Generating Piano Music with Recurrent Neural Network (RNN)

LOW, Zhi Hao, 54924670
PAK, KA YEE, 55692027

Abstract

The proposal proposes how people can create their piano music without actually having a music background or knowing many music theories by applying RNN and GAN. We will first reformat the piano music from MIDI format to a two-dimensional matrix, then create a similar model to Magenta by applying RNN. Moreover, compare the first method with the second method, GAN, the second method learns how the music melody distribution so later can generate new music with realistic musical waveforms. The general public can easily create realistic and pleasant piano music is the ultimate goal of this research proposal.

Objectives

1. Compose realistic and enjoyable piano music.
2. Realtime adjustment of the music tunes, rhythm and loudness.

Background

Each composer has their personalised style of making music, and these styles are not kinds of knowledge that can be easily transferred between people. This music's inherent pattern needs to be studied thoroughly, and not a single methodology can be applied to all kinds of music styles or artists.

Motivation

Realtime music generation is costly to work to be done by hand. The tuning of music to a different situation can be applied to gaming, physical stores and e-commerce. Music is a big part of most experiences, and it allows the customer in the mall to feel calm, which in turn increases the consumer's expenditure. Besides, gaming the simulation of the real-life situation is very dynamic, down-to-earth sound generations will definitely open a new dimension for incredible experiences.

Therefore the motivation of applying artificial intelligence methodologies to this problem is not a far fetched idea, it was tested continuously, and the music was enjoyable on multiple occasions. However, for most of the results, this music does not sound natural enough. Even more so, the model takes forever to train. In this project, we hope to simulate piano music with instrumental-only kinds of music.

Dataset

The dataset we will employ for training the GAN is from Kaggle. The music's data format is in the form of MIDI (Musical Instrument Digital Interface), which is suitable for instrumental music. MIDI data have two dimensions (Pitch versus Time).

Proposed Methods

First, music in MIDI format is turned into a 2D matrix, and we need to find an appropriate segmentation from the original length. Usually, a 96 ticks time signature has the best result because, in music, rhythm is usually in the form of multiple of 2 and 3 up to ticks 16 (excluding ticks 10). Repetition across different measures are collected into segments, and we will take 16 segments from the original music as the training input.

We will propose two methods for the model and contrast the two models and choose the best one. The first is the RNN, and we will try out the library Magenta that generates music. Next, we will create a model that is similar to Magenta that allows us to adjust the training, even more, to make the generator(Generative Adversarial Network, GAN) in a form similar to Magenta.

The second is the Generative Adversarial Network (GAN). The GAN consist of two agents, the generator and the discriminator. The music will be generated by the generator, the generator takes in noise input and generates a candidate 16 segment MIDI format same as the dataset, this candidate is passed into the discriminator to decide if this candidate is in the form of the MIDI format. For the discriminator, besides detecting if the output of the generator follows a MIDI intended format, the discriminator needs to identify if the styles of the music resemble a melody that makes the user that listens comfortably.

Finally, we will use statistical measures to determine if the resulting music follows the pitch distribution of typical piano music.

References

CodeParade. (2018, July). *Generating Songs With Neural Networks (Neural Composer).* Retrieved February 24, 2021, from https://www.youtube.com/watch?v=UWxfnNXlVy8

Generative adversarial network. (2021, February 12). Retrieved February 24, 2021, from https://en.wikipedia.org/wiki/Generative_adversarial_network

HackerPoet. (n.d.). *HackerPoet/Composer*. Retrieved February 24, 2021, from https://github.com/HackerPoet/Composer

Jhamtani, H., & Berg-Kirkpatrick, T. (2019). *Modeling self-repetition in music generation using generative adversarial networks*. In Machine Learning for Music Discovery Workshop, ICML. Long Beach, USA. http://www.cs.cmu.edu/~jharsh/papers/music_workshop_paper.pdf

Johnston, D. (2015, Aug). *Composing music with recurrent neural networks.* Retrieved from https://www.danieldjohnson.com/2015/08/03/composing-music-with-recurrent-neural-networks/

Kang, D. Kim, J. Y. & Ringdahl, S. (2018). *Project milestone: Generating music with Machine Learning*. Retrieved from http://cs229.stanford.edu/proj2018/report/18.pdf

Twilio. (2019, Jan). *Generating music with Python using Magenta for TensorFlow.* Retrieved February 24, 2021, from https://www.youtube.com/watch?v=2f20d0LJSuk&ab_channel=Twilio

Vasic, T. (2021, January 09). *Chopin midi.* Retrieved February 24, 2021, from https://www.kaggle.com/teodoravasic/chopin-midi

Yang, Li-Chia, Chou, Szu-Yu, & Yang, Yi-Hsuan. (2017). *MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation*. https://arxiv.org/pdf/1703.10847.pdf