Development of Stochastic Artificial Neural Networks for Hydrological Prediction

G. B. Kingston, M. F. Lambert and H. R. Maier

Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, University of Adelaide, Adelaide, SA, 5005, Australia. Email: gkingsto@civeng.adelaide.edu.au

Abstract: Many studies have used artificial neural networks (ANNs) for the prediction and forecasting of hydrological variables, including runoff, precipitation and river level, which are subsequently used for design or management purposes. However, although it is widely recognised that hydrological models are subject to parameter uncertainty, ANNs in this field have been almost exclusively deterministic with little attention paid to the uncertainty in the network weights. The inherent variability of hydrological processes means that no finite set of observations will give exact parameter values and therefore it is important to express network weights as a range of plausible values such that one, possibly incorrect, weight vector does not completely dominate the predictions. In this paper a synthetically generated data set is used as a tool for demonstrating the potential advantages of explicitly accounting for parameter uncertainty. A Markov chain Monte Carlo approach is used to sample from the distribution of possible network weights in an attempt to eliminate or reduce the potential problems that can be encountered during network training. By expressing the network weights as a distribution it is also possible to express the level of confidence with which ANN predictions are made.

Keywords: *Hydrology; Artificial neural network; Parameter uncertainty; Markov chain Monte Carlo*

1. INTRODUCTION

Many studies have used artificial neural networks (ANNs) for the prediction and forecasting of hydrological variables, including runoff, precipitation and river level (ASCE, 2000), which are subsequently used for design or management purposes. It has been shown that, when applied correctly, ANNs are able to perform at least as well as more conventional modelling approaches. However, although it is widely recognised that hydrological models are subject to parameter uncertainty, ANNs in this field have been almost exclusively deterministic with little attention paid to the uncertainty in the network weights.

The connection weights of an ANN are adjustable and can be compared to coefficients in statistical models. The network is "trained" or calibrated by iteratively adjusting the connection weights such that a predetermined objective function is minimised and the best fit between the model predictions and the observed data is obtained. However, the task of training a network is not always straightforward and may be complicated by the existence of local minima in the solution surface or the potential of overfitting the training data.

Using standard neural network approaches, the aim is to find an "optimal" set of network weights. However, no finite set of observations can be expected to give exact model parameter values, as the inherent variability of the hydrological process itself means that each different set of data would yield different parameter values. Therefore, it is important to express network weights as a range of plausible values such that one, possibly incorrect, weight vector does not completely dominate the predictions.

In this paper Bayesian methods are employed in order to demonstrate the potential advantages of explicitly accounting for parameter uncertainty. In particular, Bayesian methods will be applied to determine a robust range of connection weights that may then be used to express the degree of confidence with which predictions are made.

2. METHODS

2.1. Determination of Robust Connection Weights

The nonlinear characteristics of ANNs lead to the existence of multiple optima on the solution surface and, consequently, many combinations of network weights may result in similar network

performance. There is currently no training algorithm that can guarantee that the network will converge on the global optimal solution as opposed to a local minimum in the solution surface.

Local or global optimisation algorithms may be used to train an ANN. Backpropagation, a first order local method, is currently the most widely used algorithm for optimising feedforward ANNs (Maier and Dandy, 2000). This algorithm is based on the method of steepest descent, where the network weights are updated according to:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \gamma_n \mathbf{d}_n \quad (1)$$

where $\mathbf{w}$ is the vector of connection weights, γ is the step size and $\mathbf{d}$ is a vector defining the direction of descent. This algorithm is an effective way of optimising weights, however, like all local search methods, it is susceptible to becoming trapped in local minima in the error surface. Global methods have the ability to escape local minima, as they employ random search techniques to allow the simultaneous search for an optimum solution in several directions. They are often more computationally intensive than local search techniques, but with improving computer technologies, the use of global optimisation methods is increasing. Duan et al. (1992) developed the shuffled complex evolution (SCE) algorithm that uses multiple "simplexes", started from random locations in the search space, to direct the search towards the global optimum. At periodic stages of the search, the points in the simplexes are shuffled together to ensure that information is shared and that each simplex is not conducting an independent search of the global optimum.

It is not sufficient for an ANN to simply fit the training data, however, as the purpose of ANNs is to generalise, i.e. to provide good predictions when presented with new data. When ANNs learn specific characteristics in the training data set that are not true in general the network has been "overtrained". Cross validation is a method that can be used to stop training before this occurs and ensures that only the general trends in the data are learnt. A test data set is employed to determine the optimal stopping time, which is when some objective function of the test set is a minimum (ASCE, 2000). However, to do this the available data must be split into two data sets, thus reducing the size of the training set and limiting the information that may be learnt during training, particularly if the original data set is not large.

Alternatively, the size of the network, and therefore the number of free parameters, may be reduced in an effort to prevent overtraining, as it has been suggested that overtraining does not occur if the number of samples in the training data set is at least 30 times the number of free parameters (Maier and Dandy, 2000). However, if the data set is of a limited size, it may not be possible to reduce the number of free parameters to achieve this ratio.

The methods currently employed to improve the generalisation ability and prediction performance of an ANN do not guarantee that the global solution of the network will be found. By explicitly accounting for parameter uncertainty it is acknowledged that it is difficult and often unlikely to find a single optimal weight vector. A more robust model can be developed if a range of plausible values is specified for each connection weight, rather than allowing one weight vector to completely dominate the predictions.

Bayesian Methods for Quantifying Uncertainty

Bayesian methodology offers an approach for handling uncertainty explicitly. Under this paradigm all uncertain quantities are expressed as probability distributions which represent the state of knowledge of the quantities. In Bayesian inference, any prior beliefs regarding an uncertain quantity are updated, based on new information, to yield a posterior probability of the unknown quantity.

Using Bayes' Theorem, the parameters of a model may be inferred from the data under the assumption that the model (structure) is "true" as follows:

$$P(\mathbf{w}|D,M) = \frac{P(D|\mathbf{w},M)P(\mathbf{w}|M)}{P(D|M)} \quad \text{or} \quad P(\mathbf{w}|D,M) \propto P(D|\mathbf{w},M)P(\mathbf{w}|M) \quad (2)$$

where $\mathbf{w}$ is a vector of model parameters, M is the model and D are the data. The likelihood, $P(D|\mathbf{w}, M)$, in this case comes from comparing the actual measurements to the model predictions and is the function through which the prior knowledge of $\mathbf{w}$ is updated by the data. The prior, $P(\mathbf{w}|M)$, supplies any knowledge regarding the model parameters such as information gained from previous measurements or general information such as their range and whether they are non-negative.

Stochastic Neural Networks

The application of Bayesian methodology to ANN training was pioneered by Neal (1992) and MacKay (1995). The calibration of a Bayesian or stochastic ANN involves sampling from the posterior distribution of network weights,

P(**w** | D, M) rather than finding a single "optimal" set of weights. As a result, a weight vector that fits the data only slightly better than others will contribute only slightly more to the prediction rather than completely dominating it.

If it is assumed that the noise model, which describes the residuals between model predictions and observations, is Gaussian, then the conditional probability of the observations given the input and weight vectors and network structure is as follows:

$$P(\mathbf{y}|\mathbf{x},\mathbf{w},M)=\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\left(\frac{y_i-f(\mathbf{x}_i,\mathbf{w})}{\sigma}\right)^2\right) \quad (3)$$

where σ is the scale of noise and n is the number of observations in the data set. This is the likelihood of the model parameters.

The Metropolis Algorithm

The high dimensionality of the conditional probabilities in (2) makes it difficult to calculate the posterior weight distribution numerically. Consequently methods have been introduced to approximate (2). Neal (1992) introduced a Markov chain Monte Carlo (MCMC) implementation to sample from the posterior weight distribution.

A common MCMC approach is to use the Metropolis algorithm, which makes use of a symmetrical proposal distribution (e.g. Gaussian) and an adaptive acceptance-rejection criterion to generate a random walk Markov chain which adapts to the true posterior distribution of an unknown variable, e.g. connection weights. Although the Metropolis algorithm is not the most efficient MCMC method, it is often employed because of its simplicity. Details of its computational implementation can be found in Thyer et al. (2002).

Given sufficient iterations, the Markov chain induced by the Metropolis algorithm should converge to a stationary distribution. From this point samples from the Metropolis algorithm can be considered to be samples from the posterior distribution. However, it is difficult to determine whether convergence has been achieved and how many iterations are required for convergence. Haario et al. (2001) introduced a variation of the Metropolis algorithm that was developed to provide improved convergence properties. In this algorithm the proposal distribution continually adapts to the posterior distribution by taking into account all previous states of the weight vector. Therefore a Markov chain is no longer produced. The adaptive Metropolis algorithm requires that the vector of network weights be first initialised with arbitrary starting values. Generally the weights which correspond to the maximum likelihood would be used for this purpose. The adaptive Metropolis algorithm was used in this study.

2.2. Quantifying Uncertainty in Predictions

If samples are taken from the posterior distribution of the network weights and new data are input into the network, a distribution of the network outputs will be produced. It is important to keep in mind, however, that connection weights of ANNs are not unique and can be highly correlated if too many hidden nodes are included in the network. It is therefore necessary to retain this correlation structure when the weights are sampled from their respective distributions.

Once the posterior distribution of the predictions is produced, confidence intervals may also be determined enabling predictions to be made with a known level of confidence. If the confidence bounds are tight, there is little uncertainty in the prediction and vice versa.

3. CASE STUDY

3.1. Data and Model Structure

Autoregressive (AR) models are commonly used to model hydrological time series data. The autoregressive model, AR(9), given by (4), was used to generate a set of synthetic time series data which were in turn used to demonstrate the importance of accounting for parameter uncertainty.

$$x_t = 0.3x_{t-1} - 0.6x_{t-4} - 0.5x_{t-9} + \varepsilon_t \quad (4)$$

In the above equation ε_t is a normally distributed random noise component with mean of 0 and standard deviation of 1. The use of synthetic data enables the capabilities of the proposed method to be investigated without the complication of other sources of uncertainty. By using this model the driving inputs and error model were known and as much data could be generated as required.

It is generally difficult to determine the appropriate number of hidden nodes that will allow adequate representation of the underlying function and the inclusion of unnecessary hidden nodes increases the uncertainty in the network weights, making the task of finding an optimal solution more complicated. As the relationship given by (4) is linear, the optimal network structure is one that contains no hidden layer nodes. However, in order to investigate the effects of parameter uncertainty when unnecessary nodes

are included, the network used in this study included one hidden node, as shown in Figure 1.

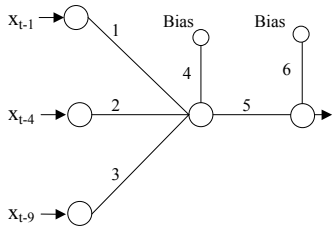

Figure 1. Network structure and parameter numbers

3.2. Determination of Robust Connection Weights

Two investigations were carried out in order to determine whether Bayesian methods could be applied to provide a more robust estimate of the connection weights than standard ANN optimisation procedures.

Local Minima

To determine the effectiveness of different training algorithms in finding the correct underlying relationship in the presence of local minima the following approaches were used and the results compared:

(a) Train the ANN with the backpropagation algorithm.

(b) Train the ANN with the SCE algorithm.

(c) Use the weights obtained from 1 to initialise the adaptive Metropolis algorithm and obtain a range of weight values.

A training data set of 300 data points was used to train approaches a, b and c. A test set of 200 data points was also used in approaches a and b for cross-validation.

Overtraining

A data set of 150 data points was used to investigate the ability of ANNs to find the correct underlying relationship given limited data. The following approaches were used and the results compared:

(d) Train the ANN on all 150 data points.

(e) Split the data into a training set of 100 data points and a test set of 50 data points and train the ANN, applying cross-validation.

(f) Use the weights obtained from d to initialise the adaptive Metropolis algorithm and obtain a range of weight values.

Given that the network in Figure 1 has 6 free parameters and there were only 150 data points in the data set (i.e. data points/free parameters < 30), it was assumed that the ANN would be overtrained using approach d. In each approach the SCE algorithm was used to train the network in an attempt to reduce the effects of becoming trapped in local minima.

3.3. Quantification of Prediction Uncertainty

Samples from the posterior distribution of each connection weight were given as the output from the adaptive Metropolis algorithm. 10,000 of these weight vectors were randomly sampled and the ANN was run for each weight vector selected. This resulted in a distribution of output values from which 95% confidence intervals were calculated.

To investigate the effect of retaining the correlation structure of the weight vectors, random samples were generated from each weight distribution, ignoring the correlation between weights. This was done by calculating the mean and standard deviation of each weight distribution and then generating samples from a normal distribution. The assumption that the weight distributions were Gaussian is a simplification, however, it was sufficient for the purposes of assessing the effects of ignoring the correlation structure. 95% confidence intervals were again calculated and compared to those determined when the correlation structure was retained.

4. RESULTS & DISCUSSION

The parameter numbers in the following results correspond to those displayed in Figure 1. Results are also given for a seventh parameter, which is the standard deviation of the model residuals (σ in (3)). Results will be presented in terms of the overall connection weights of network inputs, as it is considered that this measure provides a better indication of how well the underlying relationship has been estimated than an error measure such as the RMSE. Details regarding this measure can be found in Kingston et al. (2003).

4.1. Determination of Robust Connection Weights

The results of the investigations described in Section 3.2 are given in Table 1. The values in the shaded cells were calculated using the modes of the weight distributions determined by the adaptive Metropolis algorithm. The mode of the distribution is considered to give a good indication of the average weight value, as it is this value that would be used to make predictions with the greatest frequency. The error measure given in

Table 1 was calculated by:

$$\sqrt{\frac{\sum_{i=1}^{3}(C_i - A_i)^2}{3}} \qquad (5)$$

where C_i is the overall connection weight of input i and A_i is the actual weight for input i.

Table 1. Results of trained networks

		Approach used to Estimate Weights					
Input	Actual	a	b	c	d	e	f
x_{t-1}	0.3	0.31	0.33	0.29	0.40	0.40	0.36
x_{t-4}	-0.6	-0.75	-0.68	-0.66	-0.77	-0.76	-0.65
x_{t-9}	-0.5	-0.62	-0.57	-0.54	-0.61	-0.58	-0.52
Error	-	0.112	0.065	0.040	0.128	0.122	0.046

Table 1 shows that when the adaptive Metropolis algorithm was used to obtain a distribution of values for each parameter (c), the network was able to improve upon not only the results of the network trained by backpropagation (a), but also the results of the network trained by the SCE algorithm (b), in terms of determining the correct weightings of the model inputs.

It is also shown in Table 1 that, by obtaining a distribution of values for each parameter, the network was able to find a robust estimate of the weight vector given limited data (f). This approach was therefore able to overcome the effects of overtraining while still using all of the information contained in the data set. The results of the network that used cross-validation to prevent overtraining (e) showed little improvement over the overtrained network (d). This is most likely due to the loss of information resulting from the reduced size of the training set.

Approaches b and e are essentially the same, except that approach b was trained on twice the amount of data that approach e was trained on. The improvement in the results of b compared to e is significant. On the other hand, by comparing the results of approaches c and f it can be seen that only a minor improvement in the results was achieved by using twice the amount of data when Bayesian methods were employed. It is expected that with increasing data the results obtained using Bayesian methods would continue to improve, however, it has been shown that in order to produce reasonable results a large data set is not necessary.

Overall, the network trained on 300 data points and employing Bayesian methods (c) performed the best. However, the network trained on 150 data points, also employing Bayesian methods (f) performed better than both of the networks trained with 300 data points using standard neural network methods (a and b).

4.2. Quantification of Prediction Uncertainty

The Metropolis output for parameter 1 is plotted against the output for parameter 2 in Figure 2. For comparison, the Metropolis output for a network with no hidden nodes (optimal structure) has also been included in the figure. It can be seen that the parameter ranges for the network with 1 hidden node are significantly wider than the ranges when there are no hidden nodes, indicating a higher degree of uncertainty. Additionally, the parameters display a high degree of correlation when a hidden node is included in the network.

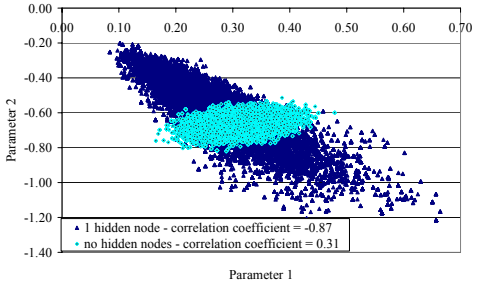

Figure 2. Metropolis output of parameters 1 & 2

A plot of output from approach c (detailed in Section 3.2) is displayed in Figure 3. 95% confidence intervals are included in the figure for the cases when the correlation structure between network weights was retained and when it was not.

An inspection of the covariance matrix of the parameters showed that parameters 1, 2, 3 and 5 were highly correlated with covariance values greater than 0.85. Given the reasonably wide ranges of the parameters (e.g. Figure 2), the tightness of the 95% confidence intervals obtained by retaining the correlation structure in the weights indicates that, although the values for the weights are not unique, as long as the correlation structure between the weights is preserved, predictions may be made with confidence. This was confirmed by inspecting the 95% confidence intervals obtained when the correlation structure was ignored. These bounds are much wider, indicating that there is much greater uncertainty in the predictions.

5. CONCLUSIONS

This study has demonstrated that the explicit assessment of parameter uncertainty can be extremely beneficial in generating accurate predictions from ANNs. The investigations carried out enabled the following conclusions

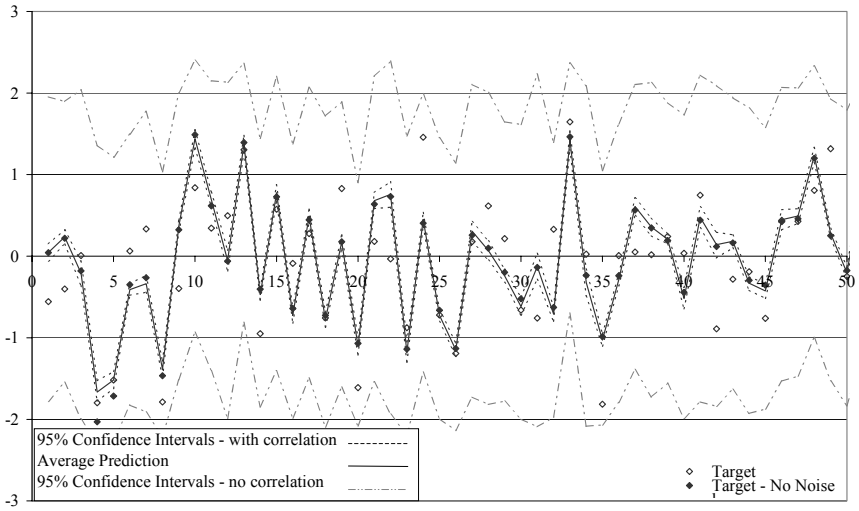

Figure 3. Plot of outputs from ANN with 95% confidence intervals.

to be made:

- The incorporation of Bayesian approaches provide ANNs with the ability to find robust weight estimates in the presence of local minima.
- The incorporation of Bayesian approaches provide ANNs with the ability to find robust weight estimates given limited data and these ANNs are capable of performing better than networks trained on larger data sets using standard approaches.
- The results obtained using ANNs that incorporate Bayesian methods improve with increased data, however comparable results may be obtained with a significantly smaller data set and therefore a large data set is not essential for the success of this approach.

It has also been shown that, although the connection weights of an ANN are not unique, confident predictions may be made with ANNs as long as the correlation structure between the weights is considered. Future investigations will include examining the correlations between weights to help determine the optimal network structure.

6. ACKNOWLEDGMENTS

The authors would like to thank Tom Micevski from the University of Newcastle, Australia, for providing the code for the adaptive Metropolis algorithm.

7. REFERENCES

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. Artificial neural networks in hydrology. I: Preliminary concepts, *Journal of Hydrologic Engineering, ASCE*, 5(2), 115-123, 2000.

Duan, Q., S. Sorooshian, and V. K. Gupta. Effective and efficient global optimisation for conceptual rainfall-runoff models, *Water Resources Research*, 28(4), 1015-1031, 1992.

Haario, H., E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm, *Bernoulli*, 7(2), 223-242, 2001.

Kingston, G. B., H. R. Maier, and M. F. Lambert. Understanding the mechanisms modelled by artificial neural networks for hydrological prediction, *MODSIM*, Townsville, Australia, 2003.

MacKay, D. J. C. Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks, *Network: Computation in Neural Systems*, 6(3), 469-505, 1995.

Maier, H. R., and G. C. Dandy. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, *Environmental Modelling and Software*, 15, 101-124, 2000.

Neal, R. M. Bayesian training of backpropagation networks by the hybrid Monte Carlo method, *Technical Report CRG-TR-92-1*, Department of Computer Science, University of Toronto, Toronto, 1992.

Thyer, M., G. Kuczera, and Q. J. Wang. Quantifying uncertainty in stochastic models using the Box-Cox transformation, *Journal of Hydrology*, 265, 246-257, 2002.