Available online at www.sciencedirect.com
ScienceDirect
Applied Soft Computing 8 (2008) 928–936
Applied Soft Computing
www.elsevier.com/locate/asoc

Using adaptive neuro-fuzzy inference system for hydrological time series prediction

Mohammad Zounemat-Kermani [a,*], Mohammad Teshnehlab [b]

[a] K.N. Toosi University of Technology, Department of Civil Engineering, Tehran, Iran
[b] K.N. Toosi University of Technology, Department of Electronic Engineering. Tehran, Iran

Received 13 December 2006; received in revised form 1 May 2007; accepted 28 July 2007
Available online 19 August 2007

Abstract

Conventionally, the multiple linear regression procedure has been known as the most popular models in simulating hydrological time series. However, when the nonlinear phenomenon is significant, the multiple linear will fail to develop an appropriate predictive model. Recently, intelligence system approaches such as artificial neural network (ANN) and neuro-fuzzy methods have been used successfully for time series modelling. In most instances for neural networks, multi layer perceptrons (MLPs) that are trained with the back-propagation algorithm have been used. The major shortcoming of this approach is that the knowledge contained in the trained networks is difficult to interpret. Using neuro-fuzzy approaches, which enable the information that is stored in trained networks to be expressed in the form of a fuzzy rule base, would help to overcome this issue. In the present study, a time series neuro-fuzzy model is proposed that is capable of exploiting the strengths of traditional time series approaches. The aim of this article is to investigate the potential of a neuro-fuzzy system with a Sugeno inference engine, considering different numbers of membership functions. Three rivers have been selected and daily prediction for them was applied. For better judgment, outcomes of the network have been compared to an autoregressive model.

Keywords: Neuro-fuzzy network; Sugeno fuzzy inference system; Time series prediction; River flow

1. Introduction

Time series forecasting has received tremendous attention of researchers in the last few decades. The time series forecasting methods have found applications in very wide areas including but not limited to finance and business, computer science, all branches of engineering, medicine, physics, chemistry and many interdisciplinary fields.

Conventionally, the researchers have employed traditional methods of time series analysis, modelling, and forecasting, e.g. Box–Jenkins methods of auto-regressive (AR), auto-regressive moving average (ARMA), auto-regressive integrated moving average (ARIMA), etc. The conventional time series modelling methods have served the scientific community for a long time; however, they provide only reasonable accuracy and suffer from the assumptions of stationery and linearity.

In the field of civil and hydraulic engineering, the demand on water resources has been rapidly rising as the nation implements its development programme to meet the increasing needs for irrigation, safe potable water, flood control and management, etc. So as to satisfy aforesaid requirements and other interrelated issues, a water programme needs to be modelled. One of the most significant desires around water modelling would tend to predictive models.

For instance, considering flood management, the information of a river flow discharge may be used to evaluate the performance of water resource planning and management. Therefore, how to accurately analyse the flow discharge based on measuring records at a specified flow domain becomes a crucial issue either from academic or practical standpoints.

The approach of intelligence systems networks has the advantages of using field recorded data directly without simplification and is not like regression analysis, which needs to make an assumption of equation in advance. Furthermore, this method is capable of executing parallel computations, and can simulate a nonlinear system which is hard to describe by traditional physical modelling. This has allowed for a wide

* Corresponding author. Present address: Department of civil engineering, K.N. Toosi university of technology, Tehran, Iran. Tel.: +98 913 3418165.
E-mail address: zounemat@computermail.net (M. Zounemat-Kermani).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2007.07.011

range of applications in water, hydraulic and hydrological engineering to be extensively published in recent years [1–6]. Therefore using artificial intelligence in related fields of study could be cited as estimation of scour depth around bridge piers [23,24] and/or ski jump bucket [22], rainfall and runoff modelling [25,27], estimation of evaporation [26], estimation of sediment load in rivers [18,19], prediction of river flow [17,20,21] and etc. This paper demonstrates the advantage of a neuro-fuzzy computing technique in modelling the prediction of flow from daily behaviour of river flow.

Three rivers have been selected for modelling. Rensselaer (New York), Brevard and De soto (Florida) in the USA. The prediction begins by generating the fuzzy rules that define the relationship between the input and output data. Thereafter, the optimization of the fuzzy rule set was done with an adaptive neuro-fuzzy inference system [7]. As a comparison, the performance of the neuro-fuzzy model relative to the linear regression analysis was also examined.

2. Neuro-fuzzy network

A neuro-fuzzy network consists of two major parts namely: (a) Neural Network and (b) Fuzzy system. Composition of these two parts would lead to a neuro-fuzzy network. This section is devoting to present and describe the basics of neuro-fuzzy system.

2.1. Artificial neural networks basics

The foundation of artificial neural network is referred to the modelling of neuron cells in the body. Each neuron consists of three parts which are the cell body, axon and dendrite. These three parts will make a neuron as a neural cell which has the ability to receive and send information. The operation of biological neurons (including data storage and analysis) is concealed under neurons aptitude and their relationships. The ability of modelling a biological neural network would lead us to design an artificial neural network. In general, a simple structure of an artificial neural network is shown in Fig. 1.

Input information are entered as parameter p, then input information are multiplied by some synaptical weight and adds to a bios b, which are consider as random numbers at first. Thereafter the calculated number is put in a function called f and after that the output amount a will be prescribed. Outputs of every layer could be considered as inputs for next layer. Artificial neural network consists one or more active layers and each layer includes several neurons. It is understandable that by increasing layers and neurons in each layer, network complication will enhance. At the end the produced output is compared with the target value, and then the amount of error will be calculated and spread through the variable values (w and b) of the network (Perceptron Neural Network). This process (network training) will be continued while the error amount is near to zero [8]. After that the network is called trained and can be utilized as an executive model.

2.2. Fuzzy systems

In the main, each fuzzy system consists of three main sections, Fuzzifier, Fuzzy data base and Defuzzifier. At first, input information is made as fuzzy data after bypassing the fuzzifier section. This operation is done by membership functions, in which the precise amount value becomes as fuzzy values by membership functions.

Later then, fuzzy parameters are entered to the fuzzy data base. Fuzzy data base includes two main sections, Fuzzy rule base, and inference engine. In fuzzy rule base, rules related to fuzzy propositions are described. Thereafter analysis operation is applied by fuzzy inference engine. There are several fuzzy inference engines which can be utilized for this purpose, which Sugeno and Mamdani are of the most important ones. The operator of each engine can be presumed minimum or product. Eqs. (1) and (2) show the Sugeno model with product and minimum operator.

$$\text{Output}\,(x) = \frac{\sum_{l=1}^{M} F^l(x)\left(\prod_{i=1}^{n} \mu A_i^l(x_i)\right)}{\sum_{l=1}^{M}\left(\prod_{i=1}^{n} \mu A_i^l(x_i)\right)} \quad (1)$$

$$\text{Output}\,(x) = \frac{\sum_{l=1}^{M} F^l(x)\left(\min_{i=1}^{n}(\mu A_i^l(x_i))\right)}{\sum_{l=1}^{M}\left(\min_{i=1}^{n}(\mu A_i^l(x_i))\right)} \quad (2)$$

$$F^l(x) = \alpha_0^l + \alpha_1^l x_1 + \alpha_2^l x_2 + \ldots + \alpha_n^l x_n \quad \text{(first order)}$$

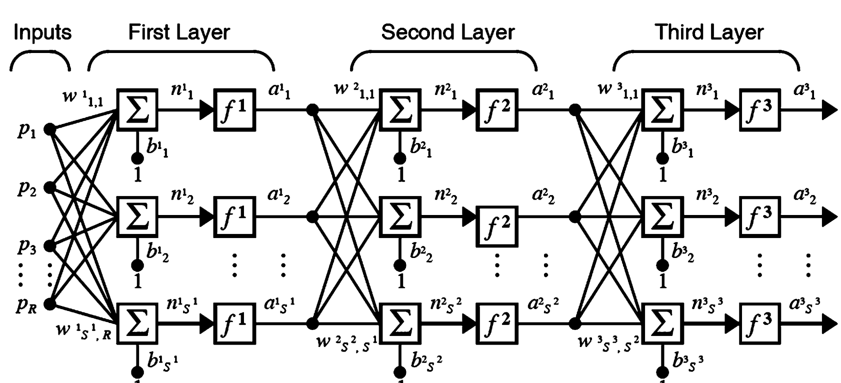

Fig. 1. Simple structure of an artificial neural network.

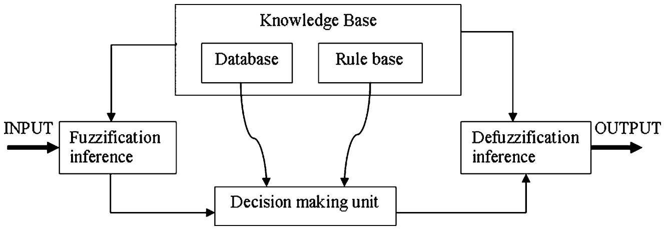

Fig. 2. Fuzzy inference system with crisp output.

where $F^l(x)$ is the first order of lth rule. Afterward output values are entered into the defuzzifier section and forward the categorical value. Fig. 2 shows a simple fuzzy system [9,10].

2.3. The adaptive neuro-fuzzy inference system (ANFIS)

The adaptive neuro-fuzzy inference system (ANFIS) uses the multi-valued logical system, namely, fuzzy logic, to account for a hidden imprecision in data and to make accurate mapping accordingly [9,10]. This is done by fuzzification of the input through membership functions, where a curved relationship maps the input value within the interval of [0–1]. The parameters associated with input as well as output membership functions are trained using an algorithm like back-propagation and/or least squares. Thus unlike the multi layer perceptron (MLP), where weights are tuned, in ANFIS fuzzy language rules or conditional (*if-then*) statements are determined in order to train the system. The general structure of the ANFIS is presented in Fig. 3. Selection of the fuzzy inference system (FIS), is the major concern when designing an ANFIS to model a specific target system. Various types of FIS are reported in the literature [11–13], and each is characterized by their consequent parameters only. The consequent part of this FIS is a linear equation and the parameters can be estimated by a simple least squares error method.

For instance, consider that the FIS has two inputs x and y and one output z. For the first order Sugeno fuzzy model, a typical rule set with two fuzzy if-then rules can be expressed as:

$$\text{Rule 1}: \quad \text{If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \quad \text{then } f_1 = p_1x + q_1y + r_1 \tag{3}$$

$$\text{Rule 2}: \quad \text{If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \quad \text{then } f_2 = p_2x + q_2y + r_2 \tag{4}$$

where A_1, A_2 and B_1, B_2 are the membership functions for inputs x and y, respectively; p_1, q_1, r_1 and p_2, q_2, r_2 are the parameters of the output function. Fig. 3a illustrates the fuzzy reasoning mechanism for this Sugeno model to derive an output function f from a given input vector [x,y].

The corresponding equivalent ANFIS architecture is presented in Fig. 3b, where nodes of the same layer have similar functions. The functioning of the ANFIS is as follows:

Layer 1: Each node in this layer generates membership grades of an input variable. The node output OP_i^l is defined by:

$$\mathrm{OP}_i^l = \mu_{Ai}(x) \quad \text{for } i = 1, 2 \text{ or} \tag{5}$$

$$\mathrm{OP}_i^l = \mu_{\mathrm{Bi}-2}(y) \quad \text{for } i = 3, 4 \tag{6}$$

where x (or y) is the input to the node; A_i (or $B_i - 2$) is a fuzzy set associated with this node, characterized by the shape of the membership functions in this node and can be any appropriate functions that are continuous and piecewise differentiable such as Gaussian, generalized bell shaped, trapezoidal shaped and triangular shaped functions. Assuming a generalized bell function as the membership function, the output OP_i^l can be computed as:

$$\mathrm{OP}_i^l = \mu_{Ai}(x) = \frac{1}{1 + (x - c_i/a_i)^{2b_i}} \tag{7}$$

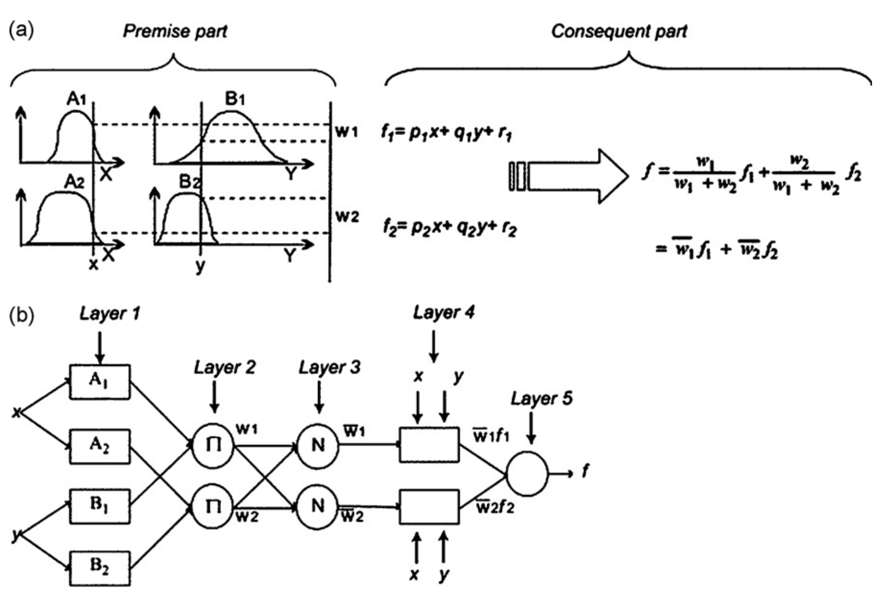

Fig. 3. (a) Sugeno's fuzzy if then rule and fuzzy reasoning mechanism; (b) equivalent ANFIS architecture.

Where $\{a_i, b_i, c_i\}$ is the parameter set that changes the shapes of the membership function with maximum equal to *1* and minimum equal to *0*.

Layer 2: Every node in this layer multiplies the incoming signals, denoted as π, and the output OP_i^l that represents the firing strength of a rule is computed as:

$$\mathrm{OP}_i^2 = w_i = \mu_{Ai}(x)\mu_{Bi}(y), \quad , i = 1, 2. \tag{8}$$

Layer 3: The *i*th node of this layer, labelled as *N*, computes the normalized firing strengths as:

$$\mathrm{OP}_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2. \tag{9}$$

Layer 4: Node *i* in this layer computes the contribution of the *i*th rule towards the model output, with the following node function:

$$\mathrm{OP}_i^4 = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \tag{10}$$

where $\bar{w}$ is the output of layer 3 and $\{pi, qi, ri\}$ is the parameter set.

Layer 5: The single node in this layer computes the overall output of the ANFIS as: [7,14].

$$\mathrm{OP}_i^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{11}$$

2.4. Hybrid algorithm

The parameters for optimization in an ANFIS are the premise parameters $\{a_i, b_i, c_i\}$, which describe the shape of the membership functions, and the consequent parameters $\{p_i, q_i, r_i\}$ which describe the overall output of the system. The learning algorithm for ANFIS is a hybrid algorithm, which is a combination of the gradient descent and least-squares method [7]. For simplicity, the adaptive network has only one output and is assumed to be as follows:

$$\text{Output} = F(\vec{I}, S) \tag{12}$$

where $\vec{I}$ is the set of input variables and *S* is the set of parameters. If there exists a function *H* such that the composite function $H{\cdot}F$ is linear in some of the elements of *S*, then these elements can be identified by the least-squares method. More formally, if the parameter set *S* can be decomposed into two sets:

$$S = S_1 \oplus S_2 \tag{13}$$

(where $\oplus$ represents direct sum) such that $H{\cdot}F$ is linear in the element S_2, then upon applying *H* to Eq. (12), we have:

$$H(\text{output}) = H \cdot F(\vec{I}, S) \tag{14}$$

which is linear in the elements of S_2. Now given values of elements of S_1, the *P* training data can be plugged into Eq. (14) and a matrix equation is obtained:

$$AX = B, \tag{15}$$

where *X* is an unknown vector whose elements are parameters in S_2. Let $|S_2| = M$, then the dimensions of *A*, *X* and *B* are $P \times M$, $M \times 1$ and $P \times 1$, respectively. Therefore *A* and *B* can refer to the input and target value matrixes. Since the number of training data pairs (*P*) is usually greater than the number of linear parameters *M* (number of linear parameters), this is an over-determined problem and generally there is no exact solution to Eq. (15). Instead, a least-squares estimate of *X* can be sought that minimizes the squared error $||AX - B||^2$.

Based on the ANFIS architecture shown in Fig. 3, we observe that the values of the premise parameters are fixed, and the overall output can be expressed as a linear combination of the consequent parameters. In symbols, the output *f* in Fig. 3 can be rewritten as:

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_2 \\ &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 \\ &\quad + (\bar{w}_2) r_2 \end{aligned} \tag{16}$$

which is linear in the consequent parameters (p_1, q_1, r_1; p_2, q_2, and r_2). As a result, the set of total parameters (*S*) can be separated into two such that $S_1 = a$ set of premise parameters and $S_2 =$ a set of consequent parameters. Consequently the hybrid-learning algorithm can be used for an effective search of the optimal parameters of the ANFIS. More specifically, in the forward pass of the hybrid-learning algorithm, node outputs go forward until layer (4) and the consequent parameters are identified by the least-squares method. In the backward pass, the error signals propagate backward and the premise parameters are updated by gradient descent. As mentioned earlier, the consequent parameters thus identified are optimal under the condition that the premise parameters are fixed. Accordingly, the hybrid approach converges much faster since it reduces the dimension of the search space of the original back-propagation method [7,14].

3. Model development

In this section, general information about the applied model is explained.

3.1. Using neuro-fuzzy so as to nonlinear time series predictions

In the present study, flow prediction of three rivers (Brevard, De Soto in Florida state and Rensselaer in New York State) have been studied and investigated. Ten years daily stream flows (from 1992 to 2002) were chose as input data for neuro-fuzzy network. Also in order to test and model verification, the daily streamflow of each river in 2003 year was considered. Input data were entered into the neuro-fuzzy network after normalization. For this purpose, Eq. (17) was utilized.

$$z = \left(\frac{(x - \bar{x})}{\sigma}\right) \tag{17}$$

While *z* presents the normalized data, *x* is referred to the original data, $\bar{x}$ stands for the average of data and σ shows the standard deviation of data. In this case all of the input data would get values between zero and one, which is proper for network training. Output results will be prescribed as predicted values after denormalization.

3.2. The employed neuro-fuzzy network

In the present investigation Sugeno model with product operator was opted for the executive fuzzy inference system. Also two types of neuro-fuzzy networks with different membership functions were applied. In the first network, input data were considered as flow discharge at the current time and previous time ($Q(t)$,$Q(t-1)$),whereas in the second network, three depending values were considered as input data, namely the current time, previous time and a step before previous time ($Q(t)$,$Q(t-1)Q(t-2)$).

In each network, the amounts of membership functions were considered *2*, *3*, *4*, *6* and *8* within five network statuses. The entire membership functions were presumed once as Gaussian and once as triangular shaped. Considering the foresaid comments, the applied fuzzy rule base has two inputs of $Q(t)$,$Q(t-1)$ as for x,y concerning the first mentioned network, and has three inputs of $Q(t)$,$Q(t-1)Q(t-2)$ as for x,y,z relating to the second network (see Section 2.3). Attained outcomes were induced as the final results in the diffuzification part according to the relation (11).

It is evident that 10 different networks were executed and after that the obtained results were assessed which are presented in the next sections.

3.3. Autoregressive model

An autoregressive model (AR), has been considered in this study. For model application, data were normalized (Eq. (17)) to have a mean of 0.0 and variance of 1.0 before exploring for the autocorrelation structure.

The structure of the AR model can be represented by the following equation:

$$Q(t)=\sum_{i=1}^{p}\alpha(i)Q(t-i)+\varepsilon(t) \tag{18}$$

where $Q(t)$ is the daily streamflow being modelled, $Q(t-i)$ stands for the past stream flow, α is the auto-regressive parameters to be determined, P the order of the AR process, I an index representing the order of the AR process and $\varepsilon(t)$ a random variable and t is an index representing time (day in this case). The AR parameters can be obtained by Yule Walker equations [15]. After obtaining AR parameter it is possible to validate model by computing the performance statistics during both calibration and testing data sets.

3.4. Performance evaluation

Three different types of standard statistical were considered as statistical performance evaluation. The correlation coefficients (R), root mean square error (RMSE) and mean absolute percentage error (MAPE) were used. The three performance evaluation criteria used in this study can be calculated utilizing the following equations.

$$R=\left[\frac{\sum_{i=1}^{n}\left(Q_i^o-\bar{Q}^o\right)\left(Q_i^p-\bar{Q}^p\right)}{\sqrt{\sum_{i=1}^{n}(Q_i^o-\bar{Q}^o)\sqrt{\sum_{i=1}^{n}\left(Q_i^p-\bar{Q}^p\right)}}}\right] \tag{19}$$

$$\mathrm{RMSE}=\sqrt{\frac{\sum_{i=1}^{n}(Q_i^o-Q_i^p)}{n}} \tag{20}$$

$$\mathrm{MAPE}=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{Q_i^p-Q_i^o}{Q_i^o}\right|\times 100 \tag{21}$$

where, Q_i^o is the observed streamflow at time t, and Q_i^p is the predicted stream flow at time t.

4. Case study

4.1. Study area and data set

Gathered data for this study was collected from USGS Internet site [16]. Data investigation of three aforementioned rivers shows that a great disorder and variance dominates the observed data. For instance, flow behaviour of De Soto, Brevard and Rensselaer rivers through 2000–2003 is shown in Fig. 4.

Statistical parameters interrelated to three aforesaid rivers are provided in Table 1.

As it is rises from Table 1, the amount of standard deviation and difference between maximum and minimum is considerable, therefore modelling of the rivers would need a complicated and flexible operation.

4.2. Results and discussion

In order to assess obtained results, network was run in various manners, and it is available to evaluate results by R, RMSE and MAPE. Results are tabulated in Tables 2–4 for abovementioned Rivers. It is worth noting that, the presented

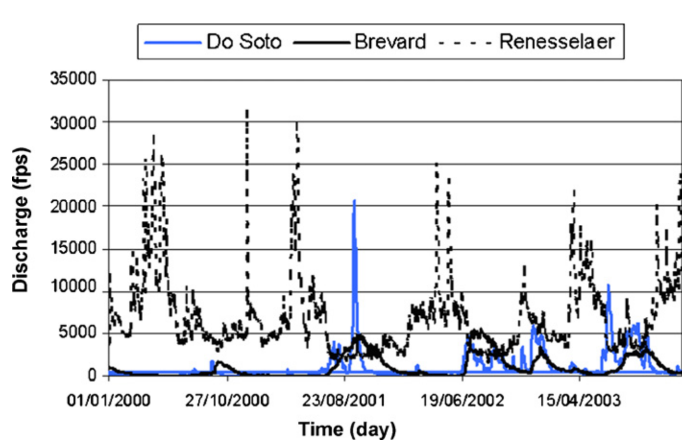

Fig. 4. Behaviour of De Soto, Brevard and Rensselaer rivers through 2000–2003.

Table 1
Statistical parameters interrelated to rivers within 10 years (1992–2002)

Statistical parameters	De Soto		Brevard		Rensselaer	
	1992–2002	2003	1992–2002	2003	1992–2002	2003
Average (fps)	1005.7	1856.3	1165.0	1202.3	6542.5	7686.2
Standard deviation (fps)	1634.0	2055.4	1374.6	949.9	4814.1	4343.8
Minimum amount (fps)	5.6	153	0	0	1300	2600
Maximum amount (fps)	20070	10700	7560	3450	41200	24000
Ratio of average to standard deviation (fps)	0.61	0.90	0.87	1.26	1.35	1.77

Table 2
Error analysis of flow prediction in De Soto river in 2003

Row	Different input manners	Numbers of membership functions	MAPE	RMSE	R
1	$Q(t)$, $Q(t-1)$	2	47.82	16.67	0.9962
2	$Q(t)$, $Q(t-1)$	3	22.77	8.60	0.9986
3	$Q(t)$, $Q(t-1)$	4	17.64	17.90	0.9867
4	$Q(t)$, $Q(t-1)$	6	11.17	6.61	0.9988
5	$Q(t)$, $Q(t-1)$	8	14.83	9.47	0.9966
6	$Q(t)$, $Q(t-1)$, $Q(t-2)$	2	7.07	11.66	0.9952
7	$Q(t)$, $Q(t-1)$, $Q(t-2)$	3	6.03	7.38	0.9994
8	$Q(t)$, $Q(t-1)$, $Q(t-2)$	4	10.05	8.802	0.9987
9	$Q(t)$, $Q(t-1)$, $Q(t-2)$	6	12.21	12.07	0.9961
10	$Q(t)$, $Q(t-1)$, $Q(t-2)$	8	17.78	15.06	0.9907

Table 3
Error analysis of flow prediction in Rensselaer river in 2003

Row	Different input manners	Numbers of membership functions	MAPE	RMSE	R
1	$Q(t)$, $Q(t-1)$	2	12.05	59.29	0.9868
2	$Q(t)$, $Q(t-1)$	3	7.29	37.11	0.9952
3	$Q(t)$, $Q(t-1)$	4	5.40	28.20	0.9992
4	$Q(t)$, $Q(t-1)$	6	4.38	28.53	0.9972
5	$Q(t)$, $Q(t-1)$	8	4.54	31.47	0.9943
6	$Q(t)$, $Q(t-1)$, $Q(t-2)$	2	24.31	104.1	0.9071
7	$Q(t)$, $Q(t-1)$, $Q(t-2)$	3	13.97	81.44	0.9348
8	$Q(t)$, $Q(t-1)$, $Q(t-2)$	4	10.75	66.22	0.9573
9	$Q(t)$, $Q(t-1)$, $Q(t-2)$	6	7.52	43.46	0.9852
10	$Q(t)$, $Q(t-1)$, $Q(t-2)$	8	8.78	45.24	0.9845

results are briefed for only Gaussian membership function as it had slightly better results (about 3%) than triangular membership functions.

4.2.1. De Soto river results

Obtained results for De Soto river for flow prediction in 2003, are presented in Table 2 in different manners.

Fig.'s 5 curves illustrate the predicted values using neuro-fuzzy network and observed data in 2003 year (graph has been prepared on the basis on 7th row in Table 2).

4.2.2. Rensselaer river results

The attained prediction results for Rensselaer river are presented in Table 3.

Table 4
Error analysis of flow prediction in Brevard river in 2003

Row	Different input manners	Numbers of membership functions	MAPE	RMSE	R
1	$Q(t)$, $Q(t-1)$	2	32.87	7.43	0.9976
2	$Q(t)$, $Q(t-1)$	3	20.43	4.76	0.9981
3	$Q(t)$, $Q(t-1)$	4	10.17	3.11	0.9985
4	$Q(t)$, $Q(t-1)$	6	7.44	2.85	0.9990
5	$Q(t)$, $Q(t-1)$	8	10.97	3.01	0.9989
6	$Q(t)$, $Q(t-1)$, $Q(t-2)$	2	6.87	2.54	0.9993
7	$Q(t)$, $Q(t-1)$, $Q(t-2)$	3	5.83	2.31	0.9998
8	$Q(t)$, $Q(t-1)$, $Q(t-2)$	4	5.89	2.38	0.9997
9	$Q(t)$, $Q(t-1)$, $Q(t-2)$	6	8.64	2.96	0.9991
10	$Q(t)$, $Q(t-1)$, $Q(t-2)$	8	15.34	3.68	0.9987

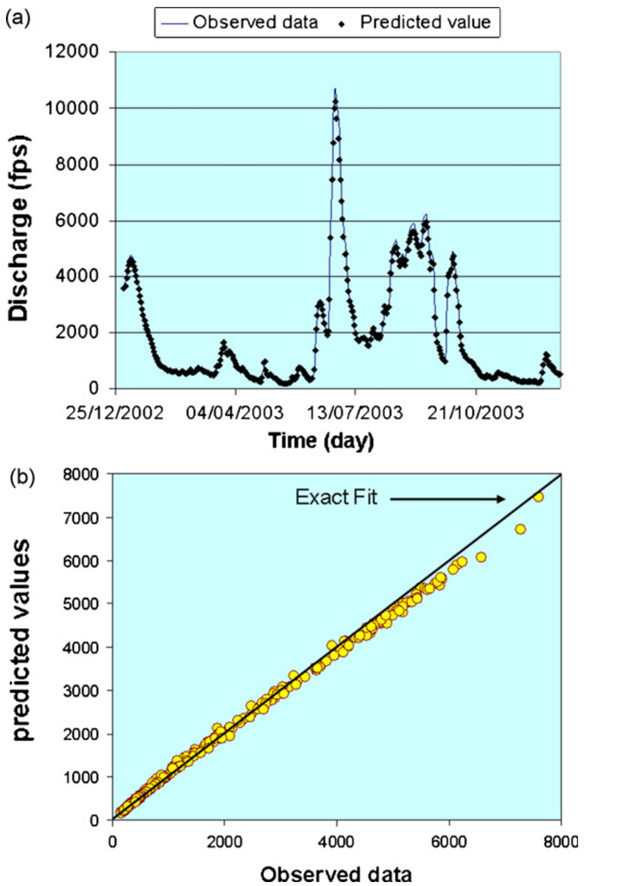

Fig. 5. Comparison between obtained results and observed data in De Soto river in 2003. (a) Behaviour of stream flow, (b) comparing obtained and observed values.

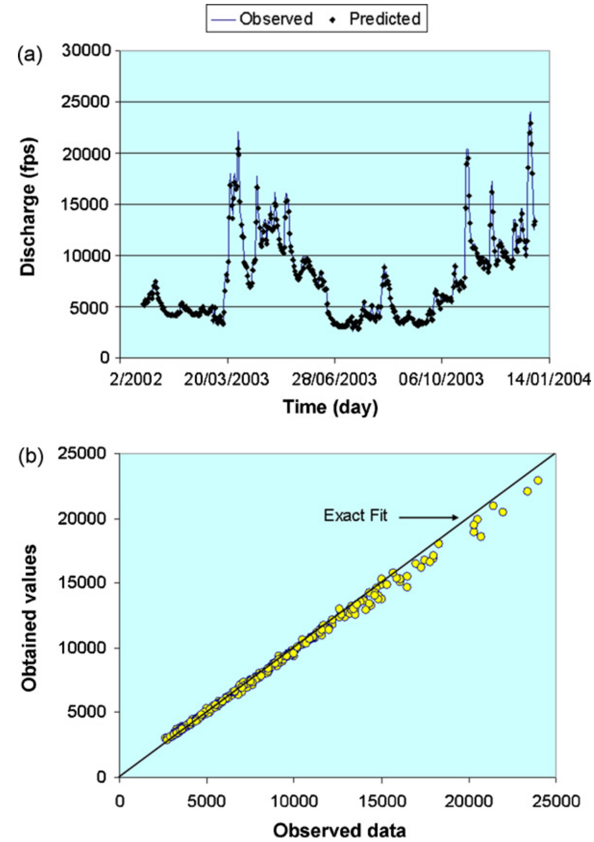

Fig. 6. Comparison between obtained results and observed data in Rensselaer river in 2003. (a) Behaviour of stream flow, (b) comparing obtained and observed values.

Fig.'s 6 curves show the predicted values using neuro-fuzzy network and observed data in 2003 year (graph has been prepared on the basis on 4th row in Table 3).

4.2.3. *Brevard river results*

The achieved prediction results for Rensselaer river are presented in Table 4.

Fig.'s 7 curves shows the predicted values using neuro-fuzzy network and observed data in 2003 year (graph has been prepared on the basis on 7th row in Table 4).

4.3. *Comparison with AR model results and result analysis*

The results in terms of various performance evaluation measures are presented in Table 5. In order to assess the ability of neuro-fuzzy models relative to an AR model, the AR results are tabulated in comparison with ANFIS results.

It is clear that the applied ANFIS model provides better results rather than AR model. As rises from the comparison of the obtained results of three rivers, by decreasing the ratio of average to standard deviation, the calculated error (MAPE), in neuro-fuzzy network shows an increment that is because of the higher disorder in the observed data. It is remarkable, that by increasing in number of membership functions, the calculated error has an inferior error manner firstly, and thereafter gets a higher manner. The reason of error rise is due to immense increasing in the network nonlinearity. So the proper number of membership functions should be considered (which commonly demonstrates six membership functions for time series prediction with $Q(t)Q(t-1)$ input structure).

In addition, analysis of three rivers demonstrates that there is not a specific neuro-fuzzy model structure which could give the best result. For instance the most excellent results for De Soto river is refereed to a neuro-fuzzy network with $Q(t)$, $Q(t-1)$, $Q(t-2)$ input structure and three membership functions, whilst Rensselaer river had the minimum error on the $Q(t)$, $Q(t-1)$ structure and six membership functions. But in general, it could be mentioned (regarding to Table 1) that by data disorder increment (lower amount of average to standard division ratio),

Table 5
performance indices for ANFIS and AR models

River	Model	Calibration			Validation		
		MAPE	RMSE	*R*	MAPE	RMSE	*R*
Do Soto	ANFIS	5.94	7.12	0.9997	6.03	7.38	0.9994
	AR	32.8	15.3	0.9870	36.2	16.8	0.9832
Rensselaer	ANFIS	4.12	26.9	0.9976	4.38	28.53	0.9972
	AR	24.3	60.1	0.9823	26.7	63.4	0.9801
Brevard	ANFIS	5.23	2.23	0.9998	5.83	2.31	0.9998
	AR	18.65	5.86	0.9983	20.17	5.98	0.9989

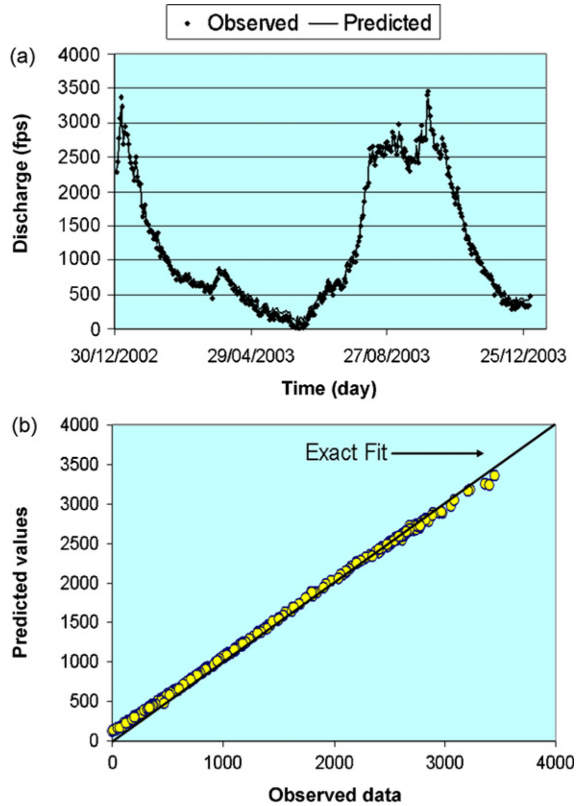

Fig. 7. Comparison between obtained results and observed data in Rensselaer river using neuro-fuzyy network in 2003. (a) Behaviour of stream flow, (b) comparing obtained and observed values.

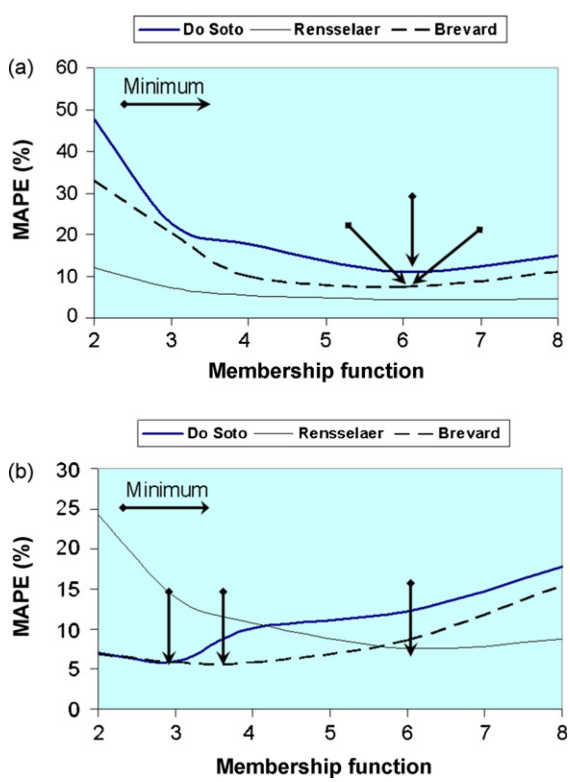

Fig. 8. Evaluation performance for rivers depends on number of membership functions. (a) $Q(t)$, $Q(t-1)$ input structure; (b) $Q(t)$, $Q(t-1)Q(t-2)$ input structure.

better outcomes might appear by increasing membership functions rather than increasing dependent input data ($Q(t) \ldots Q(t-n)$) to the network. Fig. 8 depicts the MAPE (which could be considered as evaluation parameter for three rivers) versus number of membership functions.

5. Conclusion

Using neuro-fuzzy for time series predictions could be presented as the one of foremost methods. In the presented article an adaptive neuro-fuzzy inference system was applied for predicting daily behaviour of river flow. Two types of neuro-fuzzy networks were considered regarding to the inputs variables pattern and each network was executed with 2, 3, 4, 6 and 8 membership functions. For attaining this purpose 10 years daily flow of three different rivers data have been gathered and used so as to training network, and 1 year daily behaviour of river flow was applied as the test data and model validation. The network was run in various statuses and the outcomes were compared to the observed data. Results show that a regular structure for the neuro-fuzzy network could not be mentioned as an unsurpassed network. Moreover, comparing the attained results of neuro-fuzzy network and an autoregressive (AR) model shows the superiority of the neuro-fuzzy model in river flow prediction relative to conventional methods.

References

[1] H. Adeli, Neural network in civil engineering: 1989–2000, Comput.-Aided Civ. Infrastructure Eng. 16 (2001) 126–142.
[2] R. Baratti, B. Cannas, A. Fanni, M. Pintus, G.M. Sechi, N. Toreno, River flow forecast for reservoir management through neural networks, Neurocomputing 55 (2003) 421–437.
[3] L. Bodri, V. Cermik, Prediction of extreme precipitation using a neural network: application to summer flood occurrence in Moravia, Adv. Eng. Software 31 (2000) 311–321.
[4] F.J. Chang, Y.C. Chen, A counter propagation fuzzy-neural network modeling approach to real-time streamflow prediction, J. Hydrol. 245 (2001) 153–164.
[5] R.H. Maier, T. Sayed, B.J. Lence, Forecasting cyanobacterium Anabanea spp. In the river Murray, South Australia, Using B-spline neurofuzzy models, Ecol. Modell. (2001) 85–96.
[6] B. Bazartseren, G. Hildebrandt, K.P. Holz, Short-term water level prediction using neural networks and neuro-fuzzy approach, Neurocomputing 55 (3–4) (2003) 439–450.
[7] M. Aqil, I. Kita, A. yano, S. Nishiyama, Analysis and prediction of flow from local source in a river basin using a neuro-fuzzy modelling tool, J. Environ. Manage. 85 (1) (2007) 215–223.
[8] I. Flood, N. Kartam, Neural networks in civil engineering I: principles and understanding, J. Comput. Civ. Eng., ASCE 8 (2) (1994) 131–148.
[9] J. Jantzen, Neurofuzzy Modelling, Technical University of Denmark, 2005.
[10] O. Wolkenhauer, Fuzzy Mathematics in Systems Theory and Data Analysis, John Wiley & Sons, Inc., 2001.
[11] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man-Machine Stud. 7 (1) (1975) 1–13.
[12] Y. Tsukamoto, An approach to fuzzy reasoning method, in: M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.), Advances in Fuzzy Set Theory and Application, North-Holland, Amsterdam, 1979, pp. 137–149.
[13] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control., IEEE Trans. Syst. Man Cybernetics 15 (1998) 116–132.

[14] J.-S.R. Jang, C.-T. Sun, Neuro-fuzzy modelling and control, in: Proceedings IEEE 83, 1995, pp. 378–406.
[15] J.D. Salas, J.W. Delleur, V. Yevjevich, W.L. Lane, Applied Modelling of Hydrologic Time Series, Water research publication, Littleton, CO, USA, 1997.
[16] US Geological Survey, Water Resources of the United States. http://water.usgs.gov.
[17] P. Singh, M.C. Deo, Suitability of different neural networks in daily flow forecasting, Appl. Soft Computing 7 (3) (2007) 968–978.
[18] M. Alp, H.K. Cigizoglu, Suspended sediment load simulation by two artificial neural network methods using hydrometeorological data, Environ. Modell. Software 22 (1) (2007) 2–13.
[19] H.K. Cigizoglu, M. Alp, Generalized regression neural network in modelling river sediment yield, Adv. Eng. Software 37 (2) (2006) 63–68.
[20] G. Tayfur, V.P. Singh, Predicting longitudinal dispersion coefficient in natural streams by artificial neural network, J. Hydr. Energy (2005) 131–991.
[21] R.P. Deka, V. Chandramouli, J. Fuzzy Neural Network Model Hydrologic Flow Hydrologic Energy 10 (2005) 302–316.
[22] H. Md. Azmathullah, M.C. Deo, P.B. Deolalikar, Neural networks for estimation of scour downstream of a ski-jump bucket, J. Hydr. Energy (2005) 131–145.
[23] S.M. Bateni, S.M. Borghei, D.S. Jeng, Neural network and neuro-fuzzy assessments for scour depth around bride piers, Eng. Appl. Artif. Intelligence 20 (3) (2007) 401–414.
[24] S.M. Bateni, D.S. Jeng, B.W. Melville, Bayasian neural networks for prediction of equilibrium and time-dependent scour depth around bridge piers, Adv. Eng. Software 38 (2007) 102–111.
[25] S. Riad, J. Mania, Rainfall-runoff model using an artificial neural network approach, Math. Comput. Modell. (2004) 839–846.
[26] Ö. Terzi, M.E. l Keskin, E.D. Taylan, Estimating Evaporation Using ANFIS, J. Irrigation Drainage Eng., ASCE (2006) 503–507.
[27] E. Toth, A. Brath, A. Montanari, Modeling of the daily rainfall-runoff relationship with artificial neural network, J. Hydrol. 239 (2000) 132–147.