

RETI NEURALI

Reti Neurali

Sono una nuova tecnologia informatica:

- architettura innovativa (non Von Neumann): molti semplici processori paralleli fortemente integrati da una rete di connessioni che realizzano un modello computazionale distribuito
- analogia con la struttura del cervello: molti neuroni (10^{10}) fortemente connessi da sinapsi attraverso cui le computazioni si diffondono in parallelo nella corteccia cerebrale

- nuove prestazioni ottenibili: tempo reale in problemi complessi, autoapprendimento, resistenza a guasti ed errori, degradazione graduale

Come funzionano le Reti Neurali?

Le Reti Neurali (dette anche Modelli Connessionisti) utilizzano una architettura elaborativa che presenta, nel trattamento delle informazioni, una serie di analogie con l'intelligenza naturale:

- forte parallelismo
- forte interconnessione
- rapida comunicazione
- semplici unità elaborative
- grande quantità di memoria (nelle connessioni)
- naturalmente associativa
- non programmata, ma addestrata mediante apprendimento automatico

Reti Neurali (RN) - Definizione

Una RN è un modello computazionale parallelo, costituito da numerose unità elaborative omogenee fortemente interconnesse mediante collegamenti di varia intensità.

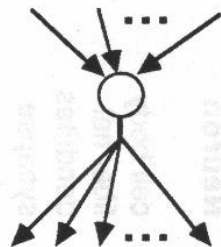
L'attività della singola unità è semplice (funzione di trasferimento) e la potenza del modello risiede nella configurazione delle connessioni (topologia e intensità).

Partendo dalle unità di input, a cui vengono forniti i dati del problema da risolvere, la computazione si propaga in parallelo nella rete fino alle unità di output, che forniscono il risultato.

Una RN non viene programmata per eseguire una certa attività, ma addestrata (utilizzando un algoritmo di apprendimento automatico) mediante una serie di esempi della realtà da modellare.

**Singola
Unita'
elaborativa**

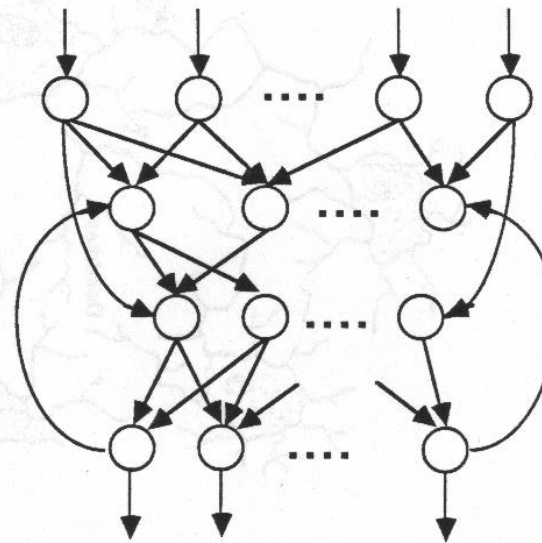
INPUTS



OUTPUTS

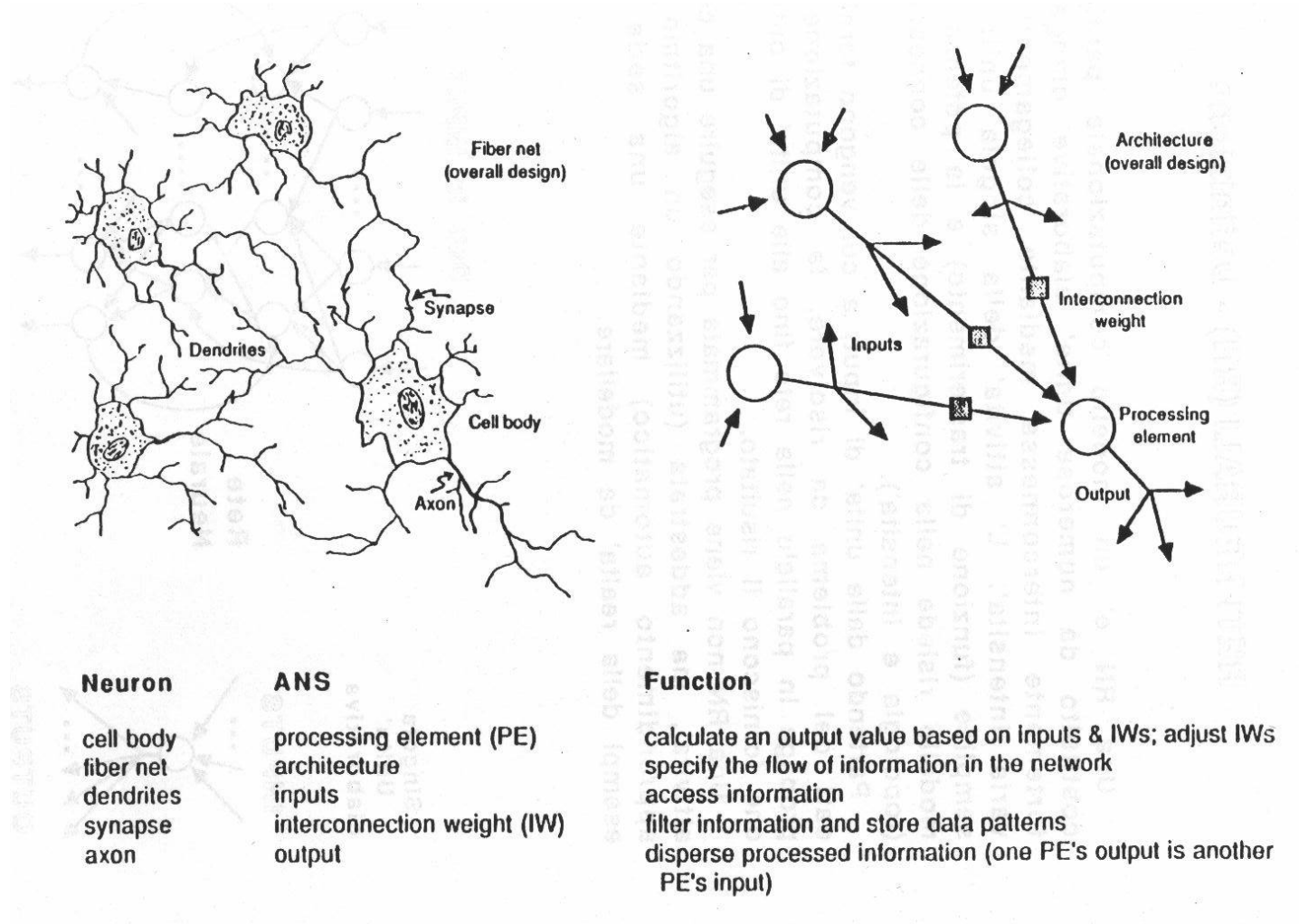
**Rete
Neurale**

Dati in INPUT



RISULTATI in OUTPUT

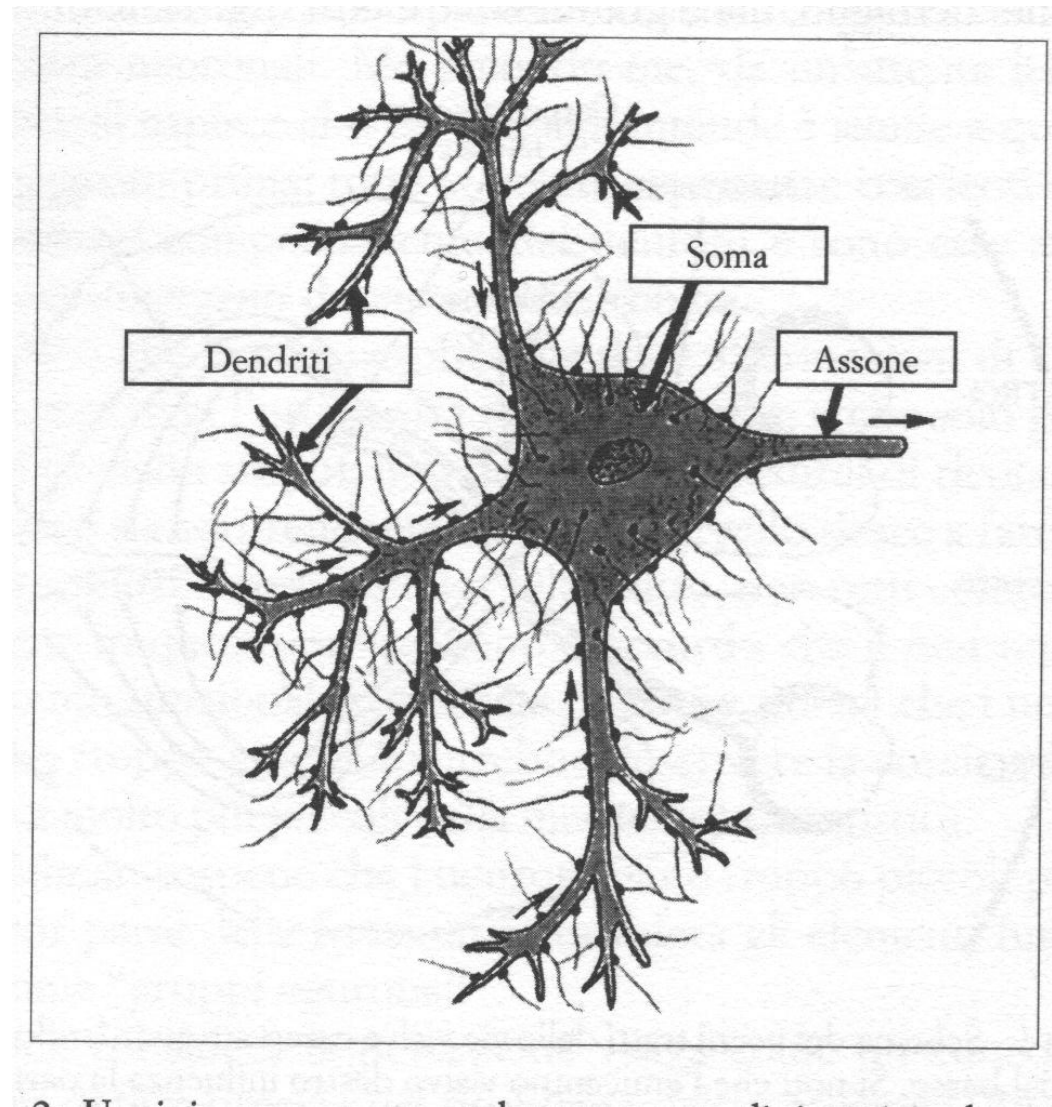
Analogia con la struttura cerebrale



Come funziona un neurone biologico

Il neurone è una cellula come qualsiasi altra, con una membrana ed un nucleo centrale, ma si differenzia notevolmente sia anatomicamente che fisiologicamente.

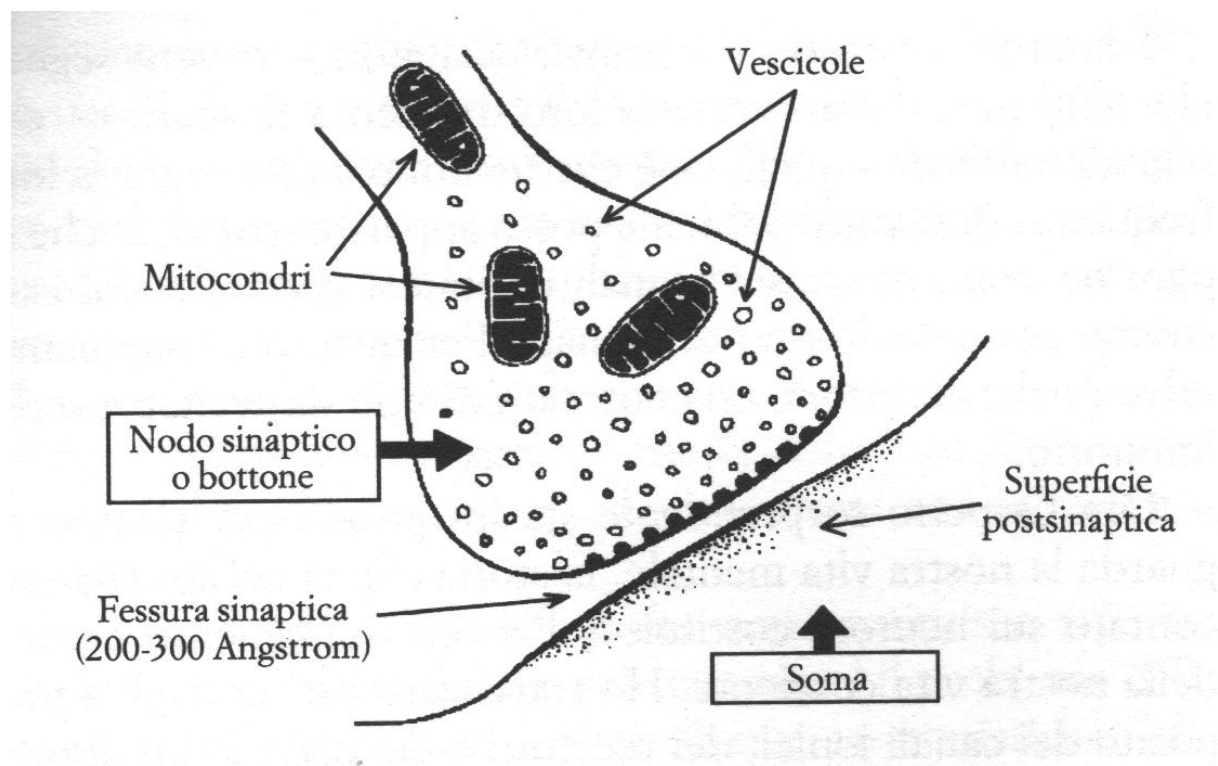
Il neurone più "tipico" (*garden-variety neuron*) presenta da un lato una protuberanza piuttosto lunga simile ad un filo chiamata **assone** e dall'altro lato una serie di fili ramificati più corti, spinosi e aguzzi chiamati **dendriti**.



Ciascun neurone riceve segnali attraverso i suoi dendriti, li elabora nel corpo cellulare o **soma** e poi lancia un segnale tramite l'assone al neurone successivo.

Il neurone si eccita mandando un impulso elettrico lungo l'assone.

L'assone di un neurone non è direttamente collegato ai dendriti di altri neuroni: il punto in cui il segnale viene trasmesso da una cellula ad un'altra è un piccolo spazio denominato **fessura sinaptica**.



Una sinapsi è formata caratteristicamente da una protuberanza sull'assone chiamata "**bottone**" o "**nodo sinaptico**" che sporge a forma di fungo e che si incastra con una prominenza a forma di spina dorsale sulla superficie del dendrite.

L'area tra il bottone e la superficie dendritica post-sinaptica è chiamata fessura sinaptica ed è attraverso di essa che il neurone eccitato trasmette il segnale.

Il segnale viene trasmesso non da una connessione elettrica diretta tra il bottone e la superficie dendritica, ma dall'emissione di una piccola quantità di fluidi chiamati **neurotrasmettitori**.

Quando il segnale elettrico si muove dal corpo cellulare lungo l'assone fino alla fine del bottone, questo provoca l'emissione di fluidi neurotrasmettitori nella fessura sinaptica.

Questi ultimi entrano in contatto con dei recettori posti sul lato dendritico post-sinaptico. Ciò causa l'apertura dei canali e gli ioni - atomi e gruppi di atomi carichi elettricamente - entrano ed escono dal lato dendritico, alterando così la carica elettrica del dendrite.

Lo schema è dunque il seguente: vi è un segnale elettrico sul lato dell'assone seguito da una trasmissione chimica nella fessura sinaptica, seguito da un segnale elettrico sul lato del dendrite.

La cellula riceve un'intera serie di segnali di suoi dendriti, li somma all'interno del suo corpo cellulare e, sulla base della somma, aggiusta la frequenza delle scariche da inviare alla cellula successiva.

I neuroni ricevono sia segnali eccitatori - ovvero segnali che tendono ad aumentare la loro frequenza di scarica - che segnali inibitori - quelli cioè che tendono a diminuire la loro frequenza di scarica.

Anche se ogni neurone riceve sia segnali eccitatori che inibitori, esso emette poi un solo tipo di segnale.

Per quanto si sa, salvo poche eccezioni un neurone è di tipo sia eccitatorio che in inibitorio.

La struttura e il funzionamento dei neuroni costituiscono l'intero fondamento causale della nostra vita cosciente.

Sistemi HW di Reti Neurali (6^a gen.) e computer convenzionali (4-5^a gen.)

Le RN possono essere simulate in software su computers tradizionali o implementate in hardware dando origine ad una nuova generazione di strumenti di calcolo.

L'aspetto che distingue nettamente un sistema di Reti Neurali dai computers tradizionali è l'architettura sui cui sono basati.

- L'architettura su cui si basano i computers attuali risale a John Von Neumann (1940) e prevede una unità operativa centrale (CPU) e una memoria, collegate da un canale di transito dati.

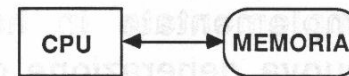
Questa architettura ha avuto grande successo portando allo sviluppo attuale dell'informatica, ma è intrinsecamente sequenziale e impiega una notevole percentuale di tempo del processore per accedere alla memoria.

- L'approccio a Rete Neurale evita il collo di bottiglia dell'accesso alla memoria ed è intrinsecamente parallelo: infatti integra *memorizzazione* (nelle connessioni) e *computazione* (in tante unità elaborative autonome), in analogia con l'intelligenza biologica.

Rapporto Memoria/Elaborazione in alcuni sistemi di calcolo

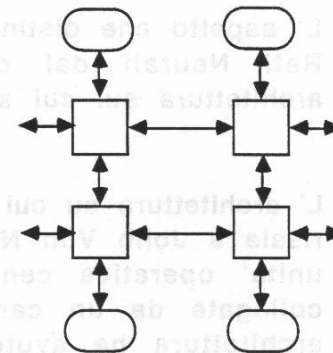
COMPUTER SEQUENZIALE
(Von Neumann)

MEMORIA ESTERNA



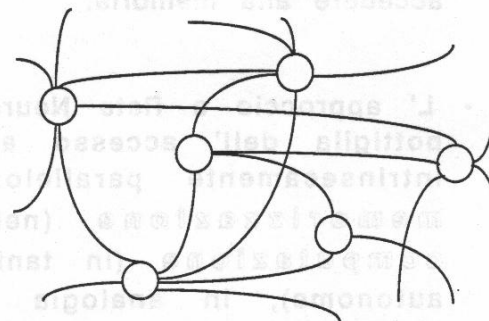
MULTIPROCESSORE

MEMORIA INTERNA/ESTERNA

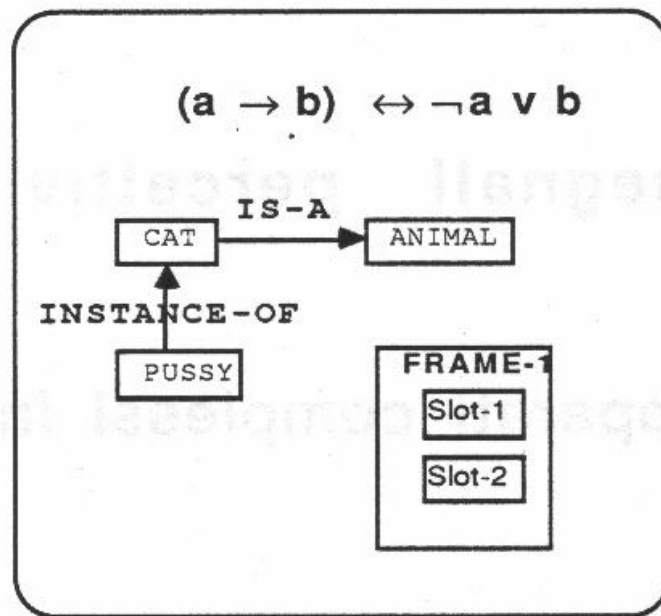


RETI NEURALI

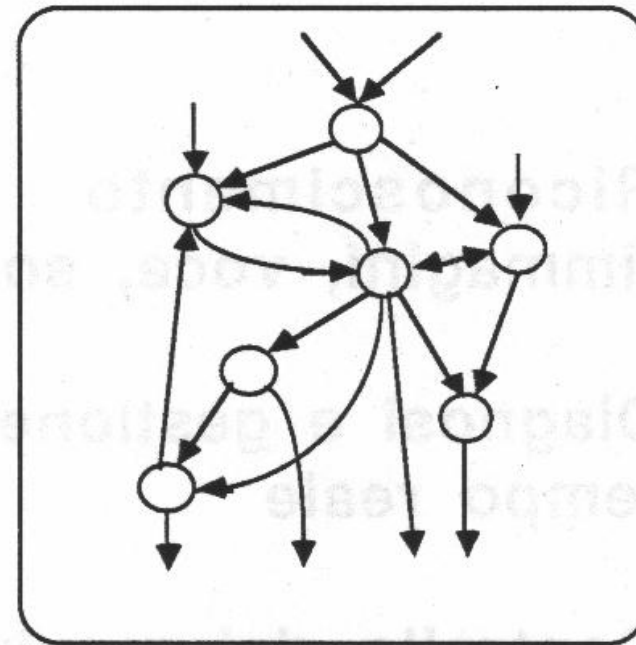
MEMORIA INTERNA
(nelle CONNESSIONI)



Reti Neurali e Intelligenza Artificiale



IA simbolica



IA neurale

L'Intelligenza Artificiale rende più utili i calcolatori costruendo sistemi che mostrano comportamenti intelligenti e risolvono problemi complessi.

- In tale ambito, le tecniche NEURALI sono l'alter-ego delle tecniche SIMBOLICHE basate su conoscenza esplicita fornita da un esperto:
sono infatti basate su conoscenza autoappresa e implicitamente codificata in una complessa rete di connessioni, in analogia con l'intelligenza biologica;

- possono essere una valida alternativa alle tecniche simboliche nella risoluzione approssimata di certi problemi complessi richiedenti un alto grado di parallelismo (esempi: visione artificiale, riconoscimento vocale, veicoli autonomi, robot, diagnosi e gestione in tempo reale di sistemi complessi, etc.)

In sintesi, quindi, le RN rientrano a pieno titolo fra le tecniche di IA, costituendo il naturale complemento dei metodi simbolici, con cui dovranno integrarsi nei futuri sistemi intelligenti.

All'interno dell'IA si collocano nel settore dell'Apprendimento Automatico, affiancandosi ai metodi di apprendimento simbolico.

Applicazioni delle reti Neurali

- Riconoscimento di segnali percettivi (immagini, voce, sonar)
- Diagnosi e gestione di apparati complessi in tempo reale
- Controllo del movimento di robot e veicoli autonomi
- Classificazione ed interpretazione di dati rumorosi
- Memoria associativa: accesso in tempo reale a grandi quantità di dati
- Ricostruzione di informazioni parziali o corrotte da rumore
- Soluzioni approssimate in tempo reale di problemi computazionalmente intrattabili

Breve storia dei modelli neurali

- 1943.

Il primo modello di neurone viene proposto da McCulloch e Pitts. Il "nucleo pensante" è rappresentato da una porta logica che può assumere due soli stati interni, "*acceso*" o "*spento*".

Ciascun neurone è dotato di un insieme di ingressi attraverso i quali riceve gli stimoli dall'ambiente circostante e/o i responsi prodotti dai neuroni adiacenti. La somma pesata di tali impulsi, confrontata con un'opportuna soglia interna al neurone, ne determina lo stato finale ed il conseguente segnale in uscita.

- 1950-60.

Obiezione di Von Neumann: i singoli neuroni mancano della robustezza, garantita dalla ridondanza dell'informazione spartita da più neuroni appartenenti ad un "*pool*".

Cragg e Temperly (1954) adottano il concetto di "*insieme di neuroni*" dando a questa entità una chiara connotazione fisica in termini del comportamento collettivo delle cellule aggregate.

- 1960.

Nasce il perceptron ad opera di Frank Rosenblatt.

- 1969-1985.

Gli anni della crisi, si aprono con la pubblicazione di "*Perceptrons*", di Minsky e Papert, che evidenzia i limiti fondamentali delle architetture a cella singola e ne delinea chiaramente la ristrettezza del campo applicativo.

Tuttavia, studio delle **memorie autoassociative** (Kohonen, Amari, Fukushima).

- 1986-oggi.

Rumelhart, Hinton e Williams (1986) descrivono l'algoritmo Back-Propagation che apre la strada allo studio ed alle applicazioni delle reti "*feedforward*" multistrato.

Nascono le reti "*ricorrenti*" che simulano il comportamento dei sistemi dinamici.

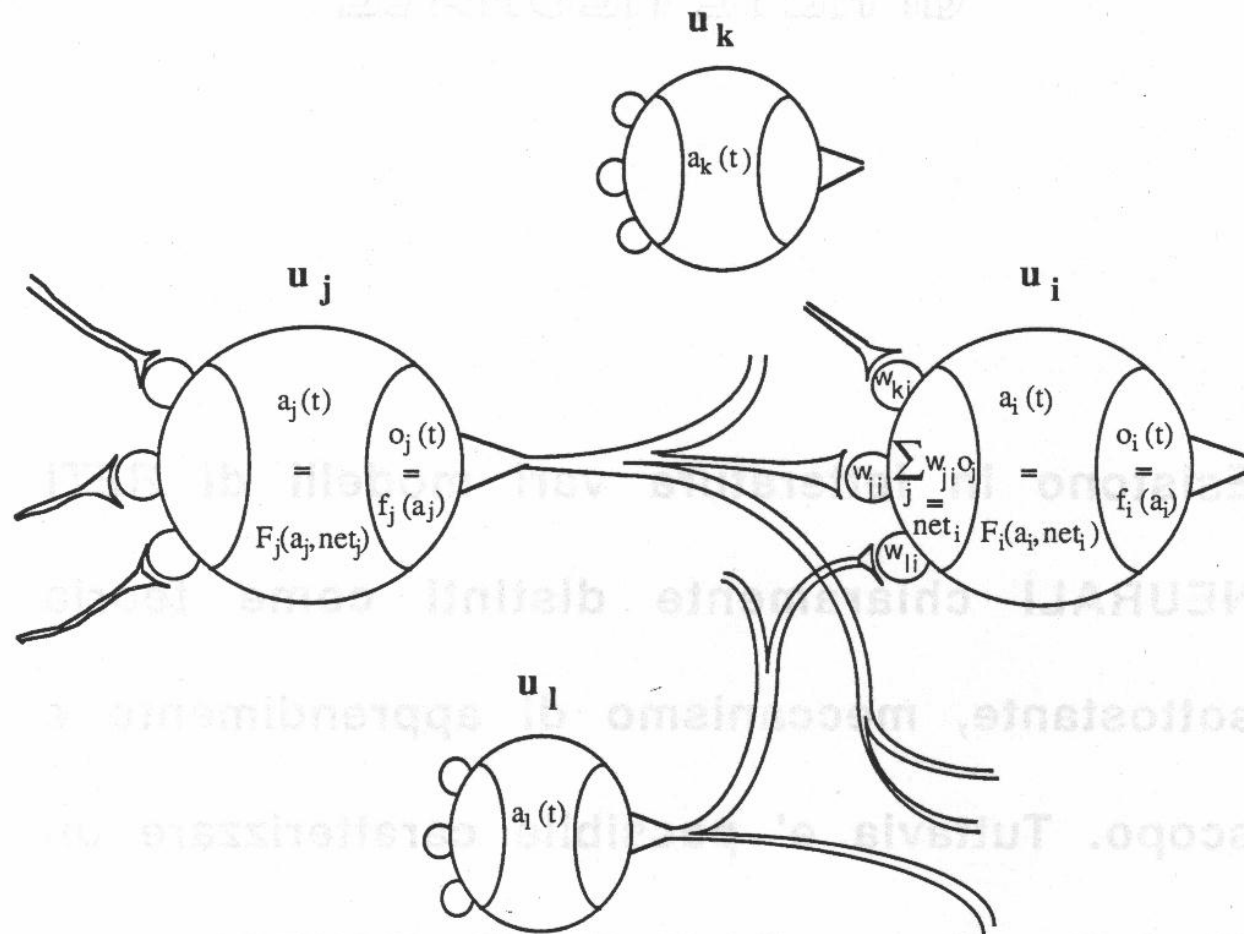
Le applicazioni in campo scientifico e industriale delle architetture neurali sono sempre più numerose e significative.

Un Modello Generale di Rete Neurale

Esistono in letteratura vari modelli di RETI NEURALI chiaramente distinti come teoria sottostante, meccanismo di apprendimento e scopo.

Tuttavia è possibile caratterizzare un Modello Generale di Rete Neurale che riassume le principali caratteristiche comuni.

Componenti di Base di un Modello Generale di rete Neurale



dove:

- $a_j(t)$ valore di attivazione corrente dell'unità elaborativa j-esima
- net_j input
- $F_j(a_j, net_j)$ funzione di attivazione
- $o_j(t)$ output
- $f_j(a_j)$ funzione di output
- w_{ji} peso sulla connessione dall'unità j-esima alla i-esima.

Componenti del Modello Generale di rete Neurale

- un insieme di *unità elaborative*
- uno *stato di attivazione*
- una *funzione di output* f_i per ogni unità
- le *connessioni* fra le unità
- una *regola di propagazione* per propagare i valori di output attraverso la rete di connessioni
- una *funzione di attivazione* F_i per combinare gli input con il valore di attivazione corrente e produrre un nuovo livello di attivazione
- una *regola di apprendimento* per modificare le connessioni

Unità Elaborative

- le unità elaborative sono semplici e uniformi
- l'attività di una unità consiste nel ricevere un input da un insieme di unità (connesse in input) e calcolare un valore di output da inviare ad un altro insieme di unità (connesse in output)
- il sistema è inerentemente parallelo poiché molte unità possono effettuare la loro computazione in modo concomitante
- generalmente si distinguono tre tipi di unità:
 - unità di input
 - unità di output
 - unità interne (Hidden)

Connessioni

Le unità sono connesse in una struttura a rete (digrafo).

È definita una matrice di connessioni \mathbf{W} in cui ogni connessione w_{ji} è caratterizzata da:

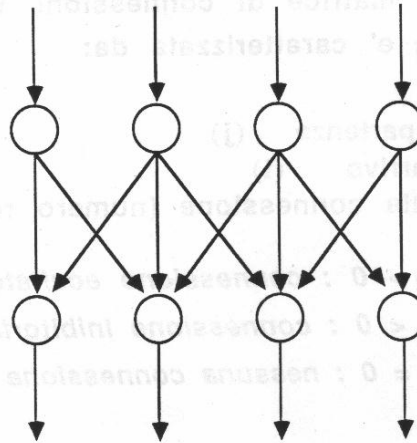
- unità di partenza (j)
- unità di arrivo (i)
- forza della connessione (numero reale)
 - $w_{ji} > 0$ connessione eccitatoria
 - $w_{ji} < 0$ connessione inibitoria
 - $w_{ji} = 0$ nessuna connessione

L'insieme delle connessioni costituisce la conoscenza della rete neurale e determina la sua risposta agli input:

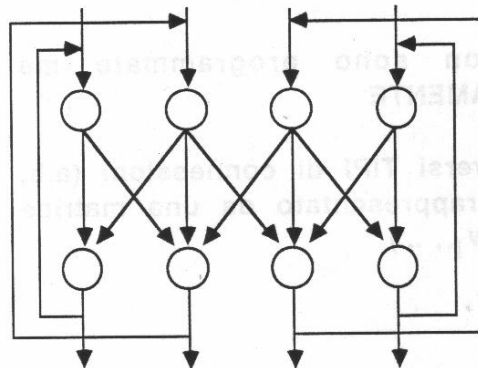
- la MEMORIA risiede nelle connessioni
- le CONNESSIONI non sono programmate ma APPRESE AUTOMATICAMENTE
- vi possono essere diversi TIPI di connessioni (a, b, ...), ognuno dei quali rappresentato da una matrice di connessione (\mathbf{w}_a , \mathbf{w}_b , ...)

Connessioni: Tipi di Architetture

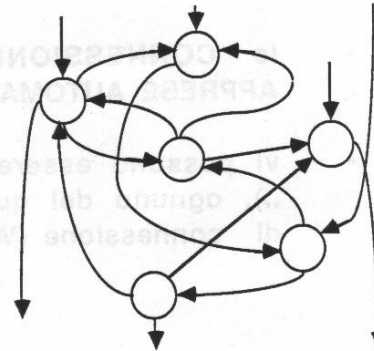
Feedforward



Feedback
(ordered)



Feedback
(amorphous)



Stato di attivazione della rete

Ogni unità u_i è caratterizzata da un valore di attivazione al tempo t : $a_i(t)$.

Modelli diversi fanno assunzioni diverse circa i valori di attivazione che le unità possono assumere:

- valori discreti:
 - o binari (attivo/inattivo) es.: $\{0,1\}$, $\{+1,-1\}$
 - o insieme ristretto es.: $\{-1,0,+1\}$, $\{1,2, \dots 9\}$
- valori continui:
 - o limitati es.: $[0,1]$
 - o illimitati $[-\infty, +\infty]$

Output delle unità elaborative

Le unità interagiscono trasmettendo segnali ai loro vicini (unità connesse).

La forza dei loro segnali, e quindi il grado di influenza esercitata sui vicini dipende dal loro grado di attivazione.

Ogni unità u_i ha associata una funzione di output $f_i(a_i(t))$ che trasforma il valore corrente di attivazione $a_i(t)$ nel segnale in output $o_i(t)$:

$$o_i(t) = f_i(a_i(t))$$

- In alcuni modelli il valore di output è uguale a quello di attivazione: $f(x) = x$;
- In altri f è una funzione a soglia, cioè un'unità non influenza le altre se la sua attivazione non supera un certo valore;
- In altri ancora f è una funzione stocastica del valore di attivazione.

Regola di propagazione

- È la modalità con cui gli output delle unità $o(t)$ sono propagati attraverso le connessioni della rete per produrre i nuovi input.
- Se vi sono più tipi di connessioni diverse (es.: eccitatorie, inibitorie, ...) la propagazione avviene indipendentemente per ogni tipo di connessione.
- Denominiamo $net_{ai}(t)$ l'input di tipo a dell'unità i -esima.

Spesso questa regola prevede semplicemente la somma degli input all'unità pesati dall'intensità delle connessioni:

$$net_{ai} = \sum_j w_{aji} o_j$$

o, in forma vettoriale:

$$\mathbf{net}_a = \mathbf{w}_a \mathbf{o}$$

Funzione di attivazione

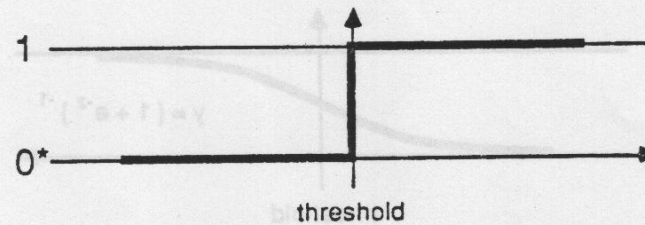
È una funzione, F , che prende il valore corrente di attivazione $a(t)$ e il vettore net_k di input all'unità ($k =$ tipo di connessione) e produce un nuovo valore di attivazione.

$$a(t+1) = F(a(t), net_1(t), net_2(t), \dots)$$

In genere F è:

- una funzione a soglia
- una funzione quasi-lineare (es. sigmoide)
- una funzione stocastica

Binary threshold: $B[z]$



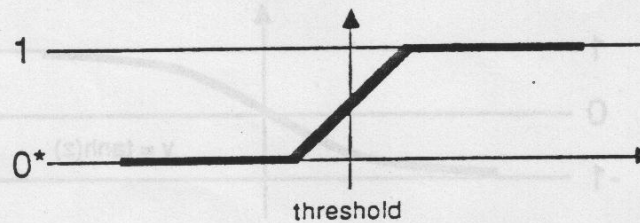
Advantages

- Fast and easy to use, especially in hardware
- Makes hard decisions

Disadvantages

- Not invertible
- No linear zone, can not smoothly imitate functions

Linear Ramp: $R[z]$



Advantages

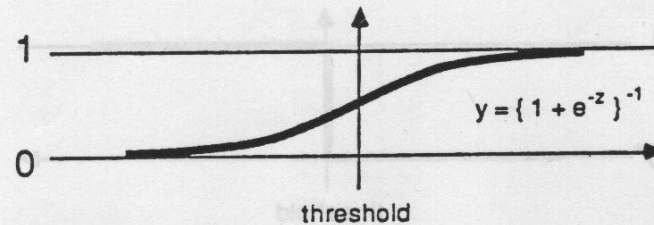
- Has a linear zone, can imitate functions
- Relatively easy to implement

Disadvantage

- Not invertible

* May be -1 instead of 0.

Sigmoid: $S[z]$



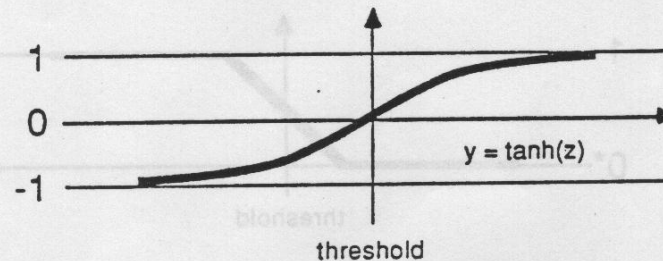
Advantages

- Invertible and continuously differentiable
- Semilinear zone for function imitation
- Can make "soft" or fuzzy decisions
- Biologically relevant--similar to neuron firing rate curve

Disadvantages

- More difficult to implement than piecewise linear functions
- Output limited to positive values

Hyperbolic Tangent: $\tanh[z]$



Advantages

- Invertible and continuously differentiable
- Semilinear zone for function imitation
- Can make "soft" or fuzzy decisions
- Can produce positive or negative outputs

Disadvantage

- More difficult to implement than piecewise linear functions

Apprendimento come modifica delle connessioni della rete neurale

Le RETI NEURALI non sono programmate da un esperto umano ma si autodefiniscono mediante *apprendimento automatico*.

L' apprendimento in una RN consiste nella modifica delle connessioni effettuata mediante una regola di apprendimento.

vi possono essere tre tipi di modifiche:

- 1) la **creazione** di nuove connessioni
- 2) la **perdita** di connessioni esistenti
- 3) la **modifica** della **forza** di connessioni esistenti

ma i casi (1) e (2) possono essere visti come
casi particolari del caso (3)

Regole di apprendimento

Quasi tutte le regole di Apprendimento possono considerarsi varianti della regola di Hebb (1949), la cui idea di base è:

“se un'unità u_i riceve un input da una unità u_j e se entrambi sono fortemente attive, il peso w_{ji} , da u_j a u_i deve essere rafforzato”

che può essere estesa nella regola:

$$\Delta w_{ji} = g(a_i(t), d_i(t)) \cdot h(o_j(t), w_{ji})$$

che dice che il cambiamento della connessione da u_j a u_i è dato dal prodotto di una funzione g (dell'attivazione di u_i e di un suo *input di addestramento* d_i) e di una funzione h (dell'output di u_j e della forza della connessione w_{ji}).

Nella versione più semplice della regola generale di apprendimento non vi è alcun *maestro* ($d_i = 0$) e g e h sono semplicemente funzioni lineari dei loro primi argomenti, per cui ci si riconduce alla *regola di Hebb classica*:

$$\Delta w_{ji} = \eta a_i o_j$$

dove η rappresenta la velocità di apprendimento.

Un'importante variante è la Delta rule, così detta perché l'apprendimento è proporzionale alla *differenza (o delta)* fra il valore corrente di attivazione e quello desiderato, fornito dal *maestro*:

$$\Delta w_{ji} = g(d_i(t) - a_i(t)) \cdot o_j(t)$$

che è una generalizzazione della regola del *perceptron* (Roseblatt, 1962).

Modelli di RN:

- o Associatore lineare
- o Hopfield nets
- o Perceptron
- o Multi Layer Perceptron
- o Kohonen self-organizing maps
- o Boltzmann Machine (qui non trattate)

Tipi di apprendimento e di operazione

APPRENDIMENTO:

- NON ITERATIVO (Fixed Point):
 - ❖ associatore lineare
 - ❖ Hopfield nets
- ITERATIVO:
 - ❖ perceptron, MLP
 - ❖ Kohonen maps
 - ❖ Boltzmann

OPERAZIONE I/O:

- PROPAGAZIONE VALORI:
 - ❖ associatore lineare
 - ❖ perceptron, MLP
 - ❖ Kohonen maps
- RILASSAMENTO A UN MINIMO DELL'ENERGIA:
 - ❖ Hopfield nets
 - ❖ Boltzmann

Operazioni tra matrici (promemoria)

$$\begin{matrix} \mathbf{W} & = & \mathbf{U} & \times & \mathbf{V} \\ (n \times m) & & (n \times h) & & (h \times m) \end{matrix}$$

$$W_{ij} = \sum_{k=1}^h U_{ik} V_{kj}$$

l'elemento della matrice risultato è dato dalla somma dei prodotti degli elementi della i-esima riga per i corrispondenti elementi della j-esima colonna

$$\mathbf{W}\mathbf{v} = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 0 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} =$$

$$\begin{bmatrix} 1 * 3 + 1 * (-1) + 2 * 2 \\ 2 * 3 + 0 * (-1) + (-1) * 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

Prodotto interno e prodotto esterno di matrici

$$v = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} \quad u = \begin{bmatrix} 0 \\ 4 \\ 1 \end{bmatrix}$$

PRODOTTO INTERNO (inner product):

$$\mathbf{v}^T \mathbf{u} = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \end{bmatrix} = 6$$

PRODOTTO ESTERNO (outer product):

$$\mathbf{uv}^T = \begin{bmatrix} 0 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 12 & 4 & 8 \\ 3 & 1 & 2 \end{bmatrix}$$

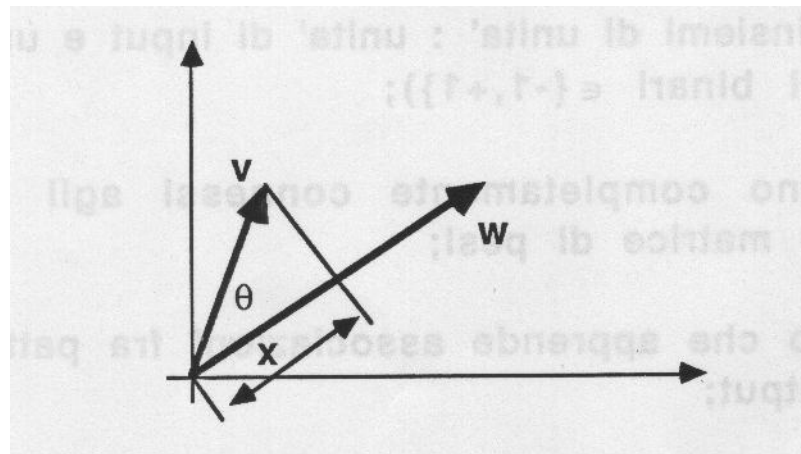
Prodotto interno di vettori

$$v = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} \quad w = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

$$\mathbf{v}^T \mathbf{w} = \begin{bmatrix} 3 & -1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} =$$

$$(3 * 1) + (-1 * 2) + (2 * 1) = 3$$

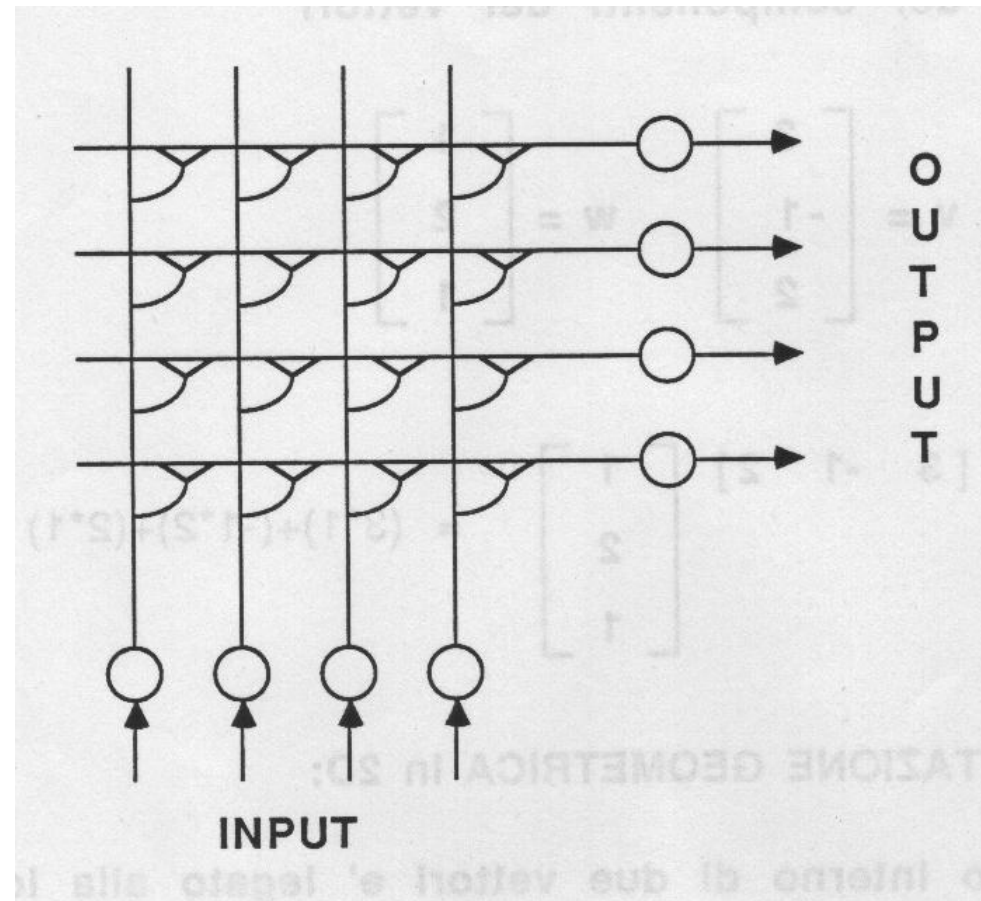
Interpretazione geometrica in 2D:
Il prodotto interno di due vettori è legato alla loro proiezione,
e dà una misura della somiglianza dei vettori.



$$\mathbf{v}^T \mathbf{w} = \|\mathbf{v}\| \cdot \|\mathbf{w}\| \cdot \cos \Theta$$

$$x = \|\mathbf{v}\| \cdot \cos \Theta = \frac{\mathbf{v}^T \mathbf{w}}{\|\mathbf{w}\|}$$

Associatore Lineare



- vi sono due insiemi di unità: unità di input e unità di output (valori binari $\in \{-1, +1\}$);
- gli input sono completamente connessi agli output mediante una matrice di pesi;
- è un modello che apprende associazioni fra pattern di input e di output;
- apprende con la regola di Hebb (diretta interpretazione con l'algebra lineare) o con la delta rule;
- serve per CLASSIFICAZIONE o MEMORIA ASSOCIATIVA.

Associatore Lineare (PA) addestrato con la Hebb Rule

Memorizziamo una associazione fra un pattern di input \mathbf{i}_p e un pattern di output \mathbf{o}_p .

Seguendo la Hebb Rule, definiamo i pesi così:

$$w_{ji} = \varepsilon \cdot o_{ip} \cdot i_{jp}$$

ovvero, in forma vettoriale:

$$\mathbf{W} = \varepsilon \cdot \mathbf{o}_p \mathbf{i}_p^T$$

Quindi, il PA apprende i pesi di una associazione mediante il prodotto esterno dei vettori da associare

poi, presentiamo un pattern di test i_t in input, ed esaminiamo il risultato;

poiché le unità sono lineari, il valore di attivazione delle unità di output è:

$$O_{kt} = \sum_j W_{jk} \cdot i_{jt}$$

ovvero, in forma vettoriale:

$$\mathbf{o}_t = \mathbf{w} \cdot \mathbf{i}_t$$

e, sostituendo \mathbf{w} con la sua definizione si ha:

$$O_t = \varepsilon \cdot O_p \cdot \mathbf{i}_p^T \cdot \mathbf{i}_t$$

e, per la proprietà associativa del prodotto tra vettori:

$$\mathbf{O}_t = \varepsilon \cdot \mathbf{O}_p \cdot (\mathbf{i}_p^T \cdot \mathbf{i}_t)$$

quindi, l'output del test è proporzionale al prodotto dell'output dell'associazione appresa per il prodotto interno dell'input dell'associazione appresa e dell'input del test.

Sappiamo che il prodotto interno tra due vettori ($\mathbf{v}^T \mathbf{u}$) da una *misura della somiglianza* dei due vettori (proiezione);

in particolare se gli elementi dei vettori $\in \{-1, +1\}$, il prodotto interno normalizzato (diviso per il numero n di elementi del vettore: $(\mathbf{v}^T \cdot \mathbf{u}) / n$) è compreso fra -1 e +1 e da una misura della somiglianza dei due vettori (pattern):

- +1: vettori identici (pattern uguali)
- 1: vettori opposti (pattern complementari)
- 0: vettori ortogonali (pattern scorrelati)

possiamo quindi scrivere:

$$\mathbf{o}_t = \eta \cdot \varepsilon \cdot \mathbf{o}_p \cdot (\mathbf{i}_p^T \cdot \mathbf{i}_t)$$

quindi, l'output dei test è uguale all'output dell'associazione appresa moltiplicato per uno scalare dipendente dal grado di somiglianza dell'input di test con l'input dell'associazione appresa.

APPRENDIMENTO DI PIÙ ASSOCIAZIONI

dopo l'apprendimento di un insieme di associazioni
(i_p, o_p) i pesi sono:

$$W_{ji} = \varepsilon \sum_p o_{ip} \cdot i_{jp}$$

ovvero, in forma vettoriale:

$$W = \varepsilon \cdot \sum_p o_p \cdot i_p^T$$

quindi l'output prodotto da un pattern di test è:

$$o_t = n\varepsilon \cdot \sum_p o_p \cdot (i_p^T \cdot i_t)$$

Quindi, l'output della rete in risposta ad un pattern di test \mathbf{i}_t è la somma dei pattern di output delle associazioni memorizzate durante l'apprendimento (\mathbf{o}_p), ognuno pesato con la somiglianza ($\mathbf{i}_p^T \cdot \mathbf{i}_t$) del corrispondente pattern di input (\mathbf{i}_p) con il pattern di test (\mathbf{i}_t).

Ne segue che:

in generale, l'output di un test sarà una combinazione lineare degli output delle associazioni apprese, ognuno di essi pesato dal grado di somiglianza del rispettivo input con quello di test.

Osservazioni sul PA

- apprende col prodotto esterno;
- opera col prodotto interno;
- è in grado di memorizzare più associazioni in una stessa rete (matrice di pesi) se i pattern di input sono ortogonali; se non sono ortogonali si creano dei pattern ibridi e non è più possibile operare le associazioni originarie;
- quindi in casi reali il numero di associazioni memorizzabili è fortemente limitato;
- sono state proposte varie estensioni e miglioramenti (es. Hopfield Nets).