Hydrol. Earth Syst. Sci. Discuss., 4, 287–326, 2007
www.hydrol-earth-syst-sci-discuss.net/4/287/2007/

Hydrology and Earth System Sciences Discussions

Papers published in *Hydrology and Earth System Sciences Discussions* are under open-access review for the journal *Hydrology and Earth System Sciences*

Neural network emulation of a rainfall-runoff model

R. J. Abrahart[1] **and L. M. See**[2]

[1]School of Geography, University of Nottingham, Nottingham NG7 2RD, UK
[2]School of Geography, University of Leeds, Leeds, LS2 9JT, UK

Received: 8 January 2007 – Accepted: 9 February 2007 – Published: 22 February 2007

Correspondence to: R. J. Abrahart (bob.abrahart@nottingham.ac.uk)

Abstract

The potential of an artificial neural network to perform simple non-linear hydrological transformations is examined. Four neural network models were developed to emulate different facets of a recognised non-linear hydrological transformation equation that possessed a small number of variables and contained no temporal component. The modeling process was based on a set of uniform random distributions. The cloning operation facilitated a direct comparison with the exact equation-based relationship. It also provided broader information about the power of a neural network to emulate existing equations and model non-linear relationships. Several comparisons with least squares multiple linear regression were performed. The first experiment involved a direct emulation of the Xinanjiang Rainfall-Runoff Model. The next two experiments were designed to assess the competencies of two neural solutions that were developed on a reduced number of inputs. This involved the omission and conflation of previous inputs. The final experiment used derived variables to model intrinsic but otherwise concealed internal relationships that are of hydrological interest. Two recent studies have suggested that neural solutions offer no worthwhile improvements in comparison to traditional weighted linear transfer functions for capturing the non-linear nature of hydrological relationships. Yet such fundamental properties are intrinsic aspects of catchment processes that cannot be excluded or ignored. The results from the four experiments that are reported in this paper are used to challenge the interpretations from these two earlier studies and thus further the debate with regards to the appropriateness of neural networks for hydrological modelling.

1 Introduction

The last decade has witnessed a virtual explosion of neural network (NN) modelling activities throughout the hydrological sciences. It is readily apparent from the increasing number of published case studies that the development of data-driven solutions based on the use of neural tools or smart technologies is being trialled and tested in most sectors of hydrological modelling and hydraulic engineering. Numerous extended descriptions exist and for detailed summaries the interested reader is referred to the following papers: ASCE (2000a, b); Maier and Dandy (2000); Dawson and Wilby (2001) and edited volumes: Govindaraju and Rao (2000); Abrahart et al. (2004). NN continue to make enormous strides in their struggle to become established as recognised tools that offer efficient and effective solutions for modelling and analysing the behaviour of complex dynamical systems. Time series forecasting has been a particular focus of interest and superior performing models have been reported in a diverse set of fields that include rainfall-runoff modelling (ASCE, 2000a, b; Dawson and Wilby, 2001; Birikundavy et al., 2002; Campolo et al., 2003; Huang et al., 2004; Riad et al., 2004; Hettiarachchi et al., 2005; Senthil Kumar et al., 2005) and sediment prediction (Abrahart and White, 2001; Nagy et al., 2002; Yitian and Gu, 2003; Kisi, 2004, 2005; Bhattacharya et al., 2005). Two recent catchment studies have nevertheless questioned the use of such tools for non-linear hydrological modelling purposes. Gaume and Gosset (2003) and Han et al. (2007) concluded that: (1) that for short term forecasting purposes neural solutions offered no real advantages over traditional linear transfer functions; (2) that the demands and complexities involved in the development of neural solutions made them difficult to use and therefore "uncompetitive" (Han et al., 2007, p. 227); (3) that there is still much to be done to improve our understanding about the uncertain nature and hydrological characteristics of neural forecasters "before [such mechanisms] could be used as a practical tool in real-time operations" (Han et al., 2007, p. 228); and (4) that the potential merit of putting further resources into the development of black box computational intelligence methodologies such as feed-forward neural networks remains questionable since "the quest for a universal model requiring no hydrological expertise might well be hopeless" (Gaume and Gosset, 2003, p. 705). This paper examines the modelling assumptions and reported interpretations that are recorded in the two listed papers and puts forward a counter argument based on a set of emulation experiments that are designed to establish the real relationship

HESSD
4, 287–326, 2007

that exists between neural network solutions and traditional linear transfer functions within a hydrological modelling context.

2 Two critical hydrological studies

Non-linear transformation is a fundamental aspect of most hydrological modelling operations and real-time forecasting applications. Gaume and Gosset (2003) reported that a NN model can sometimes produce results that are similar to a weighted linear transfer function (WLTF) model. Their first comparison exercise involved one-step-ahead discharge forecasting models developed for the Marne River in France. Inputs comprised discharge records and included the last observed record at their point of forecast. The NN model showed marginal improvements over a WLTF model developed on an identical set of inputs and this result was attributed to the near-linear nature of the process that was being modelled comprising "mainly small hydrograph shifts and tributary flow additions" (p. 700). The two approaches as expected exhibited close behaviour and produced similar outputs in terms of predicted flood hydrographs; however, the NN estimates were usually closer to the observed measurements than their corresponding WLTF outputs. The rising and falling limb predictions were very close; during other phases, such as peak floods and low water periods, the NN showed marginal improvement, and both models showed the same inability to anticipate increasing discharge. Their second comparison exercise involved one-step-ahead discharge forecasting models developed for the Le Sauzay River (81 km^2) in France. Inputs comprised discharge, potential evaporation and precipitation records and included the last observed record at their point of forecast. The NN model showed marginal improvements over a WLTF model developed on a similar but not identical set of inputs. Their results were also compared to a conceptual model but that part of their assessment exercise is not considered in the present paper. The rainfall-runoff process is often described as being highly non-linear, particularly in small catchments such as the one that was studied, and a non-linear representation is assumed to have been encapsulated in the selected modelling inputs and outputs. The two approaches in contrast to expectations exhibited close behaviour. Their predicted outputs produced similar hydrographs, in which the peak flood events, as well as the rising and falling limb predictions, appeared to be more or less equivalent. The NN model was nevertheless once again found to be more accurate than the WLTF model during low water periods. The two experiments in this respect exhibited similar results but such outcomes were unsurprising as the number of low water period measurements greatly outnumbered the records for other types of event. The poor relative improvement in forecasting outputs related to the use of neural solutions was attributed to potential noise in the observed hydrological records or to the limited lengths of the datasets upon which the various models were developed.

Han et al. (2007) in the second paper reported that a NN model offered no advantages for short-term forecasts over a weighted linear transfer function (WLTF) model. Their comparison exercise involved a number of models developed on similar but not identical datasets for Bird Creek (2344 km^2) in the USA. It also compared models developed on eight different forecasting horizons and involved testing a number of different configurations: NN "Type A" entailed building a dedicated "direct" forecasting model for each required forecasting horizon, i.e. eight-step-ahead forecasting required eight individual models; NN "Type B" entailed building a single model that produced eight outputs, i.e. one output related to each of the eight required forecasting horizons (e.g. Toth et al., 2000); NN "Type C" entailed building one "iterative" model that preserved the same input structure and had one output, but which was run with consecutive updating of inputs, such that the last output produced in the previous time step was incorporated, i.e. "marching forward scheme" (e.g. Abrahart, 1998; Varoonchotikul, 2003). Inputs included discharge and precipitation records and the last observed record at their point of forecast. For short range forecasts the NN model was found to be inferior to the WLTF model; however, for longer-term forecasts the opposite situation arose in that the NN model did show some aspects of superior performance. This is a medium sized catchment and a non-linear representation of the rainfall-runoff process is once again assumed to have been encapsulated in the selected modelling inputs and outputs. The use of eight individual direct forecasting models was considered to have produced the best overall result but with the caveat that this approach would of course require much more effort on the part of both developer and user. The pattern of statistical deterioration associated with the use of longer forecasting horizons was not consistent during the final stages. However, the extent to which this result is a function of the evaluation statistic used as opposed to something more fundamental is unclear.

To further consider the earlier reported findings this paper will test the capabilities of a NN to produce a non-linear solution using a series of controlled experiments based on an ideal non-linear hydrological modelling problem set in a data-rich environment. NN solutions were developed to emulate a recognised non-linear hydrological model: the Xinanjiang Rainfall-Runoff Model (Zhao et al., 1980). This model was formulated as a single equation that had a small number of input variables and no temporal component. No specific river records were involved; hence no sweeping generalisations based on the results of individual catchment studies or the particularities of observed datasets will be produced. The inputs to the rainfall-runoff transformation equation were produced using a statistical random pattern generator. This cloning operation facilitated a direct comparison with the computed mathematical relationship whilst at the same providing more general information on the power of a NN to model non-linear relationships. The mathematical relationship was transparent and the nature of the relationship is clearly non-linear. The results are compared to traditional Least Squares Multiple Linear Regression (MLIN) models developed on the same datasets. Neural solutions were also developed on a smaller number of input variables: omission and conflation of the original inputs was used to reveal the changing nature of different models and their computed outputs. The final experiment used derived variables to model intrinsic but otherwise concealed internal relationships that are of hydrological interest.

HESSD

4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

EGU

3 Emulation of the Xinanjiang rainfall-runoff model

Van den Boogaard and Kruisbrink (1996) list eight possible integration options for hybrid neural network modelling. The first such option is "reproduction modelling" which is hereinafter refereed to in computer science terms as the process of "emulation". The aim of an emulator, in the most general sense, is to duplicate the functions of one system with a different system, so that the second system appears to behave like the first system. However, unlike a simulator, it does not attempt to precisely model the state of the device being emulated; it only attempts to reproduce its behaviour. The act of NN emulation or cloning is in this instance the process of using a neural model to mimic an existing equation-based solution – including its inherent imperfections. In addition to offering rapid improvement in processing speed and data handling capabilities, for instance in an integrated optimisation procedure (Rogers and Dowla, 1994; Solomatine and Avila Torres, 1996), emulators can be used to reduce existing model calibration time through building a response surface that relates internal parameters to original output (Liong and Chan 1993), or to sidestep potential difficulties associated with the calibration of more popular models under conditions of limited discharge and sediment concentration records (Hsu et al., 2003). NN emulators have also been used for the detection of important internal processes occurring inside a conceptual model (Wilby et al., 2003). Emulators can be constructed to include additional variables or to omit certain variables in those instances where one or more standard inputs are not available or withheld. It is also possible to mimic the internal functions of an existing model, for model reduction purposes, or for rapid prototyping, sensitivity analysis and bootstrapping operations. Less obvious is the use of neural emulators to mimic spatial distributions, thus making redundant our existing problems of storing and accessing copious amounts of spatial input, and enabling models to switch from file-based data retrieval (slower) to chip-based data computation (faster) operations.

The Xinanjiang Rainfall-Runoff Model (named after the river to which it was first applied) was developed in 1973 and first published in 1980 (Zhao et al., 1980; Zhao, 1992). This is a semi-distributed conceptual rainfall-runoff forecasting tool, which was designed for use in humid and semi-humid regions, and is based on the concept of runoff formation on repletion of storage i.e. runoff is not produced until the soil moisture content of the aeration zone reaches field storage capacity and thereafter runoff equals rainfall excess without further loss. The model has been applied with success to large areas including all of the agricultural, pastoral and forested lands (except for the loess) of China (Zhao and Liu, 1995, p. 230). This model has a small number of parameters, its structure and components have strong physical meaning, and these factors in combination make it a popular tool for hydrological modelling. The basic model has experienced numerous internal modifications e.g. modified soil moisture storage component (Jayawardena and Zhou, 2000). The model has also been coupled to mesoscale precipitation forecasts where it produced encouraging flood simulation outputs (Lin et al., 2006). The non-complicated nature of this model continues to make it a popular choice for hydrological experimentation e.g. for testing intelligent calibration procedures (Cheng et al., 2002, 2006) or for distributed modelling purposes (Su et al., 2003; Chen et al., 2007). This model has also been incorporated into nationwide forecasting methodologies, e.g. USA National Weather Service River Forecasting System (M. Kane, Riverside Technology Inc., personal communication). The ARNO Rainfall-Runoff Model (Todini, 1995) (which was derived from the Xinanjiang Model) has been incorporated into a climate model (Dümenil and Todini, 1992); the NUARNO Model (which is based on the ARNO Model) is an integrated part of the UK NERC-ESRC Land Use Programme Decision Support System where it is used to predict the direction and magnitude of the hydrological response that results from proposed changes in land-use (Adams et al., 1995, p. 56–58).

In its simplest form the model comprises a single equation:

$$R = P - (Wm - Wo) + Wm\left[\left(1 - \frac{Wo}{Wm}\right)^{\frac{1}{1+b}} - \frac{P}{(1+b)Wm}\right]^{1+b} \tag{1}$$

where R is runoff, P is precipitation, Wm is the maximum field storage capacity, Wo is the initial field storage capacity and b is an exponent representing non-uniform spatial distribution. The term in square brackets can also be treated as an independent item [AUX]:

$$\text{If AUX} \leq 0 \qquad R = P - (Wm - Wo) \tag{2}$$

$$\text{If AUX} > 0 \qquad R = P - (Wm - Wo) + Wm\,(\text{AUX})^{1+b} \tag{3}$$

$$\text{If } Wo = Wm \qquad R = P \tag{4}$$

A dataset of random input variables was created comprising 5000 records. The initial values for precipitation [P], maximum soil water [Wm] and the curve fitting exponent [b] were random samples taken from uniform distributions computed in MINITAB. Each distribution was generated between fixed limits:

- Precipitation (mm/h) between 0 and 50
- Maximum soil water (mm) between 50 and 100
- b (dimensionless) between 0.1 and 0.5

These upper and lower limits were considered reasonable based on the recognised need to have a broad range of different input scenarios. Initial soil water [Wo], the fourth input variable, was assigned a random number generated between the "half-full" and "maximum" soil water values.

A dedicated software program was written to calculate the numerical rainfall-runoff response. This program was written to permit a number of different options to be implemented including the addition of noise or error based on a set of independent external records. It also performed a linear standardisation of the input variables and output responses that were scaled to a fixed range [0–1]. To minimise the number of data conversions required all computed runoff values are henceforth reported in terms of standardised discharge units (sdu) or standardised ratio units (sru). The same calculated numerical dataset was used to explore a number of different models and relationships. In each case a linear comparison was performed that involved the construction of a MLIN model developed on the same predictors and predictands.

Seven different performance statistics were used for comparing the output results. HydroTest was used to perform the required calculations; further particulars and the relevant equations that describe each metric can be found on that web site (http://www.hydrotest.org.uk) and appear in its related paper (Dawson el al., 2007). Model performance was assessed on the basis of two absolute statistics, two relative statistics and three dimensionless indices: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Relative Error (MARE), Mean Squared Relative Error (MSRE), Coefficient of Efficiency (CofE), Coefficient of Determination (RSqr) and Index of Agreement (IoAd).

4 Neural network experiments

Four individual experiments were performed in a controlled environment. Each NN model contained 2 hidden layers and each hidden layer contained 12 hidden units. No attempt was made to develop an optimal architectural configuration that could deliver the best permitted set of modelling output accuracies. No attempt was made to develop a minimal architectural configuration that might equate to a parsimonious modelling solution. The adopted method was instead to develop a series of complex models that contained redundant component parameters and which allowed for no missed opportunities. The justification for such actions is twofold:

1. It is important to distinguish between the number of available parameters that remains fixed and the number of effective parameters that increases during training (Weigend et al., 1992). The number of useful parameters at the start of the calibration process is zero, since having been subjected to a random initialisation

HESSD
4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

EGU

process, such parameters have yet to learn anything useful with regard to either problem solving activities or the requirements of the user. It is not a simple question of counting up the initial number of parameters in a model. It is a matter of needing to have a sufficient number of potential parameters at the start of the learning exercise that could, if required, be used to support the modelling process. Too few a number will put constraints on what can or cannot be achieved and could produce needless simplifications or unwarranted generalities.

2. Neural networks are designed to model continuous nonlinearities in dynamical systems, or with special modification, piecewise continuous functions (Selmic et al., 2002). Two hidden layers will permit more complicated target functions, and their related solution surfaces, to be modelled in an efficacious manner (Sarle, 1997). Each unit in the second hidden layer will enable a separate peak or trough to be fitted. However, under difficult situations, two hidden layer solutions are also capable of approximating discontinuous functions in a purposeful context (Sontag, 1992).

4.1 Experiment 1

The Stuttgart Neural Network Simulator (SNNS: Zell et al., 1995) was used to develop a two-hidden-layer feedforward NN. Each processing unit was connected to all processing units in the adjacent layers and full connection was maintained throughout. The input layer had four input units that corresponded to the input variables: precipitation [P]; initial soil water [Wo]; maximum soil water [Wm]; and the curve fitting exponent [b]. Each of the two hidden layers contained twelve hidden processing units and there was one output unit in the final output layer for the single output variable [R]. NN initialisation involved assigning random weights to all connections and processing unit biases. The permitted range of random weights was set at ± 1. The 5000 patterns of standardised variables were split into two equal groups, one for training purposes, and the other for split sample testing operations. The network was trained using "backpropagation without momentum"; the accepted shorthand for such solutions is a Backpropagation Neural Network (BPNN). The learning rate was set and adjusted as per Table 1. Training material was presented to the network in random order and training was stopped at 8000 epochs. Error reduction expressed in terms of Sum Squared Error (SSE) was observed to flatten out after 2000 epochs which suggested broad-scale convergence. This use of fixed stopping conditions does not prevent overfitting but substantial deficiencies in such respects were considered improbable due to: (1) the smooth nature of the mathematical function that was being emulated; (2) the comprehensive nature of the numerical representations upon which the models were developed and tested; and (3) the achievement of similar performance statistics on the two split sample datasets.

4.2 Experiment 2

SNNS was used to construct a second BPNN model. Experiment 2 differs from Experiment 1 in that the curve fitting exponent [b] which is designed to account for a non-uniform spatial distribution of responses was omitted from the initial set of four inputs. The input layer had three input units that corresponded to: precipitation [P]; initial soil water [Wo]; and maximum soil water [Wm]. There was one output unit in the final output layer for the output predictand [R]. This experiment produces a simpler model in which the need to include a difficult to determine curve fitting exponent is eliminated. The model development and testing process was in all other respects identical to that described for Experiment 1.

4.3 Experiment 3

SNNS was used to construct a third BPNN model. Experiment 3 differs from Experiment 2 in that an even simpler model was created: it had two input units in the input layer corresponding to the input variables precipitation [P] and "soil water ratio" (initial soil water [Wo] divided by maximum soil water [Wm]). There was one output unit in the final output layer for the single output predictand [R]. The two soil water input measurements were thus conflated into a single ratio and the curve fitting exponent was again omitted. This experiment represents a further loss of detailed hydrological information and represents something of a minimalist position with regard to what can or cannot be modelled without resorting to the development of a simple "rating curve". The model development and testing process was in all other respects identical to that described for Experiments 1 and 2.

4.4 Experiment 4

SNNS was used to construct a fourth BPNN model. This exploration used derived variables to model intrinsic but otherwise concealed internal relationships that are of hydrological interest. This model had two input units in the input layer, corresponding to the derived input variables "soil water ratio" (initial soil water divided by maximum soil water [*Wo/Wm*]) and "input-storage ratio" (precipitation divided by maximum soil water [*P/Wm*]). There was one output unit in the final output layer for the single derived output predictand "runoff ratio" (runoff divided by precipitation [*R/P*]). Each input and output variable is a dimensionless index that is of hydrological significance so it is interesting to model the non-linear relationship that is occurring between these factors inside the Xinanjiang Rainfall-Runoff Model. The model development and testing process was in all other respects identical to previous experiments. The "captured transfer function surface" was also visualized: (i) to help understand the exact nature of the black-box modelling solution that has been encapsulated; and (ii) to provide additional insights into the underlying hydrological processes and relationships. Figure 9 shows the distribution of points in input space. Two interpolated output surfaces were constructed in input space using inverse distance weighting based on (1) the original calculated outputs as depicted in Fig. 10 and (2) the neural network outputs as depicted in Fig. 11.

5 Results

BPNN training programme results for the four reported experiments are provided in Table 1. MLIN parameter coefficients are provided in Table 2. HydroTest BPNN and MLIN performance evaluation statistics are provided in Table 3. Figures 1, 3, 5 and 7 provide BPNN scatterplots of actual against predicted values for Experiments 1–4. Figures 2, 4, 6 and 8 provide companion MLIN scatterplots for the same experiments. Training [A] and testing [B] dataset outputs are provided in each case; BPNN outputs are for the final product that was created during the last stage of the learning and development process i.e. trained for 8000 epochs. The superior performance of the four non-linear BPNN solutions, in contrast to the poor relative performance of their linear MLIN counterparts, is apparent throughout most sectors of each individual experiment.

5.1 Experiment 1

The output results for Experiment 1 are illustrated in Figs. 1 and 2. BPNN (Figs. 1a, b) outputs exhibit near-perfect agreement between the expected and predicted runoff values. The two scatterplots thus demonstrate that neural solutions are able to emulate this particular hydrological function and can perform a non-linear modelling operation in an efficacious manner. Figures 2a, b contain the corresponding results for MLIN. The scatterplots in this instance reveal a curvilinear profile that is "twisted" around the "line of perfect agreement". The higher and lower level values are underpredicted whilst the central region values are overpredicted. Moreover, the lower level values exhibit the highest levels of spread. The resultant pattern of error indicates that the linear model has failed to capture important non-linearities. These findings are reflected in the evaluation statistics provided in Table 3. MLIN, in comparison to BPNN, exhibits poorer performance on each of the seven evaluation metrics. The two split sample datasets were observed to produce similar outputs in each case which is indicative of unbiased solutions and minimal potential overfitting.

HESSD
4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

Title Page
Abstract | Introduction
Conclusions | References
Tables | Figures
◀ | ▶
Back | Close
Full Screen / Esc
Printer-friendly Version
Interactive Discussion

EGU

5.2 Experiment 2

SSE for the trained model was much higher than that obtained in Experiment 1, levelling off at a little under 1.14 sdu, compared to 0.04 sdu. The output results for Experiment 2 are illustrated in Figs. 3 and 4. BPNN (Figs. 3a, b) outputs exhibit a clear trend, but the results are somewhat less impressive in comparison to the results that were produced from the “four input variable” model – especially in the uppermost (underestimated) and lower (minor scattering) sections of the scatterplots. Nevertheless, the omission of a principal non-linear component meant that there was less information contained within the input variables. The end result, however, is still considered to be a reasonable non-linear approximation. Figures 4a, b show the corresponding results for MLIN. The scatterplots reveal a similar pattern of errors to that found in Experiment 1 and this result is reflected in the evaluation statistics. MLIN, in comparison to the BPNN, shows poorer performance on each of the seven evaluation metrics (Table 3). The two split sample datasets were observed to produce similar outputs in each case which is indicative of unbiased solutions and minimal potential overfitting.

5.3 Experiment 3

SSE for the trained model was higher, in comparison to the two previous solutions, levelling off at a liitle over 3.76 sdu. BPNN (Figs. 5a, b) outputs still exhibit a clear trend, but the output values form a scattered cloud, having a spread of values situated both above and below the line of perfect agreement. The spread of error is least at the upper and lower ends of the range and greatest in the central regions. Nevertheless, the omission of one principal component and the conflation of two others meant that there was even less information in the input variables upon which to develop a model. However, broad levels of non-linear approximation can still be achieved. Figures 6a, b show the corresponding results for MLIN. The scatterplots in this instance reveal a similar pattern of errors to that found in Experiments 1–2 and this result is reflected in the evaluation statistics. MLIN, in comparison to the BPNN, shows poorer performance on each of the seven evaluation metrics (Table 3). The two split sample datasets were observed to produce similar outputs in each case which is indicative of unbiased solutions and minimal potential overfitting.

5.4 Experiment 4

SSE for the trained model levelled off at a little under 8.05 standardised ratio units (sru). Error is much higher in this experiment, compared to that found in the earlier modelling scenarios, which is indicative of a more challenging "modelling situation". BPNN outputs are shown in Figs. 7a, b. The scatterplots reveal a reasonable level of agreement between the original computations and the neural outputs in the upper section of the graphs, but there is a broad spread of predictions in the lower sections, and a clear cut-off point below which the neural solution does not produce output predictions. This particular situation represents a different form of generalisation that occurs under conditions of insufficient information, or detrimental contradictions, in the dataset. It can be equated to "pit-filling" in the lower regions of the target output. Indeed, in different regions of the solution space, a crude generalisation is the best that can be done under such circumstances which serves to confirm that when there is a firm relationship to be modelled, the neural solution will extract it, whereas in other cases it will attempt to fit a broad higher level approximation. Figures 8a, b show the corresponding results for MLIN and depict the failure of this method to capture important non-lineararites. The scatterplots in this instance reveal a different pattern of errors to that found in Experiments 1–3. The output in both plots is "pivoted" around a central point on the "line of perfect agreement" and the distribution of values spreads out from this point in both directions such that the maximum spread of values occurs in the uppermost and lowermost regions. Higher level values are underpredicted. Lower level values straddle the line of perfect agreement. The differences between such findings and the previous results can be related to the fact that the non-linear nature of the modelling exercise has changed. This result is reflected in the evaluation statistics. MLIN, in comparison to the BPNN, shows poorer performance on six of the seven evaluation metrics

(Table 3). MSRE (Mean Squared Relative Error) records as a ratio the level of overall agreement between the observed and modelled datasets and on this particular occasion depicts superior scores. However, a low score on this metric does not necessarily indicate a good model in terms of accurate forecasts, since positive and negative errors will tend to cancel each other out. The nature of the reported error as shown on the scatterplot confirms that such oddities can be explained in terms of numbers related to high levels of potential "cancelling out". The two split sample datasets were observed to produce similar outputs in each case which is indicative of unbiased solutions and minimal potential overfitting.

Figures 10–11 reveal the complex nature of the target solution surface. Figure 9 provided information about the unequal distribution of the output points in input space from which the two interpolated surfaces were constructed. Figure 10 identifies various conflicts and inconsistencies in the calculated estimations that provided target outputs for the lower levels of *P/Wm* and *Wo/Wm*. *R/P*, in this particular region of the initial computations, contained a "speckled" pattern such that the neural solution was required to implement a certain degree of "simplification" or "averaging". The derived surface nevertheless exhibits a recognisable trend that extends across the diagram from bottom-left to top-right. This trend becomes more well defined when it reaches mid to higher values of either *P/Wm* or *Wo/Wm*. Here the influence of soil water is less significant, producing a clear-cut, diagonal switchover, that runs more or less from top-left to bottom-right on the diagram i.e. opposite direction to the main trend that increases from zero-zero in the bottom-left corner to one-one in the top-right corner. This main trend is also curved and appears to have a pivotal point in the bottom-right corner, with steep changes in surface, contrasting with the outer boundaries that have shallower increases. The overall appearance resembles a traditional "hand fan" and the inherent nature of the non-linear internal relationship that has been modelled in this experiment is revealed. Figure 11 confirms the non-linear nature of the neural network solution that has been developed for Experiment 4. The construction process is observed to have (1) produced a "hand fan" model; (2) maintained important upper regions relationships; and (3) resolved potential conflicts in the lower regions of the output surface. Further analysis of "surface residuals" was not undertaken but might be instructive in terms of subsequent hydrological modelling experiments.

6 Discussion

Two earlier published papers considered the nature of the relationship that existed between NN and WLTF models developed on hydrological datasets. The four controlled experiments that are reported in this paper permit the earlier findings to be viewed in context. It is important to consider the previous interpretations and conclusions in terms of the theoretical justifications that underpinned each investigation. The earlier reported studies would appear to have been based on the following assumptions:

1. The rainfall-runoff relationship is a recognised non-linear catchment process;
2. Measurements of observed rainfall and runoff can be used to develop a non-linear catchment response dataset;
3. The measured datasets and the manner in which such datasets are used has encapsulated the non-linear catchment response in a suitable format for subsequent identification and extraction using machine learning algorithms.

This list does not however contain an explicit method of testing for the presence of non-linear relationships in each dataset. The simplest test would be to develop a linear model on the selected dataset, as a measure of the extent to which a linear or near-linear relationship exists, and thereafter select the most appropriate tool for subsequent modelling operations.

The exact manner in which a specific problem has been formulated is also important. The two earlier papers did not consider the extent to which the required solution called for the production of a simple model, with limited non-linear modelling capabilities, and no real need for the incorporation of complex dynamics. The solutions incorporated the last observed record at their point of forecast and as such the modelling operation might simply amount to calculating the change in discharge, which in most cases will be a near-linear operation, as opposed to something more physical. Further particulars on the dominant effect of including the last observed record at the point of forecast and reported findings related to countering or suppressing such factors using a constraint-based method can be found in Abrahart et al. (2007)[1]. It is also axiomatic in such cases that the use of smaller forecasting horizons will lead towards the development of linear or near-linear solutions. Han et al. (2007) reported modelling experiments performed over different horizons which provides further insight into this question. Indeed, as the forecasting horizon was increased, the dominant effect of the last recorded input was reduced and the requirement for producing a non-linear solution becomes more apparent. If a marked non-linear relationship exists it should be quite obvious that a non-linear solution will be required to model it – as demonstrated in this paper. However, if a theoretical non-linear relationship appears to have been captured in an acceptable manner using a linear or near-linear model, then the exact reason(s) for this unexpected result should be questioned. It is, moreover, insufficient to assert that such findings can be attributed to potential failings, either in a specific dataset, or to the tool that has been applied to model it, without first having undertaken detailed investigations that are able to confirm or disprove such matters in the manner of "hypothesis testing".

The controlled experiments that are reported in this paper have established that a NN will produce an appropriate non-linear solution if presented with an appropriate non-linear situation to model. If the problem is linear, or near-linear, it is axiomatic that WLTF models and trained NN models will produce a similar set of results; the tools are performing as expected. The anticipated "superior performance" of a NN in relation to a WLTF will, as a result, appear to be limited. It is also possible that one might obtain better generalisation using a simple linear model than a NN in the case

[1]Abrahart, R. J., Heppenstall, A. J., and See, L. M.: Timing error correction procedures applied to neural network rainfall-runoff modelling, Hydrol. Sci. J., in review, 2007.

HESSD

4, 287–326, 2007

R. J. Abrahart and L. M. See

of a function that contains mild non-linearities if the datasets are too small or contain too much noise, which will prevent the NN from accurately estimating the non-linearities (Sarle, 1997). The earlier studies can therefore be interpreted in a different light. Linear models should be used as benchmarks against which NNs are tested to indicate the degree to which the presented relationship that needs to be modelled is linear and therefore requires the application of a linear modelling solution. Moreover, if required, following the adoption of a linear modelling solution is it also the case that non-linear tools could thereafter be used for the identification of neglected non-linearities (Curry and Morgan, 2003). The question of which tool would be more appropriate in a linear or near-linear modelling situation is not a matter for scientific contest; it is a practical issue that equates to picking the "right tool for the right job".

Hydrological modelling requires consistent measures of merit and trust. Hillel (1986: p. 42) advocated that hydrological modelling solutions should be: "parsimonious" – each model should contain a minimum number of parameters that can be measured in the field; "modest" – the scope and purpose to which a specific model can be applied must not be overstated; "accurate" – the correctness of the forecast or prediction need not be better than the correctness of the input measurements; and "testable" – the limits within which the model outputs are valid can be defined. This paper and its predecessors have focused on one aspect of merit and trust: the production of more accurate outputs. However, other qualities and issues are also important with respect to practical operational implementations, and mechanistic properties such as "robustness" and "graceful degradation" will not in all cases have an optimal relationship with model output accuracies and so must be treated as independent properties that impart a set of constraints. To provide a robust solution each model must exhibit a constant or stable behaviour and be insensitive to potential uncertainties in the construction and parameterisation process e.g. problems related to measurements that cannot be obtained with sufficient accuracies or are not constant over long(er) periods. To be reliable and trusted an operational model must also exhibit the properties of "graceful degradation"; a gradual and progressive reduction in overall performance such that the model continues to operate and function in a normal manner, but provides a reduced level of service, as opposed to taking incorrect actions or suffering a total collapse of processing activities. Environmental modelling investigations into the changing nature of neural network outputs related to the provision of degraded inputs are reported for hydrological forecasting in Abrahart et al. (2001) and for sediment transfer in Abrahart and White (2000). For more detailed discussion on the requirements constraint issue the interested reader is referred to Alippi (2002).

7 Conclusions

Two important issues have been raised: a) to what extent can a neural network model perform non-linear hydrological modelling operations given a suitable problem and a set of information-rich unproblematic observations, i.e. to help overcome issues related to poor content and noise; and b) to what extent are most reported neural network hydrological modelling investigations and proposed solutions near-linear as opposed to non-linear applications. This paper has shed some light on the first question. It is possible to infer from the success of the reported modelling experiments that neural network hydrological modelling solutions can be used to perform reliable non-linear transformations and to produce different levels of hydrological process generalisation.

Minimum effort was placed on design and construction of the NN. The end product, nevertheless, was considered to be acceptable in all cases and not necessarily optimal. For operational purposes such solutions might be sufficient. The act of building a neural network emulator was also discovered to be a rather robust operation that required limited expert involvement. The networks were quick to create and simple to test which makes them ideal tools for bootstrapping, sensitivity analysis and rapid prototyping implementations. Further emulation exercises are to be encouraged.

The power of a neural network to perform simplification operations and to explore alternative internal relationships has been demonstrated. This type of exploration offers numerous interesting possibilities. It could for example be used for the assessment of more complex models, or of their internal components, in terms of visualizing difficult-to-observe process-based relationships. The power to omit one or more problematic variables under certain conditions is of particular importance in the case of scarce or difficult to obtain datasets, and, in addition, has clear cost-benefit implications.

References

Abrahart, R. J.: Neural networks and the problem of accumulated error: an embedded solution that offers new opportunities for modelling and testing, in: Hydroinformatics'98: Proceedings Third International Conference on Hydroinformatics, Copenhagen, Denmark, 24–26 August 1998, edited by: Babovic, V. and Larsen, C. L., A.A. Balkema Publishers, Rotterdam, The Netherlands, 2, 725–731, 1998.

Abrahart, R. J., Kneale, P. E., and See, L. M. (Eds.): Neural Networks for Hydrological Modelling, A.A.Balkema Publishers, Rotterdam, The Netherlands, 2004.

Abrahart, R. J., See, L., and Kneale, P. E.: Investigating the role of saliency analysis with a neural network rainfall-runoff model, Computers and Geosciences, 27(8), 921–928, 2001.

Abrahart, R. J. and White, S.: Modelling Sediment Transfer in Malawi: Comparing Backpropagation Neural Network Solutions Against a Multiple Linear Regression Benchmark Using Small Data Sets, Physics and Chemistry of the Earth (B), 26(1), 19–24, 2001.

Alippi, C.: Selecting Accurate, Robust, and Minimal Feedforward Neural Networks, IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications, 49(12), 1799–1810, 2002.

Adams, R., Dunn, S. M., Lunn, R., Mackay, R., and O'Callaghan, J. R.: Assessing the Performance of the NELUP Hydrological Models for River Basin Planning, J. Environ. Planning Manage., 38(1), 53–76, 1995.

ASCE: Artificial Neural Networks in Hydrology. I. Preliminary Concepts, J. Hydrol. Eng., 5(2), 115–123, 2000a.

ASCE: Artificial Neural Networks in Hydrology. II. Hydrologic Applications, J. Hydrol. Eng., 5(2), 124–137, 2000b.

Bhattacharya, B., Price, R. K., and Solomatine, D. P.: Data-driven modelling in the context of sediment transport, Phys. Chem. Earth, 30(4–5), 297–302, 2005.

Birikundavy, S., Labib, R., Trung, H. T., and Rousselle, J.: Performance of Neural Networks in Daily Streamflow Forecasting, J. Hydrol. Eng., 7(5), 392–398, 2002.
Campolo, M., Soldati, A., and Andreussi, P.: Artificial neural network approach to flood forecasting in the River Arno, Hydrol. Sci. J., 48(3), 381–398, 2003.
Chen, X., Chen, Y. D., and Xu, C.: A distributed monthly hydrological model for integrating spatial variations of basin topography and rainfall, Hydrol. Processes, 21(2), 242–252, 2007.
Cheng, C. T., Ou, C. P., and Chau, K. W.: Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration, J. Hydrol., 268(1–4), 72–86, 2002.
Cheng, C.-T., Zhao, M.-Y., Chau, K. W., and Wu, X.-Y.: Using genetic algorithm and TOPSIS for Xinanjiang model calibration with a single procedure, J. Hydrol., 316(1–4), 129–140, 2006.
Curry, B. and Morgan, P. H.: Neural networks, linear functions and neglected non-linearity, Computat. Manage. Sci., 1, 15–29, 2003.
Dawson, C. W., Abrahart, R. J., and See, L. M.: HydroTest: a web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts, Environ. Modelling & Software, 22(7), 1034–1052, 2007.
Dawson, C. W. and Wilby, R. L.: Hydrological modelling using artificial neural networks, Progress in Physical Geography, 25(1), 80–108, 2001.
Dümenil, L. and Todini, E.: A rainfall-runoff scheme for use in the Hamburg climate model, in: Advances in Theoretical Hydrology – A tribute to James Dooge, edited by: O'Kane, J. P., Elsevier, Amsterdam, The Netherlands, 129–157, 1992.
Gaume, E. and Gosset, R.: Over-parameterisation, a major obstacle to the use of artificial neural networks in hydrology?, Hydrol. Earth Syst. Sci., 7(5), 693–706, 2003.
Govindaraju, R. S. and Rao, A. R. (Eds.): Artificial Neural Networks in Hydrology, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.
Han, D., Kwong, T., and Li, S.: Uncertainties in real-time flood forecasting with neural networks, Hydrol. Processes, 21(2), 223–228, 2007.
Hettiarachchi, P., Hall, M. J., and Minns, A. W.: The extrapolation of artificial neural networks for the modelling of rainfall-runoff relationships, J. Hydroinformatics, 7(4), 291–296, 2005.
Hillel, D.: Modeling in soil physics: A critical review, in: Future Developments in Soil Science Research, Soil Society of America, Madison, Wisconsin, USA, 35–42, 1986.
Hsu, C. Y., Lee, H. Y., and Lin, Y. L.: Estimation of sediment yields in a watershed area using ANN method, in: River Basin Management II (Transactions of the Wessex Institute, Ecology

and the Environment Vol. 60, Progress in Water Resources Vol. 7), edited by: Brebbia, C. A., WIT Press, Southampton, UK, 461–470, 2003.

Huang, W., Xu, B., and Chad-Hilton, A.: Forecasting flows in Apalachicola River using neural networks, Hydrol. Processes, 18(13), 2545–2564, 2004.

Jayawardena, A. W. and Zhou, M. C.: A modified spatial soil moisture storage capacity distribution curve for the Xinanjiang Model, J. Hydrol., 227(1–4), 93–113, 2000.

Kisi, O.: Multi-layer perceptrons with Levenberg-Marquardt training algorithm for suspended sediment concentration prediction and estimation, Hydrol. Sci. J., 49(6), 1025–1040, 2004.

Kisi, O.: Suspended sediment estimation using neuro-fuzzy and neural network approaches, Hydrol. Sci. J., 50(4), 683–696, 2005.

Lin, C. A., Wen, L., Lu, G., Wu, Z., Zhang, J., Yang, Y., Zhu, Y., and Tong, L.: Atmospheric-hydrological modeling of severe precipitation and floods in the Huaihe River Basin, China, J. Hydrol., 330(1–2), 249–259, 2006.

Liong, S. Y. and Chan, W. T.: Runoff Volume Estimates with Neural Networks, in: Neural Networks and Combinatorial Optimization in Civil and Structural Engineering, Proceedings Third International Conference on the Application of Artificial Intelligence to Civil and Structural Engineering, Edinburgh, UK, 17-19 August 1993, edited by: Topping, B. H. V. and Khan, A. I., Civil-Comp Press, Stirling, UK, 67–70, 1993.

Maier, H. R. and Dandy, G. C.: Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, Environ. Modelling & Software, 15(1), 101–124, 2000.

Nagy, H. M., Watanabe, K., and Hirano, M.: Prediction of Sediment Load Concentration in Rivers using Artificial Neural Network Model, J. Hydraulic Eng., 128(6), 588–595, 2002.

Riad, S., Mania, J., Bouchaou, L., and Najjar, Y.: Predicting catchment flow in a semi-arid region via an artificial neural network technique, Hydrol. Processes, 18(13), 2387–2393, 2004.

Rogers, L. L. and Dowla, F. U.: Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling, Water Resour. Res., 30(2), 457–481, 1994.

Sarle, W. S. (Ed.): Neural Network FAQ, Part 3 of 7: Generalisation (periodic posting to the Usenet newsgroup comp.ai.neural-nets), ftp://ftp.sas.com/pub/neural/FAQ.html, 1997, last accessed: 8 November 2006.

Selmic, R. R. and Lewis, F. L.: Neural-network approximation of piecewise continuous functions: application to friction compensation, IEEE Transactions on Neural Networks, 13(3), 745–751, 2002.

Senthil Kumar, A. R., Sudheer, K. P., Jain, S. K., and Agarwal, P. K.: Rainfall-runoff modelling using artificial neural networks: comparison of network types, Hydrol. Processes, 19(6), 1277–1291, 2005.

Solomatine, D. P. and Avila Torres, L. A.: Neural network approximation of a hydrodynamics model in optimizing reservoir operation, in: Hydroinformatics 96: Proceedings Second International Conference on Hydroinformatics, Zurich, Switzerland, 9–13 September 1996, edited by: Müller, A., A.A. Balkema Publishers, Rotterdam, The Netherlands, 1, 201–206, 1996.

Sontag, E. D.: Feedback Stabilization Using Two-Hidden-Layer Nets, IEEE Transactions on Neural Networks, 3(6), 981–990, 1992.

Su, B., Kazama, S., Lu, M., and Sawamoto, M.: Development of a distributed hydrological model and its application to soil erosion simulation in a forested catchment during storm period, Hydrol. Processes, 17(14), 2811–2823, 2003.

Todini, E.: The ARNO rainfall-runoff model, J. Hydrol., 175(1–4), 339–382, 1995.

Toth, E., Brath, A., and Montanari, A.: Comparison of short-term rainfall prediction models for real-time flood forecasting, J. Hydrol., 239(1–4), 132–147, 2000.

Van den Boogaard, H. F. P. and Kruisbrink, A. C. H.: Hybrid modelling by integrating neural networks with numerical models, in: Hydroinformatics 96: Proceedings Second International Conference on Hydroinformatics, Zurich, Switzerland, 9–13 September 1996, edited by: Müller, A., A.A. Balkema Publishers, Rotterdam, The Netherlands, 2, 471–478, 1996.

Varoonchotikul, P.: Flood forecasting using artificial neural networks, A.A. Balkema Publishers, Rotterdam, The Netherlands, 2003.

Weigend, A. S., Huberman, B. A., and Rumelhart, D. E.: Predicting Sunspots and Exchange Rates with Connectionist Networks, in: Nonlinear Modeling and Forecasting, edited by: Casdagli, M. and Eubank, S., Addison-Wesley, Redwood City, California, USA, 395–432, 1992.

Wilby, R. L., Abrahart, R. J., and Dawson, C. W.: Detection of conceptual model rainfall-runoff processes inside an artificial neural network, Hydrol. Sci. J., 48(2), 163–181, 2003.

Yitian, L. and Gu, R. R.: Modeling Flow and Sediment Transport in a River System Using an Artificial Neural Network, Environmental Management, 31(1), 122–134, 2003.

Zell, A., Mamier, G., Vogt, M., Mache, N., Huebner, R., Doring, S., Herrmann, K.-U., Soyez, T., Schmalzl, M., Sommer, T., Hatzigeorgiou, A., Posselt, D., Schreiner, T., Kett, B., Clemente, G., and Wieland. J.: SNNS, Stuttgart Neural Network Simulator, User Manual, Version 4.1, University of Stuttgart, Institute for Parallel and Distributed High Performance Systems, Report No. 6/95, 1995.

Zhao, R.-J.: The Xinanjiang model applied in China, J. Hydrol., 135(1–4), 371–381, 1992.
Zhao, R.-J. and Liu, X.-R.: The Xinanjiang Model, in: Computer Models of Watershed Hydrology, edited by: Singh, V. P., Water Resources Publications, Fort Collins, USA, 215–232, 1995.
Zhao, R.-J., Zhang, Y.-L., Fang, L.-R., Liu, X.-R., and Zhang, Q.-S.: The Xinanjiang model, in: Hydrological Forecasting: Proceedings of the Symposium on the Application of Recent Developments in Hydrological Forecasting to the Operation of Water Resource Systems, Oxford, April 1980, International Association of Hydrological Sciences Press, Wallingford, UK, IAHS Publication No. 129, 351–356, 1980.

Table 1. BPNN training programme and last reported error for the four reported experiments.

Epochs/ training cycles	Learning rate	Exp. 1 SSE(sdu)	Exp. 2 SSE(sdu)	Exp. 3 SSE(sdu)	Exp. 4 SSE(sru)
0–2000	0.8	0.0655	1.1916	3.8582	8.4872
2001–4000	0.6	0.0484	1.1633	3.8167	8.2796
4001–6000	0.4	0.0410	1.1503	3.7878	8.1569
6001–8000	0.2	0.0376	1.1382	3.7623	8.0489

HESSD
4, 287–326, 2007

Table 2. MLIN parameter coefficients for the four reported experiments.

	Exp. 1	Exp. 2	Exp. 3	Exp. 4
Intercept	−0.1535	−0.1290	−0.2249	−0.1040
Precipitation [*P*]	0.6286	0.6294	0.6283	–
Initial Water [*Wo*]	0.6817	0.6771	–	–
Maximum Water [*Wm*].	−0.4133	−0.4093	–	–
b [curve fitting exponent]	0.0500	–	–	–
Soil Water Ratio [*Wo/Wm*].	–	–	0.3518	0.7021
Input-Storage Ratio [*P/Wm*].	–	–	–	0.5147

HESSD
4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and
L. M. See

EGU

Table 3. HydroTest evaluation statistics for the four reported experiments.

		Exp.	Equal treatment		Favour high flows		Comparison with norms		
			MAE	MARE	RMSE	MSRE	CofE	RSqr	IoAd
BPNN	Training	1	0.0030	0.1619	0.0039	3.3415	0.9997	0.9997	0.9999
		2	0.0140	0.2728	0.0213	3.0822	0.9909	0.9910	0.9977
		3	0.0269	0.3163	0.0388	3.3362	0.9700	0.9702	0.9923
		4	0.0405	0.2658	0.0569	2.7176	0.9502	0.9511	0.9872
	Testing	1	0.0029	0.1433	0.0040	7.0474	0.9997	0.9997	0.9999
		2	0.0141	0.2692	0.0217	7.0690	0.9899	0.9900	0.9975
		3	0.0264	0.3072	0.0383	6.4309	0.9685	0.9688	0.9918
		4	0.0407	0.2617	0.0569	0.4809	0.9503	0.9511	0.9872
MLIN	Training	1	0.0662	3.8984	0.0808	2,284.33	0.8697	0.8697	0.9640
		2	0.0650	3.9764	0.0849	2,796.14	0.8562	0.8636	0.9596
		3	0.0676	3.7823	0.0833	1,728.56	0.8615	0.8615	0.9615
		4	0.0725	0.3445	0.0963	1.4153	0.8572	0.8572	0.9603
	Testing	1	0.0659	3.9541	0.0804	3,155.88	0.8612	0.8625	0.9628
		2	0.0651	4.1488	0.0841	3,767.03	0.8482	0.8572	0.9587
		3	0.0666	4.7857	0.0821	9,807.15	0.8554	0.8562	0.9608
		4	0.0724	0.3356	0.0957	0.3583	0.8593	0.8594	0.9605

HESSD

4, 287–326, 2007

Title Page
Abstract Introduction
Conclusions References
Tables Figures
◄ ►
Back Close
Full Screen / Esc
Printer-friendly Version
Interactive Discussion

EGU

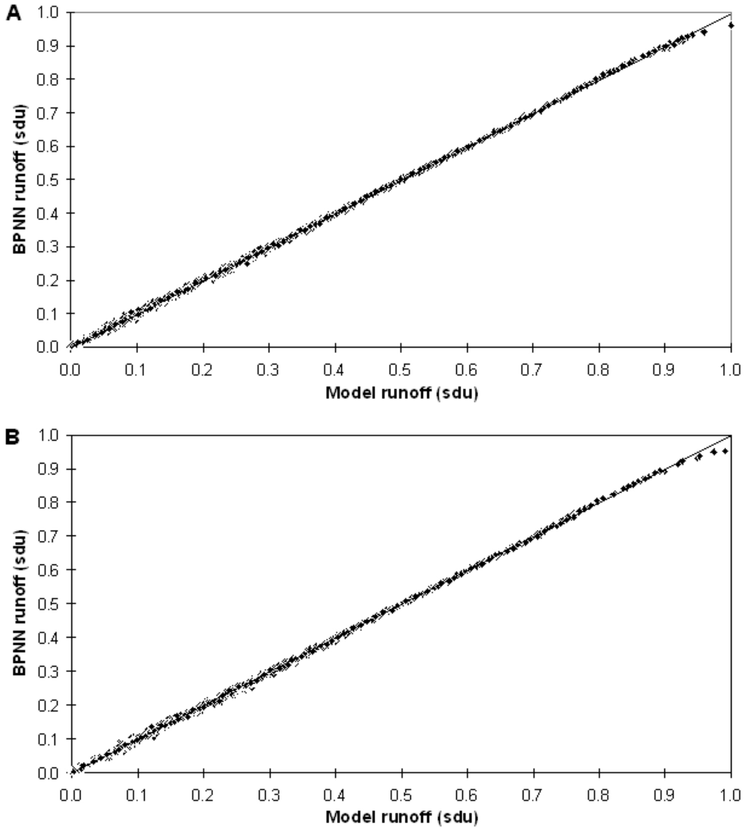

Fig. 1. BPNN training output **(A)** and testing output **(B)** scatterplots for Experiment 1.

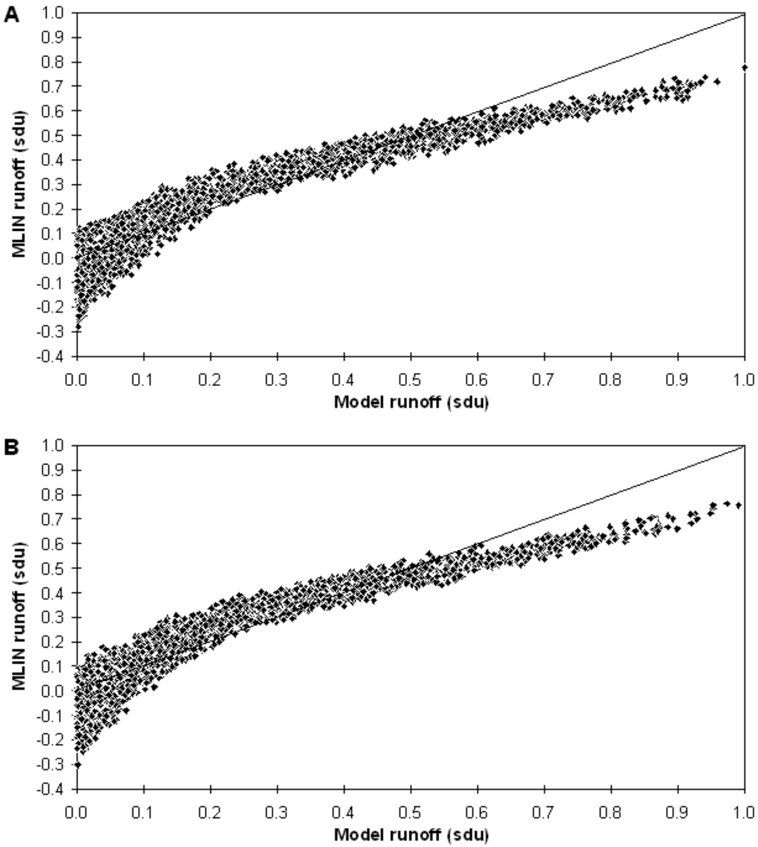

Fig. 2. MLIN training output **(A)** and testing output **(B)** scatterplots for Experiment 1.

HESSD

4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

EGU

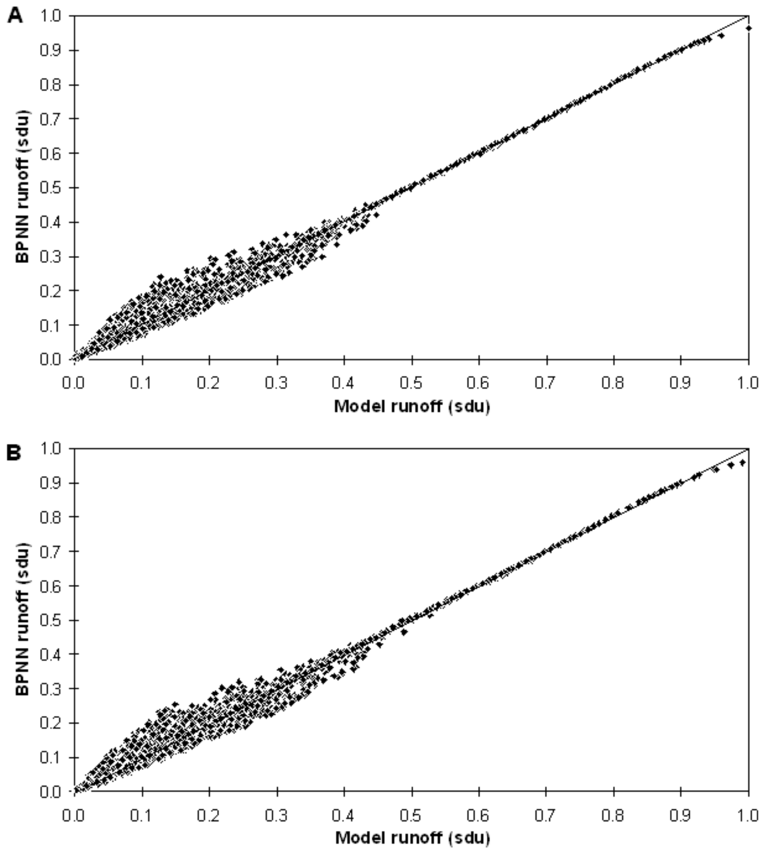

Fig. 3. BPNN training output **(A)** and testing output **(B)** scatterplots for Experiment 2.

HESSD
4, 287–326, 2007

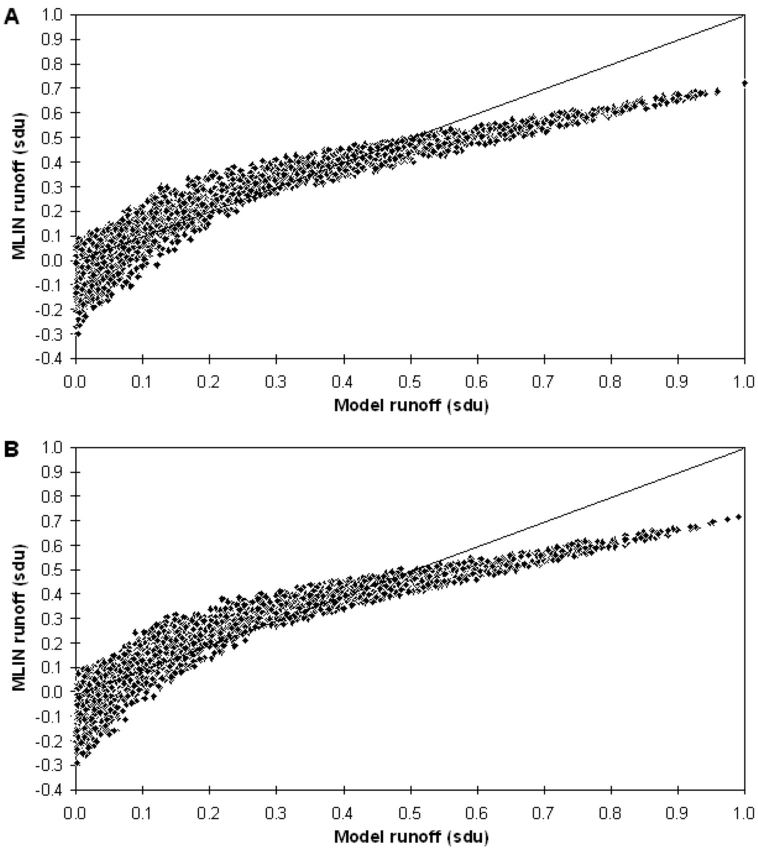

Fig. 4. MLIN training output **(A)** and testing output **(B)** scatterplots for Experiment 2.

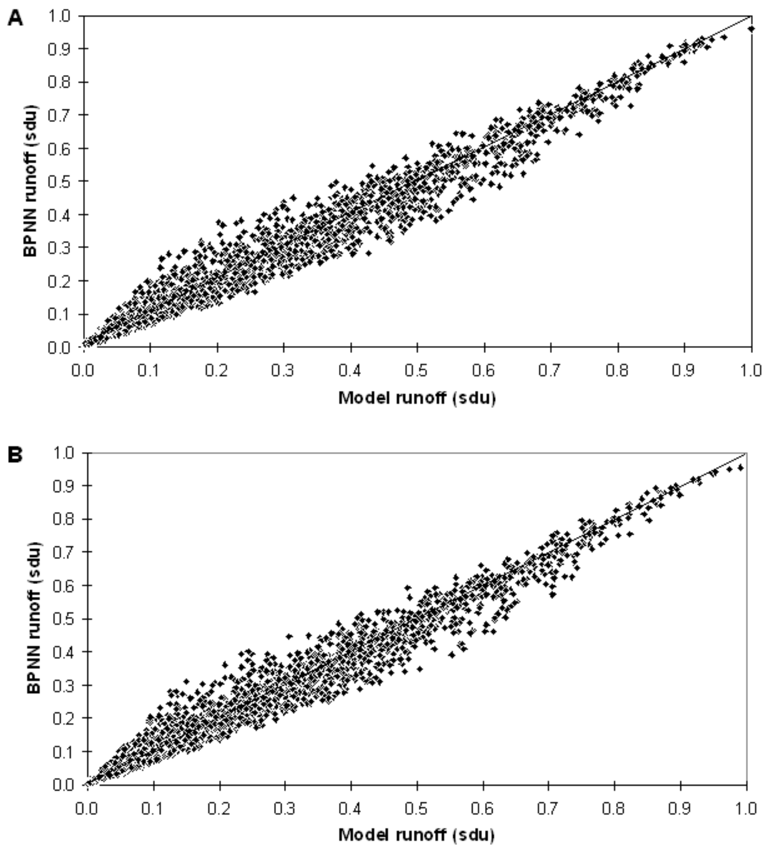

Fig. 5. BPNN training output **(A)** and testing output **(B)** scatterplots for Experiment 3.

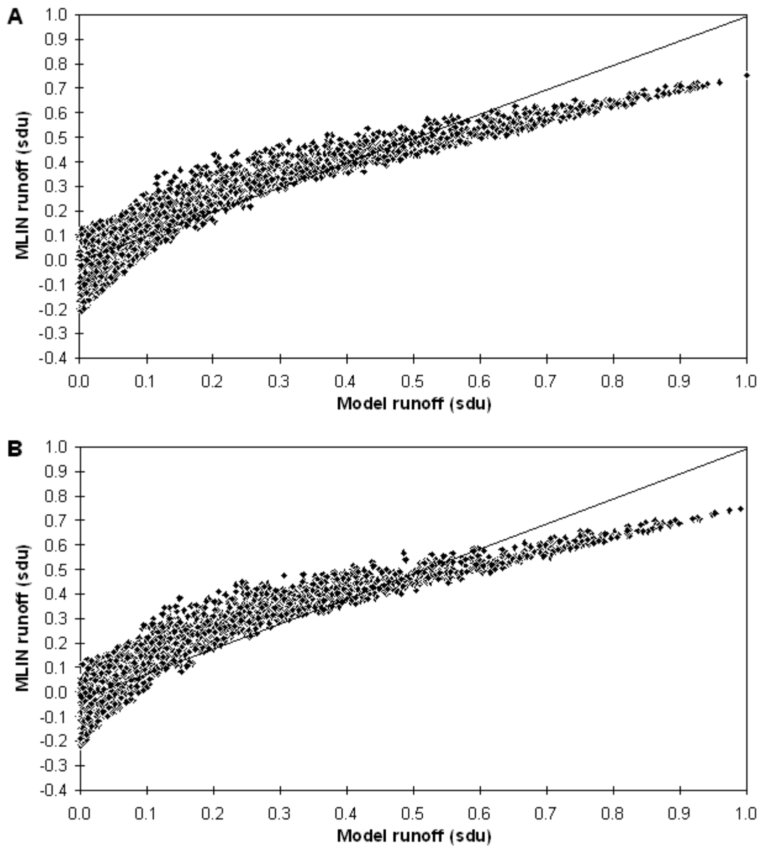

Fig. 6. MLIN training output **(A)** and testing output **(B)** scatterplots for Experiment 3.

HESSD
4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

EGU

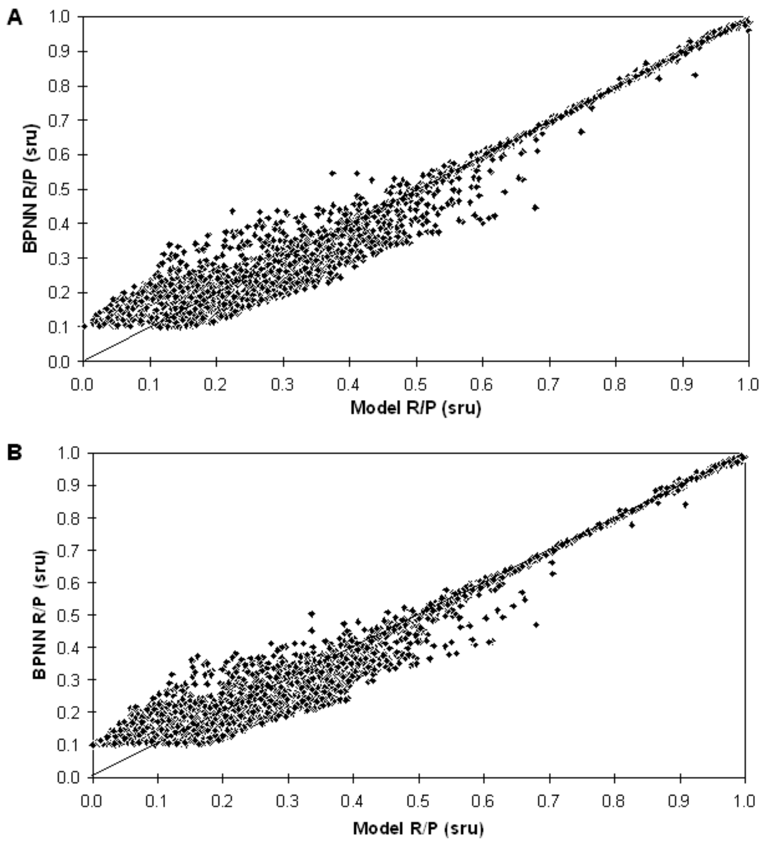

Fig. 7. BPNN training output **(A)** and testing output **(B)** scatterplots for Experiment 4.

HESSD

4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

EGU

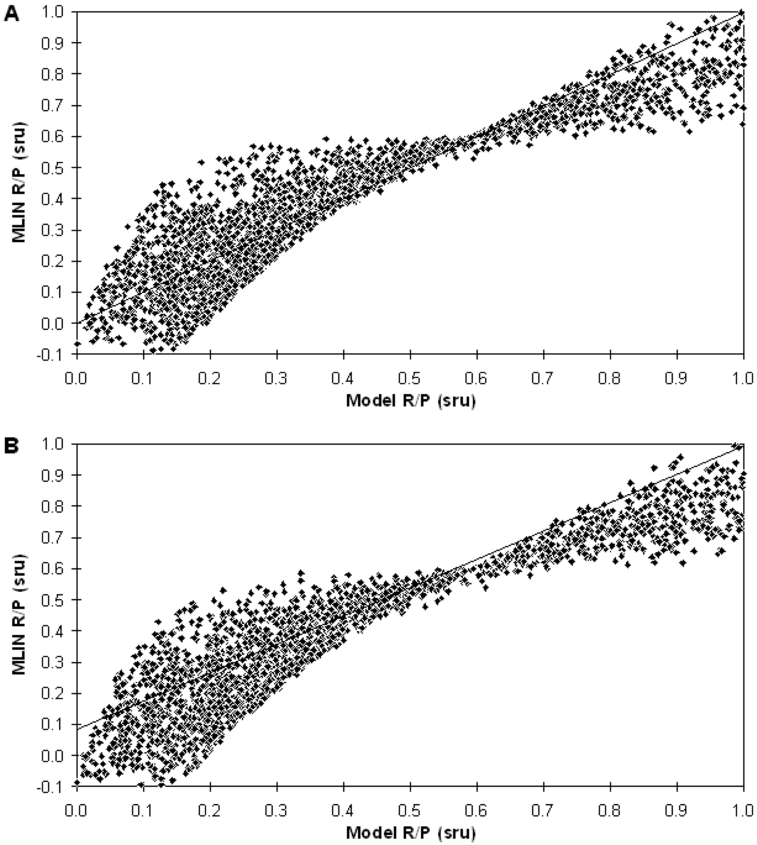

Fig. 8. MLIN training output **(A)** and testing output **(B)** scatterplots for Experiment 4.

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

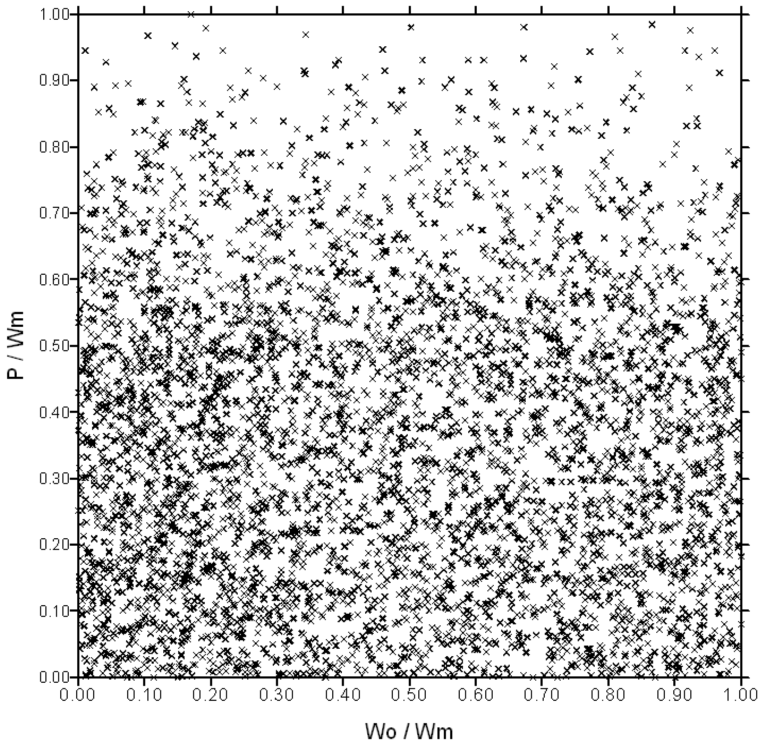

Fig. 9. Scatter of points in input space for Experiment 4.

HESSD
4, 287–326, 2007

R. J. Abrahart and L. M. See

Title Page
Abstract Introduction
Conclusions References
Tables Figures
◄ ►
Back Close
Full Screen / Esc
Printer-friendly Version
Interactive Discussion

EGU

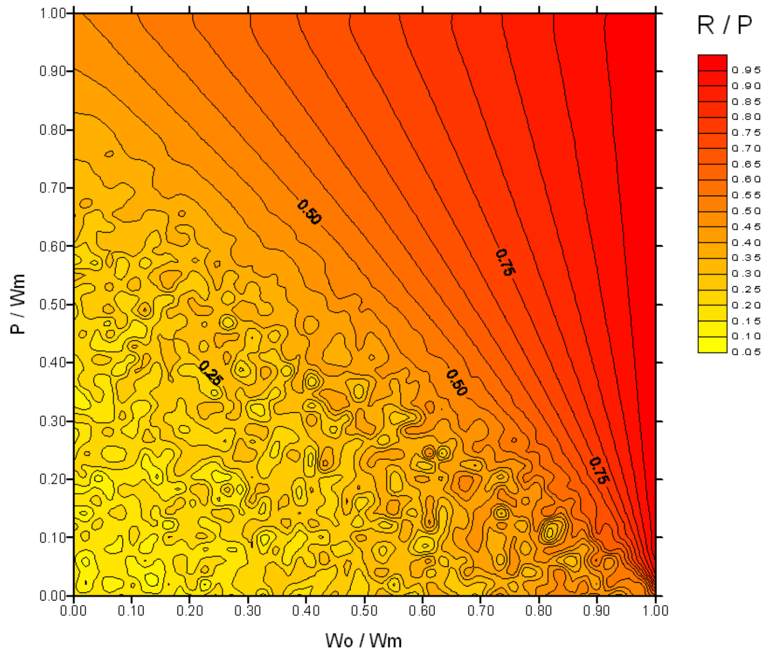

Fig. 10. Target surface interpolated from equation outputs for Experiment 4.

HESSD
4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

EGU

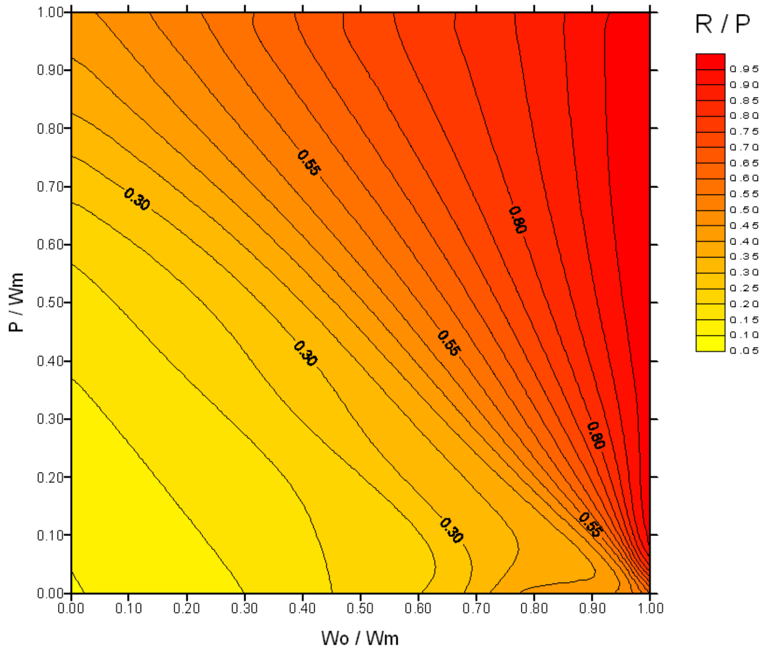

Fig. 11. Target surface interpolated from neural network outputs for Experiment 4.

HESSD

4, 287–326, 2007

Neural network emulation of a rainfall-runoff model

R. J. Abrahart and L. M. See

EGU