# An information-maximisation approach to blind separation and blind deconvolution

Anthony J. Bell and Terrence J. Sejnowski

Computational Neurobiology Laboratory
The Salk Institute
10010 N. Torrey Pines Road
La Jolla, California 92037*

**Abstract**

We derive a new self-organising learning algorithm which maximises the information transferred in a network of non-linear units. The algorithm does not assume any knowledge of the input distributions, and is defined here for the zero-noise limit. Under these conditions, information maximisation has extra properties not found in the linear case (Linsker 1989). The non-linearities in the transfer function are able to pick up higher-order moments of the input distributions and perform something akin to true redundancy reduction between units in the output representation. This enables the network to separate statistically independent components in the inputs: a higher-order generalisation of Principal Components Analysis.

We apply the network to the source separation (or cocktail party) problem, successfully separating unknown mixtures of up to ten speakers. We also show that a variant on the network architecture is able to perform blind deconvolution (cancellation of unknown echoes and reverberation in a speech signal). Finally, we derive dependencies of information transfer on time delays. We suggest that information maximisation provides a unifying framework for problems in 'blind' signal processing.

*Please send comments to tony@salk.edu. This paper will appear as *Neural Computation*, 7, 6, 1004-1034 (1995). The reference for this version is: Technical Report no. INC-9501, February 1995, Institute for Neural Computation, UCSD, San Diego, CA 92093-0523.

# 1 Introduction

This paper presents a convergence of two lines of research. The first, the development of information theoretic unsupervised learning rules for neural networks has been pioneered by Linsker 1992, Becker & Hinton 1992, Atick & Redlich 1993, Plumbley & Fallside 1988 and others. The second is the use, in signal processing, of higher-order statistics, for separating out mixtures of independent sources (blind separation) or reversing the effect of an unknown filter (blind deconvolution). Methods exist for solving these problems, but it is fair to say that many of them are *ad hoc*. The literature displays a diversity of approaches and justifications—for historical reviews see (Comon 1994) and (Haykin 1994a).

In this paper, we supply a common theoretical framework for these problems through the use of information-theoretic objective functions applied to neural networks with non-linear units. The resulting learning rules have enabled a principled approach to the signal processing problems, and opened a new application area for information theoretic unsupervised learning.

Blind separation techniques can be used in any domain where an array of $N$ receivers picks up linear mixtures of $N$ source signals. Examples include speech separation (the 'cocktail party problem'), processing of arrays of radar or sonar signals, and processing of multi-sensor biomedical recordings. A previous approach has been implemented in analog VLSI circuitry for real-time source separation (Vittoz et al 1989, Cohen et al 1992). The application areas of blind deconvolution techniques include the cancellation of acoustic reverberations (for example the 'barrel effect' observed when using speaker phones), the processing of geophysical data (seismic deconvolution) and the restoration of images.

The approach we take to these problems is a generalisation of Linsker's *infomax* principle to non-linear units with arbitrarily distributed inputs uncorrupted by any *known* noise sources. The principle is that described by Laughlin (1981) (see Fig.1a): when inputs are to be passed through a sigmoid function, maximum information transmission can be achieved when the sloping part of the sigmoid is optimally lined up with the high density parts of the inputs. As we show, this can be achieved in an adaptive manner, using a stochastic gradient ascent rule. The generalisation of this rule to multiple units leads to a system which, in maximising information transfer, also reduces the redundancy between the units in the output layer. It is this latter process, also called Independent Component Analysis (ICA), which enables the network to solve the blind separation task.

The paper is organised as follows. Section 2 describes the new information maximisation learning algorithm, applied, respectively to a single input, an $N \rightarrow N$ mapping, a causal filter, a system with time delays and a 'flexible' non-linearity. Section 3 describes the blind separation and blind deconvolution problems. Section 4 discusses the conditions under which the information maximisation process can find factorial codes (perform ICA), and therefore solve the separation and deconvolution problems. Section 5 presents results on the separation and deconvolution of speech signals. Section 6 attempts to place the theory and results in the context of previous work and mentions the limitations of the approach.

A brief report of this research appears in Bell & Sejnowski (1995).

## 2 Information maximisation

The basic problem tackled here is how to maximise the mutual information that the output $Y$ of a neural network processor contains about its input $X$. This is defined as:

$$I(Y, X) = H(Y) - H(Y|X) \tag{1}$$

where $H(Y)$ is the entropy of the output, while $H(Y|X)$ is whatever entropy the output has which didn't come from the input. In the case that we have no noise (or rather, we don't know what is noise and what is signal in the input), the mapping between $X$ and $Y$ is deterministic and $H(Y|X)$ has its lowest possible value: it diverges to $-\infty$. This divergence is one of the consequences of the generalisation of information theory to continuous variables. What we call $H(Y)$ is really the 'differential' entropy of $Y$ with respect to some reference, such as the noise-level or the accuracy of our discretisation of the variables in $X$ and $Y$[1]. To avoid such complexities, we consider here only the *gradient* of information theoretic quantities with respect to some parameter, $w$, in our network. Such gradients are as well-behaved as discrete-variable entropies, because the reference terms involved in the definition of differential entropies disappear. The above equation can be differentiated as follows, with respect

[1]see the discussion in Haykin 1994b, chapter 11, also Cover & Thomas, chapter 9.

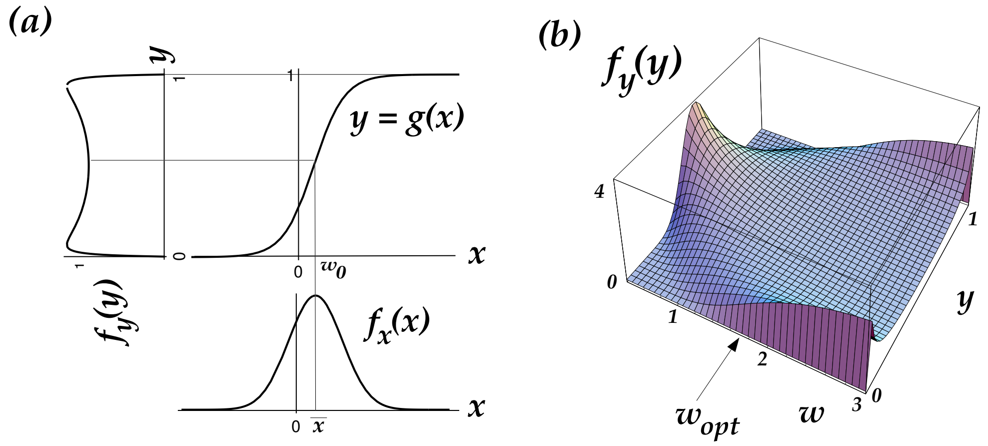


Figure 1: Optimal information flow in sigmoidal neurons (a) Input $x$ having density function $f_x(x)$, in this case a gaussian, is passed through a non-linear function $g(x)$. The information in the resulting density, $f_y(y)$ depends on matching the mean and variance of $x$ to the threshold, $w_0$, and slope, $w$, of $g(x)$ (see Schraudolph et al 1991). (b) $f_y(y)$ is plotted for different values of the weight $w$. The optimal weight, $w_{opt}$ transmits most information.

to a parameter, $w$, involved in the mapping from $X$ to $Y$:

$$\frac{\partial}{\partial w} I(Y, X) = \frac{\partial}{\partial w} H(Y) \tag{2}$$

because $H(Y|X)$ does not depend on $w$. This can be seen by considering a system which avoids infinities: $Y = G(X) + N$, where $G$ is some invertible transformation and $N$ is additive noise on the outputs. In this case, $H(Y|X) = H(N)$ (Nadal & Parga 1995). Whatever the level of this additive noise, maximisation of the mutual information, $I(Y, X)$, is equivalent to the maximisation of the output entropy, $H(Y)$, because $(\partial/\partial w)H(N) = 0$. There is nothing mysterious about the deterministic case, despite the fact that $H(Y|X)$ tends to minus infinity as the noise variance goes to zero.

Thus for invertible continuous deterministic mappings, the mutual information between inputs and outputs can be maximised by maximising the entropy of the outputs alone.

## 2.1 For one input and one output

When we pass a single input $x$ through a transforming function $g(x)$ to give an output variable $y$, both $I(y,x)$ and $H(y)$ are maximised when we align high density parts of the *probability density function* (pdf) of $x$ with highly sloping parts of the function $g(x)$. This is the idea of "matching a neuron's input-output function to the expected distribution of signals" that we find in (Laughlin 1981). See Fig.1a for an illustration.

When $g(x)$ is monotonically increasing or decreasing (ie: has a unique inverse), the pdf of the output, $f_y(y)$, can be written as a function of the pdf of the input, $f_x(x)$, (Papoulis, eq. 5-5):

$$f_y(y) = \frac{f_x(x)}{|\partial y / \partial x|} \tag{3}$$

where the bars denote absolute value. The entropy of the output, $H(y)$, is given by:

$$H(y) = -E\left[\ln f_y(y)\right] = -\int_{-\infty}^{\infty} f_y(y) \ln f_y(y) dy \tag{4}$$

where $E[.]$ denotes expected value. Substituting (3) into (4) gives

$$H(y) = E\left[\ln \left|\frac{\partial y}{\partial x}\right|\right] - E\left[\ln f_x(x)\right] \tag{5}$$

The second term on the right (the entropy of $x$) may be considered to be unaffected by alterations in a parameter $w$ determining $g(x)$. Therefore in order to maximise the entropy of $y$ by changing $w$, we need only concentrate on maximising the first term, which is the average log of how the input affects the output. This can be done by considering the 'training set' of $x$'s to approximate the density $f_x(x)$, and deriving an 'online', stochastic gradient ascent learning rule:

$$\Delta w \propto \frac{\partial H}{\partial w} = \frac{\partial}{\partial w}\left(\ln \left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1} \frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) \tag{6}$$

In the case of the logistic transfer function:

$$y = \frac{1}{1+e^{-u}}, \quad u = wx + w_0 \tag{7}$$

in which the input is multiplied by a weight $w$ and added to a bias-weight $w_0$, the terms above evaluate as:

$$\frac{\partial y}{\partial x} = wy(1-y) \tag{8}$$

$$\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) = y(1-y)(1+wx(1-2y)) \tag{9}$$

Dividing (9) by (8) gives the learning rule for the logistic function, as calculated from the general rule of (6):

$$\Delta w \propto \frac{1}{w} + x(1-2y) \tag{10}$$

Similar reasoning leads to the rule for the bias-weight:

$$\Delta w_0 \propto 1 - 2y \tag{11}$$

The effect of these two rules can be seen in Fig.1a. For example, if the input pdf $f_x(x)$ were gaussian, then the $\Delta w_0$-rule would centre the steepest part of the sigmoid curve on the peak of $f_x(x)$, matching input density to output slope, in a manner suggested intuitively by (3). The $\Delta w$-rule would then scale the slope of the sigmoid curve to match the variance of $f_x(x)$. For example, narrow pdf's would lead to sharply-sloping sigmoids. The $\Delta w$-rule is *anti-Hebbian*[2], with an *anti-decay* term. The anti-Hebbian term keeps $y$ away from one uninformative situation: that of $y$ being saturated at 0 or 1. But an anti-Hebbian rule alone makes weights go to zero, so the anti-decay term ($1/w$) keeps $y$ away from the other uninformative situation: when $w$ is so small that $y$ stays around 0.5.

The effect of these two balanced forces is to produce an output pdf, $f_y(y)$, that is close to the flat unit distribution, which is the maximum entropy distribution for a variable bounded between 0 and 1. Fig.1b shows a family of these distributions, with the most informative one occurring at $w_{opt}$.

A rule which maximises information for one input and one output may be suggestive for structures such as synapses and photoreceptors which must position the gain of their non-linearity at a level appropriate to the average value and size of the input fluctuations (Laughlin 1981). However, to see the advantages of this approach in artificial neural networks, we now analyse the case of multi-dimensional inputs and outputs.

[2]If $y = \tanh(wx + w_0)$ then $\Delta w \propto \frac{1}{w} - 2xy$

## 2.2 For an $N \to N$ network

Consider a network with an input vector $\mathbf{x}$, a weight matrix $\mathbf{W}$, a bias vector $\mathbf{w}_0$ and a monotonically transformed output vector $\mathbf{y} = g(\mathbf{W}\mathbf{x} + \mathbf{w}_0)$. Analogously to (3), the multivariate probability density function of $\mathbf{y}$ can be written (Papoulis, eq. 6-63):

$$f_{\mathbf{y}}(\mathbf{y}) = \frac{f_{\mathbf{x}}(\mathbf{x})}{|J|} \tag{12}$$

where $|J|$ is the absolute value of the Jacobian of the transformation. The Jacobian is the determinant of the matrix of partial derivatives:

$$J = \det \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix} \tag{13}$$

The derivation proceeds as in the previous section except instead of maximising $\ln |\partial y / \partial x|$, now we maximise $\ln |J|$. This latter quantity represents the log of the volume of space in $\mathbf{y}$ into which points in $\mathbf{x}$ are mapped. By maximising it, we attempt to spread our training set of $\mathbf{x}$-points evenly in $\mathbf{y}$.

For sigmoidal units, $\mathbf{y} = g(\mathbf{u})$, $\mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{w}_0$, with $g$ being the logistic function: $g(u) = (1 + e^{-u})^{-1}$, the resulting learning rules are familiar in form (proof given in the Appendix):

$$\Delta \mathbf{W} \propto \left[\mathbf{W}^T\right]^{-1} + (\mathbf{1} - 2\mathbf{y})\mathbf{x}^T \tag{14}$$

$$\Delta \mathbf{w}_0 \propto \mathbf{1} - 2\mathbf{y} \tag{15}$$

except that now $\mathbf{x}$, $\mathbf{y}$, $\mathbf{w}_0$ and $\mathbf{1}$ are vectors ($\mathbf{1}$ is a vector of ones), $\mathbf{W}$ is a matrix, and the anti-Hebbian term has become an outer product. The anti-decay term has generalised to an *anti-redundancy* term: the inverse of the transpose of the weight matrix. For an individual weight, $w_{ij}$, this rule amounts to:

$$\Delta w_{ij} \propto \frac{\mathrm{cof}\ w_{ij}}{\det \mathbf{W}} + x_j(1 - 2y_i) \tag{16}$$

where cof $w_{ij}$, the *cofactor* of $w_{ij}$, is $(-1)^{i+j}$ times the determinant of the matrix obtained by removing the $i$th row and the $j$th column from $\mathbf{W}$.

This rule is the same as the one for the single unit mapping, except that instead of $w = 0$ being an unstable point of the dynamics, now any degenerate weight matrix is unstable, since $\det \mathbf{W} = 0$ if $\mathbf{W}$ is degenerate. This fact enables different output units $y_i$ to learn to represent different things in the input. When the weight vectors entering two output units become too similar, $\det \mathbf{W}$ becomes small and the natural dynamic of learning causes these weight vectors to diverge from each other. This effect is mediated by the numerator, cof $w_{ij}$. When this cofactor becomes small, it indicates that there is a degeneracy in the weight matrix of the *rest* of the layer (ie: those weights not associated with input $x_j$ or output $y_i$). In this case, any degeneracy in $\mathbf{W}$has less to do with the specific weight $w_{ij}$ that we are adjusting. Further discussion of the convergence conditions of this rule (in terms of higher-order moments) is deferred to section 6.2.

The utility of this rule for performing blind separation is demonstrated in section 5.1.

### 2.3 For a causal filter

It is not necessary to restrict our architecture to weight *matrices*. Consider the top part of Fig.3b, in which a time series $x(t)$, of length $M$, is convolved with a causal filter $w_1, \ldots, w_L$ of impulse response $w(t)$, to give an output time series $u(t)$, which is then passed through a non-linear function $g$, to give $y(t)$. We can write this system either as a convolution or as a matrix equation:

$$\begin{aligned} y(t) &= g(u(t)) = g(w(t) * x(t)) && (17) \\ Y &= g(U) \quad = g(WX) && (18) \end{aligned}$$

in which $Y$, $X$ and $U$ are vectors of the whole time series, and $W$ is a $M \times M$ matrix. When the filtering is causal, $W$ will be lower triangular:

$$W = \begin{bmatrix} w_L & 0 & \cdots & 0 & 0 \\ w_{L-1} & w_L & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ w_1 & \cdots & w_L & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & w_1 & \cdots & w_L \end{bmatrix} \tag{19}$$

At this point, we take the liberty of imagining there is an ensemble of such time series, so that we can write,

$$f_Y(Y) = \frac{f_X(X)}{|J|} \tag{20}$$

where again, $|J|$ is the Jacobian of the transformation. We can 'create' this ensemble from a single time series by chopping it into bits (of length $L$ for example, making $W$ in (19) an $L \times L$ matrix). The Jacobian in (20) is written as follows:

$$J = \det \left[ \frac{\partial y(t_i)}{\partial x(t_j)} \right]_{ij} = (\det W) \prod_{t=1}^{M} y'(t) \tag{21}$$

and may be decomposed into the determinant of the weight matrix (19), and the product of the slopes of the squashing function, $y'(t) = \partial y(t)/\partial u(t)$, for all times $t$ [see Appendix (53)]. Because $W$ is lower-triangular, its determinant is simply the product of its diagonal values, which is $w_L^M$. As in the previous section, we maximise the joint entropy $H(Y)$ by maximising $\ln |J|$, which can then be simply written as:

$$\ln |J| = \ln |w_L^M| + \sum_{t=1}^{M} \ln |y'(t)| \tag{22}$$

If we assume that our non-linear function $g$ is the hyperbolic tangent (tanh), then differentiation with respect to the weights in our filter $w(t)$, gives two simple[3] rules:

$$\Delta w_L \propto \sum_{t=1}^{M} \left( \frac{1}{w_L} - 2x_t \; y_t \right) \tag{23}$$

$$\Delta w_{L-j} \propto \sum_{t=j}^{M} (-2x_{t-j} \; y_t) \tag{24}$$

Here, $w_L$ is the 'leading' weight, and the $w_{L-j}$, where $j > 0$, are tapped delay lines linking $x_{t-j}$ to $y_t$. The leading weight thus adjusts just as would a weight connected to a neuron with only that one input (see section 2.1). The delay weights attempt to decorrelate the past input from the present output. Thus the filter is kept from 'shrinking' by its leading weight.

The utility of this rule for performing blind deconvolution is demonstrated in section 5.2.

[3]The corresponding rules for *non*-causal filters are substantially more complex.

### 2.4 For weights with time delays

Consider a weight, $w$, with a time delay, $d$, and a sigmoidal non-linearity, $g$, so that:

$$y(t) = g[wx(t-d)] \tag{25}$$

We can maximise the entropy of $y$ with respect to the time delay, again by maximising the log slope of $y$ [as in (6)] :

$$\Delta d \propto \frac{\partial H}{\partial d} = \frac{\partial}{\partial d}(\ln |y'|) \tag{26}$$

The crucial step in this derivation is to realise that

$$\frac{\partial}{\partial d}x(t-d) = -\frac{\partial}{\partial t}x(t-d). \tag{27}$$

Calling this quantity simply $-\dot{x}$, we may then write:

$$\frac{\partial y}{\partial d} = -w\dot{x}y' \tag{28}$$

Our general rule is therefore given as follows:

$$\frac{\partial}{\partial d}(\ln |y'|) = \frac{1}{y'}\frac{\partial y'}{\partial y}\frac{\partial y}{\partial d} = -w\dot{x}\frac{\partial y'}{\partial y} \tag{29}$$

When $g$ is the tanh function, for example, this yields the following rule for adapting the time delay:

$$\Delta d \propto 2w\dot{x}y. \tag{30}$$

This rule holds regardless of the architecture in which the network is embedded, and it is local, unlike the $\Delta w$ rule in (16). It bears a resemblance to the rule proposed by Platt & Faggin (1992) for adjustable time delays in the network architecture of Jutten & Herault (1991).

The rule has an intuitive interpretation. Firstly, if $w = 0$, there is no reason to adjust the delay. Secondly, the rule maximises the delivered *power* of the inputs, stabilising when $\langle \dot{x}y \rangle = 0$. As an example, if $y$ received several sinusoidal inputs of the same frequency, $\omega$, and different phase, each with its own adjustable time delay, then the time delays would adjust until the phases of the time-delayed inputs were all the same. Then, for each input, $\langle \dot{x}y \rangle$ would be proportional to $\langle \cos \omega t \cdot \tanh(\sin \omega t) \rangle$ which would be zero.

In adjusting delays, therefore, the rule will attempt to line up similar signals in time, and cancel time delays caused by the same signal taking alternate paths.

We hope to explore, in future work, the usefulness of this rule for adjusting time delays and tap-spacing in blind separation and blind deconvolution tasks.

### 2.5 For a generalised sigmoid function

In section 4, we show how it is sometimes necessary not only to train the weights of the network, but also to select the form of the non-linearity, so that it can 'match' input pdf's. In other words, if the input to a neuron is $u$, with a pdf of $f_u(u)$, then our sigmoid should approximate, as closely as possible, the cumulative distribution of this input:

$$y = g(u) \simeq \int_{-\infty}^{u} f_u(v)dv \tag{31}$$

One way to do this is to define a 'flexible' sigmoid which can be altered to *fit* the data, in the sense of (31). An example of such a function is the asymmetric generalised logistic function (see also Baram & Roth 1994) described by the differential equation:

$$y' = \frac{dy}{du} = y^p(1-y)^r \tag{32}$$

where $p$ and $r$ are positive real numbers. Numerical integration of this equation produces sigmoids suitable for very peaked (as $p, r > 1$, see Fig.2b) and flat, unit-like (as $p, r < 1$, see Fig.2c) input distributions. So by varying these coefficients, we can mold the sigmoid so that its slope fits unimodal distributions of varying kurtosis. By having $p \neq r$, we can also account for some skew in the distributions. When we have chosen values for $p$ and $r$, perhaps by some optimisation process, the rules for changing a single input-output weight, $w$, and a bias, $w_0$, are subtly altered from (10) and (11), but clearly the same when $p = r = 1$:

$$\Delta w \propto \frac{1}{w} + x(p(1-y) - ry) \tag{33}$$

$$\Delta w_0 \propto p(1-y) - ry \tag{34}$$

The importance of being able to train a general function of this type will be explained in section 4.

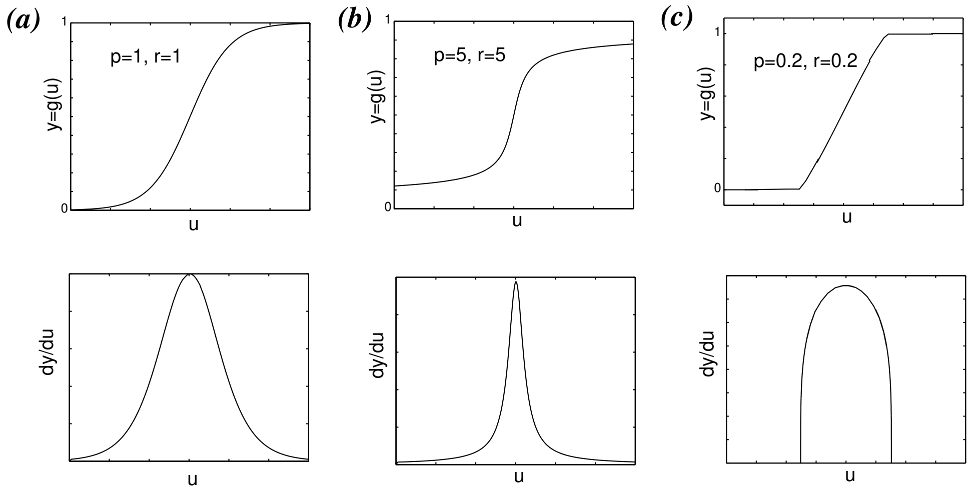


Figure 2: The generalised logistic sigmoid (top row) of (32), and its slope, $y'$, (bottom row), for (a) $p = r = 1$, (b) $p = r = 5$ and (c) $p = r = 0.2$. Compare the slope of (b) with the pdf in Fig.5a: it provides a good match for natural speech signals.

# 3 Background to blind separation and blind deconvolution

Blind separation and blind deconvolution are related problems in signal processing. In *blind separation*, as introduced by Herault & Jutten (1986), and illustrated in Fig.3a, a set of sources, $s_1(t), \ldots, s_N(t)$, (different people speaking, music etc) are mixed together linearly by a matrix $\mathbf{A}$. We do not know anything about the sources, or the mixing process. All we receive are the $N$ superpositions of them, $x_1(t), \ldots, x_N(t)$. The task is to recover the original sources by finding a square matrix, $\mathbf{W}$, which is a permutation and rescaling of the inverse of the unknown matrix, $\mathbf{A}$. The problem has also been called the 'cocktail-party' problem[4]

In *blind deconvolution*, described in (Haykin 1991, 1994a) and illustrated in Fig.3b, a single unknown signal $s(t)$ is convolved with an unknown tapped delay-line filter $a_1, \ldots, a_K$, giving a corrupted signal $x(t) = a(t) * s(t)$ where $a(t)$ is the impulse response of the filter. The task is to recover $s(t)$ by convolving

[4]though for now, we ignore the problem of signal propagation delays.

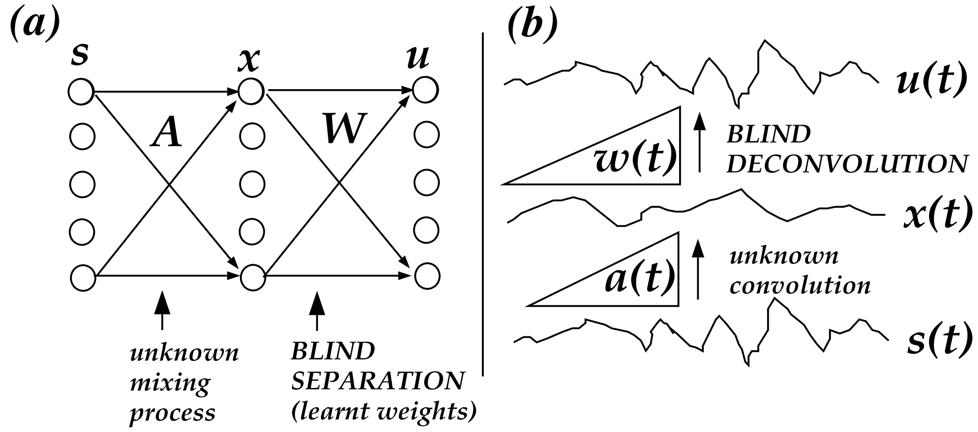


Figure 3: Network architectures for (a) blind separation of 5 mixed signals, and (b) blind deconvolution of a single signal.

$x(t)$ with a learnt filter $w_1, \ldots, w_L$ which reverses the effect of the filter $a(t)$.

There are many similarities between the two problems. In one, sources are corrupted by the superposition of other sources. In the other, a source is corrupted by time-delayed versions of itself. In both cases, unsupervised learning must be used because no error signals are available. In both cases, second-order statistics are inadequate to solve the problem.

For example, for blind separation, a second-order decorrelation technique such as that of Barlow & Földiák (1989) would find uncorrelated, or linearly independent, projections, $\mathbf{y}$, of the input data, $\mathbf{x}$. But it could only find a symmetric decorrelation matrix, which would not suffice if the mixing matrix, $\mathbf{A}$, were asymmetric (Jutten & Herault 1991). Similarly, for blind deconvolution, second-order techniques based on the autocorrelation function, such as prediction-error filters, are *phase-blind*. They do not have sufficient information to estimate the phase of the corrupting filter, $a(t)$, only its amplitude (Haykin 1994a).

The reason why second-order techniques fail is that these two 'blind' signal processing problems are information theoretic problems. We are assuming, in the case of blind separation, that the sources, $\mathbf{s}$, are statistically independent and non-gaussian, and in the case of blind deconvolution, that the original signal, $s(t)$, consists of independent symbols (a white process). Then blind separation becomes the problem of minimising the mutual information between outputs, $u_i$, introduced by the mixing matrix $\mathbf{A}$; and blind deconvolution becomes the problem of removing from the convolved signal, $x(t)$, any statistical dependencies across time, introduced by the corrupting filter $a(t)$. The former process, the learning of $\mathbf{W}$, is called the problem of Independent Component Analysis, or *ICA* (Comon 1994). The latter process, the learning of $w(t)$ is sometimes called the *whitening* of $x(t)$. Henceforth, we use the term *redundancy reduction* when we mean either ICA or the whitening of a time series.

In either case, it is clear in an information-theoretic context, that second-order statistics are inadequate for reducing redundancy, because the mutual information between two variables involves statistics of all orders, except in the special case that the variables are jointly gaussian.

In the various approaches in the literature, the higher-order statistics required for redundancy reduction have been accessed in two main ways. The first way is the explicit estimation of cumulants and polyspectra. See Comon (1994) and Hatzinakos & Nikias (1994) for the application of this approach to separation and deconvolution respectively. The drawbacks of such direct techniques are that they can sometimes be computationally intensive, and may be inaccurate when cumulants higher than 4th order are ignored, as they usually are. It is currently not clear why direct approaches can be surprisingly successful despite errors in the estimation of the cumulants, and in the usage of these cumulants to estimate mutual information.

The second main way of accessing higher-order statistics is through the use of static non-linear functions. The Taylor series expansions of these non-linearities yield higher-order terms. The hope, in general, is that learning rules containing such terms will be sensitive to the right higher-order statistics necessary to perform ICA or whitening. Such reasoning has been used to justify both the Herault-Jutten (or 'H-J') approach to blind separation (Comon et al 1991) and the so-called 'Bussgang' approaches to blind deconvolution (Bellini 1994). The drawback here is that there is no guarantee that the higher-order statistics yielded by the non-linearities are weighted in a way relating to the calculation of statistical dependency. For the H-J algorithm, the standard approach is to try different non-linearities on different problems to see if they work.

Clearly, it would be of benefit to have some method of rigorously linking our choice of a static non-linearity to a learning rule performing gradient ascent in some quantity relating to statistical dependency. Because of the infinite number of higher-order statistics involved in statistical dependency, this has generally been thought to be impossible. As we now show, this belief is incorrect.

# 4 When does information maximisation reduce statistical dependence?

In this section, we consider under what conditions the information *maximisation* algorithm presented in section 2 *minimises* the mutual information between outputs (or time points) and therefore performs redundancy reduction.

Consider a system with two outputs, $y_1$ and $y_2$ (two output channels in the case of separation, or two time points in the case of deconvolution). The joint entropy of these two variables may be written as (Papoulis, eq. 15-93):

$$H(y_1, y_2) = H(y_1) + H(y_2) - I(y_1, y_2). \tag{35}$$

Maximising this joint entropy consists of maximising the individual entropies while minimising the mutual information, $I(y_1, y_2)$, shared between the two. When this latter quantity is zero, the two variables are statistically independent, and the pdf can be factored: $f_{y_1 y_2}(y_1, y_2) = f_{y_1}(y_1) f_{y_2}(y_2)$. Both ICA and the 'whitening' approach to deconvolution are examples of minimising $I(y_1, y_2)$ for all pairs $y_1$ and $y_2$. This process is variously known as factorial code learning (Barlow 1989), predictability minimisation (Schmidhuber 1992) as well as independent component analysis (Comon 1994) and redundancy reduction (Barlow 1961, Atick 1992).

The algorithm presented in section 2 is a stochastic gradient ascent algorithm which maximises the joint entropy in (35). In doing so, it will, in general, reduce $I(y_1, y_2)$, reducing the statistical dependence of the two outputs.

However, it is not guaranteed to reach the absolute minimum of $I(y_1, y_2)$, because of interference from the other terms, the $H(y_i)$. Fig.4 shows one pathological situation where a 'diagonal' projection, 4c, of two independent, uniformly-distributed variables $x_1$ and $x_2$ is preferred over an 'independent' projection, 4b. This is because of a 'mismatch' between the input pdf's and the slope of the sigmoid non-linearity. The learning procedure is able to achieve higher values in 4c for the individual output entropies, $H(y_1)$ and $H(y_2)$, because the pdf's of $x_1 + x_2$ and $x_1 - x_2$ are triangular, more closely matching the slope of the sigmoid. This interferes with the minimisation of $I(y_1, y_2)$.

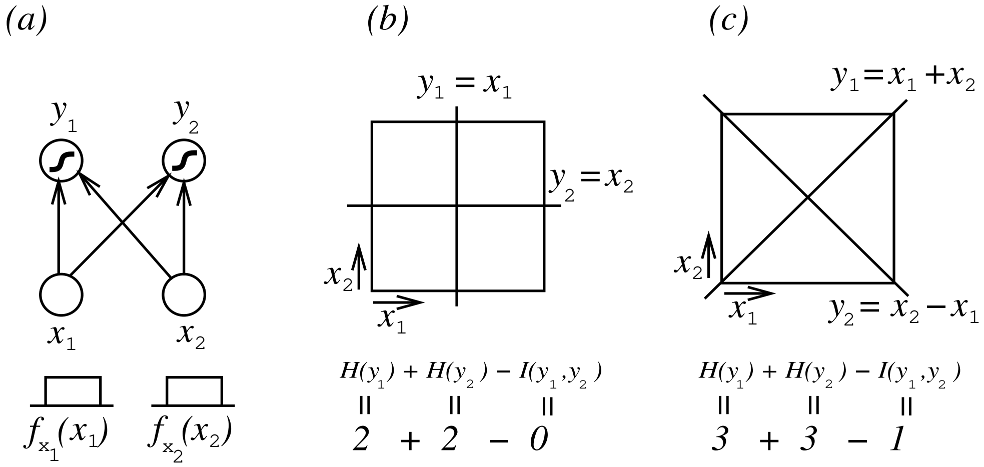


Figure 4: An example of when joint entropy maximisation fails to yield statistically independent components. (a) Two independent input variables, $x_1$ and $x_2$, having uniform (flat) pdf's are input into an entropy maximisation network with sigmoidal outputs. Because the input pdf's are not well-matched to the non-linearity, the 'diagonal' solution (c) has higher joint entropy than the 'independent-component' solution (b), despite its having non-zero mutual information between the outputs. The values given are for illustration purposes only.

In many practical situations, however, such interference will have minimal effect. We conjecture that only when the pdf's of the inputs are *sub-gaussian* (meaning their kurtosis, or 4th order standardised cumulant, is less than 0), may unwanted higher entropy solutions for logistic sigmoid networks be found by combining inputs in the way shown in 4c (Kenji Doya, personal communication). Many real-world analog signals, including the speech signals we used, are super-gaussian. They have longer tails and are more sharply peaked than gaussians (see Fig.5). For such signals, in our experience, maximising the joint entropy in simple logistic sigmoidal networks always minimises the mutual information between the outputs (see the results in section 5).

We can tailor conditions so that the mutual information between outputs *is* minimised, by constructing our non-linear function, $g(u)$, so that it matches, in the sense of (31), the known pdf's of the independent variables. When this is the case, $H(y)$ will be maximised (meaning $f_y(y)$ will be the flat unit

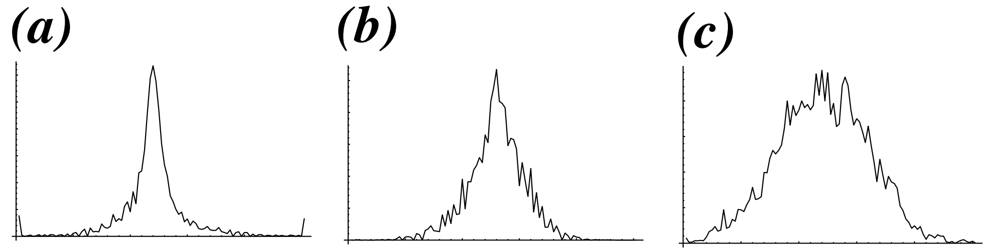


Figure 5: Typical probability density functions for (a) speech (b) rock music and (c) gaussian white noise. The kurtosis of pdf's (a) and (b) was greater than 0, and they would be classified as super-gaussian.

distribution) only when $u$ carries one single independent variable. Any linear combination of the variables will produce a 'more gaussian' $f_u(u)$ (due to central limit tendencies) and a resulting suboptimal (non-flat) $f_y(y)$.

We have presented, in section 2.5, one possible 'flexible' non-linearity. This suggests a two-stage algorithm for performing Independent Component Analysis. First, a non-linearity such as that defined by (32) is optimised to approximate the cumulative distributions, (31), of known independent components (sources). Then networks using this non-linearity are trained using the full weight matrix and bias vector generalisation of (33) and (34):

$$\Delta \mathbf{W} \propto \left[\mathbf{W}^T\right]^{-1} + [p(\mathbf{1}-\mathbf{y}) - r\mathbf{y}]\mathbf{x}^T \tag{36}$$

$$\Delta \mathbf{w}_0 \propto p(\mathbf{1}-\mathbf{y}) - r\mathbf{y} \tag{37}$$

This way, we can be sure that the problem of maximising the mutual information between the inputs and outputs, and the problem of minimising the mutual information between the outputs, have the same solution.

This argument is well-supported by the analysis of Nadal & Parga (1995), who independently reached the conclusion that in the low-noise limit, information maximisation yields factorial codes when both the non-linear function, $g(u)$, and the weights, $w$, can be optimised. Here, we provide a practical optimisation method for the weights and a framework for optimising the non-linear function. Having discussed these caveats, we now present results for blind separation and blind deconvolution using the standard logistic function.

# 5 Methods and results

The experiments presented here were obtained using 7 second segments of speech recorded from various speakers (only one speaker per recording). All signals were sampled at 8 kHz from the output of the auxiliary microphone of a Sparc-10 workstation. No special post-processing was performed on the waveforms, other than that of normalising their amplitudes so they were appropriate for use with our networks (input values roughly between -3 and 3). The method of training was stochastic gradient ascent, but because of the costly matrix inversion in (14), weights were usually adjusted based on the summed $\Delta\mathbf{W}$'s of small 'batches' of length $B$, where $5 \leq B \leq 300$. Batch training was made efficient using vectorised code written in MATLAB. To ensure that the input ensemble was stationary in time, the time index of the signals was permuted. This means that at each iteration of the training, the network would receive input from a random time point. Various learning rates[5] were used (0.01 was typical). It was helpful to reduce the learning rate during learning for convergence to good solutions.

## 5.1 Blind separation results

The architecture in Fig.3a and the algorithm in (14) and (15) was sufficient to perform blind separation. A random mixing matrix , $\mathbf{A}$, was generated with values usually uniformly distributed between -1 and 1. This was used to make the mixed time series, $\mathbf{x}$ from the original sources, $\mathbf{s}$. The matrices $\mathbf{s}$ and $\mathbf{x}$, then, were both $N \times M$ matrices ($N$ signals, $M$ timepoints), and $\mathbf{x}$ was constructed from $\mathbf{s}$ by (1) permuting the time index of $\mathbf{s}$ to produce $\mathbf{s}^\dagger$, and (2) creating the mixtures, $\mathbf{x}$ by multiplying by the mixing matrix: $\mathbf{x} = \mathbf{A}\mathbf{s}^\dagger$. The unmixing matrix, $\mathbf{W}$, and the bias vector $\mathbf{w}_0$ were then trained.

An example run with five sources is shown in Fig.6. The mixtures, $\mathbf{x}$, formed an incomprehensible babble. This unmixed solution was reached after around $10^6$ time points were presented, equivalent to about 20 passes through the complete time series.[6], though much of the improvement occurred on the first few passes through the data. Any residual interference in $\mathbf{u}$ is inaudible.

[5]The learning rate is defined as the proportionality constant in (14)-(15) and (23)-(24).

[6]This took on the order of 5 minutes on a Sparc-10. Two hundred data points were presented at a time in a 'batch', then the weights were changed with a learning rate of 0.01 based on the sum of the 200 accumulated $\Delta w$'s.

This figure contains speech waveforms arranged in 3 columns: the original waveforms, the mixed signals, and the unmixed signals. The postscript of this figure is 4MB, hence its omission from this electronic version.

Figure 6: A $5 \times 5$ information maximisation network performed blind separation, learning the unmixing matrix $\mathbf{W}$. The outputs, $\mathbf{u}$, are shown here unsquashed by the sigmoid. They can be visually matched to their corresponding sources, $\mathbf{s}$, even though their order was different and some (for example $u_1$) were recovered as negative (upside down).

This is reflected in the permutation structure of the matrix $\mathbf{WA}$:

$$\mathbf{WA} = \begin{bmatrix} \boxed{-4.09} & 0.13 & 0.09 & -0.07 & -0.01 \\ 0.07 & \boxed{-2.92} & 0.00 & 0.02 & -0.06 \\ 0.02 & -0.02 & -0.06 & -0.08 & \boxed{-2.20} \\ 0.02 & 0.03 & 0.00 & \boxed{1.97} & 0.02 \\ -0.07 & 0.14 & \boxed{-3.50} & -0.01 & 0.04 \end{bmatrix} \tag{38}$$

As can be seen, only one substantial entry (boxed) exists in each row and column. The interference was attenuated by between 20 and 70dB in all cases, and the system was continuing to improve slowly with a learning rate of 0.0001.

In our most ambitious attempt, ten sources (six speakers, rock music, raucous laughter, a gong and the Hallelujah chorus) were successfully separated, though the fine tuning of the solution took many hours and required some annealing of the learning rate (lowering it with time). For two sources, convergence is normally achieved in less than one pass through the data (50,000 data points), and on a Sparc-10 on-line learning can occur at twice the speed at which the sounds themselves are played. Real-time separation for more than, say, 3 sources, may require further work to speed convergence, or special-purpose hardware.

In all our attempts at blind separation, the algorithm has only failed under two conditions:

1. when more than one of the sources were gaussian white noise, and

2. when the mixing matrix, $\mathbf{A}$ was almost singular.

Both are understandable. Firstly, no procedure can separate out independent gaussian sources since the sum of two gaussian variables has itself a gaussian distribution. Secondly, if $\mathbf{A}$ is almost singular, then any unmixing $\mathbf{W}$ must also be almost singular, making the learning in (14) quite unstable in the vicinity of a solution.

In contrast with these results, our experience with tests on the H-J network of Jutten & Herault (1991) has been that it occasionally fails to converge for two sources and only rarely converges for three, on the same speech and music signals we used for separating ten sources. Cohen & Andreou (1992) report separation of up to six sinusoidal signals of different frequencies using analog VLSI H-J networks. In addition, in Cohen & Andreou (1995), they report results with mixed sine waves and noise in 5x5 networks, but no separation results for more than two speakers.

How does convergence time scale with the number of sources, $N$? The difficulty in answering this question is that different learning rates are required for different $N$ and for different stages of convergence. We expect to address this issue in future work, and employ useful heuristic or explicit 2nd order techniques (Battiti 1992) to speed convergence. For now, we present rough estimates for the number of epochs (each containing 50,000 data vectors) required to reach an average signal to noise ratio on the ouput channels of 20dB. At such a level, approximately 80% of the each output channel amplitude is devoted to one signal. These results were collected for mixing matrices of unit determinant, so that convergence would not be hampered by having to find an unmixing matrix with especially large entries. Therefore these convergence times may be lower than for randomly-generated matrices. The batch size, $B$, was in each case, 20.

The average numbers of epochs to convergence (over 10 trials) and the computer times consumed per epoch (on a Sparc-10) are given in the following table:

| no. of sources, N | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| learning rate | 0.1 | 0.1 | 0.1 | 0.05 | 0.05 | 0.025 | 0.025 | 0.025 | 0.0125 |
| epochs to convergence | $<1$ | $<1$ | 2.25 | 5.0 | 9.0 | 9.2 | 13.8 | 14.9 | 30.6 |
| time in secs./epoch | 12.1 | 13.3 | 14.6 | 15.6 | 16.9 | 18.4 | 19.9 | 21.7 | 23.6 |

## 5.2 Blind deconvolution results

Speech signals were convolved with various filters and the learning rules in (23) and (24) were used to perform blind deconvolution. Some results are shown in Fig.7. The convolving filters generally contained some zero values. For example, 7e is the filter [0.8,0,0,0,1]. In addition, the taps were sometimes adjacent to each other 7a-d and sometimes spaced out in time 7i-l. The 'leading weight' of each filter is the rightmost bar in each histogram.

For each of the three experiments shown in Fig.7, we display the convolving filter, $a(t)$, the truncated inverting filter, $w_{ideal}(t)$, the filter produced by our algorithm, $w(t)$, and the convolution of $w(t)$ and $a(t)$. The latter should be a delta-function (ie: consist of only a single high value, at the position of the leading weight) if $w(t)$ correctly inverts $a(t)$.

The first example, Fig.7a-d, shows what happens when one tries to 'deconvolve' a speech signal that has not actually been corrupted (filter $a(t)$ is a delta function). If the tap spacing is close enough, (in this case, as close as the samples), the algorithm learns a whitening filter 7c which flattens the amplitude spectrum of the speech right up to the Nyquist limit, the frequency corresponding to half the sampling rate. The spectra before and after such 'deconvolution' are shown in Fig.8. Whitened speech sounds like a clear sharp version of the original signal since the phase structure is preserved. Using all available frequency levels equally is another example of maximising the information throughput of a channel.

This shows that when the original signal is not white, we may recover a whitened version of it, rather than the exact original. However, when the taps

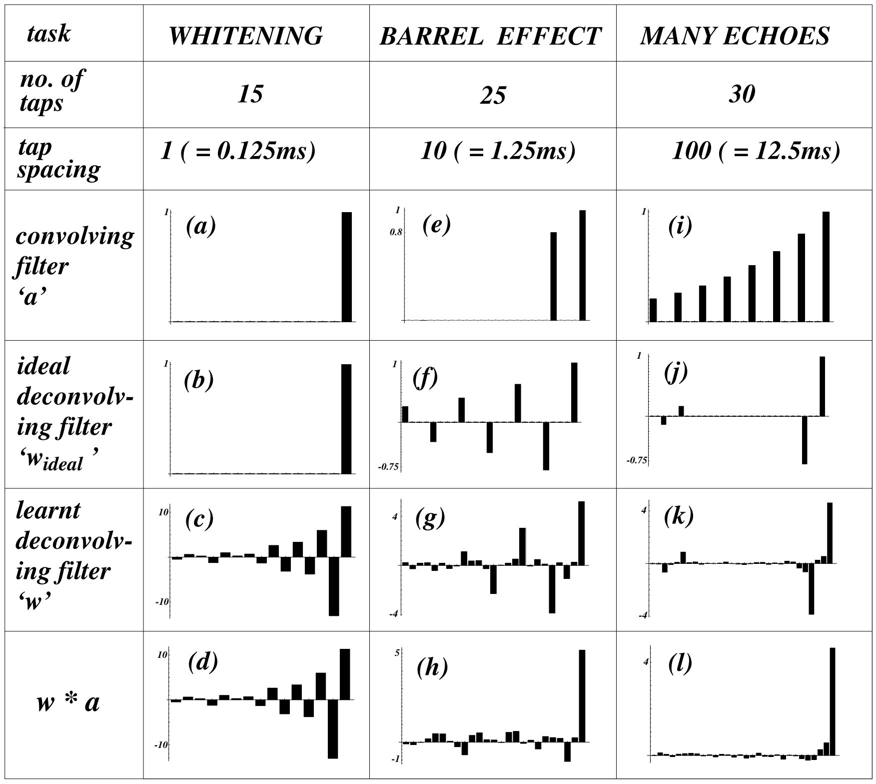


Figure 7: Blind deconvolution results. (a,e,i) Filters used to convolve speech signals, (b,f,j) their inverses, (c,g,k) deconvolving filters learnt by the algorithm, and (d,h,l) convolution of the convolving and deconvolving filters. See text for further explanation.

are spaced out further, as in 7e-l, there is less opportunity for simple whitening.

In the second example, 7e, a 6.25 ms echo is added to the signal. This creates a mild audible "barrel effect"[7] Because filter 7e is finite in length, its inverse, 7f, is infinite in length, shown here truncated. The inverting filter learnt in 7g resembles it, though the resemblance tails off towards the left since we are really learning an optimal filter of finite length, not a truncated infinite filter. The resulting deconvolution, 7h, is quite good.

The cleanest results, though, come when the ideal deconvolving filter is of

[7]An example of the barrel effect are the acoustic echoes heard when someone talks into a 'speaker-phone'.

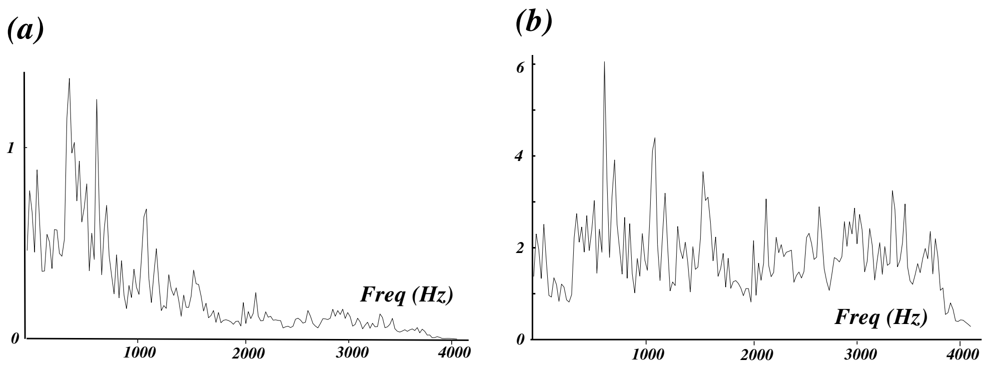


Figure 8: Amplitude spectra of a speech signal (a) before and (b) after the 'whitening' performed in Fig.7c.

finite length, as in our third example. A set of exponentially decaying echoes spread out over 275 ms, 7i, may be inverted by a two-point filter, 7j, with a small decaying correction on its left, an artifact of the truncation of the convolving filter 7i. As seen in 7k, the learnt filter corresponds almost exactly to the ideal one, and the deconvolution in 7l is almost perfect. This result shows the sensitivity of the learning algorithm in cases where the tap-spacing is great enough (12.5 ms) that simple whitening does not interfere noticeably with the deconvolution process. The deconvolution result, in this case, represents an improvement of the signal-to-noise ratio from -23dB to 12dB. In all cases, convergence was relatively rapid, with these solutions being produced after on the order of 70,000 data points were presented, which amounts to 2 seconds training on 8 seconds of speech, amounting to four times as fast as real-time on a Sparc-10.

## 5.3 Combining separation and deconvolution

The blind separation rules in (14) and (15) and the blind deconvolution rules in (23) and (24) can be easily combined. The objective then becomes the maximisation of the log of a Jacobian with *local* lower triangular structure. This yields exactly the learning rule one would expect: the leading weights in the filters follow the blind separation rules and all the others follow a decorrelation rule similar to (24) except that now there are tapped weights $w_{ikj}$ between an input $x_j(t-k)$ and an output $y_i(t)$.

We have performed experiments with speech signals in which signals have been simultaneously separated and deconvolved using these rules. We used mixtures of two signals with convolution filters like those in 7e and 7i, and convergence to separated, deconvolved speech was almost perfect.

# 6 Discussion

We will consider these techniques firstly in the context of previous information theoretic approaches within neural networks, and then in the context of related approaches to 'blind' signal processing problems.

## 6.1 Comparison with previous work on information maximisation

Many authors have formulated optimality criteria similar to ours, for both neural networks and sensory systems (Barlow 1989, Atick 1992, Bialek, Ruderman & Zee 1991). However, our work is most similar to that of Linsker, who in 1989 proposed an 'infomax' principle for linear mappings with various forms of noise. Linsker (1992) derived a learning algorithm for maximising the mutual information between two layers of a network. This 'infomax' criterion is the same as ours [see eq.(1)]. However, the problem as formulated here is different in the following respects:

1. There is no noise, or rather, there is no noise *model* in this system.
2. There is no assumption that inputs or outputs have gaussian statistics.
3. The transfer function is in general non-linear.

These differences lead to quite a different learning rule. Linsker's 1992 rule uses (for input signal $X$ and output $Y$) a Hebbian term to maximise $H(Y)$ when the network receives both signal and noise, an anti-Hebbian term to minimise $H(Y|X)$ when the system receives only noise, and an anti-Hebbian lateral interaction to decorrelate the outputs $Y$. When the network is deterministic, however, the $H(Y|X)$ term does not contribute. A deterministic linear network can increase its information throughput without bound, as the $[\mathbf{W}^T]^{-1}$ term in (14) suggests.

However, the information capacity in the networks we have considered *is* bounded, not by noise, but by the saturation of a squashing function. Our network shares with Linsker's the property that this bound gives rise to an anti-Hebbian term in the learning rule. This is true for various squashing functions (see the table in the Appendix).

This non-linear, non-gaussian, deterministic formulation of the 'infomax' problem leads to more powerful algorithms, since, as demonstrated, the non-linear function enables the network to compute with non-gaussian statistics, and find higher-order forms of redundancy inherent in the inputs. (As emphasised in section 3, linear approaches are inadequate for solving the problems of blind separation and blind deconvolution.) These observations also apply to the approaches of Atick & Redlich (1993) and Bialek, Ruderman & Zee (1991).

The problem of information maximisation through non-linear sigmoidal neurons has been considered before without a learning rule actually being proposed. Schraudolph et al (1991), in work that inspired this approach, considered it as a method for initialising weights in a neural network. Before this, Laughlin (1981) used it to characterise as optimal, the exact contrast sensitivity curves of interneurons in the insect's compound eye. Various other authors have considered unsupervised learning rules for non-linear units, without justifying them in terms of information theory (see Karhunen & Joutsensalo 1994, and references therein).

Several recent papers, however, have touched closely on the work presented in this paper. Deco & Brauer (1995) use cumulant expansions to approximate mutual information between outputs. Parra & Deco (1995) use symplectic transforms to train *non-linear* information-preserving mappings. Most notably, Baram & Roth (1994) perform substantially the same analysis as ours, but apply their networks to probability density estimation and time series forecasting. None of this work was known to us when we developed our approach.

Finally, another group of information theoretic algorithms have been proposed by Becker & Hinton (1992). These employ non-linear networks to *maximise* mutual information between different sets of outputs. This *increasing* of redundancy enables the network to discover invariants in separate groups of inputs (see also Schraudolph & Sejnowski 1992). This is, in a sense, the opposite of our objective, though some way may be found to view the two in the same light.

## 6.2 Comparison with previous work on blind separation

As indicated in section 3, approaches to blind separation and blind deconvolution have divided into those using non-linear functions (Jutten & Herault 1991, Bellini 1994) and those using explicit calculations of cumulants and polyspectra (Comon 1994, Hatzinakos & Nikias 1994). We have shown that an information maximisation approach can provide a theoretical framework for approaches of the former type.

In the case of blind separation, the architecture of our $N \to N$ network, although it is a feedforward network, maps directly onto that of the recurrent Herault-Jutten network. The relationship between our weight matrix, $\mathbf{W}$, and the H-J recurrent weight matrix, $\mathbf{W}_{HJ}$, can be written as: $\mathbf{W} = (\mathbf{I} + \mathbf{W}_{HJ})^{-1}$, where $\mathbf{I}$ is the identity matrix. From this we may write

$$\Delta \mathbf{W}_{HJ} = \Delta \left( \mathbf{W}^{-1} \right) = \left( \mathbf{W}^{-1} \right) \Delta \mathbf{W} \left( \mathbf{W}^{-1} \right) \tag{39}$$

so that our learning rule, (14), forms part of a rule for the recurrent H-J network. Unfortunately, this rule is complex and not obviously related to the non-linear anti-Hebbian rule proposed for the H-J net:

$$\Delta \mathbf{W}_{HJ} \propto -g(\mathbf{u}) h(\mathbf{u})^T \tag{40}$$

where $g$ and $h$ are odd non-linear functions. It remains to conduct a detailed performance comparison between (40) and the algorithm presented here. We have performed many simulations in which the H-J net failed to converge, but because there is substantial freedom in the choice of $g$ and $h$ in (40), we cannot be sure that our choices were good ones.

We now compare the convergence criteria of the two algorithms in order to show how they are related. The explanation (Jutten & Herault 1991) for the success of the H-J network is that the Taylor series expansion of $g(\mathbf{u})h(\mathbf{u})^T$ in (40) yields odd cross moments, such that the weights stop changing when:

$$\sum_{i,j} b_{ijpq} \langle u_i^{2p+1} u_j^{2q+1} \rangle = 0 \tag{41}$$

for all output unit pairs $i \neq j$, for $p, q = 0, 1, 2, 3 \ldots$, and for the coefficients $b_{ijpq}$ coming from the Taylor series expansion of $g$ and $h$. This, they argue, provides an "approximation of an independence test".

This can be compared with the convergence criterion of our algorithm. For the tanh non-linearity, we derive:

$$\Delta\mathbf{W} \propto \left[\mathbf{W}^T\right]^{-1} - 2\mathbf{y}\mathbf{x}^T \tag{42}$$

This converges in the mean when (ignoring bias weights and assuming $\mathbf{x}$ to be zero mean):

$$\left[\mathbf{W}^T\right]^{-1} = 2\langle \tanh(\mathbf{W}\mathbf{x})\mathbf{x}^T\rangle. \tag{43}$$

This condition can be readily rewritten (multiplying in by $\mathbf{W}^T$ and using $\mathbf{u} = \mathbf{W}\mathbf{x}$) as:

$$\mathbf{I} = 2\langle \tanh(\mathbf{u})\mathbf{u}^T\rangle. \tag{44}$$

Since tanh is an odd function, its series expansion is of the form $\tanh(u) = \sum_j b_j u^{2p+1}$, the $b_j$ being coefficients, and thus the convergence criterion (44) amounts to the condition

$$\sum_{i,j} b_{ijp}\langle u_i^{2p+1} u_j\rangle = 0 \tag{45}$$

for all output unit pairs $i \neq j$, for $p = 0, 1, 2, 3 \ldots$, and for the coefficients $b_{ijp}$ coming from the Taylor series expansion of the tanh function.

The convergence criterion (45) involves fewer cross moments than that of (41) and in this sense, may be viewed as a less restrictive condition. More relevant, however, is the fact that the weighting, or relative importance, $b_{ijp}$, of the moments in (45) is determined by the information theoretic objective function in conjunction with the non-linear function $g$, while in (41), the $b_{ijpq}$ values are accidents of the particular non-linear functions, $g$ and $h$, that we choose. These observations may help to explain the existence of spurious solutions for H-J, as revealed, for example, in the stability analysis of Sorouchyari (1991).

Several other approaches to blind separation exist. Comon (1994) expands the mutual information in terms of cumulants up to order 4, amounting to a truncation of the constraints in (45). A similar proposal which combines separation with deconvolution is to be found in Yellin & Weinstein (1994). Such cumulant-based methods seem to work, though they are complex. It is not clear how the truncation of the expansion affects the solution. In addition, Molgedey & Schuster (1994) have proposed a novel technique that uses time delayed correlations to constrain the solution. Finally, Hopfield (1991) has applied a variant of the H-J architecture to odor separation in a model of the olfactory bulb.

## 6.3 Comparison with previous work on blind deconvolution

In the case of blind deconvolution, our approach most resembles the 'Bussgang' family of techniques (Bellini 1994, Haykin 1991). These algorithms assume some knowledge about the input distributions in order to sculpt a non-linearity which may be used in the creation of a memoryless conditional estimator for the input signal. In our notation, the non-linearly transformed output, $y$, is exactly this conditional estimator:

$$y = g(u) = E[s|u] \tag{46}$$

and the goal of the system is to change weights until $u$, the actual output is the same as $y$, our estimate of $s$. An error is thus defined, $error = y - u$, and a stochastic weight update rule follows directly from gradient descent in mean-squared error. This gives the blind deconvolution rule for a tapped delay weight at time $t$ [compare with (24)]:

$$\Delta w_{L-j}(t) \propto x_{t-j}(y_t - u_t) \tag{47}$$

If $g(u) = \tanh(u)$ then this rule is very similar to (24). The only difference is that (24) contains the term $\tanh(u)$ where (47) has the term $u - \tanh(u)$, but as can be easily verified, these terms are of the same sign at all times, so the algorithms should behave similarly.

The theoretical justifications for the Bussgang approaches are, however, a little obscure, and, as with the Herault-Jutten rules, part of their appeal derives from the fact that they have been shown to work in many circumstances. The primary difficulty lies in the consideration, (46), of $y$ as a conditional estimator for $s$. Why, *a priori*, should we consider a non-linearly transformed output to be a conditional estimator for the unconvolved input? The answer comes from Bayesian considerations. The output, $u$, is considered to be a noisy version of the original signal, $s$. Models of the pdf's of the original signal and this noise are then constructed, and Bayesian reasoning yields a non-linear conditional estimator of $s$ from $u$, which can be quite complex [see (20.39) in Haykin 1991]. It is not clear, however, that the 'noise' introduced by the convolving filter, $a$, is well-modelled as gaussian. Nor will we generally have justifiable estimates of its mean and variance, and how they compare with the means and variances of the input, $s$.

In short, the selection of a non-linearity, $g$, is a black art. Haykin does note, though, that in the limit of high convolutional noise, $g$, can be well approximated by the tanh sigmoid non-linearity [(20.44) in Haykin 1991], exactly the non-linearity we have been using. Could it be that the success of the Bussgang approaches using Bayesian conditional estimators are due less to the exact form of the conditional estimator than to the general goal of squeezing as much information as possible through a sigmoid function? As noted, a similarity exists between the information maximisation rule (24), derived without any Bayesian modelling, and the Bussgang rule (47) when convolutional noise levels are high. This suggests that the higher-order moments and information maximisation properties may be the important factors in blind deconvolution, rather than the minimisation of a contrived error measure, and its justification in terms of estimation theory.

Finally, we note that the idea of using a variable-slope sigmoid function for blind deconvolution was first described in Haykin (1992).

## 6.4 Conclusion

In their current forms, the algorithms presented here are limited. Firstly, since only single layer networks are used, the optimal mappings discovered are constrained to be linear, while some multi-layer system could be more powerful. With layers of hidden units, the Jacobian in (13) becomes more complicated, as do the learning rules derived from it. Secondly, the networks require, for $N$ inputs, that there be $N$ outputs, which makes them unable to perform the computationally useful tasks of dimensionality reduction or optimal data compression. Thirdly, realistic acoustic environments are characterised by substantial propagation delays. As a result, blind separation techniques without adaptive time delays do not work for speech recorded in a natural environment. An approach to this problem using 'beamforming' may be found in (Li & Sejnowski 1994). Fourthly, no account has yet been given for cases where there is known noise in the inputs. The beginning of such an analysis may be found in Nadal & Parga (1995), and Schuster (1992) and it may be possible to define learning rules for such cases.

Finally, and most seriously from a biological point of view, the learning rule in equation (16) is decidedly non-local. Each 'neuron' must know the cofactors either of all the weights entering it, or all those leaving it. Some architectural trick may be found which enables information maximisation to take place using only local information. The existence of local learning rules such as the H-J network, suggests that it may be possible to develop local learning rules approximating the non-local ones derived here. For now, however, the network learning rule in (14) remains unbiological.

Despite these concerns, we believe that the information maximisation approach presented here could serve as a unifying framework that brings together several lines of research, and as a guiding principle for further advances. The principles may also be applied to other sensory modalities such as vision, where Field (1994) has recently argued that phase-insensitive information maximisation (using only second order statistics) is unable to predict local (non-Fourier) receptive fields.

# Appendix — proof of learning rule (14)

Consider a network with an input vector $\mathbf{x}$, a weight matrix $\mathbf{W}$, a bias vector $\mathbf{w}_0$ and a non-linearly transformed output vector $\mathbf{y} = g(\mathbf{u})$, $\mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{w}_0$. Providing $\mathbf{W}$ is a square matrix and $g$ is an invertible function, the multivariate probability density function of $\mathbf{y}$ can be written (Papoulis, eq. 6-63):

$$f_{\mathbf{y}}(\mathbf{y}) = \frac{f_{\mathbf{x}}(\mathbf{x})}{|J|} \tag{48}$$

where $|J|$ is the absolute value of the Jacobian of the transformation. This simplifies to the product of the determinant of the weight matrix and the derivatives, $y_i'$, of the outputs, $y_i$, with respect to their net inputs:

$$J = (\det \mathbf{W}) \prod_{i=1}^{N} y_i' \tag{49}$$

For example, in the case where the non-linearity is the logistic sigmoid,

$$y_i = \frac{1}{1+e^{-u_i}} \quad \text{and} \quad y_i' = \frac{\partial y_i}{\partial u_i} = y_i(1-y_i). \tag{50}$$

We can perform gradient ascent in the information that the outputs transmit about inputs by noting that the information gradient is the same as the entropy gradient (2) for invertible deterministic mappings. The joint entropy of the outputs is:

$$\begin{aligned} H(\mathbf{y}) &= -E[\ln f_{\mathbf{y}}(\mathbf{y})] && (51)\\ &= E[\ln |J|] - E[\ln f_{\mathbf{x}}(\mathbf{x})] \qquad \text{from (48)} && (52) \end{aligned}$$

Weights can be adjusted to maximise $H(\mathbf{y})$. As before, they only affect the $E[\ln |J|]$ term above, and thus, substituting (49) into (52):

$$\Delta \mathbf{W} \propto \frac{\partial H}{\partial \mathbf{W}} = \frac{\partial}{\partial \mathbf{W}} \ln |J| = \frac{\partial}{\partial \mathbf{W}} \ln |\det \mathbf{W}| + \frac{\partial}{\partial \mathbf{W}} \ln \prod_i |y_i'| \tag{53}$$

The first term is the same regardless of the transfer function, and since $\det \mathbf{W} = \sum_j w_{ij} \operatorname{cof} w_{ij}$ for any row $i$, ($\operatorname{cof} w_{ij}$ being the cofactor of $w_{ij}$), we have, for a single weight:

$$\frac{\partial}{\partial w_{ij}} \ln |\det \mathbf{W}| = \frac{\operatorname{cof} w_{ij}}{\det \mathbf{W}} \tag{54}$$

For the full weight matrix, we use the definition of the inverse of a matrix, and the fact that the *adjoint* matrix, $\operatorname{adj} \mathbf{W}$, is the transpose of the matrix of cofactors. This gives:

$$\frac{\partial}{\partial \mathbf{W}} \ln |\det \mathbf{W}| = \frac{(\operatorname{adj} \mathbf{W})^T}{\det \mathbf{W}} = \left[\mathbf{W}^T\right]^{-1} \tag{55}$$

For the second term in (53), we note that the product, $\ln \prod_i y_i'$, splits up into a sum of log-terms, only one of which depends on a particular $w_{ij}$. The calculation of this dependency proceeds as in the one unit case of (8) and (9). Different squashing functions give different forms of anti-Hebbian terms. Some examples are given in Table 1.

Thus, for units computing weighted sums, the information-maximisation rule consists of an *anti-redundancy* term which always has the form of (55), and an *anti-Hebb* term which keeps the unit from saturating.

Several points are worth noting in Table 1:

1. The logistic (A) and tanh (B) functions produce anti-Hebb terms which use higher-order statistics. The other functions use the net input $u_i$ as their output variable, rather than the actual, non-linearly transformed output $y_i$. Tests have shown the erf function (D) to be unsuitable for blind separation problems. In fact, it can be shown to converge in the mean when [compare with (44)]: $\mathbf{I} = 2\langle \mathbf{u}\mathbf{u}^T \rangle$, showing clearly that it is just a decorrelator.

2. The generalised cumulative gaussian function (E) has a variable exponent, $r$. This can be varied between 0 and $\infty$ to produce squashing functions suitable for symmetrical input distributions with very high or low kurtosis. When $r$ is very large, then $g(u_i)$ is suitable for unit input distributions such as those in Fig.4. When close to zero, it fits high kurtosis input distributions, which are peaked with long tails.

3. Analogously, it is possible to define a generalised 'tanh' sigmoid (F), of which the hyperbolic tangent (B) is a special case ($r = 2$). The values of function F can in general only be attained by numerical integration (in both directions) of the differential equation, $g'(u) = 1 - |g(u)|^r$, from a boundary condition of $g(0) = 0$. Once this is done, however, and the values are stored in a look-up table, the slope and anti-Hebb terms are easily evaluated at each presentation. Again, as in section 2.5, it should be useful for data which may have flat ($r > 2$) or peaky ($r < 2$) pdf's.

4. The learning rule for a gaussian radial basis function node (G) shows the unsuitability of non-monotonic functions for information maximisation learning. The $u_i$ term on the denominator would make such learning unstable when the net input to the unit was zero.

| | Function : $y_i = g(u_i)$ | Slope : $y_i' = \dfrac{\partial y_i}{\partial u_i}$ | Anti − Hebb term : $\dfrac{\partial}{\partial w_{ij}} \ln \lvert y_i' \rvert$ |
|---|---|---|---|
| $A$ | $\dfrac{1}{1+e^{-u_i}}$ | $y_i(1-y_i)$ | $x_j(1-2y_i)$ |
| $B$ | $\tanh(u_i)$ | $1-y_i^2$ | $-2x_j y_i$ |
| $C$ | $\arctan(u_i)$ | $\dfrac{1}{1+u_i^2}$ | $-\dfrac{2x_j u_i}{1+u_i^2}$ |
| $D$ | $\mathrm{erf}(u_i)$ | $\dfrac{2}{\sqrt{\pi}} e^{-u_i{}^2}$ | $-2x_j u_i$ |
| $E$ | $\int_{-\infty}^{u_i} e^{-\lvert v \rvert^r} dv$ | $e^{-\lvert u_i \rvert^r}$ | $-r x_j \lvert u_i \rvert^{r-1} \mathrm{sgn}(u_i)$ |
| $F$ | $\int_{-\infty}^{u_i} (1 - \lvert g(v) \rvert^r) dv$ | $1 - \lvert y_i \rvert^r$ | $-r x_j \lvert y_i \rvert^{r-1} \mathrm{sgn}(y_i)$ |
| $G$ | $e^{-u_i^2}$ | $-2u_i y_i$ | $x_j \dfrac{1+2u_i^2}{u_i}$ |

Table 1: Different non-linearities, $g(u_i)$, give different slopes and anti-Hebbian terms that appear when deriving information maximisation rules using eq.(53).

## Acknowledgements

This research was supported by a grant from the Office of Naval Research. We are much indebted to Nicol Schraudolph, who not only supplied the original idea in Fig.1 and shared his unpublished calculations (schraudolph et al. 1991), but also provided detailed criticism at every stage of the work. Many helpful observations also came from Paul Viola, Barak Pearlmutter, Kenji Doya, Misha Tsodyks, Alexandre Pouget, Peter Dayan, Olivier Coenen and Iris Ginzburg.

## References

[1] Atick J.J. 1992. Could information theory provide an ecological theory of sensory processing? *Network* 3, 213-251

[2] Atick J.J. & Redlich A.N. 1993. Convergent algorithm for sensory receptive field development, *Neural Computation* 5, 45-60

[3] Baram Y. & Roth Z. 1994. Multi-dimensional density shaping by sigmoidal networks with application to classification, estimation and forecasting, CIS report no. 9420, October 1994, Centre for Intelligent systems, Dept. of Computer Science, Technion, Israel Inst. of Technology, Haifa, submitted for publication

[4] Barlow H.B. 1961. Possible principles underlying the transformation of sensory messages, in *Sensory Communication*, Rosenblith W.A. (ed), MIT press

[5] Barlow H.B. 1989. Unsupervised learning, *Neural Computation* 1, 295-311

[6] Barlow H.B. & Földiák P. 1989. Adaptation and decorrelation in the cortex, in Durbin R. et al (eds) *The Computing Neuron*, p.54-72, Addison-Wesley

[7] Battiti R. 1992. First- and second-order methods for learning: between steepest descent and Newton's method, *Neural Computation*, 4, 2, 141-166

[8] Becker S. & Hinton G.E. 1992. A self-organising neural network that discovers surfaces in random-dot stereograms, *Nature* 355: 161-163

[9] Bell A.J. & Sejnowski T.J. 1995. A non-linear information maximisation algorithm that performs blind separation. in *Advances in Neural Information Processing Systems 7*, G. Tesauro et al (eds.), MIT Press, Cambridge

[10] Bialek W. Ruderman D.L. & Zee A. 1991. Optimal sampling of natural images: a design principle for the visual system? in *Advances in Neural Information Processing Systems 3*, R.P.Lippmann et al. (eds.), Morgan-Kaufmann

[11] Bellini S. 1994. Bussgang techniques for blind deconvolution and equalisation, in [23]

[12] Burel G. 1992. Blind separation of sources: a non-linear neural algorithm, *Neural Networks* 5, 937-947

[13] Cohen M.H. & Andreou A.G. 1992. Current-Mode Subthreshold MOS Implementation of the Herault-Jutten Autoadaptive Network, *IEEE J. Solid-State Circuits* vol. 27, 5, 714-727

[14] Cohen M.H. & Andreou A.G. 1995. Analog CMOS integration and experimentation with an autoadaptive independent component analyzer, *IEEE Transactions on Circuits and Systems-II: analog and digital signal processing*, vol. 42, no. 2, 65-77

[15] Comon P., Jutten C. & Herault J. 1991. Blind separation of sources, part II: problems statement, *Signal processing* 24, 11-21

[16] Comon P. 1994. Independent component analysis, a new concept? *Signal processing* 36, 287-314

[17] Cover T.M. & Thomas J.A. 1991. *Elements of information theory*, John Wiley.

[18] Deco G. & Brauer W. 1995. Non-linear higher-order statistical decorrelation by volume-conserving neural architectures, *Neural Networks*, in press

[19] Field D.J. 1994. What is the goal of sensory coding? *Neural Computation* 6, 559-601

[20] Hatzinakos D. & Nikias C.L. 1994. Blind equalisation based on higher-order statistics, in [23], 181-258

[21] Haykin S. 1991. *Adaptive filter theory*, 2nd ed., Prentice-Hall

[22] Haykin S. 1992. Blind equalisation formulated as a self-organized learning process, *Proceedings of the 26th Asilomar conference on signals, systems and computers*, Pacific Grove, CA

[23] Haykin S. (ed.) 1994a. *Blind Deconvolution*, Prentice-Hall, New Jersey.

[24] Haykin S. (ed.) 1994b. *Neural networks: a comprehensive foundation*, MacMillan, New York

[25] Herault J. & Jutten C. 1986. Space or time adaptive signal processing by neural network models, in Denker J.S. (ed), *Neural networks for computing: AIP conference proceedings 151*, American Institute for physics, New York

[26] Hopfield J.J. 1991. Olfactory computation and object perception, *Proc. Natl. Acad. Sci. USA*, vol. 88, pp.6462-6466

[27] Jutten C. & Herault J. 1991. Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture, *Signal processing* 24, 1-10

[28] Karhunen J. & Joutsensalo J. 1994. Representation and separation of signals using non-linear PCA type learning, *Neural networks* 7, 1, 113-127

[29] Laughlin S. 1981. A simple coding procedure enhances a neuron's information capacity, *Z. Naturforsch.* 36, 910-912

[30] Linsker R. 1989. An application of the principle of maximum information preservation to linear systems, in *Advances in Neural Information Processing Systems 1*, Touretzky D.S. (ed), Morgan-Kauffman

[31] Li S. & Sejnowski T.J. 1994. Adaptive separation of mixed broadband sound sources with delays by a beamforming Herault-Jutten network, *IEEE Journal of Oceanic Engineering*, 20, 1, 73-79

[32] Linsker R. 1992. Local synaptic learning rules suffice to maximise mutual information in a linear network, *Neural Computation* 4, 691-702

[33] Molgedey L. & Schuster H.G. 1994. Separation of independent signals using time-delayed correlations, *Phys. Rev. Letts.* 72, 23, 3634-3637

[34] Nadal J-P. & Parga N. 1994. Non-linear neurons in the low noise limit: a factorial code maximises information transfer. *Network*, 5, 565-581.

[35] Papoulis A. 1984. *Probability, random variables and stochastic processes, 2nd edition*, McGraw-Hill, New York

[36] Parra L., Deco G. & Miesbach S. 1995. Statistical independence and novelty detection with information preserving nonlinear maps, *Neral Comput.*, in press

[37] Platt J.C. & Faggin F. 1992. Networks for the separation of sources that are superimposed and delayed, in Moody J.E et al (eds) *Advances in Neural Information Processing Systems 4*, p.730-737, Morgan-Kaufmann

[38] Plumbley M.D. & Fallside F. 1988. An information-theoretic approach to unsupervised connectionist models, in Touretzky D, Hinton G & Sejnowski T (eds) *Proceedings of the 1988 Connectionist Models Summer School*, p.239-245, Morgan-Kaufmann 1988

[39] Schmidhuber J. 1992. Learning factorial codes by predictability minimisation, *Neural Computation* 4, 6, 863-87

[40] Schraudolph N.N., Hart W.E. & Belew R.K. 1991. Optimal information flow in sigmoidal neurons, *unpublished manuscript*

[41] Schraudolph N.N. & Sejnowski T.J. 1992. Competitive anti-Hebbian learning of invariants, in *Advances in Neural Information Processing Systems 4*, Moody J.E. et al (eds), Morgan-Kauffman

[42] Schuster H.G. 1992. Learning by maximizing the information transfer through nonlinera noisy neurons and "noise breakdown", *Phys. Rev. A*, 46,4, 2131-2138

[43] Sorouchyari E. 1991. Blind separation of sources, part III: stability analysis, *Signal processing* 24, 1, 11-20

[44] Vittoz E.A. & Arreguit X. 1989. CMOS integration of Herault-Jutten cells for separation of sources, in Mead C. & Ismail M. (eds) *Analog VLSI implementation of neural systems*, p.57-84, Kluwer, Boston

[45] Yellin D. & Weinstein E. 1994. Criteria for multichannel signal separation, *IEEE transactions on signal processing*, vol. 42, no. 8, 2158-2168