Pang *et al.*
EURASIP Journal on Advances in Signal Processing (2022) 2022:121
https://doi.org/10.1186/s13634-022-00956-2

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

A MLE-based blind signal separation method for time–frequency overlapped signal using neural network

Lihui Pang[1,2*], Yilong Tang[1], Qingyi Tan[1], Yulang Liu[1] and Bin Yang[1]

*Correspondence: sunshine.plh@hotmail.com

[1] School of Electrical Engineering, University of South China, Hengyang 421001, China
[2] Department of Software, Sungkyunkwan University, Suwon 440746, South Korea

Abstract

The blind signal separation (BSS) algorithm obtains each original/source signal from the observed signal collected by the receiving antenna or sensor. Objective/loss/cost function and optimization method are two key parts of BSS algorithm. Modifying the objective function and optimization from the perspective of neural network (NN) is a novel concept in BSS domain. L_2 regularization is adopted as a term of maximum likelihood estimation (MLE)-based objective function like in Liu et al. (Sensors 21(3):973, 2021); however, we modified the probability density function (PDF) term of the objective function and used the kernel density estimation method for time–frequency overlapped digital communication signal. Multiple optimizers are studied in this paper, and we figure out the right optimizer for our application scenario. A varies of comparison experiments—whose separation results will be provided in forms of correlation coefficient and performance index—are carried out, which indicate our method can converge quickly and achieve satisfactory separation results with performance index (PI) lower than 0.02 when signal-to-noise ratio (SNR) no less than 10dB. Additionally, it demonstrates performance of our method is better than that of typical separation—FastICA, especially for the lower SNR environment, and it shows that our method is not sensitive to the frequency overlap level (FOL) of the source signal, even FOL as high as 100%; it still can get high-precision separation results with PI < 0.02.

Keywords: Blind signal separation, Time–frequency overlapped signal, Neural networks, Maximum likelihood estimation, Kernel density estimation

1 Introduction

In nowadays information era, the types of communication or radar electronic equipment for both military and civilian applications are increasing significantly, which leads to various communication and radar signals overcrowded in time domain, overlapped in frequency domain and intertwined in space domain shaping a more complicated electromagnetic environment [1]. As a result, the interception probability of time–frequency overlapped signals has been improved for communication reconnaissance equipment. In order to accurately capture the interested signal or discover the interference signal, separating these time–frequency overlapped signals and extracting the information implied in

the useful signal have become a research task with significant for electromagnetic surveillance domain. Since the middle of the 1990s, the blind signal separation (BSS) [2] problem—with aim of separating the source signals from mixed observation signal without knowing information of the original signal and transmission system—has been addressed by many researchers, with expertise in various domains: electromagnetic surveillance and reconnaissance, biomedical signal processing, array signal processing, speech signal processing, image processing [3], wireless communication, neural networks, etc. Many classic BSS algorithm theories have been proposed, such as independent component analysis (ICA), sparse component analysis (SCA) and nonnegative matrix factorization (NMF).

Independent component analysis (ICA) is the most popular and widely used BSS algorithm, and it is mainly used for over-determined and determined BSS—the number of mixed observation signals is more than or equal to the number of original signals to be separated and requires the source signal to have independent characteristics. Jutten et al. [4] firstly made a rigorous mathematical description for the blind signal separation problem and proposed independent component analysis (ICA). Comon [5] gave a detailed explanation of ICA and proposed the mathematical model, basic assumptions and separability conditions of ICA. Bell et al. [6] used the information theory criteria to construct the cost function combining the neural network learning algorithm successfully completed the separation task of ten speech signals. Since then, ICA has attracted the interest of many researchers and proposed many ICA-based BSS methods, such as second-order blind identification algorithm (SOBI) [7], fourth-order blind identification algorithm (FOBI) [8], joint approximate diagonalization of eigenmatrices (JADE) [9] and Fix-point ICA [10]. Among them the fix-point ICA algorithm proposed by Hyvarinen [10] with fast convergence speed and good robustness was the most popular one, well known as fast ICA (FastICA). FastICA algorithm was expanded and improved, Ollila et al. [11] provide a rigorous statistical analysis of the deflation-based FastICA estimator, Dermoune et al. [12] gave a rigorous analysis of the asymptotic errors of FastICA estimators, Wei [13] derived the general and rigorous expression of the limiting distribution and the asymptotic statistics of the FastICA algorithm, and so on. Oja et al. [14] provided a rigorous convergence analysis for FastICA. Novey et al. [15] proposed a complex fast independent component analysis (c-FastICA) algorithm to solve the ICA problems with complex-valued data. FastICA algorithm has been successfully applied in different fields, such as electroencephalography (EEG) processing [16, 17], single-channel digital communication signal separation [18], modern power systems [19] and joint radar and communication signal separation [20]. Additionally, some researchers finished the implication of FastICA algorithm, like Shyu et al. [21] implemented the FastICA algorithm in a field-programmable gate array (FPGA), with the ability of real-time sequential mixed signals processing by the proposed pipelined FastICA architecture.

Sparse component analysis (SCA) is a simple yet powerful framework for blind signal separation, especially for the under-determined signal separation—the number of mixed observation signals is less than that of original signal number, and SCA has been successfully applied in BSS for the original signal which can be represent sparsely in a given basis, even for the independence assumption is dropped [22]. SCA has been applied in image mixture separation [23–25], speech signal separation [26, 27], biological signal separation [28, 29] and so on. Reference [23, 24] separated a mixture of images using wavelet

sparsification technology. Bofill et al. [26] proposed a cluster algorithm-based—with the assumption signal has sparsity character in the frequency domain—under-determined signal separation methods for speech and music signals. Yang et al. [30] proposed a new two-stage scheme combining density-based clustering and sparse reconstruction to estimate mixing matrix and sources for speech signal separation. Li et al. [28] proposed a separation method based on SCA, which focused on the applications of sparse representation in brain signal processing, including components extraction, BSS and EEG inverse imaging, feature selection and classification. Tsouri et al. [29] proposed and evaluated a method of 12-lead electrocardiogram (ECG) reconstruction from a three-lead set. Rahbar et al. [31] discussed a frequency-domain method based on SCA for blind identification of multiple-input multiple-output (MIMO) convolutive channels driven by white quasistationary sources.

Except SCA, some under-determined BSS methods utilize nonnegative matrix factorization (NMF) to exploit the nonnegativeness signal, such as speech/audio signal [32–34], image [35] and biological signal [36]. Gao et al. [32] proposed a new unsupervised single-channel source separation method for mixed audio signal, which employed gammatone filterbank to replace time–frequency representation. Nikunen et al. [33] addressed the problem of sound source separation from a multi-channel microphone array capture via estimation of source spatial covariance matrix (SCM) of a short-time Fourier-transform mixture signal. Pezzoli et al. [34] proposed a ray-space-based multi-channel NMF method for audio source separation. Yang et al. [35] proposed an adaptive non-smooth NMF separation method for image signal. Gurve et al. [36] proposed a method for separation of fetal electrocardiogram (ECG) from abdominal ECG using activation scaled NMF. Gao et al. [37] proposed a graph-based blind hyperspectral unmixing via NMF.

BSS problem has three mainstream methods, such as ICA, SCA and NMF, but not limited to those three methods, taking source signal characteristics-based BSS method proposed in reference [38–40], for example. Szu et al. [38] proposed an effective single-channel BSS method based on the limited character set feature of digital communication signal. Warner et al. [39] presented a single-channel separation approach based on the differences between shaping filters. Pang et al. [40] proposed a novel BSS method for single-input multi-output (SIMO) system based on the periodicity of original signal, which can separate time–frequency overlapped multi-component signal effectively. Recently, a BSS method, combining the maximum likelihood estimation (MLE) criterion and a neural network (NN) with a bias term, is proposed in reference [41]. Based on this architecture, we employ neural network to implicate time–frequency signal communication signal separation based on MLE, the main difference to reference [41] is the application field, and the main innovation is that—for our application area time–frequency overlapped digital communication signal separation—we use the kernel density estimation method to estimate the probability density of the digital communication signal instead that in paper [41] using fixed function expression based on the type of the source signals.

1.1 Our contributions

The main contributions and results are summarized as follows.

- To the best of our knowledge, we are the first to explicitly explore the applicability of using neural network to accomplish time–frequency overlapped digital signals separation based on maximum likelihood estimation. In contrast, the prior work [41] employed a fixed function to express the original signals' probability density based on signal type—super-Gaussian distribution or sub-Gaussian distribution or Gaussian distribution; however, we use kernel density estimation method to estimate the probability density of the original digital communication signal, and then, the estimation results will be regarded as a term of cost function.
- We provide the cost function based on MLE—the detail will be introduced in Sect. 3.1, and we further examine the convergence and the separation performance of different optimizers, such as Adam and RMSprop, which will be provided in Sect. 4.
- We formulate critical performance metrics to evaluate the separation results, i.e., correlation coefficient (ζ) and performance index (PI), and perform an extensive evaluation of the separation methodology to validate the efficacy of the formulations. Additionally, we compare the separation performance of our method with most widely used BSS algorithms—FastICA and JADE.

1.2 Paper organization

In Sect. 2, we provide signal mix model, separation model and separation results evaluation index. In Sect. 3, we present our theoretical framework and provide each parts in detail, as signal preprocessing, cost function, probability density function estimation, optimizer and the used NN structure. We further provide a discussion together with future work in Sect. 5. In Sect. 4, by using two BPSK and one QPSK time–frequency overlapped signal as a case study, we examine and compare our separation method's performance—including comparison between different optimizers—with FastICA and JADE in terms of correlation coefficient (ζ) and performance index (PI). We conclude this work in Sect. 6.

2 Signal model

The aim of the blind signal separation is to obtain each original signal from mixed observation signal. Generally, according to whether the mixed observation signal contains reflection component or time-delay component of original signal, the signal mixed model can be divided into three types, linear instantaneous mixing model, linear delay mixing model and linear convolutional mixing model. In this paper, we are focusing on deal with the separation problem of linear instantaneous mixing model. The instantaneous linear mixture of several independent original signals can be expressed as Eq. (1).

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{v}, \tag{1}$$

where

$$\mathbf{s} = [s_1, s_2, \ldots, s_D]^T, \tag{2}$$

$$\mathbf{x} = [x_1, x_2, \ldots, x_M]^T, \tag{3}$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1D} \\ a_{21} & a_{22} & \dots & a_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M1} & a_{M2} & \dots & a_{MD} \end{bmatrix}, \tag{4}$$

where D and M represent source signal number and observation signal number, respectively. T means transpose operation. $\mathbf{A} \in \mathbb{R}^{M \times D}$ is mixed matrix, which is a full rank matrix. $\mathbf{v}$ is the additive white Gaussian noise with variance σ^2.

The signal separation system is shown in Fig. 1, $\mathbf{W} \in \mathbb{R}^{D \times M}$, stands for the unmixing or separation matrix, and our goal is to find a unmixing matrix $\mathbf{W}$ which is approximately equal to the inverse matrix $\mathbf{A}$, as shown in Eq. (5).

$$\mathbf{Wx} = \mathbf{W}(\mathbf{As} + \mathbf{v}) \approx \mathbf{Is} + \mathbf{Wv} = \hat{\mathbf{s}} + \mathbf{Wv}, \tag{5}$$

where $\mathbf{Wv}$ is the noise component; in the theoretical derivation process, we ignored this noise component, and then, Eq. (5) can be simplified as:

$$\mathbf{Wx} = \mathbf{W}(\mathbf{As}) \approx \mathbf{Is} \rightarrow \hat{\mathbf{s}}. \tag{6}$$

However, the noise component will be given full consideration in the simulation, and we will add a bias term b into the our cost function. The bias term b is the just component that represents the noise part and participates in the optimization process of the

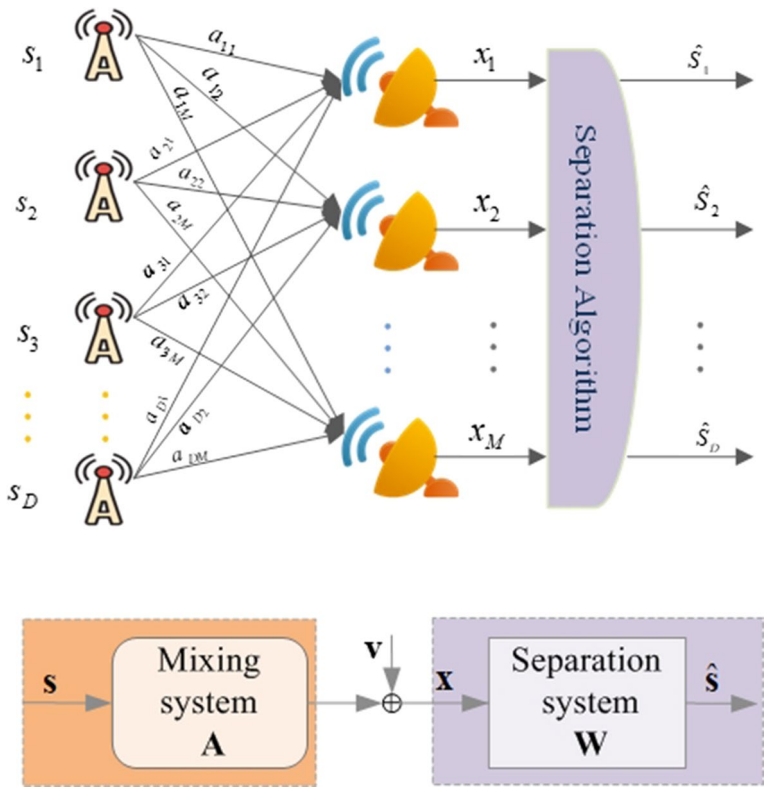


Fig. 1 Overview of signal separation model

proposed separation algorithm. The bias term b is not only beneficial for reducing the static error of the separation system, but also improved the flexibility of the separation system.

In this work, the correlation coefficient $\zeta_{\mathbf{s_i}\hat{\mathbf{s}}_i}$—between s_i and its corresponding estimated signal $\hat{s}_i(i = 1, 2, \ldots, D)$, and the performance index PI [42–44] are employed to measure the separation performance. The definition of $\zeta_{\mathbf{s_i}\hat{\mathbf{s}}_\mathbf{i}}$ and PI is shown in Eqs. (7) and (8), respectively.

$$\zeta_{\mathbf{s_i}\hat{\mathbf{s}}_i} = \frac{\operatorname{cov}(s_i, \hat{s}_i)}{\sqrt{V(s_i)}\sqrt{V(\hat{s}_i)}} = \frac{E\left[(s_i - E(s_i))(\hat{s}_i - E(\hat{s}_i))\right]}{\sqrt{V(s_i)}\sqrt{V(\hat{s}_i)}}, \tag{7}$$

$$\mathrm{PI} = \frac{1}{\mathrm{D}}\sum_{\mathrm{i}=1}^{\mathrm{D}}\left(\sum_{\mathrm{j}=1}^{\mathrm{M}}\frac{|\mathrm{p_{ij}}|}{\max_{\mathrm{k}}|\mathrm{p_{ik}}|} - 1\right) + \frac{1}{\mathrm{M}}\sum_{\mathrm{j}=1}^{\mathrm{M}}\left(\sum_{\mathrm{i}=1}^{\mathrm{D}}\frac{|\mathrm{p_{ij}}|}{\max_{\mathrm{k}}|\mathrm{p_{kj}}|} - 1\right), \tag{8}$$

where $\operatorname{cov}(\cdot)$, $E(\cdot)$ and $V(\cdot)$ represent covariance, mean value and variance, respectively. $0 \leq \zeta_{\mathbf{s_i}\hat{\mathbf{s}}_i} \leq 1$ and the lager $\zeta_{\mathbf{s_i}\hat{\mathbf{s}}_\mathbf{i}}$ is, the better separation performance will be. p_{ij} is the ith row and jth column of matrix $\mathbf{P}$:

$$\mathbf{P} = \mathbf{WA}, \mathbf{P} \in \mathbb{R}^{D\times D}. \tag{9}$$

$\mathrm{PI} \geq 0$, and the lower PI is, the higher the separation accuracy will be, and $\mathrm{PI} < 0.1$ typically indicating the algorithm is performing adequately [44].

3 Separation model

The theoretical framework of blind signal separation can be divided into two parts: objective function and optimization algorithm. The objective function is usually called the cost function. Figure 2 provides the separation methodology's topological structure diagram—expanded around that two core parts objective/cost function and optimization algorithm—of this paper.

As shown in Fig. 2, the topological structure of our separation methodology includes observation signal model, signal separation model and estimated signal. The observation signal model has been introduced in detail in Sect. 2. The signal separation model—the core of this paper—contains cost function and its optimization—detail introduction will

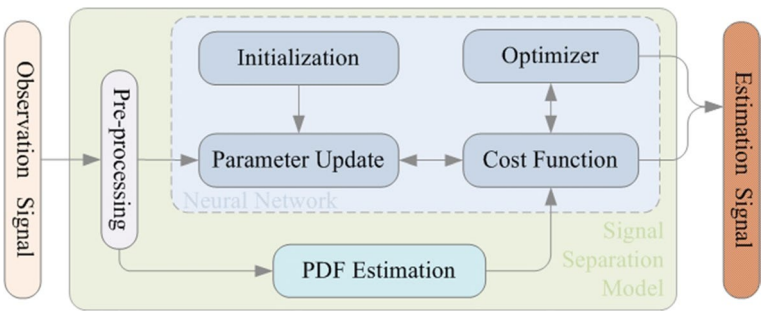


Fig. 2 The topological structure diagram of the separation methodology of this paper

be given in Sects. 3.2 and 3.4, respectively, and we employed the neural network(NN)—detail introduction as shown in Sect. 3.5—to complete this task. The inputs of NN contains preprocessed—as it is introduced in Sect. 3.1—original mixed signal and its corresponding probability destiny function (PDF) estimation—one term of the cost function—as it is given in Sect. 3.3. The estimated signal obtained will be evaluated by ς and PI, defined by Eqs. (7) and (8), respectively.

3.1 Preprocessing

Preprocessing on received mixed observation signal includes de-averaging and whiten, and the corresponding mathematical explanation is shown in Eqs. (10) and (11).

$$\mathbf{x} \leftarrow \mathbf{x} - E(\mathbf{x}), \tag{10}$$

where $E(\cdot)$ represents taking the mean value. The zero-mean signal form can simplify the separation process.

$$\mathbf{x}' = \mathbf{V}\mathbf{x}, \tag{11}$$

where $\mathbf{V}$ is the whiten matrix:

$$\mathbf{V} = \mathbf{G}^{-\frac{1}{2}}\mathbf{E}^{H}, \tag{12}$$

$$\mathbf{G} = \operatorname{diag}\left(g_1, g_2, \ldots, g_D\right), \tag{13}$$

$$\mathbf{E} = [e_1, e_2, \ldots, e_D], \tag{14}$$

where $\mathbf{G}$ is a diagonal matrix and its diagonal element g_i is eigenvalue of the covariance matrix of $\mathbf{x}$, and e_i is their corresponding eigenvectors, $i = 1, 2, \ldots, D$. H means conjugate transpose operation.

It is worth to mention that the mixed matrix $\mathbf{A}$ has changed into $\mathbf{A}' = \mathbf{V}\mathbf{A}$ after whiten. Therefore, we should take the whitening matrix into consideration when calculate PI.

3.2 Cost function

The cost function of our separation method is built based on maximum likelihood estimation (MLE). First, the maximum likelihood (ML) estimation derivative process for blind signal separation will be illustrated. Then, the cost function of our separation method will be provided based on ML criterion. Additionally, the probability density function of original signal will be estimated through kernel density function estimation method.

3.2.1 Maximum likelihood criterion

After preprocessing the observation signal can be expressed as $\mathbf{x} = \mathbf{V}\mathbf{A}\mathbf{s}$, and its joint probability density function is shown in Eq. (15).

$$p_{\mathbf{x}}(\mathbf{x}; \mathbf{W}) = \left|\det(\mathbf{W})\right| p_{\mathbf{s}}(\mathbf{W}\mathbf{x}), \tag{15}$$

where $\mathbf{W}$ is the unmixed/separation matrix, and $p_{\mathbf{s}}$ is the joint probability density function of the source components. We can assume that the source signal is statistically independent. Using $\mathbf{w}_i$ to represent the ith column vector of $\mathbf{W}$, then:

$$\begin{aligned} p_{\mathbf{x}}(\mathbf{x};\mathbf{W}) &= \left|\det(\mathbf{W})\right| \prod_{i=1}^{D} p_{s_i}(\mathbf{w}_i\mathbf{x}) \\ &= \left|\det(\mathbf{W})\right| \prod_{i=1}^{D} p_{s_i}(\hat{\mathbf{s}}_i). \end{aligned} \tag{16}$$

Using $\hat{s}_i[n]\,(n = 1, 2, \ldots, N)$ to express the sample points of estimated signal $\hat{\mathbf{s}}_i$, N is the total sampling points number. Then, we can implement the likelihood function operation by Eq. (16) [44, 45]:

$$p_{\mathbf{x}}(\mathbf{x};\mathbf{W}) = \left|\det(\mathbf{W})\right| \prod_{i=1}^{D}\prod_{n=1}^{N} p_{s_i}(\hat{s}_i[n]). \tag{17}$$

Performing logarithmic operation and dividing the number of samples on both sides of Eq. (17):

$$\begin{aligned} L(\mathbf{W}) &= \frac{1}{N}[\log(\mathrm{p}_{\mathbf{x}}(\mathbf{x};\mathbf{W}))] \\ &= D\log\left|\det(\mathbf{W})\right| + \frac{1}{N}\left[\sum_{i=1}^{D}\sum_{n=1}^{N}\log\left(p_{s_i}(\hat{s}_i[n])\right)\right]. \end{aligned} \tag{18}$$

According to the maximum likelihood estimation criterion, we can obtain the optimal solution by maximizing $L(\mathbf{W})$. Therefore, $-L(\mathbf{W})$ function is employed as part components of our cost function.

3.2.2 MLE-based cost function

The MLE-based cost function of our method is composed by log-likelihood function and a bias term (b) [41]. However, the bias term (b) in our method is much different from that of reference [41]. We modified the second part of log-likelihood function and use the kernel density estimation method to obtain the joint probability density function of the original signal. Additionally, we add a constant in the cost function in case there appear illegal values. Then, the cost function used in this paper is shown in Eq. (19).

$$\begin{aligned} \operatorname{argmin} J(\mathbf{W}, b) = &-D\log\left|\det(\mathbf{W})\right| \\ &- \frac{1}{N}\left[\sum_{i=1}^{D}\sum_{n=1}^{N}\log\left(p_{s_i}(\hat{s}_i[n]) + c\right)\right] \\ &+ \frac{1}{D}\frac{\lambda}{2}\left(\|\mathbf{W}\|_2^2 + \|b\|_2^2\right), \end{aligned} \tag{19}$$

where 'argmin' means taking the minimum value. The first two parts are derived from MLE; we add a constant '$\mathbf{c}$' in the second part, which is used to avoid illegal values in the original signal joint probability density function estimation. The third part of cost function is L_2 regularization, which plays a key role in preventing over-fitting during

optimization, and a comparison between L_2 and L_1 regularization—together with the regularization parameter (λ)—will be given in Sect. 4.2. By minimizing the cost function as Eq. (19), the optimal unmixing matrix $\mathbf{W}$ and bias term b can be obtained.

3.3 Probability density function estimation

The probability density function (PDF) of original/source signal is an necessary part of MLE-based cost function as shown in Eq. (19). Liu et al. [41] employed the simple PDF estimation method, which adopted three approximate functions to represent the probability density function of super-Gaussian signal, sub-Gaussian signal and Gaussian signal, respectively, and then selected one approximate function as the PDF estimation of the source signal based on the its distribution. In practical application, super-Gaussian signal or sub-Gaussian signal has a relatively wide range; therefore, using an approximate function to describe a class of signals (super-Gaussian signal/sub-Gaussian signal) will inevitably introduce absolute error.

Histogram method is a traditional PDF estimation algorithm. Comparing with histogram method, kernel density estimation can provide a smoother PDF curve [46, 47]. Therefore, in order to minimize the influence introduced by PDF estimation on separation accuracy, we employ the kernel density estimation (KDE) [46–48] to estimate the probability density function of the source signal.

Let the series $\{x_1, x_2, \ldots, x_N\}$ be an independent and identically distributed sample of observation signal with an unknown probability distribution function $p(x)$. KDE $\hat{p}(x)$ of original $p(x)$ assigns each nth sample data point x_n a function $K(x_n, t)$ called a kernel function in the following way [46, 47]:

$$\hat{p}(x) = \frac{1}{N}\sum_{n=1}^{N} K(x_n, t), \tag{20}$$

where $0 < K(x, t) < \infty$, and

$$\int_{-\infty}^{\infty} K(x, t)\mathrm{d}t = 1. \tag{21}$$

Equation (21) ensures the required normalization of KDE $\hat{p}(x)$:

$$\int_{-\infty}^{\infty} \hat{p}(x)\mathrm{d}t = \frac{1}{N}\sum_{n=1}^{N}\int_{-\infty}^{\infty} K(x_n, t)\mathrm{d}t = 1. \tag{22}$$

That is to say, KDE transforms the location of x_n into a self-centered interval, symmetrically or asymmetrical. Many kernel functions both symmetric and asymmetric have been published as shown in "Appendix." However, in practical applications, the symmetric kernel function is more widely used than asymmetry. Symmetry property allows to write the kernel function in a form used most frequently [46]:

$$K(x, t) = \frac{1}{h}K\left(\frac{x - t}{h}\right), \tag{23}$$

where h is the smoothing parameter who governs the amount of smoothing applied to the sample. Too small value of h may result the estimator to show insignificant details, while too large value of h causes over smoothing of the information contained in the sample, which, in consequence, may mask some of important characteristics, e.g., multimodality [46], of $p(x)$. Therefore, a certain compromise is necessary in actual application.

Multivariate extensions of the kernel approach generally rely on the product kernel [49]; taking bivariate data $(x_n, y_n), n = 1, 2, \ldots, N$, for example, the bivariate kernel estimator can be expressed as:

$$\begin{aligned}\hat{p}(x,y) &= \frac{1}{N}\sum_{n=1}^{N} K(x_n,t)K(y_n,t) \\ &= \frac{1}{Nh_xh_y}\sum_{n=1}^{N} K\left(\frac{x_n-x}{h_x}\right)K\left(\frac{y_n-y}{h_y}\right),\end{aligned} \tag{24}$$

where $(x_n, y_n), n = 1, 2, \ldots, N$ is a sample, and h_x and h_y are smoothing parameters. Based on the Euclidean distance between an arbitrary point (x, y) and sample point $(x_n, y_n), n = 1, 2, \ldots, N$, the bivariate kernel estimator shown in Eq. (24) can be changed into:

$$\hat{p}(x,y) = \frac{1}{Nh_xh_y}\sum_{n=1}^{N} K\left(\sqrt{\left(\frac{x_n-x}{h_x}\right)^2+\left(\frac{y_n-y}{h_y}\right)^2}\right), \tag{25}$$

where $K(\cdot)$ is the kernel function and "Appendix" gives several kernel function including symmetric kernel functions and asymmetry ones. The effective of KDE will be exhibited in Sect. 4 through signal separation results.

3.4 Optimization algorithm

The optimization of traditional blind signal separation method includes negative gradient descent algorithm [50], Newton algorithm [51], fixed point algorithm [2] and so on. Recently, some research has been done on adaptive gradient optimization algorithms and its variant for training deep neural networks, such as stochastic gradient descent (SGD) [52–54], Adagrad [55, 56], RMSprop [41, 56–58] and Adam [59]. The optimization process of those algorithms can be considered as the problem of minimum the cost function (or objective function) in the form of summation:

$$J(w) = \frac{1}{N}\sum_{n=1}^{N} J_n(w), \tag{26}$$

where w is the estimated parameter by minimizing $J(w)$. Each sum and function $J_n(w)$ are typically associated with the n-th observation in the data set. One thing worth mentioning is that the parameter b to be estimated in Eq. (19) is omitted by Eq. (26), but it will participate in the actual optimization. In the following, we will briefly introduce each optimization algorithm.

3.4.1 Stochastic gradient descent

SGD is an iterative method for optimizing an objective function with smoothness properties (e.g., differentiable or sub-differentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). In SGD algorithm, the true gradient of objective function is approximated by a gradient at a single example:

$$w_{t+1} = w_t - \eta \nabla J_n(w), \tag{27}$$

where η is a step size or called learning rate in machine learning. Sutskever et al. [54] proposed that a SGD method with momentum remembers the update Δ at each iteration and determines the next update as a linear combination of the gradient and the previous update:

$$\Delta w_{t+1} = \rho \Delta w_t - \eta \nabla J_n(w), \tag{28}$$

$$w_{t+1} = w_t + \Delta w_{t+1}, \tag{29}$$

where ρ is an exponential decay factor between 0 and 1, which determines the relative contribution of the current gradient and earlier gradients to the weight change. Combining Eqs. (28) and (29), we can get the final update formula of SGD with momentum:

$$\Delta w_{t+1} = w_t - \eta \nabla J_n(w) + \rho \Delta w_t. \tag{30}$$

SGD with momentum (named SGDM) tends to keep convergence in the same direction, preventing oscillations.

3.4.2 Adagrad

Duch et al. [55] proposed a modified stochastic gradient descent algorithm with per-parameter learning rate, named adaptive gradient algorithm (Adagrad), which improved convergence performance of SGD in settings where data are sparse and sparse parameters are more informative. The update formula of Adagrad [55] is:

$$w_{t+1} = w_t - \eta \mathrm{diag}(\mathrm{G})^{-\frac{1}{2}} \odot \mathrm{g}, \tag{31}$$

or written in the form of per-parameter updates:

$$w_j = w_j - \frac{\eta}{\sqrt{G_{j,j}}} g_j, \tag{32}$$

where $\odot$ means the element-wise product. $\{G_{j,j}\}$ is a vector which is the diagonal of the outer product matrix G:

$$G = \sum_{\tau=1}^{t} g_\tau g_\tau^T, \quad g_\tau = \nabla J_n(w), \tag{33}$$

where g_τ is the gradient at iteration τ, and the diagonal of G is given by

$$G_{j,j} = \sum_{\tau=1}^{t} g_{\tau,j}^2. \tag{34}$$

As in reference [55, 56], Adagrad was designed for convex problems; however, it has been successfully applied to non-convex optimization [60].

3.4.3 RMSprop

Root mean square propagation (RMSprop) is also a method in which the learning rate is adapted for each of the parameters. The idea is to divide the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight [58]. So, first the running average is calculated in terms of means square:

$$r(w,t) = \rho r(w,t-1) + (1-\rho)\nabla(J_n(w))^2, \tag{35}$$

where ρ is the forgetting factor, and $r(w,t)$ is the gradient accelerating variable. Then, the parameters are updated as:

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{r(w,t)}}\nabla J_n(w). \tag{36}$$

RMSprop has shown good adaptation of learning rate in different applications. RMSprop can be seen as a generalization of resilient back-propagation (BP) and is capable to work with mini-batches as well opposed to only full-batches [58]. Reference [41] improved RMSprop by introducing in the estimation of the first-order moment of the gradient ($g(w,t)$), and the original $r(w,t)$ is modified to the central second-order moment through the operation $(r(w,t) - \big(g(w,t)\big)^2)$:

$$g(w,t) = \rho g(w,t-1) + (1-\rho)\nabla J_n(w), \tag{37}$$

$$v(w,t) = \beta v(w,t-1) + \frac{\eta}{\sqrt{r(w,t) - \big(g(w,t)\big)^2 + \epsilon}}\nabla J_n(w), \tag{38}$$

$$w_{t+1} = w_t - v(w,t), \tag{39}$$

where ρ is the decay rate of the exponential moving average between 0 and 1, β is the momentum term, and ϵ is a small scalar (e.g., 10^{-8}), which avoids divide-by-zero errors in the update process.

The introduction of first-order and second-order moment in RMSprop (named RMSpropM) stabilized the exponentially weighted root mean square, and this operation flattens the steep gradient in the parameter space [41]. In practice, the algorithm finds a smoother descent direction in the parameter space, increasing the training speed.

3.4.4 Adam

Adaptive moment estimation (Adam) is an update method of RMSprop optimizer. In this optimization algorithm, running averages of both the gradients and the second

moments of the gradients are used. Given parameters $w^{(t)}$ and a loss function $J^{(t)}$, where t indexes the current training iteration, Adam's parameter update is given by [59]:

$$m(w, t+1) = \beta_1 m(w, t) + (1-\beta_1)\nabla J_n(w), \tag{40}$$

$$v(w, t+1) = \beta_2 v(w, t) + (1-\beta_2)\nabla (J_n(w))^2, \tag{41}$$

$$\hat{m}_w = \frac{m(w, t+1)}{1-\beta_1^{t+1}}, \tag{42}$$

$$\hat{v}_w = \frac{v(w, t+1)}{1-\beta_2^{t+1}}, \tag{43}$$

$$w_{t+1} = w_t - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \epsilon}, \tag{44}$$

where ϵ is a small scalar (e.g., 10^{-8}). $m(w, t)$ and $v(w, t)$ are the first moments of gradients and second moments of gradients, respectively, and β_1 and β_2 are their corresponding forgetting factor between 0 and 1 (e.g., $\beta_1 = 0.9$, $\beta_2 = 0.999$).

The optimization algorithm of neural network includes SGD, Adagrad, RMSprop and Adam, but not limited to them, e.g., Adadelta [?], the detailed introduction is omitted here. We will show their performance in optimizing the signal separation cost function Eq. (19) in Sect. 4.

3.5 Neural network

A neural network (NN), in the case of artificial neurons called artificial neural network (ANN) or simulated neural network (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation. In the artificial intelligence field, artificial neural networks have been applied successfully to speech recognition [61], image analysis [62], pattern recognition [63], data classification [64], through a learning process. The input of the NN is the feature vector corresponding to observation signal.

As shown in Fig. 3, the NN architecture has four layers, input layer, dense layer, lambda layer and output layer, and the relation between the neural network and separation process is shown in Table. 1. The input layer corresponds to the observed signal $\mathbf{x}$ and bias term b of the separation system. The neuron number of input layer is $(M+1)$, where M is the observation signal number, and the other neuron is used to input the initialization of bias term b—as analyzed in Sect. 2. The dense layer is used to optimize the separation matrix $\mathbf{W}$ and the bias term b, and the dense layer has D neurons, where D is number of original signal. Lambda layer is the self-definition layer with two neurons, one neuron is for the regularization of $\mathbf{W}$ and b, and the other neurons stand for the second term of cost function as shown in Eq. (19). The output layer with one neuron is used to provide the sum value of cost function.

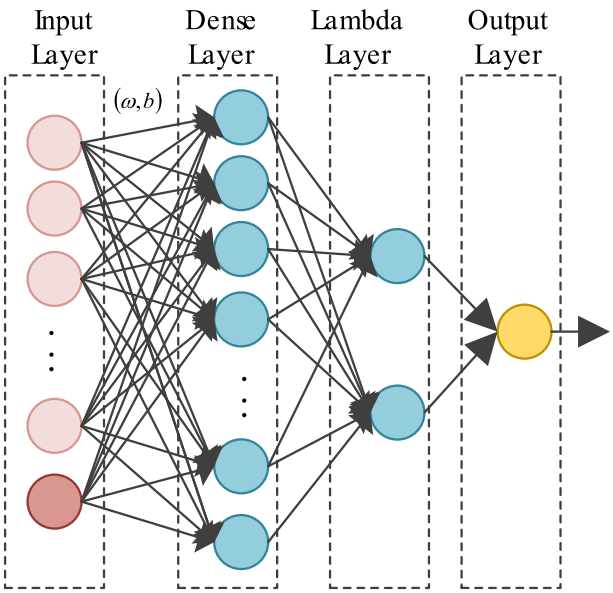


Fig. 3 Neural network structure

Table 1 The relation between the neural network and separation process

NN layer	Dimension	Information
Input layer	$M+1$	Observation **x** and initial bias term b
Dense layer	D	Optimization Parameters: **W** and b
Lambda layer	2	Cost function as Eq. (19)
Output layer	1	Loss value

4 Numerical simulation and analysis

This section presents numerical simulation results of our separation method for time–frequency overlapped—the definition of frequency overlapped level (FOL) as shown in Eq. (45)—digital communication signal together with the corresponding analysis and comparison.

$$\begin{cases} \psi_{s_i} = \frac{\text{Overlapped bandwidth of } S_i}{\text{Bandwidth of } S_i} \\ \psi = \max\left(\psi_{s_i}\right), \end{cases} \tag{45}$$

where s_i, $i = 1, 2, \ldots, D$ are the original signal and D is the original signal number. Without loss of generality, here we employ two binary phase-shift keying (BPSK) signal—regarded as s_1 and s_2—and one quadrature phase-shift keying (QPSK) signal—regarded as s_3—as the original signal, and their corresponding carrier frequency f_c is set to 12 MHz, 14 MHz and 16 MHz, respectively, and their corresponding bit transmission rate r_b is 2 MHz, 2 MHz and 4 MHz, respectively, and the bits number of original signal is equal to 1000. Then, we can obtain the FOL of each original signal by Eq. (45), as $\psi_{s_1} = 50\%$, $\psi_{s_2} = 100\%$, $\psi_{s_3} = 50\%$ and $\psi = 100\%$, and we regard this experiment as case 1, as shown in Table 2. The mixed matrix set to $\mathbf{A} = [1, 0.5, 0.5; 0.5, 1, 0.5; 0.5, 0.5, 1]$ and the sample frequency f_s takes 100 Mhz. Signal-to-noise (SNR) ratio is defined by the

Table 2 The original/source signal setting

Signal type	Case 1			Case 2			Case 3		
	r_b (MHz)	f_c (MHz)	ψ	r_b (MHz)	f_c (MHz)	ψ	r_b (MHz)	f_c (MHz)	ψ
BPSK	2	12	50	2	12	100	1	12	100
BPSK	2	14	100	4	14	100	2	13	100
QPSK	4	16	50	4	16	100	2	14	100

logarithmic form of the ratio of the observed signal power to the noise power and multiplied by 100 Mhz. In the following, we will exhibit the separation performance of the our method from different aspects.

4.1 Comparison between different optimizers

In this experiment, we will inspect the convergence speed of different optimizers and the corresponding separation efficacy in the form of correlation coefficient (ζ) and performance index (PI), and the simulation condition as the case 1 is shown in Table 2 with SNR $=$ 10dB and $\lambda = 0.015$ using L_2 regularization.

Table 3 shows the optimizer candidates participating in the comparison and their empirical parameter setting in the first two rows. Figure 4 gives the convergence speed of each optimizer, and we can see all the optimizers can reach convergence state with

Table 3 The separation results of different optimizers in the form of ζ and PI with SNR=10dB

Optimizer	SGD	SGDM	Adagrad	Adam	Adadelta	RMSprop	RMSpropM
Parameter setting	$\eta = 2.0$	$\eta = 0.8$ $\rho = 0.89$	$\eta = 5.0$ $\epsilon = 10^{-8}$	$\eta = 1.9$ $\beta_1 = 0.845$ $\beta_2 = 0.89$ $\epsilon = 10^{-8}$	$\eta = 152$ $\rho = 0.1$ $\epsilon = 10^{-6}$	$\eta = 0.24$ $\rho = 0.24$ $\epsilon = 10^{-8}$	$\eta = 0.2$ $\rho = 0.2$ $\beta = 0.3$ $\epsilon = 10^{-8}$
Correlation coefficient (ζ)	0.7112 0.7119 0.7264	0.7522 0.7270 0.7696	0.7398 0.7335 0.7564	0.8865 0.8868 0.8852	0.8865 0.8868 0.8853	0.8850 0.8854 0.8853	0.8864 0.8868 0.8852
Performance index (PI)	0.9680	0.8029	0.7667	0.0239	0.0239	0.0302	0.0249

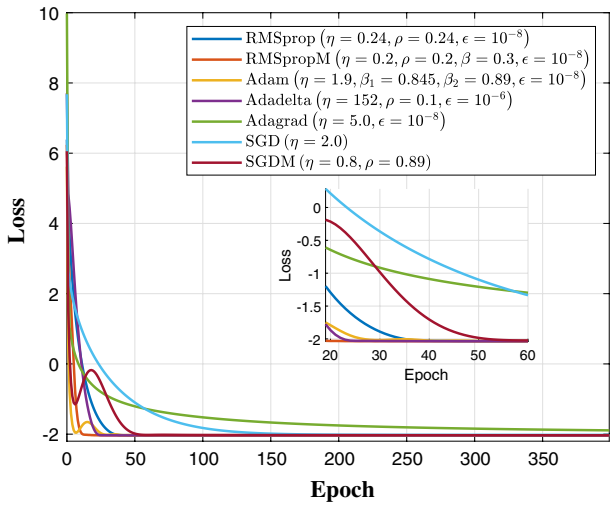


Fig. 4 The convergence speed of different optimizers with SNR $=$ 10 dB

epoch less than 50. To be precise, the convergence of RMSprop, RMSpropM, Adam and Adadelta optimizer can be completed with epoch less than 40, and their convergence value—smaller than − 2.2—is smaller than the other three optimizers—SGD, SGDM and Adagrad.

The separation results of different optimizers in the form of ζ and PI, while SNR=10dB with 200 times Monte Carlo test as shown in Table 3. We can see the separation accuracy of RMSprop, RMSpropM, Adam and Adadelta optimizer—with PI < 0.08 and $\zeta > 0.85$ —is much better than that of SGD, SGDM and Adagrad—with PI > 0.8 and $\zeta < 0.75$. As PI < 0.1 typically indicating that the algorithm is performing adequately [44], we can say RMSprop, RMSpropM, Adam and Adadelta optimizer are more suitable for our application scenarios—time–frequency overlapped digital communication signal separation—than the other three optimizers—SGD, SGDM and Adagrad. Therefore, those four optimization algorithms will be employed in the following simulation test.

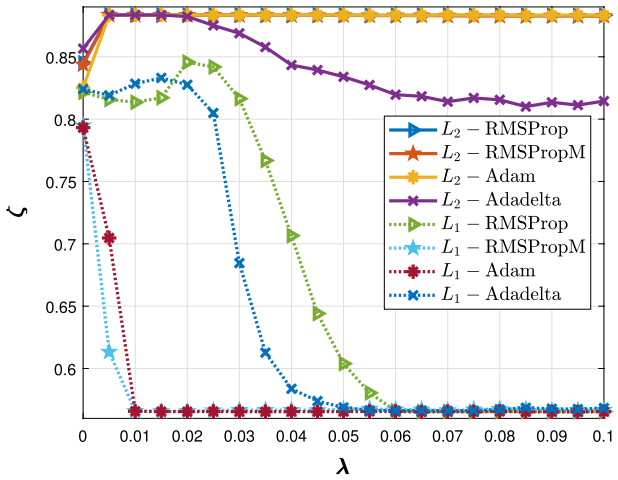
4.2 Comparative test for regularization term of cost function

A comparative test for regularization term of cost function will be presented in this subsection with regularization term parameter λ. The simulation conditions and optimizers—RMSprop, RMSpropM, Adam and Adadelta—parameters keep the same as that of the experiment in Sect. 4.1, except for the regularization term—varying from L_1 to L_2—and its parameter λ—changing from 0 to 0.1, the simulation results—average value of 200 times Monte Carlo test—as shown in Fig. 5. There has one thing worth mentioning that the correlation coefficient (ζ) is the average value of each original signal: $\zeta = \frac{1}{D}\sum_{i=1}^{D}\zeta_{s_i\hat{s}_i}$, where D is the number of original signal, and s_i and $\hat{s}_i$ are the ith ($i =, 1, 2, \ldots, D$) original signal and its corresponding estimation, respectively.

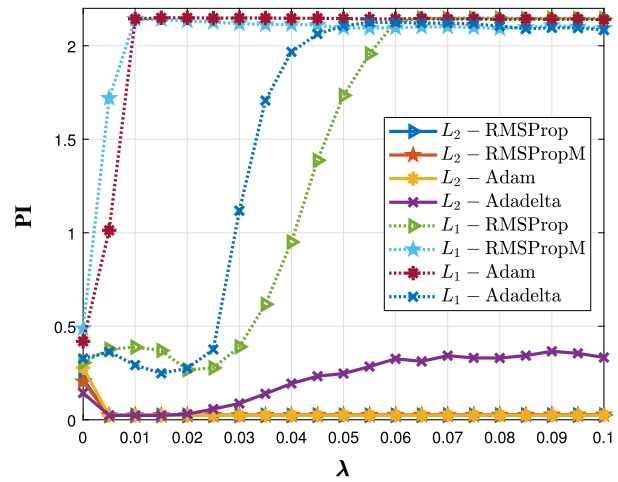
From Fig. 5, we can see when L_2 regularization is employed in the cost function, the separation accuracy gradually improves with λ increasing from 0 to 0.01, and then, it will reach a stable level ($\zeta \approx 0.85$ and PI ≈ 0.025), while $\lambda \in [0.01, 0.1]$, expect for the Adadelta whose separation accuracy gradually decreased for λ changing from 0.01 to 0.1. On the contrary, when L_1 regularization is selected, the separation accuracy of our method will decrease rapidly—with slight fluctuations for RMSprop optimizer—while λ increases from 0.01 to 0.1, and it will reach a stable level when $\lambda \in [0.01, 0.1]$ for RMSpropM and Adam optimizer, and $\lambda \in [0.06, 0.1]$ for RMSprop and Adadelta optimizer. Additionally, the best separation that can be achieved using L_1 regularization is $\zeta \approx 0.78$ and PI ≈ 0.5, which is much lower than that of L_2 regularization method. Therefore, L_2 regularization is the best choice for our cost function, and according to the above analysis, we set λ to 0.015.

4.3 Separation performance against noise and comparison with typical methods

The purpose of this experiment is to figure out the performance of our separation method against noise and analysis its computational complexity. In addition, a comparison with the typical separation methods—FastICA and JADE—will be carried out. The simulation conditions still keep the same as the first two experiments—as case 1



(a) ζ.



(b) PI.

Fig. 5 The separation results in the form of ζ and PI for L_1 and L_2 regularization with SNR = 10 dB and regularization parameter λ changing from 0 to 0.1

described in Table 2, except for SNR. Based on the experimental analysis in Sect. 4.2, L_2 regularization term is set to 0.015. Figure 6 shows the separation performance of our methods (including RMSprop, RMSpropM, Adam and Adadelta four optimizers) changing trend with SNR—varying from 5dB to 25dB—and compared with FastICA and JADE in the form of ζ and PI.

As shown in Fig. 6, the separation accuracy gradually improves with SNR increasing for both of our method and FastICA/JADE. When SNR $\geq$ 14 dB, the improvement speed of the separation results becomes slower compared with that of SNR $\leq$ 14 dB, especially for the performance index PI. To be precise, when SNR $\geq$ 14 dB, the average value of the each source signals' correlation coefficient ζ will be higher than 0.95 and PI will be lower than 0.01, no matter RMSprop, RMSpropM, Adam or Adadelta optimizer is used.

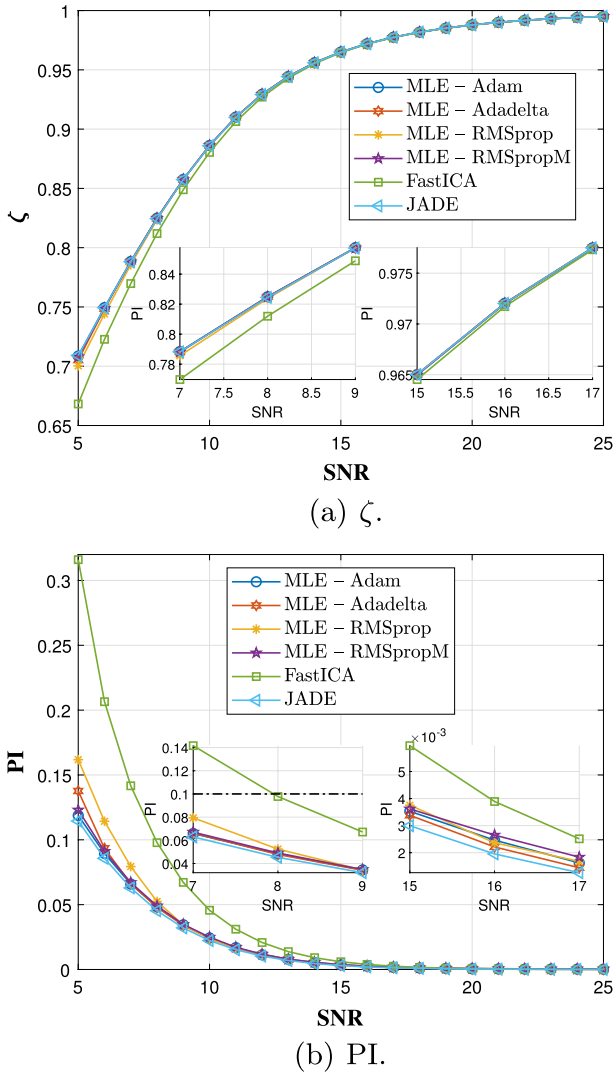


Fig. 6 The separation performance changing trend with SNR and compared with FastICA and JADE in the form of ζ and PI

What's more, the performance of our method outperforms classical algorithms—FastICA, especially for SNR $\leq$ 14 dB. To be exactly, as shown in Fig. 6a, the ζ obtained by our method is bigger than that of FastICA and keeps the same level with JADE method. For SNR $\geq$ 14 dB, the separation performance of all methods reaches a similar stable high accuracy level with $\zeta > 0.97$. Meanwhile, the elevation in the form of performance index PIis lower than 0.1 for SNR no less than 8 dB for all method—as shown in Fig. 6b, we can see the PI achieved by our method is much lower than that of FastICA and keeps the same level with JADE method, while SNR $\leq$ 14 dB, and they will converge to similar stable low level, while SNR $\geq$ 15 dB, to be precise, PI on more than 0.01. In other words, in the low SNR environment (SNR $\leq$ 14 dB), the separation performance of our method is much better than that of the classical FastICA method. As the SNR increases, our separation method can converge to the same level as the classical method.

Table 4 The comparison between proposed method and typical ones in the form of ζ , PI and running time with SNR = 10 dB and bits number equal to 1000

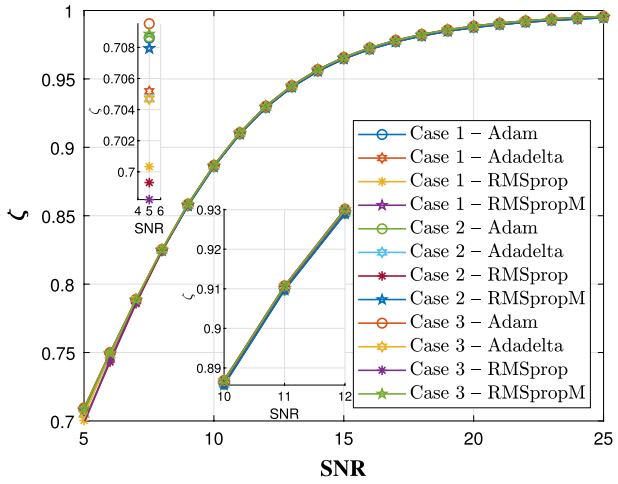
	$\zeta_{(s_1,s_2,s_3)}$	PI	Running time
MLE-Adam	0.8862,0.8865,0.8850	0.0252	0.4032 s
MLE-RMSprop	0.8863,0.8869,0.8852	0.0242	0.5167 s
MLE-RMSpropM	0.8862,0.8868,0.8851	0.0244	0.2923 s
MLE-Adadelta	0.8863,0.8869,0.8853	0.0240	0.4418 s
FastICA	0.8797,0.8824,0.8795	0.0463	11.2785 ms
JADE	0.8864,0.8867,0.8852	0.0226	7.3812 ms

Additionally, the computational complexity comparison between our proposed method and typical ones is shown in Table 4 in the form of running time, and the simulation conditions keep the same as the performance comparison test except for SNR=10dB. We can see the separation results of the proposed method are similar to that of JADE, but it is much better than that of FastICA—the PI value is about twice of the proposed method and JADE algorithm. The running time of the proposed method is 0.3–0.5 s; however, the typical separation methods only need about 10 ms. Our method improves the signal separation result, but it costs longer time. To be specify, $2 \times (M+1) \times D \times N$ flops—one multiplication and one addition named one flop—computation is needed for FastICA in one iteration loop [65], and $(M+15) \times D \times N$ flops for the proposed method; therefore, optimization methods with low computational complexity will be studied in future work.

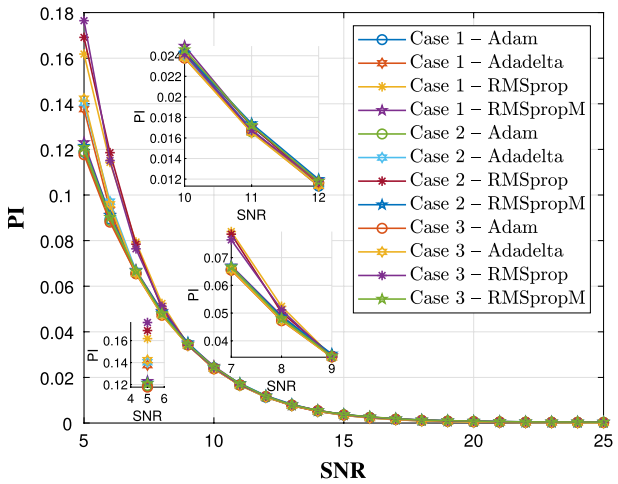
4.4 Comparative test for frequency overlapped level

This experiment is used to evaluate the effect of original signal's FOL on the separation results through three of experiments as three cases shown in Table 2 with mixing matrix $\mathbf{A} = [1, 0.5, 0.5; 0.5, 1, 0.5; 0.5, 0.5, 1]$ and $f_s = 100\,\text{MHz}$. The other simulation conditions setting as: adopted L_2 regularization term with $\lambda = 0.015$, employed four effective optimizers (including RMSprop, RMSpropM, Adam and Adadelta), implement 200 times Monte Carlo tests and set 400 epochs. Figure 7 shows the separation performance changing trend with SNR in varied FOL environment evaluated in the form of ζ and PI.

From the simulation conditions shown in Table 2, we can see the difference between case 1 and case 2 is the FOL of original signal, to be exactly, and the FOL of each signal is $\psi_{s_1} = 50\%$, $\psi_{s_2} = 100\%$ and $\psi_{s_3} = 50\%$ in case 1, respectively, and in Case 2 ψ_{s_1} and ψ_{s_3} are all increase to 100% by changing their bit transmission rate (r_b). However, the separation results of those two case are almost the same as shown in Fig. 7, especially for SNR $\geq 8\,\text{dB}$ situation. By comparing the simulation conditions of case 3 with that of case 2, we can see the center frequency interval between original signal of case 3 is half of case 2; in other words, although $\psi_{s_i}(i = 1, 2, 3)$—as defined in Eq. (45)—are the same in those two case, the signal dense in frequency domain of Case 3 is twice of Case 2. Therefore, to a certain extent, the frequency-domain overlap complexity of Case 3 is higher than that of Case 2. The separation results of those two cases keep a high degree



(a) ζ.



(b) PI.

Fig. 7 The separation performance changing trend with SNR in varied FOL environment in the form of ζ and PI

of consistency, especially while SNR $\geq$ 8 dB. Through the comparative analysis of case 1 with case 2 and case 2 with case 3, we can draw a conclusion that our signal separation method is not sensitive to FOL, no matter the FOL reaches 100% or the frequency-domain complexity is high, and our method still can obtain high separation accuracy.

4.4.1 Section summary

Firstly, we studied optimizer (RMSprop, RMSpropM, Adam and Adadelta), regularization term (L_2) and its parameter ($\lambda = 0.015$) that match our application scenarios—time–frequency overlapped digital communication signal separation. Then, the performance of our method was given by comparing with typical method, and the simulation results show our method is much better than that of FastICA, especially for SNR $\leq$ 14 dB. After that, through three groups comparative experiment, we illustrated our method not sensitive to the FOL and the frequency-domain complexity degree, even

for $\psi = 100\%$ and high-frequency complexity condition, our method still can provide satisfied result.

5 Discussion

5.1 Separation method

For the over-determined/determined blind signal separation problem, ICA—in particularly, FastICA [10]—is the most widely used and most popular separation method. ICA and its variants only require that the source signals are independent to each other and have been successfully applied in all kinds of signal separation, like speech signal [66], biomedical signal—e.g., electroencephalographic (EEG) and magnetoencephalographic (MEG) [67], and so on. What's more, ICA also can be used to undetermined signal separation problem under certain condition that the under-determined observation matrix can be transformed into an observation matrix whose rank is no less than source signal number [18]. For signals with sparsity, sparse component analysis (SCA) is another popular method, and it has been successfully separate image mixture separation [23–25], speech signal [26, 27], biological signal [28, 29] and so on. Additionally, SCA can handle under-determined signal separation problems apart from over-determined and determined situation, in under-determined music and speech signal separation in [26]. Except for SCA, NMF is another main under-determined signal separation method with successful application in various signal separations [32–37].

One common of those three popular and successful separation methods is that they are all use traditional signal separation methods. Liu et al. [41] introduced a separation method using neural network (NN) and applied machine learning mechanisms and optimization methods to signal separation domain. As an important term of observation/cost function, the probability density function (PDF) term was expressed by a fixed function based on the type of the source signal—super-Gaussian distribution or sub-Gaussian distribution or Gaussian distribution, which can hardly handle complex time–frequency overlapped digital communication signal separation. In this paper, we employed the kernel density estimation method to estimate signal PDF instead of one simply fixed expression, and we achieved satisfactory separation results, to be exactly; it can provide similar separation accuracy as the most famous traditional signal separation methods—FastICA and JADE—for time–frequency overlapped digital communication signal separation.

5.2 Bias term and optimizer

A regularization term was added to observation/cost/loss function, and simulation test shows L_2 regularization can improve signal separation accuracy; however, the participation L_1 regularization will bring in negative effect. Additionally, the regularization term parameter λ is set to 0.015 based on simulation tests. Meanwhile, we figured out four optimizers—RMSprop [58], RMSpropM [41], Adam [59] and Adadelta [?]—that are more friendly to our application background.

5.3 Future work

Future work can be carried out from the perspective of both objective/loss/cost function and optimization of BSS and improve its performance from the perspective of neural networks (NNs), which is a new concept in BSS domain [41]. We can combine conventional separation algorithms' estimation criterion and the advantages of NN or other excellent machine learning framework to modify—even derive novel—objective/loss/cost function and improve the convergence, computational complexity as well as separation accuracy of BSS algorithm.

6 Conclusion

In this paper, we introduced a maximum likelihood estimation (MLE)-based blind time–frequency overlapped digital communication signal separation method using neural network, in which L_2 regularization is employed as one term of observation function, and kernel density estimation is selected to estimate the PDF. Through theoretical introduction and experimental analysis, we figured out the optimizer of neural network suitable for our application background, to be exactly, RMSprop, RMSpropM, Adam and Adadelta, with $\zeta > 0.82$ and PI < 0.01—typically indicating the algorithm is performing adequately [44]—while SNR ≤ 8 dB, and ζ will increase to 0.97 and PI decreases to 0.01 for SNR ≥ 15 dB.

The comparison between our method and typical separation method (FastICA/JADE) indicated our method performance better than FastICA in low SNR environment, and it can achieve the same stable high precision level as FastICA/JADE, while SNR > 15 dB with $\zeta > 0.96$ and PI < 0.004. Comparison tests for different FOL cases and frequency complexity cases demonstrate our method not sensitive to the FOL and the frequency-domain complexity degree, even for $\psi = 100\%$ and high-frequency complexity condition, our method still can provide satisfied result, to be precise, $\zeta > 0.90$ and PI < 0.02 for SNR ≈ 10 dB.

Appendix: Kernel functions

See Table 5.

Table 5 Examples of symmetrical and asymmetrical kernel functions [46]

Symmetrical Kernel functions		Asymmetrical Kernel functions	
Epanechnikov	$K(t) = \begin{cases} \frac{3}{4\sqrt{5}}\left(1 - \frac{1}{5}t\right)^2, & \lvert t\rvert < \sqrt{5} \\ 0, & \lvert t\rvert \geq \sqrt{5}. \end{cases}$	Gamma 1	$K(x, a; t) = \frac{t^{x/a} e^{-t/a}}{a^{x/a+1}\Gamma(x/a+1)}$.
Biweight	$K(t) = \begin{cases} \frac{15}{16}\left(1 - t^2\right)^2, & \lvert t\rvert < 1 \\ 0, & \lvert t\rvert \geq 1. \end{cases}$	Gamma 2	$K(\rho_a(x), a; t) = \frac{t^{\rho_a(x)-1} e^{-t/a}}{a^{\rho_a(x)}\Gamma(\rho_a(x))}$, $\rho_a(x) = \begin{cases} x/a, & x \geq 2a \\ \frac{1}{4}(x/a)^2 + 1, & x \in [0, 2a). \end{cases}$
Gaussian	$K(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$	Reciprocal inverse Gaussian	$K(x, a; t) = \frac{1}{\sqrt{2\pi a t}} e^{-\frac{x-a}{2a}\left(\frac{t}{x-a} - 2 + \frac{x-a}{t}\right)}$.
Rectangular	$K(t) = \begin{cases} \frac{1}{2}, & \lvert t\rvert < 1 \\ 0, & \lvert t\rvert \geq 1. \end{cases}$	Lognormal	$K(x, a; t) = \frac{1}{\sqrt{8\pi \ln(1+a)t}} e^{-\frac{(\ln t - \ln x)^2}{8\ln(1+a)}}$.

Abbreviations

BSS	Blind signal separation
NN	Neural network
MLE	Maximum likelihood estimation
PDF	Probability density function
KDE	Kernel density estimation
PI	Performance index
SNR	Signal-to-noise ratio
FOL	Frequency overlap level
ICA	Independent component analysis
SCA	Sparse component analysis
NMF	Nonnegative matrix factorization
SOBI	Second-order blind identification algorithm
FOBI	Fourth-order blind identification algorithm
JADE	Joint approximate diagonalization of eigenmatrices
FastICA	Fast independent component analysis
c-FastICA	Complex fast independent component analysis
MIMO	Multiple-input multiple-output
SCM	Spatial covariance matrix
SIMO	Single-input multi-output
BPSK	Binary phase shift keying
QPSK	Quadrature phase shift keying
SGD	Stochastic gradient descent
Adagrad	Adaptive gradient
RMSprop	Root mean square propagation
Adam	Adaptive method
SGDM	Stochastic gradient descent with momentum
BP	Back-propagation
ANN	Artificial neural network
SNN	Simulated neural network
MEG	Magnetoencephalographic

Acknowledgements
The authors would like to thank the handing Associate Editor and the anonymous reviewers for their valuable comments and suggestions for this paper.

Author contributions
LZ designed the work, analyzed and interpreted the data and drafted the manuscript. QH participated in the design of the study, performed the experiments and analysis and helped to draft the manuscript. DD and SZ contributed to literature investigation. GK and LL contributed to revise the manuscript. All the authors read and approved the final manuscript.

Funding
This work was supported in part by the National Natural Science Foundation of China (Nos. 61901209, 61871210 and 61901149), in part by Natural Science Foundation of Hunan Province (No. 2022JJ40377) and in part by the Scientific Research Project of Hunan Provincial Education Department (No. 19C1591).

Availability of data and materials
Please contact the authors for data requests.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 15 September 2022 Accepted: 2 December 2022
Published online: 14 December 2022

References

1. L. Pang, Research on signal separation method for time-frequency overlapped digital communication signal from single antenna. Ph.D. Dissertation, University of Electronic Science and Technology of China (2015)
2. P. Comon, C. Jutten, *Handbook of Blind Source Separation-Independent Component Analysis and Applications* (Elsevier Ltd, Amsterdam, 2010)
3. K.-C. Kwak, W. Pedrycz, Face recognition using an enhanced independent component analysis approach. IEEE Trans. Neural Netw. **18**(2), 530–541 (2007)
4. C. Jutten, J. Hérault, Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture. Signal Process. **24**, 1–10 (1991)
5. P. Comon, Independent component analysis: a new concept? Signal Process. **36**(3), 287–314 (1994)

6. A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution. Neural Comput. **7**(6), 1129–1159 (1995)
7. A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, E. Moulines, A blind source separation technique using second-order statistics. IEEE Trans. Signal Process. **45**(2), 434–444 (1997)
8. L. Tong, R.-W. Liu, V. Soon, Y.-F. Huang, Indeterminacy and identifiability of blind identification. IEEE Trans. Circuits Syst. **38**(5), 499–509 (1991)
9. J. Cardoso, Blind beamforming for non-Gaussian signals. IEE Proc. **140**(6), 362–370 (1993)
10. A. Hyvarinen, Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. Neural Netw. **10**(3), 626–634 (1999)
11. E. Ollila, The deflation-based FastICA estimator: statistical analysis revisited. IEEE Trans. Signal Process. **58**(3), 1527–1541 (2010)
12. A. Dermoune, T. Wei, Fastica algorithm: five criteria for the optimal choice of the nonlinearity function. IEEE Trans. Signal Process. **61**(8), 2078–2087 (2013)
13. T. Wei, A convergence and asymptotic analysis of the generalized symmetric FastICA algorithm. IEEE Trans. Signal Process. **63**(24), 6445–6458 (2015)
14. E. Oja, Z. Yuan, The FastICA algorithm revisited: convergence analysis. IEEE Trans. Neural Netw. **17**(6), 1370–1381 (2006)
15. M. Novey, T. Adali, On extending the complex FastICA algorithm to noncircular sources. IEEE Trans. Signal Process. **56**(5), 2148–2154 (2008)
16. C. Hesse, C. James, The FastICA algorithm with spatial constraints. IEEE Signal Process. Lett. **12**(11), 792–795 (2005)
17. L.-D. Van, D.-Y. Wu, C.-S. Chen, Energy-efficient FastICA implementation for biomedical signal separation. IEEE Trans. Neural Netw. **22**(11), 1809–1822 (2011)
18. L. Pang, Z. Qi, S. Li, B. Tang, A blind signal separation method for single-channel electromagnetic surveillance system. Int. J. Electron. **102**(10), 1634–1651 (2015)
19. J. Liu, H. Song, H. Sun, H. Zhao, High-precision identification of power quality disturbances under strong noise environment based on FastICA and random forest. IEEE Trans. Ind. Inform. **17**(1), 321 (2020)
20. A. Naeem, H. Arslan, Joint radar and communication based blind signal separation using a new non-linear function for fast-ica, in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pp. 1–5 (2021)
21. K.-K. Shyu, M.-H. Lee, Y.-T. Wu, P.-L. Lee, Implementation of pipelined FastICA on FPGA for real-time blind source separation. IEEE Trans. Neural Netw. **19**(6), 958–970 (2008)
22. R. Gribonval, S. Lesage, A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges, in *ESANN'2006 Proceedings—European Symposium on Artificial Neural Network*, pp. 323–330 (2006)
23. P. Georgiev, F. Theis, A. Cichocki, Sparse component analysis and blind source separation of underdetermined mixtures. IEEE Trans. Neural Netw. **16**(4), 992–996 (2005)
24. M. Zibulevsky, P. Kisilev, Y.Y. Zeevi, B.A. Pearlmutter, Blind source separation via multinode sparse representation. Adv. Neural Inf. Process. Syst. **14**, 2353–2362 (2002)
25. F. Georgiev, F. Theis, A. Cichocki, Blind source separation and sparse component analysis of overcomplete mixtures, in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. V-493 (2004)
26. B. Pau, Z. Michael, Underdetermined blind source separation using sparse representations. Signal Process. **81**(11), 2353–2362 (2001)
27. J. Yang, Y. Guo, Z. Yang, S. Xie, Under-determined convolutive blind source separation combining density-based clustering and sparse reconstruction in time-frequency domain. IEEE Trans. Circuits Syst. I Regul. Pap. **66**(8), 3015–3027 (2019)
28. Y. Li, Z.L. Yu, N. Bi, Y. Xu, Z. Gu, S.-I. Amari, Sparse representation for brain signal processing: a tutorial on methods and applications. IEEE Signal Process. Mag. **31**(3), 96–106 (2014)
29. G.R. Tsouri, M.H. Ostertag, Patient-specific 12-lead ECG reconstruction from sparse electrodes using independent component analysis. IEEE J. Biomed. Health Inform. **18**(2), 476–482 (2014)
30. Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, J. Zhang, Blind spectral unmixing based on sparse nonnegative matrix factorization. IEEE Trans. Image Process. **20**(4), 1112–1125 (2011)
31. K. Rahbar, J. Reilly, J. Manton, Blind identification of MIMO fir systems driven by quasistationary sources using second-order statistics: a frequency domain approach. IEEE Trans. Signal Process. **52**(2), 406–417 (2004)
32. B. Gao, W.L. Woo, S.S. Dlay, Unsupervised single-channel separation of nonstationary signals using gammatone filterbank and itakura-saito nonnegative matrix two-dimensional factorizations. IEEE Trans. Circuits Syst. I Regul. Pap. **60**(3), 662–675 (2013)
33. J. Nikunen, T. Virtanen, Direction of arrival based spatial covariance model for blind sound source separation. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(3), 727–739 (2014)
34. M. Pezzoli, J.J. Carabias-Orti, M. Cobos, F. Antonacci, A. Sarti, Ray-space-based multichannel nonnegative matrix factorization for audio source separation. IEEE Signal Process. Lett. **28**, 369–373 (2021)
35. Z. Yang, Y. Xiang, K. Xie, Y. Lai, Adaptive method for nonsmooth nonnegative matrix factorization. IEEE Trans. Neural Netw. Learn. Syst. **28**(4), 94 (2016)
36. D. Gurve, S. Krishnan, Separation of fetal-ECG from single-channel abdominal ECG using activation scaled non-negative matrix factorization. IEEE J. Biomed. Health Inform. **24**(3), 669–680 (2020)
37. B. Gao, W.L. Woo, B.W.-K. Ling, Machine learning source separation using maximum a posteriori nonnegative matrix factorization. IEEE Trans. Cybern. **44**(7), 1169–1179 (2014)
38. H. Szu, P. Chanyagorn, I. Kopriva, Sparse coding blind source separation through powerline. Neurocomputing **48**(1), 1015–1020 (2002)
39. E. Warner, I. Proudler, Single-channel blind signal separation of filtered MPSK signals. IEE Proc. Radar Sonar Navig. **150**(6), 396–402 (2003)
40. L. Pang, B. Tang, A novel method for blind signal separation of single-channel and time-frequency overlapped multi-component signal. Int. J. Inf. Commun. Technol. **8**(2–3), 123–139 (2016)

41. S. Liu, B. Wang, L. Zhang, Blind source separation method based on neural network with bias term and maximum likelihood estimation criterion. Sensors **21**(3), 973 (2021)
42. S. Amari, A. Cichocki, H.H. Yang, A new learning algorithm for blind signal separation, in *Advances in Neural Information Processing Systems*, pp. 757–163 (1996)
43. A.S. Cichocki, Blind source separation: new tools for extraction of source signals and denoising, in *Independent Component Analyses, Wavelets, Unsupervised Smart Sensors, and Neural Networks III*, vol. 5818, pp. 11–25 (2005)
44. H.L. Li, T.T. Adali, Algorithms for complex ml ICA and their stability analysis using Wirtinger calculus. IEEE Trans. Signal Process. **58**(12), 6156–6167 (2010)
45. M. Novey, T.T. Adali, Complex ICA by negentropy maximization. IEEE Trans. Neural Netw. **19**(4), 596–609 (2008)
46. S. Weglarczyk, Kernel density estimation and its application, in *XLVIII Seminar of Applied Mathematics, ITM Web of Conferencess*, vol. 23, p. 00037 (2018)
47. B.W. Silverman, *Density Estimation for Statistics and Data Analysis* (T &F eBook, New York, 1998)
48. G.R. Terrell, D.W. Scott, Variable kernel density estimation. Ann. Stat. **20**(3), 1236–1265 (1992)
49. D.W. Scott, Multivariate density estimation: theory, practice, and visualization. Springer Handbooks of Computational Statistics (2011)
50. A. Van Den Bos, Complex gradient and hessian. IEE Proc. Vis. Image Signal Process. **141**(6), 380–382 (1994)
51. O. Guler, *Foundations of Optimization* (Springer, Berlin, 2010)
52. T. Schaul, S. Zhang, Y. LeCun, No more pesky learning rates, in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, no. 3, PMLR, pp. 343–351 (2013)
53. L. Bottou, Stochastic gradient descent tricks, in *Neural Networks: Tricks of the Trade* (2012)
54. I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, no. 3, PMLR, pp. 1139–1147 (2013)
55. J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
56. M. Mukkamala, M. Hein, Variants of RMSPROP and ADAGRAD with logarithmic regret bounds, in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. PMLR (2017)
57. T. Tieleman, G. Hinton, Lecture 6.5-RMSPROP: divide the gradient by a running average of its recent magnitude, in *COURSERA: Neural Networks for Machine Learning* (2012)
58. G. Hinton, Lecture 6e RMSPROP: divide the gradient by a running average of its recent magnitude, in *COURSERA: Neural Networks for Machine Learning* (2020)
59. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
60. M.R. Gupta, S. Bengio, J. Weston, Training highly multiclass classifiers. J. Mach. Learn. Res. **15**, 1461–1492 (2014)
61. L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learning for speech recognition and related applications: an overview, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8599–8603 (2013)
62. J. Bernal, K. Kushibar, D.S. Asfaw, S. Valverde, A. Oliver, R. Marí, X. Lladó, Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. Artif. Intell. Med. **95**, 64–91 (2018)
63. H.K. Kwan, Y. Cai, A fuzzy neural network and its application to pattern recognition. IEEE Trans. Fuzzy Syst. **2**(3), 185–193 (1994)
64. M.J. El-Khatib, B.S. Abu-Nasser, S.S. Abu-Naser, Glass classification using artificial neural network. Int. J. Acad. Pedagog. Res. **3**(2), 25–31 (2019)
65. V. Zarzoso P. Comon, Comparative speed analysis of FastICA, in *International Conference on Independent Component Analysis and Signal Separation*, Springer, pp. 293–300 (2007)
66. S.C. Douglas, M. Gupta, H. Sawada, S. Makino, Spatio-temporal FastICA algorithms for the blind separation of convolutive mixtures. IEEE Trans. Audio Speech Lang. Process. **15**(5), 1511–1520 (2007)
67. R. Vigário, J. Sarela, V. Jousmiki, M. Hamalainen, E. Oja, Independent component approach to the analysis of EEG and meg recordings. IEEE Trans. Biomed. Eng. **47**(5), 58 (2000)

Publisher's Note