Term Project Report
GamelanAI
Gamelan Music Generator using Recurrent Neural Network

Fawwaz Dzaky Zakiyal (201899213)
Tanjung Dion (201883621)

***Abstract*—Music generation is one of interested topic areas in deep learning topics. In this project, we generates Indonesian traditional instrument called Gamelan using Recurrent Neural Network (RNN). We also compare the performance of two different types of RNN for generating Gamelan, namely, Long Short-Term Memory (LSTM) network and Gated Recurrent Unit(GRU). Our work is implemented using music21 for processing MIDI (Musical Instrument DIgital Interface) and Keras which is built on the top of Tensorflow framework for training and generating music composition. Our experiment shows that GRU give lower loss than LSTM on all instrument's models, because GRU have fewer parameters and thus may train a bit faster or need less data to generalize.**

***Index Terms*—Music generation, RNN, LSTM, GRU, Gamelan**

I. Introduction

In the recent years, neural network have been used in many aspect in our live. They have not only proven to be worked on image and text recognition, but have also shown a good result in music generation. Music is a story telling that is consist of melody and cord. Generating good combination of melody and cord to be harmonized using simple computation process are not enough. Many researcher have proposed neural based approaches to overcome this problem. Chu et al [1], proposed a model called hierarchical Recurrent Neural Networks to generate pop music, where the melody, chords, and other instrument can generate a song.

Inspired by this research that was able to generate new music, we decide to implement deep learning approach to generate Indonesian traditional music which use instruments called Gamelan. Gamelan is one of the musical instrument from Indonesia that consist of several instruments such as gongs, gong chime, metallophone, and drum [2]. The illustration of Gamelan instruments can be seen in Figure 1

Although the popularity of Gamelan has declined since the introduction of pop music, Gamelan is still commonly played on formal occasions and in many traditional Indonesian ceremonies. For most Indonesians, Gamelan is an integral part of Indonesian culture. Therefore, in this project we would like to create GamelanAI, a Gamelan Music Generator using Recurrent Neural Network. This project aims to generate gamelan music, where melodi, chords and other instruments played together in harmony. Contributions of this project as follows:

- Present the implementation of recurrent neural network for generating gamelan music
- Compare the performance of LSTM and GRU network to the task of music sequence modeling
- Use multiple instrument to generate Gamelan music that can be played in harmonic

In the section 2, we are talking about several related works that have been succeeded in generating music using recurrent neural network. Section 3 explain about the differences between LSTM and GRU as network model in our training process. We also explain music theory overview and MIDI format information. Section 4 is methodology where we explain about system design that cover all process in our implementation. After the implementation, we present some experiment result related to the comparison between LSTM and GRU in the term of loss function, with respect to the depth, sequence number, and unit size. We also present our conclusions in the section 5.

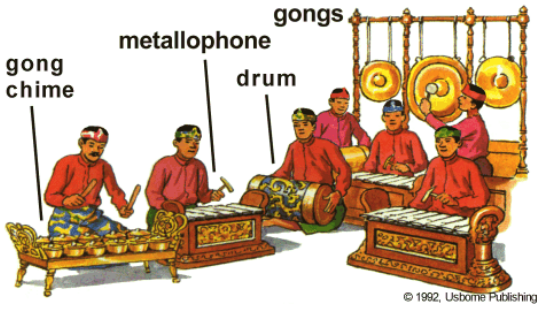

Fig. 1: Gamelan Instruments

II. Related Work

Recurrent neural network have been successfully implemented in many research for generating music. In 2002, [3] uses LSTM network to learn the long term structure of the Blues songs using binary vectors to model notes appearing in a fixed sequence. Author in [4] try to learn relationship within MIDI files that represents chords and notes of the music

using LSTM network. The implemented network also capable to generate automated music. GRU have been used by [5] to model polyphonic music and speech signal using MIDI dataset. The result shows that GRU is comparable to LSTM and both of them is better than traditional recurrent units such as tanh units.

All of these works listed above have applied recurrent network such as LSTM and GRU to the task of music sequence modeling. They use chords and notes in music sequence to generate automated music. The result also show that the generated music has similarity with the inputted dataset. However, they only modeling one kind of musical instrument especially for piano. There are many instrument employed together in a composition that produce beautiful sound. For example, orchestra is one of musical instrument that consist several sound produced from brass, stringed, and percussion instrument. In this sense, we extract all instrument in Gamelan music and use top-5 instruments to generate new Gamelan. We also make a harmonic Gamelan music by considering tempo and duration for each notes.

III. Concepts

A. Recurrent Neural Network

In this project, we use Long Short-Term Memory network (LSTM) and Gated Recurrent Unit (GRU) type of Recurrent Neural Network (RNN) model. Both LSTM and GRU have ability to keep memory/state from previous activation more longer rather than vanilla RNN. GRU add update gate to decide whether to pass previous output to the next cell. On the other hand, LSTMs uses 2 more gates called forget and output in addition of update gate. Therefore, LSTMs introducing more controlling knobs for better control ability. However, it comes with more complexity and operating cost. The illustration of LSTM and GRU can be seen in Figure 2. In this project, we will not only generating new music but also comparing the performance of LSTM and GRU by evaluating on the training loss.

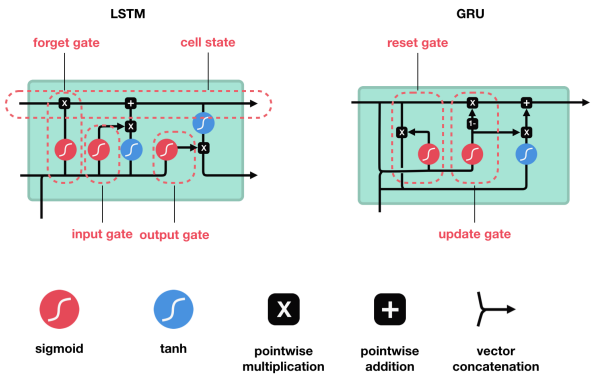

Fig. 2: LSTM vs GRU

B. Music theory overview

We start by introducing the basic notations and definitions of music theory. Music element consist of note, bar, scale, and chord. A note is the pitch and duration of a sound that be represented as musical notation. Typically, pitch classes are represented as seven letters of the Latin alphabet A, B, C, D, E, F, and G where each latter names are modified by accidentals. A bar is a boundaries that correspond to a specific number of beats (notes). Scale is an organized sequence of notes in Major or Minor scale. Chord is a group of notes that are played in harmonic. Music can be represented as sheet music and ABC notation. In sheet music, music is represent by a sequence of musical notes where each note is separated by a space. Compared to sheet music, ABC notation represent a sequence of character where each character represent some musical node.

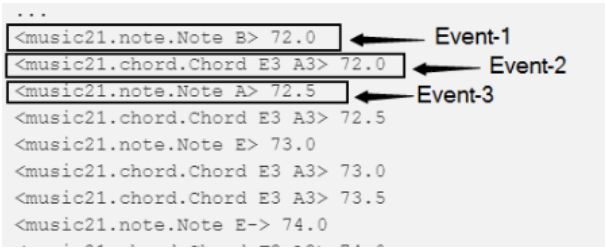

Fig. 3: MIDI format

Musical Instrument Digital Interface (MIDI) is a technical standard that describes communication protocol that interpret message like 'note on', 'note off', 'note/pitch', 'chord', and many more. The message interpreted by MIDI to produce sounds. Figure 3 shows MIDI in ABC representation generated using Music21, a python library for generates MIDI format. From the figure, Event-1 and Event-2 show information about note B and A, and Event-3 represents chord E3 A3 and so on.

IV. Methodology

A. System Design

The process consists of pre-processing of MIDI dataset (MIDI to notes conversion, notes to sequences conversion & sequences reshaping). After that, split the dataset into training & validation data, next we train & evaluate the model. Finally, we use the model to generate notes and we put it back to MIDI format which we can listen. The system design can be seen in Figure 4.

B. Data

The data used for our input is the Gamelan musics that come from a MIDI Gending Manten collection site [6]. This dataset consist of 26 MIDI file format, where the file size is between 8 - 163 KB and have duration between 48 - 652 seconds. This dataset contains 1350-35814 notes that are played in different instrument such as Celesta, Marimba, Bass, Flute, Sitar and many more. We use gamelan music because its produce beautiful musical instrument where each instrument is played in different style.

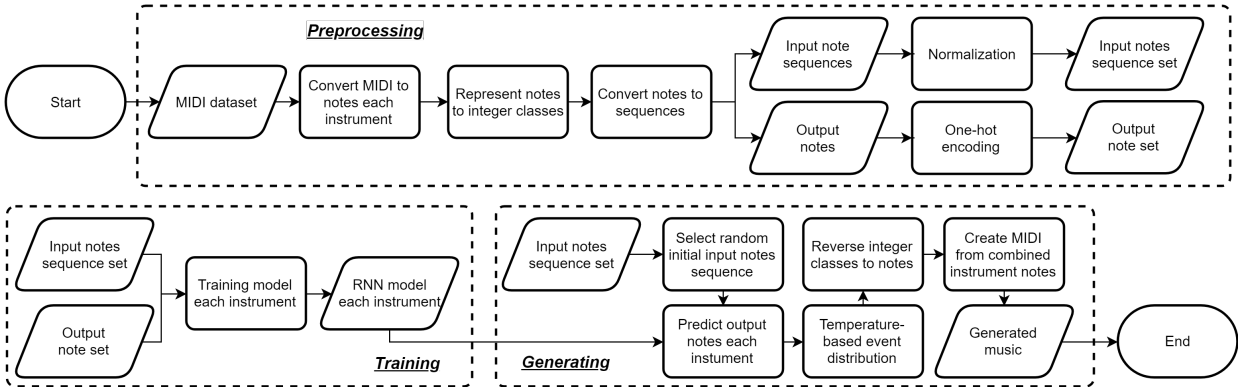

Fig. 4: System Design

C. Prepossessing

In the beginning of prepossessing, we load the dataset into our system then convert it from MIDI file format to ABC notation using Music21 library. For each notation, we make partition based on its instruments. After conducted several experiment, we found out that each gamelan has different used instruments that could generate unharmonic sound. Therefore, we decided to use only 5 instruments that are mostly used in Gamelan. For each instrument, we extract the information of notes, chords, and rest. We also extract the information of note notation, note duration, and note offset from notes and chords. This 3 information is our baseline to generate harmonic Gamelan instruments. In this case, we append the pitch of every note object using its string notation since the most important part to generate different sound. For chord notation, we append every chord by encoding the id of every note in the cord together into single string that are separated by a comma. We also insert the note duration and offset in each note and chord string, separated by a vertical line character. These encoding allows us to easily decode the output generated by the network into the correct notes, chords, duration, and offset. In the end of MIDI extraction, we have a sequential list [”Notes|Duration|Offset”,”chordId.ChordId|Duration|Offset”].

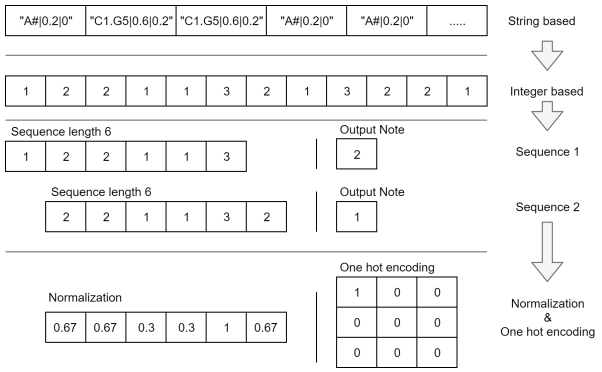

Fig. 5: The process of mapping list from text to integer, build sequence then do normalization, and generate output set for one spot encoding

Figure 5 shows the pre-processing step from generating a list of notes/chords, mapping the list from text to integer, then build input note sequence set and output notes set. In this process, we create a mapping function to map string based categorical data to integer based categorical data. This mapping process is implemented because neural network has better performance when learning integer based than string based. Next, we generate sequence set and output notes set based on the result of the mapping process. Then, we generate a sequences set for input and output note set for predicting. In example from Figure 5, it create several sequences with the length of 6. First sequence is generated by append item from index 0 to 5 in the list, then use item in the next index for output note. The next sequence has same process by increment the depth for example depth of 1. For each sequence, we do normalization to make train class in the range of 0 to 1 by dividing the integer value with the maximum value. We also built one hot encoding based on output note. Both, normalization and one spot encoding is important to help network learning easily.

D. Training

In this section, we explain about our model architectures, it consist of:

- Input layer: it become the layer that obtain input note sequences set
- RNN layer (LSTM or GRU): as the main layer to learn the sequences, it also applied weight decay as regularizer.
- Batch Normalization Dropout layer: it used as regularizer to increase model preformance.
- Dense layers with Softmax activation function: it used to output the prediction.

E. Generating

After the training process is finished, all of the trained data will be saved as files for future use. These files containing the weight that are saved during the training section. When we want to generating new musical instrument, we load these files without doing training process from the scratch.

Since we have full list of a sequences notes from the training section, the system will pick random index in the list as starting point. Then, we have to create a mapping function

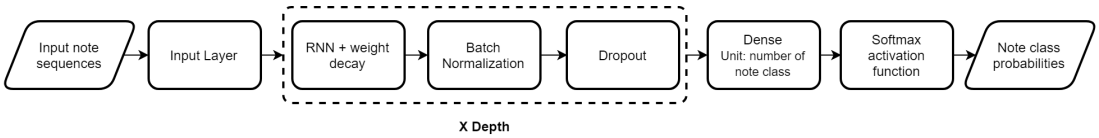

Fig. 6: Model architecture

to change integer based back to string based from the pre-processing process. We also decode the string based value to musical notes, duration and offset. If string is indicated as chord (has several note and separated by dot character), we have to separate each note than append it based on the same offset location. This method will make several note will played together for creating a harmonic chords. In this process we decided to generate 500 notes for 2 minutes of music. For each notes, we append the location based on the offset. Moreover, duration is also added to note for indicator how long this note sound. For every sub-sequence that used as input, the first note in the sequence will be deleted. Then, we have to insert the output note from the previous sequence.

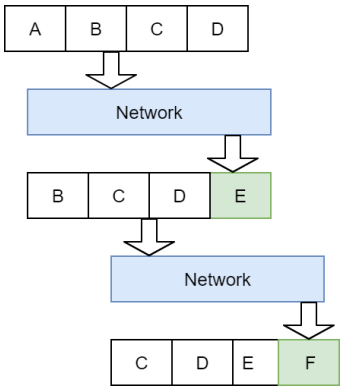

Fig. 7: Sequence modeling

Figure 7 shows the sequence modeling. We also have to generate the next note by predicting the highest value from the network output. Figure 8 shows how our method to selecting note. In this figure, note C has the highest probability. Therefore, we choose note C as the next music element. We could also control the output of the model by selecting

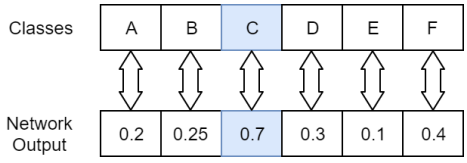

Fig. 8: Prediction based on the highest probability for selecting note in the next sequence

value for a parameter referred to as temperature that affect the randomness of the sample. If we set temperature to 1.0, our model predicted event distribution. Decreasing temperature will make music generation to be more repetitive. On the other hand, higher temperature will produce more variation in the note selection. After all instrument is created, we will combine them as one Gamelan instruments. Lastly, we decode this musical note into midi format using Music21 library.

V. Experiments

To evaluate the proposed method, this project conduct experiment using Keras and Music21 on Python programming language. The experiments were carried out on a Intel i5-7500 PC with 4 GB RAM, running Windows 10 64-bit OS, and utilizing GPU NVidia GForce GTX 1060 3GB.

In term of building the RNN model, this project define some default experiment parameter. It trained on 50 epoch and 20 batch size. It using SGD momentum optimizer with learning rate 0.01. It applied weight decay on the RNN in the number of 0.0001, and it followed by batch normalization and drouput layer with the probability of 0.2. Lastly, this model use Cross Entropy as the loss function.

A. Evaluation on Depth of RNN Layer

In the early scenario, we applied 100 note sequences length for the pre-processing. The result return that Celesta contains 6124 sequences with 1972 unique notes, Bass contains 5288 sequences with 1592 unique notes, Marimba contains 5170 sequences and 1063 unique notes, Flute contains 1153 sequences with 701 unique notes, and Sitar contains 1194 with 184 unique notes. After, receiving the pre-processing result. We applied fixed parameter of RNN unit size with 256, next we evaluate the impact of RNN layer depth to the model by the number of 1, 2, and 3 times. The result can be seen on Figure 9. It seems the Bass, Marimba, Flute model get greater loss by increasing depth. The Celesta and Sitar model get lowest loss in 2 RNN layer depth. Overall, it shows that GRU give lower loss than LSTM on all instrument's models, because GRU have fewer parameters and thus may train a bit faster or need less data to generalize [7].

B. Evaluation on Size of RNN Unit

In this scenario, we use early pre-processing result. Next, we evaluate the impact of RNN unit size to the model by the number of 256, 512, and 1024. The result can be seen on Figure 10. It seems that 1024 RNN unit size lead to highest loss in all instrument's model. All model except Sitar get lowest loss by using 512 RNN unit size. Overall, it shows that GRU give lower loss than LSTM on all instrument's models, because GRU have fewer parameters and thus may train a bit faster or need less data to generalize [7].

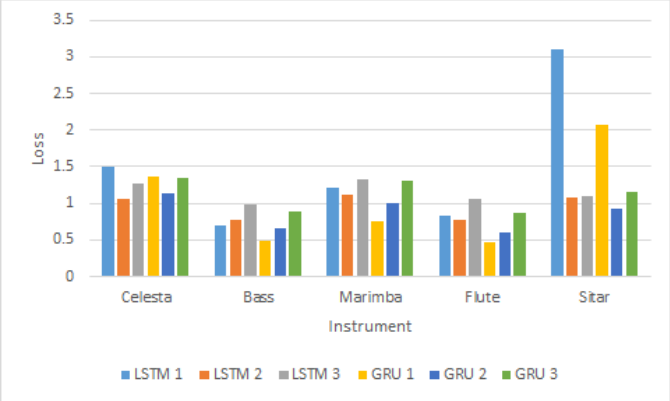

Fig. 9: Evaluation on Depth of RNN Layer

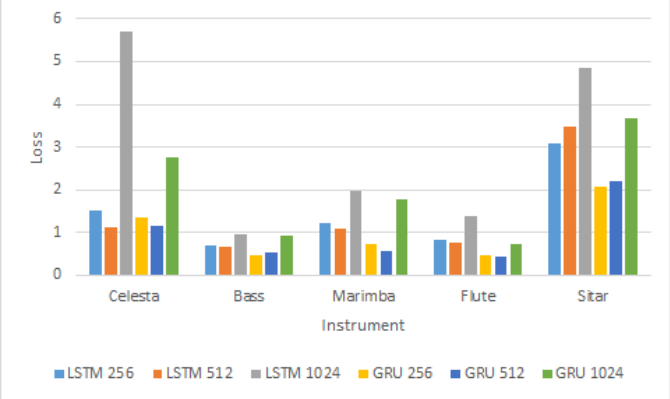

Fig. 10: Evaluation on Size of RNN Unit

C. Evaluation on Length of Sequences

In this scenario, we applied early RNN unit size (256). Next, we evaluate the impact of sequence length to the model by the number of 100, 200, and 300. The result can be seen on Figure 11. It seems that increasing sequence lead to lower loss in all instrument's model. Overall, it shows that GRU give lower loss than LSTM on all instrument's models, because GRU have fewer parameters and thus may train a bit faster or need less data to generalize [7].

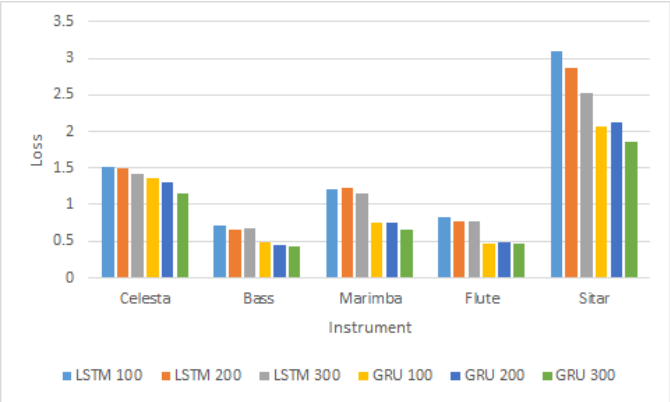

Fig. 11: Evaluation on Length of Sequences

VI. Conclusion

This project succeeds in building model to generate Gamelan music that contains multiple instrument. From the experiments, it obtained that 2 RNN layer depth, 512 RNN unit size, and 300 sequence length give best result than the other parameters. Overall, it shows that GRU give lower loss than LSTM on all instrument's models, because GRU have fewer parameters and thus may train a bit faster or need less data to generalize.

Future work that could be conducted are further explore in tuning the other parameters, try to considering the other MIDI's properties like tempo, try to learning all of the instruments in one model that produce more harmonic music.

VII. Project Statements

Contributions from Fawwaz Dzaky Zakiyal:

- Code the preprocessing method.
- Build Long Short-Term Memory (LSTM) type of Recurent Neural Network model.

Contributions from Tanjung Dion:

- Code the generating method.
- Build Gated Recurrent Units (GRU) type of Recurent Neural Network model.

References

[1] H. Chu, R. Urtasun, and S. Fidler, "Song from pi: A musically plausible network for pop music generation," 11 2016.

[2] . En.Wikipedia.Org, *Gamelan*, 2019 (accessed June 3, 2019). [Online]. Available: https://en.wikipedia.org/wiki/Gamelan

[3] D. Eck and J. Schmidhuber, "Learning the long-term structure of the blues," in *ICANN*, 2002.

[4] P. Agrawal, S. Banga, N. Pathak, S. Goel, and S. Kaushik, "Automated music composition using lstm," 01 2018.

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 12 2014.

[6] . Music Style, *Kumpulan Midi Gending Manten*, 2019 (accessed May 3, 2019). [Online]. Available: https://musicstyle29.blogspot.com/2019/01/kumpulan-midi-gending-manten.html

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.