International Journal of Engineering Research and Advanced Technology (IJERAT)

ISSN: 2454-6135 (Online) [Volume. 02 Issue.02, February-2016]

Novel Hetero-Associative Memory: A Modified Bidirectional Associative Memory

Nisreen Abd Alhadi Jabr[1] and Emad I Abdul Kareem[2]

[1,2]College of education in Computer science, AL Mustansiriya University,

Baghdad, Iraq.

[1] Correspondence author

Abstract

Associative memory is a data collectively stored in the form of a memory or weight matrix, which is used to generate output that corresponds to a given input, can be either auto-associative or hetero-associative memory. A Bidirectional Associative memory neural network is one of the most commonly used neural network models for hetero-association and optimization tasks, it has several limitations. For example, it is well known that Bidirectional Associative memory neural networks has limited stored patterns, local minimum problems, limited noise ratio and shifting and scaling problems. This research will suggest to improve the Bidirectional Associative Memory neural network by modifying the net architecture, learning and convergence processes, and this modification is to increase the performance of the associative memory neural network by avoiding most of the Bidirectional Associative Memory neural network (BAM) limitations. This research will propose to modify Bidirectional associative memory (MBAM) to improve the efficiency of BAM in order to decrease the network size and weight size. In additional to increase the ability for noise robust as well as speed up its learning and convergence process. The results proved that the MBAM net can learn and recognize unlimited patterns in varying sizes with the acceptable percentage noise and overcomes most of limitations BAM except shifting and scaling problems.

Key Words: *Associative memory, Bidirectional Associative Memory, Neural network, Pattern recognition.*

1. INTRODUCTION

Bidirectional Associative Memory neural network is one of the neural networks that are used for hetero-association and optimization tasks [1][11]. This net has many limitations that are affecting its performance. These limitations consisted: firstly the number of patterns that can be stored and accurately recalled is severely limited [2]. Secondly, it is obviously desirable to reach a global minimum rather than sitting down at a local minimum as it can happen while using the energy function [3] [12]. Thirdly correlation problem happens when an input pattern share many bits with another pattern [4] [13]. Fourthly the ratio of missing and mistake data in the input patterns is limited [5]. Fifthly, it is impossible to retrieve the stored pattern when it enters to the network with shifting or scaling [6].

MBAM net avoids most of the BAM limitation, except two which it the shifting and scaling problem, in addition to the smaller size of net and the efficient learning and convergence process. This research proposed at present to improve the efficiency of MBAM in order to decrease the network size and weight size. In additional to increase the ability for noise robust as well as speed up its learning and convergence process. Similar to the BAM neural network and MBAM is a two-layer neural network, which uses hetero-association tasks and work in two phases (learning and convergence phases).

The experiments performed show promising results when MBAM shows high efficiency to recognize many noisy patterns in varying size compared with the traditional Bidirectional Associative Memory (BAM) neural network.

2. ASSOCIATIVE MEMORY

It is believed that human memory is stored in the form of complex interconnections among various neurons. In artificial neural networks simulating associative memory, data are collectively stored in the form of a memory or weight matrix, which is used to generate output that corresponds to a given input. Such a process is referred to as learning or storing the desired patterns, while the retrieval or recall process is referred to as the generation of an output pattern [13][7].

An associative memory can be applied in either auto-associative or heteroassociative applications. Mathematically, it is a mapping from an input space to an output space. In other words, when the network is presented with a pattern similar to the member of the stored set, it may associate the input with the closed stored pattern [15][8].

Generally, in hetero-associative applications, the dimensions of the input space and the output space are different, as illustrated in Figure 1 [15] [14].

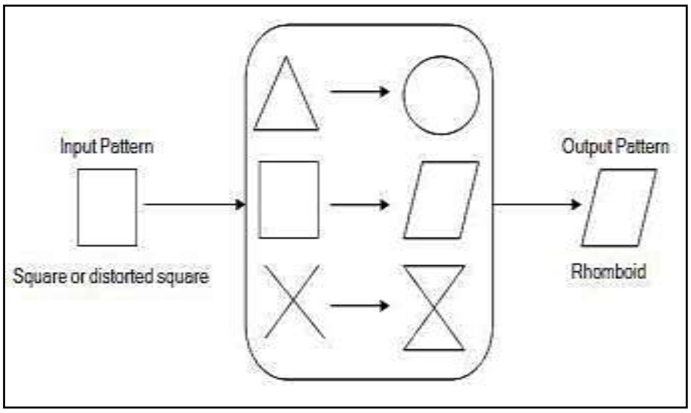

Figure 1: Hetero-association response

One of the hetero-associative memory neural networks is the Bidirectional Associative Memory (BAM), which is presented in the next section.

3. BIDIRECTIONAL ASSOCIATIVE MEMORY (BAM) NEURAL NETWORK

This section presents one of the neural network models; i.e., the “Bidirectional Associative Memory (BAM) network”, which is the most commonly used for heteroassociation and optimization tasks. The Bidirectional associative memory is heteroassociative, developed by Kosko (1988, 1992a). A BAM consists of neurons arranged in two layers X and Y. The neurons in one layer are fully interconnected to the neurons in the second layer. There is no interconnection among neurons in the same layer. The weight from layer X to layer Y is same as the weights from layer Y to layer X [9][16].

Dynamics involves two layers of interaction. Because the memory process information in time and involves bidirectional data flow, it differs in principle from a linear association, although both networks are used to store association pairs. It also differs from the recurrent auto- associative memory in its update mode [14][17]. The next subsection will discuss the architecture of the Bidirectional associative Memory neural network.

The bidirectional associative memory Neural Network is the hetero-associative system. The single-layer nonlinear feedback BAM network (with hetero-associative content- addressable memory) has n units in its X-layer and m units in its Y-layer. The connections between the layers are bidirectional; i.e., if the weight matrix for signals sent from the X-layer to the Y-layer is W, the weight matrix for signals sent from the Y-layer to the X-layer is WT, show Figure 2, Architecture of the BAM [15][10][18].

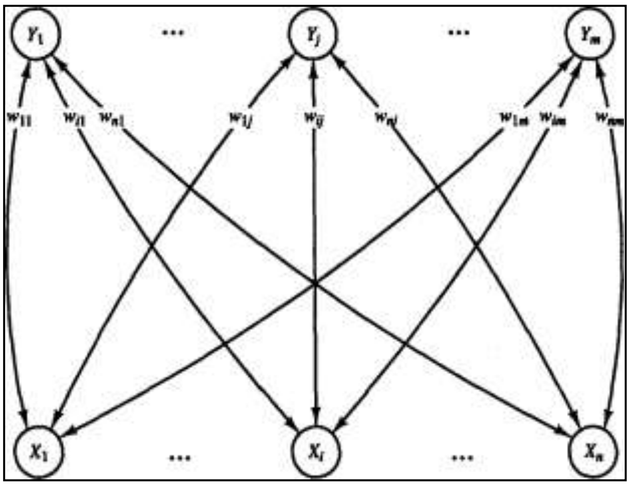

Figure 2: The architecture of the Bidirectional Associative Memory (BAM), where:
Wij: Weights Y1, Y2,…, Yn and X1, X2,…, Xn Neurons.

4. MODIFY BIDIRECTIONAL ASSOCIATIVE MEMORY (MBAM).

The size of a BAM network (number of neurons in the network) depends on the pattern length (i.e., a pattern with length *LP* and code with *LC* requires a BAM network of size *LP*LC*). Where MBAM will be 2**LC* because the pattern with length *LP* will be divided to many vectors with length 2. MBAM will learn each vector independently, to be stored in a lookup table. According to the first principle, the pattern will be divided into a number of vectors with length two; this means that the size of the network will be fixed (i.e., four). Therefore, the network deals with parts of the pattern instead of the entire pattern as one vector. This leads to the advantage of working with the smallest network size regardless of the pattern length, as well as multiple connections between the four neurons (two neurons for each layer, i.e. input layer and output layer). The new architecture allowed the possibility of avoiding learning the same vectors (that represents specific part of the pattern) several times. This arrangement achieves the second principle described above (see Figure 3).

With a bipolar pattern representation, the elements will be either 1 or -1. The reason for choosing this length of the vector is the shortest even length of any vector is two. However, just as in the traditional BAM neural network, each node in the input layer is connected to every node in output layer but not to itself. These connections represent the corresponding weight of each vector in the pattern. Although the expected number of vectors is, the number of connections will be just four, an advantage that helps deal with the smallest network size regardless of the pattern length. Technically, as with traditional BAM nets, this modified net has two phases (learning and convergence phases). In this research, these two processes will be modified.

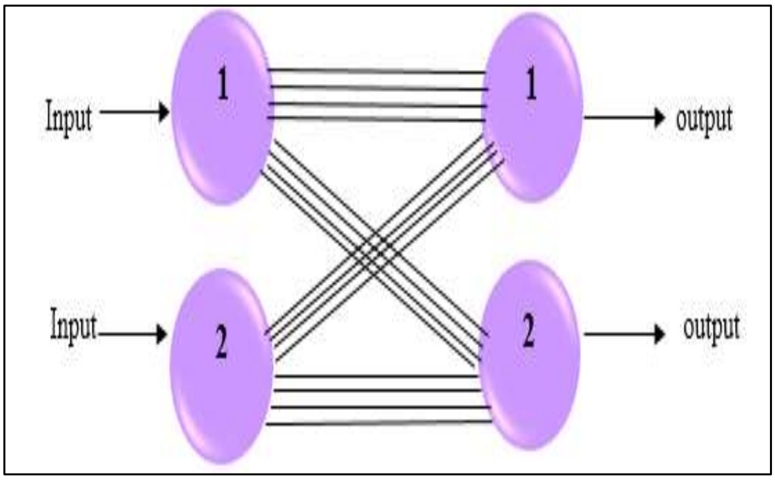

Figure 3: The Modified bidirectional associative memory (MBAM)

4.1 Learning Phase (learning process)

This section presents the learning algorithm shown in Figure 4. The network is able to recognize all the required bipolar patterns through the implementation of this phase. Therefore, the input for this phase is a flow of patterns and codes to be considered as training patterns for this network. The output of this phase is the results of the learning process, which is stored in a specific lookup table. Since the training patterns are divided into vectors with length two, these results are a set of weights indexes for each vector of the training patterns, which are called stored vector weights *svw*.

```
Algorithm 1: Learning phase for MBAM
Input: training patterns $p$ with code $c$.
Output: lookup table for all $n$ corresponding stored patterns.
Step1: Repeat steps 1.1 and 1.2 to the end of training pattern $p$ with code $c$:
    Step 1.1: Divide the training pattern $p$ to $n$ vectors $v$ with length two.
    Step 1.2: For each vector $v$, repeat steps 1.2.1, 1.2.2 and 1.2.3:
        Step 1.2.1: Assign the weight for each $n$ vectors $v$, weights matrix $w$ as follows:
                                $w_i = v_i * c$
Where: $c$ is code
        Step 1.2.2: Assign the stored vector's weight $svw$ as follow:
$svw = f(Dcode(v)) \begin{cases} 0 & \{means w0 \\ 1 & \{means w1 \\ 2 & \{means w2 \\ 3 & \{means w3 \end{cases}$
Where: Decode is a function to convert the binary number to decimal number.
            Step 1.2.3: Save $svw$ for this vector in the lookup table.
Step 2: End.
```

Figure 4. Algorithm 1 of the learning phase.

4.2 Convergence Phase (convergence process)

The output of this phase is dependent on the learning phase. Although the modification has been carried out during the learning phase, it is necessary to have a comprehensive modification of the convergence phase (see Figure 5, 6. 7 and 8) in order to ensure the overall efficiency of the Modified Bidirectional Associative Memory (MBAM). Convergence phase has two Bi-direction, the first direction is code to pattern and the second direction is a pattern to code.

4.2.1 Convergence Phase (pattern to code)

There are three proposed patterns to code convergence methods as follows:

- Traditional convergence method.
- Random convergence method.
- The first and last quarter convergence method.

4.2.2 *Traditional Convergence Method.*

The convergence code algorithm shows that the unknown pattern is divided into k vectors with each two elements. The modification process ensures the increasing role of the energy function in the convergence process. The energy function is a function whereby whenever the state of any node changes, the functions, permanent decreases progressively to arrive the minimum.

This phase will be stopped when all the vectors with length two in the unknown pattern are multiply with svw, they stored in the lookup table with the minimum energy function, that lead to the end of the convergence process. In the MBAM, an energy function is utilized to correlate the unknown pattern and the codes that have stored its data (i.e. *svw*) in the lookup table through the learning phase.

Algorithm 2: The convergence phase algorithm for the traditional convergence.

```
Input: n of unknown patterns p.
Output: Convergence code c.
Step 1: Repeat steps 1.1 and 1.2 until the end of unknown pattern p:
        Step 1.1: Divide the unknown pattern p to n vectors v with length two.
        Step 1.2: Sum up the energy function for all n vectors v in the unknown pattern each with its
        corresponding vector in the stored patterns:
                $ep = -0.5 \sum_{i=1}^{n} v_i * svw_i * y_i$
Where $y_i$ is :
                $y_i = v_i * svw_i$
Step 2: Determine the stored code number $minc$ with the minimum energy function to converge the unknown
pattern towards it:
    $minc = \min(ep)$
    Where the min function is to determine the minimum energy function in ep array.
Step 3: End.
```

Figure 5. Algorithm 2: The convergence code algorithm for the traditional convergence method.

4.2.3 Random Convergence Method.

This method used part of the pattern instead of all pattern. Thus, this algorithm just like the previous method the pattern will be divided to vectors with length two. Unlike the previous method (see Figure 5), this method will not use all these vectors. Thus, this method will choose the vectors randomly with the particular ratio (i.e. R %).

4.2.4 The first and last quarter Convergence Method.

This method used the first and the last quarter of the size pattern instead of all pattern. Just like the previous method (see Figure 5), the pattern will be divided to vectors with length two. Unlike the previous method, this method will not use all these vectors. Thus, this method will choose the vectors the first and last quarter with particular ratio

```
Algorithm 3: The convergence phase algorithm for the random convergence.
Input: $n$ of unknown patterns $IP$, Ratio of random locations of the pattern $R$.
Output: Convergence code $c$.
Step 1: Repeat steps 1.1 , 1.2 and 1.3 until the end of unknown pattern $IP$:
    Step 1.1: Taking random elements from unknown pattern $IP$ with particular ratio $R$%, called $p$.
    Step 1.2: Divide the unknown pattern $p$ to $n$ vectors $v$ with length two.
    Step 1.3: Sum up the energy function for all $n$ vectors $v$ in the unknown pattern each with its
    corresponding vector in the stored patterns:
        $ep = -0.5 \sum_{i=1}^{n} v_i * svw_i * y_i$
Where $y_i$ is :
        $y_i = v_i * svw_i$
Step 2: Determine the stored code number $minc$ with the minimum energy function to converge the
unknown pattern towards it:
        $minc = \min(ep)$
  Where the min function is to determine the minimum energy function in $ep$ array .
Step 3: End.
```

Figures6. Algorithm 3: The convergence code algorithm for the Random convergence.

```
Algorithm 4: The convergence phase algorithm for the first and last quarter convergence.
Input : $n$ of unknown patterns $p$.
Output : Convergence code $c$.
Step 1: Divide unknown pattern to four quarters, then used first quarter $q1$ and last quarter $q4$.
Step 2: Repeat steps 2.1 and 2.2 until the unknown pattern $p$ is ended:
    Step 2.1: Divide the unknown pattern $q1$ and $q4$ to $n$ vectors $v$ with length two.
    Step 2.2: Sum up the energy function for all $n$ vectors $v$ in the unknown pattern each
    with its corresponding vector in the stored patterns:
        $ep = -0.5 \sum_{i=1}^{n} v_i * svw_i * y_i$
Where $y_i$ is :
        $y_i = v_i * svw_i$
Step 3: Determine the stored code number $minc$ with the minimum energy function to
converge the unknown pattern towards it:
        $minc = \min(ep)$
    Where the min function is to determine the minimum energy function in $ep$ array .

    Step 4: End.
```

Figure 7. Algorithm 4: The convergence code algorithm for the first and last quarter convergence

4.3Convergence phase (code to pattern):

The convergence phase is carried out via computing the sum of all the values of the energy function between unknown code and each corresponding weight stored in the lookup table. Determine the stored pattern with minimum energy function to converge towards it. Finally, the pattern will be built according to the chosen pattern from the lookup table with the input code. It is worth mentioning that the convergence phase is not a repeated process like the bidirectional associative memory.

```
Algorithm 5: b. Convergence phase algorithm for the MBAM
Input: $n$ of unknown codes $c$.
Output: Convergence pattern $CP$.

Step 1: Repeat steps 1.1 until the lookup table is ended:
        Step 1.1: Find energy function $ep$:
                $ep = -0.5 * \sum_{i=1}^{n} x_i . c$
Where $x_i$ :
                $x_i = y_i * svw_i$
Where $y_i$ :
                $y_i = c * svw_i$
Where $c$: is unknown code
$svw$: is matrix weights stored in lookup table.

Step 2: Determine the stored pattern number $minp$ with the minimum energy function to converge the
        unknown code towards it:
$minp = \min(ep)$

   Where the $min$ function is to determine the minimum energy function in $ep$ array.

Step 3: Build the final converge pattern $cp$:
                $cp = svw_{minp} * c$

Step 4: End.
```

Figure 8. Algorithm 5: The convergence pattern algorithm for the MBAM.

5. EVALUATION OF MBAM

This section presents three experiments carried out to evaluate the MBAM associative memory. The experimental protocols were applied to both the traditional BAM neural network and the MBAM associative memory to allow comparison. The experiments considered the efficiency of the MBAM associative memory and were compared with the traditional BAM neural network by analyzing the noise rate (missing and mistake bits), a large number of training patterns and the small size of the net, which was two with all training pattern lengths and/or numbers.

5.1 Different Number of Stored Patterns vs. Convergence Rate

In this experiment, for both MBAM associative memory and the traditional BAM network, the task was to learn the maximum number of patterns with codes for the pattern; the process was stopped when one/both of them completely failed to recognize the stored patterns. The test patterns that were presented were alphabetical letters in 32×32 size without noise (see Figure 9).

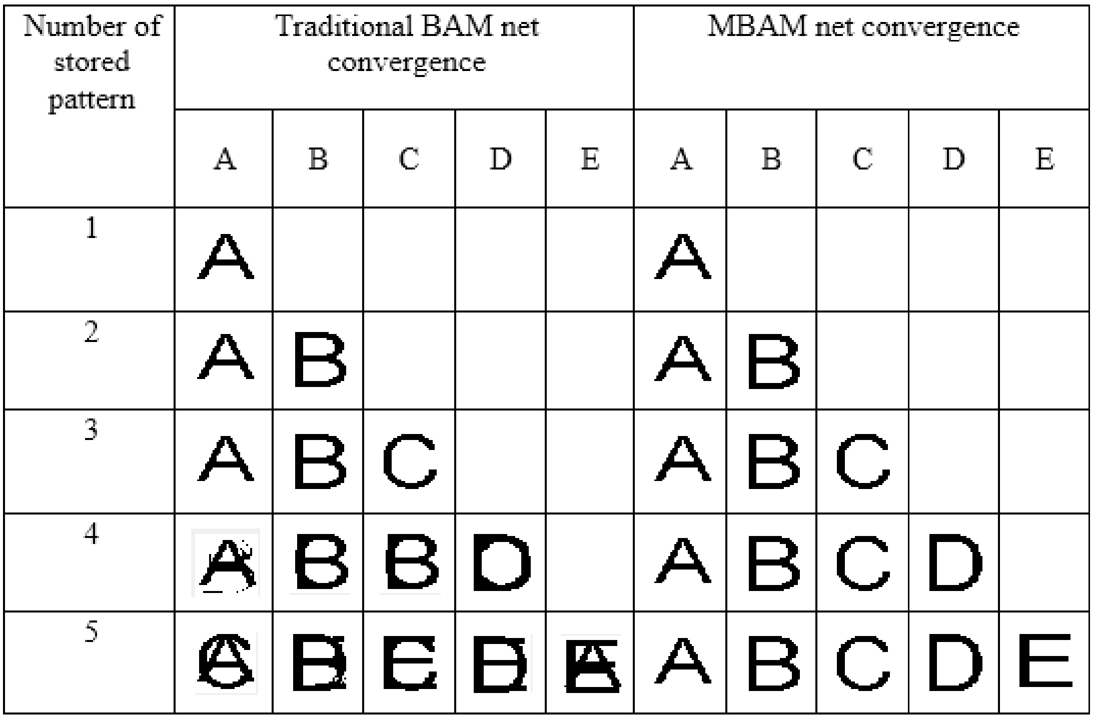
Number of stored pattern
Traditional BAM net convergence
MBAM net convergence
A
B
C
D
E
A
B
C
D
E
1
2
3
4
5

Figure 9. Traditional BAM net and MBAM learned four alphabetical English letters

In Figure 9, it is clear that the learning process happened gradually, starting from pattern one to five, to make sure that both were able to retrieve most of the patterns until one of them or both reached the case of complete failure. The BAM neural network began to fail to recognize some patterns when the number of patterns that had been learned was 4. The complete failure occurred when the number of patterns became

5.2 Results Discussion and Analysis

Due to a correlation problem, the convergence rate of the traditional BAM net decreased when the number of stored patterns increased. Convergence failure started with 4 stored patterns and complete failure to recognize any stored pattern occurred with 5 stored patterns. Whereas MBAM associative memory kept the same convergence rate (which was 100%) even when the number of stored patterns increased to 5 (see Figure 10).

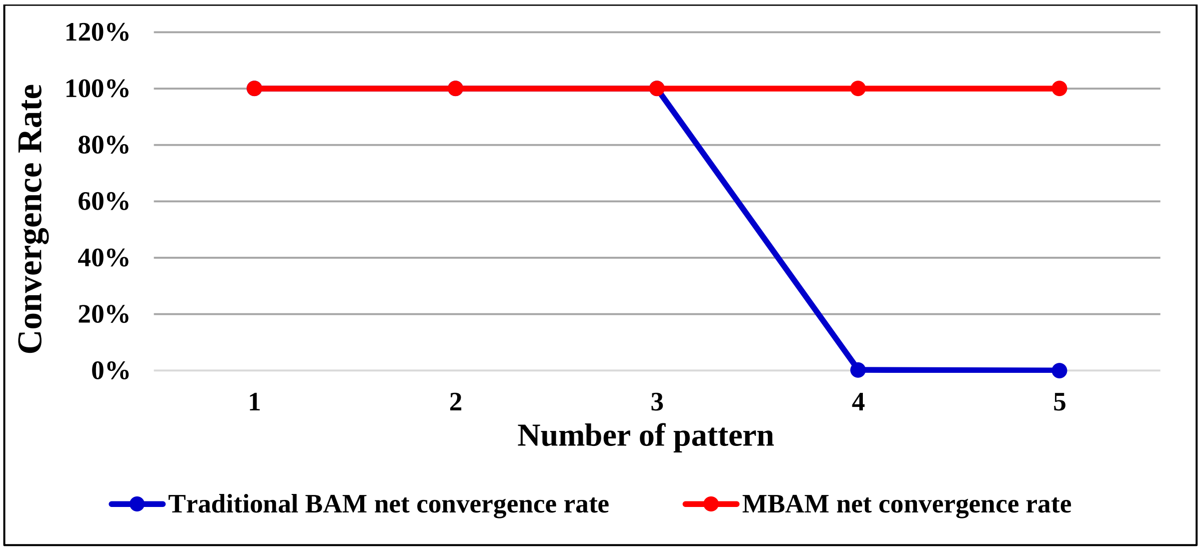

Figure 10. The diagram illustrates a comparison between traditional BAM net and MBAM net for a number of stored patterns vs. Convergence rate for both of them.

The grid size of the stored patterns in this experiment was 32×32. Thus, the traditional BAM net size was 1024 nodes (or neurons), however, the MBAM associative memory, network was maintained at a 4 node size because, as mentioned previously, the traditional BAM net size depends on the pattern size, whereas the MBAM net size is 4 nodes with any pattern size.

5.3 Different Noise Rates vs. Convergence Rate Depending on the Previous

Based on the previous experiment, traditional BAM net convergence maintained efficiency at 100% with input patterns having 0% noise until the number of stored patterns was 3 (see Figure 9). Thus, in the experiment, the MBAM associative memory, and traditional BAM nets learned just 3 patterns with a grid size of 32×32. This experiment had three tests with each having three different random noise patterns for each pattern (A, B and C). The convergence process was implemented with both by testing the input of these patterns with 10% to 90% random noise.

Table 1 shows the responses of both networks when the same three forms that the networks had learned previously were presented with different noise ratios ranging from 10% to 90%.

Noise Ratio	BAM	MBAM		
		Traditional Method	Random Method	Quarter Method
	Convergence ratio of Patterns with size 32×32	Convergence ratio of Patterns with size 32×32	Convergence ratio of Patterns with size 32×32	Convergence ratio of Patterns with size 32×32
10%	96.67%	100%	100%	100%
20%	88	100%	100%	100%
30%	86.33%	100%	100%	99.67%
40%	79.67%	99.67%	96.33%	96.33%
50%	75.33%	96.33%	85%	83.67%

60%	70.67%	82.33%	74%	70.33%
70%	65.67%	64.67%	57.33%	55.33%
80%	59.67%	48.33%	45.67%	41.67%
90%	57.67%	36.67%	39.67%	36%

Table I. Diagram illustrating the comparison between the traditional BAM net, MBAM net for input patterns with different noise rates vs. convergence rates for each.

5.4 Results, Discussion and Analysis

The efficiency of the BAM network convergence rate was 96.67% with a 10 % noise rate, where the network where low to converge to most of the stored patterns even the noise rate become 60% due to the local minimum problem. Where this problem prevents the BAM neural network to correctly converge to one of the stored pattern, it would generate a new pattern which is not concerning to any stored patterns. In comparison with the MBAM associative memory, the efficiency, convergence washes high with all random noise percentage.

Figure 11 represents the comparison of the accepted noise percentage for each MBAM and BAM network, whereas, Figure 12 represents the standard deviation of the 100 test results for both networks. Standard deviation measures the spread of the data about the mean value. It is helpful in comparing the two sets of the network tests results for both networks. Figure 12 shows that the test results corresponding to MBAM have a low standard deviation and the values aren't spread out too much explaining the stabilization of responses for MBAM. On the contrary, the test results corresponding to BAM neural network values are more spread out because of the correlation and local minimum problems.

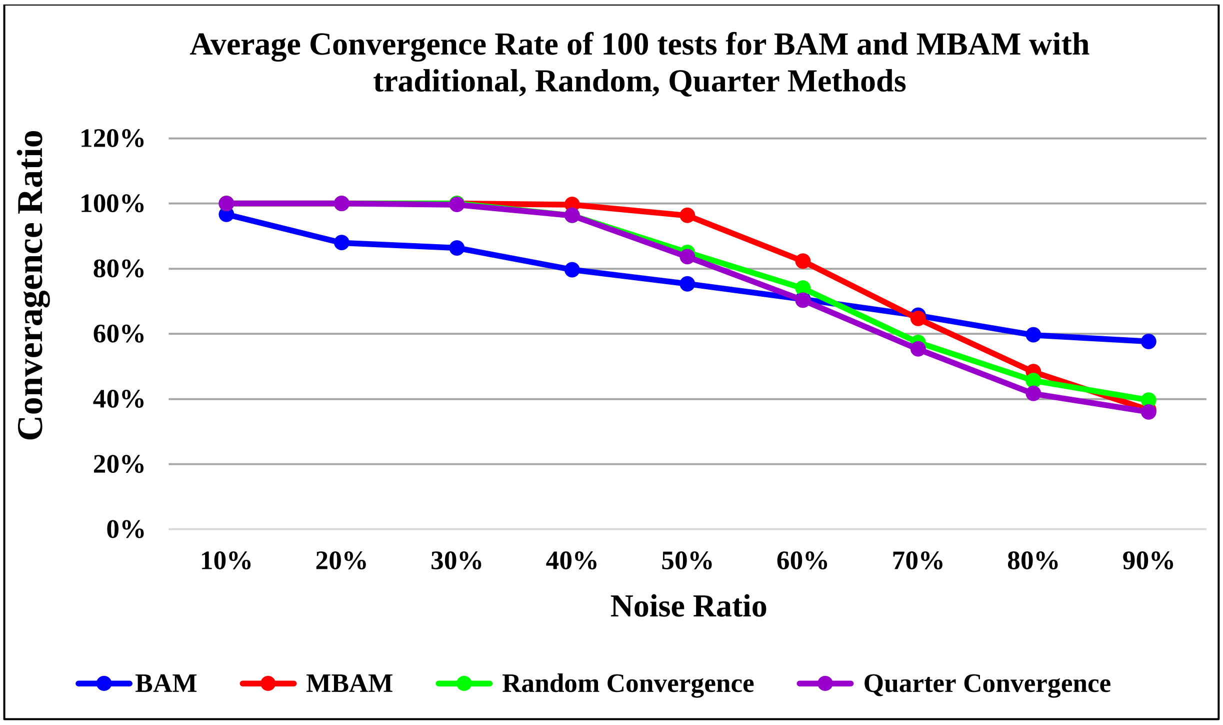
Average Convergence Rate of 100 tests for BAM and MBAM with traditional, Random, Quarter Methods

Figure 11. A diagram illustrating the comparison between the traditional BAM net and the MBAM associative memory network for input patterns with different noise rates vs. convergence rates for each.

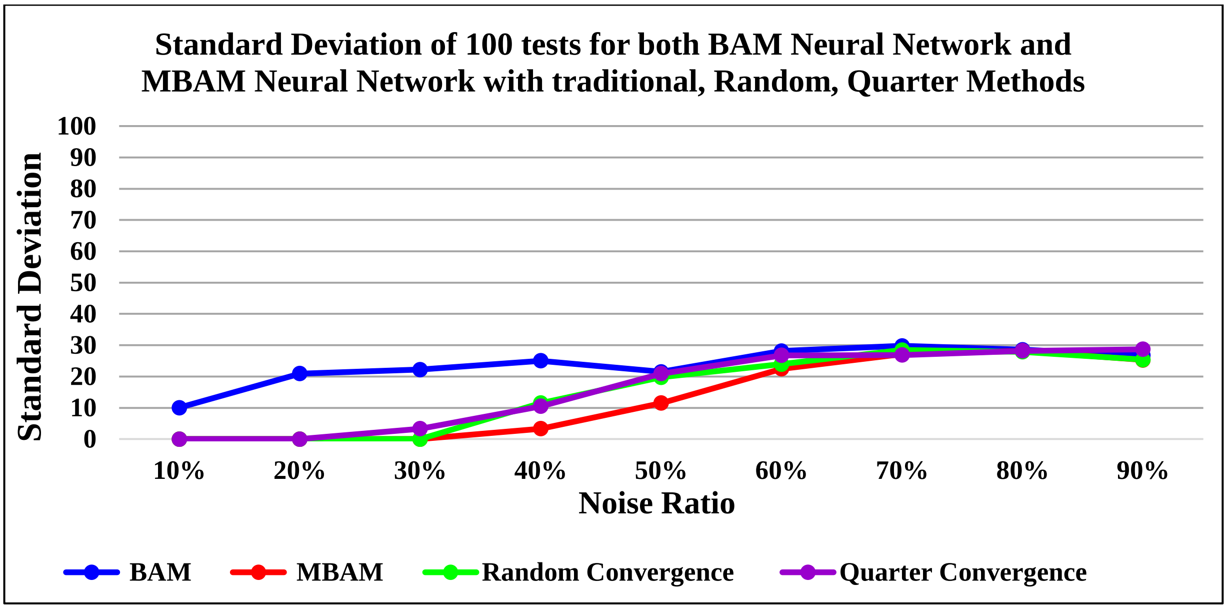

Figure 12: The standard deviation for the100 test results for both the BAM neural network and the MBA

5.5 Different Patterns Sizes with Different Noise Rates vs. Convergence Rates for MBAM Associative Memory

In this experiment, 100 patterns were presented in various sizes to the traditional method associative memory network with different noise ratios as unknown images to determine convergence rates (see Table 2).

Noising Ratio	MBAM (Traditional Method)				
	Convergence ratio of Pattern with size 10×10	Convergence ratio of Patterns with size 16×16	Convergence ratio of Patterns with size 32×32	Convergence ratio of Patterns with size 64×64	Convergence ratio of Patterns with size 128×128
10%	100%	100%	100%	100%	100%
20%	99.67%	100%	100%	100%	100%
30%	99%	99.67%	100%	100%	100%
40%	93%	95%	99.67%	100%	100%
50%	80.33%	84.33%	96.33%	100%	100%
60%	61.33%	71%	82.33%	98 %	100%
70%	48.33%	55.33%	64.67%	77.33%	100%
80%	43.33%	45.33%	48.33%	54.67%	100%
90%	39.33%	40%	36.67%	38%	100%

Table 2. Five tests of 100 different size patterns for the traditional method associative memory net with convergence rates.

Also, this experiment 100 patterns were presented in various sizes to the Random method associative memory network with different noise ratios as unknown images to determine convergence rates (see Table 3).

Noising Ratio	MBAM (Random Method)				
	Convergence ratio of Pattern with size 10×10	Convergence ratio of Patterns with size 16×16	Convergence ratio of Patterns with size 32×32	Convergence ratio of Patterns with size 64×64	Convergence ratio of Patterns with size 128×128
10%	98.67%	100%	100%	100%	100%
20%	94.67%	100%	100%	100%	100%
30%	81.33%	92.67%	100%	100%	100%
40%	72.67%	84.67%	96.33%	95.67%	100%
50%	62%	71.67%	85%	85.33%	100%
60%	49.33%	57.33%	74%	73.67%	92.33%
70%	44.33%	50.33%	57.33%	64.33%	72%
80%	40.67%	38.67%	45.67%	51.67%	59.33%
90%	35.33%	34%	39.67%	45%	56.33%

Table 3. Five tests of 100 different size patterns for the Random method associative memory net with convergence rates.

Lastly, in this experiment 100 patterns were presented in various sizes to the first and last quarter method associative memory network with different noise ratios as unknown images to determine convergence rates (see Table 4).

Noising Ratio	MBAM (The first and last quarter Method)				
	Convergence ratio of Pattern with size 10×10	Convergence ratio of Patterns with size 16×16	Convergence ratio of Patterns with size 32×32	Convergence ratio of Patterns with size 64×64	Convergence ratio of Patterns with size 128×128
10%	99%	100%	100%	100%	100%
20%	98%	98.33%	100%	100%	100%
30%	88.33%	93.67%	99.67%	100%	100%
40%	79.33%	83.67%	96.33%	99.33%	100%
50%	67.33%	66%	83.67%	95.67%	100%
60%	51.33%	58.67%	70.33%	88.67%	91.67%
70%	48.33%	47.33%	55.33%	67.67%	71.67%
80%	41.33%	36.67%	41.67%	49.33%	50.67%
90%	34%	31.33%	36%	40.67%	34.67%

Table 4. Five tests of 100 different size patterns for the first and last quarter method associative memory net with convergence rates.

5.6 Results Discussion and Analysis

Tables (2, 3 and 4) shows that the MBAM associative memory convergence rate was more efficient when the size of the patterns was increased (see Figures 13, 14 and 15). Once again this situation occurred due to the random noise operation, which was carried out earlier in the Second Experiment.

These Figures illustrating a comparison between the five tests of 100 different size patterns with different noise rates for the three methods net vs. convergence rates for each pattern size.

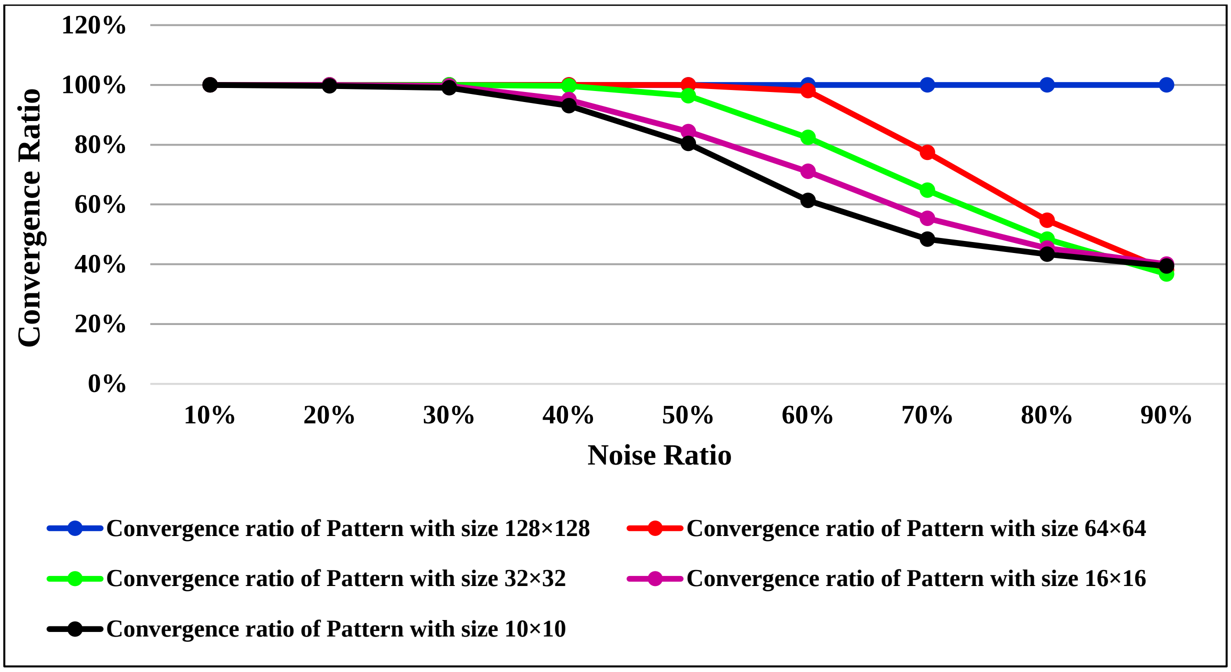

Figure 13: Diagram illustrating a comparison between the five tests of 100 different size patterns with different noise rates for the traditional method net vs. convergence rates for each pattern size.

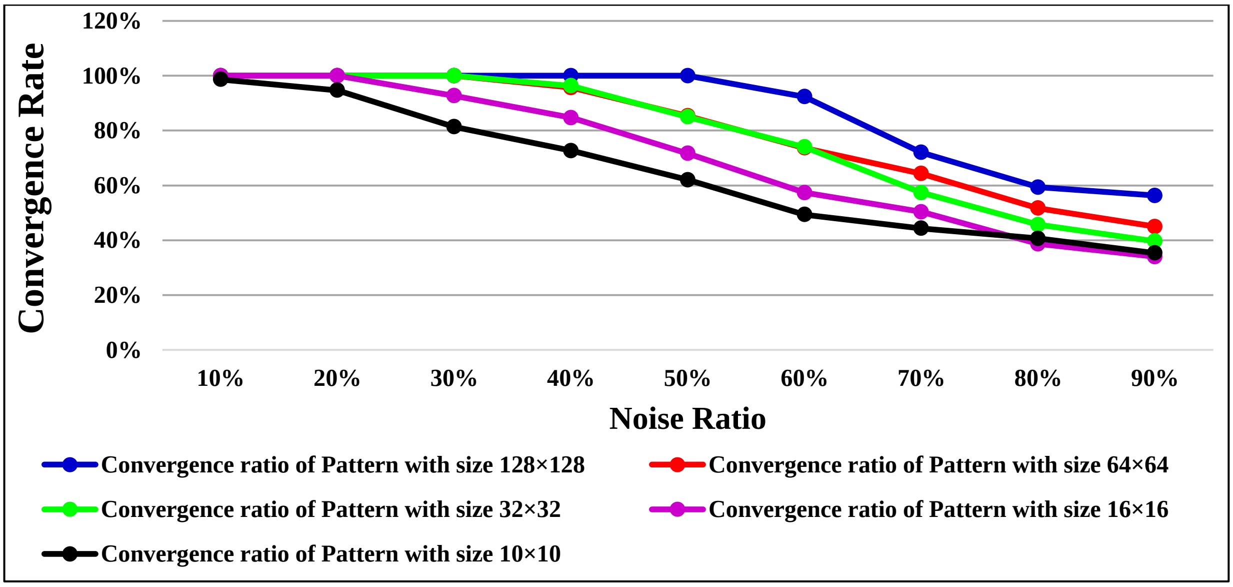

Figure 14: Diagram illustrating a comparison between the five tests of 100 different size patterns with different noise rates for the Random method net vs. convergence rates for each pattern size.

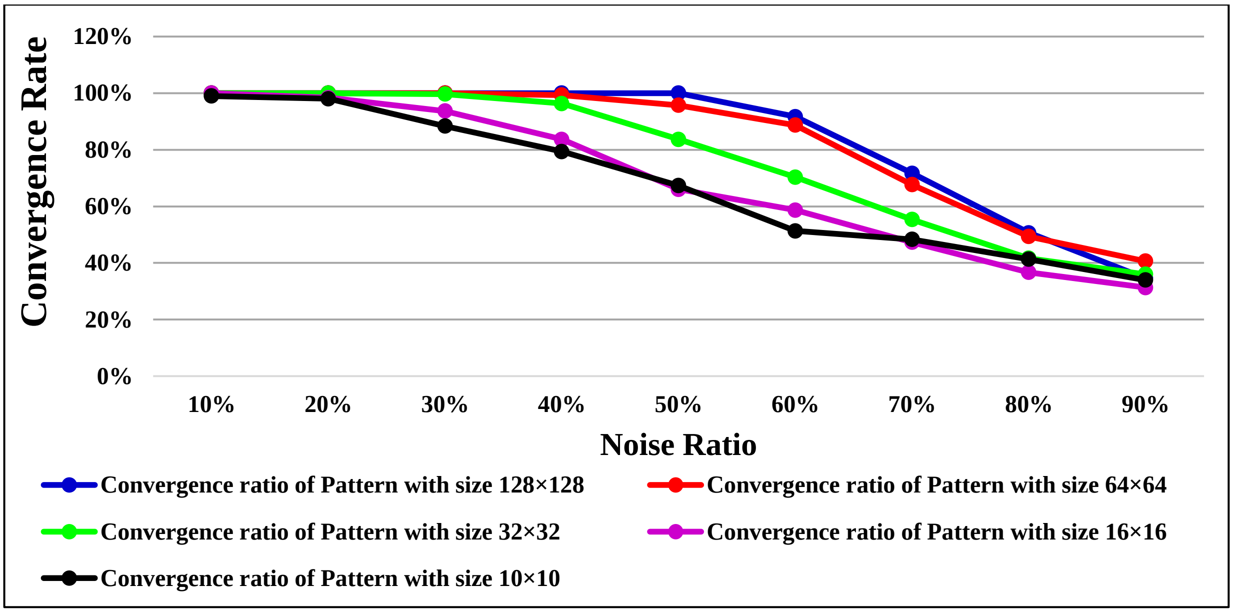

Figure 15: Diagram illustrating a comparison between the five tests of 100 different size patterns with different noise rates for the first and last quarter method net vs. convergence rates for each pattern size

6. CONCLUSION

To enhance the efficiency of the association tasks, this paper proposed MBAM as a modified BAM neural network via modified the net structure in addition to the learning and convergence process. For structure modification the size of the net becomes fixed (four neurons) with any patterns size. This size of net caused that the size of the learning weigh matrix becomes small (four matrices). For learning process modification, this net reached unlimited stored pattern, i.e. MBAM still efficient even when the number of stored patterns increases. Finally, for convergence process correlation problem has been solved, thus with MBAM net, it can store and retrieve the correlation patterns efficiently. In the other hand MBAM net still failed to recognize any stored patterns if scaling or shifting version of this stored patterns are presented to the net.

REFERENCES

[1] Christophe, T., et al., Bidirectional Associative Memory for Short-term Memory Learning. The Annual Meeting of the Cognitive Science Society COGSCI, 2014.

[2] Nong, H.T., and Bui, T. D., A fast iterative learning strategy for Bidirectional Associative Memory. In Proceedings of the Conference on Information Science and
Technology, 2012.

[3] Christophe, T., et al., Bidirectional Associative Memory and Learning of Nonlinearly Separable Tasks. ICCM International Conference on Cognitive Modelling,
2013.

[4] Mo Wei, et al., Bit Importance strategy for Bidirectional Associative Memory. IEEE
Conference on Decision and Control, and the European control conference, 2005.

[5] Osana, Y., et al., Chaotic bidirectional associative memory. In Proceedings of the
IEEE International Conference on Neural Networks, 1996. Vol. 2, p. 816-821.

[6] Yano, Y., and Osana, Y., Chaotic complex-valued bidirectional associative memory. In Proceedings of the International Joint Conference on Neural Networks,
2009.

[7] Sylvain, C., and Mounir, B., A Chaotic Bidirectional Associative Memory. In

Springer International Conference, 2011.

[8] Maria, E., et al., A New Model of BAM: Alpha-Beta Bidirectional Associative

Memories. In Lecture Notes in Computer Science, 2006.

[9] Tae-Dok, E., et al., Generalized Asymmetrical Bidirectional Associative Memory.

Applied Mathematics and Computation, 1999. Vol. 127, p.221-233.

[10] Chi Sing Leung. 1994. "Optimum Learning for Bidirectional Associative Memory

in the Sense of Capacity". IEEE Trans. on Systems, Man and Cybernetics.

[11] Melissa J. and Sylvain C., 2014. "Increasing Accuracy in a Bidirectional

Associative Memory through Expended Databases". International Conference, AGI

2014, Proceedings Springer International Publishing.

[12] Wang, Y.-F., el at., 1990. "Two Coding Strategies for Bidirectional Associative

Memory". IEEE Transactions on Neural Networks and Learning Systems, pp. 81 - 92.

[13] Kishan Mehrotra, C.K.M.a.S.R., Elements of Artificial Neural Networks.

Bradford Books, Complex Adaptive Systems, New York, USA, 1996.

[14] Zurada, J.M., Introduction to Artificial Neural System. PWS Publishing Co.,

Boston, MA, USA, 1999.

[15] Fausett, L., Fundamental of Neural Networks Architecture, Algorithms and

Application Prentice Hall, 1994.

[16] Lippmann, R.P., An Introduction to Computing with Neural Nets. IEEE Assp

Magazine 1987.

[17] V, S.G., Neural Network and its Application in Pattern Recognition. Institute of

Technology, Bombay, India, 2004.

[18] Rao, V.B., C++ Neural Networks and fuzzy logic. MT Books, IDG Books

Worldwide, 1995.