Machine Learning Project 3
Comparing Neural Network Classifiers for Audio Categorization

Team CAAK: Alyshia Bustos, Aislinn Handley, Carolyn Atterbury, Keira Haskins

11 May 2020

Description

In this project, we classify the genre of audio recordings using both Convolutional Neural Networks (CNNs) and Feed Forward Neural Networks (FFNNs). Using mp3 files from the Free Music Archive, we had access to 2400 mp3s which we used for training the neural networks and developing our own accuracy scores. We also had access to 1400 unclassified mp3s which we were able to classify and upload to kaggle for an additional accuracy score. Each of the songs in the training set were classified as one of six genres: Rock, Pop, Folk, Instrumental, Electronic, Hip Hop. We extracted three features from the audio data which we used to train the neural networks. The three features were Spectrograms, Timeseries, and MFCCs. Each of the features were stored and read into the neural networks as image data.

For our Feed Forward Network we overall find that the MFCC data produces the best overall accuracy, followed by the timeseries data, and lastly the spectogram data. We find that adjustment of our hyper-parameters makes a big difference for MFCC, with our accuracy ranging from around 16% upwards of 45% depending on our values of batch_size, learning_rate, momentum, decay, as well as the actual structure of the network itself. On the other hand we find that for the most part, adjustment of hyper-parameters and network design have little to no impact on the accuracy of our model when running our model on spectogram data, with timeseries somewhere in between the two.

For our Convoluntional Neural Network we find that it takes a lot of memory to load the network as well as the data onto the GPU which makes it near impossible to use the full dimensions of our images. It is for this reason, along with problems we encountered when attempting to run our code on the CS GPUs that we reduced the dimensionality of both our data as well as our CNN to allow it to fit into about 2 GiB of memory on the GPU of a laptop. For timeseries we are able to achieve an accuracy of about 25%, about 45% for MFCC, and about 15-17% for our spectograms.

Overall it seems to be the case that our spectograms are not ideal given the low accuracy scores which we have been getting for both the Feed Forward Neural Network as well as the Convolutional Neural Network. Given more time we would probably work on figuring out why the spectograms turned out the way they did and we would probably work on making them better for both networks. As are however, they do not appear to have enough detail in order to be able to train either network very well.

Classifiers

In order to classify the genre of a song, we used Convolutional Neural Networks, and Feed Forward Neural Networks. We wanted classifiers that could train over the same set of features, so we can compare the effectiveness of each classifier with this particular classification problem. We selected Convolutional Neural Networks as they are known to perform well on image classification tasks and have translation invariant properties. Like CNNs, feed forward neural networks are also universal approximators and can perform classification tasks. Each of these networks were trained over the same three features using back-propagation methods.

Feed Forward Neural Network

The image RGB data is processed and flattened into a 1D array before being fed into the feed forward neural network. The order of the images are shuffled as they are passed through the network, and our data is segmented so that 80% of the data becomes training data, while the remaining 20% becomes testing data that is used for evaluating the accuracy of the network. We choose this classifier because we primarily want to focus our efforts on implementing a convolutional neural network, but we also wanted a simple feed-forward neural network in order to create a baseline model with which to compare against.

The feed forward neural network uses stochastic gradient descent with Nesterov momentum as an optimization strategy, with six dense layers ranging in size from 500 to 6 nodes. We initially created our network with 4 layers, 500, 100, 10, and 6 all using ReLU activation functions. As we developed our network we ended up trying a combination of various activation functions including ReLU and Tanh, ReLU and leaky ReLU, Tanh and leaky ReLU, and all three. We also experimented from between 4 to about 8 layers, however overall we found that more and more layers seemed to degrade the overall accuracy of our model. We also tried starting with a smaller number of nodes and increasing them: 128 up to 512, 256 up to 512 as examples. We also spent a fair amount of time tweaking our hyper parameters including the learning rate, batch size, momentum, decay, and the alpha value for the leaky ReLU. We were able to achieve a high accuracy score of around 41-45% using a network with 5 layers, using combinations of ReLU and leaky ReLU, with 500, 200, 50, 10, and 6 nodes per layer.

We tuned our hyperparameters on this network by first running with 30 epochs in order to observe the change in accuracy. We want to observe slow but steady increases in accuracy per epoch, however we find that the accuracy seems to be bounded around 40-45% for MFCC, about 24-27% for time-series, and a high of 19% for spectRograms. Overall we have found it considerably more difficult to improve the accuracy of our feed-forward neural network given spectrograms as input.

Convolutional Neural Networks

Similar to our FFNN, the image RGB data is once again processed and first flattened into a 1D array before being fed into the network. Once again we use a distribution of 80% training data to 20% testing data. We choose this classifier because they are known for being effective for the analysis of image data in machine learning.

The CNN uses the same type of model for training as the FFNN, or stochastic gradient descent with Nesterov momentum for optimization. The model that we used in the end has six layers: a 32 node

2D convolutional layer with a 5 x 5 kernel size, a max pooling layer which is 2 x 2, a 64 node 2D convolutional layer with a 5 x 5 kernel size, another max pooling layer of the same size as the first one, a 100 node dense layer, and finally a softmax layer. We use ReLU for our activation functions in the convolutional layers as well as the dense layer.

Features

Each of the classifiers are trained on three features: Spectrograms, Timeseries data, and MFCC. Each feature is represented as an image of similar size that can be processed and used by both of our classifier methods. In order to extract the feature information from the given mp3 files, we first convert the mp3 files to wav files, and then perform a procedure on the timeseries data to ensure that all of the data is normalized by the mean and standard deviation. This ensures consistency across all data. From there, the normalized wav files can be transformed into the various feature image types. The images were each 277 by 372 pixels in size.

Spectrograms

Spectrograms represent the frequency and amplitude of a given frequency at a particular time. While the frequencies increase along the y-axis, time increases along the x-axis, and the amplitude of each frequency can be seen by the intensity of the color in the spectrogram chart. In order to reduce the size of our spectrograms, we included frequencies of up to 5000 HZ, with a maximum intensity of 0.06.

Timeseries

Timeseries data can be read directly from the wav files and plotted over time. The y-axis of the timeseries plots represent the amplitude of the wave at a particular point in time, which increases along the x-axis.

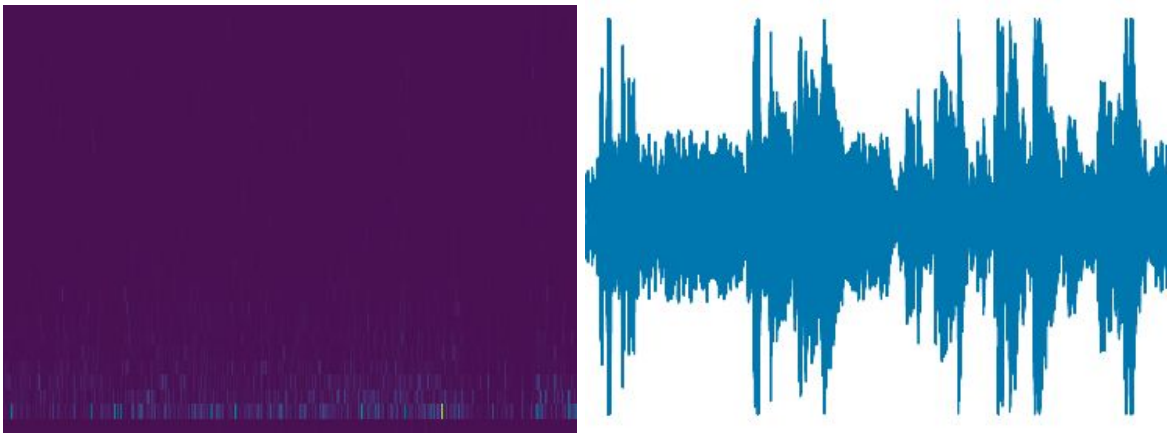


Figure 1. Left: Example spectrogram image. Right: Example timeseries image.

MFCC

The Mel Frequency Cepstrum Coefficients create a ceptrum representation of the wav file but on the mel frequency scale. The MFCC represents the power spectrum of a sound over time.

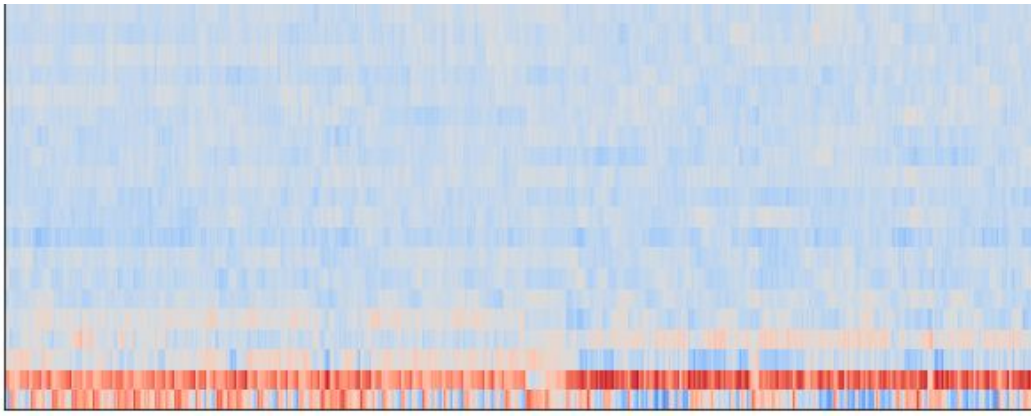


Figure 2. Example MFCC image.

Results

In this section we go over the results for the CNN and FFNN, and compare the effectiveness of each classifier with the three inputs. For each network we had difficulties with running out of memory on the GPU, and so we could only read a portion of the image, and never the full image. The FFNN was able to read about 50% of the original image size before running out of memory, while the CNN could only read about 10%. Despite these differences, the FFNN and the CNN had a similar accuracy score. The FFNN was faster to run than the CNN, but it took more epochs to converge on a particular accuracy value. The CNN gained accuracy quickly with initial epochs, but then it loses accuracy over time due to overfitting. Both the CNN and FFNN have higher accuracy for the MFCC feature, and have the lowest accuracy with the spectrograms. When looking at the spectrogram images, they do look more uniform as a group than the MFCC and timeseries images. That could be the reason why the spectrogram images are harder to classify.

Convolutional Neural Network

The convolutional neural network had a maximum accuracy of 45% with a confidence interval of 2.1% after training over 50 epochs on a batch size of 50. This accuracy was achieved with a learning rate of 0.001, a decay of 1e-6, and a Nesterov Momentum of 0.9. The confusion matrix for the final epoch of the maximum run is below. The various genres and their ids are: Rock:0, Pop:1, Folk:2, Instrumental:3, Electronic:4, Hip-Hop:5. The Electronic category was misclassified the highest number of times as being Hip-Hop, with 48 occurences in the final epoch. The Folk and Instrumental genres were often misclassified, as well as Rock and Pop.

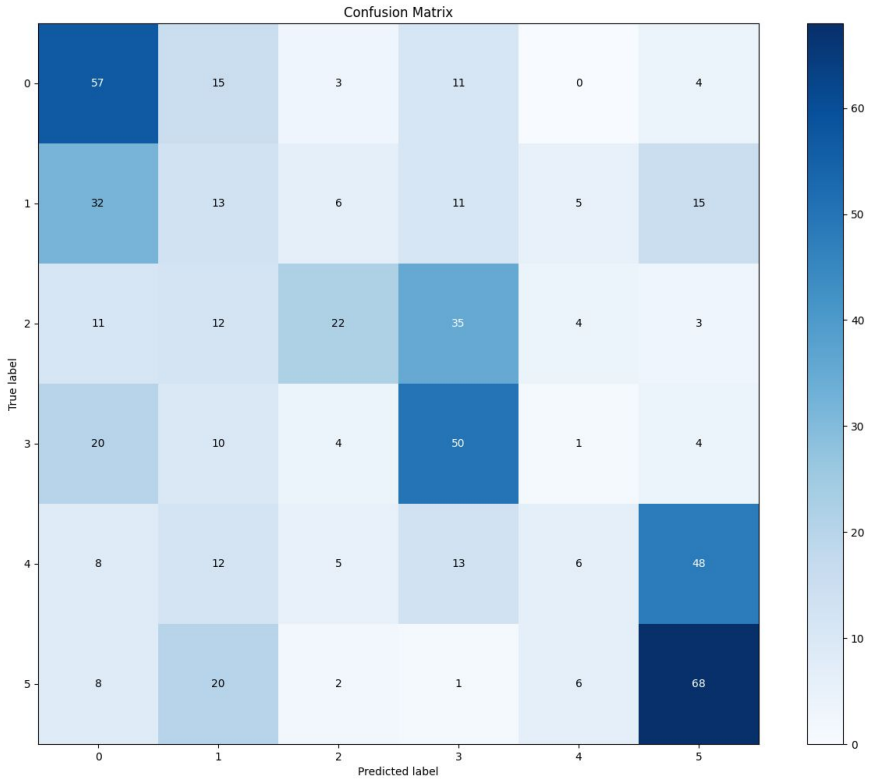
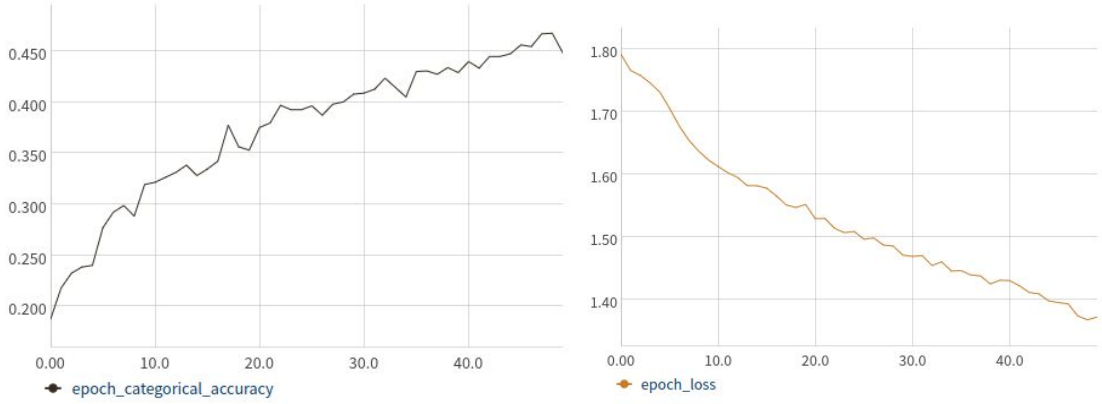


Figure 3: Confusion Matrix for CNN

In Figure 4 we can see the epoch categorical accuracy, as well as the epoch loss. The epoch categorical accuracy increases from 20% to 45% throughout the 50 epochs, while the epoch loss decreases from 1.8 to less than 1.4.



Feed Forward Neural Network

The feed forward neural network had a maximum accuracy of 45% with a confidence interval of 2.1%. The feed forward neural network ran for 200 epochs with a batch size of 150. It had a learning rate of 0.01, and used nesterov momentum. The confusion matrix for the final epoch of the FFNN is below. The various genres are represented with their corresponding ids: Rock:0, Pop:1, Folk:2, Instrumental:3, Electronic:4, Hip-Hop:5. The highest misclassified genre was Hip-Hop, as seen in the high number of occurrences in the fifth column of the confusion matrix. Similar to the results with the CNN, the Electronic category was often misclassified as Hip-Hop, with 54 occurences in the final epoch. The Pop category was also misclassified as being Hip-Hop, with 33 occurences in the final epoch.

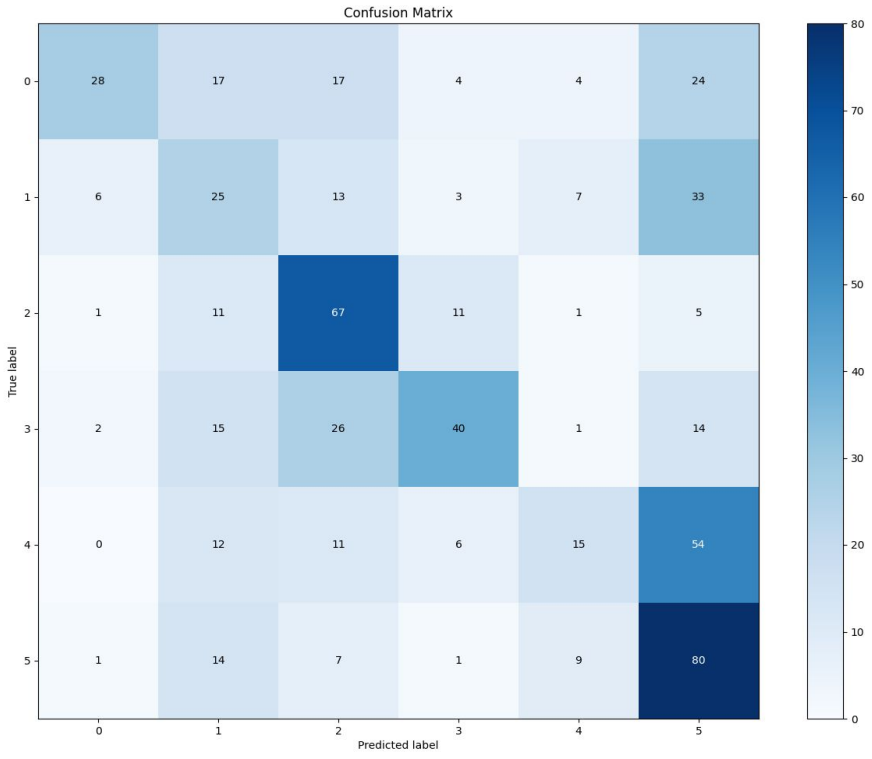


Figure X. Confusion Matrix for FFNN

Below, we can see the epoch categorical accuracy as well as the epoch categorical loss. The epoch categorical accuracy increases rapidly in the first 50 epochs, and then continues to increase up to 200 epochs. The variance remains consistent, or even decreases with higher categorical accuracy. The epoch loss starts at 1.8 and then decreases over the 200 epochs, eventually ending up below 1.4.

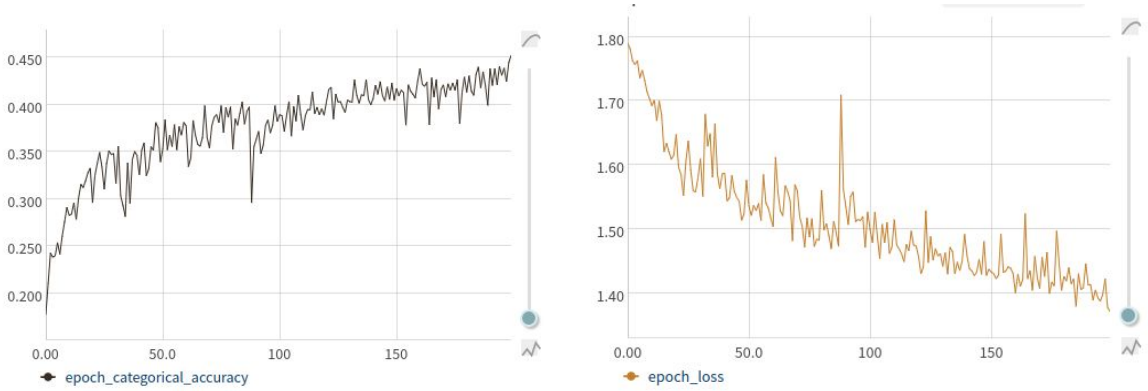


Figure X. Left: Epoch categorical accuracy, Right: Epoch loss

Conclusion

Overall we found the FFNN and CNN were equally effective at classifying the audio files, despite the fact that the CNN was more computationally expensive to run. If we had more computational resources, the CNN would have been able to read a larger percentage of the image and potentially have a higher accuracy score. Given more time and computational resources, we could have experimented with adding more layers to both of the neural networks to improve the accuracy. We also looked into methods for tuning hyperparameters, by using genetic algorithms, unsupervised learning methods, or a brute force approach. Each of these options had the potential to improve the accuracy, but they would have taken more time to implement.

Since all of the training data came from the Free Music Archive, it is subject to bias. It will likely be better at classifying other other songs in the archive. This classification task is limited to 6 genres of data, which does not include many different types of music, but this method could be expanded to include other genres.