Deep Music Generation

Tulane University
SCHOOL OF
SCIENCE AND ENGINEERING

Chanho Lim – Mathematics and Computer Science
Roland Nguyen – Mathematics and Computer Science
Dr. Jihun Hamm – Department of Computer Science

Introduction

Deep Music Generation is a deep learning project with two sides. Firstly, to generate polyphonic music using deep learning models that replicate the style of composers such as Bach and Handel with variety.

Secondly, to create metrics that evaluate the quality of generated samples and the performance of the model.

Problem statement

Deep Music Generation has two main problems – generating with deep learning models and evaluating the generated samples and performance of the model.

Various deep learning models do exist for generating music, but the quality of the generated are questionable. While the generated samples did have chord structures, the flow of the pieces was weak. Hence, we modified a famous deep generative model called MuseGAN to enhance the quality of generated samples.

Various evaluation metrics do exist for deep image generation, but there is a lack of robust metric for deep music generation. While some samples did resemble structured music, we wanted to take a scientific approach in our judgement. Hence, we created a metric that evaluates the quality of generated music samples and performance of the model.

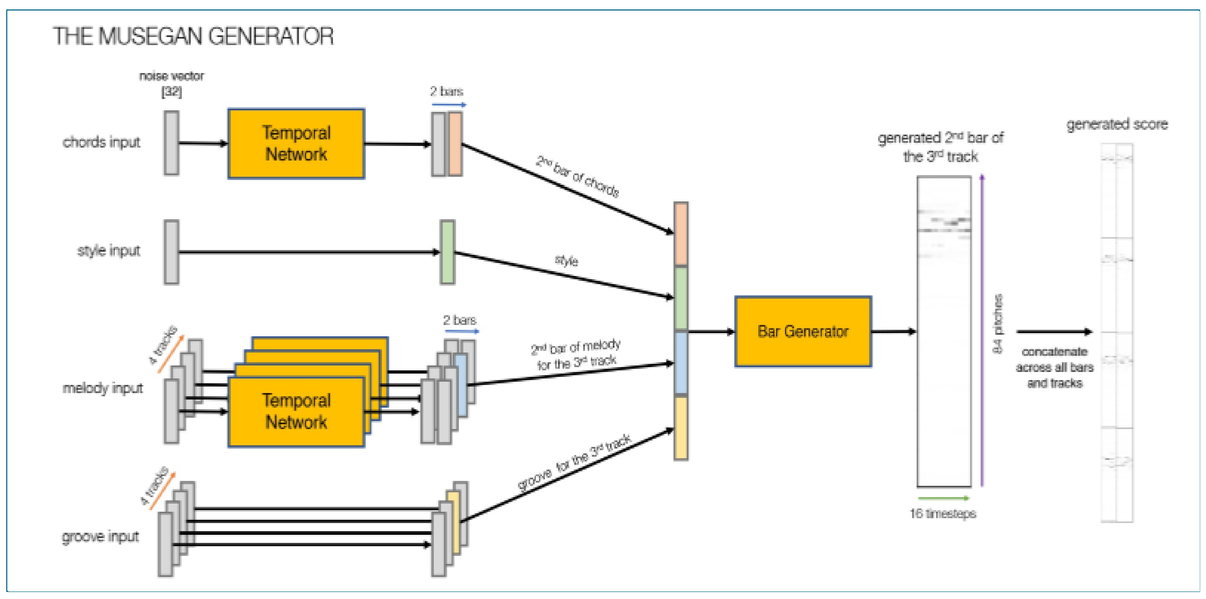

Fig. 1 High-level diagram of the MuseGAN generator, Hao Wen Dong and David Foster

Methods/approach

1. Generation
   - I. Constructed classical music archive of MIDI files.
   - II. Preprocessed Music (MIDI) to tensors where the dimensions of the tensors are represented by number of tracks, number of timesteps in a beat, and number of bars we are generating.
   - III. Instead of the original author's approach to define the resolution of music by the number of timesteps in a beat where the size of each timestep was dependent on the ratio of timesteps per beat, we fixed size of timesteps and format of each bar to enhance the musical flow by "tricking" the generator.
2. Evaluation
   - I. Trained classifiers with real music pieces from classical composers by taking n-grams of chord sequences then using tf-idf vectorizer to assess whether the generated samples replicates the style of the original composer.
   - II. Designed a LSTM-RNN classifier based on the duration of notes and the distribution of notes using n-hot encoding as an enhanced approach to tackle multi-composer classifications.
   - III. Aimed to create a music-version of Inception Score, a scoring metric based on KL-Divergence to evaluate the quality of the generated samples and the performance of the model.

Results

First, we built our classical music archive of MIDI files. We scoured various sources and compiled over 200 composers, each with a varying number of musical pieces per composer.

Second, to preprocess the classical music pieces in the form of MIDI files, we built a converter and a slicer. This music converter filters out MIDI files of invalid types then breaks down the timesteps of the pieces into arrays of pitch values for the tensor. We then built a method to automate the slicing of each MIDI file into smaller sequences based on the number of bars.

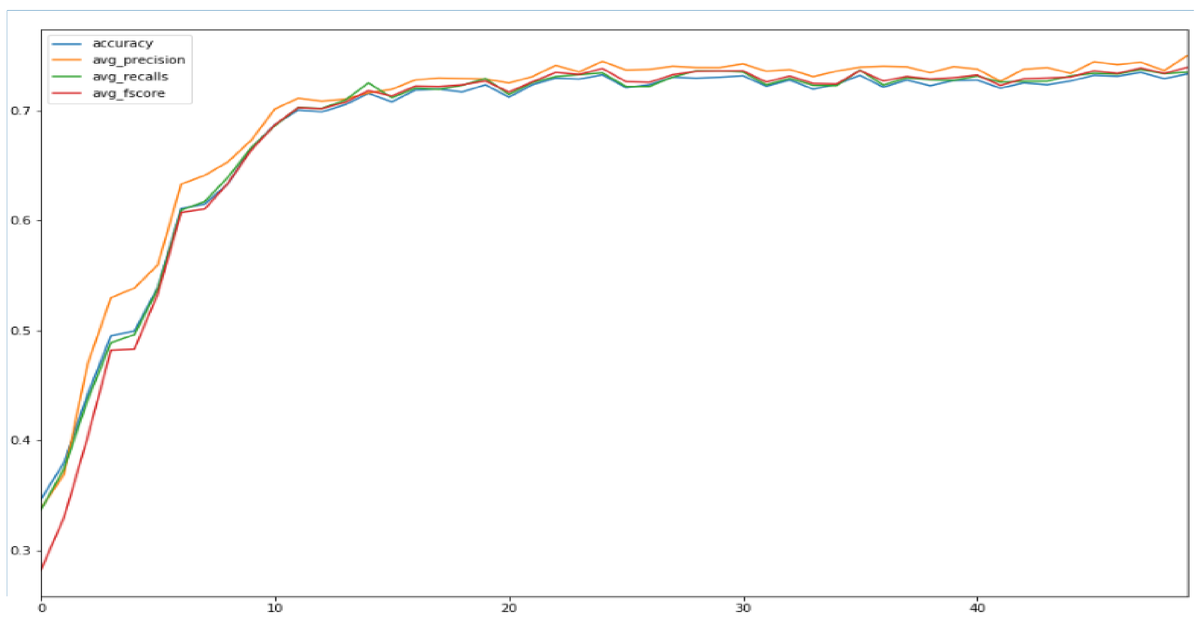

Fig. 2 Training measures of LSTM-RNN model with 14 composers over 50 epochs

Third, our approach of fixing the size of timesteps and the format of each bar allowed us to "trick" the generator and take the input as 2 bars with 128 timesteps each instead of 16 bars with 16 timesteps each. This allowed for better bar-to-bar transition, hence, improving the quality of the generated samples.

Fourth, while achieving 0.76 test accuracy in binary classification of Handel vs Bach pieces, our traditional classifiers with n-grams of chord sequences performed poorly with multi-composer classification. Also, this approach does not measure the quality of each sample and the performance of the model which leads to our final work.

For our final work, we built an LSTM-RNN based classifier that takes key signal transpose and the number of occurrences for the notes to serve as a MIDI-classifier. We achieved 0.803 test accuracy with 8 composers.

Lastly, using the concept of Inception Score, we have been working to adapt it towards the domain of music, in order to examine the quality of the pieces as well generative model. With the 8 composers, we achieved a score of 5.8 (out of 8).

Conclusions/Discussion/Future work

While we have been working towards adapting our Inception Score metric to evaluate our deep music generation, therein lies issue in the fact that our LSTM-RNN classifier needs improved results, test accuracy wise, for our metric to grasp a better idea of the performance of MuseGAN. Thus, we intend to continue working to improve the classifier model.

As seen in figure 2, the MuseGAN model trained with 14 composers achieves results that did not perform as well as that with 8 composers. There does lie the issue of insufficient training pieces from some composers explaining why the 14 composers had such an accuracy. So we will continue to build our classical music archive.

Following our results from our Inception Score metric, a method in which we would be to further validate our metric would be finding the correlation between human reaction and the scores. We can achieve this by conducting a crowdsourced survey of the generated pieces.

References

Dong, Hao-Wen. "MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment." ArXiv.org, 24 Nov. 2017, arxiv.org/abs/1709.06298.

Foster, David. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play. O'Reilly, 2019.

Salimans, Tim, and Ian Goodfellow. "Improved Techniques for Training GANs." ArXiv.org, 10 June 2016, arxiv.org/abs/1606.03498.

Acknowledgements in free form for administrative and financial help for the project, e.g. "The authors thank Academy of Motion Pictures for giving them the Oskar award. Funding provided by NSF grant #314159265".