A New Method to Prune the Neural Network

Wan, Weishui
Department of Electrical and Electronic Systems Engineering, Kyushu University : Graduate Student

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

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

https://doi.org/10.15017/1513726

出版情報：九州大学大学院システム情報科学紀要. 5 (1), pp.43-49, 2000-03-24. 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5巻 第1号 平成12年3月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.5, No.1, March 2000

A New Method to Prune the Neural Network

Weishui WAN* , Kotaro HIRASAWA**, Jinglu HU** and Chunzhi JIN**

(Received December 10, 1999)

Abstract: Using backpropagation algorithm(BP) to train neural networks is a widely adopted practice in both theory and practical applications. But its distributed weight representation,that is the weight matrix of final network after training by using BP are usually not sparsified, and prohibits its use in the rule discovery of inherent functional relations between the input and output data, so in this aspect some kinds of structure optimization are needed to improve its poor performance. In this paper with this in mind a new method to prune neural networks is proposed based on some statistical quantities of neural networks. Comparing with the other known pruning methods such as structural learning with forgetting(SLF)[1] and RPROP algorithm[2], the proposed method can attain comparable or even better results over these methods without evident increase of the computational load. Detailed simulations using the Iris data sets exhibit our above assertion.

Keywords: Prune,Neural networks,Statistical quantity, Backpropagation algorithm.

1. Introduction

In neural network training, the most well-known online training method is the standard backpropagation algorithm (BP) [3], which is a first-order stochastic descent method. Its advantage includes simple structure of algorithm and local computation. On the other hand these features often lead to small performance achievement . One main disadvantage is the appearance of distributed weights representations on the hidden layers, which hinders regularity and rule extraction, and makes the trained network very difficult to understand. Current algorithms which are used to remedy this phenomenon and enhance the convergence rate , can be roughly divided into three types.

1. Many modified schemes using some kinds of approximation of the Hessian matrix of the error function(second order information), have been suggested in an attempt to speed up the training process [4][5]. The above algorithms only try to reduce the bias error with the approximation error being fixed to be constant[6].
2. The more efficient approach is to try to reduce the two components of the error simultaneously through optimizing the network structure to attain satisfactory solutions. Among them there are structure learning with forgetting algorithm[1], OBD[7] , and weight elimination method[8] etc. These methods often are specially tailored and do not detect correlated elements[4], which is one main disadvantange. Another difficult point of using these methods with the Hessian matrix is that one must guarantee that the Hessian matrix must not degenerate. Among these algorithms structure learning with forgetting algorithm[1] is a typical algorithm.
3. The theory supported methods are based on the use of the following evaluation criterion : network information Criterion(NIC) [10] and Vapnik-Chervonenkis(VC) dimension[11] etc.. Although they are supported by statistical theory and have the ability to detect the correlated information among nodes in theory, these methods tend to put too much ideal prerequisites to make their applications in practice possible or easy to be carried out , and the error bounds they offer are given in the distribution-free, worst case analysis form , which are often too large to be applicable practically except their theoretical guidance.

Our proposed method roughly belongs to the second type ,while using some ideas from statistics,and combining the respective advantages of the last two types. Also our algorithm is a general hybrid schema, in that one can use any BP-like learning algorithm which can replace the standard BP algorithm, and the network can be of any form such as forward neural network(NN)etc.. In this paper we use the forward NN for explanation.

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

** Dept. of Electrical and Electronic Systems Engineering

The rest of this paper is organized as follows. In section 2 we introduce a new method to prune the network during the process of BP algorithm . The proposed pruning method is not constrained to BP and it can be used to the other learning algorithm. In section 3 a short introduction to the SLF and RRPOP is made. In section 4 simulation results on Iris dada sets are given. The final section is conclusion.

2. New pruning algorithm

The main disadvantage of training network by BP is that the final weight matrix of the resulting network is often not sparse,and this property can deteriorate the generalization ability of networks. In simulation section there is a concrete example for this. The proposed new algorithm tries to prune the network with two simple statistical quantities during the process of network learning which is easy to use and in this algorithm correlation among nodes during training are considered. This makes the proposed new algorithm much more powerful than the other algorithms.

Before describing the proposed algorithm , we first introduce some notations which are needed in the following explanation. Suppose there are M sample pairs $\{(\xi_k^\mu, \zeta_k^\mu)\}_{\mu=1}^M$. Also a three-layer feed-forward NN with full connections between the consecutive layers is considered and no connection among the nodes in the same layer is supposed without loss of generality. The notations are given below.

Notations:

ξ_k^μ	input patterns ,$k = 1, 2, \cdots, K$
ζ_i^μ	corresponding output values , $i = 1, 2, \cdots, I$
h_j^μ	input to hidden layer ,$j = 1, 2, \cdots, J$
H_i^μ	input to output layer ,$i = 1, 2, \cdots, I$
o_j^μ	output of hidden layer ,$j = 1, 2, \cdots, J$
w_{jk}	weights for the arcs from the input layer to the hidden layer
W_{jk}	weights for the arcs from the hidden layer to the output layer
g_j	athe j-th node of the hidden layer
g'_j	first derivatives of $g(.)$
G_i	activation function for the j-th node of the output layer
G'_i	first derivatives of $G(.)$
$\phi(w, W)$	object function (error function)
$\phi_\mu(w, W)$	individual error function

A graphical representation of the neural network is shown as **Fig.1**

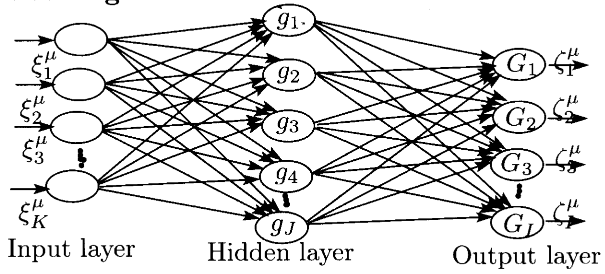

Fig.1: General Backpropagation

The proposed method is described as follows:

1. Define the following quantities : σ and δ which will be used in the following step. The selection of these two quantities determines the pruning rate and pruning performance. If they are too large, no pruning or small pruning is effected, on the other hand if they are too small , the pruning time may be long.
2. Choose the pattern(μ),randomly initialize the weights of the network;
3. Compute h_j^μ for all $j = 1, \cdots, J$;
4. Compute H_i^μ for all $i = 1, \cdots, I$;
5. Update the weights for the arcs from hidden to output layer by
$$W_{qr}^{\text{new}} = W_{qr}^{\text{old}} + \eta[\zeta_q^\mu - G_q(H_q^\mu)] \times G'_q(H_q^\mu) g_r(h_r^\mu) \quad (1)$$
for $q = 1, \cdots, I$ and $r = 1, \cdots, J$;
6. Update the weights for the arcs from input to hidden layer by
$$w_{st}^{\text{new}} = w_{st}^{\text{old}} + \eta \sum_{i=1}^{I} [\zeta_i^\mu - G_i(H_i^\mu)] \times G'_i(H_i^\mu) \times W_{is}^{\text{old}} g_s(h_s^\mu) \quad (2)$$
for $s = 1, \cdots, J$ and $t = 1, \cdots, K$;
7. Compute the following quantities for each hidden unit $\{j = 1, 2, \ldots, J\}$ at predefined iteration steps: $\{t_i\}_{i=1}^{\infty}$.

$$Mean_{j,\text{in}}^{\text{new}} = \frac{\sum_{k=1}^{K} w_{jk}^{\text{new}}}{K} \quad (3)$$

$$Var_{j,\text{in}}^{\text{new}} = \frac{\sum_{k=1}^{K} (w_{jk}^{\text{new}} - Mean_{j,\text{in}}^{\text{new}})^2}{K} \quad (4)$$

$$Mean_{j,\text{out}}^{\text{new}} = \frac{\sum_{i=1}^{I} W_{ij}^{\text{new}}}{I} \quad (5)$$

$$Var_{j,\text{out}}^{\text{new}} = \frac{\sum_{i=1}^{I} (W_{ij}^{\text{new}} - Mean_{j,\text{out}}^{\text{new}})^2}{I} \quad (6)$$

$$Mean_{j,\text{total}}^{\text{new}} = \frac{K}{K+I} Mean_{j,\text{in}}^{\text{new}} +$$

$$\frac{I}{K+I} Mean^{\text{new}}_{j,\text{out}} \tag{7}$$

$$Var^{\text{new}}_{j,\text{total}} = \frac{K}{K+I} Var^{\text{new}}_{j,\text{in}} + \frac{I}{K+I} Var^{\text{new}}_{j,\text{out}} \tag{8}$$

8. Compare the following pair quantity: $\{Mean^{\text{new}}_{j,\text{total}}, Var^{\text{new}}_{j,\text{total}}\}$ with the last corresponding quantity: $\{Mean^{\text{old}}_{j,\text{total}}, Var^{\text{old}}_{j,\text{total}}\}$ for each hidden unit $\{j = 1, 2, \ldots, J\}$, then only the following cases will appear:
   - $|Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}}| \leq \sigma$ and $|Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}}| \leq \delta$. In this case there are two cases which should be discriminated;
     (a) If the error between the calculated value and the true value is very small, we can fix the only related weights, not letting these weights to be trained in the following steps;
     (b) If the error between the calculated value and the true value is not very small , then we will add random values only to these weights . This method can be termed a local version of the widely used restart methods in that we do change only part of the weights instead of all the weights in the network.
   - $|Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}}| \leq \sigma$ and $Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}} > \delta$. In this case we can think there is no evident change in the mean value of weights while evident increase exists between the variance of weights. This also means one can delete some small weights which are of little effects on the representative of neural networks. This is very useful for attaining sparse representation of neural networks;
   - $|Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}}| \leq \sigma$ and $Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}} < -\delta$. In this case we can think that there is no evident change in the mean value of weights while evident decrease exists between the variance of the weights. In this case, to sparsify the network , we try to increase the learning rate by adding a random value to the current learning rate to change the related weights .
   - $Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}} > \sigma$ and $|Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}}| \leq \delta$. In this case we can think these weights are being trained,no pruning is needed at this time;
   - $Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}} > \sigma$ and $Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}} > \delta$. This is an ideal case and the structure of the network is becoming compact ;
   - $Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}} > \sigma$ and $Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}} < -\delta$ .In this case a new random value is added to the current learning rate to change the related weights;
   - $Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}} < -\sigma$ and $|Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}}| \leq \delta$. In this case the mean value of weights are decreasing while its variance remains unchanged approximately, that is to say, although these weights are almost simultaneously decreasing, some weights are still needed to be trained to get a good final solution. Therefore after some steps only some of these weights will be deleted;
   - $Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}} < -\sigma$ and $Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}} > \delta$. This is an acceptable case , after certain steps, some weights connecting to this node may be deleted without affecting on the representation of network. This is just the pruning;
   - $Mean^{\text{new}}_{j,\text{total}} - Mean^{\text{old}}_{j,\text{total}} < -\sigma$ and $Var^{\text{new}}_{j,\text{total}} - Var^{\text{old}}_{j,\text{total}} < -\delta$. This phenomenon tells that these weights are all simultaneously degenerating . They may be deleted at a future appropriate step. This case may result in the deletion of node, because the weights connecting to this node are simultaneously decreasing to zero.

Note that in the above algorithm although we use the pair quantities $\{Mean^{\text{new}}_{j,\text{total}}, Var^{\text{new}}_{j,\text{total}}\}$, you can also use the pair quantity $\{Mean_{j,\text{in}}, Var_{j,\text{in}}\}$ and $\{Mean_{j,\text{out}}, Var_{j,\text{out}}\}$ serially or use only one pair of them.

3. Related algorithms

The algorithms listed in this section will be used to make comparison with our proposed algorithm.

3.1 structural learning with forgetting(SLF)

This algorithm is proposed by Ishikawa[1],it is a simple yet effective learning method ,which has the following significant advantages as claimed:

1. SLF can discover regularities or extract rules from data without initial theories and processing;

2. Learning with trial and error for finding an appropriate network structure is no longer necessary because a skeletal network emerges;
3. SLF has good generalization ability due to the simplicity of the resulting network structure;
4. SLF has a simple criterion function and learning rule. Therefore it means only small extra computational cost;
5. SLF can determine the relative weight of a penalty term in cooperation with NIC etc.;
6. SLF can be extended to the learning of recurrent networks.

In our above notations ,structural learning with forgetting can be summarized as composing of the following steps:

1. Apply the learning with forgetting to obtain a rough skeleton network structure. In this step the criterion function used is

$$F_f = \sum_{k=1}^{K} \phi(w, W) + \epsilon \times \sum_{i=1}^{J}(\sum_{k=1}^{K} |w_{ik}| + \sum_{j=1}^{K} |W_{ik}|) \qquad (9)$$

where ϵ is a relative weight. The weight changes(same for W_{ik}) are given as follows:

$$\Delta w_{ik} = -\eta \times \frac{\partial F}{\partial w_{ik}} = -\eta \times \frac{\partial \phi(w, W)}{\partial w_{ik}} + \epsilon \times sgn(w_{ik}) \qquad (10)$$

where ϵ is the learning rate.

2. Apply the learning with hidden units clarification to dissipate distributed representations. This is skipped when the target outputs are not binary. In this step the criterion function is

$$F_h = F_f + c \times \sum_{j=1}^{J} min\{1 - h_j, h_j\} \qquad (11)$$

where h_j is the output of hidden unit j satisfying $h_j \in [0, 1]$ and c is the relative weight of the penalty term.

3. Apply both the learning with selective forgetting and hidden units clarification to get better learning performance in terms of mean square error. In this step the following criterion is used

$$F_s = \sum_{k=1}^{K} \phi(w, W) + \epsilon' \times (\sum_{|w_{ik}|<\theta} |w_{ik}| + \sum_{|W_{ik}|<\theta} |W_{ik}|) \qquad (12)$$

where θ is a threshold.

3.2 RPROP algorithm

This algorithm is proposed by Riedmiller et. al.[2], after the extensive experiments using typical data sets Moreira and Fiesler[5] concluded that RPROP algorithm has the average good performance with lowest computational complexity among some neural network algorithms with adaptive learning rate and momentum terms. In fact this is not an adaptive learning rate method. Referring to the learning rate adaptation techniques in general , the authors state that as the weight update value is composed both of the adapted learning rate and the value of the derivative , the effect of the former can be in some cases be disturbed by the unforeseeable values of the latter, making the recently performed learning adaptation useless. Therefore they proposed RPROP algorithm using a technique that although based on the sign of the local gradient in consecutive iterations ,differs considerably from all the other adaptation techniques and even from the standard BP algorithm, since the weight updates is done directly without using the derivative nor the learning rate.That means that at each iteration a certain quantity Δw_{ik} is added to each weight. The concrete updating formulae is:

$$\Delta w_{ik}^{\text{new}} = \begin{cases} 1.2 \times \Delta w_{jk}^{\text{old}} & \text{if } \frac{\partial \phi^{\text{new}}}{\partial w_{ik}^{\text{new}}} \times \frac{\partial \phi^{\text{old}}}{\partial w_{ik}^{\text{old}}} > 0 \\ 0.5 \times \Delta w_{jk}^{\text{old}} & \text{if } \frac{\partial \phi^{\text{new}}}{\partial w_{ik}^{\text{new}}} \times \frac{\partial \phi^{\text{old}}}{\partial w_{ik}^{\text{old}}} < 0 \\ \Delta w_{ik}^{\text{old}} & \text{otherwise} \end{cases}$$

further

$$w_{ik}^{\text{new}} = \begin{cases} w_{jk}^{\text{old}} + \Delta w_{ik}^{\text{new}} & \text{if } \frac{\partial \phi^{\text{new}}}{\partial w_{ik}^{\text{new}}} > 0 \\ w_{jk}^{\text{old}} - \Delta w_{ik}^{\text{new}} & \text{if } \frac{\partial \phi^{\text{new}}}{\partial w_{ik}^{\text{new}}} < 0 \\ w_{ik}^{\text{old}} & \text{otherwise} \end{cases}$$

$\Delta w_{ik} = 0$ is the proposed initialization. More detail please refer to [2].

4. Simulation results

The two algorithms introduced in section 3 will be used to make comparison with the proposed algorithm.

The classification of Iris data problem is a well known data classification problem[9] with continuous inputs and binary outputs. Irises are classified into three categories: setosa, versicolor and virginica. Each category has 50 samples. Each sample possesses 4 attributes : sepal length, sepal width,petal width and petal length. The characteristics of this data samples are that one of the 3 classes is well separated from the other two, which are not easily separable due to the overlapping of their convex hulls.

A network structure used here is a three-layer network : an input layer with 4 unit, hidden layer with 4 units and an output layer with 3 units. A subset of data is randomly chosen for training, and data are normalized before training. The generalization ability of the resulting network is evaluated by the percentage of classification errors on the average of 5 trials on the testing data. The assignment of input vectors to classes was based on a winner-takes-all strategy.

First we illustrate the disadvantage of distributed weight representation by BP learning through classification of Iris data problem. The weight distribution of one of the resulting network is shown in **Fig.2**. In **Fig.2** the upper part is the bar representation of weight matrix 3×4 from the output to the hidden layer, the lower part is the representation of weight matrix 4×4 from the hidden to the input layer. In this case the RPROP uses almost all of the units and connections. The simulation results are summarized in **Table 1** , **Table 2**(with 4 hiden units), **Table 3** and **Table 4**(with 10 hidden units), **Table 5** and **Table 6**(with 15 hidden units). The parameters in SLF and RPROP are taken from [1)2)] respectively. The parameters needed in the proposed algorithm are set as follows: $\sigma = 0.07$,$\delta = 0.005$, $\{t_i\}_{i=1}^{\infty} = \{100, 200, 300, ...\}$ and the added random values in the proposed algorithm are produced by the uniform distribution on the interval $[-0.07, 0.07]$.

When we set the No. of hidden units to 10 and 15, the following common phenomenon appears.

- From **Table 1** ,**Table 3** and **Table 5**, we can see when the No. of training samples is relatively small or large, the generalization capability of these 3 algorithms is almost the same. For the rest of data SLF and our algorithm always outperform the RPROP,in terms of not only the generalization ability but also the compactness of network. The latter point is clear from **Table 2** , **Table 4** and **Table 6**.

  For a specific task like this example there is a critical interval in terms of No. of training samples(if we fix the network structure and allow the training time sufficient long). For our example this domain consists of No. of training samples $\in$ [9,90] for 3-layer network with 4 hidden units. Just in this critical domain we can differentiate the efficiency and efficacy of differerent algorithms. Outside this interval, the resulting network trained by different algorithms tend to behave similarly, either poor in representation capability or rich in representation capability, which gives little room for the improvement of design of the algorithm. A similar phenomenon is also found in [1)],but is not expressed so clear as here.
- In the above critical interval the proposed algorithm is better in terms of generalization and the weight matrix of hidden units is much more sparse than RPROP and approximately equal to SLF in the mean sense. This is easily understood because SLF is much more refined and complicated algorithm ,while the proposed algorithm is just as simple as RPROP, and is easy to understand and use.
- From **Table 2** ,**Table 4** and **Table 6** we can see that for the same problem, the final weight matrix of the network trained by using the proposed algorithm and SLF becomes always much more sparse, with almost better generalization ability than that trained by RPROP . This is clear from the corresponding terms in **Table 1**,**Table 3** and **Table 5**.

5. Conclusions

The advantages of the new proposed method lies in that (1)our method does not use the second-order derivatives to make pruning like other widely used pruning methods ,such as OBD [7)], and the problems of OBD-like methods are that the Hessian matrix may degenerate. This will make the OBD-like methods invalid and a large storage is needed in implementation or need other numerical method to cope with this problem ; (2) The interrelation among related nodes can be easily considered and easily incorporated into the proposed pruning algorithm with the statistical methods. In this paper we used only first order correlation information of weight matrix. If the high order correlation information is used we can obtain much more efficient pruning algorithm; (3)From simulation we can easily see the network trained by using the proposed method is more sparsely distributed than the other methods. This is especially evident when compared with the BP algorithm. This point is especially useful for extracting the fundamental properties or rules from the data sets.

In the end of paper[4)],Reed suggests that: "A compensating advantages of the penalty-term methods is that training and pruning are effectively done in parallel so the network can adapt to minimize errors introduced by pruning." We think this conclusion is also true for all pruning methods. The new proposed method is developped to make the pruning during the traning process. The simulation partly proved Reed's assertion.

One disadvantages of the proposed method is that the attained network is not a robust or fault tolerant network. We think the fault tolerant property and rule extraction are completely different in that this two aims can not be attained simultaneously.

The proposed algorithm is a general schema, in that it can be incorporated into other kinds of NNs and other kinds of learning algorithms.

References

1) M.Ishikawa," Structure learning with forgetting ", *Neural Networks*, **Vol.9**,pp.509-521,1996
2) Martin Riedmiller and Heinrich Braun ," A direct adaptive method for faster backpropagation learning : the RPROP algorithm ." *Proceedings of the 1993 IEEE international conference on Neural Networks*, **Vol.1**,pp.586-591,1993
3) D.E.Rumelhart ,G.E.Hinton and R.J.Williams, "learning internal representations by error propagation", In *Parallel Distribution Processing* , D.E.Rumelhart and J.L.Mccleland, Eds.,**Vol.I**, Cambridge,MA:MIT Press,1986, pp.318-362
4) Russel Reed," Pruning algorithms—a survey",*IEEE Trans. Neural Networks*,**Vol.4**,pp. 740-747,1993
5) M. Moreira and E. Fiesler," neural netoworks with learning rate and momentum terms ",*IDIAP Technical Report*,No. 95-04,1995
6) S.Geman,E.Bienenstock and R. Doust ," Neural networks and the bias/variance dilemma",*Neural Computa.*,**Vol.4**,pp. 1-58,1992
7) Y. Le Cun ,J.S. Denker and S.A. Solla ," Optimal brain damage" ,*Advances in Neural Information Processing(2)* ,D.S.Toutetzky, Ed.(Denver 1989) pp. 598-605,1990
8) A.S.Weigend,D.E.Rumelhart and B.A.Huberman," Generalization by weight elimination with application to forcasting" ,*Advances in Neural Information Processing(3)* ,R.Lippmann , J.Moody and D.S.Toutetzky, Eds.1991 pp.875-882
9) P.M.Murphy and D.W.Aha, " UCI repository of machine learning databases",Department of information and computer science, University of California,Irvine(1992).
10) N.Murata, S.Yoshizawa and S.Amari," Network information criterion—determining the number of hidden units for an artificial neural network model",*IEEE Trans. Neural Networks*,**Vol.5**,pp. 860-872,1994
11) V.Vapnik(1995)," The nature of statistical learning theory", Springer.

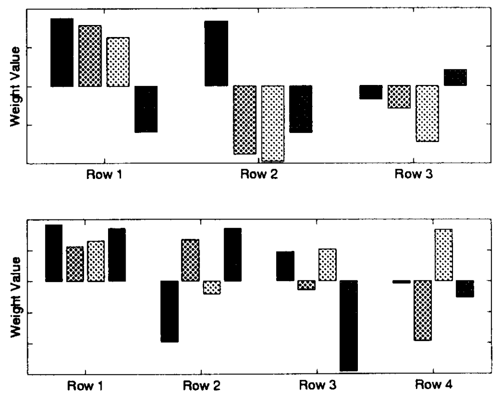

Fig.2 weight distribution by BP

Table 1: Generalization ability in the classification of testing Iris data with 4 hidden numbers

NO.	of Data	No.	of	Errors
Training	Testing	SLF	RPROP	Ours
9	141	90.2	90.5	91.4
15	135	49.1	60.4	50.2
21	129	40.1	52.2	40.4
30	120	21.2	41.4	20.5
60	90	15.9	31.11	16.6
90	60	4.0	4.5	4.3

Table 2: Distribution of weight matrix in the classification of IRIS data with 4 hidden numbers

NO.	of Data	No.	of	Errors
Training	Testing	SLF	RPROP	Ours
9	141	25.3	25.3	25.2
15	135	20.6	24.7	19.8
21	129	20.8	23.6	18.4
30	120	19.5	23.1	19.4
60	90	18.2	21.5	18.6
90	60	13.2	20.3	13.4

Table 3: Generalization ability in the classification of testing Iris data with 10 hidden numbers

NO.	of Data	No.	of	Errors
Training	Testing	SLF	RPROP	Ours
9	141	80.6	80.0	79.6
15	135	43.2	60.3	41.3
21	129	32.5	46.3	35.8
30	120	16.8	23.9	17.3
60	90	5.4	10.4	5.6
90	60	2.9	3.4	3.1

Table 4: Distribution of weight matrix in the classification of Iris data with 10 hidden numbers

NO.	of Data	No.	of	Errors
Training	Testing	SLF	RPROP	Ours
9	141	20	22.6	25.2
15	135	16.6	20.5	16.4
21	129	14.8	20.6	14.3
30	120	14.5	19.4	15.6
60	90	13.2	18.7	13.5
90	60	10.2	16.6	10.4

Table 5: Generalization ability in the classification of testing Iris data with 15 hidden numbers

NO.	of Data	No.	of	Errors
Training	Testing	SLF	RPROP	Ours
9	141	80.5	80.4	81.3
15	135	36.7	54.8	37.8
21	129	28.9	40.3	30.8
30	120	14.7	20.6	15.7
60	90	4.2	8.9	4.3
90	60	2.1	2.6	2.5

Table 6: Distribution of weight matrix in the classification of Iris data with 15 hidden numbers

NO.	of Data	No.	of	Errors
Training	Testing	SLF	RPROP	Ours
9	141	16.5	21.3	16.4
15	135	12.4	18.4	13.8
21	129	12.6	18.2	13.0
30	120	11.7	16.7	12.3
60	90	10.5	15.8	10.8
90	60	8.9	15.9	8.7