Iterative deblending for simultaneous source data using the deep neural network

Zu, Shaohuan; Cao, Junxing; Qu, Shan; Chen, Yangkang

DOI
10.1190/geo2019-0319.1

Publication date
2020

Document Version
Final published version

Published in
Geophysics

Citation (APA)
Zu, S., Cao, J., Qu, S., & Chen, Y. (2020). Iterative deblending for simultaneous source data using the deep neural network. *Geophysics*, *85*(2), V131-V141. https://doi.org/10.1190/geo2019-0319.1

Iterative deblending for simultaneous source data using the deep neural network

Shaohuan Zu[1], Junxing Cao[1], Shan Qu[2], and Yangkang Chen[3]

ABSTRACT

Simultaneous source technology can accelerate data acquisition and improve subsurface illumination. But those advantages are compromised due to dense interference. To address the intense interference in simultaneous source data, we have investigated a method based on a deep neural network. The designed architecture consists of convolutional and deconvolutional networks. The convolutional network can learn the local features of the training data set, and the deconvolutional network constructs the output using the extracted features to match the ground truth. Because the main computational cost results from the optimization of the network parameters, the trained network can separate simultaneous source data efficiently. Besides, with the given dithering code, we embed the trained network into an iterative framework that can further improve the deblending. A numerical test on synthetic data demonstrates that the iterative framework with the trained network can obtain comparable performance with high efficiency compared to the conventional method. Next, we test our method with two different trained networks (one is from a synthetic data set, and the other is from a field data set) on field data. The test results confirm the performance of our method.

INTRODUCTION

Conventional seismic acquisition requires a large temporal interval to ensure that receivers measure the response from only one source. In simultaneous source acquisition, multiple sources are fired and receivers measure responses from multiple sources, which is very different from traditional seismic acquisition (Beasley, 2008; Berkhout, 2008; Abma et al., 2010). In recent decades, simultaneous source technology has become popular in academia and industry, due to its advantages in enhancing acquisition efficiency and improving the quality of imaging (Abma et al., 2012; Bagaini et al., 2012; Beasley et al., 2012; Alexander et al., 2013; Baardman and van Borselen, 2013; Ibrahim and Sacchi, 2013; Li et al., 2013). Due to the improved source sampling and higher energy of the signal ratio, it is beneficial for obtaining high-quality imaging and accurate inversion (Beasley et al., 2012; Berkhout and Blacquière, 2013; Wu et al., 2018). Unfortunately, those benefits are hindered due to interference caused by the blending.

To obtain good results from simultaneous source data, two categories of methods are explored. The first category is direct imaging, which migrates the blended data directly and uses some additional constraints to suppress the interference (Verschuur and Berkhout, 2011; Xue et al., 2016b; Bai et al., 2018; Zhang et al., 2018b, 2018c). Berkhout et al. (2012) illustrate how the illumination property can be further improved using blended data. Xue et al. (2016b) implement shaping regularization into reverse time migration to attenuate the interference caused by simultaneous sources.

Deblending is the second category, which first separates the simultaneous source data into the deblended data as received from traditional seismic acquisition, so the conventional seismic workflow can directly process the deblended data (Doulgeris et al., 2012; Mahdad et al., 2012; Wapenaar et al., 2012; Cheng and Sacchi, 2015, 2016; Kumar et al., 2015; Zhou et al., 2016; Andersson et al., 2017; Chen et al., 2018). In this paper, we focus on the second category.

Deblending methods can also be summarized in two ways. With a dithering code, the response of one special source is viewed as the coherent signal, whereas that of other sources is viewed as incoherent interference in the common-receiver domain. The first deblending method considers the coherent components as signal and the incoherent components as noise. Thus, denoising is implemented to separate

Manuscript received by the Editor 17 May 2019; revised manuscript received 27 October 2019; published ahead of production 15 November 2019; published online 30 January 2020.

[1]Chengdu University of Technology, State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation, Chengdu, China and Chengdu University of Technology, College of Geophysics, Chengdu 610059, China. E-mail: zushaohuan19@cdut.edu.cn (corresponding author); caojx@cdut.edu.cn.
[2]Delft University of Technology, Delphi Consortium, Delft, Netherlands. E-mail: s.qu@tudelft.nl.
[3]Zhejiang University, School of Earth Sciences, Hangzhou 310027, China. E-mail: yangkang.chen@zju.edu.cn.

blended data (Huo et al., 2012). Filtering-based deblending has high efficiency, but when the subsurface structure is complex, the deblending performance will not be good.

The second deblending method is by inversion that estimates coherent and incoherent components from the blended record simultaneously. Compared to denoising, deblending by inversion has good performance (van Borselen et al., 2012) because deblending via inversion takes all of the information into consideration, such as the dithering time and the spatial positions of shooting sources. The inversion problem is underdetermined because the number of the blended gather is less than that of the unblended gather. A constraint term is required to regularize the inversion to be the desired solution, which includes sparsity constraints in the Radon domain (Akerberg et al., 2008; Ibrahim and Sacchi, 2013), curvelet domain (Lin and Herrmann, 2009), seislet domain (Chen et al., 2014), and the low-rank constraint (Cheng and Sacchi, 2015; Xue et al., 2016a).

In addition to deblending algorithms, some researchers aim to optimize simultaneous source survey parameters to yield a notable enhancement in the deblending (Abma, 2014; Nakayama et al., 2018). If the survey parameters (such as dithering time, shot inline spacing) are not appropriate, the coherency difference between the signal and the interference will be small, which makes the deblending difficult. To maximize the coherence difference in a limited range, Zu et al. (2016) propose a periodically varying code to improve the deblending performance for marine seismic acquisition. Halliday and Moore (2018) investigate the difference between simultaneous source encoding using time dithers, time sequences, and phase sequences.

In recent years, deep learning has shown excellent performance in seismic exploration. Wang et al. (2018) implement a deep-learning-based approach that learns the nonlinear map from the training data to interpolate seismic data. Yu et al. (2019) introduce deep learning to seismic noise attenuation, which does not require linear events, sparsity, or low-rank assumptions. Zhang et al. (2018a) apply conventional neural networks (CNNs) to predict lithology and have obtained accurate predicted results. Yang and Ma (2019) use the deep convolutional neural network to build the initial velocity for full-waveform inversion. This big-data-driven method does not require physical prior knowledge. Based on a deep neural network, Li et al. (2019) propose SeisInvNet to learn the mapping relation from time-series data to a spatial image. Chen et al. (2019) implement a convolutional neural network to classify waveforms, and then they pick arrivals from the classified waveform group. Similar to image segmentation, Shi et al. (2018) design a multilayer convolutional neural network to recognize salt bodies, where the labels for training the network are the manual interpretation results. Pham et al. (2018) adopt the encoder-decoder convolutional neural network to detect channels automatically. Richardson and Feller (2019) adopt a U-net model incorporating a ResNet architecture to separate blended common-offset gathers where the training data are generated from field and synthetic data. Baardman and Tsingas (2019) implement a convolutional neural network to classify the blended and unblended data patches, and then they use a regression algorithm to separate the identified blended patch.

In this paper, we implement a deep neural network to separate blended data iteratively. The designed architecture includes two parts. The first part is a convolutional network to extract local features. The second part is a deconvolutional network to construct the deblended data using the features extracted from the convolutional network. The highest computational cost is spent on the training period to optimize the parameters; subsequently, the optimized network is applied on test data efficiently. Meanwhile, with the known dithering code, we embed the trained network into an iterative deblending framework. A test on synthetic data demonstrates that the iterative scheme can obtain appealing deblended results compared with the rank-reduction method. Furthermore, two tests on blended field data (the common-receiver gather [CRG] and common-offset gather) are respectively provided to verify the deblending performance of our proposed method.

THEORY

Supervised learning that requires a large training data set is one type of machine learning. Figure 1 shows the architecture of our designed network, where the left input (the blended patch) and the right target (the unblended patch) are the training data. The designed architecture includes convolutional and deconvolutional networks. The convolutional network containing a convolutional layer, activation layer, and batch normalization (BN) layer is for extracting the local features. The deconvolutional network including deconvolutional layer and activation layer is for constructing the output that is close to the ground truth.

A single convolutional layer can be formulated as

$$\mathbf{Y}_j = \sum_{i}^{I} \mathbf{W}_j * \mathbf{X}_i + \mathbf{b}_j, \quad j = 1, 2, \ldots, J, \tag{1}$$

where $\mathbf{X}_i$ denotes the input in the jth layer, I denotes the total number of inputs, $\mathbf{W}_j$ denotes the convolutional kernel at the jth layer, which can be viewed as a weight matrix, J is the number of convolutional kernels, $\mathbf{Y}_j$ denotes the extracted feature map, and $\mathbf{b}_j$ is the basis that controls the horizontal shift of outputs. Because a single convolutional layer is linear, it cannot describe the nonlinear relationship. Jain et al. (1996) propose the multilayer neural networks that are composed of convolution and activation functions to solve the nonlinear problem. Combined with the activation function, equation 1 can be written as

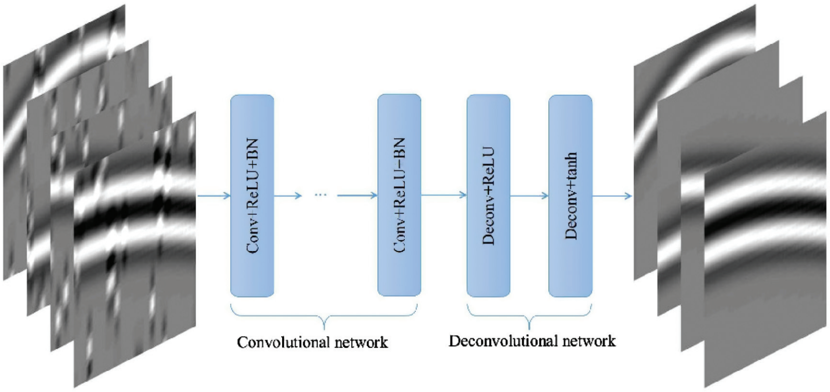

Figure 1. The architecture of a deep neural network that includes convolutional, ReLU activation, BN, deconvolutional, and tanh activation operations.

$$\mathbf{Y}_j = f\left(\sum_i^I \mathbf{W}_j * \mathbf{X}_i + \mathbf{b}_j\right), \quad j = 1, 2, \ldots, J, \tag{2}$$

where f denotes the activation function that rescales the extracted feature map in a nonlinear manner (Oyedotun and Khashman, 2017). The common activation functions include tanh, sigmoid, and rectified linear unit (ReLU). Yu et al. (2019) compare the three activation functions. In the designed architecture, two activation functions (ReLU and tanh) are used herein. The ReLU operation can be formulated as

$$f(x) = \max(0, x). \tag{3}$$

The ReLU activation returns its argument x when it is greater than zero and returns zero otherwise. Although ReLU is relatively simple, it still maintains the nonlinearity or "switch on" characteristic that is similar to a biological neuron. Because the target value varies from -1 to 1, the tanh activation function is applied in the last layer as shown in Figure 1. The tanh function is expressed as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{4}$$

Ioffe and Szegedy (2015) point out that the BN operation enjoys fast training, better performance, and low sensitivity to initialization. We add a BN layer after the activation layer and obtain the following formula:

$$\mathbf{Y}_j = Bf\left(\sum_i^I \mathbf{W}_j * \mathbf{X}_i + \mathbf{b}_j\right),$$
$$j = 1, 2, \ldots, J, \tag{5}$$

where B denotes the BN operation. Equation 5 describes the step "Conv + ReLU + BN" shown in Figure 1. For a layer with d-dimensional input $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k, \ldots, \mathbf{u}_d)$, the BN operation can be expressed as

$$B(\mathbf{u}_k) = \lambda \frac{\mathbf{u}_k - E(\mathbf{u}_k)}{\sqrt{V(\mathbf{u}_k)}} + \beta, \tag{6}$$

where $E(\mathbf{u}_k)$ denotes the mean value of $\mathbf{u}_k$, $V(\mathbf{u}_k)$ denotes the variance of $\mathbf{u}_k$, and λ and β denote the scale and shift parameters, which will be updated at each layer. In the designed architecture, the deconvolutional operation is the transposed convolution that yields the output that is close to the ground truth (Zeiler et al., 2010, 2011; Noh et al., 2015). Tsai et al. (2018) use a deconvolutional operation to recover the first-break arrival time. In this paper, we use the deconvolutional operation to restore the deblended data.

Before updating the network parameters, we need to set the cost function. Given the input $\mathbf{x}$ and target $\mathbf{y}$, the loss function can be characterized as

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \|\mathcal{F}(\mathbf{x}_i, \Theta) - \mathbf{y}_i\|_F^2, \tag{7}$$

where $\mathbf{x}_i \in \mathbf{x}$ denotes the input patch, Θ denotes the network parameters, N denotes the total training pairs, $\|\cdot\|_F$ represents the Frobenius norm, $\mathbf{y}_i \in \mathbf{y}$ denotes the ground truth patch, $\mathcal{F}(\mathbf{x}_i, \Theta)$ denotes

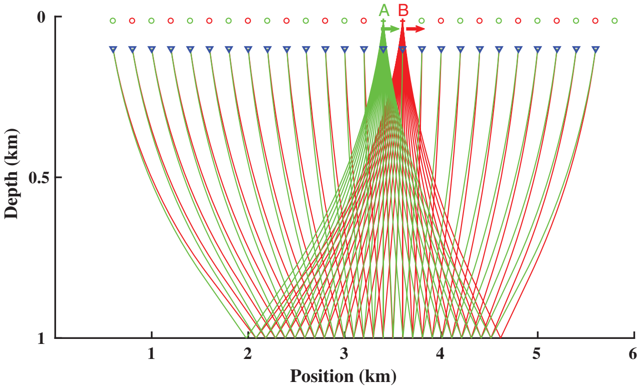

Figure 2. The geometry of the simultaneous source survey, where two sources denoted by crosses (A and B) move in the same direction with a small spacing, circles denote shooting positions and triangles denote receivers.

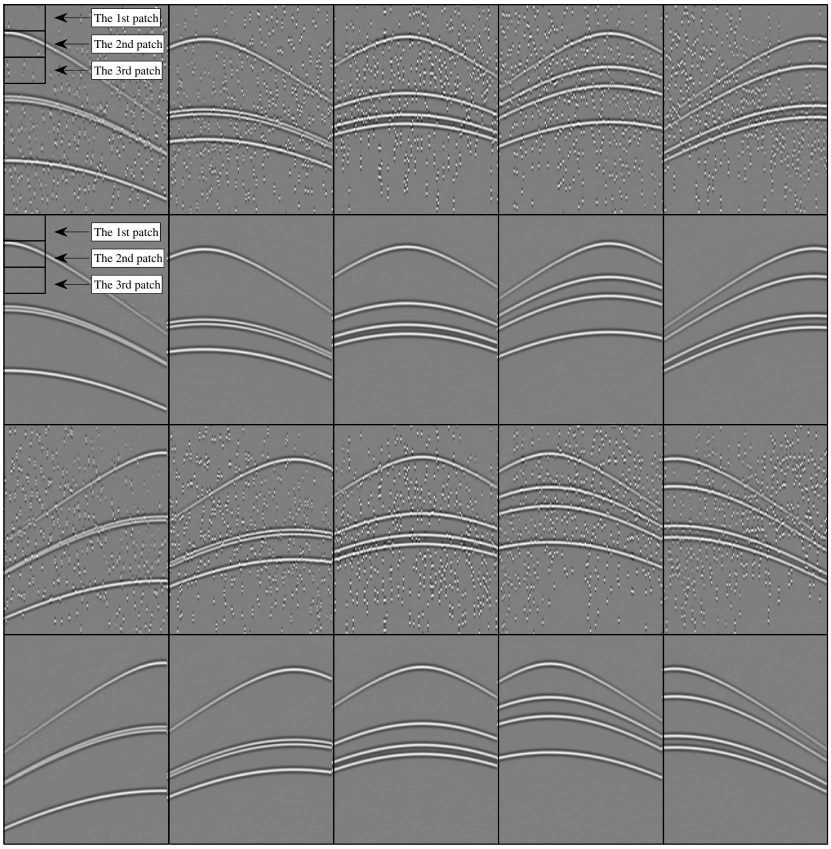

Figure 3. Ten training pairs are randomly selected from 1000 training pairs, where the first and third rows show the blended data (input) and the second and fourth rows display the unblended data (ground truth).

the estimated output $\tilde{\mathbf{y}}_i$, and the process from $\mathbf{x}_i$ to $\tilde{\mathbf{y}}_i$ is shown in Figure 1. Usually, the stochastic gradient descent (SGD) algorithm can provide an effective way to train the network. However, the SGD algorithm requires the hyperparameter to be chosen. In this paper, we choose the ADADELTA algorithm. The ADADELTA algorithm can modify the learning rate automatically and is robust to noisy gradient information. For further detail about ADADELTA, the reader is referred to Zeiler (2012).

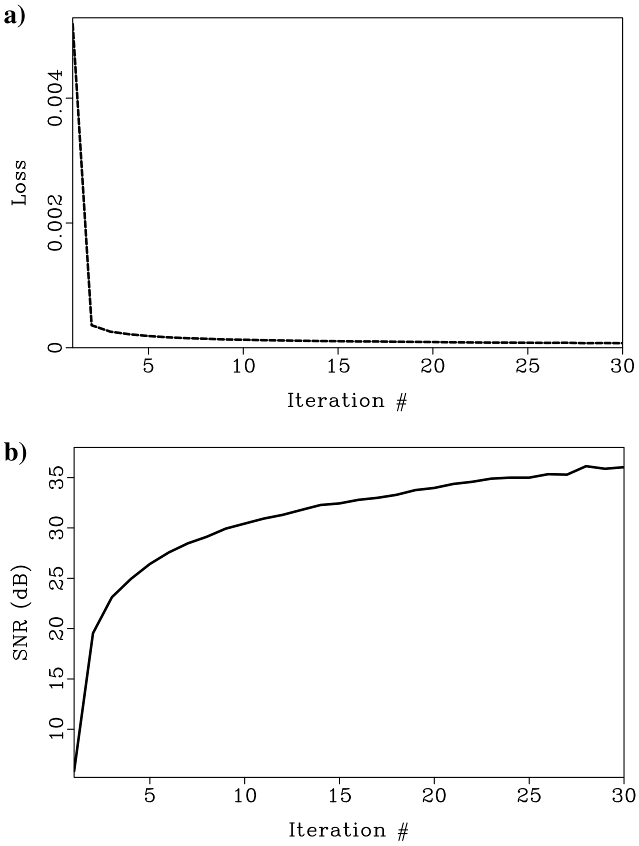

Figure 4. (a) The loss curve varies with the iterations, where it is not easy to detect the change after five iterations. (b) The S/N curve changes with the iterations.

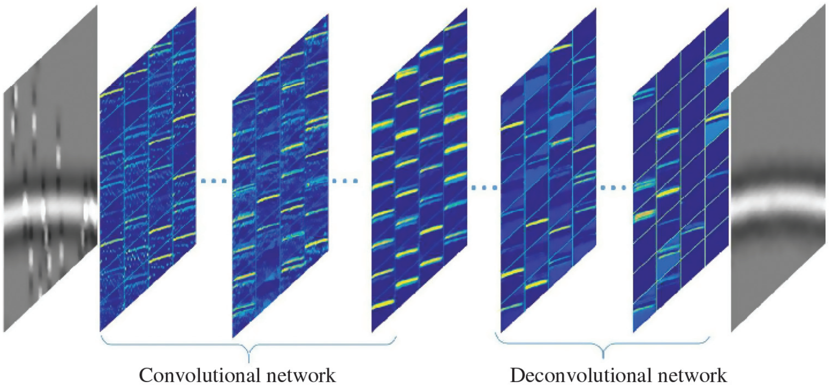

Figure 5. The detailed mapping process separates the blended patch using the trained network, where the convolutional part is to extract the features and the deconvolutional part is to generate the estimated data from the extracted features.

Iterative deblending based on the deep neural network

Taking two-source acquisition as an example, the survey is displayed in Figure 2, where two crosses denote the shooting sources, circles represent the predesigned shooting positions, and triangles stand for receivers. In this survey, the first source (A) fires without dithering time, the second source (B) fires with the dithering time. According to the linearity of the wavefield, the blending formula in the common-receiver domain can be expressed as

$$\mathbf{d}_{\text{ble}} = \mathbf{d}_1 + \Gamma_2 \mathbf{d}_2, \tag{8}$$

where $\mathbf{d}_i (i = 1, 2)$ denotes the individual record that is from all of the corresponding shooting positions for the ith source and Γ_2 is the blending operator for the second source. Applying the inverse of Γ_2 to equation 8, we can obtain pseudodeblended data for the second source

$$\Gamma_2^{-1} \mathbf{d}_{\text{ble}} = \Gamma_2^{-1} \mathbf{d}_1 + \mathbf{d}_2. \tag{9}$$

Combining equations 8 and 9, we get

$$\mathbf{D} = \mathbf{F}\mathbf{m}, \tag{10}$$

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}_{\text{ble}} \\ \Gamma_2^{-1} \mathbf{d}_{\text{ble}} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{I} & \Gamma_2 \\ \Gamma_2^{-1} & \mathbf{I} \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix}. \tag{11}$$

When the trained network Θ is established, the estimation of $\mathbf{m}$ from equation 10 can be achieved using

$$\hat{\mathbf{m}} = \mathcal{F}(\mathbf{D}, \Theta). \tag{12}$$

Using different training data sets, the trained network can accomplish different tasks. In this paper, we use the blended data as input and the unblended data as ground truth to train the network. Therefore, the trained network can separate the blended data. Note that deblending is a little different from denoising. On the one hand, the incoherent interference is the useful signal for other sources, on the other hand, with the dithering time, the incoherent interference can be computed and subtracted iteratively (Mahdad et al., 2011). Denoising aims to reserve useful signal and reject noise; however, deblending is not only to recover the coherent signal but also the incoherent interference. Given the dithering code, the iterative deblending framework is expressed as

$$\mathbf{m}_i = \mathcal{F}(\mathbf{D} - (\mathbf{F} - \mathbf{I})\mathbf{m}_{i-1}, \Theta), \tag{13}$$

where $i \in (1, 2, \ldots, n)$ represents the ith iteration. In the first iteration $(i = 1, \mathbf{m}_0 = \mathbf{0})$, $\mathbf{m}_1 = \mathcal{F}(\mathbf{D}, \Theta)$, which is the same as equation 12. In the next iteration, the input data are $\mathbf{D} - (\mathbf{F} - \mathbf{I})\mathbf{m}_1$, where the incoherent interference has been estimated and subtracted from the blended record to a certain extent. With several iterations, $\mathbf{m}_{i-1}$ converges to the unblended data $\mathbf{m}$. Then, equation 13 can be rewritten as

$$\begin{aligned} \mathbf{m}_i &= \mathcal{F}(\mathbf{D}-(\mathbf{F}-\mathbf{I})\mathbf{m}_{i-1},\Theta)\\ &= \mathcal{F}(\mathbf{Fm}-(\mathbf{F}-\mathbf{I})\mathbf{m}_{i-1},\Theta),\\ &= \mathcal{F}(\mathbf{m}_{i-1},\Theta) \end{aligned} \tag{14}$$

which means that the input is the same as the output in the last iteration.

EXAMPLES

Synthetic data

In the supervised learning literature, the trained network can finish different tasks using different training data sets. To train the network for deblending, we first simulate 1000 training data sets. Each training data set has 256 time samples and 128 traces. Figure 3 displays 10 training pairs randomly selected from the 1000 training pairs, where the first and third rows show the input (blended) data and the second and fourth rows show the ground truth (unblended) data. Before training the network, we split the training data set into patches (32×32 pixels) without overlap as the black rectangles shown in Figure 3. So the total number of patches is $1000\times256/32\times128/32 =$ 32,000. In the training step, the ADADELTA method is implemented to minimize the loss function.

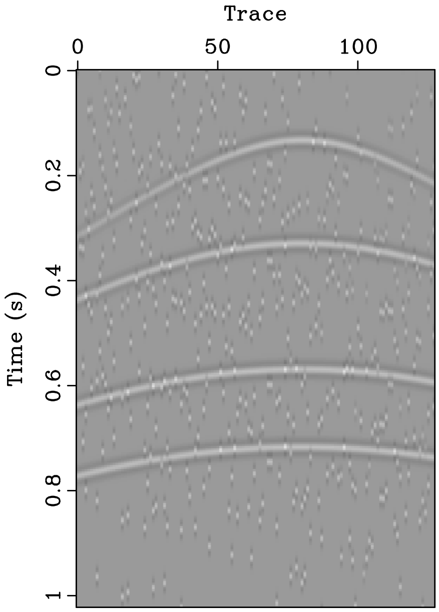

Figure 6. The test blended CRG, where the record of the first source is coherent and that of the second source is incoherent.

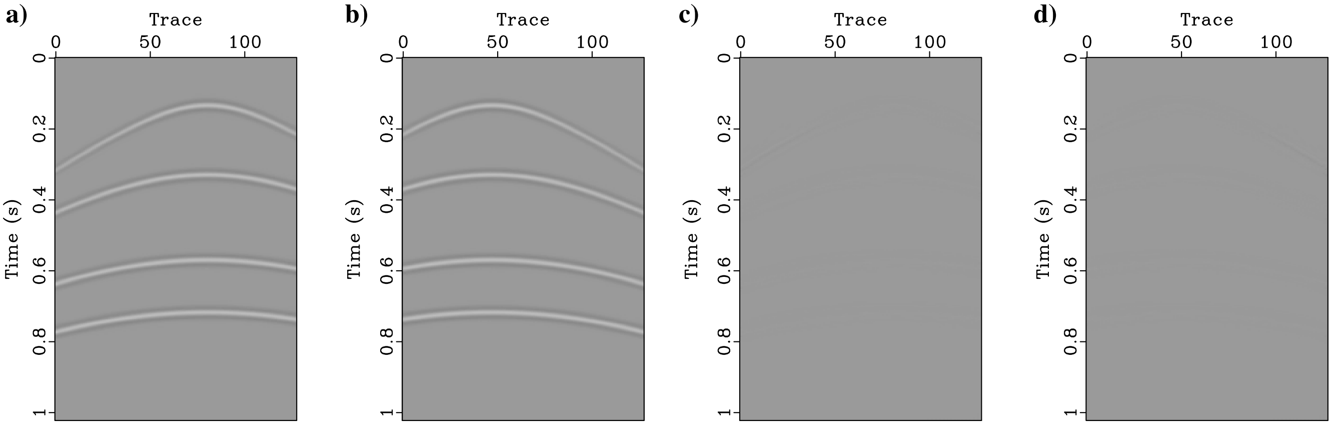

Figure 7. The deblending performance using the proposed method. (a) The deblended result of the first source that corresponds to the coherent components in the test data shown in Figure 6. (b) The deblended result of the second source that corresponds to the incoherent components in the test data. (c and d) The estimation errors.

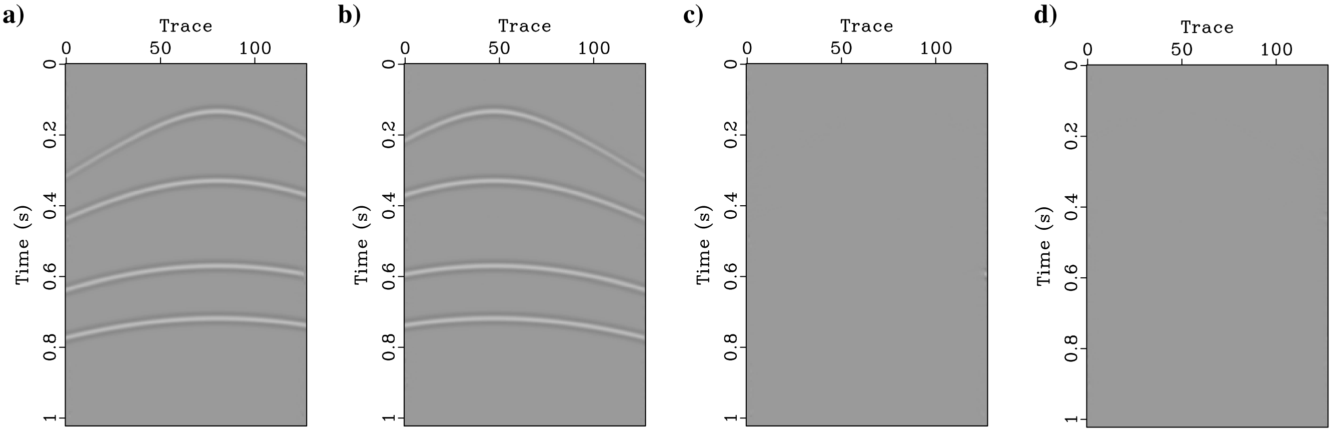

Figure 8. The deblended results corresponding to (a) the first and (b) second source using the rank-reduction method. (c and d) The corresponding errors.

ADADELTA can be viewed as parameter-free because it is insensitive to the hyperparameter selection and it can also avoid the continual decay of the learning rate (Zeiler, 2012). The mean square error as expressed in equation 7 is the common method to monitor the training process. The closer $\tilde{\mathbf{y}}_i$ is to $\mathbf{y}_i$, the smaller $J(\Theta)$ is. Another way to evaluate the deblending performance is the signal-to-noise ratio (S/N), which can be expressed as

$$\mathrm{S/N} = 10\log_{10}\frac{\|\mathbf{y}_i\|^2}{\|\mathbf{y}_i-\tilde{\mathbf{y}}_i\|^2}. \tag{15}$$

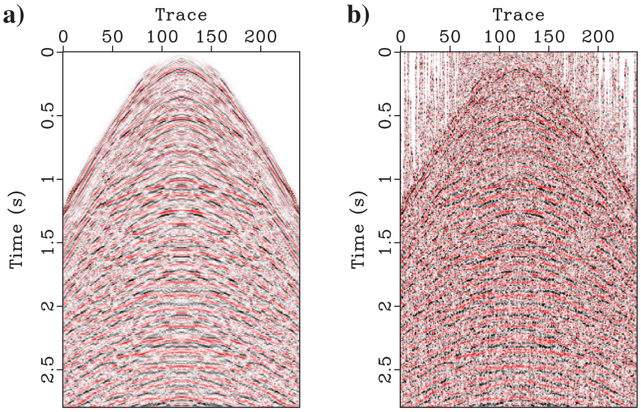

Figure 9. (a) The unblended CRG. (b) The blended CRG where the signal is coherent and the interference is incoherent.

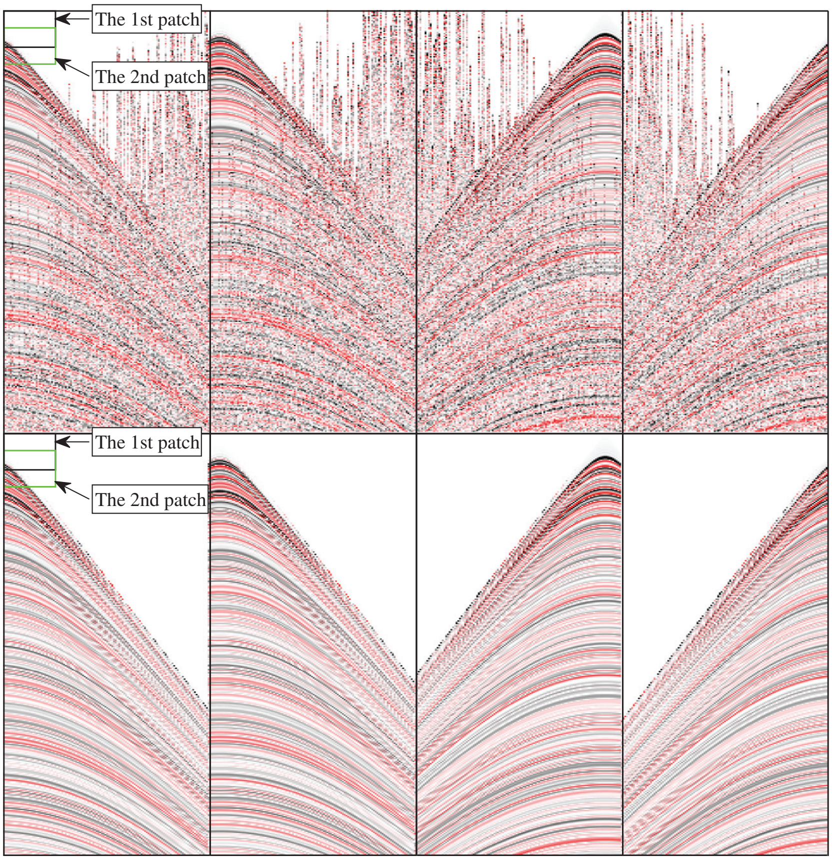

Figure 10. The training data set is used to train the network for separation of the real blended CRG. The top row shows the blended data, and the bottom row shows the ground truth. In this case, we shift five pixels to extract patches as pointed out by the black and green rectangles.

When training the network, two methods are implemented to evaluate the training process. Figure 4 shows the training loss and S/N varying with iterations. In Figure 4a, the loss decreases sharply at the first two iterations. However, after five iterations, it is difficult to observe the obvious loss variation. From the S/N curve shown in Figure 4b, we can observe that the S/N increases quickly at the first two iterations. Although the increase of S/N slows down after five iterations, we can see the improvement clearly. After finishing the training step, the updated network can be implemented to separate the test data. Figure 5 illustrates how to separate the blended data using the trained network. In Figure 5, one blue panel shows the 32 extracted features from the previous layer. With the deepening of the layer, more important features are captured by the convolutional network. The deconvolutional network is to construct the deblended result using the extracted features from the convolutional network. Figure 6 shows the test data, which is the blended CRG. With the dithering code, the record of the second source is incoherent. Before testing, we split the blended data into patches (the size is 32 × 32, and the stride is four) as inputs to the network. After three iterations, we obtain the two deblended results shown in Figure 7a and 7b. The result shown in Figure 7a corresponds to the coherent components in Figure 6, and the result shown in Figure 7b corresponds to the incoherent components in Figure 6. Figure 7c and 7d shows the estimation errors corresponding to the two sources. For a quantitative evaluation of the deblending performance, we calculate the S/N of the deblended result shown in Figure 7a in the three iterations. In the three iterations, the S/N values s are 21.5, 24.8, and 25.3 dB, respectively. The improvement of the S/N demonstrates the effectiveness of our proposed method clearly. All of the tests were carried out on a personal laptop with the Keras platform and the graphics processing unit (GPU) was a Quadro M600M. The size of the training patch is 32, 000 × 32 × 32, and the training process including 30 iterations took 1464.2 s. The size of the test patch is 2850 × 32 × 32, and the test process takes 2.2 s at each iteration. Figure 8a and 8b shows the deblended results by the rank-reduction method, where the fixed size of the window is 20 × 24, the preserved rank is three, and the number of iterations is 50. Figure 8c and 8d displays the estimation errors corresponding to the rank-reduction method. In this example, the rank-reduction method took 71.8 s and the final S/N is 25.7 dB. Through the comparison of computational time and S/N, we can initially conclude that after training the network, our proposed method can obtain comparable performance with high efficiency compared to the rank-reduction method.

Field data examples

In this section, two field data sets are provided to demonstrate the effectiveness of our proposed method. The first field data are a CRG as shown in Figure 9a. Although there is some random noise, we can distinguish the coherent events. To simulate the blended data shown in Figure 9b, we first shift the CRG using the dithering time to generate the incoherent interference, and then we add the incoherent interference to the coherent data. The coherent signal is contaminated by the incoherent interference because they have the same energy. To separate the blended CRG, we simulate four pairs of hyperbolic data sets as shown in Figure 10, where the first row represents the blended CRG and the second row shows the ground truth. Each data set has 500 time samples and 120 traces. To enlarge the training data size, we extract patches (30×30 pixels) by shifting five pixels. The total number of training patches is $4 \times ((500 - 30)/5 + 1) \times ((120 - 30)/5 + 1) = 7220$. The architecture is the same as that previously described in Figure 1. After training, the trained network is embedded into the iterative framework to separate the blended CRG as shown in Figure 9b. Because the blended CRG is generated from one CRG with dithering code, we only show one deblended result to avoid redundancy. Figure 11a displays the deblended result, where the incoherent interference is separated out and the random noise is partially suppressed. From the deblended result, it is easier to distinguish the coherent events. To determine the damage to the signal clearly, we calculate the difference between the deblended result and the unblended data (Figure 9a) and show it in Figure 11b. The damages to the coherent signal are pointed out by arrows. The S/N improves from −0.7 to 5.3 dB. Another way to observe the damage to signal is the local similarity (Fomel, 2007; Zu et al., 2019). The local similarity measurement assumes that signal and noise are orthogonal. If signal is leaked, the local similarity value will be large. Figure 11c represents the local similarity between Figure 11a and 11b. From the local similarity, we can clearly see the damage to the signal indicated by the red values. Figure 11d shows the removed incoherent interference from the blended CRG. From this case, we conclude that our proposed method can well separate the blended field data using the network trained from synthetic data set.

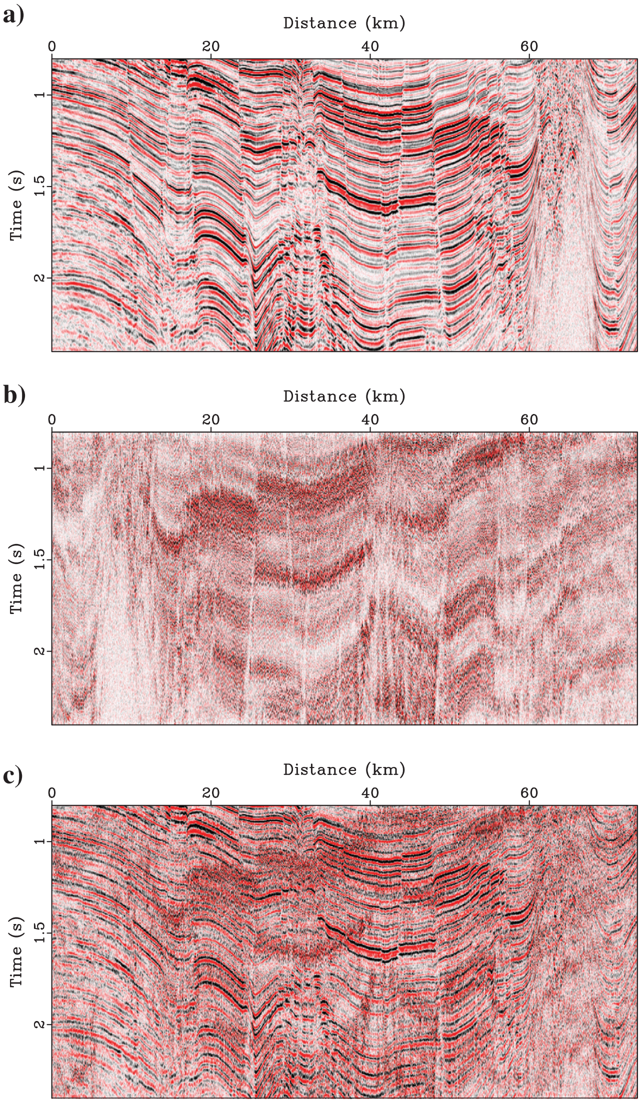

Figure 12. The second training data set: (a) the unblended data (ground truth), (b) the incoherent interference, and (c) the input data that is very noisy because the incoherent interference has the same energy as the unblended data.

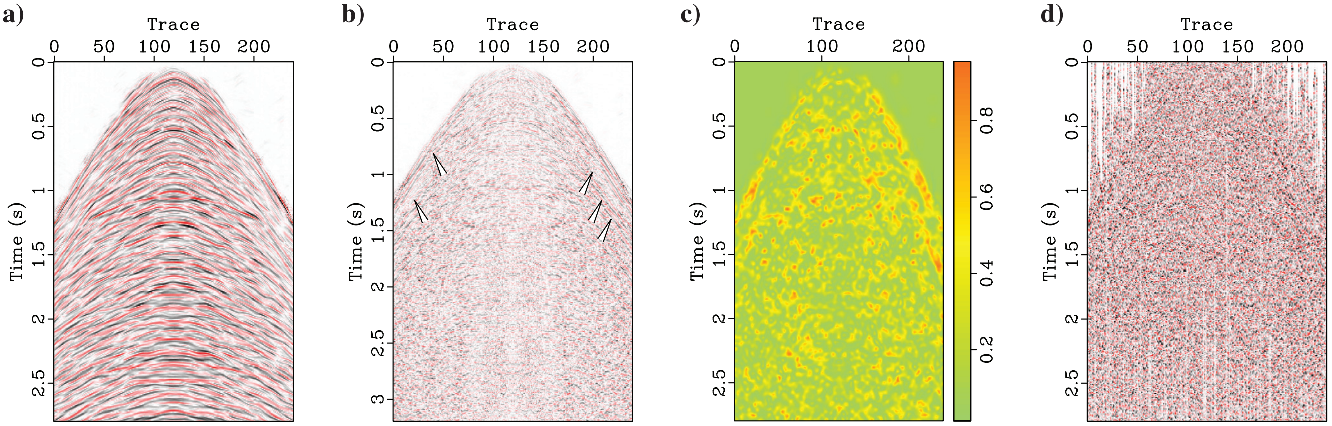

Figure 11. (a) The deblended result, (b) the difference between deblended data and unblended data (Figure 9a), (c) the similarity between (a and b), and (d) the removed components from Figure 9b.

In the second field example, we choose a field data set to train the network. The training pair is shown in Figure 12. Figure 12a shows the ground truth, which is part of a stacked profile from the Gulf of Mexico. The stacked profile contains dipping events, horizontal events, and faults. Before training the network, the input data should be simulated. We flip the stacked profile and shift it using a dithering code to generate the incoherent interference, which is shown in Figure 12b; then we add the incoherent interference to the stacked profile to generate the input data as shown in Figure 12c. Because the incoherent interference is caused by other sources that have the same energy as the coherent events, the input data are very noisy. Note that the simulated input data are different from the stacked data that are processed by the traditional seismic workflow using the blended data. Similar to earlier examples, we adopt the overlap scheme to extract patches. The size of the stacked profile as shown in Figure 12a is 400×2300, the patch is 20×20, and the sliding step is 10. Thus, the total number of training patches is $[(400-20)/10+1] \times [(2300-20)/10+1] = 8931$. After training, we test our proposed method on a common-offset gather from another region. Figure 13a displays the conventional common-offset gather. Figure 13b shows the blended common-offset gather. Compared to the training data set, the test data set is relatively simple. In other words, the training data contain most local features of the test data. Figure 14a and 14b shows the two deblended results. After three iterations by our proposed method, we can obtain two clean common-offset gathers from one noisy blended common-offset gather. In the two deblended results, the weak and strong events are clear and random noise and incoherent interference are separated out. Figure 14c shows the difference between the original data as shown in Figure 13a and the deblended data as shown in Figure 14a. Through Figure 14c, it can be observed that the original data contain much random noise and there is a little damage to the coherent signal between 0.4 and 0.8 s. Figure 14d displays the difference between blended data as shown in Figure 13b and deblended data as shown in Figure 14a, which contains random noise and incoherent interference. To see the performance in detail, a magnification of the rectangles in Figures 13 and 14 is shown in Figure 15. Figure 15a shows the magnified section from the original data as shown in Figure 13a, which contains some random noise. Figure 15b shows the magnified section from blended data as shown in Figure 13b, where the incoherent interference seriously contaminates the coherent events. Figure 15c shows the magnified section of the deblended data shown in Figure 14a, where the random noise and incoherent interference are separated. Figure 15d displays the magnified section from Figure 14c, which is random noise in the original data. Figure 15e shows the magnified section

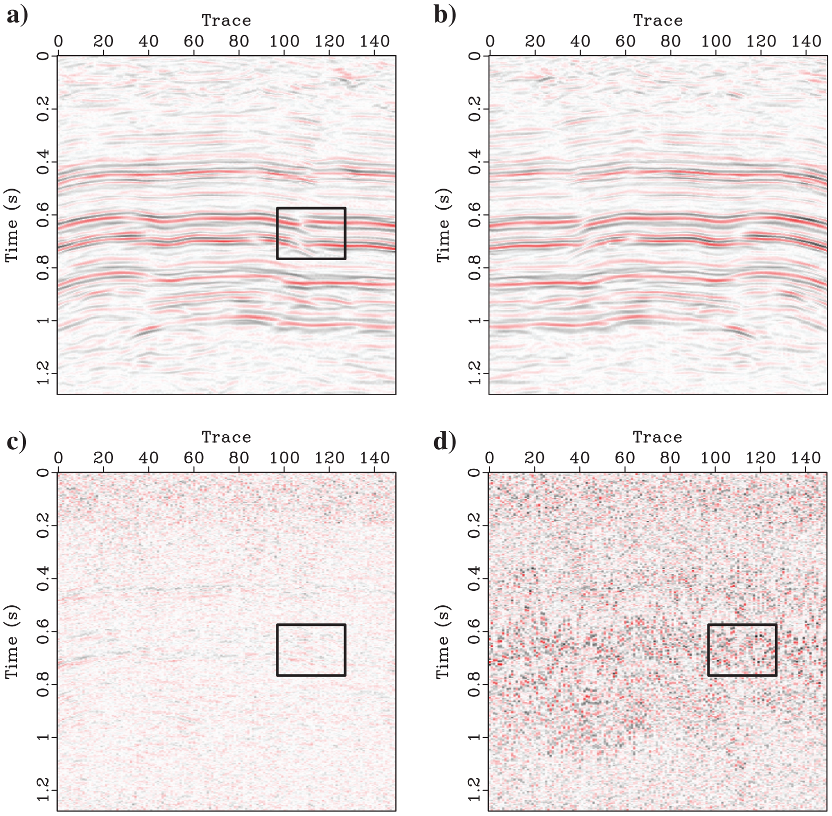

Figure 14. The two deblended results corresponding to (a) the first source and (b) the second source, (c) the difference between Figures 13a and 14a, and (d) the difference between Figures 13b and 14a.

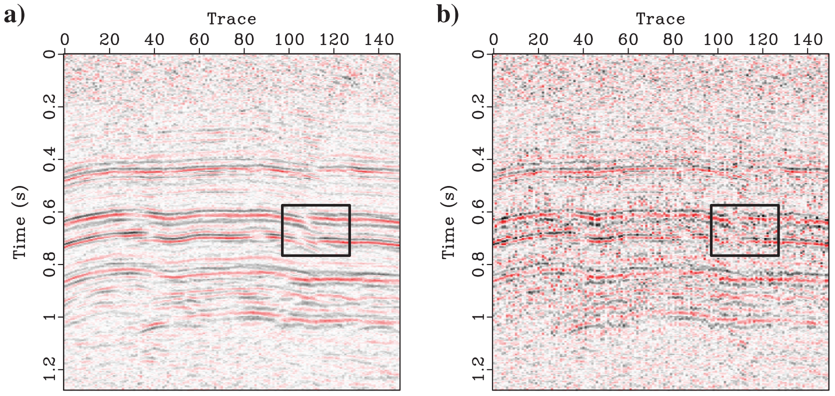

Figure 13. (a) The conventional common-offset gather and (b) the blended common-offset gather.

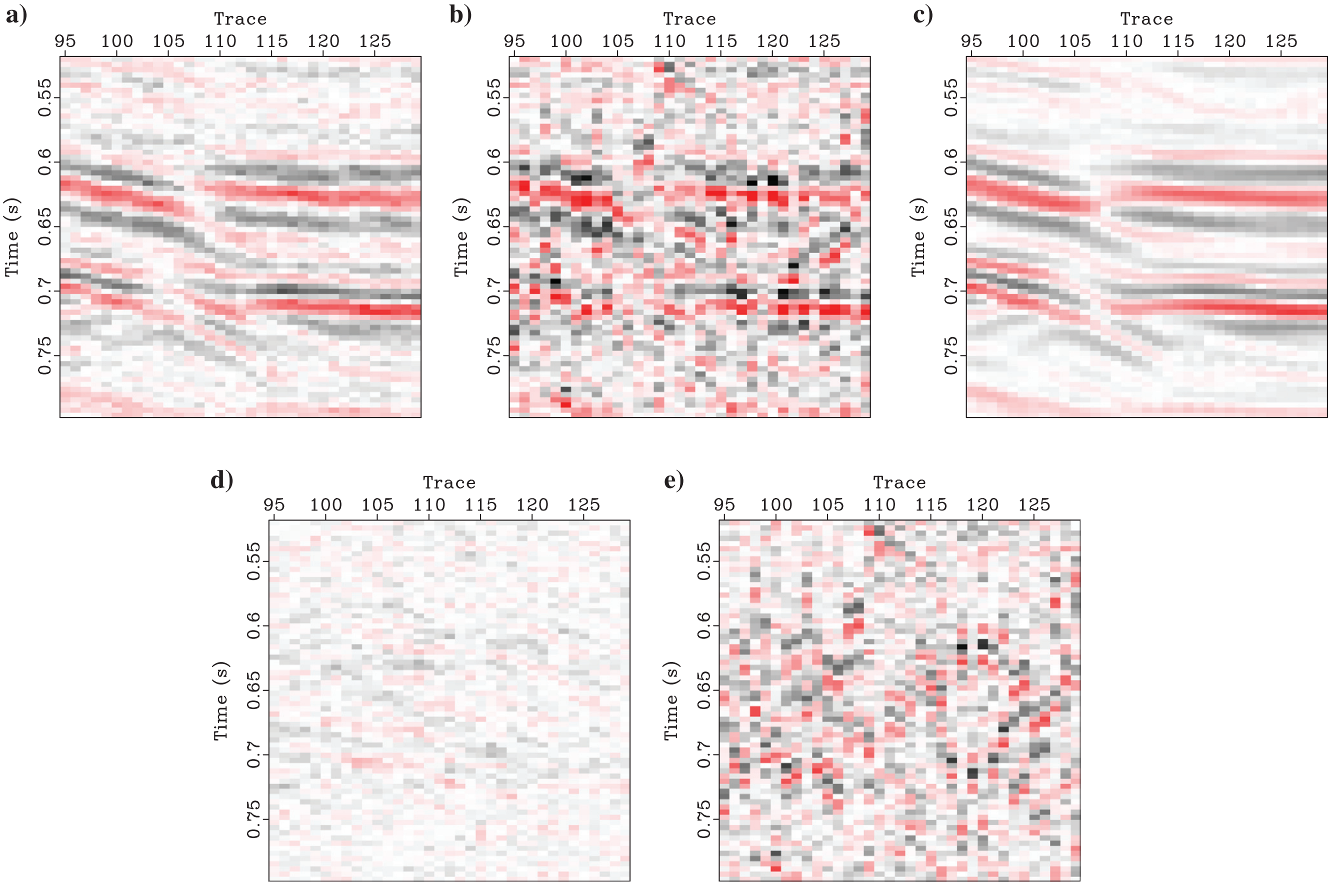

Figure 15. Magnified sections from (a) Figure 13a, (b) Figure 13b, (c) Figure 14a, (d) Figure 14c, and (e) Figure 14d.

from Figure 14d, which represents the removed components by the proposed method. The magnified section confirms the deblending performance of the proposed method.

CONCLUSION

We have proposed a deblending method based on a deep neural network to separate blended data. In the designed network, the convolutional network extracts features from a training data set and then the deconvolutional network constructs the deblended data using the extracted features. Because the trained network can be efficiently applied on new test data, we embed the trained network into the iterative framework to further improve the deblending performance. The deblended results of the synthetic example confirm that the iterative framework combining the trained network can obtain good performance. Furthermore, the examples of deblending of field data prove that the networks trained with synthetic data sets and trained with field data sets can well separate the real blended data. Although our proposed method requires iteration, the increase of computational cost is very low. Thus, our proposed method can be easily applied on large data sets.

ACKNOWLEDGMENTS

The authors appreciate K. Hornman and three anonymous reviewers for their constructive suggestions that improved the quality of the paper significantly. The research was partly supported by the National Natural Science Foundation of China (grant no. 41430323) and National Science and Technology Major Project under grant no. 2016ZX05024-005-007 and 2017ZX05072-002. The work of Y. Chen was supported by the Thousand Youth Talents Plan and the Starting Funds from Zhejiang University.

DATA AND MATERIALS AVAILABILITY

Data associated with this research are available and can be obtained by contacting the corresponding author.

REFERENCES

Abma, R., 2014, Shot scheduling in simultaneous shooting: 84th Annual International Meeting, SEG, Expanded Abstracts, 94–98, doi: 10.1190/segam2014-0812.1.

Abma, R., Q. Zhang, A. Arogunmati, and G. Beaudoin, 2012, An overview of BP's marine independent simultaneous source field trials: 82nd Annual International Meeting, SEG, Expanded Abstracts, doi: 10.1190/segam2012-1404.1.

Abma, R. L., T. Manning, M. Tanis, J. Yu, and M. Foster, 2010, High quality separation of simultaneous sources by sparse inversion: 72nd Annual International Conference and Exhibition, EAGE, Extended Abstracts, B003, doi: 10.3997/2214-4609.201400611.

Akerberg, P., G. Hampson, J. Rickett, H. Martin, and J. Cole, 2008, Simultaneous source separation by sparse Radon transform: 78th Annual International Meeting, SEG, Expanded Abstracts, 2801–2805, doi: 10.1190/1.3063927.

Alexander, G., R. Abma, R. Clarke, and S. L. Dart, 2013, Processing results of simultaneous source surveys compared to conventional surveys: 83rd

Annual International Meeting, SEG, Expanded Abstracts, 104–108, doi: 10.1190/segam2013-0641.1.

Andersson, F., J. O. Robertsson, D.-J. van Manen, J. Wittsten, K. Eggenberger, and L. Amundsen, 2017, Flawless diamond separation in simultaneous source acquisition by seismic apparition: Geophysical Journal International, **209**, 1735–1739, doi: 10.1093/gji/ggx124.

Baardman, R., and C. Tsingas, 2019, Classification and suppression of blending noise using convolutional neural networks: Annual Technical Conference and Exhibition, SPE, Extended Abstracts, doi: 10.2118/194731-MS.

Baardman, R., and R. van Borselen, 2013, A simulated simultaneous source experiment in shallow waters and the impact of randomization schemes: 83rd Annual International Meeting, SEG, Expanded Abstracts, 4382–4386, doi: 10.1190/segam2013-1199.1.

Bagaini, C., M. Daly, and I. Moore, 2012, The acquisition and processing of dithered slip-sweep vibroseis data: Geophysical Prospecting, **60**, 618–639, doi: 10.1111/j.1365-2478.2012.01085.x.

Bai, M., J. Wu, J. Xie, and D. Zhang, 2018, Least-squares reverse time migration of blended data with low-rank constraint along structural direction: Journal of Seismic Exploration, **27**, 29–48.

Beasley, C. J., 2008, A new look at marine simultaneous sources: The Leading Edge, **27**, 914–917, doi: 10.1190/1.2954033.

Beasley, C. J., B. Dragoset, and A. Salama, 2012, A 3D simultaneous source field test processed using alternating projections: A new active separation method: Geophysical Prospecting, **60**, 591–601, doi: 10.1111/j.1365-2478.2011.01038.x.

Berkhout, A., 2008, Changing the mindset in seismic data acquisition: The Leading Edge, **27**, 924–938, doi: 10.1190/1.2954035.

Berkhout, A. J., D. J. Verschuur, and G. Blacquière, 2012, Illumination properties and imaging promises of blended, multiple-scattering seismic data: A tutorial: Geophysical Prospecting, **60**, 713–732, doi: 10.1111/j.1365-2478.2012.01081.x.

Berkhout, G., and G. Blacquière, 2013, Effect of noise in blending and deblending: Geophysics, **78**, no. 5, A35–A38, doi: 10.1190/geo2013-0103.1.

Chen, Y., S. Fomel, and J. Hu, 2014, Iterative deblending of simultaneous-source seismic data using seislet-domain shaping regularization: Geophysics, **79**, no. 5, V179–V189, doi: 10.1190/geo2013-0449.1.

Chen, Y., G. Zhang, M. Bai, S. Zu, Z. Guan, and M. Zhang, 2019, Automatic waveform classification and arrival picking based on convolutional neural network: Earth and Space Science, **6**, 1244–1261, doi: 10.1029/2018EA000466.

Chen, Y., S. Zu, Y. Wang, and X. Chen, 2018, Deblending of simultaneous source data using a structure-oriented space-varying median filter: Geophysical Journal International, **216**, 1214–1232, doi: 10.1093/gji/ggy487.

Cheng, J., and M. Sacchi, 2015, Separation and reconstruction of simultaneous source data via iterative rank reduction: Geophysics, **80**, no. 4, V57–V66, doi: 10.1190/geo2014-0385.1.

Cheng, J., and M. D. Sacchi, 2016, Fast dual-domain reduced-rank algorithm for 3D deblending via randomized QR decomposition: Geophysics, **81**, no. 1, V89–V101, doi: 10.1190/geo2015-0292.1.

Doulgeris, P., K. Bube, G. Hampson, and G. Blacquière, 2012, Convergence analysis of a coherency-constrained inversion for the separation of blended data: Geophysical Prospecting, **60**, 769–781, doi: 10.1111/j.1365-2478.2012.01088.x.

Fomel, S., 2007, Shaping regularization in geophysical-estimation problems: Geophysics, **72**, no. 2, R29–R36, doi: 10.1190/1.2433716.

Halliday, D. F., and I. Moore, 2018, A comparison of random and periodic marine simultaneous-source encoding: The Leading Edge, **37**, 471a1–471a11, doi: 10.1190/tle37060471a1.1.

Huo, S., Y. Luo, and P. G. Kelamis, 2012, Simultaneous sources separation via multidirectional vector-median filtering: Geophysics, **77**, no. 4, V123–V131, doi: 10.1190/geo2011-0254.1.

Ibrahim, A., and M. D. Sacchi, 2013, Simultaneous source separation using a robust Radon transform: Geophysics, **79**, no. 1, V1–V11, doi: 10.1190/geo2013-0168.1.

Ioffe, S., and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift: arXiv:1502.03167.

Jain, A. K., J. Mao, and K. Mohiuddin, 1996, Artificial neural networks: A tutorial: Computer, **29**, 31–44, doi: 10.1109/2.485891.

Kumar, R., H. Wason, and F. J. Herrmann, 2015, Source separation for simultaneous towed-streamer marine acquisition — A compressed sensing approach: Geophysics, **80**, no. 6, WD73–WD88, doi: 10.1190/geo2015-0108.1.

Li, C., C. C. Mosher, L. C. Morley, Y. Ji, and J. D. Brewer, 2013, Joint source deblending and reconstruction for seismic data: 83rd Annual International Meeting, SEG, Expanded Abstracts, 82–87, doi: 10.1190/segam2013-0411.1.

Li, S., Y. Ren, Y. Chen, S. Yang, Y. Wang, and P. Jiang, 2019, Deep learning inversion of seismic data: arXiv:1901.07733.

Lin, T. T., and F. J. Herrmann, 2009, Designing simultaneous acquisitions with compressive sensing: 71st Annual International Conference and Exhibition, EAGE, Extended Abstracts, S006, doi: 10.3997/2214-4609.201400276.

Mahdad, A., P. Doulgeris, and G. Blacquière, 2011, Separation of blended data by iterative estimation and subtraction of blending interference noise: Geophysics, **76**, no. 3, Q9–Q17, doi: 10.1190/1.3556597.

Mahdad, A., P. Doulgeris, and G. Blacquière, 2012, Iterative method for the separation of blended seismic data: Discussion on the algorithmic aspects: Geophysical Prospecting, **60**, 782–801, doi: 10.1111/j.1365-2478.2012.01084.x.

Nakayama, S., G. Blacquière, T. Ishiyama, and S. Ishikawa, 2018, Blended-acquisition design with irregular geometries for wide-azimuth sampling: 88th Annual International Meeting, SEG, Expanded Abstracts, 126–130, doi: 10.1190/segam2018-2979763.1.

Noh, H., S. Hong, and B. Han, 2015, Learning deconvolution network for semantic segmentation: Proceedings of the IEEE International Conference on Computer Vision, 1520–1528, doi: 10.1109/ICCV.2015.178.

Oyedotun, O., and A. Khashman, 2017, Iris nevus diagnosis: Convolutional neural network and deep belief network: Turkish Journal of Electrical Engineering & Computer Sciences, **25**, 1106–1115, doi: 10.3906/elk-1507-190.

Pham, N., S. Fomel, and D. Dunlap, 2018, Automatic channel detection using deep learning: 88th Annual International Meeting, SEG, Expanded Abstracts, 2026–2030, doi: 10.1190/INT-2018-0202.1.

Richardson, A., and C. Feller, 2019, Seismic data denoising and deblending using deep learning: arXiv:1907.014971v1.

Shi, Y., X. Wu, and S. Fomel, 2018, Automatic salt-body classification using a deep convolutional neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 1971–1975, doi: 10.1190/segam2018-2997304.1.

Tsai, K. C., W. Hu, X. Wu, J. Chen, and Z. Han, 2018, First-break automatic picking with deep semisupervised learning neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 2181–2185, doi: 10.1190/segam2018-2998106.1.

van Borselen, R., R. Baardman, T. Martin, B. Goswami, and E. Fromyr, 2012, An inversion approach to separating sources in marine simultaneous shooting acquisition — Application to a Gulf of Mexico data set: Geophysical Prospecting, **60**, 640–647, doi: 10.1111/j.1365-2478.2012.01076.x.

Verschuur, D., and A. Berkhout, 2011, Seismic migration of blended shot records with surface-related multiple scattering: Geophysics, **76**, no. 1, A7–A13, doi: 10.1190/1.3521658.

Wang, B., N. Zhang, W. Lu, and J. Wang, 2018, Deep-learning-based seismic data interpolation: A preliminary result: Geophysics, **84**, no. 1, V11–V20, doi: 10.1190/geo2017-0495.1.

Wapenaar, K., J. van der Neut, and J. Thorbecke, 2012, Deblending by direct inversion: Geophysics, **77**, no. 3, A9–A12, doi: 10.1190/geo2011-0497.1.

Wu, S., G. Blacquiére, and G.-J. A. van Groenestijn, 2018, Shot repetition: An alternative seismic blending code in marine acquisition: Geophysics, **83**, no. 6, P29–P37, doi: 10.1190/geo2017-0649.1.

Xue, Y., F. Chang, D. Zhang, and Y. Chen, 2016a, Simultaneous sources separation via an iterative rank-increasing method: IEEE Geoscience and Remote Sensing Letters, **13**, 1915–1919, doi: 10.1109/LGRS.2016.2617338.

Xue, Z., Y. Chen, S. Fomel, and J. Sun, 2016b, Seismic imaging of incomplete data and simultaneous-source data using least-squares reverse time migration with shaping regularization: Geophysics, **81**, no. 1, S11–S20, doi: 10.1190/geo2014-0524.1.

Yang, F., and J. Ma, 2019, Deep-learning inversion: A next-generation seismic velocity model building method: Geophysics, **84**, no. 4, R583–R599, doi: 10.1190/geo2018-0249.1.

Yu, S., J. Ma, and W. Wang, 2019, Deep learning for denoising: Geophysics, **84**, no. 6, V333–V350, doi: 10.1190/geo2018-0668.1.

Zeiler, M. D., 2012, Adadelta: An adaptive learning rate method: arXiv:1212.5701.

Zeiler, M. D., D. Krishnan, G. W. Taylor, and R. Fergus, 2010, Deconvolutional networks: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2528–2535, doi: 10.1109/CVPR.2010.5539957.

Zeiler, M. D., G. W. Taylor, and R. Fergus, 2011, Adaptive deconvolutional networks for mid and high level feature learning: International Conference on Computer Vision, **1**, 6, doi: 10.1109/ICCV.2011.6126474.

Zhang, G., Z. Wang, and Y. Chen, 2018a, Deep learning for seismic lithology prediction: Geophysical Journal International, **215**, 1368–1387, doi: 10.1093/gji/ggy344.

Zhang, Q., W. Mao, and Y. Chen, 2018c, Attenuating crosstalk noise of simultaneous-source least-squares reverse time migration with GPU-based excitation amplitude imaging condition: IEEE Transactions on Geoscience and Remote Sensing, **57**, 587–597, doi: 10.1109/TGRS.2018.2858850.

Zhang, Q., W. Mao, H. Zhou, H. Zhang, and Y. Chen, 2018b, Hybrid-domain simultaneous-source full waveform inversion without crosstalk noise: Geophysical Journal International, **215**, 1659–1681, doi: 10.1093/gji/ggy366.

Zhou, Y., J. Gao, W. Chen, and P. Frossard, 2016, Seismic simultaneous source separation via patchwise sparse representation: IEEE Transactions on Geoscience and Remote Sensing, **54**, 5271–5284, doi: 10.1109/TGRS.2016.2559514.

Zu, S., H. Zhou, Y. Chen, S. Qu, X. Zou, H. Chen, and R. Liu, 2016, A periodically varying code for improving deblending of simultaneous sources in marine acquisition: Geophysics, **81**, no. 3, V213–V225, doi: 10.1190/geo2015-0447.1.

Zu, S., H. Zhou, R. Wu, W. Mao, and Y. Chen, 2019, Hybrid-sparsity constrained dictionary learning for iterative deblending of extremely noisy simultaneous-source data: IEEE Transactions on Geoscience and Remote Sensing, **57**, 2249–2262, doi: 10.1109/TGRS.36.