



Discussion

Reply to “Comment on ‘Comparison of static-feedforward and dynamic feedback neural networks for rainfall-runoff modeling’ by Y.M. Chiang, L.C. Chang, and F.J. Chang, 2004. Journal of Hydrology 290, 297–311”

Fi-John Chang^{a,*}, Li-Chiu Chang^b, Yen-Ming Chiang^a

^aDepartment of Bioenvironmental Systems Engineering and Hydrotech Research Institute, National Taiwan University, Taipei, Taiwan, ROC

^bDepartment of Water Resources and Environmental Engineering, Tamkang University, Taipei, Taiwan, ROC

Received 20 January 2005; accepted 24 March 2005

The authors would like to thank the Journal of Hydrology for providing the opportunity of discussing research, and also extend our appreciation for Jain (2005) spending his time to address a discussion of our paper (Chiang et al., 2004). Unfortunately, he seems to overlook the general theme of the paper and to pay attention to some trivial issues. The writers clarify the main aim of our paper and the concept of application of ANNs, firstly. A brief summary of the discussor's comments follow, and then our response is given.

In our study, we aim to provide a systematic comparison for two basic types of neural network—static and dynamic algorithms—in modeling rainfall-runoff processes based on the available input-output patterns and the qualitative and quantitative aspects of data sets. Without doubt, ANNs are certainly able to provide good results if the data sets are designed properly on condition that the measured data have enough available data to describe the whole system,

but they don't possess generalization ability when trained by partial cases drawn from system. In our study, the four different designed cases were used to test the two compared neural networks to learn which model could get better generalization capability and stability in different situations.

1. Brief summary of the discussor's comments

The writers will only focus on appropriate issues and briefly summarize some points of the discussor's comments: (1) It can be confusing that the role of the validation set is to prevent over-training or under-training by stopping the training at a certain number of epochs. (2) There are studies that demonstrate that the SNNs can produce satisfactory results if developed properly even with very limited data. (3) Using spatially averaged rainfall in the input vector can offer the advantage of having simpler ANN structure due to fewer neurons in the input vector. (4) Correlation functions (e.g. ACF, PACF, CCF) can be used to decide on the number of significant explanatory variables in the input layer as guidelines for developing

* Corresponding author. Tel.: +886 2 23639461; fax: +886 2 23635854.

E-mail address: changfj@ntu.edu.tw (F.-J. Chang).

ANNs. (5) The scatter plots presented in Fig. 10 are for training, validation, or testing phase is not clear. (6) The authors have carried out a limited evaluation of the developed ANN models using only two statistical performance evaluation criteria. (7) The discussor does not agree with the authors' statement that since model 3 produced the best results, the average lag time is no more than three hours. The time of concentration is a deterministic characteristic of the watershed. (8) The procedures of developing RTRL models are not clearly explained and RTRL networks will not be useful in the water resources management activities since it would require a time-consuming training step to be carried out at each time step.

2. The writers' response

2.1. Issues 1 and 2

Issues 1 and 2 are fundamental concepts of building an applicable and reliable ANN for forecasting, and we are unclear why the discussor has questioned our approach. In our paper, we have clearly explained the individual significance of the three data sets (training, validation and testing data) for constructing the architecture and evaluating the performance of ANNs. The task of validation data is to avoid an "over-designed" architecture that could indeed lead to over-fitting. As stated in Ham and Kostanic's (2001) text book (page 112), the training data are split into two partitions (training and validation data); the first partition is used to update the weights in the network, and the second partition is used to validate the training performance and decide the number of neurons. The test data are then used to assess how well the network has generalized. We believe this is the best way to avoid the over-fitting or under-fitting problem.

"The discussor claimed his studies demonstrate the SNNs can produce satisfactory results if developed properly even with very limited data. In his study, the ANN model with more than two hundred connected weights was trained by using 208 data and tested by only 37 data. The study was not following the common procedure of building an ANN model, which usually must have suitable (enough) data and divide them into three independent data sets (as we mentioned early), and was criticized by Minocha

(2004), who stated such a poorly trained ANN model is over-fitted, and the results of such model are not dependable at the validation stage. Owing to the fact that his process as well as conclusion could be misleading, we refute his comment."

2.2. Issue 3

This point may be suitable for the considerable number of rain gauges or the equal concentration time of rain gauges, but in the study area, only four rain gauges are available and the rainstorms are usually very local.

2.3. Issue 4

The writers agree that using ACF, PACF and CCF can identify what variables have a linearly significant effect and should be chosen as input. We can't ignore the linear effect of variables upon the output; however, the nonlinear effects of other variables are absolutely trimmed off from the linear analysis. Hence, the trial-and-error procedure is a compromise to determine the number of variables with linear or nonlinear influence on the output.

2.4. Issue 5 and 6

Owing to the continually updated and learning characteristics of the RTRL network, its operation could be treated as a whole procedure for each case, which is different from the CG network that divides available data into three phases. Tables 5 and 6 and Fig. 9 of our paper demonstrate a well-trained network (CG), which has very small RMSE (79 and 70 in Tables 5 and 7, respectively); unfortunately, it could produce a much larger RMSE in the test set (i.e. 225 and 249 in Tables 5 and 7, respectively). These results suggest that a suitable training set, which includes enough length and diversity of data, could be the most important matter to build up a useful forecasting model. The comparative results between CG and RTRL methods were made by using four statistics, that is, \hat{Q}_p , MAE, RMAE, and RMSE.

2.5. Issue 7

The discussor misquotes our statement and overlooks the main feature of the ANNs, which have the ability to learn from their environment and adapt to it in an interactive manner similar to their biological counterparts. As stated in our paper, the study watershed is small (less than 1000 km²) and the basin slope is quite steep; consequently, the time of concentration is short (less than 3 h). To determine an appropriate network structure for forecasting the stream flow at time $t+1$, we develop four different models (time delay patterns, i.e. t to $t-3$ for models 1–4). The results show that Model 3 produces the best performance. This result “indirectly provides evidence” that the average lag time is no more than 3 h.

2.6. Issue 8

Regarding the number of hidden neurons used in the RTRL, we stated (on p. 305 of our paper) that it was determined by using trial-and-error. A brief explanation of the RTRL algorithm was also given in our paper, and detailed information can be found in our previous papers (Chang et al., 2002; 2004). The discussor apparently misses our point. The main objective of our paper was to investigate the applicability and stability of the static and dynamic ANNs under different lengths of training data. This is precisely what was done. Our results clearly indicate that the RTRL method has better generalization capability and stability to produce more accurate performance than the traditional batch learning algorithms used in SNNs under limited data (case 3 and case 4). Furthermore, we protest against the charge of “RTRL networks will not be useful in the water resources operation and management activities due to time-consuming training process and slightly better accuracy”. Owing to the fact that the input dimension of the RTRL is usually relatively small, the update training process can be done instantaneously and does not exist “time-consuming” problem. Moreover, it has been increasingly noticed that the RTRL neural network became more popular when dealing with signals that have rich and unknown dynamical characteristics (e.g. Hussein et al., 2004; Leclercq et al., 2005; Schiller and Steil, 2005), and we all know that hydrological

processes and water resources management are complex with notoriously nonlinear and dynamical characteristics.

3. Conclusion

RTRL belongs to one of the simple recurrent networks, which became more popular when dealing with dynamical systems such as hydrological processes. Although its adjustments are made to the weights of a fully connected network in real time, the input dimension of RTRL is usually relatively small and the update training process is simple and efficient. Our results do indicate that the RTRL method has better generalization capability and stability to produce more accurate performance than the traditional batch learning algorithms used in SNNs under limited data. The writers are grateful to the discussor for enhancing the understanding of our work.

References

- Chang, F.J., Chang, L.C., Huang, H.L., 2002. Real-time recurrent neural network for stream-flow forecasting. *Hydrological Processes* 16, 2577–2588.
- Chang, L.C., Chang, F.J., Chiang, Y.M., 2004. A two-step ahead recurrent neural network for streamflow forecasting. *Hydrological Processes* 18, 81–92.
- Chiang, Y.M., Chang, L.C., Chang, F.J., 2004. Comparison of static feedforward and dynamic feedback neural networks for rainfall-runoff modeling. *Journal of Hydrology* 290, 297–311.
- Ham, F.M., Kostanic, I., 2001. *Principles of Neurocomputing for Science and Engineering*. McGraw-Hill, New York.
- Hussein, A., Hirasawa, K., Hu, J., 2004. A robust control method for a PV-supplied moto using universal learning networks. *Solar Energy* 76 (6), 771–780.
- Jain, A., 2005. Comment on “Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling”. *Journal of Hydrology* 2005, 2005.
- Leclercq, E., Druaux, F., Lefebvre, D., Zerkaoui, S., 2005. Autonomous learning algorithm for fully connected recurrent networks. *Neurocomputing* 63, 25–44.
- Minocha, V.K., 2004. Discussion of “Comparative analysis of event-based rainfall-runoff modeling technique-deterministic, statistical, and artificial neural networks” by Ashu Jain, S.K.V. Prasad Indurthy. *Journal of Hydrologic Engineering* 9, 550–551.
- Schiller, U.D., Steil, J.J., 2005. Analyzing the weight dynamics of recurrent learning algorithms. *Neurocomputing* 63, 5–23.