

Music Genre Classification Write-Up

Instructions for Reproducing Results:

- 1) Download GTZAN from this link: http://marsyasweb.appspot.com/download/data_sets/
- 2) Unzip the dataset and put it in the same directory as the python code
- 3) Install tensor flow using this tutorial: <https://www.tensorflow.org/install/>
- 4) Install keras (with pip or conda, in the same virtual environment as tensorflow)
- 3.5) Alternatively launch a deep learning aws ec2 instance that has keras and tensorflow already on it
- 5) Install matplotlib and librosa (via pip)
- 6) To get the spectrograms, run toSpectrogram.py
- 7) This will put all spectrograms into a "spectrogram" folder inside the genre folder
- 8) To run genreCNN.py you must take the spectrograms and put the training spectrograms in a new folder within genres called "training" in a folder with the name of their genre, and do the same for validation spectrograms.
- 9) All other scripts should be runnable out the box.

Notes on the Neural Net Scripts:

I'm not sure reproducing results will be very interesting to anyone because none of the scripts learn anything. Also the non-sequential scripts (oneDimensionalCNN and oneDimensionalCNNnewbroken), which I wrote using keras' functional API started giving me memory errors, and even when I ran them on aws I got seg-faults, so not really sure what the problem is. Besides having those issues, the scripts that only learn to be as good as random guessing also have a loss that remains exactly the same throughout training. So I'm reasonably confident that something is messed up in the backpropagation of my net, A loss that is the same throughout 200 epochs to the thousandths place is rather suspicious. I'm going to post my code in forums over the break to see if I can get any help. I also think that starting with this new paper I mentioned in my presentation from scratch could prove fruitful and fix the mistakes I have.

What I Learned:

I learned a lot about Amazon Web Services and feel very proud to have my own instances running somewhere in Oregon. I also learned more than I anticipated about spectrograms and how small tweaks in the representation of my data could affect the training of my net. I did not look deeply enough into what the spectrograms actually meant before designing my neural net architecture and by the end I sorely regretted that. I learned that there is such a thing as giving a Net too much formless data to sift through so that it is harder to find patterns, and I will keep this in mind in the future. I think what I mostly learned was that in music especially, the crux of these automations is often the representation of the data more than the

Time it Took:

Once I got a basic Neural Net up and running I would spend ~2 hours a day (or however long it took to write a new architecture that could train overnight) three days out of the week redesigning the architecture and running it overnight so I could see the results (however abysmal) in the morning. Once I got my AWS credits though, which was admittedly way after I should have, I worked on it a solid 5 hours a day trying to sort through all the issues and just get all my approaches to run. Eventually I got my sequential models to run but they still didn't learn anything so I have no doubt I'm making some kind of mistake. I will scour the forums for help to fix this by next semester so I can move on into making a almost flawless neural net for my Senior Project.

How it Compares to Proposal:

The final project model resembles what I originally proposed in that it models the paper to the best of my understanding. Unfortunately that understanding was more flawed than I realized. As such, I also stayed more or less on schedule, having a basic Neural Net up and running by Fall Break, but I now realize that my schedule assumed the tweaking of that Neural Net would be the easy part when it was, in fact, by far the hardest part.