Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model

Robert J. Abrahart, Linda See and Pauline E. Kneale

ABSTRACT

Four design tool procedures are examined to create improved neural network architectures for forecasting runoff from a small catchment. Different algorithms are used to remove nodes and connections so as to produce an optimised forecasting model, thereby reducing computational expense without loss in performance. The results also highlight issues in selecting analytical methods to compare outputs from different forecasting procedures.

Key words | hydrological modelling, neural network, optimisation

Robert J. Abrahart
School of Earth and Environmental Sciences,
University of Greenwich, Medway Campus,
Central Avenue, Chatham Maritime,
Kent ME4 4TB, UK
E-mail: *bob@ashville.demon.co.uk*

Linda See
Pauline E. Kneale
School of Geography,
University of Leeds,
Leeds LS2 9JT,
West Yorkshire, UK
E-mail: *l.see@geog.leeds.ac.uk*;
pauline@geog.leeds.ac.uk

INTRODUCTION

Neural networks have been applied to a variety of hydrological forecasting tasks. They include the use of back-propagation networks to model synthetic rainfall-runoff data (Minns & Hall 1996), the use of self-organizing maps for data sub-division to facilitate integrated multi-network modelling (Abrahart & See 1998), and the adoption of neural network solutions as embedded functions within stand-alone programs and traditional program code (Abrahart 1998). Despite significant computational and methodological advances, fundamental problems remain in the selection of optimal network architectures.

There are no strict rules governing the design of a neural network. More complex problems generally require a more complex solution. When there are many free parameters, the network will be slower to train and more susceptible to overfitting. Factors such as the number of inputs, the number of hidden nodes, and their arrangement into layers, are often determined using systematic 'trial and error' (Fischer & Gopal 1994) or based on reasonable but subjective opinion (Cheng & Noguchi 1996). Testing for optimum inputs and architectures can be a time-consuming process, and the end result may be neither informative nor convincing.

This paper reports the results of some computational experiments using automated neural network design tools. The experiments were designed to investigate the power, modelling possibilities and application potential associated with computer-based algorithms to:

- find a more suitable network architecture;
- reduce computational overheads;
- improve generalisation capabilities;
- locate non-essential inputs and provide evidence of input relevance.

The forecasting application was a 'one-step-ahead prediction' of river flow records. Four network model-building strategies were implemented. The first used standard procedures to create a set of neural network models. The second and third were similar but incorporated simple pruning algorithms to create more efficient architectures. The fourth employed a genetic algorithm package to breed optimised neural network solutions based on random mutation and survival of the fittest. The initial network architecture and 6 hour historical predictive record had earlier been found sufficient for experimental modelling (Abrahart & Kneale 1997).

The study catchment was the upper River Wye, Wales (Figure 1). This basin covers an area of 10.55 km^2,

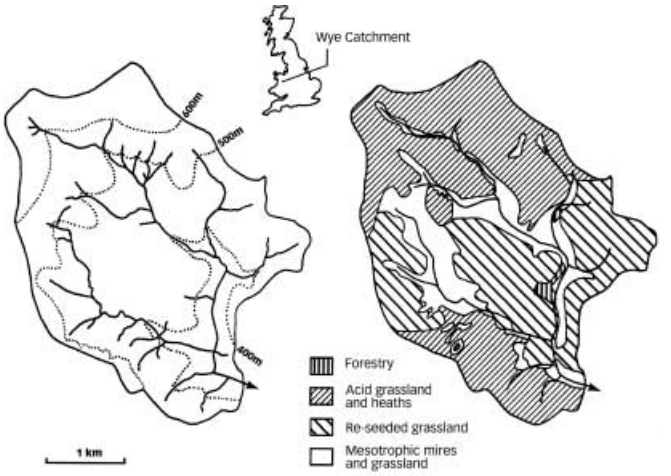

Figure 1 | Upper Wye Catchment (after Beven *et al.* 1984).

elevations range from 350–700 m, and average annual rainfall is 2,500 mm. Previous hydrological modelling of this catchment includes Beven *et al.* (1984), Bathurst (1986) and Quinn & Beven (1993). Data for the Cefn Brwyn gauging station (number 55008) comprised rainfall (RAIN), potential evapotranspiration (PET), and river flow ordinates (FLOW) on a 1 hour time step. The data were processed into 23 variables: two annual hour-counts (CLOCK), transformed into their sine and cosine equivalents, RAIN t, RAIN t-1 to t-6, PET t, PET t-1 to t-6, FLOW t-1 to t-6, and FLOW t. All variables were normalised, between zero and one, then split into annual data sets: 1984, 1985 and 1986. Flow values are reported here in normalised flow units (nfu).

METHODS

Initial network architecture

The starting point for each investigation was a two-hidden-layer feed-forward network with a 22:16:14:1 architecture. All between-layer connections were enforced and no cross-layer connections were permitted (Figure 2). The input nodes correspond to sin[CLOCK], cos[CLOCK], current and previous RAIN [t, t-1 to t-6], current and previous PET [t, t-1 to t-6], and the FLOW ordinates [t-1 to t-6]. The output node was for current FLOW [t].

Optimisation methods

Two methodologies were examined.

1. *Pruning algorithms* that remove inconsequential links or nodes. Two procedures were investigated: magnitude-based pruning, which eliminates unwanted links; and skeletonization, which eliminates unwanted nodes.
2. *Genetic algorithms* that manipulate links and nodes according to the principles of evolution. This allowed deletion of links and nodes, replacement of deleted items, and the addition of new ones.

Training and testing

Stuttgart Neural Network Simulator (SNNS) was used to perform the basic neural network modelling operations. It also implemented both pruning algorithms. Training was based on enhanced backpropagation (BPROP). The initial 22:16:14:1 network was trained on one annual data set and then tested with the other two. This operation was repeated for each of the three data sets and an optimal solution, for each model-building scenario, selected. Statistical and graphical analysis of the preferred neural network solutions followed. Multiple training and testing with data from three different time periods facilitated several informative comparisons: 1984 was a drought year; 1985 had a few intermediate events; and 1986 contained a higher proportion of major flood occurrences.

Three standard training runs were undertaken to provide a benchmark. Each annual data set was used to train the initial network. Low rates of learning and momentum were used throughout. Sum-squared error statistics were computed at 100 epoch intervals and translated into a combined graph from which the best overall modelling solution for each annual data set could be selected. Each optimal model was selected at that point where the error associated with one or other of the two validation data sets began to increase in a continuous manner and with no subsequent fallback (Figure 3). With magnitude-based

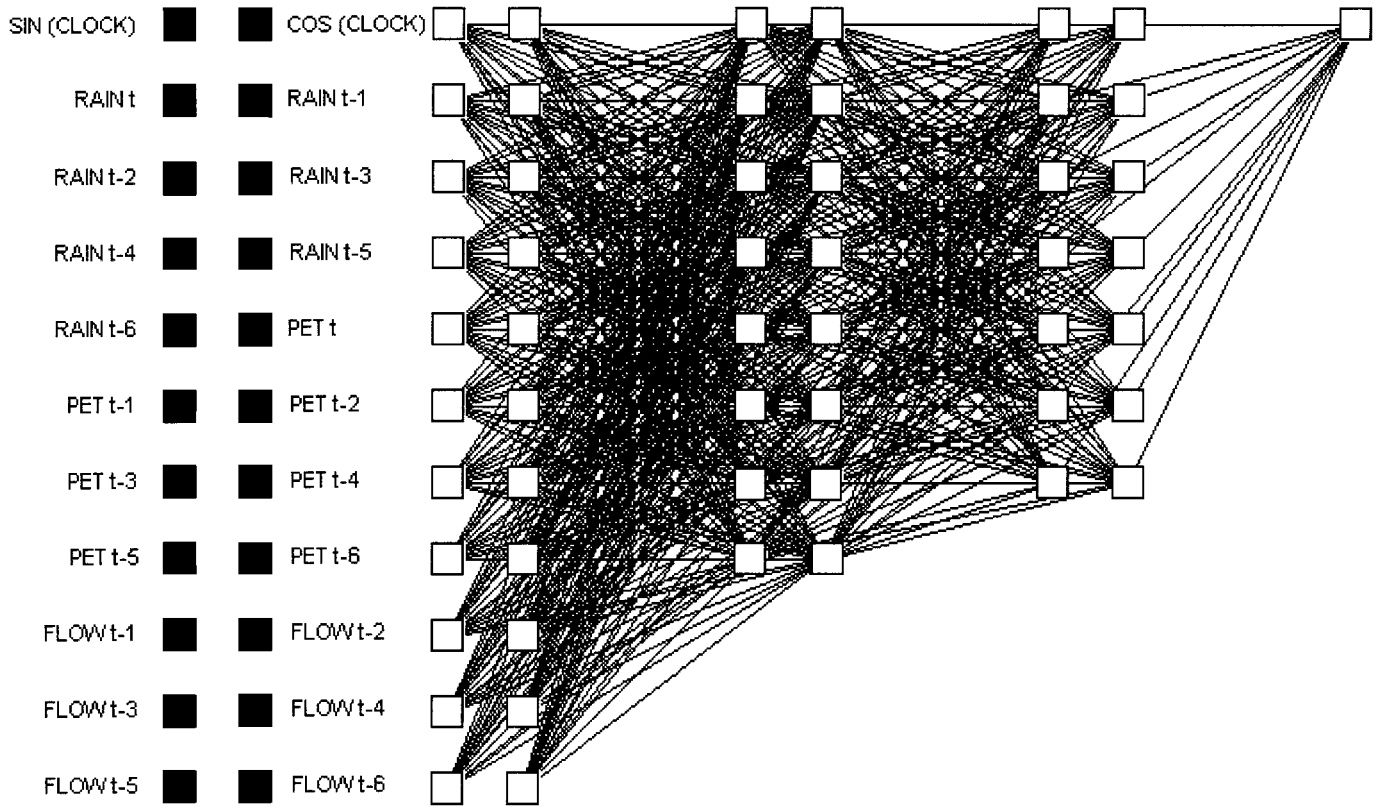

Figure 2 | Initial network architecture for all models. Each layer has been folded in half to produce a more compact diagram.

pruning after each period of 100 epochs the five lowest weighted connections were deleted. With skeletonization after each period of 100 epochs the node that produced the least overall change in the global sum squared error statistic, when omitted, was deleted. Both elimination procedures were run until the network could no longer function and the best solution selected from an inspection of the training graphs.

ENZO is a genetic algorithm (GA) tool adapted for use with SNNS. All global optimisation procedures must balance the level of exploration and the level of exploitation, because full exploration is computation-expensive, and the exploitation of additional derived information carries the risk of becoming trapped in a poor local minimum. Evolution-based algorithms avoid becoming trapped in local minima by using a parallel search process. This procedure is, however, biased towards exploitation because the fittest 'parents' are selected to create future generations. GA tools are problem independent, and therefore neglect vital problem knowledge, such as gradient information relating to the solution surface. So the use of a pure evolution-based GA should at best produce modest results compared with those that can exploit additional factors. But a neural network is also capable of moving down the solution gradient, so the application of a hybrid evolution-based method will allow us to restrict the search space to a set of local optima. Hence each network was also trained. The batch-learning algorithm Resilient Propagation (RPROP) provided a fast mechanism that was suitable for training many networks. The starting pattern of hidden nodes was based on random selection and weak connections were deleted on a regular basis. Training stopped when the mean error was 0.0005 nfu or when 100 epochs had been reached. The total number of neural network models that were generated and evaluated in this manner was 2,400. Each

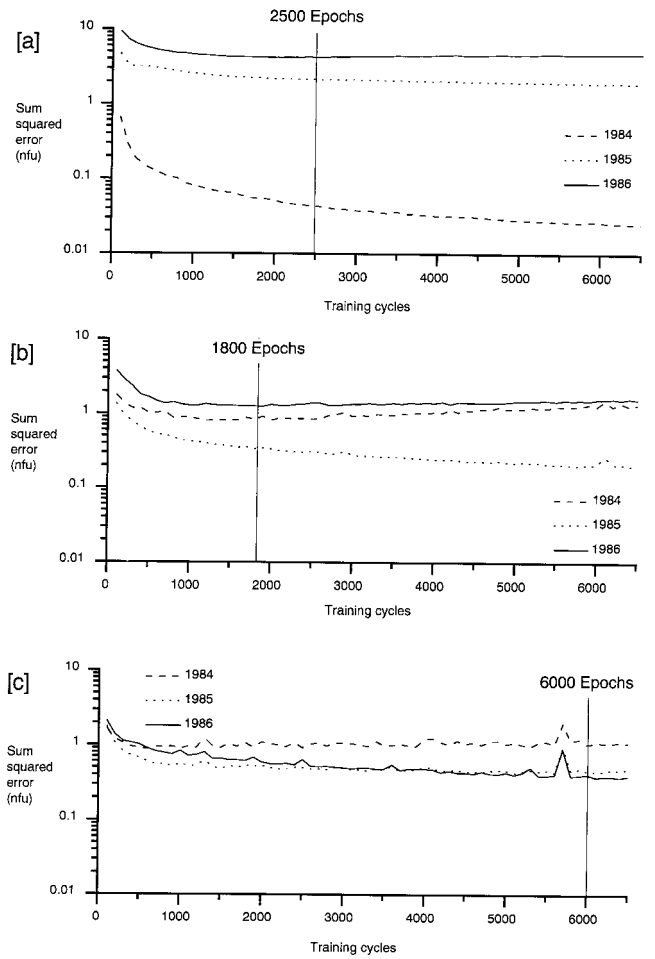

Figure 3 | Selection of optimal networks based on standard procedures using (a) 1984 (b) 1985 and (c) 1986 training data. Each vertical line indicates the chosen model.

solution was constructed from one annual data set and fitness evaluated with another. Low mutation probabilities were maintained throughout.

RESULTS

Tables 1 and 2 summarise the architectural results for each different procedure. Both pruning algorithms reduced the original network architecture (Table 1). Marked differences were created in the number and distribution of nodes. The input layer saw the greatest variation (3–15). The total number of connections in each solution ranged from 64 to 258, with mixed relationships between the number of nodes in each layer and the total number of connections. Magnitude based pruning maintained several input links (Figure 4). The most recent past river flow value (FLOW t-1) had the most connections. These three networks maintained several input links with current rainfall (RAIN t). The situation for other input links associated with earlier FLOW and RAIN data is less clear-cut and there is some degree of variation from network to network—although the 1984 model had more RAIN and FLOW links. Models built with 1985 and 1986 data had no input connections with CLOCK or PET, while the 1984 network maintained minor links with both. Skeletonization created networks with similar or fewer inputs but more connections (Figure 5). In all cases, several links were maintained with FLOW t-1 and t-2. The 1985 and 1986 models have various links with FLOW t-3 to t-6, whereas the 1984 model does not. The models for 1985 and 1986 maintained links with current rainfall (RAIN t), 1986 also with RAIN t-1. The 1984 and 1985 models have links to CLOCK, and the 1986 model has links with past PET values.

The GA-RPROP combination also produced extensive reductions in network architecture (Table 2). The most striking result is that the fittest networks all contained a full set of input nodes. Some input node mutation occurred but these features were not passed on to subsequent generations. It is unclear whether this is the result of low mutation probabilities, improved fitness performance from multiple inputs, or a spurious artefact associated with the training programme and node insertion procedure. Both hidden layers experienced a massive reduction in the number of hidden nodes, ranging from 2 to 12 in the first hidden layer, and from 2 to 5 in the second. The total number of connections varied from 49 to 269. There also appeared to be no explicit relation between the number of nodes in each layer and the total number of connections. These experiments are more difficult to interpret, and input relevance must be determined from an examination of the connection patterns.

Statistical interpretation

Because there was no definitive test to evaluate the success of each forecast a multi-criteria assessment was

Table 1 | Number of components in optimal pruned networks [percentage of original].

	BPROP training data	Number of nodes in input layer	Number of nodes in 1st hidden layer	Number of nodes in 2nd hidden layer	Total number of connections
Initial network	—	2	16	14	590
Magnitude based pruning	1984	15 [68.2]	14 [87.5]	8 [57.1]	104 [17.6]
Magnitude based pruning	1985	9 [40.9]	13 [81.3]	7 [50.0]	65 [11.0]
Magnitude based pruning	1986	8 [36.4]	13 [81.3]	6 [42.9]	64 [10.8]
Skeletonization	1984	3 [13.6]	11 [68.8]	9 [64.3]	141 [23.9]
Skeletonization	1985	8 [36.4]	13 [81.3]	11 [78.6]	258 [43.7]
Skeletonization	1986	9 [40.9]	9 [56.3]	11 [78.6]	191 [32.4]

Table 2 | Number of components in fittest genetic algorithm networks [percentage of original].

GA fitness evaluation data	RPROP training data	Number of nodes in input layer	Number of nodes in 1st hidden layer	Number of nodes in 2nd hidden layer	Total number of connections
Initial network	—	22	16	14	590
1984	1985	22 [100]	6 [37.5]	4 [28.6]	150 [25.4]
	1986	22 [100]	8 [50.0]	5 [35.7]	119 [20.2]
1985	1984	22 [100]	8 [50.0]	2 [14.3]	116 [19.7]
	1986	22 [100]	12 [75.0]	5 [35.7]	269 [45.6]
1986	1984	22 [100]	7 [43.8]	4 [28.6]	117 [19.8]
	1985	22 [100]	2 [12.5]	2 [14.3]	49 [8.3]

carried out. Eight global evaluation statistics were applied to each output:

- minimum and maximum: the largest negative error (under-prediction) and largest positive error (over-prediction);
- Standard Error of the Estimate (SEE);
- Sum-Squared Error (SSE);
- Higher-Order Error Function (S4E): to examine the model fit at peak flows, error values were raised to the fourth power and summed (Blackie & Eeles 1985);
- Mean Absolute Error (MAE): this is a global average where all deviations from the original data, positive or negative, are treated on an equal basis. Variations in sample size are accounted for and the statistic is not weighted towards high flow events.
- Root Mean Squared Error (RMSE)
- Coefficient Of Efficiency (COE)

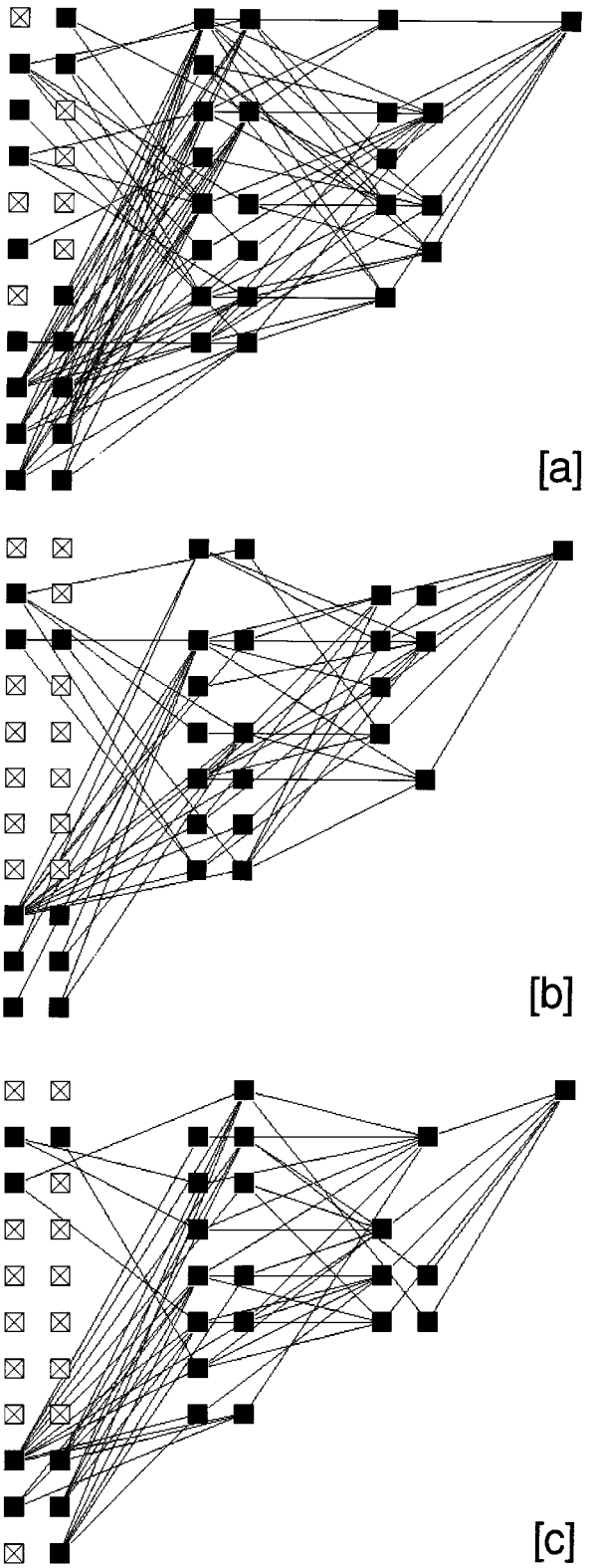

Figure 4 | Selected optimal network architectures from magnitude-based pruning using (a) 1984 (b) 1985 and (c) 1986 testing data.

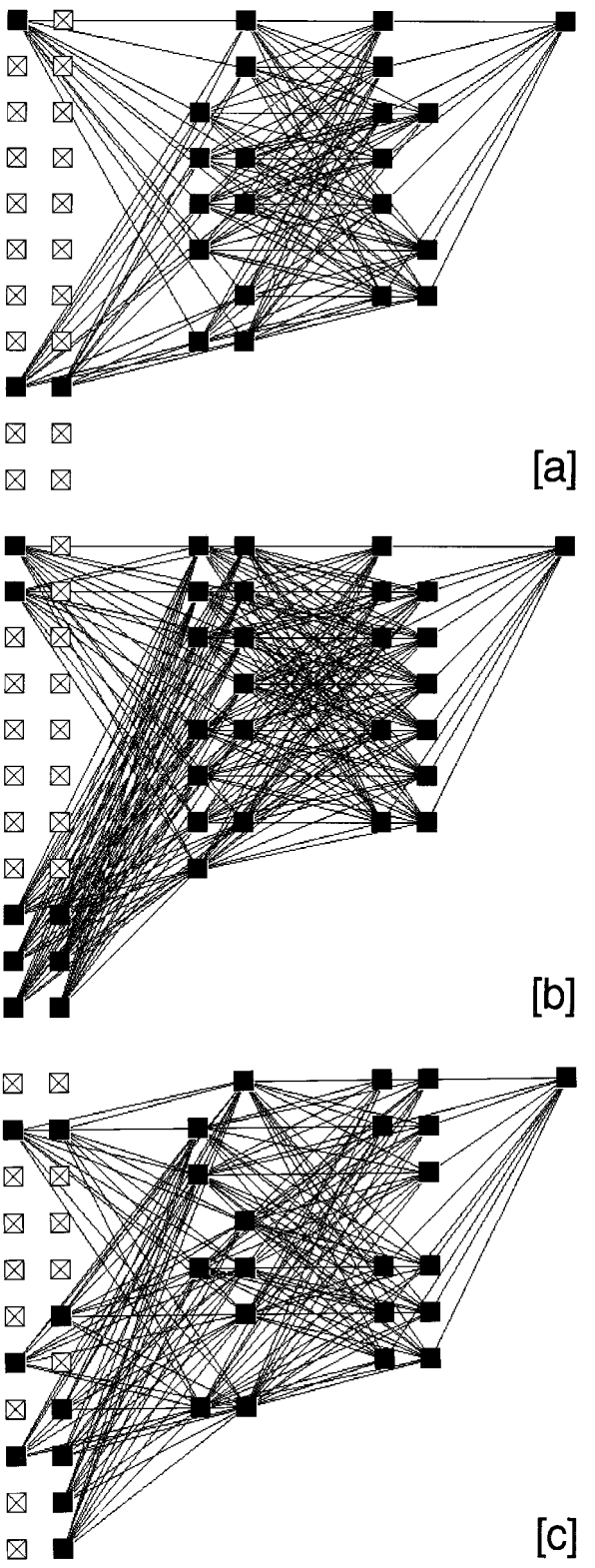

Figure 5 | Selected optimal network architectures from skeletonization using (a) 1984 (b) 1985 and (c) 1986 testing data.

The results (Tables 3 and 4) show no one best overall solution. The best result in each table is shaded. Comparing the pattern of best performing statistics revealed the following.

1. All models produced good results. The overall level of prediction is similar, with no strong evidence of overfitting, which validates the selection method.
2. There is no outright statistical winner. Different models appear to have different qualities; so in all cases the criteria for selection must be determined according to the application, and the use of alternative objective functions should be considered, e.g. specific to reservoir management, flood forecasting, or habitat preservation purposes.
3. The different training sets contained different types or amounts of information, which produced different levels of generalisation, for each situation. This will have strong repercussions on the use of individual modelling solutions built for one period and then applied to another.

The level of variation for each data set, exhibited between the different neural network solutions, was also investigated. High levels of variation reflect marked differences in the test statistic, which is indicative of dissimilar generalisation or poor modelling capabilities. This might be applicable on an annual basis, on a localised event basis, or on a combination of both. Table 5 compares between-model variation measured using the Coefficient of Determination (standard deviation expressed as a percentage of the mean).

Hydrograph interpretation

Plots of actual and predicted flows were inspected for bias in network performance. This is vital because it is possible to get significant statistical relationships on long time series, where the low flows are modelled accurately, but high flows are wrong. These plots were also used to check for a consistent temporal response. It was anticipated, for example, that there could be greater errors in forecasts involving winter snowmelt, which are rare occurrences in the training data.

Figure 6 shows hydrographs for three 50 hour periods in 1986. These illustrate (a) low flow, (b) medium flow and (c) high flow events. The three periods are discontinuous but occurred in the same hydrological season. The forecasting models were those that produced the best overall performance in the model building exercise.

DISCUSSION

Pruning algorithms and genetic algorithms provided multiple solutions of a similar, but not identical, nature. The simpler architectures were effective and this research has produced evidence to show which inputs were the most influential. The trial and error element is reduced.

Despite producing reasonable results, major variation in network complexities existed. This is both of scientific interest and a possible cause for concern. With no consistent outcome, it is possible to deduce that no optimal solution exists, and that what appear to be improved architectural solutions are in fact manifestations of a random sampling process with no real meaning in the arrangement of the network nodes and weights. However, it is also possible to conclude that the exact intricacies of the architecture are not that important, which in turn suggests that less effort should be expended on searching for an optimal solution when, for most practical purposes, a simple sub-optimal solution would be sufficient for forecasting and much quicker to obtain. More radical and extensive analytical experimentation, coupled with more detailed internal inspection of the final models, is required to test this hypothesis.

Statistical assessment showed no 'winning' solution. The magnitude-based pruning model trained on 1985 data was the overall leader, in terms of collective performance. The best genetic model was built from a combination of 1986 (training) and 1984 (fitness evaluation) data. This successful use of the 1984 data is controversial because these data are known to be 'information poor' and, all other things being equal, should therefore have given the weakest performance. In Table 5, high values for a particular method indicate variable results. S4E, in all but one instance, exhibited the greatest degree of variation, which

Table 3 | Statistical evaluation of optimal network forecasting based on (a) 1984, (b) 1985 and (c) 1986 data. Best results obtained from independent validation data shaded.

	Network trained with standard procedures			Network trained with magnitude based pruning			Network trained with skeletonization		
(a)	**BPROP training data**			**BPROP training data**			**BPROP training data**		
	1984	**1985**	**1986**	**1984**	**1985**	**1986**	**1984**	**1985**	**1986**
Min.	– 0.0276	– 0.1212	– 0.2800	– 0.0251	– 0.0958	– 0.0780	– 0.3021	– 0.0574	– 0.0694
Max.	0.0232	0.2748	0.3073	0.0244	0.2113	0.2798	0.1063	0.1518	0.1529
SEE	0.0022	0.0096	0.0108	0.0020	0.0073	0.0091	0.0079	0.0073	0.0090
SSE	0.0431	0.8393	1.0244	0.0345	0.4924	0.7760	0.5617	0.4938	0.7300
S4E	0.0000	0.0154	0.0257	0.0000	0.0057	0.0137	0.0100	0.0022	0.0032
RMSE	0.0022	0.0098	0.0108	0.0020	0.0075	0.0094	0.0080	0.0075	0.0091
MAE	0.0016	0.0047	0.0041	0.0013	0.0039	0.0046	0.0044	0.0043	0.0059
% COE	99.75	95.04	93.75	99.80	97.17	95.53	96.63	97.17	95.69
(b)	**BPROP training data**			**BPROP training data**			**BPROP training data**		
	1984	**1985**	**1986**	**1984**	**1985**	**1986**	**1984**	**1985**	**1986**
Min.	– 0.2998	– 0.17660	– 0.1299	– 0.2327	– 0.1713	– 0.1435	– 0.2298	– 0.1919	– 0.1621
Max.	0.0949	0.1250	0.2482	0.1978	0.1727	0.2424	0.1812	0.1322	0.1792
SEE	0.0153	0.0062	0.0072	0.0098	0.0069	0.0069	0.0096	0.0075	0.0078
SSE	2.1447	0.3345	0.4516	0.8526	0.4234	0.4205	0.8098	0.4933	0.5347
S4E	0.0355	0.0021	0.0068	0.0098	0.0032	0.0054	0.0101	0.0036	0.0042
MAE	0.0088	0.0030	0.0027	0.0048	0.0032	0.0031	0.0041	0.0035	0.0043
RMSE	0.0157	0.0062	0.0072	0.0099	0.0070	0.0069	0.0096	0.0075	0.0078
% COE	88.33	98.09	97.44	95.18	97.59	97.65	95.39	97.19	96.95
(c)	**BPROP training data**			**BPROP training data**			**BPROP training data**		
	1984	**1985**	**1986**	**1984**	**1985**	**1986**	**1984**	**1985**	**1986**
Min.	– 0.4307	– 0.3703	– 0.1740	– 0.2534	– 0.3148	– 0.1885	– 0.3238	– 0.3139	– 0.1946
Max.	0.1031	0.1408	0.1127	0.1940	0.1583	0.1839	0.1674	0.1866	0.2165
SEE	0.0218	0.0120	0.0068	0.0135	0.0118	0.0093	0.0152	0.0126	0.0104
SSE	4.2554	1.2682	0.4129	1.5940	1.2254	0.7818	2.0234	1.3978	0.9414
S4E	0.1892	0.0435	0.0028	0.0253	0.0275	0.0092	0.0569	0.0316	0.0114
MAE	0.0096	0.0043	0.0027	0.0056	0.0043	0.0039	0.0055	0.0048	0.0050
RMSE	0.0220	0.0120	0.0069	0.0135	0.0118	0.0095	0.0152	0.0126	0.0104
% COE	90.35	97.06	99.05	96.32	97.16	98.23	95.32	96.76	97.82

Table 4 | Statistical evaluation of fittest network forecasting based on (a) 1984, (b) 1985 and (c) 1986 data. Best results obtained from non-fitness-evaluation data shaded.

	GA fitness evaluation data					
	1984		**1985**		**1986**	
(a)	**RPROP training data**		**RPROP training data**		**RPROP training data**	
	1985	**1986**	**1984**	**1986**	**1984**	**1985**
Min.	– 0.3319	– 0.0740	– 0.2717	– 0.1797	– 0.4381	– 0.3800
Max.	0.2466	0.2538	0.1431	0.2167	0.0989	0.1567
SEE	0.0154	0.0165	0.0175	0.0154	0.0158	0.0142
SSE	2.1328	2.4306	3.9664	2.0891	2.3532	1.8277
S4E	0.0419	0.0451	0.0236	0.0145	0.1073	0.0564
MAE	0.0106	0.0025	0.0168	0.0010	0.0094	0.0086
RMSE	0.0156	0.0167	0.0213	0.0155	0.0164	0.0145
% COE	87.28	85.41	83.54	87.23	86.57	89.15
(b)	**RPROP training data**		**RPROP training data**		**RPROP training data**	
	1985	**1986**	**1984**	**1986**	**1984**	**1985**
Min.	– 0.5087	– 0.2534	– 0.4460	– 0.2472	– 0.6106	– 0.5540
Max.	0.1267	0.1761	0.1421	0.1605	0.1034	0.1462
SEE	0.0173	0.0124	0.0276	0.0145	0.0246	0.0195
SSE	2.6134	1.3399	6.8210	1.8493	5.3159	3.3369
S4E	0.1779	0.0159	0.2078	0.0193	0.4621	0.2722
MAE	0.0078	0.0067	0.0190	0.008	0.0123	0.0084
RMSE	0.0173	0.0124	0.0279	0.0145	0.0246	0.0195
% COE	85.08	92.36	61.98	89.50	69.77	80.94
(c)	**RPROP training data**		**RPROP training data**		**RPROP training data**	
	1985	**1986**	**1984**	**1986**	**1984**	**1985**
Min.	– 0.6847	– 0.4238	– 0.6684	– 0.3692	– 0.7737	– 0.7122
Max.	0.1419	0.1738	0.1581	0.1702	0.1035	0.1572
SEE	0.0350	0.0199	0.0359	0.0236	0.0433	0.0381
SSE	10.7149	3.4832	11.6565	4.9099	16.4931	12.7187
S4E	2.1035	0.1782	1.3823	0.1922	4.1816	2.7851
MAE	0.0114	0.0084	0.0189	0.0106	0.0143	0.0117
RMSE	0.0350	0.0199	0.0365	0.0237	0.0434	0.0381
% COE	75.21	91.93	73.82	88.71	61.89	70.58

Table 5 | Coefficient of determination (%).

	Standard procedure and pruning experiments			Genetic algorithm experiments		
Test data	**1984**	**1985**	**1986**	**1984**	**1985**	**1986**
Min.	87.84	27.45	31.03	48.12	35.30	27.52
Max.	62.67	29.45	23.28	33.59	17.87	17.08
SEE	43.45	32.65	33.42	7.05	30.38	27.52
SSE	61.14	78.43	72.41	31.01	59.85	49.25
S4E	102.14	115.35	129.13	67.98	87.12	86.33
MAE	38.17	44.82	37.39	70.81	43.80	29.04
RMSE	43.14	33.65	33.68	14.45	30.75	27.55
% COE	2.10	3.17	2.62	2.20	14.79	14.72

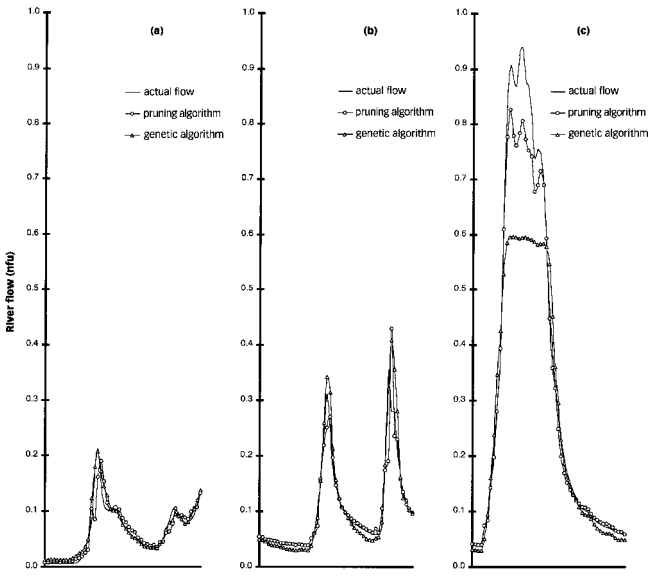

Figure 6 | Three 50 hour hydrograph plots taken from Autumn/Winter 1986: actual flows (solid line); best performing pruning model predictions (solid line with circles); and best performing genetic algorithm model predictions (solid line with triangles).

is interesting because it is this statistic that places particular emphasis on the model fit at peak flows. This means that it is the fit of the various neural network models to such phenomena that exhibits the greatest level of variation across the numerous different solutions. Percentage COE produced the least amount of variation per test data set. It was therefore unable to offer sufficient differentiation between the numerous neural network models. Some important between-method comparisons can also be made for various statistical measures related to each annual data set. In various instances, marked similarities can also be observed between the results obtained from testing with 1985 and 1986 data, and marked differences likewise observed between these two results and those obtained for 1984, the drought year.

The three hydrographs, Figure 6 a–c, contain a wealth of additional information about the underlying functions. Most low flow situations have been modelled well. Pruning and genetic algorithms produced similar, accurate results that are sufficient for this section of the solution surface. But the genetic model still generates a noticeable number of underpredictions at low flow levels, and when the level of flow is falling. Small to medium events are also modelled in an acceptable manner, but there are problems in the timing and magnitude of peak predictions. Pruning generates greater peak flow errors and these predictions are all late. The genetic model is better in medium event situations. The high flow event illustrates the principal differences between these models.

Pruning was the better of the two with reasonable timing but poor level prediction. The genetic model seriously underpredicted peak flow. These observations confirm the statistical results. Low flow and limited change situations are modelled quite well. Peak flow event forecasts could be improved, and considerable variation exists between the different modelling solutions and the manner in which these items are modelled. It must be remembered that direct comparison between the pruned and genetic models is limited because one model was produced from a desire to create an optimal solution, whereas the other model was produced from a desire to create an optimal architecture based on a fixed level of error. The question remains as to what extent these various differences in output can be attributed to differences in the method of model creation. For example, did the use of a batch update procedure and a fixed stopping condition prevent the genetic model from producing more accurate high flow prediction? Questions of this nature are the subject of further research.

In these studies, development overheads and application run times were not considered to be the most important issue. Inclusion of the pruning algorithms did not make a significant difference to the overall modelling process although the genetic algorithm investigations did take somewhat longer to run. To provide a more objective assessment, computational performance-related evaluation criteria will be established in the next stage of this research. These criteria will consider whether or not the additional computational effort associated with the implementation of pruning algorithms and genetic algorithms is worthwhile, given that improvement in prediction was at best slight, and in certain instances prediction was worse. These measures would also need to discriminate between the extra time that is taken to develop a less complicated architecture and the benefits of faster simpler neural network solutions. The tools and algorithms in these initial investigations were applied to a simple function and resulted in massive architectural savings. It is therefore anticipated that the application of these automated model-building procedures to larger and more complicated problems will generate parsimonious solutions, with a lower computational burden, which could be orders of magnitude faster. Faster solutions would be important for bootstrapping statistical population distributions and producing confidence limits, or where the modelling solution is applied to large data sets for long periods of time, for example in the prediction of detailed surface impacts related to global warming.

Placing a reduced *a priori* knowledge-based starting point on the modelling solution was considered inappropriate for the purposes of this research. Instead, the initial architecture that was adopted comprised a large network that had been used in previous empirical experimentation and which was thought to contain numerous redundant components. This initial structure also had full connection, which is the prevalent default condition, with all nodes in each layer being connected to all nodes in the adjacent layers. So removal of superfluous connections would also demonstrate that the standard setup is often a sub-optimal solution with non-essential items that handicap computational performance. More efficient starting points might be obtained using structures based on first principles but this would also create an imposed solution, that might not exist within the data set, or be supported to a great extent, or could be at an inappropriate level of generalisation for transfer to other data sets. Moreover, pruning algorithms require an excess from which items can be removed over time, and with the genetic package it is important to have a broad set of starting conditions in order to search for optimal solutions in all regions of the solution space. It would, however, be possible in the latter case to perform a limited amount of pre-training and thus generate a semi-trained starting point from which a fitter initial population could then be created. This procedure would help to speed up the convergence process and the software provides for various options. The science of neural network hydrological modelling is still new and with information gained from greater experience on network behaviour and architectures it might nevertheless be feasible at some later date to relate the starting conditions to match a specified hydrological problem, e.g. using information gained from saliency analysis.

CONCLUSIONS

From a hydrological perspective these experiments showed that simple neural network models can be

produced and evaluated using automated techniques. Several thousand models were created and tested using batch programs and overnight runs, a cost–benefit advantage for model development and application times. Acceptable results were produced from a limited number of input measurements. These experiments also showed that the procedures effectively fused different types of data from various sources for different temporal periods. Many hydrological models focus on peak flow prediction. The neural network approach offers a complete hydrological modelling solution, and provides excellent low flow predictors. This has potential advantages for water resource applications in drought regions, reservoir management in drought periods, and river balance planning and supply operations.

Iterative learning was extended in this research to create a more complex procedure that focused on the progressive removal of 'unimportant components' in a destructive cycle of training and pruning. Network reconstruction sequences and fitness testing were then investigated within an automated model-building environment. In some cases the evidence suggests that a more suitable network architecture, with improved generalisation capabilities, was found. In all cases there was a substantial reduction in the network architecture, producing simpler neural network models, with fewer computational overheads. The removal of non-essential inputs, another characteristic of the pruned networks, has clear implications for data collection and information processing times.

These experiments also highlighted the fact that there is still no reliable scoring system that overcomes the difficulties of measuring peaks and troughs, or performs event-based separation of appropriate statistical descriptors. Heterogeneous evaluation, with appropriate weightings based on specific end-user requirements, offers one possible method to achieve this goal. But the potential application of all such subjective approaches must be examined in a rigorous and comprehensive manner. There is also a pressing need for the creation of dedicated software programs that can perform multi-criteria assessment, perhaps in an interactive manner, and with direct links to a data analysis toolbox.

ACKNOWLEDGEMENTS

SNNS was developed at the Institute for Parallel and Distributed High Performance Systems, University of Stuttgart. ENZO was developed at the Institute for Logic, Complexity and Deduction Systems, University of Karlsruhe. Upper River Wye data were collected by the UK Institute of Hydrology.

REFERENCES

Abrahart, R. J. 1998 Neural networks and the problem of accumulated error: an embedded solution that offers new opportunities for modelling and testing. In *Hydroinformatics '98* (Proceedings of the Third International Conference on Hydroinformatics, Copenhagen, Denmark, 24–26 August 1998) (ed. Babovic, V. M. & Larsen, L. C.), vol. 2, pp. 725–731. Balkema, Rotterdam.

Abrahart, R. J. & Kneale, P. E. 1997 Exploring neural network rainfall-runoff modelling. In *Proceedings of the Sixth National Hydrology Symposium, University of Salford, UK, 15–18 September 1997*, pp. 9.35–9.44.

Abrahart, R. J. & See, L. 1998 Neural Network vs. ARMA Modelling: constructing benchmark case studies of river flow prediction. In *GeoComputation '98* (Proceedings of the Third International Conference on GeoComputation, University of Bristol, United Kingdom, 17–19 September 1998). (CD-ROM.)

Bathurst, J. 1986 Sensitivity analysis of the Systeme Hydrologique Europeen for an upland catchment. *J. Hydrol.* **87**, 103–123.

Beven, K., Kirkby, M. J., Schofield, N. & Tagg, A. F. 1984 Testing a physically-based flood forecasting model (TOPMODEL) for three U.K. catchments. *J. Hydrol.* **69**, 119–143.

Blackie, J. R. & Eeles, W. O. 1985 Lumped catchment models. In *Hydrological Forecasting* (ed. Anderson, M. G. & Burt, T. P.), chapter 11. John Wiley, Chichester, UK.

Cheng, X. & Noguchi, M. 1996 Rainfall-runoff modelling by neural network approach. In *Proceedings of International Conference on Water Resources and Environment Research: Towards the 21st Century, Kyoto, Japan, 29-31 October 1996*, vol. 2, pp. 143–150.

Fischer, M. M. & Gopal, S. 1994 Artificial neural networks: a new approach to modelling interregional telecommunication flows. *J. regional Sci.* **34**, 503–527.

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

Quinn, P. F. & Beven, K. J. 1993 Spatial and temporal predictions of soil moisture dynamics, runoff, variable source areas and evapotranspiration for Plynlimon, mid-Wales. *Hydrological Processes* **7**, 425–448.