Music style generation based on Bi-directional LSTM

Shang Zhen
218012058@link.cuhk.edu.cn

Yue Guan
218012038@link.cuhk.edu.cn

Yicong Wang
218012038@link.cuhk.edu.cn

Abstract

In this paper, we describe a model that can learn to perform sheet music. We start by reproducing the Bi-directional LSTM model to capture dynamic music styles structure, and explore various network architectures and training strategies. Our research presented model evaluation and used survey to conclude that the generated performances are distinguishable from a human performance, thereby passing a test in the spirit of a "musical Turing test".

1 Introduction

Music play an important role in our life[1]. Because the expression of music is mainly based on composition and performance[2], even we use the same music sheet to perform, it results in a variety of realizations and performance, not to mention music styles.

In this project, we focus on how to automatically synthesize musical performances that are indistinguishable from a human performance. Specifically, we postulate an important assumption that based on the same composition, music style is impacted by every notes velocity. The different speed of notes will influence pitches effect[3].

2 Related Work and Methodology

Our aim is to predict the note velocities from a sequence of notes. Traditional methods for capturing musical structure and properties are neither efficient nor work-well. Fortunately, the recent development in RNN[4] are designed to capture dynamic structures by retaining a "memory" of previous patterns[5]. In order to avoid vanishing gradients problem and build a sequence of music, our initial idea was to use LSTM to build our models[6].

Musical styles can be categorized by genre. We describe the architecture StyleNet, based on the previous work. StyleNet predicts the dynamics of a musical input such as sheet music. In this case, the similar feature shared between two inputs is the sheet music. The task at hand is to produce different outputs for the sheet music. The styleNet architecture has two main components: the interpretation layer and the GenreNet unit.

Interpretation Layer: The interpretation layer converts the musical input into its own representation of the sheet music. As this layer is shared, the number of parameters the network needs to learn are reduced. The input interpretation layer is set to be 176 nodes wide and only one layer deep.

GenreNet Unit: These subnetworks are attached to the interpretation layer. There are two GenreNet units: one for jazz style and one for classical style. Each GenreNet is three layers deep. The GenreNet units consists of two main layers: the bidirectional LSTM layers and the linear layer. The bidirectional architectural choice is based on the real task of reading sheet music. This would be analogous to using a bidirectional LSTM layer give us this foresight. Furthermore, we add linear layer to scale the

Paper Report of Final Project for CIE6032 (MDS6232)- Selected Topics in Deep Learning.

output to represent a larger range of values. A linear layer performs a linear transformation on its input.

In our model, we tried to evaluate four models such as bidirectional LSTM, bidirectional GRU, LSTM, and GRU. Showed as Figure 1.

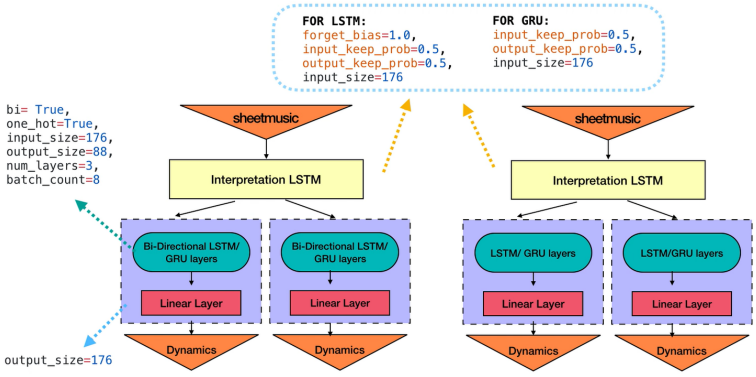

Figure 1: Four StyleNet Models

3 Experience

We present the Piano dataset. The dataset contains Piano MIDI files within the Classical and Jazz stylen music. All MIDI files are in 44 time.In our work, we colllect 349 MIDI classical files and 183 jazz files[7], which set to be 95% training set and 5% validation set respectively.

Input Representation: the model needs to know what notes are being played at a given time-step. A note can set to be three states: note is on, note is off, or note is sustained from the previous time-step by used a binary vector. Then, the note pitch needs to be one-hot encoded. a matrix with the first dimension representing MIDI pitch number is created. The second dimension represents a quantized time-step or a 116 note.

Output Representation: Similar to input matrix above, the columns of our matrix represent pitch and the rows represent time-step. The velocities of the notes are encoded into the matrix. The velocities are preprocessed, and use normalization so the network does not have learn the scale itself. This means all the velocities are between 0 and 1.Figure 2,3 shows the input and output visualization.

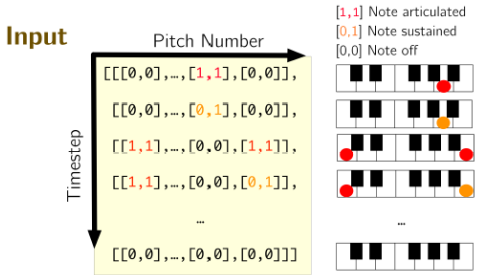

Figure 2: Input

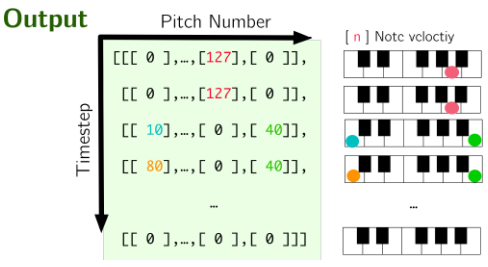

Figure 3: Output

In the training model, the length of each song is basically more than 1000-time steps. In order to get batch, padding is performed on each sequence first, and the dimensions of each batch are kept equal.Then padding increases the length of each sequence, which has a great influence on the training efficiency. In order to improve the training efficiency, we truncated to 200 time-steps to reduce training time. Figure 4 shows the treatment of squence truncated.

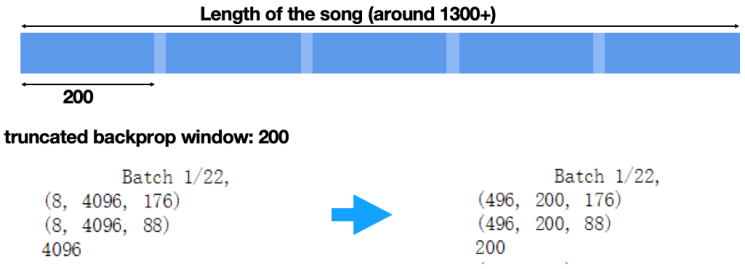

Figure 4: Truncated squence

4 Results

Figure 5, 6 shows the training curve and test curve of the graph. The training error of the Bidirectional LSTM & GRU model decreases rapidly, but after 200 epochs, the training error and test error achieved by the last four models are similar, about 0.0018 and 0.002 respectively.

Figure 7 shows the visualization of the velocity matrix of a classical song by using bidirectional LSTM model to verify after 200 epochs. In the figure, the color bar represents the actual velocity and the line represents the time step and the column represents the pitch, the second graph and the third graph are the music rates generated by the two different style networks respectively. The fourth graphs are the difference between the actual velocity and the third column to generate the jazz matrix. It can be seen that the velocity of the generated jazz style is different from the original classical velocity.

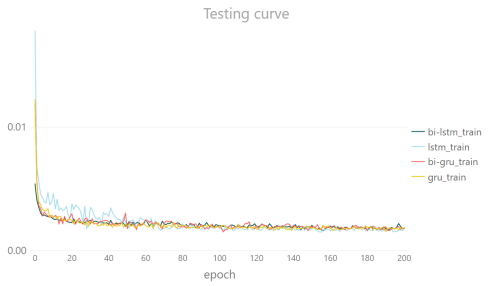

Figure 5: Training curve

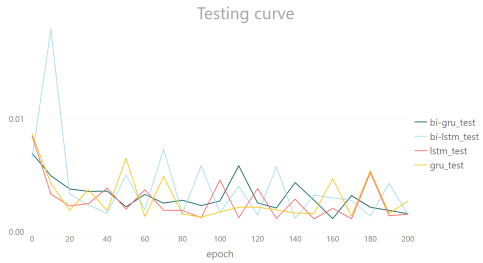

Figure 6: Testing curve

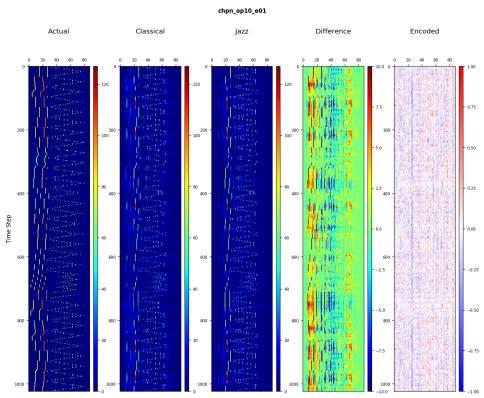

Figure 7: Velocity matrix

In response to the results of different styles of songs, we used questionnaire to measure the effect, setting up three questions to test whether a machine can perform sheet music like human[8] and

different musical performance based on different styles of input music. We finally collected 137 samples and got the above three results corresponding to three questions. Figure 8 shows the statistical results of the questionnaire.

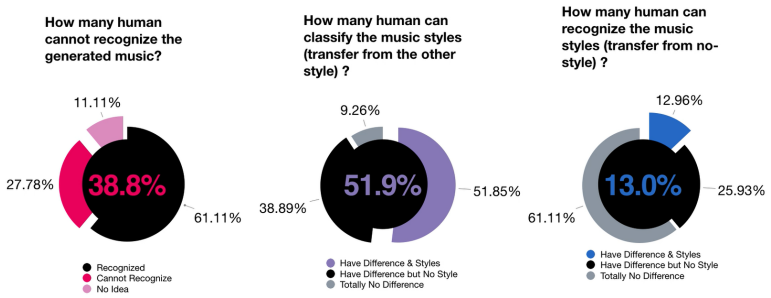

Figure 8: Statistical percentage

Identify the Human Test: The first question shows that 61.11% of people recognized that this is machine-generated music. There is no known benchmark for this problem. Thus, a baseline is a random guess. This reveals that on average, 4.74% from the participant pool could perform better than random guessing. It can conclude that the model passed the Turing test.

Identify the Style Test: The second question shows that 51.85% of people thought that the generated music had difference but no style. The third question shows that 61.11% of people cannot tell difference. Similar to previous test, the baseline of this test is randomly guessing between both answers. The analysis of this number shows that the structure of the Style model is not sufficient to separate the characteristics between the two styles. Also, the input music with labeled style have better performance for input music without labeled style to distinguish their difference.

5 Future Work

Although there are differences in the rate of music generated by different styles, it is also not so well to distinguish the difference from the original song. So, we tried to use the same data set and LSTM model to generate music melody[9]. Figure 9 shows the graph after 10 epoch's music generation and Figure 10 shows the graph after 200 epoch's music generation. Get the following results, but the effect is not so well. We expected that in the future work, we will use the recently popular model like CycleGan[10] or WaveNet[11] to process the music sequence.

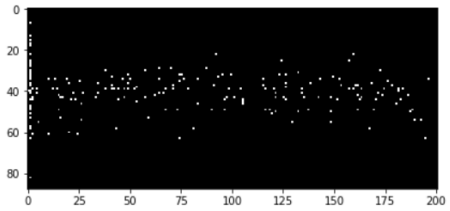

Figure 9: 10 epoch result

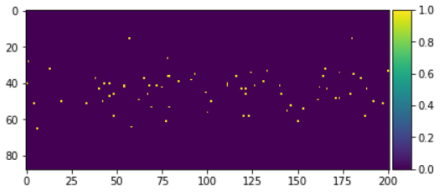

Figure 10: 200 epoch result

6 Conclusion

In this paper, we have presented a model that is capable of creating natural sounding performances which are indistinguishable from a human performance. Our style model is based on a LSTM network. We have a better understanding of the processing of music sequences, the padding processing and the processing of truncated subsequences in improving model efficiency. Also, we are more familiar with the LSTM and GRU models.

Reference

[1] Morley I R M. A multi-disciplinary approach to the origins of music: perspectives from anthropology, archaeology, cognition and behaviour[J]. Journal of Anthropological Sciences, 2014, 92: 147-177.

[2] De Mantaras R L, Arcos J L. AI and music: From composition to expressive performance[J]. AI magazine, 2002, 23(3): 43.

[3] Malik I, Ek C H. Neural translation of musical style[J]. arXiv preprint arXiv:1708.03535, 2017.

[4] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. CoRR, 2015.

[5] Ellis D P W, Poliner G E. Identifyingcover songs' with chroma features and dynamic programming beat tracking[C]//Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. IEEE, 2007, 4: IV-1429-IV-1432.

[6] Eck D, Schmidhuber J. Finding temporal structure in music: Blues improvisation with LSTM recurrent networks[C]//Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on. IEEE, 2002: 747-756.

[7] Yamaha International Piano-e-Competition. URL http://www.piano-e-competition.com/.

[8] M Alan. Turing. Computing machinery and intelligence. Mind, 59(236):433–460, 1950. ISSN 0026-4423. doi: http://dx.doi.org/10.1007/978-1-4020-6710-5_3.

[9] Boulanger-Lewandowski N, Bengio Y, Vincent P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription[J]. arXiv preprint arXiv:1206.6392, 2012.

[10] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[J]. arXiv preprint, 2017.

[11] Van Den Oord A, Dieleman S, Zen H, et al. WaveNet: A generative model for raw audio[C]//SSW. 2016: 125.