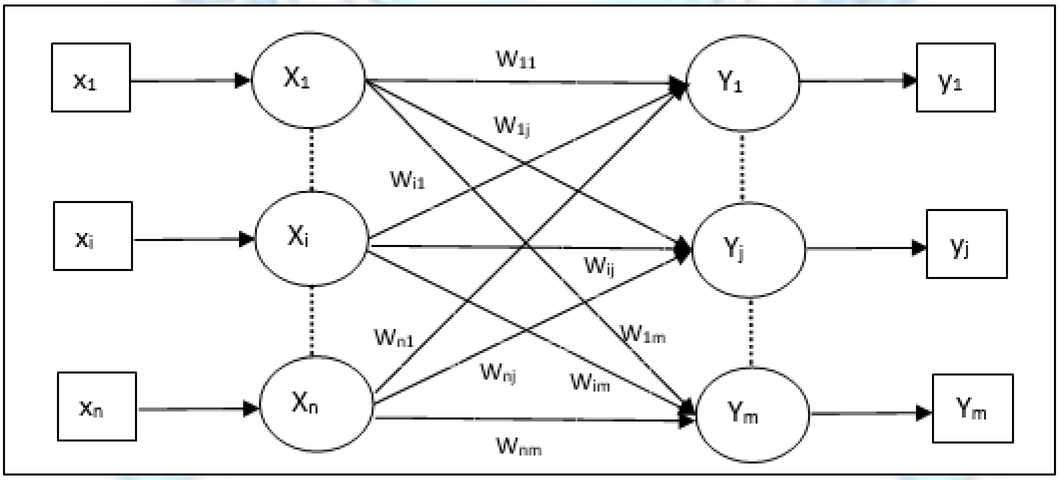
HETRO ASSOCIATIVE MEMORY

In a hetero-associate memory, the training input and the target output vectors are different. The weights are determined in a way that the network can store a set of pattern associations.

The association here is a pair of training input target output vector pairs (s(p), t(p)), with p = 1,2,…p. Each vector s(p) has n components and each vector t(p) has m components. The determination of weights is done either by using Hebb rule or delta rule. The net finds an appropriate output vector, which corresponds to an input vector x, that may be either one of the stored patterns or a new pattern.



HETERO ASSOCIATIVE MEMORY ALGORITHM

Training Algorithm

Step 1 − Initialize all the weights to zero as wij = 0 i= 1 to n, j= 1 to m

Step 2 − Perform steps 3-4 for each input vector.

Step 3 − Activate each input unit as follows −xi=si(i=1ton)

Step 4 − Activate each output unit as follows −$y_j=s_j (j=1 to m)$

Step 5 − Adjust the weights as follows −$w_{ij}(new)=w_{ij}(old)+x_i y_j$

The weight can also be determine from the Hebb Rule or Outer Products Rule learning $w=\sum_{p=1}^{n} S^T(p).S(p)$

Testing Algorithm

Step 1 − Set the weights obtained during training for Hebb's rule.

Step 2 − Perform steps 3-5 for each input vector.

Step 3 − Set the activation of the input units equal to that of the input vector.

Step 4 − Calculate the net input to each output unit j = 1 to m; $y_{inj}=\sum_{i=1}^{n} x_i w_{ij}$ Step 5 − Apply the following activation function to calculate the output $y_j=f(y_{inj})=\{+1 \text{ if } y_{inj}>0,\ 0 \text{ if } y_{inj}=0,\ -1 \text{ if } y_{inj}<0$

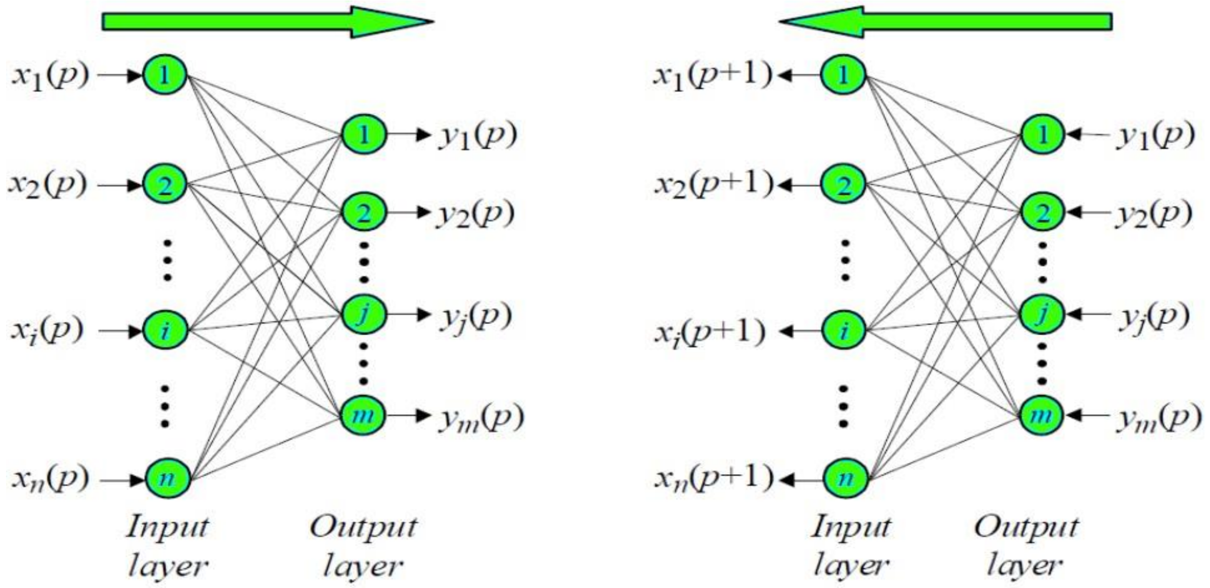
BIDIRECTIONAL ASSOCIATIVE MEMORY (BAM)

Bidirectional Associative Memory (BAM) is a supervised learning model in Artificial Neural Network. This is hetero-associative memory, for an input pattern, it returns another pattern which is potentially of a different size. This phenomenon is very similar to the human brain. Human memory is necessarily associative. It uses a chain of mental associations to recover a lost memory like associations of faces with names, in exam questions with answers, etc. In such memory associations for one type of object with another, a Recurrent Neural Network (RNN) is needed to receive a pattern of one set of neurons as an input and generate a related, but different, output pattern of another set of neurons.

Why BAM is required?

The main objective to introduce such a network model is to store hetero-associative pattern pairs. This is used to retrieve a pattern given a noisy or incomplete pattern.

BAM Architecture: When BAM accepts an input of n-dimensional vector X from set A then the model recalls m-dimensional vector Y from set B. Similarly, when Y is treated as input, the BAM recalls X.



(*a*) Forward direction. (*b*) Backward direction.

Algorithm:

1. **Storage (Learning):** In this learning step of BAM, weight matrix is calculated between M pairs of patterns (fundamental memories) are stored in the synaptic weights of the network following the equation

2. **Testing:** We have to check that the BAM recalls perfectly for corresponding and recalls for corresponding . Using,

$$Y_m = \text{sign}\left(W^T X_m\right), \quad m = 1.2, \ldots, M$$
$$X_m = \text{sign}\left(W Y_m\right), \quad m = 1.2, \ldots, M$$

All pairs should be recalled accordingly.

3. **Retrieval:** For an unknown vector X (a corrupted or incomplete version of a pattern from set A or B) to the BAM and retrieve a previously stored association

4. Initialize the BAM:

$$X(0) = X, \quad p = 0$$

Calculate the BAM output at iteration:

$$Y(p) = \text{sign}\left[W^T X(p)\right]$$

Update the input vector :

$$X(p+1) = \text{sign}[W Y(p)]$$

- Repeat the iteration until convergence, when input and output remain unchanged.

Limitations of BAM:

- Storage capacity of the BAM: In the BAM, stored number of associations should not be exceeded the number of neurons in the smaller layer.
- Incorrect convergence: Always the closest association may not be produced by BAM.

TEMPORAL ASSOCIATIVE MEMORY

Temporal associative memory networks are neural networks that can process temporal information to perform tasks like classification and prediction. They can be used to model complex systems, such as the brain, and to analyze data from a variety of sources. Types of temporal associative memory networks

- **Spatio-temporal associative memory (STAM):** A system that uses spatio-temporal learning (STL) to train a model on data. STAM models can be trained on new variables and used to capture spatio-temporal patterns.
- **Vector-quantized temporal associative memory (VQTAM):** An unsupervised neural modeling technique that uses Kohonen's self-organizing map (SOM) to approximate nonlinear dynamical mappings.

- **Finite state network (FSN):** A new class of temporal associative neural network.

Applications of temporal associative memory networks Modeling brain signals, Analyzing audio-visual data, Analyzing seismic sensory data, Analyzing financial and economic data, and Approximating nonlinear dynamical mappings.