'95 KACC (1995. 10. 23 ~ 25)

Neural Network Structure Design Using Genetic Algorithm

Junichi MURATA*, °Kei TANAKA*, Masaru KOGA* and Kotaro HIRASAWA*
*Department of Electrical Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812, Japan
Tel : +81-92-641-1101 (Ext 5315), Fax : +81-92-631-2790
E-mail : murata@ee.kyushu-u.ac.jp

Abstract A method is proposed which searches for optimal structures of Neural Networks (NN) using Genetic Algorithm (GA). The purpose of the method lies in not only finding an optimal NN structure but also leading us to the goal of self-organized control system that acquires its structure and its functionality by itself depending on its environment.

Keywords Genetic Algorithms, Neural Networks, Genetic coding, Environment dependence

1. INTRODUCTION

Applications of NN to control and identification are actively studied because of their approximating ability of nonlinear functions. However there is one problem of how to set up NN structures; NN structures have crucial effect on NN performance, but there are a large number of alternative structures. GA can find out a solution to a computationally complex problem of combinatorial optimization, utilizing the searching principle based on evolution and genetic operations, *crossover*, *mutation* and *selection*. Hence GA is adopted for NN structure design. Various methods have been proposed for designing NN by GA. Miller proposed a genetic representation of an NN structure by 0-1 binary genes [1], which gives a strict one-to-one mapping between the genotype and the phenotype (NN). Harp proposed more indirected coding [2]. However the crossover operation is likely to destroy parts of an NN functioning as building blocks. The major issues to be considered are determination of appropriate genetic coding and establishment of suitable fitness functions in GA. Especially, a genetic coding is made up from the point of self-organized control system.

2. GENETIC CODING

According to our purpose mentioned above, a good genetic coding must be such that enables the GA to do the following: to deal with various types of NN nodes with different functions; to find out *building blocks*, i.e. parts of NN which have peculiar functions; and to allow other factors than genome, e.g. the environment surrounding the NN, to play a certain role in determining the NN structure. A new genetic coding is proposed, where the chromosome has two-dimensional structure and can describe the types of nodes and the strength of attracting forces between the specific types of nodes. In addition, an NN structure (phenotype) and a chromosome (genotype) do not have a strict one-to-one correspondence, which leaves a possibility of changes in phenotype depending on the environment.

Hidden layers of a network (a part surrounded by broken lines) shown in **Fig.1.1** is expressed by a chromosome shown in **Fig.1.2 A**. Each small rectangular in chromosome corresponds to each node in the NN. Here, node functions have two different types, sigmoid and linear functions which are denoted by S and L, respectively (see **Fig.1.2 B**). Symbols O_S, O_L, I_S and I_L are defined as follows,

O_S:strength of attracting force to output side S nodes,
O_L:strength of attracting force to output side L nodes,
I_S:strength of attracting force to input side S nodes,
I_L:strength of attracting force to input side L nodes.

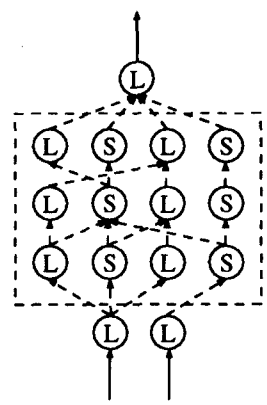

Fig.1.1 network

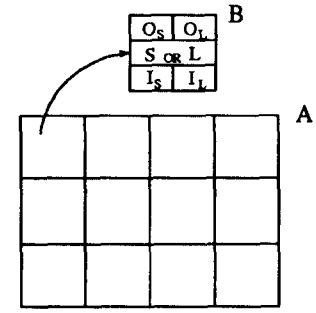

Fig.1.2 chromosome

3. STRUCTURE DESIGN

3.1 Connecting process

Connecting process is composed of three steps. First, two nodes whose distance is the nearest in NN are selected. Second, from the chromosome, the attracting force is calculated which is compared with the distance between the nodes and determines whether the nodes will be connected or not. At last, when the connection is formed, the attracting force decreases.

For instance, let us consider six nodes and their attracting force shown in **Fig.2.1**. We start with the node 6 (function:S). Then, the node 3 (function:S) is

selected because the node 3 is the nearest node to the node 6. Since both the nodes have function S and the node 3 is input side to the node 6, using $I_S(6) = 4$ and $O_S(3) = 3$, the strength of attracting force between these nodes is defined as follows,

$$\frac{I_S(6) + O_S(3)}{2} = \frac{4+3}{2} = 3.5. \quad (1)$$

If this strength of attracting force is larger than the distance between the two nodes, the connection between these nodes will be formed. On the other hand, if the strength is smaller, the connection will not be formed. When the connection is formed, the strength of attracting force of the output side node is reduced by half of the distance. In this example, the connection between the node 3 and the node 6 is formed because the attracting force 3.5 from equation (1) is larger than the distance of these nodes ($d_{36} = 1$). So, $I_S(6)$ decreases as follows,

$$I_S(6) = I_S(6) - \frac{d_{36}}{2} = 4 - \frac{1}{2} = 3.5. \quad (2)$$

Fig.2.2 shows the network structure at this time. After that, the same procedure is carried out for the second, the third nearest nodes and so on (see **Fig.2.3**). These procedures are applied to all nodes, then a network structure is established from the chromosome.

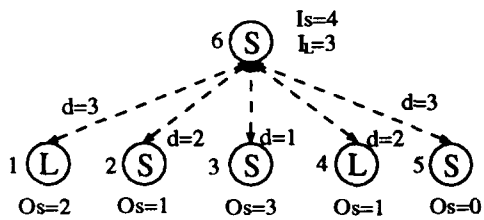

Fig.2.1

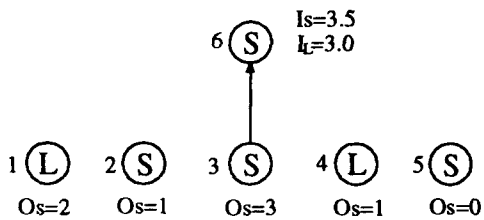

Fig.2.2

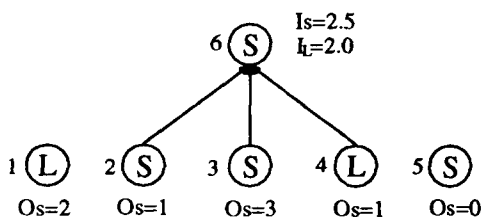

Fig.2.3

Fig.2 connecting process

3.2 Decoding

An NN structure (phenotype) and a chromosome (genotype) do not have a strict one-to-one correspondence, which enables phenotype to be a simple structure or a complex structure depending on its environment in a genetic decoding phase. We propose two concepts that leave a possibility of changes in phenotype depending on the environment.

One is to change the distance between NN nodes depending on the environment.

When the distance is small, it gives a complex NN structure (see **Fig.3.1 Network A**). When the distance is large, it gives a simple NN structure (see **Fig.3.1 Network B**).

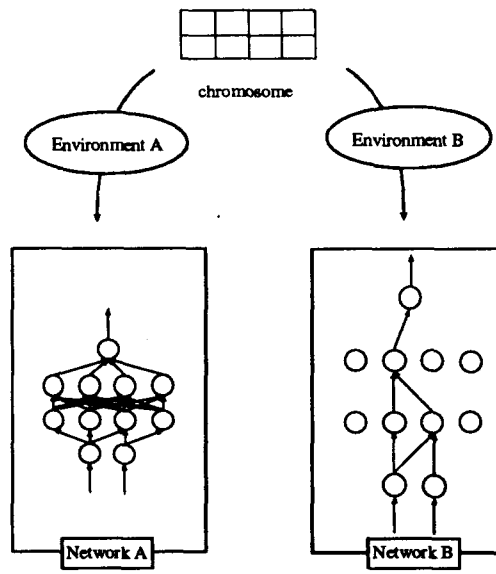

Fig.3.1 decoding based on the NN size

The other is to provide nodes with equal kinetic energy depending on the environment. The attracting force between the nodes is regarded as the source of potential energy. If the potential field around a node captures another node vibrating with the specified kinetic energy at a specified distance, then the nodes will be connected each other.

In case of low kinetic energy, a complex NN structure is established (see **Fig.3.2 Network A**). On the other hand, in case of high kinetic energy, a simple NN structure is established (see **Fig.3.2 Network B**).

These concepts are introduced to imitate biological evolution where life acquires its structure and its function by itself depending on its environment. At the same time, there is another reason in view of an NN ability of approximating datasets.

For example, there are two datasets shown in **Fig.4**. One is an ideal dataset which does not have noise (data A). The other is a noisy dataset (data B). It is necessary for an NN to have many nodes in approximating data A. In approximating data B, an NN which has many nodes may represent the noise contained in data

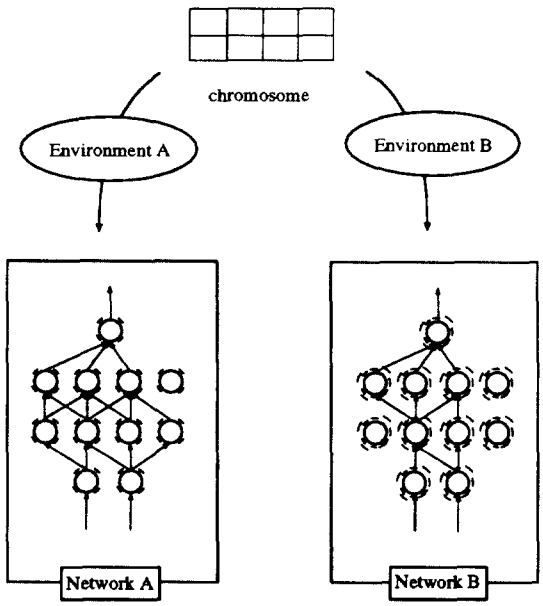

Fig.3.2 decoding based on the NN energy

B (curve X). As far as an NN ability of smooth approximating is concerned, we had better use an NN with fewer nodes for approximating data B (curve Y). Here, we can regard the noise as a part of the environment. If we make the kinetic energy given to each node or the distance between the nodes proportional to the magnitude of the noise, an NN with appropriate number of nodes will be derived through the decoding process.

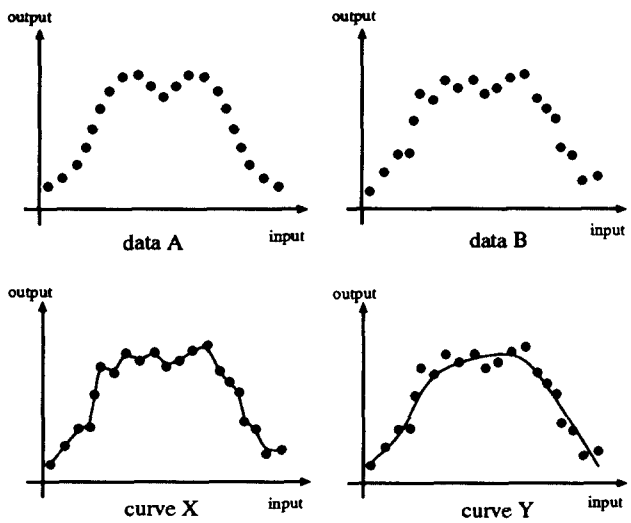

Fig.4 datasets for NN training

4. EVALUATION OF NN STRUCTURES

The qualities of NN structures must be evaluated as fitness values in GA. It is unable to find out an optimal NN structure unless the evaluation is set up appropriately. A new fitness is proposed which regards an accurate but less complicated NN as a good one. The accuracy of NN is measured by the errors between target signals and NN outputs, and the complexity of the structures is quantified by the number of different signals in NN (input signals, neuron signals and output signals).

4.1 The complexity of the NN structure

"The number of all signals + the number of useless signals" is proposed as a standard evaluation of NN complexity. The smaller this standard evaluation is, the less complex the NN is. The number of useless signals is added to the number of all signals to penalize an NN having a useless or redundant part. In the space of signals the uselessness of signals can be measured by their linear dependency. Here matrix S_0 is defined as follows,

$$S_0 = \underbrace{\overbrace{\left[\begin{array}{ccccccccccc} * & \cdots & * & 1 & * & \cdots & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ * & \cdots & * & 1 & * & \cdots & * & * & \cdots & * \end{array}\right]}^{\text{input signals}\quad\text{neuron signals}\quad\text{output signals}}\Bigg\} T}_{n},$$

T:the number of training data.

Using a singular value σ_i of matrix S_0, the equivalent number of linearly independent signals out of n signals in matrix S_0 is

$$(\sum_{i=1}^{n}\sigma_i)^2/\sum_{i=1}^{n}\sigma_i{}^2. \tag{3}$$

A zero singular value corresponds to a linearly dependent column vector, while a large value means linear dependency. So, the sum of all the singular values represents the number of independent vectors provided that it is normalized by the sum of the norms of the vectors which is equal to $\sum\sigma_i^2$. From the above, we get,

$$\begin{aligned} & \textit{all signals} + \textit{useless signals} \\ = \; & \textit{all signals} + \\ & \{\textit{all signals} - \textit{linearly independent signals}\} \\ = \; & n + \left\{ n - (\sum_{i=1}^{n}\sigma_i)^2/\sum_{i=1}^{n}\sigma_i{}^2 \right\} \\ = \; & 2n - (\sum_{i=1}^{n}\sigma_i)^2/\sum_{i=1}^{n}\sigma_i{}^2. \end{aligned} \tag{4}$$

4.2 The accuracy of NN

The accuracy of NN is measured by the errors between target signals and NN outputs. A vector e_j is defined as follows,

$$e_j = \left[\begin{array}{c} \text{output signal}_j(1) - \text{target signal}_j(1) \\ \vdots \\ \text{output signal}_j(T) - \text{target signal}_j(T) \end{array}\right],$$

$j = 1, \cdots, p$,
p:the number of target signals.

Let us consider an augmented matrix S,

$$S = [S_0|e_1, \cdots, e_p].$$

If the errors are all zero, the value of equation (4) calculated for matrix S will be equal to that for S_0. However if the errors are not zero, the denominator will increase by introduction of $\|e_1\|^2 + \cdots + \|e_p\|^2$. This increase is a penalty for the non zero errors. Thus, we define the quality measure C of NN as,

$$C = 2n - \frac{(\sum_{i=1}^{n} \sigma_i)^2}{\sum_{i=1}^{n} \sigma_i^2 (1 + R_1 \sum_{j=1}^{p} \|e_j\|^2)}. \quad (5)$$

5. SIMULATIONS

Simulations are carried out to design an NN structure (one input, one output and one hidden layer which has at most ten nodes), using two training datasets; noiseless one and noisy one shown in **Fig.5.1**. The noise is regard as the environment surrounding the NN. In decoding, the distance between the nodes in the NN is defined to be proportional to the magnitude of the noise (environment).

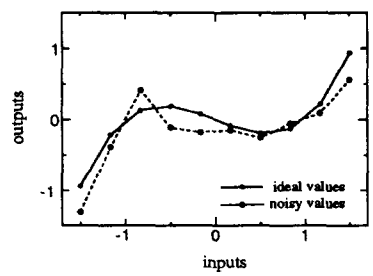

Fig.5.1 datasets for NN training

Fig.5.2 shows an NN which is given by GA operation in the case that the training dataset does not have noise. And **Fig.5.3** shows the NN outputs at that time. This obtained NN is one of the best networks trained using noiseless dataset because the NN has small errors and simple structure.

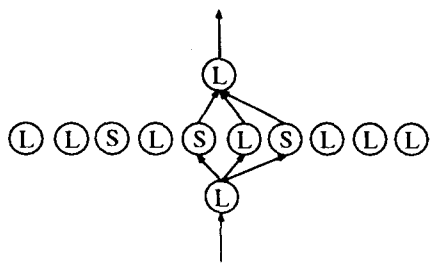

Fig.5.2 the obtained NN structure

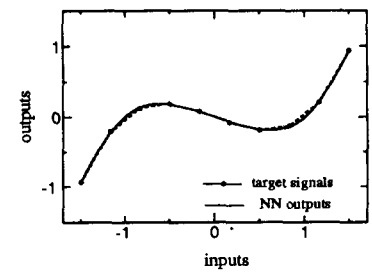

Fig.5.3 the relation of inputs and outputs

Fig.5.4 and **Fig.5.5** show an NN structure and NN outputs in the case of the noisy training dataset. This obtained NN approximates the noisy dataset smoothly.

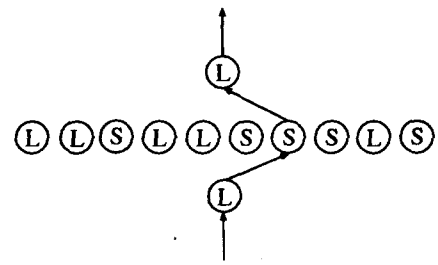

Fig.5.4 the obtained NN structure

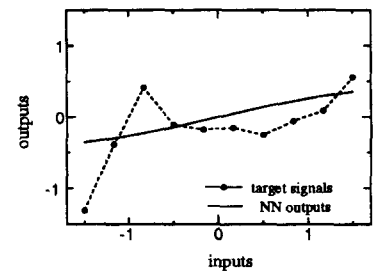

Fig.5.5 the relation of inputs and outputs

6. CONCLUSION

In this paper, A method is proposed for finding an optimal NN structure using GA automatically.

Simulation results show the effectiveness of the proposed method in NN structure design. The method is able to design an NN containing two different types of node functions, sigmoid and linear functions; owing to the proposed fitness, the obtained NN has small errors and a simple structure according to the environment.

REFERENCES

[1] G. Miller, P. Todd and S. Hegde, "Designing Neural Networks using Genetic Algorithms," *Proc. of ICGA-89*, 1989.

[1] S. Harp, T. Samad and A. Guha, "Towards the Genetic Synthesis of Neural Networks," *Proc. of ICGA-89*, 1989.