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 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. On the 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. 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. CNNs are applied to not only dynamic association memory but also optimization and complex computing. Additionally, Kuremoto et al. have combined multiple CNNs to be a multi-layer chaotic neural network (MCNN). 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, or upper and lower limit of recalling times, and meta-heuristics (e.g., genetic algorithm, particle swarm optimization (PSO)). MCNN has been applied to a mathematical model of hippocampus to realize a transform of long-term memory. 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)\textsuperscript{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\textsuperscript{15} and PSOC were compared by the simulation results.

![Diagram](image1)

Fig. 1 Association memory models: (a) a chaotic neural network (CNN)\textsuperscript{2,3}, (b) a generalized multi-layer chaotic neural network (GMCNN)\textsuperscript{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)\textsuperscript{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\textsuperscript{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\textsuperscript{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:

\[
\eta_j^{(i)}(t+1) = k_1^{(i)} \eta_j^{(i)}(t) + \sum_{n \neq j} w_{j,n}^{(i)} x_n^{(i)}(t)
\]

\[
\xi_j^{(i)}(t+1) = k_2^{(i)} \xi_j^{(i)}(t) - k_3^{(i)} x_j^{(i)}(t) + a_j^{(i)}
\]

\[
\xi_j^{(i)}(t+1) = k_4^{(i)} \xi_j^{(i)}(t) + \sum_{n \neq j} w_{j,n}^{(i)} x_j^{(i)}(t)
\]

\[
y_j^{(i)}(t+1) = \eta_j^{(i)}(t+1) + \epsilon_j^{(i)}(t+1) + \xi_j^{(i)}(t+1)
\]

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, \( \eta_j^{(i)}(t), \xi_j^{(i)}(t), \gamma_j^{(i)}(t) \) are internal states for feedback inputs from self-layer, refractoriness and inputs from another layers respectively, \( w_{j,n}^{<,m>} \) is a connection weight from \( j \) th neuron of \( m \) th layer to \( i \) th neuron of \( n \) th layer, \( k_1^{(i)}, k_2^{(i)}, k_3^{(i)}, k_4^{(i)} \) are decay parameters for feedback inputs from self-layer, refractoriness and inputs from another layers respectively, \( k_5^{(i)} \) is a refractory scaling parameter, \( a_j^{(i)} \) is a summation of threshold and external input, \( \gamma \) is a rate of effectiveness from another layers, \( y_j^{(i)}(t) \) is an internal state, \( x_j^{(i)}(t) = \{x_n^{(i)}(t), \ldots, x_j^{(i)}(t), \ldots\} \) is an output of CNN, \( R^{(i)}(t) \) is a propositional function ( \( n \) th CNN layer performs as "association layer" if \( R^{(i)}(t) \) is fulfilled, otherwise the layer performs as "display layer"). \( f(\cdot) \) is a output function of the neuron:

\[
f(y) = \frac{1}{1 + e^{-y/\epsilon}}
\]

where, \( \epsilon \) is a steepness parameter. \( k^{(i)} = \{k_1^{(i)}, \ldots, k_5^{(i)}\} \) 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\textsuperscript{16} for storing the plural time series patterns into the multiple CNN layers alternately.

Published by Atlantis Press
Copyright: the authors
where, \( t_j \) is a storage time, \( g(t) = (x(t), \ldots, x_n(t)) \) is a partial pattern of the time series patterns at \( t_j \), \( g(t_k) \) is the layer number to store a pattern, \( \varepsilon^{(mn)}(t_k) \) is a storage time for \( m \) th layer at \( t_k \), \( \chi \) 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:

\[
g(t_k) = t_j \mod M + 1
\]

The direct control method (DC)\(^{15}\) controls “association layer” in GMCNN by switching the internal parameters \( k^{(x(t))} \) 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. Meta-heuristic Control Method

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 \( k^{(x(t))} \). The propositional function \( R^{(s)}(t) \) 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 as follows:

\[
n'(t + 1) = \begin{cases} n'(t) \mod M + 1 & \text{if } Q(t) \\ n'(t) & \text{otherwise} \end{cases}
\]

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:

\[
z_j(t) = \begin{cases} 1 & \text{if } x_j^{(x(t))}(t) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}
\]
Step 5 This process is repeated from Step 3 if an iteration step \( s \) fulfills \( s < S^{(max)} \). Otherwise, this process ends and “association layer” is renewed by Eq. (1)-(2) \( (t \leftarrow t + 1) \) after \( \mathbf{u}(s) \) is applied to \( \mathbf{v}^{(}\epsilon(t)) \) \( (k^{(\epsilon(t))} \leftarrow k_{\mathbf{u}}(0)) \). \( Q(t) \) of Eq. (7) is defined as \( E(t) \leq E(t + 1) \).

where, \( U(a,b) \) is an uniform random number in a range \( [a, b] \), \( v^{(\min)}_d \) and \( v^{(\max)}_d \) are the minimum value and the maximum value in a \( d \)-dimensional range \( v^{(\min)}_d, v^{(\max)}_d \), \( v_{pd}(s) \) is a velocity of \( d \in \{1, \ldots, D\} \) parameter of \( p \in \{1, \ldots, P\} \) th particle at step \( s \) , \( E(t+1) \) is the NE of Eq.(6) in “association layer” renewed by Eq.(1)-(2) in a case that parameters \( \mathbf{u}_{pd}(s) = \{v_{pd}(s), \ldots, v_{pd}(s)\} \) of \( p \) th particle at step \( s \) are applied to \( \mathbf{k}^{(\epsilon(t))} \) of the layer at time \( t \), \( F(\mathbf{u}_{pd}(s)) \) is the fitness of \( p \) th particle, \( \omega \) is an inertia coefficient, \( c_1, c_2 \) are coefficients for local and global search, \( f_1, f_2 \) are uniform random numbers in a range \( [0, 1] \), \( \mathbf{u}_p(s) = \{u_1(s), \ldots, u_p(s)\} \) and \( \mathbf{u}(s) = \{u_1(s), \ldots, u_p(s)\} \) are the best parameters in all past of \( p \) th particles and all particles and defined as follows:

\[
\mathbf{u}_p(s) = \arg \max_{\mathbf{u}_p(s) \in \mathbb{R}^D} F(\mathbf{u}_p(s))
\]

\[
\mathbf{u}(s) = \arg \max_{\mathbf{u}(s) \in \mathbb{R}^P} F(\mathbf{u}(s)) \tag{12}
\]

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 \( \mathbf{d}(t) \) 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:

\[
\mathbf{d}(t) = c_e(\mathbf{z}, -1) \tag{13}
\]

where, \( c_e \) is a scaling coefficient for external input, \( \mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\} \) 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:

\[
\chi(t) = \begin{cases} 
\mathbf{r}(0,1) & \text{if } 0 \leq t_1 \leq 4 \\
\mathbf{r}(0,1) & \text{if } 7 \leq t_1 \leq 11 \\
0.5 & \text{otherwise}
\end{cases} \tag{14}
\]

where, \( \mathbf{r}(0,1) \) is a binary pattern randomized with 0 or 1, \( \chi(0) = 0.5 \) means no pattern can be observed (because \( \chi(t) \) is in Eq. (4)). Eq. (14) gives 2 time series patterns \( (0 \leq t_1 \leq 4, 7 \leq t_1 \leq 11) \) which are consisted of 5 random binary patterns. An interval between these time series patterns are 2 from storage time \( t_1 = 5 \) to \( t_1 = 6 \). 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 \( (t_1 = 5, 6, t_1 = 12, 13) \).

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

<table>
<thead>
<tr>
<th>Table 1 Parameters setting.</th>
<th>Name</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of neurons in a CNN</td>
<td>( N )</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Initial internal states</td>
<td>( \eta(t)(0), \xi(t)(0), \eta(t)(0) )</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Initial connection weight</td>
<td>( w^{(\infty)} )</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Initial patterns</td>
<td>( \eta(0) )</td>
<td>Random pattern</td>
<td></td>
</tr>
<tr>
<td>Steepness parameter</td>
<td>( \varepsilon )</td>
<td>0.015</td>
<td></td>
</tr>
<tr>
<td>Dimension of parameters</td>
<td>( D )</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Number of particle</td>
<td>( P )</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Maximum step</td>
<td>( \chi^{(\max)} )</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Search space range</td>
<td>( \mathbf{u}^{(\min)} )</td>
<td>{1,1,20,1}</td>
<td></td>
</tr>
<tr>
<td>Search velocity range</td>
<td>( \mathbf{u}^{(\max)} )</td>
<td>{0,0,0,0}</td>
<td></td>
</tr>
<tr>
<td>Inertia coefficient</td>
<td>( \omega )</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Search coefficients</td>
<td>( c_1, c_2 )</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Initial association layer</td>
<td>( n(0) )</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2 Parameter setting of each simulation environment.</th>
<th>Environment number</th>
<th>The number of layer</th>
<th>Rate of effectiveness</th>
<th>Learning rate</th>
<th>Scaling coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Env.1</td>
<td>( M )</td>
<td>2</td>
<td>1</td>
<td>0.2</td>
<td>6</td>
</tr>
<tr>
<td>Env.2</td>
<td></td>
<td>3</td>
<td>0.5</td>
<td>0.3</td>
<td>9</td>
</tr>
<tr>
<td>Env.3</td>
<td></td>
<td>4</td>
<td>1/3</td>
<td>0.4</td>
<td>12</td>
</tr>
</tbody>
</table>

\[
\chi(t) = \begin{cases} 
\mathbf{r}(0,1) & \text{if } 0 \leq t_1 \leq 4 \\
\mathbf{r}(0,1) & \text{if } 7 \leq t_1 \leq 11 \\
0.5 & \text{otherwise}
\end{cases} \tag{15}
\]
The Recollection Characteristics of GMCNN

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 \( t = 2 \) given by Eq. (14).

The left column of Fig.2 (a)-(b) shows the storage processes of GMCNN for storing 2 time series pattern \( \chi(0)\ldots\chi(1) \) 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 \( \chi(0)\ldots\chi(5) \), \( \chi(7)\ldots\chi(10) \) 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.

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

<table>
<thead>
<tr>
<th>Stored pattern</th>
<th>g(0)</th>
<th>g(1)</th>
<th>g(2)</th>
<th>g(3)</th>
<th>g(4)</th>
<th>g(5)</th>
<th>g(6)</th>
<th>g(7)</th>
<th>g(8)</th>
<th>g(9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>g(0)</td>
<td>18.89</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.05</td>
<td>0.00</td>
</tr>
<tr>
<td>g(3)</td>
<td>0.00</td>
<td>20.65</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
</tr>
<tr>
<td>g(4)</td>
<td>0.13</td>
<td>0.89</td>
<td>7.92</td>
<td>22.62</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.68</td>
<td>0.00</td>
</tr>
<tr>
<td>g(5)</td>
<td>(0.95)</td>
<td>(5.50)</td>
<td>(8.71)</td>
<td>(8.71)</td>
<td>(8.71)</td>
<td>(8.71)</td>
<td>(8.71)</td>
<td>(8.71)</td>
<td>(8.71)</td>
<td>(8.71)</td>
</tr>
<tr>
<td>g(6)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>21.92</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>g(7)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>5.88</td>
<td>22.54</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>g(8)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>24.73</td>
<td>12.96</td>
<td>17.43</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>g(9)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>g(10)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>20.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>g(11)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total retrieval</td>
<td>56.80</td>
<td>61.49</td>
<td>59.01</td>
<td>52.12</td>
<td>52.81</td>
<td>50.82</td>
<td>51.00</td>
<td>52.19</td>
<td>7.18</td>
<td></td>
</tr>
<tr>
<td>Total switching</td>
<td>34.07</td>
<td>35.31</td>
<td>33.71</td>
<td>36.60</td>
<td>33.60</td>
<td>32.30</td>
<td>29.90</td>
<td>32.03</td>
<td>23.11</td>
<td></td>
</tr>
</tbody>
</table>

(a) DC

(b) PSOC

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


S.Watanabe, T. Kuremoto, S. Mabu, M. Obayashi and K. Kobayashi

Published by Atlantis Press

Copyright: the authors 78