2011 International Symposium on Nonlinear Theory and its Applications
NOLTA2011, Kobe, Japan, September 4-7, 2011

Associative Dynamics with Structured Relations Among Memories in a Chaotic Neural Network

Makito Oku[†] and Kazuyuki Aihara[†,‡]

†Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
‡Institute of Industrial Science, The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
Email: oku@sat.t.u-tokyo.ac.jp

Abstract—
The memories in our minds have associative relations among them, which form a complex network. Such a memory relation network (MRN) may be reflected in spontaneous transitions among brain states during resting and sleeping. Although such transitory dynamics can be described by the chaotic neural network (CNN) model, the influence of an MRN on the dynamics of the CNN model has not yet been investigated. Here, we consider a CNN with two types of connections. The first type is that of the auto-associative connections, which represent the memories themselves, whereas the second type is that of the hetero-associative connections, which represent the relations among these memories. We show numerically that the CNN partially follows a given MRN and chaotically visits the memories in some parameter regions. Our methods provide a way of introducing the MRN to the CNN model without destroying chaos.

1. Introduction

The memories in our minds have associative relations among them, which form a complex network [1, 2]. Here, each memory is considered a node in the network, and the associative relations among these memories correspond to edges. Such a memory relation network (MRN) would be involved in various cognitive tasks such as memory search. The MRN may also influence spontaneous transitions among brain states during resting [3] and sleeping.

The chaotic neural network (CNN) model [4–6] is a neural network model composed of chaotic elements called chaotic neurons. The CNN exhibits chaotic behavior in some parameter regions, which is characterized by irregular transitions among the quasi-attractors. Such transitory dynamics can be used for solving optimization problems [7–14] and for the phenomenological modeling of cognitive dynamics [6, 15–18].

However, the influence of the MRN on the dynamics of the CNN model has not been investigated so far. This is because the network size was small in the previous studies [6, 15, 16], and thus only few memories could be stored in the CNN. Therefore, a complex MRN structure was impracticable.

On the other hand, we have recently developed certain simulation techniques and visualization methods for a large-scale CNN [19–21]. To reduce the computational costs, we have proposed parallel computing and a sparse-connection regime [19]. It then becomes possible to simulate networks composed of more than one million units, a number which is about 10^4 times that of the previous studies [6, 15, 16]. Furthermore, we have also developed several methods to convert color images to binary codes so that the CNN can memorize them [20, 21]. These methods can be used to visualize the CNN's states by mapping them to the corresponding color images.

In this study, we consider a CNN with two types of connections. This network architecture is based on the Kleinfeld model [22]. The first type is that of the auto-associative connections, which represent the memories themselves. The second type is that of the hetero-associative connections, which represent the relations among these memories. In addition, signal propagation through connections of the second type involves some delay. Whereas the MRN in the original study [22] was a simple chain, here we consider a much more complex network that includes "one-to-many associations" [23], i.e., each node in the MRN has multiple outgoing edges.

2. Methods and Models

The CNN is composed of N units, each of which has two internal variables, η_i and ζ_i, and one output variable, y_i. Let us adopt the vector representation $\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_N\}^{\mathrm{T}}$, $\boldsymbol{\zeta} = \{\zeta_1, \ldots, \zeta_N\}^{\mathrm{T}}$, and $\boldsymbol{y} = \{y_1, \ldots, y_N\}^{\mathrm{T}}$. Then, the model dynamics can be described by the following difference equations:

$$\begin{aligned}
\boldsymbol{\eta}(t+1) &= k_f\,\boldsymbol{\eta}(t) + W\boldsymbol{y}(t) + \lambda\, V\boldsymbol{y}(t-\tau), && (1)\\
\boldsymbol{\zeta}(t+1) &= k_r\,\boldsymbol{\zeta}(t) - \alpha\,\boldsymbol{y}(t) + \boldsymbol{a}, && (2)\\
\boldsymbol{y}(t+1) &= \boldsymbol{f}(\boldsymbol{\eta}(t+1) + \boldsymbol{\zeta}(t+1)), && (3)
\end{aligned}$$

where $W = (w_{ij})$ and $V = (v_{ij})$ denote the $N\times N$ weight matrices for the auto-associative connections and the hetero-associative connections, respectively; $0 \le k_f, k_r \le 1$, the decay constants; $\lambda \geq 0$, the strength of the hetero-associative connections; $\tau \in \mathbb{N}$, the synaptic delay; $\alpha \geq 0$, the strength of the refractoriness; and $\boldsymbol{a} = \{a_1, \ldots, a_N\}^{\mathrm{T}}$, the bias that includes the time-invariant external input and the threshold. The activation function $\boldsymbol{f}$ is an operation that applies the logistic function $f(x) = (1 + \exp(-x/\epsilon))^{-1}$ to each element of the argument vector, where ϵ denotes the steepness parameter.

The auto-associative weight matrix W represents the memories themselves in the same manner as that in the conventional associative memory model [24]. On the other hand, the hetero-associative weight matrix V represents the directional relations among the memories in the same manner as those in the sequential associative memory model [22, 25, 26] and the bidirectional associative memory model [27]. Here, the memories are K binary patterns $\boldsymbol{s}^k = \{s_1^k \ldots, s_N^k\}^{\mathrm{T}}$ ($k = 1, \ldots, K$, $s_i^k \in \{-1, 1\}$). For simplicity, we assume that each pattern contains an equal number of 1's and -1's. Then, the matrices W and V are determined by the correlation matrices of the patterns as follows:

$$W = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{s}^k (\boldsymbol{s}^k)^{\mathrm{T}}, \tag{4}$$

$$V = \frac{1}{|S|} \sum_{e_{kl} \in S} \boldsymbol{s}^k (\boldsymbol{s}^l)^{\mathrm{T}}, \tag{5}$$

where S is the edge set of the MRN and e_{kl} is the edge from node l to node k.

We remove a part of connections from the CNN to reduce the computational costs. Each unit receives projections only from L units that are selected at random. All other connections are removed. In other words, we set the synaptic weights of these connections to 0.

To visualize the CNN's state, we use the color images shown in Fig. 1A as the memory patterns. We also use the MRN shown in Fig. 1B. The MRN has no self-loops and no multiple edges, and every node has two incoming edges and two outgoing ones.

The color images are converted to binary patterns so that the CNN can memorize them. The *RGB* (red, green, and blue) values are represented by integer values from 0 to 255. We convert each value to an 8-bit binary segment using the reversible code discussed in [21]. These binary segments of 1 and 0 are transformed to those of 1 and -1. Finally, we concatenate all the binary segments into a single binary pattern of length $N = 8 \times 3 \times 128^2 = 393\,216$. The reversible code guarantees that the statistics of the long binary patterns are effectively the same with those of random patterns.

The CNN output is decoded in the opposite manner. The analog outputs of the units are quantized, i.e., $\tilde{y}_i(t) = u(y_i(t) - 0.5)$, where u is the step function. The quantized data are interpreted by once again using the reversible code, and the corresponding color images are reconstructed. Because the CNN memorizes not only the given binary patterns but also their reversed ones, $-\boldsymbol{s}^1, \ldots, -\boldsymbol{s}^K$ in Eq. 4, the reversible code guarantees that any binary pattern and its reversed one are decoded to the same color image.

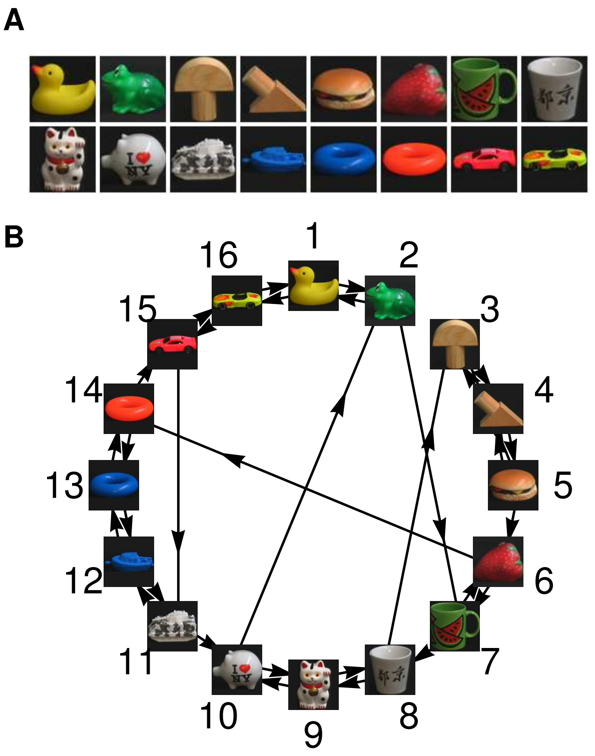

Figure 1: (**A**) The 16 color images (128×128 pixels, the 24-bit RGB mode) stored in the CNN. The images are sampled from the Columbia Object Image Library (COIL-100) [28]. (**B**) Memory relation network represented by the hetero-associative connections.

We also introduce an overlap, $m^k(t)$ ($k = 1, \ldots, K$), to characterize the CNN's state quantitatively, as follows:

$$m^k(t) = 1 - \frac{1}{N} \sum_{i=1}^{N} \left| \frac{s_i^k + 1}{2} - \tilde{y}_i(t) \right|, \tag{6}$$

where s_i^k denotes the ith element of the kth binary pattern. $m^k(t)$ takes 1 when the CNN retrieves the kth pattern completely, whereas it takes 0 when the reversed kth pattern is retrieved.

We perturb the CNN slightly to facilitate chaotic transitions among the memory patterns. The perturbation is only applied at the point where the CNN's state is maximally disordered. This is because, at that point, the sensitivity to perturbation is high and the visual effect induced by it is negligible. To evaluate the order of the CNN, we use the following quasi-energy function $QE(t)$:

$$QE(t) = -\frac{1}{2} \boldsymbol{y}(t)^{\mathrm{T}} W \boldsymbol{y}(t) - (\boldsymbol{a} + \lambda V \boldsymbol{y}(t - \tau))^{\mathrm{T}} \boldsymbol{y}(t). \tag{7}$$

If $QE(t-2) < QE(t-1)$ and $QE(t-1) > QE(t)$, a perturbation is applied (see Fig. 2). To avoid small peaks in $QE(t)$,

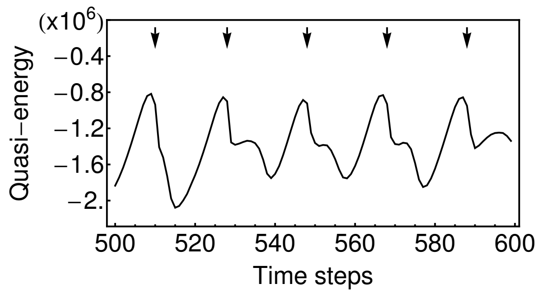

Figure 2: Time series of quasi-energy $QE(t)$. Arrows indicate perturbation times.

we set a waiting period of t_{wp} steps after each perturbation during which another perturbation is prohibited. To apply a perturbation, we multiply all the internal variables by a scholar value $r \in \mathbb{R}$, i.e., $\boldsymbol{\eta}(t) \leftarrow r\,\boldsymbol{\eta}(t)$ and $\boldsymbol{\zeta}(t) \leftarrow r\,\boldsymbol{\zeta}(t)$.

3. Simulations and Results

In the following simulations, we set $k_f = 0.8$, $k_r = 0.9$, $\alpha = 12$, $\epsilon = 0.015$, $L = 480$, $\lambda = 0.1$, $\tau = 10$, $r = 0.25$, and $t_{\text{wp}} = 10$. a_i follows a uniform distribution in $[2, 4]$. As an initial condition, $\eta_i(0)$ follows a uniform distribution in $[0, 1]$, and $\zeta_i(0) = 0$. The simulations are run on a cluster of four Linux server machines, each having two 3.0-GHz single-core processors and 2.0-GB RAM. The computation of one step of the simulation takes approximately 1 s.

Figure 3 shows a typical time series of the decoded CNN output, where many stored images appear successively. The quasi-energy $QE(t)$ is low when any stored image is apparent, whereas it is high when a noisy image is seen.

We use the following criteria to detect the memory retrievals. If $m^k(t) > 0.8$ or $m^k(t) < 0.2$, then we decide that the kth memory is retrieved at time t. More than two memories cannot be retrieved simultaneously. Figure 4 shows a plot of the retrieval times of the memories, where the order of memory recall is not periodic, but some sort of structure is also seen such as transitions among consecutive memories.

Figure 5 shows the actual transition frequencies among the memories. The observed transitions are divided into two types based on whether or not the transition is consistent with the MRN shown in Fig. 1B. We can see that most of the observed transitions are consistent with the given MRN. Furthermore, most edges in the MRN are realized at least once, and only few edges are unrealized.

We also investigate the effect of the strength of the hetero-associative connections λ, as shown in Fig. 6. We can see that consistent transitions become dominant as λ increases. However, unrealized transitions also become dominant with large λ. This is because all the transitions are limited in a subnetwork of the MRN that has only few nodes and edges.

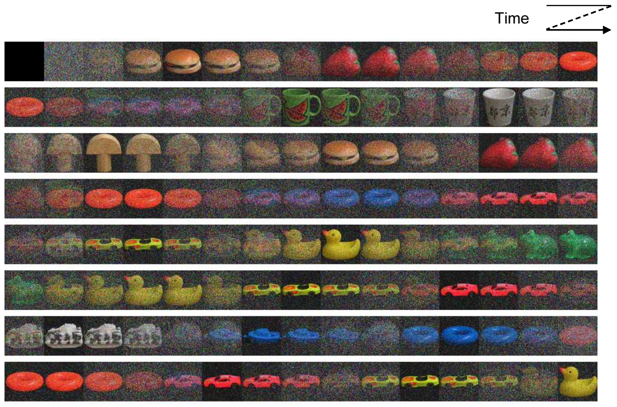

Figure 3: Example of time series of the decoded CNN output displayed with 4-step intervals.

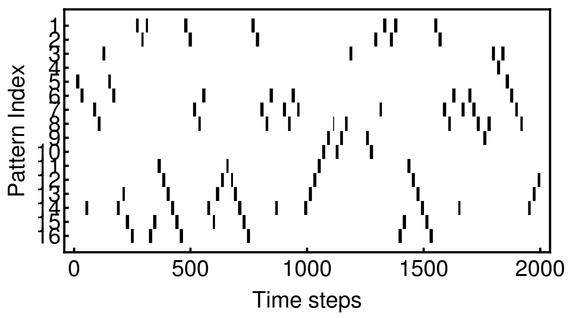

Figure 4: Retrieval times of the memories.

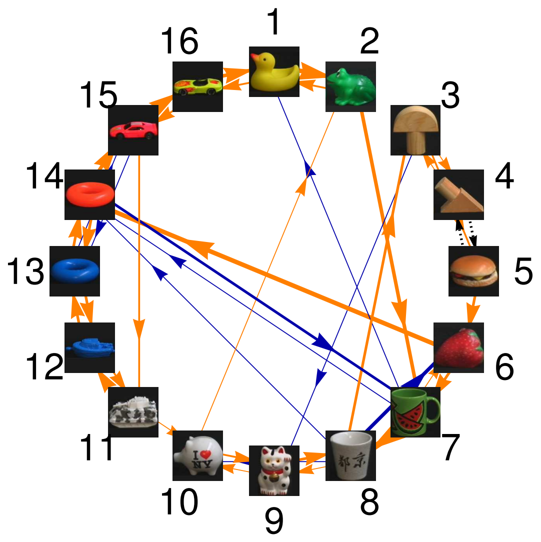

Figure 5: Transition frequencies among the memories observed in 2000 steps. The size of the arrows is proportional to the frequency. Orange, blue, and dotted arrows indicate consistent, inconsistent, and unrealized transitions, respectively.

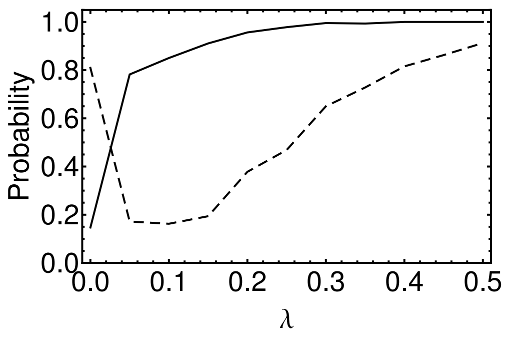

Figure 6: Effect of the strength of the hetero-associative connections λ. Solid curve shows the proportion of consistent transitions among all observed transitions. Dashed curve shows the proportion of unrealized transitions among all edges in the MRN. Averaged over 10 simulation trials of 2000 steps.

4. Discussion and Conclusions

We have proposed a way of introducing the MRN to the CNN model. The CNN has two types of connections, which represent the memories themselves and the directional relations among these memories. The MRN includes one-to-many associations, and the chaotic dynamics chooses a memory pattern among the candidate memories for the next transition. We have shown numerically that the CNN partially follows the given MRN and chaotically visits the memories in some parameter regions. Our future work is to analyze the chaotic dynamics in detail and characterize it quantitatively.

Acknowledgments

This research is partially supported by the Japan Society for the Promotion of Science, a Grant-in-Aid for JSPS Fellows (21·937) and the Aihara Project, the FIRST program from JSPS, initiated by CSTP.

References

[1] A. Collins, M. Quillian *J. Verb. Learn. Verb. Behav.*, 8(2):240–247, 1969.

[2] Y. Miyashita *Science*, 306(5695):435–440, 2004.

[3] J. Ito, A. R. Nikolaev, C. van Leeuwen *Hum. Brain Mapp.*, 28(9):904–913, 2007.

[4] K. Aihara, “Chaotic neural networks,” in *Bifurcation phenomena in nonlinear systems and theory of dynamical systems* (H. Kawakami, ed.), pp. 143–161, Singapore: World Scientific, 1990.

[5] K. Aihara, T. Takabe, M. Toyoda *Phys. Lett. A*, 144(6–7):333–340, 1990.

[6] M. Adachi, K. Aihara *Neural Netw.*, 10(1):83–98, 1997.

[7] T. Yamada, K. Aihara, M. Kotani in *Proc. IJCNN'93*, vol. 2, pp. 1549–1552, 1993.

[8] L. Chen, K. Aihara *Neural Netw.*, 8(6):915–930, 1995.

[9] M. Hasegawa, T. Ikeguchi, K. Aihara *Phys. Rev. Lett.*, 79(12):2344–2347, 1997.

[10] I. Tokuda, T. Nagashima, K. Aihara *Neural Netw.*, 10(9):1673–1690, 1997.

[11] M. Hasegawa, T. Ikeguchi, K. Aihara *Neural Netw.*, 15(2):271–283, 2002.

[12] Y. Horio, T. Ikeguchi, K. Aihara *Neural Netw.*, 18(5-6):505–513, 2005.

[13] Y. Horio, K. Aihara *Physica D*, 237(9):1215–1225, 2008.

[14] T. Ikeguchi, S. Motohashi, T. Matsuura in *Proc. NOLTA'10*, pp. 293–296, 2010.

[15] Z. Wang, H. Fan, K. Aihara *Int. J. Bifurc. Chaos*, 17(9):3085–3097, 2007.

[16] G. He, M. D. Shrimali, K. Aihara *Neural Netw.*, 21(2–3):114–121, 2008.

[17] N. Nagao, H. Nishimura, N. Matsui *Neural Proc. Lett.*, 12(3):267–276, 2000.

[18] Y. Kakimoto, K. Aihara *New Math. Nat. Comput.*, 5(1):123–134, 2009.

[19] M. Oku *et al. IEICE Tech. Rep.*, 109(269):55–59, 2009. (in Japanese).

[20] M. Oku, K. Aihara in *Proc. NOLTA'10*, pp. 465–468, 2010.

[21] M. Oku, K. Aihara *NOLTA, IEICE*, 2011. (accepted).

[22] D. Kleinfeld *Proc. Natl. Acad. Sci. U.S.A.*, 83(24):9469–9473, 1986.

[23] Y. Osana, M. Hattori, M. Hagiwara in *Proc. ICNN'96*, vol. 2, pp. 816–821, 1996.

[24] J. J. Hopfield *Proc. Natl. Acad. Sci. U.S.A.*, 79(8):2554–2558, 1982.

[25] S. Amari *IEEE Trans. Comput.*, c-21(11):1197–1206, 1972.

[26] H. Sompolinsky, I. Kanter *Phys. Rev. Lett.*, 57(22):2861–2864, 1986.

[27] B. Kosko *IEEE Trans. Syst. Man Cybern.*, 18(1):49–60, 1988.

[28] S. Nene, S. Nayar, H. Murase, “Columbia Object Image Library (COIL-100),” Tech. Rep. CUCS-006-96, Department of Computer Science, Columbia University, 1996.