Separation of nonlinearly-mixed Sources using end-to-end Deep Neural Networks

Hojatollah Zamani, Saeed Razavikia, Hatef Otroshi-Shahreza, Arash Amini *Senior Member, IEEE*

***Abstract*—In this paper, we consider the problem of blind source separation under certain nonlinear mixing conditions using a deep learning approach. Conventionally, the separation of sources within linear mixtures is achieved by applying the independence property of the sources. In the nonlinear regime, however, this property is no longer sufficient. In this paper, we consider nonlinear mixing operators where the non-linearity could be fairly approximated using a Taylor series. Next, for solving the nonlinear BSS problem, we design an end-to-end recurrent neural network (RNN) that learns the inverse of the system, and ultimately separates the sources. For training the RNN, we employ a set of multi-variate polynomial functions to simulate the Taylor expansion of the nonlinear mixture. Numerical experiments show that the proposed method successfully separates the sources with a performance superior to the recent approach devised in [1].**

I. Introduction

The problem of blind source separation (BSS) is well-known and well-studied in the signal processing community [2], [3]. In simple words, a number of source signals are combined in a specific but unknown way to generate the observations. Then, the objective is to reconstruct the sources based on the observations. The blindness of the method refers to the fact that the exact structure of the mixing operator is not known; however, a general mixing model (such as linearity) is available. The problem of BSS has applications in various fields; most notably, it is used for separation of audio signals, e.g., isolation of the speech of a single person from the recordings of a group of people talking simultaneously. Other applications include separation of EEG and ECG signals [4], images separation, feature extraction, and wireless communication [3]. The BSS problem is generally ill-posed in the sense that the solution is not unique. Hence, it is common to include the statistical properties of the sources or the general model of the mixing operator, or both. In the conventional linear case where the independence of the sources is assumed, there is still ambiguity in the amplitude and order of the sources.

The BSS problem is commonly solved by first estimating the mixing operator, then, forming an inverse operator and finally, applying the inverse operator to the observations to obtain the sources. The availability of the statistics of the sources is helpful in estimating the mixing and the inverse-mixing operators [3]. In fact, the application of the correct inverse operator to the observations shall result in signals with matching statistics. The linear mixing model is possibly the simplest; in this model, the observations are found by $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$, where $\mathbf{s}(t)$, $\mathbf{x}(t)$, and $\mathbf{A}$ represent the vector of all sources at time t, the vector of observations at time t, and the mixing operator as a matrix, respectively. When $\mathbf{A}$ is a square matrix (but unknown), the recovery of $\mathbf{s}(t)$ was addressed in [5] with the assumption that the sources are statistically independent (independent component analysis or ICA). The estimation of matrix $\mathbf{A}$ requires a measure of independence between two random variables. In other words, instead of just determining whether two random variables are independent or not, we need to quantify to what extent the independence condition holds. With this measure, $\mathbf{A}$ could be found by maximizing an overall independence measure among the sources. The choice of this measure is not unique and various techniques are proposed, such as HOSVD [6] and JADE [7]. An adaptive approach is also studied in [8], known as EASI. Another popular assumption instead of independence is the sparsity of the sources [3].

While the linear mixing model fits in numerous applications, there are practical settings in which the physics of the problem impose non-linearity. Examples include hyperspectral imaging [9], [10], remote sensing [11] and removing show-through in scanned documents [12]. In contrast to linear BSS, there are no general theoretical results on the separability and identifiability of the sources in a nonlinear regime. To better highlight the distinction between linear and nonlinear regimes, let us consider the following example from [13]:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \cos\big(\alpha(\mathbf{s}(t))\big) & -\sin\big(\alpha(\mathbf{s}(t))\big) \\ \sin\big(\alpha(\mathbf{s}(t))\big) & \cos\big(\alpha(\mathbf{s}(t))\big) \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \quad (1)$$

where $\alpha(\mathbf{s}(t)) = \theta_0(1 - r_{\mathbf{s}})^n$ for $0 \leq r_{\mathbf{s}} \leq 1$ and zero otherwise, given that $r_{\mathbf{s}} = \sqrt{s_1^2(t) + s_2^2(t)}$, $\theta_0 \in \mathbb{R}^+$ and $n \in \mathbb{N}$. It is shown in [13] that if the sources follow a point-wise i.i.d. uniform distribution in the interval $[-1, 1]$, the observations $x_1(t)$, $x_2(t)$ also exhibit the same distribution. Therefore, the ICA criterion cannot determine whether the observation are the same as the sources or they need to be unmixed. While some relatively successful non-linear ICA methods are proposed in the past (e.g., [14]), the above example reveals that the independence condition is not generally sufficient. Recently, the nonlinear BSS problem is addressed in [1] by approximating the nonlinear operator as a locally linear operator in each interval. This simplifies the task into the linear BSS problem in each interval.

In this work, we replace the algebraic procedure of estimating the unmixing operator with a deep neural network (DNN). More precisely, the DNN will be responsible for automatically estimating the unmixing operator and recovering

Authors are with Advanced Communication Research Institute (ACRI), Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. (Emails:{hojatollah.zamani, saeed.razavikia, hatef.otroshi}@ee.sharif.edu, aamini@sharif.edu)

the sources. The use of DNNs in the BSS problem has been previously considered in a number of works. In [15], through learning audio signals, DNNs are employed to make the source separation process robust to non-negative matrix factorization. In a setup where the observations are generated by a linear mixing operator and are corrupted by additive Gaussian noise, a structure based on recurrent neural networks (RNN) is proposed in [16].

In this paper, we focus on the nonlinear BSS problem. Our approach is to represent the nonlinear mixing operator with its Taylor expansion. Then, we train an end-to-end RNN to estimate the inverse of the nonlinear operator using a dataset of multi-variate polynomials corresponding to the Taylor series.

The rest of the paper is organized as follow: we first review the nonlinear parametric BSS model in Section II. The main results including the new model and the structure of the DNN are covered in Section III. In Section IV we present the simulation results and finally, we conclude the paper in Section V.

II. Signal model

In general, we assume the existence of n sources $\{s_j(t)\}_{j=1}^n$, where t represents the time instance. In the BSS problem, instead of the samples of s_js, we are given a number of mixtures (possibly nonlinear) of s_js, where the mixing rule is unknown. Let $\{x_i(t)\}_{i=1}^m$ represent the available observations (the mixtures) which follow:

$$x_i(t) = f_i\big(s_1(t), \dots, s_n(t)\big), \tag{2}$$

where $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ stands for the mixing rule that yields x_i. Here, we have implicitly assumed the simplified version of the BSS problem with the instantaneous mixing rule; in other words, each observed time sample $x_i(t)$ is generated by the samples of the sources at the same time instance (and not their past). To simplify the notations, let $\mathbf{s}(t) = [s_1(t), \dots, s_n(t)]^T$ and $\mathbf{x}(t) = [x_1(t), \dots, x_m(t)]^T$. This allows us to write

$$\mathbf{x}(t) = \mathbf{f}\big(\mathbf{s}(t)\big), \tag{3}$$

where $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$ represents the unknown measurement operator (set of all mixing rules). It is noteworthy that in practical applications, t usually belongs to a finite set such as $\{T_s, 2T_s, \dots N_{\max}T_s\}$, where T_s stands for the sampling period. In a BSS problem, we aim at recovering $\mathbf{s}(t)$ by knowing n and observing $\mathbf{x}(t)$. Traditionally, $\mathbf{f}$ needs to be estimated first in order to recover $\mathbf{s}(t)$. Indeed, $\mathbf{f}$ needs to be invertible; otherwise, even by knowing $\mathbf{f}$ one cannot uniquely recover $\mathbf{s}(t)$. For this purpose, we assume $m = n$ in this paper; however, our proposed method can work with general invertible $\mathbf{f}$.

When $\mathbf{f}$ is invertible, there exists $\mathbf{g} : \mathbb{R}^m \mapsto \mathbb{R}^n$ which acts as the inverse of $\mathbf{f}$. In this paper, instead of estimating $\mathbf{f}$ and then, finding its inverse, we directly aim at finding and implementing $\mathbf{g}$. Fig. 1 shows a system level block diagram for the special case of $m = n = 2$. It is important to highlight that when $\mathbf{f}$ is unknown, the problem inherently includes an ambiguity regarding the order of the sources; i.e., any permutation of the input sources results in the same set of observations if the input-output relationship of $\mathbf{f}$ is also permuted accordingly. Hence, for the unique recovery of the input (and $\mathbf{f}$), we shall impose asymmetric constraints on $\mathbf{f}$.

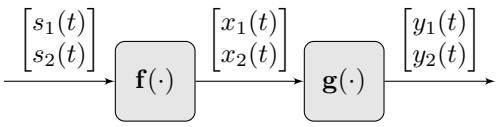


Fig. 1. Nonlinear BSS problem basic model.

A. Taylor series

Estimating a general nonlinear mixing operator $\mathbf{f}$ is very difficult in practice. Here, we restrict $\mathbf{f}$ to be smooth and continuous. Therefore, it can be fairly approximated using its truncated Taylor series around the mean of $\mathbf{s}$ in time:

$$f_i\big(\mathbf{s}(t)\big) \approx \underbrace{\sum_{\substack{j_1+\dots+j_n=0 \\ 0\le j_1,\dots,j_n}}^{j_1+\dots+j_n=N} \gamma^{(i)}_{j_1,\dots,j_n} \big(s_1(t)-\bar{s}_1\big)^{j_1} \dots \big(s_n(t)-\bar{s}_n\big)^{j_n}}_{\hat{f}_{i;N}}, \tag{4}$$

where

$$\gamma^{(i)}_{j_1,\dots,j_n} = \frac{1}{j_1! \dots j_n!} \frac{\partial^{(j_1+\dots+j_n)}}{\partial s_1^{j_1} \dots \partial s_n^{j_n}} f_i(\mathbf{s})\Big|_{\mathbf{s}=\bar{\mathbf{s}}}, \tag{5}$$

and $\bar{\mathbf{s}} = [\bar{s}_1, \dots, \bar{s}_n]$ denotes the mean of $\mathbf{s}(t)$ over time. Here N is the degree of the approximating Taylor polynomial; by increasing N, we improve the error in approximating f_i with $\hat{f}_{i;N}$. Nevertheless, the series shall consist of more terms and the model becomes more complicated.

Approximating f_is with their Taylor series enables us to parametrize the nonlinear mixing operator. This way, instead of estimating the operator $\mathbf{f}$, we can estimate these parameters.

III. Unmixing using neural networks

In this section, we introduce the proposed neural network structure for separating the sources from nonlinear mixtures. As the training phase is based on polynomials, we expect the method to work well when the nonlinear functions can be fairly approximated with their Taylor series.

A. Main strategy

Our approach in this paper is to train a neural network based on a training dataset. However, there is no available dataset that includes various types of nonlinear mixtures. For this reason, we synthetically generate combination of sources and nonlinear mixtures in form of polynomials. In this way, for each set of observations, the original sources are known. With this technique, we build a database for the training phase. If we succeed in training a suitable network, we expect the network to unmix polynomially mixed sources. This is likely to hold also for nonlinear mixtures that can be fairly approximated with their Taylor series.

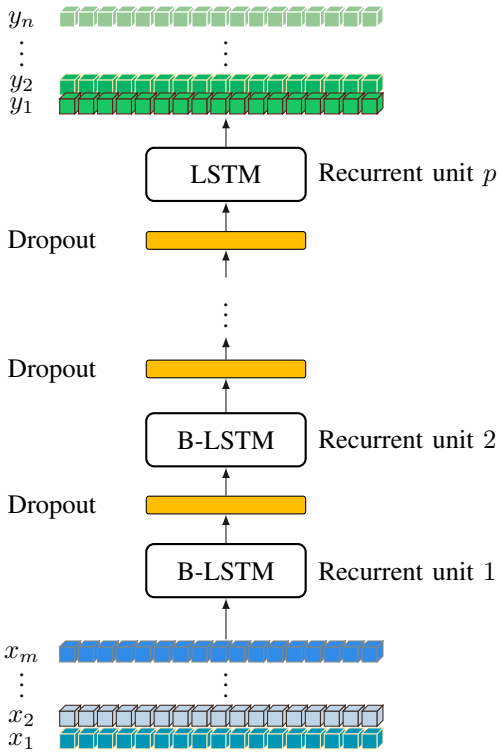


Fig. 2. Structure of the proposed neural network: The network composes with p layers, includes $p-1$ bidirectional layers and a one-directional LSTM layer. Dropout technique inserted between layers to avoid overfitting issue.

B. Structure

The input and output signals of the BSS problem can be considered as sequences. Hence, we need neural network structures that can deal with sequential data. The RNN structure is the best known example. The high-level block diagram of the RNN structure for m inputs, n outputs with k samples, and p-stacked recurrent units is drawn in Fig. 2. In this paper, we focus on the cases of $m = n = 2$ and $m = n = 3$ by setting $k = 1000$ and $p = 3$ (both cases). Also, the output size of each recurrent unit (except the last one) is set to 128. For $n > 3$, most likely we need larger p values.

We construct our end-to end network with three recurrent layers (for $p = 3$) each containing a long short-term memory (LSTM) unit [17]. The first two layers are bidirectional and the last layer is one-directional. We choose the bidirectional LSTM for the first two layers because of their better prediction performance compared to the one-directional ones [18]. In this setup, the neural network receives a block of m source mixtures and returns 128 sequences at the output of the first layer with the same length as the input block. The second layer repeats the same procedure except that it receives 128 sequences and returns 128 sequences. Finally, the third layer converts the resulting 128 sequences into n separated sources.

C. Training procedure

As explained earlier, we approximate the nonlinear mixing operator with a set of n-variable polynomials. Thus, the role of the neural network is to invert a polynomial system of equations. For the training of the network, we generate a dataset of random polynomials. For this purpose, we generate polynomials with random coefficients following uniform distribution within $[-1, 1]$; we limit the degree of the polynomials to 2 in this paper for the sake of simplicity. Next, we generate up to n random signals and combine them via the polynomial mixing operators (less than n sources is interpreted as having sources with all zero values). In this way, we generate a dataset for training the proposed neural network. Note that during the training phase, we have access to both the sources and their mixtures. The blindness of the method refers to the tests in which we only observe the mixtures.

We divide the dataset into three parts: 70% for training, 10% for validation and 20% for the test. For optimizing the weights of the neural network, the Adam optimizer [19] was used in the experiments; the values of the hyper-parameters (for the optimizer) are set as suggested in [19]. Moreover, the mean absolute error (MAE) between the predicted and source signals is used as the loss function. To avoid overfitting, we apply the dropout technique [20] with probability 0.2 at the output of each bidirectional layer.

IV. Experiments and Discussion

We assume the cases of $m = n = 2$ and $m = n = 3$ in our experiments; therefore, we are dealing with quadratic polynomials with 2 and 3 variables, respectively (i.e., polynomials with 6 and 10 coefficients, respectively). For generating the training dataset, we apply the technique of [1] for generating random signals. More precisely, we use integral of sinusoid and saw-tooth functions with varying (random) frequencies distributed uniformly within $[0, 100]$. We set the length of the signals as 1000 and extend the dataset to include $100,000$ sets of mixtures (whether $n = 2$ or $n = 3$). Our experiments are conducted on a Linux desktop computer with an Intel Core-i7 3.6GHz CPU and 32GB RAM. With this machine, the training procedure with 150 epochs took roughly 16 hours for each of the cases $n = 2, 3$.

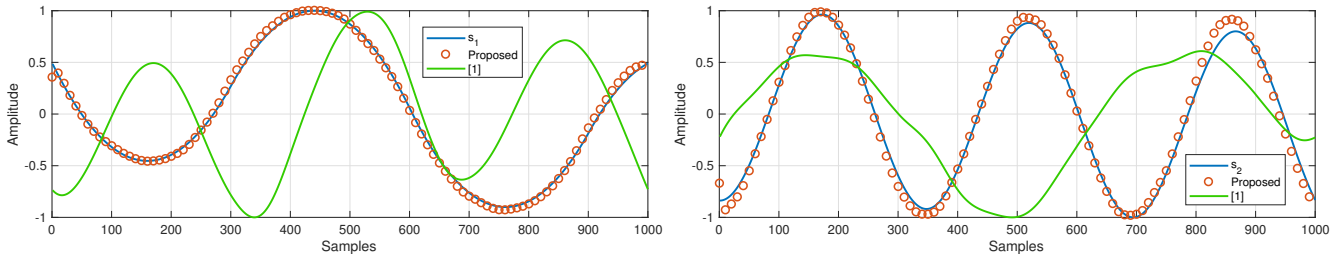
By training the network for $n = 2$ using the generated dataset, we achieve mean absolute error (MAE) values of 0.0308, 0.0312 and 0.0315 for the train, validation and test data, respectively. Also, the average MAE value of a 5-fold cross validation is obtained as 0.0307. As expected, the MAE values increase for $n = 3$; for instance, 0.266 is achieved for the test data.

A. Results

We evaluate the quality of the reconstructed signals using the two metrics of MAE and N-ENF (introduced in [1] as the *error of nonlinear fit*) defined as:

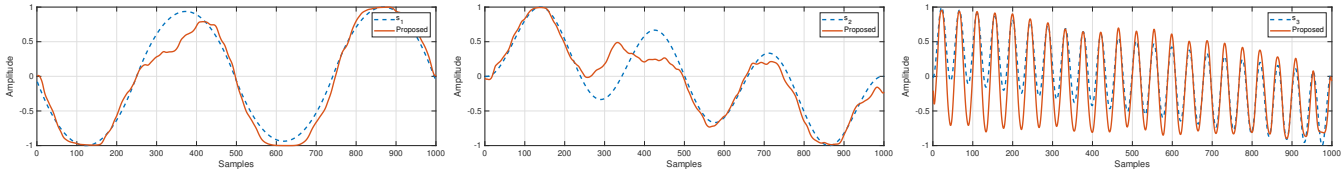
$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{s}_i - \hat{\mathbf{s}}_i\|_1, \qquad \text{N-ENF} = \frac{1}{n}\sum_{i=1}^{n}\frac{\|\hat{c}(\mathbf{s}_i) - \hat{\mathbf{s}}_i\|_2}{\|\hat{c}(\mathbf{s}_i)\|_2}, \tag{6}$$

where $\mathbf{s}_i$ and $\hat{\mathbf{s}}_i$ represent the 1000×1 vector of the ith original source and its estimate, respectively. $\hat{c}(\mathbf{s}_i)$ is a smoothing spline that minimizes $\|\hat{c}(\mathbf{s}_i) - \hat{\mathbf{s}}_i\|_2^2 + \delta\|\hat{c}''(\mathbf{s}_i)\|_2^2, \forall i \in [n]$, where $\hat{c}''(\mathbf{s})$ is the second-order time-derivative of $\hat{c}(\mathbf{s})$ and



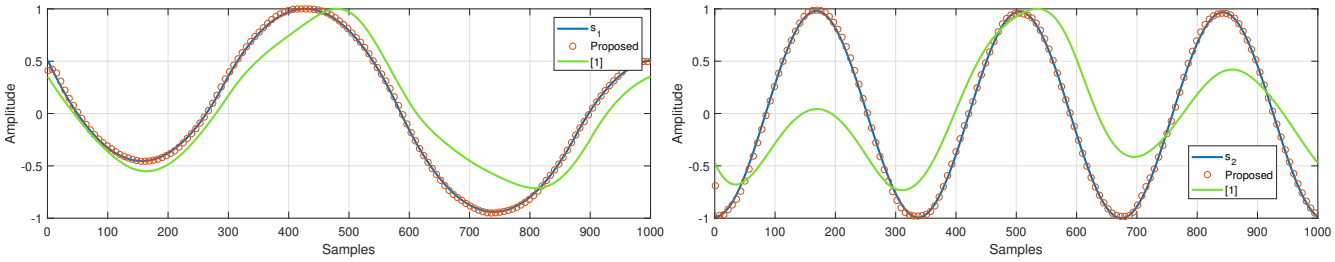
(a) (b)

Fig. 3. Comparing the result of the source separation with proposed approach and method of [1]. The considered nonlinear mixture is $x_1 = s_1 + 0.2s_2^3$, $x_2 = s_2 - 0.2s_1^3$. (a) and (b) are $s_1(t)$ and $s_2(t)$ signals with its estimation, respectively.



(a) (b) (c)

Fig. 4. The result of source separation with 3 sources using proposed approach. The used nonlinear mixture is $x_1 = -3s_2 + 0.76\cos(s_1) - \cos(2s_2) + 0.2\sin(3s_3) + s_3^2 + \exp(s_2)$, $x_2 = \exp(s_3) + 1.24\sin(s_1) + 0.43\cos(s_2) + \sin(2s_3) + s_1$, $x_3 = \exp(s_1) + 0.76\sin(s_2) + 0.43\cos(s_1) + \sin(2s_2) + \exp(s_3)$. (a), (b) and (c) are $s_1(t)$, $s_2(t)$ and $s_3(t)$ signals with its estimation, respectively.



(a) (b)

Fig. 5. Comparing the result of the source separation with proposed approach and method of [1] in linear mixture model. The considered linear mixture is $x_1 = 0.2s_1 + 0.5s_2$, $x_2 = 0.4s_1 + 0.1s_1$. (a) and (b) are $s_1(t)$ and $s_2(t)$ signals with its estimation, respectively. The MAE and N-ENF metrics for proposed method are 0.027 and 0.002, respectively. The MAE and N-ENF metrics for [1] method are 0.074 and 0.037, respectively.

TABLE I
COMPARING THE PERFORMANCE OF PROPOSED METHOD WITH [1] IN TERMS OF MAE AND N-ENF

Nonlinear Mixture	**MAE Metric**		**N-ENF Metric**	
	Proposed	**[1]**	**Proposed**	**[1]**
$x_1 = s_1 + 0.2s_2^3$ $x_2 = s_2 - 0.2s_1^3$	0.030	0.368	0.019	0.174
$x_1 = \sin(2s_1 - s_2)$ $x_2 = \sin(s_1 - s_2)$	0.018	0.503	0.017	0.135
$x_1 = \cos(\alpha_1)\, s_1 - \sin(\alpha_1)\, s_2$ $x_2 = \sin(\alpha_1)\, s_1 + \cos(\alpha_1)\, s_2$	0.044	0.397	0.003	0.450
$x_1 = \cos(\alpha_2)\, s_1 - \sin(\alpha_2)\, s_2$ $x_2 = \sin(\alpha_2)\, s_1 + \cos(\alpha_2)\, s_2$	0.046	0.334	0.003	0.088
$x_1 = s_1 \exp(s_1) - \cos(s_2)$ $x_2 = s_2 \sin(s_1) + \exp(s_2)$	0.146	0.472	0.012	0.074

$\alpha_1 = \frac{\pi}{2}\left(1 - \sqrt{s_1^2 + s_2^2}\right)^2$, $\alpha_2 = \frac{\pi}{8}\sin(\frac{s_1\pi}{\|s_1\|_\infty})\sin(\frac{s_2\pi}{\|s_2\|_\infty})$

δ is a fixed smoothing parameter [1]. Table I compares the performance of the proposed method with that of [1] in terms of MAE and N-ENF for a number of nonlinear mixtures including polynomials, trigonometric functions and exponential functions. The results reveal the superiority of the proposed method in terms of both metrics. As typical examples, a case of nonlinear mixing for 2 and another for 3 sources (and their estimated versions) are shown in Fig. 3 and Fig. 4, respectively.

In Fig. 5, we compare our method with [1] with the conventional setting of a linear mixing model for $n = 2$. The results again confirm the superiority of the proposed method.

V. CONCLUSION

In this paper, we introduced a blind source separation techniques for the case of nonlinear mixture operator. For this purpose, we employed the truncated Taylor series to model the nonlinear mixing operator. Next, we designed an end-to-end recurrent neural network to invert the nonlinear mixing operator and separate the sources. This way, the neural network is responsible for both estimating the mixing operator and find its inverse.

REFERENCES

[1] B. Ehsandoust, M. Babaie-Zadeh, B. Rivet, and C. Jutten, "Blind source separation in nonlinear mixtures: Separability and a basic algorithm," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4339–4352, Aug 2017.

[2] J. Herault and C. Jutten, "Space or time adaptive signal processing by neural network models," in *AIP Conference Proceedings 151 on Neural Networks for Computing*. Woodbury, NY, USA: American Institute of Physics Inc., 1987, pp. 206–211.

[3] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[4] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang, "BCI competition 2003-data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1067–1072, 2004.

[5] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994, higher Order Statistics.

[6] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(R1 ,R2 ,. . .,RN) approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[7] J. F. Cardoso and A. Souloumiac, "An efficient technique for the blind separation of complex sources," in *[1993 Proceedings] IEEE Signal Processing Workshop on Higher-Order Statistics*, June 1993.

[8] J. . Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, Dec 1996.

[9] N. Dobigeon, J. Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, "Nonlinear unmixing of hyperspectral images: Models and algorithms," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 82–94, Jan 2014.

[10] M. Golbabaee, S. Arberet, and P. Vandergheynst, "Compressive source separation: Theory and methods for hyperspectral imaging," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5096–5110, Dec 2013.

[11] I. Meganem, P. Deliot, X. Briottet, Y. Deville, and S. Hosseini, "Physical modelling and non-linear unmixing method for urban hyperspectral images," in *2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, June 2011, pp. 1–4.

[12] F. Merrikh-Bayat, M. Babaie-Zadeh, and C. Jutten, "Linear-quadratic blind source separating structure for removing show-through in scanned documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 4, pp. 319–333, Dec 2011.

[13] M. Babaie-Zadeh, "On blind source separation in convolutive and nonlinear mixtures," *Ph.D. dissertation, Grenoble, INPG*, 2002.

[14] L. B. Almeida, "MISEP–linear and nonlinear ICA based on mutual information," *Journal of Machine Learning Research*, vol. 4, no. Dec, pp. 1297–1318, 2003.

[15] E. M. Grais, M. U. Sen, and H. Erdogan, "Deep neural networks for single channel source separation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 3734–3738.

[16] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, Sept 2016.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.