Name: Jason Zhou
Mentor: Dr. Dongjin Song
Status Report #: 18
Time Spent on Research This Week: 5.75
Cumulative Time Spent on Research: 131.75
Miles Traveled to/from Mentor This Week: 0
Cumulative Miles Traveled to/from Mentor: 0
==========

Monday, February 7th, 2022: (0.5 Hrs)

On this day, I had my weekly mentor with my mentor, and his student, Binghao Lu, joined in. During this meeting, I had some questions about the design of neural networks and how to determine the size of a hidden layer. They told me that it was mostly dependent on the quality of my computer. For example, if a computer has more memory, one would make the size of the hidden layers bigger (and the opposite case is also true).

I also asked them about GPU acceleration. Specifically, GPUs process code a lot faster than CPUs. Normally, all I would need to run a program is a CPU; however, when I am using 47,000 files, it definitely helps to speed up the computing time.

Finally, I set up another meeting with Binghao because he wanted to know more about the project I was doing (the objective, the type of data, the methods being used, etc).

Tuesday, February, 8th, 2022: (1 Hrs)

Last week, I had issues with padding the data so that it all followed a uniform structure for the neural network. However, it also occurred to me that the way I was coding was quite sloppy and hard to work with.

To solve this problem, I ended up watching a video about how to properly set up the data that I would be using. This led to me creating a class for the data and implementing functions that made working with the data easier to use and less confusing to navigate.

```
class DCASE(Dataset):

    def __init__(self, path, arr, sr, samples):
        path = path + '\\**\\*.wav'
        self.dir = glob.glob(path, recursive=True)
        self.lMspec = arr
        self.targetSampleRate=sr
        self.numSamples=samples

    def __len__(self):
        return len(self.dir)

    def getDir(self):
        return self.dir

    def getlMspec(self):
        return self.lMspec

    def __getitem__(self, index):
        signal, sr = librosa.load(self.dir[index])
#         signal = self.cutWave(signal, sr)
#         signal = self.rightPad(signal)
        return signal, sr
```

(A screenshot of the class I made to make it easier to use my data.)

Wednesday, February, 9th, 2022: (4 Hrs)

After creating the class for the data, I looked up the programming documentation about how to pad data. However, I ended up watching a video instead because it was easier to understand.

While learning how to pad data, it also occurred to me that I might need to "trim" the data[1]. However, upon further investigation, I realized that this was not necessary because most of the data adhere to a certain size, and the ones that do not are actually smaller in size.

After creating the functions to pad my data, I worked on integrating this into the larger function of my data class that converts those audio files into a mel spectrogram. While trying to test the code to make sure it worked, I ran into a multitude of errors. Although most of them were simple syntax errors, there was one error which required me to update the programming libraries I was using. Unfortunately, I did not realize this at first, and it took me a little to notice what the problem was.

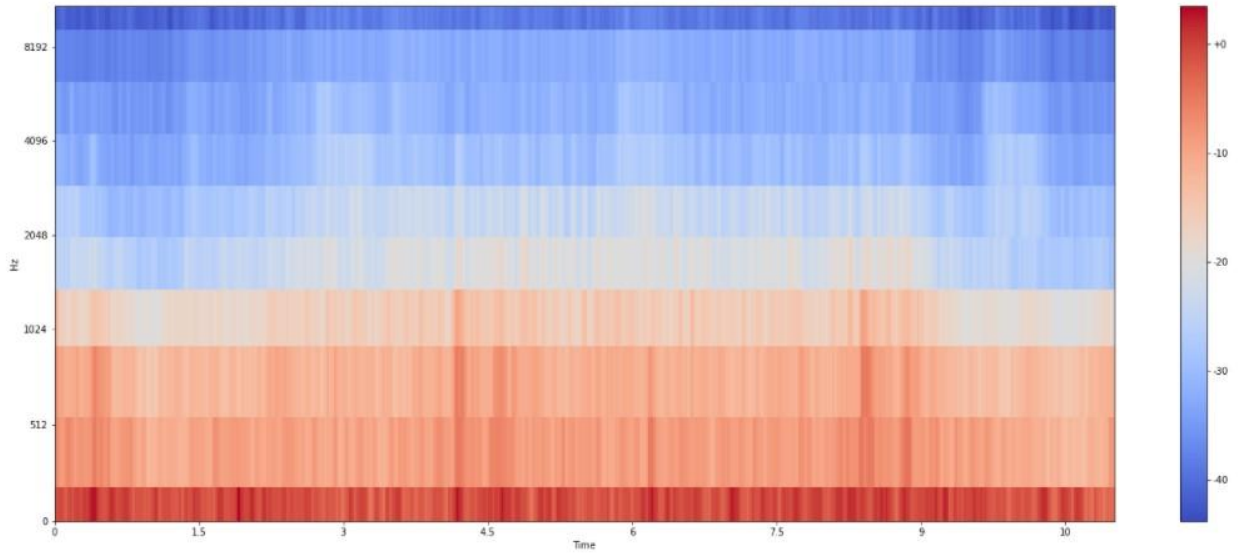
Besides that, the rest of my time was used to test the functions that I had written and running small scale simulations of converting audio files, padding them, and then storing them to make sure the process worked. Once I was fairly certain that nothing would go wrong, I created a class for my data with all 47,000 files and ran the code to preprocess them.

Friday, February 11th, 2022: (0.25 Hrs)

[1] If a padding is used to extend the data (in this case, extend the duration of an audio clip), then "trimming" would be used to minimize the data (decrease the duration of an audio clip). These processes are done to keep the durations of the clip constant so that it can be fed into the neural network properly.

On this day, I met with Binghao to discuss the details of my project. I explained to him the objective of my project and explained the methods I would be trying. Currently, as a starting point, I would like to try making a convolutional autoencoder (CAE), which would take in Mel spectrograms, deconstruct them, and then attempt to reconstruct them. The worse it reconstructs the Mel spectrogram, the more likely it is that the audio file is an anomaly[2].

Later, I showed an example of some of my data.



(An example of the data that I showed Binghao. It is a screenshot of a Mel spectrogram)

After he understood the general basis of my project, he recommended that after I make a CAE, I might try making a Transformer-based model because there has recently been research to suggest that Transformers perform very well when given time-series data (e.g audio).

References

Velardo, V. (2021, June 7). *Custom audio PyTorch Dataset with Torchaudio* [Video]. YouTube. https://www.youtube.com/watch?v=88FFnqt5MNI&list=PL-wATfeyAMNoirN4idjev6aRu8ISZYVWm&index=4

[2] This is different from the GMADE approach I presented during my ARM grant funding simulation. I want to try making a CAE because it is easier to do and would be less complex for someone of my level to code.

Velardo, V. (2021, June 14). *Extracting Mel spectrograms with PyTorch and Torchaudio* [Video]. YouTube. https://www.youtube.com/watch?v=lhF_RVa7DLE&list=PL-wATfeyAMNoirN4idjev6aRu8ISZYVWm&index=5

Velardo, V. (2021, June 17). *Pre-processing audio with different durations* [Video]. YouTube. https://my.noodletools.com/web2.0/bibliography.html

Velardo, V. (2021, June 21). *Pre-processing audio for deep learning on GPU* [Video]. YouTube. https://www.youtube.com/watch?v=3wD_eocmeXA&list=PL-wATfeyAMNoirN4idjev6aRu8ISZYVWm&index=7