**Batch: C2 Roll No.: 16010122323**

**Experiment No. 10**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

**Title: : Study experiment on Bi-directional Associative Memory, example**

---

**Aim :** To understand the Bi-Directional Associative Memory

---

**Expected Outcome of Experiment:**

**CO3 :** Understand perceptron's and counter propagation networks

---

**Books/ Journals/ Websites referred:**

---

**Theory**:

**Bi-Directional Associative Memory (BAM)** is a type of recurrent neural network designed for associative memory tasks, meaning it can store and recall patterns based on partial or noisy inputs. The fundamental concept behind BAM is that it can learn and retrieve pairs of patterns in both directions: from input to output and from output back to input.

Characteristics include:

1. **Associative Memory**: Unlike traditional memory systems that require specific addressing, BAM can retrieve stored patterns based on partial inputs, making it highly flexible and robust.
2. **Bidirectionality**: Each layer in BAM can function independently, allowing for inputs to be transformed into outputs and vice versa, which is useful in various applications, including pattern recognition and data compression.

3. **Symmetric Weights**: The weights connecting the two layers are often symmetric. This means that the weight from neuron iii in Layer A to neuron jjj in Layer B is equal to the weight from neuron jjj in Layer B to neuron iii in Layer A. This symmetry simplifies the retrieval process.
4. **Activation Functions**: The neurons in BAM typically employ threshold functions (binary step functions) that output a value (often 0 or 1) based on whether the input exceeds a certain threshold. This enables the network to work effectively with binary data.
5. **Convergence Properties**: BAM networks are designed to converge to stable states, meaning that when a pattern is presented, the network will stabilize on a stored pattern associated with that input.

## Architecture of BAM:

The BAM architecture consists of two main layers:

1. **Input Layer (Layer X)**:
   - Contains neurons representing the input patterns.
   - The number of neurons corresponds to the features of the input data.
2. **Output Layer (Layer Y)**:
   - Contains neurons representing the output patterns.
   - The number of neurons corresponds to the features of the output data.
3. **Weight Matrix**:
   - A weight matrix WWW connects Layer A to Layer B. Each entry WijW_{ij}Wij represents the weight from neuron iii in Layer A to neuron jjj in Layer B.
4. **Activation Mechanism**:
   - Each neuron applies an activation function (typically a binary threshold function) to determine its output based on the weighted inputs it receives.

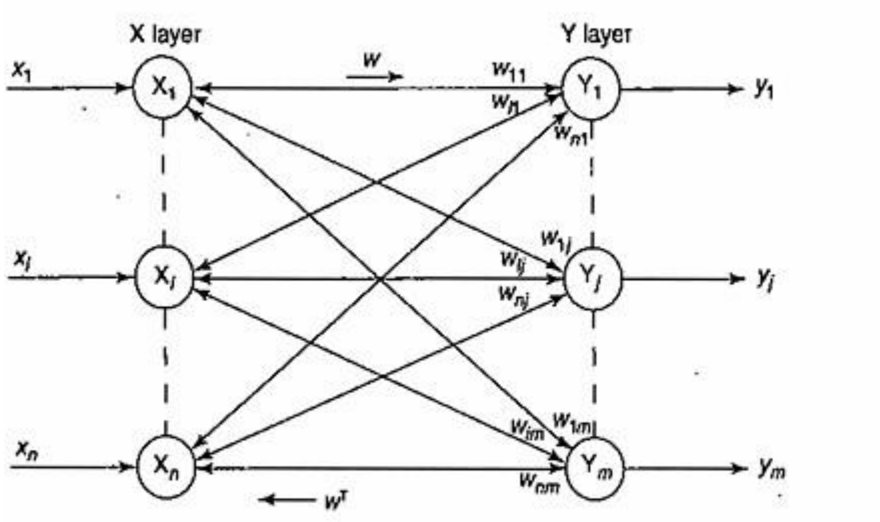


**Figure 4-6** Bidirectional associative memory net.

## Training and Testing of Discrete BAM

### Training Process:

1. **Initialization:**
   - Begin by initializing the weight matrix to small random values or zeros.

2. **Input Patterns:**
   - Prepare a set of paired patterns for training, where each input pattern in Layer A corresponds to an output pattern in Layer B.

3. **Weight Adjustment**:
   - For each input-output pair $(A_k, B_k)$, update the weights using the Hebbian learning rule:

$$W_{ij} = W_{ij} + \eta(a_i b_j)$$

where $a_i$ and $b_j$ are the activations of the respective neurons.

3. **Iteration:**
   - Repeat the training process until convergence or a specified number of iterations is reached.

**Testing Process:**

1. **Input Presentation:**
   - Present a test input pattern to Layer A.
2. **Forward Propagation:**
   - Calculate activations in Layer B using the learned weights.
3. **Output Retrieval:**
   - Retrieve the output pattern from Layer B.
4. **Reverse Retrieval:**
   - Optionally, present the output back to Layer A to verify the bidirectional capability.

**Training and Testing of Continuous BAM**

**Training Process:**

1. **Initialization:**
   - Similar to discrete BAM, initialize the weight matrix.
2. **Continuous Patterns:**
   - Prepare continuous input-output pairs for training.

3. **Weight Adjustment**:

   - Use a continuous adaptation of the Hebbian rule:

$$W_{ij} = W_{ij} + \eta(a_i b_j)$$

where $a_i$ and $b_j$ can take real values.

**4.Iteration**:

- Continue adjusting weights until convergence.

**Testing Process**:

1. **Input Presentation**:
   - Present a continuous input pattern to Layer A.
2. **Activation Calculation**:
   - Compute the activations in Layer B.
3. **Output Retrieval**:
   - Retrieve the output from Layer B.
4. **Robustness Testing**:
   - Test the network's robustness by introducing noise to the input and assessing its ability to reconstruct the original output.

**Numerical to solve (handwritten solution) :**

**Question 1:**

Construct and test a BAM network to associate letters T and O with simple bipolar input–output vectors. The target output for T is $(1, -1)$ and for O is $(1, 1)$. The display matrix size is $4 \times 3$. The input patterns are

| | | | | | | |
|---|---|---|---|---|---|---|
| * | * | * | | * | * | * |
| • | * | • | | * | • | * |
| • | * | • | | * | • | * |
| • | * | • | | * | * | * |
| | "T" | | | | "O" | |

DOMS Page No.
Date / /

## Exp-10 Roll no: 16010122318

Q.1) ~~Inputs~~

```
* * *      * * *
• * •      * • *
• * •      * • *
• * •      * * *
 "T"        "O"
```

Inputs ⇒ $* \rightarrow 1$, $\bullet \rightarrow -1$

| I/P Pattern | I/P | Target | Weight |
|---|---|---|---|
| T | [1 1 1 -1 1 -1 -1 1 -1 -1 1 -1] | [1, -1] | $W_1$ |
| O | [1 1 1 1 -1 1 1 -1 1 1 1 1] | [1, 1] | $W_2$ |

i) Vectors as i/p : The wt. matrix is obtained by,

$$W = \sum S^T(P) + (P)$$

$W_1 = [\ ]\,[1\ -1] = [\ ]$

$W_2 = [\ ]\,[1\ 1] = [\ ]$

$$W = W_1 + W_2 = \begin{bmatrix} 2 & 0 \\ 2 & 0 \\ 2 & 0 \\ 0 & 2 \\ 0 & -2 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 0 & 2 \\ 0 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Testing the n/w w/ test vector 'T' & 'O' for test pattern 'T'
computing net input we get:

$$y_{in} = [1\ 1\ 1\ -1\ 1\ -1\ -1\ 1\ -1\ -1\ 1\ -1] \begin{bmatrix} 2 & 0 \\ 2 & 0 \\ 2 & 0 \\ 0 & 2 \\ 0 & -2 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 0 & 2 \\ 0 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix} = [8\ \ -16]$$

Applying activa$^n$ f$^n$ $f(x) = \begin{cases} 1, & x > 0 \\ -1, & x \leq 0 \end{cases}$

we get, $y = [1\ -1]$

Hence correct o/p is obtained.

For testing 'O' patteren, computing net I/P we get,

$$y_{in} = [1\,1\,1\,1\,{-1}\,1\,1\,1\,{-1}\,1\,1\,1\,{-1}\,1\,1] \cdot \begin{bmatrix} 2 & 0 \\ 2 & 0 \\ 2 & 0 \\ 0 & 2 \\ 0 & -2 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix} = [8 \quad 16]$$

$\Downarrow$

Applying activa$^n$ func$^n$

$$f(x) = \begin{cases} 1 & , \lambda > 0 \\ -1 & , \lambda \leq 0. \end{cases}$$

we get, $y = [1 \quad 1]$.

Hence correct output is obtained.

**Question 2:**

Find the weight matrix in bipolar form for the BAM using outer products rule for the following binary input–output vector pairs.

$$s(1) = (1 \quad 0 \quad 0 \quad 0), \quad t(1) = (0 \quad 1)$$
$$s(2) = (0 \quad 1 \quad 1 \quad 0), \quad t(2) = (1 \quad 0)$$

Using the unit step function as the output unit's activation function, test the response of the network on each of the input patterns. Also test the response of the network on various combinations of input pattern with "mistakes" or "missing" data.

# Ans:

Q.2) The wt. matrix for storing the two I/P vector in bipolar form is $\Rightarrow W = \sum_{p=1}^{n} S^T(p) + (p)$

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

Converting to Bipolar,

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

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

Unit step $f^n$ for binary w/ threshold $0$ is used.

For y layer $\Rightarrow y_j = \begin{cases} 1, & y_{inj} > 0 \\ y_j, & y_{inj} = 0 \\ 0, & y_{inj} < 0 \end{cases}$

For X layer $\Rightarrow x_i = \begin{cases} 1, & x_{ini} > 0 \\ x_i, & x_{ini} = 0 \\ 0, & x_{ini} < 0 \end{cases}$

$S(1) = [1\ 0\ 0\ 0]$

computing net i/p, we have,

$t_{inj} = [1\ 0\ 0\ 0] \begin{bmatrix} -2 & 2 \\ 2 & -2 \\ 2 & -2 \\ 0 & 0 \end{bmatrix} = [-2\ 2]$.

Applying activaⁿ funcⁿ we get $t_j = [0\ 1]$

which is correct.

$S(2) = [0\ 1\ 1\ 0]$.

computing net i/p we have,

$t_{inj} = [0\ 1\ 1\ 0] \begin{bmatrix} -2 & 2 \\ 2 & -2 \\ 2 & -2 \\ 0 & 0 \end{bmatrix} = [4\ -4]$.

Applying activaⁿ fⁿ we get $t_j = [1\ 0]$.

which is correct.

**Date: 22/10/2024** **Signature of faculty in-charge**