2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

One-to-Many Association Ability of Chaotic Quaternionic Multidirectional Associative Memory — Relation between Damping Factors and One-to-Many Association Ability —

Hayato KAWANO and Yuko OSANA

School of Computer Science, Tokyo University of Technology
1404-1 Katakura, Hachioji, Tokyo, 192-0982, Japan Email: osana@stf.teu.ac.jp

Abstract—The association ability of associative memories composed of chaotic neuron models or chaotic neuron-based models such as chaotic complex-valued neuron model and chaotic quaternionic neuron model are very sensitive to chaotic neuron parameters such as scaling factor of refractoriness α, damping factors (k_m and k_r) and so on. In this paper, we investigate the relation between the damping factors (k_m and k_r) and one-to-many association ability of the Chaotic Quaternionic Multidirectional Associative Memory (CQMAM). The CQMAM is based on the Multidirectional Associative Memory and composed of quaternionic neurons and chaotic quaternionic neurons, and it can realize one-to-many associations of M-tuple multi-valued patterns.

1. Introduction

Although a lot of associative memories have been proposed, most of these models can deal with only one-to-one associations[1][2]. In contrast, as a model which can realize one-to-many associations, some models which are based on the chaotic neuron models[3] or chaotic neuron-based models such as chaotic complex-valued neuron model[4] and chaotic quaternionic neuron model[5] have been proposed[6]–[13]. However, the association ability of neural networks composed of chaotic neuron models or chaotic neuron-based models are very sensitive to parameters such as scaling factor of refractoriness α, damping factors (k_m and k_r) and so on.

In this paper, we investigate the relation between the damping factors (k_m and k_r) and one-to-many association ability of the Chaotic Quaternionic Multidirectional Associative Memory (CQMAM)[13]. The CQMAM is based on the Multidirectional Associative Memory and composed of quaternionic neurons[14] and chaotic quaternionic neurons[5], and it can realize one-to-many associations of M-tuple multi-valued patterns.

2. Chaotic Quaternionic Multidirectional Associative Memory

Here, we explain the Chaotic Quaternionic Multidirectional Associative Memory (CQMAM)[13] which are investigated in this research.

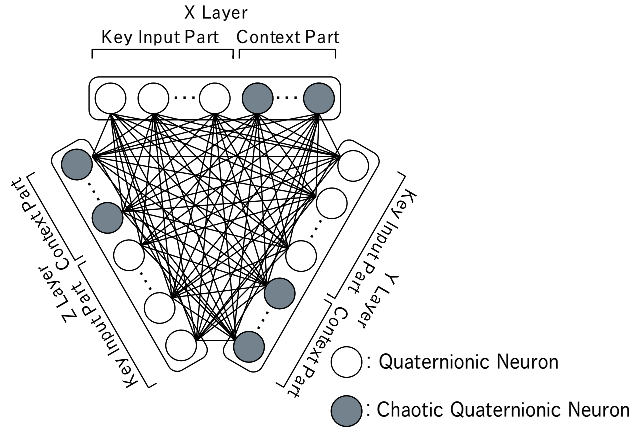


Figure 1: Structure of CQMAM.

2.1. Structure

The CQMAM has more than two layers as similar as the conventional Multidirectional Associative Memory[2]. Figure 1 shows the structure of the 3-layered CQMAM. In this model, each layer composed of two parts; (1) Key Input Part composed of quaternionic neurons[14] and (2) Context Part composed of chaotic quaternionic neurons[5].

2.2. Learning Process

In the CQMAM, the connection weights are trained by the orthogonal learning. However, the orthogonal learning can not deal with the training pattern set including one-to-many relations because the stored common data cause superimposed patterns. In the CQMAM, the patterns with its own contextual information are memorized by the orthogonal learning as similar as the conventional CCMAM[9].

The connection weights from the layer y to the layer x, $\boldsymbol{w}^{xy}$ and the connection weights from the layer x to the layer y , $\boldsymbol{w}^{yx}$ are determined as follows:

$$\boldsymbol{w}^{xy} = \boldsymbol{X}_y(\boldsymbol{X}_x^*\boldsymbol{X}_x)^{-1}\boldsymbol{X}_x^* \tag{1}$$

$$\boldsymbol{w}^{yx} = \boldsymbol{X}_x(\boldsymbol{X}_y^*\boldsymbol{X}_y)^{-1}\boldsymbol{X}_y^* \tag{2}$$

where * shows the conjugate transpose, and -1 shows the inverse. $\boldsymbol{X}_x$ and $\boldsymbol{X}_y$ are the training pattern matrix which are memorized in the layer x and the layer y, and are given by

$$\boldsymbol{X}_x = \{\boldsymbol{X}_x^{(1)}, \cdots, \boldsymbol{X}_x^{(p)}, \cdots, \boldsymbol{X}_x^{(P)}\} \tag{3}$$

$$\boldsymbol{X}_y = \{\boldsymbol{X}_y^{(1)}, \cdots, \boldsymbol{X}_y^{(p)}, \cdots, \boldsymbol{X}_y^{(P)}\} \quad (4)$$

where $\boldsymbol{X}_x^{(p)}$ is the pattern p which is stored in the layer x, $\boldsymbol{X}_y^{(p)}$ is the pattern p which is stored in the layer y and P is the number of the training pattern sets.

2.3. Recall Process

Since contextual information is usually unknown for users, in the recall process, only the Key Input Part receives input in the first step. For example, in the training sets which is given by

$$\begin{aligned}&\{(\boldsymbol{X}_1\_\boldsymbol{C}_{X1}, \boldsymbol{Y}_1\_\boldsymbol{C}_{Y1}, \boldsymbol{Z}_1\_\boldsymbol{C}_{Z1}),\\ &(\boldsymbol{X}_1\_\boldsymbol{C}_{X2}, \boldsymbol{Y}_2\_\boldsymbol{C}_{Y2}, \boldsymbol{Z}_2\_\boldsymbol{C}_{Z2}),\\ &(\boldsymbol{X}_2\_\boldsymbol{C}_{X3}, \boldsymbol{Y}_3\_\boldsymbol{C}_{Y3}, \boldsymbol{Z}_3\_\boldsymbol{C}_{Z3})\}, \quad (5)\end{aligned}$$

and $\boldsymbol{X_1}$ is used as an input to the CQMAM. Here, $\boldsymbol{C}_{\mathbf{xx}}$ (such as $\boldsymbol{C}_{X1}$ and $\boldsymbol{C}_{Y1}$) shows the contextual information. In the CQMAM, when $\boldsymbol{X}_1$ is given to the network as an initial input, since the chaotic quaternionic neurons in the Context Part change their states by chaos, one-to-many associations can be realized as follows:

$$\begin{aligned}(\boldsymbol{X}_1\_\boldsymbol{0}, ?, ?) \rightarrow \cdots \rightarrow (\boldsymbol{X}_1\_\boldsymbol{C}_{X1}, \boldsymbol{Y}_1, \boldsymbol{Z}_1) \rightarrow \cdots\\ \rightarrow (\boldsymbol{X}_1\_\boldsymbol{C}_{X2}, \boldsymbol{Y}_2, \boldsymbol{Z}_2) \rightarrow \cdots \quad (6)\end{aligned}$$

The recall process of the CQMAM has the following procedures when the input pattern is given to the layer x.

Step 1 : Input to Layer x

The input pattern is given to the layer x.

Step 2 : Propagation from Layer x to Other Layers

When the pattern is given to the layer x, the information is propagated to the Key Input Part in the other layers. The output of the neuron k in the Key Input Part of the layer y ($y \neq x$), $\boldsymbol{x}_k^y(t)$ is given by

$$\boldsymbol{x}_k^y(t) = f\left(\sum_{j=1}^{N^x} \boldsymbol{w}_{kj}^{yx} \boldsymbol{x}_j^x(t)\right) \quad (7)$$

where N^x is the number of neurons in the layer x, $\boldsymbol{w}_{kj}^{yx}$ is the connection weight from the neuron j in the layer x to the neuron k in the layer y, and $\boldsymbol{x}_j^x(t)$ is the output of the neuron j in the layer x at the time t. $f(\cdot)$ is the output function which is given by

$$\boldsymbol{f}(u) = f^{(e)}(u^{(e)}) + f^{(i)}(u^{(i)})\boldsymbol{i} + f^{(j)}(u^{(j)})\boldsymbol{j} + f^{(k)}(u^{(k)})\boldsymbol{k} \quad (8)$$

$$f^{(e)}(u) = f^{(i)}(u) = f^{(j)}(u) = f^{(k)}(u) = \tanh\left(\frac{u}{\varepsilon}\right) \quad (9)$$

where ε is the steepness parameter, and $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ are imaginary units.

Step 3 : Propagation from Other Layers to Layer x

The output of the neuron j in the Key Input Part of the layer x, $\boldsymbol{x}_j^x(t+1)$, is given by

$$\boldsymbol{x}_j^x(t+1) = f\left(\sum_{y \neq x}^{M}\left(\sum_{k=1}^{n^y} \boldsymbol{w}_{jk}^{xy} \boldsymbol{x}_k^y(t)\right) + \boldsymbol{v}\boldsymbol{A}_j\right) \quad (10)$$

where M is the number of layers, n^y is the number of neurons in the Key Input Part of the layer y, $\boldsymbol{w}_{jk}^{xy}$ is the connection weight from the neuron k in the layer y to the neuron j in the layer x, $\boldsymbol{v}$ is the connection weight from the external input, and $\boldsymbol{A}_j$ is the external input (See **2.4**) to the neuron j in the layer x.

The output of the neuron j of the Context Part in the layer x, $\boldsymbol{x}_j^x(t+1)$ is given by

$$\begin{aligned}\boldsymbol{x}_j^x(t+1) \;=\; & f\left(\sum_{y \neq x}^{M}\left(\sum_{k=1}^{n^y} \boldsymbol{w}_{jk}^{xy} \sum_{d=0}^{t} k_m^d \boldsymbol{x}_k^d(t-d)\right)\right.\\ & \left. -\alpha(t) \sum_{d=0}^{t} k_r^d \boldsymbol{x}_j^x(t-d)\right) \quad (11)\end{aligned}$$

where k_m and k_r are damping factors. And, $\alpha(t)$ is the scaling factor of the refractoriness at the time t, and is given by

$$\alpha(t) = a + b \cdot \sin(c \cdot t) \quad (12)$$

where a, b and c are coefficients.

Step 4 : Repeat

Steps 2 and **3** are repeated.

2.4. External Input

In the CQMAM, the external input $\boldsymbol{A}_j$ is always given so that the key pattern does not change into other patterns.

If the pattern is given to the layer x and the initial input does not include noise, we can use the initial input pattern $\boldsymbol{x}_j^x(0)$ as the external pattern. However, since the initial input pattern sometimes includes noise, so we use the following pattern $\hat{\boldsymbol{x}}_j^x(t_{in})$ when the network becomes stable t_{in} as an external input. Here, t_{in} is given by

$$t_{in} = \min\left\{t \;\middle|\; \sum_{j=1}^{n^x}(\hat{\boldsymbol{x}}_j^x(t) - \hat{\boldsymbol{x}}_j^x(t-1)) = 0\right\} \quad (13)$$

where n^x is the number of neurons in the Key Input Part of the layer x. And $\hat{\boldsymbol{x}}_j^x(t)$ is the quantized output of the neuron j in the layer x at the time t.

3. Computer Experiment Results

Figures 2~5 show the relation between damping factors (k_m and k_r) and one-to-many association ability. From these results, we confirmed that the combination of the damping factors k_m and k_r influences the one-to-many association ability.

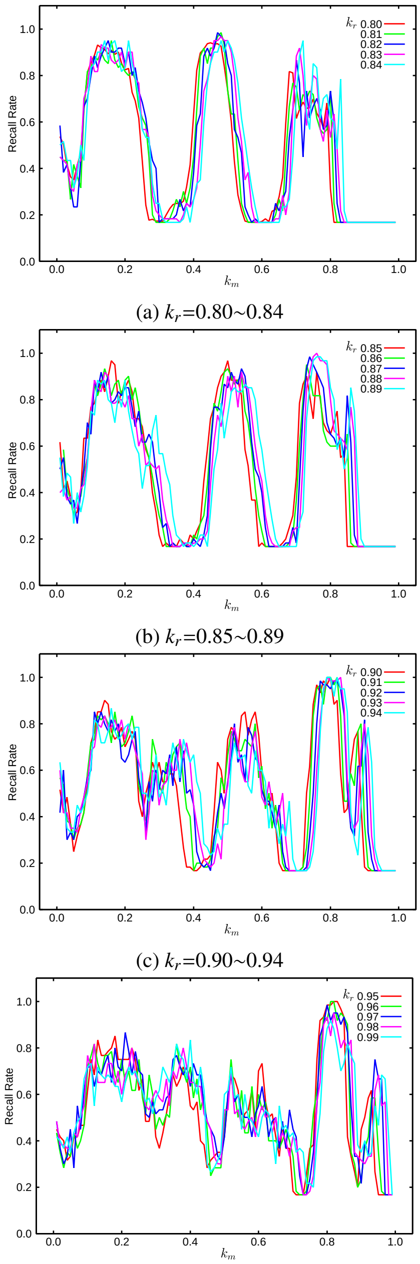
4. Conclusions

In this paper, we investigated the relation between the damping factors (k_m and k_r) and one-to-many association ability of the Chaotic Quaternionic Multidirectional Associative Memory (CQMAM). As a result, we confirmed that the combination of the damping factors k_m and k_r influences the one-to-many association ability.

References

[1] B. Kosko: "Bidirectional associative memories," IEEE Trans. Systems Man and Cybernetics, Vol.18, No.1, pp.49–60, 1988.

[2] M. Hagiwara : "Multidirectional associative memory," Proceedings of IEEE and INNS International



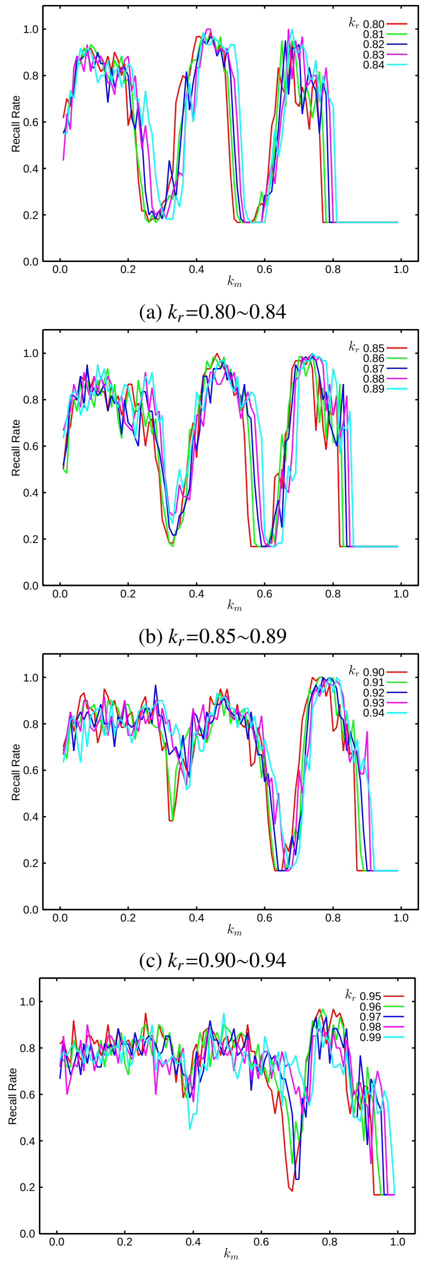
(a) k_r=0.80~0.84

(b) k_r=0.85~0.89

(c) k_r=0.90~0.94

(d) k_r=0.95~0.99

Figure 2: Relation between k_m and One-to-Many Association Ability (3-layered CQMAM).



(a) k_r=0.80~0.84

(b) k_r=0.85~0.89

(c) k_r=0.90~0.94

(d) k_r=0.95~0.99

Figure 3: Relation between k_m and One-to-Many Association Ability (4-layered CQMAM).

Joint Conference on Neural Networks, Washington D.C., Vol.1, pp.3–6, 1990.

[3] K. Aihara, T. Takabe and M. Toyoda: “Chaotic neural networks,” Physics Letter A, 144, No.6, 7, pp.333–340, 1990.

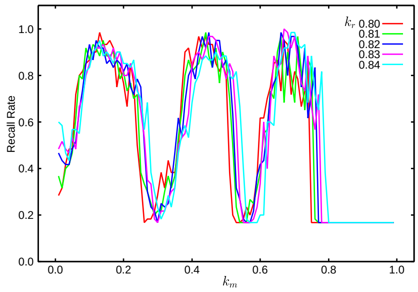
[4] M. Nakada and Y. Osana: “Chaotic complex-valued associative memory,” Proceedings of International Symposium on Nonlinear Theory and its Applications, Vancouver, 2007.

[5] Y. Osana : ”Chaotic quaternionic associative memory,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Brisbane, 2012.

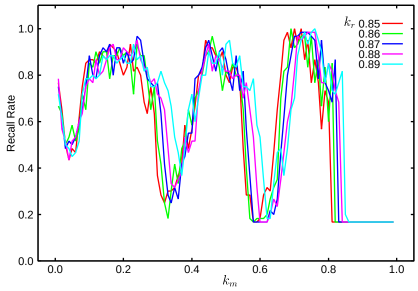
[6] Y. Osana, M. Hattori, and M. Hagiwara: “Chaotic bidirectional associative memory,” Proceedings of IEEE International Conference on Neural Networks, pp.816–821, 1996.

[7] Y. Osana, M. Hattori and M. Hagiwara : “Chaotic multidirectional associative memory,” Proceedings of IEEE International Conference on Neural Networks, Washington D.C., Vol.2, pp.1210–1215, 1997.

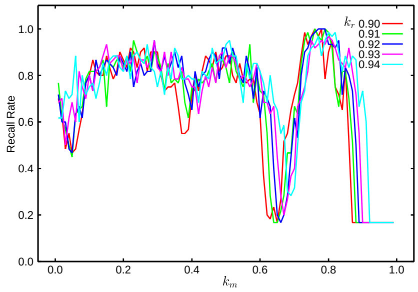
[8] Y. Yano and Y. Osana : “Chaotic complex-valued



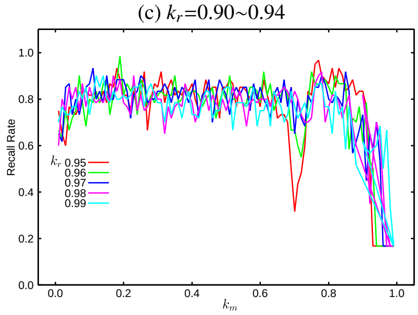
(a) k_r=0.80~0.84



(b) k_r=0.85~0.89

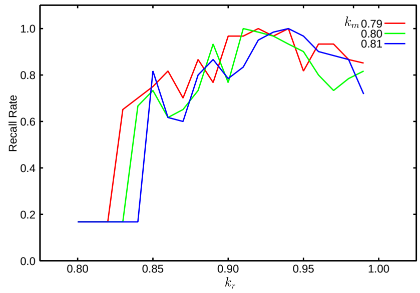


(c) k_r=0.90~0.94

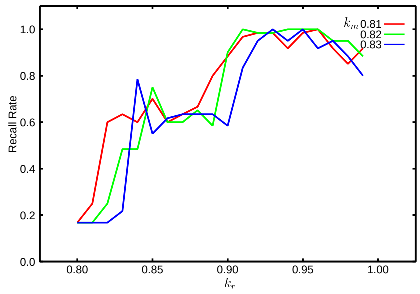


(d) k_r=0.95~0.99

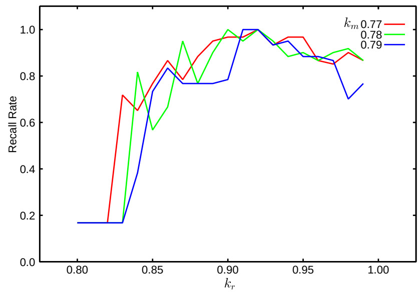
Figure 4: Relation between k_m and One-to-Many Association Ability (5-layered CQMAM).



(a) 3-layered



(b) 4-layered



(c) 5-layered

Figure 5: Relation between k_r and One-to-Many Association Ability.

bidirectional associative memory," Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Atlanta, 2009.

[9] Y. Shimizu and Y.Osana : "Chaotic complex-valued multidirectional associative memory," Proceedings of IASTED Artificial Intelligence and Applications, Innsbruck, 2010.

[10] T. Chino and Y. Osana : "Chaotic complex-valued multidirectional associative memory with adaptive scaling factor," Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Dallas, 2013.

[11] T. Chino and Y. Osana : "Generalization ability of chaotic complex-valued multidirectional associative memory with adaptive scaling factor," Proceedings of International Conference on Neural Information Processing, Daegu, 2013.

[12] Y. Osana : "Chaotic quaternionic associative memory," Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Brisbane, 2012.

[13] T. Okutsu and Y. Osana : "Chaotic quaternionic multidirectional associative memory," Proceedings of International Symposium on Nonlinear Theory and its Applications, Luzern, 2014.

[14] T. Isokawa, H. Nishimura, N. Kamiura and N. Matsui : "Fundamental properties of quaternionic Hopfield neural network," International Journal of Neural Systems, Vol.18, No.2, pp.135-145, 2008.