Global Nest: the Int. J. Vol 6, No 3, pp 205-211, 2004
Copyright© 2004 GLOBAL NEST
Printed in Greece. All rights reserved

APPLICATION OF ARTIFICIAL NEURAL NETWORKS FOR FLOOD FORECASTING

D.F. LEKKAS[1,*]
C. ONOF[1]
M. J. LEE[1]
E.A. BALTAS[2]

[1]Department of Civil and
Environmental Engineering,
Imperial College of Science,
Technology and Medicine,
London, SW7 2BU
[2]School of Agriculture
Aristotle University of Thessaloniki, Greece

Received: 5/1/05
Accepted: 20/1/05

to whom all correspondence should be addressed:
e-mail: d.lekkas@env.aegean.gr

ABSTRACT

In hydrology, as in a number of diverse fields, there has been an increasing use of Artificial Neural Networks (ANN) as black-box simplified models. This is mainly justified by their ability to model complex non-linear patterns; in addition they can self-adjust and produce a consistent response when 'trained' using observed outputs.

This paper utilises various types of ANNs in an attempt to assess the relative performance of existing models. Ali Efenti, a subcatchment of the river Pinios (Greece), is examined and the results support the hypothesis that ANNs can produce qualitative forecasts. A 7-hour ahead forecast in particular proves to be of fairly high precision, especially when an error prediction technique is introduced to the ANN models.

KEYWORDS: Artificial Neural Networks, Real-Time Flood Forecasting, River Pinios.

INTRODUCTION

The temporal and spatial variability that characterises a river system makes flow forecasting a very demanding task. Flow forecasting is a crucial part of flow regulation and water resources management, as it is related to issues such as drought prevention, flood forecasting for dam and human safety and ecosystem sustainability. As it is reported, floods and droughts kill more people and cause more damage than any other natural disaster (WMO, 92). Consequently, there is a need for systems capable of efficiently forecasting water levels or discharge rates in rivers.

Previous work has been supportive of the neural networks for flow forecasting; (Imrie *et al.*, 2000; Minns and Hall, 1996; Minns and Hall, 1997; Dawson and Wilby, 1998; Dawson and Wilby, 1999; Campolo *et al.*, 1999; Liong *et al.*, 2000). Artificial neural networks provide a fast and flexible means for developing non-linear flow routing models. However, it has been found in previous studies (Minns and Hall, 1996; See *et al.*, 1997; Dawson and Wilby, 1998; Campolo *et al.*, 1999) that since those networks perform poorly outside the calibration range, they cannot be reliably used in situations where significant events outside the calibration range are important. Obviously, flood forecasting is one such application since we are often interested in the extremes and are regularly faced with a limited amount of calibration data.

Most of the previous work considered complicated models and large networks in order to represent the hydrological system as precisely as possible; these models have the advantage of a good calibration performance, but compromise with a poor performance in verification data, as they fail to generalise.

ANN models are considered as they can be simple (small networks) yet still keep the non-linear characteristics required to predict the river flow. Comparisons are made between the performance of different ANN structures. Furthermore, a real-time updating technique, an important component of a flood forecasting system in our opinion, is applied in order to demonstrate the operational performance benefits of real-time updating. In the next section a short presentation of ANN methodology is given, focusing on some key problems and suggesting methods to that can be used to overcome them.

ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are a type of parallel computer structure, within which a number of processing units are linked together so that the computer's memory is distributed and information is passed in a parallel manner. A large number of ANN architectures and algorithms have been developed so far, multi-layer feedforward networks (Rumelhart *et al.*, 1986), self-organising feature maps (Kohonen, 1982), Hopfield networks (Hopfield, 1987), counterpropagation networks (Hecht-Nielsen, 1987), radial basis function networks (Powell, 1987) and recurrent ANNs (Elman network; Elman, 1988). Of these networks, the most commonly used are feedforward networks and radial basis function networks (Karunanithi, *et al.*, 1994; Bishop, 1995). Multi-layer feedforward networks have been found to perform best when used in hydrological applications (Hsu *et al.*, 1995; Dauson and Wilby, 1999) and as such they are by far the most commonly used (Maier and Dandy, 2000). The attempt to choose between different methods and define which is the superior, is likely to fail as in most cases the choice should be "application oriented". It is preferable for every new application to test different types of ANNs rather than use a pre-selected one.

Training

To train an ANN, the following procedure is generally applied. Training data patterns are fed sequentially into the input layer, and this information is propagated through the network. The resulting output predictions $y_j(t)$ are compared with a corresponding desired or actual output, $d_j(t)$. The mean squared error at any time t, $E(t)$, may be calculated over the entire data set using Equation 1. The intermediate weights are adjusted using an appropriate learning rule until $E(t)$ has decayed sufficiently.

$$MSE(t) = \frac{1}{2}\sum_{j=1}^{n}(y_j(t) - d_j(t))^2 \qquad (1)$$

A wide range of training algorithms has been developed to achieve optimum model performance. For feedforward ANNs, the error backpropagation algorithm with the gradient descent update rule (Rumelhart *et al.*, 1986) is most commonly employed. However, there are a number of inconvenient drawbacks associated with the use of this algorithm. For example, prior to ANN training it is necessary to specify the network architecture, that is, the number and configuration of its hidden units.

The learning ability and performance of an ANN model depends on the suitability of its architecture. If the network is too small, it may have insufficient degrees of freedom to fully capture all the underlying relationships in the data. Conversely, if the network is too large, it may fail to generalise, memorising events in the training data that are not necessarily representative of the system under consideration. To overcome this type of dilemma a simplistic but effective method should be applied, as it is presented in the paragraph below that will give an unbiased solution (in this case the correct ANN architecture).

Generalisability

The main concern when referring to ANNs is whether the trained network will generalise when presented with new data and especially when these data are outside the range of the calibration data. When a network is trained repeatedly in order to improve its performance on the training data, there is the possibility that the network will finally "memorise" the training samples and not 'learn' the underlining pattern. This is called overfitting. This is more likely to happen to networks with a large number of processing units and results in poor generalisability. The ability to generalise, that is to produce outputs from unknown inputs, is very important when the neural network is used in applications like real-time flow forecasting.

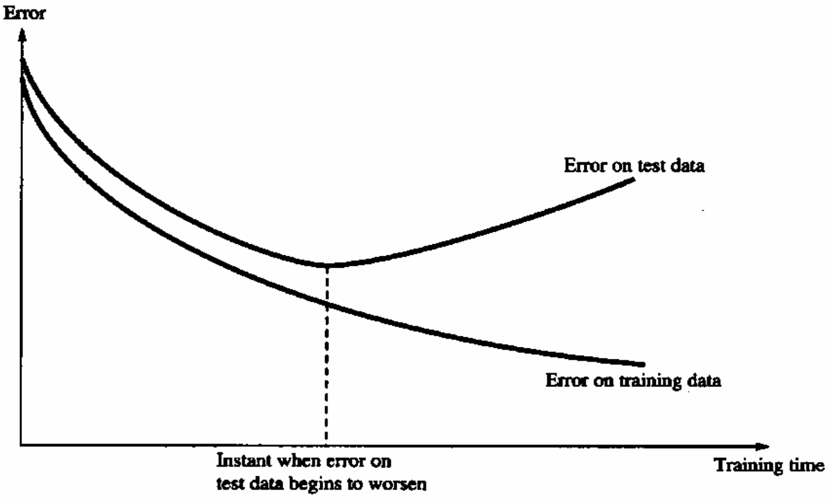

Figure 1. Change in error with training time, on training set and test set (Mehrota et al., 1997)

For this reason networks with, fewer parameters are preferred, just enough to provide an adequate fit in order to avoid over-training.

One solution that is suggested, is to use an extra set of data (together with the training set and the validation set), a test set, and monitor the performance of the network on this data set. (Mehrota *et al.*, 1997). The network parameters (weights and biases) should be adjusted only on the account of the training set, but the error should be monitored on the test set. The error on the test data will normally decrease during the initial iterations together with the error on the training set. However, when the network begins to overfit the data, the error on the test set will begin to rise, as presented in Figure 1. The training is then stopped, and the weights and biases producing the minimum error on the test set are returned. This is also called ‘early stopping’.

This method can be generalised and introduced as well for the identification of the appropriate network architecture (number layers and number of neurons per layer). This technique is applied in the case study presented below.

Another method for improving generalisation, which involves modifying the performance function, is called regularisation. It is possible to improve generalisation by adding a term in the performance function (MSE) that consists of the mean of the sum of squares of the network weights and biases MSW :

$$MSE_{impr} = \gamma MSE + (1-\gamma)MSW \qquad (2)$$

where MSE_{impr} is the improved mean square error, γ is the performance ratio,

$$MSW = \frac{1}{N}\sum_{j=1}^{N} w_j^2 \qquad (3)$$

where w_j is a weight or bias in the *j*th layer.

This function will cause the network to develop smaller weights and biases and will force the network response to be smoother and less likely to overfit.

There are certain applications in which it is better to converge more slowly, when for example early stopping is used. Inconsistent results are likely to be obtained if an algorithm that converges too quickly is applied.

River flow prediction

River flow data from two gauging stations, Pili and Ali Efenti, located within the catchment of Ali Efenti, subcatchment of the River Pinios in Greece are used. These are half-hourly data and they cover three separate periods: 30/11/90-25/12/90 (26 days), 5/2/91-28/2/91 (23 days) and 1/4/92-11/5/92 (42 days). The data were obtained from National Data Bank of Hydrological & Meteorological Information of Greece. The size of the catchment upstream of Ali Efenti is 2763 km^2 and the distance between the two gauging stations is 55 Km. The periods presented in Figure 2 were used for calibration, testing and verification, respectively.

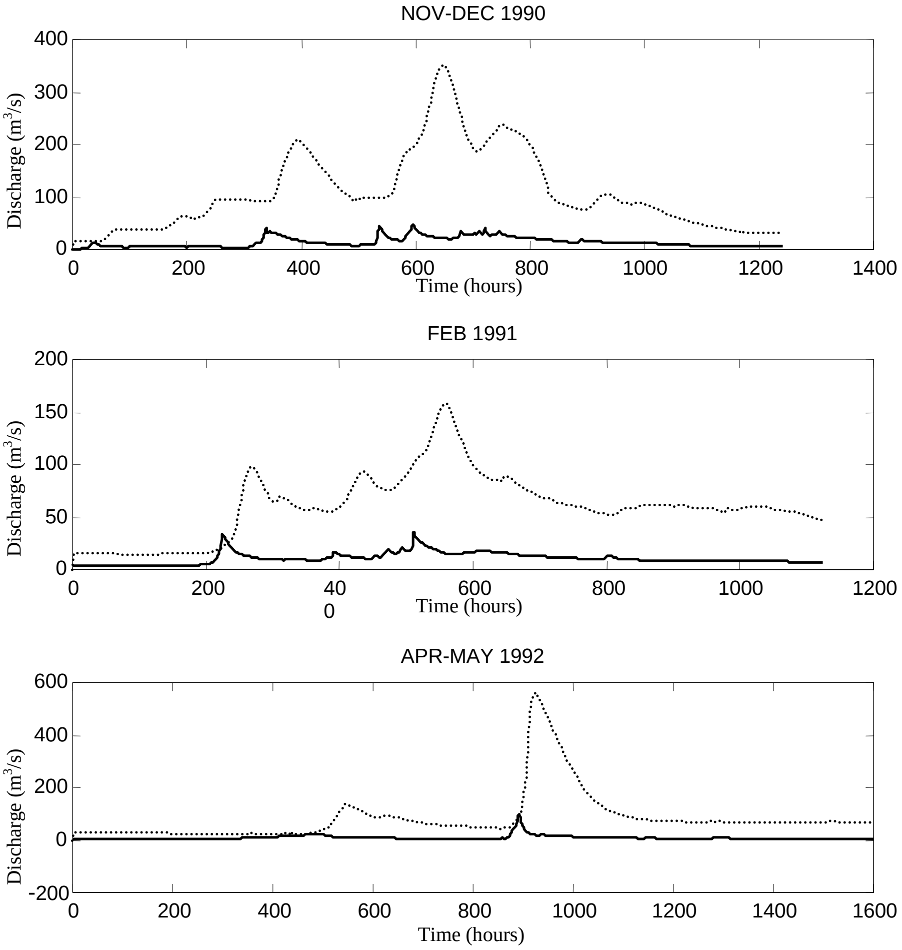

Figure 2. Discharge data record at, Pili (solid line) and Ali Efenti (dotted line)

The reason these periods were chosen was to produce more general models. First a medium flow period was used for training, then a low flow period was used for testing and finally a high flow period, with values over those in the previous data sets, was used to validate the ability of the produced ANN to generalise. The aim was to create models that could forecast the flow with a lead-time of 7 hours that can be used for operational applications.

A correlation analysis was performed on the data, to identify suitable lags to be applied to the upstream gauging station (Pili) time series, in order to form the ANN's input patterns.

For the application of ANN, the data were scaled so that the training and verification record lay within the range of [-1 1]. The input is then introduced to the input layer of each of the ANN with a 14-hour lag.

Three ANN models were developed, based upon the input data described above. All the ANNs incorporated linear activation functions at the output layer.

The first type was a traditional feed-forward backpropagation ANN trained with the gradient descent method. The backpropagation ANN (BP-ANN) had two hidden layers with 1 neuron in each layer. The option of using a BP ANN with linear transfer functions (positive linear) in the hidden layer, was also tested.

The second type was an Adaptive Linear Neuron Network (ADALINE). This type of ANN has the advantage of using only linear transfer functions, which lead to a single error minimum, making parameter estimation easier. The ADALINE used had a single hidden layer combined with a tapped delay line with delays ranging from 14-25h. The network once initialised and operating adapts at each time step to minimise the error. In training mode, the network is set to pass 10 times through the input sequence.

The third type was an Elman network. These networks are two-layer backpropagation networks, with an additional feedback connection (recurrent) from the output of the hidden layer to its input. This feedback mechanism works as temporal memory helping Elman networks to learn, recognise and generate time varying patterns. The network used in this case study had one hidden layer with 15 neurons.

The Error Prediction method was used as an updating technique, as it is simple, robust and can be used together with any type of model, as it is a totally separate procedure. With error prediction, the difference (error) between the observed flow and the model output is modelled using an ARMA model. In the verification mode, this ARMA model is used to forecast the error and is afterwards added to the flow forecast in order to correct it.

RESULTS AND DISCUSSION

The overall performance of each model obtained was judged with respect to the verification data on the basis of the coefficient of efficiency, R^2, defined as follows:

$$R^2 = 1 - \frac{\sum_{p=i}^{n} (y_p - d_p)^2}{\sum_{p=i}^{n} (d_p - \overline{d})^2} \qquad (4)$$

where y_p, and d_p are the model predictions and target values for each pattern (sample) p respectively, and $\overline{d}$ is the mean target output.

The results obtained using each of the ANN methods over the verification period are presented in Table 1. It can be seen that the simple BP network with linear activation function (poslin), provides the poorest predictions of all the models in calibration, but still manages to generalise better than the non-linear BP (logsig). However, when the error prediction method is used in conjunction with the ANN models, the model performance assessed by the coefficient of efficiency, increases significantly from 0.455 to 0.895 and from 0.412 to 0.889 respectively.

In comparison with the Feed forward ANN models both the non-linear ELMAN and the adaptive ADALINE, provide considerably better flow forecasts. As it is presented in Figure 3 the BP network, (that applies for both transfer functions), fails to generalise and predict the river flow above a certain level.

The recurrent ANN model (ELMAN) provides reasonable forecasts, better than any other ANN (Table1). The recurrent term seems to improve the network’s performance. When the updating method is introduced, the performance of all the tested models is improved. Error prediction performs extremely well, correcting the model output significantly (Table 1). The updated model output over the verification period (APR-MAY 1992) producing a flow forecast at Ali Efenti that is plotted in Figure 4. As can be seen from Figure 4 and Table 1, both the ADALINE and ELMAN models perform well, with the recurrent ANN model performing marginally better. This may be attributed to its greater non-linearity and the information introduced to the ANN by the recurrent (feed-back) process. The

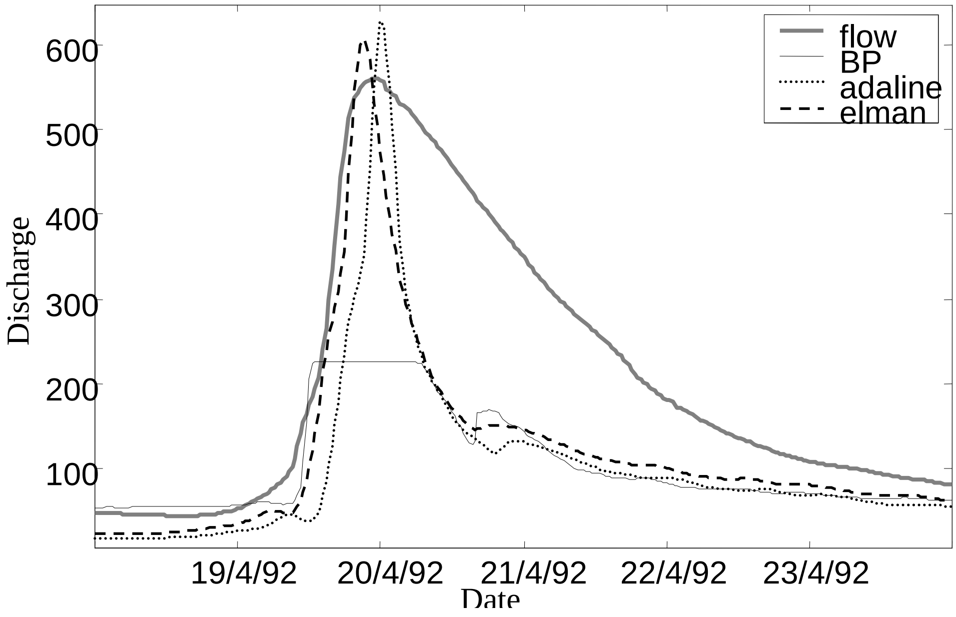

Figure 3. Flow forecast results at Ali Efenti with a lead-time of 7 hr

Table 1. Results obtained for 7-hour flow forecasts with different types of ANN

Type	Calibration	Testing	Verification	Updated
FF1,1,1 (logsig)	0.737	0.414	0.412	0.889
FF1,1,1 (poslin)	0.709	0.531	0.455	0.895
ADALINE	0.722	0.766	0.598	0.871
ELMAN	0.723	0.674	0.622	0.927

FF(logsig):Feed forward back bropagation NN with Log sigmoid transfer function
FF(poslin):Feed forward back bropagation NN with positive linear transfer function

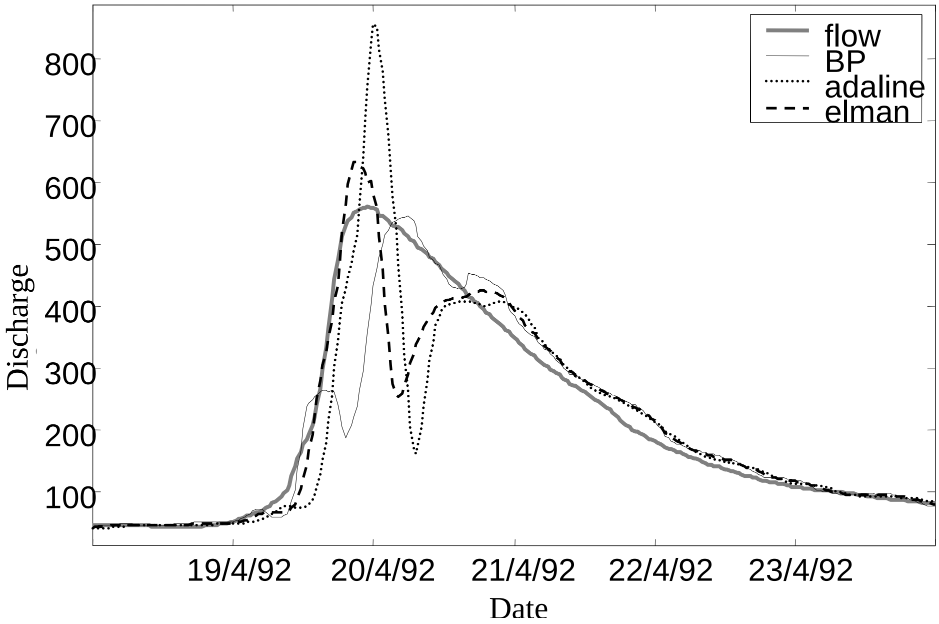

Figure 4. Updated flow forecast at Ali Efenti with a lead-time of 7h.

updating technique is able to correct the timing of the rising limb and the recession to an extent that is defined by the performance of the simulation model. The instability (double peaks), shown in Figure 4 of the updated models, when compared to non-updated forecasts, can be attributed to the long lead time of the updating procedure (7hrs: 14 half hour steps).

CONCLUSIONS

A multilayer backpropagation network with enough processing units can reproduce just about any function. However backpropagation will not always find the correct weights and biases for the optimum solution. Building a network for an application requires experience and use of suitable methods, such as the simple early stopping techniques presented above As the initial values in optimisation algorithms are very important, it is advisable to re-initialise the network several times to guarantee that the best solution is obtained.

The case study presented above shows that ANNs, as also suggested by other authors, can be used for flow forecasting. The main concern is whether the trained model will be able to generalise. As shown earlier, ANNs models that are trained using a simple method like early stopping, that forces them be simple and general (with few hidden layers and neurons), can predict the flow regime sufficiently. When coupled together with an updating technique, ANNs can form a reliable flow forecasting system, which can be used as a tool for water resource management.

In real-time applications, like flow regulation and flood forecasting where the precision and modelling speed are crucial, black box models and signal processing techniques need to be implemented. As is presented by this case study, error prediction manages to correct the model output and produce an excellent forecast. Even if error prediction has no physical interpretation, it would be wrong not to use it in real-life applications where issues like human safety and economic disaster are concerned.

REFERENCES

Bishop C.M. (1995),Neural Networks for Pattern Recognition, Clarendon Press, Oxford.

Campolo M., Andreussi P. and Soldati A., (999), River flood forecasting with a neural network model, *Water resources research*, **35**, 1191-1197.

Dawson C.W. and Wilby R. (1998) An artificial neural network approach to rainfall-runoff modelling, *Hydrolog. Sci. J.*, **43**, 47-66.

Dawson C.W. and Wilby R.L. (1999), A comparison of artificial neural networks used for river flow forecasting, *Hydrol.Earth Syst. Sc.*, **3**, 529-540.

Elman J.L. (1988), Finding structure in time, In: *CRL Technical Report 8801*, University of California at San Diego, Centre for Research in Language.

Hecht-Nielsen R. (1987) Counterpropagation networks, *Appl. Optics*, **26**, 4979-4984.

Hopfield J.J. (1987), Learning algorithms and probability distributions in feed-forward and feed-back networks, *P. Natl. Acad. Sci. USA*, **84**, 8429-8433.

Hsu K., Gupta H.V. and Sorooshian S. (1995), Artificial neural network modelling of rainfall-runoff process, *Water Resources Research*, **31**, 2517-2530.

Imrie C.E., Durucan S. and Korre A. (2000a), River flow prediction using artificial neural networks: generalisation beyond the calibration range, *J. Hydrol.*, **233**, 138-153.

Karunanithi N., Grenney W.J. Whitley D. and Bovee K. (1994), Neural Networks for River Flow Prediction, *J. Comput. Civil Eng.*, **8**, 201-219.

Kohonen T. (1982), Self-organized formation of topologically correct feature maps, *Biol. Cybern.*, **43**, 59-69.

Liong S.Y., Lim W.H. and Paudyal G.N. (2000), River stage forecasting in Bangladesh: neural network approach, *J. Comput. Civil Eng.*, **14**, 1-8.

Maier H.R. and Dandy G.C. (2000), Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, *Environ. Modell. Softw.*, **15**, 101-124.

Mehrota K., Mohan C.K. and Ranka S. (1997), Elements of Artificial Neural Networks, In: *The MIT Press*, Cambridge, Massachusetts.

Minns A.W. and Hall M.J. (1996) Artificial neural networks as rainfall-runoff models, *J. Hydrolog. Sci.*, **41**, 399-417.

Minns A.W. and Hall M.J. (1997), Living with the ultimate black box: more on artificial neural networks, In: *BHS 6th National Hydrology Symposium*, Salford, UK, pp 9.45-9.49.

Powell M.J.D. (1987), Radial basis functions for multivariable interpolation: a review, In: *Algorithms for Approximation*, J.C. Mason and M.G. Cox (eds.), Clarendon Press, Oxford, pp143-167.

Rumelhart D.E., Hinton E. and Williams J. (1986), Learning internal representation by error propagation, *Parallel Distributed Processing*, **1**, 318-362.

See L., Corne S., Dougherty M. and Openshaw S. (1997), Some initial experiments with neural network models of flood forecasting on the River Ouse, In: *Proceedings of the 2nd International Conference on GeoComputation*, Dunedin, New Zealand.

WMO (1992), Simulated real-time intercomparison of hydrological models, In: *WMO operational hydrology report*, no 38, WMO no 779, World Meteorological Organisation, Geneva.