Ensemble Neural Network with Type-1 and Type-2 Fuzzy Integration for Time Series Prediction and Its Optimization with PSO

Patricia Melin, Martha Pulido and Oscar Castillo

Abstract This paper describes the design of ensemble neural networks using Particle Swarm Optimization (PSO) for time series prediction with Type-1 and Type-2 Fuzzy Integration. The time series that is being considered in this work is the Mackey-Glass benchmark time series. Simulation results show that the ensemble approach produces good prediction of the Mackey-Glass time series.

Keywords Ensemble neural networks · Particle swarm · Optimization · Time series prediction

1 Introduction

Time Series is defined as a set of measurements of some phenomenon or experiment recorded sequentially in time. The first step in analyzing a time series is to plot it, this allows: to identify the trends, seasonal components and irregular variations. A classic model for a time series can be expressed as a sum or product of three components: trend, seasonality and random error term.

Time series predictions are very important because based on them we can analyze past events to know the possible behavior of futures events and thus we can take preventive or corrective decisions to help avoid unwanted circumstances.

The contribution of this paper is the proposed approach for ensemble neural network optimization using particle swarm optimization. The proposed models are also used as a basis for statistical tests [1–4, 9, 10, 12, 14, 15, 19–22].

The rest of the paper is organized as follows: Sect. 2 describes the concepts of optimization, Sect. 3 describes the concepts of particle swarm optimization, Sect. 4

P. Melin (✉) · M. Pulido · O. Castillo
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: pmelin@tectijuana.mx

P. Angelov and S. Sotirov (eds.), *Imprecision and Uncertainty in Information Representation and Processing*, Studies in Fuzziness and Soft Computing 332,
DOI 10.1007/978-3-319-26302-1_22

describes the concepts of Fuzzy Systems as Methods of integration, Sect. 5 describes the problem and the proposed method of solution, Sect. 6 describes the simulation results of the proposed method, and Sect. 7 shows the conclusions.

2 Optimization

Regarding optimization, we have the following situation in mind: there exists a search space ***V***, and a function:

$$g\colon V \rightarrow \mathbb{R}$$

and the problem is to find

$$\underset{v \in V}{\mathbf{arg\,min\,g}}.$$

Here, ***V*** *is vector of decision variables,* and ***g*** *is the objective function.* In this case we have assumed that the problem is one of minimization, but everything we say can of course be applied *mutatis mutandis* to a maximization problem. Although specified here in an abstract way, this is nonetheless a problem with a huge number of real-world applications.

In many cases the search space is discrete, so that we have the class of *combinatorial optimization problems* (COPs). When the domain of the ***g*** function is continuous, a different approach may well be required, although even here we note that in practice, optimization problems are usually solved using a computer, so that in the final analysis the solutions are represented by strings of binary digits (bits) [32].

There are several optimization techniques that can be applied to neural networks, some of these are: evolutionary algorithms [18], ant colony optimization [5] and Particle swarm [7].

3 Particle Swarm Optimization

The Particle Swarm Optimization algorithm maintains a swarm of particles, where each particle represents a potential solution. In analogy with evolutionary computation paradigms, a swarm is a population, while a particle is similar to an individual. In simple terms, the particles are "flown" through a multidimensional search space where the position of each particle is adjusted according to its own experience and that of their neighbors. Let x_i(t) denote the position of particle *i* in the search space at time step *t* unless otherwise selected, *t* denotes discrete time steps. The position of the particle is changed by adding a velocity, v_i(t) to the current position i.e.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{1}$$

with $x_i(0) \sim U(X_{min}, X_{max})$.
It is the velocity vector the one that drives of the optimization process, and reflects both the experimental knowledge of the particles and the information exchanged in the vicinity of particles. The experimental knowledge of a particle which is generally known as the cognitive component, which is proportional to the distance of the particle from its own best position (hereinafter, the personal best position particles) that are from the first step. Socially exchanged information is known as the social component of the velocity equation.

For the gbest PSO, the particle velocity is calculated as:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_1 \left[y_{ij}(t) - x_{ij}(t)\right], + c_2 r_2(t) \left[\hat{y}_j(t) - x_{ij}(t)\right] \tag{2}$$

where $v_{ij}(t)$ is the velocity of the particle *i* in dimension *j* at time step *t*, c_1 *y* c_2 are positive acceleration constants used to scale the contribution of cognitive and social skills, respectively, y $r_{1j}(t)$, y $r_{2j}(t) \sim U(0, 1)$ are random values in the range [0, 1].

The best personal position in the next time step $t + 1$ is calculated as:

$$y_i(t+1) = \begin{cases} y_i(t) & if\ f(x_i(x_i(t+1)) \geq f\, y_i(t)) \\ x_i(t+1) & if\ f(x_i(x_i(t+1)) > f\, y_i(t)) \end{cases} \tag{3}$$

where $f\colon \mathbb{R}^{nx} \to \mathbb{R}$ is the fitness function, as with EAs, measuring fitness with the function will help find the optimal solution, for example the objective function quantifies the performance, or the quality of a particle (or solution).

The overall best position, $\hat{y}$(t) at time step t, is defined as:

$$\hat{y}(t) \quad \epsilon\{y_0(t), \ldots, y_{ns}(t)\} f(y(t)) = \min\{f(y_0(t)), \ldots f(y_{ns}(t)),\} \tag{4}$$

where n_S is the total number of particles in the swarm. Importantly, the above equation defining and establishing $\hat{y}$ the best position is uncovered by either of the particles so far as this is usually calculated from the best position best personal [5, 6, 10].

The overall best position may be selected from the actual swarm particles, in which case:

$$\hat{y}(t) = \min\{f(x_o(t)), \ldots f(x_{ns}(t)),\} \tag{5}$$

4 Fuzzy Systems as Methods of Integration

Fuzzy logic was proposed for the first time in the mid-sixties at the University of California Berkeley by the brilliant engineer Lofty A. Zadeh., who proposed what it's called the principle of incompatibility: "As the complexity of system increases, our ability to be precise instructions and build on their behavior decreases to the threshold beyond which the accuracy and meaning are mutually exclusive characteristics." Then introduced the concept of a fuzzy set, under which lies the idea that the elements on which to build human thinking are not numbers but linguistic labels. Fuzzy logic can represent the common knowledge as a form of language that is mostly qualitative and not necessarily a quantity in a mathematical language [29].

Type-1 Fuzzy system theory was first introduced by Zadeh [13] in 1965, and has been applied in many areas such as control, data mining, time series prediction, etc.

The basic structure of a fuzzy inference system consists of three conceptual components: a rule base, which contains a selection of fuzzy rules, a database (or dictionary) which defines the membership functions used in the rules, and reasoning mechanism, which performs the inference procedure (usually fuzzy reasoning) [14].

Type-2 Fuzzy systems were proposed to overcome the limitations of a type-1 FLS, the concept of type-1 fuzzy sets was extended into type-2 fuzzy sets by Zadeh in 1975. These were designed to mathematically represent the vagueness and uncertainty of linguistic problems; thereby obtaining formal tools to work with intrinsic imprecision in different type of problems; it is considered a generalization of the classic set theory. Type-2 fuzzy sets are used for modeling uncertainty and imprecision in a better way [15–17].

5 Problem Statement and Proposed Method

The objective of this work is to develop a model that is based on integrating the responses of an ensemble neural network using type-1 and type-2 fuzzy systems and their optimization. Figure 1 represents the general architecture of the proposed method, where historical data, analyzing data, creation of the ensemble neural network and integrate responses of the ensemble neural network with type-2 fuzzy system integration and finally obtaining the outputs as shown. The information can be historical data, these can be images, time series, etc., in this case we show the application to time series prediction of the Dow Jones where we obtain good results with this series.

Figure 2 shows a type-2 fuzzy system consisting of 5 inputs depending on the number of modules of the neural network ensemble and one output. Each input and output linguistic variable of the fuzzy system uses 2 Gaussian membership functions. The performance of the type-2 fuzzy integrators is analyzed under different levels of uncertainty to find out the best design of the membership functions and consist of 32 rules. For the type-2 fuzzy integrator using 2 membership functions,

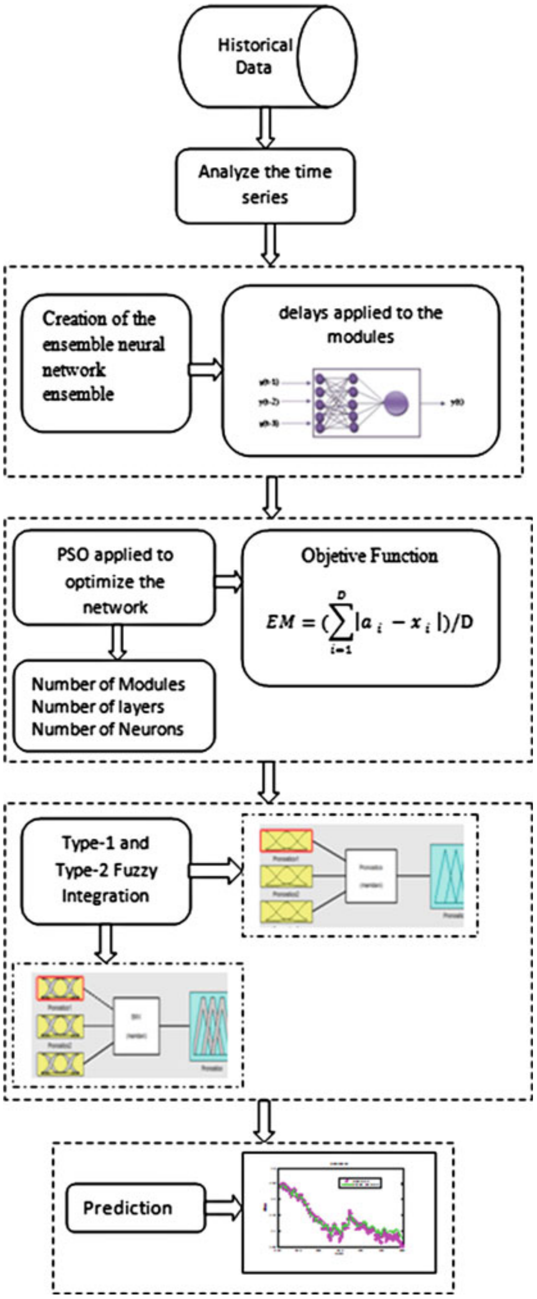


Fig. 1 General architecture of the proposed method

Historical Data
Analyze the time series
Creation of the ensemble neural network ensemble
delays applied to the modules
PSO applied to optimize the network
Objetive Function
$EM = (\sum_{i=1}^{D} |a_i - x_i|)/D$
Number of Modules
Number of layers
Number of Neurons
Type-1 and Type-2 Fuzzy Integration
Prediction

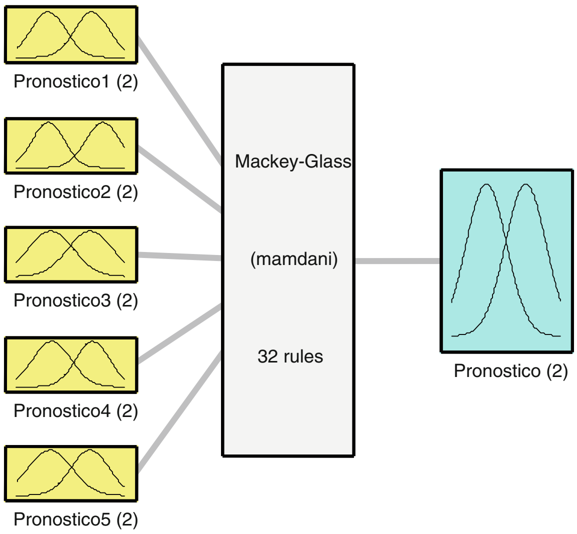
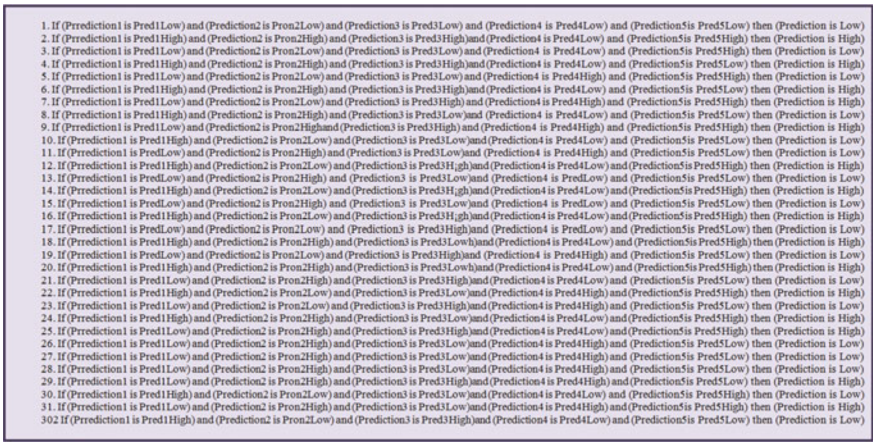


Fig. 2 Type-2 fuzzy system for the Mackey Glass time series

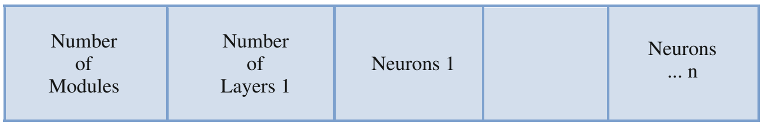


1. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3Low) and (Prediction4 is Pred4Low) and (Prediction5is Pred5Low) then (Prediction is Low)
2. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2High) and (Prediction3 is Pred3High)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is High)
3. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3Low) and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is Low)
4. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2High) and (Prediction3 is Pred3High)and (Prediction4 is Pred4Low) and (Prediction5is Pred5Low) then (Prediction is High)
5. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3Low) and (Prediction4 is Pred4High) and (Prediction5is Pred5High) then (Prediction is Low)
6. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2High) and (Prediction3 is Pred3High)and (Prediction4 is Pred4Low) and (Prediction5is Pred5Low) then (Prediction is High)
7. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3High) and (Prediction4 is Pred4High) and (Prediction5is Pred5High) then (Prediction is High)
8. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4Low) and (Prediction5is Pred5Low) then (Prediction is Low)
9. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2Highand (Prediction3 is Pred3High) and (Prediction4 is Pred4High) and (Prediction5is Pred5High) then (Prediction is High)
10. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4Low) and (Prediction5is Pred5Low) then (Prediction is Low)
11. If (Prrediction1 is PredLow) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4High) and (Prediction5is Pred5Low) then (Prediction is Low)
12. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3H¡gh)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is High)
13. If (Prrediction1 is PredLow) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is PredLow) and (Prediction5is Pred5Low) then (Prediction is Low)
14. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3H¡gh)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is High)
15. If (Prrediction1 is PredLow) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is PredLow) and (Prediction5is Pred5Low) then (Prediction is Low)
16. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3H¡gh)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is High)
17. If (Prrediction1 is PredLow) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3High)and (Prediction4 is PredLow) and (Prediction5is Pred5Low) then (Prediction is Low)
18. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Lowh)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is High)
19. If (Prrediction1 is PredLow) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3High)and (Prediction4 is Pred4High) and (Prediction5is Pred5Low) then (Prediction is Low)
20. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Lowh)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is High)
21. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2High) and (Prediction3 is Pred3High)and (Prediction4 is Pred4Low) and (Prediction5is Pred5Low) then (Prediction is Low)
22. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4High) and (Prediction5is Pred5High) then (Prediction is High)
23. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3High)and (Prediction4 is Pred4High) and (Prediction5is Pred5Low) then (Prediction is Low)
24. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is High)
25. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2High) and (Prediction3 is Pred3High)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is High)
26. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4High) and (Prediction5is Pred5Low) then (Prediction is Low)
27. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4High) and (Prediction5is Pred5Low) then (Prediction is Low)
28. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4High) and (Prediction5is Pred5Low) then (Prediction is Low)
29. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2High) and (Prediction3 is Pred3High)and (Prediction4 is Pred4High) and (Prediction5is Pred5Low) then (Prediction is High)
30. If (Prrediction1 is Pred1High) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4Low) and (Prediction5is Pred5High) then (Prediction is Low)
31. If (Prrediction1 is Pred1Low) and (Prediction2 is Pron2High) and (Prediction3 is Pred3Low)and (Prediction4 is Pred4High) and (Prediction5is Pred5High) then (Prediction is High)
302 If (Prrediction1 is Pred1High) and (Prediction2 is Pron2Low) and (Prediction3 is Pred3High)and (Prediction4 is Pred4Low) and (Prediction5is Pred5Low) then (Prediction is Low)

Fig. 3 Rules of the type-2 fuzzy inference system for the Dow Jones time series

which are called low prediction and high prediction for each of the inputs and output of the fuzzy system. The membership functions are of Gaussian type, and we consider 3 sizes for the footprint uncertainty 0.3, 0.4 and 0.5 to obtain a better prediction of our time series.

In this Fig. 3 shows the possible rules of a type-2 fuzzy system.



Number of Modules	Number of Layers 1	Neurons 1		Neurons ... n

Fig. 4 Particle structure to optimize the ensemble neural network

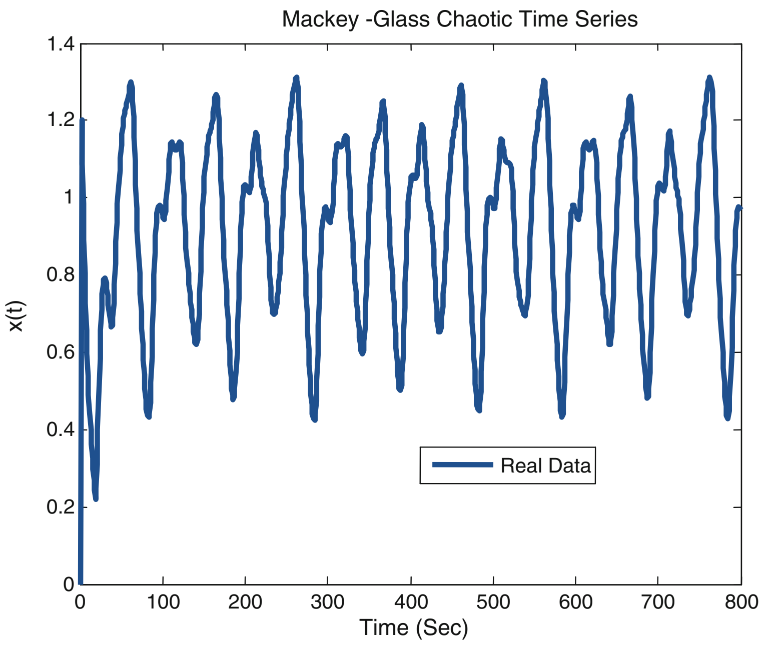


Fig. 5 Mackey Glass time series

Figure 4 represents the Particle Structure to optimize the ensemble neural network, where the parameters that are optimized are the number de modules, number of layers, number of neurons.

Data of the Mackey-Glass time series was generated using Eq. (6). We are using 800 points of the time series. We use 70 % of the data for the ensemble neural network trainings and 30 % to test the network.

The Mackey-Glass Equation is defined as follows:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{6}$$

where it is assumed x(0) = 1.2, $\tau = 17$, $\tau = 34$, and 68 x(t) = 0 for t < 0. Figure 5 shows a plot of the time series for these parameter values.

This time series is chaotic, and there is no clearly defined period. The series does not converge or diverge, and the trajectory is extremely sensitive to the initial conditions. The time series is measured in number of points, and we apply the fourth order Runge-Kutta method to find the numerical solution of the equation [10, 11].

6 Simulation Results

In this section we present the simulation results obtained with the integration of the ensemble neural network with type-2 fuzzy integration and its optimization with the genetic algorithm for the Mackey-Glass time series.

Table 1 shows the particle swarm optimization where the best prediction error is of 0.0063313.

Table 1 Particle swarm results for the ensemble neural network $\tau = 17$

No.	Iterations	Panicles	Number modules	Number layers	Number neurons	Duration	Prediction error
1	100	100	4	2	20, 14 13, 16 17, 8 6, 26	02:23:18	0.0076048
2	100	100	2	2	12, 16 12, 26	01:45:45	0.0063313
3	100	100	2	3	17, 5, 18 6, 25, 24	01:28:42	0.0018838
4	100	100	4	2	7, 24 14, 22 1, 8 15, 23	02:40:20	0.0073005
5	100	100	4	2	14, 9 11, 26 27, 16 11, 13	02:11:34	0.0081418
6	100	100	4	2	16, 16 9, 19 6, 6 9, 12	01:34:05	0.0087983
7	100	100	2	3	11, 23, 26 15, 15, 5	02:09:17	0.0076315
8	100	100	2	2	14, 10 14, 21	01:23:28	0.0061291
9	100	100	3	2	9, 5 23, 20 22, 13	02:17:06	0.0053679
10	100	100	3	3	23, 14, 16 19, 10, 23 22, 12, 11	02:20:04	0.0061983

Table 2 Results of Type-1 fuzzy integration for $\tau = 17$

Experiment	Prediction error with fuzzy integration Type-1
Experiment 1	0.1879
Experiment 2	0.1789
Experiment 3	0.2221
Experiment 4	0.1888
Experiment 5	0.1521
Experiment 6	0.2561
Experiment 7	0.1785
Experiment 8	0.1942
Experiment 9	0.2536
Experiment 10	0.1965

Table 3 Results of Type-2 fuzzy integration for $\tau = 17$

Experiment	Prediction error 0.3 Uncertainty	Prediction error 0.4 Uncertainty	Prediction error 0.5 Uncertainty
Experiment 1	0.2385	0.2385	0.3952
Experiment 2	0.2489	0.2231	0.3909
Experiment 3	0.2482	0.2226	0.3642
Experiment 4	0.2214	**0.1658**	0.3856
Experiment 5	0.2658	0.2234	0.3857
Experiment 6	0.2756	0.2592	**0.3134**
Experiment 7	**0.1785**	0.2352	0.3358
Experiment 8	0.1825	0.2546	0.4561
Experiment 9	0.2018	0.2373	0.3394
Experiment 10	0.2076	0.2003	0.3687

Fuzzy integration is performed initially by implementing a type-1 fuzzy system in which the best result was in the experiment of row number 5 of Table 2 with an error of: 0.1521.

Fuzzy integration is performed by implementing a type-1 fuzzy system in which the results were as follows: for the best evolution with a degree of uncertainty of 0.3 a forecast error of 0.1785 was obtained, and with a degree of uncertainty of 0.4 a forecast error of 0.1658 and with a degree of uncertainty of 0.5 a forecast error of 0.3134 was obtained, as shown in Table 3.

Table 4 shows the particle swarm optimization where the best prediction error is of 0.0019726.

Fuzzy integration is performed by implementing a type-1 fuzzy system in which the best result was in the experiment of row number 2 of Table 5 with an error of: 0.4586.

Table 4 Particle swarm results for the ensemble neural network for $\tau = 34$

No.	Iterations	Particles	Number modules	Number layers	Number neurons	Duration	Prediction error
1	100	100	4	3	12, 23, 12 9, 19, 7	02:45:14	0.0019726
2	100	100	4	3	19, 11, 11 16, 11, 14 15, 24, 19 22, 13, 27	01:28:06	0.0063623
3	100	100	3	2	4, 9 9, 20 10, 11 23, 20	02:03:06	0.0046644
4	100	100	4	2	14, 18 12, 19 20, 17 10, 6	03:22:13	0.0072153
5	100	100	3	2	7, 6 10, 15 12, 16	01:39:13	0.0075658
6	100	100	3	3	14, 20, 18 15, 21, 12 19, 17, 26	03:08:02	0.0047515
7	100	100	2	2	4, 24 9, 26	02:00:10	0.003601
8	100	100	2	2	24, 17 14, 23	02:27:21	0.0065506
9	100	100	3	3	7, 11, 8 23, 21, 21 17, 8, 11	02:03:12	0.0037758
10	100	100	2	3	20, 28, 15 15, 12, 24	02:04:18	0.0066375

Table 5 Results of Type-1 fuzzy integration for $\tau = 34$

Experiment	Prediction error with fuzzy integration Type-1
Experiment 1	0.9587
Experiment 2	**0.4586**
Experiment 3	0.5871
Experiment 4	1.2569
Experiment 5	0.9517
Experiment 6	1.556
Experiment 7	1.0987
Experiment 8	1.9671
Experiment 9	1.698
Experiment 10	1.4626

Table 6 Results of Type-2 fuzzy integration for $\tau = 34$

Evolution	Prediction error 0.3 Uncertainty	Prediction error 0.4 Uncertainty	Prediction error 0.5 Uncertainty
Evolution 1	**0.6036**	0.8545	0.4570
Evolution 2	1.5862	1.0021	1.3533
Evolution 3	0.8002	0.6943	**0.3893**
Evolution 4	1.4032	0.9617	0.9665
Evolution 5	0.8658	0.8299	0.6358
Evolution 6	1.3986	0.1052	1.2354
Evolution 7	1.465	1.3566	0.6646
Evolution 8	1.7453	0.8966	0.8241
Evolution 9	0.9866	**0.6524**	0.6661
Evolution 10	1.4552	0.9956	0.7557

Table 7 Particle swarm results for the ensemble neural network for $\tau = 68$

No.	Iterations	Particles	Number of modules	Number of layers	Number of neurons	Duration	Prediet ion error
1	**100**	**100**	**2**	**3**	**17, 5, 18** **6, 25, 19**	**02:05:53**	**0.0019348**
2	100	100	2	2	7, 8 6, 20	04:1936	0.0041123
3	100	100	2	3	21, 11, 16 5, 10, 10	02:23:02	0.0042367
4	100	100	4	3	15, 7, 4 11, 22, 5 24, 19, 22 4, 14, 11	02:37:06	0.0050847
5	100	100	3	2	22, 23 2, 21 10, 2	01:5	0.0037132
6	100	100	4	3	10, 13, 22 24, 8, 17 13, 16, 20 7, 24, 17	02:10:27	0.0057235
7	100	100	2	2	8, 20 15, 23		0.0033082
8	100	100	3	2	28, 6 2, 16 18, 10	01:40:18	0.0057402
9	100	100	3	2	22, 17 10, 10 21, 12	02:45:31	0.0047309
10	100	100	2	3	22, 11, 18 27, 7, 14	01:35:13	0.0044649

Table 8 Results of Type-1 fuzzy integration for $\tau = 68$

Experiment	Prediction error with fuzzy integration Type-1
Experiment 1	0.8753
Experiment 2	0.3625
Experiment 3	0.6687
Experiment 4	**0.3254**
Experiment 5	0.5489
Experiment 6	1.3183
Experiment 7	1.8972
Experiment 8	1.6977
Experiment 9	1.5879
Experiment 10	0.9652

Fuzzy integration is performed by implementing a type-2 fuzzy system in which the results were as follows: for the best evolution with a degree of uncertainty of 0.3 a forecast error of 0.6036 was obtained, and with a degree of uncertainty of 0.4 a forecast error of 0.6524 and with a degree of uncertainty of 0.5 a forecast error of 0.3893 was obtained, as shown in Table 6.

Table 7 shows the particle swarm optimization where the prediction error is of 0.0019348.

Fuzzy integration is performed by implementing a type-1 fuzzy system in which the best result was in the experiment of row number 4 of Table 8 with an error of: 0.32546.

Fuzzy integration is also performed by implementing a type-2 fuzzy system in which the results were as follows: for the best evolution with a degree of uncertainty of 0.3 a forecast error of 0.6825 was obtained, and with a degree of uncertainty of 0.4 a forecast error of 0.7652 and with a degree of uncertainty of 0.5 a forecast error of 0.6581 was obtained, as shown in Table 9.

Table 9 Results of Type-2 fuzzy integration for $\tau = 68$

Evolution	Prediction error 0.3 Uncertainty	Prediction error 0.4 Uncertainty	Prediction error 0.5 Uncertainty
Evolution 1	0.7895	0.9631	0.7365
Evolution 2	0.9875	1.2365	1.564
Evolution 3	0.9874	0.7965	**0.6581**
Evolution 4	1.5325	0.9874	0.9723
Evolution 5	0.7763	0.9723	0.9858
Evolution 6	0.8694	0.9235	1.3697
Evolution 7	**0.6825**	1.4263	0.6646
Evolution 8	1.336	0.8963	0.8288
Evolution 9	0.9852	**0.7652**	0.7234
Evolution 10	1.365	1.4224	1.5984

7 Conclusions

Using the technique of PSO particle we can reach the conclusion that this algorithm is good for reducing the execution time compared to other techniques such as genetic algorithms, and also architectures for ensemble neural network are small and they applied to the time series, as in this case the time series of Mackey-Glass. Also the outputs results obtained integrating the results of the neural network with type-1 and type-2 fuzzy systems and integrated type-2 the best results with type 2 are very good.

Acknowledgments We would like to express our gratitude to the CONACYT, Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. Castillo, O., Melin, P.: Type-2 Fuzzy Logic: Theory and Applications. Neural Networks, pp. 30–43. Springer, New York (2008)
2. Castillo, O., Melin, P.: Hybrid intelligent systems for time series prediction using neural networks, fuzzy logic, and fractal theory. IEEE Trans. Neural Netw. **13**(6), 1395–1408 (2002)
3. Castillo, O., Melin, P.: Simulation and forecasting complex economic time series using neural networks and fuzzy logic. In: Proceedings of the International Neural Networks Conference, vol. 3, pp. 1805–1810 (2001)
4. Castillo, O., Melin, P.: Simulation and forecasting complex financial time series using neural networks and fuzzy logic. In: Proceedings the IEEE the International Conference on Systems, Man and Cybernetics, vol. 4, pp. 2664–2669 (2001)
5. Eberhart, R.C., Kennedy, J.: A new optimizer particle swarm theory. In: Proceedings of the sixth Symposium on Micromachine and Human Science, pp. 39–43 (1995)
6. Eberhart, R.C.: Fundamentals of Computational Swarm Intelligence, pp. 93–129. Wiley, New York (2005)
7. Jang, J.S.R, Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Sof Computing, Prentice Hall, Englewood Cliffs (1996)
8. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings Intelligent Symposium, pp. 80–87. April 2003
9. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. in: Tesauro, G., Touretzky, D., Leen, T. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 231–238, 1001. MIT Press, Cambridge, Denver (1995)
10. Mackey, M.C.: Adventures in Poland: having fun and doing research with Andrzej Lasota. Mat. Stosow **8**, 5–32 (2007)
11. Mackey, M.C., Glass, L.: Oascillation and chaos in physiological control systems. Science **197**, 287–289 (1997)
12. Maguire, L.P., Roche, B., McGinnity, T.M., McDaid, L.J.: Predicting a chaotic time series using a fuzzy neural network. Adv. Soft Comput. **12**(1–4), 125–136 (1998)
13. Multaba, I.M., Hussain, M.A.: Application of Neural Networks and Other Learning. Technologies in Process Engineering, Imperial Collage Press, London (2001)
14. Plummer, E.A.: Time series forecasting with feed-forward neural networks: guidelines and limitations. University of Wyoming, July 2000

15. Pulido, M., Mancilla, A., Melin, P.: An ensemble neural network architecture with fuzzy response integration for complex time series prediction. Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control, vol. 257/2009, pp. 85–110. Springer, Berlin (2009)
16. Sharkey, A.: One combining Artificial of Neural Nets. Department of Computer Science, University of Sheffield, Sheffield (1996)
17. Sharkey, A.: Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. Springer, London (1999)
18. Sollich, P., Krogh, A.: Learning with ensembles: how over-fitting can be useful. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) Advances in Neural Information Processing Systems, vol. 8, pp. 190–196. MIT Press, Denver, Cambridge (1996)
19. Yadav, R.N., Kalra, P.K., John, J.: Time series prediction with single multiplicative neuron model. Soft Comput. Time Ser. Predict. Appl. Soft Comput. **7**(4), 1157–1163 (2007)
20. Yao, X., Liu, Y.: Making use of population information in evolutionary artificial neural networks. IEEE Trans. Syst. Man Cybern. Part B Cybern. **28**(3), 417–425 (1998)
21. Zhao, L., Yang, Y.: PSO-based single multiplicative neuron model for time series prediction. Expert Syst. Appl. **36**(2 Part 2), 2805–2812 (2009)
22. Zhou, Z.-H., Jiang, Y., Yang, Y.-B., Chen, S.-F.: Lung cancer cell identification based on artificial neural network ensembles. Artif. Intell. Med. **24**(1), 25–36 (2002)