Computer and Information Science; Vol. 11, No. 3; 2018
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Music Generation Based on Convolution-LSTM

Yongjie Huang[1], Xiaofeng Huang[1] & Qiakai Cai[1]

[1] Internet of Things Engineering, Electrical and Information College of Jinan University, Zhuhai, China

Correspondence: Yongjie Huang, Internet of Things Engineering, Electrical and Information College of Jinan University, Zhuhai 519070, China. E-mail: huangyongjie19@foxmail.com

Received: May 22, 2018 Accepted: May 30, 2018 Online Published: June 18, 2018

doi:10.5539/cis.v11n3p50 URL: https://doi.org/10.5539/cis.v11n3p50

Abstract

In this paper, we propose a model that combines Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) for music generation. We first convert MIDI-format music file into a musical score matrix, and then establish convolution layers to extract feature of the musical score matrix. Finally, the output of the convolution layers is split in the direction of the time axis and input into the LSTM, so as to achieve the purpose of music generation. The result of the model was verified by comparison of accuracy, time-domain analysis, frequency-domain analysis and human-auditory evaluation. The results show that Convolution-LSTM performs better in music genertaion than LSTM, with more pronounced undulations and clearer melody.

Keywords: Convolution-LSTM, music generation, feature extraction

1. Introduction

Music is one of the most widely used signal streams. However, the cost and difficulty of music creation is increasing, and more and more people are starting to like music of small crowd, which will cause music unable to meet people's needs.

However, computer music creation is still a new and difficult direction. In this field, scholars have made many attempts and put forward many methods and conclusions. In recent years, shallow structure music generation algorithms have been proposed one after another. For example, Frank et al. proposed a Tree-Base music generation method, Walter et al. proposed a HMM-based music generation model, Huang et al. proposed a deep belief network for music generation. The above methods are based on the characteristics of the music sequence data using a sequence model for modeling, completing the generation task through the signal reconstruction theory.

With the development of deep learning and neural networks, especially the achievements of CNNs, RNNs, and other tasks in natural language processing, image tagging, and speech recognition, academia has begun to apply deep learning techniques to music generation. For example, Matic uses a combination of neural networks and cellular automata to generate melody, Boulanger-Lewandowski et al. used RNN-RBMs for polyphonic music generation, Lyu proposed a music generation model based on LSTM and RTRBM, Choi et al. proposed a music generation method in MIDI format based on the RNN.

CNN and LSTM have achieved good results in text categorization and text generation. CNN can extract potential features, and it may also identify melodic features in the music. LSTM is suitable for time series modeling, which inspires us to combine CNN and LSTM and apply them to music generation.

2. Convolutional Neural Network and LSTM

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) can learn the characteristics of two-dimensional data, and complete the extraction and classification of features. The CNN avoids explicit feature extraction, but implicitly learns features from the training data, and the neuron weights on the same feature mapping surface are the same, and the network can learn in parallel. In music sequence modeling, CNN can identify and extract the potential features of the melody, so as to better model.

2.2 Long Short-Term Memory

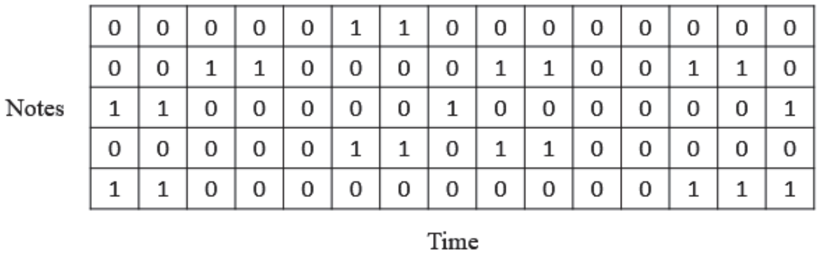
Recurrent Neural Network (RNN) is a neural network for time-series modeling. In RNN, the output of a network at a certain moment is the result of the interaction of the current input and all historical inputs. Long Short-Term

Memory (LSTM) is a special type of RNN that can learn long-term dependencies and is suitable for processing and predicting important events with relatively long intervals and delays in time series. Since music sequences are time series with long-term dependency, it is appropriate to use the LSTM.

3. Convolution-LSTM for Music Generation

3.1 Data Representation

When reading a MIDI file, sampling and quantification are required, so as to represent the music score using a matrix. We set vertical axis of the matrix to different notes and horizontal axis to time. The MIDI file is sampled at a rate of 1/8 second, with 1 indicating that the musical note was on and 0 indicating that the musical note was off. The sampled musical score matrix *X* can be shown in figure 1, where there are 156 different Notes (simplified into 5 notes in the Figure 1).



	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	1	1	0	0	1	1	0
Notes	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1
	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1

Time

Figure 1. The musical score matrix

3.2 Establishing Convolution-LSTM Model

CNN has the function of feature recognition and extraction, which inspired us to use the convolution layer to extract the features of musical melody. After using the convolution layer to extract feature of the music score matrix, the extracted feature map is input into the LSTM for time series modeling, and then the state of each note at the next time is generated.

It should be noted that the pooling layer that commonly used in CNN cannot be applied to this model because the pooling layer has the shift-invariance, which may lead to modified tone or looping melodic.

The structure of the Convolution-LSTM (C-LSTM) model is shown in figure 2 (the number of squares in the figure does not represent the actual number of data).

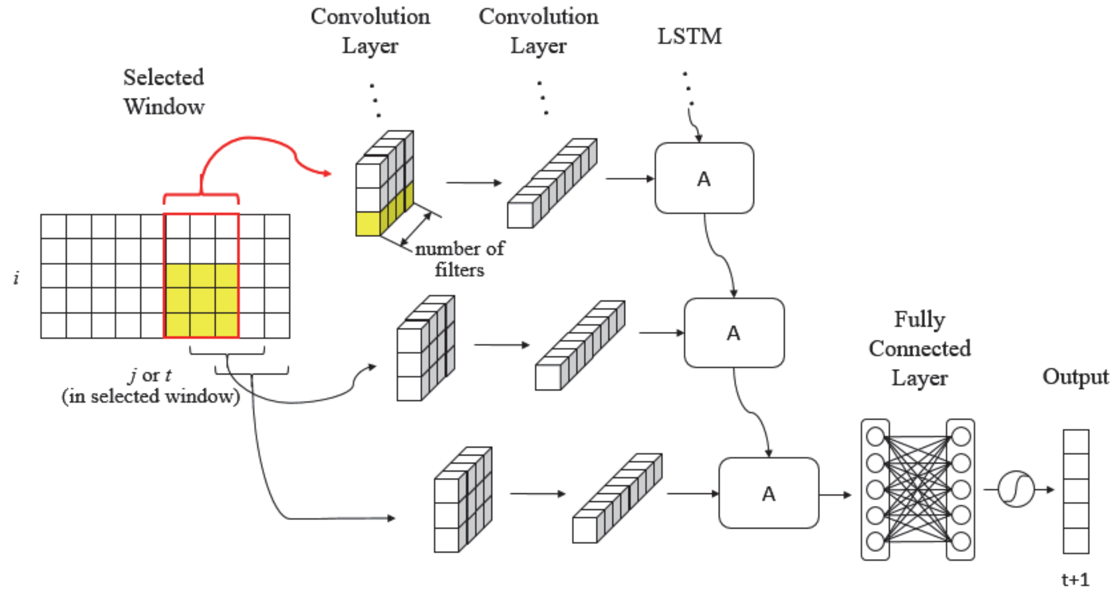


Figure 2. The structure of the C-LSTM model

1) Selected Window

We select a fixed length *N* in the time direction of the matrix and then select [*t-N+1,t*] as the window range at time *t*. The selected window will be the complete input for each training of the model.

2) Convolution Layer 1

This layer convolve with multi-core in the selected window. The normal convolution result should be a three-dimensional matrix, but we split it in the direction of the time axis and split it into many two-dimensional matrices, so as to sequentially input into the LSTM. This layer can be expressed by the following formula:

$$y^{(1)l}_{i,j} = \theta(\sum_{m}\sum_{n} x_{i+m-1,j+n-1} \cdot w^{l}_{m,n} + b^{l}) \tag{1}$$

where *m,n* represents the height and width of the convolution kernels, *l* represents different convolution kernels, *w* represents the weight of the convolution kernels, b^l represents the offset coefficient, θ represents the activation function and *Relu* is selected here.

3) Convolution Layer 2

In a normal convolutional neural network, the convolutional layer may be followed by a pooling layer, which can reduce parameter and feature extraction. However, the max-pooling has a shift-invariance, which allows the feature to have the same output in any position. But in music modeling, this will directly lead to modified tone or looping melodic, so the pooling layer cannot be used here. The second convolutional layer is used here to reduce parameters and extract features.

$$y^{(2)l'}_{i,j} = \theta(\sum_{m'}\sum_{n'} y^{(1)l}_{i+m'-1,j+n'-1} \cdot w^{l'}_{m',n'} + b^{l'}) \tag{2}$$

where *m',n'* represents the height and width of the convolution kernels, *l'* represents different convolution kernels, *w* represents the weight of the convolution kernels, $b^{l'}$ represents the offset coefficient, θ represents the activation function and *Relu* is selected here.

4) LSTM

The LSTM receives data in sequence from previous layer. The number of input data of each time is equal to the number of filters in convolution layer 2. After receiving data from previous layer, LSTM will output the result.

The input of the LSTM can be expressed as:

$$s_t = y^{(2)l'}_{i,t} \tag{3}$$

where *t=j+n-1* and *i=1* (because the kernels of the convolution layer 2 cover all of the notes in vertical axis).

The internal structure of LSTM and the information transfer process can be expressed as follows.

$$f_t = \sigma(W_f \cdot [h_{t-1}, s_t] + b_f) \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, s_t] + b_i) \tag{5}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, s_t] + b_C) \tag{6}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{7}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, s_t] + b_o) \tag{8}$$

$$h_t = o_t * \tanh(C_t) \tag{9}$$

where f_t represents the forgetting gate, i_t represents the input gate, $\tilde{C}_t$ represents the state of the cell at the previous moment, C_t represents the state of the current cell, o_t represents the output gate, and h_t represents the output of the current cell.

The final output of the LSTM can be expressed as:

$$Y^{(3)} = h_t \tag{10}$$

5) Fully Connected Layer

The fully connected layer receives the final output of the LSTM and outputs the result after one fully connected layer. The result is the states of a group of notes at time *t+1*.

$$y_v = \theta'(w_{u,v} \cdot y^{(3)}{}_u + b_{u,v}) \tag{11}$$

where y_u represents the outputs of the previous layer, y_v represents the final outputs, *w* represents weight coefficients, *b* represents offset coefficients, and θ' represents activation function and *Sigmoid* is selected here.

After the above process is completed, the selected window moves one unit towards the time axis direction, and the above process is repeated for training or generating purposes.

3.3 Training Model

We downloaded 500 MIDI-format piano songs from *midiworld.com* as the training data of the model. We define the loss function as binary-crossentropy and use the Adam optimization algorithm to train the model. The parameters of the training process are shown in table 1.

Table 1. Parameters of the training process.

	Value
N (length of the selected window)	90
Size of kernels in convolution layer 1	3*3
Number of kernels in convolution layer 1	32
Size of kernels in convolution layer 2	154*1
Number of kernels in convolution layer 2	128
Number of output parameters in LSTM	1024
Length of each generation (unit of time)	9
Number of neurons in fully connected layer	156*9=1404
Batch size	32
Epochs	64

4. Result and Analysis

Since the model is for generation which has more uncertainty and randomness, there is no very objective evaluation indicator to evaluate the model's effect. Therefore, we first use the accuracy of the prediction to simply evaluate the effect of the model. Then, we randomly take a song (named *Duet No. 3, BWV 804*, not in the training data) as a sample song, and then compare the sample song with the generated song and analyze them.

4.1 Accuracy of Prediction

We established RNN, LSTM, and C-LSTM respectively and tested them. The prediction accuracy of each model on the status of the next time is shown in the table 2.

Table 2. Accuracy of prediction for different models

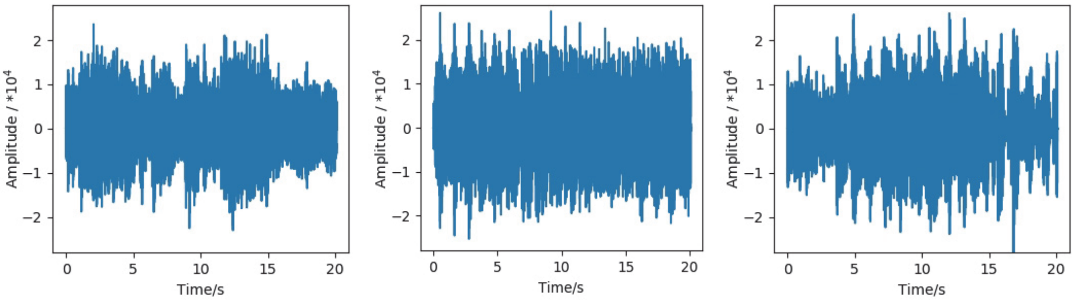
	Accuracy
RNN	95.15
LSTM	96.74
C-LSTM	97.81

When training the RNN, the vanishing gradient problem was found and the accuracy was low. The accuracy of C-LSTM was the highest among three models, followed by LSTM.

However, this is a model for generating, which is different from the model used for forecasting. Therefore, using the accuracy to evaluate the model is not very appropriate. Therefore, we analyze from more perspectives and further analyze the effects of music generated by LSTM and C-LSTM.

4.2 Time-domain Analysis

Time-domain analysis can be very intuitive to see the music waveform changes over time, mainly through time-domain analysis can observe two pieces of information: volume and volume fluctuations. Among them, the ups and downs of music can show emotional changes to some extents. The faster the music fluctuates, the more expressive it is. On the contrary, if the music does not fluctuate, or fluctuates slowly, the emotions expressed are relatively simple and boring.



(a) (b) (c)

Figure 3. Time-domain waveform of the three songs

Note. (a) is the sample song, (b) is the song generated by LSTM, and (c) is the song generated by C-LSTM.

Sample song: The volume has a very obvious ups and downs, that is, the change in amplitude is very obvious. This is very similar to the music we hear usually, and it can clearly show the change of emotions.

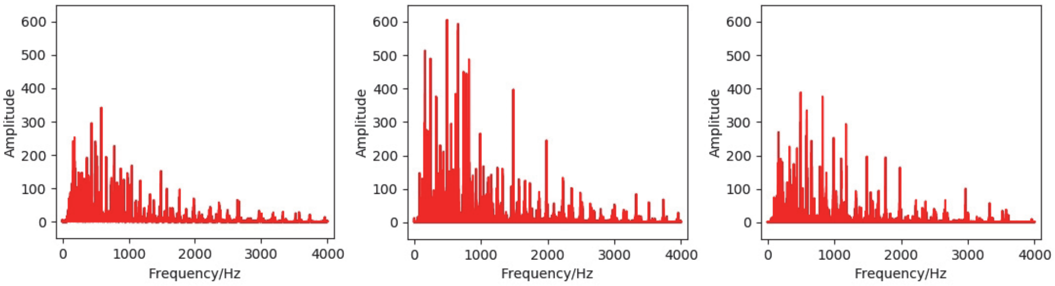
LSTM: The volume has almost no ups and downs, or the process is very slow, so it lacks expressiveness in emotional changes. In addition, its volume is always at a large value, which may cause the listener's ear cannot be rested and easily lead to hearing fatigue.

C-LSTM: Compared with the song generated by LSTM, the volume has some ups and downs, but the ups and downs are not as fast as the sample songs. We can also see the high tide and low tide and so the music has some sense of hierarchy.

From the comparison analysis of the three models in time domain, the effect of C-LSTM is better than that of LSTM, but the song generated by C-LSTM is still not as good as the sample song.

4.3 Frequency-Domain Analysis

Frequency domain is a coordinate system used to describe the characteristics of a signal in frequency, and the frequency domain diagram shows the value of the components at each frequency of a signal. Using the Fourier transform, a time-domain signal can be converted into a frequency-domain signal, which can provide more abundant information than time-domain signal.



(a) (b) (c)

Figure 4. Spectrum of the three songs

Note. (a) is the sample song, (b) is the song generated by LSTM, and (c) is the song generated by C-LSTM.

Sample song: The frequency is mainly concentrated below 1 kHz and is uniformly distributed in 0-1 kHz, which is in line with the spectral law of the general songs.

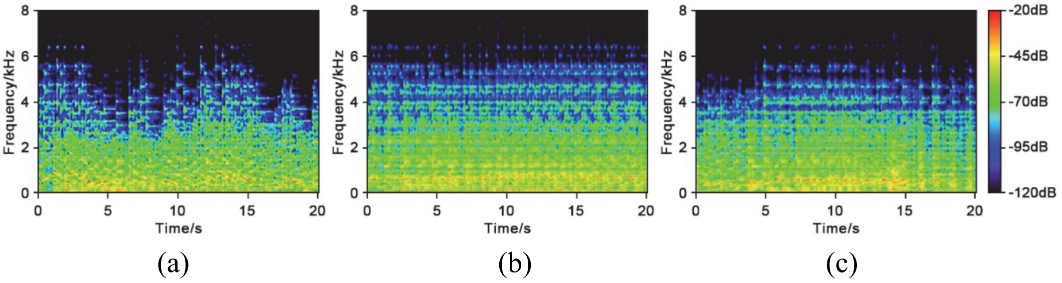
LSTM: The overall amplitude of the frequency is relatively large, and most of them are concentrated below 1.3 kHz. However, some high-frequency frequencies also have large amplitudes and the frequency distribution is dispersed. Such music is more scattered and messy, and the melody is not clear enough.

C-LSTM: The frequency is mostly concentrated below 1.2 kHz. It can be seen that some high frequencies also have large amplitudes, and the frequency distribution is also dispersed. However, the high-frequency components are slightly smaller than the song generated by LSTM.

From the comparison and analysis of the three in the frequency domain, the frequency of the song generated by LSTM is not sufficiently concentrated, and the frequency spectrum of the song generated by C-LSTM is closer to the sample song. It can be seen that C-LSTM can accurately identify and generate the frequency of the song.

4.4 Sonogram

Sonogram is widely used in the analysis and processing of audio signals. Sonogram is a two-dimensional map, but each point on the map corresponds to a three-dimensional value, the horizontal axis represents time, the vertical axis represents frequency, and different colors represent different intensities of sound. Using the sonogram, we can make deeper analysis of the music.



(a) (b) (c)

Figure 5. Sonograms of the three songs

Note. (a) is the sample song, (b) is the song generated by LSTM, and (c) is the song generated by C-LSTM.

It can be seen that the changes in the frequency of the songs generated by the sample song and the C-LSTM over time mainly come from the changes in the high-frequency component, that is, changes in the number and distribution of high-frequency notes. This means that the low frequency is responsible for the rhythm while the high frequency is responsible for the melody, which is consistent with the general song. The frequency distribution of music generated by LSTM hardly changes with time, and the frequency is relatively simple.

4.5 Human-auditory Evaluation

Compared with using quantitative indicators for analysis, human auditory is more sensitive, and can directly evaluate the melody of songs. We were inspired by Turing's test ideology and decided to adopt a similar approach to this evaluation. We selected 5 students as experimenters to listen to each of the 10 segments (10 seconds for each segment, from different songs) of the sample song and songs generated by LSTM and C-LSTM on the premise that they did not know the sources of the songs. Then we let the experimenter determine whether the segments was created manually or computer generated. The results show that among the 10 different segments of each model, the sample song segment and the LSTM-generated segment can be correctly recognized, and two segments of the C-LSTM-generated segment are mistaken for artificial creation. It can be seen that the music generated by C-LSTM is closer to that of artificial music, but there is still a distance.

In addition, we also conducted interviews with experimenters. Experimenters generally say that computer-generated music has a relatively clear sense of rhythm, but the melody is relatively unclear, but it also has its own characteristics: the songs generated by LSTM does not change much, and there are more repetitions and loops; the songs generated by C-LSTM has some changes and it will not repeat the same melody exactly.

5. Conclusion

In this paper, we establish LSTM and C-LSTM for music sequence modeling to achieve the purpose of music generation, and we also compare and analyze the generated music. The results showed that compared with LSTM, the music generated by C-LSTM had a more pronounced high tide and low tide, which did not cause hearing fatigue and the melody was more pronounced. Therefore, C-LSTM is more effective in music generation, and it provides another optional model for music generation.

References

Boulangerlewandowski, N., Bengio, Y., & Vincent, P. (2012). Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. *Chemistry A European Journal, 18*(13), 3981-3991.

Broughton, S. A., & Bryan, K. (2008). Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing. Wiley.

Choi, K., Fazekas, G., & Sandler, M. (2016). Text-based lstm networks for automatic music composition.

Chris, O. (2015). *Understanding LSTM Networks.* Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs.

Drewes, F., & Hogberg, J. (2007). An algebra for tree-based music generation. *International Conference on Algebraic Informatics* (Vol.4728, pp.172-188). Springer-Verlag. https://doi.org/10.1007/978-3-540-75414-5_11

Graves, A. (2013). Generating sequences with recurrent neural networks. *Computer Science*.

Huang, Q., Huang, Z., Yuan, Y., & Tian, M. (2016). A New Method Based on Deep Belief Networks for Learning Features from Symbolic Music. *International Conference on Semantics, Knowledge and Grids* (pp.231-234). IEEE. https://doi.org/10.1109/skg.2015.12

Ivana D. M., António, P. O., & Amílcar, C. (2013). Automatic melody generation using Neural Networks and Cellular Automata. *Neural Network Applications in Electrical Engineering (NEUREL)*. https://doi.org/10.1109/neurel.2012.6419972

Kong, X., & Guan, J. H. (2009). Waveform Music Retrieval Measured by Similarity of Spectrogram. *Computer Engineering and Applications, 45*(13), 136-141. https://doi.org/10.3778/j.issn.1002-8331.2009.13.040

Merwe, A. V. D., & Schulze, W. (2011). Music generation with markov models. *IEEE Multimedia, 18*(3), 78-85. https://doi.org/10.1109/mmul.2010.44

Qi, L., Wu, Z., Zhu, J., & Meng, H. (2015). Modelling high-dimensional sequences with LSTM-RTRBM: application to polyphonic music generation. *International Conference on Artificial Intelligence* (pp.4138-4139). AAAI Press.

Wang, L. Z., Li, H. X., Lin, M., Li, J., Lou, F. B., & Chen, J. (2008). The time and frequency domain analysis of weak electromagnetic signals in plants. *Journal of China Jiliang University, 16*(4), 294-298. https://doi.org/10.3969/j.issn.1004-1540.2005.04.009

Copyrights