1995 International Symposium on Nonlinear Theory and its Applications, pp.37-42,
NOLTA'95, Las Vegas, December, 1995.

Recurrent Neural Networks For Blind Separation of Sources

S. Amari †*, A. Cichocki** and H. H. Yang*

†University of Tokyo,Tokyo 113, Japan, amari@sat.t.u-tokyo.ac.jp
** ABS Lab., FRP, RIKEN, Wako-Shi, Saitama, 351-01, Japan, cia@kamo.riken.go.jp
* IR Lab., FRP, RIKEN, Wako-Shi, Saitama, 351-01, Japan, hhy@koala.riken.go.jp

ABSTRACT

In this paper, fully connected recurrent neural networks are investigated for blind separation of sources. For these networks, a new class of unsupervised on-line learning algorithms are proposed. These algorithms are the generalization of the Hebbian/anti-Hebbian rule. They are not only biologically plausible but also theoretically sound. An important property of these algorithms is that the performance of the networks is independent of the mixing matrix and the scaling factor of the input sources. This property is verified by analyses and simulations.

I. Introduction

The problem of multi-channel blind separation of sources such as the "cocktail-party" problem arises in diverse fields in neural computation (including hearing and olfactory systems) and in applied science (including radar, speech processing and digital communication). The problem is how to separate signal sources from the sensor output in which the sources are mixed in an unknown channel, a multiple-input multiple-output linear system. The problem is to recover the original waveforms of the sources. In the "cocktail-party" problem, a person who wants to listen to a single speaker or locate a sound source must filter out noise and interferences such as other people's voice and echoes. An under water sonar system has to solve a similar problem in order to recognize a target.

The area of blind separation is closely related to the modeling and signal processing problems in neuroscience. Animals and human beings have remarkable capabilities to localize sounds and to recognize speeches. They can also recognize different odors in a complicated environment by their olfactory systems. However, the detailed functional behavior of the olfactory system is still an open problem [7, 8]. It has been a challenge for both theoreticians and engineers to design some neural networks and associated adaptive learning algorithms to separate and localize unknown odor sources from mixing odors.

One interesting and challenging application of the blind separation of sources is the analysis of electroencephalographic (EEG) data. The problem is to separate and localize meaningful sources in the brain based on EEG data. This problem is mathematically underdetermined. However, an appropriately selected adaptive learning algorithm is able to extract independent sources from highly correlated EEG signals. Some preliminary but promising results have been reported in [12].

Most of the approaches to the blind separation of sources are based on the concept of Independent Component Analysis (ICA) which is an extension or generalization of Principal Component Analysis (PCA) [2, 4, 14]. Based on this concept and assumed that the source signals $s_i(t)$ are statistically independent, the source separation can be achieved successful when the output signals of the separation network (the inverse system) become independent. It should be noted here that the assumption of the independence on the sources is sufficient but not necessary. Comon(1994)[4] defines ICA in terms of a contrast function which essentially measures the degree of independence among the outputs. The contrast function is defined as the Kullback-Leibler divergence between the joint and marginal distributions of the outputs. To evaluate the contrast function, either Gram-Charlier expansion [1] or Edgeworth expansion [4] can be used. One method is to use high order statistics (HOS) to implement the ICA. One disadvantage of this method is that it is usually computationally very intensive and may be inaccurate when the cumulants which are higher than 4-th order are neglected. It seems the HOS approach is not plausible for biological systems since it violates two fundamental principals: locality and simplicity.

Our goal is to develop a class of efficient on-line adaptive learning algorithms which can be easily hardwired on neural networks and can automatically search for synaptic weights.

II. Basic Model and Learning Algorithm

Let us consider n unknown source signals: $s_i(t), i = 1, \cdots, n$. The model for the sensor output is

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

where $\mathbf{A}$ is an unknown non-singular mixing matrix,

$$\mathbf{s}(t) = (s_1(t), \cdots, s_n(t))^T,$$

$$\mathbf{x}(t) = (x_1(t), \cdots, x_n(t))^T$$

and $(.)^T$ denotes the transpose of a vector.

To recover the unknown sources, we use the following linear recurrent neural network which takes the sensor output as its input:

$$\tau_i \frac{dy_i}{dt} + y_i = x_i(t) - \sum_{i=1}^{n} \widehat{w}_{ij}(t) y_j \qquad (1)$$

where τ_i are time constants and $\widehat{w}_{ij}$ are synaptic weights. The model (1) can be put into the following compact form:

$$\mathcal{T} \frac{d\mathbf{y}}{dt} + \mathbf{y} = \mathbf{x}(t) - \widehat{\mathbf{W}}(t)\mathbf{y}$$

where

$\mathcal{T} = \mathrm{diag}(\tau_1, \cdots, \tau_n)$ and
$\mathbf{y}(t) = (y_1(t), \cdots, y_n(t))^T$.

After a short transience, we get a so-called adiabatic approximation

$$\mathbf{y}(t) = (\mathbf{I} + \widehat{\mathbf{W}}(t))^{-1}\mathbf{x}(t) \qquad (2)$$

under the condition that all eigenvalues of the matrix $\mathbf{I} + \widehat{\mathbf{W}}(t)$ have positive real parts. This condition often holds in our simulations even for ill-conditioned mixing matrix. In order to simplify the analysis, we take the adiabatic approximation directly as the output of the network (1). This type of neural network for signal separation has been used by many researchers such as Jutten-Herault (1991)[9], Hopfield (1991)[8] and Matsuoka et al (1995)[13]. However, all these researchers did not use self-inhibitory connections in their models. This is also the case in the novelty filter described by Kohonen (1984)[10] and de-correlating network proposed by Barlow and Földiák (1989) [6] and Földiák (1989)[5].

In contrast to these models, our network is fully connected with self-inhibitory connections. We shall show that these self-loops play an essential role in improving the performance of the network in separating sources.

For the model (1), we have developed the following on-line learning algorithm:

$$\begin{aligned} \frac{d\widehat{w}_{ii}}{dt} = & \quad -\mu(t)\{\lambda_i(\widehat{w}_{ii}(t) + 1) - [f(y_i(t)) \\ & \quad + \sum_{k=1}^{n} \widehat{w}_{ik}(t) f(y_k(t))] g_i(y_i(t))\}, \qquad (3) \end{aligned}$$

for $i \neq j$,

$$\begin{aligned} \frac{d\widehat{w}_{ij}}{dt} = & \quad -\mu(t)\{\lambda_i(\widehat{w}_{ij}(t) - [f(y_i(t)) \\ & \quad + \sum_{k=1}^{n} \widehat{w}_{ik}(t) f(y_k(t))] g_j(y_j(t))\} \qquad (4) \end{aligned}$$

where $\lambda_i > 0$ are scaling factors (typically $\lambda_i = 1$), $\mu(t) > 0$ is a learning rate function (usually exponentially decreased to zero), and $f(y)$ and $g(y)$ are two odd activation functions. Two typical choices for the activation functions are:

1. $f(y) = y^3$ and $g(y) = y$;
2. $f(y) = y$ and $g(y) = \tanh(10y)$.

The algorithm (4) can be put into a compact matrix form as following:

$$\frac{d\widehat{\mathbf{W}}}{dt} = -\mu(t)[\mathbf{I} + \widehat{\mathbf{W}}][\mathbf{\Lambda} - \mathbf{f}(\mathbf{y}(t))\mathbf{g}^T(\mathbf{y}(t))] \qquad (5)$$

where

$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$ with positive entries,

typically, $\mathbf{\Lambda} = \mathbf{I}$,

$\mathbf{f}(\mathbf{y}) = (f(y_1), f(y_2), \cdots, f(y_n))^T$,

$\mathbf{g}(\mathbf{y}) = (g(y_1), g(y_2), \cdots, g(y_n))^T$.

When an auxiliary inter-neuron layer is incorporated to generate

$$\mathbf{z}(t) = \mathbf{f}(\mathbf{y}(t)) + \widehat{\mathbf{W}}(t)\mathbf{f}(\mathbf{y}(t)),$$

the algorithm (5) can be implemented simply by the following algorithm:

$$\frac{d\widehat{\mathbf{W}}}{dt} = -\mu(t)[\mathbf{\Lambda} + \widehat{\mathbf{W}}\mathbf{\Lambda} - \mathbf{z}(t)\mathbf{g}^T(\mathbf{y}(t))]. \qquad (6)$$

This algorithm can be considered as a generalization of the anti-Hebbian rule. To compute $\mathbf{y}(t)$, it is not necessary to compute the inverse of the matrix $\mathbf{I} + \widehat{\mathbf{W}}(t)$ explicitly. Instead of using (2), we use the the following recursive relation to compute $\mathbf{y}(t)$:

$$\mathbf{y}(t) = \mathbf{x}(t) - \widehat{\mathbf{W}}(t)\mathbf{y}(t - \tau)$$

where τ is a small delay. Figure 1 below illustrates the implementation of our algorithm.

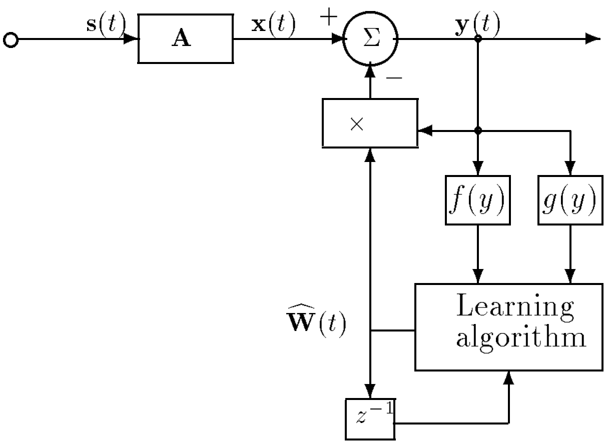

Figure 1: A Functional block diagram of the separation network

III. Extensions of The Basic Learning Algorithm

There are many possible extensions of the learning algorithm (5). A general form of the learning algorithm is the following:

$$\frac{d\widehat{\mathbf{W}}}{dt} = -\mu(t)(\widehat{\mathbf{W}} + \mathbf{I})\mathbf{G}(\mathbf{y}(t)) \qquad (7)$$

where the matrix $\mathbf{G}(\mathbf{y})$ can take various forms such as:

$$\begin{aligned}
\mathbf{G}_1(\mathbf{y}) &= \Lambda - \mathbf{f}(\mathbf{y})\mathbf{g}^T(\mathbf{y}),\\
\mathbf{G}_2(\mathbf{y}) &= \Lambda - \mathbf{y}\mathbf{y}^T - \mathbf{f}(\mathbf{y})\mathbf{g}^T(\mathbf{y}) + \mathbf{g}(\mathbf{y})\mathbf{f}^T(\mathbf{y}),\\
\mathbf{G}_3(\mathbf{y}) &= \Lambda - \mathbf{f}(\mathbf{y})\mathbf{g}^T(\mathbf{y}) - \mathrm{diag}\{\mathbf{y}\mathbf{y}^T - \mathbf{f}(\mathbf{y})\mathbf{g}^T(\mathbf{y})\},\\
\mathbf{G}_4(\mathbf{y}(t)) &= \Lambda - \sum_{k=0}^{p} \mathbf{y}(t)\mathbf{y}^T(t-kT), \quad p = 1,2,3.
\end{aligned}$$

Note the above continuous-time algorithm can be easily transformed to a discrete-time (iterative) algorithm as

$$\widehat{\mathbf{W}}(k+1) = \widehat{\mathbf{W}}(k) - \eta(k)[(\widehat{\mathbf{W}}(k) + \mathbf{I})\mathbf{G}(\mathbf{y}(k\Delta t)),$$

$$k = 0, 1, 2, \ldots,$$

where $\eta(k) > 0$ is a sequence of learning rates and Δt is a sample interval.

The choice of the matrix $\mathbf{G}(\mathbf{y})$ may depend on many factors, e.g., distribution of sources, convergence speed, and the complexity of implementations. The idea behind using function $f(y)$ and $g(y)$ is to cancel the higher order (nonlinear) correlations or cross-cumulants. Let us take an expectation of both side of equation (7). If the synaptic weights $<\widehat{\mathbf{W}}(t)>$ approaches a constant matrix, then the term $<\mathbf{G}(\mathbf{y})>$ approaches to zero as t approaches to infinity, i.e., the nonlinear correlations with respect to the function $f(y)$ and $g(y)$ decrease to zero.

Although the algorithm (7) works well in simulations, the theory for this algorithm is not mature. It is not an easy task to derive it rigorously. In [1], we use the Kullback-Leibler divergence between the joint and marginal distributions of the outputs to measure dependency among the output signals. We derived a learning algorithm for the feedforward network by minimizing the Kullback-Leibler divergence. The Gram-Charlier expansion is applied in evaluating the Kullback-Leibler divergence. If the natural gradient of the divergence is used to minimize the divergence, the learning algorithm is the following:

$$\frac{d\mathbf{W}}{dt} = \mu(t)\mathbf{G}(\mathbf{y})\mathbf{W}. \qquad (8)$$

where the function $\mathbf{G}(\mathbf{y})$ is in the following special form:

$$\begin{aligned}
\mathbf{G}(\mathbf{y}) &= \mathbf{I} - \mathbf{f}(\mathbf{y})\mathbf{y}^T,\\
f(y) &= \frac{3}{4}y^{11} + \frac{25}{4}y^9 - \frac{14}{3}y^7 - \frac{47}{4}y^5 + \frac{29}{4}y^3.
\end{aligned}$$

From (8), we can easily obtain the algorithm (7) by using a transform $\mathbf{W} = (\widehat{\mathbf{W}} + \mathbf{I})^{-1}$.

It is still an open question why the algorithm (7) works in simulations for other types of functions such as $\mathbf{G}_i(\mathbf{y})$, $i = 1,2,3,4$.

IV. Equivariant Property and Performance Factor

A major advantage of the algorithm (7) is that the performance of the proposed algorithm does not depend at all on the mixing matrix and the scaling factor of the input sources. To get this property, we put the algorithm (7) into the following form:

$$(\widehat{\mathbf{W}} + \mathbf{I})^{-1}\frac{d\widehat{\mathbf{W}}}{dt} = -\mu(t)\mathbf{G}(\mathbf{y}(t))$$

Hence

$$\frac{d(\widehat{\mathbf{W}} + \mathbf{I})^{-1}}{dt}(\widehat{\mathbf{W}} + \mathbf{I}) = \mu(t)\mathbf{G}(\mathbf{y}(t)).$$

Therefore,

$$\frac{d(\widehat{\mathbf{W}} + \mathbf{I})^{-1}}{dt} = \mu(t)\mathbf{G}(\mathbf{y}(t))(\widehat{\mathbf{W}} + \mathbf{I})^{-1}.$$

Multiplying the above equation by the mixing matrix $\mathbf{A}$ from right on each side of the equation, we have

$$\frac{d\mathbf{P}}{dt} = \mu(t)\mathbf{G}(\mathbf{y}(t))\mathbf{P}, \qquad (9)$$

where $\mathbf{P} = (\widehat{\mathbf{W}} + \mathbf{I})^{-1}\mathbf{A}$. The matrix $\mathbf{P}(t)$ will be called the performance matrix of the learning equation (7). During the learning process the performance matrix tends to a generalized permutation matrix in which each column and each row has one and only one dominant element. To evaluate the performance of the separation algorithm, we define the performance factor associated to the matrix $\mathbf{P}(t) = (p_{ij}) \in \mathbf{R}^{n\times n}$ as following:

$$E_1 = \sum_{i=1}^{n}(\sum_{j=1}^{n} \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1) + \sum_{j=1}^{n}(\sum_{i=1}^{n} \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1)$$

The performance of the separation network is fully determined by the equation (9). Since it does not depend on the mixing matrix $\mathbf{A}$ directly, we can assume $\mathbf{G}(\mathbf{y}(t)) = \mathbf{G}(\mathbf{P}(t)\mathbf{s}(t))$ where the source signals $\mathbf{s}(t)$ have a normalized distribution. This is a very good property especially when the problem is ill-posed or ill-conditioned. It should be noted that our algorithms has the same "equivariant" property as the algorithms developed by Cardoso and Laheld in [2, 11] for feedforward networks.

Another advantage of the algorithm (7) is that the operational range of $\widehat{\mathbf{W}}$ is bounded. Usually,

$$-1 \leq \widehat{w}_{ij} \leq 1$$

hold even for an ill-conditioned mixing matrix. When the synaptic weights are small the system (1) is stable.

V. Simulations

Two simulation examples are given to demonstrate the performance of the algorithm (7). The simulation results are shown in Figures 2-6 located at the end of this paper.

Example 1: Consider the following three unknown sources:

$$\begin{aligned} s_1(t) &= n(t) \\ s_2(t) &= 10^{-3}\cos(400t + 10\sin(90t)) \\ s_3(t) &= 10^{-4} sign[\cos(550t) - 5\sin(99t))] \end{aligned}$$

where $n(t)$ is a Gaussian noise with variance $\sigma^2 = 1$. These three sources are mixed by a mixing matrix

$$\mathbf{A} = \begin{bmatrix} 0.20 & -0.61 & 0.62 \\ 0.91 & -0.89 & -0.33 \\ 0.59 & 0.76 & 0.60 \end{bmatrix}$$

Choose $f(y) = y^3$, $g(y) = y$, and $\mu(t) = 1000\exp(-15t)$ as the learning rate function in the algorithm (5). The dynamics of both $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are shown in Figure 2. The first three signals denoted by X1, X2 and X3 represent mixing (sensor) signals: $x_1(t)$, $x_2(t)$ and $x_3(t)$. The last three signals denoted by O1, O2 and O3 represent the output signals: $y_1(t)$, $y_2(t)$, and $y_3(t)$. When the algorithm is convergent, the performance matrix $\mathbf{P}(t)$ becomes a generalized permutation matrix in which each column has one and only one dominant element. We define the performance factor as the summation of the all absolute values of the elements in $\mathbf{P}(t)$ except those dominant elements.

The performance factor for the algorithm (7) and the evolution of the synaptic weights $\widehat{w}_{ij}$ are shown in Figure 3.

A feedforward network is proposed in [3] and [2] for blind separation. The following algorithm is proposed in [3] to find the de-mixing matrix $\mathbf{W}$:

$$\frac{d\mathbf{W}}{dt} = \mu(t)[\mathbf{I} - \mathbf{f}(\mathbf{y}(t))\mathbf{g}^T(\mathbf{y}(t))]\mathbf{W}. \qquad (10)$$

Choose the same activation function as those in our recurrent network. The performance factor for the algorithm (10) and the evolution of the weights w_{ij} are shown in Figure 4.

Comparing Figure 3 with Figure 4, we have the following observations:

1. The performance factor for our recurrent network is generally smaller than the one for the feedforward network.

2. Although the separation times of the two networks are comparable, the operational range for the synaptic weights $\widehat{w}_{ij}$ in the recurrent network is much smaller than the one for the synaptic weights w_{ij} in the feedforward network.

Example 2: Let the three unknown sources be the following:

$$\begin{aligned} s_1(t) &= n(t) \\ s_2(t) &= 10^{-9}\sin(900t) * sin(60t) \\ s_3(t) &= 10^{-6}\sin(234t). \end{aligned}$$

Assume the noise $n(t)$ and the mixing matrix are the same as those in **Example 1**.

Choose $f(y) = y$, $g(y) = tanh(10y)$, and a constant learning rate $\mu(t) = 100$. The simulation result is shown in Figure 5. The performance factor and the evolution of the synaptic weights $\widehat{w}_{ij}$ are shown in Figure 6.

VI. Conclusions

We have developed a general learning algorithm to train a recurrent network for blind separation of sources. Like the feedforward network, the recurrent network also has equivariant property, i.e., the performance of these networks is independent of the mixing matrix and the scaling factor of the input sources. In contrast to the feedforward network, the recurrent network needs a much smaller operational range for the synaptic weights. Therefore, in the implementation of the separation networks, the hardware requirement for the recurrent network is less than the one for the feedforward network.

Although the general algorithm (7) can be justified for some special forms of the function $\mathbf{G}(\mathbf{y})$, it is still an open problem to derive the algorithm rigorously for other forms of $\mathbf{G}(\mathbf{y})$ working well in simulations.

The recurrent network model proposed in this paper can be extended to a model in which each output is a mixture of delayed sources. For this model with delays, we can use an algorithm similar to the algorithm (7) for blind source separation when the delays are known.

Acknowledgment

We would like to thank Dr. Xiao Yan SU for the proofreading of the manuscript.

References

[1] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems, 8, eds. David S. Touretzky, Michael C. Mozer and Michael E. Hasselmo, MIT Press: Cambridge, MA. (to appear)*, 1996.

[2] J.-F. Cardoso and Beate Laheld. Equivariant adaptive source separation. *To appear in IEEE Trans. on Signal Processing*, 1996.

[3] A. Cichocki and R. Unbehauen. *Neural Networks for Optimization and Signal Processing (Second Edition)*. John Wiley, New York, 1994.

[4] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[5] P. Földiák. Adaptive network for optimal linear feature extraction. In *Proc. IEEE/INNS Int. Joint Conf. Neural Net.*, volume 1, pages 401–405, 1989.

[6] H. B. Barlow and P. Földiák. Adaptation and decorrelation in the cortex. In C. Miall, R. M. Durbin, and G. J. Mitchison, editors, *The computing neuron*, pages 54–72. Addison-Wesley, New York, 1989.

[7] O. Hendin, D. Horn, and J. J. Hopfield. Decomposition of a mixture of signals in a model of the olfactory bulb . *Proc. Natl. Acad. Sci. USA*, 91:5942–5946, June 1994.

[8] J. J. Hopfield. Olfactory computation and object perception . *Proc. Natl. Acad. Sci. USA*, 88:6462–6466, August 1991.

[9] C. Jutten and J. Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[10] T. Kohonen. *Self-organization and Associative Memory.* Springer-Verlag, Heidelberg, Germany, 1984.

[11] B. Laheld and J.-F. Cardoso. Adaptive source separation without prewhitening. In *Proc. EUSIPCO, Edinburgh*, pages 183–186, September 1994.

[12] S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski. Independent component analysis of electroencephalographic data. Technical report, Naval Health Research Center and The Salk Institute, 1995.

[13] K. Matsuoka, M. Ohya, and M. Kawamoto. A neural net for blind separation of nonstationary signals. *Neural Networks*, 8(3):411–419, 1995.

[14] L. Wang, J. Karhunen, and E. Oja. A bigradient optimization approach for robust PCA, MCA and source separation. In *ICNN'95*, November/December 1995.

Figures:

(Note: Plots for Figure2-6 are in a separate tared file called noltaFigs.tar)

Figure 2: The mixed signals versus the separated signals in **Example 1**

Figure 3: The performance factor and the evolution of the synaptic weights $\widehat{w}_{ij}$ in **Example 1**

Figure 4: The performance factor and the evolution of the synaptic weights w_{ij} for the algorithm with the feedforward network in **Example 1**

Figure 5: The mixed signals versus the separated signals in **Example 2**

Figure 6: The performance factor and the dynamics of the synaptic weight $\widehat{w}_{ij}$ in **Example 2**