MCA: A Developed Associative Memory Using Multi-Connect Architecture
Emad I Abdul Kareem, Aman Jantan
Abstract
Although Hopfield neural network is one of the most commonly used neural network models for auto-association and optimization tasks, it has several limitations. For example, it is well known that Hopfield neural networks has limited stored patterns, local minimum problems, limited noise ratio, retrieve reverse value of pattern, and shifting and scaling problems. This research will propose multi-connect architecture (MCA) associative memory to improve the Hopfield neural network by modifying the net architecture, learning and convergence processes. This modification is to increase the performance of associative memory neural network by avoiding most of the Hopfield neural network limitations. In general, MCA is a single layer neural network uses auto-association tasks and working in two phases, that is learning and convergence phases. MCA was developed based on two principles. First, the smallest net size will be used rather than depending on the pattern size. Second, the learning process will be performed to the limited parts of the pattern only to avoid learning similar parts several times. The experiments performed show promising results when MCA shows high efficiency associative memory by avoiding most of the Hopfield net limitations. The results proved that the MCA net can learn and recognize unlimited patterns in varying size with acceptable percentage noise rate in comparison to the traditional Hopfield neural network.

Keywords: Neural Network, Associative Memory, Hopfield Neural Network, Pattern Recognition.

1. Introduction
Hopfield neural network is one of the neural networks that are used for auto-association and optimization tasks. This net has many limitations which effected its performance. These limitations are: First, the number of patterns that can be stored and accurately recalled is severely limited. Second, it is obviously desirable to reach a global minimum rather than setting down at a local minimum as it can happen while using the energy function. Third, correlation problem happens when an input pattern share many bits with another pattern. Fourth the ratio of missing and mistake data in the input patterns is limited. Fifth there is a possibility for retrieving the stored pattern with reversed values and sixth it is impossible to retrieve the stored pattern when it enters to the network with shifting or scaling.

The researches that addressed the modified Hopfield neural network to solve specific problems has been divided in to two parts, first for computer vision applications because this research modified Hopfield neural network for the same purpose. Second will be allocated for other problems that were solved by modifying this neural network, too.

This research proposed a novel associative memory named Multi-connect Architecture MCA associative Memory by modifying the net architecture, the learn process and the convergence process. Similar to the Hopfield neural network, the MCA is a single layer neural network that uses auto-association tasks.

The experiments performed show promising results when MCA shows high efficiency to recognize many noisy patterns in varying size comparing with the traditional Hopfield neural network. The proposed net avoid most of the Hopfield limitation except one which it the shifting and scaling problem, in addition to the smaller size of net and the efficient learning and convergence process.
2. Related works

Many researchers proposed different researches to modified Hopfield neural network to improve its efficiency. For example,

Abbiss et al. (1991) described the implementation of a super resolution (or spectral extrapolation) procedure on a neural network that is based on Hopfield model using different coding schemes and networks consists simply of two elements while those made up of more complex nodes are capable of representing a continuum[1]. Dotsenko et al (1991) studied a mean-field theory, which is like Hopfield neural networks but with modified interactions. The modification of interactions is achieved by a special thermally noised iterative procedure. The resulting couplings have an intermediate form between the Hebb-like learning rule and the pseudo inverse one; and Replica-symmetric free energy of the model is obtainable. Statistical properties of the model depend on three parameters: the reduced number of the stored patterns of alpha, and of iteration steps of the modified lambda procedure and the temperature. The phase diagram in the space of these parameters is obtained[6].

Due to the rugged energy function of the original Hopfield networks, the output is usually one local minimum in the energy function. Lee et al. (1991) presented an analysis on the locations of local minima in Hopfield networks and described modified network architecture to eliminate such local minima. In particular, another amplifier is introduced at the processor nodes to give correct terms. This modified Hopfield network has been applied on the construction of analogy-to-digital converters with optimal solutions [16]. Li et al. (1996) proposed a new method, called the augmented Lagrange-Hopfield (ALH) method to improve Hopfield-type neural networks in both the convergence and the solution quality in solving combinatorial optimization. It uses the augmented Lagrange method, which combines both the Lagrange and the penalty methods, to effectively solve the dilemma [17].

A fuzzy-logic based algorithm presented by cavalieri et al. (1998) to determine the coefficients automatically, thus it limits the required human intervention. Also, they define the fuzzy rules that reproduce the acquired manual experience when determining the coefficients of Hopfield network in a number of applications [5]. Zeng et al. (1999) showed that the performance of the Hopfield network can be improved by using a relaxation rate to control the relaxation process. The analysis suggested that the relaxation process had an important impact on the quality of a solution. A relaxation rate was, then, introduced to control the relaxation process in order to achieve solutions with better quality[24].

Gimenez et al. (2000) described a new procedure of implementing a recurrent neural network (RNN), based on adopting the well-known Hopfield auto-associative memory. RNN is seen as a complete graph G and the learning mechanism is also based on Hebb’s law, but with a very significant difference: the weights, which control the dynamics of the net, are obtained by colouring the graph G.

Once the training is complete, the synaptic matrix of the net will be the weight matrix of the graph. Any one of these matrices will fulfill some spatial properties, for this reason, they will be referred to as tetrahedral matrices. The geometrical properties of these tetrahedral matrices may be used for classifying the n-dimensional state-vector space in n classes. In the recall stage, a parameter vector is introduced, which is related with the capacity of the network [12]. Garcia et al. (2004) provided a new insight into the training of Hopfield associative memory neural network using the kernel theory, that was drawn from the work on kernel learning machines and their related algorithms. The kernel "trick" is used to define embedding memory patterns into (higher or infinite dimensional) memory feature vectors that allow the training of the network to be carried out in the feature space without explicitly representing it. The introduction of a kernel function in the training phase of the network improves considerably the performance of the network [13].

Hopfield has shown that models of physical systems could be used to solve some difficult computational problems. Such systems could be implemented in hardware by combining standard components such as capacitors and resistors. Bauk et al. (2005a) replaced capacitor with an inverse polarized diode with large capacity in aim to shift, in a way, Hopfield network parameters analysis from the field of the linear differential equations to the field of the exponential transcendental equations, solvable analytically by the linear approximation method based upon STFT (Special Trans Function Theory) [2]. While, Bauk et al. (2005b) modify Hopfield network in aim to allow its behavior description by the system of transcendental exponential equations solvable analytically by the Special Trans Function Theory (STFT) [3]. Wu et al. (2005) proposed a sequential algorithm using modified
Hopfield neural network based on continuous state change, and on maximal energy descent in the update rule. Finally, one property of Hopfield neural networks is the monotone minimization of energy as time proceeds [23]. Dehghan et al. (2009) applied this property to minimize the energy functions obtained by finite different techniques of Helmholtz-equation. The mathematical representation and correlation between the finite different techniques and the modified Hopfield neural networks of the Helmholtz equation are presented [7].

2.1. Associative memory

Associative memory neural network is single layer nets in which the weights are determined in such a way that the net can store a set of pattern associations. Each association is an input-output vector pair s:t if each vector t is the same as the vector s with which it is associated, then the net is called an auto-associative memory. If the t’s are different from the s’s, the net is called a hetero-associative memory [2]. Although Hopfield neural network (Section 1.3) is one of the neural network models most commonly used for auto-association and optimization tasks and most of the associative nets have been developed from this net, it has many limitations.

Figure (1) shows general block diagram of an associative memory performing an associative mapping of an input vector x into an output vector v (see equation 1).

\[ V = M[x] \]  

An associative memory can be applied in either Auto-associative or Hetero-associative application. 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 [10][8][14].

Generally, in Hetero-associative application the dimension of the input space and the output space are different, as illustrated in Figure (2). Nevertheless, the training input and target output vectors of auto-associative memory are identical, as illustrated in figure (3) [13] [9] [14].

**Figure 1.** Block diagram of an associative memory [3].

**Figure 2.** Hetero-association response [14]
One of the auto-associative memories neural networks are Hopfield Associative Memory, which will be presented in Section 2.2.

### 2.2. Hopfield Neural network

Hopfield networks are auto-associates in which node values are interactively updated based on a local computation principle: the new state of each node depends only on its net weighted input at a given time [8] [9] [17] [14]. As shown in Figure (4), any Hopfield Neural Network has N nodes; each node is connected to another node (but not to itself) and the connection strengths or weights are symmetric in that the weight from node i is to node j is the same as that of the node j to node i. That is, \( w_{ij} = w_{ji} \) and \( w_{ii} = 0 \) for all i and j. Additionally, the thresholds are all assumed to be zero. Notice that the flow of information in this type of net is not in a single direction, since it is possible for signals to flow from a node back to itself via other nodes. So, one say that there is feedback in the network or a recurrence because nodes may be used repeatedly to process information. Such a network can be contrasted with the forward feed nets that have been used exclusively so far. The input \((x_1, x_2, \ldots, x_n)\) and output \((y_1, y_2, \ldots, y_n)\) of this neural network takes on the value +1 and -1 [18] [8] [14].

![Figure 4. Single-layer n-neuron Hopfield Network Architecture [6]](image)

**Algorithm**

In this algorithm (see Figure 5), the learning phase will create a symmetric weight \( (t_i = t_j) \) for the training patterns, which will not change afterwards. If there are more than one training patterns \((s_1, s_2, \ldots, s_n)\), additional operation to all the symmetric weight of all the training patterns \((t'_1, t'_2, \ldots, t'_n)\) will be performed to create associative weight for all the training pattern. After initializing the Hopfield network with an unknown input pattern, the Converge phase will start such an operation will be repeated until there is no more change in Hopfield network output throughout successive iterations. Then, the process will be stopped and the recovered pattern is matched with the stored patterns and assigned a class [19] [18].
Figure 5. Algorithm 1: Hopfield Neural Network [19] [20] [8] [14].

Figure 6 shows an example of the behavior of a Hopfield net when used as content-addressable memory. A 120 nodes net was trained using the eight exemplars shown in (A). The pattern for the digit “3” was corrupted by randomly reversing each bit with a probability of 0.25, and then applied on the net at time zero. Outputs at time zero and after the first seven iterations are shown in (B)[19].

Figure 6. Example of the behavior of a Hopfield net.
Energy function

To prove that the previous mentioned iterated convergence process is stable one should consider the value of the energy function, which plays a vital role in the authenticity of the process as holes[21]. The existence of such a function enables one to prove that the net will converge to a stable set of activations, rather than oscillating. This function decreases as the system states change. Such a function needs to be found and watched as the network operation continues from one cycle to another. The least mean squared error is an example of such a function. Energy function usage assures a stability of the system that cannot occur without convergence. It is convenient to have one value that of the energy functions to specify the system behavior. In the Hopfield network, an energy function is a function that is bounded below and whenever the state of any unit changes. This function always decreases gradually to reach a minimum and then stops when the network is stable. As stated in equation 2 below [10][14].

\[
E = \frac{1}{2} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} v_i v_j + \sum_{k=1}^{n} \delta_k v_k \right)
\]

Where

- \( E \) = an artificial network energy.
- \( n \) = the number of elements in the vector \( V \).
- \( w_{ij} \) = weight from the output of neuron \( i \) to the input of neuron \( j \).
- \( \delta \) = the limiting value, which is equal to zero in Hopfield network.

For the observation by other researches such as [14] [10] and [21] the energy function is the result of the sum of products of the outputs of different neurons with the connection weight between them. The pairs of neuron outputs are multiplied in each term. The energy function can be calculated for every input vector created in the network. If one calculates the energy function for all the possible input vectors, one gets an energy landscape with maximums and minimums. The minimums of the energy function are patterns of a Hopfield network.

Hopfield neural network limitations

Beside Hopfield neural network various applications many researchers agree that this net is not void of any limitations, are as follows:

1. The number of patterns that can be stored and accurately recalled is severely limited. If too many patterns are stored, the net may converge to a novel spurious pattern different from all exemplar patterns [19]. In other words, the number of vector can be stored equal to the number of nodes in the network minus one [14]. For example, Hopfield Network with four nodes can store three orthogonal vectors (i.e. each vector is orthogonal to each of the other two vectors).

2. During the network converges, it is obviously desirable to reach a global minimum, rather than to set down at a local minimum, as it happens while using the energy function. Figure (7) clarifies the distinction between a local minimum and a global minimum. In this figure one may find the graph of an energy function and the two points of \( A \) and \( B \). These points show that the energy level at that point is smaller than the energy levels at any point in their vicinity, so we can say that represent points of minimum energy that in the overall or global minimum. Considering figure 7, one can notice the energy level at point \( B \) is smaller than that at the point \( A \). So, \( A \) corresponds only to a local minimum, and it is desirable to get to \( B \) and do not to stop at \( A \) itself, in the pursuit of a minimum of the energy function. If point \( C \) is reached, one is advised to go further towards \( B \) and not to stop at \( A \). Similarly, if a point near \( A \) is reached, one should proceed to \( B \) instead of settling at \( A \)[21].
3- The correlation problem happens when an input patterns share many bits (elements in the vector of the input patterns) with another pattern. A pattern is considered as unstable if it is applied at time zero and the net converges to some other pattern. Thus, all the saved patterns must be orthogonal with one another [8].

4- The ratio of missing and mistake data in the input patterns is limited [8].

5- There is a possibility for retrieving the stored pattern with reversed values. In other words, all 1s will be 0s (or -1s using bipolar values) and vice versa [14].

6- It cannot retrieve the stored pattern when it enters to the network with shifting or scaling. [14].

Next section will describe the proposed associative memory MCA.

3. Multi-connect Architecture Associative Memory (MCA)

MCA was developed from Hopfield neural network. Thus its architecture, learning process and convergence process are modified based on two principles:

1. Use smallest net size.

2. Learning process will be carried out to a finite number of pattern’s vectors to avoid learning the same vector several times.

Depending on the first principle, Section (3.1) presents the modified architecture of MCA. Furthermore, based on the second principle, modified learning and convergence process is carried out by building two new algorithms; one for the learning process (Section 3.2.1) and the other for the convergence process (Section 3.2.2).

3.1. Architecture

This research attempt to apply the concepts outlined above. The construction of a neural network capable of performing associative recall. According to the first principle, the network has only three neurons regardless of the length of the training vectors (see Figure 8). However, it just like traditional Hopfield neural network each node is connected to every other node but not to itself by at least one connection and the maximum is limited to four connections. Each connection strength or weights is symmetric so that the weight from node i to node j is the same as that from node j to node i, that is, \(w_{ij}=w_{ji}\) and \(w_{ii}=0\). This structure net leading to assumed thresholds to be zero. Technically, like traditional Hopfield net, this net has two phases (learning and convergence phase). These processes will be modified. It is noted, the information in this type of net is not in a single direction, since it is possible for signals to flow from a node back to itself via other nodes. We say that there is feedback in the network, or that it is a recurrent because nodes may be used repeatedly to process information. This is to be contrasted with the feed-forward nets that have been used exclusively so far.
3.2. Algorithms

To achieve much more clarity, this section presents two algorithms for both phases (learning and convergence) rather than combine them in one algorithm.

Learning phase (learning process)

Based on the MCA learning algorithm (see Figure (9)) the training pattern will be divided into n vectors named \( v_k \) (where \( 0 < k <= n \)). Each vector \( v \) will be in size three. Therefore, there are no more than eight possibilities for these vectors. This limited number of these vectors led to make the required weights also limited. Although, these weights have the same features of that weight resulted from Hopfield neural network learning process as a square matrices, symmetric matrices and the diagonal elements of these matrices are zero, but the size is \( 3 \times 3 \) regardless of patterns’ vector length and the fact that the number of these matrices is four.

![Figure 8](image)

**Figure 8.** Single-layer, 3-neuron Multi-Connect Architecture Associative Memory (MCA), where \( 1 <= n >= 4 \).

**Algorithm 2: Learning phase for MCA**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1    | Initialize the four connection weights matrices: 

\[
\begin{bmatrix}
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0
\end{bmatrix}, \quad 
\begin{bmatrix}
-1 & 0 & -1 \\
-1 & 0 & 1 \\
-1 & 1 & 0
\end{bmatrix}, \quad 
\begin{bmatrix}
0 & -1 & 1 \\
-1 & 0 & 1 \\
1 & 0 & -1
\end{bmatrix}, \quad 
\begin{bmatrix}
-1 & 0 & 1 \\
0 & 1 & 0 \\
1 & -1 & 0
\end{bmatrix}
\]

| Step 2 | Initialize the energy function matrix: 

\[
\begin{bmatrix}
1 & 3 & 1 & 1 \\
1 & 3 & 1 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix}
\]

| Step 3 | Repeat step 3.1 until end of training pattern \( p \): 

Step 3.1: Divide the training pattern \( p \) to \( n \) vectors \( v \) with length three. 

Step 3.2: For each vector \( v \) repeat steps 3.2.1, 3.2.2 and 3.2.3: 

Step 3.2.1: Assign the vector majority description \( smd \) as following: 

\[
\text{mld}(v) = \sum_{i=1}^{3} v_i
\]

\[
\text{smd} = \text{hard-limit} (\text{mld}(v)) \begin{cases} 
1 & \text{if } \text{mld}(v) \geq 1 \\
0 & \text{otherwise}
\end{cases}
\]

Step 3.2.2: Assign the vector weight \( sww \) as following: 

\[
\text{sww} = f(\text{Decode}(v)) \begin{cases} 
0 & \text{if } \text{mld}(v) = 0 \\
1 & \text{if } \text{mld}(v) = 1 \\
2 & \text{if } \text{mld}(v) = 2 \\
3 & \text{if } \text{mld}(v) = 3
\end{cases}
\]

Where Decode function to convert binary number to decimal number. 

Step 3.2.3: Saves \( smd \) and \( sww \) for this vector in the lookup table.

Step 4 | End.

![Figure 9](image)

**Figure 9.** Learning algorithm for MCA net.
Convergence Phase (convergence process)

Depending on MCA convergence algorithm (see Figure 10), just like MCA learning algorithm all the four weights and energy function matrices ($w_0, w_1, w_2, w_3$ and $e$) will be initialized and the unknown pattern will be dividing into $k$ vectors $v$ with size three. Using the initial weight and energy matrices, this algorithm will calculate the summation of energy function between all vectors in the unknown pattern with all weights matrices in each stored pattern and stored these matrices in the energy function matrix in the lookup table (i.e. calculate energy function between the unknown pattern and all the stored pattern weight matrices $w$ in the lookup table).

**Algorithm 3: Convergence phase for MCA.**

**Input:** unknown pattern $p$

**Output:** converge pattern $cp$

**Step 1:** Initialize the four connection weights matrices:

$$
\begin{align*}
    w_0 &= \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \\
    w_1 &= \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}, \\
    w_2 &= \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}, \\
    w_3 &= \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix}
\end{align*}
$$

**Step 2:** Initialize the energy function matrix:

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

**Step 3:** Repeat steps 3.1, 3.2 and 3.3 until end of unknown pattern $p$:

Step 3.1: Divide the unknown pattern $p$ to $n$ vectors $v$ with length three.

Step 3.2: Assign the weight row for all vector $v$ of the test unknown pattern as following:

$$
\begin{align*}
    \text{row} &= f(\text{Decode}(v)) \\
    0 & \text{ or } 7 & 0 \quad \{\text{means} \quad w_0\} \\
    1 & \text{ or } 6 & 1 \quad \{\text{means} \quad w_1\} \\
    2 & \text{ or } 5 & 2 \quad \{\text{means} \quad w_2\} \\
    3 & \text{ or } 4 & 3 \quad \{\text{means} \quad w_3\}
\end{align*}
$$

Step 3.3: Sum the energy function for all the $n$ vectors in the unknown pattern with each corresponding vector in the stored patterns:

$$
ep = \sum_{i=1}^{n} e[v_i, \text{row}_i]
$$

Step 4: Determine the stored pattern number $\text{min} ep$ with minimum energy function to converge the unknown pattern towards it:

$$
\text{Mmin}(\text{min}(ep))
$$

where $\text{min}$ is function to determine the minimum energy function in $ep$ array.

Step 5: Repeat steps 5.1, 5.2 and 5.3 to build the final converge pattern $cp$

Step 5.1: Assign $\text{temp} e_v$ for each vector in the test unknown pattern $p$

$$
\text{temp} e_v = v_i \times \text{W}_\text{temp}
$$

Step 5.2: Assign the result vector majority description $\text{rd}_{\text{nu} e_v}$ to each $\text{temp} e_v$ vector as following:

$$
\text{rd}_{\text{nu} e_v} = \text{hard-limiter}(\text{mul}(\text{temp} e_v, w_i) \geq 1) \cdot \text{mul}(\text{temp} e_v, w_i) \leq 0
$$

Step 5.3: Create the converge vector $cv$ in the converge pattern $cp$

$$
\text{cv}_i = (\text{mul}(\text{nu} e_v, \text{rd}_{\text{nu} e_v}) \times \text{temp} e_v_i
$$

Step 6: End.

**Figure 10.** Convergence algorithm for MCA.
4. Experiment and result analysis

This section presents three experiments. These experiments are applied to the traditional and MCA neural network to realize a comparison between them. This experiments highlights the efficiency of MCA by dealing with the high noise rate (missing and mistake bits), a large number of training patterns and the small size of net.

4.1. Different number of stored patterns vs. percentage convergence rate

In this experiment, for both of them (MCA and traditional Hopfield nets) we tried to learn them the maximum number of patterns, and we stopped until one/both of them failed completely to recognize the stored patterns. These patterns were alphabetical letters patterns, with size 10×10 and without noise test pattern (see figure 11 and 12). The experiment showed:

1-Due to the correlation problem, convergence rate percentage of the traditional Hopfield net was decreased when the number of stored patterns increased. Convergence failure started with four stored patterns, until the net has failed completely to recognize any stored pattern with five and six stored patterns. However, MCA net kept the same convergence rate percentage (which it is 100%) even when the number of stored patterns increased to six (see Figure (12)).

2-The size of the stored patterns in this experiment was 10×10. Thus, Traditional Hopfield net size was 100 nodes (or neurons) but MCA kept 3 nodes size because the traditional Hopfield net size depends on patterns size [8][9] where MCA net size is three neurons with any patterns size.

<table>
<thead>
<tr>
<th>Number of stored patterns</th>
<th>Traditional Hopfield net convergence</th>
<th>MCA net convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>B</td>
</tr>
</tbody>
</table>

Figure 11. Experiment one: Traditional Hopfield net and MCA learned one to six alphabetical English letters (A, B, C, D, E and F) with the converge patterns for both of them.

Figure 12. Experiment one: Diagram illustrate a comparison between traditional Hopfield net and MCA net for number of stored patterns vs. convergence rate to both of them.
4.2. Different noise percentage rates vs. percentage convergence rate.

Depending on the previous experiment, traditional Hopfield net convergence kept efficiency at 100% with input patterns have 0% noise until test number three (number of stored patterns were three (see Figure (11)). Thus, in this experiment, MCA and traditional Hopfield net learned just three patterns with size 10*10. This experiment has three tests with each one having three different random noise patterns for each pattern (A, B and C). Convergence process was implemented to both of them by test the input of this patterns with 10% to 90% random noise rate (see figure (13) and (14)). The experiment showed:

1-The efficiency of traditional Hopfield net convergence was high (100%) until noise rate became 40%. In this noise rate, the net failed to converge some of the stored patterns (see Figure (16) and (17)). Compared to MCA, the convergence efficiency was high (100%) until noise rate percentage became 50% (see Figure (15) and (16)).

2-It was observed that the traditional Hopfield net may retrieve any stored pattern (correct or incorrect pattern) with reflect color (black pixel will be white and vice versa). On the contrary this phenomenon was not detected in the MCA. (see Figure(15) and (16)).

3-It was observed that the traditional Hopfield net sometimes failed completely to converge any stored patterns, compared to with MCA which at least converged with low convergence rates. In the case of net failure, traditional Hopfield net converged with patterns which are not related to any stored patterns, where MCA net may converged incorrectly to one of the stored patterns (see figure (13) and (15)).

![Figure 13. Experiment two: Traditional Hopfield net learned just three alphabetical English letters (A,B and C) with its converge patterns.](image)

![Figure 14. Experiment two: Diagram for input patterns with different noise rate vs. traditional Hopfield net convergence rate.](image)
Figure 15. Experiment two: MCA net learned just three alphabetical English letters (A, B and C) with its converge patterns.

Figure 16. Experiment two: Diagram for input patterns with different noise rate vs. MCA net convergence rate.

4.3. Different pattern size with different noise percentage rate vs. percentage convergence rate.

This experiment proves that the proposed MCA net deals with the big size patterns more efficiently than with the small patterns size. This experiment presents 177 patterns to the MCA net with a different noise ratio as training images to know its convergence responds. The results showed the MCA convergence respond was more efficient when the size of the patterns was increased (see figure 18).

This advantage gained because of the proposed MCA net divided the patterns to vectors with size three. Logically, when one learned the net with a pattern then enter this pattern after changed all the elements in this pattern i.e. completely different pattern, the net may not recognize this pattern because it didn’t learn it before. For this reason with the small patterns and high random noise ratio, there is a possibility for the pattern’s vectors to be completely different comparing with the original vectors. Exclusively, in this case the proposed MCA associative memory can’t recognize these vectors properly leading to an incorrect convergence response. Experimentally, this case reduced whenever the patterns size increased (the relation is in inverse proportion).
MCA: A Developed Associative Memory Using Multi-Connect Architecture
Emad I. Abdul Kareem, Aman Jantan
International Journal of Intelligent Information Processing, Volume 2, Number 1, March 2011

Figure 18. Experiment three: Diagram illustrates a comparison between the four tests of 177 different size patterns with different noise rate for MCA net vs. convergence rate for each one.

Through experiments above, we find that this paper proposed MCA as a modified Hopfield 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 (three neurons) with any patterns size are. This size of net caused that the size of the learning weigh matrix becomes small and fixed (four matrices with size 3x3). For learning process modification this net reached unlimited stored pattern, i.e. MCA still efficient even when the number of stored patterns increases. Finally, for convergence process correlation problem has been solved, thus with MCA net, it can store and retrieve the correlation patterns efficiently. In addition, the net is able to avoid retrieving any stored patterns with the reflect bits (reflect bits means 1 will be -1 and vice versa). In the other hand MCA net still failed to recognize any stored patterns if scaling or shifting version of this stored patterns are presented to the net.

5. conclusions

The aim of this paper is to improve the efficiency of the associative memory. This improvement has been achieved through Hopfield neural network modification. This modification had been achieved by proposed a new network architecture of the network, in addition to adapt the both processes (learning process and convergence process) accordance with the new architecture of the proposed network. The new network resulting from this modification called the MAC, which bypassed most of the limitations that suffered by Hopfield network in particular, and associative memory in general.

6. References