Speech and Speaker Recognition Project
Source separation using Deep Neural Networks

Corentin Abgrall, Marc Beillevaire, Eric Masseran

Speech and Spearker Recognition, Department of Computer Science
KTH, Sweden

Abstract

Speech separation is one of the most useful and challenging problem in sound processing and has applications to many fields. And with the advent of fast parallel computations it becomes now easier to use Deep Learning to solve such complicated problems. In this paper we focus on male and female voice separation, and provide a method to split one source into two different ones. Several deep neural networks are implemented and tested on some mixed digits from the TIDIGITS

1. Introduction

The methods used in the last laboratory to solve the speech recognition problem on the TIDIGITS dataset are efficient and relatively easy to implement. The results of this model are summed up in the figure 1.

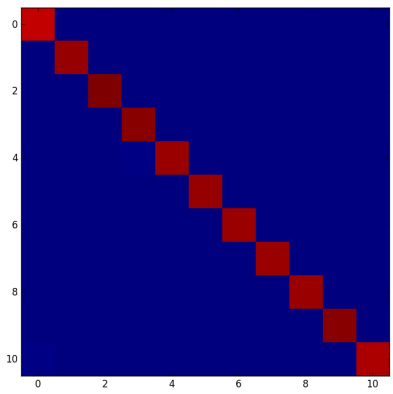

Figure 1: Confusion matrix on digit recognition with the original TIDIGITS dataset

The correlation matrix has a strong diagonal and very few insertions and deletions. But the robustness of those models are not as high as we could expected. Indeed these results have been obtained with very good recordings. But the model is unable to understand correctly the digits when for instance two persons are speaking.

Therefore, we will now try to differentiate the voices when two different persons are speaking. Several papers already deal with solving the source separation problem. They applied their solution on songs and wanted to remove the lyrics from them [1]. They used a deep recurrent neural networks on their database which is composed of the sounds produced by several voices and instruments. But other papers [4] [2] [5] also presented some various implementations of this solution applied on different datasets. We decided to focus our efforts on the implementation proposed in the paper [4], and try to separate the

voices of one man an one woman.
Therefore, our goal was to apply the paper's method to the TIDIGIT dataset. In this paper we will explain how the database has been created, the solution based on the deep neural network and finally present some results.

2. Method

2.1. Separate source

Our problem is only restricted to the source separation starting from one input source. We want to split the source into two targets, one being the man's speech and the other being the woman's. In order to split a source into two targets, we use the masking technique. A mask M is applied on the Fourier transform of the mixture (x) in order to choose which frequencies to keep and which one to crop.

$$\begin{aligned} t_s &= M_s * x \\ t_n &= M_n * x \end{aligned} \tag{1}$$

There are two types of mask:

- Binary mask (value: 0 or 1)
- Soft mask (value: $0 \leq v \leq 1$)

2.2. Performance of source separation

There are three measures that are used to define the performance of sound separation [3]:

- Source to Interference Ratio (SIR)
- Source to Artifacts Ratio (SAR)
- Source to Distortion Ratio (SDR)

The interferences define the presence of the other sources not wanted inside the targeted one. The artifacts are the transformation in the separated source created by the algorithm of source separation. Finally, the distortion is the global performance between the predicted source and the targeted source including all the other previous deformations. The better the quality of the separation is, the bigger are the values measured.

3. Neural Network

3.1. Architecture

This network is a Recurrent Neural Network and implement this paper's architecture: [4].

The paper actually uses a recurrent network. In RNNs, the weights for each hidden layer l are computed using both the input $\mathbf{x_t}$ of the layer and the output of this layer for the previous training example $h^{(l)}(\mathbf{x_{t-1}})$:

$$h^{(l)}(\mathbf{x_t}) = f\left(\mathbf{W}\, h^{(l-1)}(\mathbf{x_t}) + \mathbf{b} + \mathbf{U}\, h^{(l)}(\mathbf{x_{t-1}})\right)$$

where $\mathbf{W}$ and $\mathbf{U}$ are matrices of weights for layer l, and $\mathbf{b}$ a bias vector. In the classical dense networks the time parameter $\mathbf{U}$ is equal to zero.

The network architecture features the following layers:

- One input layer
- Several hidden layers with variable number of nodes
- One output layer with size 1024, which is twice the length of the Fourier transforms

This networks takes as input the short time Fourier transforms (STFT) of the signal.

The output are masks to apply to each frequency of the input signal. Once computed, half of the mask can be applied on the initial signal to get the man's voice and the other half to get the woman's voice from the input.

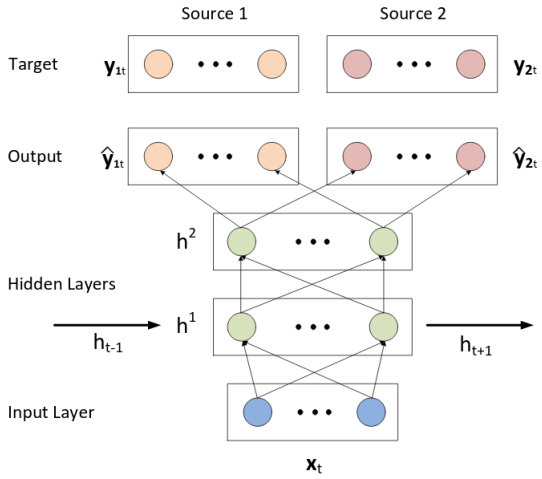

Figure 2: RNN architecture, with source separation at output [4]

3.2. Last Layer

The last layer outputs two spectrums that we wish to apply to the input STFT to separate both male voice and female voice in the input file. But we also need to ensure that these two masks sum to the original signal.

To check that, our network doesn't use directly these spectrums $\hat{\mathbf{y}}_{1,t}$ and $\hat{\mathbf{y}}_{2,t}$. For the binary mask, the output masks are really easy to implement. We set 1 to the bigger coefficient between $\hat{\mathbf{y}}_{1,t}$ and $\hat{\mathbf{y}}_{2,t}$ and 0 to the other one:

$$\mathbf{M}_{1,t}(f) = \begin{cases} 1 & \text{if} \quad \hat{y}_{1,t}(f) > \hat{y}_{2,t}(f) \\ 0 & \text{else} \end{cases}$$

$$\mathbf{M}_{2,t}(f) = \begin{cases} 1 & \text{if} \quad \hat{y}_{2,t}(f) > \hat{y}_{1,t}(f) \\ 0 & \text{else} \end{cases}$$

For the softmask technique, we need to resize the output depending on the sum of $\hat{\mathbf{y}}_{1,t}$ and $\hat{\mathbf{y}}_{2,t}$.

$$\mathbf{M}_{1,t}(f) = \frac{\hat{y}_{1,t}(f)}{\hat{y}_{1,t}(f) + \hat{y}_{2,t}(f)}$$

$$\mathbf{M}_{2,t}(f) = \frac{\hat{y}_{2,t}(f)}{\hat{y}_{1,t}(f) + \hat{y}_{2,t}(f)}$$

They are then applied by computing the element-wise product with the input vector (the STFT).

Our last layer thus computes the masked spectrum of the input, and compares it to the target spectrum of both the clean voice and the noise to obtain the cost.

3.3. Cost function

The cost function takes as input the prediction of the neural network $(\hat{y}_{1,t}, \hat{y}_{2,t})$ and the data from the original sources $(y_{1,t}, y_{2,t})$. The output of the networks is a mask for the two sources. The output of those masks applied to the input x_t are $(\hat{y}_{1,t}, \hat{y}_{2,t})$.
The cost function needs to reflect how far are those outputs from each other. Therefore, the first part D_{direct} of the cost function is simply the euclidean distance between those two elements. However, it is also good idea to add the crossed distances $D_{crossed}$ between $||\hat{y}_{1,t} - y_{2,t}||_2^2$, indeed by subtracting this term we want to minimize the similarities between the sources and the others sources. Therefore the cost function can be written as :

$$Loss = D_{direct} - \gamma D_{crossed}$$

Where

$$D_{direct} = ||\hat{y}_{i,t} - y_{i,t}||_2^2 + ||\hat{y}_{2,t} - y_{2,t}||_2^2$$

$$D_{crossed} = ||\hat{y}_{1,t} - y_{2,t}||_2^2 + ||\hat{y}_{2,t} - y_{1,t}||_2^2$$

and γ is a parameter to increase or reduce the penalty of the crossed term.

4. Experiment

4.1. The data

The original TIDIGITS database is an audio dataset of more than 5000 elements. The sounds are simply recorded digits. The speaker is half of the time a man or a woman.

We have split the dataset in two equal parts of 2500 samples. One is used as a training set and the other is used as a test set.

The input is a set of wav files obtained by mixing one man and one woman speaker file from the Tidigits dataset, converted into shrottime Fourier transforms. And the output is a target spectrum of the two separated voices.

Our output represent therefore twice the amount of data since there are two target spectrums.

4.2. Tests performed

We applied the data described above to our RNN, and performed several tests to see how performances could be increased.

We tried several numbers of hidden layers in the network, and the number of node per layer.

Finally we compare our RNN to a classical dense neural network, to see which one has better performances on our dataset. We focus first on a RNN since speech is continuous and the precedent frames is strongly correlated with the current one. We want to see if the previous frames of the samples can help the network to find a better mask.

4.3. Results

Our results shows the input (mixed) spectrum, and the two output spectrums. These two spectrums are the sum to the input, and we can clearly see the pronounced digits in the spectrum. The output spectrum is still a bit different from the input but the correlation is quite important.

To test the accuracy of these results, we can compute several values:

- The sound-distorsion ratio (SDR) indicates how different the output sound is from the original one
- The sound-interference ration (SIR) show how the second source influence the first output sound
- And the sound-artifact ratio (SAR) indicates the amount of noise introduced by the algorithm itself.

Fig. 4 is a summary of all the tests we computed and the signal/noise ratios we got.

Several models were tested. First a recurrent network has been compared to a classical dense neural network. We also changed the number of hidden layers (each *HL* value on fig. 4), and the number of nodes per hidden layer. We finally tried using the sigmoid instead of the rectified linear unit.

These different ratios quite reflect what we can hear in the results: in each separated source, we hear moslty the target voice, but there is still some residues of the other voice in it which is considered as noise when performing the benchmark.

Finally, we applied our inital GMM-HMM speech recognition algorithm to see whether recognition can be performed on the separated sources. It turns out that the results are quite good, though a bit weaker than on the pure TIDIGITS dataset of course.

The soft mask seem to work better than the binary mask. This was expected since the binary mask either cuts or keeps frequencies, while the soft mask keeps a certain percentage of each frequency. This results on a more rough separation with the binary mask, and a more precise with the soft mask.

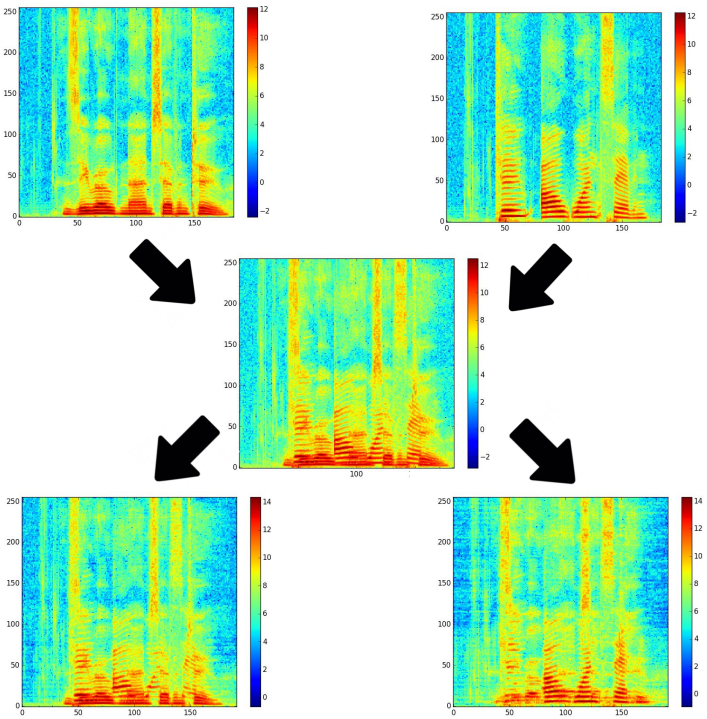

Figure 3: Spectrum input above, mix with both male and female voices, and below the separation computed by the network

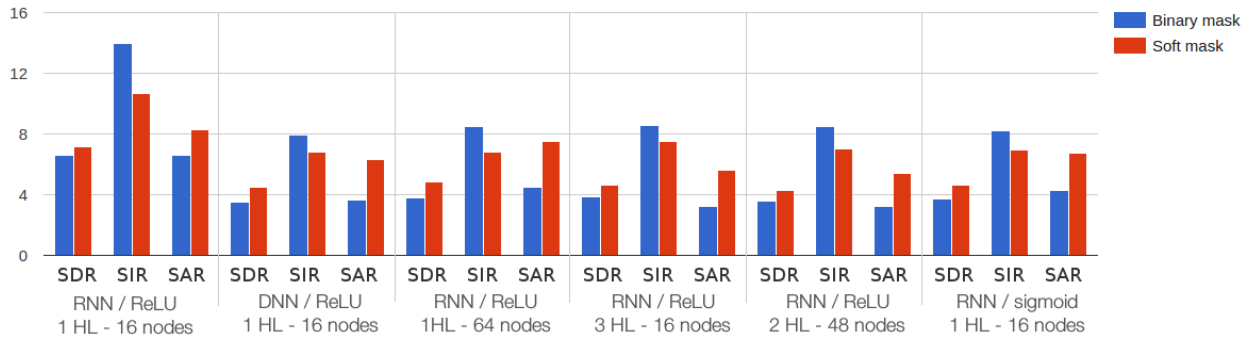

Figure 4: Our results on several types of networks

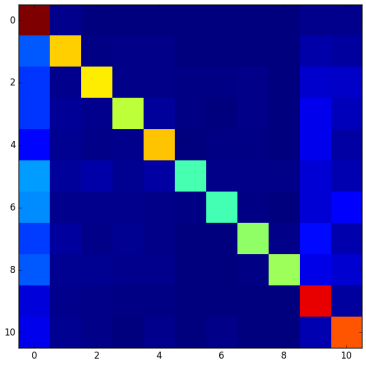

Figure 5: Recognition performance with a binary mask

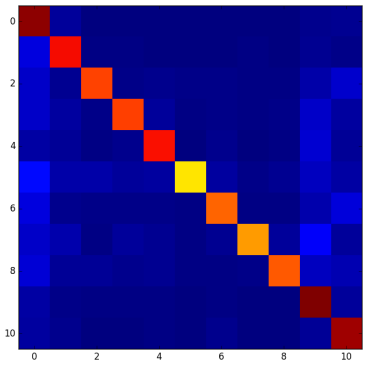

Figure 6: Recognition performance with a soft mask

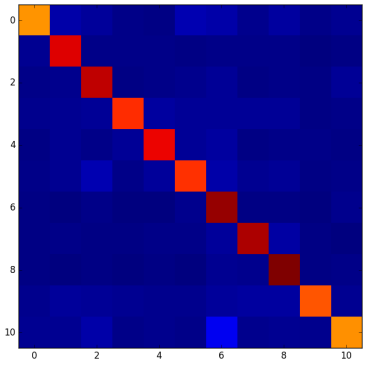

Figure 7: Recognition performance with a soft trained mask

5. Conclusion

In this work, we applied a solution used for source separation presented in the paper [4] to our own dataset. This dataset is a modified version of the TIDIGITS, we have mixed the files containing a male voice and a female voice and used them for training the networks. This method uses a recurrent neural network or a fully connected deep neural network. The training was involving a loss function computing the mean square error of the input signal and the reconstituted signal created with the mask produced by the network and applied to the original input. Several metrics has been used to measure the results.

The results are satisfactory because the method is able to separated the two sources even if there are still some residues coming from the other source. To know in what extend these residues are important, we used the outputs of the RNN to feed a GMM-HMM model and observe that the model was able to understand and recognize the words pronounced. However, the results were slightly better with the original TIDIGITS dataset.

Therefore this method is interesting and a promising solution for source separation which might be improved in several ways : by increasing the time memory or increasing the size of the network.

6. References

[1] Mark D. Plumbley Andrew J.R. Simpson, Gerard Roma. *Deep Karaoke: Extracting Vocals from Musical,Mixtures Using a Convolutional Deep Neural Network*. Centre for Vision, Speech and Signal Processing, University of Surrey Guildford, UK, 2015.

[2] Hakan Erdogan Emad M. Grais, Mehmet Umut Sen. *Deep neural networks for single channel source separation*. Faculty of Engineering and Natural Sciences, Sabanci University, Orhanli Tuzla, 34956, Istanbul.

[3] Rmi Gribonval Emmanuel Vincent and Cdric Fvotte. *Performance Measurement in Blind Audio Source Separation*. EEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 14, NO. 4, 2006.

[4] Mark Hasegawa-Johnson Paris Smaragdis Po-Sen Huang, Minje Kim. *Singing-voice separation from monaural recordings using deep recurrent neural networks*. Department of Electrical and Computer Engineering and Department of Computer Science, University of Illinois at Urbana-Champaign, USA.

[5] Andrew J.R. Simpson. *Probabilistic Binary-Mask Cocktail-Party Source Separation in a Convolutional Deep Neural Network*. Centre for Vision, Speech and Signal Processing, University of Surrey, UK.