Q}} |R(q) - R_{emp}(q)| > \eta)$$

Errore di stima: problemi di classificazione

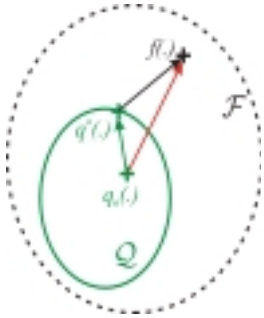
Teorema: *Qualunque sia la densita' di probabilita' delle variabili casuali* $\{\bar{x}^j\}_{j=1}^{n}$, *data una precisione* $\eta > 0$ *vale la seguente disuguaglianza*

$$Prob(\sup_{q \in \mathcal{Q}} |R(q) - R_{emp}(q)| > \eta) \leq 6 \exp\left(VC(\mathcal{Q})(1 + ln(\frac{n}{VC(\mathcal{Q})})\right) \exp(\frac{-n\eta^2}{8})$$

ove $VC(\mathcal{Q})$ *denota la dimensione di Vapnik e Chervonenkis della classe di funzioni approssimanti* $\mathcal{Q}$ *e si assume* $VC(\mathcal{Q}) < +\infty$ *e* $n > VC(\mathcal{Q})$

- Il teorema fornisce una maggiorazione alla probabilita' di ottenere un insieme di dati per cui $\sup_{q \in \mathcal{Q}} |R(q) - R_{emp}(q)| > \eta$
- Il teorema e' significativo solo se il maggiorante e' minore di uno ...
- Se $VC(\mathcal{Q}) < +\infty$, la maggiorazione garantisce che

$$\lim_{n \to +\infty} Prob(|\inf_{q \in \mathcal{Q}} R(q) - R(q_n)| > \eta) = 0$$



Intuitivamente, se $VC(\mathcal{Q}) < +\infty$ *l'errore di stima tende a zero al crescere del numero dei dati*

Relazione col numero di parametri e di neuroni

$$Prob(\sup_{q \in \mathcal{Q}} |R(q) - R_{emp}(q)| > \eta) \leq 6 \exp\left(VC(\mathcal{Q})(1 + ln(\tfrac{n}{VC(\mathcal{Q})})\right) \exp(\tfrac{-n\eta^2}{8})$$

Percettrone di Rosenblatt: $VC(\mathcal{Q}) = d + 1$

Rete neurale MLP con funzioni di attivazione $sign(\cdot)$*:* $VC(\mathcal{Q}) \leq 2W \ln(eN)$

Per un numero di dati fissato, il maggiorante cresce al crescere del numero di neuroni e di pesi della rete.

Intuitivamente: piu' la rete neurale e' flessibile, piu' e' difficile contenere l'errore di stima, a parita' del numero di dati !

In pratica: usare un numero di neuroni/pesi molto inferiore al numero dei dati a disposizione per garantire un basso errore di stima

Bilanciamento tra errore di stima e di approssimazione

Lo scopo ultimo non e' diminuire l'errore di stima ma quello di generalizzazione per un numero di dati fissato

$$R(q_n) = \underbrace{R(q^o)}_{} + \underbrace{(R(q_n) - R(q^o))}_{}$$

Errore di approssimazione:

in generale decresce al crescere di $VC(\mathcal{Q})$

Errore di stima:

in generale cresce al crescere di $VC(\mathcal{Q})$

La topologia ottimale della rete neurale dipende dal bilanciamento di questi due fattori !

Estensioni

- La dimensione di Vapnik e Chervonenkis si generalizza a classi di funzioni a valori reali (MLP con funzioni di attivazione sigmoidali, RBF ...)

- Anche per problemi di regressione si ottengono maggiorazioni del tipo

$$Prob(\sup_{q \in \mathcal{Q}} |R(q) - R_{emp}(q)| > \eta) \leq \omega(\eta, n, W, N)$$

che suggeriscono sempre di limitare la complessita' della rete rispetto al numero dei dati in modo da contenere l'errore di stima.

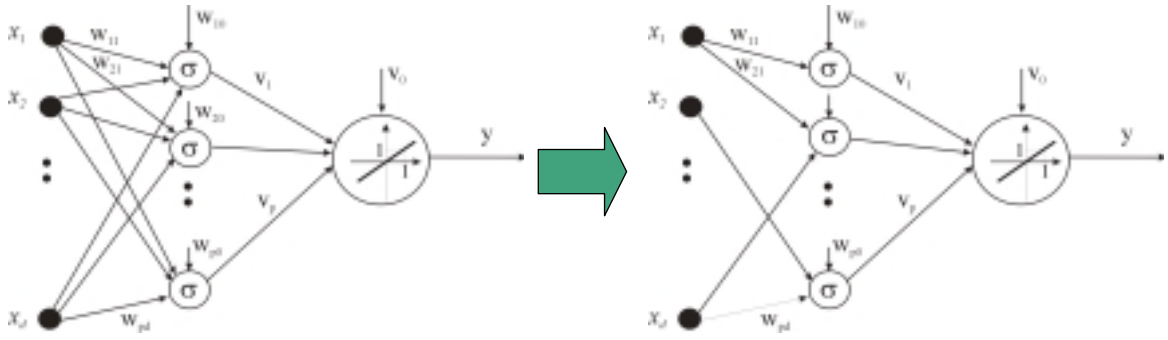
Difficolta':

- I maggioranti $\omega(\eta, n, W, N)$ non sono stretti $\Rightarrow$ in pratica non si usano per determinare il numero di neuroni e di pesi ottimo ...

Necessita' di regole empiriche per la taratura ottimale della rete neurale

Regole empiriche per la taratura di reti neurali MLP

Tecniche di Pruning: Addestrare una rete MLP completamente connessa, tagliare le connessioni "meno rilevanti" e riaddestrare la nuova rete.



- Tolgo le connessioni associate a pesi "piccoli"
  - puo' essere una scelta sbagliata se l'uscita della rete e' molto sensitiva a questi pesi ...

Optimal Brain Damage: calcolare la sensitivita' di ogni peso $S_{ij} = \partial^2 R_{emp} / \partial w_{ij}^2$ e tagliare i rami associati a pesi con piccola rilevanza $r_{ij} = S_{i,j} w_{ij}^2$

Regole empiriche per la taratura di reti neurali MLP

Weight Decay: per addestrare la rete minimizzo

$$R(\bar{w}_1, \ldots, \bar{w}_p, \bar{v}) = \frac{1}{2n}\sum_{j=1}^{n}(\hat{y}^j - y^j)^2 + \lambda \sum_{i=1}^{p}\sum_{j=1}^{d} w_{ij}^2 \qquad \lambda > 0$$

Idea: *compromesso tra minimizzazione del rischio empirico e principio di parsimonia nell'uso dei pesi.*

- il secondo addendo forza l'uso di pesi piccoli, se possibile. Piu' λ e' grande piu' la rete MLP dovrebbe essere "semplice"...

Problema: λ regola il bilanciamento tra i due termini: come sceglierlo ?

Cross-validazione !

Esistono numerose altre regole empiriche per la taratura di reti neurali...

Conclusioni

- Le reti neurali sono metodi (molto diffusi) per risolvere problemi di modellistica black-box
  - inizialmente sono state ispirate dal funzionamento di agglomerati di neuroni
  - oggigiorno si usa il termine "rete neurale" anche per denotare metodi che non hanno alcuna interpretazione biologica !
- Oltre alle reti neurali MLP o RBF ne esistono numerose altre
  - reti di Hopfield
  - reti neurali di regolarizzazione
  - self-organizing feature maps
  - support vector machines
  - ....
- Il problema di progettare reti neurali "ottime" addestrate su un numero di esempi finito e' lontano dall'essere completamente compreso e/o risolto !

Bibliografia sulle reti neurali

Libri:

- Simon Haykin, *Neural Networks, a comprehensive foundation*. Macmillan, 1994
- C.M. Bishop, *Neural Networks for pattern recognition*. Oxford University Press, 1995
- V.N. Vapnik, *The nature of statistical learning theory*. Springer Verlag, 1995

Articoli introduttivi:

- D.R. Hush and B.G. Horne, *Progress in supervised Neural Networks*. IEEE Signal Processing Magazine, pp. 8-39, January 1993
- T. Poggio and F. Girosi, *Networks for approximation and learning*. IEEE Proceedings, Vol. 78, pp. 1481-1497, 1990