A NEURAL NETWORK ALTERNATIVE TO CONVOLUTIVE AUDIO MODELS FOR SOURCE SEPARATION

Author(s) Name(s) omitted for double blind review

Author Affiliation(s) omitted for double blind review

ABSTRACT

***Index Terms*—** Auto-encoders, source separation, deep learning.

1. INTRODUCTION

Several neural network architectures have been proposed to develop algorithms for supervised source separation and speech enhancement. (Chandna monaural, Grais Single channel, Park Fully convolutional, Venkataramani end-to-end source). Currently, these networks are trained to learn discriminative audio models extensively. In other words, the spectrogram of the mixture is given as an input to the network. The goal of the network then, is to learn suitable time-frequency masks that separate the input spectrogram into the source and the interference componenents. Thus, the networks learn a basis decomposition that often works only for a specific source-interference pair, i.e., these models are not transferable. If the interfering signal changes, these networks have to be re-trained to learn suitable models to separate the new interfering signal in the mixture from the source.

A popular technique to learn transferable audio models for supervised source separation is the use of Nnon-negative matrix factorization (NMF). Non-negative matrix factorization (NMF) matrix of non-negative elements $\mathbf{X} \in \mathbb{R}^{\geq 0}_{M\times N}$ as a product of the basis matrix $\mathbf{W}$ and the activation matrix $\mathbf{H}$. The notation $\mathbb{R}^{\geq 0}_{M\times N}$ represents the set of matrices of non-negative elements of size $M \times N$. In this factorization, the basis matrix $\mathbf{W} \in \mathbb{R}^{\geq 0}_{M\times r}$, the activation matrix $\mathbf{H} \in \mathbb{R}^{\geq 0}_{r\times N}$ and r represents the rank of the decomposition. In the case of audio signals, we apply such a factorization on audio spectrograms. In this setting, the columns of $\mathbf{W}$ begin to act as representative basis vectors for the source. The rows of $\mathbf{H}$ indicate the activity of these basis vectors in time. As shown in [](ICASSP Paper reference...), the notion of non-negative audio modeling can be easily generalized by interpreting NMF as a neural network. We can interpret NMF as a non-negative auto-encoder in the following manner,

$$1^{\text{st}} \text{ layer: } \mathbf{H} = g(\mathbf{W}^{\ddagger} \cdot \mathbf{X}) \tag{1}$$

$$2^{\text{nd}} \text{ layer: } \mathbf{X} = g(\mathbf{W} \cdot \mathbf{H}) \tag{2}$$

Here, $\mathbf{X}$ represents the input spectrogram, $\mathbf{W}^{\ddagger}$ represents a form of pseudo-inverse of $\mathbf{W}$ and $g(.) : \mathbf{R} \rightarrow \mathbf{R}^{\geq 0}$ is an element-wise function that maps a real number to the space of positive real numbers. As before, the columns of $\mathbf{W}$ act as representative basis vectors and the corresponding rows of $\mathbf{H}$ indicate their respective activations. Although non-negativity of the models is not explicitly guaranteed in this formulation, applying a suitable sparsity constraint allows the network to learn suitable non-negative models. Additionally, this interpretation enables a pathway to propose multi-layer extensions by exploiting the wealth of available neural net architectures that could potentially lead to superior separation performance.

Spectrograms of speech and audio signals incorporate temporal dependencies that span multiple time frames. However, NMF and its neural network equivalent are unable to explicitly utilize these cross-frame patterns available in a spectrogram. To alleviate this drawback, Smaragdis [](Paris convolutive speech bases) proposed a convolutive version to NMF that allows spectro-temporal patterns as representative basis elements. In this paper, we develop a neural network alternative to such convolutive audio models for supervised source separation. In doing so, we solve two fundamental problems associated with this task. The first step is to develop a suitable neural network architecture to learn convolutive audio models in an adaptive manner. Utilizing the models to separate a source from a given mixture forms the second step. The remainder of the paper is organized as follows. In section **??**, we develop an auto-encoder that can act as an equivalent to conv-NMF audio models. Section **??** proposes a novel approach to utilize these models for supervised source separation. We evaluate these models in terms of their separation performance in section **??** and conclude in section **??**.

Thanks to XYZ agency for funding.

978-1-5090-6341-3/17/$31.00 ©2017 IEEE

2. NON-NEGATIVE CONVOLUTIIONAL AUTO-ENCODERS

2.1. Network Architecture

The convolutive NMF model [](Smaragdis convolutive speech bases) approximates a non-negative matrix $\mathbf{X} \in \mathbb{R}^{\geq 0}_{M \times N}$ as,

$$\mathbf{X}(f,t) \approx \sum_{i=1}^{r} \sum_{k=0}^{T-1} \mathbf{W}_i(k,f) \cdot \mathbf{H}(i,t-k) \tag{3}$$

Here, $\mathbf{W}_i \in \mathbb{R}^{\geq 0}_{M \times T}$ acts as the i^{th} basis matrix and $\mathbf{H} \in \mathbb{R}^{\geq 0}_{r \times N}$ contains the corresponding weights. The notation $\mathbf{X}(i,j)$ represents the element of $\mathbf{X}$ indexed by the i^{th} row and the j^{th} column. Thus, we can interpret this operation as a two-step convolutional auto-encoder as follows,

$$1^{\text{st}} \text{ layer: } \mathbf{H}(i,t) = \sum_{i=1}^{r} \sum_{j=0}^{M-1} \sum_{k=0}^{T-1} \mathbf{W}_i^{\ddagger}(j,k)\mathbf{X}(j,t-k) \tag{4}$$

$$2^{\text{nd}} \text{ layer: } \hat{\mathbf{X}}(f,t) = \sum_{i=1}^{r} \sum_{k=0}^{T-1} \mathbf{W}_i(k,f) \cdot \mathbf{H}(i,t-k) \tag{5}$$

subject to non-negativity of $\mathbf{W}_i$ and $\mathbf{H}$. Here, we assume that the convolutional filters $\mathbf{W}$, $\mathbf{W}^{\ddagger}$ have a size of $M \times T$ where, T represents the depth of the convolution. In this representation, $\mathbf{W}_i$ and $\mathbf{H}$ correspond to the i^{th} basis matrix and the activation matrix respectively. The filters of the first convolutional layer act as inverse filters in defining the auto-encoder. In the remainder of this section, we will refer to the first convolutional layer as the encoder that estimates a code from the input representation. The second convolutional layer generates an approximation of the input from the code and will be referred to as the decoder. We can simplify the non-negativity constraints by incorporating a non-linearity into the definitions of the encoder and the decoder. Thus,

$$1^{\text{st}} \text{ layer: } \mathbf{H}(i,t) = g\left(\sum_{i=1}^{r} \sum_{j=0}^{M-1} \sum_{k=0}^{T-1} \mathbf{W}_i^{\ddagger}(j,k)\mathbf{X}(j,t-k)\right) \tag{6}$$

$$2^{\text{nd}} \text{ layer: } \hat{\mathbf{X}}(f,t) = g\left(\sum_{i=1}^{r} \sum_{k=0}^{T-1} \mathbf{W}_i(k,f) \cdot \mathbf{H}(i,t-k)\right) \tag{7}$$

Here, the non-linearity $g(.) : \mathbb{R} \rightarrow \mathbb{R}^{\geq 0}$ applies an element-wise non-negativity constraint and ensures that the activation matrix and the reconstruction are non-negative. We now note a couple of key points about the convolutional auto-encoder. (i) The output of the encoder gives the latent representation of the decomposition. (ii) The weights of the decoder act as the basis vectors of the decomposition. (iii) We do not explicitly apply non-negativity constraints on the weights. Thus, the basis matrices (decoder filters) can assume negative values.

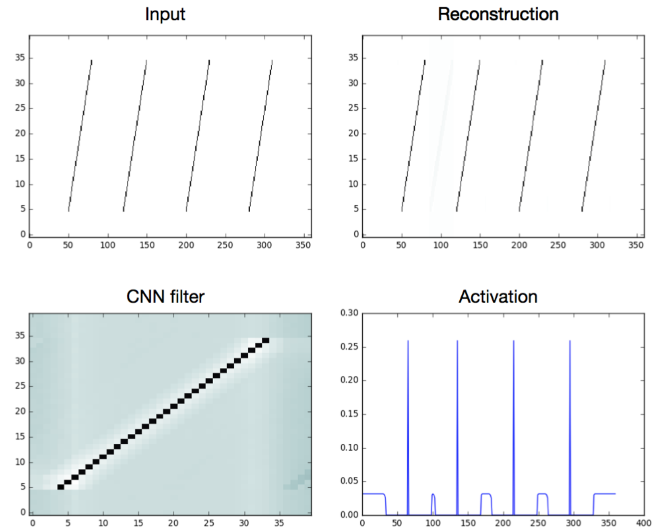

Fig. 1. ...

To train the auto-encoder and learn the basis and activation matrices, in the remainder of the paper, we apply the following approach. We minimize the KL-divergence between the input spectrogram $\mathbf{X}$ and its reconstruction $\hat{\mathbf{X}}$ given by,

$$D\left(\mathbf{X}, \hat{\mathbf{X}}\right) = \sum_{i,j} \mathbf{X}(i,j) \cdot \log \frac{\mathbf{X}(i,j)}{\hat{\mathbf{X}}(i,j)} - \mathbf{X}(i,j) + \hat{\mathbf{X}}(i,j)$$

The network is trained by applying a batch gradient descent training procedure with a batchsize of (...) and the parameters updated using the RMSProp algorithm (**RMSProp reference**), with a learning rate and momentum of (..) and (..) respectively.

2.2. Practical Considerations

Having developed the convolutional auto-encoder equivalent to convolutive NMF, we can now begin to understand the nature of the basis and activation matrices learned by the network. To do so, we train the convolutive auto-encoder defined by (7) on a simple toy example as shown in figure **??**(**Figure reference**).

3. EXTENSIONS TO INFINITE SUMMATION

4. SUPERVISED SOURCE SEPARATION

The problem of supervised source separation is solved as a two-step procedure [](**Paris supervised and semi-supervised**). The fist step of the procedure is to learn suitable models for a given source. We refer to this step as the training step. In the second step, we use these models to explain the contribution of the source in an unknown mixture. In section 2.1, we have developed the auto-encoder architecture to learn suitable convolutive models for a given source. We now turn our attention to the problem of using the models for separating the source in an unknown mixture.

The previous approach to source separation [](**ICASSP paper**) involves estimating the latent representation of the bases, which captures the contribution of the bases to each frame of the mixture spectrogram. This does not utilize the encoder component of the auto-encoder for the separation task. In this paper, we present a novel-separation scheme that utilizes the complete auto-encoder architecture for separation. We do so by using the following setup for separation. Given an input spectrogram $\mathbf{X}$, the auto-encoder produces an approximation that models the spectrogram in terms of its weights. We will denote to this approximation as,

$$\hat{\mathbf{X}} = Ae(\mathbf{X}|\theta) \tag{8}$$

Here, θ denotes the weights (parameters) of the auto-encoder. Thus, given the trained auto-encoders, i.e., given θ_1 and θ_2, the goal of separation is to identify suitable spectrograms $\mathbf{X}_1$ and $\mathbf{X}_2$ such that,

$$\mathbf{X}_m = Ae(\mathbf{X}_1|\theta_1) + Ae(\mathbf{X}_2|\theta_2) \tag{9}$$

Here, $\mathbf{X}_m$ represents the spectrogram of the mixture and $\mathbf{X}_1$, $\mathbf{X}_2$ denote the separated source spectrograms. In other words, the separation procedure attempts to estimate the mixture spectrogram as the sum of spectrograms of the individual sources. This assumption of additivity of spectrograms is similar to the separation procedure used in [](**NMF, Icassp**). However, we directly estimate the source spectrograms without estimating the latent representation. We train the network defined by (9) for an appropriate input instead of training for the weights of the network. As before, we minimize the KL divergence between mixture spectrogram $\mathbf{X}_m$ and its approximation $\mathbf{X}_1 + \mathbf{X}_2$. Conceptually, this problem is not different to training a neural network. The equivalence can be seen by applying a transposition to the auto-encoder definitions in (7). This also allows a generalized separation procedure to be applied even when the underlying architectures of the auto-encoders are changed.

Having obtained the contributions of the sources (separated spectrograms), the next step is to transform these spectrograms back into the time domain. This is given as,

$$x_i(t) = \text{STFT}^{-1}\left(\frac{\mathbf{X_i}}{\sum_i \mathbf{X}_i} \odot \mathbf{X}_m \odot e^{i\Phi_m}\right) \text{ for } i \in \{1,2\} \tag{10}$$

Here $x_i(t)$ denotes the separated speech signal in time and Φ_m represents the phase of the mixture and STFT^{-1} is the inverse short-time Fourier transform operation that transforms the complex spectrogram into its corresponding time domain representation. Also, $\odot$ represents the element-wise multiplication operation and the division is also element-wise.

5. EXPERIMENTS

We now describe the experimental setup used to evaluate our auto-encoder based convolutive audio models. We do so by comparing the separation performance of these models to the neural network based models proposed in [](**ICASSP paper**). For a uniform experimental setup, we apply a separation scheme described in 4 to both the models. We compare the performance in terms of median BSS_eval metrics [](**BSS eval paper**) viz., signal-to-distortion (SDR), signal-to-interference (SIR) and signal-to-artifact ratio (SAR) parameters. We also compare the median intelligibility scores in terms of the STOI index [](**STOI paper**). To compare these models, we use the TIMIT corpus [] (**Timit reference**). To form these mixtures, we randomly select a pair of male-female speakers from the TIMIT corpus. Of the 10 utterances available for each speaker, one utterance is randomly selected for each speaker. These two selected utterances are mixed at $0dB$ to generate the testing mixture. For the evaluation, we generate 20 such mixtures and compare the models for different parameter configurations. As the pre-processing step, we apply a 1024 point short-time Fourier transform representation with a hop of 25%. The magnitude spectrogram is then given as an input to the network.

We perform these evaluations at multiple parameter configurations with varying values for decomposition rank and filter size.

*** Talk about how you vary or select the values of K and T***

5.1. Results and Discussion

6. CONCLUSION

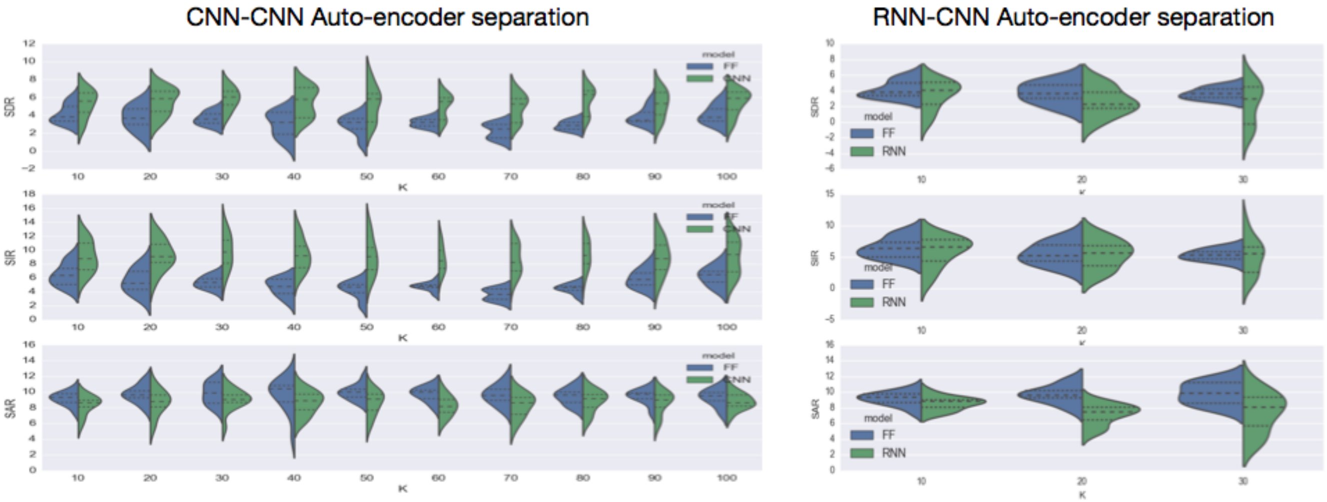
CNN-CNN Auto-encoder separation
SDR
12
10
8
6
4
2
0
−2
10
20
30
40
50
60
70
80
90
100
K
model
FF
CNN
SIR
18
16
14
12
10
8
6
4
2
0
K
model
FF
CNN
SAR
16
14
12
10
8
6
4
2
0
K
model
FF
CNN
RNN-CNN Auto-encoder separation
SDR
10
8
6
4
2
0
-2
-4
-6
model
FF
RNN
10
20
30
K
SIR
15
10
5
0
-5
model
FF
RNN
K
SAR
16
14
12
10
8
6
4
2
0
model
FF
RNN
K

Fig. 2. ...