NONLINEAR BLIND SOURCE SEPARATION BY SPLINE NEURAL NETWORKS

Mirko Solazzi(*), Francesco Piazza (*) and Aurelio Uncini (**)

(*) Dipartimento di Elettronica e Automatica - University of Ancona
Via Brecce Bianche, 60131 Ancona-Italy.
Fax:+39 (071) 2204464 - email: mrksol@eealab.unian.it - Internet: http://nnsp.eealab.unian.it/
(**) Dipartimento INFOCOM - University of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome - Italy.
Fax +39 (06) 4873300 email: aurel@ieee.org - Internet: http://infocom.uniroma1.it/aurel

ABSTRACT

In this paper a new neural network model for blind demixing of nonlinear mixtures is proposed. We address the use of the Adaptive Spline Neural Network recently introduced for supervised and unsupervised neural networks. These networks are built using neurons with flexible B-spline activation functions and in order to separate signals from mixtures, a gradient-ascending algorithm which maximize the outputs entropy is derived.
In particular a suitable architecture composed by two layers of flexible nonlinear functions for the separation of nonlinear mixtures is proposed. Some experimental results that demonstrate the effectiveness of the proposed neural architecture are presented.

1. INTRODUCTION

The problem of blind source separation (BSS) consists on the recovery of independent sources from their mixture. This is important in several applications like speech enhancement, telecommunication, biomedical signal processing, etc. Most of the work on BSS mainly addresses the cases of instantaneous linear mixture [1-2][12].

Let **A** a real or complex rectangular ($n\times m$; $n\geq m$) matrix, the data model for linear mixture can be expressed as

$$\mathbf{x}(t) = \mathbf{A}\,\mathbf{s}(t) \qquad (1)$$

where $\mathbf{s}(t)$ represents the statistically independent sources array while $\mathbf{x}(t)$ is the array containing the observed signals.

For real world situation, however, the basic linear mixing model (1) is too simple for describing the observed data. In many applications such as the nonlinear characteristic introduced by preamplifiers of receiving sensors, we can consider a non-linear mixing. So a nonlinear mixing is more realistic and accurate than linear model.

For instantaneous mixtures, a general nonlinear data model can have the form

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)) \qquad (2)$$

were **f** represents an unknown vector of real functions.

Although some algorithms for nonlinear BSS have already been proposed, see for example [6-11] and the references therein, in this paper we proposed a new neural network model for blind demixing of nonlinear mixtures. In particular we address the use of the Adaptive Spline Neural Network (ASNN) introduced for supervised neural networks in [4-5].

The basic scheme of the ASNN is very similar to classical neural structures, but with improved non linear flexible activation functions. These functions can change their shapes adapting few control points by the learning algorithm.

Recently, under some control points constraints, the ASNNs have been successfully applied using an unsupervised learning algorithm for linear BSS problems [3].

As demonstrate in [3-5], the flexible activation functions used in the ASNNs have several interesting features: they 1) are easy to adapt, 2) have the necessary smoothing characteristics, 3) are easy to implement both in hardware and in software.

2. NON-LINEAR MIXING/DEMIXING BSS MODEL

Let assume for simplicity that the number of independent source signals is equals to the number of mixtures such that n=m, the model for the mixture assumed in this paper, shown in figure 1, is a simple nonlinear mixing model without cross-channel non-linearity.

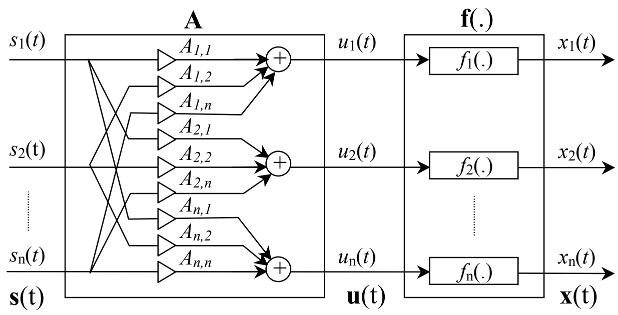

Fig. 1: Non linear mixing model without cross-channel non-linearity.

$\mathbf{s}(t) = [s_1(t), s_2(t), ..., s_n(t)]^T$ is the unknown instantaneous source vector, where components $s_i(t)$ is supposed to be mutually independent (different people speaking, noise, music,…). **A** is an unknown mixing matrix ($n\times n$) that represents the linear superposition of original signals for each channel. Vector $\mathbf{f}(.) = [f_1(.), f_2(.), ... f_n(.)]^T$ denotes component-wise nonlinear channel transfer function, which is different for each channel.. In synthesis, the unknown mixing system is modeled as a cascaded instantaneous linear mixing and non-linear function transformation. So, it is obvious to consider the corresponding separating system as the inverse transformation **g**(.) of the non-linear part followed by a demixing matrix **W** opportunely defined during an adaptation process. This separating system is illustrated in figure 2.

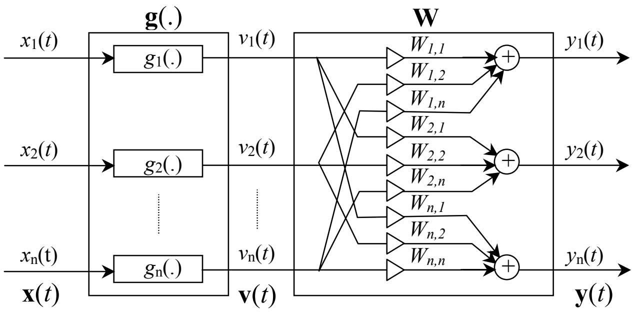

Fig. 2: Non linear blind separation system.

3. INFORMATION MAXIMIZATION FOR NON-LINEAR BLIND SOURCE SEPARATION

It is well-known that separation of independent sources is possible using concepts derived from information theory. Two or more random variables s_j (j=1,2,..,n) are stochastically independent if knowledge of the values one of them tells us nothing about the values of others. More generally a set of signals are independent if their join probability density function (pdf) can be decomposed as:

$$q(s_1, s_2, .., s_n) = \prod_{j=1}^{n} q_j(s_j) \tag{3}$$

where $q_j(s_j)$ is the pdf of j-th source signal.

Let **y** be estimated source signal vector and let $p_y(\mathbf{y})$ be its pdf. In order to measure the degree of independence, we use an adequately chosen independent probability distribution $q(\mathbf{y}) = \prod_i q_i(y_i)$ and consider the Kullback-Leibler (KL) divergence between two probability distributions $p_y(\mathbf{y})$ and $q(\mathbf{y})$:

$$KL\left[p_y(\mathbf{y}) \,\|\, q(\mathbf{y})\right] = \int p_y(\mathbf{y}) \log \frac{p_y(\mathbf{y})}{q(\mathbf{y})} d\mathbf{y} \tag{4}$$

This measure is non-negative and reaches its minimum value or vanishes if and only if p=q, in other terms when the vector **y** is mutually independent component-wise. Minimizing the KL divergence can make the estimated source signals independent. Under some hypothesis it is also equivalent to maximizing the entropy of estimated signals, as demonstrated by Bell and Sejnowski [1]:

$$KL\left[p_y(\mathbf{y}) \,\|\, q(\mathbf{y})\right] = \prod_{i=1}^{N} H(y_i) - H(\mathbf{y}) \tag{5}$$

where $H(\mathbf{y}) = -E\left\{\log\left(p_y(\mathbf{y})\right)\right\}$ is the entropy of **y** and $H(y_i)$ the entropy of its i-th component.

Another important property of the KL divergence is the invariance under an invertible non-linear transformation g(.) of data samples:

$$KL\left[p(\mathbf{y}) \,\|\, q(\mathbf{s})\right] = KL\left[p(g(\mathbf{y})) \,\|\, q(g(\mathbf{s}))\right] \tag{6}$$

As a consequence, the mutual independence will not be affected by any invertible non-linear function transformation. If we introduce a non-linear differentiable mapping **z**=h(**y**)=g(**x**) that represent the overall separating system, the relations between input and output joint distributions of this mapping is:

$$p_z(\mathbf{z}) = \frac{p_x(\mathbf{x})}{|\mathbf{J}|} \tag{7}$$

where |**J**| is the determinant of the Jacobian matrix of the transformation:

$$|\mathbf{J}| = \det \begin{bmatrix} \partial z_1/\partial x_1 & \cdots & \partial z_1/\partial x_N \\ \vdots & & \vdots \\ \partial z_N/\partial x_1 & \cdots & \partial z_N/\partial x_N \end{bmatrix} \tag{8}$$

The entropy of the transformed output signal **z** is given by:

$$H(\mathbf{z}) = -E\left\{\log\left(p_z(\mathbf{z})\right)\right\} = E\left\{\log|\mathbf{J}|\right\} + H(\mathbf{x}) \tag{9}$$

The optimal solution of the information-theoretic criterion is obtained when the maximum entropy of **z** is reached, when $p_z(\mathbf{z})$ is a uniform distribution. In this case we have $p_x(\mathbf{x})$ =|**J**|. Since the second term in (9) does not contain any model parameter, thus maximization of H(**z**) is performed by only maximizing the first term with respect to model parameter set Ω . Using the gradient ascent learning algorithm, we have to consider the derivative of the entropy function H(**z**) with respect to model parameters:

$$\Delta\Omega = \eta \frac{\partial H(\mathbf{z})}{\partial \Omega} = \eta \frac{\partial}{\partial \Omega} \log|\mathbf{J}| = \eta \left(\frac{\partial}{\partial \Omega} \log\left(\prod_{i=1}^{n} \frac{\partial z_i}{\partial y_i} \right) + \right.$$

$$\left. + \frac{\partial}{\partial \Omega} \log|\mathbf{W}| + \frac{\partial}{\partial \Omega} \log\left(\prod_{i=1}^{n} \frac{\partial v_i}{\partial x_i} \right) \right) \tag{10}$$

4. THE ADAPTIVE SPLINE NN MODEL

4.1 Adaptive Spline Neuron

Spline activation functions are smooth parametric curves, divided in multiple tracts (spans) each controlled by four control points. Let $h(x)$ be the non-linear function to reproduce, then the spline activation function can be expressed as:

$$y = h(x) = \bar{h}(u, i) \tag{11}$$

i.e. as a composition of (N-2) spans (where N is the total number of the control points Q_j (j=1,2,..,N) each depending from a local variable $u \in [0,1)$ and controlled by the $Q_i, Q_{i+1}, Q_{i+2}, Q_{i+3}$ control points (see Fig. 3). The parameters i, u can be derived by a dummy variable z

$$z = \frac{x}{\Delta x} + \frac{N-1}{2} \tag{12}$$

$$z = \begin{cases} 1 & \text{if } z < 1 \\ z & \text{if } 1 \le z \le N-3 \\ N-3 & \text{if } z > N-3 \end{cases} \tag{13}$$

where Δx is the fixed distance between two adjacent control points; constraints imposed by equation (13) are necessary to keep the input within the active region that encloses the control points. Separating z into integer and fractional parts using the floor operator $\lfloor . \rfloor$ finally we get

$$i = \lfloor z \rfloor \qquad \text{and} \qquad u = z - i \tag{14}$$

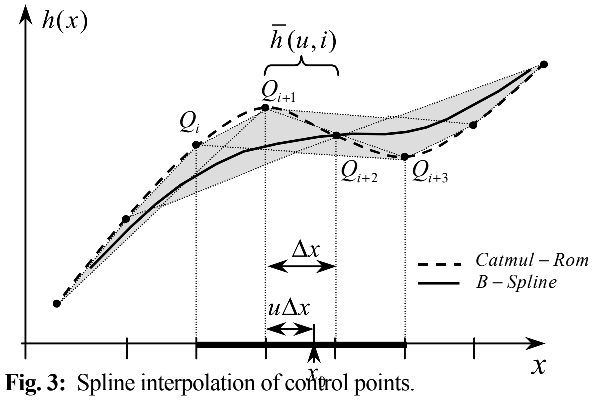

Fig. 3: Spline interpolation of control points.

In matrix form the output can be expressed as

$$y = \bar{h}(u,i) = \mathbf{T}_u \cdot \mathbf{M} \cdot \mathbf{Q}_i \tag{15}$$

where :

$$\mathbf{T}_u = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \tag{16}$$

$$\mathbf{M} = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \tag{17}$$

$$\mathbf{Q}_i = \begin{bmatrix} Q_i & Q_{i+1} & Q_{i+2} & Q_{i+3} \end{bmatrix}^T \tag{18}$$

The corresponding structure is shown in Fig. 4. In order to ensure the monotonously increasing characteristic of the function, the following additional constraint must be imposed:

$$Q_1 < Q_2 < \ldots < Q_N \tag{19}$$

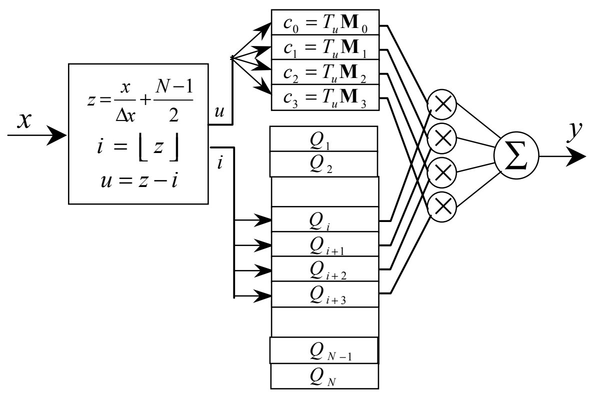

Fig. 4: The adaptive spline activation function.

4.2 Structure of the Spline Neural Separating System

We employed the spline neuron structure to implement the blind separation system. Both output non-linearities and input non-linearities are spline based functions. The structure of the non-linear BSS system is depicted in figure 5. The parameter set Ω for this model include elements of demixing matrix $\mathbf{W}$, spline control points $Q^{g_j}_{i+m}$ of each input non-linearity $g_j(.)$ and spline control points $Q^{h_j}_{i+m}$ of each output non-linearity $h_j(.)$

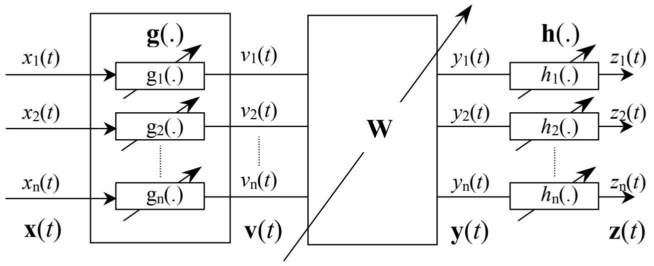

Fig. 5: Adaptive spline blind separation system.

4.3 Learning algorithm

We can derive the learning algorithm of parameters of the nonlinear separating model using a gradient ascent method, on the basis of the maximization entropy criterion. According to equation (10) we have:

$$\frac{\partial H(\mathbf{z})}{\partial Q^{h_j}_{i+m}} = \frac{\partial}{\partial Q^{h_j}_{i+m}} \log\left(\prod_{k=1}^{n} \frac{\partial z_k}{\partial y_k}\right) = \left.\frac{\dot{\mathbf{T}}_u \cdot \mathbf{M}_m}{\dot{\mathbf{T}}_u \cdot \mathbf{M} \cdot \mathbf{Q}^{h_j}_i}\right|_{\substack{u=u(y_j)\\ i=i(y_j)}} \tag{20}$$

$$\frac{\partial H(\mathbf{z})}{\partial Q^{g_j}_{i+m}} = \frac{\partial}{\partial Q^{g_j}_{i+m}} \log\left(\prod_{k=1}^{n} \frac{\partial z_k}{\partial y_k}\right) + \frac{\partial}{\partial Q^{g_j}_{i+m}} \log\left(\prod_{k=1}^{n} \frac{\partial v_k}{\partial x_k}\right) =$$

$$= \left(\mathbf{H}(\mathbf{y}) \cdot \mathbf{W}\right)_j \left(\mathbf{T}_{u(x_j)} \cdot \mathbf{M}_m\right) + \left.\frac{\dot{\mathbf{T}}_u \cdot \mathbf{M}_m}{\dot{\mathbf{T}}_u \cdot \mathbf{M} \cdot \mathbf{Q}^{g_j}_i}\right|_{\substack{u=u(x_j)\\ i=i(x_j)}} \tag{21}$$

$$\frac{\partial H(\mathbf{z})}{\partial \mathbf{W}} = \frac{\partial}{\partial \mathbf{W}} \log\left|\mathbf{W}\right| + \frac{\partial}{\partial \mathbf{W}} \log\left(\prod_{i=1}^{n} \frac{\partial z_i}{\partial y_i}\right) =$$

$$= \left[\mathbf{W}^T\right]^{-1} + \mathbf{H}(\mathbf{y})^T \cdot \mathbf{v}^T \tag{22}$$

where: $\dot{\mathbf{T}}_u = \begin{bmatrix} 3u^2 & 2u & 1 & 0 \end{bmatrix}$; $m \in (0,1,2,3)$; $\mathbf{M}_m$ is the m-th column of $\mathbf{M}$ matrix and:

$$\mathbf{H}(\mathbf{y}) = \begin{bmatrix} \frac{\ddot{h}_1(y_1)}{\dot{h}_1(y_1)} & \frac{\ddot{h}_2(y_2)}{\dot{h}_2(y_2)} & \cdots & \frac{\ddot{h}_n(y_n)}{\dot{h}_n(y_n)} \end{bmatrix} \tag{23}$$

$$\ddot{h}_j(y) = \ddot{\bar{h}}_j(u,i) = \begin{bmatrix} 6u & 2 & 0 & 0 \end{bmatrix} \cdot \mathbf{M} \cdot \mathbf{Q}^{h_j}_i \tag{24}$$

$$\dot{h}_j(y) = \dot{\bar{h}}_j(u,i) = \dot{\mathbf{T}}_u \cdot \mathbf{M} \cdot \mathbf{Q}^{h_j}_i \tag{25}$$

We can adjust the matrix $\mathbf{W}$ also using the natural gradient method proposed by Amari [2], which simplifies the learning rule, avoiding the inversion of $\mathbf{W}$, and accelerates the convergence of learning process:

$$\Delta\mathbf{W} = \eta \frac{\partial H(\mathbf{z})}{\partial \mathbf{W}} = \eta\left[\mathbf{I} + \mathbf{H}(\mathbf{y})^T \cdot \mathbf{y}^T\right] \cdot \mathbf{W} \tag{26}$$

where $\mathbf{I}$ denotes the identity matrix.

5. EXPERIMENTAL RESULTS

Experiments presented here were obtained using one second segments of speech from two different speakers and an uniform distributed noise.

All signals were sampled at 8kHz; no special post-processing was performed on the waveforms, other than that of normalizing their amplitudes so they were appropriate for use with the separating system. In the first experiment a female speech segment ad the noise were employed. In order to mix the two signals a randomly nonsingular mixing matrix was chosen: $\mathbf{A} = [0.65, -0.38; 0.45, 0.71]$.

In the second experiment a female speech and a male speech segments were mixed using the same mixing matrix. In both experiments a different non-linear transformation on each linear channel was applied., i.e. [11]

$$[f_1(u_1), f_2(u_2)] = [(u_1 + u_1^3)/2, (0.5u_2 + \tanh(0.3u_2))] .$$

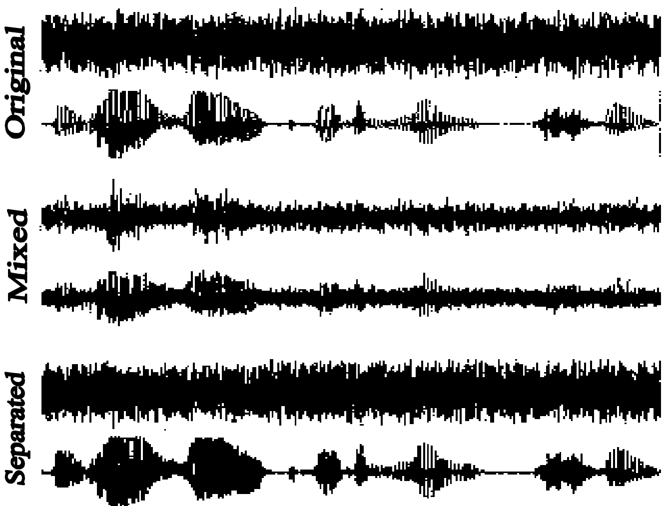

Fig. 6: Experiment results for non-linear mixing of noise and speech.

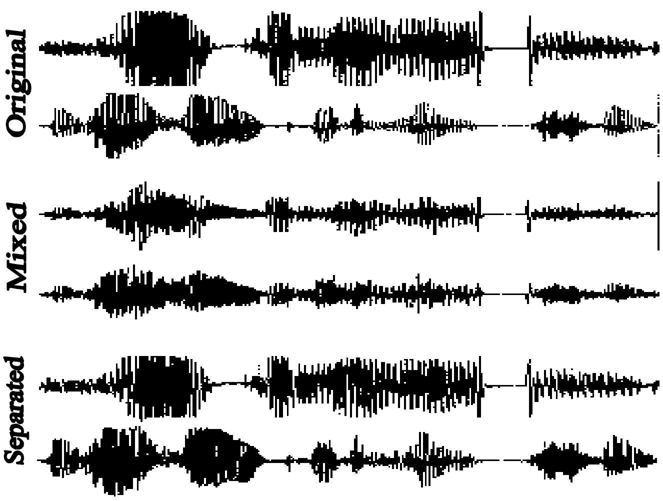

Fig. 7: Experiment results for non-linear mixing of two speech signals.

6. CONCLUSIONS

A neural network model for blind demixing of nonlinear mixtures based on flexible B-spline activation function neuron, has been proposed.

Based on a gradient ascent method on the basis of the maximization entropy criterion, a suitable learning algorithm of parameters of the nonlinear separating model has been derived.

Although, from theoretical point of view the only adaptation of the nonlinear functions at the input layer of the demixing model is sufficient [11], (some experiments are not reported in this paper), better performance are reached by the adaptation of both input and output layer nonlinear functions.

7. REFERENCES

[1] A.J.Bell, T.J.Seinowski, "An information-maximization approach to blind separation and blind deconvolution", Neural Computation vol.7 pag.1129-1159, 1995.

[2] S.I. Amari, A. Cichocki, "Adaptive Blind Signal Processing - Neural Network Approaches", Proceedings of IEEE, Vol. 86, No. 10, Oct. 1998.

[3] A. Pierani, F. Piazza, M. Solazzi, A. Uncini, "Low Complexity Adaptive Non-Linear Function for Blind Signal Separation", Proc. of IEEE IJCNN2000, Como (Italy), July 2000.

[4] S. Guarnieri, F. Piazza and A. Uncini, "Multilayer Feedforward Networks with Adaptive Spline Activation Function", IEEE Trans. on Neural Network, Vol. 10, No. 3, pp.672-683, May 1999.

[5] A.Uncini, L.Vecci, P.Campolucci, F.Piazza, "Complex-valued neural networks with adaptive splines activation function", IEEE Trans. On Signal Processing, vol.47 N°2 Feb.1999.

[6] Y.Tan, J. Wang, "Nonlinear Blind Separation Using an RBF Network Model", Proc. of IEEE ISCAS 2000, Geneva, Switzerland, vol.3,pp.634-637, May 28-31, 2000.

[7] Burel, G. "Blind separation of sources: A nonlinear neural algorithm" Neural Networks, Vol. 5, no. 6, pp 937-947, 1992.

[8] A. Hyvarinen and P. Pajunen, "Nonlinear independent component analysis: existence and uniqueness results", Neural Networks, vol. 12, no. 2, pp. 209-219, 1999.

[9] Lee, T.-W., Koehler, B., and Orglmeister, R. "Blind source separation of nonlinear mixing models", Neural networks for Signal Processing VII, 1997.

[10] Yang, H. H., Amari, S.-I., and Cichocki, A., "Information-theoretic approach to blind separation of sources in non-linear mixture" Signal Processing, Vol. 64 no.3, pp 291-300, 1998.

[11] A.Taleb and C. Jutten "Source separation in Post-NonLinear Mixtures" IEEE Trans. On Signal Processing, vol.47 N°10 pp. 2807-2820, Ott.1999.

[12] A. Cichocki, R. Unbehauen, "Robust neural networks with on-line learning for blind identification and blind separation of sources", IEEE Transaction on Circuits and Systems-I: Fundamental Theory and Applications, Vol.43, Nov. 1996, pp.894-906.