19th International Congress on Modelling and Simulation, Perth, Australia, 12–16 December 2011
http://mssanz.org.au/modsim2011

Investigation of artificial neural network models for streamflow forecasting

H. D. Tran[a,b], N. Muttil[a] and B. J. C. Perera[c]

[a] *School of Engineering and Science, Victoria University, Victoria*
[b] *Institute for Sustainability and Innovation, Victoria University, Victoria*
[c] *Faculty of Health, Engineering and Science, Victoria University, Victoria*
Email: dung.tran@vu.edu.au

Abstract: Time series forecasting is the use of a model to forecast future events based on known past events. Accurate forecasts of time series variables at different time scales are becoming increasingly necessary to facilitate mitigation of negative impacts of climate change, maximisation of system benefits from improved planning and management, and minimisation of system failure risks in all social, economical and environmental activities. Examples include hourly forecasts of rainfall (1-48 hours ahead), which are useful for flood warning, and monthly forecasts of streamflow (1-12 months ahead), which are beneficial for planning and operation of water supply systems. There are many other applications in finance and economics such as tourism demand forecasting and stock forecasting.

Artificial neural network (ANN) models, which are considered as a category of the data-driven techniques, have been widely used in streamflow forecasting. Several distinguishing features of ANN models make them valuable and attractive for forecasting tasks. First, there are few a priori assumptions about the models as opposed to model-driven techniques. They learn from examples and capture the functional relationships among the data even if the underlying relationships are too complex to specify. Second, ANN models can generalize after learning from the sample data presented to them. Third, ANN models are universal functional approximators for any continuous function to the desired accuracy. Fourth, ANN models have flexible structures that allow multi-input and multi-output modelling. This is particularly important in streamflow forecasting where inflows at multiple locations are considered within a catchment.

This paper investigated two basic ANN models, namely, feed forward neural network (FFNN) and layered recurrent neural network (LRNN) for streamflow forecasting in an attempt to understand why ANN models were used successfully in some streamflow forecasting studies but not always. In our study, two hypothetical and two real datasets were used to test performance of two different ANN models using feed forward and layered recurrent structures. Furthermore, an existing input selection technique using partial mutual information (PMI) approach, which can remove the insignificant inputs and thus potentially enhance the performance of ANN models, is also investigated.

The results showed that the PMI approach correctly identified significant inputs of the two hypothetical datasets. However, the forecasting performance of FFNN and LRNN were not enhanced, when PMI identified inputs were used in comparison to using all inputs. The LRNN did not outperform the FFNN, although it is expected to perform better. Performance of both FFNN and LRNN models are related to noise level and autoregressive feature of time series data.

Keywords: *artificial neural networks, streamflow forecasting, input selection*

1. INTRODUCTION

Forecasting streamflow is an important task since it can help in short term operation of water supply systems as well as providing early warning of river flooding. The general difficulty associated with streamflow forecasting is the non-linear and non-stationary characteristics which are often encountered in most streamflow time series data (Coulibaly et al. 2001).

Artificial neural networks (ANN) models, which are considered as a category of the data-driven techniques, have been widely used in streamflow forecasting. Several distinguishing features of ANN models make them valuable and attractive for forecasting tasks (Maier and Dandy 2000; Samarasinghe 2006). First, there are few a priori assumptions about the models as opposed to model-driven techniques. They learn from examples and capture the functional relationships among the data even if the underlying relationships are too complex to specify. Second, ANN models can generalize after learning from the sample data presented to them. Third, ANN models are universal functional approximators for any continuous function to the desired accuracy. Fourth, ANN models have flexible structures that allow multi-input and multi-output modelling. This is particularly important in streamflow forecasting where inflows at multiple locations are considered within a catchment.

There are many different types of ANN architectures and among them, feed forward and recurrent neural networks have recently gained attention in literature. Feed forward networks are static, and are the most common form applied in hydrology due to its simple framework (Güldal and Tongal 2010). The static networks can only simulate the short-term memory structures within processes. In contrary, the recurrent neural networks provide a representation of dynamic internal feedback loops to store information for later use and to enhance the efficiency of learning.

This paper investigated two basic ANN models using feed forward and recurrent structures for streamflow forecasting in an attempt to understand why ANN models were used successfully in some streamflow forecasting studies but not always. This inconsistency was observed in a recently published study by Wang et al. (2009) involving similar ANN models for forecasting streamflow time series at two different locations, one demonstrating a high E (Nash-Sutcliffe efficiency coefficient) of 0.87, while the other showing a medium E of 0.61, between observed and forecasted values on validation datasets. In our study, two hypothetical and two real datasets were be used to test performance of these ANN models. The modelling difficulty of all four datasets was assessed by comparing with the well-known multiple linear regression (MLR) model. Furthermore, an input selection technique using partial mutual information approach (May et al. 2008), which can remove the insignificant inputs and thus potentially enhance the performance of ANN models, was also investigated.

2. DATASETS

Four time series datasets were used in this study to investigate the performance of the two ANN models. The first two datasets were hypothetical time series data and the last two datasets were real streamflow time series data. The description of 4 datasets and their time-series plots are given below.

(1) AR9 (hypothetical data) – Figure 1a

This dataset is a widely used hypothetical dataset with autoregressive feature (May et al. 2008) in which Y_t depends on its three time-lags and a random noise:

$$Y_t = 0.3Y_{t-1} - 0.6Y_{t-4} - 0.5Y_{t-9} + e_t[N_{[0,1]}] \quad (1)$$

(2) YX (hypothetical data) – Figure 1b

This dataset does not have the autoregressive feature since Y_t depends on an independent time series X_t, whose values follow normal distribution, and a random noise (Muñoz and Czernichow 1998):

$$Y_t = 0.5X_{t-1} - 0.6X_{t-4} - 0.2X_{t-4}^2 + 0.3X_{t-6} + 0.3X_{t-6}^2 + e_t[N_{[0,1/20]}] \quad (2)$$

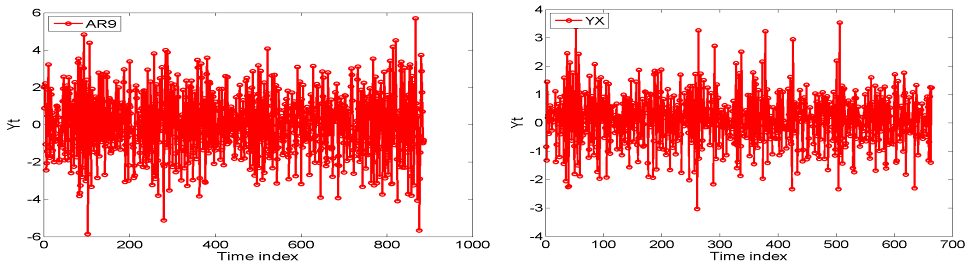
$$X_t = N_{[0,1]} \quad (3)$$

(3) US dataset – Figure 2a

This dataset is based on an observed time series of daily discharge from the gauge 01470500 in U.S. from 2006 to 2008 (Guven 2009).

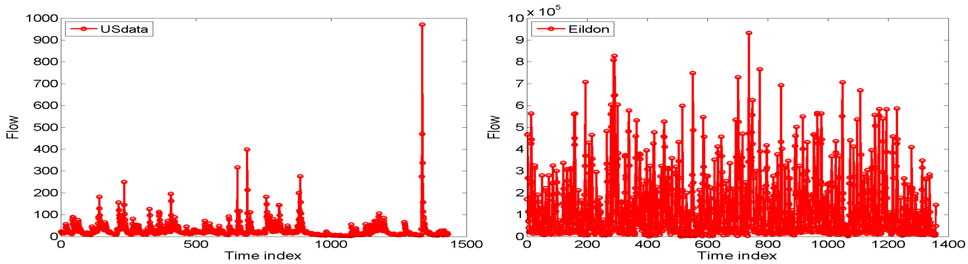
(4) Eildon dataset – Figure 2b

This dataset is based on a monthly time series of Eildon river flow in the North-Western region of Victoria (Australia), from January, 1891 to June, 2007.



(a) (b)

Figure 1 Time series plots of hypothetical AR9 (a) and YX (b) datasets



(a) (b)

Figure 2 Time series plots of real daily flow (m^3/s) for USdata dataset (a) and monthly total flow (m^3) for Eildon (b) dataset

3. PARTIAL MUTUAL INFORMATION (PMI)

Mutual information (MI) measures the dependence between two variables *X* and *Y* by using their joint distributions and marginal densities (Sharma 2000). In other words, the MI measures the reduction in uncertainty of *Y* due to the knowledge of the variable *X*. Based on the MI approach, the partial mutual information (PMI) technique was developed to better reveal the true relationship between *X* and *Y* by removing the interactive relationships with one or more other variables (May et al. 2008):

$$PMI = \iint f_{X',Y'}(x',y')\ln\left[\frac{f_{X',Y'}(x,y)}{f_{X'}(x)f_{Y}^{'}(y)}\right]dx'dy' \quad (4)$$

$$x' = x - E[x\,|\,\mathbf{z}] \quad (5)$$

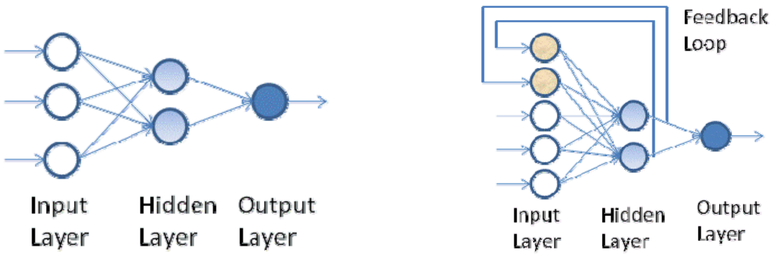
$$y' = y - E[y\,|\,\mathbf{z}] \quad (6)$$

where *X'* and *Y'* are generalized to represent Y_t and lagged time Y_{t-h} with time step *h* conditional on ***Z*** which is a set of remaining time-lag variables. In performing the PMI technique, the input variable that has the highest conditional PMI value at each iteration is added to the selection set. Three statistical tests including bootstrap, Akaike information criteria (AIC) and Hampel distance detailed in May et al. (2008) were also employed in this study to eliminate the occurrence by chance of the dependency between input variables and the output variable identified by the PMI technique. The bootstrap is used to test the quality of statistical estimates based on a sample of data. The AIC is based on penalizing model complexity and the Hampel test is based on outlier detection, which determines whether a given value, *x*, is significantly different to other values.

4. NEURAL NETWORK MODELS

Many types of artificial neural networks have been developed in the last few decades starting from the well known multilayer perceptron structure (Samarasinghe 2006). Each perceptron is a signal processing unit and a network of connected perceptrons is capable of performing classifying and forecasting tasks. The classifying and forecasting tasks are done through a 2-step process of training (i.e. learning from sample data) and simulating (i.e. generalizing to classify or forecast). Multilayer perceptron networks are further classified into static feed-forward neural networks (FFNN) and dynamic recurrent neural networks (RNN) with regards to how signals are transferred between layers of perceptrons. In the static FFNN, signals essentially propagate in the forward direction, while a backward propagation of so-called feedback signals can occur in the RNN (Güldal and Tongal 2010). The static FFNN with a typical layout as shown in Figure 3a, can achieve satisfactory analytical outcomes if sufficient data are available. Nevertheless, in consideration of the temporal or short-term dynamic property, the static neural network has difficulty in recognizing and predicting the reality. The FFNN is investigated in this study.

The definition of a RNN, so-called dynamic neural network, is that at least one feedback link is added to the static neural network as shown in Figure 3b. The RNN allows signals to propagate in both forward and backward directions, which offers the network dynamic memories. In other words, RNN has additional feed back loops either from the output or hidden layer to the input layer that delay and store information from the previous time step that is not present in the architecture feed forward networks. The presence of these feed back loops has a profound impact on the learning capability and performance of the neural network, which allows the network to capture the true hidden dynamic memories or persistence components of nonlinear time-series systems (Carcano et al. 2008). The RNN has been proved to be a powerful method for handling complex systems such as nonlinear time-varying systems. A layered recurrent neural network (LRNN), which is a form of RNN, is also investigated and compared with the FFNN in this study.



(a) (b)

Figure 3 A typical layout of feed-forward neural networks (FFNN) (adapted from Carcano et al. 2008) and recurrent neural networks (RNN) (adapted from Güldal and Tongal 2010)

5. PERFORMANCE INDICATORS

Model performance in this study was evaluated using three performance indicators, the mean square error (MSE), Nash-Sutcliffe coefficient (E) and the mean absolute percentage error (MAPE), as shown in Equations 7-9 (Carcano et al. 2008). The MSE quantifies the differences between observed and predicted values along the time series axis and penalises for the large differences because of its square power. This allows the ANN models to better capture peak values of time series data. The MSE is commonly used as the training objective to be minimized in ANN models. E captures the goodness of fit of the model of interest by comparing it with a naïve model in which the mean value is used as predicted values. The value of E lies between one and minus infinity. A value of unity implies that the model exactly matches observations, zero implies that the model is no better than assuming the mean value, and a negative value indicates that the mean values of the observed time series would be a better predictor than the model. However, the limitations of MSE and E are that they are unable to provide the practical assessment of difference between predicted and observed values. For example, MSE provides similar value of 4 for the difference between observation of 1000 and prediction of 1002 as well as between observation of 1 and prediction of 3 in which it is obviously that the former prediction is very good while the latter prediction is very poor. The MAPE was therefore used in this study to provide scale-free percentage difference between observed and predicted values (Coulibaly et al. 2001).

$$MSE = \frac{\sum_{i=1}^{n}(Q_{obsevred}(i) - Q_{predicted}(i))^2}{n} \quad (7)$$

$$E = 1 - \frac{\sum_{i=1}^{n}(Q_{obsevred}(i) - Q_{predicted}(i))^2}{\sum_{i=1}^{n}(Q_{observed}(i) - \overline{Q_{observed}})^2} \quad (8)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Q_{obsevred}(i) - Q_{predicted}(i)}{Q_{observed}(i)}\right| \quad (9)$$

where n is number of data point.

6. RESULTS

The PMI tool was obtained from May et al. (2008), and was used to select model inputs from potential 15 inputs obtained through time lagged observations for four datasets. That is, 15 past values of observations (i.e. streamflow) were used as the model inputs to forecast current output as one-step ahead forecasting.

The feed-forward neural network (FFNN), layered recurrent neural network (LRNN) and multiple linear regression (MLR) models, which are available in the Matlab software, were used in this study. Both neural network models used only one hidden layer for ease of comparison.

All four datasets were divided into training (60%), validation (20%) and testing (20%) datasets. The testing dataset is the last part of the time series and was kept in time series order. The remaining 80% of dataset was then randomly divided into training and validation datasets to have better chance of representative learning data for FFNN and LRNN models. The MLR model used the whole 80% dataset without splitting into a validation dataset. The training and validation datasets were used in the training process in FFNN and LRNN models, in which the training dataset was used to provide ‘learning knowledge’, while the validation dataset was used to avoid over-fitting. The testing dataset was used only in assessing the forecasting performance of MLR, FFNN and LRNN models. All four datasets were scaled between [-1, 1] and then used in the training and testing.

The training process for FFNN and LRNN models was carried out using Levenberg-Marquardt algorithm. Least square method was used for the calibration of the MLR model. The single objective of minimizing mean square error (MSE) was used in the training process of all models. The remaining two performance indicators (i.e. E and MAPE) were then computed using the training (i.e. combination of training and validation) and testing outcomes. The suitable number of hidden neurons for FFNN and LRNN models was selected using trial and error.

6.1 Input selection

The results of input selection using the PMI method are shown in Table 1. As can be seen, the PMI method correctly identified three lagged inputs of two hypothetical datasets (i.e. AR9 and YX). For the two real datasets, the selected inputs are shown but could not be assessed since the true inputs are unknown.

Table 1 Result of input selection using PMI technique

Dataset	True Inputs	PMI_selected inputs
AR9	Y_{t-1}, Y_{t-4}, Y_{t-9}	Y_{t-1}, Y_{t-4}, Y_{t-9}
YX	X_{t-1}, X_{t-4}, X_{t-6}	X_{t-1}, X_{t-4}, X_{t-6}
USdata	unknown	Q_{t-1}
Eildon	unknown	Q_{t-1}, Q_{t-3}, Q_{t-13}

6.2 Forecasting performance

As stated in Section 1, the MLR model was used to assess how difficult it was to model the four datasets using ANN models. The results of the MLR model are shown in the last column of Tables 2 and 3 in which the MLR was run using all 15 past values as model inputs to predict the current values. The E values computed from the MLR models are within the range (0.4-0.8) for all datasets, which indicates some modelling difficulty.

The forecasting performances of FFNN and LRNN models are shown in Table 2 for training dataset and Table 3 for testing dataset. It should be noted that the MLR model was run only for the case of all 15 inputs for all datasets for comparative purposes. In comparison with the E of MLR model on four datasets, both FFNN and LRNN slightly outperformed the MLR mode with the exception of US dataset.

For AR9 dataset, there is no significant difference in model performance as demonstrated through 3 performance indicators for both training and testing datasets (all inputs or PMI-selected inputs) in FFNN and LRNN models. For remaining datasets, the performance of both FFNN and LRNN models were also not enhanced when PMI-selected inputs was used.

Although the LRNN model is theoretically better than the FFNN model in handling dynamic time series data, the performance of the LRNN in comparison with the FFNN model as evidenced from this study is not up to its expectation.

Table 2 Training performances of FFNN and LRNN

Training dataset	FFNN				LRNN				MLR
	MSE	MAPE	E	Hidden neurons	MSE	MAPE	E	Hidden neurons	E
AR9	0.036	167.9	0.681	10	0.027	163.1	0.7	4	0.646
AR9_pmi	0.031	189.0	0.645	6	0.031	188.7	0.657	8	
YX	0.0043	105.7	0.948	6	0.004	97.4	0.948	4	0.737
YX_pmi	0.045	111.8	0.947	4	0.046	104.6	0.948	12	
USdata	0.026	34.8	0.397	4	0.011	28	0.534	6	0.495
USdata_pmi	0.015	38.0	0.506	6	0.017	19.7	0.504	8	
Eildon	0.037	271	0.595	2	0.039	266	0.61	8	0.549
Eildon_pmi	0.049	226	0.575	8	0.042	254	0.54	8	

Table 3 Testing performances of FFNN and LRNN

Training dataset	FFNN				LRNN				MLR
	MSE	MAPE	E	Hidden neurons	MSE	MAPE	E	Hidden neurons	E
AR9	0.049	294	0.68	10	0.046	253	0.67	4	0.699
AR9_pmi	0.053	164	0.662	6	0.004	205	0.701	8	
YX	0.004	55	0.934	6	0.005	117	0.924	4	0.787
YX_pmi	0.004	46.7	0.935	4	0.045	110	0.932	12	
USdata	0.084	41.4	0.455	4	0.075	36.3	0.363	6	0.470
USdata_pmi	0.063	45.1	0.62	6	0.065	21.9	0.525	8	
Eildon	0.038	42	0.613	2	0.036	38	0.57	8	0.590
Eildon_pmi	0.032	40	0.695	8	0.033	37	0.56	8	

Both FFNN and LRNN models produced good forecasting performance for the hypothetical YX dataset and reduced performance with the hypothetical AR9 dataset. By examining the data generating mechanisms of AR9 and YX datasets, it is suggested that the reason could be related to the autoregressive feature of the AR9 data as well as the noise level (typical noise level in AR9 dataset and small noise level in YX dataset). Both real datasets also showed the autoregressive feature through the significant relation between current and past values (Table 1) and the performance of FFNN and LRNN models were also not high on these two real datasets. Further similar testing of more complex or more challenging autoregressive datasets are therefore required to clarify the noise level and the relationships between autoregressive feature and the performance of ANN models.

Scatter plots between observed and predicted values for Eildon dataset by FFNN and LRNN models are shown in Figure 4. These plots indicate that both FFNN and LRNN models under-predict high flow values. This is another limitation of ANN models that also require future attention. Techniques such as wavelet and singular spectrum analysis described recently in literature could be used to decompose autoregressive time series into time series components that contain important features such as trend and noise of the autoregressive data. These time series components can be used as model inputs to ANN modelling. These techniques will be sought in the future.

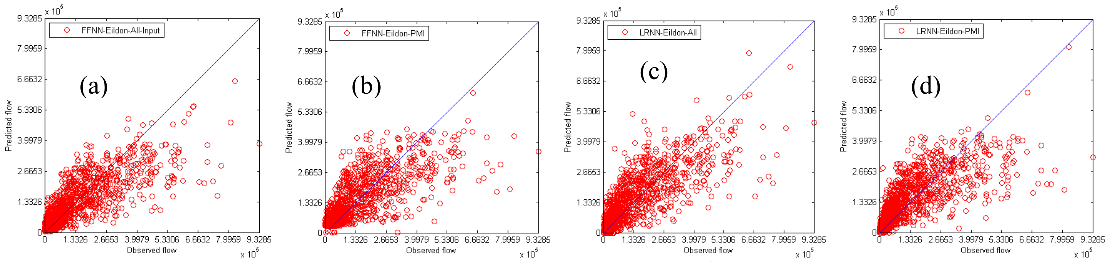


Figure 4 Scatter plots of observed and predicted flow (monthly m^3) for Eildon dataset using 15 lagged inputs (a and c) and PMI-selected inputs (b and d) with FFNN (a and b) and LRNN (c and d) models

7. CONCLUSIONS

This study investigated feed forward neural network (FFNN) and layered recurrent neural network (LRNN) models and a non-linear input selection technique using partial mutual information (PMI) approach for modeling and forecasting dynamic time series data with noise. Two hypothetical datasets with and without autoregressive feature were used together with two real datasets of daily and monthly streamflow. The results showed that the PMI approach correctly identified significant inputs of the two hypothetical datasets. However, the forecasting performance of FFNN and LRNN were not enhanced when PMI identified inputs were used in comparison to using all inputs. The LRNN did not outperform the FFNN, although it is expected to perform better. Initial findings indicated that performance of both FFNN and LRNN are related to noise level and auto-regressive feature of time series data. Future works will focus on further similar testing of more complex or more challenging autoregressive datasets.

REFERENCES

Carcano, E C, Bartolini, P, Muselli, M & Piroddi, L 2008, 'Jordan Recurrent Neural Network versus IHACRES in Modelling Daily Streamflows,' *Journal of Hydrology*, 362, 291-307.

Coulibaly, P, Anctil, F & Bobee, B 2001, 'Multivariate Reservoir Inflow Forecasting Using Temporal Neural Networks,' *Journal of Hydrologic Engineering*, 6(5), 367-376.

Güldal, V & Tongal, H 2010, 'Comparison of Recurrent Neural Network, Adaptive Neuro-Fuzzy Inference System and Stochastic Models in Egirdir Lake Level Forecasting,' *Water Resources Management*, 24, 105-128.

Guven, A 2009, 'Linear Genetic Programming For Time-Series Modelling of Daily Flow Rate,' *Journal of Earth Systems and Science*, 118(2), 137-146.

Krauser, P & Boyle, D P 2005, 'Comparison of Different Efficiency Criteria for Hydrological Model Assessment,' *Advance GeoScience*, 5, 89-97.

Maier, H R & Dandy, G C 2000, 'Neural Networks for The Pediction And Forecasting of Water Resources Variables: A Review of Modelling Issues and Applications,' *Environmental Modelling and Software*, 15(1), 101-124.

May, R J, Maier, H R, Dandy, G C & Fernando, T M K G 2008, 'Non-Linear Variable Selection For Artificial Neural Networks Using Partial Mutual Information,' *Environmental Modelling & Software*, 23(10-11), 1312-1326.

Muñoz, A & Czernichow, T 1998, 'Variable Selection Using Feedforward And Recurrent Neural Networks,' *Engineering Intelligent Systems for Electrical Engineering and Communications*, 6(2), 91-102.

Samarasinghe, S 2006, *Neural Networks for Applied Sciences and Engineering,* Auerbach Publications, New York.

Sharma, A 2000, 'Seasonal to Interannual Rainfall Probabilistic Forecasts For Improved Water Supply Management: Part 1-A Strategy For System Predictor Identification,' *Journal of Hydrology*, 239(1-4), 232-239.

Wang, W-C, Chau, K-W, Cheng, C-T & Qiu, L 2009, 'A Comparison of Performance of Several Artificial Intelligence Methods for Forecasting Monthly Discharge Time Series,' *Journal of Hydrology*, 374(3), 294-306.