Learning of Parameter Variables and Searching Time Delays of Universal Learning Network Considering Extended Criterion Function

Han, Min
Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Hirasawa, Kotaro
Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Ohbayashi, Masanao
Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Obayashi, Masanao

他

https://doi.org/10.15017/1475362

出版情報：九州大学大学院システム情報科学紀要. 2 (1), pp.39-45, 1997-03-26. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

九州大学大学院
システム情報科学研究科報告
第2巻 第1号 平成9年3月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.2, No.1, March 1997

Learning of Parameter Variables and Searching Time Delays of Universal Learning Network Considering Extended Criterion Function

Min HAN *, Kotaro HIRASAWA *, Masanao OHBAYASHI * and Hirofumi FUJITA **

(Received December 24, 1996)

Abstract: Universal Learning Network(U.L.N.), which can model and control the large scale complicated systems naturally, consists of nonlinearly operated nodes and multi-branches that may have arbitrary time delays including zero or minus ones. Therefore, U.L.N. can be applied to many kinds of systems which are difficult to be expressed by ordinary first order difference equations with one sampling time delays. It has been already reported that the learning algorithm of parameter variables in U.L.N. by forward and backward propagation is useful for modeling, managing and controlling of the large scale complicated systems such as industrial plants, economic, social and life phenomena. But, in the previous learning algorithm of U.L.N., time delays between the nodes were fixed, in other words, the criterion function of U.L.N. was improved by adjusting only parameter variables. In this paper, a new learning algorithm is proposed, where not only parameter variables but also time delays between the nodes can be adjusted. Because time delays are integral numbers, adjustment of time delays can be carried out by a kind of random search procedure which executes intensified and diversified search in a single framework. By the way, an excessive dimensionality of the network implies lengthened processing and learning times. So, another assertion of this paper is that when adjusting parameter variables and time delays, an extended criterion function considering performance and compactness of the network is adopted.

Keywords: Universal learning network, Back propagation learning, Random search, Parameter variables, Time delays, Performance and compactness

1. Introduction

Neural networks have been widely studied in recent years. By learning algorithms, neural networks can simulate a certain complicated systems. But, general current neural networks are composed of fixed nodes and branches and can not be equipped with arbitrary time delays. So it is difficult to apply these kinds of neural networks to modeling of the large scale complicated systems. In order to solve this problem, Universal Learning Network(U.L.N.) has been proposed[1)2)].

The basic idea of U.L.N. is that most of the large scale complicated systems can be modeled by the networks which consist of nonlinearly operated nodes and multi-branches that may have arbitrary time delays including zero or minus ones. Therefore, U.L.N. can be applied to many kinds of systems which are difficult to be expressed as ordinary first order difference equations with one sampling time delays.

It has been already reported that the learning algorithm of parameter varibles in U.L.N. by forward and backward propagation is useful for modeling, managing and controlling of the large scale complicated systems such as industrial plants, economic, social and life phenomena. But, in the previous learning algorithm of U.L.N., time delays between the nodes were fixed, in other words, the criterion function of U.L.N. was improved by adjusting only parameter variables.

In this paper, a new learning algorithm is proposed, where not only parameter variables but also time delays between the nodes can be adjusted.

Because time delays are integral numbers, adjustment of time delays can be carried out by a kind of random search procedure which executes intensified and diversified search in a single framework. By the way, an excessive dimensionality of the network implies lengthened processing and learning times. So, another assertion of this paper is that when adjusting parameter variables and time delays, an extended criterion function considering performance and compactness of the network is adopted[3)].

Nowadays, adaptive time delay neural network

* Department of Electrical and Electronic Systems Engineering
** Department of Energy Conversion Engineering, Interdisciplinary Graduate School of Engineering Sciences

which can also adjust time delays by the gradient method has been proposed[5]. The difference between our method and adaptive time delay neural network is that the searching of optimal time delay in our method is based on a random search different from the gradient method. And our proposed method can be applied to not only feed forward networks but also recurrent networks, while adaptive time delay neural network can only be used for feed forward networks. It is also one of the features of our method that the extended criterion function which searches for the optimal structure of the network is used.

2. Basic Structure of Multi-branch Universal Learning Network

The structure of Universal Learning Network with multi-branches and filtering structures[3] is shown in **Fig.1**. In order to make the network compact, each of the branches has the filtering structures, namely, the switching functions such as $\alpha_{ij}(p)$ on pth branch from i node to j node. The learning parameter variables in the switching function should be learned so that $\alpha_{ij}(p)$ becmes 0.0 if the branch from i node to j node is unnecessary and $\alpha_{ij}(p)$ becomes 1.0 if the branch from i node to j node is necessary for the network to have predetermined performance.

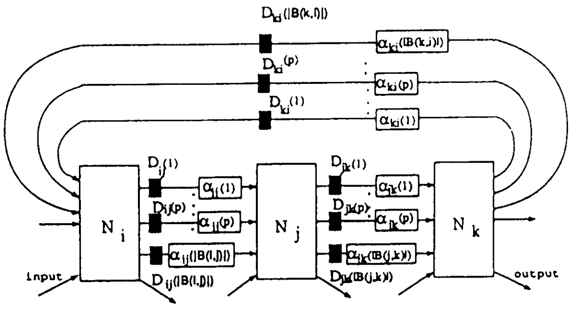

Fig.1 Structure of U.L.N. with multi-branches and filtering structures

Basic equation of U.L.N. is represented by Equation (1)

$$h_j(t) = O_j(\{h_i(t - D_{ij}(p)) | i \in JF(j), p \in B(i,j)\}, \{r_n(t) | n \in N(j)\}, \{\lambda_m(t) | m \in M(j)\}), j \in J, t \in T \quad (1)$$

where

$h_j(t)$: output value of j node at time t,

$\lambda_m(t)$: value of mth parameter variable at time t,

$r_n(t)$: value of nth external input variable at time t,

O_j : nonlinear function of j node,

$D_{ij}(p)$: time delay of pth branch from i node to j node,

$JF(j)$: set of numbers of nodes whose outputs are connected to j node,

$JB(j)$: set of numbers of nodes whose inputs are connected from j node,

$B(i,j)$: set of numbers of branches from i node to j node,

$N(j)$: set of numbers of external input variables that are connected to j node,

N : set of numbers of external input variables,

$M(j)$: set of numbers of parameter variables, with respect to these parameter variables output of j node can be partially differentiable,

M : set of numbers of parameter variables,

J : set of numbers of nodes,

T : set of sampling times.

As mentioned before there are the switching functions between the nodes, that are used for controlling the branch deletion. The switching function $\alpha_{ij}(p)$ is supposed to be

$$\alpha_{ij}(p) = \frac{1}{1 + e^{-\varphi \beta_{ij}(p)}} \quad (2)$$

Parameter variable $\beta_{ij}(p)$ is also adjusted in the same way as other parameter variables in order to minimize the extended criterion function including compactness evaluation.

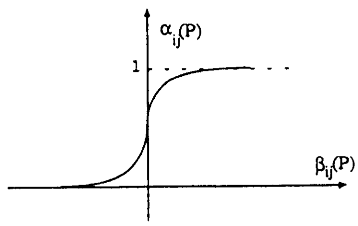

Fig.2 Switching function

It is also an important features of the proposed method that ψ in Equation (2) is set to a small value at the begining of learning, and is scheduled to increase gradually as the learning progresses. By executing the above procedure, upper bound $\alpha_{ij}(p) = 1.0$ or lower bound $\alpha_{ij}(p) = 0.0$ which mean connection and disconnection of the pth branch from i node to j node can be acquired at the end of learning.

The extended criterion function including both the usual criterion function E and the compactness of the network is given as follows:

$$L = E + R_\alpha \sum_i \sum_j \sum_p (\alpha_{ij}(p))^2 \quad (3)$$
$$E = E(\{h_r(s)\}, \{\lambda_m(s)\})$$

where

E : usual criterion function representing the general error,

R_α : weight coefficient,

$\alpha_{ij}(p)$: switching function of pth branch from i node to j node.

Depending on the value of weight coefficient R_α, the balance between the criterion function of E and the compactness of the network may be adjusted.

3. Learning of Parameter Variables λ_m and $\beta_{ij}(p)$

Learning of U.L.N.[1)] is to adjust λ_m and $\beta_{ij}(p)$ by back propagating $\frac{\partial L}{\partial h_j(t)}$, in the same way as that commonly used in neural networks. The different point of U.L.N. from commonly used neural networks is that U.L.N. can have arbitray time delays between the nodes and has multi-branches between the nodes. From the reference 1), λ_m and $\beta_{ij}(p)$ can be adjusted as follows.

$$\lambda_m \leftarrow \lambda_m - \gamma \frac{\partial^\dagger L}{\partial \lambda_m} \quad (4)$$

$$\beta_{ij}(p) \leftarrow \beta_{ij}(p) - \gamma \frac{\partial^\dagger L}{\partial \beta_{ij}(p)} \quad (5)$$

$$\frac{\partial^\dagger L}{\partial \lambda_m} = \sum_{t' \in T} \sum_{d \in JD(\lambda_m)} \left[\frac{\partial h_d(t')}{\partial \lambda_m} \delta(d, t')\right] + \frac{\partial L}{\partial \lambda_m} \quad (6)$$

$$\frac{\partial^\dagger L}{\partial \beta_{ij}(p)} = \sum_{t' \in T} \left[\frac{\partial h_j(t')}{\partial \beta_{ij}(p)} \delta(j, t')\right] + \frac{\partial L}{\partial \beta_{ij}(p)} \quad (7)$$

$$\delta(j,t) = \sum_{k \in JB(j)} \sum_{p \in B(j,k)} \left[\frac{\partial h_k(t + D_{jk}(p))}{\partial h_j(t)} \times \delta(k, t + D_{jk}(p))\right] + \frac{\partial L}{\partial h_j(t)} \quad (8)$$
$$j \in R, \quad t \in T$$

where

$JD(\lambda_m)$: set of numbers of nodes whose outputs can be partially differentiable with respeat to λ_m

$\frac{\partial^\dagger L}{\partial \lambda_m}$, $\frac{\partial^\dagger L}{\partial \beta_{ij}(p)}$, in Equation(4) and Equation(5) are the ordered derivative proposed by Werbos[4)].

In this paper, the nonlinear function of the node in U.L.N. was supposed to be a sigmoid function. Then, the followings are obtained.

$$h_j(t) = A \frac{1 - e^{-\alpha_j}}{1 + e^{-\alpha_j}} \quad (9)$$

$$\alpha_j = \sum_{i \in JF(j)} \sum_{p \in B(i,j)} h_{ij}(p) \alpha_{ij}(p) \times h_i(t - D_{ij}(p)) \quad (10)$$

$$\frac{\partial h_j(t)}{\partial h_{ij}(p)} = \frac{A}{2}\{1 - (\frac{h_j(t)}{A})^2\} \alpha_{ij}(p) \times h_i(t - D_{ij}(p)) \quad (11)$$

$$\frac{\partial h_j(t)}{\partial \beta_{ij}(p)} = \frac{A}{2}\{1 - (\frac{h_j(t)}{A})^2\} h_{ij}(p) \times h_i(t - D_{ij}(p)) \psi \times \alpha_{ij}(p)(1 - \alpha_{ij}(p)) \quad (12)$$

$$\frac{\partial L}{\partial \beta_{ij}(p)} = 2 R_\alpha \psi \alpha_{ij}(p)^2 (1 - \alpha_{ij}(p)) \quad (13)$$

$$\frac{\partial h_k(t + D_{jk}(p))}{\partial h_j(t)} = \frac{A}{2}\{1 - (\frac{h_k(t + D_{jk}(p))}{A})^2\} \times h_{jk}(p) \alpha_{jk}(p) \quad (14)$$

4. Random Search of Time Delays

In the previous learning algorithm of U.L.N., time delays between the nodes were fixed, that is, the criterion function of U.L.N. was improved by adjusting only parameter variables. In order to minimize the extended criterion function a new learning algorithm is proposed to adjust time delays while parameters are also learnt. The basic idea is to search for the optimal time delays by a kind of random search procedure, which has the intensification and diversification capability.

The feature of the proposed random search is to define lth neighborhood $N^l(x)$ of the current solution x satisfying Equation(15) and to search for the optimal time delays using $N^l(x)$ in such a way as when there is quite a possibility of finding good solutions around the current one, intensified search for the vicinity of the current solution is carried out(l is small), on the other hand, when there is no possibility of finding good solutions, diversified search is executed in order to find good solutions in the region far from the current solutions(l is large).

$$N^l(x) \subset N^{l+1}(x) \quad (15)$$

5. Simulation Results of a Nonlinear System Identification

As an example of learning, a nonlinear system identification problem was studied, where a system is supposed to be a cascade connection of the nonlinear subsystem $g_N(.)$ and the linear subsystem $g_L(.)$ shown in **Fig.3**.

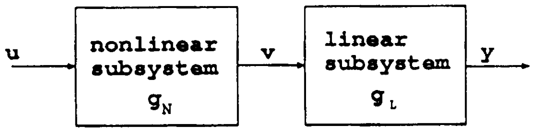

Fig.3 Cascade connection of nonlinear and linear subsystem

Generally speaking, nonlinar relations between input and output can be expressed by the following Equation(16) or (17).

$$y(k+1) = \begin{cases} f_1(z) & z \in \Phi_1 \\ \dots & \dots \\ f_m(z) & z \in \Phi_m \end{cases} \quad (16)$$

$$y(k+1) = \sum_{i=0}^{n_L-1} a_i y(k-i) + \sum_{i=0}^{n_N-1} b_i v(k-i) \quad (17)$$

where,

$$z = (y(k), .., y(k+1-n_L), u(k), ..., u(k+1-n_N))$$

$y(k)$: output value of linear subsystem at time k,
$u(k)$: input value of nonlinear subsystem at time k,
$v(k)$: input value of linear subsystem at time k,
n_L, n_N : dimension of denominator and numerator of linear subsystem,
Φ_1, ..., Φ_m : input space corresponding to function $f_1(z)$, .., $f_m(z)$.

For example, let the nonlinear subsystem be a relay element which can be expressed by Equation(18).

$$v(k) = \begin{cases} \nabla & u(k) \geq 0 \\ -\nabla & u(k) < 0 \end{cases} \quad (18)$$

And the linear subsystem be represented by the following second order differential equation.

$$\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y = \omega_n^2 v \quad (19)$$

Approximating the derivatives by the differences as

$$\dot{y} = \frac{y(k) - y(k-1)}{T_s}$$

$$\ddot{y} = \frac{\dot{y}(k+1) - \dot{y}(k)}{T_s} = \frac{y(k+1) - 2y(k) + y(k-1)}{T_s^2} \quad (20)$$

T_s : samlping time

Equation(19) can be transformed into the following discrete form

$$y(k+1) = a_0 y(k) + a_1 y(k-1) + b_0 v(k), \quad (21)$$

where,

$$\left.\begin{aligned} a_0 &= 2.0 - 2.0\zeta\omega_n T_s - \omega_n^2 T_s^2 \\ a_1 &= -1.0 + 2.0\zeta\omega_n T_s \\ b_0 &= \omega_n^2 T_s^2 \end{aligned}\right\}$$

From Equation(18) and Equation(21),$y(k+1)$ is expressed by,

$$y(k+1) = \begin{cases} a_0 y(k) + a_1 y(k-1) + b_0\nabla, & u(k) \geq 0 \\ a_0 y(k) + a_1 y(k-1) - b_0\nabla, & u(k) < 0 \end{cases} \quad (22)$$

The first and second expression in Equation(22) correspond to $f_1(z)$ and $f_2(z)$ in Equation(16) respectively, and in this case, $n_L = 2$, $n_N = 1$.

As a concrete simulation example, an extended nonlinear system of Equation (23) was considered and the system was modeled by the Universal Learning Network shown in **Fig.1**, which has 5 nodes and fully recurrent connections with 1 branch between the nodes and one external input and one output.

Input value $u(k)$ and output value $y(k)$ of the system to be identified in simulations are shown in Equation(23) and **Fig.4**.

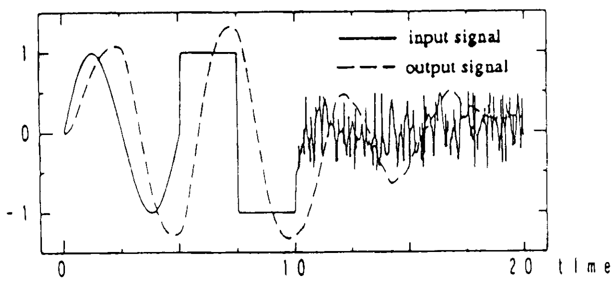

Fig.4 Input and output values of the system to be identified

$$y(k+1) = \begin{cases} 1.34y(k) - 0.277y(k-2) \\ -0.80y(k-4) + 0.01, & u(k) \geq 0 \\ 1.34y(k) - 0.277y(k-2) \\ -0.80y(k-4) - 0.01, & u(k) < 0 \end{cases} \quad (23)$$

Table1. Simulation conditions

number of nodes	J=5	criterion function	
number of branches between nodes	1	identification error E coefficient R_α	root square error 0.1, 0.5
nonlinear function	$f(x)= A\frac{1-e^{-x}}{1+e^{-x}}$ A=1.5	number of learning of parameters	500000
initial value of parameter λ_n $\beta_{ij}(p)$	random numbers in (-1.0,1.0) 0.3 (fully connected)	number of time delay search	50, 500, 5000
learning coefficient of λ_n $\beta_{ij}(p)$	γ=0.00002 γ=0.0002	φ in switching function	increase from 20 to 5000

Table 2. Identification results (weight coefficient R_α=0.1)

case	1	2	3	4	5
parameter learning number after search of delay	500000*	500000	10000	1000	100
search number	0	0	50	500	5000
residual branches	13	8	9	8	9
average error × $(10)^{-3}$	6.33	5.25	5.18	3.68	3.78

Table 3. Identification results (weight coefficient R_α=0.5)

case	6	7	8	9	10
paramete learning number after search of delay	500000*	500000	10000	1000	100
search number	0	0	50	500	5000
residual branches	7	7	7	7	5
average error × $(10)^{-3}$	10.4	9.0	7.85	7.65	5.5

*: time delays of all branches are assumed to be 1 sampling time.

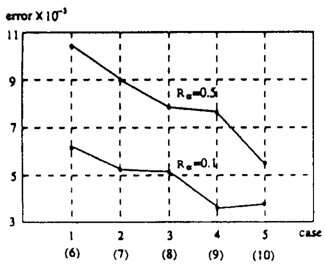

Fig.5 Average identification error of each simulation case

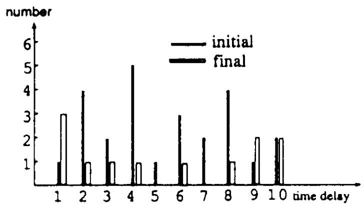

Fig.6 Distribution of time delays

Input value $u(k)$ to the nonlinear system was assumed to be

$$u(k) = \begin{cases} \sin(\frac{\pi}{50}k), & 0 \le k < 100 \\ 1.0, & 100 \le k < 150 \\ -1.0, & 150 \le k < 200 \\ \text{uniform random}, & 200 \le k < 400 \\ \text{numbers in}(-0.5, 0.5), & \end{cases} \tag{24}$$

Simulation conditions are shown in **Table 1**. Simulation cases are as follows:

[1]Only parameter variable learning without search of time delay (all time delays are one sampling time, (case1, 6)

[2]Only parameter variable learning without search of time delay(time delay was assumed to be random numbers in [1,10], case 2,7)

[3]Parameter variable learning and search of time delay combined (total paramete:r variable learning is 500000 times, case 3,4,5,8, 9,10)

Table 2 and **3**, **Fig.5** – **9** show average identification errors, learning and searching curves, identified and teaching signals, number of residual branches, distribution of time delays and curves of $\alpha_{ij}(p)$. In **Table 2**, **3** and **Fig.5**, average identification errors were calculated by averaging the 7 cases of 3 initial parameter variables × 3 initial time delays (the best and the worst errors were omitted). From the results, it is clarified that adjusting of not only parameter variables but also time delays is effective to improve the identification errors. And, it is also seen that the identification error and compactness of the network are appropriately determined corresponding to the coefficcient R_α. In **Fig.7**, learning and searching curves of minimizing the extended criterion function related to the identification error and compactness of the network, and identified and teaching signals are shown. **Fig.6** and **9** show distribution of time delays and time delays of residual branches.

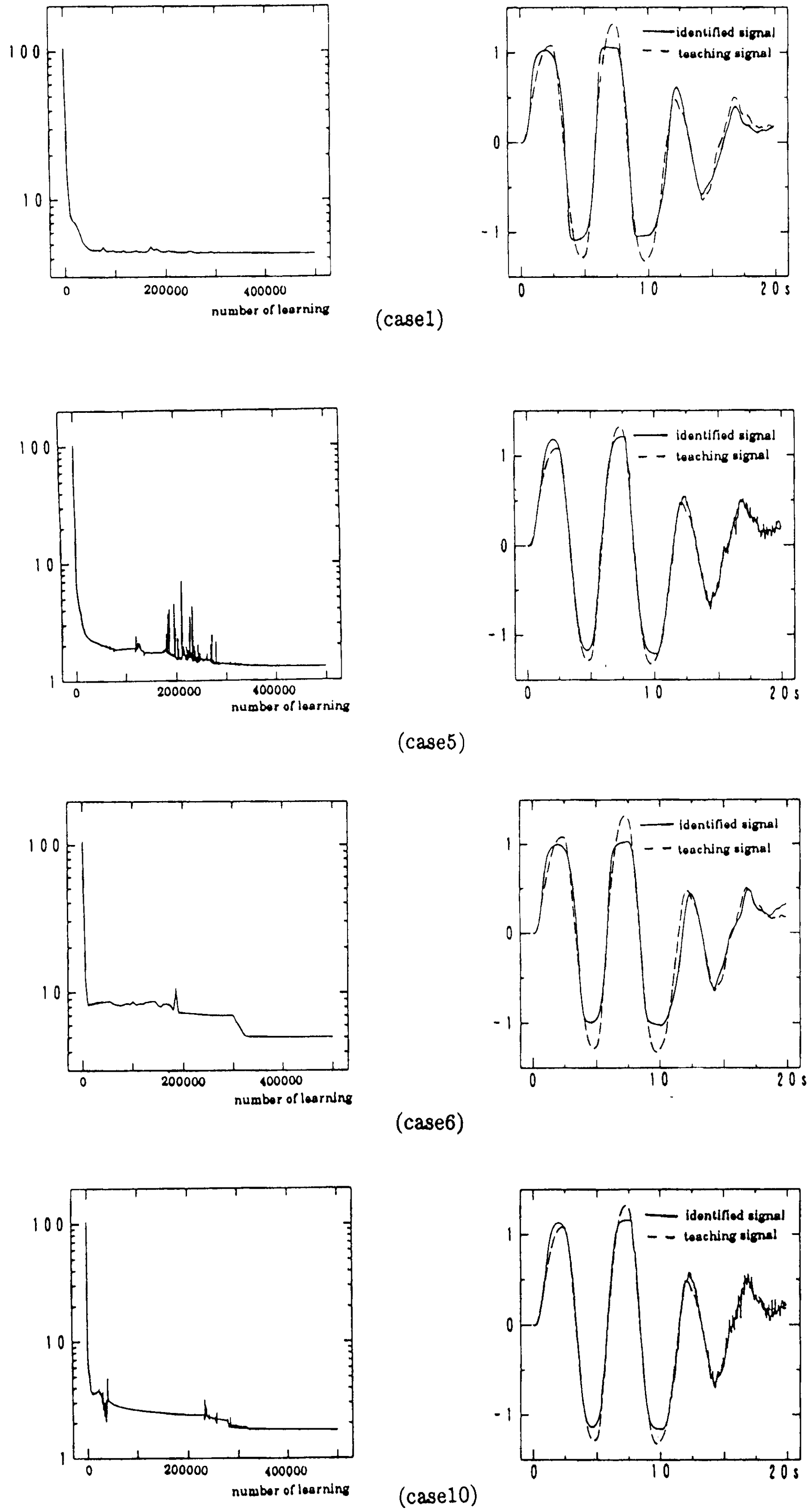

Fig.7 Learning and searching curves, identified and teaching signals

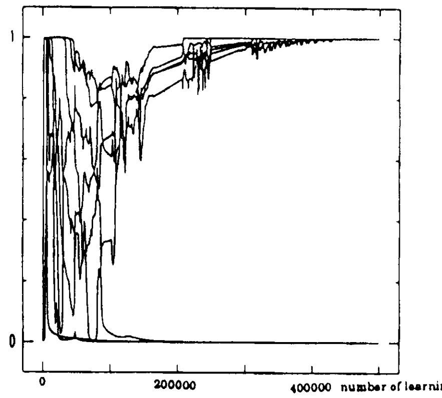

Fig.8 Curves of $\alpha_{ij}(p)$(case10)

i \ j	1	2	3	4	5
1	1	7	4	4	8
2	9	10	9	7	10
3	8	10	4	5	4
4	3	6	5	1	1
5	1	1	6	10	8

initial

i \ j	1	2	3	4	5
1	1		4	4	8
2					10
3	9				4
4		6	4		1
5			6		

number of time delay search 50

i \ j	1	2	3	4	5
1	1			5	10
2	3				6
3				4	5
4		8		10	1
5			9		

number of time delay search 500

i \ j	1	2	3	4	5
1	1			10	7
2					8
3					1
4				4	1
5	10	1			

number of time delay search 5000

Fig.9 Time delays of residual branches(case 10)

Learning curves of switching function $\alpha_{ij}(p)$ in **Fig.8** show that $\alpha_{ij}(p) = 1.0$ or 0.0 can be achieved after learning and searching because of the gradual increase of φ in switching functions. And, it has been also shown that the extended criterion function shown in Equation (3) is useful to determine the appropriate balance between the performance and the compactness of the network. From these results, it has been shown that adjusting of not only parameter variables but also time delays is useful for identification of a nonlinear dynamic system by using Universal Learning Network with the extented criterion function.

6. Conclusions

The Universal Learning Network is proposed for modeling and controlling the large scale complicated systems. One of the important features in U.L.N. is that U.L.N. can optimize the structure of large scale systems considering both the modeling error and the compactness of the network structure. In this paper a new learning algorithm which can adjust parameter variables as well as time delays at the same time is presented. The simulation results indicate that the proposed algorithm is effective. Especially, the identification error of a nonlinear system by the network which has less searching for time delays become worse compared with the identification error by the network whose time delays are sufficiently adjusted.

References

1)K. Hirasawa, M. Ohbayashi,J. Murata :Universal Learning Network and Computation of its Higher Order Derivatives, Proc. of 1995 IEEE International Conference on Neural Networks,1995

2)K. Hirasawa,M. Ohbayashi,M. Koga and M. Harada :Forward Propagation Universal Learning Network, Proc. of 1996 IEEE International Conference on Neural Networks, 1996

3)M.Han K.Hirasawa,M.Ohbayashi,H.Fujita : Modeling Dynamic Systems using Universal Learning Network, Proc. of 1996 IEEE International Conference on Systems, Man and Cybernetics, 1996

4)P.Werbos : Beyond regression: New Tools for Prediction and Analysis in the Behavior Science, Ph.D. dissertation, Harvard University, 1974

5) D.T.Lin, J.E. Dayhoff, P.A. Ligomenides: Trajectory Production with the Adaptive Time-Dalay Neural Network, Neural Networks, Vol.8, No3,pp447-461,1995