Journal of Engineering and Applied Sciences

A Experiment of Reccurent Neural Network Methods for Generating a Music

Dimas Fiqh Puskoaji, Tito Waluyo Purboyo and Randy Erfa Saputra
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

Key words: Music Generation, Recurrent Neural Network (RNN), Python, music, MIDI

Corresponding Author:
Dimas Fiqh Puskoaji
Department of Computer Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

Page No.: 2619-2622
Volume: 15, Issue 12, 2020
ISSN: 1816-949x
Journal of Engineering and Applied Sciences

Abstract: Music is a composed sound so that it contains rhythm, song and harmony. Every music has different chord provisions and arrangements which makes making music difficult. Based on these problems, a solution was made in the form of Music Generation with the Recurrent Neural Network (RNN) method to facilitate the making of a music. Recurrent Neural Network (RNN) is a class of artificial neural networks where the connection between nodes forms a directed graph along the sequence. This allows it to show temporary dynamic behavior for a time sequence. The data used is in the form of a Musical Instrument Digital Interface (MIDI) format file with sufficient amounts. The system will produce a midi file that has undergone the process of making chords and merging randomly, so that, it becomes a new music. Then implemented in Python. The output in this system produces a music that allows it to be processed again in the Digital Audio Workstation (DAW) to produce the desired music.

INTRODUCTION

In this era, making music automatically is very necessary because many people who want to make music but have less musical skills do not even understand music at all. With the technology, we can make music quickly even though, we do not understand music by making a MIDI from studying the desired MIDI results. In the research I will do using the file format Musical Instrument Digital Interface (MIDI) which contains many layers of digital instructions including pitch height (pitch), the volume level of each note, to the transport functions (play, pause, stop) and other functions. reading MIDI files, the system will translate existing MIDI files into notes and chords by the help of the Music21 module and will make a new song composition with a Music Generator application that uses the Recurrent Neural Network method.

MATERIALS AND METHODS

In this study, we discuss and assess song results from the recurrent neural network process.

Music: Music is the art of sound in the form of songs or compositions that express the feelings and thoughts of the creator through the elements of music that is rhythm, melody, harmony, shape and structure of the song. According to Syafiq "music is the art of expressing ideas through sound whose basic elements are

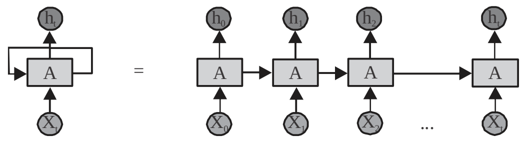

Fig. 1: Reccurent neural network loop[1]

melody, rhythm and harmony with supporting elements in the form of ideas, nature and color of sound".

Recurrent neural network: This recurrent neural network is known as a technique that has the ability to classify problems. A system that processes data repeatedly, reusing data that has been used in the previous process to process the next input data[1].

From Fig. 1, input from Xt then it processed in A after that before it becomes an output of the value of A, this A value issues a new input wich will later be used in the next process and so on. After Xt processed in A, it will produce output an store it to ht.

LSTM: The origin of LSTM is because RNN has a deficiency in the process that is when the output file will be reinserted as input, the output from the RNN process allows the output to be the same as the previous value. Then the memory is made so that it can store the previous output, so that, no same output occurs as before[2].

When the output will be re-entered as a new input will be stored in memory but at this stage the memory can only save and delete the recordings from the RNN then the selection is added. A general picture of long-short term memory is in Fig. 2[3].

MIDI FILE: Musical instrument digital interface or commonly called MIDI is an international hardware or software standard for exchanging data between electronic music devices and computers from different brands[4]. MIDI captures event notations and changes in attribute and tone accents and then encodes them into a digital message and forwards them as messages to other devices to manage the sound produced and its parameters. MIDI is usually recorded with sequencers such as Fruity Loops, Logic Pro, Cubase, Studio One, etc. MIDI also carries musical events consisting of notation, pitch and pressure (velocity), signal volume control (automation), vibrato, audio panning and clock signal as a tempo regulator. In one MIDI file there are usually sixteen channels of music information, each of which can be directed to a different device. Figure 3 is an illustration when the MIDI file is opened in a piano roll in DAW.

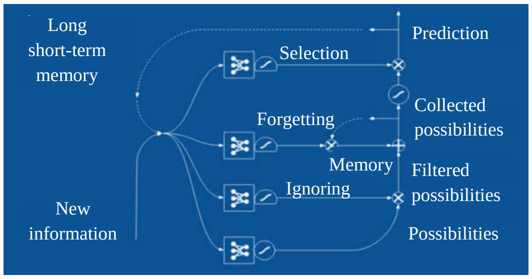

Fig. 2: Diagram flow LSTM[5]

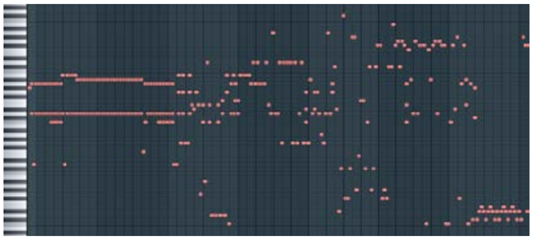

Fig. 3: The MIDI on piano roll

DAW (Digital Audio Workstation): Digital Audio Workstation or commonly known as DAW is a software used to record, edit and produce audio. DAW can also be used to turn MIDI files into audio by inserting notations in MIDI files into virtual studio technology-instrument or commonly known as VSTi. VSTi functions as a Virtual Instrument that can play notations that are formed in MIDI files and produce a sound (Wikipedia, "Digital Audio Workstation") (Fig. 4).

Process: The first thing to do is load the contents of the midi file to find out the chords and notes. At this phase the midi file is tested on the song_midi folder to get the training data file used in the process of making new music. In this test the author uses as many as 25 MIDI files to be tested.

In Fig. 5, contains many midi files that will be used during the training process of this model. The following MIDI files are obtained by downloading on a website that provides MIDI files. In the next stage, the writer tests with the system that has been made.

In Fig. 6, the system Load songs that are in the midi_song folder to be performed in the midi extraction process to get notes and chords that are on the midi. The midi extraction process is assisted with the Music21 module. After getting all the notes and chords that are in the MIDI file that was tested, the next process the system will make the training data file. This system using 50 epoch. In this process the training data file is obtained per-epoch which will run and the file that will be obtained is only a file that has less loss than the previous epoch. The system performs a process of training data with MIDI

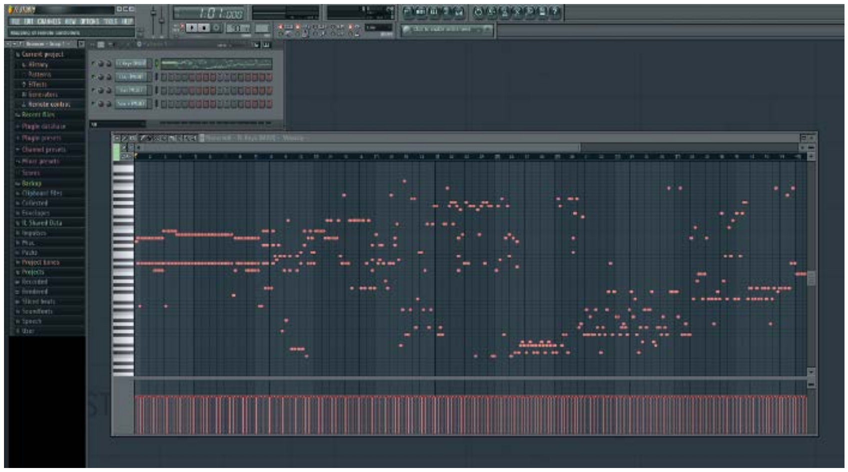

Fig. 4: DAW fruity loop interface

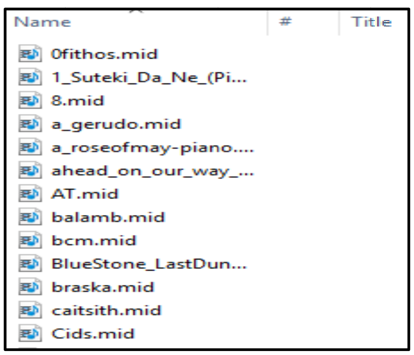

Fig. 5: File train MIDI

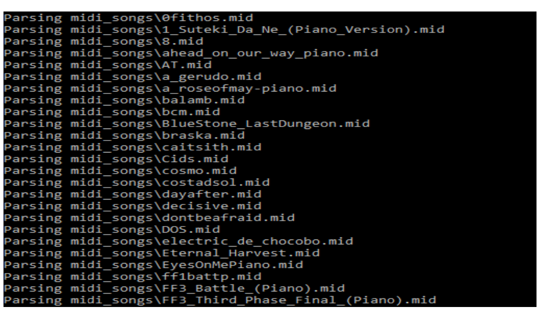

Fig. 6: Load songs

data that was loaded during the MIDI loading process. The results of this data training process can be seen in the output folder file obtained from the results of the training data process. Then, the files can be used as data to predict notes and chords that will be carried out in the next process.

Figure 7 are result from the training data process. then the system will use one of the results of the training data process that has the smallest loss of all files. Before testing the file is changed to weights HDF5, so, the

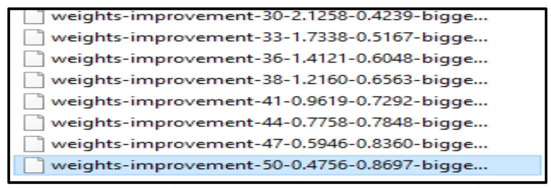

Fig. 7: Result of training data

system can load the file. The file used has a loss of 0.4756 and an accuracy of 0.8697. Loss is the error between the prediction (output) of the model and the target in the training data. The smaller the value of the loss the results obtained will be maximum. In this process, the system also predicts notes and chords based on the results file from the training data process. After getting the results of processing, the system immediately translates it again and reprocesses it to be included in the MIDI file. After processing is complete the system will produce a .mid file. The mid file is put in the out_generated_midi folder with the file name test_output.mid. After getting the output file from this process, the writer does this process twice to get the MIDI file again. The results of MIDI files that have been obtained can be heard directly or edited again in DAW.

Testing epoch: At this stage the systemtested with using epoch 10, 20, 30, 40 and 50. For the number of MIDI files at this stage the system using 5 MIDI files. Tests are the same as previous tests but at this stage the system will replaces the epoch in each test with a predetermined one. In this test the author can find out the efficiency of the system that has been made and also the number of epochs suitable for this system. Here are the results of this epoch test.

In Table 1, it can be concluded that the number of epochs affects the value of loss and also affects the running time of the system. In this test the best results

Table 1: Epoch testing results

Test.num	Midi	Epoch	Time	Loss
1	5	10	314s	3.0056
2	5	20	634s	2.1754
3	5	30	988s	1.0983
4	5	40	1332s	0.7423
5	5	50	1601s	02379

Table 2: Survey result

Number sample	Musicality (1-10)	Efficiency (1-10)	Pattern (1-10)	Average
1	8	7	8	7.67
2	8	6	8	7.33
3	8	7	8	7.67
Rata-rata total	7.56			

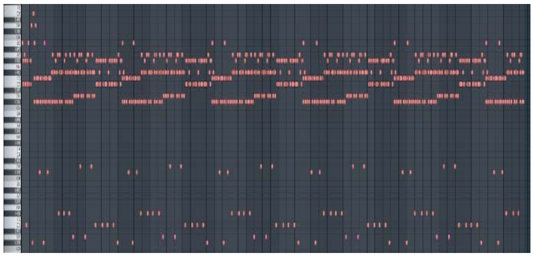

Fig. 8: Midi results 4th test

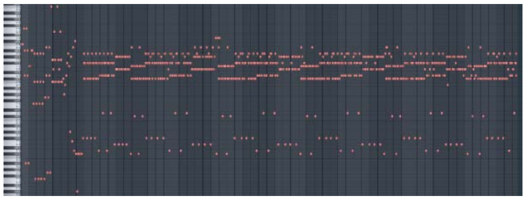

Fig. 9: Midi results 5th test

were obtained in the 4th and 5th tests which had the smallest loss value and midi results that had variations of notes and had compatibility with music (Table 2).

In Fig. 8 and 9 shows the results of MIDI from the 4 and 5th tests. The results of the 4 and 5th tests have quite varied variations of notes and can form a music.

RESULTS AND DISCUSSION

Because music is subjective, the best way to assess the results of this method is by survey from expert. the results of several songs produced can be seen in Table 2.

In Table 2 it can be seen that musicality is good because it appears rhythmic-rhythmic which is quite unusual or not commonly found in a series of tones. so it produces unique phrasing. Although, the pattern does not feel complex, it is quite varied and the tone and rhythmic movements are also quite efficient.

CONCLUSION

Based on the process that has been carried out in this project, starting from the design to the testing and analysis of the system, several things can be concluded including that recurrent neural network can be used to make music generator programs, loss is the error between the prediction (output) of the model and the target in the training data and The smaller the value of Loss, the more maximum results obtained. Experts say that for this system “musicality is good because there are many rhythms that are quite unusual or not commonly found in a series of tones, so, it produces unique phrasing. Although, the pattern does not feel complex, it is quite varied and the tone and rhythmic movements are also quite efficient”.

RECOMMENDATIONS

For further research is expected to improve the weaknesses and weaknesses that exist in this research.Some suggestions that can be given that on the next research is needed to train the recurrent neural network system, especially for research on the Music Generator can get the perfect music results with genre that user want and optimization of the system so that it can run without spending a lot of resources.

REFERENCES

01. He, T. and J. Droppo, 2016. Exploiting LSTM structure in deep neural networks for speech recognition. Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’16), March 20-25, 2016, IEEE, Shanghai, China, pp: 5445-5449.
02. Emami, A. and L. Mangu, 2007. Empirical study of neural network language models for Arabic speech recognition. Proceedings of the 2007 IEEE International Workshop on Automatic Speech Recognition & Understanding (ASRU’07), December 9-13, 2007, IEEE, Kyoto, Japan, pp: 147-152.
03. Hersch, R., 2001. Long short-term memory in recurrent neural networks. Ph.D. Thesis, Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland.
04. WF., 2019. MIDI. Wikimedia Foundation, San Francisco, California, USA. https://id.wikipedia.org/wiki/MIDI
05. Brohrer, B., 2019. How recurrent neural networks and long Short- term memory work. GitHub Inc., San Francisco, California, USA. https://elham-khanche.github.io/blog/RNNs_and_LSTM/