The Nonlinear PCA Learning Rule and Signal Separation - Mathematical Analysis

Erkki Oja

Helsinki University of Technology
Laboratory of Computer and Information Science
Rakentajanaukio 2 C, FIN-02150 Espoo, Finland
Fax: +358-0-4513277
Email: erkki.oja@hut.fi

Report A26 (August 1995)

ISBN 951-22-2706-1
ISSN 0783-7445

TKK OFFSET

Abstract

It has been verified experimentally that nonlinear versions of the PCA network learning rules for the weights of a neural layer produce neurons that have signal separation capabilities. One of the learning rules earlier proposed by the author is studied here mathematically to analyze why and how the algorithm works in this application. It is shown that for input vectors whose density is symmetrical around the origin and has equal variances for each element, the weight matrix obtained as the asymptotic solution of the nonlinear PCA learning rule is in some cases a rotation of the input vector to statistically independent directions. This explains why it can be used for image and speech signal separation. Sufficient conditions are formulated, depending on the nonlinear neuron activation function and on the probability densities of the original signal components. It is shown that a sigmoidal nonlinearity as the activation function is feasible for flat sub-Gaussian densities of the original signals, while polynomial activation functions are feasible for sharp super-Gaussian densities. An example with uniform densities and a hyperbolic tangent activation function is analyzed in depth both by mathematical analysis and by simulations on the learning rule.

Acknowledgement. The author wishes to thank Dr. Juha Karhunen and Mr. Aapo Hyvrinen for helpful comments on the preliminary version of the manuscript.

1 Introduction

The PCA neuron [12] and the PCA network [15, 17] were introduced by the author with the purpose of demonstrating how the well-known Principal Component Analysis expansion can be realized in a simple parallel layer of adaptive elements. These artificial neurons were linear because PCA is an inherently linear technique. For one linear neuron, the basic *Constrained Hebbian learning rule* is:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu_k[\mathbf{x}_k - \mathbf{w}_k y_k]y_k \tag{1}$$

where $\mathbf{x}_k, \; k = 1, 2, ...$ is a sequence of samples from the random n-dimensional input vector $\mathbf{x}$, with k denoting discrete time, $\mathbf{w}_k$ is the weight vector of the neuron with output $y_k = \mathbf{w}_k^T\mathbf{x}_k$, and μ_k is a learning rate sequence. It can be shown that under conditions specified in [14], $\mathbf{w}_k$ tends to the dominant eigenvector of the input correlation matrix.

The extension of (1) to a set of m parallel neural units (the Subspace rule, [15]) is:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{x}_k - \mathbf{W}_k\mathbf{y}_k]\mathbf{y}_k^T \tag{2}$$

where now $\mathbf{W}_k = [\mathbf{w}_k(1) \; \mathbf{w}_k(2) \; ... \; \mathbf{w}_k(m)]$ is the weight matrix, whose columns are the individual neuron weight vectors $\mathbf{w}_k(i)$, and $\mathbf{y}_k = \mathbf{W}_k^T\mathbf{x}_k$ is the output vector of m elements. This algorithm and some extensions were analyzed in [17]. Now the weight vectors $\mathbf{w}_k(i)$ become orthonormal and tend to a basis of the m-dimensional dominant eigenvector subspace of the input correlation matrix; however, usually the individual weight vectors do not tend to the eigenvectors.

It was obvious from the start that exactly the same learning rules and implementations could be extended in a straightforward way to *nonlinear neurons*; some such extensions were proposed in [16]. Quite formally, without any reference to an optimization problem, we might introduce the *Nonlinear PCA learning rules* simply by replacing the linear neurons whose outputs are $y_k = \mathbf{w}_k^T\mathbf{x}_k$, by nonlinear neurons with outputs $g(y_k) = g(\mathbf{w}_k^T\mathbf{x}_k)$, with $g(y)$ the activation function of the neurons. Typically, $g(.)$ might be a sigmoidal function. Then the one-unit rule becomes

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu_k[\mathbf{x}_k - \mathbf{w}_k g(y_k)]g(y_k), \tag{3}$$

and the multi-unit rule becomes

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{x}_k - \mathbf{W}_k g(\mathbf{y}_k)]g(\mathbf{y}_k^T) \tag{4}$$

with $\mathbf{y}_k$ the vector

$$\mathbf{y}_k = \mathbf{W}_k^T\mathbf{x}_k. \tag{5}$$

In eq. (4), the output vector $g(\mathbf{y}_k)$ must be defined element by element, i.e.,

$$g(\mathbf{y}_k) = [g(\mathbf{w}_k(1)^T\mathbf{x}_k)...g(\mathbf{w}_k(m)^T\mathbf{x}_k)]^T. \tag{6}$$

Figure 1: The nonlinear PCA layer.

On the right hand side, these nonlinear PCA learning rules have a *Hebbian term* (input times output), and another normalizing term that will prevent excessive growth of the weights. This rule can be implemented in a one-layer Nonlinear PCA network shown in Fig. 1.

In [16], these rules were not motivated from optimization criteria, nor were they applied to realistic signal processing problems. In two recent articles [6, 7], it was shown by Karhunen and Joutsensalo that the rule (4) is an approximate stochastic gradient algorithm for minimizing the mean square representation error

$$J(\mathbf{W}) = E\{\|\mathbf{x} - \mathbf{W}g(\mathbf{W}^T\mathbf{x})\|^2\} = E\{\|\mathbf{x} - \sum_{j=1}^{m} g(\mathbf{w}(j)^T\mathbf{x})\mathbf{w}(j)\|^2\}. \tag{7}$$

The result given in [6] is that the stochastic gradient algorithm for the cost function (7) is

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{x}_k\mathbf{e}_k^T\mathbf{W}_k\mathbf{F}(\mathbf{x}_k^T\mathbf{W}_k) + \mathbf{e}_k g(\mathbf{x}_k^T\mathbf{W}_k)] \tag{8}$$

with

$$\mathbf{e}_k = \mathbf{x}_k - \mathbf{W}_k g(\mathbf{W}_k^T\mathbf{x}_k) \tag{9}$$

and

$$\mathbf{F}(\mathbf{x}_k^T\mathbf{W}_k) = diag[g'(\mathbf{x}_k^T\mathbf{w}_k(1)), ..., g'(\mathbf{x}_k^T\mathbf{w}_k(m))]. \tag{10}$$

There $g'(y)$ denotes the derivative of the activation function $g(y)$. As motivated in more detail in [6], writing the update rule (8) for an individual weight vector $\mathbf{w}_k(i)$ shows that the first term in the brackets on the right hand side of algorithm (8) affects the update of weight vectors much less than the second term, if the error $\mathbf{e}_k$ is relatively small in norm compared to the training vectors $\mathbf{x}_k$. If the first term is omitted, then the nonlinear PCA rule (4) is obtained with the substitution $\mathbf{y}_k = \mathbf{W}_k^T \mathbf{x}_k$.

It turns out that the connection between the criterion (7) and the nonlinear PCA learning rule has quite important implications in the problems of *Exploratory Projection Pursuit, Independent Component Analysis*, and *Source separation.* For review articles on these techniques, see [2], [4], [1]. Neural learning algorithms have been recently presented by a number of authors, see [3], [5], [10]. The full implications of the Nonlinear PCA learning algorithm, together with other related learning rules, in these practical signal processing problems will be shown elsewhere [18, 9]. We only give here an example.

Example 1 (from [18]). We construct here the input vector $\mathbf{x}$ as a combination of three digital images, shown in the uppermost row in Fig. 2. The images are first *sampled row by row*; let us denote the pixels $s_k(1), s_k(2), s_k(3)$ where the index in brackets refers to image 1 (flower), 2 (model), or 3 (waterfall), and the subscript k denotes the scanning order from top left to bottom right. Then three *linear mixtures*, with different coefficients, are constructed:

$$x_k(1) = a_{11}s_k(1) + a_{12}s_k(2) + a_{13}s_k(3), \tag{11}$$

$$x_k(2) = a_{21}s_k(1) + a_{22}s_k(2) + a_{23}s_k(3), \tag{12}$$

$$x_k(3) = a_{31}s_k(1) + a_{32}s_k(2) + a_{33}s_k(3), \tag{13}$$

to produce a set of input vectors $\mathbf{x}_k = [x_k(1)\ x_k(2)\ x_k(3)]^T$. The mixture coefficients a_{ij} can be arbitrary, e.g. randomly chosen, except that the mixtures have to be linearly independent. Each of the three mixtures $x_k(i)$ can be represented as a square array by reversing the scanning process, and the result is shown on the second row in Fig. 2. It is not possible to separate the original signals from the mixtures visually. The problem of signal separation is now to form the inverse mapping, to get back the signals $s_k(i)$ from the mixtures, *without knowing the original signals or the mixture coefficients.*

To achieve this in a Nonlinear PCA network, yet another processing step is needed: it is best to *whiten* or *sphere* the inputs before using the algorithm (4). This means that all the vectors $\mathbf{x}_k$ are transformed in such a way that the elements become uncorrelated and the variances equal to one. The elements of the whitened vectors $\mathbf{v}_k$ are shown as square arrays on the third row of Fig. 2.

When now vectors $\mathbf{v}_k$ are used as the training inputs to the algorithm (4) and it is run for a long time with a small learning rate μ_k, with $g(y)$ a sigmoidal function, the weight matrix becomes such that the elements of the output vector, $\mathbf{y}_k = \mathbf{W}_k^T \mathbf{v}_k$, become very close to the original signals $s_k(i)$. The outputs are shown on the fourth row of Fig. 2. Full details of the implementation are given in [18].

The purpose of this paper is to analyze why this happens in the learning rule

Figure 2: Separation of images from mixtures using the Nonlinear PCA algorithm. 1st row: Original images, 2nd row: linear mixtures, 3rd row: whitened mixtures, 4th row: outputs from the nonlinear PCA after convergence.

(4), especially to give an asymptotic analysis of the behavior over a long training sequence. We shall concentrate on the signal separation and independent component analysis performance of the algorithm, and therefore the following assumption will be made throughout:

Assumption 1. The input sequence to the learning algorithm $\mathbf{x}_k, k = 1, 2, ...$ is an i.i.d. sample from a random vector $\mathbf{x}$ that is zero mean and has unit covariance: $E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{I}$. Furthermore, there exists an orthogonal matrix $\mathbf{B}$ such that the vector $\mathbf{u} = \mathbf{B}^T\mathbf{x}$ (also having $E\{\mathbf{u}\} = 0,\ E\{\mathbf{u}\mathbf{u}^T\} = \mathbf{I}$), has *statistically independent elements.*

The independence of the elements of $\mathbf{u}$ means that, denoting $\mathbf{u} = [u(1)...u(n)]^T$, the joint density is separable:

$$p_{\mathbf{u}}(t_1, ..., t_n) = \Pi_{j=1}^{n} p_{u(j)}(t_j), \tag{14}$$

where $p_{u(j)}(t_j)$ is the marginal density of $u(j)$. Note that the requirement of zero mean and unit covariance for the input vectors $\mathbf{x}_k$ is not a severe limitation, as *any* vector set can be transformed to satisfy these conditions by subtracting the mean and rotating by a scaled version of the eigenvector matrix. However, the assumption that $\mathbf{x}$ can be rotated to another vector having independent elements is a strong one; it means that there is an underlying model, by which $\mathbf{x}$ is formed as a mixture of independent signals, as in the above example.

The main result to be shown is Theorem 1, stating that under certain conditions the weight matrix sequence $\mathbf{W}_k$ of the algorithm (4) will in fact tend to a separating matrix, making the outputs of the algorithm (elements of the vector $\mathbf{y}_k = \mathbf{W}_k^T\mathbf{x}_k$) asymptotically independent and equal to the elements of the (unknown) vector $\mathbf{u}$, or some possibly scaled permutations of them. Theorem 1 is illustrated by some examples and compared to simulation runs using an averaged version of the learning algorithm.

Before going into Theorem 1, the next Section 2 motivates informally why the learning algorithm minimizing the cost function $J(\mathbf{W})$ of eq. (7) will produce a rotation into independent components in some cases. Section 3 introduces a mathematically convenient linear transformation into the algorithm and presents Theorem 1, whose proof is postponed into the Appendix. Another result, Theorem 2 is also given that holds for the asymptotic solutions of the one-unit learning rule. Some conclusions are presented in Section 4.

2 Optimal nonlinear representation of separable densities: an example

Consider the criterion (7). If the function $g(y)$ were linear, as in the standard PCA technique, and the number m of terms in the sum were equal to n, the dimension

Figure 3: The uniform density. Original coordinates: $x(1), x(2)$. Rotated independent coordinates: $y(1), y(2)$. Vectors $\mathbf{w}(1), \mathbf{w}(2)$ are the ones minimizing the cost function $J(\mathbf{W})$.

of $\mathbf{x}$, then the representation error would be trivially zero, if the weight vectors are chosen orthonormal. For a nonlinear function $g(y)$, however, this is usually not true. Instead, in some cases at least, it turns out that the optimal basis vectors $\mathbf{w}(i)$ minimizing (7) will be aligned with the *independent components* of the input vectors. Consider the following simple example:

Example 2. Assume that $\mathbf{x}$ is 2-dimensional and has a uniform density in a unit square that is *not* aligned with the coordinate axes $x(1), x(2)$, according to Fig. 3. Then it is easily shown that the elements $x(1), x(2)$ are uncorrelated and have equal variances (equal to 1/3), thus, $\mathbf{x}$ is whitened (sphered). However, the elements are not independent. The problem is to find a rotation $\mathbf{y} = \mathbf{R}\mathbf{x}$ of $\mathbf{x}$ such that the elements of the rotated vector $\mathbf{y}$ are statistically independent. It is obvious from Fig. 3 that the elements of $\mathbf{y}$ must be aligned with the orientation of the square, because then and only then the density is separable into two marginal densities: $p(\mathbf{y}) = p_1(y(1))p_2(y(2))$, with both p_1 and p_2 one-dimensional uniform densities.

The solution minimizing the criterion (7), with $\mathbf{W} = [\mathbf{w}(1)\mathbf{w}(2)]$ a 2×2 matrix and $g(.)$ a suitable nonlinearity, provides now such a rotation. This can be seen as follows. Assume that g is a very sharp sigmoid, e.g. $g(y) = tanh(10y)$. Then it can be approximated by the *sign* function, and the term $\mathbf{W}g(\mathbf{W}^T\mathbf{x})$ in criterion (7)

Figure 4: The minimum value of the cost function $J(\mathbf{W})$ when the columns of $\mathbf{W}$ are orthogonal and rotated by an angle θ from the $(y(1), y(2))$ axes.

becomes

$$\begin{aligned} \mathbf{W}g(\mathbf{W}^T\mathbf{x}) &= \mathbf{w}(1)g(\mathbf{w}(1)^T\mathbf{x}) + \mathbf{w}(2)g(\mathbf{w}(2)^T\mathbf{x}) \quad (15) \\ &\approx \mathbf{w}(1)sign(\mathbf{w}(1)^T\mathbf{x}) + \mathbf{w}(2)sign(\mathbf{w}(2)^T\mathbf{x}). \quad (16) \end{aligned}$$

Thus according to (7), each $\mathbf{x}$ should be optimally represented by a point $(\pm\mathbf{w}(1) \pm \mathbf{w}(2))$, with the signs depending on the angles between $\mathbf{x}$ and the basis vectors. Each choice of the two orthogonal basis vectors divides the square of Fig. 3 into four quadrants, and by the criterion (7), all the points in a given quadrant must be represented in the least mean square sense by just one point (see Fig. 3); e.g. in the first quadrant, where the angles between $\mathbf{x}$ and the basis vectors are positive, by the point $\mathbf{w}(1)+\mathbf{w}(2)$. From Fig. 3, it can be seen that the optimal fit is obtained when the basis vectors are aligned with the axes $y(1), y(2)$, and the point $\mathbf{w}(1) + \mathbf{w}(2)$ is the center of the smaller square bordered by the positive $y(1), y(2)$ axes.

For further confirmation, it is easy to compute the theoretical value of the cost function $J(\mathbf{W})$ of eq. (7) when the basis vectors $\mathbf{w}(1)$ and $\mathbf{w}(2)$ are arbitrary orthogonal vectors. Denoting the angle between $\mathbf{w}(1)$ and the $y(1)$ axis in Fig. 3 by θ, Fig. 4 shows the minimal value (with respect to the lengths of the basis vectors) of $J(\mathbf{W})$ for each θ in the range $[0, \pi/4]$. The rotation $\theta = 0$ gives the absolute minimum, and then the lengths of the orthogonal vectors are equal to 0.5. The vectors are shown in Fig. 3.

In the above example, it was assumed that the marginal densities of the elements were uniform. For some other densities, the same effect of rotation into independent directions would not be achieved. Certainly, this would not take place for Gaussian densities with equal variances, for which the criterion $J(\mathbf{W})$ would be independent of the orientation. For *super-Gaussian* densities that are "sharper" than the Gaussian, e.g. the Laplacian or two-sided exponential density, minimization of the criterion $J(\mathbf{W})$ would result in the most dependent directions. Note that uniform density is *sub-Gaussian* or broader than the Gaussian; the relevant measure here is the kurtosis (see e.g. [11]), defined for a random variable y as

$$kur(y) = E\{y^4\} - (E\{y^2\})^2. \tag{17}$$

The kurtosis is negative for sub-Gaussians, zero for Gaussians, and positive for super-Gaussians.

The example was also dependent on the nonlinearity $g(y)$. A more general result showing the asymptotic values of algorithm (4) will be given in Section 3.

3 Analysis of the Nonlinear PCA learning rule

We start from the learning rule (4)

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{x}_k - \mathbf{W}_k g(\mathbf{y}_k)]g(\mathbf{y}_k^T) \tag{18}$$

with $\mathbf{y}_k = \mathbf{W}_k^T \mathbf{x}_k$. This covers also the one-unit case of eq. (3) when matrix $\mathbf{W}_k$ has only one column. Assumption 1 given in Section 1 holds. Thus the input vectors $\mathbf{x}_k$ are whitened: $E\{\mathbf{x}_k \mathbf{x}_k^T\} = \mathbf{I}$, and we assume that there exists a square separating matrix $\mathbf{B}$ such that vector $\mathbf{u}_k = \mathbf{B}^T \mathbf{x}_k$ has independent elements and also unit variances: $E\{\mathbf{u}_k \mathbf{u}_k^T\} = \mathbf{I}$.

This implies that the separating matrix must be orthogonal:

$$\mathbf{I} = E\{\mathbf{u}_k \mathbf{u}_k^T\} = E\{\mathbf{B}^T \mathbf{x}_k \mathbf{x}_k^T \mathbf{B}\} = \mathbf{B}^T E\{\mathbf{x}_k \mathbf{x}_k^T\}\mathbf{B} = \mathbf{B}^T\mathbf{B}, \tag{19}$$

hence $\mathbf{B}^{-1} = \mathbf{B}^T$. To make the analysis easier, we proceed by making a linear transformation to the learning rule (4): we multiply both sides by $\mathbf{B}^T$, with $\mathbf{B}$ an orthogonal separating matrix. We obtain

$$\begin{aligned} \mathbf{B}^T\mathbf{W}_{k+1} &= \mathbf{B}^T\mathbf{W}_k + \mu_k[\mathbf{B}^T\mathbf{x}_k - \mathbf{B}^T\mathbf{W}_k g(\mathbf{W}_k^T\mathbf{x}_k)]g(\mathbf{x}_k^T\mathbf{W}_k) && (20) \\ &= \mathbf{B}^T\mathbf{W}_k + \mu_k[\mathbf{B}^T\mathbf{x}_k - \mathbf{B}^T\mathbf{W}_k g(\mathbf{W}_k^T\mathbf{B}\mathbf{B}^T\mathbf{x}_k)]g(\mathbf{x}_k^T\mathbf{B}\mathbf{B}^T\mathbf{W}_k) && (21) \end{aligned}$$

where we have used the fact that $\mathbf{B}\mathbf{B}^T = \mathbf{I}$. Denoting for the moment $\mathbf{S}_k = \mathbf{B}^T\mathbf{W}_k$ and using the definition $\mathbf{u}_k = \mathbf{B}^T\mathbf{x}_k$ given above, we have

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \mu_k[\mathbf{u}_k - \mathbf{S}_k g(\mathbf{S}_k^T\mathbf{u}_k)]g(\mathbf{u}_k^T\mathbf{S}_k)]. \tag{22}$$

This equation has exactly the same form as the original one. Geometrically the transformation by the orthogonal matrix $\mathbf{B}$ simply means a coordinate change to a new set of coordinates such that the elements of the input vector expressed in these coordinates are statistically independent.

3.1 The multi-unit case

The goal here is to show that if the number of neural units is *equal to the dimensionality*, i.e. the weight matrix $\mathbf{W}$ is square, then the matrix sequence $\mathbf{W}_k$ of eq. (4) will tend to a scaled version of a separating matrix $\mathbf{B}$. Note that $\mathbf{B}$ is not unique in the sense that the elements of $\mathbf{u}_k$ can be permuted and multiplied by -1 without changing the independence and unit variance assumptions. Now the basic requirement for the transformed weight matrix $\mathbf{S}_k = \mathbf{B}^T\mathbf{W}_k$ in eq. (22) is that, starting from an arbitrary initial value, it should tend to the unit matrix or a permutation matrix. Then also $\mathbf{y}_k = \mathbf{S}_k^T\mathbf{u}_k$ would tend to the input vector $\mathbf{u}_k$, or a permuted version, with independent components.

However, it turns out that in the learning rule (22), the unit matrix or a permutation matrix cannot generally be a fixed point. We make instead the more general requirement that $\mathbf{S}_k$ tend to a diagonal matrix or a diagonal matrix times a permutation matrix. In this case the elements of $\mathbf{y}_k = \mathbf{S}_k^T\mathbf{u}_k$ will become the elements of input vector $\mathbf{u}_k$, in some order, multiplied by some numbers. In view of the original problem, in which the amplitudes of the signals $s_k(i)$ remain unknown, this is actually no restriction.

To proceed, the difference equation (22) can be further analyzed by writing down the corresponding averaged differential equation; for a discussion of the technique, see e.g. [13]. The limit of convergence of the difference equation is among the asymptotically stable solutions of the averaged differential equation. Taking averages in (22) with respect to the density of $\mathbf{u}_k$, and using $\mathbf{Z} = \mathbf{Z}(t)$ as the continuous-time counterpart of the transformed weight matrix $\mathbf{S}_k$, we obtain

$$d\mathbf{Z}/dt = G(\mathbf{Z}) - \mathbf{Z}H(\mathbf{Z}), \tag{23}$$

with

$$\begin{aligned} G(\mathbf{Z}) &= E\{\mathbf{u}g(\mathbf{u}^T\mathbf{Z})\}, &(24)\\ H(\mathbf{Z}) &= E\{g(\mathbf{Z}^T\mathbf{u})g(\mathbf{u}^T\mathbf{Z})\}. &(25)\end{aligned}$$

The expectations are over the (unknown) density of vector $\mathbf{u}$. We are ready to prove the main result of this section:

Theorem 1. In the matrix differential equation (23), assume the following:

1. The random vector $\mathbf{u}$ has a symmetrical density with $E\{\mathbf{u}\} = 0$;
2. The elements of $\mathbf{u}$, denoted here $u_1, ..., u_n$, are statistically independent;
3. The function $g(.)$ is odd, i.e., $g(y) = -g(-y)$ for all y, and at least twice differentiable everywhere;

4. The function $g(.)$ and the density of $\mathbf{u}$ are such that the following conditions hold for all $i = 1, ..., n$:

$$A_i = E\{u_i^2 g'(\alpha_i u_i)\} - 2\alpha_i E\{g(\alpha_i u_i) g'(\alpha_i u_i) u_i\} - E\{g^2(\alpha_i u_i)\} < 0, \qquad (26)$$

where the $\alpha_i,\ i = 1, ..., n$ are scalars satisfying

$$E\{u_i g(\alpha_i u_i)\} = \alpha_i E\{g^2(\alpha_i u_i)\}. \qquad (27)$$

5. Denoting

$$\begin{aligned} \sigma_i^2 &= E\{u_i^2\} & (28)\\ B_i &= E\{g^2(\alpha_i u_i)\} & (29)\\ C_i &= E\{g'(\alpha_i u_i)\} & (30)\\ F_i &= E\{u_i g(\alpha_i u_i)\}, & (31) \end{aligned}$$

both the eigenvalues of the 2×2 matrix

$$\begin{pmatrix} \sigma_i^2 C_j - \alpha_i C_j F_i - B_j & -\alpha_i C_i F_j \\ -\alpha_j C_j F_i & \sigma_j^2 C_i - \alpha_j C_i F_j - B_i \end{pmatrix} \qquad (32)$$

have strictly negative real parts for all $i, j = 1, ..., n$.

Then the matrix

$$\mathbf{Z} = \mathbf{D} = diag(\alpha_1, ..., \alpha_n) \qquad (33)$$

is an asymptotically stable stationary point of (23), where α_i satisfies eq. (27).

The proof is given in the Appendix.

Note 1. We only consider a diagonal matrix $\mathbf{D}$ as the asymptotically stable solution. However, any permutation of $\mathbf{D}$ can be shown to be an asymptotically stable solution, too, by making another orthogonal rotation of the coordinate axes that will permute some of them. This simply means re-indexing of the vector elements u_i. Mathematically, by going from $\mathbf{Z}(t)$ to $\mathbf{PZ}(t)$, with $\mathbf{P}$ a permutation (an orthogonal matrix), an analogous differential equation is obtained, and the conditions of the Theorem are unaltered.

Note 2. Due to the oddity of function $g(y)$, the signs of the α_i cannot be determined from eq. (27); if $+\alpha_i$ is a solution, then so is also $-\alpha_i$. If the weight matrix $\mathbf{S}_k$ of eq. (22) converges to $\mathbf{D}$, then asymptotically the i-th element of $\mathbf{y}_k = \mathbf{D}\mathbf{u}_k$ is the i-th element of $\mathbf{u}_k$ multiplied by $\pm\alpha_i$. The sign has no influence on the absolute magnitude.

Note 3. Theorem 1 allows non-monotonic activation functions. However, if $g(y)$ is monotonic, then eq. (27) in fact implies that it must be an *increasing* function.

If $g(y)$ were monotonically decreasing and odd, then the left hand side would be negative for positive α_i and positive for negative α_i; but then there could not be any solution because $E\{g^2(\alpha_i u_i)\} > 0$.

The exact conditions of Theorem 1 in a general case are fairly complex. Let us *simplify the situation* by an extra assumption: all the marginal densities $p(u_i)$ are identical. Then all the α_i parameters from eq. (27) will be the same except for the arbitrary sign. Also, all the parameters σ_i^2, B_i, C_i, F_i will be independent of i. Let us look at the stability condition for the points $\mathbf{D} = diag(\alpha, \alpha, ..., \alpha)$, i.e. for the case that all the α parameters have the same positive sign. We have

Corollary 1. Assume that the conditions of Theorem 1 hold, and assume further that the probability densities of all the u_i are identical. Consider only the case $sign(\alpha_i) = 1$ for all i. Then the larger eigenvalue of the matrix (32) is equal to

$$\lambda_{max} = \sigma^2 C - B, \tag{34}$$

with σ^2, C, and B given in eqs. (28), (30), and (29).

The proof is given in the appendix.

The corollary will now be illustrated for two types of nonlinear activation functions: polynomial functions $g(y) = y^s$, with s an odd positive integer, and sigmoidal functions $g(y) = tanh(\beta y)$, with β a positive slope parameter. All these functions obviously satisfy the condition (3) of the Theorem.

1. *Polynomials.* The family of odd polynomial functions

$$g(y) = y^s, \; s = 1, 3, 5, 7, ... \tag{35}$$

is interesting in the present context because all the relevant variables in the conditions (4) and (5) of Theorem 1, for any probability density, will become *moments* of u_i. These functions also contain the linear function for $s = 1$.

First, for α we get from eq. (27):

$$E\{u^{s+1}\} = \alpha^{s+1} E\{u^{2s}\}. \tag{36}$$

Substituting this in eq. (26) of condition (4), we find that this condition is always satisfied.

Now, using the expression given in Corollary 1 for the larger eigenvalue gives the stability condition (condition 5 of Theorem 1)

$$E\{u^{s+1}\} - sE\{u^2\}E\{u^{s-1}\} > 0. \tag{37}$$

Consider first the case

$$s = 1, \; g(u) = u. \tag{38}$$

Clearly, the condition (37) is not satisfied. *The linear function never gives asymptotic stability, if the densities of the elements of u_i are all similar.* Inequality (37) is satisfied as an equality, however, which means that the solution in this case is stable but not asymptotically stable. In fact the stability of the linear function (when the nonlinear PCA learning rule becomes the Subspace rule) has been already established earlier, see e.g. [17].

Consider next the case

$$s = 3, \; g(u) = u^3. \tag{39}$$

Now (37) gives

$$E\{u^4\} - 3(E\{u^2\})^2 > 0. \tag{40}$$

This expression is exactly the *kurtosis* or the fourth order cumulant of u [11]. If and only if the density is *positively kurtotic* or *super-Gaussian*, this condition is satisfied and the cubic polynomial $g(u) = u^3$ *gives asymptotic stability.*

Likewise, for $s = 5$ we get the condition

$$E\{u^6\} - 5E\{u^2\}E\{u^4\} > 0, \tag{41}$$

etc. While this expression is not directly a cumulant, it is related to the second, fourth, and sixth order cumulants. The sixth order cumulant is [11]

$$\begin{aligned}
cum^6(u) &= E\{u^6\} - 15E\{u^2\}E\{u^4\} + 30(E\{u^2\})^3 && (42)\\
&= E\{u^6\} - 5E\{u^2\}E\{u^4\} - 10E\{u^2\}E\{u^4\} + 30(E\{u^2\})^3 && (43)\\
&= E\{u^6\} - 5E\{u^2\}E\{u^4\} - 10E\{u^2\}[E\{u^4\} - 3(E\{u^2\})^2] && (44)\\
&= E\{u^6\} - 5E\{u^2\}E\{u^4\} - 10E\{u^2\}cum^4(u), && (45)
\end{aligned}$$

showing that the left hand side of condition (41) can also be written as $cum^6(u) + 10cum^2(u)cum^4(u)$.

2. *Hyperbolic tangents.* Consider then the sigmoidal activation function $g(y) = tanh(\beta y)$, for $\beta > 0$, that has the $sign(y)$ function as the limit as $\beta \to \infty$. Assuming $\sigma^2 = 1$, the stability condition of Corollary 1 becomes $C < B$, or $E\{g'(\alpha u)\} < E\{g^2(\alpha u)\}$. For the hyperbolic tangent, $g'(y)$ has a peak around the origin and decreases to both sides, while $g^2(y)$ is zero at the origin and increases to both sides (see Fig. 5). In this case it is clear that a peaked super-Gaussian density for u makes $E\{g'(\alpha u)\}$ large and $E\{g^2(\alpha u)\}$ small, while a flat sub-Gaussian does just the opposite. The latter case is then more stable. The same effect was discussed in the example of Section 2.

A simple example of a sub-Gaussian density is the uniform density on $[-1, 1]$. Let us assume this for the elements of $\mathbf{u}$ to illustrate the Theorem 1. Condition 1 of Theorem 1 is then satisfied. It remains to check the stability conditions 4 and 5 of Theorem 1. Now, a closed form solution for α in eq. (27) is not feasible and numerical methods must be used. Fig. 6 shows the value of α and the two eigenvalues of Condition (5) as values of β. Condition (4) is always satisfied. The

Figure 5: $g'(y)$ and $g^2(y)$ for the function $g(y) = tanh(2y)$.

conclusion is that *for the uniform density the sigmoidal function gives asymptotic stability* when $\beta > 0$. However, the larger of the two negative eigenvalues is quite close to zero for small values of β; once α is known, it is possible to evaluate λ_{max} from Corollary 1 in closed form and the result is

$$\lambda_{max} = tanh(\alpha\beta)\frac{\beta + 3}{3\alpha\beta} - 1. \tag{46}$$

This is negative but tends to zero as $\beta \to 0$.

This result was tested with a simulation using the original continuous-time learning equation (23). The input vector dimension was 3, and the continuous uniform densities of its elements were replaced here by a grid of equiprobable points from -1.0 to 1.0, with spacing 0.2, for each element of $\mathbf{u}$. Then the expectations on the right hand side of (23) were easily computed as sums. The initial deviation of the solution matrix from the theoretical limit could be adjusted. In Fig. 7, the effect of β on the convergence is shown. The curves are the 3 diagonal elements of $\mathbf{Z}(t)$ and the absolute value of the largest off-diagonal element. For small initial deviations at least, the diagonal elements should tend to the respective value of α and the off-diagonals should tend to zero, i.e. $\mathbf{Z}(t) \to diag(\alpha, \alpha, \alpha)$. As can be seen from Fig. 7, this happens for $\beta = 1, 5$, and 20, while for $\beta = 0.5$ the off-diagonals fail to tend to zero. The initial deviation was here $norm(\mathbf{Z}(0) - diag(\alpha, \alpha, \alpha)) = 0.1$.

Asymptotic stability is a local effect, and Theorem 1 does not say anything about the basin of attraction of the asymptotic solution, i.e., *global stability*. This was tested numerically in another series of runs, in which the value $\beta = 5.0$ was chosen for the sigmoid parameter and the initial deviation was varied. For this value of β, and the uniform densities used, $\alpha = 0.6998$. Global stability can be seen from Fig. 8. The deviation was increased up to 100.0 and the algorithm converged invariably

Figure 6: The value of α (norm of the columns of the asymptotic solution $\mathbf{D}$) (solid line) and the two eigenvalues of Theorem 1 (dotted lines), when the steepness β of the sigmoidal nonlinearity was varied from 0.25 to 128.

to one of the asymptotically stable solutions predicted by Theorem 1. Note that when the initial deviation is increased, it may happen that the asymptotic limit for $\mathbf{Z}(t)$ will not be $diag(\alpha, \alpha, \alpha)$ but a permutation, with possibly changed signs. Thus e.g. in Fig. 8c, all the diagonal elements tend to zero, while the largest off-diagonal tends to $\alpha = 0.6998$. The reason is that in this case the asymptotic value turned out to be

$$lim\ \mathbf{Z}(t) = \begin{pmatrix} 0.0000 & -0.6998 & 0.0000 \\ 0.0000 & 0.0000 & 0.6998 \\ 0.6998 & 0.0000 & 0.0000 \end{pmatrix} \tag{47}$$

which is a permutation of matrix $\mathbf{D}$. Note also the negative sign in one of the nonzero elements.

3.2 The one-unit case

Finally, consider the one-unit algorithm (1). This algorithm will approximately minimize the cost function

$$J(\mathbf{w}) = E\{\|\mathbf{x} - g(\mathbf{w}^T\mathbf{x})\mathbf{w}\|^2\}. \tag{48}$$

Thus, if $g(y)$ is the sign function as in the example of Section 2, $\mathbf{x}$ should be approximated by $\mathbf{w}$ everywhere in the half-space defined by $\mathbf{w}^T\mathbf{x} > 0$ and by $-\mathbf{w}$ in the

Figure 7: The three diagonal elements (the three upper curves) and the absolute value of the largest off- diagonal element of matrix $\mathbf{Z}(t)$ (the lowermost curve). During one epoch, a set of equidistant sampling points within the unit cube were taken and the average of the right hand side of the differential equation was computed. The value of steepness parameter β is shown on the graphs. The norm of the deviation of $\mathbf{Z}(0)$ from the theoretical limit was always 0.1.

Figure 8: As in Fig. 7, but now $\beta = 5.0$ and the initial deviation was varied. The value of the initial deviation is shown on the graphs.

Figure 9: Gaussian mixture with equal variances in each direction.

complementary half-space. For example, if the density of $\mathbf{x}$ is the one given in Fig. 9 (which has equal variances in every direction), and the Gaussian peaks are sharp enough, then the minimum value for $J(\mathbf{w})$ is attained when $\mathbf{w}$ is the central point of one of the Gaussian peaks. Thus, the orientation of $\mathbf{w}$ would be aligned with the non-Gaussian (horizontal) axis. This is the goal in *Exploratory Projection Pursuit* [4]: to find orientations in the input space that are as non-Gaussian as possible. For connections of nonlinear PCA and Projection Pursuit, see [18].

The purpose of this subsection is to give a similar result to Theorem 1 for the one-unit case, which could then be analyzed in a number of special cases. Again, the same transformation can be applied to the learning rule as earlier to make the elements of the input vectors independent, and the ensuing averaged differential equation will be

$$d\mathbf{z}/dt = G(\mathbf{z}) - \mathbf{z}H(\mathbf{z}) \tag{49}$$

with

$$\begin{aligned} G(\mathbf{z}) &= E\{\mathbf{u}g(\mathbf{u}^T\mathbf{z})\}, & (50)\\ H(\mathbf{z}) &= E\{g^2(\mathbf{u}^T\mathbf{z})\}. & (51) \end{aligned}$$

Thus, H is now a scalar function of $\mathbf{z}$.

We can show the following result, in which the assumptions of Theorem 1 of Section 3.1 are used.

Theorem 2. Let the assumptions 1 to 4 of Theorem 1 hold. With σ_i^2, B_i and C_i defined as in Theorem 1, assume that

$$\sigma_j^2 C_i - B_i < 0 \tag{52}$$

for all $j \neq i$. Then the vector

$$\mathbf{z} = (0, 0, ..., \alpha_i, ..., 0), \tag{53}$$

i.e., the i-th column of matrix $\mathbf{D}$ of Theorem 1, is an asymptotically stable solution of (49).

The proof is given in the Appendix.

Again, this case can be simulated with artificial data, showing a good agreement between the theoretical result and Theorem 2.

The importance of the one-unit case is that now it is possible to separate *only one of the independent components* from all the others. With the weight vector of the one neuron tending to (53), the linear output of the neuron is $\mathbf{z}^T\mathbf{u} = \alpha_i u_i$ which is a scaled version of the element u_i. Which of the component signals is obtained depends on the activation function and the signal densities in the special way given by eq. (52). If all the densities are equal, then condition (52) becomes the same as the condition (34) of Corollary 1, and the special cases discussed after Corollary 1 apply also for the one-unit case. Thus, e.g. for uniform densities and a $tanh(y)$ activation function, all the columns of the matrix $diag(\alpha, ..., \alpha)$ are asymptotically stable. It depends then on the initial value where the weight vector will converge.

4 Conclusions

The purpose here was to give rigorous mathematical results on the Nonlinear PCA learning rule, with little experimental validation outside artificial examples. More detailed experiments with real data will be reported elsewhere [8], [9], [19]. These papers also review and introduce a number of other related neural network algorithms for computing the independent components, like the Bigradient learning rule that is minimizing or maximizing a cost function directly related to higher-order cumulants like the kurtosis.

The main result of this paper is that when a *nonlinear activation function* is introduced to the basic linear neurons of the PCA network, keeping the learning rule otherwise unaltered, the performance of the network is radically altered. While the linear network finds the PCA subspace, and some versions of the network like the SGA (see e.g. [17]) also find the individual principal component directions, the nonlinear network is not as good in PCA. But if the effect of the second order statistics is removed altogether by whitening or sphering the input data, then the remaining higher order effects make it possible for the network to find the *statistically independent directions* of the input space. A sufficient requirement is that the input signals are sub-Gaussian (negatively kurtotic), i.e., histogram equalized signals or images, and the activation function is a *tanh* sigmoid. Another possibility is a polynomial activation function and a super-Gaussian (positively kurtotic) input signal density. These effects can be understood because the learning rule is approximately minimizing a nonlinear least mean square representation of the input vectors.

5 APPENDIX

Proof of Theorem 1. First, we show that (33) is indeed a stationary point. Substitution on the right side of eq. (23) gives the requirement $G(\mathbf{D}) = \mathbf{D}H(\mathbf{D})$, whose i,j-th element is

$$E\{u_i g(\alpha_j u_j)\} = \alpha_i E\{g(\alpha_i u_i) g(\alpha_j u_j)\}. \tag{54}$$

Because $g(.)$ is an odd function and the density of each u_i is symmetrical and zero mean, it follows that $E\{g(\alpha_i u_i)\} = 0$ for all i. Because u_i and u_j are statistically independent, both sides of (54) are zero for $i \neq j$ and the equation is satisfied. For the diagonal elements we obtain eq. (27). So, $\mathbf{D}$ is indeed a stationary point.

To show the asymptotic stability, we make a deviation to the stationary point and show that the deviation converges to zero. Therefore, let $\mathbf{Z}(t) = \mathbf{D} + \mathbf{E}(t)$ with $\mathbf{D}$ the stationary point above, and assume that the initial value $\mathbf{E}(0)$ is a small arbitrary matrix. It holds for $\mathbf{E}(t)$:

$$d\mathbf{E}/dt = d\mathbf{Z}/dt = G(\mathbf{D}+\mathbf{E}) - (\mathbf{D}+\mathbf{E})H(\mathbf{D}+\mathbf{E}), \tag{55}$$

giving the corresponding equation for the i,j-th element of $\mathbf{E}$, say, e_{ij}:

$$de_{ij}/dt = G_{ij}(\mathbf{D}+\mathbf{E}) - \sum_{k=1}^{n} (\mathbf{D}+\mathbf{E})_{ik} H_{kj}(\mathbf{D}+\mathbf{E}). \tag{56}$$

Taking the first term on the right hand side gives $G_{ij}(\mathbf{D}+\mathbf{E}) = E\{u_i g(\mathbf{u}^T\mathbf{d}_j + \mathbf{u}^T\mathbf{e}_j)\}$ with $\mathbf{d}_j$ and $\mathbf{e}_j$ denoting the j-th columns of $\mathbf{D}$ and $\mathbf{E}$, respectively. Because $g(.)$ is at least twice differentiable, we obtain $g(\mathbf{u}^T\mathbf{d}_j + \mathbf{u}^T\mathbf{e}_j) = g(\mathbf{u}^T\mathbf{d}_j) + g'(\mathbf{u}^T\mathbf{d}_j)(\mathbf{u}^T\mathbf{e}_j) + O(e_{ij}^2)$, and substituting this gives

$$\begin{aligned}
G_{ij}(\mathbf{D}+\mathbf{E}) &= E\{u_i g(\mathbf{u}^T\mathbf{d}_j)\} + E\{u_i g'(\mathbf{u}^T\mathbf{d}_j)(\mathbf{u}^T\mathbf{e}_j)\} + O(e_{ij}^2) && (57)\\
&= G_{ij}(\mathbf{D}) + E\{u_i g'(\alpha_j u_j)(\textstyle\sum u_k e_{kj})\} + O(e_{ij}^2) && (58)\\
&= G_{ij}(\mathbf{D}) + \textstyle\sum e_{kj} E\{u_k u_i g'(\alpha_j u_j)\} + O(e_{ij}^2) && (59)\\
&= G_{ij}(\mathbf{D}) + e_{ij} E\{u_i^2 g'(\alpha_j u_j)\} + O(e_{ij}^2). && (60)
\end{aligned}$$

Above, we have used the independence of u_k and u_i to set $E\{u_k u_i g'(\alpha_j u_j)\} = 0$ for $k \neq i$.

Consider then the term $H_{kj}(\mathbf{D}+\mathbf{E})$ in eq. (56). As above, we get by a series expansion

$$\begin{aligned}
H_{kj}(\mathbf{D}+\mathbf{E}) &= E\{g(\mathbf{d}_k^T\mathbf{u} + \mathbf{e}_k^T\mathbf{u}) g(\mathbf{u}^T\mathbf{d}_j + \mathbf{u}^T\mathbf{e}_j)\} && (61)\\
&= E\{g(\alpha_k u_k + \mathbf{e}_k^T\mathbf{u}) g(\alpha_j u_j + \mathbf{e}_j^T\mathbf{u})\} && (62)\\
&= E\{g(\alpha_k u_k) g(\alpha_j u_j)\} + E\{g'(\alpha_k u_k) g(\alpha_j u_j)(\mathbf{e}_k^T\mathbf{u}) && (63)\\
&+ g(\alpha_k u_k) g'(\alpha_j u_j)(\mathbf{e}_j^T\mathbf{u})\} + O(e_{ij}^2) && (64)\\
&= H_{kj}(\mathbf{D}) + E\{g'(\alpha_k u_k) g(\alpha_j u_j) u_j\} e_{jk} && (65)\\
&+ E\{g(\alpha_k u_k) g'(\alpha_j u_j) u_k\} e_{kj} + O(e_{ij}^2), && (66)
\end{aligned}$$

where again by the independence of the elements of $\mathbf{u}$ we have used the substitution $E\{g'(\alpha_k u_k)g(\alpha_j u_j)(\sum u_s e_{sk})\} = E\{g'(\alpha_k u_k)g(\alpha_j u_j)u_j\}e_{jk}$.

Finally, the term $(\mathbf{D}+\mathbf{E})_{ik}$ in (56) becomes

$$(\mathbf{D}+\mathbf{E})_{ik} = \mathbf{D}_{ik} + \mathbf{E}_{ik}. \tag{67}$$

Combining terms yields first

$$\begin{aligned}
& \sum_k (\mathbf{D}+\mathbf{E})_{ik} H_{kj}(\mathbf{D}+\mathbf{E}) && (68)\\
= & \sum_k (\mathbf{D})_{ik} H_{kj}(\mathbf{D}) + \sum_k (\mathbf{D})_{ik} E\{g'(\alpha_k u_k) g(\alpha_j u_j) u_j\} e_{jk} && (69)\\
+ & \sum_k (\mathbf{D})_{ik} E\{g(\alpha_k u_k) g'(\alpha_j u_j) u_k\} e_{kj} + \sum_k (\mathbf{E})_{ik} H_{kj}(\mathbf{D}) + O(e_{ij}^2) && (70)\\
= & \sum_k (\mathbf{D})_{ik} H_{kj}(\mathbf{D}) + \alpha_i E\{g'(\alpha_i u_i) g(\alpha_j u_j) u_j\} e_{ji} && (71)\\
+ & \alpha_i E\{g(\alpha_i u_i) g'(\alpha_j u_j) u_i\} e_{ij} + \sum_k e_{ik} E\{g(\alpha_k u_k) g(\alpha_j u_j)\} + O(e_{ij}^2) && (72)\\
= & \sum_k (\mathbf{D})_{ik} H_{kj}(\mathbf{D}) + \alpha_i E\{g'(\alpha_i u_i) g(\alpha_j u_j) u_j\} e_{ji} && (73)\\
+ & \alpha_i E\{g(\alpha_i u_i) g'(\alpha_j u_j) u_i\} e_{ij} + e_{ij} E\{g^2(\alpha_j u_j)\} + O(e_{ij}^2). && (74)
\end{aligned}$$

Collecting all the terms on the right hand side of eq. (56) we obtain

$$\begin{aligned}
de_{ij}/dt &= G_{ij}(\mathbf{D}) + e_{ij} E\{u_i^2 g'(\alpha_j u_j)\} && (75)\\
&- \sum_k (\mathbf{D})_{ik} H_{kj}(\mathbf{D}) - \alpha_i E\{g'(\alpha_i u_i) g(\alpha_j u_j) u_j\} e_{ji} && (76)\\
&- \alpha_i E\{g(\alpha_i u_i) g'(\alpha_j u_j) u_i\} e_{ij} - e_{ij} E\{g^2(\alpha_j u_j)\} + O(e_{ij}^2) && (77)\\
&= (E\{u_i^2 g'(\alpha_j u_j)\} - \alpha_i E\{g(\alpha_i u_i) g'(\alpha_j u_j) u_i\} - E\{g^2(\alpha_j u_j)\}) e_{ij} && (78)\\
&- \alpha_i E\{g'(\alpha_i u_i) g(\alpha_j u_j) u_j\} e_{ji} + O(e_{ij}^2), && (79)
\end{aligned}$$

where the fact that $\mathbf{D}$ is a fixed point has been used.

The linear part of the differential equation (79) determines the asymptotic stability. Consider first the case of diagonal elements e_{ii}: we get

$$de_{ii}/dt = A_i e_{ii} \tag{80}$$

with

$$A_i = E\{u_i^2 g'(\alpha_i u_i)\} - 2\alpha_i E\{g(\alpha_i u_i) g'(\alpha_i u_i) u_i\} - E\{g^2(\alpha_i u_i)\}. \tag{81}$$

This must be negative in order for e_{ii} to tend to zero, giving the condition (4) of the Theorem.

Consider then the off-diagonal elements e_{ij}. Because $i \neq j$, the coefficients in eq. (79) decouple. Let us introduce some notation for the statistics involved:

$$\begin{aligned}
\sigma_i^2 &= E\{u_i^2\} && (82)\\
B_i &= E\{g^2(\alpha_i u_i)\} && (83)\\
C_i &= E\{g'(\alpha_i u_i)\} && (84)\\
F_i &= E\{u_i g(\alpha_i u_i)\}. && (85)
\end{aligned}$$

Substituting these in eq. (79) and neglecting the higher order terms yields now

$$de_{ij}/dt = (\sigma_i^2 C_j - \alpha_i C_j F_i - B_j)e_{ij} - \alpha_i C_i F_j e_{ji}. \tag{86}$$

For symmetry reasons a similar equation holds for de_{ji}/dt:

$$de_{ji}/dt = (\sigma_j^2 C_i - \alpha_j C_i F_j - B_i)e_{ji} - \alpha_j C_j F_i e_{ij}. \tag{87}$$

We have two linear first order differential equations in two functions. The asymptotic convergence of e_{ij} and e_{ji} to zero depends on the real parts of the roots of the characteristic polynomial, which must be strictly negative. This is the condition (5) of the Theorem.

Proof of Theorem 2. The proof uses heavily the proof of Theorem 1 above. first, it is easy to show by the same arguments that the vector $\mathbf{z}$ of eq. (53) is a fixed point under the given condition for α_i. Perturbing this fixed point by a vector $\mathbf{e} = (e_1, ..., e_n)^T$ with small elements we obtain for e_i

$$de_i/dt = E\{u_i^2 g'(\alpha_i u_i)\}e_i - E\{g^2(\alpha_i u_i)\}e_i - 2\alpha_i E\{g(\alpha_i u_i)g'(\alpha_i u_i)u_i\}e_i + O(\|\mathbf{e}\|^2). \tag{88}$$

The condition (4) of Theorem 1 ($A_i < 0$) ensures that the coefficient of e_i on the right hand side will be negative and therefore e_i will tend to zero. When $j \neq i$, we obtain from the differential equation

$$de_j/dt = E\{u_j^2\}E\{g'(\alpha_i u_i)\}e_j - E\{g^2(\alpha_i u_i)\}e_j + O(\|\mathbf{e}\|^2), \tag{89}$$

and if the condition (52) holds, then the coefficient is negative and e_j will tend to zero.

Proof of Corollary 1. The matrix (32) becomes now

$$\begin{pmatrix} \sigma^2 C - \alpha CF - B & -\alpha CF \\ -\alpha CF & \sigma^2 C - \alpha CF - B \end{pmatrix} \tag{90}$$

and it is easy to see that its characteristic polynomial is

$$\Phi(\lambda) = (\sigma^2 C - \alpha CF - B - \lambda)^2 - (\alpha CF)^2. \tag{91}$$

The roots are

$$\lambda_{1,2} = \sigma^2 C - \alpha CF - B \pm \alpha CF. \tag{92}$$

From the definitions of C and F given in Theorem 1, it is clear that $\alpha CF > 0$ if $\alpha > 0$. Thus the larger root is equal to $\sigma^2 C - B$.

References

[1] P. Comon, "Independent component analysis - a new concept?," *Signal Processing*, **36**, 287-314 (1994).

[2] J. Friedman, "Exploratory projection pursuit," *J. Amer. Statistical Assoc.*, **82**, no. 397, 249-266 (1987).

[3] C. Fyfe, D.R. McGregor, and R. Baddeley, "Exploratory projection pursuit: an artificial neural network approach," Department of Computer Science, University of Strathclyde (Glasgow, Scotland), Research Report /94/160, Oct. 1994.

[4] P. Huber, "Projection pursuit," *The Annals of Statistics*, **13**, no. 2, 435-475 (1985).

[5] C. Jutten and J. Herault, "Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture," *Signal Processing*, **24**, no. 1, 1-10 (1991).

[6] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning," *Neural Networks*, **7**, no. 1, 113-127 (1994).

[7] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks*, **8**, no. 4, 549 - 562 (1995).

[8] J. Karhunen, L. Wang, and J. Joutsensalo, "Neural estimation of basis vectors in Independent Component Analysis," To appear in *Proc. Int. Conf. on Artificial Neural Networks* (Paris, France, 1995).

[9] J. Karhunen, L. Wang and R. Vigario, "Nonlinear PCA type approaches for source separation and independent component analysis," To appear in *Proc. Int. Conf. on Neural Networks* (Perth, Australia, 1995).

[10] B. Laheld and J.-F. Cardoso, "Adaptive source separation with uniform performance," in *Signal Processing VII: Theories and Applications (Proc. EUSIPCO-94)*, eds. M. Holt et al. (Lausanne: EURASIP, 1994, vol. 2, pp. 183-186).

[11] C. Nikias and J. Mendel, "Signal processing with higher-order spectra," *IEEE Signal Proc. Mag.*, July 1993, 10 - 37 (1993).

[12] E. Oja, "A simplified neuron model as a principal component analyzer," *J. Math. Biol.*, **15**, 267-273 (1982).

[13] E. Oja, *Subspace Methods of Pattern Recognition.* Letchworth, England: Research Studies Press, and New York, NY: J. Wiley (1983).

[14] E. Oja and J. Karhunen, " On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *J. Math. Anal. Appl.* **106**, 69-84 (1985).

[15] E. Oja, "Neural networks, principal components, and subspaces," *Int. J. Neural Systems*, **1**, 61-68 (1989).

[16] E. Oja, H. Ogawa, and J. Wangviwattana, " Learning in nonlinear constrained Hebbian networks," In *Artificial Neural Networks*, eds. T. Kohonen, K. Mkisara, O. Simula, and J. Kangas, (Amsterdam: Elsevier, 385 - 390 (1991)).

[17] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, **5**, 927 - 935 (1992).

[18] E. Oja, J. Karhunen, L. Wang, and R. Vigario, "Principal and independent components in neural networks - recent developments," To appear in *Proc. Italian Workshop on Neural Networks WIRN'95* (Vietri, Italy, 1995).

[19] L. Wang, J. Karhunen, E. Oja, and R. Vigario, "Blind separation of sources using nonlinear PCA type learning algorithms," To appear in *Proc. Int. Conf. on Neural Networks and Signal Processing* (Nanjing, P.R. China, 1995).