Signal Separation by Nonlinear Hebbian Learning

Erkki Oja and Juha Karhunen
Helsinki University of Technology
Laboratory of Computer and Information Science
Rakentajanaukio 2 C, FIN-02150 Espoo, Finland
Email: Juha.Karhunen@hut.fi, Erkki.Oja@hut.fi

Introduction

Principal Component Analysis (PCA) is a widely used technique in signal processing. It is now well-known how it can be realized in different ways using neural networks; for examples, cf. [4, 9, 21]. Recently, there has been an increasing interest in extending the unsupervised Hebbian learning rules used in PCA to *nonlinear Hebbian learning*: such techniques are often called *nonlinear PCA* methods. The main reason for this interest is that even though PCA is optimal for example in approximating the input data in the mean-square error sense, the representation that it provides is often not the most meaningful in describing some fundamental properties of the data. In PCA, the data are represented in an orthonormal basis determined merely by the second-order statistics (covariances) of the input data.

Various nonlinear PCA methods take into account higher-order statistics, too, and they may have efficient implementations in learning neural networks [20, 11, 12, 13]. Nonlinear or robust PCA type methods can be developed from various starting points, usually leading to mutually different solutions. We have earlier derived robust and nonlinear extensions of PCA starting either from maximization of output variances or minimization of mean-square representation error [11, 13]. Several other authors have proposed neural extensions of PCA by choosing optimization of some information-theoretic criterion as their starting point; see [9, 26] for further information.

Independent Component Analysis (ICA) is a useful extension of PCA that was developed in context with source or signal separation applications [6, 10]. In a sense, it is an extension of PCA: instead of requiring that the coefficients of a linear expansion of data vectors be uncorrelated, in ICA they must be mutually independent or as independent as possible. This implies that second order moments are not sufficient, but higher order statistics are needed in determining ICA. As will be seen later on, ICA provides in many cases a more meaningful representation of the data than PCA.

In this paper, we introduce a neural network that can be used for both source separation and the estimation of the basis vectors of ICA. The remainder of the paper is organized as follows. The next section presents the necessary background on ICA and source separation. In the third section, we introduce and justify the basic neural network learning algorithms for signal separation. The fourth section provides mathematical analysis justifying the separation ability of the nonlinear PCA type learning algorithm. The fifth section then introduces the ICA neural network, a three-layer network whose layers perform input data whitening, separation, and ICA basis vector estimation, respectively. In the sixth section, we present experimental results. In the last section, the conclusions of this study are presented, and some possibilities for extending the data model are outlined.

Source Separation and Independent Component Analysis

In source separation for linear memoryless channels, and the related technique of Independent Component Analysis (ICA), the following basic data model is usually assumed (see e.g. [10, 6, 4]). There are M scalar-valued signals $s_k(1), ..., s_k(M)$ indexed by an index k. We assume that the signals have zero mean and they are mutually statistically independent. More concretely, the signals could be sampled speech waveforms; for different speakers the signals are then at least approximately independent. Then the index k represents discrete time. Another example are discrete images: in Fig. 3, first row, the source signals are two-dimensional image arrays. In this example, the index i in the signal $s_k(i)$ stands for one of the images ($i = 1, 2, 3$), while the two-dimensional index k denotes the pixels. In the following discussion, Fig. 3 is repeatedly referred to as an example.

We assume that the original signals are *unobservable*, and all we have are a set of noisy linear mixtures $x_k(1), ..., x_k(L)$, with

$$x_k(j) = \sum_{i=1}^{M} s_k(i)a(ij) + n_k(j). \tag{1}$$

The coefficients $a(ij)$ are *unknown*. However, we assume that the mixtures are all different in the sense that the M vectors of mixture coefficients $\mathbf{a}(i) = (a(i1), ..., a(iL))^T$ are linearly independent; this also implies that the number of mixtures L must be equal to or larger than the number of signals M. Such mixtures arise in several practical situations like speech separation or antenna array processing. A pictorial example of mixture signals is provided by the second row of Fig. 3.

Denote by $\mathbf{x}_k = (x_k(1)....x_k(L))^T$ the L-dimensional k-th data vector made up of the mixtures (1) at discrete time (or point) k. By eq. (1), we can write *the signal model*:

$$\mathbf{x}_k = \mathbf{A}\mathbf{s}_k + \mathbf{n}_k = \sum_{i=1}^{M} s_k(i)\mathbf{a}(i) + \mathbf{n}_k, \tag{2}$$

where in the M-vector $\mathbf{s}_k = (s_k(1), \ldots, s_k(M))^T$, the element $s_k(i)$ denotes the ith source signal (independent component) at time k, $\mathbf{A} = (\mathbf{a}(1), \ldots, \mathbf{a}(M))$ is the $L \times M$ 'mixing matrix' whose columns $\mathbf{a}(i)$ are the basis vectors of ICA, and $\mathbf{n}_k$ denotes the vector of noise components. The noise term $\mathbf{n}_k$ is often omitted from (2), because under the weak assumptions made here it is usually not possible to separate the noise from the source signals.

The *source separation problem* [10, 3, 16] is now to find an $M \times L$ separating matrix $\mathbf{B}$ so that the M-vector

$$\mathbf{y}_k = \mathbf{B}\mathbf{x}_k \tag{3}$$

is an estimate $\mathbf{y}_k = \hat{\mathbf{s}}_k$ of the original independent source signals.

An example is given by the fourth row of Fig. 3, where the images have been obtained by applying a linear separation transformation to the mixture images on the second row of Fig. 3.

Various approaches have been proposed for achieving separation. The learning algorithms are constructed in such a way that they should satisfy some kind of independence condition after convergence. An example is the seminal Herault-Jutten (HJ) algorithm [10], which uses a neural-like network structure. This algorithm is simple and elegant, but may fail in separating more than two independent sources.

Most of the approaches dealing with signal separation and ICA are non-neural, and are based on some batch type or adaptive signal processing algorithm. However, some of these adaptive algorithms, such as the HJ algorithm and its modifications as well as the PFS/EASI algorithm proposed by Cardoso and Laheld [3, 16], can be interpreted as learning algorithms of a neural network. Quite recently, some authors [1, 7] have derived unsupervised "neural" learning rules from information-theoretic measures. The resulting algorithms show good separation performance, but are not truly realizable in neural networks because they are based on relatively complicated numerical operations, requiring for example the inversion of a matrix.

Several separation algorithms utilize the fact that if the data vectors $\mathbf{x}_k$ are first preprocessed by *whitening* or *sphering* them, i.e., if $E\{\mathbf{x}_k\mathbf{x}_k^T\} = \mathbf{I}$ (with $E\{.\}$ denoting the expectation), then the separating matrix $\mathbf{B}$ in (3) becomes orthogonal: $\mathbf{B}\mathbf{B}^T = \mathbf{I}$. This can be seen by writing $E\{\mathbf{y}_k\mathbf{y}_k^T\} = \mathbf{B}E\{\mathbf{x}_k\mathbf{x}_k^T\}\mathbf{B}^T = \mathbf{B}\mathbf{B}^T$, and this matrix must be at least diagonal because we require that the elements of y_k are zero-mean and statistically independent, hence uncorrelated. There is no way to find the energies of the individual source signals $s_k(i)$, as they are absorbed in the mixing coefficients, and therefore the elements of the vector y_k can be scaled so that y_k has unit covariance. In this case, $E\{\mathbf{y}_k\mathbf{y}_k^T\} = \mathbf{B}\mathbf{B}^T = \mathbf{I}$, hence $\mathbf{B}$ is an orthogonal matrix.

The whitening step was used in the example of Fig. 3, and the third row shows the whitened signals obtained from the mixtures on the second row. More details of this example are given in the sixth section of this paper.

It is usually not possible to verify the independence condition exactly in practice because the involved probability densities are unknown. Therefore, approximating *contrast functions* which are maximized for a separating matrix have been introduced [6]. Even these often lead to relatively intensive batch type computations. However, for *prewhitened input vectors* it can be shown [18] that a relatively simple contrast function, the sum of absolute values of the fourth order cumulants (kurtoses)

$$J_{kurt}(\mathbf{y}) = \sum_{i=1}^{M} \mid \mathrm{cum}(y(i)^4) \mid \; = \; \sum_{i=1}^{M} \mid E\{y(i)^4\} - 3E^2\{y(i)^2\} \mid \tag{4}$$

is maximized by a separating matrix $\mathbf{B}$ under certain conditions given by Moreau and Macchi [18]. The central conclusion is that among vectors of the form $\mathbf{y} = \mathbf{H}\mathbf{s}$, with vector $\mathbf{s}$ having independent components, the criterion (4) is maximized when $\mathbf{y}$ is either $\mathbf{s}$ itself or a permutation of its elements, possibly with changed signs.

For whitened vectors it holds $\mathrm{E}\{y(i)^2\} = 1$, implying that $\mathrm{cum}(y(i)^4) = E\{y(i)^4\} - 3$. Thus (4) is maximized by *minimizing* the sum of the fourth moments $\sum_{i=1}^{M} \mathrm{E}\{y(i)^4\}$ for *negatively kurtotic* sources, and by *maximizing* $\sum_{i=1}^{M} \mathrm{E}\{y(i)^4\}$ for *positively kurtotic* sources. Densities that have negative kurtosis are also called *sub-Gaussian*, because they are typically flatter than the Gaussian density whose kurtosis is zero; a typical example is the uniform density. Likewise, positively kurtotic densities are called *super-Gaussian*, and they are sharper than the Gaussian; an example is the two-sided exponential density.

Kurtosis minimization/maximization is one of the criteria to be used in context with our neural nonlinear PCA type learning algorithms outlined in detail in the next Section.

Neural Independent Component Analysis

Separation algorithms

In [20], one of the authors proposed several nonlinear extensions of his learning rule for computing the standard PCA subspace. These extensions can be applied to learning a separating matrix for prewhitened inputs. One of the proposed extensions is the *Nonlinear PCA rule* [11, 13, 20]:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{v}_k - \mathbf{W}_k g(\mathbf{y}_k)]g(\mathbf{y}_k^T). \tag{5}$$

We assume here that the input vectors $\mathbf{v}_k$ are obtained from the mixture vectors $\mathbf{x}_k$ by whitening them, and we denote $\mathbf{y}_k = \mathbf{W}_k^T\mathbf{v}_k$. The matrix $\mathbf{W}_k$ is the weight matrix, and μ_k is a small positive learning rate parameter. $g(t)$ is a nonlinear function and $g(\mathbf{y})$ is understood component-wise as a vector whose ith component is $g(y(i))$. In all these algorithms, the function $g(t)$ is usually chosen to be odd for stability and separation reasons; an example is $g(t) = tanh(t)$.

The algorithm (5) can be interpreted as a learning rule for a neural layer, in which $\mathbf{W}_k$ is the weight matrix and the neurons have the activation function $g(y)$. Thus the term $\mathbf{v}_k g(\mathbf{y}_k^T) = \mathbf{v}_k g(\mathbf{v}_k^T\mathbf{W}_k)$, with $\mathbf{v}_k$ the input vector to this layer, is the product of the inputs and the outputs of the layer, or a Hebbian term. We can justify that the Nonlinear PCA rule (5) converges to a separating matrix $\mathbf{W}$ by an analysis presented in the next section.

We have recently developed another learning rule, the so-called *bigradient algorithm* [27], which is applied for learning the orthonormal separating matrix $\mathbf{W}^T$ as follows:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k\mathbf{v}_k g(\mathbf{y}_k^T) + \gamma_k\mathbf{W}_k(\mathbf{I} - \mathbf{W}_k^T\mathbf{W}_k). \tag{6}$$

Here μ_k is again a small learning rate, this time positive or negative, and γ_k is another positive gain parameter, usually about 0.5 or 1 in practice. The bigradient learning rule is a stochastic gradient algorithm for maximizing (if $\mu_k > 0$) or minimizing (if $\mu_k < 0$) the criterion $\sum_{j=1}^{M} E\{f(y(j))\}$, with $g(y) = d/dyf(y)$, under the constraint that the weight matrix $\mathbf{W}$ must be orthonormal. The algorithm (6) is derived and discussed in more detail in [27]. Kurtosis can be minimized or maximized by choosing $g(y) = y^3$, although other choices are possible, too.

Both these algorithms require that the original source signals have a kurtosis with the same sign: $\mathrm{sgn}(\mathrm{cum}[s(i)^4]) = +1$ or -1 for $i = 1, \ldots, M$. In [3], Cardoso and Laheld show that this condition can be mildened for their PFS/EASI algorithm somewhat so that the sum of kurtosises must have the same sign for any two sources pairwisely. The same condition seems to hold for the neural learning algorithms (5),(6) in practice. A more extensive discussion about neural separation algorithms is given in [15].

Estimation of the basis vectors of ICA

If the goal is signal separation only, then the ICA basis vectors are not needed explicitly. However, they may be useful because they show the directions in the input space aligned with the independent components; in a way, the ICA basis vectors are generalizations of the Principal Component Analysis eigenvectors but in many cases the ICA basis vectors characterize the data better [14]. They should be useful e.g. in Exploratory Projection Pursuit [8] where one tries to project the data onto directions that reveal as much of the structure as possible. In a sense, ICA basis vectors provide such directions.

Assuming now that the matrix $\mathbf{B}_k$ has converged to a separating solution $\mathbf{B}$ (usually as a product of the whitening matrix and the separating matrix), the basis vectors of ICA can be mathematically estimated by using the theory of pseudoinverses. If $\mathbf{x}_k$ is solved directly from (3), in the general case $L > M$ there exist infinitely many possible solutions. Among them, the unique minimum-norm (pseudoinverse) solution is

$$\hat{\mathbf{x}}_k = \hat{\mathbf{A}}\mathbf{y}_k = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}\mathbf{y}_k = \sum_{j=1}^{M} \hat{s}_k(j)\hat{\mathbf{a}}(j). \tag{7}$$

Here $\hat{\mathbf{a}}(j)$ denotes the jth column of the $L \times M$ matrix $\hat{\mathbf{A}} = \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}$. Comparing this with the ICA expansion (2), it is seen that the vectors $\hat{\mathbf{a}}(j)$ are the desired estimates of the basis vectors of ICA. They can be normalized and ordered suitably.

A completely neural algorithm for estimating the basis vectors of ICA which does not require any inversion of matrices can be developed as follows [14]. Let us denote by $\mathbf{Q}$ the $L \times M$ weight matrix whose columns are the desired estimates $\hat{\mathbf{a}}(j)$, $j = 1, ..., M$ of the basis vectors of the ICA expansion (in any order). Replacing $\hat{\mathbf{A}}$ in (7) by $\mathbf{Q}$ it can be seen that for estimating $\mathbf{Q}$, two conditions must be satisfied:

1. The mean square error $E\{||\mathbf{x} - \mathbf{Q}\mathbf{y}||\}$ must be minimal, for obtaining the pseudoinverse solution;

2. The components of vector $\mathbf{y}$ must be statistically independent.

It is usually possible to satisfy both of these requirements, leading to the required ICA solution [14]. Assume that as a result of the whitening and separation stages, the matrix $\mathbf{B}_k$ has converged to a separating solution $\mathbf{B}$, and the components of $\mathbf{y}$ are as independent as possible. The second condition above is then satisfied, and it suffices to solve the first condition, i.e. to search for the matrix $\mathbf{Q}$ which minimizes the mean-square error.

Omitting the expectation, the gradient of $\| \mathbf{x} - \mathbf{Q}\mathbf{y} \|^2$ with respect to $\mathbf{Q}$ is $-2(\mathbf{x} - \mathbf{Q}\mathbf{y})\mathbf{y}^T$, which in a standard way yields the stochastic gradient algorithm

$$\mathbf{Q}_{k+1} = \mathbf{Q}_k + \mu_k(\mathbf{x}_k - \mathbf{Q}_k\mathbf{y}_k)\mathbf{y}_k^T \tag{8}$$

($\mu_k > 0$) for learning $\mathbf{Q}$. This algorithm can be used for estimating the basis vectors of ICA in context with any suitable separation algorithm. Naturally, it would be possible to use here more complicated but faster converging algorithms (for example conjugate gradient type) for minimizing the MSE error.

Mathematical analysis

The bigradient algorithm (6) has been derived in the first place for the constrained minimization/maximization of the cost function $\sum_{j=1}^{M} E\{f(\mathbf{v}^T\mathbf{w}(j))\}$, under the constraints $\mathbf{w}(i)^T\mathbf{w}(j) = \delta_{ij}$, with $g(y) = d/dyf(y)$, and so it can be expected to converge to an orthogonal matrix that will minimize (or maximize, depending on the sign of the gains μ_k) the cost function. Thus kurtosis is minimized/maximized when $f(y(j)) = y(j)^4$, when we assume that $E\{y(j)^2\} = 1$. However, the relation of the Nonlinear PCA learning rule (5) is only indirectly related to an optimization criterion [13], and so a convergence analysis should be provided. This section provides some results on the asymptotic solutions of the Nonlinear PCA learning rule.

We start from the learning rule (5)

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu_k[\mathbf{v}_k - \mathbf{W}_k g(\mathbf{y}_k)]g(\mathbf{y}_k^T) \tag{9}$$

with $\mathbf{y}_k = \mathbf{W}_k^T \mathbf{v}_k$. The input vectors $\mathbf{v}_k$ are whitened: $E\{\mathbf{v}_k \mathbf{v}_k^T\} = \mathbf{I}$, and we assume that there exists a square separating matrix $\mathbf{R}$ such that the vector $\mathbf{u}_k = \mathbf{R}^T \mathbf{v}_k$ 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. To make the analysis easier, we proceed by making a linear transformation to the learning rule (5): we multiply both sides by $\mathbf{R}^T$, with $\mathbf{R}$ an orthogonal separating matrix [23]. We obtain

$$\begin{aligned}\mathbf{R}^T\mathbf{W}_{k+1} &= \mathbf{R}^T\mathbf{W}_k + \mu_k[\mathbf{R}^T\mathbf{v}_k - \mathbf{R}^T\mathbf{W}_k g(\mathbf{W}_k^T\mathbf{v}_k)]g(\mathbf{v}_k^T\mathbf{W}_k) &(10)\\ &= \mathbf{R}^T\mathbf{W}_k + \mu_k[\mathbf{R}^T\mathbf{v}_k - \mathbf{R}^T\mathbf{W}_k g(\mathbf{W}_k^T\mathbf{R}\mathbf{R}^T\mathbf{v}_k)]g(\mathbf{v}_k^T\mathbf{R}\mathbf{R}^T\mathbf{W}_k) &(11)\end{aligned}$$

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

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \mu_k[\mathbf{u}_k - \mathbf{S}_k g(\mathbf{S}_k^T\mathbf{u}_k)]g(\mathbf{u}_k^T\mathbf{S}_k)]. \tag{12}$$

This equation has exactly the same form as the original one. Geometrically the transformation by the orthogonal matrix $\mathbf{R}$ 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. If $\mathbf{S}_k$ tends to a scaled version of the unit matrix, then in the original equation $\mathbf{W}_k = \mathbf{R}\mathbf{S}_k$ tends to a similarly scaled version of the separating matrix $\mathbf{R}$.

To show this, the difference equation (12) can be further analyzed by writing down the corresponding averaged differential equation; for a discussion of the technique, see e.g. [19]. The limit of convergence of the difference equation is among the asymptotically stable solutions of the averaged differential equation. Taking averages in (12) 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

$$d\mathbf{Z}/dt = G(\mathbf{Z}) - \mathbf{Z}H(\mathbf{Z}), \tag{13}$$

with

$$\begin{aligned}G(\mathbf{Z}) &= E\{\mathbf{u}g(\mathbf{u}^T\mathbf{Z})\}, &(14)\\ H(\mathbf{Z}) &= E\{g(\mathbf{Z}^T\mathbf{u})g(\mathbf{u}^T\mathbf{Z})\}. &(15)\end{aligned}$$

The expectations are over the (unknown) density of vector $\mathbf{u}$. We are ready to state the main result of this section, which is a simplified version of a more general theorem originally presented and proven by one of the authors in [23]:

Theorem 1. In the matrix differential equation (13), 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 mutually independent and all have the same density;
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:

$$A = E\{u^2 g'(\alpha u)\} - 2\alpha E\{g(\alpha u) g'(\alpha u) u\} - E\{g^2(\alpha u)\} < 0, \tag{16}$$

where $g'(t)$ is the derivative function of $g(t)$ and α is a scalar satisfying

$$E\{u g(\alpha u)\} = \alpha E\{g^2(\alpha u)\}. \tag{17}$$

5. The following condition holds:

$$E\{u^2\} E\{g'(\alpha u)\} - E\{g^2(\alpha u)\} < 0. \tag{18}$$

Then the matrix

$$\mathbf{Z} = \mathbf{D} = diag(\alpha, ..., \alpha) = \alpha \mathbf{I} \tag{19}$$

is an asymptotically stable stationary point of (13), where α is the positive solution to eq. (17).

The proof is given in [23].

Note 1. We only consider a diagonal matrix $\mathbf{D} = \alpha \mathbf{I}$ 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 cannot be determined from eq. (17); if $+\alpha$ is a solution, then so is also $-\alpha$. If the weight matrix $\mathbf{S}_k$ of eq. (12) 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 $\pm\alpha$. The sign has no influence on the absolute magnitude. For the negative α, a similar result to the above holds.

Note 3. Theorem 1 allows non-monotonic activation functions. However, if $g(y)$ is monotonic, then eq. (17) 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 and positive for negative α; but then there could not be any solution because $E\{g^2(\alpha u)\} > 0$.

The Theorem 1 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. For more details, see [23].

1. Polynomials.

The family of odd polynomial functions

$$g(y) = y^s, \ s = 1, 3, 5, 7, ... \tag{20}$$

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. These functions also contain the linear function for $s = 1$.

First, for α we get from eq. (17):

$$E\{u^{s+1}\} = \alpha^{s+1} E\{u^{2s}\}. \tag{21}$$

Substituting this in eq. (16) in the condition 4, we find that this condition is always satisfied.

Now, the stability condition 5 of Theorem 1 becomes

$$E\{u^{s+1}\} - sE\{u^2\}E\{u^{s-1}\} > 0. \tag{22}$$

Consider first the case

$$s = 1,\ g(u) = u. \tag{23}$$

Clearly, the condition (22) is not satisfied. *The linear function never gives asymptotic stability.* Consider next the case

$$s = 3,\ g(u) = u^3. \tag{24}$$

Now (22) gives

$$E\{u^4\} - 3(E\{u^2\})^2 > 0. \tag{25}$$

This expression is exactly the *kurtosis* or the fourth order cumulant of u (see end of the second section). 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

$$E\{u^6\} - 5E\{u^2\}E\{u^4\} > 0, \tag{26}$$

etc.

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 5 of the Theorem becomes $E\{g'(\alpha u)\} < E\{g^2(\alpha u)\}$. 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. 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.

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. (17) is not feasible and numerical methods must be used. It turns out that Condition (5) holds for $\beta > 0$ (for details, see [23]). Condition (4) is always satisfied. The conclusion is that *for the uniform density the sigmoidal function gives asymptotic stability* when $\beta > 0$.

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 a series of runs, in which the input data were 3-dimensional, each element having an identical uniform density, the value $\beta = 5.0$ was chosen for the sigmoid parameter and the initial deviation of $\mathbf{Z}(0)$ from the theoretical limit was varied. For this value of β, and the uniform densities used, $\alpha = 0.6998$. The deviation was increased up to 100.0 and the algorithm converged invariably to the asymptotically stable solution $\mathbf{D}$ predicted by Theorem 1, or a variation of $\mathbf{D}$: when the initial deviation is increased, it may happen that the asymptotic limit for $\mathbf{Z}(t)$ will not be $\mathbf{D} = diag(\alpha, \alpha, \alpha)$ but a permutation, with possibly changed signs. Thus e.g. for the initial value

$$\mathbf{Z}(0) = \begin{pmatrix} -6.1953 & 0.2556 & -0.0945 \\ 6.3493 & -2.1398 & -3.6694 \\ 0.0710 & 7.6358 & -5.7220 \end{pmatrix} \tag{27}$$

Figure 1: The ICA network. Weight matrix **V**: the whitening transformation. Weight matrix **W**: the separation transformation. Weight matrix **Q**: basis vector estimation

the asymptotic value turned out to be

$$lim\ \mathbf{Z}(t) = \begin{pmatrix} -0.6998 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.6998 \\ 0.0000 & 0.6998 & 0.0000 \end{pmatrix} \tag{28}$$

which is a permutation of matrix **D**. Note also the negative sign in two of the nonzero elements.

The overall conclusion of this section is that, while the Nonlinear PCA learning rule is not directly a gradient method for a cost function, its limits of convergence are nevertheless separating matrices, if the nonlinear activation function $g(y)$ is adapted to the original signal densities, especially the sign of the kurtosis. We proceed now to put together the various parts of our analysis to introduce a multilayer neural network for signal separation and Independent Component Analysis.

The ICA network

To recapitulate, we can achieve source signal separation from a set of mixtures by first whitening (sphering) the input vectors, and then using a separation algorithm. In addition to this, if also the ICA basis vectors are needed, they can be obtained by one of the methods given above in Subection 3.2.

Because both the whitening, the separation, and the ICA basis vector estimation can be performed adaptively in subsequent on-line processes, the first one receiving a sequence of mixture signals $\mathbf{x}_k$ as inputs, a neural network is a possible implementation for the algorithms. Consider the 3-layer network of Fig. 1, called the *Independent Component Analysis (ICA) network*. The input model and the consequent layers are explained in the following.

As the starting point we have a set of M statistically independent, zero-mean signals $s_k(i),\ i = 1, ..., M$ that are unobservable. By an unknown mixing process, a set of L observable linear mixtures $(M \leq L)\ x_k(j),\ j = 1, ..., L$ are available according to the model (1). For simplicity, let us assume that the additive noise is zero and is omitted from the sum (1). Also, assume that the amplitudes of the source signals $s_k(i)$ have been absorbed in the mixing coefficients $a(ij)$ in such a way that we can assume that the variances of all $s_k(i)$ are equal to one. Thus we have from eq. (2): $\mathbf{x}_k = \mathbf{A}\mathbf{s}_k$ with $E\{\mathbf{s}_k\mathbf{s}_k^T\} = \mathbf{I}$. These vectors $\mathbf{x}_k$ are now the input stream to the first layer of the ICA network of Fig. 1.

1. Whitening (sphering): by a linear transformation $\mathbf{V}$, the signal vectors $\mathbf{x}_k$ are transformed to new signal vectors $\mathbf{v}_k = \mathbf{V}\mathbf{x}_k$ such that $E\{\mathbf{v}_k\mathbf{v}_k^T\} = \mathbf{I}$. The elements of $\mathbf{v}_k$ have variances equal to 1 and are uncorrelated - but in general not yet independent. In the ICA network of Fig. 1, the first layer is linear and whitens the input vectors $\mathbf{x}_k$. The whitening transformation $\mathbf{V}$ is implemented by the weights of the linear layer and can be learned in a neural learning algorithm introduced by Plumbley [25]:

$$\mathbf{V}_{k+1} = \mathbf{V}_k + \mu_k(\mathbf{V}_k\mathbf{x}_k\mathbf{x}_k^T\mathbf{V}_k^T - \mathbf{I})\mathbf{V}_k. \tag{29}$$

2. Separation: by a linear transformation $\mathbf{B}$, the whitened vectors $\mathbf{v}_k$ are transformed into output signal vectors

$$\mathbf{y}_k = \mathbf{B}\mathbf{v}_k \tag{30}$$

such that the elements of $\mathbf{y}_k$ are not only uncorrelated, but statistically independent. This is done in the second layer of the ICA network of Fig. 1. For the layer weights, the Nonlinear PCA learning rule (5), or the bigradient learning rule (6) can be used, with the whitened vectors $\mathbf{v}_k$ now coming as inputs instead of the original $\mathbf{x}_k$. The nonlinear gradient function g must be chosen so that kurtosis is minimized/maximized; the hyperbolic tangent or a polynomial is a suitable function, depending on the case (see the mathematical analysis in the previous section). After a proper learning period using a sample of the mixture signals $\mathbf{x}_k$, the output of the second layer $\mathbf{y}_k = \mathbf{W}_k^T\mathbf{v}_k$ tends to a vector that approximates the original signal vector $\mathbf{s}_k$, although the order of the signals and their amplitudes may have changed. This is because $\mathbf{W}_k^T$ tends to a separating matrix $\mathbf{R}$. After learning, the network can be used to separate additional samples from the same signals, not occurring in the training set.

3. ICA basis vector estimation: the task of the last layer in the network of Fig. 1 is to estimate the basis vectors $\mathbf{a}(i)$, $i = 1, \ldots, M$, of ICA. This can be done with the neural learning algorithm (8) for the weight matrix $\mathbf{Q}$. Note that in this algorithm, we need both the independent component vectors $\mathbf{y}_k$ produced by the second layer of the ICA network, and the original input vectors $\mathbf{x}_k$. In this sense, this resembles the auto-associative MLP network in which the inputs are used as the desired outputs. It must be emphasized, however, that the inputs $\mathbf{x}_k$ are the mixture signals, so the original independent signals $s_k(i)$ are never used in training and they can be completely unknown.

Experimental results

In this section, we demonstrate the performance of the ICA network of Fig. 1 using both artificial and real-world data.

A. The artificial data by Comon. Consider first a test example used earlier by Comon [5]. Here, the original 3 source signals $s_k(1)$, $s_k(2)$, and $s_k(3)$ in (2) consist of uniformly distributed noise, a ramp signal, and a pure sinusoid. Figure 2a shows 100 samples of them. Actually two of the source signals are deterministic waveforms, allowing easy visual inspection of the separation results. All the three sources have a negative kurtosis. Fig. 2b depicts the respective components of the 3-dimensional data vectors $\mathbf{x}_k$, which are linear mixtures of

Figure 2: a) Original source signals in Comon's example. b) The mixtures, inputs to the ICA network. c) The separated outputs given by the Nonlinear PCA algorithm

the source signals. They were formed using the model (2), where the true normalized basis vectors of ICA were $\mathbf{a}(1) = [0.0891, -0.8909, 0.4454]^T$, $\mathbf{a}(2) = [0.3906, -0.6509, 0.6509]^T$, and $\mathbf{a}(3) = [-0.3408, 0.8519, -0.3976]^T$. The additive noise $\mathbf{n}_k$ was zero.

We chose the simplest learning algorithms, so that the algorithm (29) was used for whitening, the Nonlinear PCA rule (5) for separation, and the LS rule (8) for estimating the basis vectors of ICA. The 100 data vectors were used 60 times sequentially in teaching the ICA network of Fig. 1. The learning parameter μ_k was 0.01 both in (29) and (5). The learning function was $g(t) = tanh(t)$. After teaching, the data vectors $\mathbf{x}_k$, $k = 1, \ldots, 100$, were input to the network of Fig. 1. Fig. 2c shows the separated signals $y_k(1)$, $y_k(2)$, and $y_k(3)$ (outputs of the second layer), which are good estimates of the original source signals. In the last layer of the ICA network, the algorithm (8) learned a matrix $\mathbf{Q}$ whose normalized columns $\hat{\mathbf{a}}(1) = [-0.1054, 0.8917, -0.4401]^T$, $\hat{\mathbf{a}}(2) = [0.3918, -0.6541, 0.6470]^T$, and $\hat{\mathbf{a}}(3) = [0.3319, -0.8519, 0.4073]^T$ are good estimates of the theoretical basis vectors of ICA.

The results were roughly similar, when the bigradient algorithm (6) was used for estimating the separating matrix $\mathbf{W}^T$ with the same learning function and parameters, or alternatively using the learning function $g(t) = t^3$ and a negative gain parameter $\mu_k = -0.003$. The other parameter γ was 0.9. Also the PFS/EASI algorithm [3, 16] performs well with suitable choices.

B. Image data. Here we present a larger scale experiment with image data, taken from [22]. The 3 source signals were the digital images shown in Fig. 3, first row (flowers, model, waterfall). We have not tested the mutual independence of these sources in any way. All the sources except the third one have a negative kurtosis; the kurtosis of the waterfall image has a small positive value, so that the sum of pairwise kurtoses for any two sources is always negative. The size of the source images is 387 × 306; by row-wise scanning, they were coded into signal sequences with 118422 elements. Each 3-dimensional source vector $\mathbf{s}_k$ in (2) contained the kth components of the three source images. These were multiplied by a nonorthogonal full-rank 3 × 3 ICA basis matrix $\mathbf{A}$, yielding the 118422 data vectors $\mathbf{x}_k$ used in the simulation. The 3 components of $\mathbf{x}_k$, compiled back into rectangular arrays, are depicted on the second row of Fig. 3; they look rather similar, revealing little of the structure of the original source images.

Each of the 3 images on the third row of Fig. 3 contains one component of the whitened vectors $\mathbf{v}_k$, $k = 1, \ldots, 118422$. In this experiment, we used PCA whitening. The whitening matrix was computed using standard numerical software. These images already show some structure, but are still far from the original sources.

For separation, we used the Nonlinear PCA rule (5). The data vectors were used 20 times sequentially, and the gain parameter μ_k decreased slowly from its initial value 0.0005. The learning function was $g(t) = tanh(t)$. The fourth row of Fig. 3 shows the component images of the vectors $\mathbf{y}_k$, $k = 1, \ldots, 118422$. These were obtained as responses from the second layer of the ICA network after learning, when the data vectors $\mathbf{x}_k$, $k = 1, \ldots, 118422$, were used as inputs. The component images have been rescaled so that their gray level range is the same as in the original images, and in some cases their sign has been changed to opposite. The separation results are good, even though some noise is visible. This example demonstrates clearly the usefulness of nonlinearities in PCA type learning algorithms. The definitely poorer results on the third row of Fig. 3 show what standard PCA is able to achieve in this application.

A more extensive demonstration in which 6 other source images are added to the set is given in [15]. The system is able to separate the 9 images with good results.

In the simulations described above, the sources are mostly sub-Gaussian with a negative kurtosis. However, we have applied especially the bigradient algorithm for super-Gaussian sources that have positive kurtosis, too. In [28], up to 10 real speech signals were separated from their mixtures using the bigradient algorithm. The speech signals are typically super-Gaussian [1]. In these experiments, the learning functions and parameter values were chosen in much the same manner as before, but μ_k must have the opposite sign, because the sum of the fourth moments is maximized instead of minimizing it.

Finally, we emphasize that preprocessing the input data by whitening is essential for achieving good separation results using nonlinear PCA type learning algorithms. Without whitening, the algorithms are able to somehow separate sinusoidal signals [11], but usually not other signals. The obvious reason is that without whitening the algorithms still largely respond to second-order statistics in spite of using nonlinearities.

Conclusions and remarks

In this paper, we have introduced a neural network for performing Independent Component Analysis (ICA). After learning, the network has a standard multilayer feedforward structure. The basic ICA network consists of whitening, separation, and basis vector estimation layers. It can be used for both source separation and estimation of the basis vectors of ICA. We have presented several alternative learning procecdures for each layer, and modified our previous PCA algorithms to nonlinear versions so that their separation capabilities are greatly improved. The proposed ICA network yields good results in test examples.

In any of the three layers of the complete ICA network, it is possible to use either a neural or a non-neural learning method. In practice, it may be advisable to learn neurally only the critical part, source separation, because efficient standard numerical methods are available for whitening and estimation of the basis vectors of ICA.

Another remark concerns the linear ICA model (2), which is relatively simple. It would be of interest to extend the results of this paper to more general cases, where for example the data are nonstationary, or the data model is nonlinear, or contains time delays. Attempts to extend source separation and ICA into these directions have already been made by some authors [2, 7, 17, 24].

Acknowledgement. We are grateful to Mr. Liuyue Wang for the experiments on the one-dimensional signals, and to Mr. Ricardo Vigario for the experiments on the image data.

References

[1] A. Bell and T. Sejnowski, "Blind separation and blind deconvolution: an information-theoretic approach," in *Proc. 1995 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, USA, May 1995, pp. 3415-3418.

[2] G. Burel, "Blind separation of sources: a nonlinear neural algorithm," *Neural Networks*, vol. 5, pp. 937-947, 1992.

[3] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," manuscript submitted to *IEEE Trans. on Signal Processing*, October 1994.

[4] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing.* New York: John Wiley, 1993.

[5] P. Comon, "Separation of stochastic processes," in *Proc. of Workshop on Higher-Order Spectral Analysis*, Vail, Colorado, June 1989, pp. 174-179.

[6] P. Comon, "Independent component analysis - a new concept?," *Signal Processing*, vol. 36, pp. 287-314, 1994.

[7] G. Deco and W. Brauer, "Nonlinear higher-order statistical decorrelation by volume-conserving neural architectures," *Neural Networks*, vol. 8, pp. 525 - 535, 1995.

[8] J. Friedman, "Exploratory projection pursuit," *J. Amer. Statistical Assoc.*, vol. 82, no. 397, pp. 249-266, 1987.

[9] S. Haykin, *Neural Networks: A Comprehensive Foundation.* New York: IEEE Computer Society Press and Macmillan, 1994.

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

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

[12] J. Karhunen, "Optimization criteria and nonlinear PCA neural networks," in *Proc. 1994 IEEE Int. Conf. on Neural Networks*, Orlando, Florida, June 1994, pp. 1241-1246.

[13] J. Karhunen and J. Joutsensalo, "Generalizations of principal component analysis, optimization problems, and neural networks," *Neural Networks*, vol. 8, pp. 549 - 562, 1995.

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

[15] J. Karhunen, L. Wang, and R. Vigario, "Nonlinear PCA type approaches for source separation and Independent Component Analysis", in *Proc. 1995 IEEE Int. Conf. on Neural Networks*, Perth, Australia, Dec. 1995.

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

[17] K. Matsuoka, M. Ohya, and M. Kawamoto, “A neural net for blind separation of nonstationary signals,” *Neural Networks*, vol. 8, no. 3, pp. 411-419, 1995.

[18] E. Moreau and O. Macchi, “New self-adaptive algorithms for source separation based on contrast functions,” in *Proc. IEEE Signal Proc. Workshop on Higher Order Statistics*, Lake Tahoe, USA, June 1993, pp. 215-219.

[19] E. Oja, Subspace methods of pattern recognition. Letchworth: Research Studies Press, 1983.

[20] E. Oja, H. Ogawa, and J. Wangviwattana, “Learning in nonlinear constrained Hebbian networks,” in *Artificial Neural Networks (Proc. ICANN-91)*, T. Kohonen et al. (Eds.). Amsterdam: North-Holland, 1991, pp. 385-390.

[21] E. Oja, “Principal components, minor components, and linear neural networks,” *Neural Networks*, vol. 5, pp. 927-935, 1992.

[22] 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, May 1995.

[23] E. Oja, ”The Nonlinear PCA learning rule and signal separation – mathematical analysis”. Technical Report A26. August 1995, Helsinki University of Technology, Lab. of Computer and Information Science.

[24] J. Platt and F. Faggin, “Networks for the separation of sources that are superimposed and delayed,” in *Advances in Neural Processing Systems 4*, J. Moody et al. (Eds.). San Mateo, California: Morgan Kaufmann, 1991, pp. 730-737.

[25] M. Plumbley, “A Hebbian/anti-Hebbian network which optimizes information capacity by orthonormalizing the principal subspace, in Proc. IEE Conf. on Artificial Neural Networks, Brighton, UK, May 1993, pp. 86-90.

[26] J. Taylor and M. Plumbley, “Information theory and neural networks,” in *Mathematical Approaches to Neural Networks*, J. Taylor (Ed.). Amsterdam: Elsevier Science Publ., 1993, pp. 307-340.

[27] L. Wang, J. Karhunen, and E. Oja, “A bigradient optimization approach for robust PCA, MCA, and source separation,” to appear in *Proc. 1995 IEEE Int. Conf. on Neural Networks*, Perth, Australia, November 1995.

[28] 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, Dec. 1995.