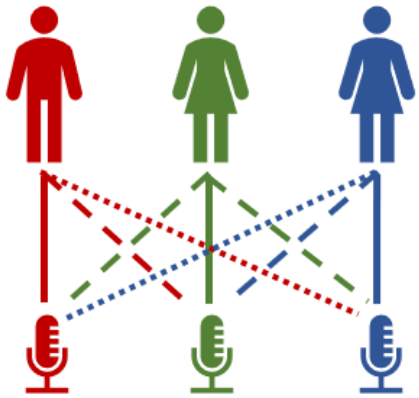
The Cocktail Party Problem

CS 419: Intro to Machine Learning

Aditya Sriram
Jujhaar Singh
Raavi Gupta
Rishabh Ravi
Vedang Gupta

CS 419 Course Project under
Prof. Abir De



Department of Computer Science and Engineering
IIT Bombay

Contents

1 Introduction

This project is inspired by the famous Cocktail Party Effect - the ability of our brain to focus on one stimulus in the presence of many similar stimuli. The effect is apparent when one tries to have a conversation with someone in a noisy environment. The Cocktail Party Problem is an example of Source Separation (also known as Blind Source Separation) which involves separating a set of source signals from a mixed-signal with no information about the sources.

Earlier attempts to solve the problem generally assumed that the number of speakers is fixed while other speech separation methods separate speech from background noise and not multi-speaker audio. Additionally, many of these methods addressed tasks with limited vocabulary and perform extremely well.

Deep Clustering aims to solve the more general problem of speaker-independent speech separation with a variable number of speakers with no constraints on vocabulary and grammar. Deep Clustering aims to learn embeddings for each input element which can be fed into a clustering algorithm to separate the audio track.

2 Background

In Deep Clustering, the raw signal x is processed by applying a Short time Fourier Transform with overlapping windows, creating a spectrogram X_n where n is a time-frequency index (t, f). We seek to learn an embedding $V = f_\theta(x)$ of unit norm using a Deep Neural Network for every time-frequency bin. It is assumed that in each bin, one of the sources dominates and being able to partition the spectrogram would allow us to create masks that when applied can give rise to the individual sources.

This requires labels $Y = \{y_{n,c}\}$, such that $y_{n,c} = 1$ if time-frequency bin X_n corresponds to speaker c. Defining the objective to be minimised,

$$C(\theta) = \|VV^T - YY^T\|_W^2 = \sum_{i,j:y_i=y_j} \frac{|v_i - v_j|^2}{d_i} + \sum_{i,j} \frac{\left(|v_i - v_j|^2 - 2\right)^2}{4\sqrt{d_i d_j}} - N$$

where $|A|_W^2 = \sum_{i,j} w_{i,j} a_{i,j}^2$ is a weighted Frobenius norm with $d_i = |\{j : y_i = y_j\}|$. Intuitively, the objective minimises the distance between embeddings belonging to the same partition. At test time, embeddings computed using the trained model are clustered used to generate masks required to separate the sources.

3 Dataset

LibriMix is an open source dataset for source separation in noisy environments. It is derived from LibriSpeech signals (clean subset) and consists of two- or three-speaker mixtures combined with ambient noise samples from WHAM! which is a noise audio dataset that was collected at various urban locations throughout the San Francisco Bay Area in late 2018.

4 Model

4.1 Pre - processing

To make computations faster and less expensive, the data was downsampled to 8kHz. The input fed to the model are the log spectral magnitude of the original raw signal, obtained via taking a STFT(short - time Fourier

transform(STFT)). It employs an encoder which transforms the input signal into a domain suitable for source separation and a decoder which transforms the extracted source signals back to the time domain. Using STFT, time-frequency(t-f) bins that correspond to each source can be separated. The window for computing STFT is taken to be of 32ms with a 8ms window shift.
Binary masks are employed to build the targets. For each t-f bin, for the source having the maximum magnitude the mask is set to 1, otherwise, 0.
Additionaly, the bins having magnitude $\leq -40dB$ were dropped to avoid training on regions that contain silence.

4.2 Model Architecture

The network consists of two BLSTM(Bidirectional Long Short-Term Memory) having 600 hidden cells each having a dropout rate of 0.3. Tanh was used as the activation function of the embedding layer. The outputs of the embedding layer are reshaped and normalised as per the requirements.

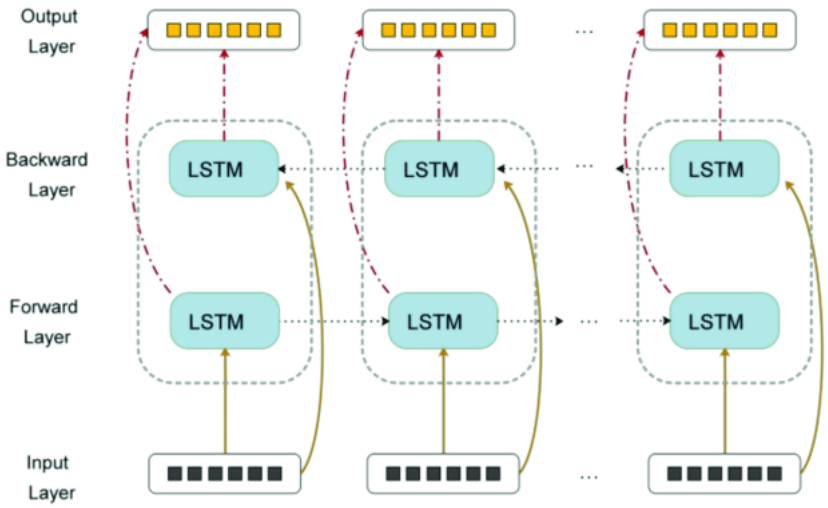


Figure 1: Architecture of a BLSTM Layer

KMeans Clustering is used as the clustering algorithm. The number of clusters are kept to be the number of speakers. The embedded layer output

is used for formulating the clusters.
For creating the masks, energy based voice activity detection is used. The masks are consequently applied and separated wav files are created using the masks.
While training, a deep clustering loss function is utilized and an SGD optimizer helps in reducing the overall loss and for improving the accuracy.

5 Discussion and Future Work

- Currently our model has been trained on clean dataset i.e. audio sources containing negligible noise. Future tasks include handling noises in the signal mixture.
- Further improvements in the project can be separating sound sources having unknown number of speakers.
- The current model employs STFT. Since STFT transforms the signal to frequency domain, the algorithm needs to deal with both the magnitude and phase of the signal. Because it is difficult to modify the phase, the method significantly modifies only the magnitude of STFT by calculating a t-f mask for each source. This puts an upper bound on the performance of the model. Several new methods have been proposed that carries computation in the time domain itself. Such methods also need to be explored.

6 References

https://www.merl.com/publications/docs/TR2016-003.pdf
https://arxiv.org/pdf/2005.11262.pdf
https://www.openslr.org/12
https://github.com/asteroid-team/asteroid
https://github.com/JorisCos/LibriMixFeatures
https://arxiv.org/pdf/1508.04306.pdf