Neural Network with Fuzzy Weights Using Type-1 and Type-2 Fuzzy Learning for the Dow-Jones Time Series

Fernando Gaxiola, Patricia Melin and Fevrier Valdez

Abstract In this paper, type-1 and type-2 fuzzy inferences systems are used to obtain the type-1 or type-2 fuzzy weights in the connections between the layers of a neural network. We use two type-1 or type-2 fuzzy systems that work in the backpropagation learning method with the type-1 or type-2 fuzzy weight adjustment. The mathematical analysis of the proposed learning method architecture and the adaptation of type-1 or type-2 fuzzy weights are presented. The proposed method is based on recent methods that handle weight adaptation and especially fuzzy weights. In this work neural networks with type-1 fuzzy weights or type-2 fuzzy weights are presented. The proposed approach is applied to the case of Dow-Jones time series prediction for evaluating its efficiency.

1 Introduction

Neural networks have been applied in several areas of research, like in the time series prediction area, which is the study case for this paper, the study case is applied for the Dow-Jones time series.

The approach presented in this paper works with type-1 and type-2 fuzzy weights in the neurons of the hidden and output layers of the neural network used for prediction of the Dow-Jones time series. These type-1 and interval type-2 fuzzy weights are updated using the backpropagation learning algorithm. We used two type-1 inference systems and two type-2 inference systems with Gaussian membership functions for fuzzy weight adjustment.

F. Gaxiola · P. Melin · F. Valdez (✉)
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: fevrier@tectijuana.edu.mx

F. Gaxiola
e-mail: fergaor_29@hotmail.com

P. Melin
e-mail: pmelin@tectijuana.mx

P. Melin et al. (eds.), *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization*,
Studies in Computational Intelligence 601, DOI 10.1007/978-3-319-17747-2_6

The proposed approach is applied to time series prediction for the Dow-Jones time series. The objective is obtaining the minimal prediction error for the data of the time series.

We used a supervised neural network, because this type of network is the most commonly used in the area of time series prediction.

The weights of a neural network are an important part in the training phase, because these affect the performance of the learning process of the neural network.

This paper is focused in the managing of weights, because on the practice of neural networks, when performing the training of neural networks for the same problem is initialized with different weights or the adjustment are in a different way each time it is executed, but at the final is possible to reach a similar result.

The next section presents a background about modifications of the backpropagation algorithm and different management strategies of weights in neural networks, and basic concepts of neural networks. Section 3 explains the proposed method and the problem description. Section 4 describes the neural network with type-1 fuzzy weights proposed in this paper. Section 5 describes the neural network with type-2 fuzzy weights proposed in this paper. Section 6 presents the simulation results for the proposed method. Finally, in Sect. 7 some conclusions are presented.

2 Background and Basic Concepts

In this section a brief review of basic concepts is presented.

2.1 Neural Network

An artificial neural network (ANN) is a distributed computing scheme based on the structure of the nervous system of humans. The architecture of a neural network is formed by connecting multiple elementary processors, this being an adaptive system that has an algorithm to adjust their weights (free parameters) to achieve the performance requirements of the problem based on representative samples [8, 26].

The most important property of artificial neural networks is their ability to learn from a training set of patterns, i.e. they are able to find a model that fits the data [9, 34].

The artificial neuron consists of several parts (see Fig. 1). On one side are the inputs, weights, the summation, and finally the transfer function. The input values are multiplied by the weights and added: $\sum x_i w_{ij}$. This function is completed with the addition of a threshold amount i. This threshold has the same effect as an input with value -1. It serves so that the sum can be shifted left or right of the origin. After addition, we have the f function applied to the sum, resulting in the final value of the output, also called y_i [28], obtaining the following equation:

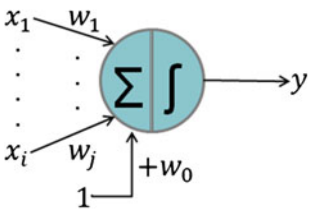

Fig. 1 Schematics of an modular artificial neural network

$$y_i = \mathrm{f}\left(\sum_{i=1}^{n} x_i w_{ij}\right) \tag{1}$$

where f may be a nonlinear function with binary output ±1, a linear function f(z) = z, or as sigmoid logistic function:

$$f(z) = \frac{1}{1 + \mathrm{e}^{-z}}. \tag{2}$$

2.2 Overview of Related Works

The backpropagation algorithm and its variations are the most useful basic training methods in the area of neural networks. However, these algorithms are usually too slow for practical applications.

When applying the basic backpropagation algorithm to practical problems, the training time can be very high. In the literature we can find that several methods have been proposed to accelerate the convergence of the algorithm [2, 18, 28, 37].

There exists many works about adjustment or managing of weights but only the most important and relevant for this research will be mentioned here [4, 10, 31, 36]:

Momentum method—Rumelhart, Hinton and Williams suggested adding in the increased weights expression a momentum term β, to filter the oscillations that can be formed a higher learning rate that lead to great change in the weights [19, 32].

Adaptive learning rate—focuses on improving the performance of the algorithm by allowing the learning rate changes during the training process (increase or decrease) [19].

Conjugate Gradient algorithm—A search of weight adjustment along conjugate directions. Versions of conjugate gradient algorithm differ in the way in which a constant βk is calculated.

- Fletcher-Reeves update [12].
- Polak-Ribiere updated [12].
- Powell-Beale Restart [3, 33].
- Scaled Conjugate Gradient [29].

Kamarthi and Pittner [24], focused in obtaining a weight prediction of the network at a future epoch using extrapolation.

Ishibuchi et al. [21], proposed a fuzzy network, where the weights are given as trapezoidal fuzzy numbers, denoted as four trapezoidal fuzzy numbers for the four parameters of trapezoidal membership functions.

Ishibuchi et al. [22], proposed a fuzzy neural network architecture with symmetrical fuzzy triangular numbers for the fuzzy weights and biases, denoted by the lower, middle and upper limit of the fuzzy triangular numbers.

Feuring [11], based on the work by Ishibuchi, where triangular fuzzy weights are used, developed a learning algorithm in which the backpropagation algorithm is used to compute the new lower and upper limits of weights. The modal value of the new fuzzy weight is calculated as the average of the new computed limits.

Castro et al. [6], use interval type-2 fuzzy neurons for the antecedents and interval of type-1 fuzzy neurons for the consequents of the rules. This approach handles the weights as numerical values to determine the input of the fuzzy zneurons, as the scalar product of the weights for the input vector.

Gaxiola et al. [13–17], proposed at neural network with type-2 fuzzy weights using triangular membership functions.

In addition, recent works on type-2 fuzzy logic have been developed in time series prediction, like that of Castro et al. [6], and other researchers [1, 7].

3 Proposed Method and Problem Description

The objective of this work is to use type-1 and interval type-2 fuzzy sets to generalize the backpropagation algorithm to allow the neural network to handle data with uncertainty. The Dow-Jones time series is utilized for testing the proposed approach.

The updating of the weights will be done differently to the traditional updating of the weights performed with the backpropagation algorithm (Fig. 2).

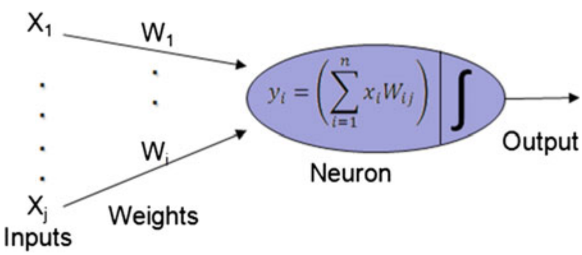

Fig. 2 Scheme of current management of numerical weights (type-0) for the inputs of each neuron

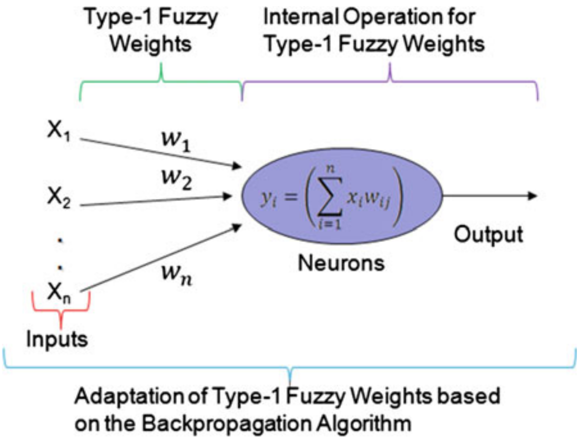

Fig. 3 Schematics of each neuron with the proposed management of weights using type-1 fuzzy sets

The proposed method performs the updating of the weights working with interval type-2 fuzzy weights. This method uses two type-1 and two type-2 inference systems with Gaussian membership functions for obtaining the type-1 and interval type-2 fuzzy weights using in the neural network, and obtaining the outputs taking into account the possible change in the way we work internally in the neuron, and the adaptation of the weights given in this way (Figs. 3 and 4) [30].

We developed a method for adjusting weights to achieve the desired result, searching for the optimal way to work with type-1 fuzzy weights or type-2 fuzzy weights [23].

We used the sigmoid activation function for the hidden neurons and the linear activation function for the output neurons, and we utilized this activation functions because these functions have obtained good results in similar approaches.

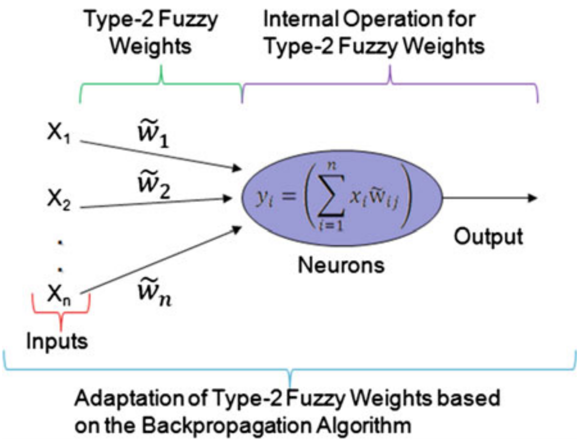

Fig. 4 Schematics of each neuron with the proposed management of weights using interval type-2 fuzzy sets

4 Neural Network Architecture with Type-1 Fuzzy Weights

The proposed neural network architecture with type-1 fuzzy weights (see Fig. 5) is described as follows:

Layer 0 Inputs.

$$x = [x_1, x_2, \ldots, x_n] \tag{3}$$

Layer 1 type-1 fuzzy weights for the hidden layer.

$$w_{ij} \tag{4}$$

Layer 2 Equations of the calculations in the hidden neurons using type-1 fuzzy weights.

$$Net = \sum_{i=1}^{n} x_i w_i \tag{5}$$

Layer 3 Equations of the calculations in the outputs neurons using type-1 fuzzy weights.

$$Out = \sum_{i=1}^{n} y_i w_i \tag{6}$$

Layer 4 Obtain the output of the neural network.

We considered a neural network architecture with 1 neuron in the output layer and 30 neurons in the hidden layer.

This neural network uses two type-1 fuzzy inference systems, one in the connections between the input neurons and the hidden neurons, and the other in the connections between the hidden neurons and the output neuron. In the hidden layer and output layer of the network we are updating the weights using the two type-1

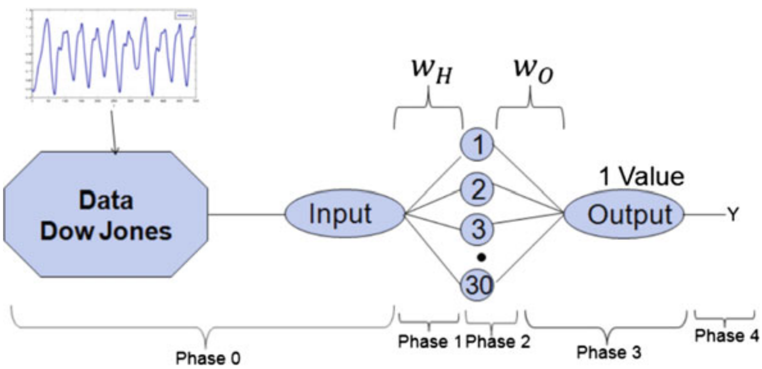

Fig. 5 Proposed neural network architecture with type-1 fuzzy weights

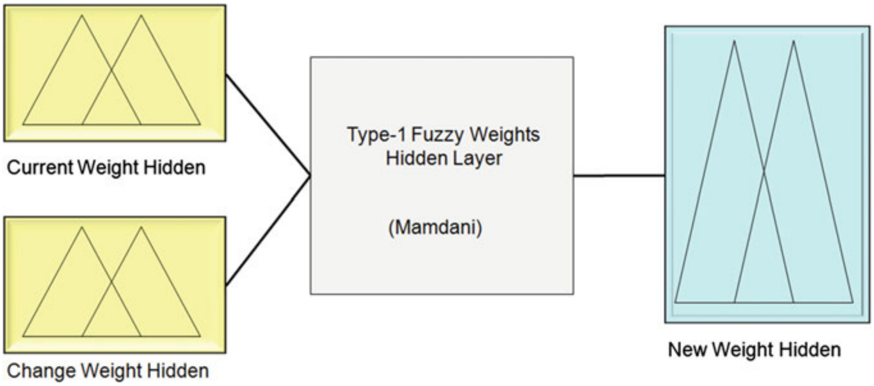

Fig. 6 Structure of the two type-1 fuzzy inference systems that are used to obtain the type-1 fuzzy weights in the hidden and output layer

fuzzy inference system that obtains the new weights in each epoch of the network on base at the backpropagation algorithm.

The two type-2 fuzzy inference systems have the same structure and consist of two inputs (the current weight in the actual epoch and the change of the weight for the next epoch) and one output (the new weight for the next epoch) (see Fig. 6).

We used two Gaussian membership functions with their corresponding range for delimiting the inputs and outputs of the two type-1 fuzzy inference systems (see Figs. 7 and 8).

We obtain the two type-1 fuzzy inference systems empirically.

The two type-1 fuzzy inference systems used the same six rules, the four combinations of the two Gaussian membership function and two rules added for null change of the weight (see Fig. 9).

5 Neural Network Architecture with Type-2 Fuzzy Weights

The proposed neural network architecture with interval type-2 fuzzy weights (see Fig. 10) is described as follows:

Layer 0 Inputs.

$$x = [x_1, x_2, \ldots, x_n] \tag{7}$$

Layer 1 Interval type-2 fuzzy weights for the connection between the input and the hidden layer of the neural network.

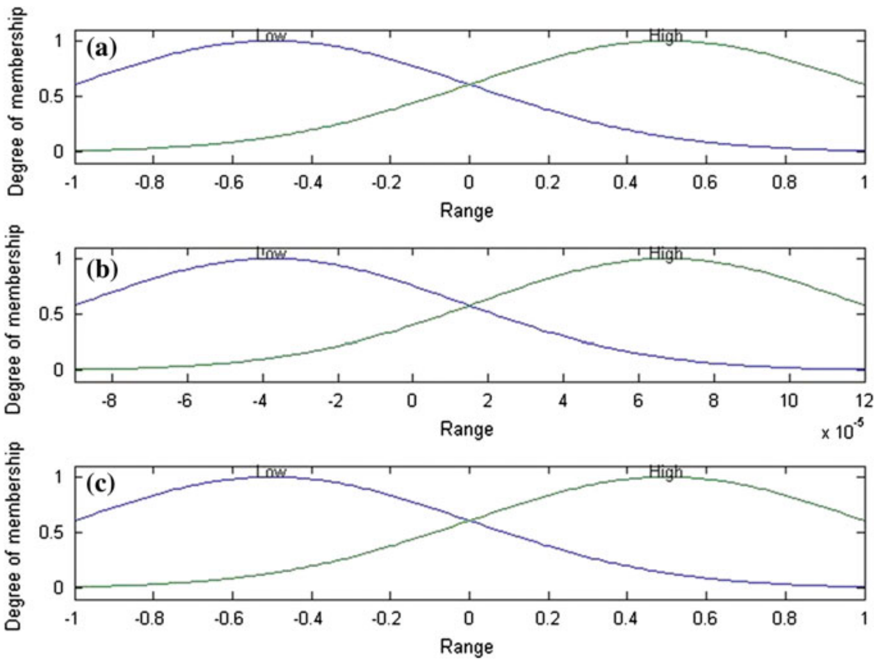

Fig. 7 Inputs (**a** current weight and **b** change of weight) and output (**c** new weight) of the type-1 fuzzy inference systems that are used to obtain the type-1 fuzzy weights in the hidden layer

$$\tilde{w}_{ij} = \left[\bar{w}_{ij}, \underline{w}_{ij}\right] \tag{8}$$

where $\tilde{w}_{ij}$ are the weights of the consequents of each rule of the type-2 fuzzy system with inputs (current type-2 fuzzy weight, change of weight) and output (new fuzzy weight).

Layer 2 Equations of the calculations in the hidden neurons using interval type-2 fuzzy weights.

$$Net = \sum_{i=1}^{n} x_i \tilde{w}_{ij} \tag{9}$$

Layer 3 Equations of the calculations in the output neurons using interval type-2 fuzzy weights.

$$Out = \sum_{i=1}^{n} y_i \tilde{w}_{ij} \tag{10}$$

Layer 4 Obtain a single output of the neural network.

We applied the same neural network architecture used in the type-1 fuzzy weights for the type-2 fuzzy weights (see Fig. 6).

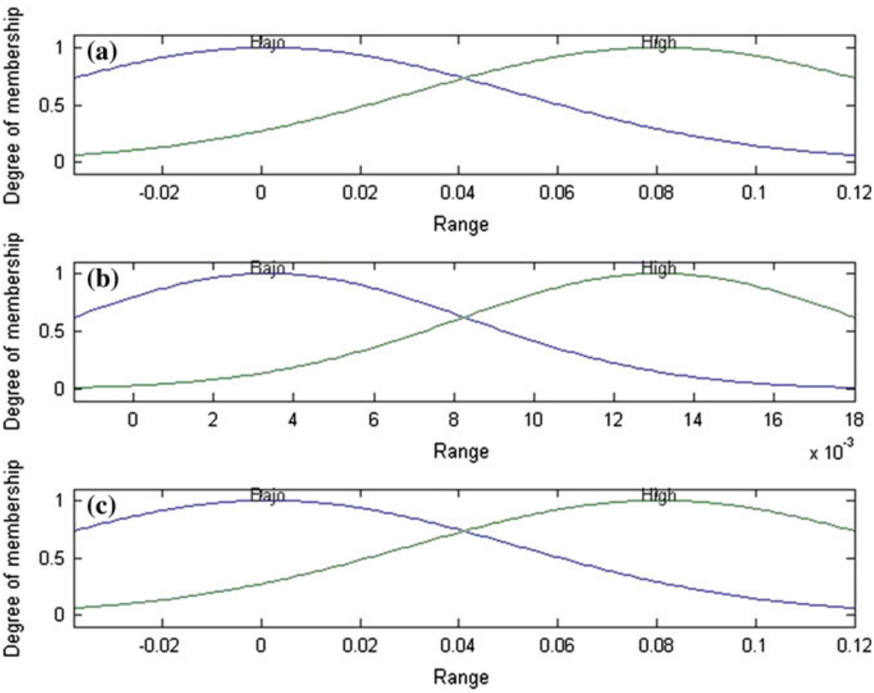

Fig. 8 Inputs (**a** current weight and **b** change of weight) and output (**c** new weight) of the type-1 fuzzy inference systems that are used to obtain the type-1 fuzzy weights in the output layer

1. (Current_Weight is lower) and (Change_Weight is lower) then (New_Weight is lower)
2. (Current_Weight is lower) and (Change_Weight is upper) then (New_Weight is lower)
3. (Current_Weight is upper) and (Change_Weight is lower) then (New_Weight is upper)
4. (Current_Weight is upper) and (Change_Weight is upper) then (New_Weight is upper)
5. (Current_Weight is lower) then (New_Weight is lower)
6. (Current_Weight is upper) then (New_Weight is upper)

Fig. 9 Rules of the type-1 fuzzy inference system used in the hidden and output layer for the neural network with type-1 fuzzy weights

We used two type-2 fuzzy inference systems to obtain the type-2 fuzzy weights and work in the same way like with the type-1 fuzzy weights.

The structure and rules (see Fig. 9) of the two type-2 fuzzy inference systems are the same of the type-1 fuzzy inference systems, the difference is in the memberships functions, Gaussian membership functions for type-2 [5, 20, 27, 35].

We used two Gaussian membership functions with their corresponding range for delimiting the inputs and outputs of the two type-2 fuzzy inference systems (see Figs. 11 and 12).

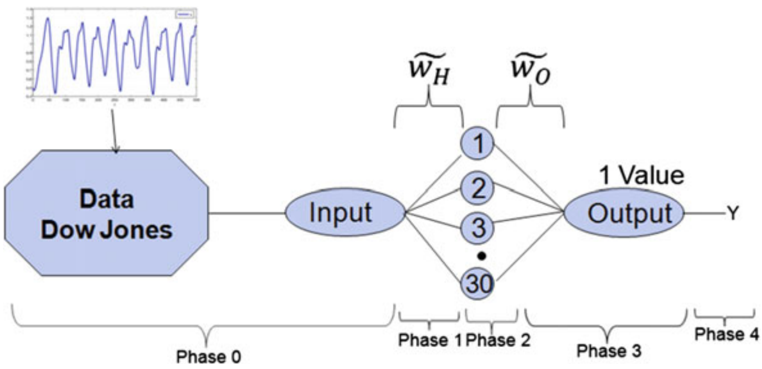

Fig. 10 Proposed neural network architecture with interval type-2 fuzzy weights

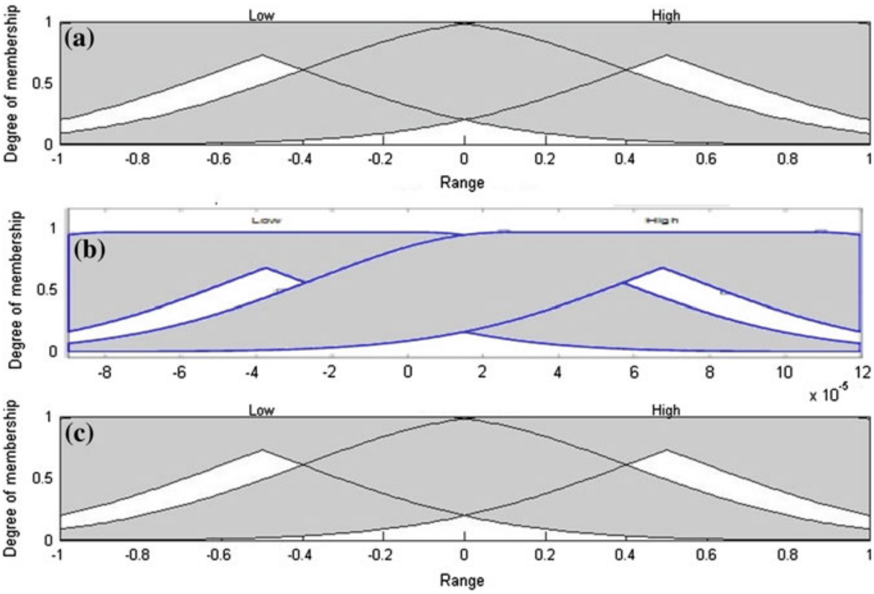

Fig. 11 Inputs (**a** current weight and **b** change of weight) and output (**c** new weight) of the type-2 fuzzy inference systems that are used to obtain the type-2 fuzzy weights in the hidden layer

We obtain the type-2 fuzzy inference systems incrementing and decrementing 20 percent the values of the centers of the Gaussian membership functions and the same standard deviation of the type-1 Gaussians membership functions, we use this method to obtain the footprint of uncertainty (FOU) for the type-2 fuzzy inference systems used in the neural network with type-2 fuzzy weights.

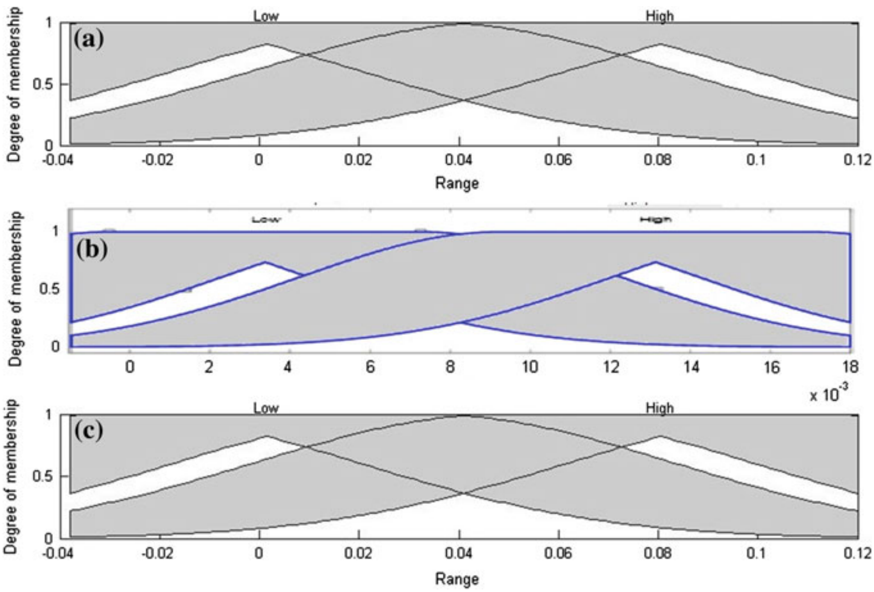

Fig. 12 Inputs (**a** current weight and **b** change of weight) and output (**c** new weight) of the type-2 fuzzy inference system that are used to obtain the type-2 fuzzy weights in the output layer

6 Simulation Results

We performed experiments in time-series prediction, specifically for the Dow-Jones time series.

We presented the obtained results of the experiments performed with the neural network with type-1 fuzzy weights (NNT1FW) and the neural network with type-1 fuzzy weights (NNT1FW), these results are achieved without optimizing of the neural network and the type-1 fuzzy systems, which means that all parameters of the neural network and the range and values of the membership functions of the type-1 fuzzy systems are established empirically. The average error was obtained of 30 experiments.

In Table 1, we present the prediction error obtained with the results achieved as output of NNT1FW. The best prediction error is of 0.0097 and the average prediction error is of 0.0201.

In Fig. 13 we show prediction data with type-1 fuzzy weights against the test data of the Dow-Jones time series

In Table 2, we present the prediction error obtained with the results achieved as output of NNT2FW. The best prediction error is of 0.0080 and the average prediction error is of 0.0104.

In Fig. 14 we show the prediction data with type-2 fuzzy weights against the test data of the Dow-Jones time series.

Table 1 Prediction error for the neural network with type-1 fuzzy weights for Dow-Jones time series

No.	Epochs	Network error	Prediction error
E1	100	1×10^{-8}	0.0121
E2	100	1×10^{-8}	0.0195
E3	100	1×10^{-8}	0.0198
E4	***100***	$\boldsymbol{1 \times 10^{-8}}$	***0.0097***
E5	100	1×10^{-8}	0.0238
E6	100	1×10^{-8}	0.0213
E7	100	1×10^{-8}	0.0231
E8	100	1×10^{-8}	0.0171
E9	100	1×10^{-8}	0.0216
E10	100	1×10^{-8}	0.0147
Average prediction error			0.0201

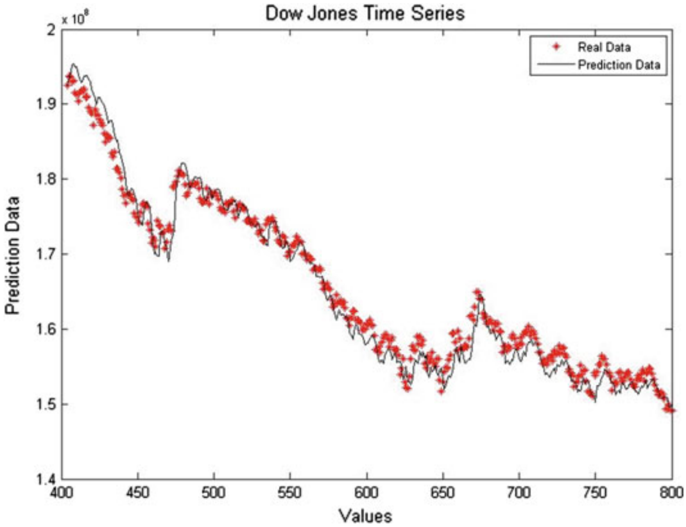

Fig. 13 Plot of the prediction data with NNT1FW against the test data of the Dow-Jones time series

7 Conclusions

In this paper, we proposed a new learning method that updates weights (type-1 or type-2 fuzzy weights) in each connection between the neurons of the layers of neural network using a type-1 or type-2 fuzzy inference system with Gaussians membership functions applied to the Dow-Jones time series.

Additionally, the neurons work internally with the type-1 or type-2 fuzzy weights and therefore, obtaining results at the output of each neuron of the neural network. The modifications performed in the neural network, that allows working

Table 2 Prediction error for the neural network with interval type-2 fuzzy weights for the Dow-Jones time series

No.	Epochs	Network error	Prediction error
E1	100	1×10^{-8}	0.0105
E2	100	1×10^{-8}	0.0082
E3	100	1×10^{-8}	0.0125
E4	100	1×10^{-8}	0.0114
E5	100	1×10^{-8}	0.0108
E6	100	1×10^{-8}	0.0095
E7	***100***	$\boldsymbol{1 \times 10^{-8}}$	***0.0080***
E8	100	1×10^{-8}	0.0101
E9	100	1×10^{-8}	0.0093
E10	100	1×10^{-8}	0.0117
Average prediction error			0.0104

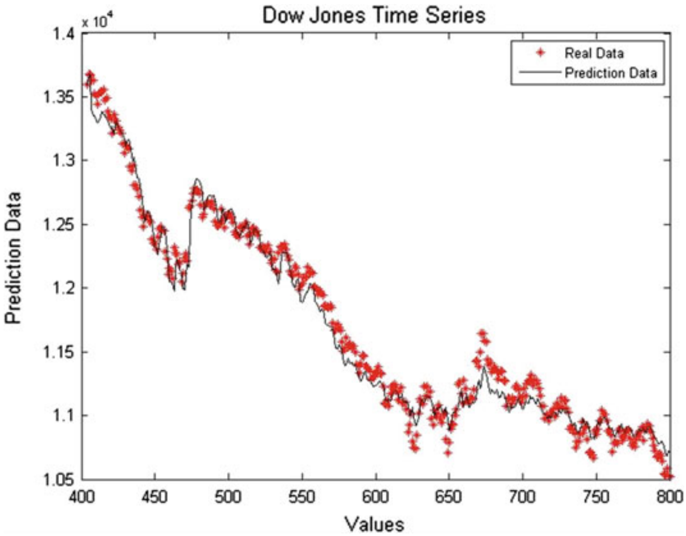

Fig. 14 Plot of the prediction data with NNT2FW against the test data of the Dow-Jones time series

with type-1 or type-2 fuzzy weights, provide the neural network with greater robustness and less susceptibility at the noise in the data of the time series.

The prediction error of 0.0080 of the neural network with type-2 fuzzy weights for the Dow-Jones time series is better than the prediction error of 0.0097 of the neural network with type-1 fuzzy weights (as shown in Tables 1 and 2).

This result is good considering that the used parameters for the neural networks at the moment are determined in an empirical way.

References

1. Abiyev, R.: A Type-2 Fuzzy Wavelet Neural Network for Time Series Prediction. Lecture Notes in Computer Science, vol. 6098, pp. 518–527 (2010)
2. Barbounis, T.G., Theocharis, J.B.: Locally recurrent neural networks for wind speed prediction using spatial correlation. Inf. Sci. **177**(24), 5775–5797 (2007)
3. Beale, E.M.L.: A derivation of conjugate gradients. In: Lootsma, F.A. (ed.) Numerical Methods for Nonlinear Optimization, pp. 39–43. Academic Press, London (1972)
4. Casasent, D., Natarajan, S.: A classifier neural net with complex-valued weights and square-law nonlinearities. Neural Netw. **8**(6), 989–998 (1995)
5. Castillo, O., Melin, P.: A review on the design and optimization of interval type-2 fuzzy controllers. Appl. Soft Comput. **12**(4), 1267–1278 (2012)
6. Castro, J., Castillo, O., Melin, P., Rodríguez-Díaz, A.: A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. Inf. Sci. **179**(13), 2175–2193 (2009)
7. Castro, J., Castillo, O., Melin, P., Mendoza, O., Rodríguez-Díaz, A.: An interval type-2 fuzzy neural network for chaotic time series prediction with cross-validation and Akaike test". In: Soft Computing for Intelligent Control and Mobile Robotics, pp. 269–285 (2011)
8. Cazorla, M., Escolano, F.: Two Bayesian methods for junction detection. IEEE Trans. Image Proc. **12**(3), 317–327 (2003)
9. De Wilde, O.: The magnitude of the diagonal elements in neural networks. Neural Netw. **10** (3), 499–504 (1997)
10. Draghici, S.: On the capabilities of neural networks using limited precision weights. Neural Netw. **15**(3), 395–414 (2002)
11. Feuring, T.: Learning in fuzzy neural networks. In: IEEE international conference on neural networks, vol. 2, pp. 1061–1066 (1996)
12. Fletcher, R., Reeves, C.M.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)
13. Gaxiola, F., Melin, P., Valdez, F., Castillo, O.: Neural network with type-2 fuzzy weights adjustment for pattern recognition of the human Iris biometrics. In: Proceedings of the 11th Mexican international conference on advances in computational intelligence, MICAI '12, vol. 2, pp. 259–270 (2012)
14. Gaxiola, F., Melin, P., Valdez, F., Castillo, O.: Optimization of type-2 fuzzy weight for neural network using genetic algorithm and particle swarm optimization. In: 5th World Congress on Nature and Biologically Inspired Computing, pp. 22–28 (2013)
15. Gaxiola, F., Melin, P., Valdez, F.: Backpropagation method with type-2 fuzzy weight adjustment for neural network learning. In: Fuzzy Information Processing Society (NAFIPS). Annual meeting of the North American, pp. 1–6 (2012)
16. Gaxiola, F., Melin, P., Valdez, F.: Genetic optimization of type-2 fuzzy weight adjustment for backpropagation ensemble neural network. In: Recent Advances on Hybrid Intelligent Systems, pp. 159–171 (2013)
17. Gaxiola, F., Melin, P., Valdez, F.: Neural network with lower and upper type-2 fuzzy weights using the backpropagation learning method. In: IFSA World Congress and NAFIPS annual meeting (IFSA/NAFIPS), pp. 637–642 (2013)
18. Gedeon, T.: Additive neural networks and periodic patterns. Neural Netw. **12**(4–5), 617–626 (1999)
19. Hagan, M.T., Demuth, H.B., Beale, M.H.: Neural Network Design. PWS Publishing, Boston, p. 736 (1996)
20. Hagras, H.: Type-2 fuzzy logic controllers: a way forward for fuzzy systems in real world environments. In: IEEE World Congress on Computational Intelligence, pp. 181–200 (2008)
21. Ishibuchi, H., Morioka, K., Tanaka, H.: A fuzzy neural network with trapezoid fuzzy weights, fuzzy systems. In: IEEE World Congress on Computational Intelligence, vol. 1, pp. 228–233 (1994)

22. Ishibuchi, H., Tanaka, H., Okada, H.: Fuzzy neural networks with fuzzy weights and fuzzy biases. In: IEEE international conference on neural networks, vol. 3, pp. 1650–165 (1993)
23. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence, p. 614. Prentice-Hall, Inc., New Jersey (1997)
24. Kamarthi, S., Pittner, S.: Accelerating neural network training using weight extrapolations. Neural Netw. **12**(9), 1285–1299 (1999)
25. Karnik, N., Mendel, J.: Applications of type-2 fuzzy logic systems to forecasting of time-series. Inf. Sci. **120**(1–4), 89–111 (1999)
26. Martinez, G., Melin, P., Bravo, D., Gonzalez, F., Gonzalez, M.: Modular neural networks and fuzzy sugeno integral for face and fingerprint recognition. Adv. Soft Comput. **34**, 603–618 (2006)
27. Melin, P.: Modular Neural Networks and Type-2 Fuzzy Systems for Pattern Recognition, pp. 1–204. Springer, Berlin (2012)
28. Meltser, M., Shoham, M., Manevitz, L.: Approximating functions by neural networks: a constructive solution in the uniform norm. Neural Netw. **9**(6), 965–978 (1996)
29. Moller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. **6**, 525–533 (1993)
30. Monirul Islam, M.D., Murase, K.: A new algorithm to design compact two-hidden-layer artificial neural networks. Neural Netw. **14**(9), 1265–1278 (2001)
31. Neville, R.S., Eldridge, S.: Transformations of Sigma–Pi nets: obtaining reflected functions by reflecting weight matrices. Neural Netw. **15**(3), 375–393 (2002)
32. Phansalkar, V.V., Sastry, P.S.: Analysis of the back-propagation algorithm with momentum. IEEE Trans. Neural Netw. **5**(3), 505–506 (1994)
33. Powell, M.J.D.: Restart procedures for the conjugate gradient method. Math. Program. **12**, 241–254 (1977)
34. Salazar, P.A., Melin, P., Castillo, O.: A new biometric recognition technique based on hand geometry and voice using neural networks and fuzzy logic. In: Soft Computing for Hybrid Intelligent Systems, pp. 171–186 (2008)
35. Sepúlveda, R., Castillo, O., Melin, P., Montiel, O.: An efficient computational method to implement type-2 fuzzy logic in control applications. In: Analysis and Design of Intelligent Systems using Soft Computing Techniques, pp. 45–52 (2007)
36. Yam, J., Chow, T.: A weight initialization method for improving training speed in feedforward neural network. Neurocomputing **30**(1–4), 219–232 (2000)
37. Yeung, D., Chan, P., Ng, W.: Radial basis function network learning using localized generalization error bound. Inf. Sci. **179**(19), 3199–3217 (2009)