Journal of Signal and Information Processing, 2012, 3, 192-197
doi:10.4236/jsip.2012.32026 Published Online May 2012 (http://www.SciRP.org/journal/jsip)

Scientific Research

Complex Valued Recurrent Neural Network: From Architecture to Training

Alexey Minin[1,2], Alois Knoll[1], Hans-Georg Zimmermann[2]

[1]Technische Universität München—Robotics and Embedded Systems, München, Germany; [2]Siemens AG, Corporate Technology, München, Germany.
Email: alexey.minin@gmail.com

Received December 7th, 2011; revised February 11th, 2012; accepted March 18th, 2012

ABSTRACT

Recurrent Neural Networks were invented a long time ago, and dozens of different architectures have been published. In this paper we generalize recurrent architectures to a state space model, and we also generalize the numbers the network can process to the complex domain. We show how to train the recurrent network in the complex valued case, and we present the theorems and procedures to make the training stable. We also show that the complex valued recurrent neural network is a generalization of the real valued counterpart and that it has specific advantages over the latter. We conclude the paper with a discussion of possible applications and scenarios for using these networks.

Keywords: Complex Valued Neural Networks; Complex Valued System Identification; Recurrent Neural Networks; Complex Valued Recurrent Neural Networks

1. Introduction

Current paper aims to give the complete guidance from the state space models with complex parameters to the complex valued recurrent neural network of a special type. This paper is unique in translating the models suggested by Zimmermann in [1] to the complex valued case. Moreover one can see unique approach for managing the problems with transition functions which arise in complex-valued case, new approach for treating the error function which gives the unique advantages for the complex-valued neural networks. A lot of research in the area of complex valued recurrent neural networks is currently ongoing. One can find the works of Mandic [2,3], Adali [4] and Dongpo [5]. Mandic and Adali pointed out the advantages of using the complex valued neural networks in many papers. This paper will supply the neural network community with new architecture which shows better results in its complex-valued case.

We start the paper with the description on the state space models and then proceed with the very detailed explanations regarding the complex valued neural networks. We discuss complex valued system identification, error function properties in its complex valued case, complex valued back-propagation and break points with transition functions. Paper ends up with the small discussion on applications and advantages which arise form the complex valued case of the considered architecture.

State space techniques may be used to model recurrent dynamical systems. There are two principle ways of modeling dynamical systems: 1) use a feed-forward neural network and use delayed inputs or 2) use a recurrent architecture and model the dynamics itself. The first approach is based on Takens theorem [6] that a dynamical system or the attractor of the dynamical system can be reconstructed by a set of previous values of the realizations of the dynamical system (expectations). This is true for chaotic systems, but in real world applications feed forward networks cannot be used for forecasting the states of dynamical systems. Therefore, recurrent architectures are the only sensible way of forecasting dynamical systems, *i.e.* to represent the dynamics in the recurrent connection weights. This approach was first suggested by Elman [7] and later extended by Zimmermann [8]. As an example for this paper we will consider the so called open-system for which we will build a state space model based on the recurrent complex valued neural network. Such an open-system (open means that the system is driven not only by its internal state changes but also by external stimuli) is given as follows:

$$\begin{cases} s_{t+1} = f\left(s_t, u_t\right) \\ y_t = g\left(s_t\right) \end{cases} \tag{1}$$

Here, the states of the system ($S_{t=1\cdots n}$) depend on the previous states as well as some system input u_t through some non-linear function f. The output of the system depends on the current state of the system mapped through

another non-linear function g. A graphical representation is in **Figure 1**.

In the rest of this paper, we will use networks described by Equation (1) and **Figure 1**. In order to generalize the approach, we now assume that the dynamic system's behavior is described by complex numbers, which means that $S_t, u_t, S_{t+1}, y_t \in C$ and functions $f, g : C \to C$ are defined on the domain of complex numbers.

2. Complex Valued Neural Network

The Complex Valued Recurrent Neural Network (further CVRNN) is a straight forward generalization of the real-valued RNN. The algorithms which are used for CVRNNs can be also used for RNNs without loss of generality. To describe the CVRNN we start with a feed-forward path, and then we will discuss the error back-propagation algorithm (further CVEBP) and the training of such architectures.

2.1. Architecture Description and Feed Forward Path

The system represented by **Figure 1** and Equation (1) can be realized as follows (as suggested by Zimmermann [9]): consider a set of 3-layer-feed-forward networks (further FFNN), whose hidden layers are connected to each other. This connection represents the evolution of the corresponding dynamical system inside the RNN. The structure of this type of network is shown in **Figure 2**.

The dynamical system develops based on 1) internal evolution of the system state governed by the matrix $\boldsymbol{A}$ and the activation function. Matrices $\boldsymbol{B}$, $\boldsymbol{C}$ convert the external stimulus to the state (in the sense of data compression) and produce the output from the state (state decompression). Therefore, one can write the following system of equations, which describe the system in **Figure 2**.

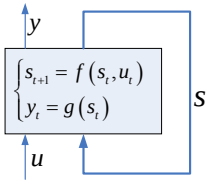

Figure 1. Dynamical system representation.

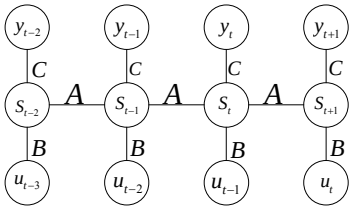

Figure 2. Recurrent neural network.

$$\begin{cases} s_{t+1} = \tanh\left(\boldsymbol{A}s_t + \boldsymbol{B}u_t\right) \\ y_t = \boldsymbol{C}s_t \end{cases} \tag{2}$$

where we have selected $\tanh(\cdot)$ as an activation function $f(\cdot)$, which performs the non-linear transformation of the state. Thus, all temporal relations of the dynamical system are represented in the matrix $\boldsymbol{A}$ (to be learned during training), the compression, and the decompression ability represented by the matrixes $\boldsymbol{B}$, $\boldsymbol{C}$ respectively (note that all elements of matrixes $\boldsymbol{A}_{ij}, \boldsymbol{B}_{ij}, \boldsymbol{C}_{ij} \in \boldsymbol{C}$ are complex numbers).

One word about the weights matrices: the matrices between the layers are always the same (suggested as the "shared weights concept" in [8]). It is exactly this property that makes this network recurrent. Next, we will discuss the back propagation for the shared weights concept.

Summarizing, the RNN has actuation inputs $u_t \in \boldsymbol{C}$; it has observable outputs $y_t \in \boldsymbol{C}$, it has states $s_t \in \boldsymbol{C}$, which evolve under the regime of the matrix $\boldsymbol{A}$, and it has a non-linear activation function $f(\cdot)$.

2.2. Error Back Propagation for the CVRNN

The first variant of Complex Valued Error Back Propagation was described by Haykin in [7]. First, we have to define the error function. Since, in the complex valued case there are no "greater/less than" relations, the output of the error function must be a real number in order to make it possible to evaluate the training result and to guide it into the direction of an error reduction. The procedure of the whole network training is as follows: find the network parameters, which are those weights that produce the minimum of the error function:

$$E(w)_t = \left(f(u_t, w) - \hat{y}_t\right) \to \min_w \tag{3}$$

where w are weights, f is the activation function, u are the complex valued inputs of the system, and $\hat{y}_t$ are observables.

One class of functions, which produces real-valued output from complex arguments, is the following:

$$E(w) = \left(y(w, u_t) - \hat{y}_t\right) \times \overline{\left(y(w, u_t) - \hat{y}_t\right)} \in R \tag{4}$$

where the over bar denotes complex value conjugate $\overline{(a+ib)} = a - ib$.

This current error function is not analytic, *i.e.*, the derivative $\mathrm{d}E/\mathrm{d}w$ is not defined over the entire range of input values. Therefore, back propagation cannot be applied.

The requirement for the analyticity of any function E is given by the Cauchy-Riemann conditions:

$$\frac{\partial u(a,b)}{\partial a} = \frac{\partial v(a,b)}{\partial b} \quad \text{and} \quad \frac{\partial u(a,b)}{\partial b} = -\frac{\partial v(a,b)}{\partial a} \tag{5}$$

where the function E is described with the following equation:

$$E(z)=u(a,b)+iv(a,b) \tag{6}$$

where $u(\cdot),v(\cdot)$ are some real differentiable functions of two real variables.

The requirement for the error function to produce real output means that $v(\cdot)=0$.

$$E(z)=u(a,b)+i\underbrace{v(a,b)}_{=0} \tag{7}$$

If we want $v(\cdot)=0$, we have to take an error function similar to (4) since our error function $E(z)=u(a,b)$, the optimality conditions are given by:

$$\begin{cases}\dfrac{\partial u(a,b)}{\partial a}=0\\ \dfrac{\partial u(a,b)}{\partial b}=0\end{cases} \tag{8}$$

The function $u(\cdot)$ makes a mapping of the following type: $R^2\to R$ instead of $C\to R$. In order to calculate the derivative of the function $u(\cdot)$ one should use the so called Wirtinger derivative (discussed, e.g., in Brandwood [10]): the Wirtinger derivative with respect to z and $\overline{z}$ can be calculated in the following way:

$$\begin{cases}\dfrac{\partial E}{\partial z}\triangleq\dfrac{1}{2}\left(\dfrac{\partial E}{\partial a}-i\dfrac{\partial E}{\partial b}\right), z=a+ib\\ \dfrac{\partial E}{\partial \overline{z}}\triangleq\dfrac{1}{2}\left(\dfrac{\partial E}{\partial a}+i\dfrac{\partial E}{\partial b}\right), \overline{z}=a-ib\end{cases} \tag{9}$$

For the real functions of complex variables $\partial E/\partial\overline{z}\neq 0$ therefore the minimization of the error function can be done in both directions z or $\overline{z}$.

Now we have the derivatives of the error function defined in the Wirtinger sense.

Note that this error function "minimizes" the complex number, which in the Euler notation (see Equation (10)) would mean, that it minimizes both amplitude and phase of the complex number, which is in our case $\left(y(w,u_t)-\hat{y}_t\right)$:

$$y=\overbrace{\mathrm{Re}(y)}^{a}+i\overbrace{\mathrm{Im}(y)}^{b}=\underbrace{\sqrt{a^2+b^2}}_{r}e^{\overbrace{i\cdot\mathrm{arctg}\left(\frac{b}{a}\right)}^{\phi}}=re^{i\cdot\phi} \tag{10}$$

This error function has very unique and desirable properties. Let us describe these properties more in detail. We rewrite (4) into Euler notation:

$$E=r^2+\hat{r}^2-r\hat{r}\underbrace{\left(e^{i(\phi-\hat{\phi})}+e^{i(\hat{\phi}-\phi)}\right)}_{2\cos(\Delta\phi)} \tag{11}$$

The discriminant of (11) is negative and only can be equal to zero that the equation has 1 root:

$$D=\sqrt{4\hat{r}^2\cos^2(\Delta\phi)-4\hat{r}^2}=2\hat{r}\sqrt{-\sin^2(\Delta\phi)}=0$$
$$\Rightarrow\Delta\phi=0 \text{ then } r=\hat{r}$$

We can also rewrite (4) in the following way:

$$E(y,\hat{y})=a^2+b^2+\hat{a}^2+\hat{b}^2-2a\hat{a}-2b\hat{b}=\cdots$$
$$\cdots=(a-\hat{a})^2+\left(b-\hat{b}\right)^2\xrightarrow[b\to\hat{b}]{a\to\hat{a}}\min \tag{12}$$

One can see that error function (4) minimizes both real and imaginary parts of the complex number.

After defining a suitable error-function, we can now start with the CVEBP description. The procedure for CVEBP is shown in **Figure 3**. It follows the description in [9] or the RNN. The "ladder" algorithm allows a local and efficient computation of the recurrent network partial derivatives of the error with respect to the weights. The advantage of the algorithm shown in **Figure 3** is that it intelligently unites the equations, the architecture and the locality of the CVEBP.

In **Figure 3**, one can see the CVEBP which is done for the shared matrix $\boldsymbol{A}$ and for the case when all NN parameters are complex numbers.

2.3. Weights Update Rule for the CVRNN

In order to find the training rule for the weights update, we introduce the Taylor expansion of the error function:

$$E(w+\Delta w)=E(w)+g^T\Delta w+\frac{1}{2}\Delta w^T G\Delta w \tag{13}$$

where (one can note that Δw has to be equal to the $\overline{g}$

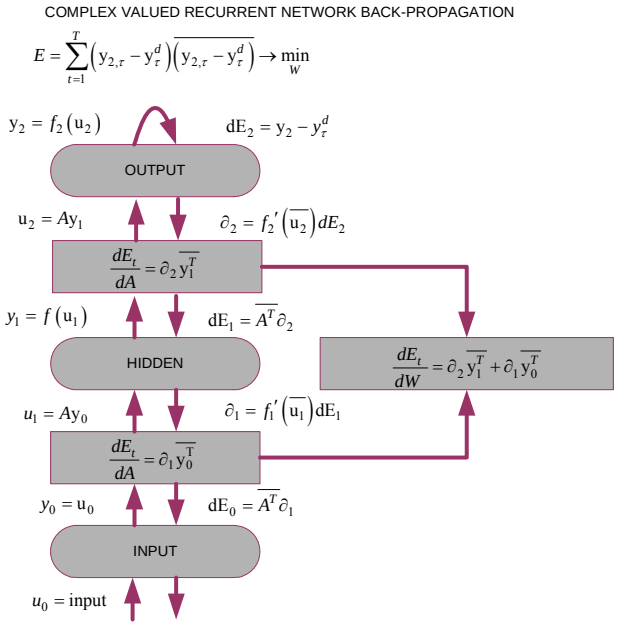

Figure 3. Complex valued error back propagation for the derivatives with respect to $\overline{z}$.

that Taylor expansion exist):

$$\frac{\partial E}{\partial \overline{w}} = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - y_t^d\right)\frac{\partial \overline{y}}{\partial \overline{w}} =: \overline{g} \tag{14}$$

Following Johnson in his paper [11], two useful theorems to calculate the derivatives can be applied.

Theorem 1. If the function $f(z,\overline{z})$ is real-valued and analytic with respect to z or $\overline{z}$, all stationary points can be found by setting the derivatives in Equation (9) with respect to either z or $\overline{z}$ to zero.

Theorem 2. By treating z and $\overline{z}$ as independent variables, the quantity pointing in the direction of the maximum rate of change of $f(z,\overline{z})$ is $\nabla_{\overline{z}} f(z)$.

The proof of the theorems was demonstrated by Johnson in [11].

Following Karla in [12] and Adali in [13], if minimization goes in the direction of $\overline{z}$, then $\Delta g = -2\eta\left\|\nabla_z g\right\|^2$. Otherwise, if we minimize in the direction of z, it results in $\Delta g = -2\eta\,\mathrm{Re}\left\{\left\langle \nabla_{\overline{z}} g, \nabla_z g\right\rangle\right\}$ which need not necessarily be negative. This will lead us in the direction of a different minimization.

Following Theorem 2 and Equation (7), we consider $\frac{\partial E}{\partial w} = 0$. The Taylor expansion exists, since the derivatives are defined and we can obtain a training rule for the optimization of weights in the direction of $\overline{z}$:

$$\Delta w = -\eta \cdot \overline{g} \tag{15}$$

Notice that **Figure 3** is very similar to the real valued RNN, despite the conjugations instead of the transposes.

One should also note that this error function is universal because it optimizes both the real and the imaginary part of the complex number. It has a simple derivative, and it is a parabola, which means it has only one minimum and smooth bounds. A typical convergence of the error during the training is presented in **Figure 4**.

Note that this error function is a real value. **Figure 4** shows the modulus, *i.e.*, exactly the error function, the angle error is:

$$E_{\text{angle}} = \left|\arctan\left(\frac{\mathrm{Re}(y)}{\mathrm{Im}(y)}\right) - \arctan\left(\frac{\mathrm{Re}(\hat{y})}{\mathrm{Im}(\hat{y})}\right)\right| \tag{16}$$

After presenting the CVEBP and discussing the convergence of the error, we now discuss the final aspect of the CVRNN, which is the activation (or transition) function.

2.4. Activation Function in the Complex Valued Domain

It is well known that for real valued networks, one of the requirements for the activation function is to be continuous (ideally: bounded), and it should have at least one derivative defined for the whole search space.

Unfortunately, this is not the case for the complex valued functions due to the Liouville theorem [10]. Moreover, all transition functions which are not linear have an unlimited growth at their bounds (example: sine-function) or have singularity points an (example is the tanh-function, see **Figure 5** below).

Based on the following Theorem 3, we can make several remedies:

Liouville Theorem 3. If a complex analytic function is bounded and complex differentiable on the whole complex plain, it is constant.

This theorem has been proven in Remmert [14].

Remedy 1. Choose bounded functions which are only real valued but not complex differentiable:

$$\begin{aligned} &f(a+ib) = \tanh(a) + i\tanh(b) \\ &\text{or} \quad f\left(r\cdot e^{i\phi}\right) = \tanh(r)e^{i\phi} \end{aligned} \tag{17}$$

Remedy 2. Constrain the optimization procedure in order to stay in the area, where there are no singularities:

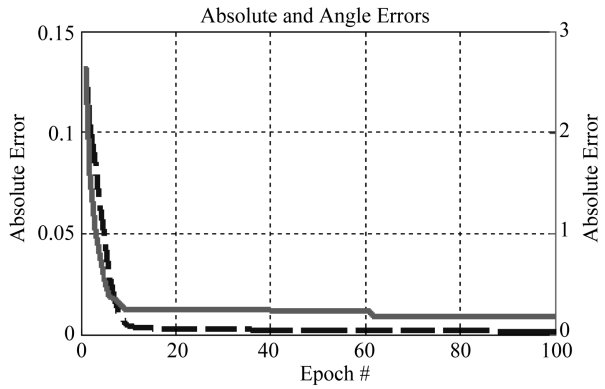

Figure 4. Error convergence for the absolute part of the error (dashed line) and for the phase of the error (solid line).

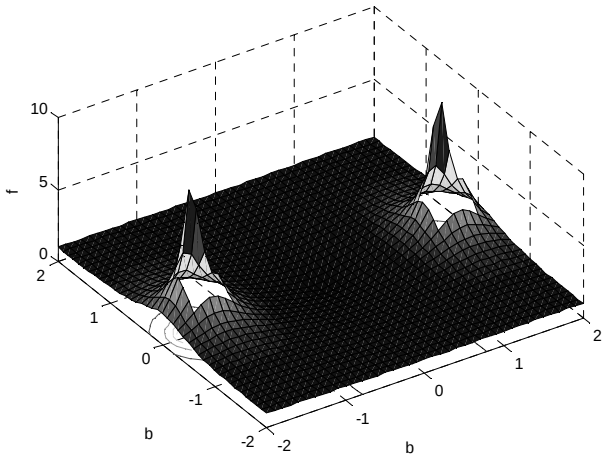

Figure 5. Absolute part behavior of the tanh function: it shows two singularities, which are periodic at π/2.

$$\begin{cases} \operatorname{logsig}(z) = re^{i\phi} = \operatorname{logsig}(z); r < 1 \\ \operatorname{logsig}'(z) = \operatorname{logsig}(z)\left(1 - \operatorname{logsig}(z)\right) \end{cases} \quad (18)$$

Remedy 3. All real analytic functions are differentiable in complex domain using the Wirtinger Calculus.

One can also try to substitute the problematic regions of the functions with different functions which do not have the problem in the following region (the problem is the presence of singularity point) following the (19) below:

$$\begin{cases} f_1(z) = \tanh(z), \\ z \in \mathbb{C} \setminus \left[-\frac{\pi}{2} - \varepsilon; -\frac{\pi}{2} + \varepsilon\right] \cup [-\varepsilon; \varepsilon] \\ f_2(z) = \tanh\left(\operatorname{Re}(z)\right) e^{i\phi(z)}, \\ z \in \left[-\frac{\pi}{2} - \varepsilon; -\frac{\pi}{2} + \varepsilon\right] \\ f_3(z) = \log\cosh(z), z \in [-\varepsilon; \varepsilon] \\ f_1'(z) = f_2'(z); z \in \left[-\frac{\pi}{2} - \varepsilon; -\frac{\pi}{2} + \varepsilon\right] \\ f_1'(z) = f_3'(z); z \in [-\varepsilon; \varepsilon] \end{cases} \quad (19)$$

The result of such experiment is shown in **Figure 6** below.

Typical use of the CVRNN with the activation function $\tanh(\cdot)$ will be possible with the non-linear function, as long as the weights are initialized with small numbers and the error minimization goes in the correct direction (*i.e.*, the error decreases and steps of the weight update becomes smaller as training time increases). Also, the weights do not go above 1, which means they do not approach the singularities of the function.

3. Summary and Outlook

In this paper we discussed several aspects of CVRNN.

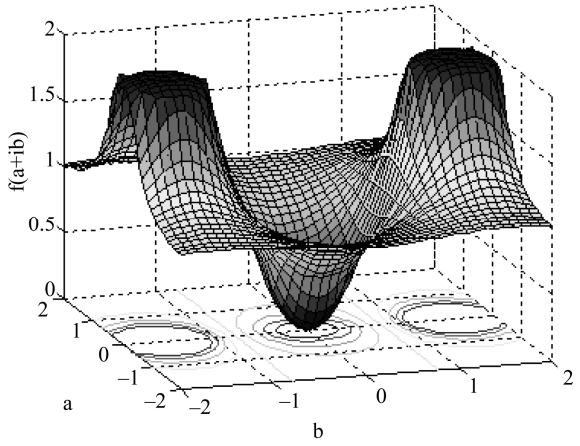

Figure 6. The transition function for the substitute functions.

We showed the architecture of the CVRNN, discussed the feed forward operation as well as the back-propagation CVEBP and the weights update rules. We discussed problems with the activation and error functions and showed how to overcome these problems.

There are many advantages of using CVRNN: continuous time modeling, modeling of electrical devices and energy grids, robust time series prediction, physical models of the brain, etc. Future work will focus on applications and evaluation of CVRNNs.

REFERENCES

[1] H. G. Zimmermann and R. Neuneier, "Modeling Dynamical Systems by Recurrent Neural Networks, Data Mining II," *Second International Conference on Data Mining*, Cambridge, 5-7 July 2000, pp. 557-566.

[2] S.-L. Gohand and D. Mandic, "A Complex-Valued RTRL Algorithm for Recurrent Neural Networks," *Neural Computation*, Vol. 16, No. 12, 2006, pp. 2699-2713.

[3] P. Mandic, "Complex Valued Recurrent Neural Networks for Noncircular Complex Signals," *International Joint Conference on Neural Networks*, 14-19 June 2009, pp. 1987-1992. doi:10.1109/IJCNN.2009.5178960

[4] T. Adali and H. Li, "A Practical Formulation for Computation of Complex Gradients and Its Application to Maximum Likelihood ICA," 2007 *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, 2007, pp. II-633-II-636. doi:10.1109/ICASSP.2007.366315

[5] D. P. Xu, H. S. Zhang and L. J. Liu, "Convergence Analysis of Three Classes of Split-Complex Gradient Algorithms for Complex-Valued Recurrent Neural Networks," *Neural Computation*, Vol. 22, No. 20, 2010, pp. 2655-2677. doi:10.1162/NECO_a_00021

[6] F. Takens, "Detecting Strange Attractors in Turbulence, Dynamical Systems and Turbulence," *Lecture Notes in Mathematics*, Vol. 898, Springer-Verlag, New York, 1981, pp. 366-381.

[7] H. Leung and S. Haykin, "The Complex Back Propagation," *IEEE Transactions on Signal Processing*, Vol. 39, No. 9, 1991, pp. 2101-2104. doi:10.1109/78.134446

[8] G. Zimmermann, A. Minin and V. Kusherbaeva, "Historical Consistent Complex Valued Recurrent Neural Network," *Lecture Notes in Computer Science*, Part 1, Vol. 6791, 2011, pp. 185-192. doi:10.1007/978-3-642-21735-7_23

[9] H.-G. Zimmermann, A. Minin and V. Kusherbaeva, "Comparison of the Complex Valued and Real Valued Neural Networks Trained with Gradient Descent and Random Search Algorithms," 19*th European Symposium on Artificial Neural Networks*, Bruges, 27-29 April 2011, pp. 216-222.

[10] D. H. Brandwood, "A Complex Gradient Operator and Its Application in Adaptive Array Theory," *IEEE Proceedings*, *F*: *Communications*, *Radar and Signal Processing*, Vol. 130, No. 1, 1983, p. 1116.

[11] D. Johnson, "Optimization Theory," Optimization Theory Page from the Connexions Project. http://cnx.org/content/m11240/latest/

[12] P. Kalra, A. Gangal and D. Chauhan, "Performance Evaluation of Complex Valued Neural Networks Using Various Error Functions," *World Academy of Science, Engineering and Technology*, Vol. 29, 2007, pp. 27-32.

[13] H. L. Li and T. Adali, "A Class of Complex ICA Algorithms Based on the Kurtosis Cost Function," *IEEE Transactions on Neural Networks*, Vol. 19, No. 3, 2008, pp. 408-420. doi:10.1109/TNN.2007.908636

[14] R. Remmert, "Theory of Complex Functions," Springer, New York, 1991. doi:10.1007/978-1-4612-0939-3

