Name: Jason Zhou
Mentor: Dr. Dongjin Song
Status Report #: 12
Time Spent on Research This Week: 6.25
Cumulative Time Spent on Research: 82.25
Miles Traveled to/from Mentor This Week: 0
Cumulative Miles Traveled to/from Mentor: 0
==========

Monday, December 6th, 2021: (0.25 Hrs)

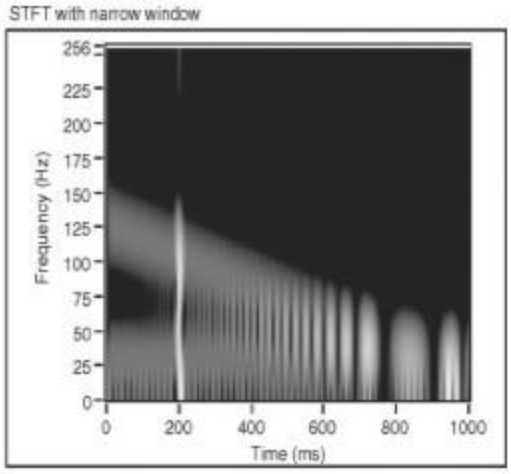
On Monday, I had my weekly Zoom meeting with my mentor. Other than reporting what I had done over Thanksgiving break (which, was not much), I asked my mentor a couple of questions about Gaussian Mixture Models (GMMs). Specifically, GMMs only work well if the data naturally follows a normal distribution. The problem, then, is what if the data does not follow a normal distribution? My mentor recommended that if this does happen, I can create a variational autoencoder to reshape the data so that it better fits with a GMM.

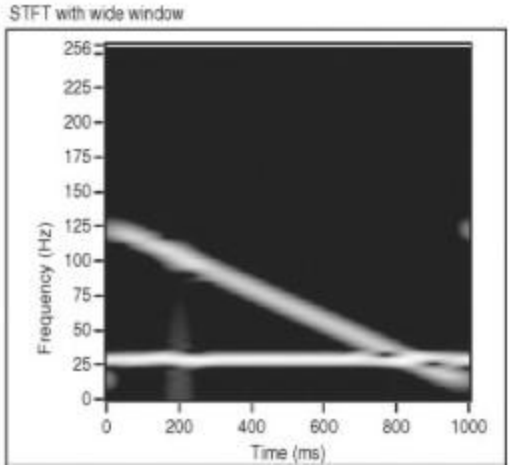
Wednesday, December 8th, 2021: (2 Hrs)

I spent this day studying the Short-Term Fourier Transform (STFT). Essentially, the STFT is a way to convert an audio file into a graph that gives the intensities of certain frequencies that are present at a certain time, which covers one of the main weaknesses of the Discrete Fourier Transform (DFT) because it only scans the entire audio file to give the intensities of the frequencies within a file; however, it does not indicate the time at which these frequencies appear/happen.

Specifically, the STFT achieves this effect by splitting the audio file into multiple frames/windows/segments that represent different time periods/slots. For example, if the audio clip is 10 seconds long, then it could be split into 5 segments, each of which is 2 seconds long. Then, the DFT is used on each segment to convert the raw audio into its frequencies and intensities. By taking these windows and plotting them on the x-axis (which represents time) and y-axis(which represents frequency), one can measure when a certain frequency appears.

However, although STFT is extremely useful and one of the weaknesses of the DFT, it still has its drawbacks. To illustrate, when using the STFT, one must be aware of the Time/Frequency Trade-Off, which states that if one places emphasis on time resolution, one will lose frequency resolution and vice versa. In other words, the more segments the audio file is split into (which makes the time measurements more precise), the harder it will be to measure the frequency at a certain time. Likewise, the fewer segments the audio file is split into (less precise time measurements), the more accurate the frequency measurements will be.





(An example of the Time/Frequency Trade-Off described above. As one can see, when the window size is narrow, the audio will be split into more windows. This leads to a more precise time measurement; however, it becomes hard to accurately determine what the frequencies are because the lines blur. In contrast, a wide window size, and therefore fewer windows, leads to more accurate frequency measurements)

Thursday, December 9th, 2021: (1 Hr)

On this day, I decided to revisit some fundamentals of NumPy because I often found myself forgetting some basic commands or concepts. Specifically, I focused on relearning arrays in NumPy.

I revisited some basic concepts like creating, reshaping, and manipulating arrays. Additionally, I also reviewed some of the various methods and functions that exist within the Numpy module such as creating and choosing random data, and operations like finding the mean, max, min, variance, standard deviation, and the sum of NumPy arrays.

```
f.sum() #take the sum of all items within an array
f.max() #returns the max item in the array
f.mean() #returns the mean of all items
f.var() #returns the variance
f.std() #returns the standard deviation
```

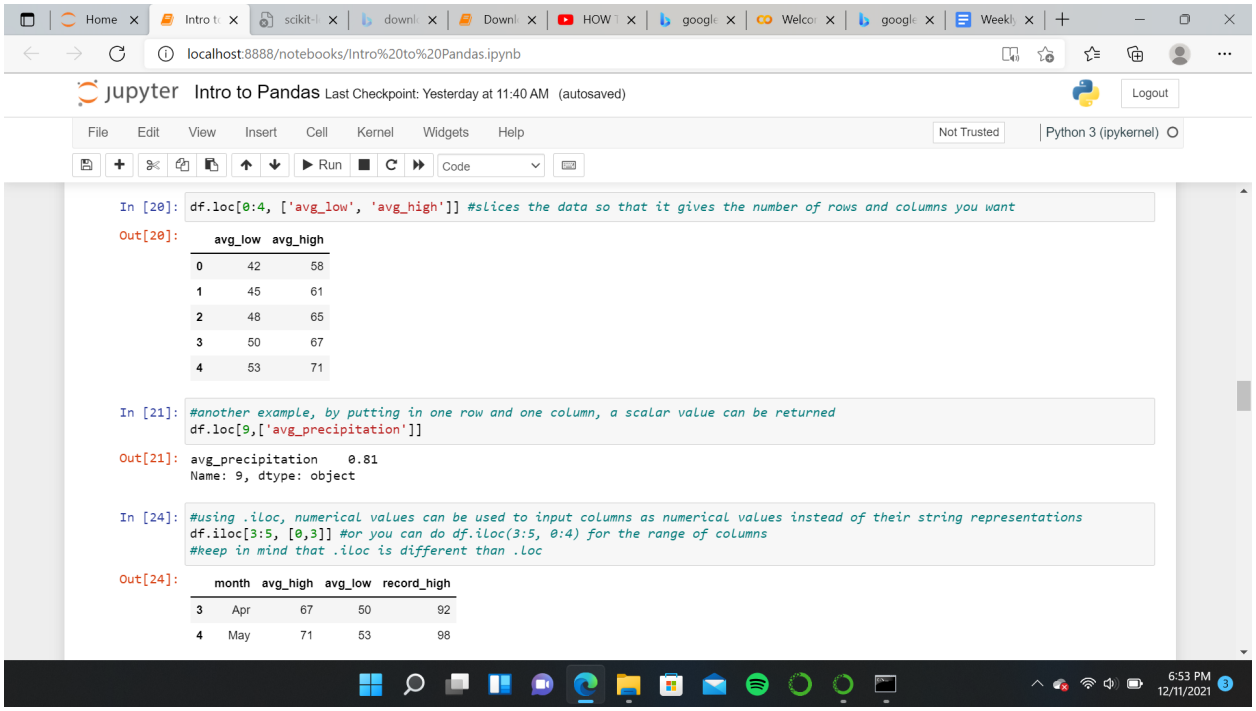
(A picture of the various functions that can be called with the NumPy module. Also, in the context of this picture, “f” is equal to a 1 by 5 NumPy array)

Friday, December 10th, 2021: (3 Hrs)

During release time, I realized that I did not actually understand how to use the Pandas [1]DataFrame object, which I suspect will be an important tool in my project.

First, because Pandas uses tabular data, the columns and rows must be defined. Usually, the rows are defined as indices (e.g 1, 2, 3, 4, 5 etc.), while the columns are labeled with actual words or categories such as color, height, weight, etc.

Second, Pandas has a lot of functionality that allows the user to manipulate and rearrange the data, which is called slicing. So, I learned how to pinpoint various rows, columns, and locations within the DataFrame and call them for later reference.



(A picture of code showing the slicing of a DataFrame to get/obtain certain values.)

[1] The Pandas module is basically a higher level version of the NumPy module. Also, instead of handling raw data, Pandas is good at storing and utilizing tabular (table) data.

Finally, I learned how to input blanks into the data. To explain, sometimes within an experiment there will be boxes in a table that are left blank because there was no data for that specific entry. Also, even with a blank entry, the Pandas DataFrame is still able to run calculations without any errors, which is something that NumPy cannot do.

Up until this point, I had just been trying to learn everything that I could about machine learning and audio. However, after learning Pandas, I finally began actually doing my project. Specifically, I focused on loading the data into a format that I could use in a coding environment. This proved to be hard because I had over 100 files of audio, meaning I would have to write a line of code to import each file into the code. Thus, I began trying to find ways to import more than one file at once. Because I had no wifi at the high school, it made it hard to find solutions, so, unfortunately, I did not get very far. However, I did manage to find a way that I think has the potential to work (but I have not gotten the chance to try it).

References

James, J. (2018, May 17). *Python: Pandas tutorial | intro to dataframes* [Video]. YouTube. https://www.youtube.com/watch?v=e60ItwlZTKM

James, J. (2018, June 11). *Python: NUMPY | numerical Python arrays tutorial* [Video]. YouTube. https://www.youtube.com/watch?v=8Mpc9ukltVA

Velardo, V. (2020, September 7). *Short-Time Fourier transform explained easily* [Video]. YouTube. https://www.youtube.com/watch?v=-Yxj3yfvY-4