Hydrological Sciences Journal

ISSN: 0262-6667 (Print) 2150-3435 (Online) Journal homepage: www.tandfonline.com/journals/thsj20

An artificial neural network approach to rainfall-runoff modelling

CHRISTIAN W. DAWSON & ROBERT WILBY

To cite this article: CHRISTIAN W. DAWSON & ROBERT WILBY (1998) An artificial neural network approach to rainfall-runoff modelling, Hydrological Sciences Journal, 43:1, 47-66, DOI: 10.1080/02626669809492102

To link to this article: https://doi.org/10.1080/02626669809492102

Published online: 25 Dec 2009.

Submit your article to this journal

Article views: 6196

View related articles

Citing articles: 83 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=thsj20

An artificial neural network approach to rainfall–runoff modelling

CHRISTIAN W. DAWSON
School of Mathematics and Computing, University of Derby, Kedleston Road, Derby DE22 1GB, UK

ROBERT WILBY
School of Environmental and Applied Science, University of Derby, Kedleston Road, Derby DE22 1GB, UK

Abstract This paper provides a discussion of the development and application of Artificial Neural Networks (ANNs) to flow forecasting in two flood-prone UK catchments using real hydrometric data. Given relatively brief calibration data sets it was possible to construct robust models of 15-min flows with six hour lead times for the Rivers Amber and Mole. Comparisons were made between the performance of the ANN and those of conventional flood forecasting systems. The results obtained for validation forecasts were of comparable quality to those obtained from operational systems for the River Amber. The ability of the ANN to cope with missing data and to "learn" from the event currently being forecast in real time makes it an appealing alternative to conventional lumped or semi-distributed flood forecasting models. However, further research is required to determine the optimum ANN training period for a given catchment, season and hydrological contexts.

Une approche de la modélisation pluie–debit par des réseaux neuronaux artificiels

Résumé Ce document traite du développement et de l'application des réseaux neuronaux artificiels (RNA) à la prévision des débits de deux bassins versants du Royaume Uni sujets aux inondations grâce à l'utilisation de données hydrométriques réelles. Partant d'un ensemble restreint de données d'apprentissage, il a été possible de réaliser des modèles pour la prévision des débits au pas de temps de 15 min à échéance de 6 heures pour les rivières Amber et Mole. On a comparé les performances des RNA et des systèmes conventionnels d'annonce de crue. Les résultats obtenus lors de la validation des prévisions des RNA étaient de qualité comparable à ceux obtenus par les systèmes actuellement utilisés opérationnellement sur la Rivière Amber. La capacité des RNA à gérer les données manquantes et à "apprendre" en temps réel à partir de l'évènement en cours, fait de ces outils une alternative séduisante aux actuels modèles de prévision agrégés ou semi-distribués. De plus amples recherches sont cependant nécessaires pour déterminer la période d'apprentissage optimale des RNA pour un bassin donné et selon le contexte climatique et hydrologique.

INTRODUCTION

The United Nations General Assembly declared the 1990s to be the International Decade for Natural Disaster Reduction with the specific intent to "disseminate existing and new information related to measures for the assessment, prediction, prevention and mitigation of natural disasters" (WMO, 1992). A prominent element

Open for discussion until 1 August 1998

within this programme has been the development of operational flood forecasting systems. These systems have evolved through advances in mathematical modelling (Wood & O'Connell, 1985; O'Connell, 1991; Lamberti & Pilati, 1996), the installation of telemetry and field monitoring equipment at critical sites in drainage networks (Alexander, 1991), through satellite and radar sensing of extreme rainfalls (Collier, 1991), and through the coupling of precipitation and runoff models (Georgakakos & Foufoula-Georgiou, 1991; Franchini *et al.*, 1996). However, in practice, successful real-time flood forecasting often depends on the efficient integration of all these separate activities (Douglas & Dobson, 1987). Under the auspices of the World Meteorological Organization (WMO, 1992) a series of projects were implemented to compare the characteristics and performance of various operational models and their updating procedures. A major conclusion of the most recent intercomparison exercise was the need for robust simulation models in order to achieve consistently better results for longer lead times even when accompanied by an efficient updating procedure.

The attractiveness of Artificial Neural Networks (ANNs) to flood forecasting is threefold. Firstly, ANNs can represent any arbitrary nonlinear function given sufficient complexity of the trained network (see below). Secondly, ANNs can find relationships between different input samples and, if necessary, can group samples in analogous fashion to cluster analysis. Finally, and perhaps most importantly, ANNs are able to generalize a relationship from small subsets of data whilst remaining relatively robust in the presence of noisy or missing inputs, and can adapt or learn in response to changing environments. However, despite these potential advantages, ANNs have found rather limited application in hydrology and related disciplines. For example, French *et al.* (1992) used a neural network to forecast rainfall intensity fields in space and time, whilst Raman & Sunilkumar (1995) used an ANN to synthesize reservoir inflow series for two sites in the Bharathapuzha basin, South India. Similarly, Hewitson & Crane (1994) described a range of climatological ANN applications such as snowfall prediction, classifying arctic cloud and sea ice, precipitation and, more recently, climate change impacts modelling (Hewitson & Crane, 1996).

However, the use of artificial neural networks for flood forecasting is an area which has yet to be fully explored (Cheng & Noguchi, 1996). Up until now the majority of work in this area has been mainly theoretical, concentrating on neural network performance with artificially generated rainfall–runoff data (Minns & Hall, 1996). However, these theoretical approaches tend to overlook the difficulty in converting and applying actual data to artificial neural network topologies. Hall & Minns (1993) go some way to address this criticism by applying neural networks to a small urban catchment area. However, their discussion is limited to the performance of a neural network on a small number of events.

This paper goes one stage further by discussing how artificial neural networks may be developed and used on "real" hydrological data. It discusses the problems that need to be addressed when applying neural networks to rainfall–runoff modelling and demonstrates the effectiveness of artificial neural networks in this particular domain. By applying a neural network to flood simulation in two UK catchments, the prospects for the use of ANNs in real-time flood forecasting are evaluated. Finally, suggestions are made concerning necessary refinements to the existing ANN prior to transfer to operational use.

ARTIFICIAL NEURAL NETWORKS

In this section a basic overview of artificial neural networks is provided. More thorough discussions are given in texts such as Wasserman (1989) and Gallant (1993).

Overview

Figure 1 provides an overview of ANN topology. A network is made up of a number of interconnected nodes (called neurons) arranged into three basic layers—*input*, *hidden* and *output* (there are variations on this topology but these are beyond the scope of this paper). The input nodes in this representation perform no computation but are used to distribute inputs into the network. This kind of network is called a feed forward network as information passes one way through the network from the input layer, through the hidden layer and finally to the output layer. Recurrent networks, such as Hopfield nets (Hopfield, 1982, 1984), allow feedback between layers.

The number of input nodes, N, and the number of output nodes, M, in an ANN are dependent on the problem to which the network is being applied. Unfortunately, there are no fixed rules as to how many nodes should be included in the hidden layer. If there are too few nodes in the hidden layer the network may have difficulty generalizing to problems it has never encountered before. On the other hand, if there are too many nodes in the hidden layer, the network may take an unacceptably long time to learn anything of any value. Different numbers of hidden nodes were used in the networks developed in this study for rainfall–runoff modelling. The best results are presented later.

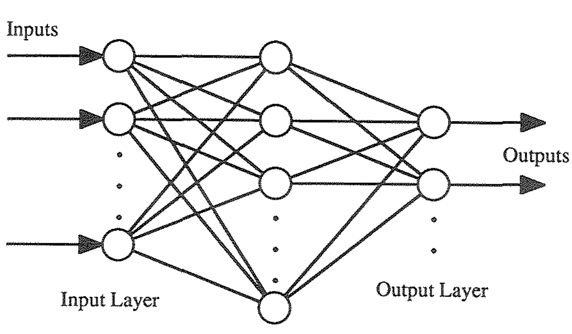

Fig. 1 A basic overview of artificial neural network topology.

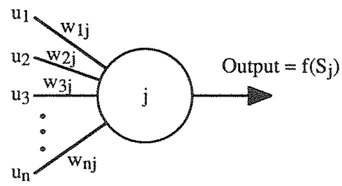

Fig. 2 An artificial neuron.

Figure 2 provides a closer look at an individual neuron (in the hidden and output layers). Each neuron, j, has a number of input arcs, u_1 to u_n. Associated with each arc, i, is a weight, w_{ij}, which represents a factor by which any values passing into the neuron are multiplied. A neuron, j, sums the values of all inputs according to equation (1):

$$S_j = \sum_{i=1}^{n} w_{ij} u_j + w_{0j} \tag{1}$$

In equation (1) an additional term, w_{0j}, has been included called a *bias*. An activation function is applied to the value S_j, to provide the final output from the neuron. This activation function can be linear, discrete, or some other continuous distribution function. However, in order to use the back-propagation algorithm to train a network (see below), this function must have the property of being everywhere differentiable. The sigmoid function satisfies this criterion and is the function generally used in most feed forward neural network applications. This function is represented by:

$$\mathrm{f}(x) = \frac{1}{1+\mathrm{e}^{-x}} \tag{2}$$

Training neural networks

Having outlined briefly how an ANN operates it is now possible to discuss how it is trained. A network learns by adjusting the biases and weights that link its neurons. However, before training can begin, a network's weights and biases must be set to small random values. A practical rule of thumb is to set the weights and biases to random values in the range $(-2/\Omega, 2/\Omega)$ for a neuron with Ω inputs (Gallant, 1993, p. 220). If initial random weights are not limited to this kind of range, network learning may be slow as extreme initial positioning on the sigmoid function can restrict the extent to which weight changes are made by the training algorithm. The extent to which this initial random weight range affects training in an ANN is beyond the scope of this paper.

Once a network has been initialized with preliminary weights and biases, the network is then trained by providing it with a number of examples (training pairs from the calibration set) which show the network how it is expected to behave. Each training pair has a particular input value (several, if there is more than one input node) and an expected output that the network should generate based on that input. The network is thus presented with this calibration data repeatedly (a specified number of *epochs*) until it is able to match its outputs with those that are expected (or closely enough to be acceptable). The way in which this training occurs is through the use of a training algorithm called *back-propagation*. This algorithm was rediscovered and popularized by Rumelhart & McClelland (1986) and is currently the most common approach to training feed forward ANNs (Gallant, 1993).

The basis of the back-propagation algorithm is that a training pair is selected from the training set and applied to the network. The network calculates what it "thinks" the output should be based on the inputs provided in this training pair. The resultant outputs from the network are then compared with the expected outputs identified by the training pair. The weights and biases of each neuron are then adjusted by a factor based on the derivative of the sigmoid function, the differences between the expected network outputs and the actual outputs (the error), and the actual neuron outputs. Through these adjustments it is possible to improve the results that the network generates, and thus the network is seen to *learn*. How much each neuron's weights and bias are adjusted in the back-propagation algorithm also depends on a *learning parameter*—a single factor by which all adjustments are multiplied. A large learning parameter can mean that training oscillates from one poor extreme result to another, whilst a small learning parameter can lead to a situation where the network does not learn anything and is caught in a local minimum, unable to take a bold step to reach a more accurate set of weights. Figure 3 provides an example where only one weight is adjusted in order to reduce a network's error.

W_1 in Fig. 3 highlights the concept of local minima in which a network can become trapped during training if the learning parameter is too small. In this case the adjustment cannot lift the weight over the "hills" on either side of W_1 and the network stabilizes with this error. Ideally the network would like to stabilize at W_2 but unless the learning parameter is increased this is impossible. One way around this problem is to use a variation of the back-propagation algorithm (Dawson, 1996), where the learning parameter is dynamically adjusted or, alternatively, retraining the network from scratch starting with a different set of initial weights and biases that may, by chance, be closer to W_2 to start with. Obviously, it takes more than one iteration of the back-propagation algorithm for a network to learn. In addition, a

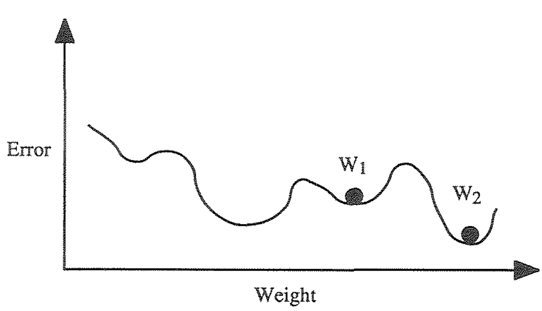

Fig. 3 Function showing weight *vs* error.

network must also be shown all the training pairs that are available, otherwise it will learn only one input and output combination and will not be able to generalize.

Although improvements can be made to back-propagation (for example, by adding *momentum*; Gallant, 1993) the exact detail of the algorithm is beyond the scope of this paper. However, the algorithm is well documented in the majority of texts devoted to the subject and the reader is directed to these for a more detailed explanation (Wasserman, 1989; Gallant, 1993).

Standardization

Due to the nature of the sigmoid function used in the back-propagation algorithm, it is prudent to standardize (i.e. convert to the range (0, 1)) all input values before passing them into a neural network. Without this standardization, large values input into an ANN would require extremely small weighting factors to be applied. This can cause a number of problems:

(a) Due to inaccuracies introduced by floating point calculations on microcomputers, one should avoid using the very small weighting values that would be required.

(b) Without using extremely small initial weights, changes made by the back-propagation algorithm would be insignificantly small, and training would be very sluggish, as the gradient of the sigmoid function at extreme values would be approximately zero. It is this gradient that is used in the adjustment of weights and biases in an ANN during training.

Due to the output range of the sigmoid function, all values leaving an ANN are automatically output in a standardized format. These output values must be "destandardized" to provide meaningful results. This can be achieved by simply reversing the standardization algorithm used on the input nodes. This is particularly troublesome when one handles real rainfall–runoff data as one must standardize all the data involved as well as deciding on the optimum way to achieve this.

There are two ways to approach data standardization:

- the values are standardized with respect to the range of all values; and
- the values are standardized with respect to the sum of squares of all values.

For example, for input values, these calculations are performed as follows:

$$N_i = \frac{R_i - \mathrm{Min}_i}{\mathrm{Max}_i - \mathrm{Min}_i} \tag{3}$$

$$N_i = \frac{R_i}{\sqrt{SS_i}} \tag{4}$$

where R_i is the real value applied to node i; N_i is the subsequent standardized value calculated for node i; Min_i is the minimum value of all values applied to node i; Max_i is the maximum value of all values applied to node i; and Ss_i is the sum of squares of all values applied to node i.

There are no fixed rules as to which approach should used in particular circumstances and there has been very little research on the subject. As a result, both techniques were used in the rainfall-runoff models discussed in this paper and the most accurate results are presented.

Network performance

In order to train and test artificial neural networks it is necessary to have two sets of training data—a calibration set and a validation set. Having trained a network with calibration data the accuracy of the results obtained from that network can be assessed by comparing its responses with the validation set. In this study the comparison was made using the mean squared relative error (MSRE) calculated as the mean of the square of the errors relative to each actual expected value in the validation set and the root mean squared error (RMSE). Although the MSRE provides more meaningful measures of overall network performance than relative differences alone, RMSEs were also calculated to enable comparisons to be made with other model results cited in the literature. In addition, visual comparisons can be made by plotting observed and modelled results. Graphs showing the results of the ANN rainfall-runoff models are presented later.

STUDY CATCHMENTS

Following consultations with the England and Wales Environment Agency (Severn-Trent and Thames regions) 15-min rainfall-runoff data were acquired for two flood-prone catchments with areas of approximately 140 km^2 (see Table 1): the River Amber and the River Mole.

The River Amber

The River Amber at Wingfield Park is an upland tributary of the Derbyshire Derwent. River flows are routinely gauged using a Crump profile Flat V weir,

Table 1 Hydrometric statistics for the Rivers Amber and Mole, UK.

	River Amber	River Mole
Gauge	Wingfield Park	Kinnersley Manor
Grid reference	SK 376520	TQ 262462
Catchment area (km^2)	139	142
Mean annual rainfall (mm)	789	793
Mean annual runoff (mm)	316	445
Mean flow ($m^3\ s^{-1}$)	1.4	1.95
10% flow ($m^3\ s^{-1}$)	2.9	4.2
Peak flow ($m^3\ s^{-1}$)	30.9	68.5
Max specific yields ($m^3\ s^{-1}\ km^{-2}$)	0.222	0.482

Source: *Hydrometric Register and Statistics 1986–1990* (NERC, 1993).

although peak flows are gauged from a bridge upstream. The catchment contains Ogsten reservoir and has substantial flow augmentation from mine water discharges and sewage. The upper moorland reaches are underlain by Millstone Grit, partially blanketed with Boulder Clay, whereas the lower half is underlain by Coal Measures. Hydrometric data for the River Amber were available for three time periods:

22 January 1995–6 February 1995: used for calibration;

6 February 1995–21 February 1995: used for calibration; and

21 December 1994–5 January 1995: used for validation.

Flows were available in Ml day^{-1} at 15-min time intervals whilst rainfall was measured as tip times (for every 0.5 mm) at four rainfall stations—Ogston Reservoir, Longcliffe, Carsington Dam, and Sutton-in-Ashfield. Tip time data were converted into 15-min rainfall totals for each gauge to coincide with the flow data format.

The choice of which data to use for calibration and which data to use for validation was based on the following two points: (a) the calibration period chosen had a high "information content" i.e. there were a large number of flood events in this period; and (b) the validation period was chosen to coincide with the availability of data obtained from tests of the Severn-Trent Flood Forecasting System (Cross, personal communication).

The River Mole

By contrast, the River Mole at Kinnersley Manor is a lowland tributary of the River Thames which drains a largely impervious catchment consisting mostly of Weald Clay. Flows are gauged using a rectangular flume, calibrated by current meter gauging which extend beyond bankfull. The Environment Agency supplied rainfall–runoff data for the River Mole for the whole of 1994. These data represented rainfall (mm) measured at 15-min intervals at the Burstow raingauge and 15-min flow measurements made at Kinnersley Manor (m^3 s^{-1}). The river has experienced significant and increasing net imports of sewage effluent through time. The land use is very mixed with rural tracts and urban centres such as Crawley and Gatwick Airport.

As a full year of data at 15-min time intervals represented a particularly large data set (35 040 values), and covered all seasons, it was decided to train the neural networks on the last 100 days of the data (autumn and early winter). This enabled the first 100 days of the data (winter and early spring) to be used as a validation set. The reasons for this choice were as follows: (a) calibrating on more recent data and validating on earlier data was consistent with the approach adopted with the River Amber; and (b) the latter half of the year also contained the highest observed flow in the available data.

METHODS

This section briefly describes the computer programs that were written to implement the ANNs. It goes on to discuss how data from the two sources identified above were manipulated into a form that could be used by the ANNs and how the hydrological data were processed to improve the effectiveness of the trained ANNs.

The programs

Rather than using an off-the-shelf ANN package for this work, it was felt that an implementation of the back-propagation algorithm in a high level language would be more appropriate. The reasons for this choice were: (a) the neural network algorithm is relatively simple and easy to implement in a high level language; (b) a number of ANN programs, with which the authors are familiar, had already been developed at the University of Derby for a similar project and this eliminated the need for training on commercial or shareware packages; and (c) network performance can be improved by variations to the back-propagation algorithm; these enhancements were already included in the prewritten programs.

The artificial neural networks were developed by implementing the back-propagation algorithm in Pascal on a Silicon Graphics' Indy in the School of Mathematics and Computing running UNIX. Programs were written to build and train the networks and a number of others were developed to analyse and modify the data supplied.

Data manipulation

This section discusses how the data were analysed and transformed into inputs for the ANNs. The ANNs were developed to predict river flow at a time t_0 from events (for example rainfall, flow, storms) that had occurred earlier—at times t_{-n}; which represent times n min before t_0. The rainfall–runoff data were analysed to identify significant relationships between events (at a time t_{-n}) and current river flow (at time t_0). An arbitrary lead time of six hours (t_{-360}) was chosen as an initial starting point for the analysis as shorter lead times would limit the value of such a model for flood prediction.

Data analyses and manipulations were performed *a priori* to ANN training in order to establish the most appropriate averaging periods for the input data. This, in effect, provided the ANN with a short "memory" or moving average of previous events and antecedent conditions that were thought to be of significance to "current" and forecast hydrological conditions within each of the river catchments. The analyses identified the strongest correlations for given time intervals and lead times for the "average" condition of the calibration data set. Clearly, the validity of the average lead times and smoothing intervals will vary between seasons, different catchments, and even within individual flood events.

Case 1—River Mole

Significant correlations were obtained between flow and the first five factors shown in Table 2. In addition, it was also noted that rainfall with a 15-h lead time (t_{-900}) had a strong correlation with river flow at t_0 and this was also chosen as an another input driver for the ANN (input driver 6).

It was also apparent that individual storm events have a significant effect on river flow. With this in mind a program was written that isolated storm events in a rainfall sequence. The algorithm that was implemented filtered out low levels of precipitation from rainfall data by firstly identifying storms as prolonged periods of rainfall lasting over one hour (over 0.5mm per 15-min interval) and secondly by aggregating precipitation over that time period. This did not lead to any loss of data as 15-min rainfall totals were still being used as an input to the ANN. The storm data generated

Table 2 River Mole ANN input drivers to predict flow at time t_0.

Input number	Name	Lead time
1	Flow	t_{-360}
2	30-min moving average of flow	t_{-360}
3	20-h moving average of rainfall	t_{-360}
4	Previous 24 h total flow	t_{-360}
5	Previous 24 h total rainfall	t_{-360}
6	15-min rainfall total	t_{-900}
7	Storm events	t_{-1620}

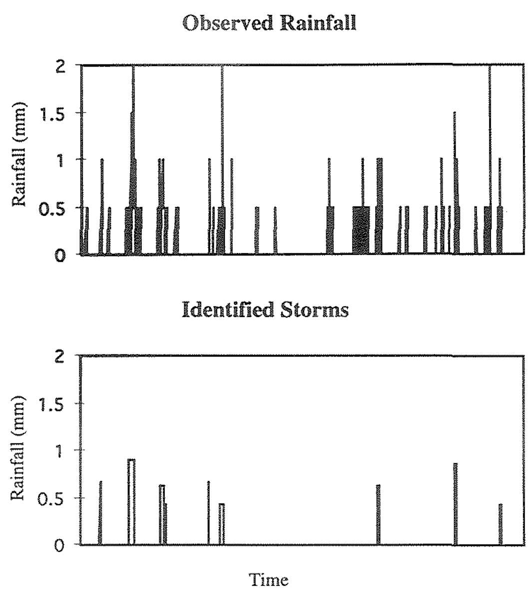

Fig. 4 An example of storm identification in observed rainfall.

did, however, supplement the ANN with a clearer view of significant prolonged rainfall spells that might otherwise have been distorted by intermittent periods of low rainfall. The results of this manipulation can be seen in the example provided in Fig. 4. This Figure shows how storm events have been highlighted by eliminating small and isolated rainfall amounts and by smoothing out rainfall peaks. Hydrologically, this process is analogous to the "filtering" of small (< 1 mm) and isolated precipitation events due to vegetation canopy interception, surface depression storage and subsequent evaporation.

The rainfall data were thus converted into a file representing storm events and analysed to determine the most influential lead time on river flow. This resulted in another input driver (number 7 in Table 2) for the ANN, i.e. storm events with a lead time of 27 h (t_{-1620}).

The seven factors shown in Table 2 represent the main flow drivers that were available from the data supplied and thus constituted the seven inputs that were used to train a number of ANNs with different topologies and different input standardization approaches for the River Mole. Although these data contained a number of missing flow values (due to problems with the original data source) no additional identification was imparted since ANNs are able to deal with missing or noisy data and such "flagging" is unnecessary.

Case 2—River Amber

Work began by determining the strongest relationships between flow in the River Amber, at time t_0, and factors such as rainfall, storms and previous flow data starting with an initial minimum lead time of 6 h (t_{-360}). Analyses resulted in the 15 factors shown in Table 3 being selected as the input drivers for river flow for training the ANNs. These drivers represent the main influences on river flow at time t_0. Note that

Table 3 River Amber ANN input drivers to predict flow at time t_0.

Input number	Name	Lead time
1	Flow	t_{-360}
2	30-min moving average of flow	t_{-360}
3	24-h moving average of flow	t_{-360}
4	15-min rainfall total at Ogston Reservoir	t_{-600}
5	15-min rainfall total at Longcliffe	t_{-660}
6	15-min rainfall total at Carsington Dam	t_{-720}
7	15-min rainfall total at Sutton-in-Ashfield	t_{-660}
8	Storm events at Ogston Reservoir	t_{-600}
9	Storm events at Longcliffe	t_{-600}
10	Storm events at Carsington Dam	t_{-720}
11	Storm events at Sutton-in-Ashfield	t_{-480}
12	19-h rainfall moving average—Ogston Reservoir	t_{-360}
13	18-h rainfall moving average—Longcliffe	t_{-360}
14	20-h rainfall moving average—Carsington Dam	t_{-360}
15	20-h rainfall moving average—Sutton-in-Ashfield	t_{-360}

the rainfall series of the four individual gauges (as opposed to lumped area averages) were used as inputs to the ANN. This approach maximizes the information content of the available data and acknowledges the fact that different areas of the catchment will exhibit different lag times between rainfall–runoff, information that is readily assimilated by the ANN.

Finally, Table 4 summarizes the ANN topologies, epochs and standardization approaches used for the rainfall–runoff models of the River Mole and the River Amber.

Table 4 ANN characteristics summary.

River	Input nodes	Hidden nodes	Output nodes	Standardization	Epochs	Learning parameter
Mole	7	5, 10, 20	1	Range, Sum of squares	500, 1000, 2000	0.1
Amber	15	5, 10, 15, 20, 30, 50	1	Range, Sum of squares	500, 1000, 2000	0.1

RESULTS

River Mole

Several networks were trained using the seven input drivers identified in Table 2 with one output, the river flow measured at 15-min intervals at time t_0. The networks were trained using variations in the number of hidden nodes, standardization approaches (as identified earlier) and for different numbers of epochs.

When comparing the results of all trained ANNs with the calibration set all the networks were extremely accurate in their predictions of river flow. As an example, Fig. 5 shows the results of the ANN with 20 hidden nodes, trained for 2000 epochs, with input and output values standardized by range. In this example the MSRE was calculated as 0.339 and the RMSE as 1.906 $m^3 s^{-1}$. Figure 5 indicates that the ANN reproduced the largest flow (which was actually close to the historic record) and intervening low flows, but was less proficient with regard to the earlier flood peaks of the autumn recharge period.

What is more interesting, however, is how well a trained network forecasts peak flows with a lead time of 6 h on the validation set, given that the data are from different seasons, i.e. winter to early spring. The ANN validation results presented in Fig. 6 have a MSRE of 0.350 and a RMSE of 3.618 $m^3 s^{-1}$. Although the peaks and troughs are followed closely for the first 50 days of the validation period, it is evident that there is some "drift" in the ANN performance throughout the last 50 days. This discrepancy was attributed to the sigmoid function and the sum of squares standardization routines (see above) which yield better estimates of peak flows at the expense of relatively poor low flow simulation. In practice, flood-forecasting models are seldom employed over such long periods without some means

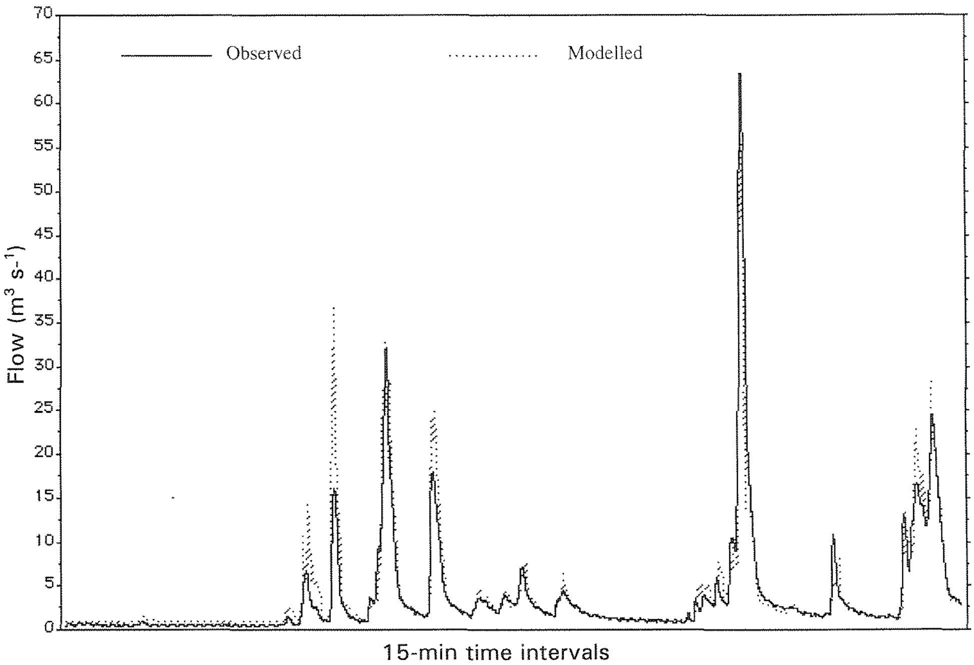

Fig. 5 ANN calibration results for the River Mole at Kinnersley Manor (23 September 1994–31 December 1994).

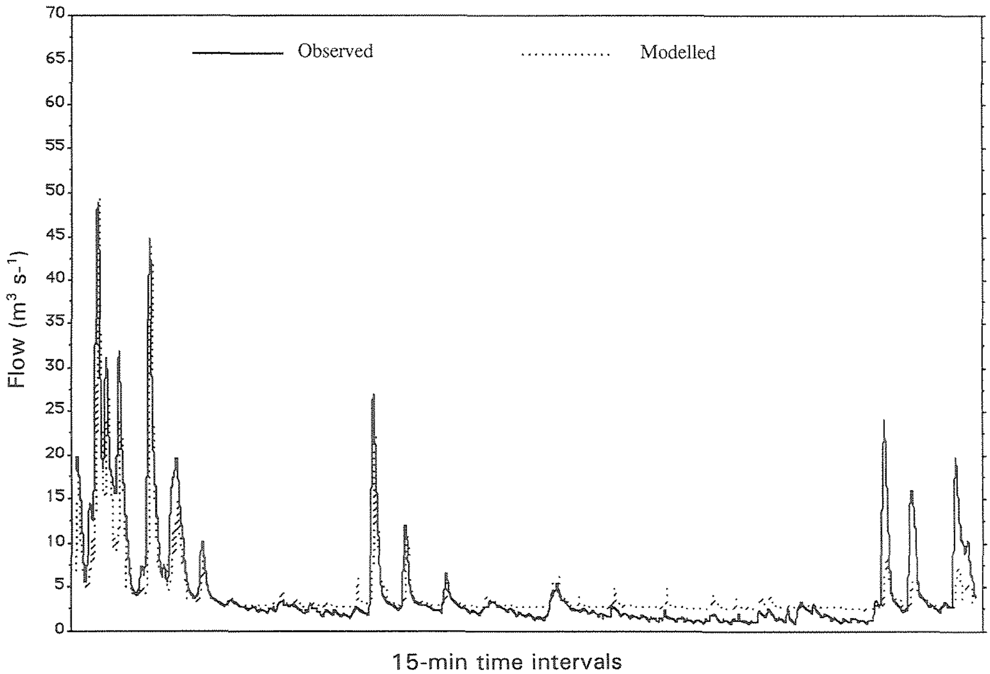

Fig. 6 ANN validation results for the River Mole at Kinnersley Manor (1 January 1994–10 April 1994).

of updating state variables or parameters: the uncorrected flows with a lead time of 6 h shown in Fig. 6 are, therefore, a very severe test of the ANN forecasting capability.

River Amber

The River Amber data provided 2880 training pairs initially but, after moving averages had been calculated, this reduced the calibration set to 2761 training pairs. Similarly, after calculations had been performed, the validation set was reduced to 1321 pairs.

The best results were obtained from a network comprised of 10 hidden nodes with inputs and outputs standardized by range and trained for 500 epochs. In this case the MSRE was calculated as 0.059 for the calibration data and 0.205 for the validation data. For comparative purposes the RMSE was also calculated and resulted in 2.679 $m^3 s^{-1}$ for the calibration set and 1.181 $m^3 s^{-1}$ for the validation data (Figs 7, 8). However, a visual inspection of the calibration results shown in Fig. 7 suggests that the ANN consistently underestimates the magnitude of peak flows, particularly when those flows are preceded by another flood event. Hydrologically, this deficiency may be interpreted as a consequence of the increased rainfall–runoff efficiency of the catchment with successive flood events. Under these conditions, the input driver for the previous 24-h moving average of flow (see Table 3) is clearly inadequate. In other words, the ANN does not have a sufficiently long memory of preceding catchment conditions, or a continuously accounting parameter that is analogous to the soil moisture deficit term used by conventional lumped-conceptual flood models.

In order to compare the performance of the ANN with an existing model, flow forecasts were obtained from the Severn-Trent Environment Agency's flood forecasting system (FFS) for the River Amber at Wingfield Park (Cross, personal communication). Figures 9 and 10 show the 15-min flows for two periods corresponding to those used for the ANN 6-h forecasts during calibration (Fig. 7) and validation (Fig. 8) series respectively. Note that the FFS hydrographs, shown in Figs 9 and 10, were obtained during model calibration and, as such, represent the "best case" scenario. In practice, the Environment Agency issues flood warnings with a 2-h lead time for this reach of the River Amber. Therefore, the validation results obtained from the ANN with a 6-h lead time constitutes another severe test of the ANN model performance. Furthermore, unlike the FFS, the ANN forecasts were <u>not</u> adjusted incrementally according to previous forecasting errors.

From Figs 7 and 9 it is evident that the performance of the ANN during the calibration period was comparable to that of the FFS. The timing and magnitude of the ANN flood peaks on 26 and 28 January 1995 were particularly noteworthy given that the maximum flow on the first occasion actually exceeded the historic peak cited in the NERC (1993) Hydrometric Register (see Table 1). Although the FFS yielded a better estimate of the magnitude of the second flood, the timing of the peak was

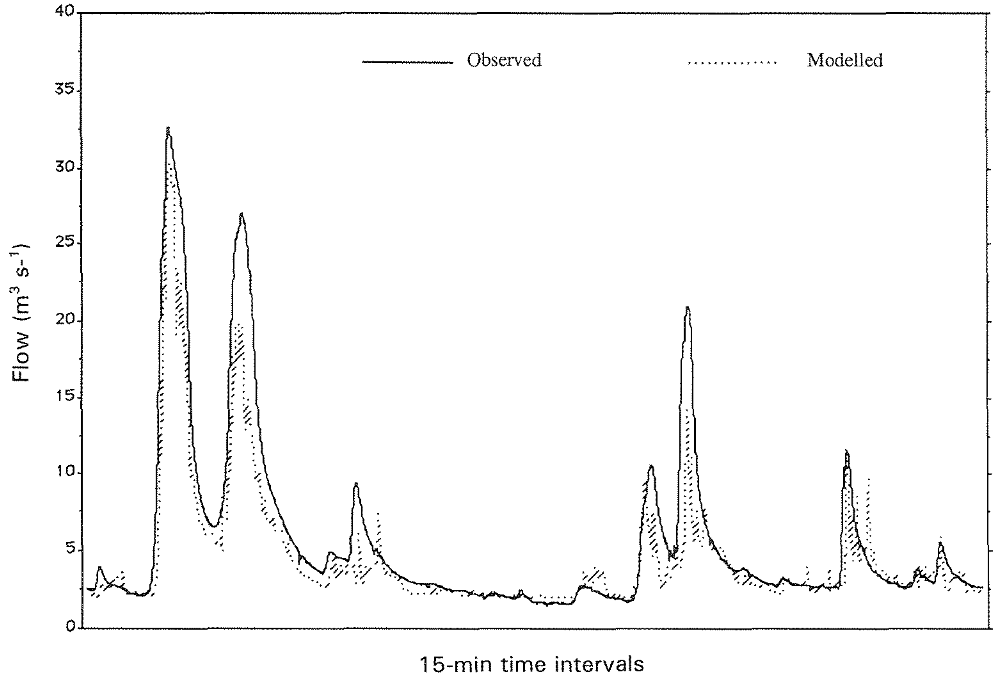

Fig. 7 ANN calibration results for the River Amber at Wingfield Park (22 January 1995–21 February 1995).

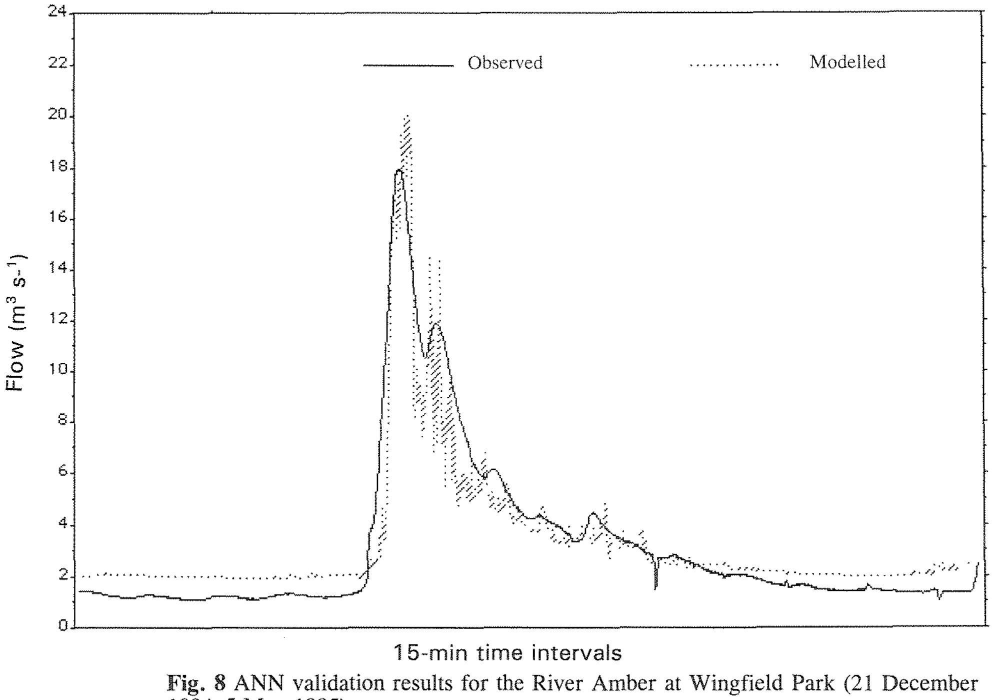

Fig. 8 ANN validation results for the River Amber at Wingfield Park (21 December 1994–5 May 1995).

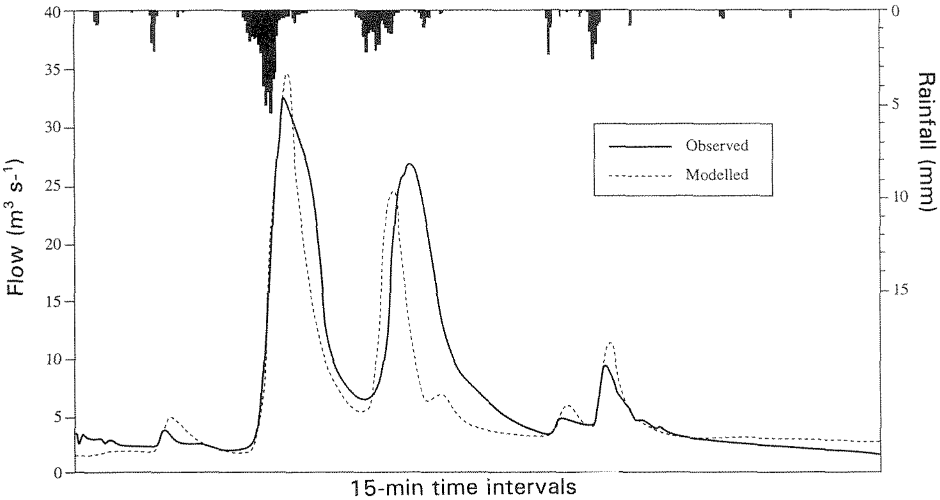

Fig. 9 Severn-Trent Environment Agency, UK Flood Forecasting System (FFS) results for the River Amber at Wingfield Park (22 January 1995–6 February 1995). Note that these model results correspond to the first sixteen days shown in Fig. 7.

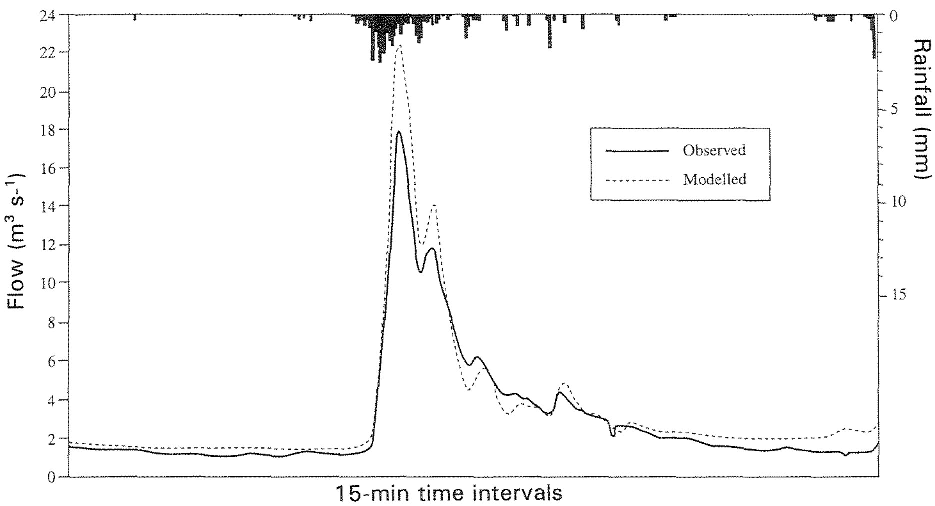

Fig. 10 Severn-Trent Environment Agency, UK Flood Forecasting System (FFS) results for the River Amber at Wingfield Park (21 December 1994–5 January 1995). Note that these model results correspond to those shown in Fig. 8.

better in the ANN simulation. The shape of the hydrograph for 21 December 1994 to 5 January 1995 (Figs 8 and 10) was also reproduced well by both models. However, it is interesting to note that both the ANN and FFS overestimated the magnitude of the two flood peaks relative to the observed values. This may be an artefact of inaccurate peak flow measurements (which are gauged from a bridge upstream of the weir and are bypassed on the right hand bank) rather than a deficiency in the models *per se* (NERC, 1993). However, compared with the FFS, the flood hydrograph produced by the ANN contains much more noise on the recession limb; clearly, in this case, the ANN is over-sensitive to the influence of individual 15-min rainfall totals (see Table 3). This deficiency may be attributed to the exceptional hydrological conditions of the training period, during which record-breaking peak flows were logged at Wingfield Park in late January 1995. Under these conditions, the saturated catchment would have resulted in high rainfall–runoff efficiencies and relatively short times to peak, both of which would have contributed to the high responsiveness of the validated ANN to rainfall.

Sensitivity of models to calibration season and forecast times

It was recognized above that the trained ANNs could be sensitive to the choice of calibration season. In the case of the ANN for the River Amber, which was trained and validated on winter data, seasonal changes in catchment properties will be less significant. However, the ANN for the River Mole, which was trained in autumn/ early winter and validated in winter/ early spring, the ANN is likely to be less responsive to rainfall than if training had been undertaken using data for the wetter winter period. Similarly, if the ANNs are to be used operationally in the longer term it would also be necessary to take into account other catchment processes such as urbanization, land use change or trends in seasonal abstraction/effluent returns.

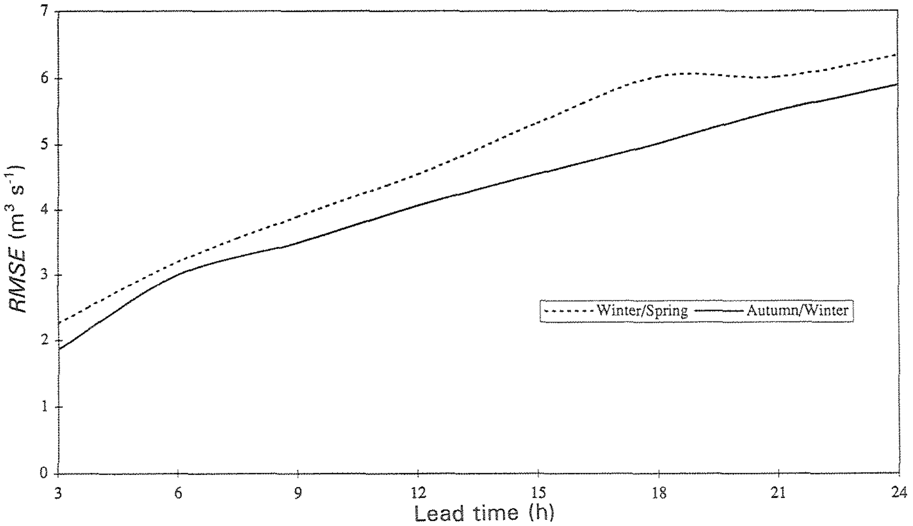

Fig. 11 ANN validation results obtained for the River Mole using 15-min precipitation and discharge series at time t_0 to forecast flows at time t_{0+n} (where n is 3, 6, 9, ..., 24 h from present). The two lines are indicative of the different training periods used to calibrate the ANN (i.e. winter/spring *vs* autumn/winter).

In order to address these issues, the sensitivity of the models to the training season and forecast lead time was investigated by calibrating an ANN using present rainfall and discharge as inputs to predict flows 3, 6, 9, ..., 21, 24 h hence. The River Mole ANN was first trained against winter/spring rainfall–runoff and then validated against autumn/winter data. Then, in a second set of simulations, the data used in the calibration and validation periods were interchanged. Figure 11 shows the RMSEs achieved by the ANNs trained using different seasons for successive increases in the forecast lead time. As would be expected the RMSE increased progressively as the forecast lead time became greater. However, it is evident that the performance of the ANN trained using autumn/winter rainfall–runoff data and validated against winter/spring flows was superior to the winter/spring trained ANN. Thus, the choice of the calibration and validation periods does have a demonstrable effect on the forecasting ability of the ANN. Ideally, the ANN should be calibrated and validated using data for a common season; alternatively, additional input variables (such as temperature or evaporation) should be employed to represent the seasonally-dependent antecedent conditions or dominant flood generating processes.

DISCUSSION

ANNs are like conventional hydrological models in that different attributes of the hydrograph are simulated to varying degrees of success (*cf* Sorooshian, 1991). The optimization criteria used here (MSRE) is unbiased, whereas an R^2 statistic or the RMSE would have simulated peaks better than low flows. An appropriate choice of optimization criterion would, therefore, have facilitated improved training of the flood magnitudes, times to peak, or low flows presented herein, whichever property is at issue. Similarly, it was demonstrated that the choice of standardization technique (sum of squares or range) also has consequences for the ANN's skill at simulating peak as opposed to low flows. Therefore, experience gained from rainfall–runoff modelling of the Rivers Amber and Mole suggests that accurate flood forecasting using ANNs requires: (a) training of the ANN against selected elements (i.e. individual flood hydrographs) contained in the total flow data set; (b) the use of sum of squares standardization of ANN output; (c) an optimization criterion such as the RMSE which is biased towards peak flow estimation; (d) the inclusion of input drivers such as temperature which can be used to distinguish between different precipitation types (i.e. rainfall *vs* snowfall/snowmelt), and hence the flood generating mechanism; and (e) finally, input drivers which contain some "memory" of the antecedent catchment conditions.

The RMSEs cited earlier for the validation simulations in both the River Amber and Mole lie well within the ranges given for the WMO (1992) simulated real-time flood forecasting intercomparison exercise. For the comparable 100 km^2 Orgeval catchment in France, the eleven models in the WMO study returned RMSEs of 2.5–8.5 m^3 s^{-1} for 6-h forecasts. Taken in conjunction with the Severn-Trent FFS model comparisons, the results of the present pilot study suggest that there is considerable scope for the development of a fully operational ANN flood forecasting system. The ability of the ANN to cope with missing data and to update forecasts in real time makes it an appealing alternative to conventional lumped or semi-distributed flood forecasting models.

However, further research is required to determine the optimum training period for given catchment and climatic contexts which could, perhaps, be established by a second ANN in a real time system. Additionally, improvements might be achieved by training ANNs on the rate of change of flow from one time period to the next, rather than absolute runoff values. This may, however, lead to networks "snowballing" incorrect estimates into later predictions.

As Minns & Hall (1996) observed, the ANN is susceptible to becoming a "prisoner of its training data". The number of hidden nodes and training epochs will determine the specificity of the resultant model weights to the given calibration data. As French *et al.* (1992) note, increasing the number of training iterations with no change in ANN structure improves the performance on the training data but not necessarily for independent data. A compromise must therefore be established between constructing an ANN that faithfully reproduces the key elements of the flow series in the calibration period yet is sufficiently robust in the face of previously unseen data. For operational purposes, the moving training period must also provide sufficient data for model recalibration and an adequate length of memory of antecedent behaviour. Conversely, the training period should be sufficiently concise, recognizing the software and hardware constraints imposed by real-time forecasting and model updating.

The results of the pilot study presented herein have demonstrated the potential for ANN development in the field of rainfall–runoff modelling and flood forecasting. Given relatively brief calibration data sets it has been possible to construct robust models of 15-min flows with 6-h lead times for two flood-prone catchments. Future research should extend the techniques to other catchments and forecasting lead times. There is also a need for thorough investigation into the relationship between the training period length (or information content) and the hydrological realism of the ANN forecasts. Preliminary consideration of the ANN representation of the River Mole suggested that the fine detail of individual flood hydrographs could be lost or generalized by an ANN that has been trained using 100 days of hydrometric data. This fact may be a reflection of the changing antecedent conditions and seasonality in the responsiveness of the catchment. It also points to the need for flexibility in the selection of appropriate inputs, data lagging and averaging periods. Such effects might be usefully investigated by comparing optimum ANN weights derived for floods arising in different months or seasons.

Acknowledgements The authors are grateful to Richard Cross (Severn-Trent Environment Agency) and Brian Greenfield (Thames Environment Agency) for provision of the hydrometric data used in this study. They also thank the anonymous referees for their constructive comments. Finally, they acknowledge the support and assistance of Reginald Dawson and Fabrice Bolarin.

REFERENCES

Alexander, D. (1991) Information technology in real time for monitoring and managing natural disasters. *Progress in Phys. Geogr.* **15**, 238–260.

Cheng, X. & Noguchi, M. (1996) Rainfall-runoff modelling by neural network approach. *Proc. Int. Conf. on Water Resour. & Environ. Res.* **2**, 143–150.

Collier, C. G. (1991) Problems of estimating extreme rainfall from radar and raingauge data illustrated by the Halifax storm, 19 May 1989. *Weather* **46**, 206–210.

Dawson, C. W. (1996) A neural network approach to software project effort estimation. *Applications of Artificial Intelligence in Engng* **1**, 229–237.

Douglas, J. R. & Dobson, C. (1987) Real time flood forecasting in diverse drainage basins. In: *Weather Radar and Flood Forecasting* (ed. by V. K. Collinge & C. Kirkby), 153–169. Wiley, London.

Franchini, M., Helmlinger, K. R., Foufoula-Georgiou, E. & Todini, E. (1996) Stochastic storm transposition coupled with rainfall runoff modeling for estimation of exceedance probabilities of design floods. *J. Hydrol.* **175**(1–4), 511–532.

French, M. N., Krajewski, W. F. & Cuykendall, R. R. (1992) Rainfall forecasting in space and time using a neural network. *J. Hydrol.* **137**, 1–31.

Gallant, S. I. (1993) *Neural Network Learning and Expert Systems.* MIT Press, London.

Georgakakos, K. P. & Foufoula-Georgiou, E. (1991) Real time coupling of hydrologic and meteorological models for flood forecasting. In: *Proc. NATO Advanced Study Institute on Recent Advances in the Modelling of Hydrologic Systems* (Sintra, Portugal, 10–23 July 1988) (ed. by D. S. Bowles & P. E. O'Connell). Kluwer, The Netherlands.

Hall, M. J. & Minns, A. W. (1993) Rainfall-runoff modelling as a problem in artificial intelligence: experience with a neural network. *BHS 4th National Hydrology Symposium, Cardiff,* 5.51–5.57.

Hewitson, B. C. & Crane, R. G. (1996) Climate downscaling: techniques and application. *Climate Res.* **7**, 85–95.

Hewitson, B. C. & Crane, R. G. (1994) *Neural Nets: Applications in Geography* Kluwer, Dordrecht, The Netherlands.

Hopfield, J. J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.* **79**, 2554–2558.

Hopfield, J. J. (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci.* **81**, 3088–3092.

Lamberti, P. & Pilati, S. (1996) Flood propagation models for real-time forecasting. *J. Hydrol.* **175**(1–4), 239–266.

Minns, A. W. & Hall, M. J. (1996) Artificial neural networks as rainfall-runoff models. *Hydrol. Sci. J.* **41**(3), 399–417.

NERC (Natural Environment Research Council) (1993) *Hydrometric Register and Statistics 1986–1990,* 190. Institute of Hydrology and British Geological Survey, Wallingford, UK.

O'Connell, P. E. (1991) A historical perspective. In: *Proc. NATO Advanced Study Institute on Recent Advances in the Modelling of Hydrologic Systems* (Sintra, Portugal, 10–23 July 1988) (ed. by D. S. Bowles & P. E. O'Connell). Kluwer, Dordrecht, The Netherlands.

Raman, H. & Sunilkumar, N. (1995) Multivariate modelling of water resources time series using artificial neural networks. *Hydrol. Sci. J.* **40**(2), 145–163.

Rumelhart, D. E. & McClelland, J. L. (eds) (1986) *Parallel Distributed Processing: Explorations in the Microstructures of Cognition,* 1. MIT Press, Cambridge.

Sorooshian, S. (1991) Parameter estimation, model identification, and model validation: conceptual-type models. In: *Proc. NATO Advanced Study Institute on Recent Advances in the Modelling of Hydrologic Systems* (Sintra, Portugal, 10–23 July 1988) (ed. by D. S. Bowles & P. E. O'Connell). Kluwer, Dordrecht, The Netherlands.

Wasserman, P. D. (1989) *Neural Computing Theory and Practice.* Van Nostrand Reinhold, New York.

Wood, E. F. & O'Connell, P. E. (1985) Real time forecasting. In: *Hydrological Forecasting* (ed. by M. G. Anderson & T. P. Burt), Chapter 15. Wiley, Chichester.

WMO (World Meteorological Organization) (1992) Simulated real-time intercomparison of hydrological models. *Operational Hydrology Report no. 38, WMO 779, Geneva, Switzerland.*

Received 13 January 1997; accepted 4 June 1997