グラフィクスとCAD 54－6
(1991. 12. 20)

Homogeneous Neural Networks for Principal Component Analysis

Erkki Oja† Hidemitsu Ogawa‡ Jaroonsakdi Wangviwattana‡
† Department of Information Technology, Lappeenranta University of Technology, Finland
‡ Department of Computer Science, Tokyo Institute of Technology, Japan

あらまし — 主成分分析を行うためには，信号の相関行列の固有ベクトルを求めなければいけない．そのためのアルゴリズムとして，レーリーの原理による方法が知られている．しかし，これは非対称な基準に基づくものであり，一様ニューラルネットで実現することはできない．一方，完全に対称な基準に基づく「部分空間ネットワーク」は，一様ニューラルネットで実現できるが，固有ベクトルそのものではなく，固有ベクトルを任意に回転したものしか求めることができない．そこで本論文では，固有ベクトルそのものを求めることができる評価基準を提案し，「荷重部分空間基準」と名付ける．さらに，部分空間ネットワークと同様の学習アルゴリズムを導出し，一様なニューラルネットで実現できることを示す．

Abstract — **The Subspace Network, a type of homogeneous neural network, does not converge to the dominant eigenvectors, which provide the principal component axes, due to its complete symmetric criterion. A new criterion called the Weighted Subspace Criterion whose solutions are the true eigenvectors is derived by making a small symmetry–breaking change to the Subspace Criterion. A modified learning rule, which can still be implemented on a homogeneous network and gives the dominant eigenvectors, is given by making the corresponding change to the learning rule of the Subspace Network.**

1 Introduction

Typically, a picture or speech signal contains much redundant information which can be considerably removed. In feature extraction and data compression, the Principal Component Analysis, or PCA, is a well–known standard and general purpose technique of dimensionality reduction [8], [17].

The technique reduces the number of dimensions by discarding the linear combinations which have small variances, or by finding a linear transformation which transforms a random data vector x to a lower dimensional vector y where the sum of variances of each element of y is maximum. In other words, the lower dimensional vector y contains as much information as possible about the data vector x where the mean squared error is used as the criterion. The optimal solution to the PCA problem is given in terms of the eigenvectors of the data covariance matrix [1], [5].

Since computing the eigenvectors is very complicated, it is usually not possible to solve the PCA problem in on–line data compression applications like real–time image or speech coding, so some approximation techniques like the Discrete Cosine Transform (DCT) are used instead [25], [2]. A large number of algorithms improving the speed of computing the PCA problem have been suggested.

Recently, parallel on–line computations and neural networks have offered advantages in computing speed and hardware modeling. To solve the PCA problem, many algorithms for parallel computations and for neural networks have been researched.

Firstly, the Constrained Hebbian Learning rule was introduced as a *PCA Neuron* [16], a basic building block for feature extraction under a simple Hebbian rule with a nonlinear feedback term. It is shown that this neuron is able to find the first principal component of the input vector stream using a simple learning rule. After that, various learning algorithms and related networks have been suggested for principal components computation [6], [9], [11], [12], [14], [17], [18], [19], [20], [23], [24], [26], [27]. Especially in [17] and [19], the learning rules were associated with constrained optimization of a PCA Criterion in a rigorous way. Because of their parallelism and adaptivity to input data, this type of learning algorithms and their implementations in neural networks are potentially useful in real–time feature detection and on–line data compression tasks.

To model biological neural networks and to implement artificial neural networks in parallel, the following properties are required:

- the value of any parameter in the network is bounded.
- all processing elements in the network use the same computational algorithm, called *homogeneity.*

In all the PCA learning suggested here, the growths of values of all parameters are controlled by a feedback term, and the change of any weight of any neuron uses only local variables: a value of that weight, an output of that neuron, an input to that weight, and a feedback term from other neurons.

The networks, suggested in [19] and [24], which learn the true principal component vectors, are not homogeneous as different neurons use different algorithms.

However, for [6], [11], [18], and [20], the homogeneity property is valid, since every neuron learns with exactly the same learning algorithm. In these homogeneous learning networks, unfortunately, the true principal component vectors are not obtained, but any basis of this principal components subspace is possible. It is because of the fully symmetrical criterion which underlies the learning algorithm. In many applications of data compression, the principal component subspace seems to be insufficient [24].

A new learning algorithm, which can learn the true principal component vectors and also has the homogeneity property, has been studied here. It is shown that the homogeneous network can be used to produce the true principal component vectors when a small change is made in the learning algorithm. If each neuron has a scalar parameter of its own which is a little different from that of the other neurons, then the learning algorithm of the network breaks the complete symmetry but still can be implemented in a homogeneous network. Note that in the biological neural networks, the gain or parameter of each neuron would not be exactly the same as that of other neurons, but it is true that each neuron uses the same algorithm as the others.

Chapter 2 gives various criteria for the PCA optimization problem. The basic PCA Criterion is shown. The Asymmetrical Criterion giving the true principal component vectors, and the Subspace Criterion giving only the subspace of the principal component vectors are described. The Weighted Subspace Criterion, whose solutions are the principal component vectors, is introduced.

Based on the optimization criteria, the learning algorithms are derived in Chapter 3 by using a gradient ascent technique. Chapter 4 gives a simulation of these learning algorithms. The conclusions are given in Chapter 5.

2 Optimization Problems

2.1 Asymmetrical Criterion

Assume that x is a random zero–mean vector in $\mathcal{R}^K$, and w_1 is a weight vector in $\mathcal{R}^K$. Then $y_1 = w_1^T x$ is the first principal component of x if the variance of y_1 is maximally large under the constraint that the norm of w_1, $\|w_1\| = (w_1^T w_1)^{1/2}$, is constant. Usually this constant is taken as one. The PCA Criterion (see e.g., [1]) is

Criterion 1 *PCA Criterion*

$$\begin{aligned} \text{maximize:} \quad & J_1^{\text{PCA}}(w_1) = E_x\{y_1^2\} \\ & = E_x\{(w_1^T x)^2\} \\ & = w_1^T C w_1, \\ \text{constraint:} \quad & \|w_1\| = 1, \end{aligned}$$

where $E_x\{\cdot\}$ is the expectation over x, and the matrix C is the $K \times K$ covariance matrix defined by $C = E_x\{xx^T\}$.

It is well–known that the solution is given in terms of the eigenvectors $c_1, \ldots, c_K$ of matrix C [1]. Assume that the corresponding eigenvalues, $\lambda_1, \ldots, \lambda_K$, is in decreasing order,

$$\lambda_1 > \lambda_2 > \cdots > \lambda_K > 0. \tag{1}$$

Since λ_1 is larger than the others, then there are only two solutions,

$$w_1 = \pm c_1.$$

It simplifies the consequent analysis if we assume from now on that the condition in (1)

holds for the eigenvalues of covariance matrix C. All the consequent analysis could be carried out without this assumption but in some cases the results would become more complicated. Usually in a real–world situation, e.g., in speech or image processing, the eigenvalues of covariance matrices will be all different and strictly positive.

Criterion 1 can be generalized to N principal components, with $1 \leq N \leq K$. Denoting the n^{th} principal component vector by w_n and the n^{th} principal component of x by $y_n = w_n^T x$ where $1 \leq n \leq N$, a possible extension is

Criterion 2 *Asymmetrical Criterion*

$$\begin{aligned} \text{maximize:} \quad & J_1^{\text{PCA}}(w_n) = E_x\{y_n^2\} \\ & = w_n^T C w_n, \\ \text{constraint:} \quad & w_m^T w_n = \delta_{mn}, \ m \leq n. \end{aligned}$$

With this criterion, the first principal component vector w_1 is defined independently of the others, the second vector w_2 is orthogonal to w_1 but independent of the others, and so on. The criterion is not symmetrical with respect to w_m and w_n, $m \neq n$. Therefore, this is called the *Asymmetrical Criterion.* There are only two unique solutions

$$w_n = \pm c_n$$

for each $n \leq N$ under the assumption in (1). Note that the terms $y_n = w_n^T x$ become uncorrelated.

2.2 Subspace Criterion

The fully *symmetrical criterion* was considered in [8], [15], as an extension of Criterion 1:

Criterion 3 *Subspace Criterion*

$$\begin{aligned} \text{maximize:} \quad & J_N^{\text{PCA}}(w_1, \ldots, w_N) \\ & = E_x\{\sum_{n=1}^{N} y_n^2\} \\ & = \sum_{n=1}^{N} w_n^T C w_n, \\ \text{constraint:} \quad & w_m^T w_n = \delta_{mn}. \end{aligned}$$

This criterion can be written in terms of a matrix $W = (w_1 \cdots w_N)$ whose columns are the weight vectors w_n:

$$\begin{aligned} \text{maximize:} \quad & J_N^{\text{PCA}}(w_1, \ldots, w_N) \\ & = \text{trace}(W^T C W), \qquad (2) \\ \text{constraint:} \quad & W^T W = I. \qquad (3) \end{aligned}$$

Eq. (3) implies that the matrix $P = WW^T$, is an orthogonal projection matrix satisfying $P^2 = P$, $P^T = P$. Due to these properties, (2) can be written as

$$\begin{aligned} J_N^{\text{PCA}}(w_1, \ldots, w_N) &= \text{trace}(W^T C W) \\ &= \text{trace}(CWW^T) = \text{trace}(CP) \\ &= \text{trace}(PCP) = \text{trace}(E_x\{Pxx^T P^T\}) \\ &= E_x\{\|Px\|^2\}. \qquad (4) \end{aligned}$$

Thus the problem is completely equivalent to the problem of finding the N–dimensional subspace of $\mathcal{R}^K$ such that the squared norm of the projection of x onto the subspace is maximally large on the average. Therefore, Criterion 3 will be called the *Subspace Criterion.*

This problem was solved in [15]. The solution was shown to be the subspace $\mathcal{L}(c_1, \ldots, c_N)$ spanned by the N eigenvectors $c_1, \ldots, c_N$ of the matrix C corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_N$. Thus the solution of this criterion is given by

$$\mathcal{L}(w_1, \ldots, w_N) = \mathcal{L}(c_1, \ldots, c_N).$$

This means that a set of $w_1, \ldots, w_N$ can be *any* basis of the subspace $\mathcal{L}(c_1, \ldots, c_N)$. Thus it does not follow that the terms $y_n = w_n^T x$ will be uncorrelated.

2.3 Weighted Subspace Criterion

Usually the numbers y_n are coded e.g. by the Huffman code or arithmetic code before being stored or transmitted [25], [2]. In this case, compression is optimal if the numbers y_n have as unequal variances as possible. Also, sometimes the higher–order principal components are very small and can be omitted altogether.

Therefore, it is important in practice to find not only the principal eigenvector subspace but also the principal components themselves. It is the purpose to show in the following how a small change in the Subspace Criterion will reduce the

ambiguous solution to a unique set of eigenvectors.

From Criterion 3, we make a change in the constraint to change the problem to:

Criterion 4 *Weighted Subspace Criterion*
Let ω_n be any fixed real numbers such that

$$\omega_1 > \omega_2 > \cdots > \omega_N > 0. \tag{5}$$

$$\text{maximize:} \quad J_N^{PCA}(w_1, \ldots, w_N) = \sum_{n=1}^{N} w_n^T C w_n, \tag{6}$$

$$\text{constraint:} \quad w_m^T w_n = \omega_n \delta_{mn}, \tag{7}$$

In terms of matrix $W = (w_1 \cdots w_N)$, (7) becomes

$$W^T W = \Omega = \text{diag}(\omega_1 \cdots \omega_N).$$

Because of the weighting parameters ω_n in (7), this will be called the *Weighted Subspace Criterion*. This criterion is not symmetrical with respect to w_m and w_n, $m \neq n$.

It turns out that when these ω_n are not equal, even if they are arbitrarily close to one, the problem has a *unique* solution (except for the sign). This is shown in the following.

Theorem 1 *Assume that the eigenvalues of C satisfy the condition in (1). Then the Criterion 4 is satisfied if and only if*

$$w_n = \pm\sqrt{\omega_n}\, c_n, \tag{8}$$

where c_n is an eigenvector of C corresponding to the eigenvalue λ_n.

One can check easily that when (8) holds, J_N^{PCA} has the maximum value,

$$\max \; J_N^{\text{PCA}} = \sum_{n=1}^{N} \omega_n c_n^T C c_n = \sum_{n=1}^{N} \omega_n \lambda_n.$$

The complete proof is given in [21].

3 Learning Algorithms

Theorem 1 gives the closed–form solutions for the Criterion 4, which can be used to get numerical values in an application. However, to use the closed–form solutions directly, the covariance matrix C must be known. This is rarely true in practice. Usually it must be estimated from samples, and then an infinite number of samples are needed to get a zero–error estimate.

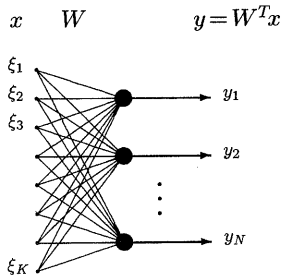

Figure 1: *The linear neural network.*

Another alternative, emphasized earlier in [19], is to use the constrained optimization problems suggested above to derive corresponding constrained gradient ascent algorithms. Using this technique for each suggested criteria, the corresponding algorithms can be derived. The algorithms can be implemented on a neural network shown in Figure 1. The weights of the network will then converge to the solutions of the problems.

3.1 Asymmetrical PCA Learning

In [19] and [24], the learning algorithms are obtained from Criterion 2 based on the idea of the numerical technique for computing several eigenvectors of a matrix. The first principal component vector, or the first eigenvector of the covariance matrix of input data is learned by the first neuron independently from other neurons, while the second principal component vector is learned by the second neuron using a feedback from the first neuron, etc. According to that, all the true principal component coefficient vectors are learned by these learning networks.

Other algorithms based on Criterion 2 have been proposed in [6], [9], [23], and [24]. Similar units were employed in the Perceptual Network of [14], and it was shown that they can learn efficient features from the input space. It was also pointed out that the PCA criterion is optimal in terms of information loss.

3.2 Subspace Learning

One of the learning rule for matrix W which is obtained from Criterion 3, was presented in [18], based on the mathematical analysis in [19] by using the following constrained gradient ascent technique. First, the gradient of $\Sigma_{m=1}^{N}(w_m^T x)^2$ with respect to w_n is $2(w_n^T x)x$, then an unconstrained gradient ascent algorithm based on samples $x^{(i)}$ would be

$$w_n^{(i+1)} = w_n^{(i)} + \gamma_i x^{(i)} x^{(i)T} w_n^{(i)}, \tag{9}$$

where i denotes the discrete time, $i = 0, 1, 2, \ldots$, and γ_i is a positive scalar step size or gain which usually depends on the time step i. In matrix form, with $W = (w_1 ... w_N)$, (9) is equivalent to

$$W^{(i+1)} = W^{(i)} + \gamma_i x^{(i)} x^{(i)T} W^{(i)}.$$

Next, in order to have orthonormal column vectors $w_n^{(i+1)}$ for $W^{(i+1)}$, orthonormalization after each step has to be done:

$$\tilde{W}^{(i+1)} = W^{(i)} + \gamma_i x^{(i)} x^{(i)T} W^{(i)}, \tag{10}$$

$$W^{(i+1)} = \tilde{W}^{(i+1)} (\tilde{W}^{(i+1)T} \tilde{W}^{(i+1)})^{-\frac{1}{2}}. \tag{11}$$

This clearly implies

$$W^{(i+1)T} W^{(i+1)} = I,$$

or

$$w_m^{(i+1)T} w_n^{(i+1)} = \delta_{mn}.$$

If (10) and (11) are expanded as a power series of the parameter γ_i, and second order terms are omitted, the following *Subspace Learning Algorithm (SLA)* is obtained:

Algorithm 1 *SLA*

$$\begin{aligned} w_n^{(i+1)} &= w_n^{(i)} + \gamma_i (y_n^{(i)} x^{(i)} - y_n^{(i)} f^{(i)}), \\ y_n^{(i)} &= w_n^{(i)T} x^{(i)}, \\ f^{(i)} &= \sum_{n=1}^{N} y_n^{(i)} w_n^{(i)}, \\ n &= 1, \ldots, N. \end{aligned}$$

It has been shown in stochastic approximation theory (cf. e.g. [13]) that when γ_i is small, this algorithm will converge to the solution of the Criterion 3. The algorithm was analyzed in more detail in [22].

Compared to the conventional way of estimating the data covariance matrix and computing its eigenvectors and eigenvalues, this type of algorithm has some advantages: it needs no storage for the covariance matrix because the basis vectors are computed directly from input data, and it can also be used in cases when the input is nonstationary, to track slow changes in statistics. It also has some relevance to models of feature extraction in biological neural networks. Especially, this algorithm is suitable for implementations on massively parallel networks. The network implementation was considered in [18] where it was called the *Subspace Network*. An analysis of the same learning rule was earlier presented in [26].

It was shown in [4] and [7] that using the 3–layer auto–associative MLP net and the standard Back Propagation algorithm, the hidden layer will learn the principal component subspace of inputs, (see also [3], [10]).

3.3 Weighted Subspace Learning

In analogy with the derivation of Algorithm 1, the following constraint gradient ascent algorithm is firstly obtained from Criterion 4:

$$\tilde{W}^{(i+1)} = W^{(i)} + \gamma_i x^{(i)} x^{(i)T} W^{(i)}, \tag{12}$$

$$W^{(i+1)} = \tilde{W}^{(i+1)} (\tilde{W}^{(i+1)T} \tilde{W}^{(i+1)})^{-\frac{1}{2}} \Omega^{\frac{1}{2}}, \tag{13}$$

corresponding to (10) and (11), respectively. These imply

$$W^{(i+1)T} W^{(i+1)} = \Omega = \mathrm{diag}(\omega_1 \cdots \omega_N).$$

Assume that γ is small, then we obtain (omitting the step index i)

$$W^{(i+1)} = W + \gamma_i (x x^T W - W W^T x x^T W \Omega^{-1}).$$

Denoting

$$D = \mathrm{diag}(\theta_1 \cdots \theta_N) = \Omega^{-1},$$

or

$$\theta_n = \frac{1}{\omega_n}, \; n = 1, \ldots, N, \tag{14}$$

the following *Weighted Subspace Learning Algorithm* is then obtained:

Algorithm 2 *Weighted SLA*

$$\begin{aligned} w_n^{(i+1)} &= w_n^{(i)} + \gamma_i(y_n^{(i)}x^{(i)} - \theta_n y_n^{(i)} f^{(i)}), \\ y_n^{(i)} &= w_n^{(i)T} x^{(i)}, \\ f^{(i)} &= \sum_{n=1}^{N} y_n^{(i)} w_n^{(i)}, \\ n &= 1, \ldots, N. \end{aligned}$$

Comparing to Algorithm 1, the difference is the set of parameters θ_n. They are scalar parameters that can be arbitrarily close to one.

This learning rule can be easily implemented in the Subspace Network in [18]. The only difference is that the parameter θ_n must be available at each neuron n. Note especially that the effect of parameter θ_n is local only to unit n itself; the feedback signal $f^{(i)}$ remains the same as in the original version. Therefore, also this algorithm fulfills the requirements for homogeneity.

By Theorem 1, if the θ_n are positive and all different, then the true eigenvectors can be obtained. This happens even when the deviation of θ_n is arbitrarily small.

4 A Simulation

The convergences of two types of learning algorithms: the SLA (Algorithm 1) [18] and the new algorithm, the Weighted SLA (Algorithm 2), has been studied by a small–scale numerical simulation. In Algorithm 2, the $\theta_n = \omega_n^{-1}$ had values

$$\theta_1 = 0.9,\ \theta_2 = 1.0,\ \theta_3 = 1.1.$$

The initial values for the weights W were random numbers, and γ_i was the following decreasing sequence:

$$\gamma_i = \frac{1}{2+j} \ ; \ j = \mathrm{trunc}(\frac{i}{200}).$$

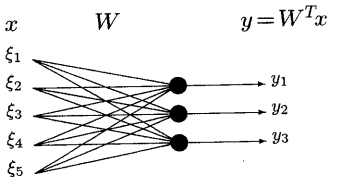

Figure 2: *The five–input three–output linear network used in the simulation*

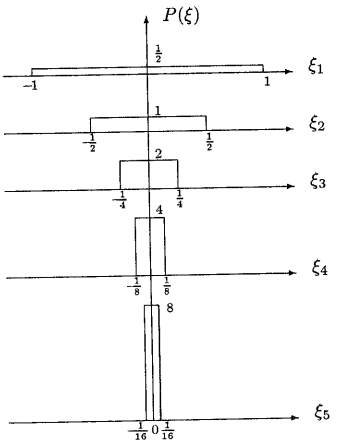

Figure 3: *The distributions of uncorrelated elements of a five–dimensional input vector x used in the simulation*

A sequence of zero–mean random input vectors $x = [\xi_1 \cdots \xi_5]^T$ was applied to the five–input three–output linear network shown in Figure 2. The distributions of uncorrelated elements of an input vector x are shown in Figure 3.

Since the distributions of input elements are known exactly, the results can be computed theoretically. Because all elements of an input vector x are uncorrelated, $E\{\xi_i \xi_j\} = 0$, $i \neq j$, and the correlation matrix can be written as

$$\begin{aligned} &E_x\{xx^T\} \\ &= \begin{bmatrix} E\{\xi_1\xi_1\} & \cdots & E\{\xi_1\xi_5\} \\ \vdots & & \vdots \\ E\{\xi_5\xi_1\} & \cdots & E\{\xi_5\xi_5\} \end{bmatrix} = \begin{bmatrix} E\{\xi_1^2\} & & 0 \\ & \ddots & \\ 0 & & E\{\xi_5^2\} \end{bmatrix}. \end{aligned}$$

Because the distribution of each element of an input vector x is uniform, this expectation value can be computed easily:

$$\begin{aligned} E\{\xi_n^2\} &= \int_{-\infty}^{\infty} \xi_n^2 P(\xi_n) d\xi_n \\ &= \int_{-2^{1-n}}^{2^{1-n}} \xi_n^2 (2^{n-2}) d\xi_n = \frac{1}{3 \cdot 2^{2n-2}}. \end{aligned}$$

Then the correlation matrix can be written as

$$E_x\{xx^T\} = \begin{bmatrix} \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{12} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{48} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{192} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{768} \end{bmatrix}.$$

Table 1:
WEIGHTED VECTORS AND THEIR INNER PRODUCTS AFTER 40,000 TRAINING STEPS

Algorithm	$W = (w_1 w_2 w_3)$	$W^T W$
Algorithm 1 *(SLA)*	$\begin{bmatrix} 0.945 & 0.326 & 0.038 \\ -0.165 & 0.371 & 0.914 \\ 0.284 & -0.870 & 0.404 \\ -0.001 & 0.001 & 0.001 \\ 0.000 & 0.000 & -0.001 \end{bmatrix}$	$\begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$
Algorithm 2 *(Weighted SLA)*	$\begin{bmatrix} \mathbf{1.054} & -0.002 & -0.002 \\ 0.002 & \mathbf{1.000} & 0.001 \\ 0.003 & -0.001 & \mathbf{0.954} \\ -0.001 & 0.002 & -0.001 \\ 0.000 & -0.001 & 0.000 \end{bmatrix}$	$\begin{bmatrix} 1.111 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.909 \end{bmatrix}$

Thus, the eigenvectors are $c_1 = (1\ 0\ 0\ 0\ 0)^T$, $c_2 = (0\ 1\ 0\ 0\ 0)^T$, and so on, and the eigenvalues are $\lambda_1 = \frac{1}{3}$, $\lambda_2 = \frac{1}{12}$, and so on. These satisfy the condition of the decreasing order in (1).

Table 1 shows the weight matrix W and the inner products between the weight vectors w_n, after 40,000 training steps.

In both cases, the weight vectors became orthogonal and spanned the subspace of the three dominant eigenvectors, $\mathcal{L}(c_1, c_2, c_3)$. However, only Algorithm 2 produced the directions of the eigenvectors themselves. The weight vector of the first neuron, with the smallest θ, tends to the first eigenvector multiplied by a scalar number, and so on. The inner products $w_n^T w_n = \|w_n\|^2$,

$$\|w_1\|^2 = \omega_1 = \theta_1^{-1} = 0.9^{-1} = 1.1111,$$
$$\|w_2\|^2 = \omega_2 = \theta_2^{-1} = 1.0^{-1} = 1.0000,$$
$$\|w_3\|^2 = \omega_3 = \theta_3^{-1} = 1.1^{-1} = 0.9091,$$

are in good agreement with (14) and Theorem 1.

5 Conclusions

A new criterion, called the Weighted Subspace Criterion (Criterion 4), has been proposed to give the true eigenvector basis as the unique solution. A new corresponding algorithm, called the Weighted Subspace Learning Algorithm, (Algorithm 2), was derived from the criterion by using the constrained gradient ascent technique. This new algorithm can still be implemented in the homogeneous neural network like the Subspace Network. The simulation showed that the convergence of the algorithm is in good agreement with theory. Table 2 shows that this algorithm has both of required properties comparing to other algorithms which derived from previously known criteria.

Many of the results, like convergence proofs, the Non–linear Weight Subspace Learning, and its parallel neural networks implementation, are included in [21] and [22].

Table 2:
COMPARISON OF THREE DIFFERENT STATISTICAL OPTIMIZATION CRITERIA

Criterion	Algorithm	Homogeneity holds	Solutions are principal component vectors
Criterion 2 *(Asymmetrical)*	e.g. [17],[24]	No	Yes
Criterion 3 *(Subspace)*	e.g. Algorithm 1 [18]	Yes	No
Criterion 4 *(Weighted Subspace)*	Algorithm 2	Yes	Yes

Acknowledgments

This work was undertaken while the first author (E.O) held the Toshiba Endowed Chair at Tokyo Institute of Technology during 1990–1991. He is grateful to Toshiba Co. and T.I.T. for this opportunity. This work was partially supported by the Grant-in-Aid for Scientific Research in Priority Areas #03244101, and the Grant-in-Aid for Scientific Research #02452155.

References

[1] T. W. Anderson, "An Introduction to Multivariate Statistical Analysis", John Wiley, New York, 1958.

[2] P. Ang, P. Ruetz, D. Auld, "Video compression makes big gains", *IEEE Spectrum,* vol. 28, Oct. 1991.

[3] P. Baldi, K. Hornik, "Neural Networks and Principal Components Analysis: learning from examples without local minima", *Neural Networks,* vol. 2, pp. 52–58, Jan. 1989.

[4] H. Bourlard, Y. Kamp, "Auto-association by Multilayer Perceptrons and Singular Value Decomposition", *Biol. Cybern.,* vol. 59, pp. 291–294, Sep. 1988.

[5] N. C. Giri, "Multivariate Statistical Inference", *Probability and Mathematical Statistics: A Series of Monographs and Textbooks,* Academic Press, London, 1977.

[6] Y. Chauvin, "Principal Component Analysis by Gradient Descent on a Constrained Linear Hebbian Cell", *Proc. IJCNN–89,* Washington DC, Jun. 1989.

[7] G. W. Cottrell, P. W. Munro, D. Zipser, "Image Compression by Back-propagation: a demonstration of extensional programming", ICS Report 8702, UCSD, Institute of Cognitive Science, Feb. 1987.

[8] P. A. Devijver, J. Kittler, "Pattern Recognition: a statistical approach", Prentice-Hall, London, 1982.

[9] P. Földiàk, "Adaptive Network for Optimal Linear Feature Extraction", *Proc. IJCNN–89,* Washington DC, pp. 401–405, Jun. 1989.

[10] B. Irie, M. Kawato, "Acquisition of Internal Representation by Multi–Layered Perceptrons", *Tran. IEICE,* vol. J73–D–II, pp. 1173–1178, Aug. 1990.

[11] A. Krogh, J. Hertz, "Hebbian Learning of Principal Components", *NORDITA Preprint,* No. 89/50S, Denmark, 1990.

[12] S. Kung, K. Diamantras, "A Neural Network Learning Algorithm for Adaptive Principal Component Extraction (APEX)", *Proc. ICASSP–90,* Albuquerque NM, pp. 861–864, Apr. 1990.

[13] H. Kushner, D. Clark, "Stochastic Approximation Methods for Constrained and Unconstrained Systems", Springer–Verlag, New York, 1978.

[14] R. Linsker, "Self–organization in a Perceptual Network", *Computer,* vol. 21, Mar. 1988.

[15] H. Ogawa, "On the subspace which provides the best approximation to a set of patterns", *Paper of Technical Group PRU90–67,* pp. 67–72, IEICE, Japan, Oct. 1990 (in Japanese).

[16] E. Oja, "A Simplified Neuron Model as a Principal Component Analyzer", *J. Math. Biol.,* vol. 15, pp. 267–273, Nov. 1982.

[17] E. Oja, *Subspace Methods of Pattern Recognition,* Research Studies Press and John Wiley, Letchworth, 1983.

[18] E. Oja, "Neural Networks, Principal Components, and Subspace", *Int. J. Neural Systems,* vol. 1, pp. 61–68, Apr. 1989.

[19] E. Oja, J. Karhunen, "On Stochastic Approximation of the Eigenvectors and Eigenvalues of a Random Matrix", *J. Math. Anal. & Appl.,* vol. 106, pp. 69–84, Jan. 1985.

[20] E. Oja, H. Ogawa, J. Wangviwattana, "Learning in Nonlinear Constrained Hebbian Networks", in T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas (Eds.), *Artificial Neural Networks,* pp. 385–390, Elsevier Science Publishers, Amsterdam, Jun. 1991.

[21] E. Oja, H. Ogawa, J. Wangviwattana, "Principal Component Analysis by Homogeneous Neural Networks, Part I: the Weighted Subspace Criterion", Submitted.

[22] E. Oja, H. Ogawa, J. Wangviwattana, "Principal Component Analysis by Homogeneous Neural Networks, Part II: Analysis and Extensions of the Learning Algorithms", Submitted.

[23] J. Rubner, P. Tavan, "A Self–organizing Network for Principal Components Analysis", *Europhysics Letter,* vol. 10, pp. 693–689, Dec. 1989.

[24] T. D. Sanger, "Optimal Unsupervised Learning in a Single–layer Linear Feedforward Network", *Neural Networks,* vol. 2, pp. 459–473, Dec. 1989.

[25] G. K. Wallace, "The JPEG Still Picture Compression Standard", *Commun. ACM,* No. 34, pp. 30–44, Apr. 1991.

[26] R. Williams, "Feature Discovery through Error–correcting Learning", TR–8501 UCSD, Institute of Cognitive Science, May 1985.

[27] L. Xu, "Least MSE Reconstruction: a Principle for Self–organizing Neural Nets", Concordia University, Department of Computer Science Report TR–X9152, Jun. 1991.