

EVOLUZIONE DEL MODELLO NEURALE

1 RETI A DECISIONE BINARIA

(McCulloch e Pitts, 1943)

Per la prima volta viene proposto un modello di elaborazione dell'informazione la cui singola unità è il neurone

2 IL PERCEPTRON

(Rosenblatt, 1962 – Minsky e Papert, 1967)

Si affronta il problema della determinazione dei coefficienti per reti costituite da unità di elaborazione del tipo di McCulloch e Pitts

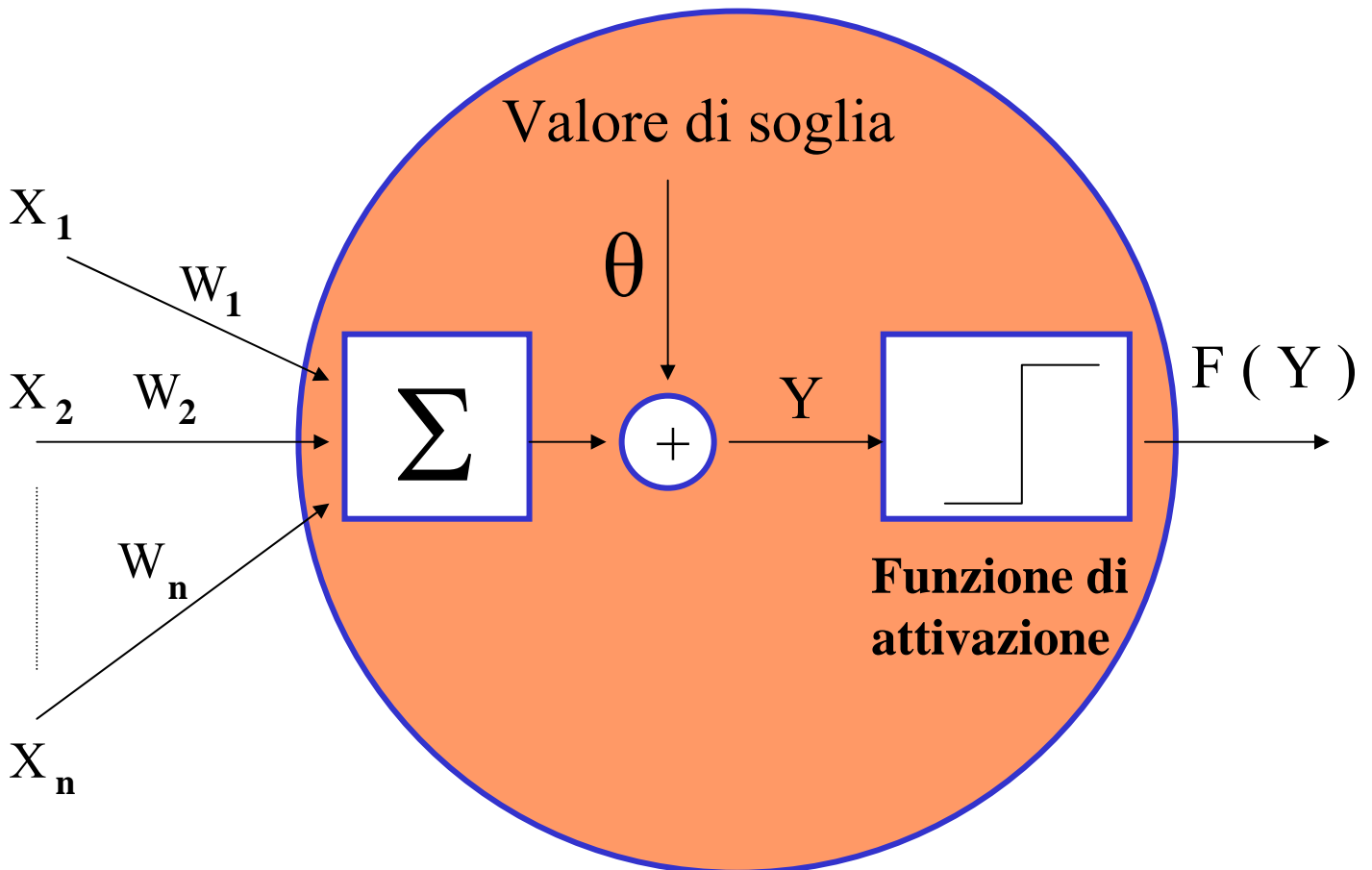
3 LA BACK-PROPAGATION

(Werbos, 1974 –

Rumelhart, Hilton e Williams, 1986)

Viene introdotta una procedura per la determinazione dei coefficienti degli strati interni

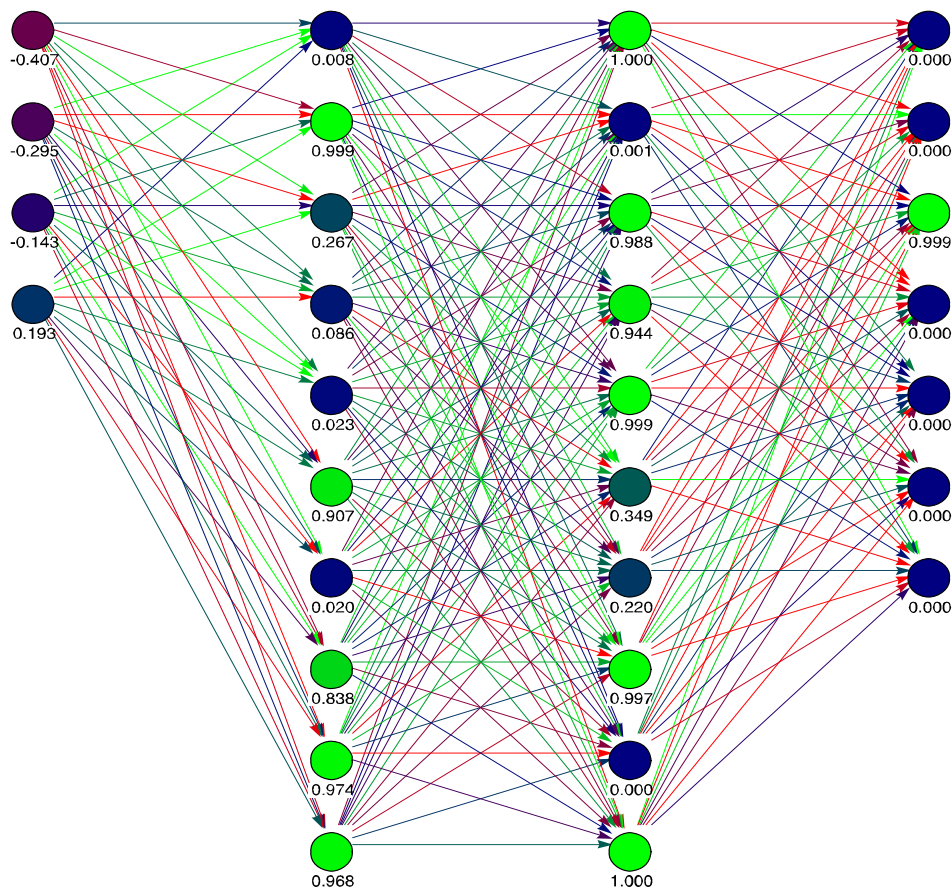
1 Il neurone



$$\text{Funzione Gradino: } F(Y) = \begin{cases} 1 & Y > 0 \\ 0 & Y \leq 0 \end{cases}$$

Una rete formata da tali unità è, con la scelta degli opportuni coefficienti, in linea di principio capace di rappresentare qualsiasi funzione logica

2 Il perceptron



Rosenblatt dimostra analiticamente e sperimentalmente come sia possibile addestrare la rete tramite un algoritmo iterativo e convergente (teorema della convergenza del perceptron) in grado di riconoscere un prefissato insieme di patterns (F. Rosenblatt, *Principles of Neurodynamics*, Spartan, 1962)

Principio della regola di addestramento: presentare degli esempi alla rete e modificare i pesi delle connessioni per ottenere un miglioramento del risultato, così da avvicinarsi alle uscite desiderate

Minsky e Papert dimostrano che il teorema è valido solo per quel tipo di situazioni risolvibili con il perceptron a singolo strato

Esisteva tuttavia una vasta classe di problemi anche elementari che il perceptron non era in grado di risolvere. Esempio classico: **OR esclusivo**. Bisognava aggiungere altri strati ma a questo punto veniva a mancare l'algoritmo opportuno per la determinazione dei coefficienti degli strati interni : *problema dell'assegnazione delle responsabilità* (M.L. Minsky, S.A. Papert, *Perceptrons*, MIT Press, 1969).



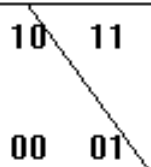

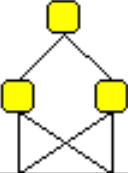
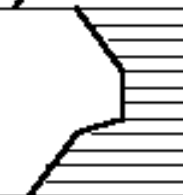
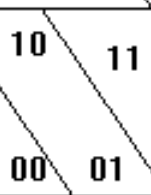

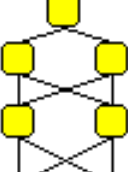

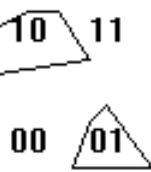

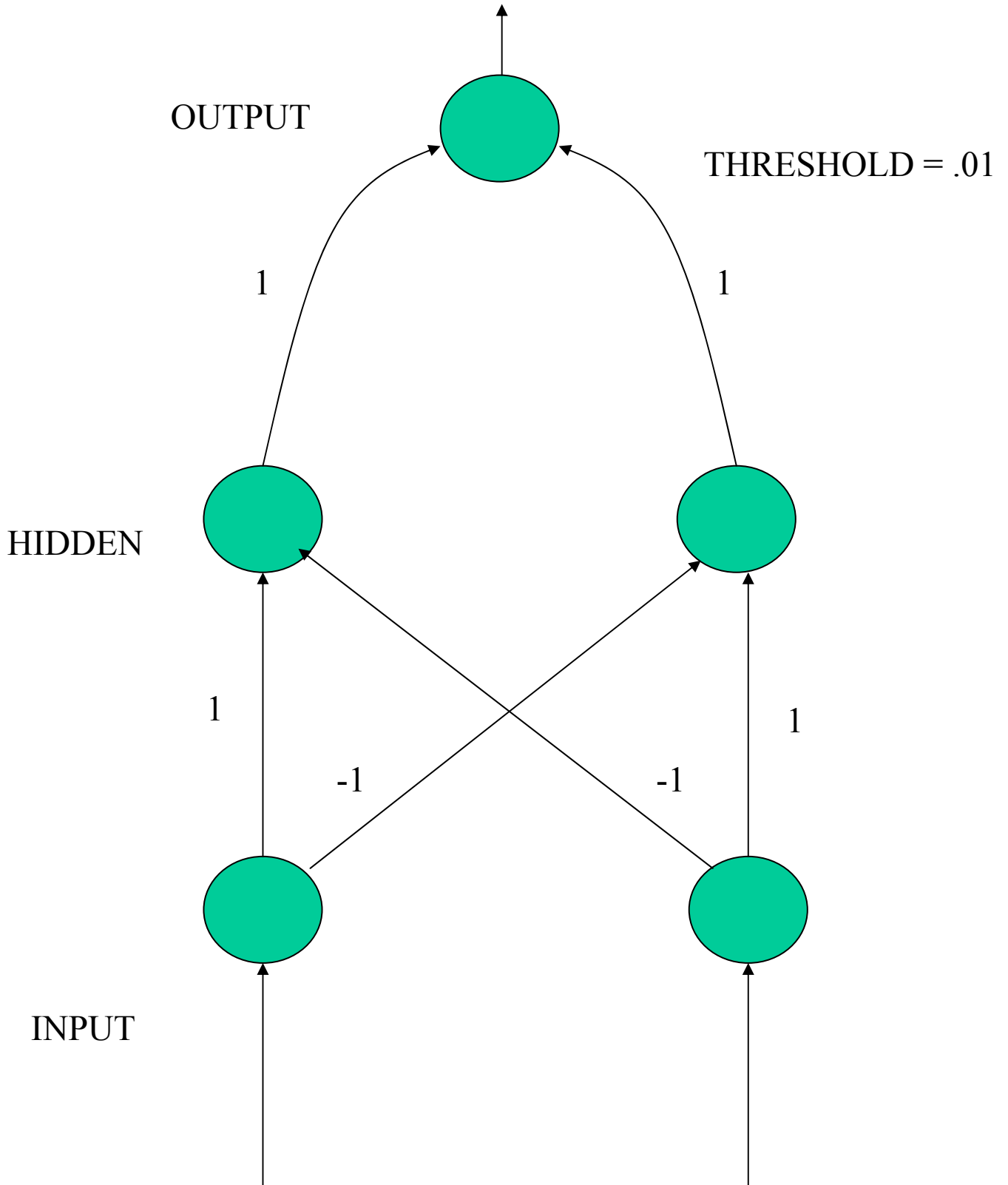
numero strati	forme	xor	note	regioni incuneate
 1			semipiani	
 2			regioni convesse	
 3			regioni complesse	

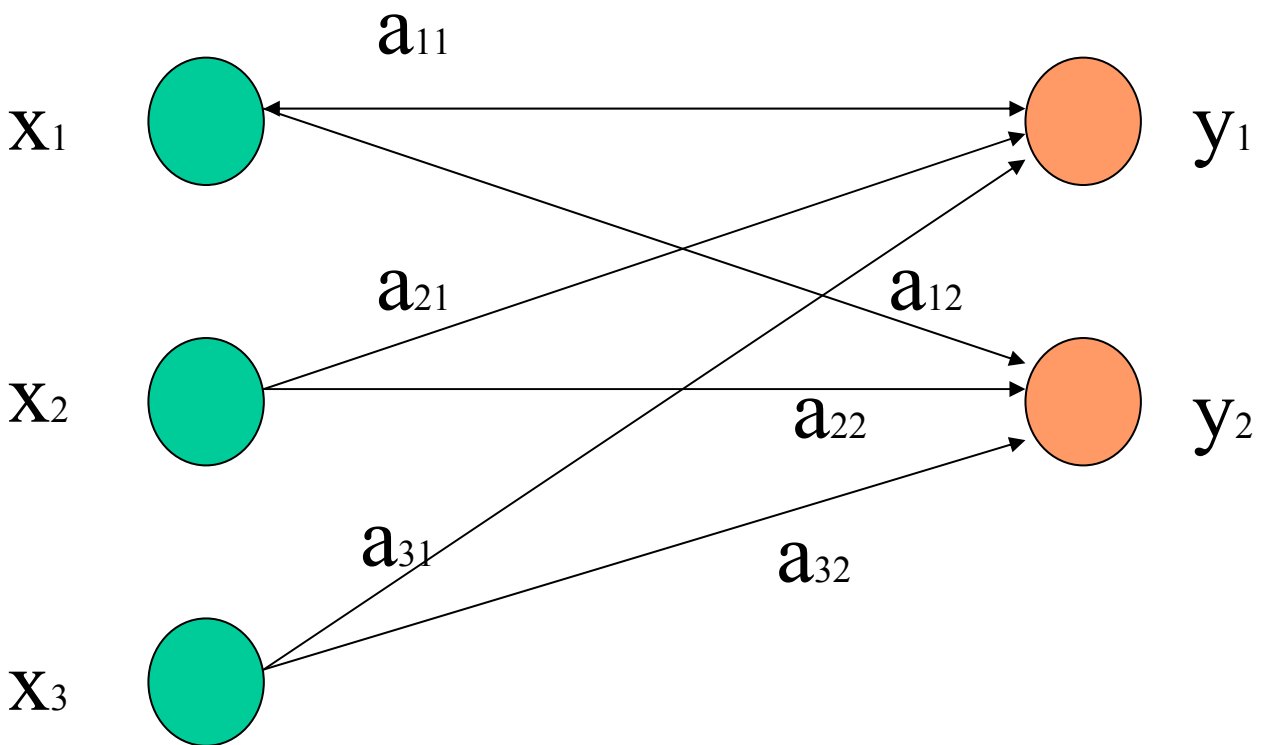
fig.7 forme riconoscibili con diversi numeri di strati

XOR NETWORK



Le regressioni lineari possono essere viste come forme degeneri di reti perceptron senza strati interni.

Esempio



Se le funzioni di attivazione sono funzioni identità arriviamo a:

$$y_1 = a_{11} X_1 + a_{21} X_2 + a_{31} X_3$$

$$y_2 = a_{12} X_1 + a_{22} X_2 + a_{32} X_3$$

3 La back-propagation

P. Werbos. Beyond regression: new tools for prediction and analysis in the behavioural sciences, *Ph.D. Thesis*. Harvard University, 1974.

D.E. Rumelhart, G.E. Hinton, R.J. Williams, *Learning internal representations by error propagation*, MIT press, 1986.

Risolve la difficoltà dell'addestramento dei neuroni interni.

Si considera una rete con funzioni di attivazione differenziabili, cosicché le funzioni di attivazione dell'uscita diventano funzioni differenziabili sia delle variabili di ingresso che dei pesi e dei valori di soglia. Le derivate possono essere utilizzate per trovare i valori dei coefficienti che minimizzano la funzione errore

$$E = \frac{1}{2} \sum_p \sum_{k=1}^{N_{output}} (y_k - t_k)^2$$

Nel 1989 Hornik et al. dimostrano che il perceptron multistrato è in grado di approssimare qualsiasi funzione continua a valori reali: **le reti perceptron multistrato costituiscono una classe di approssimatori universali**

K. Hornik, M. Stinchcombe, A. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, 1989.

Classificazione di reti neurali

Possiamo distinguere 3 elementi fondamentali:

1. Topologia della rete:

Feedforward, ricorsiva, completamente connessa

2. Caratteristiche del neurone:

Funzione di attivazione **sigmoidale**, a gradino, gaussiana

3. Regola di apprendimento:

Con supervisione

Senza supervisione

**Percettrone multi-strato
(multi-layer perceptron)**

Regolarizzazione

Si incoraggia la rete a fornire curve più semplici e con fluttuazioni di ampiezza più contenute

Si aggiunge una funzione penalità Ω alla funzione di errore E giungendo a definire una nuova funzione E'

$$E' = E + \mu \Omega$$

Dove il parametro μ controlla con che intensità il termine di penalità influenza la forma della soluzione

Una delle forme più semplici e usate di regolarizzatori è quella a decadimento del peso:

$$\Omega = \frac{1}{2} \sum_i w_i^2$$

L'abbassamento dei valori dei coefficienti smorza la tendenza a generazioni di funzioni con andamenti particolarmente irregolari.

Anche l'aggiunta di rumore nei dati può agire come regolatore

Stabilizzazione strutturale

Tecniche per tentativi

si addestra un insieme di reti con un numero differente di unità nascoste e si seleziona quella che dà il risultato migliore

Tecniche di pruning

Si allena una rete inizialmente sovradimensionata. Terminato l'addestramento, la rete viene analizzata attraverso l'importanza di ciascuno dei suoi pesi, e il meno importante viene eliminato. La rete ottenuta viene poi nuovamente riaddestrata

Nota. Estendendo le tecniche di pruning fino ad arrivare a interessare lo strato di ingresso può essere un utile metodo per individuare i canali con maggiore contenuto applicativo

Tecniche di growing

Si parte con una rete sottodimensionata e man mano durante l'addestramento vengono aggiunte nuove unità o strati congelando i coefficienti determinati nelle fasi precedenti.

La riduzione delle dimensionalità

Motivazioni

- ☑ Migliora le proprietà di generalizzazione della rete
- ☑ Accelera il processo di addestramento

Tecniche

- ☑ Analisi a componenti principali
- ☑ Reti autoassociative (analisi a componenti principali non lineari)

LE RETI NEURALI

VIZI.....

- ❑ Problema dei minimi locali
- ❑ I pattern di addestramento devono rappresentare esaurientemente la statistica

E VIRTÙ

- ❑ Approssimatori universali
- ❑ Portabilità
- ❑ Robustezza
- ❑ Facilità di implementazione

```

;Function      : neural
; INPUT       : Features
  OUTPUT      : Gives out the neural response on the probability
for the object to be an oil spill
;-----
common NEURAL, scaling_coef,mn_threshold,mn_conn
act=fltarr(28)
act(0)=Features.area
act(1)=Features.perimeter
.
act(10)=Features.gradient[3]

for i=0,10 do act(i)=scaling_coef(0,i)*act(i) + scaling_coef(1,i)

for i=11,18 do begin
  act(i)=0
  for j=0,10 do act(i)= act(i) + mn_conn(j,i)*act(j)
  act(i)=act(i) + mn_threshold(i)
  act(i)=1/(1+ exp(-act(i)))
endfor

for i=19,26 do begin
  act(i)=0
  for j=11,18 do act(i)= act(i) + mn_conn(j,i)*act(j)
  act(i)=act(i) + mn_threshold(i)
  act(i)=1/(1+ exp(-act(i)))
endfor

act(27)=0
for j=19,26 do act(27)= act(27) + mn_conn(j,27)*act(j)
act(27)=act(27) + mn_threshold(27)
act(27)=1/(1+ exp(-act(27)))

```

ftp.informatik.uni-stuttgart.de

cd /pub/SNNS

STUTTGART NEURAL NETWORK SIMULATOR

SNNS

V 4.2

Andreas Zell, Günter Mamier, Michael Vogt
Niels Mache, Tilman Sommer, Ralf Hübner
Michael Schmalzl, Tobias Soyez, Sven Döring, Dietmar Posselt
Kai-Uwe Hermann, Artemis Hatzigeorgiou

external contributions by:
Martin Riedmiller, Heike Speckmann, Martin Reczko
Jamie DeCoster, Jochen Biedermann, Joachim Dänz, Christian Wehrfritz
Randolf Werner, Michael Berthold

(c) 1990-95, IPVR, University of Stuttgart
(c) 1996-98, WSI, University of Tübingen

FILE	CONTROL	INFO	DISPLAY	3D DISPLAY	GRAPH	BIGNET
PRUNING	CASCADE	KOHONEN	WEIGHTS	PROJECTION	ANALYZER	INVERSION
PRINT	HELP	CLASSES				QUIT

SNNS 4.2 (c) 1990-98 SNNS-Group at IPVR and WSI

*  *  *

Riferimenti bibliografici

PARALLEL DISTRIBUTED PROCESSING

Exploration in the Microstructure of Cognition
Volume 1: Foundations

Volume 2: Psychological and Biological Models

**David E. Rumelhart, James L. McClelland,
and the PDP Research Group**

The MIT Press, 1989.

NEURAL NETWORKS FOR PATTERN RECOGNITION

Christopher M. Bishop

Oxford University Press, 1995.

NEURAL NETWORKS: A COMPREHENSIVE FOUNDATION

Simon Haykin, S.

Prentice Hall, 1998.