

Article

# Neutrosophic Compound Orthogonal Neural Network and Its Applications in Neutrosophic Function Approximation

Jun Ye \*  and Wenhua Cui

Department of Electrical and Information Engineering, Shaoxing University, 508 Huancheng West Road, Shaoxing 312000, China; wenhuacui@usx.edu.cn

\* Correspondence: yehjun@aliyun.com; Tel.: +86-575-88327323

Received: 16 January 2019; Accepted: 28 January 2019; Published: 29 January 2019



**Abstract:** Neural networks are powerful universal approximation tools. They have been utilized for functions/data approximation, classification, pattern recognition, as well as their various applications. Uncertain or interval values result from the incompleteness of measurements, human observation and estimations in the real world. Thus, a neutrosophic number (NsN) can represent both certain and uncertain information in an indeterminate setting and imply a changeable interval depending on its indeterminate ranges. In NsN settings, however, existing interval neural networks cannot deal with uncertain problems with NsNs. Therefore, this original study proposes a neutrosophic compound orthogonal neural network (NCONN) for the first time, containing the NsN weight values, NsN input and output, and hidden layer neutrosophic neuron functions, to approximate neutrosophic functions/NsN data. In the proposed NCONN model, single input and single output neurons are the transmission notes of NsN data and hidden layer neutrosophic neurons are constructed by the compound functions of both the Chebyshev neutrosophic orthogonal polynomial and the neutrosophic sigmoid function. In addition, illustrative and actual examples are provided to verify the effectiveness and learning performance of the proposed NCONN model for approximating neutrosophic nonlinear functions and NsN data. The contribution of this study is that the proposed NCONN can handle the approximation problems of neutrosophic nonlinear functions and NsN data. However, the main advantage is that the proposed NCONN implies a simple learning algorithm, higher speed learning convergence, and higher learning accuracy in indeterminate/NsN environments.

**Keywords:** Neutrosophic compound orthogonal neural network; Neutrosophic number; Neutrosophic function; Function approximation

## 1. Introduction

Neural networks are powerful universal approximation tools. They have been utilized for data modeling, function approximation, classification analysis, pattern recognition, as well as their various applications. Uncertain or interval values result from the incompleteness of measurements, human observation and estimations in the real world. Hence, Baker and Patil [1] proposed an interval neural network (INN), which used interval weights rather than interval data input to approximate an interval function. Then, Hu et al. [2] presented an INN with interval weights, where the network is modeled like the problem of the solution equations, implying its complexity in the solution process. Rossi and Conan-Guez [3] introduced a multilayer perceptron neural network on interval data for the classification analysis of interval data. For processing the interval neural network, Patiño-Escarcina [4] presented an INN, where one of its input, output, and weight sets is interval values and the output set

is a binary one, therefore its outputs are binaries too for classifiers. Recently, Lu et al. [5] introduced a neural network-based interval matcher corresponding to linguistic IF-THEN constructions, which is an interval pattern matcher to identify patterns with interval elements using the neural network, which can handle interval inputs values and interval output values based on a traditional neural network and is only suitable for interval pattern matching. Kowalski and Kulczycki [6] presented the interval probabilistic neural network (IPNN) for the classification of interval data, where the IPNN structure is based on Specht's probabilistic network [7].

In indeterminate environments, neutrosophic theory [8–10] has been used for various applications [11–14]. Since a neutrosophic number (NsN) [8–10] can represent both certain and uncertain information in indeterminate settings and contain a changeable interval depending on its indeterminate ranges, NsNs have been widely applied to decision making [15–17], fault diagnoses [18,19], linear and nonlinear optimization problems [20–23], expression and analysis of the rock joint roughness coefficient (JRC) [24–27]. However, there is no study on neutrosophic neural networks with NsNs in existing literature, while existing INNs also cannot deal with uncertain problems with NsNs. Therefore, this original study proposes a neutrosophic compound orthogonal neural network (NCONN) for the first time, which contains the NsN weight values, NsN input and output neurons and hidden layer neutrosophic neurons, to approximate neutrosophic functions and NsN data. In the proposed NCONN model, single input and single output data are NsNs (changeable interval numbers) and hidden layer neutrosophic neuron functions are composed of the Chebyshev neutrosophic orthogonal polynomial and neutrosophic sigmoid function. In addition, illustrative and actual examples are provided to verify the effectiveness and performance of the proposed NCONN model in approximating neutrosophic nonlinear functions and NsN data. The contribution of this study is that the proposed NCONN can handle the approximating and modelling problems of neutrosophic functions and NsN data for the first time. The main advantage is that the proposed NCONN implies a simple learning algorithm, higher speed learning convergence, and higher learning accuracy in indeterminate/NsN environments.

This study was formed as the following framework. The second section introduces the basic concepts and operations of NsNs. The third section proposes a NCONN structure and its learning algorithm. Then, two illustrative examples about neutrosophic nonlinear function approximations and an actual example (a real case) about the approximation problem of rock JRC NsNs are presented in the fourth section and the fifth section, respectively, to verify the effectiveness and performance of the proposed NCONN in approximating neutrosophic nonlinear functions and NsN data under indeterminate/NsN environments. The last section contains conclusions and future work.

## 2. Basic Concepts and Operations of NsNs

In an uncertain setting, Smarandache [8–10] introduced the NsN concept represented by the mathematical form  $N = c + uI$  for  $a, b \in R$  (all real numbers) and  $I$  (indeterminacy), in which the certain part  $c$  with its uncertain part  $uI$  for  $I \in [I^-, I^+]$  are combined. Hence, it can depict and express the certain and/or uncertain information in indeterminate problems.

Provided there is the NsN  $N = 5 + 3I$ , it depicts that the certain value is five and its uncertain value is  $3I$ . Then, some interval range of the indeterminacy  $I \in [I^-, I^+]$  is possibly specified in actual applications to satisfy some applied requirement. For instance, the indeterminacy  $I$  is specified as such a possible interval  $I \in [0, 2]$ . Thus, it is equivalent to  $N = [5, 11]$ . If  $I \in [1, 3]$ , then there is  $N = [8, 14]$ . It is obvious that it is a changeable interval depending on the specified indeterminate range of  $I \in [I^-, I^+]$ , which is also denoted by  $N = [c + uI^-, c + uI^+]$ .

In some special cases, a NsN  $N = c + uI$  for  $N \in U$  ( $U$  is all NsNs) may be represented as either a certain number  $N = c$  for  $uI = 0$  (the best case) or an uncertain number  $N = uI$  for  $c = 0$  (the worst case).

Provided that there are two NsNs  $N_1 = c_1 + u_1I$  and  $N_2 = c_2 + u_2I$  for  $N_1, N_2 \in U$  and  $I \in [I^-, I^+]$ , then their operational laws are introduced as follows [21]:

$$N_1 + N_2 = c_1 + c_2 + (u_1 + u_2)I = [c_1 + c_2 + u_1I^-, c_1 + c_2 + u_1I^+ + u_2I^+] \tag{1}$$

$$N_1 - N_2 = c_1 - c_2 + (u_1 - u_2)I = [c_1 - c_2 + u_1I^- - u_2I^-, c_1 - c_2 + u_1I^+ - u_2I^+] \tag{2}$$

$$N_1 \times N_2 = c_1c_2 + (c_1u_2 + c_2u_1)I + u_1u_2I^2 = \left[ \begin{array}{l} \min \left( \begin{array}{l} (c_1 + u_1I^-)(c_2 + u_2I^-), (c_1 + u_1I^-)(c_2 + u_2I^+), \\ (c_1 + u_1I^+)(c_2 + u_2I^-), (c_1 + u_1I^+)(c_2 + u_2I^+) \end{array} \right), \\ \max \left( \begin{array}{l} (c_1 + u_1I^-)(c_2 + u_2I^-), (c_1 + u_1I^-)(c_2 + u_2I^+), \\ (c_1 + u_1I^+)(c_2 + u_2I^-), (c_1 + u_1I^+)(c_2 + u_2I^+) \end{array} \right) \end{array} \right] \tag{3}$$

$$\frac{N_1}{N_2} = \frac{c_1 + u_1I}{c_2 + u_2I} = \frac{[c_1 + u_1I^-, c_1 + u_1I^+]}{[c_2 + u_2I^-, c_2 + u_2I^+]} = \left[ \begin{array}{l} \min \left( \frac{c_1 + u_1I^-}{c_2 + u_2I^+}, \frac{c_1 + u_1I^-}{c_2 + u_2I^-}, \frac{c_1 + u_1I^+}{c_2 + u_2I^+}, \frac{c_1 + u_1I^+}{c_2 + u_2I^-} \right), \\ \max \left( \frac{c_1 + u_1I^-}{c_2 + u_2I^+}, \frac{c_1 + u_1I^-}{c_2 + u_2I^-}, \frac{c_1 + u_1I^+}{c_2 + u_2I^+}, \frac{c_1 + u_1I^+}{c_2 + u_2I^-} \right) \end{array} \right] \tag{4}$$

Regarding an uncertain function containing NsNs, Ye [21,22] defined a neutrosophic function in  $n$  variables (unknowns) as  $y(x, I): U^n \rightarrow U$  for  $x = [x_1, x_2, \dots, x_n]^T \in U^n$  and  $I \in [I^-, I^+]$ , which is then a neutrosophic nonlinear or linear function.

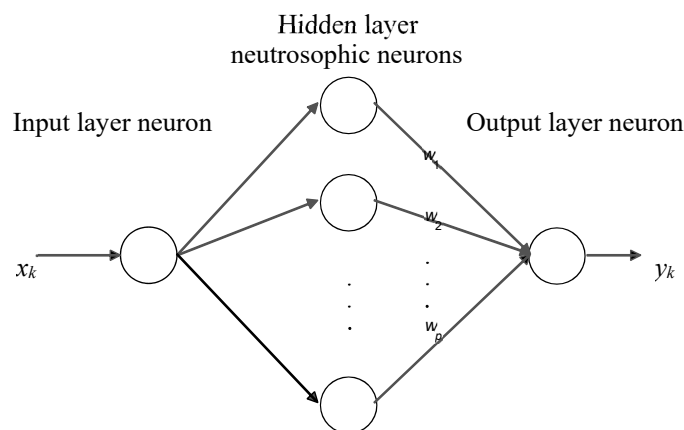
For example,  $y_1(x, I) = N_1x \cos(x) = (c_1 + u_1I)x \cos(x)$  for  $x \in U$  and  $I \in [I^-, I^+]$  is a neutrosophic nonlinear function, while  $y_2(x, I) = N_1x_1 + N_2x_2 + N_3 = (c_1 + u_1I)x_1 + (c_2 + u_2I)x_2 + (c_3 + u_3I)$  for  $x = [x_1, x_2]^T \in U^2$  and  $I \in [I^-, I^+]$  is a neutrosophic linear function.

Generally, the values of  $x$  and  $y(x)$  are NsNs (usually, but not always).

### 3. NCONN with NsNs

This section proposes a NCONN structure and its learning algorithm based on the NsN concept for the first time.

A three-layer feedforward NCONN structure with a single input, single output, and hidden layer neutrosophic neurons are indicated in Figure 1. In Figure 1, the weight values between the input layer neuron and the hidden layer neutrosophic neurons are equal to the constant value 1 and the NsN weight values between the hidden layer neutrosophic neurons and the output layer neuron are  $w_j$  ( $j = 1, 2, \dots, p$ );  $x_k$  ( $k = 1, 2, \dots, n$ ) is the  $k$ th NsN input signal;  $y_k$  is the  $k$ th NsN output signal; and  $p$  is the number of the hidden layer neutrosophic neurons.



**Figure 1.** A three-layer feedforward neutrosophic compound orthogonal neural network (NCONN structure).

In the learning process, when each NsN input signal is given by  $x_k = c_k + u_k I = [c_k + u_k I^-, c_k + u_k I^+]$  ( $k = 1, 2, \dots, n$ ) for  $I \in [I^-, I^+]$ , the actual output value is given as:

$$y_k = \sum_{j=1}^p w_j \tilde{q}_j, k = 1, 2, \dots, n \quad (5)$$

where the neutrosophic neuron functions of the hidden layer  $\tilde{q}_j$  for  $j = 1, 2, \dots, p$  are the Chebyshev compound neutrosophic orthogonal polynomial:  $\tilde{q}_1 = [1, 1]$ ,  $\tilde{q}_2 = \tilde{X}$ ,  $\tilde{q}_j = 2\tilde{X} \cdot \tilde{q}_{j-1} - \tilde{q}_{j-2}$ , and  $\tilde{X}$  is specified as the following unipolar neutrosophic sigmoid function (the neutrosophic S-function):

$$\tilde{X} = \frac{1}{1 + e^{-\alpha x_k}} \quad (6)$$

The neutrosophic S-function can transform NsN into the interval (0, 1) and the different scalar parameters of  $\alpha$  can change the slant degree of the neutrosophic S-function curve.

Then, the square interval of output errors between the desired output  $y_k^d = c_k^d + u_k^d I$  and the actual output  $y_k = c_k + u_k I$  for  $I \in [I^-, I^+]$  is given as follows:

$$\tilde{E}_k^2 = [(c_k^d + u_k^d I^- - c_k - u_k I^-)^2, (c_k^d + u_k^d I^+ - c_k - u_k I^+)^2] \quad (7)$$

Whereas, the learning performance index of the proposed NCONN is specified as the following requirement:

$$\tilde{E} = \frac{1}{2} \sum_{k=1}^n \tilde{E}_k^2 \quad (8)$$

The NCONN weight values can be adjusted by the following formula:

$$\tilde{W}_k(l+1) = \tilde{W}_k(l) + \lambda \tilde{E}_k \tilde{Q}, k = 1, 2, \dots, n \quad (9)$$

where  $\tilde{W}_k(l) = [\tilde{w}_1(l), \tilde{w}_2(l), \dots, \tilde{w}_q(l)]^T$  and  $\tilde{Q}_k(l) = [\tilde{q}_1(l), \tilde{q}_2(l), \dots, \tilde{q}_p(l)]^T$  is the NsN weight vector and the function vector of the hidden layer neutrosophic neurons,  $\lambda$  is the learning rate of the NCONN to determine the convergence velocity for  $\lambda \in (0, 1)$ , and  $l$  is the  $l$ th iteration learning of the NCONN.

Thus, this NCONN learning algorithm can be described below:

Step 1: Give  $\tilde{W}_k(0)$  by small random values,

Step 2: Input a NsN and calculate the actual output of a NCONN based on Equations (5) and (6),

Step 3: Calculate the output error by using Equations (7) and (8),

Step 4: Adjust weight values by using Equation (9),

Step 5: Input the next NsN and return to Step 2.

In the NCONN learning process, the learning termination condition depends on the requirement of the specified learning error or iteration number.

Since NsN can be considered as a changeable interval depending to its indeterminacy  $I \in [I^-, I^+]$ , the learning algorithm of NCONN permits changeable interval operations, which are different from existing neural network algorithms and show its advantage of approximating neutrosophic nonlinear functions/NsN data in an uncertain/NsN setting.

Generally, the more the hidden layer neutrosophic neurons are, the higher the approximation accuracy of the proposed NCONN is. Then, the number of the hidden layer neutrosophic neurons determined in actual applications will depend on the accuracy requirements of actual approximation models.

#### 4. NsN Nonlinear Function Approximation Applied by the Proposed NCONN

To prove the effectiveness of approximating any neutrosophic nonlinear function based on the proposed NCONN model, we present two illustrative examples in this section.

**Example 1.** Supposing there is a neutrosophic nonlinear function:

$$y_1(x, I) = 1 + 0.3I + (0.5 + 0.2I)x \cos(\pi x) \text{ for } I \in [0, 1].$$

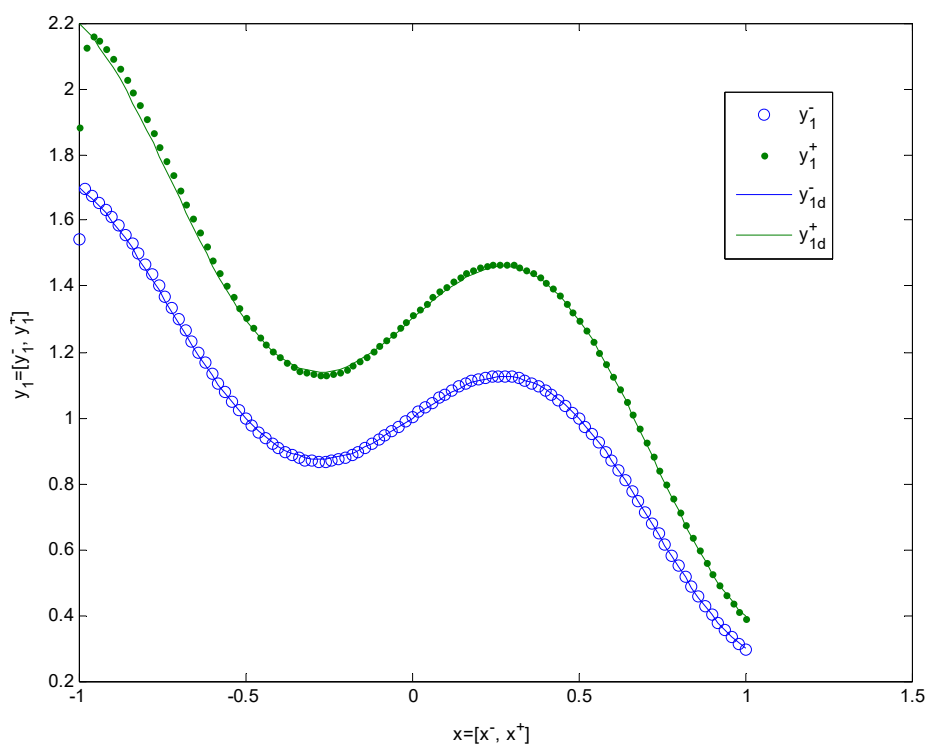
For  $x \in [-1+0.02I, 1+0.023I]$  and  $I \in [0, 1]$ , the proposed NCONN needs to approximate the above neutrosophic nonlinear function.

To prove the approximation ability of the proposed NCONN, we give the proposed NCONN structure with eight hidden layer neutrosophic neurons ( $p = 8$ ) and learning parameters, which are indicated in Table 1.

**Table 1.** The NCONN structure and learning parameters.

NCONN Structure	$\alpha$	$\lambda$	The Number of the Specified Learning Iteration	$\tilde{E}$
$1 \times 8 \times 1$	2.5	0.25	20	[3.2941, 8.5088]

Then, the desired output  $y_{1d} = [y_{1d}^-, y_{1d}^+]$  and actual output  $y_1 = [y_1^-, y_1^+]$  of the proposed NCONN are shown in Figure 2. Obviously, the desired output curves and the actual output curves were very close to each other, to demonstrate the better approximation accuracy in the neutrosophic nonlinear function approximation of the proposed NCONN. Hence, the proposed NCONN indicated the better approximation performance regarding the neutrosophic nonlinear function.



**Figure 2.** The desired output  $y_{1d} = [y_{1d}^-, y_{1d}^+]$  and actual output  $y_1 = [y_1^-, y_1^+]$  of the proposed NCONN.

**Example 2.** Considering a neutrosophic nonlinear function:

$$y_2(x, I) = (0.6 + 0.3I) \sin(\pi x) + (0.3 + 0.15I) \sin(3\pi x) + (0.1 + 0.05I) \sin(5\pi x) \text{ for } I \in [0, 1].$$

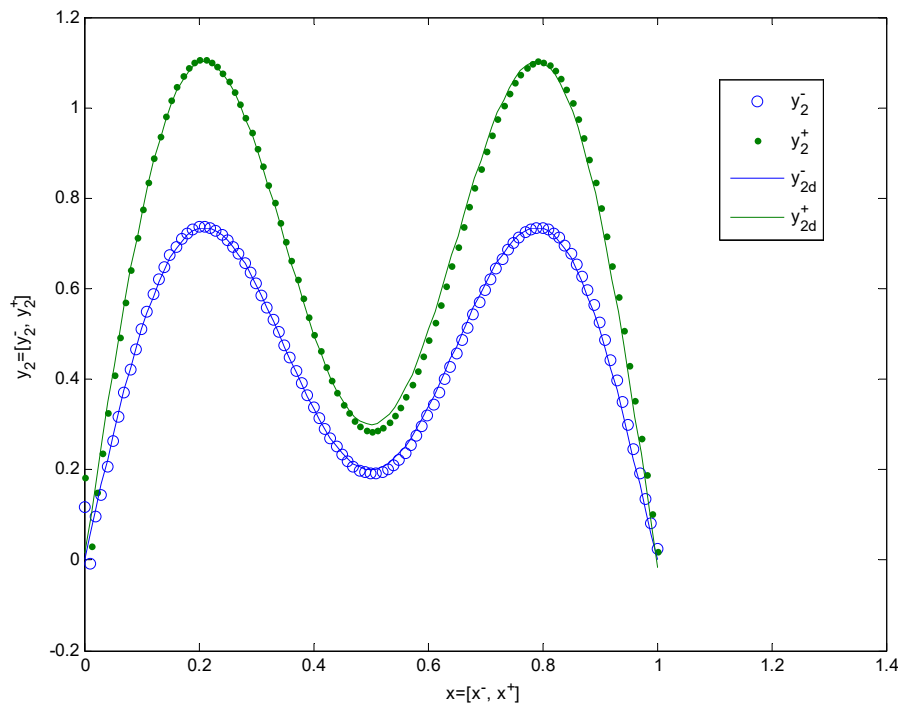
For  $x \in [0 + 0.002I, 1 + 0.002I]$  and  $I \in [0, 1]$ , the proposed NCONN needs to approximate the above neutrosophic nonlinear function.

To prove the approximation ability of the proposed NCONN model, we also give the NCONN structure with eight hidden layer neutrosophic neurons ( $p = 8$ ) and learning parameters, which are indicated in Table 2.

**Table 2.** The NCONN structure and learning parameters.

NCONN Structure	$\alpha$	$\lambda$	The Number of the Specified Learning Iteration	$\tilde{E}$
$1 \times 8 \times 1$	8	0.3	20	[0.5525, 1.1261]

Thus, the desired output  $y_{2d} = [y_{2d}^-, y_{2d}^+]$  and actual output  $y_2 = [y_2^-, y_2^+]$  of the proposed NCONN are indicated in Figure 3. It was obvious that the desired output curves and the actual output curves were also very close, so as to demonstrate the better approximating accuracy and performance in the neutrosophic nonlinear function approximation of the proposed NCONN.

**Figure 3.** The desired output  $y_{2d} = [y_{2d}^-, y_{2d}^+]$  and actual output  $y_2 = [y_2^-, y_2^+]$  of the proposed NCONN.

Corresponding to the learning results obtained from the above two illustrative examples, we could see that the proposed NCONN showed faster learning velocity and a higher learning accuracy, which indicated a better approximation performance regarding the neutrosophic nonlinear functions.

### 5. Actual Example on the Approximation of the JRC NsNs Based on the Proposed NCONN

In rock mechanics, the JRC of rock joints implies uncertainty in different sampling lengths and directions of rock joints. Therefore, JRC uncertainty may make the shear strength of joints uncertain because of the corresponding relationship between JRC and the shear strength, which results in the difficulty of making assessments of side stability [25–27]. However, the lengths of the testing samples can affect JRC values, which indicates their scale effect. To establish a relationship between the sampling lengths  $L$  and the JRC values in an uncertain/NsN setting, existing literature [25–27] used the uncertain/neutrosophic statistic method and fitting functions to establish some related model of  $L$  and the JRC. Since the proposed NCONN is able to approximate NsN data, the proposed NCONN could be applied to the relative approximation model between the sampling length  $L$  and the NsN data of the JRC by an actual example (a real case) in this section, to show its effectiveness.

According to the testing samples of the specified area in Shaoxing city, China and data analysis, we found a relationship between the sampling length  $L$  and the NsN data of JRC, which are shown in Table 3.

**Table 3.** NsN data of rock joint roughness coefficient (JRC) regarding different sampling lengths for  $I \in [0, 1]$ .

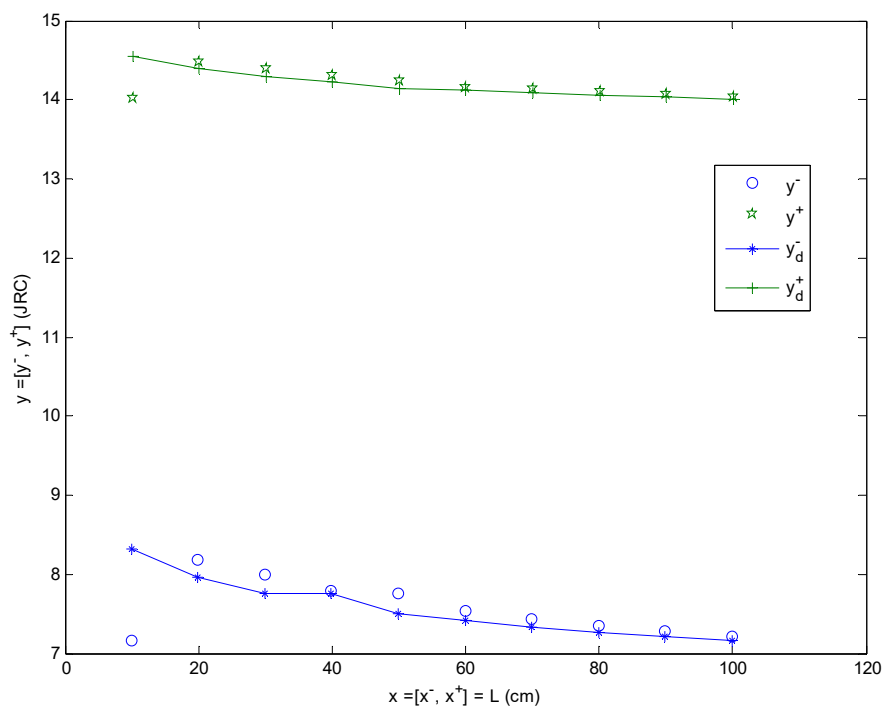
Sample Length $L$ (cm)	$x_k$	JRC	$y_k$
$9.8 + 0.4I$	[9.8, 10.2]	$8.321 + 6.231I$	[8.321, 14.552]
$19.8 + 0.4I$	[19.8, 20.2]	$7.970 + 6.419I$	[7.970, 14.389]
$29.8 + 0.4I$	[29.8, 30.2]	$7.765 + 6.529I$	[7.765, 14.294]
$39.8 + 0.4I$	[39.8, 40.2]	$7.762 + 6.464I$	[7.762, 14.226]
$49.8 + 0.4I$	[49.8, 50.2]	$7.507 + 6.64I$	[7.507, 14.147]
$59.8 + 0.4I$	[59.8, 60.2]	$7.417 + 6.714I$	[7.417, 14.131]
$69.8 + 0.4I$	[69.8, 70.2]	$7.337 + 6.758I$	[7.337, 14.095]
$79.8 + 0.4I$	[79.8, 80.2]	$7.269 + 6.794I$	[7.269, 14.063]
$89.8 + 0.4I$	[89.8, 90.2]	$7.210 + 6.826I$	[7.210, 14.036]
$99.8 + 0.4I$	[99.8, 100.2]	$7.156 + 6.855I$	[7.156, 14.011]

To establish the approximation model of the proposed NCONN regarding the actual example, we took the NCONN structure with eight hidden layer neutrosophic neurons ( $p = 8$ ) and indicated the learning parameters in Table 4.

**Table 4.** The NCONN structure and learning parameters regarding the actual example.

NCONN Structure	$\alpha$	$\lambda$	The Number of the Specified Learning Iteration	$\tilde{E}$
$1 \times 8 \times 1$	8	0.11	5	[3.2715, 22.3275]

From Figure 4, we can see that the proposed NCONN could approximate the JRC NsN data regarding different sampling lengths  $L$  and showed a higher speed convergence and higher approximating accuracy in its learning process for the actual example. Obviously, the proposed NCONN could find the approximating model between different sampling lengths  $L$  and JRC NsN data, while existing neural networks cannot do them in the uncertain/NsN setting.



**Figure 4.** The proposed NCONN approximation results of the JRC NsN data regarding different sampling lengths  $L$ .

## 6. Conclusions

In a NsN setting, this original study presented a NCONN to approximate neutrosophic functions/NsN data for the first time. It is a three-layer feedforward neutrosophic network structure composed of a single input, a single output, and hidden layer neutrosophic neurons, where the single input and single output information are NsNs and hidden layer neutrosophic neuron functions are composed of both the Chebyshev neutrosophic orthogonal polynomial and the neutrosophic sigmoid function. Illustrative and actual examples were provided to verify the effectiveness and rationality of the proposed NCONN model for approximating neutrosophic nonlinear functions and establishing the approximation model of NsN data. Therefore, the contribution of this study is that the proposed NCONN could handle the approximating and modeling problems of uncertain/interval/neutrosophic functions and NsN data. Here, the main advantage is that the proposed NCONN implies a simpler learning algorithm, higher speed learning convergence, and higher learning accuracy in indeterminate/NsN environments.

In the future work, we shall propose further NCONN with multi-inputs and multi-outputs and apply them to the modeling and approximating problems of neutrosophic functions and NsN data, the clustering analysis of NsNs, medical diagnosis problems, and possible applications for decision-making and control in robotics [14,28] in an indeterminate/NsN setting.

**Author Contributions:** J.Y. proposed the NCONN model and learning algorithm; W.H.C. provided the simulation analysis of illustrative and actual examples; J.Y. and W.H.C. wrote the paper together.

**Funding:** This paper was supported by the National Natural Science Foundation of China (Nos. 71471172, 61703280).

**Conflicts of Interest:** The authors declare that we have no conflict of interest regarding the publication of this paper.

## References

1. Baker, M.R.; Patil, R.B. Universal approximation theorem for interval neural networks. *Reliab. Comput.* **1998**, *4*, 235–239. [\[CrossRef\]](#)
2. Beheshti, M.; Berrached, A.; Korvin, A.D.; Hu, C.; Sirisaengtaksin, O. On interval weighted three-layer neural networks. In Proceedings of the 31st Annual Simulation Symposium, Boston, MA, USA, 5–9 April 1998; pp. 188–195.
3. Rossi, F.; Conan-Guez, B. Multi-layer perceptron on interval data. Classification, clustering, and data analysis. In *Studies in Classification, Data Analysis, and Knowledge Organization*; Springer: Berlin, Germany, 2002; pp. 427–434.
4. Patiño-Escarcina, R.E.; Callejas Bedregal, B.R.; Lyra, A. Interval computing in neural networks: One layer interval neural networks. In *Intelligent Information Technology. Lecture Notes in Computer Science*; Das, G., Gulati, V.P., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3356, pp. 68–75.
5. Lu, J.; Xue, S.; Zhang, X.; Han, Y. A neural network-based interval pattern matcher. *Information* **2015**, *6*, 388–398. [\[CrossRef\]](#)
6. Piotr, A.; Kowalski, P.K. Interval probabilistic neural network. *Neural Comput. Appl.* **2017**, *28*, 817–834.
7. Specht, D.F. Probabilistic neural networks. *Neural Netw.* **1990**, *3*, 109–118. [\[CrossRef\]](#)
8. Smarandache, F. *Neutrosophy: Neutrosophic Probability, Set, and Logic*; American Research Press: Rehoboth, DE, USA, 1998.
9. Smarandache, F. *Introduction to Neutrosophic Measure, Neutrosophic Integral, and Neutrosophic Probability*; Sitech & Education Publisher: Craiova, Columbus, 2013.
10. Smarandache, F. *Introduction to Neutrosophic Statistics*; Sitech & Education Publishing: Columbus, OH, USA, 2014.
11. Ye, J. A multicriteria decision-making method using aggregation operators for simplified neutrosophic sets. *J. Intell. Fuzzy Syst.* **2014**, *26*, 2459–2466.



12. Broumi, S.; Bakali, A.; Talea, M.; Smarandache, F.; Uluçay, V.; Sahin, M.; Dey, A.; Dhar, M.; Tan, R.P.; Bahnasse, A.; et al. Neutrosophic sets: An overview. In *New Trends in Neutrosophic Theory and Applications*; Smarandache, F., Pramanik, S., Eds.; Pons Publishing House: Brussels, Belgium, 2018; Volume II, pp. 403–434.
13. Broumi, S.; Bakali, A.; Talea, M.; Smarandache, F.; Vladareanu, L. Computation of shortest path problem in a network with SV-trapezoidal neutrosophic numbers. In *Proceedings of the 2016 International Conference on Advanced Mechatronic Systems*, Melbourne, Australia, 30 November–3 December 2016; pp. 417–422.
14. Gal, I.A.; Bucur, D.; Vladareanu, L. DSMT decision-making algorithms for finding grasping configurations of robot dexterous hands. *Symmetry* **2018**, *10*, 198. [[CrossRef](#)]
15. Ye, J. Multiple-attribute group decision-making method under a neutrosophic number environment. *J. Intell. Syst.* **2016**, *25*, 377–386. [[CrossRef](#)]
16. Chen, J.Q.; Ye, J. A projection model of neutrosophic numbers for multiple attribute decision making of clay-brick selection. *Neutrosophic Sets Syst.* **2016**, *12*, 139–142.
17. Ye, J. Bidirectional projection method for multiple attribute group decision making with neutrosophic numbers. *Neural Comput. Appl.* **2017**, *28*, 1021–1029. [[CrossRef](#)]
18. Kong, L.W.; Wu, Y.F.; Ye, J. Misfire fault diagnosis method of gasoline engines using the cosine similarity measure of neutrosophic numbers. *Neutrosophic Sets Syst.* **2015**, *8*, 43–46.
19. Ye, J. Fault diagnoses of steam turbine using the exponential similarity measure of neutrosophic numbers. *J. Intell. Fuzzy Syst.* **2016**, *30*, 1927–1934. [[CrossRef](#)]
20. Jiang, W.Z.; Ye, J. Optimal design of truss structures using a neutrosophic number optimization model under an indeterminate environment. *Neutrosophic Sets Syst.* **2016**, *14*, 93–975.
21. Ye, J. Neutrosophic number linear programming method and its application under neutrosophic number environments. *Soft Comput.* **2018**, *22*, 4639–4646. [[CrossRef](#)]
22. Ye, J. An improved neutrosophic number optimization method for optimal design of truss structures. *New Math. Nat. Comput.* **2018**, *14*, 295–305. [[CrossRef](#)]
23. Ye, J.; Cui, W.H.; Lu, Z.K. Neutrosophic number nonlinear programming problems and their general solution methods under neutrosophic number environments. *Axioms* **2018**, *7*, 13. [[CrossRef](#)]
24. Ye, J.; Chen, J.Q.; Yong, R.; Du, S.G. Expression and analysis of joint roughness coefficient using neutrosophic number functions. *Information* **2017**, *8*, 69. [[CrossRef](#)]
25. Chen, J.Q.; Ye, J.; Du, S.G. Scale effect and anisotropy analyzed for neutrosophic numbers of rock joint roughness coefficient based on neutrosophic statistics. *Symmetry* **2017**, *9*, 208. [[CrossRef](#)]
26. Chen, J.Q.; Ye, J.; Du, S.G.; Yong, R. Expressions of rock joint roughness coefficient using neutrosophic interval statistical numbers. *Symmetry* **2017**, *9*, 123. [[CrossRef](#)]
27. Ye, J.; Yong, R.; Liang, Q.F.; Huang, M.; Du, S.G. Neutrosophic functions of the joint roughness coefficient (JRC) and the shear strength: A case study from the pyroclastic rock mass in Shaoxing City, China. *Math. Probl. Eng.* **2016**, 4825709. [[CrossRef](#)]
28. Vlădăreanu, V.; Dumitrache, I.; Vlădăreanu, L.; Sacală, I.S.; Tonț, G.; Moisescu, M.A. Versatile intelligent portable robot control platform based on cyber physical systems principles. *Stud. Inform. Control* **2015**, *24*, 409–418. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).