2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Learning Ability of Evolving Affordable Neural Networks with Genetic Algorithm

Yoko Uwate and Yoshifumi Nishio

Department of Electrical and Electronic Engineering
Tokushima University, JAPAN
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
Email: {uwate, nishio}@ee.tokushima-u.ac.jp

Abstract—In our previous research, we have proposed new network structure with affordable neurons in hidden layer of the feedforward neural network to reflect real brain mechanism. We named this proposed network "Affordable Neural Network." We consider that operation of affordable neurons are especially effective on evolution of neural network in brain. In this study, we investigate the performance of affordable neural network when the hidden layer is evolved using Genetic Algorithm. By computer simulations, we consider that the affordable neurons exert an important influence on evolution process in the hidden layer of the network.

1. Introduction

"What is life?","How does evolution occur?" To find a solution of these questions is one of very important goal in natural science. In Artificial Life, studies of to explain life using various kinds of methods are improving steadily. If behavior of life is expressed by artificial systems, mysteries of life will be cleared. Genetic Algorithm (abbr. GA) [1] is one example of artificial life. GA were invented to mimic some of the process observed in natural evolution and this algorithm use to solve good solution of optimization problems. Furthermore, research of fusion of neural network and GA was reported for much further engineering application. Effectiveness of neural network has been confirmed in pattern recognition, system control, signal processing, and so on [2]-[4], since the back propagation (BP) learning [5] was proposed. When the real brain works some sort of process, the all of neurons do not operate and involved neurons operate as often as required. In our previous research, we have proposed a new network structure with affordable neurons in the hidden layer of the feedforward neural network [6][7]. We named this proposed network "Affordable Neural Network." By computer simulations, the affordable neural network has been confirmed to gain better performance for the BP learning on both convergence speed and learning efficiency. However, we consider that a lot of good points of the affordable neural network are still veiled. In order to make clear many high level abilities, we need to investigate the affordable neural network in detail. We consider that the operation of the affordable neural network is especially effective on evolution of neural network in brain.

In this study, we investigate the performance of the affordable neural network when the hidden layer is evolved using GA. We apply this affordable neural network with GA to learning chaotic time series generated by skew tent map. And we investigate the network producing the same output as the input data. By computer simulations, we consider that the affordable neurons exert an important influence on evolution process in the hidden layer of the network.

2. Affordable Neural Network

We have proposed the network structure with affordable neurons in the hidden layer of the feedforward neural network for more efficient BP learning [?][6]. We introduced the affordable neurons to reflect a function of the brain. The extra neurons in the hidden layer are prepared in advance. During the BP learning, all of the neurons in the hidden layer are not used at every updating. Namely, some of the neurons are selected for the learning and the rest of the neurons are deactivated. The network model with the affordable neurons is shown in Fig. 1.

For the proposed network, some of the neurons have to be selected at every updating of the BP learning. In our previous research, we have investigated three cases "chaos", "regular" and "random" methods as selection of the affordable neurons in the hidden layer. From simulated results, we confirmed that the chaos and the random selection methods gain good performance for some of learning examples. In this study, we use the random method as the selection of the affordable neurons. For the random method, the affordable neurons in the hidden layer are selected completely at random.

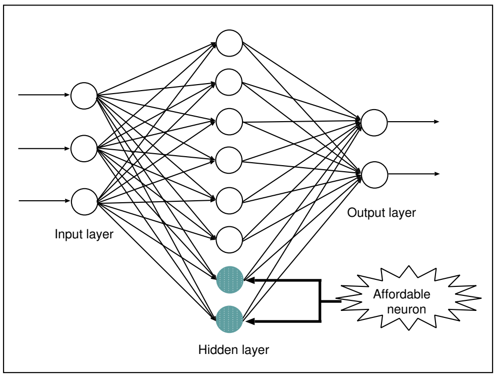


Figure 1: Network model with affordable neurons.

3. Evolving Affordable Neural Network with Genetic Algorithm

Genetic Algorithms were invented to mimic some of the processes observed in natural evolution. Many researchers, including biologists, are astonished that life at the level of complexity that we observe could have evolved in the relatively short time suggested by the fossil record. The idea with GA is to use this power of evolution to solve optimization problems. The father of the original GA was John Holland who invented it in the early 1970's. Holland said, "genetic algorithms have already demonstrated the ability to made breakthroughs in the design of such complex systems as jet engines." We consider that operation of affordable neural network is especially effective on evolution of neural network in brain.

In this section, we explain how to apply GA to the hidden layer structure of the affordable neural network. Correspondence between GA and hidden layer is shown in Tab. 1.

Table 1: Correspondence relationship GA and hidden layer of neural network.

GA	hidden layer of neural network
individual	network structure of hidden layer
population	population of networks
chromosome	weights of neurons in hidden layer
fitness	error for learning data

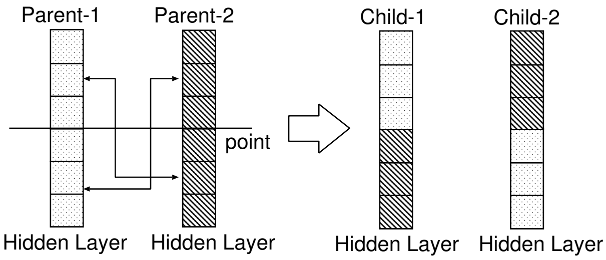
Next, the genetic operation in this study is defined as follows:

1. **Start:** Generate 10 neural networks with weight setting at random.
2. **Learning:** 10 networks learn the learning data while 20 iteration time by BP.
3. **Fitness:** Evaluate the error between output data of the network and learning data.
4. **New network:** Create new networks following steps.
   (a) **Selection:** Chooses 2 networks as parents for the next generation based on their scaled values from the error for the learning data.
   (b) **Crossover:** Cross over the parents to form 8 new children.
   (c) **Copy:** Children is an exact copy of parents.
   (d) **Mutation:** With a mutation probability mutate new children by applying changes to weights between selected 2 neurons of one parent at random.
5. **Test:** If the end condition (= 100 generations) is satisfied, **Stop**.
6. **Loop:** Go to step 2.

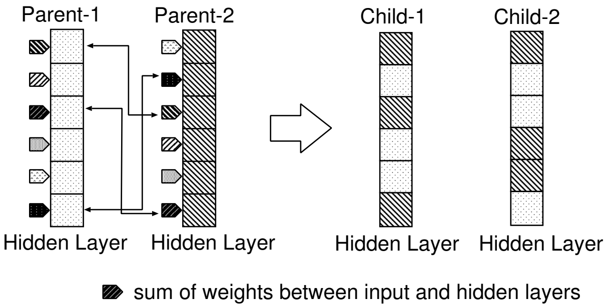
Next, we explain using crossover methods. Figure 2(a) shows the one point crossover. A crossover point on the parent organism string is selected. All worths of neurons in hidden layer beyond that point in the organism string is swapped between the two parent organisms. In this study, we introduce 3 kinds of crossover methods in order to that the networks operate efficiency for good learning and good evolving. Comparing weights of neurons in hidden layer between input and hidden layers of two parents, a certain number of the neurons in the order of the values of weights of neurons are changed (Fig. 2(b)). Similarly, crossover by comparing weights of neurons in hidden layer between hidden and output layers of two parents, and cross over by comparing weights of neurons in hidden layer between input and hidden layers and between and hidden layers show in Figs. 2(a) and (b) respectively. We prepare 10 networks as individuals. The new 10 networks are generated as children of the next generation by the 4 kinds of crossovers and copy of each parent. There are 8 children from the 4 kinds of crossovers. There are 2 children from the copy of the parents. Furthermore, the new 10 children can be mutated with a mutation probability by applying exchange of the weights between selected 2 neurons. The mutation probability is set to 0.01.

4. Simulated Results

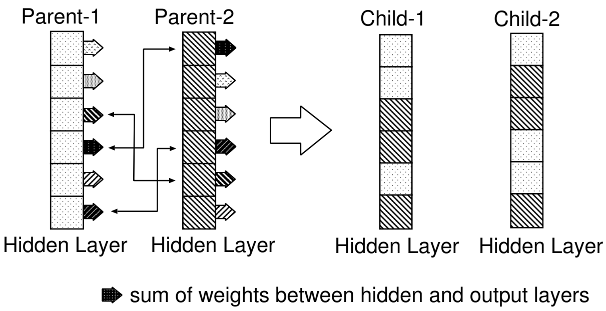
The standard BP learning algorithm was introduced in [5]. The BP is the most common learning algorithm for feedforward neural networks. In this study, we use the batch BP learning algorithm. The batch BP learning algorithm is expressed by similar formula of the standard BP learning algorithm. The difference lies in the timing of the update of the weight. The update of the standard BP is performed after each single input data, while the update of the batch BP is performed after all different input data.



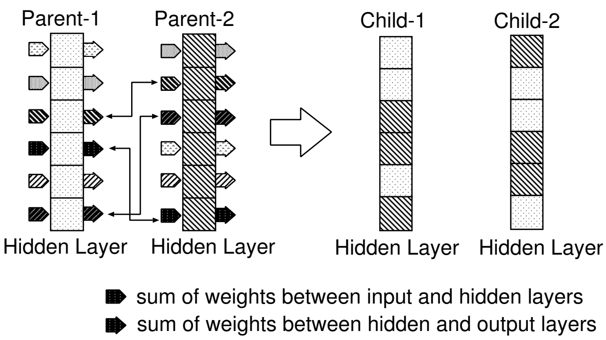
(a) One point crossover.



(b) Crossover by comparing weights of neurons between input and hidden layers.



(c) Crossover by comparing weights of neurons between hidden and output layers.



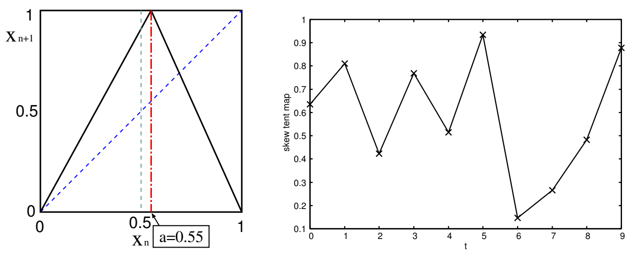
(d) Crossover by comparing weights of neurons between input and hidden layers and between hidden and output layers.

Figure 2: Four kinds of crossover of the hidden layer. (The darkness of the arrows shows the value of sum of the weights.)

4.1. Learning Data and Network Structure

In this study, we consider the feedfoward neural network producing the same chaotic time series as the inputted chaotic time series generated by a skew tent map as one learning example. The skew tent map is defined by the following equation and the map and time series are shown in Fig. 3.

$$x_{n+1} = \begin{cases} \dfrac{x_n}{0.5 + a} & (0 \leq x_n \leq a) \\ \dfrac{-x_n + 1}{1 - a} & (a < x_n \leq 1) \end{cases} \quad (1)$$



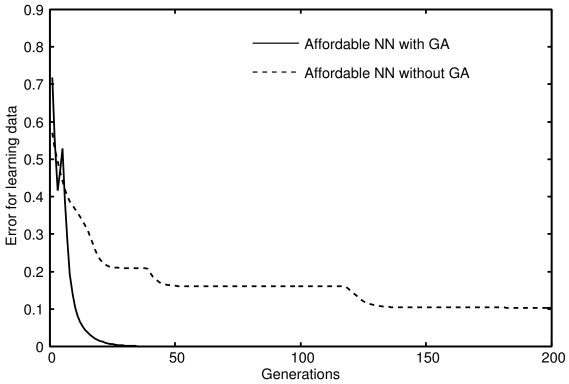
(a) (b)

Figure 3: Learning data. (a) skew tent map. (b) one example of time series.

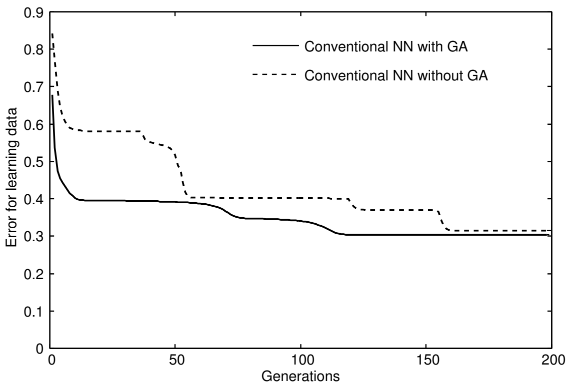
The length of chaotic time series is set to 10 and the number of learning data is set to 20. We carried out the BP learning by using following parameters. The parameter of the learning rate is fixed as $\eta = 0.0002$ and initial values of the weights are given between -1.0 and 1.0 at random. The learning time of one generation is set to 10 and the generation number is set to 100. So the network with GA learn the learning data 1000 times. For comparison, we investigate the performance of the network without GA. In this case, we prepare 10 networks as well as the network with GA. The learning time of each network is set to 1000. We investigate the learning efficiency as the error between the output and the desired target, when the network structure of the hidden layer is evolved by GA. We prepare 20 neurons in the hidden layer and the number of the affordable neurons is set to 6. Furthermore, we investigate the performance of conventional network without affordable neurons to make clear the learning ability of the affordable neural network.

4.2. Performance of Learning Process

Figure 4 shows the simulation result for the case that mutation probability is set to 0.01. The horizontal axis is the number of generations and the vertical axis is the error for the learning data. From this figure, the error of the affordable neural network with GA converge to the optimal solution by increasing the number of generations. However, the network without GA can not escape local minima. We can see that the network with the affordable neurons gains good performance by applying GA as shown in Fig. 4(a). On the other hand, the network without the affordable neurons does not work well by applying GA as shown in Fig. 4(b). From these results, we consider that the affordable neurons exert an important influence on evolution process in the hidden layer of the network.



(a) Affordable neural network.



(b) Conventional neural network.

Figure 4: Generations and error. (mutation probability=0.01).

4.3. Performance for Unknown Data

In this section, after learning of the affordable neural network, we input an unknown chaotic time series generated the same skew tent map as another learning data. We investigate the performance of the affordable neural network producing the same output the input data. The simulated results for unknown chaotic time series after learning of network is shown in Tab. 2. From this table, we can see that the affordable neural network with GA gains best performance for unknown chaotic time series data.

From these results, we can say that the affordable neural network perform well when the GA is applied to the hidden layer of the network both on learning data and unknown data. We consider that the affordable neurons assist to evolve the hidden layer of the network efficiency.

Table 2: Error for unknown chaotic data.

Network structure	Error
Affordable NN with GA	0.45790460
Affordable NN without GA	0.82027979
Conventional NN with GA	1.04816755
Conventional NN without GA	0.98854522

5. Conclusions

In this study, we investigated the performance of affordable neural network when the hidden layer is evolved using GA. We applied this affordable neural network with GA to learning chaotic time series generated by skew tent map. And we investigated the network producing the same output as the input data. By computer simulation, we consider that the affordable neurons exert an important influence on evolution process in the hidden layer of the network.

References

[1] J.H. Holland, "Adaptation in Natural and Artificial Systems," Univ. of Michigan Press, 1975.

[2] T.J. Sejnowski and C.R. Rosenberg, "Parallel Networks that Learn to Pronounce English Text," *Complex Systems*, vol. 1, pp. 145-168, 1987.

[3] M. Kawano, Y. Uno, M. Isobe and R. Suzuki, "A Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics," *IEEE Control Systems Magazine*, vol. 8, pp. 8-16, 1988.

[4] Y. LeCun, B. Boser, J.S. Denker, et al, "Handwritten Digit Recognition with a Back-Propagation Network," *Advances in Neural Information Processing Systems*, vol. 2, MIT Press, MA, 1990.

[5] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, vol. 1, MIT Press, MA, pp. 318-362, 1986.

[6] Y. Uwate and Y. Nishio, "Back Propagation Learning of an Affordable Neural Network for Chaotic Time Series," *Proceedings of RISP International Workshop on Nonlinear Circuits and Signal Processing (NCSP'05)*, pp. 139-142, Mar. 2005.

[7] Y. Uwate and Y. Nishio, "Back Propagation Learning of Neural Networks with Chaotically-Selected Affordable Neurons," *IEEE International Symposium on Circuits and Systems (ISCAS'05)* (to appear).