Auto Chord Detection

Bozhang Chen, Jiazheng Wang, Xixiang Liu

University of Illinois at Urbana-Champaign

Introduction

Goal

Our mission has two parts: firstly, to create a classifier that can identify a single guitar chord from a recording. Secondly, to develop a model that can detect the sequence of chords in a complete guitar piece.

1. Background Knowledge

Musical Background

1. A musical note represents an acoustic output generated by a musical instrument, distinguished by a specific fundamental frequency. Within the framework of Western musical theory, the chromatic scale encompasses twelve distinct auditory pitches, each denoted by alphabetical designations: A, A sharp, B, C, C sharp, D, D sharp, E, F, F sharp, G, and G sharp.
2. In music, harmonics are overtones accompanying a fundamental pitch, creating complex tones. For example, the A at 440 Hz has harmonics at 880 Hz, 1320 Hz, 1760 Hz, and so on. Harmonics plays a crucial role in defining the unique sound characteristics of different musical instruments.
3. In music, a chord is a harmonic set of pitches consisting of multiple notes that are played or sung simultaneously. A chord is characterized by its notes; for example, an A major chord contains the notes A, C sharp, and E. Chords are foundational elements in much of Western music.

Constant-Q transform

Similar to STFT, it transforms a data series $x[n]$ into a time/frequency graph (spectrogram).

- In STFT, for a frequency index k and a time sample m, the spectrogram is calculated as

$$X[k,m] = \sum_{n=0}^{N-1} W[n-m] \cdot x[n] \cdot e^{-j2\pi kn/N}$$

  where W is the windowing function and N is the length of each window.

- In Constant-Q transform (CQT), we see each frequency k as a filter f_k which is logarithmically spaced in frequency, and the spectrogram is calculated as

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k,n] \cdot x[n] \cdot e^{\frac{-j2\pi Qn}{N[k]}}$$

  where 1. Q is the quality factor which is defined as the ratio of the center frequency of the filter and the bandwidth δf_k 2. $N[k]$ is the window length for k-th frequency

$$\delta f_k = (2^{1/B})^k \cdot \delta f_{min}$$

- For CQT, it has smaller window length for higher frequency and larger window length for lower frequency.
- A series of experiments show that CQT achieves better results than STFT in music and speech analysis.

Long Short-Term Memory (LSTM)

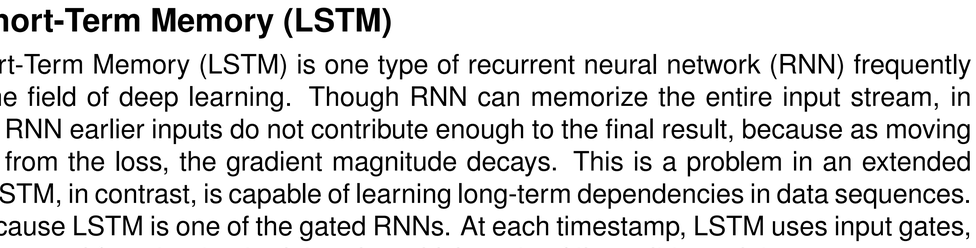
Long Short-Term Memory (LSTM) is one type of recurrent neural network (RNN) frequently seen in the field of deep learning. Though RNN can memorize the entire input stream, in traditional RNN earlier inputs do not contribute enough to the final result, because as moving backward from the loss, the gradient magnitude decays. This is a problem in an extended context. LSTM, in contrast, is capable of learning long-term dependencies in data sequences. This is because LSTM is one of the gated RNNs. At each timestamp, LSTM uses input gates, output gates, and forget gates to determine which parts of the unit to update.

2. Our Procedures

2.1 Stage One

2.1.1 Data Preparation

1. The GuitarSet data contains guitar recordings using different pickups and microphones, as well as chord notations with time stamps. For stage one, we use mono-pickup because of its smaller size and cleaner quality (recordings using a microphone will contain richer harmonics).
2. We only use the accompaniment tracks since solo tracks contain little harmonic information musically.
3. For each guitar piece we cut it into single chords and perform constant-Q transformation.
4. Then we perform the pitch class (or chroma feature) profile to the spectrogram after CQT. A pitch class is defined as the set of all pitches that share the same chroma. In music, that is to group a pitch in different octaves into one octave since the harmonic information has nothing to do with octaves.



(a) CQT (b) Chroma CQT

Figure 1: *Comparison between ONLY CQT and Chroma CQT*

5. Finally we zero-pad them into the same length. We decided to pad them into 100, which has the best precision.

2.1.2 Experiment Setup

- Classification Model Used:
  - Kernel SVM Classifier
  - Multilayer Perceptron
  - Gaussian Naive Bayes
- Random seed as 2048
- Training set is 80% and testing set is 20%

2.1.3 Prediction Result

- **For only predicting chord's root:**
  - Kernel SVM Classifier: Accuracy is 87.73%
  - Multilayer Perceptron: Accuracy is 88.66%
  - Gaussian Naive Bayes: Accuracy is 79.86%
- **For complete chord:**
  - Kernel SVM Classifier: Accuracy is 78.70%
  - Multilayer Perceptron: Accuracy is 81.94%
  - Gaussian Naive Bayes: Accuracy is 38.65%

2.1.4 Comparison between CQT and STFT

Under same experiment setup:

- **For only predicting chord's root:**
  - Kernel SVM Classifier: Accuracy is 80.09%
  - Multilayer Perceptron: Accuracy is 80.79%
  - Gaussian Naive Bayes: Accuracy is 72.45%
- **For complete chord:**
  - Kernel SVM Classifier: Accuracy is 67.13%
  - Multilayer Perceptron: Accuracy is 75.23%
  - Gaussian Naive Bayes: Accuracy is 38.43%

Conclusion: CQT-Chroma has better prediction than STFT-Chroma

2.2 Stage Two

2.2.1 A Try of LSTM

Music is a typical kind of extended context. LSTM is robust to varying sequence lengths, and useful for capturing temporal relationships in the sequences, so it may be a good choice.

Before Training Enumerate the annotation files to get the chord labels, then the .jam files to get labels, starts, durations, and thus the ends. Use the pairs of (start, end) to segment the corresponding .wav files with librosa. For each segment, calculate and pad the CQT feature, since its length varies. The input to the model is CQT features, and the output is predicted chord labels, converted into numeric class index.

The Model Two stacked LSTM layers and one linear layer are used for the following reasons:

1. Enhanced Feature Learning: The basic patterns are learnt by the first layer, and the second layer can capture more intrinsic dependencies.
2. Balanced Depth: while a deeper neural network may be more powerful, it may probably overfit and have a too high computational complexity. Two LSTM layers is usually a balanced choice.
3. Decision Making Linear Layer: It is common to use a linear layer to make decision at last. It can usually make a good final prediction combining the insights from previous non-linear layers.

Use the Model For a given .wav guitar music test file, librosa.onset.onset_detect is used to get the list of possible chord changing point, and librosa.frames_to_time to get the corresponding chord changing time. Again, we apply the method before used before training to compute CQT features and feed them to the model to compute the labels. Finally, a list of chords in the file and their corresponding start and end time are retrieved.

Performance The result is not good. The predicted chords change too fast. This might be due to the inaccuracy of librosa.onset.onset_detect.

```
start: 0.46439909297052157 end: 0.5108390022675737 chord:G#:min
start: 0.5108390022675737 end: 0.9287981859410431 chord:B:maj
start: 0.9287981859410431 end: 1.0448979591836736 chord:G#:min
start: 1.0448979591836736 end: 1.230657596371882 chord:F#:maj
start: 1.230657596371882 end: 1.4164172335600906 chord:C:min
start: 1.4164172335600906 end: 1.4628571428571429 chord:C#:maj
start: 1.4628571428571429 end: 1.6950566893424037 chord:D#:min
```

Figure 2: *Part of the LSTM Result*

3. Future Directions

1. Our stage two (using complete tracks to determine included chords) is not finished now, then we will use some different models to finish the task. We are going to use a similar approach to stage 1, where we first convert the entire audio to a cqt-chromagram and then try to separate the entire chromagram into similar sub-chromagram by clustering, and then classify each of the sub-chromagram.
2. We may try other datasets that include more instruments, for example, the beatles collection.
3. Try other approaches to complete stage two, for example, HMM and transformer.