Speech Separation based on Deep Belief Network

Wu Haijia [a], Zhang Xiongwei [b], Zhang Liangliang [c] and Zou Xia [d]

College of Command Information and Systems, PLA University of Science and Technology, Nanjing, China, 210007

[a] wu_haijia@163.com, [b] xwzhang9898@163.com,
[c] vermouthlove@hotmail.com, [d] zlc1997@163.com

Keywords: speech separation, deep learning, deep belief network, restricted Boltzmann machine, autoencoder

Abstract. Thanks to its hierarchical and generative nature, Deep Belief Network (DBN) is effective to feature representation and extraction in signal processing. In this paper, DBN is investigated and implemented to monaural speech separation. Firstly, two separate DBNs are trained to extract features from mixed noisy signals and target clean speech respectively. Subsequently, the two types of extracted features are associated together by training a BP neural network to obtain a mapping from the features of mixed signals to the features of target speech. Finally, by performing DBN and the above mapping neural network, target speech can be estimated from the input mixed signals. Experiments are conducted on different kinds of mixed signals including female/male speech mixtures, human-speech/Gaussian-noise audio mixtures, and human-speech/music audio mixtures. The PESQ scores of the extracted speech are 3.32, 2.59, and 3.42 respectively, which illustrates that the model performs well on speech separation tasks, especially on the mixed signals where the inference signals have obvious spectral structures.

Introduction

Recorded speech signals are often contaminated by noise, multi-speakers' interference, background music, and so on. Therefore, the first step in information processing of speech signal is usually speech separation from the contaminated inputs in order to acquire a good front-end model. Up to now, many approaches have been proposed for speech separation, including independent component analysis (ICA) [1], non-negative matrix factorization (NMF) [2], subspace decomposition algorithm [3], and tools in computational auditory scene analysis [4] etc. Those approaches can only obtain good performance on speech separation when the target speech and interference signals satisfy certain constraints. However, the constraints are usually difficult to be satisfied in naturally recorded signals, which severely limit the application of these approaches in practice.

A recently proposed machine learning tool, Deep Neural Networks (DNNs), has been used to speech separation, which has demonstrated positive performance in [5][6][7]. DNN learns multiple levels of representation, where higher level representation leads to more abstract features. The high-level abstraction can hopefully make it easier to separate signal mixtures from each other by exploiting the various interpretable factors in the data [8]. However, a prominent problem in training such a deep model is that the cost function can easily get stuck in poor local optima due to the sever non-linearity of the problem. Besides, unlike the supervised learning problem in traditional DNN training, due to lacking of manually made labels, DNN for speech separation has to be trained in an unsupervised way or a semi-supervised fashion. Hinton et al. proposed a greedy layer-wise algorithm for DNN unsupervised learning, which is called Deep Belief Networks (DBNs) [9]. The algorithm can alleviate the local-optimum problem and is able to produce relatively accurate layer-wise representations.

Given the above facts, in this paper, we take DBN as our main model for extracting speech features from speech signals. With its generative nature, the DBN can reconstruct the input data based on the values of the outputs [10][11]. Hence, if we map the features from mixed signals to features of target

speech, the reconstructed outputs from DBN will be the target speech. In this paper, we use a BP neural network for feature mapping.

The remaining part of the paper is organized as follows: In section 2, we introduce DBN briefly. In section 3, we describe the speech separation approach based on DBN. In section 4, we present our experimental results. And finally, we conclude our work in section 5.

Deep Belief Network

DBNs are probabilistic generative models that are composed of multiple layers of Restricted Boltzmann Machines (RBMs) [12]. A graphical depiction of RBM and DBN is shown in Figure 1. DBNs are learned layer-by-layer by treating the outputs of the RBM of a low-layer as the inputs for the RBM of the consecutive high layer.

An RBM is a special type of Markov random field that has one layer of (typically Bernoulli) stochastic hidden units and one layer of (typically Bernoulli or Gaussian) stochastic visible or observable units. RBMs can be represented as bipartite graphs, where all visible units are connected to all hidden units, and there are no visible-visible or hidden-hidden connections.

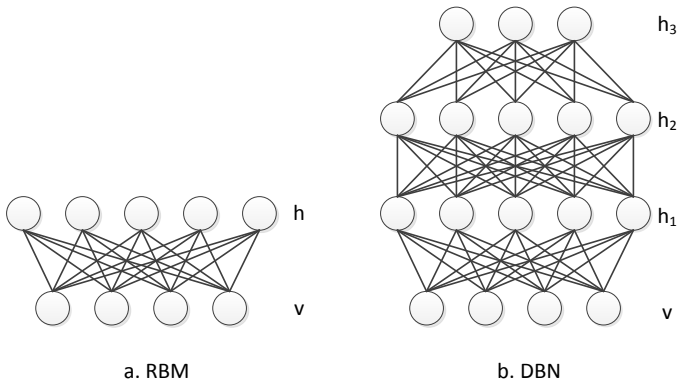

Fig.1 An illustration of RBM and DBN

In an RBM, the joint distribution $p(v,h;\theta)$ over the visible units v and hidden units h, is defined in terms of an energy function $E(v,h;\theta)$ of

$$p(v,h;\theta)=\frac{exp(-E(v,h;\theta))}{Z}$$

where θ refers to the model parameters and $Z=\sum_{v}\sum_{h}exp(-E(v,h;\theta))$ is a normalization factor or partition function.

For a Bernoulli(visible)-Bernoulli(hidden) RBM, the energy function is defined as

$$E(v,h;\theta)=-\sum_{i=1}^{I}\sum_{j=1}^{J}w_{ij}v_ih_j-\sum_{i=1}^{I}b_iv_i-\sum_{j=1}^{J}a_jh_j$$

where w_{ij} represents the symmetric interaction term between visible unit v_i and hidden unit h_j; b_i and a_j are the bias terms; and I and J are the numbers of visible and hidden units. The conditional probabilities can be efficiently calculated as

$$p(h_j=1\,|\,v;\theta)=\sigma\left(\sum_{i=1}^{I}w_{ij}v_i+a_j\right)$$

$$p(v_i=1\,|\,h;\theta)=\sigma\left(\sum_{j=1}^{J}w_{ij}h_j+b_i\right)$$

where $\sigma(x)=1/(1+e^{-x})$.

Similarly, for a Gaussian(visible)-Bernoulli(hidden) RBM, the energy is

$$E(v,h;\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} v_i h_j - \frac{1}{2}\sum_{i=1}^{I}(v_i - b_i)^2 - \sum_{j=1}^{J} a_j h_j$$

The corresponding conditional probabilities become

$$p(h_j = 1 \mid v;\theta) = \sigma\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right)$$

$$p(v_i \mid h;\theta) = N\left(\sum_{j=1}^{J} w_{ij} h_j + b_i, 1\right)$$

where v_i takes real values and follows a Gaussian distribution with mean $\sum_{j=1}^{J} w_{ij} h_j + b_i$ and variance one. It is easy to see from the above definitions, that Gaussian-Bernoulli RBMs can be used to convert real-valued stochastic variables to binary stochastic variables, which can be further processed using the Bernoulli-Bernoulli RBMs.

Taking the gradient of the log likelihood $\log p(v;\theta)$ we can derive the update rule for the RBM weights as

$$\Delta w_{ij} = E_{data}(v_i h_j) - E_{model}(v_i h_j)$$

where $E_{data}(v_i h_j)$ is the expectation observed in the training set and $E_{model}(v_i h_j)$ is the expectation under the distribution defined by the model. Unfortunately, $E_{model}(v_i h_j)$ is intractable to compute. So, the contrastive divergence (CD) approximation to the gradient is used where $E_{model}(v_i h_j)$ is replaced by running the Gibbs sampling initialized at the data for one full step. The steps in approximating $E_{model}(v_i h_j)$ are as follows:

Initialize v_0 at training data
Sample $h_0 \sim p(h \mid v_0)$
Sample $v_1 \sim p(v \mid h_0)$
Sample $h_1 \sim p(h \mid v_1)$

Then (v_1, h_1) is a sample from the model, as a very rough estimate of $E_{model}(v_i h_j)$. Use of (v_1, h_1) to approximate $E_{model}(v_i h_j)$ gives rise to the algorithm of CD-1.

Stacking a number of RBMs learned layer by layer in a bottom-up way will produce a DBN. The stacking procedure is as follows. After learning a Gaussian-Bernoulli RBM, we treat the activation probabilities of its hidden units as the data for training the Bernoulli-Bernoulli RBM one layer up. The activation probabilities of the second-layer Bernoulli-Bernoulli RBM are then used as the visible data input for the third-layer Bernoulli-Bernoulli RBM. And so on. Some theoretical justification of this efficient layer-by-layer greedy learning strategy is given in [9], where it is shown that the stacking procedure above improves a variational lower bound on the likelihood of the training data under the composition model. That is, the greedy procedure above achieves approximate maximum likelihood learning. And this learning procedure is unsupervised and requires no class label.

After performing the layer-wise learning, we have obtained a good initialization for the parameters (i.e. hidden weights and biases of all the layers) of the DBN. Then the back-propagation (BP) algorithm can be used to fine-tune the network weights in the same way as for the standard feed-forward neural network.

Speech Separation based on DBN

There are totally six steps to construct a speech separation model based on DBN, as shown in Figure 2.

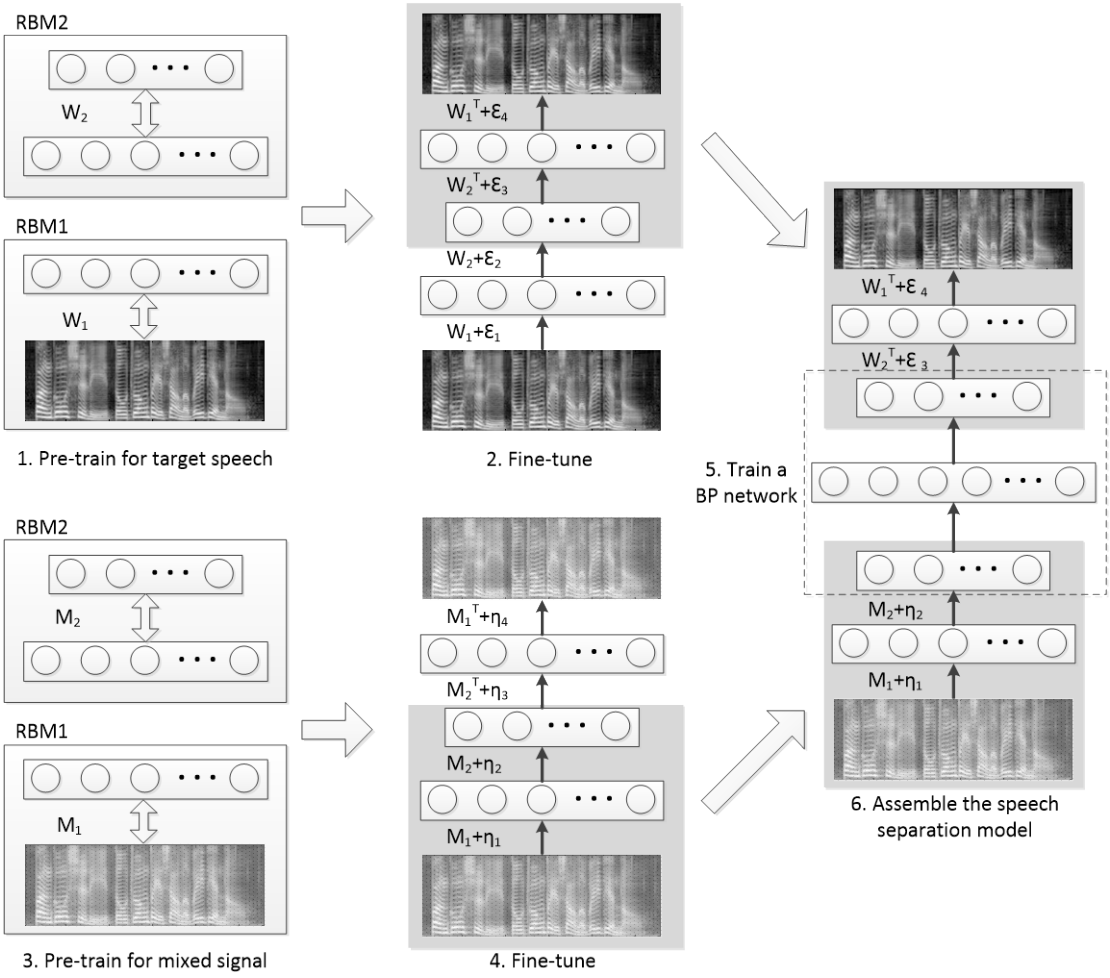

Fig.2 Procedure for speech separation based on DBN

In step 1 and step 3, we pre-train two DBNs separately for target clean speech and mixed signal in the training dataset. The two DBNs have the same structure which is consists of two RBMs. The first layer is a Gaussian-Bernoulli RBM. As described in [11], DBN behaves in a fairly binary way for reasonably long windows (9 and 13), but not for short windows. Following this instruction, for both the target clean speech (in step 1) or the mixed signal (in step 3), we take 9-frame speech spectra patches with normalized log power as the input vectors (visible layers) of the RBMs. Every frame contains 129 frequency bins which are computed from 256 point FFT. So every 9-frame patch is a 1161-by-1 vector. The number of hidden units of the output layer of RBM is set 3000. The connection weights and biases can be learned efficiently using the CD approximation to the log likelihood gradient. After learning the first layer RBM, we treat the activation probabilities of its hidden units as the inputs for training the second layer Bernoulli-Bernoulli RBM [9] which has 3000 binary input/visible variables and 312 binary output/hidden variables. Now, we get two DBNs with three layers: 1161-3000-312. And this has been tested as a low-distortion speech coding architecture in [11].

In step 2 and step 4, we fine-tune the DBNs constructed in the last steps. Firstly, we “unroll” the two DBNs by using their weight matrices to create two five-layer deep networks separately whose lower layers use the matrices to encode the input and whose upper layers use the matrices in reverse order to decode the input. The two deep networks are then fine-tuned using back-propagation of error-derivatives to make their outputs as close as possible to their inputs. We call the two deep networks target-speech-autoencoder and mixed-signal-autoencoder separately. Details of the process, including the number of training passes(epochs) in pre-training and fine-tuning, the division of the training set into mini-batches, the learning rate, the weight decay, and the threshold used to force binary codes, etc., are important to obtain good reconstruction results.

In step 5, we train a three-layer BP neural network such that it learns to map features from mixed signal to features from target speech. As shown in the dashed box of Figure 2, the input of the BP neural network is the activations from the middle layer of the mixed-signal-autoencoder, and the supervision of the input is the activations from the middle layer of the target-speech-autoencoder. The number of hidden variables in the BP neural network has some impacts on the mapping performance, as will be tested in section 4.

In step 6, we assemble the speech separation model. As shown in Figure 2, the lower layers of the model (in the shadowed box) are duplicated from the lower half part of the mixed-signal-autoencoder, and the upper layers of the model (in the shadowed box) are duplicated from the upper half part of the target-speech-autoencoder. The two parts are connected by the BP neural network trained in step 5.

Finally, the speech separation model based on DBN is done. By feeding the 9-frame spectral patches of the contaminated signal into the model, we can estimate the 9-frame spectral patches of the of target clean speech. Then, we use the overlap-and-add technique to reconstruct the full-length speech spectrogram. At the end, the time-domain speech signal can be reconstructed using the real-time spectrum inversion algorithm proposed in [13].

Experiments and Results

We have examined properties of the speech separation model discussed above, and conducted experiments on different types of mixed signals including female/male speech mixtures, human-speech/Gaussian-noise audio mixtures, and human-speech/music audio mixtures.

Data utilized in our experiments was selected from TIMIT database, where the training part is consist of 1000 female utterances (with duration 49min24sec) and 1000 male utterances (with duration 51min58sec), and the testing part is consist of 132 female utterances (with duration 7min10sec) and 132 male utterances (with duration 7min21sec). All waveforms were down-sampled to 8 KHz, and the corresponding frame length was set to 256 points (i.e.32ms) with a frame shift of 128 pints (i.e. 16ms). 256-point FFT was used to compute the spectrum of each frame. Then 129-dimensional normalized log-power spectrum was computed for each frame. Subsequently, we jointed 9 consecutive frames together as the input vector for the model. The number of epoch for each layer of RBM pre-training was 50. Learning rate was set at 0.1 for the first 10 epochs, and then decreased at a rate of 1/10 after each epoch. The batch size was set to 100.

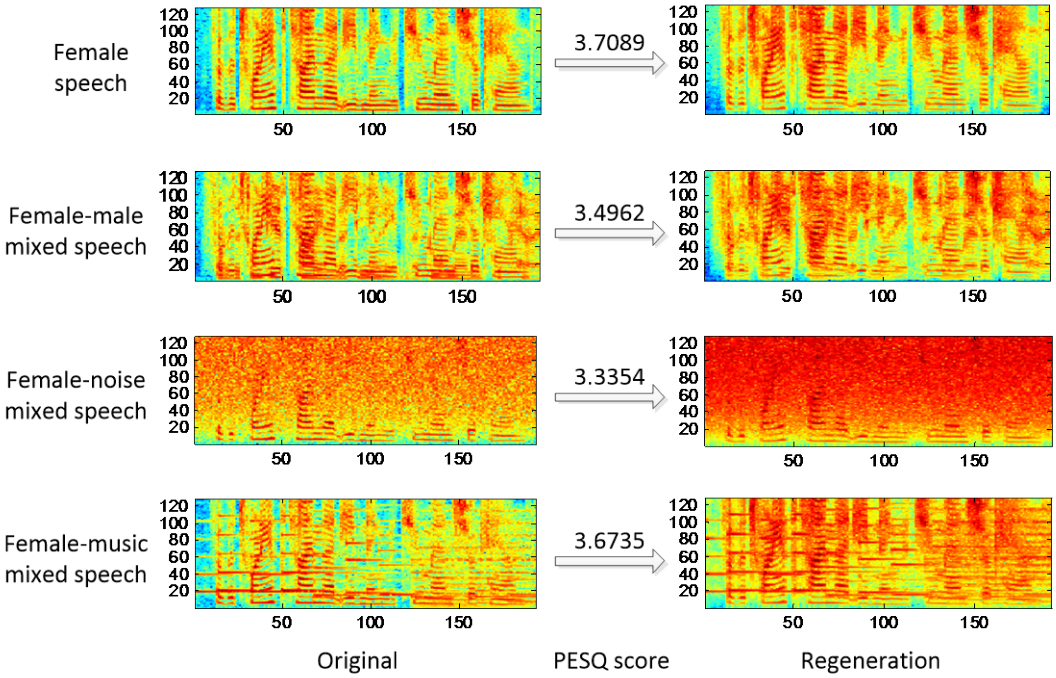

Fig.3 Regeneration performance of autoencoders

Evaluation of the autoencoders. We set the speech of the female speakers as the target, and trained a target-speech-autoencoder with the selected 1000 female utterances. Then we mixed the 1000 female utterances respectively with 1000 male utterances, Gaussian-noise signals (SNR=3.52dB), and background music signals (in our experiment, we choose Beethoven's "For Alice" as the background music). Three mixed-signal-autoencoders were trained separately on these three types of mixtures. The spectra of the female speech and the three mixtures were shown in the left rows of Figure 3. Through the target-speech-autoencoder and the three mixed-signal-autoencoders, we got the regenerated spectra which are shown in the right rows of Figure 3. The PESQ scores[14] of the regenerations are shown in the middle row of Figure 3.

Evaluation of the hidden variables of BP neural network. As mentioned in step 5, section 3, the number of hidden variables in the BP neural network influences the mapping performance. As is shown in Table 1, we tested different numbers of hidden variables for three BP neural networks which were trained separately for mapping features from three different types of mixed signals (i.e. female-male mixed speech, female-noise mixed speech, and female-music mixed speech) to target speech. The maximum training epoch was set to 1000, the minimum mean square error was set to 10^{-2}, and the minimum gradient was set to 10^{-5}. The training data was the activations of the middle layer of the mixed-signal-autoencoders which were fed with the mixed training speech, and the supervisions for them were the activations of the middle layer of the target-speech-autoencoder which were fed into the target training speech corresponding to the mixed training speech. The fitting performances of the BP neural networks were measured by the mean square errors(MSEs) of the training datasets and the test datasets separately. We can see from Table 1 that the best number of hidden variables in the BP neural networks is 500 which will be adopted as the optimal number of hidden variables in section 4.3.

Table 1. MSE residuals of the BP neural networks

Number of hidden variables	MSEs of Training / Testing		
	Female-male	Female-noise	Female-music
100	11.5343 / 30.9848	18.9889 / 47.7841	8.5127 / 22.6533
200	4.2681 / 14.3670	7.3864 / 26.7860	2.9932 / 12.0423
300	0.5121 / 9.5428	0.7771 / 15.3162	0.3513 / 7.9322
400	0.4788 / 5.6098	0.7379 / 12.9976	0.3588 / 4.6657
500	0.4064 / 2.1854	0.6130 / 8.6779	0.3060 / 1.7615
600	0.6351 / 5.8168	0.9790 / 10.7320	0.4856 / 3.9849

Evaluation of separation models. Through the above tests, we obtain the optimal structure of the speech separation model, whose numbers of variables from lower layer to upper layer are 1161-3000-312-500-312-3000-1161. We trained speech separation models with such a structure, and Figure 4 shows the separation performance. The PESQ scores of the extracted speech via target speech are 3.32(female-male), 2.59(female-noise), and 3.42(female-music). The inference in female-male mixed signal, that is the male speech, has a similar spectrum structure with the target speech. The inference in female-noise mixed single, that is the Gaussian noise, has no obvious patterns in spectrum structure. And the inference in female-music mixed single, which is the Beethoven's piano music "To Alice", has strong patterns in spectra. From this we can infer that the more regular the structures of the inference signal behaves, the better the separation model performs.

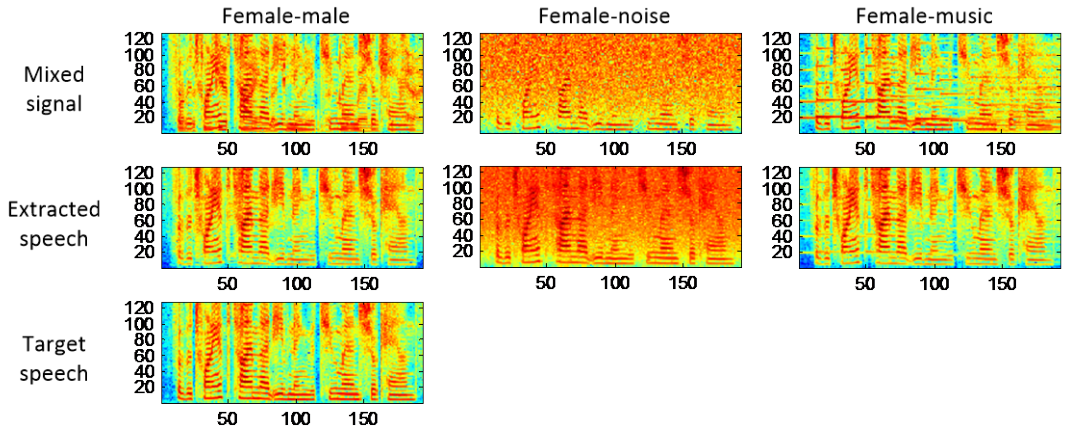

Fig.4 Separation performance of three speech separation models

Summary

In this paper, a novel method for speech separation was proposed. The proposed method extracted features from mixed signals and target speech by DBNs separately. The extracted features were associated by training a BP neural network to obtain a mapping from the features of mixed signals to the features of target speech. By training the model, it was able to extract the target speech by reconstructing the input mixed signal. According to our experiments, the model performed well on speech separation tasks, especially to the mixed signals where the inference signals have obvious spectral patterns. In future work, we will improve the model's performance on speech separation of mixed signals whose inference signals show no obvious spectral structures.

References

[1] F. Nesta, P. Svaizer, M. Omologo. Convolutive BSS of Short Mixtures by ICA Recursively Regularized Across Frequencies[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2011.3, 19(3):624-639.

[2] Virtanen, Tuomas. Monaural Sound Source Separation by Nonnegative Matrix Factorization with Temporal Continuity and Sparseness Criteria[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(1):1-12.

[3] Y. Ephraim, H. L. Van Trees. A Signal Subspace Approach for Speech Enhancement[J]. IEEE Transactions on Speech and Audio Processing, 1995.7, 3(4):251-266.

[4] G. J. Brown, M. P. Cooke. Computational Auditory Scene Analysis[J]. Computer Speech and Language, 1994, 8:297-336.

[5] G. N. Hu, D. L. Wang. Segregation of Unvoiced Speech from Nonspeech Interference[J]. Journal of the Acoustical Society of America, 2008, 124:1306-1319.

[6] Z. Z. Jin, D. L. Wang. A Supervised Learning Approach to Monaural Segregation of Reverberant Speech[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2009, 17(4):625-638.

[7] Y. Xu, J. Du, L. R. Dai, C. H. Lee. An Experimental Study on Speech Enhancement Based on Deep Neural Networks[J]. IEEE Signal Processing Letters,2014.1, 21(1): 65-68.

[8] Y. Bengio, O. Delalleau. On the Expressive Power of Deep Architectures[C]. Proceeding of the 14th International Conference on Discovery Science, Berlin, 2011.1.

[9] G. E. Hinton, S. Osindero, Y. W. Teh. A Fast Learning Algorithm for Deep Belief Nets[J]. Neural Computation, 2006, 18:1527-1554.

[10] M. A. Keyvanrad, M. Pezeshki, M. M. Homayounpour. Deep Belief Networks for Image Denoising[C]. International Conference on Learning Representations (ICLR2014), Banff, Canada, 2014.4.

[11] L. Deng, M. Seltzer, D. Yu, etc. Binary Coding of Speech Spectrograms Using a Deep-encoder[C]. ISCA, Chiba, Japan, 2010.9.

[12] D. Yu, L. Deng. Deep Learning and its Applications to Signal and Information Processing[J]. IEEE Signal Processing Magazine, 2011, 28(1):145-154.

[13] X. Zhu, G. T. Beauregard, L. L. Wyse. Real-time Signal Estimation From Modified Short-time Fourier Transform Magnitude Spectra[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(5): 1645-1653.

[14] Y. Hu, P. Loizou. Evaluation of Objective Measures for Speech Enhancement[C]. Proceedings of INTERSPEECH, Philadelphia, PA, 2006.9.