Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Variable-Sized KFM Associative Memory with Refractoriness based on Area Representation

Tomohisa Imabayashi
Graduate School of Bionics, Computer and Media Sciences
Tokyo University of Technology
Hachioji, Tokyo

Yuko Osana
School of Computer Science
Tokyo University of Technology
Hachioji, Tokyo
osana@cs.teu.ac.jp

***Abstract*—In this paper, we propose a variable-sized KFM associative memory with refractoriness based on area representation. In the proposed model, the connection weight fixed and semi-fixed neurons are introduced, and the pattern that has already been learned is not destroyed and a new pattern can be memorized. Moreover, when unknown patterns are given, neurons can be added in the map layer if necessary. We carried out a series of computer experiments, and confirmed that the proposed model can learn new patterns which has one-to-many relations successively, neurons can be added in the map layer if necessary, and the proposed model has robustness for noisy input and damaged neurons.**

***Index Terms*—Kohonen Feature Map (KFM) Associative Memory, Successive Learning, Addition of Neurons**

I. 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 networks, a lot of models have been proposed such as the Back Propagation (BP) algorithm, the Kohonen Feature Map (KFM)[1], the Hopfield network, and the Bidirectional Associative Memory (BAM). 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 Adaptive Resonance Theory (ART). 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[2]-[4]. 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 Hebbian learning.

On the other hand, the Kohonen Feature Map (KFM) associative memory [5] has been proposed. Although the KFM associative memory is based on the local representation as similar as the ART, it can learn new patterns successively[6], and its storage capacity is larger than that of models in refs.[2]-[4]. It can deal with auto and hetero associations and the associations for plural sequential patterns including common terms[7]. Moreover, the KFM associative memory with area representation[8] has been proposed. In the model, the area representation[9] 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. Moreover, we have proposed the KFM associative memory with refractoriness based on area representation (KFMAM-R-AR)[10]. In the model, one-to-many associations are realized by refractoriness of neurons. However, since the storage capacities of these models depend on the number of neurons in map layer, they can not learn new patterns more than their original storage capacity.

In this paper, we propose the variable-sized KFM associative memory with refractoriness based on area representation. In the proposed model, the connection weight fixed and semi-fixed neurons are introduced, and the pattern that has already been learned is not destroyed and a new pattern can be memorized. Moreover, when unknown patterns are given, neurons can be added in the map layer if necessary.

II. Variable-Sized KFM Associative Memory with Refractoriness based on Area Representation

Here, we explain the proposed Variable-Sized KFM Associative Memory with Refractoriness based on Area Representation (VS-KFMAM-R-AR).

A. Structure

Figure 1 shows the structure of the proposed model. As shown in Fig.1, the proposed model has two layers; (1) input/output (I/O) layer and (2) map layer, and the I/O layer is divided into some parts. In the proposed model, neurons can be added in the map layer if necessary, so the distance between neurons in the map layer is not equal.

B. Learning Process

In the proposed model, if enough area corresponding to the learning pattern can not be taken, some neurons are added in the map layer.

(1) In the network with the map layer composed of $x_{max} \times y_{max}$ neurons, the connection weights are initialized randomly. Here, x_{max} is the initial number of neurons

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

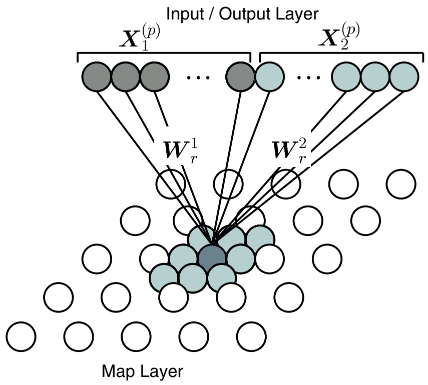


Fig. 1. Structure of Proposed Model.

of a horizontal direction, and y_{max} is the initial number of neurons of a vertical direction. In the initial state, $x_{max} \times y_{max}$ neurons are arranged at the coordinates $(0,0)$, $(1,0)$, $\cdots$, $(x_{max}-1,0)$, $(1,0)$, $\cdots$, $(x_{max}-1, y_{max}-1)$.

(2) The Euclid distances between the input vector $\boldsymbol{X}$ and the weight vector $\boldsymbol{W}_i$, $d(\boldsymbol{X}, \boldsymbol{W}_i)$ are calculated.

(3) The neuron r is decided by

$$r = \underset{i}{\operatorname{argmin}}\, d(\boldsymbol{X}, \boldsymbol{W}_i), \quad (1)$$

and if

$$d(\boldsymbol{X}, \boldsymbol{W}_r) \leq \theta^l \quad (2)$$

is satisfied, the input vector $\boldsymbol{X}$ is regarded as the known pattern, and go to (15). Otherwise, go to (4).

(4) If there is no weight-fixed neuron, the neuron r is selected as the center of the learning area, and go to (12). Otherwise go to (5).

(5) Whether the area corresponding to the input pattern $\boldsymbol{X}$ can be taken without overlapping to the areas for stored patterns is checked. For the weight-fixed neurons z,

$$H_{i,z}^{area(1)} = \frac{1}{1 + \exp\left(-\dfrac{d_{iz} - (d^{max} + d_z^{min})D}{\varepsilon}\right)} \quad (3)$$

is calculated, and if

$$H_{i,z}^{area(1)} > \theta^c \quad (4)$$

is satisfied for all weight-fixed neurons, the neuron i can be a center of the learning area. Here, θ^c is the threshold. The neurons that satisfy the condition given by Eq.(4) for all weight-fixed neuron z are selected as the candidate of the center of the learning area.

In Eq.(3), d_{iz} is the distance between the neuron i and the weight-fixed neuron z. In the proposed model, the map layer is treated as torus, so the distance between the neurons i and j, d_{ij} is given by

$$d_{ij} = \sqrt{(d_{ij}^x)^2 + (d_{ij}^y)^2} \quad (5)$$

$$d_{ij}^x = \begin{cases} |x_j - x_i| - x_{max} & (|x_j - x_i| > x_{max}/2) \\ x_{max} - |x_j - x_i| & (\text{otherwise}) \end{cases} \quad (6)$$

$$d_{ij}^y = \begin{cases} |y_j - y_i| - y_{max} & (|y_j - y_i| > y_{max}/2) \\ y_{max} - |y_j - y_i| & (\text{otherwise}) \end{cases} \quad (7)$$

where x_i and y_i are the coordinates of the neuron i in the map layer, x_j and y_j are the coordinates of the neuron j in the map layer. d_z^{min} is the distance between the weight-fixed neuron z and the nearest neuron, and is given by

$$d_z^{min} = \min_i d_{iz}. \quad (8)$$

d^{max} is the maximum distance between adjacent neurons, and d^{max} is set to 1 in the proposed model. D is the constant which decides the area size (the number of neurons in each area).

If there are some candidate neurons, go to (9). Otherwise, go to (6).

(6) Whether the area corresponding to the input pattern $\boldsymbol{X}$ can be taken without overlapping to the areas for stored patterns when the distance between neurons in the area for stored patterns is reduced to $\phi_n(d_z^{min})$ is checked. Here, $\phi_n(\cdot)$ is given by

$$\phi_n(d) = \begin{cases} \dfrac{d}{2^n}, & \left(\dfrac{d}{2^n} > d^{min}\right) \\ d, & (\text{otherwise}) \end{cases} \quad (9)$$

where n is the number of check in (6). d^{min} is the minimum distance between adjacent neurons. In the area for the input pattern, the distance between adjacent neurons is set to $\phi_{n-1}(d^{max})$.

For the weight-fixed neuron z,

$$H_{i,z}^{area(2n)} = \frac{1}{1 + \exp\left(-\dfrac{d_{iz} - (\phi_{n-1}(d^{max}) + \phi_n(d_z^{min}))D}{\varepsilon}\right)} \quad (10)$$

is calculated, and if

$$H_{i,z}^{area(2n)} > \theta^c \quad (11)$$

is satisfied for all weight-fixed neurons, the neuron i can be a center of the learning area. The neurons that satisfy the condition given by Eq.(11) for all weight-fixed neuron z are selected as the candidate of the center of the learning area. If there are some candidate neurons, go to (9). Otherwise, go to (7).

(7) Whether the area corresponding to the input pattern $\boldsymbol{X}$ can be taken without overlapping to the areas for stored patterns when the distance between neurons in the area for stored patterns is reduced to $\phi_n(d_z^{min})$ and the distance between neurons in the area for the input pattern is set to $\phi_n(d^{max})$ is checked.

For the weight-fixed neuron z,

$$H_{i,z}^{area(2n+1)} = \frac{1}{1+\exp\left(-\frac{d_{iz}-(\phi_n(d^{max})+\phi_n(d_z^{min}))D}{\varepsilon}\right)} \quad (12)$$

is calculated, and if

$$H_{i,z}^{area(2n+1)} > \theta^c \quad (13)$$

is satisfied for all weight-fixed neurons, the neuron i can be a center of the learning area. The neurons that satisfy the condition given by Eq.(13) for all weight-fixed neurons z are selected as the candidate of the center of the learning area. If there are some candidate neurons, go to (9). Otherwise, go to (8).

(8) In (6) and (7), if

$$\phi_{n+1}(d^{max}) \leq d^{min} \quad (14)$$

is satisfied, back to (6). Otherwise, it judges that the input pattern can not be learned as a new pattern.

(9) From the neurons which are selected as the center candidates of the learning area in (5)∼(7), the neuron c that the Euclid distance between the input vector $\boldsymbol{X}$ and its weight vector $\boldsymbol{W}_i$ is minimum is selected.

$$c = \underset{c \in C_{center}}{\operatorname{argmin}} \; d(\boldsymbol{X}, \boldsymbol{W}_i) \quad (15)$$

where C_{center} is the set of the neurons which are the center candidates.

(10) If the center candidates are selected in (6) or (7), the distance in the areas for stored patterns is reduced, and some neurons are added.

If the center candidates are selected in (6), for the area whose center is the neuron z which satisfies

$$\frac{1}{1+\exp\left(-\frac{d_{cz}-(\phi_{n-1}(d^{max})+\phi_{n-1}(d_z^{min}))D}{\varepsilon}\right)} < \theta^c, \quad (16)$$

the distance between neurons in the area is reduced, and neurons are added. The neurons i which satisfy

$$\frac{1}{1+\exp\left(-\frac{d_{iz}-d_z^{min}D}{\varepsilon}\right)} < \theta^c \quad (17)$$

are generated as new neurons. The neuron i' corresponding to the neuron i $((x_i,\ y_i))$ is generated at $(x_{i'},\ y_{i'})$. Here, $x_{i'}$ and $y_{i'}$ are given by

$$x_{i'} = (x_i - x_z)\phi_n(d_z^{min}) + x_z \quad (18)$$

$$y_{i'} = (y_i - y_z)\phi_n(d_z^{min}) + y_z. \quad (19)$$

If the neuron exists at $(x_{i'},\ y_{i'})$, no neuron is added there. The weight vector of the neuron i' $\boldsymbol{W}_{i'}$ is set as

$$\boldsymbol{W}_{i'} = \boldsymbol{W}_i. \quad (20)$$

If the center candidates are selected in (7), the new neurons are added in the area whose center z that satisfy

$$\frac{1}{1+\exp\left(-\frac{d_{cz}-(\phi_n(d^{max})+\phi_{n-1}(d_z^{min}))D}{\varepsilon}\right)} < \theta^c. \quad (21)$$

(11) If the center candidates are selected in (6) or (7), new neurons are added in the area for the new pattern $\boldsymbol{X}$.

If the center candidates are selected in (6) and $n \geq 1$, the neurons are added in the area whose center is the neuron c. The neurons which satisfy

$$\frac{1}{1+\exp\left(-\frac{d_{ic}-\phi_{n-1}(d^{max})D}{\varepsilon}\right)} < \theta^c \quad (22)$$

are generated. The neuron i' corresponding to the neuron i $((x_i,\ y_i))$ is generated at $(x_{i'},\ y_{i'})$. Here, $x_{i'}$ and $y_{i'}$ are given by

$$x_{i'} = (x_i - x_c)\phi_{n-1}(d^{max}) + x_c \quad (23)$$

$$y_{i'} = (y_i - y_c)\phi_{n-1}(d^{max}) + y_c. \quad (24)$$

If the neuron exists at $(x_{i'},\ y_{i'})$, no neuron is added there. The weight vector $\boldsymbol{W}_{i'}$ is generated randomly.

If the center candidates are selected in (7), new neurons are added in the area whose center is the neuron c. The neurons which satisfy

$$\frac{1}{1+\exp\left(-\frac{d_{ic}-\phi_n(d^{max})D}{\varepsilon}\right)} < \theta^c \quad (25)$$

are generated.

(12) The input pattern $\boldsymbol{X}$ is trained in the area whose center is the neuron c. The connection weights which are not fixed are updated by

$$\boldsymbol{W}_i(t+1) = \boldsymbol{W}_i(t) + H(d_{ii^*})\beta(t)h_{ci}(\boldsymbol{X} - \boldsymbol{W}_i(t)). \quad (26)$$

Here, $H(d_{ii^*})$ is given by

$$H(d_{ii^*}) = \frac{1}{1+\exp\left(-\frac{d_{ii^*}-d_c^{min}D}{\varepsilon}\right)} \quad (27)$$

where d_{ii^*} is the distance between the neuron i and the nearest weight-fixed neuron i^*. d_c^{min} is the distance between the neurons in the area whose center is the neuron c, and is given by

$$d_c^{min} = \begin{cases} d^{max}, & \text{(center candidates selected in (5))} \\ \dfrac{d^{max}}{2^{n-1}}, & \text{(center candidates selected in (6))} \\ \dfrac{d^{max}}{2^n}, & \text{(center candidates selected in (7))} \end{cases} \quad (28)$$

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

$$\beta(t) = \frac{-\beta_0(t-T)}{T} \tag{29}$$

where β_0 is the initial value of $\beta(t)$, and T is the upper limit of the learning iterations. In Eq.(26), h_{ci} is the neighborhood function, and is given by

$$h_{ci} = \exp\left(\frac{-\|\boldsymbol{c}-\boldsymbol{i}\|^2}{2\sigma(t)^2}\right). \tag{30}$$

$\sigma(t)$ is given by

$$\sigma(t) = \sigma_i \left(\frac{\sigma_f}{\sigma_i}\right)^{t/T}. \tag{31}$$

In this equation, σ_i is the initial value of $\sigma(t)$ and $\sigma(t)$ varies from σ_i to σ_f ($\sigma_i > \sigma_f$).

(13) (12) is iterated until $d(\boldsymbol{X}^{(p)}, \boldsymbol{W}_c) \le \theta^l$.
(14) The connection weights of the neuron c, $\boldsymbol{W}_c$ are fixed.
(15) (2)∼(14) are iterated when a new pattern set is given.

C. Recall Process

In the recall process of the proposed model, when the pattern $\boldsymbol{X}$ is given to the I/O layer, the output of the neuron i in the map layer at the time t, $x_i^{map}(t)$ is given by

$$x_i^{map}(t) = H^{recall}(d_{ri}) f(u_i^{map}(t)). \tag{32}$$

$$H^{recall}(d_{ri}) = \frac{1}{1+\exp\left(\dfrac{d_{ri}-d_r^{min}D}{\varepsilon}\right)} \tag{33}$$

where d_{ri} is the Euclid distance between the winner neuron r and the neuron i, and the neuron r is determined as follows:

$$r = \underset{i}{\mathrm{argmax}}\, u_i^{map}(t). \tag{34}$$

d_r^{min} is the distance between the winner neuron r and the nearest neuron. $f(u_i^{map}(t))$ is given by

$$f(u_i^{map}(t)) = \begin{cases} 1, & (u_i^{map}(t) > \theta^{map} \text{ and } u_i^{map}(t) > \theta^{min}) \\ 0, & \text{(otherwise)} \end{cases} \tag{35}$$

where $u_i^{map}(t)$ is the internal state of the neuron i in the map layer at the time t, θ^{map} and θ^{min} are the thresholds of the neuron, and θ^{map} is given by

$$\theta^{map} = u_{min} + a(u_{max} - u_{min}) \tag{36}$$

$$u_{min} = \min_i u_i^{map}(t) \tag{37}$$

$$u_{max} = \max_i u_i^{map}(t) \tag{38}$$

where a ($0.5 < a < 1$) is the coefficient.

When the binary pattern $\boldsymbol{X}$ is given, the internal state of the neuron i in the map layer at the time t $u_i^{map}(t)$ is given by

$$u_i^{map}(t) = 1 - \frac{d^{in}(\boldsymbol{X}, \boldsymbol{W}_i)}{\sqrt{N^{in}}} - \alpha \sum_{d=0}^{t} k_r^d x_i^{map}(t-d) \tag{39}$$

where $d^{in}(\boldsymbol{X}, \boldsymbol{W}_i)$ is the Euclid distance between the input patterns $\boldsymbol{X}$ and the connection weights $\boldsymbol{W}_i$. In the recall process, since all neurons in the I/O layer not always receive the input, the distance for the part where the pattern is given is calculated by

$$d^{in}(\boldsymbol{X}, \boldsymbol{W}_i) = \sqrt{\sum_{k\in C}(X_k - W_{ik})^2} \tag{40}$$

where C shows the set of the neurons in the I/O layer which receive the input. In Eq.(39), N^{in} is the number of neurons which receive the input in the I/O layer, α is the scaling factor of the refractoriness, and $k_r (0 \le k_r < 1)$ is the damping factor. The output of the neuron k in the I/O layer at the time t $x_k^{in}(t)$ is calculated by

$$x_k^{in}(t) = \begin{cases} 1, & (u_k^{in}(t) \ge \theta_b^{in}) \\ 0, & \text{(otherwise)} \end{cases} \tag{41}$$

$$u_k^{in}(t) = \frac{1}{\sum_i x_i^{map}(t)} \sum_{i:x_i > \theta^{out}} W_{ik} \tag{42}$$

where θ_b^{in} is the threshold of the neuron in the I/O layer, and θ^{out} is the threshold for the output of the neuron in the map layer.

On the other hand, when the analog pattern is given to the I/O layer, $u_i^{map}(t)$ is calculated by

$$u_i^{map}(t) = \frac{1}{N^{in}} \sum_{k\in C} g(X_k - W_{ik}) - \alpha \sum_{d=0}^{t} k_r^d x_i^{map}(t-d) \tag{43}$$

where $g(\cdot)$ is given by

$$g(b) = \begin{cases} 1, & (|b| < \theta^d) \\ 0, & \text{(otherwise)} \end{cases} \tag{44}$$

where θ^d is the threshold.

The output of the neuron k in the I/O layer at the time t, $x_k^{in}(t)$ is calculated as follows:

$$x_k^{in}(t) = \frac{1}{\sum_i x_i^{map}(t)} \sum_{i\,:\,x_i^{map} > \theta^{out}} W_{ik}. \tag{45}$$

III. Computer Experiment Results

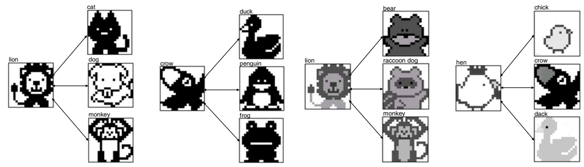
Here, we show the computer experiment results to demonstrate the effectiveness of the proposed model.

A. Association Result

In this experiment, the proposed model composed of 800 (= 400×2) neurons in the I/O layer and 100 (= 10×10) in the initial map layer, and the training pattern shown in Fig.2 were memorized. As shown in Figs.3 and 4, the proposed model could realize one-to-many associations.

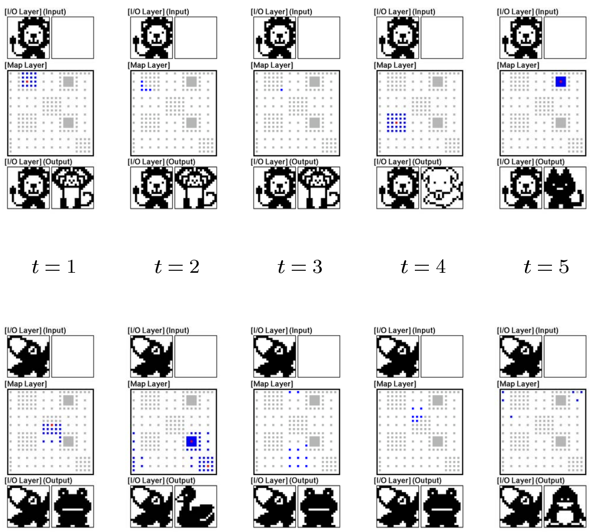
B. Variation of Map Layer

Figure 5 shows the variation of the map layer in the learning. In this experiment, we used the network composed of 800 (= 400 × 2) neurons in the I/O layer and 100 (= 10 × 10) neurons in the initial map layer, and 60 random binary pattern pairs were memorized. In the network after 60 random pattern pairs were memorized, the map layer has 1930 neurons.



(a) Binary Patterns. (b) Analog Patterns.

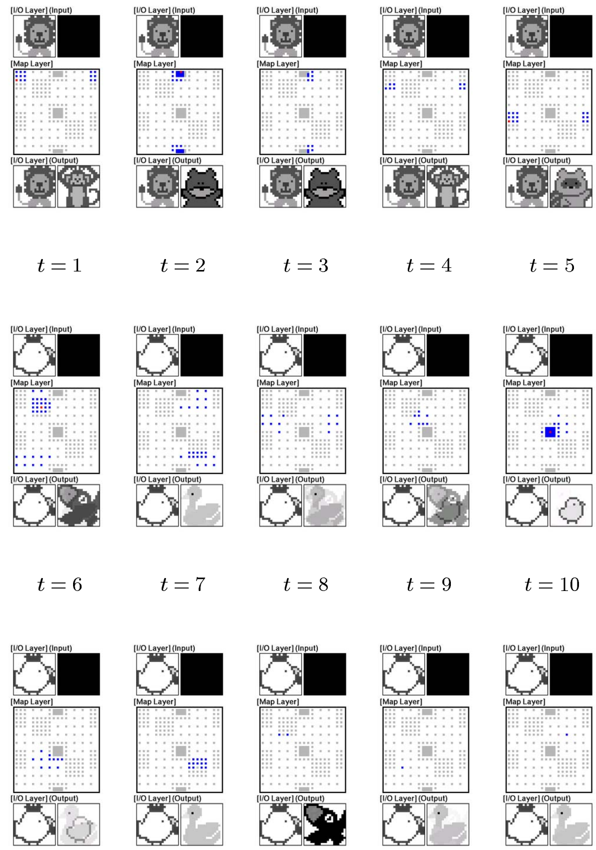
Fig. 2. Training Patterns.



$t = 1$ $t = 2$ $t = 3$ $t = 4$ $t = 5$

$t = 6$ $t = 7$ $t = 8$ $t = 10$ $t = 11$

Fig. 3. Association Results of Binary Patterns.



$t = 1$ $t = 2$ $t = 3$ $t = 4$ $t = 5$

$t = 6$ $t = 7$ $t = 8$ $t = 9$ $t = 10$

$t = 11$ $t = 12$ $t = 13$ $t = 14$ $t = 15$

Fig. 4. Association Results of Analog Patterns.

C. Storage Capacity

Figure 6 shows the storage capacity of the proposed model composed of 100 (= 50 × 2) in the I/O layer and 100 (= 10 × 10) neurons in the initial map layer.

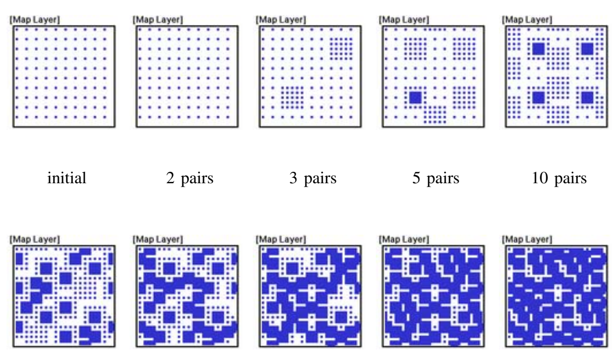
D. Robustness for Noisy Input/Damaged Neurons

Figures 7 and 8 show the robustness for noisy input and damaged neurons. As shown these figures, the proposed model has robustness for noisy input and damaged neurons.

IV. Conclusion

In this paper, we have proposed the variable-sized KFM associative memory with refractoriness based on area representation. We carried out a series of computer experiments and confirmed that the proposed model has following features.

(1) It can learn new patterns which has one-to-many relations successively, and neurons can be added in the map layer if necessary.
(2) It has robustness for noisy input.
(3) It has robustness for damaged neurons.



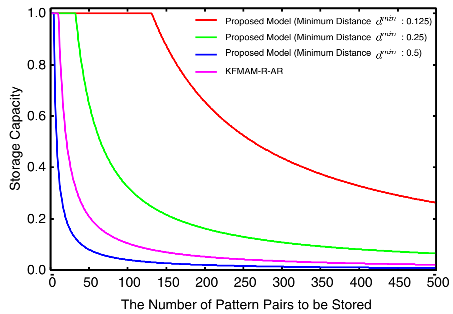
initial 2 pairs 3 pairs 5 pairs 10 pairs

20 pairs 30 pairs 40 pairs 50 pairs 60 pairs

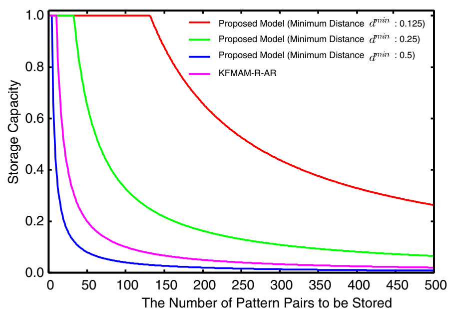
Fig. 5. Variation of Map Layer.

References

[1] T. Kohonen : Self-Organizing Maps, Springer, 1994.

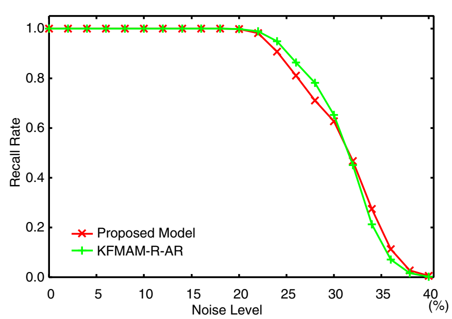


(a) Binary Patterns.

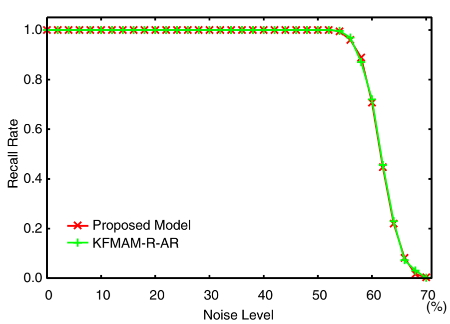


(b) Analog Patterns.

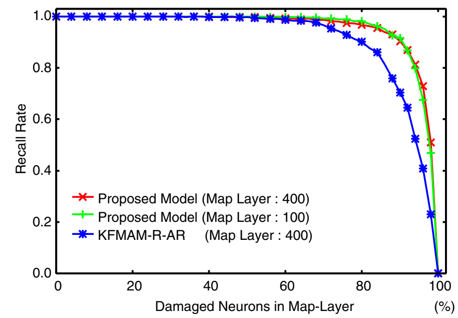
Fig. 6. Storage Capacity.



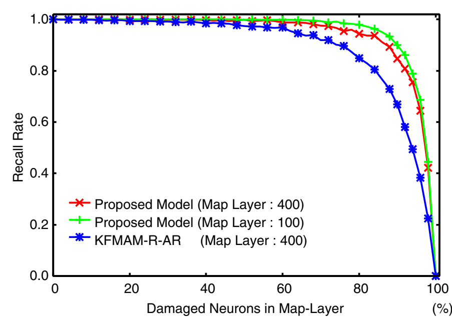
(a) Binary Patterns.



(b) Analog Patterns.



(a) Binary Patterns.



(b) Analog Patterns.

Fig. 8. Robustness for Damaged Neurons.

[2] M. Watanabe, K. Aihara and S. Kondo : "Automatic learning in chaotic neural networks," IEICE-A, Vol.J78-A, No.6, pp.686–691, 1995 (in Japanese).

[3] Y. Osana and M. Hagiwara : "Successive learning in chaotic neural network," International Journal of Neural Systems, Vol.9, No.4, pp.285–299, 1999.

[4] N. Kawasaki, Y. Osana and M. Hagiwara : "Chaotic associative memory for successive learning using internal patterns," IEEE International Conference on Systems, Man and Cybernetics, 2000.

[5] H. Ichiki, M. Hagiwara and M. Nakagawa : "Kohonen feature maps as a supervised learning machine," Proc. of IEEE International Conference on Neural Networks, pp.1944–1948, 1993.

[6] T. Yamada, M. Hattori, M. Morisawa and H. Ito : "Sequential learning for associative memory using Kohonen feature map," Proc. of IEEE and INNS International Joint Conference on Neural Networks, paper no.555, Washington D.C., 1999.

[7] M. Hattori, H. Arisumi and H. Ito : "SOM associative memory for temporal sequences," Proc. of IEEE and INNS International Joint Conference on Neural Networks, pp.950–955, Honolulu, 2002.

[8] H. Abe and Y. Osana : "Kohonen feature map associative memory with area representation," Proceedings of IASTED Artificial Intelligence and Applications, Innsbruck, 2006.

[9] N. Ikeda and M. Hagiwara : "A proposal of novel knowledge representation (area representation) and the implementation by neural network", International Conference on Computational Intelligence and Neuroscience , III, pp,430–433, 1997.

[10] T. Imabayashi and Y. Osana : "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, 2008.