International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:9, 2011

Oscillation Effect of the Multi-stage Learning for the Layered Neural Networks and Its Analysis

Isao Taguchi, Yasuo Sugai

Abstract—This paper proposes an efficient learning method for the layered neural networks based on the selection of training data and input characteristics of an output layer unit. Comparing to recent neural networks; pulse neural networks, quantum neuro computation, etc, the multilayer network is widely used due to its simple structure. When learning objects are complicated, the problems, such as unsuccessful learning or a significant time required in learning, remain unsolved.

Focusing on the input data during the learning stage, we undertook an experiment to identify the data that makes large errors and interferes with the learning process. Our method devides the learning process into several stages. In general, input characteristics to an output layer unit show oscillation during learning process for complicated problems.

The multi-stage learning method proposes by the authors for the function approximation problems of classifying learning data in a phased manner, focusing on their learnabilities prior to learning in the multi layered neural network, and demonstrates validity of the multi-stage learning method.

Specifically, this paper verifies by computer experiments that both of learning accuracy and learning time are improved of the BP method as a learning rule of the multi-stage learning method.

In learning, oscillatory phenomena of a learning curve serve an important role in learning performance. The authors also discuss the occurrence mechanisms of oscillatory phenomena in learning. Furthermore, the authors discuss the reasons that errors of some data remain large value even after learning, observing behaviors during learning.

Keywords—data selection, function approximation problem, multi-stage leaning, neural network, voluntary oscillation.

I. Introduction

THIS paper proposes an efficient learning method for the layered neural networks based on the selection of training data and input characteristics of an output layer unit. Comparing to recent neural networks; pulse neural networks[1], quantum neuro computation[2], etc, the multilayer network is widely useddue to its simple structure. When learning objects are complicated, the problems, such as unsuccessful learning or a significant time required in learning, remain unsolved.

Focusing on the input data during the learning stage, we undertook an experiment to identify the data that makes large errors and interferes with the learning process. Our method devides the learning process into several stages. In general, input characteristics to an output layer unit show oscillation during learning process for complicated problems.

Isao Taguchi
Faculty of International Studies, Keiai University 1–5–21, Anagawa, Inage-ku, Chiba-shi, 263-8588
e-mail: taguchi@u-keiai.ac.jp
Yasuo Sugai
Graduate School of Engineering, Chiba University 1-33, Yayoi-cho, Inage-ku, Chiba-shi, 263-8522
e-mail: sugai@faculty.chiba-u.jp

We have suggested a multi-stage learning method with the following characteristics that are in contrast to the multi-layered neural network[3].

- It is clear that some learning data are difficult-to-learn and some are easy-to-learn.
- In the multi-stage learning method, difficult-to-learn data is given preference and simple data is added incrementally.
- Learning ratios of data are not constant but are set based on errors.
- Therefore, learning time is significantly reduced and accuracy is increased.
- It is said that the oscillation phenomena of the learning curve is effective for complex subjects, so some studies force outward oscillations [4]. In the multi-stage learning method, oscillations occur spontaneously during learning and they play an important role in the reduction of learning errors.

However, there are the following problems:

(1) The learning of complex functions has not been concretely evaluated in cases where a simple Back-Propagation Method (BP method) is used [5]$\sim$ [8] and the BP method is used in conjunction with the multi-stage learning method.
(2) The mechanisms of the oscillation phenomena[10]that have a major impact on learning accuracy have not been identified.
(3) It is not clearly understood why, upon completion of learning, some learned data has a larger percentage of errors, compared to other learned data.

Hence, in this paper, the followings are discussed:

(1') Learning time and accuracy for complicated functions were weighted by using the multi-stage learning method.
(2') Mechanisms of oscillation phenomena are analyzed by using an updated weight vector during learning.
(3') The reasons for learned data with a large percentage of errors at the time of completion of learning was discussed, relating it to oscillation phenomena.

The composition of this paper is as follows: II. will describe the introduction of indexes that evaluate the state of updated weight during learning, and III. will present the effectiveness of incorporating weight updating rules into the multi-stage learning method, through computer experiments [11]$\sim$ [13] where function approximation problems were used as examples. Then, IV. will discuss the mechanisms of the

oscillation phenomena during learning, and causes for the data with large percentage of errors after completion of learning [14],[15] and $V.$ gives the conclusions.

II. Introduction of indexes to evaluate the state of the updated weight during learning, and rules of updating weight.

When letting a desired signal vector against the pattern $p(=1,\cdots,P)$ be $\boldsymbol{d}^p = \{d_i^p\}(i=1,\cdots,n)$ and letting an output vector be $\boldsymbol{o}^p = \{o_i^p\}(i=1,\cdots,n)$, an output error vector $\boldsymbol{\epsilon}_{rr}^p$ is $\boldsymbol{\epsilon}_{rr}^p = \boldsymbol{d}^p - \boldsymbol{o}^p$. Also, let a weight coefficient matrix at the epoch t during learning be $\boldsymbol{W}(t) = \{W_{ji}(t)\}$. The $W_{ji}(t)$ is a weight coefficient from unit i to unit j.

Here, learning ratios[3] in according to errors will be described. The learning ratio η_p against the input pattern p will be set as follows:

$$\eta_p = \eta \frac{(|\boldsymbol{\epsilon}_{rr}^p|)^2}{\bar{E^2}} \qquad (p=1,\cdots,P), \tag{1}$$

$$\bar{E^2} = \frac{1}{P}\sum_{p=1}^{P}(|\boldsymbol{\epsilon}_{rr}^p|)^2. \tag{2}$$

In the formulas, $\eta\ (>0)$ means a standard learning coefficient, $|\boldsymbol{\epsilon}_{rr}^p|$ is the magnitude of an output error vector and $\bar{E^2}$ presents the average of square errors. In the multi-stage learning method, a batch-learning method where the amount of updated weight at an epoch is calculated by using a learning ratio according to each input pattern is employed.

In the t epoch, let the set of patterns where absolute errors more increase than the last epoch or absolute errors are equal to the last epoch be $\boldsymbol{P}^+(t)$,

$$\boldsymbol{P}^+(t) \equiv \{p : |\boldsymbol{\epsilon}_{rr}^p(t)| - |\boldsymbol{\epsilon}_{rr}^p(t-1)| \geq 0\}. \tag{3}$$

Similarly, let the set of patterns where absolute errors decrease be $\boldsymbol{P}^-(t)$,

$$\boldsymbol{P}^-(t) \equiv \{p : |\boldsymbol{\epsilon}_{rr}^p(t)| - |\boldsymbol{\epsilon}_{rr}^p(t-1)| < 0\}. \tag{4}$$

The number of all patterns is:

$$\forall t > 0 \ \ |\boldsymbol{P}^+(t)| + |\boldsymbol{P}^-(t)| = P. \tag{5}$$

In the formulas (1) through (4), the mark, $|\cdot|$, expresses the magnitude of error vectors and the mark $|\cdot|$ in the formula (5) indicates the number of elements of the set. In the computer experiments of this paper, updating weight coefficients are batched. Hence, updating weight at the t epoch is:

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \Delta\boldsymbol{W}(t). \tag{6}$$

Also, let the total amount of updated weight which belongs to $\boldsymbol{P}^+(t)$ be $\Delta\boldsymbol{W}^+(t)$ and the amount of updated weight which belongs to $\boldsymbol{P}^-(t)$ be $\Delta\boldsymbol{W}^-(t)$, the formulas are:

$$\Delta\boldsymbol{W}^+(t) \equiv \sum_{p\in\boldsymbol{P}^+} \Delta\boldsymbol{W}^p(t), \tag{7}$$

$$\Delta\boldsymbol{W}^-(t) \equiv \sum_{p\in\boldsymbol{P}^-} \Delta\boldsymbol{W}^p(t). \tag{8}$$

Here, $\Delta\boldsymbol{W}^+(t)$ and $\Delta\boldsymbol{W}^-(t)$ are called error increase vectors and error decrease vectors respectively (the number of elements for both vectors are n). The amount of updated weight coefficients of the formula (6) relates to

$$\Delta\boldsymbol{W}(\mathrm{t}) = \Delta\boldsymbol{W}^+(t) + \Delta\boldsymbol{W}^-(t). \tag{9}$$

III. Computer experiments by using function approximation problems as an example

In this section, computer experiments will be carried out by using function approximation problems as examples. The formulas (1) through (4) are expressed by using vectors as the general case where there are several units of the output layer; however, in function approximation problems, the number of output units is one, so a scalar is used, instead of vectors.

The Schwefel's function, Rastrigin function and Ridge function, all of which are frequently used in function optimization problems, are used as examples in this paper. Table I shows these functions.

The reason why the range of variables between function optimization problems and function approximation problems differs is that the range of learning is narrowed for learning in function approximation problems, as learning sometimes does not progress even if all of the methods are used. Also, in function optimization problems, the number of variables can be freely selected, but this paper uses two variables, x and y.

A. Learning data

For Table I(a), (b) and (c), each step size of x and y directions was set to 5.0, 0.1, and 1.2, areas were divided into three grids: 11×11 (121 grid points), 17×17 (289 grid points) and 11×11 (121 grid points). Values against these grid points are the set of the learning data.

Basically, it is difficult to learn data with larger absolute values, or data where the distance between the learning data is far, although the distance between input patterns are close [3]. To selectively learn this data, it is arranged in order of difficulty. This arranged data is equally divided into three groups. In the 1st step, the 1st group is learned. In the 2nd step, the 2nd group is added and all data is learned in the 3rd step.

Selection methods of learning data will be described below by using the Rastrigin function as an example, where, due to the difficulty of the subject, many experiments were carried out.

In function approximation problems, partial differential values can be used as a substitute for selection of data where the distance between learning data is far although the distance between input patterns are close. For selecting learning data of the Rastrigin function, partial differential values of x and y directions, $f_x(x,y)$ and $f_y(x,y)$, were used. In the 1st step, 24 pieces of learning data that satisfied $|f(x,y)| \geq 0.90$, 34 pieces of learning data that satisfied $|f(x,y)| \geq 0.90$ and 34 pieces of learning data that satisfied $|f_y(x,y)| \geq 60.3$ were included. Among the data, 2 pieces of data overlapped and 90 pieces of data were (31.1% of the all data) selected. In the 2nd step, 58.1% (168 pieces of data) of all the data was

TABLE I
LEARNING FUNCTIONS

(a)Schwefels Function
$f(x,y) = -x\sin\sqrt{\|x\|} - y\sin\sqrt{\|y\|}$
$(-30.0 \leq x \leq 30.0\Gamma - 30.0 \leq y \leq 30.0)$
$(-47.95 \leq f(x,y) \leq 47.95)$
(b) Rastrigin Function
$f(x,y) = x^2 - 10\cos(2\pi x) + y^2 - 10\cos(2\pi y)$
$(-0.8 \leq x \leq 0.8\Gamma - 0.8 \leq y \leq 0.8)$
$(-20.0 \leq f(x,y) \leq 20.5)$
(c)Ridge Function
$f(x,y) = 2x^2 + 2xy + y^2$
$(-6.0 \leq x \leq 6.0, -6.0 \leq y \leq 6.0)$
$(0.0 \leq f(x,y) \leq 180.0)$

selected. In the data, 52 pieces of learning data that satisfied $|f(x,y)| \geq 0.80$, 68 pieces of learning data that satisfied $|f_x(x,y)| \geq 60.1$ and 68 pieces of learning data that satisfied $|f_y(x,y)| \geq 60.1$ were included. Out of the data, 20 pieces of data overlapped and the total pieces of data was 168. In the 3rd step, all pieces of learning data were used.

The same methods were used for selecting learning data for the Schwefel's function and the Ridge function.

B. Comparison methods and conditions for experiments

To validate effectiveness of the multi-stage learning method, the BP method and a case where the BP method was applied as an updating weight of the multi-stage learning method, were compared in the computer experiment. In experiments for the multi-stage learning method, the same weight updating rules were employed from the 1st step through the 3rd step.

The neutral network composition was a feed-forward-type three-layered network which is comprised of two elements in the input layer, nine elements in the middle layer and one element in the output layer, by using a sigmoid function. The number of elements in the middle layer was determined based on the preliminary experiments. In addition, a batch updating method at every epoch, as described in section II, was used for updating weight coefficients.

For a root mean square error (" RMSE "), which evaluates the learning results, the average value of five initial values of weight coefficients set by using values against the learning data, was used. The initial value of a weight coefficient was randomly set in the range of [-0.01, 0.01]. The number of learning in the suggested method was 7000 epochs in total, comprising of 2333 epochs for the 1st stage, 2333 epochs for the 2nd stage and 2334 epochs for the 3rd stage. In the BP method, all pieces of learning data were always used in all 7000 epochs. The specification of the computer used for the experiment was as follows: OS: Windows XP, CPU: Pentium 4, 3.0GHz, RAM: 2GB.

C. Computer experiment results and discussion

Table II and III show the accuracy and learning time in cases where the BP method was incorporated into the multi-stage learning method and where the BP method was used alone against the Ridgefunction, Rastrigin function, and Schwefels function. The mark, "+", of the multi-stage learning method in the Tables indicates the BP method incorporated into the multi-stage learning method, and the values are from the results of five experiments where the initial values of weight were randomly set. The accuracy of learning was evaluated by the RMSE.

TABLE II
RMSE FOR MULTI-STAGE LEARNING AND TRADITIONAL METHODS IN LEARNING OF THREE FUNCTIONS.

Function	Method	
	+BP	BP
Ridge mean	0.107	0.241
Max	0.235	0.243
Min	0.038	0.234
Rastrigin mean	0.071	0.259
Max	0.095	0.483
Min	0.039	0.193
Schwefels mean	0.126	0.295
Max	0.134	0.343
Min	0.127	0.244

TABLE III
LEARNING TIME FOR MULTI-STAGE LEARNING AND TRADITIONAL METHODS IN LEARNING OF THREE FUNCTIONS.

Function	Method	
	+BP	BP
Ridge	419	620
Rastrigin	1204	1956
Schwefels	574	966

First, the accuracy of learning will be described. According to Table II, for the Ridge function and Rastrigin function, errors are fewer in the cases where the BP method was incorporated into the multi-stage learning method. These results validate that incorporating the BP into the multi-stage learning method is effective.

From Table II, errors of the Schwefel's function decreases when the +BP method is used.

Next, the learning time will be examined. From Table III, the learning time when the BP method was incorporated into the multi-stage leaning method was 59% to 68% compared to that of when the BP method was not incorporated into the multi-stage learning method. These values correspond to the values in the cases of $s = 3$ against the estimated formula $\frac{s+1}{2s}$ in the s step of the reference [3].

Fig. 1 and 2 show the typical examples of characteristics of the input into the elements in the output layer in the 3rd step during learning. These Figures present the input into the elements in the output layer, that is, the summation of the input from elements in the middle layer through weight, when one piece of learning data was selected and fixed at the start of learning, and the learning data was input at each epoch. The horizontal axis indicates the number of epochs and the vertical axis is the value input into the elements in the output layer. On the number of epochs of the horizontal axis, the final epoch of the 2nd step is set to 0.

The RMSE values at the time of completion of learning against the characteristics shown in Fig.1 and Fig. 2 were 0.206 and 0.039 respectively.

Fig. 1 expresses the characteristics of the BP method against the Rastrigin function. Although the method used is not the multi-stage learning method, the epoch that is equal to

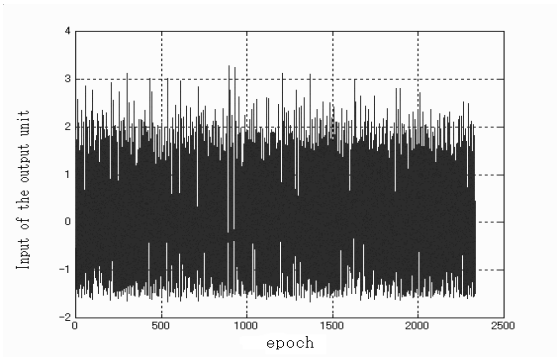

Fig. 1 Input characteristics for the BP methods in the output unit (Rastrigin function, $\eta = 0.8$).

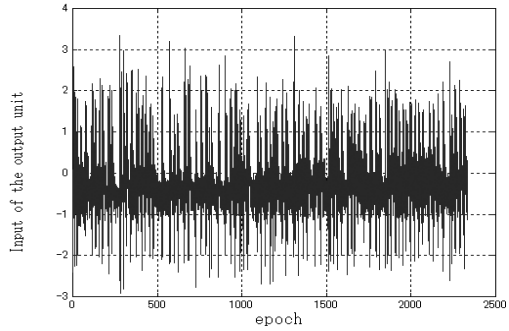

Fig. 2 Input characteristics for the +BP methods in the output unit (Rastrigin function, $\eta = 0.8$).

the final epoch in the 2nd stage is set to zero for the number of epochs on the horizontal axis. The characteristics of Fig.1 are that irregular large vibrations continue until the last. The RMSE value after the learning was 0.206, which is relatively large, and this is an example of when learning fails.

Fig. 2 shows the characteristics of +BP against the Rastrigin function. The oscillation phenomena where irregular vibration is added to the regular oscillation continue.

D. BP method and oscillation effectiveness

When the learning ratio is decreased in the BP method, the learning curve vibrates similar to that shown in Fig. 1, but its amplitude becomes smaller. Furthermore, when the learning ratio is greatly decreased, the vibration stops; however, errors are large regardless of whether there is vibration or not, and the Ridge function and Rastrigin function cannot be successfully learned.

Also, in the multi-stage learning method, the learning ratio η (see the formula (1)) was reduced, the oscillation stopped, as in the BP method, but successful learning did not take place. For example, when the +BP method was applied with the learning ratio of $\eta = 0.05$ to the Rastrigin function learning, oscillation did not occur during learning, but the RMSE value was 0.135, which means a large error. In contrast with this result, in learning where the learning ratio of $\eta = 0.80$ was employed, the oscillation phenomena occurred and the RMSE value is 0.071 according to Table II. The computer experiments demonstrated that the RMSE values were 0.1 or smaller when the learning ratio was in the range of $\eta = 0.2 \sim 0.9$.

From the above statement, as a learning subject becomes more complicated, the vibration phenomena play an increasingly important role in learning. Therefore, inertial items which can suppress vibration are not incorporated into the multi-stage learning method [3]. The vibration during learning occurs spontaneously, without external stimulus. The cause of the spontaneous occurrence of the vibration phenomena will be discussed in IV..

IV. Mechanisms of Oscillation Occurrence and Learning Behaviors of Data with Large Errors Even After Completion of Learning

A. Oscillation state and non-oscillation state

As presented in Fig.1 and Fig. 2, there are various types of oscillation phenomena occurring in the characteristics of the input into elements in the output layer. Many experiments have demonstrated cases where the oscillation state changed to the non-oscillation state and vice versa. Fig. 3 shows the outline of typical changes of the magnitude and directions of $\Delta \boldsymbol{W}^+(t)$ and $\Delta \boldsymbol{W}^-(t)$ when the oscillation state changes to the non-oscillation state and vice versa. The direction of the resultant vector $\Delta \boldsymbol{W}(t)$ is drawn with reference to its facing right. For the vibration state, $\frac{|\Delta \boldsymbol{W}^+(t)|}{|\Delta \boldsymbol{W}^-(t)|} < 1$ and $\frac{|\Delta \boldsymbol{W}^+(t)|}{|\Delta \boldsymbol{W}^-(t)|} > 1$ are repeated. When the learning state is stable under a non-oscillation state, the entire errors monotonically decrease, resulting in $\frac{|\Delta \boldsymbol{W}^+(t)|}{|\Delta \boldsymbol{W}^-(t)|} < 1$. Error reduction vectors effectively work all the time. Under the non-oscillation state, the number of error increase patterns $|\boldsymbol{P}^+(t)|$ and the number of error reduction patterns $|\boldsymbol{P}^-(t)|$ change little, even though the epoch progresses and the relationship of $\frac{|\boldsymbol{P}^+(t)|}{|\boldsymbol{P}^-(t)|} \leq 1$ is maintained.

In Fig. 3, $a1$ indicates the case of $|\Delta \boldsymbol{W}^+(t)| > |\Delta \boldsymbol{W}^-(t)|$, $a2$, conversely, means the case of $|\Delta \boldsymbol{W}^-(t)| > |\Delta \boldsymbol{W}^+(t)|$, and a3 and a4 show that $|\Delta \boldsymbol{W}(t)|$ is smaller against a1 and a2. In $a5$, the oscillation state is $|\Delta \boldsymbol{W}^+(t)| \approx 0$, and $|\Delta \boldsymbol{W}^-(t)|$ and $|\Delta \boldsymbol{W}(t)|$ correspond to each other. When the learning curve does not oscillate, the decrease vector is dominant, and the magnitude of the vector is smaller. This means that this state is maintained in the resultant vector $|\Delta \boldsymbol{W}(t)|$. In the non-oscillation state, the relationship of $\frac{|\Delta \boldsymbol{W}^+(t)|}{|\Delta \boldsymbol{W}^-(t)|} \ll 1$ is maintained and all errors monotonously decrease.

B. Occurrence of oscillation

In Fig. 3, b1∼b7 show drawings of vectors changing from the non-oscillation state to the oscillation state. Even if $\Delta \boldsymbol{W}$ is updated to the direction of error reduction in all patters under the non-oscillation state, it is hard to imagine that the state continues until the end of learning. Hence, we assume that a pattern $p \in \boldsymbol{P}^+(t)$ occurs at an epoch t. This pattern belongs to another pattern $\boldsymbol{P}^-(t)$ and $|\Delta \boldsymbol{W}^-(t)|$ is dominant over

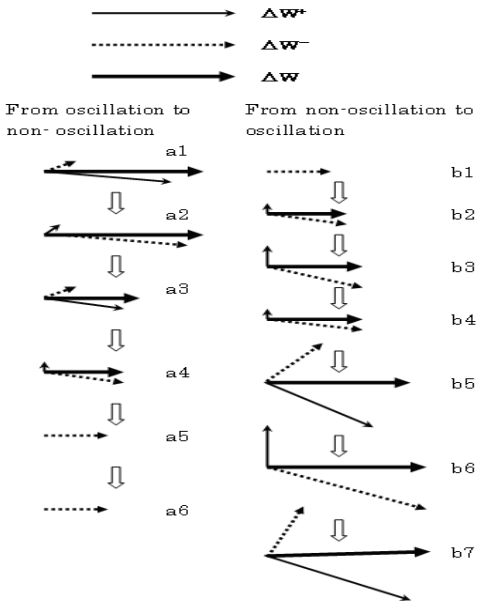

Fig. 3 The weight renewal vector for the multi-stage learning methods.

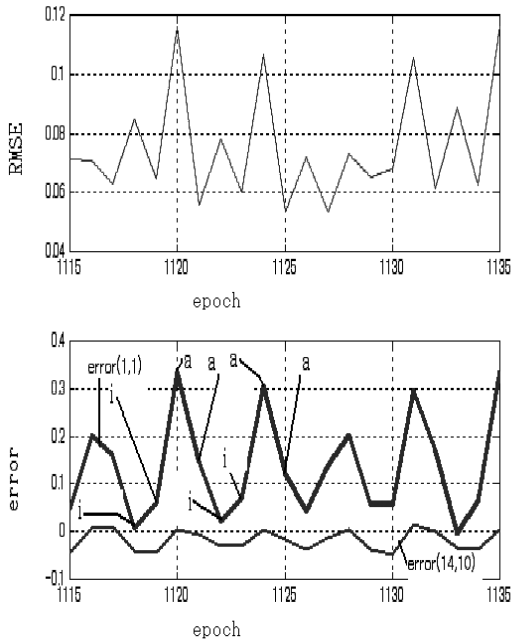

Fig. 4 The behavior of the paturn p^* between 1115∼1135 epochs at the third stage in the multi-stage learning($\eta = 0.4$).

the amount of updated weight $|\Delta \boldsymbol{W}(t)|$. Therefore, weight is corrected to the direction where errors against the pattern p increase. So, the following states

$$|\Delta \boldsymbol{W}^+(t+1)| > |\Delta \boldsymbol{W}^+(t)|$$

continue for several epochs. As clarified in the formula (1), the learning ratios are effective when errors are raised to the second power and the increase speed of $|\Delta \boldsymbol{W}^+(t)|$ is accelerated. This will make the pattern p which was $p \in \boldsymbol{P}^+(t)$ at the epoch t be $p \in \boldsymbol{P}^-(t')$ at an epoch $t'(> t)$. However, as the error of the pattern p becomes smaller at the epoch $(t'+1)$, the pattern p becomes $p \in \boldsymbol{P}^+(t'+1)$ again. These processes are repeated, causing the vibration phenomena.

C. Learning behaviors of learning with large errors, even after completion of learning

There are some patterns with large errors even after completion of learning. These patterns are not changed in accordance with learning methods or each learning, but some specified patterns apply to such patterns. For instance, $(x, y) = (1, 1), (1, 15), (1, 17)$ apply to the patterns as to the Rastrigin function learning. In this section, we will discuss why the errors of the pattern p^* do not become smaller even after learning by using $p^* \equiv (1, 1)$ as an example.

For $\boldsymbol{P}^{\pm}(t)$ in the formulas (3) and (4), when $|\boldsymbol{P}^{\pm}(t)| > |\boldsymbol{P}^{\mp}(t)|$ (double-sign corresponds), let us call the former the majority and the latter the minority. Generally, when $|\boldsymbol{P}^{\pm}(t)| > |\boldsymbol{P}^{\mp}(t)|$, most cases fall under $|\Delta \boldsymbol{W}^-(t)| > |\Delta \boldsymbol{W}^+(t)|$ (double-sign corresponds) as to the weigh correction. The RMSE values decrease when $|\Delta \boldsymbol{W}^-(t)| > |\Delta \boldsymbol{W}^+(t)|$.

The pattern p^* is data where the absolute value of the learning data is large and which is used for learning from the 1st stage in the multi-stage learning method. For example, in an experiment where the +BP method was used for the Rastrigin function learning, the initial value of the absolute value of errors of the pattern p^* was $\epsilon_{rr}^{p^*}(0) = 0.164$, the learning curve vibrated in the range of 0.013∼0.014 when the 1st stage was completed, the curve vibrated in the range of 0.020∼0.028 when the 2nd stage was completed and then the curve vibrated in the range of 0.094∼0.125 when the 3rd stage was completed. Accordingly, in the 1st stage, only data which was determined to be difficult was learned and errors became smaller, to some extent. When new learning data was added in the 2nd and 3rd stages, initial errors of the data were large and weight correction vectors were forcibly moved in the direction which reduces errors of newly added data. Hence, the pattern p^* belonged to the minority and errors that had been reduced, increased again. On the other hand, the initial value of an absolute data of errors of data added in the 3rd stage was $\epsilon_{rr}^{p^\dagger}(4667) = 0.029$, as to the pattern $p^\dagger \equiv$(14,10), the learning curve vibrated in the range of 0.011∼0.047 with learning of 5 epochs and errors of this pattern were smaller compared to those of the pattern p^*. At this point, a relative relationship between pattern p^* and pattern $p^\dagger$ was fixed as to the magnitude of errors. Until the completion of learning, the relationship continued.

Fig. 4 shows the RMSE values at 1115 to 1135 epochs and

behaviors of the pattern p^* and $p^\dagger$ in the 3rd stage. The chart above expresses the RMSE value and the one below represents the errors. In the Fig. 4 the bold line corresponds to the pattern p^*, and the narrow line to the pattern $p^\dagger$. The letter "a" is the majority and "i" is the minority. As you can see from Fig. 4, the pattern p^* repeats in the following order: the minority, the minority, the majority and the majority. When the pattern p^* belongs to the majority, errors decrease and errors increase when the pattern p^* belongs to the minority. The same applies to $p^\dagger$. Thus, the state where the average of errors decreases only slightly continues until the completion of learning while the relative relationship as to errors is maintained. As a result, learning data, has a pattern similar to the pattern p^*, whose errors increase even after reducing once in the 1st stage, and are large even after learning is completed .

V. Conclusion

This paper extends the multi-stage learning method proposed by the authors for the function approximation problems of classifying learning data in a phased manner, focusing on their learnabilities prior to learning in the multi layered neural network, and demonstrates validity of the extended multi-stage learning method. Specifically, this paper verifies by computer experiments that both of learning accuracy and learning time were improved even when the BP method is used as a learning rule of the multi-stage learning method.

In learning, oscillatory phenomena of a learning curve serve an important role in learning performance. The authors also discuss the occurrence mechanisms of oscillatory phenomena in learning. Furthermore, the authors discuss the reasons that errors of some data remain large value even after learning, observing behaviors during learning.

The multi-stage learning method focuses on the distance between output vectors, to the distance between input vectors and the magnitude of output vectors, (output values of the function approximation problems in this paper) for categorizing learning data. A further expansion of this study on the multi-stage learning method, will be to apply it to learning subjects such as discrimination problems where such distance relationships are not available.

References

[1] Makoto Motoki, Seiichi Koakutsu, Hironori Hirata:"A Supervised Learning Rule Adjusting Input-Output Pulse Timing for Pulsed Neural Network", The transactions of the Institute of Electronics, Information and Communication Engineers, Vol.J89-D-II, No.12, pp.2744-2756 (2006)
[2] Noriaki Kouda, Nobuyuki Matsui, Haruhiko Nishimura:"A Multi-Layered Feed-Forward Network Based on Qubit Neuron Model", The transactions of the Institute of Electronics, Information and Communication Engineers, Vol.J85-D-II, No.4, pp.641-648 (2002)
[3] Isao Taguchi and Yasuo Sugai :"An Efficient Learning Method for the Layered Neural Networks Based on the Selection of Training Data and Input Characteristics of an Output Layer Unit", The Trans. of The Institute of Electrical engineers of Japan, Vol.129-C, No.4, pp.1208-1213, (2009)
[4] Nobuyuki Matsui and kenichi Ishimi:"A Multilayered Neural Network Including Neurons with fluctuated Threshold", The Trans. of The Institute of Electrical Engineers of Japan, Vol.114-C, No.11, pp.1208-1213 (1994)
[5] D.E.Falleman:"An Empirical Study of Learning Speed in Back-Propagation Network", Technical Report CMU-CS-88-162, Carnegie-Mellon University, Computer Sceinece Dept., (1988)
[6] M. Riedmiller and H. Braun:"A DirectbAdaptive Method for Faster Backpropagation Learning: The RPROP Algorithm", Proc. ICNN, San Fransisco, (1993)
[7] Isao Taguchi and Yasuo Sugai :"An Input Characteristic of Output Layer Units in the Layered Neural Networks and Its Application to an Efficient Learning", Proc. of the Electronics, Information and Systems Conference, Electronics, Information and systems Society, IEE. of Japan, pp.931-934 (2004)
[8] Isao Taguchi and Yasuo Sugai :"An Efficient Learning Method for the Layered Neural Networks Based on the Selection of Training Data and Input Characteristics of an Output Layer Unit", The Trans. of The Institute of Electrical engineers of Japan, Vol.129-C, No.4, pp.1208-1213, (2009)
[9] D.E.Rumelhart, J.L.McCleland, and the PDP Research Group: "Parallel Distributed Processing Vo.1", MIT Press (1986).
[10] Takashi Kanemaru and Masatoshi Sekine:"Oscillations and Synchronizations in Class 1 Neural Networks", TECHNICAL REPORT OF THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS, NC2003-138, pp.17-22 (2004)
[11] Ting Wang and Yasuo Sugai:"A Wavelet Neural Network for the Approximation of Nonlinear multivariable Functions", The Trans. of The Institute of Electrical engineers of Japan, Vol.120-C, No.2, pp.185-193 (2000)
[12] Ting Wang and Yasuo Sugai:"A Wavelet Neural Network for the Approximation of Nonlinear Multivariable Functions", Proc.of IEEE International Conference on System, Man, and Cybernetics, III, pp.378-383 (1999)
[13] K. Funahashi:"On the Approximate Realization of Continuous Mapping by Neural Networks", Vol.2, No.3, pp.183-192 (1989)
[14] Yasuo Sugai, Hiroshi Horibe, and Tarou Kawase:"Forecast of Daily Maximum Electric Load by Neural Networks Using the Standard Electric Load", The Trans. of The Institute of Electrical engineers of Japan, Vol.117-B, No.6, pp.872-879 (1997)
[15] Souiti Umehara, teru Yamazaki, and Yasuo Sugai:"A Precipitation Estimation System Based on Support Vector Machine and Neural Network", The transactions of the Institute of Electronics, Information and Communication Engineers, Vol.J86-D-II, No.7, pp.1090-1098 (2003)
[16] Charles K.Chui:"An Introduction to Wavelets", Academic Press (1992)
[17] L. K .Jones:"Constructive Approximations for Neural Networks by Sigmoidal Functions", Proc. IEEE, Vol.78, No.10, pp.(1990)
[18] B.Irie and S.Miyake:"Capabilities of Three Layered Perceptrons", Proc.ICNN, Vol.1, pp.641-648 (1988)

Isao Taguchi Isao Taguchi received the D.Eng. course of urban environmental systems, graduate school of engineering, Chiba university, in 2011. I am a professor, course of Faculty of international studies, Keiai university , in 2005. My research area are the neural networks, information science, information education, and their applications. I am a member of IEEJ.

Yasuo Sugai Yasuo Sugai received the D.Eng. degree from the Tokyo Institute of Technology in 1985. He is a professor, course of urban environmental systems, graduate school of engineering, Chiba university. His research area are the neural networks, the emergent computation, the optimization engineering, and their applications. He is a member of IEICE, IPSJ and SICE.