Semi-Supervised RNN-GAN for Audio Chord Estimation

Charles Zhijie Chen, Elizabeth Chu, and Corey McNeish
{zcchen, elizachu, cmcneish}@stanford.edu CS230 - Spring 2018

Background

Audio chord recognition is the classification of a snippet of single-channel audio containing notes played simultaneously. For example, if an audio recording contains frequencies at 523Hz, 659Hz, and 784Hz for a given duration, the audio recording is said to contain a C major chord during that time. This problem is challenging for a number of reasons. A significant factor is the presence of harmonics, produced by every instrument, which can mask the dominant frequencies. Another is the quantity of notes played at once in most music, commonly known as polyphony.

We implement a sequence GAN model with bidirectional LSTM-RNNs to categorize music snippets into chords. This works well as it is able to capture not only the music at a particular moment, but the context before and after the moment in question.

Dataset and Preprocessing

The McGill Billboard Project has annotations for 890 unique songs. These songs were chosen from music used in presentations at ISMIR 2011 and music used in the evaluation of chord estimation algorithms for MIREX 2012. McGill distributes chroma vector (chromagram) feature sets of their music database, as their actual dataset is protected under copyright [1]. We train our network to consume and produce these chromagrams for use in chord estimation, as these chromagrams are easily generated—either from audio or at random—for data augmentation purposes.

The most relevant parameters of the dataset are summarized in Table 1:

Song Count	890	**Sampling Rate** (Hz)	44,100	**Frame Size** (ms)	372
Average Length (s)	209	**Slicing Window**	Hanning	**Frame Increment** (ms)	46

Table 1: Dataset size and sampling parameters of input songs.

A chromagram is a binned spectrum of 12 treble and 12 bass semitones. Specifically, the soundtrack in time domain is normalized with its running mean and running standard deviation. (This is also known as spectral whitening.) Then, the waveform is transformed to a binned spectrum, by aggregating frequencies around each semitone. Finally, to obtain the chromagram, this semitone spectrum is multiplied with the desired profile (treble or bass) and then mapped to the corresponding 12 bins [2].

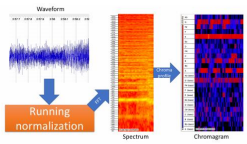

Figure 1: From waveform to chromagram.

Model

We use GAN for this project. Beyond the original setting of GAN, we adopt semi-supervised learning as described in [3], using the discriminator as a chord classifier for a predefined, constant K number of chord classes, and a K+1 class to predict whether the input data was generated or real. We performed training on the discriminator, and fork the generator's core module from Google's open-source Magenta project. Magenta uses WaveNet, and a combination of attention RNN, lookback RNN, and basic RNN to generate music.

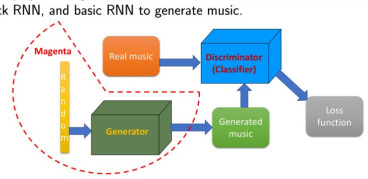

Figure 2: GAN model architecture. Magenta (red dotted lines) was forked from open-source code.

Discriminator: Bidirectional LSTM-RNN

A chord is recognized with information both locally and remotely. We use bi-directional LSTM structure for this purpose, as shown in Figure 3.

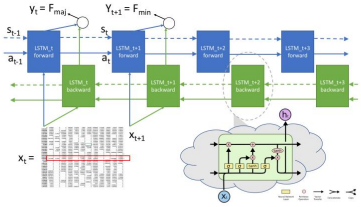

Figure 3: Bi-directional LSTM RNN structure used for discriminator.

The loss function of discriminator we use is directly adopted from [3]:

$$L_{\text{unsupervised}} = -\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\log[1 - p_{\text{model}}(y = K+1|\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim G}\log[p_{\text{model}}(y = K+1|\boldsymbol{x})] \quad (1a)$$

$$L_{\text{supervised}} = -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x})}\log p_{\text{model}}(y|\boldsymbol{x}, y < K+1) \quad (1b)$$

$$L_D = L_{\text{supervised}} + L_{\text{unsupervised}} \quad (1c)$$

Generator: Pipeline Protocol

While we use Magenta for the generator, the output of Magenta is raw audio music. We implemented a data preprocessing pipeline to convert the music into chromagrams, which is consistent with [1]. The pipeline process is:

1. Generate .MIDI files with Magenta [4].
2. Convert .MIDI files to .wav files with Fluidsynth [5];
3. Convert to chromagrams with SonicAnnotator-vamp:NNLS-Chroma [2].

Results

We trained on 89 labeled songs with an initial learning rate of 0.005, using the AdamOptimizer to minimize binary cross entropy. The result loss values are shown above. We have not reached convergence, and therefore the corresponding overlap ratio is 5%, compared to 80% in RNN implementations in recent results [6]. Note that the expected overlap ratio of random uniform classification for 25 chords is less than 4%, because the prior input distribution is non-uniform (there are biases towards certain more popular chords). Thus, a performance of 5% indicates that our network does learn to predict chords.

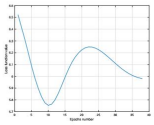

Figure 4: Training loss of the first 39 epochs. The training time of one epoch is exceedingly long due to large song sequences.

Discussion and Future Works

We have provided a proof of concept that Bidirectional LSTM-RNN could be used for the audio chord estimation task. The network trains in a semi-supervised framework on both labelled and unlabelled (generated) data, and the discriminator learns to classify the 25 chord labels in addition to telling the difference between true and fake data. The fact that loss value goes down, and overlap ratio is above random classification, indicates that the framework works and could perform better given more training, regularization, and hyper-parameter search. Going forward, we would like to explore more discriminator model architectures such as ones including GRU cells, attention models, and beam search. Additionally, training with shorter snippets of audio—rather than whole songs—could improve both our training speed and the distribution of chords we train.

[1] Burgoyne, John, et al., An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis, in Proceedings of the 12th International Society for Music Information Retrieval Conference, ed. Anssi Klapuri and Colby Leider (Miami, FL, 2011), pp. 63338.

[2] http://www.isophonics.net/nnls-chroma, 2018-06-09.

[3] Salimans, Tim, et al. "Improved techniques for training gans." Advances in Neural Information Processing Systems. 2016.

[4] https://github.com/tensorflow/magenta, 2018-06-09.

[5] https://github.com/FluidSynth/fluidsynth, 2018-06-09.

[6] Boulanger-Lewandowski, Nicolas, et al. "Audio Chord Recognition with Recurrent Neural Networks." ISMIR. 2013.