Extraction of independent signal sources using a deflationary exploratory projection pursuit network with lateral inhibition

M. Girolami
C. Fyfe

Indexing terms: Projection pursuit network, Source separation

Abstract: A nonlinear self-organising neural network is proposed, which employs hierarchic linear negative feedback, and this network is applied to the blind separation of independent source signals from their mixtures. Blind separation of sources has become an important area of research, with significant contributions recently being made from both the statistical signal processing and artificial neural network research communities. A nonlinear extension of a negative feedback network is developed and it is shown that hierarchic linear feedback provides a deflation of the network residuals, which are employed in the Hebbian learning of the network. As each of the output neuron weights converge to a separating vector, then the weighted feedback will remove the contribution of the extracted source from the remaining residual mixture. It is shown that the data driven self-organisation of the proposed network using only Hebbian and anti-Hebbian learning will extract the underlying source signals from the received mixture. The results of a simulation are reported, which demonstrates the ability of the network in restoring images after degradation with noise and interfering images.

1 Introduction

The problem of multi-channel blind separation of sources (BSS) occurs in many application areas of signal processing such as speech, radar, medical instrumentation, mobile telephone communication and hearing aid devices. The problem is defined as the recovery of original unknown source signals from a sensor output when the sensor receives an unknown mixture of the source signals.

Blind separation of sources (BSS) is an underdetermined problem since the source signal statistics, the mixing, and transfer channels are unknown and closed form solutions are impossible. In this paper we shall concentrate on an artificial neural network (ANN) based solution to this problem, noting however that there are also nonneural batch and serial based algorithms [1–3].

Jutten and Herrault [4] were the first to develop a neural architecture and learning algorithm for BSS. They proposed a linear recurrent network and employed a form of nonlinear anti-Hebbian learning, which attempted to cancel out the higher order odd moments of the data. The choice of nonlinear function is critical for the quality of separation and that choice, as has been reported in subsequent literature, is *ad hoc*. Since then, a number of ANN architectures and algorithms have been developed specifically for BSS, a number of which are discussed in the review paper by Karhunen [5].

2 Blind separation of sources

Assume that there is an M-dimensional zero mean vector $\mathbf{s}(t)$ such that $\mathbf{s}(t) = [s_1(t), ..., s_M(t)]^T$, the components of which at each time instant t are mutually independent. The vector $\mathbf{s}(t)$ corresponds to M independent scalar valued source signals $s_i(t)$. We can write the multivariate probability density function (PDF) of the vector as the product of marginal independent distributions.

$$p_s(\mathbf{s}) = \prod_{i=1}^{M} p_i(s_i) \tag{1}$$

A data vector $\mathbf{x}(t) = [x_1(t), ..., x_N(t)]^T$ is observed at each time point t, such that $x(t) = \mathbf{A}\,\mathbf{s}(t)$, where $\mathbf{A}$ is a full rank $N \times M$ scalar matrix. As the components of the observed vector are no longer independent, the multivariate PDF will not satisfy the PDF product equality. In this paper we shall consider the case where $N = M$. If the components of $\mathbf{s}(t)$ are such that at most one source is normally distributed, then it is possible to extract the sources $\mathbf{s}(t)$ from the received mixtures $x(t)$ [6].

The mutual information of the observed vector is given as the Kullback divergence of the multivariate density from the density written in product form.

$$I(\mathbf{x}) = \int p_x(\mathbf{x}) \log \frac{p_x(\mathbf{x})}{\prod_{i=1}^{N} p_i(x_i)} d\mathbf{x} \tag{2}$$

The mutual information will always be positive and will only equal zero when the components are inde-

IEE Proceedings online no. 19971418
Paper first received 23rd October 1996 and in revised form 9th June 1997
The authors are with the Department of Computing and Information Systems, University of Paisley, High Street, Paisley, Scotland PA1 2BE, UK

pendent [6]. If the observed vector has a finite covariance matrix $\mathbf{C} = E\{\mathbf{xx}^T\}$ then the mutual information can be expressed as

$$I(\mathbf{x}) = J(\mathbf{x}) - \sum_{i=1}^{N} J_i(x_i) + \frac{1}{2}\log\frac{\prod_{i=1}^{N}\mathbf{C}_{ii}}{\det(\mathbf{C})} \quad (3)$$

Where $J(\mathbf{x}) = \int p_x(\mathbf{x}) \log (p_x(\mathbf{x})/p_G(\mathbf{x}))\, d\mathbf{x}$ and $J_i(x_i) = \int p_i(x_i) \log (p_i(x_i)/p_G(x_i))\, dx_i$ are the multivariate and marginal negentropies which are defined as the Kullback divergences of the actual PDF from a Gaussian with the same mean and variance, [6].

If we use a spatial whitening transformation to remove the second order redundancy in the data, such that $\mathbf{z} = \mathbf{Ux}$ and $\mathbf{C}_{zz} = E\{\mathbf{xx}^T\} = \mathbf{I}$, then $\det(\mathbf{C}_{zz}) = \Pi_{i=1}^{N}\, \mathbf{C}_{zz(ii)} = \det(\mathbf{I}) = 1$. The mutual information of the spatially white data is now reduced to

$$I(\mathbf{z}) = J(\mathbf{z}) - \sum_{i=1}^{N} J_i(z_i) \quad (4)$$

A further transformation, say $\mathbf{y} = \mathbf{Wz}$, is now required to reduce the remaining redundancy within the vector attributed to the non-Gaussian characteristics of the data. A truncated Edgeworth expansion [7] of the data PDF written in terms of nth order cumulants and Hermite polynomials, denoted by κ_n and h_n, is as follows

$$\begin{aligned} p_y(y) = p_G(y)\Big(1 &+ \frac{1}{3!}\kappa_3 h_3(y) + \frac{1}{4!}\kappa_4 h_4(y) \\ &+ \frac{10}{6!}\kappa_3^2 h_6(y) + \frac{1}{5!}\kappa_5 h_5(y) \\ &+ \frac{35}{7!}\kappa_3\kappa_4 h_7(y) + \frac{280}{9!}\kappa_3^3 h_9(y) \\ &+ \frac{1}{6!}\kappa_6 h_6(y) + \frac{56}{8!}\kappa_3\kappa_5 h_8(y) \\ &+ \frac{35}{8!}\kappa_4^2 h_8(y) + \frac{2100}{10!}\kappa_3^2\kappa_4 h_{10}(y) \\ &+ \frac{15400}{12!}\kappa_3^4 h_{12}(y) + \cdots\Big) \end{aligned}$$

The validity of a truncated series expansion approximation for a PDF is discussed in [7]. Expansion terms higher than fourth order can give rise to excessive fluctuations at the tails of the distribution leading potentially to negative values. By truncating the expansion at fourth order and substituting into the expression for marginal negentropy, it is shown [6] that

$$J_i(y_i) \cong \frac{1}{12}\kappa_3^2 + \frac{1}{48}\kappa_4^2 + \frac{7}{48}\kappa_3^4 - \frac{1}{8}\kappa_3^2\kappa_4 \quad (5)$$

If we make the assumption that the PDFs of the signals under consideration are approximately symmetric, then the third order cumulants will have a negligible contribution to eqn. 5. The mutual information of the transformed data is now approximated by

$$I(\mathbf{y}) \cong J(\mathbf{y}) - \frac{1}{48}\sum_{i=1}^{N}\kappa_4^2(i) \quad (6)$$

Owing to the invariance of differential entropy under orthogonal transformations we can see that

$$I(\mathbf{y}) \cong J(\mathbf{z}) - \frac{1}{48}\sum_{i=1}^{N}\kappa_4^2(i)$$

Under an orthogonal transformation, the mutual information of the data can be approximately minimised by maximising the sum of squares of fourth order cumulants. Therefore, the following contrast function is proposed

$$\Phi_{Max} = \sum_{i=1}^{N}\kappa_4^2(i) \quad (7)$$

The link between BSS and Exploratory Projection Pursuit (EPP) is made here. EPP is a statistical tool, which allows structure in high dimensional data to be identified [8]. This is achieved by projecting the data on to a low dimensional subspace and searching for structure in the projection. By defining indices which give a measure of how 'interesting' a given projection is, projection of the data onto a subspace that maximises the given index will then provide a maximally 'interesting' direction. Departures from a Gaussian distribution are viewed as 'interesting', as skewed or multi-modal distributions present certain higher order structures within the data. If we then use an index, which is a function of the direction of projection, index maximisation will provide a direction furthest from Gaussian. So in the case of BSS, if we use an index such as eqn. 7, we will find a direction which will yield distributions that are as independent as possible.

An input variable $\mathbf{x}$ is transformed by an invertible matrix $\mathbf{U}$ into $\mathbf{z}$, having diagonal covariance matrix $\mathbf{D}_1$. Then

$$\mathbf{C}_{zz} = \mathbf{U}\mathbf{C}_{xx}\mathbf{U}^T = \mathbf{D}_1 \quad (8)$$

and all subsequent linear transformations resulting in a diagonal identity covariance matrix are defined as

$$\mathbf{PDWD}_1^{-1/2}\mathbf{U} \quad (9)$$

where $\mathbf{P}$ is a permutation, $\mathbf{D}$ is an invertible diagonal scaling and $\mathbf{W}$ is an orthogonal rotation matrix [9]. The term $\mathbf{D}_1^{-1/2}\mathbf{U}$ diagonalises the covariance to an identity and spatially whitens the input. We can then write the BSS transformation as a matrix $\mathbf{W}$ such that

$$\Phi(\mathbf{W})_{Max} = \sum_{i=1}^{N}\kappa_4^2(i) \quad \text{and} \quad \mathbf{WW}^T = \mathbf{I}$$

$$\mathbf{y} = \mathbf{WD}_1^{-1/2}\mathbf{Ux} = \mathbf{WUx} \quad \text{if} \quad \mathbf{D}_1 = \mathbf{I} \quad (10)$$

Cayley parameterisation of the rotation matrix and gradient descent is employed in [9] in maximising eqn. 4; standard principal components analysis (PCA) is used as part of the whitening process.

This is essentially a batch-processing algorithm, as all available data is required to compute the sample cumulants, however self-organising neural networks can be useful for online adaptive BSS. We now consider a self-organising neural network, which will perform the BSS transformation in eqn. 10 in an online adaptive manner.

3 Deflationary exploratory projection pursuit network with lateral inhibition

Consider the neural network model shown in Fig. 1. This is an extension of the EPP network originally proposed by Fyfe and Baddeley [10]. Girolami and Fyfe [11–14] have further developed the EPP network to perform a generalised BSS transformation. The input to the network at time t is $\mathbf{x}(t) = \mathbf{As}(t)$, which is the unknown mixture of the unknown sources. The output of the first layer is given as

$$\mathbf{z} = [\mathbf{I} + \mathbf{U}]\mathbf{x} \equiv \mathbf{U}_I\mathbf{x}$$

The adaptation of the weights is driven by simple normalised anti-Hebbian learning

$$\Delta \mathbf{U} = \alpha(\mathbf{I} - \mathbf{z}\mathbf{z}^T) \quad (11)$$

The learning rate is denoted by α. $E\{\Delta\mathbf{U}\} = \alpha(\mathbf{I} - \mathbf{C}_{zz}) \to 0 \Rightarrow \mathbf{C}_{zz} \to \mathbf{I}$, where $\mathbf{C}_{zz}$ is the covariance matrix of $\mathbf{z}$. The anti-Hebbian learning of the input layer yields a spatially white output vector $\mathbf{z}$.

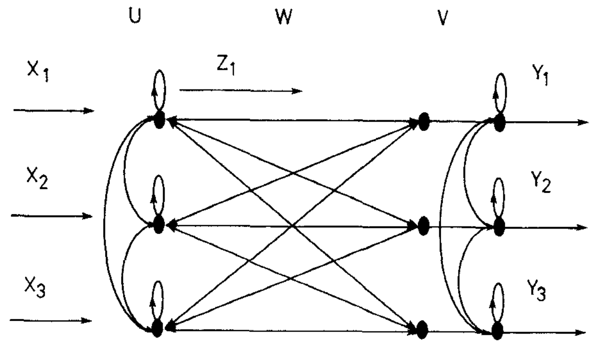

Fig. 1 *The extended exploratory projection pursuit network*

The $\mathbf{z}$ values are fed forward through the $\mathbf{W}$ weights to the output neurons where there is a second layer of lateral weights. However, before the activation is passed through this layer it is passed back to the originating $\mathbf{z}$ values in a hierarchical manner, as inhibition and then a nonlinear function of the inputs is calculated. The linear activation at the output neurons is

$$\mathbf{y} = \mathbf{W}^T\mathbf{z} = \mathbf{W}^T\mathbf{U}_I\mathbf{x} \quad (12)$$

The output neurons are nonlinear and a nonlinear functional is applied to the weighted sum. We shall discuss the form of the nonlinearity shortly. The outputs also have lateral and self-connections to the nonlinear output neurons and so the nonlinear output is defined as

$$\mathbf{y}_{nl} = (\mathbf{I} + \mathbf{V}) \equiv \mathbf{V}_I f(\mathbf{y}) \quad (13)$$

Simple hierarchical Hebbian learning is used to update the feedforward weight values, which in vector notation is

$$\Delta\mathbf{W} = \beta\left(\mathbf{z}\mathbf{y}_{nl}^T - \mathbf{W} \times \text{upper}[\mathbf{W}^T\mathbf{z} \times \mathbf{y}_{nl}]\right) \quad (14)$$

The upper [...] operator sets the matrix argument upper triangular. Similarly, anti-Hebbian learning is applied at the output layer as at the input layer

$$\Delta\mathbf{V} = \gamma(\mathbf{I} - \mathbf{y}_{nl}\mathbf{y}_{nl}^T) \quad (15)$$

The weight update equation (eqn. 14) is originally derived from the maximisation of an objective function under the constraints of orthonormality [15]. As eqn. 14 provides an orthogonal rotation to maximise the objective function $J(\mathbf{W}) = \mathbf{1}^T E\{g(\mathbf{W}^T\mathbf{z})\}$, with a suitable choice of the function $g(...)$ it can be made equivalent to the sum of squares of fourth order marginal cumulants. Note that maximising the sum of the absolute values of fourth order cumulants $\Sigma_{i=1}^N |\kappa_4(i)|$ will yield identical optimal weight parameters for function maximisation as the computationally more complex sum of squared values $\Sigma_{i=1}^N \kappa_4^2(i)$. The fourth order cumulant of a variable with zero mean and unit variance is equal to

$$\kappa_4(i) = E\{y_i^4\} - 3 \quad (16)$$

In this case, the fourth order cumulant is identical to the kurtosis of the data. Setting $g(y) = |y^4 - 3|$ will then yield for the objective function

$$\begin{aligned} J(\mathbf{W}) &= \mathbf{1}^T E\{g(\mathbf{W}^T\mathbf{x})\} \\ &= \sum_{i=1}^N E\{g(\mathbf{w}_i^T\mathbf{x})\} \\ &= \sum_{i=1}^N |\kappa_4(\mathbf{w}_i^T\mathbf{x})| \\ &= \sum_{i=1}^N |\kappa_4(y_i)| \end{aligned}$$

where $\mathbf{w}_i$ is the ith column of the weight matrix $\mathbf{W}$. Taking the instantaneous gradient and using compact matrix notation we have

$$\begin{aligned} \frac{dJ(\mathbf{W})}{d\mathbf{W}} &\propto \text{signum}(\kappa_4)\mathbf{x}(\mathbf{x}^T\mathbf{W})^3 \\ &= \text{signum}(\kappa_4)\mathbf{x}(\mathbf{y}^3)^T \end{aligned}$$

The signum operator returns +1 for arguments greater than zero and −1 for argument s less than zero. The term signum(κ_4) is the diagonal matrix whose elements are the sign of the kurtosis value of all the $\mathbf{w}_i^T\mathbf{x}$ terms, so $f(y) = \text{signum}(\kappa_4(y))y^3$. This indicates that if there are independent sources in the mixed vector $\mathbf{x}$ which have negative kurtosis the nonlinear function to be used for maximisation of the sum of squares of fourth order cumulants is $f(y) = -y^3$, and for positively kurtotic sources $f(y) = +y^3$. Simple cubic activation functions give algorithmic instability in eqn. 14 as discussed in [15]. We have proposed the following nonlinearity in [16, 17]

$$f(y) = y \pm \tanh(y) = y - \text{signum}(\kappa_4)\tanh(y)$$

Taking a Taylor expansion of the hyperbolic tangent it is clear we are attempting to maximise

$$E\left\{\int y - \text{signum}(\kappa_4)\tanh(y)dy\right\}$$

$$\cong E\left\{\int\left(y - \text{signum}(\kappa_4)\left(y - \frac{y^3}{3} + \frac{2y^5}{15} - \cdots\right)\right)dy\right\}$$

For zero mean data with unit variance $E\{y\} = 0$, $E\{y^2\} = 1$ the form of $y + \tanh(y)$ will have $E\{-y^4/12\}$ as the dominant term, as the inequality $y^4/12 > 2y^6/90$ is valid [10]. The weight update (eqn. 14) will then approximately minimise the kurtosis at each output. For the form of $y - \tanh(y)$, the $E\{y^4/12\}$ term will dominate in the objective function. This allows approximate maximisation of the kurtosis for super-Gaussian data. The additional advantage of the function $y - \tanh(y)$ is that it is an approximately linearly increasing function which is numerically stable in the stochastic algorithms in eqn. 14. *A priori* knowledge of the source signal statistics has been required to choose the sign for the compound nonlinearity

$$f(y) = y - \text{signum}(\kappa_4)\tanh(y)$$

With this knowledge mixtures of both sub- and super-Gaussian signals have been separated [11]. In keeping with the blind form of the proposed separation, we can estimate online the kurtosis of each neuron using a simple moving average estimator

$$\hat{m}_p[y_i(t+1)] = [1 - \eta(t)]\hat{m}_p[y_i(t)] + \eta(t)y_i^p(t) \quad (17)$$

$$\hat{k}_4[y_i(t+1)] = \frac{\hat{m}_4[y_i(t)]}{\hat{m}_2^2[y_i(t)]} - 3 \quad (18)$$

Eqn. 17 estimates online the pth order moments of the data, $\eta(t)$ being a small learning constant. Eqn. 18 is an estimate of the kurtosis for zero mean, unit variance data. The network nonlinearity will then be

$$f[y_i(t)] = y_i(t) - \text{signum}(\hat{k}_4[y_i(t)])\tanh[y_i(t)] \quad (19)$$

For the whitened data, the output matrix **V** will be diagonal [12, 13] and the individual elements of the weight update matrix (eqn. 14) are given by

$$\Delta w_{ji} = \eta_t(1 + v_{jj}) f\left(\sum_{k=1}^{N} w_{ik} z_k(t)\right) \times \left\{ z_j(t) - \sum_{l=1}^{i} w_{lj} \sum_{p=1}^{N} w_{ip} z_p(t) \right\} \quad (20)$$

Consider the rightmost term in eqn. 20

$$\left\{ z_j(t) - \sum_{l=1}^{i} w_{lj} \sum_{p=1}^{N} w_{ip} z_p(t) \right\} = z_j(t) - \sum_{l=1}^{i} w_{lj} y_l = z_j(t) - \sum_{l=1}^{i} w_{lj} \hat{s}_l$$

We can see that the linear network activation, which can be considered as an approximation to each independent source, is fed back in a hierarchic manner. The residual that is formed is used in the simple nonlinear Hebbian learning (eqn. 20). As each linear output converges to an approximation of one of the underlying sources, the subsequent neurons in the hierarchy will have their weights updated using residuals which have had weighted approximations of the previously extracted sources removed.

The anti-Hebbian learning at the output requires some consideration. Once the input data has been whitened, the lateral output connection weights will tend to zero [11, 12] as there are no remaining second order correlations driving the anti-Hebbian learning. However, the self-connections have a self-reinforcing effect, especially in the case where the nonlinearity $f(y) = y - \tanh(y)$ is used. Small values of y will cause $f(y) \to 0$, the self connecting weights will be self-reinforcing as $\Delta v_{ii} \cong y$, and so the Hebbian learning of eqn. 20 will continue to update the feedforward weights. This form of self-reinforcement was originally proposed by Cichocki *et al.* [18] for separation of ill-conditioned mixtures of sources. The network of Fig. 1 therefore provides a flexible online adaptive method of performing the BSS transformation given in eqn. 10.

4 Simulation

Ten greyscale images were chosen based on their kurtosis values. Each image was 202 × 202 pixels in size, with each pixel using an eight-bit greyscale representation. Fig. 2 shows the original images and Fig. 3 shows the probability density of each image as a histogram. Table 1 gives a listing of both the original and mixed data statistics (Note the negligible value of the third order cumulant of each image). From Table 1 and Fig. 3 it is clear that the majority of the images are

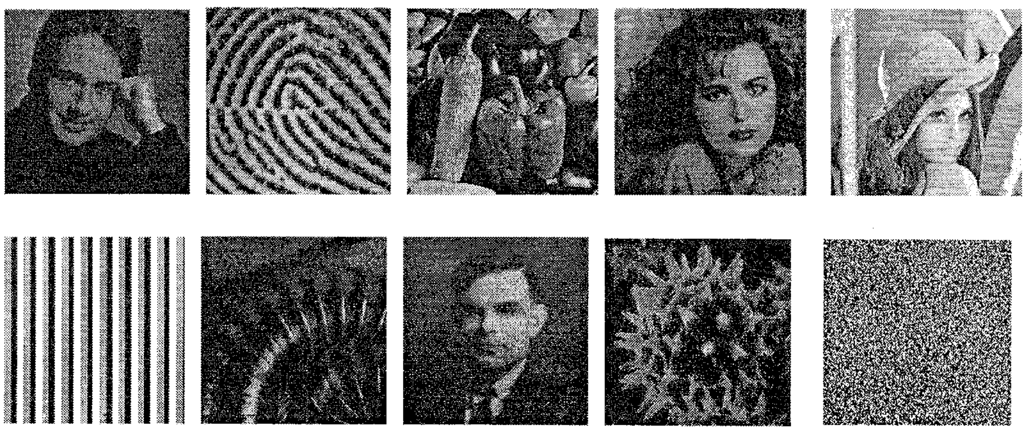

Fig. 2 *Original source images*
Top (left to right): Mulder, Finger, Peppers, Scully, Lena
Bottom (left to right): Stripe, Worm, Turing, Virus, Noise

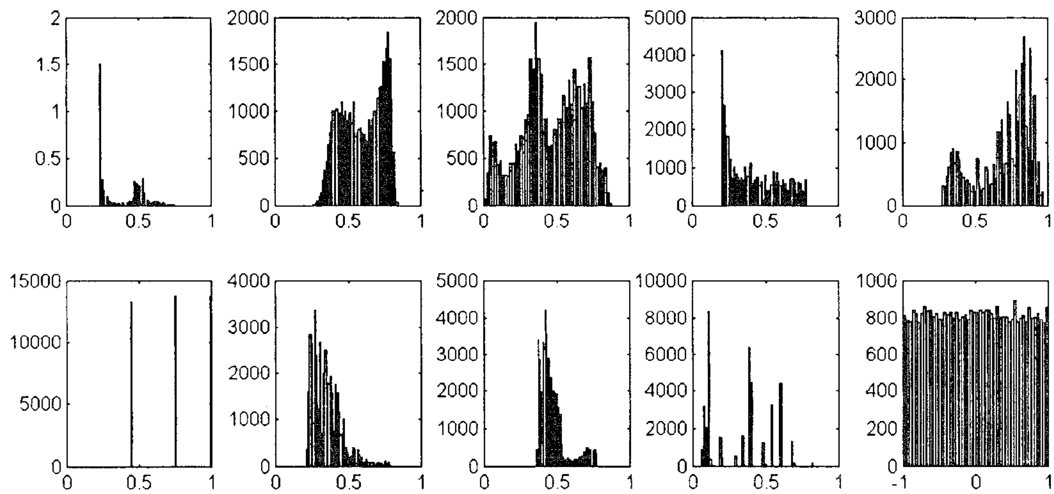

Fig. 3 *Histogram distributions for original source images*
Top (left to right): Mulder, Finger, Peppers, Scully, Lena
Bottom (left to right): Stripe, Worm, Turing, Virus, Noise

negatively kurtotic, although the images of 'Turing' and 'Worm' both have positive values of kurtosis.

Table 1: Original and mixed image statistics

Image	Mean	Kurtosis	Skew	Mixed Images	Kurtosis
Mulder	0.370	−1.245	0.0014	Mix 1	−0.0806
Turing	0.470	+1.467	0.0016	Mix 2	−0.3562
Lena	0.690	−0.730	−0.0049	Mix 3	−0.2652
Noise	0.000	−1.193	0.00063	Mix 4	−0.3479
Finger	0.590	−1.285	−0.00056	Mix 5	−0.2688
Peppers	0.460	−0.878	−0.0023	Mix 6	−0.3458
Scully	0.420	−1.144	0.0023	Mix 7	−0.3999
Stripe	0.730	−1.484	−0.0018	Mix 8	−0.3939
Worm	0.350	+2.031	0.0015	Mix 9	−0.3319
Virus	0.320	−1.279	0.0014	Mix 10	−0.3211

A randomly generated 10 × 10 matrix carried out the mixing. Figs. 4 and 5 show the mixed images and the corresponding histograms. As is apparent from Table 1, the values of kurtosis have all tended to small values, and the corresponding histograms show quite clearly the onset of central limit effects with each mixed image having almost Gaussian statistics. From Fig. 4 we can see that the mixing has caused significant degradation of the images, making them indistinguishable. Each mixture has the mean value removed prior to presenting the data to the network. The input weights were initially set to zero, while the feedforward weights were randomly set to values between −1 and 1. We use the sum of squares of fourth order marginal cumulants as a measure (or contrast) of the overall separating performance of the network. Fig. 6 shows the development of the contrast at the end of each adaptation epoch.

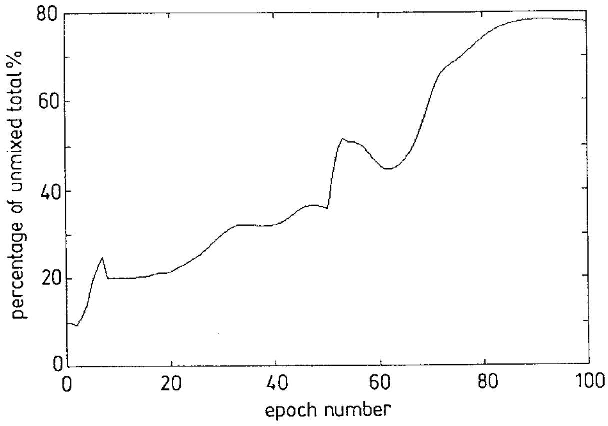

Fig. 6 *Contrast development during learning*

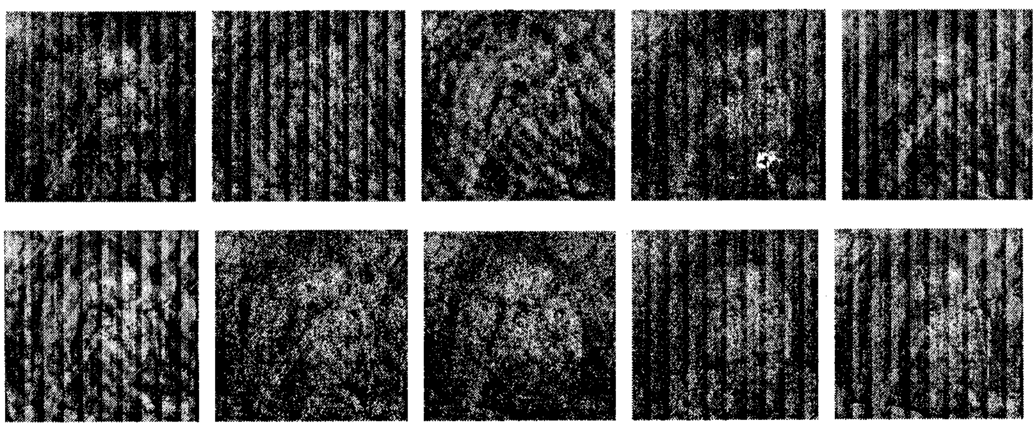

Fig. 4 *Mixed images*
Top (left to right): Mix 1 – Mix 5
Bottom (left to right): Mix 6 – Mix 10

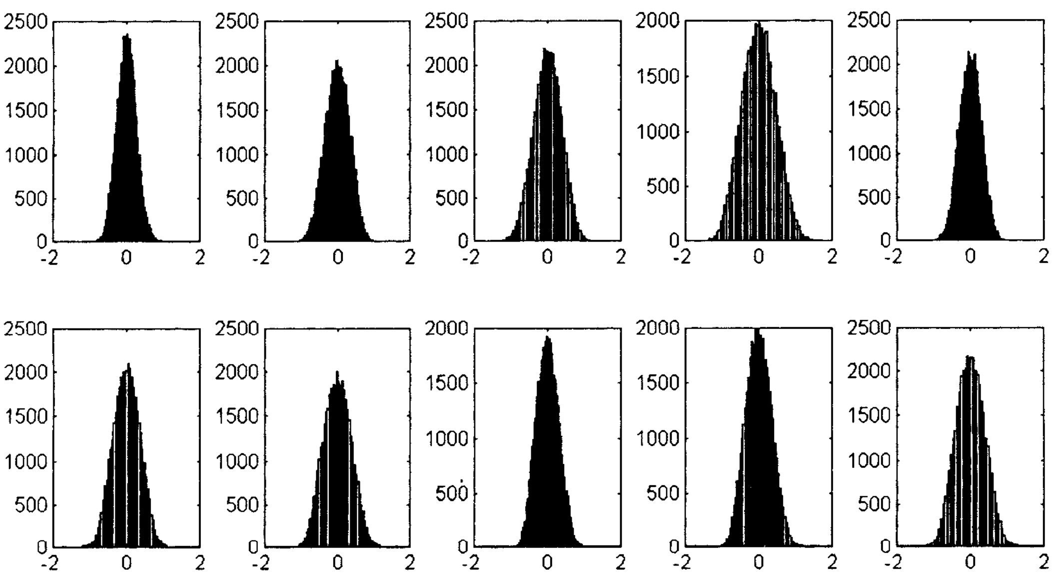

Fig. 5 *Histograms associated with mixed images*
Top (left to right): Mix 1 – Mix 5
Bottom (left to right): Mix 6 – Mix 10

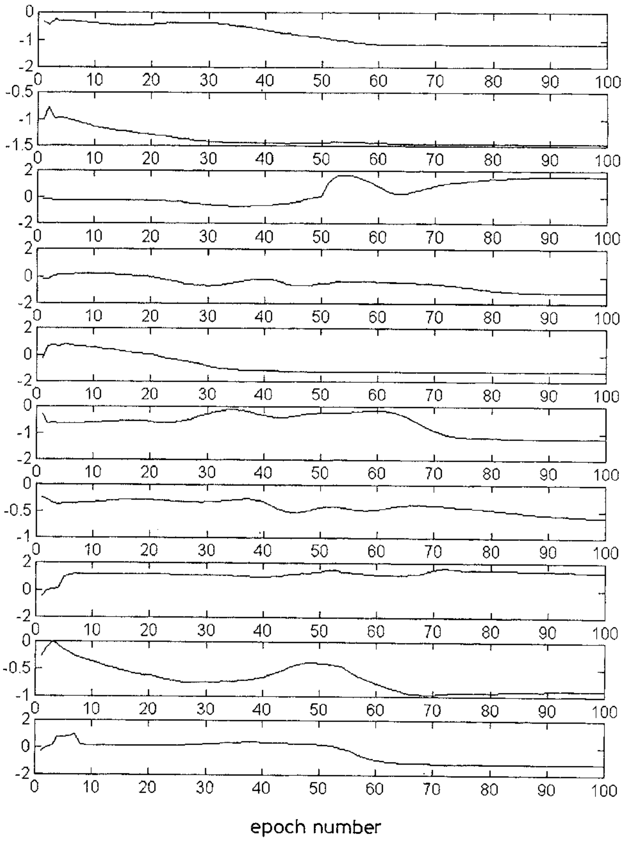

Fig. 7 *Development of the data kurtosis at each output neuron*
It is clear that each output converges to a maximally kurtotic distribution in a phased though not necessarily sequential manner

The overall contrast reaches a maximal value of 81% of the original value. One aspect of the learning that is of particular interest is the kurtosis sign development. Inspection of the sign of the kurtosis at each adaptation epoch shows the switching from positive to negative at certain outputs until stability is reached. If we consider the kurtosis development of each individual output neuron, it is clear that a phased extraction of the sources is occurring, (Fig. 7).

The manner of the extraction is dependent on the correct development of the kurtosis sign at each output. If the kurtosis sign for a particular neuron is incorrect at some point of the learning, then the neuron will pursue an inappropriate direction for maximisation. However, as the weight matrix is strictly orthonormal during the learning period, the linear independence of the weight matrix columns will force a change of the kurtosis sign. This changes the direction of pursuit and restores the strict orthonormal nature of the weight matrix. Fig. 8 shows the sign of the nonlinear term at each neuron and how these have changed during learning. Figs. 9 and 10 show the extracted sources; there is still some residual degradation indicated by the contrast value of 81%.

5 Conclusions

By applying phased linear feedback in the extended EPP network, we have extracted source images from a received mixture. Using the proposed form of eqn. 19 and online kurtosis estimation we have been able, with

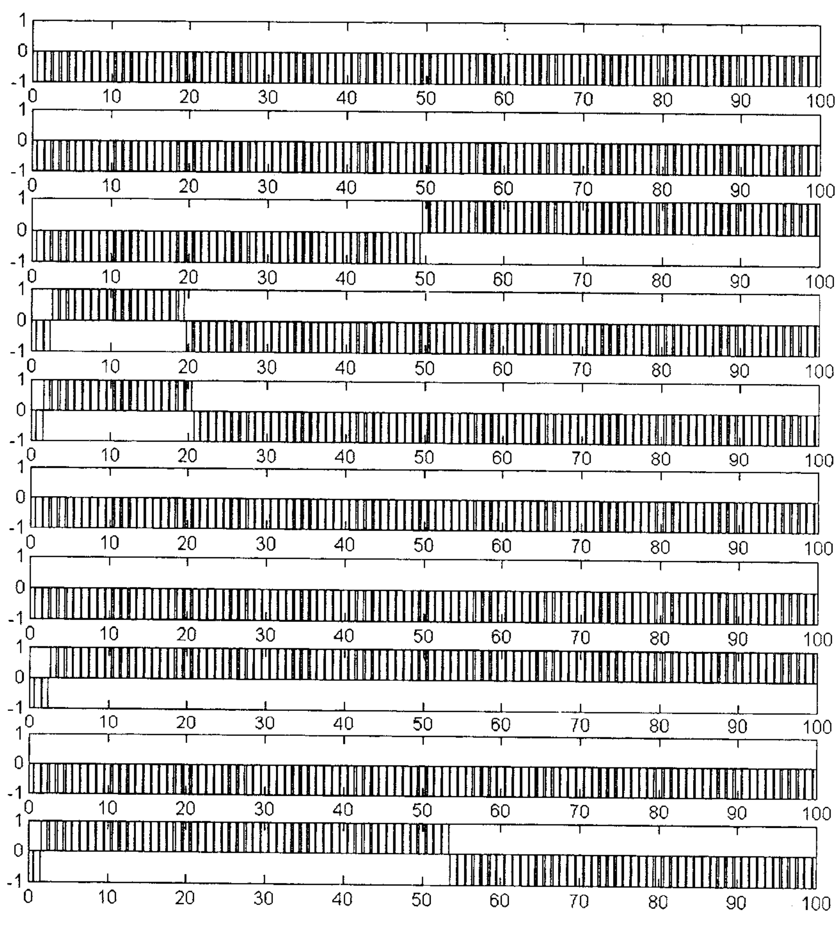

Fig. 8 *Development of the kurtosis sign at each output neuron*
The rows signify the nonlinearity sign (κ_4) term at each one of the output neurons. The changes of sign during learning can be seen at neurons 3, 4, 5, 8 and 10

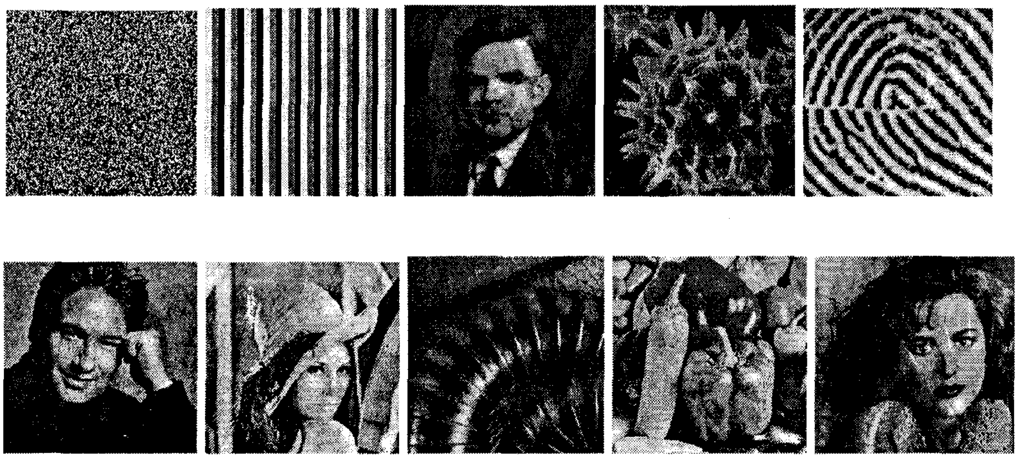

Fig. 9 *Network output: final recovered images*
Top (left to right): Noise, Stripe, Turing, Virus, Finger
Bottom (left to right): Mulder, Lena, Worm, Peppers, Scully

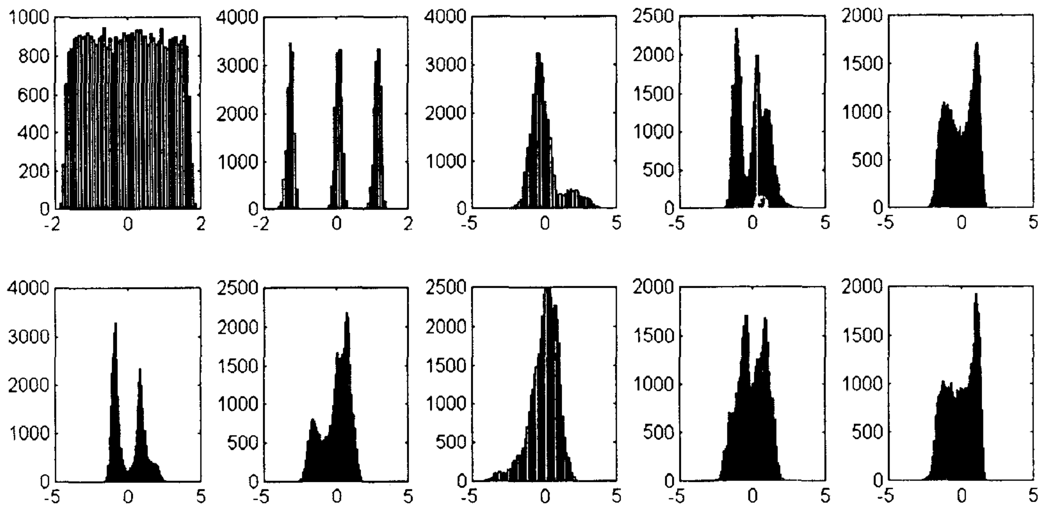

Fig. 10 *Network output: histograms*
Top (left to right): Output.1 – Output.5
Bottom (left to right): Output.6 – Output.10

no *a priori* knowledge of the source statistics, to extract images which have both sub- and super-Gaussian distributions. We have proposed the use of hierarchic linear feedback in an extension of an EPP network. The orthonormal nature of the network learning and the linear feedback of activation allow a deflationary approach to BSS. We have extended the network nonlinearity to a self-adaptive model whose adaptation is driven by the kurtosis of the output neurons. The requirement for source signals to have the same sign of kurtosis and *a priori* knowledge of the kurtosis sign has been removed.

This paper has studied the BSS transformation for the case where the number of observations equals the number of sources. If the number of observations is greater than the number of sources, then this particular technique may be applied in extracting the original sources. There will, however, be a large degree of redundancy with certain sources possibly being extracted several times at differing outputs. This situation has been studied by Cichocki *et al.* [19]. The case where the number of observations is less than the number of sources is a more complex and demanding problem. A most promising technique has been recently proposed by Pajunen [20], however this is only valid for binary valued data and cannot be extended to continuous valued data. The separation of observed mixtures of continuous valued data sources which have smaller dimension than the source data is a demanding research problem which is the subject of further research work.

6 Acknowledgments

M.A. Girolami is supported by a grant from NCR (Ltd.) Technology Development Division. The authors are grateful to the reviewers for their detailed and useful comments.

7 References

1 DELFOSSE, N., and LOUBATON, P.: 'Adaptive blind separation of independent sources: A deflation approach', *Signal Process.*, 1995, **45**, pp. 59–83
2 CARDOSO, J.F.: 'Source separation using higher-order moments'. Proc IEEE ICASSP, 1989, pp. 2109–2112
3 CARDOSO, J.F.: 'Eigen-structure of the fourth-order cumulant tensor with applications to the blind source separation problem'. Proc IEEE ICASSP, 1990, pp. 2655–2658
4 JUTTEN, C., and HERAULT, J.: 'Blind separation of sources, Part 1: An adaptive algorithm based on neuromimetic architecture', *Signal Process.*, 1991, **24**, pp. 1–10
5 KARHUNEN, J.: 'Neural approaches to independent component analysis and source separation'. Proc. ESANN'96, 1996, pp. 249–266
6 COMON, P.: 'Independent component analysis: A new concept?', *Signal Process.*, 1994, **36**, pp. 287–314

7 STUART, A., and ORD, J.K.: 'Kendall's advanced theory of statistics, Vol. 1: Distribution theory' (Wiley, New York, 1987, 5th edn.)
8 RIPLEY, B.: 'Pattern recognition and neural networks' (C U P, 1996, 1st edn.)
9 DECO, G., and OBRADOVIC, D.: 'An information-theoretic approach to neural computing' (Springer–Verlag, New York, 1996, 1st edn.)
10 FYFE, C., and BADDELEY, R.: 'Non-linear data structure extraction using simple Hebbian networks', *Biol. Cybern.*, 1995, **72**, (6), pp. 533–541
11 GIROLAMI, M., and FYFE, C.: 'Kurtosis extrema and identification of independent components: A neural network approach'. Proceedings of IEEE conference on *Acoustics, speech and signal processing*, 1997, Vol. 4, pp. 3329–3333
12 GIROLAMI, M., and FYFE, C.: 'Higher order cumulant maximisation using non-linear Hebbian and anti-Hebbian learning for adaptive blind separation of source signals' *Proceedings of IWSIP-96* (Elsevier, 1996), pp. 141–144
13 GIROLAMI, M., and FYFE, C.: 'Multivariate density factorisation for independent component analysis: An unsupervised artificial neural network approach'. Proceedings of third international workshop on *Artificial intelligence and statistics*, 1997, pp. 223–230
14 GIROLAMI, M., and FYFE, C.: 'Blind separation of sources using exploratory projection pursuit networks'. Proceedings of international conference on the *Engineering applications of neural networks*, 1996, pp. 249–252
15 KARHUNEN, J., and JOUTENSALO, J.: 'Generalisations of principal components analysis, optimisation problems, and neural networks', *Neural Netw.*, 1995, **8**, (4), pp. 549–562
16 GIROLAMI, M., and FYFE, C.: 'Stochastic ICA contrast maximisation using Oja's non-linear PCA algorithm', *Int. J. Neural Syst.*, (to be published)
17 GIROLAMI, M., and FYFE, C.: 'Fahlman type activation functions applied to non-linear PCA networks provide a generalised independent component analysis'. Proceedings of international conference on *Artificial neural networks and genetic algorithms*, 1997 (Springer, to be published)
18 CICHOCKI, A., UNBEHAUEN, R., and RUMMERT, E.: 'Robust learning algorithm for blind separation of signals', *Electron. Lett.*, 1994, **30**, (17), pp. 1386–1387
19 CICHOCKI, A., KASPRZAK, W., and AMARI, S.: 'Neural network approach to blind separation and enhancement of images', *Signal Process.*, 1996, **1**, pp. 579–582
20 PAJUNEN, P.: 'A competitive learning algorithm for separating binary sources'. Proc. ESANN'97, 1997, pp. 255–260