Web traffic prediction with artificial neural networks

Adam Gluszek, Michal Kekez, Filip Rudzinski

Department of Electrical and Computer Engineering
Kielce University of Technology
Al. 1000-lecia P. P. 7, 25-314 Kielce, Poland
a.gluszek@tu.kielce.pl, m.kekez@tu.kielce.pl, f.rudzinski@tu.kielce.pl

ABSTRACT

The main aim of the paper is to present application of the artificial neural network in the web traffic prediction. First, the general problem of time series modelling and forecasting is shortly described. Next, the details of building of dynamic processes models with the neural networks are discussed. At this point determination of the model structure in terms of its inputs and outputs is the most important question because this structure is a rough approximation of the dynamics of the modelled process. The following section of the paper presents the results obtained applying artificial neural network (classical multilayer perceptron trained with backpropagation algorithm) to the real-world web traffic prediction. Finally, we discuss the results, describe weak points of presented method and propose some alternative approaches.

Keywords: neural networks, prediction, time series, dynamic process modelling

1. INTRODUCTION

Artificial neural networks [2,7,9] are signal processing systems that make use of some of the known organizing principles of the human brain. They consist of some number of independent, simple processors called artificial neurons. These neurons communicate each other with weighted connections called the synaptic weights. The learning process of the neural network consists in adaptive modifying of the synaptic weights to improve the functioning of the entire network as the parallel signal processing system. The well-known Frank Rosenblatt’s linear perceptron from 1958 [8] was the first model of the artificial neural network. Using simple training rule and a set of examples, the perceptron was able to learn to generate desired outputs to given input signals. Unfortunately, soon became obvious that the perceptron was not able to learn to solve relatively simple problems (e.g. XOR problem [6,7]) and neural networks research slowed for several years. Interest of the neural network returned in mid-80’s, when efficient backpropagation learning algorithm was applied to train multilayer linear perceptrons. Today there are many structures of the artificial neural networks and a lot of training algorithms applied in their learning process. They are used in control, approximation, classification, decision support, modelling, prediction, data mining etc. and their area of applications is still developed.

Making predictions is one of the basic subjects in science and technology. Building the model (by means of statistical methods, artificial neural network, fuzzy system etc.) for forecasting a time series is one of the tools that can be used to analyse the mechanism that generated the data. Given the time series \( \{x(t)\}_{t=1}^{N} \) of observations of the variable \( x \) at different points in time there are two extreme situations that can occur:

- The value of the variable \( x \) at time \( t + \tau \) is uniquely determined by the values of the variable at certain times in the past, so the time series is described by the following relation:
  \[
  x(t + \tau) = f(x(t), x(t - \tau), ..., x(t - n \tau)),
  \]
  where \( f \) is some unknown function, \( \tau \) is specified time delay and \( n \) is some integer value. In this case the system is fully deterministic and its behaviour is predictable. One could derive the mapping \( f \) from initial value of the variable \( x \), provided that all the interactions between elements in the system are clearly known, or reconstruct its shape from the data \( \{x(t)\}_{t=1}^{N} \).
The values of $x(t)$ are independent random variables, so that the past values of $x$ do not influence at all its future values. There is no deterministic mechanism underlying the data, and prediction is not possible at all.

In most practical cases one has to deal with time series with properties that lie between these two extremes. For example, one could have a series of observations of a variable whose time evolution is governed by a deterministic set of equations, but the measurements are affected by noise. In this case a more appropriate model for the time series would be:

$$x(t + \tau) = f(x(t), x(t - \tau), \ldots, x(t - n \tau)) + \xi_t,$$  \hspace{1cm} (2)

where $\{\xi_t\}$ is a set of random variables. If one could exactly model the function $f$, the state of the system at time $t + \tau$ could be predicted within an accuracy that depends only on the variance of the random variables $\{\xi_t\}$. A system of this type is therefore intrinsically deterministic but some stochasticity enters at the measurement level. In other cases, as for the time series generated by the noise in a semiconductor device, the system could be intrinsically stochastic. However, since the observations are correlated in time, it might be still predictable to some extent. It is usually very difficult to discover from a time series what kind of mechanism generated the data, but building models of the type (1) is often useful to understand the relevant variables of the system and its degree of randomness.

The main aim of this paper is to present application of the artificial neural network in the web traffic prediction. Next section discusses the details of building of dynamic processes models with the neural networks. At this point determination of the model structure in terms of its inputs and outputs is the most important question because this structure is a rough approximation of the dynamics of the modelled process. The following section of the paper presents the results obtained applying artificial neural network (classical multilayer perceptron trained with backpropagation algorithm) to the real-world web traffic prediction. At the end, we discuss the results, describe weak points of presented method and propose some alternative approaches.

### 2. MODELLING OF THE DYNAMIC PROCESSES WITH ARTIFICIAL NEURAL NETWORKS

Let’s consider a dynamic system with $r$ physical inputs $u_1, u_2, \ldots, u_r$ ($u_k \in U_k$) and $s$ physical outputs $z_1, z_2, \ldots, z_s$ ($z_l \in Z_l$). The behaviour of the system is described by $T$ input-output sets of data:

$$L = \{u'_t, z'_t\}_{t=1}^T,$$  \hspace{1cm} (3)

where $u'_t = (u'_t, u'_{2t}, \ldots, u'_{rt}) \in U = U_1 \times U_2 \times \ldots \times U_r$ and $z'_t = (z'_{1t}, z'_{2t}, \ldots, z'_{st}) \in Z = Z_1 \times Z_2 \times \ldots \times Z_s$.

The first step of a neural network model design consists in the determination of the model structure in terms of its inputs and outputs, because in fact each neural network is itself a static system. Therefore, neural network structure (in terms of its inputs and outputs) is an approximation of the dynamics of the system to be modelled. Collections of inputs and outputs of the neural model and the system to be modelled are identical only in the case of a static system. For example, for a static system with one input $u$ and one output $z$, its neural network based model also has one input $u_t$ and one output $z_t$ and is described by the following formula:

$$z_t = f(u_t),$$  \hspace{1cm} (4)

where $f(\ )$ is a function, which is to be identified based on data $L$ (3). On the other hand, for a dynamic system with one input $u$ and one output $z$, its model – in general – can be described by the following formula:

$$z_t = f(u_{t-1}, u_{t-2}, \ldots, u_{t-M}, z_{t-1}, z_{t-2}, \ldots, z_{t-N}),$$  \hspace{1cm} (5)
where physical input $u$ and physical output $z$ taken from selected previous time instants are treated as additional inputs of the model.

Formulas (4) and (5) can be easily generalized for the broader case of the system with $r$ inputs and $s$ outputs. However, for the problem of time series modelling considered dynamic system has only one output $z$ and its model is described by less complicated formula:

$$z_t = f(z_{t-1}, z_{t-2}, \ldots, z_{t-N}).$$

(6)

Assume that the structure of the neural network based model in terms of its inputs and outputs has been determined, and the model has $n$ inputs $x_1, x_2, \ldots, x_n$, $x_j \in X_j$ ($n \geq r$), and $m$ outputs $y_1, y_2, \ldots, y_m$, $y_j \in Y_j$ ($m$ is usually equal to $s$).

The initial data set $L(3)$ must be now reedited to the ‘static’ form (according to the model’s structure):

$$S = \{x'_p, y'_p\}_{p=1}^P,$$

(7)

where $x'_p = (x'_1, x'_2, \ldots, x'_n) \in X = X_1 \times X_2 \times \ldots \times X_n$, $y'_p = (y'_1, y'_2, \ldots, y'_m) \in Y = Y_1 \times Y_2 \times \ldots \times Y_m$ and $p$ is the number of model’s input-output static data pattern. Data $S(7)$ are the learning data for the static neural network used as a model of a dynamic system described by $L(3)$. Additionally, let $S_x = \{x'_p\}_{p=1}^P \subset X$.

The problem of designing the model of the considered system – within the framework of neural network methodology – consists in finding of a mapping $M: X \rightarrow Y$ provided its restriction on data $S$ (learning data) $M_S: S_x \rightarrow Y$ is known. It is worth emphasizing that ideal case when the whole learning data set is exactly mapped by $M$, that is, the learning error is equal to zero is in fact not required, because it usually means an overtraining of the neural system, which results in its poor generalizing capabilities. Usually the learning of the neural-network-based system is a compromise between obtaining - on one hand - a sufficiently accurate mapping of the learning data set, and - on the other hand - good generalization. Therefore, the actual restriction of the mapping $M$ for the learning data domain is usually an approximation of the true mapping $M_S$.

### 3. WEB TRAFFIC PREDICTION

This section presents the results obtained applying artificial neural network to the real-world web traffic prediction. Our problem was to forecast future traffic volume on the basis of its some previous values. Considered data are taken from the San Francisco State University campus network traffic report [5]. The data represent total number of bytes received and sent by the network measured over 30 minute intervals for 14 days (from 22/04/2004 to 05/05/2004). In our numerical experiments we used classical multilayer perceptron trained with backpropagation algorithm [2,7,9]. The perceptron had 3 hidden layers with 20, 10 and 5 artificial neurons with sigmoid activation function. We used data for the first ten days as the training data. The data for next four days weren’t shown to the network and were applied as the testing set.

For the assessment of the prediction accuracy we used two quantitative criteria. First of them was root mean square error $RMSE$:

$$RMSE = \sqrt{\frac{1}{K} \sum_{k=1}^K (y_{k}^0 - y_k^{'})^2},$$

(8)

where $y_{k}^0$ denotes the $k$-th sample of output learning/test data and $y_k^{' }$ is a numerical response of the predicting system for the $k$-th sample of the corresponding input data. The second criterion was $RMSE/VAR$ ratio, where $VAR$ stands for variance calculated according to the following formula:
\[ VAR = \frac{1}{K} \sum_{k=1}^{K} (y_k^* - \bar{y})^2 , \]  

(9)

where \( y_k^* \) denotes the \( k \)-th sample of output learning/test data and \( \bar{y} \) is the average value of the learning/test data set. The quantity \( RMSE/VAR \) is particularly significant, because if it has value zero then prediction is perfect, while if it is equal to 1 then prediction is no better than performed by the system which permanently generates average of the predicted values.

The first step of the applying of the neural network in considered problem was the determination of the model structure in terms of its inputs according to formula (6). We tested some models with increasing number of the inputs corresponding to predicted variable taken from consecutive previous time instants. The figure 1 presents plot of the \( RMSE \) error (both for training and test data) versus maximum time delay of the model input.

![Graph showing RMSE error versus maximum time delay of the model input](image)

Fig. 1. \( RMSE \) error versus maximum time delay of the model input \( N \).

The best results we obtained for model with time delay of the input \( N = 3 \). For this model the value of \( RMSE \) for testing data was the minimal among all considered models. \( RMSE \) for training data was slightly bigger than for model with \( N = 7 \), but in this case structure of the neural network was significantly more complicated (model had 7 inputs while for \( N = 3 \) only 3). It means that optimal structure of the model for considered problem has a form:

\[ z_t = f(z_{t-1}, z_{t-2}, z_{t-3}) . \]  

(10)

Figure 2 illustrates the functioning of the model for both training and testing data sets. The dashed line represents actual values taken from traffic report [5] and the solid one represents values generated by the model. The vertical solid line is the border between training and test data.
As it is shown in the picture 2 results of forecasting are quite good for middle range of predicted variable. Prediction error significantly increases only for rapid growths of the traffic during the day (probably corresponding to transferring a huge mass of data e.g. video files) and for falls of the data flow in the night. Day peaks of data flow are completely unpredictable because they depend on individual users of the net, but the accuracy of the model for minimal night traffic probably can be increased.

At the end of our considerations we performed comparison of proposed neural network based model with some alternative approaches. The following methods have been tested: neuro-fuzzy system nfgMod [3,4], classical linear autoregression and regression tree system CHAID [1]. Structure of these models was consistent with formula (10). Table 1 summarizes the accuracy analysis of all considered models.

<table>
<thead>
<tr>
<th></th>
<th>Neural network</th>
<th>Auto-regression</th>
<th>Neuro-fuzzy system nfgMod</th>
<th>System CHAID</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE (learning data)</td>
<td>$5.139 \cdot 10^8$</td>
<td>$4.730 \cdot 10^8$</td>
<td>$4.704 \cdot 10^8$</td>
<td>$4.676 \cdot 10^8$</td>
</tr>
<tr>
<td>RMSE (test data)</td>
<td>$6.453 \cdot 10^8$</td>
<td>$5.663 \cdot 10^8$</td>
<td>$5.539 \cdot 10^8$</td>
<td>$5.911 \cdot 10^8$</td>
</tr>
<tr>
<td>RMSE/VAR (learning data)</td>
<td>0.508</td>
<td>0.468</td>
<td>0.465</td>
<td>0.462</td>
</tr>
<tr>
<td>RMSE/VAR (test data)</td>
<td>0.622</td>
<td>0.546</td>
<td>0.534</td>
<td>0.570</td>
</tr>
</tbody>
</table>

Accuracy of all considered models was comparable. The best mapping of training data was achieved by regression tree system CHAID. For the test data the best value of prediction error had the neuro-fuzzy system nfgMod. Efficient
functioning of the model for the test data set is very important question, because it confirms generalizing abilities of the predicting system and in result verify its actual usefulness. In conclusion we can state that considered traffic data are to some extent predictable and for future research we expect that quality of forecasting can be improved e.g. by preprocessing the raw data or applying model structure different from described by formula (10).

4. CONCLUSIONS

The paper presents application of the artificial neural network in the web traffic prediction. In the first section, the general problem of time series modelling and forecasting is shortly described. Next section discusses the details of building of dynamic processes models with the neural networks. At this point we paid special attention to determination of the model structure in terms of its inputs and outputs. The following section of the paper presents the results obtained applying artificial neural network (classical multilayer perceptron trained with backpropagation algorithm) to the real-world web traffic prediction. Finally, we discuss the results of comparison of proposed neural network based model with some alternative approaches.

5. REFERENCES

5. http://ipaudit.sfsu.edu/~ipaudit/cgi-bin/