Domare gli attrattori: reti neurali

Reti neurali

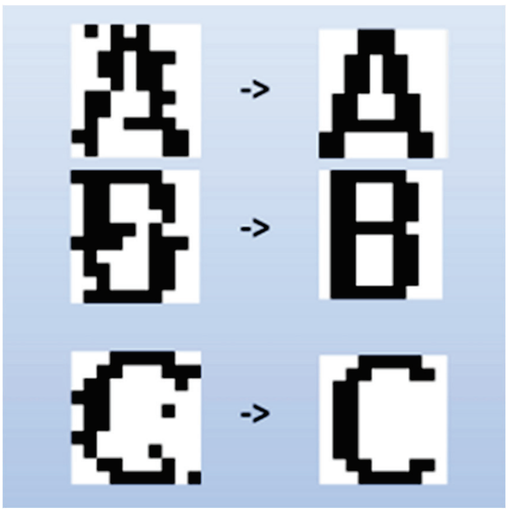
- **Corso di laurea in Informatica**
  (anno accademico 2024/2025)
  - Insegnamento: Apprendimento ed evoluzione in sistemi artificiali
  - Docente: Marco Villani

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA
Dipartimento di Scienze Fisiche, Informatiche e Matematiche

Visione dinamica dell'elaborazione

- Un programma per computer distingue di solito fra una parte di input e un output
- Anche un sistema dinamico elabora informazione
- Il caso più studiato è quello in cui l'input è costituito dalla condizione iniziale, o da una sua parte, e l'output dalla condizione finale (o da una sua parte)
- vedremo l'esempio delle reti neurali, ma lo schema è generale

Possiamo manipolare gli attrattori?



Ispirazione biologica

- La relazione fra macchine complesse e sistemi viventi
  - cibernetica, intelligenza artificiale, vita artificiale
- come vedremo, spesso l'imitazione (anche approssimativa) di sistemi biologici costituisce una buona euristica per costruire sistemi artificiali interessanti

Differenze cervello-computer

- Differenze cervello-computer
  - velocità di calcolo
  - gestione di grandi quantità di dati …
- ma anche
  - comprensione del linguaggio naturale
  - comprensione di immagini
  - gestione di informazioni imprecise …

Reti neurali

- Il computer incorpora un modello di cosa sia la computazione
- basato sulla manipolazione centralizzata di simboli fisici
  - questo modello è stato anche proposto come adatto a descrivere le attività mentali umane
- Cognitivismo: l'hardware è marginale, ciò che conta è la capacità di manipolare simboli
  - è possibile una intelligenza artificiale analoga a quella umana, basata su una architettura fisica molto diversa da quella del cervello

Reti neurali

- Ma il computer è una macchina capace di simulare i sistemi più diversi
- fiumi che scorrono, galassie che evolvono, stelle che esplodono…
- quindi possiamo cercare di simulare sul computer anche altre "architetture per l'intelligenza"
- ad esempio, quelle ispirate proprio dall'organizzazione fisica del cervello
- i primi modelli neurali

Intelligenza artificiale

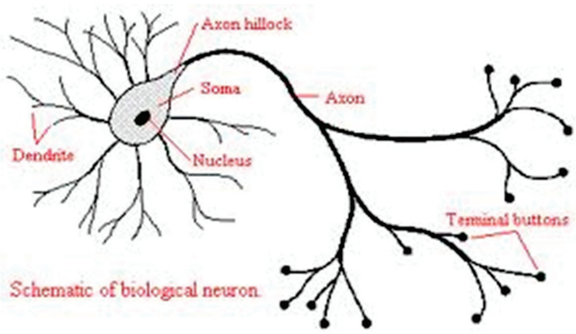
- Nasce nel secondo dopoguerra
  - Dartmouth 56
- l'intuizione dei fondatori
- l'approccio simbolico
  - sistemi "generalisti"
  - l'importanza della conoscenza specifica e la diffusione dei sistemi specializzati
- sistemi esperti
- la prevalenza dell'approccio simbolico

Reti neurali

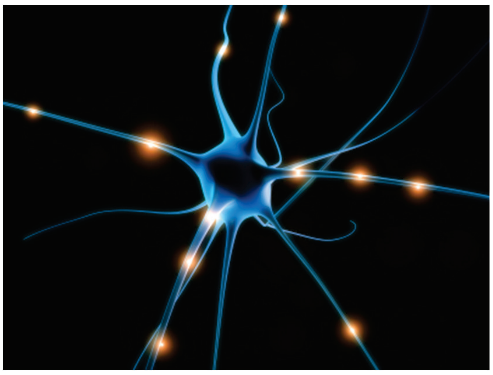
- I sistemi simbolici sono basati su qualche tipo di logica
  - regole del tipo SE… ALLORA…
- sono fragili rispetto alla presenza di contraddizioni
  - e quindi di conoscenze che cambiano nel tempo
- la realizzazione di sistemi esperti richiedeva sforzi notevoli
  - ingegneria della conoscenza
- lo stimolo verso sistemi robusti rispetto al rumore e alle contraddizioni, e capaci di apprendere da esempi
- la rinascita di interesse per le reti neurali negli anni '80

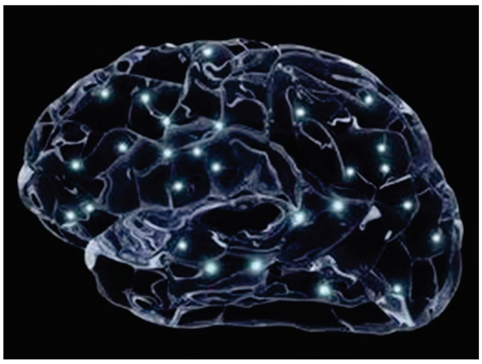
Reti neurali

- Sono ispirate dalla biologia, in particolare da studi sul funzionamento del cervello
- nel cervello l'attività di elaborazione delle informazioni viene svolta essenzialmente dai neuroni



Rappresentazione dell'informazione





grandmother neuron vs neural assemblies (Hebb)

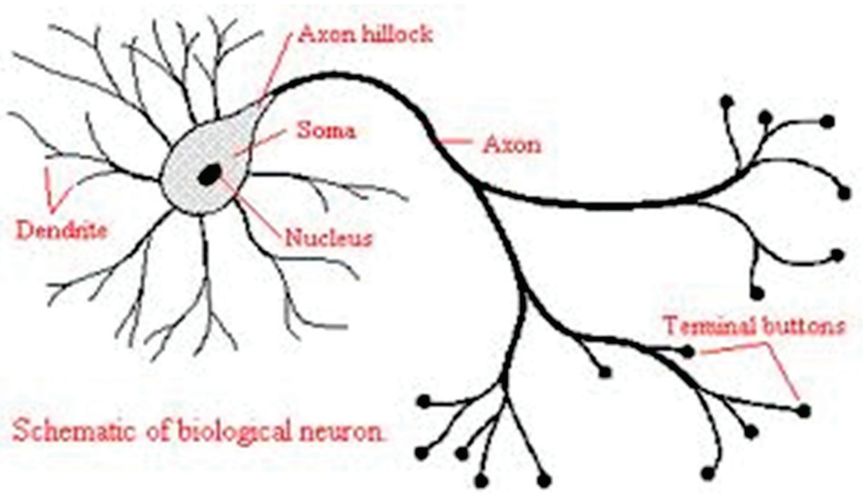
Rappresentazione dell'informazione

- Si comprese che il modello "un neurone, un concetto" era inadeguato
- enfasi sulle rappresentazioni distribuite
- a cui si devono anche le notevoli proprietà di robustezza del cervello
- oggi si è convinti che ci siano in effetti alcuni neuroni speciali, ma che la rappresentazione distribuita sia prevalente
  - solo un gruppo di neuroni è coinvolto nella rappresentazione di un concetto

Reti neurali

- Negli anni ‘40 si era capito che il trasferimento dell’impulso nervoso seguiva un meccanismo di tipo “tutto o niente”
- la relativa semplicità del singolo neurone è un altro indizio in favore delle rappresentazioni distribuite
  - i singoli neuroni possono morire, avere guasti etc.

Il neurone



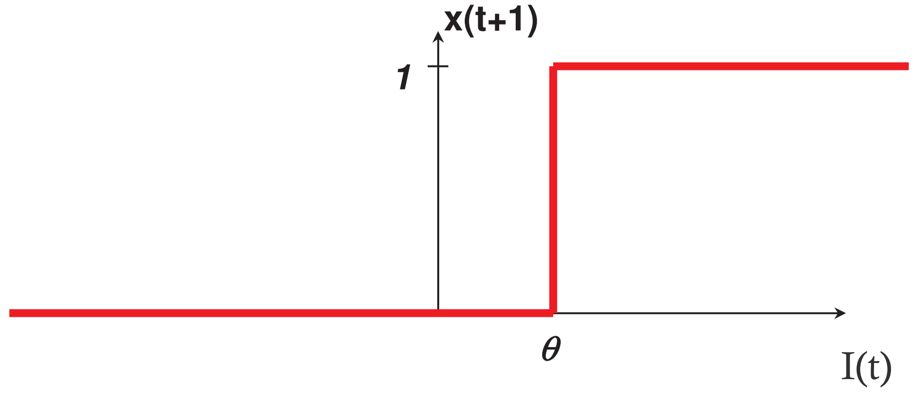
Il neurone

- Il neurone "spara" (trasmette un impulso lungo l'assone) se la somma dei suoi ingressi supera una certa soglia
- $I(t) = \Sigma\, s(t_i)\phi(t-t_i)$
- la funzione ϕ è una funzione decrescente al crescere dell'intervallo $t-t_i$. e misura la durata del ricordo di un impulso arrivato in precedenza
- se arrivano molti impulsi in un tempo sufficientemente breve, il neurone spara

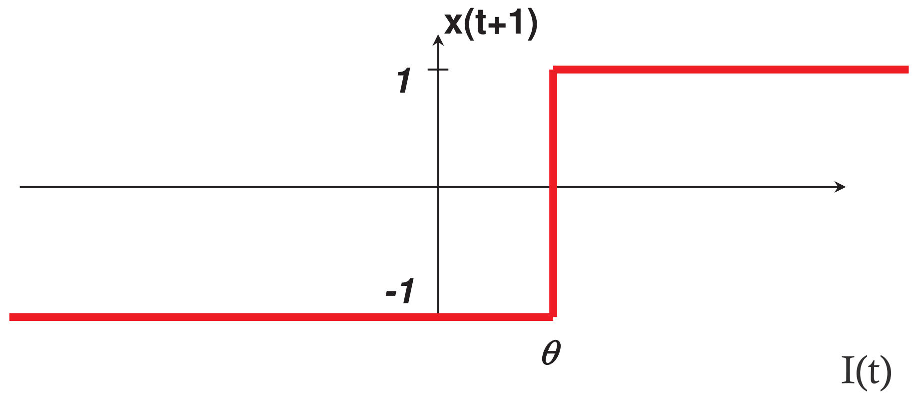
McCulloch e Pitts

- Proposero un modello semplificato, in cui ogni neurone viene aggiornato a tempi discreti
- non si considera l'affievolimento graduale del segnale: il neurone al tempo t+1 risente degli impulsi inviati al tempo t, e non di quelli precedenti
- il fatto che un neurone spari o non spari, a un certo istante, consente una modellistica booleana
  - lo stato è 1 (firing all'istante t) o 0
- Un neurone spara se la somma dei suoi ingressi supera il valore della soglia

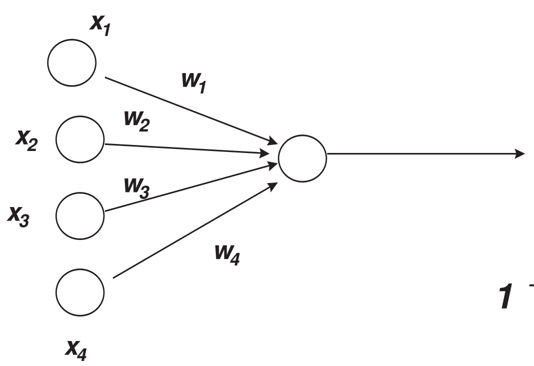
Modello booleano (McCulloch e Pitts)



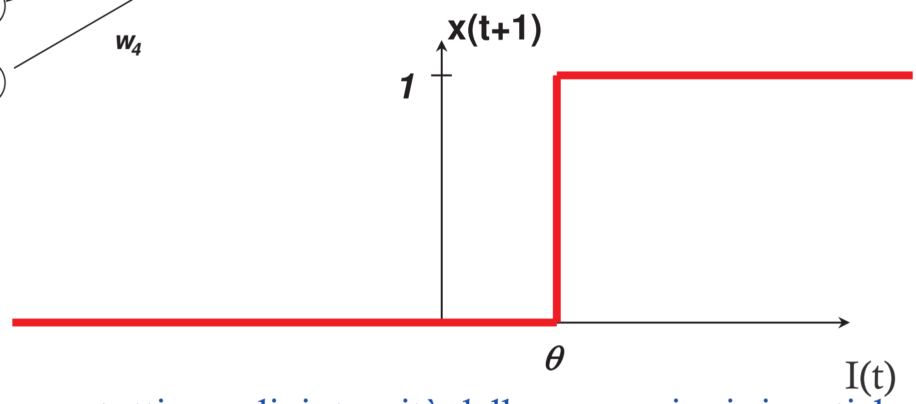
Modello booleano (McCulloch e Pitts)



Modello booleano a soglia



$$I = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4$$



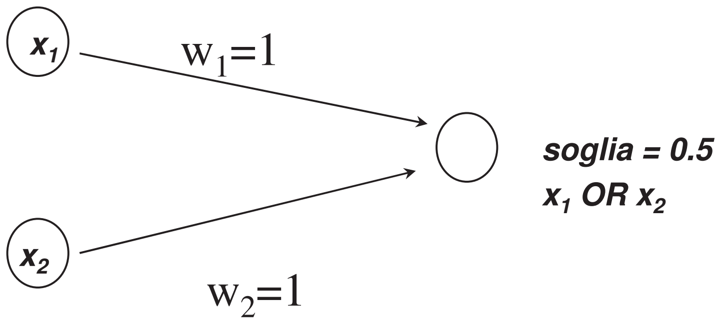
I neuroni non sono tutti uguali: intensità delle connessioni sinaptiche

Modello booleano a soglia

- McCulloch e Pitts dimostrarono che una rete di neuroni booleani a soglia poteva calcolare qualunque funzione logica (finita)
- esempio: AND, OR, (XOR)

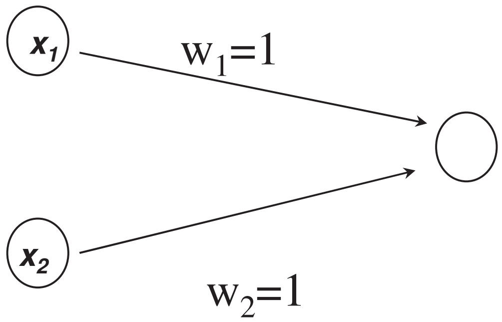
Modello booleano a soglia

- come realizzare la funzione OR



Modello booleano a soglia

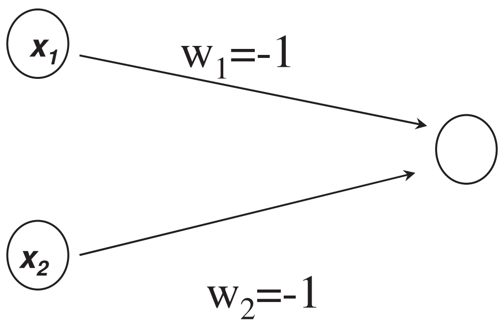
- come realizzare la funzione AND



soglia = 1.5
x_1 AND x_2

Modello booleano a soglia

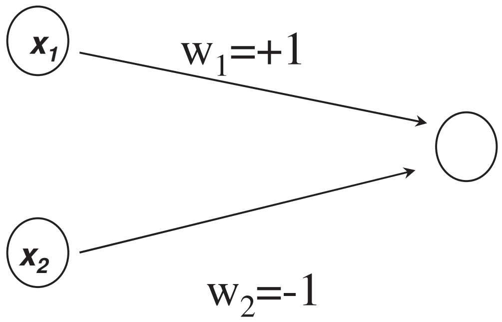
- come realizzare la funzione NAND



soglia = -1.5
NOT (x_1 AND x_2)

Modello booleano a soglia

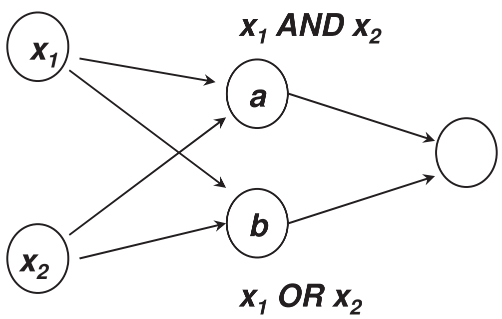
- come realizzare la funzione AND NOT



soglia = 0.5
x_1 AND NOT x_2

Modello booleano a soglia

- come realizzare la funzione XOR



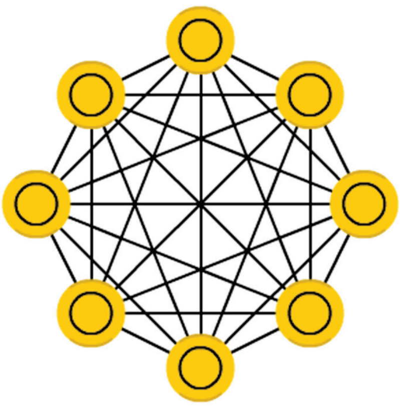
b AND NOT a

Modello booleano a soglia

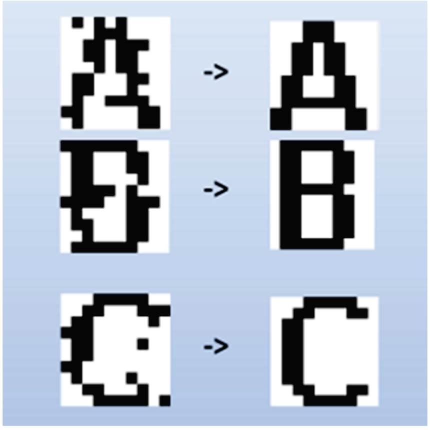
- Il risultato di McCulloch e Pitts dimostra che, con elementi "simili" ai neuroni, si può costruire una rete capace di calcolare ogni funzione logica
- Un risultato clamoroso
- ma lascia aperto il problema di come costruirla
- sarebbe molto interessante avere a disposizione un sistema capace di apprendere automaticamente quali sono i giusti valori di connessioni e di soglie
- perché una rete evolva in modo da realizzare un compito "utile"

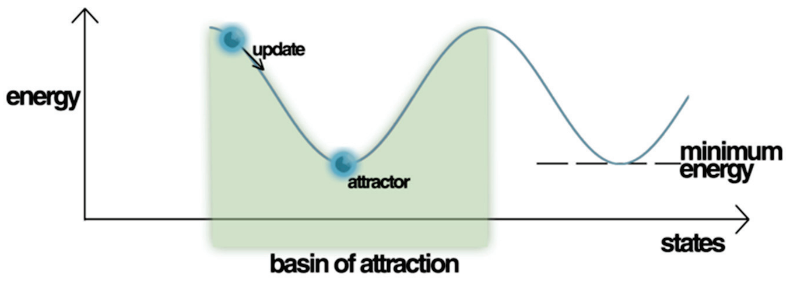
Il modello di Hopfield

- Esistono diverse varianti; discuteremo il modello booleano, con aggiornamento asincrono
- la rete è composta di N neuroni booleani a soglia
- i neuroni possono essere numerati 1, 2, …N
- x_i indica il valore della attivazione del neurone i-esimo
  - $x_i \in \{0,1\}$
- Il vettore X descrive lo stato della rete intera
  - $X = [x_1, \dots x_N] \in \{0,1\}^N$
  - p.es. N=8; X = [1,0,1,1,1,0,1,0]



Riconoscimento di pattern

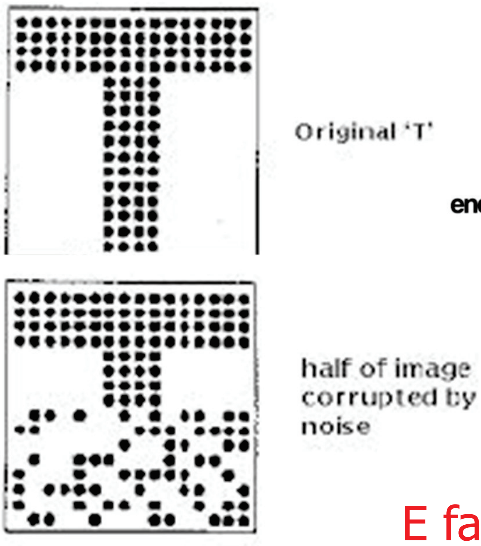


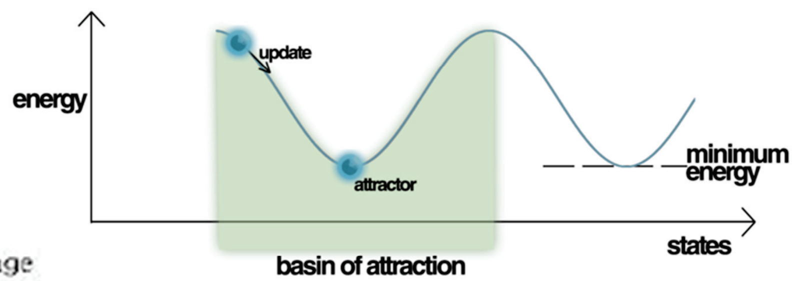


Come è possibile manipolare gli attrattori?

NB: i pattern non sono solo immagini!

Riconoscimento di pattern

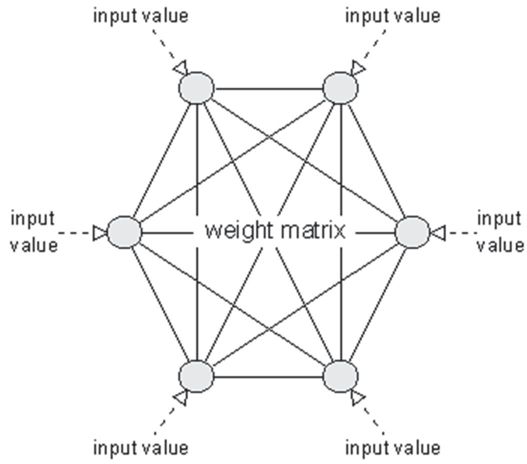




E far sì che riconosca un pattern anche in presenza di rumore

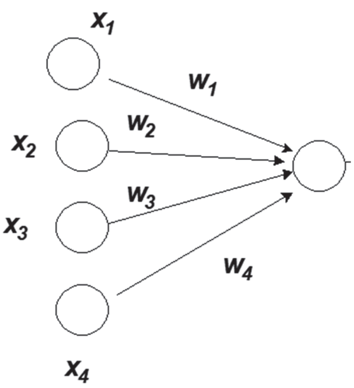
Il modello di Hopfield

- Il tempo evolve in passi discreti: t, t+1, t+2 …
  - il sistema passa quindi attraverso una successione di stati X(0), X(1) …
- per ogni coppia di neuroni è definita una intensità di connessione (peso sinaptico); w_{ik} misura l'intensità di connessione dal neurone *k* al neurone *i*
- $w_{ik} \in \mathbb{R}$; $w_{ii}=0$
  - si noti che in questo modello ogni neurone è connesso con ogni altro neurone
  - L'autoeccitazione e l'auto inibizione sono proibite



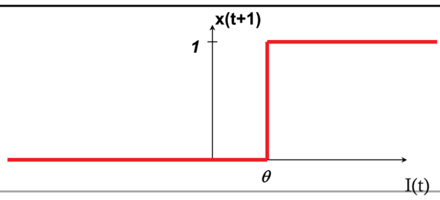
Il modello di Hopfield

- Ad ogni neurone è assegnata una soglia $\theta_i \in \mathbb{R}$
- definiamo l'input netto al neurone i-esimo, all'istante t, come segue:



$$I_i(X(t)) = \Sigma_k w_{ik} x_k(t)$$

$$\boxed{I_i = w_{i1}x_1 + w_{i2}x_2 + w_{i3}x_3 + w_{i4}x_4}$$



Il modello di Hopfield

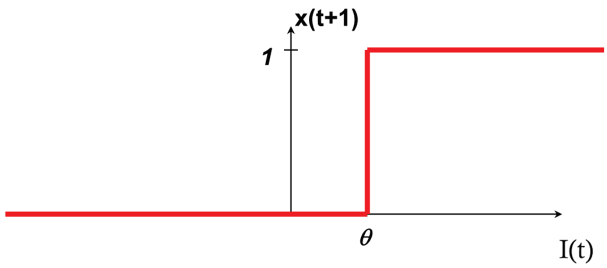
- La regola di evoluzione, che determina il nuovo stato X(t+1) in funzione del precedente, a valori costanti dei pesi sinaptici, è la seguente:
- scegli quale neurone aggiornare all'istante t+1
  - sia esso l'i-esimo

$x_i(t+1) = 1$ se $I_i(t) > \theta_i$

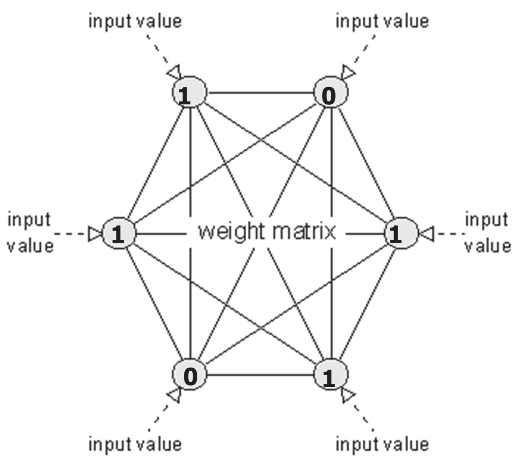
$x_i(t+1) = 0$ se $I_i(t) < \theta_i$

$x_i(t+1) = x_i(t)$ se $I_i(t) = \theta_i$

$x_k(t+1) = x_k(t)$ per ogni $k \neq i$

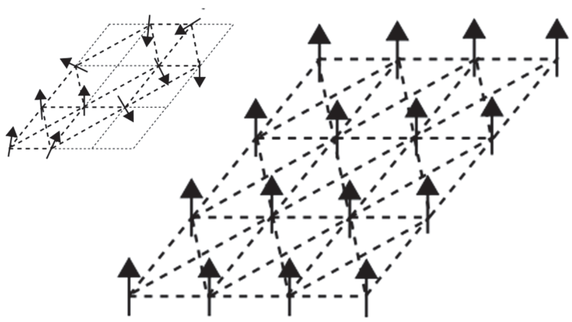


Relazione coi materiali magnetici disordinati



- Corrispondenza fra stati: neuroni booleani e spin booleani
  - acceso-spento, su-giù

- Momenti magnetici permanenti che, secondo le leggi della meccanica quantistica, assumono valori discreti
  - il caso più frequente è quello di due distinti valori



Relazione coi materiali magnetici disordinati

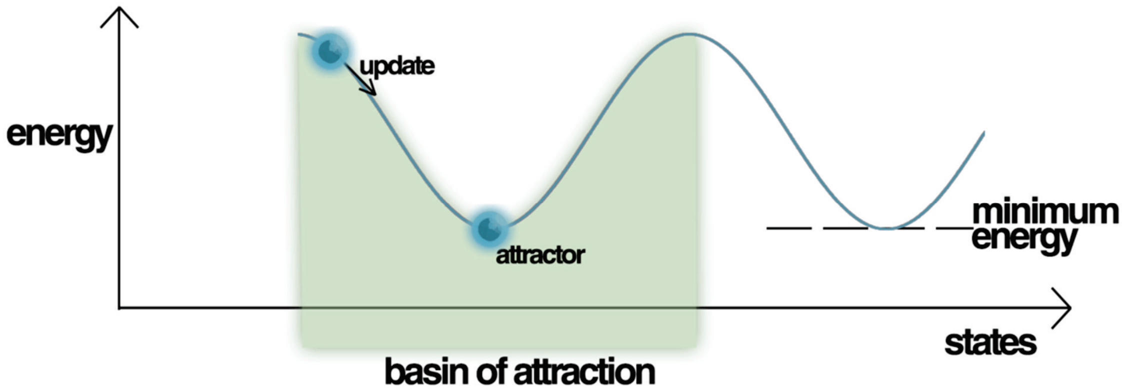
- Interazioni
  - nei materiali ferromagnetici le interazioni sono cooperative (tutti i "pesi" sono positivi), mentre in quelli antiferromagnetici sono anticooperative (tutti i pesi negativi)
  - nei cosiddetti vetri di spin sono presenti sia interazioni eccitatorie che inibitorie
- Corrispondenza fra pesi: reti neurali-vetri di spin
  - nei vetri di spin le interazioni sono locali, nelle reti neurali vi sono connessioni a distanza
- Grande importanza storica!

Dinamica della rete (con valori costanti dei pesi)

- Noi studieremo il caso di aggiornamento asincrono: ad ogni passo viene aggiornato un solo neurone, scelto a caso
- Altre alternative sono l'aggiornamento asincrono secondo una sequenza fissa, oppure secondo una sequenza modificata dopo avere aggiornato tutti gli N neuroni
- Si può dimostrare che la rete, con aggiornamento asincrono, tende sempre a uno stato costante (punto fisso) nel caso di connessioni simmetriche $w_{ik}=w_{ki}$
  - non ci sono cicli

Funzione energia

$$E(X) = -\frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{N} w_{ij}x_i x_j + \sum_{i=1}^{N} \vartheta_i x_i$$



Funzione energia

$$E(X) = -\frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{N} w_{ij}x_i x_j + \sum_{i=1}^{N} \vartheta_i x_i$$

- Si può dimostrare che, se W è una matrice simmetrica, l'energia è una funzione non crescente nella dinamica asincrona
  - $E(X(t+1)) \leq E(X(t))$
  - e che $E(X(t+1)) = E(X(t)) \Rightarrow X(t+1)=X(t)$
  - L'energia è quindi una funzione di Lyapunov per il modello di Hopfield simmetrico, che tende a un punto fisso

Evoluzione verso un punto fisso

- Il sistema parte in un certo stato iniziale X(0), con energia E(0)
- evolve in maniera asincrona, aggiornando un neurone alla volta; se all'istante t lo stato del neurone prescelto cambia, allora E(t+1)<E(t), altrimenti E(t+1)=E(t)
- Il sistema non può quindi trovarsi, in due momenti successivi, in due stati diversi con la stessa energia
- Il numero di stati possibili è finito (2^N)
- pertanto il sistema evolve fino a raggiungere uno stato costante (attrattore) corrispondente a un minimo locale dell'energia, tale cioè per cui tutti gli stati che differiscono da esso per un solo neurone hanno energia maggiore

Fac: Dimostrazione del fatto che l'energia è una funzione non crescente

- Supponiamo che all'istante t venga aggiornato il neurone k-esimo, e sia $\delta x_k(t)$ la sua variazione; la variazione di E è

$$\delta E(t) = -\frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{N} w_{ij}\big((x_i(t+1)x_j(t+1) - x_i(t)x_j(t)\big)(\delta_{ik} + \delta_{jk}) +$$

$$+\sum_{i=1}^{N} \vartheta_i (x_i(t+1) - x_i(t))\delta_{ik} =$$

- δ_{ij}: simbolo di Kroneker

$$= -\frac{1}{2}\sum_{\substack{j=1\\ j\neq k}}^{N} w_{kj}(x_k(t+1)x_j(t+1) - x_k(t)x_j(t)) -$$

$$-\frac{1}{2}\sum_{\substack{i=1\\ i\neq k}}^{N} w_{ik}(x_i(t+1)x_k(t+1) - x_i(t)x_k(t)) + \vartheta_k(x_k(t+1) - x_k(t))$$

Fac: Dimostrazione del fatto che l'energia è una funzione non crescente – 2

$$\delta E(t) = -\frac{1}{2}\sum_{\substack{j=1\\ j\neq k}}^{N} w_{kj}(x_k(t+1)x_j(t+1) - x_k(t)x_j(t)) -$$

$$-\frac{1}{2}\sum_{\substack{i=1\\ i\neq k}}^{N} w_{ik}(x_i(t+1)x_k(t+1) - x_i(t)x_k(t)) + \vartheta_k(x_k(t+1) - x_k(t)) =$$

$$= -\frac{1}{2}\sum_{\substack{j=1\\ j\neq k}}^{N}(w_{kj} + w_{jk})\left[x_j(t+1)x_k(t+1) - x_j(t)x_k(t)\right] + \vartheta_k(x_k(t+1) - x_k(t)) =$$

- $j\neq k$, x_j non cambia, quindi

$$\left[x_j(t+1)x_k(t+1) - x_j(t)x_k(t)\right] = x_j(t)\left[x_k(t+1) - x_k(t)\right]$$

Fac: Dimostrazione del fatto che l'energia è una funzione non crescente – 3

- poiché solo il k-esimo neurone è cambiato, otteniamo

$$\delta E(t) = -\frac{1}{2}\sum_{\substack{j=1\\ j\neq k}}^{N}(w_{kj} + w_{jk})\left[x_k(t+1) - x_k(t)\right]x_j(t) + \vartheta_k(x_k(t+1) - x_k(t)) =$$

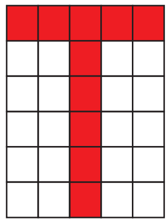
$$= -\frac{1}{2}\delta x_k(t)\sum_{\substack{j=1\\ j\neq k}}^{N}(w_{kj} + w_{jk})x_j(t) + \vartheta_k\,\delta x_k(t)$$

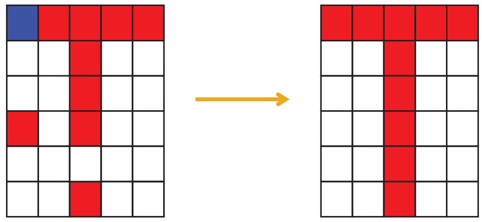
se $w_{kj}=w_{jk}$ si ha infine

$$\delta E(t) = -\delta x_k(t)\sum_{\substack{j=1\\ j\neq k}}^{N}(w_{kj}x_j(t) - \vartheta_k) = -\delta x_k(t)\left[I_k(X(t)) - \vartheta_k\right] \leq 0 \quad \text{QED}$$

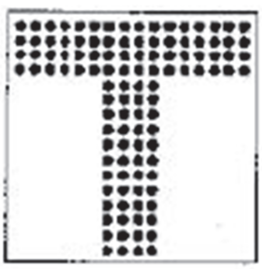
Evoluzione della rete di Hopfield

- Per poter utilizzare la rete per svolgere compiti interessanti, bisogna far sì che la sua evoluzione dinamica corrisponda a operazioni significative
- la rete può quindi essere utilizzata per riconoscere o classificare
- per fare questo è necessario far sì che gli attrattori coincidano con schemi dotati di significato





Riconoscere anche nel rumore



Original 'T'

half of image corrupted by noise

Apprendimento

- Si separa una fase di apprendimento, in cui si modificano i pesi, da una fase di riconoscimento, in cui la rete evolve mantenendo costanti le connessioni sinaptiche
  - L'apprendimento può essere raccontato come "presentazione" di una serie di pattern alla rete, in cui ad ogni presentazione di un pattern segue una modifica delle connessioni
  - In seguito, la rete funzionerà come sistema dinamico a connessioni fisse, associando ogni condizione iniziale a uno stato finale

Un principio di auto-organizzazione

- L'apprendimento deve far in modo che l'evoluzione della rete vada in un senso "utile"
- ovvero deve far sì che gli attrattori corrispondano a pattern dotati di significato
- e che i loro bacini di attrazione siano adeguati

Apprendimento hebbiano

- Donald Hebb (1949)
  - quando l'assone della cellula A è sufficientemente vicino da eccitare una cellula B e ripetutamente e in modo persistente partecipa alla sua attivazione, allora si verifica qualche processo di crescita o cambiamento metabolico in una o entrambe le cellule, in modo da aumentare l'efficienza di A nell'eccitare B

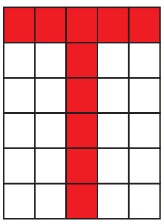
- Per realizzare l'idea di Hebb nel modello si rafforzano le connessioni fra i neuroni che compaiono nello stesso stato, e si indeboliscono quelle fra neuroni che si trovano in stati diversi

Apprendimento hebbiano

- Per tradurre l'ipotesi nel modello di Hopfield, dobbiamo fare in modo che, insegnando un pattern $A = (a_1 \ldots a_N)$, le sinapsi dei neuroni che sono nello stesso stato aumentino di intensità

- una possibilità è $\Delta w_{ik} = (2A_i\text{-}1)(2A_k\text{-}1)$

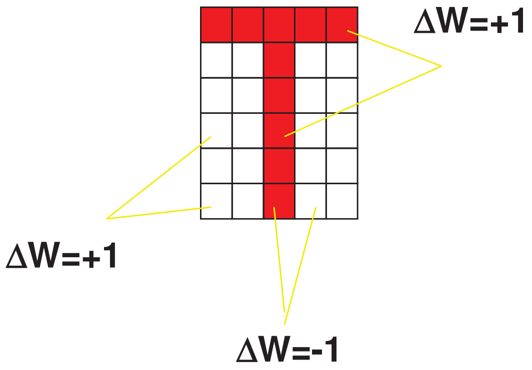
Caso booleano con 0 ed 1



ΔWij	Ai=0	Ai=1
Ak=0	1	-1
Ak=1	-1	1

Regola di apprendimento

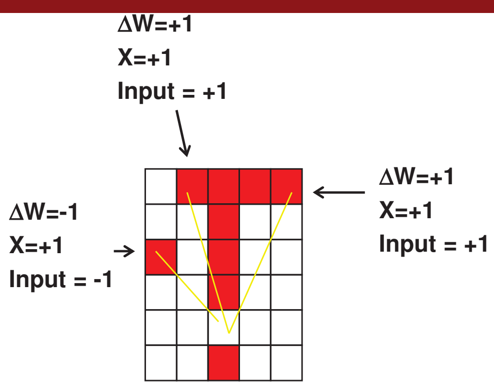
- Per “apprendere” uno schema
- se due neuroni sono entrambi accesi, $\Delta W_{ij} = +1$
- se sono entrambi spenti, allora $\Delta W_{ij} = +1$
- se sono uno spento e uno acceso, $\Delta W_{ij} = -1$



Caso booleano con 0 ed 1

Ricostruzione di uno schema

- I neuroni “accesi nello schema” forniscono un contributo che tende ad accendere quello erroneamente spento

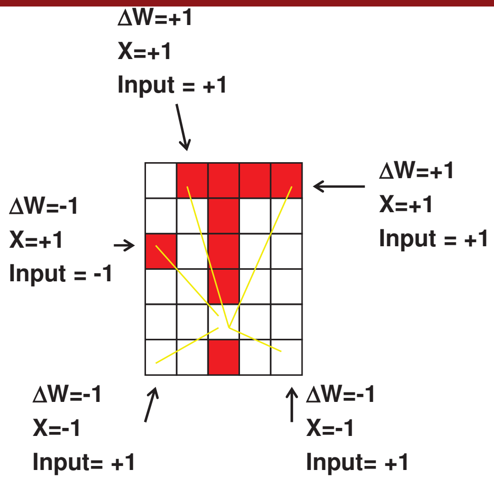


- eventuale rumore aggiuntivo può creare disturbo, ma è tollerabile (finché non supera un certo livello)

Caso booleano con 0 ed 1

Ricostruzione di uno schema

- I neuroni "accesi nello schema" forniscono un contributo che tende ad accendere quello erroneamente spento
- anche i neuroni "spenti nello schema" possono dare un contributo analogo
- eventuale rumore aggiuntivo può creare disturbo, ma è tollerabile (finché non supera un certo livello)



Caso booleano con -1 ed 1

Evoluzione e computazione

- Per insegnare un pattern A = (a_1 … a_N), le sinapsi dei neuroni che sono nello stesso stato aumentano di intensità
- $\Delta w_{ik} = (2A_i-1)(2A_k-1)$
- Regola per l'apprendimento di M pattern $A^1, \ldots A^M$
- $\Delta w_{ik} = \Sigma_m(2A_i^m-1)(2A_k^m-1)$

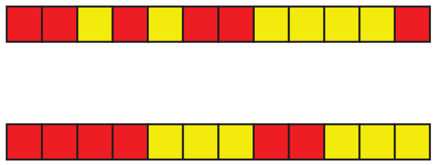
Comportamento del modello di Hopfield

Comportamento del modello di Hopfield

- Il sistema di Hopfield è in grado di memorizzare diversi pattern e quindi di riconoscere diverse condizioni iniziali come "versioni perturbate" di uno schema di fondo
- in questo modo si realizza un apprendimento basato sulle proprietà di autoorganizzazione di un sistema dinamico!
- A volte il riconoscimento non è corretto, e a volte si creano attrattori spuri, diversi da tutti quelli appresi
- Per capire meglio come funziona analizziamo il comportamento del modello nel caso in cui siano stati appresi pochi pattern

Comportamento del modello di Hopfield

- Definizioni:



X

Y

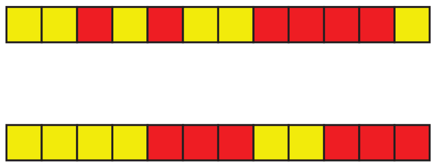
sovrapposizione di due pattern:

$Q(X,Y) = \Sigma X_k Y_k$

coincide col numero di posizioni in cui X e Y presentano il valore "1"

Q(X,Y) = 3

- complemento di un pattern X:
  $X' = \{x'_i \mid x'_i = 1 - x_i,\ i=1 \ldots N\}$
- $Q(X,Y)+Q(X',Y)=N_Y$



X'

Y'

Relazione fra input netto e overlap

$$I_i(X) = \sum_{j\neq i}\sum_m (2A_i^m - 1)(2A_j^m - 1)X_j$$

ma

$$2A_i^m - 1 = A_i^m + (1 - A_i'^m) - 1 = A_i^m - A_i'^m$$

quindi

$$I_i(X) = \sum_m (2A_i^m - 1)\sum_{j\neq i}(A_j^m - A_j'^m)X_j =$$

$$\sum_m (2A_i^m - 1)(Q(A^m,X) - Q(A'^m,X)) - X_i\sum_m (A_i^m - A_i'^m)^2 =$$

$$\sum_m (2A_i^m - 1)(Q(A^m,X) - Q(A'^m,X)) - MX_i$$

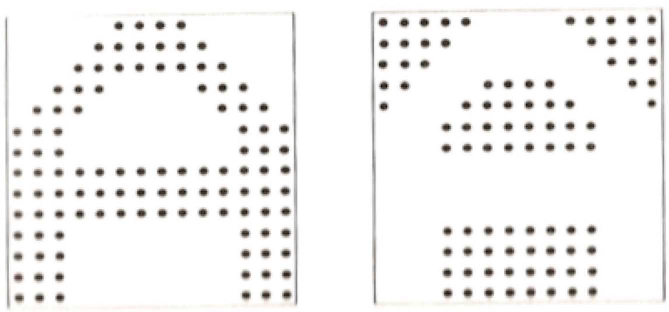
- $I_i(X)=(2A_i-1)[Q(A,X)-Q(A',X)]-X_i$

Un solo pattern A, soglia nulla

- $I_i(X)=(2A_i-1)[Q(A,X)-Q(A',X)]-X_i$
- una proprietà da ricordare
- un termine del tipo $(2A_i-1)$ moltiplicato per un numero positivo "abbastanza grande" tende a far sì che $X_i \rightarrow A_i$
- infatti se $A_i=1$ l'input netto è positivo (sopra soglia) quindi, quando si aggiorna il neurone i-esimo, esso si porta nello stato 1, indipendentemente da quale fosse il suo stato precedente
- mentre se $A_i=0$ l'input netto è negativo (sotto soglia) quindi, quando si aggiorna il neurone i-esimo, esso si porta nello stato 0, indipendentemente da quale fosse il suo stato precedente

Un solo pattern A, soglia nulla

- $w_{ik} = (2A_i-1)(2A_k-1)$
- si noti che
- $w_{ik} = (2A_i-1)(2A_k-1) = (2(1-A'_i)-1)(2(1-A'_k)-1) = (2A'_i-1)(2A'_k-1)$
- insegnare un pattern o il suo complementare è la stessa cosa!



Un solo pattern A, soglia nulla

- $I_i(X)=(2A_i-1)[Q(A,X)-Q(A',X)]-X_i$
- si verifica facilmente che A e A' sono punti fissi
  - se $Q(A,A)-Q(A',A)>1$ $I_i(A)$ ha lo stesso segno di A_i, che resta quindi inalterato
  - se $Q(A,A)-Q(A',A)=1$ $I_i(A)=A_i-1$ e A_i resta inalterato
  - se $Q(A,A)-Q(A',A)=0$ allora $A=(0\ldots0)$, $I_i(A)=0$ e lo stato resta inalterato
- se $A,A' \neq 0$ lo stato $(0\ldots0)$ è comunque un punto fisso
  - perché l'input netto è sempre 0 e lascia tutto inalterato

Uno stato tende all'attrattore cui assomiglia

- consideriamo il caso $X(0)\neq A, A', 0$
- $X(0) \rightarrow A$ se $Q(A,X(0)) > Q(A', X(0))$
- $X(0) \rightarrow A'$ se $Q(A,X(0)) < Q(A', X(0))$

Dimostrazione

- $I_i(X)=(2A_i-1)[Q(A,X)-Q(A',X)]-X_i$
- se $Q(A,X)>Q(A',X)+1$, allora $I_i(X)$ ha lo stesso segno di $2A_i-1$, quindi ogni neurone tende al valore corrispondente in A_i
- se $Q(A,X)=Q(A',X)+1$, allora
  - se $X_i=0$ $I_i(X)$ ha lo stesso segno di $2A_i-1$, e il neurone tende al valore che ha in A;
  - se $X_i=1$ e $A_i=1$, $I_i(X)$ vale 0 e il neurone resta=1, come in A
  - se $X_i=1$ e $A_i=0$ $I_i(X)$ ha valore negativo e il neurone tende a 0 come in A

Altre proprietà

- La convergenza verso lo stato cui X assomiglia maggiormente (nel senso specificato dalla misura di overlap Q) è veloce
- si realizza in un numero di iterazioni non superiore al numero R di passi necessari per aggiornare almeno una volta ogni elemento che non si trova inizialmente nello stesso stato del suo attrattore

Altre proprietà

- Lo stato X=(0,0 … 0) è un punto fisso; tuttavia esso non può essere raggiunto a partire da nessuno stato iniziale (a meno che A o A' non coincidano con esso)
  - lo stato (0…0) acquista un bacino di attrazione se la soglia è diversa da zero
- se Q(A,X)=Q(A' ,X) la rete può evolvere verso uno o l'altro dei due attrattori in funzione della sequenza di aggiornamento
  - precisamente verso quello che ha uno "0" in corrispondenza della prima posizione che viene aggiornata in cui X(0) abbia valore "1"
- Sistema non deterministico

Il risultato dipende dalla sequenza?

- se Q(A,X)=Q(A' ,X), allora
- $I_i(X)=(2A_i-1)[Q(A,X)-Q(A',X)] -X_i = -X_i$
- se viene aggiornato un neurone che vale "0", esso rimane invariato e rimane Q(A,X)=Q(A' ,X)
- la prima volta che viene aggiornato un neurone che vale "1", esso diventa 0, e a quel punto i due overlap non sono più uguali
- è maggiore l'overlap col pattern (A o A') che ha il valore "0" in quella posizione, e da quel momento in poi il sistema tende verso quel pattern
  - si noti che i bacini di attrazione non sono disgiunti, nel caso di aggiornamento asincrono

Due pattern A e B, soglia nulla

- $I_i(X) = (2A_i-1)[Q(A,X)-Q(A',X)] + (2B_i-1)[Q(B,X)-Q(B',X)] -2X_i$
- def: pattern di base $P \subset \{A, B, A', B'\}$

A e B sono punti fissi

- dimostriamo dapprima che i pattern base sono punti fissi *
- $I_i(A)= (2A_i-1)Q(A,A) + (2B_i-1)[Q(B,A)-Q(B',A)] -2A_i$
  - ma $Q(B',A)=Q(A,A)-Q(B,A)$ quindi
- $I_i(A)= (2A_i-1)Q(A,A) + (2B_i-1)[2Q(B,A)-Q(A,A)] -2A_i =$
- $= 2 (A_i-B_i)Q(A,A) + 2(2B_i-1)Q(B,A)-2A_i$

A e B sono punti fissi

- $I_i(A) = 2\,(A_i - B_i)Q(A,A) + 2(2B_i - 1)Q(B,A) - 2A_i$
- casi possibili (si noti che $Q(B,A) \geq 1$, altrimenti B coinciderebbe con A')
- $A_i = B_i = 1$, $I_i = 2Q(B,A) - 2 \geq 0$, il nuovo valore coincide con A_i
- $A_i = 1$, $B_i = 0$, $I_i = 2Q(A,A) - 2Q(B,A) - 2 \geq 0$, il nuovo valore $= A_i$
- $A_i = 0$, $B_i = 1$, $I_i = -2Q(A,A) + 2Q(B,A) \leq 0$, il nuovo valore $= A_i$
- $A_i = B_i = 0$, $I_i = -2Q(B,A) - 2 \leq 0$, il nuovo valore $= A_i$

Due pattern A e B, soglia nulla

- $I_i(X) = (2A_i - 1)[Q(A,X) - Q(A',X)] + (2B_i - 1)[Q(B,X) - Q(B',X)] - 2X_i$
- ricorda: pattern di base $P \subset \{A, B, A', B'\}$
- Se $Q(X(0),P)$ è maggiore dell'overlap con gli altri 3 pattern di base, allora X -> P

Evoluzione da uno stato arbitrario: formule utili *

- sia Q(A,X) maggiore dell'overlap con gli altri pattern base
- e sia Q(B,X) ≥Q(B',X)
  - senza perdita di generalità: è sufficiente nominare opportunamente "B" e "B'"
  - per def.:
  - Q(A',X) = Q(X,X) - Q(A,X)
  - Q(B',X) = Q(X,X) - Q(B,X)
  - pertanto Q(B',X) = Q(X,X) - Q(B,X) > Q(X,X) – Q(A,X) = Q(A',X)
- quindi Q(A,X)-Q(A',X) > Q(B,X)-Q(B',X)

Evoluzione da uno stato arbitrario: formule utili *

- poiché si ha a che fare con valori interi di Q
- Q(A,X) ≥ Q(B,X)+1 ; Q(A',X) ≤ Q(B',X)+1
- quindi Q(A,X)-Q(A',X) ≥ Q(B,X)-Q(B'X) + 2

X tende al pattern base con cui ha overlap maggiore

$$I_i(X) = (2A_i - 1)(Q_{AX} - Q_{A'X}) + (2B_i - 1)(Q_{BX} - Q_{B'X}) - 2X_i$$

- Supponiamo che X abbia un overlap con A maggiore di quello con gli altri pattern base
- I_i è la somma di
  - un termine pari a $(2A_i-1)$ moltiplicato per un numero positivo
    - questo termine tende a fare assumere al neurone i il valore A_i
  - la somma di altri due termini; nel "caso peggiore" hanno entrambi segno opposto ad A_i, e valgono $Q_{BX}-Q_{B'X}-2$, che non può mai superare $Q_{AX}-Q_{A'X}$; quindi il nuovo valore è A_i, l'overlap con A resta lo stesso o aumenta: X tende ad A

Equisovrapposizione di due pattern *

- sia Q(A,X)=Q(B,X) > Q(A' X), Q(B',X)
- $I_i(X) = (2A_i-1)[Q(A,X)-Q(A',X)] + (2B_i-1)[Q(B,X)-Q(B',X)] -2X_i$
- ma Q(A',X)=Q(X,X)-Q(A,X) quindi anche Q(A',X)=Q(B',X) per cui
- $I_i(X) = [(2A_i-1)+ (2B_i-1)] [Q(A,X) - Q(A',X)] -2X_i$
- $I_i(X) =2 \{(A_i+B_i-1)[(Q(A,X) - Q(A',X)] -X_i\}$

- 8 casi possibili per Ai, Bi, Xi

Ai	Bi	Xi	nuovo Xi
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- si noti che il nuovo valore di X_i è sempre pari al prodotto A_iB_i

- in due casi X_i cambia valore senza modificare l'uguaglianza degli overlap
- in due casi X_i cambia valore cambiando anche l'uguaglianza degli overlap: 0 1 1 e 1 0 1
- sono i casi in cui X_i=1 e A_i e B_i hanno valori diversi: in questi casi cambia l'overlap con uno dei due pattern A o B, e X tende poi ad uno di essi
- se non esiste nessuna cella in cui X_i=1 e $A_i \neq B_i$, allora, poiché il nuovo valore di X_i è sempre pari al prodotto A_iB_i, il pattern X tende all'AND dei pattern A e B

Commento

- A volte quindi il pattern X tende all'AND dei pattern A e B
  - è un comportamento interessante
- la rete non ha elementi per decidere
  - E genera un nuovo attrattore (diverso da quelli appresi) che in questo caso ha un significato leggibile

- Nota: se la sovrapposizione è identica con tutti e 4 i pattern di base, l'attrattore finale dipende dall'aggiornamento

Tre pattern

- Se i pattern sono più di due, non è più possibile garantire che essi siano punti fissi

$$I_i(X) = (2A_i - 1)(Q_{AX} - Q_{A'X}) + (2B_i - 1)(Q_{BX} - Q_{B'X}) + (2C_i - 1)(Q_{CX} - Q_{C'X}) - 3X_i$$

- il primo termine stabilizza A
- ma il contributo degli ultimi 3 termini può superare quello di Q(A,A) e far sì che A non sia più punto fisso
- si tratta però di un caso abbastanza estremo, richiede che l'overlap di ognuno dei pattern base (o del suo complementare) con A sia confrontabile con Q(A,A)
- quindi che i pattern base siano molto simili

Più pattern

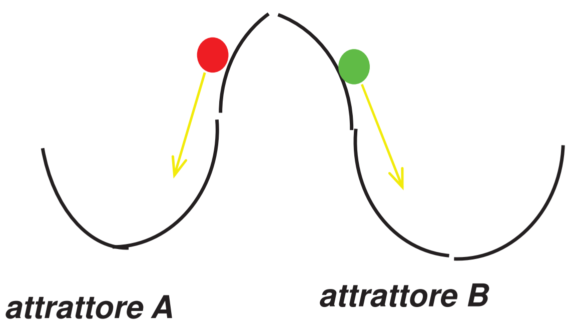
- consideriamo il caso in cui lo stato iniziale è proprio A; essa diventa $(2A_i-1)$ Q(A,A)
- questo termine dà un contributo positivo a tutti i neuroni che hanno valore 1 in A, e negativo a quelli che hanno valore 0: tende quindi a riprodurre A
- se i pattern memorizzati sono tali per cui A ha approssimativamente la stessa sovrapposizione sia con gli altri pattern B che coi loro complementari B', allora [Q(A,B)-Q(A,B')] << Q(A,A)
- il termine Q(A,A) domina la somma, tende a stabilizzare A; è quindi possibile memorizzare diversi stati, se i pattern sono scorrelati

Più pattern, evoluzione da uno stato arbitrario

- La funzione di input è infatti una somma, su tutti i pattern del training set, di termini del tipo $(2A_i-1)[Q(A,X)-Q(A',X)]$
  - trascuriamo il contributo -MXi supponendo che sia molto più piccolo di Q(A,A)
  - assunto giustificato se il numero di pattern è molto inferiore a quello dei neuroni, M<<N)
- se lo stato iniziale X ha una forte sovrapposizione con A e una modesta sovrapposizione con gli altri pattern, allora
- $Q(A,X) \approx Q(A,A)$, $Q(B,X)-Q(B',X) \approx 0$
- per cui il termine $(2A_i-1)Q(A,X)$ domina la somma, e X →A, ovvero al pattern cui somiglia maggiormente

Evoluzione e computazione

- Se l'input netto è dominato dal termine $(2A_i-1)Z$, con Z>0, il sistema tenderà verso A_i
  - Quando viene aggiornato il neurone i-esimo X_i diventerà identico ad A_i
- la regola di Hebb "scava buche" nel paesaggio dell'energia in corrispondenza dei pattern da apprendere
  - "computazione con attrattori dinamici"



Interferenza

- Pattern simili tendono invece a dare origine a fenomeni di interferenza
- L'entità del fenomeno dipende dai pattern
- Se tutti i pattern appresi sono casuali e scorrelati, con $pr\{A_i^m=1\}=pr\{A_i^m=0\}=1/2$, allora si può apprendere un numero di pattern M_{max} che scala approssimativamente con
- $M_{max} \approx \alpha N$ ($\alpha \approx 0.14$)

Proprietà del modello

- Riconoscimenti robusti anche in presenza di rumore nell'input
- o di malfunzionamento di qualche neurone
  - Rappresentazione distribuita dei concetti!
- Dal punto di vista del controllo, gli stati spuri sono perturbazioni
- Dal punto di vista dell'emergenza, gli stati spuri rappresentano elaborazioni che il sistema astrae dalla base di esempi

Archetipi

- Insegnamento di k versioni distorte del pattern U
  - (u^1 ...u^k): ognuna è ottenuta a partire da U, modificando alcuni elementi scelti a caso (ogni bit ha la stessa probabilità di essere modificato)
  - si trascurano le interferenze da altri pattern; supponiamo che lo stato iniziale abbia una elevata somiglianza con U, così da trascurare le sovrapposizioni coi complementari: $Q(u^p,X) \approx Q(U,X)$
- la funzione di input è dominata da $Q(U,X)\Sigma_m(2u_i^m-1)$, il sistema tende a U: il modello è capace di ricostruire l'archetipo partendo dalla presentazione di sue versioni distorte

L'importanza del modello

- Importanza storica
  - la rinascita di interesse per le reti neurali
- Evidenzia proprietà collettive della rete
  - rappresentazioni distribuite, robustezza
- Evidenzia in maniera chiara la relazione fra proprietà dinamiche e prestazioni della rete
  - elaborazione basata su attrattori
- Consente una analisi approfondita della dinamica
- Mostra la formazione spontanea di "concetti" non compresi fra quelli utilizzati per l'addestramento

L’importanza del modello

- Mostra come sia possibile creare riconoscitori robusti
  - che non sono inficiati dalla presenza di un certo livello di rumore della rete
- Capacità di generalizzazione
- Applicabile non solo a immagini ma anche a elaborazioni simboliche
  - frasi ambigue
  - descrizioni
  - …