Multidirectional associative memory with self-connections

Shogo Furusawa and Masaki Kobayashi

Abstract: Multidirectional associative memory (MAM) enables associations among many items. Its architecture is very simple and provides high parallelism. However, recall results of MAMs are inconsistent for contradictory inputs. The recall results depend on the order of update. If a given input includes an incorrect pattern, we expect that the incorrect pattern will be corrected, and the recalled pattern is determined by the majority. In this work, we propose MAMs with self-connections and a new learning algorithm. These MAMs provide recall results independent of the order of update. Furthermore, they maintain the advantages of MAMs, such as their simple architecture, high parallelism and stability.

Key Words: multidirectional associative memory, self-connection, parallel processing, majority decision

1. Introduction

Three decades ago, Hopfield proposed the Hopfield associative memory (HAM) as an associative memory model that functions through neural networks [4, 5]. A HAM can store some training patterns. Given an input pattern, a HAM recalls the stored training pattern closest to the given pattern. A few years later, Kosko proposed another associative memory model, the bidirectional associative memory (BAM) [11, 12]. A BAM is a mutual associative memory while a HAM is an auto-associative memory. A BAM consists of two layers and can store pairs of items; thus, it enables mutual associations between two items. Hagiwara extended BAMs to multidirectional associative memories (MAMs) [1]. A MAM consists of multiple layers and can store combinations of more than three items; it enables associations among many items. Moreover, it has a very simple architecture. Some advanced models of MAMs, such as complex-valued MAMs, MAMs with context parts, and MAMs with hidden layers, have been proposed [8–10, 13, 14]. Hattori and Hagiwara applied MAMs to knowledge processing systems [2]. Kitabata et al. [6] and Kitabata and Takefuji [7] applied MAMs to face recognition.

Each neuron of MAM is included in a layer and is connected to all neurons in the other layers, but not to any neurons in the same layer. All of the neurons in any one layer can be independently updated. Figure 1 shows a five-layered MAM.

MAMs have the following advantages:

1. The architecture of the MAM is very simple.
2. A MAM enables associations among many items.

3. MAMs can update many neurons simultaneously, because all of the neurons in any one layer are independent; MAMs have high parallelism.

4. A MAM surely achieves a stable state. This fact is proven by using the energy function.

On the other hand, MAMs have the following disadvantages:

1. The information of the updating layers is not available, because each neuron can receive information only from the other layers. The information of the first updating layer in particular is meaningless.

2. Recall results are inconsistent, because they depend on the order of update.

3. Layers with more neurons have more influence on recall result than those having fewer neurons.

We describe the behavior of the MAM and these disadvantages in detail. Let us consider a MAM consisting of three layers, Layer-1, Layer-2 and Layer-3. Let \( N_i \) be the number of neurons in Layer-\( i \). Now let us consider two training patterns, \((A_1, A_2, A_3)\) and \((B_1, B_2, B_3)\), where the suffix indicates the layer. If \((A'_1, A'_2, A'_3)\), where \(A'_i\) is \(A_i\) with noise, is given and Layer-1, Layer-2 and Layer-3 are updated in this order, the recall process is as follows:

\[
(A'_1, A'_2, A'_3) \rightarrow (A_1, A'_2, A'_3) \rightarrow (A_1, A_2, A'_3) \rightarrow (A_1, A_2, A_3).
\]

Thus, a MAM can remove the noise and recall the training pattern \((A_1, A_2, A_3)\) in the recall process. For any pattern \(A'_1\), which is updated first, the same recall result is obtained. Therefore, the information of Layer-1 is meaningless. Suppose that a contradictory input \((A_1, A_2, B_3)\) is given. In such a case, we would conclude that the information of Layer-3 is incorrect and we would expect that \((A_1, A_2, A_3)\) would be obtained by the majority decision. When \(N_3\) is larger than \(N_2\) and Layer-1 is updated first, then the state of Layer-1 changes to \(B_1\), because the weighted input from Layer-3 is stronger than that from Layer-2. Subsequently, if Layer-2 is updated, \((B_1, B_2, B_3)\) is obtained. Consequently, the recall process is as follows:

\[
(A_1, A_2, B_3) \rightarrow (B_1, A_2, B_3) \rightarrow (B_1, B_2, B_3).
\]
This result is different from the one we expected. When Layer-3 is updated first, \((A_1, A_2, A_3)\) is obtained. Therefore, the recall result depends on the order of update. To decide by majority, it is necessary that information on the updating layer be available and that all layers have an even influence on the recall result.

2. Multidirectional associative memory

2.1 Architecture

A MAM consists of multiple layers with full connections among the layers. However, there are no connections within any given layer. Figure 2 shows the architecture of a four-layered MAM. Let \(L\) and \(N_i\) be the number of layers and the number of neurons in Layer-\(i\), respectively. The connection weight matrix \(W_{ji}\) from Layer-\(i\) to Layer-\(j\) is an \(N_j \times N_i\) matrix. Connection weight matrices are required to satisfy a symmetric condition \(W_{ji} = W_{ij}^T\), where the superscript \(T\) means the transpose.

We denote the state vector of Layer-\(i\) by an \(N_i\)-dimensional vector \(X_i\) whose elements are \(\pm 1\). The weighted sum input \(S_j\) to Layer-\(j\) is \(\sum_{i \neq j} W_{ji} X_i\). The new state of \(X_j\) maximizes \(X_j^T S_j\).

For simplicity, we suppose \(L = 3\). From a given initial state \((X_1, X_2, X_3)\), all layers are iteratively updated in a certain order until the MAM becomes stable. If the order of update is Layer-1, Layer-2 and Layer-3, the recall process is as follows:

\[
(X_1, X_2, X_3) \rightarrow (X'_1, X_2, X_3) \rightarrow (X'_1, X'_2, X_3) \rightarrow (X''_1, X'_2, X_3) \rightarrow \cdots \rightarrow (\hat{X}_1, \hat{X}_2, \hat{X}_3).
\]

The final state \((\hat{X}_1, \hat{X}_2, \hat{X}_3)\) is a stable state.

To prove that a MAM always achieves a stable state, we introduce the energy function \(E\) of MAM, which is defined as follows:

\[
E = -\frac{1}{2} \sum_i \sum_{j \neq i} X_i^T W_{ij} X_j. \tag{1}
\]

Suppose that the state of Layer-\(k\) changes from \(X_k\) to \(X'_k\). Then the difference \(\Delta E\) of the energy is as follows:

\[
\Delta E = -\frac{1}{2} \sum_{j \neq k} X'_k^T W_{kj} X_j - \frac{1}{2} \sum_{i \neq k} X_i^T W_{ik} X'_k
+ \frac{1}{2} \sum_{j \neq k} X_k^T W_{kj} X_j + \frac{1}{2} \sum_{i \neq k} X_i^T W_{ik} X_k \tag{2}
\]

\[
= -X'_k^T \sum_{j \neq k} W_{kj} X_j + X_k^T \sum_{j \neq k} W_{kj} X_j \tag{3}
\]

\[
= -X'_k^T S_k + X_k^T S_k. \tag{4}
\]
Therefore, $\Delta E$ is not positive, because $X'_i$ maximizes $X'^T_i S_k$. We have thus proved that the energy $E$ always decreases. Thus, we find that the MAM surely achieves a stable state.

### 2.2 Learning algorithm

We denote the pattern vector of Layer-$i$ of the $p$th training pattern by $A^p_i$. Then, the connection weight matrices are given as follows:

$$W_{ji} = \sum_p A^p_j A^p_i^T. \tag{5}$$

When the $q$th training pattern is given, the weighted sum input $S_j$ to Layer-$j$ is as follows:

$$S_j = \sum_{i\neq j} W_{ji} A^q_i \tag{6}$$

$$= \sum_{i\neq j} \sum_p A^p_j A^p_i^T A^q_i \tag{7}$$

$$= \sum_{i\neq j} N_i A^q_i + \sum_{i\neq j} \sum_{p \neq q} A^p_j A^p_i^T A^q_i. \tag{8}$$

The first term helps recall the $q$th training pattern. The second term is the crosstalk term.

For simplicity, let us consider a three-layered MAM and two training patterns ($A_1, A_2, A_3$) and ($B_1, B_2, B_3$). Suppose $N_3$ is larger than $N_2$. When a training pattern with noise ($A'_1, A'_2, A'_3$) is given, we can obtain the original pattern ($A_1, A_2, A_3$) through the recall process. Suppose that a contradictory pattern ($A_1, A_2, A_3$) is given. It is natural that we expect to obtain the training pattern ($A_1, A_2, A_3$) by majority decision. When Layer-3 is updated first, we obtain our expected pattern ($A_1, A_2, A_3$), however, when Layer-1 is updated first, the weighted inputs from Layer-2 and Layer-3 are $N_2 A_1$ and $N_3 B_1$, respectively, where the crosstalk terms are ignored.

### 3. Multidirectional associative memory with self-connections

#### 3.1 Architecture

To make a decision by majority, we introduce self-connections and a learning algorithm for our proposed MAMs. In this subsection, we present the architecture. The connection weight matrix among the different layers is identical to that of a conventional MAM. We add self-connections $W_{jj}$ by $a_j I_{N_j}$, where $a_j$ and $I_{N_j}$ are a positive number and an identity matrix with size $N_j$, respectively. Figure 3 shows the architecture of our proposed MAM. The weighted sum input $S_j$ to Layer-$j$ is as follows:

$$S_j = \sum_i W_{ji} X_i. \tag{9}$$

Note that all neurons in the same layer can be independently updated, because the connection weights among different neurons in the same layer are zero. The new state of $X_j$ maximizes $X'^T_j S_j$.

We define the energy function $E$ of our proposed MAM as follows:

$$E = -\frac{1}{2} \sum_{i,j} X'^T_i W_{ij} X_j. \tag{10}$$

Suppose that the state of Layer-$k$ changes from $X_k$ to $X'_k$. Then the difference $\Delta E$ of the energy is as follows:

$$\Delta E = -\frac{1}{2} \sum_{j \neq k} X'^T_k W_{kj} X_j - \frac{1}{2} \sum_{i \neq k} X'^T_i W_{ik} X'_k - \frac{1}{2} X'^T_k X'_k.$$
The sum of the first and second terms $-X_k' S_k + X_k S_k$ is not positive, because $X_k'$ maximizes $X_k T S_k$. Furthermore, the third term $-\frac{1}{2}a_k \|X_k' - X_k\|^2$ is not positive, because $a_k$ is positive. Therefore, $\Delta E$ is not positive. We have proved that the energy $E$ always decreases. Thus, we find that the MAM surely achieves a stable state.

### 3.2 Learning algorithm

First, we give a learning algorithm for the connection weight matrices among different layers in order to make weighted sum inputs from all layers have the same influences. The new learning algorithm is given as follows:

$$W_{ji} = \frac{1}{N_i N_j} \sum_p A^p_j A^p_i. \quad (16)$$

When the $q$th training pattern is given, the weighted input $S_{ji}$ from Layer-$i$ to Layer-$j$ is as follows:

$$S_{ji} = W_{ji} A^q_i = \frac{1}{N_i N_j} \sum_p A^p_j A^p_i A^q_i. \quad (17)$$

The main term $\frac{1}{N_j} A^q_j$ is independent of Layer-$i$. 
Next, we give the learning algorithm for self-connections as \( W_{jj} = \frac{1}{N_j} I_{N_j} \). When the \( q \)th training pattern is given, the weighted input \( S_{jj} \) from Layer-\( j \) to Layer-\( j \) is as follows:

\[
S_{jj} = \frac{1}{N_j} A_j^q.
\]  

(20)

Consequently, all layers have the same influence on an updating layer and the MAM can recall by majority decision.

3.3 Recall process

We have introduced self-connections and a new learning algorithm in order to realize majority decision. Once all layers are updated, information is distributed to all layers based on majority decision and self-connections are not necessary. We consider the following update modes.

Mode I Self-connections are always available.

Mode II First, all layers are updated sequentially using self-connections; each neuron uses its own self-connection only once. Next, all layers are updated sequentially again without self-connections until the MAM achieves a stable state.

4. Computer simulation

4.1 Recall process

We performed computer simulations for the recall processes of a conventional MAM and our proposed MAM. We used binary images to visualize the recall processes. Figure 4 shows the training patterns. The black and white pixels mean 1 and \(-1\), respectively. The numbers of neurons are \( N_1 = 25 \), \( N_2 = 49 \) and \( N_3 = 400 \). The number of training patterns is \( P = 3 \).

Layer-1  Layer-2  Layer-3

<table>
<thead>
<tr>
<th>Pattern1</th>
<th>Pattern2</th>
<th>Pattern3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>B1</td>
<td>C1</td>
</tr>
<tr>
<td>A2</td>
<td>B2</td>
<td>C2</td>
</tr>
<tr>
<td>A3</td>
<td>B3</td>
<td>C3</td>
</tr>
</tbody>
</table>

Fig. 4. Training patterns.
We input a contradictory pattern \((A_1, A_2, B_3)\) into the conventional MAM. Our expected recall result is \((A_1, A_2, A_3)\) by majority decision. Figure 5 shows the recall process by update in ascending order, i.e., Layer-1, Layer-2, Layer-3. When Layer-1 was updated, the weighted inputs from Layer-2 and Layer-3 were \(N_2A_1\) and \(N_3B_1\), respectively, where the crosstalk terms were ignored. Since \(N_2 < N_3\), pattern \(B_1\) is recalled. When Layer-2 was updated, the weighted sum input was \((N_1 + N_3)B_2\) and pattern \(B_2\) was recalled. The conventional MAM achieved a stable state with a recall result of \((B_1, B_2, B_3)\), which was not our expected result. Figure 6 shows the recall process by update in
Recall process of a MAM with self-connections in ascending order.

1) The initial state is \((A_1, A_2, B_3)\).  2) After updating Layer-1, the state is \((A_1, A_2, B_3)\).  3) After updating Layer-2, the state is \((A_1, A_2, B_3)\).  4) After updating Layer-3, the state is \((A_1, A_2, A_3)\) and stable.

descending order, i.e., Layer-3, Layer-2, Layer-1. When Layer-3 was updated, the conventional MAM achieved a stable state and recalled the pattern \((A_1, A_2, A_3)\), which is our expected result. The recall results of the conventional MAM depend on update order and are inconsistent.

We next input the same pattern \((A_1, A_2, B_3)\) into our proposed MAM. Figure 7 shows the recall process by update in ascending order. When Layer-1 was updated, the weighted input from Layer-2 and self-feedback from Layer-1 were \(\frac{1}{N_1} A_1\), and the weighted input from Layer-3 was \(\frac{1}{N_1} B_1\). The weighted sum input was \(\frac{1}{N_1} (2A_1 + B_1)\). Then Layer-1 recalled \(A_1\). When Layer-2 was updated, Layer-2 recalled \(A_2\) in the same way. Finally, Layer-3 recalled \(A_3\) and our proposed MAM achieved a stable state \((A_1, A_2, A_3)\). The recall result was our expected result. Figure 6 also shows the recall process of our proposed MAM by update in descending order. When Layer-3 was updated, the weighted sum input was \(\frac{1}{N_3} (2A_3 + B_3)\). The proposed MAM achieved a stable state with a recall result of \((A_1, A_2, A_3)\). Consequently, we obtained our expected result with both the ascending and descending orders of recall. In these computer simulations, modes I and II produced the same recall results, because our proposed MAM achieved a stable state once all layers were updated.
4.2 Noise robustness

Noise robustness is one of the most important properties of associative memories. We performed computer simulations for noise robustness of the MAMs under consideration.

To add the noise, we randomly selected $p\%$ neurons from all the neurons of a given pattern and reversed their states. Rate $p$ is referred to as the noise rate. The simulations were carried out as follows:

1. We added the noise of noise rate $p$ to a given training pattern and input it into the MAM as the initial state.
2. The MAM achieved a stable state.
3. If the recalled pattern was the original training pattern, we regarded the trial as successful. Otherwise, we regarded the trial as failed.

First we used the training patterns shown in Fig. 4. Figure 8 shows the example of Pattern 1 with a noise rate of $p = 10$. For each $p$ and training pattern, we carried out 500 trials, yielding a total of 1500 trials for each $p$. Figures 9 and 10 show the simulation results in ascending and descending orders of recall, respectively. The horizontal and vertical axes show the noise rate and success rate, respectively. The noise robustness of our proposed MAM with either mode is less than that of the conventional MAM. Our proposed MAM with Mode II showed greater noise robustness than the proposed MAM with Mode I.

Next, we performed computer simulations for patterns generated randomly under the conditions $L = 4$, $N_1 = 50$, $N_2 = 100$, $N_3 = 150$, $N_4 = 200$ and $P = 5$. We generated 100 training pattern sets and performed 100 trials for each $p$ and training pattern, yielding a total of 50000 trials for

\[\text{Fig. 8. Example of Pattern 1 with a noise rate of } p = 10.\]

\[\text{Fig. 9. Noise robustness in ascending order of recall.}\]
Fig. 10. Noise robustness in descending order of recall.

Fig. 11. Noise robustness for random patterns in ascending order of recall.

Each $p$. Figures 11 and 12 show the simulation results in ascending and descending orders of recall, respectively. The conventional MAM achieved the best noise robustness, but the differences were small.

The self-connections tend to stabilize any states. Some patterns other than training patterns are stabilized and more spurious states are generated. Thus, our proposed MAM has less noise robustness than the conventional MAM. The noise robustness of the conventional MAM in ascending order of recall was a little better than that in descending order. In conventional MAMs, the information of the first updated layer is not available. The number of neurons in the first updated layer in ascending
order was lower than that in descending order and the conventional MAM in ascending order of recall was able to use more information than that in descending order. The first simulation showed a greater difference than the second one, because the ratio of neurons in the layer with the most neurons was larger.

4.3 Storage Capacity
Storage capacity is one of the most important properties of associative memories. We performed computer simulations to test the storage capacity of the MAMs as follows:

![Success Rate vs Noise Rate](image1)

**Fig. 12.** Noise robustness for random patterns in descending order of recall.

![Success Rate vs Number of Training Patterns](image2)

**Fig. 13.** Storage capacity with $N_1 = 25$, $N_2 = 49$, $N_3 = 400$ and $L = 3$. 

232
1. The training patterns were generated randomly and learned.

2. If all training patterns were stable, we concluded that they were successfully stored.

In the case of our proposed MAM with Mode II, the simulations were performed without self-connections. For each number of training patterns $P$, we carried out 500 trials.

First, we performed computer simulations for a MAM with $N_1 = 25, N_2 = 49, N_3 = 400$ and $L = 3$. Figure 13 shows the simulation result. The horizontal and vertical axes show the number of training patterns and success rate, respectively. Next, we performed computer simulations under the condition $N_1 = 50, N_2 = 100, N_3 = 150, N_4 = 200$ and $L = 4$. Figure 14 shows the simulation result. We obtained similar results in both cases.

The self-connections increase the main term of the weighted sum input and help tolerate more crosstalk term. Thus, our proposed MAM in Mode I showed greater storage capacity than the conventional MAM in both cases. Our proposed MAM in Mode II had less storage capacity than the conventional MAM.

5. Conclusion

A MAM enables associations among many items and has several advantages, such as its simple architecture, high parallelism and stability. On the other hand, recall results of MAMs are inconsistent for contradictory inputs. It is natural that we expect the majority decision for a contradictory input. However, a conventional MAM provides different recall results depending on the order of update.

In the present work, we proposed a MAM with self-connections and a new learning algorithm. Our proposed MAM maintains the advantages of MAMs. Through computer simulation, we found that our proposed MAM with Mode I or II realized the majority decision. Moreover, although our proposed models have less noise robustness than conventional MAMs, our proposed model with Mode I, was found to have greater storage capacity than conventional MAMs.

BAM cannot recall by the majority decision, because it recalls using two given patterns. We can consider the problems of majority decision only for MAMs with more than two layers, and the present discussion is applicable to any MAMs. In the case of three-layered MAMs, the self-connections are particularly important because a third set of information is not available without those self-
Our proposed learning algorithm is based on the Hebbian learning rule. Therefore, a MAM can store only a few training patterns. It is necessary to provide advanced learning algorithms in future research [3, 8].

Acknowledgments
The authors would like to thank E. Yanase of the University of Yamanashi for designing the training patterns.

References