Journal of Computer and Communications, **2014, 2, 27-32**
Published Online January 2014 (http://www.scirp.org/journal/jcc)
http://dx.doi.org/10.4236/jcc.2014.21005

Learning Dynamics of the Complex-Valued Neural Network in the Neighborhood of Singular Points

Tohru Nitta

National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan.
Email: tohru-nitta@aist.go.jp

Received December 4th, 2013; revised December 28th, 2013; accepted January 4th, 2014

ABSTRACT

In this paper, the singularity and its effect on learning dynamics in the complex-valued neural network are elucidated. It has learned that the linear combination structure in the updating rule of the complex-valued neural network increases the speed of moving away from the singular points, and the complex-valued neural network cannot be easily influenced by the singular points, whereas the learning of the usual real-valued neural network can be attracted in the neighborhood of singular points, which causes a standstill in learning. Simulation results on the learning dynamics of the three-layered real-valued and complex-valued neural networks in the neighborhood of singularities support the analytical results.

KEYWORDS

Complex-Valued Neural Network; Complex Number; Learning; Singular Point

1. Introduction

Complex-valued neural networks have been applied in various fields dealing with complex numbers or two-dimensional data such as signal processing and image processing [1,2]. The complex-valued neural network can represent more information than the real-valued neural network because the inputs, the weights, the threshold values, and the outputs are all complex numbers, and the complex-valued neural network has some inherent properties such as the ability to transform geometric figures [3-5] and the orthogonal decision boundary [6,7].

In the applications of the multi-layered type real-valued neural networks, the error back-propagation learning algorithm (called here, *Real-BP* [8]) has often been used. Naturally, the complex-valued version of the Real-BP (called here, *Complex-BP*) can be considered, and was actually proposed by several researchers independently in the early 1990's [3-5,9-11]. This algorithm enables the network to learn complex-valued patterns naturally.

On one hand, the researches on the singularity of the learning machines with a hierarchical structure have progressed in the past several years [12-15]. It has turned out that the singularity has a negative effect on learning dynamics in the learning machines such as real-valued neural networks and gaussian mixture models. Here, a singular point is a point on which the derivatives of an error function are all zero, that is, it is a critical point and can be a local minimum, a local maximum or a saddle point.

In this paper, the singularity and its effect on learning dynamics in the complex-valued neural network are elucidated. As a result, we find that the linear combination structure in the updating rule of the complex-valued neural network increases the speed of moving away from the singular points; the complex-valued neural network cannot be easily influenced by the singular points, whereas the learning of the usual real-valued neural network can be attracted in the neighborhood of singular points, which causes a standstill in learning. Simulation results on the learning dynamics of the three-layered real-valued and complex-valued neural networks in the neighborhood of singularities support the analytical results. It should be noted here that it has been reported that the learning speed of the Complex-BP is two or three times faster than that of the Real-BP on average via computer simulations [5,16]. This is due to the learning structure of the complex-valued neural networks described above.

2. Problem on the Singularity

Recently, it has turned out that the singularity has a negative effect on learning dynamics in the real-valued neural networks [12,13,15]. That is, the hierarchical structure or a symmetric property on exchange of weights of the the real-valued neural networks have singular points. For example, if a weight *v* between a hidden neuron and an output neuron is eaual to zero, then no value of the weight vector *w* between the hidden neuron and the input neurons affects the output value of the real-valued neural network. Then, the weight *v* is called an unidentifiable parameter, which is a kind of singular point. It has been proved that singular points affect the learning dynamics of learning models, and that they can cause a standstill in learning.

3. The Complex-Valued Neural Network

This section describes the complex-valued neural network used in the analysis. First, we will consider the following complex-valued neuron. 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: $U_n = \sum_m W_{nm} X_m + V_n$, where W_{mm} is the (complex-valued) weight connecting the complex-valued neurons *n* and *m*, X_m is the (complex-valued) input signal from the complex-valued neuron *m*, and V_n is the (complex-valued) threshold value of the complex-valued neuron *n*. To obtain the (complex-valued) output signal, convert the net input U_n into its real and imaginary parts as follows: $U_n = x + \mathrm{i}y = z$, where i denotes $\sqrt{-1}$. The (complex-valued) output signal is defined to be

$$\varphi_C(z) = \varphi(x) + \mathrm{i}\varphi(y), \tag{1}$$

where

$$\varphi(u) \overset{\text{def}}{=} \tanh(u)$$
$$= (\exp(u) - \exp(-u)) / (\exp(u) + \exp(-u)),$$

$u \in \boldsymbol{R}$ ($\boldsymbol{R}$ denotes the set of real numbers) and is called *hyperbolic tangent*. Note that $-1 < \mathrm{Re}[\varphi_C], \mathrm{Im}[\varphi_C] < 1$. Note also that $\varphi_C(z)$ is not holomorphic as a complex function because the Cauchy-Riemann equations do not hold:

$$\partial\varphi_C(z)/\partial x + \mathrm{i}\partial\varphi_C(z)/\partial y$$
$$= \left(1 - \varphi_C(x)^2\right) + \mathrm{i}\left(1 - \varphi_C(y)^2\right) \neq 0$$

where $z = x + \mathrm{i}y$.

A complex-valued neural network consists of such complex-valued neurons described above. The network used in the analysis will have 3 layers: *L*-*H*-1 network. The activation function ψ_C of the output neuron is linear, that is, $\psi_C(z) = z$ for any $z \in \boldsymbol{C}$ where $\boldsymbol{C}$ denotes the set of complex numbers. For any input pattern $\boldsymbol{x} = (x_1, \cdots, x_L)^{\mathrm{T}} \in \boldsymbol{C}^L$ to the complex-valued neural network where $x_k \in \boldsymbol{C}$ is the input signal to the input neuron $k (1 \le k \le L)$ and T denotes transposition, the output value of the output neuron is defined to be

$$f^{(H)}\left(\boldsymbol{x}; \boldsymbol{\theta}^{(H)}\right) = \sum_{j=1}^{H} v_j \varphi_C\left(\tilde{\boldsymbol{w}}_j^T \tilde{\boldsymbol{x}}\right) + v_0 \in \boldsymbol{C}, \tag{2}$$

where $\tilde{\boldsymbol{w}}_j = \left(w_{j0} \quad \boldsymbol{w}_j^{\mathrm{T}}\right)^{\mathrm{T}} \in \boldsymbol{C}^{L+1}$, $w_{j0} \in \boldsymbol{C}$ is the threshold of the hidden neuron j, $\boldsymbol{w}_j = \left(w_{j1}, \cdots, w_{jL}\right)^{\mathrm{T}} \in \boldsymbol{C}^L$ is the weight vector of the hidden neuron j ($w_{jk} \in \boldsymbol{C}$ is the weight between the input neuron k and the hidden neuron j) $(1 \le j \le H)$, $\tilde{\boldsymbol{x}} = \left(1 \quad \boldsymbol{x}^{\mathrm{T}}\right)^{\mathrm{T}} \in \boldsymbol{C}^{L+1}$, $v_j \in \boldsymbol{C}$ is the weight between the hidden neuron j and the output neuron $(1 \le j \le H)$, $v_0 \in \boldsymbol{C}$ is the threshold of the output neuron, and $\boldsymbol{\theta}^{(H)} = \left(v_0, v_1, \cdots, v_H, \tilde{\boldsymbol{w}}_1^{\mathrm{T}}, \cdots, \tilde{\boldsymbol{w}}_H^{\mathrm{T}}\right)^{\mathrm{T}}$ which summarizes all the parameters in one large vector.

Given N complex-valued training data $\left\{\left(x^{(p)}, y^{(p)}\right) \in \boldsymbol{C}^L \times \boldsymbol{C} \,\middle|\, p = 1, \cdots, N\right\}$, we use a complex-valued neural network to realize the relation expressed by the data. The objective of the training is to find the parameters that minimize the error function defined by

$$E_H\left(\boldsymbol{\theta}^{(H)}\right) = \sum_{p=1}^{N} l\left(y^{(p)}, f^{(H)}\left(\boldsymbol{x}^{(p)}; \boldsymbol{\theta}^{(H)}\right)\right) \in \boldsymbol{R}, \tag{3}$$

where $l(y,z): \boldsymbol{C} \times \boldsymbol{C} \to \boldsymbol{R}$ is a loss function such that $l(y,z) \ge 0$ and the equality holds if and only if $y = z$. Note that l is not holomorphic as a complex function because it takes a real value. To the author's knowledge, all of the multi-layered complex-valued neural networks proposed so far (for example [3-5,9-11]) employ the mean square error $l(y,z) = (1/2)|y - z|^2$ which takes a real value.

4. Analytical Dynamics

This section reveals the behavior of the three-layered complex-valued neural network in the neighborhood of singular points, compared with that of the three-layered real-valued neural network via theoretical analysis.

Consider a 1-1-1 complex-valued neural network described in Section 3 for the sake of simplicity. For any input signal $x_1 \in \boldsymbol{C}$ to the complex-valued neural network, the output value of the output neuron is defined to be

$$f^{(1)}\left(x_1; \boldsymbol{\theta}^{(1)}\right) = v_1 \varphi_C\left(\tilde{\boldsymbol{w}}_1^{\mathrm{T}} \tilde{\boldsymbol{x}}\right) + v_0 \in \boldsymbol{C} \tag{4}$$

where $\tilde{\boldsymbol{w}}_1 = \begin{pmatrix} w_{10} & w_{11} \end{pmatrix}^{\mathrm{T}} \in \boldsymbol{C}^2$, $w_{10} \in \boldsymbol{C}$ is the threshold of the hidden neuron, $w_{11} \in \boldsymbol{C}$ is the weight between the input neuron and the hidden neuron, $\tilde{\boldsymbol{x}} = \begin{pmatrix} 1 & x_1 \end{pmatrix}^{\mathrm{T}} \in \boldsymbol{C}^2$, $v_1 \in \boldsymbol{C}$ is the weight between the hidden neuron and the output neuron, $v_0 \in C$ is the threshold of the output neuron, and $\boldsymbol{\theta}^{(1)} = \left(v_0, v_1, \tilde{\boldsymbol{w}}_1^{\mathrm{T}}\right)^{\mathrm{T}}$. The number of learnable parameters (weights and thresholds) is 8 where a complex-valued parameter is counted as two because it consists of a real part and an imaginary part. The loss function is defined as

$$l\left(y, x_1; \boldsymbol{\theta}^{(1)}\right) = \frac{1}{2}\left|y - f^{(1)}\left(x_1; \boldsymbol{\theta}^{(1)}\right)\right|^2, \tag{5}$$

where $y \in \boldsymbol{C}$ is the training signal for the output neuron.

In the case of the standard gradient learning method, the average learning dynamics is given as follows:

$$\dot{v}_1(t) = -\varepsilon\left[\frac{\partial l\left(y, \hat{x}_1; \boldsymbol{\theta}^{(1)}\right)}{\partial v_1^R} + \mathrm{i}\frac{\partial l\left(y, \hat{x}_1; \boldsymbol{\theta}^{(1)}\right)}{\partial v_1^I}\right], \tag{6}$$

$$\dot{w}_{11}(t) = -\varepsilon\left[\frac{\partial l\left(y, \hat{x}_1; \boldsymbol{\theta}^{(1)}\right)}{\partial w_{11}^R} + \mathrm{i}\frac{\partial l\left(y, \hat{x}_1; \boldsymbol{\theta}^{(1)}\right)}{\partial w_{11}^I}\right], \tag{7}$$

$$\dot{v}_0(t) = -\varepsilon\left[\frac{\partial l\left(y, \hat{x}_1; \boldsymbol{\theta}^{(1)}\right)}{\partial v_0^R} + \mathrm{i}\frac{\partial l\left(y, \hat{x}_1; \boldsymbol{\theta}^{(1)}\right)}{\partial v_0^I}\right], \tag{8}$$

$$\dot{w}_{10}(t) = -\varepsilon\left[\frac{\partial l\left(y, \hat{x}_1; \boldsymbol{\theta}^{(1)}\right)}{\partial w_{10}^R} + \mathrm{i}\frac{\partial l\left(y, \hat{x}_1; \boldsymbol{\theta}^{(1)}\right)}{\partial w_{10}^I}\right], \tag{9}$$

where $\hat{x}_1 \overset{\mathrm{def}}{=} E[x_1]$, $z^R = \mathrm{Re}[z]$ and $z^I = \mathrm{Im}[z]$ for a learnable parameter z.

Then, we investigate the behavior of the weight v_1 between the hidden neuron and the output neuron in the neighborhood of the singularity $v_1 = 0$. Letting $v_1 = 0$, from Equation (6), we can easily obtain

$$\dot{v}_1(t) = \varepsilon\left(y - v_0\right) \cdot \overline{\varphi_C\left(\tilde{\boldsymbol{w}}_1^{\mathrm{T}} \hat{\tilde{\boldsymbol{x}}}\right)}, \tag{10}$$

where $\hat{\tilde{\boldsymbol{x}}} = \begin{pmatrix} 1 & \hat{x}_1 \end{pmatrix}^{\mathrm{T}}$.

Next, consider a 2-1-2 real-valued neural network. For any input signal $\boldsymbol{x} = \left(x_1, x_2\right)^{\mathrm{T}} \in \boldsymbol{R}^2$ to the real-valued neural network, the output value of the real-valued neural network is defined to be

$$g(\boldsymbol{x}; \boldsymbol{\theta}) = \left(v_1 \varphi\left(\tilde{\boldsymbol{w}}^{\mathrm{T}} \tilde{\boldsymbol{x}}\right) + v_{10}, v_2 \varphi\left(\tilde{\boldsymbol{w}}^{\mathrm{T}} \tilde{\boldsymbol{x}}\right) + v_{20}\right)^{\mathrm{T}} \in \boldsymbol{R}^2, \tag{11}$$

where $\tilde{\boldsymbol{w}} = \begin{pmatrix} w_{10} & \boldsymbol{w}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}} \in \boldsymbol{R}^3$, $w_{10} \in \boldsymbol{R}$ is the threshold of the hidden neuron, $\boldsymbol{w} = \left(w_{11}, w_{12}\right)^{\mathrm{T}} \in \boldsymbol{R}^2$ is the weight vector between the input neurons and the hidden neuron ($w_{1j} \in \boldsymbol{R}$ is the weight between the input neuron j and the hidden neuron $(j = 1, 2)$), $\tilde{\boldsymbol{x}} = \begin{pmatrix} 1 & \boldsymbol{x}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}} \in \boldsymbol{R}^3$, $\boldsymbol{v} = \left(v_1, v_2\right)^{\mathrm{T}} \in \boldsymbol{R}^2$ is the weight vector between the hidden neuron and the output neurons ($v_l \in \boldsymbol{R}$ is the weight between the hidden neuron and the output neuron l $(l = 1, 2)$), $\boldsymbol{v}_0 = \left(v_{10}, v_{20}\right)^{\mathrm{T}} \in \boldsymbol{R}^2$ is the threshold vector of the output neurons ($v_{l0} \in \boldsymbol{R}$ is the threshold of the output neuron l $(l = 1, 2)$), $\varphi(u) \overset{\mathrm{def}}{=} \tanh(u), u \in \boldsymbol{R}$, and $\boldsymbol{\theta} = \left(\boldsymbol{v}_0^{\mathrm{T}}, \boldsymbol{v}^{\mathrm{T}}, \boldsymbol{w}^{\mathrm{T}}\right)^{\mathrm{T}}$. The number of learnable parameters (weights and thresholds) is 7, which is almost equal to that of the complex-valued neural network described above. Thus, the comparison of the learning dynamics using those neural networks is fair. The loss function is defined as

$$l(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{2}\left|\boldsymbol{y} - g(\boldsymbol{x}; \boldsymbol{\theta})\right|^2, \tag{12}$$

where $\boldsymbol{y} = \left(y_1, y_2\right)^{\mathrm{T}} \in \boldsymbol{R}^2$ is the training signal for the output neurons.

The average learning dynamics of the real-valued neural network using the standard gradient learning method is given as follows:

$$\dot{v}_1(t) = -\varepsilon \frac{\partial l(\boldsymbol{y}, \hat{\boldsymbol{x}}; \boldsymbol{\theta})}{\partial v_1}, \tag{13}$$

$$\dot{v}_2(t) = -\varepsilon \frac{\partial l(\boldsymbol{y}, \hat{\boldsymbol{x}}; \boldsymbol{\theta})}{\partial v_2}, \tag{14}$$

$$\dot{w}_{11}(t) = -\varepsilon \frac{\partial l(\boldsymbol{y}, \hat{\boldsymbol{x}}; \boldsymbol{\theta})}{\partial w_{11}}, \tag{15}$$

$$\dot{w}_{12}(t) = -\varepsilon \frac{\partial l(\boldsymbol{y}, \hat{\boldsymbol{x}}; \boldsymbol{\theta})}{\partial w_{12}}, \tag{16}$$

$$\dot{w}_{10}(t) = -\varepsilon \frac{\partial l(\boldsymbol{y}, \hat{\boldsymbol{x}}; \boldsymbol{\theta})}{\partial w_{10}}, \tag{17}$$

$$\dot{v}_{10}(t) = -\varepsilon \frac{\partial l(\boldsymbol{y}, \hat{\boldsymbol{x}}; \boldsymbol{\theta})}{\partial v_{10}} \tag{18}$$

$$\dot{v}_{20}(t) = -\varepsilon \frac{\partial l(\boldsymbol{y}, \hat{\boldsymbol{x}}; \boldsymbol{\theta})}{\partial v_{20}}, \tag{19}$$

where $\hat{\boldsymbol{x}} \overset{\mathrm{def}}{=} E[\boldsymbol{x}]$. Letting $\boldsymbol{v} = 0$, from Equations (13) and (14), we can easily obtain

$$\dot{v}_1(t) = \varepsilon\left(y_1 - v_{10}\right) \cdot \varphi\left(\tilde{\boldsymbol{w}}^{\mathrm{T}} \hat{\tilde{\boldsymbol{x}}}\right), \tag{20}$$

$$\dot{v}_2(t) = \varepsilon\left(y_2 - v_{20}\right) \cdot \varphi\left(\tilde{\boldsymbol{w}}^{\mathrm{T}} \hat{\tilde{\boldsymbol{x}}}\right), \tag{21}$$

where $\hat{\tilde{\boldsymbol{x}}} = \begin{pmatrix} 1 & \hat{\boldsymbol{x}}^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}$. This is the average dynamics of the weights $\boldsymbol{v} = \left(v_1, v_2\right)^{\mathrm{T}}$ between the hidden neuron and the output neurons in the neighborhood of the singularity $\boldsymbol{v} = 0$.

We compare below the average dynamics of the complex-valued neural network with that of the real-valued neural network in the neighborhood of singularity. Equation (10) can be rewritten as

$$\dot{v}_1^R(t) = \varepsilon\Big[\big(y^R - v_0^R\big)\cdot\varphi\big(w_{11}^R x_1^R - w_{11}^I x_1^I + w_{10}^R\big) + \big(y^I - v_0^I\big)\cdot\varphi\big(w_{11}^R x_1^I + w_{11}^I x_1^R + w_{10}^I\big)\Big], \tag{22}$$

$$\dot{v}_1^I(t) = \varepsilon\Big[-\big(y^R - v_0^R\big)\cdot\varphi\big(w_{11}^R x_1^I + w_{11}^I x_1^R + w_{10}^I\big) + \big(y^I - v_0^I\big)\cdot\varphi\big(w_{11}^R x_1^R - w_{11}^I x_1^I + w_{10}^R\big)\Big]. \tag{23}$$

As shown in Equations (22) and (23), $\dot{v}_1^R(t)$ and $\dot{v}_1^I(t)$ consist of two linear combinations of φ, respectively. The following are its explanatory equations.

$$\Delta\big(\text{Parameter of the Complex } NN\big) = a\varphi(u_1) + b\varphi(u_2), \tag{24}$$

$$\Delta\big(\text{Parameter of the Real } NN\big) = c\varphi(u_3). \tag{25}$$

For example, if $a = b = c$ and $u_1 = u_2 = u_3$, then Δ (Parameter of the Complex NN) = $2a\varphi(u_1) = 2\times\Delta$ (Parameter of the Real NN) holds. And also, Δ (Parameter of the Complex NN) cannot be equal to zero easily because $a\varphi(u_1)$ is not necessarily equal to zero even if one term on $b\varphi(u_2)$ is almost equal to zero. This structure causes a high possibility of $\dot{v}_1^R(t)$ and $\dot{v}_1^I(t)$ of taking larger values, compared with $\dot{v}_1(t)$ and $\dot{v}_2(t)$ (Equations (20) and (21)). Thus, we can assume that the speed of the complex-valued neural network of moving away from the singularity is faster than that of the real-valued neural network.

5. Simulations

We present below the simulation results on the learning dynamics of the three-layered complex-valued neural networks using the standard gradient learning method in the neighborhood of singularities, and compare them with those of the three-layered real-valued neural networks using the standard gradient learning method.

In the experiments, the three sets of (complex-valued) learning patterns shown in Table 1 were used, and the learning constant ε was 0.5. We chose the three-layered 1-1-1 complex-valued neural network where $\tilde{\boldsymbol{w}}_1 = 0$ and $v_1 = 0$ were singular points, and the three-layered 2-1-2 real-valued neural network where $\tilde{\boldsymbol{w}} = 0$ and $v = 0$ were singular points described in Section 4. The comparison using those neural networks is fair because the numbers of the parameters (weights and thresholds) are almost the same: the number of parameters for the 1-1-1 complex-valued network is 8, and that for the 2-1-2 real-valued neural network 7 where a complex-valued parameter $z = x + \mathrm{i}y$ is counted as two because it consists of a real part x and an imaginary part y. In the real-valued neural network, the real component of a complex number was input into the first input neuron, and the imaginary component was input into the second input neuron; the output from the first output neuron was interpreted to be the real component of a complex number, and the output from the second output neuron was interpreted to be the imaginary component. The initial values of the weights and the thresholds were set as shown in Table 2. Note that the initial values of the weights between the hidden layer and the output layer were set in the neighborhood of the singular points: $v_1 = 0.000001 + 0.000001\mathrm{i}$ in the complex-valued neural network, and $v_1 = 0.000001, v_2 = 0.000001$ in the real-valued neural network. The eight initial values (eight cases) were used for the weights w_{11} or (w_{11}, w_{12}) between the input layer and the hidden layer. We judged that learning finished, when the training error was equal to 0.0001, that is, $\left|y - f^{(1)}\left(x_1;\boldsymbol{\theta}^{(1)}\right)\right| = 0.0001$ in the case of the complex-valued neural network, and $\left|\boldsymbol{y} - g(\boldsymbol{x};\boldsymbol{\theta})\right| = 0.0001$ in the case of the real-valued neural network.

Table 1. **Learning patterns.**

(a) Learning pattern 1

Input pattern	Output pattern
1.0	i

(b) Learning pattern 2

Input pattern	Output pattern
i	−i

(c) Learning pattern 3

Input pattern	Output pattern
0.1 + 0.1i	0.5 + 0.5i

We investigated the learning speed (*i.e.*, learning cycles needed to converge) for each of the 3 learning patterns in the experiments described above. The results of the experiments are shown in Table 3. We can find from these experiments that the average learning speed of the complex-valued neural network is approximately 1.4 times faster than that of the real-valued neural network. The superscript * of a number means that the weights between the hidden layer and the output layer stayed in the neighborhood of the singular point 0 or $(0,0)$ from the beginning to the end of leaning. Table 4 shows the Euclidean distances between the weights (between the hidden layer and the output layer) and the singular point 0 or $(0,0)$ after the first learning cycle. In every case, the weights of the complex-valued neural network moved in the distance from the singular point 0 compared with those of the real-valued neural network. We believe that this phenomenon originates in the linear combinations of φ in Equations (22) and (23) shown in Section 4.

In the experiments on the learning patterns 1 and 2, the learning speeds of the cases * were slow uniformly. All the average learning speeds except the cases * were faster than the average learning speed. In the cases *, the

Table 2. Initial values of the weights and the thresholds.

(a) Complex-valued neural network

w_{10}	v_1	v_0
0.5 + 0.5i	0.000001 + 0.000001i	0.0

	case 1	case 2	case 3	case 4
w_{11}	−0.5	−0.5 + 0.5i	0.5i	0.5 + 0.5i

	case 5	case 6	case 7	case 8
w_{11}	0.5	0.5 – 0.5i	–0.5i	–0.5 – 0.5i

(b) Real-valued neural network

w_{10}	v_1	v_2	v_{10}	v_{20}
0.5	0.000001	0.000001	0.0	0.0

	case 1	case 2	case 3	case 4
w_{11}	−0.5	−0.5	0.0	0.5
w_{12}	0.0	0.5	0.5	0.5

	case 5	case 6	case 7	case 8
w_{11}	0.5	0.5	0.0	−0.5
w_{12}	0.0	−0.5	−0.5	−0.5

Table 3. Learning speed (the number of learning cycles needed to converge). Case number means the initial values of the weights between the input layer and the hidden layer (See Table 2). The superscript * of a number means that the weights between the hidden layer and the output layer stayed in the neighborhood of the singlar point 0 or (0, 0) from the beginning to the end of leaning.

(a) Learning pattern 1

Case number	1	2	3	4	5	6	7	8
Complex NN	7	5	3	4	3	5	7	13*
Real NN	13*	13*	7	5	5	5	7	13*

Case number	Average	Average except the cases*
Complex NN	5.9	4.9
Real NN	8.5	5.8

(b) Learning pattern 2

Case number	1	2	3	4	5	6	7	8
Complex NN	7	13*	7	5	3	4	3	5
Real NN	7	5	5	5	7	13*	13*	13*

Case number	Average	Average except the cases*
Complex NN	5.9	4.9
Real NN	8.5	5.8

(c) Learning pattern 3

Case number	1	2	3	4	5	6	7	8
Complex NN	7	7	6	5	5	5	6	7
Real NN	9	8	8	8	8	8	9	9

Case number	Average	Average except the cases*
Complex NN	6.0	6.0
Real NN	8.4	8.4

Table 4. The Euclidean distances between the weights (between the hidden layer and the output layer) and the singular point 0 or (0, 0) after the first learning cycle: $|v_1|$ for the complex-valued network, and $|v|$ for the real-valued neural network. Case number means the initial values of the weights between the input layer and the hidden layer (See Table 2).

(a) Learning pattern 1

Case number	1	2	3	4	5
Complex NN	0.23	0.38	0.44	0.53	0.44
Real NN	0.00	0.00	0.23	0.38	0.38

Case number	6	7	8	Average
Complex NN	0.38	0.23	0.00	0.33
Real NN	0.38	0.23	0.00	0.20

(b) Learning pattern 2

Case number	1	2	3	4	5
Complex NN	0.23	0.00	0.23	0.38	0.44
Real NN	0.23	0.38	0.38	0.38	0.23

Case number	6	7	8	Average
Complex NN	0.53	0.44	0.38	0.33
Real NN	0.00	0.00	0.00	0.20

(c) Learning pattern 3

Case number	1	2	3	4	5
Complex NN	0.21	0.21	0.23	0.25	0.25
Real NN	0.14	0.16	0.18	0.19	0.18

Case number	6	7	8	Average
Complex NN	0.25	0.23	0.21	0.23
Real NN	0.16	0.14	0.13	0.16

weights between the hidden layer and the output layer could not move away from the singular point, but, as a result, the learning speed became slow. And also, the number of cases * of the complex-valued neural network was 1, and that of the real-valued neural network 3, which suggested that the complex-valued neural network was not influenced by singular points compared with the real-valued neural network.

In the experiment on the learning pattern 3, the average learning speed of the complex-valued neural network was 1.4 times faster than that of the real-valued neural network, although there was no cases * in both of the neural networks. Figure 1 shows the weights between the hidden layer and the output layer after the first learning cycle and those after the learning finished for the learning pattern 3. Judging from Figure 1 and Table 4, the speed of the complex-valued neural network of moving away from the singularity is faster than that of the real-valued neural network, and it causes a difference of the learning speed.

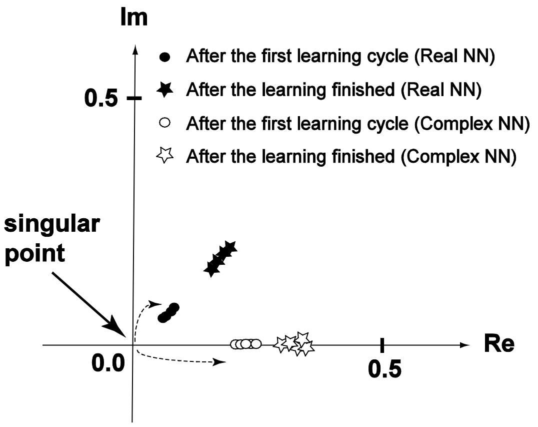

Figure 1. The behavior of the weights between the hidden layer and the output layer. Their initial values were set in the neighborhood of the singular point (the origin). The eight initial values (eight cases) were used for the weights between the input layer and the hidden layer (see Table 2). The distance of the weight of the complex-valued neural network from the singularity after the first learning cycle was larger than that of the real-valued neural network.

6. Conclusions

We compared theoretically and experimentally the influence of the singular points on the learning dynamics in the complex-valued neural network with that in the real-valued neural network. As a result, we found that the linear combination structure in the updating rule of the complex-valued neural network increased the speed of moving away from the singular points; the complex-valued neural network could not be easily influenced by the singular points. This is considered to be a result which supports the fast convergence of the Complex-BP algorithm.

It might be premature to conclude the statements described above are true because this paper deals with only very simple cases such as the network structures and the learning patterns. In the future, we will investigate more complicated cases.

Acknowledgements

The author extends special thanks to the anonymous reviewers for valuable comments.

REFERENCES

[1] A. Hirose, Ed., "Complex-Valued Neural Networks," World Scientific Publishing, Singapore, 2003.

[2] T. Nitta, Ed., "Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters," *Information Science Reference*, Pennsylvania, 2009. http://dx.doi.org/10.4018/978-1-60566-214-5

[3] T. Nitta and T. Furuya, "A Complex Back-Propagation Learning," *Transactions of Information Proceedings of the Society of Japan*, Vol. 32, No. 10, 1991, pp. 1319-1329.

[4] T. Nitta, "A Complex Numbered Version of the Back-Propagation Algorithm," Proceedings of the World Congress on Neural Networks, Portland, Vol. 3, 1993, pp. 576-579.

[5] T. Nitta, "An Extension of the Back-Propagation Algorithm to Complex Numbers," *Neural Networks*, Vol. 10, No. 8, 1997, pp. 1392-1415. http://dx.doi.org/10.1016/S0893-6080(97)00036-1

[6] T. Nitta, "Orthogonality of Decision Boundaries in Complex-Valued Neural Networks," *Neural Computation*, Vol. 16, No. 1, 2004, pp. 73-97. http://dx.doi.org/10.1162/08997660460734001

[7] T. Nitta, "Complex-Valued Neural Network and Complex-Valued Back-Propagation Learning Algorithm," In: P. W. Hawkes, Ed., *Advances in Imaging and Electron Physics*, Elsevier, Amsterdam, Vol. 152, 2008, pp. 153-221.

[8] D. E. Rumelhart, *et al*., "Parallel Distributed Processing," Vol. 1, MIT Press, 1986.

[9] N. Benvenuto and F. Piazza, "On the Complex Backpropagation Algorithm," *IEEE Transactions on Signal Processing*, Vol. 40, No. 4, 1992, pp. 967-969. http://dx.doi.org/10.1109/78.127967

[10] G. M. Georgiou and C. Koutsougeras, "Complex Domain Backpropagation," *IEEE Transactions on Circuits and Systems—II*: *Analog and Digital Signal Processing*, Vol. 39, No. 5, 1992, pp. 330-334.

[11] M. S. Kim and C. C. Guest, "Modification of Backpropagation Networks for Complex-Valued Signal Processing in Frequency Domain," *Proceedings of the International Joint Conference on Neural Networks*, Vol. 3, 1990, pp. 27-31.

[12] S. Amari, H. Park and T. Ozekii, "Singularities Affect Dynamics of Learning in Neuromanifolds," *Neural Computation*, Vol. 18, No. 5, 2006, pp. 1007-1065. http://dx.doi.org/10.1162/neco.2006.18.5.1007

[13] F. Cousseau, T. Ozeki and S. Amari, "Dynamics of Learning in Multilayer Perceptrons near Singularities," *IEEE Transactions on Neural Networks*, Vol. 19, No. 8, 2008, pp. 1313-1328. http://dx.doi.org/10.1109/TNN.2008.2000391

[14] T. Nitta, "Local Minima in Hierarchical Structures of Complex-Valued Neural Networks," *Neural Networks*, Vol. 43, 2013, pp. 1-7. http://dx.doi.org/10.1016/j.neunet.2013.02.002

[15] H. Wei, J. Zhang, F. Cousseau, T. Ozeki and S. Amari, "Dynamics of Learning near Singularities in Layered Networks," *Neural Computation*, Vol. 20, 2008, pp. 813-843. http://dx.doi.org/10.1162/neco.2007.12-06-414

[16] F. M. De Azevedo, S. S. Travessa and F. I. M. Argoud, "The Investigation of Complex Neural Network on Epileptiform Pattern Classification," *Proceedings of the 3rd European Medical and Biological Engineering Conference* (*EMBEC*'05), 2005, pp. 2800-2804.