CS230

Mozart 2.0: Milestone 2 Report (Music Generation)

Wen Zhong
wenzhong@stanford.edu

Abstract

We are interested in using deep learning models to generate new classical style music, especially Mozart style music. We build a LSTM-based architecture trained with tokenized MIDI files with Maestro Dataset. The performance of our model is measured by comparing the generated notes with the ground truth. While music generation is a relatively well-explored field of deep learning, we can still do new things like music style transfer.

1 Introduction

Despite the evolving of modern music, classical music still matters. Over the centuries, classical music has transformed itself to become a building block, setting the framework for musicians of all types today. Classical music has some general characteristics. It has short and clearly defined musical phrases with two or more contrasting themes. The rhythm is very defined and regular. Deep Learning has been introduced to various fields including those related to art. We have seen some interesting stories like style transfer. As a creative process, music composition can sometimes be very hard for composers lack of inspiration. We want to know whether Deep Learning can help them in composition. Because of the importance and characteristics of classical music, we think deep learning with classical music is a good starting point.

Our project is focused on music composed by Mozart because he is one of the most prolific composers and he has a piece left unfinished, Mozart's Requiem Mass in D minor, which can make the application of our model more interesting. Our project is mainly about building a LSTM model which can generate classical music in Mozart style. The input of the model is MIDI files.

2 Related work

Different models are studied for music generation. Since music is sequential, RNN models are largely applied. The MelodyRNN models [1] proposed by the Magenta Project from the Google Brain team are possibly among the most famous examples of symbolic-domain music generation by neural networks. In total, three RNN-based models were proposed, including two variants that aim to learn longer-term structures, the lookback RNN and the attention RNN. Song from PI [2] is a hierarchical RNN model that uses a hierarchy of recurrent layers to generate not only the melody but also the drums and chords, leading to a multitrack pop song. This model nicely demonstrates the ability of RNNs in generating multiple sequences simultaneously. However, it requires prior knowledge of the musical scale and some profiles of the melody to be generated, which is not needed in many other models. Some researchers work on music generation with GAN. MuseGAN [3] proposed three models for symbolic multi-track music generation under the framework of GAN. It also used bars instead of notes as the basic compositional unit. We also see some work about using

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

CNN in music generation. A WaveNet [4] model proposed by DeepMind shows that CNNs can also generate realistic musical waveforms in the audio domain. Other works mix RNN, GAN and CNN and their variants. MidiNet [5] used GAN together with CNN. It proposed a conditional mechanism to exploit available prior knowledge, so that the model can generate melodies either from scratch, by following a chord sequence, or by conditioning on the melody of previous bars (e.g. a priming melody), among other possibilities. Our project is different from past work as we want to generate music in only Mozart style. With limiting the style of generated music, we face some new challenges like overfitting.

3 Dataset and Features

We used MIDI files consisting of a single instrument piano throughout the project but any set of MIDI files consisting of a single instrument would work for our purposes. There are two datasets involved in this project. Final Fantasy soundtracks is used for pre-trained model. We picked Final Fantasy music due to the very distinct and beautiful melodies that the majority of the pieces have and the sheer amount of pieces that exist. In pre-trained model, 92 MIDI files are used. The dataset can be found ***here***. Since we use transfer learning in our project, besides pre-trained model, for the model for generating Mozart style music, the dataset used is ***Maestro dataset***.

We use Python package music21 for data processing. Below is an excerpt of Sonata in F Major, K. 280, 1st mov extracted by music21.

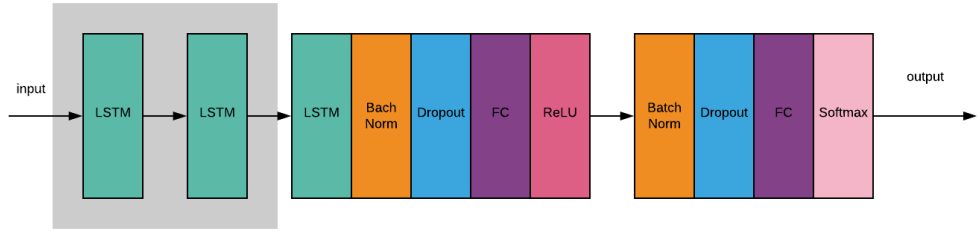
```
{0.0} <music21.stream.Part 0x7f4ba8823d30>
{0.0} <music21.instrument.Piano Piano>
{0.0} <music21.tempo.MetronomeMark animato Quarter=120.0>
{0.0} <music21.meter.TimeSignature 4/4>
{0.0} <music21.stream.Voice 0x7f4ba640b828>
{0.0} <music21.note.Rest rest>
{1.75} <music21.note.Note C>
{1.75} <music21.note.Rest rest>
{2.0} <music21.note.Note F>
{2.0} <music21.note.Note A>
{2.25} <music21.note.Note F>
{3.75} <music21.chord.Chord C5 A4>
...
```

As we can see from the excerpt, this MIDI contains a piece played by Piano. The time signature is 4/4. The number in curly bracket at the beginning of each line is the offset referring to where the note is located in the piece. We will only learn about notes and chords in this project despite we have some other information like rests and durations. Further work can be done on more music elements.

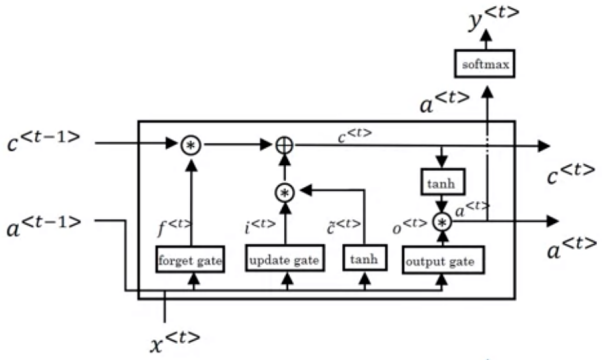
We encode string-based categorical data to integer-based numerical data using Bag-of-words model. To be more specific, assuming we have a corpus like {<music21.note.Note C>, <music21.note.Note D>, <music21.note.Note E>, <music21.note.Note F>, <music21.note.Note G>} and a piece is {<music21.note.Note D>, <music21.note.Note D>, <music21.note.Note G>}. Then with Bag-of-words model, the piece will be represented as [1, 1, 4]. For the corpus our model trained on, there are 413 distinct notes and chords for training data.

4 Methods

The neural network we used in this project can be represented as following graph:



The main building blocks are LSTM layers. LSTM (Long Short Term Memory networks) are a special kind of Recurrent Neural Networks, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) [6]. One LSTM layer is a chain of LSTM units. Each unit can be represented as:



In the graph, t means time. a is for hidden state. c is cell state. Concrete meanings of each symbols and more detailed explanation of LSTM can be found in this ***blog***.

For our model, we used 3 LSTM layers, each of which has 512 hidden units. The first two LSTM layers (in grey block) is pre-trained. We pre-trained the same model with Final Fantasy music dataset and extracted the wights of the first two layers. The remaining layers (starting from the third LSTM layer) were trained with Mozart music from Maestro dataset. Regularization techniques Batch Norm and Dropout are used in our model. The drop rate we used is 0.3. We have put the length of each sequence to be 100 notes/chords. This means that to predict the next note in the sequence the network has the previous 100 notes to help make the prediction. The parameters of each layer are summarized as below:

Layer (type)	**Output Shape**	**Param #**
lstm_1 (LSTM)	(None, 100, 512)	1052672
lstm_2 (LSTM)	(None, 100, 512)	2099200
lstm_4 (LSTM)	(None, 512)	2099200
batch_normalization_3 (Batch Normalization)	(None, 512)	2048
dropout_3 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 256)	131328
activation_3 (Activation)	(None, 256)	0
batch_normalization_4 (Batch Normalization)	(None, 256)	1024
dropout_4 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 413)	106141
activation_4 (Activation)	(None, 413)	0
Total params: 5,491,613	Trainable params: 2,338,205	None-trainable params: 3,153,408

How music generation works is given a note or chord, we pick the most likely note or chord from all possibilities. Therefore, we use softmax as last layer and cross entropy as loss function.

5 Experiments/Results/Discussion

5.1 Training

Our model training involves two parts, pre-training and training. It took about 36 hours to train the model once. For pre-training, we used 200941 samples tokenized from Final Fantasy soundtracks. For training, we used 12938 samples tokenized from 39 Mozart works. We split the dataset into train set and test set with the ratio 9:1.

5.2 Hyperparameter Tuning

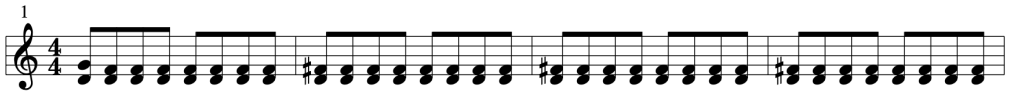
There are bunch of hyperparameters involved in the model like epochs, batch_size, learning_rate. We trained the model using a batch_size of 32, 64, 128 and epochs 50, 100, 200. Loss was reduced with more epochs but the decreasing speed was slowed with more iteration. Considered of loss, accuracy and efficiency, we found the best combination of batch_size and epochs is 128 and 200.

Batch Size	Epochs	Accuracy	Loss
32	50	0.0309	4.9858
32	100	0.0391	2.4029
32	200	0.0471	1.5167
64	50	0.0336	4.4647
64	100	0.0360	2.2219
64	200	0.0465	1.1829
128	50	0.0374	4.1604
128	100	0.0421	2.0119
128	200	0.0482	1.1711

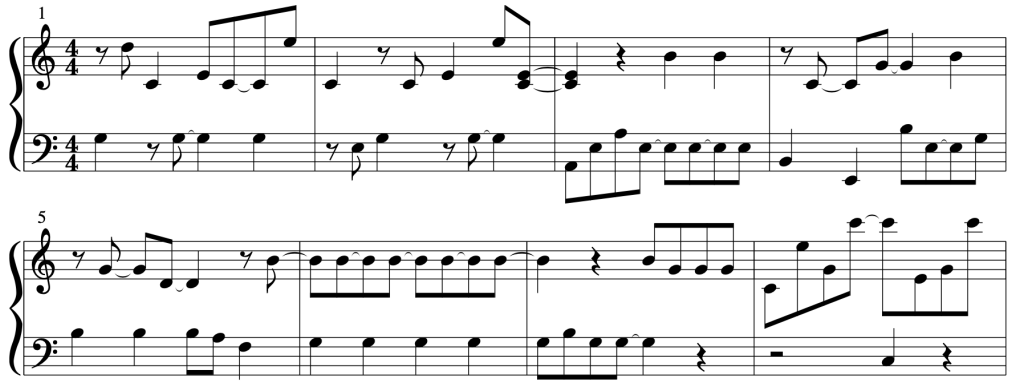
5.3 Error Analysis and Results

We observed the accuracy of the model is relative low. We think it can be caused by several reasons. On one hand, the distribution of the dataset used for pre-trained model is different from that used for transferred model. This issue can be mitigated by using other pieces from Maestro dataset for pre-training. On the other hand, since music piece is usually long, some information might not be captured after some period of time. We think we can try Attention to keep more information.

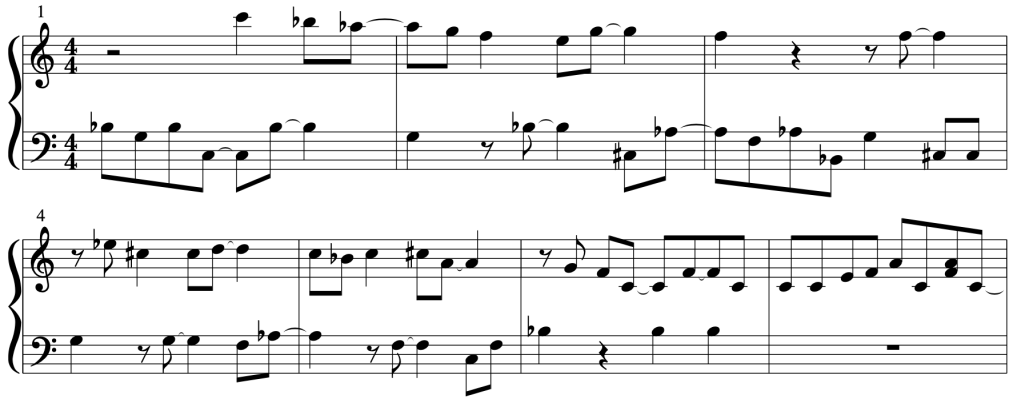
Below is the first 4 or 8 measures of generated piece at different epochs of our model.
Epoch 1:



Epoch 30:



Epoch 70:



Based on the results, it seems the model learns music by first learning some notes or chords, then patterns and finally the combination of different patterns. The result looks great. It contains various notes, chords, rests and different durations. Prev and next note pairs are reasonable - not a big span. The music sounds beautiful from our perspective.

6 Conclusion/Future Work

For this project, we used LSTM with transfer learning to generate decent and beautiful music in Mozart style. For future work, we recommend extending this technique to generate music in multiple instruments and other genres. We feel it would be interesting if we can compose pop with a taste of Mozart just like the example that we draw the Louvre Museum in the style of Claude Monet in the class.

7 Code

Code for this project is available at `https://github.com/wwwennn/Mozart2.0.git`.

8 Contributions

Wen Zhong worked independently on this project.

References

[1] Elliot Waite, Douglas Eck, Adam Roberts, and Dan Abolafia. Project Magenta: Generating longterm structure in songs and stories, 2016. https: //magenta.tensorflow.org/blog/2016/ 07/15/lookback-rnn-attention-rnn/.

[2] Hang Chu, Raquel Urtasun, and Sanja Fidler. Song from PI: A musically plausible network for pop music generation. arXiv preprint arXiv:1611.03477, 2016.

[3] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, Yi-Hsuan Yang, *MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment*

[4] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

[5] Li-Chia Yang, Szu-Yu Chou, Yi-Hsuan Yang. MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory