ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 261-266

A Linear Adaptive Neural Network for Extraction of Independent Components

Zied Malouche , Odile Macchi

Laboratoire des Signaux et Systèmes, CNRS, Supélec
Plateau de Moulon 91192 Gif-sur-Yvette Cedex, France

Abstract. In this paper we introduce a linear adaptive neural network for extracting all independent components contained in a linear mixture. Each neuron of the network is updated with the same extended local anti-Hebbian rule [7] and is capable of extracting one component. However the adequate initialisations are hard to perform. Therefore we add a second global term in the learning rule of each neuron that involves informations from the other neurons and forces them to extract different components. When there are at least as many observations as components, all the components are extracted, at least by one neuron.

1. Introduction

The Independent Components Analysis (ICA) plays an important role in an increasing range of fields e.g. biology, wireless communications, array processing, etc. Since Hérault and Jutten have intoduced a first adaptive linear feedback neural network solution to this problem [3], many other adaptive solutions have been proposed. The theoretical principles of linear ICA have then been established by Comon [2]. Macchi and Moreau [6] have also studied the learning rule of [3] and applied it to a novel architecture. Cardoso and Laheld [1] proposed an adaptive algorithm whose performances are independent of the mixture. Karhunen and Joutsensalo [5] performed linear adaptive ICA by generalizing the Principal Component Analysis algorithm of Oja [8] after introducing suitable nonlinear functions in the updating. In [7][4] linear adaptive network are proposed. In [7], all the neurons are updated with the same rule. Each neuron is capable of extracting one component whose label depends on initialization. Nevertheless, there is no interaction between all the neurons, so one particular component may not be extracted while other may be recovered many times. In this present contribution the proposed solution is a neural network built with m single neurons that jointly optimize a global criterion. For each neuron, the corresponding adaptive rule is composed of a first local term responsible for the extraction of one component, and a second term involving informations from the other neurons, and contributing in the achievement of perfect extraction.

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 261-266

2. Problem statement

Suppose that m sensors provide m observations x_j which are linear mixtures of p independent zero-mean components a_i according to the model

$$\mathbf{x} = \mathbf{Ga} \tag{1}$$

where $\mathbf{a}$ is the $p \times 1$ vector of components a_i, $\mathbf{G}$ the $m \times p$ fixed mixture matrix and $\mathbf{x}$ the $m \times 1$ vector of observations x_j. It is assumed that

$$m \geq p \tag{2}$$

i.e. we have enough sensors, and that $\mathbf{G}$ is full rank

$$\text{rank}(\mathbf{G}) = p. \tag{3}$$

Taking indeterminacies in (1) into account, it is not a loss of generality to assume that the components have unit power

$$E\{a_i^2\} = 1 \ , \ i = 1, \ldots, p. \tag{4}$$

The problem is unsupervised when $\mathbf{a}$ and $\mathbf{G}$ are all unknown and only $\mathbf{x}$ is

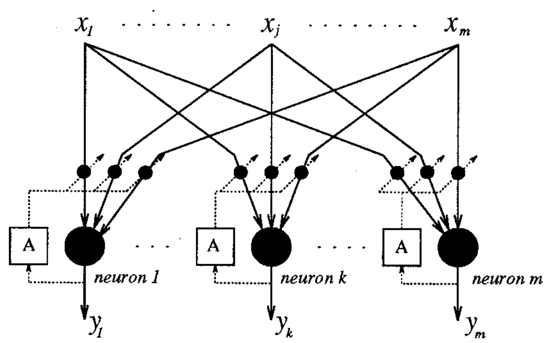

Figure 1: Neural network with local updating

observed. The neural network in Fig.1 is intended to extract one source a_i on each output y_k. In [7], a single linear neuron, with weight vector $\mathbf{h}_k = (h_{1k}, \ldots, h_{mk})^T$, computing

$$y_k = \mathbf{h}_k^T \mathbf{x} \tag{5}$$

is adaptively trained with the local extended anti-hebbian rule

$$\Delta \mathbf{h}_k = -\mu(y_k^3 - y_k)\mathbf{x}. \tag{6}$$

The rule (6), which aims at minimizing the average square distance $(y_k^2 - 1)^2$, is a stochastic gradient way to minimize the criterion

$$I(\mathbf{h}_k) = E\{y_k^4 - 2y_k^2\}. \tag{7}$$

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 261-266

This method is capable of isolating one of the components a_i on the output y_k, provided a_i has a negative kurtosis. The adaptation (6) is local as it only uses the input vector $\mathbf{x}$ and the output y_k of the considered k-th neuron (see Fig.1). Clearly if the weights of neurons k and l have identical initial values, they deliver $y_k = y_l$ and restore the same component a_i. Therefore, if it is desired that all components be extracted, some kind of coupling between the various weights $\mathbf{h}_k$ has to be introduced. Hence the adaptation cannot remain local (see Fig.2).

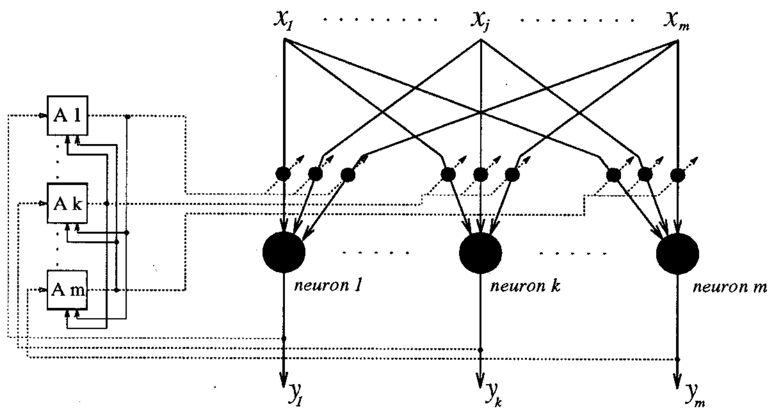

Figure 2: Neural network with interacting neurons updating

3. Criterion and algorithm

It is not possible to ensure that all components will be extracted at least by one neuron whithout loosing the local neuron character of the rule. For this sake we propose to jointly minimize the m local criteria $I(\mathbf{h}_k)$ as well as an interaction (global) term that involves all the m neuron weight vectors $\mathbf{h}_k$ through the $m \times m$ matrix $\mathbf{H}$ whose columns are the weight vectors $\mathbf{h}_k$. The necessary condition to extract the p components is that the overall matrix

$$\mathbf{C} = \mathbf{G}^T\mathbf{H} \tag{8}$$

has the following structure

$$\mathbf{C} = [\mathbf{\Lambda} \;\; \mathbf{R}]\mathbf{P} \tag{9}$$

where $\mathbf{P}$ is an arbitrary $m \times m$ permutation matrix, $\mathbf{\Lambda}$ a $p \times p$ invertible diagonal matrix and $\mathbf{R}$ a $p \times (m-p)$ matrix whose columns contain at most one non zero entry. This is equivalent to saying that p components are extracted by p neurons and the remaining $(m-p)$ neurons (if $m > p$) extract other copies of the p components. Hearafter we choose to force $\mathbf{H}$ to have full rank, i.e.,

$$\text{rank}(\mathbf{H}) = m. \tag{10}$$

When $m = p$, (10) is a necessary condition to obtain (9). However when $m > p$, (10) is no longer necessary. Nevertheless we keep this condition (10) and we choose to minimize the criterion

$$J(\mathbf{H}) = \sum_{k=1}^{m} I(\mathbf{h}_k) + \rho\, g(\det(\mathbf{H}))\ ,\ \rho > 0. \tag{11}$$

where $I(.)$ is given in (7) and the even positive function $g(u)$ has a maximum in $u = 0$. This criterion will penalize the case where at least two weight vectors $\mathbf{h}_k$ and $\mathbf{h}_l$ are linearily dependent because then $\det(\mathbf{H}) = 0$ which is a maximum of the second part of the criterion. The stochastic gradient algorithm to minimize (11) provides the new learning rule

$$\Delta\mathbf{h}_k = -\mu \left\{ (y_k^3 - y_k)\mathbf{x} + \rho \det(\mathbf{H})\, g'(\det(\mathbf{H}))\, \mathbf{col}(k, \mathbf{H}^{-T}) \right\} \tag{12}$$

where $\mathbf{col}(k, \mathbf{M})$ denotes the kth column of a matrix $\mathbf{M}$ and μ a positive step-size. The choice of the function $g(.)$ is different in the two cases $m = p$ and $m > p$.

• $\underline{\mathbf{m} = \mathbf{p}}$, i.e., p is known a priori. Then any function $g(.)$ with a maximum in 0 is suitable. The choice

$$g(u) = -\mathrm{Ln}(|u|). \tag{13}$$

is advantageous to simplify the adaptation.

• $\underline{\mathbf{m} > \mathbf{p}}$, This case generally occurs when p is unknown. In this case we have shown that, unless $g(u)$ exhibits a finite non zero minimum, the algorithm (12) cannot be convergent.

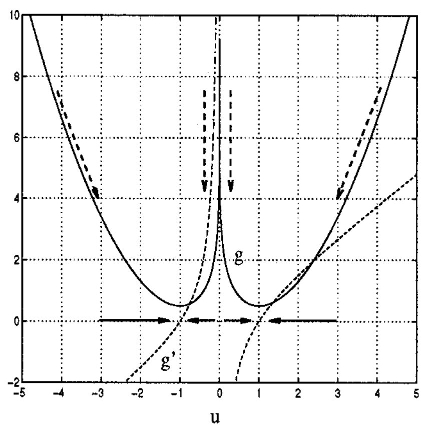

Figure 3: The function $g(u) = u^2/2 - \mathrm{Ln}(|u|)$ is plotted in continuous line and its derivative $g'(u) = u - 1/u$ is plotted in dashed line. The continuous (resp. dashed) arrows show the direction of evolution of $\det(\mathbf{H})$ (resp. $g(\det(\mathbf{H}))$) under the control of algorithm (15)

We can choose the function e.g.

$$g(u) = \frac{u^2}{2} - \mathrm{Ln}(|u|) \tag{14}$$

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 261-266

depicted in Fig.3. Then the learning rule (12) reads

$$\Delta \mathbf{h}_k = -\mu \left\{ (y_k^3 - y_k)\mathbf{x} + \rho \left(\det(\mathbf{H})^2 - 1\right) \mathbf{col}(k, \mathbf{H}^{-T}) \right\} \tag{15}$$

It is shown here that the algorithm (15) exhibits extracting equilibria. These equilibria have different expressions depending on whether $m = p$ or $m > p$. They are stable when the components a_i have negative kurtosises under some auxiliary conditions on μ, ρ and initialization.

4. Simulations

First consider a mixture of $p = 3$ independent components (a_1 is 4-PAM, a_2 is 8-PAM and a_3 is uniform) observed through $m = 3$ sensors. The mixture matrix is

$$\mathbf{G}_1 = \begin{bmatrix} 1 & 0.5 & 0.4 \\ 0.5 & 1 & 0.6 \\ 0.4 & 0.6 & 1 \end{bmatrix} \tag{16}$$

For $\mu = 0.01$, $\rho = 0.1$ and initial network weights $\mathbf{H}_0 = 0.2\,\mathbf{Id}_3$, after 3000 iterations, the overall matrix is

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{0.7807} & 0.0108 & -0.0185 \\ -0.0072 & \mathbf{0.7466} & -0.0137 \\ -0.0237 & -0.0400 & \mathbf{0.7406} \end{bmatrix} \tag{17}$$

The significant values in $\mathbf{C}$ correspond to the extraction of components. The neuron index corresponds to the column label and the index of the extracted component is the row label. According to (17), a_1 is extracted by the first neuron, a_2 is extracted by the second neuron and a_3 is extracted by the third neuron.

Now with the same 3 components a_1, a_2, a_3 and observations x_1, x_2, x_3, suppose that two additionnal observations x_4, x_5 are available, the mixture matrix being

$$\mathbf{G}_2 = \begin{bmatrix} & \mathbf{G}_1 & \\ 0.9 & 0.3 & 0.5 \\ 0.3 & 1 & 0.4 \end{bmatrix} \tag{18}$$

For the same values of μ, ρ and for $\mathbf{H}_0 = 0.2\,\mathbf{Id}_5$, and after 3000 iterations, the global matrix is

$$\mathbf{C}_2 = \begin{bmatrix} \mathbf{0.7645} & 0.0184 & 0.0172 & \mathbf{0.7649} & 0.0183 \\ -0.0058 & \mathbf{0.7397} & 0.0449 & -0.0058 & \mathbf{0.7398} \\ 0.0001 & -0.0249 & \mathbf{0.7337} & 0.0007 & -0.0246 \end{bmatrix} \tag{19}$$

According to (19), a_1 has been extracted by the first and the fourth neuron, a_2 has been extracted by the second and the fifth neuron, and a_3 has been extracted only by the third neuron. Note that each time a component is extracted it has the same gain.

ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 261-266

5. Conclusion

A neural network with a new updating rule for the adaptive unsupervised extraction of independent components is proposed in this paper. Adaptation of each neuron contains two parts: the first part minimizes a local extraction criterion, and the second global part minimizes a suitable function of all the neuron weights in order to ensure that all components are indeed extracted. When the number of observations is the same as the number of components, all the components are extracted by exactly one neuron. Moreover when the number of observations is greater than the number of components, all the components are also extracted, but some of them more than once. The equilibria of the adaptive algorithm are stable iff the components have negative kurtosises. It is possible to slightly modify the rule to encompass the case of positive kurtosises [7]. Simulations have been run for the two cases and have confirmed the satisfactory behaviour of the algorithm.

References

[1] Cardoso J.F. and Laheld B., "Equivariant adaptive source separation", Submitted to *IEEE Transactions on Signal Processing*, October 22, 1994.

[2] Comon P., "Independent component analysis, a new concept?", *Signal Processing*, Vol. 36, no. 3, pp 287-314, April 1994.

[3] Hérault J. and Jutten C., "Blind separation of sources, Part I : An adaptive algorithm based on neuromimetic architecture", *Signal Processing*, Vol. 24, pp. 1-10, 1991.

[4] Hyvärinen A., "Simple one-unit neural algorithms for blind source separation and blind deconvolution", *ICONIP*, Honk Kong, Sep 25-27 1996.

[5] Karhunen J. and Joutsensalo J., "Representation and separation of signals using nonlinear PCA type learning", *Neural Networks*, Vol. 7, No. 1, pp. 113-127, 1994.

[6] Macchi O. and Moreau E., "Self-adaptive source separation, Part I: Convergence analysis of a direct linear neural network controlled by the Hérault and Jutten adaptation law", Submitted to the *IEEE Transactions on Signal Processing*, May 1994.

[7] Malouche Z. and Macchi O., "Extended anti-Hebbian adaptation for unsupervised source extraction", *Proc. ICASSP*, Atlanta, 1996, Vol. 3 ,pp. 1665-1668.

[8] Oja E., "Principal components, minor components, and linear neural networks", *Neural Networks*, Vol. 5, pp. 927-935, 1992.