Music generation using LSTM (Long Short-Term Memory) network

Nevena Rokvić
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia
Email: nevrokvic@gmail.com

***Abstract*—Through the history music was considered exclusively as a human-made set of notes. With the development of different aspects of technology, this concept has been broaden and modernized, and music started to be a the object of automatic composing. To be able to generate music, there are plenty of obstacles that need to be resolved. This work offers the presentation of a music generation process using LSTM (Long Short-Term Memory), that is a type of RNN(Recurrent Neural Network), from data collection, through model construction, training and finally - music generation. Data is collected in musical instrument digital interface (MIDI) format, since it provides useful information. Trained model generates new sequences from randomly chosen starting input. Full network architecture is described in this work, including layers, parameters and activation functions.**

***Keywords*- LSTM; RNN; music generation.**

I. Introduction

Music is the term that is not easy to define, especially because it represents a complex combination of its creators idea and emotions. Some of the definitions explain music as tones ordered horizontally as notes, and vertically as harmony, other have some more practical definitions that define music as medium of communication between group members. On the other hand, many groups don't use work definition for music, since they feel it as an essential part of everyday life and the way of living[1].

Above mentioned definitions of music motivated me to explore automatic music generation, and to get a glance of the progress of this broad area. For this purpose I collected musical data, analyzed it and constructed and trained a model of a Recurrent Neural Network. Afterwards, using that model I was able to generate musical sequences.

It is challenging to tackle music generation problem, due to it's complexity and numerous dependencies. The dataset chosen for the experiment plays a significant role, since it contains only jazz tracks. They are specific since they tend to contain many improvisation segments. The big dilemma is how comparable is music generated by the machine with the music humans compose, since it has this important segment of human interaction and emotional involvement.

In further text I present dataset with more details. Next chapter contains some related works and summary of their experiments. Chapter three explains the methodology that is used in this work, including the dataset and the neural network. Results are presented in chapter four with graphic representation. Summary of the experiment is in chapter five, including some ideas for future enhancements.

II. Related work

In the field of music generation there are numerous investigations. In the past few years, growing popularity of deep learning enhanced the possibilities of developing more stable and precise generative models and better understanding of how to come to those results.

Many approaches have been used in order to tackle music generation and all the supporting issues that this task brings. One of them is used by investigators from Medi-Caps University's Computer Science and Engineering department. They developed a model that is used to learn the sequences of polyphonic musical notes over a single-layered LSTM (Long Short-Term Memory)[2] network. Their goal was to generate music using the model that can learn, analyze and generate completely new set of notes. Their LSTM network architecture consists of a single LSTM layer with 512 neurons, dropout layer, used to prevent overfitting, another dense (fully connected layer), activation layer and finally the optimizer, where they used RMSprop optimizer. Finally, after 200 epochs the accuracy this model has goes up to 0.97[3].

Another paper that deals with automatic music generation using LSTM tackles the problem of long training duration and shows some effective data preprocessing techniques that contribute to generating more syntactically correct sheet music. They accomplish that by removing unessential information leaving the model with the focus to learn only the textual musical notes. Taking that into account, for the same number of epochs model can generate better results. To evaluate the quality of generated music they used subjective method that involves a survey among peers that have different musical taste. They came to conclusion that generated and human composed have a comparable quality. [4].

Researchers from Lennon Lab HIFIVE Tech Co propose an approach that involves musical sequence generation using bidirectional LSTM network. They chose this approach since this network has the ability to efficiently explore the complex relationship between notes. The main contributions of this paper can be find in a novel sampling strategy, improved loss function and a simplified representation of input data [5].

III. Methodology

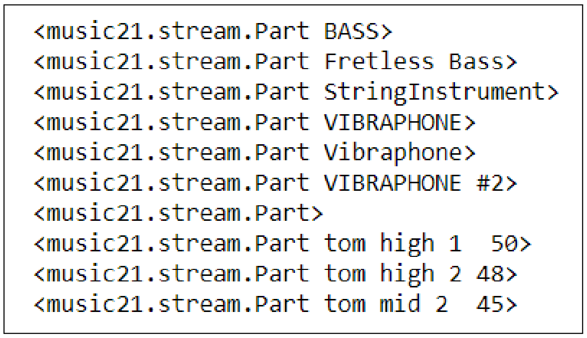
Methodology used in this work is explained in this section. I have collected different jazz tracks in midi format, built and trained LSTM network, a type of Recurrent Neural Network, and finally, generated new set of notes using the trained model. The whole process was conducted using different Python libraries for data manipulation and deep learning.

A. Dataset

The dataset used in this work contains around 900 songs scraped from website that offers a variety of midi songs grouped by different parameters such as artist, genre, etc. Target genre was jazz, because of it's improvisation nature[6].

Data collection was performed using a library for web scraping [7]. Next step was to load midi files using music21 library[8]. This library has a broad usage field. Among many functionalities it can be used to analyze musical datasets, to understand musical theory and to compose music. The toolkit exists since 2008., and it is continuously developing. Since midi files contain information such as note's notation, pitch and velocity that exist in a track, this library can easily access to that information, as well as to the information about the instruments, that can be seen on Fig.1.

Once all the tracks were loaded as a list of instruments,chords and notes, I divided them according to the instrument. I chose to explore piano, bass, trumpet and drums as one of the most common jazz instruments. The biggest variation was held in piano section as expected, followed by bass, trumpet and lastly drums. With further exploration I discovered the number of unique notes (around 1200) and their frequency, as it can be seen on Fig.2. This distribution shows that most of the notes have low frequency.



```
<music21.stream.Part BASS>
<music21.stream.Part Fretless Bass>
<music21.stream.Part StringInstrument>
<music21.stream.Part VIBRAPHONE>
<music21.stream.Part Vibraphone>
<music21.stream.Part VIBRAPHONE #2>
<music21.stream.Part>
<music21.stream.Part tom high 1  50>
<music21.stream.Part tom high 2 48>
<music21.stream.Part tom mid 2  45>
```

Fig. 1. Partition by instrument using music21 library

B. Training

Music generation can be compared to a text generation, where in music, instead of next character, we predict next note/chord.

- Long Short-Term Memory(LSTM)

LSTM is a type of Recurrent Neural Network(RNN) that emerged as a solution to several learning problems that a simple RNN possesses. It has a broad application in many fields such as generation, speech recognition, speech synthesis, etc. The main idea supporting this architecture lies in a memory cell that has the ability to persist its state over time, and gating units that control the input and output flow of the cell.[2]

The collected data is used to feed the network with sequences, so it can learn to generate new sequence on its own. It is necessary to put the data in format so it can be fed to the network. Input data is formatted and encoded that way it can be easily decoded at the output. String representation of notes is mapped to integers, since network works better with integers. For an input sequence, output will be the chord or note that comes right after that sequence. Input sequence number was changed various times, but set to 100 in the final training. Final step was to do one-hot encoding and data normalization.

Several network architectures were used in this project, with different layer variation. The best result was obtained with the architecture from Fig.3. This architecture contains three LSTM layers with 256 neurons each, followed by batch normalization layer, dense layer, 'relu' activation function[9], another batch normalization layer, dense layer, and finally softmax activation function layer[10]. Training is done using Keras library[11] in Python, and number of epochs was set to 100. After 41 epochs and around 4 hours of training, the loss function started to increase, and it is where the training is stopped. This was taken as the best model, and used to generate sequence. Other model architectures that were used for training involved smaller number of neurons, and different optimizers, but still were giving worse results in 100 epochs of training.

IV. Results

Being the part that contains most of the information in notes and chords, piano part of tracks was used for training and generation.

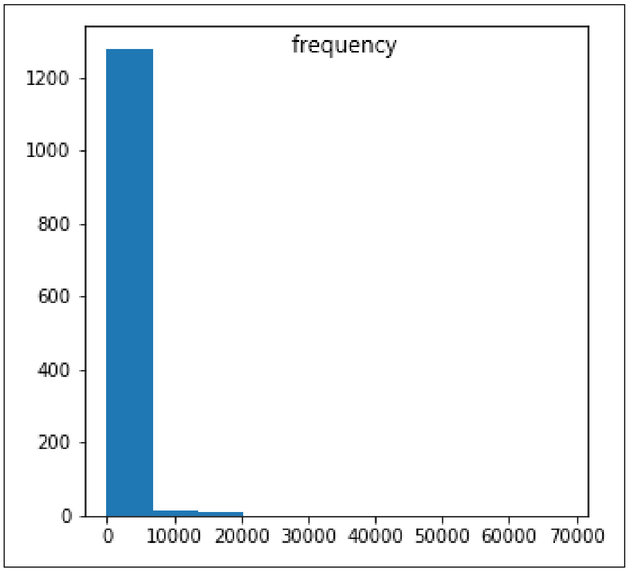
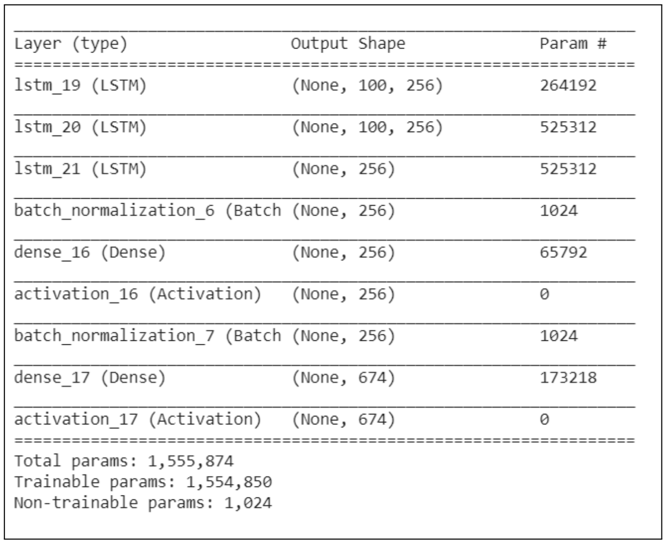


Fig. 2. Unique notes frequency



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_19 (LSTM)               (None, 100, 256)          264192
_________________________________________________________________
lstm_20 (LSTM)               (None, 100, 256)          525312
_________________________________________________________________
lstm_21 (LSTM)               (None, 256)               525312
_________________________________________________________________
batch_normalization_6 (Batch (None, 256)               1024
_________________________________________________________________
dense_16 (Dense)             (None, 256)               65792
_________________________________________________________________
activation_16 (Activation)   (None, 256)               0
_________________________________________________________________
batch_normalization_7 (Batch (None, 256)               1024
_________________________________________________________________
dense_17 (Dense)             (None, 674)               173218
_________________________________________________________________
activation_17 (Activation)   (None, 674)               0
=================================================================
Total params: 1,555,874
Trainable params: 1,554,850
Non-trainable params: 1,024
```

Fig. 3. Network architecture

To start generating music, random sequence is chosen from the input to be the beginning point. After that, next step would be preparing sequences for prediction, and after the output is ready, to convert it back to midi format (to decode it).

Every integer is converted to corresponding chord and note. The offset(the length of time from the start of the piece) is increased in every iteration to prevent notes to get glued together. The model is able to generate a track that contains different notes, and it can be used as an inspiration for human composers or as a bridge between some already composed notes, as it is able to offer different combinations of notes and chords. With further parameter tuning and playing with the number of notes in a sequence used for prediction model can get refined and can predict theoretically more correct notes in jazz domain, where that aspect was not the focus of this work.

Notes with high frequency are dominating in a predicted sequence as well, as it is shown on Fig.4. There are three generated tracks, all marked with different colors. The one thing in common is the convergence to one chord/note when coming closer to the end of the sequence (Fig. 5), which can probably be improved with bigger dataset and additional data preprocessing.

The fact that the result is not giving a big variety of notes can be blamed also on the dataset, and the genre that was chosen for this experiment.

V. Conclusion and Future Work

With the evolution of technology, growing popularity of automatic music generation doesn't come as a surprise. Nowadays there are plenty of powerful AI-based tools for music generation that are built using different deep learning techniques. In this work I tried to explore how LSTM networks can be used for this task, to scrape the surface of this broad field and get the sense of how it can be beneficial for the process of composing.

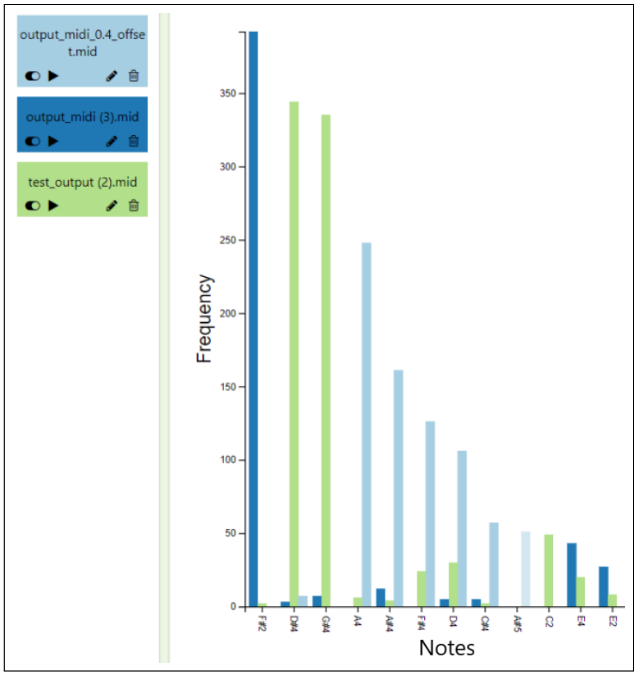


Fig. 4. Frequency of generated notes

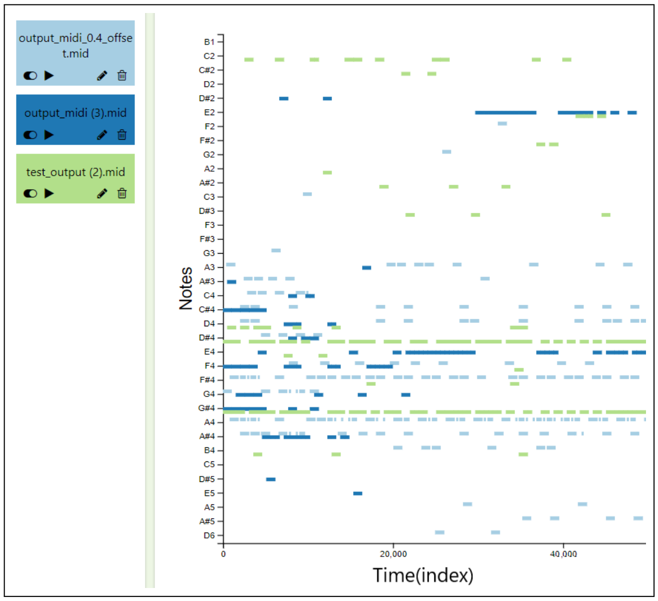


Fig. 5. Visualization of generated notes

When it comes to dataset analysis, I discovered that the most dominant instruments in a jazz dataset I collected include piano, bass and trumpet, and that there is a big improvisation factor that can affect the process of learning.

In terms of prediction, I discovered that experimenting with the number of sequences used for prediction of the next sequence can vastly impact the quality of prediction. Also, the offset makes a big difference in the final result, as it changes the dynamics of the track and when the note will be triggered.

As a potential future task more data can be collected in order to obtain more diverse notes. It would be interesting to explore how to build a network, or a combination of networks that can generate music for other instruments as well, since in this work I generate only piano music. Some other network architecture can also be explored in order to get better results.

Most of the experiments in this area have the same concern: how to generate music that can get closer to human composed one. There are many factors that this depends on, and a big one for sure is the lack of emotions that machines have, that is a common trigger for some kind of inspiration in humans. Thanks to numerous ongoing experiments and popularity of deep learning that is continuously rising, there is no doubt that the results in this field will keep improving and getting closer to its goal.

REFERENCES

[1] F. Furukawa. Fundamentals of music properties of sound – music theory lesson, 2015. Accessed: 2021-05-02.

[2] Klaus Greff, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017.

[3] Sanidhya Mangal, Rahul Modak, and Poorva Joshi. LSTM based music generation system. *CoRR*, abs/1908.01080, 2019.

[4] Sarthak Agarwal, Vaibhav Saxena, Vaibhav Singal, and Swati Aggarwal. Lstm based music generation with dataset preprocessing and reconstruction techniques. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 455–462, 2018.

[5] Tianyu Jiang, Qinyin Xiao, and Xueyuan Yin. Music generation using bidirectional recurrent network. In *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pages 564–569, 2019.

[6] Free midi - best free high quality midi site. http://https://freemidi.org/. Accessed: 2021-05-02.

[7] Leonard Richardson. Beautiful soup documentation. *April*, 2007.

[8] Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. 2010.

[9] Abien Fred Agarap. Deep learning using rectified linear units (relu). 03 2018.

[10] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.

[11] Francois Chollet et al. Keras, 2015.