ELSEVIER

Neurocomputing 50 (2003) 237–247

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Partial least-squares algorithm for weights initialization of backpropagation network

Tzu-Chien Ryan Hsiao[a], Chii-Wann Lin[b], Huihua Kenny Chiang[a,*]

[a] *Institute of Biomedical Engineering, National Yang-Ming University, No. 155, Sec. 2, Li-Nung St., Taipei 112, Taiwan, ROC*

[b] *Institute of Biomedical Engineering, College of Medicine and Engineering, National Taiwan University, Taipei 100, Taiwan, ROC*

Received 8 March 2001; accepted 23 November 2001

Abstract

This paper proposes a hybrid scheme to set the weights initialization and the optimal number of hidden nodes of the backpropagation network (BPN) by applying the loading weights and factor numbers of the partial least-squares (PLS) algorithm. The joint PLS and BPN method (PLSBPN) starts with a small residual error, modifies the latent weight matrices, and obtains a near-global minimum in the calibration phase. Performances of the BPN, PLS, and PLSBPN were compared for the near infrared spectroscopic analysis of glucose concentrations in aqueous matrices. The results showed that the PLSBPN had the smallest root mean square error. The PLSBPN approach significantly solves some conventional problems of the BPN method by providing the good initial weights, reducing the calibration time, obtaining an optimal solution, and easily determining the number of hidden nodes.

Keywords: Weights initialization; Backpropagation network; Partial least-squares; Feedforward neural networks

1. Introduction

Feedforward neural networks have been studied extensively and applied to many areas, such as parallel distributed processing [17], forecasting time series [23], and spectroscopic signal measurement [11]. However, achieving efficient learning and better performance remain important issues to be addressed. The learning schemes of

* Corresponding author. Tel.: +886-2-282-67027; fax: +886-2-282-10847.
E-mail address: hkchiang@ym.edu.tw (H.K. Chiang).

0925-2312/03/$ - see front matter
PII: S0925-2312(01)00708-1

these networks can be classified into supervised and unsupervised learning algorithms according to the algorithm's modifying rules. The backpropagation network (BPN) is one of the most commonly used, supervised learning algorithms in the feedforward neural networks. Traditionally, the initial values of weights are determined randomly in the BPN and modified by the generalized delta rule. Although the BPN has been implemented in many applications, it still has some major drawbacks; namely, its convergence tends to be very slow, the optimal number of hidden nodes is difficult to determine, and usually it does not yield optimal solutions [1,6].

Many researchers have investigated these drawbacks and proposed various techniques to remedy them: for example, the adaptive learning algorithm for solving the long-training-time problem [3], the optimal learning rate and momentum for obtaining efficient BPN learning [26], the weight pruning process in enhanced performance for better pattern classification [25], the efficient mapping of the BPN into a network of workstations for better computational efficiency [22], the genetic algorithm for global optimization [19], and simulated annealing to escape the influence of local minima [12]. However, these approaches do not statistically explain the physical meaning of the weight contents, nor do they address the optimal number of hidden nodes.

The weights initialization technique is important for increasing the performance of the BPN and is a way to explain the weight contents [24]. Oja [15] and Sanger [18] have proposed principal component analysis to explain the weight contents of the three-layered feedforward neural networks. Its purpose is to derive the maximum variance weights only from the input patterns [14]. Therefore, it is suitable to apply principal component analysis for the weights initialization in the unsupervised feedforward neural networks, which require no information from output patterns. However, the BPN algorithm that uses the generalized delta rule for weight modification is based on cross-correlation between the input and output patterns. Although both the adaptive learning algorithm [3] and the variable step-size method [13] have been used to select the learning rate and have resulted in faster convergence, the BPN often converges to a poor local minimum.

The PLS algorithm is commonly employed in multivariate spectroscopic analysis [2,4]. It is derived from the ordinary least-squares regression method, which is a way to compute generalized matrix inverses and is a learning algorithm for pattern recognition. The PLS algorithm features the cross-training residual learning and consists of two steps: data compression (compressing input pattern to relevant compressed factors) and the linear regression of these scores on the output pattern of interest. Hence, the PLS can reduce the observed variables of the input matrix to a few underlying variables, equivalent to the number of hidden nodes in the BPN structure.

The structure of PLS can be treated as a simplified three-layered BPN with a linear activation function [8]. The loading weights P and Q represent the weights matrices between the input and the hidden nodes and between the hidden and the output nodes, respectively. The PLS can determine the weights initialization of the BPN in the specified chaos prediction [7]. Hence, a joint PLS and BPN method (PLSBPN) is proposed for the weights initialization of the BPN. The initial weights of PLSBPN are determined by the PLS loading weights, P and Q. These matrices are derived from the input and output patterns. Consequently, the PLSBPN algorithm can achieve conver-

gence in the early stage of the calibration phase and is also suitable for determining the optimal number of hidden nodes by means of the PLS cross-validation curve. To determine the efficiency of each, we compared the performances of the BPN, PLS, and PLSBPN methods for multivariate spectroscopic analysis of the near-infrared (NIR) glucose aqueous matrix.

2. Materials and methods

2.1. PLS architecture

The general schematic diagram of the PLS is shown in Fig. 1. The relations between the input and output matrices are

$$X = U\widehat{P}^{\mathrm{T}} + E = u_1\widehat{p}_1^{\mathrm{T}} + u_2 p_2^{\mathrm{T}} + \cdots + u_a\widehat{p}_a^{\mathrm{T}} + E, \tag{1}$$

$$Y = V\widehat{Q}^{\mathrm{T}} + F = v_1\widehat{q}_1 + v_2\widehat{q}_2 + \ldots + v_a\widehat{q}_a + F, \tag{2}$$

where a is the number of the PLS regression factors; the superscript T denotes the transposition of matrices; the matrices U and V are the latent variables for the input and output matrices X and Y, respectively; and $\widehat{P}$ and $\widehat{Q}$ are the PLS loading weights. The matrices E and F are the residuals of the matrices X and Y, respectively. In the general PLS, a model $F(u_a)$ is assumed to relate the latent variables u_i and v_i as

$$v_i = F(u_i). \tag{3}$$

The training procedure of PLS (Fig. 1) includes the following steps: ① The matrix Y is used as the temporal U. ② The loading weight $\hat{P}$ is calculated by using least-squares method and scaled vector to length 1. ③ The score U is estimated via matrices X and $\hat{P}$. ④ The other score V is calculated via the score U. ⑤ The loading weight $\hat{Q}$ is estimated by using the least-squares method. However, in this paper, the linear model $F(u_i) = u_i$ is used. Thus the matrices X and Y can be replaced by the matrices

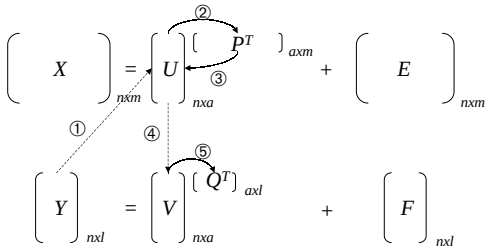


Fig. 1. Schematic diagram of the PLS. ① The matrix Y is used as the temporal U. ② The loading weight P is calculated by using least-squares method and scaled the vector to length 1. ③ The score U is estimated via matrices X and P. ④ The other score V is calculated via U ($V = U$ in this paper). ⑤ The loading weight Q is estimated by using the least-squares method.

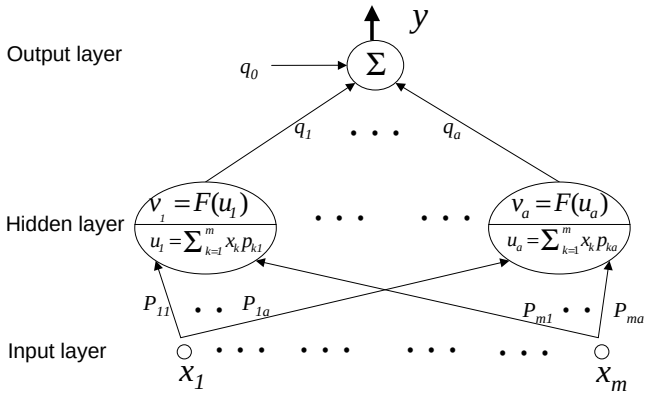


Fig. 2. The PLSBPN can be represented as a three-layered feedforward neural network. X: input, V: hidden, Y: output nodes, P and Q: weights.

$U(=V)$, $\widehat{P}$, and $\widehat{Q}$. Therefore, the relationship between the matrices X and Y can be represented by the loading weights $\widehat{P}$ and $\widehat{Q}$ and the regression factor U by making $\|E\|$ and $\|F\|$, respectively, as small as possible.

The PLS can be treated as a three-layered, supervised feedforward neural network (see Fig. 2). The matrices X, V, and Y represent the input, hidden, and output patterns, respectively. The major advantage of the PLS is that it can efficiently compress a large number of the input variables (input nodes) into a set of PLS regression factors (hidden nodes) that are more relevant to the output variables (output nodes). Therefore, the $\widehat{P}$ matrix denotes the weights between the input and the hidden nodes, and $\widehat{Q}$ matrix denotes the weights between the hidden and the output nodes, respectively.

2.2. PLSBPN architecture

The PLSBPN is a hybrid approach in which the initial weights and the number of hidden nodes are determined by the PLS. The training procedure of PLSBPN (Fig. 3) includes the following steps. ① The PLS determines the weights initialization and the number of hidden nodes. The loading weights $\widehat{P}$ and $\widehat{Q}$ are used as the initial weights for the BPN. ② The network processes and estimates the output values. In order to interpret the concept of PLSBPN, the linear activation function is used in this paper. ③ The cost function decides whether to finish the training process or to continue training. ④ The generalized delta rule modifies the weights matrices.

In step ③ , the most commonly used cost function is the sum of squared errors:

$$\mathrm{SSE} = \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2, \tag{4}$$

where y_i and $\widehat{y}_i$ represent the desired and estimated values, respectively. In general, if step ③ indicates that the weights matrices are not perfectly trained, the output values

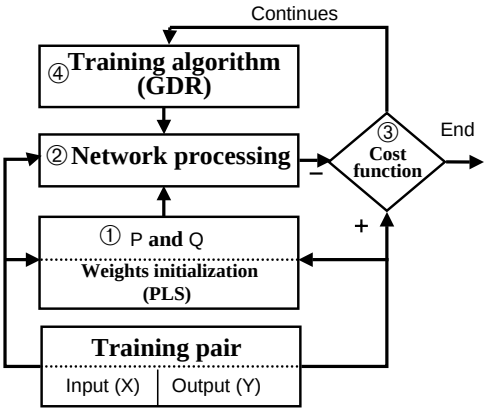


Fig. 3. The PLSBPN training procedure: ① PLS weights initialization, ② network processing, ③ cost function decision, and ④ training algorithm for weights modification. GDR = generalized delta rule.

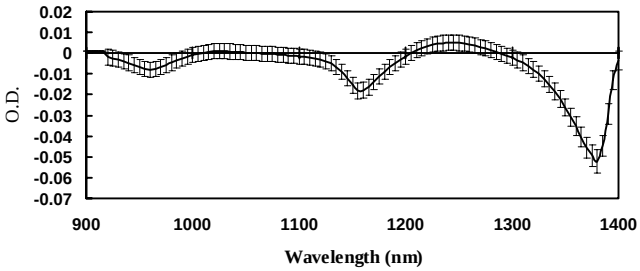


Fig. 4. The NIR absorption spectra of different glucose concentrations in deionized water after subtracting the absorption spectrum of the water.

will differ somewhat from the desired values. Hence, the contribution of each weights matrix is evaluated by the generalized delta rule.

2.3. NIR glucose aqueous matrices

Fig. 4 shows the averaged NIR absorption spectra of different glucose concentrations in deionized water after subtracting that of water. The NIR absorption spectra with 36 different glucose aqueous matrices (41–389 mg/dl) were used for the multivariate analysis. The NIR spectra were collected by a CDI/OS256L NIR spectrophotometer with an InGaAs diode array detector (Control Development, Inc., South Bend, IN, USA). The light source used for the NIR absorption measurement was a feedback-controlled, stable halogen lamp. The collected spectra span from 900 to 1400 nm with a 5-nm spectral resolution. For verification, the glucose concentrations were measured with an YSI1500 glucose analyzer (Yellow Spring Instruments Co., Inc., Yellow Spring, OH, USA).

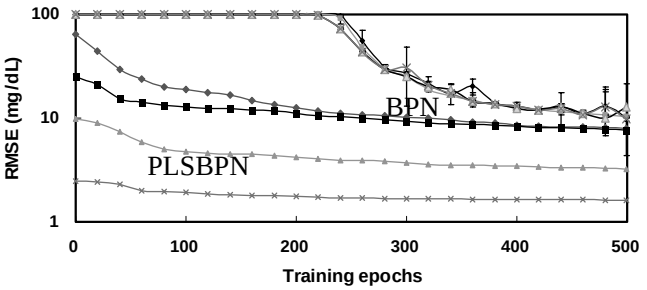


Fig. 5. Training epochs in the calibration phase of the BPN and PLSBPN using 1(♦), 5(■), 10(▲), and 15 (×) hidden nodes, respectively. Each BPN curve represents a 10-times-averaged calibration result, and the error bar represents one standard deviation.

2.4. Computational program

All three methods, PLS, BPN, and PLSBPN, were programmed on an Intel Pentium III-850 personal computer. All data analyses and implementations were performed with software written in LabVIEW (version 6i, National Instruments, Austin, TX, USA). Both the BPN and the PLSBPN have the same training data set, total training time of 500 epochs, and Jacob's delta–bar–delta algorithm [16]. The setting parameters of acceleration are 5×10^{-5} for initial learning rate, 1.4 for positive increased ratio, and 0.7 for negative decreased ratio. Because of the random initial weights, the BPN training procedure is repeated 10 times, and the averaged results are plotted with a standard deviation error bar.

3. Results

The performance of the BPN, PLS, and PLSBPN is calculated by means of the root mean square error value: $\mathrm{RMSE}=\sqrt{\sum_{i=1}^{n}(y_i-\widehat{y}_i)^2/n}$. The training times (epochs) in the calibration phase of BPN and PLSBPN, which used 1, 5, 10, and 15 hidden nodes, are plotted in Fig. 5. The results indicate that the PLSBPN performs much better than the BPN with the same number of hidden nodes. The initial RMSE value of the PLSBPN is much lower than that of the BPN. The better initial condition is realized by setting the initial weights according to the PLS loading weights, $\widehat{P}$ and $\widehat{Q}$, which are derived by searching the relevant weights between the input and output patterns. In addition, all RMSE values of the PLSBPN converge early in the calibration phase, whereas the BPN with random initial weights not only cannot guarantee to reach a convergence but also needs a much longer epoch to achieve the same convergence if convergence exists.

Fig. 6 shows the influence of the number of hidden nodes on the RMSE value in the calibration phase. Since the PLS adopts the residual learning algorithm, its error value monotonically diminishes with the increasing number of hidden nodes. By using

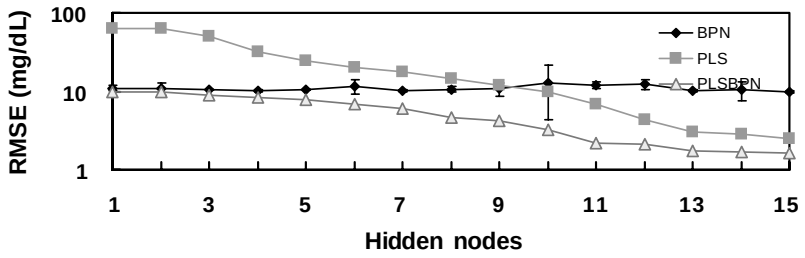


Fig. 6. The root mean square error (RMSE) of the BPN, PLS, and PLSBPN in the calibration phase at 500 training epochs.

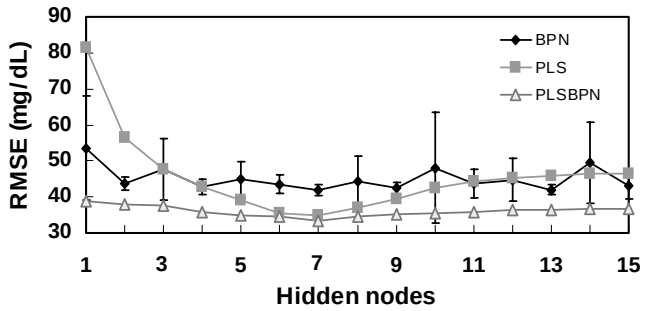


Fig. 7. The RMSE of the BPN, PLS, and PLSBPN in the cross-validation phase at 500 training epochs.

the PLS as the weights initialization method, PLSBPN has a trend similar to that of PLS. On the other hand, the BPN has large variation than PLSBPN because of the random initial weights of the BPN. To further differentiate between the weights influences of PLSBPN and BPN, the ratios of their RMSE values ($RMSE_{PLSBPN}/RMSE_{BPN}$) are calculated. These values indicate that the PLS weights initialization of the PLSBPN outperforms the conventional random weights initialization of the BPN. Better calibration results can be achieved with a greater number of hidden nodes. In this example, the best $RMSE_{PLSBPN}/RMSE_{BPN}$ value is 16.67% with 15 hidden nodes.

Fig. 7 illustrates the prediction capability of these methods in the cross-validation analysis. In general, the position of the lowest RMSE value indicates the choice of the optimal number of hidden nodes [21,20]. The choice of an insufficient number can lead to a higher RMSE value in the cross-validation analysis. On the other hand, the choice of more hidden nodes than needed leads to an overfit and a higher RMSE value too. In Fig. 7, the cross-validation curve of PLS obviously indicates that the optimal number of hidden nodes is 7. Because the PLS determines the initial weights in the PLSBPN, the cross-validation curve of PLSBPN shows the same optimal number (7) of hidden nodes.

4. Discussion

Weights initialization has been recognized as one of the most effective approaches for increasing the performance of the BPN. Some investigators have explored the influence of weights initialization for training enhancement. Ivanova [9], for example, regarded the weights initialization as a machine-learning problem, and his decision generator was an efficient process for setting the initial weights. Lehtokangas [10] focused on the weights initialization with the use of reference patterns. His experiment showed that the proposed method could outperform the conventional random initialization. Yam [24] developed a determining algorithm by means of Cauchy's inequality and a linear algebraic method. The results showed that the training epochs were only 3.03% of the conventional BPN. Even though these proposed weights initialization methods could perform better than the conventional BPN, they explain neither the weights content nor the effects of the number of hidden nodes. The PLS method, however, has been successfully used to explain the weights content of the BPN.

The present study demonstrated that the PLSBPN, with its small RMSE values and early convergence, is an efficient approach in speeding up the training epochs in the calibration phase. This superior performance of the PLSBPN is attributed from good initial weights and the optimal number of hidden nodes. Because of its computing efficiency and capability for statistical analysis, the PLSBPBN is suitable for online multivariate spectroscopic analysis (including the recalibration purpose).

The basic derivations of the principal component analysis and PLS are similar in linear algebra. However, the principal component analysis searches the maximum variance from the input patterns, and the PLS uses information from both input and output patterns. Hence, the principal component analysis has the effect of enhancing maximum variance within the input pattern and can be profitably applied to the weights initialization problems in the unsupervised feedforward neural networks. By contrast, the PLS obtains the latent variables from the input and output patterns. Therefore, it is suitable for the weights initialization problem in the supervised feedforward neural networks.

Another important issue in designing a BPN is how many nodes are needed in the hidden layer. Fujita [5] and Reed [16] indicated that the determination of the number of hidden nodes depends on many factors, including the number of input/output nodes and the number of training samples. Small numbers of hidden nodes cannot sufficiently present the weights between the input and output nodes. On the other hand, a larger-than-needed number of hidden nodes results in overfitting and a longer computational time. The PLS cross-validation curve is a workable method for determining the optimal number of regression factors. It represents the latent variables between the input and output patterns. Hence, it efficiently determines the needed number of hidden nodes in the three-layered BPN. Just as the optimal number of hidden nodes in the PLS can be determined by the cross-validation method, so too does the PLSBPN cross-validation curve show the optimal number of hidden nodes.

The PLSBPN has adopted two learning algorithms. One uses the PLS for determining the initial weights and the other uses the generalized delta rule for modifying the weights matrices. Both the calibration and the cross-validation prediction have shown that the PLSBPN algorithm has a shorter training epoch and a better RMSE value than

the BPN. On the basis of these findings, we conclude that the PLS loading weights are a very good method for weights initialization in the BPN. In future studies, we will apply these findings to investigation of the nonlinear relation of the BPN with a nonlinear activation function.

5. Conclusions

In this paper, we propose a novel PLSBPN method, which its weights initialization and number of hidden nodes are determined by the PLS. With its combination of the features of both BPN and PLS, the hybrid PLSBPN substantially improves the training performance and efficiently achieves an optimal solution. Furthermore, our study shows that the cross-validation curve can provide an optimal number of hidden nodes in the three-layered feedforward neural networks.

Acknowledgements

The authors would like to thank the anonymous reviewers for their useful comments and suggestions. Parts of this work were supported by the National Science Council, ROC under Contrast number NSC 89-2736-L-010-001.

References

[1] O. Baldi, K. Hornik, Neural networks and principal component analysis: learning from examples and local minima, Neural Networks 2 (1989) 53–58.
[2] P. Bhandare, Y. Mendelson, R.A. Peura, G. Janatsch, J.D. Kruse-Jarres, R. Marbach, H.M. Heise, Multivariate determination of glucose in whole blood using partial least-squares and artificial neural networks based on mid-infrared spectroscopy, Appl. Spectrosc. 47 (1993) 1214–1221.
[3] L.-H. Chen, S. Chang, An adaptive learning algorithm for principal component analysis, IEEE Trans. Neural Networks 6 (1995) 1225–1263.
[4] M. Fredic, N.K. Ivica, M.C. Glenn, R.G. Brent, Determination of glucose concentrations in an aqueous matrix from NIR spectra using optimal time-domain filtering and partial least-squares regression, IEEE Trans. Biomed. Eng. 44 (6) (1997) 475–485.
[5] O. Fujita, Statistical estimation of the number of hidden units for feedforward neural networks, Neural Networks 11 (1998) 851–859.
[6] M. Gori, A. Tesi, On the problem of local minima in backpropagation, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 76–85.
[7] T.C. Hsiao, C.W. Lin, H.K. Chiang, Partial least-squares learning regression for backpropagation network, 22nd Annual International Conference of the IEEE Engineering in Medical and Biology Society, Chicago, IL, July 2000 (CD-ROM).
[8] T.C. Hsiao, C.W. Lin, M.T. Tseng, H.K. Chiang, The implementation of partial least-squares with artificial neural network architecture, 20th Annual International Conference of the IEEE Engineering in Medical and Biology Society, Vol. 20, 1998, pp. 1341–1343.
[9] I. Ivanova, M. Kubat, Initialization of neural networks by means of decision trees, Knowledge-based Systems 8 (1995) 333–344.
[10] M. Lehtokangas, J. Saarinen, Weight initialization with reference patterns, Neurocomputing 20 (1998) 265–278.
[11] C.W. Lin, T.C. Hsiao, M.T. Tseng, H.K. Chiang, Artificial neural networks for spectroscopic signal measurement, in: M. Akay (Ed.), Nonlinear Biomedical Signal Processing, Vol. I: Fuzzy Logic, Neural Networks, and New Algorithms, IEEE Press Series on Biomedical Engineering, IEEE Press, Piscataway, NJ, 2000.

[12] G.D. Magoulas, M.N. Vrahatis, G.S. Androulakis, On the alleviation of the problem of local minima in backpropagation, Nonlinear Anal. Theory Methods Appl. 30 (1997) 4545–4550.

[13] G.D. Magoulas, M.N. Vrahatis, G.S. Androulakis, Effective backpropagation training with variable stepsize, Neural Networks 10 (1997) 69–82.

[14] H. Martens, T. Nas, Multivariate Calibration, Wiley, New York, 1996.

[15] E. Oja, A simplified neuron model as a principal component analyzer, J. Math. Biol. 15 (1982) 267–273.

[16] R.D. Reed, R.J. Marks, Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks, MIT Press, Cambridge, MA, 1999.

[17] D.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing, MIT Press, Cambridge, MA, 1986.

[18] T.D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, Neural Network 2 (1989) 459–473.

[19] R.S. Sexton, R.E. Dorsey, J.D. Johnson, Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation, Decision Support System 22 (1998) 171–185.

[20] R.D. Snee, Validation of regression models: methods and examples, Technometrics 19 (1976) 415–428.

[21] M. Stone, Corss-validatory choice and assessment of statistic prediction, J. R. Statist. Soc. B 36 (1974) 111–133.

[22] V. Sudhakar, C.S.R. Murthy, Efficient mapping of backpropagation algorithm onto a network of workstations, IEEE Trans. Systems Man Cybernet.—Part B: Cybernet. 28 (1998) 841–848.

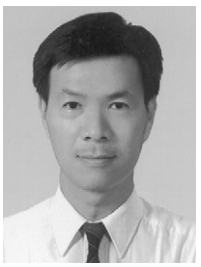
[23] V.R. Vemuri, R.D. Rogers, Artificial Neural Networks: Forecasting Time Series, IEEE Computer Society Press, Piscataway, NJ, 1994.

[24] Y.F. Yam, T.W.S. Chow, A weight initialization method for improving training speed in feedforward neural network, Neurocomputing 30 (2000) 219–232.

[25] Z. Yilbas, M.S.J. Hashmi, Simulation of weight pruning process in backpropagation neural network for pattern classification: a self-running threshold approach, Comput. Methods Appl. Mech. Eng. 166 (1998) 233–246.

[26] X.-H. Yu, G.-A. Chen, Efficient backpropagation learning using optimal learning rate and momentum, Neural Networks 10 (1997) 517–527.

Tzu-Chien Ryan Hsiao was born in Taoyuan Taiwan in 1971. He received the M.S. degree from the Department of Physics at the National Sun Yat-Sen University, Taiwan, in 1995. He is a doctoral candidate of the Institute of Biomedical Engineering at National Yang-Ming University. He is also the author of LabVIEW Series in Chinese, Gau-Lin Book Co. His research interests include neural networks, virtual instrument, and spectral analysis.



Huihua Kenny Chiang received the B.S. degree in electrical engineering from the National Tsing-Hua University, Taiwan, in 1982. He received the MS and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology, Atlanta, GA in 1987 and 1991, respectively. In 1992, he has been employed as a Research Scientist at the Georgia Tech Research Institute, GA. In 1993, he joined the Institute of Biomedical Engineering, the National Yang-Ming University, Taipei, Taiwan, as an Associate Professor, and now is Chairman and Professor of the Institute Biomedical Engineering (1999-present). His current research interests include medical ultrasound signal processing, noninvasive optical diagnostic techniques, and cardiac signal processing.

Chii-Wann Lin is currently an associate professor in the Institute of Biomedical Engineering at National Taiwan University, Taipei, Taiwan. He received the B.S. and MS degree in electrical engineering from the National Cheng-Kung University and biomedical engineering from the National Yang-Ming University, Taiwan, in 1984 and 1984, respectively. He received the Ph.D. degrees in biomedical engineering from the Case Western Reserve University, Cleveland, Ohio, in 1993.In 1993, he has been employed as a Research Associate at the Neurology Department, Case Western Reserve University, Ohio. In 1993, he joined the Center for Biomedical Engineering, College of Medicine, National Taiwan University, Taipei, Taiwan, as a Research Assistant Professor, and now is Associate Professor of the Institute Biomedical Engineering, College of Medicine and College of Engineering, National Taiwan University (1998-present). His current research interests include quantitative spectral measurement, medical instrumentation standardization, and Biochip.