MLP-BASED SOURCE SEPARATION FOR MLP-LIKE NONLINEAR MIXTURES

R. Martín-Clemente[◇], *S. Hornillo-Mellado*[◇], *J. I. Acha*[◇], *F. Rojas*[†], *C. G. Puntonet*[†]

[◇]Área de Teoría de la Señal y Comunicaciones, [†]Dpto. de Arquit. y Tecnol. de Computadores
Universities of Sevilla[◇] and Granada[†] — (SPAIN)
E-mails: {ruben,susanah,acha}@us.es, {frojas,carlos}@atc.ugr.es

ABSTRACT

In this paper, the nonlinear blind source separation problem is addressed by using a multilayer perceptron (MLP) as separating system, which is justified in the universal approximation property of MLP networks. An adaptive learning algorithm for a perceptron with two hidden-layers is presented. The algorithm minimizes the mutual information between the outputs of the MLP. The performance of the proposed method is illustrated by some experiments.

1. INTRODUCTION.

Blind Source Separation (BSS) is a fundamental problem in signal processing. It consists of retrieving unobserved sources $s_1(t)$, ..., $s_N(t)$, assumed to be statistically independent (which is phisically plausible when the sources have different origins), from only M observed signals $x_1(t)$, ..., $x_M(t)$ which are unknown functions or mixtures of the sources. In general, samples of each source are not assumed to be independent and identically distributed (i.i.d) and no assumption concerning the temporal dependence between them is used. In this paper, we restrict the study to the case $N = M$, where the number of sources is equal to the number of sensors.

Starting from the seminal work [12, 13], this problem has been intensively studied over the last decade and there exist elegant solutions when the mixtures are linear and instantaneous (see [4, 6, 9] and the references therein). If the mixture is *nonlinear*, on the contrary, few algorithms have been presented and they are not completely effective.

In this paper, the latter problem is addressed by using a multilayer perceptron (MLP) as separating system, which is justified in the universal approximation property of MLP networks [7]. An adaptive learning algorithm for minimizing the mutual information between the outputs of a perceptron with two hidden-layers is presented. The performance of the proposed method is then illustrated by some experiments. The paper also introduces the problems of using MLPs with more-than-one hidden layer in the context of BSS.

1.1. Nonlinear BSS: a short review

The problem of nonlinear mixtures is that separation is *impossible* without additional prior knowledge of the mixing model, as the independence assumption is not strong enough [10, 19]. In practice, *special nonlinear mixing models are assumed in order to simplify the problem*: for example, consider Wiener and Hammerstein models (see [4], chapter 12 and [20]) and the post-nonlinear mixture case [19]. In addition, Deco and Brauer [5] have addressed the problem by considering that the mixing mapping satisfies a volume conserving condition, which ensures that it is invertible. Hyvärinen and Pajunen [10] have shown that, in the two-source case, separation is feasible if the mixing function is a conformal mapping.

Several algorithms and methods show promise in the nonlinear BSS problem. To our knowledge, the first solution was given by Burel [2], who proposed a neural network to minimize the energy of the difference between the joint probability density function (*pdf*) and the marginal *pdfs* of the estimated sources. Almeida [1] and Koutras *et al* [16] have addressed the problem using a nework with an adaptable nonlinearity as separating system. They claim that it shows a great flexibility towards fitting complex nonlinear mixing functions. Radial Basis Functions have also been employed as separating system: specifically, good results have been reported by Tan *et al* [21].

Locally linear BSS methods have been recently explored by Karhunen *et al* [15] using a K-means-clustering-based method. Pajunen *et al* (see [9], Chapter 17 and the references therein) use Kohonen's self-organizing-feature maps (SOFM). Their approach holds when the sources have *pdfs* with bounded supports. See also [14] and the references therein.

One of the greatest problems encountered in nonlinear source separation is that algorithms that are based on a gradient-descent adaptation are often trapped within local minima. For this reason, Puntonet *et al* [17] use simulated annealing to avoid undesired minima in the training of a modified Kohonen's network. In addition, Rojas *et al* [18] propose a separating system which approximates the nonlinearities of the post-nonlinear mixture model by means of odd polynomials and makes use of genetic algorithms for the optimization of the system.

The post-nonlinear case is also dealt by Taleb and Jutten [19], who propose to minimize the mutual information between the estimated sources using a nonlinear system

that precedes a linear separating stage.

The nonlinear mapping from the observations to the sources can also be modeled using multilayer perceptrons (MLP). Yang *et al* [22] use a two-layer perceptron as system to separate the sources. They ensure that the neural network is *invertible* by setting the number of neurons in the hidden layer to the number of sources. *This is a very severe constraint that endangers the approximation capabilities of the net.* Nevertheless, if such a constraint is eliminated (as, for example, in the *ensemble learning approach* [14]), one meets serious mathematical dificulties and a high computational complexity. Hence, *rather than increasing the number of neurons in the first hidden layer, the solution may be the use of two of more hidden layers.* In Sections 2 and 3, this basic idea is developed into a practical proposal. In Section 4, learning rules for the MLP are derived. Section 5 contains the results of experiments conducted to show the performance and potential problems. Section 6 contains our main conclusions.

2. MODEL STRUCTURE

Let $\mathbf{s}(t) = [s_1(t), \ldots, s_N(t)]^T$ with $s_i(t)$, $i = 1, \ldots, N$ being N mutually independent random processes whose *pdfs* are unknown. Suppose that we have N sensors; the output of each one denoted by $x_i(t)$, $i = 1, \ldots, N$, which measure a combination of the N sources. In a vector form, this is expressed as

$$\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t)) \tag{1}$$

where $\mathbf{x}(t) = [x_1(t), \ldots, x_N(t)]^T$ is called the observation vector, being the information that is available, and $\mathcal{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is an unknown memoryless differentiable bijective (reversible) mapping.

The task of BSS is *that of recovering the sources from the observed signals.* Here, the idea is to approximate the inverse of $\mathcal{F}$ by using the neural network shown in Figure 1, as MLPs have the universal approximation property for smooth continuous mappings. Such a network is described by the equations:

$$\mathcal{F}^{-1}(\mathbf{x}(t)) \approx \mathbf{y}(t) = \mathbf{W}_1\mathbf{g}(\mathbf{u}(t) + \mathbf{b}_1) \tag{2a}$$

being

$$\mathbf{u}(t) = \mathbf{W}_2\mathbf{f}(\mathbf{w}(t) + \mathbf{b}_2) \tag{2b}$$

and

$$\mathbf{w}(t) = \mathbf{W}_3\mathbf{x}(t) \tag{2c}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$ are square matrices,

$$\mathbf{g}(\mathbf{vector}) = [g_1(vector_1), \ldots, g_N(vector_N)]^T,$$

$$\mathbf{f}(\mathbf{vector}) = [f_1(vector_1), \ldots, f_N(vector_N)]^T,$$

where $\mathbf{vector} = [vector_1, \ldots, vector_N]^T$, $g_i(\cdot)$ and $f_i(\cdot)$ are any continuous sigmoid-type function and both $\mathbf{b}_1$ and $\mathbf{b}_2$ are $N \times 1$ vectors. Since the mixing system is memoryless, notice that *we will drop time index t* in the following.

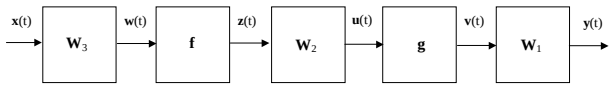

Figure 1: Neural Network Arquitecture.

3. CRITERION FUNCTION

3.1. Information-Theoretic Criterion

The guiding principle of *unsupervised* source separation is, in most approaches, to transform the observed data so that the transformed variables are as mutually independent as possible. Even thought that this transformation is not *unique* in the non-linear mixture case, *numerous* experiments show that one often *separates* the sources (see, for example, [16, 21, 22]). In explanation of this favourable behaviour, one can conjecture that separating systems with few parameters (*i.e.*, degrees of freedom) work because they are not able to produce signals that are more independent than the original sources, provided that the nonlinear mixture of the sources was smooth and can be undone through a smooth transformation. The conjecture seems to be valid for a wide range of source *pdfs*.

The degree of dependence between the outputs is commonly quantified by their *mutual information*, which is defined as:

$$I(\mathbf{y}) = -H(\mathbf{y}) + \sum_{i=1}^{N} H(y_i) \tag{3}$$

where $H(\cdot)$ is the Shannon differential entropy. A well-known property is that $I(\mathbf{y}) \geq 0$ with equality if and only if the outputs are independent. For the sake of simplicity, other measures of independence (such as, for example, that given in [2] or Renyi's mutual information [8]) have not been taken into account.

3.2. Practical Cost Function

By using (2), $-H(\mathbf{y})$ can be easily expanded as:

$$-H(\mathbf{y}) = -H(\mathbf{x}) - \sum_{i=1}^{3} \log \mid \mathbf{W}_i \mid -$$

$$- \sum_{i=1}^{N} E[\log \mid g_i'(u_i + b_i^1) \mid] - \sum_{i=1}^{N} E[\log \mid f_i'(w_i + b_i^2) \mid] \tag{4}$$

where $H(\mathbf{x})$ is the joint entropy of the observed signals,

$$\mid \mathbf{W}_i \mid = \mid \det(\mathbf{W}_i) \mid,$$

b_i^j stands for the i-th component of vector $\mathbf{b}_j$ and $g_i'(\cdot)$, $f_i'(\cdot)$ are the first-order derivatives of $g_i(\cdot)$ and $f_i(\cdot)$, respectively.

Since the term $H(\mathbf{x})$ in expression (4) *does not depend on the parameters of the MLP*, the minimization of the mutual information between the outputs *is equivalent to minimize the index*:

$$\mathcal{I}(\mathbf{y}) \stackrel{def}{=} I(\mathbf{y}) + H(\mathbf{x}) \tag{5}$$

One serious problem is that the exact calculation of the marginal entropies $H(y_i)$ is rather involved. If we assume

that the outputs are *standardized* (*i.e.*, they are zero-mean unit-variance signals), their entropies can be approximated as (see [9], chapter 5 and [22]):

$$H(y_i) \approx \frac{1}{2}\log(2\pi e) - \frac{(\kappa_3^i)^2}{12} - \frac{(\kappa_4^i)^2}{48} + \frac{3}{8}(\kappa_3^i)^2\,\kappa_4^i + \frac{(\kappa_4^i)^3}{16} \quad (6)$$

where $\kappa_3^i = E[(y_i)^3]$ is the skewness measure of y_i and $\kappa_4^i = E[(y_i)^4] - 3$ equals its kurtosis. In order to encourage such a standardization, Tikhonov regularization terms are added to (5) according to

$$\mathcal{J}(\mathbf{y}) = \mathcal{I}(\mathbf{y}) + \lambda_1 \sum_{i=1}^{N} (E[y_i])^2 + \lambda_2 \sum_{i=1}^{N} (E[y_i^2] - 1)^2 \quad (7)$$

Using (4) and (6) in (7), $\mathcal{J}(\mathbf{y})$ can be estimated and minimized.

We may impose some additional constraints or prior information on the sources (*e.g.* sparsity, super-gaussianity, and so on). Even though that the MLP may create sparse or super-gaussian outputs that do not recreate the original sources, Tan *et al* [21] have reported good results by imposing perfect matching of moments between the outputs of the net and the sources.

In addition, we are conscious of the asymmetry of the cost function, since $H(\mathbf{y})$ is *exactly* calculated whereas the marginal entropies $H(y_i)$ are only approximated by using a Gram-Charlier expansion (which, in addition, assumes that the *pdfs* are not very far from the Gaussian density).

Both problems should be addressed in future investigations.

4. LEARNING RULES

In the following, let b_i^j denote the i-th entry of vector $\mathbf{b}_j$. Similarly, w_{qp}^k will stand for the (q,p)-th component of matrix $\mathbf{W}_k$.

4.1. Differentiating the joint entropy $H(\mathbf{y})$

It is well-known that

$$\frac{\partial \log \mid \mathbf{W}_i \mid}{\partial \mathbf{W}_i} = \mathbf{W}_i^{-T} \quad (8)$$

hence,

$$\boxed{\frac{\partial H[\mathbf{y}]}{\partial \mathbf{W}_1} = \mathbf{W}_1^{-T}} \quad (9)$$

Similarly, we easily obtain

$$\boxed{\frac{\partial H[\mathbf{y}]}{\partial \mathbf{W}_2} = \mathbf{W}_2^{-T} - E[\Phi_g[\mathbf{u}]\mathbf{z}^T]} \quad (10)$$

where

$$\Phi_g[\mathbf{u}] \stackrel{def}{=} -[\frac{g_1''(u_1+b_1^1)}{g_1'(u_1+b_1^1)}, \ldots, \frac{g_N''(u_N+b_N^1)}{g_N'(u_N+b_N^1)}]^T \quad (11)$$

and

$$\boxed{\frac{\partial H[\mathbf{y}]}{\partial \mathbf{b}_1} = -E[\Phi_g[\mathbf{u}]]} \quad (12)$$

Now, observe that

$$\frac{\partial}{\partial w_{kp}^3} \log \mid g_i'(u_i + b_i^1) \mid = \frac{g_i''}{g_i'} \frac{\partial u_i}{\partial w_{kp}^3} \quad (13)$$

since

$$u_i = \sum_{j=1}^{N} w_{ij}^2 f_j(\sum_{q=1}^{N} w_{jq}^3 x_q + b_j^2) \quad (14)$$

it follows that

$$\sum_i \frac{\partial}{\partial w_{kp}^3} \log \mid g_i' \mid = \sum_i \frac{g_i''}{g_i'} w_{ik}^2\, f_k'\, x_p \quad (15)$$

where f_k' is the first order derivative of f_k. Hence, by setting

$$\mathbf{D}_f(\mathbf{w}) \stackrel{def}{=} diag(f_1'(w_1 + b_1^2), \ldots, f_N'(w_N + b_N^2)) \quad (16)$$

it can be written that

$$\sum_i \frac{\partial}{\partial \mathbf{W}_3} \log \mid g_i' \mid = -\mathbf{D}_f(\mathbf{w})\, \mathbf{W}_2^T\, \Phi_g(\mathbf{u})\, \mathbf{x}^T \quad (17)$$

Similarly,

$$w_i = \sum_j w_{ij}^3\, x_j \quad (18)$$

thus,

$$\sum_i \frac{\partial}{\partial w_{kp}^3} \log \mid f_i'(w_i + b_i^2) \mid = \frac{f_k''}{f_k'}\, x_p \quad (19)$$

and, consequently,

$$\sum_i \frac{\partial}{\partial \mathbf{W}_3} \log \mid f_i'(w_i + b_i^2) \mid = -\Phi(\mathbf{w})\, \mathbf{x}^T \quad (20)$$

where

$$\Phi_f[\mathbf{w}] \stackrel{def}{=} -[\frac{f_1''(w_1+b_1^2)}{f_1'(w_1+b_1^2)}, \ldots, \frac{f_N''(w_N+b_N^2)}{f_N'(w_N+b_N^2)}]^T \quad (21)$$

Finally, we obtain

$$\boxed{\frac{\partial H[\mathbf{y}]}{\partial \mathbf{W}_3} = \mathbf{W}_3^{-T} - E[\mathbf{D}_f(\mathbf{w})\, \mathbf{W}_2^T\, \mathbf{\Phi}_g(\mathbf{u})\, \mathbf{x}^T + \mathbf{\Phi}_f(\mathbf{w})\, \mathbf{x}^T]} \quad (22)$$

and, similarly

$$\boxed{\frac{\partial H[\mathbf{y}]}{\partial \mathbf{b}_2} = -E[\mathbf{D}_f(\mathbf{w})\, \mathbf{W}_2^T\, \mathbf{\Phi}_g(\mathbf{u}) + \mathbf{\Phi}_f(\mathbf{w})\,]} \quad (23)$$

4.2. Differentiating the marginal entropies $H(y_i)$

Using (6), some algebra shows that

$$\frac{\partial H(y_i)}{\partial \alpha} = E[\hat{y_i} \frac{\partial y_i}{\partial \alpha}] \quad (24)$$

where

$$\hat{y_i} = \{-\frac{\kappa_3^i}{2} + \frac{9}{4}\kappa_3^i\,\kappa_4^i\,\}y_i^2 + \{\frac{3}{4}(\kappa_4^i)^2 + \frac{3}{2}(\kappa_3^i)^2 - \frac{1}{6}\kappa_4^i\}\, y_i^3 \quad (25)$$

being $\kappa_3^i = E[(y_i)^3]$ and $\kappa_4^i == E[(y_i)^4] - 3$. Hence, when $\alpha = w_{ij}^1$

$$\frac{\partial}{\partial w_{ij}^1} \sum_k H(y_k) = E[\hat{y_i}\, v_j] \quad (26)$$

and, consequently

$$\boxed{\frac{\partial}{\partial \mathbf{W}_1} \sum_{k=1}^{N} H(y_k) = E[\hat{\mathbf{y}}\, \mathbf{v}^T]} \quad (27)$$

where $\hat{\mathbf{y}} = [\hat{y_1}, \ldots, \hat{y_N}]^T$. Secondly, when $\alpha = w_{ij}^2$ and using that

$$\frac{\partial y_k}{\partial w_{ij}^2} = w_{ki}^1 \, g_i' \, z_j \tag{28}$$

we obtain

$$\frac{\partial}{\partial w_{ij}^2} \sum_k H(y_k) = \sum_k E[\hat{y}_k \, w_{ki}^1 \, g_i' \, z_j] \tag{29}$$

Thus, it follows that

$$\boxed{\frac{\partial}{\partial \mathbf{W}_2} \sum_k H(y_k) = E[\mathbf{D}_g \, \mathbf{W}_1^T \, \hat{\mathbf{y}} \, \mathbf{z}^T]} \tag{30}$$

where

$$\mathbf{D}_g(\mathbf{u}) \stackrel{def}{=} diag(g_1'(u_1 + b_1^1), \ldots, g_N'(u_N + b_N^1)) \tag{31}$$

Using a similar procedure, it can be obtained that

$$\boxed{\frac{\partial}{\partial \mathbf{b}_1} \sum_k H(y_k) = E[\mathbf{D}_g \, \mathbf{W}_1^T \, \hat{\mathbf{y}}]} \tag{32}$$

The most involved calculation is the following one: since

$$y_k = \sum_p w_{kp}^1 g_p(u_p + b_p^1) \tag{33}$$

and

$$\frac{\partial u_p}{\partial w_{ij}^3} = w_{pi}^2 \, f_i' \, x_j \tag{34}$$

we obtain that

$$\frac{\partial}{\partial w_{ij}^3} \sum_k H(y_k) = \sum_k \sum_p E[\hat{y}_k \, w_{kp}^1 \, g_p' \, w_{pi}^2 \, f_i' \, x_j] \tag{35}$$

or, in matrix form,

$$\boxed{\frac{\partial}{\partial \mathbf{W}_3} \sum_k H(y_k) = E[\mathbf{D}_f \, \mathbf{W}_2^T \, \mathbf{D}_g \, \mathbf{W}_1^T \hat{\mathbf{y}} \, \mathbf{x}^T]} \tag{36}$$

Similarly,

$$\boxed{\frac{\partial}{\partial \mathbf{b}_2} \sum_k H(y_k) = E[\mathbf{D}_f \, \mathbf{W}_2^T \, \mathbf{D}_g \, \mathbf{W}_1^T \hat{\mathbf{y}}]} \tag{37}$$

4.3. Taking Tikhonov regularization terms into account

Tikhonov terms are

$$\lambda_1 \sum_i (E[y_i])^2 + \lambda_2 \sum_i (E[y_i^2] - 1)^2 \tag{38}$$

which is similar to (6) in the sense that both are a combination of statistics. Hence, the derivatives of the Tikhonov terms can be easily taking into account by incorporating the following terms into the definition of vector $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} \leftarrow \hat{\mathbf{y}} + 2 \, \lambda_1 \, E[\mathbf{y}] + 4 \, \lambda_2 \, E[\mathbf{y} \odot \mathbf{y} - \mathbf{1}] \odot \mathbf{y} \tag{39}$$

where $\odot$ stands for the Hadamard product and $\mathbf{1}$ is a vector of ones.

4.4. Natural gradient

Finally, to avoid inverse matrix operations, we use the *natural gradient* rule (see [6], chapter 1) and derive the learning algorithm that appears in Table 1, *i.e.*, the unsupervised learning rule for minimizing the mutual information between the outputs of a perceptron with two hidden layers. It is worth noticing that similar rules can be obtained *by maximizing the entropy* of these outputs instead of minimizing their mutual information.

1. $\frac{d}{dt}\mathbf{W}_1 = \{I - E[\hat{\mathbf{y}}\mathbf{y}^T]\}\mathbf{W}_1$
2. $\frac{d}{dt}\mathbf{W}_2 = \{I - E[\Phi_g \, \mathbf{u}^T - \mathbf{D}_g \, \mathbf{W}_1^T \hat{\mathbf{y}}\mathbf{u}^T]\}\mathbf{W}_2$
3. $\frac{d}{dt}\mathbf{W}_3 = \{I - E[\mathbf{D}_f \mathbf{W}_2^T \Phi_g \mathbf{w}^T + \Phi_f \mathbf{w}^T + + \mathbf{D}_f \mathbf{W}_2^T \mathbf{D}_g \mathbf{W}_1^T \hat{\mathbf{y}} \mathbf{w}^T]\}\mathbf{W}_3$
4. $\frac{d}{dt}\mathbf{b}_1 = -E[\Phi_g + \mathbf{D}_g \, \mathbf{W}_1^T \hat{\mathbf{y}}]$
5. $\frac{d}{dt}\mathbf{b}_2 = -E[\mathbf{D}_f \mathbf{W}_2^T \Phi_g + \Phi_f + \mathbf{D}_f \mathbf{W}_2^T \mathbf{D}_g \mathbf{W}_1^T \hat{\mathbf{y}}]$

Table 1: Learning Rules.

5. COMPUTER SIMULATIONS

In order to check the validity and performance of the proposed adaptive learning algorithm, it has been extensively simulated on a computer. Due to limited space, we shall present in this paper only a few illustrative examples. In all of them, we have employed a batch version of the learning algorithm (block size and learning rate were set to 100 samples and 0.001 respectively) and both regularization parameters λ_1 and λ_2 were set to 10. A little momentum term was also added to speed up the learning process.

5.1. Experiment 1. Post-nonlinear mixture of two sources.

A simple experiment in which the net separates a post-nonlinear mixture of two signals. All the signals are depicted in Figure 2. The mixtures were generated using the model:

$$\mathbf{x} = \tanh(\mathbf{A} \, \mathbf{s}),$$

where

$$\mathbf{A} = \begin{array}{cc} 0.3382 & 0.4768 \\ -0.1091 & 0.6422 \end{array}$$

Separation is clearly achieved. In fact, separation is always possible under mild conditions [10] in the two-source case.

5.2. Experiment 2.- "Hard" non-linear mixture of three sources.

In this case, the mixtures were generated as:

$$\mathbf{x} = \mathbf{A}_1 \tanh(\mathbf{A}_2 \, \mathbf{s}),$$

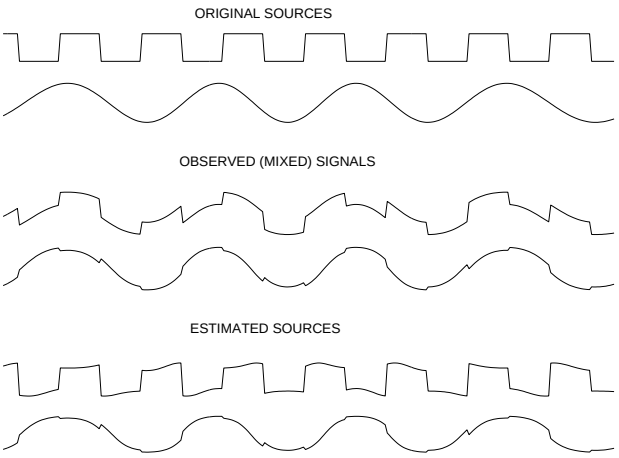

Figure 2: From the top to the bottom: sources, post-nonlinear mixtures and estimated sources after 10 sweeps.

where

$$\mathbf{A}_1 = \begin{bmatrix} -0.0882 & -0.1747 & -0.6919 \\ -0.2947 & -0.7114 & -0.8542 \\ -0.7857 & 0.2968 & -0.7538 \end{bmatrix},$$

and

$$\mathbf{A}_2 = \begin{bmatrix} -0.7207 & 0.8083 & -0.4853 \\ -0.2900 & 0.6680 & 0.5059 \\ 0.8585 & 0.9536 & -0.0655 \end{bmatrix}$$

The mixing function is strongly nonlinear: it is noteworthy that the popular algorithm JADE ([3], available at [11]) which was originally devised for linear mixtures, is not able to separate the sources.

A 1000-sample training set was used for adjusting the network. In this experiment, the algorithm converges in about 20 sweeps. It has been found experimentally that matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ adapt at a much slower pace than matrix $\mathbf{W}_3$.

Results are depicted in Figure 3 (only 80 samples of each signal are plotted for the sake of clarity). The estimation of the first and second sources seems to be acceptable. On the contrary, the third source *is still distorted* after separation. Figure 4 shows the magnitude of the Fourier Transforms of the third source and its estimate. Two interfering peaks which are caused by the other sources are clearly visible and we can easily realize that they are not harmonic components of the Fourier Transforms. Hence, it makes good sense to remove them in a post-processing stage even if we do not know that the three sources are periodic signals.

5.3. Experiment 3.- Local minima.

This experiment demonstrates the existence of spurious local minima. We consider a nonlinear mixture of five uniform sources in which, according to our calculations, the minimum value of $\mathcal{J}(\mathbf{y})$ is about 10.

Each learning curve in Figure 5 corresponds to different initial conditions. Separation is achieved in the experiment that corresponds to the bottom curve. It is noteworthy that

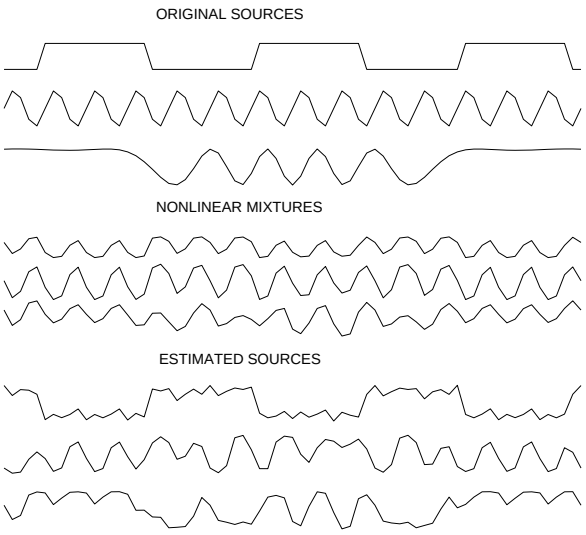

Figure 3: From the left to the right, *a*) source signals *b*) nonlinear mixtures *c*) estimated sources (after 20 sweeps).

both $\mathbf{W}_1$ and $\mathbf{W}_2$ are often close to permutation matrices in the vicinity of the local minima.

6. DISCUSSION AND FUTURE RESEARCH

The neural network has been applied to the nonlinear BSS problem. Our experiments mainly show that:

- In this context, networks with two hidden layers are more prone to fall into bad local minima than networks with a single hidden layer [22]. To avoid such undesired minima, we are currently investigating meta-heuristics and global search methods[1]. Promising results have been obtained by using an evolutionary algorithm [18]. Further research in this field would be clearly fruitful.
- Separation is hindered by the fact that independence-based cost functions, such as (7), can not distinguish between the estimated sources y_1, y_2 and y_3 and any of their functions $h_1(y_1)$, $h_2(y_1)$ and $h_3(y_1)$, provided that y_1, y_2 and y_3 are mutually independent.

7. REFERENCES

[1] L. B. Almeida, "ICA of Linear and Nonlinear Mixtures Based on Mutual Information", *Proc. of International Joint Conference on Neural Networks (IJCNN 2001)*, available at http://www.cnel.ufl.edu/info/infopapers.html.

[2] G. Burel, "Blind Separation of Sources – A nonlinear neural algorithm", *Neural Networks*, vol. 5, No. 6, pp 937-947, 1992.

[1]The *Genetic Optimization Toolbox* was used, available at *www.mathtools.net*.

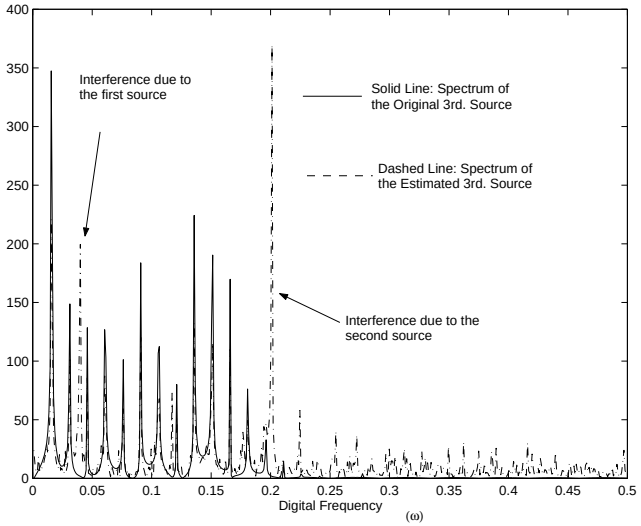

Figure 4: Fourier Transforms of the original and estimated source.

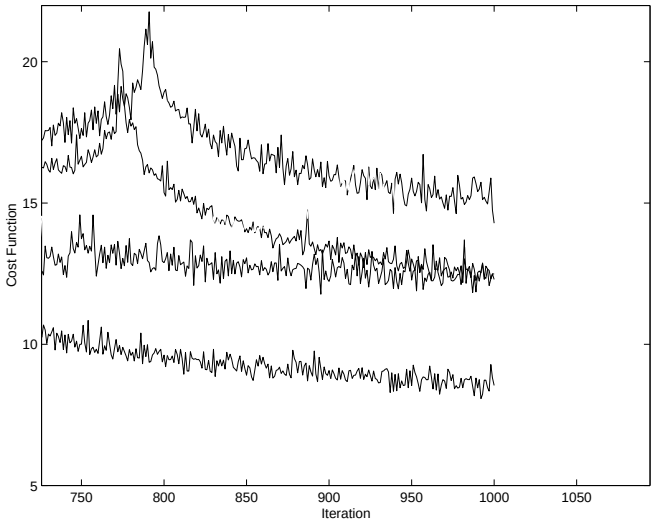

Figure 5: Learning curves for different initial conditions. Only the curve plotted at the bottom of the figure corresponds to a successful separation.

[3] J-F. Cardoso and A. Souloumiac, " Blind Beamforming for non-Gaussian Signals ", *Proceedings of the Inst. Elect. Eng.*, Vol.140 (F6), pp.362-370, 1993.

[4] A. Cichocki, S.I.Amari, "Adaptive Blind Signal and Image Processing", *John Willey and Sons*, 2002.

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

[6] S. Haykin, Ed., "Unsupervised Adaptive Fitering. Volume I: Blind Source Separation", *John Willey and Sons*, 2000.

[7] S. Haykin, "Neural Networks – A comprehensive Foundation", *Prentice-Hall*, 1998.

[8] K.E. Hild II, D. Erdogmus, J.C. Príncipe, "Blind Source Separation Using Renyi's Mutual Information", *IEEE Signal Processing Letters*, vol. 8, no. 6, pp. 174-176, 2001.

[9] A. Hyvärinen, J. Karhunen and E. Oja, "Independent Component Analysis", *John Willey and Sons*, 2001.

[10] A. Hyvärinen and P. Pajunen, "Nonlinear Independent Component Analysis: Existence and uniqueness results", *Neural Networks*, vol. 12, No. 3, pp 429-439, 1999.

[11] http://sig.enst.fr:80/∼cardoso/stuff.html

[12] C. Jutten and J. Herault, "Blind Separation of Sources, Part I: an adaptive algorithm based on neuromimetic architecture", *Signal Processing*, vol. 24, pp.1-10, 1991.

[13] C. Jutten and A. Taleb, "Source Separation: from dusk till dawn.", *Proc. 2nd. Int. Workshop on Independent Component Analysis and Blind Source Separation*, pp. 15-26, Helsinki, Finland, 2000.

[14] J. Karhunen, "Nonlinear Independent Component Analysis". Everson and S. Roberts (Eds.), *"ICA: Principles and Practice"*, pp. 113-134 Cambridge University Press, 2001. (available at: http://www.cis.hut.fi/juha/papers/nica-chap.ps.gz)

[15] J. Karhunen, S. Malaroiu and M. Ilmoniemi, "Local linear ICA based on clustering", *International Journal of Neural Systems*, vol. 10, no. 6, pp. 439-451, 2000.

[16] A. Koutras, E. Dermatas and G. Kokkinakis, "Neural Network Based Blind Source Separation of Non-Linear Mixtures", *Proc. ICANN*, pp. 561-567, Vienna, Austria, 2001.

[17] C. G. Puntonet, A. Mansour, C. Bauer and E. Lang, "Separation of Sources using Simulated Annealing and Competitive Learning", *Neurocomputing* (to appear).

[18] F. Rojas, I. Rojas, R. Martín-Clemente and C.G.Puntonet, "Nonlinear Blind Source Separation using Genetic Algorithms", *Proc. ICA 2001*, pp.771-774, San Diego, USA, 2001

[19] A. Taleb and C. Jutten " Source Separation in Post-Nonlinear Mixtures " *IEEE Trans. on Signal Proc.*, Vol. 47, No. 10, pp.2807-2820, 1999.

[20] A. Taleb, J. Sol and C. Jutten " Blind Inversion of Wiener Systems " *Proc. IWANN 99*, pp.655-664, 1999

[21] Y. Tan, J. Wang and J.Zurada, " Nonlinear Blind Source Separation using a Radial Basis Function Network " *IEEE Trans. on Neural Networks.*, Vol. 12, No. 1, pp.124-134, 2001.

[22] H.H. Yang, S.Amari and A. Cichocki, "Information-Theoretic Approach to Blind Separation of Sources in Non-Linear mixture", in *Signal Processing*, vol. 64, No. 3, pp. 291-300, 1998.