International Journal of the Physical Sciences Vol. 6(7), pp. 1766-1772, 4 April, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.105

Full Length Research Paper

New activation functions for complex-valued neural network

Hamid A. Jalab[1]* and Rabha W. Ibrahim[2]

[1]Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University Malaya, 50603 Kuala Lumpur, Malaysia.
[2]School of Mathematical Sciences, Faculty of Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor Darul Ehsan, Malaysia.

Accepted 14 March, 2011

This paper presents a new types of complex-valued sigmoid function for a fully multi-layered complex-valued neural network (CVNN). By using the concept of the subordination between analytic functions in open disc, we able to study the reducibility of CVNN. A real-world problem example has been used as a classifier. The simulations results reveal that the proposed fully complex-valued network, been better trained reduces the testing time by 54% compared to the choice of using the traditional sigmoid activation function.

Key words: Complex valued neural network, activation functions, reducibility, irreducibility, subordination.

INTRODUCTION

The study on theory and applications of artificial neural network had increased because of their outstanding capability of fitting nonlinear models. Neural network had successfully been applied across an extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology and physics due to their strong capacity to handle complex problems and to improve system performance (Subramaniam et al., 2010; Taqa and Jalab, 2010). Artificial neural network is a mathematical model which emulates the activity of biological neural networks in the human brain. Each neuron in the ANN (Artificial neural network) has a number of inputs and one output (Sivanandam, 2006).

The complex valued neural network are those neural network whose weights, threshold values, input, output signals all are complex numbers and the activation function and its derivatives have to be well behaved every where in the complex plane (Kim and Adali, 2002). However, the complex valued neural network is extending its field both in theories and applications. They are used to express real-world phenomena like time series analysis, signal amplitude and phase, and to analyze various mathematical and geometrical relationships. Also the complex valued neural network (CVNN) had shown more powerful capability than real-valued neural network in processing real-valued signals (Nitta, 2004a).

In complex-valued neural network, one of the main problems is selecting of nodes activation function (Kim and Adali, 2002). In real case, the node activation function is usually chosen to be a continuous, bounded and nonconstant function. These conditions on the activation function are very mild and there is no problem in selecting a real function that satisfies these requirements and that is also smooth (derivative exists). In CVNN, any regular analytic function cannot be bounded unless it reduces to a constant. This is known as the Liouvilles's theorem. In complex case, the main constraints that the activation function should satisfy can be found in literatures (Georgin and Koutsougeras, 2002; Haykin, 2008; Ganesh and Balasubramanian, 2009).

Nitta (2004b) studied the reducibility of multilayer complex-valued neural network, in which the reducibility is expressed by $n\pi/2$ rotation equivalence instead of sign equivalence which is an extension to Sussmann's work of the real-valued neural network . Sussmann (1992) presented necessary and sufficient conditions to reduce the number of hidden neurons for

*Corresponding author. E-mail: hamidjalab@um.edu.my

real-valued neural networks, and using the important notion reducibility/ irreducibility of the real valued neural network he devised. He proved that the 3-layered real-valued neural network was uniquely determined by its input-output map, up to an obvious finite group, provided that the real-valued neural network was irreducible (*Uniqueness Theorem*). Thus, the reducibility is closely related to the redundancy of the real-valued neural network, and is needed for proving the uniqueness theorem. The uniqueness theorem is important for investigating the properties based on the hierarchical structure of the real-valued neural network.

COMPLEX-VALUED ACTIVATION FUNCTIONS

Complex plane is two dimensional with respect to real numbers and is one dimensional with respect to complex number. The complex numbers have a magnitude associated with them and a phase that locates the complex number uniquely on the plane. Here, we consider the proposed activation function which maps complex-values into complex and has the form of $F : \mathbb{C} \rightarrow \mathbb{C}$

$$F(z) = f_{\mathbb{R}}(x) + i f_{\mathbb{R}}(y), \ (z \in \mathbb{C}) \tag{1}$$

where in general

$$f(a) = \frac{tanh\ (a)}{1-(a-3)\,e^{-a}}$$

This arrangement ensures that the magnitude of real and imaginary part of $F(z)$ is bounded between -1 and 1. But now the function $F(z)$ is no longer holomorphic, because the Cauchy-Riemann equation does not hold (Goodman, 1983), that is,

$$\partial \frac{F(z)}{\partial x} + i\partial \frac{F(z)}{\partial y} \neq 0.$$

So, effectively, the holomorphy is compromised for boundedness of the activation function. Our consideration of $F(z)$ is held between the input layer and the hidden layer while the function $G(z) = tanh(z)$ is considered between the hidden and the output (Figures 1 and 2).

Here, we consider the following 3-layers complex-valued neurons, there is one hidden layer between the input and output layers. The input signals, weights, thresholds and output signals are all complex numbers.

The net input U_n to a complex-valued neuron, n is defined as:

$$U_n = \sum_m W_{mn} X_m + T_n, \tag{2}$$

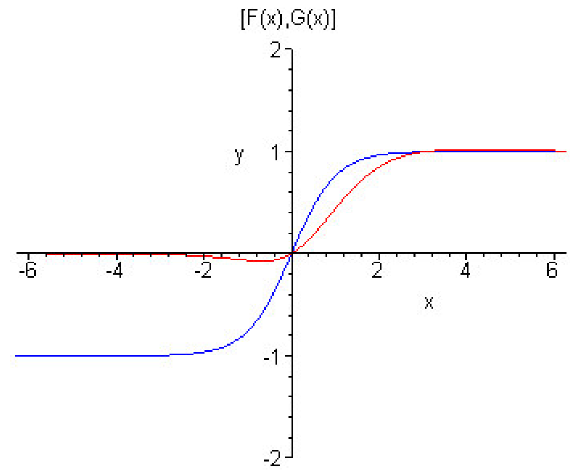

Figure 1. Activation function F and G.

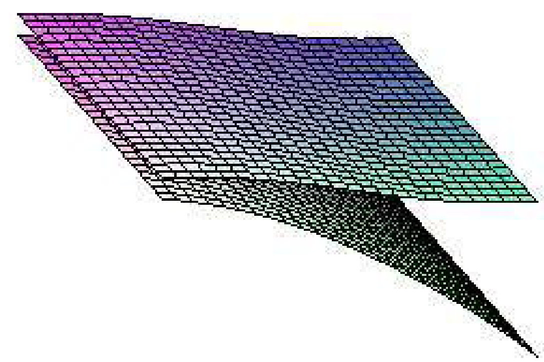

Figure 2. Subordination between F and G

where W_{mn} is the (complex-valued) weight connecting the complex-valued neurons m and n, T_n is the (complex valued) threshold value of the complex-valued neuron n, and X_m is the (complex-valued) input signal from the complex-valued neuron m. To obtain the (complex-valued) output signal, convert the net output U_n into its real and imaginary parts as follows: $U_n = x + iy = z$. The (complex-valued) output signal of the hidden and the output neurons are defined as respectively.

$$\Sigma(z) = \frac{\tanh\ (x)}{1-(x-3)\,e^{-x}} + i\,\frac{\tanh\ (y)}{1-(y-3)\,e^{-y}}, \tag{3}$$

$$\sigma(z) = \tanh(x) + i\tanh(y) \tag{4}$$

Assume that $w_{ij} \in C$ is the weight between the input neuron i and the hidden neuron j; $c_j \in \mathbb{C}$ the weight between the hidden neuron j and the output neuron; $s_j(z)$ denote the output values of the neuron j; $g_k(z)$ denotes the output neuron for the input pattern $z=[z_1,...,z_m]^t$, and let $v_j(z)$ and $u_k(z)$ denote the net inputs to the hidden neuron j and the output neuron for the input pattern $z \in \mathbb{C}^m$, respectively.

That is $v_j = \sum_{i=1}^{m} w_{ij} z_i + t_j$, where t_j is the threshold of the hidden neuron j, $u_k(z) = \sum_{j=1}^{n} c_j s_j(z) + c_k$, where c_k is the threshold of the output neuron, $s_j(z) = \Sigma(v_j(z))$ and $g_k(z) = \sigma(u(z))$. Denoted by $N_{m,n}$, the set of all $m-n-1$ complex-valued neural network described previously is the object of this work.

To illustrate our main results, we need the following concept. Given two functions, $F(z)$ and $G(z)$, which are analytic in open disc, the function $F(z)$ is said to be subordinate to $G(z)$ denoted by $F(z) \prec G(z)$ if there exists a function $h(z)$, analytic in open disc with $h(0)=0$ and $|h(z)|<1$ such that $F(z)=G(h(z))$ (Miller and Mocanu, 2000). More applications of this concept can be found in Ibrahim and Darus (2008).

REDUCIBILITY OF THE CVNN

Here, we show the reducibility of the complex-valued neural network described in 'Complex-valued activation functions'. First, we need the following preliminaries in the sequel (Nitta, 2004b):

1. For a fixed m, two complex-valued neural network $N_1 \in N_{m,n1}$ and $N_2 \in N_{m,n2}$ are called $I-O$ equivalent if their corresponding complex-valued functions are the same. It is not essential that the number of neurons or parameters in the layers are equal.
2. Two complex-valued linear affine functions $\alpha : \mathbb{C}^m \to \mathbb{C}$ and $\beta : \mathbb{C}^m \to \mathbb{C}$ are called $n\pi/2$ rotation-equivalent if one of the following conditions holds:

$$\forall z \mathbb{C}^m; \alpha(z) = \beta(z)\ (= exp[i0].\beta(z)) \tag{5}$$

$$\forall z \mathbb{C}^m; \alpha(z) = -\beta(z)\ (= exp[i\pi].\beta(z)) \tag{6}$$

$$\forall z \mathbb{C}^m; \alpha(z) = i\beta(z)\ (= exp[i\frac{\pi}{2}].\beta(z)) \tag{7}$$

$$\forall z \mathbb{C}^m; \alpha(z) = -i\beta(z)\ (= exp[i\frac{3\pi}{2}].\beta(z)) \tag{8}$$

3. A complex-valued neural network $N \in N_{m,n}$ is called reducible if one of the following three conditions holds:

a) One of the weights between the hidden layer and the output neuron is zero: $1 \le \exists j \le n; c_j = 0$.
b) There exist two hidden neurons such that the net inputs to them are $n\pi/2$ rotation-equivalent: $1 \le \exists j_1; \exists j_2 \le n; v_{j1}$ and v_{j2} are $n\pi/2$ rotation-equivalent.
c) There exists a hidden neuron such that the net input to it is a constant: $1 \le \exists j \le n; v_j$ is a constant.

In the next result, we show how a 3-layered complex-valued neural network preserves the reducibility for sandwich (subordination and superordination) sigmoid activation functions, that is, $F(z) \prec G(z)$.

Theorem 1

If a 3-layered complex-valued neural network $N \in N_{m,n}$ of two subordination activation functions $(F(z) \prec G(z))$ is reducible, then it is I-O equivalent to another 3-layered complex-valued neural network with the activation function $G(z)$ and fewer hidden neurons.

Proof

Assume that a three-layered complex-valued neural network $N \in N_{m,n}$ is reducible.

Case I: Consider that $1 \le \exists j \le n$ such that $c_j = 0$. In this case by the subordination of F and G, that is, $F(0)=G(0)$. This implies that the hidden neuron does not have effect on the output of the complex-valued neural network, the hidden neuron j can be deleted. Hence their corresponding complex-valued functions are the same.

Case II: Let $1 \le \exists j_1, j_2 \le n$ such that v_{j_1} and v_{j_2} are $n\pi/2$ rotation equivalent. Let c_{j_1} be the weight between the hidden neuron j_1 and the output neuron, and c_{j_2} the weight between the hidden neuron j_2 and the output neuron. In this case, by the subordination of F and G ,that is, the image of F is a proper subset of the image of $G(F(U) \subset G(U))$, where U is an open disc in $\mathbb{C}$, we obtain the same input-output map by removing the

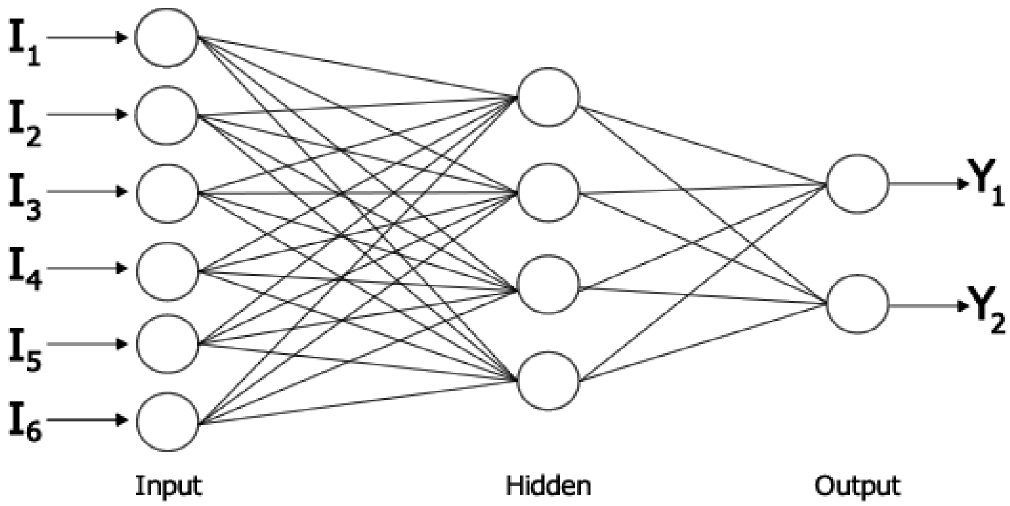

Figure 3. Three-layer feed-forward CVNN implementing the back propagation algorithm.

hidden neuron j_2 and changing the weight c_{j_1} to $c_{j_1} + rc_{j_2}$, where $r \in \{1, -1, i, -i\}$. Thus their corresponding complex-valued functions are still the same.

Case III: Finally, if for $1 \le \exists j \le n;\ v_j \equiv \delta,$ where δ is a constant. The same input-output map can be obtained by removingthe hidden neuron j and changing the threshold of the output neuron c to $c + s_j = c + \Sigma(v_j(z)) = c + \Sigma(\delta)$ because the output $\Sigma(v_j(z))$ of the hidden neuron j is a constant which leads that $\sigma(u(z))$ is a constant under the subordination between Σ and σ, that is, their corresponding complex-valued functions remain the same. This completes the proof.

Corollary 1

If a 3-layered complex-valued neural network $N \in N_{m,n}$ is reducible, then it is I-O equivalent to another 3-layered complex-valued neural network with fewer hidden neurons (Nitta, 2004b).

Corollary 2

If a 3-layered complex-valued neural network $N \in N_{m,n}$ has weight $w_{ij} = 0$, then it is I-O equivalent to another 3-layered complex-valued neural network with fewer hidden neurons.

IMPLEMENTATION

The neural network used in the simulation process is a 3-layer feed-forward CVNN implementing the back propagation algorithm as shown in Figure 3. In this network, all the inputs, outputs, weights, and biases are complex values. To implement our CVNN, we used Matlab R2009b on Intel(R) Core TM2Duo processor with 3.00 GHz and 3 GB RAM.

At the beginning of the learning process, the weight matrices between input and hidden layer and between hidden and output layer are initialized with the random complex values. Vectors for hidden neuron biases b1 and output neuron biases b2 are also initialized with random complex values. The goal of back propagation (BP) learning algorithm is to minimize the error-energy at the output layer. The neural network learns the relationships among sets of input-output data (training sets) that are characteristic of the component under consideration. First, input data are presented to the input neurons, and then output data are computed. The output data are compared with the desired value and the errors are computed. Further, error derivatives are calculated and summed up for each weight and bias until whole training set has been presented to the network. These error derivatives are then used to update the weights and biases for neurons in the model. The training process proceeds until errors lower than the prescribed values is reached. Once trained, the network provides a fast response for different input data.

We have tested the behavior of the neural network by using fully complex activation functions, verifying the

Table 1. Weights (complex values) between input layer and hidden layer.

Node	Node H1	Node H2	Node H3	Node H4
Input 1	0.701918692787938 + 0.219021586670218i	1.376393682831280 + 0.292232742536123i	0.091148032849257 + 1.016530388375190i	1.03777587861251 + 1.23026776583944i
Input 2	0.004612544021455 + 1.095807166349050i	0.191484488236877 + 0.127132376049907i	0.496416668904873 + 0.540523136153263i	0.470620411933422 + 1.39605474922888i
Input 3	0.380170225741953 + 0.088187685676538i	1.136936654333990 + 0.999558556492102i	0.417811427060826 + 0.286635042168629i	1.145834793604170 + 0.635027432710810i
Input 4	0.949555473578221 + 0.741861737013464i	1.35149224857271 0+ 1.00923039132627i	0.728083182731416 + 0.332296585286707i	0.902518962398014 + 0.048918713439151i
Input 5	0.607415857741603 + 1.327356842432400i	0.244507475736920 + 0.706615592016859i	1.179766452622360 + 0.622994627960622i	0.064772451991841 + 0.748245343249140i
Input 6	0.266892491698015 + 0.315022189982046i	0.865716138991662 + 0.373969313064443i	1.19191208111544 0+ 0.670712134181853i	0.059172055445492 + 0.045198273999110i

Table 2. Weights(complex values) between hidden layer and output layer.

	Node output1	Node output 2
Node H1	36.2407385621735 - 236.495229913697i	0.928191869810991 + 0.532431465140660i
Node H2	36.0075745920745 - 236.450555806004i	1.59082043936045 - 0.418882557002526i
Node H3	37.2207291386245 - 237.041399244184i	1.24633171732695 - 0.582224146084517i
Node H4	36.7091930058420 - 235.918493305041i	1.42438050039180 - 0.150026132837350i

correctness and analyzing the improvement of these functions over traditional artificial neural network (ANN) solutions to specific real-world problems. These steps are discussed as follows.

For the first model, in the hidden and output layers, the sigmoid activation function have been used as a transfer function. While, in the second model, for the same network, we used a pair of complex activation functions representing the real and the imaginary component of z:

1. For the transfer function of hidden layer:

$$y = tanh(z)/(1-(z-3)*exp(-z))$$

2. For the transfer function of output layer :

$$y = tanh(z)$$

3. The performance criterion used is the sum of square due to error(SSE).

A real-world problem illustrates using CVNN neural network as a classifier to identify the sex of crabs from its physical dimensions (data were taken from Mathworks). In this example, we built a classifier that can identify the sex of a crab from its physical measurements. Six physical characteristics of a crab are considered: Species, frontallip, rear width, length, width and depth. The six physical characteristics will act as inputs to a neural network and the sex of the crab will be target. The classification process consists of two phases: Training phase and testing phase.A training set is used in supervised training to present the proper network behavior, where each six inputs observed values for the physical characteristics of a crab is introduced with its corresponding correct target. As these inputs are applied to the network, the network outputs are compared to the targets. The neural network is expected to identify if the crab is male or female. A 1-hidden layer feed forward network is created with 4 neurons. The values of weights (complex values) between input layer and hidden layer obtained during training phase, are shown in Table 1, where the rows correspond to input nodes and columns correspond to hidden nodes, while Table 2 shows the values of weights (complex values) between hidden layer and output layer obtained during training phase. The rows correspond to hidden nodes and columns correspond to output nodes.

Table 3. Bias (complex values) at the hidden nodes.

Node H1	0.753173772130708 + 0.232078094143457i
Node H2	0.591588200811105 + 0.536528269078829i
Node H3	0.822891032240768 + 0.430618407870931i
Node H4	0.329578870595237 + 0.403581086734115i

Table 4. Bias (complex values) at the output nodes.

Node output 1	Node output 2
26.0483208415974 - 165.264442609909i	1.14856923573488 + 0.41888255700252i

Table 5. Comparison of testing time.

Activation function	Testing time (seconds)
Traditional sigmoid activation function	0.007170
Proposed complex activation function	0.003912

Table 6. Irreducible CVNN.

Activation function	Iteration number
Traditional sigmoid activation function	541
Proposed complex activation function	306

Table 7. Reducible CVNN.

Activation function	Iteration number
Traditional sigmoid activation function	379
Proposed complex activation function	267

The bias (complex values) at the hidden nodes, and at the output nodes are shown in Table 3 and 4, respectively.

RESULTS AND DISCUSSION

The neural network has been tested with the testing samples. This will give us a sense of how well the network will do when applied to data from the real world. Table 5 shows the performance CVNN based classifier in term of classification time only, while the classification accuracy was not affected by proposed activation function. As seen from Table 5, the time value for CVNN with the proposed activation function is less than that of traditional activation function using the same training and testing data.

In the first model, the neural network works reliably when using the proposed complex activation functions.

Less errors are found in the outputs, with respect to the low iteration numbers, while in the second model, the network's performance is dropping when using traditional sigmoid activation function, because of high iteration numbers. This indicates how slowly a neuron adjusts its weight and bias values according to the error.

Tests by using fully complex activation functions, reduces the testing time by 54% compared to the choice of using the traditional sigmoid activation function, this improves the testing time of the network and prevent the network from starting oscillation as shown in Table 5.

In our simulation example, for comparison, another test has been performed for both models to investigate the effect of the reducibility on the model's convergence. The results of this test for the same sum of squared error (SSE) are shown in Table 6 (Irreducibility) and in Table 7 (Reducibility). Irreducibility results in Table 6, show that the second model has better ability of quick learning and global convergence than the first model. Table 7 shows that for reducibility, the iteration number of the second model decreased 56 % compared with the measured iteration number of the first model which decreased by 70 %.

The increasing of the training rate of the CVNN for reducibility is due to the tightly initial distribution of complex random weights and complex activation functions which tend to slow convergence and improves stability of CVNN.

Conclusion

In this paper, we have shown the efficiency of a complex valued network based on the study of the history of artificial neural networks, and the simulation of the CVNNs is discussed.

A live example illustrates using CVNN neural network as a classifier to identify the sex of crabs from physical dimensions based on a fully complex back propagation neural network. The simulation results of this paper shows acceptable results for the reducible and not the irreducible performance. The performance of a fully complex back propagation neural network has been substantially improved by the proposed approach.

As previously disccused using the subordination relation, we defined and studied the reducibility of 3-layers CVNN with two different activation functions which satisfied $F(z) \prec G(z)$.

This idea leads to N-layers CVNN under the condition:

$$F_1(z) \prec F_2(z) \prec ... \prec F_{N-2} \prec G(z),$$

In this case, we used the subordination relation to get the sandwich assertion. Further, the variety of using subordinate activation functions in one CVNN does not change the reducibility of the network. This leads to the questions: Do Theorem 1 hold for non subordinate activation functions? More specifically, is there another relation on F and G such that Theorem 1 satisfies?

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their comments which helped to improve the presentation of the paper.

REFERENCES

Ganesh A, Balasubramanian G (2009). Novel Complex Valued Neural Networks. Int. J. Comput. Appl. Math., 4: 165-171.

Georgin GM, Koutsougeras C (2002). Complex domain backpropagation. Circuits and Systems II: Analog and Digital Signal Processing. IEEE Trans., 39: 330-334.

Goodman A (1983). Univalent Functions. Mariner Publ. Co., Tampa, Fl, Vol. I.

Haykin S (2008). Adaptive filter theory: Pearson Education India.

Ibrahim RW, Darus M (2008). Subordination and superordination for univalent solutions for fractional differential equations. J. Math. Anal. Appl., 345: 871-879.

Kim T, Adali T (2002). Fully complex multi-layer perceptron network for nonlinear signal processing. J. VLSI Sig. Process., 32: 29-43.

Miller SS, Mocanu P (2000). Differential subordinations: Theory and applications: CRC.

Nitta T (2004a). Orthogonality of decision boundaries in complex-valued neural networks. Neural Comput., 16: 73-97.

Nitta T (2004b). Reducibility of the Complex-valued Neural Network. Neural Inf. Process.-Lett. Rev., 2: 53-56.

Sivanandam S (2006) Introduction to neural networks using MATLAB 6.0: Tata McGraw-Hill.

Subramaniam T, Jalab HA, Taqa AY (2010). Overview of textual anti-spam filtering techniques.

Sussmann HJ (1992). Uniqueness of the weights for minimal feedforward nets with a given input-output map. Neural Netw., 5: 589-593.

Taqa AY, Jalab HA (2010). Increasing the reliability of skin detectors. Sci. Res. Essays, 5: 2480-2490.