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

Rainfall modeling using Artificial Neural Network for a mountainous region in West Iran

F. Mekanik[a], T.S. Lee[b] and M. A. Imteaz[a]

[a] Faculty of Engineering and Industrial Science, Swinburne University of Technology, Melbourne, VIC, Australia,
[b]Faculty of Engineering, Universiti Putra Malaysia, Serdang, Malaysia
Email: fmekanik@groupwise.swin.edu.au

Abstract: One of the major problems in water resources management is rainfall forecasting. With the effect of rainfall on water resources as a foregone conclusion, more accurate prediction of rainfall would enable more efficient utilization of water resources and power generation. Countries depending on agro-based economy could benefit tremendously from accurate long-term rainfall predictions. Thus, long-range forecasts require indefatigable effort and long planning using different methods. This study gives attention to long-term rainfall modelling since a long-term forecasting could provide better information for optimal management of a resource that is to be used over a substantial period of time. The aim of the study is also to investigate the capability of non-linear techniques on long-term rainfall forecasting. One of the non-linear techniques being widely used is the Artificial Neural Networks (ANN) approach which has the ability of mapping between input and output patterns without a priori knowledge of the system being modelled.

The main aim of the study is to develop a model which is capable of forecasting 12 months rainfall in advance. A feedforward Artificial Neural Network (ANN) rainfall model was developed to investigate its potentials in forecasting rainfall. The study area is the west mountainous region of Iran and the model was developed for a synoptic station in this region. Three separate ANN models with three different input data sets were trained. The first model investigated the effect of the number of lags on the performance of the ANN. The number of lags varied from 1-12 previous months. The second model investigated the effect of adding monthly average to the inputs, and the third model considered seasonal average as an extra input data in addition to the ones in the second model. The effect of the number of hidden neurons on ANN modeling was also examined. The models were trained based on the Levenberg-Marquardt algorithm with tansigmoid activation function for the hidden layer and purelin activation function for the output layer. Monthly rainfall data of 1977-2002 were used for training the models. The models were tested with monthly rainfall data of 2003. It was proven that the larger lags outperform the lower ones in ANN modeling. Also, adding the extra monthly and seasonal average to the input data set leads to better model performance. The number of hidden neurons was varied from 1-30. It was demonstrated that input neurons have more effect on performance criteria than the hidden neurons. Simulation results for the independent testing data series showed that the model can perform well in simulating one year monthly rainfall in advance.

Keywords: *Artificial Neural Network, rainfall, long-term prediction*

1. INTRODUCTION

Long-term rainfall prediction models have not been very satisfactory in terms of accuracies when compared with short-term rainfall prediction models. The probable reasons that make conducting long-term rainfall prediction difficult are the complexity of the atmospheric processes and the uncertainty of relationships between rainfall and hydro-meteorological variables. Assuming these relationships, however, might further add to the complexity of forecasting. Also long-term rainfall prediction with the use of numerical models has not demonstrated useful performance since rainfall prediction from such models is an average over grid point values, and therefore being a function of the model spatial resolution only while neglecting the temporal variation will lead to consistent inaccuracies since rainfall is highly variable both in space and time. Most available time series analyses consider linear relationships between variables. However, in the real world, temporal variations in data are difficult to analyze and predict since they do not show simple regularities. The use of nonlinear models such as Artificial Neural Networks (ANN), which are capable of modelling complex nonlinear problems, can be suitable for real world temporal data. Neural networks procedure is considered data driven as opposed to model driven procedures. This is due to its dependence on the available data. Many researchers have been using neural network to forecast future values of possibly noisy time series based on only the past histories.

Many researchers have conducted long-term rainfall modelling in tropical regions like Indian subcontinent either with the use of linear techniques like Box-Jenkins methods (Mishra & Desai, 2005) or nonlinear techniques like ANN modelling. (Chakraverty & Gupta, 2008; Chattopadhyay, 2007; Chattopadhyay & Chattopadhyay, 2010; Guhathakurta, 2008). However, few studies have been conducted for rainfall modelling in mountainous regions like Iran. (Mekanik *et al.*, 2009; Nourani *et al.*, 2009)

2. STUDY AREA

The western part of Iran is basically a mountainous region with steep hillsides, densely populated and mostly relying on rain-fed agriculture. Two of the more important problems affecting the region are the unexpected flash floods and the occurrence of droughts which affect rain-fed cultivation. Due to its steep hillsides, floods bring major disasters to the densely populated cities and villages of this region. Also, most of the cultivation in western Iran is dependent on rainfall thus not having rain at the desired month will affect agriculture. In addition, the region's main sources of municipal and agricultural water consumption are the dams and deep wells and these hydraulic structures are only useful if they are filled by the timely precipitation. All the factors mentioned above makes the pertinent long-term prediction of rainfall necessary in order to reduce the risk in decision making. A long-term rainfall prediction model would be beneficial for water resources management to give a measure of guarantee of water resources for agricultural and industrial requirements.

Hamedan Froudgah station is one of the synoptic stations in the region that is of high concern. The station is located at the hillside of Mount Alvand , the highest summit of pro-Zagros range mountain and it is very close to the city, the airport, the main dam and also the agricultural fields. The station is located at 48^0 32' E, 34^0 51' N and has a height of 1749 meters above sea level. The monthly rainfall data of this station was extracted from the Meteorological Organization of the Islamic Republic of Iran website. (http://www.weather.ir).

3. METHODOLOGY

3.1. Overview

According to Kumar and Satish (2008) the parameters for ANN modelling are basically network topology, neurons characteristics, training and learning rules. Multi-layered perceptrons (MLP) are feed-forward nets with one or more hidden layers between the input and output neurons (Figure 1). The number of input and

output neurons is based on the number of input and output data. Basically, the input layer only serves as receiving the input data for further processing in the network. The hidden layers are a very important part in a MLP since they provide the nonlinearity between the input and output sets. More complex problems can be solved by increasing the number of hidden layers. The process of developing an ANN model is to find a) suitable input data set, b) determine the number of hidden layers and neurons, and c) training and testing the network. (Mekanik, 2011).

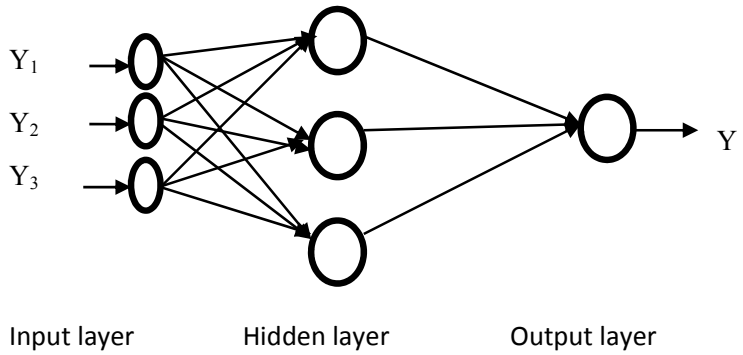

Figure1. Typical ANN architecture

3.2. Input selection

Designing the number and structure of the network inputs involves adequately characterizing the data since rainfall is a complex stochastic process which involves a number of unknown effects (Bodri & ermàk, 2000).Three primary sets of input data were chosen as the inputs of the network, thus three different networks were trained for each data set. The first set is a sliding window which goes through the series and has twelve subsets of its own. Assume $\{Y_t\}$ as the rainfall series, twelve subsets were considered as the inputs. The first subset consist of the first previous month, the second subset would be the previous two consecutive months, and so on until the twelve subset which would be the previous twelve consecutive months. For example, if the output of the model is supposed to be January 1990, the first input would be December 1989. The model will be trained and the results are saved. For the next subset the next previous month, e.g. November 1989 will be added to the input set (November 1989+December 1989) and the model will be trained again. This process will continue until all 12 lags have been added subsequently. Among all these subsets the model with the least mean square error (MSE) is considered. The best model is a 12-M-1 model. i.e. 12 is the number of inputs, M is the number of hidden neurons, 1is the output which will be the next month rainfall. This model is addressed as Model A.

The second set of input data are the rainfall values of the twelve previous months rainfalls plus the monthly average of the output month. This model is a 13-M-1 model which is addressed as Model B. Finally, the third set of input data considered are the rainfall values of the twelve previous months, the monthly average of the output and also the seasonal average of the output. This model is 14-M-1 model, addressed as Model C. As it can be seen, each network consists of different number of input neurons. The only neuron of the output layer is the rainfall at time t+1.

3.3. Hidden layer and neurons

Choosing the number of hidden layers and neurons in the hidden layer is also a critical task. These neurons are responsible for mapping the complex nonlinear relationship between the inputs and the output. One can say that the major concern of developing an ANN structure is the determination of the appropriate hidden

neurons in the hidden layer. There is no systematic way for selecting the best number of hidden neurons while developing an ANN and it is basically problem dependent. Hornik (1991) proved that a single hidden layer network with sufficiently large number of neurons can be used to approximate any measurable functional relationship between input data and the output variable to any desired accuracy. In this study one hidden layer is used. The number of hidden neurons in the hidden layer was determined using the algorithm of dynamic creation. Initially, one hidden neuron is used and the training process is carried out. The final error is calculated. Then the algorithm progressively adds on hidden neurons and calculates the related MSE. A MATLAB M-file was produced that concurrently added the hidden neurons as well as the monthly lagged inputs to the model. The model chosen is the combination of the number of hidden neurons and lagged rainfall that had the least MSE.

3.4. Training and testing the model

The training of an ANN is a nonlinear optimization process. Basically, the error between the model output and the target output is minimized by a predetermined algorithm which repeatedly changes the values of ANN's connection weights. ANN parameters are adjusted during the training process through the minimization of the mean square error (MSE). The *trainlm* function in MATLAB is used for this purpose. This function updates weights and bias values according to Levenberg-Marquardt algorithm. (The MathWorks, 2009)

Every time a network is trained, the weights are randomly initialized. To reduce the effect of random weights on training the network, each network was trained 100 times. The average performance of each of the architecture was calculated and the model with the minimum average performance among all the averages was chosen. The performance criterion is the mean square error. The training was stopped using the "early-stop" technique to decide the optimal learning. This method stops the training when the MSE over the validation set was found to be rising instead of reducing even though the MSE over the training set was still reducing. This technique is used to stop the network from overfitting. An overfitted ANN would perform very well in the training set but fails to maintain the same level of accuracy when applied to the test set.

The best model was applied to the test set to investigate the performance on an unseen set of data. Mean absolute error (MAE) and root mean square error (RMSE) were used as the performance criteria. The index of agreement (*d*) was chosen as a criterion to compare models. The performance criteria are defined as follows:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \quad (1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \quad (2)$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \quad (3)$$

$$d = 1 - \left(\frac{[\sum|\hat{y}_i - y_i|^2]}{[\sum(|\hat{y}_i - \overline{y_i}| + |y_i - \overline{y_i}|)^2]}\right) \quad (4)$$

In these equations $\hat{y}_i$ represents the predicted value of the *ith* observation, y_i is the *ith* observation, and *n* is the sample size.

4. RESULTS AND DISCUSSIONS

4.1. Training results

Three models were developed and trained using the existing data sets. As mentioned in section 3.2, Model A consists of checking 12 different inputs progressively. The performance of Model A with different number of neurons in the hidden layer during training process and different inputs (varying from 1 to 12) for each data set is shown in Figure 2.

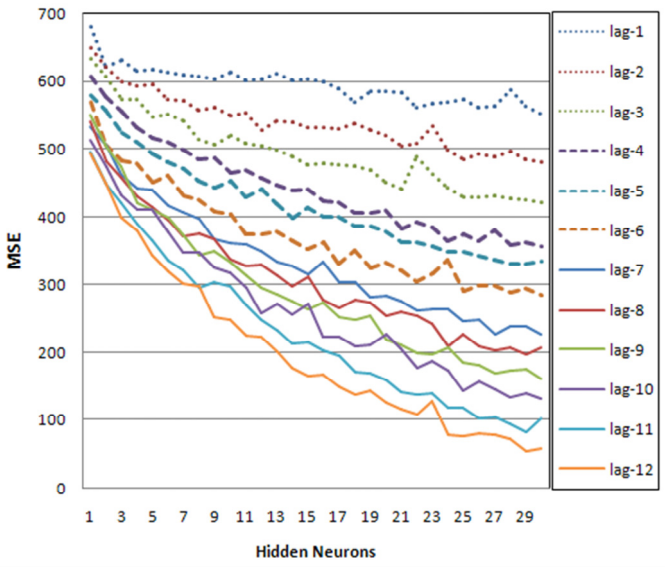

Figure 2: Performance for Model A

Two main results can be derived from Figure 2. Firstly, the plot shows that the MSE decreases with each additional lag. Secondly, the addition of additional hidden neurons also reduces the MSE. Therefore the combination of lags and hidden neurons leads to a reduction in the MSE. It is also clear that the hidden neurons have a positive effect on the model performance, more so than the input neurons. The best architecture for the data set with the minimum corresponding MSE (=54.2) is a 12-29-1 model.

The next two architectures consist of a 13-M-1(Model B) and 14-M-1 (Model C) in which M is the number of hidden neurons in the hidden layer. The performance plot of the models is shown in Figure 3.

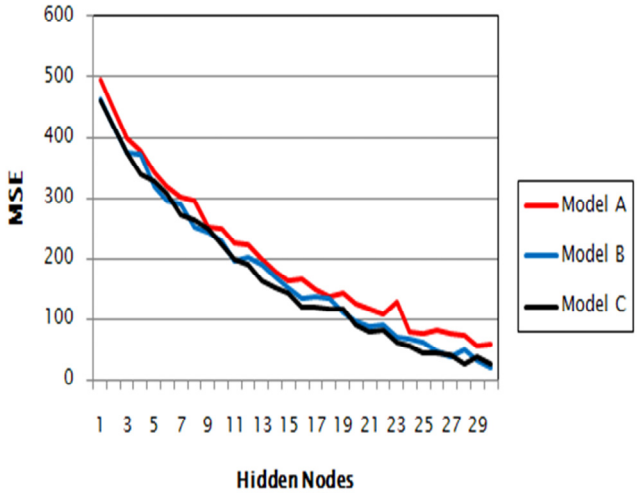

Figure3: Comparison of performance of Model A, B and C

Table 1 shows the effect of adding inputs to the network regarding the MSE. It is obvious that adding monthly average and seasonal average rainfall values to the input data set improves the performance criteria. Model B and C also indicate the same decline of MSE as the number of hidden neurons increases.

However, it seems that considering seasonal average as an extra input data has not reduced the model error. This might be due to adding too much information to the model which has already been captured by the existing inputs. Thus Model B is considered to be the best model for the series.

Figure 4 shows the fitted ANN model to the rainfall time series. It is obvious that ANN could find the seasonal pattern of the series and can simulate the extreme rainfalls as well as the non-rainy months. Some

unusual peaks can be seen in the predicted results. This may force the models to produce a poor generalization results in the testing phase.

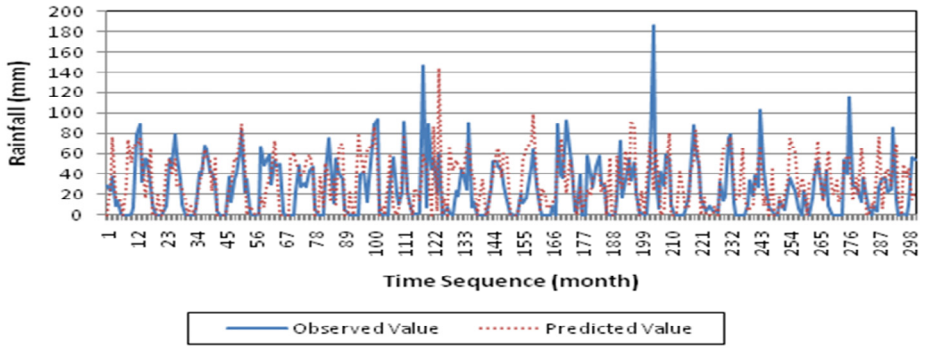

Figure 4: Observed and predicted rainfall in training set

Architecture	MSE
12-29-1 (A)	54.24
13-30-1 (B)	20.97
14-30-1 (C)	24.84

Table 1. Comparison of models fitted to the rainfall series

4.2. Testing results

The next step is predicting the future values of rainfall using the trained model. Figure 5 shows the monthly prediction and also the actual observation for Hamedan Foroudgah for year 2003. It is obvious that the ANN model can follow the pattern of the series. This prediction is quite acceptable since after forecasting the first month (January 2003) the ANN starts using the predicted values of itself as input for the following months forecasts, thus the error is expected to increase as the time sequence goes further.

	MAE	RMSE	*d*
Training	19.48	27	0.69
Testing	15.22	21.41	0.84

Table 2: Index of agreement for the training and testing data sets

Table 2 shows the performance criteria of the model for the training and testing set. The smaller the RMSE and MAE the better the model can forecast. For better understanding of model performance, an addition criterion, the Index of agreement (*d*), has been chosen for model comparison. The closer the value of *d* is to 1, the better the model has been fitted. From Table 2 it can be concluded that Model B has a good generalization. It is necessary to note that while there is enough lagged data to train the model in the training set, this is not the case for the testing set. For predicting 12 months in advance, the proposed model needs to have at least the 12 previous lagged months but this is not valid for this case. After predicting the first month, the model starts using its own prediction to forecast the next 11 months, and as the

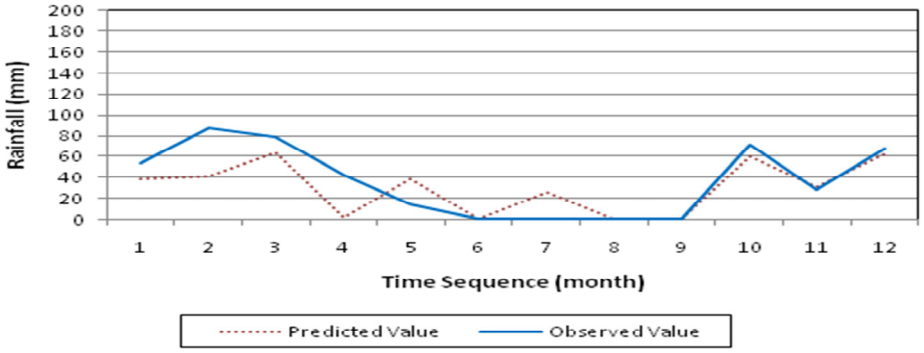

Figure 5: Observed and predicted rainfall (testing set)

prediction goes further more and more predicted values are acting as inputs thus adding to the forecasting error. Considering this fact, the model is still showing good generalization for the testing set. However, updating the model with the real values after each prediction can be addressed through future research.

5. CONCLUSION AND RECOMMENDATIONS

This study attempted to investigate the effectiveness of Artificial Neural Networks on monthly rainfall forecasting for a highland region. A monthly feed forward multi layer perceptron neural network (ANN) rainfall forecasting model was developed for a station in the west mountainous region of Iran. The model was trained based on Levenberg-Marquardt algorithm with tansigmiod activation function for the hidden layer and purelin activation function for the output layer. It was shown that networks with higher lags outperforms the ones with lower lags which reveals the long-term memory characteristic of rainfall at the station. Simulation results for the independent test set showed that although the model has to start using its own prediction as the input after lag-2 of the test set, it still has a good prediction. The ANN model has been constructed to depict the non-linear relationship among rainfall series. It was shown that ANN has the ability of forecasting rainfall for one year in advance. The results may be applicable for future water resources management, drought forecasting and municipal decision making and planning. The results of this study are from one stations only, thus the results need to be explored further through similar studies in the region at similar and also different elevation.

This study used a MLP network for rainfall modeling and prediction, hence checking other types of ANN like recurrent network architectures might be a good suggestion in order to examine the capability of ANN in long-term forecasting of rainfall. The authors recommend a future study which updates the rainfall values after each month of predicting so the model will be able to use real values other than the predicted ones for its future forecast.

6. REFERENCES

Bodri, L., & ermàk, V. (2000). Prediction of extreme precipitation using a neural network: application to summer flood occurrence in Moravia. *Advances in Engineering Software, 31*(5), 311-321.

Chakraverty, S., & Gupta, P. (2008). Comparison of neural network configurations in the long-range forecast of southwest monsoon rainfall over India. *Neural Computing & Applications, 17*(2), 187-192.

Chattopadhyay, S. (2007). Feed forward Artificial Neural Network model to predict the average summer-monsoon rainfall in India. *Acta Geophysica, 55*(3), 369-382.

Chattopadhyay, S., & Chattopadhyay, G. (2010). Univariate modelling of summer-monsoon rainfall time series: Comparison between ARIMA and ARNN. *Comptes Rendus Geosciences, 342*(2), 100-107.

Guhathakurta, P. (2008). Long lead monsoon rainfall prediction for meteorological sub-divisions of India using deterministic artificial neural network model. *Meteorology and Atmospheric Physics, 101*(1), 93-108.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks, 4*(2), 251-257.

Islamic republic of Iran meteorological organization, veiwd Dec 2008, www.weather.ir

Kumar, D. N., & Sathish, T. (2008). Forecasting Hydrologic Time Series Using Artificial Neural Networks.

The MathWorks, Inc 1984-2009, www.mathworks.com

Mekanik, F. (2011). Rainfall Time Series Modeling for a Mountainous Region in West Iran. Master Thesis, Universiti Putra Malaysia.

Mekanik, F., Lee, T. S., & Shitan, M., (2009) Time Series Modelling of Rainfall in Hamedan, Iran. Proceedings of the Sains Matematik Ke-17, Melaka, Malaysia, Dec. 15-17

Mishra, A., & Desai, V. (2005). Drought forecasting using stochastic models. *Stochastic Environmental Research and Risk Assessment, 19*(5), 326-339.

Nourani, V., Alami, M. T., & Aminfar, M. H. (2009). A combined neural-wavelet model for prediction of Ligvanchai watershed precipitation. *Engineering Applications of Artificial Intelligence, 22*(3), 466-472.