Study on Dynamic Recurrent Neural Networks and Its Application to Agricultural Water Resources

Tsung-yi Pan and Ru-yih Wang***

***Department of Bioenvironmental Systems Engineering. National Taiwan University*
No.1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106
** Corresponding author, Professor Ru-yih Wang, E-mail: wangry@ccms.ntu.edu.tw*

ABSTRACT

In this study, a dynamic recurrent neural network was calibrated by the rainfall-runoff records of selected typhoon events. The calibrating process of a dynamic recurrent neural network includes two steps: firstly, the structure of a dynamic recurrent neural network was calibrated according to the selected typhoon records and determined the initial weights of a dynamic recurrent neural network; secondly, according to the results of former step, the initial weights of an artificial neural network was determined. Finally, more rainfall-runoff records of typhoon events, except training sets, were utilized to validate the network, and a further discussion about the transition of the unit hydrograph of a watershed was performed. The analytical approaches adopted and results achieved in this study can be applied in the planning of agricultural water resources.

Keywords: State space, Recursion, Artificial neural networks.

1. INTRODUCTION

Flood walls, water pump stations, flood gates, and other engineering structures are often built along the shores of Taiwan's rivers to protect vast amount of farmland. Each year, when flooding occurs as a result of typhoons, flood gates must be able to shut prior to the peak flooding time to allow people and machinery to be safely relocated on to shore. The modeling of the rainfall-runoff process to predict the possible time of flooding has become one of the agricultural sector's main research topics in terms of water management. Typically, an artificial neural network is used to model and predict rainfall-runoff process, but such network does not provide for a mechanism which calculates the amount of current rainfall that will translate into potential runoff. We have replaced traditional static neural networks with dynamic recurrent neural networks for this study to better understand the working mechanics behind hydrological systems, and hoping to increase the accuracy in modeling the rainfall-runoff process.

The rainfall-runoff process can be viewed as a cause and effect relationship based on time in the hydrological cycle system. This relationship can be separated into a physical model and a black-box model. Data produced from the physical model is based upon the physics of the rainfall-runoff process. Thus, it requires complete control of the physical mechanics of the rainfall-runoff process in order to model the relationship. In fact, the physical mechanism of rainfall-runoff processes is caused by hydrometeorological and geomorphological factors. Generally, physical model can answer questions about the watershed before even physically altering it (Chow *et al.*, 1964; Chow *et al.*, 1988). However, black-box models ignore the complex physical mechanism and directly determine the optimum parameters of black-box models from the input/output relation of a rainfall-runoff system. Applying neural networks to describe rainfall-runoff processes is one of the manners based on black-box models.

Research regarding neural networks within hydrological field is very vast. But a vast majority of the research only focuses on static neural network's ability to memorize, sort, and understand the data collected. The majority of the research overlooks hydrological cycle as a dynamic system. The rainfall-runoff process is a small process in the entire hydrological system. Thus, compared to the traditional static neural network, the neural network allows has the ability to model the rainfall-runoff process because of its dynamic recursive equation that provides the dynamic-system character. This is a much better mechanism for operating a realistic system. Not only does one need to know the relationship between inputs and outputs, one must also understand the changes within the system to have a grasp of the entire system. Therefore, dynamic recurrent neural network uses it unique calculation model to not only provides output data, but also, the changing process of the rainfall-runoff process.

2. BASIC THEORIES

2.1 Dynamic Recurrent Neural Networks

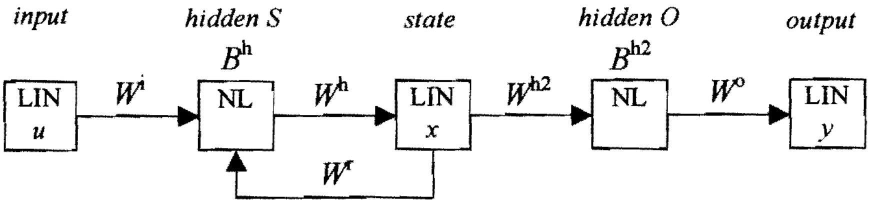
Dynamic recurrent neural networks are neural networks with one or more feedback loops which can be of a local or global type. The architecture of a dynamic recurrent neural network consists of six components: neurons, activation functions, layers, connections, architectures, and a clock. A neuron is a standalone unit that performs a static map between the input and output through an activation function that can be linear or nonlinear function. Remember that the dynamics of the network comes from the connections among neurons, not the neurons themselves (Haykin and Simon, 1999). A collection of neurons arranged conveniently in a one dimensional array is called a layer. Generally, a dynamic recurrent neural network includes an input layer, an output layer, and hidden layers. The greater the number of hidden layers embedded in a dynamic recurrent neural network, the greater the complexity of the model. No matter if they are in the same layer or not, the transference of messages between two neurons is performed through a connection that would be a local (in the same layer) or global connection (in the different layer). Additionally, an arrangement of neurons interconnected by connections in space is named architecture. In view of time domain, there is a global clock controlling the operation of all the neurons in the architecture. At the tick of the clock pulse, all neurons will perform the designated computation. These computations will be completed by the end of the clock pulse.

For the purpose of investigating the transition of rainfall-runoff systems, the dynamic recurrent neural network adopted here is embedded in a state space form. The dynamic recurrent neural network in state space form comprises two main advantages (Zamarreño, 2000):

Being a neural model, it has the flexibility to learn and represent any function.

Being a state space model, the number of outside connections is minimal. The inputs will be the causes that drive the system operation and the outputs will be the effects observed on the system. Parallel inputs/outputs (causes/effects) are established between the neural model and the physical system.

The dynamic recurrent neural network is composed of five layers: input layer, hidden layer S, state layer, hidden layer O, and output layer. The input layer takes the input signals and delivers these inputs to every neuron in the next layer, hidden layer S, which represents any function that specifies the states' behavior. State layer receives the signals from hidden layer S, and each neuron in this layer represents one state whose output value is the value of the state. After hidden layer O, which represents the features that relates the outputs of the neural network to the states, gets the signals from state layer, output layer takes the hidden layer O signals adds them to each output neuron. These outputs are, finally, the outputs of the dynamic recurrent neural network in state space form as Fig. 1.



LIN: Linear Processing Elements
NL: Non-linear Processing Elements (Sigmoid)

Fig. 1. Block scheme of the dynamic recurrent neural network in state space form. Each block represents a layer of neurons.

The mathematical representation of a deterministic system in state space form is

$$x_{k+1} = F(x_k, u_k), \tag{1}$$

$$y_k = G(x_k), \tag{2}$$

where u_k, y_k, and x_k with m, l, and n ranks denote, respectively, the input, output and state vectors at time k. $F: n\times m \rightarrow n$ and $G: n \rightarrow l$ are two static mappings.

Every linear/non-linear function can be represented by a neural network containing a single hidden layer composed of neurons whose transfer function is bounded (Zamarreño, 2000). Therefore, the mathematical form of the dynamic recurrent neural network embedded state space form can be written as

$$\hat{x}_{k+1} = W^h \cdot f_1\left(W^r \cdot x_k + W^i \cdot u_k + B^h\right), \tag{3}$$

$$\hat{y}_k = W^o \cdot f_2\left(W^{h2} \cdot x_k + B^{h2}\right), \tag{4}$$

where W^h, W^r, W^i, W^o, and W^{h2} are matrices with dimensions $n\times h$, $h\times n$, $h\times m$, $m\times h2$, and $h2\times n$ as the weights of the dynamic recurrent neural network in state space form, respectively. B^h and B^{h2} are two vectors with h and $h2$ elements as biases. f_1 and f_2 are two functions (linear/nonlinear) for describing the behavior of the system.

2.2 Realization of a linear dynamic system

Consider the linear, discrete, time invariant, multivariable system with state space representation

$$x_{k+1} = Ax_k + Bu_k, \tag{5}$$

$$y_k = Cx_k + Du_k, \tag{6}$$

where u_t, y_t, and x_t with m, l, and n ranks denote, respectively, the input, output, and state vectors at time t. In addition, *A*, *B*, *C*, and *D* with $n\times n$, $n\times m$, $l\times n$, and $l\times m$ ranks are system matrices. The state vectors are composed of a set of state variables whose values are supposed to contain sufficient information to predict the future evolution of the system (András and Szöllösi-nagy, 1982).

A dynamic system is a system whose state varies with time. Therefore, the motion of a system can be described as the motion of a point in an *n*-dimensional state space. In other words, a dynamic system described as *n*-order differential equations must have *n*-independent variables, and we can adequately represent the motion of the system as soon as the time response of the *n*-independent variables is carried out.

Through replacing the x_k term in observed equation (6) with solving the state equation (5) recursively, the output response of the system is given by

$$y_k = CA^k x_0 + \sum_{p=1}^{k} CA^{p-1} Bu_{k-p} + Du_k, \tag{7}$$

and for systems initially at rest, i.e., $x_0 = 0$, equation (7) is rewritten as

$$y_k = \sum_{p=0}^{k} h_p u_{k-p}, \tag{8}$$

where the Markov parameters are given by

$$\begin{aligned} h_k &= CA^{k-1}B \quad & k \geq 1 \\ h_k &= D & k = 1 \end{aligned} \tag{9}$$

The equation (9) shows the relationship between the system responses of a unit impulse and system matrices that will be used to present the transition of a unit hydrograph (Cooper and Wood, 1982).

3. EXAMPLE

The upstream watershed of Wu-Tu was chosen as the study area for evaluating the rainfall-runoff simulating ability of the three models. The watershed surrounds the city of Taipei in the north of Taiwan, as illustrated in Fig. 2. The area of the Wu-Tu watershed is about 204 km^2. The mean annual precipitation and runoff depth are 2,865 mm and 2,177 mm, respectively. Owing to the rugged topography of the watershed, the runoff path-lines are short and steep, and the rainfall is non-uniform in both time and space. Large floods arrive rapidly in the middle-to-downstream reaches of the watershed, causing serious damage.

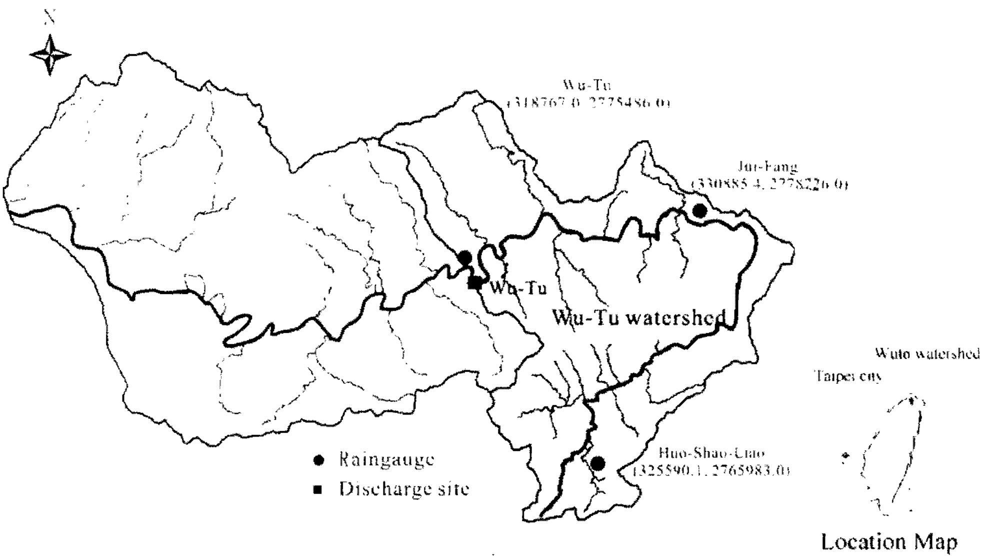


Fig. 2. The maps of Wu-Tu watershed showing the study area near Taipei, Taiwan.

The Wu-Tu watershed contains three rain gauges (Jui-Fang, Wu-Tu and Huo-Shao-Liao) and one discharge site (Wu-Tu), and the 41 rainfall-runoff events with 2,944 data recorded between 1966 and 1997 were used herein as the study sample.

In these 41 rainfall-runoff events, there were 32 typhoons and 9 storms while all events can also be classed as 14 multi-peak and 27 single-peak events. The models we studied are fed average rainfall (mm) and runoff (m^3/s) hourly as the input and the output. The average rainfall is calculated from the rainfall of three rain gauges via Kriging method (Cheng, 2002).

The study procedure can be separated into two parts: firstly, the indirect system identification is utilized to calibrate the parameters of a state space model (Wang and Pan, 2001), and a dynamic recurrent neural network is constructed based on the state space model;

secondly, the dynamic recurrent neural network is applied to simulate rainfall-runoff processes, and a new learning method developed from the interchange of the roles of the network states and the weight matrix is applied to train the neural networks (Amir and Alexander, 2000). The flowchart is shown in Fig. 3. Furthermore, for the purpose of investigating the applicability of the dynamic recurrent neural network, three kinds of models were adopted here to simulate rainfall-runoff processes. There are a state space model, a dynamic recurrent neural network whose initial weights are generated randomly (denoted as DRNN(1)), and a dynamic recurrent neural network whose initial weights are determined via indirect system identification (denoted as DRNN(2)), respectively.

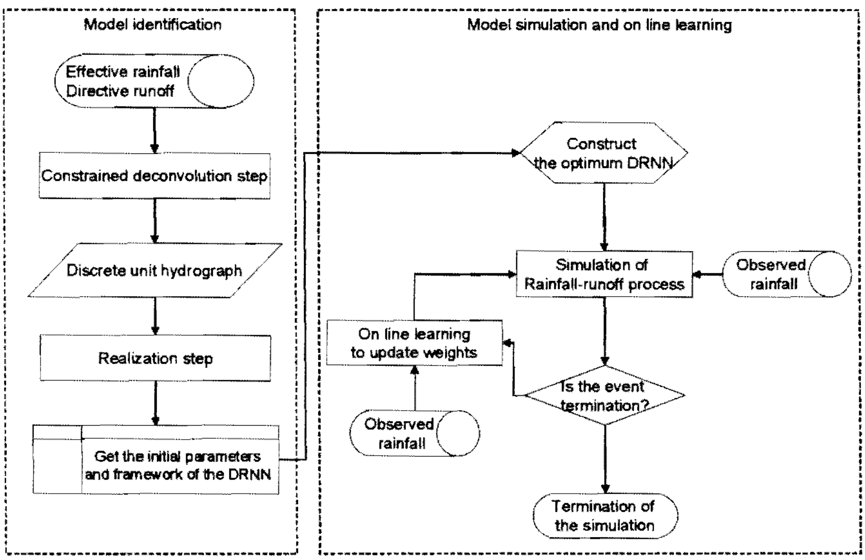


Fig.3. The flowchart of the generation and application of a DRNN(2).

The performances of rainfall-runoff simulations of the three models were evaluated via four criteria as follows:

(a) coefficient of efficiency, CE, is defined as

$$CE = 1 - \frac{\sum_{k=1}^{K} \left[Q_{est,k} - Q_{obs,k} \right]^2}{\sum_{k=1}^{K} \left[Q_{est,k} - \overline{Q}_{obs} \right]^2}, \tag{10}$$

where $Q_{est,k}$ denotes the discharge of the simulated hydrograph for time index k (m^3/s), $Q_{obs,k}$ is the discharge of the observed hydrograph for time index k (m^3/s), and $\overline{Q}_{obs}$ is the mean of the discharge of the observed hydrograph during whole event period K. The better the fit, the closer CE is to 1.

(b) The error of peak discharge, EQ_p (%), is defined as

$$EQ_p(\%) = \frac{Q_{p,est} - Q_{p,obs}}{Q_{p,obs}} \times 100\%, \tag{11}$$

where $Q_{p,est}$ denotes the peak discharge of the simulated hydrograph (m^3/s) and $Q_{p,obs}$ is the peak discharge of the observed hydrograph (m^3/s).

(c) The error of the time for peak to arrive, ET_p, is defined as

$$ET_p = T_{p,est} - T_{p,obs}, \tag{12}$$

where $T_{p,est}$ denotes the time for the simulated hydrograph peak to arrive (hours) and $T_{p,obs}$ represents the time required for the observed hydrograph peak to arrive (hours).

(d) The error of total discharge volume, $VER(\%)$, is defined as

$$VER(\%) = \frac{\left(\sum_{k=1}^{K} Q_{est,k} - \sum_{k=1}^{K} Q_{obs,k}\right)}{\sum_{k=1}^{K} Q_{obs,k}} \times 100\%, \tag{13}$$

where $Q_{est,k}$ denotes the discharge of the simulated hydrograph for time index k (m^3/s), $Q_{obs,k}$ is the discharge of the observed hydrograph for time index k (m^3/s). The better the fit, the closer EQ_p, ET_p and VER are to 0.

4. RESULTS AND DISCUSSIONS

The purpose of this study is to apply a dynamic recurrent neural network, calibrated by indirect system identification, to simulate rainfall-runoff processes. Furthermore, unit hydrographs are represented from the weights of the dynamic recurrent neural network based on equation (9). One state space model and two dynamic recurrent neural networks (DRNN(1) and DRNN(2)) are adopted to compare the performance of rainfall-runoff simulation and the comparison is discussed later.

4.1 Calibration of a Dynamic Recurrent Neural Network

For setting up a DRNN(2), ten events from 1966 to 1971 are selected to calibrate models via indirect system identification, and a singular value plot, shown as Fig. 4 is obtained during the indirect system identification. From the Fig. 4, the first two singular values are significant hence DRNN(2) has at least 2 neurons in its hidden layer. However, the other singular values do not go down dramatically that causes the optimum number of neurons in hidden layer is not easily gotten. Therefore, the relation between coefficient of efficiency and number of neurons in hidden layer of a DRNN(2) is illustrated as Fig. 5, and 6 neurons in hidden layer are selected as the optimum DRNN(2) according to this figure.

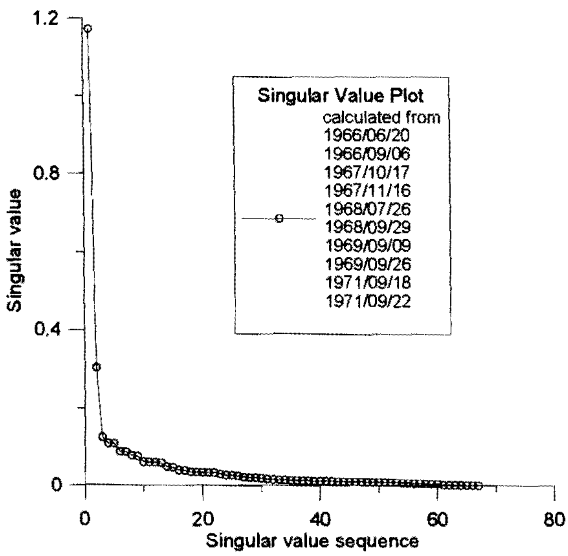


Fig. 4. Singular value plot.

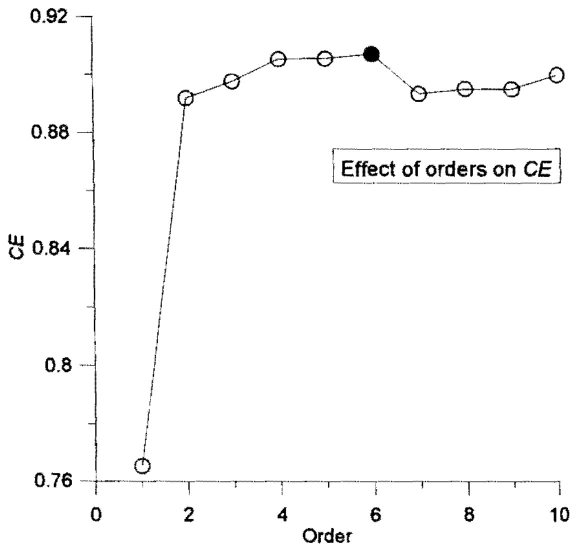


Fig. 5. The relation between coefficient of efficiency and number of neurons in hidden layer of a DRNN.

The combination of neural networks and indirect system identification to recognize the appropriate DRNN(2) is proposed in this paper. Generally, the weights of a neural network are generated randomly and are set as initial weights after training. However, indirect system identification not only offers a efficient way to design the appropriate number of neurons in the hidden layer of a DRNN(2), but also determines the weights of the DRNN(2) without training procedure.

4.2 The Realization of a Unit Hydrograph

One of the advantages of applying dynamic recurrent neural networks to simulate rainfall-runoff processes is the capacity for on-line learning. Through the realization of a linear system, the changes of the weights of a DRNN can be transformed into the transition of unit hydrographs and the result is shown as Fig. 6. In Fig. 6, three unit hydrographs are original discrete unit hydrograph directly carried out by linear programming, the unit hydrograph realized from DRNN(2) before on-line learning, and the unit hydrograph realized from DRNN(2) after on-line learning, respectively.

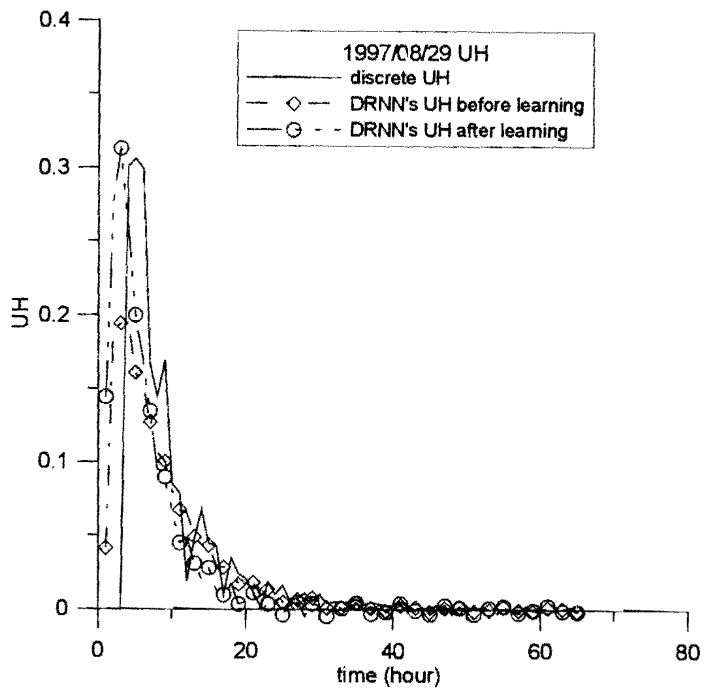


Fig. 6. The transition of unit hydrographs during Amber typhoon event of 29 August 1997 on the Wu-Tu watershed.

Obviously, in the tail of the original discrete unit hydrograph appears an oscillation caused by linear programming. However, the other unit hydrographs realized from DRNN(2) are smoother because the indirect system identification diminishes the high-order noise through selection of the significant singular values from singular value plot. In other words, unit hydrographs derived from DRNN(2) are nearer physical unit hydrographs than the discrete unit hydrograph carried out by linear programming. Furthermore, the two unit hydrographs realized from DRNN(2) show the capacity of on-line learning. After on-line learning, the unit hydrograph derived from DRNN(2) after on-line learning appears closer to original discrete unit hydrograph than before on-line learning.

4.3 Rainfall-runoff Simulation

41 rainfall-runoff events of Wu-Tu watershed are simulated by three models: state space model, DRNN(1), and DRNN(2). The simulated hydrographs (Amber typhoon event on Aug. 29, 1997) are shown as Fig. 7, and the performances evaluated by four criteria are listed as Table 1. From Fig. 7, the upper *x*-axis shows the rainfall series that cause the runoff series on lower *x*-axis, and the causal relationship implies that the runoff series should reveal the trend of rainfall series, no matter what hydrological model adopted. In other words, even the rainfall-runoff event is multi-peaks, the performance of simulation is good when the trends of rainfall and runoff are similar. Furthermore, Fig. 6 also shows that the dynamic recurrent neural networks perform better than a state space model because dynamic recurrent networks can modify the weights on-line.

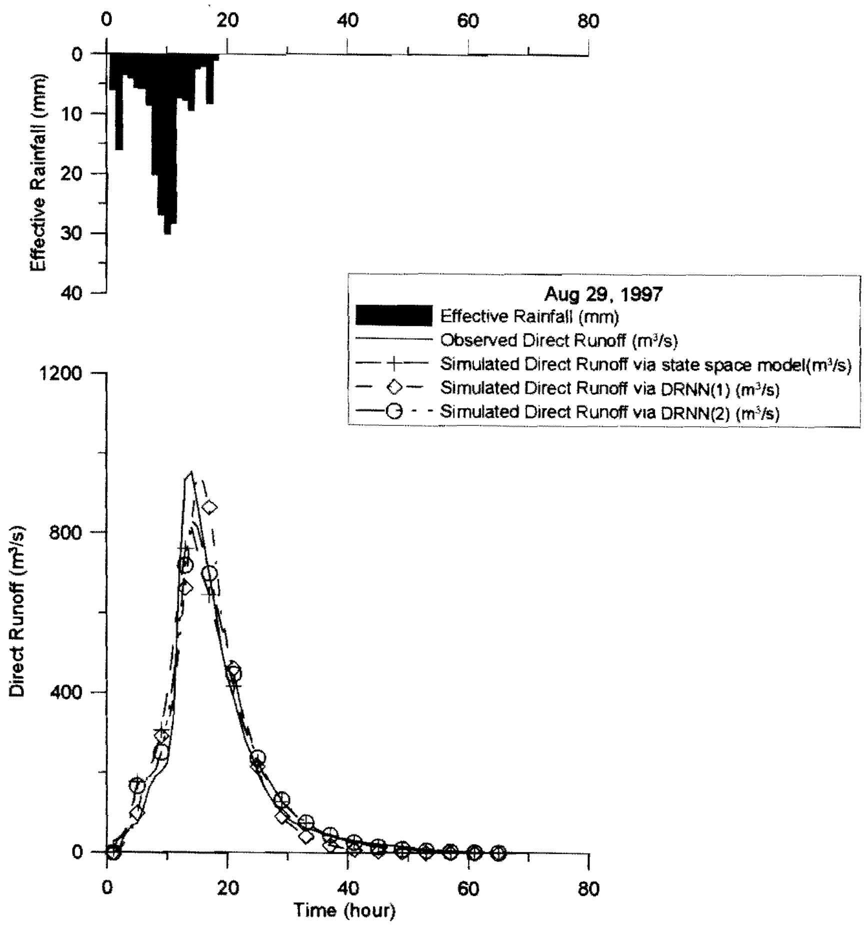


Fig. 7. Example of rainfall and runoff observations and simulations during Amber typhoon.

Table 1. Absolute averages of criteria.

model	absolute averages of criteria			
	CE	EQp(%)	ETp(hour)	VER(%)
DRNN(1)	0.8993	13.1369	1.1250	3.6090
DRNN(2)	0.8812	15.1128	1.3438	4.0181
SS	0.8506	15.5575	1.4063	3.3474

DRNN(1): DRNN constructed by trial-and-error method

DRNN(2): DRNN constructed by indirect system identification

SS: state space model

From Table 1, DRNN(1) performs best all criteria, except *VER*; the performance of DRNN(2) is close to DRNN(1). To compare DRNN(1) and DRNN(2), both neural networks adopt the same on-line learning algorithm, but the performances are different. The main reason is initial weights. The major difference between DRNN(1) and DRNN(2) is the weights of the recurrent links among the hidden layer. The weights of DRNN(1) are generated randomly, and that gives the dynamic recurrent neural network more elasticity to capture the character of a dynamic system on-line. However, considering the calibration procedure and the validation procedure, DRNN(2) is recommended because applying indirect system identification helps to calibrate a dynamic recurrent neural network quickly.

5. CONCLUSION

In this paper, dynamic recurrent neural networks are adopted to simulate rainfall-runoff processes, and an indirect system identification is proven as an efficient algorithm to recognize the optimum structure of a dynamic recurrent neural network and determine the weights of networks without training processes. Furthermore, the capacity for unit hydrograph representation, that helps to regain more information from the weights of the DRNN, is endowed with the DRNN. Finally, the transition of rainfall-runoff processes can be presented from the change of unit hydrographs that are realized from the weights of the DRNN modified via on-line learning.

ACKNOWLEDGEMENTS

The authors would like to thank the Council of Agriculture, Executive Yuan of the Republic of China for financially supporting this research.

REFERENCE

Amir, F. A. and Alexander, G. P. (2000) New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Trans. on Neural Networks*, 11(3), 697-709.

András and Szöllösi-nagy (1982) The discretization of the continuous linear cascade by means of state space analysis. *Journal of Hydrology*, 58, 223-236.

Cheng, S. J. and Wang, R. Y. (2002). An approach for evaluating the hydrological effects of urbanization and its application. *Hydrol. Process.*, 16, 1403-1418.

Chow, V. T., et al. (1964) Handbook of applied hydrology. *McGraw-Hill Book Company.*

Chow, V. T., Maidment, D. R. and Mays, L. W. (1988) Applied hydrology, *McGraw-Hill Book Company.*

Cooper, D. M. and Wood, E. F. (1982) Identification of multivariate time series and multivariate input-output models. *Water Resour. Res.*, 18(4), 937-946.

Haykin, S. (1999) Neural networks: a comprehensive foundation. *Prentice-Hall.*

Wang, R. Y. and Pan, T. Y. (2001) A study on the state space rainfall-runoff model and its application. *Journal of Chinese Agricultural Engineering*, 47(2), 1-16.

Zamarreño, J. M., Vega, P., García, L. D., and Francisco, M. (2000) State-space neural network for modeling: prediction and control. *Contr. Engin. Practice*, 8, 1063-1075.