Journal of Hydroscience and Hydraulic Engineering
Vol. 22, No.2 November, 2004, 141-154

IMPROVING NEURAL NETWORK MODEL PERFORMANCE IN RUNOFF ESTIMATION BY USING AN ANTECEDENT PRECIPITATION INDEX

By

S. Lipiwattanakarn

Kasetsart University, Bangkok, Thailand

N. Sriwongsitanon

Kasetsart University, Bangkok, Thailand

and

S. Saengsawang

Kasetsart University, Bangkok, Thailand

SYNOPSIS

Rainfall-runoff modeling using process models attempts to simulate the complex processes that rainfall affects runoff, whereas artificial neural network (ANN) modeling can simulate the non-linear relationship between rainfall and runoff without requiring any understanding of the rainfall-runoff process. In this study, an ANN model was used to estimate runoff by using a short period of rainfall data as inputs. The accuracy of runoff estimation improves significantly when the soil moisture content (represented as an antecedent precipitation index or API) is provided as an additional input. The ANN model with API simulates the peak flows and the overall runoff hydrograph more accurately than a traditional conceptual rainfall-runoff model (NAM), however the NAM model simulates the baseflows more accurately.

INTRODUCTION

Rainfall-runoff processes are non-linear complex systems involving several contributing factors such as rainfall depth, rainfall distribution, land use, soil type, soil moisture content, etc. The variety of models that have been developed and applied to simulate these processes can be classified into black-box models, conceptual models, and physically-based models. Normally, conceptual models and physically-based models are based on numerical representations of the complex processes affecting rainfall-runoff and theoretically these models should be more accurate. However, they do require large amounts of observational data, and are time consuming and difficult to calibrate. Due to process and model complexity, these models are often fitted without serious consideration of parameter values, resulting in poor performance during verification (8). Another problem with both conceptual and physically-based models is that empirical regularities or periodicities are not always evident and can often be masked by noise (18). Black-box models using artificial neural networks (ANNs) have been proposed as a feasible alternative approach because they are more flexible and can capture the non-linearity in rainfall-runoff processes ((9), (16)).

Modeling and forecasting water resources variables, including rainfall-runoff processes by ANNs, have been mainly performed by multi-layer feedforward networks with a back propagation algorithm developed by Rumelhart *et al.* ((15), (11)). An ANN was used to systematically formulate the rainfall-runoff process by Dawson and Wilby (4). Usually, ANN models consist of three layers - an input layer, a hidden layer, and an output layer. In runoff estimation, the input layer is usually composed of nodes that indicate information that influences runoff occurrence, such as rainfall and climatic data. However, short period rainfall data alone was insufficient to estimate runoff satisfactorily ((12), (14)). Several researchers have introduced other variables to improve runoff estimation. These variables include rainfall index (17), historical discharge ((9), (12)) and the observed soil moisture (8). A study using observations of soil moisture together with historical rainfall data showed satisfactory runoff estimation. However, in many cases, observations of soil moisture are either limited or unavailable.

Historical discharge has been used as the sole input for flood forecasting, with promising results ((2), (18)). However, a runoff estimation model is normally a cause-and-effect model, so historical discharge records should not be used as inputs for runoff estimation. In particular, this type of model cannot be applied when historical discharge data are unavailable.

In this study, an ANN model was used to estimate runoff using historical rainfall data, evaporation and a representation of the catchment soil moisture content (the antecedent precipitation index (API)) as inputs. A conceptual rainfall-runoff model (the NAM model) was also tested by using the same input data, and the results from the two models were compared.

ARTIFICIAL NEURAL NETWORKS

ANNs are mathematical models with a highly connected structure inspired by the structure of the brain and nervous systems. ANN processes operate in parallel, which differentiates them from conventional computational methods. ANNs consist of multiple layers - an input layer, an output layer and one or more hidden layers - as shown in Fig. 1. Each layer consists of a number of nodes or neurons which are inter-connected by sets of correlation weights. The input nodes receive input information that is processed through a non-linear transfer function to produce outputs to nodes in the next layer. These processes are carried out in a forward manner hence the term multi-layer feed-forward model is used. A learning or training process uses a supervised learning algorithm that compares the model output to the target output and then adjusts the weight of the connections in a backward manner. The process can be summarized in mathematical form as follows.

$$\mathrm{net}_j = \sum_{i=0}^{N} W_{ij} X_i \tag{1}$$

where X_o and W_{oj} are the bias ($X_o = 1$) and its bias weight, respectively. N represents the number of input nodes. Each hidden node input (net_j) is then transformed through the non-linear transfer function to produce a hidden node output, Y_j. The most common form of the transfer function is a sigmoid function and is expressed as follows:

$$Y_j = f(\mathrm{net}_j) = \frac{1}{1+e^{-\mathrm{net}_j}} \tag{2}$$

Similarly, the output values between the hidden layer and the output layer are defined by

$$\mathrm{net}_k = \sum_{j=0}^{M} W_{jk} Y_j \;; \quad Z_k = f(\mathrm{net}_k) = \frac{1}{1+e^{-\mathrm{net}_k}} \tag{3}$$

where M = the number of hidden nodes; W_{jk} = the connection weight from the j-th hidden node to the k-th output node; and Z_k = the value of the k-th output node.

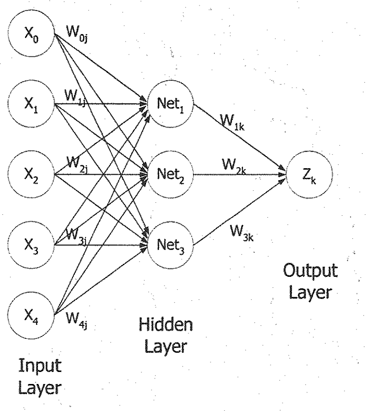


Fig. 1 The structure of ANN

A CONCEPTUAL RAINFALL-RUNOFF MODEL

The rainfall-runoff process was calibrated and verified by using the NAM model (5), which is a conceptual rainfall-runoff model. The model is a lumped type, i.e. the basin is considered as a whole. The NAM model represents various components of the rainfall-runoff process by continuously account-ing for the moisture content in four different but interrelated storages, which represent physical elements of the basin. These storages are snow storage, surface storage, lower zone storage and groundwater storage. The meteorological input data are precipitation and potential evapotranspiration and the result is catchment runoff. The resulting runoff is split conceptually into overland flow, interflow and baseflow components. More details of the NAM model can be found in DHI (5) or Madsen (10).

METHODS

Study area and data set

The study area is the Mae Ngat River basin in northern Thailand as shown in Fig. 2. The data used in this study was daily river discharge at station P.28 with the catchment area of 1,261 square kilometres and daily evaporation and rainfall data from two stations located in the basin. All data were collected over a period of six years (1973-1978).

The ANN model runs were performed by using a split-sample technique with an early stopped training approach (3). Accordingly, the data were split into three sets: a training set, a validation set, and a testing set. The training data set was three years (1974-1976). A one year validation data set from 1973 was used to stop the training to avoid underfitting or overfitting on training, and to enhance the generalization ability of the models. The testing data set (from 1977-1978) was used to verify the effectiveness of the trained model in non-trained events.

The antecedent precipitation index (API) used in this study was defined by (6)

$$API_t = (API_{t-1} + P_{t-t})e^{\alpha \Delta t} \qquad (4)$$

where API_t = an antecedent precipitation index at time t; P_{t-1} = rainfall amount at time t-1; Δt = a time step (daily basis); and α = a constant. In this study, α was taken as 0.01.

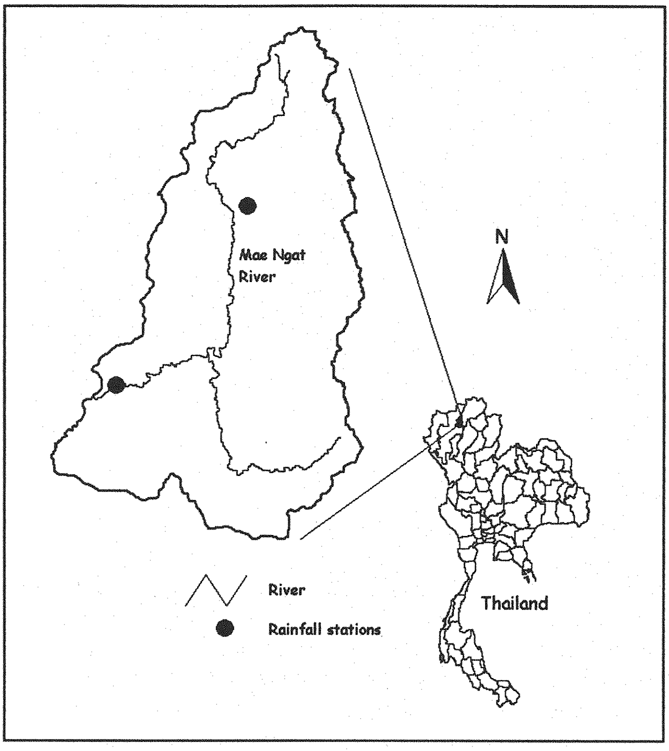


Fig. 2 Location of the Mae Ngat River basin and rainfall stations

For ANN simulations, all data were normalized in the range 0.05 and 0.95 to decrease the effect of the magnitude of the different variables, and to enable the use of a sigmoid function as a transfer function.

ANN formulation

Four different ANN models were formulated and the ability of each to represent the rainfall-runoff process was tested. The basic model (Rain model) used only a short period of historical rainfall data as input. The Rain-E model was the Rain model, with evaporation data as an additional input. The API model was the same as the Rain model, with API as an additional input. The API-E model was the same as the Rain model, with API and evaporation data as additional inputs. All models produced discharge as the output.

The determination of appropriate lags for rainfall data can be performed by a prior knowledge of the rainfall-runoff process in conjunction with inspections of correlation plots between potential inputs and outputs (11). Dolling and Varas (7) recommended that to select the adequate group of input variables, a sensitivity analysis and a multivariate analysis should be used. However, in this study, the contribution of weights from potential inputs to an output of an ANN model without a hidden layer was used to determine the appropriate lags. The relation between weights and potential inputs was determined based on the data collected during the year of 1973. The result was shown in Fig. 3. Therefore, the proposed ANN models can mathematically be written as:

$$\text{Rain model}: Q_t = f(R1_t, R1_{t-1}, R1_{t-2}, R1_{t-3}, R1_{t-4}, R1_{t-5}, R2_t, R2_{t-1}, R2_{t-2}) \qquad (5)$$

$$\text{Rain-E model}: Q_t = f(E_t, R1_t, R1_{t-1}, R1_{t-2}, R1_{t-3}, R1_{t-4}, R1_{t-5}, R2_t, R2_{t-1}, R2_{t-2}) \quad (6)$$

$$\text{API model}: Q_t = f(API_t, R1_t, R1_{t-1}, R1_{t-2}, R1_{t-3}, R1_{t-4}, R1_{t-5}, R2_t, R2_{t-1}) \quad (7)$$

$$\text{API-E model}: Q_t = f(E_t, API_t, R1_t, R1_{t-1}, R1_{t-2}, R1_{t-3}, R1_{t-4}, R1_{t-5}, R2_t, R2_{t-1}) \quad (8)$$

where Q_t = discharge at time t; $R1_t$ = rainfall data from station one at time t; $R2_t$ = rainfall data from station two at time t; E_t = evaporation data at time t; and API_t = an antecedent precipitation index at time t. The subscripts t-1, t-2, --, t-n represent the time at the previous 1, 2, --, and n days, respectively.

The Stuttgart Neural Network Simulator, SNNS (19), was selected to perform the ANN simulations. Training was based on back-propagation with a momentum algorithm. A network with only three layers was selected for all models. For each model, the initial network structure was set so that the number of hidden nodes was equal to the number of input nodes. Afterwards, the model was subjected to hidden node pruning using a skeletonization algorithm, which eliminated unwanted nodes (1). The skeletonization prunes nodes by estimating a change in the error function when a node is removed. If the change is within an acceptable limit, the node is removed. For each node, an attentional strength is introduced into the net input (Eq. 3) to form a different equation as follows:

$$net_k = \sum_{j=0}^{M} W_{jk} \alpha_j Y_j \quad (9)$$

where α_j = an attentional strength of the unit Y_j. When the unit is removed, the change in the error function can be defined as:

$$\rho_j = \frac{\partial E}{\partial \alpha_j} \quad (10)$$

where ρ_j = the change in the error function after the unit is removed and E is the linear error function.

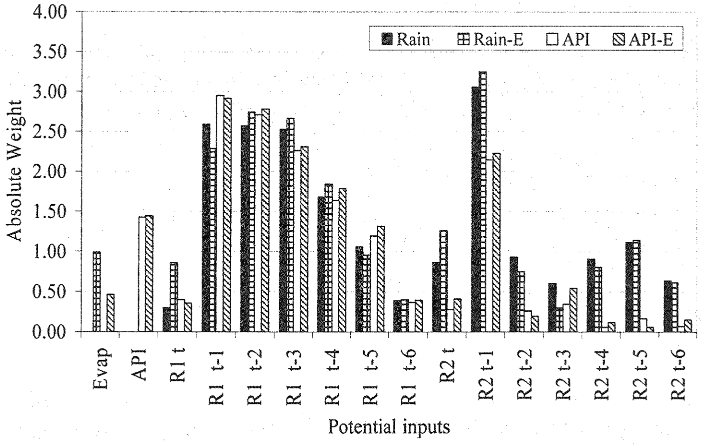


Fig. 3 Weighting distribution of potential inputs

Calibration of the NAM model

Nine parameters of the NAM model were calibrated for the Mae Ngat River basin using the same data sets as for the ANN models. The training data set (1974-1976) was used for calibration and the testing data set (1977-1978) was used for verification.

Assessment of the model performance

To assess the accuracy of a rainfall-runoff model, more than one criterion should be used. Madsen (10) recommended four criteria for successful calibration of a rainfall-runoff model. These criteria were good agreement in terms of: (1) water balance, (2) overall shape of the hydrograph, (3) peak flows, and (4) low flows. Therefore, six different goodness-of-fit measures were used to test the agreement between observed and simulated discharges. The details of each criterion are as follows:

1. Correlation coefficient (r)

$$r = \frac{\sum_{i=1}^{N} Q_{obs,i} \cdot Q_{sim,i}}{\sqrt{\sum_{i=1}^{N} Q_{obs,i}^{2} \cdot \sum_{i=1}^{N} Q_{sim,i}^{2}}} \qquad (11)$$

The correlation coefficient is described in Eq. 11, where Q_{obs} = the observed discharge; Q_{sim} = the simulated discharge; and N = the number of observations. The correlation coefficient measures how well each observed discharge value correlates with the simulated discharge. The value is between -1 and 1. The value of one means perfect correlation, whereas zero means that there is no correlation. This criterion can be used to measure the agreement between the overall shape of the observed and simulated hydrographs.

2. Root mean square error (RMSE)

$$RMSE = \left(\frac{\sum_{i=1}^{N} (Q_{obs,i} - Q_{sim,i})^2}{N} \right)^{\frac{1}{2}} \qquad (12)$$

The root mean square error as shown in Eq. 12 measures the average error between the observed and simulated discharges. The closer the RMSE value is to zero, the better the performance of the model. The RMSE can be used to measure the agreement between the observed and simulated water balance.

3. Efficiency index (EI)

$$EI = 1 - \left(\frac{\sum_{i=1}^{N} (Q_{obs,i} - Q_{sim,i})^2}{\sum_{i=1}^{N} (Q_{obs,i} - \overline{Q}_{obs})^2} \right) \qquad (13)$$

The efficiency index or Nash-Sutcliffe criterion (13) as shown in Eq. 13 is often used to measure the performance of a hydrological model. The value is in the range of [-∞, 1]. The zero value means that the model performs equal to a naive prediction; that is, a prediction using an average observed value. The value less than zero means the model performs worse than the average observed value. A value of one is a perfect fit.

4. Water balance error (WBE)

$$\mathrm{WBE} = \frac{\sum_{i=1}^{N} Q_{obs,i} - \sum_{i=1}^{N} Q_{sim,i}}{\sum_{i=1}^{N} Q_{obs,i}} \times 100 \tag{14}$$

The water balance error as described in Eq. 14 measures the agreement in water balance. The closer the WBE is to zero, the better is the simulated discharge.

5. Root mean square error of peak flows ($\mathrm{Peak}_{\mathrm{RMSE}}$)

$$\mathrm{Peak}_{\mathrm{RMSE}} = \frac{1}{P}\sum_{j=1}^{P}\left[\frac{1}{MP}\sum_{i=1}^{MP}\left(Q_{obs,i} - Q_{sim,i}\right)^2\right]_j^{1/2} \tag{15}$$

The root mean square error of peak flows is defined by Eq. 15, where P = the number of peak flow events; MP = the number of observations in those events, where the observed discharge was greater than or equal to QT; and QT = the threshold value for peak flow at 98% probability of exceedance. The threshold value for peak flow for the data set used in this study was 98.0 m^3/s. This criterion measures model performance in simulating peak flows. The closer the $\mathrm{Peak}_{\mathrm{RMSE}}$ is to zero, the better the model simulates peak flows.

6. Root mean square error of baseflows ($\mathrm{BF}_{\mathrm{RMSE}}$)

$$\mathrm{BF}_{\mathrm{RMSE}} = \frac{1}{B}\sum_{j=1}^{B}\left[\frac{1}{MB}\sum_{i=1}^{MB}\left(Q_{obs,i} - Q_{sim,i}\right)^2\right]_j^{1/2} \tag{16}$$

The root mean square error of baseflows is shown in Eq. 16, where B = the number of baseflow events; MB = the number of occurrences where the observed discharge was less than or equal to QB; and QB = the threshold value for baseflow at 20% probability of exceedance. The threshold value for baseflow for the data set used in this study was 1.86 m^3/s. This criterion measures model performance in simulating baseflows. The closer $\mathrm{BF}_{\mathrm{RMSE}}$ is to zero, the better the model simulates baseflows.

RESULTS AND DISCUSSION

The six goodness-of-fit statistics are summarized in Table 1 for the ANN models and the NAM model. The ANN model with API and evaporation data (API-E model) performs best, except for baseflow simulation. The Rain-E model performs better than the Rain model and the API-E model performs slightly better than the API model. This indicates that the inclusion of evaporation data can improve the performance of the ANN rainfall-runoff models.

Table 1 clearly shows that the performance of the API model is significantly better than the Rain model. To examine the effects in more detail, the observed and simulated discharges from both models are plotted (Fig. 4). The hydrograph from the API model shows more acceptable simulation of peak flows than the Rain model. From Table 1, the values of the root mean square error of peak flows ($\mathrm{Peak}_{\mathrm{RMSE}}$) of the API model for both training and testing periods are the lowest, which are 31.71 m^3/s for the training period and 50.56 m^3/s for the testing period. Without the antecedent precipitation index, the Rain model generated the same runoff value when there was no rainfall, resulting in non-realistic hydrograph shape (Fig. 4). This is due to the fact that without API data, when rainfall data are zero, the input nodes of the Rain model are zero. When input information does not vary, the ANN model will generate the same result.

Table 1 Comparison of the model performance

Model Criteria	Training					Testing				
	Rain model	E-Rain model	API model	API-E Model	NAM model	Rain model	E-Rain model	API model	API-E Model	MNA model
r	0.759	0.806	0.894	0.895	0.859	0.668	0.610	0.796	0.776	0.673
EI	0.52	0.62	0.67	0.75	0.71	0.43	0.36	0.57	0.58	0.41
RMSE	16.72	14.82	13.75	12.15	13.02	13.17	13.95	11.41	11.29	13.39
WBE	-26.70	-19.35	45.24	35.14	4.25	-16.44	-16.30	16.74	12.64	-27.88
Peak$_{RMSE}$	79.13	55.66	31.71	37.54	40.70	60.42	85.95	50.56	62.93	128.66
BF$_{RMSE}$	3.56	3.42	3.10	3.58	0.96	4.92	4.69	3.55	4.11	0.61

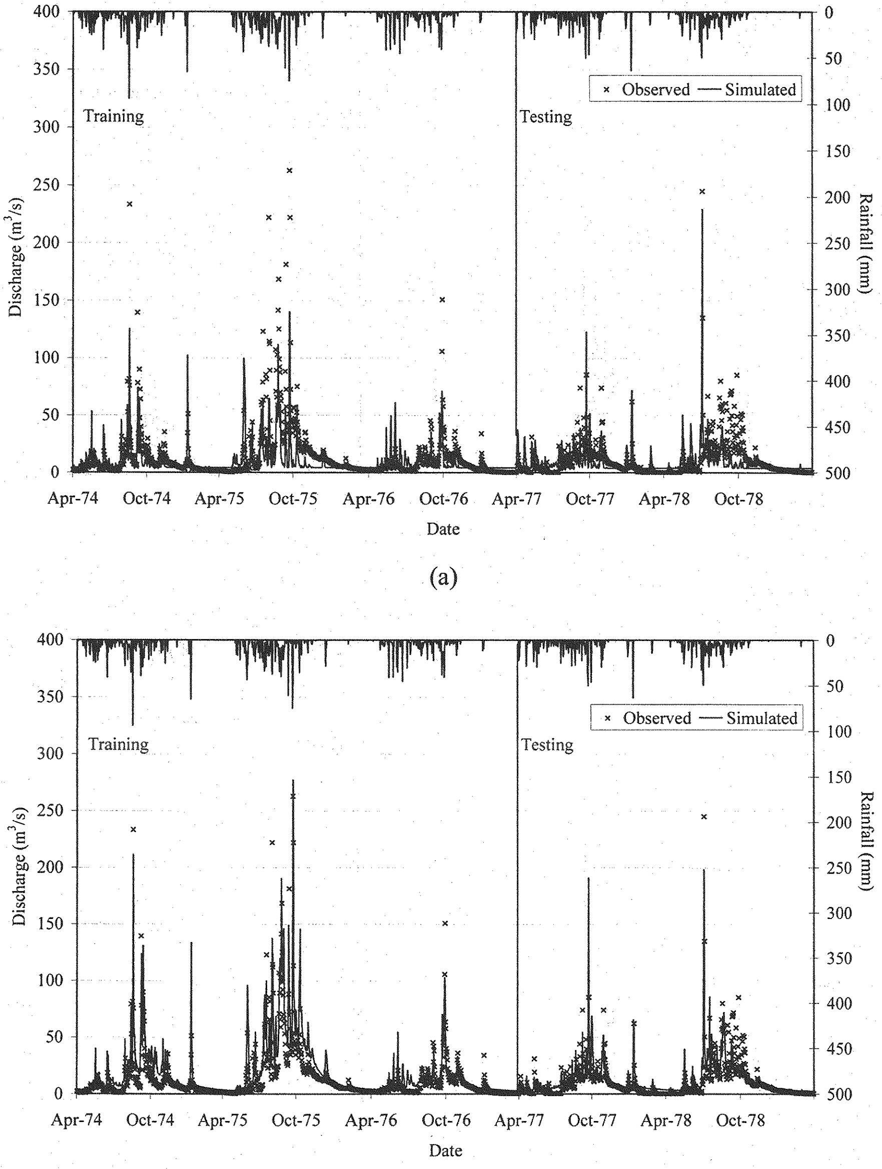
The Rain and Rain-E models underestimate basin runoff, whilst the API and API-E models overestimate basin runoff. The water balance error (WBE) values for the Rain model are -26.70% and -16.44% for training and testing, respectively, and those for the Rain-E model are -19.35% and -16.30% for training and testing, respectively. Whereas, the WBE values for the API and API-E models are 45.24% and 35.14% for training, and 16.74% and 12.64% for testing, respectively.

For further analysis of the ANN model performance, the results of the API-E model were compared with those of the conceptual rainfall-runoff model (the NAM model). The simulated discharges from both models are plotted with the observed discharge in Fig. 5. In general, the API-E model performs better than the NAM model. However, the API-E model is more effective in simulateing peak flows than baseflows. The Peak$_{RSME}$ values for the API-E model are 37.54 m^3/s for training and 62.93 m^3/s for testing, compared to 53.91 m^3/s for training and 128.66 m^3/s for testing for the NAM model (Table 1). The BF$_{RSME}$ values for the API-E model are 3.58 m^3/s for training and 4.11 m^3/s for testing, compared to 0.96 m^3/s for training and 0.61 m^3/s for testing for the NAM model. It is also interesting to note that all four ANN models show similar performance in simulating baseflows, with the BF$_{RSME}$ values in the range of 3.10-3.58 m^3/s for training and 3.55-4.92 m^3/s for testing. These results clearly confirm the finding that the rainfall-runoff neural network models are less effective in simulating baseflows than the NAM model. The results obtained in this study for the API-E model contradicts the work of Zealand *et al.* (18), who found that the ANN model performed better in simulating baseflows than peak flows. However, the results of this study agree with the work of Coulibaly *et al.* (3), who also found that the ANN model was more effective in forecasting peak flows than baseflows.

In our view, one of the reasons for this is the underlying theory of the back-propagation algorithm that minimizes the error by adjusting weights. The error function is the mean square error (MSE) between model outputs and targets. One of the advantages of using the MSE function is that it penalizes large errors. However, by doing this the model tends to adjust towards high values, in this case, peak flows.

To further investigate these effects, the testing results of the API-E model at each 100 epochs between 100-4,000 epochs were analyzed for RMSE, Peak$_{RMSE}$ and BF$_{RMSE}$. The validation results and the training results were subjected to the same analysis to show the effects of using the early stopped training approach. Fig. 6 shows plots of BF$_{RMSE}$, Peak$_{RMSE}$ and RMSE from training, testing and validation. The BF$_{RMSE}$ graph (Fig. 6 (a)) shows the same pattern in all three data sets. This indicates that the mapping function between inputs and the baseflow output of the API-E model is the same for all the data sets. This result also suggests that baseflow has little effect on early stopped training.

The Peak$_{RMSE}$ graph of training (Fig. 6 (b)) shows the same pattern as BF$_{RMSE}$. But the testing and



(a)

(b)

Fig. 4 Observed and simulated discharges from the Rain (a) and API (b) models

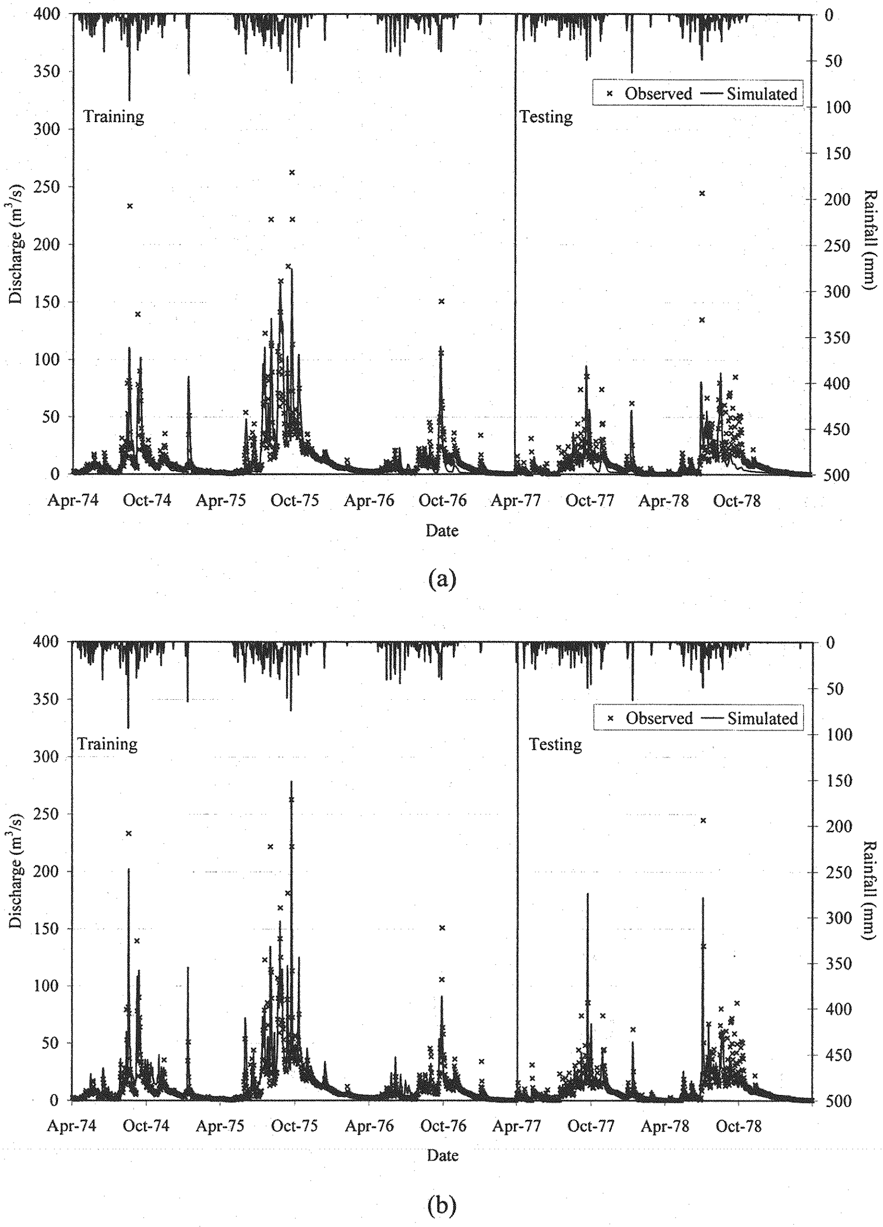
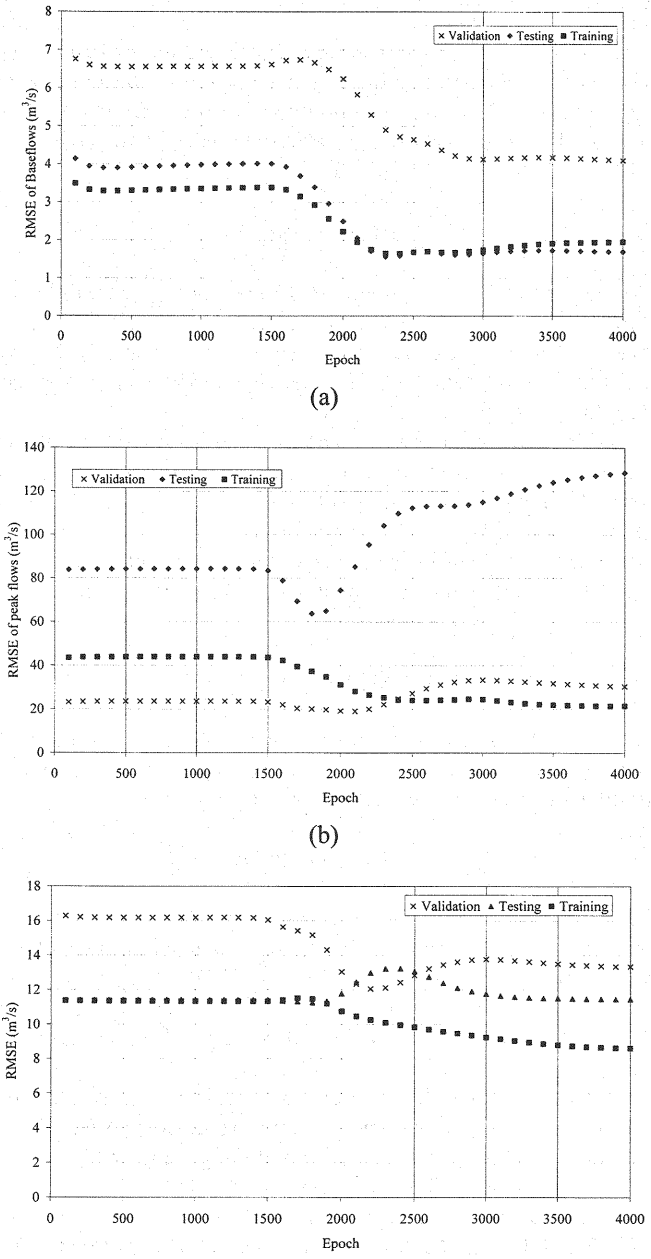


Fig. 5 Observed and simulated runoff from the NAM (a) and API-E (b) models



(a)

(b)

(c)

Fig. 6 RMSE of baseflows, peakflows and RMSE from the API-E model

validation graphs for $\mathrm{Peak}_{\mathrm{RMSE}}$ differ. They clearly show minimum points, at epoch 2,100 for validation and at epoch 1,800 for testing. This means that the mapping function between inputs and the peak flow output for the training set is different from the one for the validation and testing sets. The results also show that the API-E model can be overfitted to the peak flow training set if it is overtraining. This result implies that peak flows have strong effects when the early stopped training approach is applied. The RMSE graph (Fig. 6(c)) shows almost the same pattern as the $\mathrm{Peak}_{\mathrm{RMSE}}$ graph. This indicates that peak flows are more dominant than baseflows even though there are fewer data points, i.e. the peak flow threshold value is 98% probability of exceedance whereas the baseflow threshold value is 20% probability of exceedance.

CONCLUSION

In this study, four ANN models were trained using the early stopped training approach for rainfall-runoff modeling. The differences between the four models are the inclusion or exclusion of evaporation and antecedent precipitation index data. The results showed that the ANN models with evaporation data performed slightly better than the ANN models without evaporation data. The inclusion of API data significantly improved model performance.

The performance of the ANN models was compared with a conceptual rainfall-runoff model, the NAM model. The ANN models were more effective in simulating peak flows, whereas the conceptual model was more effective in simulating baseflows. In general, the ANN models perform better than the conceptual model.

This study provides evidence that artificial neural network modeling of rainfall-runoff can be a valid alternative to conventional modeling, especially where internal processes are not clearly understood.

ACKNOWLEDGEMENT

The authors wish to thank Dr. Peter Hawkins of Sydney Water Corporation, Australia, for his valuable suggestions. This work is financially supported by Kasetsart University Research and Development Institute under the grant number 01009489.

REFERENCES

1. Abrahart, R.J., L. See, and P.E. Kneale : Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model, Journal of Hydroinformatics, Vol.01.2, pp.103-114, 1999.
2. Atiya, A.F., S.M. El-Shoura, S.I. Shaheen, and M.S. El-Sherif : A comparison on neural-network forecasting techniques- case study: river flow forecasting, IEEE Transactions on Neural Network, Vol.10(2), pp.402-409. 1999.
3. Coulibaly, P., F. Anctil, and B. Bobee : Daily reservoir inflow forecasting using artificial neural networks with stopped training approach, Journal of Hydrology, Vol.230, pp.244-257, 2000.
4. Dawson, C.W. and R.L. Wilby : Hydrological modelling using artificial neural networks, Progress in Physical Geography, Vol.25(1), pp.80-108, 2001.
5. Danish Hydraulic Institute (DHI) : NAM Technical Reference and Model Documentation, 1999.
6. Descroix, L., J.F. Nouvelot, and M. Vauclin : Evaluation of an antecedent precipitation index to model runoff yield in the western Sierra Madre (North-west Mexico), Journal of Hydrology, Vol.263, pp.114-130, 2002.
7. Dolling, O.R. and E.A. Varas : Artificial neural network for streamflow prediction, Journal of Hydraulic Research, Vol.40(5), pp.547-554, 2002.
8. Gautam, M.R., K. Watanake, and H. Saegusa : Runoff analysis in humid forest catchment with artificial neural network, Journal of Hydrology, Vol.235, pp.117-136, 2000.

9. Hsu, K., H.V. Gupta, and S. Sorooshian : Artificial neural network modeling of the rainfall-runoff process, Water Resources Research, Vol.31(10), pp.2517-2530, 1995.
10. Madsen, H. : Automatic calibration of a conceptual rainfall-runoff model using multiple objectives, Journal of Hydrology, Vol.235, pp.276-288, 2000.
11. Maier, H.R. and G.C. Dandy : Neural networks for prediction and forecasting of water resources variables : a review of modeling issues and applications, Environmental Modeling & Software, Vol.15, pp.101-124, 2000.
12. Minns, A.W. and J. Hall : Artificial neural networks as rainfall-runoff models, Hydrological Sciences Journal, Vol.41(3), pp.399-417, 1996.
13. Nash, J.E. and J.V. Sutcliffe : River flow forecasting through conceptual models, part 1–A discussion of principles, Journal of Hydrology, Vol.10, pp.282-290, 1970.
14. Randall, W.A. and G.A. Tagliarini : Using feed forward neural networks to model the effect of precipitation on the water levels of the Northeast Cape Fear river, Proceedings of IEEE Southeast Conference, pp.338 –342, 2002.
15. Rumelhart, D.E., G.E. Hinton, and R.J. Williams : Learning internal representations by error propagation, Parallel Distributed Processing: Explorations in the microstructures of Cognition, Rumelhart, D.E. and J.L. Mc.Clelland, eds, Vol.1, pp.318-362, the MIT Press, 1986.
16. Sajikumar, N. and B.S. Thandaveswara : A non-linear rainfall-runoff model using an artificial neural network, Journal of Hydrology, Vol.216, pp.31-55, 1999.
17. Shamseldin, A.Y. : Application of a neural network technique to rainfall-runoff modeling, Journal of Hydrology, Vol.199, pp.272-294, 1997.
18. Zealand, C.M., D.H. Burn, and S.P. Simonovic : Short term streamflow forecasting using artificial neural networks, Journal of Hydrology, Vol.214, pp.32-48, 1999.
19. Zell A., et. al. : SNNS Stuttgart Neural Network Simulator User Manual, Version 4.2, University of Stuttgart and University of Tubingen, 2000.

APPENDIX – NOTATION

The following symbols are used in this paper:

API_t	= an antecedent precipitation index at time t;
B	= number of baseflow events;
BF_{RMSE}	= root mean square error of baseflows;
E	= the linear error function;
EI	= efficiency index;
MB	= number of observations in baseflow events;
MP	= number of observations in peak flow events;
net_j, net_k	= hidden node and output node input;
N	= the number of discharge observations;
P	= number of peak flow events;
P_{t-1}	= rainfall amount at time t-1;
$Peak_{RMSE}$	= root mean square error of peak flows;
$Q_{obs,i}$	= observed discharge;
$Q_{sim,i}$	= simulated discharge;

Q_t	= discharge at time t;
QB	= threshold value for baseflow at 20% probability of exceedance;
QT	= threshold value for peak flow at 98% probability of exceedance;
r	= correlation coefficient;
$R1_t$, $R2_t$	= rainfall data from station one and station two at time t;
RMSE	= root mean square error;
t	= time;
W_{ij}	= connection weight from the i-th node to the j-th node;
WBE	= water balance error;
X_i	= the i-th input node;
Y_j	= the j-th hidden node output;
Z_k	= the k-th output node output;
α	= a constant;
α_j	= an attentional strength of the j-th hidden unit;
Δt	= a time step;
ρ_j	= the change in the error function after the hidden unit is removed.

(Received March 22, 2004 ; revised April 19, 2004)