Soft Comput (2008) 12:183–190
DOI 10.1007/s00500-007-0186-7

FOCUS

Linguistic time series forecasting using fuzzy recurrent neural network

R. A. Aliev · B. Fazlollahi · R. R. Aliev · B. Guirimov

Published online: 31 May 2007

Abstract It is known that one of the most spread forecasting methods is the time series analysis. A weakness of traditional crisp time series forecasting methods is that they process only measurement based numerical information and cannot deal with the perception-based historical data represented by linguistic values. Application of a new class of time series, a fuzzy time series whose values are linguistic values, can overcome the mentioned weakness of traditional forecasting methods. In this paper we propose a fuzzy recurrent neural network (FRNN) based time series forecasting method for solving forecasting problems in which the data can be presented as perceptions and described by fuzzy numbers. The FRNN allows effectively handle fuzzy time series to apply human expertise throughout the forecasting procedure and demonstrates more adequate forecasting results. Recurrent links in FRNN also allow for simplification of the overall network structure (size) and forecasting procedure. Genetic algorithm-based procedure is used for training the FRNN. The effectiveness of the proposed fuzzy time series forecasting method is tested on the benchmark examples.

Keywords Fuzzy time series · Fuzzy recurrent neural network · Genetic algorithm

R. A. Aliev (✉) · B. Guirimov
Azerbaijan State Oil Academy, Baku, Azerbaijan
e-mail: raliev@iatp.az

B. Fazlollahi
Georgia State University, Atlanta, USA
e-mail: dscbbf@langate.gsu.edu

R. R. Aliev
Eastern Mediterranean University, Mersin, Turkey
e-mail: rashad.aliyev@edu.emu.tr

1 Introduction

Forecasting activities on the basis of prediction result of time series play an important role in different areas of human activity, including weather forecasting, economic and business planning, inventory and production control, etc. Many available data in such real-world systems, where a human plays the basic decision maker role, are linguistic values or words with fuzzy meaning. The main advantage of using fuzzy approach is to apply human expertise throughout the forecasting procedure. This type of time series significantly differs from traditional time series and the methods of the latter are not applicable in this case (Zuoyoung et al. 1998). So we deal with a new class of time series, a fuzzy time series, whose values are linguistic values. There are several approaches to modeling fuzzy time series.

In Song and Chissom (1993a,b, 1994), Sullivan and Woodall (1994), fuzzy time series models are proposed and their applications to forecasting problems are considered. They have proposed several definitions and theorems related to some properties of fuzzy time series. Some extension of fuzzy relational model based time series is analyzed in Tsai and Wu (2000). Applications of fuzzy time series are considered in Hwang et al. (1996), Tsai and Wu (1999). Modification of fuzzy time series models for forecasting of university enrollments is discussed in Hwang et al. (1998a).

In Castillo and Melin (2001) a fuzzy fractal method for forecasting financial and economic time series is described. All the works on fuzzy time series mentioned above are based on fuzzy relational equations (Zadeh 1975; Nikravesh et al. 2004; Pedrycz 1989, 1991). The forecasting methods based on fuzzy relational equations suffer from a number of shortcomings among which the main ones are computational complexity, difficulty of choosing an optimal or near-optimal fuzzy implication, difficulty of training and adaptation of the rule base (relational matrices) and its parameters and others. These drawbacks lead to difficulties in reaching the desirable degree of forecasting accuracy. Application of fuzzy neural network for time series forecasting can overcome these weaknesses (Aliev et al. 2004). Note also that recurrent neural networks are found to be very effective and efficient technique to use in many dynamic or time series related applications. In this paper we propose a FRNN based fuzzy time series forecasting method which allows effectively handle fuzzy time series to apply human expertise throughout the forecasting procedure. This method is characterized by the less computational complexity, learning by experiments, and adaptability.

Using benchmark tests, it will be shown that the forecasting error of this method is significantly smaller than that of existing fuzzy time series approaches (Chen and Hwang 2000; Hwang et al. 1998a; Song and Chissom 1993a,b, 1994; Tsai and Wu 2000).

The paper is structured as follows. Next Sect. 2 gives the statement of forecasting problem in which the historical data are represented by perception based information. Section 3 discusses the solution of the considered problem on the basis of fuzzy recurrent neural network trained by a genetic algorithm based technique. Section 4 deals with computational experiments. Section 5 concludes the paper.

2 Statement of problem

Given a series of values of some business or economic variable, measured at successive equal intervals of time (for example, quarterly sales over a few years, daily measured temperature values in a region, etc. make up a time series), it is necessary to forecast the values in the future, i.e., for the next year, quarter, or day. Note that the considered data are fuzzy numbers or linguistic values with specified membership functions.

Suppose that at various time instant t we are presented with perception based information y_t described by fuzzy sets. Formally, the fuzzy n-order time series problem can be represented as:

$$y_{t+1} = F(y_t, y_{t-1}, \dots, y_{t-n+1}), \tag{1}$$

where F is fuzzy set valued mapping of values $y_t, y_{t-1}, \dots, y_{t-n+1} \in \varepsilon^n$ from ε^n into $y_{t+1} \in \varepsilon^1$ to be estimated, ε^n and ε^1 are spaces of fuzzy sets, y_t is fuzzy valued data at time interval t, y_{t+1} is fuzzy valued forecasted value for time interval $t+1$.

The existing fuzzy time series forecasting models widely suggested in the literature, e.g., in Song and Chissom (1993a, 1994), for calculating y_{t+1}, replace the Eq. (1) by:

$$y_{t+1} = (y_t \times y_{t-1} \times \cdots \times y_{t-n+1}) \circ R(t, t-n+1), \tag{2}$$

where $\times$ is the Cartesian product, $\circ$ is Max–Min composition operator, and R is fuzzy relation. The solution of (2) is connected with high time consuming and requirements of choice of optimal or near-optimal fuzzy implication, membership function shape, and a number of others, which significantly complicate the task, especially, in the case of high order fuzzy time series.

Fuzzy neural networks are universal approximators (Aliev et al. 2004; Liu and Li 2004), and hence can be used to construct fuzzy set valued mapping F in (1). In a simple case, we need a FNN with one hidden layer, n input and one output nodes which can express the relationships:

$$\hat{y}_{t+1} = \hat{F}_{NN}(y_t, y_{t-1}, \dots, y_{t-n+1}), \tag{3}$$

where an estimate $\hat{F}_{NN}$ for F is constructed from a large class of fuzzy neural network based mappings. In its turn $\hat{F}_{NN}$ is determined by fuzzy weights of neuron connections, fuzzy biases, and neuron activation functions. The problem is in adjusting the weights vectors to minimize a cost function $E(F, \hat{F}_{NN})$ (for instance, as fuzzy hamming distance), defined on the basis of (1) and (3) and representing distance measure between the fuzzy neural network output and the desired output pattern. The type and architecture of the neural network, training procedure, and network validation are considered in the next section.

3 Architecture and training of fuzzy recurrent neural network

The structure of the fuzzy recurrent neural network for the realization of (3) is presented in Fig. 1. The box elements represent memory cells that store values of activation of neurons at previous time step, which is fed back to the input at the next time step.

In general, the network may have virtually any number of layers. We number the layers successively from 0 (the first or input layer) to L (last or output layer). The neuron in the first (layer 0) layer is only distributing the input signal without modifying the values.

$$z^0(t) = x^0(t) \tag{4}$$

The neurons in the layers 1 to layer N_L are dynamic and compute their output signals as follows:

$$z_i^l(t) = F\left(\theta_i^l + \sum_j x_j^l(t) w_{ij}^l + \sum_j z_j^l(t-1) v_{ij}^l\right), \tag{5}$$

where $x_j^l(t)$ is j-th fuzzy input to the neuron i at layer l at the time step t, $z_i^l(t)$ is the computed output signal of the neuron at the time step t, w_{ij} is the fuzzy weight of the connection to neuron i from neuron j located at the previous layer, θ_i is the fuzzy bias of neuron i, and $z_j^l(t-1)$ is the activation of neuron j at the time step $(t-1)$, v_{ij} is the recurrent

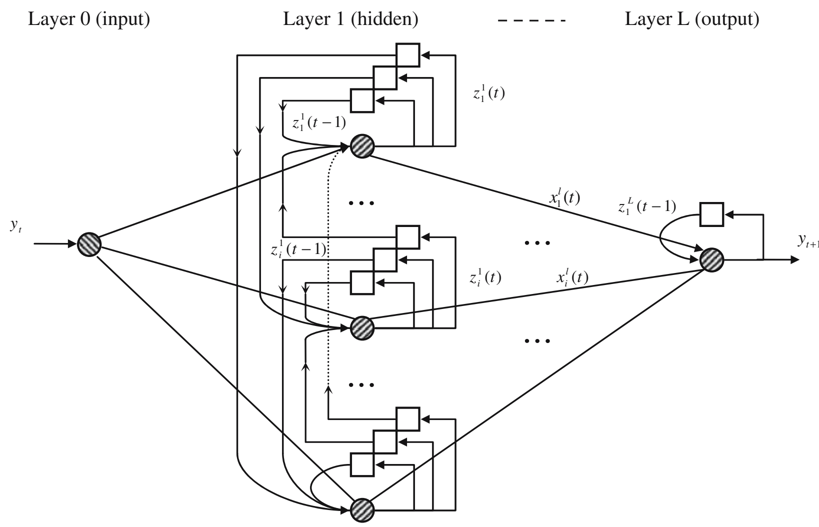

Fig. 1 The structure of a simple FRNN

connection weight to neuron i from neuron j at the same layer.

Note that regarding the considered time series forecasting problem, the FRNN input $x^0(t)$ will represent the time series element y_t and the FRNN output z^L will represent the time series element y_{t+1}.

The activation F for a total input to the neuron s is calculated as

$$F(s) = \frac{s}{1+|s|} \tag{6}$$

So, the output of neuron i at layer l is calculated as follows:

$$z_i^l(t) = \frac{\theta_i^l + \sum_j x_j^l(t) w_{ij}^l + \sum_j z_j^l(t-1) v_{ij}^l}{1 + \left|\theta_i^l + \sum_j x_j^l(t) w_{ij}^l + \sum_j z_j^l(t-1) v_{ij}^l\right|} \tag{7}$$

All fuzzy signals and connection weights and biases are general fuzzy numbers that with any required precision can be represented as

$$T(L_0, L_1, \ldots, L_{n-1}, R_{n-1}, R_{n-2}, \ldots R_0),$$

where $[L_i, R_i](i = \overline{0, (n-1)})$ are n intervals that approximate the shape of membership function (e.g., in case of $n = 2$, we get the traditional trapezoidal numbers).

In case the original learning patterns are crisp, we need to sample data into fuzzy terms, i.e., to fuzzify the learning patterns. The fuzzifiers can be created independently for specific problems.

In spite of great importance of fuzzy feed-forward and recurrent neural networks for solving wide range of real-world problems, today there are no effective training algorithm for them. Currently there are two approaches for training of FNN. First approach is based on application of the level-sets of fuzzy numbers and the back-propagation (BP) algorithm. The second approach involves genetic algorithms (GA) to minimize error function and determine the fuzzy connection weights and biases. The method based on the second approach was proposed in Aliev et al. (2001). In contrast to the BP and other supervised learning algorithms, GAs don't require or use derivative information, and hence, they are most effective in case where the derivative is very difficult to obtain or even unavailable. In this paper we use genetic algorithm-based learning algorithm for FRNN with fuzzy inputs, fuzzy weights and biases, and fuzzy outputs suggested in Aliev et al. (2006).

To apply genetic algorithm based approach for FRNN training, all adjustable parameters i.e., connection weights and biases are coded as bitstrings. A combination of all weight and bias bitsrings compose a genome representing a potential solution to the problem.

The selection of best genomes from the population is done on the basis of the genome fitness value, which is calculated from the FRNN error performance index. The calculation of the fitness value of a particular genome require restoration of the coded genome bits back to fuzzy weight coefficients and biases of FRNN, in other words, we need to get a phenotype from the genotype.

The FRNN error performance index can be calculated as follows:

$$E_{tot} = \sum_p D(y_p, y_p^{des}), \tag{8}$$

where E_{tot} is the total error performance index for all learning data entries p.

We shall assume Y is a finite universe $Y = \{y_1, y_2, \ldots, y_n\}$; D is an error function such as the distance measure between two fuzzy sets, the desired y_p^{des} and the computed y_p outputs. The efficient strategy is to consider the difference of all the points of the used general fuzzy number. The considered distance metrics is based on Hamming distance

$$D(T1, T2) = \sum_{i=0}^{i=n-1} k_i |L_{T1i} - L_{T2i}| + \sum_{i=0}^{i=n-1} k_i |R_{T1i} - R_{T2i}|, \quad (9)$$

where:

$D(T_1, T_2)$ is the distance measure between two fuzzy numbers $T_1(y_p^{des})$ and $T_2(y_p)$;

$0 \le k_0 \le k_1 \cdots \le k_{n-2} \le k_{n-1}$ are some scaling coefficients.

Once the total error performance index for a combination of weights has been calculated the fitness f of the corresponding genome is set as:

$$f = \frac{1}{1 + E_{tot}} \quad (10)$$

The learning may be stopped once we see the process does not show any significant change in fitness value during many succeeding regenerations. In this case we can specify new mutation (and maybe crossover) probability and continue the process. If the obtained total error performance index or the behavior of the obtained network is not desired, we can restructure the network by adding new hidden neurons, or do better sampling (fuzzification) of the learning patterns.

The genomes in the current population undergo specific genetic operators, which leads to a change in population: new child genomes, often called offsprings, are produced. To rank these genomes, their fitness values are calculated. To do this first they are converted into the network representation and the network error is calculated, and then formula (10) is applied to calculate the fitness.

During the selection processes low fitness genomes have low probability to survive and be saved into a new population for participation in future reproduction. The process is repeated iteratively. At every generation we have a solution that corresponds to a genome with the highest fitness function. The farther we go with generations the higher is the chance to find a better solution.

The used GA can be described as follows:

1. Prepare the genome structure according to the structure of FRNN.
2. If we already have a good genome (an existing network solution), put it into population.
   else generate a random network solution and get the genome and put it into population.
3. Generate at random new *PopSize*-1 genomes and put them into population.
4. Apply genetic crossover operation to *PopSize* genomes in the population.
5. Apply mutation operation to the generated offsprings.
6. Get phenotype and rank (i.e., evaluate and assign fitness values to) all the offsprings.
7. Create new population with N_{best} best parent genomes and (*PopSize*-N_{best}) best offsprings.
8. Display fitness value of the best genome.
   If termination condition is met go to Step 9.
   Else go to step 4.
9. Get phenotype of the best genome in the population. Store network weights file.
10. Stop.

In above algorithm *PopSize* is minimum population size and N_{best} is the number of best parent genomes always kept in the newly generated population.

4 Computational experiments

In all our experiments all connection weights and biases are coded as 64 bits long genes. In simulation experiments we use fuzzy numbers of type: $T(L_0, L_1, L_2, R_2, R_1, R_0)$, i.e., so called 6-points fuzzy numbers. 12 bits for coding L_2 and R_2, 12 bits for coding L_1 and R_1 and 8 bits for coding L_0 and R_0. Thus, for example, if we have in total 54 adjustable parameters in the network, the genome length will be $54 \times 64 = 1,088$ bits.

For better learning we use 100 genomes. All 100 genomes undergo the crossover operation. We use so-called multi-point crossover, in which the bits from one genome is inherited from the mate with a particular probability (e.g., 0.25), i.e., several bits in a genome can be changed after the crossover. After the crossover, the mutation operation is applied to the offspring genomes. With a specified probability, called the mutation probability, the particular bits in all processed genomes are inverted.

Then every 90 best offspring genomes plus 10 best parent genomes make a new population of 100 genomes (we preserve best 10 parent genomes in every next generation).

The selection of 100 best genomes is done on the basis of the genome fitness value.

For this particular case the distance between two particular fuzzy numbers $T1$ and $T2$ is calculated as follows:

$$\begin{aligned} D(T1, T2) = {} & k_0|L_{T10} - L_{T20}| + k_1|L_{T11} - L_{T21}| \\ & + k_2|L_{T12} - L_{T22}| + k_2|R_{T12} - R_{T22}| \\ & + k_1|R_{T11} - R_{T21}| + k_0|R_{T10} - R_{T20}|, \end{aligned}$$

The weighting coefficients k_0, k_1, k_2 are set as $k_0 = 0.10$, $k_1 = 0.35, k_2 = 0.55$

In GA described at the end of previous section we used *PopSize*=100 and $N_{\text{best}} = 10$.

Table 1 Forecasting enrollments of University of Alabama

	Years	Actual	FRNN output (fuzzy)	Defuz
	1971	13055		
	1972	13563		
	1973	13867		
	1974	14696	[14690,14692,14694,14695,14698,14701]	14694.9
	1975	15460	[15420,15457,15460,15460,15466,15471]	15460.2
	1976	15311	[15301,15308,15312,15312,15313,15320]	15312.1
	1977	15603	[15588,15595,15601,15603,15606,15611]	15602.4
Learning samples	1978	15861	[15859,15861,15862,15862,15870,15879]	15861.5
	1979	16807	[16799,16803,16807,16808,16815,16890]	16807.6
	1980	16919	[16897,16904,16920,16920,16923,16930]	16919.5
	1981	16388	[16379,16388,16388,16389,16394,16401]	16388.4
	1982	15433	[15408,15432,15433,15434,15440,15449]	15433.3
	1983	15497	[15444,15476,15497,15498,15506,15552]	15497.8
	1984	15145	[15104,15134,15145,15146,15172,15190]	15145.3
	1985	15163	[15114,15151,15163,15164,15177,15196]	15163.2
	1986	15984	[15897,15949,15983,15984,15997,16007]	15984.4
	1987	16859	[16832,16854,16857,16858,16863,16906]	16858.2
RMSE=0.63039699				
	Years	Actual	FRNN output (fuzzy)	Defuz
	1988	18150	[17969,17973,17976,17977,17983,17800]	17976.3
Test samples	1989	18970	[18900,18915,18930,18932,18935,18942]	18931.5
	1990	19328	[19116,19119,19127,19128,19133,19133]	19127.5
	1991	19337	[19149,19155,19160,19160,19160,19162]	19160
	1992	18876	[19170,19171,19172,19173,19177,19179]	19172.5
RMSE=195.4920473				

4.1 Benchmark testing

To test the suggested FRNN based fuzzy time series forecasting method the benchmark problem—the forecasting enrolments of University of Alabama has been used Hwang et al. (1998b). Table 1 shows actual enrolments and enrolments forecasted by the suggested FRNN based approach. Seventy percent of available data have been used for learning the system and the remaining 30% have been used to calculate the forecasting accuracy. The purpose of the forecasting system designed on the base of the developed FRNN is to predict the number of enrolled students for the next year given the actual number of enrolled students in the current year. The network had three layers, with one neuron in the input layer, ten neurons in the hidden layer, and one neuron in the output layer.

The offered approach's root mean square error (RMSE=194) is smaller than the Chen's method (630.86) Chen (1996), the Song-Chissom method (421.32) Song and Chissom (1993a), and the Tsai and Wu method using high order fuzzy time series (199.32) Tsai and Wu (2000).

Table 2 shows forecasting error produced by different methods in mean absolute percentage error (MAPE).

Table 2 Comparison of different forecasting methods

Song Chissom (1993)	Hwang et al. (1998)	Sullivan Woodall (1994)	Our approach (FRNN)
3.15%	2.79%	2.6%	0.9%

It can be seen from the test that the feasibility of the use of suggested FRNN based method to forecast time series is evident.

4.2 Temperature prediction

Table 3 shows a fragment of historical data of the daily average temperature in Taipei (Chen and Hwang 2000). Eighty five percent of the daily average temperature values in June, July, and September, fuzzified in advance as shown in Table 3, were used for training of FRNN.

As can be seen from Table 3 the data were fuzzified by 9 linguistic terms: "very-very low", "very low", "low", "more of less low", "average", "more or less high", "high", "very high", and "very-very high".

Table 3 Average temperature in Taipei (°C) in June 1996

Day	Crisp Temp.	Linguistic temperature								
		Very-very low	Very low	low	More or less low	Average	More or less high	High	Very high	Very-very high
1	26.1	0	0	0	0.93333	0.06667	0	0	0	0
2	27.6	0	0	0	0	0.93333	0.06667	0	0	0
3	29.0	0	0	0	0	0	1	0	0	0
4	30.5	0	0	0	0	0	0	1	0	0
5	30.0	0	0	0	0	0	0.33333	0.66667	0	0
6	29.5	0	0	0	0	0	0.66667	0.33333	0	0
7	29.7	0	0	0	0	0	0.53333	0.46667	0	0
8	29.4	0	0	0	0	0	0.73333	0.26667	0	0
9	28.8	0	0	0	0	0.13333	0.86667	0	0	0
10	29.4	0	0	0	0	0	0.73333	0.26667	0	0
11	29.3	0	0	0	0	0	0.8	0.2	0	0
12	28.5	0	0	0	0	0.33333	0.66667	0	0	0
13	28.7	0	0	0	0	0.2	0.8	0	0	0
14	27.5	0	0	0	0	1	0	0	0	0
15	29.5	0	0	0	0	0	0.66667	0.33333	0	0
16	28.8	0	0	0	0	0.13333	0.86667	0	0	0
17	29.0	0	0	0	0	0	1	0	0	0
18	30.3	0	0	0	0	0	0.13333	0.86667	0	0
19	30.2	0	0	0	0	0	0.2	0.8	0	0
20	30.9	0	0	0	0	0	0	0.73333	0.26667	0
21	30.8	0	0	0	0	0	0	0.8	0.2	0
22	28.7	0	0	0	0	0.2	0.8	0	0	0
23	27.8	0	0	0	0	0.8	0.2	0	0	0
24	27.4	0	0	0	0.06667	0.93333	0	0	0	0
25	27.7	0	0	0	0	0.86667	0.13333	0	0	0
26	27.1	0	0	0	0.26667	0.73333	0	0	0	0
27	28.4	0	0	0	0	0.4	0.6	0	0	0
28	27.8	0	0	0	0	0.8	0.2	0	0	0
29	29.0	0	0	0	0	0	1	0	0	0
30	30.2	0	0	0	0	0	0.2	0.8	0	0
31	26.1	0	0	0	0.93333	0.06667	0	0	0	0

Figure 2 shows the curves of the actual temperature and the forecasted temperature using the proposed method. The mean absolute percentage error (MAPE) achieved by our approach was 2.61%, RMSE = 0.90. This error value is lower than the error values (ranging from 2.75 to 3.49% produced by different algorithms) obtained by the method suggested in Chen and Hwang (2000).

4.3 Application of FRNN for forecasting demand in GAC

To test the suggested FRNN on an application problem we used the customer demand volumes for a real company. The example considered is based on the data of general appliance company (GAC) from the case prepared by Prof. Morris A.Cohen and Thomas F.Kendall, the Wharton School, University of Pennsylvania and Prof. Ricardo Ernst, Georgetown University. GAC's product line consisted of electric and gas ranges and ovens, clothes washers and dryers, and dishwashers. The appliances produced in 4 plants are shipped to 5 distribution centres which then distribute them to 12 customer zones for sale. Data for one (the starting) year are presented in Table 4.

Each next year the demand growth rates changes dynamically by some unknown rules. Table 5 shows approximate values of demand growth values for different zones.

Table 4 Appliances produced for the starting year (Year 0)

Customer zone	Washer	Dryer	Dish washer	Oven/range	Total
Atlanta	108,000	81,000	33,000	49,000	271,000
Chicago	120,000	61,000	26,000	39,000	246,000
Dallas	69,000	43,000	19,000	25,000	156,000
Denver	55,000	44,000	18,000	23,000	140,000
Detroit	82,000	45,000	22,000	40,000	189,000
L.A.	73,000	58,000	17,000	27,000	175,000
New Orleans	54,000	56,000	26,000	42,000	178,000
Philadelphia	85,000	57,000	21,000	41,000	204,000
St. Louis	71,000	72,000	21,000	48,000	212,000
Seattle	54,000	31,000	18,000	25,000	128,000
Canada	25,000	22,000	9,000	15,000	71,000
Foreign	6,000	8,000	11,000	12,000	37,000
Total	802,000	578,000	241,000	386,000	2,007,000

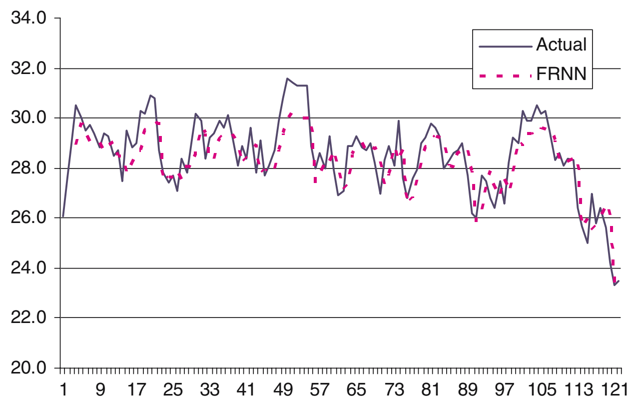

Fig. 2 The curve of actual temperatures and FRNN forecasted temperatures

Table 5 Approximate values of demand growth values for different zones

Zones:	Washer	Dryer	Washer	Range
Eastern	0.04	0.04	0.05	0.03
Central US	0.03	0.03	0.04	0.02
Western US	0.02	0.02	0.04	0.02
Canada	0.03	0.03	0.06	0.02
Foreign	0.04	0.03	0.05	0.03

Table 6 The actual customer demand values for 10 years

Year	Washer	Dryer	Dish washer	Oven/range	Total
0	802, 000	578,000	241,000	386,000	2,007,000
1	826, 230	595,390	251,470	394,740	2,067,830
2	810, 000	613,433	262,676	404,789	2,090,898
3	873, 494	540,000	274,591	414,770	2,102,855
4	901, 862	600,000	286,594	425,407	2,213,862
5	930, 572	670,041	299,049	435,574	2,335,236
6	1, 040, 000	690,748	380,000	445,651	2,556,399
7	994, 324	712,778	380,000	455,757	2,542,858
8	1, 025, 478	735,306	400,000	466,294	2,627,079
9	1, 054, 753	759,929	430,000	476,222	2,720,904

Table 7 Comparison of the actual and forecasted demand values

Year	Actual	FRNN based (Defuzzif.)
0	2,007,000	
1	2,067,830	
2	2,090,898	
3	2,102,855	2,102,855
4	2,213,862	2,213,862
5	2,335,236	2,335,236
6	2,556,399	2,556,399
7	2,542,858	2,542,858
8	2,627,079	2,612,333
9	2,720,904	2,549,967

Data in Table 6 shows the actual customer demand values for 10 years. The FRNN was trained on eight data and the last two data were used to test the performance.

Table 7 shows the results of comparison of the actual total demand data and the total demand data forecasted by the FRNN. For years 0–7, the network outputs with learned data and for years 8 and 9—forecasting is made on the basis of unknown data.

Figure 3 shows the total customer demand data for different appliances along with the forecasting results.

The obtained results compared with the actually materialized company demand data appeared to be quite acceptable

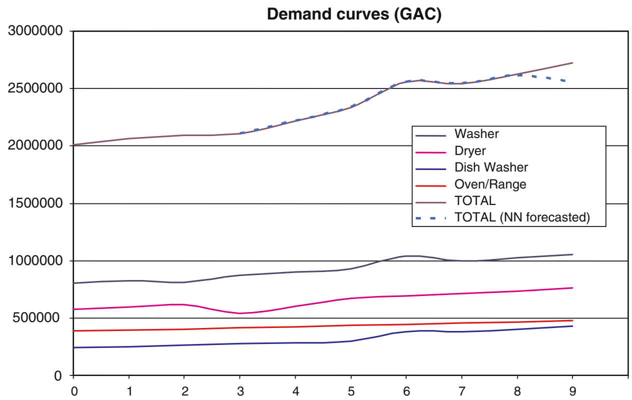

Fig. 3 Supplier demand actual and forecasted data

and proved the feasibility of using FRNN to forecast time series.

5 Conclusions

In this paper we have proposed a new fuzzy recurrent neural network based time series forecasting method which can deal with historical perception type data described by linguistic values. The distinguishing features of the proposed forecasting method are: ability to process both numerical and linguistic information and to apply human expertise throughout the forecasting procedure; being based on FRNN the proposed fuzzy time series model is able to update the forecasting rules extracted from a dynamic data mining procedure; less computational complexity in comparison to existing fuzzy time series models due to parallel processing of perceptions and data and fast fuzzy inference; significantly high degree of forecasting accuracy in comparison with fuzzy time series models based on fuzzy relational equation caused mainly by the features of proposed method: universal approximation and learning from time series data base.

The developed forecasting method was applied to three data sets: the benchmark problem of forecasting of enrolments to the University of Alabama (obtained accuracy was 0.9%, reduced from 2.6% by the best other method), a temperature prediction system (obtained accuracy was 2.61%, reduced from 2.75% by the best other method), and the problem of forecasting demand for a general appliance company. The performance of the proposed method for forecasting fuzzy time series shows its high efficiency and effectiveness for a wide domain of application areas ranging from weather forecasting to planning in economics and business.

References

Aliev RA, Fazlollahi B, Aliev RR (2004) Soft computing and its applications in business and economics. Springer, Heidelberg
Aliev RA, Fazlollahi B, Vahidov R (2001) Genetic algorithm-based learning of fuzzy neural networks. Part 1: feed-forward fuzzy neural networks. Fuzzy Sets Syst 118:351–358
Aliev RA, Guirimov BG, Fazlollahi B, Aliev RR (2006) Genetic algorithm-based learning of fuzzy neural networks. Part 2: recurrent fuzzy neural networks. Fuzzy Sets Syst (submitted)
Castillo O, Melin PA (2001) New-fractal approach for forecasting financial and economic time series. J IEEE, 929–934
Chen SM (1996) Forecasting enrolments based on fuzzy time series. Fuzzy Sets Sys 81:311–319
Chen SM, Hwang JR (2000) Temperature prediction using fuzzy time series. Trans Syst Man Cybern Part B: Cybern 30(2):263–275
Hwang JR, Chen SM, Lee CH (1996) A new method for handling forecasting problems based on fuzzy time series. In: 7th international conference on information management. Chungli, Taoyuan, pp 312–321
Hwang JR, Chen SM, Lee CH (1998) Handling forecasting problems using fuzzy time series. J Fuzzy Sets Syst 100:217–228
Hwang JR, Chen SM, Lee CH (1998) Handling forecasting problems using fuzzy time series. Fuzzy Sets Syst 100:217–228
Liu P, Li H (2004) Fuzzy neural network theory and applications. World Scientific, Singapore
Nikravesh M, Zadeh L, Korotkikh V (eds) (2004) Fuzzy partial differential equations and relational equations. Springer-Verlag, Heidelberg
Pedrycz W (1989) Fuzzy control and fuzzy systems. Wiley, New York
Pedrycz W (1991) Neurocomputations in relational systems. IEEE Trans Pattern Anal Mach Intell 13(3):289–297
Song Q, Chissom BS (1993) Forecasting enrollments with fuzzy time series—part I. J Fuzzy Sets Syst 54:1–9
Song Q, Chissom BS (1993) Fuzzy time series and its models. J Fuzzy Sets Syst 54:269–277
Song Q, Chissom BS (1994) Forecasting enrollments with fuzzy time series—part II. J Fuzzy Sets Syst 62:1–8
Sullivan J, Woodall WH (1994) A comparison of fuzzy forecasting and Markov modeling. J Fuzzy Sets Syst 64:279–293
Tsai CC, Wu SJ (1999) A Study for second-order modeling of fuzzy time series. In: IEEE International fuzzy systems conference, Seoul
Tsai CC, Wu SJ (2000) Forecasting enrolments with high-order fuzzy time series. In: Proceedings of 19th international conference of the North American Fuzzy Information Processing Society (NAFIPS), pp 196–200
Zadeh L (1975) The concept of a linguistic variable and its application to approximate reasoning. J Inf Sci 8:43–80
Zuoyoung L, Zhenpei C, Jitao LA (1998) Model of weather forecast by fuzzy grade statistics. J Fuzzy Sets Syst 26:275–281