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

We present a convolutiional auto-encoder that can act as a neural network equivalent to convolutive audio models. We demonstrate the ability of the network to learn optimal bases that resemble phoneme magnitude-spectra and also show how it can be used for supervised separation of speech mixtures. In terms of separation performance, the ability of the network to learn cross-frame characteristics from input spectrograms enables these models to significantly outperform their feed-forward counterpart. More importantly, we can now exploit the available diversity of neural networks to propose novel and potentially powerful extensions to existing auto-encoder architectures, based on a cascade of recurrent and convolutional neural network layers.

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

1. INTRODUCTION

Non-negative matrix factorization (NMF) of magnitude-spectrograms has been a very popular method for modeling sources for supervised source separation applications [1, 2]. NMF factorizes a 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 NMF on audio spectrograms. In the audio 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. **Cem: I am not sure how smooth the transition to neural nets here.** As shown in [3], 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,

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

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

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 network-parameters (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 [4] proposed a convolutive version to NMF (conv-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.

Developing a neural network alternative to conv-NMF also results in a significant advantage in terms of model reusability. Several neural network architectures have been recently proposed for supervised source separation [5, 6]. These networks are trained to learn discriminative audio models by training them on mixture spectrograms. Thus, the networks learn a decomposition that works only for a specific source-interference pair. Developing an alternative to conv-NMF allows us to learn audio-models by training neural networks on clean source spectrograms. These models can be easily re-used irrespective of the interfering component. 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 sepa-

Thanks to XYZ agency for funding.

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

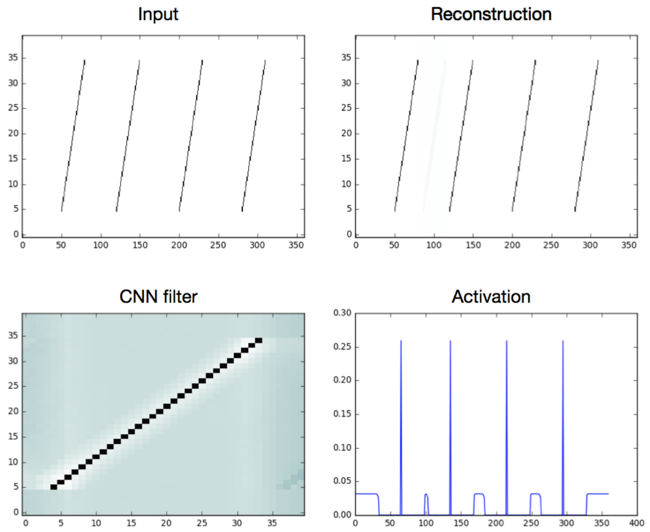

Fig. 1. Basis decomposition of a toy-illustration obtained using a CNN-CNN auto-encoder.

ration. We evaluate these models in terms of their separation performance in section **??** and conclude in section **??**.

2. NON-NEGATIVE CONVOLUTIIONAL AUTO-ENCODERS

2.1. Network Architecture

The convolutive NMF model [4] 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. We can interpret this operation as a two-layer convolutional auto-encoder (CAE) as follows,

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

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

subject to non-negativity of $\mathbf{W}_i$ and $\mathbf{H}$. Here, we assume that the convolutional layer filters $\mathbf{W},\ \mathbf{W}^{\ddagger}$ have a size of $M \times T$ where, T represents the depth of the convolution and M denotes the height of the input matrix $\mathbf{X}$. 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 neural network (CNN) 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 CNN layer generates an approximation of the input from the code and will be referred to as the "decoder". We can satisfy the non-negativity constraints by incorporating a non-linearity into the definitions of the encoder and the decoder. Thus,

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

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

Here, the $g(.) : \mathbb{R} \rightarrow \mathbb{R}^{\geq 0}$ applies an element-wise non-linearity and ensures that the activation matrix and the reconstruction are non-negative. In our experiments, we used the soft-plus function which is given by the formula $SP(x) = \log(1 + \exp(x))$ as the non-linearity. Based on (7), we now note some key points about the CAE. (i) The output of the encoder gives the latent representation (analog of activations in NMF) 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. To train the auto-encoder, we minimize the KL-divergence between the input spectrogram $\mathbf{X}$ and its reconstruction $\hat{\mathbf{X}}$ given by,

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

Although the filters can assume negative values, the use of a non-linearity does not allow cross-cancellations across the basis elements.

2.2. Practical Considerations

Having developed the CAE equivalent to conv-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 **??**. We also incorporate sparsity constraints on the activation, i.e., the output of the first CNN. In this example, the CNN filter resembles a snippet of the input spectrogram. We also see that the activation comprises a series of impulse trains. Thus, the encoder acts as a matched-filter and identifies the points in time when the corresponding

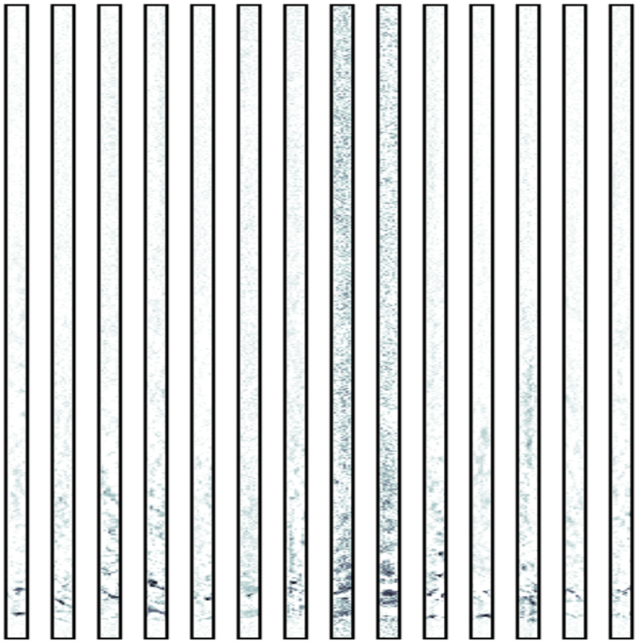

Fig. 2. A subset of decoder filters obtained by training the CAE on magnitude-spectrograms of utterances of a male speaker. This decomposition is obtained for the configuration $K = 80$ and $T = 8$. We see that the filters resemble snippets of a speech spectrogram.

pattern becomes active. As shown in (7), the time-frequency pattern is captured by the filters of the decoder. Given the nature of the activation we see that the encoder attempts to learn the inverse filter to the decoder. Figure 2 shows the decoder filters obtained by training the CAE on speech utterances of a male speaker. Similar to the previous toy-example, the decoder filters learn patterns that resemble snippets of a speech spectrogram.

The decoder CNN can be interpreted as a finite impulse response (FIR) filter. The encoder attempts to approximate the inverse of the decoder FIR filter as an FIR filter. From our knowledge of signal processing, it can be shown that the inverse filter of an FIR filter has an infinite impulse response (IIR) [] (**Give reference**). Thus, approximating the encoder as an FIR filter or a CNN will hinder the decoder from learning patterns that are invertible only by an IIR filter. One such example is shown in figure **??**. Therefore, we explore using a Recurrent Neural Network (RNN) to get an IIR encoder.

3. USING AN INFINITE IMPULSE RESPONSE (IIR) FILTER IN THE ENCODER

In this section, the goal is to construct a recurrent encoder analogous to the convolutive encoder we discussed in the previous section. The potential gain of using a recurrent encoder over a finite length convolutive encoder is due to the fact that a recurrent filter in theory can capture arbitrarily long temporal dependencies. From a signal processing point of view, the motivation for going with a recurrent filter is that, the inverse of an finite length filter (the convolutive basis in the encoder) is given by a recurrent filter. We explore the idea, and obtain source separation results on the same dataset.

The way we go about building an encoder is by passing the input through K separate recurrent neural networks (RNNs) (Note that K was the number of filters/components in the convolutive model). The k'th RNN recursion in the encoder is given by the following equation:

$$\mathbf{Z}(k_1, t, k) = \tanh\left(\sum_{k_2=1}^{K_{in}} \mathbf{W}^k(k_1, k_2)\mathbf{Z}(k_2, t-1, k) + \sum_{l} \mathbf{U}^k(k_1, l)\mathbf{X}(l, t) \right),\ k \in \{1, \ldots, K\}, \quad (8)$$

where $\mathbf{Z}(:, t, k) \in \mathbb{R}^{K_{in}}$ denotes the latent state vector of the k'th RNN at time t. We denote the hidden state dimensionality of each RNN with K_{in}. The recurrent and projection matrices of the $k'th$ RNN are respectively denoted with $\mathbf{W}^k$, and $\mathbf{U}^k$. Note that although the given recursion corresponds to the vanilla-RNN architecture, there is no restriction on the RNN architecture choice. In our experiments, we have used the LSTM architecture [cite]. After going through the RNN recursions, the encoder output $\mathbf{H}(i, t)$ is obtained by summing the RNN outputs over the first dimension:

$$\mathbf{H}(i, t) = \sum_{k_1=1}^{K_{in}} \mathbf{Z}(k_1, t, i) \quad (9)$$

4. SUPERVISED SOURCE SEPARATION

The problem of supervised source separation is solved as a two-step procedure [7]. 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 [3] using neural-network based audio-models 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.

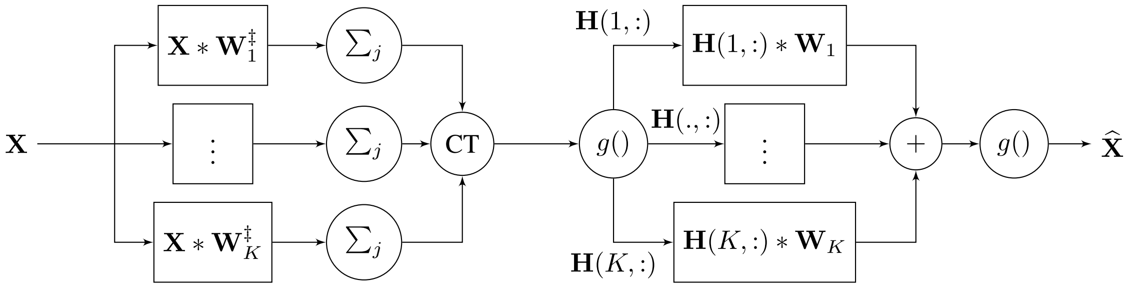
X
$\mathbf{X} * \mathbf{W}_1^{\ddagger}$
⋮
$\mathbf{X} * \mathbf{W}_K^{\ddagger}$
$\sum_j$
$\sum_j$
$\sum_j$
CT
$g()$
$\mathbf{H}(1,:)$
$\mathbf{H}(.,:)$
$\mathbf{H}(K,:)$
$\mathbf{H}(1,:) * \mathbf{W}_1$
⋮
$\mathbf{H}(K,:) * \mathbf{W}_K$
+
$g()$
$\widehat{\mathbf{X}}$

Fig. 3. Block Diagram of CNN-CNN Autoencoder

Given an input spectrogram $\mathbf{X}$, the auto-encoder produces an approximation of the input spectrogram which is a linear combination of its weights. We will denote to this approximation as,

$$\hat{\mathbf{X}} = Ae(\mathbf{X}|\theta) \tag{10}$$

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

$$\mathbf{X}_m = Ae(\mathbf{X}_1|\theta_1) + Ae(\mathbf{X}_2|\theta_2) \tag{11}$$

In this equation, $\mathbf{X}_m$ represents the spectrogram of the mixture and $\mathbf{X}_1$, $\mathbf{X}_2$ denote the separated source spectrograms. Thus, similar to NMF, this approach assumes that the magnitude-spectrogram of the mixture is the sum of magnitude-spectrograms of the underlying sources. However, in this separation procedure, we directly estimate the source magnitude-spectrograms without estimating the latent representation. To do so, we train the network defined by (11) for an appropriate input $\mathbf{X}_1$, $\mathbf{X}_2$, 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,

$$x_i(t) = \text{STFT}^{-1}\left(\frac{\mathbf{X_i}}{\sum_i \mathbf{X}_i} \odot \mathbf{X}_m \odot e^{i\Phi_m}\right) \text{ for } i \in \{1,2\} \tag{12}$$

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 construct a set of training and test examples using the TIMIT corpus [8] for the evaluation. To form the examples, 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 0 dB to generate the testing mixture. The remaining 9 utterances are used as training data to construct models for the sources. In other words, these examples are used to train the respective (convolutional/ feed-forward) auto-encoders. For the evaluation, we generate 20 such mixtures and compare the models for different parameter configurations. As a 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.

The neural networks are initialized using the Xavier initialization scheme [9]. The networks are trained by applying a batch gradient descent training procedure and the parameters updated using the RMSProp algorithm [10], with a learning rate and momentum of 0.001 and 0.7 respectively.

The CNN filters are selected to be 512-point tall and 8-point wide. Thus, the convolutions are performed only along the time axis. The number of CNN filters also decides the number of components in the decomposition. We evaluate these models over a varying number of CNN filters ranging from 10 to 100 uniformly in steps of 10. We compare the separation performance in terms of median BSS_eval metrics [11] viz., signal-to-distortion (SDR), signal-to-interference (SIR) and signal-to-artifact ratio (SAR) parameters.

5.1. Results and Discussion

Figure 4 gives the separation performance for the CNN-CNN models (left) and RNN-CNN models (right) for varying values of number of filters K. We evaluate the proposed CAE models by comparing the separation performance to their equivalent feed-forward (FF) counterparts proposed in [3].

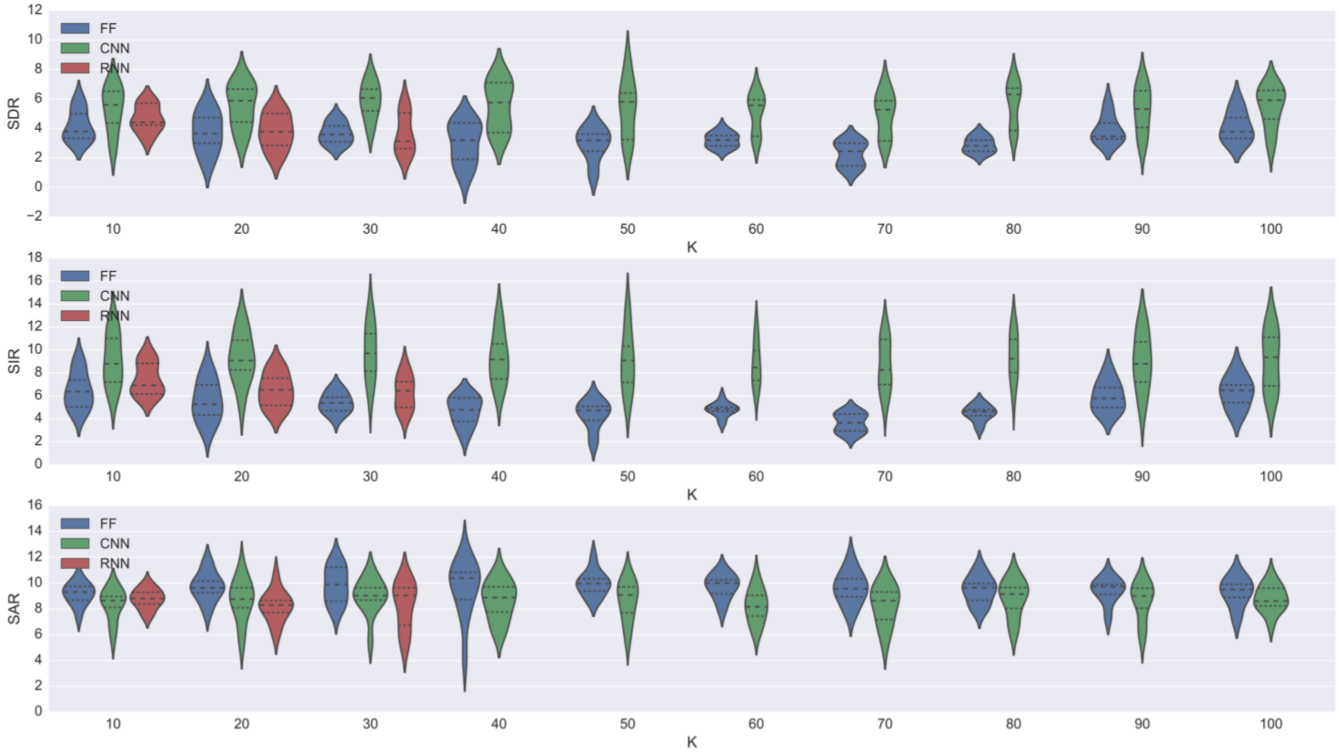

Fig. 4. Separation performance of convolutive models obtained using a CNN-CNN Auto-encoder for varying values of K.

In order to maintain a uniform experimental setup, we apply the separation scheme described in 4 to all the models. We plot the results in terms of a violin plot that also indicates the median value and the corresponding inter-quartile range for each K.

We see that the CAE models significantly out-perform their corresponding FF versions. This can be seen from the fact that the inter-quartile range in SDR for CAE models is higher than the inter-quartile range of corresponding FF models for several values of K. This improvement is a consequence of reduced interference generated by these models in source separation (See SIR plots). Although the SAR for CAE models degrades slightly as compared to FF models, the improvement in significant improvement in SIR compensates for this loss. The performance of the models achieves a peak value for $K = 80$. However, the separation performance does not degrade significantly for other values of K. Thus, the choice of K does not appear to be a very critical consideration for these models.

The convolutive speech models obtained by training the RCAE also outperform the FF auto-encoder models, as seen by the median values and inter-quartile ranges. This improvement in performance is not as pronounced as the CAE models for the case of speech mixtures. However, the experimentation serves as a proof of concept that exploring non-uniform auto-encoder architectures could lead to interesting and potentially powerful algorithms for several applications.

6. CONCLUSION

In this paper, we developed and investigated the use of a convolutional auto-encoder as an alternative to learn convolutive basis decompositions of speech and audio signals. The ability of the networks to include temporal dependencies allow the auto-encoders to learn cross-frame structures in the input spectrogram. We demonstrated that this results in a significant improvement in separation performance as compared to feed-forward auto-encoder models. This approach also allows for several extensions and generalizations to convolutive audio models, that can be easily implemented using the available diversity of neural networks. One such extension considered in this paper is the use of auto-encoders formed by a cascade of recurrent and convolutive layers. Although these models are not as potent as CAE models for separation of speech mixtures, these extensions could lead to interesting non-uniform auto-encoder architectures for several applications.

7. REFERENCES

[1] Paris Smaragdis, Cedric Fevotte, Gautham J Mysore, Nasser Mohammadiha, and Matthew Hoffman, "Static and dynamic source separation using nonnegative factorizations: A unified view," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 66–75, 2014.

[2] Tuomas Virtanen, Jort Florent Gemmeke, Bhiksha Raj, and Paris Smaragdis, "Compositional models for audio processing: Uncovering the structure of sound mixtures," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 125–144, 2015.

[3] Paris Smaragdis and Shrikant Venkataramani, "A neural network alternative to non-negative audio models," *CoRR*, vol. abs/1609.03296, 2016.

[4] Paris Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.

[5] Emad M Grais and Mark D Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," *arXiv preprint arXiv:1703.08019*, 2017.

[6] Shrikant Venkataramani and Paris Smaragdis, "End-to-end source separation with adaptive front-ends," *arXiv preprint arXiv:1705.02514*, 2017.

[7] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," *Independent Component Analysis and Signal Separation*, pp. 414–421, 2007.

[8] John S Garofolo, Lori F Lamel, Jonathan G Fiscus William M Fisher, David S Pallett, Nancy L Dahlgren, and Victor Zue, "Timit acoustic phonetic continuous speech corpus," 1993.

[9] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks.," .

[10] Tijmen Tieleman and Geoffrey Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.

[11] Cédric Févotte, Rémi Gribonval, and Emmanuel Vincent, "Bss_eval toolbox user guide–revision 2.0," 2005.