Geophysical Research Abstracts, Vol. 7, 03180, 2005
SRef-ID: 1607-7962/gra/EGU05-A-03180

Identification of trends in training of ANN rainfall runoff models for improved training performance

S. K. Jha and Ashu Jain

Department of Civil Engineering, Indian Institute of Technology Kanpur - 208016, India
(ashujain@iitk.ac.in / Fax: 91+512+259 7395)

Introduction: The use of data-driven methods such as artificial neural networks (ANNs) for modeling hydrological processes and flood forecasting has increased tremendously over the last couple of decades. Although there is abundance of research works reported in the area of application of ANN models in wide variety of disciplines including hydrology, there are no specific guidelines for carrying out various steps involved in developing an ANN model. The training of ANN models has remained a brute force method, which is computationally inefficient. The training of an ANN model normally involves thousands of iterations until acceptable level of error is achieved, which causes significant computational burden on the training algorithm. It may be possible to lessen such computational burden by identifying certain trends during training and exploiting them to advantage. This paper makes an effort in such a direction. The daily rainfall and flow data derived from the Kentucky River Basin for a period of 26 years have been employed to develop ANN models and identify trends during training. The drainage area of the basin considered is approximately 10,240 km2. The data were divided into two sets: a training data set consisting of data for thirteen years (1960-1972), and a testing data set of thirteen years (1977-1989).

ANN Model Development: The ANN model developed in this study consisted of three layers: an input layer, a hidden layer, and an output layer. The input vector consists of five neurons representing the total rainfall at times t, t-1, and t-2 {P(t) P(t-1), and P(t-2)} and the observed discharges at times t-1 and t-2 {Q(t-1) and Q(t-2)}. The significant input variables were determined using auto-, partial-, and cross-correlation analysis. The only neuron in the output layer represented the flow at time t, Q(t), being modeled. The sigmoid activation function was used as the transfer function at both hidden and output layers. Standard back-propagation training algorithm with generalized delta rule (Rumelhart et al., 1986) was employed to train all ANN architectures investigated. The performance of the ANN models developed was evaluated using four error statistics, namely, average absolute relative error (AARE), correlation coefficient (R), coefficient of efficiency (E), and normalized root mean square error (NRMSE) (see Jain and Srinivasulu, 2004 for details). The values of AARE, R, E, and NRMSE for the best ANN model were found to be 9.7%, 0.9602, 0.9127, and 0.1538 during testing data set. These statistics indicate an excellent generalization capability of the developed ANN model.

The statistical results in terms of the above statistics were further analyzed to identify any trends in the training process. It has been observed that at certain number of iteration, there is sudden improvement in the performance of the ANN model in terms of all the statistics. This may be called the critical plateau, a critical plateau is one in which the training performance of an ANN model suddenly jumps. Further, the critical plateau is normally located close to the acceptable level of error. The objective in ANN training, therefore, should be on the identification of the critical plateau rather than continuing the training for several thousand iterations.

The critical plateau can be determined using a graphical analysis. In the present study, the critical plateau was observed in the ANN model performance during iteration numbers 4900 and 5000 in terms of all the error statistics. No significant improvement was observed in the training of the ANN model before and after this critical plateau. Many of the ANN applications exhibit such a behaviour during training. The training of the developed ANN model was further explored to examine the sensitivity of the critical plateau to the training parameters. It was found that the location of the critical plateau shifted from approximately 22,000 iterations to approximately 200 iterations when the value of learning rate is decreased from 0.012 to 0.005.

Conclusions: This paper presents the findings of a study aimed at identifying certain trends during training of the ANN rainfall-runoff models. It has been found that there exists a critical plateau close to the acceptable level of error during the training of ANN rainfall-runoff models. Since the location of the critical plateau is sensitive to the training parameters, they can be adjusted to improve the computational efficiency of the training performance. The findings reported in this study can be vital from the view point of improving the computational efficiency during the training of ANN models. Instead of training an ANN model for the specified number of maximum iterations (normally several thousands), the emphasis should be on reaching the critical plateau or a similar trend in training faster either by fine tuning the training parameters or some other methods. Such an approach can be computationally more efficient as compared to the usual brute force method normally employed for the training of ANN models. Further, since each problem is unique, the trends in training may also be unique in nature that need to be identified and exploited for developing computationally efficient methods for training ANN models. It is hoped that future research efforts will focus in these directions for the improvement of the ANN models for flood forecasting and modeling of the hydrological processes.

REFERENCES

Jain, A. and Srinivasulu, S. (2004), “Development of Effective and Efficient Rainfall-Runoff Models using Integration of Deterministic, Real-Coded Genetic Algorithms, and Artificial Neural Network Techniques”, Wat. Resour. Res., 40(4), W04302, doi:10.1029/2003WR002355.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986), Learning Internal Representations by Error Back Propagation, in D.E. Rumelhart and J.L. McClelland (eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1, Foundations, The MIT Press, Ch 8.