Coda: Instrument Recognition in Audio Clips

ECE 324 Final Report

Lisa Li (1003924532)

Mengyu Yang (1003922767)

December 3, 2019

Word count: 1966
Penalty: 0%

Yes, we agree to have our video, final report, and source code posted.

Introduction

Coda is a deep learning model which classifies the main instruments within multi-instrumental music. In this context, the main instruments are defined as those which sound the most prevalent, as determined by the authors of the dataset used to train the model.

The significant rise in the popularity of music streaming services has led to a revolution in how music is categorized, with ever-increasing specificity for the purpose of personalized recommender systems. Coda contributes to this area by providing a tool for sorting music based on instruments, a key factor in acoustic-based genres such as classical and jazz.

Deep learning applies well to this situation since audio contains a significant level of variation and noise. Common examples include background noise and tuning deviations, which may alter similar pieces of audio. As a result, the flexibility of a deep learning model can be used to determine the characteristic features associated with various instruments.

Illustration

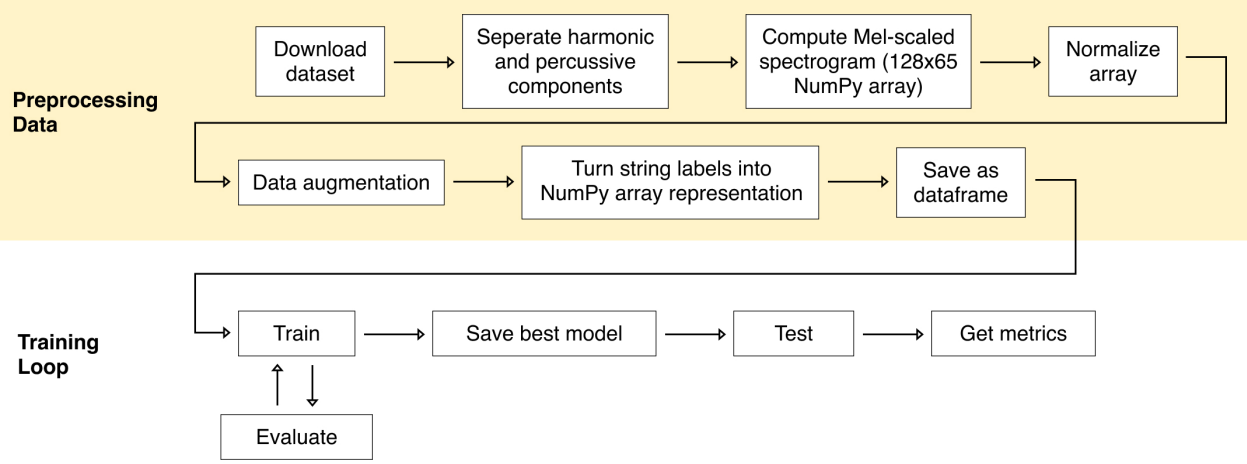

Figure 1: *Pipeline of our software.*

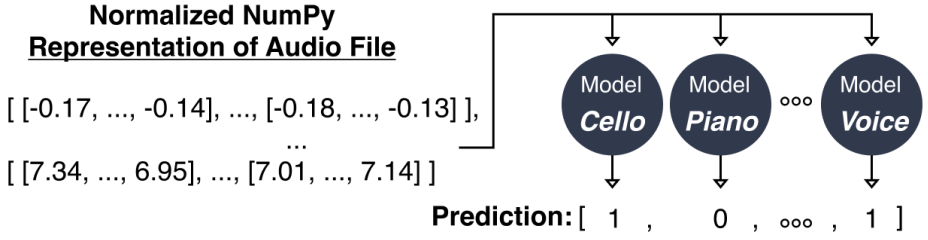

Figure 2: *General overview of the structure of our final model, the multi-binary MLP (MBMLP). The audio file (represented as a normalized NumPy array) is the input for each of the 11 binary MLPs, who return either 1 or 0, giving a final output of an 11-element list for each of the 11 instrument classes.*

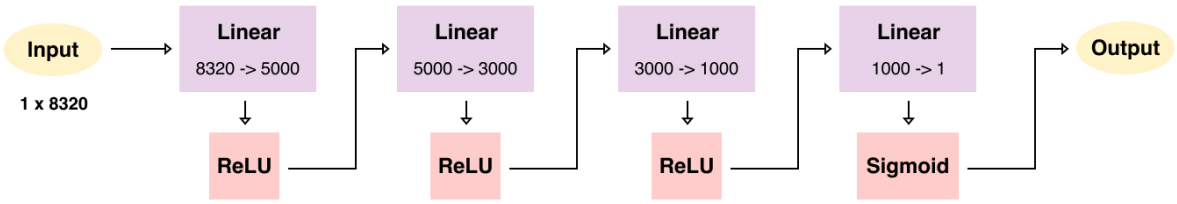

Figure 3: *The architecture for each binary MLP from Figure 2.*

Background

Gomez et al. [1] explored the classification of solo instruments in jazz ensemble music. To begin, they first established a baseline model proposed by Han et al. [2] as shown in Figure 5. Then, they attempted to improve this model for their specific jazz use-case by preprocessing the audio to separate the solo instrument from the accompaniment instruments which led to a better generalization of unseen data.

Additionally, Hershey et al. [3] explored the performance of popular CNN architectures for general audio classification. Models tested include AlexNet, VGG, Inception V3, and ResNet-50. The baseline model consisted of a simple, fully connected model with 3 layers of 1000 units each followed by a ReLU activation. The underlying motivation was that audio classification is similar to image classification, where the models can detect specific features from the spectral representation of audio files and relate those to the corresponding labels.

Data Collection/Processing

Audio data was taken from the IRMAS dataset, containing 6,705 training examples and 2,874 test examples spanning across 11 instrument classes. Each training example contains a 3 second long *.wav* file with a single corresponding label in a *.txt* file, seen in Figure 4. Each test example contains an audio file of 5-20 seconds and contained up to 5 labels. All of the audio files were hand-labeled by the authors, who ensured that the annotated instruments are the same throughout the whole excerpt. The statistics of the training set is shown in Table 1.

Figure 4*: An example of a piece of training data, with audio file (left) and corresponding single label (right). The test data has similar format except for the fact that each label file may contain multiple labels. "pia" is an abbreviation of piano.*

Instrument Class	Number of Training Examples
Cello	388
Clarinet	505
Flute	451
Acoustic Guitar	637
Electric Guitar	760
Organ	682
Piano	721
Saxophone	626
Trumpet	577
Violin	580
Voice	778

Table 1: *Statistics for number of training examples for each of the 11 instrument classes.*

A Python script was written to load each audio file into the LibROSA library for further processing while parsing the *.txt* file for the label.

To begin preprocessing, the harmonic and percussive components of the audio were separated to further isolate the main instrument. Using only the harmonic components, we computed the Mel-scaled spectrogram, which is the representation of audio in the frequency domain scaled by a non-linear function. The output is a 128x65 NumPy array, where each column represents 0.05 seconds in time while the rows are frequency bins. The array columns were then normalized to a mean of 0 and standard deviation of 1 using:

$$col = \frac{col - mean_{col}}{sd_{col}}$$

Due to the small number of training examples for each class, data augmentation was also implemented. We added Gaussian noise to the Mel-scaled spectrogram and also slowed down the audio to 0.5 speed before cropping to 3 seconds. These actions essentially triples our data.

The data was saved as a Pandas dataframe, where the first column contains the NumPy arrays and the second column contains the labels. This pipeline is visually represented in Figure 1.

Baseline Model

The baseline model is a fully-connected MLP with 3 layers, 1000 neurons each, 64 batch-size, learning rate of 0.0001, and 50 epochs. The first two layers use a ReLU output while the final layer uses a sigmoid. The 128x65 NumPy arrays were reshaped to 1x8320 to be a suitable input for the MLP.

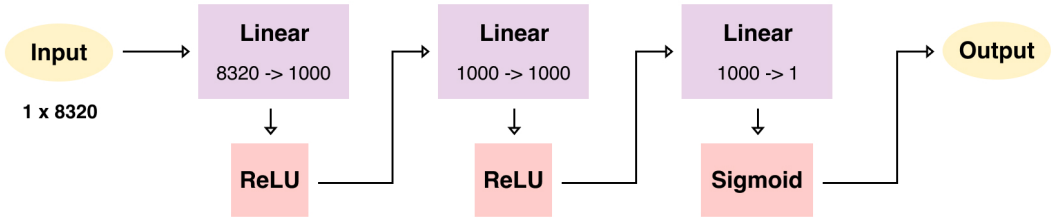

***Figure 5**: 3 layered MLP used for baseline model.*

Architecture

As training examples contain one label while test examples contain multiple, we decided to construct a multi-binary MLP (MBMLP) system consisting of a fully-connected MLP binary classifier for each instrument class, seen previously in Figure 2.

Developing the model consists of training 11 MLP binary classifiers. For each instrument class, we collect all of the training data which contains the label of our target instrument. Then, out of the remaining training data of the other 10 instruments, we randomly select an equal number of examples to balance our training set.

We implemented a coarse-to-fine hyperparameter search strategy to fine tune our model. This ultimately resulted in 4 fully connected layers starting with 5000 neurons and then going to 3000, 1000, and finally 1 neuron for the final binary output. We used a 64 batch-size, learning rate of 0.0001, and 50 epochs. BCELoss was chosen as the criterion and Adam was chosen as the optimizer.

The resulting MBMLP is then able to predict multiple instruments. Given the same processed audio array, each individual binary classifier will make a prediction of whether its specific instrument is present or not. As a result, the output of the MBMLP model is an 11-element list which displays the predictions of each binary classifier.

Quantitative Results

Figure 6 shows the accuracy and loss for both training and validation of our final MBMLP model. Analyzing accuracy, we see that training accuracy averages at around 77% while the validation accuracy is around 70%, with the highest value being 75.3%. Given the context of 11 instrument classes, these accuracy values are quite good, since a randomly guessing model will only achieve a 9% accuracy.

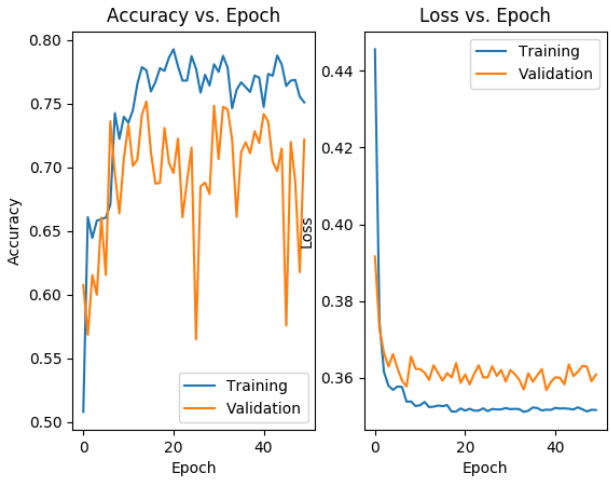

***Figure 6**: Training and validation plots of MBMLP loss and accuracy.*

Additionally, in both the loss and accuracy plots, the validation values closely follow the training values, indicating that the model is not overfitting. As mentioned above, the difference in training and validation accuracy differs by around 7%, and we can determine visually that the loss does not overfit either. Minimal overfitting is further evident by the fact that the test accuracy of this model is 68.4%. Overall, we can conclude that the MBMLP model generalizes well.

One significant characteristic of the validation values is the large fluctuations, especially apparent in the accuracy plot. Although the reason for this is not quite clear, it may be useful for future reference to train the model for more epochs, as this has empirically been shown to generate a smoother curve.

Cello
[[2201 123]
 [153 107]]

F1 Score: 0.941

Clarinet
[[2434 42]
 [51 57]]

F1 Score: 0.981

Flute
[[2335 64]
 [38 147]]

F1 Score: 0.979

Guitar - Acoustic
[[1606 290]
 [192 496]]

F1 Score: 0.870

Guitar - Electric
[[1696 47]
 [177 664]]

F1 Score: 0.938

Organ
[[2123 81]
 [56 324]]

F1 Score: 0.969

Piano
[[1540 58]
 [96 890]]

F1 Score: 0.952

Saxophone
[[2168 72]
 [59 285]]

F1 Score: 0.971

Trumpet
[[2357 63]
 [21 143]]

F1 Score: 0.982

Violin
[[2047 186]
 [159 192]]

F1 Score: 0.922

Voice
[[2433 3]
 [1 147]]

F1 Score: 0.999

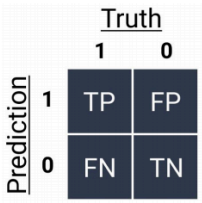

***Figure 7**: Confusion matrix of each instrument.*

Finally, Figure 7 shows the confusion matrices of the 11 instrument classes after running the MBMLP for 50 epochs. We see the larger values are along the diagonal, further suggesting the accuracy of our model. To minimize both false positives and negatives, the F1 score for each matrix was calculated, with most of them in the mid-to-high 0.90 range.

Qualitative Results

Looking at the confusion matrices from Figure 7, we notice that instruments with over 100 false positive and negatives are the cello, acoustic guitar, electric guitar, and violin. Accordingly, their F1 scores are also the lowest amongst all classes. These aforementioned instruments are all strings, which suggests that it is harder for the model to discern amongst the same instrument category. In contrast, voice has the highest F1 score, suggesting it is easier for the model to distinguish due to its distinct timbre and having no other classes sound similar to it.

To obtain more insight into the high number of false positives and false negatives amongst string instruments, the model was given a test dataset containing string instruments. Some results are shown in Table 2.

	Prediction	Label	Interpretation
1	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1] Electric guitar, voice	[0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1] Electric guitar, violin, voice	Violin may have been “lost” behind the guitar
2	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0] Electric guitar	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0] Acoustic guitar	Misclassified guitar
3	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0] Electric guitar	[0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0] Acoustic guitar, Electric guitar	Only “heard” one guitar
4	[1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0] Cello, acoustic guitar	[1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0] Cello, violin	Thought violin was a guitar

***Table 2**: Comparison of predictions and labels containing string instruments.*

Although they may sound different to an experienced human ear, the MBMLP model solely makes its prediction based on the frequency values of the instruments. String instruments can share many of the same frequencies, especially when played within the same range of pitches. Although there are inherent frequency differences between string instruments in terms of overtones, these small distinctions may be lost due to the existence of other overlapping instruments. This is especially apparent in Entry 3 of Table 2, where the model only ‘heard’ the more powerful electric guitar, which obscured the small distinctions in frequency that differentiates it with the acoustic guitar.

Discussion and Learnings

Given the relatively high difficulty of the task which our model attempts to complete, we are satisfied with the performance of our model. A main contribution to the difficulty of this project is that audio is inherently messy, as there can be many different types of variation such as background noise and tuning deviations which can lead to difficulty in generalizing a model.

Compounded with this, classifying one instrument out of many playing at the same time is much harder than classifying a lone instrument. Features from multiple instruments overlap within the frequency domain, as different instrument sounds share many of the same frequency values. As a result, this means for a given frequency strength, the model must determine which instruments contribute to it as well as how much each of those instruments contribute.

We also noticed that most of the literature utilizes a single multi-class model to classify one main instrument only, while we are classifying multiple dominant instruments. Intuitively, this is a much more difficult problem, which may explain why our accuracies are lower than those found in literature. During our model exploration, we discovered that our MBMLP model performed better than traditional single multi-class models such as CNNs and RNNs. This suggests that given addition time and resources, coupled with our MBMLP model, we may get exceedingly better accuracies.

Going back to our results, we initially expected that the CNN and RNN would perform better than our MLP. The discrepancy may be due to the nature of image vs. audio inputs. In our data preprocessing, the use of Fourier transforms to extract important and distinguishing features within the audio is analogous to applying a CNN on the raw audio data. Thus, the MLP performed better as it acted as the linear layer following the ‘convolutions’ which the Fourier transform produced. It also shows that performing a CNN or RNN on already feature-extracted data proved to negate its effectiveness and worsen the results.

For future reference, we may consider using our baseline MLP on the raw audio before the Fourier transform so we have a better reference to compare against. Additionally, we would look into additional methods of data preprocessing to further isolate the main instrument and get a higher accuracy.

Ethical Framework

The implementation of Coda within music streaming services will allow for greater specificity in the categorization and playlist curation of music. Since the underlying goal of recommender systems is to introduce users to new music they enjoy while minimizing those they do not, Coda may further isolate artists who specialize in less popular instruments. As a result, this negatively affects the autonomy of individual artists, who become more alienated should they choose a less mainstream musical path. If we specifically examine Spotify, one of the largest music streaming services, whose playlist recommendations play an important factor in what music becomes popular [4], the impact to an artist’s success can be significant.

On the other hand, a smarter recommender system can improve the user experience of Spotify's massive customer base of 99 million paid users. This has a positive effect on Coda's principle of beneficence and justice, as it provides a useful tool to a large population.

Finally, Spotify records data on which tracks get skipped and excludes those with high skip rates from future playlists. This is shaping songwriting and production as shorter tracks with flashy intros perform better in the algorithm [5]. Essentially, Spotify's current algorithm is unnaturally influencing the evolution of music. This has significant impacts on music producers, who must adapt to the faster-changing tastes in music. This introduces harm to these stakeholders, which also negatively impacts Coda's non-maleficence principle.

Sources

[1] Gomez, J., Abeßer, J. and Cano, E. (2018). JAZZ SOLO INSTRUMENT CLASSIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS, SOURCE SEPARATION, AND TRANSFER LEARNING. *Proceedings of the 19th ISMIR Conference, Paris, France, September 23-27, 2018*. Available at: http://ismir2018.ircam.fr/doc/pdfs/145_Paper.pdf

[2] Han, Y., Kim, J. and Lee, K. (2016). Deep convolutional neural networks for predominant instrument recognition in polyphonic music. [online] *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. Available at: https://arxiv.org/pdf/1605.09507.pdf.

[3] Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., … Wilson, K. (2017). CNN architectures for large-scale audio classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Available at: https://arxiv.org/pdf/1609.09430.pdf

[4] Aguiar, L. and Waldfoge, .J. (2018). Platforms, Promotion, and Product Discovery: Evidence from Spotify Playlists. *Joint Research Centre*. Available at: https://ec.europa.eu/jrc/en/publication/eur-scientific-and-technical-research-reports/platforms-promotion-and-product-discovery-evidence-spotify-playlists

[5] https://www.theguardian.com/music/2019/apr/28/streaming-music-algorithms-spotify

Iqbal, N. (2019). Forget the DJs: Spotify playlists are the new musical starmakers. *The Guardian*. Available at: https://www.theguardian.com/music/2019/apr/28/streaming-music-algorithms-spotify