Symmetry Detection and Pattern Recognition Using Complex Value Neural Networks

Ngong Ivoline-Clarisse Kieleh

Computer Engineering, Faculty of Engineering and Technology
Konya Technical University, Konya, 42075 Turkey
ivolinengong@gmail.com

Abstract

In this paper, we apply the backpropagation algorithm on complex numbers. We show the potential computational power Complex-Valued Neural Networks possess by solving the classic XOR problem with a 2 layered complex-valued neural network which cannot be solved with a 2 layered real-valued neural network. The study also explores 4 different bit lengths in order to detect symmetry. The effect of the learning rate and mapping angle on the CVNN are investigated and it is observed that CVNN is a powerful tool for symmetry detection.

1. Introduction

The use of complex numbers in neural networks is currently growing at a speedy rate. Lately, the range of applications has expanded to fields like remote sensing, optoelectronics, image processing, information processing, quantum neural devices and systems. [1]

Complex numbers are represented as an ordered pair of real numbers that is the real and imaginary parts (amplitude and phase). This characteristic of complex numbers makes it very useful. Though a variety of useful neural systems concentrate on the phase information as it is thought of to be more important than the magnitude, in some fields like image processing, the performance significantly improves when both information is processed. [2] The level of freedom in CVNNs or self-organization can be reduced to achieve better generalization characteristics, which is beneficial with frequency-domain treatment through Fourier transform. [3] Phase information was first introduced by Eiichi Goto in 1954 in his invention of "Parametron" where he used the phase of a high-frequency carrier to represent binary or multivalued information.[1]. In recent times, researchers have extended the structure of complex-amplitude information for use in the pattern processing fields. In image processing for example, a method is proposed where they identify a point scattering function in the frequency domain that is used for adaptive processing for blur compensation.[4] Hirose in another publication found that the most important characteristic in CVNNs is the phase rotation in complex multiplication.[3] Gangal, Kalra and Chauhan compare the performance of complex valued neural networks using various error functions where they found that for some error functions the performance of the CVNN depended on the architecture while for a few other error functions, the convergence speed of the network.[5] Ceylan in his study presented a new version CVANN for complex- valued pattern recognition and classification. The method combines two complex artificial neural networks 6]. Goh on his part proposed a back propagation neural network for modeling complex systems.[7]

CVNN is therefore useful not just today but in the future as it has many applications. It is important to throw more light on the computational power of CVNN. As indicated in many studies, CVNNs has many advantages over Real Valued Neural Networks (RVNN)[8-10].

2. Materials and Methods

Just like in RVNN, the CVNN process can be broken in a few steps:

- Feed the input
- Find weighted sums and apply activation function
- Get the output from the network
- Get error by comparing desired output with derived output
- Back propagate the error from output to input
- Update weights and repeat process

2.1 Forward Processing in CVNN

A 3-layered complex valued neural network is shown, input, hidden layer and output. A 2-input 2-output CVNN with a 3-neuron hidden layer is shown in Fig. 1.

In this complex valued neural network:

X input values

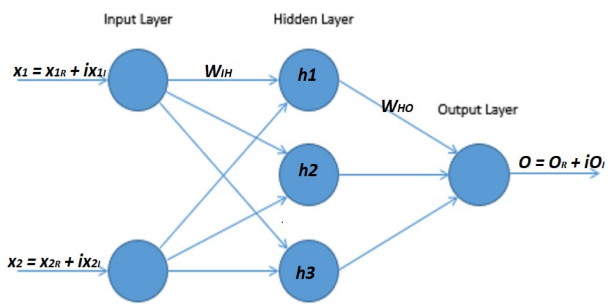
H hidden neurons

O output neurons

K learning rate

W_{IH} the weight connecting the input neuron to a neuron in the hidden layer

W_{HO} the weight connecting the hidden neuron to the output neuron



After feeding the inputs X1 and X2 into the network, we go ahead to calculate the outputs of the hidden neurons and the output vector of the output neuron.
The weighted sums of the input values are calculated and then passed to an activation function. The result of the weighted sum is compared to a certain threshold value from the activation function, if it exceeds this threshold, the neuron will be activated and not activated otherwise. The output of the hidden neuron is calculated as:

$$H = f\left(\sum W_{IH} X\right) \qquad (1)$$

If the neuron is activated the output of the hidden neuron is forwarded to the output neuron the weighted sum is calculated as well and an activation function is applied thus:

$$O = f\left(\sum W_{HO} X\right) \qquad (2)$$

Where $f(x)$ is an activation for each vector element x defined as

$$f(x) = \tanh(|x|)\exp ¿ \qquad (3)$$

The activation used for both the hidden and output layers is the tangent.

2.2 Back-propagation Processing in CVNN

After getting the output, we can now get find the error. The error function used is the least mean square function which is defined as:

$$E=\frac{1}{2}\sum_{k=1}^{K}\left|O\left(X_k\right)-O_K\right|^2 \qquad (4)$$

The operations for updating the weights and thresholds are:

$$\left|W_{HO}^{new}\right|=\left|W_{HO}^{old}\right|-K\frac{\partial E}{\partial \vee W_{HO} \vee ¿¿}$$

(5)

$$arg\,W_{HO}^{new}=arg\,W_{HO}^{old}-K\frac{}{¿\,W_{HO}\vee ¿}$$

(6)

$$\frac{\partial E}{\partial\left|W_{HO}\right|}=\left(1-|O|^2\right)\left(|O|-|\hat{O}|\cos\left(arg\,O-arg\,\hat{O}\right)\right)|¿$$

(7)

$$¿\,W_{HO}\vee ¿\frac{\partial E}{\partial\left(arg\,W_{HO}\right)}=\left(1-|O|^2\right)\left(|O|-|\hat{O}|\cos\left(a\right.\right.$$

(8)

Where K is the learning rate.

And finally we update the H neurons

$$H=\left(f\left(O\right)^{¿}W_{HO}\right)¿*¿ \qquad (9)$$

2.3 Detection of Symmetry

The goal here is to determine if a one-dimensional array of binary bits that is symmetrical around the center. In the case of 3 inputs, Table 1 shows the input-output mapping.

If the output is 1, then input is symmetrical else it isn't. In the case of CVNN, we have 2 outputs, y1 and y2. Such that when the input is symmetrical y1 is 1 and y2 is 0therwise, as shown in Table 2.

Table 1 Symmetry Detection with 3 input

Input			Output
X_1	X_2	X_3	y
0	0	0	1
0	0	1	0
0	1	0	1
1	0	0	0
0	1	1	0
1	0	1	1
1	1	0	0
1	1	1	1

Table 2 Detection of Symmetry with 3 inputs and 2 outputs

Input			Output	
X_1	X_2	X_3	Y_1	Y_2
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0
1	0	0	0	1
0	1	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

Notice that the values in the tables are real number values. Hence, it is necessary to convert the data to complex values. This conversion operation is typically done by angle based encoding. Using Euler's formula we get the complex number:

$$z=e^{j\varphi}=\cos\varphi+j\,.\,sin\varphi \qquad (10)$$

Where

$$\varphi=\frac{\theta\left(x-a\right)}{b-a} \qquad (11)$$

3. Experimental Results

The XOR Problem (which is shown in Table 3) was solved using the CVNN. In order to solve the problem using the complex valued network, the input-output mapping is encoded as shown in Table 4. With a learning rate of 0.2 and maximum iteration of 3000 during training the target and outputs of the CVNN are shown in Table 5. An accuracy of 90.6% was gotten.

Table 3 XOR Problem with 4 patterns

Inputs		Output
X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Table 4 XOR Problem for 16 Complex-Valued Patterns

Input 1	Input 2	Output
0	0	1
0	i	i
i	0	0
i	i	1+i
i	1	i
1	1	1+i
1+i	i	i
1+i	1+i	1
0	1	i
0	1+i	0
i	1+i	0
1	0	0
1	i	i
1	1+i	0
1+i	0	0
1+i	1	i

Table 5 Results from CVNN

Target	Output of CVNN
1	0,999 + 0,000i
i	0,002 + 0,997i
0	0,008 + 0,483i
1+i	0,708+ 0,706i
i	0,001 + 0,998i
1+i	0,706 + 0,708i
i	0,002 + 0,999i
1	0,993 + 0,002i
i	0,002 + 0,999i
0	0,003 + 0,883i
0	0,000 + 0,850i
0	0,001 + 0,758i
i	0,000 + 0,999i
0	0,003 + 0,853i
0	0,006 + 0,912i
i	0,002+ 0,999i

3.1 Learning Rate

The performance of the CVNN can also depend on the relation between the learning rate and the mapping angle. This is investigated and confusion matrices are presented in Table 6. The results were gotten from 5-bit length symmetry detection.

3.2 Symmetry With Different Lengths

The simulated neural network consisted of 2 inputs, one layer (3 neurons) and 1 output.

Symmetry detection was done for 4 different bit lengths:

3 Inputs – 2 Outputs
5 Inputs – 2 Outputs
7 Inputs – 2 Outputs
9 Inputs – 2 Outputs

The confusion matrices gotten from the simulations are presented in Table 6. A constant learning rate of 0.2 and 3000 iterations were used.

Table 6 Learning Rate

Learning Rate	φ	Confusion Matrix

0.25	π	$\begin{bmatrix}2 & 0\\ 7 & 23\end{bmatrix}$
	π/2	$\begin{bmatrix}4 & 0\\ 5 & 23\end{bmatrix}$
	π/3	$\begin{bmatrix}5 & 0\\ 3 & 23\end{bmatrix}$
0.2	π	$\begin{bmatrix}1 & 3\\ 8 & 20\end{bmatrix}$
	π/2	$\begin{bmatrix}5 & 6\\ 4 & 17\end{bmatrix}$
	π/3	$\begin{bmatrix}5 & 0\\ 4 & 23\end{bmatrix}$
0.1	π	$\begin{bmatrix}3 & 6\\ 6 & 17\end{bmatrix}$
	π/2	$\begin{bmatrix}1 & 3\\ 8 & 20\end{bmatrix}$
	π/3	$\begin{bmatrix}4 & 5\\ 5 & 18\end{bmatrix}$
0.5	π	$\begin{bmatrix}3 & 10\\ 9 & 13\end{bmatrix}$
	π/2	$\begin{bmatrix}2 & 4\\ 7 & 19\end{bmatrix}$
	π/3	$\begin{bmatrix}3 & 1\\ 6 & 22\end{bmatrix}$

Table 7 Results for Symmetry Detection

Structure	φ	Confusion Matrix
3bit Complex	π	$\begin{bmatrix}4 & 1\\ 0 & 3\end{bmatrix}$
	π/2	$\begin{bmatrix}4 & 2\\ 0 & 2\end{bmatrix}$
	π/3	$\begin{bmatrix}3 & 1\\ 1 & 3\end{bmatrix}$
5bit Complex	π	$\begin{bmatrix}1 & 3\\ 8 & 20\end{bmatrix}$
	π/2	$\begin{bmatrix}4 & 7\\ 5 & 16\end{bmatrix}$
	π/3	$\begin{bmatrix}5 & 6\\ 4 & 17\end{bmatrix}$
7bit Complex	π	$\begin{bmatrix}6 & 26\\ 9 & 87\end{bmatrix}$
	π/2	$\begin{bmatrix}3 & 19\\ 12 & 94\end{bmatrix}$
	π/3	$\begin{bmatrix}3 & 19\\ 12 & 94\end{bmatrix}$
9bit complex	π	$\begin{bmatrix}3 & 20\\ 12 & 93\end{bmatrix}$
	π/2	$\begin{bmatrix}0 & 14\\ 29 & 469\end{bmatrix}$
	π/3	$\begin{bmatrix}0 & 9\\ 29 & 474\end{bmatrix}$

Conclusion

In this study, a complex back-propagation algorithm has been used for pattern recognition. The XOR problem and the detection of symmetry problem which cannot be solved using a 2-layered real-valued neural network [11], has been solved using a 2-layered complex-valued neural network. This shows the computational power of CVNNs should. In addition, the effect of the mapping angle and learning rate in symmetry detection is also studied. According to the results, if the learning rate is small enough, the mapping angle converges faster. π/3 is the most powerful mapping angle. It can also be noted that, the value of the learning rate should be chosen carefully because a small change in the mapping angle causes a big change in the CVNNs performance.

References

[1] A. Hirose, “ Complex Valued Neural Networks – Advances and Applications”, IEEE Press,2013

[2] Y. Acar, M. Ceylan, E. Yaldiz, “Complex-valued neural network design to classify real-valued balanced/imbalanced data”

[3] A. Hirose,”Nature of complex number and complex-valued neural networks”, Higher Education Press and Springer-Verlag Berlin Heidelberg,2011.

[4] I. Aizenberg, D. Paliy, J. Zurada, J. Astola. “Blur identification by multilayer neural network based on

multivalued neurons”, IEEE Transactions on Neural Networks,May 2008.

[5] A. Gangal, P. Kalra, and D.Chauhan, “Performance Evaluation of Complex Valued Neural Networks Using Various Error Functions”, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:1, No:5, 2007.

[6] M. Ceylan,” Combined Complex-Valued Artificial Neural Network (CCVANN)”, Proceedings of the World Congress on Engineering 2011 Vol II.

[7] A. Goh, “Back-propagation neural networks for modeling complex systems”, Artificial Intelligence in Engineering 9, 1995 Elsevier Science Limited.

[8] F. Amin, K. Murase,”Single-layered complex-valued neural network for real-valued classification problems”, Neurocomputing, 2009.

[9] T. Nitta,”An Extension of the Back-Propagation Algorithm to Complex Numbers”, Neural Networks, 1997.

[10] K. Terabayashi,R. Natsuaki, and A. Hirose,”Ultrawideband Direction-of-Arrival Estimation Using Complex-Valued Spatiotemporal Neural Networks”, IEEE Transactions on Neural Networks and Learning Systems, 2014

[11] T. Nitta, “Solving the XOR problem and the detection of symmetry using a single complex-valued neuron”, Neural Networks,2003.