Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 53 (2007) 41–53
An International Journal
computers & mathematics with applications
www.elsevier.com/locate/camwa

A recurrent neural network computing the largest imaginary or real part of eigenvalues of real matrices

Yiguang Liu[a,*], Zhisheng You[a], Liping Cao[b]

[a] *Institute of Image and Graphics, School of Computer Science and Engineering, Sichuan University, Chengdu, 610064, PR China*
[b] *Sichuan University Library, Sichuan University, Chengdu, 610064, PR China*

Received 30 December 2005; received in revised form 8 September 2006; accepted 14 September 2006

Abstract

As the efficient calculation of eigenpairs of a matrix, especially, a general real matrix, is significant in engineering, and neural networks run asynchronously and can achieve high performance in calculation, this paper introduces a recurrent neural network (RNN) to extract some eigenpair. The RNN, whose connection weights are dependent upon the matrix, can be transformed into a complex differential system whose variable $z(t)$ is a complex vector. By the analytic expression of $|z(t)|^2$, the convergence properties of the RNN are analyzed in detail. With general nonzero initial complex vector, the RNN obtains the largest imaginary part of all eigenvalues. By a rearrangement of connection matrix, the largest real part is obtained. A practice of a 7×7 matrix indicates the validity of this method. Two matrices, whose dimensionalities are 50 and 100, respectively, are employed to test the efficiency of this approach when dimension number becomes large. The results imply that the iteration number at which the network enters into equilibrium state is not sensitive with dimensionality. This RNN can be used to estimate the largest modulus of eigenvalues, etc. Compared with other neural networks designed for the similar aims, this RNN is applicable to general real matrices.

Keywords: Recurrent neural network; Complex differential system; Real matrix; Eigenvalues; Imaginary part; Real part

1. Introduction

Using neural networks to compute eigenpairs has parallel, and quickness, virtues. Quick extraction of eigenpairs has many applications such as principal component analysis (PCA) [1] and real-time signal processing [2] etc. So, a lot of literature on this field exist [3–20]. Recently, Yi et al. [3] proposed a neural network-based approach to compute eigenvectors corresponding to the largest or smallest eigenvalue of any real symmetric matrix. Ref. [4] generalized the well-known Oja network to include nonsymmetrical matrices. Liu [5] meliorated the model in Ref. [3] and introduced a simpler model accomplishing analogous calculations. Ref. [6] introduced a RNN model computing the largest modulus eigenvalues and their corresponding eigenvectors of an anti-symmetric matrix. So the known results mainly focus on solving eigenpairs of real symmetric, or anti-symmetric, matrices. Although Ref. [4] involved

* Corresponding author. Tel.: +86 28 854 12565; fax: +86 28 854 10604.
E-mail address: lygpapers@yahoo.com.cn (Y. Liu).

0898-1221/$ - see front matter
doi:10.1016/j.camwa.2006.09.004

nonsymmetrical matrices, its section "The nonlinear homogeneous case" has the assumption of $\alpha(w) \in R$, which requires the matrix to occupy real eigenvalues. Following a search of the relevant literature, we were unable to locate a neural network-based method for calculating eigenvalues of a general real matrix, whether the eigenvalue is a real or a general complex number. This paper is an exploration in this direction, and proposes a recurrent neural network-based approach described by Eq. (1) to estimate the largest modulus of eigenvalues of a general real matrix.

$$\frac{\mathrm{d}v(t)}{\mathrm{d}t} = A'v(t) - |v(t)|^2 v(t), \tag{1}$$

where $t \geq 0$, $v(t) \in R^{2n}$, $A' = \begin{pmatrix} 0 & A \\ -A & 0 \end{pmatrix}$, A is the general real matrix requiring the calculation of eigenpairs. $|v(t)|$ denotes the modulus of $v(t)$. When $v(t)$ is seen as the states of neurons, $\left[A' - |v(t)|^2 U\right]$ (U denotes a suitable dimensional identity matrix) is looked at as synaptic connection weights, and the activation functions are assumed to be pure linear functions, Eq. (1) describes a continuous-time recurrent neural network. Under the realization of $\dot{v}(t) = A'v(t)$, using a calculation unit to obtain $|v(t)|^2$ and subtracting it from all diagonal entries of A', we can realize the RNN.

Let $v(t) = \left(x^{\mathrm{T}}(t), y^{\mathrm{T}}(t)\right)^{\mathrm{T}}$, $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^{\mathrm{T}} \in R^n$ and $y(t) = (y_1(t), y_2(t), \ldots, y_n(t))^{\mathrm{T}} \in R^n$. From Eq. (1), it follows that

$$\begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} = Ay(t) - \displaystyle\sum_{j=1}^{n} \left[x_j^2(t) + y_j^2(t)\right] x(t) \\ \dfrac{\mathrm{d}y(t)}{\mathrm{d}t} = -Ax(t) - \displaystyle\sum_{j=1}^{n} \left[x_j^2(t) + y_j^2(t)\right] y(t). \end{cases} \tag{2}$$

Let

$$z(t) = x(t) + \mathrm{i}y(t), \tag{3}$$

where i denotes the imaginary unit, it easily follows from Eq. (2) that

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} + \mathrm{i}\frac{\mathrm{d}y(t)}{\mathrm{d}t} = -A\mathrm{i}\left[x(t) + \mathrm{i}y(t)\right] - \sum_{j=1}^{n} \left[x_j^2(t) + y_j^2(t)\right] \left[x(t) + \mathrm{i}y(t)\right],$$

i.e.

$$\frac{\mathrm{d}z(t)}{\mathrm{d}t} = -Az(t)\mathrm{i} - z^{\mathrm{T}}(t)\,\bar{z}(t)z(t) \tag{4}$$

where $\bar{z}(t)$ denotes the complex conjugate vector of $z(t)$. Obviously, Eq. (4) is a complex differential system. A set of ordinary differential equations is just a model which may or may not approximate the real behavior of some neural network, although there are differences between them. In the following sections, we will discuss the convergence properties of Eq. (4) instead of RNN (1).

2. Analytic expression of $|z(t)|^2$

All eigenvalues of A are denoted as $\lambda_1^R + \lambda_1^I\mathrm{i}, \lambda_2^R + \lambda_2^I\mathrm{i}, \ldots, \lambda_n^R + \lambda_n^I\mathrm{i}$ ($\lambda_k^R, \lambda_k^I \in R, k = 1, 2, \ldots, n$), corresponding complex eigenvectors are denoted as $\mu_1, \ldots, \mu_n$.

With any general real matrix A, there are two cases for $\mu_1, \ldots, \mu_n$: (1) when rank of A is deficient, some of $\lambda_1^R + \lambda_1^I\mathrm{i}, \lambda_2^R + \lambda_2^I\mathrm{i}, \ldots, \lambda_n^R + \lambda_n^I\mathrm{i}$ may be zeros. When $\lambda_j^R + \lambda_j^I\mathrm{i} = 0$, u_j can be randomly chosen ensuring $\mu_1, \ldots, \mu_n$ construct a basis in $C^{n\times n}$; (2) A is full rank, $\mu_1, \ldots, \mu_n$ are decided by A. Although they may not be orthogonal to each other, they can still construct a basis in $C^{n\times n}$.

Let $S_k = \mu_k / |\mu_k|$, obviously $S_1, \ldots, S_n$ construct a normalized basis in $C^{n\times n}$.

Theorem 1. *Let $z_k(t) = x_k(t) + \mathrm{i}y_k(t)$ denote the projection value of $z(t)$ onto S_k. The analytic expression of $|z(t)|^2$ is*

$$|z(t)|^2 = \frac{\sum_{k=1}^{n} \exp\left(2\lambda_k^I t\right) |z_k(0)|^2}{1 + 2\sum_{j=1}^{n} \left|z_j(0)\right|^2 \int_0^t \exp\left(2\lambda_j^I \tau\right) \mathrm{d}\tau} \tag{5}$$

for $t \geq 0$.

Proof. From the denotations of z_k, it follows that

$$z(t) = \sum_{k=1}^{n} [x_k(t) + \mathrm{i}y_k(t)]\, S_k. \tag{6}$$

As S_k is a normalized complex eigenvector and $\lambda_k^R + \lambda_k^I\mathrm{i}$ is the associated eigenvalue, substituting Eq. (6) into Eq. (4) gives that

$$\begin{aligned}\sum_{k=1}^{n}\left[\frac{\mathrm{d}}{\mathrm{d}t}x_k(t) + \mathrm{i}\frac{\mathrm{d}}{\mathrm{d}t}y_k(t)\right] S_k &= -\sum_{k=1}^{n} A\,[x_k(t) + \mathrm{i}y_k(t)]\, S_k\mathrm{i} - \sum_{j=1}^{n}\left|z_j(t)\right|^2 \sum_{k=1}^{n}[x_k(t) + \mathrm{i}y_k(t)]\, S_k \\ &= -\sum_{k=1}^{n}\left(\lambda_k^R + \lambda_k^I\mathrm{i}\right) S_k\,[x_k(t) + \mathrm{i}y_k(t)]\,\mathrm{i} - \sum_{j=1}^{n}\left|z_j(t)\right|^2 \sum_{k=1}^{n}[x_k(t) + \mathrm{i}y_k(t)]\, S_k \\ &= \sum_{k=1}^{n}\left\{\left[\lambda_k^I x_k(t) + \lambda_k^R y_k(t)\right] - \left[x_k(t)\lambda_k^R - y_k(t)\lambda_k^I\right]\mathrm{i}\right\} S_k \\ &\quad - \sum_{j=1}^{n}\left|z_j(t)\right|^2 \sum_{k=1}^{n}[x_k(t) + \mathrm{i}y_k(t)]\, S_k,\end{aligned}$$

in each direction S_k, there exist

$$\frac{\mathrm{d}}{\mathrm{d}t}x_k(t) = \lambda_k^I x_k(t) + \lambda_k^R y_k(t) - \sum_{j=1}^{n}\left|z_j(t)\right|^2 x_k(t), \tag{7}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}y_k(t) = -x_k(t)\lambda_k^R + y_k(t)\lambda_k^I - \sum_{j=1}^{n}\left|z_j(t)\right|^2 y_k(t). \tag{8}$$

From Eqs. (7) and (8), it follows that

$$\frac{\mathrm{d}}{\mathrm{d}t}|z_k(t)|^2 = 2\lambda_k^I |z_k(t)|^2 - 2\sum_{j=1}^{n}\left|z_j(t)\right|^2 |z_k(t)|^2. \tag{9}$$

For $|z_k(t)|^2 \neq 0$, $|z_r(t)|^2 \neq 0$ $(k \neq r)$, we have

$$\frac{1}{|z_k(t)|^2}\frac{\mathrm{d}}{\mathrm{d}t}|z_k(t)|^2 - \frac{1}{|z_r(t)|^2}\frac{\mathrm{d}}{\mathrm{d}t}|z_r(t)|^2 = 2\lambda_k^I - 2\lambda_r^I,$$

so

$$\frac{\mathrm{d}}{\mathrm{d}t}\left\{\ln\frac{|z_k(t)|^2}{|z_r(t)|^2}\right\} = 2\lambda_k^I - 2\lambda_r^I,$$

i.e.

$$\frac{|z_k(t)|^2}{|z_r(t)|^2} = \frac{|z_k(0)|^2}{|z_r(0)|^2}\exp\left[2\left(\lambda_k^I - \lambda_r^I\right)t\right]. \tag{10}$$

Consider the relation between $|z_k(t)|^2$ ($t > 0$) and $|z_k(0)|^2$, there are four cases: (1) using Eq. (9) gives that the following case is sound.

$$|z_k(0)|^2 = 0 \quad \text{and} \quad |z_k(t)|^2 = 0. \tag{11}$$

(2) $|z_k(0)|^2 = 0$ and $|z_k(t)|^2 \neq 0$; (3) $|z_k(0)|^2 \neq 0$ and $|z_k(t)|^2 = 0$. As $|z_k(t)|^2$ is continuously dependent upon t, for case (2), using Eq. (9) gives $|z_k(t)|^2 = |z_k(0)|^2 \exp(2\lambda_k^I - 2\sum_{j=1}^{n} \left|z_j(t)\right|^2)$, it is a contradiction to this case. Similarly we can prove that case (3) is not compatible with Eq. (9). (4) $|z_k(0)|^2 \neq 0$ and $z_k(t) \neq 0$. For this case we have

$$\frac{1}{|z_k(t)|^2} \frac{\mathrm{d}}{\mathrm{d}t} |z_k(t)|^2 - 2\lambda_k^I \frac{1}{|z_k(t)|^2} = -2 \sum_{j=1}^{n} \frac{\left|z_j(t)\right|^2}{|z_k(t)|^2},$$

i.e.

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{1}{|z_k(t)|^2} + 2\lambda_k^I \frac{1}{|z_k(t)|^2} = 2 \sum_{j=1}^{n} \frac{\left|z_j(t)\right|^2}{|z_k(t)|^2}. \tag{12}$$

By case (1), we know $\left|z_j(t)\right|^2 = 0$ when $\left|z_j(0)\right|^2 = 0$. So whether $\left|z_j(t)\right|^2 = 0$ or $\left|z_j(t)\right|^2 \neq 0$, Eq. (10) always holds. Substituting Eq. (10) into Eq. (12) gives that

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\exp\left(2\lambda_k^I t\right)}{|z_k(t)|^2} = 2 \sum_{j=1}^{n} \frac{\left|z_j(0)\right|^2}{|z_k(0)|^2} \exp\left(2\lambda_j^I t\right). \tag{13}$$

Integrating two sides of Eq. (13) with respect to t gives that

$$\frac{\exp\left(2\lambda_k^I t\right)}{|z_k(t)|^2} - \frac{1}{|z_k(0)|^2} = 2 \sum_{j=1}^{n} \frac{\left|z_j(0)\right|^2}{|z_k(0)|^2} \int_0^t \exp\left(2\lambda_j^I \tau\right) \mathrm{d}\tau,$$

so

$$|z_k(t)|^2 = \frac{\exp\left(2\lambda_k^I t\right) |z_k(0)|^2}{1 + 2 \sum_{j=1}^{n} \left|z_j(0)\right|^2 \int_0^t \exp\left(2\lambda_j^I \tau\right) \mathrm{d}\tau} \tag{14}$$

for $t > 0$. Apparently, Eq. (14) involves Eq. (11). Thus they both can be uniformly expressed by Eq. (14). Under Eq. (14), Eq. (5) is easily followed. This theorem is proved. □

3. Convergence analysis

If an equilibrium vector of RNN (1) exists, let ξ denote it, and there exists

$$\xi = \lim_{t \to \infty} z(t). \tag{15}$$

Theorem 2. *If each eigenvalue is a real number, then* $|\xi| = 0$.

Proof. From Theorem 1, we know

$$|z| = \sqrt{\frac{\sum_{k=1}^{n} \exp\left(2\lambda_k^I t\right) |z_k(0)|^2}{1 + 2 \sum_{j=1}^{n} \left|z_j(0)\right|^2 \int_0^t \exp\left(2\lambda_j^I \tau\right) \mathrm{d}\tau}}. \tag{16}$$

So

$$|\xi| = \lim_{t\to\infty} |z(t)| = \lim_{t\to\infty} \sqrt{\frac{\sum_{k=1}^{n} \exp\left(2\lambda_k^I t\right) |z_k(0)|^2}{1 + 2\sum_{j=1}^{n} \left|z_j(0)\right|^2 \int_0^t \exp\left(2\lambda_j^I \tau\right) \mathrm{d}\tau}}. \tag{17}$$

Since each eigenvalue is real,

$$\lambda_1^I = \lambda_2^I = \cdots = \lambda_n^I = 0. \tag{18}$$

From Eqs. (17) and (18), it follows that

$$|\xi| = \lim_{t\to\infty} \sqrt{\frac{\sum_{k=1}^{n} |z_k(0)|^2}{1 + 2\sum_{j=1}^{n} \left|z_j(0)\right|^2 t}} = 0.$$

This theorem is proved. This theorem implies that if a matrix only has real eigenvalues, RNN (1) will converge to zero point, which is independent upon initial complex vector. □

Theorem 3. *Denote* $\lambda_m^I = \max_{1\le k\le n} \lambda_k^I$. *If* $\lambda_m^I > 0$, *then* $\xi^{\mathrm{T}}\bar{\xi} = \lambda_m^I$.

Proof. Using Eq. (15) and Theorem 1 gives that

$$\xi^{\mathrm{T}}\bar{\xi} = \lim_{t\to\infty} |z(t)|^2 = \lim_{t\to\infty} \frac{\sum_{k=1}^{n} \exp\left(2\lambda_k^I t\right) |z_k(0)|^2}{1 + 2\sum_{j=1}^{n} \left|z_j(0)\right|^2 \int_0^t \exp\left(2\lambda_j^I \tau\right) \mathrm{d}\tau},$$

i.e.

$$\begin{aligned}
\xi^{\mathrm{T}}\bar{\xi} &= \lim_{t\to\infty} \frac{\exp\left(2\lambda_m^I t\right) |z_m(0)|^2 + \sum_{k=1,k\ne m}^{n} \exp\left(2\lambda_k^I t\right) |z_k(0)|^2}{1 + 2|z_m(0)|^2 \int_0^t \exp\left(2\lambda_m^I \tau\right) \mathrm{d}\tau + 2\sum_{j=1,j\ne m}^{n} \left|z_j(0)\right|^2 \int_0^t \exp\left(2\lambda_j^I \tau\right) \mathrm{d}\tau} \\
&= \lim_{t\to\infty} \frac{|z_m(0)|^2 + \sum_{k=1,k\ne m}^{n} \exp\left[2\left(\lambda_k^I - \lambda_m^I\right) t\right] |z_k(0)|^2}{\exp\left(-2\lambda_m^I t\right)\left[1 + 2\sum_{j=1,j\ne m}^{n} \left|z_j(0)\right|^2 \int_0^t \exp\left(2\lambda_j^I \tau\right) \mathrm{d}\tau\right] + \frac{1}{\lambda_m^I} |z_m(0)|^2 \left[1 - \exp\left(-2\lambda_m^I t\right)\right]} \\
&= \lambda_m^I.
\end{aligned}$$

This theorem is proved. From this theorem, we know that when the maximal imaginary part of eigenvalues is positive, RNN (1) will converge to a nonzero equilibrium vector, in addition, the square modulus of the vector is equal to the largest imaginary part of all eigenvalues. □

Theorem 4. *If* A' *is replaced by* $\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$, *then*

$$|z(t)|^2 = \frac{\sum_{k=1}^{n} \exp\left(2\lambda_k^R t\right) |z_k(0)|^2}{1 + 2\sum_{j=1}^{n} \left|z_j(0)\right|^2 \int_0^t \exp\left(2\lambda_j^R \tau\right) \mathrm{d}\tau}.$$

Proof. When $A' = \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$, by a similar way of obtaining Eq. (4), RNN (1) is transformed into

$$\frac{\mathrm{d}z(t)}{\mathrm{d}t} = Az(t) - z^{\mathrm{T}}(t)\bar{z}(t)z(t). \tag{19}$$

Using the denotations of $x_k(t)$ and $y_k(t)$, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} x_k(t) = \lambda_k^R x_k(t) - \lambda_k^I y_k(t) - \sum_{j=1}^{n} \left[x_j^2(t) + y_j^2(t) \right] x_k(t), \tag{20}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} y_k(t) = \lambda_k^I x_k(t) + \lambda_k^R y_k(t) - \sum_{j=1}^{n} \left[x_j^2(t) + y_j^2(t) \right] y_k(t). \tag{21}$$

From Eqs. (20) and (21), it follows that

$$\frac{\mathrm{d}}{\mathrm{d}t} |z_k(t)|^2 = 2\lambda_k^R |z_k(t)|^2 - 2\sum_{j=1}^{n} \left| z_j(t) \right|^2 |z_k(t)|^2. \tag{22}$$

Comparing Eq. (22) with Eq. (9), we find that replacing λ_k^I in Eq. (9) by λ_k^R, Eq. (22) is obtained. So, like the derivation from Eq. (10) to Eq. (14), it follows that

$$|z(t)|^2 = \frac{\sum_{k=1}^{n} \exp\left(2\lambda_k^R t\right) |z_k(0)|^2}{1 + 2\sum_{j=1}^{n} \left| z_j(0) \right|^2 \int_0^t \exp\left(2\lambda_j^R \tau\right) \mathrm{d}\tau}.$$

This theorem is proved. This theorem provides a path to extract the maximal real part of all eigenvalues through rearranging the connection weights. □

Theorem 5. *Let $\lambda_m^R = \max_{1 \le k \le n} \lambda_k^R$. When A' is replaced by $\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$, if $\lambda_m^R \le 0$, then $\xi^{\mathrm{T}}\bar{\xi} = 0$. If $\lambda_m^R > 0$, then $\xi^{\mathrm{T}}\bar{\xi} = \lambda_m^R$.*

Proof. Using Eq. (15) and Theorem 4 gives that

$$\xi^{\mathrm{T}}\bar{\xi} = \lim_{t\to\infty} |z(t)|^2 = \lim_{t\to\infty} \frac{\sum_{k=1}^{n} \exp\left(2\lambda_k^R t\right) |z_k(0)|^2}{1 + 2\sum_{j=1}^{n} \left| z_j(0) \right|^2 \int_0^t \exp\left(2\lambda_j^R \tau\right) \mathrm{d}\tau}. \tag{23}$$

From Eq. (23), if $\lambda_m^R < 0$, it easily follows that

$$\xi^{\mathrm{T}}\bar{\xi} = 0, \tag{24}$$

if $\lambda_m^R = 0$, it follows that

$$\begin{aligned} \xi^{\mathrm{T}}\bar{\xi} &= \lim_{t\to\infty} \frac{\exp\left(2\lambda_m^R t\right) |z_m(0)|^2 + \sum_{k=1,k\neq m}^{n} \exp\left(2\lambda_k^R t\right) |z_k(0)|^2}{1 + 2|z_m(0)|^2 \int_0^t \exp\left(2\lambda_m^R \tau\right) \mathrm{d}\tau + 2\sum_{j=1,j\neq m}^{n} \left| z_j(0) \right|^2 \int_0^t \exp\left(2\lambda_j^R \tau\right) \mathrm{d}\tau} \\ &= \lim_{t\to\infty} \frac{|z_m(0)|^2 + \sum_{k=1,k\neq m}^{n} \exp\left(2\lambda_k^R t\right) |z_k(0)|^2}{1 + 2|z_m(0)|^2 t + 2\sum_{j=1,j\neq m}^{n} \left| z_j(0) \right|^2 \int_0^t \exp\left(2\lambda_j^R \tau\right) \mathrm{d}\tau} \\ &= 0 \end{aligned} \tag{25}$$

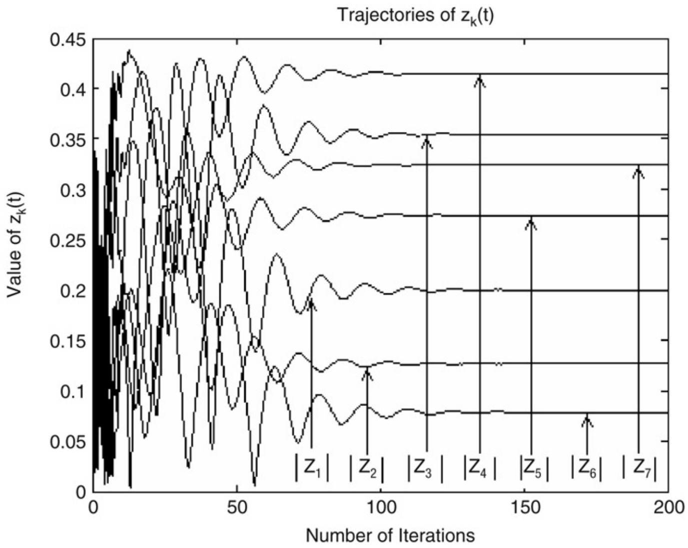

Fig. 1. The trajectories of $|z_k(t)|$ in searching λ_m^I when $n = 7$.

and if $\lambda_m^R > 0$, like the deductive procedures of Theorem 3, it follows that

$$\xi^{\mathrm{T}}\bar{\xi} = \lambda_m^R. \tag{26}$$

From Eqs. (24)–(26), we know that the theorem is proved. This theorem indicates that when the largest real part of all eigenvalues is positive, the slightly changed RNN (1) converges to a nonzero equilibrium vector, and the square modulus of it is equal to the maximal real part. □

4. Simulations and discussions

To illustrate the method, two examples are given. Example 1 uses a 7 × 7 matrix to show the validity. A 50 × 50, and a 100 × 100 matrices are employed to test the method when dimensionality increases in Example 2.

Example 1. Evaluate A with randomly generated values like

$$A = \begin{pmatrix} 0.1347 & 0.0324 & 0.8660 & 0.8636 & 0.6390 & 0.1760 & 0.4075 \\ 0.0225 & 0.7339 & 0.2542 & 0.5676 & 0.6690 & 0.0020 & 0.4078 \\ 0.2622 & 0.5365 & 0.5695 & 0.9805 & 0.7721 & 0.7902 & 0.0527 \\ 0.1165 & 0.2760 & 0.1593 & 0.7918 & 0.3798 & 0.5136 & 0.9418 \\ 0.0693 & 0.3685 & 0.5944 & 0.1526 & 0.4416 & 0.2132 & 0.1500 \\ 0.8529 & 0.0129 & 0.3311 & 0.8330 & 0.4831 & 0.1034 & 0.3844 \\ 0.1803 & 0.8892 & 0.6586 & 0.1919 & 0.6081 & 0.1573 & 0.3111 \end{pmatrix}.$$

Eigenvalues directly computed are $\lambda_1 = 2.9506, \lambda_2 = 0.7105, \lambda_3 = -0.3799 + 0.4808\mathrm{i}, \lambda_4 = \bar{\lambda}_3, \lambda_5 = 0.0286 + 0.5397\mathrm{i}, \lambda_6 = \bar{\lambda}_5$ and $\lambda_7 = 0.1274$, so $\lambda_m^I = 0.5397$ and $\lambda_m^R = 2.9506$.

When the initial vector is $z(0) = [0.0506+0.0508\mathrm{i}, 0.2690+0.1574\mathrm{i}, 0.0968+0.1924\mathrm{i}, 0.2202+0.0049\mathrm{i}, 0.1233+0.2511\mathrm{i}, 0.1199 + 0.2410\mathrm{i}, 0.1517 + 0.2093\mathrm{i}]^{\mathrm{T}}$, we get the equilibrium vector $\xi = [0.1335 - 0.1489\mathrm{i}, 0.0204 + 0.1250\mathrm{i}, 0.3471 + 0.0709\mathrm{i}, -0.1727 + 0.3771\mathrm{i}, -0.0531 - 0.2687\mathrm{i}, -0.0719 - 0.0317\mathrm{i}, -0.0970 - 0.3091\mathrm{i}]^{\mathrm{T}}$. So the computed maximum imaginary part is $\underline{\lambda}_m^I = \xi^{\mathrm{T}}\bar{\xi} = 0.5397$.

Comparing λ_m^I with $\underline{\lambda}_m^I$, it can be easily seen that they are very close. The trajectories of $|z_k(t)|$ $(k = 1, 2, \ldots, 7)$ and $z^{\mathrm{T}}(t)\,\bar{z}(t)$ which will approach λ_m^I are shown in Figs. 1 and 2.

When $A' = \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}$, the computed maximum real part is

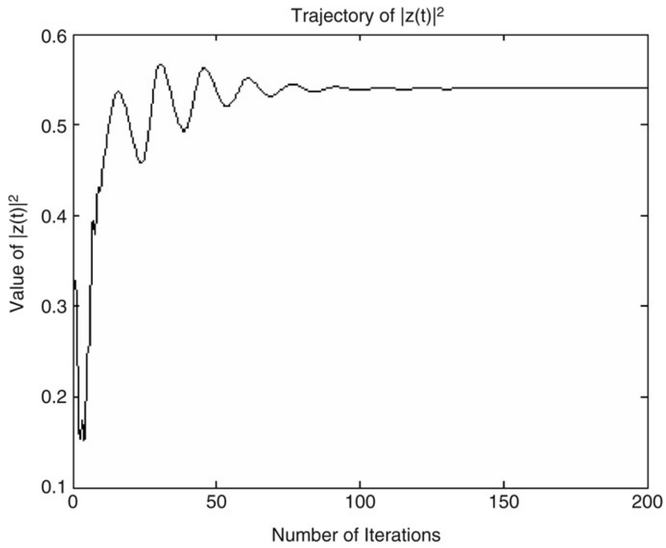

Fig. 2. The trajectory of $|z(t)|^2$ in searching λ_m^I when $n = 7$.

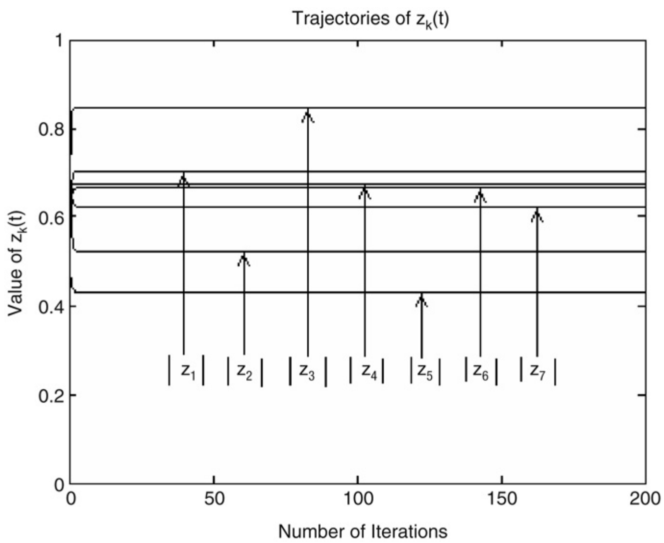

Fig. 3. The trajectories of $|z_k(t)|$ in searching λ_m^R when $n = 7$.

$$\begin{aligned}\underline{\lambda}_m^R &= |(0.4981 + 0.4949\text{i}\ 0.3701 + 0.3678\text{i}\ 0.6003 + 0.5965\text{i}\ 0.4773 + 0.4742\text{i}\ 0.3059\\ &\quad + 0.3039\text{i}\ 0.4719 + 0.4689\text{i}\ 0.4418 + 0.4390\text{i})^{\mathrm{T}}|^2\\ &= 2.9504.\end{aligned}$$

Comparing $\underline{\lambda}_m^R$ with λ_m^R, the absolute difference value is $\Delta\lambda_m^R = \left|\lambda_m^R - \underline{\lambda}_m^R\right| = |2.9506 - 2.9504| = 0.0002$, so, $\underline{\lambda}_m^R$ is very close to λ_m^R. The trajectories of $|z_k(t)|$ $(k = 1, 2, \ldots, 7)$ and $z^{\mathrm{T}}(t)\,\bar{z}(t)$ are shown in Figs. 3 and 4. With the variation of $z(0)$, the trajectories of $|z_k(t)|$ $(k = 1, 2, \ldots, 7)$ and $z^{\mathrm{T}}(t)\,\bar{z}(t)$ will vary. But $\underline{\lambda}_m^I$ and $\underline{\lambda}_m^R$ do insistently approach the corresponding true values.

Example 2. How does the approach behave when the dimensionality increases, if a 50×50 matrix is randomly produced, the expression may be too long for presentation, thus a 50×50 matrix is specially given as

$$a_{ij} = \begin{cases}(-1)^i (50 - i)/100, & i = j\\ (1 - i/50)^i - (j/50)^j, & i \neq j.\end{cases}$$

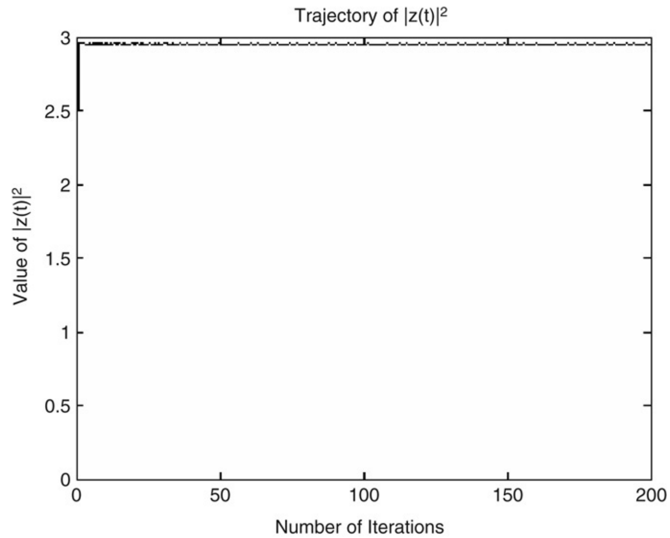

Fig. 4. The trajectory of $|z(t)|^2$ in searching λ_m^R when $n = 7$.

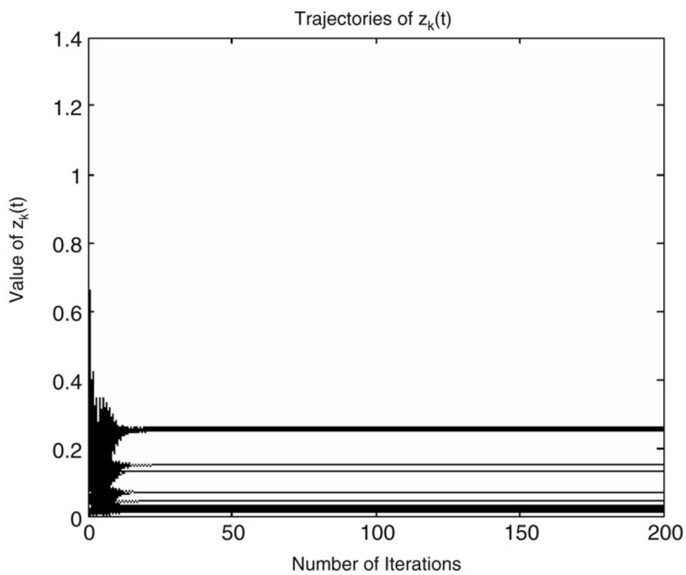

Fig. 5. The trajectories of $|z_k(t)|$ in searching λ_m^I when $n = 50$.

The calculated results are $\underline{\lambda}_m^I = 0.3244$ and $\underline{\lambda}_m^R = 4.6944$. Convergence behaviors of $|z_k(t)|$ and $z^{\mathrm{T}}(t)\,\bar{z}(t)$ in searching $\underline{\lambda}_m^I$ and $\underline{\lambda}_m^R$ are shown in Figs. 5–8. Comparing $\underline{\lambda}_m^I$ and $\underline{\lambda}_m^R$ with corresponding true ones $\lambda_m^I = 0.3246$ and $\lambda_m^R = 4.7000$, we find that each comparing pair is very close. From Figs. 5–8, we also can see that the system gets to equilibrium state soon though the dimensionality has reached 50.

To ulteriorly show the validity of this approach when dimensionality becomes large, let $n = 100$. The corresponding convergence behaviors are shown in Figs. 9–12. $\underline{\lambda}_m^R = 7.3220$ keeps very close to $\lambda_m^R = 7.3224$, and $\underline{\lambda}_m^I = 0.2541$ is very close to $\lambda_m^I = 0.2540$. From Figs. 5–12, we can see that the iteration number at which the system enters into equilibrium state is not sensitive with dimensionality.

In realization, since $\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} = \begin{pmatrix} 0 & A \\ -A & 0 \end{pmatrix}\begin{pmatrix} 0 & -U \\ U & 0 \end{pmatrix}$ and $\begin{pmatrix} 0 & A \\ -A & 0 \end{pmatrix} = -\begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix}\begin{pmatrix} 0 & -U \\ U & 0 \end{pmatrix}$, the two RNNs easily switch. The former relation is used to transform RNN (1) into the RNN introduced in Theorem 4, and the latter for the reverse.

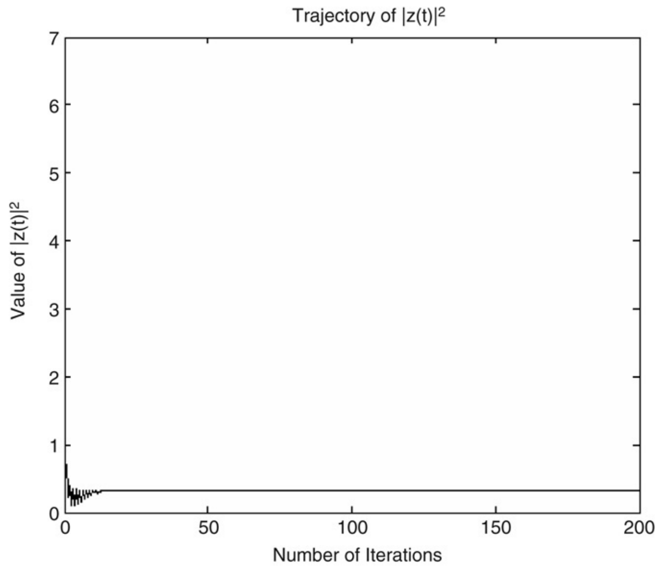

Fig. 6. Trajectories of $|z(t)|^2$ in searching λ_m^I when $n = 50$.

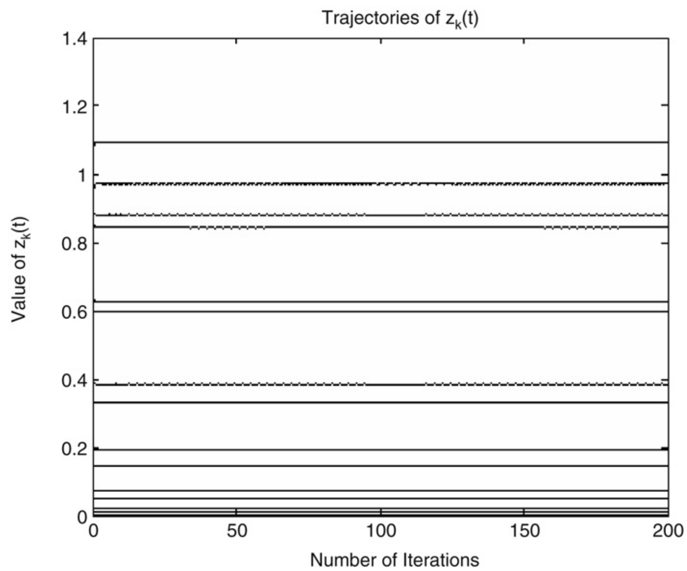

Fig. 7. The trajectories of $|z_k(t)|$ in searching λ_m^R when $n = 50$.

Since the simulation platform only occupies one CPU, the merit of quickness is not fully exhibited as the simulating operation runs sequentially in nature. Roughly speaking, only when neural network is implemented by electronic circuit, the concurrent, and quickness, virtues can be wholly acquired since in such an instance all calculational nodes can run synchronously. This paper only provides a potential approach for large-scale eigenvalue module estimation, which can be used for spectral radius estimation for real-time signal processing, etc. Compared with traditional numerical methods such as power method, QR algorithm and all kinds of recursive methods [8,15], which have sequential processing nature, this algorithm is expected to achieve higher computational performance for large-scale problems.

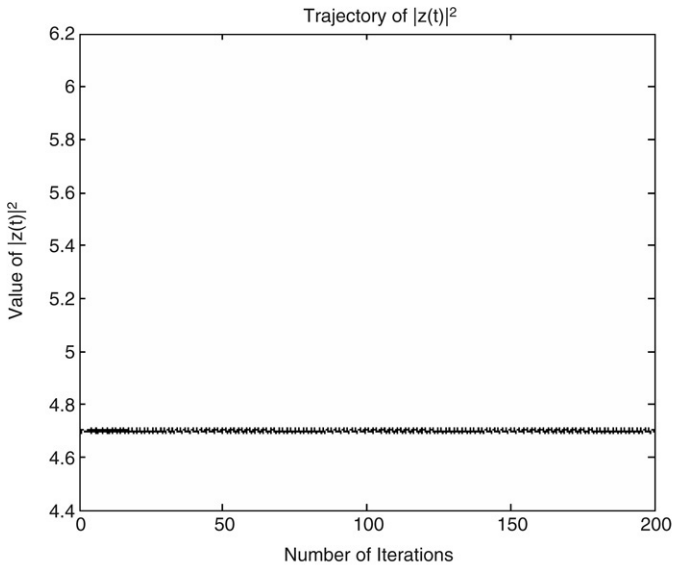

Fig. 8. Trajectories of $|z(t)|^2$ in searching λ_m^R when $n = 50$.

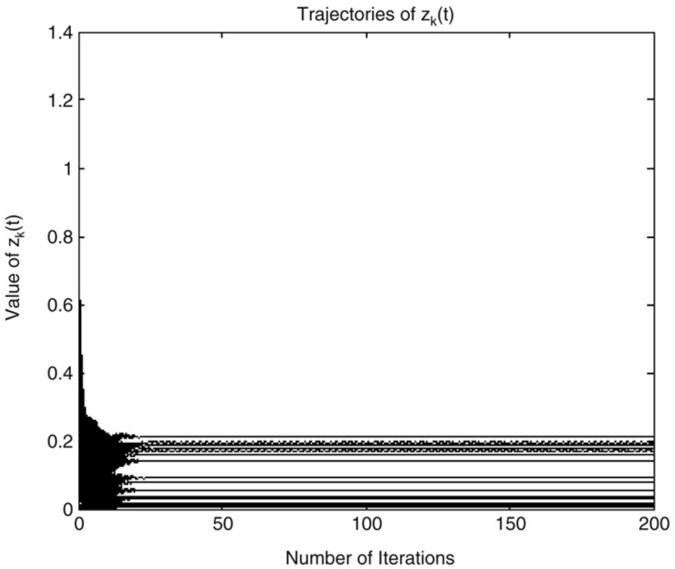

Fig. 9. The trajectories of $|z_k(t)|$ in searching λ_m^I when $n = 100$.

5. Conclusions

The quick computation of eigenvalues of a general real matrix is very important for some special applications such as real-time principal component analysis, etc. This paper is an exploration in this field. Because neural network runs in a concurrent manner, using it to perform the computation can achieve high speed. Although lots of research focuses on using neural network to compute eigenpairs of a real symmetric, or anti-symmetric, matrix, the works about accomplishing similar computations for a general real matrix are scarce in the relevant literature. Therefore, this paper proposes a recurrent neural network model calculating the largest imaginary, or real, part, of a general real matrix's eigenvalues. The network is described by a set of differential equations, which is transformed into a complex differential system. After obtaining the square module of the system variable, the convergence behavior of the network is discussed in detail. The simulation results of a test on a 7×7 general real matrix indicate that the computed values

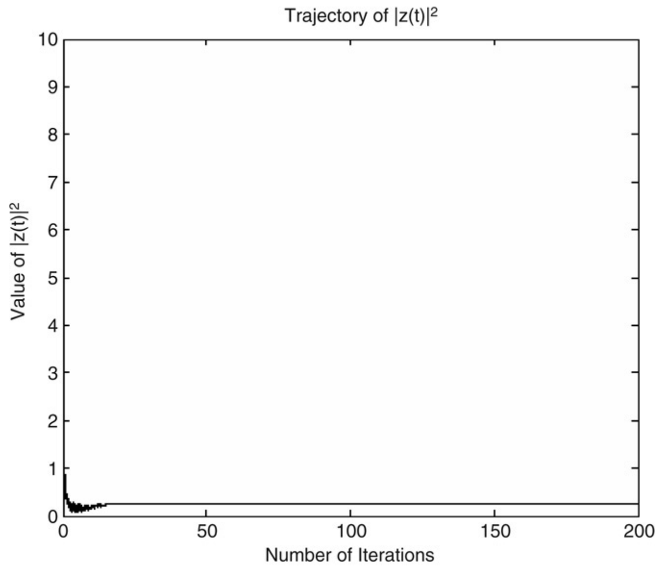

Fig. 10. Trajectories of $|z(t)|^2$ in searching λ_m^I when $n = 100$.

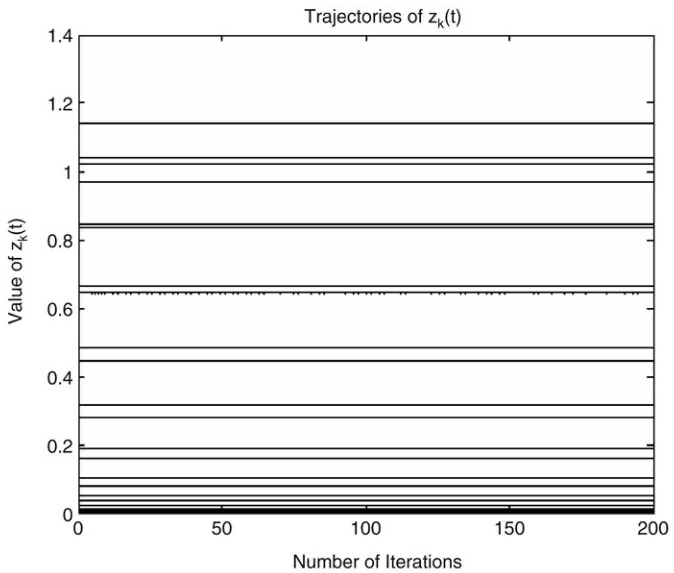

Fig. 11. The trajectories of $|z_k(t)|$ in searching λ_m^R when $n = 100$.

are very close to the corresponding true ones. For showing the approach's performance when dimensionality increases, a 50×50, and a 100×100, matrices are used for testing. The simulation results accord with corresponding true ones nicely. It can also be seen from the results that the iteration number at which the network enters into equilibrium state is not sensitive with the dimension number. This approach can be potentially used to estimate the largest modulus of a general real matrix's eigenvalues.

Acknowledgements

The authors would like to thank the editor and the anonymous referees for their helpful comments. This work was supported by the Youth Foundations of Sichuan University (No. 200574) and School of Computer Science and Engineering, and Natural Science Foundation of China (No. 60303034).

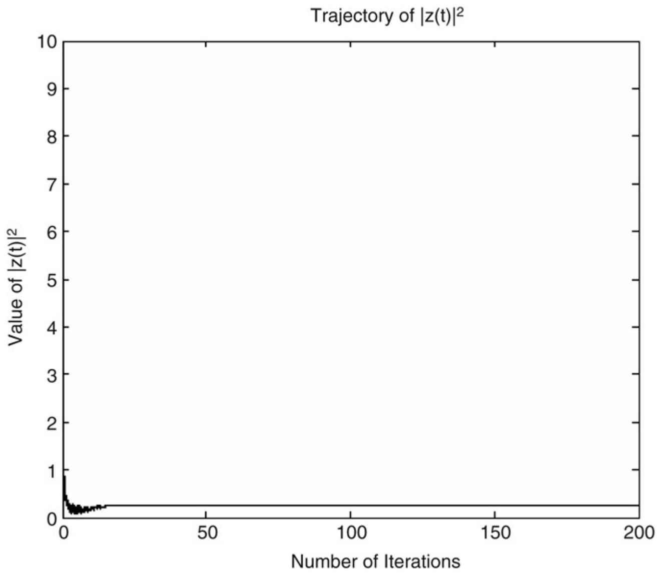

Fig. 12. Trajectories of $|z(t)|^2$ in searching λ_m^R when $n = 100$.

References

[1] L. Xu, I. King, A PCA approach for fast retrieval of structural patterns in attributed graphs, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 31 (2001) 812–817.

[2] B. Zhang, M. Fu, H. Yan, A nonlinear neural network model of mixture of local principal component analysis: Application to handwritten digits recognition, Pattern Recognition 34 (2001) 203–214.

[3] Z. Yi, Y. Fu, J. Hua, Neural networks based approach computing eigenvectors and eigenvalues of symmetric matrix, Computers and Mathematics with Applications 47 (2004) 1155–1164.

[4] J.M. Vegas, P.J. Zufiria, Generalized neural networks for spectral analysis: Dynamics and Liapunov functions, Neural Networks 17 (2004) 233–245.

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

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

[7] A. Cichocki, R. Unbehauen, Neural networks for computing eigenvalues and eigenvectors, Biological Cybernetics 68 (1992) 155–164.

[8] U. Helmke, J.B. Moore, Optimization and Dynamical Systems, Springer Verlag, London, 1994.

[9] H. Kakeya, T. Kindo, Eigenspace separation of autocorrelation memory matrices for capacity expansion, Neural Networks 10 (1997) 833–843.

[10] N. Li, A matrix inverse eigenvalue problem and its application, Linear Algebra and its Applications 266 (1997) 143–152.

[11] F.-L. Luo, R. Unbehauen, Y.-D. Li, A principal component analysis algorithm with invariable norm, Neurocomputing 8 (1995) 213–221.

[12] F.-L. Luo, R. Unbehauen, A. Cichocki, A minor component analysis algorithm, Neural Networks 10 (1997) 291–297.

[13] F.-L. Luo, Y.-D. Li, Real-time neural computation of the eigenvector corresponding to the largest eigenvalue of positive matrix, Neurocomputing 7 (1995) 145–157.

[14] R. Perfetti, E. Massarelli, Training spatially homogeneous fully recurrent neural networks in eigenvalue space, Neural Networks 10 (1997) 125–137.

[15] V.U. Reddy, G. Mathew, A. Paulraj, Some algorithms for eigensubspace estimation, Digital Signal Processing 5 (1995) 97–115.

[16] N. Samardzija, R.L. Waterland, A neural network for computing eigenvectors and eigenvalues, Biological Cybernetics 65 (1991) 211–214.

[17] J. Song, Y. Yam, Complex functional neural network for computing the inverse and pseudo-inverse of the complex matrix, Applied Mathematics and Computing 93 (1998) 195–205.

[18] Y. Tan, Z. He, Neural network approaches for the extraction of the eigenstructure, in: Neural Networks for Signal Processing VI — Proceedings of the 1996 IEEE Workshop, Kyoto, Japan, 1996, pp. 23–32.

[19] Y. Tan, Z. Liu, On matrix eigen-decomposition by neural networks, Neural Networks World 8 (1998) 337–352.

[20] J. Wang, Recurrent neural networks for computing pseudoinverses of rank-deficient matrices, SIAM Journal on Scientific Computing 18 (1997) 1479–1493.