CS623: Introduction to Computing with Neural Nets *(lecture-11)*

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

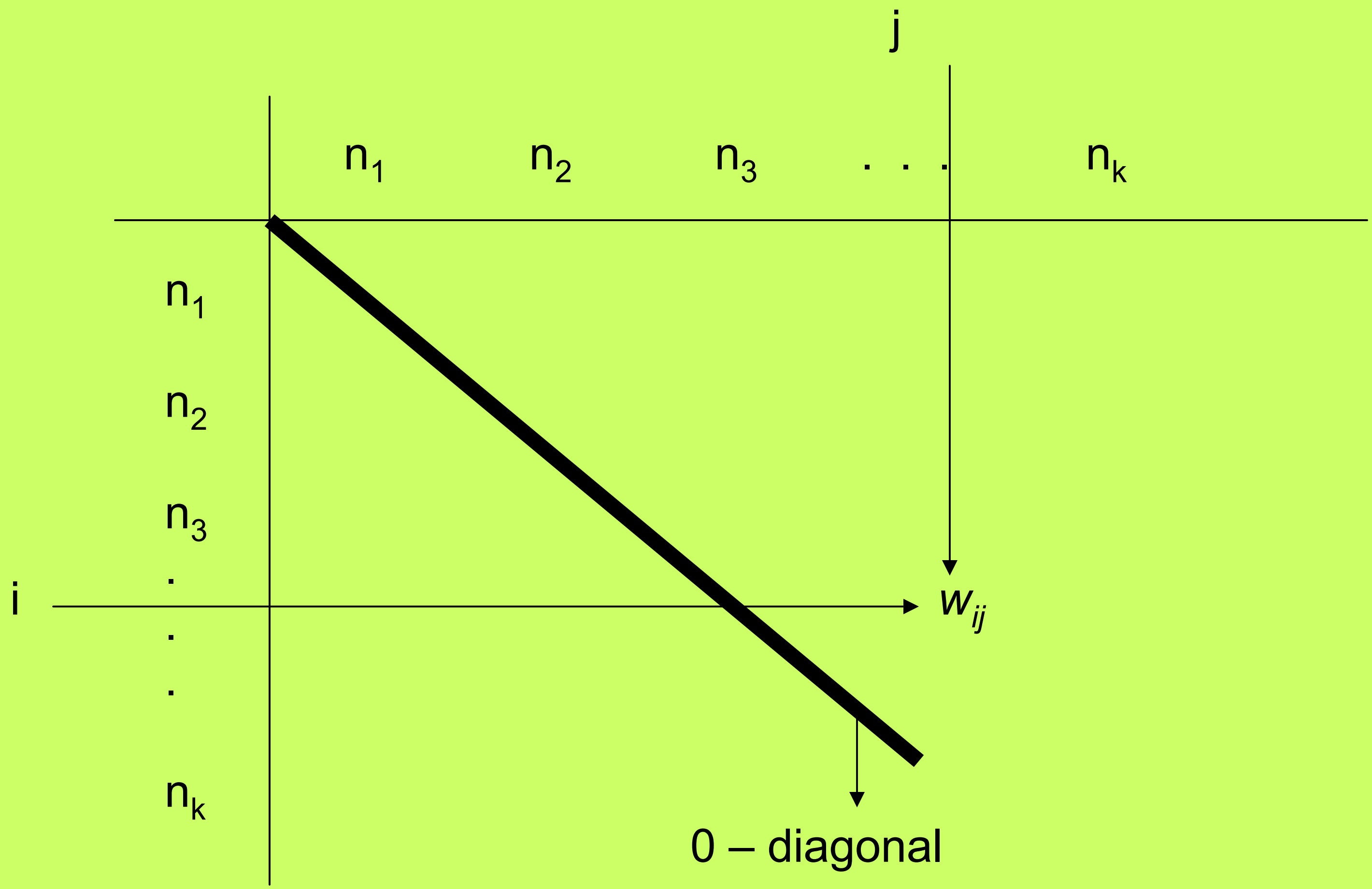
Hopfield net (recap of main points)

- Inspired by associative memory which means memory retrieval is not by address, but by part of the data.
- Consists of

  N neurons fully connected with

  symmetric weight strength $w_{ij} = w_{ji}$
- No self connection. So the weight matrix is 0-diagonal and symmetric.
- Each computing element or neuron is a linear threshold element with threshold = 0.

Connection matrix of the network, 0-diagonal and symmetric



Example

$w_{12} = w_{21} = 5$

$w_{13} = w_{31} = 3$

$w_{23} = w_{32} = 2$

<u>At time *t=0*</u>

$s_1(t) = 1$

$s_2(t) = -1$

$s_3(t) = 1$

Unstable state: Neuron 1 will flip.

A stable pattern is called an attractor for the net.

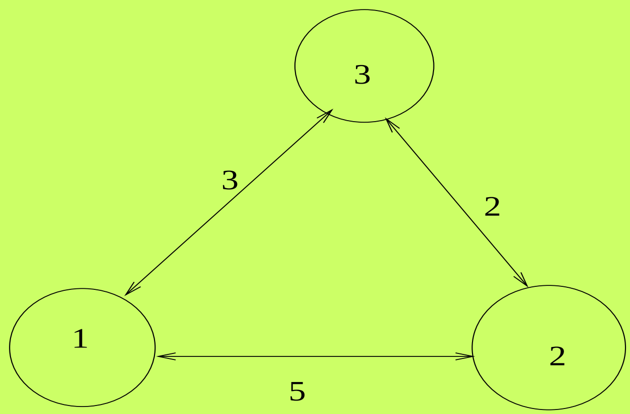


Figure: An example Hopfield Net

Concept of Energy

- Energy at state *s* is given by the equation:

$$
\begin{aligned}
E(s) = -\Big[& w_{12}x_1x_2 + w_{13}x_1x_3 + \ldots + w_{1n}x_1x_n \\
& + w_{23}x_2x_3 + \ldots + w_{2n}x_2x_n + \\
& \vdots \\
& + w_{(n-1)n}x_{(n-1)}x_n \Big]
\end{aligned}
$$

Relation between weight vector W and state vector X

$$W \cdot X^T$$

Weight vector — Transpose of state vector

For example, in figure 1,
At time $t = 0$, state of the neural network is:
$s(0) = \langle 1, -1, 1 \rangle$ and corresponding vectors are as shown.

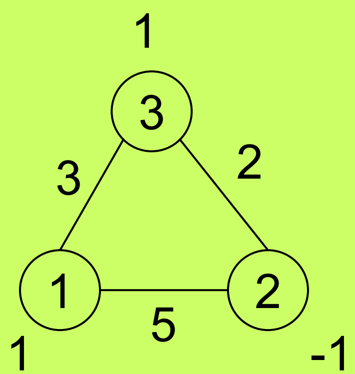


Fig. 1

$$W = \begin{bmatrix} 0 & 5 & 3 \\ 5 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} \qquad X^T = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$$W \cdot X^T = \begin{bmatrix} 0 & 5 & 3 \\ 5 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$W.X^T$ gives the inputs to the neurons at the next time instant

$$W \cdot X^T = \begin{bmatrix} \mathbf{0} & \mathbf{5} & \mathbf{3} \\ \mathbf{5} & \mathbf{0} & \mathbf{2} \\ \mathbf{3} & \mathbf{2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

$$= \begin{bmatrix} -2 \\ 7 \\ 1 \end{bmatrix}$$

$$\operatorname{sgn}(W.X^T) = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

This shows that the n/w will change state

Theorem

- In the asynchronous mode of operation, the energy of the Hopfield net <u>always</u> decreases.
- Proof:

$$
\begin{aligned}
E(t_1) = -\Big[& w_{12} x_1(t_1) x_2(t_1) + w_{13} x_1(t_1) x_3(t_1) + \ldots + w_{1n} x_1(t_1) x_n(t_1) \\
& + w_{23} x_2(t_1) x_3(t_1) + \ldots + w_{2n} x_2(t_1) x_n(t_1) + \\
& \vdots \\
& + w_{(n-1)n} x_{(n-1)}(t_1) x_n(t_1) \Big]
\end{aligned}
$$

Proof

- Let neuron 1 change state by summing and comparing
- We get following equation for energy

$$
\begin{aligned}
E(t_2) = -\Big[& w_{12}x_1(t_2)x_2(t_2) + w_{13}x_1(t_2)x_3(t_2) + \ldots + w_{1n}x_1(t_2)x_n(t_2) \\
& + w_{23}x_2(t_2)x_3(t_2) + \ldots + w_{2n}x_2(t_2)x_n(t_2) + \\
& \vdots \\
& + w_{(n-1)n}x_{(n-1)}(t_2)x_n(t_2) \Big]
\end{aligned}
$$

Proof: *note that only neuron 1 changes state*

$$\Delta E = E(t_2) - E(t_1)$$

$$= -\{[w_{12}x_1(t_2)x_2(t_2) + w_{13}x_1(t_2)x_3(t_2) + \ldots + w_{1n}x_1(t_2)x_n(t_2)]$$

$$-[w_{12}x_1(t_1)x_2(t_1) + w_{13}x_1(t_1)x_3(t_1) + \ldots + w_{1n}x_1(t_1)x_n(t_1)]\}$$

$$= -\sum_{j=2}^{n} w_{1j}\left[x_1(t_2)\cdot x_j(t_2) - x_1(t_1)\cdot x_j(t_1)\right]$$

Since only neuron 1 changes state, $x_j(t_1)=x_j(t_2)$, j=2, 3, 4, …n, and hence

$$= \sum_{j=2}^{n}\left[w_{1j}\cdot x_j(t_1)\right]\left[x_1(t_1) - x_1(t_2)\right]$$

Proof *(continued)*

$$= \sum_{j=2}^{n} \underbrace{\left[w_{1j} \cdot x_j(t_1)\right]}_{(S)} \underbrace{\left[x_1(t_1) - x_1(t_2)\right]}_{(D)}$$

- Observations:
  - When the state changes from -1 to 1, *(S) has to be +ve and (D) is –ve;* so ΔE becomes negative.
  - When the state changes from 1 to -1, *(S) has to be -ve and (D) is +ve;* so ΔE becomes negative.
- Therefore, Energy for any state change always decreases.

The Hopfield net has to “converge” in the asynchronous mode of operation

- As the energy *E* goes on decreasing, it has to hit the bottom, since the weight and the state vector have finite values.
- That is, the Hopfield Net has to converge to an energy minimum.
- Hence the Hopfield Net reaches stability.

Training of Hopfield Net

- Early Training Rule proposed by Hopfield
- Rule inspired by the concept of electron spin
- Hebb's rule of learning
  - If two neurons *i* and *j* have activation x_i and x_j respectively, then the weight w_{ij} between the two neurons is directly proportional to the product $x_i \cdot x_j$ *i.e.*

$$w_{ij} \propto x_i \cdot x_j$$

Hopfield Rule

- Training by Hopfield Rule
  - Train the Hopfield net for a specific memory behavior
  - Store memory elements
  - *How to store patterns?*

Hopfield Rule

- To store a pattern

  $<x_n, x_{n-1}, \ldots., x_3, x_2, x_1>$

  make

$$w_{ij} = \frac{1}{(n-1)} \cdot x_i \cdot x_j$$

- Storing pattern is equivalent to '*Making that pattern the stable state*'

Training of Hopfield Net

- Establish that

  $$<x_n, x_{n-1}, \ldots., x_3, x_2, x_1>$$

  is a stable state of the net

- To show the stability of

  $$<x_n, x_{n-1}, \ldots., x_3, x_2, x_1>$$

  impress at $t=0$

  $$<x^t_n, x^t_{n-1}, \ldots., x^t_3, x^t_2, x^t_1>$$

Training of Hopfield Net

- Consider neuron *i* at *t=1*

$$a_i(t=1) = \operatorname{sgn}(net_i(t=0))$$

$$net_i(t=0) = \sum_{j \neq i, j=1}^{n} w_{ij} \cdot (x_j(t=0))$$

Establishing stability

$$\sum_{j \neq i,\, j=1}^{n} w_{ij} x_j(t=0)$$

$$= \frac{1}{(n-1)} \sum_j (x_i(t=0)) \cdot (x_j(t=0)) \cdot (x_j(t=0))$$

$$= \frac{1}{(n-1)} \sum_j (x_i(t=0)) \cdot [x_j(t=0)]^2$$

$$= \frac{1}{(n-1)} (x_i(t=0)) \cdot \sum_{j=1,\, j \neq i} 1$$

$$= \frac{1}{(n-1)} \cdot (x_i(t=0)) \cdot (n-1)$$

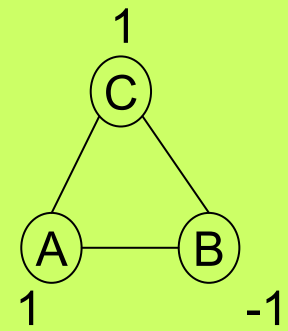
$$= x_i(t=0)$$

Thus ,

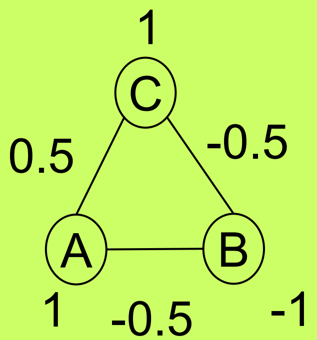
$$x_i(t=1) = \mathrm{sgn}(x_i(t=0))$$

Example

- We want <1, -1, 1> as stored memory
- Calculate all the w_{ij} values
- $w_{AB} = 1/(3-1) * 1 * -1 = -1/2$
- Similarly $w_{BC} = -1/2$ and $w_{CA} = ½$
- Is <1, -1, 1> stable?



Initially



After calculating weight values

Observations

- How much deviation can the net tolerate?
- What if more than one pattern is to be stored?

Storing *k* patterns

- Let the patterns be:

  $P_1 : \langle x_n, x_{n-1}, \ldots., x_3, x_2, x_1 \rangle^1$
  $P_2 : \langle x_n, x_{n-1}, \ldots., x_3, x_2, x_1 \rangle^2$
  .
  .
  .
  $P_k : \langle x_n, x_{n-1}, \ldots., x_3, x_2, x_1 \rangle^k$

- Generalized Hopfield Rule is:

P^{th} pattern

$$wij = \frac{1}{(n-1)} \sum_{p=1}^{k} x_i \cdot x_j \,\Big|_p$$

Storing k patterns

- Study the stability of

  $<x_n, x_{n-1}, \ldots., x_3, x_2, x_1>$

- Impress the vector at t=0 and observer network dynamics
- Looking at neuron i at $t=1$, we have

Examining stability of the qth pattern

$$x_i(1)|_q$$

$$= \mathrm{sgn}(net_i(1)|_q); net_i(1)|_q = \sum_{j=1, j\neq i}^{n} w_{ij} \cdot x_j(0)|_q$$

$$w_{ij} \cdot x_j(0)|_q$$

$$= \frac{1}{(n-1)}[\sum_{p=1}^{k} x_i(0)|_p \cdot x_j(0)|_p] \cdot x_j(0)|_q$$

$$= \frac{1}{(n-1)}[\sum_{p=1, p\neq q}^{k} x_i(0)|_p \cdot x_j(0)|_p] \cdot x_j(0)|_q + \frac{1}{(n-1)} x_i(0)|_q \cdot x_j(0)|_q \cdot x_j(0)|_q$$

$$= \frac{1}{(n-1)}[\sum_{p=1, p\neq q}^{k} x_i(0)|_q \cdot x_j(0)|_p] + \frac{1}{(n-1)}(x_i(0)|_q) \cdot [x_j(0)|_q]^2$$

$$= Q + \frac{1}{(n-1)} \cdot x_i(0)|_q \cdot 1$$

$$= Q + \frac{x_i(0)|_q}{(n-1)}$$

Examining stability of the q^{th} pattern

$$Thus,$$

$$x_i(1) = \text{sgn}[\sum_{j=1, j\neq i}^{n} Q + \sum_{j=1, j\neq i}^{n} \frac{x_i(0)}{(n-1)}]$$

$$= \text{sgn}[\sum_{j=1, j\neq i}^{n} Q + x_i(0)]$$

$$= \text{sgn}[\underbrace{\sum_{j=1, j\neq i}^{n} \sum_{p=1, p\neq q}^{k} (x_i(0)|_q \cdot x_j(0)|_p) + x_i(0)}]$$

Small when $k << n$

Storing *k* patterns

- Condition for patterns to be stable on a Hopfield net with *n* neurons is:

$$k << n$$

- The storage capacity of Hopfield net is very small
- Hence it is not a practical memory element