Complex-Valued Neural Network for Hermitian Matrices

Qian Zhang, and Xuezhong Wang,

Abstract—**This paper proposes neural network for computing the eigenvectors of Hermitian matrices. For the eigenvalues of Hermitian matrices, we establish an explicit representation for the solution of the neural network based on Hermitian matrices and analyze its convergence property. We also consider to compute the eigenvectors of skew-symmetric matrices and skew-Hermitian matrices, corresponding to the imaginary maximal or imaginary minimal eigenvalue, based on the neural network for computing the eigenvectors of Hermitian matrices. Numerical examples are given to illustrate our theoretical result are valid.**

Index Terms—**Complex differential equations, complex-valued neural network, Hermitian matrix, eigenvectors.**

I. Introduction

THE unknown variables are complex-valued vectors in many scientific and engineering problems. A main aim is to find these variables by minimization of a complex-valued optimization problem with constraints [1], [2]. Complex-valued systems arise applications from adaptive signal processing for highly functional sensing and imaging, in automatic control in unknown and changing environment, in brain-like information processing and in robotics inspired by human neural systems. In the field of signal processing, for example, complex-valued systems are widely applied, as in land-surface classification, in the generation of digital elevation maps and in speech synthesis [3], [4], [5], [6], [7], [8].

For background to the eigenvalue problem, we recommend the monographs by Stewart and Sun [9], Golub and Van Loan [10], and Wilkinson [11]. There are several numerical methods for solving the eigenvalue problem. They include approaches based on the power and inverse iteration methods, QR methods, Lanczos based methods and the Jacobi-Davidson method [12], [13], [14].

Throughout this paper, we assume that n will be reserved to denote the index upper bounds, unless stated otherwise, and i, j, k will be reserved to denote the index. We take Greek alphabet $\alpha, \beta, \gamma, \ldots$ for scalars, small letters $x, u, v, \ldots$ for vectors, capital letters $A, B, C, \ldots$ for $n \times n$ matrices, and calligraphic letters $\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$ for $2n \times 2n$ matrices. For a given matrix $A \in \mathbb{C}^{m\times n}$, we denote $A^\top$ and A^* the transpose and the complex conjugated transpose of A, respectively. We use $|\alpha|$ to denote the model of a complex number $\alpha \in \mathbb{C}$.

Manuscript received January 10, 2017; revised June 22, 2017.

Q. Zhang is with the School of Mathematics and Statistics, Longdong University, Qingyang, 745000, P. R. of China. E-mail: zhqian9912@163.com.

X. Wang is with the School of Mathematics and Statistics, Hexi University, Zhangye, 734000, P. R. of China. E-mail:xuezhongwang77@126.com.

A. Eigenpairs of Hermitian matrices

The eigenvalue problem of $A \in \mathbb{C}^{n\times n}$ are elements of the set $\lambda(A)$ defined by [9], [10] as follows

$$\lambda(A) = \{\lambda \in \mathbb{C} : \det(A - \lambda I) = 0\}.$$

If $\lambda \in \lambda(A)$ and two nonzero vectors $u, v \in \mathbb{C}^n$ satisfy

$$Au = \lambda u, \quad v^* A = \lambda v^*,$$

then u and v are called the right and left eigenvectors of A, corresponding to λ, respectively.

For a given Hermitian matrix $A = B + \iota C$, where $\iota = \sqrt{-1}$, $B,\ C \in \mathbb{R}^{n\times n}$ with $B^\top = B$ and $C^\top = -C$, it is well known that there exist a unitary matrix $Q \in \mathbb{C}^{n\times n}$ and a diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$ such that

$$A = Q\Lambda Q^*, \quad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n).$$

We observe the right and left eigenvectors of the Hermitian matrix A corresponding to λ are the same. If we rewrite $Q = U + \iota V$ with $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times n}$, then we obtain

$$\begin{pmatrix} B & -C \\ C & B \end{pmatrix} \begin{pmatrix} U & -V \\ V & U \end{pmatrix} = \begin{pmatrix} U & -V \\ V & U \end{pmatrix} \begin{pmatrix} \Lambda & 0 \\ 0 & \Lambda \end{pmatrix},$$

and

$$\begin{aligned} \begin{pmatrix} U & -V \\ V & U \end{pmatrix} \begin{pmatrix} U^\top & V^\top \\ -V^\top & U^\top \end{pmatrix} &= \begin{pmatrix} U^\top & V^\top \\ -V^\top & U^\top \end{pmatrix} \begin{pmatrix} U & -V \\ V & U \end{pmatrix} \\ &= \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}. \end{aligned}$$

For a given eigenpair (λ, z) of the Hermitian matrix A, we have

$$\begin{pmatrix} B & -C \\ C & B \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} x \\ y \end{pmatrix},$$

and

$$\begin{pmatrix} B & -C \\ C & B \end{pmatrix} \begin{pmatrix} -y \\ x \end{pmatrix} = \lambda \begin{pmatrix} -y \\ x \end{pmatrix},$$

where the vector z can be written as $x + \iota y$ with $x, y \in \mathbb{R}^n$.

Suppose that

$$\mathcal{A} = \begin{pmatrix} B & -C \\ C & B \end{pmatrix},$$

since $\mathcal{A} \in \mathbb{R}^{2n\times 2n}$ is symmetric, we can compute the eigenpairs of an Hermitian matrix A by the eigenpairs of the associated symmetric matrix $\mathcal{A}$.

We notice that it is required to convert a complex-valued eigenvalue problem into a real-valued one by splitting the complex numbers into their real and imaginary parts. However, the major disadvantage of this method is that the resulting algorithm will double the dimension compared with the original problem and may break the special data structure. Moreover, they will suffer from high computational complexity and have a slow convergence when the problem size is large.

B. Goals and organization

The neural network approach to parallel computing and signal processing has been successfully demonstrated through a variety of neurodynamic model with learning capabilities [15], [16], [17]. Many scholars have developed numerical methods based on neural network approach to compute eigenvectors of the symmetric matrices [18], [19], [20], [21] and solve linear matrix equations or nonlinear matrix equations on the basis of recurrent neural networks [22], [23], [24], [25]. Moreover, by the neural networks for computing eigenvectors of the symmetric matrices, Tang and Li [26], Liu, You and Cao [27], [28] introduce neural networks for extracting the eigenvectors of the real skew-symmetric matrices corresponding to the the maximal or minimal modulus eigenvalues. Liu, You and Cao [29], [30] also give recurrent neural networks to extract some eigenpairs of a general real matrix. Zhang and Leuang [31] propose a dynamic system for solving the eigenvalue problem with complex matrices.

In the above subsections, we can convert the eigenpairs of Hermitian matrices to the eigenvalue problem with an associated symmetric matrix. Then, the eigenvectors of the symmetric matrix $\mathcal{A}$ can be computed by the following proposed neural network model:

$$\frac{ds(t)}{dt} = [s(t)^\top s(t)]\mathcal{A}s(t) - [s(t)^\top \mathcal{A}s(t)]s(t), \qquad (1)$$

for $t \geq 0$, where $s = (s_1, s_2, \ldots, s_{2n})^\top \in \mathbb{R}^{2n}$ represents the state of the network.

However, the network (1) is summarized by $2n$-dimensional ordinary differential equations. We note that computation of $2n$-dimensional ordinary differential equation inevitable adds much workload and is computationally inefficient. Thus, in order to avoid redundant computation in a double real-valued space and reduce a low model complexity and storage capacity, we need to design the proposed complex-valued neural dynamical approach for computing the eigenvectors of Hermitian matrices and the Takagi vectors of complex symmetric matrices. We can refer to [32], [33], [34], [35], [36] for solving a complex-valued nonlinear convex programming problem by a complex-valued neural network.

The dynamics of the complex-valued neural network model for computing the eigenvectors of an Hermitian matrix $A \in \mathbb{C}^{n\times n}$ is described by

$$\frac{dz(t)}{dt} = -z(t) + f(z(t)), \qquad (2)$$

for $t \geq 0$, where

$$f(z(t)) = [(z^*z)A + (1 - z^*Az)]z$$

and $z = (z_1, z_2, \ldots, z_n)^\top \in \mathbb{C}^n$ represents the state of the network.

Through some basic operations, we perform the above network for computing the eigenvectors of skew-symmetric matrices and skew-Hermitian matrices.

The rest of our paper is organized as follows. In Section II, we establish the relationship between the solutions of the network (2) and the eigenvectors of Hermitian matrices. We obtain an explicit expressions for the solutions of the network (2) and derive its convergence properties. We use the network (2) for computing the eigenvectors of skew-symmetric and skew-Hermitian matrices in Section III. We illustrate the theoretical results via computer simulations in Section IV. Finally, we conclude our paper in Section V.

II. Hermitian matrices

In this section, we analyze the equilibrium points of the network (2) and derive an explicit representation for the solutions of the network (2). We show that a nonzero solution of the network (2) converges to an eigenvector of the associated Hermitian matrix. Some applications for the network (2) are explored.

A. Properties

We utilize the following lemma to illustrate a basic property for the solutions of the network (2).

Lemma II.1. *If the nonzero vector $z(t)$ is a solution of the network* (2) *for $t \geq 0$, then $z(t)^*z(t)$ keeps invariant for $t \geq 0$, i.e., $z(t)^*z(t) = z(0)^*z(0)$.*

Proof: Since the matrix A is Hermitian, then z^*Az is real for all vectors $z \in \mathbb{C}^n$.

Let $z(t) = x(t) + \imath y(t)$. Then, we have

$$z(t)^*z(t) = x(t)^\top x(t) + y(t)^\top y(t).$$

Meanwhile, the network (3) can be rewritten as

$$\frac{ds(t)}{dt} = [s(t)^\top s(t)]\mathcal{A}s(t) - [s(t)^\top \mathcal{A}s(t)]s(t),$$

where $s(t) = (x(t)^\top, y(t)^\top)^\top$ and the matrix

$$\mathcal{A} = \begin{pmatrix} B & -C \\ C & B \end{pmatrix}.$$

If the nonzero vector $s(t)$ is a solution of the above network for $t \geq 0$, then the derivative of $s(t)^\top s(t)$ is

$$\begin{aligned} [s(t)^\top s(t)]' &= 2s(t)^\top (s(t))' \\ &= 2s(t)^\top [s(t)^\top s(t)\mathcal{A}s(t) - s(t)^\top \mathcal{A}s(t)s(t)] \\ &= 0. \end{aligned}$$

Hence, $s(t)^\top s(t)$ keeps invariant for $t \geq 0$. The proof is complete. ■

A vector $u \in \mathbb{C}^n$ is called an *equilibrium* point of the network (2) if and only if it satisfies

$$-u + f(u) = 0, \qquad (3)$$

that is, $(u^*u)Au - (u^*Au)u = 0$.

Denote $\mathbb{E}$ by the set of all equilibrium points of the network (2). For a given eigenvalue λ of an Hermitian matrix $A \in \mathbb{C}^{n\times n}$, denote $\mathbb{V}_\lambda$ by the invariant eigenspace corresponding to λ.

Theorem II.1. *Suppose that $A \in \mathbb{C}^{n\times n}$ is Hermitian. The set of all the equilibrium points of* (2) *is equal to the union of all invariant subspaces of A, i.e.,*

$$\mathbb{E} = \cup_\lambda \mathbb{V}_\lambda.$$

Proof: Given any $u \in \mathbb{V}_\lambda$, if $u = 0$, clearly, $u \in \mathbb{E}$. Suppose that u is nonzero, there exists a scalar λ such that $u \in \mathbb{V}_\lambda$, i.e., $Au = \lambda u$ and $u^*A = \lambda u^*$. Thus,

$$(u^*u)Au - (u^*Au)u = (u^*u)\lambda u - \lambda(u^*u)u = 0.$$

By equation (3), we have $u \in \mathbb{E}$. This implies $\mathbb{E} \supseteq \cup_\lambda \mathbb{V}_\lambda$.

On the other hand, for any $u \in \mathbb{E}$, it holds that $(u^* u)Au - (u^* Au)u = 0$. If $u = 0$, then $u \in \mathbb{V}_\lambda$. If u is nonzero, then it is obvious that

$$Au = \frac{u^* Au}{u^* u} u,$$

which shows that u is an eigenvector of A. It follows that

$$\mathbb{E} \subseteq \cup_\lambda \mathbb{V}_\lambda.$$

Thus, $\mathbb{E} = \cup_\lambda \mathbb{V}_\lambda$. This completes the proof. ■

The above theorem shows that any equilibrium state of network (3) is an eigenvector of A if it is not the zero vectors. If the convergence of the network can be proved, then it can provide a method to compute the eigenvectors of A. Next we consider how to derive an explicit representation for the solutions of the network (3).

Since A is an Hermitian matrix, there exists a group of orthogonal basis of $\mathbb{C}^n$ composed by eigenvectors of $\mathbb{C}^n$. Let λ_i $(i = 1, 2, \ldots, n)$ be eigenvalues of A and z_i $(i = 1, 2, \ldots, n)$ be the corresponding eigenvectors that compose an orthonormal basis of A. Then, for any $z \in \mathbb{C}^n$, it can be represented as $z = \sum_{i=1}^n (\alpha_i + \iota\beta_i) z_i$, where α_i and β_i $(i = 1, 2, \ldots, n)$ are some real constants. Meanwhile, we have that $\|z\|_2^2 = \sum_{i=1}^n (\alpha_i^2 + \beta_i^2)$.

Suppose that $z_i = x_i + \iota y_i$ and let

$$s_i = \begin{cases} (x_i^\top, y_i^\top)^\top, & i = 1, 2, \ldots, n, \\ (-y_i^\top, x_i^\top)^\top, & i = n+1, n+2, \ldots, 2n. \end{cases}$$

For any $z \in \mathbb{C}^n$, there exists a unique vector $s \in \mathbb{R}^{2n}$, such that

$$s = \sum_{i=1}^n \alpha_i s_i + \sum_{i=n+1}^{2n} \beta_{i-n} s_i.$$

Since $\|z_i\|_2 = 1$, the set $\{s_1, s_2, \ldots, s_{2n}\}$ composes an orthonormal basis of $\mathbb{R}^{2n}$. All vectors s_i $(i = 1, 2, \ldots, 2n)$ are the eigenvectors of $\mathcal{A}$, corresponding to the eigenvalues

$$\lambda_1, \lambda_2, \ldots, \lambda_n; \lambda_1, \lambda_2, \ldots, \lambda_n,$$

respectively.

Theorem II.2. *Let $A \in \mathbb{C}^{n \times n}$ be Hermitian with the eigenvectors $z_i \in \mathbb{C}^n$. Given any nonzero vector $z(0) \in \mathbb{C}^n$, if there exist $2n$ real scalars $\alpha_i(0)$ and $\beta_i(0)$ $(i = 1, 2, \ldots, n)$ such that $z(0) = \sum_{i=1}^n c_i(0) z_i$, where there exists an integer k such that $\alpha_k(0) \neq 0$ or $\beta_k(0) \neq 0$, then the solution of the network (2) starting from $z(0)$ can be represented as*

$$z(t) = \sum_{i=1}^n \sqrt{\frac{\|z(0)\|_2^2}{\sum_{j=1}^n \|c_j(0)\|_2^2 \exp\left[2\|z(0)\|_2^2 (\lambda_j - \lambda_i) t\right]}} c_i(0) z_i, \tag{4}$$

for all $t \geq 0$, where $c_j(0) = \alpha_j(0) + \iota\beta_j(0)$ $(j = 1, 2, \ldots, n)$.

Proof: Define two vectors $b, a(0) \in \mathbb{R}^{2n}$ as

$$b_i = \begin{cases} \lambda_i, & i = 1, 2, \ldots, n, \\ \lambda_{i-n}, & i = n+1, n+2, \ldots, 2n, \end{cases}$$

and

$$a_i(0) = \begin{cases} \alpha_i(0), & i = 1, 2, \ldots, n, \\ \beta_{i-n}(0), & i = n+1, n+2, \ldots, 2n. \end{cases}$$

Since $z(0)$ is a nonzero vector and $z_i = x_i + \iota y_i$, we have

$$\begin{aligned} z(0) &= \sum_{i=1}^n c_i(0) z_i = \sum_{i=1}^n [\alpha_i(0) + \iota\beta_i(0)] z_i \\ &= \sum_{i=1}^n [\alpha_i(0) x_i - \beta_i(0) y_i] + \iota[\alpha_i(0) y_i + \beta_i(0) x_i]. \end{aligned}$$

The vector $s(0)$ can be represented as

$$s(0) \quad = \quad \sum_{i=1}^n \alpha_i(0) s_i + \sum_{i=n+1}^{2n} \beta_{k-n}(0) s_i = \sum_{i=1}^{2n} a_i(0) s_i.$$

Here, we know that $s(0)$ is a nonzero vector. On the basis of [21], the representation of a solution of the network (1) is

$$s(t) = \sum_{i=1}^{2n} \sqrt{\frac{s(0)^\top s(0)}{\sum_{j=1}^{2n} a_j^2(0) \exp[2 s(0)^\top s(0)(b_j - b_i)t]}} a_i(0) s_i, \tag{5}$$

for $t \geq 0$.

Now, we shall consider how to derive the representation of a solution of (4) based on (5). First, we can rewrite $s(t)$ as

$$\begin{aligned} s(t) &= \sum_{i=1}^n \sqrt{\frac{s(0)^\top s(0)}{\sum_{j=1}^{2n} a_j^2(0) \exp[2 s(0)^\top s(0)(b_j - b_i)t]}} \alpha_i(0) s_i \\ &+ \sum_{i=n+1}^{2n} \sqrt{\frac{s(0)^\top s(0)}{\sum_{j=1}^{2n} a_j^2(0) \exp[2 s(0)^\top s(0)(b_j - b_i)t]}} \beta_{i-n}(0) s_i. \end{aligned}$$

Since $z(0)^* z(0) = s(0)^\top s(0) = \sum_{i=1}^n (\alpha_i(0)^2 + \beta_i(0)^2)$ holds,

$$\begin{aligned} s(t) &= \sum_{i=1}^n \sqrt{\frac{\|z(0)\|_2^2}{\sum_{j=1}^{2n} a_j^2(0) \exp[2\|z(0)\|_2^2 (b_j - b_i)t]}} \alpha_i(0) s_i \\ &+ \sum_{i=n+1}^{2n} \sqrt{\frac{\|z(0)\|_2^2}{\sum_{j=1}^{2n} a_j^2(0) \exp[2\|z(0)\|_2^2 (b_j - b_i)t]}} \beta_{i-n}(0) s_i. \end{aligned}$$

Moveover

$$\begin{aligned} &\sum_{j=1}^{2n} a_j^2(0) \exp[2\|z(0)\|_2^2 (b_j - b_i)t] \\ &= \sum_{j=1}^n \alpha_j^2(0) \exp[2\|z(0)\|_2^2 (b_j - b_i)t] \\ &\quad + \sum_{j=n+1}^{2n} \beta_{j-n}^2(0) \exp[2\|z(0)\|_2^2 (b_j - b_i)t. \end{aligned}$$

If $i = 1, 2, \ldots, n$, then

$$\begin{aligned} &\sum_{j=1}^n \alpha_j^2(0) \exp[2\|z(0)\|_2^2 (b_j - b_i)t] \\ &= \sum_{j=1}^n \alpha_j^2(0) \exp[2\|z(0)\|_2^2 (\lambda_j - \lambda_i)t)]. \end{aligned}$$

Hence, we have

$$\begin{aligned}&\sum_{j=1}^{2n} a_j^2(0)\exp[2\|z(0)\|_2^2(b_j-b_i)t]\\&=\sum_{j=1}^{n}[\alpha_j^2(0)+\beta_j^2(0)]\exp[2\|z(0)\|_2^2(\lambda_j-\lambda_i)t],\end{aligned} \tag{6}$$

If $i=n+1,n+2,\dots,2n$, then

$$\begin{aligned}&\sum_{j=1}^{2n} a_j^2(0)\exp[\|z(0)\|_2^2(b_j-b_i)t]\\&=\sum_{j=1}^{2n} a_j^2(0)\exp[2\|z(0)\|_2^2(b_j-\lambda_{i-n})t]\\&=\sum_{j=1}^{n} \alpha_j^2(0)\exp[2\|z(0)\|_2^2(b_j-\lambda_{i-n})t]\\&\quad+\sum_{j=n+1}^{2n} \beta_{j-n}^2(0)\exp[2\|z(0)\|_2^2(\lambda_{j-n}-\lambda_{i-n})t]\\&=\sum_{j=n+1}^{2n}[\alpha_{j-n}^2(0)+\beta_{j-n}^2(0)]\exp[2\|z(0)\|_2^2(\lambda_{j-n}-\lambda_{i-n})t]\\&=\sum_{j=1}^{n}[\alpha_j^2(0)+\beta_j^2(0)]\exp[2\|z(0)\|_2^2(\lambda_j-\lambda_i)t].\end{aligned}$$

Hence, the solution of (2) starting from $z(0)$ can be represented as

$$z(t)=\sum_{i=1}^{n}\sqrt{\frac{\|z(0)\|_2^2}{\sum_{j=1}^{n}\|c_j(0)\|_2^2\exp[2\|z(0)\|_2^2(\lambda_j-\lambda_i)t]}}c_i(0)z_i,$$

for $t\geq 0$. This completes the proof. ■

Theorem II.2 shows the solutions of network (2) can be represented in terms of the set of all orthogonal eigenvectors. This property will be quite convenient for analyzing the convergence of the network (2) in Section II-B.

B. Convergence analysis

In this subsection, convergence of the network (2) will be analyzed. For an artificial neural network of continuous model, the convergence property is crucial. From Theorem II.2, the convergence property can be developed on a sound foundation.

Theorem II.3. *Each solution of* (2) *starting from any nonzero points in* $\mathbb{C}^n$ *converges to an eigenvector of any Hermitian matrix* $A\in\mathbb{C}^{n\times n}$.

Before we obtain a theorem that gives the necessary and sufficient conditions for the network (2) to converge to the eigenvectors of Hermitian matrices $A\in\mathbb{C}^{n\times n}$, corresponding to the maximal eigenvalue, some notations will be listed.

Suppose that $A\in\mathbb{C}^{n\times n}$ is Hermitian. Let λ_i $(i=1,2,\dots,n)$ be all eigenvalues of A with $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_n$, and z_i $(i=1,2,\dots,n)$ be the corresponding eigenvectors that compose a group of orthogonal basis of $\mathbb{C}^n$. Let δ_i $(i=1,2,\dots,m,\ m\leq n)$ be all the distinct eigenvalues of A ordered by $\delta_1\geq\delta_2\geq\cdots\geq\delta_m$. For any i $(1\leq i\leq m)$, denote the algebraic sum of the multiplicity of δ_i $(i=1,2,\dots,m)$ by k_i. Clearly, $k_m=n$. For convenience, denote $k_0=1$. It is easy to see that $\lambda_i=\delta_r$, and $z_i\in\mathbb{V}_{\delta_r}$, for all $i\in\{k_{r-1},k_{r-1}+1,\dots,k_r\}$.

Theorem II.4. *Suppose that* $A\in\mathbb{C}^{n\times n}$ *is Hermitian. Let* $z(0)\in\mathbb{C}^n$ *be a nonzero vector, if* $z(0)$ *is not orthogonal to* $\mathbb{V}_{\delta_1}$*, then the solution of the network* (2) *starting from* $z(0)$ *converges to an eigenvector of* A*, corresponding to the maximal eigenvalue.*

According to the above theorem, we have the following remark for illustrating the maximal eigenvalue of an Hermitian matrix $A\in\mathbb{C}^{n\times n}$.

Remark II.1. *Under the conditions of Theorem* II.4*, the value* $\frac{z^*(t)Az(t)}{z^*(t)z(t)}$ *converges to the maximal eigenvalue* λ_1*, where* $z(t)$ *is the solution of the network* (2) *starting from some nonzero vectors* $z(0)\in\mathbb{C}^n$.

These conditions in Theorem II.4 depend on the eigenspace $\mathbb{V}_{\delta_1}$. However, the space $\mathbb{V}_{\delta_1}$ is not known a prior, it is not practical to choose initial values that satisfy Theorem II.4 in advance. This problem can be solved by the initial values under some random perturbations, since the dimension of $\mathbb{V}_{\delta_1}$ is always less than the dimension of $\mathbb{C}^n$. In this way, one can obtain a high probability of having initial values satisfying the conditions in Theorem II.4.

Theorem II.4 shows how to compute the eigenvectors of an Hermitian matrix $A\in\mathbb{C}^{n\times n}$, corresponding to the maximal eigenvalue. Under some operations, we can also compute the eigenvectors of the matrix A, associated with the minimal eigenvalue.

Theorem II.5. *Replacing* A *in network* (2) *with* $-A$*, and suppose that* $z(0)$ *is a nonzero vector in* $\mathbb{C}^n$ *which is not orthogonal to* $\mathbb{V}_{\delta_m}$*, then the solution starting from* $z(0)$ *converges to an eigenvector of the matrix* A*, corresponding to the minimal eigenvalue.*

Remark II.2. *Under the conditions of Theorem* II.5*, the value* $\frac{z^*(t)Az(t)}{z^*(t)z(t)}$ *converges to* λ_n*, where* $z(t)$ *is the solution of* (2) *starting from the nonzero vector* $z(0)$.

More general, we have the following theorem to compute the eigenvectors of an Hermitian matrix $A\in\mathbb{C}^{n\times n}$, corresponding to the other eigenvalues (neither the maximal nor the minimal eigenvalues), by the network (2).

Theorem II.6. *Suppose that* $A\in\mathbb{C}^{n\times n}$ *is Hermitian and* z_i *are its eigenvectors with* $i=1,2,\dots,n$*. If* $z(0)$ *is orthogonal to each* z_i *with* $i=1,2,\dots,m$ *and* $m\leq n$*, then the solution of the network* (2) *starting from* $z(0)$ *converges to an eigenvector of the matrix* A*, which is also orthogonal to each* z_i *with* $i=1,2,\dots,m$.

According to the explicit representation for the solutions of the network (2) and the content in [21], it is easy to prove Theorems II.4, II.5 and II.6. Here, we omit the proof.

III. Applications for the network (2)

In this section, we investigate how to generalize the network (2) for computing the eigenvectors of the skew-symmetric and skew-Hermitian matrices, corresponding to the maximal (or the minimal) imaginary part eigenvalue.

A. The eigenvalue problem with skew-symmetric matrices

A matrix $A\in\mathbb{R}^{n\times n}$ is skew-symmetric if it satisfies the relation $A^\top=-A$. Let $A\in\mathbb{R}^{n\times n}$ be a skew-symmetric. It is obvious to see that

$$(-\iota A)^*=-\iota A,\quad (\iota A)^*=\iota A.$$

Therefore, $-\iota A$ is an Hermitian matrix. The properties for the eigenvalues of the matrices A [37] are listed as follows:

(1) All eigenvalues of the matrix A are zero or pure imaginary numbers;
(2) If the pair $(\iota\lambda, u)$ is an eigenpair of the matrix A, then the $(-\iota\lambda, \overline{u})$ is another eigenpair of the matrix A, where $\lambda \in \mathbb{R}$;
(3) If the pair $(\iota\lambda, u)$ is an eigenpair of the matrix A, then the pair (λ, u) is an eigenpair of the matrix $-\iota A$ and the pair $(-\lambda, u)$ is an eigenpair of the matrix ιA, where $\lambda \in \mathbb{R}$.

When we compute the eigenvectors of the matrix $-\iota A$, corresponding to the maximal eigenvalue λ, by the network (2), it is easy to obtain the eigenvectors of the matrix A, associated with the maximal imaginary part eigenvalue $\iota\lambda$.

B. The eigenvalue problem with skew-Hermitian matrices

A matrix $A \in \mathbb{C}^{n\times n}$ is skew-Hermitian if it satisfies the relation $A^* = -A$ [38]. We list the properties [38] of a skew-Hermitian matrix $A \in \mathbb{C}^{n\times n}$ as follows:

(1) All eigenvalues of the matrix A are zero or pure imaginary numbers;
(2) Both ιA and $-\iota A$ are Hermitian;
(3) If the pair $(\iota\lambda, u)$ is an eigenpair of the matrix A, then the pair (λ, u) is an eigenpair of the matrix $-\iota A$ and the pair $(-\lambda, u)$ is an eigenpair of the matrix ιA, where $\lambda \in \mathbb{R}$.

If we compute the eigenvectors of the matrix $-\iota A$, corresponding to the maximal eigenvalue λ based on the network (2), it is easy to obtain the eigenvectors of the skew-Hermitian matrix A, corresponding to the maximal imaginary part eigenvalue $\iota\lambda$.

IV. NUMERICAL EXAMPLES

In this section, some computer simulation results are given to illustrate our theory. All computations are carried out in Matlab Version 2013a, which has a unit roundoff $2^{-53} \approx 1.1 \times 10^{-16}$, on a laptop with Intel Core i5-3470M CPU (3.20GHz) and 4GB RAM. All floating point numbers have four digits. Suppose that $\epsilon = 1e - 10$.

In order to compute the eigenvectors of an Hermitian matrix $A \in \mathbb{C}^{n\times n}$, we use the following difference equations to approximate the network (2):

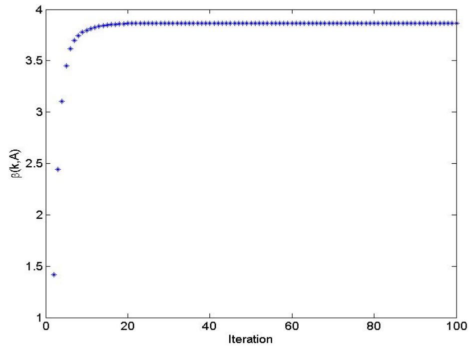
$$z(k+1) = z(k) + \alpha[z(k)^* z(k) A z(k) - z(k)^* A z(k) z(k)],$$

where α is a learning rate and $z(0) \in \mathbb{C}^n$ is any nonzero vectors.

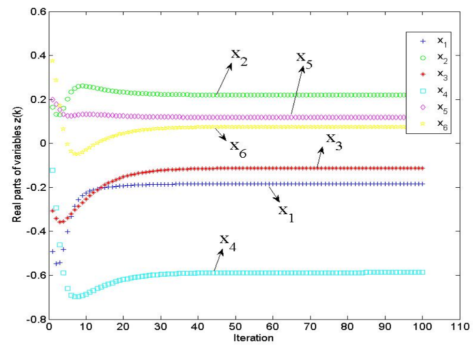
For the pair $(\beta(k, A), z(k))$ with $\beta(k, A) = \frac{z(k)^* A z(k)}{z(k)^* z(k)}$, we define

$$\mathrm{ERR}(k) := \frac{\|Az(k) - \beta(k, A) z(k)\|_2}{\||A| \cdot |z(k)| + |\beta(k, A)| \cdot |z(k)|\|_2}.$$

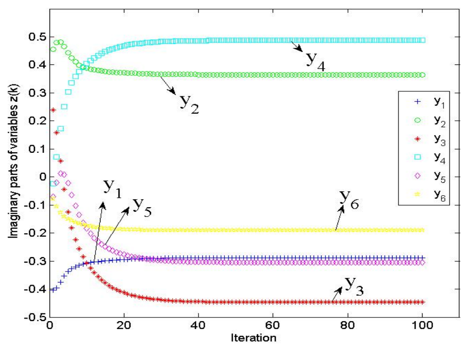
If there exists a positive integer k_0 such that $\mathrm{ERR}(k) \leq \epsilon$ for all $k \geq k_0$, then the pair $(\beta(k, A), z(k))$ is an approximate eigenpair of the matrix A. According to Theorem II.4, we have that $\beta(k, A)$ is the approximation for the maximal eigenvalue of the matrix A.



(a) The maximal eigenvalue.



(b) The real parts of variable.



(c) The imaginary parts of variable.

Fig. 1. Illustrate the solutions of the network (2) converges to an eigenvectors of the matrix A, corresponding to the maximal eigenvalue.

Example IV.1. *We test the Hermitian matrix A from* [39]*:*

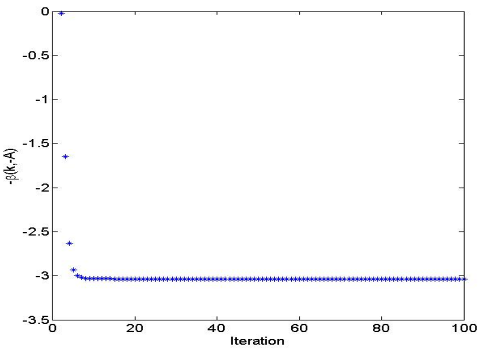
$$A = \begin{pmatrix} -0.1 & -0.6+0.7\iota & 0.8+0.2\iota & -0.3+1.2\iota & -0.3+0.3\iota & -0.1+0.1\iota \\ -0.6-0.7\iota & -0.5 & -0.8+0.3\iota & -0.3-1.7\iota & 0.1-0.5\iota & -1+0.2\iota \\ 0.8-0.2\iota & -0.8-0.3\iota & 1.6 & 0.1-0.1\iota & 0.9 & 1.2-0.9\iota \\ -0.3-1.2\iota & -0.3+1.7\iota & 0.1+0.1\iota & 1.9 & -1.5-0.3\iota & 0.3\iota \\ -0.3-0.3\iota & 0.1+0.5\iota & 0.9 & -1.5+0.3\iota & 0.4 & 0.1-0.2\iota \\ -0.1-0.1\iota & -1-0.2\iota & 1.2+0.9\iota & -0.3\iota & 0.1+0.2\iota & -0.5 \end{pmatrix}.$$

We compute the approximate maximal and minimal eigenvalues of A by the 'eig' function in Matlab as $\lambda = 3.8673$

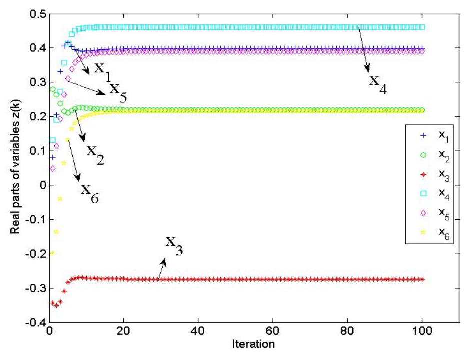
and $\lambda = -3.0348$, respectively.

We set $z = x + \iota y$ with $x, y \in \mathbb{R}^6$. We can use the network (2) with a random initial $z(0)$ to find an eigenvector of the matrix A, corresponding to the approximate maximal eigenvalue $\lambda = 3.8673$, the results about the trajectories for $\beta(k, A)$, and the real part and imaginary part of $z(k)$ are shown in Figure 1.

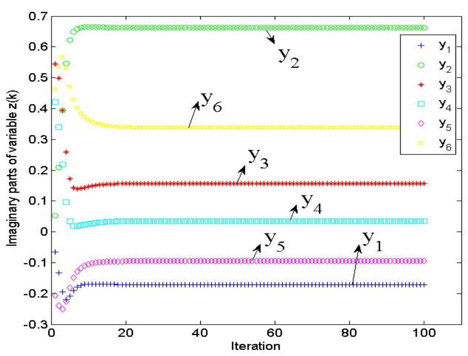
On the other hand, we set the temporary matrix B as $B = -A$. Notice that λ is an eigenvalue of the matrix A implies that $-\lambda$ is an eigenvalue of B, and vice versa. Hence, we can compute the eigenvectors of A, corresponding to the minimal eigenvalue, employing the network (2) to the matrix B. The trajectories of $-\beta(k, -A)$, and the real part and imaginary part of $z(k)$ are shown in Figure 2.



(a) The minimal eigenvalue.



(b) The real parts of variable.



(c) The imaginary parts of variable.

Fig. 2. Illustrate the solutions of the network (2) converges to an eigenvectors of the matrix A, corresponding to the minimal eigenvalue.

Example IV.2. *The testing matrix A is adopted from Tang and Li* [26] *and its entries are given by:*

$$A = \begin{pmatrix} 0 & 0.6837 & 0.0194 & 0.1856 & 0.0885 & 0.7982 & -0.4444 \\ -0.6837 & 0 & 0.3056 & 0.0305 & -0.1352 & 0.0307 & 0.0248 \\ -0.0194 & -0.3056 & 0 & 0.2671 & 0.3347 & 0.6079 & 0.5217 \\ -0.1856 & -0.0305 & -0.2671 & 0 & -0.5776 & 0.4037 & 0.7374 \\ -0.0885 & 0.1352 & -0.3347 & 0.5776 & 0 & -0.2048 & -0.4198 \\ -0.7982 & -0.0307 & -0.6079 & -0.4037 & 0.2048 & 0 & 0.4930 \\ 0.4444 & -0.0248 & -0.5217 & -0.7374 & 0.4198 & -0.4930 & 0 \end{pmatrix}.$$

The set of the approximate eigenvalues of A is $\{\pm 1.5195\iota, \pm 0.9850\iota, \pm 0.7115\iota, 0\}$, computed by the 'eig' function in Matlab.

Let $B = -\iota A$, then B is an Hermitian matrix. In term of the analysis in Section 2.3, the approximate eigenvalues of B are ± 1.5195, ± 0.9850, ± 0.7115 and 0 with the same eigenspaces, respectively.

If we employ the network (2) to compute the eigenvectors of B, associated with the maximal eigenvalue, we can derive the eigenvectors of A, corresponding to the maximal imaginary part eigenvalue. The trajectories for $\beta(k, -\iota A)$, and the real part and imaginary part of $z(k)$ are shown in Figure 3.

We obtain an eigenvector of A, associated with the minimal imaginary part eigenvalue, if we apply the network (2) to ιA. The results about the trajectories for $-\beta(k, \iota A)$, and the real part and imaginary part of $z(k)$ are shown in Figure 4.

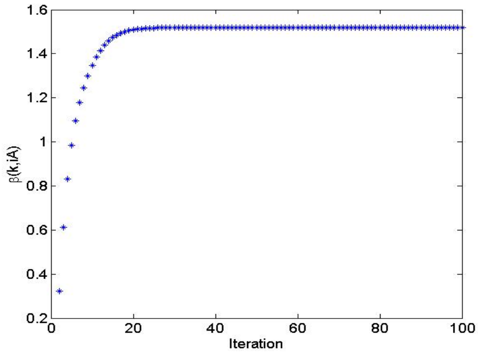
Example IV.3. *We choose the skew-Hermitian matrix from* [40]*:*

$$A = 10^2 \begin{pmatrix} 0 + 1.8193\iota & 0.0041 & -0.0067 & -0.0072 & 0.0012 & 0.0068 \\ -0.0041 & 0 + 1.8178\iota & -0.0073 & 0.0009 & 0.0036 & 0.0003 \\ 0.0067 & 0.0073 & 0 + 1.8084\iota & 0.0025 & -0.0072 & -0.0001 \\ 0.0072 & -0.0009 & -0.0025 & 0 + 1.8082\iota & -0.0028 & 0.0051 \\ -0.0012 & -0.0036 & 0.0072 & 0.0028 & 0 + 1.8197\iota & -0.0063 \\ -0.0068 & -0.0003 & 0.0001 & -0.0051 & 0.0063 & 0 + 1.8002\iota \end{pmatrix}.$$

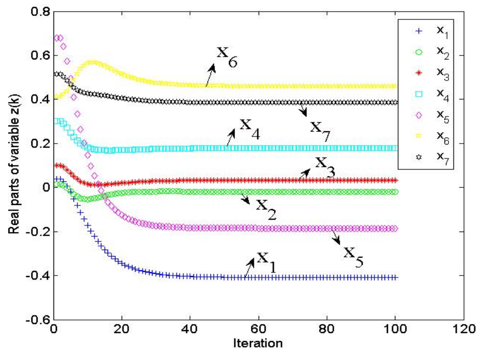
The set of the approximate eigenvalues of A is $\{1.8319e + 2\iota, 1.7921e + 2\iota, 1.8212e + 2\iota, 1.8186e + 2\iota, 1.8071e + 2\iota, 1.8026e + 2\iota\}$, computed by the 'eig' function in Matlab.

Let $B = -\iota A$, then B is an Hermitian matrix. According to the analysis in Section 2.3, the approximate eigenvalues of B are 183.19 and 179.21 with the same eigenspaces, respectively.

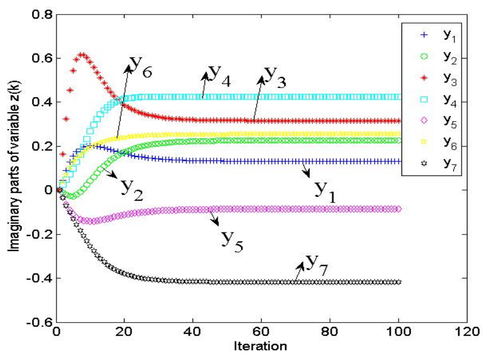
If we compute the eigenvectors of B, associated with the maximal eigenvalue, by using the network (2), we can



(a) The maximal imaginary part eigenvalue.

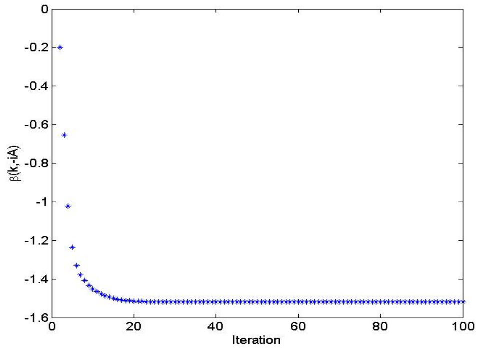


(b) The real parts of variable.

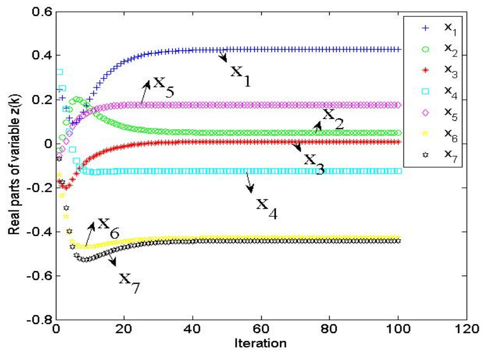


(c) The imaginary parts of variable.

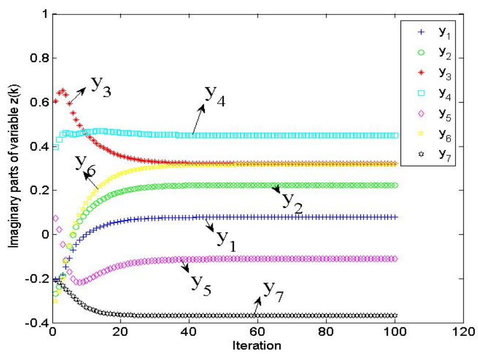
Fig. 3. Illustrate the solutions of the network (2) converges to an eigenvectors of the matrix A, corresponding to the maximal imaginary part eigenvalue.



(a) The minimal imaginary part eigenvalue.



(b) The real parts of variable.



(c) The imaginary parts of variable.

Fig. 4. Illustrate the solutions of the network (2) converges to an eigenvectors of the matrix A, corresponding to the minimal imaginary part eigenvalue.

derive the eigenvectors of A, corresponding to the maximal imaginary part eigenvalue. The trajectories for $\beta(k, -\iota A)$, and the real part and imaginary part of $z(k)$ are shown in Figure 5.

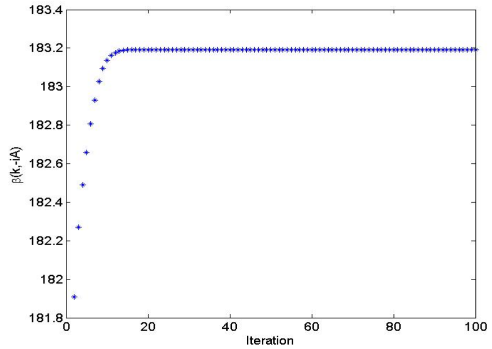
Also, we obtain an eigenvector of A, corresponding to the minimal imaginary part eigenvalue, by applying the network (2) to ιA. The results about the trajectories for $-\beta(k, \iota A)$, and the real part and imaginary part of $z(k)$ are shown in Figure 6.

V. Conclusion

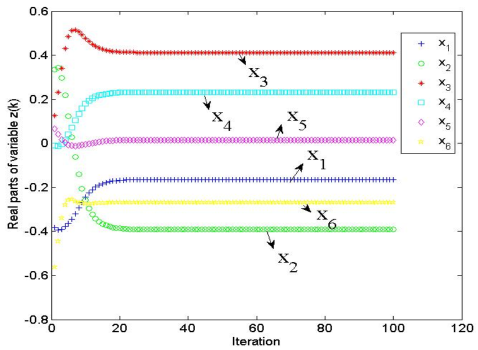
We present the complex-valued neural network for computing the eigenvectors of the Hermitian matrices, corresponding to the maximal and minimal eigenvalues. Considering the complex-valued neural network discussed in this paper is different from dealing with the general complex-valued neural networks [32], [33], [34], [35].

Here, we derive the explicit representations for the solutions of the proposed neural networks (2) according to the matrix structures, and analyze the convergence of the networks. Based on the network (2), we also consider two special cases of the eigenvalue problem, the eigenvalue problem with skew-symmetric matrices and skew-Hermitian matrices.

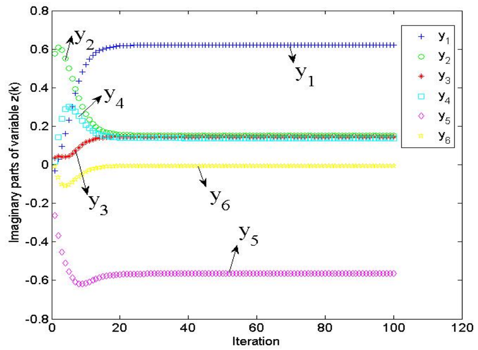
In addition, for numerical examples in Section IV, we shall point that we do not choose the optimal learning rate for finding the approximate eigenvectors. That is, we may need less iterative steps for achieving our goals in each computation.



(a) The maximal imaginary part eigenvalue.

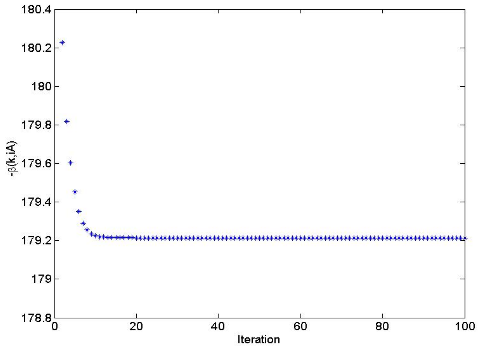


(b) The real parts of variable.

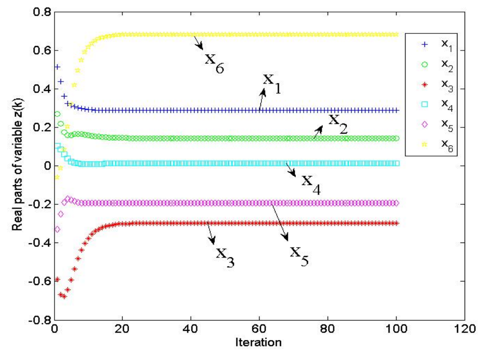


(c) The imaginary parts of variable.

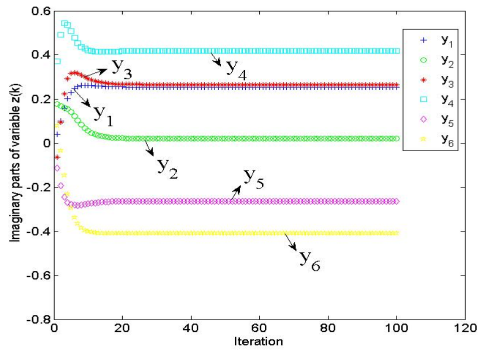
Fig. 5. Illustrate the solutions of the network (2) converges to an eigenvectors of the matrix A, corresponding to the maximal imaginary part eigenvalue.



(a) The minimal imaginary part eigenvalue.



(b) The real parts of variable.



(c) The imaginary parts of variable.

Fig. 6. Illustrate the solutions of the network (2) converges to an eigenvectors of the matrix A, corresponding to the minimal imaginary part eigenvalue.

REFERENCES

[1] S. Zhang, Y. Xia, and J. Wang, "A complex-valued projection neural network for constrained optimization of real functions in complex variables," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 3227–3238, 2015.

[2] S. Zhang, Y. Xia, and W. Zheng, "A complex-valued neural dynamical optimization approach and its stability analysis," *Neural Networks*, vol. 61, pp. 59–67, 2015.

[3] E. Y. Bobrovnikova and S. A. Vavasis, "A norm bound for projections with complex weights," *Linear Algebra and Its Applications*, vol. 307, no. 1C3, pp. 69–75, 2000.

[4] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[5] B. Loesch and B. Yang, "Cramer-Rao bound for circular complex independent component analysis," *IEEE Transactions on Signal Processing*, vol. 61, pp. 365–379, 2013.

[6] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–95, 2007.

[7] N. N. Vakhaniya and N. P. Kandelaki, "Random vectors with values in complex Hilbert spaces," *Theory of Probability and Its Applications*, vol. 41, no. 1, pp. 31–52, 1997.

[8] Y. Xia and G. Feng, "A neural network for robust LCMP beamforming," *Signal Processing*, vol. 86, no. 10, pp. 2901–2912, 2006.

[9] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, ser. Computer Science and Scientific Computing. Academic Press, Inc., Boston, MA, 1990.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., ser. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 2013.

[11] J. H. Wilkinson, *The Algebraic Eigenvalue Problem.* Oxford University Press, 1965.

[12] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, Eds., *Templates for the Solution of Algebraic Eigenvalue Problems. A Practical Guide*, ser. Software, Environments, and Tools. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000, vol. 11. [Online]. Available: http://dx.doi.org/10.1137/1.9780898719581

[13] J. W. Demmel, *Applied Numerical Linear Algebra.* Society for

Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. [Online]. Available: http://dx.doi.org/10.1137/1.9781611971446

[14] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, 2nd ed. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011.

[15] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences*, vol. 81, no. 10, pp. 3088–3092, 1984.

[16] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, 1987.

[17] Y. Pan and Y. Shi, "A grey neural network model optimized by fruit fly optimization algorithm for short-term traffic forecasting," *Engineering Letters*, vol. 25, no. 2, pp. 198–204, 2017.

[18] A. Cichocki and R. Unbehauen, "Neural networks for computing eigenvalues and eigenvectors," *Biological Cybernetics*, vol. 68, no. 2, pp. 155–164, 1992.

[19] Y. Liu, Z. You, and L. Cao, "A simple functional neural network for computing the largest and smallest eigenvalues and corresponding eigenvectors of a real symmetric matrix," *Neurocomputing*, vol. 67, pp. 369–383, 2005.

[20] N. Samardzija and R. Waterland, "A neural network for computing eigenvectors and eigenvalues," *Biological Cybernetics*, vol. 65, no. 4, pp. 211–214, 1991.

[21] Z. Yi, Y. Fu, and H. J. Tang, "Neural networks based approach for computing eigenvectors and eigenvalues of symmetric matrix," *Computers Mathematics with Applications*, vol. 47, no. 8-9, pp. 1155–1164, 2004. [Online]. Available: http://dx.doi.org/10.1016/S0898-1221(04)90110-1

[22] F. Ding and T. Chen, "Gradient based iterative algorithms for solving a class of matrix equations," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1216–1221, 2005.

[23] D. Guo, C. Yi, and Y. Zhang, "Zhang neural network versus gradient-based neural network for time-varying linear matrix equation solving," *Neurocomputing*, vol. 74, no. 17, pp. 3708–3712, 2011.

[24] B. Zhou and G. Duan, "An explicit solution to the matrix equation ax-xf = by," *Linear Algebra and its Applications*, vol. 402, pp. 345–366, 2005.

[25] D. Mishra and P. K. Kalra, "Modified hopfield neural network approach for solving nonlinear algebraic equations," *Engineering Letters*, vol. 14, no. 1, pp. 135–142, 2007.

[26] Y. Tang and J. Li, "Another neural network based approach for computing eigenvalues and eigenvectors of real skew-symmetric matrices," *Computers Mathematics with Applications*, vol. 60, no. 5, pp. 1385–1392, 2010.

[27] Y. Liu, Z. You, and L. Cao, "A functional neural network for computing the largest modulus eigenvalues and their corresponding eigenvectors of an anti-symmetric matrix," *Neurocomputing*, vol. 67, pp. 384–397, 2005.

[28] ——, "A concise functional neural network computing the largest modulus eigenvalues and their corresponding eigenvectors of a real skew matrix," *Theoretical Computer Science*, vol. 367, no. 3, pp. 273–285, 2006.

[29] ——, "A functional neural network computing some eigenvalues and eigenvectors of a special real matrix," *Neural Networks*, vol. 18, no. 10, pp. 1293–1300, 2005.

[30] ——, "A recurrent neural network computing the largest imaginary or real part of eigenvalues of real matrices," *Computers Mathematics with Applications*, vol. 53, no. 1, pp. 41–53, 2007.

[31] Q. Zhang and Y.-W. Leung, "Dynamic system for solving complex eigenvalue problems," *IEE Proceedings-Control Theory and Applications*, vol. 144, no. 5, pp. 455–458, 1997.

[32] I. Aizenberg, *Complex-Valued Neural Networks with Multi-Valued Neurons*, ser. Studies in Computational Intelligence. Springer-Verlag, Berlin, 2011, vol. 353. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20353-4

[33] M. Bohner, V. S. H. Rao, and S. Sanyal, "Global stability of complex-valued neural networks on time scales," *Differential Equations and Dynamical Systems*, vol. 19, no. 1-2, pp. 3–11, 2011.

[34] A. Hirose, *Complex-Valued Neural Networks: Theories and Applications*. World Scientific, 2003, vol. 5.

[35] J. Hu and J. Wang, "Global stability of complex-valued recurrent neural networks with time-delays," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 6, pp. 853–865, 2012.

[36] T. Nitta, *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*. Hershey, PA, USA, 2009.

[37] D. C. Youla, "A normal form for a matrix under the unitary congruence group," *Canadian Journal of Mathematics*, vol. 13, pp. 694–704, 1961.

[38] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge University Press, Cambridge, 2013.

[39] L. Dieci, A. Papini, and A. Pugliese, "Approximating coalescing points for eigenvalues of hermitian matrices of three parameters," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 2, pp. 519–541, 2013.

[40] M. Dehghan and M. Hajarian, "Two algorithms for finding the hermitian reflexive and skew-hermitian solutions of sylvester matrix equations," *Applied Mathematics Letters*, vol. 24, no. 4, pp. 444–449, 2011.