Sound Recognition using Machine Learning

Steve Kim and Faiz Ganz

18 December, 2019

Course: PY 895 M8: Machine Learning for Physicists
Instructor: Professor Pankaj Mehta

1 Introduction

The goal of our project is to perform machine learning for the purpose of sound recognition. According to Freesounds Kaggle competition, due to the many different kinds of sounds that exist, a general algorithm for sound detection does not exist. Usually, machine learning for sound recognition is limited to music recognition software such as Shazam. Sounds recognition can be applied to a myriad of fields. In biological research, sound recognition can be used to identify unique species of animals. In defense, passive sonar data is used to identify maritime vehicles. Popular consumer technologies such as Alexa use sound/speech recognition to receive commands and authenticate the voice of the caller. Video tagging could also be improved with accurate sound recognition.

2 Data Set

2.1 Urban Sounds Dataset

The dataset chosen for the project is the UrbanSounds8K dataset which can be found at https://urbansounddataset.weebly.com/urbansound8k.html. It contains 8732 labeled sounds belonging to 10 different classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street. Each class has about 1000 corresponding sounds besides for the classes gunshot and car horn, which contain about 400 sounds each. All sounds have a duration less than or equal to 4 seconds and most of them (about 99%) are exactly 4 seconds long, which facilitated the creation a design matrix, and all of them are sampled at 44100 Hz.

2.2 Other Datasets Considered

Different datasets were taken into consideration for the project. However, due to the duration of the excerpts being different for different sound in the other

datasets, the UrbanSounds8K dataset was chosen as the sole source for the project, as standardizing all sounds to have the same time length might have compromised the information contained in the dataset and a method to circumvent such problem has not been identified. Furthermore, all other datasets besides for UrbanSound8K, contained about less than 100 sounds per label, which made them inefficient to for the purposes of this project. The other datasets taken into consideration where the Environmental Sounds, Heartbeat Sounds (Kaggle), and Avian Vocalizations (Kaggle) datasets. These would be interesting to explore and analyze further later in the future.

3 Data Processing

The audio data that we found was in mono .wav format. Using the librosa package for Python, we were able to load data as a one dimensional array of amplitudes. In addition, the sampling rate was provided. Using the one data and the sampling rate, we were able to construct a waveform respresntation of the data. An example is shown in figure 1.

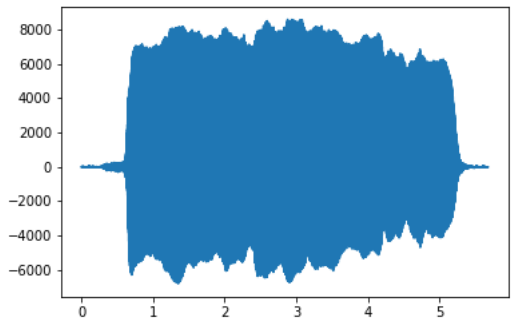


Figure 1: The figure above shows a waveform plot of a flute playing the note A4

In order to process the data so that important features can be extracted by machine learning techniques, we converted the waveform data into a spectrogram. A spectrogram is a graph, whose y-axis has units of frequency and x-axis has units of time. This is obtained by the short fourier transform algorithm, which obtains frequency components of time domain data per window of time. A visualization of a spectrogram is shown in figure 2. It is also important to note that we have processed the data set so that only sounds of 10 unique categories were included.

Evidently, a spectrogram is much like a monochromatic image in that it is an $n \times m$ matrix with "intensity" values. By modifying the parameters of the scipy.spectrogram function, we were able to produce a 300×300 spectrogram. In

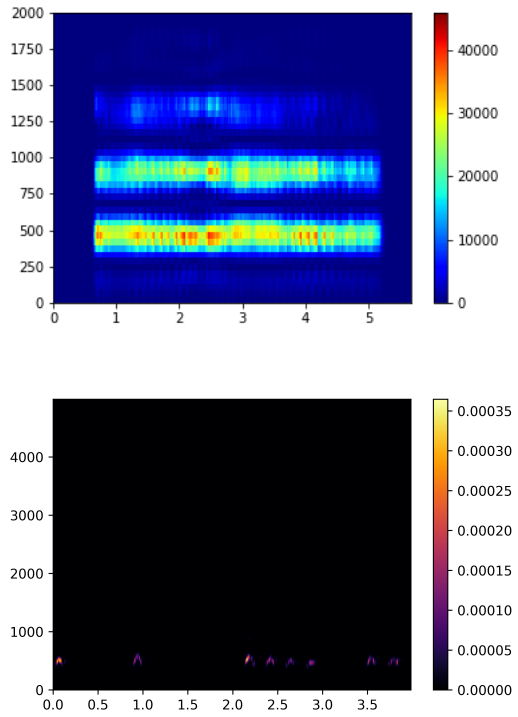


Figure 2: The first spectrogram above shows the sound of a clarinet playing the key of A4. The high intensity lines represent the frequencies which are represented in the sound. The mode was 440Hz, which is the key of A4. The second spectrogram shows a dog barking. The colored spots represent at which times and at which frequencies the dog barks.

other words, there were 300 frequency bins split from $0Hz$ to $\frac{SamplingRate}{2} + 1$, and 300 time bins split from $0s$ to the length of the audio file. Another $70x70$ spectrogram dataset was created for the purpose of computational efficiency.

4 Methods

4.1 Principal Component Analysis (PCA) and t-SNE

PCA was used to find the most contributing components in our data set to variance and to find an efficient dimensionality reduction. t-SNE was also attempted in this part of our analysis. We hoped to cluster the results of our PCA and t-SNE analysis, but found it would have been meaningless to do so. This is explained further in the results section.

4.2 Convolutional Neural Network

Given the image like nature of our data set, we decided a Convolutional Neural Network (CNN) would be appropriate for sound recognition. We decided to use two convolutions with two 2×2 max pool filters with a stride of 2. Afterwards, the convolved and max pooled image was passed through 3 dense layers. The first two dense layers used a relu activation function and the last layer used softmax. The ADAM optimizer was used.

5 Results

5.1 Principal Component Analysis (PCA) and t-SNE

A graph of the results of our PCA analysis is shown in figure 3. Figure 4 shows the 10 different categories isolated in our PCA analysis.

As expected, similar sounds appear to be clustered together. Unfortunately, clustering was unfruitful, as most sounds were grouped together. Furthermore, as shown in the figures, extreme outliers appear to exist. This can be due to large variance in a particular category of sound. For example, in the case of a dog bark, a Chihuahua and a German Sherpherd sounds drastically different. However, when analyzing the categories by themselves, some patterns appear to be present as shown in figure 3 and 4.

An interesting analysis we did is shown in figure 5. Here, we show the first 10 principal components, which should be the frequencies that contribute the most to our $300x300$ data. As one would expect, higher frequencies (above 750 Hz) appear to contribute the least to our data. Most sounds heard in a day to day basis are of below the aforementioned frequency.

Below, in figures 6 and 7, are graphs showing the statistical analysis of our PCA. The minimum amount of components that explain 99% of the variance

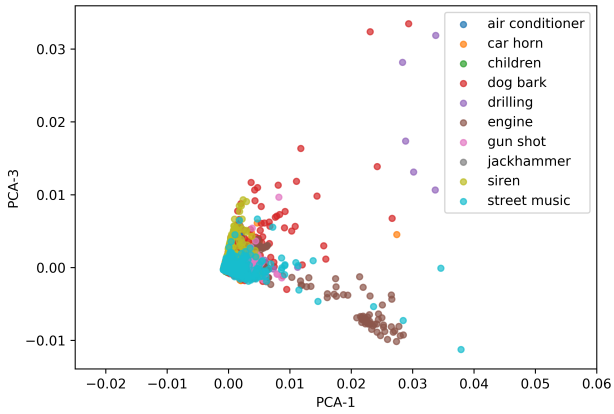


Figure 3: The figure above shows PCA axis 3 vs PCA axis 1.

is 3124 out of 90000 as shown in figure 6. In figure 7, we see that the most significant components are present by the elbow.

5.2 Convolutional Neural Network

In running our CNN with the architecture described above, we were able to achive a test accuracy of only 0.49 with 75 epochs. This was done using the 70×70 data set, as 300×300 would take many hours to run with the computing power we have. Given that we have 10 categories, the lowest accuracy we could achieve should be 0.1. Our performance is reasonable given a worst case accuracy. Another attempt was done with the 300×300, except we cropped out higher frequencies. The peak accuracy achieved was .3 in this case. In attempting this, we used very large convolution filters on the order of 50 to reduce computing time.

6 Future Improvements and Conclusion

Overall, this project allowed us to gain a deeper insight into the field of signal processing and machine learning. Despite somewhat unsatisfactory results, we were able to make senese of it. In the future we hope to be able to modify the architecture of the convolutional neural network to get higher accuracies. Furthermore, we hope to run the full 300×300 data set on a GPU based cloud computer to achieve reasonable computing time.

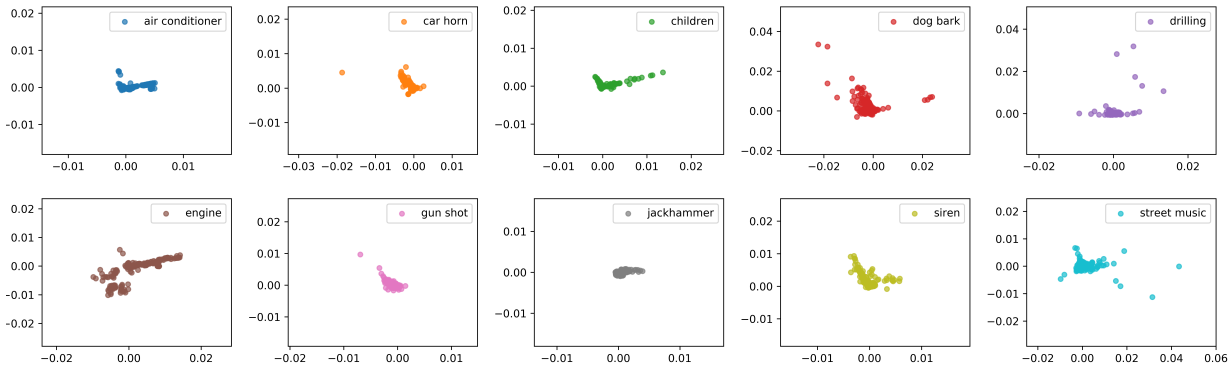
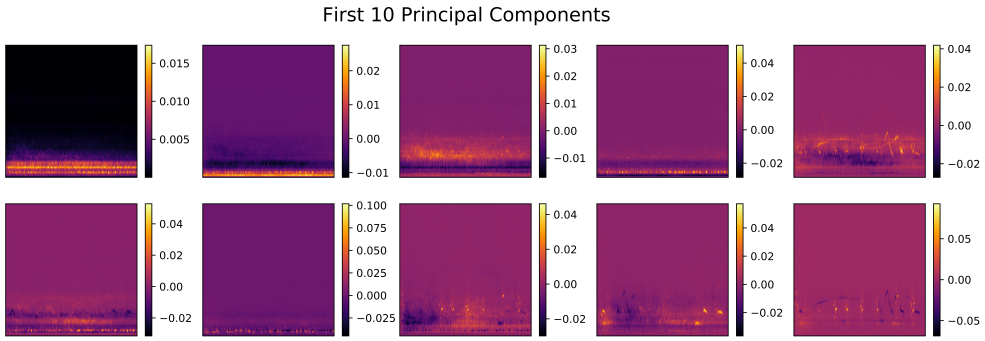


Figure 4: The figure shows the 10 selected categories transformed under PCA.



(

Figure 5: The first 10 principal components of our data set.

)

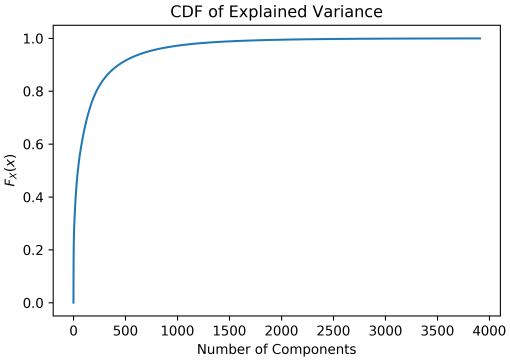


Figure 6: the CDF of explained variance

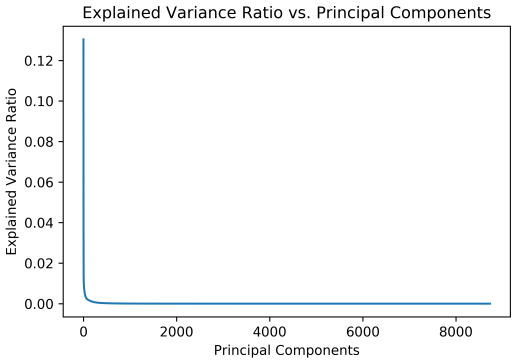


Figure 7: Explained variance ratio versus the amount of principal components.