12-2003

Predicting nonlinear network traffic using fuzzy neural network

Zhaoxia WANG
Singapore Management University, zxwang@smu.edu.sg

Tingzhu HAO

Zengqiang CHEN

Zhuzhi YUAN

ICICS-PCM 2003
15-18 December 2003
Singapore

Predicting nonlinear network traffic using Fuzzy neural network

Zhaoxia Wang[a]; Tingzhu Hao[b]; Zengqiang Chen[a]; Zhuzhi Yuan[a] *Senior Member IEEE*
a. Department of Automation, Nankai University, Tianjin 300071, P. R. of China
b. Department of Opto-Electronic Information and Electronic Engineering, Tianjin University of Technology, Tianjin 300191, P.R.of china

Abstract—Network traffic is a complex and nonlinear process, which is significantly affected by immeasurable parameters and variables. This paper addresses the use of the five-layer fuzzy neural network (FNN) for predicting the nonlinear network traffic. The structure of this system is introduced in detail. Through training the FNN using back-propagation algorithm with inertial terms the traffic series can be well predicted by this FNN system. We analyze the performance of the FNN in terms of prediction ability as compared with solely neural network. The simulation demonstrates that the proposed FNN is superior to the solely neural network systems. In addition, FNN with different fuzzy reasoning approaches is discussed.

Index Terms—Fuzzy neural network, Traffic prediction, Back-propagation algorithms, Inertial terms

1. Introduction

It has been proved that a neural network system with appropriate structures is able to approximate an arbitrary nonlinear function [1]. As neural networks (NN) has flexible learning capabilities that make it possible to develop nonlinear models using only input-output data, NN has been widely studied in the traffic control or traffic prediction of the computer network [2-4]. Although NN is capable of learning complex nonlinear relationships, it is difficult to fine modeling the logical process of human reasoning. On the other hand, fuzzy systems are universal approximators and are capable of approximating any real continuous function [5]. Fuzzy systems store rules and estimate the functions from linguistic input to linguistic output [6]. However, fuzzy systems lack the ability of learning and adapting. Thus, a combined fuzzy-neural network approach offers interesting potential for nonlinear modeling. A combination of the fuzzy system and the neural network is called the fuzzy neural network systems (FNN), which utilize both the linguistic, human-like reasoning of fuzzy systems and the powerful computing ability of neural network. They can avoid some drawbacks of solely fuzzy or neural network systems. It had been proven that "fuzzy-neural network can approximate any nonlinear function to any desired accuracy because of the universal approximation theorem" [7].

Recently, there have been considerable interests in the application of FNN. A number of several successful FNN systems were reported in the literature [8]. Some works have been carried out on FNN for nonlinear time series prediction or other problems of prediction [9-11]. A FNN with a general parameter learning algorithm and heuristic model structure determination had been proposed for modeling nonlinear time-series [9]. And an alternative FNN architecture had been proposed to predict a chaotic time series. Such work has demonstrated the superior prediction capabilities of a fuzzy neural network as compared with the conventional neural network approach [10]. The paper [11] employed a five-layer FNN to predict the quality of chemical components of the finished sinter mineral and obtained very good performance.

In this paper, the authors propose a five-layer FNN to predict the traffic of video and voice sources. The FNN uses Mamdani's inference which includes the min-max operator and is introduced in detail. The traffic is characterized by a continuous-state discrete-time autoregressive (AR) Markov process. The improved BP algorithm [13] is adopted to train the FNN. The architecture provides a comparable degree of accuracy to the solely neural network. The abilities of this architecture to learn and generalize have been demonstrated by its application in the traffic prediction.

2. The Structure of Proposed FNN

The knowledge representation in fuzzy models was developed by Mamdani [12]. The knowledge is presented in these models as follows.

0-7803-8185-8/03/$17.00 © 2003 IEEE

R_i : if x_1 is A_1^i, x_2 is A_2^i, …, and x_n is A_n^i, then y is B^i (1)

Where R_i (i=1, 2, …, c) denotes the ith fuzzy rule, and x_j (j=1, 2, …, n) is the input variables and y is the output variable of the fuzzy rule R_i, A_1^i, A_2^i, …, A_n^i are fuzzy sets of input space and B^i is a fuzzy set of output space. The corresponding membership functions of these fuzzy sets are, $\mu_{A_1}^i(x_1), \mu_{A2}^i(x_2), \cdots, \mu_{An}^i(x_n)$ and $\mu_B^i(y)$ respectively, which are bell-shaped trapezoidal, or triangular, et al., and usually associated with linguistic terms.

The proposed FNN is a five-layer fuzzy neural network and the structure of the FNN is shown in Fig. 1. Let N_k denotes the number of nodes of each layer and M_i denotes the terms number of input linguistic variables.

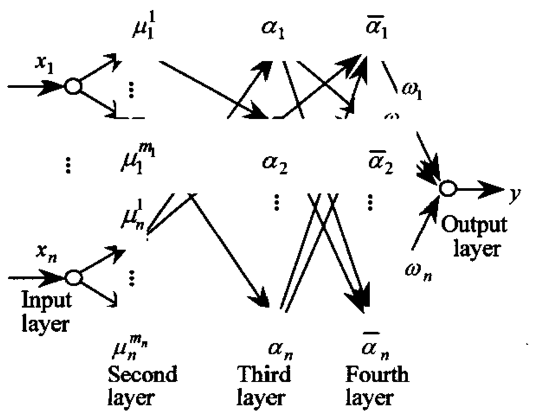

Fig. 1 The schematic structure of the FNN

The functions of each layer and each node are described as follows.

Layer 1: The INPUT-FN's [9] is used in this layer. Each INPUT-FN node in this layer is called input node and corresponds to one input linguistic variable. The number of nodes in this layer is N_1. Each node transmits the input to the next layer directly, i.e., $y_i^{(1)} = x_i^{(1)}$ ($i = 1,2,...,N_1$).

Layer 2: Nodes in this layer are called input term nodes and each represents a term of an input linguistic variable. In other word, the membership value belongs to a fuzzy set is calculated in this layer. The membership function is:

$$\mu_i^j = \mu_{A_i^j}^j(x_i) \qquad i = 1,2,\cdots,n; j = 1,2,\cdots,M_i \tag{2}$$

where n is the number of input and M_i is the terms number of input variable x_i. In this paper, the Gaussian function is chosen as the node function. And it is assumed that each input linguistic variable has the same number of term numbers. μ_i^j can be defined as:

$$\mu_i^j = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}} \tag{3}$$

where c_{ij} and σ_{ij} are the center and width of the Gaussian membership. The number of nodes in this layer is:

$$N_2 = \sum_{i=1}^{n} M_i = N_1 \cdot M_i \tag{4}$$

And the connection weight between the input layer and the second layer is [11]:

$$w^{(2)}(i,j) = \begin{cases} 1, & i = (km+1,...,km+m) \cap (j = k+1) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $k = 0,1,...,N_2 - 1$, $i = 1,2,...,N_2$, and $j = 1,2,...,N_1$. The input and output of each node of the second layer are

$$x_i^{(2)} = w^{(2)}(i,j) x_j^{(1)} \tag{6}$$

$$y_i^{(2)} = e^{-\frac{(x_i^{(2)} - c_i)^2}{\sigma_i^2}} \tag{7}$$

where $i = 1,2,...,N_2$, and $j = 1,2,...,N_1$.

Layer3: Each node in this layer represents one fuzzy logic rule and performs precondition matching of a rule. The min operator is chosen as follows.

$$\alpha_j = \min\{\mu_1^{i_1}, \mu_2^{i_2}, \cdots, \mu_n^{i_n}\} \tag{8}$$

where $i_1 \in \{1, 2, …, m_1\}$, $i_2 \in \{1, 2, …, m_2\}$,…, $i_n \in \{1, 2, …, m_n\}$, j=1, 2, …, m; $m = \prod_{i=1}^{n} m_i$, $m_i \equiv M_i$, and $n = N_2$, so the total nodes of this layer is:

$$N_3 = \prod_{i=1}^{n} m_i = \prod_{i=1}^{N_1} M_i = M_i^{N_1} \tag{9}$$

And the connection weight between layer 2 and layer 3 is:

$$w^{(3)}(i,j) = \begin{cases} 1 & \left\{ \begin{array}{c} kM_i^{(m-1)} + 1 \le i \le (k+1)M_i^{(m-1)} \\ \cap kN_1 + 1 \le j \le (k+1)N_1 \end{array} \right\} \\ 0 & \text{otherwise;} \end{cases} \tag{10}$$

where $m = N_1$, $k = 0,1,...,M_i - 1$, $i = 1,2,...,N_3$, and $j = 1,2,...,N_2$.

Let vector $p = [p_1 \quad p_2 \quad p_3]$, $q = [q_1 \quad q_2 \quad q_3]^T$, and $p\Theta q = [p_1 q_1 \quad p_2 q_2 \quad p_3 q_3]$[11]. The input and output of each node of the third layer are obtained by the equations (11) and (12).

$$\vec{x}_i^{(3)} = \vec{w}^{(3)}(i,:)\vec{y}^{(2)} \tag{11}$$

$$y_i^{(3)} = \min\{x_i^{(3)}(1), x_i^{(3)}(2), ..., x_i^{(3)}(N_2)\} \tag{12}$$

where $\vec{y}^{(2)}$ is a vector which represents the outputs of the nodes in the second layer, $\vec{x}_i^{(3)}$ is a vector which represents the inputs of the *i*th node in the third layer. And $y_i^{(3)}$ is a scalar which represents the outputs of the *i*th node in the third layer.

Layer 4: The nodes in this layer are called output term nodes. The number of nodes in this layer is the same as that of the third layer. These nodes realize following normalized calculation.

$$x_j^{(4)} = \frac{y_j^{(3)}}{\sum_{i=1}^{N_3} y_i^{(3)}} \qquad j = 1,2,\cdots,N_4 \tag{13}$$

The output of each node of layer 4 is:

$$\overline{\alpha} = y_j^{(4)} = x_j^{(4)} \quad j = 1,2,\cdots,N_4 \tag{14}$$

Layer 5 is output layer. The node is a neuron. The sigmoid function is chosen as the activation function of the neuron for this traffic model. In this paper, there is only one node in layer 5. The input and the output are defined as:

$$x_i^{(5)} = \sum_{j=1}^{m} \omega_{ij} y_j^{(4)} = \sum_{j=1}^{m} \omega_j \overline{\alpha}_j \tag{15}$$

$$y_{sigmoid}^{(5)} = \frac{1}{1 + e^{-x_i^{(5)}}} \tag{16}$$

This architecture can be readily implemented on the MATLAB neural network toolbox and trained using improved back-propagation algorithms.

3. Predicting problem presentation

The traffic characterization is presented in [2] and [14]. The scalar traffic series is denoted by $y(t)$, and the FNN model used will be

$$y(t) = F_{FNN}\{y(t-1), y(t-2), ..., y(t-m)\} + \varepsilon(t) \tag{17}$$

where F_{NN} is an approximate function of the nonlinear function f and $\varepsilon(i)$ is the prediction error. The main difficulty in implementing prediction model is that function f is actually unknown. And the only information available is the set of observables: $y(1), y(2), y(3), ..., y(n)$, where n is the total length of the traffic series. It is the goal of prediction scheme to approximate this function. FNN is known to be the good function approximator [5-7]. Their real attraction lies in their ability to learn by the examples of NN and fuzzy concept, fuzzy judgment and fuzzy reasoning of fuzzy system. However, in order to obtain a network that produces a desirable output, Fuzzy neural network must typically be trained upon the available examples many times.

During training, a FNN is presented with several input/output pairs just as training the NN, and is expected to learn the functional relationship between inputs and outputs of the simulation model. Therefore, the trained FNN can predict the output for inputs other than the ones presented during training.

4. Simulation Results and Discusses

The traffic series used in this work are generated from video and voice sources. A continuous-state discrete-time autoregressive (AR) Markov process is proposed in [14] and used by [2] to characterize this traffic. Let $\lambda(n)$ represent the bit rate of a single source during the *n*th frame. A first order Markov process AR (1) is generated by the recursive relation:

$$\lambda(n+1) = a\lambda(n) + bw(n) \tag{18}$$

where $a = 0.8781$, $b = 0.1108$ and $w(n)$ is a sequence of independent Gaussian white noise with mean = 0.572 and variance = 1.

A summary of the implementation results obtained is presented in Table 1. All the simulations use 500 points from the traffic series as training data and use a further 1500 points as test data. And FNN network is trained for 3000 epochs. To compare the performance of the FNN, similar sized neural networks are also implemented. Five NN networks are used. One is with 2 input nodes and 6 nodes in hidden layer, such a neural network is denoted as (2: 6: 1); the second is with 2 input nodes and 9 nodes in hidden layer (2: 9: 1); the third is with 2 input nodes and 12 nodes in hidden layer (2: 12: 1); the fourth is with 2 input nodes and 15 nodes in hidden layer (2: 15: 1) and the fifth is five-layer forward networks with 2 input nodes and 6 nodes, 9 nodes, 9 nodes in the next three hidden layers, respectively, and one node in output layer (2: 6: 9: 9: 1).

Table 1

Implementation results of predicting the traffic

The structure	RMSE (training)	RMSE (testing)
FNN	0.016	0.012
NN(2: 6: 1)	0.076	0.070
NN(2: 9: 1)	0.067	0.065
NN(2: 12: 1)	0.101	0.099
NN (2: 15: 1)	0.127	0.122

The results are presented in terms of the accuracy of the prediction using the root-mean-square error (RMSE). The values for RMSE (training) are obtained using the training data and similarly the values of RMSE (testing) are obtained using the test data.

The results presented in table 1 illustrate the benefits of the FNN approach compared to the NN as the error metric is lower. Table 1 illustrates that the prediction accuracy improves using 9 neurons in input layer rather than 6 nodes in hidden layer. Another result of the implementations is that the prediction accuracy degrades when moving from 12 nodes of hidden layer to 15 nodes of hidden layer. And it is difficult to predict this traffic series when we use the (2-6-9-9-1) five-layer NN which are trained for 3000epchs. This can be explained by the larger size of the network with higher number of hidden layer which introduces extra parameters and hence increases the training difficulties.

The accuracy of the predictions is illustrated in Fig. 2. The simulation results, which are obtained by the FNN, demonstrate that the predicted and actual points of the traffic series are almost heavy to match. The results establish that the FNN can approximate the traffic series to the satisfied degree of accuracy.

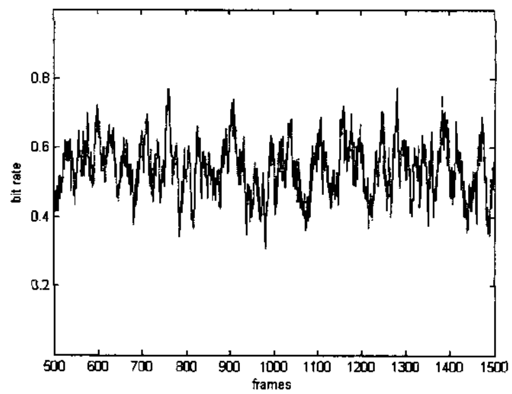

Fig. 2 Prediction of the traffic series as compared with the actual data (solid line)
------ prediction
— actual data

Table 2

Implementation results of predicting the traffic

The structure of the FNN	RMSE (training)	RMSE (testing)
FNN with 2 fuzzy sets in each input variable	0.032	0.028
FNN with 3 fuzzy sets in each input variable	0.016	0.012
FNN with 4 fuzzy sets in each input variable	0.017	0.012
FNN with 5 fuzzy sets in each input variable	0.024	0.021

For further analysis, the proposed FNN are extended to account for different numbers of fuzzy sets of the input space. In general, the prediction capability of a fuzzy set system improves when increasing the number of fuzzy set of the input variables. However, the results presented in Table 2 illustrate that the prediction capability degrades when increasing the number of fuzzy sets from 3 to 5. The results indicate that "increasing the number of fuzzy seta not only increases the network dimensions and training times but may also over-parameterise the problem which may degrade its performance" [10]. The results also illustrate that the FNN with 3 fuzzy sets maintains a similar degree of accuracy compared to the FNN with 4 fuzzy sets of the input space. Thus 3 fuzzy sets in each input domain are sufficient for this problem.

5. Conclusions and Future Work

A five-layer FNN has been used for predicting the traffic of video and voice sources. The simulation results demonstrate that FNN is capable of predicting this traffic series to any desired degree of accuracy and the FNN is superior to the solely neural network systems. The architecture not only provides a comparable degree of accuracy to the solely neural network, but offers the additional advantage of the reduced dimensions. Traffic prediction has become one of the most important problems for internet management. Although the problem of predicting time series using FNN has been widely considered as an imposing research object, it has not made enough influences on the traffic forecasting. Further research on real traffic prediction using FNN is required, and the structure and learning algorithms of the FNN can be improved.

References

[1] K. Hotnik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," Neural Networks, Vol.2, No.5, pp.359-366, 1989.

[2] Z. Fan and P. Mar, "Access flow control scheme for ATM networks using neural network based traffic prediction," IEE Proc.-Commun., Vol. 144, No. 5, pp.295-300, Oct. 1997.

[3] A. A. Tarraf, I. W. Habib, and T. N. Saadawi, "Intelligent traffic control for ATM Broadband Networks," IEEE Communications Magazine, Vol.33, No.10, pp76-85, Oct. 1995.

[4] D. P. A. Greenwood and R. A. Carrasco, "Neural networks for the adaptive control of disruptive nonlinear network traffic," IEE Proc.-Commun., Vol.147, No.5, pp.285-291, Oct. 2000.

[5] L. X. Wang, "Fuzzy systems are universal approximations," Proc. IEEE Fuzzy Syst. Conf., 8-12. pp. 1163-1170, Mar. 1992.

[6] H. K. Kwan and Y. L. Cai, "A fuzzy neural network and its application to pattern recognition," IEEE Transactions on Fuzzy Systems, Vol.2, No.3, pp.185-193, Aug. 1994.

[7] Chi-Hsu Wang, Han-Leih Liu, and Tsung-Chih Lin, "Direct Adaptive Fuzzy-Neural Control With State Observer And Supervisory Controller For Unknown Nonlinear Dynamical Systems," IEEE Transactions on Fuzzy Systems, Vol.10, No. 1, pp.39-49, Feb. 2002.

[8] Y.-H. Kuo, C.-I, Kao, and J.-J. Chen, "A fuzzy neural network model and its hardware implementation", IEEE Transactions on Fuzzy Systems, Vol.1, No.3, pp.171 -183, Aug. 1993.

[9] D. F. Akhmetov, Y. Dote, and S. J. Ovaska, "Fuzzy neural network with general parameter adaptation for modeling of nonlinear time-series," IEEE Transactions on Neural Networks, Vol.12, No.1, pp.148-152, Jan. 2001.

[10] L. P. Maguire, B. Roche, T. M. McGinnity, and L. J. McDaid, "Predicting a chaotic time series using a fuzzy neural network," Information Sciences, Vol. 112, No. 1-4, pp.125-136, 1998.

[11] J. E. Meng, J. Liao, and J. Y. Lin, "Fuzzy neural networks-based quality prediction system for sintering process," IEEE Transactions on Fuzzy Systems, Vol. 8, No.3, pp.314-324, Jun. 2000.

[12] E. Mamdani, "Advances in the linguistic synthesis of fuzzy controllers," Int. J. Man-Manchine Studies, Vol. 8, pp.669-678, 1976.

[13] K. A. de Oliveira, Á. Vannucci and E. C. da Silva, "Using artificial neural networks to forecast chaotic time series," Physica A: Statistical Mechanics and its Applications, Vol. 284, No. 1-4, pp. 393-404, Sep. 2000.

[14] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet video communications," IEEE Transactions on Communications, Vol.36, No.7, pp. 834-844, Jul. 1988.