

Intelligenza Artificiale

Introduzione alle Reti Neurali

Roberto Marmo

Introduzione alle Reti Neurali

Parte 1. Il cervello umano

Parte 2. La rete neurale

Parte 3. L'apprendimento della rete

Parte 4. Creare una rete per classificare

Parte 5. Esempio di applicazione

Testi consigliati

- Bishop, *Neural networks for pattern recognition*, Clarendon Press, Oxford, 1996
- Domeniconi, Jordan, *Discorsi sulle reti neurali e l'apprendimento*, Franco Angeli, 2001
- Cammarata, *Reti neuronali*, Etas
- Floreano, *Manuale sulle reti neurali*, Il Mulino
- *L'apprendimento delle reti artificiali di neuroni*. Le Scienze n.291, novembre 1992
- De Luca, Caianiello, *Introduzione alla cibernetica*, Franco Angeli
- <ftp://ftp.sas.com/pub/neural/FAQ.html> nozioni e link su Internet

Parte 1

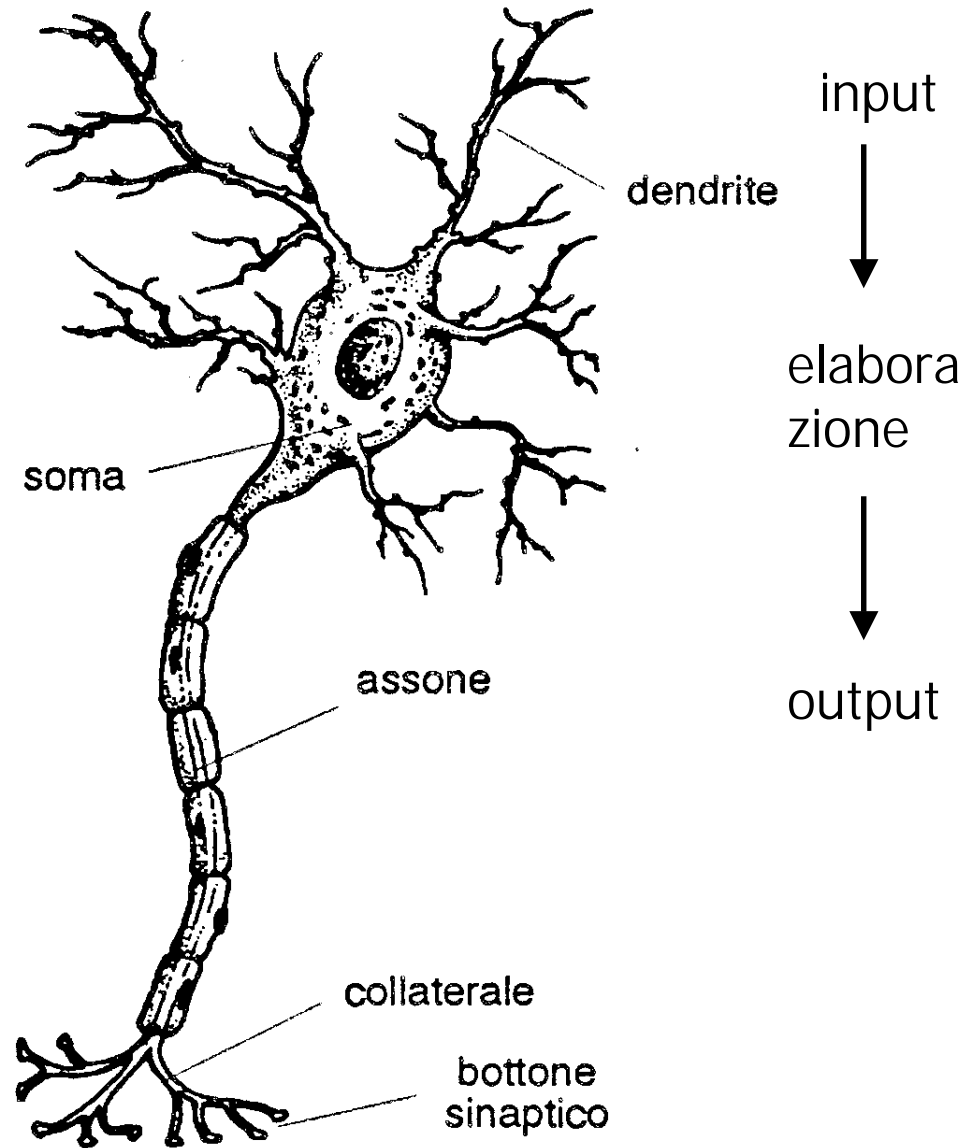
Il cervello umano

John von Neuman: non c'è alcun modello del cervello più semplice del cervello stesso.

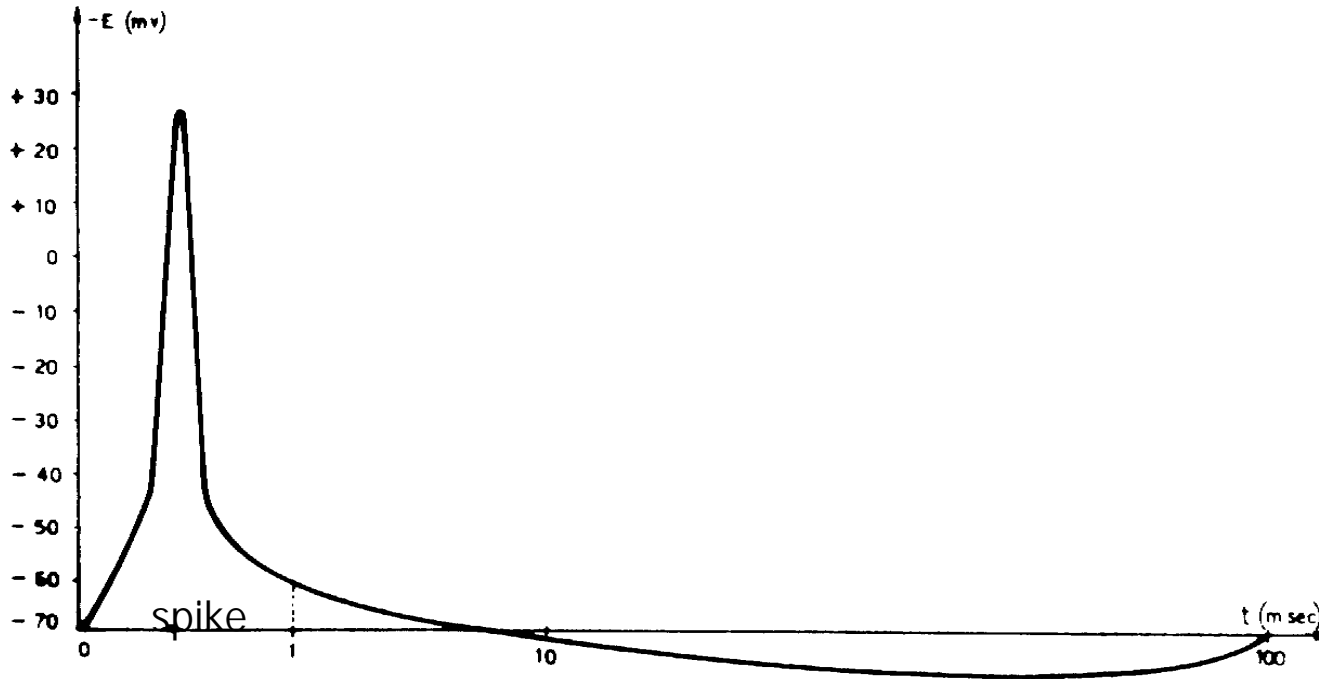
Per simulare attraverso un insieme di regole di calcolo alcuni comportamenti della struttura cerebrale degli esseri viventi, occorre conoscere la struttura del cervello umano e cercare di riprodurlo con un modello matematico.

La cellula neuronale

Unità fondamentale del cervello umano. Ogni neurone riceve come input i segnali elettrici da tutti i dendriti, e se la somma pesata supera il valore di attivazione emette un impulso elettrico in uscita verso l'assone.



Funzione di sparo del neurone

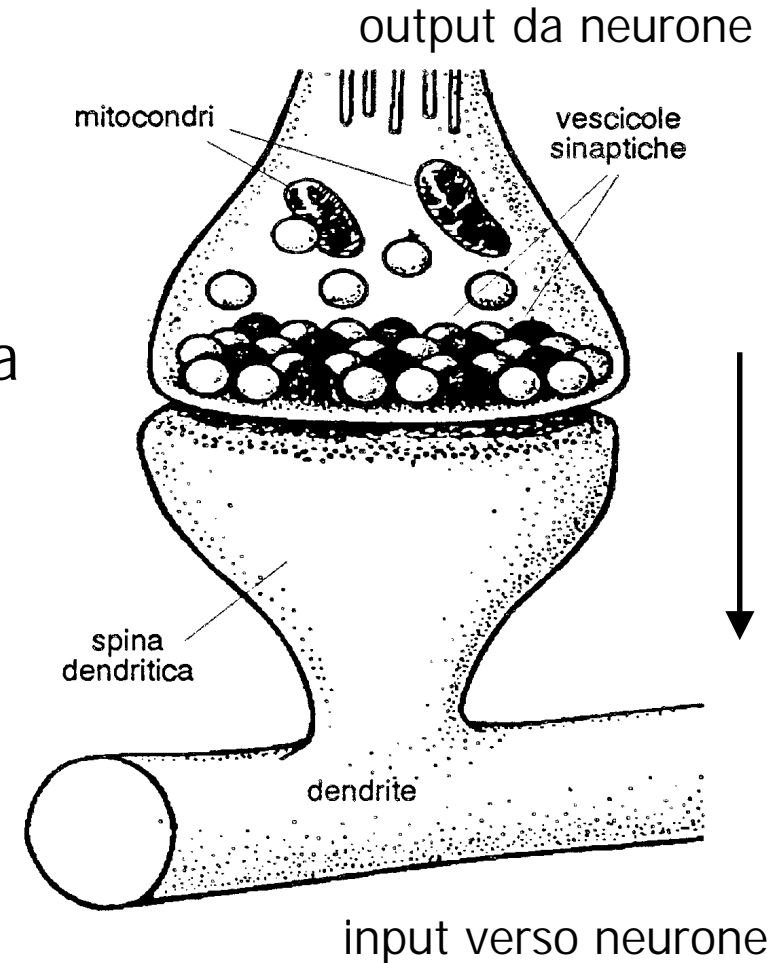


Andamento di $-E$, differenza di potenziale fra interno ed esterno della membrana cellulare del neurone.

Per uno stimolo con adeguata intensità, il neurone risponde con uno spike o non risponde; non c'è nessun tipo di risposta intermedia.

Contatto sinaptico

Punto di contatto tra due neuroni. Le sinapsi aumentano o diminuiscono nel tempo. La parte superiore trasforma il segnale elettrico in sostanza chimica (neurotrasmettitore) che passa dall'altra parte e viene riconvertito in segnale elettrico. La sinapsi può fare azione eccitatoria o inibitoria del collegamento tra i due neuroni variando i neurotrasmettitori.



Il cervello umano

- esistono vari tipi di neuroni
- 10 miliardi di neuroni
- 80000 neuroni per mm^2
- ogni neurone interagisce con 1000-10000 neuroni
- sinapsi tra neuroni anche molto distanti
- l'elaborazione dell'informazione in parallelo e diffusa tra migliaia di neuroni porta all'emergere di processi cognitivi
- l'elaborazione risiede in intere regioni ben localizzate
- struttura cerebrale sempre in evoluzione

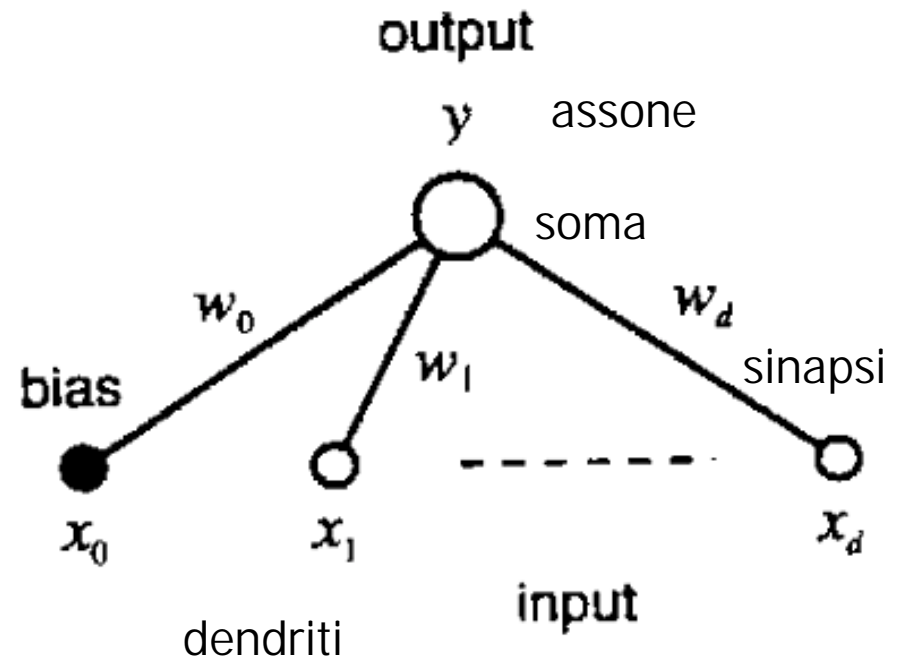
Parte 2

La rete neurale

Il neurone artificiale

Modello matematico molto semplificato del neurone biologico.

Ad ogni input x_i è associato un peso w_i con valore positivo o negativo per eccitare o inibire il neurone. Il bias varia secondo la propensione del neurone ad attivarsi, per variare la soglia di attivazione del neurone



Algoritmo del neurone

1. Caricare i valori degli input x_i e dei pesi relativi w_i
2. Calcolare la somma dei valori input pesata con i relativi pesi
3. Calcolare il valore della funzione di attivazione g con il risultato della somma pesata
4. L'output del neurone y è il risultato della funzione di attivazione

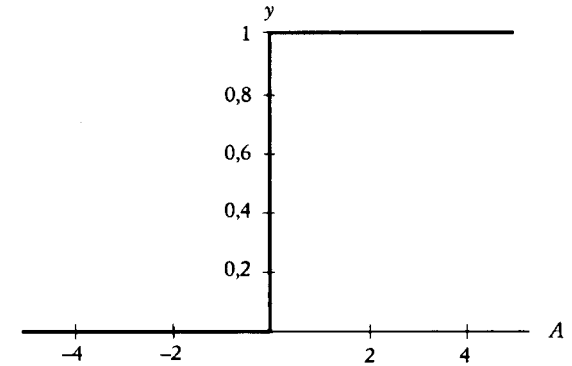
$$y(x) = g\left(\sum_{i=1}^d w_i x_i + w_0\right) = \overline{w}^T \overline{x}$$

Funzioni di attivazione

Determina la risposta del neurone.

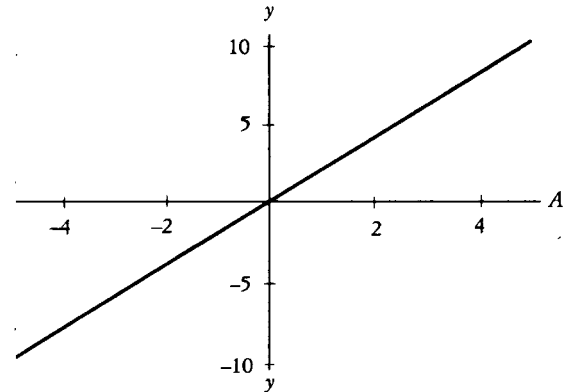
A gradino

$$g(A) = \begin{cases} 1 & \text{se } A > 0 \\ 0 & \text{altrimenti} \end{cases}$$



Lineare continua

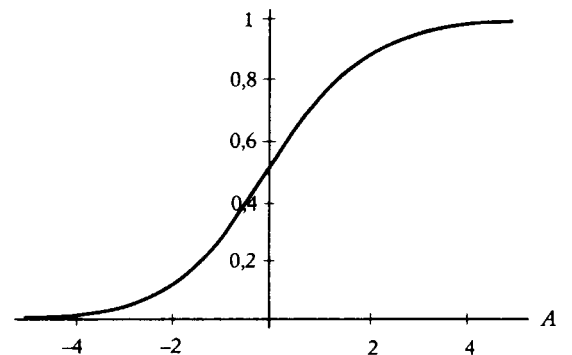
$$g(A) = kA$$



Sigmoide o logistica

$$g(A) = \frac{1}{1 + e^{-A}}$$

valori positivi, continua e derivabile



La rete neurale

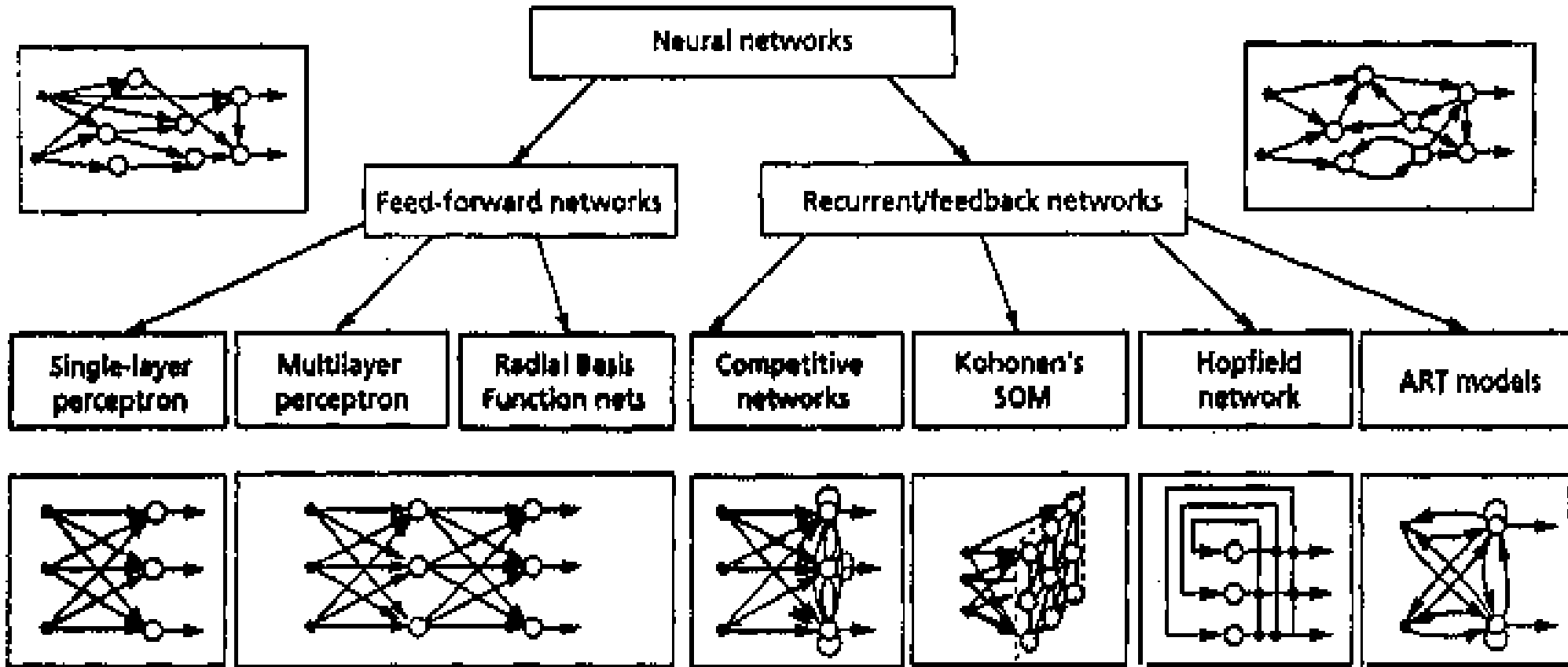
Sistema dinamico avente la topologia di un grafo orientato con nodi, i neuroni artificiali, ed archi, i pesi sinaptici. Il termine rete è riferito alla topologia dei collegamenti tra i neuroni.

Altre definizioni:

- scatola nera, di cui si può ignorare il funzionamento, che associa un input a un output
- modello matematico che calcola la funzione $\text{output} = f(\text{input}, \text{pesi})$ al variare dei pesi e senza specificare la forma della funzione f

In Bishop un approccio statistico e dimostrazioni matematiche.

Esempi di reti neurali



Vantaggi

- adatte per problemi che non chiedono risposte accurate, ma risposte approssimate con un grado di errore o di variazione
- generalizzazione: producono buone risposte anche con input non considerati durante la creazione e l'addestramento
- facili da implementare, basta definire il neurone e poi crearne delle copie e creare i collegamenti tra i neuroni
- funzionamento veloce perché parallelo; ogni neurone usa solo il suo input
- stabilità dell'output rispetto a valori di input: incompleti, con rumore, non ben noti, che accettano un grado di errore o di variazione
- determinano il risultato tenendo conto contemporaneamente di tutti gli input

Svantaggi

- incapacità di rendere conto dell'elaborazione: non si può capire perché ha dato quel risultato specifico
- non si può descrivere e localizzare la conoscenza memorizzata nella rete
- carenza di hardware con cui implementare, si usano su computer seriali
- tecniche di addestramento sofisticate che richiedono molto tempo di calcolo
- non sempre esiste una rete che risolve il problema, perché non sempre esiste un algoritmo di apprendimento che converge dando un output della rete con basso errore
- i valori di output non sono precisi, ma hanno un margine in cui possono variare
- serve una casistica di esempi molto ampia per ottenere un buon apprendimento e un basso errore di output

Campi di applicazione

1. Pattern classificazione
2. Clustering
3. Approssimazione di funzioni
4. Predizioni in serie temporali
5. Ottimizzazione
6. Memorie associative
7. Controllo di apparati
8. Elaborazione di segnali ed immagini

Rapporti con algoritmi genetici

Gli algoritmi genetici possono essere applicati alle reti neurali per risolvere problemi come:

- scegliere la struttura della rete
- scegliere i valori dei pesi per ridurre l'errore di output

Rapporti con logica fuzzy

Le reti neurali non calcolano con le variabili linguistiche e non usano regole qualitative. Esistono le reti neuro-fuzzy, strumento matematico molto efficiente e complesso che unisce i vantaggi delle due tecniche.

Rapporti con i sistemi esperti

Le reti neurali si differenziano molto dai sistemi esperti perché:

- non usano conoscenze esplicite, ma conoscenze implicite contenute in una casistica molto ampia di esempi formata da vettori di numeri
- non usano regole tipo IF...THEN..
- non vengono programmate, vengono addestrate con l'apprendimento
- accettano dati parziali e con rumore
- il sistema esperto può spiegare perché e come ha ottenuto una conclusione, le reti neurali no

Rapporti con I.A. classica

- Le reti neurali non ragionano con simboli ma con connessioni tra neuroni, per cui L'I.A. classica può spiegare come ha ottenuto un risultato, le reti neurali no
- L'I.A. può fare ragionamenti come "Socrate è uomo, uomini sono mortali, Socrate è mortale", la rete neurale no ma può facilmente riconoscere una faccia.
- L'I.A. classica non tiene conto del supporto materiale che ragiona, ma si dedica solo alla funzione del ragionamento e cerca di imitare la mente umana. Il connessionismo è l'approccio allo studio della mente con le reti neurali; considera fondamentale la struttura del supporto materiale che ragiona, per cui cerca di imitare il cervello umano

Parte 3

L'apprendimento della rete

Non programmare ma apprendere

La programmazione serve solo per creare la rete e l'algoritmo di apprendimento: per insegnare alla rete a risolvere un problema, occorre un periodo di apprendimento in cui insegnare alla rete come comportarsi con l'input che riceve, perché all'inizio la rete non ha nessuna forma di conoscenza.

La conoscenza è memorizzata sui pesi e la rete apprende usando tecniche di ottimizzazione per variare i valori dei pesi. Tipi di apprendimento:

- supervisionato
- non supervisionato

I modelli di rete neurale si differenziano anche per il tipo di apprendimento: ogni modello ha il suo tipo e non può apprendere con l'altro

Apprendimento supervisionato

Alla rete viene presentato un training set preparato da un supervisore esterno, e composto da molte coppie significative di valori (input, output atteso);

- con l'input la rete calcola il suo output
- l'errore è dato dalla differenza tra il suo output e l'output atteso; serve a supervisionare l'apprendimento per far capire alla rete quanto sbaglia nel calcolare su quell'input
- la rete modifica i pesi in base all'errore cercando di minimizzarlo e commetterà sempre meno errori.

Occorre quindi preparare alcuni esempi di funzionamento studiati appositamente, e la rete impara da questi esempi.

Modelli che usano questo apprendimento:

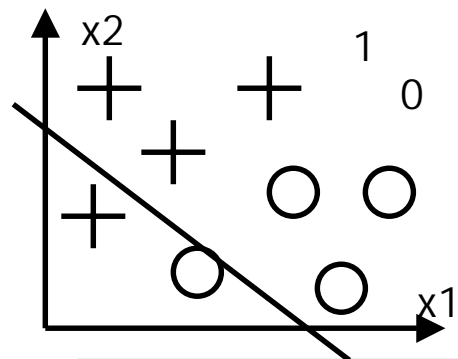
- perceptron
- multi layer perceptron
- radial basis function

Il perceptron

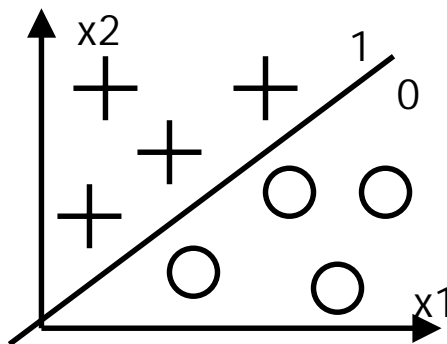
Modello semplice creato da Rosenblatt nel 1957 per riconoscere immagini. Composto da un neurone con funzione di attivazione a gradino ed uscita con valori binari: 1 indica la presenza di un oggetto, 0 l'assenza. L'apprendimento consiste nel variare i pesi e nel muovere la retta di separazione fino a dividere i valori input correttamente in due gruppi separati.

Spazio di 2 valori input linearmente separabile e perceptron trova retta di separazione tra i due gruppi di oggetti:

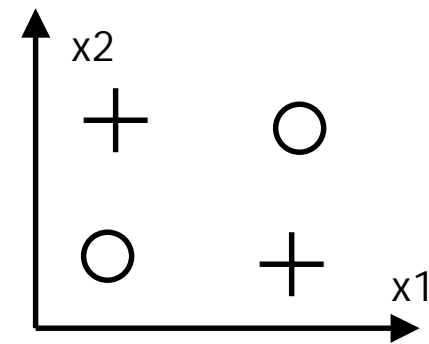
prima di apprendimento



dopo

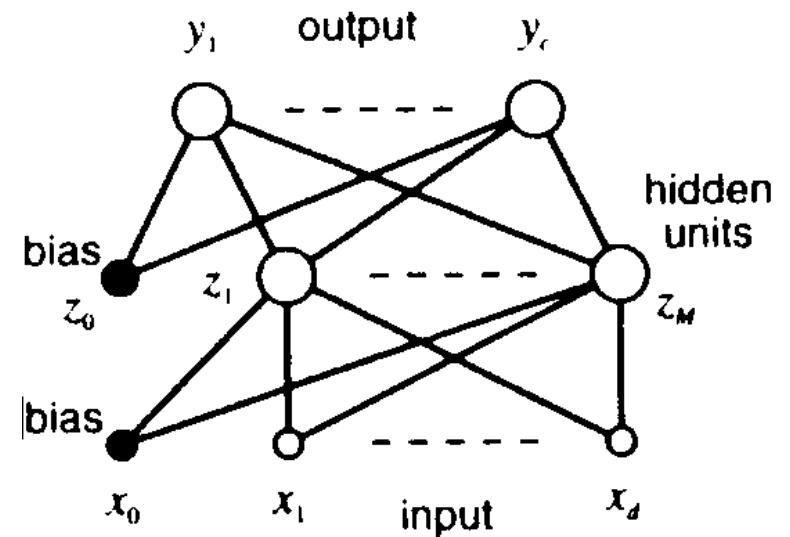
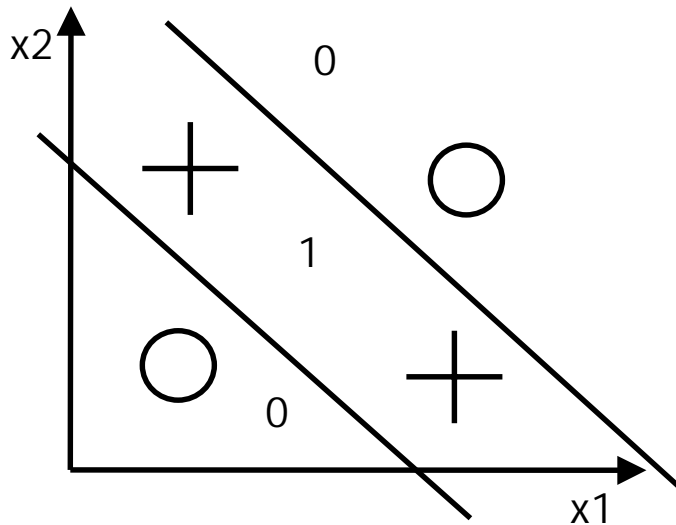


Spazio di 2 valori input non linearmente separabile e perceptron non trova una retta di separazione



Multi Layer Perceptron (Mlp)

Supera i limiti del perceptron usando strati di neuroni hidden che realizzano una rappresentazione interna dell'input più complessa, perché individua regioni arbitrarie intersecando iperpiani nello iperspazio dei valori input.



Mlp con d neuroni input, M neuroni hidden, c neuroni output, g funzione di attivazione dei neuroni, w_{kj} il peso dal neurone k al neurone j

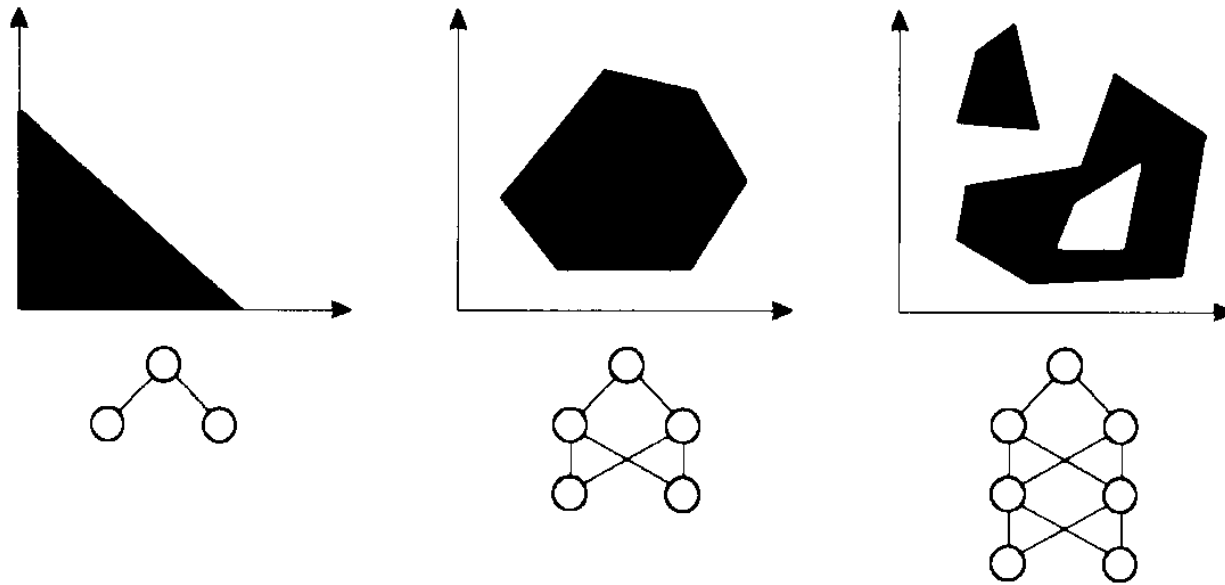
$$y_k(x) = g \left(\sum_{j=0}^M w_{kj}^{(2)} g \left(\sum_{i=0}^c w_{ji}^{(1)} x_i \right) \right)$$

Struttura di Mlp

- il livello di input non contiene neuroni, per cui i livelli si contano dal successivo e nel precedente caso sono 2
- ogni neurone è collegato con tutti i neuroni dello strato precedente e successivo, i neuroni sullo stesso strato non sono collegati; tutti i neuroni hanno la stessa funzione di attivazione
- la conoscenza viene elaborata dal livello input verso il livello output e non ci sono cicli
- la funzione di attivazione softmax crea ogni output in $[0,1]$ e somma di output pari a 1, in modo da interpretare la risposta della rete come stime di probabilità

$$g(a_i) = \frac{e^{a_i}}{\sum_{i=1}^n e^{a_i}}$$

Regioni delimitate

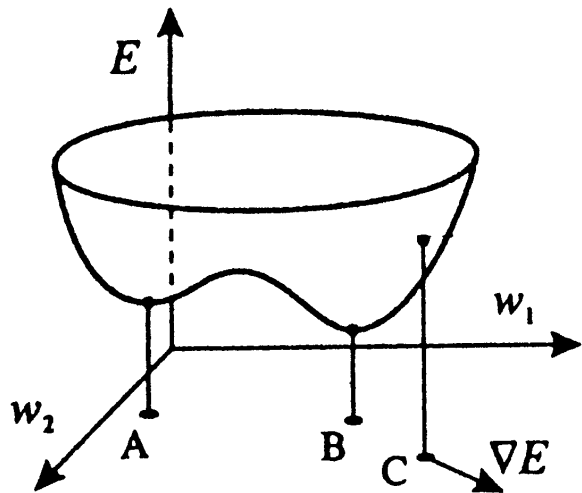


A sinistra la regione delimitata da un Perceptron, al centro la regione delimitata da Mlp a 1 livello hidden, a destra la regione delimitata da Mlp con 2 livelli hidden; con tale modello si delimitano aree arbitrarie.

Funzione di errore

Esprime la differenza fra l'output della rete y e l'output desiderato y' nell'apprendimento. $E(w)$ perché l'output è ottenuto in base al valore dei pesi; quindi E è derivabile nei pesi w e occorre trovare il vettore w che rende minimo l'errore. Esistono molte formulazioni di E .

$$E(w) = \sum_{i=1}^c ((y_i - y'_i)^2)$$



E è non lineare: gli algoritmi cercano un minimo nella sua superficie con modifiche di w in base al gradiente di E . A sinistra c'è una superficie di E quadratica nei pesi w_1 e w_2 ; A e B sono minimi e C punto di calcolo del gradiente locale della superficie, si va nella direzione opposta al gradiente. Può non convergere verso il minimo assoluto B ma verso un minimo locale A; in alcuni punti il gradiente è nullo.

Algoritmo back propagation

Si crea un training set composto da coppie (input, output) significative e rappresentative e si sceglie una funzione di errore E . Esegue un'epoca di apprendimento attuando per tutte le coppie i seguenti passi:

1. prende una coppia e calcola la risposta della rete per quell'input; il calcolo procede dal livello input verso il livello output, quindi propaga in avanti l'errore
2. calcola l'errore E tra l'output della rete e output della coppia
3. propaga all'indietro l'errore verso il livello di input, aggiungendo ad ogni peso un incremento positivo o negativo determinato dalla sua influenza nella formazione dell'errore
4. ripete dal passo 1 fino a terminare le coppie, poi va al passo 4
5. calcola l'errore globale dato dalla somma degli errori su tutte le coppie; se l'errore globale è ancora alto, si ripete l'epoca di apprendimento.

Algoritmo lento che può finire intrappolato in un minimo di E credendo di aver trovato il valore ottimo dei pesi che rende minima la E . Esistono tecniche che migliorano la convergenza verso il minimo assoluto di E .

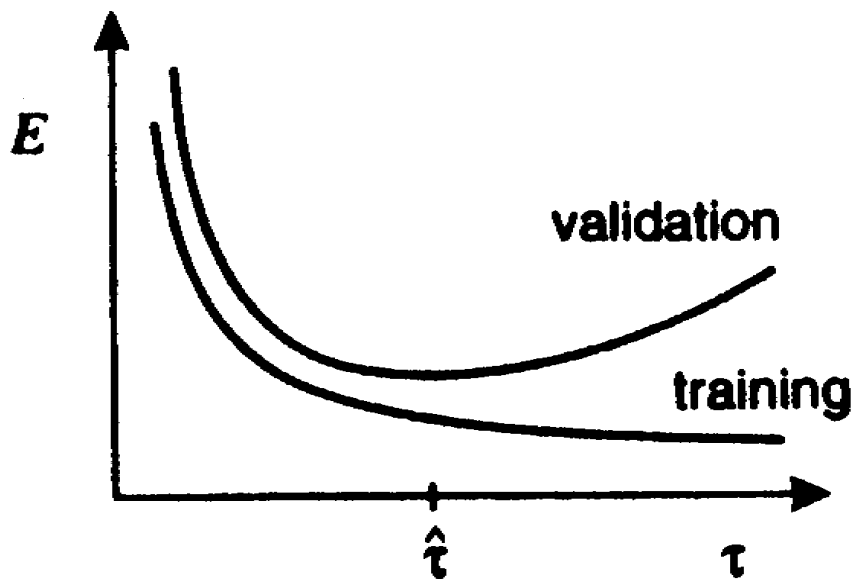
Problemi del back propagation

- algoritmo lento che può finire intrappolato in un minimo di E credendo di aver trovato il valore ottimo dei pesi che rende minima la E .
- dipendenza dal valore iniziale dei pesi w ; può capitare di cominciare da un punto della superficie di errore già molto vicino a un minimo locale

Esistono tecniche che migliorano la convergenza verso il minimo assoluto di E o che permettono di uscire da un minimo locale.

Overfitting

La rete neurale deve avere capacità di comprensione del modello statistico dei dati, non memorizzare i soli dati del training set. Solo così può generalizzare, cioè rispondere esattamente a input non in training set. La tecnica migliore per evitare ciò è l'early stopping. Nell'apprendimento l'errore E tende a 0 perché la rete sta imparando meglio. Usando un validation set di coppie non usate nel training set,



si misura la E e si crea la seconda curva che tende a un minimo e poi ricresce: da quel punto la rete sta imparando il training set e non il suo modello statistico. Si ferma lo apprendimento al minimo $\hat{\tau}$ di E rispetto al validation set.

Valutare prestazioni

Per misurare le prestazioni di una rete neurale dopo l'apprendimento, si crea il test set formato da coppie non usate per i training e validation set. In genere il test set è un terzo del training set ed è composto da input critici su cui la risposta della rete deve essere buona, altrimenti si butta via la rete.

Per ogni coppia del test set:

- calcolare la risposta della rete all'input
- calcolare l'errore dato dalla differenza tra output rete e output di coppia

L'errore totale è dato dalla somma degli errori avuti per ogni coppia.

Secondo il tipo di problema, si usano varie tecniche statistiche per decidere se usare o meno la rete creata; in genere si accetta una rete se sul test set ha mostrato un errore inferiore al 20-25% delle coppie.

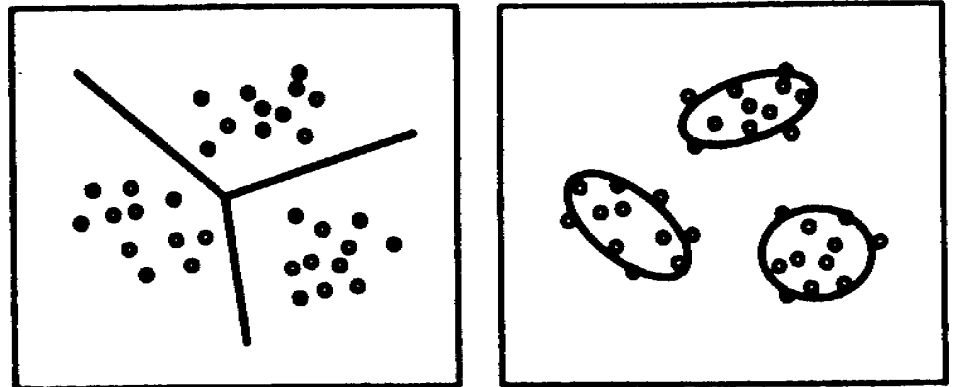
Radial basis function (Rbf)

Un solo livello hidden di M neuroni che realizza funzioni di base $\phi_j(x)$ circolari centrate sui punti dello spazio di input; esistono varie forme delle funzioni, in genere si usano gaussiane. Ideale per interpolazioni di funzioni e di predizione in serie temporali, in cui stimare il valore successivo in una serie di numeri.

$$y_k(x) = \sum_{i=1}^M w_{kj} \phi_j(x) + w_{k0}$$

Equazione del neurone output k su input x , w_{kj} è il peso tra neuroni k e j

Divisione dello iperspazio dei valori input: a sinistra Mlp con semipiani, a destra Rbf con cluster



Apprendimento non supervisionato

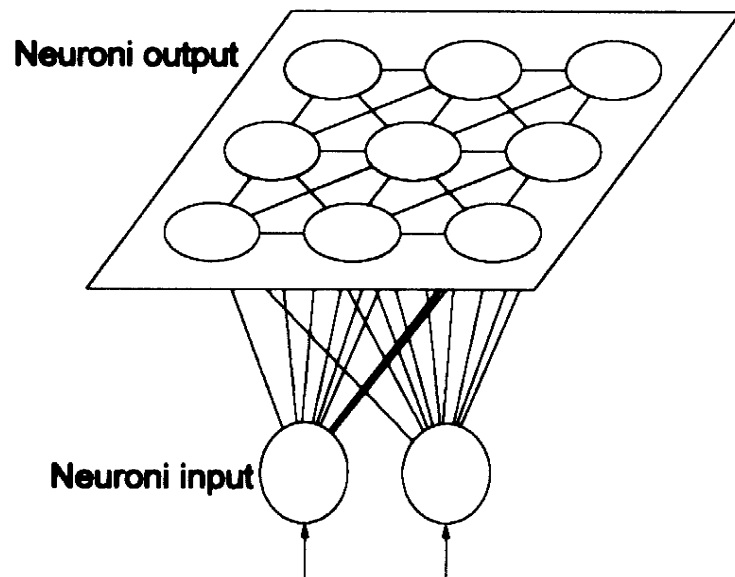
Alla rete vengono presentati solo i valori di input e la rete li divide autonomamente in gruppi usando misure di similarità, senza usare confronti con output noti, e cercando di mettere input simili nello stesso gruppo. E' un apprendimento autonomo e non c'è controllo esterno sull'errore. Adatto per ottimizzare risorse e se non si conoscono a priori i gruppi in cui dividere gli input.

Modelli che usano questo apprendimento:

- Kohonen
- Hopfield

Self Organizing Maps SOM

Ideate da Tuevo Kohonen nel 1982 ispirandosi alla topologia della corteccia del cervello. Tengono conto delle connessioni tra neuroni e dell'influenza che può avere un neurone sui suoi vicini: i neuroni vicini a neuroni attivi rinforzano i legami, mentre a quelli che si trovano ad una data distanza vengono indeboliti i legami.



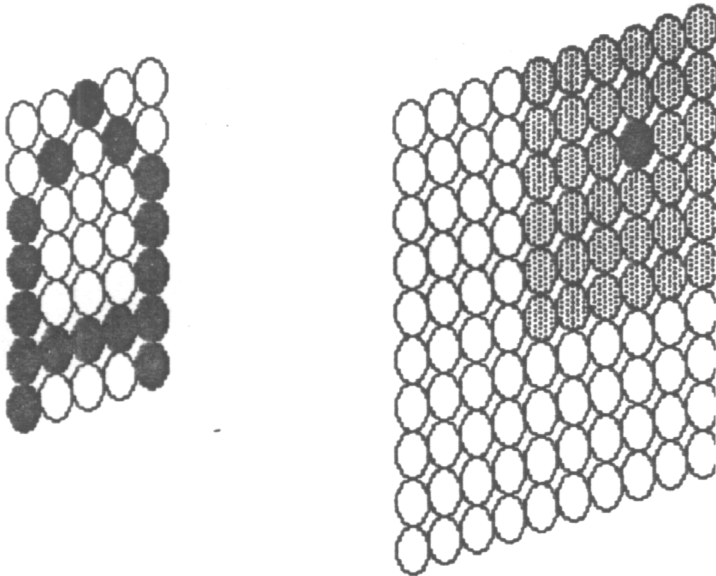
Una rete SOM è composta da un livello di input ed un livello di competizione che localizzati su una griglia ad una o due dimensioni e con diverse forme. Ciascun neurone di input è connesso a tutti i neuroni della griglia; ogni neurone di output ha il vettore dei pesi con le stesse dimensioni del vettore di input.

Trovano ottime applicazioni nel data mining e nella classificazione di stelle e galassie.

Interpretazione della mappa

La configurazione finale dei pesi dei singoli neuroni permette di suddividere gli elementi forniti in ingresso in cluster (raggruppamenti di "similarità") che di quegli elementi rappresentano una classificazione.

Occorre localizzare sulla mappa i neuroni attivi ed associarli con gli input presentati.



Alla fine dell'addestramento, ogni lettera attiverà una differente combinazione di neuroni contenente il neurone vincitore per quella lettera. Il neurone vincente sarà attivato anche da sottomissione della lettera parziale o con rumore.

Apprendimento in Som

Creare un training set con molti vettori di input significativi; stavolta non ci sono output con cui paragonare la risposta della rete.

Per ogni vettore di input:

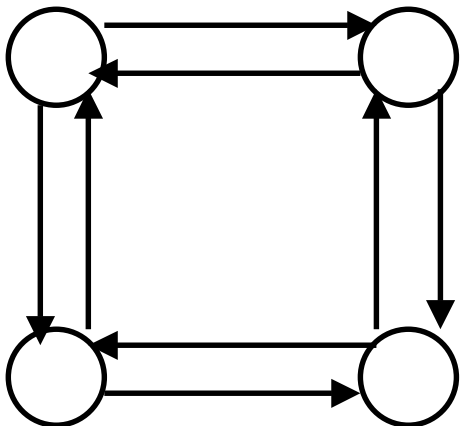
1. ogni neurone di output riceve le componenti del vettore input e ne calcola la distanza euclidea dal suo vettore di pesi
2. il neurone di output, che ha la minima distanza euclidea dall'input, si attiva e dà una risposta maggiore, modifica poi i suoi pesi e quelli dei neuroni vicini per avvicinarli all'input.

Così input simili attiveranno neuroni vicini.

Occorre stabilire quali sono i neuroni vicini e come modificare i loro pesi.

Reti di Hopfield

Hopfield nel 1982 propone una rete per memorizzare informazioni. E' composta da neuroni completamente connessi con funzione di attivazione; ogni neurone è nodo di ingresso e di uscita ed ha stato attivo o disattivo



Alla rete è associata una funzione energia da minimizzare durante la sua evoluzione con una successione di stati, fino a raggiungere uno stato finale corrispondente al minimo della funzione energia.

Hopfield

Proprietà:

- rappresentazione distribuita su tutti i neuroni
- adatta per realizzare una memoria indirizzabile per contenuti: per recuperare un'informazione basta produrne una versione parziale o qualche indizio
- adatta per compiti di ottimizzazione vincolata, quando il problema può essere espresso in termini di una funzione obiettivo da trasformare in una funzione energia
- recupero efficiente di informazioni anche con danni alla rete

Parte 4

Creare una rete
per classificare

La classificazione

Classificare significa dividere un insieme di oggetti in insiemi disgiunti secondo un criterio stabilito a priori; in genere si assegna una etichetta ad ogni insieme creato.

Il pattern recognition (riconoscimento di configurazioni) è la tecnica che consente di creare classificatori numerici e automatici.

Ogni oggetto deve essere rappresentato con un vettore di numeri per essere classificato da una rete neurale, per cui ad ogni oggetto si associa un pattern, un vettore di feature che contraddistingue univocamente l'oggetto.

Il classificatore numerico

Un classificatore numerico si può così definire: date N classi di appartenenza tra cui discriminare, il vettore di input x a L dimensioni delle feature da classificare, il vettore di uscita y che individua la classe formato da N valori, un classificatore riceve in input il vettore x e restituisce in uscita il vettore y dove $y_i=1$ se l'oggetto con input x appartiene alla classe i e $y_j=0$ per $i \neq j$, per $i, j=1..N$.

E' quindi un mapping, o corrispondenza tra valori di input ed output che può essere modellato con una funzione non lineare; data la non linearità, conviene impiegare una rete neurale.

Scelta delle feature

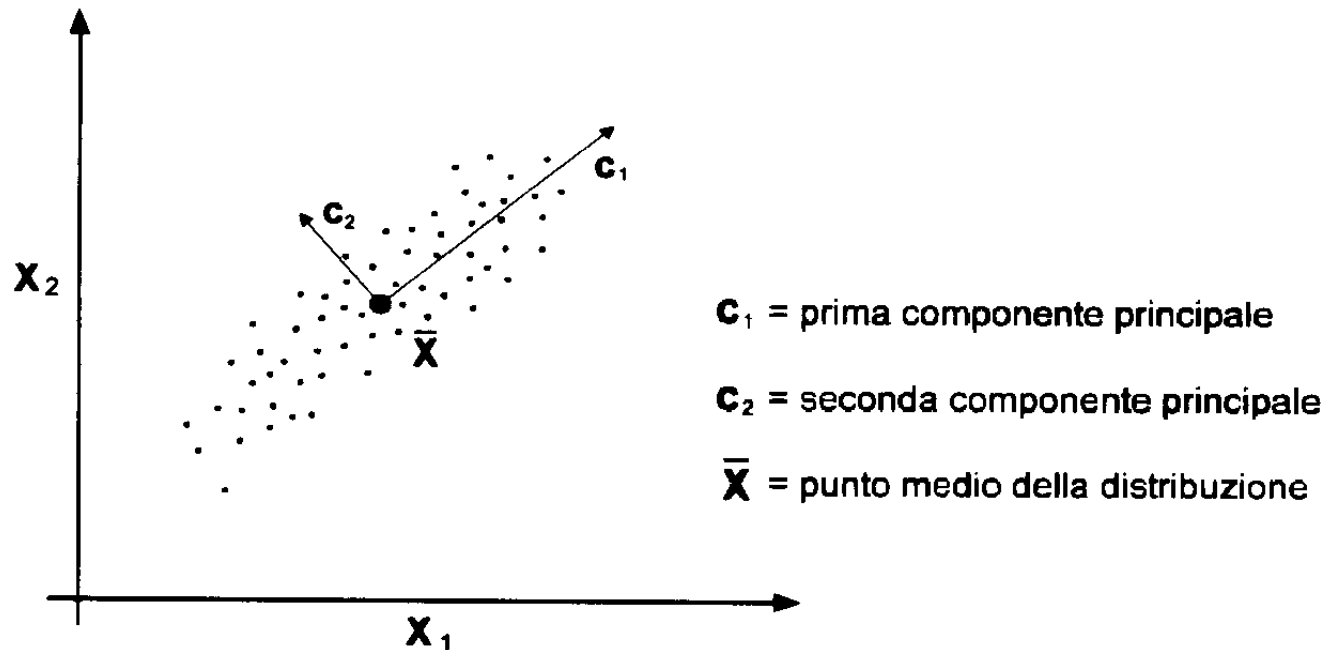
Le feature sono caratteristiche numeriche di un oggetto, ricavate misurando alcune sue proprietà significative. Devono essere scelte con le proprietà:

- discriminanza: i valori delle feature sono simili per oggetti appartenenti alla stessa classe e sono molto diversi per oggetti appartenenti a classi diverse
- indipendenza: i valori delle feature non devono essere correlati tra loro
- minimalità: devono essere il minimo numero possibile di proprietà
- disponibilità: facili e veloci da calcolare.

L'intervallo dei valori di ogni feature deve essere normalizzato in $[0,1]$ o $[-1,1]$ per non avere ordini di grandezza troppo diversi nell'apprendimento.

Riduzione delle feature

Ridurre il numero delle feature può velocizzare l'apprendimento (crescita di dimensionalità). Una tecnica statistica efficace è la P.C.A., l'analisi delle componenti principali dell'insieme dei valori delle feature. Per descrivere l'insieme nella figura bastano i valori delle prime due componenti.



Progettare la rete

Per un tipo di apprendimento supervisionato:

- Individuare le classi in cui dividere l'input secondo il tipo di problema
- Scegliere le feature analizzando matematicamente gli oggetti in input
- Definire molte coppie (input, output) per i set di training (60%), validation (20%), test (20%)
- Definire la codifica numerica: per l'input valori in $[-1,1]$; per l'output valori binari $\{0,1\}$

Progettare la rete

- Scegliere un modello di rete e definirne l'architettura con:
 - Funzione di attivazione per ogni neurone
 - Numero di livelli hidden e numero di neuroni per ogni livello hidden (non esistono precise regole per determinarli, solo con tentativi e verifiche degli errori commessi)
 - Numero di neuroni per lo strato input: tanti quanti i valori delle feature
 - Numero di neuroni per lo strato output: tanti quante sono le classi
 - Scegliere un algoritmo di apprendimento e i suoi parametri di controllo (es. back propagation)
 - Scegliere una tecnica per controllare l'apprendimento (es. early stopping)
 - Scegliere alcuni criteri per valutare la qualità della risposta globale della rete sul test set
-

Usare la rete neurale

- Dopo l'addestramento e la prova col test set, si congela la rete e non si modificano più i pesi. Per usarla in un impianto occorre:
- calcolare le feature del nuovo input e di cui non si conosce l'output
 - passarle in input alla rete neurale che le elabora calcolando le risposte dai neuroni dal livello input verso il livello output
 - la risposta della rete va interpretata per decidere quale classe ha scelto; in genere non si ha un neurone con valore 1 e tutti gli altri 0 per effetto delle funzioni di attivazione e di apprendimento: la classe assegnata è quella con valore di output più alto
 - Secondo il tipo di problema, se il valore di output è basso (es. 0,7 invece di 1), si può decidere di usarlo o di rifiutarlo facendo classificare ad un altro esperto

Strumenti software

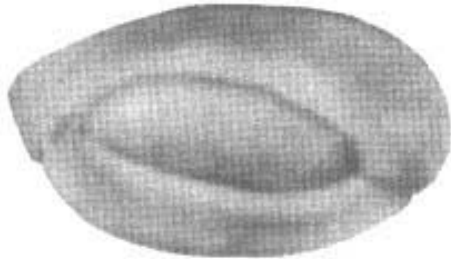
- MatLab, si usano due pacchetti specialistici ben fatti:
 - Neural Toolbox,
<http://www.mathworks.com/products/neuralnet/>
 - NetLab, <http://www.ncrg.aston.ac.uk/netlab/>
- Java, esistono vari pacchetti gratuiti su Internet, link su queste pagine:
 - <http://www.geocities.com/fastiland/NNwww.html>
 - http://www.mathtools.net/Java/Neural_Networks/
 - <http://diwww.epfl.ch/w3mantra/tutorial/english/index.html>

Basta scegliere nei menù le voci corrispondenti alla struttura della rete e alle funzioni desiderate; il toolbox ha algoritmi ottimizzati per l'apprendimento desiderato e fornisce l'output della rete.

Parte 5

Esempio di applicazione

Esempio di classificazione: modi di accrescimento di camere in foraminiferi



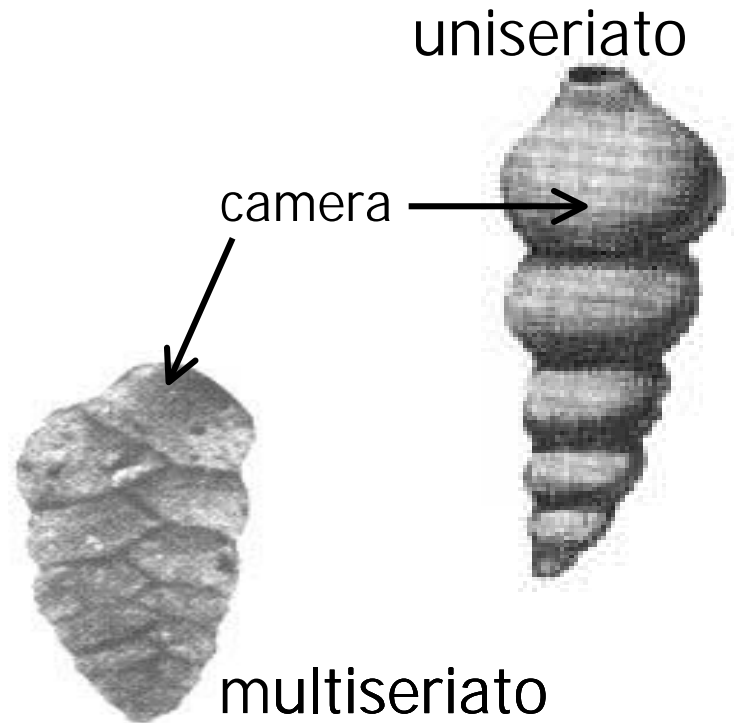
agatostego



spiralato



irregolare



camera

uniseriato

multiseriato

Obiettivo della classificazione

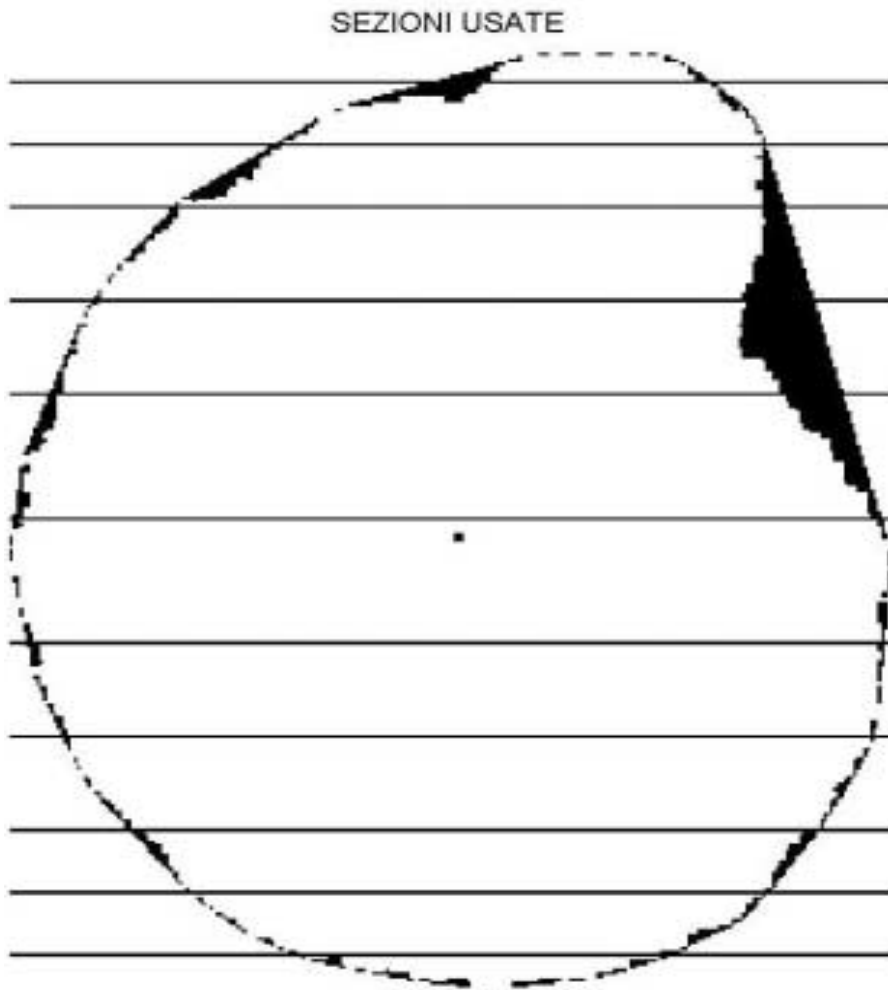
Classificare il modo di accrescimento delle camere di un foraminifero intero setacciato da arenaria e fotografato da microscopio

Si sceglie un apprendimento supervisionato perché si conoscono già le classi ed esistono numerosi esempi di classificazione presi da cataloghi di immagini. Essendo una classificazione, si sceglie la Mlp.

I profili degli oggetti sono molto diversi quindi è inutile cercarne una descrizione matematica; non è possibile individuare le singole camere e contarle. Quindi si può usare:

- un campionamento non uniforme e adattivo dello spessore del guscio in varie parti
- feature calcolate su tutto il guscio

Feature scelte



In alto e in basso si misurano più spessori perché in tali zone si differenziano maggiormente gli spessori dei gusci. Altre feature:

- rapporto tra area di cerchio e area del fossile
- eccentricità del fossile (0=retta, 1=cerchio)
- spessore del guscio in pixel in 11 sezioni

totale di 13 feature ridotte a 10

Rete neurale per classificare

- ogni valore di input normalizzato in $[-1,1]$
- ogni valore di output assume valore in $[0,1]$ e somma 1
- rete Mlp: 10 nodi input , 8 nodi interni, 5 nodi output
- funzione attivazione neurone: softmax
- funzione di errore: cross-entropy
- funzione minimizzazione errore: quasi-newton
- apprendimento con riduzione di errore di generalizzazione tramite early stopping
- training set di 209 immagini
- validation set di 68 immagini

Risultati della classificazione

test set di 70 immagini , 1 non classificato bene percentuale
correttezza 98,57%

matrice di confusione

	1	2	3	4	5	riga=classe da rete
1	17	0	0	0	0	colonna=classe reale
2	0	17	1	0	0	1=agatostego
3	0	0	15	0	0	2=seriale
4	0	0	0	4	0	3=multiseriale
5	0	0	0	0	16	4=irregolare
						5=spiralato

Esempi su sequenze di numeri

- Predizione: una serie di dati può essere data in input ad una rete neurale per fargli decidere quale sarà il prossimo valore, in base al modello statistico che la rete neurale si crea con i dati precedenti
- Interpolazione: un sensore fornisce un dato ma ogni tanto si guasta e non fornisce il dato; una rete neurale può prendere in input tutti i dati forniti dal sensore e ricavare i dati mancanti, in base al modello statistico che la rete neurale si crea con tutti i dati