Video Traffic Prediction Using Neural Networks

Miloš Oravec, Miroslav Petráš, Filip Pilka

Department of Telecommunications, Faculty of Electrical Engineering and
Information Technology, Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
milos.oravec@stuba.sk, mirops@gmail.com, pilka_filip@nextra.sk

Abstract: In this paper, we consider video stream prediction for application in services like video-on-demand, videoconferencing, video broadcasting, etc. The aim is to predict the video stream for an efficient bandwidth allocation of the video signal. Efficient prediction of traffic generated by multimedia sources is an important part of traffic and congestion control procedures at the network edges. As a tool for the prediction, we use neural networks – multilayer perceptron (MLP), radial basis function networks (RBF networks) and backpropagation through time (BPTT) neural networks. At first, we briefly introduce theoretical background of neural networks, the prediction methods and the difference between them. We propose also video time-series processing using moving averages. Simulation results for each type of neural network together with final comparisons are presented. For comparison purposes, also conventional (non-neural) prediction is included. The purpose of our work is to construct suitable neural networks for variable bit rate video prediction and evaluate them. We use video traces from [1].

Keywords: data prediction, video traffic, neural network, multilayer perceptron, radial basis function network, backpropagation through time

1   Prediction of Video Traffic

The role of dynamic allocation of bandwidth is to allocate resources for variable-bit rate (VBR) video streams while capturing the bursty character of video traffic. By using prediction schemes, it is possible to increase utilization of network resources and to fulfill QoS (quality of service) requirements [2].

Multimedia traffic, especially MPEG video traffic became dominant component of network traffic. It is due to excessive use of services like video-on-demand (VoD), videoconferencing, broadcast and streaming video. Periodic correlation structure, complex bit rate distribution and noisy streams are some characteristics of MPEG video traffic [3]. Some traffic and congestion control procedures must be used, among which connection admission control (CAC), usage parameter control (UPC), traffic shaping, congestion indication, priority control, packet discarding are examples of the most important procedures.

The design of a bandwidth allocation scheme for VBR video is very difficult [4]. This is due to bursty character of such traffic, while at the same time VBR video requires strongest QoS characteristics such as delay, loss and jitter.

Neural networks are generally considered to be one of the most effective tools for prediction. Due to their analogy with biological neural networks (human brain), they seems to be suitable to solve prediction related tasks. For prediction, herein we use feedforward networks – multilayer perceptron (MLP) and radial basis functions (RBF) network, and recurrent backpropagation-through-time (BPTT) network.

2   Neural Networks

2.1   Neuron, Neural Network and Learning

The origin of artificial neural networks (ANN) [5] was inspired by the biological nervous system. The main inspiration was the human brain and the way it processes information. The human brain consists of very large number of elements (neurons), which are massively interconnected.

The basic element of each ANN is the neuron. The basic scheme of the neuron is shown in Fig. 1. Connecting such elements in various ways leads to different architectures of neural networks.
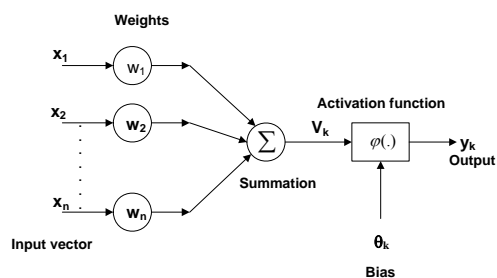

Figure 1

Basic model of neuron

The ANN learns by example. Learning (in the terminology of neural networks) is the process by which the weights are adapted. This process is represented by a learning algorithm. Many learning algorithms exist [5], [6], they differ in the way they adjust the weights of particular neurons. In this paper, we use the supervised learning paradigm [5], [6].

2.2   Multilayer Perceptron

The multilayer perceptron (MLP) [5], [6] is probably the best-known and frequently used neural network. It consists of one input and one output layer and it can contain one or more hidden layers of neurons. For this type of networks, the sigmoidal activation functions (including most popular logistic function) are mainly used.

MLP is trained by the backpropagation algorithm [5], [6]. The error signal of neuron j is defined as the difference between its desired and actual output:

$$e_j(n) = d_j(n) - y_j(n) \tag{1}$$

From error signal, the local gradient needed for the backpropagation algorithm can be computed:

$$\delta_j(n) = e_j(n)\varphi_j'(v_j(n)), \tag{2}$$

where $v_j(n)$ is the inner activity of the neuron.

Then the weight adaptation is done as follows:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \tag{3}$$

where $\Delta w_{ji}(n)$ is the weight adjustment in time n, η is the learning rate, $\delta_j(n)$ is the local gradient and $y_i(n)$ is the input signal of neuron j.

In general, we can write

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \tag{4}$$

If the neuron is the output neuron of the network, then we can use the presented algorithm to compute the weights adaptation. But if it is a hidden neuron, its desired output is not known and this is why the error signal for the hidden neuron has to be calculated recursively from the error signals of the neurons directly connected to the neuron (from output layer).

2.3   RBF Networks

When we look at the design of the neural networks from the perspective of approximation in multidimensional space, then learning is equivalent to finding such a plane in the multidimensional space which best approximates the training data. Neurons in the hidden layer represent a set of functions which represents the basis functions for the transformation of input vectors to the space of hidden neurons. These functions are called the radial basis functions (RBF) [7], [8], [5], [6].

The interpolation scheme using RBFs can be represented as follows

$$f(\mathbf{x}) = \sum_{j=1}^{N} w_j \phi\left(\left\|\mathbf{x} - \mathbf{c}_j\right\|\right) \tag{5}$$

Function $f(x)$ is the interpolation function which uses N radial basis functions ϕ_i, where $\phi_i : R^p \to R, i = 1, \dots, N$ and $\phi_i = \phi\left(\left\|\mathbf{x} - \mathbf{c}_i\right\|\right)$, $\phi : R^+ \to R$, $\mathbf{x} \in R^p$, $\left\|\cdot\right\|$ is norm on R^p (often Euclidian), $\mathbf{c}_i \in R^p$ are centers of RBFs, $\mathbf{w}_j$ are coefficients for linear combination of RBFs which we want to find.

Since $f(\mathbf{x}_i) = d_i, \forall i = 1, \dots, N$, we get the equation (5) in matrix representation

$$\begin{bmatrix} \phi_{11} & \cdots & \phi_{1N} \\ \vdots & \ddots & \vdots \\ \phi_{N1} & \cdots & \phi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix}, \tag{6}$$

where the elements of the matrix are

$$\phi_{ij} = \phi\left(\left\|\mathbf{x}_i - \mathbf{c}_j\right\|\right), \; i, j = 1, \dots, N. \tag{7}$$

When ϕ is a regular matrix we can find one exact solution. Many functions guarantee the regularity of the matrix. The most common is the Gaussian function

$$\phi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{8}$$

where σ is the width of the Gaussian RBF.

Presented scheme can be easily extended to mapping $F : R^p \to R^q$, F and $\boldsymbol{d}$ are then in the form $(f_1, \cdots, f_q)$, $(d_1, \dots, d_q)$, respectively.

The fact, that the number of RBFs is the same as the number of data points that are to be interpolated is a main disadvantage of interpolation scheme. Typically, there is a smaller number of RBFs compared to the number of given data points. Then we speak about approximation scheme, the matrix ϕ is no more square and its inverse matrix does not exist. Solution can be found by least-squares optimalization method, by pseudoinverse matrix or by RBF neural network with hidden neurons representing radial basis functions.

The training process of RBF network then consists of three steps. The first step is to find the centers of the basis functions. The second step adjusts additional parameters of RBFs (if any). The third step serves for output weights computation. More information about the training can be found in [7] and [8].

2.4   BPTT Networks

The backpropagation through time (BPTT) neural network [5], [9], [10], [11] belongs to a class of recurrent neural networks. Such networks contain feedback connections. Their training algorithms often compute the gradient of an error measure in weight space. BPTT learning algorithm is based on unfolding the temporal operation of the network into a multilayer feedforward network, where one layer is added at every time step. In this manner, the network is converted from a feedback system to purely feedforward system. The gradients of weights for a recurrent network are approximated using a feedforward network containing a fixed number of layers. More details about the BPTT networks and training algorithms can be found in [9], [10], [11].

2.5   Optimal Linear Prediction

One of the methods to predict the future samples of a time series is the autocorrelation method of autoregressive (AR) modeling. The idea of this method is to find the best coeficients of a prediction filter [12].

If we assume as in [13] that the video transmition rate sequence for linear prediction is $\{x(n)\}$, then the estimated (predicted) series can be expessed as follows:

$$\hat{x}(n+1) = -\sum_{i=1}^{m} a(i).x(n+1-i) \tag{9}$$

Coeficients $a(i)$ are the coeficients of the prediction filter and m is the order of AR model. The coeficients $a(i)$ are computed by the following equations:

$$X.a = b \tag{10}$$

where

$$X = \begin{bmatrix} x(1) & 0 & \cdots & 0 \\ x(2) & x(1) & & 0 \\ \vdots & x(2) & & x(1) \\ x(m) & \vdots & & (2) \\ \vdots & x(m) & & \vdots \\ 0 & \cdots & 0 & x(m) \end{bmatrix}, \; a = \begin{bmatrix} 1 \\ a(2) \\ \vdots \\ a(p+1) \end{bmatrix}, \; b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{11}$$

and m is the length of input vector x. Through the least squares problem using the equation

$$X^H.X.a = X^H.b \tag{12}$$

we come to the Yule-Walker equations [12]

$$
\begin{bmatrix}
r(1) & r(2)^* & \cdots & r(p)^* \\
r(2) & r(1) & \ddots & \vdots \\
\vdots & \ddots & \ddots & r(2)^* \\
r(p) & \cdots & r(2) & r(1)
\end{bmatrix}
\begin{bmatrix}
a(2) \\
a(3) \\
\vdots \\
a(p+1)
\end{bmatrix}
=
\begin{bmatrix}
-r(2) \\
-r(3) \\
\vdots \\
-r(p+1)
\end{bmatrix},
\tag{13}
$$

where the factor r is an autocorrelative estimation of the input vector x. This equation can be solved using the Levinson's recursion [12].

3 Simulation Results

3.1 Used Data Set and Simulation Tools

The data used for the training and test sets is taken from the video stream files of Telecommunication Networks Group, Technical University of Berlin, Germany [1]. We used trace file from MPEG-4 Jurassic Park I movie in high quality. Both the training and the test set consist of 2000 patterns (first 2000 representing the training, next 2000 the test set) and they are shown in Fig. 2. Traces from [1] were also used etc. in [2], [3].
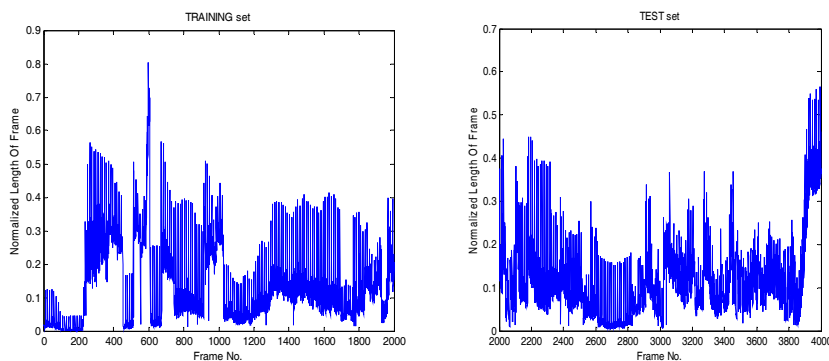
Figure 2

Training (left) and test (right) sets – originals

Our predictions were based on taking N previous patterns to predict one following pattern. Since we used a supervised learning paradigm, our networks worked with desired values of target patterns during the training process.

All our simulations were done using Stuttgart Neural Network Simulator [14] and Matlab environment [15].

3.2   Prediction by MLP

In order to achieve good results, probably one of the most important problems is to choose the appropriate configuration of neural network. During training of MLPs, we tried many types of configurations for our predictions. There were notable differences of prediction errors among them. We chose the MSEnorm (normalized mean square error) as an objective criterion to compare them.

We made experiments with the number of input neurons changing from 1 to 7 and also experiments with various number of hidden neurons and number of hidden layers. We achieved the best results of training the MLP network using network configuration 3-10-1 (which means: 3 input neurons, 10 neurons in hidden layer, 1 output neuron), Levenberg-Marquardt training algorithm and learning-rate parameter 0.1. The results of the prediction for the training and test set are shown in Fig. 3. The detail of the test set can be seen in Fig. 4.
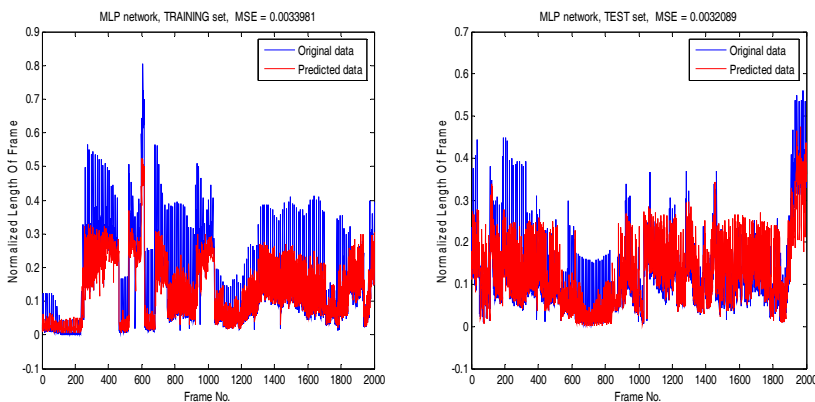
Figure 3

Prediction results for training (left) and test (right) set by MLP network 3-10-1
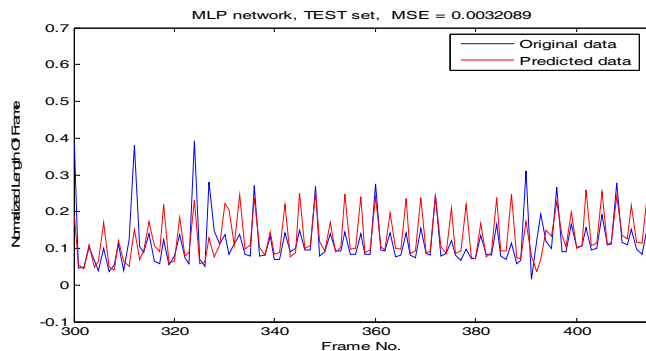

Figure 4

Prediction results for test set by MLP network 3-10-1– detail

In order to make the prediction more effective, it is possible to take also the character of the time series into account. For our data, approximately each 12th pattern forms a peak (in other words, the distance of the consecutive peaks is mostly 12 patterns). This is why we chose also 12 input patterns (besides our examined 3 inputs) for the prediction. The result for test set can be seen in Fig. 5 for MLP with configuration 12-20-10-1 (again, this is the most appropriate configuration after doing examination of various MLP configurations).
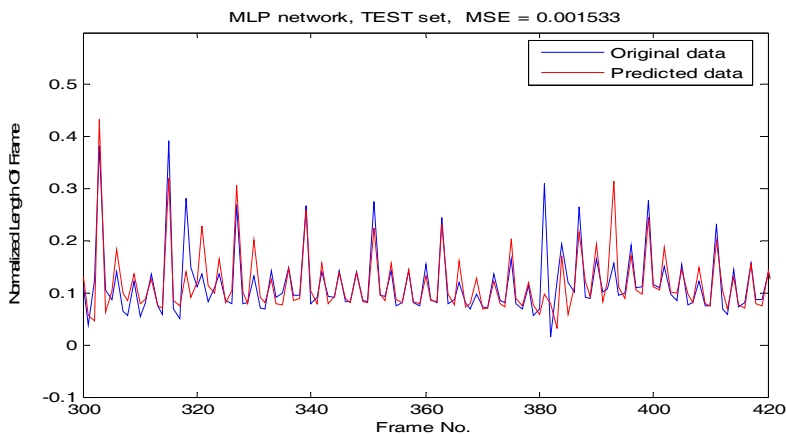
Figure 5
Prediction results for test set by MLP network 12-20-10-1– detail

3.3   Prediction by RBF Networks

In this section, we present the results of the RBF neural networks. We have tried different network configurations. Similar to searching optimal network configuration for MLPs, we tried various number of input and hidden neurons (unlike MLP, RBF network contains exactly one hidden layer). Fundamental question when using RBF networks is the number of hidden neurons to be used. The number of hidden neurons for our experiments altered from 10 to 2000. We have found out that the best approach for our data is to use 500 hidden neurons. We present best results achieved by network configuration 3-500-1. The results for the training and the test sets for this network are shown in Fig. 6. Fig. 7 contains detail for the test set.

Similar to MLP, we examined also RBF networks with 12 inputs (due to peak nature of video stream). The result is shown in Fig. 8.
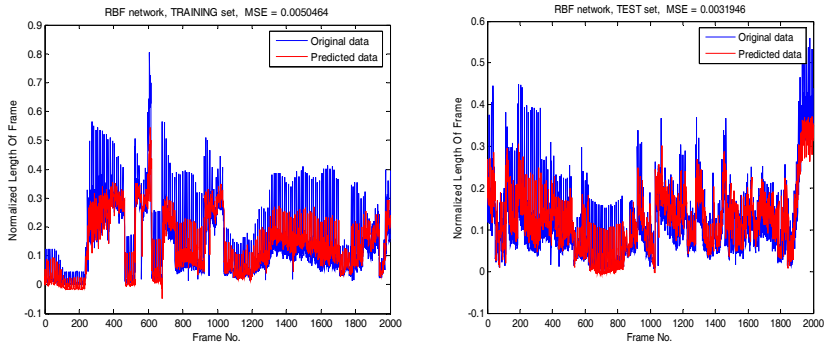
Figure 6

Prediction results for training (left) and test (right) set for RBF network 3-500-1
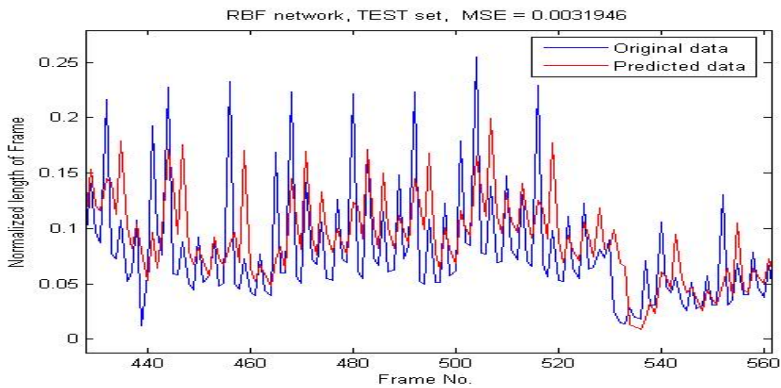

Figure 7

Prediction results for test set by RBF network 3-500-1 – detail
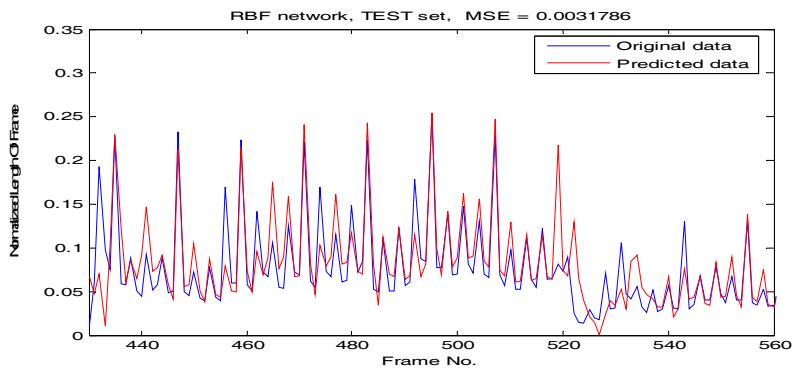

Figure 8

Prediction results for test set by RBF network 12-500-1 – detail

The comparison between RBF network of the configuration 3-500-1 and 3-2000-1 network is shown in Fig. 9. By comparing the values of the MSE for both networks, one can see that 3-500-1 RBF network behaves better.

As we mentioned earlier, our training and test sets both have 2000 elements. It means that if we create a RBF network with configuration 3-2000-1 then each hidden neuron represents one element of the training set. This approach is known as an interpolation scheme. The approximation scheme means that we use less then 2000 neurons in the hidden layer.
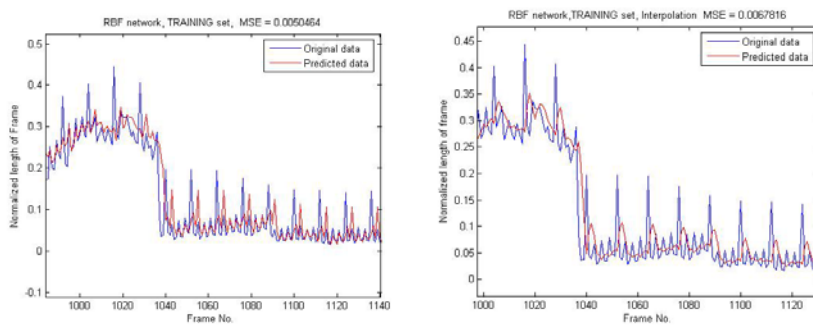

Figure 9

The comparison of two RBF networks (left: 3-500-1, right 3-2000-1) – details

In both graphs of Fig. 9 the details are shown, so we can see not only the MSE of the two networks, but it is also possible to compare the subjective performance.

3.4   Prediction by BPTT Network

For the BPTT networks, we used similar approach as for other neural network models. At first, we tried different configurations, until we found one with the best values of MSE. Starting with five hidden neurons only, the results were unsatisfactory, so we chose other configurations. The configuration containing 20 hidden neurons gave best results.

The configuration used by the BPTT is 3-20-1. The results are shown in Fig. 10 with detail included in Fig. 11. Comparing the values of MSE, we can see that the results for the BPTT networks are comparable with those of other neural networks.

Fig. 12 shows detail of the prediction using BPTT network with 12 inputs (motivation for using 12 inputs is the same as for MLP and RBF networks discussed above).
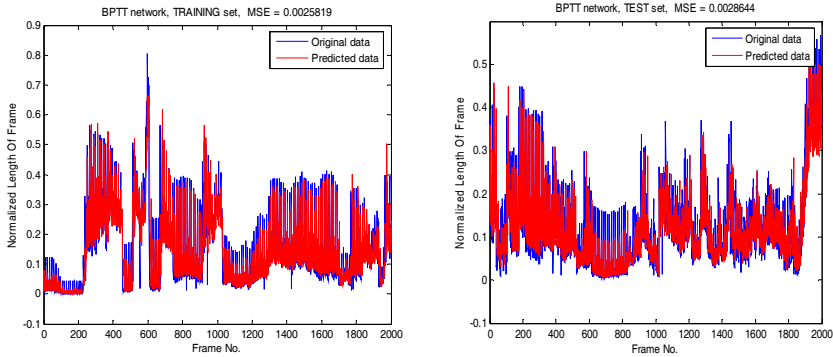
Figure 10

Prediction results for training (left) and test (right) set for BPTT network 3-20-1
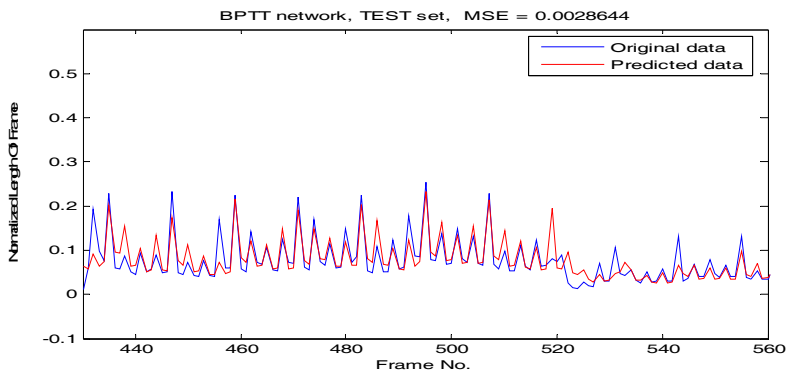

Figure 11

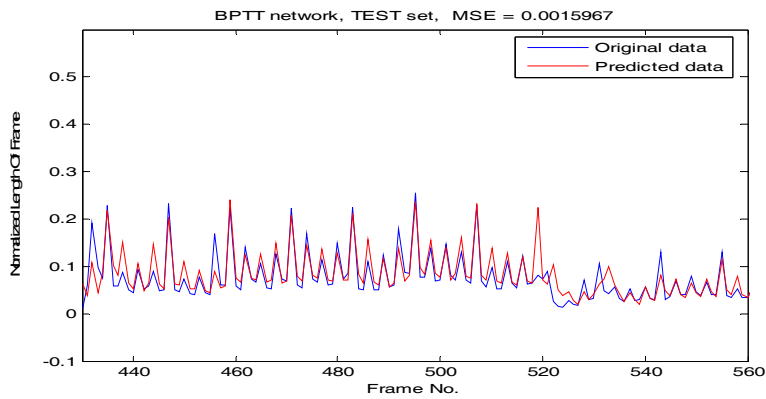Prediction results for test set by BPTT network 3-20-1 – detail

Figure 12

Prediction results for test set by BPTT network 12-20-1 – detail

3.5   Predictions Using Moving Average

Since presented trace files contain fast changes (peaks), one can expect better results when processing them not directly but using some technique to make the input signal smoother (simpler). Therefore we propose simple method of using moving averages as the input to neural network to eliminate the fast changing of signal. The principle of the moving averages (MA) is shown in Fig. 13.

We take N patterns of the original signal and compute the moving averages (summation of patterns divided by N, we move forward over one pattern) and we predict next pattern from M values of MA. Since at the time when we are predicting the next MA we know the previous patterns of the original signal too, we can obtain the next pattern of the original signal easily. We just need to multiply the predicted MA by N and subtract the last N-1 known patterns of the original signal (see Fig. 13).
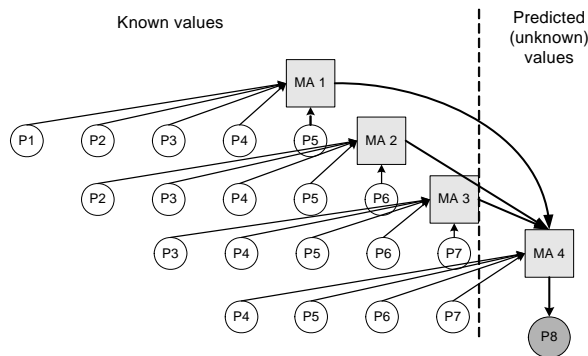

Figure 13
The principle of the moving average prediction

We can see the method of prediction of 8th pattern (P8) of the original signal in Fig. 13. Each MA is calculated from five corresponding patterns (e.g. MA1 = $1/5*\Sigma$ P(1-5)) and the next MA is predicted from three previous MA. It means, when we already know MA1 to MA3, we predict MA4 from them. Because the original patterns P4-P7 are known at that time, we just need to use inverse sequence of steps – we multiply MA4 by five and subtract the sum of P4-P7. The result is the next pattern of the original signal, i.e. P8.

The main advantage of this prediction method is that the input to the neural network is not the original data, but much less dynamic data (especially if we compute the moving averages for such number of patterns that each MA contains one peak only) – then MA input is evidently smoother than the original signal. One of the disadvantages is that small error of MA prediction can cause large error of the original signal during its reverse calculation from MA.

Moving averages from 12 patterns of training and test sets are shown in Fig. 14.
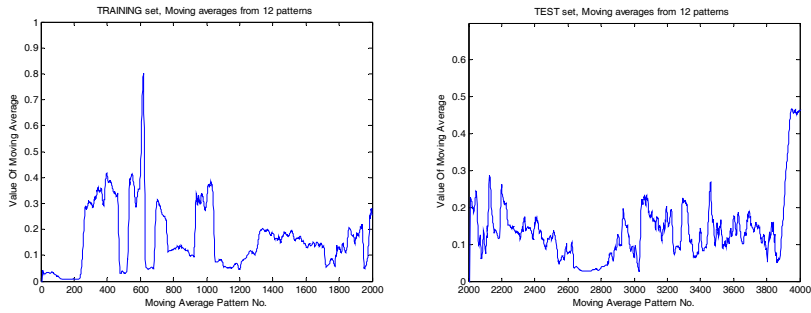
Figure 14

Moving averages of 12 patterns for training (left) and test (right) set

Prediction results using moving average for MLP network with configuration 3-10-1 are shown in Fig. 15 (with detail of the test set shown in Fig. 16).
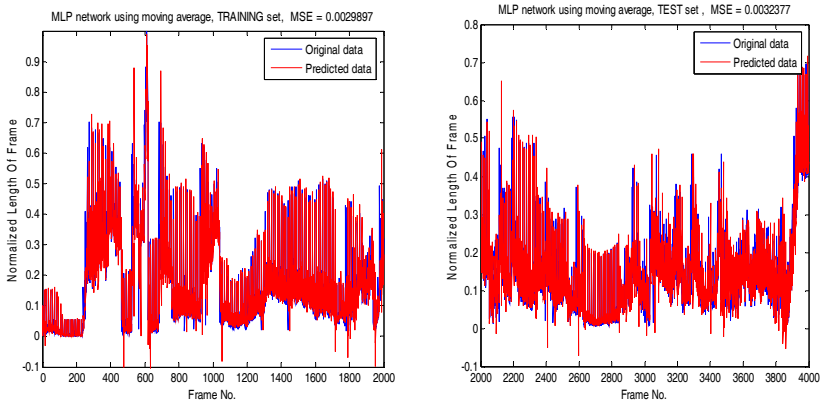

Figure 15

Prediction results for training (left) and test (right) set using moving averages for MLP network 3-10-1
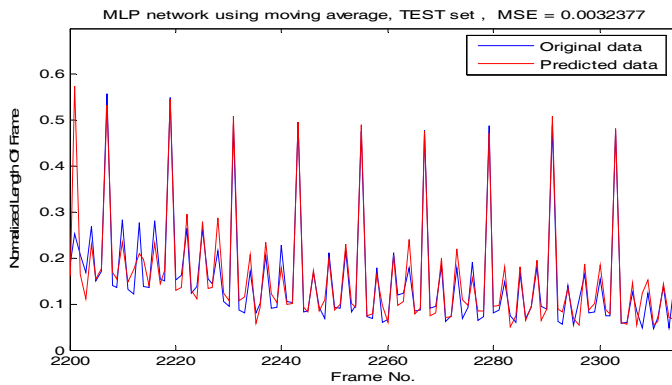

Figure 16

Prediction results for test set using moving averages for MLP network 3-10-1 – detail

We applied the concept of moving average also for MLP with 12 inputs because of periodic appearance of peaks in time series. The result is shown in Fig. 17.
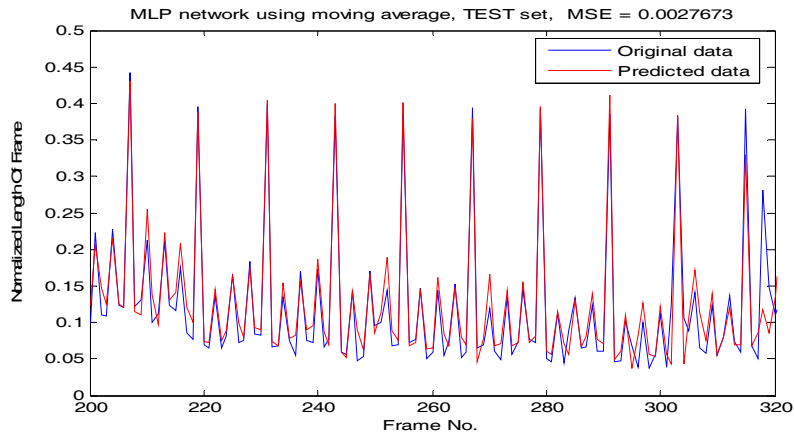

Figure 17

Prediction results for test set by using moving averages on MLP network 12-20-10-1 – detail

The results for RBF network using the concept of moving average are shown in Fig. 18. The presented network configuration is 3-500-1. Fig. 19 contains detail of moving average simulation for RBF network with 12 inputs.
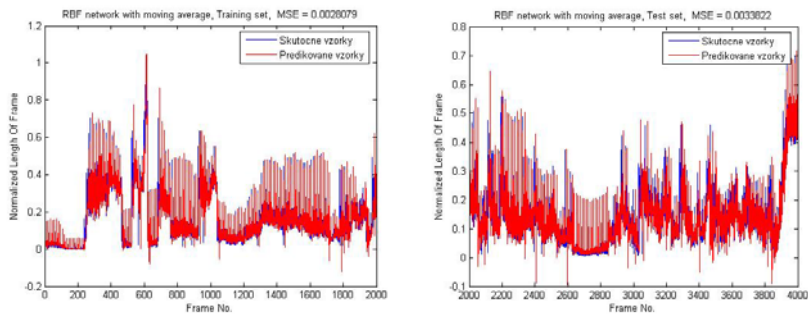

Figure 18

Prediction results for training (left) and test (right) set for RBF network 3-500-1
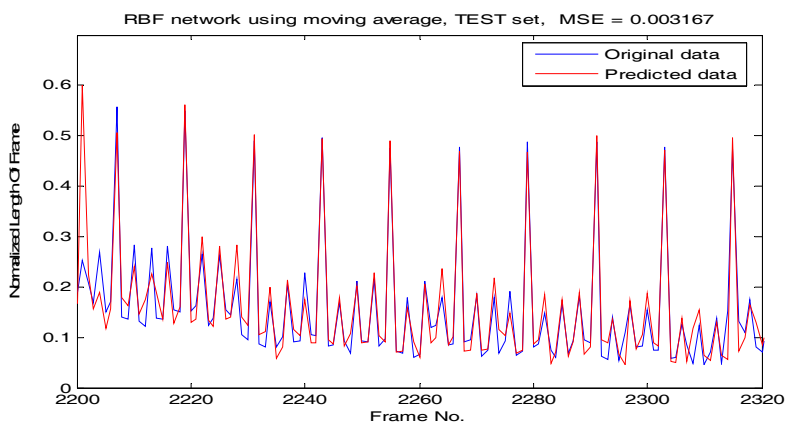
Figure 19

Prediction results for test set by using moving averages on RBF network 12-500-1 – detail

3.6　Comparison

Tab. 1 and Fig. 20 show best prediction results obtained by networks with three inputs. We got the best results using BPTT network of configuration 3-20-1 for both sets of data (normalized MSE for the training set and test set is 0.0025819 and 0.0028644, respectively). Although the results for other types of networks are little bit worse, the differences are visually not so significant. Tab. 1 also includes results from the prediction using LP (linear predictor) with 12 and 75 coefficients. In Fig. 20, the results for LP with 75 coefficients are shown.

Table 1

The comparison of MSE for the networks with 3 inputs and comparison with LP

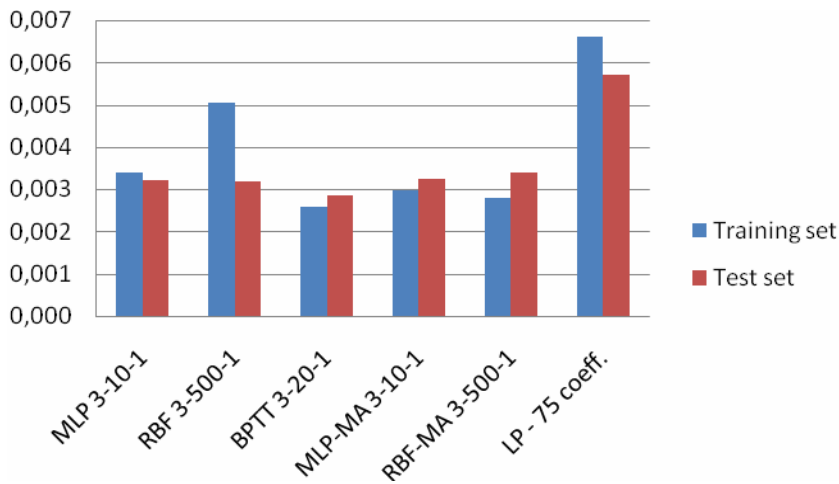| Network	Training set	Test set
MLP 3-10-1	0,0033981	0,0032089
RBF 3-500-1	0,0050464	0,0031946
BPTT 3-20-1	0,0025819	0,0028644
MLP-MA 3-10-1	0,0029892	0,0032377
RBF-MA 3-500-1	0,0028079	0,0033822
LP - 12 coeff.	0,0159024	0,0085851
LP - 75 coeff.	0,0065945	0,0057113

Figure 20

The comparison of results of test set for all types of networks using 3 inputs and comparison with LP

As we already discussed, our examined streams contain peaks, approximately at each 12th position. This is why we simulated also neural networks with 12 input neurons. The values of normalized MSE for networks using 12 inputs are shown in Tab. 2 and in Fig. 21 (also the comparison with LP is present).

Fig. 22 presents summary of best results obtained using neural networks with 3 and 12 input patterns and the comparison with the linear predictor using 75 prediction coefficients. It can be seen that better results can be obtained using 12 inputs. Since there was exactly one peak in each set of patterns fed to the network, it has learned the positions of the peaks better. Of course, the disadvantage of this way of prediction is that it strongly depends on used video stream – and different streams can have different distance between the peaks (eventually, the stream does not need to contain peaks in such periodic manner).

Table 2

The comparison of MSE of the networks with 12 inputs and LP

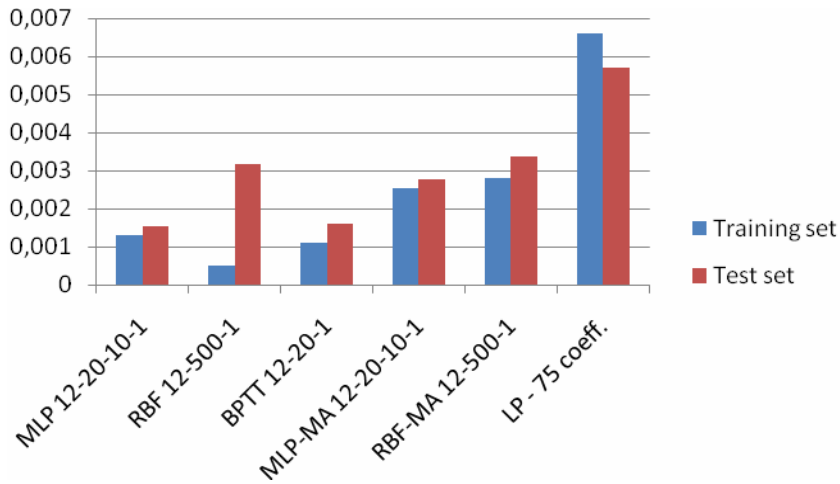| Network	Training set	Test set
MLP 12-20-10-1	0,0012937	0,0015330
RBF 12-500-1	0,0005020	0,0031786
BPTT 12-20-1	0,0010890	0,0015967
MLP-MA 12-20-10-1	0,0025311	0,0027673
RBF-MA 12-500-1	0,0028079	0,0033822
LP - 12 coeff.	0,0159024	0,0085851
LP - 75 coeff.	0,0065945	0,0057113

Figure 21

The comparison of results of test set for all types of networks using 12 inputs and comparison with LP
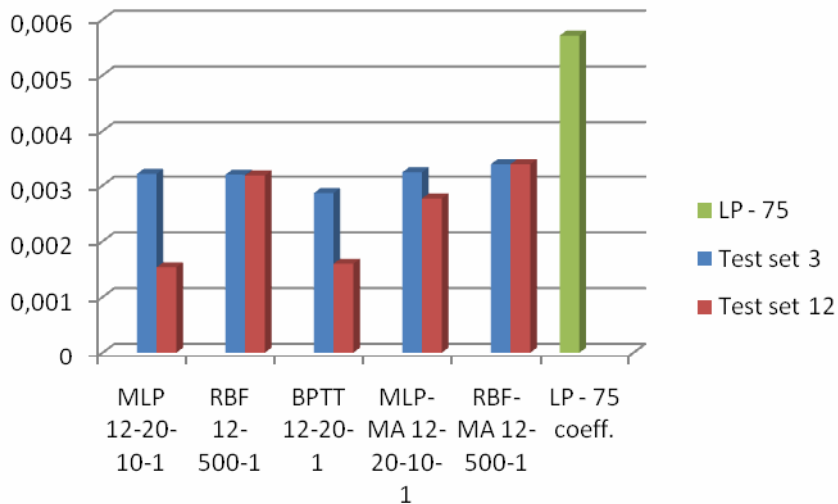
Figure 22

The comparison of results of test set of networks with 3 and 12 inputs and comparison with LP

Conclusions

Efficient data compression methods followed by efficient prediction schemes are very important in order to achieve required QoS of multimedia traffic.

The goal of this paper was to predict the video time series for an efficient bandwidth allocation of the video signal using neural networks. The application of presented methods is in traffic and congestion procedures of communication networks.

We have tried many configurations and types of neural networks for video stream data prediction. First, we tried to find suitable network configurations. This process led us to using three input patterns in each step of prediction. As we can see in Fig. 20, the best results of prediction for test set were achieved by using BPTT network with configuration 3-20-1, while the results achieved by MLP with configuration 3-10-1 and RBF with configuration 3-500-1 were a bit worse and both comparable.

Although the results for the networks using MA were very similar to the networks without MA concept, the network training was much faster (for both MLP and RBF network) which could be useful for adaptive prediction systems. Especially RBF network training using MA takes approximately 50% time duration comparing to "direct" approach to prediction.

For comparison purposes, we tried to predict the data using 12 input patterns. We chose 12 patterns in order to have just one peak in each step of prediction. Of course, the number of input patterns was found empirically and it depended on the behavior of the time series. Although better prediction results can be obtained in such manner, it is evident that the mentioned number of input patterns is not suitable for any type of time series.

In order to compare neural and conventional prediction, we presented the results of linear predictions. As can be seen from Figs. 20, 21 and 22 the results of neural networks are significantly better.

From Fig. 22, we can see that the best results of prediction for three inputs can be achieved by BPTT network. The results for other types of network were for a certain extent worse; especially the result of test set for RBF network was obviously the worst (even though the result for the training set was the best of all networks).

Besides efficient prediction schemes also efficient data (herein video data) compression methods must be used. These compression methods must take QoS requirements into account. Optimization of compressed bit flows is necessary; such optimization can be based on channel capacity (allocation of sufficient bit flow through a channel) or on receiver quality requirements. Example of such optimized coder can be found in [16].

Yet other approaches lowering requirements for channel coding are used. MPEG (as well as JPEG) standard uses block-based coding techniques. For high compression ratios, the effect of block loss or random bit error during transmission can cause a serious problem. It is possible to use error concealment algorithms [17] in block-based image coding systems, where the information of pixels surrounding a damaged block is used to reconstruct the damaged or lost blocks.

At the same time, it is necessary to take into account some other facts. Recent papers show that video traffic is of self-similar nature. In [18], VBR (variable bit

rate) MPEG-4 video is studied from the point of view of self-similarity. Authors show that modeling video sources by short-range dependent models can be unsatisfactory. VBR video may exhibit scaling behavior and thus long-range dependence must be considered. Due to its nature, wavelet-based methods can serve as a suitable tool to evaluate self-similarity [18], [19] and to determine its parameters like H (Hurst) parameter [19].

Acknowledgement

References

[1]   MPEG-4 and H.263 Video Traces for Network Performance Evaluation, http://www-tkn.ee.tu-berlin.de/research/trace/trace.html, or http://trace.eas.asu.edu/TRACE/trace.html

[2]   Yao Liang, Mei Han: Dynamic Bandwidth Allocation Based on Online Traffic Prediction for Real-Time MPEG-4 Video Streams, EURASIP Journal on Advances in Signal Processing, Volume 2007, 2007, pp. 51-60, http://www.ari.vt.edu/People/YLiang_Docs/YLiang02ASP2007.pdf

[3]   Abdennour, A.: Short-term MPEG-4 Video Traffic Prediction using ANFIS, International Journal of Network Management, Volume 15, Issue 6, 2005, pp. 377-392, http://www3.interscience.wiley.com/cgi-bin/fulltext/112139016/PDFSTART

[4]   Guoqiang Maoy, Huabing Liu: Real Time Variable Bit Rate Video Traffic Prediction, Int. Journal of Communication Systems, Vol. 20, Issue 4 (April 2007), pp. 491-505, http://www.ee.usyd.edu.au/~guoqiang/papers/Mao05realSub.pdf

[5]   Haykin, S.: Neural Networks - A Comprehensive Foundation, New York: Macmillan College Publishing Company, 1994

[6]   M. Oravec, J. Polec, S. Marchevský a kol.: Neural Networks for Digital Signal Processing (in Slovak language), Faber, Bratislava, 1998

[7]   Hlaváčková, K., Neruda, R.: Radial Basis Function Networks, Neural Network World, No. 1, 1993, pp. 93-102

[8]   Poggio, T., Girosi, F.: Networks for Approximation and Learning, Proc. of the IEEE, Vol. 78, No. 9, Sept. 1990, pp. 1481-1497

[9]   Williams, R. J., Peng, J.: An Efficient Gradient-based Algorithm for On-Line Training of Recurrent Network Trajectories, Neural Computation, *2*, 1990, pp. 490-501, ftp://ftp.ccs.neu.edu/pub/people/rjw/fast-bptt-nc-90.ps

[10]   Williams, R. J., Zipser, D.: Gradient-based Learning Algorithms for Recurrent Networks and their Computational Complexity. In: Y. Chauvin and D. E. Rumelhart (Eds.) Back-propagation: Theory, Architectures and

Applications, Hillsdale, NJ: Erlbaum, 1995, ftp://ftp.ccs.neu.edu/pub/people/rjw/bp-chap-95.ps

[11] Saliza Ismail, Abdul Manan bin Ahmad: Recurrent Neural Network with Backpropagation Through Time Algorithm for Arabic Recognition, 2004, http://www.scs-europe.net/services/esm2004/

[12] Vaidyanathan, P. P.: The Theory of Linear Prediction, Morgan & Claypool Publishers, 2008

[13] Wang, Z., Li, M., Kok, C. W.: Online Smooth Transmission with Rate Prediction for Video Streaming, Accepted by IEEE, Consumer Communications & Network Conference, Jan. 2007

[14] Zell A. et al: The SNNS Neural Network Simulator, University of Stuttgart, http://www-ra.informatik.uni-tuebingen.de/SNNS/

[15] Matlab, http://www.mathworks.com/products/matlab/

[16] Polec, J., Ondrušová, S., Kotuliaková, K., Karlubíková, T.: Hierarchical Transform Coding Using NURBS Approximation, Proceedings Elmar-2008: 50th International Symposium ELMAR-2008, Zadar, Croatia, 10-12 September 2008, Croatian Society Electronics in Marine, 2008, ISBN 978-953-7044-09-1, Vol. 1, pp. 65-68

[17] Pavlovičová, J., Polec, J., Keleši, M., Mokoš, M.: Detection of Transmission Errors in Block Codes Images, In: Journal of Electrical Engineering, ISSN 1335-3632, Vol. 56, No. 5-6 (2005), pp. 121-127

[18] Arifler, D., Evans, B. L.: Modeling the Self-Similar Behavior of Packetized MPEG-4 Video Using Wavelet-based Methods, Proc. IEEE Int. Conf. on Image Processing, Sept. 22-25, 2002, Vol. 1, Rochester, NY USA, pp. 848-851

[19] Vargic, R., Kotuliak, I.: Wavelet-Domain-based Estimation of Self-similarity for Multimedia Traffic, In: Proc. of Redzur International Workshop on Speech and Signal Processing, May 11, 2007, pp. 15-17, Bratislava, Slovakia