

Classification of Musical Instruments by Sound



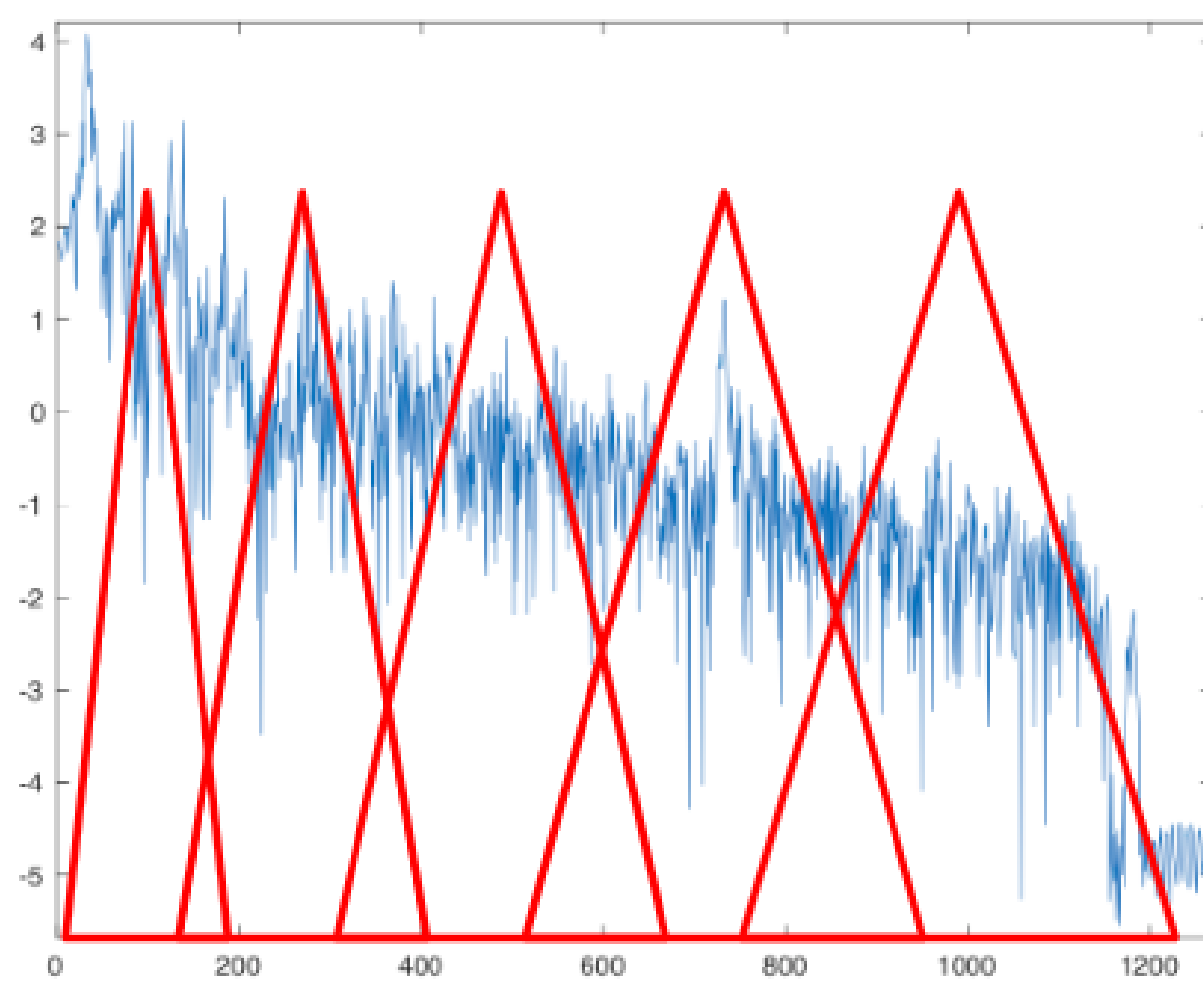
Abstract:

Sound classification is an interesting problem to tackle due to how varying sound can be and the issue of needing to identify what object produced the sound. Using deep learning and machine learning, we created several learning models to classify the type of musical instrument from a 10 second sound clip and compare the performance against each other. The best model we have is the full-connected neural net with reduced classes.

Data

The 10 second sound clips comes from Google's Audioset, which pulls music clips from Youtube. The data provided by Audioset include the video link, the seconds within that video that was used, and the label(s) for what instrument(s) is being played. Due to Audioset containing data irrelevant for what we are doing (i.e. music genres, music moods), we reduced the dataset to only contain data pertaining to the 89 labels. The Google dataset is split into three parts: balanced, unbalanced, and test. There are 59 examples per class in the balanced training set, including multi-label samples. There is also an unbalanced dataset available, with many more examples of some classes. However, Google has cautioned that some of the labels on the unbalanced set may not have been rigorously validated. The test set also has 59 examples of each class.

Mel-Frequency Cepstrum Coefficients

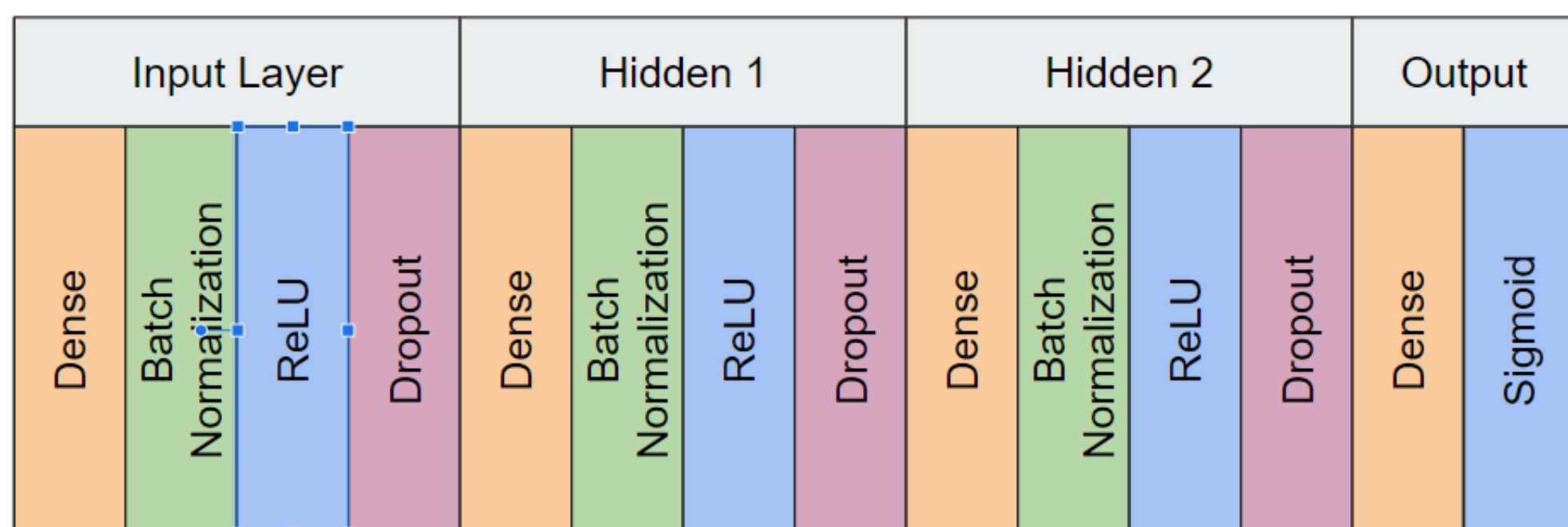


Features

There are a total of 128 features for every second in the clip, so in total we have 1280 features for each video input. To generate these features, we use the Mel-Frequency Spectrum transform, which is commonly used in audio applications to approximate human hearing. This transform works by taking the FFT of the signal, using a set of triangular filters placed on a log scale to resample to frequency domain, taking the log of the results, then taking a discrete cosine transform of the results.

Models

Fully-Connected Neural Network - After using an automated hyperparameter optimization tool, we selected a model containing 1 input layer and 2 1024-neuron hidden layers with 0.1 dropout and RELU activation, followed by a dense layer using either softmax activation for the output. We also trained a separate fully-connected neural network that grouped categories, such as singing, organs, drums and bells to check the improvement on classification.



Softmax

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \text{for } i = 1, \dots, K$$

Relu

$$\text{relu}(\mathbf{x})_i = \begin{cases} x_i & \text{if } x_i > 0 \\ 0 & \text{if } x_i \leq 0 \end{cases}$$

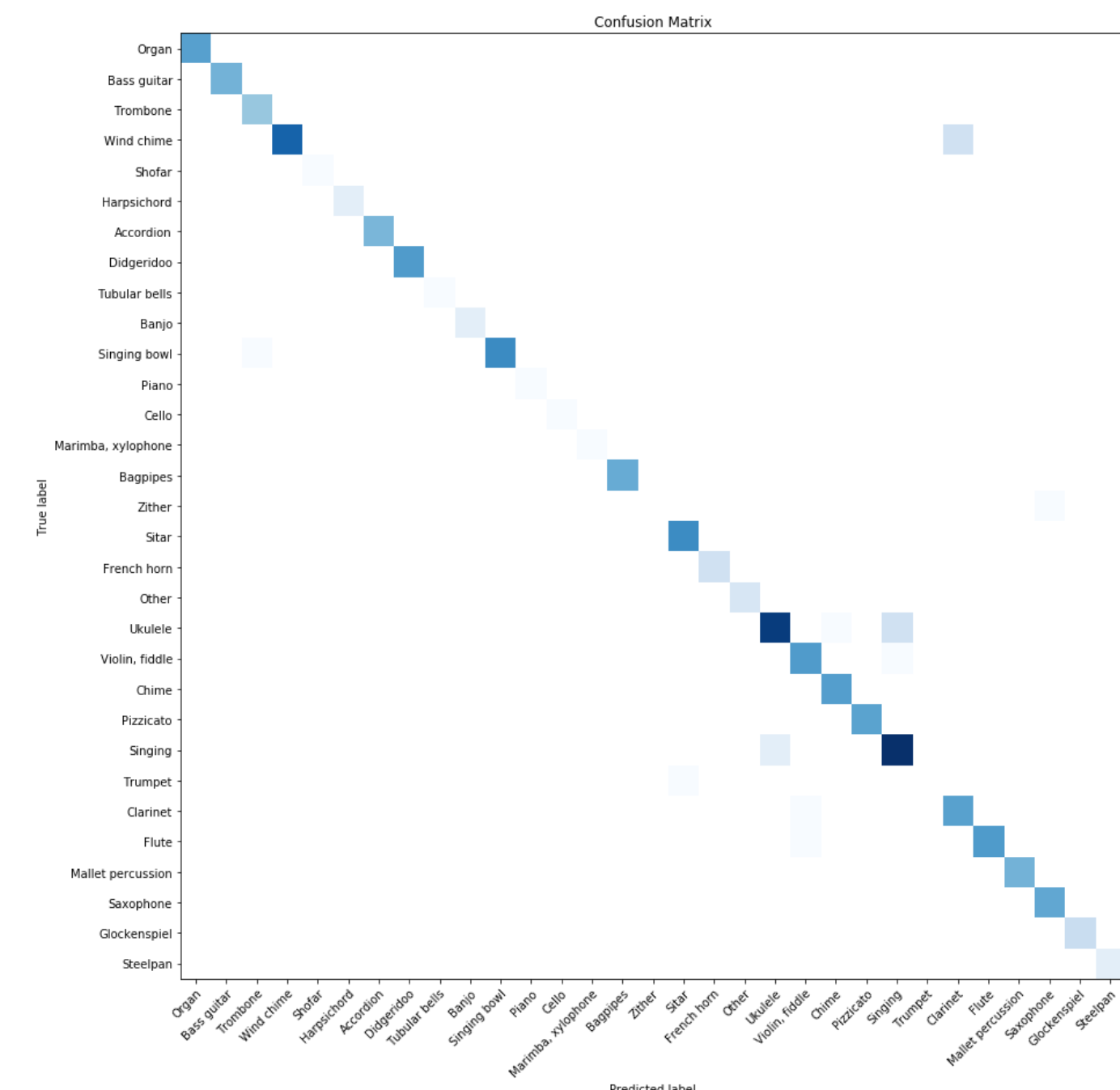
Random Forest - We reshaped the input data to a 1x1280 input vector, then used a random grid optimizer for the hyperparameters.

SVM One vs Rest - It creates a linear SVC for each individual class vs all other classes. We get a score that represents the distance from each hyperplane for each sample.

Results

Model	Testing Acc.
	Training Acc.

Model	Top 1	Top 2	Top 3	Top 4	Top 5
FC-NN (reduced classes, single label)	68%	82%	87%	92%	93%
	100%	100%	100%	100%	100%
FC-NN (reduced classes, multi-label)	70%	83%	88%	91%	93%
	100%	100%	100%	100%	100%
FC-NN (single label)	46%	61%	69%	73%	76%
	98%	99%	100%	100%	100%
FC-NN(multi-label)	52%	63%	70%	75%	79%
	92%	97%	98%	99%	100%
Random Forest	61%	N/A			
	100%				
SVM One vs Rest	38%	50%	56%	61%	64%
	100%	100%	100%	100%	100%



Discussion

The FC-NN results were better than the random forest results, and both were significantly better than the SVM-based One vs. Rest method. The neural network has a number of advantages in this situation. In particular, the neural network architecture lends itself more readily to leveraging relationships between frequencies, which is especially important in audio analysis. In addition, the dropout functions available for neural nets help protect against overfitting. SVMs have a similar problem: they work on defined hyperplane boundaries, whereas NNs can more easily work with numerical relationships between input features.

Combining similar instruments into the same class helped significantly. Including only segments with a single label reduced ambiguity, but note that we cannot straightforwardly compare the accuracy of the single- and multi-label cases.

Future

Given more time, we would create a separate classifier for all grouped categories in reduced classes case to create hierarchical classification. We would also experiment more with pre-processing and filtering of the audio clips we were given instead of using the default pre-processing provided. It would also be interesting to compare these models against the performance of a CNN.

References

- [1] P. Hamel, S. Wood, and D. Eck, "Automatic Identification of Instrument Classes in Polyphonic and Poly-Instrument Audio.," 2009, pp. 399–404.
- [2] McKay, Cory & Fujinaga, Ichiro. (2019). Automatic music classification and the importance of instrument identification.
- [3] Hamel, Philippe & Wood, Sean & Eck, Douglas. (2009). Automatic Identification of Instrument Classes in Polyphonic and Poly-Instrument Audio.. 399-404.
- [4] L. Guignard and G. Kehoe. (2015). "Learning Instrument Identification.," 2015.
- [5] Chollet, Francois and others. "Keras," 2015.
- [6] Prahallad, Kishore. "Spectrogram, Cepstrum and Mel-Frequency Analysis." http://www.speech.cs.cmu.edu/15-492/slides/03_mfcc.pdf