

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Improved Kohonen Feature Map Probabilistic Associative Memory Based on Weights Distribution

Shingo Noguchi and Osana Yuko

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/51581>

1. Introduction

Recently, neural networks are drawing much attention as a method to realize flexible information processing. Neural networks consider neuron groups of the brain in the creature, and imitate these neurons technologically. Neural networks have some features, especially one of the important features is that the networks can learn to acquire the ability of information processing.

In the field of neural network, many models have been proposed such as the Back Propagation algorithm [1], the Kohonen Feature Map (KFM) [2], the Hopfield network [3], and the Bidirectional Associative Memory [4]. In these models, the learning process and the recall process are divided, and therefore they need all information to learn in advance.

However, in the real world, it is very difficult to get all information to learn in advance, so we need the model whose learning process and recall process are not divided. As such model, Grossberg and Carpenter proposed the ART (Adaptive Resonance Theory) [5]. However, the ART is based on the local representation, and therefore it is not robust for damaged neurons in the Map Layer. While in the field of associative memories, some models have been proposed [6 - 8]. Since these models are based on the distributed representation, they have the robustness for damaged neurons. However, their storage capacities are small because their learning algorithm is based on the Hebbian learning.

On the other hand, the Kohonen Feature Map (KFM) associative memory [9] has been proposed. Although the KFM associative memory is based on the local representation as similar as the ART[5], it can learn new patterns successively [10], and its storage capacity is larger than that of models in refs.[6 - 8]. It can deal with auto and hetero associations and the asso-

ciations for plural sequential patterns including common terms [11, 12]. Moreover, the KFM associative memory with area representation [13] has been proposed. In the model, the area representation [14] was introduced to the KFM associative memory, and it has robustness for damaged neurons. However, it can not deal with one-to-many associations, and associations of analog patterns. As the model which can deal with analog patterns and one-to-many associations, the Kohonen Feature Map Associative Memory with Refractoriness based on Area Representation [15] has been proposed. In the model, one-to-many associations are realized by refractoriness of neurons. Moreover, by improvement of the calculation of the internal states of the neurons in the Map Layer, it has enough robustness for damaged neurons when analog patterns are memorized. However, all these models can not realize probabilistic association for the training set including one-to-many relations.

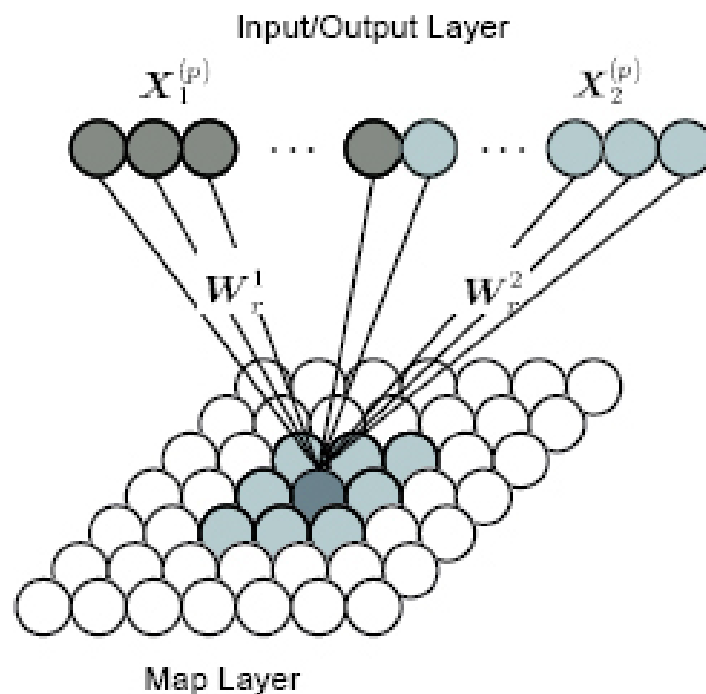


Figure 1. Structure of conventional KFMPAM-WD.

As the model which can realize probabilistic association for the training set including one-to-many relations, the Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution (KFMPAM-WD) [16] has been proposed. However, in this model, the weights are updated only in the area corresponding to the input pattern, so the learning considering the neighborhood is not carried out.

In this paper, we propose an Improved Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution (IKFMPAM-WD). This model is based on the conventional Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution [16]. The proposed model can realize probabilistic association for the training set including one-to-many relations. Moreover, this model has enough robustness for noisy input and damaged neurons. And, the learning considering the neighborhood can be realized.

2. KFM Probabilistic Associative Memory based on Weights Distribution

Here, we explain the conventional Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution (KFMPAM-WD)(16).

2.1. Structure

Figure 1 shows the structure of the conventional KFMPAM-WD. As shown in Fig. 1, this model has two layers; (1) Input/Output Layer and (2) Map Layer, and the Input/Output Layer is divided into some parts.

2.2. Learning process

In the learning algorithm of the conventional KFMPAM-WD, the connection weights are learned as follows:

1. The initial values of weights are chosen randomly.
2. The Euclidian distance between the learning vector $X^{(p)}$ and the connection weights vector W_i , $d(X^{(p)}, W_i)$ is calculated.
3. If $d(X^{(p)}, W_i) \leq \theta$ is satisfied for all neurons, the input pattern $X^{(p)}$ is regarded as an unknown pattern. If the input pattern is regarded as a known pattern, go to (8).
4. The neuron which is the center of the learning area r is determined as follows:

$$r = \underset{\substack{i : D_{iz} + D_{zj} < d_{iz} \\ (\text{for } \forall z \in F)}}{\operatorname{argmin}} d(X^{(p)}, W_i) \quad (1)$$

where F is the set of the neurons whose connection weights are fixed. d_{iz} is the distance between the neuron i and the neuron z whose connection weights are fixed. In Eq.(1), D_{ij} is the radius of the ellipse area whose center is the neuron i for the direction to the neuron j , and is given by

$$D_{ij} = \begin{cases} a_i, & (d_{ij}^y = 0) \\ b_i, & (d_{ij}^x = 0) \\ \sqrt{\frac{a_i^2 b_i^2}{b_i^2 + m_{ij}^2 a_i^2} (m_{ij}^2 + 1)}, & (\text{otherwise}) \end{cases} \quad (2)$$

where a_i is the long radius of the ellipse area whose center is the neuron i and b_i is the short radius of the ellipse area whose center is the neuron i . In the KFMPAM-WD, a_i and b_i can be set for each training pattern. m_{ij} is the slope of the line through the neurons i and j . In Eq.(1), the neuron whose Euclidian distance between its connection weights and the learning vector is minimum in the neurons which can be take areas without

overlaps to the areas corresponding to the patterns which are already trained. In Eq.(1), a_i and b_i are used as the size of the area for the learning vector.

5. If $d(X^{(p)}, W_r) > \theta^t$ is satisfied, the connection weights of the neurons in the ellipse whose center is the neuron r are updated as follows:

$$W_i(t+1) = \begin{cases} W_i(t) + \alpha(t)(X^{(p)} - W_i(t)), & (d_{ri} \leq D_{ri}) \\ W_i(t), & (\text{otherwise}) \end{cases} \quad (3)$$

where $\alpha(t)$ is the learning rate and is given by

$$\alpha(t) = \frac{-\alpha_0(t-T)}{T}. \quad (4)$$

Here, α_0 is the initial value of $\alpha(t)$ and T is the upper limit of the learning iterations.

6. (5) is iterated until $d(X^{(p)}, W_r) \leq \theta^t$ is satisfied.
7. The connection weights of the neuron r W_r are fixed.
8. (2)~(7) are iterated when a new pattern set is given.

2.3. Recall process

In the recall process of the KFMPAM-WD, when the pattern X is given to the Input/Output Layer, the output of the neuron i in the Map Layer, x_i^{map} is calculated by

$$x_i^{map} = \begin{cases} 1, & (i=r) \\ 0, & (\text{otherwise}) \end{cases} \quad (5)$$

where r is selected randomly from the neurons which satisfy

$$\frac{1}{N^{in}} \sum_{k \in C} g(X_k - W_{ik}) > \theta^{map} \quad (6)$$

where θ^{map} is the threshold of the neuron in the Map Layer, and $g(\cdot)$ is given by

$$g(b) = \begin{cases} 1, & (|b| < \theta^d) \\ 0, & (\text{otherwise}). \end{cases} \quad (7)$$

In the KFMPAM-WD, one of the neurons whose connection weights are similar to the input pattern are selected randomly as the winner neuron. So, the probabilistic association can be realized based on the weights distribution.

When the binary pattern X is given to the Input/Output Layer, the output of the neuron k in the Input/Output Layer x_k^{io} is given by

$$x_k^{io} = \begin{cases} 1, & (W_{rk} \geq \theta_b^{no}) \\ 0, & (\text{otherwise}) \end{cases} \quad (8)$$

where θ_b^{io} is the threshold of the neurons in the Input/Output Layer.

When the analog pattern X is given to the Input/Output Layer, the output of the neuron k in the Input/Output Layer x_k^{io} is given by

$$x_k^{io} = W_{rk}. \quad (9)$$

3. Improved KFM Probabilistic Associative Memory based on Weights Distribution

Here, we explain the proposed Improved Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution (IKFMPAM-WD). The proposed model is based on the conventional Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution (KFMPAM-WD) [16] described in 2.

3.1. Structure

Figure 2 shows the structure of the proposed IKFMPAM-WD. As shown in Fig. 2, the proposed model has two layers; (1) Input/Output Layer and (2) Map Layer, and the Input/Output Layer is divided into some parts as similar as the conventional KFMPAM-WD.

3.2. Learning process

In the learning algorithm of the proposed IKFMPAM-WD, the connection weights are learned as follows:

1. The initial values of weights are chosen randomly.
2. The Euclidian distance between the learning vector $X^{(p)}$ and the connection weights vector W_i , $d(X^{(p)}, W_i)$, is calculated.
3. If $d(X^{(p)}, W_i) \leq \theta^t$ is satisfied for all neurons, the input pattern $X^{(p)}$ is regarded as an unknown pattern. If the input pattern is regarded as a known pattern, go to (8).
4. The neuron which is the center of the learning area r is determined by Eq.(1). In Eq.(1), the neuron whose Euclid distance between its connection weights and the learning vector is minimum in the neurons which can be take areas without overlaps to the areas

corresponding to the patterns which are already trained. In Eq.(1), a_i and b_i are used as the size of the area for the learning vector.

5. If $d(X^{(p)}, W_r) \leq \theta^t$ is satisfied, the connection weights of the neurons in the ellipse whose center is the neuron r are updated as follows:

$$W_i(t+1) = \begin{cases} X^{(p)}, & (\theta_1^{learn} \leq H(\bar{d}_{ri})) \\ W_i(t) + H(\bar{d}_{ri})(X^{(p)} - W_i(t)), & (\theta_2^{learn} \leq H(\bar{d}_{ri}) < \theta_1^{learn} \\ & \text{and } H(\bar{d}_{i^*i}) < \theta_1^{learn}) \\ W_i(t), & (\text{otherwise}) \end{cases} \quad (10)$$

where θ_1^{learn} are thresholds. $H(\bar{d}_{ri})$ and $H(\bar{d}_{i^*i})$ are given by Eq.(11) and these are semi-fixed function. Especially, $H(\bar{d}_{ri})$ behaves as the neighborhood function. Here, i^* shows the nearest weight-fixed neuron from the neuron i .

$$H(\bar{d}_{ij}) = \frac{1}{1 + \exp\left(\frac{\bar{d}_{ij} - D}{\varepsilon}\right)} \quad (11)$$

where \bar{d}_{ij} shows the normalized radius of the ellipse area whose center is the neuron i for the direction to the neuron j , and is given by

$$\bar{d}_{ij} = \frac{d_{ij}}{D_{ij}}. \quad (12)$$

In Eq.(11), D ($1/D$) is the constant to decide the neighborhood area size and ε is the steepness parameter. If there is no weight-fixed neuron,

$$H(\bar{d}_{i^*i}) = 0 \quad (13)$$

is used.

6. (5) is iterated until $d(X^{(p)}, W_r) \leq \theta^t$ is satisfied.
7. The connection weights of the neuron r W_r are fixed.
8. (2)~(7) are iterated when a new pattern set is given.

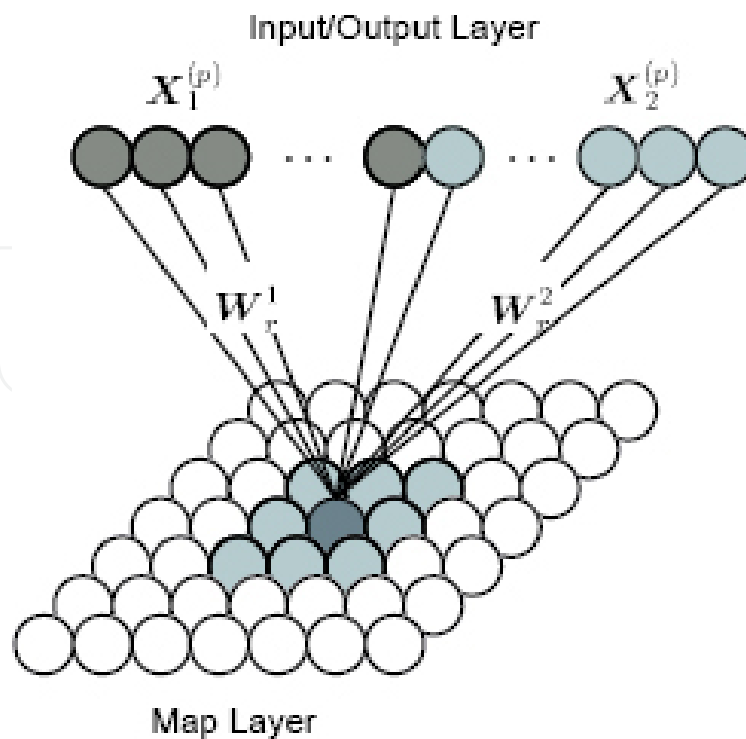


Figure 2. Structure of proposed IKFMPAM-WD.

3.3. Recall process

The recall process of the proposed IKFMPAM-WD is same as that of the conventional KFMPAM-WD described in 2.3.

4. Computer experiment results

Here, we show the computer experiment results to demonstrate the effectiveness of the proposed IKFMPAM-WD.

4.1. Experimental conditions

Table 1 shows the experimental conditions used in the experiments of 4.2 ~ 4.6.

4.2. Association results

4.2.1. Binary patterns

In this experiment, the binary patterns including one-to-many relations shown in Fig. 3 were memorized in the network composed of 800 neurons in the Input/Output Layer and 400 neurons in the Map Layer. Figure 4 shows a part of the association result when “crow” was given to the Input/Output Layer. As shown in Fig. 4, when “crow” was given to the net-

work, “mouse” ($t=1$), “monkey” ($t=2$) and “lion” ($t=4$) were recalled. Figure 5 shows a part of the association result when “duck” was given to the Input/Output Layer. In this case, “dog” ($t=251$), “cat” ($t=252$) and “penguin” ($t=255$) were recalled. From these results, we can confirmed that the proposed model can recall binary patterns including one-to-many relations.

Parameters for Learning	
Threshold for Learning	$\theta_t^{learn} 10^{-4}$
Neighborhood Area Size	$D \quad 3$
Steepness Parameter in Neighborhood Function	$\epsilon \quad 0.91$
Threshold of Neighborhood Function (1)	$\theta_1^{learn} 0.9$
Threshold of Neighborhood Function (2)	$\theta_2^{learn} 0.1$
Parameters for Recall (Common)	
Threshold of Neurons in Map Layer	$\theta^{map} 0.75$
Threshold of Difference between Weight Vector and Input Vector	$\theta_d \quad 0.004$
Parameter for Recall (Binary)	
Threshold of Neurons in Input/Output Layer	$\theta_b^{in} \quad 0.5$

Table 1. Experimental Conditions.

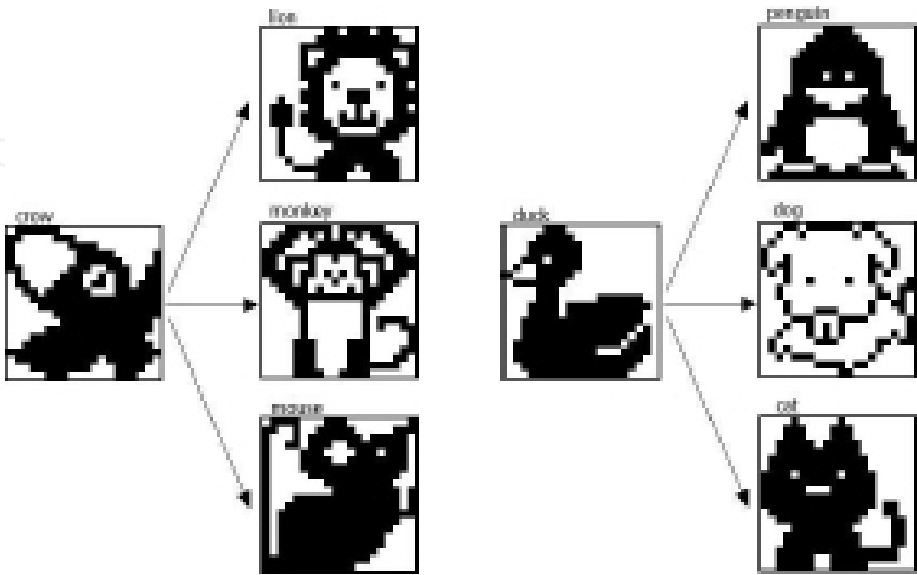


Figure 3. Training Patterns including One-to-Many Relations (Binary Pattern).

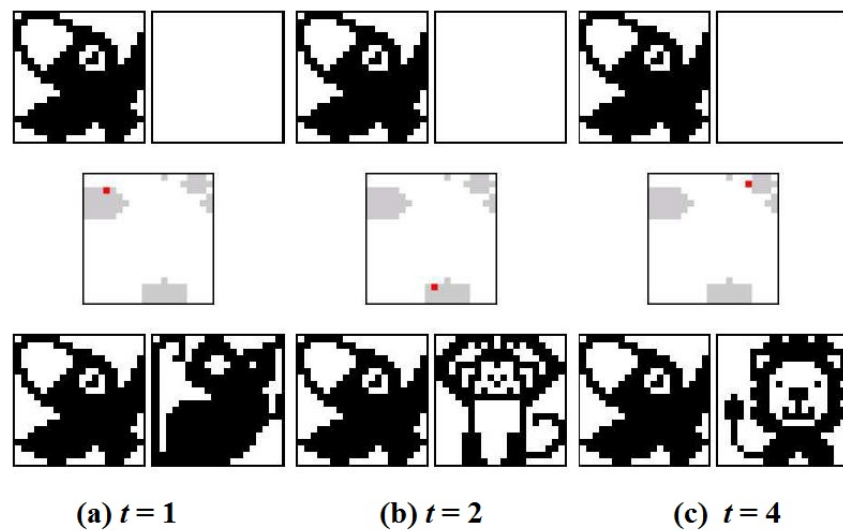


Figure 4. One-to-Many Associations for Binary Patterns (When "crow" was Given).

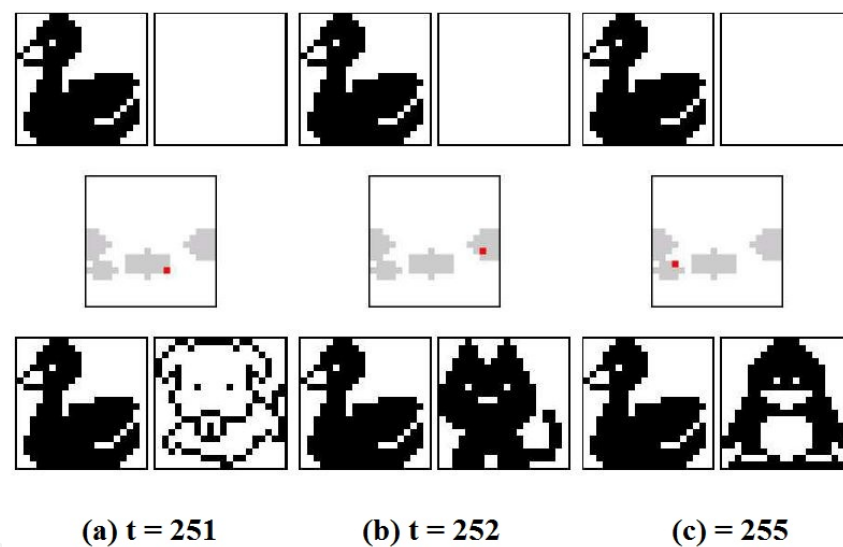


Figure 5. One-to-Many Associations for Binary Patterns (When "duck" was Given).

Figure 6 shows the Map Layer after the pattern pairs shown in Fig. 3 were memorized. In Fig. 6, red neurons show the center neuron in each area, blue neurons show the neurons in areas for the patterns including "crow", green neurons show the neurons in areas for the patterns including "duck". As shown in Fig. 6, the proposed model can learn each learning pattern with various size area. Moreover, since the connection weights are updated not only in the area but also in the neighborhood area in the proposed model, areas corresponding to the pattern pairs including "crow"/"duck" are arranged in near area each other.

Learning Pattern Long Radius a_i Short Radius b_i		
"crow" – "lion"	2.5	1.5
"crow" – "monkey"	3.5	2.0
"crow" – "mouse"	4.0	2.5
"duck" – "penguin"	2.5	1.5
"duck" – "dog"	3.5	2.0
"duck" – "cat"	4.0	2.5

Table 2. Area Size corresponding to Patterns in Fig. 3.

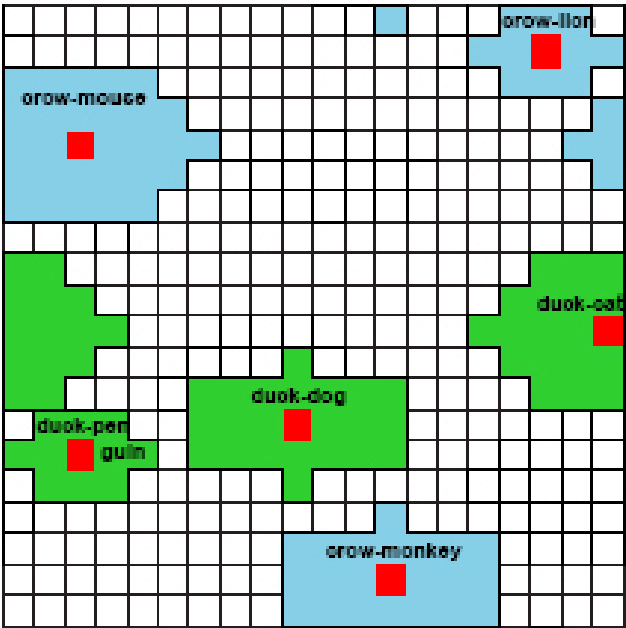


Figure 6. Area Representation for Learning Pattern in Fig. 3.

Input Pattern		Output Pattern	Area Size	Recall Times
crow	lion	11 (1.0)	43 (1.0)	
	monkey	23 (2.1)	87 (2.0)	
	mouse	33 (3.0)	120 (2.8)	
duck	penguin	11 (1.0)	39 (1.0)	
	dog	23 (2.1)	79 (2.0)	
	cat	33 (3.0)	132 (3.4)	

Table 3. Recall Times for Binary Pattern corresponding to "crow" and "duck".

Table 3 shows the recall times of each pattern in the trial of Fig. 4 ($t=1\sim250$) and Fig. 5 ($t=251\sim 500$). In this table, normalized values are also shown in (). From these results, we can confirmed that the proposed model can realize probabilistic associations based on the weight distributions.

4.2.2. Analog patterns

In this experiment, the analog patterns including one-to-many relations shown in Fig. 7 were memorized in the network composed of 800 neurons in the Input/Output Layer and 400 neurons in the Map Layer. Figure 8 shows a part of the association result when “bear” was given to the Input/Output Layer. As shown in Fig. 8, when “bear” was given to the network, “lion” ($t=1$), “raccoon dog” ($t=2$) and “penguin” ($t=3$) were recalled. Figure 9 shows a part of the association result when “mouse” was given to the Input/Output Layer. In this case, “monkey” ($t=251$), “hen” ($t=252$) and “chick” ($t=253$) were recalled. From these results, we can confirmed that the proposed model can recall analog patterns including one-to-many relations.

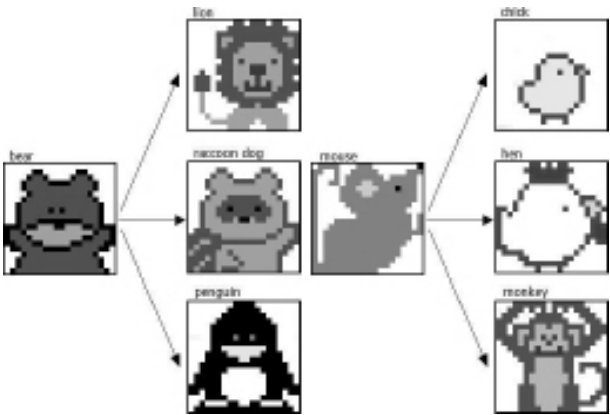


Figure 7. Training Patterns including One-to-Many Relations (Analog Pattern).

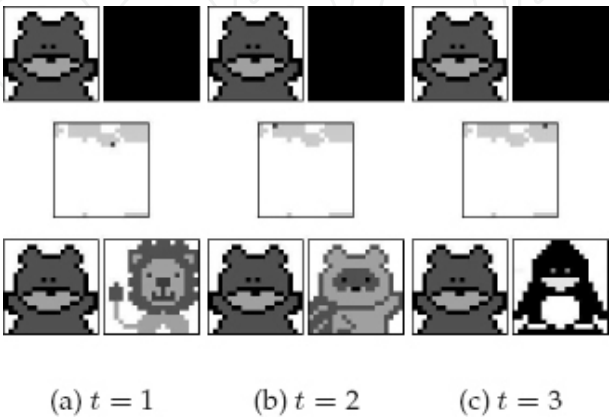


Figure 8: One-to-Many Associations for Analog Patterns (When “bear” was Given).

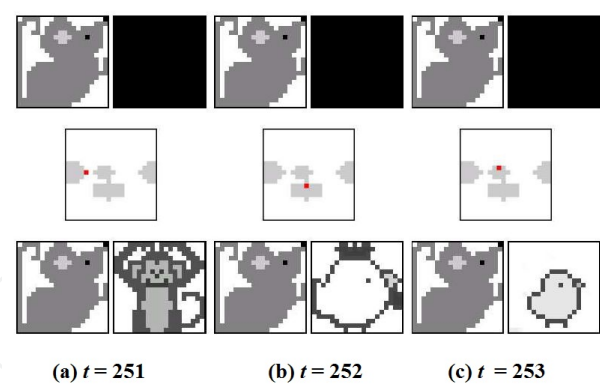


Figure 9. One-to-Many Associations for Analog Patterns (When “mouse” was Given).

Learning Pattern	Long Radius a_i	Short Radius b_i
“bear”–“lion”	2.5	1.5
“bear”–“raccoon dog”	3.5	2.0
“bear”–“penguin”	4.0	2.5
“mouse”–“chick”	2.5	1.5
“mouse”–“hen”	3.5	2.0
“mouse”–“monkey”	4.0	2.5

Table 4. Area Size corresponding to Patterns in Fig. 7.

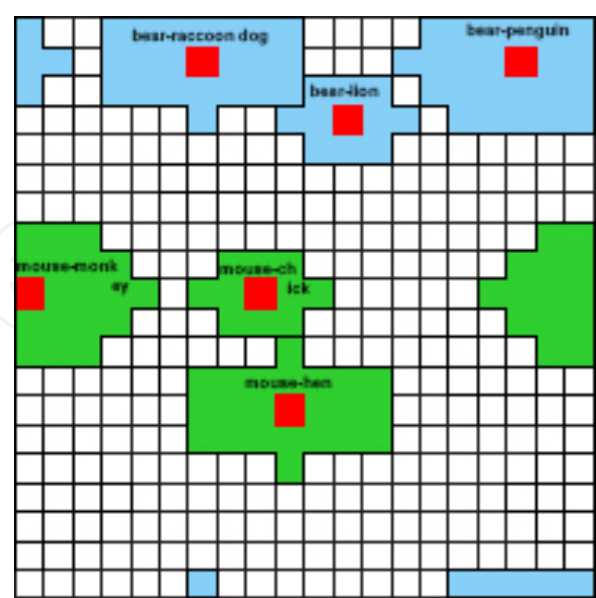


Figure 10. Area Representation for Learning Pattern in Fig. 7.

Input Pattern	Output Pattern	Area Size	Recall Times
bear	lion	11 (1.0)	40 (1.0)
	raccoon dog	23 (2.1)	90 (2.3)
	penguin	33 (3.0)	120 (3.0)
mouse	chick	11 (1.0)	38 (1.0)
	hen	23 (2.1)	94 (2.5)
	monkey	33 (3.0)	118 (3.1)

Table 5. Recall Times for Analog Pattern corresponding to “bear” and “mouse”.

Figure 10 shows the Map Layer after the pattern pairs shown in Fig. 7 were memorized. In Fig. 10, red neurons show the center neuron in each area, blue neurons show the neurons in the areas for the patterns including “bear”, green neurons show the neurons in the areas for the patterns including “mouse”. As shown in Fig. 10, the proposed model can learn each learning pattern with various size area.

Table 5 shows the recall times of each pattern in the trial of Fig. 8 ($t=1\sim 250$) and Fig. 9 ($t=251\sim 500$). In this table, normalized values are also shown in (). From these results, we can confirmed that the proposed model can realize probabilistic associations based on the weight distributions.

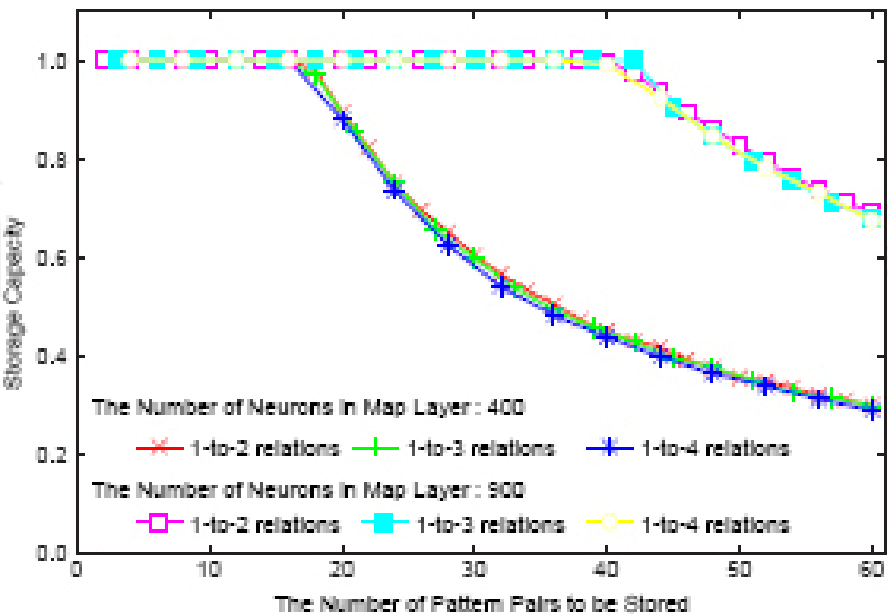


Figure 11. Storage Capacity of Proposed Model (Binary Patterns).

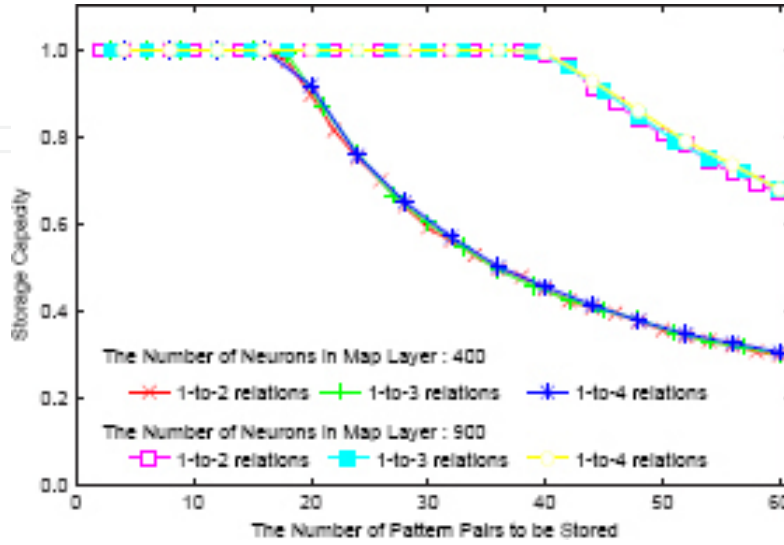


Figure 12. Storage Capacity of Proposed Model (Analog Patterns).

4.3. Storage capacity

Here, we examined the storage capacity of the proposed model. Figures 11 and 12 show the storage capacity of the proposed model. In this experiment, we used the network composed of 800 neurons in the Input/Output Layer and 400/900 neurons in the Map Layer, and 1-to- P ($P=2,3,4$) random pattern pairs were memorized as the area ($a_i=2.5$ and $b_i=1.5$). Figures 11 and 12 show the average of 100 trials, and the storage capacities of the conventional model(16) are also shown for reference in Figs. 13 and 14. From these results, we can confirm that the storage capacity of the proposed model is almost same as that of the conventional model(16). As shown in Figs. 11 and 12, the storage capacity of the proposed model does not depend on binary or analog pattern. And it does not depend on P in one-to- P relations. It depends on the number of neurons in the Map Layer.

4.4. Robustness for noisy input

4.4.1. Association result for noisy input

Figure 15 shows a part of the association result of the proposed model when the pattern “cat” with 20% noise was given during $t=1 \sim 500$. Figure 16 shows a part of the association result of the proposed model when the pattern “crow” with 20% noise was given $t=501 \sim 1000$. As shown in these figures, the proposed model can recall correct patterns even when the noisy input was given.

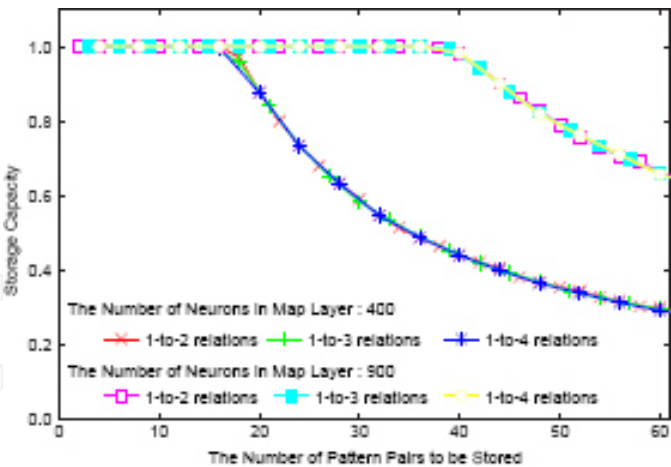


Figure 13. Storage Capacity of Conventional Model [16] (Binary Pattern)

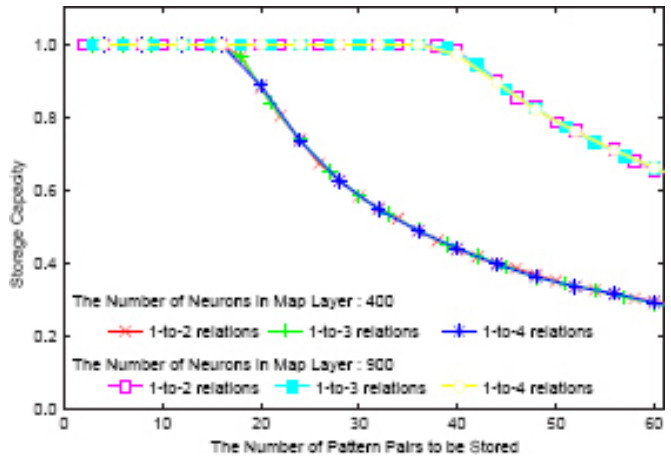


Figure 14. Storage Capacity of Conventional Model [16] (Analog Patterns).

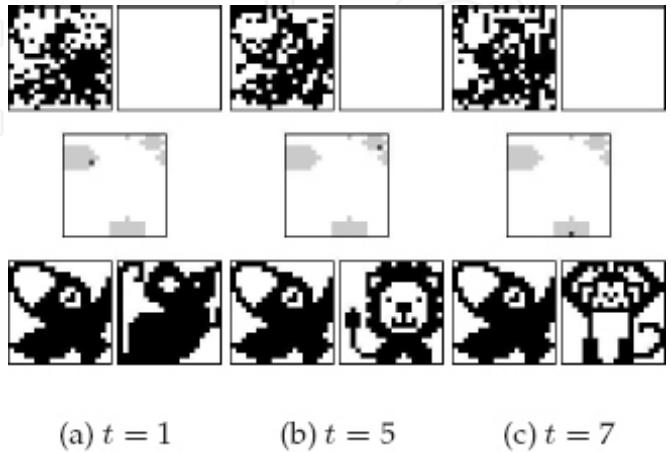


Figure 15. Association Result for Noisy Input (When "crow" was Given.).

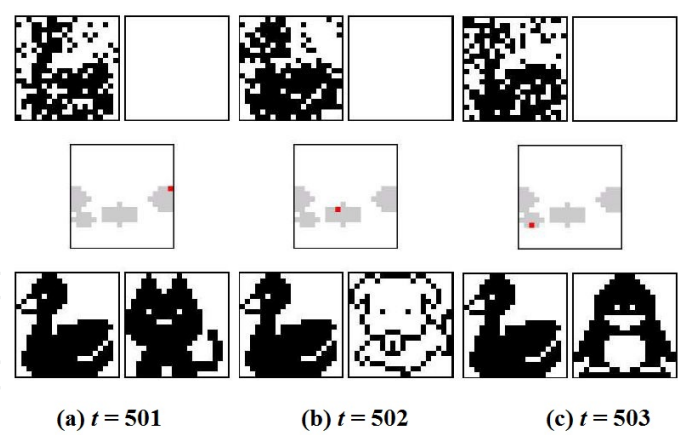


Figure 16. Association Result for Noisy Input (When “duck” was Given.).

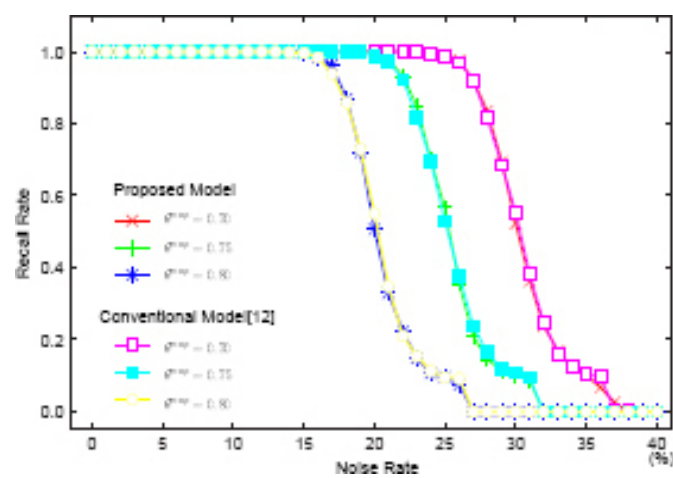


Figure 17. Robustness for Noisy Input (Binary Patterns).

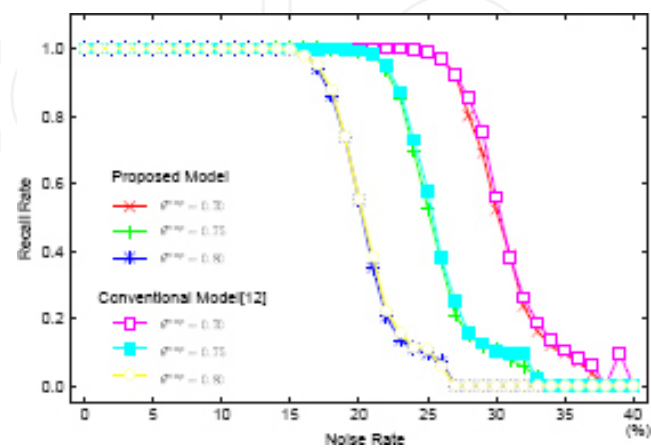


Figure 18. Robustness for Noisy Input (Analog Patterns).

4.4.2. Robustness for noisy input

Figures 17 and 18 show the robustness for noisy input of the proposed model. In this experiment, 10 random patterns in one-to-one relations were memorized in the network composed of 800 neurons in the Input/Output Layer and 900 neurons in the Map Layer. Figures 17 and 18 are the average of 100 trials. As shown in these figures, the proposed model has robustness for noisy input as similar as the conventional model(16).

4.5. Robustness for damaged neurons

4.5.1. Association result when some neurons in map layer are damaged

Figure 19 shows a part of the association result of the proposed model when the pattern “bear” was given during $t=1 \sim 500$. Figure 20 shows a part of the association result of the proposed model when the pattern “mouse” was given $t=501 \sim 1000$. In these experiments, the network whose 20% of neurons in the Map Layer are damaged were used. As shown in these figures, the proposed model can recall correct patterns even when the some neurons in the Map Layer are damaged.

4.5.2. Robustness for damaged neurons

Figures 21 and 22 show the robustness when the winner neurons are damaged in the proposed model. In this experiment, 1~ 10 random patterns in one-to-one relations were memorized in the network composed of 800 neurons in the Input/Output Layer and 900 neurons in the Map Layer. Figures 21 and 22 are the average of 100 trials. As shown in these figures, the proposed model has robustness when the winner neurons are damaged as similar as the conventional model [16].

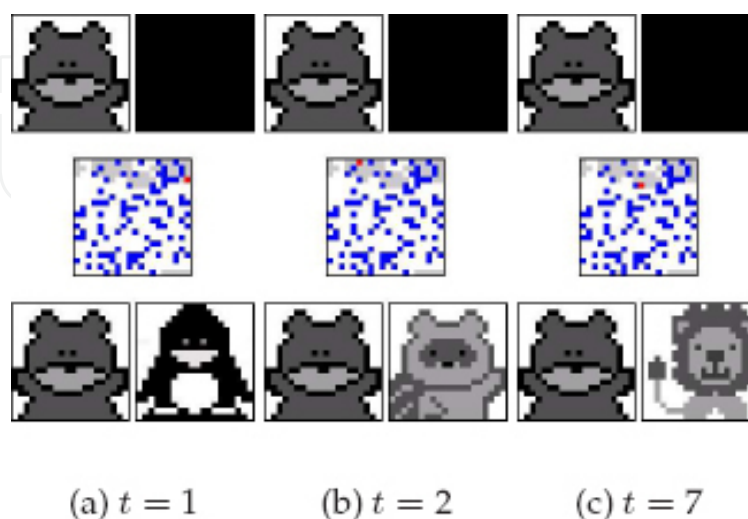


Figure 19. Association Result for Damaged Neurons (When “bear” was Given.).

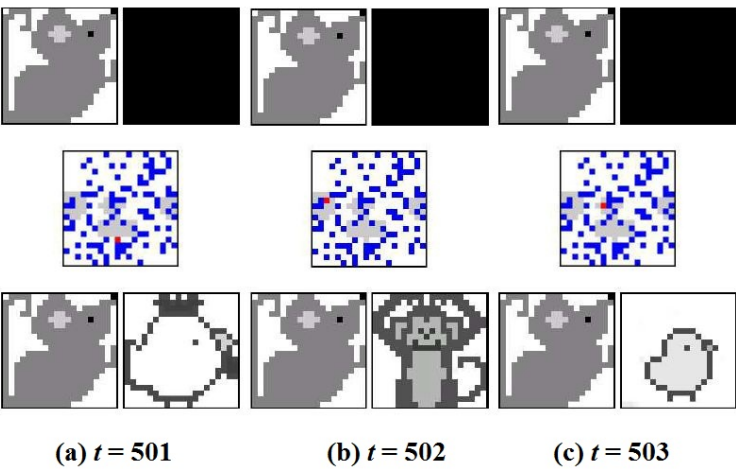


Figure 20. Association Result for Damaged Neurons (When “mouse” was Given.).

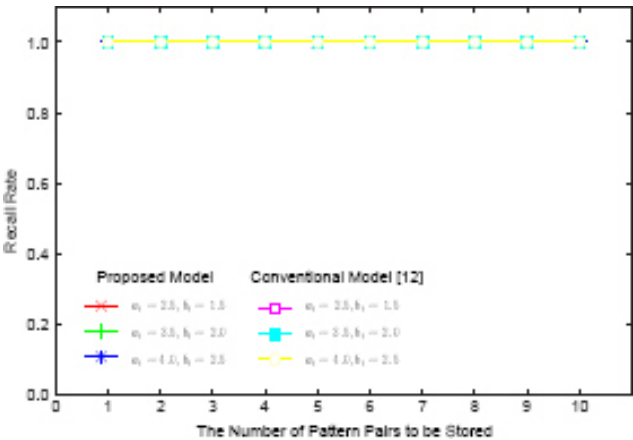


Figure 21. Robustness of Damaged Winner Neurons (Binary Patterns).

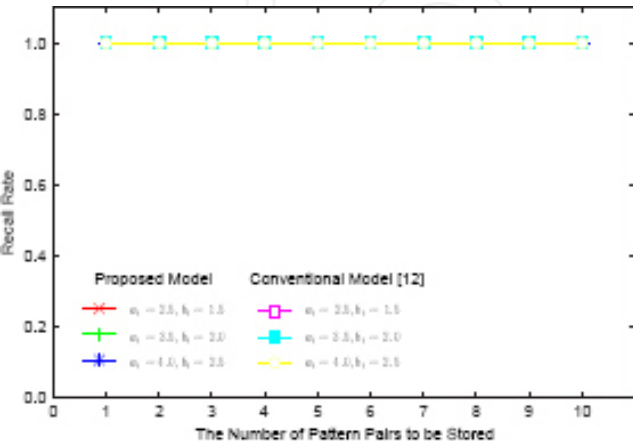


Figure 22. Robustness of Damaged Winner Neurons (Analog Patterns).

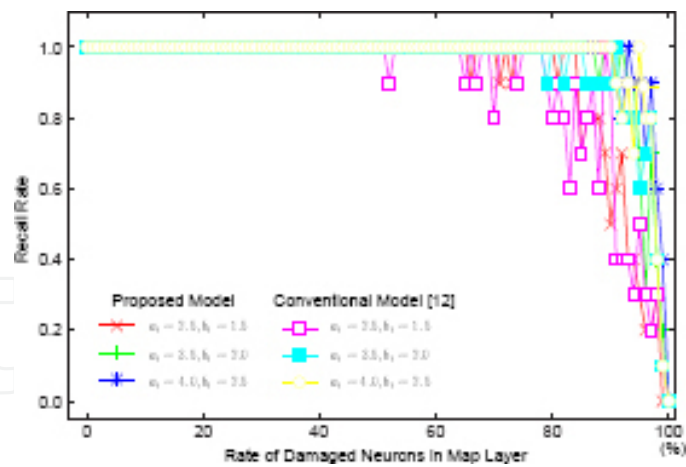


Figure 23: Robustness for Damaged Neurons (Binary Patterns).

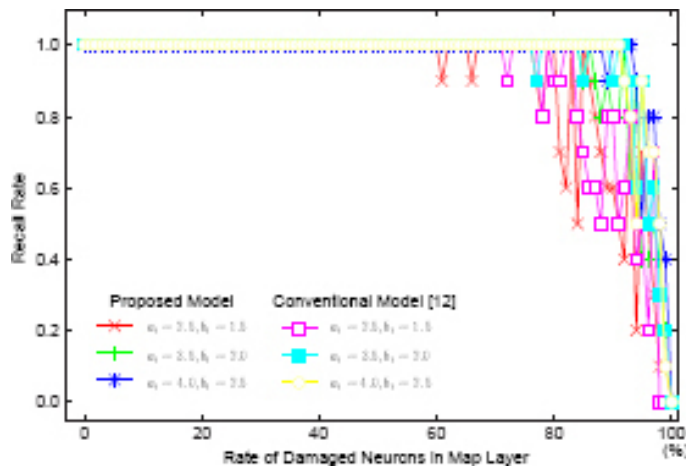


Figure 24. Robustness for Damaged Neurons (Analog Patterns).

Figures 23 and 24 show the robustness for damaged neurons in the proposed model. In this experiment, 10 random patterns in one-to-one relations were memorized in the network composed of 800 neurons in the Input/Output Layer and 900 neurons in the Map Layer. Figures 23 and 24 are the average of 100 trials. As shown in these figures, the proposed model has robustness for damaged neurons as similar as the conventional model [16].

4.6. Learning speed

Here, we examined the learning speed of the proposed model. In this experiment, 10 random patterns were memorized in the network composed of 800 neurons in the Input/Output Layer and 900 neurons in the Map Layer. Table 6 shows the learning time of the proposed model and the conventional model[16]. These results are average of 100 trials on the Personal Computer (Intel Pentium 4 (3.2GHz), FreeBSD 4.11, gcc 2.95.3). As shown in Table 6, the learning time of the proposed model is shorter than that of the conventional model.

5. Conclusions

In this paper, we have proposed the Improved Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution. This model is based on the conventional Kohonen Feature Map Probabilistic Associative Memory based on Weights Distribution. The proposed model can realize probabilistic association for the training set including one-to-many relations. Moreover, this model has enough robustness for noisy input and damaged neurons. We carried out a series of computer experiments and confirmed the effectiveness of the proposed model.

		Learning Time (seconds)
Proposed Model	(Binary Patterns)	0.87
Proposed Model	(Analog Patterns)	0.92
Conventional Model(16)	(Binary Patterns)	1.01
Conventional Model(16)	(Analog Patterns)	1.34

Table 6. Learning Speed.

Author details

Shingo Noguchi and Osana Yuko*

*Address all correspondence to: osana@cs.teu.ac.jp

Tokyo University of Technology, Japan

References

[1] Rumelhart, D. E., McClelland, J. L., & the PDP Research Group. (1986). Parallel Distributed Processing, Exploitations in the Microstructure of Cognition. 11, Foundations, The MIT Press.

[2] Kohonen, T. (1994). Self-Organizing Maps. Springer.

[3] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy Sciences USA*, 79, 2554-2558.

[4] Kosko, B. (1988). Bidirectional associative memories. *IEEE Transactions on Neural Networks*, 18(1), 49-60.

[5] Carpenter, G. A., & Grossberg, S. (1995). Pattern Recognition by Self-organizing Neural Networks. The MIT Press.

- [6] Watanabe, M., Aihara, K., & Kondo, S. (1995). Automatic learning in chaotic neural networks. *IEICE-A*, J78-A(6), 686-691, (in Japanese).
- [7] Arai, T., & Osana, Y. (2006). Hetero chaotic associative memory for successive learning with give up function -- One-to-many associations --, *Proceedings of IASTED Artificial Intelligence and Applications*, Innsbruck.
- [8] Ando, M., Okuno, Y., & Osana, Y. (2006). Hetero chaotic associative memory for successive learning with multi-winners competition. *Proceedings of IEEE and INNS International Joint Conference on Neural Networks*, Vancouver.
- [9] Ichiki, H., Hagiwara, M., & Nakagawa, M. (1993). Kohonen feature maps as a supervised learning machine. *Proceedings of IEEE International Conference on Neural Networks*, 1944-1948.
- [10] Yamada, T., Hattori, M., Morisawa, M., & Ito, H. (1999). Sequential learning for associative memory using Kohonen feature map. *Proceedings of IEEE and INNS International Joint Conference on Neural Networks*, 555, Washington D.C.
- [11] Hattori, M., Arisumi, H., & Ito, H. (2001). Sequential learning for SOM associative memory with map reconstruction. *Proceedings of International Conference on Artificial Neural Networks*, Vienna.
- [12] Sakurai, N., Hattori, M., & Ito, H. (2002). SOM associative memory for temporal sequences. *Proceedings of IEEE and INNS International Joint Conference on Neural Networks*, 950-955, Honolulu.
- [13] Abe, H., & Osana, Y. (2006). Kohonen feature map associative memory with area representation. *Proceedings of IASTED Artificial Intelligence and Applications*, Innsbruck.
- [14] Ikeda, N., & Hagiwara, M. (1997). A proposal of novel knowledge representation (Area representation) and the implementation by neural network. *International Conference on Computational Intelligence and Neuroscience*, III, 430-433.
- [15] Imabayashi, T., & Osana, Y. (2008). Implementation of association of one-to-many associations and the analog pattern in Kohonen feature map associative memory with area representation. *Proceedings of IASTED Artificial Intelligence and Applications*, Innsbruck.
- [16] Koike, M., & Osana, Y. (2010). Kohonen feature map probabilistic associative memory based on weights distribution. *Proceedings of IASTED Artificial Intelligence and Applications*, Innsbruck.

