Published in *Neural Processing Letters*, Vol.12, No.3, pp.239-246, 2000
(Submitted Aug. 13, 1999; accepted Feb. 1, 2000)

(a) **The title of the article**
An Analysis of the Fundamental Structure of Complex-valued Neurons

(b) **The authors' full names**
Tohru Nitta

(c) **Current Affiliation**
Neuroscience Research Institute,
National Institute of Advanced Industrial Science and Technology (AIST),
AIST Tsukuba Central 2, 1-1-1 Umezono, Tsukuba-shi, Ibaraki, 305-8568 Japan.
E-mail: tohru-nitta@aist.go.jp

# An Analysis on Fundamental Structure of Complex-valued Neuron

Tohru Nitta
*Electrotechnical Laboratory*

**Abstract.** This paper presents some results of an analysis on the decision boundaries of complex-valued neurons. The main results may be summarized as follows. (a) Weight parameters of a complex-valued neuron have a restriction which is concerned with two-dimensional motion. (b) The decision boundary of a complex-valued neuron consists of two hypersurfaces which intersect orthogonally, and divides a decision region into four equal sections.

**Keywords:** complex numbers, complex-valued neurons, learning, decision boundary

## 1. Introduction

Complex-valued neural networks have been proposed by several researchers in recent years [1, 2, 3, 4, 6, 7, 8], which are the extensions of usual real-valued neural networks to complex numbers. In particular, the *Complex-BP* algorithm is a complex-valued version of the usual real-valued back-propagation algorithm (called here, *Real-BP*) [9], which was proposed by several researchers independently in the early 1990's [1, 2, 4, 6, 7, 8]. This algorithm enables the network to learn complex-valued patterns naturally, and has the ability to learn 2D motion as its inherent property [6, 7, 8].

This paper makes clear the differences between the *real-valued neuron* used in the Real-BP and the *complex-valued neuron* used in the Complex-BP [6, 7, 8] by analyzing their fundamental properties from the view of architectures. The main results may be summarized as follows. (a) Weight parameters of a complex-valued neuron have a restriction which is concerned with two-dimensional motion, and learning proceeds under this restriction. (b) The decision boundary of a complex-valued neuron consists of two hypersurfaces which intersect orthogonally, and divides a decision region into four equal sections. It seems that the complex-valued neural network using such complex-valued neurons and the related Complex-BP algorithm are natural for learning of complex-valued patterns for the above reasons.

## 2. The Complex-valued Neuron

This section briefly describes the complex-valued neuron used in the Complex-BP algorithm [6, 7, 8]. The weights and threshold values of a complex-valued neuron are all complex numbers, and the output function $f_C$ of a complex-valued neuron is defined to be

$$f_C(z) = f_R(x) + i f_R(y), \tag{1}$$

where $z = x + iy$ is a complex-valued input signal to the complex-valued neuron, $i$ denotes $\sqrt{-1}$ and $f_R(u) = 1/(1 + \exp(-u))$, that is, the real and imaginary parts of the complex-valued output of a complex-valued neuron mean the sigmoid functions of the real part $x$ and imaginary part $y$ of the net input $z$ to neuron, respectively.

## 3. The Fundamental Structure of the Complex-valued Neuron

In this section, we analyze the properties of decision boundaries of the complex-valued neuron described in the previous section.

### 3.1. Weight Parameters of a Real-valued Neuron

We first examine the basic structures of weights of a real-valued neuron. Consider a real-valued neuron with $n$-inputs, weights $w_k \in \boldsymbol{R}$ $(1 \leq k \leq n)$, and a threshold value $\theta \in \boldsymbol{R}$, where $\boldsymbol{R}$ denotes the set of real numbers. Let an output function $f_R : \boldsymbol{R} \rightarrow \boldsymbol{R}$ of the neuron be $f_R(u) = 1/(1 + \exp(-u))$. Then, for $n$ input signals $x_k \in \boldsymbol{R}$ $(1 \leq k \leq n)$, the real-valued neuron generates $f_R(\sum_{k=1}^{n} w_k x_k + \theta)$ as an output. This may be interpreted as follows: a real-valued neuron moves a point $x_k$ on a real line (1 dimension) to another point $w_k x_k$ whose distance from the origin is $w_k$ times as long as that of the point $x_k$ $(1 \leq k \leq n)$, and regarding $w_1 x_1, \ldots, w_n x_n$ as vectors $\boldsymbol{w}_1 \boldsymbol{x}_1, \ldots, \boldsymbol{w}_n \boldsymbol{x}_n$, the real-valued neuron adds them, resulting in a 1-dimensional real-valued vector $\sum_{k=1}^{n} \boldsymbol{w}_k \boldsymbol{x}_k$, and finally, moves the end point of the vector $\sum_{k=1}^{n} \boldsymbol{w}_k \boldsymbol{x}_k$ to another point $(\sum_{k=1}^{n} w_k x_k) + \theta$ (Fig. 1). The output value of the real-valued neuron can be obtained by applying a nonlinear transformation $f_R$ to the value $(\sum_{k=1}^{n} w_k x_k) + \theta$. Thus, a real-valued neuron basically administers the movement of points on a real line (1 dimension), and its weight parameters $w_1, \ldots, w_n$ are completely independent of one another.

## 3.2. Weight Parameters of a Complex-valued Neuron

Next, we examine the basic structures of weights of a complex-valued neuron. Consider a complex-valued neuron with $n$-inputs, weights $w_k = w_k^r + iw_k^i \in \boldsymbol{C}$ $(1 \leq k \leq n)$, and a threshold value $\theta = \theta^r + i\theta^i \in \boldsymbol{C}$, where $\boldsymbol{C}$ denotes the set of complex numbers. Then, for $n$ input signals $x_k + iy_k \in \boldsymbol{C}$ $(1 \leq k \leq n)$, the complex-valued neuron generates

$$\begin{aligned} X+iY &= f_C\left(\sum_{k=1}^{n}(w_k^r+iw_k^i)(x_k+iy_k) + (\theta^r+i\theta^i)\right) \\ &= f_R\left(\sum_{k=1}^{n}(w_k^r x_k - w_k^i y_k)+\theta^r\right) + if_R\left(\sum_{k=1}^{n}(w_k^i x_k + w_k^r y_k)+\theta^i\right) \end{aligned} \tag{2}$$

as an output. Hence, a complex-valued neuron with $n$-inputs is equivalent to two real-valued neurons with $2n$-inputs in Fig. 2. We shall refer to a real-valued neuron corresponding to the real part $X$ of an output of a complex-valued neuron as a *Real-part Neuron*, and a real-valued neuron corresponding to the imaginary part $Y$ as an *Imaginary-part Neuron*.

Note here that

$$\begin{aligned} \begin{bmatrix} X \\ Y \end{bmatrix} &= F\left( \left[\begin{array}{cc|c|cc} w_1^r & -w_1^i & \cdots & w_n^r & -w_n^i \\ w_1^i & w_1^r & \cdots & w_n^i & w_n^r \end{array}\right] \left[\begin{array}{c} x_1 \\ y_1 \\ \hline \vdots \\ \hline x_n \\ y_n \end{array}\right] + \begin{bmatrix} \theta^r \\ \theta^i \end{bmatrix} \right) \\ &= F\left( |w_1| \begin{bmatrix} \cos\alpha_1 & -\sin\alpha_1 \\ \sin\alpha_1 & \cos\alpha_1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \cdots \right. \\ &\qquad \left. + |w_n| \begin{bmatrix} \cos\alpha_n & -\sin\alpha_n \\ \sin\alpha_n & \cos\alpha_n \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} \theta^r \\ \theta^i \end{bmatrix} \right), \end{aligned} \tag{3}$$

where, $F({}^t[x \;\; y]) = {}^t[f_R(x) \;\; f_R(y)]$, $\alpha_k = \arctan(w_k^i / w_k^r)$ $(1 \leq k \leq n)$. In equation (3), $|w_k|$ means reduction or magnification of the distance between a point $(x_k, y_k)$ and the origin in the complex plane, $\begin{bmatrix} \cos\alpha_k & -\sin\alpha_k \\ \sin\alpha_k & \cos\alpha_k \end{bmatrix}$ the counterclockwise rotation by $\alpha_k$ radians about the origin, and ${}^t[\theta^r \;\; \theta^i]$ translation. Thus, we find that a complex-valued neuron with $n$-inputs applies a linear transformation called *2D motion* to each input signal (complex number), that is, equation (3) basically involves *2D motion* (Fig. 3).

As seen in the previous section, a real-valued neuron basically administers the movement of points on a real line (1 dimension), and its weight parameters are completely independent of one another. On the other hand, as we have seen, a complex-valued neuron basically administers 2D motion on the complex plane, and we may also interpret that the learning means adjusting 2D motion. This structure imposes the following restrictions on a set of weight parameters of a complex-valued neuron (Fig. 2).

(Weight for the real part $x_k$ of an input signal to *Real-part Neuron*)
= (Weight for the imaginary part $y_k$ of an input signal to *Imaginary-part Neuron*), (4)

(Weight for the imaginary part $y_k$ of an input signal to *Real-part Neuron*)
= − (Weight for the real part $x_k$ of an input signal to *Imaginary-part Neuron*). (5)

Learning is carried out under these restrictions. From a different angle, we can see that *Real-part Neuron* and *Imaginary-part Neuron* influence each other via their weights.

Thus, we find that extending the real-valued neuron to complex numbers has varied the structure from 1 dimension to 2 dimensions. The structures of weight parameters described above will appear as orthogonality of decision boundaries in the next section.

### 3.3. Decision Boundaries in a Complex-valued Neuron

The decision boundary is a border by which pattern classifiers such as the Real-BP classify patterns, and generally consists of hypersurfaces. Decision boundaries of neural networks of real-valued neurons were examined empirically by Lippmann [5]. This section mathematically analyzes decision boundaries of complex-valued neurons.

Let the weights denote $\boldsymbol{w} = {}^t[w_1 \cdots w_n] = \boldsymbol{w}^r + i\boldsymbol{w}^i$, $\boldsymbol{w}^r = {}^t[w_1^r \cdots w_n^r]$, $\boldsymbol{w}^i = {}^t[w_1^i \cdots w_n^i]$, and let the threshold denote $\theta = \theta^r + i\theta^i$. Then, for $n$ input signals (complex numbers) $\boldsymbol{z} = {}^t[z_1 \cdots z_n] = \boldsymbol{x} + i\boldsymbol{y}$, $\boldsymbol{x} = {}^t[x_1 \cdots x_n]$, $\boldsymbol{y} = {}^t[y_1 \cdots y_n]$, the complex-valued neuron generates

$$X + iY = f_R\left([{}^t\boldsymbol{w}^r \quad -{}^t\boldsymbol{w}^i]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} + \theta^r\right) + i f_R\left([{}^t\boldsymbol{w}^i \quad {}^t\boldsymbol{w}^r]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} + \theta^i\right) \quad (6)$$

as an output. Here, for any two constants $C^R, C^I \in (0,1)$, let

$$X(\boldsymbol{x},\boldsymbol{y}) = f_R\left([{}^t\boldsymbol{w}^r \quad -{}^t\boldsymbol{w}^i]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} + \theta^r\right) = C^R, \tag{7}$$

$$Y(\boldsymbol{x},\boldsymbol{y}) = f_R\left([{}^t\boldsymbol{w}^i \quad {}^t\boldsymbol{w}^r]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} + \theta^i\right) = C^I. \tag{8}$$

Note here that expression (7) is the decision boundary for the real part of an output of the complex-valued neuron with $n$-inputs. That is, input signals $(\boldsymbol{x},\boldsymbol{y}) \in \boldsymbol{R}^{2n}$ are classified into two decision regions $\{(\boldsymbol{x},\boldsymbol{y}) \in \boldsymbol{R}^{2n} | X(\boldsymbol{x},\boldsymbol{y}) \geq C^R\}$ and $\{(\boldsymbol{x},\boldsymbol{y}) \in \boldsymbol{R}^{2n} | X(\boldsymbol{x},\boldsymbol{y}) < C^R\}$ by the hypersurface given by expression (7). Similarly, expression (8) is the decision boundary for the imaginary part. The normal vectors $H^R(\boldsymbol{x},\boldsymbol{y})$ and $H^I(\boldsymbol{x},\boldsymbol{y})$ of the decision boundaries ((7), (8)) are given by

$$\begin{aligned} H^R(\boldsymbol{x},\boldsymbol{y}) &= \left(\frac{\partial X}{\partial x_1} \cdots \frac{\partial X}{\partial x_n}\ \frac{\partial X}{\partial y_1} \cdots \frac{\partial X}{\partial y_n}\right) \\ &= f_R'\left([{}^t\boldsymbol{w}^r \quad -{}^t\boldsymbol{w}^i]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} + \theta^r\right) \cdot [{}^t\boldsymbol{w}^r \quad -{}^t\boldsymbol{w}^i], \end{aligned} \tag{9}$$

$$\begin{aligned} H^I(\boldsymbol{x},\boldsymbol{y}) &= \left(\frac{\partial Y}{\partial x_1} \cdots \frac{\partial Y}{\partial x_n}\ \frac{\partial Y}{\partial y_1} \cdots \frac{\partial Y}{\partial y_n}\right) \\ &= f_R'\left([{}^t\boldsymbol{w}^i \quad {}^t\boldsymbol{w}^r]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} + \theta^i\right) \cdot [{}^t\boldsymbol{w}^i \quad {}^t\boldsymbol{w}^r]. \end{aligned} \tag{10}$$

Noting that the inner product of expressions (9) and (10) is zero, we can find that the decision boundary for the real part of an output of a complex-valued neuron and that for the imaginary part intersect orthogonally. Therefore, the following theorem can be obtained.

THEOREM 1. *The decision boundaries for the real and imaginary parts of a complex-valued neuron intersect orthogonally.*

It can be easily shown that Theorem 1 also holds true for the other types of complex-valued neurons proposed in [1, 2, 4]. It should be noted here that there seems to be a problem on learning convergence in the formulation in [4]; the complex-valued back-propagation algorithm with the output function $f_C(z) = 1/(1+\exp(-z)), z = x + iy$ never converged in our experiments.

Generally, a real-valued neuron classifies input real-valued signals into two classes (0, 1). On the other hand, a complex-valued neuron classifies input complex-valued signals into four classes $(0,\ 1,\ i,\ 1+i)$. As described above, the decision boundary of a complex-valued neuron consists of two hypersurfaces which intersect orthogonally, and divides a decision region into four equal sections. Thus, a complex-valued neuron used in [1, 2, 4, 6, 7, 8] can be considered to have a natural decision boundary for complex-valued patterns.

For example, the fading equalization technology is an application domain suitable for the complex-valued neurons. Channel equalization in a digital communication system can be viewed as a pattern classification problem. The digital communication system receives a transmitted signal sequence with additive noise, and tries to estimate the true transmitted sequence. A transmitted signal can take one of the following four possible complex values: $-1-i, -1+i, 1-i$ and $1+i$ ($i = \sqrt{-1}$), that is, the estimate of the transmitted signal should be classified into four classes. Thus, the complex-valued neurons with orthogonal decision boundaries would be suitable for this domain.

## 4. Conclusions

We have clarified the differences between the real-valued neuron and the complex-valued neuron through theoretical analyses of their fundamental properties. In particular, we discovered that the complex-valued neuron had some inherent properties on decision boundary. The orthogonality property of decision boundary is well suited to the classification of complex-valued patterns into four classes $0$, $1$, $i$, and $1+i$. We believe that the Complex-BP algorithm employing such complex-valued neurons is a natural method to learn complex-valued patterns in this sense, and will be effectively used in fields dealing with complex numbers.

## Acknowledgements

The author would like to give special thanks to Drs. S. Akaho, Y. Akiyama, T. Furuya and the anonymous reviewers for valuable comments. He also expresses his thanks to K. Ohmaki for providing the opportunity for this study.

## References

1. N. Benvenuto and F. Piazza, "On the Complex Backpropagation Algorithm", IEEE Trans. Signal Processing, Vol.40, No.4, pp.967–969, 1992.
2. G. M. Georgiou and C. Koutsougeras, "Complex Domain Backpropagation", IEEE Trans. Circuits and Systems–II: Analog and Digital Signal Processing, Vol.39, No.5, pp.330–334, 1992.

3. A. Hirose, "Proposal of Fully Complex-valued Neural Networks", Proceedings of International Joint Conference on Neural Networks, Vol.4, pp.152–157, 1992.
4. M. S. Kim and C. C. Guest, "Modification of Backpropagation Networks for Complex-Valued Signal Processing in Frequency Domain", Proceedings of International Joint Conference on Neural Networks, Vol.3, pp.27–31, June 1990.
5. R. P. Lippmann, "An Introduction to Computing with Neural Nets", IEEE Acoustic, Speech and Signal Processing Magazine, pp.4–22, April 1987.
6. T. Nitta and T. Furuya, "A Complex Back-Propagation Learning", Transactions of Information Processing Society of Japan, Vol.32, No.10, pp.1319–1329, 1991 (in Japanese).
7. T. Nitta, "A Complex Numbered Version of the Back-Propagation Algorithm", Proceedings of INNS World Congress on Neural Networks, Portland, Vol.3, pp.576–579, July 1993.
8. T. Nitta, "An Extension of the Back-Propagation Algorithm to Complex Numbers", Neural Networks, Vol.10, No.8, pp.1392–1415, 1997.
9. D. E. Rumelhart et al., Parallel Distributed Processing, Vol.1, MIT Press, 1986.