Module 3: Unsupervised Learning - Discrete Hopfield Networks

EXTC – BE – Neural Network and Fuzzy Logic

Dr. Preeti Hemnani

Head and Professor

Dept. of Electronics and Telecommunication,

SIES Graduate School of Technology

Associative Memory Networks

- It can store a set of patterns as memories.
- When the associative memory is being presented with a key pattern, it responds by producing one of the stored patterns, which closely resembles or relates to the key pattern. Thus the recall is through association of the key pattern, with the help of information memorized.

CAM architecture

Content Addressable Memory

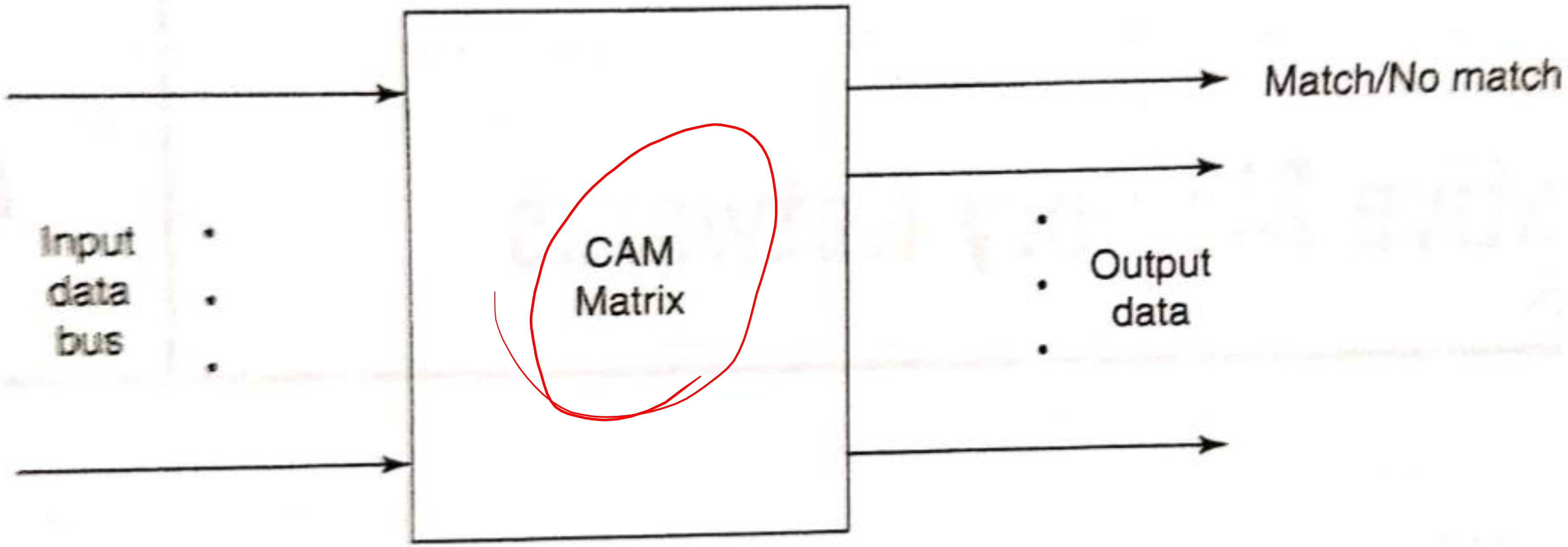
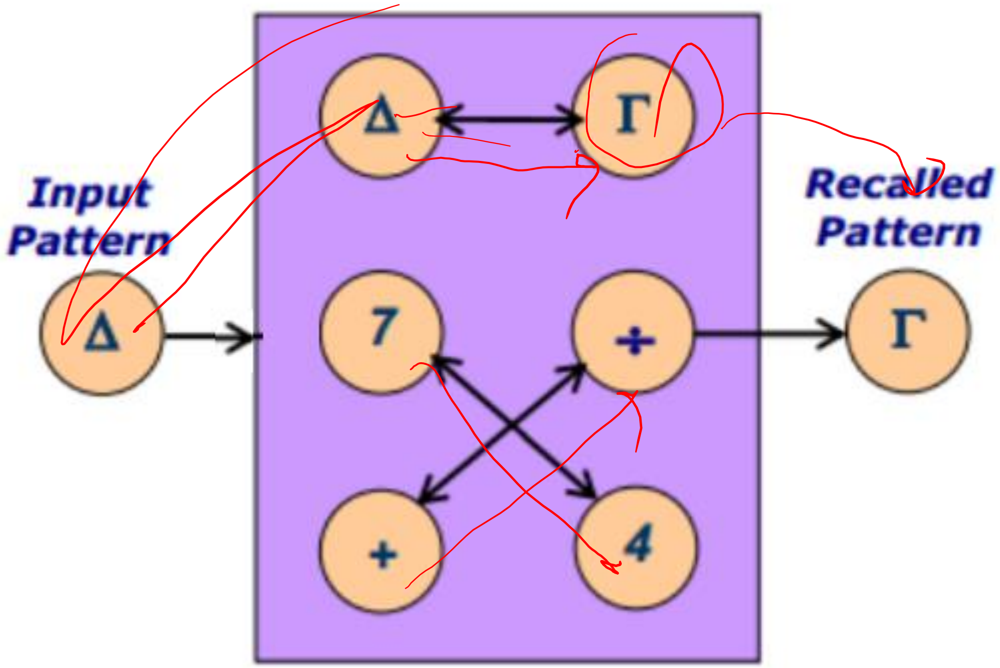


Figure 4-1 CAM architecture.

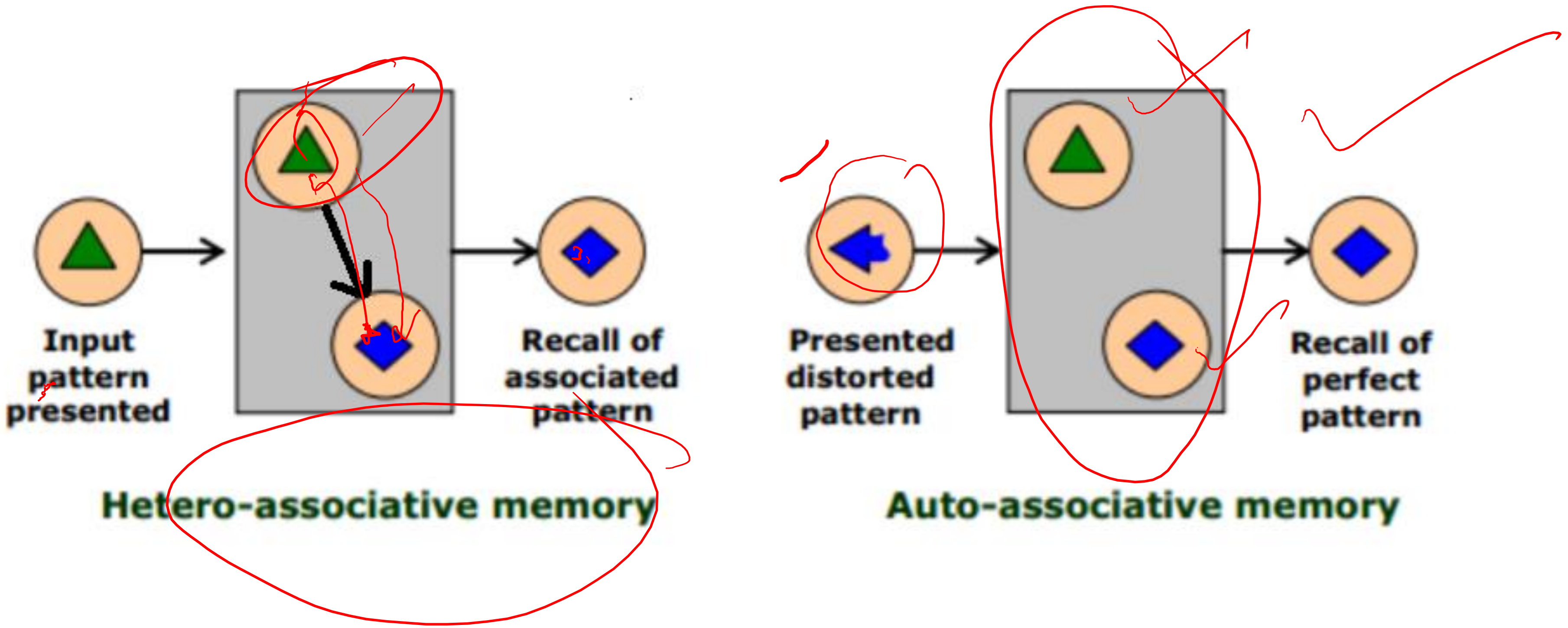
- These types of memories are also called as content-addressable memories (CAM)
- Two types –
  - Auto associative memory – o/p vectors are same as the i/p vectors with which it is associated.
  - Heteroassociative memory – o/p vectors are different from the i/p vectors with which it is associated.

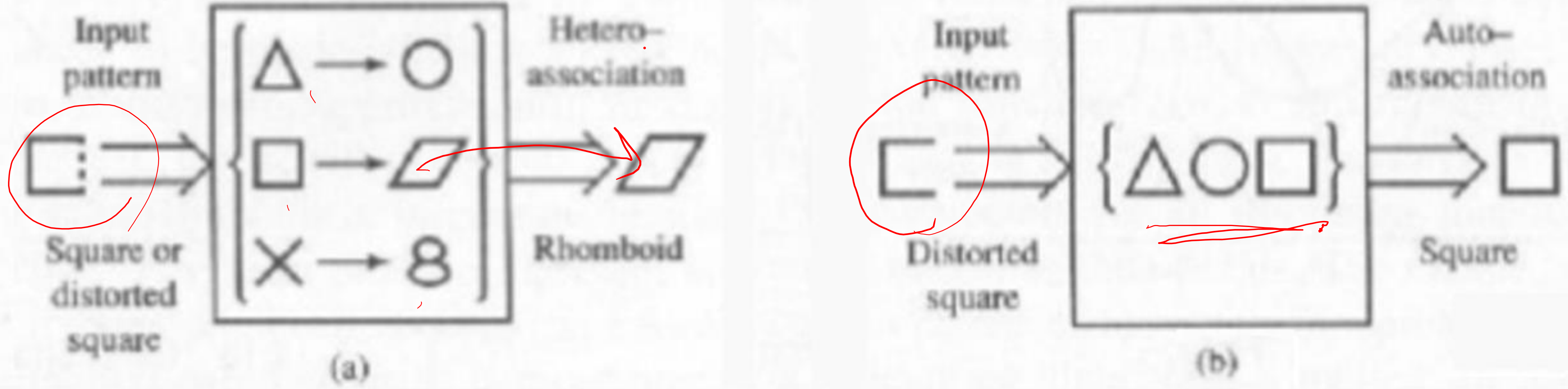
Both these networks are single layer nets

Heteroassociative Networks



Heteroassociative and Auto-associative memory





Association response: (a) heteroassociation and (b) autoassociation.

Discrete Hopfield Networks

1. Single layer N/w

2. Highly Interconnected

3. No self connection

$W_{ii} = 0$

$w_{ij} = w_{ji}$

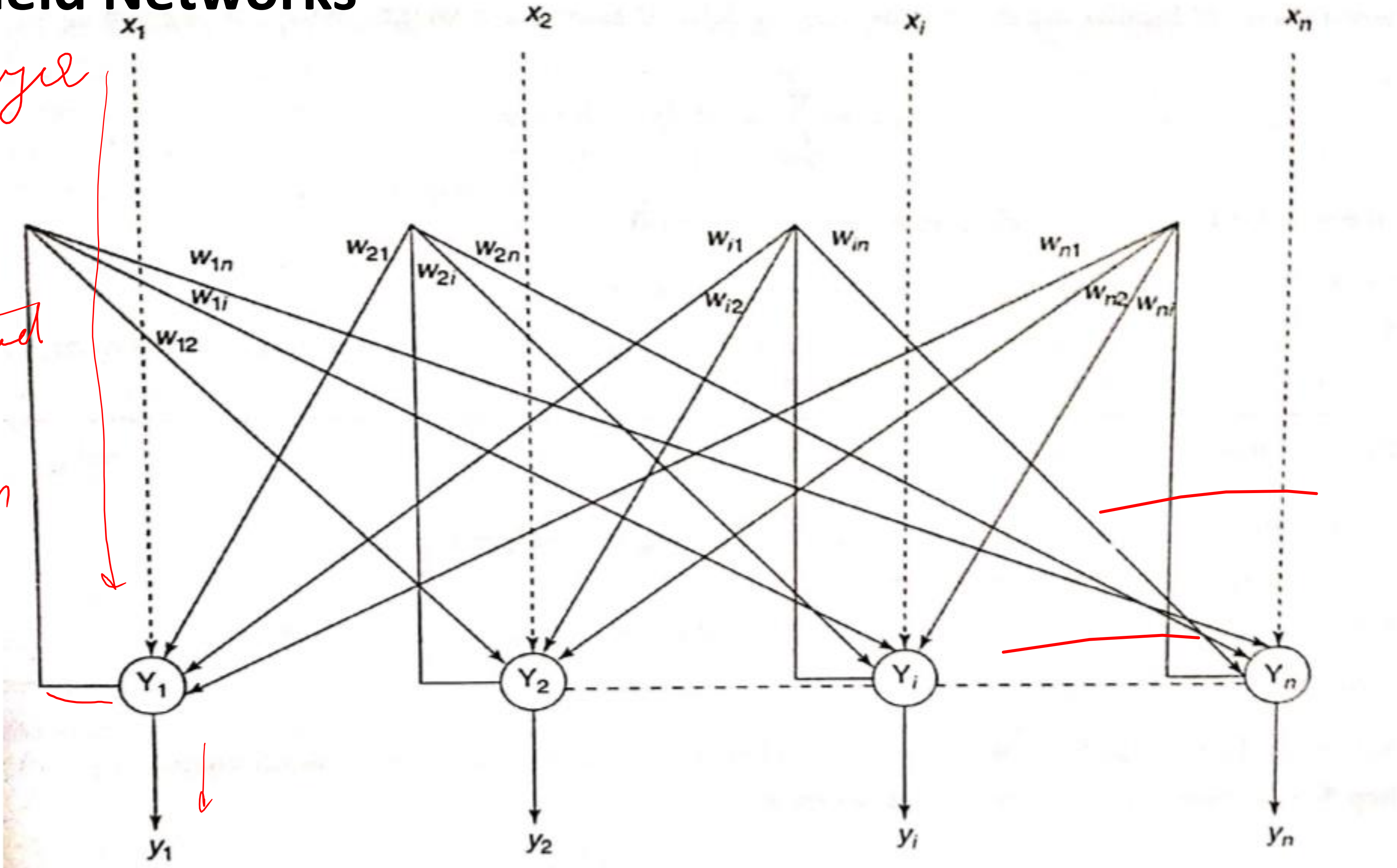


Figure 4-7 Architecture of discrete Hopfield net.

SIES Graduate School of Technology
RISE WITH EDUCATION

Discrete Hopfield Net

1. Hopfield neural network is proposed by John Hopfield in 1982 can be seen
    - as a network with associative memory
    - can be used for different pattern recognition problems.

2. It is a fully connected, single layer auto associative network
    - Means it has only one layer, with each neuron connected to every other neuron

3. All the neurons act as input and output.

Discrete Hopfield Net

- The net is a fully interconnected neural net, in the sense that each unit is connected to every other unit.
- The net has symmetric weights with no self-connections, i.e.,

$$w_{ij} = w_{ji}$$

and

$$w_{ii} = 0.$$

SIES Graduate School of Technology
RISE WITH EDUCATION

- A binary Hopfield net can be used to determine whether an input vector is a "known" or an "unknown" vector.
- The net recognizes a "known" vector by producing a pattern of activation on the units of the net that is the same as the vector stored in the net.
- If the input vector is an "unknown" vector, the activation vectors produced as the net iterates will converge to an activation vector that is not one of the stored patterns.

PROPERTIES OF HOPFIELD

- A recurrent network with all nodes connected to all other nodes.
- Nodes have binary outputs (either 0,1 or -1,1).
- Weights between the nodes are symmetric .
- No connection from a node to itself is allowed.
- Nodes are updated asynchronously (i.e. nodes are selected at random).
- The network has no hidden nodes or layer.

Training Algorithm

1. storing the pattern
weight Matrix

4.6.1.2 *Training Algorithm of Discrete Hopfield Net*

There exist several versions of the discrete Hopfield net. It should be noted that Hopfield's first description used binary input vectors and only later on bipolar input vectors used.

For storing a set of binary patterns $s(p)$, $p = 1$ to P, where $s(p) = (s_1(p), \ldots, s_i(p), \ldots, s_n(p))$, the weight matrix W is given as

$$w_{ij} = \sum_{p=1}^{P} [2s_i(p) - 1][2s_j(p) - 1], \quad \text{for } i \neq j$$

For storing a set of bipolar input patterns, $s(p)$ (as defined above), the weight matrix W is given as

$$w_{ij} = \sum_{p=1}^{P} s_i(p)s_j(p), \quad \text{for } i \neq j$$

and the weights here have no self-connection, i.e., $w_{ij} = 0$.

$$\sum_{p=1}^{P} s_i(p)\, s_j(p)$$

Construct an autoassociative network to store the vectors $x_1 = [1\ 1\ 1\ 1\ 1]$, $x_2 = [1\ -1\ -1\ 1\ -1]$, $x_3 = [-1\ 1\ -1\ -1\ -1]$. Find weight matrix with no self-connection.

$s(1) = [1, 1, 1, 1, 1]$

$s(2) = [1\ -1\ -1\ 1\ -1]$

$s(3) = [-1\ 1\ -1\ -1\ -1]$

For Bipolar i/p. weight Matrix $w_{ij} = \sum_{p=1}^{P} s_i(p)\, s_j(p)$

$w_{ij} = \sum s_i(p)\, s_j(p)$

$W = s^T(1)\, s(1) + s^T(2)\, s(2) + s^T(3)\, s(3)$

$$W = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} [1\ 1\ 1\ 1\ 1] + \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix} [1\ -1\ -1\ 1\ -1] + \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} [-1\ 1\ -1\ -1\ -1]$$

1×5

S^T

2

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 1 & 1 & 1 \\ -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \end{bmatrix}$$

$$W_{ij} = \begin{bmatrix} 3 & -1 & 1 & 3 & 1 \\ -1 & 3 & 1 & -1 & 1 \\ 1 & 1 & 3 & 1 & 3 \\ 3 & -1 & 1 & 3 & 1 \\ 1 & 1 & 3 & 1 & 3 \end{bmatrix}$$

But $w_{ii} = 0$

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

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}_{3\times 2} \begin{bmatrix} 2 & 4 & 6 \\ 3 & 5 & 7 \end{bmatrix}_{2\times 3} = \begin{bmatrix} 5 & 9 & 13 \\ 5 & 9 & 13 \\ 5 & 9 & 13 \end{bmatrix}_{3\times 3}$$

Testing Algorithm

4.6.1.3 Testing Algorithm of Discrete Hopfield Net

In the case of testing, the update rule is formed and the initial weights are those obtained from the trainin algorithm. The testing algorithm for the discrete Hopfield net is as follows:

Step 0: Initialize the weights to store patterns, i.e., weights obtained from training algorithm using Hebb rule.

Step 1: When the activations of the net are not converged, then perform Steps 2–8.

Step 2: Perform Steps 3–7 for each input vector X.

Step 3: Make the initial activations of the net equal to the external input vector X:

$$y_i = x_i \ (i = 1 \text{ to } n)$$

Step 4: Perform Steps 5–7 for each unit Y_i. (Here, the units are updated in random order.)

Step 5: Calculate the net input of the network:

$$y_{ini} = x_i + \sum_j y_j w_{ji}$$

$$y_i = x_i + \Sigma y_j w_{ji}$$

Step 6: Apply the activations over the net input to calculate the output:

$$y_i = \begin{cases} 1 & \text{if } y_{ini} > \theta_i \\ y_i & \text{if } y_{ini} = \theta_i \\ 0 & \text{if } y_{ini} < \theta_i \end{cases}$$

where θ_i is the threshold and is normally taken as zero.

Step 7: Now feed back (transmit) the obtained output y_i to all other units. Thus, the activation vectors are updated.

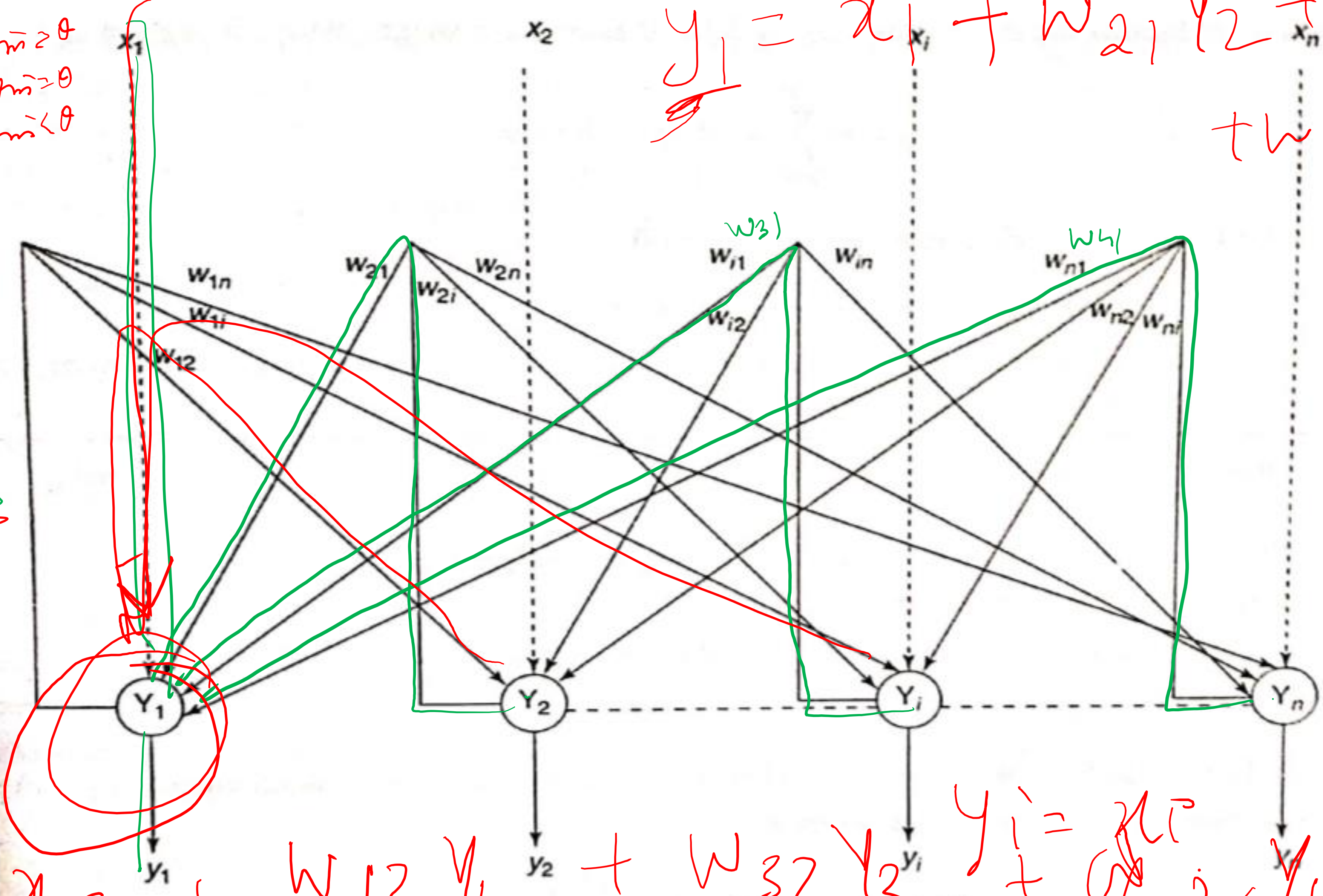
Step 8: Finally, test the network for convergence.

Graduate School of Technology
SIES
RISE WITH EDUCATION

$y_1 = \begin{cases} 1 & y_{in} \ge \theta \\ y_1 & y_{in} = \theta \\ 0 & y_{in} < \theta \end{cases}$

$y_1 = x_1 + w_{21} y_2 + w_{31} y_3 + w_{41} y_4$

$y_{in1} = x_1 + w_{21} y_2 + w_{31} y_3 + w_{41} y_4$



$y_2 = x_2 + w_{12} y_1 + w_{32} y_3$

$y_i = x_i + \sum y_j w_{ji}$

Figure 4-7 Architecture of discrete Hopfield net.

Construct an autoassociative discrete Hopfield network with input vector [1 1 1 − 1]. Test the discrete Hopfield network with missing entries in first and second components of the stored vector.

1 4 2 3

y_1

1×4

$[-1 \; -1 \; 1 \; -1]$

$w_{ij} = 0 \quad i = j$

1×4

0 0 1 0

$W = s^T(1)\, s(1)$

4×1

$W = 4 \times 4$

$$W = \begin{bmatrix} 0 & 1 & 1 & -1 \\ -1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

$x = [0 \ 0 \ 1 \ 0]$

2nd Iteration updated $y = [1 \ 1 \ 1 \ 0]$

1 2 3 4

$y_{in1} = x_1 + w_{21} y_2 + w_{31} y_3 + w_{41} y_4$

$= 0 + [1 \ 1 \ 1 \ 0] \begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \end{bmatrix}$

$= 1 + 1 = 2$

$y_1 = 1 \qquad y = [1 \ 1 \ 1 \ 0] \rightarrow$ Converged.

2 Node

$y_{in2} = x_2 + w_{12} y_1 + w_{32} y_3 + w_{42} y_4$

$= 0 + [1 \ 1 \ 1 \ 0] \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix} = 2 = \quad y_2 = f(y_{in2}) = 1$

$y = [1\ 1\ 1\ 0] \rightarrow$ Converged.

3rd node

$y_{in3} = x_3 + w_{13}\, y_1 + w_{23}\, y_2 + w_{43}\, y_4$

$= 1 + [1\ 1\ 1\ 0] \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \end{bmatrix} = 3$

$y_{in3} = 3 \qquad y_3 = f(y_{in3}) = 1 \qquad y = [1\ 1\ 1\ 0]$

Converged

4th node

$y_{in4} = x_4 + w_{14}\, y_1 + w_{24}\, y_2 + w_{34}\, y_3$

$= 0 + [1\ 1\ 1\ 0] \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix} = -3$

$y_{in4} = -3 \qquad y_4 = f(-3) = 0 \qquad y = [1\ 1\ 1\ 0]$

Converged

Consider a vector ~~(1, 1, 1, 1)~~ to be stored in a net. Test a discrete hopfield net with mistakes in first & fourth component.

$$[1, -1, 1, 1]$$

$$x = [1, 1, 1, -1] \qquad S(1) = [1, 1, 1, -1]$$

$$W_{ij} = S^T(P)\, S(P) = S^T(1)\, S(1)$$

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}_{4\times1} \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}_{1\times4} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}_{4\times4}$$

step 0 $W = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$ as $w_{ii} = 0$

step 1 $x = [1, 1, 1, 0]$ ← binary actual

$x = [0, 0, 1, 1, 0]$ ← Entry

2. $y = [0, 0, 1, 1, 0]$ 1, 4, 3, 2

3. $y_{in1} = x_1 + w_{21}y_2 + w_{31}y_3 + w_{41}y_4$

$= 0 + [0 \; 0 \; 1 \; 0] \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix}$

$$y1_{in} = x1 + [0\ 0\ 1\ 0] \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} = 1$$

4. Apply actⁿ fⁿ

$y1_{in} > 0 \qquad \underline{\underline{y1 = 1}}$

$y = [1\ 0\ 1\ 0] \neq$ No convergence

5. choose Node 4.

$y_{in4} = x_4 + w_{14} y_1 + w_{24} y_2 + w_{34} y_3$

$= 0 + [1 \; 0 \; 1 \; 0] \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \end{bmatrix} = -2$

activation fn

$y_4 = 0$ as $y_{in4} < 0$

6. $y = [1 \; 0 \; 1 \; 0] \rightarrow$ ~~no~~ conv

choose unit 3.

$y_{in3} = x_3 + [1 \; 0 \; 1 \; 0] \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \end{bmatrix}$

$$y_{in3} = 1 + \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

$y_{in3} = 2$ so $\underline{\underline{y_3 = 1}}$ as $y_{in3} > 0$

$y = [1\ 0\ 1\ 0] \rightarrow$ No convergence

6. choosing unit 2.

$$y_{in2} = 0 + \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ -1 \end{bmatrix} = 2$$

$y_{in2} = 2$

applying activation fn $y2 = 1$

$y = [1, 1, 1, 0]$

Converged

Thank You!

(preetih@sies.edu.in)