Capstone Project Proposal
Niko Laskaris
August 7, 2018

## DOMAIN BACKGROUND

Audio synthesis (generative modeling) and classification is important for a wide range of applications including text-to-speech (TTS) systems, music recognition, identification and generation, to name a few. There has been considerable work done in applying machine learning algorithms to classify audio signals. Some such projects include identifying audio sources from raw environmental audio (Li et al., 2017), to models that classify the genre, mood or artist from a sample of recorded audio (Lee et al., 2009). Many learning algorithms have been applied to the problem of audio classification, including neural network variants (DNNs, RNNs, CNNs), SVMs, and more.

As of 2017, a new dataset from the Magenta Project at Google AI, Nsynth, provides researchers with a larger and higher-quality dataset of discrete musical instrument audio signals than previously existed. With Nsynth, supervised learning and deep learning techniques can be used to predict various classifiers from the audio data in new ways. The original paper introducing Nsynth can be found here, and the dataset can be accessed here.

## PROBLEM STATEMENT

The primary goal of this project is to classify monophonic audio samples into instrument families. Classifying the instrument source of an audio sample is a problem relevant to music recommendation engines, music tagging, and more music-related fields. The Nsynth dataset includes labeling for Instrument Families and Instrument Sources, and thus supervised learning algorithms, such as SVMs, and deep learning algorithms like neural networks can be used to build a model that predicts the instrument from which an audio sample is generated. The accuracy of this model can be reported as a percentage of successful labelings on the test dataset.

## DATASETS AND INPUTS

While breakthroughs in synthesis and classification of images have been predicated on high-quality and large-scale datasets such as MNIST and CIFAR, until recently such breakthroughs have been more challenging with audio due to the lack of comparably large and high-quality datasets. The release of the Nsynth dataset through the Magenta project

at Google AI has enabled broader research into synthesis and classification for audio inputs. Nsynth consists of 306, 043 four-second annotated individual notes, each with a unique timbre, pitch and envelope. Each note is annotated with three additional features: *source* (acoustic, electric, or synthetic), *family* and *qualities*. For 1,006 instruments from commercial sample libraries, there are four second, monophonic 16kHz audio snippets, referred to as notes, created by ranging over every pitch of a standard MIDI piano (21-108) as well as five different velocities (25, 50, 75, 100, 127). The notes are held for the first three seconds and allowed to decay for the final second. As the sampling frequencies are the same for all samples we will not have to worry about transforms to accommodate for a wide frequency range across the dataset.

| Index | ID |
|---|---|
| 0 | bass |
| 1 | brass |
| 2 | flute |
| 3 | guitar |
| 4 | keyboard |
| 5 | mallet |
| 6 | organ |
| 7 | reed |
| 8 | string |
| 9 | synth_lead |
| 10 | vocal |

| Family | Acoustic | Electronic | Synthetic | Total |
|---|---|---|---|---|
| Bass | 200 | 8,387 | 60,368 | 68,955 |
| Brass | 13,760 | 70 | 0 | 13,830 |
| Flute | 6,572 | 35 | 2,816 | 9,423 |
| Guitar | 13,343 | 16,805 | 5,275 | 35,423 |
| Keyboard | 8,508 | 42,645 | 3,838 | 54,991 |
| Mallet | 27,722 | 5,581 | 1,763 | 35,066 |
| Organ | 176 | 36,401 | 0 | 36,577 |
| Reed | 14,262 | 76 | 528 | 14,866 |
| String | 20,510 | 84 | 0 | 20,594 |
| Synth Lead | 0 | 0 | 5,501 | 5,501 |
| Vocal | 3,925 | 140 | 6,688 | 10,753 |
| **Total** | 108,978 | 110,224 | 86,777 | 305,979 |

Some instruments are not capable of producing all 88 pitches in this range, resulting in an average of 65.4 pitches per instrument. Furthermore, the commercial sample packs occasionally contain duplicate sounds across multiple velocities, leaving an average of 4.75 unique velocities per pitch.

The instrument classes are denoted by the feature 'Instrument Families' as is shown above. As is clear, not all of the classes are balanced. Based on some personal research (example post), I will try unbalanced training first, use precision/recall metrics to see how the model is performing across the classes, and adapt my model accordingly. One possibility for handling imbalanced classes is to use a log-loss accuracy function, described further below.

The dataset is pre-split into training (289,205 samples), validation (12,678 samples) and test (4,096) samples. Instruments in the training set do not overlap with the validation or test sets and vice-versa. I presume this means that the model will be introduced to entirely new instruments in the validation and test sets.

NSynth was inspired by image recognition datasets that have been core to recent progress in deep learning. Similar to how many image datasets focus on a single object per example, NSynth samples hone in on a single note. I suspect this will limit the scope of an instrument identification project as any machine learning model trained on Nsynth exclusively will not be calibrated to expect multi-note (i.e., snippets of song) input.

## SOLUTION STATEMENT

I propose building a supervised learning model to train on the Nsynth dataset to predict for the instrument and/or instrument family feature that already exists in the dataset. I will start by performing feature extraction using the discrete fourier transform (DFT)[1]. Then I will conduct data exploration, using methods including but not limited to K-means clustering and Principal Component Analysis. I will measure the success of the PCA analysis, for example, by the amount of variance I can account for using a fixed number of PCAs. I will also use LibROSA, a python package for music and audio analysis to perform further data exploration and feature generation. Then, I will explore applying a range of machine learning algorithms to the dataset to classify audio samples by instrument family. I anticipate fitting SVMs, with a range of kernels, as well as attempting to build a CNN.

## BENCHMARK MODEL

As a benchmark model I will train and test an out-of-the-box Random Forest model on the Nsynth data. The goal will be for my model to outperform this baseline RF.

[1] http://cs229.stanford.edu/proj2015/010_report.pdf

Two models built by students at Stanford (which can be found here and here), both used a combination of PCA analysis and SVMs to classify audio samples by instrument source, achieving classification accuracy of 93% and 95% respectively. It must be noted that each of these models were produced before the Nsynth dataset was produced. Results may therefore differ as the data sources are not the same. I will potentially use these as secondary benchmark models.

**EVALUATION METRICS**

As our data is labeled we can use a straightforward metric to measure the accuracy of our model. A consideration will be the imbalance in classes for determining our accuracy function. I suspect that log loss will be a good candidate for this problem). I will attempt modelling the loss with simple L1 and L2 loss functions as well. Prior to training our model we will likely use PCA analysis to deploy dimensionality reduction to our data, a process which itself will be measured by the amount of variance the reduced PCAs account for.

**PROJECT DESIGN**

Phase 1: Data Exploration and Analysis

- Explore the Nsynth dataset.
- Apply k-means clustering, LibROSA and PCA to attempt to derive intuitive divisions in the data and generate features.

Phase 2: Model Selection and Training

- Create a range of SVMs, Neural Nets, XGBoost, etc.
- Train models to identify best candidates
- Further train best candidates to achieve maximum classification accuracy.

First the Nsynth dataset will be downloaded. I will spend some time exploring the samples, using both statistical and visual tools. As mentioned above several times, I suspect some combination of k-means clustering, PCA and the various tools in the LibROSA library (Mel-scaled power spectograms, Mel-frequency cepstral coefficients, chromagrams, to name a few) will be required to perform feature generation and identify which features to proceed with to model training.

Once I have performed the data exploration and feature generation I will start designing several classification models. A few models I will experiment with:

- SVMs
- CNNs

- XGBoost
- LightGBM
- More Ensemble Methods

I will explore these models with a range of hyperparameters to tune them appropriately. I suspect there will be an iterative process of massaging the results of training these models with the data in tandem with exploring loss functions.