Growing neural network with hidden neurons

Ryusuke KURINO, Masanori SUGISAKA and Katsunari SHIBATA
Dept. of Electric and Electronic Engineering, Oita University
700 Dannoharu Oita 870-1192 Japan. Email: shibata@cc.oita-u.ac.jp

Abstract

The authors have proposed “growing neural network” in which axons of neurons grow according to the concentration gradient of a chemical substance that are diffused from the output neurons in proportion to the error signal. In this paper, we propose a new technique that enables to obtain an appropriate structure including hidden neurons. The hidden neurons diffuse another chemical substance when they don’t have enough connections with the input neurons. In a simulation, it was examined that an appropriate structure could be formed through the learning of the most popular and simplest non-linear separation problem of “EXOR”.

1. Introduction

Artificial neural networks (NN) are broadly used to approximate non-linear functions because of its advantage of the learning and generalization ability. When we use the artificial NN for function approximation, three-layer structure is usually employed. The reason is that it is difficult to decide an appropriate structure adaptively for a given problem. Moreover, it is well known that 3-layer neural network with enough number of hidden neurons can approximate any continuous functions with any precision[1][2].

However, in the 3-layer NN, each hidden neuron represents just a linear and smooth classification of the input space, and it is difficult to represent complicated abstract information. On the other hand, in our humans, high order functions, which include recognition, memory, and control, are supported by the complicated structure of NN in the brain. Such functions are realized as a various levels of abstract state representation, the authors think, and it is very useful for the effective learning based on generalization. Useful abstract state representation can be formed in the hidden neurons when a series of processing from sensors to motors is acquired through learning such as reinforcement learning. That must be the origin of intelligence by which the humans are distinguished from modern robots, the authors believe. To realize an intelligent robot with such functions, autonomous acquisition of an appropriate structure of the NN including recurrent structure will be required.

It is known that in the brain of living things, the number of neurons decreases after its birth, while the number of connections between neurons increases rapidly[3]. This suggests a strategy in the brain as follows. More neurons than necessity are prepared at first, then they grow their axons and learn their synapse weights, and an appropriate network structure is formed flexibly. After that, unnecessary neurons are removed by apoptosis. It has been also reported that chemical substances such as NGF(Nerve Growth Factor) make a role of promoting the growth of axons[3].

The authors have proposed the growing NN that introduced the concept of growth of neuron into the conventional NN as an extension of learning[4]. Getting a hint from NGF mentioned above, axons grow according to the concentration gradient of a chemical substance which is diffused according to the error signal propagated in BP (Back Propagation) learning. Since the error signal is diffused as the chemical substance, it is expected that an appropriate structure for a given problem can be obtained. As the first step, simple logical functions “AND” and ”OR” were learned by the network, and two-layer structure could be obtained.

However, by two-layer structure, it is well-known that only liner separation problems can be learned. In order to solve the other type of problems that need non-linear operation, hidden neurons are necessary. The hidden neurons cannot make connections to the other neurons by the following two reasons. (1) The hidden neurons cannot grow its axons appropriately since the hidden neurons do not have the information of the input signals. (2) The input neurons cannot grow its axons to the hidden neurons since the hidden neurons do not have the information of the error signal. In order to form the network with hidden neurons, it is necessary for the hidden neurons to make connections with the input or the output neurons by some method.

In this paper, we propose a new technique that enables to obtain an appropriate structure including the hidden neurons. The hidden neurons diffuse another chemical substance not depending on the error signal and ask for the connections from nearby input neurons when they don’t have enough connections with the input neurons. In a simulation, it is examined whether an appropriate structure can be formed through the learning of the most popular and simplest non-linear separation problem of “EXOR”.

2. Growing NN with hidden neurons

2.1 Fundamental algorithm

In the growing NN with hidden neurons, the growth is formulated as an extension of BP learning. Fig.1 shows the idea of the growing NN with the hidden neurons. Fig.2 shows the flowchart of the algorithm. At first, the output and the error are calculated, and the error signal is propagated backward. If the output neuron does not have enough connections, the neuron diffuses the signal as a chemical substance. Then, the concentration gradient is formed by the diffusion around the output neuron. The growing neuron extends its axon according to the concentration gradient.

The hidden neurons that do not have enough connections diffuse another substance constantly not depending on the error. By the diffusion of the hidden neurons, the input neurons extend its axons and make connections to the hidden neurons at first. After that, the weight is increased gradually until the hidden neuron makes a connection to an output neuron. The hidden neuron that has formed the connections with the input neurons extends its axon to an output neuron referring to the input signals. From just after making the connection (synapse), the learning according to the regular BP algorithm begins.

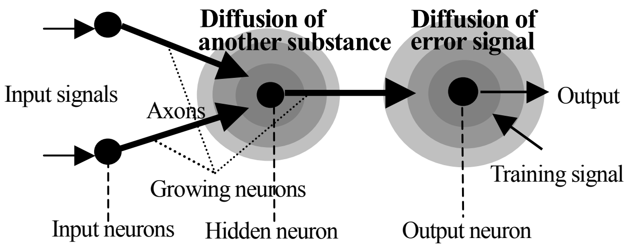


Figure 1: Growing NN with hidden neurons.

1. Decision of each neuron's position.
2. Decision of whether each connection has been made or not.
3. Setting of input and training signals.
4. Calculation of output and error.
5. Back propagation of the error signal.
6. For the neuron that can get the error signal:
   Diffusion of the error signal.
   For the other neurons: Diffusion of the other substance.
7. Calculation of the axon growth.
8. Judge of whether each connection has been made or not.
9. Update of connection weights(Only for connected neurons.)
10. Return to step 3.

Figure 2: Flowchart of the growing NN.

2.2 Diffusion

In the growing NN, the error signal is diffused as a chemical substance around the output neuron. Same as the previous paper[4], the substances for the positive error and that for the negative error are prepared. The error signal is calculated as

$$\delta_j = -\frac{\partial E}{\partial net_j} = (d_j - o_j) \cdot f'(net_j) \qquad (1)$$

where $j = 0,..,NODE_j$, $NODE_j$: the number of the output and the hidden neurons, δ_j: the error signal, net_j: the internal state of the neuron j, o_j: the output, d_j: the training signal, $f'(net_j)$: the derivative of the output function, and E: the error function. There are three kinds of substances here, and the concentration of each substance is denoted by u^p, u^n or u^h where p denotes "positive error", n denotes "negative error" and h denotes "from hidden". The diffusion of each substance is calculated according to the diffusion equation as

$$\frac{\partial u_{x,y}}{\partial t} = div + D\nabla^2 u_{x,y} \qquad (2)$$

where $u_{x,y}$: the concentration at (x,y), div: the divergence of the substance and D: a diffusion constant. At the place of the output neuron, when $\delta > 0$, $div = \rho\delta$ (ρ: a divergence constant) for u^p, and when $\delta < 0$, $div = -\rho\delta$ for u^n. Otherwise, $div = 0$. At the place of the hidden neurons, $div = \alpha$ (α: a divergence constant) for u^h, and $div = 0$ for u^p and u^n. At all the other places, $div = 0$ for all the substances.

2.3 Extension of the axon

The axon of a growing neuron extends according to the state of the neuron and the concentration gradient at the tip of the axon. Two types of neurons are prepared here. One of them called "positive neuron" makes only a positive connection. While, the other called "negative neuron" makes only a negative connection. Here, in order to make the growth of the axon smooth, the first-order delay $\boldsymbol{A}_i$ is introduced to the growth of neurons. The extension of the axon is calculated as

$$\tau_a \frac{d\boldsymbol{A}_i}{dt} = -\boldsymbol{A}_i + (\nabla(u^p - u^n) \cdot flag_i + \nabla u^h) \cdot S_i \qquad (3)$$

$$\frac{d\boldsymbol{a}_i}{dt} = \xi \cdot \boldsymbol{A}_i \qquad (4)$$

where τ_a: a time constant, $\boldsymbol{a}_i$: the position vector of the tip of the axon, ξ: a growth constant and $i = 0,\ldots,NODE_i$, $NODE_i$: the number of the growing neurons that include the input and hidden neurons. The time constant is large so as to calculate the temporal average of the growth of the axon. For the positive neurons, $flag = 1$, while for the negative neurons, $flag = -1$. The state of a growing neuron S is defined as the first-order delay of the output of the neuron. The first-order delay is introduced to adjust the delay of the error signal with a time constant τ_s. Moreover, since the hidden neurons could not extend its axon by the concentration gradient of a self-diffusion ∇u^h, the effects of the diffusion are removed in Eq (3) only for the hidden neurons.

2.4 Update of the connection weight

A connection weight is always 0 before the connection is formed. When the connection is formed, the synapse begins the learning from 0 connection weight. The update of the weight is calculated as well as the regular BP leaning as

$$\frac{dw_{ji}}{dt} = -\eta \cdot \frac{\partial E}{\partial w_{ji}} = \eta \cdot \delta_j \cdot o_i \qquad (5)$$

where w_{ji}: the connection weight from the neuron i to the neuron j and η: a learning constant.

If the hidden neuron has not made the connection to the output neuron yet when the input neurons formed the connection to the hidden neuron, the neuron cannot update the weight according to the error. Then, the weight is increased monotonically and gradually not depending on the error. The weight change is calculated using an exponential function as

$$\frac{dw_{ji}}{dt} = \gamma \exp^{-\lambda \sum_j |w_{ji}|} \cdot flag_i \tag{6}$$

where γ: a learning constant and λ: a constant that influences the slope of the exponential curve. The reason why the exponential function is used is to avoid that the weight becomes too large. Moreover, when the weight becomes less than 0 in a positive neuron, the weight is set to 0. When it becomes more than 0 in a negative neuron, it is set to 0. When the hidden neuron does not have the connection to the output neuron, biases are changed with a large time constant τ_θ so as that the temporal average of the neuron's output becomes the middle value 0.5. Otherwise, they are calculated by BP learning.

3. Simulation

3.1 Set up

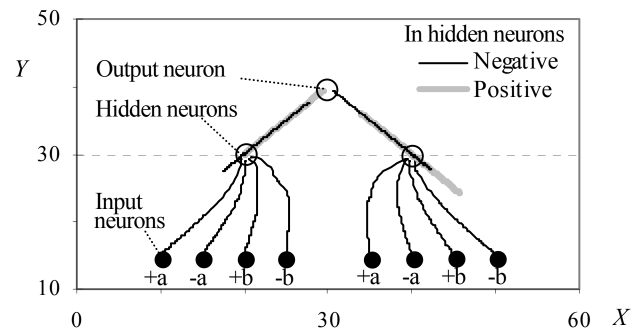
By using the growing NN with the hidden neurons, the non-linear separation problem "EXOR" was learned. The simulation is done on 0.6(mm)$\times$0.6(mm) of plane. The concentration is calculated by difference on each point on a 61×61 grid. There are eight input neurons. The position of each neuron can be found in Fig.3. All the neurons do not have any connections beforehand. Among the input neurons, the input '+a' and '+b' indicate the positive neurons, the input '-a' and '-b' indicate the negative ones. It is assumed that each positive hidden neuron and one of the negative hidden neurons are located at the same position, and the input neuron makes the connections with the both neurons simultaneously. The input and the training pattern can be seen in Table1. The patterns are presented in this order and each one is presented for 1(sec). The calculation as shown in Fig.2 is done for every 0.001(sec). The output function is sigmoid ranges from 0 to 1. The parameters used in the simulation are as follows. The diffusion constant D = 0.1(mm^2/sec), the divergence constant ρ = 10.0(1/sec), α = 0.15(1/sec) for hidden neurons, the growth constant ξ = 0.1(mm^2/sec), the time constant τ_a = 10 (sec) τ_s = 0.1(sec), τ_θ = 10 (sec), the leaning constant η = 0.2(1/sec), γ = 0.001(1/sec), the constant λ = 0.15.

3.2 Results

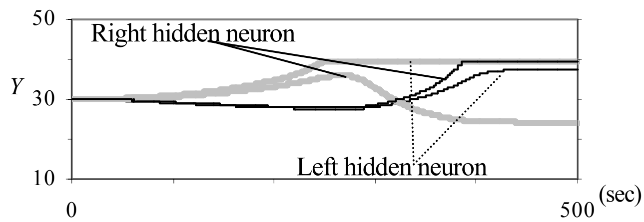
Fig.3 shows how the axon of each input or hidden neuron grows. At first, the axon of each input neuron extends according to the diffusion of the hidden neurons. Since the diffusion is constant, the axon of each input neuron continues to grow towards the closer hidden neuron. The inputs '-a' and '+b' that is close to the hidden neurons formed the connections earlier. The loci of the axons of the input neurons are curved by the influence of the other hidden neuron. After all the input neurons made the connections to the hidden neurons, the positive hidden neurons grew its axons, and the negative hidden neurons grew in the opposite direction at first. The left positive hidden neuron formed the connection to the output neuron at 246(sec). After this connection, the right positive neuron inverted the direction of the growth, and the negative hidden neurons grew to the output neuron. Then the right negative hidden neuron formed the connection to the output neuron at 387(sec). After this connection, the left negative hidden neuron and the right positive one stopped to grow.

Fig.4 shows the change of the error. The arrows in the figure indicate the timing when a hidden neuron formed the connection to the output neuron. Fig.5, 6, 7 show the change of the (input - left positive hidden), (input - right negative hidden) and (hidden - output) connection weights respectively. Table1 shows the error for each input pattern roughly. Error (a) indicates the error when there is no connection between the hidden and output neurons. Error (b) indicates the error after the left positive hidden neuron made the connection to the output.

When there is no connection between the hidden and output neurons, the error decreases only by the learning of the bias of the output neuron. When the input pattern changes from (1) to (2), the output is still 0.1 since the bias does not change instantaneously. Then, the positive error appears as shown in Table1. However, the error decreases soon by the learning of the bias. When the input pattern changes from (3) to (4), the negative error appears since the output is still 0.9.



(a) The loci of the axon of each neuron.



(b) The change of growth of the hidden neuron.

Figure 3: Growth of each neuron.

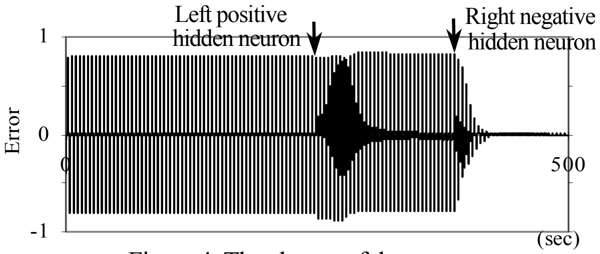


Figure 4: The change of the error.

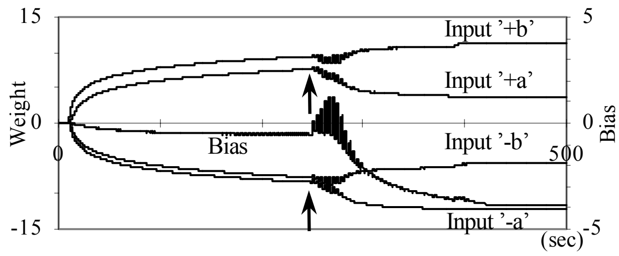


Figure 5: The change of the weight between the input and the left positive hidden neurons.

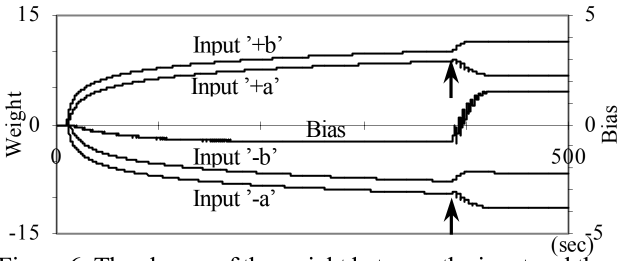


Figure 6: The change of the weight between the input and the right negative hidden neurons.

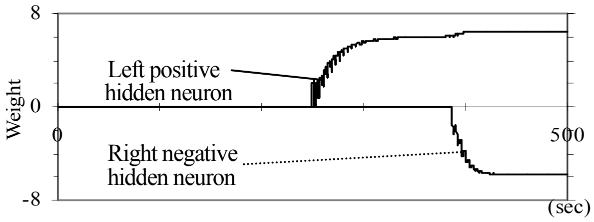


Figure 7: The change of the weight between the output and the hidden neurons.

Table1: The training signal and the error for each input pattern.

Pattern	Input'a'	Input'b'	Training	Error(a)	Error(b)
(1)	0	0	0.1	0	0
(2)	0	1	0.9	Positive	0
(3)	1	0	0.9	0	Positive
(4)	1	1	0.1	Negative	Negative

When each hidden neuron formed the connection to the output neuron, the learning of the connection weight according to the error signal begins. As shown in Fig.5, because the left positive hidden neuron made the connections with the input '-a' and '+b' earlier, their weights are larger than those of the input '+a' and '-b'. Accordingly, in case of the error (b), when the input pattern changes from (1) to (2), the input '+b' changes from 0 to 1. Then, the output of the hidden neuron increases and the output becomes larger than 0.1. Thus, for the pattern (2), the positive error decreases and finally becomes almost 0. On behalf of it, when the input pattern changes from (2) to (3), since the input '-a' changes from 0 to 1 and the input '+b' changes from 1 to 0, the output of the hidden neuron decreases and the output becomes less than 0.9. Thus, the positive error increases, and that promotes the right negative hidden neuron to grow its axon toward the output neuron. Moreover, the negative error increases slightly for the pattern (4) as shown in Fig.4 even though the connection from left positive neuron has been formed. When the input '+b' changes from 0 to 1 on Table1, the output of the hidden neuron increases and becomes larger than 0.9, and the negative error increases slightly. After the right negative hidden neuron formed the connection, the error finally becomes almost 0. Accordingly, the connection weight converged, and the left negative hidden neuron and the right positive one stopped to grow.

After the learning, the left positive hidden neuron was excited only for the pattern (2) since its bias of the hidden neuron is negative. The right negative hidden neuron was excited for the pattern (1), (2) and (4) since the bias of the hidden neuron is positive. The output neuron was inhibited by the right negative hidden neuron for the pattern (1) and (4), but was excited by the positive bias of the output neuron. The bias that is not shown in the figures is 2.4.

4. Conclusion

In this paper, a new technique that enables to obtain an appropriate structure including the hidden neurons was proposed. When the hidden neurons do not have enough connections, the hidden neurons diffuse the substance not depending on the error signal constantly. In a simulation, the input neurons formed a connection to the hidden neurons at first, and then three-layer structure could be obtained through the learning. There still remain many problems to be solved, for example, parameter tuning is necessary to modify the connection weight in the hidden neurons that do not have enough connections to the output neuron.

Acknowledgement

This research was partially supported by the Japan Society for the Promotion of Science, Grants-in-Aid for Scientific Research, #14350227 and # 15300064.

Bibliography

[1]K.Hornik, Multiayer Feedforward Networks are Universal Approximators, Neural Networks, Vol.2, pp. 359-366(1989)

[2]K.Funahasi, On the Approximate Realizaton of Continuous Mappings by Neural Networks, Neural Networks, Vol.2, pp.183-192 (1989).

[3]T.Tsumoto, Brain and Development, Asakura Publishing Co.,Ltd (1986). (in Japanese)

[4]R.Kurino, M.Sugisaka, K.Shibata, Growing neural network for acquisition of 2-layer structure, Proc. of IJCNN'03, pp.2512-2518 (2003).