CIS 5930-001: Projects for Data Science

A4: Music Genre Classification

Fatema Tabassum Liza, Spencer Cain, Wen Huang & Sajib Biswas

Introduction:

For this assignment, we have selected a dataset from kaggle's data repository and conducted music genre classification on the dataset. The dataset can be accessed using the following URL:
https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification

In the dataset, there are music samples from 10 genres in total, namely- **'blues'**, **'classical'**, **'country'**, **'disco'**, **'hiphop'**, **'jazz'**, **'metal'**, **'pop'**, **'reggae'**, and **'rock'**. We attempt to create and train a model which classifies the music samples into these genres.

Data Preprocessing:

The original dataset in kaggle's website contains 100 sample audio files from each genre we mentioned previously. So, there are about a thousand samples to train our model on. We thought increasing the size of our dataset will help to improve the model's performance. So, we used python libraries to split each audio file into three 10-second separate files, since it's sufficient for a human being to distinguish a music genre within 10 seconds.

After completing this task, the dataset we produced is stored in a google drive folder. The dataset used for this assignment can be checked out using the following link:

https://drive.google.com/drive/folders/1VUTMd1kHp1WZPr__MOga-iGDbkYjGEFB?usp=sharing

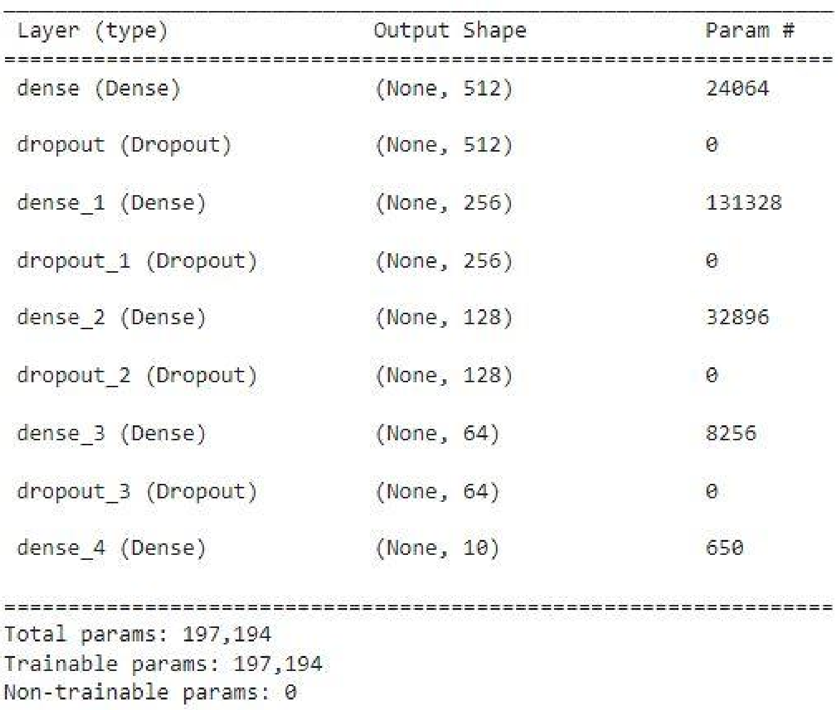
In this new dataset, we have nearly 300 audio samples from each of the 10 genres.

Feature Extraction:

Using the newly created dataset, we extracted 46 features from each of the audio files. The features we extracted are: **Chroma_stft**, **spectral_centroid**, **spectral_bandwidth**, **spectral_rolloff**, **rmse**, **zero_crossing_rate**, and **mfcc**. Mfcc itself gives 40 different features from a single audio sample.

Model:

We used a deep learning model based on Keras library. The model architecture is given below:

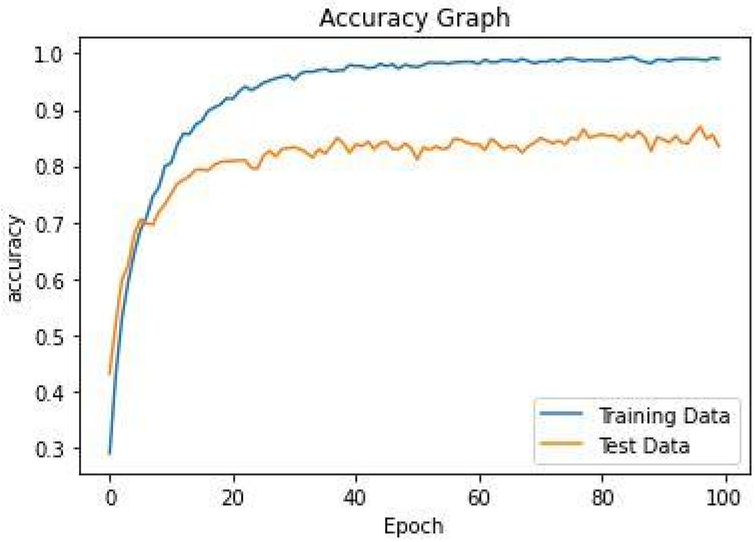


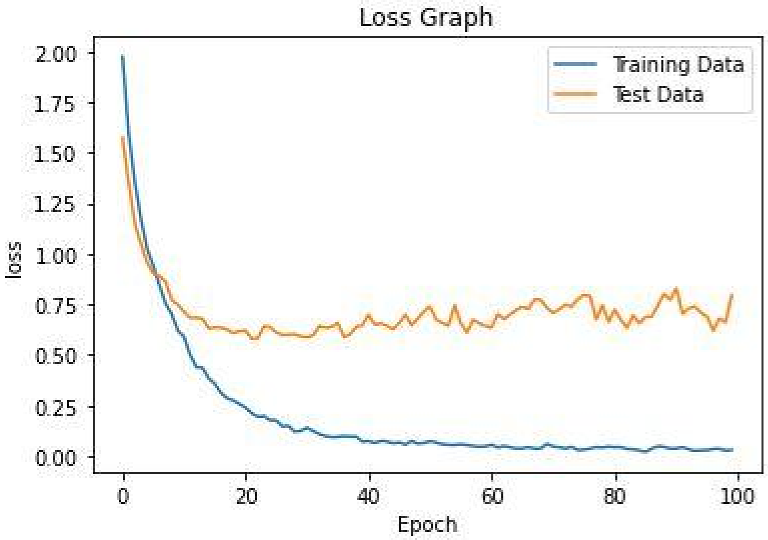
```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 512)               24064

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 256)               131328

 dropout_1 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 128)               32896

 dropout_2 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 64)                8256

 dropout_3 (Dropout)         (None, 64)                0

 dense_4 (Dense)             (None, 10)                650

=================================================================
Total params: 197,194
Trainable params: 197,194
Non-trainable params: 0
```

We trained this model for 100 epochs and ended up getting fairly good results.

Result:

We plotted the accuracy and loss curve for both of the training and testing against each epoch.




Loss Graph
Training Data
Test Data
loss
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
100
Epoch