

# Joint Optimization of Recurrent Networks Exploiting Source Auto-regression for Source Separation

Shuai Nie<sup>1</sup>, Wei Xue<sup>1</sup>, Shan Liang<sup>1</sup>, Xueliang Zhang<sup>2</sup>, Wenju Liu<sup>1</sup>, Liwei Qiao<sup>3</sup>, Jianping Li<sup>3</sup>

<sup>1</sup>National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences

<sup>2</sup>College of Computer Science, Inner Mongolia University

<sup>3</sup>Electric Power Research Institute of ShanXi Electric Power Company, China State Grid Corp

{shuai.nie, wxue, sliang, lwj}@nlpr.ia.ac.cn    cszx1@imu.edu.cn

## Abstract

In music interferences condition, source separation is very difficult. In this paper, we propose a novel recurrent network exploiting the auto-regressions of speech and music interference for source separation. An auto-regression can capture the short-term temporal dependencies in data to help the source separation. For the separation, we independently separate the magnitude spectra of speech and interference from the mixture spectra by including an extra masking layer in the recurrent network. Compared to directly evaluating the ideal mask, the extra masking layer relaxes the assumption of independence between speech and interference which is more suitable for the real-world environments. Using the separated spectra of speech and interference, we further explore a discriminative training objective and joint optimization framework for the proposed network, which incorporates the correlations and spectral dependencies of speech and interference into the separation. Systematic experiments show that the proposed model is competitive with the state-of-the-art method in singing-voice separations.

**Index Terms:** source separation, deep recurrent neural networks, discriminative training objective, autoregressive models.

## 1. Introduction

In realistic environments, the interested signals are usually interfered by noises, which substantially degrades the performances of many applications, such as automatic speech recognition (ASR) and chord recognition [1, 9, 12, 14]. To address this issue, decades of efforts have been devoted to the source separation, but the separation is still a challenging task in realistic environments, especially when noise is non-stationary, such as music interferences, and only one microphone is available.

Source separation aims to segregate the interested sources from a mixture of signals. It can be naturally formulated as a supervised learning problem. Recently, supervised source separation has been extensively studied and achieve substantial performance improvements in monaural conditions [10, 12, 20, 24].

Due to speech production mechanisms, speech has prominent short- and long-term spectral dependencies, and presents obvious harmonic structures and temporal continuities. These information can be exploited for speech separation. In previous works [20, 21, 24], a common method is to expand the feature vector with the neighboring frames or delta features. However, due to the explosive increase of input feature dimension, this

technique only has a limited capacity to capture the temporal information within a limited span. Recurrent neural networks (RNNs) respecting temporal dynamics is regarded as a very promising model for sequential data, such as speech and music [3, 7]. Through the recurrent connections, RNN can capture some spectral dependencies in data, but the vanishing gradient problem makes the optimization of RNN very difficult [2].

Speech can be described as a auto-regression process [15]. Through a  $N$  order autoregressive model (AR), the current speech frame can be predicted by its limited historical frames [15]. Unfortunately, in noisy environments, speech is interfered by noise and its harmonic structure is severely corrupted, which makes predicting speech in noisy mixture very difficult. However, recent deep neural network (DNN)-based speech separation can successfully separate speech from the noisy mixture and clearly recover its harmonic structure. Therefore, through an AR model, we can use the historical separated spectra to predict the clean spectra. In turn, the predicted spectra can be fed into the separation model to further improve the separation performance. In this paper, we propose a novel recurrent network consisting of speech separation networks (SSN) and auto-regression networks (ARN), denoted as “auto-regression-separation networks” (ARSN), to jointly model and optimize the speech auto-regression and separation processes. Then, we further explore a discriminative training objective and joint optimization method for the ARSN.

We summarize our contributions as follows: 1) proposing a novel recurrent network exploiting the source auto-regression for the separation, which can capture the short- and long-term spectral dependencies in signals; 2) exploring a discriminative training objective and joint optimization method for the proposed network; 3) relaxing the assumption of independence between speech and noise in the discriminative training objective.

## 2. A Structure Overview of ARSN

In this section, we use a specific example of singing-voice separation, shown in Fig. 1, to describe ARSN’s structure. Typically, a ARSN consists of one SSN, several memory queues and  $N$  order ARNs. The memory queues temporarily store the separated spectra from the SSN according to the time sequence. As the length of memory queue is limited, the new separated spectra chronologically squeeze out the old one. The separated spectra stored in memory queues are fed into the ARNs to predict the next frame of clean spectra. There are two ARNs in Fig. 1, one is used for predicting the singing-voice spectra, denoted as SARN, and the other is used for predicting the music accompaniments spectra, denoted as NARN. In turn, the predicted spec-

This research was partly supported by the China National Nature Science Foundation (No.91120303, No.61273267, No.90820011, No.61403370 and No.61365006).

tra are input into the SSN along with the noisy mixture spectra, to help the separation. By repeating this process along time, the separation of the whole sequence can be completed.

Although the ARSN looks like a RNN [4], they have many essential differences. The key difference is that the recurrent connection of a RNN is the self-connection of a hidden layer and that of the ARSN is the connection between the input layer and output layer. And the recurrent connection of ARSN is performed by a ARN consisting of one input layer, multiple hidden layers and one output layer, which is explicitly guided to learn short-term dependencies in data. As the recurrent connections have explicit learning targets, the ARSN can partly circumvent the vanishing gradient problem compared to RNN.

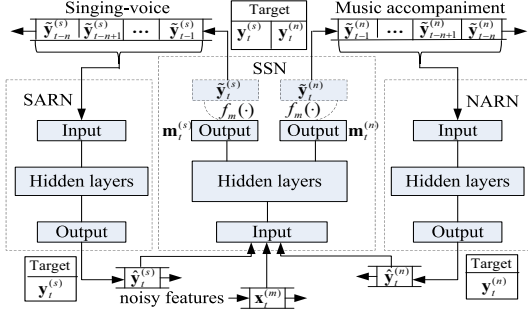


Figure 1: The structure of ARSN.

### 3. Training Objective

The goal of source separation is to segregate all sources (e.g., speech and noise) from a mixture signal. The commonest method is the time-frequency (T-F) masking technique [16, 21, 22]. Once a T-F mask  $\mathbf{m}_t$  is computed, it can be applied to the magnitude spectra  $\mathbf{x}_t^{(m)}$  of mixture signals to obtain the estimated separation spectra of one source  $\tilde{\mathbf{y}}_t^{(s)}$  (e.g., speech). If the noise is statistically independent of the speech and they are zero mean, we have  $\mathbf{x}_t^{(m)} \approx \mathbf{y}_t^{(s)} + \mathbf{y}_t^{(n)}$  [5], where  $\mathbf{y}_t^{(s)}$  and  $\mathbf{y}_t^{(n)}$  are the magnitude spectra of the clean speech and noise, respectively. Under this assumption, the estimated separation spectra  $\tilde{\mathbf{y}}_t^{(n)}$  of noise can be obtained by applying  $(1 - \mathbf{m}_t)$  to  $\mathbf{x}_t^{(m)}$ . However, in real environments, the noise is usually not statistically independent of the speech. Therefore, we relax the independence assumption and separately evaluate one mask for each source in mixture signals in this paper. Then, the evaluated separation spectra  $\tilde{\mathbf{y}}_t^{(s)}$  and  $\tilde{\mathbf{y}}_t^{(n)}$  can be computed as follows:

$$\tilde{\mathbf{y}}_t^{(s)} = f_m(\mathbf{m}_t^{(s)}) = \mathbf{m}_t^{(s)} \circ \mathbf{x}_t^{(m)} \quad (1)$$

$$\tilde{\mathbf{y}}_t^{(n)} = f_m(\mathbf{m}_t^{(n)}) = \mathbf{m}_t^{(n)} \circ \mathbf{x}_t^{(m)} \quad (2)$$

The symbol  $\circ$  denotes the element-wise multiplication and  $f_m(\cdot)$  is the T-F masking function.  $\mathbf{m}_t^{(s)}$  and  $\mathbf{m}_t^{(n)}$  are the masks of speech and noise. In ideal conditions, they are equal to  $\frac{\mathbf{y}_t^{(s)}}{\mathbf{x}_t^{(m)}}$  and  $\frac{\mathbf{y}_t^{(n)}}{\mathbf{x}_t^{(m)}}$ . We should note that  $\mathbf{x}_t^{(m)}$ ,  $\mathbf{m}_t$ ,  $\mathbf{y}_t^{(s)}$ ,  $\mathbf{y}_t^{(n)}$ ,  $\tilde{\mathbf{y}}_t^{(s)}$ ,  $\tilde{\mathbf{y}}_t^{(n)}$ ,  $\mathbf{m}_t^{(s)}$  and  $\mathbf{m}_t^{(n)}$  are  $1 \times F$  row vectors at the frame  $t$ . Here,  $F$  is the number of frequency bins.

Although we can directly learn the ideal masks  $\frac{\mathbf{y}_t^{(s)}}{\mathbf{x}_t^{(m)}}$  and  $\frac{\mathbf{y}_t^{(n)}}{\mathbf{x}_t^{(m)}}$  [21], they don't directly optimize the actual source separation objective and can't perform the discriminative training.

Compared to the direct mask approximation [21], the spectra approximation is more flexible but has a problem of estimation of unbounded values [24]. In this paper, we learn the clean magnitude spectra through jointly training the networks with the T-F masking function. The T-F masking function can be viewed as an extra layer that is added to the original output of the network as shown in Fig. 1, which is used to generate the evaluated separation spectra. We should note that the extra layer is a deterministic layer and has no connective weights to be optimized, but its resulting outputs are used to compute the error metric for optimizing the network weights.

Seemingly, we don't learn the masks for separations, but in fact, the extra layer automatically learns masks associated with the target spectra. This indirect and masking-based spectra approximation owns the advantages of both spectra and mask approximation. It can obtain the separation results with higher smoothness [11] and has less difficulties in optimization [21].

Commonly, simultaneously modeling all sources in one model can achieve better separation performance than only modeling one target source [11], which owes to the correlations and complementarity of different prediction sources in mixture signals. In addition, the different prediction outputs can be used to perform the discriminative training. In this paper, we explore a discriminative objective that not only increases the similarity between the prediction and its target, but also decreases the similarity between the prediction and the targets of other sources. In addition, to jointly optimize the SSN and the ARN in the ARSN, we further explore a joint training objective for the ARSN. Therefore, the final objective contains three terms (each term is surrounded by a pair of brackets), as shown in Eq. (3). The first and second terms are the discriminative objective of SSN, and the third term is the prediction errors of the SARN and the NARN. In this paper, we simultaneously deal with all objectives and jointly optimize them in one model, which can help the optimization of each objective.

$$J_t = \frac{1}{2} \left( \left\| \tilde{\mathbf{y}}_t^{(s)} - \mathbf{y}_t^{(s)} \right\|_2^2 + \left\| \tilde{\mathbf{y}}_t^{(n)} - \mathbf{y}_t^{(n)} \right\|_2^2 \right) - \frac{\beta}{2} \left( \left\| \tilde{\mathbf{y}}_t^{(s)} - \mathbf{y}_t^{(n)} \right\|_2^2 + \left\| \tilde{\mathbf{y}}_t^{(n)} - \mathbf{y}_t^{(s)} \right\|_2^2 \right) + \frac{\lambda}{2} \left( \left\| \hat{\mathbf{y}}_t^{(s)} - \mathbf{y}_t^{(s)} \right\|_2^2 + \left\| \hat{\mathbf{y}}_t^{(n)} - \mathbf{y}_t^{(n)} \right\|_2^2 \right) \quad (3)$$

where  $\hat{\mathbf{y}}_t^{(s)}$  and  $\hat{\mathbf{y}}_t^{(n)}$  are the outputs of SARN and NARN, respectively.  $\beta$  and  $\lambda$  specify the relative importance of each term and can be experimentally chosen.

### 4. Optimization

We use the backpropagation through time (BPTT) algorithm [17] to optimize the ARSN. The basic principle of BPTT is that of "unfolding", which means duplicating all recurrent weights spatially for the arbitrary time steps [4], here referred to as  $T$ . An ARSN can be unfolded to a very deep feedforward network along the time. Fig. 2 shows part of an unfolded ARSN ( $T = 3$ ), and the order  $N$  of ARN is set to be 2 in Fig. 2.

#### 4.1. Forward propagation

The forward pass of ARSN is the same as that of a feedforward network, except that the outputs of SSN will be fed into SARN and NARN, and in turn, the outputs of SARN and NARN will be fed into SSN. Taking the mixture magnitude spectra  $\mathbf{x}_t^{(m)}$  ( $t = 1, 2, \dots, T$ ) as the input sequence, we chronologically compute the activations of all layers in the ARSN by Eq. (4).

$$\mathbf{z}_{t,l+1} = \mathbf{W}_l \times (\mathbf{a}_{t,l})^T; \quad \mathbf{a}_{t,l+1} = f(\mathbf{z}_{t,l+1}) \quad (4)$$

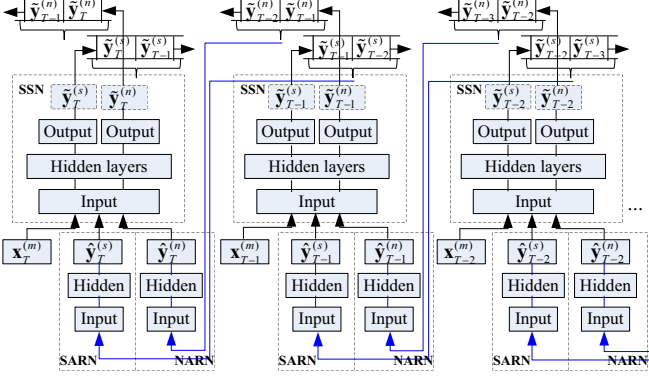


Figure 2: The effect of unfolding a ARSN for BPTT.

where  $\mathbf{a}_{t,l+1}$  is the activations of the layer  $(l+1)$  at time  $t$ ,  $\mathbf{z}_{t,l+1}$  is the total weighted sums of inputs to the layer  $(l+1)$  at time  $t$ , including the bias term, and  $\mathbf{W}_l$  is the connection weights between the layer  $l$  and the layer  $(l+1)$ .  $f(\cdot)$  is the element-wise activation function and the symbol  $(\cdot)^T$  denotes the transpose of a matrix. To simplify the mathematical derivation of optimization for the ARSN, we denote  $\mathbf{a}_{t,l+1}^{(m)}$ ,  $\mathbf{z}_{t,l+1}^{(m)}$  and  $\mathbf{W}_l^{(m)}$  for SSN, denote  $\mathbf{a}_{t,l+1}^{(s)}$ ,  $\mathbf{z}_{t,l+1}^{(s)}$  and  $\mathbf{W}_l^{(s)}$  for SARN, and denote  $\mathbf{a}_{t,l+1}^{(n)}$ ,  $\mathbf{z}_{t,l+1}^{(n)}$  and  $\mathbf{W}_l^{(n)}$  for NARN. Mathematically, they can be formulated through specializing Eq. (4).

Different with a feedforward network, for the input layers  $(l=1)$  in the ARSN,  $\mathbf{a}_{t,1}^{(m)} = [1, \mathbf{x}_t^{(m)}, \hat{\mathbf{y}}_t^{(s)}, \hat{\mathbf{y}}_t^{(n)}]$ ,  $\mathbf{a}_{t,1}^{(s)} = [1, \tilde{\mathbf{y}}_{t-2}^{(s)}, \tilde{\mathbf{y}}_{t-1}^{(s)}]$  and  $\mathbf{a}_{t,1}^{(n)} = [1, \tilde{\mathbf{y}}_{t-2}^{(n)}, \tilde{\mathbf{y}}_{t-1}^{(n)}]$ . And for the output layers  $(l=n_l)$ , the numbers of the layers in SSN, SARN or NARN),  $[\mathbf{m}_t^{(n)}, \mathbf{m}_t^{(s)}] = \mathbf{a}_{t,n_l}^{(m)}$ ,  $\hat{\mathbf{y}}_t^{(s)} = \mathbf{a}_{t,n_l}^{(s)}$  and  $\hat{\mathbf{y}}_t^{(n)} = \mathbf{a}_{t,n_l}^{(n)}$ ,  $\tilde{\mathbf{y}}_t^{(s)}$  and  $\tilde{\mathbf{y}}_t^{(n)}$  can be computed by Eqs. (1) and (2).

## 4.2. Back propagation

Using the chain rule, we can recursively compute the gradients of the loss function with respects to all weights in the network from time  $T$  to time 1, as follows:

$$\nabla \mathbf{W}_{t,l} = \frac{\partial J_t}{\partial \mathbf{W}_l} = \frac{\partial J_t}{\partial \mathbf{z}_{t,l+1}} \frac{\partial \mathbf{z}_{t,l+1}}{\partial \mathbf{W}_l} = \frac{\partial J_t}{\partial \mathbf{z}_{t,l+1}} (\mathbf{a}_{t,l})^T \quad (5)$$

To simplify notations, we introduce a variable  $\delta$  and make  $\delta_{t,l}^{(m)} = \frac{\partial J_t}{\partial \mathbf{z}_{t,l}^{(m)}}$ ,  $\delta_{t,l}^{(s)} = \frac{\partial J_t}{\partial \mathbf{z}_{t,l}^{(s)}}$  and  $\delta_{t,l}^{(n)} = \frac{\partial J_t}{\partial \mathbf{z}_{t,l}^{(n)}}$ . They measure how much the nodes of the  $l$ -th layer in SSN, SARN and NARN are ‘‘responsible’’ for any errors in their outputs, respectively.

For the  $\delta$  term of output layer  $(l=n_l)$ ,  $\delta_{t,n_l}^{(m)}$  can be computed as follows:

$$\delta_{t,n_l}^{(m)} = [\mathbf{s}_t, \mathbf{n}_t] \circ [\mathbf{x}_t^{(m)}, \mathbf{x}_t^{(m)}] \circ f'(\mathbf{z}_{t,n_l}^{(m)}) \quad (6)$$

where  $\mathbf{s}_t = -(\mathbf{y}_t^{(s)} - \tilde{\mathbf{y}}_t^{(s)}) + \beta(\mathbf{y}_t^{(n)} - \tilde{\mathbf{y}}_t^{(n)}) + \mathbf{I}_t(\delta_{t+1,1}^{(s)}) + \mathbf{I}_t(\delta_{t+2,1}^{(s)})$  and  $\mathbf{n}_t = -(\mathbf{y}_t^{(n)} - \tilde{\mathbf{y}}_t^{(n)}) + \beta(\mathbf{y}_t^{(s)} - \tilde{\mathbf{y}}_t^{(s)}) + \mathbf{I}_t(\delta_{t+1,1}^{(n)}) + \mathbf{I}_t(\delta_{t+2,1}^{(n)})$ . Here,  $\mathbf{I}_t(\delta_{t+1,1}^{(s)})$  and  $\mathbf{I}_t(\delta_{t+2,1}^{(s)})$  mean selecting the  $\delta$  terms of the input nodes corresponding to  $\tilde{\mathbf{y}}_t^{(s)}$  from  $\delta_{t+1,1}^{(s)}$  and  $\delta_{t+2,1}^{(s)}$ , respectively. Likewise,  $\mathbf{I}_t(\delta_{t+1,1}^{(n)})$  and  $\mathbf{I}_t(\delta_{t+2,1}^{(n)})$  perform similar operation but on  $\delta_{t+1,1}^{(n)}$  and  $\delta_{t+2,1}^{(n)}$ . And  $\delta_{t,n_l}^{(s)}$  can be computed as follows:

$$\delta_{t,n_l}^{(s)} = (-\lambda(\mathbf{y}_t^{(s)} - \hat{\mathbf{y}}_t^{(s)}) + \mathbf{I}_s(\delta_{t,1}^{(m)})) \circ f'(\mathbf{z}_{t,n_l}^{(s)}) \quad (7)$$

where  $\mathbf{I}_s(\delta_{t,1}^{(m)})$  means selecting the  $\delta$  terms of the input nodes corresponding to  $\hat{\mathbf{y}}_t^{(s)}$  from  $\delta_{t,1}^{(m)}$ .

For the  $\delta$  term of  $l$ -th layer  $(l=n_l-1, n_l-2, \dots, 1)$ ,  $\delta_{t,l}^{(m)}$  can be computed as follows:

$$\delta_{t,l}^{(m)} = ((\mathbf{W}_l^{(m)})^T \times \delta_{t,l+1}^{(m)}) \circ f'(\mathbf{z}_{t,l}^{(m)}) \quad (8)$$

And  $\delta_{t,l}^{(s)}$  can be computed by as follows:

$$\delta_{t,l}^{(s)} = ((\mathbf{W}_l^{(s)})^T \times \delta_{t,l+1}^{(s)}) \circ f'(\mathbf{z}_{t,l}^{(s)}) \quad (9)$$

Similar to  $\delta_{t,n_l}^{(s)}$  and  $\delta_{t,l}^{(s)}$ ,  $\delta_{t,n_l}^{(n)}$  and  $\delta_{t,l}^{(n)}$  can be computed by replacing the superscript  $(s)$  in Eqs. (7) and (9) with  $(n)$ .

After obtaining all  $\delta$  terms from time  $T$  to time 1, the partial derivatives of the loss function with respects to all weights in the ARSN can be computed by Eq. (10)

$$\nabla \mathbf{W}_l = \sum_{t=1}^T \nabla \mathbf{W}_{t,l} = \sum_{t=1}^T (\delta_{t,l} \times (\mathbf{a}_{t,l-1})^T) \quad (10)$$

Then, we use the limited memory Broyden Fletcher Goldfarb-Shanno (L-BFGS) algorithm [13] to update the weights  $\mathbf{W}_l$ .

## 5. Experiments

### 5.1. Dataset and evaluation metrics

We apply the proposed model to singing-voice separations to examine its effectiveness, but singing-voice separation is not the only application. It can be easily applied to other source separation task, e.g., speech separation. In fact, as music interferences are more complex and non-stationary, singing-voice separation is more challenging than speech separation.

We systematically evaluate the proposed model on the MIR-1k dataset [8]. This dataset contains 1000 song clips with total length of 133 min. These clips were extracted from 110 Chinese karaoke songs performed by 11 male and 8 female amateurs. For each clip, the singing voice and the music accompaniment are recorded in different channels and we mix them at 0 dB to obtain the mixture clips. For training, we randomly choose 794 clips from 9 male and 6 female amateurs as the training set. And the remaining 206 clips are used for testing. 176 clips of them are from other 2 male and 2 female amateurs, which is used to evaluate the generalization ability to the unmatched singer. The remaining 30 clips are from the same amateurs to that of the training set, but have different contents.

We take Source to Interference Ratio (SIR), Source to Artifacts Ratio (SAR), and Source to Distortion Ratio (SDR) as the evaluation metrics. They measure the ratios of source to interference, artifacts and distortion, respectively, and can be computed by the BSS Eval toolbox [19]. Higher values of SDR, SAR, and SIR mean the better separation quality.

### 5.2. Related models and configurations

We choose the deep RNN-based (DRNN) method proposed by Huang [11] for comparison, which is regarded as the state-of-the-art in singing-voice separations. We use the implementation and the best configuration setting provided by Huang [18] for DRNN in experiments. To be the same as the total number of parameters of the DRNN, we set that the SSN in ARSN has 3 hidden layers of 1000 units with the rectified linear unit (ReLU) activation function [6], and both SARN and NARN has one hidden layers of 250 units with ReLU activation function. As

the values of the masks  $m_t^{(s)}$  and  $m_t^{(n)}$  exceed 1, to avoid the unbounded values estimation problem, we constrain the output of the ARSN to  $[0, \theta]$  by using a bounded linear activation function  $f(x) = \max(0, \min(x, \theta))$  for the output layer. We set  $\theta$  to 2.5 according to the performance on the development set<sup>1</sup>.

To optimize the ARSN, we back-propagate the gradients of the loss function through 100 time steps. Hence, we randomly cascade all clips in training set into a long sequence, and then ‘chop’ it into sequences of 100 frames ( $T = 100$ ) with 50% overlap. The L-BFGS algorithm is used to train the model from the random initialization, which is paralleled on a graphics processing unit (GPU). We set the maximum epoch to 400. According to the development set<sup>1</sup>, we set  $\beta = 0.05$  and  $\lambda = 0.1$ , and the order  $N$  of ARN is set to 5. Moreover, to reduce the over-fitting on the training set, we add Gaussian noise ( $\mu = 0, \delta = 0.2$ ) to the inputs  $\hat{y}_t^{(s)}$  and  $\hat{y}_t^{(n)}$  of SSN.

In all experiments, we use magnitude spectra as the input feature. We also explore the log-mel filterbank features and log power spectrum, but empirically obtain the worse performance. The spectral representation is extracted by applying a 1024-point short time Fourier transform (STFT) with 50% overlap to the mixture signals. Moreover, we find the context features can further improve the performance for ARSN. Therefore, we use a 3-frame window of features as the input feature of all models.

### 5.3. Results and discussions

First, we compare the effects of 4 different training objectives on the separation performance, including the ideal ratio mask [23], the short-time Fourier transform spectral magnitude [24], the discriminative spectra approximation proposed by Huang [11], and the proposed mask-based discriminative spectra approximation. They are tested by the DNN-based singing-voice separation and denoted as ‘IRM’, ‘FFT-Spectra’, ‘HuangObj’ and ‘ProposedObj’, respectively. In all experiments, the used DNNs have 3 ReLU hidden layers of 1000 units but different output layers for different training objectives. From Table 1, we observe that HuangObj and ProposedObj significantly outperform IRM and FFT-Spectra, and HuangObj performs best on SDR and SAR. But compared to HuangObj, ProposedObj achieves significant improvement on SIR with only a little losses of SDR and SAR, which shows that ProposedObj can suppress more interferences with very little cost of artifacts and distortion. Moreover, HuangObj has an unbounded values estimation problem when computing the partial derivative at the output layer, which makes the optimizations very difficult.

Table 1: The effects of different training objectives on the separation performance

Objectives	Matched singer			Unmatched singer		
	SDR	SIR	SAR	SDR	SIR	SAR
Mix	0.00	0.00	$\infty$	0.00	0.00	$\infty$
IRM	8.18	11.88	8.75	7.80	11.26	8.34
Spectra	8.05	13.58	8.01	8.07	13.58	8.11
HuangObj	<b>9.48</b>	14.43	<b>9.80</b>	<b>9.04</b>	13.93	<b>9.36</b>
ProposedObj	9.43	<b>17.41</b>	9.62	8.88	<b>16.28</b>	9.07

Table 2 presents the results of different singing-voice separation methods including DNN, DRNN and ARSN. We observe

<sup>1</sup>Four clips, Ani\_2\_02, stool\_4\_01, bobon\_1\_01 and heycat\_4\_09.

that DRNN and ARSN consistently outperform DNN. It suggests that recurrent networks are likely more suitable for the singing-voice separation problem due to the capacity in capturing temporal dependences in sequence data. But due to the vanishing gradient problem, DRNN does not achieve significant improvements in separation performance compared to DNN. In contrast, ARSN shows obvious performance improvements, especially on SIR. It mainly owes to the discriminative training objective that relaxes the assumption of independence between speech and noise. In addition, the ARSN exploiting the auto-regressions of singing voice and music accompaniment can capture the short- and long-term spectral dependences in signals.

Table 2: The performances of different separation models.

Models	Matched singer			Unmatched singer		
	SDR	SIR	SAR	SDR	SIR	SAR
DNN	9.48	14.43	9.80	9.04	13.93	9.36
DRNN	9.96	15.53	10.22	9.47	15.03	9.69
ARSN	<b>10.24</b>	<b>19.92</b>	<b>10.48</b>	<b>9.50</b>	<b>18.26</b>	<b>9.70</b>

Finally, we show some results of the singing-voice separation in Fig. 3. The predicted singing voice spectra present the clear harmonic structure. It suggests that SARN can capture the short-term spectral dependences and trace the spectra structure in singing voice by the AR model. Compared to the predicted spectra, the separated spectra present more complete structure and richer details, especially in high frequency bands. It indicates that the singing voice auto-regression captures the spectral structure that can provide rich information for the separation.

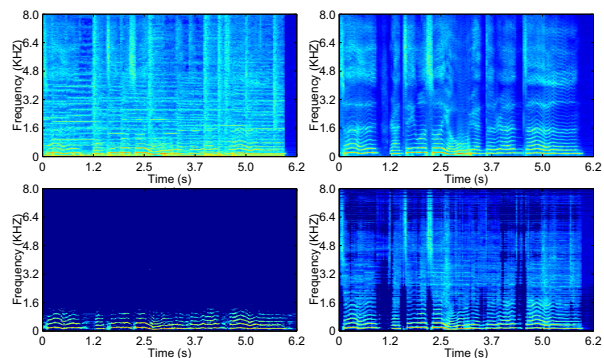


Figure 3: Top left: The mixture magnitude spectra (in log scale); Top right: The groundtruth singing voice spectra; Bottom left: the predicted singing voice spectra by 5 order ARN; Bottom right: the separated singing voice spectra by SSN.

## 6. Conclusions

In this paper, a novel recurrent network exploiting source auto-regressions is proposed and jointly optimized to capture the short- and long-term spectra dependences in signals. The experiments of singing-voice separation show that the proposed model outperforms the state-of-the-art method. Moreover, we find that relaxing the assumption of independence between speech and noise can achieve better separation performance. According to this fact, a mask-based discriminative spectra approximation objective are explored, which achieves further improvement of the separation performance.



## 7. References

- [1] J. Allen, "Articulation and intelligibility," *Synthesis Lectures on Speech and Audio Processing*, vol. 1, no. 1, pp. 1–124, 2005.
- [2] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.
- [3] S. Bock and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2012)*, 2012, pp. 121–124.
- [4] M. Bodén, "A guide to recurrent neural networks and backpropagation," *The Dallas project, SICS technical report*, 2002.
- [5] D. Burshtein and S. Gannot, "Speech enhancement using a mixture-maximum model," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 10, no. 6, pp. 341–351, 2002.
- [6] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*, vol. 15, 2011, pp. 315–323.
- [7] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2013)*, 2013, pp. 6645–6649.
- [8] C.-L. Hsu and J.-S. Jang, "On the improvement of singing voice separation for monaural recordings using the mir-1k dataset," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 18, no. 2, pp. 310–319, 2010.
- [9] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2012)*, 2012, pp. 57–60.
- [10] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2014)*, pp. 1562–1566, 2014.
- [11] —, "Singing-voice separation from monaural recordings using deep recurrent neural networks," *Proc. International Society for Music Information Retrieval (ISMIR'2014)*, 2014.
- [12] G. Kim, Y. Lu, Y. Hu, and P. Loizou, "An algorithm that improves speech intelligibility in noise for normal-hearing listeners," *The Journal of the Acoustical Society of America.*, vol. 126, pp. 1486–1494, 2009.
- [13] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.
- [14] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust asr," in *Proc. 13th Annual Conference of the International Speech Communication Association (INTERSPEECH'2012)*, 2012.
- [15] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*. Prentice-hall Englewood Cliffs, 1978, vol. 100.
- [16] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Proc. 11th Annual Conference of the International Speech Communication Association (INTERSPEECH'2010)*, 2010, pp. 717–720.
- [17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [18] (2014) Singing-voice separation from monaural recordings using deep recurrent neural networks. [Online]. Available: <https://sites.google.com/site/deeplearningsourceseparation/>
- [19] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 14, no. 4, pp. 1462–1469, 2006.
- [20] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 21, no. 7, pp. 1381–1390, 2013.
- [21] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE Trans. on Audio, Speech, and Lang. Proc.*, vol. 22, no. 12, pp. 1849–1858, 2014.
- [22] F. Weninger, F. Eyben, and B. Schuller, "Single-channel speech separation with memory-enhanced recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2014)*, 2014, pp. 3709–3713.
- [23] F. Weninger, J. Le Roux, J. R. Hershey, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. IEEE GlobalSIP 2014 Symposium on Machine Learning Applications in Speech Processing*, 2014.
- [24] Y. Xu, J. Du, L. Dai, and C. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters.*, vol. 21, no. 1, pp. 65–68, 2014.