Journal of ELECTRICAL ENGINEERING, VOL. 54, NO. 9-10, 2003, 255–259

ARTIFICIAL INTELLIGENCE MODELLING OF STOCHASTIC PROCESSES IN DIGITAL COMMUNICATION NETWORKS

Dimitar Radev * — **Svetla Radeva** **

The paper presents results from a number of investigations into the problems of implementation of Intelligent methods in prediction and simulation of the ATM traffic, based on time series and state models. The prognoses based on neuro-fuzzy model and Learning Vector Quantization (LVQ) is suggested. The implementation for the for stochastic and long range dependence source models is shown.

K e y w o r d s: digital communications, queuing networks, quality of service (QoS), artificial intelligence methods, learning vector quantization (LVQ)

1 INTRODUCTION

A promising method in integration of technologies and services in digital communication networks is the simulation modeling of sources, traffic and their management. The simulation modelling is an abstraction of real interaction between sources, which leads to simplification and limited performance of behavior of elements in real communication network with help of stochastic processes. The traffic, generated on a various application is based on study of behavior of certain classes of probability processes like time series analysis, long range dependence stochastic processes, chaotic time series, discrete-time Markov state models, queuing theory [1,8].

The major issue at the data transfer from a various sources (voice, audio, video and other) in packed-switched telecommunication networks is the prediction of the Quality of Service (QoS), which include evaluation of the maximum cell transfer delay, peak-to-peak cell delay variation, and consecutive cell loss ratio, overflow probability [2].

A new trend in investigation of the quality of service in height speed digital communication networks is the involving of simulation models based on artificial intelligence methods. These models give a possibility to use adaptive and self-learning properties of neural networks with possibilities of building modeling algorithms using fuzzy logic rules. In this way the computation complexity (especially on cell level), the big capacity of the networks, and the influence of accidentally input can be analyzed and estimated with the help of iteration prognoses of the values of time series distribution [3].

In this paper is shown several implementations of neural networks and hybrid intelligent systems for analyses and estimation of non-stationary stochastic probabilistic processes in digital communication networks.

2 PROBABILITY DENSITY FUNCTION WITH LEARNING VECTOR QUANTIZATION

At the most of stochastic models is used the probability density function of time series. With the help of probability density function can be made prognoses of the arrival traffic flow, the overflow and cell loss in queuing networks, MPEG video traffic modelling etc. [5]. For this purpose the main parameter of the stochastic process are determined as input values of a self learning neural network with the help of which is made vector quantization. This quantization is known as Learning Vector Quantization (LVQ) and is used for 1 or 2 dimensional input time series. A basic problem at LVQ is how to learn the neural network in such a manner to provide an effective analyses and estimation of received results and to reduce the prediction errors of stochastic traffic processes. At vector quantization the target classes are determined in advance and according to them is making the learning of neural network.

The input information can be treated as a fluid flow, which is given in two layered neural networks with first competitive and a second linear layer. The principal of vector quantization with 2 input time series

$$\{\lambda_1\}, \{\lambda_2\}$$

is shown in Fig. 1. The input flow values represent $n \times 2$ space (1), which consists of 2 size moving register B_1 and B_2.

$$\boldsymbol{I}(n \times 2) = (\{\lambda_1\}, \{\lambda_2\}) = \boldsymbol{I}_k = (i_1, i_2, ..., i_n) \quad (1)$$

In competitive layer is module of vector quantization VQ, where the input random variables $X_1^{(1)}, \ldots, X_n^{(1)}$, $X_1^{(2)}, \ldots, X_n^{(2)}$ are distributed into M classes $(S_1, .., S_M)$.

* Department of Communication Technique and Technologies, University of Rousse, 8 Studentska Str., 7017 Rousse, Bulgaria, E-mail: dradev@abv.bg ** Department of Computer-Aided Engineering, University of Architecture, Civil Engineering and Geodesy, 1st Hrysto Smirnensky Blvd., 1046 Sofia, Bulgaria, E-mail: svetla_fce@abv.bg

First version of this paper was presented at SympoTIC ´03, 26-28 October 2003, Bratislava, Slovakia

ISSN 1335-3632

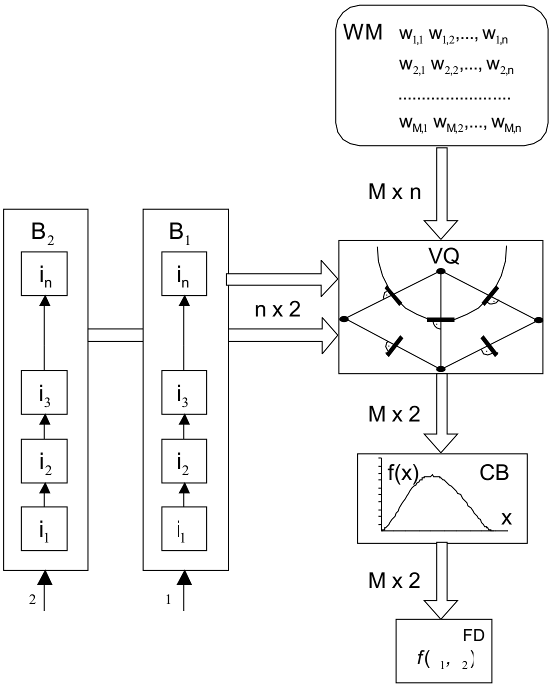

Fig. 1. Structure of 2D-LVQ models

As a criteria is used the longitude d_i, where $i = 1, ..., M$ is the index of the target class. The cell, which has the less value, is a "winner" and her output value is equal to 1. At the output of the rest cells we have as a result 0. For the k-th cell of the input space, the weight matrix WM creates the weight coefficients $w_{i,k}$ with the help of which is calculated the distance d_i as is shown in (2).

$$d_i = \|\boldsymbol{w}_{i,k} - \boldsymbol{I}_k\| = \\ = \sqrt{(w_{1i,k} - i_{1,k})^2 + ... + (w_{ni,k} - i_{n,k})^2} \qquad (2)$$

Thus we can present the probability function for the input space according to (3).

$$F(x_1^{(1)}, ..., x_n^{(1)}, x_1^{(2)}, ...x_n^{(2)}) = P\{X_1^{(1)} \le x_1^{(1)}, ..., X_n^{(1)} \\ \le x_n^{(1)}, X_1^{(2)} \le x_1^{(2)}, ..., X_n^{(2)} \le x_n^{(2)}\} \qquad (3)$$

At the linear layer is made adjusting of the boundaries of target classes in module CB (classes boundaries). In module density function FD is determining the approximated probability density function of the input vector space (4).

$$f(x_1^{(1)}, ..., x_n^{(1)}, x_1^{(2)}, ..., x_n^{(2)}) = \\ \frac{\partial^{n\times 2} F\left(x_1^{(1)}, ..., x_n^{(1)}, x_1^{(2)}, ..., x_n^{(2)}\right)}{\partial x_1^{(1)}, ..., x_n^{(1)}, x_1^{(2)}, ..., x_n^{(2)}} \qquad (4)$$

2.1 Adjusting of Target Classes

The proper definition of target classes is basic for the approximation of probability density function. The task is to determine M target classes, which boundaries are unknown in advance. The training algorithm is divided into two phases. During the first is determined classes with equally width and the number of target values for each class [7]. As is shown in Fig 2 left, at the most of cases this isn't lead to the optimal description of probability density function. That's why on the second phase is made adjusting of the boundaries of target classes. As a criteria for adjusting is taken the same number of target values in the neighbor classes $i-1$, i, $i+1$, as is shown in Fig. 2 - right. Automatically are moved the widths w_i and the longitudes d_i, of the neighbor classes in such a manner, that the face of their product is always the same (5).

$$f_i = d_i\, w_i \qquad i \in \{1 \dots M\} \qquad (5)$$

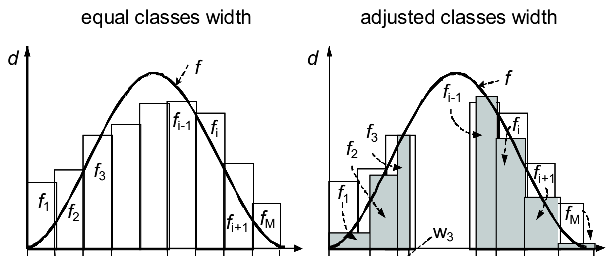

Fig. 2. Adjusting of classes boundaries

When on the input of neural model is given two input vectors, they are dividing into M target classes with equally width. For determination their probability density, first is determining the width of classes on horizontal axis and afterwards is determining the width of classes on vertical axis in such a manner to receive rectangle zones, which describe the boundaries of classes.

2.2 QoS Estimation of Queuing Networks

The queuing theory studies systems in which customers randomly arrive at a service station in order to be served, since there may be other customers ahead of them, they may need wait in a buffer or queue. Queuing models are characterized by the probability distribution of the flows: the time between arrivals, the time needed to serve a customer, size of the buffer space, queuing policy, etc. Their QoS estimation calculates transient overflow and cell losses probabilities. Here we will show that the standard description of these probabilities with the help of discrete-time Markov chain theory can be replaced with vector quantization method. Here are presented examples for tandem Jackson networks without and with feedback, in which the buffer contents n_1 and n_2 are presented as two-dimensional space.

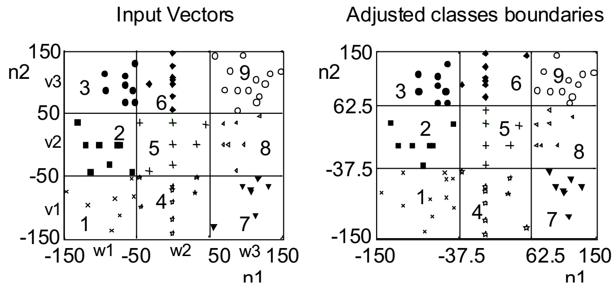

Fig. 3. 2D input vector with symmetric adjusting

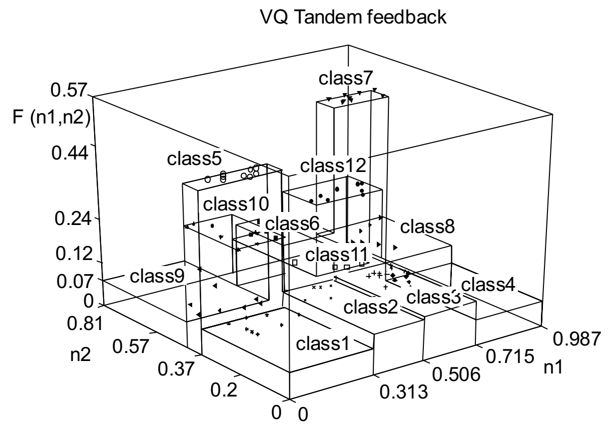

Fig. 4. 2D input vector with asymmetric adjusting

In Fig. 3 is shown a technique for symmetric adjusting of the boundaries. The 2D input flow (2x80) is created from arrivals and services of the queues n_1 and n_2. With the help of MATLAB the input time series were trained into 9 classes with equally widths, which corresponds to the number of states in discrete-time Markov model. For each class was received irregular number of target values. On the second step received probability density was adapted for the three widths of the input vectors via symmetric moving of boundaries in vertical and horizontal direction ($w_1 = v_1, w_2 = v_2, w_3 = v_3$). As a result was received the same number of target values in each class. This technique gives good results for small number of target classes.

For determining of probability density function of Jackson network with feedback were used 12 target classes, which corresponds to the four states of 2D Markov chain model. The two input time series (2x100) describe the arrivals and services of the queues n_1 and n_2. The probability density function $F(n_1, n_2)$ for each state is presented as a separate class, which widths in horizontal and vertical direction were adjusted asymmetrically (Fig.4). The width of the classes in horizontal direction is constant, for example: for classes 1, 5 and 9 the width is $w_j = 0.313$. The width of the classes in vertical direction is completely different, for example: for class 5 - $w_j = 0.06$; for class 6 - $w_j = 0.286$; for class 7 - $w_j = 0.065$; for class 8 - $w_j = 0.29$. In this way the simulation technique gives possibility for a good distribution of target values in each class, even for a large number of values in time series.

3 MODELING OF TIME SERIES WITH LONG RANGE DEPENDENCE

LVQ networks can be used successful for modeling of LRD time series [6]. A typical example about this, are video traffic models, based on MPEG statistical analysis of the registrated empirical frame size trace. Traffic analyses and simulation are realized with the help of GOP size sequence and GOP-by-GOP correlation, which represents the periodical structure of MPEG video traffic with three frame types (I, P and B-). As basic models are used histogram, simple Markov chain and scene oriented models. The values of the sequence of frame GOP patterns was presented as time series $\{x_t\}$,$(t \in 1, 2, \dots, n)$.

The histogram model is a simple application of the modeling in which is not taking into account the correlation structure of video data, because this model can't model the GOP-by-GOP correlation. For the histogram model was defined the number of targets M, which is the same as the histogram intervals and has size $y_i = S_j$ and width ΔS_j for the intervals of each target class $j \in (1, 2, \dots, M)$

$$S_j = \frac{\sum_1^k x_t}{k}, \quad \Delta S_j = \frac{(\max x_t - \min x_t)}{M} \tag{6}$$

At simple Markov chain model the time series $\{x_t\}$ was transformed into discrete states $\{y_t\}$ according to (7), where the number of states M, is the same as the number of target classes, and the size of the model $y_i = S_j$ for each state were determined as follows

$$S_j = \frac{\sum_1^k x_t \cdot y_t}{\sum_1^k y_t}, \quad \text{with} \quad y_t = \frac{x_t - \min x_t}{\max x_t - \min x_t} M \tag{7}$$

here $y_t = 1$ for $\min x_t$. The scene-oriented model is a first-order Markov chain model with a redefined set of states. The intention of this redefinition is to facilitate the modeling of scene changes and to achieve an improvement of the auto-correlation modeling properties of the Markov chain with a moderate increase in the number of states. For separated scenes, y_i is defined according to (7). In this way is reflected the variation between discrete states of time series $\{x_t\}$ for each scene. To determine of each scene the following variation-based algorithm is suggested:

1. Determining the scene boundaries based on the coefficient of variation for a sequence of consecutive values.

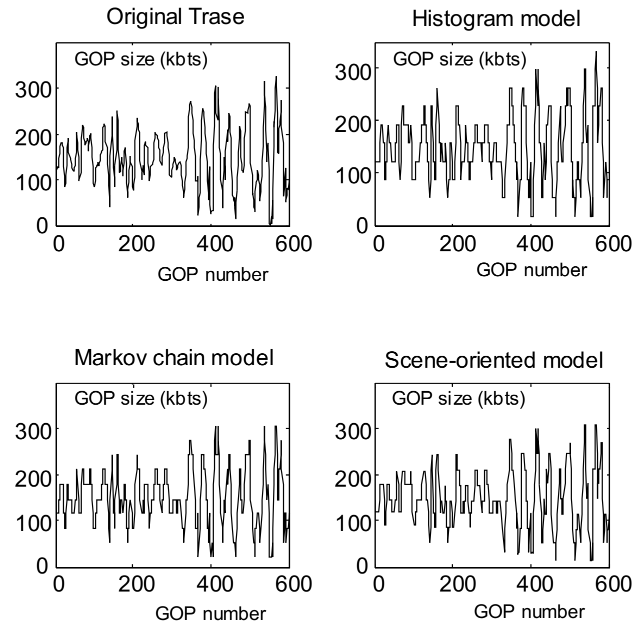

Fig. 5. Original trace, histogram-, simple Markov chain- and scene-oriented models

We add values to current scene until its weighted coefficient of variation is changing more than a preset value. The last added value is defined as the beginning of a new scene. If n_A denote the current number of them value and n_{sc} denote the current scene number we set $n_A = 1$ and $n_{sc} = 1$.The current left scene boundary $b_{left}n_{(sc)} = 1$.

2. Increment n_A by 1. Compute the coefficient of variation cv_{new} of $\{x_{b_{left}(n_{SC})}, \dots, x_{n_A}\}$.
3. Increment n_A by 1. Set $cv_{old} = cv_{new}$.
4. If $|cv_{new} - cv_{old}||(n_{sc} - b_{left} + 1) > \varepsilon$, where ε is a threshold value, then set the right scene boundary $b_{right}(n_{cs}) = n_A - 1$, and the left scene boundary of the new scene $b_{left}(n_{sc} + 1) = n_A$. Increment n_{sc} by 1 and go to step (2).

The accuracy of scene-oriented model depends on the proper determining of the threshold values between separated scenes. The accuracy of received results shows that it is impossible to work with the same threshold value ε for different scenes.

In Fig 5 are shown models for generating of time series of 600 GOP size, where for comparison, were used equal number of histogram intervals and Markov chain states $M = 10$. It is necessary to say that the histogram model don't give the dynamic structure of GOP patterns, because the target value for each class is time independent. It would be better to describe the dynamic video traffic with scene-oriented model, which reflect GOP-by-GOP correlation of the model with simple Markov chain. The correlation between studied GOP patterns gives possibility to determine four scene and ten classes of the models. Suggested method gives a possibility for automatic modeling of GOP size video traffic, when the neural network is trained with enough number of epochs.

4 NEURO–FUZZY PREDICTION OF CHAOTIC TIME SERIES

An approach for chaotic time series prediction is suggested, which is a combination of the two most popular intelligent methods: adaptive learning of perceptrons and fuzzy logic rules for modeling [4]. The algorithm of neuro-fuzzy prediction consists of the following basic steps:

- Creating of D-dimensional input vector for Adaptive Network based Fuzzy Inference System (ANFIS);
- Training of Fuzzy Inference System (FIS) membership functions;
- Determining of ANFIS Prediction Errors;
- Receiving of prognoses time series.

The task is on the base of known past values t of a time series $x(t)$ to determine the next $t+P$ future values, where P is ever less than t. The prediction is based on created of ANFIS for D time series samples, where the next sample is after Δ units in time (8) to a predicted future value $x(t + P)$.

$$L(t) = x\left[t - (D-1)\,\Delta\right], \dots, x(t - \Delta), x(t) \Rightarrow L'(t) = x(t + P) \tag{8}$$

As an example was made prognoses of video traffic for $D = 4$, $\Delta = P = 5$. For this prognoses were created 4-dimensional FIS input vector (9).

$$L(t) = [x(t - 15)x(t - 10)x(t - 5)x(t)] \Rightarrow L'(t) = x(t + 5) \tag{9}$$

The Neuro-fuzzy model gives the parameters of a FIS membership functions, which are adjusted according to the error back propagation algorithm combined with least squared method. The input time series consists of 800 values, which are divided into two parts: the first 400 - for training and the rest for checking and comparison of the model. The objective function and constraint conditions are described with two generalized bell membership functions, which were constructed on the base of 4 inputs and 16 fuzzy rules and 104 fitting parameters, presented in Fig. 6

The prediction error depends very much of the number of training epochs. When the training was made with 10 epochs, the ANFIS prediction error is significant, as is shown in Fig. 7. For better performance was applied extensive training. For criteria of satisfactory results, was used root mean squared error. During the process of investigation of error curve was received minimal RMSE after about 600 training epochs

When the number of FIS parameters is small enough compared with the input time series, the predicted traffic

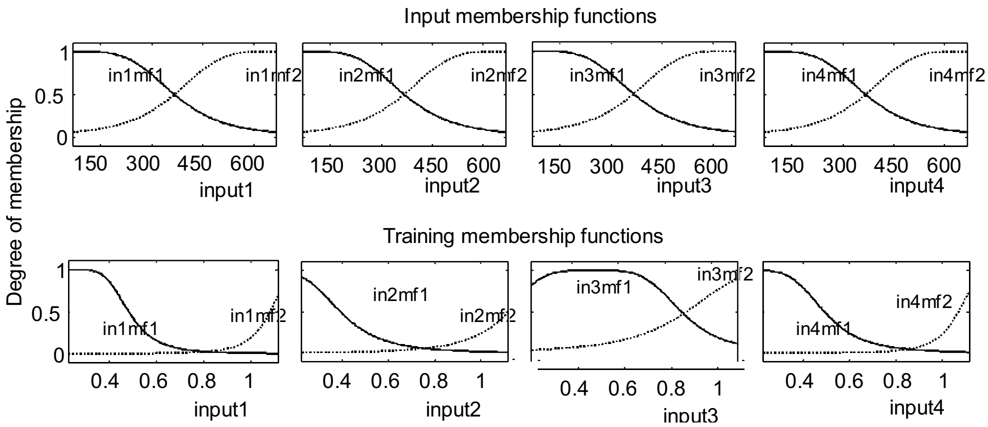

Fig. 6. Training of FIS membership functions

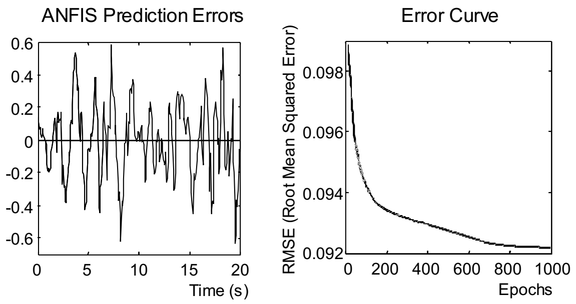

Fig. 7. Prediction errors

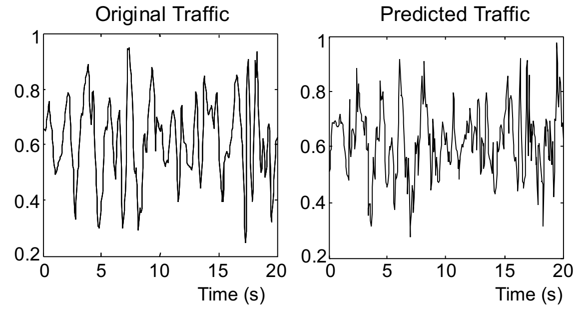

Fig. 8. Neuro-Fuzzy predicted traffic

is very closed to the original (Fig. 8). The trained in this way ANFIS can be used successfully for prognoses of the time series with 5 to 10 time less size of input data sets, compared with the size of data sets used for data learning.

5 CONSLUSIONS

Intelligent methods for prognoses and analysis of different types time series were developed. They can be used for simulation of digital communication networks for:

- Discretetime Markov state models;
- LRD stochastic processes;
- Chaotic processes.

Concrete application of suggested methods with adaptive neuro-fuzzy inference systems (ANFIS) and neural network systems with learning vector quantization (LVQ) are presented.

Acknowledgements

This work is a part of the International NATO research project PST.CLG.979333.

References

[1] BEDZHEV, B. A Method for Increasing the Reliability of Phase Radioranging Measurements: Proceedings of the VI-th Conference on Weapon Systems, Brno, Czech Republic, 1999,208-215.

[2] HAMILTON, J. Time Series Analysis: Princeton University Press, 1994.

[3] KOHONEN, T.: Self-Organizing Maps, Second Edition, Berlin: SpringerVerlag, 1997.

[4] RADEV, D.: An Approach for Prognoses of the Traffic Behavior in ATM Networks, Proceedings on Automation and Informatics'2002 **1** (2002), Sofia,17-21.

[5] RADEV, D.—RADEVA, S.: Vector Quantization for Simulation Modeling of Queuing Systems, Proceeding of the TELSIKS'2003, Nis, Serbia and Montenegro, 2003 **2**, 689-692.

[6] RADEV, D.—RADEVA, S.: Intelligent Methods for Simulation in ATM Networks, Proceedings of the SympoTIC'03, Bratislava, Slovak Republic, 2003, 90-93.

[7] RADEV, D.—RADEVA, S.—SAVOV, B.: Determining of the Boundaries Between Classes for Vector Quantization at Stochastic Non-stationary Processes, Proceeding of the VSU'2003, Vol. I, Sofia, 2003 33-38.

[8] SENNOTT, L.: Stochastic Dynamic Programming and the Control of Queueing Systems, John Wiley & Sons, 1999.

Received 10 July 2003

Dimitar Radev is Associate Professor at the department of Communication Technique and Technologies, University of Rousse, Bulgaria. Received the PhD degree in 1997. Scientific interests: queuing theory, simulation modelling in communication networks, rare event simulation, artificial intelligence methods. Author of more than 50 publications.

Svetla Radeva is Associate Professor at the department of Computer Aided Engineering, at the University of Architecture, Civil Engineering and Geodesy, Sofia, Bulgaria. Received PhD degree in 1987. Scientific interests: artificial intelligence methods, simulation modelling and prognoses of stochastic processes. Author of more than 50 publications and 2 books.