Bidirectional Associative Memory

Sayeda Suaiba Anwar
Lecturer, Department of CSE
East Delta University

Associative Memory

- An associative memory is any memory system that stores information by associating each data item with one or more other stored data items.
  - Usually store information in a distributed form.
  - Content addressable memories.
- Data are stored as patterns of activity in an associative memory.
  - Insensitive to minor differences in details.
  - Robust and can usually handle incomplete data inputs.
- There are several types of neural networks that constitute associative memories.
  - The most common types are the crossbar associative memory and the adaptive filter.

Associative Memory

- Heteroassociative and autoassociative memories are not the same.
  - In a heteroassociative memory network, the input X and the output Y are the different patterns; that is, the input and output are not the same.
  - In case of autoassociative memory networks, the input and the output patterns are the same.

Crossbar Structure

- They typically have one or two layers of artificial neurons, and each layer is fully interconnected.
- Matrix Representation is the most commonly used crossbar representation.
  - Because the weights are stored as the elements of a matrix.
  - They are also mathematically tractable, allowing simple explanations of characteristics.

Crossbar Structure

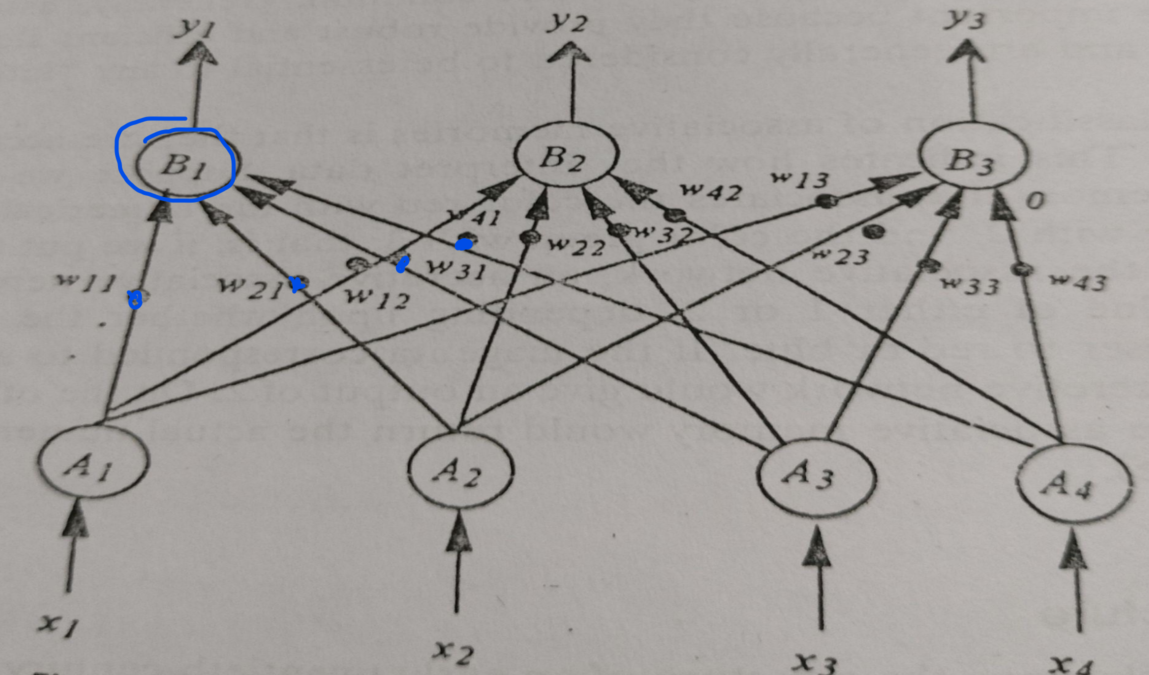

Figure 9.4 Matrix representation of a neural network.

Crossbar Structure

- If we consider the fully connected network shown, the input is a column vector **X** with components x_1, x_2, x_3 and x_4, and the output is a column vector **Y** with components y_1, y_2 and y_3, then the equation can be written as

$$\mathbf{Y = W.X} \text{.......................}\mathbf{(9.3.1)}$$

Where W is the weight matrix. If we expand the terms in equation (9.3.1) it becomes

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41} \\ w_{12} & w_{22} & w_{32} & w_{42} \\ w_{13} & w_{23} & w_{33} & w_{43} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad (9.3\text{-}2)$$

Bidirectional Associative Memory

- Mathematically, a bidirectional associative memory (BAM) developed by Kosko (1988) is also a matrix; technically it is a crossbar network with symmetric weights.
- Suppose, we construct a BAM to store three pattern pairs: $[X_1,Y_1]$, $[X_2,Y_2]$ and $[X_3,Y_3]$. Since a BAM is bidirectional we can enter any X and retrieve the corresponding Y, or we can enter any Y and retrieve the corresponding X.
- Process in a BAM is fundamentally different than the operation of other types of neural networks.
  - The weight matrix is not trained, it is constructed using the input-output pairs.
  - The process involves constructing a matrix for each input-output pair and then combining them into a master matrix.

Bidirectional Associative Memory

- X_i and Y_i are treated as column vectors, and then the matrix is produced by taking the product of the X_i vector and transpose of the Y_i vector, Y_i^T. Let us consider the three following pairs:

Practice:

the Y_i vector, Y_i^T. Let us consider the

$$X_1: \ (+1 -1 -1 -1 -1 +1) \Leftrightarrow (-1 +1 -1) \ :Y_1 \qquad (9.3\text{-}3)$$

$$X_2: \ (-1 +1 -1 -1 +1 -1) \Leftrightarrow (+1 -1 -1) \ :Y_2 \qquad (9.3\text{-}4)$$

$$X_3: \ (-1 -1 +1 -1 -1 +1) \Leftrightarrow (-1 -1 +1) \ :Y_3 \qquad (9.3\text{-}5)$$

and Y_i has 3

Bidirectional Associative Memory

Bipolar = +1, -1

Weight Matrix Representation Since $\mathbf{X}_1$ has 6 elements and $\mathbf{Y}_1$ has 3 elements, the matrix for each set of inputs result in a 6×3 matrix. It is important to note that each of the patterns is made up of $+1$ and -1 values, which means that the components are bipolar. If the patterns values are binary (i.e., made up of 1 and 0 values), they should be converted to bipolar form by substituting -1 for each 0 before they are used in a BAM. The correlation matrices $\mathbf{M}_i$ for equations (9.3-3), (9.3-4), and (9.3-5) are obtained by cross product of $\mathbf{X}_i$ and $\mathbf{Y}_i$—that is,

$$\mathbf{M}_i = \mathbf{X}_i \times \mathbf{Y}_i^T \tag{9.3-6}$$

Bidirectional Associative Memory

The three correlation matrices are

$$M_1 = \mathbf{X}_1 \times \mathbf{Y}_1^T = \begin{bmatrix} +1 \\ -1 \\ -1 \\ -1 \\ -1 \\ +1 \end{bmatrix} \times [-1 \quad +1 \quad -1] = \begin{bmatrix} -1 & +1 & -1 \\ +1 & -1 & +1 \\ +1 & -1 & +1 \\ +1 & -1 & +1 \\ +1 & -1 & +1 \\ -1 & +1 & -1 \end{bmatrix} \tag{9.3-7}$$

Bidirectional Associative Memory

Same Page

The three correlation matrices are

$$M_1 = \mathbf{X}_1 \times \mathbf{Y}_1^T = \begin{bmatrix} +1 \\ -1 \\ -1 \\ -1 \\ -1 \\ +1 \end{bmatrix} \times [-1 \quad +1 \quad -1] = \begin{bmatrix} -1 & +1 & -1 \\ +1 & -1 & +1 \\ +1 & -1 & +1 \\ +1 & -1 & +1 \\ +1 & -1 & +1 \\ -1 & +1 & -1 \end{bmatrix} \tag{9.3-7}$$

Bidirectional Associative Memory

$$M_2 = X_2 \times Y_2^T = \begin{bmatrix} -1 \\ +1 \\ -1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \times [+1 \quad -1 \quad -1] = \begin{bmatrix} -1 & +1 & +1 \\ +1 & -1 & -1 \\ -1 & +1 & +1 \\ -1 & +1 & +1 \\ +1 & -1 & -1 \\ -1 & +1 & +1 \end{bmatrix} \tag{9.3-8}$$

$$M_3 = X_3 \times Y_3^T = \begin{bmatrix} -1 \\ -1 \\ +1 \\ -1 \\ -1 \\ +1 \end{bmatrix} \times [-1 \quad -1 \quad +1] = \begin{bmatrix} +1 & +1 & -1 \\ +1 & +1 & -1 \\ -1 & -1 & +1 \\ +1 & +1 & -1 \\ +1 & +1 & -1 \\ -1 & -1 & +1 \end{bmatrix} \tag{9.3-9}$$

Bidirectional Associative Memory

Note that each value in the above matrices is a product of two quantities, one component of **X** and one component of **Y**. This product $x_i \cdot y_j$ is a classical indication that Hebbian learning is involved.

In order to obtain an associative weight memory (called the master weight matrix) capable of storing the three pairs in equations (9.3-3), (9.3-4), and (9.3-5), we simply add the three correlation matrix equations (9.3-7), (9.3-8), and (9.3-9). The result is

$$\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2 + \mathbf{M}_3 \tag{9.3-10}$$

$$\mathbf{M} = \begin{bmatrix} -1 & +3 & -1 \\ +3 & -1 & -1 \\ -1 & -1 & +3 \\ +1 & +1 & +1 \\ +3 & -1 & -1 \\ -3 & +1 & +1 \end{bmatrix} \tag{9.3-11}$$

Bidirectional Associative Memory

Matrices can be added only if they are the same size. Hence, this means that all of the X_i vector patterns must have the same number of components, and all of the Y_i vector patterns must have the same number of components. However, the number of components in the $\mathbf{X}_i$ pattern can be different from the number of components in the $\mathbf{Y}_i$ patterns (as is the case in this example). In order to put in any $\mathbf{X}_i$ and get back any $\mathbf{Y}_i$ (or put in any $\mathbf{Y}_i$ and get back any $\mathbf{X}_i$), we have to take the product of the input vector and the matrix. This is equivalent to taking the dot product of the vectors and the master matrix. The result is

$$\mathbf{X}_i = \mathbf{M} \cdot \mathbf{Y}_i \qquad (9.3\text{-}12)$$

Bidirectional Associative Memory

and

$$\mathbf{Y}_i = \mathbf{M}^T \cdot \mathbf{X}_i \tag{9.3-13}$$

where the M's are 6×3 weight matrices, $\mathbf{X}_i$ is a 6×1 column vector, and $\mathbf{Y}_i$ is a 3×1 column vector. Note that we must use the transpose of the master matrix to get $\mathbf{Y}_i$; that is,

$$\mathbf{M}^T = \begin{bmatrix} -1 & +3 & -1 & +1 & +3 & -3 \\ +3 & -1 & -1 & +1 & -1 & +1 \\ -1 & -1 & +3 & +1 & -1 & +1 \end{bmatrix}. \tag{9.3-14}$$

Operation of a BAM

The sequence of events are as follows:

- An X input pattern is presented to the BAM.
- The neurons in field X generate an activity pattern that is passed to field Y through the weight matrix M.
- Field Y accepts input from field X and generates a response back to field X through the transpose weight matrix M^T.
- Field X accepts the return response from Y, and then it generates a response back to field Y through the weight matrix M.
- The activity bounces back and forth until a “resonance” is achieved, which means that no further changes in the pattern occur. At this point, the output Y is one of the Y values stored in the master matrix, and it is the correct response for the distorted X input.

Adding and Deleting Pattern to the Master Matrix

- We can add another pattern pair $[X_4,Y_4]$ to our matrix by adding its matrix M_4 to get to the memory matrix M:

$$\textbf{New M} = M_1+M_2+M_3+M_4$$

- Alternately, we can forget or erase a pattern pair by subtracting the matrix for that pattern pair from the memory matrix.

  For instance,

$$\textbf{New M} = M - M_2$$

- **Disadvantages of Crossbar [From Book]**
- **Example 9.2**