Wireless Personal Communications (2020) 115:1277–1289
https://doi.org/10.1007/s11277-020-07624-4

Single Channel Blind Source Separation Under Deep Recurrent Neural Network

Jiai He[1] · Wei Chen[1] · Yuxiao Song[2]

Published online: 7 July 2020

Abstract
In wireless sensor networks, the signals received by sensors are usually complex nonlinear single-channel mixed signals. In practical applications, it is necessary to separate the useful signals from the complex nonlinear mixed signals. However, the traditional array signal blind source separation algorithms are difficult to separate the nonlinear signals effectively. Building upon the traditional recurrent neural network, we improved the network structure, and further proposed the three layers deep recurrent neural networks to realize single channel blind source separation of nonlinear mixed signals. The experiments and simulation were conducted to verify the performance of this method; the results showed that the mixed signals can be separated excellently and the correlation coefficient can be reached up to 99%. Thus, a new method was given for blind signal processing with artificial intelligence.

Keywords Blind source separation · Single channel · Multi-signals · Deep recurrent neural network

1 Introduction

Sensors, the basic components of the wireless sensor network (WSN), can sense the physical parameters from the surrounding environment and further obtain useful data[1]. These data from multiple channels are aggregated in the converged processor after that mixed data of useful information and noise is synthesized[2]. The key issue is how to extract the data with information from the mixed data signal with noise. Blind source separation (BSS), defined as a task of separating the source signals from the observed signals in the case that the theoretical model of the signal and the source signal can not be accurately known, can separate the multi-mixed signals and extract the source signals with information data.

✉ Wei Chen
chenwei_scholar@163.com

1 School of computer and communication, Lanzhou University of Technology, Lanzhou 730050, China

2 Shaanxi FiberHome Telecommunication Technologies Co., Ltd, Xi'an 710000, China

At the end of the 20th century, BSS algorithms developed rapidly[3]. Most research on BSS has been carried out in multi-channel array signal in the beginning. The source signal matrix is obtained by solving the mixed-signal matrix and optimizing the objective function, where each column in the mixed-signal matrix corresponds to a separated source signal. However, multi-channel BSS cannot guarantee the sequence of separated signals, and the real-time performance is unsatisfactory. On the basis of multi-channel blind source separation, single channel blind source separation is proposed. In[4–6], the independent component analysis (ICA) is employed to realize single channel blind source separation that is translated into a multi-channel separation through channel expansion. Although the efficiency of the algorithm is improved, there are still some problems such as uncertain separation sequence and the low correlation coefficient between the separated signals and the source signals.

Due to the high adaptability and scalable to complex non-linear problems, the neural network has become a hot issue, and further is utilized to process BSS. Pehlevan et al. regarding the separation of non-negative blind sources as a similarity matching problem, and then update the weight of each node according to the learning rules in biology; finally, the similarity matching target is used to complete the non-negative ICA derivation algorithm[7]. In[8–10], the support vector machine (SVM) model and deep learning model are developed to separate the mixed signals of human and machine voice. Such an approach gets a preferable performance by leveraging ICA to solve single channel blind separation through channel expansion. Sun et al.[11] used SVM to solve the problem of single channel blind source separation in MIMO systems, and achieved a better separation. In[12], Fu et al. define the source signal as the weight vector matrix of the single layer perception, use the approximate ℓ_0 norm as the output error rule of the perception, and consider the parameter sequence and the optimal learning factor as parameters in the iterative process. Subsequently, the source signal is recoded by adjusting the weight vector of the perceptron and then recovered by the sparse algorithm, which effectively reduces the computational complexity and improves the separation accuracy. Based on the traditional BSS, a self-feedback fully-connected neural network algorithm is proposed in[13], which can effectively control the local minimum and increase the convergence speed of the system. However, it is suitable for solving nonlinear mixed blind source separation problems, but not for solving single channel problems. Wang et al. proposed a fault time series prediction method based on long and short-term memory (LSTM) recurrent neural networks. To minimize the prediction error, it uses the grid search optimization algorithm, and has strong applicability and higher accuracy in time series analysis[14]. Besides, Fang et al. employ convolution neural networks to solve the blind quality assessment of super-resolution images[15]. Furthermore, Singh et al. adopt a sparse coding scheme to provide a separation basis for dictionary learning in each layer of the neural network, and then solved the single channel blind source separation problem by energy decomposition[16].

Building upon the above research, a deep circulation neural network is introduced to solve the problem of nonlinear mixed-signal separation in this paper. Firstly, the learning model of the source signal is obtained by adaptive learning about the source signal. Then the network is trained to establish a deep cycle network model. After the mixed signal is inputted into the network, the blind source separation is carried out by learning the temporal relation of the signals. Finally, we get the separated signals.

The rest of the paper is organized as follows. The single-channel blind source separation model and its adaptive learning algorithm are launched in Sect. 2. A deep recurrent blind source separation neural network is studied in Sect. 3. The simulation results are presented in Sect. 4. Finally, we come to the conclusions in Sect. 5.

2 Single Channel Blind Source Separation Model

In a wireless sensor network, the signal received by the final processor is mixed signal with noise signals, constructed using the formula

$$\boldsymbol{y}(k) = \boldsymbol{A}\left[\boldsymbol{x}_1(k) + \boldsymbol{x}_2(k) + \cdots + \boldsymbol{x}_n(k)\right], \quad k = 1, \ldots, m, \tag{1}$$

where $\boldsymbol{A}$ is the mixed signal coefficient matrix, $\boldsymbol{y}(k)$ is the mixed signal received by the receiver, and $\boldsymbol{x}_n(k)$ is source signal that needs to be separated. In the following part, we will specifically introduce how to solve the problem of single channel blind source separation using a deep recurrent neural network.

For signal $\boldsymbol{x}_n(k)$, it is necessary to use adaptive learning to obtain the corresponding estimated signal $\boldsymbol{x}_n(k)$ that is used to train the deep recurrent neural network. We first need to get a prior probability distribution of the parameter θ. Assuming that $\hat{\theta}$ represents the point estimate of θ, the point estimate of the nth point is

$$\hat{\theta}_n = g\left[x_n(1), x_n(2), \ldots, x_n(k)\right], \quad n = 1, \ldots, N, \tag{2}$$

where g obeys a uniform Gaussian distribution, i.e., $g[\boldsymbol{x}(k)] = N\left(\boldsymbol{x}(k);\theta, \sigma^2\right)$, then the probability density function of g is

$$p\left[\boldsymbol{x}(k);\theta, \sigma^2\right] = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\frac{[\boldsymbol{x}(k) - \theta]^2}{\sigma^2}\right). \tag{3}$$

The sample mean is calculated by the common estimator of the Gaussian mean, expressed as

$$\hat{\theta}_n = \frac{1}{k}\sum_{i=1}^{k} x(i). \tag{4}$$

And the unbiased estimate is used in this algorithm, i.e.,$\text{bias}\left(\hat{\theta}_n\right) = 0$.

Assuming that there is a data set $\boldsymbol{x}(k) = [x(1), x(2), \ldots, x(k)]$ with k samples, the data set is generated from the unknown real data generation distribution $p_m(x)$, and $p_m(x;\hat{\theta}_n)$ is determined by $\hat{\theta}_n$ in the same space. Namely, $p_m(x;\hat{\theta}_n)$ can map any input to a number to estimate the true $p_d(x)$. Then the maximum likelihood estimate for $\hat{\theta}_n$ is

$$\hat{\theta}_{n,ML} = \arg\max_{\hat{\theta}_n} \sum_{i=1}^{k} \log p_m\left(x(i);\hat{\theta}_n\right). \tag{5}$$

It is difficult to solve the multi-probability product problem, which is accompanied by numerical underflow simultaneously. Furthermore, it can be observed that the likelihood logarithm does not change the maximum likelihood estimate, and the product calculation can be transformed into a simpler summation form, so Eq. (5) can be converted to

$$\hat{\theta}_{n,ML} = \arg\max_{\hat{\theta}_n} \sum_{i=1}^{k} \log p_m\left(x(i);\hat{\theta}_n\right), \tag{6}$$

where the optimal estimation of the maximum likelihood estimation value $\hat{\theta}_{n,ML}$, is obtained by estimating the parameter θ; and the optimal estimation value obtained by the parameter estimation is combined with the sample data $[x_n(1), x_n(2), \ldots, x_n(k)]$ to perform likelihood and priori estimation. At the first moment, the estimated value of the sample is $\hat{x}_n(1)$, and the mean square error between the estimated value and the true value is ε_1, then the optimal estimate of the second sample can be expressed as

$$p\left[\hat{\theta}_{n,ML}|\hat{x}_n(1)\right] = \varepsilon_1 \frac{p\left[\hat{x}_n(1)|\hat{\theta}_{n,ML}\right]p\left[\hat{\theta}_{n,ML}\right]}{p\left[\hat{x}_n(1)\right]}, \tag{7}$$

$$p\left[\hat{x}_n(2)|\hat{x}_n(1)\right] = \int p\left[\hat{x}_n(2)|\hat{\theta}_{n,ML}\right]p\left[\hat{\theta}_{n,ML}|\hat{x}_n(1),\hat{x}_n(2)\right]d\hat{\theta}_{n,ML}, \tag{8}$$

where $\hat{x}_n(2)$ is the optimal estimation of the sample at the second moment. The optimal estimation formula can be extended analogically to the $k+1$ th moment

$$\begin{aligned} &p\left[\hat{\theta}_{n,ML}|\hat{x}_n(1),\hat{x}_n(2),\dots,\hat{x}_n(k)\right] \\ &\quad = \varepsilon_k \frac{p\left[\hat{x}_n(1),\hat{x}_n(2),\dots,\hat{x}_n(k)|\hat{\theta}_{n,ML}\right]p\left[\hat{\theta}_{n,ML}\right]}{p\left[\hat{x}_n(1),\hat{x}_n(2),\dots,\hat{x}_n(k)\right]}, \end{aligned} \tag{9}$$

$$\begin{aligned} &p\left[\hat{x}_n(k+1)|\hat{x}_n(1),\hat{x}_n(2),\dots,\hat{x}_n(k)\right] \\ &\quad = \int p\left[\hat{x}_n(k+1)|\hat{\theta}_{n,ML}\right]p\left[\hat{\theta}_{n,ML}|\hat{x}_n(1),\hat{x}_n(2),\dots,\hat{x}_n(k)\right]d\hat{\theta}_{n,ML}, \end{aligned} \tag{10}$$

where $\hat{x}_n(k+1)$ is the optimal estimation of the sample at time $k+1$. Then the estimated signal of source signal $\boldsymbol{x}_n(k)$ is

$$\hat{\boldsymbol{x}}_n(k+1) = \left[x_n(1), x_n(2), \dots, x_n(k+1)\right]. \tag{11}$$

Subsequently, the network is trained by using a supervised learning algorithm, and we have

$$p\left[\boldsymbol{x}_n^t(k)|\hat{\boldsymbol{x}}_n(k);\hat{\theta}_{n,ML}\right] = N\left[\boldsymbol{x}_n^t(k);\hat{\theta}_{n,ML}^T\hat{\boldsymbol{x}}_n(k),\sigma\right], \tag{12}$$

where $\boldsymbol{x}_n^t(k)$ is the output data of the training; N denotes the standard normal distribution; T means the transport of matrix. In order to control the mean of the output in the interval [0, 1], the output value is clipped using the sigmoid function as

$$p\left[\boldsymbol{x}_n^t(k) = 1|\hat{\boldsymbol{x}}_n(k);\hat{\theta}_{n,ML}\right] = \sigma\left[\hat{\theta}_{n,ML}^T\hat{\boldsymbol{x}}_n(k)\right], \tag{13}$$

where $\sigma = \frac{1}{1+e^{-x}}$ is the sigmoid function.

3 Deep Recurrent Blind Source Separation Neural Network

3.1 Stochastic Gradient Descent

In the following, the stochastic gradient descent method is used to solve the updating of weights in the network training and deep recurrent blind source separation. For training data, the likelihood function of the negative conditional logarithm can be written as

$$J(\hat{\theta}_{n,ML}) = \varepsilon_{\hat{x}_n(k),x_n^t(k)} L(\hat{\boldsymbol{x}}_n(k), \boldsymbol{x}_n^t(k), \hat{\theta}_{n,ML}) = \frac{1}{k}\sum_{k=1}^{k} (x_n(k) - x_n^t(k)) L(\hat{x}_n(k), x_n^t(k), \hat{\theta}_{n,ML}), \tag{14}$$

where L is the loss of each sample, and $L(\hat{\boldsymbol{x}}_n(k), \boldsymbol{x}_n^t(k), \hat{\theta}_{n,ML}) = -\log p(\boldsymbol{x}_n^t(k)|\hat{\boldsymbol{x}}_n(k), \hat{\theta}_{n,ML})$. Thus, the gradient of parameter $\hat{\theta}_{n,ML}$ is

$$\nabla_{\hat{\theta}_{n,ML}} J(\hat{\theta}_{n,ML}) = \frac{1}{k}\sum_{k=1}^{k} \nabla_{\hat{\theta}_{n,ML}} (\hat{x}(k), x_n^t(k), \hat{\theta}_{n,ML}). \tag{15}$$

Let $\nabla_{\hat{\theta}_{n,ML}} J(\hat{\theta}_{n,ML}) = g$, then continuous updating leads to $\hat{\theta}_{n,ML} - \mu\hat{\theta}_{n,ML} \to \hat{\theta}_{n,ML}$ in which μ denotes the learning rate, set to $\mu = 0.05$ in this work.

3.2 Blind Source Separation Network

Based on[17], an improved deep recurrent neural network model is developed to achieve the separation of the useful signal from the receiver of WSN, as shown in Fig. 1.

In Fig. 1, the left side is a schematic diagram of the blind separation using the deep recurrent neural network, and the right side is its expansion. Furthmore, $\boldsymbol{y}(k)$ is mixed input signal; $\boldsymbol{h}(k)$ is a hidden layer of the recurrent neural network; $\hat{\boldsymbol{x}}_n(k)$, generated by adaptive learning for source signal $\boldsymbol{x}_n(k)$, is the training signal for network; $\boldsymbol{L}(k)$ is defined as network loss function; and U, V, W represent the system weight coefficient matrix. The cyclic neural

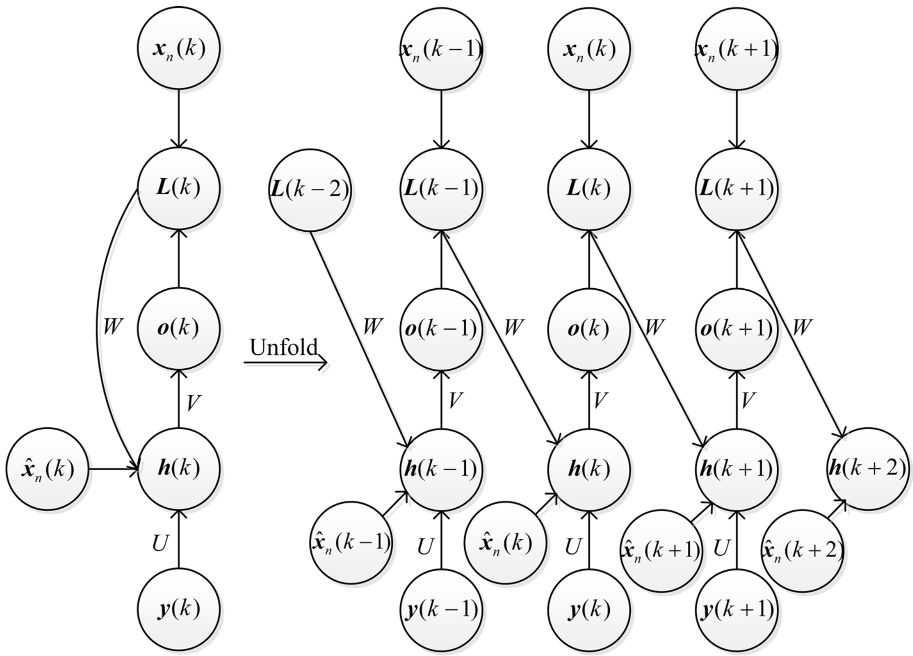


Fig. 1 Schematic Diagram of Rrcurrent Neural Network

network propagates from the initial state $\boldsymbol{h}(0)$ to the preceding term, and updates the network state according to the iteration of the time state from $t = 1$ to $t = \tau$, through using the following formula

$$\boldsymbol{a}(k) = \hat{\boldsymbol{x}}_n(k) + U\boldsymbol{y}(k) + W\boldsymbol{L}(k-1) + b, \tag{16}$$

$$\boldsymbol{h}(k) = \text{sigmoid}(\boldsymbol{a}(k)), \tag{17}$$

$$\boldsymbol{o}(k) = V\boldsymbol{h}(k) + c, \tag{18}$$

$$\boldsymbol{L}(k) = \varepsilon\left[\boldsymbol{x}_n(k), \boldsymbol{o}(k)\right]. \tag{19}$$

In the above state update equations, $\boldsymbol{a}(k)$ is the network hidden layer update function, and the sigmoid activation function is used to limit the amplitude of it. For the network loss function $\boldsymbol{L}(k)$, the decision is made by the mean square error function $\varepsilon(\cdot)$, and the network weight coefficient is updated by the gradient of the true value and the network output value error. In additional, the deviation b, c are the disturbance near the hidden layer and the output layer in the network, which can be neglected. Henceforth, Eqs. (16) and (18) can be rewritten as

$$\boldsymbol{a}(k) = \hat{\boldsymbol{x}}_n(k) + U\boldsymbol{y}(k) + W\boldsymbol{L}(k-1), \tag{20}$$

$$\boldsymbol{o}(k) = V\boldsymbol{h}(k). \tag{21}$$

For network loss function $\boldsymbol{L}(k)$, compared to output value , the total loss of true signal data $\boldsymbol{x}_n(k)$ is the sum of the losses of all time steps. Extending Eq. (19) to each time step, we have

$$\begin{aligned} \boldsymbol{L}(k) &= \varepsilon\left[(x_n(1), x_n(2), \ldots, x_n(k)), (o(1), o(2), \ldots, o(k))\right] \\ &= -\varepsilon \sum_{k=1}^{k} \log p[o(k)|x_n(k)], \end{aligned} \tag{22}$$

where $p[o(k)|x_n(k)]$ is the loss function of the output value and the true signal error on the same time step.

The gradient of the recurrent neural network is calculated by the stochastic gradient descent method for the calculated weights U, V, W and the discrete time series $\boldsymbol{y}(k)$, $\boldsymbol{h}(k)$, $\boldsymbol{o}(k)$, $\boldsymbol{L}(k)$ indexed by the discrete time point k in the recurrent neural network. Recursion $\frac{\partial L}{\partial \boldsymbol{L}(k)} = 1$ starts from the final network loss node. Since $o(\tau)$ has only $L(\tau)$ as its subsequent node in the last discrete time step τ, the gradient of node $\boldsymbol{o}(k)$ is

$$\left(\nabla_{\boldsymbol{o}(k)} L\right)_n = \frac{\partial L}{\partial \boldsymbol{o}_n(k)} = \frac{\partial L}{\partial \boldsymbol{L}(k)} \frac{\partial \boldsymbol{L}(k)}{\partial \boldsymbol{o}_n(k)}. \tag{23}$$

For the hidden node $\boldsymbol{h}(\tau)$, the source signal estimation signal $\hat{\boldsymbol{x}}_n(\tau-1)$ and the loss node $\boldsymbol{L}(\tau-1)$ have a common influence on them, so there are

$$\nabla_{\boldsymbol{h}(k)} L = V^T \nabla_{\boldsymbol{o}(k)} L + \nabla_{\boldsymbol{L}(k)} L + \nabla_{\hat{\boldsymbol{x}}(k)} L. \tag{24}$$

In the case of reverse iteration from $k = k + 1$ to k, the gradient of the hidden layer is

$$\nabla_{\boldsymbol{h}(k)}L = \left(\frac{\partial \boldsymbol{o}(k)}{\partial \boldsymbol{h}(k)}\right)^T (\nabla_{\boldsymbol{o}(k)}L) + \frac{\partial L}{\partial \boldsymbol{L}(k-1)} + \frac{\partial L}{\partial \hat{\boldsymbol{x}}(k-1)}. \tag{25}$$

Similarly, for the weight coefficients U, V, W, the gradient descent update formula is

$$\nabla_U(L) = \sum_{k=1}^{k}\sum_{n=1}^{k}\left(\frac{\partial L(k)}{\partial h_n(k)}\right)\nabla_U h_n(k), \tag{26}$$

$$\nabla_V(L) = \sum_{k=1}^{k}\sum_{n=1}^{k}\left(\frac{\partial L(k)}{\partial o_n(k)}\right)\nabla_V o_n(t), \tag{27}$$

$$\nabla_W(L) = \sum_{k=1}^{k}\sum_{n=1}^{k}\left(\frac{\partial L(k-1)}{\partial h_n(k)}\right)\nabla_W h_n(t). \tag{28}$$

Next, the mixed data signal $\boldsymbol{y}(k)$ is entered into the system, and similarity matching is implement for $\boldsymbol{y}(k)$ by learning signal $\hat{\boldsymbol{x}}_n(k)$, expressed using the formula

$$\boldsymbol{x}_{n,mat}(k) = M\left[\hat{\boldsymbol{x}}_n(k), \boldsymbol{y}(k)\right]. \tag{29}$$

Subsequently, by the multi-layer recurrent neural network, the weight coefficients are updated continuously by reducing the system training loss and the separation loss, and the updated weight coefficients are used to operate the next discrete time separation. Thus, the similarity matching function is approximated to the estimation value of the source signal gradually, until the loss function drops below the threshold. Finally, the separation signal is obtained as

$$\boldsymbol{x}_{n,mat}(k) = \left[x_{n,mat}(1), x_{n,mat}(2), \dots, x_{n,mat}(k)\right]. \tag{30}$$

The above process is summarized in Table 1.

Table 1 Deep recurrent blind source separation network steps

Step	Content
1	Obtain the estimated signal of source signal by Eqs. (2)–(11)
2	Input the learning signal into the recurrent network for training according to Eqs. (12)–(13)
3	Input the mixed signal into the network in accordance with Eqs. (16)–(21) and Fig. 1
4	Get matched signal by similarity matching using Eq. (29)
5	According to Eq. (22), establish the loss function by matched signal that has been obtained in step 4
6	According to Eqs. (23)–(28), modify the weight coefficient with the loss function continuously
7	Get the optima separation signal

4 Experimental Simulation

In order to verify the feasibility of the single channel blind source separation algorithm with the deep recurrent neural network, the three nonlinear signals are mixed into the deep recurrent neural network. And the correlation coefficient ξ between the separated signal and the real signal is used as the evaluation index[18], expressed as

$$\xi_{ij} = \xi_{ij}\left(\boldsymbol{x}_{n,mat}(k), \boldsymbol{x}_n(k)\right) = \frac{\sum_{i,j=1}^{k} x_{i,n,mat}(k), x_{j,n}(k)}{\sqrt{\sum_{i=1}^{k} x_{i,n,mat}^2(k), \sum_{j=1}^{k} s_{j,n}^2(k)}}. \tag{31}$$

Simultaneously, the loss value is used to judge the number of layers of the deep neural network.

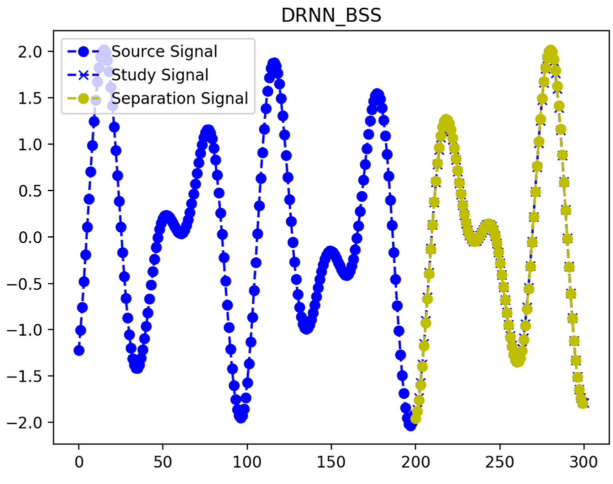
The three layer source signals are the random sine and cosine functions $\boldsymbol{x}_1(k)$, $\boldsymbol{x}_2(k)$, and $\boldsymbol{x}_3(k)$ corrupted by various interference components, which are generated by the random, math, and Numpy libraries with Python respectively. The learning signals obtained through adaptive learning are $\hat{\boldsymbol{x}}_1(k)$, $\hat{\boldsymbol{x}}_2(k)$, $\hat{\boldsymbol{x}}_3(k)$ respectively. Thus, the maxed signal is expressed as $\boldsymbol{y}(k) = \hat{\boldsymbol{x}}_1(k) + \hat{\boldsymbol{x}}_2(k) + \hat{\boldsymbol{x}}_3(k)$. When the learning signal $\hat{\boldsymbol{x}}_n(k)$ is generated through adaptive learning, the activation value generated by limiting the Sigmoid activation function is

$$\begin{bmatrix} \hat{\boldsymbol{x}}_1(k) \\ \hat{\boldsymbol{x}}_2(k) \\ \hat{\boldsymbol{x}}_3(k) \end{bmatrix} = \begin{bmatrix} 0.391 \\ 0.972 \\ 0.547 \end{bmatrix} \tag{32}$$

The signal activation value is approximated by the deep recurrent blind source separation neural network, and the experimental signal simulation diagrams are shown in Figs. 2, 3, and 4. For easy viewing, in each figure, the first half is the real source signal, and the second half the separated signal obtained by our proposed method.

It can be seen from the above three figures that adaptive learning strategy can better obtain the learning signal. And the signal can be separated well by the deep circulation blind source separation neural network. For the correlation coefficient ξ of the three

Fig. 2 The first signal



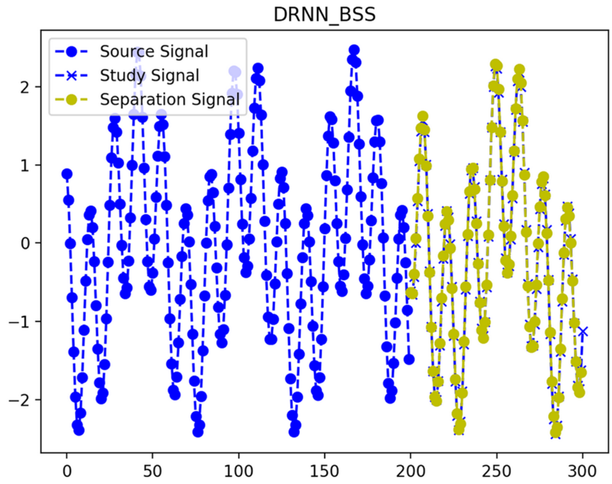


Fig. 3 The second signal

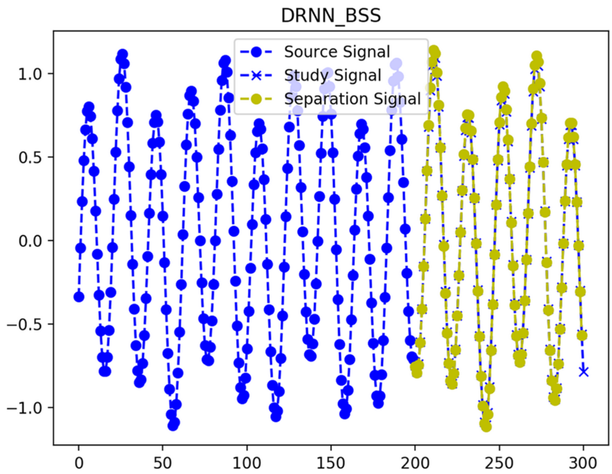


Fig. 4 The third signal

signals, the comparison values under different sampling points are utilized, and the concrete results are shown in Fig. 5.

For deep learning networks, the loss is inevitable, both in learning training and testing separation. For different sampling points, 100 Monte Carlo simulations are performed for each point, and the results are weighted averagely. The loss values of the training loss and the test loss test points are shown in Tables 2 and 3 and Fig. 6.

It can be seen from Tables 2 and 3 and Fig. 6 that, in the three layer recurrent network, the training learning loss and the test separation loss reaches basically a stable state for the signal of any sampling point, and has good convergence. It is better to separate the source signal from the nonlinear mixed signal. It can be concluded that the algorithm can solve the blind source separation problem more satisfactorily by the three layer deep recurrent neural network.

Fig. 5 Correlation coefficient reference chart

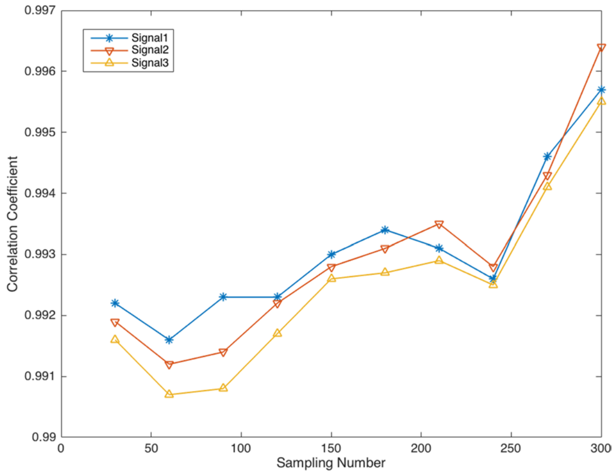


Table 2 Training learning loss

Simple	30	60	90	120	150	180	210	240	270	300
1-layer	36.484	72.093	108.436	134.630	152.797	211.138	287.929	312.492	276.583	356.617
2-layer	0.721	0.893	1.378	1.959	1.999	1.285	1.160	4.445	1.990	1.766
3-layer	0.416	0.495	0.548	0.752	0.767	0.694	0.746	1.053	0.991	0.784

Table 3 Test separation loss

Simple	30	60	90	120	150	180	210	240	270	300
1-layer	37.643	82.214	107.837	138.497	174.103	238.641	265.517	304.054	285.781	405.623
2-layer	0.726	0.875	0.952	1.863	1.392	1.154	1.246	3.745	2.245	1.922
3-layer	0.430	0.521	0.537	0.767	0.695	0.656	0.775	1.066	0.836	0.782

5 Conclusion

Based on the recurrent neural network and adaptive learning, we propose a deep recurrent blind source separation neural network algorithm in the context of wireless sensor networks. The source signal is first used to obtain the training signal through adaptive learning. After that the neural network is trained, the mixed signal is inputted into the neural network to obtain the separated signal by similarity matching. The simulation results show that the proposed algorithm can obtain better convergence effect in the three-layer deep recurrent neural network, and can solve the single-channel blind source separation problem of nonlinear mixed signals. In future work, the single-channel blind source separation problem of nonlinear mixed signals can be performed by combining the recurrent neural network and the convolutional neural network.

Fig. 6 Loss convergence

Acknowledgements The funding was provided by National Natural Science Foundation of China(Grant No. 61561031)

References

1. Jung, J. Y., Seo, D. Y., & Lee, J. R. (2018). Counter-based broadcast scheme considering reachability, network density, and energy efficiency for wireless sensor networks. *Sensors*, *18*(1), 120.
2. Zuhairy, R. M., & Al Zamil, M. G. (2018). Energy-efficient load balancing in wireless sensor network: An application of multinomial regression analysis. *International Journal of Distributed Sensor Networks*, *14*(3), 1550147718764641.
3. Belouchrani, A., Abed-Meraim, K., Cardoso, J. F., & Moulines, E. (1997). A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, *45*(2), 434–444.
4. Davies, M. E., & James, C. J. (2007). Source separation using single channel ICA. *Signal Processing*, *87*(8), 1819–1832.
5. Kawamura, A., Nada, E., & Iiguni, Y. (2013). Single channel blind source separation of deterministic sinusoidal signals with independent component analysis. *IEICE Communications Express*, *2*(3), 104–110.
6. Calderón-Piedras, J. S., Orjuela-Cañón, Á. D., & Sanabria-Quiroga, D. A. (2014). Blind source separation from single channel audio recording using ICA algorithms. In *2014 XIX symposium on image, signal processing and artificial vision* (pp. 1–5). IEEE.
7. Pehlevan, C., Mohan, S., & Chklovskii, D. B. (2017). Blind nonnegative source separation using biological neural networks. *Neural Computation*, *29*(11), 2925–2954.
8. Abouzid, H., & Chakkor, O. (2017). Blind source separation approach for audio signals based on support vector machine classification. In *Proceedings of the 2nd international conference on computing and wireless communication systems* (pp. 39).
9. Xue-Feng, X., & Guo-Dong, Z. (2016). A survey on deep learning for natural language processing. *Acta Automatica Sinica*, *42*(10), 1445–1465.
10. Wake, M., Bando, Y., Mimura, M., Itoyama, K., Yoshii, K., & Kawahara, T. (2017). Semi-Blind speech enhancement basedon recurrent neural network for source separation and dereverberation. In *2017 IEEE 27th international workshop on machine learning for signal processing (MLSP)* (pp. 1–6). IEEE.
11. Sun, C., Yang, L., Du, J., Sun, F., Chen, L., Xi, H., et al. (2017). Blind source separation and equalization based on support vector regression for MIMO systems. *IEICE Transactions on Communications*,. https://doi.org/10.1587/transcom.2016EBP3473.
12. Fu, W., Nong, B., Zhou, X., Liu, J., & Li, C. (2018). Source recovery in underdetermined blind source separation based on artificial neural network. *China Communications*, *15*(1), 140–154.

13. Li, S., Yang, B., Zhang, J., & Wu, H. (2012). BSS algorithm based on fully connected recurrent neural network and the application in separation of speech signals. In *2012 Spring Congress on Engineering and Technology* (pp. 1–3).
14. Wang, X., Wu, J., Liu, C., Yang, H. Y., Du, Y. L., & Niu, W. S. (2018). Exploring LSTM based recurrent neural network for failure time series prediction. *Journal of Beijing University Aeronautics Astronautics*, *44*(4), 772–784.
15. Fang, Y., & Zhang, C. (2017). Convolutional neural network for blind quality evaluator of image super-resolution. In *2017 International symposium on intelligent signal processing and communication systems (ISPACS)* (pp. 28–33).
16. Singh, S., & Majumdar, A. (2017). Deep sparse coding for non-intrusive load monitoring. *IEEE Transactions on Smart Grid*, *9*(5), 4669–4678.
17. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). CNN-RNN: A unified framework for multi-label image classification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2285–2294).
18. Weidong, Y. E., & Tao, Y. (2016). Single-channel vibration signal blind source separation by combining extreme-point symmetric mode decomposition with time-frequency analysis. *Journal of Computer Applications*, *36*(10), 2933–2939.

Jiai He received the B.E. degree in Physics from Northwest Normal University, Lanzhou, China, in 1993, and received the M.Sc. degree in information and communication engineering from Lanzhou University, Lanzhou, China, in 2004. From 2004 to now, he is an associate professor and the master's supervisor with the school of computer and communication, Lanzhou University of Technology. His research interests include signal processing, blind source separation, signal detection and estimation, cognitive radio, and machine learning.

Wei Chen received the B.E. degree in communication engineering from Lanzhou University of Technology, Lanzhou, China, in 2017, where he is currently pursuing the M.Sc. degree with the communication and information system in Lanzhou University of Technology, Lanzhou, China. His research interests include signal processing, cognitive radio, and machine learning.

Yuxiao Song received the B.E. degree in communication engineering from Xi'an University of Posts and Telecommunications, Xi'an, China, in 2016, and the M.Sc. degree in signal and information processing from Lanzhou University of Technology, Lanzhou, China, in 2019. He is currently an Assistant Engineer with Shaanxi FiberHome Telecommunication Technologies Co., Ltd., Xi'an, China. His research interests include signal processing, blind source separation, and machine learning.