WATER INFLOW FORECASTING BASED ON NEURAL NETWORK

T. Štokelj*, R. Golob**

**Soške elektrarne (SENG), Erjavčeva 20, 5000 Nova Gorica, Slovenia, email: tomaz@ees.fer.uni-lj.si*
***Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia, email: robert@lddees.fer.uni-lj.si*

Abstract: In the paper, a new approach to forecasting water inflow into the head hydro power plant reservoir is described. Water inflow forecasting is usually based on precipitation data collected by the ombrometer stations in the river basin. Solution of this problem is rather complex due to highly non-linear relation between the amount of precipitation at different locations and water inflow. Therefore neural networks are used to solve this problem. First, selection of input parameters is discussed. Next, the most appropriate architecture of the neural network is chosen. Finally, efficacy of the proposed method is tested for a practical case and some results are presented.

Keywords: Hydroelectric systems, Forecasts, Neural networks, Efficiency enhancement, Power-system control

1. INTRODUCTION

Optimal exploitation of the water in the head hydro power plant (HPP) reservoir is of great importance for operation of a cascaded hydro system (CHS). Problems may occur when natural water inflow exceeds the installed capacity of a head HPP and water spillage becomes necessary. It is also true, that during the drought periods, water discharge easily exceeds the natural inflow resulting in lower HPP head and consequently in decrease of electric energy production. Because of these facts it is very important for the CHS operator to be able to anticipate sudden increase in water inflow on time. Ability to properly forecast the increase of natural inflow can result in increased electric energy production due to enhanced flexibility in stored water management. Unfortunately, the mathematical relation between the amount of precipitation at different locations and natural inflow is highly non-linear and a corresponding mathematical relations are not easy to be found. Therefore, new solutions were sought. One approach that seemed promising was application of artificial neural networks. The main advantage of neural network based methods is that for system identification, which can be in turn utilised for constructing a "black-box" model of the system, no particular knowledge on the physical properties of the system itself is required. As such, they appear very suitable for natural inflow forecasting.

In the paper, a neural network based algorithm for natural water inflow forecasting is applied to a specific case of Soca river CHS. The hydro system in the Soca valley is managed by SENG company and consists of three larger and a number of smaller HPPs. The head HPP is HP Doblar which also has the largest storage capacity. A very important fact that needs to be considered in the operation planning is the Soca's river torrential character. In the periods of heavy precipitation, natural inflow into the HP Doblar's reservoir may rise up to twenty or more times the average monthly inflow in a matter of hours, thus creating a lot of problems to CHS operator. As a first step to deal with this situation, SENG company has installed a number of ombrometers in the upper part of the Soca river basin over the past years. The main task of these ombrometer stations is to collect data on the amount of precipitation in various parts of the basin and to pass the data to the CHS dispatching centre every 15 minutes. Data collection system has become operational for the first time during the last year. However, there is no computational tool for natural inflow forecasting installed in the control centre and the operators have to rely for water inflow forecasting on their own experience, which can result in rather inaccurate estimations. It is well known, that natural inflow is not governed solely by the amount of precipitation, but is also influenced by air temperature, melting of snow in the nearby mountains, earth humidity, local rainstorms, etc. Therefore, in the second phase of the project, SENG staff is planing to install some more ombrometers and also to add the devices for measuring air temperature. Of course, more information to deal with, will render the operators life even more complicated.

The science about neural networks is quite young. Its beginning starts approximately 50 years ago, while first applications were developed some 10 years ago. The utilisation of neural networks makes sense in identification, control and forecasting. The main properties of neural networks are non-linearity and generalisation, that is, neural network can give good results even for input samples quite different from learning samples (Dobnikar, 1990).

2. ARTIFICIAL NEURAL NETWORKS

2.1 Background

Neuron is an information-processing unit, whose inputs are weighted. Usually, neuron has an additionally parameter called threshold, which can be treated as a weighted input with input value 1. The sum of neuron inputs is forwarded to an activation function, while the output of the activation function represents the output of the neuron. Neural network called multilayer perceptron is a number of neurons connected together. Based on the architecture of neural network, a layer designs a set of vertically arranged neurons with inputs connected to the outputs of neurons in the previous layer and with outputs connected to the neurons in the next layer.

Input value of the j^{th} neuron can be calculated as:

$$v_j = \sum_{i \in P} \left(w_{ji} \cdot x_i \right) \tag{1}$$

$$y_j = \varphi\left(v_j\right) \tag{2}$$

where y_j is the output value of the neuron j, x_i are input values taken from neurons placed in the previous layer, with $\mathcal{P}$ denoting the set of neurons in the previous layer and w_{ji} are weights of a neuron. Input values are calculated in the same sequence as the layers lay in the network, that is first layer neurons are calculated first, second layer neurons are calculated second and so on.

2.2 Learning process

At the beginning of the learning process, enough large and representative sample database including inputs and outputs for the system to be identified is required. Next, for the same input samples, the output of neural network is calculated with initial weights. Training of neural network means changing the values of weights in such a way that the outputs produced by the neural network fit the corresponding outputs from the database as close as possible. As a result, such a set of weights must be found that the cost function, as defined in equation 3, takes minimum value. For training of multilayer perceptron (MPC), "Back-Propagation Algorithm" (BPG algorithm) is used. In that case, the cost function $\mathcal{E}$ is defined as a sum of squared difference between database output d_j and neural network output y_j for all output neurons and training samples.

$$\mathsf{E} = \sum_{n=1}^{N} \sum_{j \in O} \left(d_j(n) - y_j(n) \right)^2 \tag{3}$$

where N is number of input samples, used for learning and $\mathcal{O}$ denotes the set of neurons of the last layer, that is outputs. As it can be seen, the cost function is non-linear resulting in many local minima. That bring some difficulties in searching the global minimum of cost function. Learning algorithm can be divided in 4 steps (Haykin, 1994):

1. Neural network output for initial values of weights is calculated using equations (1) and (2).
2. Error for every output neuron is calculated as a difference between output from the database and corresponding neural network output:

$$e_j(n) = d_j(n) - y_j(n) \quad \forall \quad j \in O \tag{4}$$

3. Delta function δ_j is calculated for each neuron. First, delta functions for output layer neurons are calculated:

$$\delta_j(n) = e_j(n) \cdot \varphi_j'\left(v_j(n)\right) \quad \forall \quad j \in O \tag{5}$$

where $\varphi'(\cdot)$ indicates derivative of activation function $\varphi(\cdot)$ on its argument. Finally, for neurons from all other layers delta functions are:

$$\delta_j(n) = \varphi_j'\left(v_j(n)\right) \cdot \sum_{k \in N} \left(\delta_k(n) \cdot w_{kj}(n) \right) \quad \forall \quad j \notin O \tag{6}$$

where $\mathcal{N}$ denotes the set of neurons in the next layer.

4. Weights are then updated according to:

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot y_j(n) \quad (7)$$

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \quad (8)$$

η is the learning rate parameter which determines the speed of learning process. If the learning rate is too small, learning is slow, while large learning rate causes unstable updating of weight values.

Since the global minimum of the cost function can not be found in one iteration, this four step routine has to be repeated for a number of times. Usually, learning samples are divided into two subsets: a learning and a testing subset. With samples from the learning subset, neural network is trained, while samples from testing subset are used for controlling the learning process. When the sum squared error for testing subset (equation 3) starts growing, the learning process is completed. This mechanism also works for preventing the overfitting problem.

Described algorithm can be further enhanced by a variable learning rate.

1. When the error ε in a particular iteration exceeds the error in the previous step by more than ξ (for example 1.04), then new calculated weights are not considered and the learning rate is decreased (for example multiplied with 0.7).
2. When the error in the current iteration is greater than the error in the previous iteration, but it is not larger than ξ, then new calculated weights are considered and the learning rate is not changed.
3. When the error is smaller than the error in the previous step, then the learning rate is increased (for example multiplied with 1.05).

Learning process can be additionally improved if adjusting of weights is not calculated with equation (7) but as follows:

$$\Delta w_{ji}(n) = \alpha \cdot \Delta w_{ji}(n-1) + (1-\alpha) \cdot \eta \cdot \delta_j(n) \cdot y_j(n) \quad (9)$$

This is so called learning with momentum. Change of weights in the current iteration depends also on change of weights in previous iteration where α is momentum constant and is usually slightly less than 1 (for example 0.9). Learning speed can also be improved by a proper choice of initial weights (Nguyen, Windrow, 1990).

2.3 Implementation of neural network based algorithm

Due to a rather limited database, namely only 3 ombrometers have been operational for one year, first some basic correlation between precipitation and natural water inflow has to be drawn from SENG operator guidelines (SENG, 1996). After analysing the data measured by the ombrometers on rainy days and natural water inflow data, it was estimated that the natural inflow into the HP Doblar reservoir starts to rise some 2 to 6 hours after heavy precipitation begun. The delay time varies according to the distance of the ombrometer station from HP Doblar. Therefore, an algorithm was developed that forecasts natural water inflow for 2, 4 and 6 hours ahead. As inputs for neural network based model, following quantities were selected:

- amount of precipitation in the last 15 minutes
- amount of precipitation in the last hour
- amount of precipitation in the last 2 hours
- amount of precipitation in the last 4 hours
- current natural water inflow into the HP Doblar reservoir
- natural inflow into the HP Doblar reservoir 8 hours ago

By including data on current water inflow and water inflow 8 hours ago, the influence of earth humidity is captured.

The neural network model for water inflow forecasting includes 12 inputs regarding precipitation data (3 ombrometer stations and 4 inputs per station) and two inputs regarding water inflow, totalling 14 inputs. As already mentioned, the neural network has 3 outputs: water inflow for 2, 4 and 6 hours ahead. As a result, input matrix has dimensions 14 x 8976 and output matrix has dimensions 3 x 8976. Out of the entire database, 80 % of data are used as learning samples and 20 % as testing samples. The learning set is divided into two subsets, 70 % of learning set in learning subset and 30 % in testing subset, which are both used during the learning process.

A multilayer perceptron with two layers is used: one hidden layer and the output layer. The architecture of the neural network is shown on figure 1. The hidden layer consists of neurons with sigmoidal activation function:

$$\varphi(v) = \frac{1}{1+e^{-v}} \quad (10)$$

This activation function is non-linear and constrained. Neural network has 3 outputs with linear activation function:

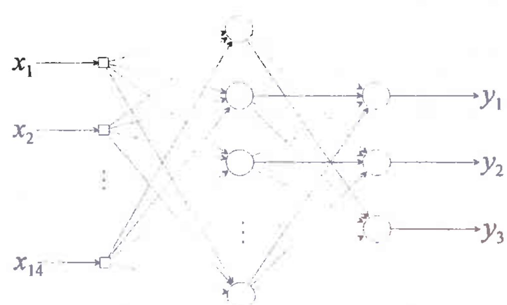

Figure 1: Neural network architecture

$$\varphi(v) = v \tag{11}$$

The neural network based model was implemented using the software package "Matlab" and its supplement "Neural Network Toolbox" (Demuth, Beale, 1993).

3. RESULTS

In figure 2, the behaviour of cost function (learning subset - figure 2a, testing subset - figure 2b) during the learning process is shown. From figure 2a, it can be observed that the cost function on learning subset is constantly decreasing. On the other hand, the cost function on testing subset, following a continuous decrease at the beginning, starts to gradually increase after some 6000 epochs. Therefore, final weights are determined as those causing the cost function on testing subset to take the minimal value.

Another important issue that needs to be addressed is the number of neurons in the hidden layer. In general,

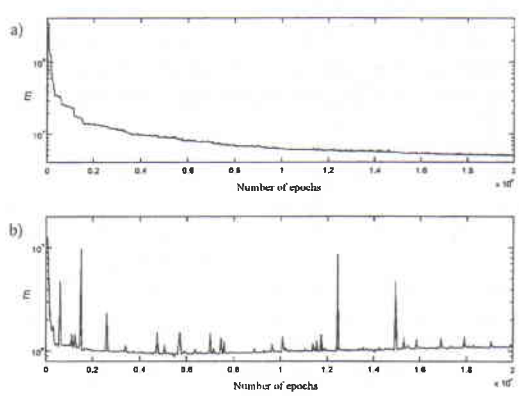

Figure 2: Value of cost function during learning process a) for learning subset and b) for testing subset

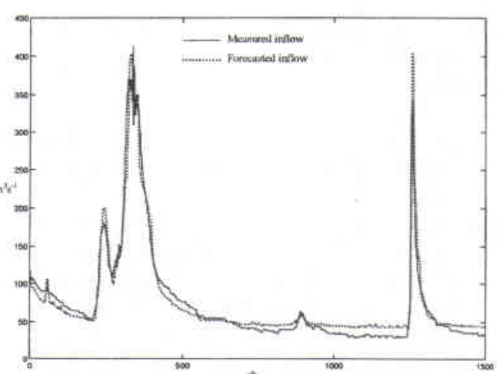

Figure 3: Measured and forecasted inflow with 7 neurons in the hidden layer for 4 hours ahead

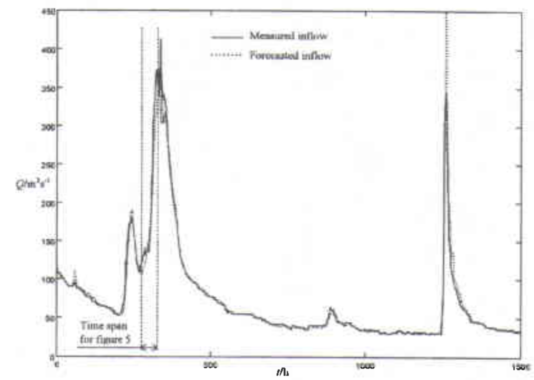

Figure 4: Measured and forecasted inflow with 40 neurons in the hidden layer for 4 hours ahead

keeping the number small is reflected in fast learning but it may cause some inaccuracy of results. On the other hand, larger number of neurons inevitably leads to longer learning process. In order to cope with this problem, a number of different architectures were analysed and hereafter, results for two examples are shown.

First, neural network was trained with only 7 neurons in the hidden layer. In figure 3, natural inflow forecasted for 4 hours ahead is compared to measured water inflow. It can be seen, that forecasted inflow is rather inaccurate, especially during the periods with low natural inflow.

Next, neural network with 40 neurons in the hidden layer was trained. For that case, results are shown in figure 4. A much higher accuracy can be observed. Some inaccurate results can only be found during the periods with high natural inflow or during the periods when water inflow changes rapidly. One example of

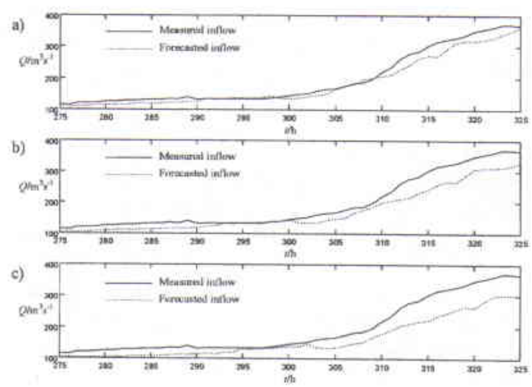

Figure 5: Measured and forecasted inflow with 40 neurons in the hidden layer a) for 2 hours ahead, b) for 4 hours ahead and c) for 6 hours ahead

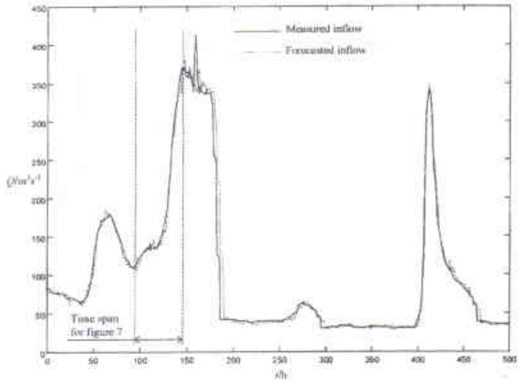

Figure 6: Measured and forecasted water inflow for reduced database with 40 neurons in the hidden layer for 4 hours ahead

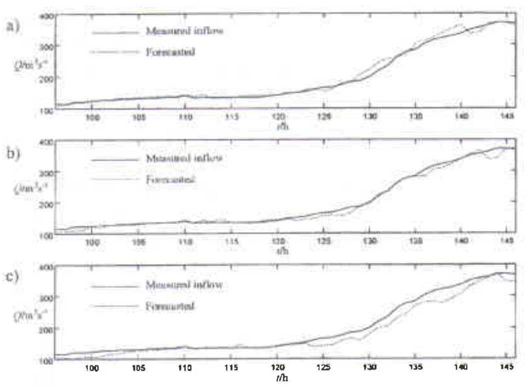

Figure 7: Measured and forecasted water inflow for reduced database with 40 neurons in the hidden layer a) for 2 hours ahead, b) for 4 hours ahead and c) for 6 hours ahead

such period is zoomed in figure 5, where forecasted and measured inflows for 2 (figure 5a), 4 (figure 5b) and 6 (figure 5c) hours ahead are presented. It can be seen, that forecast for 2 hours ahead is quite accurate, while forecasts for 4 and 6 hours ahead are more erroneous.

Similar result were obtained with some other architectures, as well. Therefore, it is obvious that the main reason for inaccurate forecasting does not lie in the neural network itself, but in database used for learning. After analysing the learning samples, it was noticed, that database includes both: days with and without precipitation. Learning samples with precipitation input values set to zero, may due to a number of reasons have different output values, thus confusing the neural network during the learning process. That was singled out as a major limitation for accurate forecasting of water inflow.

In the next step, all learning samples for days without precipitation were ignored and a reduced database was built. Namely, in the database for learning process only 3287 out of total 8976 samples were considered. That also resulted in speeding up the learning process, significantly. Some results for reduced database are shown in figure 6. One of the periods with significant increase of natural inflow is zoomed in and shown in figure 7a, 7b, and 7c for 2, 4, and 6 hours ahead, respectively.

By comparing figures 5 and 7, it can be observed that much better results are obtained when the reduced database is used during the learning process.

4. CONCLUSION

In the paper, a new algorithm for natural inflow forecasting based on artificial neural networks is developed. As input variables, precipitation data and past natural inflows were selected. Various architectures of neural networks were analysed to find the most appropriate solution for this particular problem. The algorithm was tested for Soca river cascaded hydro system. The obtained results are promising, especially when taken into account that data from 3 ombrometer stations only, were available.

The presented algorithm will be implemented in the regional control centre and will as such be of great aid to the operator in coping with control of the hydro reservoir level during the periods with heavy precipitation. In future, more ombrometer stations are planned to be installed and a system for measuring air temperature is also going to be implemented. The improved forecaster is planed to be incorporated into the software package for optimal scheduling of the Soca river cascaded hydro system, that is currently under development.

5. REFERENCES

Demuth, H., Beale, M. (1993). *Neural Network Toolbox*, MathWorks, Natick, Mass., U.S.A.

Dobnikar, A. (1990). *Neural networks*; Didakta, Radovljica, Slovenia.

Haykin, S.(1994). *Neural Networks*; Macmillan College Publishing Company, New York, U.S.A..

Nguyen, D., Windrow, B. (1990). *Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights*; International Joint Conference of Neural Networks, Vol. 3, July 1990,pp. 21-26.

SENG, (1998). *SENG operators guidelines in case of heavy precipitation*; Technical Report, SENG, N. Gorica, Slovenia.