Composing Music with Deep Learning

Min Xiao
Analytics 2019
Georgetown university
mx61@georgetown.edu

Yimeng Xu
Analytics 2019
Georgetown university
yx154@georgetown.edu

Abstract

Musicians have composed music with precise structure. The goal of this project is to build a neural networks model to generate music comparable to human-created pieces. Previous work includes various methods such as Hidden Markov Models, recurrent neural networks. In our project, we apply RNNs with long short-term memory cells on the music note sequences to predict the subsequent music notes. With the trained model, we are able to generate music pieces. To polish the generated music piece, some post-processing techniques are used.

1 Introduction

1.1 Background

Algorithmic music has been developed with various attempts in recent years. Music, like any other language, possesses its own grammar and rules (h2g2, 2006). Music theory has developed a series of fundamentals, such as motives, phrases, sentence, movement forms. Musicians such as Bach are known for their precise underlying music structure. Since there are structures and patterns in music pieces and musicians integrate these structures with their creativity, it is possible to create music using algorithms.

There are many attempts in composing music using algorithms. Common methods include Markov models, generative grammars, genetic algorithms, neural networks. Recurrent neural networks(RNNs) have shown a great advantage in dealing with sequence data. They are widely used in text mining, language processing, time series analysis as well as in music composition. RNNs make use the current and historical information to predict the future, which is very effective to capture time dependency pattern.

1.2 Literature review

A wide range of research focuses on the music composition using algorithms. One of the earliest research (Chen and Miikkulainen, 2001) adopts a rudiment recurrent neural network (based on SRN) with music rule constraints as the fitness function for model training. The current music measure is the input for generating the next measure. Chen and Miikkulainen experimented on single phonic music without chords and disregarded dotted notes and rests. Chen and Miikkulainen's work doea not produce comparable results with more recent research but they made a breakthrough to apply neural networks on music modeling. RNN-based probabilistic models (Boulanger-Lewandowski et al., 2012) beat the popular models in music information retrieval and can serve as a symbolic prior for polyphonic transcription, which improves the state-of-art. Tied Paralel LSTM-NADE and and Bi-Axial LSTM (Johnson, 2017), which can be regarded as RNN-NADE architectures modified by tied parallel techniques, are able to capture the property of translation invariance of music. TP-LSTM-NADE and BALSTM predict music sequences with a lower log-likelihood loss and a higher accuracy than the RNN-based RNN-RBM(HF) on the same four datasets in N.Boulanger-Lewandowski et al.'s research. But they occasionally appear to subsequently output discordant notes after playing a wrong one. They successfully model measure-level music structure but are not able to maintain consistent over a long time. N. Boulanger-Lewandowski et al. and D. D. Johnson's work

combine neural networks with probabilistic modeling. Another approach to music modeling and composition is towards deep learning. Seq2Seq models are able to produce human-pleasing music with appropriate music structures while character RNNs cannot learn structures and GANs yield no satisfactory results due to instable training (Agarwala et al.,).

1.3 Paper Structure

In this paper, we will discuss why Long short-term memory cells are good at capture time dependency patterns and walk through the RNNs model structure. We will train the model and use the model to generate music sequence.

2 Midi Data

Music files are structured as a series of concurrent tracks, each containing a list of meta messages and messages. Here we use two datasets. one is the Bach corpus from MuseData, and another is a mixture of many musicians. Bach is known for his structured music composition, while other musicians pay less attention to structures. By choosing two different corpora, we are interested in the effects of underlying music structure on models.

MuseData comes from musedata.org which is an electronic library for classical orchestra and piano music from CCARH4.
Piano-midi.de is also a classical piano midi archive.

We converted the classical music pieces to snippets of $T_x = 30$ music values. The total number of sinipts is 69421 and 76187 respectively. One-hot vectors are used to represent 88 unique music notes. After preprocess, the music summary is as table below.

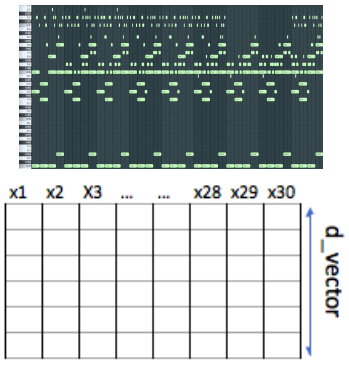


Fig1: Midi file and processed data

3 Model Description

LSTM

Long Short-Term Memory(LSTM) is able to capture the long-term temporal sequences. LSTM is designed in the 1990s to prevent the vanishing gradient problem. It can obtain constant error flow over long time periods by using Constant Error Carousels(CECs)(Johnson, 2017). By maintaining those constant errors, RNN can learn longer time steps so that they can connect the relationship over long time gaps.

The constant errors from LSTM are connected to a set of nonlinear units that can interface with the rest of the networks. LSTM contain information in the gated cells outside of the normal RNN structure. In the cells, input Gate determines how to change the memory cells; the output gate tells the information extracted by the memory cells; forget gate allow memory cells to filter the irrelevant information. The diagram below illustrates the LSTM cells and workflow.

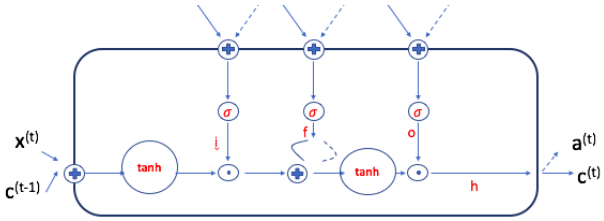


Fig1: Midi file and processed data

To capture the time dependency pattern from music sequence, we will use the LSTM as cells in the RNN structure.

Network Architecture:

We use LSTM cells to predict the next music notes for any given note series. For each sequence of 30 music notes, we use LSTM cells to loop over all music notes in the network to predict next music note.
The input x and responding sequences y are almost the same except that the responding sequence is one step ahead to the future.

$$y^t = x^{t+1}$$

For example, if we put music note $x^{(1)}$ and initialized state $a^{(0)}$ into the LSTM cell, we will $a^{(1)}$ and $c^{(1)}$ as the output. $\tilde{y}^{(1)}$ is the softmax result of $a^{(1)}$ and will be used to evaluate against $y^{(1)}$, i.e. $x^{(2)}$.

For the next step, the input music note will be $x^{(2)}$ and the input hidden state is $c^{(1)}$.
Here, We will use an LSTM with 64-dimensional hidden states and the model is unrolled in $T = 30$ time steps. The Recurrent neural network not only considers the current input music but also makes use of the past music sequence.

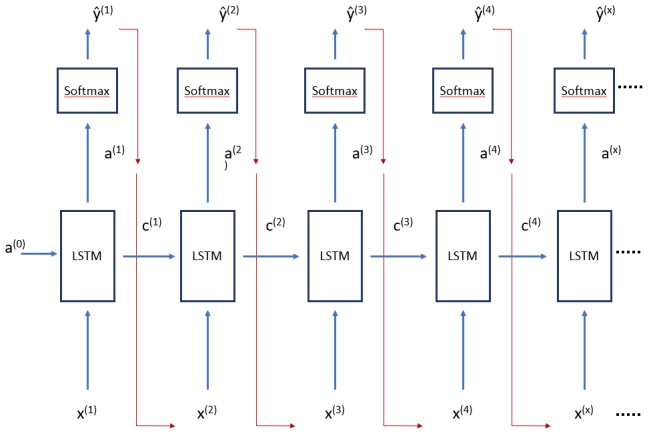


Fig3: Model Structure

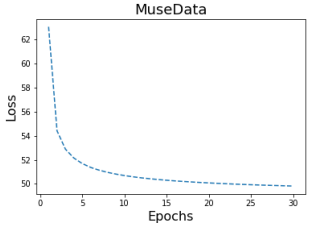
4 Experiment and Results

- **Model setting-up:**
  The model has a set of hyperparameters. Here is the table of all hyperparameters. For both the dataset, we set the number of hidden states in LSTM as 64, learning rate as 0.1. The loss function almost converges after 20 epochs, so stopping training after 30 epochs will be a wise choice.

Dataset	# Hidden States in LSTM	Learning Rate	Batch Size	Epochs
MuseData	64	0.1	128	30
Piano-midi.de	64	0.1	128	30

Table1: model parameters

- **Experiment result:**
  First, we train the model and apply the model. Here are the trajectories of loss function for both datasets.



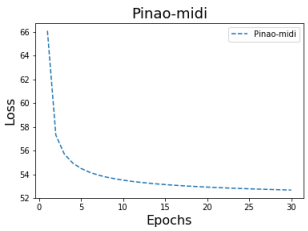
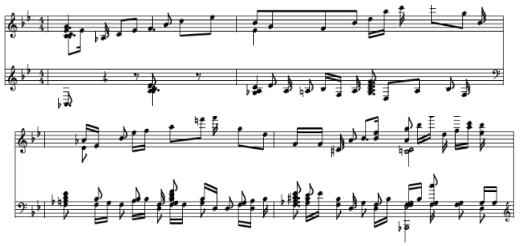


Fig4: training loss function

- **Generated Music Sample:**

  For the next step, we apply the model to generate music notes. Most computational music algorithms use some post-processing techniques. Post-processing clean generated music by making sure the same sound is not repeated too many times, that two successive notes are not too far from each other in pitch and so on. The raw generated music will be highly enhanced with the post-processing. In our generated music, we also have some post-processing.



- **Result Discussion:**

  – When training the RNN models, the loss decreased very fast at the earlier stage. After 20 epochs, it almost stabilized. To save the computational resources, we implement the model 30 epochs.

  – The model behaves fairly on both the dataset, with 60.96% on piano-midi data and 61.40% on the MuseData set. Without considering the underly music structure, the model performs very well. The main purpose of this project is to generate music sequences or create music once building the model instead of comparing with the existing samples. Thus, it is not necessary to focus on improving test accuracy. Once we get a decent model, we can use the model to generate music notes sequences.

– The generated music sequence is of high quality. We have volunteers to evaluate the music quality. Most of the volunteers can't tell the difference between pieces composed by musicians and computer.

5 Conclusion & Future Work

Conclusion

Neural networks are very flexible and can capture complicated patterns. With the deep models, we can create applications which are considered impossible before. In our model, the recurrent neural networks can capture the music patterns and are able to generate music of good quality. With around 60% accuracy with 88 classes, the model sequence reaches a comparable result. The generated music pieces sound very similar to human music.

Future Work

Although the model can generate fair results, we can improve LSTM can compose the music with fair quality. We can improve the project from the following aspects:

- The underlying music structure is very important for modeling. To improve the model performance, we should know more about the music composition architecture and integrate the principle of music structure into the model.
- Our current model just used the workflow with LSTM cells, which could be improved by using multiple models. For example, the usage of Recurrent neural networks combined with Restricted Boltzmann Machines has been a very effective method in composing music. There are plenty of models specializing in the music composition. Our next step can start by exploring various methods.

References

Structure in Classical Music. https://h2g2.com/edited_entry/A8379219

Chun-Chi J. Chen and Risto Miikkulainen, *Creating Melodies with Evolving Recurrent Neural Networks*, In Proceedings of the 2001 International Joint Conference on Neural Networks, (IJCNN-01, Washington, DC). IEEE, 2001.

Nicolas. Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent, *Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription* Proceedings of the 29th International Conference on Machine Learning, ICML 2012. 2.

Daniel D. Johnon, *Generating Polyphonic Music Using Tied Parallel Networks* EvoMUSART 2017. Lecture Notes in Computer Science, vol 10198. Springer, Cham

Nipun Agarwala, Yuki Inoue, and Axel Sly, *Music Composition using Recurrent Neural Networks*Stanford University, Technical Report in CS224 2017.

Deeplearning.Ai. *Sequence Models* Coursera.org 2018