Journal of Hydrology 497 (2013) 71–79

Contents lists available at SciVerse ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Reinforced recurrent neural networks for multi-step-ahead flood forecasts

Pin-An Chen[a], Li-Chiu Chang[b], Fi-John Chang[a,*]

[a] *Department of Bioenvironmental Systems Engineering, National Taiwan University, Taiwan, ROC*
[b] *Department of Water Resources and Environmental Engineering, Tamkang University, Taiwan, ROC*

ARTICLE INFO

Article history:
Received 1 February 2013
Received in revised form 2 May 2013
Accepted 21 May 2013
Available online 28 May 2013
This manuscript was handled by Andras Bardossy, Editor-in-Chief, with the assistance of Purna Chandra Nayak, Associate Editor

Keywords:
Reinforced real-time recurrent learning (R-RTRL) algorithm
Recurrent neural network (RNN)
Multi-step-ahead forecast
Flood forecast

SUMMARY

Considering true values cannot be available at every time step in an online learning algorithm for multi-step-ahead (MSA) forecasts, a MSA reinforced real-time recurrent learning algorithm for recurrent neural networks (R-RTRL NN) is proposed. The main merit of the proposed method is to repeatedly adjust model parameters with the current information including the latest observed values and model's outputs to enhance the reliability and the forecast accuracy of the proposed method. The sequential formulation of the R-RTRL NN is derived. To demonstrate its reliability and effectiveness, the proposed R-RTRL NN is implemented to make 2-, 4- and 6-step-ahead forecasts in a famous benchmark chaotic time series and a reservoir flood inflow series in North Taiwan. For comparison purpose, three comparative neural networks (two dynamic and one static neural networks) were performed. Numerical and experimental results indicate that the R-RTRL NN not only achieves superior performance to comparative networks but significantly improves the precision of MSA forecasts for both chaotic time series and reservoir inflow case during typhoon events with effective mitigation in the time-lag problem.

1. Introduction

Accurate multi-step-ahead (MSA) forecast is valuable and desired in many engineering problems, such as rainfall and flood forecasts, however it is a challenging task and difficult to achieve. A common approach to the MSA forecast is to update network parameters through online learning techniques. Online learning is a supervised machine-learning framework, which adopts the latest observed values to adjust model parameters for better mappings between instances and true values in a system. Because most observational disciplines tend to infer properties of an uncertain system from the analysis of time-dependent data, analytical technologies for extracting the meaningful characteristics of time series data have some inherent limitations, which has been a widely discussed issue for a long time (Brockwell and Davis, 1991; Jaeger and Haas, 2004; Jothiprakash and Magar, 2012; Nair et al., 2001). Online learning algorithms have several practical and theoretical advantages such as memory-efficient implementation, runtime-efficient implementation and strong guarantees on performance even in a highly variable data structure of time series (Shalev-Shwartz et al., 2004) owing to the continual receipt of true values for adjusting model parameters. Nevertheless, the main defect of online learning is ascribed to the requirement for continual true values. Engineering problems frequently require models to predict many time-steps into the future without the availability of measurements in the horizon of interest. The lack of true values makes it difficult to achieve MSA forecasts. In addition, many studies indicated it is not an adequate strategy to recursively adopt single-step-ahead predictions for many time-steps into the future because the errors of MSA predictors will be accumulated based on the single-step-ahead predictor (Parlos et al., 2000; Yong et al., 2010). Such time-lag problems may cause significant performance degradation when dealing with MSA forecasts for real-world applications. For the MSA streamflow forecasts during typhoon events, models with time-lag problems (i.e. no updating latest observed values) cannot keep flow trails, especially in peak flows, as the forecasting step increases. To mitigate time-lag phenomena occurred in online learning algorithms, it is argued whether iterative adjustments of model parameters based on additional information, such as the latest true values and/or antecedent model outputs, would be beneficial to MSA forecasts.

Artificial neural networks (ANNs) have the ability to approximate nonlinear functions and therefore become valuable tools for various water resources problems (Cho et al., 2011; Nayak et al., 2005; Nikolos et al., 2008; Nourani et al., 2011; Nourani and Sayyah Frad, 2012). However, static neural networks might fail to establish reliable nonlinear models for predicting dynamical

[*] Corresponding author. Tel.: +886 2 23639461; fax: +886 2 23635854.
E-mail address: changfj@ntu.edu.tw (F.-J. Chang).

0022-1694/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jhydrol.2013.05.038

systems, especially for many time-steps ahead. Alternatively, recurrent neural networks (RNNs) are computationally powerful nonlinear models capable of extracting dynamic behaviors from complex systems through internal recurrence and thus have attracted much attention for years (Assaad et al., 2005; Chiang et al., 2010; Coulibaly and Baldwin, 2005; Coulibaly, 2010; Ma et al., 2008; Muluye, 2011; Serpen and Xu, 2003). The batch training of RNNs, however, is mathematically sophisticated and time-consuming (Ahmad and Jie, 2002; Xie et al., 2006). The real-time recurrent learning (RTRL) algorithm, proposed by Williams and Zipser (1989), is an effective and efficient online learning algorithm for training recurrent networks, in which the real-time adjustments are made to the synaptic weights of recurrent networks. Several studies demonstrated that the RTRL algorithm for RNNs is very effective in modeling the dynamics of complex processes for providing accurate predictions (Chang et al., 2002, 2012; Hirasawa et al., 2000; Li et al., 2002).

The main goal of this study is to develop a reinforced RTRL algorithm for RNNs (R-RTRL NN) to mitigate time-lag effects for increasing the accuracy of MSA forecasts. The sequential formulation of the R-RTRL NN is derived and its reliability and applicability are further demonstrated through two-step-ahead (2SA), four-step-ahead (4SA) and six-step-ahead (6SA) forecasts made for a famous benchmark chaotic time series and a reservoir inflow case in Taiwan. Comparative models consist of the original RTRL algorithm for RNNs (RTRL NN), layer recurrent network (LRN) (Liu and Wang, 2008; Liu et al., 2012) and the most popular static ANN (i.e. back-propagation neural network (BPNN)).

2. Formulation of MSA R-RTRL algorithm for RNNs

2.1. Rationale of MSA online learning algorithm

Two common strategies for MSA forecasts are the iterated prediction and direct prediction. For n-step-ahead (nSA) prediction, the iterated method tackles the issue by iterating n times a one-step-ahead prediction whereas the direct method trains the model by conducting a direct forecast at time $t+n$. The debate on the superiority between these two methods still remains open; nevertheless both methods possess a common feature: the visibility of stochastic dependencies between future values becomes relatively vague as the time of prediction horizon increases, consequently the reliability and accuracy of predictions decreases. A possible way to remedy this shortcoming is to implement online learning techniques for repeatedly adjusting model parameters with the most current information including the latest true (observed) values and model's outputs. An online learning algorithm proceeds in a sequence of trials through receiving an instance and making a prediction on each online-learning round to improve model performance.

The original RTRL algorithm, an online learning algorithm, was derived for one-step-ahead forecasts from the fact that real-time adjustments are made to the synaptic weights of a fully connected recurrent neural network (Williams and Zipser, 1989). For nSA forecasts, the weight adjustment of the RTRL algorithm cannot be conducted until obtaining the observed value at time $t+n$, in which the observed values and model outputs during time $t+1$ and $t+n-1$ are totally ignored and worthless. Therefore, the effectiveness of the original RTRL algorithm decreases considerably when time step n increases, which implies time lags occur in the weight adjustment process.

In this study, a novel reinforced RTRL algorithm based on RNN infrastructures (R-RTRL NN) for MSA forecasts through incorporating the latest antecedent forecasted and observed values into consecutive temporary networks for weight adjustments in the learning process is proposed. In other words, the R-RTRL algorithm repeatedly updates the synaptic weights by utilizing the most current obtainable information. The applicability and effectiveness of the R-RTRL NN is further investigated in Section 3.

The upper diagram of Fig. 1 shows the weight adjustment procedure of the R-RTRL algorithm for 2SA forecast in our earlier attempt (Chang et al., 2012). At time $t+2$, the weights are adjusted by the differences between observed and forecasted values. A reinforced process is introduced: the RNN_{temp} with adjusted weights can be used to produce a temp output $\hat{z}(t+3,1)$ at time $t+1$, and the error between the temp output and forecasted output at time $t+1$ can then be utilized to reinforce the weight adjustments, $\Delta\widehat{W}_1(t)$ and $\Delta\widehat{V}_1(t+1)$. As this reinforced process repeats $n-1$ times, the weight adjustment procedure can be extended to a general procedure for nSA ($n \geqslant 2$, $n \in N$) forecast, shown in the lower diagram of Fig. 1. In summary, the proposed R-RTRL algorithm not only utilizes the up-to-date information of the observed values and their corresponding model outputs adequately but also strengthens the usefulness of the latest observed values by the reinforced process to mitigate the time-lag phenomenon for MSA forecasts. The detailed sequential formulation of the R-RTRL algorithm is described as follows.

2.2. Deriving the MSA R-RTRL algorithm

Fig. 2 shows the MSA RNN architecture incorporated with the R-RTRL algorithm, in which there are M external inputs and one output. Let $\mathbf{X}(t)$ denote the $M \times 1$ input vector at discrete time t, $\mathbf{Y}(t+1)$ denote the corresponding $N \times 1$ vector at time $t+1$ in the processing layer, and $\mathbf{Z}(t+n)$ denote the corresponding output value for nSA ($n \geqslant 2$, $n \in N$) forecast.

The $\mathbf{X}(t)$ and $\mathbf{Y}(t)$ are concatenated to form the $(M+N) \times 1$ vector $\mathbf{U}(t)$, whose ith element is denoted by $\mu_i(t)$. Let A denote the set of indices i for which $x_i(t)$ is an external input, and B denote the set of indices i for which $y_i(t)$ is the output of the processing layer. Thus, vector $\mu_i(t)$ can be represented as follows.

$$\mu_i(t) = \begin{cases} x_i(t) & \text{if } i \in A \\ y_i(t) & \text{if } i \in B \end{cases} \tag{1}$$

W and V denote the weight matrices in the processing layer and output layer, respectively. $W \leftrightarrow w_{ji}$ and $V \leftrightarrow v_j$ are of matrix forms. The output of neuron j in the processing layer that presents the transformation of information from the concatenated layer through nonlinear system f is given by

$$y_j(t+1) = f(net_j(t+1)) = f\left(\sum_{i \in A \cup B} w_{ji}(t)\mu_i(t)\right) \tag{2}$$

The net output of the output layer at time $t+n$ through nonlinear system f is computed by

$$z(t+n) = f(net(t+n)) = f\left(\sum_j v_j(t+1)y_j(t+1)\right) \tag{3}$$

Let $d(t+n)$ denote the target value at time $t+n$. The time-varying error e and instantaneous error E is defined by

$$E(t+n) = \frac{1}{2}e^2(t+n) = \frac{1}{2}[d(t+n) - z(t+n)]^2 \tag{4}$$

Then the weight adjustments can be computed by minimizing the instantaneous error at time $t+n$.

$$\Delta v_j(t+1) = -\eta_1 \frac{\partial E(t+n)}{\partial v_j(t+1)} \tag{5}$$

$$\Delta w_{ji}(t) = -\eta_2 \frac{\partial E(t+n)}{\partial w_{ji}(t)} \tag{6}$$

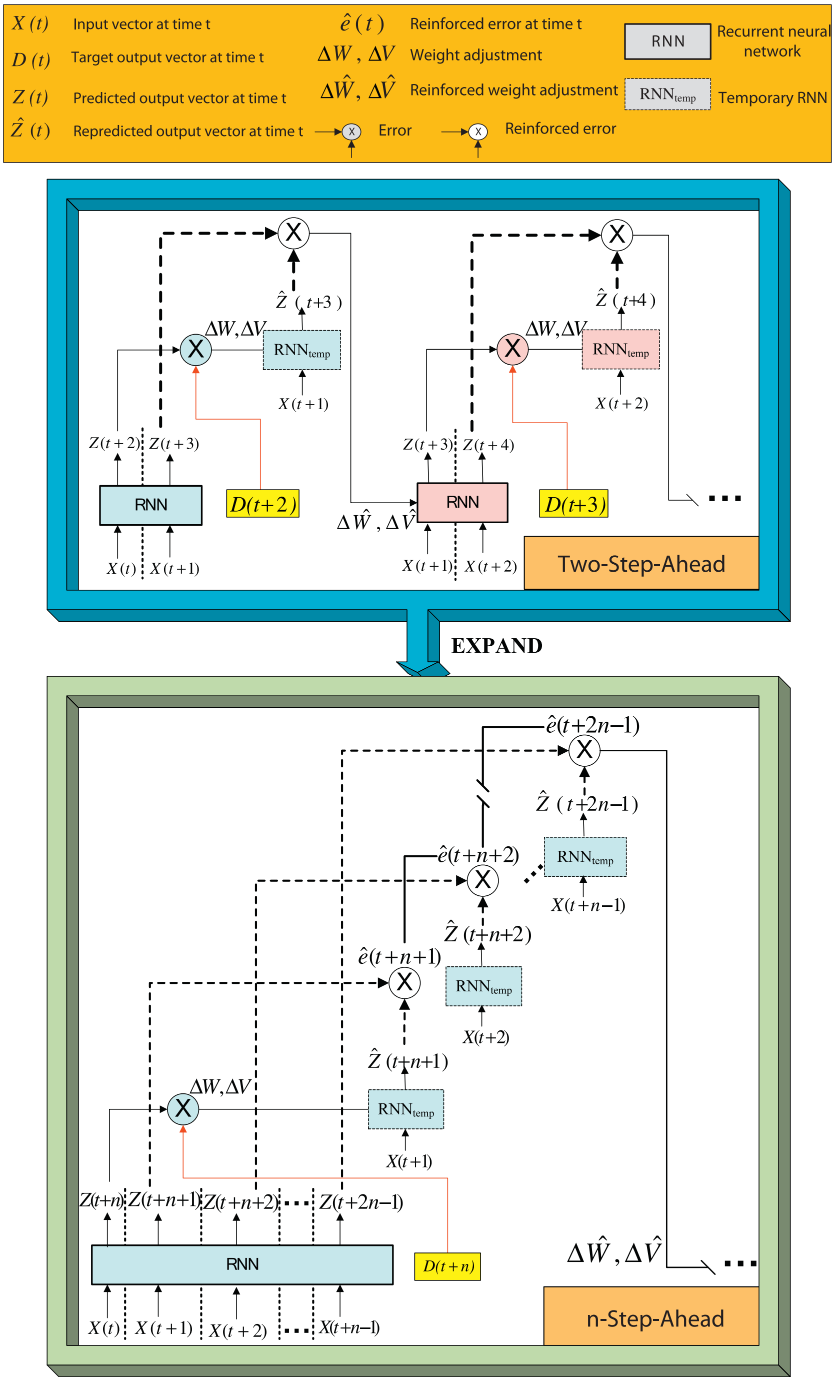
X (t) Input vector at time t
D (t) Target output vector at time t
Z (t) Predicted output vector at time t
Ẑ (t) Repredicted output vector at time t
ê (t) Reinforced error at time t
ΔW , ΔV Weight adjustment
ΔŴ, ΔV̂ Reinforced weight adjustment
X Error
X Reinforced error
RNN Recurrent neural network
RNN_temp Temporary RNN
Ẑ (t+3)
ΔW,ΔV
RNN_temp
X(t+1)
Z(t+2) Z(t+3)
RNN
D(t+2)
X(t) X(t+1)
Ẑ (t+4)
ΔW,ΔV
RNN_temp
X(t+2)
Z(t+3) Z(t+4)
RNN
D(t+3)
ΔŴ , ΔV̂
X(t+1) X(t+2)
Two-Step-Ahead
EXPAND
ê(t+2n−1)
Ẑ (t+2n−1)
RNN_temp
X(t+n−1)
ê(t+n+2)
Ẑ (t+n+2)
RNN_temp
X(t+2)
ê(t+n+1)
Ẑ (t+n+1)
RNN_temp
X(t+1)
ΔW,ΔV
Z(t+n) Z(t+n+1) Z(t+n+2) Z(t+2n−1)
RNN
D(t+n)
X(t) X(t+1) X(t+2) X(t+n−1)
ΔŴ , ΔV̂
n-Step-Ahead

Fig. 1. Weight adjustment procedure for the *n*-step-ahead R-RTRL NN.

where η_1 and η_2 are the learning-rate parameters. The detailed recurrent learning algorithm for one-step-ahead weight adjustment can be found in Williams and Zipser (1989) and two-step-ahead weight adjustment can be found in Chang et al. (2012). The entire antecedent information is considered crucial and could further diminish time-lag effects. Consequently, the reinforced two-step weight adjustments (Chang et al., 2012) can be extended to *n*-step weight adjustments, and the information obtained from time $t+1$ to $t+n-1$ can contribute to weight adjustments. The adjusted weights are used to re-calculate the forecasted values at time from $t+1$ to $t+n-1$, and then the adjusted weights are further modified by minimizing the total error between original forecasted values $(z(t+n+1)$ to $z(t+2n-1))$ and the re-forecasted values $(\hat{z}(t+n+1)$ *to* $\hat{z}(t+2n-1))$.

The re-forecasted value $\hat{z}(t+n+p)$ is calculated by the following equations:

$$\hat{y}_j(t+p+1) = f\left(\widehat{net}_j(t+p+1)\right) = f\left(\sum_{i\in A\cup B}(w_{ji}(t)+\Delta w_{ji}(t))\mu_i(t+p)\right) \quad (7)$$

$$\hat{z}(t+n+p) = f\left(\widehat{net}(t+n+p)\right) = f\left(\sum_j (v_j(t+1)+\Delta v_j(t+1))\hat{y}_j(t+p+1)\right) \quad (8)$$

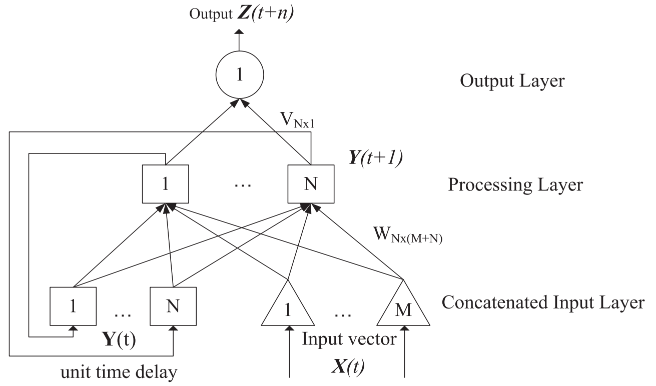

Fig. 2. Architecture of the multi-step-ahead RNN with R-RTRL online learning algorithm.

where p denotes the time step ($p = 1, 2, \ldots, n-1$). Therefore, the total reinforced error is defined by

$$\widehat{E} = \frac{1}{2}\sum_{p=1}^{n-1}\hat{e}^2(t+n+p) = \frac{1}{2}[\hat{z}(t+n+p) - z(t+n+p)]^2 \tag{9}$$

The reinforced weight adjustments can be written as

$$\Delta\hat{v}_j(t+1) = -\eta_3\frac{\partial\widehat{E}}{\partial v_j(t+1)} \tag{10}$$

$$\Delta\hat{w}_{ji}(t) = -\eta_4\frac{\partial\widehat{E}}{\partial w_{ji}(t)} \tag{11}$$

where η_3 and η_4 are the learning-rate parameters.

The weight adjustments of the R-RTRL algorithm for n-step-ahead RNNs are then shown as follows:

$$W_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t) + \Delta\widehat{w}_{ji}(t) \tag{12}$$

$$v_j(t+2) = v_j(t+1) + \Delta v_j(t+1) + \Delta\hat{v}_j(t+1) \tag{13}$$

In sum, the reinforced process is implemented for nSA forecast so that the adjusted weights are further modified through the comparison between the original forecasted value and the re-forecasted value.

3. Simulation results and discussion

3.1. Phase space reconstruction

The first stage in the analysis of chaotic time series is to implement the phase space reconstruction theory, which can reconstruct a nonlinear model with a low-dimensional phase space to reflect the actual dynamic system. Based on the Takens' embedding theorem (Takens, 1981), a series of observations from a chaotic system can reconstruct an attractor, a subset of the phase space of the system with two parameters of time delay (T) and dimension (D). If appropriate (T) and (D) are selected, the attractor will retain topological properties and reveal the hidden information of the original dynamic system. The mutual information method (Fraser and Swinney, 1986) and the Cao's method (Cao, 1997) are used to determine the proper time delay (T) and dimension (D), respectively.

Therefore, for an observed time series $x(t)$, the attractor $\mathbf{x}(t) = [x(t), x(t-T), \ldots, x(t-(D-1)T)]$ can be formed. Then the future value at time $t + n$ is determined by the nonlinear function $\boldsymbol{F}$, which governs the system. The nonlinear function $\boldsymbol{F}$ is defined as follows:

$$x(t+n) = F\{[x(t), x(t-T), \ldots, x(t-(D-1)T)]\} \tag{14}$$

The well trained ANN models can approximate the governing function $\boldsymbol{F}$. As a result, the patterns of input–output data pairs of all ANN models in this study are shown as $\{X(t), x(t+n)\}$ and applied to chaotic time series.

3.2. MSA forecast of Mackey–Glass time series

To demonstrate and evaluate whether the proposed R-RTRL NN can construct a reliable multi-step-ahead predictor by effectively utilizing the most current information, the developed neural network model is first compared with a simulated chaotic system (Mackey–Glass time series) for direct 2SA, 4SA and 6SA forecasts. Its performance is then compared with that of the RTRL NN, LRN and BPNN. The LRN and BPNN configurations are used in many filtering and modeling applications for time series, which have already been widely discussed (Chen et al., 2010; Zabiri et al., 2009).

The Mackey–Glass differential delay equation (MacKey and Glass, 1977) is defined below:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{15}$$

where the initial condition $x(0) = 1.2$ and $\tau = 17$. 1000 input–output data pairs are generated, where the first 500 pairs are used for training while the remaining 500 pairs are used for testing. The embedding dimension $D = 3$ and time delay $T = 7$ are determined according to the Cao's method (Cao, 1997) and mutual information method (Fraser and Swinney, 1986).

For forecasting 2SA, 4SA and 6SA Mackey–Glass time series, the input and processing layers of the RTRL NN and the proposed R-RTRL NN consist of 3 and 8 neurons, respectively, while the

Table 1
Model performance of two- to six-step-ahead forecast for Mackey–Glass time series.

		Training		Testing			
		RMSE	MAE	RMSE	MAE	G_{bench}	$G_{bench,II}$
R-RTRL NN	*2SA*	3.56E−03	2.79E−03	3.51E−03	2.77E−03	0.997	–
	4SA	4.23E−03	2.59E−03	4.06E−03	2.52E−03	0.999	–
	6SA	4.79E−03	3.33E−03	4.79E−03	3.35E−03	0.999	–
RTRL NN	*2SA*	6.69E−03	5.40E−03	6.51E−03	5.30E−03	0.990	−2.44
	4SA	7.43E−03	5.17E−03	7.40E−03	5.13E−03	0.997	−2.32
	6SA	9.09E−03	7.17E−03	8.94E−03	7.01E−03	0.998	−2.49
LRN	*2SA*	6.37E−03	4.44E−03	6.27E−03	4.31E−03	0.990	−1.38
	4SA	6.49E−03	5.08E−03	6.43E−03	5.00E−03	0.998	−0.80
	6SA	6.60E−03	5.17E−03	6.49E−03	5.10E−03	0.999	−2.42
BPNN	*2SA*	9.01E−03	6.16E−03	9.05E−03	6.07E−03	0.981	−5.66
	4SA	1.30E−02	1.03E−02	1.29E−02	1.02E−02	0.990	−9.08
	6SA	1.52E−02	1.32E−02	1.52E−02	1.31E−02	0.993	−9.03

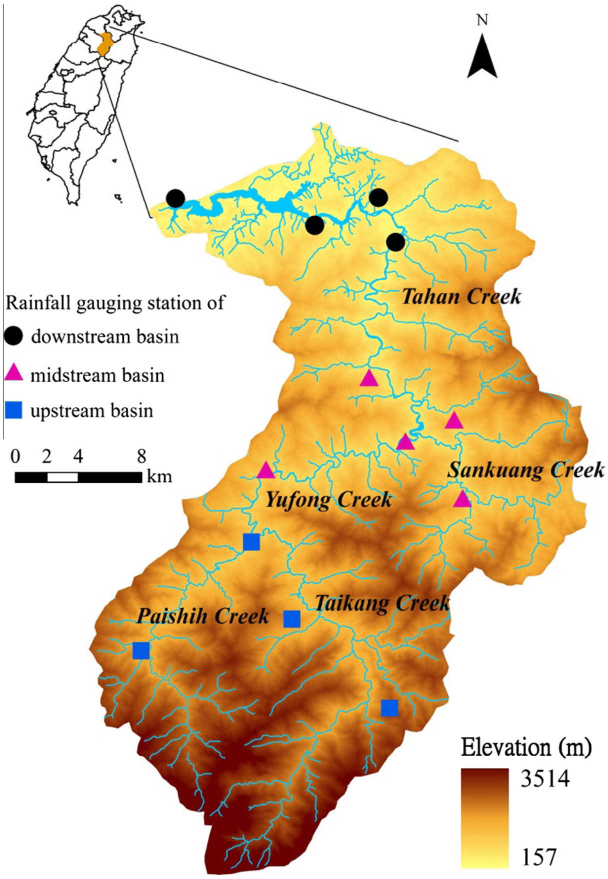

Fig. 3. Location of the Shihmen Reservoir and rainfall gauging stations.

two-layer LRN and BPNN trained by the Levenberg–Marquardt back propagation algorithm also consist of 3 and 8 neurons in the input and hidden layers, respectively. The numbers of the processing (hidden-layer) neurons and layers determined above for these four models are identified as the best structures by trial and error. A training dataset is used to construct the aforementioned four neural network models for direct 2SA, 4SA and 6SA forecasts.

The root mean square error (RMSE), mean absolute error (MAE), and the goodness-of-fit with respect to the benchmark time series (G_{bench}) are used as performance criteria (Nash and Sutcliffe, 1970; Seibert, 2001). The G_{bench} is defined as:

$$G_{bench} = 1 - \frac{\sum_{i=1}^{n}(Q_i - \widehat{Q}_i)^2}{\sum_{i=1}^{n}(Q_i - Q_{i,bench})^2} \tag{16}$$

where Q_i is the observed value in the ith step, $\widehat{Q}_i$ is the forecasted value in the ith step, and n is the number of data points. $Q_{i,bench}$ is the antecedent observed value, i.e., $Q_{i,bench} = Q_{i-n}$ for nSA forecast. To compare the proposed R-RTRL NN with comparative models, the criterion $G_{bench,II}$ is defined, where the benchmark $Q_{i,bench}$ is the forecasted value of the proposed R-RTRL NN in the ith step.

The results of model comparison are summarized in Table 1. It appears (1) all the models are suitably trained, and their training and testing results are quite consistent in all the cases; (2) the recurrent neural networks (i.e. the R-RTRL NN, RTRL NN and LRN) can provide much better performance than the static time-delay BPNN, in terms of RMSE and MAE values; and (3) the proposed R-RTRL NN has better performance than the RTRL NN, LRN and BPNN models for 2SA, 4SA and 6SA forecasts. Furthermore, the negative $G_{bench,II}$ values produced by the RTRL NN, LRN and BPNN reveal the superiority of the R-RTRL NN in MSA forecasts. In sum, the results demonstrate the proposed online learning algorithm (R-RTRL) that takes the closest antecedent information into consideration can effectively re-adjust synaptic weights in real time and the constructed model (R-RTRL NN) can provide reliable and accurate forecasts in real-time MSA forecasts.

4. Applications

4.1. Study area and datasets

Taiwan, located in the subtropical zone of the North Pacific Ocean, is an island with mountainous terrains and steep landforms, where typhoons usually couple with heavy rainfall and thus cause downstream flooding within a few hours. The Shihmen Reservoir, situated upstream of the Tahan Creek in northern Taiwan, has operated for multiple purposes including water supply, hydropower generation, flood mitigation and tourism. The reservoir has notably contributed to agricultural production, industrial development and the alleviation of drought disasters for decades.

Table 2
Summary statistics of reservoir inflow and average hourly rainfall in training and testing datasets.

	Training dataset				Testing dataset			
	Mean	SD[a]	Max	Min	Mean	SD	Max	Min
Inflow (cms)	725	1154	8594	1	720	910	5300	9
Average rainfall of downstream (mm h^{-1})	4.0	7.9	55.1	0.0	4.0	7.1	45.7	0.0
Average rainfall of midstream (mm h^{-1})	4.8	9.6	63.5	0.0	4.5	7.5	47.3	0.0
Average rainfall of upstream (mm h^{-1})	4.4	8.8	62.0	0.0	4.5	7.1	46.6	0.0

[a] Standard deviation.

Table 3
Model performance of two-step-ahead forecasts for reservoir inflow.

	Training					Testing					
	RMSE (cms)	MAE (cms)	CE	CC	G_{bench}	RMSE (cms)	MAE (cms)	CE	CC	G_{bench}	$G_{bench,II}$
R-RTRLNN	177	106	0.98	0.99	0.59	184	106	0.96	0.98	0.30	–
RTRL NN	214	112	0.97	0.98	0.40	228	124	0.94	0.97	−0.07	−0.53
LRN	211	107	0.97	0.98	0.42	220	118	0.94	0.97	−2.72E−04	−0.43
BPNN	220	113	0.96	0.98	0.37	232	125	0.93	0.97	−0.11	−0.59

Table 4
Model performance of four-step-ahead forecasts for reservoir inflow.

	Training					Testing					
	RMSE (cms)	MAE (cms)	CE	CC	G_{bench}	RMSE (cms)	MAE (cms)	CE	CC	G_{bench}	$G_{bench,II}$
R-RTRL NN	308	161	0.93	0.96	0.63	314	172	0.88	0.94	0.26	–
RTRL NN	351	177	0.91	0.95	0.53	362	198	0.84	0.92	0.02	−0.33
LRN	345	172	0.91	0.95	0.54	360	198	0.84	0.92	0.03	−0.32
BPNN	357	181	0.91	0.95	0.51	365	200	0.83	0.92	1.82E−03	−0.35

Table 5
Model performance of six-step-ahead forecasts for reservoir inflow.

	Training					Testing					
	RMSE (cms)	MAE (cms)	CE	CC	G_{bench}	RMSE (cms)	MAE (cms)	CE	CC	G_{bench}	$G_{bench,II}$
R-RTRL NN	440	238	0.86	0.93	0.62	490	302	0.71	0.85	0.07	–
RTRL NN	498	263	0.82	0.91	0.52	559	316	0.63	0.82	−0.22	−0.30
LRN	481	254	0.83	0.91	0.55	542	303	0.65	0.83	−0.15	−0.22
BPNN	525	276	0.80	0.89	0.46	600	324	0.57	0.82	−0.40	−0.50

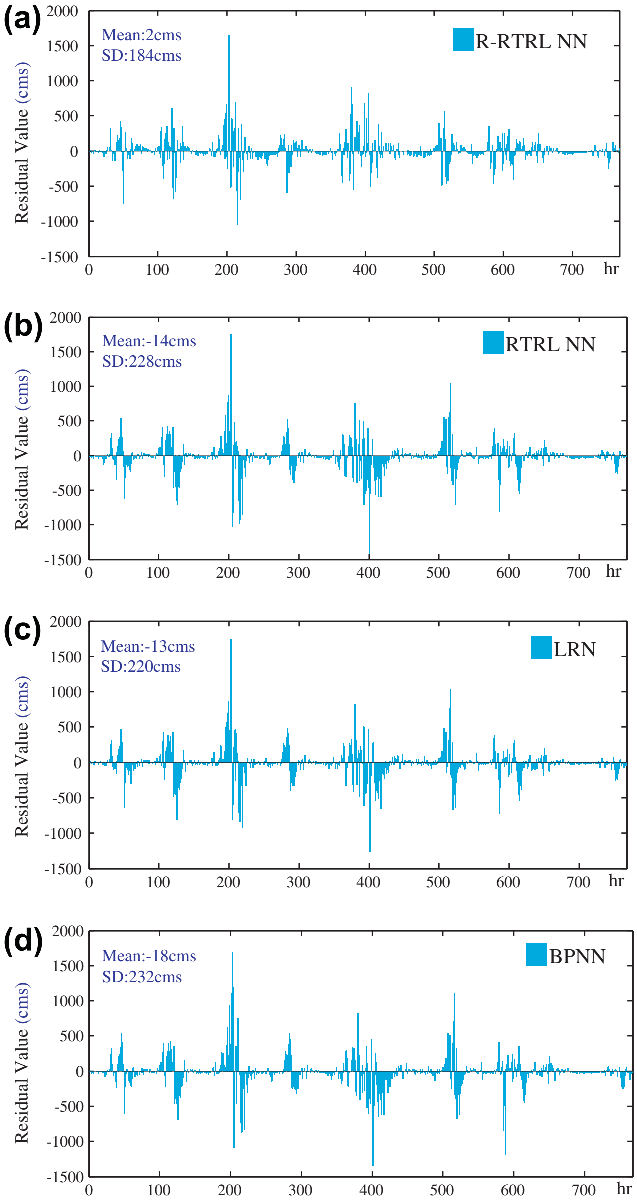

Fig. 4. *2SA* inflow forecast residuals (of testing data sets) based on (a) R-RTRL NN, (b) RTRL NN, (c) LRN, and (d) BPNN, respectively.

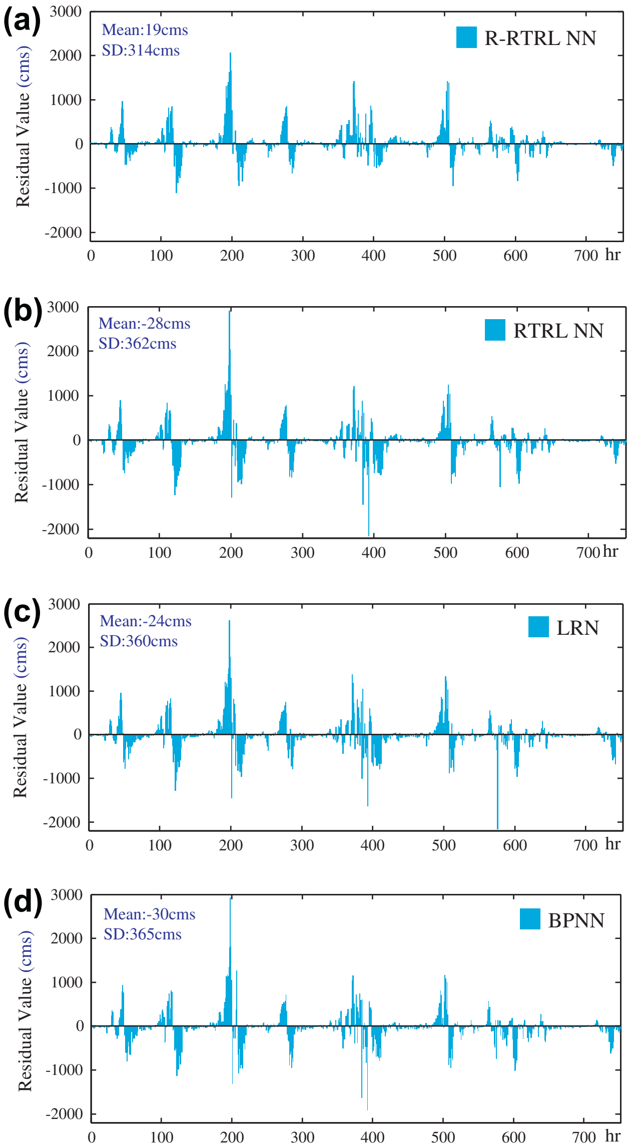

Fig. 5. *4SA* inflow forecast residuals (of testing data sets) based on (a) R-RTRL NN, (b) RTRL NN, (c) LRN, and (d) BPNN, respectively.

The basin area of the reservoir is about 763.4 km^2, and the effective waters storage of the reservoir is 251.88 million cubic meters. The Shihmen Reservoir is for sure the most important water resources facility to Taipei metropolitan areas. An accurate inflow forecast model for the reservoir is desired and crucial to flood control and water resources management. The longer the forecast steps into the future, the more beneficial it is, in terms of time to adjust reservoir operation and reduce flood damages.

Fig. 3 shows the locations of the basin and rainfall gauging stations in this case study. The hourly inflow and rainfall data of 22 typhoon events during 2001 and 2009 were collected. A total of 2136 datasets are used in this study, where 1296 datasets collected from 2001 to 2006 are used for training while the remaining 840 datasets collected from 2007 to 2009 are used to test the models independently. The Shihmen basin is briefly divided into three areas (up-, mid- and downstream areas shown in Fig. 3), and the weighted average rainfall of each area is computed by the Thiessen polygon method. The summary statistics for reservoir inflow and average rainfall datasets are presented in Table 2. It shows the extremely high (max.) inflow and large variance of inflow also implies the MSA inflow forecast is an important and challenging task in the Shihmen Reservoir.

4.2. MSA forecast of reservoir inflow time series

The proposed R-RTRL NN is applied to the Shihmen Reservoir for forecasting reservoir inflow during typhoon events and is also compared with the other three models (RTRL NN, LRN and BPNN) for performance evaluation. Because the transit time of flows moving from rainfall gauging stations to the Shihmen Reservoir is different, the Kendall's tau rank and Pearson's correlation coefficient are applied to the identification of the highest correlation among the different time lags between rainfall and reservoir inflow. The time lags of rainfall traveling from the up-, mid- and downstream basins to the reservoir are identified as 7, 6 and 5 h, respectively.

Therefore, the input layers of these different network models are established based on current inflow data ($Q(t)$) together with antecedent rainfall data (denoted by Ru(t-7), Rm(t-6) and Rd(t-5) accordingly) collected at up-, mid- and downstream basins with associated time lags. For direct *2*SA inflow forecasts, the input and processing layers of the RTRL NN and R-RTRL NN consist of 4 and 6 neurons, respectively. The two-layer LRN and BPNN trained by the Levenberg–Marquardt back propagation algorithm also consists of 4 and 6 neurons in the input and hidden layers, respectively. The numbers of the processing (hidden-layer) neurons and layers determined above for the models are mainly identified as the best structures through a great number of trial-and-error processes. The numbers of neurons in the processing and hidden layers of these four models are increased to 8 for direct 4SA and 6SA inflow forecasts. The performance of these four models is evaluated by the criteria of RMSE, MAE, coefficient of efficiency (Nash Efficiency or CE) (Nash and Sutcliffe, 1970), coefficient of correlation (CC), G_{bench} and $G_{bench,II}$.

Summarized results are presented in Tables 3–5. Results indicate that the proposed R-RTRL NN model can produces acceptable RMSE values (184–490 cms) when compared with the mean and standard deviation (720.10 ± 910.44 cms in the testing dataset of Table 2) of observed inflow data and has smaller RMSE and MAE values and higher CE, CC and G_{bench} values than the RTRL NN, LRN and BPNN models in both training and testing phases for 2SA, 4SA and 6SA inflow forecasts. It is noticed that the proposed R-RTRL NN makes less difference in RMSE values between training and testing phases than comparative models, which demonstrates the impressive generalizability of the proposed R-RTRL NN. In addition, the proposed R-RTRL NN model produces positive G_{bench} values in the testing datasets for 2SA, 4SA and 6SA forecasts whereas the comparative models only produce either negative or near-zero G_{bench} values in the testing datasets. The results indicate the proposed online learning algorithm (R-RTRL) that adopts the most current information can effectively mitigate the time-lag problem and the R-RTRL NN model can make more reliable and accurate MSA forecasts. When closely assessing the $G_{bench,II}$ values (where the bench series is the forecasted values of the proposed R-RTRL NN), all comparative models produce negative values for forecasting the 2SA, 4SA and 6SA inflow in the testing phases. This result provides extra evidence that the proposed R-RTRL NN achieves superior performance to comparative networks.

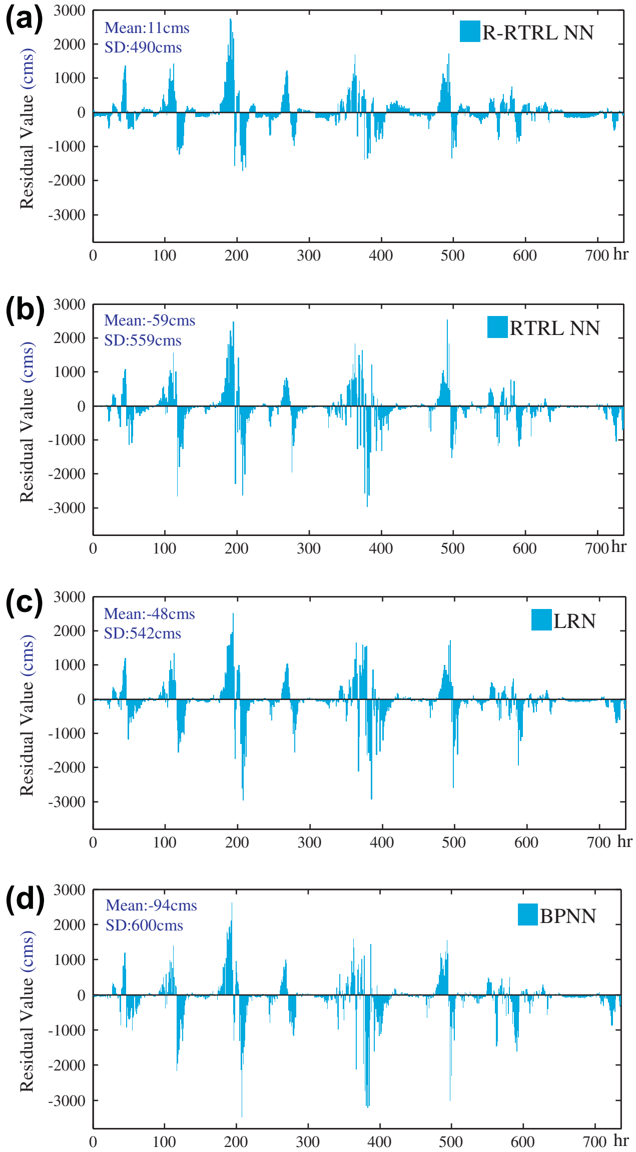

Fig. 6. 6SAinflow forecast residuals (of testing data sets) based on (a) R-RTRL NN, (b) RTRL NN, (c) LRN, and (d) BPNN, respectively.

Figs. 4–6 show the corresponding residuals and their mean and standard deviation produced by the R-RTRL NN, RTRL NN, LRN and BPNN models for 2SA, 4SA and 6SA forecasts in the testing datasets. The barplots of residuals clearly indicate that the R-RTRL NN provides the smallest residuals (the smallest absolute mean and standard deviation) as compared with the comparative models. To easily distinguish the performance of these four models, three typhoon events (Typhoons Krosa, Sinlaku and Jangmi) with high peak flow data (above 3000 cms) are extracted from the

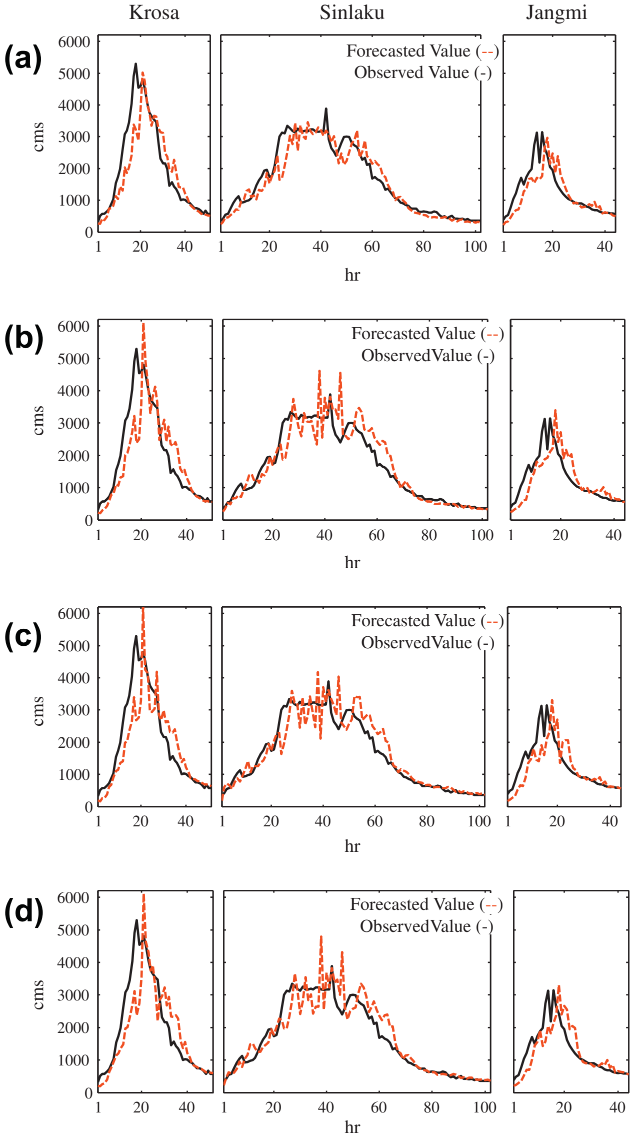

Fig. 7. 4SA inflow forecasts (Typhoons Krosa, Sinlaku and Jangmi in testing data sets) based on (a) R-RTRL NN, (b) RTRL NN, (c) LRN, and (d) BPNN, respectively.

testing dataset to illustrate the hydrographs of observed and 4SA forecasted inflow obtained from the proposed R-RTRL algorithm and comparative models (Fig. 7). It demonstrates that the proposed R-RTRL NN mitigates some time-lag problem and can well forecast 4SA inflow values whereas all comparative models not only have significant time-lag phenomena but fail to well forecast 4SA inflow values (seriously over-estimate and oscillate at peak flows; and the G_{bench} values of comparative models are negative). In addition, the low G_{bench} value (0.07) in Table 5 indicates the maximum forecast time step for the proposed R-RTRL NN to reach is six (6 h) in this study case, which can provide sufficient responding time to fully open the floodgates in the Shihmen Reservoir (usually take about 2 h) and more responding time for both reservoir flood control and flood warnings to downstream areas.

In summary, the relationship between forecast errors (RMSE) and forecast steps of these four models is presented in Fig. 8. It shows that these four models have similar error-rising rates (slopes) when the forecast step increases from 2SA to 4SA. However, the BPNN has the steepest slope (error-rising rate) when the forecast step increases from 4SA to 6SA, which indicates the static neural network fails to extract the dynamic characteristics of time variation for MSA forecasts. Alternatively, the proposed R-RTRL NN has similar error-rising rate to the LRN but has much smaller forecast errors than the LRN for 2SA, 4SA and 6SA forecasts. It appears that the proposed methodology can adequately utilize the closest antecedent information to effectively re-adjust synaptic weights. As a result, the constructed R-RTRL NN significantly diminishes time-lag effects and effectively provides much better and adequate MSA forecasts.

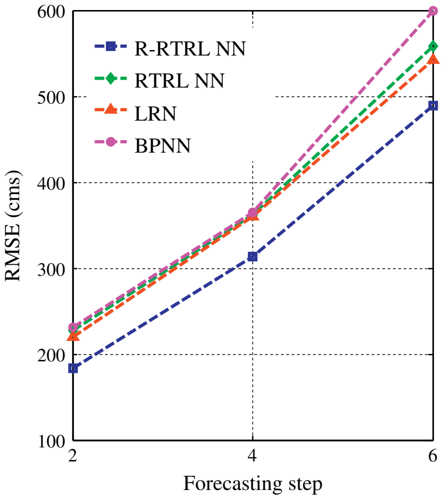

Fig. 8. Relationship between inflow forecast errors (RMSE) and forecast steps of four network models.

5. Conclusions

This study devotes to dealing with the connatural limitation of online learning algorithms that is caused by a lack of accurate target values in the future for MSA forecasts. A novel R-RTRL NN that not only adequately utilizes the antecedent information of the observations as well as model outputs but also strengthens their usefulness to mitigate time-lag phenomena as well as increases model accuracy in MSA forecasts is proposed. The rigorous demonstration with respect to the superiority of MSA R-RTRL NN necessitates the use of a benchmark chaotic time series and the real-world application of the flood series induced by typhoons at the Shihmen Reservoir in northern Taiwan. For comparison purpose, the original RTRL NN, LRN and BPNN are also performed.

In the cases of benchmark time series, results indicate that the proposed R-RTRL NN has much better performance than comparative models for MSA forecasts. When modeling the flood series during typhoon events, the proposed R-RTRL NN also shows great superiority on 2SA, 4SA and 6SA forecasts with significant reductions in time-lag effects to the original RTRL NN, LRN and BPNN by analyzing the relationship between forecast errors and forecast steps. This study demonstrates that the developed R-RTRL algorithm for RNNs by incorporating the closest antecedent information into the online learning process has good practicability and produces high accuracy for MSA forecasts.

Acknowledgments

This study was funded by the National Science Council, Taiwan, ROC (Grant No. 100-2313-B-002-011-MY3). The authors are indebted to Editors and reviewers for their valuable comments and constructive suggestions.

References

Ahmad, Z., Jie, Z., 2002. Improving long range prediction for nonlinear process modelling through combining multiple neural networks Control Applications. Proc. IEEE Conf. Control Appl., vol. 962, pp. 966–971.

Assaad, M., Boné, R., Cardot, H., 2005. Study of the Behavior of a New Boosting Algorithm for Recurrent Neural Networks Artificial Neural Networks: Formal Models and their Applications. Lect. Notes Comput. Sci., pp. 169–174.

Brockwell, P.J., Davis, R.A., 1991. Time Series: Theory and Methods. Springer-Verlag, New York.

Cao, L., 1997. Practical method for determining the minimum embedding dimension of a scalar time series. Physica D 110 (1–2), 43–50.

Chang, F.J., Chang, L.C., Huang, H.L., 2002. Real-time recurrent learning neural network for stream-flow forecasting. Hydrol. Process. 16, 2577–2588.

Chang, L.C., Chen, P.A., Chang, F.J., 2012. A reinforced two-step-ahead weight adjustment technique for on-line training of recurrent neural networks. IEEE Trans. Neural Netw. Learn. Syst. 23 (8), 1269–1278.

Chen, C.S., Chen, B.P.T., Chou, F.N.F., Yang, C.C., 2010. Development and application of a decision group Back-Propagation Neural Network for flood forecasting. J. Hydrol. 385 (1–4), 173–182.

Chiang, Y.M., Chang, L.C., Tsai, M.J., Wang, Y.F., Chang, F.J., 2010. Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and unguaged sites. Hydrol. Earth Syst. Sci. 14, 1309–1319.

Cho, K.H., Kao, S.S., Pachepsky, Y.A., Kim, K.W., Kim, J.H., 2011. Prediction of contamination potential of groundwater arsenic in Cambodia, Laos, and Thailand using artificial neural network. Water Res. 45 (17), 5535–5544.

Coulibaly, P., 2010. Reservoir computing approach to Great Lakes water level forecasting. J. Hydrol. 381, 76–88.

Coulibaly, P., Baldwin, C.K., 2005. Nonstationary hydrological time series forecasting using nonlinear dynamic methods. J. Hydrol. 307, 164–174.

Fraser, A.M., Swinney, H.L., 1986. Independent coordinates for strange attractors from mutual information. Phys. Rev. A 33 (2), 1134.

Hirasawa, K., Murata, J., Jinglu, H., Chunzhi, J., 2000. Universal learning network and its application to robust control. IEEE Trans. Syst., Man., Cybern. B, Cybern. 30, 419–430.

Jaeger, H., Haas, H., 2004. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304 (5667), 78–80.

Jothiprakash, V., Magar, R.B., 2012. Multi-time-step ahead daily and hourly intermittent reservoir inflow prediction by artificial intelligent techniques using lumped and distributed data. J. Hydrol. 450–451, 293–307.

Li, C.U., He, S.B., Liao, X.F., Yu, J.B., 2002. Using recurrent neural network for adaptive predistortion linearization of RF amplifiers. Int. J. RF Microwave Comput.-Aided Eng. 12, 125–130.

Liu, Q.S., Wang, J., 2008. A one-layer recurrent neural network with a discontinuous activation function for linear programming. Neural Comput. 20 (5), 1366–1383.

Liu, Q.S., Guob, Z.S., Wang, J., 2012. A one-layer recurrent neural network for constrained pseudoconvex optimization and its application for dynamic portfolio optimization. Neural Networks 26, 99–109.

Ma, Q.L., Zheng, Q.L., Peng, H., Zhong, T.W., Qin, J.W., 2008. Multi-step-prediction of chaotic time series based on co-evolutionary recurrent neural network. Chin. Phys. B 17 (2), 536–542.

MacKey, M.C., Glass, L., 1977. Oscillation and chaos in physiological control systems. Science 197 (4300), 287–289.

Muluye, G.Y., 2011. Improving long-range hydrological forecasts with extended Kalman filters. Hydrol. Sci. J. 56 (7), 1118–1128.

Nair, A.S., Jyh-Charn, L., Rilett, L., Gupta, S., 2001. Non-linear analysis of traffic flow. Proc. IEEE Conf. Intell. Trans. Syst., Oakland, CA, pp. 681–685.

Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models: 1. A discussion of principles. J. Hydrol. 10, 282–290.

Nayak, P.C., Sudheer, K.P., Rangan, D.M., Ramasastri, K.S., 2005. Short-term flood forecasting with a neurofuzzy model. Water Resour. Res. 41, W04004.

Nikolos, I.K., Stergiadi, M., Papadopoulou, M.P., Karatzas, G.P., 2008. Artificial neural networks as an alternative approach to groundwater numerical modelling and environmental design. Hydrol. Process. 22, 3337–3348.

Nourani, V., Sayyah Frad, M., 2012. Sensitivity analysis of the artificial neural network outputs in simulation of the evaporation process at different climatologic regimes. Adv. Eng. Softw. 47 (1), 127–146.

Nourani, V., Komasi, M., Almai, M.T., 2011. A hybrid wavelet-genetic programming approach to optimize ANN modelling of rainfall–runoff process. J. Hydrol. Eng. 17, 724–741.

Parlos, A.G., Rais, O.T., Atiya, A.F., 2000. Multi-step-ahead prediction using dynamic recurrent neural networks. Neural Networks 13, 765–786.

Seibert, J., 2001. On the need for benchmarks in hydrological modeling. Hydrol. Process. 15 (6), 1063–1064.

Serpen, G., Xu, Y., 2003. Simultaneous recurrent neural network trained with non-recurrent backpropagation algorithm for static optimisation. Neural Comput. Appl. 12, 1–9.

Shalev-Shwartz, S., Singer, Y., Ng, A.Y., 2004. Online and batch learning of pseudo-metrics. Proc. Twenty-First Int. Conf. Mach. Learn., Alberta, Canada, p. 94.

Takens, F., 1981. Detecting strange attractors in turbulence. Dynamical Systems and Turbulence. Lect. Notes Math. 898, 366–381.

Williams, R.J., Zipser, D., 1989. A learning algorithm for continually running fully recurrent neural networks. Neural Comput. 1, 270–280.

Xie, J.X., Cheng, C.T., Chau, K.W., Pei, Y.Z., 2006. A hybrid adaptive time-delay neural network model for multi-step-ahead prediction of sunspot activity. Int. J. Environ. Pollut. 28, 364–381.

Yong, S., Yibin, L., Qun, W., Caihong, L., 2010. Multi-steps prediction of chaotic time series based on echo state network. Proc. IEEE Conf. Bio-Inspired Comput.: Theor. Appl., Changsha, China, pp. 669–672.

Zabiri, H., Maulud, A., Omar, N., 2009. NN-based algorithm for control valve stiction quantification. WSEAS Trans. Syst. 4 (2), 88–97.