Kärkkäinen K. & H. Sirviö & P. Vakkilainen (1997) A comparison of Inflow Forecasting Techniques.

NHP-Raport 44. 10 p.

A COMPARISON OF INFLOW FORECASTING TECHNIQUES

K. Kärkkäinen[1], H. Sirviö[2] and P. Vakkilainen[3]

[1] Tieto Corporation / Unic, Kalkkipellontie 6, P.O. Box 101,
FIN-02601 Espoo, Finland, e-mail: Kari.Karkkainen@unic.fi
[2] Finnish Environment Institute, P.O. Box 140, FIN-00251 Helsinki, Finland, e-mail: hannu.sirvio@vyh.fi
[3] Helsinki University of Technology, Laboratory of Water Resources, P.O.
Box 5300, FIN-02015 HUT, Finland, e-mail: pv@ahti.hut.fi

ABSTRACT

An artificial neural network (NN) and time-series models were applied to predicting the inflow of Lake Päijänne. The main objective of the study was to test the applicability of the neural network technique to short-term runoff prediction. A NN consists of a set of layered neurons and their weighted interconnections. NNs generate adjust these weights by learning from examples. Previous inflows, air temperatures, snow water equivalents (SWEs), measured runoff values from small experimental basins, and Julian days were tested as input variables of the NN. The learning process was successful and the network was used for inflow prediction. The predictions obtained by the NN were compared with those made by time-series models of type PAR, ARMA and SARIMA. The results seem to indicate, that NNs are comparable to some advanced time-series models in inflow forecasting.

INTRODUCTION

Conceptual rainfall-runoff models are often used as operational lake inflow prediction models. Examples of such models are the HBV model (Bergström 1976) and its variation WSFS (Vehviläinen 1994), which is used operationally in Finland. Other conceptual models have also been tested for the finnish conditions (Karvonen 1984). Water resource managers, who have the responsibility of real-time operation of regulated lakes and reservoirs, may also appreciate predictions made using alternative methods. One possibility is to use time-series models, whose applicability to the finnish conditions has been analyzed by Malve (1986) and recently by Sirviö (1998).

A rather new innovation is to use an artificial neural networks (NNs) for inflow prediction. The use of NNs has in the recent years got plenty of interest in different fields. Applications of NNs in hydrology and water resources engineering have been reported by, e.g., Daniell (1991), Karunanithi et al.(1994), and Härkönen (1995).

The purpose of this work was to test the neural network method – backpropagation NNs were used – and to compare the predictive capabilities of an NN forecasting model to those of some time-series models.

HYDROLOGICAL DATA

Lake Päijänne is one of the biggest lakes in Finland with a catchment area of 26480 km^2. The areal percentage of lakes on the catchment is 19.5 measured at the outlet of the lake. Daily water levels and release values from the years 1960...1990 were used in this study to calculate the daily inflow values. Thus obtained net inflow values were smoothed according to the procedure described by Kärkkäinen (1997).

Daily runoff data from three small experimental basins, numbers 41 (A=29,7 km^2), 71 (A=9.4 km^2), and 72 (A=5.39 km^2) (Leppäjärvi 1992), located on the lake's watershed, was used as input data to the NN. Daily average air temperatures from the Heinola and Jyväskylä synoptic weather stations and areal snow water equivalent (SWE) data prepared by the Hydrological office of the Finnish National Board of Waters and the Environment were also used in the study.

The time step used in the study was five days forgetting the leap day in order to obtain stable 73 steps in a year.

BACKPROPAGATION NEURAL NETWORKS

The backpropagation neural network has been the most widely used type of a NN. It consists of an ordered set of layers: an input layer, a hidden layer or layers, and an output layer (Figure 1.). A layer has all of its inputs connected to the layer preceding it and all of its outputs connected to the layer succeeding it. The inputs of the input layer are the inputs of the whole network and the outputs of the output layer are the outputs of the whole network. There are usually many neurons in each layer, each neuron having multiple inputs and a single output. Each input "fibre" of a neuron has an associated weight and the neuron is activated by a weighted sum of all of its inputs (the activation value). The output of a neuron is a function of the weighted sum (an activation function). As can be seen, the number of weights even in a small NN may be quite large.

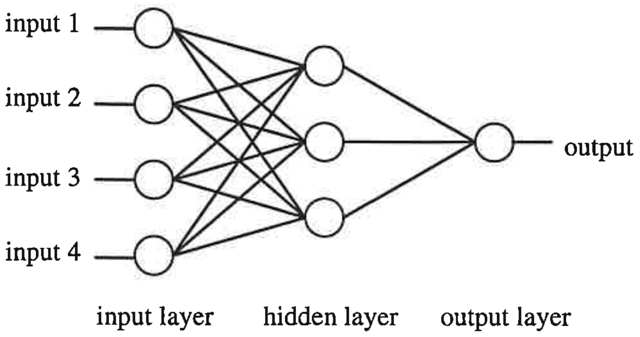

Figure 1. A fully connected backpropagation neural network

Many methods exist for determining the weights of, i.e., training, a NN. Usually the training is supervised, denoting that the output associated with each input pattern in the training set is known beforehand. Typically the training of a network is started by defining a network with an arbitrary – but small – number of hidden layers, an arbitrary number of neurons in each hidden layer, fixed connection topology – full connectivity is typically used, and randomly selected values for the weights. The training set is then presented to the network and the networks response to each pattern is compared to the desired response. Mean square error (MSE) and mean relative error (MRE) are used as measures in judging the overall performance of the network (Karunanithi *et al.* 1994). In back-propagation algorithm (see, e.g., Rumelhart *et al.* 1986) a gradient descent procedure is utilized to iteratively update the values of the weights:

$$\Delta w_{ji}(s+1) = -\eta \delta_j x_i + \alpha \Delta w_{ji}(s) \tag{1}$$

where w_{ji} is the interconnecting weight between neuron *j* in the hidden neuron and neuron *i* in the input layer

s is the iteration sweep on the training set

η is the learning factor, a value of 0.9 was used in this study

δ_j is a factor whose value depends on whether neuron *j* is an

output neuron or a hidden neuron
x_i is the input value
α is the momentum factor, a value of 0 was used in this study

For neurons in the output layer

$$\delta_j = \left(\frac{df}{d\,\mathrm{net}_j}\right)\left[y_j^{(t)} - y_j\right] \tag{2}$$

where f is the activation function
net_j is the activation value
$y_j^{(t)}$ is the desired response
y_j is the actual response

and for neurons in the hidden layer

$$\delta_j = \left(\frac{df}{d\,\mathrm{net}_j}\right)\sum_{i=1}^{q} w_{ij}\delta_i \tag{3}$$

where q is the number of neurons in the output layer

The algorithm is stopped when the total error is deemed sufficiently low. At that point the NN can be said to have learnt the training set.

Following a good modeling practice only a portion of the training set is used for determining the weights and the rest is used to validate the resulting NN forecasting model. The network's capability to predict the output of a previously unseen input pattern and the accuracy of the prediction can be assessed by observing the training and the verification period. A NN can also overtrain, i.e., learn the training data set better than the underlying general problem. An overtrained network performs badly outside of its training set, i.e., it does not have the ability to generalise. A symptom of overtraining is, e.g., an increase in the MSE or MRE during training. If overtraining is detected, the training should be stopped.

The distribution of the training patterns within the whole input space can have an important effect on the learning and generalization capability of a network. The extrapolation capabilities of NNs are usually not good and thus the quality of the training pattern is important. This experience suggests that a NN may not perform well in changing and/or new hydrological conditions.

There are no real rules to determine the structure, number of the hidden layers, number the neurons in the layers, the connection topology, of a NN from the problem posed to it – remember that the back-propagation algorithm only adjusts the weights. One indication for the modeling are the dimensions of the input and output data. Also, in a general sense, the longer the lag within the system and/or the more complex or far-away relation between the input and output variables, the more complex the structure should be. A trade-off exists in the deciding on the structure: a too complex NN may overtrain more easily while a too simple NN may not learn the necessary characteristics of the problem. The development and use of a NN thus requires experience and experimentation.

TIME-SERIES MODELS

The performance of a NN inflow prediction model was compared to three types of time-series models:

Type I: A periodic autoregressive (PAR) model
Type II: An autoregressive – moving average (ARMA) model with deseasonalization
Type III: A seasonal autoregressive integrated moving average (SARIMA) model

These models are described here only briefly. A detailed description is given, e.g., by Noakes *et al.* (1985).

A PAR model

$$\phi_m(B)^p\left(Z_t^\lambda - \mu_m\right) = a_t \tag{4}$$

where $\phi_m(B)^p$ is the periodic autoregressive polynomial
p is the degree of the autoregressive polynomial
B is the backward shift operator: $B^j X_t = X_{t-j}$
Z_t is the time-series value
λ is the Box-Cox transformation parameter
μ_m is the periodic mean inflow

a_t is the noise term
t is the time step, t=1,…,(N-1)s
m is the period, m=1, …,s
N is the number of years
s is the number of periods within a year

assumes a period-dependent mean, variance, and autocorrelation function. The polynomial $\phi(B)^p$ is defined as $\phi(B)^p = 1 - \phi_1 B^1 - \ldots - \phi_p B^p$.

A deseasonalization equation

$$W_t = \frac{Z_t^\lambda - \mu_m}{\sigma_m} \qquad (5)$$

where W_t is the deseasonalized time-series value
σ_m is the periodic standard deviation

was applied to the inflow data before fitting an ARMA model

$$\phi(B)^p W_t = \theta(B)^q a_t \qquad (6)$$

where $\phi(B)^{\mathrm{p}}$ is the autoregressive polynomial
$\theta(B)^{\mathrm{q}}$ is the moving average polynomial
q is the degree of the moving average polynomial

to the inflow time-series.

Differentiation

$$Y_t = (1 - B)^d (1 - B^s)^D (Z_t^\lambda - \mu) \qquad (7)$$

where d is the degree of the differentiation polynomial
D is the degree of the seasonal differentiation polynomial

was applied to the data before a SARIMA model

$$\phi(B)^p \Phi(B^s)^P Y_t = \theta(B)^q \Theta(B^s)^Q a_t \qquad (8)$$

where $\Phi(B^s)$ is the seasonal autoregressive polynomial
P is the degree of the seasonal autoregressive polynomial
$\Theta(B^s)$ is the seasonal moving average polynomial
Q is the degree of the seasonal moving average polynomial
Y_t is the differenced process

was fitted to it.

The time-series models where fitted by standard methods using autocorrelation and partial autocorrelation functions. Parameters were estimated using least squares method with the PAR model and maximum likelihood method with the ARMA and SARIMA models.

RESULTS

Different input variable combinations were tested for the NN model. Of these the one with five inputs, viz: present inflow, gradient of the inflow timeseries, present discharges from the small basins n:o 41 and n:o 71 and Julian day, performed best. Models which used other variables, i.e. air temperature and areal SWE performed clearly poorer.

Several structures were tested with the NN and of these the one with one hidden layer and with ten neurons performed best. The results of the training / parameter estimation (one-step-ahead predictions) are shown in the Figure 2.

A comparison between the one-step-ahead predictions generated using the best NN and the time-series models is shown in the Figure 3.

As can be seen from the Table 1. the time-series model of type II (the ARMA model with deseasonalization) performed best among the time-series models. The NN model performed as well as the best time-series model (model II) but better than the other time-series models (models I and III).

DISCUSSION AND CONCLUSION

The NN model which was developed is not completely comparable with the time-series models since it used external data while the time-series models did not. The reason why the SWE did not perform well as an input variable to the NN model was assumably due to the fact that the model was trained to predict only one step ahead. The results suggest that the ARMA model with deseasonalization is the best way to model an inflow time-series.

Table 1. The MSE in training / parameter estimation and in the validation periods in different cases.

The model	*MSE in the training / parameter estimation period*	*MSE in validation period*
Time-series type I	37	37
Time-series type II	11	12
Time-series type III	20	82
NN	11	12

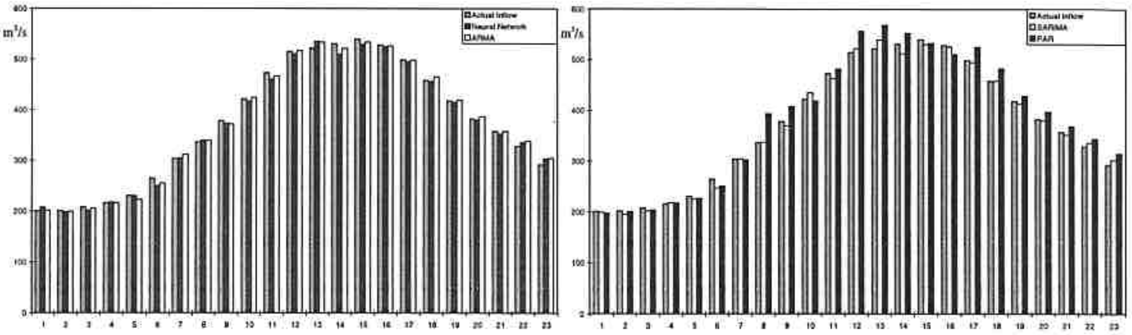

Figure 2. Comparisons of the one-step-ahead predictions in the training/parameter estimation period. The x-axis starts at March 8.1967. The figure on the left compares NN (black bars) and ARMA (white bars) to the actual inflow (grey bars) and the figure on the right compares PAR (black bars) and SARIMA (white bars) to the actual inflow (grey bars). The scale of the y-axis is from zero to 600 m^3/s.

This preliminary study indicates that it is possible to use artificial neural networks succesfully in inflow forecasting. Due to the smooth behavior of the inflow time-series in this case, a short-term forecasting model was relatively easy to develop and this seems to be one of its main benefits. The results seem to indicate, that NNs are comparable to some advanced time-series models in inflow forecasting.

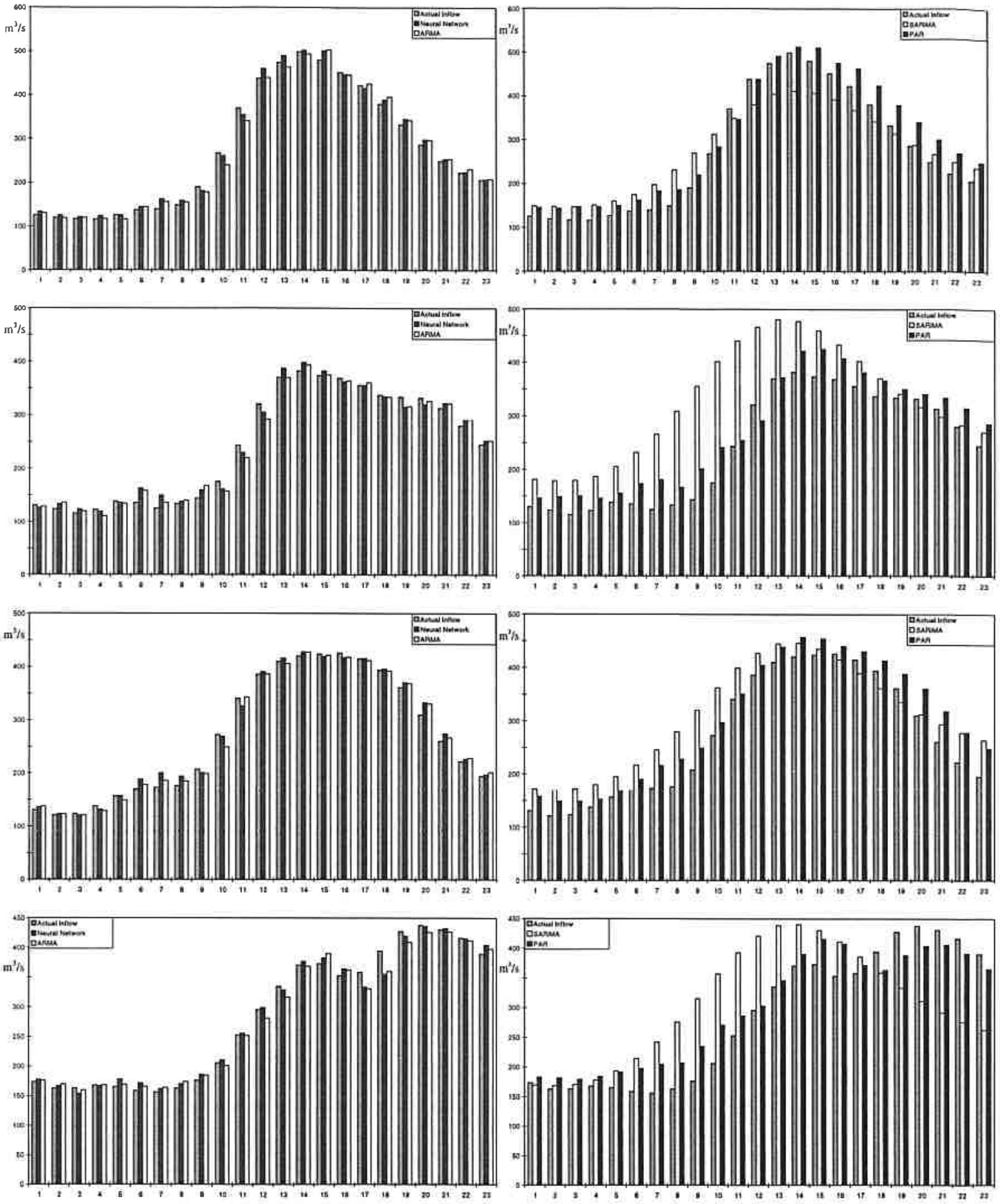

Figure 3. Successive one-step-ahead predictions in years 1970, 1985, 1986, and 1987 respectively. The x-axis starts each year at March 8th. The figure on the left compares NN (black bars) and ARMA (white bars) to the actual inflow (grey bars) and the figure on the right compares PAR (black bars) and SARIMA (white bars) to the actual inflow (grey bars). The scales of the y-axes are from zero to 600, 500, 500, and 450 m^3/s respectively.

REFERENCES

Bergström, S. 1976. Development and application of a conceptual runoff model for Scandinavian catchments. SMHI. RH7. Swedish Meteorological and Hydrological Institute, Stockholm, Sweden.

Daniell, T.M. 1991. Neural networks - applications in hydrology and water resources engineering. In: Proceedings of an International Hydrology and Water Resources Symposium. Perth, Australia. pp. 797–802.

Härkönen. P. 1995. Forecasting of freshwater phytoplankton by artificial neural networks. Masters' thesis. Helsinki University of Technology, Espoo, Finland.

Karunanithi, N., Grenney, W.J. Whitley, D., and Bovee, K. 1994. Neural networks for river flow prediction. J. Comp. in Civ. Engrg., ASCE, 8(2), 201-220.

Karvonen, T. 1984. Real-time flood forecasting using extented Kalman filter. Licentiate thesis. Helsinki University of Technology, Espoo, Finland.

Kärkkäinen, K. 1997. Forecasting inflows and optimizing outflows with neural networks. Masters' thesis. Helsinki University of Technology, Espoo, Finland. (In finnish)

Leppäjärvi, R. 1992. Hydrological yearbook 1992. National Board of Waters and the Environment, Helsinki, Finland.

Malve, O. 1986. The use of conceptual rainfall-runoff models and time series models in real-time flood forecasting. Master thesis. Helsinki University of Technology, Espoo, Finland. (in Finnish)

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. 1986. Learning internal representations by error propagation. In: Rumelhart, D.E., and McClelland, J.L. (Eds). Parallel distributed processing: explorations in the microstructure of cognition. Vol. 1. MIT, Cambridge, MA, USA.

Sirviö, H. 1998. Unpublished manuscript. (In finnish)

Vehviläinen, B. 1994. The watershed simulation and forecasting system in the National Board of Waters and the Environment. In: Publications of the Water and Environment Research Institute, 17. National Board of Waters and the Environment, Helsinki, Finland. pp 3-16.