

Department of Computer Science  
COSC 4P80 - Artificial Neural Network

## **Bidirectional Associative Memory - BAM**

**Prepared by:** Hridoy Rahman

**Instructor:** Dave Bockus

**Date:** October 8, 2024

# Abstract

Bidirectional Associative Memory is a type of recurrent neural network that uses associative memory. Associative memory stores associations between patterns and recalls a pattern when presented with another pattern. The assignment is related to BAM Networks, and for part A of the Assignment, a BAM Network is created where the network can take an input pattern and return an associated pattern. On the contrary, if an output pattern is given, it can also recall its input pattern association. The second part of the assignment is based on cross-talk of the Input vector set A. If the input vectors are not orthogonal, the response will include a portion of their target values. This is called the cross talk. For part(B), the cross-talk calculation of vector input set A is provided. In part(C), the input vector set A is extended with the provided example vector, and three more are added randomly. Then, the crosstalk calculation for extended vector set A is provided in part(C). Additionally, in part(D), a small experiment is done, including mutating the original input vector set A with a 20% mutation rate. Then, the BAM network is applied to error-correct the mutated input vector set. Following that, a table is formulated to show the runs, original vector listings, and mutated vector set listing, the hamming distance, the corrected vector set and the hamming distance between the original and error-corrected vector set. Finally, this report concludes with the learning of the experiments done in this assignment.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	BAM Network . . . . .	1
1.2	Part(A) . . . . .	3
1.3	Part(B) . . . . .	5
1.4	Part(C) . . . . .	6
1.4.1	Adding Three more Vectors . . . . .	7
1.5	Part(D) . . . . .	9
1.6	Conclusion . . . . .	10

# 1 Introduction

## 1.1 BAM Network

Bidirectional Associative Memory is a type of recurrent neural network that works using associative memory. Associative memory stores associations between patterns and recalls a pattern when presented with another pattern. BAM is a hetero-associative network that takes two different input and output sets to learn patterns. In other words, given an input pattern, BAM can retrieve another pattern of a different size. For example, take a vector from the input set A [1,-1,1,-1]. If the network is trained properly, then the BAM network will return a pattern [1, 1, -1, 1, 1] that is associated with the given input. This section exhibits a created BAM Network where the network can take an input pattern and return an associated pattern. On the contrary, if an output pattern is given, it can also recall its input pattern association. For example, Diagram 1 shows that an input vector is multiplied with a transposed weight matrix to recall the output pattern associated with the input vector. Inversely, if an output vector is multiplied by the matrix, the associated input pattern with the output vector is returned (Diagram 2). A weight matrix is needed to achieve this BAM network, which helps store the association between input and output patterns. The weight matrix is achieved by applying the following formula:  $w$ .

$$\text{Weight Matrix, } w = \sum_{k=1}^m A_k B_k^T$$

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

$$B^T = \begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

Now, we can calculate  $w$  using the formula: Where: -  $A_k$  is a row from matrix  $A$ . -  $B_k$  is a row from matrix  $B$ . This results in the following calculations:

$$\begin{aligned} w &= A_1 B_1^T + A_2 B_2^T + A_3 B_3^T \\ &= \begin{bmatrix} -1 & 1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \end{aligned}$$

The above formula,  $w$  shows the formula to calculate the weight matrix.

Here, in diagram, a vector input is multiplied with the weight matrix and an associated output vector is provided back. Which shows the association between X to Y.

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

Diagram 1:  $X \cdot W^T = Y$

On the contrary, in this following diagram, an output vector is multiplied with the weight matrix, which results back the input pattern associated with the output which in this case is the input to the BAM Network. Here, the associations shows that,  $Y \cdot X$  is also possible.

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

Diagram 2:  $Y \cdot W = X$

In the following Experiment and Result section, A set of inputs will be provided to an implemented BAM network, following that the result will be provided by emphasizing whether the BAM network recalled properly or not. The second experimental part will include the cross-talk of the Input vectors, The results should contain the calculation table of the cross-talk. Third experiment includes inserting new pairs to the BAM network and retrain the BAM model and showing the generated inference once again by checking the cross talk of the input vectors in terms of the output vectors as well as to show that the network will saturate at one point, will not recall correctly, and cross-talk increases. The Part four, includes an experiment where the original input vectors are mutated with 20% probability and then the hamming distance of original, mutated vectors are calculated. Additionally, for the result, a table is provided showing all the required outputs and a discussion of it.

## 1.2 Part(A)

After creating a BAM network, provided two input vector A, B is used as the input for the BAM Network. The output of the BAM Network is shown in Table 1 and Table 2. Table 1 shows the result for A vector set is given to BAM network as input and the BAM model predicted the Output for each associated input vector in set A. Table 2 shows that the BAM network takes B vector set as input and it predicts the input vectors associated with each output vector in B.

Table 1: Results of A to B Mappings

Input	Target	Output	Result
$A = [-1, 1, 1, 1, -1]$	$[1, 1, -1, 1]$	$[1, 1, -1, 1]$	True
$A = [-1, -1, -1, -1, 1]$	$[1, -1, -1, -1]$	$[-1, -1, 1, -1]$	False
$A = [-1, -1, -1, 1, 1]$	$[-1, -1, 1, 1]$	$[-1, -1, 1, 1]$	True

Table 2: Results of B to A Mappings

Input	Target	Output	Result
$B = [1, 1, -1, 1]$	$[-1, 1, 1, 1, -1]$	$[-1, 1, 1, 1, -1]$	True
$B = [1, -1, -1, -1]$	$[-1, -1, -1, -1, 1]$	$[1, -1, -1, -1, 1]$	False
$B = [-1, -1, 1, 1]$	$[-1, -1, -1, 1, 1]$	$[1, -1, -1, -1, 1]$	False

From the output tables, it is visible that 3 of the predictions were false. Which means the BAM network didn't recall properly. The reason is because of the activation function. The activation function for from A to B is the same as the activation function for from B to A. Which made the model predict inaccurately. With an improved activation function it is possible to improve the BAM models output. In the above table 1 and table 2, follows the below activation function.

$$b_j(t+1) = \begin{cases} 1 & \text{if } y_j > 0 \\ b_j(t) & \text{if } y_j = 0 \\ -1 & \text{if } y_j < 0 \end{cases}$$

This activation function is applied for A vectors to predict B vectors and B vector set to predict A vector set. Meaning, A vector set used for predicting B output vectors, and vice versa.

However, if the activation function from B to A is changed then the results improves.

$$\text{activationFunction1}(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ -1 & \text{if } y < 0 \end{cases}$$

The following threshold function, activateFunction1 is used for forward activation, meaning using vector A set to predict the vector B set. And activateFunction2 is used for backward activation meaning vector B set is used as input for the BAM and the input vector set A is predicted by the model.

$$\text{activationFunction2}(x) = \begin{cases} -1 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Table 3: Results of A to B Mappings

Input	Target	Output	Result
$A = [-1, 1, 1, 1, -1]$	$[1, 1, -1, 1]$	$[1, 1, -1, 1]$	True
$A = [-1, -1, -1, -1, 1]$	$[1, -1, -1, -1]$	$[-1, -1, 1, -1]$	False
$A = [-1, -1, -1, 1, 1]$	$[-1, -1, 1, 1]$	$[-1, -1, 1, 1]$	True

Table 4: Results of B to A Mappings

Input	Target	Output	Result
$B = [1, 1, -1, 1]$	$[-1, 1, 1, 1, -1]$	$[-1, 1, 1, 1, -1]$	True
$B = [1, -1, -1, -1]$	$[-1, -1, -1, -1, 1]$	$[1, -1, -1, -1, 1]$	False
$B = [-1, -1, 1, 1]$	$[-1, -1, -1, 1, 1]$	$[-1, -1, -1, 1, 1]$	True

From the Table 3 and Table 4, it is visible that it indeed improves the results by changing the threshold function.

Furthermore, if the cross-talk interference is too much between two pattern set then it is possible for the similar stored patterns to lead to false recalls.

### 1.3 Part(B)

The Cross-talk Calculation of Vector Set A:

Table 5: Vectors  $A$  and  $B$

Input Vector $A$	Output Vector $B$
$[-1, 1, 1, 1, -1]$	$[1, 1, -1, 1]$
$[-1, -1, -1, -1, 1]$	$[1, -1, -1, -1]$
$[-1, -1, -1, 1, 1]$	$[-1, -1, 1, 1]$

And the magnitude of each vector  $C_i$  is given below,

C0	-0.4	0.8	0.4	0.4
C1	-1.2	-1.2	1.2	0.0
C2	0.4	-0.8	-0.4	-0.8

C0	1.06
C1	2.08
C2	1.26

Figure 1.1: Cross Talk of Vectors

Figure 1.2: Magnitude of Vectors



## 1.4 Part(C)

Table 6: Vectors  $A$  and  $B$

Input Vector $S$	Output Vector $T$
$[-1, 1, 1, 1, -1]$	$[1, 1, -1, 1]$
$[-1, -1, -1, -1, 1]$	$[1, -1, -1, -1]$
$[-1, -1, -1, 1, 1]$	$[-1, -1, 1, 1]$
$[1, 1, 1, 1, 1]$	$[-1, 1, 1, -1]$

The output after adding the new pair provided in the assignment, the output of the BAM network is provided below,

Table 7: Predictions from  $A$  to  $B$

Input $x$	Target	Predicted Output $y$	Result
$[-1, 1, 1, 1, -1]$	$[1, 1, -1, 1]$	$[1, 1, -1, 1]$	True
$[-1, -1, -1, -1, 1]$	$[1, -1, -1, -1]$	$[1, -1, -1, 1]$	False
$[-1, -1, -1, 1, 1]$	$[-1, -1, 1, 1]$	$[-1, -1, 1, 1]$	True
$[1, 1, 1, 1, 1]$	$[-1, 1, 1, -1]$	$[-1, 1, 1, -1]$	True

Table 8: Predictions from  $B$  to  $A$

Output $y$	Target	Predicted Output $x$	Result
$[1, 1, -1, 1]$	$[-1, 1, 1, 1, -1]$	$[-1, 1, 1, 0, -1]$	False
$[1, -1, -1, -1]$	$[-1, -1, -1, -1, 1]$	$[-1, -1, -1, -1, -1]$	False
$[-1, -1, 1, 1]$	$[-1, -1, -1, 1, 1]$	$[1, -1, -1, 0, 1]$	False
$[-1, 1, 1, -1]$	$[1, 1, 1, 1, 1]$	$[1, 1, 1, 1, 1]$	True

After adding one more vector to the BAM model, the model starts predicting more inaccurately. That is because of too much of cross talk between two pairs of vectors increased. Note that, vector magnitude in figure 1.4, shows an increase in magnitude in terms of magnitude shown in figure 1.2. In figure 1.2, vector C0 and vector C2's cross talk increased in figure 1.4, which means after adding one more vector to the BAM model. Now, showing the cross-talk calculation of the new vector pairs. And the magnitude of each vector  $C_i$  is given below,

Furthermore, adding three more vectors to vector set A and B, to show that, at some point the saturation will occur, the patterns stops recalling perfectly, and the cross-talk increases.

C0	-0.6	1.0	0.6	0.2
C1	-0.6	-1.8	0.6	0.6
C2	0.6	-1.0	-0.6	-0.6
C3	-0.2	1.0	0.2	0.6

Figure 1.3: Cross Talk of Vectors

C0	1.33
C1	2.08
C2	1.44
C3	1.2

Figure 1.4: Magnitude of Vectors

#### 1.4.1 Adding Three more Vectors

Table 9: Vectors  $A$  and  $B$

Input Vector $A$	Output Vector $B$
$[-1, 1, 1, 1, -1]$	$[1, 1, -1, 1]$
$[-1, -1, -1, -1, 1]$	$[1, -1, -1, -1]$
$[-1, -1, -1, 1, 1]$	$[-1, -1, 1, 1]$
$[1, 1, 1, 1, 1]$	$[-1, 1, 1, -1]$
$[-1, -1, 1, -1, -1]$	$[1, -1, 1, -1]$
$[-1, 1, -1, 1, -1]$	$[-1, -1, -1, -1]$
$[1, -1, -1, -1, 1]$	$[1, 1, -1, -1]$

The output of the BAM model for the Table 9 Vector input/output pairs is given in Table 10 and Table 11. Note that, in Table 10, Table 11's most of predicted outputs by the BAM model is false. From A to B output as well as B to A output. This shows that after adding three more vectors in input/output set the model saturated. The models saturates is because of too many similar patterns. Additionally, it could also mean that, there are more overlapping patterns which significantly distorts the output. And the weight matrix saturated at one point hence the model's output is rather inaccurate. Again, from Figure 1.7 shows the original vector set A's magnitude of cross talk, Figure 1.8 shows the magnitude of original vector set plus the given vector pairs in the assignment's magnitude, Figure 1.9 randomly adds three more vectors to the previous vector set. Figure 1.7, 1.8, 1.9 shows the increase in cross talk with more included vector pairs. So, we can conclude that, by adding more vector pairs to the model, it saturates to a point where the model is unable to predict properly.

The cross talk of the following input/output set is provided in Figure 1.5, Figure 1.6.

Table 10: Predictions from  $A$  to  $B$ 

Input $x$	Target	Predicted Output $y$	Result
$[-1, 1, 1, 1, -1]$	$[1, 1, -1, 1]$	$[-1, 1, 1, 1]$	False
$[-1, -1, -1, -1, 1]$	$[1, -1, -1, -1]$	$[1, -1, -1, -1]$	True
$[-1, -1, -1, 1, 1]$	$[-1, -1, 1, 1]$	$[-1, -1, -1, 1]$	False
$[1, 1, 1, 1, 1]$	$[-1, 1, 1, -1]$	$[-1, 1, 1, 1]$	False
$[-1, -1, 1, -1, -1]$	$[1, -1, 1, -1]$	$[1, -1, -1, -1]$	False
$[-1, 1, -1, 1, -1]$	$[-1, -1, -1, -1]$	$[-1, -1, -1, 1]$	False
$[1, -1, -1, -1, 1]$	$[1, 1, -1, -1]$	$[1, -1, -1, -1]$	False

Table 11: Predictions from  $B$  to  $A$ 

Output $y$	Target	Predicted Output $x$	Result
$[1, 1, -1, 1]$	$[-1, 1, 1, 1, -1]$	$[1, -1, 1, -1, 1]$	False
$[1, -1, -1, -1]$	$[-1, -1, -1, -1, 1]$	$[-1, -1, -1, -1, -1]$	False
$[-1, -1, 1, 1]$	$[-1, -1, -1, 1, 1]$	$[-1, 1, -1, 1, -1]$	False
$[-1, 1, 1, -1]$	$[1, 1, 1, 1, 1]$	$[1, 1, 1, 1, 1]$	True
$[1, -1, 1, -1]$	$[-1, -1, 1, -1, -1]$	$[-1, -1, -1, -1, -1]$	False
$[-1, -1, -1, -1]$	$[-1, 1, -1, 1, -1]$	$[-1, -1, -1, -1, -1]$	False
$[1, 1, -1, -1]$	$[1, -1, -1, -1, 1]$	$[1, -1, 1, -1, 1]$	False

C0	-2.0	-0.8	1.2	0.4
C1	0.4	-1.2	0.4	0.0
C2	0.4	-0.8	-1.2	-0.8
C3	-0.8	1.6	0.0	1.6
C4	1.2	-0.4	-0.8	0.8
C5	-0.4	0.0	0.0	2.0
C6	0.0	-1.2	0.8	-0.4

Figure 1.5: Cross Talk of Vectors

C0	2.5
C1	1.33
C2	1.7
C3	2.4
C4	1.7
C5	2.04
C6	1.5

Figure 1.6: Magnitude of Vectors

C0	1.06
C1	2.08
C2	1.26

Figure 1.7: Magnitude of Vectors

C0	1.33
C1	2.08
C2	1.44
C3	1.2

Figure 1.8: Magnitude of Vectors

C0	2.5
C1	1.33
C2	1.7
C3	2.4
C4	1.7
C5	2.04
C6	1.5

Figure 1.9: Magnitude of Vectors

## 1.5 Part(D)

In this section, the table is explained in depth. The table contains the 20 recordings of runs, original vector set from Part(A), then a vector is randomly selected to create a mutated vector by flipping the bits, following that, the Hamming distance between original and the mutated vectors, then the mutated vectors were sent through recall function to self correct itself, then hamming distance between original and error corrected is shown, lastly, hamming distance between mutated and error corrected vectors are shown in the table.

Run	Original(O)	Mutated (M)	Hdist(O,M)	Error Corrected (C)	Hdist(O,C)	Hdist(M,C)
1	[-1, 1, 1, 1, -1]	[-1, -1, 1, 1, -1]	1	[-1, 1, 1, 1, -1]	0	1
4	[-1, -1, -1, 1, 1]	[-1, -1, 1, 1, 1]	1	[1, -1, -1, -1, 1]	2	3
9	[-1, 1, 1, 1, -1]	[1, -1, 1, 1, -1]	2	[-1, 1, 1, 1, -1]	0	2
9	[-1, -1, -1, 1, 1]	[-1, 1, 1, 1, 1]	2	[-1, 1, 1, 1, -1]	3	1
13	[-1, -1, -1, -1, 1]	[-1, 1, -1, -1, 1]	1	[1, -1, -1, -1, 1]	1	2
13	[-1, -1, -1, 1, 1]	[-1, 1, -1, 1, 1]	1	[1, -1, -1, -1, 1]	2	3
15	[-1, -1, -1, -1, 1]	[-1, -1, -1, -1, 1]	0	[1, -1, -1, -1, 1]	1	1
15	[-1, -1, -1, 1, 1]	[-1, 1, 1, 1, 1]	2	[-1, 1, 1, 1, -1]	3	1
17	[-1, -1, -1, -1, 1]	[-1, 1, -1, -1, 1]	1	[1, -1, -1, -1, 1]	1	2
18	[-1, -1, -1, -1, 1]	[-1, 1, 1, -1, 1]	2	[-1, 1, 1, 1, -1]	4	2
19	[-1, -1, -1, -1, 1]	[-1, 1, -1, -1, 1]	1	[1, -1, -1, -1, 1]	1	2

Figure 1.10: PART D - Table

From the table, it is visible that 20 runs were done, however due to mutation rate, the actual run became shorter. Sometimes the program didn't choose to mutate therefore the program terminated if not mutation probability is satisfied. In the mutated section, the mutated vectors differ by 1 bit, since flip bit mutation mechanism was used. Hdist(O,M) shows the hamming distance between original and the mutated vector sets. In the Error Corrected(C) column, the program self corrected the vectors by going 6 times through the recall function for each vector, and tried to get the original vector. From the Hdist(O,C), it shows that sometimes the model is able to correct the mutated vectors, and in some other cases it stays the same. Also, sometimes the model messes the vector while the hamming distance between the original and error corrected vector increases. Only 2 times the model was able to correct the mutated vector properly. While few times the distance between the original and error corrected vectors decreases, few other times it actually in-

creased from the original and mutated distance. The hamming distance between mutated and error corrected sometimes increased by more than 1 bit, or decreased by 1 bit but the model never was able to correctly self correct the mutated input. This is initially because, the model's weight matrix pattern association is limited. Since the model associated the pattern of the initially assigned vectors, it's hard to for the model to predict a mutated vector to it's original form. Futhermore, the memory of BAM model is limited as well.

## 1.6 Conclusion

All the above experiment showed that BAM is a hetero-associative memory. BAM predicts an association B for an input vector A. BAM uses a weight matrix to remember its association with input to output or output to input. BAM is limited, it can only recall the associations properly if the association pattern is stored within the weight matrix. BAM produces cross talk between input and output vectors. The magnitude of the cross talk vector is the interference between the input and output vector. With more inputs, the BAM network increases it's cross talk, meaning the interference. It is possible to error correct a distorted input however, the model has it's limitations.