CSE3013 AI

J-COMPONENT PROJECT

MUSIC GENRE CLASSIFICATION

BY:

SHUBH ALMAL (19BCE2130)

APOORVA REDDY (19BCE2196)

RIYA GUPTA (19BCE2072)

ABSTRACT

- A music genre is a conventional category that identifies some pieces of music as belonging to a shared tradition or set of conventions. It is to be distinguished from musical form and musical style.

- Categorizing music files according to their genre is a challenging task in the area of music information retrieval (MIR). Automatic music genre classification is important to obtain music from a large collection. It finds applications in the real world in various fields like automatic tagging of unknown piece of music (useful for apps like Saavn, Wynk etc.).

- Musical genres are categorical labels created by humans to characterize pieces of music. A musical genre is characterized by the common characteristics shared by its members. These characteristics typically are related to the instrumentation, rhythmic structure, and harmonic content of the music. Genre hierarchies are commonly used to structure the large collections of music available on the Web.

INTRODUCTION

- The classification of genres is a critical undertaking with several real-world applications. The amount of music released on a daily basis continues to rise, particularly on internet platforms like Saavn, Soundcloud, Spotify and Wynk. As the requirement for precise meta-data for database management and search/storage grows, so does the requirement for accurate meta-data. The ability to rapidly identify songs in any given playlist or collection by genre is a critical feature for any music streaming/purchasing business, and the statistical analysis potential provided by accurate and full labelling of music and audio is virtually endless.

PROPOSED WORK

- Our project makes a basic attack on the music genre classification problem. We used a single feature (MFCCs) throughout this project. This gives a fair comparison of learning algorithms, exploring the effectiveness of different features would help to determine which machine learning stack does best in music classification.

- Since genre classification between fairly different genres is quite successful, it makes sense to attempt finer classifications. The exact same techniques used in this project could be easily extended to classify music based on any other labelling, such as artist. In addition, including additional metadata text features such as album, song title, or lyrics could allow us to extend this to music mood classification as well.

IMPLEMENTATION

- We have developed the music genre classification using Convolutional Neural Network (CNN), using 3 convolution layers, each with its own max pool and Batch Normalization , feeding into 3 fully connected layers with ReLU activation, softmax output, and sparse_categorical_crossentropy loss. Dropout layer has been implemented in the 3rd convolutional layer after there was an overfitting in the training dataset. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase. We used sparse categorical cross entropy because our genre classes are mutually exclusive as we trained the model with 400 epochs.

- Before we use the dataset, we have extracted the features from it, so we will extract Mel-frequency cepstral coefficients also known as MFCC are a feature widely used in automatic speech and speaker recognition.

RESULT

- The CNN had a success rate of 90% when identifying some genre during the testing time. The main quantitative metric which we used to judge our model is accuracy (that is, percentage of predicted labels which matched their true labels). We selected our hyperparameters based of empirical results and industry standards. For instance, choosing 3 by 3 window for our first convolution window was a result of seeing a similar window size work in other academic results, and then fine tuning to meet our data-specific needs. We chose to use Adam optimization for a few reasons. Working with audio data over 2 dimensions cause sparse gradient problems. Adam mitigates the sparse gradient problem by maintaining a per-parameter learning rate, and mitigates the noise problem by basing updates on a weighted average of recent updates (momentum). With Adam our models trained more quickly and didn't plateau as early.

THANK YOU!