

The New Complex-Valued Wavelet Neural Network

Sufang Li, Mingyan Jiang

School of Information Science and Engineering, Shandong University, Jinan,
250100, P.R. China

e-mail: sufangli@mail.sdu.edu.cn, corresponding author:jiangmingyan@sdu.edu.cn

Abstract

A new complex-valued wavelet neural network is proposed in this paper, by introducing a modified complex-valued back propagation algorithm, in which a new error function is to be minimized by the algorithm. The improvement performance is further confirmed by the simulation results, which show that the modified algorithm is simpler than the conventional algorithm, and has better convergence, better stability and faster running speed.

Keywords: *complex-valued wavelet neural network (CVWNN); complex-valued back propagation (CVBP) algorithm; XOR*

1. Introduction

Wavelet analysis theory is considered to be a breakthrough in the Fourier analysis and has been applied in many research areas. Wavelet transform can effectively extract the local information of the signal by scaling and translation to analyze the signal [1]. Combining the wavelets with the artificial neural network (ANN), the wavelet neural network (WNN) has been developed [2]-[4]. The ANN has many important properties such as learning, generalization, and parallel computation, although it need a large number of neurons in hidden layer and cannot converge quickly. WNN has inherited the good properties of the ANN. Moreover it can converge quickly and give high precision with reduced network size because of the time–frequency localization properties of wavelets [5].

There are two types of WNN structure. The first WNN is pre-wavelet neural network, and the architecture is shown in Figure 1. The network firstly process the input signal using the orthogonal wavelet matrix, then the network put into learning and discriminating. The second WNN is called embedded wavelet neural network, the architecture of which is shown in Figure 2, in which the wavelet transform algorithm is integrated into the feed-forward neural network. In embedded wavelet neural network, wavelet functions are used in the hidden layer of the network as activation functions instead of local functions in time such as Gaussian and sigmoid functions.

Li et al. [6] proposed complex-valued wavelet artificial network (CVWNN) using Haar wavelet as the hidden layer activation function (AF) in complex-valued artificial neural network (CVANN). The complex-valued wavelet neural network is the complex version of the real-valued wavelet neural network, which has complex inputs, outputs, connection weights, dilation and translation parameters, but the nonlinearity of the hidden nodes remains a real-valued function (real-valued wavelet function). CVWNN has expanded its applications in fields dealing with complex numbers such as biomedical image processing [7], telecommunications [8],[9], carotid arterial Doppler ultrasound signals classifying [5], speech recognition [10], signal and image processing with the Fourier transformation [11].

The core algorithm of the CVWNN is complex-valued BP algorithm, which is based on gradient descent often suffers from a local minima problem and has slow convergence. Many methods [12],[13] have been proposed to improve the performance, such as the convergence and the local stability. These methods usually applied adaptive activation function and added a term to the conventional error function to speed up the convergence and prevent the learning from sticking into the local minima. Unfortunately, the local minimal problem and some errors are closely related to the neuron saturation of the activation function. When the actual output approaches the extreme value, the neurons in the output layer and the hidden layer are sensitive to input signals and the propagation chain will almost be blocked.

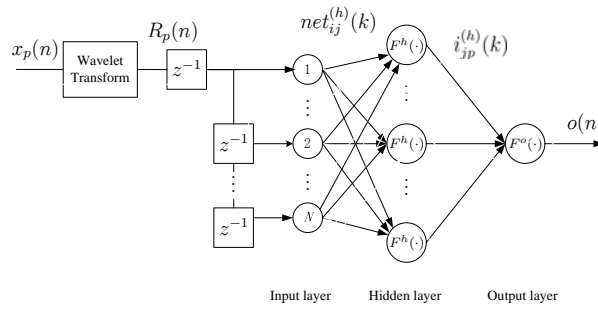


Figure 1. The architecture of the pre-wavelet neural network

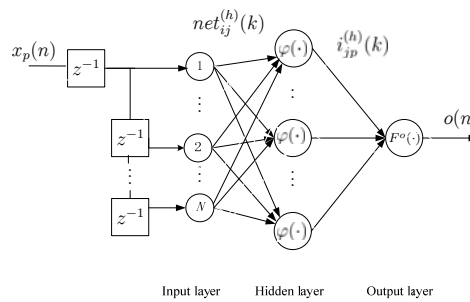


Figure 2. The architecture of the embedded wavelet neural network

In this paper, a modified CVWNN is proposed to resolve the XOR problems. The new CVBP algorithm and the wavelet function activation function in the hidden layer can improve the performance of the network, avoiding the effectness of the saturation of the activation function, having excellent functional approximation and generalization abilities.

2. WNN

Wavelet is a new powerful tool for representing nonlinearity. A function $f(x)$ can be represented by the superposition of daughters $\psi_{a,b}(x)$ of a mother wavelet $\psi(x)$, where $\psi_{a,b}(x)$ can be expressed as

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \tag{1}$$

$a \in R_+$ and $b \in R$ are, respectively, called dilation and translation parameters. The continuous wavelet transform of $f(x)$ is defined as

$$\psi_{a,b}(x) = \int_{-\infty}^{\infty} f(x)\overline{\psi_{a,b}(x)}dx \tag{2}$$

And the function $f(x)$ can be reconstructed by the inverse wavelet transform

$$f(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(a,b)\psi_{a,b}(x)\frac{dadb}{a^2} \tag{3}$$

The continuous wavelet transform and its inverse transform are not directly implementable on digital computers. When the inverse wavelet transform (3) is discretized, $f(x)$ has the following approximate wavelet-based representation form:

$$f(x) \approx \sum_{k=1}^K w_k \psi \left(\frac{x - b_k}{a_k} \right) \quad (4)$$

where w_k , b_k and a_k are weight coefficients, translations and dilations for each daughter wavelet. This approximation can be expressed as the neural network of Figure 2, which contains wavelet nonlinearities in the artificial neurons rather than the standard sigmoidal nonlinearities.

3 Traditional complex-valued BP neural network

3.1 The forward propagation process

In this paper, the classical three-layer network [14] is introduced, the architecture of which is shown in Figure 2. The input vector is $\mathbf{X}_p = (x_{p1}, x_{p2}, \dots, x_{pN})^T$, which is applied to the input layer of the network. Then the input units distribute the values to the hidden layer units. The net input to the j th hidden unit is

$$\begin{aligned} net_{pj}^h &= net_{pj,R}^h + jnet_{pj,I}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h \\ &= \sum_{i=1}^N (w_{ji,R}^h x_{pi,R} - w_{ji,I}^h x_{pi,I}) + \theta_{j,R}^h + j \left[\sum_{i=1}^N (w_{ji,R}^h x_{pi,I} + w_{ji,I}^h x_{pi,R}) + \theta_{j,I}^h \right] \end{aligned} \quad (4)$$

where w_{ji} is the complex-valued connection weight from the i th input unit to the j th unit, and θ_j^h is the bias terms in the j th unit. The “ h ” superscript is the quantities on the hidden layer. The output of the hidden neuron is

$$i_{pj} = i_{pj,R} + j i_{pj,I} = F_j^h(net_{pj}^h) = f_j^h(net_{pj,R}^h) + j f_j^h(net_{pj,I}^h) \quad (5)$$

where “ R ” superscript and “ I ” superscript are the quantities on the real part and the imaginary part of the values respectively. “ F ” is the complex-valued activation function, which is

$$F^h(x) = f^h(x) + j f^h(x) \quad (6)$$

where $f^h(x)$ refers to the formula (4).

The net input and the output of the j th output unit are

$$\begin{aligned} net_{pk}^o &= net_{pk,R}^o + jnet_{pk,I}^o = \sum_{j=1}^L w_{kj}^o i_{pj} + q_k^o \\ &= \sum_{j=1}^L (w_{kj,R}^o i_{pj,R} - w_{kj,I}^o i_{pj,I}) + q_{k,R}^o + j \left[\sum_{j=1}^L (w_{kj,R}^o i_{pj,I} + w_{kj,I}^o i_{pj,R}) + q_{k,I}^o \right] \end{aligned} \quad (7)$$

$$O_{pk} = O_{pk,R} + j O_{pk,I} = F_k^o(net_{pk}^o) = f_k^o(net_{pk,R}^o) + j f_k^o(net_{pk,I}^o) \quad (8)$$

where the “ o ” superscript is the quantities on the output layer.

3.2 The backward propagation process

The backward propagation refers to the error signal backward propagation. $\delta_{pk} = D_{pk} - O_{pk}$ is defined as the error at a single output unit, where “ p ” refers to the p th training vector, and “ k ” refers to the k th unit. The error is minimized by the complex algorithm. Since the size of complex-valued number cannot be compared, the error energy function is as follows, which is the sum of the squares of the errors of all output units.

$$\begin{aligned} E_p &= \frac{1}{2} \sum_{k=1}^M \delta_{pk} \delta_{pk}^* \\ &= \frac{1}{2} \sum_{k=1}^M [D_{pk} - f_k^o(\text{net}_{pk,R}^o) - j f_k^o(\text{net}_{pk,I}^o)] [D_{pk}^* - f_k^o(\text{net}_{pk,R}^o) + j f_k^o(\text{net}_{pk,I}^o)] \\ &= \frac{1}{2} \sum_{k=1}^M [(D_{pk,R} - f_k^o(\text{net}_{pk,R}^o))^2 + (D_{pk,I} - f_k^o(\text{net}_{pk,I}^o))^2] \end{aligned} \quad (9)$$

where the “*” is means complex conjugation and M is the node number of the output layer. In order to determine the weight changing direction, it is necessary to calculate the negative of the gradient of the E_p according to the real and imaginary part of the coefficients. The weights can be written as

$$w_{kj}^o(t) = w_{kj,R}^o(t) + j w_{kj,I}^o(t) \quad (10)$$

Firstly, the adaption rule of the output layer is considered. According to the steepest descent rule, the weights can be updated as

$$w_{kj,R}^o(t+1) = w_{kj,R}^o(t) - \eta \frac{\partial E_p}{\partial w_{kj,R}^o(t)} \quad (11)$$

$$w_{kj,I}^o(t+1) = w_{kj,I}^o(t) - \eta \frac{\partial E_p}{\partial w_{kj,I}^o(t)} \quad (12)$$

where η is learning step, which is a positive constant. Combining (11) and (12), we can have

$$w_{kj}^o(t+1) = w_{kj}^o(t) - \eta \left(\frac{\partial E_p}{\partial w_{kj,R}^o(t)} + j \frac{\partial E_p}{\partial w_{kj,I}^o(t)} \right) \quad (13)$$

Finally, according to the error function formula, we can get

$$\frac{\partial E_p}{\partial w_{kj,R}^o} + j \frac{\partial E_p}{\partial w_{kj,I}^o} = -\frac{1}{2} [(D_{pk,R} - O_{pk,R}) f_k^{\prime o}(\text{net}_{pk,R}^o) - j (D_{pk,I} - O_{pk,I}) f_k^{\prime o}(\text{net}_{pk,I}^o)] i_{pj}^* \quad (14)$$

The weight update equations can be summarized by defining a quantity

$$\delta_{pk}^o = f_k^{\prime o}(\text{net}_{pk,R}^o) \text{Re}(D_{pk} - O_{pk}) + j f_k^{\prime o}(\text{net}_{pk,I}^o) \text{Im}(D_{pk} - O_{pk}) \quad (15)$$

When the active function (AF) is sigmoid function, such as

$$f^o(x) = \frac{1}{1 + e^{-x}} \quad (16)$$

which is one of the most widely used AF for the artificial neural network. The first-order differential of the AF is

$$\begin{aligned}
f_k^o(x) &= \frac{0 - e^{-x}(-1)}{(1 + e^{-x})^2} = \frac{1}{(1 + e^{-x})^2} (1 + e^{-x} - 1) \\
&= \frac{1}{1 + e^{-x}} \left(1 - \frac{1}{1 + e^{-x}}\right) \\
&= f(x)(1 - f(x))
\end{aligned} \tag{17}$$

According to (3.2.7), we can get

$$\begin{aligned}
\delta_{pk}^o &= f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))\text{Re}(D_{pk} - O_{pk}) \\
&\quad + j f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))\text{Im}(D_{pk} - O_{pk})
\end{aligned} \tag{18}$$

Whatever form of the output layer activation function f_k^o , the weight-update equation in the output layer can be written

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta \delta_{pk}^o w_{pj}^* \tag{19}$$

Similarly we can get the adaption rule of the hidden layer,

$$w_{ji,I}^h(t+1) = w_{ji,I}^h(t) + \eta (\delta_{pj,R}^h x_{pi,R} - \delta_{pj,I}^h x_{pi,I}) \tag{20}$$

where $\delta_{pj}^h = f_j^h(\text{net}_{pj,R}^h)\text{Re}(\sum_{k=1}^M \delta_{pk}^o w_{kj}^{o*}) + j f_j^h(\text{net}_{pj,I}^h)\text{Im}(\sum_{k=1}^M \delta_{pk}^o w_{kj}^{o*})$.

In this section, it is explained the results of research and at the same time is given the comprehensive discussion. Results can be presented in figures, graphs, tables and others that make the reader understand easily [2],[5]. The discussion can be made in several sub-chapters.

4 New complex-valued wavelet neural networks

The complex-valued WNN algorithm as described above has the following questions. When the actual value O_{pk} approaches the extreme value, i.e., 0 or 1, the factors $f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))$ and $f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))$ makes the error signal very small. This means that an output unit can be maximally wrong without producing a strong error signal with which the synaptic weight should be significantly adjusted. The search for a minimum in the error will be retarded. M.Jiang et al. have introduced a modified error function for the complex-valued BP neural network to overcome the above shortcomings and avoid the delay of the convergence of the network. Instead of minimizing the squares of the differences between the actual outputs and desired outputs [16], in which the error function to be minimized is as follows,

$$\begin{aligned}
E_p &= \frac{1}{2} \sum_{k=1}^M [D_{pk,R} \ln O_{pk,R} + (1 - D_{pk,R}) \ln(1 - O_{pk,R})] \\
&\quad + \frac{1}{2} \sum_{k=1}^M [D_{pk,I} \ln O_{pk,I} + (1 - D_{pk,I}) \ln(1 - O_{pk,I})] \\
&= \frac{1}{2} \sum_{k=1}^M [D_{pk,R} \ln f_k^o(\text{net}_{pk,R}^o) + (1 - D_{pk,R}) \ln(1 - f_k^o(\text{net}_{pk,R}^o))] \\
&\quad + \frac{1}{2} \sum_{k=1}^M [D_{pk,I} \ln f_k^o(\text{net}_{pk,I}^o) + (1 - D_{pk,I}) \ln(1 - f_k^o(\text{net}_{pk,I}^o))]
\end{aligned} \tag{22}$$

where M is the total number of the output neurons, $f_k^o(\text{net}_{pk,R}^o)$, $f_k^o(\text{net}_{pk,I}^o)$ are the real part and the imaginary part of the actual outputs of the k th output neuron respectively. $D_{pk,R}$, $D_{pk,I}$ are

the real part and the imaginary part of the desired outputs of the k th output neuron. Meanwhile the back propagation will be changed, and the forward propagation remains unchanged.

4.1 The adaptation rule of the output layer

The gradients of modified error function E_p with respect to $f_k^o(\text{net}_{pk,R}^o)$ and $f_k^o(\text{net}_{pk,I}^o)$ are as follows,

$$\begin{aligned} \frac{\partial E_p}{\partial f_k^o(\text{net}_{pk,R}^o)} &= \frac{1 - D_{pk,R}}{1 - f_k^o(\text{net}_{pk,R}^o)} - \frac{D_{pk,R}}{f_k^o(\text{net}_{pk,R}^o)} \\ &= \frac{f_k^o(\text{net}_{pk,R}^o) - D_{pk,R}}{f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))} \end{aligned} \quad (23)$$

$$\begin{aligned} \frac{\partial E_p}{\partial f_k^o(\text{net}_{pk,I}^o)} &= \frac{1 - D_{pk,I}}{1 - f_k^o(\text{net}_{pk,I}^o)} - \frac{D_{pk,I}}{f_k^o(\text{net}_{pk,I}^o)} \\ &= \frac{f_k^o(\text{net}_{pk,I}^o) - D_{pk,I}}{f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))} \end{aligned} \quad (24)$$

Finally we can get

$$\begin{aligned} w_{kj,R}^o(t+1) &= w_{kj,R}^o(t) + \Delta w_{kj,R}^o(t) \\ &= w_{kj,R}^o(t) - \eta \frac{\partial E_p}{\partial w_{kj,R}^o} \\ &= w_{kj,R}^o(t) - \eta (f_k^o(\text{net}_{pk,R}^o) - D_{pk,R}) i_{pj,R} - j (f_k^o(\text{net}_{pk,I}^o) - D_{pk,I}) i_{pj,I} \end{aligned} \quad (25)$$

And

$$\frac{\partial E_p}{\partial w_{kj,R}^o} + j \frac{\partial E_p}{\partial w_{kj,I}^o} = -\frac{1}{2} [(D_{pk,R} - f_k^o(\text{net}_{pk,R}^o)) - j(D_{pk,I} - f_k^o(\text{net}_{pk,I}^o))] i_{pj}^* \quad (26)$$

In order to simply the update equation, we also introduce the error term

$$\delta_{pk}^o = D_{pk,R} - f_k^o(\text{net}_{pk,R}^o) - j(D_{pk,I} - f_k^o(\text{net}_{pk,I}^o)) \quad (27)$$

Thus the factors $(D_{pk,R} - O_{pk,R})f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))$ and $(D_{pk,I} - O_{pk,I})f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))$ are replaced by $(D_{pk,R} - O_{pk,R})$ and $(D_{pk,I} - O_{pk,I})$. Therefore, back propagation is now directly propagated to the difference between the desired value and the actual value. Formula (27) lacks the factors $f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))$ and $f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))$, so "true" error is measured.

4.2 The adaptation rule of the hidden layer

The adaptation rule of the hidden layer still is

$$w_{kj}^h(t+1) = w_{kj}^h(t) - \eta \left(\frac{\partial E_p}{\partial w_{kj,R}^h(t)} + j \frac{\partial E_p}{\partial w_{kj,I}^h(t)} \right) \quad (28)$$

Similarly, we can have

$$\begin{aligned}
\frac{\partial E_p}{\partial w_{kj}^h} = & \frac{f_k^o(\text{net}_{pk,R}^o) - D_{pk,R}}{f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))} f_k^{\prime o}(\text{net}_{pk,R}^o) w_{kj,R}^o f^{\prime h}(\text{net}_{pj,R}^h) x_{pi,R} \\
& - \frac{f_k^o(\text{net}_{pk,R}^o) - D_{pk,R}}{f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))} f_k^{\prime o}(\text{net}_{pk,R}^o) w_{kj,I}^o f^{\prime h}(\text{net}_{pj,I}^h) j x_{pi,I} \\
& + \frac{f_k^o(\text{net}_{pk,I}^o) - D_{pk,I}}{f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))} f_k^{\prime o}(\text{net}_{pk,I}^o) w_{kj,I}^o f^{\prime h}(\text{net}_{pj,I}^h) j x_{pi,I} \\
& + \frac{f_k^o(\text{net}_{pk,I}^o) - D_{pk,I}}{f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))} f_k^{\prime o}(\text{net}_{pk,I}^o) w_{kj,R}^o f^{\prime h}(\text{net}_{pj,R}^h) x_{pi,R}
\end{aligned} \tag{29}$$

And,

$$\begin{aligned}
\frac{\partial E_p}{\partial w_{kj}^h} = & - \frac{f_k^o(\text{net}_{pk,R}^o) - D_{pk,R}}{f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))} f_k^{\prime o}(\text{net}_{pk,R}^o) w_{kj,R}^o f^{\prime h}(\text{net}_{pj,R}^h) x_{pi,I} \\
& + \frac{f_k^o(\text{net}_{pk,R}^o) - D_{pk,R}}{f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))} f_k^{\prime o}(\text{net}_{pk,R}^o) w_{kj,I}^o f^{\prime h}(\text{net}_{pj,I}^h) j x_{pi,R} \\
& + \frac{f_k^o(\text{net}_{pk,I}^o) - D_{pk,I}}{f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))} f_k^{\prime o}(\text{net}_{pk,I}^o) w_{kj,I}^o f^{\prime h}(\text{net}_{pj,I}^h) j x_{pi,R} \\
& - \frac{f_k^o(\text{net}_{pk,I}^o) - D_{pk,I}}{f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))} f_k^{\prime o}(\text{net}_{pk,I}^o) w_{kj,R}^o f^{\prime h}(\text{net}_{pj,R}^h) x_{pi,I}
\end{aligned} \tag{30}$$

Finally, we can get

$$\delta_{pj}^h = f_j^{\prime h}(\text{net}_{pj,R}^h) \text{Re} \left(\sum_{k=1}^M \delta_{pk}^o w_{kj}^{o*} \right) + j f_j^{\prime h}(\text{net}_{pj,R}^h) \text{Im} \left(\sum_{k=1}^M \delta_{pk}^o w_{kj}^{o*} \right) \tag{31}$$

where $\delta_{pk}^o = D_{pk,R} - f_k^o(\text{net}_{pk,R}^o) - j(D_{pk,I} - f_k^o(\text{net}_{pk,I}^o))$. Obviously the above formula is the same with the traditional CVBP algorithm in form. But there are no $f_k^o(\text{net}_{pk,R}^o)(1 - f_k^o(\text{net}_{pk,R}^o))$ and $f_k^o(\text{net}_{pk,I}^o)(1 - f_k^o(\text{net}_{pk,I}^o))$, the "true" error can be measured.

5 The new complex-valued WNN

Complex-valued wavelet artificial neural network used Mexican hat wavelet and Haar wavelet function as hidden layer AF instead of logarithmic sigmoid activation function. In this paper Morlet wavelet (or Gabor wavelet) function is chosen as the three hidden layer AF, which is defined by

$$\psi_{Morlet} = \cos(ax) e^{-bx^2} \tag{32}$$

in which $a = 1.75b = 0.5$ via experimentation.

Like new complex-valued neural network, activation function of output layer is chosen as logarithmic sigmoid in proposed CVWANN structures. Error function is chosen as the modified error function in Eqs. (22). Mathematical formulations of proposed CVWNN structures are obtained by using wavelet function instead of using logarithmic sigmoid function. CVWNN architectures used in this paper is shown in Figure 2.

6. Simulation results

In order to verify the validity and practicability of the proposed method, the paper carries out the processing of the XOR problem. The learning pattern is called the similar XOR problem, which is shown in Table 1. The real part of the output can be seen as the XOR of the input's real and input's imaginary part, and the imaginary part of the output is equal to the real part of the input.

This problem has been simulated with a 1-3-1 complex-value network in [12, 15]. In this paper the new complex-valued WNN and conventional CVBPNN and CVWNN are applied to resolve the similar XOR problem. For the above methods, learning rate and maximum iteration number are chosen as 0.1 and 5,000, respectively. The architecture of the complex-valued neural network is 1-2-1. When the minimal error is 0.1, 0.01 and 0.001, the learning curve for the similar XOR problem is shown is Figure 3, Figure 4, and Figure 5 respectively. The success rate and average learning epochs are shown in Table 2. When the error criteria was used as the $E_p = 0.1$, the new CVWNN has 100% success rate, and its average learning epoch is only 153. This is a third of the conventional CVWNN's 475 epochs. When the error criteria is $E_p = 0.01$ and $E_p = 0.001$, the improved algorithm is faster than conventional, which can be seen from the Figure 3 and Figure 4 and Table 2.

Table 1. Learning pattern for similar XOR problem

Input pattern	Output pattern
0	0
i	1
1	1+i
1+i	i

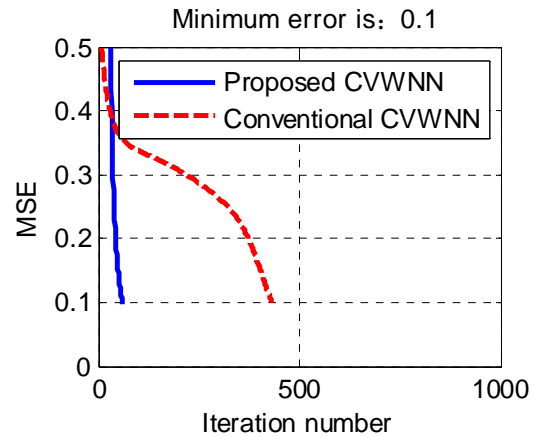


Figure 3. The comparison of the learning curve between the proposed CVWNN and the conventional CVWNN, when the minimal error is 0.1

Table 2. Simulation results for similar XOR problem

	Success rate		Average Iterations	
	Proposed CVWNN	Conventional CVWNN	Proposed CVWNN	Conventional CVWNN
Minimal error=0.1	100%	100%	153	394
Minimal error=0.01	100%	98%	380	608
Minimal error=0.001	99%	95%	2189	3743

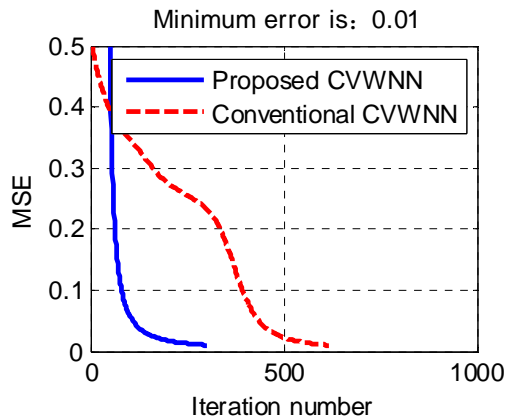


Figure 3. The comparison of the learning curve between the proposed CVWNN and the conventional CVWNN, when the minimal error is 0.01

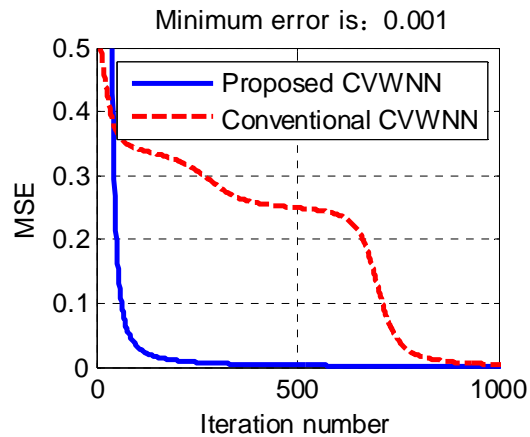


Figure 4. The comparison of the learning curve between the proposed CVWNN and the conventional CVWNN, when the minimal error is 0.001

A comparison of the proposed method and the conventional CVWNN is illustrated in Figure 3, 4, and 5, from which we can see that the proposed method has better stability convergence performance, and faster running speed than the conventional CVWNN.

7. Conclusion

In this paper, a new CVWNN is proposed, whose inputs, outputs and weights are all complex-valued, and the nonlinear activation function remains real-valued. The back propagation learning algorithm for training the complex-valued wavelet network is modified by introducing a new error function. The performance of the proposed CVWNN is illustrated with application to the XOR. The simulation results demonstrate that the CVWNN has better stability convergence performance, and faster running speed than the conventional CVWNN. Anymore, in signal processing and communication areas, there are a large number of complex-valued number to be dealt with,, thus the proposed CVWNN provides a powerful tool for such cases.

References

- [1] P. Hong, X. Liang-Zheng. Efficient Object Recognition Using Boundary Representation and Wavelet Neural Network. *IEEE Transactions on Neural Networks*. 2008; 19: 2132-2149.
- [2] Z. Qinghua. Using wavelet network in nonparametric estimation. *IEEE Transactions on Neural Networks*. 1997; 8: 227-236.
- [3] RH. Abiyev, O. Kaynak. Fuzzy Wavelet Neural Networks for Identification and Control of Dynamic Plants—A Novel Structure and a Comparative Study. *IEEE Transactions on Industrial Electronics*. 2008; 55: 3133-3140.
- [4] S. Yilmaz, Y. Oysal. Fuzzy Wavelet Neural Network Models for Prediction and Identification of Dynamical Systems. *IEEE Transactions on Neural Networks*. 2010; 21: 1599-1609.
- [5] Y. Ozbay, S. Kara, F. Latifoglu, R. Ceylan, M. Ceylan. Complex-valued wavelet artificial neural network for Doppler signals classifying. *Artif Intell Med*. 2007; 40: 143-56.
- [6] C. Li, X. Liao, J. Yu. Complex-valued wavelet network. *Journal of Computer and System Sciences*. 2003; 67: 623-632.
- [7] M. Ceylan, H. Yacar. Blood vessel extraction from retinal images using Complex Wavelet Transform and Complex-Valued Artificial Neural Network. in 2013 36th International Conference on Telecommunications and Signal Processing (TSP). 2013: 822-825.
- [8] J. Zhe, S. Zhihuan, H. Jiaming. Behavioral Modeling of Wideband RF Power Amplifiers Using Complex-valued Wavelet Networks. in 2006 International Conference on Communications, Circuits and Systems Proceedings. 2006: 820-824.
- [9] G. Meijuan, T. Jingwen, Z. Shiru. Modeling for mobile communication fading channel based on wavelet neural network. in International Conference on Information and Automation, 2008. ICIA 2008. 2008: 1566-1570.

-
- [10] RT. Anupam Shukla, Hemant Kumar Meena, Rahul Kala. Speaker Identification using Wavelet Analysis and Modular Neural Networks. *Journal of Acoustic Society of India (JASI)*. 2009.
- [11] MKS. a A. Shahbahrami. *Classification Of Ecg Arrhythmias Using Discrete Wavelet Transform And Neural Networks*. International journal of Computer Science, Engineering and Application (IJCSEA). 2012; 12: 1-13.
- [12] ZT. a CV. Xiaoming Chen, Songsong Li, Toshimi Okada. A Modified Error Backpropagation Algorithm For Complex-Value Neural Networks. *International Journal of Neural Systems*. 2005; 15: 435–443.
- [13] AS. Shafie, IA. Mohtar, S. Masrom, N. Ahmad. *Backpropagation neural network with new improved error function and activation function for classification problem*. in 2012 IEEE Symposium on Humanities, Science and Engineering Research (SHUSER), 2012: 1359-1364.
- [14] T. Nitta. *Complex-Valued Neural Networks*. IGI Global. 2009.
- [15] T. Nitta. An Extension of the Back-Propagation Algorithm to Complex Numbers. *Neural Networks*. 1997; 10: 1391-1415.
- [16] M. J. Sufang Li. A modified complex-valued BP neural network. *Journal of Computational Information Systems*. 2014; 10: 1-13.