Music Genre Classification Using Convolutional Neural Networks

George Danforth
gdanfor1@jhu.edu

Edward Li
eli8@jhu.edu

Abstract

We implement a convolutional neural network (CNN) and train it to classify between 10 different genres of music, attaining a result of 36% validation accuracy. We use training and validation data sets derived from the freely available Million Song Dataset.

1 Introduction

The task of automatically classifying pieces of music by genre has various applications; from helping people discover music they like more quickly and easily, to investigating and understanding the differences between musical genres in an empirical fashion. In spite of the potential applications, this task is also a particularly difficult one, due to the numerous, constantly evolving genres and sub-genres which separate pieces of music, often by subtle stylistic differences. In this study, we intend to implement a convolutional neural network (CNN) capable of classifying pieces of music by genre, based purely on their audio characteristics.

2 Related Work

2.1 Audio Feature Extraction

The foundation of performing any kind of automatic audio analysis is the ability to retreive descriptive features from an audio signal. Although many different methods of audio feature extraction have been proposed for various applications, some of the most widely employed features for the task of audio classification are mel-frequency cepstral coefficients (MFCCs), and pitch histograms [1]. MFCCs [2] are perceptually motivated features which, at a high level, can be said to describe the timbre of an audio signal over time, and are computed on individual frames of a signal by applying various transformations to their power spectra. Pitch histograms are a way of measuring the representation of discrete pitches (usually on a 12-tone scale) at different time frames throughout and audio signal [4].

2.2 The Million Song Dataset

The Million Song Dataset (MSD) is a collection of metadata and audio features over one million pieces of music spanning a variety of genres and time periods. In a collaboration between The Echo Nest and the Columbia University Laboratory for the Recognition and Organization of Speech and Audio (LabROSA), the MSD has been made freely available to the public with the goal of encouraging research into applying machine learning methods to musical data [3].

The metadata features included in the MSD consist of basic information such as the artist of the track, the year it was released, the album or compilation it was released on, etc. Additionally, each track contains a list of tags relating to the artist, which have been compiled by the MusicBrainz community.

Each track in the MSD also contains numerous audio features, both at the track level and corresponding to individual segments within a track, which have been extracted by The Echo Nest. The track level features include tempo, time signature, and key, as well higher level features created by The Echo Nest, such as "danceability", and "energy". For more fine-grained information, each track in the MSD was broken up into a number of variable time-length segments, each roughly corresponding to a note/bar onset or some other relevant musical event. The features included at the segment level include length-12 vectors of timbre features, which are described as MFCC-like features, length-12 vectors of pitch information, which are normalized pitch histograms, as well as the maximum loudness within each segment.

3 Method

3.1 Data Reduction and Preprocessing

Despite the wealth of information and features provided by the MSD, there was still a significant amout of work to do to reduce the data into a format suitable for our classification task. One of the major setbacks of using the MSD for this classification task is that it does not come with genre labels for each track. As a work-around, we compiled the most common MusicBrainz tags across the entire dataset, and manually selected the most common genres out of them, as well as tags and genre variants that appeared most often together with these genres. We then went back through the dataset and computed a score for each track based on the degree to which its tags intersected with the tags for each genre, and selected a genre based on this score. Any tracks lacking MusicBrainz tags completely were not included in the training or evaluation sets.

As part of the data reduction we also had to decide what features to use. Since we were interested in using low-level audio features only, and we intended to use a convolutional neural network to compare features at different time scales, we chose to use only segment-level features. Namely, for each segment we used the timbre vector, and the pitch histogram vector, and the maximum loudness. Since each track consisted of a variable number of segments, ranging from one into the thousands, we computed the minimum number of segments such that 95% of the data points would have a greater number of segments. Finding this value to be roughly 120, we chose the central 120 segments of each track having greater than or equal to 120 segments, and constructed our training and validation sets such that each data point had 3000 features: $120 \times (1 + 2 \times 12)$. As a final preprocessing step, we subracted the per-feature mean and divided by the per-feature standard deviation across the entire reduced dataset.

3.2 Convolutional Layer Implementation

Instead of treating each data sample as a set of 3000 individual features as we do under a regular (fully-connected) neural network, we can treat it as 120 slices of a time dimension with 25 unique acoustic features each. Since certain musical patterns of songs may last longer than others, we may wish to study the patterns these 25 features exhibit over different regions of time. This makes our dataset a good candidate for study under a convolutional neural network.

Unlike a regular neural network where neurons are arranged in single-dimensional layers, in a convolutional neural network, neurons are arranged in two-dimensional layers: the "length" dimension corresponds to a region of time within the song and the "depth" dimension corresponds to a "filter". Each filter is just a set of weights and a bias which are applied to a region of time smaller than the total output region passed by the previous layer. Each layer of the neural network may have multiple filters, all of which are of the same length and are unique to that layer.

In some sense, we can treat each individual filter as a fully-connected layer: a filter's forward for a single region of time acted on by that filter is the same as the forward pass of a full-connected layer. Hence, for a filter $i = \{1, 2, \ldots, n^{(l)}\}$ in layer l with weight $W_i^{(l)}$, bias $b_i^{(l)}$, and input $Z_{ij}^{(l)}$, we have

$$z_{ij}^{(l+1)} = W_i^{(l)} Z_{ij}^{(l)} + b_i^{(l)} \tag{1}$$

However, each filter in layer l acts over the entire output passed by layer $l-1$. In our implementation, we use a stride of 1 since we only have 120 slices of time. Hence, if layer l has a filter size of $a^{(l)}$ and layer $l-1$ produces an output length of $b^{(l-1)}$, then each filter input $Z_{ij}^{(l)}$ contains $a^{(l)}$ consecutive

elements of the output from layer $l-1$ and $j = \{1, 2, \ldots, b^{(l-1)} - a^{(l)} + 1\}$. Therefore, layer $l+1$ contains $b^{(l-1)} - a^{(l)} + 1$ elements in the length dimension and $n^{(l)}$ elements in the depth dimension.

Because of the behavior of each filter in a convolutional neural network layer, backpropagation here is handled a little differently from backpropagation in a fully-connected neural network layer. In our loss gradients for layer l, instead of having a contribution of loss to each of the neurons in layer $l+1$, we have a contribution of loss to each filter output, which corresponds to each depth element of layer $l+1$. Additionally, the contribution of loss to a depth element of layer $l+1$ from an input element of layer l is summed over up to $n^{(l)}$ different derivatives with respect to the weight vectors contained in the set of weights for the corresponding filter of layer l. Of course, input elements near the ends of the time dimension have fewer derivatives with respect to the weight vectors since they contribute to fewer elements in the output region.

4 Results

After implementing the convolutional layer, we found that adding it to our neural network incurred extremely long computation times which made experimentation with the overall layer structure of the CNN prohibitively difficult. The final network we trained consisted of a convolutional layer, a single hidden ReLU layer, and a fully-connected layer implementing softmax loss. As mentioned, training this network was very slow, taking roughly 30 seconds per iteration with a batch size of 100 training points. After training the network for 500 iterations to classify between 10 different genres of music, we achieved a validation accuracy of 36%. This result, although not remotely ideal does indicate that the model did not completely fail to learn any structure in the data, which would have resulted in roughly a 10% validation accuracy. Constructing a confustion matrix for the results (Figure 1), lends further insight into what the model was and was not able to distinguish. Inspecting the confusion matrix, it is evident that almost every genre included had a high propensity to be misclassified as both rock and pop. Meanwhile, folk, reggae, and classical were the least often misclassified. Intuitively, this result seems to make sense given that pop and rock both span many moods, tempos, time periods, etc. However, analysis and comparison of the spectral features of these genres remains a question for further inquiry.

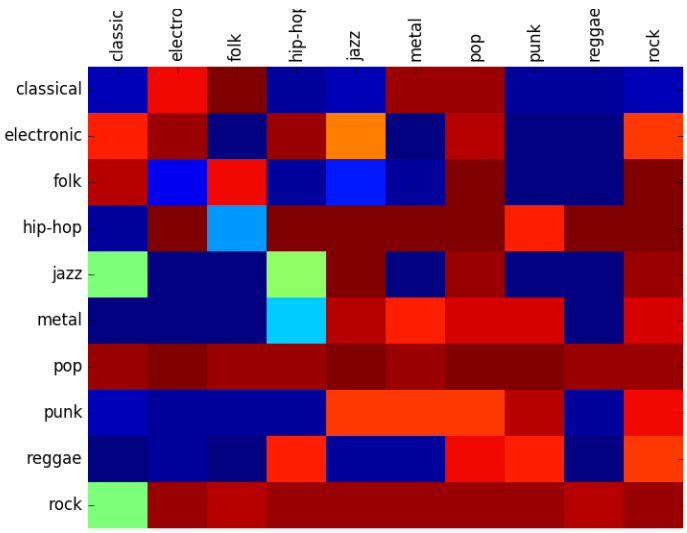

Figure 1: Confusion matrix of optimal CNN model. True values are on the vertical axis, and predicted labels are on the horizontal axis.

5 Conclusion

5.1 Discussion

In this study, we constructed a convolutional neural network with the intent of training it to classify pieces of music provided by the MSD based purely based on their audio features. Using a two-layer CNN and training it on time-separated timbre, pitch, and loudness features we acheived 36% accuracy on our validation set. Although this result is not as strong as we had hoped, performing the study still provided interesting insights into sonic similarities between different genres of music.

Carrying out this study made clear several of the limitations of our methods. First and foremost, we were perhaps somewhat naive not to anticipate the extraordinary computation times incurred by processing training points with 3000 features with a convolutional layer using only a single processor core. This was a major setback in our attempt to glean more meaningful information about the sonic characteristics of distinct musical genres, and to experiment with different network structures in order to create a truly performant model. Another limitation of the method that is worth mentioning is the fact that, given the size of the data, it is entirely possible that much of our training data was either not classified accurately or with sufficient precision. Since our chosen dataset had to be labeled and reduced by a somewhat manual process, its accuracy cannot be guaranteed.

5.2 Further Work

This study leaves open many avenues for further study. One of the most glaring issues with our method was the prohibitive computation time incurred by our implementation of a convolutional layer. As a followup to this study, it would be valuable to investigate optimization methods for training a CNN, perhaps via CPU parallelization or through the use of GPUs. Further optimization of our model would allow us to explore different network configurations and improve our results, which, in turn would provide us with further insight into the sonic differences between distinct musical genres. Another area for further study would be to investigate how the performance of the model is impacted by including more and more genres and sub-genres in the dataset. As music continues to evolve, there are ever more characteristics between which a model such as ours must distinguish.

References

[1] Zhouyu Fu, Goujun Lu, Kai Ming Ting, Dengsheng Zhang. *A Survey of Audio-Based Music Classification and Annotation*. IEEE Transactions on Multimedia, 2010.

[2] S. Davis, P. Mermelstein. *Experiments in Syllable-Based Recognition of Continuous Speech* IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 28, pp. 357–366, 1980.

[3] Thierry Bertin-Mahieux, Danile P.W. Ellis, Brian Whitman, Paul Lamere. *The Million Song Dataset*. Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

[4] George Tzanetakis, Perry Cook. *Musical genre classification of audio signals*. IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293–302, 2002.