EE603A: Course Project

Wasif Jawad Hussain
180882, wasif@iitk.ac.in

Sumit Kumar Pandey
180797, sumitkrp@iitk.ac.in

I. Introduction

In this group project, we have implemented, Audio Tagging(Task - 1) and Audio Event Detection (Task-2) as a part of the course project in EE603A. We perform the training by using raw audio files(wav files). For the testing purpose, we are supposed to test on the spectrogram data.

II. Quick Overview

Given a spectrogram of raw audio signal we first detect activity segments. Activity segments are part of audio clips which have any sound and are not silent. Once we detect activity segment we apply any one of our many classifiers to identify the class of audio present in each frame of the segment. This is followed by a max pooling and we report the dominant class of that segment along with the onset and offset time.

III. Methodology

A. Data Generation

We are given spectrogram of 12 audio samples each of which contain either music, speech or both. Some of the audios have intrinsic noise. We are given onset and offset time along with the label class for different activities in the audio clip.

We also created our very own dataset curating speech and music clips and labelling them manually.

The spectrogram have been made such that there are 513 features per each frame. The audios have been zero padded to ensure that the total number of frames are fixed at 313 for every 10 sec audio clip.

To map the onset and offset time which we are given in time domain to the discrete domain, we use the following expression

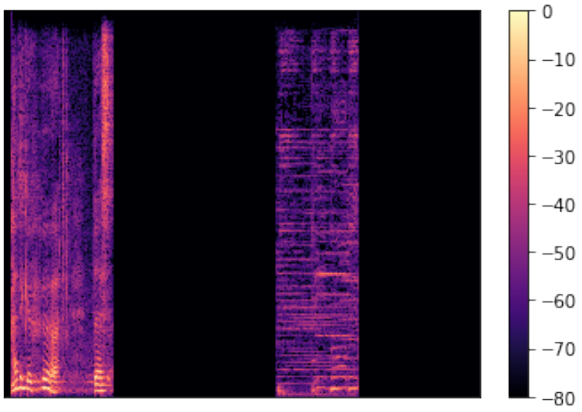
$$t = \frac{\{(w-1) \times h + w\}}{f_s} \tag{1}$$

In the above expression f_s denotes the sampling frequency, w denotes the window length and h denotes the hop size

Using the above expression we create a label array for each frame of the spectrogram. Each label is a one hot vector of length 3 corresponding to the three different classes, viz music, sound, speech.

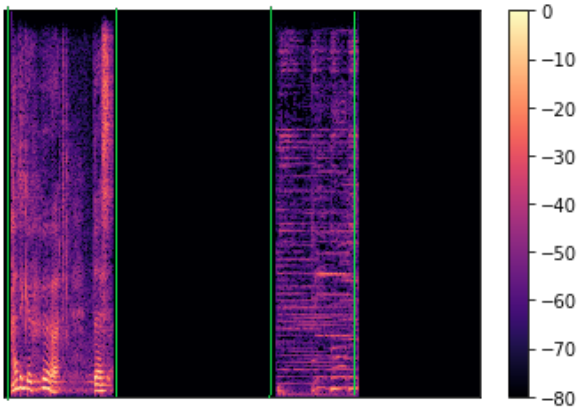
B. Audio Activity Detection

Fig. 1. Spectrogram of a raw audio signal



The spectogram data can be used to detect energy levels in various sections of the audio clip. Firstly we sum the energy of all the harmonics for a given frame. Where this sum of energy is lower than a threshold we classify it as silent frame, else it is labelled as non-silent frame. Then we use another algorithm to fine tune our previous labelling. If a few silent frames are present in between many non-silent frames we label the entire bunch as non-silent. This ensures that our activity segments don't explode in number and weak segments are grouped together to form more confident larger intervals. This algorithm gave us $2-3$ activity centers for any audio clip of 10 sec duration, which is highly consistent with the format of data we were given.

Fig. 2. Spectrogram of a raw audio signal along with activity segments(marked in green) as detected by our algorithm



C. Classification

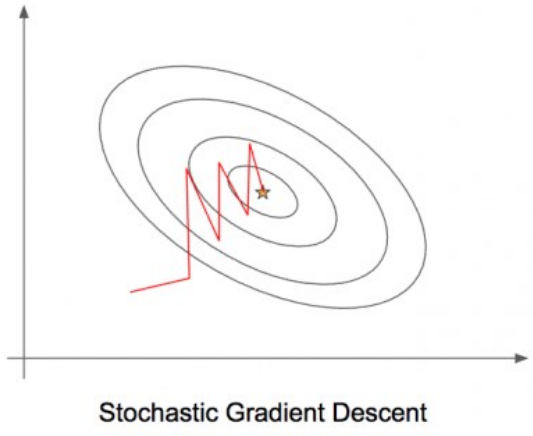
Once we have detected various activity segments we run our various classification algorithms on each frame of that segment and then max-pool the results and report the dominant class as the audio category for that segment.

IV. Classification Algorithms

We implemented the following classification algorithms.

A. Linear Classifier with Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a simple and efficient optimization algorithm used to find the values of parameters/coefficients of functions that minimize a cost function. It is used for discriminative learning of linear classifiers under convex loss functions. [1]. SGD is useful especially when we have large data-samples where normal gradient descent becomes computationally very expensive.

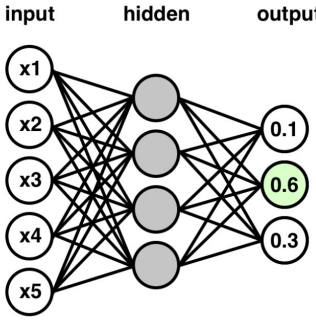


B. Neural Network Classifier

A neural network classifier consists of several layers, each containing multiple units. During the training process, we train a neural network to obtain the weights joining the nodes. A neural network can model complex relationship between the input and output by estimating the weight vectors corresponding to each layer.

For our case we take a 3 layered neural network(which includes input as well as output layers). The output consists of 3 neurons, each corresponding to one class(i.e. $music$, $silence$ or $speech$).We use $relu$ activation function in the hidden layer and $softmax$ activation function in the output layer. Furthermore, we are using categorical cross-entropy as the loss function and $rmsprop$ as the optimizer. Considering limited data, we kept the neural network with only 3 layers. Training a deep neural network requires significant amount of training data. This can help us capture very complex relationship between input and output.

Fig. 3. A 3 layered typical neural network



C. Recurrent Neural Network(LSTM)

A Recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed or undirected graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them highly useful in tasks involving sequences

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture[1] used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video).

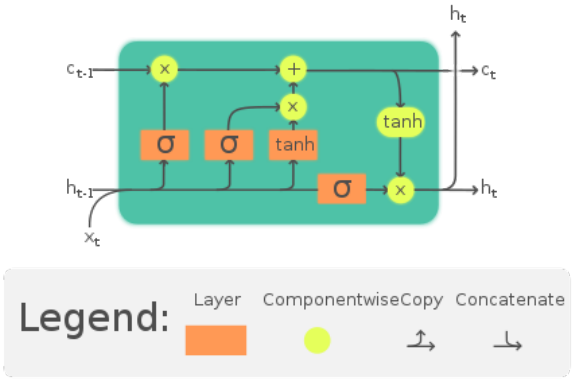


Fig. 4. Structure of a LSTM Cell

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

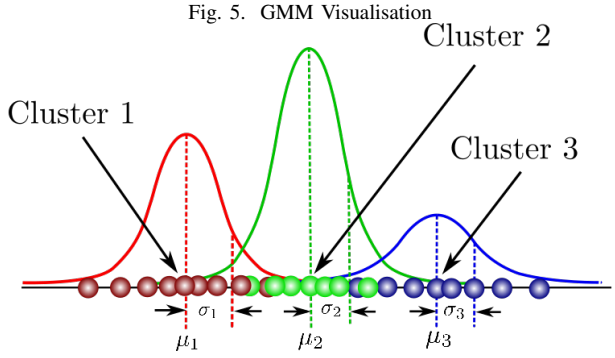
LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series

D. Gaussian Mixture Models(GMM)

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture

of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.

Fig. 5. GMM Visualisation



V. Results

We had the following results after running the above set of algorithms-

1) The detection accuracy **significantly** increases with increasing data size. We currently have 100 activity centers in our combined dataset. This accounts for $100 \times 313 = 31300$ data points for training. Typical machine learning models have much more variables than the number of available data points. Thus our algorithm exhibits a great scope of improvement by increasing the number of data points. The same was verified when we trained on different number of data samples.
2) Linear Classifier performed the best out of the given classifiers. This is because linear classifiers have much lesser parameters as compared to the other classifiers. As a result a good training can be done with the given amount of limited data points.
3) Energy profile can be a very good way to measure activity segments in audio clips. By using the energy profile information and some clubbing algorithms we were able to very precisely detect activity centers in various audio clips. Because activity centers could be detected very precisely our classifiers gave a good performance despite lack of a comprehensive training.

Significant results were obtained for Neural Network Classifier and Linear Classifier models which is given below.

Results			
Method	Accuracy on Dataset-1(30 data points)	Accuracy on Dataset-2(70 data points)	Accuracy on Dataset-3(100 data points)
Linear	52.4%	76.5%	81.7%
Neural Network	54.7%	78.2%	83.8%
LSTM	44.7%	51.2%	56.8%
GMM	40.7%	53.2%	58.8%

References

[1] Stochastic Gradient Descent
[2] Neural Network Classifiers
[3] Wikipedia: Gaussian Mixture Models