An Adaptive Learning Algorithm for Principal Component Analysis

Liang-Hwa Chen and Shyang Chang

Abstract— **Principal component analysis (PCA) is one of the most general purpose feature extraction methods. A variety of learning algorithms for PCA has been proposed. Many conventional algorithms, however, will either diverge or converge very slowly if learning rate parameters are not properly chosen. In this paper, an adaptive learning algorithm (ALA) for PCA is proposed. By adaptively selecting the learning rate parameters, we show that the m weight vectors in the ALA converge to the first m principle component vectors with almost the same rates. Comparing with the Sanger's generalized Hebbian algorithm (GHA), the ALA can quickly find the desired principal component vectors while the GHA fails to do so. Finally, simulation results are also included to illustrate the effectiveness of the ALA.**

I. Introduction

FEATURE extraction from complex, high-dimensional input data streams is a fundamentally important task for many information processing fields such as pattern recognition, communication technology, etc. It can help eliminate information redundancy or disturbing noise and allow the relevant information contained in input signal to be expressed with lower dimension. As a result, information storage, transmission,and processing, not only in software but also in hardware, can become easier due to the data dimensionality reduction.

Principal component analysis (PCA) is one of the feature extraction methods. It extracts information by finding the directions in input space in which the inputs exhibit most variation (see, e.g., [1] and [18]). It then projects the inputs onto the subspace to perform a dimensionality reduction. The desired directions are in fact along those eigenvectors associated with the m largest eigenvalues of the input covariance matrix [17] if the dimension of data is expected to be reduced from n to m. Due to the unavoidable computational complexity with conventional matrix algebraic approaches, especially when n is very large, the neural network approach for PCA has been widely studied recently. Neural networks can solve cumbersome problems by connecting many simple processing units (i.e., neurons). They can learn directly from input patterns to get expected solutions. Only a simple learning algorithm associated with simple computations is required. Moreover, their architectures are suitable for hardware implementation with VLSI (very large scale integrated) circuit.

A variety of neural network learning algorithms for PCA has been proposed [2]–[9], [15]. Most of them are based on the early work of Oja's one-unit algorithm [10]. Among these algorithms, the Sanger's generalized Hebbian algorithm (GHA) [2] which combines Gram–Schmidt orthonormalization and Oja's one-unit algorithm is usually more useful in practical applications. This is because that it can extract the principal components individually in order. In addition, it can give a reproducible result on a given data set. The problem of how to automatically and adaptively select the learning rate parameters, however, was not adequately considered. As a result, the algorithm will either converge very slowly or even diverge if the values of the learning rate parameters are not properly selected. The stability and general strategies of choosing the values of the learning rate parameters of the Oja's one-unit learning rule and some other gradient type algorithms have been discussed in [12]–[14]. As for GHA, there is no algorithm that can automatically and adaptively select the values of the learning rate parameters to get fast convergence for all desired eigenvectors independent of eigenvalues. Such an algorithm is, however, imperative when one tries to apply the GHA in practical applications.

In this paper, an adaptive learning algorithm (ALA) for PCA is proposed. The learning rate parameters can be selected automatically and adaptively according to the eigenvalues of the input covariance matrix that are estimated during the learning process. We will show that the m weight vectors in the network can converge quickly to the first m principle component vectors with almost the same learning rates. In addition, the simulation results demonstrate that the ALA can converge quickly to the desired targets while the GHA diverges in the large eigenvalue case and converges very slowly in the small eigenvalue case.

This paper is organized as follows: In Section II, some basic mathematical background and the Sanger's GHA are introduced. The parameter selection problem of the GHA is then presented. In Section III, our ALA is proposed and analyzed. Propositions concerning its properties are also presented in this section. The simulation results of the ALA are presented in Section IV. Finally, conclusions are given in Section V.

II. Mathematical Background and Previous Work

Let $\mathbf{x}$ denote the n-dimensional input data vector with probability distribution $P(\mathbf{x})$. The aim of PCA is to find a set of m orthonormal vectors in an n-dimensional data space such that they will account for as much as possible of the variance of the data. It was shown in [1] that the aforementioned orthonormal vectors were actually the m eigenvectors associated with the m largest eigenvalues of the data covariance matrix $\mathbf{T} = E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^t\}$, where t denotes the transpose operator and $\mathbf{m}_x = E\{\mathbf{x}\}$. If the eigenvalues of $\mathbf{T}$ are sorted in descending order, i.e., $\lambda_1 > \lambda_2 > \cdots > \lambda_n$ with $\lambda_1 = \lambda_{\max}$, then the kth principal component direction will be along the kth eigenvector. In general, the mean values of data can be subtracted from the data. Hence, in the following, we will

Manuscript received May 8, 1993; revised April 18, 1994 and December 27, 1994.

The authors are with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.

IEEE Log Number 9409371.

discuss zero-mean data exclusively. In case of zero-mean data, the covariance matrix $\mathbf{T}$ will be reduced to the correlation matrix $\mathbf{C} = E\{\mathbf{x}\mathbf{x}^t\}$.

To find the first principal component direction vector for zero-mean data, i.e., the first eigenvector of $\mathbf{C}$, by learning directly from data, Oja [10] proposed a one-unit learning rule

$$\Delta\mathbf{w}(t) = \eta(t)V(t)(\mathbf{x}(t) - V(t)\mathbf{w}(t)) \tag{1}$$

where $\eta(t)$ is the learning rate parameter. This rule is used to train a linear neuron whose output $V(t)$ is equal to the product of weight vector $\mathbf{w}(t)$ and input pattern $\mathbf{x}(t)$ at time t, i.e., $V(t) = \mathbf{w}^t(t)\mathbf{x}(t)$. Here, we assume that $\mathbf{x}(t)$'s are independent and identically distributed with the same distribution $P(\mathbf{x})$ as before. Under the assumption that $\eta(t)$ is sufficiently small, Oja approximated (1) by a corresponding ODE $d\mathbf{w}/dt = \mathbf{C}\mathbf{w} - (\mathbf{w}^t\mathbf{C}\mathbf{w})\mathbf{w}$ via the stochastic approximation theory (see, e.g., [11]). He then proved that the weight vector $\mathbf{w}(t)$ will asymptotically converge to the first normalized eigenvector of $\mathbf{C}$, i.e., $\pm\mathbf{v}_1$.

By combining the Oja's rule and Gram–Schmidt orthonormalization process, Sanger [2] proposed the so-called GHA

$$\Delta\mathbf{w}_i(t) = \eta(t)V_i(t)\left(\mathbf{x}(t) - \sum_{j=1}^{i} V_j(t)\mathbf{w}_j(t)\right), \quad i = 1, 2, \cdots, m \tag{2}$$

where $V_i(t) = \mathbf{w}_i^t(t)\mathbf{x}(t)$. It is used to train a one-layer m-unit network consisting of m linear neurons as to find the first m principal components. Using the same approximation technique, GHA was able to make $\mathbf{w}_i(t)$, $i = 1, 2, \cdots, m$, converge to the first m principal component directions, in sequential order: $\mathbf{w}_i(t) \to \pm\mathbf{v}_i$, where $\mathbf{v}_i$ is a normalized eigenvector associated with the ith largest eigenvalue λ_i of the correlation matrix $\mathbf{C}$. In fact, (2) can be rewritten as

$$\Delta\mathbf{w}_i(t) = \eta(t)V_i(t)\left[\left(\mathbf{x}(t) - \sum_{j=1}^{i-1} V_j(t)\mathbf{w}_j(t)\right) - V_i(t)\mathbf{w}_i(t)\right], \quad i = 1, 2, \cdots, m. \tag{3}$$

Hence, (3) can be treated as (1) with the corresponding modified input $\mathbf{x}_i(t) \equiv \mathbf{x}(t) - \sum_{j=1}^{i-1} V_j(t)\mathbf{w}_j(t)$, for neuron i, where $i = 1, 2, \cdots, m$. If $\mathbf{w}_j(t)$, $j = 1, 2, \cdots, i-1$ have converged to $\mathbf{v}_j$, $j = 1, 2, \cdots, i-1$, respectively, it can be easily shown [2] that the maximal eigenvalue λ_1^i and the associated normalized eigenvector $\mathbf{v}_1^i$ of the correlation matrix of $\mathbf{x}_i$, i.e., $\mathbf{C}_i \equiv E\{\mathbf{x}_i\mathbf{x}_i^t\}$, are exactly the ith eigenvalue λ_i and the ith normalized eigenvector $\mathbf{v}_i$ of the correlation matrix of $\mathbf{x}$, i.e., $\mathbf{C}$, respectively. Hence, neuron i can find the ith normalized eigenvector of $\mathbf{C}$, i.e., $\pm\mathbf{v}_i$. In other words, the m neurons trained by (2) can be considered to be trained by (1) with their respective modified inputs, $\mathbf{x}_i$, $i = 1, 2, \cdots, m$.

In the following, we will show that the selection of $\eta(t)$ should depend on the eigenvalues λ_i's of the correlation matrix $\mathbf{C}$. If $\eta(t)$ is bigger than $1/\lambda_1$, the learning process cannot converge[1] as expected. In addition, the learning rate will become very slow if the product value of $\eta(t)\lambda_1$ is very small. First, let us take the conditional expectation of (1) over the input distribution $P(\mathbf{x})$ given weight vector $\mathbf{w}(t)$, i.e.,

$$\begin{aligned} &E\{\Delta\mathbf{w}(t) \mid \mathbf{w}(t)\} \\ &= E\{\eta(t)V(t)(\mathbf{x}(t) - V(t)\mathbf{w}(t)) \mid \mathbf{w}(t)\} \\ &= \eta(t)[E\{\mathbf{x}(t)\mathbf{x}^t(t)\}\mathbf{w}(t) - \mathbf{w}^t(t)E\{\mathbf{x}(t)\mathbf{x}^t(t)\}\mathbf{w}(t)\mathbf{w}(t)] \\ &= \eta(t)[\mathbf{C}\mathbf{w}(t) - \mathbf{w}^t(t)\mathbf{C}\mathbf{w}(t)\mathbf{w}(t)] \end{aligned} \tag{4}$$

where we have used the following facts for derivation: $E\{\mathbf{x}(t)\mathbf{x}^t(t)\mathbf{w}(t) \mid \mathbf{w}(t)\} = E\{\mathbf{x}(t)\mathbf{x}^t(t)\}\mathbf{w}(t) = \mathbf{C}\mathbf{w}(t)$, and $E\{\mathbf{w}^t(t)\mathbf{x}(t)\mathbf{x}^t(t)\mathbf{w}(t)\mathbf{w}(t) \mid \mathbf{w}(t)\} = \mathbf{w}^t(t)E\{\mathbf{x}(t)\mathbf{x}^t(t)\}\mathbf{w}(t)\mathbf{w}(t)$ where $\mathbf{x}(t)$ and $\mathbf{w}(t)$ are independent.

Proposition 1: For learning rule 1, if $\eta(t)$ selected is not smaller than $1/\lambda_1$, then weight vector $\mathbf{w}(t)$ will not converge to $\pm\mathbf{v}_1$ even if it is initially close to the target.

Proof: See Appendix. □

From Proposition 1 we know that $\eta(t)$ should be smaller than $1/\lambda_1$ to get the expected convergence. Under this condition, the learning rate can be estimated by the value of $\eta(t)\lambda_1$.

Proposition 2: When $\eta(t)\lambda_1 < 0.5$, the smaller the value of $\eta(t)\lambda_1$ is, the slower the convergence rate of the expectation of $\mathbf{w}(t)$ is.

Proof: See Appendix. □

According to these two propositions, for each neuron i in (2), the learning rate parameter $\eta(t)$ has to satisfy $\eta(t)\lambda_i < 1$ to converge. In the meantime, it cannot be too small to have a decent learning rate. The values of λ_i's, however, are usually unknown *a priori*. Therefore, to select properly the value of $\eta(t)$ becomes a problem when one tries to apply the GHA in practical applications. For example, if one of the eigenvalues $\lambda_i = 10^4$, the $\eta(t)$ must be smaller than 10^{-4} for $\mathbf{w}_i(t)$ to converge. In GHA, however, the identical value setting of $\eta(t)$ for all neurons will slow down the learning rate of $\mathbf{w}_j(t)$ if $\lambda_j \ll 10^4$ for $j > i$.

To overcome the aforementioned problem, an adaptive learning algorithm (ALA) for PCA will be proposed in the following section. In the algorithm, the learning rate parameter for each $\mathbf{w}_i(t)$ can be selected adaptively.

III. The ALA for PCA

For n-dimensional zero-mean input pattern vector $\mathbf{x}$ probability distribution $P(\mathbf{x})$, the ALA that will find the first m principle component vectors can be described as follows.

Step 1: Set weight vectors $\mathbf{w}_i(0) \in \mathbf{R}^n$ such that $\|\mathbf{w}_i(0)\|^2 \ll 1/2^2$ and estimate of eigenvalues $\hat{\lambda}_i(0) = \delta$ (a small positive number) >0 for $i = 1, 2, \cdots, m$.

Step 2: Draw a new pattern $\mathbf{x}(t)$ at time $t, t \geq 1$, and present it to the network as input.

Step 3: Calculate the output V_i's

$$V_i(t) = \mathbf{w}_i^t(t)\mathbf{x}(t). \quad i = 1, 2, \cdots, m.$$

[1] The convergence here is in the mean square sense.

[2] $\|\mathbf{y}\|$ denotes in this paper the length of a vector $\mathbf{y}$, i.e., $\|\mathbf{y}\| = (\mathbf{y}^t\mathbf{y})^{1/2}$.

Step 4: Estimate the eigenvalues λ_i's

$$\hat{\lambda}_i(t) = \hat{\lambda}_i(t-1) + \gamma(t)[(\mathbf{w}_i^t(t)\mathbf{x}_i(t)/\|\mathbf{w}_i(t)\|)^2 - \hat{\lambda}_i(t-1)], \quad i = 1, 2, \cdots, m \quad (5)$$

where $\mathbf{x}_i(t) \equiv \mathbf{x}(t) - \sum_{j=1}^{i-1} V_j(t)\mathbf{w}_j(t)$. The value of $\gamma(t)$ is set to be smaller than one and decreased to zero as t approaches ∞.

Step 5: Modify the weights $\mathbf{w}_i$'s

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta_i(t)V_i(t)\left[\mathbf{x}(t) - \sum_{j=1}^{i} V_j(t)\mathbf{w}_j(t)\right], \quad i = 1, 2, \cdots, m \quad (6)$$

where $\eta_i(t) = \beta_i(t)/\hat{\lambda}_i(t)$. The value of $\beta_i(t)$ is set to be smaller than $2(\sqrt{2}-1)$ and decreased to zero as t approaches ∞.

Step 6: Check the length of $\mathbf{w}_i$'s (see (7) shown at the bottom of the page).

Step 7: Increase the time t by one and go back to Step 2 for the next input pattern until all of the $\mathbf{w}_i$'s are mutually orthonormal.

Remarks:

i) The procedure of eigenvalue estimation in Step 4 is the Grossberg learning rule [16]. When the value of $\gamma(t)$ is set smaller than one and decreased to zero with time, $\hat{\lambda}_i(t)$, i.e., $\hat{\lambda}_1^i(t)$, can converge to the mean of $(\mathbf{w}_i^t\mathbf{x}_i/\|\mathbf{w}_i\|)^2$ in the mean square sense.

ii) Due to the fact that the estimates of λ_i's may be inaccurate during the initial period of learning process, the normalization process in Step 6 is required. The details will be described in Section IV.

iii) The reason for using the upper bound of $\beta_i(t)$ in Step 5 will be given in Proposition 3.

Notice that in the ALA, its learning rate parameters $\eta_i(t)$ are no longer the same for all neurons. They are adaptively selected according to the corresponding eigenvalues λ_i's that are estimated by (5). We will contend in the following that $\mathbf{w}_i(t)$ can quickly converge to $\pm\mathbf{v}_i$ for all i and that they will converge with nearly the same rate.

First, let us consider the convergence of $\mathbf{w}_i(t)$ to $\pm\mathbf{v}_i$ for all i. Recall that (6) can be written as

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta_i(t)V_i(t)(\mathbf{x}_i(t) - V_i(t)\mathbf{w}_i(t)) \quad (8)$$

where $\mathbf{x}_i(t) \equiv \mathbf{x}(t) - \sum_{j=1}^{i-1} V_j(t)\mathbf{w}_j(t)$ for all i. It can be considered just as the learning rule (1) applied to every neuron i with $\mathbf{x}_i$ as its corresponding modified input. Hence, in the following, it suffices to consider (8), where $\eta_i(t) = \beta_i(t)/\hat{\lambda}_1^i(t)$, for only one neuron i and show that it can converge to the first normalized eigenvector of its corresponding input correlation matrix $\mathbf{C}_i$.

The proof will be decomposed into several parts. It will be shown first that the mean of $\mathbf{w}_i(t)$ will approach $\mathbf{v}_1^i$. That is, its length will approach one and its angle from $\mathbf{v}_1^i$ will approach zero. We will then analyze the variance and show that it will decrease to zero. Notice that in the following discussion, $\mathbf{w}_i(t)$ is no longer assumed to be close to $\mathbf{v}_1^i$ initially as in Proposition 1.

Proposition 3: In the ALA, the mean of $\mathbf{w}_i(t)$ will approach the unit hypersphere in $\mathbf{R}^n$ space.

Proof: Let $w_i(t)$ stand for a realization of $\mathbf{w}_i(t)$ at time t. An orthonormal basis of $\mathbf{R}^n$, $\{\mathbf{u}_1^i, \mathbf{u}_2^i, \cdots, \mathbf{u}_n^i\}$, such that $\mathbf{u}_1^i$ is the unit vector along the direction of $w_i(t)$ can be constructed. Hence, $w_i(t)$ is represented as

$$w_i(t) = k_u^i(t)\mathbf{u}_1^i + \sum_{j=2}^{n} \varepsilon_{uj}^i(t)\mathbf{u}_j^i$$

where $k_u^i(t)$ is equal to the length of $w_i(t)$, i.e., $\|w_i(t)\|$, and $\varepsilon_{uj}^i(t) = 0$ for $j = 2, 3, \cdots, n$. Similar to (4), we can obtain from (8)

$$E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\} = \eta_i(t)[\mathbf{C}_i w_i(t) - w_i^t(t)\mathbf{C}_i w_i(t) w_i(t)] \quad (9)$$

where $E\{\cdot \mid w_i(t)\}$ stands for $E\{\cdot \mid \mathbf{w}_i(t) = w_i(t)\}$. Project (9) onto $\mathbf{u}_j^i$, $j = 1, 2, \cdots, n$, we get

$$\begin{aligned}(\mathbf{u}_1^i)^t E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\} (\equiv \Delta k_u^i(t)) &= (\mathbf{u}_1^i)^t \eta_i(t)\left[\mathbf{C}_i k_u^i(t)\mathbf{u}_1^i - k_u^i(t)^3 \cdot (\mathbf{u}_1^i)^t\mathbf{C}_i\mathbf{u}_1^i\mathbf{u}_1^i\right] \\ &= \eta_i(t)k_u^i(t)((\mathbf{u}_1^i)^t\mathbf{C}_i\mathbf{u}_1^i)(1 - k_u^i(t)^2), \quad \text{for } j = 1\end{aligned} \quad (10a)$$

and

$$\begin{aligned}(\mathbf{u}_j^i)^t E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\} (\equiv \Delta\varepsilon_{uj}^i(t)) &= (\mathbf{u}_j^i)^t \eta_i(t)\left[\mathbf{C}_i k_u^i(t)\mathbf{u}_1^i - k_u^i(t)^3 \cdot (\mathbf{u}_1^i)^t\mathbf{C}_i\mathbf{u}_1^i\mathbf{u}_1^i\right] \\ &= \eta_i(t)k_u^i(t)(\mathbf{u}_j^i)^t\mathbf{C}_i\mathbf{u}_1^i, \quad \text{for } j = 2, 3, \cdots, n.\end{aligned} \quad (10b)$$

The $\Delta k_u^i(t)$ defined in (10a) is the component of $E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\}$ along the direction of $\mathbf{u}_1^i$. It is referred to as the "radial weight change." The $\Delta\varepsilon_{uj}^i(t)$ defined in (10b) is the component of $E\{\Delta\mathbf{w}_i(t) \mid \mathbf{w}_i(t)\}$ along the direction of $\mathbf{u}_j^i$, $j > 1$. It is referred to as the "tangential weight change." Fig. 1 is a demonstration of the case $n = 2$. It is clear that from $w_i(t)$ to $E\{\mathbf{w}_i(t+1) \mid w_i(t)\}$, the change in length is caused by $\Delta k_u^i(t)$. On the other hand, the change in direction is caused by $\Delta\varepsilon_{uj}^i(t)$.

For $E\{\mathbf{w}_i(t+1) \mid w_i(t)\}$ to be closer to the unit hypersphere than $w_i(t)$, the value of $k_u^i(t) + \Delta k_u^i(t)$ has to be closer to one than $k_u^i(t)$. From (10a), we obtain

$$k_u^i(t) + \Delta k_u^i(t) = \left[1 + \eta_i(t)((\mathbf{u}_1^i)^t\mathbf{C}_i\mathbf{u}_1^i(1 - k_u^i(t)^2))\right]k_u^i(t). \quad (11)$$

$$\mathbf{w}_i(t+1) = \begin{cases} \sqrt{1/2}(\mathbf{w}_i(t+1)/\|\mathbf{w}_i(t+1)\|), & \text{if } \|\mathbf{w}_i(t+1)\|^2 > \frac{1}{\beta_i(t+1)} + \frac{1}{2}, \\ \mathbf{w}_i(t+1), & \text{otherwise.} \end{cases} \quad (7)$$

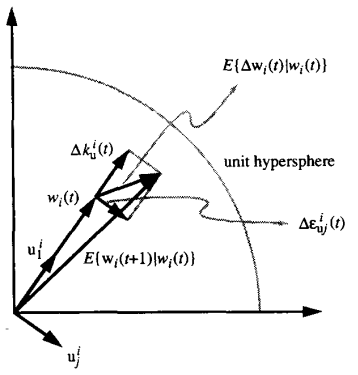

Fig. 1. Illustration of radial and tangential weight changes.

Take the square of both sides of (11) and let $\lambda_u^i \equiv (\mathbf{u}_1^i)^t \mathbf{C}_i \mathbf{u}_1^i$, we get

$$(k_u^i(t) + \Delta k_u^i(t))^2 = \left[1 + \eta_i(t)\lambda_u^i(1 - k_u^i(t)^2)\right]^2 k_u^i(t)^2. \tag{12}$$

Equation (12) takes the form of

$$z(t+1) = [1 + \alpha(1 - z(t))]^2 z(t) \tag{13}$$

where $z(t+1) = (k_u^i(t) + \Delta k_u^i(t))^2, z(t) = k_u^i(t)^2$ and $\alpha = \eta_i(t)\lambda_u^i$. It can be easily checked that for (13), the relation

$$| z(t+1) - 1 | / | z(t) - 1 | < 1 \tag{14}$$

holds if $\alpha \in (0, 2(\sqrt{2}-1))$ and $z(t) \in (0, 1+2/\alpha)$. Now, since λ_1^i is equal to $\max_{\mathbf{u}}(\mathbf{u}^t \mathbf{C}_i \mathbf{u})$ where $\mathbf{u}$ is any unit vector, thus $\eta_i(t)\lambda_u^i \le \eta_i(t)\lambda_1^i$. In addition, the value of $\eta_i(t)$ is selected such that $\eta_i(t)\lambda_1^i = \beta_i(t) < 2(\sqrt{2}-1)$. Hence, $\eta_i(t)\lambda_u^i < 2(\sqrt{2}-1)$. Moreover, $k_u^i(0)^2 (= \|w_i(0)\|^2)$ is set to be smaller than one in Step 1 and $w_i(t)$ will be bounded by $\|w_i(t)\|^2 \le 1/2 + 1/\beta_i(t) = 1/2 + 1/(\eta_i(t)\lambda_1^i) < 1 + 2/(\eta_i(t)\lambda_u^i)$ according to Step 6 of the ALA. Thus, both $(k_u^i(t) + \Delta k_u^i(t))^2$ and $k_u^i(t)^2$ satisfy (14), i.e., $| (k_u^i(t) + \Delta k_u^i(t))^2 - 1 | / | k_u^i(t)^2 - 1 | < 1$. That is, $E\{\mathbf{w}_i(t+1) \mid w_i(t)\}$ will be closer to the unit hypersphere than $w_i(t)$. Since this is true for all realizations $w_i(t)$, so $E\{\mathbf{w}_i(t+1) \mid \mathbf{w}_i(t)\}$ will be closer to the unit hypersphere than $\mathbf{w}_i(t)$. Hence, $E\{\mathbf{w}_i(t+1)\} = E\{E\{\mathbf{w}_i(t+1) \mid \mathbf{w}_i(t)\}\}$ will be closer to the unit hypersphere than $E\{\mathbf{w}_i(t)\}$. We conclude that the length of the mean of $\mathbf{w}_i(t)$ will approach one as t goes to ∞. □

Next, we will show that the direction of $\mathbf{w}_i(t)$ will approach that of $\mathbf{v}_1^i$.

Proposition 4: The angle between the mean of $\mathbf{w}_i(t)$ and $\mathbf{v}_1^i$ in the ALA will approach zero.

Proof: First, express $w_i(t)$ in the following form

$$w_i(t) = k^i(t)\mathbf{v}_1^i + \mathbf{s}_i(t) = k^i(t)\mathbf{v}_1^i + \sum_{j=2}^{n} \varepsilon_j^i(t)\mathbf{v}_j^i \tag{15}$$

where $k^i(t)$ is magnitude of $w_i(t)$ along eigenvector $\mathbf{v}_1^i$ and $\mathbf{s}_i(t) = \sum_{j=2}^n \varepsilon_j^i(t)\mathbf{v}_j^i$ the component of $w_i(t)$ perpendicular to $\mathbf{v}_1^i$. Notice that $\varepsilon_j^i(t)$ is the magnitude of $w_i(t)$ along $\mathbf{v}_j^i$. For a given $w_i(t)$, its average new location after one iteration of learning will be $E\{\mathbf{w}_i(t+1) \mid w_i(t)\} = w_i(t) + E\{\Delta\mathbf{w}_i(t) \mid w_i(t)$. Thus

$$(\mathbf{v}_1^i)^t + E\{\mathbf{w}_i(t+1) \mid w_i(t)\} (\equiv k^i(t+1)) = k^i(t) + (\mathbf{v}_1^i)^t E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\}$$

and

$$(\mathbf{v}_j^i)^t E\{\mathbf{w}_i(t+1) \mid w_i(t)\} (\equiv \varepsilon_j^i(t+1)) = \varepsilon_j^i(t) + (\mathbf{v}_j^i)^t E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\} \quad \text{for } j = 2, 3, \cdots, n.$$

According to (9) and (15), $(\mathbf{v}_j^i)^t E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\}$, $j = 1, 2, \cdots, n$, can be written as

$$(\mathbf{v}_1^i)^t E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\} = \eta_i(t)(\lambda_1^i - \sigma_\lambda^i(t))k^i(t) \tag{16a}$$

and

$$(\mathbf{v}_j^i)^t E\{\Delta\mathbf{w}_i(t) \mid w_i(t)\} = \eta_i(t)(\lambda_j^i - \sigma_\lambda^i(t))\varepsilon_j^i(t), \quad j = 2, 3, \cdots, n \tag{16b}$$

where $\sigma_\lambda^i(t) = w_i^t(t)\mathbf{C}_i w_i(t) = \lambda_1^i k^i(t)^2 + \sum_{j=2}^n \lambda_j^i \varepsilon_j^i(t)^2$. Then, we get

$$k^i(t+1) = [1 + \eta_i(t)(\lambda_1^i - \sigma_\lambda^i(t))]k^i(t)$$

and

$$\varepsilon_j^i(t+1) = [1 + \eta_i(t)(\lambda_j^i - \sigma_\lambda^i(t))]\varepsilon_j^i(t).$$

Let us denote the angle between $\mathbf{w}_i$ and $\mathbf{v}_1^i$ by $\mathrm{Ang}(\mathbf{w}_i)$. Then, $\tan^2(\mathrm{Ang}(w_i)) = \|\mathbf{s}_i\|^2/(k^i)^2 = \sum_{j=2}^n (\varepsilon_j^i)^2/(k^i)^2$. To prove $\mathrm{Ang}(E\{\mathbf{w}_i(t+1) \mid w_i(t)\})$ that will be smaller than $\mathrm{Ang}(w_i(t))$ it suffices to show that $\tan^2(\mathrm{Ang}(E\{\mathbf{w}_i(t+1) \mid w_i(t)\}))$ will be smaller than $\tan^2(\mathrm{Ang}(w_i(t)))$. That is

$$\frac{\varepsilon_j^i(t+1)^2}{k^i(t+1)^2} = \frac{[1 + \eta_i(t)(\lambda_j^i - \sigma_\lambda^i(t))]^2}{[1 + \eta_i(t)(\lambda_1^i - \sigma_\lambda^i(t))]^2} \frac{\varepsilon_j^i(t)^2}{k^i(t)^2} < \frac{\varepsilon_j^i(t)^2}{k^i(t)^2} \quad \text{for } j = 2, 3, \cdots, n. \tag{17}$$

It can be easily checked that if

$$\|w_i(t)\|^2 \le 1/(\eta_i(t)\lambda_1^i) + 1/2 \tag{18}$$

then $\sigma_\lambda^i(t) \le \lambda_1^i \|w_i(t)\|^2 \le (2 + \eta_i(t)\lambda_1^i)/(2\eta_i(t))$ and then $[1 + \eta_i(t)(\lambda_j^i - \sigma_\lambda^i(t))]^2/[1 + \eta_i(t)(\lambda_1^i - \sigma_\lambda^i(t))]^2 < 1$, for $j = 2, 3, \cdots, n$. That is, (17) can hold. Since $w_i(t)$ is, by (7), bounded such that $\|w_i(t)\|^2 \le 1/2 + 1/\beta_i(t) = 1/2 + 1/(\eta_i(t)\lambda_1^i)$, condition (18) will be satisfied. Thus, $\mathrm{Ang}(E\{\mathbf{w}_i(t+1) \mid w_i(t)\}$ will be smaller than $\mathrm{Ang}(w_i(t))$. Since this is true for all realizations $w_i(t)$, so $E\{\mathbf{w}_i(t+1) \mid \mathbf{w}_i(t)\}$ will have smaller angle from $\mathbf{v}_1^i$ than $\mathbf{w}_i(t)$. Hence, $E\{\mathbf{w}_i(t+1)\} = E\{E\{\mathbf{w}_i(t+1) \mid \mathbf{w}_i(t)\}\}$ will have smaller angle from $\mathbf{v}_1^i$ than $E\{\mathbf{w}_i(t)\}$. That is, the angle between the mean of $\mathbf{w}_i(t)$ and $\mathbf{v}_1^i$ will approach zero. □

In the following, we will analyze the variance of $\Delta\mathbf{w}_i(t)$, i.e., $\mathrm{Var}(\Delta\mathbf{w}_i(t))$.

Proposition 5: Given that $\mathbf{x}$ is normally distributed and $\|\mathbf{w}_i(t)\| \le 1$, then $\mathrm{Var}(\Delta\mathbf{w}_i(t))$ is bounded above by $3(n-1)(\eta_i(t)\lambda_1^i)^2$ where n is the dimension of the input pattern.

Proof: Recall that $\mathrm{Var}(\Delta\mathbf{w}_i(t)) = E\{\|\Delta\mathbf{w}_i(t)\|^2\} - \|E\{\Delta\mathbf{w}_i(t)\}\|^2 \le E\{\|\Delta\mathbf{w}_i(t)\|^2\}$. According to (8), we get $E\{\|\Delta\mathbf{w}_i(t)\|^2 \mid w_i(t)\} = \eta_i^2(t)\, E\{V_i^2(t)\|\mathbf{x}_i(t) - V_i(t)w_i(t)\|^2 \mid w_i(t)\} = \eta_i^2(t)\, E\{V_i^2(t)(\|\mathbf{x}_i(t)\|^2 - 2V_i^2(t) + V_i^2(t)\|w_i(t)\|^2) \mid w_i(t)\}$. If $\|w_i(t)\| \le 1$, then $E\{\|\Delta\mathbf{w}_i(t)\|^2 \mid w_i(t)\} \le \eta_i^2(t)E\{V_i^2(t)(\|\mathbf{x}_i(t)\|^2 - V_i^2(t)) \mid w_i(t)\} \le \sum_{j=2}^{n} \eta_i^2(t)E\{((\mathbf{u}_1^i)^t\mathbf{x}_i(t))^2((\mathbf{u}_j^i)^t\mathbf{x}_i(t))^2\}$, where $\mathbf{u}_j^i$, $j = 1, 2, \cdots, n$, have been defined in Proposition 3. Notice that $E\{((\mathbf{u}_1^i)^t\mathbf{x}_i(t))^2((\mathbf{u}_j^i)^t\mathbf{x}_i(t))^2\} \le 3E\{((\mathbf{u}_1^i)^t\mathbf{x}_i(t))^2\} \cdot E\{((\mathbf{u}_j^i)^t\mathbf{x}_i(t))^2\}$ if $\mathbf{x}_i$ is normally distributed. In addition, $E\{(\mathbf{u}^t\mathbf{x}_i)^2\} = \mathbf{u}^t E\{\mathbf{x}_i\mathbf{x}_i^t\}\mathbf{u} = \mathbf{u}^t\mathbf{C}_i\mathbf{u} \le (\mathbf{v}_1^i)^t\mathbf{C}_i\mathbf{v}_1^i = \lambda_1^i$, where $\mathbf{u}$ is any unit vector. Thus, $E\{\|\Delta\mathbf{w}_i(t)\|^2 \mid w_i(t)\} \le \sum_{j=2}^{n} 3\eta_i(t)^2((\mathbf{v}_1^i)^t\mathbf{C}_i\mathbf{v}_1^i)^2 = 3(n-1)(\eta_i(t)\lambda_1^i)^2$ for $\|w_i(t)\| \le 1$. Since it is true for all realizations $w_i(t)$ with $\|w_i(t)\| \le 1$, so $E\{E\{\|\Delta\mathbf{w}_i(t)\|^2 \mid \mathbf{w}_i(t)\} \mid \|\mathbf{w}_i(t)\| \le 1\} = E\{\|\Delta\mathbf{w}_i(t)\|^2 \mid \|\mathbf{w}_i(t)\| \le 1\} \le 3(n-1)(\eta_i(t)\lambda_1^i)^2$. Thus, $\mathrm{Var}(\Delta\mathbf{w}_i(t))$ for $\|\mathbf{w}_i(t)\| \le 1$ will be bounded above by

$$3(n-1)(\eta_i(t)\lambda_1^i)^2. \tag{19}$$

□

The variance $\mathrm{Var}(\Delta\mathbf{w}_i(t))$ can be decomposed into the sum of the variances along $\mathbf{u}_j^i$'s, i.e., $\mathrm{Var}(\Delta\mathbf{w}_i(t)) = \sum_{j=1}^{n} \mathrm{Var}((\mathbf{u}_j^i)^t\Delta\mathbf{w}_i(t))$. Here, the radial variance of $\Delta\mathbf{w}_i(t)$, i.e., the variance along the direction of $w_i(t)$, is

$$\begin{aligned}&\mathrm{Var}((\mathbf{u}_1^i)^t\Delta\mathbf{w}_i(t) \mid w_i(t))\\ &\quad = E\{((\mathbf{u}_1^i)^t\Delta\mathbf{w}_i(t))^2 \mid w_i(t)\} - (\Delta k_u^i(t))^2\end{aligned} \tag{20a}$$

and the tangential variances of $\Delta\mathbf{w}_i(t)$, i.e., the variances along the directions perpendicular to $w_i(t)$, are

$$\begin{aligned}&\mathrm{Var}((\mathbf{u}_j^i)^t\Delta\mathbf{w}_i(t) \mid w_i(t))\\ &\quad = E\{((\mathbf{u}_j^i)^t\Delta\mathbf{w}_i(t))^2 \mid w_i(t)\} - (\Delta\varepsilon_{uj}^i(t))^2,\\ &\qquad\qquad j = 2, 3, \cdots, n.\end{aligned} \tag{20b}$$

According to (8), we get

$$\begin{aligned}E\{(\mathbf{u}_1^i)^t\Delta\mathbf{w}_i(t))^2 \mid w_i(t)\} &= \eta_i(t)^2 k_u^i(t)^2(1 - k_u^i(t)^2)^2\\ &\quad \times E\{((\mathbf{u}_1^i)^t\mathbf{x}_i(t)\mathbf{x}_i^t(t)\mathbf{u}_1^i)^2\}\end{aligned} \tag{21a}$$

$$\begin{aligned}E\{(\mathbf{u}_j^i)^t\Delta\mathbf{w}_i(t))^2 \mid w_i(t)\} &= \eta_i(t)^2 k_u^i(t)^2\\ &\quad \times E\{((\mathbf{u}_j^i)^t\mathbf{x}_i(t)\mathbf{x}_i^t(t)\mathbf{u}_1^i)^2\},\\ &\qquad j = 2, 3, \cdots, n.\end{aligned} \tag{21b}$$

Substituting the square of (10) and (21) into (20), we get

$$\begin{aligned}&\mathrm{Var}((\mathbf{u}_1^i)^t\Delta\mathbf{w}_i(t) \mid w_i(t))\\ &\quad = \eta_i(t)^2 k_u^i(t)^2(1 - k_u^i(t)^2)^2\\ &\qquad \times \left[E\{((\mathbf{u}_1^i)^t\mathbf{x}_i(t)\mathbf{x}_i^t(t)\mathbf{u}_1^i)^2\} - ((\mathbf{u}_1^i)^t\mathbf{C}_i\mathbf{u}_1^i)^2\right]\end{aligned} \tag{22a}$$

$$\begin{aligned}&\mathrm{Var}((\mathbf{u}_j^i)^t\Delta\mathbf{w}_i(t) \mid w_i(t))\\ &\quad = \eta_i(t)^2 k_u^i(t)^2\left[E\{((\mathbf{u}_j^i)^t\mathbf{x}_i(t)\mathbf{x}_i^t(t)\mathbf{u}_1^i)^2\}\right.\\ &\qquad\qquad \left. - ((\mathbf{u}_j^i)^t\mathbf{C}_i\mathbf{u}_1^i)^2\right], \quad j = 2, 3, \cdots n.\end{aligned} \tag{22b}$$

It is obvious that the tangential variances $\mathrm{Var}\,((\mathbf{u}_j^i)^t\Delta\mathbf{w}_i(t) \mid w_i\,(t))$ increase as the length of $w_i(t)$, i.e., $\|w_i(t)\|$ or $k_u^i(t)$, increases. On the other hand, the radial variance $\mathrm{Var}((\mathbf{u}_1^i)^t\Delta\mathbf{w}_i(t) \mid w_i(t))$ vanishes as the length of $w_i(t)$ approaches one. It leads to the following propositions:

Proposition 6: When $\mathbf{w}_i(t)$ reaches the stable fixed point $\mathbf{v}_1^i$, it will fluctuate around $\mathbf{v}_1^i$ only in direction but not in length. In addition, the range θ_f can be estimated by $\theta_f \le \tan^{-1}(\sqrt{3(n-1)}\eta_i(t)\lambda_1^i)$.

Proof: Take $\mathbf{w}_i(t) = \mathbf{v}_1^i$. Then, $k_u^i(t) = 1$ and $\mathbf{u}_1^i = \mathbf{v}_1^i$. As a result, the radial variance $\mathrm{Var}((\mathbf{u}_1^i)^t\Delta\mathbf{w}_i(t) \mid \mathbf{v}_1^i)$ [referring to (22a)] vanishes because $k_u^i(t) = 1$. That is, there is no radial fluctuation of $\mathbf{w}_i(t)$ to influence its length. On the other hand, the tangential variance $\mathrm{Var}((\mathbf{u}_j^i)^t\Delta\mathbf{w}_i(t) \mid \mathbf{v}_1^i)$ [referring to (22b)] still exists. That is, there exists tangential fluctuation of $\mathbf{w}_i(t)$. Hence, $\mathbf{w}_i(t)$ fluctuates only in direction not in length. According to Proposition 5, the range θ_f of such fluctuation in direction can be estimated as follows

$$\begin{aligned}&\tan^{-1}\left.\frac{\sqrt{\mathrm{Var}(\Delta\mathbf{w}_i(t))}}{\|\mathbf{w}_i(t)\|}\right|_{\mathbf{w}_i(t)=\mathbf{v}_1^i}\\ &\quad \le \tan^{-1}\frac{\sqrt{3(n-1)}\eta_i(t)\lambda_1^i}{1}\\ &\quad = \tan^{-1}(\sqrt{3(n-1)}\eta_i(t)\lambda_1^i).\end{aligned} \tag{23}$$

□

In accordance with the above propositions, it can be seen that the learning rate of $\mathbf{w}_i(t)$ and its variance can be estimated by the value of $\eta_i(t)\lambda_1^i$. For instance, from (19) and (22), one can see that the size of the variance $\mathrm{Var}(\Delta\mathbf{w}_i(t))$ can be estimated by the value of $(\eta_i(t)\lambda_1^i)^2$. It decreases to zero as $(\eta_i(t)\lambda_1^i)^2$ decreases to zero. Since the value of $\beta_i(t)(= \eta_i(t)\lambda_1^i)$ is monotone decreasing, $\mathbf{w}_i(t)$ will then converge to $\mathbf{v}_1^i$ in the mean square sense due to the decreasing variance. On the other hand, the learning rate of the length of $\mathbf{w}_i(t)$ depends, from (12), on the value of $\eta_i(t)\lambda_u^i$. It increases as $\eta_i(t)\lambda_u^i$ increases. Since $\lambda_u^i < \lambda_1^i$, we can then use $\eta_i(t)\lambda_1^i$ to estimate the rate. Similarly, from (17), the learning rate of the direction of $\mathbf{w}_i(t)$ depends on the values of $\eta_i(t)\lambda_j^i$'s. It increases as $\eta_i(t)\lambda_j^i$'s increase for given data set. Since λ_1^i is the largest eigenvalue, it is then reasonable to estimate the rate by using the value of $\eta_i(t)\lambda_1^i$. Therefore, for different neurons corresponding to different eigenvalues, i.e., λ_1^i, the same level of learning rate can be obtained by choosing $\eta_i(t)$ such that the values of $\eta_i(t)\lambda_1^i$'s are the same. Hence, the following proposition can be obtained.

Proposition 7: The learning rates of all $\mathbf{w}_i(t)$ in the ALA are nearly the same if $\beta_i(t)$ is the same for all i.

Proof: Since $\beta_i(t)$ is the same for all i, the value of $\eta_i(t)\lambda_1^i$ will be the same for all i. Hence, the learning rates of all $\mathbf{w}_i(t)$ in the ALA will be nearly the same due to previous discussion. □

According to Proposition 7, the learning rate of $\mathbf{w}_i(t)$ will not decrease as that of GHA when λ_i decreases. Hence, the learning of ALA can be faster than that of GHA. Simulation results in the next section will confirm the effectiveness of the ALA.

IV. Simulation Results

First, let us demonstrate that the ALA can converge quickly to the desired target in small eigenvalue case as well as large

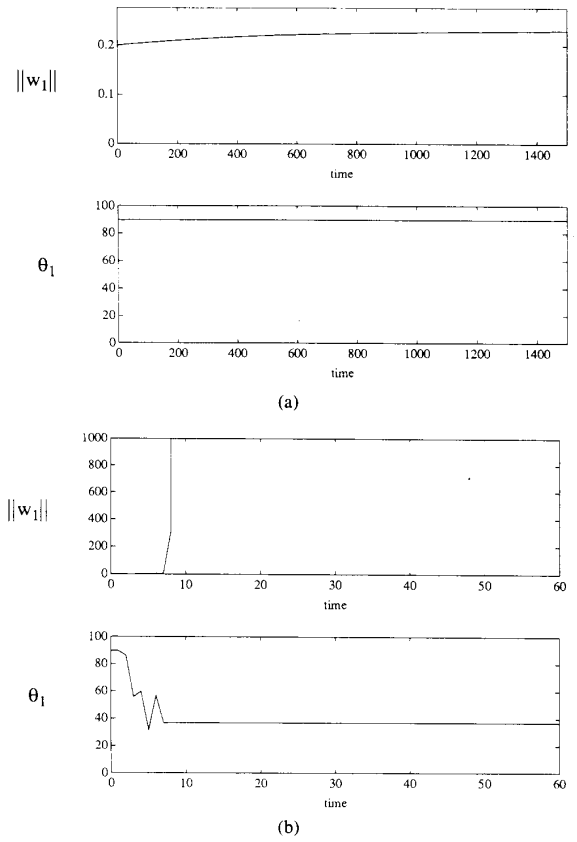

Fig. 2. Simulation results of the GHA for (a) $\lambda_1 = 0.0086$, (b) $\lambda_1 = 25.40$.

eigenvalue case while the GHA fails to do so. Two sets of three-dimensional randomly generated zero-mean data are adopted as input. The maximal eigenvalue λ_1 of the covariance matrixes of the two data sets are 0.0086 and 25.40, respectively. One neuron, i.e., $m = 1$, is considered for the moment. The weight vector $\mathbf{w}_1(t)$ is initially set to be nearly perpendicular to its target $\mathbf{v}_1$ with length about 0.2. Such an initial setting is adopted in all of the following experiments. The GHA (it is reduced to the Oja's rule here since $m = 1$) is first used to train the network. The value of $\eta(t)$ is set to be exponentially decreased with time from 0.1 to the final value 0.008. Its time function is set as $\max(0.1(0.01/0.1)^{t/1000}, 0.008)$. Fig. 2(a) and (b) are the time histories of learning corresponding to the two data sets, respectively. From Fig. 2(a), one can see that the convergence of the learning process is very slow. The angle $\theta_1(t)$ between $\mathbf{w}_1(t)$ and $\mathbf{v}_1$ is still near 90 degrees and the length of $\mathbf{w}_1(t)$, i.e.,$\|\mathbf{w}_1(t)\|$, is still much smaller than one. The reason is that the value of λ_1 (0.0086) is so small such that $\eta(t)\lambda_1$ is even smaller and the convergence rate for the learning process is very slow. On the other hand, if the same value of $\eta(t)$ is used for the other data set with $\lambda_1 = 25.40$, the learning process will fail because it is too big for this set of data. As shown in Fig. 2(b), $\|\mathbf{w}_1(t)\|$ grows to infinity and $\theta_1(t)$ cannot decrease to zero.

On the contrary, the ALA can succeed in both cases with the parameters $\beta_1(t)$ set as the $\eta(t)$ of GHA mentioned above and $\gamma(t)$ set as a constant value 0.01. First, let us discuss the procedure of eigenvalue estimation. In our experiments, $\mathbf{w}_1(t)$ is initially set far from $\mathbf{v}_1$ purposely, then the $\hat{\lambda}_1$ estimated by (5) is much smaller than its true value during the initial period of learning process. As a result, $\eta_1(t)(= \beta_1(t)/\hat{\lambda}_1(t))$ becomes much bigger than the desired value $\beta_1(t)/\lambda_1$. That is, $\eta_1(t)\lambda_1$ becomes much bigger than $\beta_1(t)$ the value we set, or even the upper bound $2(\sqrt{2}-1)$. According to (11), the length of $\mathbf{w}_1(t)$ will diverge. This minor problem, however, can be remedied by the normalization procedure in Step 6 of the ALA. In Step 6, once $\|\mathbf{w}_1(t)\| > \sqrt{1/\beta_1(t) + 1/2}$, it is normalized to $\sqrt{0.5}$. Otherwise, no normalization is required. From (18), one can see that the directional convergence of $\mathbf{w}_1(t)$ will then hold. Moreover, the convergence rate will, from (17), be faster with the bigger value of $\eta_1(t)\lambda_1$. Hence, $\mathbf{w}_1(t)$ will be close to $\mathbf{v}_1$ in direction and the mean of $(\mathbf{w}_1^t\mathbf{x}/\|\mathbf{w}_1\|)^2$ will then approach the desired value λ_1. As a result, $\hat{\lambda}_1(t)$ will converge very quickly. Fig. 3(a) and Fig. 4(a) clearly illustrate this point for these two data sets. with the accurate estimate of eigenvalue, $\mathbf{w}_1(t)$ will converge to $\mathbf{v}_1$ in the mean square sense as indicated by Propositions 3–6. It is illustrated in Fig. 3(b) and Fig. 4(b). One can see from these figures that the length of $\mathbf{w}_1(t)$ and the angle $\theta_1(t)$ between $\mathbf{w}_1(t)$ and $\mathbf{v}_1$ converge quickly to one and zero, respectively, for both data sets. To sum up, the learning process of ALA is successful for both data sets.

In the following, simulations will be used to demonstrate that the ALA can make all of the m weight vectors $\mathbf{w}_i(t), i = 1, 2, \cdots, m$, converge quickly to the desired targets independent of the eigenvalues and eigenvalue spread. Two data sets: Sandpapers and XO8 are used here. The former contains 96 four-dimensional patterns describing the texture measurements of the images of four kinds of sand and the latter contains 45 eight-dimensional patterns describing the characters "X," "O," and "8." For such nonzero-mean data, the ALA can still handle well as zero-mean data by estimating the data mean and subtracting it from the patterns. We estimate the data mean with the equation $\mathbf{m}_x(t+1) = \mathbf{m}_x(t)+(\mathbf{x}(t)-\mathbf{m}_x(t))/(t+1)$. Notice that the patterns in the data set are drawn randomly and repeatedly as the inputs presented to the network. In addition, the number of output, m, is set to n now. The parameters $\beta_i(t)$. for $i = 1, 2, \cdots, m$, are all set the same value as $\beta_1(t)$ in the previous experiments except time delay and the final value denoted by β_f. For the time delay, the learning time of neuron i starts later than that of neuron $i-1$ with a time delay t_p which is set to 500. The goal is to make all $\mathbf{w}_j(t)$ come close to $\mathbf{v}_j$ for $j < i$ when neuron i begins to learn. Moreover, to make the final angle error between $\mathbf{w}_i(t)$ and $\mathbf{v}_i$ be smaller than 1.5 degrees, the final value β_f of $\beta_i(t)$ is set, according to (23), to 0.008 for Sandpapers and 0.005 for higher dimensional XO8, respectively. The result for Sandpapers is shown in Fig. 5. It is clear that all of $\mathbf{w}_i(t)$'s can converge quickly to their corresponding $\mathbf{v}_i$'s, respectively, even when the second and third eigenvalues are very close. Moreover, one should notice that, although the differences among the eigenvalues are great, the learning rates of neurons are all nearly the same after they start learning. The eigenvalue spread λ_1/λ_n reaches about 200. The learning rate of neuron i in

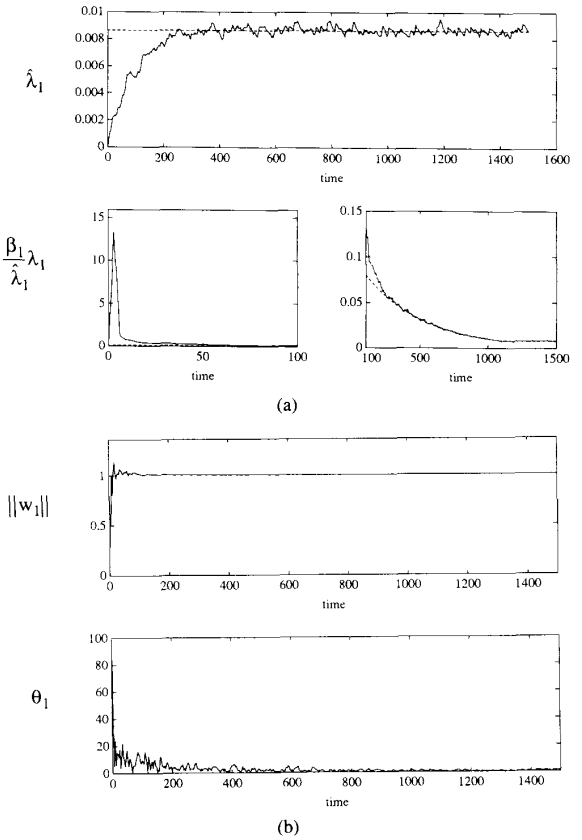

Fig. 3. Simulations results of the ALA for $\lambda_1 = 0.0086$. (a) Time histories of $\hat{\lambda}_1$ (top figure) and $(\beta_1/\hat{\lambda}_1)\lambda_1$ (bottom figure for two different time scales). The dashed line and curve denote the values of λ_1 and $\beta_1(t)$, respectively. (b) Time histories of $\|\mathbf{w}_1\|$ and θ_1 as functions of iterations.

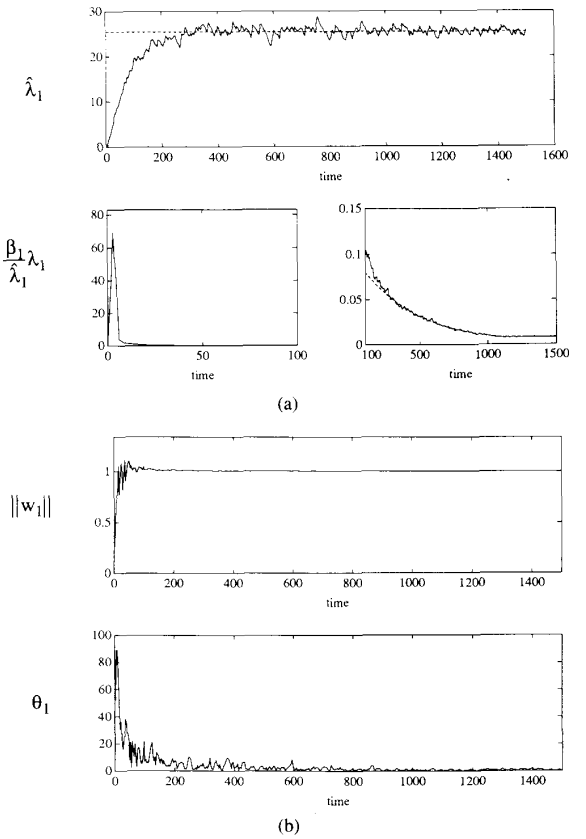

Fig. 4. Simulation results of the ALA for $\lambda_1 = 25.40$. (a) Time histories of $\hat{\lambda}_1$ (top figure) and $(\beta_1/\hat{\lambda}_1)\lambda_1$ (bottom figure for two different time scales). The dashed line and curve denote the values of λ_1 and $\beta_1(t)$, respectively. (b) Time histories of $\|\mathbf{w}_1\|$ and θ_1 as functions of iterations.

ALA, however, will not be slowed down as i increases. Table I lists the final values of $\|\mathbf{w}_i(t)\|$'s and $\theta_i(t)$'s as well as the eigenvalue estimates $\hat{\lambda}_i(t)$'s at the end of the learning process. These results are quite accurate compared with the theoretical values. Table II is the simulation result for the data set XO8. It demonstrates that the ALA also works well for higher dimensional data.

From these experiments, it is clear that the ALA can properly and automatically select the learning rate parameters such that $\mathbf{w}_i(t)$ can converge to $\mathbf{v}_i$ with almost the same rate for each i no matter what values the eigenvalues λ_i's are. Hence, it is believed that the ALA is a very effective way to execute PCA.

V. Conclusion

An ALA for PCA has been proposed in this paper. By adaptively selecting the learning rate parameters according to the eigenvalues of the input covariance matrix, we have shown that the ALA can make the m weight vectors in the network converge quickly to the first m principle component vectors with almost the same learning rates. From the simulation results, it has been confirmed that the ALA can converge very quickly to the desired target in large eigenvalue case as well as in small eigenvalue case. On the other hand, the conventional GHA diverges in the former case and converges very slowly in the later case. Moreover, from the simulation results of the two real data sets: Sandpapers and XO8, one can see that the ALA has been able to find quickly all principle component vectors even if the eigenvalue spread reaches about 200. Hence, it is believed that the ALA will make the neural network method of PCA more effective and feasible in practical applications.

Appendix

Proof of Proposition 1: Take $\mathbf{w}(t)$ to be close to $\mathbf{v}_1$, i.e., $\mathbf{w}(t) = \mathbf{v}_1 + \mathbf{e}(t)$, where $\mathbf{C}\mathbf{v}_1 = \lambda_1\mathbf{v}_1, \|\mathbf{v}_1\| = 1$ and $\|\mathbf{e}(t)\| \ll 1$. Thus, $E\{\Delta\mathbf{w}(t) \mid \mathbf{w}(t)\} = E\{\Delta\mathbf{e}(t) \mid \mathbf{e}(t)\}$ and we get by (4) $E\{\Delta\mathbf{e}(t) \mid \mathbf{e}(t)\} = \eta(t)\{\mathbf{C}(\mathbf{v}_1 + \mathbf{e}(t), -[(\mathbf{v}_1^t + \mathbf{e}^t(t))\mathbf{C}(\mathbf{v}_1 + \mathbf{e}(t))](\mathbf{v}_1 + \mathbf{e}(t))\} = \eta(t)[\mathbf{C}\mathbf{e}(t) - 2\lambda_1 \cdot \mathbf{e}^t(t)\mathbf{v}_1\mathbf{v}_1 - \lambda_1\mathbf{e}(t) + O(e^2)]$, where $O(e^2)$ denotes the higher order terms of $\mathbf{e}(t)$. Ignoring the $O(e^2)$ terms, it becomes $E\{\Delta\mathbf{e}(t) \mid \mathbf{e}(t)\} \cong \eta(t)[\mathbf{C}\mathbf{e}(t) - 2\lambda_1\mathbf{e}^t(t)\mathbf{v}_1\mathbf{v}_1 - \lambda_1\mathbf{e}(t)]$. Recall that the normalized eigenvectors associated with distinct eigenvalues of symmetric matrix $\mathbf{C}$ are orthonormal. They can form a basis spanning the $\mathbf{R}^n$ space. As a result, we can represent $\mathbf{w}(t), \mathbf{e}(t), E\{\Delta\mathbf{e}(t)\}$, etc. by their components along the directions of the normalized eigenvectors of $\mathbf{C}$.

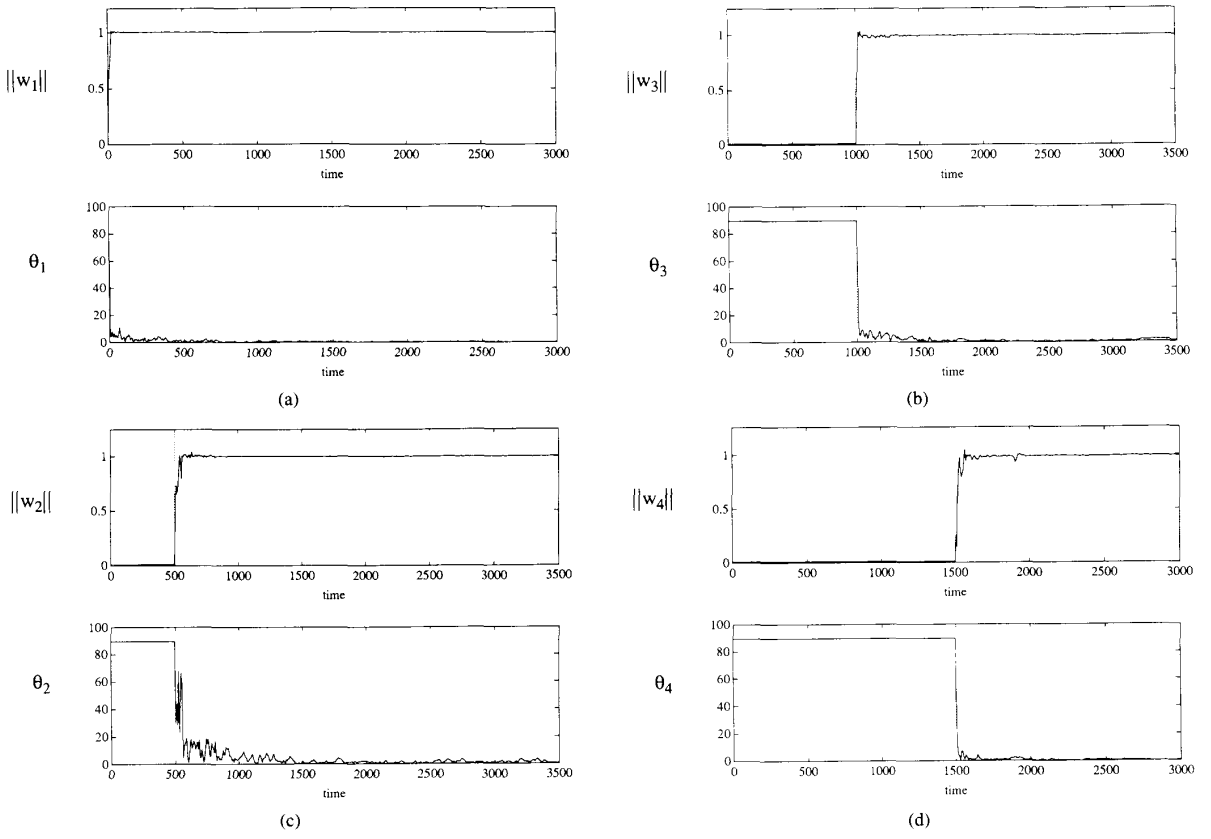

Fig. 5. Simulation results of the ALA for Sandpapers data. (a)–(d) Learning time histories of $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$, and $\mathbf{w}_4$, respectively. The vertical dotted lines denote the starting time of learning.

TABLE I
FINAL VALUES FOR SANDPAPERS USING ALA

Neuron i	θ_i	$\|\mathbf{w}_i\|$	$\hat{\lambda}_i$	λ_i
1	0.5831	1.0002	0.0266	0.0265
2	0.8085	0.9985	0.0011	0.0011
3	0.4997	0.9931	0.0009	0.0008
4	0.5086	0.9921	0.0001	0.0001

TABLE II
FINAL VALUES FOR XO8 USING ALA

Neuron i	θ_i	$\|\mathbf{w}_i\|$	$\hat{\lambda}_i$	λ_i
1	1.2992	1.0029	15.822	15.842
2	1.5005	0.9996	9.9920	10.145
3	1.1511	0.9994	6.9473	7.0847
4	1.6035	1.0001	4.8364	5.0343
5	1.7126	0.9966	3.7637	3.9537
6	1.5847	0.9909	2.2885	2.3659
7	0.6891	0.9935	1.2062	1.1999
8	0.6647	0.9884	0.6284	0.6376

Thus, along the direction of $\mathbf{v}_j, j = 1, 2, \cdots, n$, the component of $E\{\Delta\mathbf{e}(t) \mid \mathbf{e}(t)\}$ is $\mathbf{v}_j^t E\{\Delta\mathbf{e}(t) \mid \mathbf{e}(t)\} \cong -2\eta(t)\lambda_1\mathbf{v}_1^t\mathbf{e}(t)$, if $j = 1$; $-\eta(t)(\lambda_1 - \lambda_j)\mathbf{v}_j^t\mathbf{e}(t)$, if $j \neq 1$. Therefore

$$\mathbf{v}_j^t E\{\mathbf{e}(t+1) \mid e(t)\} = \mathbf{v}_j^t e(t) + \mathbf{v}_j^t E\{\Delta\mathbf{e}(t) \mid e(t)\} \cong \begin{cases} (1 - 2\eta(t)\lambda_1)\mathbf{v}_1^t e(t), & \text{if } j = 1, \\ [1 - \eta(t)(\lambda_1 - \lambda_j)]\mathbf{v}_1^t e(t), & \text{if } j \neq 1 \end{cases} \quad \text{(A1)}$$

where $e(t)$ stands for the realization of $\mathbf{e}(t)$ at time t. It can be seen that if $\eta(t) < 1/\lambda_1$, then $\mid 1 - 2\eta(t)\lambda_1 \mid < 1$ and $\mid 1 - \eta(t)(\lambda_1 - \lambda_j) \mid < 1$ for $j \neq 1$. Thus, $\mid \mathbf{v}_j^t E\{\mathbf{e}(t+1) \mid e(t)\} \mid$ will be smaller than $|\mathbf{v}_j^t e(t)|$ along all directions $\mathbf{v}_j$, $j = 1, 2, \cdots, n$. Since this is true for all realizations $e(t)$, the expectation of error will thus decrease. It implies that the expectation of $\mathbf{w}(t)$ will approach $\mathbf{v}_1$. Hence, if $\eta(t) > 1/\lambda_1$, then the expectation of $\mathbf{w}(t)$ cannot approach $\mathbf{v}_1$ and therefore $\mathbf{w}(t)$ cannot converge to $\mathbf{v}_1$. This completes the proof.

Proof of Proposition 2: When $\eta(t)\lambda_1 < 0.5$, the values of $\mid 1 - 2\eta(t)\lambda_1 \mid$ and $\mid 1 - \eta(t)(\lambda_1 - \lambda_j) \mid$, $j = 2, 3, \cdots, n$, will be closer to one if the value of $\eta(t)\lambda_1$ is closer to zero. As a result, from (A1), the expectation of error will decrease much more slowly.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their helpful comments during the revision process.

REFERENCES

[1] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[2] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, pp. 459–473, 1989.

[3] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, pp. 53–58, 1989.

[4] E. Oja, "Neural networks, principal components, and subspaces," *Int. J. Neural Systems*, vol. 1, pp. 61–68, 1989.

[5] A. L. Yuille, D. M. Kammen, and D. S. Cohen, "Quadrature and the development of orientation selective cortical cells by Hebb rules," *Biol. Cybern.*, vol. 61, pp. 183–194, 1989.

[6] P. Földiák, "Adaptive network for optimal linear feature extraction," in *Proc. Int. Joint Conf. Neural Networks*, vol. I, San Diego, 1989, pp. 401–405.

[7] J. Rubner and P. Tavan, "A self-organizing network for principal component analysis," *Europhysics Lett.*, vol. 10, pp. 693–698, 1989.

[8] R. Lenz and M. Österberg, "Computing the Karhunen–Loéve expansion with a parallel, unsupervised filter system," *Neural Computa.*, vol. 4, pp. 382–392, 1992.

[9] R. H. White, "Competitive Hebbian learning: Algorithm and demonstrations," *Neural Networks*, vol. 5, pp. 261–275, 1992.

[10] E. Oja, "A simplified neuron model as a principal component analyzer," *J. Math. Biol.*, vol. 15, pp. 267–273, 1982.

[11] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. New York: Springer-Verlag, 1978.

[12] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *J. Math. Anal. Applicat.*, vol. 106, pp. 69–84, 1985.

[13] J. Karhunen and J. Joutsensalo, "Tracking of sinusoidal frequencies by neural network learning algorithms," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Toronto, pp. 3177–3180, May 1991.

[14] C. Darken *et al.*, "Learning rate schedules for faster stochastic gradient search," in S. Y. Kung *et al.*, *Neural Networks Signal Process. II* New York: IEEE Press, pp. 3–12, 1992.

[15] S. Y. Kung, *Digital Neural Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[16] D. M. Clark and K. Ravishankar, "A convergence theorem for Grossberg learning," *Neural Networks*, vol. 3, pp. 87–92, 1990.

[17] K. Hornik and C.-M. Kuan, "Convergence analysis of local feature extraction algorithms," *Neural Networks*, vol. 5, pp. 229–240, 1992.

[18] P. Common and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, no. 8, pp. 1327–1343, Aug. 1990.