# The Optimal Structure of Backpropagation Networks

SONGYOT SUREERATTANAN and HUYNH NGOC PHIEN
Computer Science and Information management
Asian Institute of Technology
P.O. Box 4, Klong Luang, Pathumthani 12120
THAILAND

*Abstract:* - A new algorithm obtained by using network measure such as Akaike information criterion (AIC) or Bayesian information criterion (BIC) is presented to systematically select the optimal structure, via the number of hidden nodes, of Backpropagation (BP) networks. Simulation results show that the algorithm performs satisfactory in all cases considered.

*Key-Words:* - Backpropagation networks, Optimal structure of Backpropagation networks, Akaike information criterion, Bayesian information criterion *IMACS/IEEE CSCC'99 Proceedings*, Pages:2251-2255

## 1 Introduction

Backpropagation (BP) algorithm is a classical method for learning multilayer feedforward (MLFF) networks. It is *a supervised learning technique* that, for a given network, compares the output from the network (model output) to the known output given by user (actual output) and then readjusts the weights in a backward direction. In BP networks, *the steepest descent or gradient descent method* is used to minimize the sum of squared errors (system error) between the actual output and the model output. Although it is widely and successfully used in many applications, the BP algorithm suffers from a number of shortcomings. One is the slow convergence rate, with which many iterations are required to train the network even for simple problems. Another shortcoming is due to the fact that there is no known method that provides the optimal structure of the network used for a given data set. The structure of the network seriously affects its performance of the model. As the number of nodes in the input and output layers are application dependent, the remaining problem is how to optimally choose the number of hidden nodes.

Hirose et al. [1] proposed an algorithm to find the optimal number of hidden nodes by changing the number of hidden nodes dynamically until a minimal number is found for which convergence (total mean squared error, MSE of less than 0.01) occurs. A straight MSE performance measure cannot be used to compare two different models directly because different numbers of parameters may be involved [2].

Instead of the MSE, Akaike information criterion (AIC) and Bayesian information criterion (BIC) can be utilized to select the best model from candidate models having different numbers of parameters. In BP networks, the number of parameters is generally the number of weights and biases. A new algorithm is proposed to systematically determine the optimal number of the hidden nodes by employing these criteria. So, the optimal structure of BP networks is obtained.

In our experiment, we present the results of our simulation studies that were intended to assess the performance of the algorithm. For this purpose, we employed daily streamflow data (rainfall-runoff) at three stations, namely Srinagarind (SRI), Khao Laem (KLM), and K32A in the Mae Klong River Basin located in the western part of Thailand [3] in the comparative simulations.

## 2 Backpropagation Networks

Backpropagation (BP) method was discovered independently by several researchers for different reasons such as Werbos [4], Parker [5], and le Cun [6]. However, credit is usually given to Rumelhart et al. [7] who developed this method into an applicable procedure, which has been widely used.

BP method is a supervised learning technique for learning associations between input and output patterns. This method can be applied to any MLFF networks with differentiable activation functions as shown in Fig.1. It is a generalization of the original two-layer perceptron (no hidden layer) introduced by Rosenblatt [8, 9], especially the version developed

by Widrow and Hoff [10]. Thus, it is also called *the generalized delta rule*. Like the delta rule, it is an optimization method based on steepest descent method that adjusts the weights to reduce the system error.

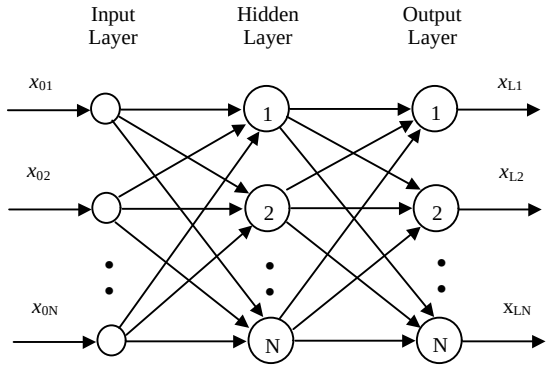


Fig.1 Backpropagation network architecture

Originally, the steepest descent method is used to train BP networks. It uses only the first derivatives of the error function. The error, *E*, for the network over all patterns is defined as (half) the sum of squared differences between the actual output and the model output in the output layer:

$$E = f(\boldsymbol{w}) = \sum_{p=1}^{M} E_p = \frac{1}{2}\sum_{p=1}^{M}\sum_{k=1}^{N_L}\left(o_{pk} - x_{pLk}\right)^2 \qquad (1)$$

where $o_{pk}$ and $x_{pLk}$ are the actual output and model output for the *k*th node in the output layer *L* and the *p*th training pattern, respectively, *M* is the number of data points, and $N_L$ is the number of nodes in the output layer.

The goal is to evaluate the weights in all layers of the network that minimize the system error. In steepest descent, the search direction at the *t*th iteration is the negative of the gradient:

$$\boldsymbol{s}^t = -\nabla f\left(\boldsymbol{w}^t\right) \qquad (2)$$

and the weight update is

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \Delta\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \eta\,\boldsymbol{s}^t = \boldsymbol{w}^t - \eta\,\nabla f\left(\boldsymbol{w}^t\right) \qquad (3)$$

where $\Delta\boldsymbol{w}^{t+1}$ is weight vector from $\boldsymbol{w}^t$ to $\boldsymbol{w}^{t+1}$, $\boldsymbol{s}^t$ is search direction of steepest descent, and $\lambda$ is step size.

To train a BP network, each input pattern is presented to the network and propagated forward layer by layer until the output of the network is calculated. Then, the model output is compared to the actual output and an error is determined. The error signals are used to readjust the weights layer by layer in a backward direction. This process is repeated for each training pattern until the system error converges to a minimum. Hence, the BP algorithm can be summarized as follows:

1. Initial all weights and biases to small random values.
2. Present a training pattern pair ($\boldsymbol{x}_0$, $\boldsymbol{o}$) where $\boldsymbol{x}_0$ is the input vector and $\boldsymbol{o}$ is the actual output vector.
3. Compute the calculated output $x_{jk}$ starting with the layer *j* from 1(the first hidden layer) and proceeding layer by layer toward the output layer *L* for every node *k*. In this case, the sigmoid function is selected as an activation function:

$$x_{jk} = f(y_{jk}) = f\left(\sum_{i=0}^{N_{j-1}} x_{j-1,i} w_{jki}\right) \qquad (4)$$

   where $y_{jk}$ is the summation output of the *k*th node in the *j*th layer and $N_j$ is the number of nodes in the *j*th layer.
4. Compute the error signals $e_{Lk}$ for the weights of the output layer and $e_{jk}$ for the weights of the hidden layers; and the weight change $\Delta w_{jki}$ starting with the layer *j* from the output layer *L* and backtracking layer by layer toward the input layer:

$$e_{Lk} = f'(y_{Lk})(o_k - x_{Lk}) = x_{Lk}(1 - x_{Lk})(o_k - x_{Lk}) \qquad (5)$$

$$e_{jk} = f'(y_{jk})\sum_{l=1}^{N_j} e_{j+1,l} w_{j+1,l,k} = x_{jk}(1 - x_{jk})\sum_{l=1}^{N_j} e_{j+1,l} w_{j+1,l,k} \qquad (6)$$

$$\Delta w_{jki}(t+1) = \eta\, e_{jk} x_{j-1,i} + \alpha \Delta w_{jki}(t) \qquad (7)$$

   where $\eta$ is the step size or learning rate constant, $\alpha$ is the momentum constant, and *t* denotes the iteration number.
5. Update the weights vector at (*t*+1)th iteration:

$$w_{jki}(t+1) = w_{jki}(t) + \Delta w_{jki}(t+1) \qquad (8)$$

6. Repeat steps 2-5 for all patterns until the system error has reached an acceptable error criterion.

## 3 Structure of Backpropagation Networks

Since BP training can be very costly, and the training cost increases as the network becomes more complex, the network should be kept as simple as

possible (as few layers and nodes as needed). Determining the number of hidden nodes is more complicated than that for the input and output nodes. The optimal number of hidden nodes is not known in advance. It is usually determined by *trial-and-error*. This approach starts with choosing an architecture of the network based on experience and tests the performance after each training phase. This process is continued as long as the performance increases and stopped once the performance begins to decrease.

Basically, network complexity measures are useful both to assess the relative contributions of different models and to decide when to terminate the network training. The performance measure should balance the complexity of the model with the number of training data and the reduction in the MSE [11].

There are two well-known network measures, namely Akaike information criterion (AIC) [12] and Bayesian information criterion (BIC) [13]. They are given as:

$$\text{AIC} = M \ln(\text{MSE}) + 2P \tag{9}$$

$$\text{BIC} = M \ln(\text{MSE}) + P \ln(M) \tag{10}$$

where $M$ is the number of data points used to train the network and $P$ is the number of parameters or the size of the model:

$$P = \sum_{i=0}^{L-1} N_{i+1}(N_i + 1) \tag{11}$$

Here $N_i$ is the number of nodes in layer $i$ and $L$ is the output layer. MSE is defined as follows:

$$\text{MSE} = \text{SE} / M \tag{12}$$

where SE is the sum of squared errors. Among the candidate models, the AIC or BIC criterion chooses the one corresponding to its minimum value.

It is noted that while the MSE is expected to progressively improve as more parameters are added to the model, the AIC and BIC penalize the model for having more parameters and therefore tend to result in smaller models. These two criteria can be used to assess the performance of the overall network, as they balance modelling error against network complexity. In Eqs. 9 and 10, the first term is a measure of fit and another term is a penalty term to prevent over fitting. As the BIC is more consistent [14], it is used in the following.

# 4 Proposed Algorithm

A new algorithm is proposed to systematically determine the optimal number of the hidden nodes using the procedure that gradually increases the network complexity and employs the BIC for terminating the training phase. The procedure starts with a small number of hidden nodes and trains the network until the system error is below an acceptable level. Then add a hidden node and retrain the network again. This process is repeated until the current BIC is greater than the previous one. The proposed algorithm can be summarized as follows:

1. Create an initial network with a tentative hidden node and randomize the weights.
2. Train the network using the chosen method i.e., BP algorithm until the system error has reached an acceptable error criterion. A simple stopping rule is introduced to indicate the convergence of the algorithm. It is based upon the relative error of the sum of squared errors (SE):

$$\left|\frac{\text{SE}(t+1) - \text{SE}(t)}{\text{SE}(t)}\right| \le \varepsilon_1 \tag{13}$$

   where $\varepsilon_1$ is a constant that indicates the acceptable level of the algorithm and SE($t$) denotes the value of SE at iteration $t$.
3. Check for terminating the training of the network. A termination criterion is suggested based on the relative BIC:

$$\left|\frac{\text{BIC}(k+1) - \text{BIC}(k)}{\text{BIC}(k)}\right| \le \varepsilon_2 \tag{14}$$

   where $\varepsilon_2$ is a constant that indicates the acceptable level for the structure of the network and $k$ denotes the loop number of the network. If the relative BIC is less than or equal to $\varepsilon_2$ or the current BIC is greater than the previous, go to step 4; otherwise add a hidden node and initialize the weights then go to step 2.
4. Reject the current network model and replace it by the previous one, then terminate the training phase.

# 5 Experimental Results

For forecasting daily streamflow in the Mae Klong River Basin located in the western part of Thailand, with forecasting lead time equal to one day, a simple network having 5 input nodes and one output node was used for the three stations considered, namely

Srinagarind (SRI), Khao Laem (KLM), and K32A [3]. An architecture of the 5-1-1 network consisting of 5 input nodes, 1 hidden node, and 1 output node was selected as the initial network. We employed the original BP algorithm as described in Section 2 for training the network. The acceptable level of the algorithm, $\varepsilon_1$, is set to 0.0001 and the acceptable level for the structure of the network, $\varepsilon_2$, is set to 0.01. As mentioned in Section 4, the algorithm is terminated when the relative BIC less than or equal to $\varepsilon_2$ or the current BIC is greater than the previous one. From these results, the algorithm is terminated when the 5-3-1 network is training. Thus, the 5-2-1 network is the best. In Tables 1-3, EI denotes the efficiency index, which is defined by Nash and Sutcliffee [15] as

$$\mathrm{EI} = 1 - \frac{\mathrm{SE}}{\mathrm{ST}} \tag{15}$$

$$\mathrm{SE} = \sum_{i=1}^{M} \left( y_i - \hat{y}_i \right)^2 \tag{16}$$

$$\mathrm{ST} = \sum_{i=1}^{M} \left( y_i - \bar{y} \right)^2 \tag{17}$$

$$\bar{y} = (1/M) \sum_{i=1}^{M} y_i \tag{18}$$

where SE = Sum of squared errors,
ST = Total variation,
$y_i$ = Actual output, i.e. observed discharge at time $i$,
$\hat{y}_i$ = Model output, i.e. forecast discharge at time $i$,
$\bar{y}$ = Mean value of the actual output,
$M$ = Number of data points.

For more information, we also considered the 5-5-1 network that has more hidden nodes. The learning curves of three discharge stations namely SRI, KLM, and K32A for various numbers of hidden nodes are shown in Figs.2-4, respectively. It is clear that the performance of the optimal network is the best in terms of both the minimum of system error and the computation time. Moreover, the optimal network can converge to the minimal point. Additional information can be obtained from Tables 1-3 that show the performance of the various architectures of the networks for SRI, KLM, and K32A, respectively.

Table 1 Comparison of the various architectures of the networks for SRI

| | 5-1-1 | 5-2-1 | 5-3-1 |
|---|---|---|---|
| Epoch | 3170 | 4644 | 6053 |
| SE | 0.85 | 0.30 | 0.30 |
| EI | 0.98 | 0.99 | 0.99 |
| BIC | -17123.94 | -19322.81 | -19256.75 |
| Time (seconds) | 824 | 1364 | 1958 |

Table 2 Comparison of the various architectures of the networks for KLM

| | 5-1-1 | 5-2-1 | 5-3-1 |
|---|---|---|---|
| Epoch | 2476 | 2360 | 2901 |
| SE | 1.73 | 1.30 | 1.34 |
| EI | 0.92 | 0.94 | 0.94 |
| BIC | -15556.93 | -16125.14 | -16011 |
| Time (seconds) | 644 | 693 | 953 |

Table 3 Comparison of the various architectures of the networks for K32A

| | 5-1-1 | 5-2-1 | 5-3-1 |
|---|---|---|---|
| Epoch | 45 | 1549 | 2254 |
| SE | 4.86 | 1.41 | 1.31 |
| EI | 0 | 0.71 | 0.73 |
| BIC | -5862.53 | -7166.56 | -7194.67 |
| Time (seconds) | 6 | 227 | 367 |

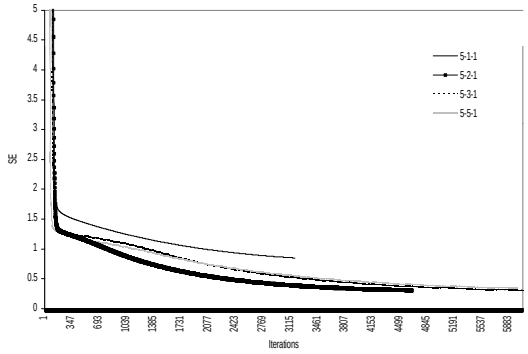


Fig.2 Learning curve of SRI for various numbers of hidden nodes

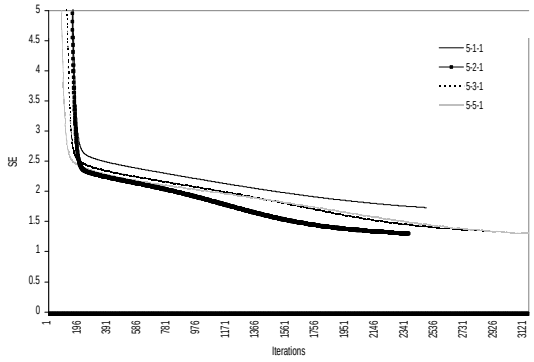


Fig.3 Learning curve of KLM for various numbers of hidden nodes

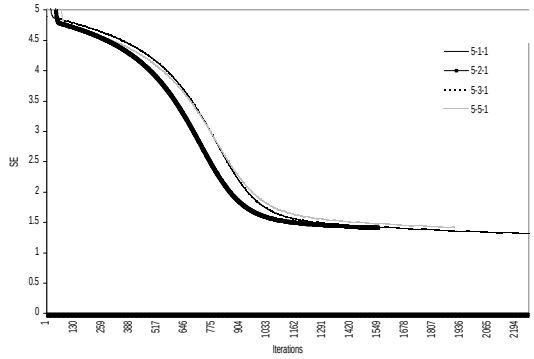


Fig.4 Learning curve of K32A for various numbers of hidden nodes

## 6 Conclusion

The BIC can be used to choose the best model from candidate models having different numbers of parameters. By using this network measure, a new algorithm was proposed to systematically determine the optimal structure of BP networks. Experimental results show that the proposed algorithm can perform well in all cases considered. Chosen by the proposed algorithm, the optimal structure of the network not only can minimize system error and computation time, but also can converge to the minimal point.

*References:*

[1] Y. Hirose, K. Yamahsita, and S. Hijiya, Back-Propagation Algorithm which Varies the Number of Hidden Units, *Neural Networks*, Vol.4, 1991, pp.61-66.

[2] M. Brown and C. Harris, *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall, UK., 1994, pp.326-327.

[3] S. Sureerattanan and H.N. Phien, Back-propagation Networks for Daily Streamflow Forecasting, *Water Resources Journal*, No.195, 1997, pp.1-7.

[4] P.J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. Thesis, Harvard University, 1974.

[5] D.B. Parker, *Learning-logic*, Technical Report TR-47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA., 1985.

[6] Y. le Cun, *Modeles Connexionnistes de l'apprentissage*, Ph.D. Thesis, University of Pierre and Marie Curie, Paris, 1987.

[7] 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: Vol.1: Foundations*, D.E. Rumelhert and J.L. McClelland (eds.), MIT Press, Cambridge, Massachusetts, 1986, pp.318-362.

[8] F. Rosenblatt, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psycho. Rev.*, Vol.65, No.6, 1958, pp.386-408.

[9] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington, D.C., 1961.

[10] B. Widrow and M.E. Hoff, Adaptive Switching Circuits, *IRE WESCON Conv. Record*, Part 4, 1960, pp.96-104.

[11] M. Pottmann and D.E. Seborg, Identification of Nonlinear Processes using Reciprocal Multiquadratic Functions, *J. Proc. Cont.*, Vol.2, No.4, 1992, pp.189-203.

[12] H. Akaike, A New look at the statistical model identification, *IEEE Trans. Autom. Control*, AC-19, 1974, pp.716-723.

[13] J. Rissanen, Modeling by Short Data Description, *Automation*, Vol.14, 1978, pp.465-471.

[14]R.L. Kashyap, *Inconsistency of the AIC rule for estimating the order of autoregressive models*, Technical Report, Dep. of Electr. Eng., Purdue Univ., Lafayette, 1980.

[15]J.E. Nash and J.V. Sutcliffee, River Flow Forecasting through Conceptual Models, *Journal of Hydrology*, Vol.10, 1979, pp.282-290.