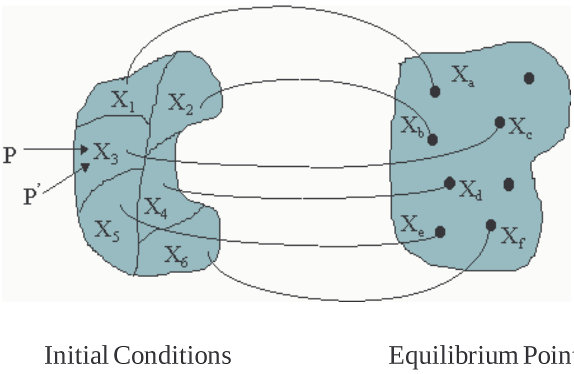
Hopfield Models

General Idea: Artificial Neural Networks ↔ Dynamical Systems



Continuous Hopfield Model

$$C_i \frac{dx_i(t)}{dt} = -\frac{x_i(t)}{R_i} + \sum_{j=1}^{N} w_{ij}\varphi_j(x_j(t)) + I_i$$

a) the synaptic weight matrix is symmetric, $w_{ij} = w_{ji}$, for all i and j.

b) Each neuron has a nonlinear activation of its own, i.e. $y_i = \varphi_i(x_i)$.

Here, $\varphi_i(\bullet)$ is chosen as a sigmoid function;

c) The inverse of the nonlinear activation function exists, so we may write $x = \varphi_i^{-1}(y)$.

Continuous Hopfield Model

Lyapunov Function:

$$E = -\frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{N} w_{ij} y_i y_j + \sum_{i=1}^{N}\frac{1}{R_i}\int_0^{x_i}\varphi_i^{-1}(y_i)dx - \sum_{i=1}^{N} I_i y_i$$

$$\frac{dE}{dt} = -\sum_{i=1}^{N}\left(\sum_{j=1}^{N} w_{ij} y_j - \frac{x_i}{R_i} + I_i\right)\frac{dy_i}{dt}$$

$$= -\sum_{i=1}^{N} C_i \left[\frac{d\varphi_i^{-1}(y_i)}{dt}\right]\frac{dy_i}{dt}$$

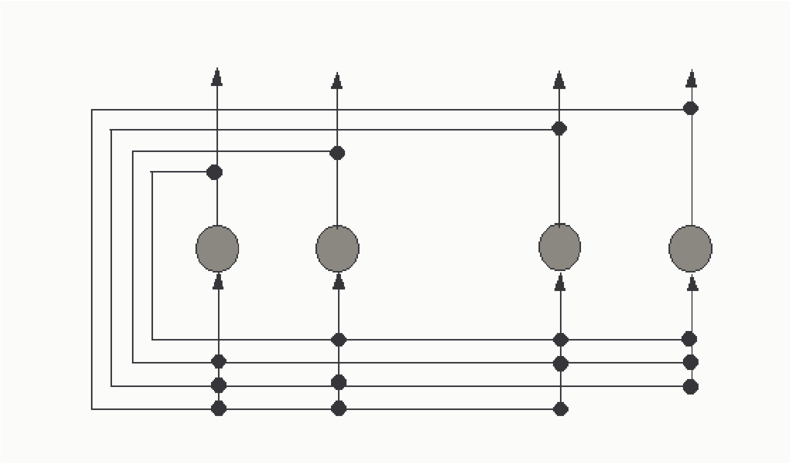
$$= -\sum_{i=1}^{N} C_i \left(\frac{dy_i}{dt}\right)^2 \left[\frac{d\varphi_i^{-1}(y_i)}{dt}\right]$$

≤ 0

Discrete Hopfield Model

- Recurrent network
- Fully connected
- Symmetrically connected ($w_{ij} = w_{ji}$, or $W = W^T$)
- Zero self-feedback ($w_{ii} = 0$)
- One layer
- Binary States:
  - $x_i = 1$ firing at maximum value
  - $x_i = 0$ not firing
- or Bipolar
  - $x_i = 1$ firing at maximum value
  - $x_i = -1$ not firing

Discrete Hopfield Model



Discrete Hopfield Model (Bipole)

Transfer Function for Neuron *i*:

$$x_i = \begin{cases} 1 & \sum_{j \neq i} w_{ij} x_j - \theta_i > 0 \\ -1 & \sum_{j \neq i} w_{ij} x_j - \theta_i < 0 \\ x_i & \sum_{j \neq i} w_{ij} x_j - \theta_i = 0 \end{cases}$$

$\mathbf{x} = (x_1, x_2 \ldots x_N)$: *bipole vector, network state.*
θ_i : *threshold value of* x_i.

$$x_i = \text{sgn}\left(\sum_{j \neq i} w_{ij} x_j - \theta_i \right) \qquad \mathbf{x} = \text{sgn}\left(\mathbf{Wx} - \Theta\right)$$

Discrete Hopfield Model

Energy Function:

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

For simplicity, we consider all threshold $\theta_i = 0$:

$$E = -\frac{1}{2}\sum_{i}\sum_{j\neq i} w_{ij} x_i x_j$$

Discrete Hopfield Model

Learning Prescription (Hebbian Learning):

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{M} \xi_{\mu,i}\xi_{\mu,j}$$

$\{\xi_\mu \mid \mu = 1, 2, ..., M\}$: M memory patterns

Pattern $\xi^s = (\xi^s_1, \xi^s_2, ..., \xi^s_n)$, where ξ^s_i take value 1 or -1

In the matrix form:

$$\mathbf{W} = \frac{1}{N}\sum_{\mu=1}^{M} \boldsymbol{\xi}_\mu \boldsymbol{\xi}_\mu^T - M\mathbf{I}$$

Discrete Hopfield Model

Energy function is lowered by this learning rule:

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ij} x_i x_j = -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ij} \xi_{\mu,i} \xi_{\mu,j}$$

$$\Leftrightarrow -\frac{1}{2}\sum_i \sum_{j \neq i} \xi_{\mu,i}^2 \xi_{\mu,j}^2$$

Discrete Hopfield Model

Pattern Association (asynchronous update):

$$\Delta_k E = E(k+1) - E(k)$$

$$= -\frac{1}{2}\sum_i \sum_{j \neq i} w_{ij} x_i (k+1) x_j + \frac{1}{2}\sum_i \sum_{j \neq i} w_{ij} x_i (k) x_j$$

$$\Leftrightarrow -\Delta x_i(k) \sum_{j \neq i} w_{ij} x_j$$

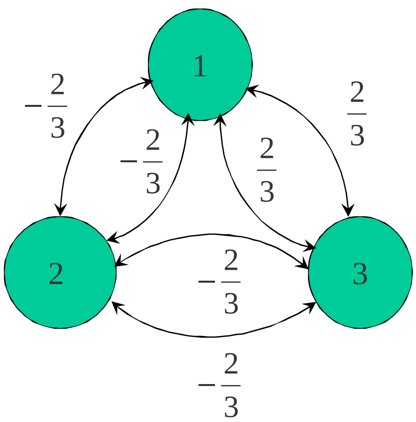
$$\mathbf{\Delta E_k \leq 0}$$

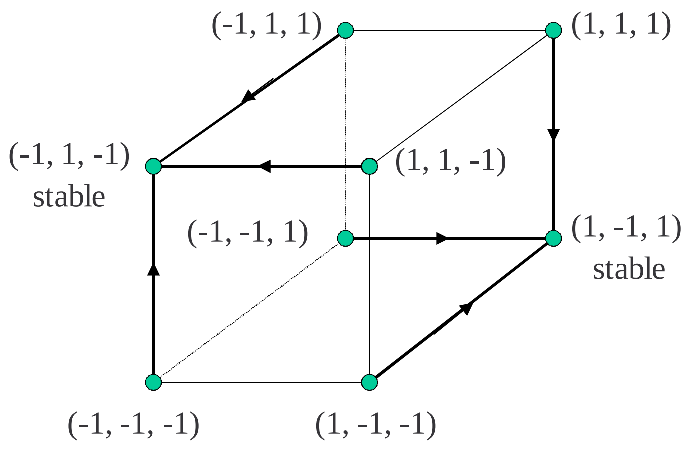
Discrete Hopfield Model

Example:

Consider a network with three neurons, the weight matrix is:

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





The model with three neurons has two fundamental memories $(-1, 1, -1)^T$ and $(1, -1, 1)^T$

State (1, -1, 1)^{T}:

$$\mathbf{Wx} = \frac{1}{3}\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 4 \\ -4 \\ 4 \end{bmatrix}$$

$$\mathrm{sgn}[\mathbf{Wx}] = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = \mathbf{x}$$

A stable state

State (-1, 1, -1)^{T}:

$$\mathbf{Wx} = \frac{1}{3}\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} -4 \\ 4 \\ -4 \end{bmatrix}$$

$$\operatorname{sgn}[\mathbf{Wx}] = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = \mathbf{x}$$

A stable state

State (1, 1, 1)^{T}:

$$\mathbf{Wx} = \frac{1}{3}\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 0 \\ -4 \\ 0 \end{bmatrix}$$

$$\operatorname{sgn}[\mathbf{Wx}] = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \neq \mathbf{x}$$

An unstable state. However, it converges to its nearest stable state (1, -1, 1)^{T}

State (-1, 1, 1)^{T}:

$$\mathbf{Wx} = \frac{1}{3}\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 0 \\ 0 \\ -4 \end{bmatrix}$$

$$\text{sgn}[\mathbf{Wx}] = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

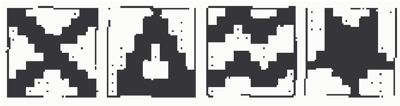
An unstable state. However, it converges to its nearest stable state (-1, 1, -1)^{T}

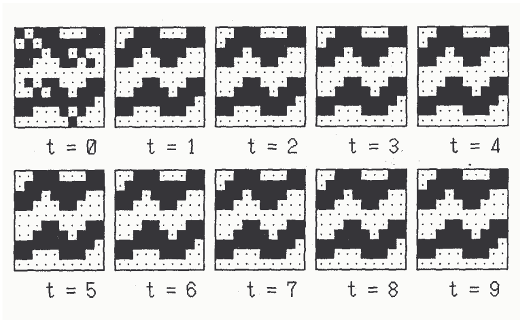
Thus, the synaptic weight matrix can be determined by the two patterns:

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

$$= \frac{1}{3}\begin{bmatrix} 0 & -2 & 2 \\ -2 & 0 & -2 \\ 2 & -2 & 0 \end{bmatrix}$$

Computer Experiments





Significance of Hopfield Model

1) The Hopfield model establishes the bridge between various disciplines.
2) The velocity of pattern recalling in Hopfield models is independent on the quantity of patterns stored in the net.

Limitations of Hopfield Model

1) Memory capacity;

The memory capacity is directly dependent on the number of neurons in the network. A theoretical result is

$$p < \frac{N}{2 \log N}$$

When N is large, it is approximately

$$p = 0.14N$$

2) Spurious memory;
3) Auto-associative memory;
4) Reinitialization
5) Oversimplification

Problema de Caixeira Viajante

- Buscar um caminho mais curto entre *n* cidades visitando cada cidade somente uma vez e voltando a cidade de partida.
- Um problema clássico de otimização combinatório;
- Algoritmos para encontrar uma solução exato são NP-difíceis

Problema de Caixeira Viajante

$$E = \frac{W_1}{2}\left\{\sum_{i=1}^{n}\left(\sum_{j=1}^{n} x_{ij} - 1\right)^2 + \sum_{j=1}^{n}\left(\sum_{i=1}^{n} x_{ij} - 1\right)^2\right\}$$

$$+\frac{W_2}{2}\left\{\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(x_{k\,j+1} + x_{k\,j-1}\right)x_{ij}\, d_{ij}\right\}$$

$$x_{i\,0} = x_{in}$$
$$x_{i\,n+1} = x_{i1}$$

d_{ij}: distancia entre cidade *i* e *j*

Cidade/Posição	1	2	3	4
1	1	0	0	0
2	0	0	1	0
3	0	0	0	1
4	0	1	0	0

x_{ij}: Output do neurônio (i, j)

N cidades, N^2 neurônios

$$y_{ij}(t+1)=ky_{ij}(t)+\alpha\left\{-W_1\left(\sum_{i\neq j}^{N}x_{ij}(t)+\sum_{k\neq i}^{N}x_{kj}(t)\right)-W_2\left(\sum_{k\neq i}^{N}d_{ik}x_{kj+1}(t)+\sum_{k\neq i}^{N}d_{ik}x_{kj-1}(t)\right)+W_3\right\}$$

$$x_{ij}(t)=\frac{1}{1+e^{-y_{ij}(t)/\varepsilon}}$$

$$x_{ij}(t)=\begin{cases}1 & iff \quad x_{ij}(t)>\sum_{k,l}x_{kl}(t)\Big/N^2\\ 0 & otherwise\end{cases}$$