# Generation of Music With Dynamics Using Deep Convolutional Generative Adversarial Network

Raymond Kwan How Toh
School of Computer Science and Engineering
Nanyang Technological University
Singapore
rtoh004@ntu.edu.sg

Alexei Sourin
School of Computer Science and Engineering
Nanyang Technological University
Singapore
assourin@ntu.edu.sg

***Abstract*—Following the rapid advancement of Artificial Intelligence and transition into the era of Big Data, researchers have started to explore the possibility of using machine learning in creative domains such as music generation. However, most research were focused on musical composition and removed expressive attributes during data pre-processing, which resulted in mechanical-sounding generated music. To address this issue, music elements, such as pitch, time and velocity, were extracted from MIDI tracks and encoded with piano-roll data representation. With the piano-roll data representation, Deep Convolutional Generative Adversarial Network (DCGAN) learned the data distribution from the given dataset and generated new data derived from the same distribution. The generated music was evaluated based on its incorporation of music dynamics and a user study. The evaluation results verified that DCGAN could generate expressive music comprising of music dynamics and syncopated rhythm.**

***Keywords-music generation; DCGAN; dynamics; pianoroll***

## I. Introduction

Music Generation is composing music with a computer that uses machine learning methods. Researchers have started to explore machine learning in music generation with the increases of data and research in artificial intelligence.

Deep Learning is a subfield of machine learning methods capable of identifying patterns and making decisions without being explicitly programmed. It shows promising results in several fields such as natural language processing for texts, computer vision for images, and voice recognition for speech and more [1]. Also, deep learning techniques in artificial music generation have been met with great success in generating human-like compositions.

However, most research focused on musical composition and ignored the aspect of expressive musical performances. Hence, music information stored in Musical Instrument Digital Interface (MIDI) tracks, such as velocity, was deemed unnecessary during training time. This causes the generated music to sound rather mechanical and dull.

This paper is about designing a music generation system that could make music that is incorporated with velocity, also known as music dynamics. To achieve it, the training data must encode velocity together with pitch and duration information. Piano-roll (an image-like data representation) was used to encode the information with one axis representing the time, and another axis representing the pitch. Each pixel intensity of 0-128 was represented as the velocity of the notes. DCGAN was chosen to be the deep learning architecture used in this paper. It can capture and learn the data distribution given a dataset and generating a sample from the same distribution. Lastly, the model will generate 4 sample excerpts: two are synthesized with music dynamics and two are not. These excerpts were analyzed, and a user study was conducted.

## II. Related Work in Music Generation

There are many deep learning architectures used for music generation system. They range from a basic feed-forward neural network to a generative adversarial network. In this paper, we will be focusing on recurrent neural network and generative adversarial network.

### A. Recurrent Neural Network

Recurrent Neural Network (RNN) [1, 2] is the most commonly used architecture in music generation system. It is a feedforward neural network that feds the output in the hidden layers back to itself and to the next hidden layer. Hence, it possesses the ability to capture information and learn from sequence data. However, RNN suffers from the vanishing and exploding gradient problem due to the constant updates of the weight matrix during backpropagation.

A variation of RNN, Long Short-Term Memory (LSTM) [3], solved the problem in a novel RNN by using various gates. These gates are the input gates, output gates, forget gates and a cell state. Cell states act as a transmission line for information to pass through. The input gates select a candidate from the inputs and update the previous relevant information. On the other hand, the forget gates remove irrelevant information from the cell states. Lastly, the output gates determine the information to pass to next hidden states.

LSTM is used in an expressive music generation system Performance RNN [4], developed by a google group Magenta. It uses the velocity, pitch and note durations found in the MIDI tracks. The generated music was observed to have dynamic and phrasing. However, it was unable to demonstrate long term structure.

2642-3596/21/$31.00 ©2021 IEEE
DOI 10.1109/CW52790.2021.00030

### B. Generative Adverserial Network

Generative Adversarial Network (GAN) [5] also was used in music generation and it shown impressive results. It consists of two network models: a generator and a discriminator which are playing a zero-sum game. The generator's objective is to transform a random noise input into a sample corresponding to the distribution of the real samples to fool the discriminator. However, the discriminator's objective is to distinguish a real sample from a generated sample. The discriminator is removed after both the network models reach Nash equilibrium.

MuseGAN [6], a music generation system developed by Music and AI Lab in Taiwan, can generate multi-track music without any human-intervention. It encoded the pitch and the note duration from MIDI tracks into a piano-roll format. Different musical instruments were encoded into different piano-roll and stacked together to form the multi-track piano-roll as the training data.

## III. Proposed method

In this paper, we designed and implemented a music generation system that can generate polyphonic music with dynamic using Deep Convolutional Generative Adversarial Network (DCGAN). With the image-like data representation and DCGAN ability to generate sample from the same data distribution, we hypothesized that the DCGAN can generate music that incorporated with music dynamics. To prove the hypothesis, a music analysis and user study was conducted.

### A. Data Representation

We used piano-roll data representation. A piano-roll is defined as a matrix where the horizontal axis represents time, and the vertical axis represents the note pitches. Each value in the matrix represents the presences of notes over the time steps. In the temporal axis, the tempo information was removed which results in the same beat length for all tempos. As Figure 1 shows, the *y*-axis represents the 128 different pitches, *x*-axis represents the timesteps, and the pixel intensity represents the velocity from 0 to 128.

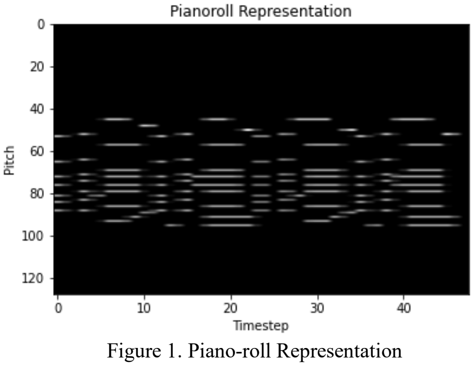


Figure 1. Piano-roll Representation

### B. Dataset

The Oracle Hip Hop Sample Pack from Cymatics [7] consists of 100 MIDI tracks and 103 WAV tracks. In this paper, we only used the MIDI tracks as it contains the musical information in symbolic form. To simplify the training data, we considered MIDI tracks with duration of either 8 or 16 seconds long and tracks with minor tonality as majority of the tracks falls under this category.

MusPy [8], an open-source Python library for symbolic music generation, was used to extract the MIDI information such as pitch, velocity and note duration. This information was then used to encode into a piano-roll data representation with the help of MusPy tools.

### C. Data Transposition

To generalize the system, a common technique is to transpose all training data to a single musical key or to all musical keys [1]. We transposed all the training data to a single musical key, C major or its relative minor A minor.

## IV. Implementation Details

### A. Data Pre-processing

In data pre-processing, we load the MIDI file into program using MusPy tools and parse it into a music object with a temporal resolution of 12 timestep per beat. The music object was constrained by clipping its velocity to at least value of 40, as it is inaudible for human ear and transpose to a single musical key of C major. As musical piece will not have all the 128 pitches, we decided to extract only up to 4 octaves of pitches (48 notes). The output dimension of the training data was (N, 48, 48, 1) where N is the number of training sample and (48, 48, 1) was the dimension of a single piano-roll.

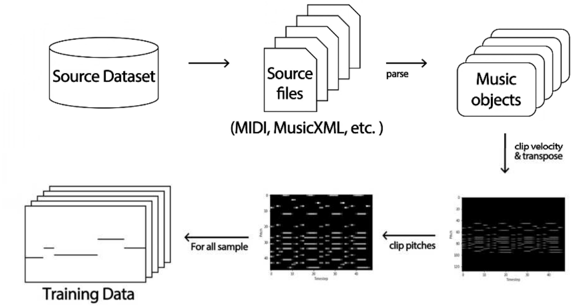


Figure 2. Data Pre-processing

### B. Generator

As Figure 3 shows, the generator takes an (100, 1) random vectors as inputs, reshape into *M*N*C*. The *M* and *N* represent the height and width, respectively, and *C* represents the number of channels. It then passes through 3 Conv2DTranspose layers with 128, 64 and 1 filters with activation function of *ReLU*, *ReLU,* and *Tanh,* correspondingly. The output dimension of the generator will be the size of piano-roll which have the shape of (48, 48, 1).

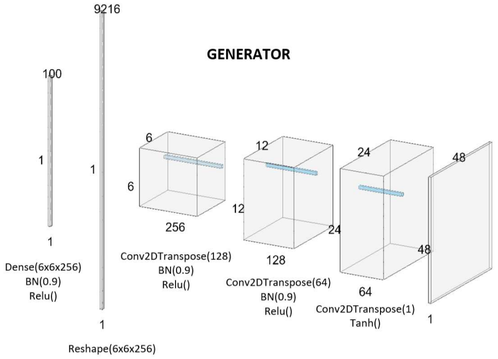


Figure 3. Generator Architecture

## C. *Discriminator*

For the discriminator, it was designed in the reversed order of the generator. It takes in the input of the piano-roll and passes through 3 Conv2D layers with activation function of *LeakyReLU*. It was then flattened and passes through 3 Dense layers before using a *Sigmoid* activation function to predict the output value.

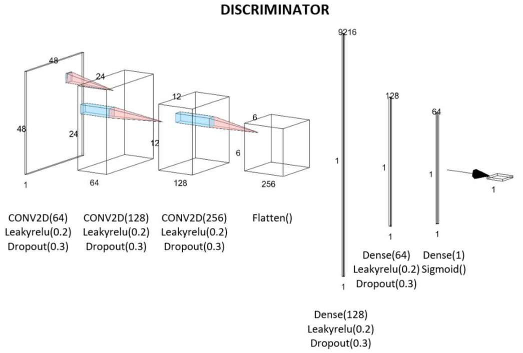


Figure 4. Discriminator Architecture

## D. *Model Training*

The model was trained on a graphics processing unit of a Nvidia GeForce RTX 2080. Both generator and discriminator use the Adam optimizer for the training. Also, to reach Nash equilibrium, the generator was updated 5 times more frequently compared to the discriminator as its task was more complex. Table 1 shows the hyperparameters used to train the model.

Table 1. Hyperparameter

| Hyperparameter | Values |
|---|---|
| Learning Rate | 5e-6 |
| Number of Epochs | 100K |
| Batch Size | 32 |
| Adam_decay_rate_1 | 0.5 |

## E. *Data Post-processing*

To return to original piano-roll before data-preprocessing, the generated piano-roll was padded to dimension of (128, 48, 1), as shown in Figure 5.

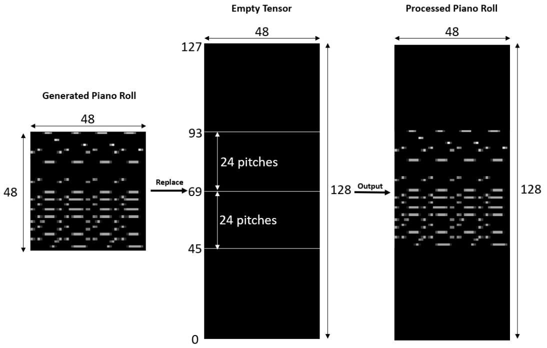


Figure 5. Data Post-processing

## F. *MIDI Generation*

To generate MIDI tracks from post-processed piano-roll, MusPy library functions were used to convert the piano roll back to MusPy music object, which was then written into a MIDI tracks.

## V. MUSIC ANAYLSIS AND USER STUDY

Music is largely subjective in nature. Hence, quantitative and qualitative measures were used to evaluate the experimental results.

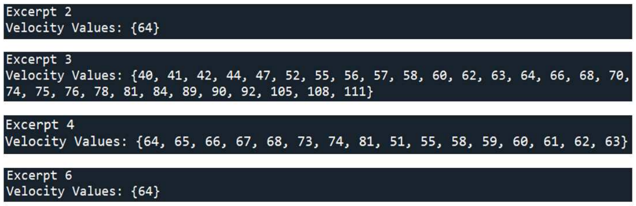
Four samples were generated by the system: two are synthesized with music dynamics and two without. Music analysis and user study were conducted to verify if the sample generated by the system was able to generate music incorporated with expressive elements.

Table 2. Excerpts Link

| Track | Link |
|---|---|
| Excerpt 2 (without dynamic) | https://tinyurl.com/5xgjfe5o |
| Excerpt 3 (with dynamics) | https://tinyurl.com/xanpq85r |
| Excerpt 4 (with dynamics) | https://tinyurl.com/11iz3otu |
| Excerpt 6 (without dynamics) | https://tinyurl.com/2hjybe8r |

## A. *Generated Music Analysis*

The generated music was parsed as a music object using python programming and extracted the music dynamics information. Figure 6 shows that samples synthesized without music dynamics only had a default value of 64, whereas samples synthesized with music dynamics had varied values across all notes.



```
Excerpt 2
Velocity Values: {64}

Excerpt 3
Velocity Values: {40, 41, 42, 44, 47, 52, 55, 56, 57, 58, 60, 62, 63, 64, 66, 68, 70,
74, 75, 76, 78, 81, 84, 89, 90, 92, 105, 108, 111}

Excerpt 4
Velocity Values: {64, 65, 66, 67, 68, 73, 74, 81, 51, 55, 58, 59, 60, 61, 62, 63}

Excerpt 6
Velocity Values: {64}
```

Figure 6. Music Dynamics Values

## B. *User Study*

The user study was designed to determine if the participants could identify the expressive elements of music dynamics and evaluate how musical the generated music are compared to human-performed music. Among the 48 participants, 27 were musically trained, 40 played an instrument, 26 had relative pitch, and 6 had perfect pitch. The user study was conducted online via Google Form and consisted of 6 questions, as shown in Table 3. Questions 2 to 6 were linear scale questions, scaled from 1 to 5, with 1 being very bad and 5 being very good.

Table 3. User Study Questions

| Question S/N | Question |
|---|---|
| 1 | What do you like or dislike about it? |
| 2 | How interesting would you rate this excerpt? |
| 3 | How melodious would you rate this excerpt? |
| 4 | How harmonious would you rate this excerpt? |
| 5 | How expressive would you rate this excerpt? |
| 6 | On a scale of 1-5, how likely was this excerpt composed and performed by a human? |

### 1) *Samples synthesized without music dynamics*

For Excerpts 2 and 6, the participants were able to identify the machine-generated music and its lack of expression. The participants also noted odd rhythm. Both excerpts received similar responses for Question 1 (Table 4).

Table 4. User Responses for Excerpts 2 and 6

| Dislike | Like |
|---|---|
| The melody's rhythm was weird | The unstable rhythm amplified the mood of the melody. |
| No fixed rhythm, no expression | Some of the ideas are interesting. The rhythm seems cool. Ideas could be taken further. |
| Sounds robotic | I like the odd time signature |
| Felt like the tempo was not consistent. | It has nice rhythm, but the notes sound boring and monotone |
| Sounds quite dull | Interesting time signature |
| Rhythm inconsistent, hard to listen to | There were some instances of interesting harmonies |

### 2) *Samples synthesized with music dynamics*

For Excerpts 3 and 4, most of the participants concluded that the music was likely played by humans. Most also commented on the syncopated rhythm, and some were able to recognize the variations in music dynamics. Both excerpts received similar responses for Question 1, shown in Table 5.

Table 5. User Responses for Excerpts 3 and 4

| Dislike | Like |
|---|---|
| Emphasis on some notes especially loud, does not sound pleasant or in harmony | More expression and varying strength of notes |
| Not conveying much message | Interesting dynamic |
| I feel like the dynamics don't really sync up with the melody | It is clearer in its expression. It has a great use of accents and softer notes that gel together. |
| The rhythm at some points were odd. the accents of some notes felt too forced too. | Good flow and dynamics |
| Some notes end abruptly | More dynamics and musicality |

The excerpts rating on expressiveness are found in Figure 7. Excerpts 2 and 6, which were machine-generated without dynamics, produced a positively skewed distribution. This suggests that most participants felt that the excerpts were not expressive. In contrast, Excerpts 3 and 4, which were machine-generated with dynamics, produced a negatively skewed distribution. This suggests that the excerpts were expressive. We also noted that Excerpts 3 and 4 are comparable to Excerpt 1, which was played by human with dynamics, based on the given distribution. The participants described the excerpts as “quite melodic” and “active and lively”. These responses indicate that the music generation system can generate interesting harmonic progression and melodic sequences.

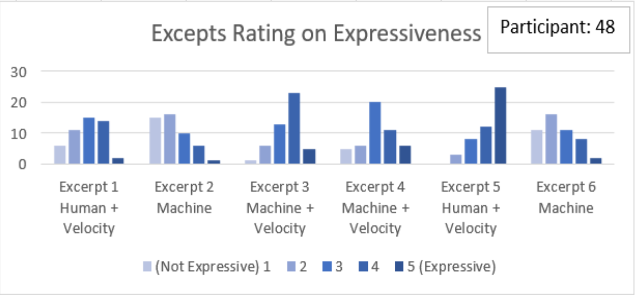


Figure 7. User Study Expressive Results

## VI. Conclusion

In this paper, a music generation system was implemented under the framework of GANs, DCGAN. The generated samples were analyzed, and a user study was conducted. The results of the analysis and the user study prove our hypothesis that the system can generate expressive music with music dynamics and syncopated rhythm. Future research using GAN in music generation could focus on generating arbitrary length of music or gather a larger data set for training.

## References

[1] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, "Deep learning techniques for music generation--a survey," *arXiv preprint arXiv:1709.01620,* 2017.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[4] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications,* vol. 32, no. 4, pp. 955-967, 2020.

[5] I. J. Goodfellow *et al.*, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661,* 2014.

[6] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.

[7] "Cymatics.fm - The #1 Site For Serum Presets, Samplepacks & More!" https://cymatics.fm/ (accessed 4 Oct, 2020).

[8] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "MusPy: A Toolkit for Symbolic Music Generation," *arXiv preprint arXiv:2008.01951,* 2020.