Journal of Robotics, Networking and Artificial Life, Vol. 1, No. 1 (June 2014), 73-79

The Recollection Characteristics of Generalized MCNN Using Different Control Methods

Shun Watanabe, Takashi Kuremoto*, Shingo Mabu, Masanao Obayashi
Graduate School of Science and Engineering, Yamaguchi University
Ube, Yamaguchi, 755-8611, Japan
E-mail: {s005we, wu, mabu, m.obayas}@yamaguchi-u.ac.jp

Kunikazu Kobayashi
School of Information Science and Technology, Aichi Prefectural University
Nagakute, Aichi, 480-1198, Japan
E-mail: kobayashi@ist.aichi-pu.ac.jp

**Corresponding author. E-mail: wu@yamaguchi-u.ac.jp*
Tel: +81-836-85-9520

Abstract

Kuremoto et al. proposed a multi-layer chaotic neural network (MCNN) combined multiple Adachi et al.'s CNNs to realize mutual auto-association of plural time series patterns. However, the MCNN was limited in a two-layer model. In this paper, we extend the MCNN to be a general form (GMCNN) with more layers and use particle swarm optimization (PSO) to improve the recollection performance of GMCNN. The recollecting characteristics by different parameter-control methods were investigated by computer simulations.

Keywords: chaotic neural network, association memory, time-series pattern, particle swarm optimization.

1. Introduction

Hopfield model has characteristics of a transfer of network energy to a stable state and an ability to recollect a stored pattern stably[1]. On other hand, a chaotic neural network (CNN) proposed by Aihara et al. is able to recollect multiple stored patterns dynamically because CNN consists of neurons which internal states perform unstably and chaotically[2,3]. Generally, chaotic neural network models proposed in the 1990s are known as their recollection characteristics of "chaotic itinerancy", which means stored patterns are recollected one by one in aperiodic states[2-4]. CNNs are applied to not only dynamic association memory[3] but also optimization[5] and complex computing[6]. Additionally, Kuremoto et al. have combined multiple CNNs to be a multi-layer chaotic neural network (MCNN)[7-11]. MCNN is able to recollect mutually multiple time series patterns by controlling the unstable behaviors of CNN. Control methods for MCNN are proposed by various approaches, e.g., by observing the amount of dynamical state changes[9,10], or upper and lower limit of recalling times[12,13], and meta-heuristics (e.g., genetic algorithm, particle swarm optimization (PSO))[14]. MCNN has been applied to a mathematical model of hippocampus to realize a transform of long-term memory[8-11]. However, in these works, the number of layers in MCNN was limited to two layers. So we modified MCNN to be a generalized MCNN (GMCNN) and have investigated recollect characteristics of GMCNN using direct control method (DC)[15]. However, characteristics of GMCNN using other control methods was not investigated and compared.

In this paper, a meta-heuristics control method using PSO (PSOC) for GMCNN was proposed. Computational simulations of 2, 3, and 4 layers GMCNN were reported and the recollection characteristics of GMCNNs using DC[15] and PSOC were compared by the simulation results.

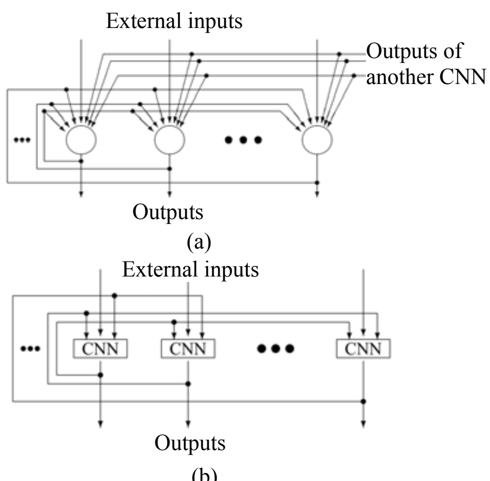


Fig. 1 Association memory models: (a) a chaotic neural network (CNN)[2,3], (b) a generalized multi-layer chaotic neural network (GMCNN)[7-11].

2. Association Memory Model

In this section, a generalized multi-layer chaotic neural network (GMCNN) which consists of plural chaotic neural networks (CNNs)[7-11] of Aihara et al. and its storage/recollection methods of multiple time series patterns are introduced.

2.1. *Generalized Multi-layer Chaotic Neural Network*

GMCNN is structured by interconnecting chaotic neurons of CNN layers which have same network structures[15]. In Fig.1 (a), a CNN is shown and in Fig.1 (b) a GMCNN which has multiple CNN layers is shown. The chaotic neurons in CNN layer connect to each other as complete connected recurrent network as same as Hopfield network[1]. CNN layers in Fig.1 (b) connect to each other and play either role of "display layer" or "association layer". "display layer" means a state of the layer keeps static. "association layer" means the layer output patterns dynamically. GMCNN is able to recollect multiple time series patterns by switching the roles between CNN layers. The dynamics of the chaotic neuron of CNN layer in GMCNN is defined as follows:

$$
\begin{aligned}
\eta_i^{\langle n\rangle}(t+1) &= k_1^{\langle n\rangle}\eta_i^{\langle n\rangle}(t) + \sum_{j=1}^{N} w_{ij}^{\langle n,n\rangle} x_j^{\langle n\rangle}(t) \\
\zeta_i^{\langle n\rangle}(t+1) &= k_2^{\langle n\rangle}\zeta_i^{\langle n\rangle}(t) - k_3^{\langle n\rangle} x_i^{\langle n\rangle}(t) + a_i^{\langle n\rangle} \\
\xi_i^{\langle n\rangle}(t+1) &= k_4^{\langle n\rangle}\xi_i^{\langle n\rangle}(t) + \sum_{m\neq n}^{M}\sum_{j=1}^{N} w_{ij}^{\langle n,m\rangle} x_j^{\langle m\rangle}(t) \\
y_i^{\langle n\rangle}(t+1) &= \eta_i^{\langle n\rangle}(t+1) + \zeta_i^{\langle n\rangle}(t+1) + \xi_i^{\langle n\rangle}(t+1)
\end{aligned}
\qquad (1)
$$

$$
x_i^{\langle n\rangle}(t+1) = \begin{cases} f\left(y_i^{\langle n\rangle}(t+1)\right) & \text{if } \quad R^{\langle n\rangle}(t), \\ x_i^{\langle n\rangle}(t) & \text{otherwise} \end{cases} \qquad (2)
$$

where, t is an association time, $n, m \in \{1,\ldots,M\}$ is CNN layer number, $i, j \in \{1,\ldots,N\}$ is neuron number of CNN layer, M is the number of CNN layers in GMCNN, N is the number of the neurons of one CNN layer, $\eta_i^{\langle n\rangle}(t), \zeta_i^{\langle n\rangle}(t), \xi_i^{\langle n\rangle}(t)$ are internal states for feedback inputs from self-layer, refractoriness and inputs from another layers respectively, $w_{ij}^{<n,m>}$ is a connection weight from j th neuron of m th layer to i th neuron of n th layer, $k_1^{\langle n\rangle}, k_2^{\langle n\rangle}, k_4^{\langle n\rangle}$ are decay parameters for feedback inputs from self-layer, refractoriness and inputs from another layers respectively, $k_3^{\langle n\rangle}$ is a refractory scaling parameter, $a_i^{\langle n\rangle}$ is a summation of threshold and external input, γ is a rate of effectiveness from another layers, $y_i^{\langle n\rangle}(t)$ is an internal state, $\mathbf{x}^{\langle n\rangle} = \{x_1^{\langle n\rangle}(t),\ldots,x_N^{\langle n\rangle}(t)\}$ is an output of CNN, $R^{\langle n\rangle}(t)$ is a propositional function (n th CNN layer performs as "association layer" if $R^{\langle n\rangle}(t)$ is fulfilled, otherwise the layer performs as "display layer"). $f(\cdot)$ is a output function of the neuron:

$$
f(y) = \frac{1}{1+e^{-y/\varepsilon}} \qquad (3)
$$

where, ε is a steepness parameter. $\mathbf{k}^{\langle n\rangle} = \{k_1^{\langle n\rangle},\ldots,k_4^{\langle n\rangle}\}$ are called internal parameters in this paper.

2.2. *Storage Process of Time Series Patterns*

GMCNN stores each partial pattern of plural time series patterns into the multiple CNN layers alternately. Hebbian learning rule[16] for storing the plural time series patterns are defined as follows:

$$\begin{aligned} g(t_l) &= t_l \bmod M + 1 \\ \tau^{\langle m\rangle}(t_l) &= M\left\lfloor \frac{t_l - m + 1}{M} \right\rfloor + m - 1 \\ \delta_{ij}\left(t_l, \tau^{\langle m\rangle}(t_l)\right) &= \beta\left(2\chi_i(t_l) - 1\right)\left(2\chi_j\left(\tau^{\langle m\rangle}(t_l)\right) - 1\right) \\ \Delta w_{ij}^{\langle n,m\rangle}(t_l) &= \begin{cases} \mathbf{0} & \textbf{if} \quad n \neq g(t_l), \\ \mathbf{0} & \textbf{if} \quad n = m \wedge i = j, \\ \delta_{ij}\left(t_l, \tau^{\langle m\rangle}(t_l)\right) & \textbf{otherwise} \end{cases} \end{aligned} \tag{4}$$

$$w_{ij}^{\langle n,m\rangle} \leftarrow w_{ij}^{\langle n,m\rangle} + \Delta w_{ij}^{\langle n,m\rangle}(t_l) \tag{5}$$

where, t_l is a storage time, $\chi(t_l) = \{\chi_1(t_l), \ldots, \chi_N(t_l)\}$ is a partial pattern of the time series patterns at t_l, $g(t_l)$ is the layer number to store a pattern, $\tau^{\langle m\rangle}(t_l)$ is a storage time for m th layer at t_l, $\lfloor\cdot\rfloor$ is floor function, β is a storage rate, $\Delta w_{ij}^{\langle n,m\rangle}(t_l)$ is a correction mass of the weight, Eq.(5) is an equation to renew state at t_l.

2.3. *Network Energy*

GMCNN has a characteristic of transfer network energy (NE) to a local minimal state when the network outputs its stored pattern (also called "recollection")[1] because GMCNN store the patterns by Hebbian learning rule (which is also used in Hopfield network[1], CNN[2,3], and MCNN[8-11]). In this paper, the equation of NE is defined as a follow:

$$E(t) = -\frac{1}{2}\sum_{n=1}^{M}\sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}^{\langle n,m\rangle} x_i^{\langle n\rangle}(t) x_j^{\langle m\rangle}(t) \tag{6}$$

3. Control Methods

The meta-heuristics control method using PSO (PSOC) is proposed for controlling the state of each CNN layer and switching the roles of "display layer" and "association layer" for GMCNN. The controlled the parameter of CNN layer are $\mathbf{k}^{\langle n\rangle}$. The propositional function $R^{\langle n\rangle}(t)$ of Eq.(5) is defined as $n = n'(t)$ in order to make only one CNN layer as "association layer" at all times. $n'(t)$ is the layer number as "association layer" at time t and is defined a follow:

$$n'(t+1) = \begin{cases} n'(t) \bmod M + 1 & \text{if} \quad Q(t) \\ n'(t) & \text{otherwise} \end{cases} \tag{7}$$

where, $Q(t)$ is a propositional function for switching the role of CNN layer to be "display layer" or "association layer" at time t. The output of GMCNN for observing the output of "association layer" is defined as a follow:

$$z_i(t) = \begin{cases} 1 & \text{if} \quad x_i^{\langle n'(t)\rangle}(t) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The direct control method (DC)[15] controls "association layer" in GMCNN by switching the internal parameters $\mathbf{k}^{\langle n'(t)\rangle}$ to different parameters from the amount of decrease of state changes of the layer. However, the switching parameters were set arbitrarily. In this paper, optimal parameters are found by a meta-heuristic method described in the next subsection and they are used to "association layer" to control the layer.

3.1. *Particle Swarm Optimization Control Method (PSO)*

Particle swarm optimization (PSO) is a meta-heuristic method base on an exploratory behavior of a group of birds or fishes proposed by Kennedy and Eberhart[17,18]. An exploration of optimal parameters by PSO is run according to a joint ownership parameter in a search space. Here, PSO is used to find the optimal parameters of each CNN layer in GMCNN. In Refs.14, Kuremoto et al. used PSO to find the optimal parameters of MCNN, however, to evaluate fitness of each particle, it is used stored patterns during recalling process. To eliminate this supposition, the stability of NE given by Eq.(6) can be used to evaluate the fitness of each particle. An algorithm for the control method using PSO named "PSOC" is shown as follow:

[Algorithm for PSOC]

Step 1 Initializations of the particles are defined as follows:

$$\begin{aligned} s &\leftarrow 0 \\ u_{pd}(s) &\leftarrow U\left(u_d^{\langle\min\rangle}, u_d^{\langle\max\rangle}\right) \\ v_{pd}(s) &\leftarrow U\left(v_d^{\langle\min\rangle}, v_d^{\langle\max\rangle}\right) \end{aligned} \tag{9}$$

Step 2 $\mathbf{x}^{\langle n'(t)\rangle}(t+1)$ is calculated by Eq. (1)-(2) using $\mathbf{k}^{\langle n'(t)\rangle}$ given by $\mathbf{u}_p(s)$. The fitness of p th particle is defined as follows:

$$\begin{aligned} \Delta E\left(t, \mathbf{u}_p(s)\right) &= E(t) - E(t+1)\Big|_{\mathbf{k}^{\langle n'(t)\rangle} \leftarrow \mathbf{u}_p(s)} \\ F\left(\mathbf{u}_p(s)\right) &= \max\left\{\Delta E\left(t, \mathbf{u}_p(s)\right), 0\right\} \end{aligned} \tag{10}$$

Step 3 Equations for renewing the particles are defined as follows:

$$\begin{aligned} v_{pd}(s+1) \leftarrow{} & \omega u_{pd}(s) + c_1 r_1\left(\hat{u}_{pd}(s) - u_{pd}(s)\right) \\ & + c_2 r_2\left(\tilde{u}_d(s) - u_{pd}(s)\right) \\ u_{pd}(s+1) \leftarrow{} & u_{pd}(s) + v_{pd}(s+1) \end{aligned} \tag{11}$$

Step 4 The particles are reevaluated by Eq.(10).

Step 5 This process is repeated from Step 3 if an iteration step s fulfills $s < s^{\langle \max \rangle}$. Otherwise, this process ends and "association layer" is renewed by Eq. (1)-(2) ($t \leftarrow t+1$) after $\tilde{\mathbf{u}}(s)$ is applied to $\mathbf{k}^{\langle n'(t) \rangle}$ ($\mathbf{k}^{\langle n'(t) \rangle} \leftarrow \tilde{\mathbf{u}}(s)$). $Q(t)$ of Eq. (7) is defined as $E(t) \le E(t+1)$.

where, $U(a,b)$ is an uniform random number in a range $[a,b]$, $u_d^{\langle \min \rangle}$ and $u_d^{\langle \max \rangle}$ are the minimum value and the maximum value in a d -dimensional range $\left[u_d^{\langle \min \rangle}, u_d^{\langle \max \rangle}\right]$, $v_{pd}(s)$ is a velocity of $d \in \{1, \ldots, D\}$ th parameter of $p \in \{1, \ldots, P\}$ th particle at step s, $E(t+1)\big|_{\mathbf{k}^{\langle n'(t) \rangle} \leftarrow \mathbf{u}_p}$ is the NE of Eq.(6) in "association layer" renewed by Eq.(1)-(2) in a case that parameters $\mathbf{u}_p(s) = \{u_{p1}(s), \ldots, u_{pD}(s)\}$ of p th particle at step s are applied to $\mathbf{k}^{\langle n'(t) \rangle}$ of the layer at time t, $F(\mathbf{u}_p(s))$ is the fitness of p th particle, ω is an inertia coefficient, c_1, c_2 are coefficients for local and global search, r_1, r_2 are uniform random numbers in a range [0,1], $\hat{\mathbf{u}}_p(s) = \{\hat{u}_{p1}(s), \ldots, \hat{u}_{pD}(s)\}$ and $\tilde{\mathbf{u}}(s) = \{\tilde{u}_1(s), \ldots, \tilde{u}_D(s)\}$ are the best parameters in all past of p th particles and all particles and defined as follows:

$$\begin{aligned} \hat{\mathbf{u}}_p(s) &= \mathop{\arg\max}_{\mathbf{u}_p(s')|0 \le s' \le s} F(\mathbf{u}_p(s')) \\ \tilde{\mathbf{u}}(s) &= \mathop{\arg\max}_{\mathbf{u}_p(s)|0 \le p \le P} F(\hat{\mathbf{u}}_p(s)) \end{aligned} \tag{12}$$

4. Computational Simulations

In this section, the storage / recollection simulations of GMCNN using DC[15] and PSOC and their results are reported. The neurons of all CNN layers are stimulated by the external input $a_i^{\langle n \rangle}$ in each simulation. And recollections of partial patterns of time series patterns are observed as the output of GMCNN. The external input is defined as a follow:

$$a_i^{\langle n \rangle} = c_e(2\hat{\chi}_i - 1) \tag{13}$$

where, c_e is a scaling coefficient for external input, $\hat{\boldsymbol{\chi}} = \{\hat{\chi}_1, \ldots, \hat{\chi}_N\}$ is an external input pattern given to each CNN layer.

Table 1 shows parameter setting in all simulations. Table 2 shows parameter setting in 3 kind of the simulation environments: Env.1 with 2 CNN layers, Env.2 with 3 CNN layers and Env.3 with 4 CNN layers.

In this paper, the number of kinds of the time series patterns stored in GMCNN is 2 defined as follows:

$$\boldsymbol{\chi}(t_l) = \begin{cases} \mathbf{r}(0,1) & \textbf{if} \quad 0 \le t_l \le 4 \\ \mathbf{r}(0,1) & \textbf{if} \quad 7 \le t_l \le 11 \\ \mathbf{0.5} & \textbf{otherwise} \end{cases} \tag{14}$$

Table 1 Parameters setting.

Name	Symbol	Value
Number of neurons in a CNN	N	100
Initial internal states	$\eta_i^{\langle n \rangle}(0), \zeta_i^{\langle n \rangle}(0), \xi_i^{\langle n \rangle}(0)$	0
Initial connection weight	$w_{ij}^{\langle n,m \rangle}$	0
Initial patterns	$x_i^{\langle n \rangle}(0)$	Random pattern
Steepness parameter	ε	0.015
Dimension of parameters	D	4
Number of particle	P	10
Maximum step	$s^{\langle \max \rangle}$	20
Search space range	$\mathbf{u}^{\langle \max \rangle}$	{1,1,20,1}
	$\mathbf{u}^{\langle \min \rangle}$	{0,0,0,0}
Search velocity range	$\mathbf{v}^{\langle \max \rangle}$	{1,1,20,1}
	$\mathbf{v}^{\langle \min \rangle}$	{-1,-1,-20,-1}
Inertia coefficient	ω	1
Search coefficients	c_1, c_2	1
Initial association layer	$n'(0)$	1

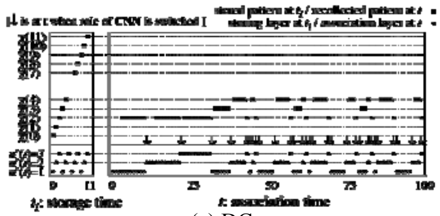
Table 2 Parameter setting of each simulation environment.

Environment number	The number of layer	Rate of effective ness	Learning rate	Scaling coefficient
	M	γ	β	c_e
Env.1	2	1	0.2	6
Env.2	3	0.5	0.3	9
Env.3	4	1/3	0.4	12

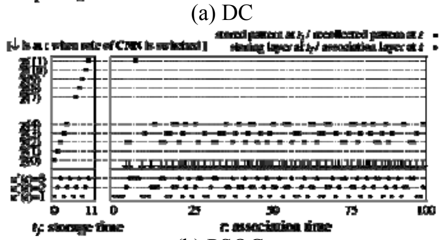
$$\boldsymbol{\chi}(t_l) = \begin{cases} \mathbf{r}(0,1) & \textbf{if} \quad 0 \le t_l \le 4 \\ \mathbf{r}(0,1) & \textbf{if} \quad 7 \le t_l \le 11 \\ \mathbf{r}(0,1) & \textbf{if} \quad 4 \le t_l \le 18 \\ \mathbf{0.5} & \textbf{otherwise} \end{cases} \tag{15}$$

where, $\mathbf{r}(0,1)$ is a binary pattern randomized with 0 or 1, $\boldsymbol{\chi}(t_l) = \mathbf{0.5}$ means no pattern can be observed (because $\delta_{ij}\left(t_l, \tau^{\langle m \rangle}(t_l)\right)$ in Eq. (4)). Eq. (14) gives 2 time series patterns ($0 \le t_l \le 4$, $7 \le t_l \le 11$) which are consisted of 5 random binary patterns. An interval between these time series patterns are 2 from storage time $t_l = 5$ to $t_l = 6$. On the other hand, Eq.(15) gives 3 time series patterns consisted of 5 random binary patterns which intervals between these time series patterns are 2 respectively ($t_l = 5,6$, $t_l = 12,13$).

In computational simulations, recollection processes of the stored plural time series patterns were observed from $t = 1$ to $t = 100$ in Env.1-Env.3 and in GMCNNs using DC[15] and PSOC. The recollection processes were repeated 100 times (as 100 samples) to evaluate a performance of GMCNN by each average stored pattern retrieval times in Env.1-Env.3.



(a) DC



(b) PSOC

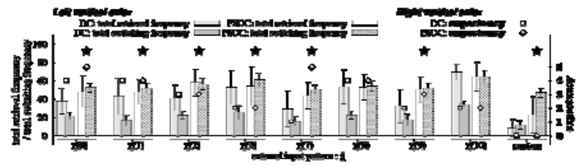
Fig. 2 Examples of GMCNNs which stored the times series patterns of Eq. (14) in Env.2 and given $\chi(2)$ of Eq. (14) as external input.

Fig. 2 shows storage processes and recollection processes of GMCNN with 3-CNN layers (Env.2). The plural time series patterns stored in GMCNN is the patterns of Eq. (14). Fig. 2 (a) is using DC. Fig. 2 (b) is using PSOC. The external input to GMCNN was a random pattern as same as the stored pattern at the storage time $t_l = 2$ given by Eq. (14).

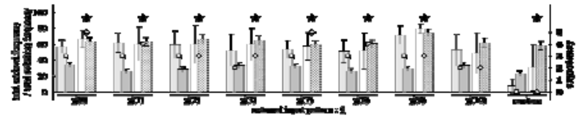
The left column of Fig.2 (a)-(b) shows the storage processes of GMCNN for storing 2 time series pattern $\chi(0),\ldots,\chi(11)$ of Eq.(14) by Eq.(4)-(5). The right column of Fig.2 (a)-(b) shows the recollections of the stored patterns ($\chi(0),\ldots,\chi(4)$ and $\chi(7),\ldots,\chi(11)$) on an above ten lines and the role of "association layer" on an under three lines in GMCNN with 3-CNN layers from the association time $t=1$ to $t=100$. From Fig.2, we can confirm that the switching of roles of CNNs worked constantly and $\chi(2)$, $\chi(3)$ and $\chi(4)$ were recollected in different association times. And PSOC resulted more role switching times of 3-CNN layers than DC, comparing the role switching times of DC and PSOC (↓ in Fig.2).

To investigate the relationship between the external input and the output of GMCNN (retrieval patterns), on columns in Table 3, we show the average retrieval frequencies of 10 stored patterns ($\chi(0),\ldots,\chi(4)$ and $\chi(7),\ldots,\chi(11)$) in GMCNNs with Env.2 given respective 9 kinds of the external input patterns ($\chi(0),\ldots,\chi(3)$, $\chi(7),\ldots,\chi(10)$ and a non-stored random pattern).

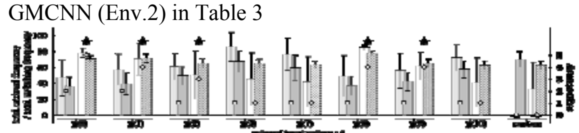
In Table 3, above data in cells mean the average retrieval times, and under data in parentheses are deviations.



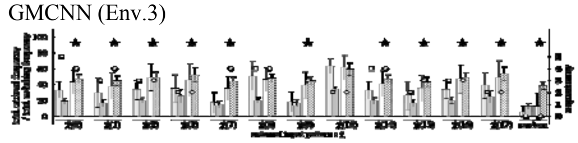
(a) A case of 2 time series patterns in Eq.(13) for a 2-layer GMCNN (Env.1)



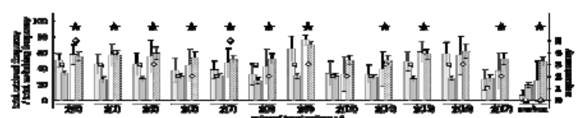
(b) The case of 2 time series patterns in Eq.(13) for a 3-layer GMCNN (Env.2) in Table 3



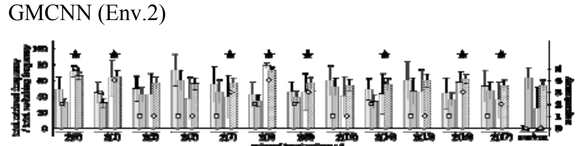
(c) A case of 2 time series patterns in Eq.(13) for a 4-layer GMCNN (Env.3)



(d) A case of 3 time series patterns in Eq.(14) for a 2-layer GMCNN (Env.1)



(e) A case of 3 time series patterns in Eq.(14) for a 3-layer GMCNN (Env.2)



(f) A case of 3 time series patterns in Eq.(14) for a 4-layer GMCNN (Env.3)

Fig. 3 The average total retrieval frequencies, the average total switching frequencies and the expectancies of GMCNNs (for an association time from t=1 to 100 in 100 samples) given by different parameter control methods: DC and PSOC; "★" denotes that over one of total retrieval, total switching and "expectancy" in the GMCNN using PSOC are higher than in one using DC.

Table 3 The average retrieval frequencies of the part of the stored time series patterns, the average of their totals, the average total switching frequencies and the expectancies of the number of the patterns recollected for an association time from t=1 to 100 (and their standard deviations) in GMCNN which store Eq. (14) in Env.2 (in 100 sample).

(a) DC

Stored pattern	External input pattern: $\hat{\chi}$								
	$\chi(0)$	$\chi(1)$	$\chi(2)$	$\chi(3)$	$\chi(7)$	$\chi(8)$	$\chi(9)$	$\chi(10)$	random
$\chi(0)$	**18.89** (2.15)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.05 (0.50)	0.00 (0.00)
$\chi(1)$	**6.22** (4.28)	**20.31** (5.81)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.01 (0.10)
$\chi(2)$	**31.56** (6.19)	**19.66** (8.51)	**23.50** (8.61)	0.26 (2.59)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.61 (1.77)
$\chi(3)$	0.00 (0.00)	**20.63** (5.09)	**16.75** (9.95)	**28.75** (8.71)	0.00 (0.00)	0.09 (0.63)	0.00 (0.00)	0.38 (2.35)	1.29 (2.58)
$\chi(4)$	0.13 (0.93)	0.89 (1.56)	**17.82** (7.01)	**22.62** (13.88)	0.00 (0.00)	0.10 (0.70)	0.00 (0.00)	0.09 (0.63)	1.68 (3.51)
$\chi(7)$	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	**21.92** (5.08)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
$\chi(8)$	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	**5.88** (4.39)	**22.54** (7.39)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
$\chi(9)$	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	**24.73** (6.05)	**12.96** (7.41)	**17.43** (1.80)	0.00 (0.00)	0.50 (2.63)
$\chi(10)$	0.00 (0.00)	0.00 (0.00)	0.09 (0.90)	0.06 (0.42)	0.02 (0.20)	**13.87** (5.49)	**24.27** (11.57)	**28.70** (8.34)	0.39 (1.60)
$\chi(11)$	0.00 (0.00)	0.00 (0.00)	0.85 (3.93)	0.43 (2.54)	0.26 (1.02)	1.26 (2.07)	**29.30** (3.56)	**22.97** (14.46)	2.70 (5.20)
Total retrieval	56.80 (8.46)	61.49 (12.31)	59.01 (17.39)	52.12 (20.34)	52.81 (11.30)	50.82 (14.38)	71.00 (11.81)	52.19 (19.70)	7.18 (8.71)
Total switching	34.07 (3.51)	26.15 (3.29)	28.06 (3.46)	32.98 (4.43)	32.31 (3.22)	26.30 (3.21)	29.30 (3.22)	32.93 (4.40)	23.11 (3.67)
expectancy	**3**	**3**	**3**	**2**	**3**	**3**	**3**	**2**	**0**

(b) PSOC

Stored pattern	External input pattern: $\hat{\chi}$								
	$\chi(0)$	$\chi(1)$	$\chi(2)$	$\chi(3)$	$\chi(7)$	$\chi(8)$	$\chi(9)$	$\chi(10)$	random
$\chi(0)$	**13.82** (6.70)	0.25 (0.90)	0.22 (0.70)	0.00 (0.00)	0.22 (0.61)	0.14 (0.79)	0.00 (0.00)	0.38 (1.14)	0.45 (1.27)
$\chi(1)$	**13.14** (7.60)	**14.56** (6.97)	0.33 (1.00)	0.00 (0.00)	0.15 (0.74)	0.29 (0.98)	0.00 (0.00)	0.33 (1.11)	0.65 (1.67)
$\chi(2)$	**21.72** (2.31)	**17.29** (7.71)	**17.51** (9.79)	**13.19** (10.83)	1.56 (4.82)	1.38 (4.39)	0.12 (1.19)	3.73 (7.31)	5.18 (8.55)
$\chi(3)$	**9.97** (6.53)	**17.40** (7.54)	**15.74** (10.00)	**22.94** (6.20)	1.89 (4.97)	1.90 (4.78)	0.15 (1.49)	4.69 (7.87)	5.57 (8.39)
$\chi(4)$	**7.49** (6.29)	**6.71** (5.60)	**15.68** (9.82)	**20.39** (7.11)	1.70 (4.61)	1.69 (4.45)	0.12 (1.19)	4.00 (7.41)	5.20 (8.28)
$\chi(7)$	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	**4.93** (3.19)	0.04 (0.31)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
$\chi(8)$	0.00 (0.00)	0.00 (0.00)	0.02 (0.20)	0.20 (1.32)	**3.68** (3.18)	**8.61** (6.40)	0.00 (0.00)	0.00 (0.00)	0.11 (0.82)
$\chi(9)$	0.00 (0.00)	1.08 (3.67)	3.27 (6.96)	0.52 (2.55)	**16.59** (7.18)	**14.39** (8.26)	**26.88** (3.13)	8.63 (10.03)	4.29 (7.98)
$\chi(10)$	0.01 (0.10)	1.22 (3.99)	3.24 (6.95)	0.63 (2.84)	**13.55** (6.65)	**14.63** (8.34)	**25.88** (3.24)	**14.14** (10.03)	4.38 (8.12)
$\chi(11)$	0.00 (0.00)	1.24 (4.02)	3.42 (7.12)	1.46 (3.98)	**12.79** (6.45)	**8.18** (7.05)	**26.10** (3.26)	**12.53** (10.18)	4.67 (8.42)
Total retrieval	66.15 (10.98)	59.75 (21.49)	59.43 (24.16)	59.33 (20.26)	57.06 (17.61)	51.25 (23.35)	79.25 (6.27)	48.43 (26.10)	30.50 (29.43)
Total switching	62.87 (5.39)	62.66 (5.48)	65.94 (6.09)	64.43 (6.27)	59.13 (4.93)	60.29 (5.24)	73.83 (4.97)	61.37 (6.19)	58.12 (5.73)
expectancy	**5**	**4**	**3**	**3**	**5**	**4**	**3**	**2**	**0**

The **bold number** in Table 3 means the average retrieval frequency is larger than its deviation (that is the recollection of the stored pattern is expected and it is called "expectancy" in this paper). Height of the "expectancy" indicates higher recollection performance of GMCNN. GMCNN using PSOC showed higher expectancy, comparing GMCNNs using DC in Table 3(a) and PSOC in Table 3(b).

Total retrieval frequencies of all stored patterns, total switching frequencies and expectancies for each external input pattern in Table 3 and other simulations are illustrated in Fig. 3. Fig. 3 (a) is in a case of 2 time series patterns given by Eq. (14) in 2-layer GMCNN (Env.1). Fig. 3 (b) and (c) is in 3-layer GMCNN and in 4-layer GMCNN. Fig. 3 (d)-(f) is in a case of 3 time series patterns given by Eq. (15) in 2, 3 and 4-layer GMCNN.

The better performance of GMCNN using PSOC than DC (marked with ★) is concluded, comparing the total retrieval frequencies, total switching frequencies and expectancies in respective external input patterns. Therefore, PSOC proposed in this paper is suggested preferentially as the control method of the switching the role of CNN layers as "display layer" and "association" in GMCNN.

5. Conclusion

In this paper, we proposed the control method using PSO as meta-heuristics which finds and uses the optimal parameters in GMCNN. In the computational simulations using 3 kinds of parameter setting, 2 kinds of time series patterns, different external input patterns and 2 kinds of the control methods (DC and PSOC), the better performance of GMCNN using PSOC was confirmed by comparing the processing results of DC and PSOC. The highest expectancy is in 3-layer GMCNN. As future works, GMCNN is expected to be adopted into long term memory models and the limbic model of the brain.

References

1. Hopfeld JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. U.S.A. 81:3088-3092
2. Aihara K, Tanaka T, Toyoda M (1990) Chaotic neural network. Phys. Lett. A 144(6-7):333-340
3. Adachi M, Aihara K (1997) Associative dynamics in a chaotic neural network. Neural Networks 10(1):83-98
4. Kaneko K (1990) Clustering coding, switching, hierarchical ordering, and control in a network of chaotic elements. Physica D: Nonlinear Phenomena 41:137-172
5. Hasegawa M, Ikeguchi T, Aihara K (2002) Solving large scale traveling salesman problems by chaotic neurodynamics. Neural Networks 15(2):271-283

6. Horio Y, Ando H, Aihara K (2009) Fundamental technology for complex computational systems. IEICE Fundamentals Review 3(2):34-44
7. Kuremoto T, Eto T, Kobayashi K, Obayashi M (2005) A multilayer chaotic neural network for associative memory. Proc. of SICE Annual Conf.
8. Kuremoto T, Eto T, Kobayashi K, Obayashi M. (2005) A chaotic model of hippocampus-neocortex. Lect. Notes Comput. Sci. 3610:439-448
9. Kuremoto T, Eto T, Kobayashi K, Obayashi M. (2007) A hippocampus-neocotex model for chaotic association. in Trends in Neural Computation (Eds: K. Chen and L. Wang), Studies in Computational Intelligence 35:111-133
10. Kuremoto T, Ohata T, Kobayashi K, Obayashi M. (2009) A dynamic associative memory system by adopting an amygdala model. AROB 13(2):478-482
11. Kuremoto T, Ohata T, Kobayashi K, Obayashi M. (2009) A functional model of limbic system of brain. Lect. Notes Comput. Sci. 5819:135-146
12. He G, Cao Z, Chen H, Zhu P (2003) Controlling chaos in a neural network based on the phase space constraint. Int. J. Mod Phys B 7(22-24):4209-4214
13. He G, Shrimali MD, Aihara K (2008) Threshold control of chaotic neural network. Neural Networks 21:114-121
14. Kuremoto K, Watanabe S, Kobayashi K, Obayashi M. (2011) Dynamical recollection of interconnected neural networks using metaheuristics (in Japanese). IEEJ Trans. EIS 131(8):1475-1484
15. Watanabe S, Kuremoto T, Kobayashi K, S. Mabu, M. Obayashi (2013) The recollection characteristics of a generalized MCNN. Proc. of SICE Annual Conf. 1375-1380
16. Hebb DO (1949) The organization of behavior. New York: Wiley
17. Kennedy J, Eberhart RC, Shi Y (1995) Particle swarm optimization. Proc. of the IEEE Int. Conf. on Neural Networks :1942-1948
18. Kennedy J, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA