This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Decomposition of Rotor Hopfield Neural Networks Using Complex Numbers

Masaki Kobayashi

***Abstract*— A complex-valued Hopfield neural network (CHNN) is a multistate model of a Hopfield neural network. It has the disadvantage of low noise tolerance. Meanwhile, a symmetric CHNN (SCHNN) is a modification of a CHNN that improves noise tolerance. Furthermore, a rotor Hopfield neural network (RHNN) is an extension of a CHNN. It has twice the storage capacity of CHNNs and SCHNNs, and much better noise tolerance than CHNNs, although it requires twice many connection parameters. In this brief, we investigate the relations between CHNN, SCHNN, and RHNN; an RHNN is uniquely decomposed into a CHNN and SCHNN. In addition, the Hebbian learning rule for RHNNs is decomposed into those for CHNNs and SCHNNs.**

***Index Terms*— Hebbian learning rule, rotor Hopfield neural networks (RHNNs), symmetric complex-valued Hopfield neural networks (SCHNNs), widely linear estimation.**

I. Introduction

Acomplex-valued Hopfield neural network (CHNN) is a multistate model of a Hopfield neural network [1]–[3]. It is often applied to image storage [4], [5]. Several modifications and extensions of CHNN have been proposed [6]–[16]. A symmetric CHNN (SCHNN) is a modification of a CHNN [17]. An SCHNN has the same number of connection parameters and almost the same storage capacity as a CHNN. Since an SCHNN does not have rotational invariance, however, it has much better noise tolerance. A rotor Hopfield neural network (RHNN) is an extension of a CHNN [18]–[20]. Like SCHNN, RHNN does not have rotational invariance and improves noise tolerance significantly. In addition, RHNN has twice the storage capacity of CHNN, although twice many connection parameters are necessary [20]–[22]. RHNN has been applied to dynamic associative memories, such as chaotic associative memories that never recall rotated patterns [23], [24].

In this brief, the relations between CHNNs, SCHNNs, and RHNNs are studied. CHNNs and SCHNNs are organized based on complex numbers. Meanwhile, RHNNs are organized based on 2-D vectors and 2×2 matrices. RHNNs include both CHNNs and SCHNNs; CHNNs and SCHNNs can be represented by RHNNs. It is shown that an RHNN is uniquely decomposed into a CHNN and SCHNN. The decomposition matches a widely linear estimation [25]. However, there is a difference. In a widely linear estimation, the connections are one way. Meanwhile, in an RHNN, they are mutual. Widely linear estimation with complex numbers has been utilized in communications and adaptive filters [26]–[28]. In recent years, applications of widely linear estimation have been extended to quaternionic signal processing [29]. Moreover, the Hebbian learning rule for RHNNs is decomposed into those for CHNNs and SCHNNs. This brief provides a new perspective on the RHNN, and extends the strategy of learning algorithms.

II. Complex-Valued Hopfield Neural Networks

In this section, a CHNN, which is a multistate extension of a Hopfield neural network using complex numbers, is briefly described [3].

Manuscript received July 31, 2016; revised October 6, 2016 and December 1, 2016; accepted January 22, 2017.

The author is with the Mathematical Science Center, University of Yamanashi, Kofu 400-8511, Japan.

Digital Object Identifier 10.1109/TNNLS.2017.2657781

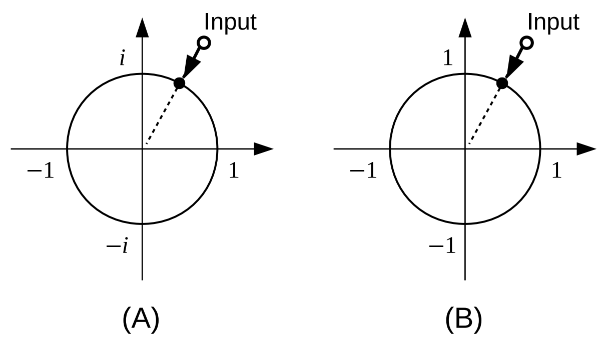

Fig. 1. (A) Complex-valued phasor activation function. (B) Rotor activation function. They are different only in the vertical axis.

Let $z_j = x_j + y_j i$ and u_{jk} be the state of neuron j and the connection weight from neuron k to neuron j, respectively. The connection weights satisfy the stability condition $u_{jk} = \overline{u_{kj}}$, where $\overline{u}$ is the complex conjugate of u. The weighted sum input I_j^C to neuron j is defined as

$$I_j^C = \sum_{k \neq j} u_{jk} z_k. \tag{1}$$

Two types of activation functions, the phasor and the multistate, are often employed. The phasor activation function is described first. For a weighted sum input $I = r\exp(i\theta)$, where $r > 0$ and $0 \leq \theta < 2\pi$, the phasor activation function $f(I)$ is defined as $f(I) = \exp(i\theta)$. Fig. 1(A) shows the phasor activation function. Turning now to the multistate activation function, we denote $\theta_K = \frac{\pi}{K}$ with the quantization level K. The multistate activation function $f(I)$ is defined as follows:

$$f(I) = \begin{cases} \exp(i\theta_K) & (0 \leq \theta < 2\theta_K) \\ \exp(3i\theta_K) & (2\theta_K \leq \theta < 4\theta_K) \\ \vdots & \vdots \\ \exp((2K-1)i\theta_K) & (2(K-1) \leq \theta < 2\pi). \end{cases} \tag{2}$$

Fig. 2(A) is an illustration of the multistate activation function with $K = 6$. For a weighted sum input I^C, the neuron state is determined by $f(I^C)$. In particular, $|z_j| = 1$ holds. We denote a state of a CHNN by $\mathbf{z} = (z_1, z_2, \cdots, z_N)$, where N is the number of neurons. Then, the energy of the CHNN is defined as follows:

$$E^C(\mathbf{z}) = -\frac{1}{2}\sum_{j \neq k} \overline{z_j} u_{jk} z_k. \tag{3}$$

The energy is a real number from $\overline{E^C} = E^C$; it does not increase under an asynchronous updating mode, and a CHNN always converges.

To describe the Hebbian learning rule for CHNNs, let $z_j^p = x_j^p + y_j^p i$ be the jth component of the pth training pattern. Then, the connection weight u_{jk}^H given by the Hebbian learning rule is as

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

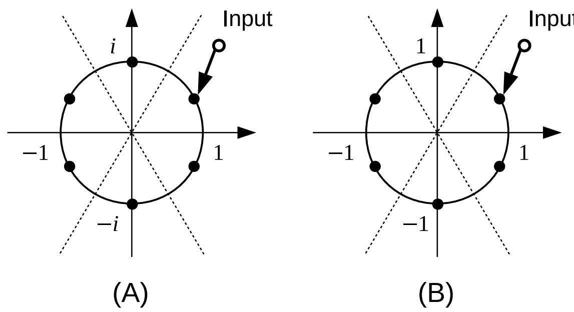

Fig. 2. (A) Complex-valued multistate activation function with $K = 6$. (B) Rotor multistate activation function with $K = 6$. They are different only in the vertical axis.

follows:

$$u^H_{jk} = \sum_p z^p_j \overline{z^p_k}. \tag{4}$$

These connection weights satisfy the stability condition $u^H_{jk} = \overline{u^H_{kj}}$.

III. Symmetric Complex-Valued Hopfield Neural Networks

In this section, an SCHNN is briefly described [17]. An SCHNN is a modification of a CHNN. The symbols z_j and z^p_j used in the CHNN are also employed in the SCHNN. The connection weights are denoted by v_{jk}. They satisfy the stability condition $v_{jk} = v_{kj}$. An SCHNN modifies the weighted sum input of CHNN as follows:

$$I^S_j = \sum_{k \neq j} v_{jk} \overline{z_k}. \tag{5}$$

For a weighted sum input I^S, the neuron state is determined by $f(I^S)$. We denote the state of an SCHNN by $\mathbf{z} = (z_1, z_2, \cdots, z_N)$. Then, the energy of the SCHNN is defined as

$$E^S(\mathbf{z}) = -\frac{1}{2}\mathrm{Re}\left(\sum_{j \neq k} \overline{z_j} v_{jk} \overline{z_k}\right) \tag{6}$$

where $\mathrm{Re}(\cdot)$ means the real part. Since the above summation is not a real number, it is necessary to take its real part. The energy does not increase under an asynchronous updating mode, and an SCHNN always converges.

To describe the Hebbian learning rule for SCHNNs, the connection weight v^H_{jk} given by the Hebbian learning rule is as follows:

$$v^H_{jk} = \sum_p z^p_j z^p_k. \tag{7}$$

These connection weights satisfy the stability condition $v^H_{jk} = v^H_{kj}$.

IV. Rotor Hopfield Neural Networks

An RHNN is an extension of a CHNN [20]. An RHNN has twice the CHNN's large storage capacity and a much higher noise tolerance [20]–[22]. The neuron states and connection weights are represented by 2-D vectors and 2×2 matrices, respectively, whose elements are real numbers. Let $\mathbf{z}_j$ and $\hat{\omega}_{jk}$ be the state of neuron j and the connection weight from neuron k to neuron j, respectively. The connection weight $\hat{\omega}_{jk}$ is described using the elements as follows:

$$\hat{\omega}_{jk} = \begin{pmatrix} a_{jk} & b_{jk} \\ c_{jk} & d_{jk} \end{pmatrix}. \tag{8}$$

The state $\mathbf{z}_j = (x_j \;\; y_j)^T$ of the rotor neuron is identical to the state $x_j + y_j i$ of the complex-valued neuron. Here, the superscript T means the transpose. The activation function of an RHNN is also identical to that of a CHNN. In particular, $|\mathbf{z}_j|^2 = x_j^2 + y_j^2 = 1$ holds. Figs. 1(B) and 2(B) show the rotor versions of the phasor and multistate activation functions, respectively. The weighted sum input $\mathbf{I}^R_j$ to neuron j is defined as follows:

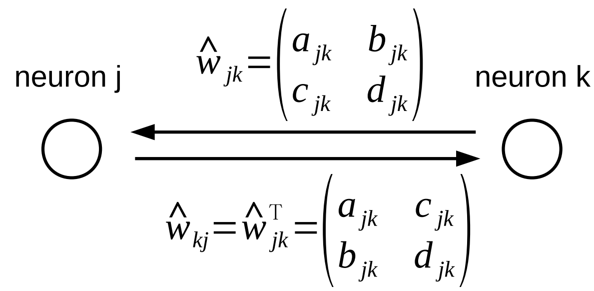

Fig. 3. Mutual connections of RHNNs are transpose matrices.

$$\mathbf{I}^R_j = \sum_{k \neq j} \hat{\omega}_{jk} \mathbf{z}_k. \tag{9}$$

Since $\hat{\omega}_{jk}$ is a 2×2 matrix, the weighted sum input $\mathbf{I}^R_j$ is a 2-D vector. The connection weights satisfy the stability condition $\hat{\omega}_{jk} = \hat{\omega}^T_{kj}$. Fig. 3 shows the mutual connection between two rotor neurons.

We denote the state of an RHNN by $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N)$. Then, the energy of the RHNN is defined as

$$E^R(\mathbf{z}) = -\frac{1}{2}\sum_{j \neq k} \mathbf{z}^T_j \hat{\omega}_{jk} \mathbf{z}_k. \tag{10}$$

The energy is a real number. It does not increase under asynchronous updating mode, and an RHNN always converges.

To define the Hebbian learning rule for RHNNs, we denote the jth component of the pth training pattern as $\mathbf{z}^p_j = (x^p_j \;\; y^p_j)^T$. Then, the connection weight $\hat{\omega}^H_{jk}$ given by the Hebbian learning rule is as follows:

$$\hat{\omega}^H_{jk} = \sum_p \hat{\omega}^p_{jk}, \tag{11}$$

$$\hat{\omega}^p_{jk} = \mathbf{z}^p_j (\mathbf{z}^p_k)^T = \begin{pmatrix} x^p_j \\ y^p_j \end{pmatrix} \begin{pmatrix} x^p_k & y^p_k \end{pmatrix} \tag{12}$$

$$= \begin{pmatrix} x^p_j x^p_k & x^p_j y^p_k \\ y^p_j x^p_k & y^p_j y^p_k \end{pmatrix}. \tag{13}$$

The equality $\hat{\omega}^p_{jk} \mathbf{z}^p_k = \mathbf{z}^p_j$ then holds.

V. Representation of CHNN Using RHNN

A CHNN is described using an RHNN. We first denote that $u_{jk} = \alpha_{jk} + \beta_{jk} i$. A connection weight takes a transformation role and can be represented by a matrix. The connection weight $\hat{u}_{jk}$ of the RHNN corresponding to u_{jk} is considered as follows:

$$\hat{u}_{jk} = \begin{pmatrix} \alpha_{jk} & -\beta_{jk} \\ \beta_{jk} & \alpha_{jk} \end{pmatrix}. \tag{14}$$

From $u_{jk} = \overline{u_{kj}}$, the stability condition $\hat{u}_{jk} = \hat{u}^T_{kj}$ holds. Thus, a CHNN is considered an RHNN. Fig. 4(A) shows the mutual connections between two neurons by complex numbers; Fig. 4(B) shows mutual connections between two neurons by RHNN.

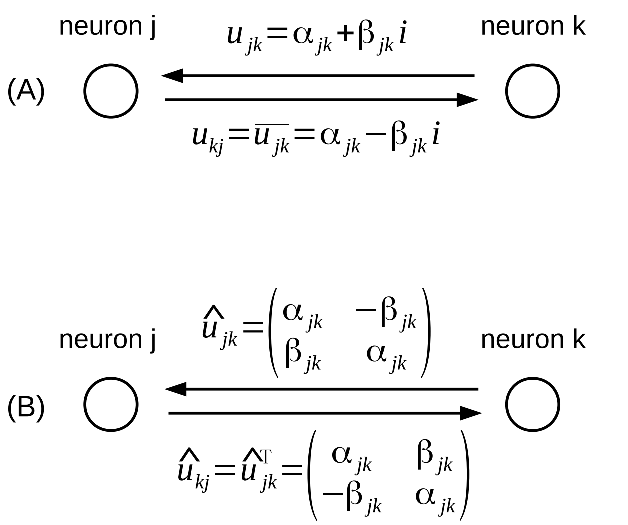

Fig. 4. (A) Complex-valued representation of mutual connections of a CHNN. They are complex conjugate. (B) Representation by an RHNN.

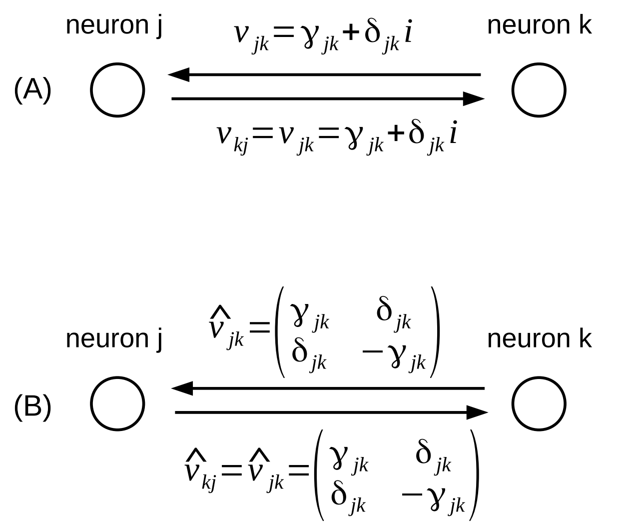

Fig. 5. (A) Complex-valued representation of mutual connections of an SCHNN. They are symmetric. (B) Representation by an RHNN.

The energy can be rewritten as follows:

$$E^C(\mathbf{z}) = -\frac{1}{2}\sum_{j\neq k}(x_j - y_j i)(\alpha_{jk}+\beta_{jk}i)(x_k+y_k i) \tag{15}$$

$$= -\frac{1}{2}\sum_{j\neq k}(x_j - y_j i)((\alpha_{jk}x_k - \beta_{jk}y_k)+(\beta_{jk}x_k+\alpha_{jk}y_k)i) \tag{16}$$

$$= -\frac{1}{2}\sum_{j\neq k}(x_j(\alpha_{jk}x_k - \beta_{jk}y_k)+y_j(\beta_{jk}x_k+\alpha_{jk}y_k)) \tag{17}$$

$$= -\frac{1}{2}\sum_{j\neq k}(x_j \quad y_j)\begin{pmatrix}\alpha_{jk}x_k - \beta_{jk}y_k\\ \beta_{jk}x_k+\alpha_{jk}y_k\end{pmatrix} \tag{18}$$

$$= -\frac{1}{2}\sum_{j\neq k}(x_j \quad y_j)\begin{pmatrix}\alpha_{jk} & -\beta_{jk}\\ \beta_{jk} & \alpha_{jk}\end{pmatrix}\begin{pmatrix}x_k\\ y_k\end{pmatrix} \tag{19}$$

$$= -\frac{1}{2}\sum_{j\neq k}\mathbf{z}_j^T\hat{u}_{jk}\mathbf{z}_k. \tag{20}$$

Since $E^C(\mathbf{z})$ is a real number, the imaginary part in (17) vanishes.

By identification of a complex number $x+yi$ and a matrix $\begin{pmatrix}x & -y\\ y & x\end{pmatrix}$, the Hebbian learning rule for CHNNs can be rewritten as follows:

$$\hat{u}^H_{jk} = \sum_p \hat{u}^p_{jk}, \tag{21}$$

$$\hat{u}^p_{jk} = \begin{pmatrix}x^p_j & -y^p_j\\ y^p_j & x^p_j\end{pmatrix}\begin{pmatrix}x^p_k & y^p_k\\ -y^p_k & x^p_k\end{pmatrix} \tag{22}$$

$$= \begin{pmatrix}x^p_j x^p_k + y^p_j y^p_k & x^p_j y^p_k - y^p_j x^p_k\\ y^p_j x^p_k - x^p_j y^p_k & y^p_j y^p_k + x^p_j x^p_k\end{pmatrix}. \tag{23}$$

Then, the equality $\hat{u}^p_{jk}\mathbf{z}^p_k = \mathbf{z}^p_j$ holds.

VI. Representation of SCHNN Using RHNN

An SCHNN is described using an RHNN. We denote $v_{jk} = \gamma_{jk} + \delta_{jk}i$. The following equality holds:

$$v_{jk}\overline{z_k} = (\gamma_{jk}x_k + \delta_{jk}y_k) + (\delta_{jk}x_k - \gamma_{jk}y_k)i. \tag{24}$$

Therefore, the connection weight $\hat{v}_{jk}$ of the RHNN corresponding to v_{jk} is considered as follows:

$$\hat{v}_{jk} = \begin{pmatrix}\gamma_{jk} & \delta_{jk}\\ \delta_{jk} & -\gamma_{jk}\end{pmatrix}. \tag{25}$$

From $v_{jk} = v_{kj}$, the stability condition $\hat{v}_{jk} = \hat{v_{kj}}^T$ holds. Thus, an SCHNN can be considered as an RHNN. Fig. 5(A) shows the mutual connections between two neurons by complex numbers, and Fig. 5(B) shows mutual connections between two neurons by RHNN.

The energy can be rewritten as follows:

$$E^S(\mathbf{z}) = -\frac{1}{2}\mathrm{Re}\left(\sum_{j\neq k}(x_j - y_j i)(\gamma_{jk}+\delta_{jk}i)(x_k - y_k i)\right) \tag{26}$$

$$= -\frac{1}{2}\sum_{j\neq k}\mathrm{Re}((x_j - y_j i)$$

$$\{(\gamma_{jk}x_k + \delta_{jk}y_k) + (\delta_{jk}x_k - \gamma_{jk}y_k)i\}) \tag{27}$$

$$= -\frac{1}{2}\sum_{j\neq k}(x_j(\gamma_{jk}x_k + \delta_{jk}y_k) + y_j(\delta_{jk}x_k - \gamma_{jk}y_k)) \tag{28}$$

$$= -\frac{1}{2}\sum_{j\neq k}(x_j \quad y_j)\begin{pmatrix}\gamma_{jk}x_k + \delta_{jk}y_k\\ \delta_{jk}x_k - \gamma_{jk}y_k\end{pmatrix} \tag{29}$$

$$= -\frac{1}{2}\sum_{j\neq k}(x_j \quad y_j)\begin{pmatrix}\gamma_{jk} & \delta_{jk}\\ \delta_{jk} & -\gamma_{jk}\end{pmatrix}\begin{pmatrix}x_k\\ y_k\end{pmatrix} \tag{30}$$

$$= -\frac{1}{2}\sum_{j\neq k}\mathbf{z}_j^T\hat{v}_{jk}\mathbf{z}_k. \tag{31}$$

$\overline{z}$ is identified with

$$\begin{pmatrix}x\\ -y\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix}. \tag{32}$$

Then, the Hebbian learning rule for SCHNNs can be rewritten as follows:

$$\hat{v}^H_{jk} = \sum_p \hat{v}^p_{jk}, \tag{33}$$

$$\hat{v}^p_{jk} = \begin{pmatrix} x^p_j & -y^p_j \\ y^p_j & x^p_j \end{pmatrix} \begin{pmatrix} x^p_k & -y^p_k \\ y^p_k & x^p_k \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{34}$$

$$= \begin{pmatrix} x^p_j x^p_k - y^p_j y^p_k & x^p_j y^p_k + y^p_j x^p_k \\ y^p_j x^p_k + x^p_j y^p_k & y^p_j y^p_k - x^p_j x^p_k \end{pmatrix}. \tag{35}$$

The matrix $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ corresponds to the complex conjugate. Then, the equality $\hat{v}^p_{jk}\mathbf{z}^p_k = \mathbf{z}^p_j$ holds.

VII. Decomposition of Rotor Hopfield Neural Networks

In this section, an RHNN is decomposed into a CHNN and an SCHNN.

Theorem 1: For connection weights $\hat{w}_{jk}$ of an RHNN, there uniquely exist connection weights $\hat{u}_{jk}$ and $\hat{v}_{jk}$ of a CHNN and an SCHNN, such that $\hat{w}_{jk} = \hat{u}_{jk} + \hat{v}_{jk}$.

Proof: We determine connection weights $\hat{u}_{jk}$ and $\hat{v}_{jk}$ of the CHNN and SCHNN, such that $\hat{w}_{jk} = \hat{u}_{jk} + \hat{v}_{jk}$. By solving the equation

$$\begin{pmatrix} a_{jk} & b_{jk} \\ c_{jk} & d_{jk} \end{pmatrix} = \begin{pmatrix} \alpha_{jk} & -\beta_{jk} \\ \beta_{jk} & \alpha_{jk} \end{pmatrix} + \begin{pmatrix} \gamma_{jk} & \delta_{jk} \\ \delta_{jk} & -\gamma_{jk} \end{pmatrix} \tag{36}$$

$$= \begin{pmatrix} \alpha_{jk} + \gamma_{jk} & -\beta_{jk} + \delta_{jk} \\ \beta_{jk} + \delta_{jk} & \alpha_{jk} - \gamma_{jk} \end{pmatrix} \tag{37}$$

the unique decomposition is obtained as follows:

$$\hat{u}_{jk} = \frac{1}{2}\begin{pmatrix} a_{jk} + d_{jk} & b_{jk} - c_{jk} \\ c_{jk} - b_{jk} & a_{jk} + d_{jk} \end{pmatrix}, \tag{38}$$

$$\hat{v}_{jk} = \frac{1}{2}\begin{pmatrix} a_{jk} - d_{jk} & b_{jk} + c_{jk} \\ b_{jk} + c_{jk} & d_{jk} - a_{jk} \end{pmatrix}, \tag{39}$$

$$\hat{w}_{jk} = \hat{u}_{jk} + \hat{v}_{jk}. \tag{40}$$

■

We put u_{jk} and v_{jk} as follows:

$$u_{jk} = \frac{1}{2}(a_{jk} + d_{jk}) + \frac{1}{2}(c_{jk} - b_{jk})i, \tag{41}$$

$$v_{jk} = \frac{1}{2}(a_{jk} - d_{jk}) + \frac{1}{2}(b_{jk} + c_{jk})i. \tag{42}$$

Then, the following equality holds:

$$u_{jk}z_k + v_{jk}\overline{z_k} = v_{jk}(x_k + y_k i) + u_{jk}(x_k - y_k i) \tag{43}$$

$$= a_{jk}x_k + b_{jk}y_k + (c_{jk}x_k + d_{jk}y_k)i. \tag{44}$$

This expression (44) is identical to the 2-D vector $\hat{w}_{jk}\mathbf{z}_k$. Therefore, an RHNN can be represented with complex numbers; an RHNN is identical to a combination of a CHNN and an SCHNN. A rotor neuron is identical to a complex-valued neuron. The connection weights of the RHNN are considered pairs (u_{jk}, v_{jk}) of complex numbers, where u_{jk} and v_{jk} correspond to the CHNN and SCHNN, respectively. The weighted sum input I^R_j to neuron j is defined as follows:

$$I^R_j = \sum_{k \neq j}(u_{jk}z_k + v_{jk}\overline{z_k}) = I^C_j + I^S_j. \tag{45}$$

This matches the widely linear estimation [25]. The state of neuron j is determined by $f(I^R_j)$. Here, I^R_j is a complex number identical to the weighted sum input $\mathbf{I}^R_j$ of the RHNN. The energy of the RHNN is also decomposed into those of the CHNN and SCHNN.

Corollary 1: $E^R(\mathbf{z}) = E^C(\mathbf{z}) + E^S(\mathbf{z})$

Proof: From equalities (20) and (31) and Theorem 1, the following equalities hold:

$$E^R(\mathbf{z}) = -\frac{1}{2}\sum_{j \neq k} \mathbf{z}^T_j \hat{w}_{jk}\mathbf{z}_k \tag{46}$$

$$= -\frac{1}{2}\sum_{j \neq k} \mathbf{z}^T_j (\hat{u}_{jk} + \hat{v}_{jk})\mathbf{z}_k \tag{47}$$

$$= -\frac{1}{2}\sum_{j \neq k} \mathbf{z}^T_j \hat{u}_{jk}\mathbf{z}_k - \frac{1}{2}\sum_{j \neq k} \mathbf{z}^T_j \hat{v}_{jk}\mathbf{z}_k \tag{48}$$

$$= E^C(\mathbf{z}) + E^S(\mathbf{z}). \tag{49}$$

■

In Corollary 1, $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N)$ in $E^R(\mathbf{z})$ is identified with $\mathbf{z} = (z_1, z_2, \cdots, z_N)$ in $E^C(\mathbf{z})$ and $E^S(\mathbf{z})$. Thus, the Hebbian learning rule for RHNNs is decomposed into those for CHNNs and SCHNNs. In fact, from the Hebbian learning rules (11), (13), (21), (23), (33), and (35), the following decomposition of the Hebbian learning rule is obtained.

Theorem 2: The connection weights obtained by the Hebbian learning rule for RHNNs are decomposed into those for CHNNs and SCHNNs as follows:

$$\hat{w}^H_{jk} = \frac{1}{2}(\hat{u}^H_{jk} + \hat{v}^H_{jk}). \tag{50}$$

VIII. Conclusion

This brief provides the unique decomposition of an RHNN into a CHNN and an SCHNN. This decomposition provides a new perspective on the RHNN, and could extend the strategy of learning algorithms. For example, it enables the corresponding CHNN and SCHNN to learn independently. The Hebbian learning rule for RHNNs is identical to the combination of those for CHNNs and SCHNNs. It is an important fact that the weighted sum input of the RHNN matches a widely linear estimation. We plan to study new learning algorithms utilizing the decomposition of RHNNs.

References

[1] A. J. Noest, "Phasor neural networks," in *Proc. Neural Inf. Process. Syst.*, 1988, pp. 584–591.

[2] A. J. Noest, "Discrete-state phasor neural networks," *Phys. Rev. A, Gen. Phys.*, vol. 38, no. 4, pp. 2196–2199, 1988.

[3] S. Jankowski, A. Lozowski, and J. M. Zurada, "Complex-valued multistate neural associative memory," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1491–1496, Nov. 1996.

[4] M. K. Muezzinoglu, C. Guzelis, and J. M. Zurada, "A new design method for the complex-valued multistate Hopfield associative memory," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 891–899, Jul. 2003.

[5] G. Tanaka and K. Aihara, "Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1463–1473, Sep. 2009.

[6] A. Hirose, *Complex-Valued Neural Networks: Theories and Applications* (Series on Innovative Intelligence), vol. 5. Singapore: World Scientific, 2003.

[7] A. Hirose, Ed., *Complex-Valued Neural Networks* (Series on Studies in Computational Intelligence), 2nd ed. Springer, 2012.

[8] A. Hirose, Ed., *Complex-Valued Neural Networks: Advances and Applications* (The IEEE Press Series on Computational Intelligence). Hoboken, NJ, USA: Wiley, 2013.

[9] T. Nitta, *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*. Information Science Publishing, 2009.

[10] M. Kobayashi, "Hyperbolic Hopfield neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 2, pp. 335–341, Feb. 2013.

[11] T. Isokawa, H. Nishimura, and N. Matsui, "On the fundamental properties of fully quaternionic Hopfield network," in *Proc. IEEE World Congr. Comput. Intell.*, 2012, pp. 1246–1249.

[12] M. Kobayashi, "Attractors accompanied with a training pattern of multivalued Hopfield neural networks," *IEEJ Trans. Elect. Electron. Eng.*, vol. 10, no. 2, pp. 195–200, Mar. 2015.

[13] M. Kobayashi, "Rotational invariance of quaternionic Hopfield neural networks," *IEEJ Trans. Elect. Electron. Eng.*, vol. 11, no. 4, pp. 516–520, Jul. 2016.

[14] Y. Suzuki and M. Kobayashi, "Complex-valued bipartite auto-associative memory," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E97-A, no. 8, pp. 1680–1687, 2014.

[15] M. Kobayashi, "Hybrid quaternionic Hopfield neural network," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E98-A, no. 7, pp. 1512–1518, 2015.

[16] T. Minemoto, T. Isokawa, M. Kobayashi, H. Nishimura, and N. Matsui, "On the performance of quaternionic bidirectional auto-associative memory," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2015, pp. 2910–2915.

[17] M. Kobayashi, "Symmetric complex-valued Hopfield neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[18] M. Kobayashi, "Information geometry of rotor Boltzmann machines," *Nonlinear Theory Appl.*, vol. 7, no. 2, pp. 266–282, 2016.

[19] M. Kitahara and M. Kobayashi, "Fundamental abilities of rotor associative memory," in *Proc. IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, Aug. 2010, pp. 497–502.

[20] M. Kitahara and M. Kobayashi, "Projection rule for rotor Hopfield neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1298–1307, Jul. 2014.

[21] J. Cook, "The mean-field theory of a Q-state neural network model," *J. Phys. A, Math. General*, vol. 22, no. 12, pp. 2057–2068, 1989.

[22] Y. Nakamura, K. Torii, and T. Munakata, "Neural-network model composed of multidimensional spin neurons," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 2, pp. 1538–1546, Feb. 1995.

[23] M. Kitahara, M. Kobayashi, and M. Hattori, "Chaotic rotor associative memory," in *Proc. Int. Symp. Nonlinear Theory Appl.*, 2009, pp. 399–402.

[24] Y. Suzuki, M. Kitahara, and M. Kobayashi, "Rotor associative memory with a periodic activation function," in *Proc. IEEE World Congr. Comput. Intell.*, Jun. 2012, pp. 720–727.

[25] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 2030–2033, Aug. 1995.

[26] H. Gerstacker, R. Schober, and A. Lampe, "Receivers with widely linear processing for frequency-selective channels," *IEEE Trans. Commun.*, vol. 51, no. 9, pp. 1512–1523, Sep. 2003.

[27] R. Schober, W. H. Gerstacker, and L. H. J. Lampe, "Data-aided and blind stochastic gradient algorithms for widely linear MMSE MAI suppression for DS-CDMA," *IEEE Trans. Signal Process.*, vol. 52, no. 3, pp. 746–756, Mar. 2004.

[28] D. P. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*. Hoboken, NJ, USA: Wiley, 2009.

[29] Y. Xia, C. Jahanchahi, T. Nitta, and D. P. Mandic, "Performance bounds of quaternion estimators," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3287–3292, Dec. 2015.