

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.

as a non-negative auto-encoder in the following manner,

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

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

1. INTRODUCTION

Recently, 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 components. 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 generative audio models for supervised source separation is the use of Non-negative matrix factorization (NMF). Non-negative matrix factorization (NMF) matrix of non-negative elements $\mathbf{X} \in \mathbb{R}_{M \times N}^{\geq 0}$ as a product of the basis matrix \mathbf{W} and the activation matrix \mathbf{H} . The notation $\mathbb{R}_{M \times N}^{\geq 0}$ represents the set of matrices of non-negative elements of size $M \times N$. In this factorization, the basis matrix $\mathbf{W} \in \mathbb{R}_{M \times r}^{\geq 0}$, the activation matrix $\mathbf{H} \in \mathbb{R}_{r \times N}^{\geq 0}$ 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 [1] (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

Here, \mathbf{X} represents the input spectrogram, \mathbf{W}^\dagger represents a form of pseudo-inverse of \mathbf{W} and $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{\geq 0}$ 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 enabled 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 net equivalent are unable to explicitly utilize these cross-frame patterns available in a spectrogram. To alleviate this drawback, Smaragdakis [2] (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.

2. NON-NEGATIVE CONVOLUTIONAL AUTO-ENCODERS

2.1. Network Architecture

The convolutive NMF model [1] (Smaragdis convolutive speech bases) approximates a non-negative matrix $\mathbf{X} \in \mathbb{R}_{M \times N}^{\geq 0}$ as,

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

Here, $\mathbf{W}_i \in \mathbb{R}_{M \times T}^{\geq 0}$ acts as the i^{th} basis matrix and $\mathbf{H} \in \mathbb{R}_{r \times N}^{\geq 0}$ 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,

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

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

subject to non-negativity of \mathbf{W}_i and $\mathbf{H} \geq 0$. Here, we assume that the convolutional filters \mathbf{W} , \mathbf{W}^{\dagger} 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,

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

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

Here, the non-linearity $g(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{\geq \kappa}$ 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.

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,

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

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.

The approximation \hat{X} for a given spectrogram X is computed as follows:

$$\begin{aligned} \hat{H}(k, t) &= \sigma_1 \left(\sum_{f, t'} X(f, t - t') F_e(f, t', k) \right) \\ \hat{X}(f, t) &= \sigma_2 \left(\sum_k \sum_{t'} \hat{H}(k, t - t') F_d(f, t', k) \right) \end{aligned} \quad (8)$$

3. EXTENSIONS TO INFINITE SUMMATION

4. SUPERVISED SOURCE SEPARATION

Describe the separation procedure.

5. EXPERIMENTS

Try each model in a given K range in speech-speech source separation task.

5.1. Experimental setup

5.2. Results and Discussion

6. CONCLUSION