information

MDPI

Article

Comparison of T-Norms and S-Norms for Interval Type-2 Fuzzy Numbers in Weight Adjustment for Neural Networks

Fernando Gaxiola [1], Patricia Melin [2], Fevrier Valdez [2], Oscar Castillo [2,*] and Juan R. Castro [3]

[1] Faculty of Engineering, Autonomous University of Chihuahua, 31110 Chihuahua, Mexico; fergaor_29@hotmail.com

[2] Division of Graduate Studies and Research, Tijuana Institute of Technology, 22414 Tijuana, Mexico; pmelin@tectijuana.mx (P.M.); fevrier@tectijuana.edu.mx (F.V.)

[3] Faculty of Engineering and Chemistry Sciences, Autonomous University of Baja California, 22390 Tijuana, Mexico; jrcastror@uabc.edu.mx

* Correspondence: ocastillo@tectijuana.mx; Tel.: +52-664-623-6318

Received: 29 August 2017; Accepted: 18 September 2017; Published: 20 September 2017

Abstract: A comparison of different T-norms and S-norms for interval type-2 fuzzy number weights is proposed in this work. The interval type-2 fuzzy number weights are used in a neural network with an interval backpropagation learning enhanced method for weight adjustment. Results of experiments and a comparative research between traditional neural networks and the neural network with interval type-2 fuzzy number weights with different T-norms and S-norms are presented to demonstrate the benefits of the proposed approach. In this research, the definitions of the lower and upper interval type-2 fuzzy numbers with random initial values are presented; this interval represents the footprint of uncertainty (FOU). The proposed work is based on recent works that have considered the adaptation of weights using type-2 fuzzy numbers. To confirm the efficiency of the proposed method, a case of data prediction is applied, in particular for the Mackey-Glass time series (for $\tau = 17$). Noise of Gaussian type was applied to the testing data of the Mackey-Glass time series to demonstrate that the neural network using a interval type-2 fuzzy numbers method achieves a lower susceptibility to noise than other methods.

Keywords: fuzzy numbers; type-2 fuzzy weights; neural networks; backpropagation; time series prediction

1. Introduction

In the literature there exists research based on a similar idea to this paper, but with different approaches and implementations, such as the adjustment of fuzzy number weights in the input and output layer in the training process for the neural network [1], or the proposal of fuzzy number operations in a fuzzy neural network [2], and also because the proposed work operates with interval type-2 fuzzy numbers weights using different T-norm and S-norm in the adaptation of the weights, which represent the contribution and main difference with respect to the methods in the literature [3–6].

The proposed method in the present research is different to other papers, such as in Gaxiola et al. [7,8], where the fuzzy weights are obtained using interval type-2 fuzzy inference systems in [7] and generalized type-2 fuzzy inference systems in [8] for the connections between the layers, and without any changes for obtaining the change of the weights for each epoch of the backpropagation algorithm.

In the present approach, the use of interval type-2 fuzzy number weights is proposed. The lower and upper fuzzy numbers are obtained with the Nguyen-Widrow algorithm for the initial weights, modifying the internal calculations of the neurons by performing the multiplications of the inputs for the lower and upper type-2 fuzzy numbers weights separately, and then applying the T-norm for the lower outputs and the S-norm for the upper outputs; furthermore, we modified the backpropagation algorithm to achieve the lower change of the weights and the upper change of the weights in each epoch, respectively.

The proposed approach has the goal in the data of time series of achieving the best prediction error, which is the minimal error. In this case, the prediction for the Mackey-Glass time series is utilized to verify the efficiency of the proposed approach.

This paper is focused on analyzing fuzzy neural networks with the interval type-2 fuzzy number weights and providing a comparison with respect to the traditional neural networks. A same learning algorithm is used for the two neural models. On the other hand, with the purpose of further analyzing the performance of the model, we also applied noise in the real test data.

A comparison of the performance for the traditional neural network against the proposed fuzzy neural network with interval type-2 fuzzy numbers weights is performed in this paper. This comparison is based on the use of fuzzy numbers for the weights instead of the real numbers utilized in the traditional neural network; this modification is of great importance, due to the fact that the learning process of a neural network is directly affected by obtaining the optimal weights and consequently this has a critical impact on obtaining better results [9–11].

In the fuzzy neural network with interval type-2 fuzzy numbers weights, different T-norms and S-norms are applied for obtaining prediction error results, like the sum-product, Dombi [12], Hamacher [13,14] and Frank [15].

The adjustment of the weights in the backpropagation learning using interval type-2 fuzzy numbers is the main contribution of the proposed work in this paper for neural networks. This contribution provides to the neural network the robustness to support real data with uncertainty [16–18].

The main contribution of the proposed work is to improve backpropagation learning with the use of lower and upper type-2 fuzzy numbers for the adaptation for the weights. The use of the T-Norm and S-Norm to obtain the outputs of the neurons with the approaches of the sum-product, Dombi, Hamacher and Frank, enables the achievement of a better support for the uncertainty in the training process. With this, better results can be accomplished [19,20].

The next section presents a background of research on fuzzy numbers, and other methods of adaptation of weights and previous work of modifications to the backpropagation learning in neural networks. Section 3 explains the proposed methodology and the description of the problem to solve in the paper. Section 4 presents the simulation results for the proposed approach and the statistical tests. Finally, in Section 5, conclusions are offered and future work outlined.

2. Related Work

In the neural networks area, the backpropagation algorithm and variations of it is the training method that researchers use in literature [21–23]. In the bibliography, several papers have proposed different methods to improve the convergence of the backpropagation training algorithm [3,4,6]. In this paper, the most significant and essential works about the representation or managing of fuzzy numbers will be reviewed [9–11].

Dunyak et al. [1] presented a new algorithm for obtaining new weights (inputs and outputs) in the phase of training with any type of fuzzy numbers for a fuzzy neural network. Fard et al. [24] presented the sum and the product of two interval type-2 triangular fuzzy numbers and, basing it on the Stone-Weierstrass theorem, a fuzzy neural network working with interval type-2 fuzzy logic is developed. Li Z. et al. [2] described a fuzzified neural network computing the results of operations in two fuzzy numbers such as addition, subtraction, multiplication and division.

Asady B. [25] outlined a method for approximating trapezoidal fuzzy numbers in comparison with other methods of approximation. Coroianu L. et al. [26] described the inverse F-transform to accomplish optimal fuzzy numbers, maintaining the support and the convergence of the core.

Yang D. et al. [27] presented an interval and modified interval neuron perceptron with interval weights and biases, and modified the learning algorithm for this approach.

Requena et al. [28] presented trapezoidal fuzzy numbers with the conventional parameters (a, b, c, d) in an artificial neural network, and also proposed a decision personal index (DPI) to obtain the distance between the numbers. Kuo et al. [29] described a fuzzy neural network using the real-coded genetic algorithm to generate the initial fuzzy weights, and using the extension principle, the fuzzy operations are determined. Molinari [30] presented generalized triangular fuzzy numbers and a comparison with others fuzzy numbers. Chai et al. [31] described a representation of fuzzy numbers, establishing the theorem "that for each fuzzy number there exists a unique skew fuzzy number and a unique symmetric fuzzy number".

Figueroa-García et al. [32] made a comparison between interval type-2 fuzzy numbers using distance measures. Ishibuchi et al. [33] presented a comparison between real numbers, and different fuzzy numbers such as symmetric triangular, asymmetric triangular and symmetric trapezoidal as weights in the connections between layers in a neural network. Karnik et al. [34] presented mathematical operations of type-2 fuzzy sets for obtaining the join and meet under t-norm.

Raj et al. [35] described fuzzy ranking alternatives for fuzzy numbers as linguistic variables for fuzzy weights. Chu et al. [36] proposed a ranking of fuzzy numbers with a zone between the original point and the centroid point and making numerical examples with triangular fuzzy numbers.

Ishibuchi et al. [37,38] proposed to use the weights for a fuzzy neural network like triangular or trapezoidal fuzzy numbers. Feuring [39] presented a new backpropagation algorithm for learning in the neural network, in which the new lower and upper limits of weights are computed. Castro et al. [40] proposed a type-2 fuzzy neurons model, in which the rules used interval type-2 fuzzy neurons in the antecedents and an interval of type-1 fuzzy neuron in the consequents.

In the area of time series prediction there exists in the literature many recent works developed by and based on the use of type-2 fuzzy logic, like Castro et al. [41], and other researchers [42–44].

3. Proposed Methodology

The proposed method in this work has the goal of generalizing the backpropagation learning algorithm by using interval type-2 fuzzy numbers in the calculations, and this approach gives the neural network less susceptibility to data with uncertainty. In interval type-2 fuzzy numbers, it will be necessary to obtain the interval of the fuzzy numbers, which consists in the footprint of uncertainty (FOU), and the calculations in the neurons was obtained with the T-norms and S-norms of sum-product, Dombi, Hamacher and Frank for the corresponding applications [45–48].

The method of adjustment of weights for each neuron in the connections between the layers in the backpropagation learning algorithm is modified from the original adjustment (Figure 1).

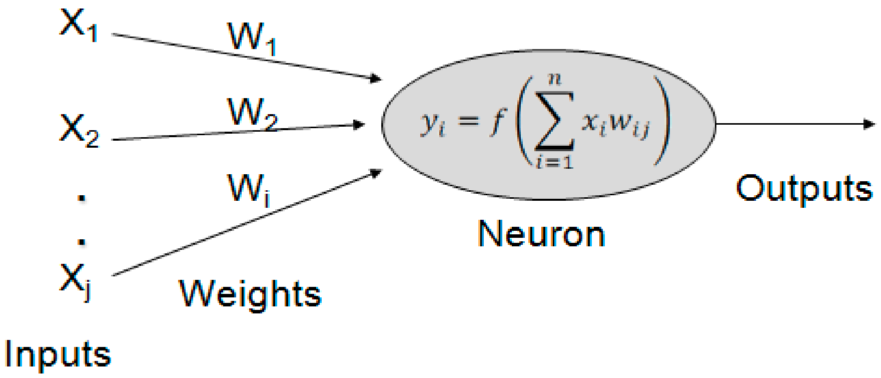

Figure 1. Structure and equation of a neuron with traditional numerical weights.

The method in this paper consists of utilizing interval type-2 fuzzy number weights in neurons. This development modifies the internal calculation for the neurons and the adjustment of the weights to allow handle fuzzy numbers (Figure 2) [49].

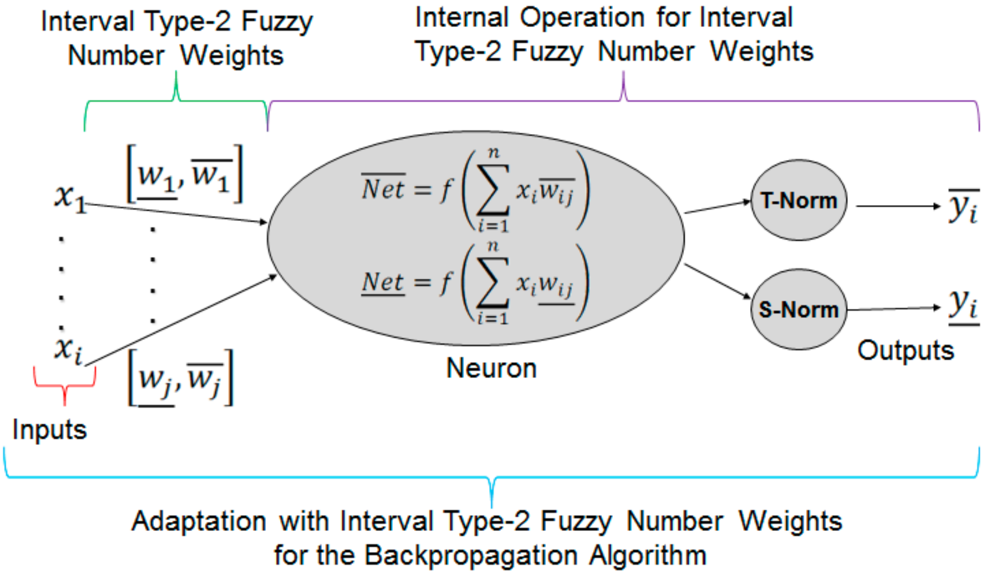
Interval Type-2 Fuzzy Number Weights
Internal Operation for Interval Type-2 Fuzzy Number Weights
x_1
$[\underline{w_1}, \overline{w_1}]$
$\overline{Net} = f\left(\sum_{i=1}^{n} x_i \overline{w_{ij}}\right)$
$\underline{Net} = f\left(\sum_{i=1}^{n} x_i \underline{w_{ij}}\right)$
T-Norm
$\overline{y_i}$
S-Norm
$\underline{y_i}$
x_i
$[\underline{w_j}, \overline{w_j}]$
Neuron
Outputs
Inputs
Adaptation with Interval Type-2 Fuzzy Number Weights for the Backpropagation Algorithm

Figure 2. Scheme of the proposed structure and equations of the neuron with interval type-2 fuzzy number weights.

We modified the operations in the neurons and the backpropagation learning to adjust the fuzzy numbers weights and accomplish the desired result, working to find the optimal process to operate with interval type-2 fuzzy number weights [50,51].

To determine the appropriate activation function *f*(-) to utilize, the linear and secant hyperbolic functions were considered in this approach.

3.1. Architecture of the Traditional Neural Network

The architecture of the neural network used in this work (see Figure 3) consists of a hidden layer with 16 neurons and of an output layer with 1 neuron, and a training data of the Mackey-Glass time series to the input data in the input layer.

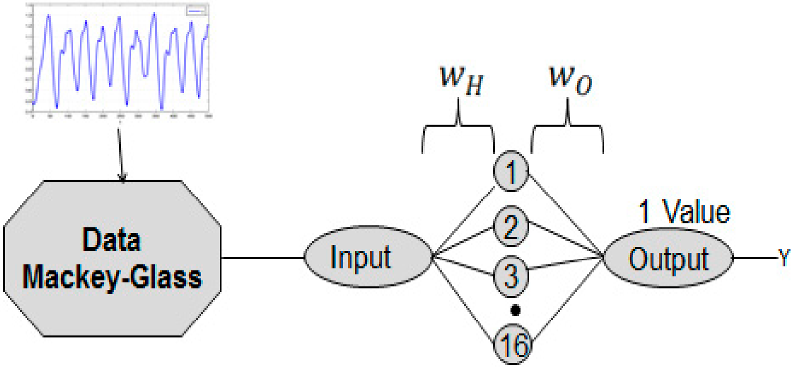

Figure 3. Illustration of the architecture of the neural network.

3.2. Architecture of the Fuzzy Neural Network with Interval Type-2 Fuzzy Numbers Weights

In Figure 4a scheme of the proposed methodology of the fuzzy neural network with interval type-2 fuzzy numbers weights (FNNIT2FNW) is presented.

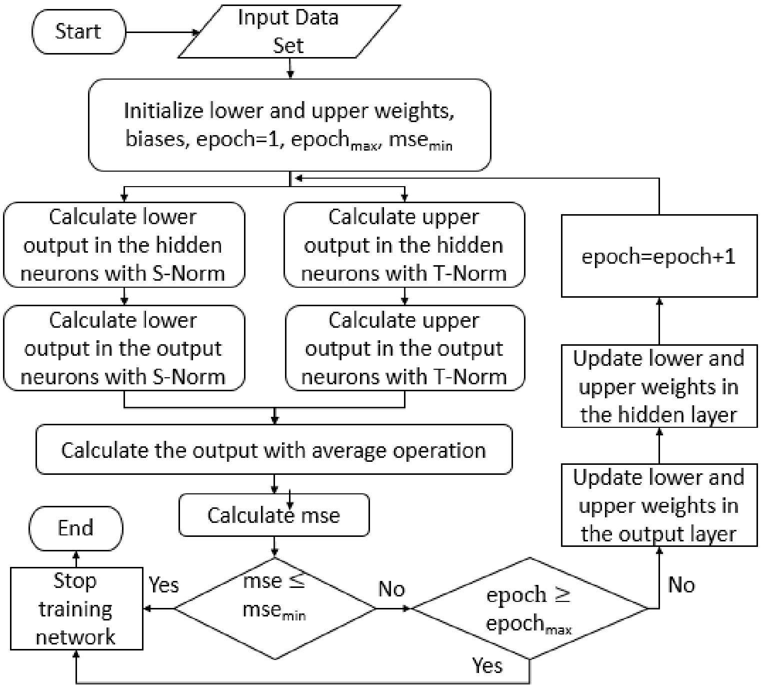
Start
Input Data Set
Initialize lower and upper weights, biases, epoch=1, epoch$_{max}$, mse$_{min}$
Calculate lower output in the hidden neurons with S-Norm
Calculate upper output in the hidden neurons with T-Norm
epoch=epoch+1
Calculate lower output in the output neurons with S-Norm
Calculate upper output in the output neurons with T-Norm
Update lower and upper weights in the hidden layer
Calculate the output with average operation
Update lower and upper weights in the output layer
End
Calculate mse
Stop training network
Yes
mse ≤ mse$_{min}$
No
epoch ≥ epoch$_{max}$
No
Yes

Figure 4. Flow chart of the FNNIT2FNW.

The architecture of fuzzy neural network with interval type-2 fuzzy number weights (see Figure 5) is explained as follow:

Phase 0: equation of the inputs data.

$$x = [x_1,\ x_2, \ldots,\ x_n] \tag{1}$$

Phase 1: Representation of the Interval type-2 fuzzy number weights [36].

$$\tilde{w} = [\underline{w}, \overline{w}] \tag{2}$$

where $\overline{w}$ and $\underline{w}$ are generated with the Nguyen-Widrow algorithm [52] for the initial weights.

Phase 2: Calculation of the output of the hidden neurons with interval type-2 fuzzy number weights.

$$\underline{Net} = f\left(\sum_{i=1}^{n} x_i \underline{w_{ij}}\right) \tag{3}$$

$$\overline{Net} = f\left(\sum_{i=1}^{n} x_i \overline{w_{ij}}\right) \tag{4}$$

We used the secant hyperbolic as the activation function for the hidden neurons. Subsequently, we applied the S-Norm and T-Norm for calculating the lower and upper outputs of the hidden neurons, respectively.

$$\underline{y} = TNorm\left(\underline{Net}, \overline{Net}\right) \tag{5}$$

$$\overline{y} = SNorm\left(\underline{Net}, \overline{Net}\right) \tag{6}$$

where: we used the T-norms and S-norms.

Sum-Product:

$$TNorm\left(\underline{Net}, \overline{Net}\right) = \underline{Net}\ .*\ \overline{Net} \tag{7}$$

$$SNorm\left(\underline{Net}, \overline{Net}\right) = \underline{Net} + \overline{Net} - TNorm\left(\underline{Net}, \overline{Net}\right) \tag{8}$$

Dombi: for $\gamma > 0$.

$$TNormD\left(\underline{Net},\overline{Net},\gamma\right) = \frac{1}{1+\left[\left(\underline{Net}^{-1}-1\right)^{\gamma}+\left(\overline{Net}^{-1}-1\right)^{\gamma}\right]^{1/\gamma}} \tag{9}$$

$$SNormD\left(\underline{Net},\overline{Net},\gamma\right) = \frac{1}{1+\left[\left(\underline{Net}^{-1}-1\right)^{-\gamma}+\left(\overline{Net}^{-1}-1\right)^{-\gamma}\right]^{-1/\gamma}} \tag{10}$$

Hamacher: for $\gamma > 0$.

$$TNormH\left(\underline{Net},\overline{Net},\gamma\right) = \frac{\underline{Net}\ .*\overline{Net}}{\gamma+(1-\gamma)\left(\underline{Net}+\overline{Net}-\underline{Net}\ .*\overline{Net}\right)} \tag{11}$$

$$SNormH\left(\underline{Net},\overline{Net},\gamma\right) = \frac{\underline{Net}+\overline{Net}+(\gamma-2)\left(\underline{Net}\ .*\overline{Net}\right)}{1+(\gamma-1)\left(\underline{Net}\ .*\overline{Net}\right)} \tag{12}$$

Frank: for $s > 0$.

$$TNormF\left(\underline{Net},\overline{Net},s\right) = log_s\left[\frac{1+\left(s^{\underline{Net}}-1\right)\left(s^{\overline{Net}}-1\right)}{s-1}\right] \tag{13}$$

$$SNormF\left(\underline{Net},\overline{Net},s\right) = 1-log_s\left[\frac{1+\left(s^{1-\underline{Net}}-1\right)\left(s^{1-\overline{Net}}-1\right)}{s-1}\right] \tag{14}$$

Phase 3: Calculation of the output for the output neuron with interval type-2 fuzzy number weights.

$$\underline{Out} = f\left(\sum_{i=1}^{n}\underline{y_i}\underline{w_{ij}}\right) \tag{15}$$

$$\overline{Out} = f\left(\sum_{i=1}^{n}\overline{y_i w_{ij}}\right) \tag{16}$$

In the output neuron the linear activation function is utilized. Subsequently, we applied the S-Norm and T-Norm for the lower and upper outputs of the output neurons, respectively.

$$\underline{y} = SNorm\left(\underline{Out},\overline{Out}\right) \tag{17}$$

$$\overline{y} = TNorm\left(\underline{Out},\overline{Out}\right) \tag{18}$$

where: we used the T-norms and S-norms.

Sum-Product:

$$TNorm\left(\underline{Out},\overline{Out}\right) = \underline{Out}\ .*\overline{Out} \tag{19}$$

$$SNorm\left(\underline{Out},\overline{Out}\right) = \underline{Out}+\overline{Out}-TNorm\left(\underline{Out},\overline{Out}\right) \tag{20}$$

Dombi: for $\gamma > 0$.

$$TNormD\left(\underline{Out},\overline{Out},\gamma\right) = \frac{1}{1+\left[\left(\underline{Out}^{-1}-1\right)^{\gamma}+\left(\overline{Out}^{-1}-1\right)^{\gamma}\right]^{1/\gamma}} \tag{21}$$

$$SNormD\left(\underline{Out},\overline{Out},\gamma\right)=\frac{1}{1+\left[\left(\underline{Out}^{-1}-1\right)^{-\gamma}+\left(\overline{Out}^{-1}-1\right)^{-\gamma}\right]^{-1/\gamma}} \tag{22}$$

Hamacher: for $\gamma>0$.

$$TNormH\left(\underline{Out},\overline{Out},\gamma\right)=\frac{\underline{Out}\;.*\;\overline{Out}}{\gamma+(1-\gamma)\left(\underline{Out}+\overline{Out}-\underline{Out}\;.*\;\overline{Out}\right)} \tag{23}$$

$$SNormH\left(\underline{Out},\overline{Out},\gamma\right)=\frac{\underline{Out}+\overline{Out}+(\gamma-2)\left(\underline{Out}\;.*\;\overline{Out}\right)}{1+(\gamma-1)\left(\underline{Out}\;.*\;\overline{Out}\right)} \tag{24}$$

Frank: for $s>0$.

$$TNormF\left(\underline{Out},\overline{Out},s\right)=log_s\left[\frac{1+\left(s^{\underline{Out}}-1\right)\left(s^{\overline{Out}}-1\right)}{s-1}\right] \tag{25}$$

$$SNormF\left(\underline{Out},\overline{Out},s\right)=1-log_s\left[\frac{1+\left(s^{1-\underline{Out}}-1\right)\left(s^{1-\overline{Out}}-1\right)}{s-1}\right] \tag{26}$$

Phase 4: The single output of the neural network is obtained:

$$y=\frac{\underline{y}+\overline{y}}{2} \tag{27}$$

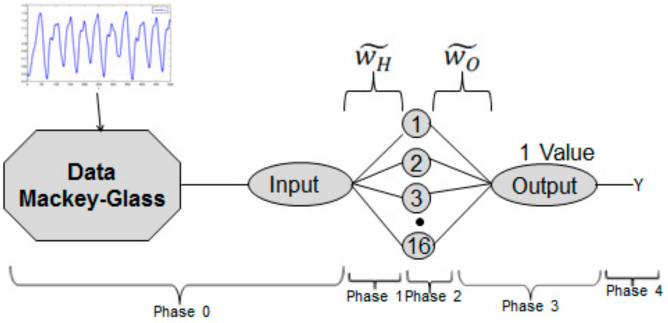

Figure 5. Illustration of the architecture of the Fuzzy Neural network.

3.3. Proposed Adjustment for Interval Type-2 Fuzzy Numbers with Backpropagation Learning

The backpropagation learning algorithm is performed by the adjustment of the interval type-2 fuzzy number weights, described as follows:

Stage 1: The Nguyen-Widrow algorithm is utilized to initialize the lower and upper values of the interval type-2 fuzzy numbers weights for the neural network.

Stage 2: The input pattern and the wanted output for the neural network is established.

Stage 3: The output of the neural network is calculated. In the first instance, the inputs for the network are introduced and the output of the network is obtained performing the calculations of the outputs from the input layer until the output layer.

Stage 4: Determine the error terms for the neurons of the layers. In the output layer, the calculation of lower ($\underline{\delta_{pk}^{O}}$) and upper ($\overline{\delta_{pk}^{O}}$) delta for each neuron "k" is performed with the follow equations:

$$\underline{\delta_{pk}^{O}}=\left(d_{pk}-y_{pk}\right)f_k^{O'}\left(\underline{y}\right) \tag{28}$$

$$\overline{\delta^{O}_{pk}} = \left(d_{pk} - y_{pk}\right) f_k^{O'}(\overline{y}) \tag{29}$$

In the hidden layer, the calculation of lower ($\underline{\delta^{h}_{pj}}$) and upper ($\overline{\delta^{h}_{pj}}$) delta for each neuron "*j*" is perform with the follow equations:

$$\underline{\delta^{\mathrm{h}}_{pj}} = f_j^{h'}(\underline{Net}) \sum_k \underline{\delta^{O}_{pk}}\,\underline{w_{kj}} \tag{30}$$

$$\overline{\delta^{\mathrm{h}}_{pj}} = f_j^{h'}(\overline{Net}) \sum_k \overline{\delta^{O}_{pk}}\,\overline{w_{kj}} \tag{31}$$

Stage 5: The utilization of a recursive algorithm allows the actualization of the interval type-2 fuzzy number weights, beginning from the output neurons and updating backwards until the neurons in the input layer. The adjustment is described as follows:

The calculation of the change of interval type-2 fuzzy number weights is achieved with the equations described as follows:

Calculations of the output neurons:

$$\underline{\Delta w_{kj}(t+1)} = \underline{\delta^{O}_{pk}}\,\underline{y_{pj}} \tag{32}$$

$$\overline{\Delta w_{kj}(t+1)} = \overline{\delta^{O}_{pk}}\,\overline{y_{pj}} \tag{33}$$

Calculations of the hidden neurons:

$$\underline{\Delta w_{ji}(t+1)} = \underline{\delta^{h}_{pj}}\,x_{pi} \tag{34}$$

$$\overline{\Delta w_{ji}(t+1)} = \overline{\delta^{h}_{pj}}\,x_{pi} \tag{35}$$

Stage 6: The method is recurrent until for each of the learned patterns the error terms are small enough.

$$E_p = \frac{1}{2}\sum_{k=1}^{M} \delta^2_{pk} \tag{36}$$

Alternatively, we have the option of working with fuzzy inputs and fuzzy targets. In this case, the modification of the proposed neural network must be in phase 2, multiplying by the lower input in Equation (2) and the upper input in Equation (3); besides, in phase 4, we can maintain the lower and upper final outputs of the output neuron because there is no need for performing the average.

The neural network used in this research consists of 16 neurons for the hidden layer, based on a study of the performance of the neural networks modifying the numbers of neurons in the hidden layer, starting with 5 neurons and increasing one by one, until reaching 120 neurons; we are presenting this study in the following section. To test the proposed method, experiments in the time series prediction are performed. A benchmark and chaotic time series such as the Mackey-Glass time series (for τ = 17) is used in this study.

Based on previous work, in the experiments, the backpropagation algorithm applying gradient descent and adaptive learning rate is utilized.

The neural networks manage interval type-2 fuzzy number weights in the hidden and output layer [53,54]. In the hidden and output layer of the network we used the backpropagation algorithm modified for working with interval type-2 fuzzy numbers to achieve new weights for the next epochs of the network [55–57].

4. Simulation Results

We achieved the experiments for the Mackey-Glass time series, and for this we used 1000 data points. In this case, 500 data points are considered for the training stage and 500 data points for the testing stage.

4.1. Neural Network with Interval Type-2 Fuzzy Numbers Weights (NNIT2FNW) for T-Norm and S-Norm of Sum-Product

We performed an experiment for determining manually the optimal number of neurons in the hidden layer of the fuzzy neural network with the interval type-2 fuzzy numbers; we increase the number of neurons by one unit at a time in the interval of 5 to 120 neurons. The acquired results from the experiments are presented in Table 1. The fuzzy neural network with 16 neurons in the hidden layer obtained the best result with 0.0149 for the best prediction error, and 0.0180 for the average error (MAE). This experiment was realized with the T-norm and S-norm of sum-product.

The mean absolute error (MAE) is considered in obtaining the results of the experiments. The average error was obtained by taking 30 experiments with the equal parameters and conditions for all experiments.

The parameters for the fuzzy neural network with interval type-2 fuzzy numbers are of 500 epochs and 0.0000001 of error for the training phase.

We observe from Table 1 that the fuzzy neural network with interval type-2 fuzzy numbers and T-norm and S-norm of sum-product with 16 neurons in the hidden layer (FNNIT2FNSp) shows better results than the others; so, based on this fact, in the following experiments we work with this architecture for the neural network.

In Figure 6, we are presenting the plot of the real data of the Mackey-Glass time series against the predicted data of the interval type-2 fuzzy neural network (FNNIT2FNSp) with 16 neurons in the hidden layer. In Figure 7, an illustration of the convergence curves in the training process is presented.

Table 1. Results for the fuzzy neural network with interval type-2 fuzzy numbers with T-norm of sum-product in time series prediction using Mackey-Glass time series.

No. Neurons	Best Prediction Error MAE	Average MAE
5	0.0187	0.0240
6	0.0197	0.0245
7	0.0188	0.0250
8	0.0172	0.0231
9	0.0198	0.0259
10	0.0170	0.0246
11	0.0190	0.0252
12	0.0192	0.0248
13	0.0198	0.0255
14	0.0191	0.0251
15	0.0185	0.0227
16	0.0149	0.0180
17	0.0180	0.0238
18	0.0202	0.0242
19	0.0205	0.0239
20	0.0164	0.0247
21	0.0201	0.0243
22	0.0189	0.0241
23	0.0178	0.0249
24	0.0195	0.0250
25	0.0195	0.0259
26	0.0189	0.0233
27	0.0195	0.0246

Table 1. *Cont.*

No. Neurons	Best Prediction Error MAE	Average MAE
28	0.0191	0.0248
29	0.0175	0.0245
30	0.0149	0.0233
31	0.0193	0.0245
32	0.0182	0.0259
33	0.0195	0.0252
34	0.0170	0.0243
35	0.0195	0.0241
36	0.0188	0.0251
37	0.0209	0.0248
38	0.0187	0.0243
39	0.0195	0.0254
40	0.0190	0.0246
41	0.0188	0.0263
42	0.0172	0.0233
43	0.0188	0.0249
44	0.0192	0.0237
45	0.0192	0.0247
46	0.0157	0.0247
47	0.0188	0.0252
48	0.0189	0.0246
49	0.0204	0.0247
50	0.0151	0.0246
51	0.0190	0.0250
52	0.0179	0.0239
53	0.0191	0.0242
54	0.0177	0.0240
55	0.0168	0.0240
56	0.0202	0.0251
57	0.0196	0.0255
58	0.0181	0.0250
59	0.0192	0.0248
60	0.0173	0.0239
61	0.0168	0.0236
62	0.0188	0.0239
63	0.0168	0.0240
64	0.0183	0.0238
65	0.0169	0.0252
66	0.0185	0.0250
67	0.0174	0.0253
68	0.0171	0.0230
69	0.0185	0.0244
70	0.0186	0.0248
71	0.0210	0.0251
72	0.0182	0.0249
73	0.0206	0.0247
74	0.0169	0.0249
75	0.0170	0.0240
76	0.0174	0.0233
77	0.0206	0.0245
78	0.0185	0.0244
79	0.0190	0.0247
80	0.0178	0.0246
81	0.0179	0.0247
82	0.0185	0.0243
83	0.0192	0.0254
84	0.0170	0.0237

Table 1. *Cont.*

No. Neurons	Best Prediction Error MAE	Average MAE
85	0.0178	0.0242
86	0.0186	0.0260
87	0.0197	0.0233
88	0.0197	0.0256
89	0.0178	0.0252
90	0.0191	0.0257
91	0.0183	0.0265
92	0.0193	0.0240
93	0.0199	0.0240
94	0.0166	0.0242
95	0.0206	0.0248
96	0.0181	0.0236
97	0.0191	0.0252
98	0.0199	0.0248
99	0.0173	0.0249
100	0.0181	0.0248
101	0.0168	0.0237
102	0.0173	0.0250
103	0.0198	0.0245
104	0.0191	0.0237
105	0.0205	0.0245
106	0.0197	0.0246
107	0.0179	0.0256
108	0.0185	0.0244
109	0.0189	0.0241
110	0.0164	0.0242
111	0.0190	0.0254
112	0.0198	0.0250
113	0.0173	0.0245
114	0.0203	0.0244
115	0.0168	0.0248
116	0.0170	0.0233
117	0.0199	0.0254
118	0.0188	0.0252
119	0.0196	0.0247
120	0.0189	0.0250

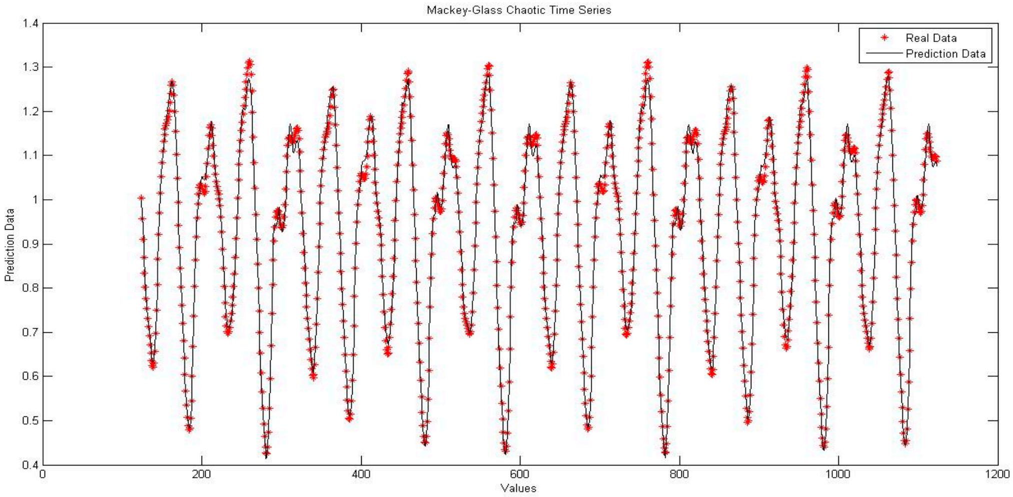

Figure 6. Illustration of the real data against the prediction data of the Mackey-Glass time series for the fuzzy neural network.

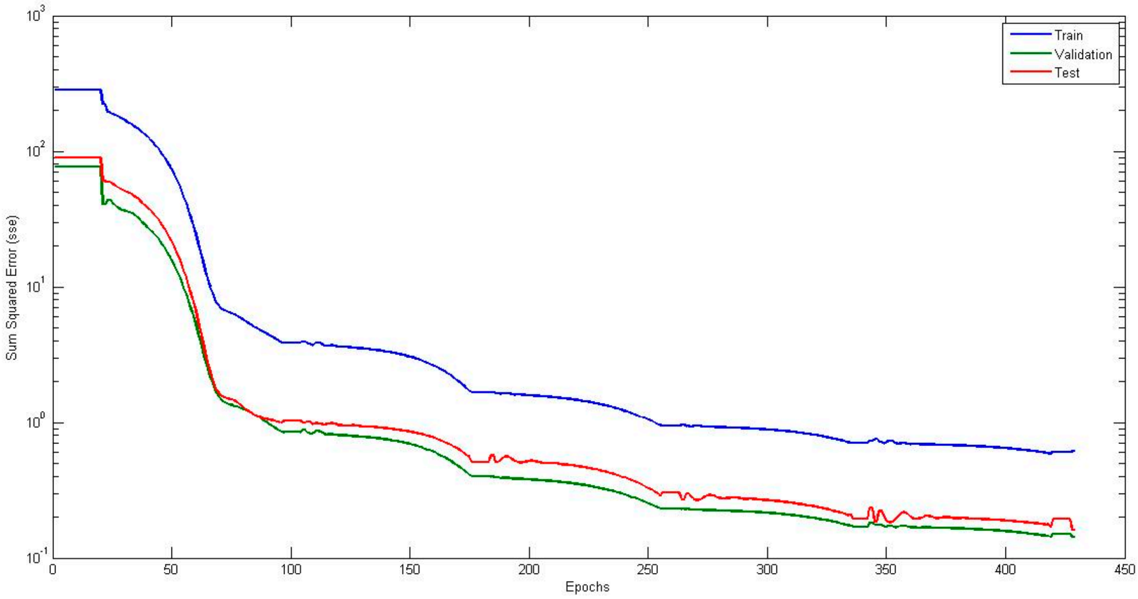

Figure 7. Illustration of the convergence curves in the training process for FNNIT2FNSp.

We performed the same experiment that we presented before with the T-norm and S-norm of Dombi, Hamacher and Frank.

4.2. NNIT2FNW for T-Norm and S-Norm of Dombi

The architecture for the fuzzy neural network with T-norm of Dombi (FNNIT2FND) has 4 neurons in the hidden layer and $\gamma = 0.8$, the best results with 0.0457 for the best result for the prediction error, and 0.0622 for the average error. We show some results for this architecture in Table 2, Figures 8 and 9.

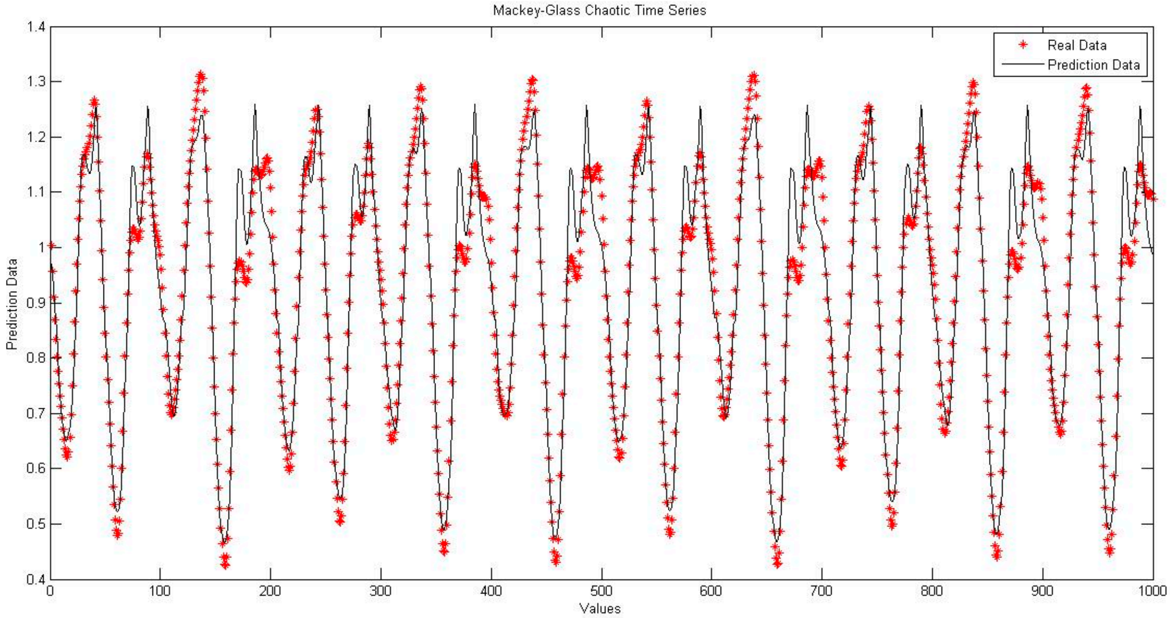

Figure 8. Illustration of the prediction data of the FNNIT2FND against the real data for the Mackey-Glass time series.

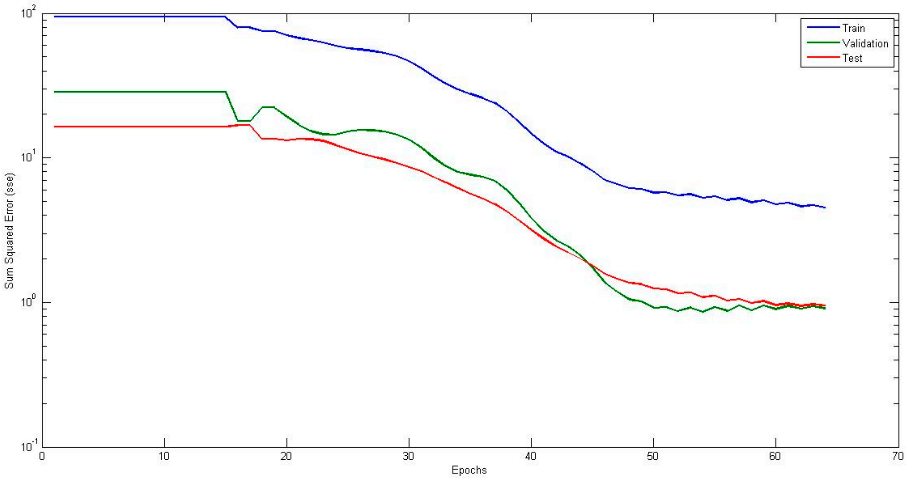

Figure 9. Illustration of the convergence curves in the training process for FNNIT2FND.

Table 2. Results for the fuzzy neural network with interval type-2 fuzzy numbers with T-norm of Dombi in time series prediction using Mackey-Glass time series.

Experiment	Prediction Error
1	0.0457
2	0.0466
3	0.0549
4	0.0581
5	0.0599
6	0.0636
7	0.0656
8	0.0671
9	0.0675
10	0.0694
Average	0.0622

4.3. NNIT2FNW for T-Norm and S-Norm of Hamacher

The architecture for the fuzzy neural network with the T-norm of Hamacher (FNNIT2FNH) has 39 neurons in the hidden layer and $\gamma = 1$, the best result with 0.0130 for the best prediction error, and 0.0164 for the average error. We show some results for this architecture in Table 3, Figures 10 and 11.

Table 3. Results for the fuzzy neural network with interval type-2 fuzzy numbers with T-norm of Hamacher in time series prediction using Mackey-Glass time series.

Experiment	Prediction Error
1	0.0130
2	0.0138
3	0.0149
4	0.0154
5	0.0163
6	0.0165
7	0.0170
8	0.0175
9	0.0177
10	0.0183
Average	0.0164

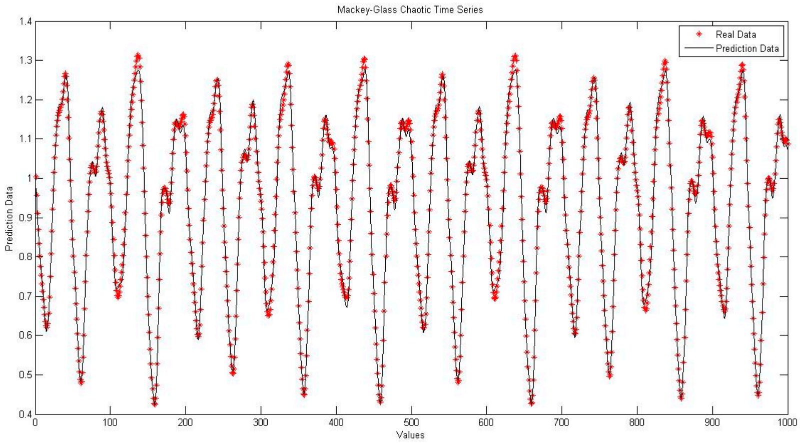

Figure 10. Illustration of the prediction data of the FNNIT2FNH against the real data for the Mackey-Glass time series.

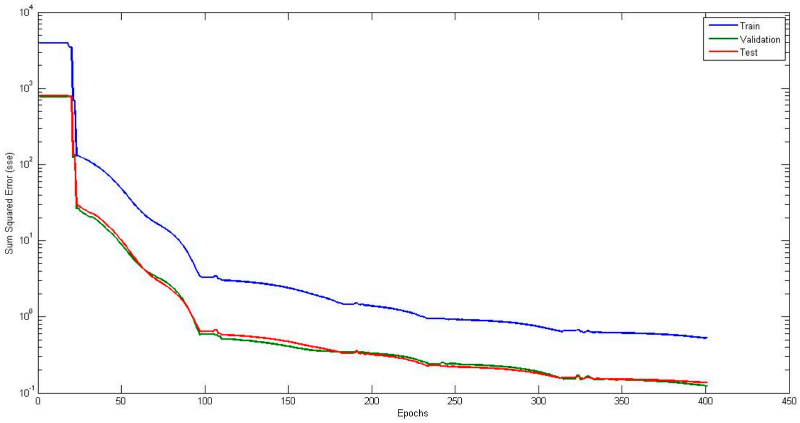

Figure 11. Illustration of the convergence curves in the training process for FNNIT2FNH.

4.4. NNIT2FNW for T-Norm and S-Norm of Frank

The architecture for the fuzzy neural network with T-norm of Frank (FNNIT2FNF) has 19 neurons in the hidden layer and $\gamma = 2.8$, the best results with 0.0117 for the best prediction error, and 0.0167 for the average error. We show some results for this architecture in Table 4, Figures 12 and 13.

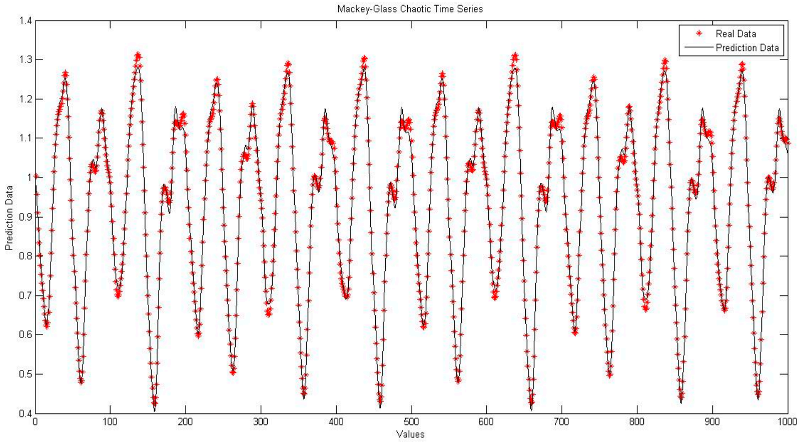

Figure 12. Illustration of the prediction data of the FNNIT2FNF against the real data for the Mackey-Glass time series.

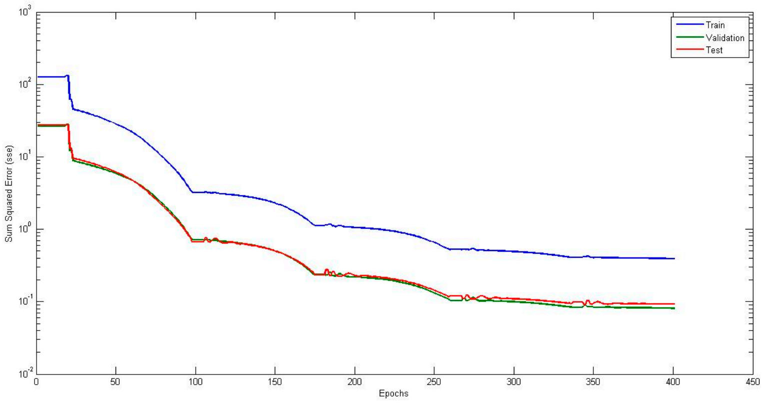

Figure 13. Illustration of the convergence curves in the training process for FNNIT2FNF.

Table 4. Results for the fuzzy neural network with interval type-2 fuzzy numbers with T-norm of Frank in time series prediction using Mackey-Glass time series.

Experiment	Prediction Error
1	0.0117
2	0.0140
3	0.0153
4	0.0156
5	0.0158
6	0.0163
7	0.0170
8	0.0175
9	0.0177
10	0.0179
Average	0.0167

4.5. Comparison of Traditional Neural Network Against NNIT2FNW for T-Norm and S-Norm

The acquired results in the experiments with the traditional neural network (TNN) are present on Table 5 and Figures 14 and 15, and the neural network parameters are obtained are based on empirical testing. The best result of the prediction errors is of 0.0169, and the average error is of 0.0203 (MAE). In Table 2, we present the comparison of these results against the results of the fuzzy neural network with interval type-2 fuzzy numbers for all T-norms (FNNIT2FNSp, FNNIT2FND, FNNIT2FNH, FNNIT2FNF).

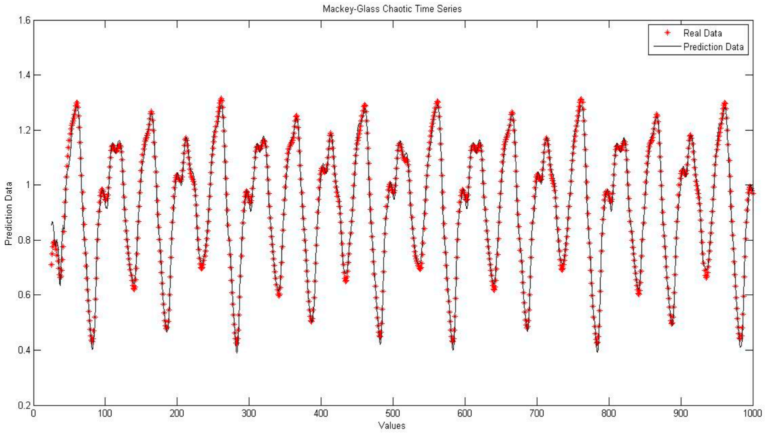

Figure 14. Illustration of the prediction data against for the traditional neural network the real data of the Mackey-Glass time series.

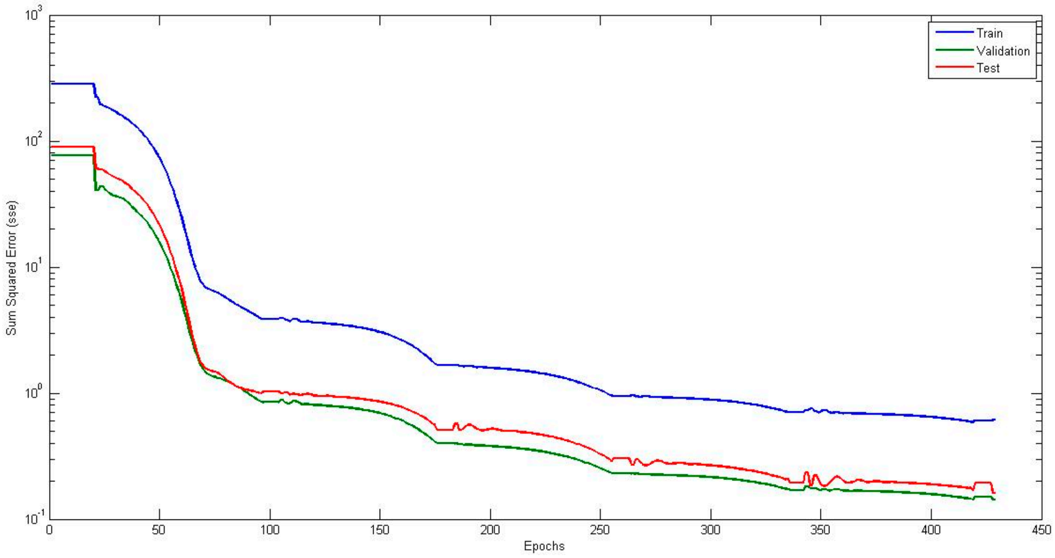

Figure 15. Illustration of the convergence curves in the training process for traditional neural network.

Table 5. Results for the traditional neural network (TNN) in the Mackey-Glass time series and the comparison against the FNNIT2FNSp, FNNIT2FND, FNNIT2FNH, and FNNIT2FNF.

	Best Prediction Error	**Average**
TNN	0.0169	0.0203
FNNIT2FNSp	0.0149	0.0180
FNNIT2FND	0.0457	0.0622
FNNIT2FNH	0.0130	0.0164
FNNIT2FNF	0.0117	0.0167

4.6. Comparison of the Proposed Methods for Mackey-Glass Data with Noise

We also implemented an experiment utilizing noisiness in the interval 0.1 to 1 in the test data to analyze the robustness of the traditional neural network (TNN) and the fuzzy neural network with interval type-2 fuzzy numbers for all T-norms (FNNIT2FNSp, FNNIT2FND, FNNIT2FNH, FNNIT2FNF). The obtained results for these experiments are presented in Table 6. We applied the noise by using the following equations:

$$DataNoise = Data + (NoiseLevel \times Noise) \tag{37}$$

$$Noise = 2 \times [rand(1, nData) - 0.5] \tag{38}$$

where: "Data" are the test data points of the Mackey-Glass Time series, "NoiseLevel" is the level of noise in the range (0.1–1), "rand" is a uniformly distributed function for random numbers used for obtained the values for the noise.

We observe in Table 6 that the better performance was accomplished with the fuzzy neural network with interval type-2 fuzzy numbers with all T-norms for almost all the levels of noise in the data test. The prediction error for the traditional neural network was increasing considerably with the higher noise levels as a difference to the fuzzy neural network, which maintains the prediction error below 0.21.

We observe in Figure 16 that when at the test data is applied noise, the fuzzy neural network with interval type-2 fuzzy numbers with the Hamacher and Frank T-norms achieve minor prediction errors compared with the traditional neural network in the different levels of noise. An important fact is that the fuzzy neural network with T-norm of Dombi presents better performance than the others in the test with noise but without noise has a higher prediction error.

Table 6. Results for the traditional neural network and fuzzy neural networks with all T-norms in the Mackey-Glass time series under different noise levels (n).

Noise Level	TNN	FNNIT2FNSp	FNNIT2FND	FNNIT2FNH	FNNIT2FNF
$n = 0$	0.0169	0.0149	0.0457	0.0130	0.0117
$n = 0.1$	0.0564	0.0617	0.0704	0.0556	0.0594
$n = 0.2$	0.1115	0.1135	0.0981	0.0960	0.0954
$n = 0.3$	0.1749	0.1275	0.1168	0.1171	0.1175
$n = 0.4$	0.2311	0.1554	0.1362	0.1360	0.1419
$n = 0.5$	0.3124	0.1661	0.1502	0.1536	0.1571
$n = 0.6$	0.3676	0.1897	0.1485	0.1576	0.1589
$n = 0.7$	0.4250	0.1866	0.1684	0.1770	0.1736
$n = 0.8$	0.4941	0.2018	0.1744	0.1811	0.1808
$n = 0.9$	0.5411	0.2077	0.1775	0.1887	0.1858
$n = 1$	0.5684	0.2075	0.1858	0.1920	0.1935

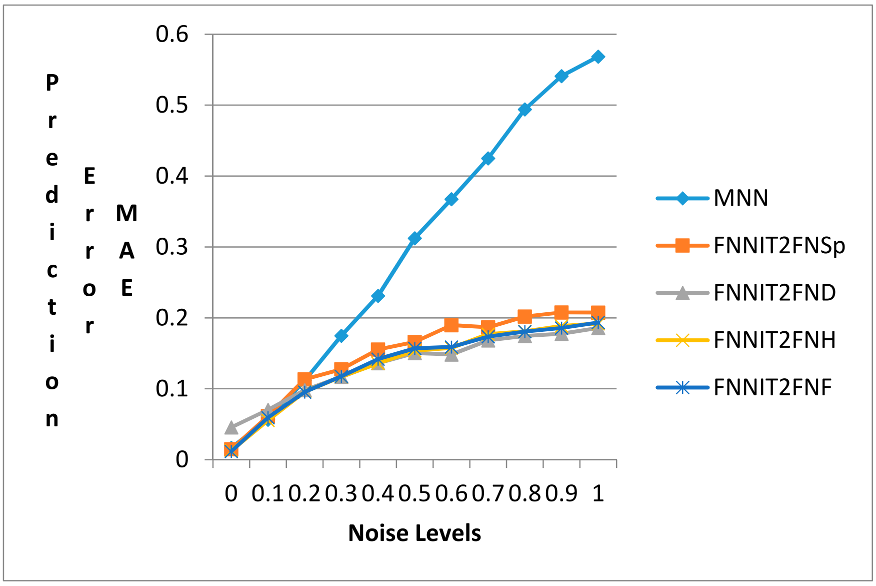

Figure 16. Illustration of the results of prediction error of the TNN against the results of FNNIT2FNSp, FNNIT2FND, FNNIT2FNH, and FNNIT2FNF for data with Gaussian noise of the Mackey-Glass time series for MAE.

A statistical test, the t-student test, was applied to perform a comparison of the performance of TNN against FNNIT2FNH, and FNNIT2FNF in the prediction error; we selected these two fuzzy neural networks because they presented a better performance than FNNIT2FNSp, and FNNIT2FND. In the statistical tests we consider 30 experiments and a 95% of reliability in the tests.

We present in Table 7 the parameters for the statistical test of the TNN and FNNIT2FNHmodel. We used a Hypothesis testing of H0: TNN = FNNIT2FNHand for the alternative hypothesis H1: TNN > FNNIT2FNH by the comparison of these models; H0: TNN = FNNIT2FNF and H1: TNN > FNNIT2FNF.

Table 7. Parameters used in the *t*-student statistical test for the TNN against FNNIT2FNH and FNNIT2FNF.

	TNN	FNNIT2FNH	FNNIT2FNF
No. Experiments	30	30	30
Mean Data	0.02028	0.01638	0.01665
Standard Deviation	0.00158	0.00133	0.00123
Standard error of the mean	0.00029	0.00024	0.00023

The results obtained with the statistical test for the prediction errors for TNN against FNNIT2FNH are of 0.003907 in the estimated mean difference, 0.003153 in the lower limit of the difference, a t value of 1037, p value of 0.0001 and 56 degrees of freedom.

The results obtained with the statistical test for prediction errors for TNN against FNNIT2FNFare of 0.003631 in the estimated mean difference, 0.002897 in the lower limit of the difference, a t value of 9.93, p value of 0.0001 and 54 degrees of freedom.

The results obtained with the statistical test for prediction errors for FNNIT2FNHagainst FNNIT2FNF are of -0.00027 in the estimated mean difference, -0.000940 in the lower limit of the difference, a t value of -0.84, p value of 0.407 and 57 degrees of freedom.

The results demonstrate that there exists significant statistical evidence to affirm that the FNNIT2FNH and FNNIT2FNHF are better than the TNN, and that the FNNIT2FNH is equal to the FNNIT2FNF.

5. Conclusions

Based on the experiments, we have reached the conclusion that the fuzzy neural network with interval type-2 fuzzy number weights with T-norm of Sum-product (FNNIT2FNSp), Hamacher (FNNIT2FNH) and Frank (FNNIT2FNF) achieved better results than the traditional neural network for the benchmark time series used in this work, Mackey-Glass. This affirmation is based on the prediction errors of 0.0169 for TNN, and 0.0149, 0.0130 and 0.0117 for FNNIT2FNSp, FNNIT2FNH and FNNIT2FNF, respectively; and the average errors obtained of 30 experiments are of 0.0203, and 0.0180, 0.0164 and 0.0167, respectively.

The fuzzy neural network with interval type-2 fuzzy number weights with the T-norm presented a better tolerance and behavior than the traditional neural network when Gaussian noise is applied at the testing data. This inference was reached by analyzing that the fuzzy neural network with interval type-2 fuzzy number weights with T-norms show only minor prediction errors compared to the traditional neural network at increasing the levels of noise. Besides, analyzing Tables 5 and 6, and Figure 10, from which the FNNIT2FNH and FNNIT2FNF show only minor prediction errors compared to the other paradigms in this work for the Mackey-Glass time series, and furthermore observing the results for the statistical tests performed to these paradigms.

The method proposed in this work, a fuzzy neural network with interval type-2 fuzzy number weights with T-norms, presents better performance, robustness and achieves lower results of prediction errors than the traditional neural network. Furthermore, the interval type-2 fuzzy number weights allow the neural network to have a minor susceptibility to increment the results of the prediction error when the real data is affected with noise.

Author Contributions: Patricia Melin proposed the idea of using fuzzy weights in the backpropagation algorithm. Juan R. Castro proposed the idea of using the T-norm and S-norm in the method. Patricia Melin, Juan R. Castro, Oscar Castillo and Fernando Gaxiola conceived and designed the experiments; Patricia Melin, Fevrier Valdez and Fernando Gaxiola performed the experiments; Juan R. Castro and Fevrier Valdez analyzed the data; Fernando Gaxiola and Oscar Castillo wrote the paper. In addition, Oscar Castillo review the correctness of the use of type 2 fuzzy logic.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dunyak, J.; Wunsch, D. Fuzzy Number Neural Networks. *Fuzzy Sets Syst.* **1999**, *108*, 49–58. [CrossRef]
2. Li, Z.; Kecman, V.; Ichikawa, A. Fuzzified Neural Network based on fuzzy number operations. *Fuzzy Sets Syst.* **2002**, *130*, 291–304. [CrossRef]
3. Beale, E.M.L. A Derivation of Conjugate Gradients. In *Numerical Methods for Non-Linear Optimization*; Lootsma, F.A., Ed.; Academic Press: London, UK, 1972; pp. 39–43.
4. Fletcher, R.; Reeves, C.M. Function Minimization by Conjugate Gradients. *Comput. J.* **1964**, *7*, 149–154. [CrossRef]

5. Moller, M.F. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. *Neural Netw.* **1993**, *6*, 525–533. [CrossRef]
6. Powell, M.J.D. Restart Procedures for the Conjugate Gradient Method. *Math. Program.* **1977**, *12*, 241–254. [CrossRef]
7. Gaxiola, F.; Melin, P.; Valdez, F.; Castillo, O. Interval Type-2 Fuzzy Weight Adjustment for Backpropagation Neural Networks with Application in Time Series Prediction. *Inf. Sci.* **2014**, *260*, 1–14. [CrossRef]
8. Gaxiola, F.; Melin, P.; Valdez, F.; Castillo, O. Generalized Type-2 Fuzzy Weight Adjustment for Backpropagation Neural Networks in Time Series Prediction. *Inf. Sci.* **2015**, *325*, 159–174. [CrossRef]
9. Casasent, D.; Natarajan, S. A Classifier Neural Net with Complex-Valued Weights and Square-Law Nonlinearities. *Neural Netw.* **1995**, *8*, 989–998. [CrossRef]
10. Draghici, S. On the Capabilities of Neural Networks using Limited Precision Weights. *Neural Netw.* **2002**, *15*, 395–414. [CrossRef]
11. Kamarthi, S.; Pittner, S. Accelerating Neural Network Training using Weight Extrapolations. *Neural Netw.* **1999**, *12*, 1285–1299. [CrossRef]
12. Dombi, J. A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness induced by fuzzy operators. *Fuzzy Sets Syst.* **1982**, *8*, 149–163. [CrossRef]
13. Weber, S. A general concept of fuzzy connectives, negations and implications based on *t*-norms and *t*-conorms. *Fuzzy Sets Syst.* **1983**, *11*, 115–134. [CrossRef]
14. Hamacher, H. Über logische verknupfungen unscharfer aussagen und deren zugehorige bewertungsfunktionen. In *Progress in Cybernetics and Systems Research, III*; Trappl, R., Klir, G.J., Ricciardi, L., Eds.; Hemisphere: New York, NY, USA, 1975; pp. 276–288. (In German)
15. Frank, M.J. On the simultaneous associativity of F(x, y) and x + y − F(x, y). *Aequ. Math.* **1979**, *19*, 194–226. [CrossRef]
16. Neville, R.S.; Eldridge, S. Transformations of Sigma–Pi Nets: Obtaining Reflected Functions by Reflecting Weight Matrices. *Neural Netw.* **2002**, *15*, 375–393. [CrossRef]
17. Yam, J.; Chow, T. A Weight Initialization Method for Improving Training Speed in Feedforward Neural Network. *Neurocomputing* **2000**, *30*, 219–232. [CrossRef]
18. 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.* **2006**, *34*, 603–618.
19. De Wilde, O. The Magnitude of the Diagonal Elements in Neural Networks. *Neural Netw.* **1997**, *10*, 499–504. [CrossRef]
20. Salazar, P.A.; Melin, P.; Castillo, O. A New Biometric Recognition Technique Based on Hand Geometry and Voice Using Neural Networks and Fuzzy Logic. *Soft Comput. Hybrid Intell. Syst.* **2008**, *154*, 171–186.
21. Cazorla, M.; Escolano, F. Two Bayesian Methods for Junction Detection. *IEEE Trans. Image Process.* **2003**, *12*, 317–327. [CrossRef] [PubMed]
22. Hagan, M.T.; Demuth, H.B.; Beale, M.H. *Neural Network Design*; PWS Publishing: Boston, MA, USA, 1996; p. 736.
23. Phansalkar, V.V.; Sastry, P.S. Analysis of the Back-Propagation Algorithm with Momentum. *IEEE Trans. Neural Netw.* **1994**, *5*, 505–506. [CrossRef] [PubMed]
24. Fard, S.; Zainuddin, Z. Interval Type-2 Fuzzy Neural Networks Version of the Stone–Weierstrass Theorem. *Neurocomputing* **2011**, *74*, 2336–2343. [CrossRef]
25. Asady, B. Trapezoidal Approximation of a Fuzzy Number Preserving the Expected Interval and Including the Core. *Am. J. Oper. Res.* **2013**, *3*, 299–306. [CrossRef]
26. Coroianu, L.; Stefanini, L. General Approximation of Fuzzy Numbers by F-Transform. *Fuzzy Sets Syst.* **2016**, *288*, 46–74. [CrossRef]
27. Yang, D.; Li, Z.; Liu, Y.; Zhang, H.; Wu, W. A Modified Learning Algorithm for Interval Perceptrons with Interval Weights. *Neural Process Lett.* **2015**, *42*, 381–396. [CrossRef]
28. Requena, I.; Blanco, A.; Delgado, M.; Verdegay, J. A Decision Personal Index of Fuzzy Numbers based on Neural Networks. *Fuzzy Sets Syst.* **1995**, *73*, 185–199. [CrossRef]
29. Kuo, R.J.; Chen, J.A. A Decision Support System for Order Selection in Electronic Commerce based on Fuzzy Neural Network Supported by Real-Coded Genetic Algorithm. *Expert. Syst. Appl.* **2004**, *26*, 141–154. [CrossRef]
30. Molinari, F. A New Criterion of Choice between Generalized Triangular Fuzzy Numbers. *Fuzzy Sets Syst.* **2016**, *296*, 51–69. [CrossRef]

31. Chai, Y.; Xhang, D. A Representation of Fuzzy Numbers. *Fuzzy Sets Syst.* **2016**, *295*, 1–18. [CrossRef]
32. Figueroa-García, J.C.; Chalco-Cano, Y.; Roman-Flores, H. Distance Measures for Interval Type-2 Fuzzy Numbers. *Discret. Appl. Math.* **2015**, *197*, 93–102. [CrossRef]
33. Ishibuchi, H.; Nii, M. Numerical Analysis of the Learning of Fuzzified Neural Networks from Fuzzy If–Then Rules. *Fuzzy Sets Syst.* **1998**, *120*, 281–307. [CrossRef]
34. Karnik, N.N.; Mendel, J. Operations on type-2 fuzzy sets. *Fuzzy Sets Syst.* **2001**, *122*, 327–348. [CrossRef]
35. Raj, P.A.; Kumar, D.N. Ranking Alternatives with Fuzzy Weights using Maximizing Set and Minimizing Set. *Fuzzy Sets Syst.* **1999**, *105*, 365–375.
36. Chu, T.C.; Tsao, T.C. Ranking Fuzzy Numbers with an Area between the Centroid Point and Original Point. *Comput. Math. Appl.* **2002**, *43*, 111–117. [CrossRef]
37. Ishibuchi, H.; Morioka, K.; Tanaka, H. A Fuzzy Neural Network with Trapezoid Fuzzy Weights. In Proceedings of the Fuzzy Systems, IEEE World Congress on Computational Intelligence, Orlando, FL, USA, 26–29 June 1994; Volume 1, pp. 228–233.
38. Ishibuchi, H.; Tanaka, H.; Okada, H. Fuzzy Neural Networks with Fuzzy Weights and Fuzzy Biases. In Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, USA, 28 March–1 April 1993; Volume 3, pp. 1650–1655.
39. Feuring, T. Learning in Fuzzy Neural Networks. In Proceedings of the IEEE International Conference on Neural Networks, Washington, DC, USA, 3–6 June 1996; Volume 2, pp. 1061–1066.
40. Castro, J.; Castillo, O.; Melin, P.; Rodríguez-Díaz, A. A Hybrid Learning Algorithm for a Class of Interval Type-2 Fuzzy Neural Networks. *Inform. Sci.* **2009**, *179*, 2175–2193. [CrossRef]
41. 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. *Soft Comput. Intell. Control Mob. Robot.* **2011**, *318*, 269–285.
42. Abiyev, R. A Type-2 Fuzzy Wavelet Neural Network for Time Series Prediction. *Lect. Notes Comput. Sci.* **2010**, *6098*, 518–527.
43. Karnik, N.; Mendel, J. Applications of Type-2 Fuzzy Logic Systems to Forecasting of Time-Series. *Inform. Sci.* **1999**, *120*, 89–111. [CrossRef]
44. Pulido, M.; Melin, P.; Castillo, O. Genetic Optimization of Ensemble Neural Networks for Complex Time Series Prediction. In Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN), San Jose, CA, USA, 31 July–5 August 2011; pp. 202–206.
45. Pedrycz, W. *Granular Computing: Analysis and Design of Intelligent Systems*; CRC Press/Francis Taylor: Boca Raton, FL, USA, 2013.
46. Tung, S.W.; Quek, C.; Guan, C. eT2FIS: An Evolving Type-2 Neural Fuzzy Inference System. *Inform. Sci.* **2013**, *220*, 124–148. [CrossRef]
47. Zarandi, M.H.F.; Torshizi, A.D.; Turksen, I.B.; Rezaee, B. A new indirect approach to the type-2 fuzzy systems modeling and design. *Inform. Sci.* **2013**, *232*, 346–365. [CrossRef]
48. Zhai, D.; Mendel, J. Uncertainty Measures for General Type-2 Fuzzy Sets. *Inform. Sci.* **2011**, *181*, 503–518. [CrossRef]
49. Biglarbegian, M.; Melek, W.; Mendel, J. On the robustness of Type-1 and Interval Type-2 fuzzy logic systems in modeling. *Inform. Sci.* **2011**, *181*, 1325–1347. [CrossRef]
50. Jang, J.S.R.; Sun, C.T.; Mizutani, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*; Prentice Hall: Englewood Cliffs, NJ, USA, 1997; p. 614.
51. Chen, S.; Wang, C. Fuzzy decision making systems based on interval type-2 fuzzy sets. *Inform. Sci.* **2013**, *242*, 1–21. [CrossRef]
52. Nguyen, D.; Widrow, B. Improving the Learning Speed of 2-Layer Neural Networks by choosing Initial Values of the Adaptive Weights. In Proceedings of the International Joint Conference on Neural Networks, San Diego, CA, USA, 17–21 June 1990; Volume 3, pp. 21–26.
53. Montiel, O.; Castillo, O.; Melin, P.; Sepúlveda, R. The evolutionary learning rule for system identification. *Appl. Soft Comput.* **2003**, *3*, 343–352. [CrossRef]
54. 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*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 41, pp. 45–52.

55. Castillo, O.; Melin, P. A review on the design and optimization of interval type-2 fuzzy controllers. *Appl. Soft Comput.* **2012**, *12*, 1267–1278. [CrossRef]
56. Hagras, H. Type-2 Fuzzy Logic Controllers: A Way Forward for Fuzzy Systems in Real World Environments. *IEEE World Congr. Comput. Intell.* **2008**, *5050*, 181–200.
57. Melin, P. *Modular Neural Networks and Type-2 Fuzzy Systems for Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2012; 204 p.