Violin Etude Composing based on LSTM Model

Xuanhe Zhang *

Department of Information Science, North China University of Technology, Beijing, China

* Corresponding author Email: 19101020111@mail.ncut.edu.cn

Abstract. After the development of deep learning and computer music, there is more and more electronic music composed by computers and algorithms. However, there are very few projects for violin etudes created based on artificial intelligence. In 2020, The Shanghai Symphony Orchestra Concert Hall hosted a special concert of Mozart's 80-year-old works, the last three symphonic pieces were generated by the AI based on learning a large amount of Mozart's past music, which inspire me. In this paper, classical violin etude (i.e., Rudolphe Kreutzer 42 Studies) is used for the data. Based on several studies, the LSTM model is the framework of this study. In the multiple times of changing seeds and temperature, the creation demo has creativity and similarity, but with less education. The educational meaning is also the future research to focus on. This study makes a step in making violin etudes from the classical violin etudes, which can create new etudes for future violin players. Moreover, the new etudes shed light on Inheriting and innovating the predecessors' composing style and thinking.

Keywords: Machine Learning; Music Composing; Computer Science; Deep Learning.

1. Introduction

With the development of computers and the usage of computers and digital devices, music composition, and its production have become extent than before. Its main interest to a composer is that computer synthesis implies an extraordinarily larger sound space [1]. Dealing with sound synthesis needs a variety of knowledge, which include physics, psychology, computer science, and mathematics. The next step of music composing is that programming and composition. Because of the intricacy of processing sound and translating music to physical specifications, high-level languages ought to be adopted to meet that need.

Machine learning has already been a widely used discipline. It studies how to imitate the learned behavior of humans and how to realize it on a computer, then getting new knowledge and skills. The core of machine learning is summarizing and synthesizing. Machine learning is already used in a wide range of applications such as search engines and medical diagnostics. It is one of the core research topics in artificial intelligence and neural computing. The existing computer systems and artificial intelligence have only limited learning ability and cannot meet the new requirements of science and technology and production. The discussion of machine learning and the progress of machine learning research will promote the further development of artificial intelligence

The attempt about learning musical form and using that knowledge to guide composition has been unsuccessful. The reason for this failure seems to be that RNNs cannot keep track of temporally distant events that indicate global music structure [2]. Long Short-Term Memory (LSTM) has the capability of fixing the failure [3, 4]. The LSTM network is an alternative architecture for recurrent neural networks inspired by human memory systems. In the current study, LSTM is a good mechanism for learning to compose music. By introducing gate functions into the cell structure, the long short-term memory (LSTM) could handle the problem of long-term dependencies well [5-7]. Studies on musical composition using LSTM found in the literature started in 2002 with the paper of Eck and Schmidhuber, who used the LSTM to learn musical forms of blues [8]. In the second stage, the network is conducted to compose new melodies in blues style without losing the relevant structures over a long time. Two additional experiments are further realized. To be specific, in the first one, the network learns a sequence of chords; while in the second, it learns to play the chords along with the melody.

This paper will realize violin etude composing. With the development of computer music and deep learning. Using models in deep learning to compose music become more and more widespread. There are all kinds of music created by artificial intelligence, e.g., funk music, blues music, pop music. Contemporarily, there is little attention has been paid to AI composing classical music. However, one concert at Shanghai Symphony Orchestra Concert Hall gave some instructions. They use AI to compose symphonies that are composed from the perspective of Mozart in his 40s, 60s, and 80s. Violin etudes as educational works, and it has significant meaning to continue creating new etudes for more and more young people who learn to play the violin. The rest part of the paper will introduce the dataset and research methods which include models, loss function, and evaluation of the models at first. The next part is results and discussion. In this section, one could get more details on model training, evaluations of the model, and the demo visualization. For the limitations and future outlook's part, this research will discuss the shortcut of the research and the prospect of violin etude composing in deep learning. The final section is the conclusion of the paper.

2. Data & Method

2.1 Data

In the program, this study uses the classical violin etude - Rudolphe Kreutzer 42 Studies as the dataset. First, the pdf version is downloaded on the internet. Then, by using musescore (music midi tool), the pdf version is converted to midi files. The midi files, it has 42 educational violin etudes separated by chapter. During the converting procedure, it loses some of the notes and rest because of the bad sharpness. Some chapters lose notes and the rest are deleted to avoid misunderstanding the computer. At last, there are 40 chapters(20.2kb) remaining.

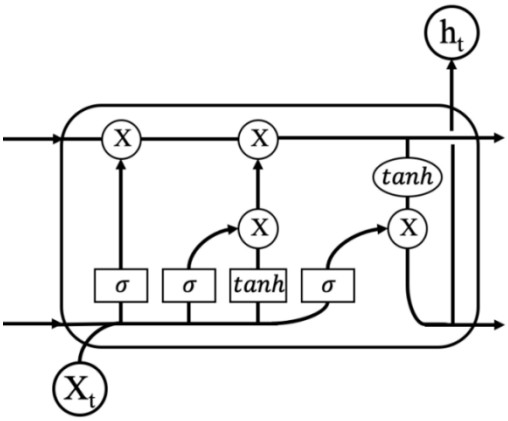


Fig 1. A sketch of the LSTM.

2.2 Models

This study uses the LSTM models to complete the training and generating processes as shown in Figure 1. The core of LSTM is the cell statement (the top conveyor) which is controlled by sigmoid and dot products. The sigmoid and dot product are also called control gates, which decide how much information to send. There are three different control gates: forget gate, input gate, and output gate. The role of the forget gate is to forget some of the past information (1). The role of the input gate is to memorize some of the information which is just input to the model (2, 3). Then, there is an important process is to combine the memory and present memory to the cell (4). The role of the output gate is to deliver the information to the next state (5, 6). The loss function is sparse_categorical_crossentropy. The sparse_categorical_crossentropy has the same mathematics deducing as categorical_crossentropy (7). Sparse_categorical_crossentropy is often used with softmax. The metric in the model is sparse_categorical_accuracy. The optimizer is Adam in the project. About the activation, this study chooses the softmax as a predefined activation function which is applied to the last dimension of the input data. The relevant formulae are given as follows

$$f_t = \sigma\left(W_f \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_f}\right) \quad (1)$$

$$i_t = \sigma(W_i \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_i}) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_C}) \quad (3)$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o \cdot [\mathrm{h_{t-1}}, \mathrm{x_t}] + \mathrm{b_o}) \quad (5)$$

$$h_t = o_t \tanh(C_t) \quad (6)$$

$$Loss = -\sum_{i=1}^{output\ size} y_i \cdot \log \hat{y}_i \quad (7)$$

2.3 Preprocessing

First, this research loads the midi file by using the music21 library to a midi list. Within music21, the computer can understand the language that connects music and computer. It is important to think about the way to translate the midi file to an understandable pattern for the computer. In the midi file, one note has pitch and duration [9, 10]. Therefore, the study needs to build a pattern. In this pattern, pitches are considered as different numbers (int): C4 is equal to 60. One step is equal to 1, so C4# is equal to 61. The rule of note length is stipulated that 1 is equal to the duration of the quarter length. And using one character("_") to represent 0.25, and the quarter note is grouped by one pitch and three characters. This group makes up the whole quarter note("60", "_", "_", "_"). It is necessary to remeber the rest in the midi list. This paper makes "r" to represent rest. Hence, when one hears a break in the midi files, it supposes to be supplemented by "r". As different chapters have different keys, it is not easy to generate the data file for the computer to build the model. The next step is to transpose the songs into the same key (C major of A minor). A data file that contains the numbers, "_" and "r" was generated by the rules above. In the file, there are still strings that can not be used for one-hot code. In this case, a map that maps different integers, and durations, and rests into a set of integers(60: 0, 61: 2) is created. The total number of elements on the map is 58. Using the map and the data file, a new list called int_songs which only has integers occur. Final part of the preprocessing is to generate the inputs and targets sequence as an array. This is the most important part of building a model. Priamrily, this study set a sequence length(64: 16 beats). Sequence length is a core parameter in the whole program, which will decide how long and how many the sequence should be trained. This process uses a loop to create the inputs and targets. The range parameter of the loop is the number of sequences that is equal to the length of the int_songs minus sequence length. The inputs[0] is equal to int_songs[i : i+sequence length], the targets[0] is equal to int_songs[i + sequence length]. The meaning of the targets list is that if the computer receives the input[0], the corresponding target is targets[0]. Subsequently, the function to convert the input list is used to one-hot encoding(a size is a different number in the map, so it is 58). Lastly, Transforming inputs and targets lists into an array.

3. Results & Discussion

3.1 Models

The number of layers is 256, the learning rate is 0.001, epochs are 50, the batch size is 64, and output units are 58. Those are the important parameter of the model. This research imports the Keras to build the LSTM model. Here are two steps. The first one is to build the model. Using the Keras functions to create the frame of the model which includes the input gate, output gate, and forget gate. In the input gate, this model set 256 layers in the model with the inputs array which is getting above. To avoid overfitting dropping out some parts for each input is necessary, this forgetting parameter is set to 0.2. The output gate is decided by the output units which are equal to the different integers in the map and the activation-softmax function. In the model training, the model.fit function's purpose is to realize the purpose. In each epoch and 64 batch size, the model will return the loss and accuracy with learning the inputs and targets. The more epochs were completed, the less and the accuracy higher.

Firstly, a seed melody is set for the model, in which case, the model can predict the next melody sequence according to the seed. Thus, the seed sequence is split into a list, then mapping the seed list to integers. If the seed sequence is longer than the sequence length (64), it will be limited. Afterwards, the list is converted to the one-hot encode so the model could get it correctly. After getting the seed, the next step is to predict the following melody sequences. Using the model. predict function, one can obtain the model and invoke the prediction message. It can receive a batch of samples and then it will return the probability of each batch. Nevertheless, the first probability distribution is needed because there is only one sample in the model. The result of the prediction is an array of different distribution probabilities of different symbols. If the sum of the values is equal to 1, the distribution probability of each symbol is considered as the probability of the symbol coming out after the softmax. Temperature sampling would be better because it is more flexible. If the temperature(t) goes toward infinity, the probability distribution gets completely remodeled (8). As a result, the probability distribution becomes a homogeneous distribution. If the temperature(t) gets closer to zero, the probability is going to be remodeled in such a way that the original value that has the highest probability now is picked. So, the whole thing is getting deterministic. Then, the expectant symbol which is attached to the end of the seed is produced. After finishing all the predictions and mapping the integer back to the midi symbols that music21 can read and get, the demo is generated.

$$P(x|x_{1:t-1}) = \frac{\exp(\mu_t/t)}{\sum_{t'} \exp(\mu_{t'}/t)} \quad t \in [0, 1) \tag{8}$$

3.2 Evaluations

According to the multiple attempts, it shows that the temperature should adjust on time if the seed changes. Thus, the model will retain the rhythm pattern of the original score based on the input seed and make some innovations to the melody. There are two seeds, for example, seed 0 is from the first four beats of the first line of the original score and the temperature is set to 0.4, and seed 1 is from the first four beats of the third line of the original score and the temperature is set to 0.7 as shown in Fig. 2. The two presented results were selected from two seed-generated demos (result 0 and result 1) as shown in Fig. 3.
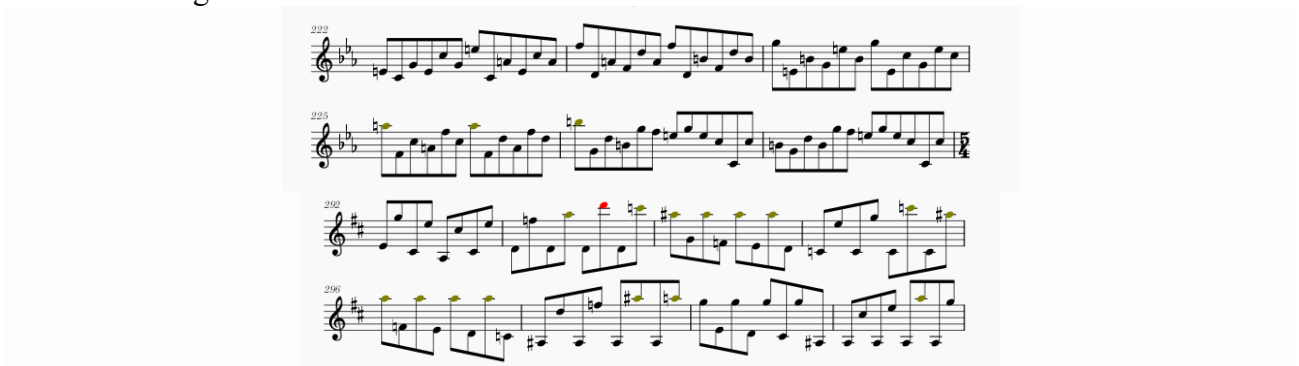


Fig 2. The original score of seed 0 and seed 1.

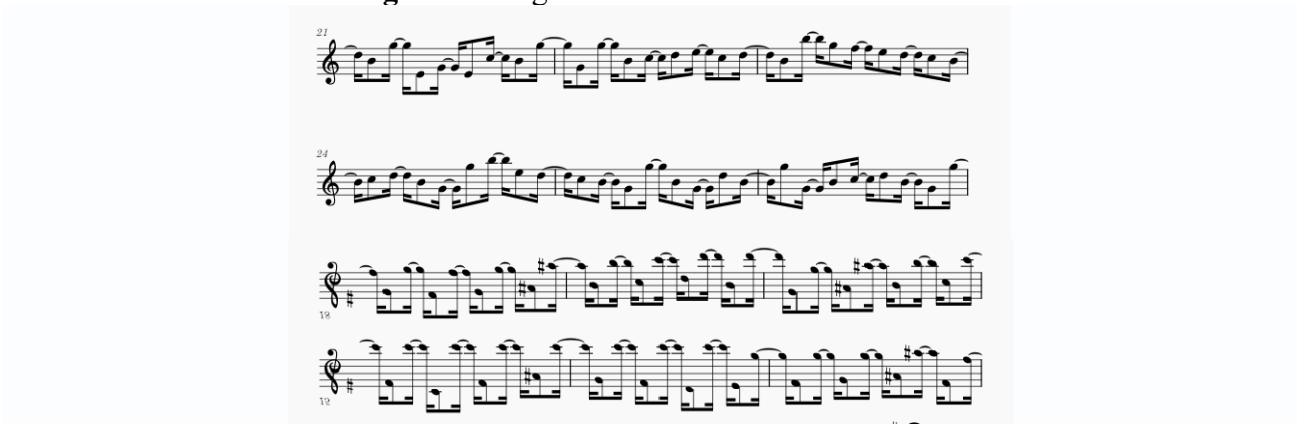


Fig 3. The results score of result 0 and result 1.

First, the degree of similarity, the seeds are drawn from the random chapter's beginning, which means that the beginning as a symbol for each chapter, the model could predict the next score to continue the origin chapter, but it does not show in the way. The reason for the failure could be that the pattern for each chapter is no more regular or frequent than the one in the middle of the chapter. As a consequence, the computer cannot follow the origin score. Another reason for this could be the epochs and batch size are not big enough. If the epochs and batch size are bigger to fit the need, it might show a better result.

Subsequently, the degree of creativity. First, based on the seeds given, the model creates a rhythm pattern that is typical of the original score. Because those rhythm patterns appear a lot in the original score, the computer has been trained a bunch of times. In the case of keeping the rhythm type consistent, the melody is innovative. It is not just following the melody patterns. In the original score, after the 6 pitches are alternatively climbing up and getting to the top pitch, it will drop down immediately. In the generating melody, it changes the climbing and dropping pattern. It drops down after 2 or 3 or even more climbing notes.

Finally, the degree of education. As an innovation of violin etudes, A good violin etude should be melodic and can be practiced by the person practicing it with a left hand, right hand, and both hands. So according to the multiple different results, it can meet the requirements. But it should be examined and evaluated by a professional.
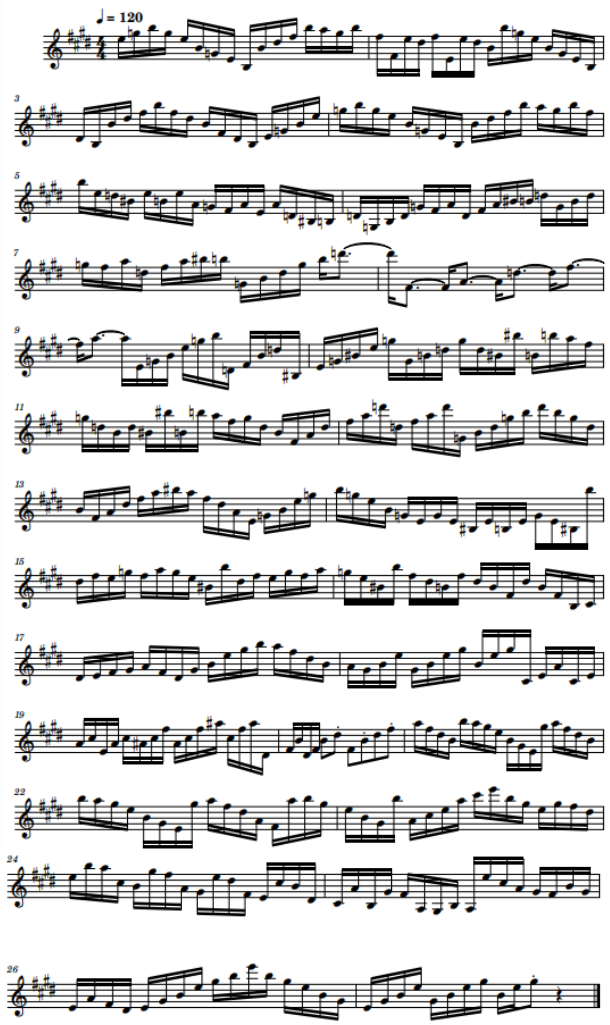


Fig 4. Visualization of the Demo.

3.3 Demo

Eventually, the demo is constructed, where the Seed is "81 84 88 84 81 76 72 69", temperature 0.4. The final demo is a version that partially imitates the rhythm rule of chapter 2, but it creates a different

and wonderful melody as depicted in Fig. 4. The online version can be connected through the link given in Ref. [11].

4. Limitations&Future Outlooks

The data of this study is from the same series of violin etudes, which have three limitations. The first one is that the rhythm pattern is different in each chapter. Thereby, an intact series of violin etudes cannot be created simultaneously. On the other hand, creating violin etudes should also consider the rationality of fingering. Owing to the temperature in the study, some fingering in the final demo is not reasonable for a real violin player. It creates trouble for using demos in real practice, it should be adapted to a better version. The second one is if the purpose is that train one specific rhythm pattern in real practice, with different fingering techniques. Hence, there is only one similar or same pattern in one chapter, which means the computer only gets one sample in training and generating. By changing temperature, the results are almost the same with a little different fingering. Lastly, the initial purpose of this study is that create educational violin etudes for future violin players and inherit the style and composition of their predecessors, so the data selection is a problem. In music history, there are different genre composers from different eras, so in this study, the data extraction is not comprehensive.

For violin etudes AI composition is just the beginning, this paper is mainly a preliminary exploration of the possibility of etudes AI composition. The discovery of the above limitations in the study generates outlooks for the future. First, in the future, the classification of violin etudes can be classified not only by the author but also by rhythm, fingering, or melody direction. This process needs to be combined with experts from the conservatory of music and better AI models to achieve this classification. After completing this step, one can select different types of etude data sets according to different needs for training and generating new music to realize the educational significance of violin etudes. According to the same rhythm, a series of etudes with different melodies can be generated. This kind of etudes can practice fingering change and coordination of left and right hands to achieve the purpose of etudes. Once etudes have been explored completely, people can aim for more complex forms: solo, duo, concerto, and symphony compositions. These creations will help future players not forget the music of traditional composers and continue their style.

5. Conclusion

In summary, this paper investigates violin etudes composed based on the LSTM model. Specifically, this paper creates violin etudes according to Rudolphe Kreutzer's 42 Studies. According to the analysis, the process is converting the midi series to the data form that the computer can understand, training the models, and generating a demo. Besides, by adjusting the temperature parameter, batch size, and epochs, the model is tuned to better fit. Moreover, a complete demo which is fit the pattern of chapter 2 of the original score from Rudolphe Kreutzer's 42 Studies. The demo has certain innovations and imitations. Nevertheless, ascribed to the limitation of data extraction and selection, the final result is not educational. In the future, the outlook of creating more possible series etudes will be realized by solving the classification of violin etudes. Overall, these results offer a guideline for future violin etudes composing and inheriting predecessors' works.

References

[1] Cheng Qiyun, Sun Caixin, and Zhang Xiaoxing. Short-Term load forecasting model and method for power system based on complementation of neural network and fuzzy logic. Transactions of China Electrotechnical Society 19.10 (2004): 53-58.

[2] Fang Fang. Research on power load forecasting based on improved BP neural network. Harbin Institute of technology, 2011.

[3] Roads Curtis. The computer music tutorial. MIT press, 1996.

[4] Sherstinsky Alex. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena 404 (2020): 132306.

[5] Selvin Sreelekshmy, et al. Stock price prediction using LSTM, RNN and CNN-sliding window model. 2017 international conference on advances in computing, communications and informatics (icacci). IEEE, 2017.

[6] Yu Yong, et al. A review of recurrent neural networks: LSTM cells and network architectures. Neural computation 31.7 (2019): 1235-1270.

[7] Eck Douglas, and Juergen Schmidhuber. Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. Proceedings of the 12th IEEE workshop on neural networks for signal processing. IEEE, 2002.

[8] Eck Douglas, and Jürgen Schmidhuber. Learning the long-term structure of the blues. International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2002.

[9] Klapuri Anssi, and Manuel Davy, eds. Signal processing methods for music transcription, 2007.

[10] De Mantaras R. Making music with AI: Some examples. Proceeding of the 2006 conference on Rob Milne: A Tribute to a Pioneering AI Scientist, Entrepreneur and Mountaineer. 2006.

[11] Demo online: retrieved from: https://pan.baidu.com/s/1WNQ-VKvr227UnZurjNfhDA.