Chord Progression and Note Sequence Generation – A Text-based LSTM Approach

Xingxing Yang[*]
Department of Music
Stanford University
xingxing@stanford.edu

Abstract

This paper introduces my work in using deep learning to generate chord progressions and note sequences. The model I used is generally based on LSTM(Long short-term memory). As a result, the model can generate the chord and note sequences successfully. The system I proposed can be useful in computer-aided composition for humans.

1 Introduction

AI composition has long been a research topic in music technology field. To some extent, music composition is quite similar to creative writing. Thus, the concept of text generation in deep learning can also be applied to music composition in some ways. So, is it possible for a computer to generate music using the same strategy of generating text in deep learning? The answer is yes.

Inspired by a paper that was able to generate chord progressions and drum tracks, I decided to apply the same strategy to (chord progressions and) note sequence generation for jazz improvisation.

In my project, for the chord progression part, my input is a preprocessed text file of chord progressions. I then use a LSTM sequence model to output a new chord progression. For the note sequence part, my input is a preprocessed text of midi sequence. I then use the same LSTM sequence model to output a new midi sequence.

2 Related work

One of the papers on deep learning-generated music, written by Robert and David[11], uses probabilistic grammars to generate jazz melodies. They analyzed the grammar of the melodies and harmonies, and then uses the probabilistic method to do the prediction and finally generate the improvisation. They didn't mention in the paper that they used the deep learning library.

Another paper, written by Nipun, Yuki and Axel[12], utilizes 4 different models to do the automatic music composition, which are Continuous Bag-Of-Words, character RNN, Sequence-to-Sequence and Generative Adversarial Networks. Their input is preprocessed music encoding data converted from ABC notation format. They generated the best results with seq-to-seq model, followed by character RNN.

Moritz and Novin, in their paper[13] uses a similar character RNN. But they use a different encoding method to extract midi information into a text file. They achieved a quite good result.

[*] Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Keunwoo, George and Mark[1] have done the similar work, too. But instead of generating note sequences, they use the character RNN to generate the chord progression and drum tracks.

In Allen and Raymond's paper[3], They proposed an end-to-end learning using a multi-layer LSTM, character-level language model and generation with deep neural nets to generate music. Their data can be mainly divided into 2 categories: midi files with minimal preprocessing and a "piano-roll" representation of midi files.

My project is generally based on Keunwoo, George and Mark's work. And then from generating chord progression, I see that it can also be applied to generate note sequences very similarly.

3 Dataset and Features

For the chord progression part, 2487 original jazz scores were used in the dataset. They were converted into .xlab format, and then transposed to the key of C. Then the chord features were extracted into a single text file. (The data was preprocessed by Keunwoo, George and Mark.)

A snippet of the chord text file: C:maj C:maj C:maj C:maj C:maj C:maj C:maj C:maj G:9 G:9 G:9 G:9 G:9 G:9 G:9 G:9 C:9 C:9 C:9 C:9 C:7 C:7 C:7 C:7 F:maj F:maj F:maj F:maj F:min7 F:min7 F:min7 F:min7 C:maj C:maj C:maj C:maj G:min7 G:min7 G:min7 G:min7 A:7 A:7 A:7 A:7 D:9 D:9 D:9 D:9 G:9 G:9 G:9 G:9 E:min7 E:min7 G:7 G:7 C:maj C:maj C:maj C:maj C:maj C:maj C:maj C:maj A:min7 A:min7 A:min7 A:min7 A:min7 A:min7 A:min7 A:min7

For the note sequence part, I currently only used one single MIDI file, which is 'A Sleepin' Bee.mid'(a jazz piano piece). It was converted into a text file, which contains its noteOn, noteOff, control change, resolution and tempo info.

A snippet of the MIDI text file

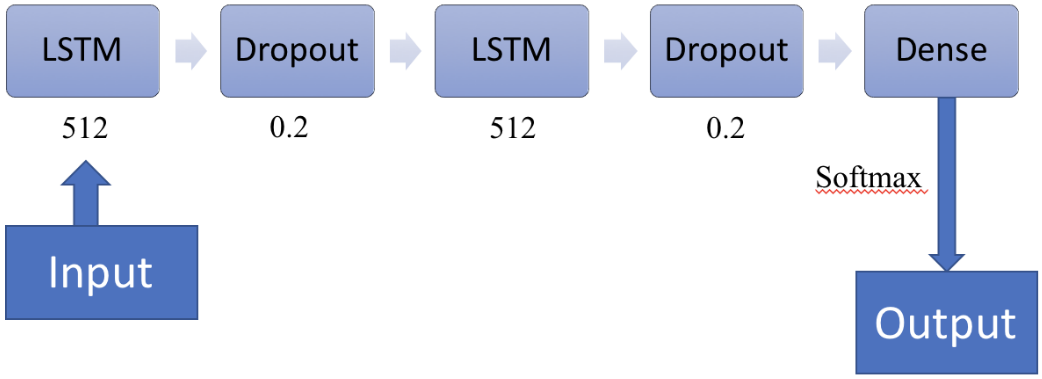
rs_960 0_st_94_631579 0_cc_0_64_127 0_cc_0_64_0 0_cc_0_64_127 0_cc_0_64_0 0_cc_0_64_127 0_cc_0_64_0 12_no_39_44 119_cc_0_64_127 486_no_51_57 8_no_60_31 5_no_46_37 13_no_56_59 48_no_60_0 12_no_51_0 39_no_56_0 203_no_46_0 32_no_54_80 4_cc_0_64_0 1_no_49_75 6_no_63_72 23_no_59_46 2_no_60_49 56_no_60_0 113_cc_0_64_127 311_no_59_0 17_no_54_0 40_no_63_0 115_no_49_0 216_cc_0_64_0 0_no_65_92 4_no_61_92 1_no_56_98 21_no_51_83

4 Methods

For the chord progression part, (1) First, read the chord text file, then make it a list. There are mainly 2 derived important lists – sentences, which contains a list of several chords and next_chars, which contains the chord after the several chords in the original file. (2) Build the model. Here I use a single LSTM Sequential model. (3) Train the model. (4) Generate the chords and write them into a file.

For the note sequence part, it is quite similar, except that the input and output are different. And for the generated textfile, which is quite similar to figure 2, there's a post-processing step for the text. It's converting the text file into midi file.

My model can be seen as below:



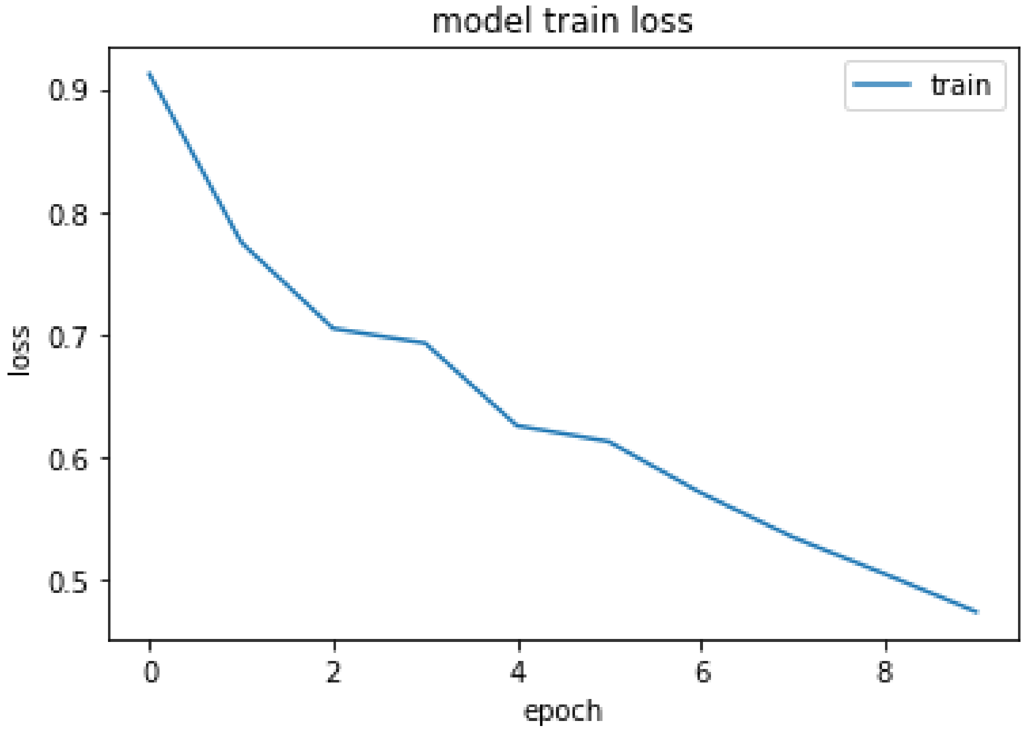
It is a simple Sequential model with 2 LSTM layers and 2 Dropout layers.

5 Experiments/Results/Discussion

After 10 epoches:

A snippet of the generated chords:

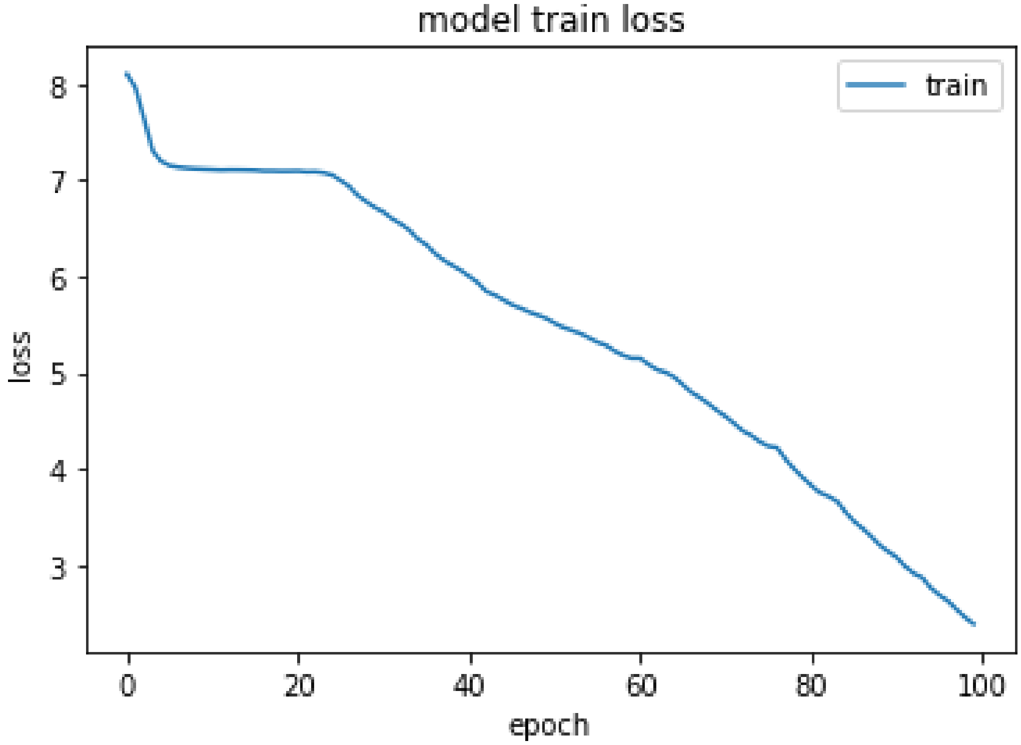
E:7 E:7 E:7 E:7 A:min7 A:min7 A:min7 A:min7 D#:7 D#:7 D#:7 D#:7 D:min7 D:min7 D:min7 D:min7 G:7 G:7 G:7 G:7 C:maj C:maj C:maj C:maj G:min7 G:min7 C:7 C:7 F:7 F:7 F:7 F:7 A#:7 A#:7 A#:7 A#:7 E:min E:min E:min7 E:min7 A:7(s9,s11,b13) A:7(s9,s11,b13) A:7(s9,s11,b13) A:7(s9,s11,b13) D:min7 D:min7 D:min7 D:min7



After 100 epoches:

A snippet of the generated note sequence:

170_no_70_52 6_no_59_46 3_no_63_48 11_no_58_47 5_no_67_44 365_cc_0_64_127 53_no_67_0 29_no_59_0 45_no_63_0 4_no_58_0 11_cc_0_64_0 20_no_70_0 48_no_63_21 12_no_69_41 2_no_57_40 4_no_59_42 131_cc_0_64_127 235_no_59_0 14_no_63_0 33_cc_0_64_0 24_no_68_55 53_no_68_0 1_no_71_63 179_no_77_0 48_no_72_0 65_no_72_0 2_no_74_71 17_no_37_42 30_cc_0_64_127 14_no_73_0 92_no_70_0 74_no_82_0 34_no_65_64



The note sequence is later converted back to a midi file, which is included in the code repository.

The model can generate some valid chord progressions and note sequences. However, for music improvising, the generated note sequence, when viewing in a midi viewer, looks very messy in timing. It is probably because of the processing of the time(MIDI tick) data. Or if I choose another music encoding format such as abc notation or musedata, it can get better.

6 Conclusion/Future Work

Here I list possible future works of this project: 1. Try other models to see what I can generate. 2. Use more training data. 3. Try another music encoding format. 4. Try another way to hangle the MIDI data.

7 Contributions

Since it's a one-person project. This section can be neglected.

References

[1] Choi, Keunwoo, George Fazekas & Mark B. Sandler. (2016) Text-based LSTM networks for Automatic Music Composition. CoRR abs/1604.05358

[2] Hadjeres, Gaëtan & Pachet, Francois. (2016) DeepBach: a Steerable Model for Bach chorales generation.

[3] Huang, Allen & Wu, Raymond. (2016) Deep Learning for Music.

[4] Brunner, Gino & Wang, Yuyi & Wattenhofer, Roger & Wiesendanger, Jonas. (2017) JamBot: Music Theory Aware Chord Based Generation of Polyphonic Music with LSTMs.

[5] Yuki Inoue Nipun Agarwala & Alex Sly. (2017) Music composition using recurrent neural networks. Technical report

[6] Keras

[7] codecs

[8] python-midi

[9] $https://github.com/keunwoochoi/lstm_real_book$

[10] $https://github.com/tatsuyah/deep-improvisation$

[11] Keller, Robert & R Morrison, David. (2007) A grammatical approach to automatic improvisation. Proceedings of the 4th Sound and Music Computing Conference, SMC 2007.

[12] Yuki Inoue, Nipun Agarwala and Alex Sly. (2017) Music composition using recurrent neural networks. Technical report

[13] Moritz Hilscher, Novin Shahroudi. (2018) Music Generation from MIDI datasets.