Available online at www.sciencedirect.com
SCIENCE DIRECT®
An International Journal
computers & mathematics with applications
ELSEVIER Computers and Mathematics with Applications 47 (2004) 1155–1164
www.elsevier.com/locate/camwa

Neural Networks Based Approach for Computing Eigenvectors and Eigenvalues of Symmetric Matrix

ZHANG YI AND YAN FU
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, 610054, P.R. China
<zhangyi><fuyan>@uestc.edu.cn

HUA JIN TANG
Department of Electrical and Computer Engineering
National University of Singapore
4 Engineering Drive 3, Singapore 117576
engp1880@nus.edu.sg

(Received September 2002; revised and accepted October 2003)

Abstract—Efficient computation of eigenvectors and eigenvalues of a matrix is an important problem in engineering, especially for computing eigenvectors corresponding to largest or smallest eigenvalues of a matrix. This paper proposes a neural network based approach to compute eigenvectors corresponding to the largest or smallest eigenvalues of any real symmetric matrix. The proposed network model is described by differential equations, which is a class of continuous time recurrent neural network model. It has parallel processing ability in an asynchronous manner and can achieve high computing performance. This paper provides a clear mathematical understanding of the network dynamic behaviors relating to the computation of eigenvectors and eigenvalues. Computer simulation results show the computational capability of the network model.

Keywords—Recurrent neural networks, Eigenvalues, Eigenvectors, Eigenspace, Symmetric matrix.

1. INTRODUCTION

Computing eigenvectors of a matrix is an important and interesting problem in engineering, especially for computing eigenvectors corresponding to largest or smallest eigenvalues. There are many important applications of such computations, for example, applications in adaptive signal processing. Engineers and scientists often desire tools for fast computation of eigenvectors of matrices. Recently, many research works on this problem by using neural networks have been reported, say for examples [1–14]. Most of these works are focused on computing the eigenvectors of positive definite symmetric matrices corresponding to the largest or smallest eigenvalues. A more general case will be studied in this paper. A method based on neural network approach

0898-1221/04/$ - see front matter Typeset by AMS-TEX
doi:10.1016/j.camwa.2004.04.001

will be developed to compute eigenvectors of any real symmetric matrix. The proposed model of neural network will be described by differential equations, which is adapted from [3–5,11]. It is a model of artificial recurrent neural networks that have asynchronous parallel processing ability and can achieve high computing performance.

Let A be a $n \times n$ real symmetric matrix. The dynamics of the proposed neural network model is described by

$$\frac{dx(t)}{dt} = -x(t) + f(x(t)), \tag{1}$$

for $t \geq 0$, where

$$f(x) = \left[x^\top x A + \left(1 - x^\top A x\right) I\right] x$$

and $x = (x_1, \ldots, x_n)^\top \in R^n$ represents the state of the network, I is the $n \times n$ identity matrix. Clearly, this is a class of recurrent neural networks.

Network (1) is a nonlinear differential equation. Its dynamic behavior plays a crucial and important part to its applications. From the engineering point of view, neural networks that possess well understood dynamic behaviors are most attractive. This paper clearly explores the dynamic behaviors of the network model.

The rest of this paper is organized as follows. In Section 2, the properties of equilibrium points of the network model will be studied. In Section 3, an interesting representation of solutions of the network will be established. Convergence of the network will be discussed in Section 4. Some computer simulations are given in Section 5. Finally, in Section 6, the conclusion follows.

2. EQUILIBRIUM POINTS ANALYSIS

A vector $\xi \in R^n$ is called an equilibrium point of (1) if and only if it satisfies that $-\xi + f(\xi) = 0$, i.e.,

$$\xi^\top \xi A \xi - \xi^\top A \xi \xi = 0. \tag{2}$$

Denote by E the set of all equilibrium points of (1). Given an eigenvalue λ of A, denote by V_λ the eigenspace corresponding to λ.

THEOREM 1. *The set of all the equilibrium points of (1) is equal to the union of all eigenspace of A, i.e.,*

$$E = \bigcup_\lambda V_\lambda.$$

PROOF. Given any $\xi \in \bigcup_\lambda V_\lambda$, if $\xi = 0$, clearly, $\xi \in E$. Suppose that $\xi \neq 0$, there is an eigenvalue λ of A such that $\xi \in V_\lambda$, i.e., $A\xi = \lambda\xi$. Thus,

$$\xi^\top \xi A \xi - \xi^\top A \xi \xi = \xi^\top \xi \lambda \xi - \lambda \left(\xi^\top \xi\right) \xi = 0.$$

By (2), $\xi \in E$. This implies $E \supseteq \bigcup_\lambda V_\lambda$.

On the other hand, for any $\eta \in E$, it holds that $\eta^\top \eta A \eta - \eta^\top A \eta \eta = 0$. If $\eta = 0$, clearly, $\eta \in \bigcup_\lambda V_\lambda$. If $\eta \neq 0$, it holds that

$$A\eta = \frac{\eta^\top A \eta}{\eta^\top \eta} \eta.$$

This shows that η is an eigenvector of A. It follows that $E \subseteq \bigcup_\lambda V_\lambda$. Thus, $E = \bigcup_\lambda V_\lambda$. This completes the proof.

The above theorem shows that any equilibrium state of network (1) is an eigenvector of A if it is not the zero vector. Thus, if the convergence of the networks can be proved, it can provide a method to compute the eigenvectors of A.

Once an eigenvector is obtained, it is easy to compute its corresponding eigenvalue. Let v be an eigenvector of A, the corresponding eigenvalue can be computed by $\lambda = v^\top A v / v^\top v$. This is because

$$Av = \frac{v^\top A v}{v^\top v} v.$$

This provides a method to compute eigenvalues by known eigenvectors.

3. REPRESENTATION OF SOLUTIONS

In this section, a representation of the solutions of network (1) will be established. It is clear that this network model is a nonlinear differential equation. Generally speaking, it is not easy to solve nonlinear differential equations for solutions. However, due to the fact that the matrix included in network (1) is a symmetric matrix, this enables us to establish an interesting representation of the solutions of network (1).

Since A is a symmetric matrix, then there exists an orthonormal basis of R^n composed by eigenvectors of R^n. Let λ_i $(i = 1, \ldots, n)$ be eigenvalues of A and S_i $(i = 1, \ldots, n)$ be the corresponding eigenvectors that compose an orthonormal basis of A. Then, for any $x \in R^n$, it can be represented as $x = \sum_{i=1}^{n} z_i S_i$, where z_i $(i = 1, \ldots, n)$ are some constants.

THEOREM 2. *Given any $x(0) \in R^n$, suppose $x(0) = \sum_{i=1}^{n} z_i(0) S_i \neq 0$. The solution of (1) starting from $x(0)$ can be represented as*

$$x(t) = \sum_{i=1}^{n} \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=1}^{n} z_j^2(0) e^{2x(0)^\top x(0)(\lambda_j - \lambda_i)t}}} z_i(0) S_i,$$

for all $t \geq 0$.

PROOF. Clearly, network (1) is equivalent to

$$\dot{x}(t) = x(t)^\top x(t) A x(t) - x(t)^\top A x(t) x(t),$$

for $t \geq 0$.

For any $t \geq 0$, let $x(t)$ be the solution of (1) starting from $x(0)$. It follows that

$$\begin{aligned} \frac{d[x(t)^\top x(t)]}{dt} &= 2x(t)\dot{x}(t) \\ &= x(t)^\top x(t) x(t)^\top A x(t) - x(t)^\top A x(t) x(t)^\top x(t) \\ &= 0, \end{aligned}$$

for $t \geq 0$. Then, it holds that $x(t)^\top x(t) = x(0)^\top x(0)$, for all $t \geq 0$.

Rewrite (3) as

$$\dot{x}(t) = x(0)^\top x(0) A x(t) - x(t)^\top A x(t) x(t),$$

for $t \geq 0$. Let $x(t)$ be represented under the basis of S_i $(i = 1, \ldots, n)$ as

$$x(t) = \sum_{i=1}^{n} z_i(t) S_i,$$

for $t \geq 0$, where $z_i(t)$ $(i = 1, \ldots, n)$ are some differentiable functions defined on $[0, +\infty)$. Then, from (4), it follows that

$$\dot{z}_i(t) = x(0)^\top x(0) \lambda_i z_i(t) - \left(\sum_{j=1}^{n} \lambda_j z_j^2(t) \right) z_i(t),$$

for $t \geq 0$ and all $i = 1, \ldots, n$.

Since $x(0) \neq 0$, there exists an $r (1 \leq r \leq n)$ such that $z_r(0) \neq 0$. Using (5), it follows that

$$z_r(t) = z_r(0) e^{\int_0^t \left[x(0)^\top x(0) \lambda_r - \left(\sum_{j=1}^{n} \lambda_j z_j^2(\theta) \right) \right] d\theta} \neq 0,$$

for all $t \geq 0$. Without lose of generality, suppose $z_r(0) > 0$, then it holds that $z_r(t) > 0$, for all $t \geq 0$.

It follows from (5) that

$$\frac{d}{dt}\left[\frac{z_j(t)}{z_r(t)}\right] = x(0)^\top x(0)(\lambda_j - \lambda_r)\left[\frac{z_j(t)}{z_r(t)}\right] \qquad (j = 1, \ldots, n),$$

for all $t \geq 0$. Then,

$$\frac{z_j(t)}{z_r(t)} = \frac{z_j(t)}{z_r(t)} e^{x(0)^\top x(0)(\lambda_j - \lambda_r)} \qquad (j = 1, \ldots, n),$$

for all $t \geq 0$. Next, $z_r(t)$ will be found.

Using (5), it gives that

$$\frac{dz_r(t)}{dt} = x(0)^\top x(0)\lambda_r z_r(t) - \left(\sum_{j=1}^{n} \lambda_j z_j^2(t)\right) z_r(t),$$

for $t \geq 0$, then

$$\frac{d}{dt}\left[\frac{1}{z_r^2(t)}\right] = -2\lambda_r x(0)^\top x(0)\left[\frac{1}{z_r^2(t)}\right] + 2\sum_{j=1}^{n} \lambda_j \left[\frac{z_j(t)}{z_r(t)}\right]^2,$$

for $t \geq 0$. Substituting (6) into the right-hand side of the above equation, it gives that

$$\frac{d}{dt}\left[\frac{1}{z_r^2(t)}\right] = -2\lambda_r x(0)^\top x(0)\left[\frac{1}{z_r^2(t)}\right] + 2\sum_{j=1}^{n} \lambda_j \left[\frac{z_j(0)}{z_r(0)}\right]^2 e^{2(\lambda_j - \lambda_r)t},$$

for $t \geq 0$. By integrating, it follows that

$$\begin{aligned}\frac{1}{z_r^2(t)} &= \frac{1}{z_r^2(0)} e^{-2x(0)^\top x(0)\lambda_r t} \\ &\quad + 2\sum_{j=1}^{n} \lambda_j \left[\frac{z_j(0)}{z_r(0)}\right]^2 \int_0^t e^{-2x(0)^\top x(0))(\lambda_r t - \lambda_j s)} ds \\ &= \frac{1}{z_r^2(0) x(0)^\top x(0)} \sum_{j=1}^{n} z_j^2(0) e^{2x(0)^\top x(0)(\lambda_j - \lambda_r)t},\end{aligned}$$

for $t \geq 0$. Thus,

$$z_r(t) = \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=1}^{n} z_j^2(0) e^{2x(0)^\top x(0)(\lambda_j - \lambda_r)t}}}\, z_r(0).$$

Substituting (7) into (6), it follows that

$$\begin{aligned}z_i(t) &= z_r(t)\frac{z_i(0)}{z_r(0)} e^{x(0)^\top x(0)(\lambda_i - \lambda_r)t} \\ &= \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=1}^{n} z_j^2(0) e^{2x(0)^\top x(0)(\lambda_j - \lambda_i)t}}}\, z_i(0),\end{aligned}$$

for all $t \geq 0$ and $i = 1, \ldots, n$. Therefore,

$$x(t) = \sum_{i=1}^{n} \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=1}^{n} z_j^2(0) e^{2x(0)^\top x(0)(\lambda_j - \lambda_i)t}}}\, z_i(0) S_i,$$

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

Theorem 2 above shows the solutions of network (1) can be represented in terms of a set of orthogonal eigenvectors. This property will be quite convenient for studying the convergence of the network in the next section.

4. CONVERGENCE ANALYSIS

In this section, convergence of the neural network (1) will be analyzed. For an artificial neural network of continuous model, the convergence property is crucial. It is a basic requirement for designing a neural network to guarantee convergence. In the last section, an interesting representation of the solutions of network (1) has been established. Using this representation, the convergence property can be studied on a sound foundation.

THEOREM 3. *Each solution of (1) starting from any nonzero points in R^n will converge to an eigenvector of A.*

PROOF. Let $\lambda_1, \ldots, \lambda_n$ be all the eigenvalues of A ordered by $\lambda_1 \geq \cdots \geq \lambda_n$. Suppose that S_i $(i = 1, \ldots, n)$ is an orthonormal basis in R^n such that each S_i is an eigenvector of A corresponding to the eigenvalue λ_i. Let σ_i $(i = 1, \ldots, m)$ be all the distinct eigenvalues of A ordered by $\sigma_1 > \cdots > \sigma_m$. For any i, $1 \leq i \leq m$, denote the algebraic sum of the multiplicity of $\sigma_1, \ldots, \sigma_i$ by k_i. Clearly, $k_m = n$. For convenience, denote $k_0 = 1$. It is easy to see that $\lambda_i = \sigma_r$, for all $i \in [k_{r-1}, k_r]$, and $S_i \in V_{\sigma_r}$, for all $i \in [k_{r-1}, k_r]$.

Let $x(0)$ be a nonzero point in R^n and suppose that $x(0) = \sum_{i=1}^n z_i(0)S_i$. Define

$$l = \min\{l \mid 1 \leq l \leq n,\ z_l(0) \neq 0\},$$

then there exists an $r \in \{1, \ldots, m\}$ such that $k_{r-1} \leq l \leq k_r$. By Theorem 2, the solution of (1) starting from $x(0)$ must satisfy

$$\begin{aligned}
x(t) &= \sum_{i=1}^{n} \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=1}^{n} z_j^2(0) e^{2x(0)^\top x(0)(\lambda_j - \lambda_i)t}}}\, z_i(0) S_i \\
&= \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=l}^{k_r} z_j^2(0) + \sum_{j=k_r+1}^{n} z_j^2(0) e^{2x(0)^\top x(0)(\lambda_j - \sigma_r)t}}} \sum_{i=l}^{k_r} z_i(0) S_i \\
&\quad + \sum_{i=k_r+1}^{n} \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=l}^{k_r} z_j^2(0) e^{2x(0)^\top x(0)(\sigma_r - \lambda_i)t} + \sum_{j=k_r+1}^{n} z_j^2(0) e^{2x(0)^\top x(0)(\lambda_j - \lambda_i)t}}}\, z_i(0) S_i \\
&\to \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=l}^{k_r} z_j^2(0)}} \sum_{i=l}^{k_r} z_i(0) S_i \in V_{\sigma_r}
\end{aligned}$$

as $t \to +\infty$. This completes the proof.

THEOREM 4. *Given any nonzero vector $x(0) \in R^n$, if $x(0)$ is not orthogonal to V_{σ_1}, then the solution of (1) starting from $x(0)$ converges to an eigenvector corresponding to the largest eigenvalue of A.*

PROOF. Suppose $x(0) = \sum_i^n z_i(0)S_i$, then $x(0)$ is not orthogonal to V_{σ_1} if and only if there exists an l $(1 \leq l \leq k_1)$ such that $z_l(0) \neq 0$. From the proof of Theorem 3, it follows that

$$x(t) \to \sqrt{\frac{x(0)^\top x(0)}{\sum_{j=l}^{k_1} z_j^2(0)}} \sum_{i=l}^{k_1} z_i(0) S_i \in V_{\sigma_1},$$

as $t \to +\infty$. This completes the proof.

The above theorem gives necessary and sufficient condition for the network to converge to eigenvectors corresponding to the largest eigenvalue. These conditions depend on eigenspace V_{σ_1}. Since V_{σ_1} is not known *a prior*, it is not practical to choose initial values that are always orthogonal to V_{σ_1} in advance. This problem can be solved by giving the initial values some random perturbations, since the dimension of V_{σ_1} is always less than the dimension of R^n. In this way, one can obtain a high probability of having initial values that are not orthogonal to V_{σ_1}. Simulations in next section will further confirm this point.

Theorem 4 shows how to compute eigenvectors corresponding to the largest eigenvalue. In many applications, it is also very interesting to compute eigenvectors corresponding to the smallest eigenvalue. The next theorem provides such a method.

THEOREM 5. *Replacing A in network (1) with $-A$, and suppose $x(0)$ is a nonzero vector in R^n which is not orthogonal to V_{σ_m}, then the solution starting from $x(0)$ will converge to an eigenvector corresponding to the smallest eigenvalue of A.*

PROOF. Noting the fact that $-\sigma_m$ is the largest eigenvalue of $-A$, by applying Theorem 4 it follows that the solution of (1) starting from $x(0)$ will converge to an eigenvector v corresponding to the largest eigenvalue $-\sigma_m$ of $-A$. Then, $-Av = -\sigma_m v$, that is, $Av = \sigma_m v$. This shows that v is actually an eigenvector corresponding to the smallest eigenvalue σ_m of A. The proof is completed.

Theorem 5 shows how to compute eigenvectors corresponding to the smallest eigenvalue of A, just by simply replacing A in network (1) with $-A$.

THEOREM 6. *If $x(0)$ is orthogonal to each S_i $(i = 1, \ldots, c)$, where $c \leq n$ is a constant, then the solution of (1) starting from $x(0)$ converges to an eigenvector which is also orthogonal to each S_i $(i = 1, \ldots, n)$.*

PROOF. Let $x(0)$ be represented under the basis S_i $(i = 1, \ldots, n)$ as $x(0) = \sum_{i=1}^{n} z_i(0) S_i$. Since $x(0)$ is orthogonal to each S_i $(i = 1, \ldots, c)$, then $z_i(0) = 0$ $(i = 1, \ldots, c)$. Clearly,

$$c \leq l = \min\{i \mid 1 \leq i \leq n,\ z_i(0) \neq 0\}.$$

By the proof of Theorem 3, the solution starting from $x(0)$ converges to

$$\sqrt{\frac{x(0)^\top x(0)}{\sum_{j=l}^{k_r} z_j^2(0)}} \sum_{i=l}^{k_r} z_i(0) S_i \in V_{\sigma_r}$$

which is orthogonal to each S_i $(i = 1, \ldots, n)$. This completes the proof.

5. COMPUTER SIMULATION RESULTS

In this section, some computer simulation results will be given to illustrate the above theory. The simulation will show that the proposed network can calculate the eigenvectors corresponding to the largest and smallest eigenvalues of any symmetric matrix.

Symmetric matrix can be randomly generated in a simple way. Let Q be any randomly generated real matrix, define

$$A = \frac{Q^\top + Q}{2}.$$

Clearly, A is a symmetric matrix.

In the first example, a 5×5 symmetric matrix A is generated as

$$A = \begin{bmatrix} 0.7663 & 0.4283 & -0.3237 & -0.4298 & -0.1438 \\ 0.4283 & 0.2862 & 0.0118 & -0.2802 & 0.1230 \\ -0.3237 & 0.0118 & -0.9093 & -0.4384 & 0.7684 \\ -0.4298 & -0.2802 & -0.4384 & -0.0386 & -0.1315 \\ -0.1438 & 0.1230 & 0.7684 & -0.1315 & -0.4480 \end{bmatrix}.$$

Using the network model, the estimated eigenvectors corresponding to $\lambda_{\max}$ and $\lambda_{\min}$, respectively, are

$$\xi_{\max} = \begin{bmatrix} 1.0872 \\ 0.6264 \\ -0.0809 \\ -0.4736 \\ -0.0472 \end{bmatrix}, \qquad \xi_{\min} = \begin{bmatrix} 0.1882 \\ 0.0600 \\ 1.3209 \\ 0.3697 \\ -0.8446 \end{bmatrix}.$$

Applying the network model on A leads to Figure 1. The result is an estimation of the eigenvector, that is, $\xi_{\max}$ above, corresponding to the largest eigenvalue. An immediate following result is an estimation to the maximum eigenvalue of 1.2307, which is an accurate approximation to the true value with a precision of 0.0001. For the reader's reference, the true maximum and minimum eigenvalues as computed by MATLAB are shown here:

$$\begin{aligned} &\text{maximum eigenvalue} && \lambda_{\max} = 1.2307 \\ &\text{minimum eigenvalue} && \lambda_{\min} = -1.5688 \end{aligned}$$

Another advantage of the technique is that it allows the calculation of an eigenvector corresponding to the minimum eigenvalue. This is due to Theorem 5. The result is shown in Figure 2. By feeding the network with $-A$, it gets that $\xi_{\min}$, an estimation to the desired eigenvector, as well as the magnitude of the smallest eigenvalue, 1.5688, which is an accurate estimation just with the sign flipped. The generated eigenvector also corresponds to the target eigenvalue.

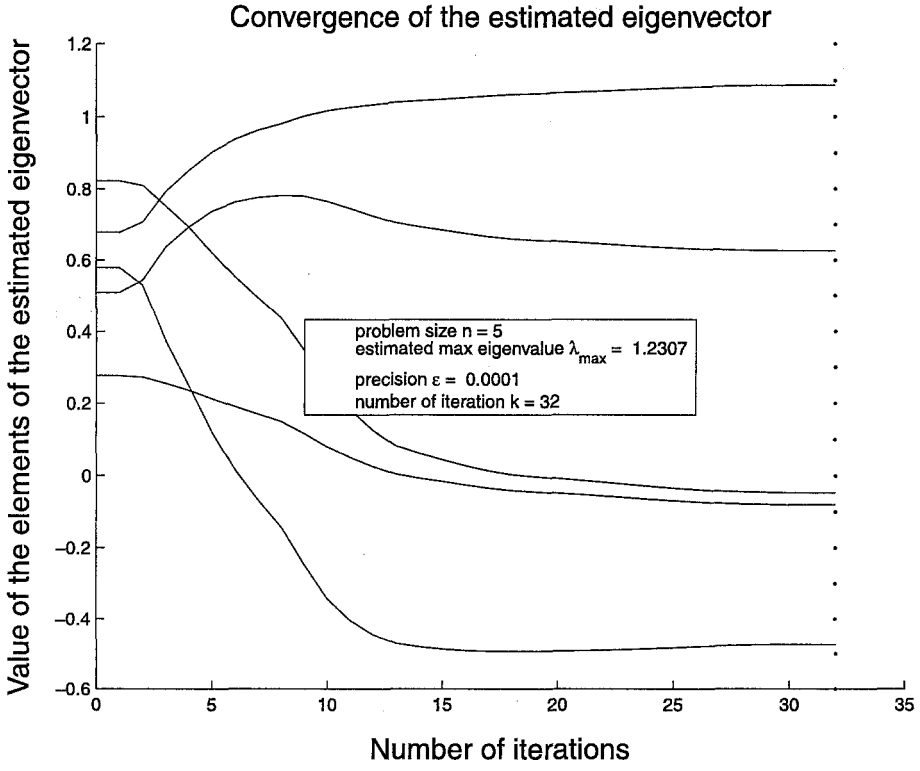

Figure 1. Estimation of an eigenvector corresponding to the largest eigenvalue of the matrix A.

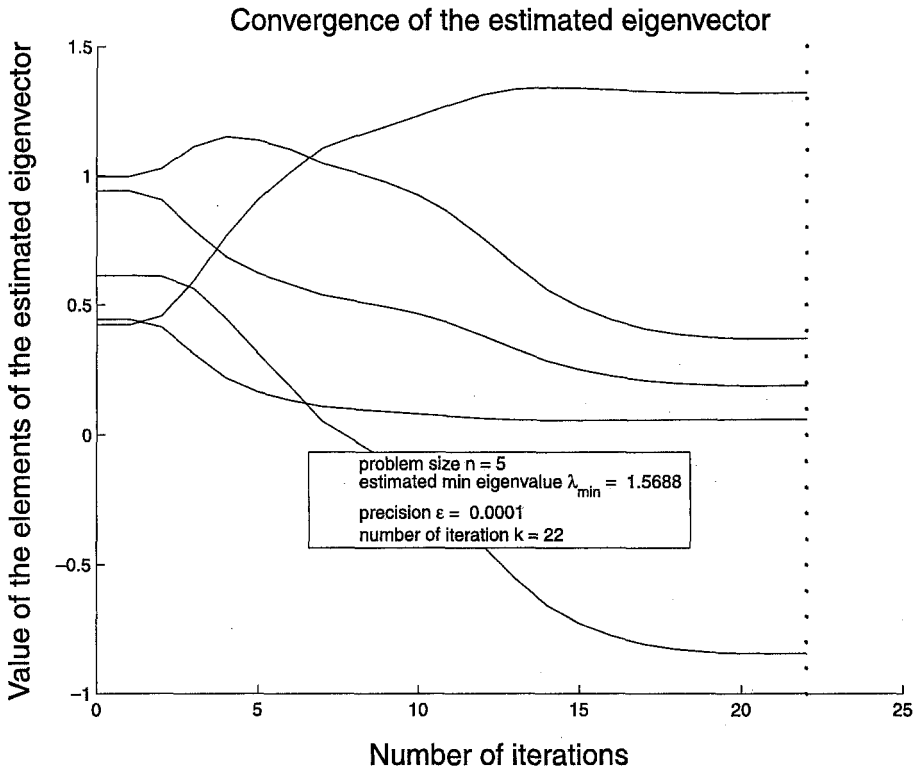

Figure 2. Estimation of an eigenvector corresponding to the smallest eigenvalue of the matrix A.

In the next example, a 7×7 symmetric matrix B is randomly generated as

$$B = \begin{bmatrix} 0.6037 & 0.5587 & -0.3907 & 0.1629 & -0.2079 & 0.3688 & 0.3990 \\ 0.5587 & 0.9918 & -0.1018 & 0.4110 & 0.1256 & -0.2908 & -0.0014 \\ -0.3907 & -0.1018 & -0.3977 & -0.2967 & 0.0273 & -0.3578 & 0.2106 \\ 0.1629 & 0.4110 & -0.2967 & -0.9101 & 0.6153 & 0.3519 & -0.3277 \\ -0.2079 & 0.1256 & 0.0273 & 0.6153 & -0.8451 & -0.0503 & -0.7252 \\ 0.3688 & -0.2908 & -0.3578 & 0.3519 & -0.0503 & 0.3475 & 0.1201 \\ 0.3990 & -0.0014 & 0.2106 & -0.3277 & -0.7252 & 0.1201 & -0.9565 \end{bmatrix}.$$

Using the network model, the estimated eigenvectors corresponding to $\lambda_{\max}$ and $\lambda_{\min}$, respectively, are

$$\xi_{\max} = \begin{bmatrix} 1.1439 \\ 1.3386 \\ -0.3907 \\ 0.3747 \\ 0.0256 \\ 0.2602 \\ 0.1042 \end{bmatrix}, \qquad \xi_{\min} = \begin{bmatrix} 0.0602 \\ -0.0392 \\ 0.2305 \\ 0.4682 \\ -0.8533 \\ -0.0371 \\ -0.6757 \end{bmatrix}.$$

Similarly, accurate estimations are produced using the model, for both eigenvectors corresponding to the maximum and minimum eigenvalues. Figures 3 and 4 show the convergence of the eigenvectors estimation of the network for both cases, respectively. The true maximum and minimum eigenvalues as computed by MATLAB are also listed below:

$$\begin{aligned} &\text{maximum eigenvalue} && \lambda_{\max} = 1.5599 \\ &\text{minimum eigenvalue} && \lambda_{\min} = -1.7461 \end{aligned}$$

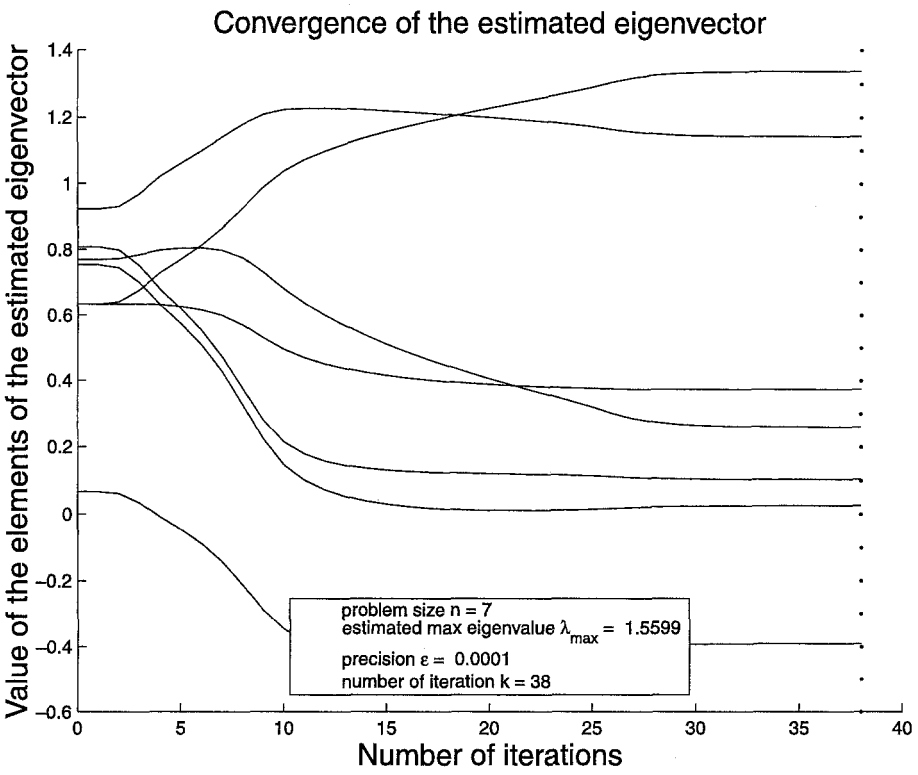

Figure 3. Estimation of an eigenvector corresponding to the largest eigenvalue of the matrix B.

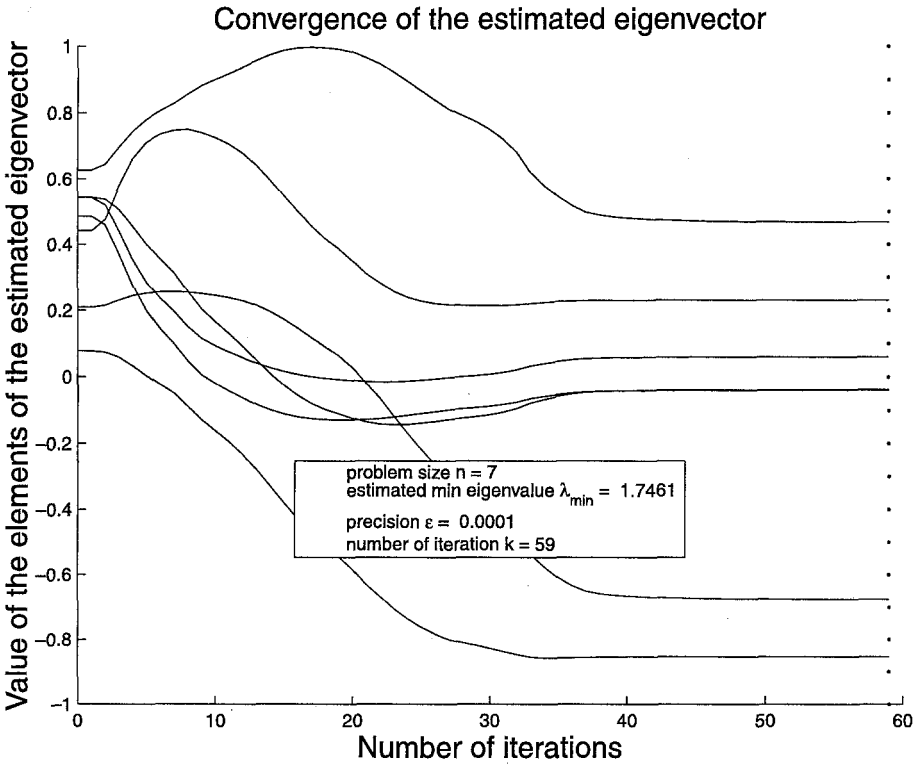

Figure 4. Estimation of an eigenvector corresponding to the smallest eigenvalue of the matrix B.

For both simulations, the desired precision is set to 0.0001. This accuracy level can be changed according to the need of the user. The network converges up to this predefined accuracy successfully, as shown theoretically and practically.

Besides the above simulations, many high-dimensional matrices are used for simulation to test the ability of the network. All the simulation results are satisfied. The simulation results further confirm that the network model can be used to compute the eigenvectors and eigenvalues of any real symmetrical matrix. It should be noted that the computer simulations shown here are used only for illustrating the capability of the network, the network is expected to achieve high computational performance by implementing the network in the hardware in the future.

6. CONCLUSIONS

The problem of computing eigenvectors of symmetric matrix by an approach of neural networks has been studied in this paper. A class of artificial recurrent neural networks has been proposed, which can be used to compute eigenvectors corresponding to the largest or smallest eigenvalues of any real symmetric matrix. This network model has asynchronous parallel processing ability and hence can achieve high computing performance. A rigorous and clear mathematical understanding of the dynamic behaviors of the network model has been provided. Simulation results show the network model works well. Through out this paper, the matrix A is assumed to be symmetric. For the case that the matrix A is not symmetric, it is not clear whether or not the network model can still work. Further study in this direction is required.

REFERENCES

1. K.I. Diamantaras, K. Hornik and M.G. Strintzis, Optimal linear compression under unreliable representation and robust PCA neural models, *IEEE Trans. Neural Networks* **10** (5), 1186–1195, (1999).
2. K. Hornik and C.M. Kuan, Convergence analysis of local feature extraction algorithms, *Neural Networks* **5**, 229–240, (1992).
3. F. Luo, R. Unbehauen and A. Cichocki, A minor component analysis algorithm, *Neural Networks* **10** (2), 291–297, (1997).
4. F. Luo and R. Unbehauen, A minor subspace analysis algorithm, *IEEE Trans. Neural Networks* **8** (5), 1149–1153, (1997).
5. F. Luo, R. Unbehauen and Y.D. Li, A principal component analysis algorithm with invariant norm, *Neurocomputing* **8**, 213–221, (1995).
6. G. Mathew and V.U. Reddy, Development and analysis of a neural network approach to Pisarenko's harmonic retrieval method, *IEEE Trans. Signal Processing* **42** (3), 663–667, (1994).
7. G. Mathew and V.U. Reddy, Orthogonal eigensubspace estimation using neural networks, *IEEE Trans. Signal Processing* **42** (7), 1803–1811, (1994).
8. G. Mathew, V.U. Reddy and S. Dasgupta, Adaptive estimation of eigensubspace, *IEEE Trans. Signal Processing* **43** (2), 401–411, (1995).
9. E. Oja and J. Karhunen, On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix, *J. of Math. Anal. Appl.* **106**, 69–84, (1985).
10. E. Oja, Principal components, minor components, and linear neural networks, *Neural Networks* **5**, 927–935, (1992).
11. K. Reif, F. Lou and R. Unbehauen, The exponential stability of the invariant norm PCA algorithm, *IEEE Trans. Circuits and Sys.-II* **44** (10), 873–876, (1997).
12. A. Taleb and G. Cirrincione, Against the convergence of the minor component analysis neurons, *IEEE Trans. Neural Networks* **10** (1), 207–210, (1999).
13. L. Xu, E. Oja and C. Suen, Modified Hebbian learning for curve and surface fitting, *Neural Networks* **5**, 441–457, (1992).
14. J.F. Yang and M. Kaveh, Adaptive eigensubspace algorithms for direction or frequency estimation and tracking, *IEEE Trans. Acoust., Speech, Signal Processing* **36** (2), 663–667, (1988).