Recognizing instruments with neural networks

Elon Glouberman (eglouberman@ucla.edu)
Department of Psychology, 502 Portola Plaza
Los Angeles, CA 90095 USA

Trevor Harrison (Tkharrison@ucla.edu)
Department of Psychology, 502 Portola Plaza
Los Angeles, CA 90095 USA

Abstract

Can we mimic the biological audio functions of humans by training a neural network to identify instruments? This paper analyzes whether spectral features in the absence of temporal ones are sufficient to classify a sound's instrument family. More specifically, we hypothesized that spectral features combined with temporal features would achieve the best accuracy when trained on a neural network. However, we found that spectral features achieved 94 percent accuracy when input through a convolutional neural network (CNN). The CNN received input from over 20,000 acoustic instrument samples from Google's NSynth dataset and was given the task to classify instruments according to the Mel-Frequency Cepstral Coefficients (MFCC) of the audio source. Our CNN performed better than a simple multi-layer perceptron model that was trained on both spectral qualities and temporal ones like decay type, velocity, and distortion.

Keywords: Audio classification, Convolutional Neural Network, Multi-layer perceptron, MFCC, instrument classification

Introduction

What is sound in music? Musical instruments are unique to each other for several reasons. Primarily, instruments contain varying harmonic overtones, which make up its timbre. Notes from different instruments are composed of unique combinations of harmonic frequencies. Sound can be identified by being deconstructed into its constituent frequencies.

Spectral features are obtained by converting the time-based signal into the frequency domain using the Fourier Transform. Biological evidence demonstrates spectral feature detection occurring in nature. The basilar membrane of the cochlea, found in the inner ear, is tonotopically organized with one best frequency increasing the firing rate of a specific neuron. This process is similar to Fourier Analysis which breaks down a sound into harmonic frequencies. In audio pre-processing for machine learning and neural network tasks, this can be easily done using the Librosa library. Previous studies have used spectral data to classify monophonic instruments, resulting in 70 percent accuracy (Agostini et al., 2003). Other works have also used spectral data to recognize a chorus of musical instruments playing at the same time, but resulting in 53 percent accuracy (Essid, Richard, & David, 2005).

Since many neural network projects have succeeded with classification tasks using spectral features, we decided to first analyze how temporal features can also be utilized to accomplish the same task. Human beings are proven to recognize non-percussive instruments using the attack of its envelope. One study found that multi-layer perceptrons could identify instruments with a 93% accuracy and 80% accuracy only looking at attack (Toghiani-Rizi & Windmark, 2017). Without attack, the system was only 71% accurate. This indicates temporal features to be salient in instrument identification.

Dataset and Training Examples

Google's NSynth dataset contained around 300,000 music notes from 12 different instrument families with a mix of three different instrument styles. The dataset also came with wav files corresponding to each note and having a normalized length time of four seconds. From the Nsyth dataset, we randomly sampled 3,000 instruments from each of the eight acoustic instrument sources, giving us a total of 24,000 samples. We then split the samples 80-20 to achieve a train-test split. To test the effect of temporal features, we utilized many of the features in the NSynth dataset including fast decay, long release, velocity, percussive, and reverb. Table 1 demonstrates each of the features used. Spectral qualities supplied by NSynth include brightness, darkness, and multiphonic features. We combined these two types of features, in the absence of the audio data, to test whether the combined spectral and temporal metadata qualities were sufficient in recognizing instrument families.

Table 1: Spectral and Temporal qualities trained on multi-layer perceptron

Spectral	Temporal
Brightness, darkness, multiphonic	Fast decay, long release, velocity, distortion, percussive, reverb, tempo-synced, non-linear envelope, pitch

To test spectral qualities, we utilized the audio data alone. Since preprocessing hundreds of thousands of audio files takes up an extraordinary amount of time, we randomly sampled 3,000 instruments from eight acoustic instrument sources, giving us a total of 24,000 samples. We then split the samples 80-20 to achieve a train-test split.

We utilized a Python-compatible audio preprocessing library, Librosa, to convert the digital waveform of the audio file into three feature types: melspectrogram, log-melspectrogram, and Mel-Frequency Cepstral Coefficients (MFCC). All of these features utilize Fourier Transform to break up the overall frequencies of the waveform into constituent smaller frequencies, which make-up the blueprint of a waveform's audio source. All features' frequencies are also scaled according to the Mel scale, a biologically based scale that mimics how human beings perceive frequencies in relation to other frequencies. Melspectrogram and log-melspectrogram have a lot of overlapping data, which can introduce noise to any learning model. MFCC's, however, apply a discrete cosine transform to its filter bank coefficients, effectively removing and decorrelating high energies to a more compact lower range.

Neural Network Architecture

We generated two multi-layer perceptrons and one convolutional neural network. The first network had the task of learning binary classifications between eight different instruments. We thus had a simple multi-layer perceptron with 3 layers, having an input layer of 12 (relu activation), hidden layer of 6 (relu activation), and output layer of 2 (sigmoid activation). Our loss function utilized sparse categorical cross entropy, compiled with an Adam optimizer, and trained using 30 epochs and batch size of 12.

Next, we modified our multi-perceptron to handle multi-class classification. We thus had a simple multi-layer perceptron with 3 layers, having an input layer of 12 (relu activation), hidden layer of 12 (relu activation), and output layer of 12 (softmax activation). Our loss function utilized categorical cross entropy, compiled with an adam optimizer, and trained using 8 epochs and batch size of 12.

Binary Classification

Our binary classification network achieved fairly accurate results. We tested each instrument against each other and resulted in accuracy scores ranging from 0.69 to 0.96. Since there were eight instruments, there were a total of 28 comparisons. The highest average accuracy score when compared to all other instruments were vocals with a mean of 86 percent accuracy. The lowest average accuracy instrument when compared to all other instruments was reed with a mean 76 percent accuracy. Figure 1 demonstrates a matrix that represents the one versus one scores by binary classification.

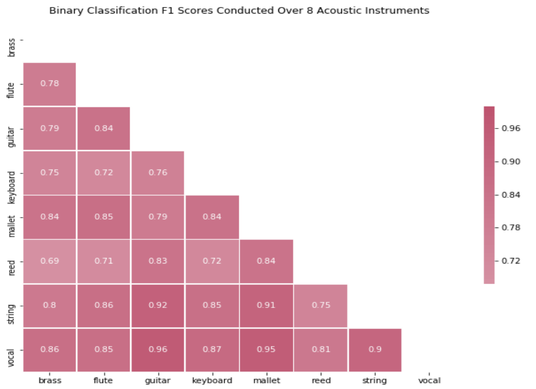

Figure 1: Matrix to demonstrate binary classification model accuracy across all instruments

Multi-Class Classification

When testing multiple classes on our network, we did not achieve highly accurate results. Overall, we achieved a 40 percent accuracy score on our test set. When analyzing the resulting confusion matrix, the vocal instrument had the best combined recall and precision score with an 81 percent accuracy. Mallet came in second with 52 percent accuracy, and keyboard had the worst score with 18 percent. In order to truly understand how the system works, we decided to remove data and see how it performed on our model. Given that we wanted to test the importance of temporal features, we first removed temporal data from our input features.

Removing Temporal Data and its Effect Because of the experiment that demonstrated multi-layer perceptrons achieving highly accurate scores with "attack" as the primary feature (Toghiani-Rizi & Windmark, 2017) we first removed "percussive", which most resembles a fast attack. When removed, the system did not do much worse on the test set, resulting in a 39 percent accuracy. Next we removed "fast decay" which can be a strong indicator of instrumentation. The effect of removing fast decay along with percussive decreased the accuracy model a little bit more, resulting in around 36 percent accuracy on the test set. The system did not officially "break" since it still performed better than random chance.

Lastly, to test the effect of removing spectral features, we removed "bright" and "dark" from our feature set. The model still performed as accurately as the original full-feature model, yielding a score of 39 percent. The fact that network performance went down by more points when removing temporal features as opposed to spectral features demonstrated that our system was more sensitive to temporal information. Figure 2 summarizes our breakdown results.

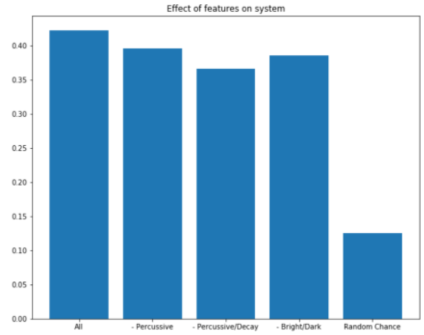

Figure 2: Bar plot that demonstrates the effect of different features on the multiclass model's accuracy.

Convolutional Neural Network Architecture

Convolutional Neural Networks (CNN) have empirically been proven to be successful when training to identify patterns and features, specifically during image recognition and audio classification tasks. Researchers claim that the reason this is the case is because CNN's "mimic the biological structure of the human sensory system… [performing] different transformations of the incoming signal, hence "learning" information in a distributed topology (Huang, 2018).

Our CNN was modelled after a publication found on *Medium* by author Mike Smales[1]. We tested three different inputs on our CNN, giving us three different input sizes: melspectrogram, log-melspectrogram, and MFCC. The CNN has four Conv2D layers with a dense layer at each output. The layer filters increase in size from 16, 32, 64, 128, with a kernel size of 2. Each layer is activated by a relu activation function and a 20 percent dropout to reduce overfitting. Our final layer has an output of size eight, where each index represents the probability that the input is one of our eight instruments. We apply a softmax activation function in order to have all probabilities add up to one. Lastly, the CNN is compiled using an Adam optimizer, cross entropy loss function, and trained using 72 epochs and batch size of 128.

Melspectrogram as Input

Melspectrograms are visual representations of a sound wave's frequencies calculated using Fourier Transform, varying over time. Frequencies are displayed on vertical axis, with time on horizontal axis, and amplitude is represented by intensity or color of the image. The frequencies are scaled according to the Mel scale, a biologically based scale that mimics how human beings perceive frequencies in relation to other frequencies.

We achieved an accuracy of 72 percent, with vocal instrument having a combine recall/precision accuracy of 96 percent. The system did have some trouble identifying reed with an accuracy of 46 percent. When reed was input into the network, 30 percent of the time the network guessed it to be brass.

Log-Melspectrogram as Input

Log-melspectrograms are melspectrograms but a logarithmic compression on the magnitude of time frequency bins is performed. Our motivation for doing so was due to an empirical study demonstrating this format to be the best preprocessing technique for audio classification tasks using a convolutional neural network (Choi et al., 2017). This can be attributed to the fact that by compressing the magnitude of the bins, the magnitudes are then converted to a gaussian distribution.

Our results improved from our previous model, achieving an accuracy of 90 percent. Vocals, brass, and keyboard seem to be the most distinct to the CNN, as they all achieved independent high accuracies.

Mel-Frequency Cepstral Coefficients as Input

Lastly, we trained our CNN utilizing Mel-Frequency Cepstral Coefficients as the input. MFCC's apply a discrete cosine transform to a logarithmic compression of its filter bank coefficients, thereby decorrelating repeated energies and creating a compact and unique fingerprint.

Our MFCC model achieved even better than log-melspectrogram, scoring a 94 percent accuracy. All instruments had an accuracy score greater than 90 percent, with vocals having 100 percent accuracy and strings being second highest with 97 percent. Reed achieved the lowest score, 83 percent. The confusion matrix in figure 3 demonstrates our exact results.

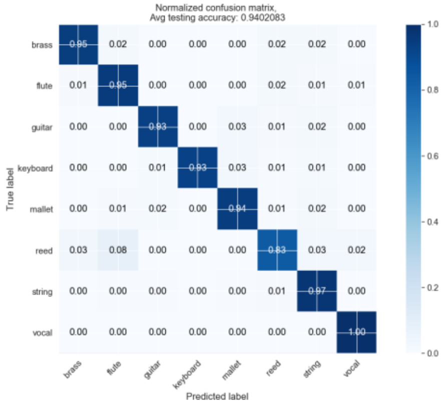

Figure 3: Normalized confusion matrix to demonstrate the accuracy of MFCC model. Very high percentages run across the diagonal of the matrix, indicating a high accuracy.

Breaking the Model To truly understand how the MFCC model works, we decided to test it on non-acoustic instrument groups as well as on reversed MFCC data.

As mention earlier, Google's NSynth dataset came with three different instrument sources: acoustic, electronic, and

[1] Link to publication: https://medium.com/@mikesmales/sound-classification-using-deep-learning-8bc2aa1990b7

synthetic. If the model did very well on acoustic instruments, would it be able to perform well on electronic and synthetic as well? We found that our accuracy declines significantly when trying to generalize our model to other instrument types. When input electronic instrument MFCC data, the system achieved an accuracy of 35 percent. For synthetic instruments, it performed even worse at 22 percent. This tells us that even though an electric guitar may sound similar to an acoustic guitar to the human ear, their MFCC data are uniquely different and the system has troubling handling this extra noise. However, the performance does better than random chance, leading us to believe that there are latent similarities between acoustic, electronic, and synthetic instrument families.

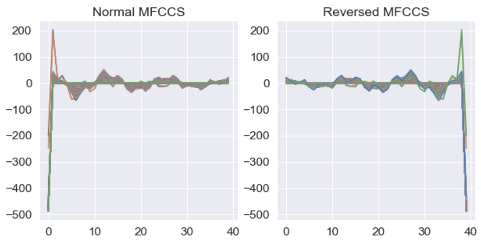

Figure 4: A photo representing the difference between regular MFCC input versus reversed. Both are symmetrical.

If MFCC data is truly measuring the effect of spectral data, then a reversal of that data should yield accurate results as well (figure 4). When reversing MFCC input into our accuracy MFCC model, we achieved a very low result of 28 percent. Mallet and flute were the only instruments to achieve higher than 50 percent accuracy. The inaccurate behavior the MFCC reversal demonstrates that while we attempted to isolate spectral from temporal qualities for our model architecture, temporal data was in fact salient in the MFCC feature. Thus, we need to better find a method to isolate spectral qualities to test our hypothesis more accurately. Figure 5 demonstrates the model breaking over different input types.

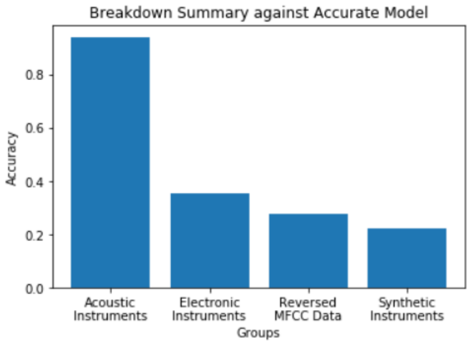

Figure 5: As we try to generalize to other instrument sources and reversed MFCC data, our model breaks down.

Discussion

Our results show that a convolutional neural network processing (mostly) spectral data is more accurate for most instrument recognition tasks than a multi-layer perceptron trained on multiple types of features. This suggests that biological audition could reach a high-level performance solely by processing spectral features.

Our experiments are limited in describing both human and machine intelligence because we focused on performance instead of tight control. Furthermore, we did not use perceptron models that captured audio data and future experiments should attempt to utilize simple networks.

Additionally, the features we used in both neural networks included both spectral and temporal qualities, due to the difficult nature to isolate the two. Future experiments should do a better job to isolate both variables. Lastly, more work could be done to build system that more accurately mimics biological constraints such as mixing acoustic, electronic, and synthetic audio sources into one model. Another way to do this would be to capture non-normalized data from the real world with plenty of background noise, which mimics the environment where humans would be hearing those same notes.

References

Agostini, Giulio, Maurizio Longari, and Emanuele Pollastri. (2003). "Musical instrument timbres classification with spectral features." *EURASIP Journal on Advances in Signal Processing* 2003, no. 1. 943279.

Choi, K., Fazekas, G., Cho, K., & Sandler, M. (2017). A Comparison of Audio Signal Preprocessing Methods for Deep Neural Networks on Music Tagging. Arxiv.org. Retrieved 23 June 2018, from https://arxiv.org/abs/1709.01922

Essid, S., Richard, G., & David, B. (2005). Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(1), 68-80.

Huang, N., Slaney, M., & Elhilali, M. (2018). Connecting Deep Neural Networks to Physical, Perceptual, and Electrophysiological Auditory Signals. *Frontiers in neuroscience*, *12*, 532. doi:10.3389/fnins.2018.00532

Toghiani-Rizi, B., & Windmark, M. (2017). Musical Instrument Recognition Using Their Distinctive Characteristics in Artificial Neural Networks. *arXiv preprint arXiv:1705.04971*.