Rainfall-Runoff Modeling Using Artificial Neural Network

Sobri Harun**, Ph.D** & Nor Irwan Ahmat Nor**, M.Eng**
Department of Hydraulics and Hydrologic
Faculty of Civil Engineering,
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia

Amir Hashim Mohd. Kassim**, Ph.D**
Faculty of Engineering,
Kolej Universiti Teknologi Tun Hussein Onn

ABSTRACT

The Artificial Neural Network (ANN) is a method of computation inspired by studies of the brain and nervous systems in biological organisms. A neural network method is considered as a robust tools for modelling many of complex non-linear hydrologic processes. It is a flexible mathematical structure which is capable of modelling the rainfall-runoff relationship due to its ability to generalize patterns in imprecise or 'noisy' and ambiguous input and output data sets. This paper describes the application of multilayer perceptron (MLP) and radial basis function (RBF) to predict daily runoff as a function of daily rainfall for the Sungai Lui, Sungai Klang, Sungai Bekok, Sungai Slim and Sungai Ketil catchments area. The performance of ANN is evaluated based on the efficiency and the error. It has been found that the ANN has a potential for successful application to the problem of runoff prediction.

Keywords: *Artificial Neural Network, MLP, RBF, Rainfall-Runoff Modelling*

INTRODUCTION

The relationship of rainfall-runoff is known to be highly non-linear and complex. The rainfall-runoff relationship is one of the most complex hydrologic phenomena to comprehend due to the tremendous spatial and temporal variability of watershed characteristics and precipitation patterns, and the number of variables involved in the modelling of the physical processes [1]. Hydrologists are often confronted with problems of prediction and estimation of runoff, precipitation, contaminant concentrations, water stages, and so on [2]. Although many watersheds have been gauged to provide continuous records of stream flow, hydrologists are often faced with situations where little or no information is available. In such instances, simulation models are often used to generate synthetic flows. The available rainfall-runoff models are HEC-HMS, MIKE-11, SWMM, etc. These models are useful for the hydrologic and hydraulic engineering planning and design as well as water resources management; e.g., hydropower generation, flood protection and irrigation. The existing popular model is considered as not flexible and they require many parameters. Obviously, the models have their own weaknesses. Therefore, in view of the importance of the relationship between rainfall-runoff, the present study was undertaken in order to develop rainfall-runoff models that can be used to provide reliable and accurate estimates of runoff.

ANN models have been used successfully to model complex non-linear input-output relationships in an extremely interdisciplinary field. The natural behaviour of hydrological processes is appropriate for the application of ANN method. In terms of hydrologic applications, this modelling tool is still in its nascent stages [2]. Several studies indicate that ANN have proven to be potentially useful tools in hydrological modelling such as for modelling of rainfall-runoff processes [3, 4, 5, 1]; flow prediction [6, 7]; water quality predictions [8]; operation of reservoir system [9, 10]; groundwater reclamation problems [11]; etc. The objective of the present study are to develop rainfall-runoff models using ANN methods. The modelling work is carried out using 5 years period of the rainfall and runoff records from five selected catchments in Peninsular of Malaysia. Those are Sungai Lui (Selangor), Sungai Bekok (Johor), Sungai Slim (Perak), Sungai Ketil (Kedah), and Sungai Klang (Kuala Lumpur) in the central part of Malaysia. Sg. Ketil catchment is a fully natural area and covers about 704 km^2 of catchment area. The Sg. Lui (68.1 km^2), Sg. Bekok (350 km^2) and Sg. Slim (455 km^2) catchments are semi-developed area. While, the Sg. Klang catchment is a fully developed area consists of 468 km^2 of catchment area.

NEURAL NETWORK MODEL

An ANN can be defined as 'a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain' [12]. The ANN try mimic the functioning of the human brain, which contains billions of neurons and their interconnections. Two types of neural network architectures, namely multilayer perceptron (MLP) and radial basis function (RBF) network are implemented. The architecture of an ANN is designed by weights between neurons, a transfer function that controls the generation of output in a neuron, and learning laws that define the relative importance of weights for input to a neuron [13]. The objective of ANN is to process the information in a way that is previously trained, to generate satisfactory results. Neural network can learn from experience, generalize from previous examples to new ones, and abstract essential characteristics from inputs containing irrelevant data [14]. The main control parameters of ANN model are interneuron connection strengths also known as weights and the biases. In all cases, the output layer had only one neuron, that is, the runoff.

Multilayer Perceptron

The first technique of neural network modelling is the MLP model, and the architecture of a typical neuron with single hidden layer is shown in Figure 1. Basically the MLP consists of three layers: the input layers, where the data are introduced to the network; the hidden layer, where the data are processed (that can be one or more) and the output layer, where the results for given inputs are produced.

Each layer is made up of several nodes, and layers are interconnected by sets of correlation weights. Each input node unit (*i=1,…,m*) in input layer broadcasts the

input signal to the hidden layer. Each hidden node (*j=1,…,n*) sums its weighted input signals according to

$$z\_in_j = w_{0j} + \sum_{i=1}^{m} x_i w_{ij} \tag{1}$$

applies its activation function to compute its output signal from the input data as

$$z_j = f(z\_in_j) \tag{2}$$

and sends this signal to all units in the hidden layer. Note that w_{ij} is the weight between input layer and hidden layer, w_{0j} is the weight for the bias; and x_i is the input rainfall signal. The net of a neuron is passed through an activation or transfer function to produce its result. Therefore, continuous-transfer functions are desirable. In this study, a sigmoid function used is hyperbolic-tangent (tansig) as proposed by [1]. This function is continuous, differentiable everywhere, monotonically increasing, and it is the most commonly used in the backpropagation networks. The output is always bounded between 1 and -1, and the input to the function can vary between plus or minus infinity $(\pm\infty)$. The tansig sigmoid activation function will process the signal that passes from each node by

$$f(z\_in_j) = \frac{2}{1 + e^{-2z\_in_j}} - 1 \tag{3}$$

Then from second layer the signal is transmitted to third layer. The output unit ($k=1$) sums its weighted input signals and applies its activation function to compute its output signal. The output node ($k=1$) receives a target pattern corresponding to the input training pattern, computes its error information, calculates its weight correction (used to update $c_j^{(k)}$ later), and its bias correction (used to update $c_0^{(k)}$ later) term. Note that, $c_j^{(k)}$ is the weight between second layer and third layer; $c_0^{(k)}$ is the weight for bias, and $\vec{y}^{(k)}$ is the neural network output. The error information is transfer from the output layer back to early layers. This is known as the backpropagation of the output error to the input nodes to correct the weights.

Back-propagation is the most commonly used supervised training algorithm in the multilayer feed-forward networks. The objective of a backpropagation network is to find the weight that approximate target values of output with a selected accuracy. The network weights are modified by minimizing the error between a target and computed outputs. The error between the output of the network and the target outputs are computed at the end of each forward pass. If an error is higher than a selected value, the procedure continuous with a reverse pass, otherwise, training is stopped. The weights are updated continuously until minimum error is achieved. The least mean square error (MSE) method is used to optimize the network weights in backpropagation networks.

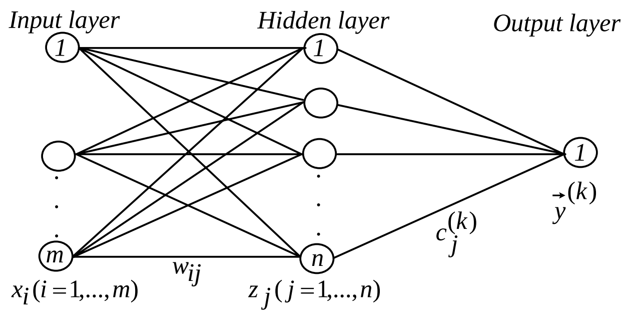

Figure 1: Structure of a MLP model with single hidden layer

Radial Basis Function

The second technique of the neural network modeling is the RBF. RBF is supervised and feed forward neural network. Figure 2 illustrates the designed architecture of the RBF. The RBF can be considered as a three layer network. The hidden layer of RBF network consists of a number of nodes and a parameter vector called a ‘center’ which can be considered the weight vector. The standard Euclidean distance is used to measure how far an input vector from the center is. In the RBF, the design of neural networks is a curve-fitting problem in a high dimensional space [15]. Training the RBF network implies finding the set of basis nodes and weights. Therefore, the learning process is to find the best fit to the training data.

The transfer functions of the nodes are governed by nonlinear functions that is assumed to be an approximation of the influence that data points have at the center. The transfer function of a RBF network is mostly built up of Gaussian rather than sigmoids (see[6]). The Gaussian functions decrease with distance from the center. The transfer functions of the nodes are governed by nonlinear functions that is assumed to be an approximation of the influence that data points have at the center.

The Euclidean length is represented by r_j that measures the radial distance between the datum vector $\underline{y}(y_1, y_2, ..., y_m)$; and the radial center $\underline{Y}^{(j)} = (w_{1j}, w_{2j}, ..., w_{mj})$; can be written as:

$$r_j = \left\| \underline{y} - \underline{Y}^{(j)} \right\| = \left[\sum_{i=1}^{m} (y_i - w_{ij})^2 \right]^{\frac{1}{2}} \qquad (4)$$

A suitable transfer function is then applied to r_j to give,

$$\phi(r_j) = \phi\left(\left\|y - \underline{Y}^{(k)}\right\|\right) \tag{5}$$

Finally the output layer ($k=1$) receives a weighted linear combination of $\phi(r_j)$,

$$\vec{y}^{(k)} = w_0 + \sum_{j=1}^{n} c_j^{(k)} \phi(r_j) \;= w_0 + \sum_{j=1}^{n} c_j^{(k)} \phi\left(\left\|\underline{y} - \underline{Y}^{(j)}\right\|\right) \tag{6}$$

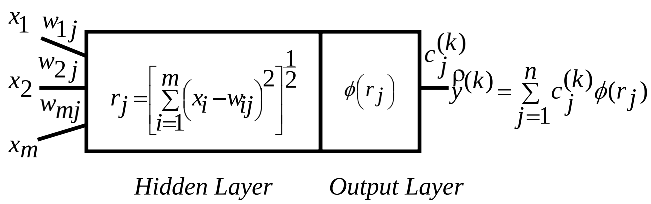

Figure 2: Structure of a RBF model

MODEL APPLICATION

The steps involved in the identification of a nonlinear model of a system are selection of input-output data suitable for calibration and verification; selection of a model structure and estimation of its parameters; and validation of the identified models.

The selection of training data that represents the characteristics of a watershed and meteorological patterns is extremely important in modeling [16]. Input variable (rainfall) were selected to describe the physical phenomena of the rainfall-runoff process, in order to forecast runoff. Record of 5 years of daily rainfall-runoff series of Sungai Lui, Sungai Klang, Sungai Bekok, Sungai Slim and Sungai Ketil catchments is selected to evaluate the performance of the neural network model. The data used consist of two sets: the first four years was used for model calibration (training) in the case of ANN, and the remaining one year of data was used for model verification (testing). The most current data were used in the test set to illustrate the capability of model in predicting future occurrences of runoff, without directly including the land-use characteristics of the watersheds. The original rainfall- runoff data are normalized before the neural network computation is carried out for MLP model. Normalization will transform the original data into the range of +1 to –1. The equation of normalization is:

$$y_t = 2 \times \frac{(x_t - x_{\min})}{(x_{\max} - x_{\min})} - 1 \qquad (7)$$

where x_t is the original series, y_t is the transformed series, $x_{\min}$ is the minimum value of the original series and $x_{\max}$ is the maximum value of the original series.

The ANN model is suited for modelling highly nonlinear input-output relationships such as those encountered in the transformed from rainfall to runoff. In this particular study,the structure of ANN model is designed based on methods by [1]. This model treat the rainfall as directly related to runoff at the present time t, by using the following equation,

$$y(t) = f\{x(t)\} \qquad (8)$$

The goodness-of-fit statistics were computed for both training and testing for each ANN architecture. The input node at $(t-1)$ is added as an additional input variable to the model. During training and testing the goodness of fit statistics is used to evaluate the suitability of input variable $(t-1)$. This procedure is repeated by adding rainfall at previous time periods as input variable until there is no significant change in model accuracy. There are no fixed rules about the number of nodes in the hidden layer. A trial and error procedure is generally applied in selecting the number of hidden layers and in assigning the number of nodes to each of these layers. [17] proposed that normally neural networks were developed using 15, 30, 45, 60 and 100 hidden nodes. This procedure is also considered to examine the performance of neural network model with different number of hidden nodes and hidden layers.

MODEL PERFORMANCE CRITERIA

The performance of each model is evaluated using the coefficient of efficiency (COE), mean square error (MSE), mean absolute error (MAE), and mean relative error (MRE). MSE, MAE, and MRE are the most commonly used performance measures in hydrological modeling and the ideal value is zero. Computed value of COE exhibits the model efficiency and the ideal value is 1.0. The MSE, MAE, and MRE are expressed as the following equations:

$$MSE = \left[\frac{1}{n}\sum_{i=1}^{n}\left(y_{pi} - y_{oi}\right)^2\right]^{\frac{1}{2}} \qquad (9)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left| y_{pi} - y_{oi} \right| \tag{10}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left| \frac{\left(y_{pi} - y_{oi} \right)}{y_{oi}} \right| \tag{11}$$

where, y_{pi} and y_{oi} are the predicted and observed values of output respectively; n is the number of observations or time periods over which the errors are computed. A model with the minimum error is considered the best choice.

RESULTS AND DISCUSSION

Tables 1-5 present the COE, MSE, MAE, and MRE resulting from MLP and RBF models for the five catchments (Sungai Lui, Sungai Klang, Sungai Bekok, Sungai Slim and Sungai Ketil).

Table 1: Results of the Sungai Lui catchment

MODEL		*TRAINING*			
		COE R	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*18-18-1	0.635	0.926	0.685	1.208
	*18-18-7-1	0.622	***0.921***	***0.680***	***1.172***
RBF	18 input nodes	0.982	0.898	0.618	***0.769***
MODEL		*TESTING*			
		COE R	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*18-18-1	0.641	0.654	0.518	1.385
	*18-18-7-1	0.654	0.650	0.515	1.343
RBF	18 input nodes	0.966	0.596	0.442	1.510

*input nodes-hidden nodes-output nodes; cumecs-meter cubic second

Table 2: Results of Sungai Klang catchment

MODEL		*TRAINING*			
		COER	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*17-19-1	0.689	16.942	11.658	0.750
	*17-19-8-1	0.637	***16.604***	***11.920***	***0.811***
RBF	17 input nodes	0.645	14.645	9.866	***0.582***
MODEL		*TESTING*			
		COER	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*17-19-1	0.759	16.618	11.717	0.630
	*17-19-8-1	0.785	16.493	12.174	0.688
RBF	17 input nodes	0.710	14.995	9.814	0.435

*input nodes-hidden nodes-output nodes; cumecs-meter cubic second

Table 3: Results of Sungai Bekok catchment

MODEL		*TRAINING*			
		COER	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*17-23-1	0.711	0.727	0.560	0.109
	*17-23-2-1	0.714	***0.755***	***0.591***	***0.119***
RBF	17 input nodes	0.761	0.740	0.564	***0.110***
MODEL		*TESTING*			
		COER	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*17-23-1	0.737	0.522	0.420	0.086
	*17-23-2-1	0.731	0.586	0.490	0.103
RBF	17 input nodes	0.782	0.498	0.382	0.078

*input nodes-hidden nodes-output nodes; cumecs-meter cubic second

Table 4: Results of Sungai Slim catchment

MODEL		*TRAINING*			
		COER	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*17-20-1	0.661	0.145	0.114	0.002
	*17-20-12-1	0.624	***0.148***	***0.115***	***0.002***
RBF	17 input nodes	0.903	0.140	0.104	***0.002***
MODEL		*TESTING*			
		COER	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*17-20-1	0.729	0.159	0.130	0.002
	*17-20-12-1	0.710	0.157	0.126	0.002
RBF	17 input nodes	0.958	0.159	0.114	0.002

*input nodes-hidden nodes-output nodes; cumecs-meter cubic second

Table 5: Results of Sungai Ketil catchment

MODEL		*TRAINING*			
		COER	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*17-18-1	0.613	0.600	0.442	0.015
	*17-18-5-1	0.620	***0.618***	***0.453***	***0.015***
RBF	17 input nodes	0.690	0.574	0.407	***0.013***
MODEL		*TESTING*			
		COER	**MSE (cumecs)**	**MAE (cumecs)**	**MRE**
MLP	*17-18-1	0.740	0.637	0.525	0.018
	*17-18-5-1	0.726	0.605	0.482	0.016
RBF	17 input nodes	0.678	0.561	0.411	0.014

*input nodes-hidden nodes-output nodes; cumecs-meter cubic second

Table 6: The average, minimum, and maximum flow

TRAINING	Flow (cumecs)		
	Average	**Minimum**	**Maximum**
Sungai Lui	1.501	0.00	7.73
Sungai Klang	17.503	2.00	119.00
Sungai Bekok	1.998	0.91	4.19
Sungai Slim	0.676	0.43	1.37
Sungai Ketil	1.921	0.70	5.15
TESTING	**Flow (cumecs)**		
	Average	**Minimum**	Maximum
Sungai Lui	0.880	0.02	2.83
Sungai Klang	20.732	5.00	117.00
Sungai Bekok	1.774	1.17	3.50
Sungai Slim	0.708	0.50	1.37
Sungai Ketil	1.817	1.15	4.47

* cumecs-meter cubic second

In general, there are two families of model (MLP and RBF) with three types of neural models. The first is MLP with one hidden layer; the second is MLP with two hidden layer; and finally the third is the RBF model. The development of neural network model structure adopt the method by Tokar and Johnson (1999). The results are shown in Table 1 (Sungai Lui with 18 input nodes); Table 2 (Sungai Klang with 17 input nodes); Table 3 (Sungai Bekok with 17 input nodes); Table 4 (Sungai Slim with 17 input nodes); and Table 5 (Sungai Ketil with 17 input nodes). Sungai Ketil catchment (704 km^2) is 10 times bigger than Sungai Lui catchment (68.1 km^2) and 2 times bigger than Sungai Bekok catchment (350 km^2). When, Sungai Klang and Sungai Slim have relatively the same magnitude of catchment area. The average, minimum, and maximum flow of the five rivers are shown in Table 6. It can be seen that the levels of efficiency of the five catchments were improved in the testing stage when the models were trained properly. Further the COE for Sungai Bekok is better than the Sungai Ketil. Probably, the size of the catchment contribute to the inaccuracy of neural modelling. A large fully developed catchment such as Sungai Klang generates considerably a higher peak flood discharge. The neural network model require sufficient amount of data with a large peak discharge during training and generalization.

Results of COE, MSE, MRE, and MAE for Sungai Lui, Sungai Bekok and Sungai Slim reflect that the RBF models consistently display a better performance compared to the MLP model. Further more, the advantage of RBF is that it can be trained much faster than the MLP. It is also found that ANN performance is hardly influenced by the level of non-linearity, and the selection of training data. A large number of training data sets are required to perform successful training.

The number of hidden layer neurons significantly influences the performance of a network. If this number is small, the network may not achieve a desired level of accuracy, while with too many nodes it will take a long time to get trained and may sometimes over fit the data. The application of two hidden layer appear to be an advantage for a bigger and large catchment such as Sungai Ketil. It can be seen that the smaller catchment as Sungai Lui and Sungai Bekok is sufficient for a single hidden layer of neural model structure.

Obviously, the application of neural network method in modelling the relationship between rainfall and runoff for Sungai Lui, Sungai Klang, Sungai Bekok, Sungai Slim and Sungai Ketil is appropriate. The results reflect that the performance of neural network model is satisfactory and it is feasible for rainfall-runoff model in Malaysia catchment. The inaccuracy of model could be clarified by utilization of longer period of training data with many events of peak discharge.

CONCLUSIONS

The potential of artificial neural network models for prediction runoff has been presented in this paper. The non-linear nature of the relationship of rainfall-runoff processes is appropriate for the application of ANN methods. Results of ANN models reflect that the application of neural network method is feasible for model the rainfall-runoff relationship in Malaysia region. Apparently, the neural network has the ability to predict runoff accurately using the rainfall data as input variable.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to the Ministry of Science, Technology, and Environment of Malaysia for the financial support for this study. Thanks also to Department of Irrigation and Drainage (DID) Ampang for providing me the data.

REFERENCES

Tokar A. S. and Johnson P. A. (1999). "Rainfall-Runoff Modelling Using Artificial Neural Networks." *J. of Hydrologic Eng.* 4(3): 232-239.

Govindaraju R. S. (2000). "Artificial Neural Networks In Hydrology, Part I: Preliminary Concepts." *J. of Hydr. Engrg.*, 5(2): 115-123.

Buch A. M., Mazumdar H. S., and Pandey P. C. (1993). "A case study of runoff simulation of a Himalayan glacier basin." *Proc. Inter. Conf. on Neural Networks,* 971-974.

Hsu K-L, Gupta H. V. and Sorooshian S. (1993). "Artificial Neural Network Modelling of the Rainfall-Runoff Process." *Water Resources Research*. 29(4): 1185-1194.

Shamseldin A. (1997). "Application of a Neural Network Technique To Rainfall-Runoff Modelling". *J. of Hydrologic*. 199: 272-294.

Dibike Y. B. and Solomatine D. P. (1999). "River Flow Forecasting Using Artificial Neural Networks." *J. of Physics and Chemistry of the Earth*: EGS1.1-0002.

Karunithi N., Grenney W.J., Whitley D., and Bovee K. (1994). "Neural Networks for River Flow Prediction". *J. of Comp. in Civil Eng*. 8(2): 201-220.

Maier H.R. and Dandy G.C. (1996). "The Use of Artificial Neural Networks for the Prediction of Water Quality Parameters." *Water Resources Research*. 32(4): 1013-1022

Harun S. (1999). "Forecasting and Simulation of Net Inflows for Reservoir Operation and Management." Ph.D. thesis, Dept. of Hydraulics and Hydrologic, Universiti Teknologi Malaysia.

Raman H. and Chandramouli V. (1996). "Deriving a General Operating Policy for Reservoirs Using Neural Network." *J. of Water Resources Planning and Management*. 122(5): 342-347.

Ranjithan S. and Eheart J. W. (1993). "Neural Network-based Screening for Groundwater Reclamation Under Uncertainty." *Water Resources Research*. 29(3): 563-574.

Tsoukalas L. H. and Uhrig R. E. (1997). "Fuzzy and Neural Approaches in Engineering." New York: John Wiley & Sons Inc.

Caudill M. (1987). "Neural Networks Primer, Part I." *AI Expert* December: 46-52.

Fausett L. (1994). "Fundamentals of Neural Networks." New Jersey: Prentice Hall, Englewood Cliffs.

Govindaraju R. S. (2000). "Artificial Neural Networks In Hydrology, Part II: Hydrologic Applications". *J. of Hydr. Engrg.,* 5(2): 124-135.

Yapo P.O., Gupta V.H., and Sorooshian S. (1996). "Automatic Calibration of Conceptual Rainfall-Runoff Models: Sensitivity to Calibration Data". *J. of Hydrology*. 181: 23-48.

French M.N., Krajewski W.F. and Cuykendall R.R. (1992). "Rainfall Forecasting In Space and Time Using a Neural Network". *J. of Hydrology,* 137(1): 1-31.