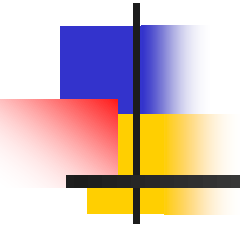


# Boltzmann Machine e Deep NN

---



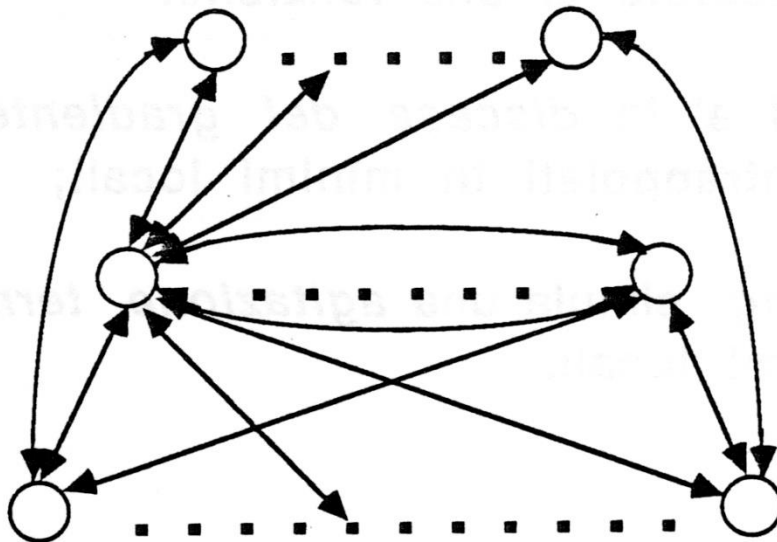


# Boltzmann Machine (Modelli Termodinamici)

---

- la Boltzmann Machine (BM) è una Rete Neurale stocastica, completamente connessa, con due tipi di unità: visibili (input, output) e invisibili (o interne);
- input e output binario  $\{0, 1\}$ ; pesi simmetrici; funzione di attivazione *stocastica* dipendente dalla *temperatura* della rete;

# Boltzmann Machine (2)





# Boltzmann Machine (3)

---

- è stato proposto un algoritmo di apprendimento che permette di apprendere i pesi delle connessioni ottimizzando una misura delle prestazioni della rete;
- per produrre un output la BM usa un metodo (Simulated Annealing, 1983) che tende a soddisfare contemporaneamente molti vincoli (input + pesi);



# Boltzmann Machine (4)

---

- la BM è in grado di completare dei pattern parziali dati sulle unità visibili. Se dividiamo le unità visibili in unità di input e di output, dato il pattern degli input la BM lo completa producendo gli output (CLASSIFICAZIONE); altrimenti opera come MEMORIA ASSOCIATIVA, restituendo il pattern più simile (fra quelli appresi) a quello (parziale) dato.



# Simulated Annealing

---

- Il *Simulated Annealing* è una tecnica di ottimizzazione per trovare il Minimo Assoluto di una funzione:
  - una tecnica standard è la *discesa del gradiente*: si può però restare intrappolati in minimi locali;
  - il *simulated annealing* simula una agitazione termica per sfuggire ai minimi locali.



# Simulated annealing: funzionamento

---

- vi sono tre elementi:
  1. Una densità di probabilità per generare le transizioni di stato;
  2. Una politica di *raffreddamento* (cooling);
  3. Una probabilità di accettazione;



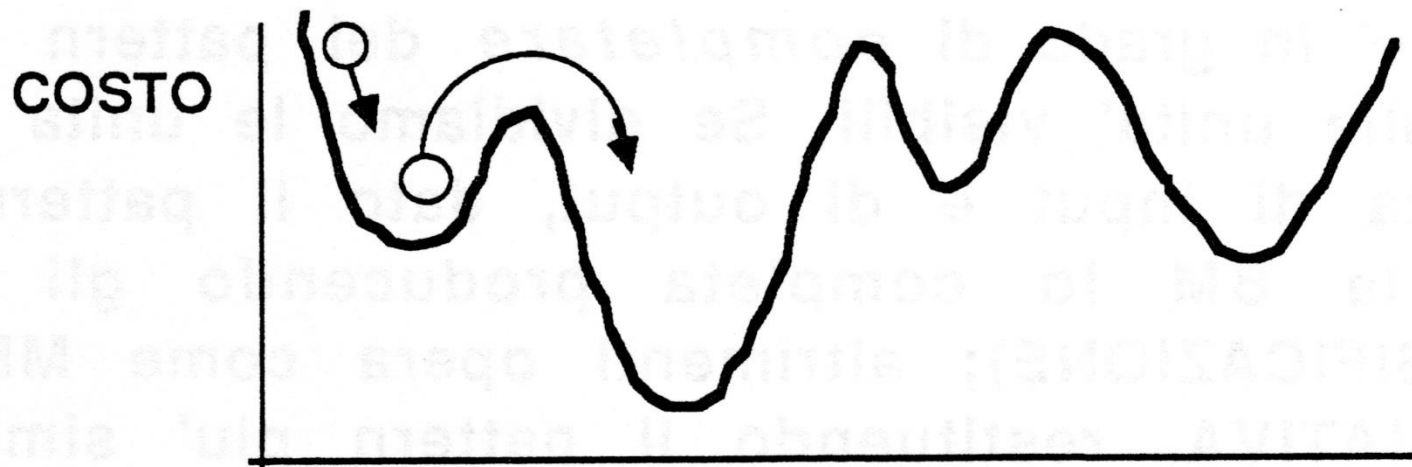
## Simulated annealing: funzionamento (2)

---

- Partendo da uno stato casuale, ad ogni passo si genera un nuovo stato secondo la probabilità di generazione.
- Se il nuovo stato ha costo più basso viene accettato; se ha un costo più alto viene accettato con una certa probabilità di accettazione. Le probabilità di generazione e di accettazione dipendono dalla temperatura.
- Si itera abbassando gradualmente la temperatura. Quando il sistema si stabilizza, esso si trova nel minimo globale della funzione costo da ottimizzare.



## Simulated annealing: funzionamento (3)





# Simulated Annealing e BM

---

- **SCOPO BM:** apprendere a SIMULARE un ambiente: dato un input la BM converge verso uno stato stabile (*si rilassa*) ricostruendo l'output corrispondente.
- **METODO di RILASSAMENTO:** SIMULATED ANNEALING: la rete converge verso lo stato a minima energia compatibile con i constraints (input e pesi).



# Simulated Annealing e BM (2)

---

SIMULATED ANNEALING su BM

- Funzione da minimizzare: ENERGIA TOTALE della BM

$$E = -\sum_{i < j} w_{ij} a_i a_j + \sum_i \theta_i a_i$$



# Simulated Annealing su BM (3)

---

- Probabilità di transizione:

*Distribuzione di Boltzmann:*

$$\frac{P_{\alpha}}{P_{\beta}} = e^{-(E_{\alpha} - E_{\beta}) / T}$$

dove  $P_{\alpha}$  è la probabilità dello stato globale  $\alpha$ , ed  $E_{\alpha}$  è l'energia totale di tale stato



# Simulated Annealing su BM (4)

---

ottenuta usando la Funzione di Attivazione:

$$p(a_k(t) = 1) = \frac{1}{1 + e^{-\Delta E_k / T}}$$

*con*

$$\Delta E_K = \sum_i w_{ik} a_i - \theta_k$$



# Simulated Annealing su BM (5)

---

All'equilibrio termico la probabilità relativa di due stati è determinata solamente dalla differenza delle loro energie: vi è una forte preferenza per gli stati a bassa energia, più marcata alle basse temperature.



# Simulated Annealing su BM (6)

---

- **Procedimento per ottenere l'OUTPUT di una BM dati gli INPUT (rilassamento mediante Simulated Annealing):**
  1. Gli input (ed eventualmente gli output) sono imposti alla rete (vettore di valori binari dati);
  2. si itera il seguente procedimento, cominciando con una temperatura elevata;



# Simulated Annealing su BM (7)

---

3. le unità aggiornano gli output per un certo intervallo di tempo;
4. la temperatura viene diminuita secondo la politica di raffreddamento;  
es.:  $T(t) = c / \log(1+t)$
5. quando la rete si stabilizza, si ferma il raffreddamento (equilibrio termico) e si leggono gli output.





# Simulated Annealing su BM (8)

---

Poiché la BM è una RN stocastica, gli output continuano ad evolvere anche all'equilibrio, senza cambiare gli input: occorre quindi collezionarli per un certo tempo per avere una stima della loro frequenza.

L'output reale della BM non è quindi un vettore binario, ma una distribuzione di probabilità sui possibili vettori binari.



# Boltzmann Machine: Algoritmo di Apprendimento

---

- Una BM può essere usata per modellare una realtà fisica (ambiente).
- Si dice che un insieme di PESI costituisce un MODELLO PERFETTO della struttura dell'ambiente se, quando la rete opera liberamente senza alcun input imposto, genera i vettori visibili esattamente con la stessa distribuzione di probabilità con cui si presentano nell'ambiente da modellare (dati di input).
- L'apprendimento avviene tramite la modifica dei pesi della rete, minimizzando una misura di distanza fra le prestazioni della rete e quelle dell'ambiente.



# Boltzmann Machine:

## Algoritmo di Apprendimento (2)

---

- Per far ciò si usa una misura presa dalla teoria dell'informazione (Kullback – 1959):

$$G = \sum_{\alpha} P^{+}(V_{\alpha}) \ln \frac{P^{+}(V_{\alpha})}{P^{-}(V_{\alpha})}$$

dove  $P^{+}(V_{\alpha})$  è la probabilità dello stato  $\alpha$  delle unità visibili quando la BM opera con gli input imposti, e  $P^{-}(V_{\alpha})$  è la corrispondente probabilità quando la rete opera liberamente.



# Boltzmann Machine:

## Algoritmo di Apprendimento (3)

---

- È stato dimostrato che, all'equilibrio termico, tutta l'informazione richiesta per cambiare il peso  $w_{ij}$  sta nel comportamento delle unità che esso connette  $u_i$  e  $u_j$ .

$$\frac{\partial G}{\partial w_{ij}} = -\frac{1}{T} [p_{ij}^+ - p_{ij}^-]$$



# Boltzmann Machine:

## Algoritmo di Apprendimento (4)

---

Dove:

- $p^+_{ij}$  = probabilità, mediata su tutti gli input dell'ambiente e misurata all'equilibrio, che le unità  $u_i$  e  $u_j$  siano entrambe attive ( $= 1$ ) quando la rete opera con l'input imposto dall'ambiente (fase<sup>+</sup>);
- $p^-_{ij}$  = la stessa probabilità, quando la rete opera liberamente senza input imposto (fase<sup>-</sup>).



# Boltzmann Machine: Algoritmo di Apprendimento (5)

---

- Si ottiene quindi la regola di modifica dei pesi:

$$\Delta w_{ij} = \eta \cdot \text{SIGN}(p_{ij}^{+} - p_{ij}^{-})$$

- L'algoritmo di apprendimento ha una interpretazione nei termini della regola di Hebb.



# Boltzmann Machine:

## Algoritmo di Apprendimento (6)

---

- L'idea è che la prestazione nella BM durante la fase<sup>+</sup> è determinata dall'input e dalla struttura interna, mentre durante la fase<sup>-</sup> è determinata solo dalla struttura interna.
- Per ottenere una prestazione dipendente solo dall'input "sottraiamo" la prestazione dovuta alla struttura interna: quindi,



# Boltzmann Machine:

## Algoritmo di Apprendimento (7)

---

- nella fase<sup>+</sup> si modificano i pesi con  $\Delta \mathbf{w}_{ij} = \eta \mathbf{a}_i \mathbf{a}_j$  (si incrementa la prestazione dovuta all'input e alla struttura interna)
  - nella fase<sup>-</sup> si modificano i pesi con  $\Delta \mathbf{w}_{ij} = -\eta \mathbf{a}_i \mathbf{a}_j$  (si decrementa la prestazione dovuta alla struttura interna)
- convergenza ad una prestazione dovuta solo all'input (**modello perfetto dell'ambiente**).





# Boltzmann Machine - osservazioni

---

- La BM è un modello più complesso di altri (es. MLP) sia come struttura (stocastica), che come modalità di generare l'output (il Simulated Annealing è più complesso della Forward propagation) e algoritmo di apprendimento.
- Gli autori propongono alcuni esempi (classificazione dello shift del vettore di input);
- Un'applicazione al riconoscimento delle classi vocaliche è stato proposto da De Mori (1987).
- Sono state riproposte in forma differente (Restricted Boltzmann Machine, RBM) da Hinton nel 2006.
- Interessanti perché presentano un modello "generativo" di un dominio.



## Boltzmann Machine – osservazioni (2)

---

- problemi della BM possono essere:
  - determinazione dei parametri dell'algoritmo di apprendimento (velocità di apprendimento  $\eta$ , numero di elementi necessari affinché le frequenze  $p^+_{ij}$  e  $p^-_{ij}$  siano significative;
  - lentezza dell'apprendimento;
  - input binari.



# Considerazioni che portano alle Deep NN

---

- Sono sufficienti 3 strati (2 strati di hidden) per risolvere qualunque problema di classificazione, ma ogni strato di hidden potrebbe dover avere un numero spropositato di nodi (1 nodo = 1 iperpiano di separazione)
- Questo porta ad un'esplosione del numero di pesi e a tempi di computazione in fase di addestramento lunghissimi.
- È stata dimostrata l'equivalenza funzionale tra uno strato largo e un insieme multistrato ma di dimensioni ridotte.

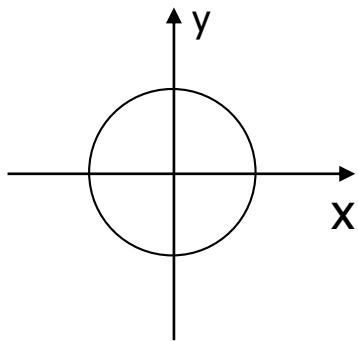


## Considerazioni che portano alle Deep NN (2)

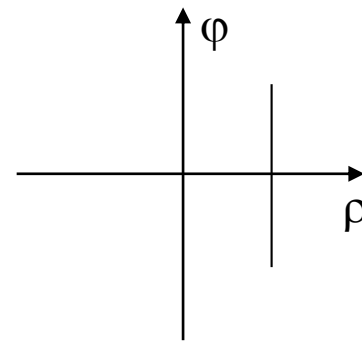
---

- A volte la proiezione di uno spazio in un altro equivalente semplifica la separazione tra classi. Esempio:

Coordinate cartesiane



Coordinate polari





## Considerazioni che portano alle Deep NN (3)

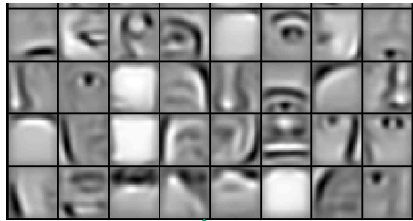
---

- In molte situazioni, si possono avere features più consistenti e altre più rumorose. Esempio: le cifre disegnate in una matrice di pixel. La rumorosità è correlata alla categoria di appartenenza.
- Correlato alla precedente: problemi di scalamento e di variabilità di forme (diversi modi di scrivere il numero 4, il 7, etc.).
- Si può richiedere che il sistema di classificazione mimi (quello che si crede essere) il comportamento umano (pixel – stroke – forme elementari – forme composte).

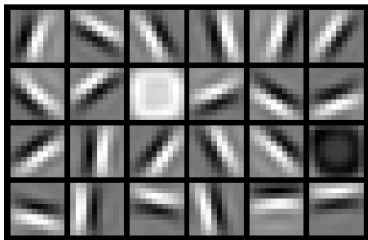
# Why feature hierarchies



Object models



Object parts  
(combination  
of edges)



edges



pixels





## Considerazioni che portano alle Deep NN (4)

---

Dal punto di vista della modellazione:

- si è dimostrato che una rete MLP, addestrata con la discesa del gradiente, corrisponde ad una rete bayesiana (trova l'ipotesi che giustifica al meglio, e quindi "spiega", i dati in input).
- Le reti bayesiane permettono con modelli semplici di fare inferenze (classificatori bayesiani).
- Servono delle trasformazioni dello spazio di input tali per cui alla fine la separazione delle classi possa avvenire con un limitato insieme di superfici di separazione, non sono posti vincoli su come questo avviene.



# Deep learning algorithms

---

- Stack sparse coding algorithm
- Deep Belief Network (DBN) (Hinton)
- Deep sparse autoencoders (Bengio)

[Other related work: LeCun, Lee, Yuille,  
Ng ...]