

# Rete di Hopfield applicata al problema del TSP

## **1. Introduzione**

Nel 1982, il fisico John J. Hopfield pubblicò un articolo fondamentale in cui presentò un modello matematico comunemente noto come rete di Hopfield. Tale rete si distinse per “ l'emergere spontaneo di nuove capacità computazionali dal comportamento collettivo di un gran numero di semplici elementi d'elaborazione ”. Hopfield ritenne che ogni sistema fisico possa essere considerato come un potenziale dispositivo di memoria, qualora esso disponga di un certo numero di stati stabili, i quali fungano da attrattore per il sistema stesso. Sulla base di questa considerazione, egli formulò la tesi secondo cui la stabilità e la collocazione di tali attrattori rappresentavano proprietà spontanee di sistemi costituiti da considerevoli quantità di neuroni reciprocamente interagenti [1].

## **2. La rete di Hopfield**

Da un punto di vista strutturale la rete di Hopfield costituisce una rete neurale ricorrente simmetrica (matrice dei pesi sinaptici simmetrica, quindi  $w_{ji} = w_{ij}$ ,  $\forall i,j$ ), completamente connessa, in cui ogni neurone è connesso a tutti gli altri, come mostrato in Figura 1.

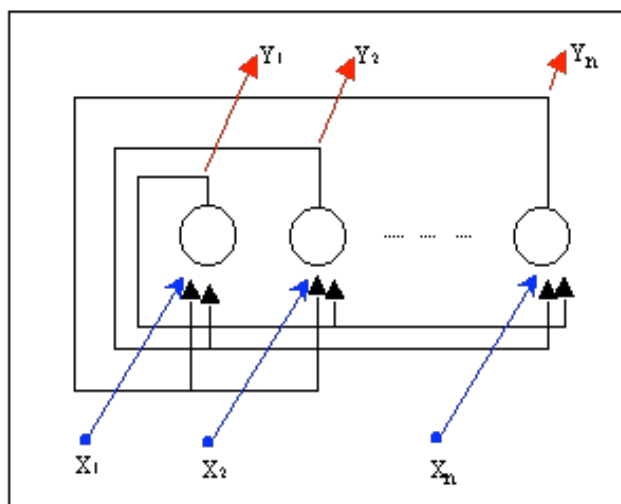


Fig.1

Una rete ricorrente è un modello neurale in cui è presente un flusso di informazioni bidirezionale; in altri termini, mentre nelle reti di tipo feedforward la propagazione

dei segnali avviene unicamente, in maniera continua, nella direzione che conduce dagli ingressi alle uscite, nelle reti ricorrenti tale propagazione può anche manifestarsi da uno strato neurale successivo ad uno precedente oppure tra neuroni appartenenti ad uno stesso strato (rete di Hopfield) e persino tra un neurone e se stesso.

La dinamica di questo modello di rete è descritta da un sistema non lineare di equazioni differenziali e il meccanismo di aggiornamento dei neuroni può essere:

- Aggiornamento asincrono: si aggiorna un neurone alla volta;
- Aggiornamento sincrono: tutti i neuroni sono aggiornati nello stesso istante;
- Aggiornamento continuo: tutti i neuroni si aggiornano continuamente.

### **3. Caso discreto**

La prima formulazione della rete di Hopfield fu una rete discreta con aggiornamento asincrono in cui egli dimostrò la globale stabilità se soddisfatte le seguenti condizioni:

- Rete completamente connessa;
- Matrice dei pesi simmetrica;
- Attivazione asincrona.

La rete è costituita da  $N$  neuroni, che sono utilizzati sia per l'input che per l'output dei dati. In questo modello, l'uscita di ogni neurone è collegata all'ingresso di tutti gli altri neuroni, formando così una topologia di rete a grafo completo unidirezionale. Ogni neurone può assumere due stati  $V_i = 0$  ("not firing") e  $V_i = 1$  ("firing at maximum rate"). Quando un neurone  $i$  è connesso ad un neurone  $j$ , la forza della connessione è determinata dal peso  $w_{ij}$ .

I pesi soddisfano i seguenti vincoli:

- i.  $w_{ii} = 0, \forall i$  (nessun neurone ha una connessione con se stesso);
- ii.  $w_{ij} = w_{ji}, \forall i, j$  (la matrice dei pesi è simmetrica).

Lo stato istantaneo del sistema è specificato elencando gli  $N$  valori di  $V_i$  che rappresentano quindi una parola binaria di  $N$  bits. Lo stato di ciascun neurone ad un certo istante è determinato da una formula di attivazione binaria a soglia, che modificherà i valori presentati in input nel modo seguente:

$$\begin{aligned}
V_i^t &\rightarrow 1 &> \theta_i \\
\text{If } \sum_j w_{ij} V_j^{t-1} &> \theta_i \\
V_i^t &\rightarrow 0 &< \theta_i
\end{aligned} \tag{1}$$

O in modo equivalente:

$$V_i^t = f_H(\sum_j w_{ij} V_j^{t-1} - \theta_i) \quad i = 1, \dots, N$$

dove  $\theta_i$  è in generale la soglia di attivazione associata al neurone  $i$ -esimo e  $f_H$  una funzione non lineare a soglia.

Tale legge sarà poi applicata in modo asincrono, cioè serialmente ad un'unità per volta, secondo un ordine prefissato o a scelta casuale. Questo processo sarà poi iterato fino a quando la rete non raggiungerà uno stato di equilibrio che corrisponderà ai valori di output.

A questo punto è lecito chiedersi da dove derivi la matrice dei pesi introdotta precedentemente, la quale determina il modo in cui si comporta la rete neurale. A tale proposito è importante sottolineare che, prima dell'utilizzo vero e proprio, la rete di Hopfield deve essere addestrata mediante un opportuno processo di training (o apprendimento). Questa fase preliminare è appunto quella che permette il calcolo della matrice dei pesi.

### 3.1. La matrice dei pesi

Il principio alla base della memorizzazione delle reti neurali è che l'informazione da memorizzare sia contenuta nei coefficienti di connessione. Nel modello di Hopfield questo lo si fa sfruttando la legge di Hebb, secondo cui “se un'unità  $u_i$  riceve un input da un'unità  $u_j$  e se entrambi sono fortemente attivi, il peso  $w_{ji}$  da  $u_j$  a  $u_i$  deve essere rafforzato”. Supponiamo di voler memorizzare il set di stati  $V^s$ ,  $s = 1, \dots, N$  utilizzando la notazione bipolare in cui  $0 \leftrightarrow -1$  e  $1 \leftrightarrow 1$ .

I pesi delle connessioni saranno assegnati secondo la seguente formula:

$$w_{ij} = \sum_s V_i^s V_j^s, \quad i \neq j, \quad i, j = 1, \dots, N \tag{2}$$

che rispetta perfettamente la regola di Hebb, infatti quando le due quantità sono uguali (1,1) o (-1, -1) il peso assegnato sarà positivo, viceversa quando ci troviamo nel caso (1, -1) o (-1,1) il peso assegnato sarà negativo.

### 3.2. La funzione energia

La rete di Hopfield può essere vista come un sistema dinamico, la cui stabilità può essere studiata introducendo un'opportuna *funzione di Lyapunov* (o funzione energia) definita nello spazio di stato.

La funzione energia introdotta per la rete di Hopfield (con variabili di stato unipolari) è definita come segue:

$$E = - \frac{1}{2} \sum_i \sum_j w_{ij} V_i V_j + \sum_i \theta_i V_i \quad (3)$$

La (3) è una funzione limitata inferiormente. Sulla base delle ipotesi (i)-(ii) e assumendo un aggiornamento di tipo asincrono, dimostriamo ora che la dinamica (1) della rete è tale che la funzione energia  $E$  risulta monotonicamente decrescente lungo una qualsiasi "traiettoria" descritta dalla rete di Hopfield nello spazio di stato, il che significa che la rete evolve verso una configurazione stabile.

**Proposizione** *Una rete di Hopfield con  $N$  unità ed una dinamica asincrona che parte da un dato stato della rete, converge ad uno stato stabile corrispondente ad un minimo locale della funzione energia.*

**Dimostrazione** Poiché la dinamica è asincrona, solo un'unità sarà valutata in ogni momento. Se la  $K$ -esima unità è selezionata, ci sono due opzioni:

1.  $V_K$  non cambia stato;
2.  $V_K$  cambia stato ( $-1 \rightarrow 1$  o  $1 \rightarrow -1$ ).

Nel primo caso la funzione energia rimane invariata:

$$E(t+1) = E(t)$$

Nel secondo caso la funzione energia cambia secondo il nuovo valore di eccitazione  $V'_K$  della  $K$ -esima unità. La differenza di energia del sistema è data da:

$$E(V) - E(V') = (-\sum_j w_{Kj} V_j V_K + \theta_K V_K) - (-\sum_j w_{Kj} V_j V'_K + \theta_K V'_K) \quad (4)$$

Poiché  $w_{KK} = 0$  per la (i):

$$E(V) - E(V') = -(V_K - V'_K) * (\sum_j w_{Kj} V_j - \theta_K) \quad (5)$$

Il secondo termine di questa equazione è l'eccitazione della  $K$ -esima unità secondo la (1). L'unità cambia il suo stato, quindi l'eccitazione ha un segno diverso che dipende da  $V_K$  e  $-V'_K$ . Si dimostra pertanto che la (5) è sempre positiva perché:

1. Se  $V_K$  cambia stato ( $-1 \rightarrow 1$ ) allora  $V_K = -1 \rightarrow V'_K = 1$

$$-(V_K - V'_K) > 0$$

2. Se  $V_K$  cambia stato ( $1 \rightarrow -1$ ) allora  $V_K = 1 \rightarrow V'_K = -1$

$$-(V_K - V'_K) > 0$$

Questo implica che  $E(V) > E(V')$ , quindi l'energia diminuirà per ogni cambiamento e poiché vi è un numero finito di stati possibili ( $2^N$ ), la rete raggiungerà uno stato in cui l'energia non potrà più diminuire quindi sarà uno stato stabile. Naturalmente lo stato stabile non sarà unico ma è certo che la rete converge (Fig.2).

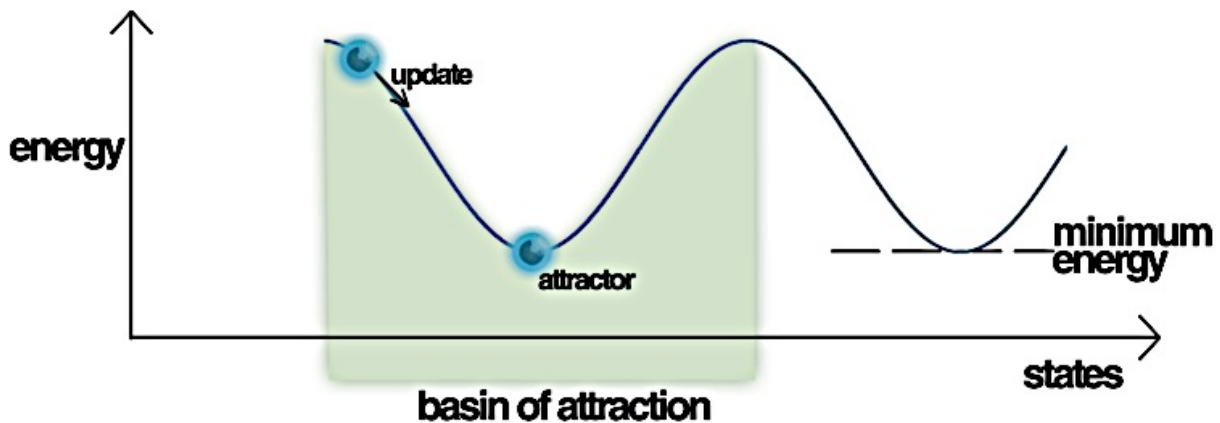


Fig.2

#### 4. Caso Continuo

La rete di Hopfield continua, pur essendo formata da un numero finito di neuroni, può assumere un numero infinito di stati, poiché è caratterizzata da una funzione di attivazione sigmoide (oppure tangente iperbolica).

$$V_i(t) = kV_i(t-1) + \alpha \sum_{j=1}^N w_{ij} F_j(t)$$

$$k, \alpha \in [0,1] \text{ cost. di attenuazione}$$

$$F_j(t) = \frac{1}{1 + e^{-\frac{V_j(t)}{T}}} \text{ f. sigmoidea}$$

Lo stato della rete all'istante  $t$  è sempre rappresentato da un vettore  $V$ , composto dagli  $N$  stati  $V_i$  dei singoli neuroni, i cui valori sono compresi nell'intervallo reale  $[0,1]$  (notazione unipolare) oppure  $[-1,1]$  (notazione bipolare).

Ciò porta ad avere la seguente funzione di energia leggermente diversa rispetto alla precedente:

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} v_i v_j + \sum_{i=1}^N \frac{1}{\beta} \int_0^{v_i} F_i^{-1}(v) dv - \sum_{i=1}^N \theta_i v_i$$

$$F_i(\theta) = \frac{1}{1 + e^{-\frac{\theta}{T}}} \quad \text{f. di attivazione (sigmoide)}$$

Così come nel caso discreto, se la matrice dei pesi è simmetrica allora la funzione è Lyapunoviana e quindi  $dE/dt \leq 0$ , dove in caso di uguaglianza ci troviamo in un punto stazionario.

Un'altra caratteristica interessante delle reti di Hopfield continue rispetto a quelle discrete, è la possibilità di essere facilmente implementate e studiate come *circuiti elettronici*.

Le applicazioni delle reti di Hopfield riguardano principalmente la realizzazione di memorie associative, resistenti all'alterazione delle condizioni operative e la soluzione di problemi di ottimizzazione combinatoriale.

Un problema di ottimizzazione combinatoria è caratterizzato dal fatto di avere un numero finito di soluzioni ammissibili. La difficoltà intrinseca di molti problemi di ottimizzazione combinatoria ne impedisce una soluzione esatta e quindi si richiede che si adoperino tecniche euristiche in grado di fornire soluzioni "buone" seppur non ottimali.

## **5. Problema del TSP (Traveling Salesman's Problem)**

La formulazione del TSP è molto semplice: *siano assegnate n città A, B, C,..., e le distanze  $d_{ij}$  fra esse esistenti, il problema consiste nella determinazione di una sequenza di città da visitare in modo che ogni città venga visitata una sola volta, minimizzando il percorso seguito e ritornando alla città di partenza.*

Il problema può essere risolto mediante una rete di Hopfield. Si associ ad ogni città un numero di unità pari ad n, ciascuna corrispondente ad una delle n posizioni che costituiscono la sequenza di città da visitare. La presenza di una determinata città in i-esima posizione corrisponde ad un valore 1 di uscita  $V_i$  dell'unità relativa alla città e alla posizione i considerata. Se il numero delle città è 6 e la città A occupa la seconda posizione nella sequenza finale, la configurazione degli stati delle unità ad essa associata è la seguente: A 0 1 0 0 0.

Per  $n$  città la rete deve essere formata da  $N = n^2$  neuroni. Lo stato di uscita di queste  $N$  unità può essere descritto mediante una matrice quadrata di dimensioni  $(n \times n)$ . Riferendosi all'esempio in cui si hanno 5 città la matrice avrà la forma come in Fig.3:

città \	posizioni					
	1	2	3	4	5	6
A	0	0	0	0	1	0
B	0	1	0	0	0	0
C	0	0	0	1	0	0
D	1	0	0	0	0	0
E	0	0	0	0	0	1
F	0	0	1	0	0	0

Fig.3

In questo caso la sequenza finale rappresentante la soluzione del problema è  $D \Rightarrow B \Rightarrow F \Rightarrow C \Rightarrow A \Rightarrow E$  mentre la lunghezza del cammino è  $d_{db} + d_{bf} + d_{fc} + d_{ca} + d_{ae} + d_{ed}$ .

Una matrice che rappresenta una soluzione corretta è caratterizzata da un solo elemento pari ad 1 in ogni riga e colonna dato che una città può essere visitata solo una volta. Le  $2^N$  configurazioni possibili si riducono quindi a un sottoinsieme pari al numero delle permutazioni delle  $n$  righe della matrice,  $n!$ . Le possibili sequenze inoltre non sono tutte distinte, poiché si considerano coincidenti sia sequenze del tipo DBFCAE, BFCAED, CAEDBF e così via, sia le sequenze DBFCAE ed EACFBD, per cui il numero di soluzioni distinte del problema diventa  $n!/2n$ .

### 5.1. Input della rete e matrice delle distanze

Gli ingressi della rete sono scelti arbitrariamente. Lo stato iniziale della rete non è quindi fissato a priori e non favorisce nessun percorso particolare. In questo tipo di modello può sorgere però un problema nel caso in cui la rete converga in un minimo locale piuttosto che in un minimo globale. Per evitare ciò, quindi, viene generato un rumore casuale e aggiunto allo stato iniziale.

La matrice delle distanze è una matrice  $D \in \mathbb{R}^{n \times n}$  dove la  $(x, y)$ -esima componente indica la distanza tra la  $x$ -esima e la  $y$ -esima città e con diagonale nulla, perché la

distanza da una città a se stessa è 0. Indicheremo con  $d_{x,y}$  la generica componente di D. Un esempio di matrice delle distanze è rappresentato in Fig.4.

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Fig.4

## 5.2. La funzione energia

La funzione energia ha varie cavità (avvallamenti) che rappresentano i pattern memorizzati nella rete. Un pattern di input sconosciuto rappresenta un particolare punto nel “*landscape*” energetico e tramite le iterazioni della rete, il punto si sposta attraverso il paesaggio verso una delle cavità (soluzione). Le iterazioni continueranno finché non si raggiungerà o un fissato intervallo di tempo o uno stato stabile.

Indichiamo con  $V \in \{0,1\}^{n \times n}$  la matrice di stato del sistema dove la  $(x, i)$ -esima componente è pari ad 1 se la  $x$ -esima città viene attraversata al  $i$ -esimo step del tour, 0 altrimenti. Indicheremo con  $v_{x,i}$  una generica componente di  $V$ .

La funzione energia utilizzata dovrà soddisfare i seguenti criteri:

- i. **Vincolo di riga:** ogni città dovrà essere visitata al massimo 1 volta, vogliamo quindi che le righe della matrice  $V$  abbiano soltanto un elemento settato ad 1 e il resto a 0;
- ii. **Vincolo di colonna:** ogni step del tour dovrà contenere al massimo una città, vogliamo quindi che ogni colonna della matrice  $V$  abbia un solo elemento settato ad 1 ed il resto a 0;
- iii. **Vincolo globale:** in tutto il percorso dovranno essere attraversate  $n$  città.
- iv. **Vincolo sulla distanza:** la distanza totale percorsa dovrà essere la più corta.



I vincoli appena descritti portano alla seguente espressione per la funzione energia:

$$E_1 = \frac{A}{2} \cdot \sum_x \sum_i \sum_{j \neq i} v_{x,i} \cdot v_{x,j}$$

(i) Vale 0 se ogni città è visitata al massimo una volta;

$$E_2 = \frac{B}{2} \cdot \sum_i \sum_x \sum_{y \neq x} v_{x,i} \cdot v_{y,i}$$

(ii) Vale 0 se ogni step del tour contiene al massimo una città;

$$E_3 = \frac{C}{2} \cdot \left( \sum_x \sum_i v_{x,i} - n \right)^2$$

(iii) Vale 0 se il numero di città attraversate è esattamente n;

$$E_4 = \frac{D}{2} \cdot \sum_x \sum_y \sum_{i=0}^n d_{x,y} \cdot v_{x,i} \cdot (v_{y,i-1} + v_{y,i+1})$$

(iv) Il valore di questo termine è minimo quando la distanza percorsa è la più breve. A questo punto una soluzione ottima del TSP richiederebbe che le 4 funzioni energia vengano minimizzate, noi abbiamo però bisogno di un'unica funzione energia, per questo esprimiamo la funzione costo totale come combinazione lineare delle funzioni energia  $E_i$ :

$$E = \underbrace{\frac{A}{2} \sum_x \sum_i \sum_{j \neq i} v_{x,i} v_{x,j}}_{E_1} + \underbrace{\frac{B}{2} \sum_i \sum_x \sum_{y \neq x} v_{x,i} v_{y,i}}_{E_2} + \underbrace{\frac{C}{2} \left( \sum_x \sum_i v_{x,i} - N \right)^2}_{E_3} + \underbrace{\frac{D}{2} \sum_x \sum_{x \neq y} \sum_i d_{x,y} v_{x,i} (v_{y,i+1} + v_{y,i-1})}_{E_4}$$

Dobbiamo ora ricondurci alla funzione di energia nella forma vista in precedenza (3), tenendo conto stavolta di un ulteriore input esterno I:

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} V_i V_j - \sum_i I_i V_i$$

Per fare questo indichiamo con  $w_{(x,i),(y,j)}$  la  $(x + i \cdot n, y + i \cdot n)$ -esima componente della matrice dei pesi e con  $I_{x,i}$  la  $(x + i \cdot n)$ -esima componente del vettore input I. Quindi la forma a cui dovremmo arrivare è:

$$E = -\frac{1}{2} \cdot \sum_x \sum_y \sum_i \sum_j w_{(x,i),(y,j)} \cdot v_{x,i} \cdot v_{y,j} - \sum_x \sum_i I_{x,i} \cdot v_{x,i}$$

Introduciamo la funzione Delta di Kronecker, per facilitare la notazione:

$$\delta_{i,j} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

e trasformiamo ciascuna delle funzioni energia  $E_i$  nella forma sopra indicata.

$$\begin{aligned} E_4 &= \frac{D}{2} \cdot \sum_x \sum_y \sum_{i=0}^n d_{x,y} \cdot v_{x,i} \cdot (v_{y,i-1} + v_{y,i+1}) = -\frac{1}{2} \cdot \sum_x \sum_y \sum_i \sum_j -D \cdot d_{x,y} \cdot v_{x,i} \cdot v_{y,j} \cdot (\delta_{j,i-1} + \delta_{j,i+1}) \\ E_1 &= \frac{A}{2} \cdot \sum_x \sum_i \sum_{j \neq i} v_{x,i} \cdot v_{x,j} = -\frac{1}{2} \cdot \sum_x \sum_y \sum_i \sum_j -A \cdot v_{x,i} \cdot v_{y,j} \cdot \delta_{x,y} \cdot (1 - \delta_{i,j}) \\ E_2 &= \frac{B}{2} \cdot \sum_i \sum_x \sum_{y \neq x} v_{x,i} \cdot v_{y,i} = -\frac{1}{2} \cdot \sum_x \sum_y \sum_i \sum_j -B \cdot v_{x,i} \cdot v_{y,j} \cdot \delta_{i,j} \cdot (1 - \delta_{x,y}) \\ E_3 &= \frac{C}{2} \cdot \left( \sum_x \sum_i v_{x,i} - n \right)^2 = -\frac{1}{2} \cdot \sum_x \sum_y \sum_i \sum_j -C \cdot v_{x,i} \cdot v_{y,j} - \sum_x \sum_i C \cdot n \cdot v_{x,i} - \frac{C}{2} \cdot n^2 \end{aligned}$$

In  $E_3$  tralasciamo il termine costante alla fine poiché non modifica il punto in cui abbiamo il minimo ma solo il valore della funzione  $E_3$  in corrispondenza del minimo, cosa che a noi non interessa, dal momento che il nostro scopo è minimizzare le 4 funzioni. Mettendo il tutto assieme otteniamo:

$$E = -\frac{1}{2} \cdot \sum_x \sum_y \sum_i \sum_j v_{x,i} \cdot v_{y,j} \left[ -D \cdot d_{x,y} \cdot (\delta_{j,i-1} + \delta_{j,i+1}) - A \cdot \delta_{x,y} \cdot (1 - \delta_{i,j}) - B \cdot \delta_{i,j} \cdot (1 - \delta_{x,y}) - C \right] - \sum_x \sum_i C \cdot n \cdot v_{x,i}$$

da cui segue che possiamo risolvere in modo approssimato il problema del TSP utilizzando la matrice dei pesi  $W$  che ha come  $(x + i \cdot n, y + i \cdot n)$ -esima componente:

$$w_{(x,i)(y,j)} = \left[ -D \cdot d_{x,y} \cdot (\delta_{j,i-1} + \delta_{j,i+1}) - A \cdot \delta_{x,y} \cdot (1 - \delta_{i,j}) - B \cdot \delta_{i,j} \cdot (1 - \delta_{x,y}) - C \right]$$

dove il primo termine garantisce il percorso a minima distanza, il secondo il vincolo di riga, il terzo il vincolo di colonna e il quarto il numero di città totali attraversate. Il vettore input esterno sarà invece il vettore  $I$  che avrà come  $(x + i \cdot n)$ -esima componente:

$$I_{(x,i)} = n \cdot C$$

### 5.3. Funzione di attivazione e di output

La funzione di attivazione segue anch'essa vari vincoli per ottenere un percorso valido e può essere definita come segue:

$$a_{ij} = \Delta t \left( -\frac{a_{ij}}{\tau} - A \sum_{j \neq i} V_{xj} - B \sum_{y \neq x} V_{yi} - C \left[ \sum_x \sum_j V_{xj} - m \right] - D \sum_y d_{xy} [V_{y,i+1} + V_{y,i-1}] \right)$$

Dove  $a_{ij}$  rappresenta l'attivazione per l'unità in riga  $i$ -esima e colonna  $j$ -esima,  $\tau$  e  $m$  rappresentano rispettivamente una costante di tempo e un parametro; il primo termine della funzione di attivazione decresce ad ogni iterazione, il secondo, terzo, quarto e quinto rappresentano i vincoli per ottenere un tour valido.

La funzione di attivazione è aggiornata come segue:

$$a_{ij(new)} = a_{ij(old)} + \Delta a_{ij}$$

La funzione di uscita invece è la seguente:

$$V_{xi} = (1 + \tanh(\lambda a_{ij})) / 2 \quad \forall x, i, j$$

dove  $V_{xi}$  è l'output del neurone, il valore di  $\lambda$  determina la pendenza della funzione e la tangente iperbolica restituisce un output come quello mostrato in Fig.5.

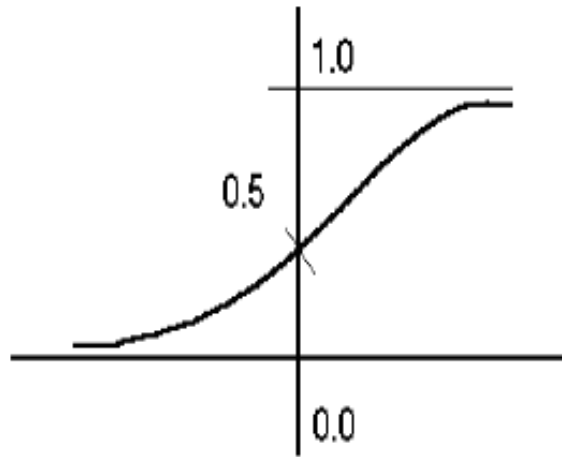


Fig.5

## 6. Conclusioni

Per risolvere con la rete di Hopfield un problema di ottimizzazione combinatoria occorre stabilire una corrispondenza tra variabili di stato della rete e variabili del problema, tra la funzione energia e la funzione obiettivo. L'evoluzione dinamica della rete, a partire dallo stato iniziale, fa diminuire la funzione energia fin quando

non si arriva in uno stato che rappresenta un minimo locale della funzione energia e che, vista la corrispondenza tra funzione energia e funzione obiettivo, potrebbe costituire una “buona” soluzione del problema combinatorio.

Tuttavia, come detto prima, l'ottimalità non è sempre garantita e il sistema potrebbe cadere in minimi locali. Il risultato potrebbe essere migliorato aggiungendo rumore alle dinamiche di rete, per creare una sorta di differenziazione nella speranza che il sistema evolva nonostante la vicinanza ad un minimo locale.

Ad ogni modo questo tipo di rete è molto utile nella risoluzione di problemi in cui un'unità può essere vista come un piccolo processore a cui le reti neurali possono offrire elevata potenza di calcolo e parallelismo (grazie all'aggiornamento asincrono della rete).

## **Riferimenti**

[1]: [http://it.wikipedia.org/wiki/Rete\\_neurale#Reti\\_di\\_Hopfield](http://it.wikipedia.org/wiki/Rete_neurale#Reti_di_Hopfield)

[2]: J. J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities* (1982)

[3]: J. J. Hopfield and D. W. Tank, *Computing with Neural Circuits: A model* (1986)

[4]: J. J. Hopfield and D. W. Tank, "Neural" Computation of Decisions in Optimization Problems (1985)