

# Circuiti Neurali per il Trattamento del Segnale

*Aurelio Uncini*

Dipartimento INFOCOM,  
Università di Roma “La Sapienza”  
Via Eudissiana, 18 I-00184 ROMA

## *I PARTE*

1. Introduzione
2. Le Reti Neurali Artificiali (ANN)
3. Proprietà matematiche delle ANN
4. Sintesi delle ANN

## *II PARTE*

5. Teoria dei Circuiti e ANN
6. Proprietà matematiche reti dinamiche
7. ANN con dinamica interna
8. Complessità delle ANN
9. ANN con funzione di attivazione adattativa
10. Applicazioni ANN per tratt. Segnali

MENTE      Forma misteriosa di materia secreta dal cervello. La sua attività principale consiste nello sforzo di accertare la propria natura, e la futilità di questo tentativo è dovuta al fatto che per conoscere se stessa non possiede altro che se stessa

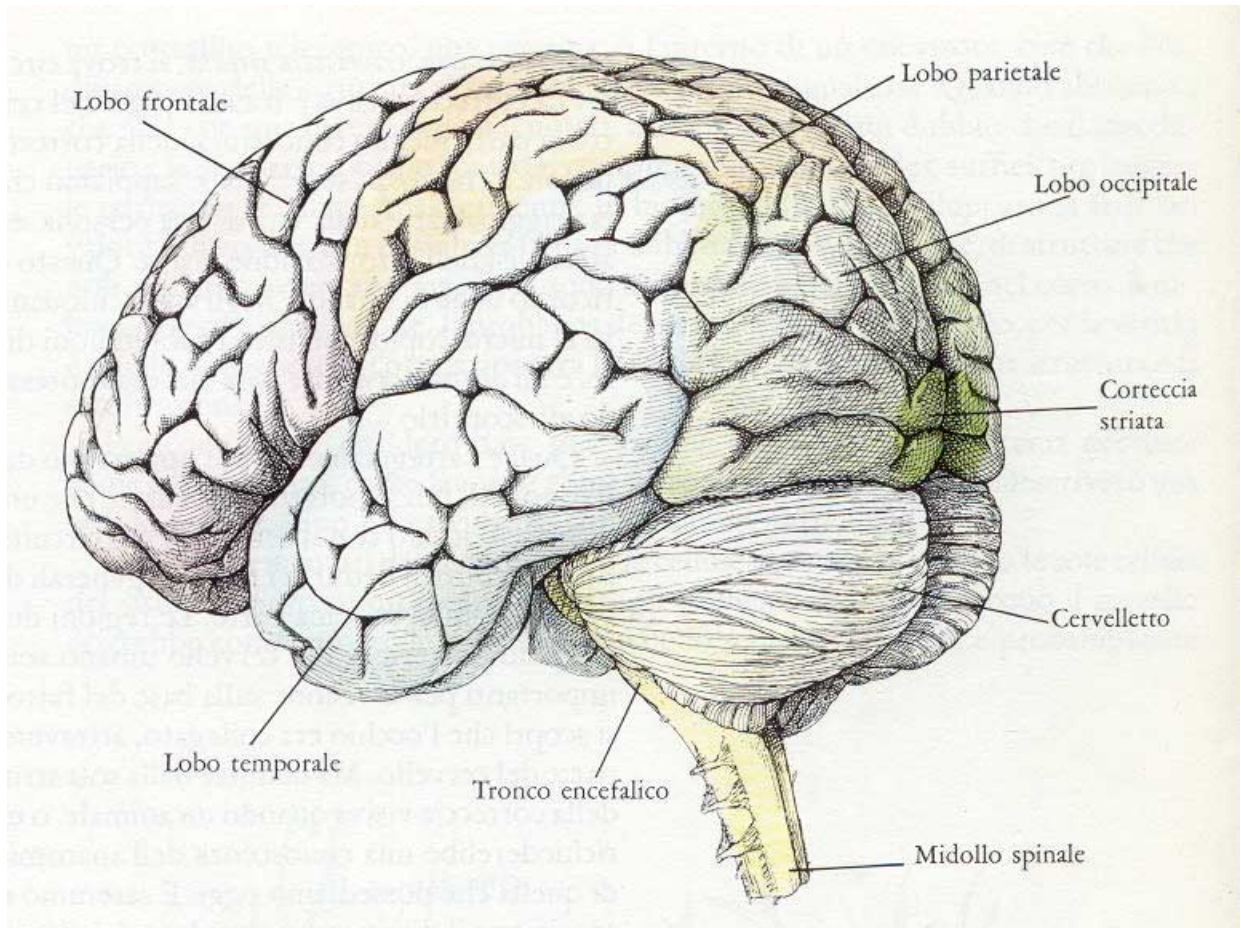
[AMBROSE BIERCE]

## CHI SCOPRÌ LE RETI NEURALI ....

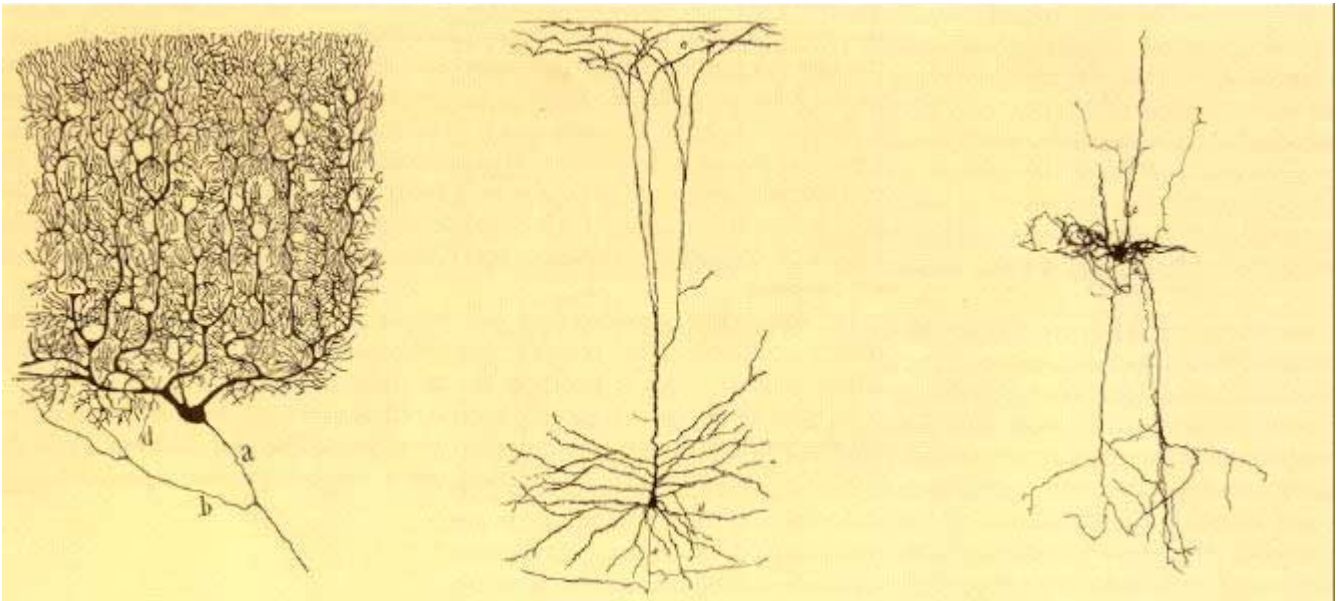
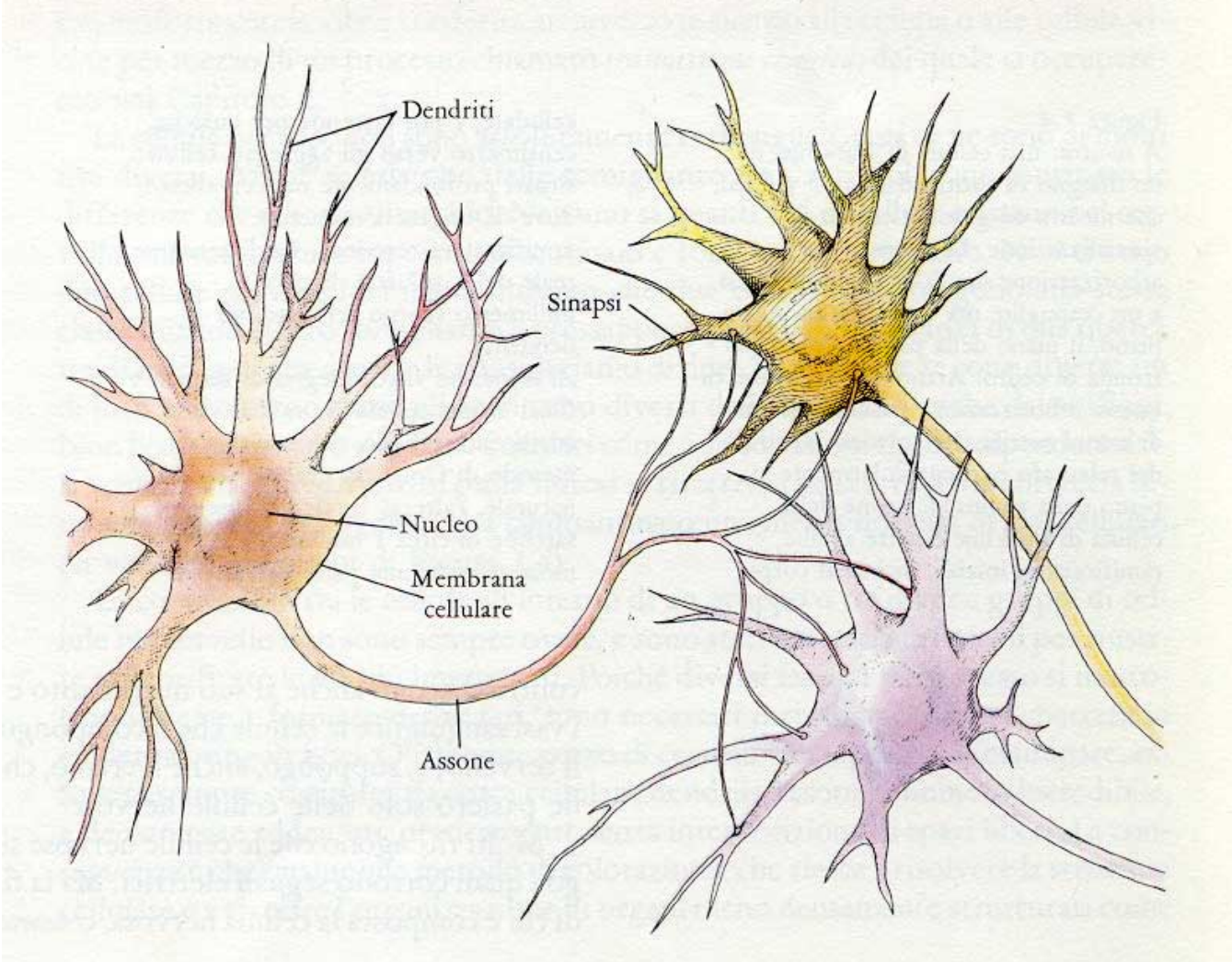


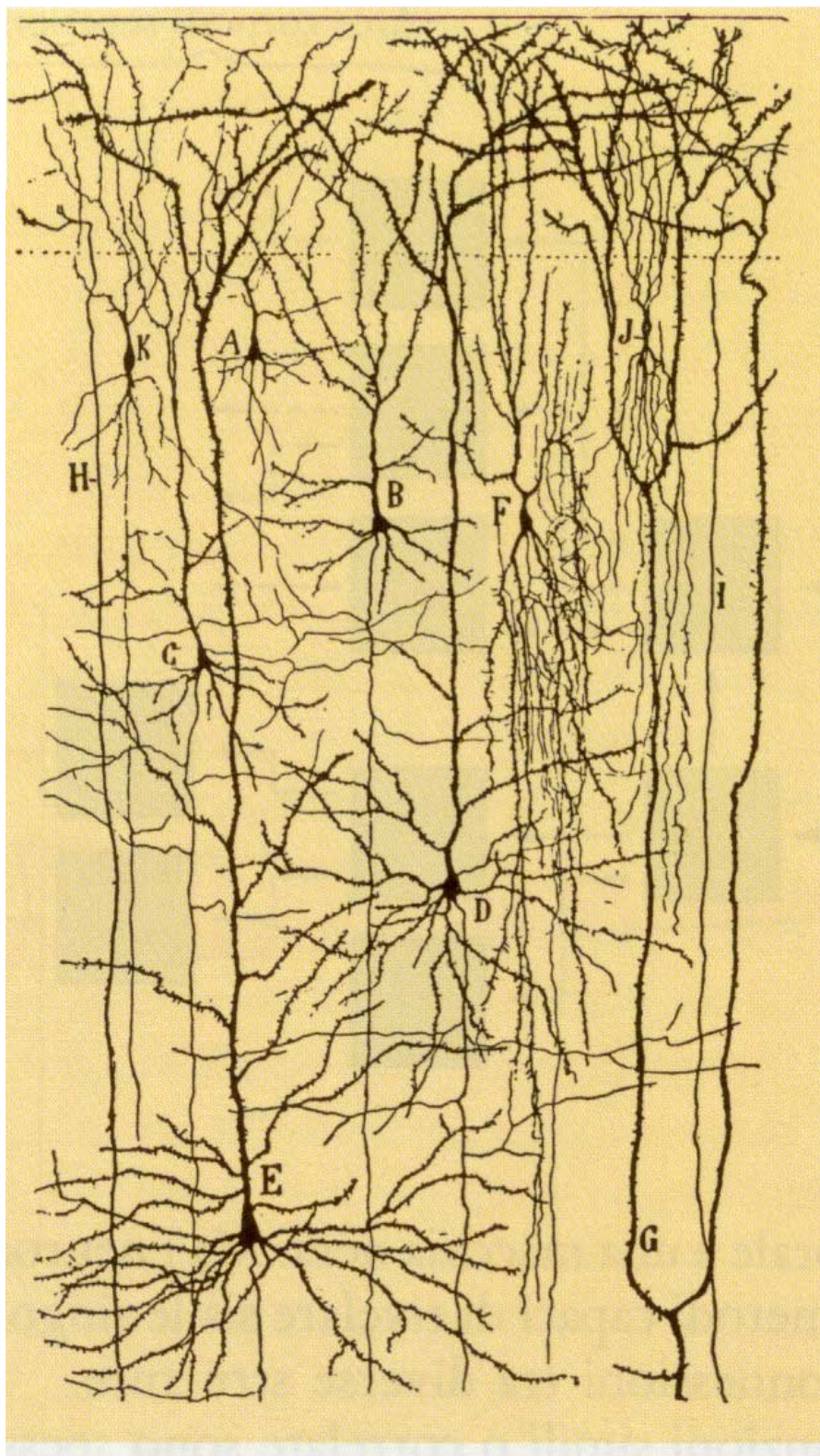
Santiago Ramòn y Cajal (Nobel 1906) ha stabilito con assoluta certezza, usando il metodo di Golgi, che le cellule nervose sono delle unità indipendenti.

# UN ESEMPIO DI RETE NEURALE MOLTO EFFICIENTE ...



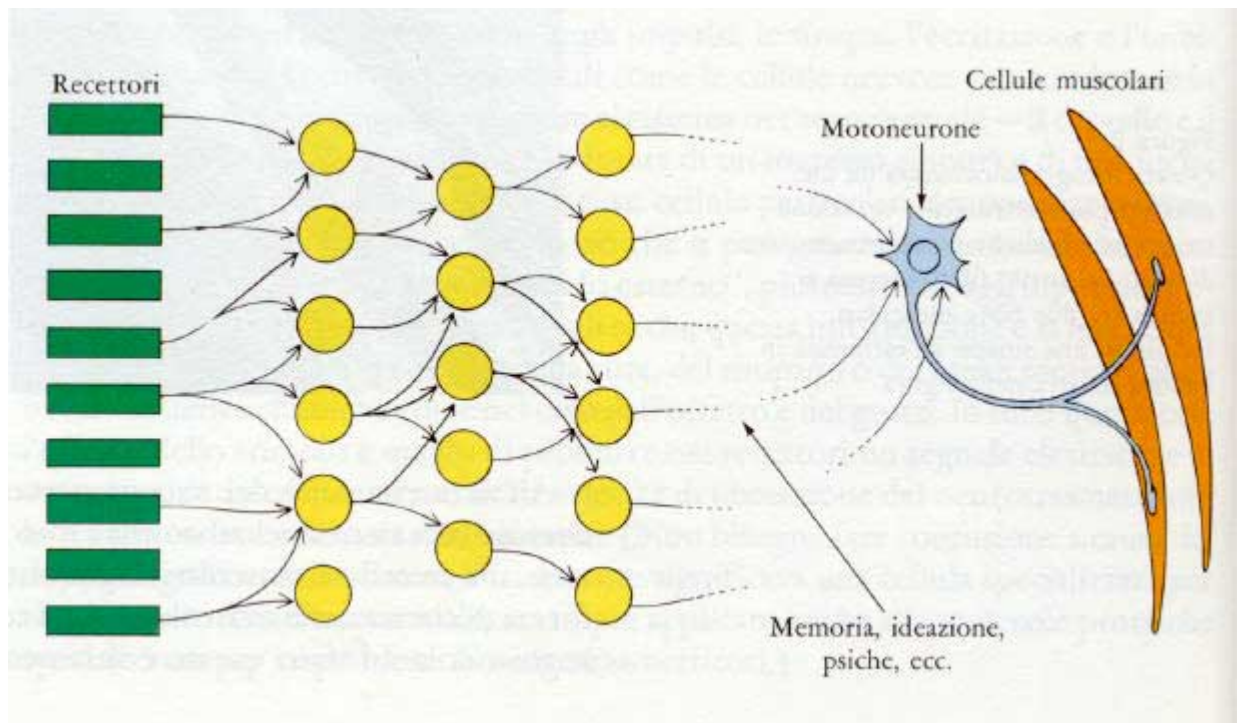
# LA CELLULA NERVOSA





Preparato colorato col Golgi, mostra alcune cellule degli strati più superficiali.

## Organizzazione a strati



Molte parti del sistema nervoso sono organizzate in stadi successivi stratificati.

Una cellula di uno stadio riceve molti *input* eccitatori e inibitori dello stadio precedente e invia un output a numerose cellule dello stadio seguente.

# Caratteristiche del cervello biologico

- Semplicità locale  
Il neurone riceve degli stimoli dagli altri neuroni a cui è collegato e produce un impulso nell'assone proporzionale alla somma pesata degli ingressi
- Complessità globale  
Il cervello umano possiede  $10^{12}$  neuroni, ognuno con anche  $10^4$  connessioni
- Apprendimento  
La forza delle connessioni varia quando la rete è esposta a stimoli esterni
- Tolleranza ai guasti  
Le prestazioni degradano lentamente con l'aumentare dei guasti.
- Velocità di elaborazione  
Capacità di risolvere operazioni molto complesse in tempi brevi (visione, memorizzazione, riconoscimento spaziale e temporale in ambiente umoroso e con dati incompleti)

## 2. LE RETI NEURALI ARTIFICIALI (ANN)

KOHONEN ha definito: "Reti neurali artificiali" delle reti "massicciamente parallele" costituite dalla interconnessione di semplici elementi adattativi che possono interagire con gli oggetti del mondo reale nello stesso modo dei sistemi neurali biologici.

### RICERCA SULLE ANN

#### • Neurobiologia

- meccanismi cellulari
- strutture neurali semplici
- etc

#### • Circuitale

- modelli di circuiti neurali
- algoritmi di apprendimento
- topologie
- etc

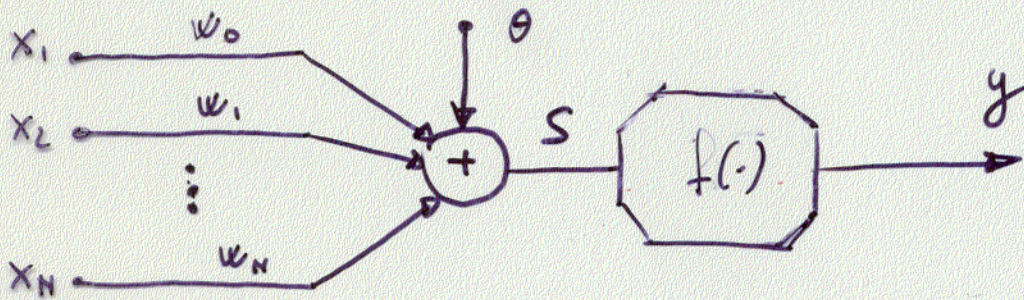
#### • Implementazioni

- simulazioni su
- circuiti VLSI o analogici
- etc

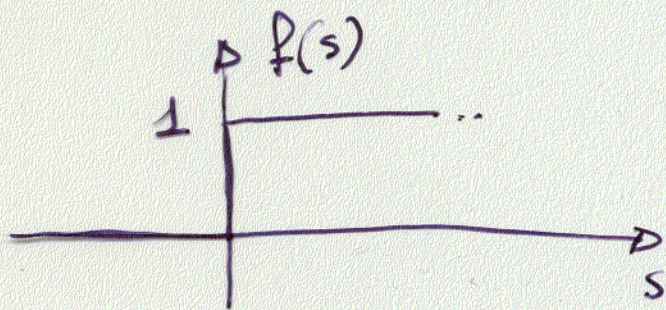
#### • Applicazioni

# IL NEURONE FORMALE

McCulloch e Pitts, sulla base degli studi in campo neurobiologico proposero un semplice modello di neurone (1943)



L'uscita  $y$  assume valore 1 solo se la somma pesata degli ingressi supera il valore di soglia  $\theta$ .



$$y = f\left(\sum_{i=1}^N x_i w_i + \theta\right)$$

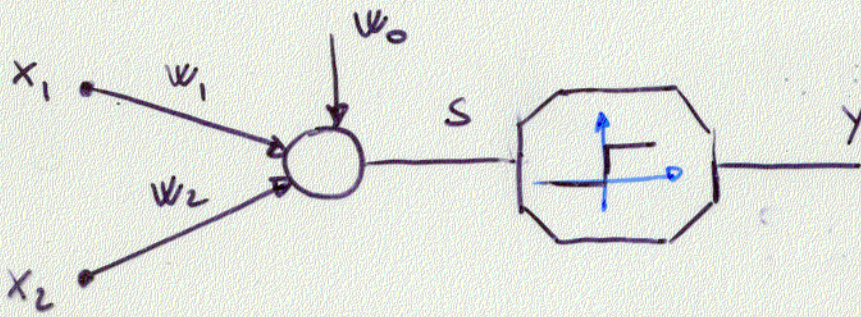
$w_i \triangleq$  peso sinaptico

$f(\cdot) \triangleq$  funzione di attivazione

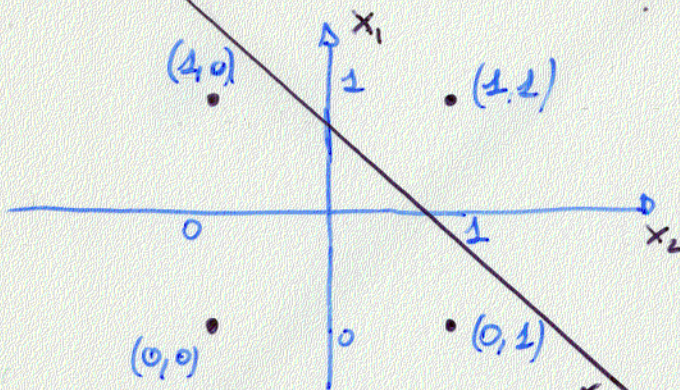
$\theta \triangleq$  soglia

combinatore lineare + funzione di attivazione

# Problema della classificazione



Un solo neurone  $\bar{i}$  in grado di dividere lo spazio degli ingressi  $x_1, x_2$  in due regioni.

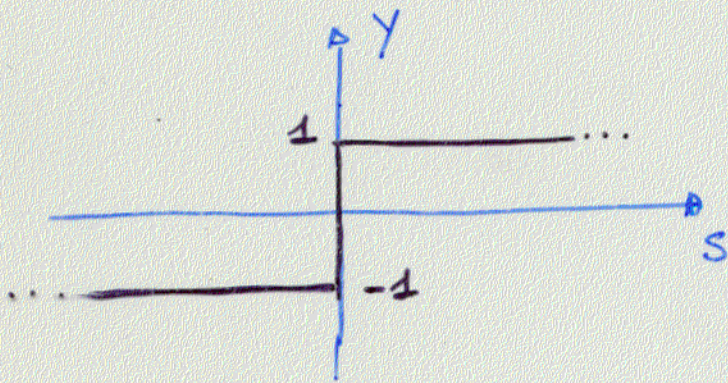


$x_1$	$x_2$	$Y$
0	0	0
0	1	0
1	0	0
1	1	1

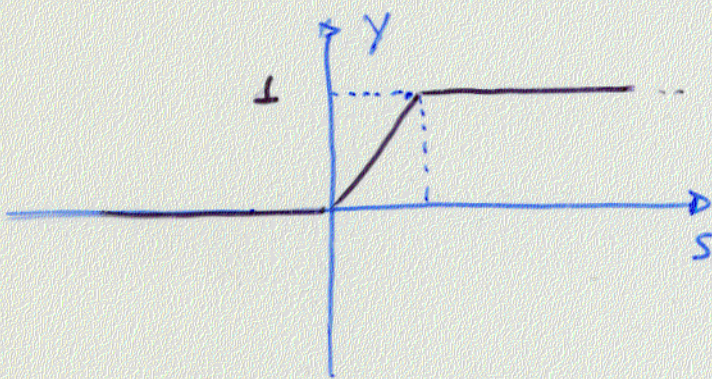
$$s = x_1 w_1 + x_2 w_2 + w_0 \equiv \phi \Rightarrow$$

$$x_2 = \frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$

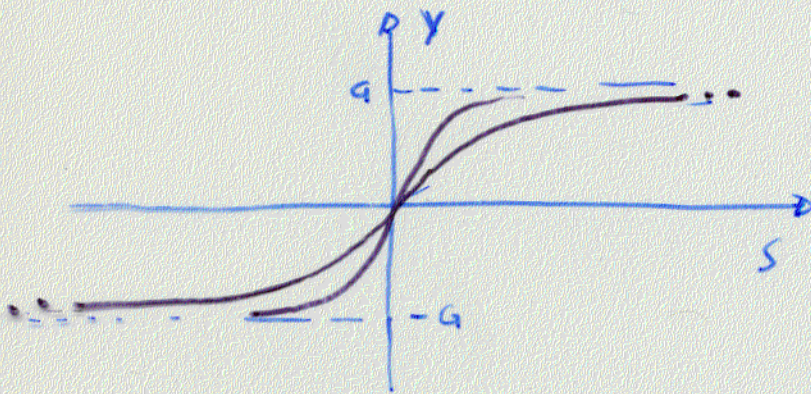
# TIPICI DI FUNZIONE DI ATTIVAZIONE



GRADINO



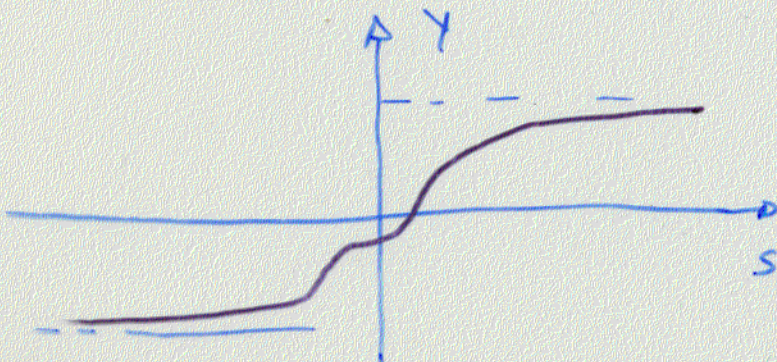
SPEZZATA



SIGMOIDE

$$f = a \frac{1 - e^{-2s}}{1 + e^{-2s}}$$

$a$  = pendenza



POLINOMIALE

SPLINE

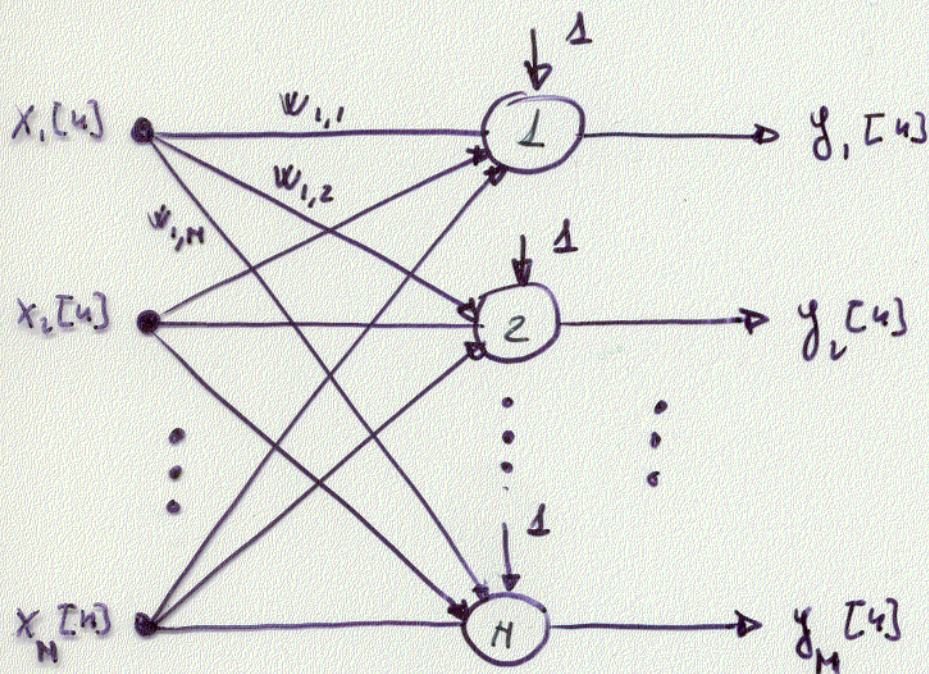
⋮

# TOPOLOGIE DI RETI NEURALI

Una rete neurale artificiale è formata dalle connessioni, effettuate in vari modi, di neuroni.

## ES1 SINGOLO STRATO DI NEURONI

Tale configurazione, proposta da Rosenblatt negli anni '50, è denominata "Perceptron".



$N$  ingressi

$M$  uscite

"Single Layer Perceptron"

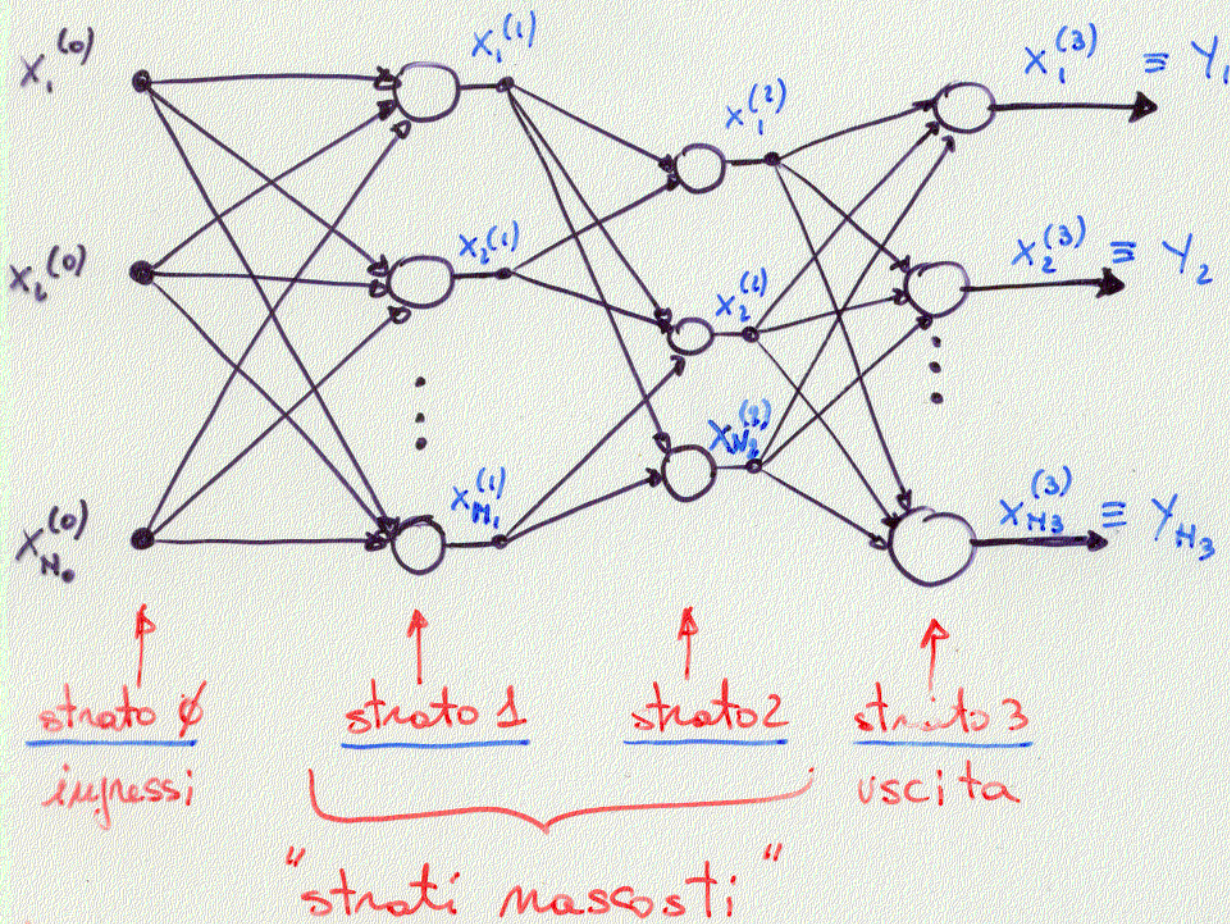
$$y_k[n] = f\left(\sum_{j=0}^M w_{k,j} \cdot x_j\right) \quad k = 1, 2, \dots, M$$

$$x_0 \equiv 1$$

$$w_{k,0} \equiv \theta = \text{soglia}$$

## ES 2 RETE NEURALE A STRATI

L'architettura multi-strato, nota come "Multi Layer Perceptron", è di gran lunga la più usata nelle applicazioni reali.



### CALCOLO DELL'USCITA

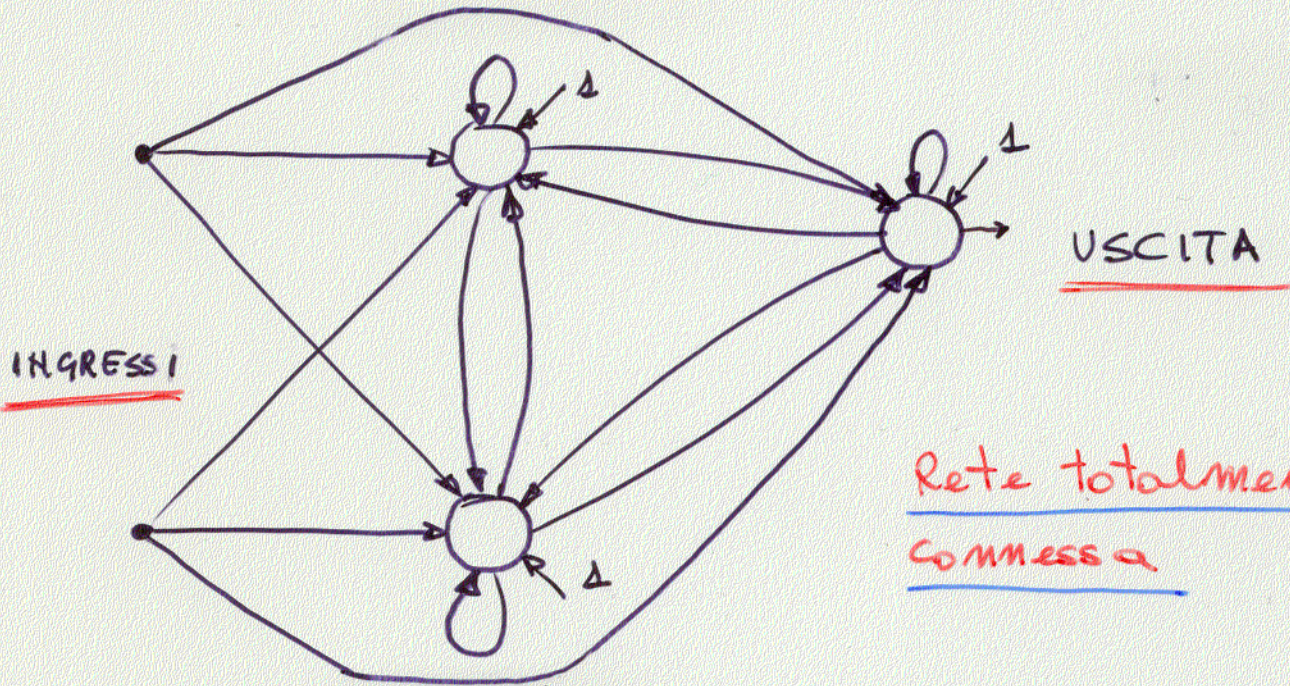
- $x_{\kappa}^{(0)} \equiv x_{\kappa} \quad \kappa = 1, \dots, N_0$  ingressi

- $$\begin{cases} s_{\kappa}^{(e)} = \sum_{j=0}^{N_{e-1}} w_{\kappa j}^{(e)} x_j^{(e-1)} \\ x_{\kappa}^{(e)} = f(s_{\kappa}^{(e)}) \end{cases} \quad x_0^{(e)} = 1 \equiv \text{soglia}$$

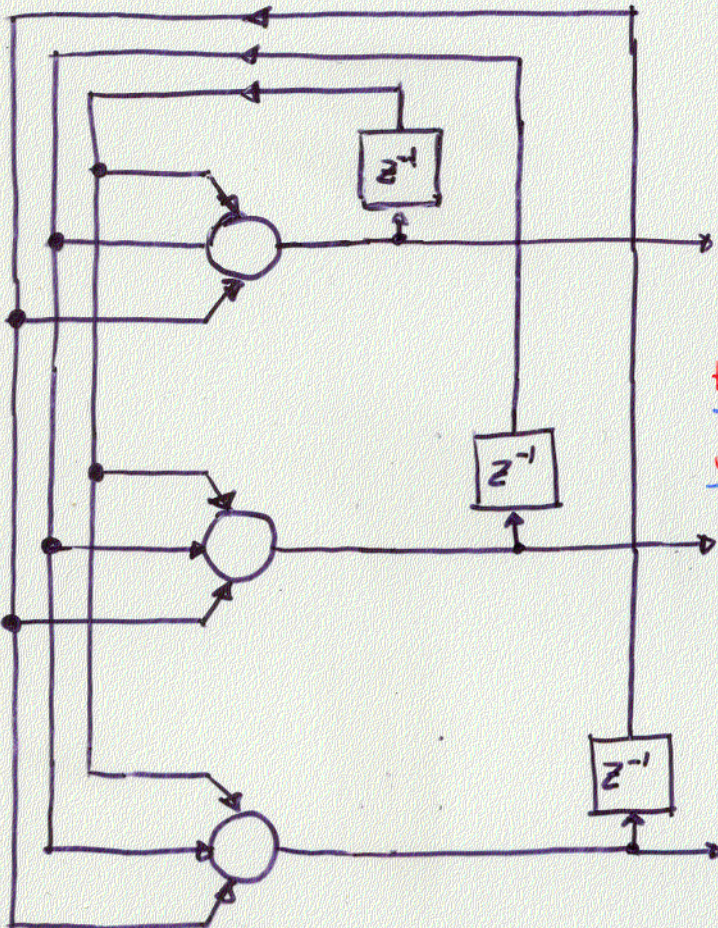
- $y_{\kappa} = x_{\kappa}^{(N)} \quad \kappa = 1, \dots, N_M$  uscite

### ES 3 RETI RICORRENTI

Ogni neurone è collegato con tutti gli altri.  
Si inseriscono meccanismi di feedback.  
La rete ha dinamica temporale.



Rete totalmente  
connessa



Rete di  
Hopfield  
a tempo discreto

### 3. PROPRIETÀ MATEMATICHE DELLE ANN

Dal punto di vista matematico una rete neurale è studiata in base alle espressioni di approssimazione o quelle di rappresentazione di funzioni di più variabili su opportune combinazioni di funzioni di una variabile.

DEF 1 Una funzione reale  $\sigma$  delle variabili  $t$  si dice sigmoideale se è

$$\sigma(t) \rightarrow \begin{cases} 1 & \text{se } t \rightarrow \infty \\ 0 & \text{se } t \rightarrow -\infty \end{cases}$$

DEF 2  $C(I_n)$  denota lo spazio di funzioni continue sul cubo unitario a  $n$  dimensioni  $I_n = [0, 1]^n$ .

$\|f\|$  denota la sup-norma di una  $f \in C(I_n)$

# Approssimazione

2 teoremi fondamentali (Cybenko 1988)

## Teorema 1:

Sia  $\sigma$  una generica funzione sigmoide continua. Le somme finite del tipo

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(w_j^T x + \theta_j)$$

sono dense in  $C(I_n)$ , cioè

$\forall f \in C(I_n)$  e  $\varepsilon > 0 \exists G(x) \therefore$

$$|G(x) - f(x)| < \varepsilon \quad \forall x \in I_n$$

conseguente: ogni funzione continua su  $I_n$  può essere approssimata con la precisione desiderata da una rete MLP ad un solo strato intermedio se funzioni sigmoide continue se non vi sono limitazioni al sul numero di nodi e pesi.

$n \rightarrow$  n° ingressi

$N \rightarrow$  n° unità nascoste

## Teorema 2:

Sia  $\sigma$  una generica funzione sigmoideale continua.  
Sia  $f$  una funzione di decisione per ogni  
partizione finita di  $I_m$  misurabile secondo  
Lebesgue (con misura  $m$ )  $\Rightarrow$

$\forall \varepsilon, \exists$  una somma finita

$$G(x) = \sum_{j=1}^N \alpha_j \sigma(W_j^T X + \theta_j),$$

$\exists$  un insieme  $D \subset I_m$  :  $m(D) \geq 1 - \varepsilon$  e

$$|G(x) - f(x)| < \varepsilon \quad \forall x \in D$$

Conseguente: Una rete MLP può approssimare  
qualsunque funzione di decisione,  
cioè tale che ad ogni partizione  
 $V_1, \dots, V_p \subset I_m$  e misurabili  
secondo Lebesgue rimane  
associata la funzione  
 $f(x) = i$  sse  $x \in V_i \subset I_m$   
(Problema delle classi/zone)

## I teoremi di Cybenko non sono costruttivi

- non forniscono indicazioni sulla caratteristica della rete

## Estensioni dei teoremi di Cybenko

- K. Hornik, M. Stinchcombe, H. White, "Multilayer Feedforward Networks are Universal Approximators", Neural Networks, Vol 2, pp 358-366, 1989.

Si dimostra come reti con un solo strato nascosto con funzioni di attivazione arbitrarie sono approssimatori universali.

- M. Stinchcombe, H. White, "Approximating and learning Unknown Mappings using Multilayer Feedforward Networks with Bounded Weights", Proc. of ISCHN, San Diego CA, USA, 1990.

Si dimostra come reti MLP con pesi e soglie limitati (così pratico) siano ancora degli approssimatori universali.

## Rappresentazione

Si intende l'esatta ricostruzione di una funzione (o funzionale), con garanzia della massima precisione.

### Teorema di Kolmogorov (1957)

Per ogni  $n \geq 2$   $\exists$  funzioni di una variabile  $\psi_{pq}$  continue su  $C(I_M)$ , monotone crescenti, tali che ogni funzione reale continua  $f(x)$  definita su  $I_M$  è rappresentabile nella forma

$$f(x) = \sum_{q=1}^{2n+1} \phi_q \left( \sum_{p=1}^M \psi_{pq}(x_p) \right); \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

dove  $\phi_q$  sono funzioni reali continue.

- Non c'è alcun legame tra  $f(x)$  e  $\psi_{pq}$ .
- Questo schema sembra garantire la rappresentazione esatta di qualsiasi funzione con una rete a due strati interni.
- Si avanzano forti riserve sulle caratteristiche delle funzioni  $\phi_q$  e  $\psi_{pq}$

## Estensione del T. di Kolmogorov

Sprecher (1965) ha elaborato una versione alternativa di questo teorema come segue:

$$f(\mathbf{X}) = \sum_{q=1}^{2H+1} g \left( \sum_{p=1}^n \lambda_p \phi_q(x_p) \right)$$

dove  $g$  è una funzione unica,  $\lambda_p$  è uno scalare.

- L'estensione di Sprecher ha interesse nel campo delle Reti Neurali in quanto:
  1. si tratta di una esatta ricostruzione della  $f(x)$  e non una approssimazione;
  2. la topologia della rete è determinata.

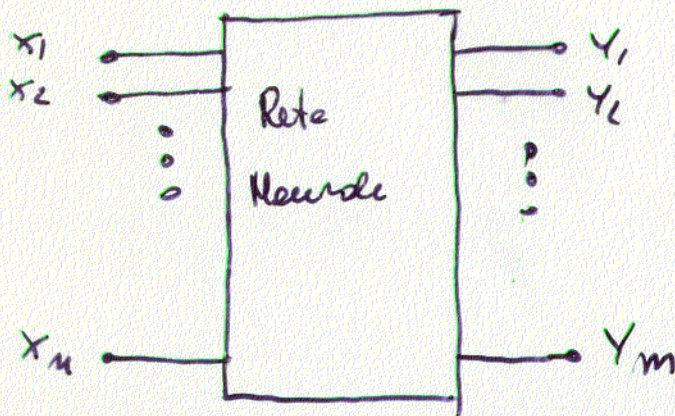
I Teoremi di Kolmogorov e Sprecher hanno solo interesse accademico in quanto non menzionano in alcun modo come le funzioni  $\phi_p, \psi_{pq}, g, \phi_q$  possano essere determinate.

## 4. LA SINTESI DELLE RETI NEURALI

- Hornik, Stinchcombe e White, hanno dimostrato, generalizzando il teorema di Stone-Weierstrass, come una RN con un solo strato nascosto con un adeguatamente grande numero di nodi (neuroni) può approssimare ogni funzione

$$f \in C(\mathbb{R}^n, \mathbb{R}^m)$$

in un sottoinsieme  $C(I_n) \subset \mathbb{R}^n$



$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m$$

- Per sintesi di Reti Neurali si intendono
  1. la determinazione della topologia
  2. la determinazione dei valori dei parametri liberi (pesi  $w$ , soglie  $\theta$ )

- la determinazione della topologia ottimale è un problema aperto.
- La determinazione dei pesi  $w$  e delle soglie  $\theta$ , è effettuata con metodi di ottimizzazione, di vario tipo, che in questo contesto sono chiamati:

## Processi di Apprendimento

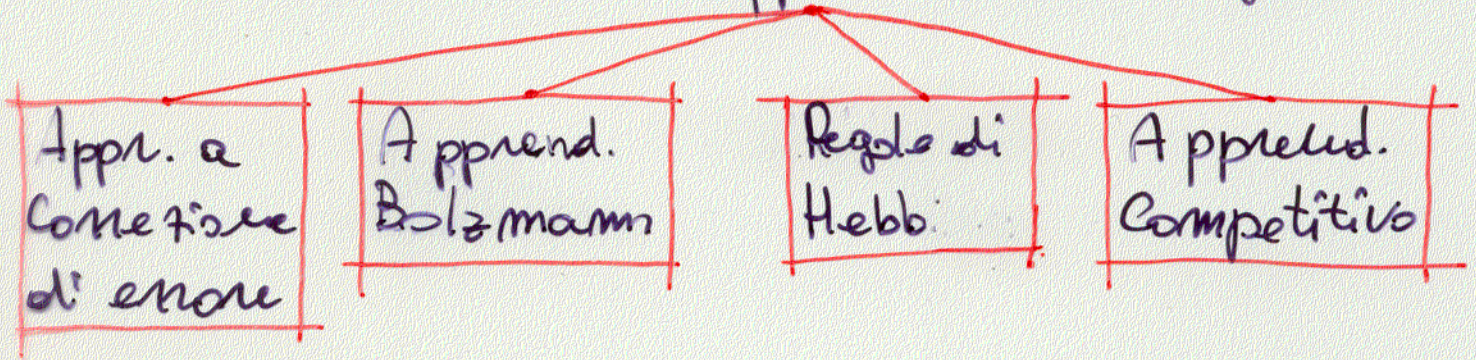
"L'apprendimento è un processo in cui i parametri liberi della rete neurale sono adattati attraverso una stimolazione continua dovuta all'ambiente circostante la rete."

1. la rete neurale è stimolata
2. i parametri interni sono cambiati  

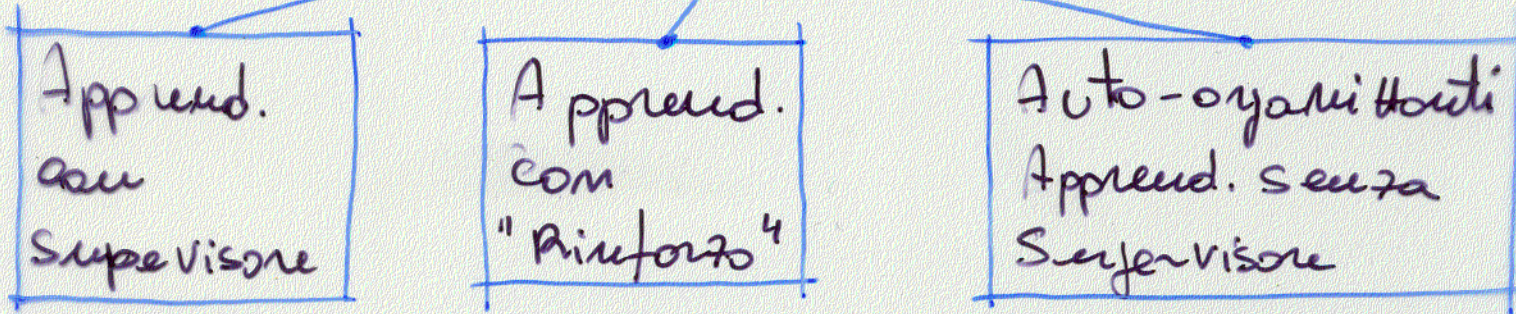
$$w(n+1) = w(n) + \Delta w(u)$$
3. la rete neurale risponde in modo diverso all'ambiente circostante.

# Tassonomia dei processi di apprendimento

## Algoritmi di apprendimento (regole)

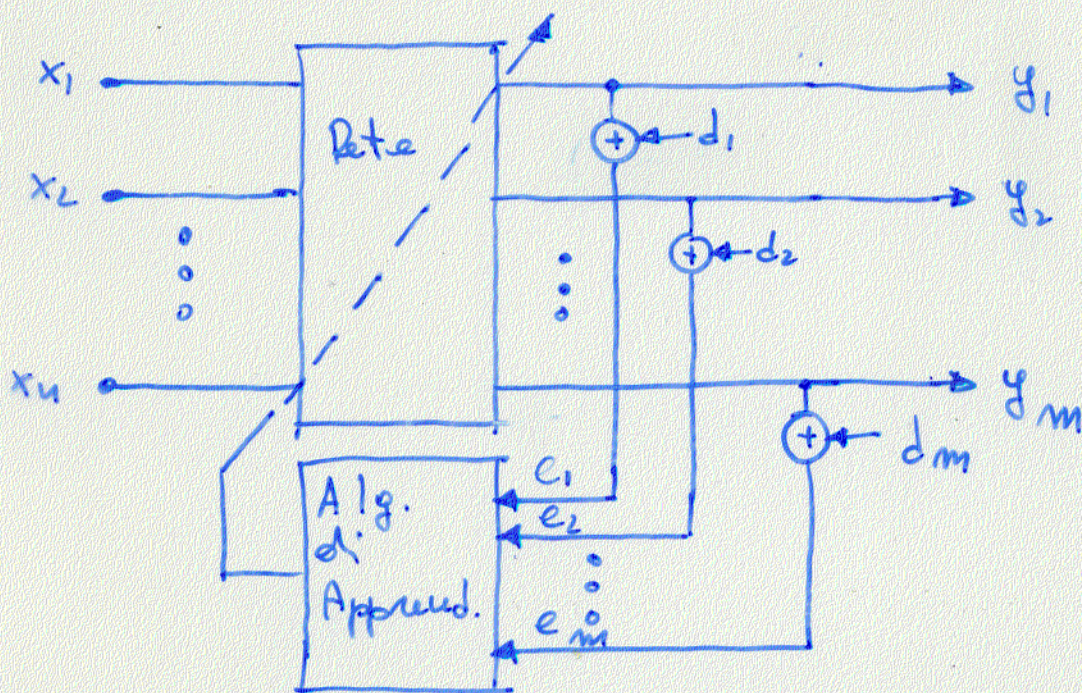


## Paradigmi di Apprendimento



# Apprendimento a Convezione di errore

## o Delta rule ( $\Delta$ -rule)



$x_k \in \mathbb{R}^m$  ingressi  
 $y_k \in \mathbb{R}^m$  uscite  
 $d_k \in \mathbb{R}^m$  uscite desiderate  
 $e_k \in \mathbb{R}^m$  errore

$\Delta$  Segnali di errore all'istante  $m$  per l'uscita  $k$ , come:

$$e_k(m) = d_k(m) - y_k(m)$$

Attraverso un criterio (generalmente il criterio di minimi quadrati) si minimizza il funzionale:

$$E(m) = E \left[ \frac{1}{2} \sum_{k=1}^M e_k^2(m) \right] \text{ in pratica si minimizza}$$

$E(\cdot) \triangleq$  aspettazione

$$E(u) = \frac{1}{2} \sum_{k=1}^M e_k^2(u)$$

L'aggiustamento  $\Delta w$  di un generico peso  $w$  all'istante  $n$ , è dato dalla equazione di Widrow-Hoff (1960)

$$\Delta w(n) = \eta e_n(n) \cdot x_j(n)$$

$\eta$   $\triangleq$  "guadagno" di apprendimento (learning rate)

il peso è aggiornato come:

$$w(n+1) = w(n) + \Delta w(n)$$

### Apprendimento con regola di Hebb (Hebbian Learning)

L'apprendimento è basato sul postulato di Hebb (1949) che tende a "premiare" i legami (pesi sinaptici) che ricevono più stimolazioni dall'esterno.

Matematicamente la Hebbian Learning rule è espresso come:

$$\Delta w(n) = F(y(n) \cdot x(n))$$

- È possibile dimostrare che la  $\Delta$ -rule è un caso particolare della Hebbian-rule.

# Apprendimento di Boltzmann

## (Boltzmann learning rule)

- L'apprendimento di Boltzmann, derivato dalla teoria dell'informazione e da considerazioni termodinamiche, è noto come "stochastic learning" e derivato dal paradigma della macchina di Boltzmann.
- La macchina di B. è caratterizzata da una funzione di energia  $E$  che è determinata dal particolare stato della macchina.
- Ogni stato della macchina è determinato dallo stato di ogni neurone

$$E = - \frac{1}{2} \sum_i \sum_{\substack{j \\ i \neq j}} w_{ij} s_i s_j$$

$w_{ij}$   $\triangleq$  peso sinaptico tra il neurone  $i$ - $j$ ;

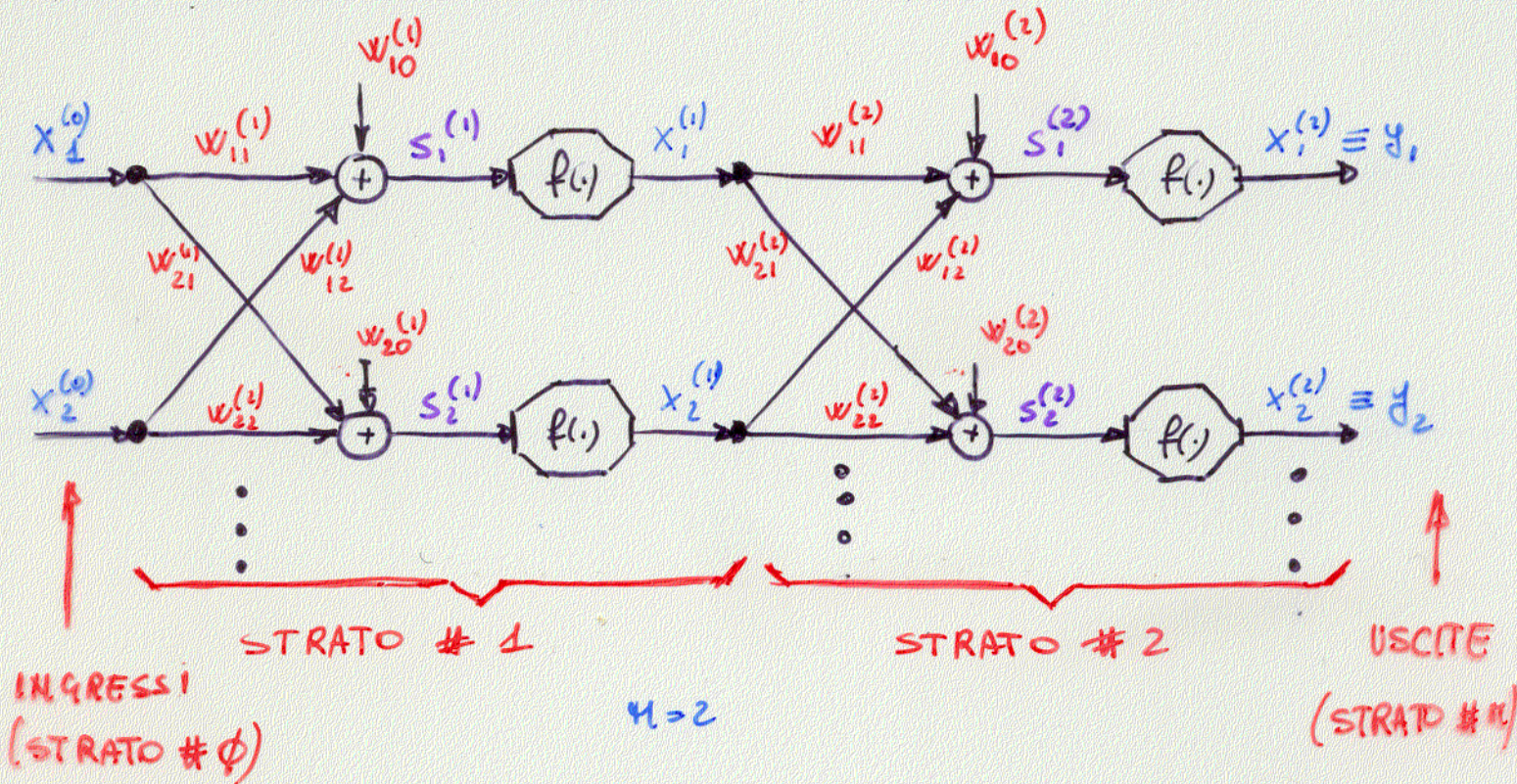
$s_i$   $\triangleq$  stato del neurone  $i$ ;

$s_j$   $\triangleq$  stato del neurone  $j$ ;

$$W(s_j \rightarrow s'_j) = \frac{1}{1 + e^{-\frac{\Delta E_j}{T}}}$$

- mosse casuali.

# Derivazione dell'algoritmo di Backpropagation per reti multistrato FEED FORWARD statiche



$M$  n° di strati (indice  $(e)$ )

$N_e$  n° di neuroni strato  $e$ -esimo

$N_0 \equiv$  n° ingressi  $N_M \equiv$  n° delle uscite

$s_k^{(e)}$  uscita del combinatorio lineare del neurone  $k$ -esimo strato  $e$ -esimo

$x_k^{(e)}$   $f(s_k^{(e)})$  uscita del neurone  $k$ -esimo strato  $e$ -esimo

$w_{kj}^{(e)}$   $\bar{e}$  il peso del neurone  $k$ -esimo dello strato  $e$ -esimo relativo all'uscita del  $j$ -esimo neurone dello strato  $(e-1)$ -esimo.  
 $w_{k0}^{(e)}$   $\bar{e}$  l'offset  $(\theta)$ .

$f(\cdot)$  funzione di attivazione per es:  $f(x) = \frac{1}{1 + e^{-x}}$

## Calcolo uscite (fase FORWARD)

•  $x_k^{(0)} = x_k \quad k = 1, 2, \dots, N_0$  (ingressi)

• 
$$\begin{cases} s_k^{(l)} = \sum_{j=0}^{N_{l-1}} w_{kj}^{(l)} x_j^{(l-1)} \\ x_k^{(l)} = f(s_k^{(l)}) \end{cases}$$

BIAS  $\rightarrow l = 0, \dots, M-1; x_0^{(l)} = 1$   
CALCOLO  $\rightarrow k = 1, \dots, N_l; l = 1, \dots, M$

•  $y_k = x_k^{(M)} \quad k = 1, 2, \dots, N_M$  (uscite)

## Apprendimento:

DATE:  $Q$  coppie  $\{x_k^{(l)} (k=1..N_0), d_k^{(l)} (k=1..N_M)\}$

$x_k \rightarrow$  ingressi

$d_k \rightarrow$  uscite desiderate (TARGET)

DETERMINARE: il valore dei pesi  $w_{kj}^{(l)}$

Minimizzando (in genere) il funzionale

$$\mathcal{E} = \sum_{m=0}^{Q-1} \sum_{k=1}^{N_M} (d_k^{(m)} - y_k^{(m)})^2 \quad \text{MSE}$$

• Problema non lineare  $\Rightarrow$   $\nexists$  soluzione determinabile in forma chiusa

• Algoritmo ricorsivo basato sulla discesa del gradiente tipo "STEEPEST DESCENT"

# in questo contesto $\Rightarrow$ BACK PROPAGATION

- derivazione equazioni:

$$E[u] = \sum_{k=1}^{NM} (d_k[u] - y_k[u])^2$$

MSE per la  $m$ -esima coppia  $(x_k[u], d_k[u])$

ometto "u" per semplicità

$$E = \sum_{k=1}^{NM} (d_k - y_k)^2 = \sum_{k=1}^{NM} (d_k - x_k^{(m)})^2$$

- $\min_{w \in W} E \Rightarrow$  calcol  $\frac{\partial (\cdot)}{\partial w_{jk}^{(e)}}$

## ULTIMO STRATO

$$\frac{\partial E}{\partial w_{kj}^{(m)}} = \frac{\partial}{\partial w_{kj}^{(m)}} \left[ \sum_{k=1}^{NM} (d_k - x_k^{(m)})^2 \right] = 2(d_k - x_k^{(m)}) \cdot \frac{\partial x_k^{(m)}}{\partial w_{kj}^{(m)}} =$$

$$= 2(d_k - x_k^{(m)}) \cdot \frac{\partial}{\partial w_{kj}^{(m)}} [f(s_k^{(m)})] = 2(d_k - x_k^{(m)}) \cdot f'(s_k^{(m)}) \cdot \frac{\partial s_k^{(m)}}{\partial w_{kj}^{(m)}}$$

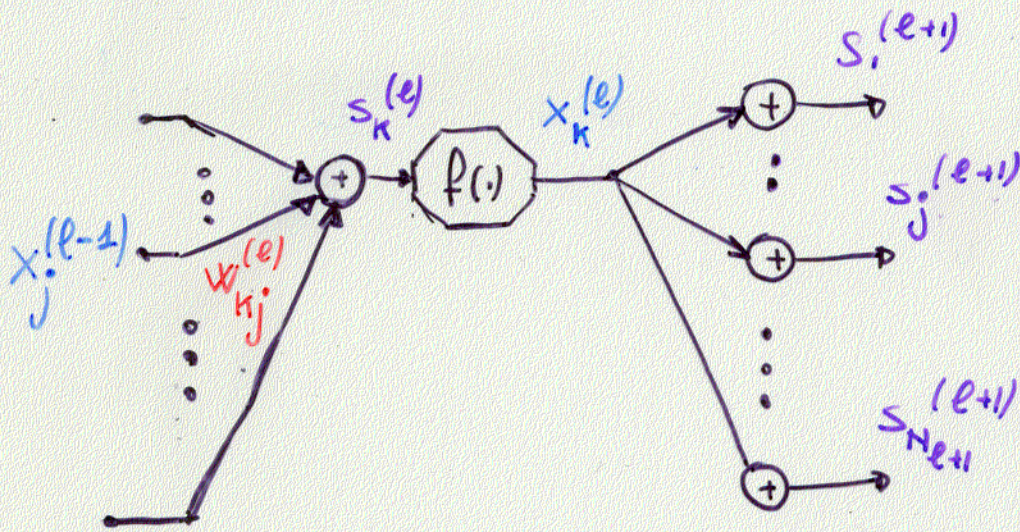
$$= \boxed{2(d_k - x_k^{(m)}) \cdot f'(s_k^{(m)}) \cdot x_j^{(m-1)}}$$

dove:

$$f'(s_k^{(m)}) \rightarrow \frac{\partial f(\cdot)}{\partial s_k^{(m)}}$$

$$\frac{\partial E}{\partial s_k^{(m)}} = 2(d_k - x_k^{(m)}) f'(s_k^{(m)})$$

# STRATI NASCOSTI



← STRATO #  $l$       ← STRATO #  $(l+1)$  →

$$\frac{\partial \mathcal{E}}{\partial w_{kj}^{(l)}} = \sum_{m=1}^{N_{e+1}} \frac{\partial \mathcal{E}}{\partial s_m^{(l+1)}} \cdot \frac{\partial s_m^{(l+1)}}{\partial x_k^{(l)}} \cdot \frac{\partial x_k^{(l)}}{\partial s_k^{(l)}} \cdot \frac{\partial s_k^{(l)}}{\partial w_{kj}^{(l)}} =$$

$$= \sum_{m=1}^{N_{e+1}} \frac{\partial \mathcal{E}}{\partial s_m^{(l+1)}} \cdot w_{mk}^{(l+1)} \cdot f'(s_k^{(l)}) \cdot x_j^{(l-1)}$$

$$\Delta e_k^{(l)} = \frac{\partial \mathcal{E}}{\partial x_k^{(l)}} ; \quad \delta_k^{(l)} = \frac{\partial \mathcal{E}}{\partial s_k^{(l)}}$$

## FASE BACKWARD

- $$e_k^{(l)} = \begin{cases} (d_k - y_k) & \text{per } l = M \\ \sum_{m=1}^{N_{e+1}} w_{mk}^{(l+1)} \delta_m^{(l+1)} & \text{per } l = (M-1), (M-2), \dots, 1 \end{cases}$$
- $$\delta_k^{(l)} = e_k^{(l)} \cdot f'(s_k^{(l)})$$
- $$w_{kj}^{(l)} = w_{kj}^{(l)} + \eta \delta_k^{(l)} \cdot x_j^{(l-1)} \quad \begin{matrix} k=1, \dots, N_e \\ j=0, \dots, N_{e-1} \end{matrix}$$

## ALGORITMO RICORSIVO

1.  $x_n^{(0)} \leftarrow x_k$  calcoli uscite con fase FORWARD

2. calcoli  $e_k^{(l)}$   $k = 1, \dots, M_l$  (ultimo stato)

3. FOR  $l = M-1$  TO  $1$

$$\delta_k^{(l)} \quad k = 1, \dots, N_l$$

$$e_k^{(l-1)} \quad k = 1, \dots, N_{l-1}$$

FOR  $k = 1$  TO  $N_l$

FOR  $j = 0$  TO  $N_{l-1}$

$$w_{kj}^{(l)} = w_{kj}^{(l)} + \eta \delta_k^{(l)} x_j^{(l-1)}$$

END;

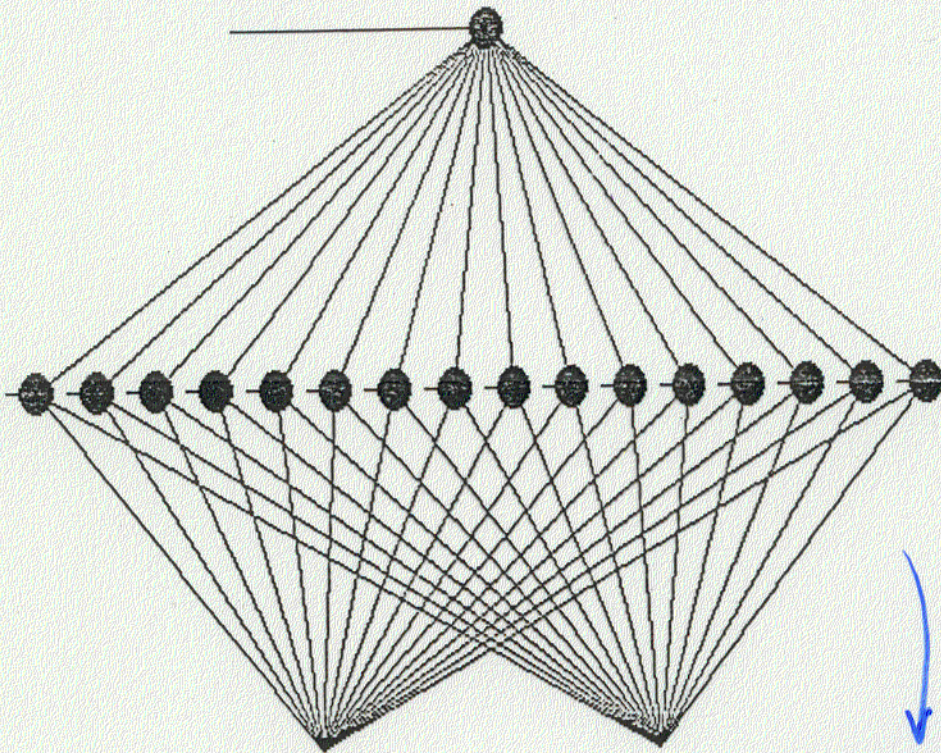
END;

END;

- pesi iniziali casuali con valori  $\pm 0.3$ ,  $\pm 0.5$
- l'aggiornamento è ripetuto ciclicamente utilizzando i pattern in-out.
- $\eta$  è scelto dall'utente

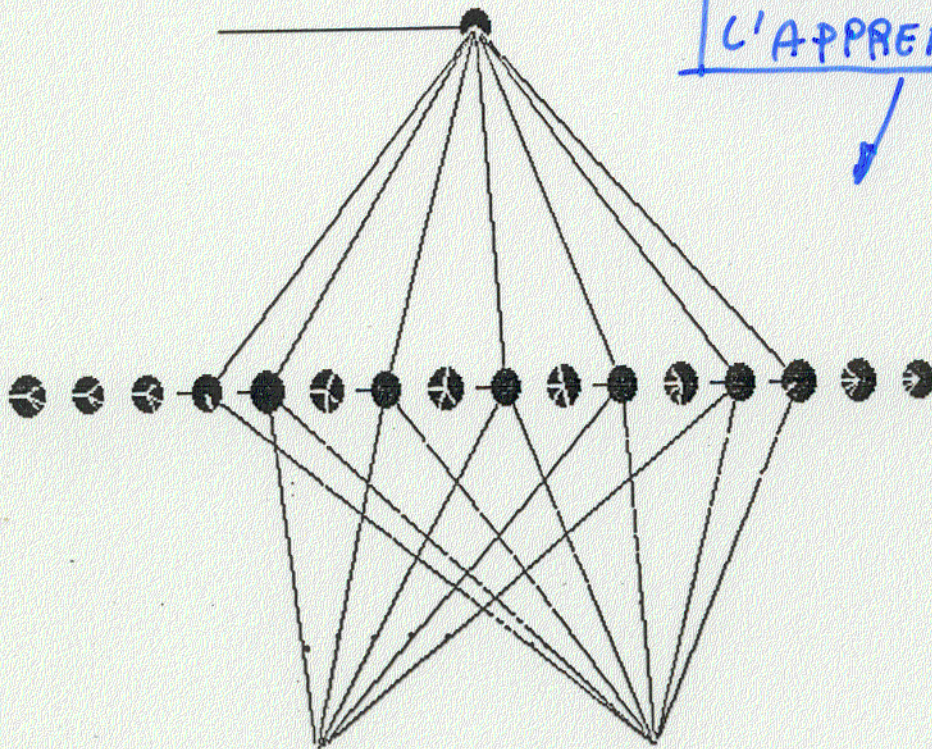
# RETI A TOPOLOGIA OTTIMALE

## RISULTATI SPERIMENTALI



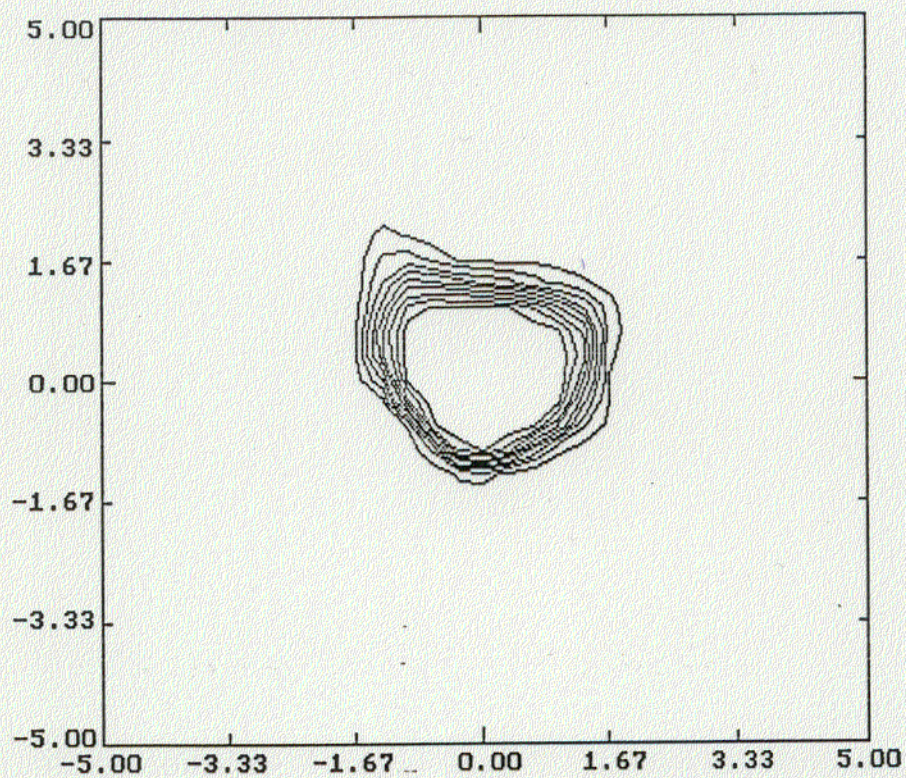
rete interamente connessa

VARIAZIONE DELLA  
TOPOLOGIA DURANTE  
L'APPRENDIMENTO

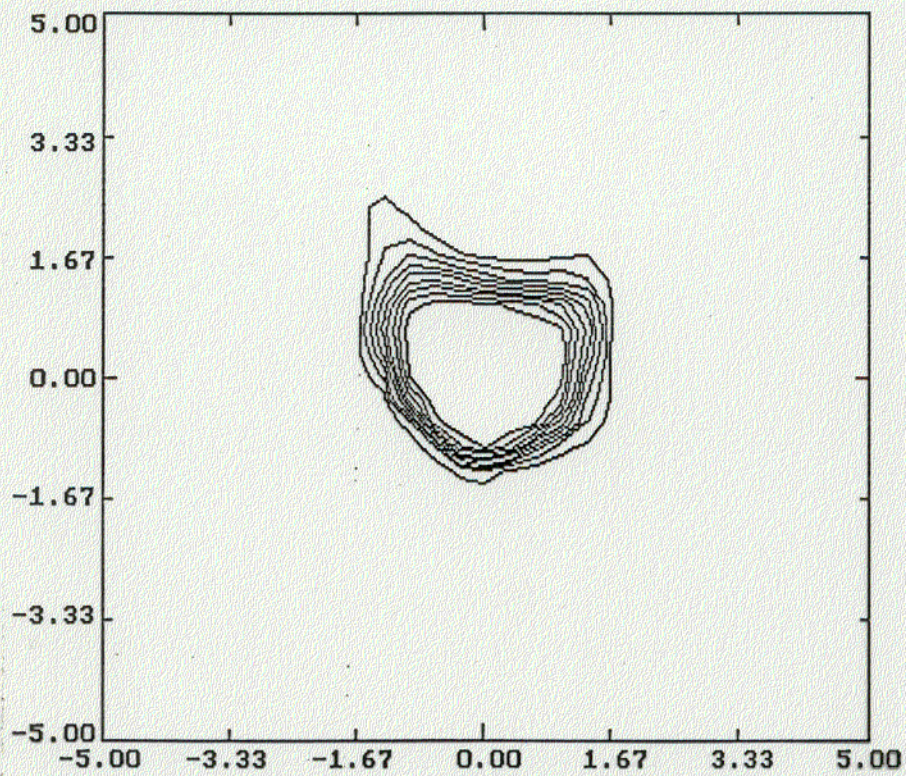


rete ridotta

# CAPACITA' DI GENERALIZZAZIONE



rete interamente connessa



rete ridotta

