

Universal Learning Networks with Branch Control

Hirasawa, Kotaro

Department of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

Hu, Jinglu

Department of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

Xiong, Qingyu

Department of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University : Graduate Student

Murata, Junichi

Department of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

<https://doi.org/10.15017/1513717>

出版情報 : 九州大学大学院システム情報科学紀要. 4 (2), pp.125-131, 1999-09-24. Faculty of Information Science and Electrical Engineering, Kyushu University

バージョン :

権利関係 :



Universal Learning Networks with Branch Control

Kotaro HIRASAWA*, Jinglu HU*, Qingyu XIONG** and Junichi MURATA*

(Received June 21, 1999)

Abstract: In this paper, Universal Learning Network with Branch Control (ULN with BC) is proposed, which consists of basic networks and branch control networks. The branch control network can be used to determine which branches of the basic network should be used. This determination depends on the inputs or the network flow of the basic network. Therefore, by using the ULN with BC, locally functions distributed networks can be realized depending on the values of the inputs of the network or the information of the network flow. The proposed network is applied to some function approximation problems. The simulation results show that the ULN with BC exhibits better performance than the conventional networks with comparable complexity.

Keywords: Neural networks, Fuzzy networks, Universal learning networks, Functional distribution, Generalization ability

1. Introduction

Commonly used neural networks, especially layered neural networks, have the structure such that all input nodes and output nodes are connected to the intermediate nodes. Therefore, functions distribution of networks can not be realized in the conventional networks. Here, functions distribution means that only parts of the network are activated depending on the input values of the network or the information of the network flow.

On the other hand, from the recent information on brain science, it has been recognized that in the brain, there are many special intellectual parts related to mathematics knowledge, music knowledge, gymnastic knowledge and so on, and each intellectual part is activated depending on the action of each knowledge. This is called "functions are locally distributed in the brain".¹⁾

In this paper, an artificial functions distributed network is proposed and it is studied how much the proposed network can improve the performance of the network compared to the conventional networks.

Now, there has been already proposed "Learning Petri Network (LPN)"^{2),3),4),5)} as functions locally distributed networks in our laboratory, which is a combination of Petri networks and neural networks. And in LPN functions distribution can be realized by using the token control of Petri Networks and its learning can be executed by the back propagation learning method of neural networks.

On the other hand, the proposed network named Universal Learning Network with Branch Control (ULN with BC) is composed of two kinds of networks such as basic networks and branch control networks. And functions distribution of the basic network is materialized by connecting or disconnecting the branches of the basic network following the instructions from the branch control network in ULN with BC.

Moreover, modular networks^{6),7)} have similar module structures to ULN with BC. But, the structure of the modular networks is a little bit different from the one of ULN with BC. While the modules in modular networks are completely separated, ULN with BC has the module structure where the different modules have the identical nodes and branches, in other words, the modules in ULN with BC are overlapped. Therefore more compact networks are expected to obtain by ULN with BC.

In this paper, Universal Learning Network (ULN)⁸⁾ is used as a framework for constructing ULN with BC because ULN is a generalized network which can deal with neural networks, radial basis function networks, fuzzy inference networks and so on. And a neural network and a fuzzy inference network are adopted as the basic network and branch control network respectively.

Simulations of ULN with BC show very interesting results that conventional neural networks with all the branches being connected can not always exhibit the best performance.

2. Universal Learning Network with

* Dept. of Electrical and Electronic Systems Engineering

** Dept. of Electrical and Electronic Systems Engineering,
Graduate Student

Branch Control

2.1 Basic Structure

Universal Learning Network with Branch Control (ULN with BC) is a network whose branches are connected or disconnected depending on the values of the inputs and the information on the network flow. Therefore functions locally distributed networks can be realized, and representation and generalization ability of the network can be expected to improve by ULN with BC.

The structure of ULN with BC is characterized by the following:

- ULN with BC is composed of basic networks and branch control networks, and the branch control network can determine which branches of the basic network should be connected or disconnected (see **Fig.1**).
- ULN with BC is structured in such a way that there is one to one correspondence between the output of the branch control network and the branch of the basic network. As a result, the output of the branch control network can control the connection and disconnection of the corresponding branch of the basic network.
- The input of the branch control network could be the input and/or the information on the flow of the basic network.

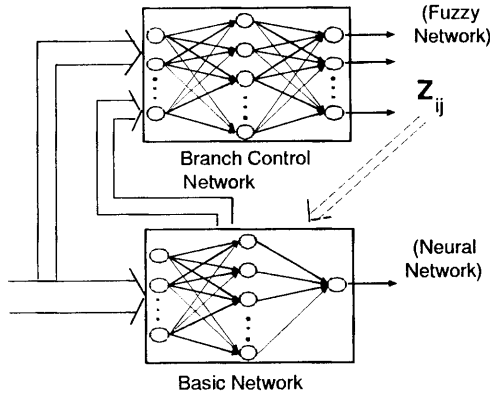


Fig.1 Structure of Universal Learning Network with Branch Control

In this paper, simulations were carried out adopting the following neural networks and fuzzy inference networks as the basic network and branch control network respectively.

- Neural Networks(x : input, y : output)

$$y = \sum_{m \in M} f_m(x) \lambda_m \quad (1)$$

$$f_m(x) = \frac{1 - e^{-\varphi_m(\omega_m^T x + \beta_m)}}{1 + e^{-\varphi_m(\omega_m^T x + \beta_m)}} \quad (2)$$

$$\omega_m = (\cdots, \omega_{mu}, \cdots)^T$$

$\omega_m, \varphi_m, \beta_m, \lambda_m$: parameters of sigmoidal function

M : set of suffixes of intermediate nodes in neural networks

- Fuzzy Inference Networks(x : input, Z_{ij} : outputs)

$$Z_{ij} = \frac{\sum_{q \in Q} [f_{xq}(x) \sigma_{ijq} \mu_{ijq}]}{\sum_{q \in Q} [f_{xq}(x) \sigma_{ijq}]} \quad (3)$$

$$f_{xq}(x) = \exp\left[-\frac{(x - \mu_{xq})^T \sum_{xq}^{-1} (x - \mu_{xq})}{2}\right] \quad (4)$$

$$f_{ijq}(Z_{ij}) = \exp\left[-\frac{(Z_{ij} - \mu_{ijq})^2}{2\sigma_{ijq}^2}\right] \quad (5)$$

$f_{xq}(x)$: membership function of the IF part of the fuzzy rule

$f_{ijq}(Z_{ij})$: membership function of the THEN part of the fuzzy rule

μ_{xq} : vector representing the mean of the IF part of the fuzzy rule

\sum_{xq}^{-1} : matrix representing the variance of the IF part of the fuzzy rule

μ_{ijq} : parameter representing the mean of the THEN part of the fuzzy rule

σ_{ijq}^2 : parameter representing the variance of the THEN part of the fuzzy rule

Q : set of suffixes of fuzzy rules

In **Fig.1**, B_{ij} stands for the branch from node i to node j in the basic network and Z_{ij} denotes the output of the branch control network, which corresponds to the branch B_{ij} . The connection or disconnection of branch B_{ij} is determined by the following way using the value of Z_{ij} .

- if $Z_{ij} \geq Z_{ij}^o$ then B_{ij} is connected
 - if $Z_{ij} < Z_{ij}^o$ then B_{ij} is disconnected
- where Z_{ij}^o : threshold value

One of the reasons of using the fuzzy inference network as the branch control network is that neighboring points in the input space should be mapped also to the neighboring points in the output space in the branch control network in order to execute the branch control effectively.

Therefore, by adopting the fuzzy inference network as the branch control network, a functions locally distributed network can be realized, where the branches of the basic network are connected or disconnected depending on the input values or the information on the network flow of the basic network.

Although Learning Petri Network (LPN)²⁾ and modular networks⁶⁾ have been also developed to materialize functions locally distributed networks as ULN with BC, the ways of realizing them are dif-

ferent. While the flow of tokens in LPN and expert networks designed for special functions in modular networks can realize the functions distribution, the branch control of the basis network is carried out by the instructions from the branch control network in ULN with BC.

Therefore, ULN with BC does not require a fairly complicated token control peculiar to LPN, and also ULN with BC can construct varying modules depending on the input conditions instead of the fixed expert modules in modular networks.

In this paper, for simplicity, simulations were carried out on condition that the inputs to the branch control network are the same as the ones to the basic network, in other words, the network flow of the basic network is not taken into account when calculating the output values of the fuzzy inference network.

2.2 Learning

As for learning, parameters of the basic network are trained in the same way as the commonly used neural networks with supervised learning.

But, as branch connection or disconnection of the basic network in ULN with BC depends on the input values of the network, it is difficult to train the parameters by using the gradient method like back propagation algorithm.

So, a kind of random search method named RasID⁹⁾ was used to train the parameters of the basic network. RasID is an abbreviation of Random Search with Intensification and Diversification, and executes the search for optimal parameters in a unified manner, where intensified search and diversified search are carried out systematically and effectively using the information on success and failure of the past search.

As for RasID learning, each criterion function for each input point was summed up, and parameters of the basic network were trained in order to minimize the above sum of the criterion functions.

The following branch connectivity of the basic network can be adjusted by setting the threshold Z_{ij}^o at an appropriate value

$$\rho = \frac{n}{N} \quad (6)$$

where n is the numbers of the connected branches and N is the total numbers of branches of the basic network. The branch connectivity ρ is increased when Z_{ij}^o takes a small value, and vice versa.

When functions locally distributed networks are

used for many applications, it is generally expected that there exists a connectivity where the criterion function is minimized like **Fig.2**. This means that higher performance can be obtained than the commonly used networks by using the network with some branches being cut depending on the input values of the network. This fact has been already verified by using Learning Petri Network developed in our laboratory.

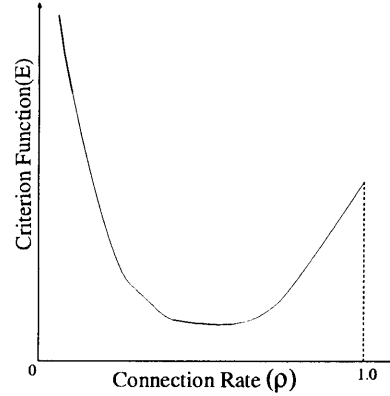


Fig.2 Relation between connection rate and generalization ability

The reason why the optimal connectivity exists in the networks is that it is appropriate to use the compact partial networks rather than the total big networks for the partial problems. In other words, it can be stated that since the compact partial networks have a small number of parameters, parameter training of the ULN with BC is expected to carry out smoothly.

3. Simulations of Universal learning Network with Branch Control

Simulations were carried out by adjusting the threshold Z_{ij}^o in order to study the fundamental characteristics of connectivity ρ of ULN with BC.

3.1 Simulation Conditions

The parameters of the membership functions of the IF parts of the fuzzy rules $f_{xq}(x)$ are fixed in such a way that the input space of the network is appropriately covered with 5 fuzzy rules as follows:

$$\mu_{xq} = (0.25, 0.25), (0.25, 0.75), (0.75, 0.25), \\ (0.75, 0.75), (0.5, 0.5)$$

\sum_{xq}^{-1} is a diagonal matrix with its element being $\frac{1}{0.03}$, σ_{ijq} is equal to 1.0 and μ_{ijq} is random numbers in $[0,1]$.

And, simulations were done to study whether ULN with BC can realize the functions locally distributed network or not, and how much the repre-

sensation and generalization ability can be improved by the functions distribution.

Simulations are executed by using the following three kinds of two dimensional functions approximation problems which are defined on $0 \leq x \leq 1, 0 \leq y \leq 1$ (see **Fig.3**).

- Function 1 :

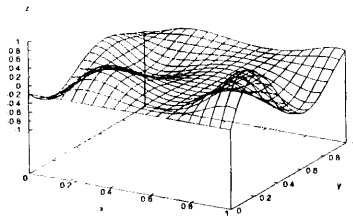
$$f(x, y) = 0.95((1.35 + \exp(x)\sin(13.0(x-0.6)^2) \cdot \exp(-y)\sin(7y)) - 2.5)/2.0 - 0.2$$
- Function 2 :

$$f(x, y) = 0.79((1.5(1.0 - x) + \exp(2x - 1.0) \cdot \sin(3\pi(x - 0.6)^2) + \exp(3.0(y - 0.5)) \cdot \sin(4\pi(x - 0.9)^2)) - 3.0)/5.0$$
- Function 3 :

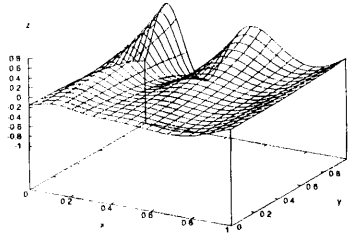
$$f(x, y) = \text{bell}(a, b, c, x) \cdot \text{bell}(a, b, c, y) - 0.5$$

where $\text{bell}(a, b, c, x_i) = \frac{1}{1 + |\frac{x_i - c}{a}|^{2b}}$

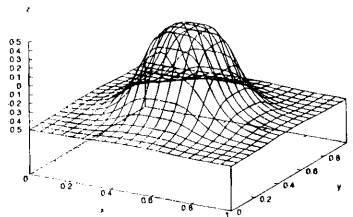
($a = 0.2, b = 2.0, c = 0.5$)



Function1



Function2



Function3

Fig.3 Shape of function

The above three functions have different shapes characteristic. Function 1 has an uniform charac-

teristic on all areas under consideration, the shape of function 2 changes sharply as y increases and function 3 is a two dimensional bell function.

The basic networks are a three layered neural network with the number of the intermediate nodes being 100, 50, 20 and 5.

Another parameters in the neural network are set like φ_m are equal to 1.0 and initial values of $\omega_m, \beta_m, \lambda_m$ are random numbers in $[-1, 1]$.

The branches from intermediate nodes to output nodes of the basic network are adopted as the branches under connection and disconnection control. Therefore, the outputs of the branch control network, in other words, the outputs of the fuzzy inference network decide whether the corresponding branches from intermediate nodes to output nodes of the basic network should be connected or disconnected.

3.2 Simulation Results

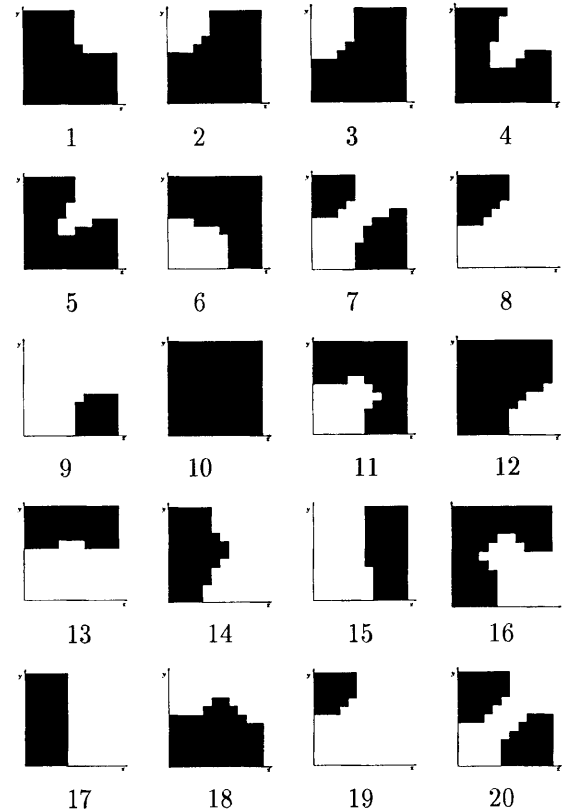


Fig.4 Connectivity from intermediate nodes to output node when $\rho = 0.5$

Fig.4 shows an example of the states of the connection and disconnection between the branches from twenty intermediate nodes to the output nodes respectively, when Z_{ij}^o s are adjusted for the connectivity ρ becomes 0.5. Black stands for the connec-

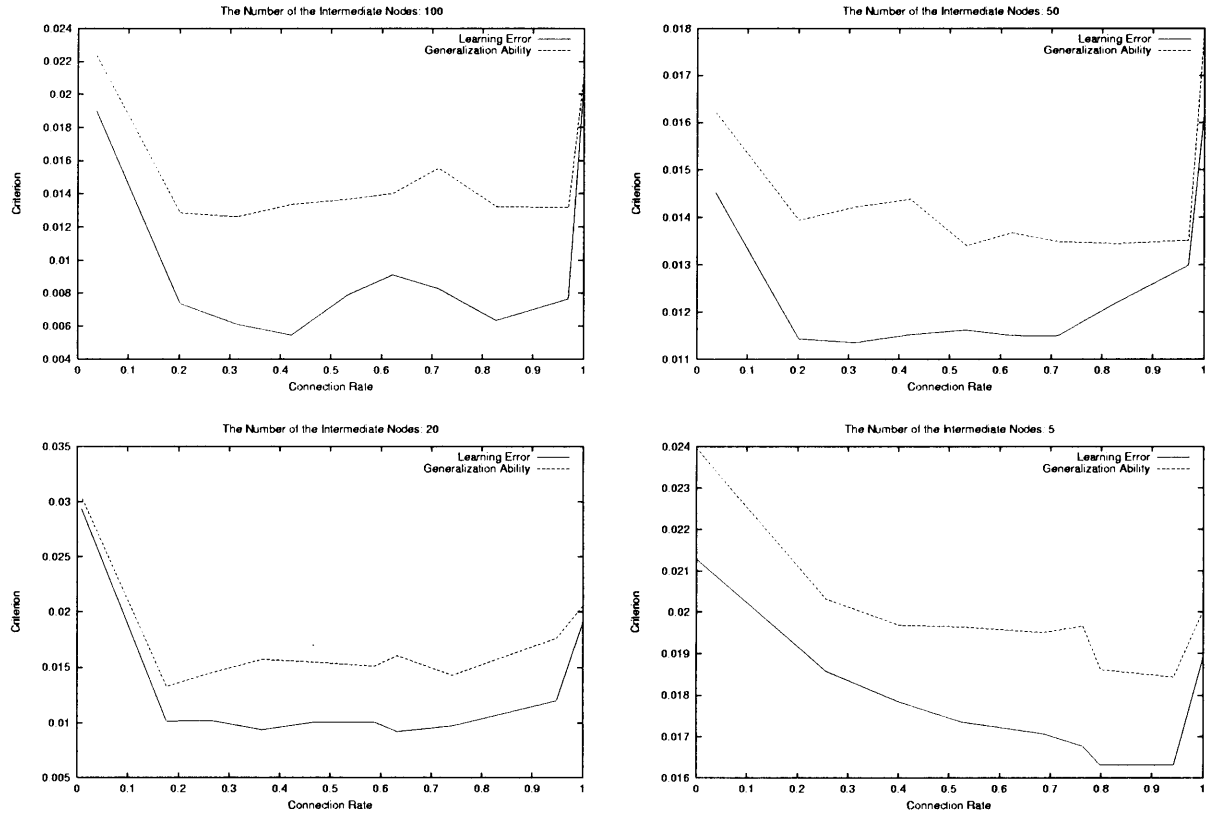


Fig.5 Simulation results of function 1

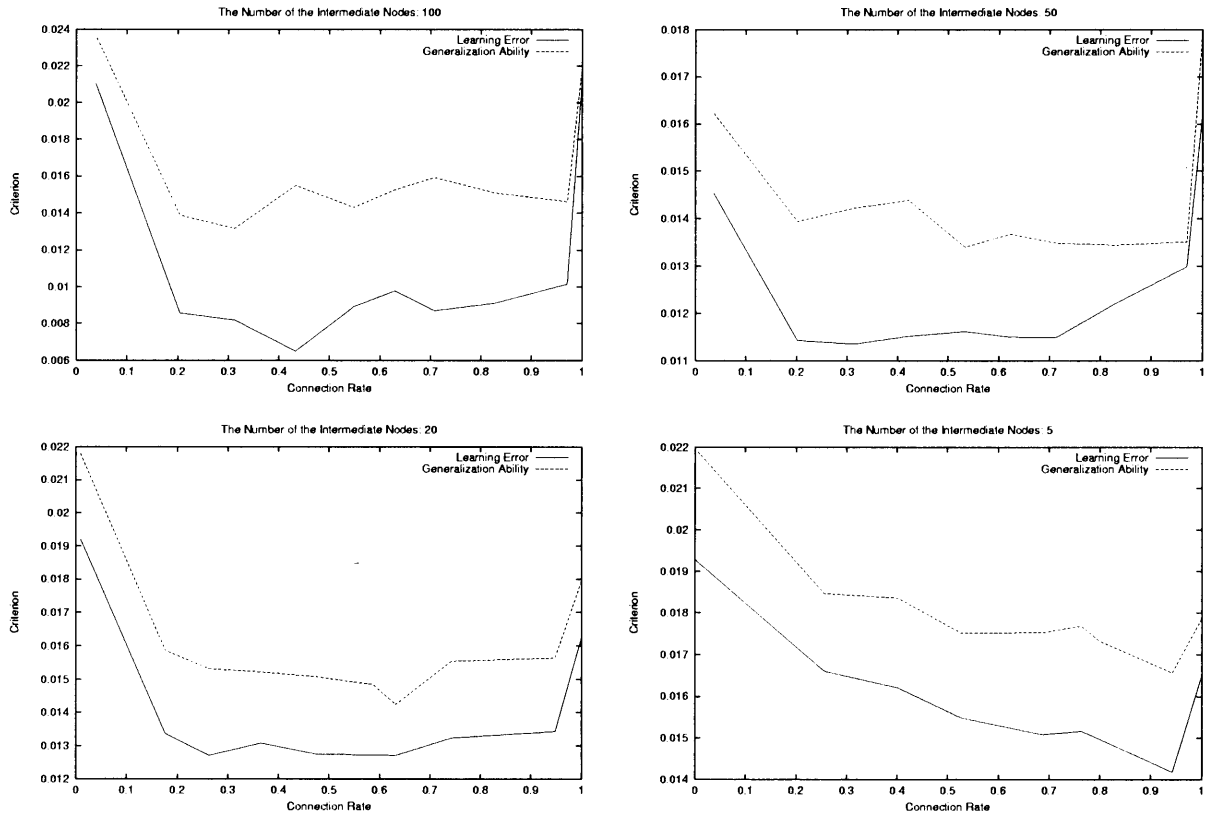


Fig.6 Simulation results of function 2

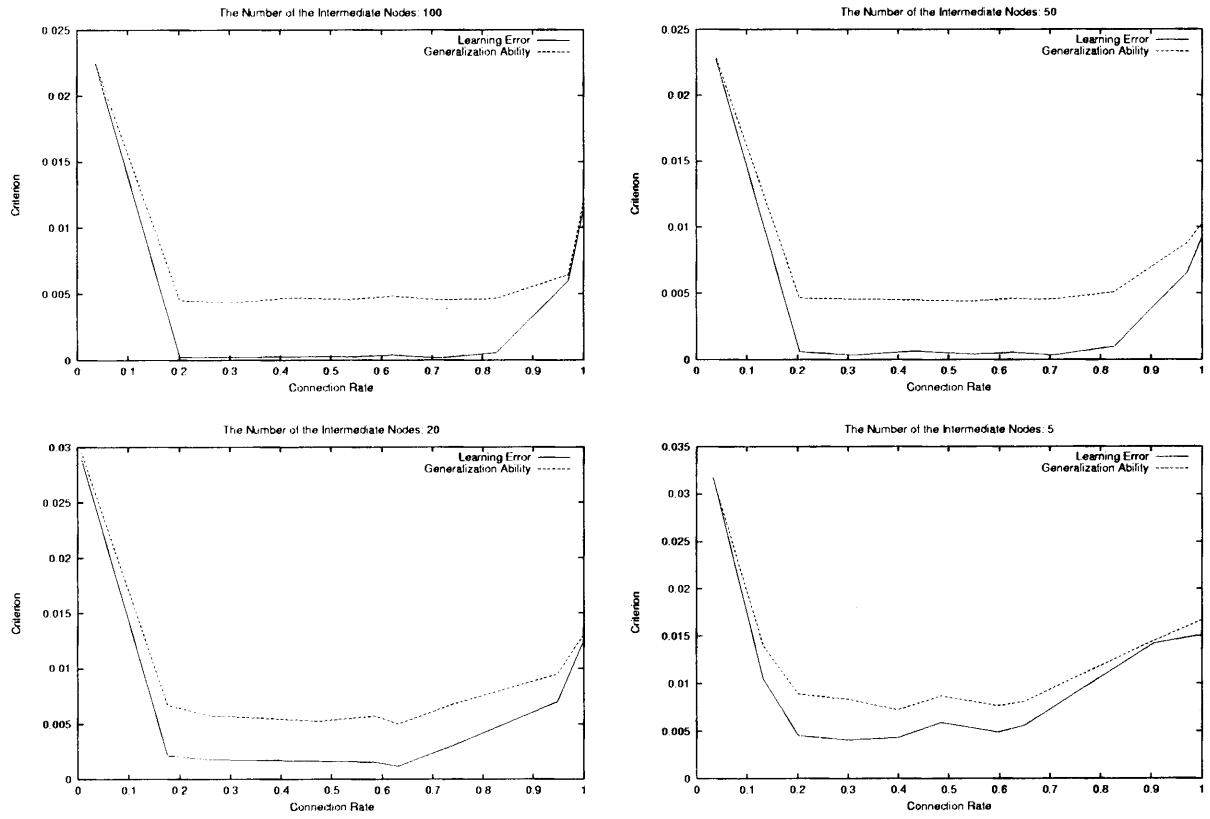


Fig.7 Simulation results of function 3

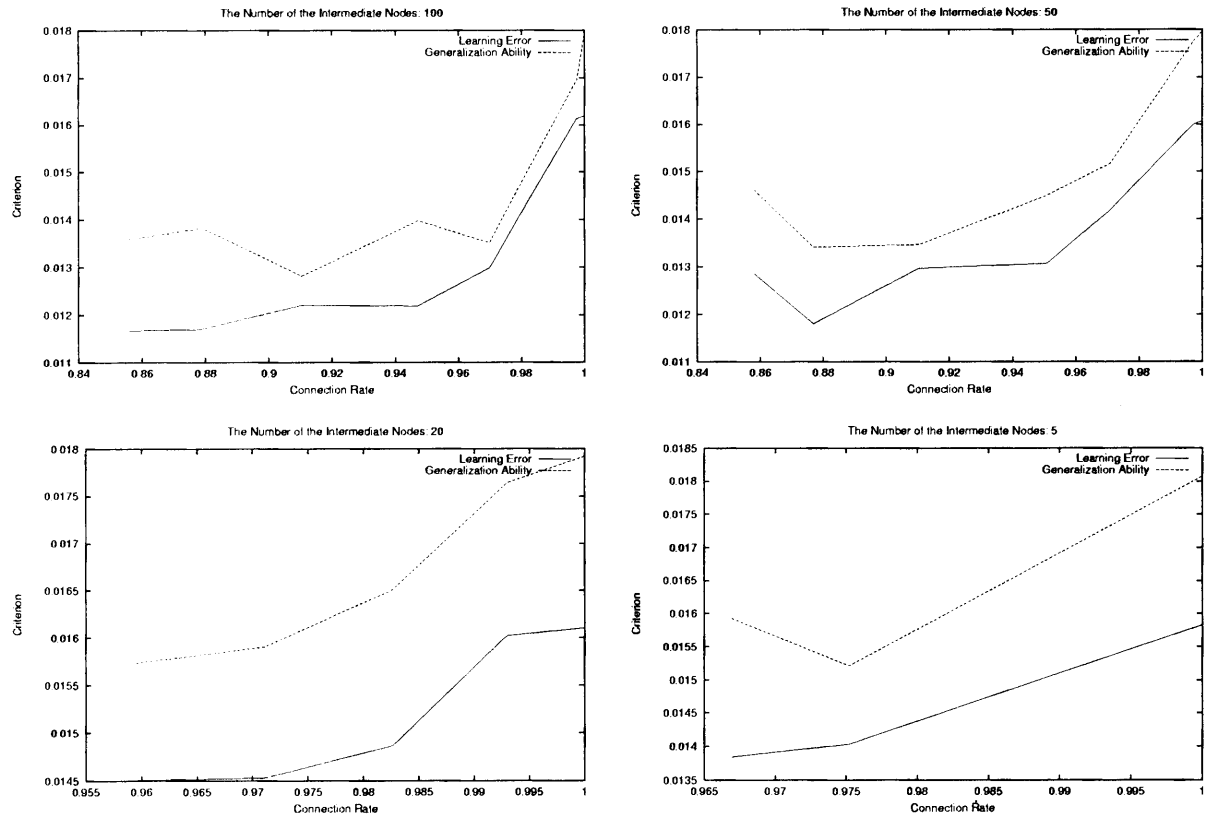


Fig.8 Detailed figures of function 1 simulations

tion and white shows the disconnection in the input space (x, y) .

It is shown in **Fig.4** that some areas in the input space (x, y) point out to the connections of the corresponding branches from the intermediate nodes to the output nodes, and other areas correspond to the disconnection, although the shape of areas is different branch by branch.

Fig.5-7 show how the representation and generalization ability of the ULN with BC change as the connectivity ρ varies by adjusting the threshold Z_{ij}^o . In this paper the generalization ability was calculated by using the neighboring points of the ones used for training.

From these figures it is clearly stated that all the figures have the tendency to have optimal connectivity, around which the best performance is obtained like **Fig.2**. The reason of piecewise linear curves in the figures is that it is hard to calculate the criterion functions at a great many points of connectivity by adjusting only the threshold Z_{ij}^o . And it can be found that when the number of the intermediate nodes is small five, the optimal connectivity tends to have almost one as shown in **Fig.5** and **6**. This is because high connectivity tends to improve the performances in the case of too small number of branches under branch control.

It is also understood from **Fig.5, 6** and **7** that around $\rho = 1.0$, the performances of the network, in other words, the representation ability and generalization ability of the network are degraded sharply. So, detailed calculations on function 1 were carried out on condition that the connectivity is around 1.0 as shown in **Fig.8**.

From **Fig.8**, it is shown that there exists a big performance difference between ρ is exactly 1.0 and ρ is around 1.0.

In the world of neural networks, functions locally distributed networks are not used widely, in other words, neural networks with the connectivity being 1.0 are mostly used. But, from the statements above, these commonly used neural networks do not always exhibit the best performance. This fact is not well known in the neural network community.

4. Conclusions

In the commonly used neural networks, especially layered neural networks, all input nodes and output nodes are generally connected to the intermediate nodes.

In this paper, a new type of functions locally distributed network is proposed, where the branch con-

trol network can control the branch connection and disconnection of the basic network. And from simulations, it has been clarified that there exists an optimal connectivity where the best performance of the network can be obtained.

The most interesting result is that the performance is fairly different between when all the branches are always connected and when a small number of branches are deleted depending on the input conditions of the network. This fact means that although all the branches of the networks are always connected in the commonly used neural networks, this structure is not always the best one. Theoretical research on these matters has not been done sufficiently until now. The enhancement of learning including training the parameters of the branch control networks as well as the study in the theoretical aspect of the proposed network are open problems.

References

- 1) T. Sawaguchi: Brain Structure and Evolution, Kaimeisha, 1989 (in Japanese)
- 2) K. Hirasawa, M. Ohbayashi, S. Sakai and J. Hu: Learning Petri Network and Its Application to Nonlinear System Control, *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, **Vol. 28**, No. 6, pp.781-789, 1998.
- 3) K. Hirasawa, S. Oka, M. Ohbayashi, S. Sakai and J. Murata: Locally Functions Distributed Network -Learning Petri Network- based on Petri Network, *Transactions of the Society of Instrument and Control Engineers of Japan*, **Vol. 32**, No. 6, pp.241-250, 1996.(in Japanese)
- 4) M. Ohbayashi, K. Hirasawa, S. Sakai and J. Hu: An Application of Learning Petri Network to Nonlinear System Control, *Transactions of the Institute of Electrical Engineers of Japan*, **Vol. 115-C**, No. 6, pp.875-881, 1998. (in Japanese)
- 5) K. Hirasawa, M. Ohbayashi, J. Murata, C. Jin, S. Oka, S. Sakai, Y. Shitamura: Modeling of Brain's Functions Distribution by Petri Network, *Transactions of the Institute of Electrical Engineers of Japan*, **Vol. 115-C**, No. 5, pp.719-727, 1995.
- 6) R.A.Jacobs and M.I.Jordan: Learning piecewise control strategies in a modular neural network architecture, *IEEE Transactions on Systems, Man, and Cybernetics*, **Vol. 23**, pp.337-345, 1993.
- 7) M.I.Jordan and R.A.Jacobs: Hierarchical mixtures of experts and the EM algorithm, *Neural Computation*, **Vol. 6**, pp.181-214, 1994.
- 8) K.Hirasawa, J.Hu, M.Ohbayashi and J.Murata : Computing Higher Order Derivatives in Universal Learning Networks, *Journal of Advanced Computational Intelligence*, **Vol.2**, No.2, pp.47-53, 1998.
- 9) J. Hu, K. Hirasawa and J. Murata : RasID - Random Search for Neural Network Training, *Journal of Advanced Computational Intelligence*, **Vol. 2**, No. 4, pp.134-141, 1998.