Research on a self-built learning control mechanism in backpropagation training : convergence improvement

バックプロパゲーショントレーニングにおける 自己設定学習制御機構の研究：収束性の改良

Muhamad SADLY* and Yoshizumi YASUDA**

ムハマドサドリイ・安田 嘉純

Abstract : The main topic of this paper is the use of the self-built learning control mechanism in backpropagation training, under the conditions encountered in processing remote sensing data. The approach based on the self-built learning control mechanism of controlling the learning rates and tuning of momentum term at the same time automatically in training of the feedforward neural network. An important feature of our novel method is that the initial learning parameters are not crucial to the success of the training, because the learning parameters are controlled that does not cause instability. Furthermore, the automatic initials learning rate selection and the optimal values of the learning control parameters were experimentally discussed. The proposed method was verified by a land cover study over Cianjur area, West-Java, Indonesia, with MOS-1 multi-spectral imagery. The simulations were presented to indicate the remarkable advantages of the proposed approach in both convergence rate and time saving.

和文概要：ニューラルネットワークによる衛星画像の分類において学習に時間がかかることが問題となる。本論文はバックプロパゲーション学習の収束速度の改良に関するものである。学習時間を短縮するため、学習速度常数(η)と安定化定数(モーメント項: α)を同時に制御するようなメカニズムを導入し、バックプロパゲーション学習速度を高める方法について検討した。提案する方法の有用性を検証するため、実際に、MOS-1 マルチスペクトル画像による西ジャワ・チャンジュール(Cianjur)の土地被覆分類に応用し、安定に、しかも、高速に学習が収束するための学習パラメータの制御条件について調べた。ネットワークは階層型

ニューラルネットワークを用いた。その結果、提案するメカニズムを用い、学習パラメータを制御することにより、約10倍の学習速度の高速化が得られることが示された。また、提案方法は、学習パラメータ(η , α)の初期設定値に依存せず、安定に収束することが知られた。

1. Introduction

The rapid development of remote sensing satellite generated multispectral data from a variety of sensors. The analysis of remote sensing data already presents a formidable challenge. The challenge in dealing with analysis of remote sensing data problems lies in the specification and development of algorithm system in which can handle the data. In contrast, the development of an artificial neural network starts with an inherently parallel processing technique. That has a natural propensity for storing experiential knowledge and making it available for use in which knowledge is acquired by the network through a learning process and a synaptic

* Graduate School of Science and Technology, Chiba University, Japan

千葉大学大学院自然科学研究科情報システム科学専攻

**Faculty of Engineering, Chiba University, Japan

千葉大学工学部

1-33 Yayoi-cho, Inage-ku, Chiba-shi, Chiba 263, Japan

〒263 千葉市稲毛区弥生町1-33

E-mail : sadly@icsd5.tj.chiba-u.ac.jp

yasuda@ics.tj.chiba-u.ac.jp

「写真測量とリモートセンシング」VOL. 38, NO. 6, 1999

weight are used to store the knowledge.

The use of artificial neural network for remote sensing has been motivated by the realization that the human brain is very efficient at processing vast quantities of data from a variety of different source. That performs more rapidly than other techniques and incorporate a priori knowledge and realistic physical constraints into the analysis¹⁾. The application of neural networks to remote sensing data analysis has been reported in^{2)~5)}. The backpropagation algorithm⁶⁾ is probably the mostly known of the learning algorithm widely used in remote sensing applications. However, it has several serious problems associated with this algorithm. Specifically, since most remotely sensed data volume are bulky and complex, and thus take the excessively long training times required in achieve the acceptable errors, its slow to convergence, and may lead to an unstable of training process.

Research into deterministic convergence of the backpropagation has been reported in^{8),9)}. In particular, Jacobs in⁸⁾ suggested heuristics for achieving faster rates of convergence, which should be viewed as useful guidelines for thinking about how to accelerate the convergence. Based upon the heuristics a learning rate update rule has been introduced. Study by Parlos et al. in⁹⁾ proposed an accelerated learning algorithm for the supervised training. The algorithm updates the weights in the direction of steepest descent, but with the learning rate a specific function of the error and of the error gradient norm have been shown. The disadvantages of these algorithms to be the possibly jumpy behavior of the weights as a local or a global minimum is approached.

As mention above, a number of modifications have been proposed to improve the convergence of the backpropagation algorithm. These methods would require a proper selection of the learning parameters by using trial and error method. Most of these methods concentrate on problems with small data sets.

In this paper, a self-built learning control mechanism in backpropagation training, under the condi-

tions encountered in processing remote sensing data is introduced. The approach based on the self-built learning control mechanism of controlling the learning rates and tuning of momentum term at the same time automatically in training of the feedforward neural network. Moreover, the automatic initials learning rate selection and the optimal values of the learning control parameters are experimentally discussed. The simulations are presented to indicate the remarkable advantages of the proposed approach in both convergence rate and time saving. The proposed method is verified by a land cover study over Cianjur area, West-Java, Indonesia, with MOS-1 multi-spectral imagery.

2 . Problem Notation and Definition

We consider a feedforward neural network with single hidden layer denoted by I-J-K (See Figure. 1), where I, J and K are the number of neurons in the input layer, the hidden layer and the output layer respectively. We assume that a number of inputs and corresponding output values are given. With, $x_i^p, i=1, 2, \dots, I$. We denote the inputs for a specific pattern p . The corresponding output values for the same pattern will be indicated $d_k^p, k=1, 2, \dots, K$. For each hidden unit, $j=1, 2, \dots, J$ define

$$o_j^p = f_j \left(\sum_{i=1}^I w_{ij} x_i^p + bias_j \right) \quad (1)$$

where f_j is the sigmoid activation function for the j th neuron in the hidden layer, and $bias_j$ is bias added to the j th neuron. Similarly, for each output unit, $k=1, 2, \dots, K$ define

$$o_k^p = f_k \left(\sum_{j=1}^J w_{jk} o_j^p + bias_k \right) \quad (2)$$

where w_{jk} weights corresponding to the connection from the j th neuron to the k th neuron (from hidden to the output layer), $bias_k$ is bias added to the k th neuron.

For a given training set, the basic mechanism behind most supervised learning rules is the updating of the weights and the bias terms, until the sum of squared error between the output predicted by

the network and the desired output (target) is achieve the error goal. The summed squared error (SSE) of the network we want to minimize in the connection weight space is defined as^{(5),(7)}

$$SSE = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^K (d_k^p - o_k^p)^2 \quad (3)$$

where d_k^p and o_k^p are, respectively, the desired and output vectors corresponding to the p_{th} training patterns and o_k^p is given by Eq. (2).

The backpropagation algorithm is normally used to train multi-layer feedforward neural network. This algorithm update weights by the method of steepest descent on the error surface. The algorithm updates the weights after each presentation of a subset of the training patterns according to the following equation^{(6),(7)}:

$$w(t+1) = w(t) - \eta \frac{\partial E_p(t)}{\partial w(t)} \quad (4)$$

where η is the constant learning rate, $w(t+1)$ is the weight update value in the corresponding weight at time $(t+1)$.

As such, the backpropagation algorithm with a fixed learning rate would be a low efficient. This is because the learning rate has to be kept small to prevent over-adjustment of weights, which may occur in the error surface regions with extreme steepness. Typically, this is due to the characteristics of the error surface. These characteristics are particularly pronounced in learning convergence problems of backpropagation method.

3 . Proposed Method

The proposed method is also a “steepest descent” algorithm, i. e. the weights are updated in the negative direction of the gradient. The difference are that we apply a self-built learning control mechanism for controlling the learning rate and tuning the momentum term at the same time based on the error condition automatically. Chosen such as to accelerate the speed of convergence and to keep the learning process that does not cause instability.

We have observed that the inclusion of the

momentum term does contribute to faster convergence of the algorithm without leading to oscillation. In order to optimize the effect of momentum term on the sequence of pattern presentations on the synaptic weights, in this research we presents how to involve and tuning the momentum term automatically during training. The algorithm updates the weights by including a momentum term can be modified in terms of the following equation^{(6),(7)},

$$w(t+1) = w(t) - \eta(t+1) \frac{\partial E(t)}{\partial w(t)} + \alpha \Delta w(t-1) \quad (5)$$

where, $\eta(t+1)$ is the learning rate value corresponding to $w(t+1)$ at time $(t+1)$ and α momentum term, respectively. $\Delta w(t-1)$ is the change in the corresponding weight at time $(t-1)$.

A . Learning phase using the self-built learning control mechanism

The control mechanism is used to control the learning parameters based on the condition of sum of squared error (SSE) as illustrated in equation (3) in the weight space during training. Let $w(t)$ denote the value of a single weight at time t and $\eta(t)$ denote the learning rate value corresponding to $w(t)$ at time t .

We now introduce the modification of learning rates dynamically and tuning the momentum term automatically using the learning control mechanism is given as follows:

(1) If the new SSE is greater than the previous SSE (if new error has increased), the learning rate is reduced multiplied by the decrease factor (C1), and the updating of learning rates dynamically can be expressed in terms of the following identities:

$$\eta(t+1) = \eta(t) \times C1; \quad 0 < C1 < 1 \quad (6)$$

Then, the momentum term (α) is switch to OFF (disable).

Where, C1 is the decrease factor of learning rate. This condition, momentum term is disable, because pushing the search in the wrong direction and the learning rate should be reduced to find the proper direction in the error surface. This switch OFF momentum from pushing a network's parameters

out of a deep valley in the error surface.

(2) If the new SSE is less than the previous SSE (if new error has decreased), the learning rate is increased multiplied by the increase factor ($C2$) and the updating of learning rates dynamically can be expressed in terms of the following identities :

$$\eta(t+1)=\eta(t)\times C2; \quad 1 < C2 \leq 1.2 \quad (7)$$

Then momentum term (α) is switch to ON.

Where, $C2$ is the increase factor of learning rate. This condition, where the error decrease, the momentum term is switch to ON with larger value as (0.9) and the learning rate is increased using increase factor.

B . Self-built learning control parameters

The choice of these learning control parameters ($C1$ and $C2$) are based on the experience through the experiment to date and will probably change as development continues, particularly on the way of the learning rates updating part. Theoretically, if the learning rate value is too large, can speed up the learning process. Unfortunately, this condition may lead to unstable training process or it may not train at all. If the learning rate value is too small, learning process takes longer training time. Based on these facts, the appropriate choice of the learning control parameters determined experimentally.

The computational procedures with the self-built learning control mechanism are as follows :

Step 1.

- Read inputs data and desired outputs
- Initialize all weight and bias values to be small random values.

Step 2.

- Set parameters for training process
- For each training cycle do
  - perform self-built learning control mechanism (controlling the learning rates and tuning the momentum term automatically)

Step 3.

- After the presentation of all the training patterns that constitute an epoch :

- Update Weights (batch update mode)

Step 4.

- Compute Sum Squares Error (SSE)

Step 5.

- Compute the learning time process and the number of epochs
- Compute the actual output neural network

Step 6.

- If SSE becomes sufficiently small (Error Goal), Then Go To STOP

Otherwise Go To Step 2.

STOP

4 . Experiments

The purpose of the experiment in this research is to demonstrate and to verify the abilities of the proposed method in the following characteristics : (a). Learning performance, included learning time process and stability of learning performance, (b). The optimization of the learning control parameters.

4 . 1 Network architecture

The architecture of neural network used in the experiment is as shown in Figure. 1. We consider a feedforward neural network with single hidden layer denoted by I-J-K, where I, J and K are the number of neurons in the input layer, the hidden

Figure. 1 Architecture of feedforward network used in the experiments

layer and the output layer respectively. The input layer neurons are fully connected to the hidden layer neurons in which are fully connected to the output layer neurons. A commonly used activation is a sigmoid function. The number of input neurons was specified by the dimension of the input patterns and contained three band of input data.

4.2 Data sets

Data from MOS-1 imagery of Cianjur area consists of three spectral bands shown in Figure. 2 was used as input to the neural network. The size of the imagery used in the experiments was 128×128 pixels. To present the data to the network, an input pattern was constructed by concatenating the data from all three bands at each pixel location. However, the proper selection of training data was a crucial step in achieving best results. In order to choose training areas or desired output of classification more precisely from the spectral information at the input data, we employed the unsupervised clustering method, ISODATA (Iterative Self-Organizing data Analysis Technique). We applied the unsupervised clustering to select the training data at different number of categories. Here a training data set was selected based on minimum spectral distance to assign a cluster for each candidate pixel into some categories. These training data set were used to train the network.

Figure. 2 Input data from MOS-1 of Cianjur area, Indonesia (3 Channels)

4.3 Comparisons of learning performances

To verify the capability on the learning performance, the proposed method was compared to the backpropagation method. The learning parameters being used in this comparison were as follows: the initial learning rates (η) of 0.001, 0.01; 0.1; 0.3; 0.6; and 0.9; the momentum terms (α) of 0.1, 0.25, 0.75 and 0.9; the stopping criteria fixed to 0.255; and the number of hidden neurons are 10 and 20 were selected. At the start of each simulation, the weights of the network were initialized to random values between -0.1 and 0.1 . During training of the network, each time step is called an epoch and is defined to be a single sweep through all training patterns. Weights updating were performed after the presentation of all training patterns that constitute an epoch (batch mode). After the learning parameters were selected, the performances of both methods were compared.

4.4 Automatic initial learning rate selection

The objective of this experiment were to investigate and compare the performance of the proposed method and the backpropagation learning algorithm using the automatic initial learning rate selection. In this investigation, the initial learning rate selection proposed by McClelland [11] was adopted. This method recommends $1/n$ as the proper size for the initial learning rate, where n is the total number of nodes in the network ($n=I+J+K$). The effects of different number of patterns (16, and 32 of categories), and different number of hidden neurons on the epoch using initial learning rate selection will be investigated. Momentum term fixed to 0.9 and 0.255 was used as stopping criteria.

4.5 Determining the optimal learning control parameters and classification

This experiment was to select the optimal of learning control parameters and to show the effect of the learning control parameters on the epoch. The self-built learning control parameters ($C1$ and $C2$) in Eq. (6) and (7) are given as follows:

Decrease factor of learning rate (C1)
: 0.2; 0.3; 0.4; 0.5.
Increase factor of learning rate (C2)
: 1.005; 1.01; 1.05; 1.10; 1.15; 1.20.

In order to select the optimal values of the learning control parameters, the experiment using each of the learning control parameters with topology of 3-10-16, the stopping criteria of 10 (Error Goal=EG=10), the initial learning rate equal to 0.01 and the momentum term of 0.9 were used.

In general, single hidden layer networks to be suitable for most classification problem⁷⁾. The classification performance on the effects of increasing number of hidden neurons (number of hidden neurons: 10, 16, and 32) at different error goal (EG=10, 1, 0.1 and 0.01) and the number of patterns was fixed to 6 will be investigated.

5 . Results and Discussions

5.1 Comparisons of learning performances

Table.1 shows the comparison results for the learning performance of backpropagation method and proposed method using MOS-1 image. Using the larger initial learning rate values (for examples, $\eta=0.6$ and 0.9), in case of backpropagation method, the learning process become unstable. That means, the learning process was never achieve the error goal or not training at all. The effects of the momentum term on the epoch using switching (ON-OFF) during training process were investigated. In case of (3-10-16) topology and initial learning rate equal to 0.01, it was found that the number of epoch needed of the proposed method using switching ON-OFF ($\alpha=0.9$) was about 29 % from the total epoch required without using momentum term ($\alpha=0$). Moreover, it was also found that, the number of epoch needed of the proposed method was about 18.8 % from the total epoch required in the backpropagation with momentum term ($\alpha=0.9$). Based on these results, the proposed method was speed-up the convergence more appropriately compared with backpropagation method.

Figure.3 Comparative results of backpropagation method and proposed method

Figure.3 shows the comparative results of proposed method and the backpropagation using the number of hidden neuron 20. To achieve the error goal, the proposed method only needed 2230 epochs and about 2.21 minutes for learning time, but the backpropagation method needed 12436 epochs and about 21.1 minutes for learning time. In other words, the proposed method was decreased learning time needed in the backpropagation about more than 10 times.

5.2 Effects of automatic initial learning rate selection

Table. 2 shows the comparison of learning performance of backpropagation method and proposed method using the initial learning rate selection proposed by McClelland at different number of hidden neurons and number of patterns. In contrast, the application of the learning rate selection method to the backpropagation during experiment needs more number of hidden neurons. Because, if using the small number of hidden neurons that did cause instability of learning process.

By using the proposed method, using a small number of hidden neurons was enough to the success of learning process, and training speed was increased. The adopted method was depend on the total number of neurons (nodes) on the network, which includes the input neurons, hidden neurons and the output neurons (the number of patterns to be

Table. 1 Comparisons on learning performance of both methods

Topology	Initial Learning Rate (η)	Momentum Term (α)	Backpropagation Method		Proposed Method	
			Learning Time	Epoch	Learning Time	Epoch
3-10-16	0.001	0	14 : 05 : 10	85990	00 : 08 : 45	7717
		0.1	08 : 06 : 00	30155	00 : 06 : 33	6658
		0.25	07 : 48 : 00	28351	00 : 05 : 32	5169
		0.75	06 : 09 : 00	22562	00 : 03 : 19	2948
		0.9	05 : 18 : 00	20475	00 : 02 : 29	2360
3-10-16	0.01	0	08 : 35 : 15	32760	00 : 08 : 15	7685
		0.1	00 : 22 : 24	13974	00 : 07 : 12	6674
		0.25	00 : 20 : 01	11726	00 : 05 : 18	5135
		0.75	00 : 20 : 27	12203	00 : 03 : 10	2909
		0.9	00 : 20 : 15	11861	00 : 02 : 19	2234
3-10-16	0.1	0	08 : 32 : 11	31892	00 : 08 : 04	7560
		0.1	00 : 22 : 01	13685	00 : 06 : 30	6632
		0.25	00 : 20 : 00	11700	00 : 05 : 12	5096
		0.75	00 : 19 : 51	11690	00 : 03 : 03	2890
		0.9	00 : 19 : 38	11644	00 : 02 : 09	2204
3-10-16	0.3	0	08 : 35 : 10	32680	00 : 08 : 29	7700
		0.1	00 : 22 : 06	13910	00 : 06 : 31	6641
		0.25	00 : 21 : 47	12741	00 : 05 : 14	5126
		0.75	00 : 20 : 50	12340	00 : 03 : 15	2915
		0.9	00 : 20 : 33	11942	00 : 02 : 24	2244
3-10-16	0.6	0	Unstable		00 : 08 : 25	7684
		0.1			00 : 06 : 31	6640
		0.25			00 : 05 : 13	5100
		0.75			00 : 03 : 08	2897
		0.9			00 : 02 : 25	2305
3-10-16	0.9	0	Unstable		00 : 08 : 28	7695
		0.1			00 : 06 : 31	6644
		0.25			00 : 05 : 24	5152
		0.75			00 : 03 : 01	2862
		0.9			00 : 02 : 33	2560
3-20-16	0.01	0.1	00 : 19 : 31	11211	00 : 06 : 27	6614
		0.25	00 : 19 : 13	11088	00 : 05 : 24	5150
		0.75	00 : 20 : 07	11838	00 : 03 : 03	2891
		0.9	00 : 21 : 06	12436	00 : 02 : 12	2230

Table. 2 Comparisons result using automatic initial learning rate selection method

Topology I-J-K	Momentum (α)	Initial learning rate $\eta=1/n$ n : total number of nodes (I+J+K)	Backpropagation Method	Proposed Method
			Epochs	Epochs
3-5-16	0.9	0.0416	Unstable	2159
3-10-16		0.0345	Unstable	2128
3-20-16		0.0256	Unstable	2120
3-30-16		0.0204	6364	2118
3-35-16		0.0185	7898	2142
3-40-16		0.0169	7223	2144
3-50-16		0.0145	8473	2147
3-5-32	0.9	0.0250	Unstable	2044
3-10-32		0.0222	Unstable	2035
3-20-32		0.0182	7038	2027
3-30-32		0.0154	7968	2021
3-35-32		0.0143	8782	2018
3-40-32		0.0133	9329	2024
3-50-32		0.0118	9856	2022

classified). The advantages of using the initial learning rate selection, the initial value of learning rate was automatically determined. The adopted method was depend on the total number of neurons (nodes) on the network, which includes the input neurons, hidden neurons and the output neurons (the number of patterns to be classified).

5.3 Optimal learning control parameters and classification performance

Figure. 4 shows the effects of learning control parameters on the epoch. It was shown that, the decrease factor of learning rate of 0.4 ($C1$) and the increase factor of learning rate of 1.05 ($C2$) were achieved as the optimal values of learning control parameters. These values were used in order to achieve fast convergence. Figure. 5 shows the average classification performance as function of error goal at different number of hidden neurons with 6 categories. It was shown that an increase in the number of hidden neuron was increased the classification performance (about 0.85% to 2.27%). Moreover, Figure. 6 shows the land cover classification

Figure. 4 Effects of learning control parameters on the epoch (initial $\eta=0.01$; $\alpha=0.9$; Error Goal=10)

result over Cianjur area with 6 categories over study area obtained from the proposed method.

Table.3 shows the categories of classification results over study area corresponding to the number of pixels of each category. The 6 categories over study area consists, class 1 (Water, pixel number : 401), class 2 (Irrigated Paddy field, pixel number :

Figure. 5 Plot of the average of classification performance as function of error goal

Figure. 6 Land cover classification result over Cianjur area with 6 categories

Table. 3 Categories of classification result and number of pixels over study area

Category (Class)	MOS-1 Image (Cianjur area)	
	Number of Pixel	Type of Class
1	401	Water
2	2961	Paddy field (irrigated)
3	3778	Grass Land
4	2367	Plantations
5	2887	Forest
6	3990	Settlement/Urban area

2961), class 3 (Grassland, pixel number : 3778), class 4 (Plantations, pixel number : 2367), class 5 (Forest, pixel number : 2887), and class 6 (Settlement, pixel number : 3990).

5.4 Learning rates updating conditions

These results presents the performances of the learning rate updating and the error characteristic corresponding to the epoch by using the proposed method with optimal learning control parameters compared to other control parameters. The same learning parameters (initial $\eta=0.01$; $\alpha=0.9$; Error Goal=10) were used.

Figure. 7 presents an example of updating of learning rate parameters until convergence using the optimal value of learning control parameters ($C1=0.4$, $C2=1.05$). It was shown that, the increasing of learning rates repeated every 100 epochs and move to the upper limit value up to about 0.6 followed by the decreasing of error. To achieve the convergence only needs about 284 epoch. Figure. 8 shows an another example of updating of learning rate parameters until convergence using the learning control parameters ($C1=0.4$, $C2=1.15$). It was shown that, the increasing of learning rates repeated every 50 epochs and move to the upper limit value more than 0.6 and needs 400 epoch until convergence.

Further, Figure. 9 using the increase factor of learning rate larger than before ($C1=0.4$, $C2=1.20$). The effect of the increasing of the increase factor of learning rate was move the updated value

Figure. 7 An example of updating of learning rate parameters using learning control parameters (Control Parameters: $C1=0.4$; $C2=1.05$; initial $\eta=0.01$; $\alpha=0.9$)

Figure. 8 An example of updating of learning rate parameters using learning control parameters (Control Parameters: $C1=0.4$; $C2=1.15$; initial $\eta=0.01$; $\alpha=0.9$)

Figure. 9 An example of updating of learning rate parameters using learning control parameters (Control Parameters: $C1=0.4$; $C2=1.20$; initial $\eta=0.01$; $\alpha=0.9$)

Figure. 10 An example of updating of learning rate parameters using learning control without momentum term (Control Parameters: $C1=0.4$; $C2=1.05$; initial $\eta=0.01$)

Figure. 11 An example of updating of learning rates using learning control parameters without momentum Control OFF (Control Parameters: $C1=0.4$; $C2=1.05$; initial $\eta=0.01$; $\alpha=0.9$)

to the higher position about 0.8 and takes 449 epoch until convergence. From the results shown in Figure. 7, 8, and 9 indicated that the increasing of increase factor of learning rate with the same decrease factor of learning rate was increased the number of epoch.

Figure. 10 presents an example of updating of learning rate parameters using the optimal value of learning control parameters ($C1=0.4$, $C2=1.05$),

but without using the switch ON-OFF of momentum term ($\alpha=0$). It was found that, to achieve convergence needs many epochs compared with others (about 747 epoch). Moreover, Figure. 11 shows an example of updating of learning rates using learning control parameters ($C1=0.4$, $C2=1.05$), but without control OFF of momentum term ($\alpha=0.9$). It was indicated that, to achieve convergence needs 1009 epochs. This occurs because dur-

ing learning process the error is search in the wrong direction on the error surface.

However, the integration between controlling of the learning rates and tuning the momentum term by using switch ON-OFF at the same time automatically that called the self-built learning control mechanism were gave the best performance compared to others.

6 . Conclusion

A self-built learning control mechanism for controlling the learning rates and tuning the momentum term automatically in backpropagation training, under the conditions encountered in processing remote sensing data has been presented. As a result of experiments, the proposed method was decreased learning time needed in the backpropagation about more than 10 times. It was shown that the number of epoch needed of the proposed method (using tuning of the momentum term during training process) was only about 29% from the total epoch required without using momentum term ($\alpha=0$). Further, the number of epoch needed of the proposed method was about 18.8% from the total epoch required in the backpropagation with momentum term ($\alpha=0.9$).

By using the proposed approach, the optimization of the learning control parameter was determined and the initial learning parameters were not crucial to the success of the training, because the learning parameters were controlled that did not cause instability. The proposed approach provides a practical and potential tool in the application of artificial neural networks for remote sensing data analysis.

(受付日1999.9.13, 受理日1999.11.12)

References

1) P.M. Atkinson and A.R.L. Tatnall, "Neural networks in remote sensing", Int. J. Remote Sensing,

Vol. 18, No. 4, 1997.

2) Y.C. Tzeng et al., "A Dynamic Learning Neural network for Remote Sensing Applications", IEEE Transactions on Geoscience and Remote Sensing, Vol. 32, No. 5, 1994.

3) A.K. Skidmore et al., "Performance of a Neural Network: Mapping Forests Using GIS and Remotely Sensed Data", Photogrammetric Engineering & Remote Sensing, No. 5, 1994.

4) M. Sadly and Y. Yasuda, "Artificial Neural Networks based Algorithm for Land Cover Classification & Mapping Using MOS-1 Image data", Proceeding of ASPRS-RTI International Conference, Tampa, Florida, U.S.A., pp.1505-1510, March 1998.

5) H. Bischof, W. Schneider, and A.J. Pinz, "Multispectral Classification of Landsat-Images Using Neural Networks", IEEE Trans. On Geoscience and Remote Sensing, Vol. 30, No. 3, 1992.

6) D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation", in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, D.E. Rumelhart and J.L. McClelland, Eds., Vol. 1, Cambridge, MA: MIT Press, 1986.

7) Simon Haykin, "Neural Networks, A Comprehensive Foundation, IEEE Press. 1994.

8) R.A. Jacobs, "Increased rates of convergence through learning adaptation", Neural Networks, Vol. 1, 1988.

9) Alexander G. Parlos, Benito Fernandez et al. "An Accelerated Learning Algorithm for Multilayer Perceptron Networks", IEEE Transaction on Neural networks, Vol. 5, 1994.

10) Marco Gori and Alberto Tesi, "On the problem of local minima in backpropagation", IEEE Transaction on PAMI, Vol. 14, No. 1, 1992.

11) D.G. Luenberger, "Linear and Nonlinear Programming", (2nd ed.), Menlo park, CA: Addison-Wesley, 1984.