GENERAL MULTILAYER PERCEPTRON DEMIXER SCHEME
FOR NONLINEAR BLIND SIGNAL SEPARATION

Wai Lok Woo and Sadettin Sali
Department of Electrical and Electronic Engineering
University of Newcastle upon Tyne
United Kingdom

Abstract

A new technique is presented for instantaneous blind signal separation from nonlinear mixtures using a general neural network based demixer scheme. The nonlinear demixer model follows directly from the general mixer model. Thus in the first part of the paper we present such a general mixer model which includes the linear mixtures as a special case. In the second part we present the general framework for the demixer based on feedforward multilayer perceptron (FMLP) employing a class of continuously differentiable nonlinear functions. A detailed derivation of the learning algorithm used to adapt the demixer’s parameters is given explicitly. Cost functions based on both Maximum Entropy (ME) and Minimum Mutual Information (MMI) have been studied. The performance of the new technique was investigated on various experiments derived from the general mixer model and on real time data. These studies illustrated the superiority and the generality of the new technique compared with the existing methods.

Key words: Nonlinear Blind Source Separation Problem, Multilayer Perceptrons and Independent Component Analysis.
1. INTRODUCTION

Most recently, Blind Source Separation (BSS) has received worldwide attention because of its potential application in speech recognition, multi-user cellular radio networks, signal and medical image processing. Most of the BSS algorithms are based on Independent Component Analysis (ICA). The task of BSS is to recover the unknown signal embedded in corrupted and imperfect observation signals at the receiver after a mixing system. In multi-user cellular radio applications, the mixing system is the propagation channel. It is expected that in the third generation systems, base station antennas will be adaptive arrays [10-13]. The most fundamental problem in ICA is to find a set of statistically independent components from the output of the mixing system. The success of the BSS problem is directly linked to finding a stable inverse (linear) neural network demixer system in order to recover the original sources. Most of the existing BSS techniques are based on ICA when the mixtures are linear [1-7]. For some channels, linear mixing models may provide sufficient approximations but more often, channel dynamics are more complex and require nonlinear models, such as in the satellite-mobile systems [10-11] where Travelling Wave Tube (TWT) amplifiers are used to boost the signal levels in the transponders on board the satellite. In full saturation, the amplitude gain and phase wrapping are defined as nonlinear functions of the signal passing through the TWT amplifier [11]. Another example is the recording of multiple speech sources using the microphones, which employ class-C amplifiers. In some signal and array processing applications, the components and sensor elements often exhibit nonlinear behaviour at certain signalling conditions such as beamforming [10]. Also the on-line recording of intra-neuronal activities in the brain is via non-linear mixers. In all of these examples, the most appropriate representation of the mixing system must be nonlinear. For nonlinear mixtures, the linear ICA models are not strictly applicable and the existing BSS schemes for the linear mixture models fail to extract the sources in nonlinear mixtures. The literature on BSS problem from linear mixtures is rather extensive. In contrast nonlinear mixtures are not treated extensively and the existing known methods, which employ linear neural network based demixers are not sufficiently general to be of use in practical applications [14-20].

2. NONLINEAR MIXING MODEL

The linear models are often used for both instantaneous and convolutive mixtures, by virtue of their simplicity and ease of reconstruction. However, a realistic mixture should be nonlinear and capable of
treated the linear mixture as a special case. Hence, the following LNL (linear-nonlinear-linear) cascade channel model is proposed.

![Nonlinear mixture model](image)

**Fig. 1: LNL (linear-nonlinear-linear) cascade nonlinear mixing model**

From Fig. 1 the input-output equation can be described as:

$$x(t) = f(s(t)) = C_1 h_1^{(1)} C_2 h_2^{(2)} \cdots \left( h_N^{(N)} \left( C_N s(t) \right) + \varsigma^{(N-1)} \right) + \varsigma^{(2)} + \varsigma^{(1)}$$

(1)

where $s(t)$ is a $(m \times 1)$ vector containing the source signals, $\{C_1, C_2, \ldots, C_N\}$ are the $(m \times m)$ mixing matrices, $\{\varsigma^{(1)}, \varsigma^{(2)}, \ldots, \varsigma^{(N-1)}\}$ are the thresholds and $\{h_1^{(1)}, h_2^{(2)}, \ldots, h_N^{(N)}\}$ are the zero preserving nonlinear functions employed to model the nonlinear mappings between $s(t)$ and $x(t)$ in the LNL model. The nonlinear mixing model adopted in [15] can also be considered as a special case of our LNL model. In the case of linear mixtures, all the $h_j^{(i)}$ s are simply set to a linear function, the thresholds $\varsigma^{(i)}$ s set to null and hence the general model becomes the composite product of linear matrices $C = C_1 \times C_2 \times C_3 \times \cdots \times C_N$, which is linear. Hence for a linear mixture model, the input-output equation is described as:

$$x(t) = \left[ C_1 \times C_2 \times C_3 \times \cdots \times C_N \right] s(t)$$

(2)

Fig. 1 depicts the LNL (linear-nonlinear-linear) cascade model with $N - 1$ layers of nonlinearity. The appropriate form of nonlinear functions and the number of nonlinearity layers are crucial in order to model correctly the demixer for successful source separation. Theoretically, a single layer of nonlinearity coupled in between two mixing matrices has been shown to approximate any arbitrary continuous
function [6,8]. In particular, any sigmoidal nonlinear function can be used as the nonlinearity in the general model since it is a non-constant, bounded and monotonically increasing function. Nevertheless, in practical cases, using only one layer of nonlinearity does not necessarily provide the required optimality in modelling most real-life-mixing environment. Especially knowing that nonlinearity exists not only in the mixing propagation medium but also during the generation of the sources and on reception of the mixture outputs [11,14,15]. Besides, the universal approximation theorem [8] assumes that there exists an unlimited size of neurons in the network i.e. $m \to \infty$ in $h_m^{(l)}$. Clearly, it is inappropriate in practice to model such mixtures with an infinite number of nonlinear neurons in a single layer, as in BSS context, this will imply that no unique solution can be guaranteed to exist. However, to ensure the existence of a unique solution, the condition described in the next section needs to be considered.

3. EXISTENCE OF UNIQUE INVERSE FUNCTION

Assume that the source signals are independently distributed i.e. $p_S(s) = \prod_j p_{S_j}(s_j)$ and mixing function $x = f(s)$ in (1) is a continuously differentiable function and that $J = \det \frac{dx}{ds}$ ≠ 0 at an arbitrary point $p$, then there exist open sets $P, Q \subseteq \mathbb{R}^m$ where $p \in P, q = f(p) \in Q$ and a uniquely determined inverse function $f^{-1}: P \to Q$ such that $f$ and $f^{-1}$ comprise a unique mapping of one-to-one on $P$ and $Q$. Moreover, $f^{-1}$ is also a continuously differentiable function and that the recovered sources via $f^{-1}$ will thus be independent.

We remark this in the following way. Upon differentiating (1),

$$\frac{dx}{ds} = C_1 \times \text{diag}[h^{(l)}(u_1)] \times C_2 \times \cdots \times \text{diag}[h^{(N-1)}(u_{N-1})] \times C_N$$

where $h^{(i)}(\cdot)$ implies element by element derivative of $h^{(i)}(u_j) = \frac{d}{du_j} h^{(i)}(u_j)$. Since the Jacobian determinant of the derivative matrix must be strictly non-zero for a unique solution to exist (i.e. $\det \frac{dx}{ds} = \left( \prod_{i=1}^N \det[C_i] \right) \left( \prod_{i=1}^{N-1} \prod_{j=1}^L h^{(i)}(u_j) \right) \neq 0$), it then follows that all the mixing matrices $C_i$ must be chosen to be full rank and that $h^{(i)}(u_j) \neq 0$ for
all \( i, j \). Since the mixing function in (1) is continuously differentiable, this demands that \( h_j^{(i)}(u_j^{(i)}) \) be continuous at all points. Thus, a sufficient condition to choose the nonlinearity is to allow \( h_j^{(i)}(\cdot) \) exists and defined at every point in the set. It is already known that [6] nonconstant, bounded and monotonically increasing function is a subset of the class of the continuously differentiable function.

4. FMLP DEMIXING MODEL AND ALGORITHM FORMULATION

In this section, a (general) feedforward multilayer perceptron (FMLP) neural network is presented as the nonlinear demixing model for blind source separation from nonlinear mixtures. The Self-Organising Maps (SOM) has been used [16] but it suffers from both network complexity and interpolation errors for continuous phase signals. It is also known that SOM is not applicable to a certain class of nonlinear functions in the mixture. Neural network models based on nonlinear ME and nonlinear MMI algorithms developed by Burel [19] which were later modified by Yang [14] are less complicated and reported to produce better results than SOM models. Similar approaches were later adopted in [17] and [18]. However Yang’s model so far is limited to 2-layer perceptrons which employ a bounded logistic function with a constant gradient. (Similar argument applies to [20]). This inadvertently sets a limitation on the class of nonlinear function in the 2-layer perceptron demixer to separate the mixture. Vaipola [20] developed a similar nonlinear scheme for training the perceptron demixer based on ensemble learning but it suffers from huge computational complexity, convergence time and therefore, lacks the practicality for on-line applications. In Maximum Likelihood (ML) context, it can easily be proven that the first order gradient of the bounded nonlinear mapping function directly maps onto the probability density function (pdf) of the source function [5]. Thus the choice and the flexibility of the nonlinear mapping in the demixer is most crucial for successful source separation. In this paper, we generalise the nonlinearity of the demixer to be given by \( L \) layers of perceptrons with variable gradient, unlike the technique proposed in [14] and [15] that consists of a 2-layer perceptron with a fixed constant gradient. Moreover, the nonlinearity assumed by the proposed MLP demixer is not necessarily limited to logistic functions only but extends to a broader class of functions. In fact, the member of the class of continuously differentiable function is a suitable candidate since \( f^{-1} \) is also a continuously differentiable function. Subsequently, we derive a general and practical framework to solve for both linear and nonlinear blind source separation using a feedforward demixer architecture [7]. Hence, the conventional linear feedforward and 2-layer
logistic perceptron demixers are treated as special cases of the general multilayer perceptron demixer introduced here. Firstly, we consider a simpler 3-layer perceptrons neural network model, shown in Fig.2, to illustrate the methodology and formulation, which are then extended to derive the general learning algorithm for the multilayer perceptron demixer. Within the hidden layers, a variable gradient in the nonlinear activation function has been introduced so that the slope can be changed and optimised accordingly with the defined cost function. In the proposed scheme, the nonlinear activation functions are constrained to be given by some zero preserving mapping functions.

Referring to Fig.2 the output of the 3-layer perceptron network is defined as

\[ y_3^{(3)} = \sum_{k=1}^{n_3} w_{3k}^{(3)} g_k^{(2)}(m_k^{(2)}y_k^{(2)} + \theta_k^{(2)}) \] with \( y_2^{(2)} = \sum_{l=1}^{n_2} w_{2l}^{(2)} g_l^{(1)}(m_l^{(1)}y_l^{(1)} + \theta_l^{(1)}) \) and \( y_1^{(1)} = \sum_{m=1}^{n_1} w_{1m}^{(1)}x_m \). In the vector form, they are equivalent to

\[ y^{(3)} = W_3 g^{(2)}(\text{diag}[M_2]y^{(2)} + \theta^{(2)}) \] with \( y^{(2)} = W_2 g^{(1)}(\text{diag}[M_1]y^{(1)} + \theta^{(1)}) \) and \( y^{(1)} = W_1 x \), where \( \{m_k^j\} \) are the diagonal elements of \( M_k \) matrix which represents the gradient of the zero preserving nonlinear activation function and \( n_k \) is the number of neurons at the \( k \)-th layer of the demixer respectively. Here, it is assumed for simplicity that \( n_k \) \( (k = 1, 2, 3) \) equals to the number of sensors, \( D \). Such a demixer scheme based on a feedforward multilayer perceptron is a general system capable of demixing both linear and nonlinear types of mixtures. When the optimum point is reached, \( f^{-1} \) can be approximated by the multilayer perceptron demixer. In the case of 3-layer perceptron, this is expressed as:

\[ f^{-1}(x(t)) = y^{(3)}(t) = W_{3,\text{opt}} g^{(2)}(\text{diag}[M_{2,\text{opt}}]W_{2,\text{opt}} g^{(1)}(\text{diag}[M_{1,\text{opt}}]W_{1,\text{opt}} x(t) + \theta_{1,\text{opt}}^{(1)}) + \theta_{0,\text{opt}}^{(1)}) \] (3)
Since we can only recover up to the scaled and permuted version of the sources, then
\[ W_{3,\text{opt}} = PDW_{3,\text{opt}} \]
where \( P \) and \( D \) are the permutation and scaling matrices. From (3), the impetus of
using more layers of nonlinearities in the demixer system is now clear and motivated from the neural
network point of view [6-9]. Assuming that \( f^{-1} \) can be approximated in the form of
\[ f^{-1}(x(t)) = A(g^{(1)}(Bx(t)) + \xi^{(1)}) \]
there is no solid evidence that this solution can be ever be reached
during training by the 2-layer perceptron demixer from arbitrary parametric initialisations. In the case
where a solution exists but cannot be reached by the 2-layer demixer perceptron but only converges to the
neighbourhood of that solution, there is still no clear assurance that the latter will correspond to the
desired solution. In fact, for information preservation to be satisfied [6], the structural complexity of the
neural network should closely match the underlying complexity of the input data and the structure of the
mixing system. Thus, it is conjectured that the use of multiple layers of nonlinearity and demixing
weights provide additional degrees of freedom in the demixer’s mapping in approximating more
accurately the inverse of the mixing function. Furthermore, an additional degree of freedom is provided
by the variable gradient matrix \( \{\mathbf{M}_k\} \) in the nonlinear function to provide more accurate mapping of the
solution on to the inverse \( f^{-1}(x(t)) \). It will be illustrated in the result section with both synthetic and
real-life experiments that improved performance figures can be achieved using the proposed multiple-
layers perceptron demixer.

The proposed FMLP demixer is a function of the weight, bias and the variable gradient nonlinear
activation function parameters \( \{m_i\} \), as illustrated in Fig.2. Thus, in what follows, the general learning
algorithm is developed to optimise the cost function and concurrently, used for updating all of these
parameters in an efficient form. Beginning with the last layer (i.e. the 3rd layer), the total differential can
be written as (see the Appendix for derivation)

\[ dy^{(3)} = d[W_3g^{(2)}(\text{diag}[M_2]y^{(2)} + \theta^{(2)})] \]
\[ = d\xi^{(3)}_3 + W_3\text{diag}[g^{(2)}]d\psi_2\text{diag}[M_2]y^{(2)} + W_3\text{diag}[g^{(2)}]\text{diag}[M_2]dy^{(2)} + W_3\text{diag}[g^{(2)}]d\theta^{(2)} \]

(4)

where \( d\xi^{(3)}_3 = dW_3W_3^{-1} \) and \( d\psi_2 = d\text{diag}[M_2]\text{diag}[M_2]^{-1} \). Similarly, the total differential of the output
from the 2nd layer could be written as follows:
\[ dy^{(2)} = d[W_2g^{(1)}(\text{diag}[M_1]y^{(1)} + \theta^{(1)})] \]
\[ = d[\xi_2]y^{(2)} + W_2\text{diag}[g^{(1)}]d\psi_4\text{diag}[M_1]y^{(1)} + W_2\text{diag}[g^{(1)}]dM_1/dy^{(1)} + W_2\text{diag}[g^{(1)}]d\theta^{(1)} \]  
(5)

where \( d[\xi_2] = dW_2W_2^{-1} \) and \( d\psi_4 = d\text{diag}[M_1]\text{diag}[M_1]^{-1} \). Finally, the total differential of the output from the input layer is given by
\[ dy^{(1)} = d[W_1x] \]
\[ = d[\xi_1]y^{(1)} + W_1dx \]  
(6)

where \( d[\xi_1] = dW_1W_1^{-1} \). The cost function for the overall network is derived, based on minimising the mutual information (MMI) [2-4] among the outputs of the last layer which can be expressed as:
\[ I(y^{(3)}) = -h(y^{(3)}) + \sum_{i=1}^{D} h_i(y_i^{(3)}) = -h(x) - \log \left| \det \frac{dy^{(3)}}{dx} \right| - \sum_{i=1}^{D} \log \tilde{q}_i(y_i^{(3)}) \]  
(7)

where \( I(y^{(3)}) \), \( h(y^{(3)}) \) and \( h_i(y_i^{(3)}) \) is the mutual information, differential joint and marginal entropy defined at the output of the 3-layer perceptron demixer respectively. The series terms \( \tilde{q}_i(y_i^{(3)}) \) represent the marginal probability density function (pdf) of the output layer and \( h(x) \) is treated as constant and thus, we only need to minimise the variable optimisation terms. Thus the effective cost function can be expressed as:
\[ J = -\log \left| \det \frac{dy^{(3)}}{dx} \right| - \sum_{i=1}^{D} \log \tilde{q}_i(y_i^{(3)}) \]
\[ = -\log \left| \det W_3 \cdot \text{diag}[g^{(2)}] \cdot \text{diag}[M_2] \cdot W_2 \cdot \text{diag}[g^{(1)}] \cdot \text{diag}[M_1] \cdot W_1 \right| - \sum_{i=1}^{D} \log \tilde{q}_i(y_i^{(3)}) \]
\[ = -\log \left| \det W_3 \right| - \log \left| \det W_2 \right| - \log \left| \det W_1 \right| - \log \left| \text{det} \text{diag}[M_2] \right| - \log \left| \text{det} \text{diag}[M_1] \right| \]
\[ - \sum_{i=1}^{D} \log \tilde{q}_i^{(1)}(m_i^{(1)}y_i^{(1)} + \theta_i^{(1)}) - \sum_{i=1}^{D} \log \tilde{q}_i^{(2)}(m_i^{(2)}y_i^{(2)} + \theta_i^{(2)}) - \sum_{i=1}^{D} \log \tilde{q}_i(y_i^{(3)}) \]  
(8)

Given the cost function in (8), the total differential of \( J \) can be obtained as:
\[ dJ = -tr[dW_3 W_3^{-1}] - tr[dW_2 W_2^{-1}] - tr[dW_1 W_1^{-1}] - tr[ddiag[M_2] diag[M_2]^{-1}] \]
\[ - tr[ddiag[M_1] diag[M_1]^{-1}] + d \left( - \sum_{i=1}^{D} \log \hat{g}_i^{(2)} (m_i^{(2)} y_i^{(2)} + \theta_i^{(2)}) \right) \]
\[ + d \left( - \sum_{i=1}^{D} \log \tilde{g}_i^{(1)} (m_i^{(1)} y_i^{(1)} + \theta_i^{(1)}) \right) + d \left( - \sum_{i=1}^{D} \log \tilde{q}_i (y_i^{(3)}) \right) \]  

(9)

where ‘\( tr \)’ is the trace operation. In order to obtain the final expression for the differential of the cost function we need to differentiate (9) for each of the variables separately. Denoting the parameters to be estimated as \( \Theta = \{W_3, W_2, W_1, M_2, M_1, \theta^{(2)}, \theta^{(1)}\} \), a very similar procedure to that described in the Appendix has been undertaken and its final expression can be obtained as followed:

\[ dJ(\Theta) = -tr[d\xi_3] - tr[d\xi_2] - tr[d\xi_1] - tr[d\psi_2] - tr[d\psi_1] + [\phi^{(3)}]^T d\psi^{(3)} \]
\[ + [\phi^{(2)}]^T \left[ d\psi_2 diag[M_2] y^{(2)} + diag[M_2] dy^{(2)} + d\theta^{(2)} \right] \]
\[ + [\phi^{(1)}]^T \left[ d\psi_1 diag[M_1] y^{(1)} + diag[M_1] dy^{(1)} + d\theta^{(1)} \right] \]  

(10)

where \( \phi^{(j)}(diag[M_j] y^{(j)} + \theta^{(j)}) = \left[ \phi_1^{(j)}(m_1^{(j)} y_1^{(j)} + \theta_1^{(j)}) \quad \cdots \quad \phi_D^{(j)}(m_D^{(j)} y_D^{(j)} + \theta_D^{(j)}) \right]^T \) and

\[ \phi_i^{(j)}(m_i^{(j)} y_i^{(j)} + \theta_i^{(j)}) = -\frac{\tilde{g}_i^{(j)}(m_i^{(j)} y_i^{(j)} + \theta_i^{(j)})}{\tilde{q}_i(y_i^{(3)})} \]

where \( \tilde{g}_i^{(j)}(m_i^{(j)} y_i^{(j)} + \theta_i^{(j)}) \) and \( \tilde{q}_i(y_i^{(3)}) \) are the first and second order derivatives of the nonlinearity with respect to the parameters. \( \phi^{(3)}(y^{(3)}) = \left[ \phi_1^{(3)}(y_1^{(3)}) \quad \phi_2^{(3)}(y_2^{(3)}) \quad \cdots \quad \phi_D^{(3)}(y_D^{(3)}) \right]^T \) with \( \phi_i^{(3)}(y_i^{(3)}) = -\frac{d \log \tilde{q}_i(y_i^{(3)})}{dy_i^{(3)}} \)

and \( \tilde{q}_i(y_i^{(3)}) \) is the marginal pdf defined at the outputs in the last layer as explained above. By considering the derivatives of the cost function with respect to the modified differentials \( d\xi_3, d\xi_2, d\xi_1, d\psi_2, d\psi_1, d\theta^{(2)}, d\theta^{(1)} \) with respect to the cost function, we can now obtain the learning algorithms used to adapt the neural network parameters. Beginning with the last layer, for the differential of the cost function with respect to the weights in the first layer, we can write

\[ \frac{dJ}{d\xi_3} = -I + \phi^{(3)} y^{(3)} = -[I - r^{(3)} y^{(3)}] \]  

(11)

where \( r^{(3)} = \phi^{(3)} \) is the score function associated with the last layer of the neural network. The weight update for the last layer can be obtained from the general definition of the steepest descent algorithm. In this case, it is the pseudo-natural gradient descent algorithm. While the notion of natural gradient [3] fits
in automatically in the linear demixer case that spans the Riemannian space, it is quite true that no such
general property associated with the latter can be found in the nonlinear case. Nevertheless, the impetus
of using such pseudo-natural gradient defined (11) is mainly invoked for avoidance in computing the
matrix inverses during the adaptation of the parameters. Because of this, the learning algorithm for the
third layer takes the following form:

$$W_3(t + 1) = W_3(t) - \frac{dW_3}{dt} = W_3(t) - \eta_w^{(3)} \frac{dJ}{d\xi_3}$$

$$= W_3(t) + \eta_w^{(3)} [I - r^{(3)}(t)y^{(3)T}(t)]W_3(t) \quad (12)$$

Similarly for the 2nd layer, we obtain the following

$$\frac{dJ}{d\xi_2} = -I + \text{diag}[M_2] \phi^{(2)}y^{(2)T} + \text{diag}[M_2] \text{diag}[g^{(2)}] W_3^T \phi^{(3)} y^{(2)T} = -[I - \text{diag}[M_2] r^{(2)} y^{(2)T}] \quad (13)$$

$$\therefore W_2(t + 1) = W_2(t) + \eta_w^{(2)} [I - \text{diag}[M_2] r^{(2)}(t)y^{(2)T}(t)]W_2(t) \quad (14)$$

where $r^{(2)} = \phi^{(2)} + \text{diag}[g^{(2)}] W_3^T \phi^{(3)}$ is the score function associated with the 2nd layer of the demixer.

Following the same steps taken from above, for the first layer weights we have

$$\frac{dJ}{d\xi_1} = -I + \text{diag}[M_1] \phi^{(1)}y^{(1)T} + \text{diag}[M_1] \text{diag}[g^{(1)}] W_2^T \text{diag}[M_2] \phi^{(2)} y^{(1)T}$$

$$+ \text{diag}[M_1] \text{diag}[g^{(1)}] W_2^T \text{diag}[M_2] \text{diag}[g^{(2)}] W_3^T \phi^{(3)} y^{(1)T}$$

$$= -[I - \text{diag}[M_1] r^{(1)} y^{(1)T}] \quad (15)$$

$$\therefore W_1(t + 1) = W_1(t) + \eta_w^{(1)} [I - \text{diag}[M_1] r^{(1)}(t)y^{(1)T}(t)]W_1(t) \quad (16)$$

where $r^{(1)} = \phi^{(1)} + \text{diag}[g^{(1)}] W_2^T \text{diag}[M_2] \phi^{(2)} + \text{diag}[g^{(1)}] W_2^T \text{diag}[M_2] \text{diag}[g^{(2)}] W_3^T \phi^{(3)}$ is the score
function associated with the 1st layer of the demixer network. Close inspection of expressions (11)-(16)
enables us to obtain the following recursive equation for the score function between the different layers in
the demixer given by:

$$r^{(j)} = \phi^{(j)} + \text{diag}[g^{(j)}] W_j^T \text{diag}[M_j] r^{(j+1)}$$

$$\quad (17)$$

Following the same procedure as previously followed in the derivation of (11)-(16) the learning algorithm
for the gradient matrices, $\text{diag}[M_2]$ and $\text{diag}[M_1]$, may also be obtained for $j=1,2$ as:
\[
\frac{dJ}{d\psi_i} = -(I - r^{(i)})[\text{diag}(M_j) y^{(j)}]^T
\]

\[
\therefore \text{diag}(M_j(t+1)] = \text{diag}(M_j(t)] + \eta_m^{(j)} \text{diag}(I - r^{(j)}(t)[\text{diag}(M_j(t)y^{(j)}(t)])^T \text{diag}(M_j(t)]
\]

Similarly, the learning algorithm for bias weights \(\theta^{(2)}\) and \(\theta^{(1)}\) may be obtained via:

\[
\frac{dJ}{d\theta^{(j)}} = \phi^{(j)} + \text{diag}[g^{(j)}] W^{T}_{j+1} r^{(j+1)} = r^{(j)}
\]

\[
\therefore \theta^{(j)}(t+1) = \theta^{(j)}(t) - \eta^{(j)} \theta^{(j)}(t)
\]

This completes the derivation for the 3-layer FMLP demixer network. Close inspection of the expressions in (11)-(21) reveals that the above derivation may easily be generalised beyond the 3-layer perceptron model. Using mathematical deduction, for a FMLP demixer with \(L\) layers, the generalised learning algorithm may be obtained as followed for \(j = 1, 2, \ldots, L\) and \(k = 1, 2, \ldots, L-1\):

\[
y^{(j)}(t) = W_j(t)g^{(j-1)}(\text{diag}(M_{j+1}(t)y^{(j-1)}(t) + \theta^{(j-1)}(t))
\]

\[
W_j(t+1) = W_j(t) + \eta^{(j)}(t)(I - \text{diag}(M_j(t)r^{(j)}(t)y^{(j)}(t))^T M_j(t)]\]

\[
M_k(t+1) = \text{diag}(M_k(t) + \eta^{(k)}(t)(I - r^{(k)}(t)[\text{diag}(M_k(t)y^{(k)}(t)])^T M_k(t)]
\]

\[
\theta^{(k)}(t+1) = \theta^{(k)}(t) - \eta^{(k)}(t)r^{(k)}(t)
\]

where

\[
r^{(j)}(t) = \begin{cases} 
\phi^{(L)} & , j = L \\ 
\phi^{(j)} + \text{diag}[g^{(j)}(t)][W^{T}_{j+1}(t)\text{diag}(M_{j+1}(t)r^{(j+1)}(t)) , otherwise 
\end{cases}
\]

with the following initialisations \(y^{(0)}(t) = x(t), \theta^{(0)}(t) = 0, M_0(k) = I = M_L(k)\) and \(g^{(0)}(y^{(0)}(t)) = x(t)\). The number of nonlinear neurons in each layer of the FMLP can be selected \textit{a priori}. As a rule of thumb for good source separation in nonlinear mixture, one must choose \(n_k \geq \text{number of sources}\) in all layers so as to maximise the information transferred across the demixer.

Nonetheless, the computational complexity of the demixer increases as \(n_k\) increases but this can be dealt with using network pruning techniques. When the cost function is based on minimising the mutual information (MMI), the score function at the output layer \(r^{(L)}(t) = \phi^{(L)}(y^{(L)}(t))\) is given by the probability series expansion around the gaussian density which is expressed in terms of either the
truncated Gram-Charlier or the Edgeworth series [2-4]. Alternatively, different cost function can also be implemented such as maximising the entropy of the outputs (i.e. ME method [5]) simply by replacing the score function with $r^{(L)}(t) = \tanh(0.5y^{(L)}(t))$ or $r^{(L)}(t) = 2\tanh(y^{(L)}(t))$ when a logistic or hyperbolic tangent function is used at the relevant layer respectively. In the ME approach, the choice of nonlinearity used in the demixer is crucial since it dictates how well the separation solution can be reconstructed. Table 1 (in the Appendix) illustrates a set of functions that can be used to represent the bounded nonlinear mapping function in the demixer. The set includes a sub-class of trigonometric functions commonly used in the neural network community. In the above derivation, we have exclusively used the pseudo-natural gradient for updating the demixer’s parameters on-line. Alternatively, one could attempt to use other optimisation schemes such as the Conjugate Gradient, Quasi-Newton and BFGS algorithms for the off-line adaptations. Although these methods normally outperform the conventional gradient-type of algorithms in general applications, nevertheless, none of these methods are guaranteed to offer better advantages than the pseudo-natural gradient in terms of convergence, computational complexity and performance evaluation in the source separation problem. Moreover, one must be extremely careful in choosing the appropriate initialisations and step sizes in those alternative algorithms.

5. RESULTS

The global rejection index [2] is widely used in assessing the performance of the linear ICA algorithms. However, this measure is unsuitable in the nonlinear ICA context and a new measure is needed. The new index follows a more direct approach to evaluate the performance of any information theoretic algorithm, which is to measure the information transfer across the demixer according to the defined cost function.

Referring to the defining equations for the generalised learning algorithm for the $L$-layer demixer network given in (22)-(24) the performance index for the mutual information cost function may be defined as follows:

$$\Omega_{MMI} = -\log \left| \det \prod_{l=1}^{L} W_{l} \text{diag}[M_{l-1}] \right| - \sum_{j=1}^{L} \sum_{i=1}^{N} \log g_{i}^{(j)}(\cdot) + \sum_{i=1}^{N} h_{i}(y^{(L)}_{i})$$

(25)

The last term can be approximated using the Gram-Charlier series. For maximum entropy (ME) method, similar performance index can be developed but with some slight changes for the sign and the parameters in the last term i.e.
\[
\Omega_{ME} = \log \left| \prod_{l=1}^{L} W_l \text{diag}[M_{l-1}] \right| + \sum_{j=1}^{L-1} \sum_{i=1}^{N} \log(g^{(j)}_{l,i}(\cdot)) + \sum_{i=1}^{N} \log(g^{(L)}_{l,i}(\cdot))
\]  

(26)

Although these indices do not always provide accurate indication about the separated sources except in the case the original sources are known a priori, such an approach can be used to quantify the information transfer across the demixer. In the first case, the smaller \( \Omega_{MMI} \) is, the better is the performance of the algorithm since it implies a lesser amount of mutual information is available at the output of the demixer. On the other hand as in the ME case, the algorithm should maximise \( \Omega_{ME} \). Due to the limitation of space, only two brief experimental results are presented to demonstrate the efficacy of the proposed FMLP algorithms.

In the first set of experiments, five sub-gaussian sources are generated synthetically from a computer with 25dB white gaussian noise perturbing each of the sensor. The mixture is given by \( C_3 \tanh(C_2 \tanh(0.5C_1 s)) \) where \( C_1 \), \( C_2 \) and \( C_3 \) are randomly generated mixing matrices. The sources are displayed in Fig.1(a) and the output of the mixture in Fig.1(b). Since the sources are all sub-gaussian distributed, we choose the mutual information cost function to train the demixers where the truncated 4th order Gram-Charlier series expansion is used to estimate the marginal entropy. In the following, we simulated the performances of (i) linear feedforward (ii) 2-layer (5-5) perceptron with fixed gradient logistic nonlinearity (iii) 2-layer (5-5) perceptron with variable gradient inverse hyperbolic tangent (iii) 3-layer (5-5-5) perceptron with variable gradient inverse hyperbolic sine in the first layer and variable gradient inverse hyperbolic tangent in the second layer. As mentioned in Section 5, the choice of the nonlinear mapping function is critical for the convergence of the learning algorithm to the right solution which explains the reason for selecting the different sigmoidal functions in the different layers of the mixer in simulations. In each case, all the possible nonlinear mapping functions (e.g. Table 1) have been attempted. Those producing the best results retained and showed in Fig.3 for each approach tested. The same strategy is also followed for Experiment 2 illustrated later in this section. Therefore, the results plotted for each scheme represents the best case. All the weights are initialised randomly and the step sizes used to update the cumulants in the Gram-Charlier series is set at 0.01 while for the parameter updates of the weights and thresholds at 0.001. The recovered sources of all the demixer schemes are displayed in Figs. 1(c)-(f). The results from the 2-layer perceptron demixer with fixed gradient logistic nonlinearity show an improvement over the linear scheme. An offset in one of the recovered sources has occurred due to the asymmetrical nature of the logistic function. The results from the 2-layer perceptron
demixer with variable gradient hyperbolic nonlinearity show a further improvement in the recovered sources. Finally, the results of the 3-layer perceptron give the best performance of all the different approaches. The performance indices of all the demixer schemes over 100 iterations are displayed in Fig. 1(g). This plot allows us to compare and quantify the performance of each demixer in minimising the mutual information. From the plot, the linear demixer has the highest index while the 3-layer perceptron demixer has the lowest and therefore, we conclude that the latter results in the minimal mutual information compared to the rest.

In the second set of experiments, real-life recording of speech signal mixture is used. The experiments were conducted in an auditorium and acoustic absorbers were used to avoid the echoes. The recordings were taken between the two speakers whilst a piece of music was playing in the background with reasonably audible volume. In the set up the distance between the sources and the microphones was 2 meter. We allowed the recording amplifier to operate in the saturation region (class-C operation). The recorded signals were sampled at 24K bits per second. The original source recordings are displayed in Fig. 2(a) along with the received signals at the input of the demixer in Fig. 2(b). The demixers used were similar to those used in the first example except that the logistic nonlinearity has been replaced by inverse hyperbolic sine function. The cost function used is the maximum entropy (ME). Fig. 2(c) shows the results of the linear demixer scheme. As expected, the linear ICA scheme is incapable of recovering the sources completely. Fig. 2(d) shows the results of the 2-layer perceptron demixer with fixed gradient inverse hyperbolic sine nonlinearity. Although it is evident that the recovery of sources is quite successful, the interfering source is still audible to the perceiving ear. The results of the 2-layer perceptron demixer with variable gradient hyperbolic tangent nonlinearity are displayed in Fig. 2(e). It shows further improvement over the fixed gradient hyperbolic sine perceptron demixers. Finally, Fig. 2(f) shows the results of the 3-layer perceptron demixer. From the plot, the two recovered signals are almost identical to the sources and the interfering source is much less audible compared to the previous results. The performance index of the demixer based on ME cost function is plotted in Fig. 2(g). This plot clearly identifies the entropy maximisation of each demixer for 1000 iterations. Amongst all the schemes tested, the 3-layer perceptron demixer configuration results in the maximal entropy.
6. CONCLUSION

A new demixer scheme is developed for the BSS problem for nonlinear mixtures using a multilayer perceptron neural network employing the class of zero preserving continuously differentiable nonlinear activation function. It is shown that the proposed algorithm is most general as it not only pertains the architecture for the nonlinear ICA problem but also the linear case by simply modifying the nonlinear activation function \( g_i^{(l)}(\cdot) \) to be a linear function. The results from the experiments have shown that the 3-layer perceptron demixer outperforms both the linear and the 2-layer perceptron demixers, which is found to be very limited. However, it must be borne in mind that a successful nonlinear reconstruction problem must still be explored further as several research issues exist which need to be addressed.

7. ACKNOWLEDGEMENTS

The authors are grateful to Qinetiq at Malvern for its supports and the discussions with Professor I.J. Clarke greatly contributed to this work. The authors would also like to thank the anonymous reviewers for greatly enhancing the insight into the first draft of this manuscript.

8. APPENDIX

9.1 Total differential of \( y^{(3)} \).

\[
dy^{(3)} = d[W_3g^{(2)}(\text{diag}[M_2]y^{(2)} + \theta^{(2)})]
\]

\[
= dW_3g^{(2)}(\text{diag}[M_2]y^{(2)} + \theta^{(2)}) + W_3d\text{diag}[g^{(2)}]d(\text{diag}[M_2]y^{(2)} + \theta^{(2)})
\]

\[
= dW_3g^{(2)}(\text{diag}[M_2]y^{(2)} + \theta^{(2)}) + W_3d\text{diag}[g^{(2)}](d\text{diag}[M_2]y^{(2)} + d\text{diag}[M_2]dy^{(2)} + d\theta^{(2)})
\]

(\text{1})

where \( \text{diag}[g^{(j)}] = \text{diag}[\hat{g}_1^{(j)} \hat{g}_2^{(j)} \cdots \hat{g}_N^{(j)}] \) and \( \frac{d\hat{g}_i^{(j)}}{d(m_i^{(j)} y_i^{(2)} + \theta_i^{(j)})} \) is the first order derivative with respect to its input parameters. Assuming the existence of full rank weights \( W_3 \) and invertible function \( g^{(2)} \), \( y^{(2)} \) could be expressed in terms of \( y^{(3)} \) simply as \( y^{(2)} = \text{diag}[M_2]^{-1} [g^{(2)}]^{-1} [W_3^{-1} y^{(3)}] - \theta^{(2)} \). Substituting this result to the above equation yields
\[ dy^{(3)} = dW_3 y^{(3)} + W_3 \text{diag}[g^{(2)}](d \text{diag}[M_2]y^{(2)} + \text{diag}[M_2]dy^{(2)} + d\theta^{(2)}) \]
\[ = dW_3 y^{(3)} + W_3 \text{diag}[g^{(2)}]d \text{diag}[M_2]y^{(2)} + W_3 \text{diag}[g^{(2)}]d \text{diag}[M_2]dy^{(2)} + W_3 \text{diag}[g^{(2)}]d\theta^{(2)} \]
\[ = dW_3 y^{(3)} + W_3 \text{diag}[g^{(2)}]d \text{diag}[M_2]y^{(2)} + W_3 \text{diag}[g^{(2)}]d \text{diag}[M_2]^{-1} \text{diag}[M_2]y^{(2)} + W_3 \text{diag}[g^{(2)}]d\theta^{(2)} \]
\[ + W_3 \text{diag}[g^{(2)}]d \text{diag}[M_2]dy^{(2)} \]

(a2)

9.2 Total differential of \( y^{(2)} \).

Assuming the existence of full rank weights \( W_2 \) and invertible function \( g^{(1)} \),

\[ dy^{(2)} = d[W_2 g^{(1)}(\text{diag}[M_1]y^{(1)} + \theta^{(1)})] \]
\[ = dW_2 g^{(1)}(\text{diag}[M_1]y^{(1)} + \theta^{(1)}) + W_2 \text{diag}[g^{(1)}]d(\text{diag}[M_1]y^{(1)} + \theta^{(1)}) \]
\[ = dW_2 g^{(1)}(\text{diag}[M_1]y^{(1)} + \theta^{(1)}) + W_2 \text{diag}[g^{(1)}]d(\text{diag}[M_1]y^{(1)} + \text{diag}[M_1]dy^{(1)} + d\theta^{(1)}) \]
\[ = dW_2 y^{(2)} + W_2 \text{diag}[g^{(1)}]d(\text{diag}[M_1] \text{diag}[M_1]^{-1} \text{diag}[M_1]y^{(1)} + W_2 \text{diag}[g^{(1)}]d\theta^{(1)} \]
\[ + W_2 \text{diag}[g^{(1)}]d(\text{diag}[M_1]dy^{(1)} \]

The derivation of the first layer output and optimisation of the cost function in the general learning algorithm, with respect to different parameters, follows the same procedure outlined above. In the derivation of the output, it is important that the nonlinear mapping functions used are continuously differentiable functions and a set of these is given in Table 1 that forms a special class of trigonometric functions (although it is unnecessarily limited to this class alone).

<table>
<thead>
<tr>
<th>Function g(u)</th>
<th>( \dot{g}(u) = \frac{d}{du} g(u) )</th>
<th>( \phi(u) = -\frac{\dot{g}(u)}{g(u)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( \frac{1}{1+e^{-u}} )</td>
<td>( g(u)[1 - g(u)] )</td>
<td>( \tanh\left(\frac{1}{2}u\right) )</td>
</tr>
<tr>
<td>2. ( \tanh(u) )</td>
<td>( \text{sech}^2(u) )</td>
<td>( 2 \tanh(u) )</td>
</tr>
<tr>
<td>3. ( \sinh(u) )</td>
<td>( \cosh(u) )</td>
<td>( -\tanh(u) )</td>
</tr>
<tr>
<td>4. ( \tanh^{-1}(u) )</td>
<td>( \frac{1}{1-u^2} )</td>
<td>( \frac{2u}{u^2 - 1} )</td>
</tr>
<tr>
<td>5. ( \sinh^{-1}(u) )</td>
<td>( \frac{1}{\sqrt{1+u^2}} )</td>
<td>( \frac{u}{1+u^2} )</td>
</tr>
</tbody>
</table>

Table 1: A sub-class of candidates for the FMLP demixer nonlinear functions
9. REFERENCES


Figure Captions:

Fig. 1 LNL (linear-nonlinear-linear) cascade nonlinear mixing model.

Fig. 2 3-layer variable gradient perceptron as the specific LNL cascade demixing model.

Fig. 3 Performance of the FMLP algorithm on synthetically generated source signals. (a) The original sources. (b) The nonlinearly mixed signals embedded in Gaussian noise. (c) Results from linear demixer [7]. (d) Results from nonlinear demixer using 2-layer fixed gradient logistic function [12]. (e) Results from FMLP using 2-layer variable gradient $\tanh(\cdot)$ mapping function. (f) Results from FMLP with 3-layer variable gradient $\sinh^{-1}(\cdot), \tanh(\cdot)$. (g) Performance index of each demixer.

Fig. 4 Performance of the FMLP algorithm on real time recorded acoustic signals. (a) The recorded speech signals. (b) The nonlinearly mixed signals. (c) Results from a linear demixer [7]. (d) Results from a nonlinear demixer with 2-layer fixed gradient with $\sinh^{-1}(\cdot)$ [12]. (e) Results from FMLP with 2-layer variable gradient $\tanh(\cdot)$ mapping function. (f) Results from FMLP with 3-layer variable gradient with $\sinh^{-1}(\cdot), \tanh(\cdot)$ mapping functions. (g) Performance index of each demixer.

List of Table:

Table 1 A sub-class of candidates for the FMLP demixer nonlinear functions