A Neural Architecture for Blind Source Separation

Ernesto Tapia and Raúl Rojas

Technical Report B-06-04

Freie Universität Berlin, Institut für Informatik
Takustr. 9, 14195 Berlin, Germany
tapia,rojas@inf.fu-berlin.de

May 2006

Abstract

A neural architecture based on linear predictability is used to separate linear mixtures of signals. The architecture is divided in two parameterers groups, one modeling the linear mixture of signals and the other computing the linear predictions of the reconstructed signals. The network weights correspond to the mixing matrices and coefficients of the linear predictions, while the values computed by the network units correspond to the predicted and reconstructed signal values. A quadratic error is iteratively minimized to approximate the mixing matrix and to maximize the linear predictability. Experiments with toy and acoustic signals show the feasibility of the architecture.

1 Introduction

The *blind source separation* (*BSS*) problem consists on recovering a set of *source signals* $\boldsymbol{s}(\tau) = (s_1(\tau), \ldots, s_m(\tau))^{\mathsf{T}}$ from a set of *mixtures* $\boldsymbol{x}(\tau) = (x_1(\tau), \ldots, x_n(\tau))^{\mathsf{T}}$ formed with a *mixing matrix* $\boldsymbol{A}$:

$$\boldsymbol{x}(\tau) = \boldsymbol{A}^{\mathsf{T}}\boldsymbol{s}(\tau), \tag{1}$$

where $\tau \in \mathcal{T}$ is an index representing temporal or spatial variation of the signals. The term blind means that the values of the mixing matrix $\boldsymbol{A}$ and the source signals $\boldsymbol{s}(\tau)$ are *unknown*.

The BSS problem is solved by finding an *unmixing matrix* $\boldsymbol{W}$ to reconstruct the sources via the transformation

$$\boldsymbol{y}(\tau) = \boldsymbol{W}^{\mathsf{T}}\boldsymbol{x}(\tau), \tag{2}$$

such that

$$\boldsymbol{y}(\tau) = \boldsymbol{D}\boldsymbol{P}\boldsymbol{s}(\tau), \tag{3}$$

where $\boldsymbol{D}$ is a diagonal matrix, and $\boldsymbol{P}$ is a permutation matrix. This means that the reconstructed signals do not keep the original order of the source signals but their "wave" form.

A general approach to solve the BSS problem is *assuming* that the source signals $s_i(\tau)$ satisfy a property P, and that they minimize (maximize) a measure $q(\boldsymbol{s})$ related to the property P. Thus, the BSS problem is yet regarded as an optimization problem: the unmixing matrix $\boldsymbol{W}$ is an optimal parameter used to transform linearly the mixtures $\boldsymbol{x}(\tau)$ into the signals $\boldsymbol{y}(\tau)$, which minimizes (maximizes) the "quality" of the reconstructed signals $q(\boldsymbol{y}(\tau)) = q(\boldsymbol{W}^{\mathsf{T}}\boldsymbol{x}(\tau))$.

In particular, many researchers use the described approach within a statistical framework [5]. They consider the signals as data drawn from an (unknown) probability distribution, which satisfies some statistical property. One of the best known assumptions is that the source signals $s_i(\tau)$ are (*mutually*) *independent*. The matrix $\boldsymbol{W}$ is estimated as the parameter which yields the minimal mutual information between the variables $y_i(\tau)$. The mutual information is an statistical measure which takes non-negative values and is zero for the case independent variables. Other widely used assumption is that the sources $s_i(\tau)$ have *non-gaussian* probability distributions. Under this assumption, the measure which is maximized depends on a quadratic error between the distributions of the signals $y_i(\tau)$ and multivariate Gaussian distributions. Such measure normally involves some high-order cumulants, such *kurtosis*, which characterize the non-gaussianity of the signals $y_i(\tau)$.

1.1 Maximum Predictability

Another assumption which has received relatively little attention is related with the *predictability* of signals. This assumption is motivated by the property of speech or audio signals to be predicted (approximated) by a linear combination of their values *in the past* [9]. The values in the past are the values of the signal at the *neighborhood* of τ formed by the indexes $\tau_1 = \tau - 1, \tau_2 = \tau - 2, \ldots, \tau_k = \tau - k$.

Motivated by such a property, we define the *prediction* of the signal $y(\tau)$ as the linear combination

$$\widetilde{y}(\tau) = \boldsymbol{\theta}^{\mathsf{T}}\overline{\boldsymbol{y}}(\tau), \tag{4}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)^{\mathsf{T}}$ is the vector of *linear coefficients,* and $\overline{\boldsymbol{y}}(\tau)$ is the vector constructed with the values of y_i at a *neighborhood* $\tau_1, \ldots, \tau_k$ of the index τ:

$$\overline{\boldsymbol{y}}(\tau) = \big(y(\tau_1), \ldots, y(\tau_k)\big)^{\mathsf{T}}. \tag{5}$$

The *maximum predictability* assumption leads to express the solution of BSS problem as the matrix $\boldsymbol{W}$ which optimize a measure involving the *residual*

$$e(y, \boldsymbol{\theta}, k, \tau) = \widetilde{y}(\tau) - y(\tau). \tag{6}$$

Some authors have already used this assumption to solve the BSS problem. For example, Hyvärinen follows the principles of information theory to characterize the predictability of signals [6]. He reduces the BSS problem to the minimization of a function closely related with the *Kolmogorov complexity* of the residual:

$$\widehat{K}(\boldsymbol{W}, \boldsymbol{x}, \boldsymbol{\theta}) = \sum_{\tau \in \mathcal{T}} H\big(e(y_i, \boldsymbol{\theta}, k, \tau)\big), \tag{7}$$

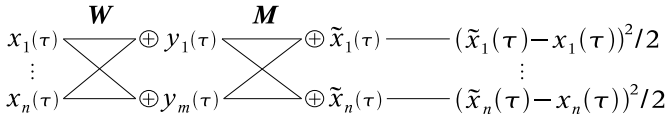

Figure 1: Parameter group in the architecture modeling the unmixing-mixing process.

where the function H is the entropy. He uses gradient descendent schema to iteratively find both the unmixing matrix $\boldsymbol{W}$ and he linear coefficients $\boldsymbol{\theta}$ which minimize an approximation of $\widehat{K}$. His results are closely related with the nongaussianity assumption used in other methods.

Another application of the maximum predictability is the work by Stone [12]. He deals with discrete-time signals and defines a measure of signal predictability

$$F(\boldsymbol{W}, \boldsymbol{x}) = \log \frac{\sum_{\tau \in \mathcal{T}} e(y_i, \boldsymbol{\theta}, k, \tau)^2}{\sum_{\tau \in \mathcal{T}} e(y_i, \boldsymbol{\theta}', k', \tau)^2}, \tag{8}$$

where the coefficients $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are *fixed* beforehand. The number of coefficients k' is typically 100 times larger than k, what means that the sum involving $e(y_i, \boldsymbol{\theta}, k', \tau)$ measures the prediction of $\boldsymbol{y}_i(\tau)$ in a long-term period, while the other sum measures the signal prediction in a short-term period. Stone solves the BSS problem by expressing the minimization of (8) as an generalized eigenvalue problem.

The method presented in this work uses an artificial neural network architecture based on the linear predictability assumption to solve the BSS problem. The network weights correspond to the mixing matrices and the coefficients of the linear approximations, while the values computed by the network units correspond to the signal values. Thus, a quadratic error involving the residual of the linear predictions is minimized iteratively.

The next section describes the network architecture, and the equations used to compute the optimal parameters which minimize the network error.

2 The Neural Network Architecture

The network architecture is divided into two main *parameter groups*. The first parameter group models the *unmixing-mixing process* and uses only the signal values at τ, see Fig. 1. This parameter group has three unit layers formed by the values $x_i(\tau)$, $y_j(\tau)$, and $\widetilde{x}_l(\tau)$, which are connected by the weights W_{ij} and M_{jl}. This is expressed algebraically with the equations

$$\boldsymbol{y}(\tau) = \boldsymbol{W}^{\mathsf{T}} \boldsymbol{x}(\tau), \tag{9}$$

$$\widetilde{\boldsymbol{x}}(\tau) = \boldsymbol{M}^{\mathsf{T}} \boldsymbol{y}(\tau). \tag{10}$$

The last layer in the group computes a quadratic error in terms of the residual

$$e(\boldsymbol{x}, \boldsymbol{M}, \tau) = \widetilde{\boldsymbol{x}}(\tau) - \boldsymbol{x}(\tau). \tag{11}$$

Observe that the minimization of this error means the approximation of the mixing matrix $\boldsymbol{A}$ with the matrix $\boldsymbol{M}$.

However, if we take *any* invertible matrix $\boldsymbol{W}$ and $\boldsymbol{M} = \boldsymbol{W}^{-1}$, the quadratic error involving Eq. (11) is zero. These trivial solutions mean that the architecture does not

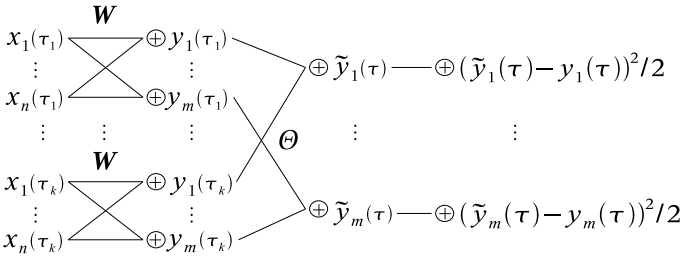

Figure 2: Parameter group in the architecture computing the linear predictions.

model completely the mixing process. In order to avoid the incompleteness of the net we extend the architecture with parameters considering the linear predictability of the reconstructed signals, what is explained below.

The second group computes the *linear approximation model* using the values of the signals at the neighboring indexes $\tau_1, \ldots, \tau_k$. This group uses the unmixing weights $\boldsymbol{W}$ to compute the reconstructed signals at the neighborhood, while the third layer uses the matrix of the linear coefficients $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$ to compute the linear predictions of the reconstructed signals, see Fig. 2. The last layer computes a quadratic error whose minimization stress the linear predictability of the signals $y_i(\tau)$. The architecture of this group corresponds algebraically to the equations

$$\boldsymbol{y}(\tau_l) = \boldsymbol{W}^{\mathsf{T}}\boldsymbol{x}(\tau_l),\ l = 1, \ldots, k, \ \text{ and} \tag{12}$$

$$\widetilde{y}_i(\tau) = \boldsymbol{\theta}_i^{\mathsf{T}}\overline{\boldsymbol{y}}_i(\tau). \tag{13}$$

The complete network architecture modeling the BSS problem is constructed by *connecting* both parameter groups. The new elements added to this architecture are m connections with a constant weight, whose value is minus one. These elements connect the units $y_i(\tau)$ of the first group to the error layer of the second group. See Fig. 3. The last layer integrates the network errors of both parameter groups into the final network error:

$$E(\boldsymbol{W}, \boldsymbol{M}, \boldsymbol{\Theta}, \tau) = \frac{1}{2}\|\widetilde{\boldsymbol{x}}(\tau) - \boldsymbol{x}(\tau)\|^2 + \frac{1}{2}\|\widetilde{\boldsymbol{y}}(\tau) - \boldsymbol{y}(\tau)\|^2. \tag{14}$$

Finally, the partial derivatives corresponding to the architecture are

$$\frac{\partial E}{\partial M_{ij}} = y_i(\tau)\big(\widetilde{x}_j(\tau) - x_j(\tau)\big), \tag{15}$$

$$\frac{\partial E}{\partial \Theta_{ij}} = y_j(\tau_i)\big(\widetilde{y}_j(\tau) - y_j(\tau)\big), \ \text{and} \tag{16}$$

$$\begin{aligned}\frac{\partial E}{\partial W_{ij}} = x_i(\tau)\Big(\sum_{l=1}^{n} M_{jl}(\widetilde{x}_l(\tau) - x_l(\tau)) - (\widetilde{y}_j(\tau) - y_j(\tau))\Big) \\ + \sum_{l=1}^{k} x_i(\tau_l)\Theta_{jl}(\widetilde{y}_j(\tau) - y_j(\tau)).\end{aligned} \tag{17}$$

The next section gives some experimental results obtained by the application of this architecture.

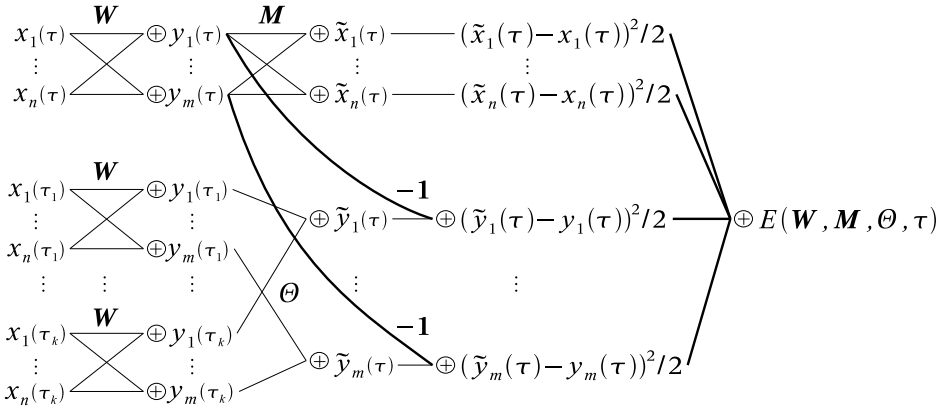

Figure 3: The complete network architecture modeling the BSS problem. The thicker lines represent the constant weights which connect the two parameters groups.

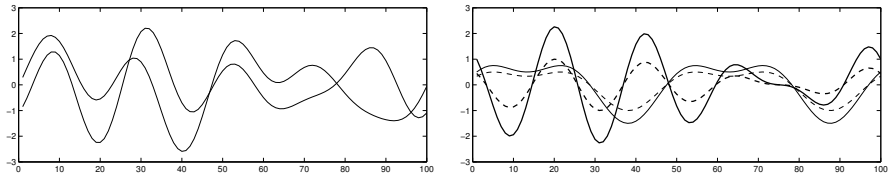

Figure 4: Toy signals used in the experiments. Left: Mixed signals. Right: Original signals (solid line) and their reconstruction (dotted lines).

3 Experimental Results

We experimented with two type of signals: sinusoidal functions and audio data. In all cases, we mixed two signals using a random matrix, whose entries were generated from a normal density with zero mean and standard deviation one. The sinusoidal functions are

$$s_1(\tau) = \sin(\alpha) + 0.5\cos(2\alpha), \tag{18}$$
$$s_2(\tau) = \cos(2.5\alpha) - 1.3\sin(2\alpha), \tag{19}$$

with $\tau = 1, \ldots, 100$ and $\alpha = 4\pi\tau/100$. See figure 4. The audio data have a duration of ten seconds and a frequency of 8MHz. They are recordings of a masculine voice and applause, see Figs. 5-6.

The network error was minimized using a gradient-based schema. The optimal parameters $p_i^{(t)}$ of the network error $E(\boldsymbol{p}, \tau)$ are computed at the iteration $t+1$ using the formula

$$p_i^{(t+1)} = p_i^{(t)} + \Delta p_i^{(t)}. \tag{20}$$

The increment $\Delta p_i^{(t)}$ where computed using the RPROP algorithm [10]. RPROP is an adaptive step algorithm which updates the increment using the sign of the the average partial derivatives of the network error. The average of partial derivatives was computed using *batches* with two to four hundred elements. The elements of batches

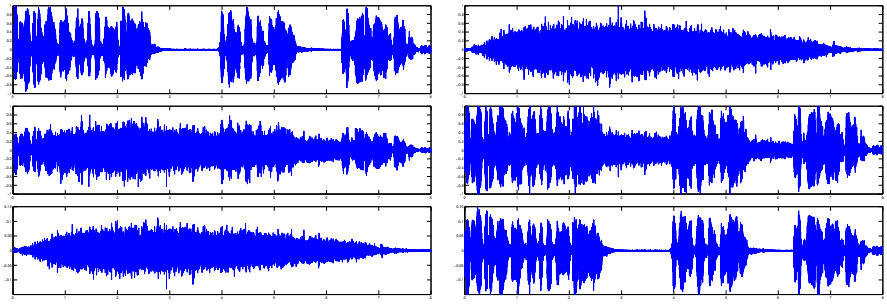

Figure 5: Wave signals used in the experiment. Top: The original signals are voice and applause. Middle: The mixed signals. Bottom: The reconstructed signals.

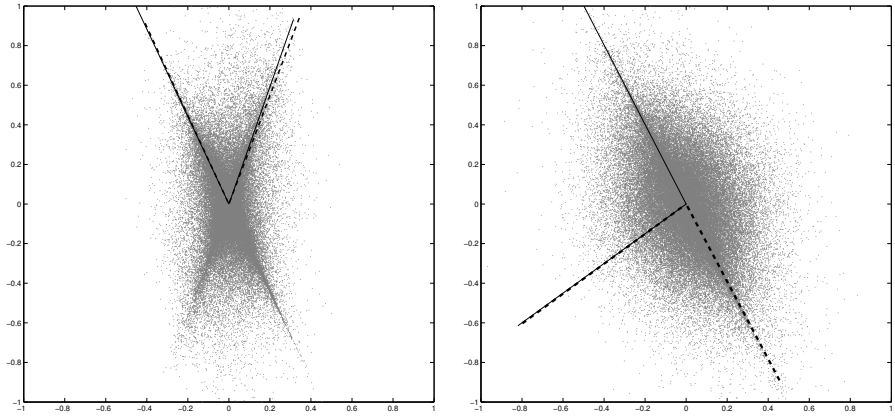

Figure 6: Scatterplot representing mixed audio signals: The dotted segments are the original independent components and the continuous segments are the found components. Left: voice-applause mix. Right: another example of mixed music-applause audio signals.

were selected randomly from the training set, and the parameters were updated using *simulated annealing.*

The accuracy of the solutions were measured using the following property: if the unmixing matrix $\boldsymbol{W}$ solves the BSS problem, the matrix $\boldsymbol{P} = \boldsymbol{AW}$ is a permutation matrix [4]. Thus, our quality function is defined as

$$Q(P) = \sum_{i=1}^{n} \Big(\sum_{j=1}^{n} \frac{|P_{ij}|}{\max_l |P_{il}|} - 1 \Big) + \sum_{i=1}^{n} \Big(\sum_{j=1}^{n} \frac{|P_{ij}|}{\max_l |P_{lj}|} - 1 \Big). \qquad (21)$$

Note that this function is always nonnegative for all P, and it is zero if P is a permutation matrix.

Figure 7 shows how the quality $Q(P_t)$ evolves respect to he number of iterations t. The scatterplot shown in the figure corresponds to the solution of the BBS problem on the audio signals, taken from then runs of the backpropagation algorithm with random initializations.

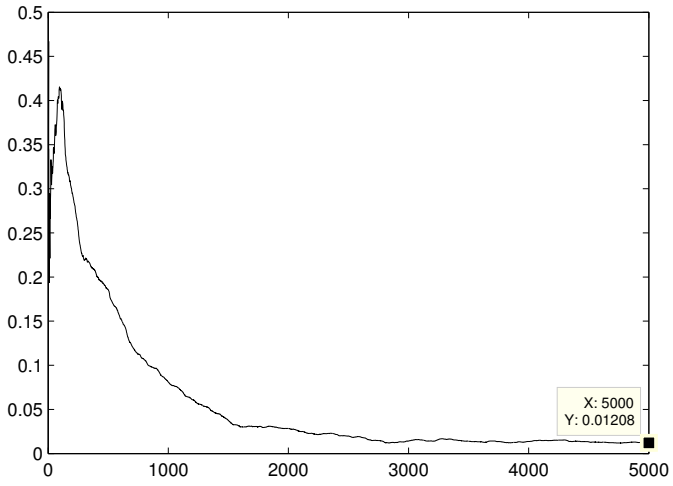

Figure 7: Accuracy of the solutions $Q(P_t)$.

4 Discussion and Further Work

We presented a neural architecture used to solve the blind source separation problem. The assumption used to overcome the inherent lack of information is the linear predictability of signals. Our experiments show good results for the separation of toy and acoustic signals.

One drawback of the method is that it can reach a local minimum. This can be overcome by running the algorithm several times, and using stochastic learning. Other drawback in the method were found when we mixed more than three audio signals. In this case, some reconstructed signals were a multiple of another, i.e. $y_i = \alpha y_j$ with $i \neq j$ and $\alpha \neq 0$. This can be interpreted as a local minimum, where the overall linear approximation is minimized.

Despite these mentioned drawbacks we think that one of the good characteristics of our method is the simplicity of the network architecture its quadratic error. Other good characteristic is that the architecture can be interpreted and extended in several ways.

For example, after some algebra we can express the partial derivative (17) as

$$\frac{\partial E}{\partial W_{ij}} = x_i(\tau) \sum_{l=1}^{n} M_{jl}(\widetilde{x}_l(\tau) - x_l(\tau)) + e(x_i, \boldsymbol{\theta}_j, k, \tau) \cdot e(y_j, \boldsymbol{\theta}_j, k, \tau). \qquad (22)$$

Interestingly, the left element of (22) is the product between the linear residual of $y_j(\tau)$ and the residual of the linear prediction of $x_i(\tau)$ using the linear coefficient $\boldsymbol{\theta}_j$. This can be interpreted as a kind of Hebbian learning, where the residuals are memorized by $\boldsymbol{W}$ during the iterative update of the parameters. This reflects the influence of the two parameter groups for the calculation of the unmixing matrix $\boldsymbol{W}$.

An extension (or simplification) of the architecture can be done when the layer used to approximate $\boldsymbol{M}$ is eliminated. This corresponds to the minimization of the quadratic error

$$E(\boldsymbol{W}, \boldsymbol{\Theta}) = \frac{1}{2} \sum_{\tau=1}^{\mathsf{T}} \|\widetilde{\boldsymbol{y}}(\tau) - \boldsymbol{y}(\tau)\|^2. \qquad (23)$$

We can interpret this error function as a simple projection pursuit method: the original signals are projected to the components which have the best linear approximation. We think this new architecture can lead to new and interesting results, although we did not experiment with it.

References

[1] A. J. Bell and T. J. Sejnowski. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[2] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, February 1997.

[3] N. Delfosse and P. Loubaton. Adaptive Blind Separation of Convolutive Mixtures. In *Proceedings of the 29th Asilomar Conference on Signals, Systems and Computers (2-Volume Set)*, 1995.

[4] X. Giannakopoulos, J. Karhunen, and E. Oja. An Experimental Comparison of Neural Algorithms for Independent Component Analysis and Blind Separation. *International Journal of Neural Systems*, 9(2):99–114, 1999.

[5] A. Hyvärinen. Survey on Independent Component Analysis. *Neural Computing Surveys*, pages 94–128, 1999.

[6] A. Hyvärinen. Complexity Pursuit: Separating Interesting Components from Time Series. *Neural Computation*, 13(4):883–898, 2001.

[7] J. Karhunen and P. Pajunen. Blind Source Separation Using Least-Squares Type Adaptive Algorithms. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97)*, 1997.

[8] B. Pearlmutter and L. Parra. A Context-Sensitive Generalization of ICA. In *Proceedings of the International Conference on Neural Information Processing*, 1996.

[9] L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, 1978.

[10] M. Riedmiller and H. Braun. RPROP – Description and Implementation Details. Technical report, Universität Karlsruhe, 1994.

[11] R. Rojas. *Neural Networks – A Systematic Introduction*. Springer, Berlin, 1996.

[12] J. V. Stone. Blind Source Separation Using Temporal Predictability. *Neural Computation*, 13(7):1559–1574, 2001.