Evolving Chaotic Neural Network for Creative Sequence Generation

Shunji Kawabata, Keiichi Sakai and Tadashi Ae

Graduate School of Engineering, Hiroshima University
1-4-1 Kagamiyama, Higashi-Hiroshima, 739-8527 Japan
fax: +81 (0)82 422 7195, e-mail: {kwbt,sakai,ae}@aial.hiroshima-u.ac.jp,
http://www.aial.hiroshima-u.ac.jp/~kwbt

Abstract
This paper describes an approach to generate a sequence requiring an unrealizable function by programs, such as a flash that is required especially in creative activity of a human. We have already proposed a recurrent neural network that demonstrates a generation of several creative sequences, but convergency and stability problems occur. On the other hand, it is known in biological experiments where the chaotic sequences can be observed from brain waves. The neural network constructed from chaotic neurons has nonlinear dynamics, but there remains the difficulty of training method. We propose an evolutional methodology to train a chaotic neural network, and introduce Darwinism for its evolving process. To determine their most suitable structure and the weights of connection, we use AIC for the fitness value.
Keywords : Evolution, Creation, Neural Network, Chaos, Sequence Generation

1 Introduction

Nowadays, though a lot of functions are simulated by programs, some of them (e.g., a flash) is not yet realized by an artificial function.

This paper describes an approach to generate a creative sequence requiring an unrealizable function such as a flash. When observing what a creative sequence means, it contains several newer sequences which we have never seen, and those sequences transit with a time shift. Thus we represent a creative sequence as a vector sequence with a time shift, and construct a model which acquires the knowledge from a creator. The model is applicable for generation of creative vector sequences corresponding to music scores.

We have already proposed a recurrent neural network, i.e., a context-sensitive neural network (in short, CSNN) that demonstrates a generation of several creative sequences. The CSNN has a finite number of context-sensitive neurons which hold their states in the past time as the recurrent neurons of hidden layers. Though the CSNN can generate several sequences, two problems occur. One problem is the convergency, that is, the CSNN terminates often at the same vector values

International Journal of Computing Anticipatory Systems, Volume 14, 2004
Edited by D. M. Dubois, CHAOS, Liège, Belgium, ISSN 1373-5411 ISBN 2-930396-00-8

permanently. The other problem is the stability, that is, it generates the same pattern of subsequences continuously.

The creation is one of the anticipation. A lot of anticipatory systems can be represented by incursive or hyperincursive equations[1]. For example, the recursive system can be written as $\boldsymbol{x}(t+1) = f(\ldots, \boldsymbol{x}(t-2), \boldsymbol{x}(t-1), \boldsymbol{x}(t); \boldsymbol{p})$ where $\boldsymbol{x}(t)$ is the vector states at time t, f is the recursive vector function and $\boldsymbol{p}$ is a set of parameters to be adjusted. When we observe $\boldsymbol{p}$ as an estimation parameter of system at time t, $\boldsymbol{x}(t+1)$ is sickly likeness of a new vector that is created from function f and past and/or present events. In some situation, there is a possibility that an incursive equation in an anticipatory system is to be a chaos system[2].

It is known in biological experiments where the chaotic sequences can be observed from brain waves. The brain consists of a large number of neurons, whose network exhibits chaotic behavior. In this sense, we introduce a brain model based on neurons including chaotic behavior. Aihara et al. have proposed a chaotic neural network constructed from neurons which has nonlinear dynamics. However, there remains the difficulty of training method, since the error back-propagation algorithm is not effective for the recurrent neural network[3].

This paper provides a method of sequence generation using the evolving model constructed from chaotic neurons. We introduce Darwinism for evolving process that imitates the processes, where living things adapt their structures to the various environments by evolution and learning. Using AIC(Akaike Information Criterion) for the fitness value of each individuals, we can determine their most suitable structure and the weights of connection. As a result, we could show a new methodology for creative sequence generation.

2 Vector Sequence

Most of creations can be represented by the sequence. For example, pictures are constructed from color and their position on the canvas, and a painter puts the color on it by brushes with a time shift. Musics are more obvious instance of it. A composer puts music notes which mean pitches of tones side by side on the score sheet.

Now we consider those creative sequence as the vector sequence, thus we formulize it as follows.

When we define the length m vector sequence as $\mathcal{V}_m$,

$$\mathcal{V}_m \stackrel{\text{def}}{=} \boldsymbol{V}_0, \boldsymbol{V}_1, \cdots, \boldsymbol{V}_m \quad where \quad \boldsymbol{V} \stackrel{\text{def}}{=} [v_1, v_2, \cdots, v_n] \quad (v_i \in \boldsymbol{R}, i = 1, 2, \cdots, n). \tag{1}$$

Note that this is not only sequence but there is the context between these sequences. Even in painting or composing, we may not easily make the sequences, but we consider the context already existing. The aim of acquiring the knowledge from creations is acquiring the context.

Fundamentally, we discuss the model which takes a vector sequence $\mathcal{V}_m$ as an input of the system, and compute $\boldsymbol{V}_{m+1}$ as an output from the system. This means that the system has predicted a future of a next state based on past and/or present state of it, and also we can say that the system has produced a new creation. In other words, when the same thing is to be applied to $\mathcal{V}_m := \boldsymbol{V}_1, \ldots, \boldsymbol{V}_m, \boldsymbol{V}_m$ as an input, the system is to generate another vector $\boldsymbol{V}_{m+1}$. We introduce a time parameter to vector sequence, and we have

$$\mathcal{V}_m(t+1) = f(\mathcal{V}_m(t), \boldsymbol{p}) \quad , \tag{2}$$

where $\mathcal{V}_m(t)$ is the length m vector sequence in time t, $\boldsymbol{p}$ is a set of parameters of the system and f is a mapping function of the system.

3 Chaotic Neurons

The concept of neural network was first proposed by McCulloch and Pitts. By using the spin-glass model, Hopfield has first shown that the energy function decreases monotonically and one energy minimum corresponds to one memory. The model has been applied to the associative memory problem and the combinatorial optimization problem, for example TSP. However, the model has the difficulty that the system cannot escape from the spurious minimum, if the system falls into it. Recently many methods are proposed to solve this difficult problem by using the chaos.

3.1 Chaotic Neural Network

Aihara et al. have proposed a model of a single neuron with chaotic dynamics, in which some properties of biological neuron are taken into account[3]. The chaotic neuron is formalized as follows;

$$\begin{cases} y(t+1) = ky(t) - \alpha f\left(y(t)\right) + a \\ x(t+1) = f(y(t+1)) \end{cases} . \tag{3}$$

Here k, α and a are parameters which represent the characteristic property of chaotic neurons, $y(t)$ is the internal state of neuron in time t, and f is the sigmoid function, that is written as follows;

$$f(x) = \frac{1}{1 + \exp\left(\frac{-x+\theta}{T}\right)} . \tag{4}$$

If we construct by connecting mutually M chaotic neurons, the response is

$$y_i(t+1) = k_i y_i(t) + \sum_{j=1}^{M} w_{ij} f\left(y_j(t)\right) - \alpha f\left(y_i(t)\right) + a_i \quad , \tag{5}$$

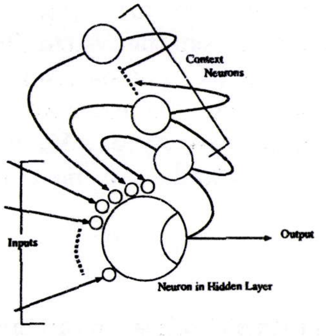

Fig. 1: Context-Sensitive Neuron

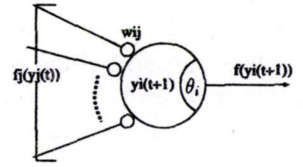

Fig. 2: Chaotic Neuron

where the index i means the serial number of neurons, and the others, $k_i, \alpha_i, a_i, w_{ij}$ are parameters. We can observe this equation as a network which has short-time memory, because the internal state in time $t+1$ depends on the state in time t.

3.2 Chaotic Dynamics

A nonlinear system can have a complicated steady-state behavior that is not equilibrium, periodic oscillation, or almost-periodic oscillation. Such a behavior is usually referred as chaos. Some of these chaotic motions exhibit randomness, despite the deterministic nature of the system.

It has been believed for a long time that a living thing keeps regular state of its inside of the body by the function called homeostasis, but from the electric physiological experiment, the data including fluctuation accompanied by time change which should be called rather dynamic homeostasis is observed. Furthermore, it turns out that the chaos-tendency seen in them appears to activity measurement data, such as brain waves and nerves, notably([4], [13]).

Generally, these chaotic dynamics can be made from nonlinear map represented by difference equation, as the following:

$$\begin{cases} \boldsymbol{V}_{t+1} = f_s(\mu_t, \boldsymbol{V}_t) \\ \mu_t = f_u(f_e(\boldsymbol{V}_t)) \end{cases} . \tag{6}$$

In the equation above, $\boldsymbol{V}_t$ is a state vector at time t, and μ_t is parameter vector which constructs the system. f_e is a estimation function of $\boldsymbol{V}_t$, and then f_s and f_u are a nonlinear mapping function and a function which feeds an estimated value back to system parameter μ, respectively.

3.3 Chaos in CSNN

In our previous research, it could be seen that CSNN has the property of convergency and stability, and those properties are not suitable for sequence generation.

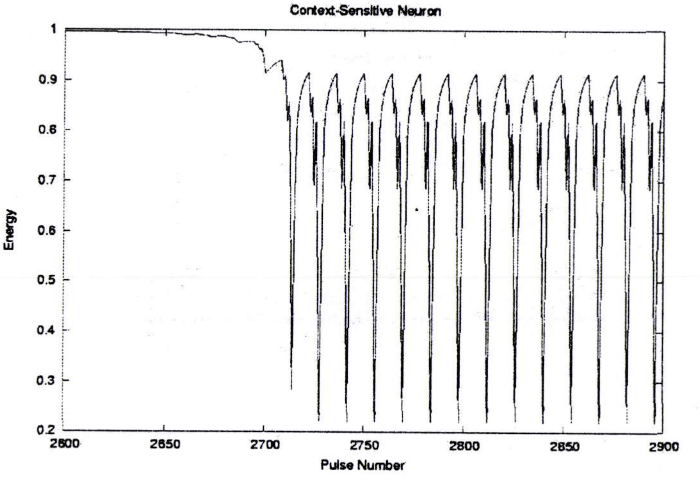

Fig. 3: A Simulation Result of Context-Sensitive Neural Network(CSNN)

However, we can explain those properties from a chaotic viewpoint. One neuron in hidden layer of CSNN is as seen in Fig.1. It's response is as the following;

$$y_i = \sum_j w_{ij} f_j(y_j) + \sum_k w_i^{C=k} f_i(y_i^{C=k}) \quad . \tag{7}$$

The label $C = k$ means the number of context neuron which holds the state of corresponding hidden neuron in the past k time from now. To be easily discussion, we consider the case of $k = 1$, and use parameter t instead of k to represent time. Then the equation above is to be as the following;

$$y_i(t+1) = \sum_j w_{ij} f_j(y_j(t)) + w_i^{C=1} f_i(y_i(t)) \quad . \tag{8}$$

In the same way as above, when $k = 2$,

$$y_i(t+1) = w_i^{C=2} f_i(y_i(t-1)) + \sum_j w_{ij} f_j(y_j(t)) + w_i^{C=1} f_i(y_i(t)) \quad . \tag{9}$$

A simulation result of eq.9 is shown as in Fig.3.

4 Evolving Process

When we determine the suitable stricture of chaotic neural network, we must find parameters by heuristic method or experience because of the response complexity.

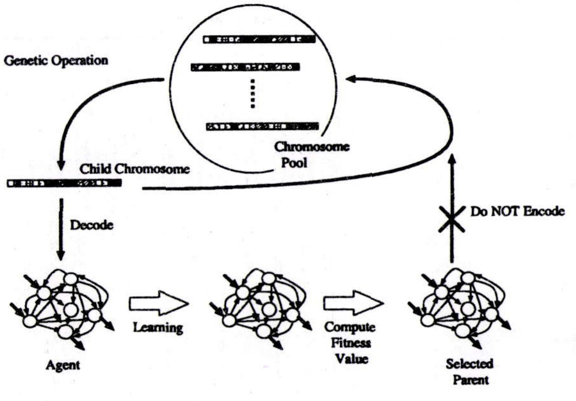

Fig. 4: Evolving Process introducing Darwinism

In this paper, we use an evolutionary method which adopts Darwinian type genetic mechanism, that is superior for adaption to dynamic environments, and we find the desired structure.

Living things acquire the knowledge by learning in the survival that is called acquired character, where the Lamarck type theory of evolution says that the acquired character is to be hereditary. Conversely, it is the Darwin type theory that the acquired character is not to be hereditary. The mainstream of the present theory of evolution serves as the Darwin type theory of evolution from molecule genetics.

The evolution algorithm is as follows.

1. Encode the chaotic neural network to genotype
2. Generate the initial group by putting the random numbers on parameters
3. Estimate the structure based on AIC, and compute the fitness value
4. Do some genetic operation using GA
5. Decide termination, else go to #3.

4.1 Genetic Algorithm(GA)

The combination of evolutionary computing and neural networks has attracted many researchers, mainly due to its promise of engineering implications and the biological plausibility of the model.

The use of GA for training the weights of neural network is covered by one of the earliest research. Kitano used GA to acquire graph rewriting rules, instead of directly acquiring the network topology. This method was further augmented

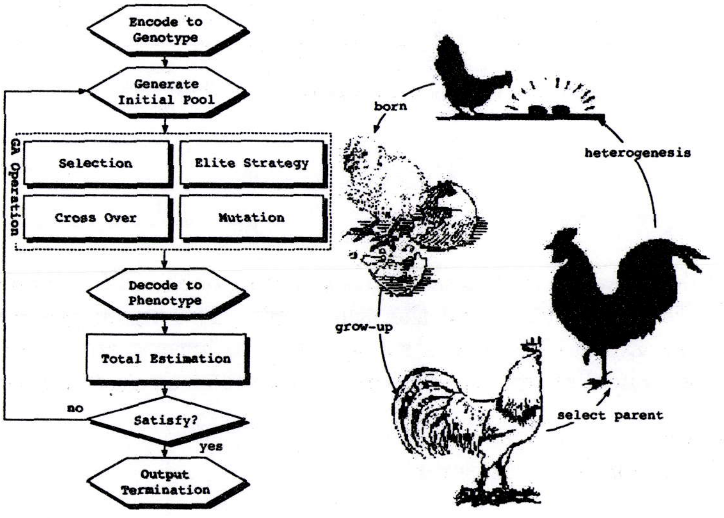

Fig. 5: GA Operations and Evolving Process

as Neurogenetic Learning, to incorporate the evolutionary acquisition of the initial weights of the neural network[5]. Kitano has proposed the extended method that is based on DNA computing[6]. In these research, learning process and genetic process are treated as which are divided into different processes.

Oeda et al. have proposed a structural adaptive learning algorithm based on the theory of evolution[7]. This imitates the processes that living things adapt their structures according to the various of environments by evolution and learning. In this method, once teaching data change during learning under dynamic environments, the learning does not start to train from the initial state again, and therefore, it is considered to be useful for adaptive learning which can take into account inheritance of the network structure, the connection vector, and the learning parameters.

This paper adopts the latter method to determine the structure of chaotic neural network. The outline of GA operation is as follows.

- Genotype

  Having values of weights, structure of network and characteristic parameters of chaotic neurons, define the contained information in one individual as one genotype.

- Selection

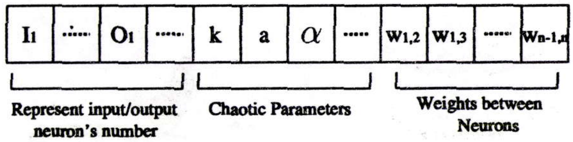

Fig. 6: Genotype

Divide the whole individuals into two group; one is 10% of whole which have the higher fitness value and the others 90%. The higher group is survive to next generation as it is(elite strategy). For the lower group, select individuals which is to be done crossover operation by roulette selection method.

- Crossover
  Using the uniform crossover.

- Mutation
  Do the local mutation that mutates in nearly the index point, and the maximal mutation that mutates the same possibility aim to extend the search space. If there is a fatality gene, it will eliminate from the pool, and the new individual generated with the random number is put in.

4.2 Akaike's Information Criterion(AIC)

AIC has two advantages in roughly discussion as follows; (1) The model which has fewer parameters gets higher fitness value, (2) Statistical computation can be executed even when comparing models have different structures each other. This is because we use AIC as a fitness value of a single genotype. AIC is defineded by the maximum likelihood estimate $\hat{\boldsymbol{\theta}}(\boldsymbol{y})$ under the model M. When make M typify models by $f(\boldsymbol{y}, \hat{\boldsymbol{\theta}}(\boldsymbol{y}))$, we estimate the cross-entropy as few deviation as possible.

If we define p is the number of parameters, AIC is given by

$$AIC = -2\log f(\boldsymbol{y}, \hat{\boldsymbol{\theta}}(\boldsymbol{y})) + 2p \quad . \tag{10}$$

When we construct a Feed-Forward type neural network, we define N_{in} as the number of neurons in the input layer, and N_{out} is for those in the output layer, then the number of weights is given by

$$p = \left(N_{in}N_1 + \sum_{x=1}^{H-1} N_x N_{x+1} + HN_{out} \right) + \left(\sum_{x=1}^{H} N_x + N_{out} \right) \quad ,$$

where N_x is the number of neurons in xth hidden layer and H is the number of hidden layers.

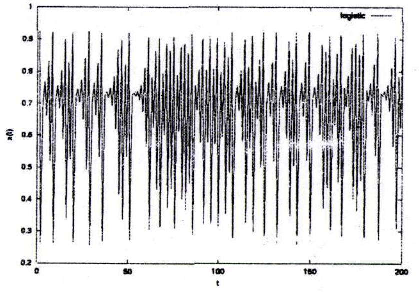

Fig. 7: Logistic Map

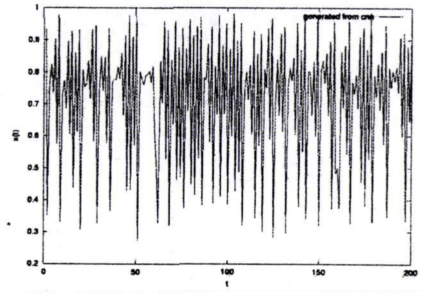

Fig. 8: Sequence generated by ECNN

On the other hand, if we connect neurons mutually and define N as the total number of neurons, since the number of weights is $N(N-1)$, and the number of threshold value is given by $N - N_{in}$, we have

$$p = N(N-1) + N - N_{in} = N^2 - N_{in} \quad .$$

Here we assume that the error function $\mathbf{e}_j = \mathbf{t}_j - \mathbf{z}_j$ between the teacher signals $\mathbf{t}_j$ and the output signals $\mathbf{z}_j$ is to follow the independent normal distribution, the log likelihood of error function is

$$\log f(\boldsymbol{y}, \hat{\boldsymbol{\theta}}(\boldsymbol{y})) = -\frac{N_{out}P}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^{P}\mathbf{e}_j^t\mathbf{e}_j \quad .$$

where P is the total number of teacher signals.

From this equation, maximizing $\log f(\boldsymbol{y}, \hat{\boldsymbol{\theta}}(\boldsymbol{y}))$ means minimizing the error function, thus we can see weights value has been estimated maximum likelihood. Individual i is the network which has smaller AIC_i value is a more adopted network, and therefore the fitness function of individual i can be defined as normalized using maximum AIC_{max} and minimum AIC_{min} of whole gene pool, we have

$$Fitness(i) = \frac{AIC_{max} - AIC_i}{AIC_{max} - AIC_{min}} \quad .$$

The benchmark simulation data which ECNN trained by sequence generated from 1-dimensional logistic map is shown as Fig.7-9.

5 Conclusion

This paper presents an approach to generate a creative sequence by a network which is constructed from chaotic neurons having nonlinear dynamics as its response. Since the chaotic neural network has the difficulty in its training method,

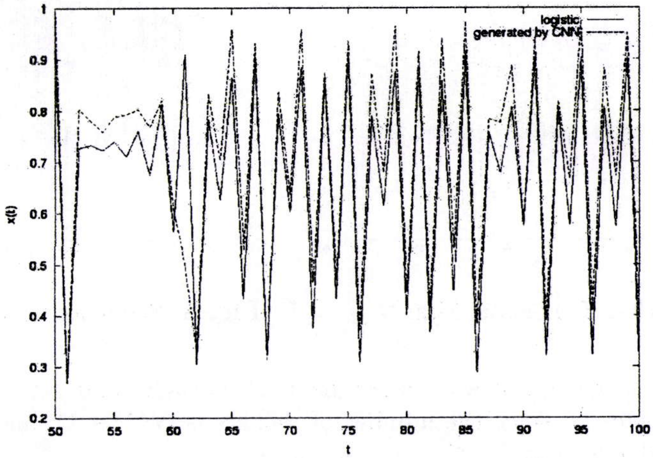

Fig. 9: Comparison Logistic Map with Sequence from ECNN

we have introduced evolving method as another training methodology which imitates the processes where living things adapt their structures to the various environments by evolution and learning. As for the estimation function in process of genetic algorithm, we adopt the AIC value of the phenotype corresponding to each individual.

We have to assemble some data about convergence of evolving chaotic neural network, or comparison of time until it converges with the other model which uses simple error back-propagation rule as its training method. Judging from the data obtained at the present stage, it will be easier that we discover some attractors which causes the nonlinear behavior containing chaos or a limit cycle.

References

[1] D. Dubois. (2000) Review of Incursive, Hyperincursive and Anticipatory Systems - Foundation of Anticipation in Electromagnetism. Computing Anticipatory Systems: CASYS'99 - Third International Conference. Edited by Daniel M. Dubois, Published by The American Institute of Physics, AIP Conference Proceedings 517, pp. 3-30.

[2] D. Dubois (2001) Theory of Incursive Synchronization and Application to the Anticipation of a Chaotic Epidemic. International Journal of Computing Anticipatory Systems. Vol. 10, pp.3-18.

[3] K. Aihara, T. Takabe, and M. Toyoda (1990) Chaotic Neural Networks. Physics Letters. Vol. 144, No. 6,7, pp. 333-340.

[4] K. Aihara (1993) Chaos in Neural Systems. Tokyo Denki University Publishing. In Japanese.

[5] H. Kitano (1990) Designing Neural Networks Using Genetic Algorithms with Graph Generation System. Complex Systems. Vol. 4, No. 4.

[6] H. Kitano (1995) A Simple Model of Neurogenesis and Cell Differentiation based on Evolutionary Large-Scale Chaos. Artificial Life. Vol. 2, No. 1.

[7] S. Oeda, T. Ichimura, M. Terauchi, T. Takahama and Y. Isomichi (2002) A Synthesis of Structural Adaptive Learning Algorithm in Neural Network Based on the Theory of Evolution. IPSJ Journal. Vol. 43, No. 8 pp. 2728-2738. In Japanese.

[8] Jeffery L. Elman (1990) Finding Structure in Time. Cognitive Science. Vol.14, pp.179-211.

[9] John R. Koza (1994) Architecture-Altering Operations for Evolving the Architecture of a Multi-Part Program in Genetic Programming. Stanford University. STAN-CS-TR-94-1528.

[10] S. Kawabata, E. Kimura and T. Ae (2002) Vector Sequence Generation using Recurrent Neural Networks. Proc. of Forum on Information Technology 2002 at Tokyo Institute of Technology in Japan. Biological Information Science. H-13. In Japanese.

[11] S. Kawabata and T. Ae (2002) An Experiment of Creation using Context Sensitive Neural Networks. Proc. of 43rd Meeting Related on Electronics and Information Technology in Chugoku Region at Shimane University in Japan. pp. 423-424. In Japanese.

[12] F. Lerdahl and R. Jackendoff (1983) A Generative Theory of Tonal Music. The MIT Press.

[13] I. Tsuda (1990) Brain from Chaotic View Point. Science Inc. In Japanese.