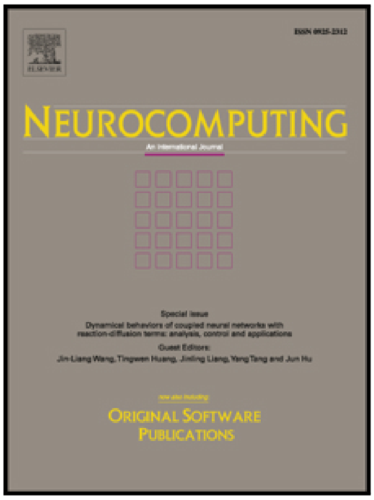
Communicated By He Huang

Accepted Manuscript



Chaotic Pseudo-Orthogonalized Hopfield Associative Memory

Masaki Kobayashi

PII: S0925-2312(17)30307-7
DOI: 10.1016/j.neucom.2017.02.037
Reference: NEUCOM 18103

To appear in: *Neurocomputing*

Received date: 31 July 2016
Revised date: 28 October 2016
Accepted date: 10 February 2017

Please cite this article as: Masaki Kobayashi, Chaotic Pseudo-Orthogonalized Hopfield Associative Memory, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2017.02.037

Chaotic Pseudo-Orthogonalized Hopfield Associative Memory

Masaki Kobayashi

Mathematical Science Center, University of Yamanashi
Takeda 4-3-11, Kofu, Yamanashi 400-8511, Japan

Abstract

A Hopfield associative memory (HAM) is a major model of associative memory using neural networks. The classical HAM model has a low storage capacity. A pseudo-orthogonalized HAM (POHAM) was therefore proposed to improve storage capacity by encoding the training patterns to be pseudo-orthogonal. In the present work, we examine a different property of POHAM. A chaotic HAM (CHAM) can explore embedded patterns, including training patterns. Although it is not desirable to recall patterns other than training patterns, at a minimum, the reversed patterns are recalled. A POHAM regards the training and reversed patterns as equivalent. To take advantage of this property, we propose a chaotic POHAM (CPOHAM). We evaluated a CHAM and a CPOHAM using computer simulations. The CPOHAM never recalled the reversed patterns. In addition, the CPOHAM recalled very few other pseudo-memories, such as mixture patterns, due to the orthogonality of the encoded training patterns, while the CHAM frequently recalled the mixture patterns.

Keywords:
Hopfield associative memory, chaotic neural network,
pseudo-orthogonalization, reversed pattern

1. Introduction

A Hopfield associative memory (HAM) is a major model of associative memory using neural networks[1]. Several extensions of HAM have been proposed. Kosko proposed a bidirectional associative memory (BAM) consisting of two layers[2]. The BAM was then extended to multidirectional associative memories consisting of multiple layers[3, 4, 5]. Multimodule associative memories have also been proposed[6, 7]. Moreover, high-dimensional models of associative memories, such as complex-valued multistate associative memories, have been proposed[8, 9, 10, 11, 12, 13, 14]. In a HAM, the storage capacity and noise tolerance have always been serious issues. Many researchers have tackled these problems[15, 16, 17, 18, 19]. Oku et al.[20] proposed a pseudo-orthogonalized HAM (POHAM) to improve storage capacity. The POHAM introduces randomly generated mask patterns, and encodes the training patterns to be pseudo-orthogonal. A training pattern is masked by a mask pattern, and is encoded to the mask/masked pattern pair. Therefore, a POHAM has twice as many neurons as a HAM. Both the orthogonality of the encoded training patterns and the increase in the number of neurons improve storage capacity. In the present work, we pay attention to a different property of POHAMs.

A chaotic HAM (CHAM) can explore embedded patterns [21, 22, 23, 24, 25, 26]. A HAM stores not only the training patterns but also the reversed patterns[27]. It is desirable to recall only the training patterns in the CHAM recall process, although at the very least, the reversed patterns are recalled. Actually, a CHAM also recalls some patterns other than the training and reversed patterns. A POHAM regards an encoded training pattern and its reversed pattern as equivalent when they are decoded. Therefore, a POHAM never recalls the reversed patterns. We proposed a chaotic POHAM (CPOHAM), a chaotic associative memory that never recalls the reversed patterns. For multilevel data, the different methods, such as reversible code and periodic activation function, were proposed[26, 28]. We performed computer simulations to evaluate the recall process. The CHAM recalled not only the training patterns but also the reversed patterns, as expected. In addition, mixture patterns, which are typical pseudo-memories, frequently appeared. In contrast, the CPOHAM never recalled the reversed patterns, and recalled very few other pseudo-memories.

2. Hopfield Associative Memory

We briefly describe a HAM. We denote the state of neuron i and the connection weight between neuron i and neuron j as x_i and w_{ij}, respectively. The neuron state is ± 1. The connection weights are symmetric, and $w_{ij} = w_{ji}$ holds. The requirement that the connection weights be symmetric ensures the convergence to an embedded pattern in an asynchronous mode.

We denote the pth training pattern $\mathbf{x}^p$ as follows:

$$\mathbf{x}^p = (x_1^p, x_2^p, \cdots, x_N^p), \tag{1}$$

where N is the number of neurons. The connection weights are determined as follows:

$$w_{ij} = \frac{1}{N}\sum_p x_i^p x_j^p. \tag{2}$$

By this learning algorithm, referred to as the Hebbian learning rule, the training patterns are embedded to a HAM. Then, however, the reversed patterns $-\mathbf{x}^p$ are also embedded. In addition, pseudo-memories other than the reversed patterns, referred to as mixture patterns, are embedded.

3. Chaotic Hopfield Associative Memory

Next, we briefly describe a CHAM. A CHAM is a dynamic HAM that uses chaotic neurons. It employs the following activation function:

$$f(x) = \tanh(\beta x), \tag{3}$$

with the steepness parameter β. The dynamics of a CHAM are defined as follows:

$$x_i(t+1) = f\left(\sum_{j\neq i} w_{ij}x_j(t) - \alpha\sum_{d=0}^{t} k^d x_i(t-d)\right), \tag{4}$$

where $x_i(t)$ is the state of neuron i at time t. The parameters $k\,(0 < k < 1)$ and $\alpha > 0$ are the decay and refractory parameters, respectively. A CHAM works in a synchronous mode; all the neurons are simultaneously updated. A CHAM can explore embedded patterns, including training patterns, and is attracted to an embedded pattern. After staying there for a while, it leaves and is attracted to another embedded pattern. A CHAM repeats this cycle and explores many embedded patterns.

A HAM stores not only training patterns but also pseudo-memories, such as reversed patterns. Although it is desirable to recall only the training patterns are recalled, the CHAM also recalls pseudo-memories. We explain the training and reversed patterns based on the patterns used in our computer

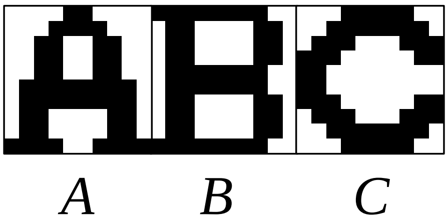


Figure 1: Training patterns used in our computer simulations. White and black pixels correspond to 1 and -1, respectively.

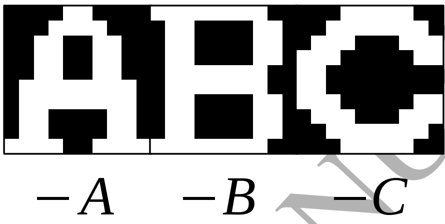


Figure 2: The reversed patterns corresponding to the training patterns. A HAM stores not only the training patterns but also the reversed patterns.

simulations. Fig. 1 illustrates the training patterns. We denote these training patterns as A, B, and C. These pictures are binary images of 10×10 pixels. White and black pixels correspond to 1 and -1, respectively. The reversed patterns invert the white and black pixels of the training patterns, as illustrated in Fig. 2. The reversed patterns corresponding to the training patterns A, B, and C are denoted as $-A$, $-B$, and $-C$, respectively.

4. Chaotic Pseudo-Orthogonalized Hopfield Associative Memory

Oku et al.[20] proposed a POHAM. For each training pattern $\mathbf{x}^p$, a mask pattern $\mathbf{y}^p = (y_1^p, y_2^p, \cdots, y_N^p)$ is randomly generalized, where $y_i^p = \pm 1$. The masked pattern $\mathbf{z}^p$ is then determined as follows:

$$\mathbf{z}^p = (z_1^p, z_2^p, \cdots, z_N^p) \tag{5}$$

$$= (x_1^p y_1^p, x_2^p y_2^p, \cdots, x_N^p y_N^p). \tag{6}$$

A training pattern $\mathbf{x}^p$ is randomized to a masked pattern $\mathbf{z}^p$. In a POHAM, a training pattern $\mathbf{x}^p$ is encoded to a pair $(\mathbf{y}^p, \mathbf{z}^p)$. These encoded patterns are randomized, and are expected to have low correlations. We consider a HAM to store these encoded patterns as a POHAM. A POHAM has the mask neurons $\{y_i\}_{i=1}^N$ and the masked neurons $\{z_i\}_{i=1}^N$. Therefore, a POHAM has $2N$ neurons. A POHAM has a larger storage capacity than a HAM, since the POHAM has twice as many neurons, and the correlations between the encoded training patterns are low. Fig. 3 illustrates the encoding process of a training pattern in a POHAM. By multiplying each component of a training pattern with the corresponding component of a mask pattern, a masked pattern is obtained. The mask/masked pattern pair is the training pattern for a POHAM. Fig. 4 illustrates the training, mask and masked patterns used in our computer simulations. The masked pattern is generated from the training and mask patterns. A CHAM learns the training patterns. A CPOHAM learns the pairs of mask/masked patterns. The storage capacity of a HAM with N neurons is $0.14N$ for random training patterns[27]. However, it is much smaller than $0.14N$ for real data, which are not random data. The corresponding POHAM has $2N$ neurons. Since the training data are randomized, that of the POHAM is $0.28N$.

We describe the decoding in a POHAM. A pattern $(\mathbf{y}, \mathbf{z}) = (y_1, y_2, \cdots, y_N, z_1, z_2, \cdots, z_N)$ of a POHAM is decoded to the pattern $(x_1, x_2, \cdots, x_N)$ as follows:

$$(x_1, x_2, \cdots, x_N) = (y_1 z_1, y_2 z_2, \cdots, y_N z_N). \tag{7}$$

Fig. 5 illustrates the pattern decoding process in a POHAM. By multiplying each mask neuron with the corresponding masked neuron, the decoded pattern $\mathbf{x}$ is obtained. An encoded training pattern $(\mathbf{y}^p, \mathbf{z}^p)$ for a POHAM is decoded as follows:

$$\begin{aligned} & (y_1^p z_1^p, y_2^p z_2^p, \cdots, y_N^p z_N^p) \\ = \; & (y_1^p x_1^p y_1^p, y_2^p x_2^p y_2^p, \cdots, y_N^p x_N^p y_N^p) && (8) \\ = \; & (x_1^p, x_2^p, \cdots, x_N^p). && (9) \end{aligned}$$

We obtained the original training pattern by decoding the encoded training pattern. A reversed pattern $(-\mathbf{y}^p, -\mathbf{z}^p)$ stored in a POHAM is decoded as follows:

$$\begin{aligned} & ((-y_1^p)(-z_1^p), (-y_2^p)(-z_2^p), \cdots, (-y_N^p)(-z_N^p)) \\ = \; & (y_1^p z_1^p, y_2^p z_2^p, \cdots, y_N^p z_N^p) && (10) \\ = \; & (x_1^p, x_2^p, \cdots, x_N^p). && (11) \end{aligned}$$

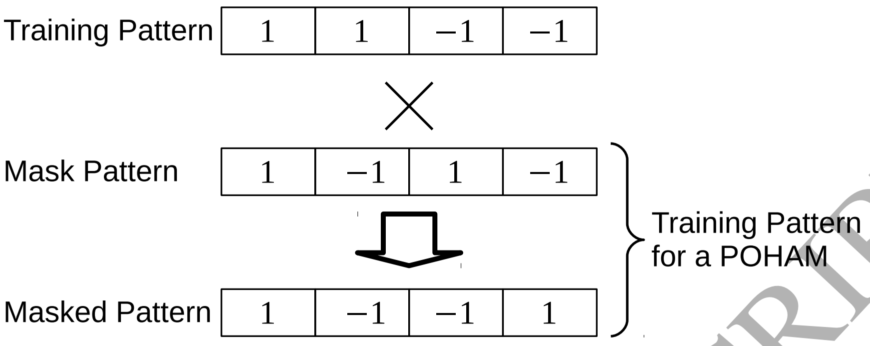


Figure 3: Encoding process. A training pattern is masked by pairing a mask pattern to a masked pattern. The training pattern is encoded to the pair of patterns, which is the training pattern for a POHAM.

In a POHAM, the reversed pattern is decoded to the training pattern. Thus, the training and reversed patterns are regarded as equivalent. Fig. 6 illustrates how the encoded training and reversed patterns are decoded. The upper pair is the encoded pattern of the training pattern A. This pair is decoded to A. The lower pair is the reversed pattern of the encoded pattern of A, which is also decoded to A.

In this study, we propose a chaotic POHAM (CPOHAM) that can take advantage of the fact that the training and reversed patterns are regarded as equivalent. A CPOHAM is a POHAM that employs chaotic neurons. A CPOHAM is expected to have the following advantages:

1. The reversed patterns are not recalled by regarding the encoded training and reversed patterns as equivalent.
2. Encoding the training patterns to be pseudo-orthogonalized means very few mixture patterns are recalled.

5. Computer Simulation

We evaluated a CHAM and a CPOHAM by conducting computer simulations. In these simulations, we employed the training patterns illustrated in Fig. 1. Each training pattern is a binary image of 10×10 pixels. White and black pixels correspond to 1 and -1, respectively. Therefore, the numbers of neurons in the CHAM and CPOHAM were $N = 100$ and 200, respectively. In the CHAM, the reversed patterns, shown in Fig. 2, would be recalled. In

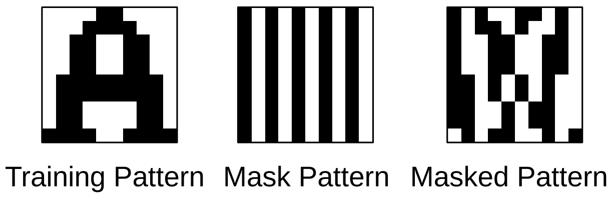


Figure 4: A masked pattern is generated by masking a training pattern with a mask pattern. A training pattern is encoded to the mask/masked pair.

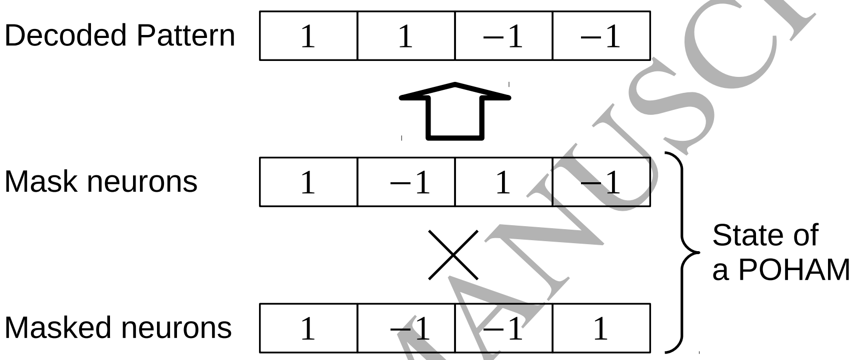


Figure 5: Decoding process. A pair of mask/masked neurons generates a decoded pattern, which is the output of a POHAM.

the CPOHAM, the training patterns A, B and C were masked with the mask patterns MA, MB and MC, respectively, illustrated in Fig. 7. Although randomly generated mask patterns were employed in Oku et al.[20], in this study, we used orthogonal masks to enhance the orthogonality. It is ideal that the encoded training patterns are orthogonal. However, it is difficult to find such mask patterns. Thus, I made the mask patterns orthogonal.

Prior to our computer simulations, we investigated the major pseudo-memories in a HAM and POHAM. In the HAM, we found the pseudo-memories X and $-X$ illustrated in Fig. 8. Each pixel of X is the majority of the pixels in the training patterns A, B, and C. These patterns are typical pseudo-memories, referred to as mixture patterns[27]. In the POHAM, we found the pseudo-memories $Y1$, $Y2$, $Y3$ and $Y4$ illustrated in Fig. 9.

We initialized a CHAM and CPOHAM by setting all the states of neurons to 1. The initial pattern corresponded to a white image. The CHAM and

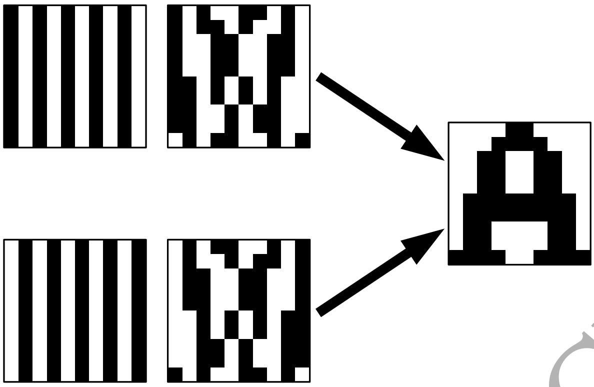


Figure 6: The upper pair is the encoded pattern of the training pattern. The lower pair is the reversed pattern of the encoded pattern. Both pairs are decoded to the training pattern.

α	A	B	C	$-A$	$-B$	$-C$	X	$-X$
0.1	171	140	1	172	145	1	139	142
0.2	73	5	277	74	9	273	136	135

Table 1: Frequencies of major patterns in the recall process of a CHAM.

CPOHAM recalled for 1,000 cycles. The parameters were $k = 0.99$, $\beta = 2$, and $\alpha = 0.1$ and 0.2. We show parts of the recall process in the case of $\alpha = 0.2$. Figs. 10 and 11 show parts of the recall processes in a CHAM and CPOHAM, respectively. White and black pixels indicate whether the corresponding neurons were positive or negative. The numbers above the images indicate image duration. In Fig. 10, both the training patterns and pseudo-memories, including the reversed and mixture patterns, appear frequently. In contrast, few patterns other than the training patterns appear in Fig. 11. In the initial CPOHAM cycles, the pattern C appeared frequently and pattern A only sometimes. Subsequently, the patterns A and B appeared frequently. Meanwhile, the pattern C never appeared. Finally, the pattern C began to appear again. In the recall process of the CHAM, we found the patterns shown in Fig. 12. These patterns appear to be mixture patterns. Tables 1 and 2 show frequencies of recalled patterns in the CHAM and CPOHAM, respectively.

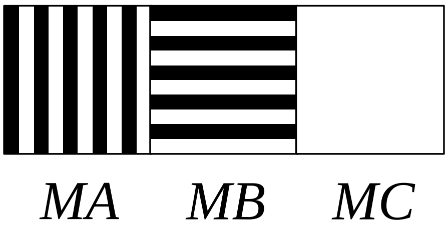


Figure 7: The mask patterns. These mask patterns are orthogonal.

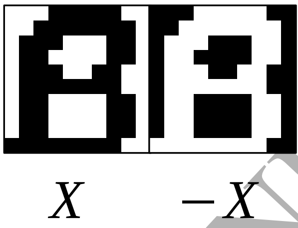


Figure 8: Major pseudo-memories of a CHAM. The pattern X is that determined by the majority of the training patterns A, B, and C. The right pattern, $-X$, is the reversed pattern of X.

6. Discussion

We discuss CHAM and CPOHAM models based on the results of our computer simulations. Although the CHAM frequently recalled reversed patterns, the CPOHAM never recalled such patterns, as expected. In the CPOHAM, no reversed patterns appeared by regarding the encoded training and reversed patterns as equivalent. The total frequency of training and reversed patterns in the CHAM was approximately 700 (Table 1). In contrast, frequency in the CPOHAM was about 950 (Table 2). The CPOHAM reduced the appearance of pseudo-memories better than the CHAM. The

α	A	B	C	$Y1$	$Y2$	$Y3$	$Y4$
0.1	423	256	276	7	12	17	9
0.2	435	213	301	9	15	13	15

Table 2: Frequencies of major patterns in the recall process of a CPOHAM.

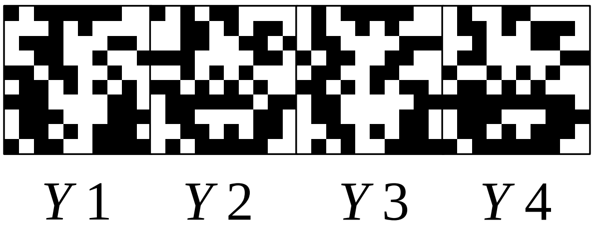


Figure 9: Major pseudo-memories of a CPOHAM.

mixture patterns X and $-X$ were strong attractors in the CHAM, and appeared frequently. The frequency of X and $-X$ was more than 250 in Table 1, while that of Y_1 to Y_4 was less than 60 in Table 2. We propose that the advantages of the CPOHAM include the following:

1. The encoded training and reversed patterns are regarded as equivalent. The reversed patterns are decoded to the training patterns.
2. Since the training patterns are encoded to be pseudo-orthogonalized, there is little recall of the mixture patterns.
3. Since a CPOHAM has twice as many neurons as a CHAM, a CPOHAM has larger storage capacity. In addition, the orthogonality of the encoded training patterns improved storage capacity.

7. Conclusion

A CHAM recalls not only training patterns but also pseudo-memories, such as reversed and mixture patterns. In the present work, we proposed a CPOHAM to solve this problem. Oku et al.[20] previously proposed a POHAM to improve storage capacity. The high orthogonality of the encoded patterns improved the storage capacity. A POHAM encodes training patterns to be pseudo-orthogonalized by appending mask neurons, which increases storage capacity. Through encoding the training patterns, the encoded training and reversed patterns become equivalent. This property has never been considered an advantage of a POHAM. We proposed a CPOHAM that takes advantage of this property. We evaluated a CHAM and CPOHAM using computer simulations. Although the CHAM recalled the reversed patterns, the CPOHAM did not. In addition, pseudo-memories frequently appeared in the recall process of the CHAM, but rarely appeared in the recall process of the CPOHAM.

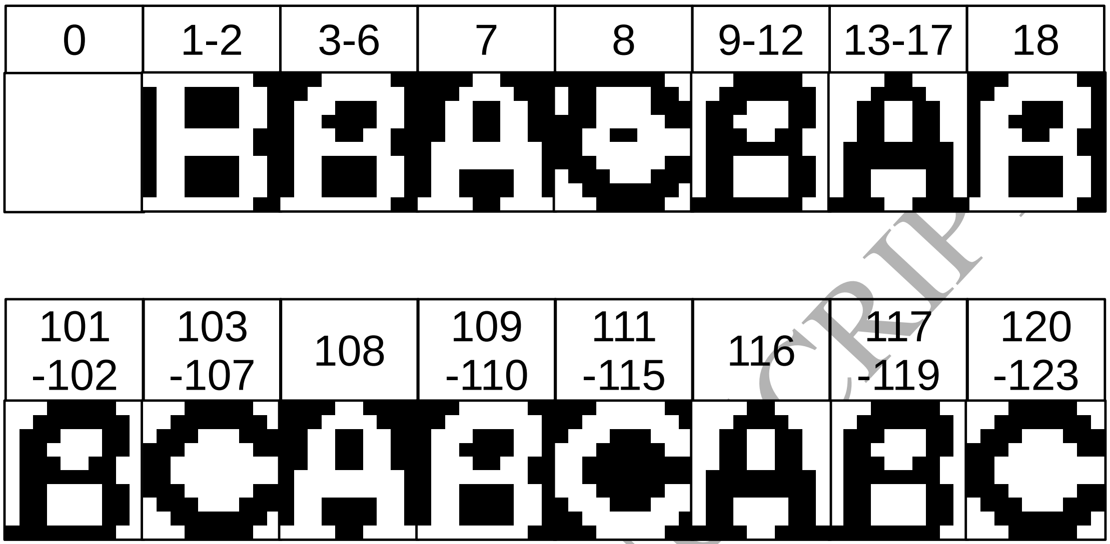


Figure 10: Recall samples of a CHAM. The numbers indicate duration of the patterns' appearance. Both the training patterns and the psuedomemories, such as reversed and mixture patterns, appeared frequently.

Two difficulties remain. The first is the need to determine good parameters α, β, and k. The dynamics of both the CHAM and CPOHAM depends on these parameters. The second is to determine good mask patterns. Although the training patterns are given, we can determine the mask patterns ourselves. If the performance of the orthogonal mask patterns is investigated more in detail in future, systematic generation of binary orthogonal patterns would be important. One approach is to use a Hadamard matrix. It is probable that paired mask patterns, such that pairs of the mask/masked patterns were orthogonal, would be better. Also, projection rules, which provide the storage capacity $N-1$, will be available[29, 30].

We proposed a CPOHAM utilizing a property of POHAMs that has received little attention. We plan to study other applications, such as sequential associative memories using asymmetric weight connections, that can take advantage of this property[26].

References

[1] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci, (1982)2554-2558.

[2] B. Kosko, Bidirectional associative memories, IEEE Transactions on Systems, Man and Cybernetics, 18(1) (1988) 49-60.

[3] M. Hagiwara, Multidirectional associative memory, Proceedings of International Joint Conference on Neural Networks, 1 (1990) 3-6.

[4] M. Kobayashi, M. Hattori and H. Yamazaki, Multidirectional associative memory with a hidden layer, Systems and Computers in Japan, 33(6) (2002)1-9.

[5] M. Kobayashi, A. Nakajima and M. Kitahara, Multidirectional associative memory with two hidden layers, IEEJ Transactions on Electrical and Electronic Engineering, 8(3) (2013) 299-300.

[6] T. Ohsumi, M. Kajiura and Y. Anzai, Multimodule neural network for associative memory, Systems and computers in Japan, 24(13) (1993) 98-108.

[7] M. Hattori and M. Hagiwara, Multimodule associative memory for many-to-many associations, Neurocomputing, 19 (1998) 99-119.

[8] S. Jankowski, A. Lozowski and J. M. Zurada, Complex-valued multistate neural associative memory, IEEE Transactions on Neural Networks, 7(6) (1996) 1491-1496.

[9] M. Kobayashi, Hyperbolic Hopfield neural networks, IEEE Transactions on Neural Networks and Learning Systems, 24(2) (2013) 335-341.

[10] M. Kitahara and M. Kobayashi, Projection rule for rotor Hopfield neural networks, IEEE Transactions on Neural Networks and Learning Systems, 25(7) (2014) 1298-1307.

[11] T.Isokawa, H.Nishimura, N.Kamiura and N.Matsui, Associative memory in quaternionic Hopfield neural network, International Journal of Neural Systems, 18(2) (2008) 135-145.

[12] T.Isokawa, H.Nishimura, and N.Matsui, Commutative quaternion and multistate Hopfield neural networks, Proceedings of IEEE World Congress on Computational Intelligence, (2010) 1281-1286.

[13] M. Kobayashi, Hybrid quaternionic Hopfield neural network, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E98-A(7) (2015) 1512-1518.

[14] M. Kobayashi, Rotational invariance of quaternionic Hopfield neural networks, IEEJ Transactions on Electrical and Electronic Engineering, 11(4) (2016) 516-520.

[15] T. Kohonen, and M. Ruohonen, Representation of associated data by matrix operators, IEEE Transactions on Computers, 100(7) (1973) 701-702.

[16] H. Oh and S. C. Kothari, Adaptation of the relaxation method for learning in bidirectional associative memory, IEEE Transactions on Neural Networks, 5(4) (1994) 576-583.

[17] M. Hattori, M. Hagiwara and M. Nakagawa, Quick Learning for Bidirectional Associative Memory, IEICE Trans. Information and Systems, E77-D(4) (1994) 385-392.

[18] M. Hattori and M. Hagiwara, Intersection learning for bidirectional associative memory, Proceedings of IEEE International Conference on Neural Networks, (1996) 555-560.

[19] D. H. Ackley, G. E. Hinton and T. J. Sejnowski, A learning algorithm for Boltzmann machines, Cognitive science, 9(1) (1985) 147-169.

[20] M. Oku, T. Makino and K. Aihara, Pseudo-orthogonalization of memory patterns for associative memory, IEEE Transactions on Neural Networks and Learning Systems, 24(11) (2013) 1877-1887.

[21] K. Aihara, T. Takabe and M. Toyoda, Chaotic neural networks, Phys. Lett. A, 144 (1990) 333-340.

[22] M. Adachi and K. Aihara, Associative dynamics in a chaotic neural network, Neural Networks, 10(1) (1997) 83-98.

[23] N. Nagao, H. Nishimura and N. Matsui, A neural chaos model of multistable perception, Neural Processing Letters, 12(3) (2000) 267-276.

[24] Y. Osana and M. Hagiwara, Chaotic associative memory for sequential patterns, Journal of Signal Processing, 4(1) (2000) 45-52.

[25] G. He, M. D. Shrimali and K. Aihara, Threshold control of chaotic neural network, Neural Networks, 21(2) (2008) 114-121.

[26] M. Oku, and K. Aihara, Associative dynamics of color images in a large-scale chaotic neural network, Nonlinear Theory and Its Applications, 2(4) (2011) 508-521.

[27] J. Hertz, A. Krogh and R. G. Palmer, Introduction to the theory of neural computation, Addison-Weasley, (1991).

[28] Y. Suzuki, M. Kitahara and M. Kobayashi, Rotor Associative Memory with a Periodic Activation Function, Proc. IEEE World Congress on Computational Intelligence, (2012) 720-727.

[29] L. Personnaz, I. Guyon and G. Dreyfus, Collective computational properties of neural networks: New learning mechanisms, Physical Review A, 34(5) (1986) 4217-4228.

[30] A. N. Michel and D. Liu, Qualitative analysis and synthesis of recurrent neural networks, Marcel Dekker Inc., NY, USA, (2002).

Biography

Masaki Kobayashi is a professor at University of Yamanashi. He received the B.S., M.S., and D.S. degrees in mathematics from Nagoya University in 1989, 1991, and 1996, respectively. He became a research associate and an associate professor at the University of Yamanashi in 1993 and 2006, respectively. He has been a professor since 2014.

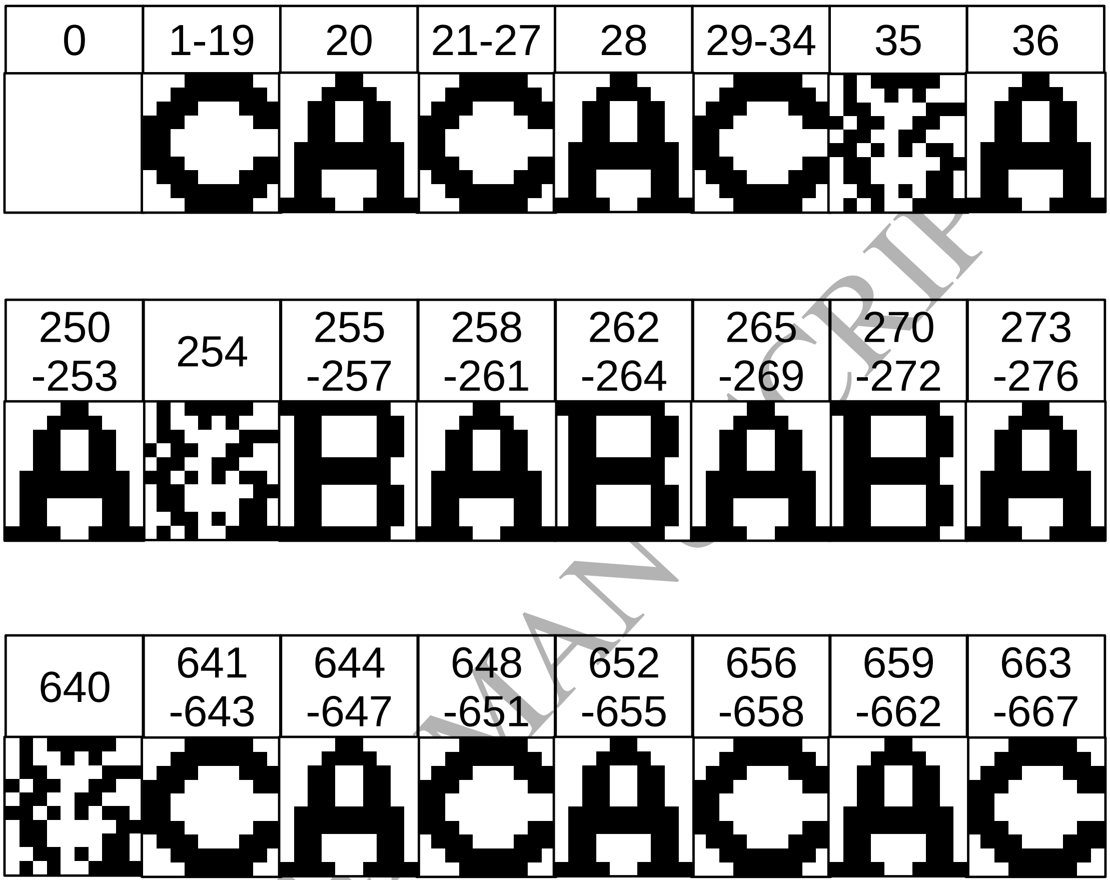


Figure 11: Recall samples of a CPOHAM. The numbers indicate duration of the patterns' appearance. The reversed patterns never appeared. In the initial CPOHAM cycles, the pattern C appeared frequently and pattern A only sometimes. Subsequently, the patterns A and B appeared frequently. Meanwhile, the pattern C never appeared. Finally, the pattern C began to appear again.

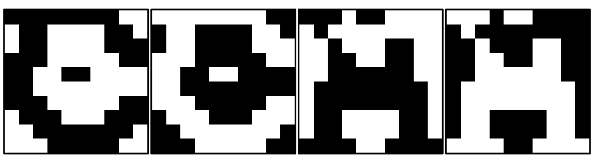


Figure 12: The other patterns in the recall process of a CHAM. These patterns appear to be mixture patterns.