Chapter 5 Feedback Neural Network

5.1 Hopfield neural network

5.2 BAM network

5.1 Hopfield neural network

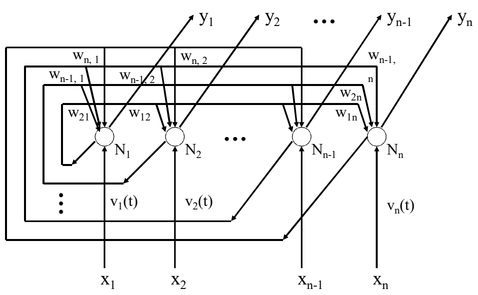
- In 1982, Hopfield J J brought forward a single layer feedback network composed of nonlinear component.
- Feedback network is a nonlinear dynamic system.
  - Nonlinear difference equation → Discrete HNN (DHNN)
  - differential equation → Continuous HNN (CHNN)

Applications: associative memory or classification

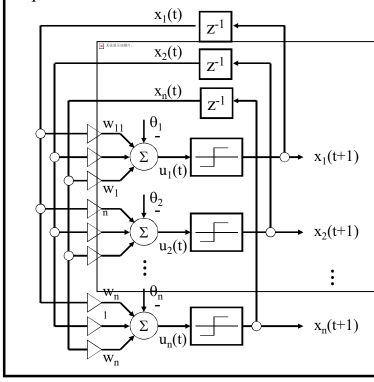
computing for optimization

5.1.1 Discrete Hopfield Neural Network

A. Hopfield network architecture



Hopfield network architecture 2



Computing formula:

$$u_i(t)=\sum_{j=1}^{n} w_{ij}x_j(t)-\theta_i$$

$$x_i(t+1)=\operatorname{sgn}[u_i(t)]$$

$$i=1,2,\dots,n$$

$$\operatorname{sgn}(x)=\begin{cases}1 & x\ge 0\\ -1 & x<0\end{cases}$$

B. Working principle

1. Asynchronous(series) pattern

$x_i(t+1)=\operatorname{sgn}[u_i(t)]$

$x_j(t+1)=x_i(t)$ ($j\neq i$)

2. Synchronous(parallel) pattern

$x_i(t+1)=\operatorname{sgn}[u_i(t)]$ ($i=1,2,\dots,n$)

3. Network stability

$$x_i(t+1)=x_i(t)=\operatorname{sgn}\left[\sum_{j=1}^{n} w_{ij}x_i(t)-\theta_i\right]$$

$$i=1,2,\dots,n$$

3. Stability analysis

Using "energy function"

$$E=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_i x_j+\sum_{i=1}^{n}\theta_i x_j$$

Matrix format:

$$E=-\frac{1}{2}X^T W X+X^T\theta$$

x=1 or –1, w and θ are limited constant, so the energy function is limited constant.

$$|E|\le\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}|w_{ij}||x_i||x_j|+\sum_{i=1}^{n}|\theta_i||x_i|$$

$$=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}|w_{ij}|+\sum_{i=1}^{n}|\theta_i|$$

If ΔE=E(t+1)-E(t) ≤ 0 then the network converges to a steady status.

Assumption: series pattern, $w_{ij}=w_{ji}$, $w_{kk}\geq 0$

$\Delta E=E(t+1)-E(t)$

$\Delta x_k=x_k(t+1)-x_k(t)$ only k^{th} node is enabled

(假设取得是异步工作方式，t时刻只有第k个神经元调整状态)

$$\because x_k(\bullet)=\pm 1$$

$$\therefore \quad \Delta x_k=\begin{cases} 0 & x_k(t)=\mathrm{sgn}\left[u_k(t)\right]=x_k(t+1) \\ -2 & x_k(t)=1,\ \ \mathrm{sgn}\left[u_k(t)\right]=-1 \\ 2 & x_k(t)=-1,\ \ \mathrm{sgn}\left[u_k(t)\right]=1 \end{cases}$$

$$\Delta E=-\frac{1}{2}\left[\Delta x_k\sum_{i=1}^{n}w_{ik}x_i(t)+\Delta x_k\sum_{j=1}^{n}w_{kj}x_j(t)+w_{kk}(\Delta x_k)^2\right]+\Delta x_k\theta_k$$

$$\because w_{ij}=w_{ji} \quad \therefore \quad \Delta E=-\Delta x_k\left[\sum_{j=1}^{n}w_{kj}x_j(t)-\theta_k\right]-\frac{1}{2}w_{kk}(\Delta x_k)^2$$

$$=-\Delta x_k u_k(t)-\frac{1}{2}w_{kk}(\Delta x_k)^2$$

$$\because \quad \Delta x_k u_k(t)=\left[x_k(t+1)-x_k(t)\right]u_k(t)$$

$$=\left\{\mathrm{sgn}\left[u_k(t)\right]-\mathrm{sgn}\left[u_k(t-1)\right]\right\}u_k(t)$$

$$=\mathrm{sgn}\left[u_k(t)\right]u_k(t)-\mathrm{sgn}\left[u_k(t-1)\right]u_k(t)$$

$$\geq 0$$

$w_{kk}\geq 0$

$\therefore \quad \Delta E\leq 0$

The energy function is limited constant, therefore the network can converge to a local minimum point.

5.1.2 Associative Memory

A. Auto-association Memory

The M samples ($\{x^i\}$, i=1,2,…,M) are stored in the network through learning process.

If input $x'=x^{\alpha}+v$, x^{α} is one of the M samples

v is bias item

Then output $Y=x^{\alpha}$

B. Hetero-association Memory

The relationship between two samples:

$X^i \rightarrow y^i$, i=1,2,…,M

For example, x^i stand for a man's picture,

y^i stand for his name.

If input $x'=x^{\alpha}+v$ then output $Y=y^{\alpha}$

Network computing phases:

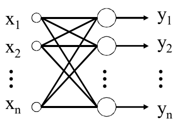
Learning : forming W(correspond to $x^1, x^2,\ldots, x^p$)

Initial: $v(0)=x$ (assuming input $x=x^k$)

Running: $v(t+1)=\mathrm{sgn}(v(t)W)$

Steady output: $v=x^k$

C. Learning rule



$$X^k=\left[x_1^k, x_2^k,\ldots,x_n^k\right]^T$$

k=1,2,…,m

that is $x_i^k=\pm 1$, i=1,2,…,n

Transfer function:

f(z)=sgn(z)

Outer product rule:

$$W=\sum_{k=1}^{m}[X^K(X^k)^T-I]$$

I is n×n unit matrix, $w_{ii}=0$

$$W=\sum_{k=1}^{m}\begin{bmatrix}x_1^k\\ \vdots\\ x_n^k\end{bmatrix}\begin{bmatrix}x_1^k & \ldots & x_n^k\end{bmatrix}-mI$$

$$=\begin{bmatrix}\sum_{k=1}^{m}(x_1^k)^2 & \sum_{k=1}^{m}x_1^kx_2^k & \ldots & \sum_{k=1}^{m}x_1^kx_n^k \\ \sum_{k=1}^{m}x_2^kx_1^k & \sum_{k=1}^{m}(x_2^k)^2 & \ldots & \sum_{k=1}^{m}x_2^kx_n^k \\ \vdots & \vdots & \ldots & \vdots \\ \sum_{k=1}^{m}x_n^kx_1^k & \sum_{k=1}^{m}x_n^kx_2^k & \ldots & \sum_{k=1}^{m}(x_n^k)^2\end{bmatrix}-m\begin{bmatrix}1 & & \ldots & 0 \\ & 1 & \ldots & \\ \vdots & \vdots & \vdots & \vdots \\ 0 & & \ldots & 1\end{bmatrix}$$

$$\because \quad \sum_{k=1}^{m}(x_1^k)^2 = \sum_{k=1}^{m}(x_2^k)^2 = \dots = \sum_{k=1}^{m}(x_n^k)^2 = m$$

$$\therefore \quad W = \begin{bmatrix} 0 & \sum_{k=1}^{m} x_1^k x_2^k & \dots & \sum_{k=1}^{m} x_1^k x_n^k \\ \sum_{k=1}^{m} x_2^k x_1^k & 0 & \dots & \sum_{k=1}^{m} x_2^k x_n^k \\ \vdots & \vdots & \dots & \vdots \\ \sum_{k=1}^{m} x_n^k x_1^k & \sum_{k=1}^{m} x_n^k x_2^k & & 0 \end{bmatrix}$$

$$\therefore \quad w_{ij} = w_{ji}, \quad w_{ii} = 0 \qquad (i, j = 1,2,\dots,n)$$

$$w_{ij} = \begin{cases} \sum_{k=1}^{m} x_i^k x_j^k & i \neq j \\ 0 & i = j \end{cases}$$

Hopfield network association and recall algorithm

*** Learning phase**

1. Give m samples $x^k = \{x_1^k, x_2^k, \dots, x_n^k\}$, k=1,2,…,m
   set W=[0], k=0, begin to iteration
2. Calculate $w_{ij}(k+1)=w_{ij}(k)+x_i^{k+1} x_j^{k+1}$, k=1,2,…,m
3. k=k+1
   if the number of iteration less than M then goto 2 else end.

*** Association and recall phase**

1. input vector x^1, $x_i(0)=x_i$, $1 \le i \le N$
2. Iteration
   $$x_j(t+1) = f_j\{\sum_{i=1}^{N} w_{ij} x_i(t)\} \qquad j = 1,2,\dots,n$$
3. If not convergnce goto 1

Example 1

$x=[\,1\ \ -1\ \ 1\,]^T$

Weight factors matrix (outer product rule):

$$W = XX^T - I = \begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

$$y = f(WX) = f(\begin{bmatrix} 0 & -1 & 1 \\ -1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix} \bullet \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix})$$

$$= f(\begin{bmatrix} 2 \\ -2 \\ 2 \end{bmatrix}) = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$y=[\,1\ \ -1\ \ 1\,]^T$

Network associative function taking into account the input with aberrance

(1) $x_1=[\,0\ \ -1\ \ 1\,]^T$

$$y_1 = f(WX_1) = f([2 \quad -1 \quad 1]^T) = [1 \quad -1 \quad 1]^T$$

(2) $x_2=[\,0\ \ 0\ \ 1\,]^T$

$$y_2 = f(WX_2) = f([1 \quad -1 \quad 0]^T) = [1 \quad -1 \quad 1]^T$$

(3) $x_3=[\,0\ \ 0\ \ 0\,]^T$

$$y_3 = f(WX_3) = f([0 \quad 0 \quad 0]^T) = [1 \quad 1 \quad 1]^T$$

Example 2

$$x^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad x^{(2)} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

$$W = x^{(1)}x^{(1)T} + x^{(2)}x^{(2)T} - 2I = \begin{bmatrix} 0 & 2 & 2 & 2 \\ 2 & 0 & 2 & 2 \\ 2 & 2 & 0 & 2 \\ 2 & 2 & 2 & 0 \end{bmatrix}$$

Recall and check

$$f(Wx^{(1)}) = f\begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = x^{(1)}$$

$$f(Wx^{(2)}) = f\begin{bmatrix} -6 \\ -6 \\ -6 \\ -6 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = x^{(2)}$$

Association memory function

① $x(0) = x^{(3)} = [-1 \quad 1 \quad 1 \quad 1]^T$

Using asynchronous(series) pattern, the sequence is 1,2,3,4

$$x_1(1) = f\left(\sum_{j=1}^{n} w_{1j} x_j(0)\right) = f(6) = 1$$

$$x_2(1) = x_2(0) = 1 \qquad x_3(1) = x_3(0) = 1 \qquad x_4(1) = x_4(0) = 1$$

$$\therefore \quad x(1) = [1 \quad 1 \quad 1 \quad 1]^T = x^{(1)}$$ Only one step converge to $x^{(1)}$

② $x(0) = x^{(4)} = [1 \quad -1 \quad -1 \quad -1]^T$

Using asynchronous pattern, the sequence is 1,2,3,4

$$x_1(1) = f\left(\sum_{j=1}^{n} w_{1j} x_j(0)\right) = f(-6) = -1$$

$$x_2(1) = x_2(0) = -1 \qquad x_3(1) = x_3(0) = -1 \qquad x_4(1) = x_4(0) = -1$$

$$\therefore \quad x(1) = [-1 \quad -1 \quad -1 \quad -1]^T = x^{(2)}$$ Only one step converge to $x^{(2)}$

③ $x(0) = x^{(5)} = [1 \quad 1 \quad -1 \quad -1]^T$

The iteration sequence is 1,2,3,4.

Step 1

$$x_1(1) = f\left(\sum_{j=1}^{n} w_{1j} x_j(0)\right) = f(-2) = -1$$

$$x_i(1) = x_i(0) \qquad i = 2,3,4$$

$that \ \ is \qquad x(1) = [-1 \quad 1 \quad -1 \quad -1]^T$ Not converge to $x^{(1)}$ or $x^{(2)}$

Step 2

$$x_2(2) = f\left(\sum_{j=1}^{n} w_{2j} x_j(1)\right) = f(-6) = -1$$

$$x_i(2) = x_i(1) \qquad i = 1,3,4$$

$$x(2) = [-1 \quad -1 \quad -1 \quad -1]^T = x^{(2)}$$ Converge to $x^{(2)}$

The iteration sequence is 3,4,1,2

Step 1

$$x_3(1) = f\left(\sum_{j=1}^{n} w_{3j} x_j(0)\right) = f(2) = 1$$

$$x_3(1) = x_i(0) \qquad i = 1,2,4$$

$x(1)=[1 \ 1 \ 1 \ -1]^T$ not converge to input vector

Step 2

$$x_4(2) = f\left(\sum_{j=1}^{n} w_{4j} x_j(1)\right) = f(6) = 1$$

$$x_i(2) = x_i(1) \qquad i = 1,2,3$$

$x(2)=[1 \ 1 \ 1 \ 1]^T=x^{(1)}$ converge to $x^{(1)}$

Synchronous（parallel） **computing**

① $x(0) = x^{(3)} = [-1 \quad 1 \quad 1 \quad 1]^T$

$$x(1) = f(Wx(0)) = f(Wx^{(3)}) = f\begin{bmatrix} 6 \\ 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$x(2) = f(Wx(1)) = f\begin{bmatrix} 6 \\ 6 \\ 6 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Converge to $x^{(1)}$

② $x(0) = x^{(4)} = [1 \quad -1 \quad -1 \quad -1]^T$

$$x(1) = f(Wx(0)) = f(Wx^{(4)}) = f\begin{bmatrix} -6 \\ -2 \\ -2 \\ -2 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

$$x(2) = f(Wx(1)) = f\begin{bmatrix} -6 \\ -6 \\ -6 \\ -6 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

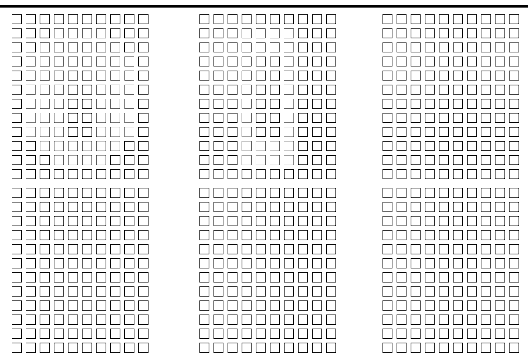
Converge to $x^{(2)}$

③ $x(0) = x^{(5)} = [1 \quad 1 \quad -1 \quad -1]^T$

$$x(1) = f(Wx(0)) = f(Wx^{(5)}) = f\begin{bmatrix} -2 \\ -2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

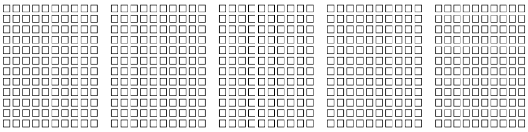
$$x(2) = f(Wx(1)) = f\begin{bmatrix} 2 \\ 2 \\ -2 \\ -2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

Oscillate between two status



每个模式为12*10大小的黑白图象，120个像素，+1表示像素为黑，-1为白。
网络规模N=120，连接数为N^2-N=12280个权。

先用外积法学习权值，然后用基本向量作为输入。经试验，所有基本向量都是网络的稳定状态。为检查其纠错能力，对基本向量加噪声污染，方法是每一像素都以0.25的概率翻转（-1改为1或反之）。测试结果如下。

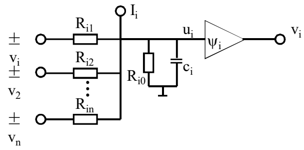


污染的6　10次迭代后　20次迭代后　30次迭代后　37次迭代后

模式：	0	1	2	3	4	6
迭代次数：	34	32	26	37	25	37

5.1.3 Continuous Hopfield Neural Network

Hopfield dynamic neural model

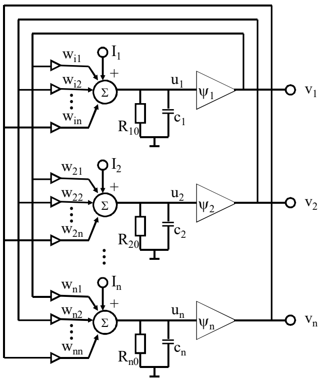


I_i — external input signal

$$v_i = \psi_i(u_i) = \frac{1}{1+e^{-u_i}}$$

$$v_i = \psi_i(u_i) = \tanh(u_i)$$

CHNN Model



$$c_i\frac{du_i}{dt} + \frac{u_i}{R_{i0}} = \sum_{j=1}^{n}\frac{1}{R_{ij}}(v_j - u_i) + I_i$$

$$c_i\frac{du_i}{dt} = \sum_{j=1}^{n}\frac{v_j}{R_{ij}} - \left(\frac{1}{R_{i0}} + \sum_{j=1}^{n}\frac{1}{R_{ij}}\right)u_i + I_i$$

$$= -\frac{u_i}{R_i} + \sum_{j=1}^{n} w_{ij}v_j + I_i$$

$$\frac{du_i}{dt} = -\frac{u_i}{\tau_i} + \sum_{j=1}^{n} w_{ij}v_j + \theta_i$$

$$w_{ij} = \frac{1}{R_{ij}c_i} \qquad \theta_i = \frac{I_i}{c_i} \qquad v_i = \psi_i(u_i)$$

If $du_i/dt=0$ then $u=\tau Wv+\theta$. The formula is similar to the one of DHNN.

Definition of Energy Function

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}v_iv_j - \sum_{i=1}^{n} v_iI_i + \sum_{i=1}^{n}\frac{1}{R_i}\int_0^{v_i}\psi_i^{-1}(v)dv$$

If the amplification is bigger enough, the third item can be ignored. The expression of energy function is the same as one of the DHNN.

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}v_iv_j - \sum_{i=1}^{n} v_iI_i$$

5.1.4 Computing for optimization

应用Hopfield网络优化计算步骤：

1. 对于待定问题，选择一种合适的表示方法，使神经元网络的输出与问题的解对应起来。
2. 构造神经元网络能量函数，使其最小值对应于问题的最佳解。
3. 由能量函数推出神经元网络结构。
4. 运行网络，稳定状态就是在一定条件下问题的解答。

Example 1: TSP—Traveling Salesman Problem

有一旅行商从某一城市出发，访问各城市一次，且仅一次后再回到原出发城市，要求找出一条最短的巡回路线。

N cities : $C=\{c_1,c_2,\ldots,c_N\}$

d_{ij} : distance between c_i and c_j

设N=5， 即A、B、C、D、E分别代表5个城市。

任选一条路径：B→D→E→A→C → B

则其总长：$S=d_{BD}+d_{DE}+d_{EA}+d_{AC}+d_{CB}$

第一步：将此问题映照到一个神经网络。

设每个神经元放大器有很高的放大倍数，神经元输出为0、1二值，如换位矩阵所示。

行表示城市，列表示巡回次序

矩阵的每个元素为一个神经元，

即N^2=25个神经元组成 Hopfield网络。

换位矩阵(Permutation Matrix)

次序 / 城市	1	2	3	4	5
A	0	0	0	1	0
B	1	0	0	0	0
C	0	0	0	0	1
D	0	1	0	0	0
E	0	0	1	0	0

第二步：把问题的目标函数转化为能量函数，并将问题的变量对应于网络的状态。

1. 一个城市只能被访问一次，即换位矩阵每行只有一个“1”。

* 即第x行的所有元素v_{xi}按顺序两两相乘之和应为0。

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} v_{xi}v_{xj}=0$$

* N行的所有元素按顺序两两相乘之和也应为0。

$$\sum_{x=1}^{N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} v_{xi}v_{xj}=0$$

将此式乘上加权系数就为网络能量函数的第一项。

$$\frac{A}{2}\sum_{x=1}^{N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} v_{xi}v_{xj}=0$$

2. 一次只能访问一个城市，即换位矩阵每列只有一个“1”。

与上同理，得能量函数的第二项：

$$\frac{B}{2}\sum_{i=1}^{N}\sum_{x=1}^{N-1}\sum_{y=x+1}^{N} v_{xi}v_{yi}=0$$

3. 总共有N个城市，即换位矩阵元素之和为N。

$$\sum_{x=1}^{N}\sum_{i=1}^{N} v_{xi}-N=0$$

得能量函数得第三项：

$$\frac{C}{2}\left[\sum_{x=1}^{N}\sum_{i=1}^{N} v_{xi}-N\right]^2$$

4. 要求巡回路径最短，即能量函数的最小值对应于TSP的最短路径。

设任意两个城市x、y之间的距离为d_{xy}，访问此二城市有两种途径：

x→y 和 y→x

$$d_{xy}v_{xi}v_{y,i+1} \quad 和 \quad d_{xy}v_{xi}v_{y,i-1}$$

顺序访问x、y两城市的所有途径（长度）为：

$$\sum_{i=1}^{N}\left(d_{xy}v_{xi}v_{y,i+1}+d_{xy}v_{xi}v_{y,i-1}\right)=\sum_{i=1}^{N} d_{xy}v_{xi}\left(v_{y,i+1}+v_{y,i-1}\right)$$

If i+1>n then i+1=1

N个城市两两之间所有可能的访问途径的长度可表示为：

$$\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{N} d_{xy}v_{xi}\left(v_{y,i+1}+v_{y,i-1}\right)$$

能量函数的第四项：

$$\frac{D}{2}\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{N} d_{xy}v_{xi}\left(v_{y,i+1}+v_{y,i-1}\right)$$

最后的网络能量函数为：

$$E=\frac{A}{2}\sum_{x=1}^{N}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} v_{xi}v_{xj}+\frac{B}{2}\sum_{i=1}^{N}\sum_{x=1}^{N-1}\sum_{y=x+1}^{N} v_{xi}v_{yi}$$

$$+\frac{C}{2}\left(\sum_{x=1}^{N}\sum_{i=1}^{N} v_{xi}-N\right)^2+\frac{D}{2}\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{N} d_{xy}v_{xi}(v_{y,i+1}+v_{y,i-1})$$

当E达到极小值时，由网络状态v_{ij}构成的换位矩阵表达了最佳旅行路径。

Hopfield能量函数：

$$E=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}v_iv_j-\sum_{i=1}^{n} v_iI_i$$

比较TSP能量函数与Hopfield标准能量函数，可以得到：

$$w_{xi,yj}=-A\delta_{xy}(1-\delta_{ij})-B\delta_{ij}(1-\delta_{xy})-C-Dd_{xy}(\delta_{j,i+1}+\delta_{j,i-1})$$

$$I_{xi}=CN$$

$$\delta_{ij}=\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases}$$

将上式代入网络运行方程式， $\boxed{\frac{du_i}{dt}=-\frac{u_i}{\tau_i}+\sum_{j=1}^{n} w_{ij}v_j+\theta_i}$ 得：

$$c_{xi}\frac{du_i}{dt}=-\frac{u_{xi}}{R_{xi}}-A\sum_{\substack{j=1\\ j\neq i}}^{N} v_{xj}-B\sum_{\substack{y=1\\ y\neq x}}^{N} v_{yi}-C\left(\sum_{x=1}^{N}\sum_{y=1}^{N} v_{xy}-N\right)-D\sum_{y=1}^{N} d_{xy}\left(v_{y,i+1}+v_{y,i-1}\right)$$

$$v_{xi}=\psi_{xi}(u_{xi})$$

反馈网络用于优化计算和作为联想存储这两个问题是对偶的。

用于优化计算时**W**已知（以目标函数和约束条件建立系统的能量函数确定），目的是找**E**达到最小的稳定状态，即是优化计算问题的解；

作联想存储时则稳定状态是给定的（对应于待存向量），要通过学习找到合适的**W**。