2019 2nd International Conference on Electronics Technology

Generating Music Algorithm with Deep Convolutional Generative Adversarial Networks

Hongyu Chen
Lennon Lab
HiFive Tech Co.
Chengdu, China
e-mail: chenhongyu@hifive.ai

Qinyin Xiao*
Data and Information Center
Sichuan Institute of Computer Sciences
Chengdu, China
e-mail: xiaoqinyin@msn.com

Xueyuan Yin
Lennon Lab
HiFive Tech Co.
Chengdu, China
e-mail: yinxueyuan@hifive.ai

***Abstract*—With the extensive development of deep learning, automatic composition has become a vanguard subject exercising the minds of scientists in the area of computer music. This paper proposes an advanced arithmetic for generating music using Generative Adversarial Networks (GAN). The music is divided into tracks and the note segment of tracks is expressed as a piano-roll, through trained a gan model which generator and discriminator continuous zero-sum game to generate a wonderful music integrallty. In most cases, Although GAN excel in image generation, the model adopts a full-channel lateral deep convolutional network structure according to the music data characteristics in this paper, generate music more in line with human hearing and aesthetics.**

Keywords-AI composition; deep convolution GAN; full-channel lateral; piano-roll; time sequence data structure algorithm optimization

I. INTRODUCTION

A. Background

Music is an art that reflects the emotions of human life. Music can improve people's aesthetic ability, purify people's minds, and establish lofty ideals. We express our emotions through music and release many of our emotions. Music has been integrated into all aspects of our lives. With the provision of living standards, the demand for music is also increasing.

Artificial Intelligence, is a new technical science that studies and develops theories, methods, techniques, and application systems for simulating, extending, and extending human intelligence. Artificial intelligence gained more and more attention in the computer field. And it also has gained important applications in robotics, economic and political decision-making, control systems, and simulation systems.

Computer music is the product of the fusion of computer and music art. The computer is rigid and lifeless. The music is flexible and full of art. Before the AI was combined with music, the computer made people have no hope in music creation, but when AI and music were connected, computer music creation seemed to see the dawn, and computer music creation came alive. This is also an essential basis for the study of AI music in this paper.

B. Research Status

As an auditory art discipline, music can't be qualitatively and quantitatively analyzed and even can't be described by simple words, and there is no clear indicator to assess. Above mentioned is the biggest difficulty in Music Information Retrieval. Fortunately, music has its own expression language, consisting of a series of consecutive specific note sequences. This will at least make it easier for the computer to express the music in time series, and it will greatly facilitate our research and calculation of music.

At present, the integration of AI and music has been studied at home and abroad, mainly in deep learning models such as GAN[1], CNN[2] and LSTM[3]. This is because music has the characteristics of typical time series data, which is in line with the advantages of the above model.

C. Introduction to GAN

GAN (generative adversarial network) is an artificial intelligence algorithm for unsupervised learning. It was proposed by Goodfellow et al. in 2014[4]. GAN is constituted by two networks, generation network G and discriminator network D, G is responsible for generating target objects. D is responsible for discriminating the object generated by the generator from the real object. With the separate training of the two networks and play zero-sum games with each other in the adaptation training. The generator will generate objects that are very similar to the genuine objects, so that the discriminator network cannot distinguish between the generated objects and the real objects, thereby achieving GAN training.

* Corresponding author: Qinyin Xiao(xiaoqinyin@msn.com)

978-1-7281-1618-1/19/$31.00 ©2019 IEEE

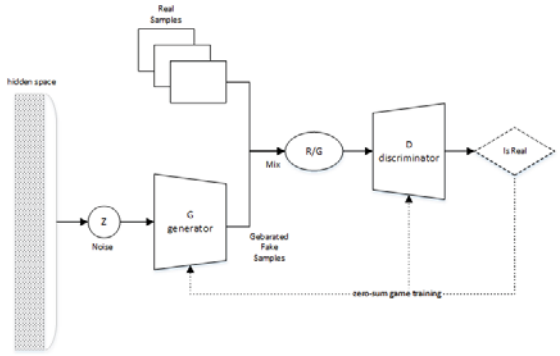


Figure 1. GAN Network architecture. the generator obtains random noise Z in the hidden space, generates a batch of data and inputs it into the discriminator together with the real data, and the discriminator D trains to separate the real data and the generated data as a target, and after the discriminator D is stable, generates The device is trained for the purpose of deceiving the discriminator D, so that the data is generated by the generator G so that the data generated by the generator G is similar to the real data.

Objective function equations:

$$\min_G \max_D V(D,G) = \sum_{x \sim P_{data}(x)} [\log D(x)] + \sum_{z \sim P_z(z)} [\log(1 - D(G(x)))]$$

x represents real data, and z represents noise input to the G-network. G(z) represents the data generated by the G-network, and D(x) represents the probability of judging whether the real data x is true (this value is close to 1 indicating that the D-network is better), then D(G(z)) indicates the probability that the D-network determines whether the data generated by the G-network is true.

In the above equations, the purpose of the G-network is to make the data generated by the D-network to be as small as possible, so it is recorded as min(G). The purpose of the D-network is to be able to distinguish the data generated by the G-network from the real data and obtain the maximum discriminant probability, which is denoted as max(D).

GAN has been widely researched and apply. Here's a brief list of GAN's advantage and disadvantages.

1) Advantage

- GAN is a generative model that uses only backpropagation compared to other generation models (Boltzmann machines[5] and GSNs[6]) without the need for complex Markov chains[7].
- GAN can produce a more clear, realistic sample closer to the real object.
- GAN adopts an unsupervised learning style training, which can be widely used in unsupervised learning and semi-supervised learning.
- Compared to the variational autoencoders[8], GAN does not introduce any deterministic bias, and the variational method introduces deterministic bias because they optimize the lower bound of the log likelihood rather than the likelihood itself. Compared with VAE, GAN has no variation lower bound. In other words, GAN is progressive, but VAE is biased.

2) Disadvantages

- Training GAN needs to reach Nash equilibrium [9], sometimes it can be done by gradient descent method or not, We have not found a good way to reach Nash equilibrium, so training GAN is unstable compared to VAE or PixelCNN[10], but I think in practice it is still more stable than training the Boltzmann machine.
- GAN is not appropriate for processing discrete forms of data, such as text.
- GAN has problems of unstable training, gradient disappearance, and mode collapse.

3) Variant

- **DCGAN:** Deep Convolutional Generative Adversarial Networks[11], greatly improves the stability of GAN training and the quality of the resulting results.
- **WGAN:** Wasserstein GAN[12], solve problems such as mode collapse, improve learning stability, and provide meaningful learning curves useful for debugging and hyperparametric searches.
- **LSGAN:** Least Squares Generative Adversarial Networks[13], Solving the vanishing gradients problem during the learning process by using a least squares loss function for the discriminator.
- **BEGAN:** Boundary Equilibrium GAN[14], training auto-encoder based Generative Adversarial Networks which Replace the similarity between the distribution by estimating the similarity between the distribution errors of the distribution to achieve fast and stable training.

II. ALGORITHM DESIGN

A. Data Structure

Music is a sequence of notes arranged in time. Music can be separated into main tone music and polyphonic music according to the tone classification. A complete music according to the performance mode, the instrument can be generally separated into: keyboard, wind, string, percussion, electronic. Music can be used in the computer piano_roll is expressed in the form of piano_roll. In order to solve a problem that can solve polyphonic music and multi-track orbit generation, this paper proposes a scheme consistent with this algorithm. The form of the data is presented in the following figure. The data dimension is: [N * track * time_step * pitch].

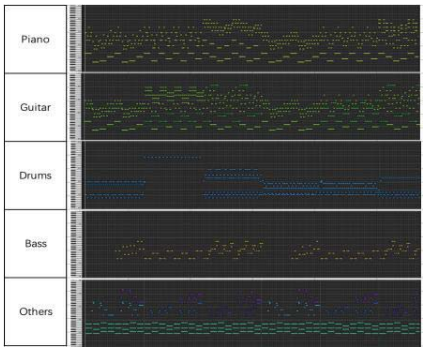


Figure 2. Spilt five tracks in the piano-roll.

B. Generator & Discriminator

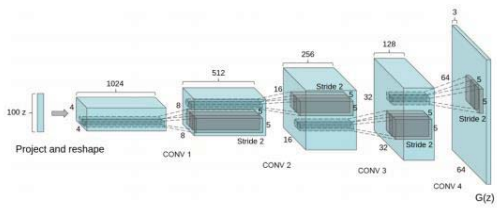


Figure 3. DCGAN generator is used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with various feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64 × 64 pixel image. Notably, no fully connected or pooling layers are used.

The network structure of the discriminator is opposite to that of the generator. In addition to the final output layer, we use the sigmoid activation function to distinguish between real data and generated data.

In addition, the discriminator of the GAN model has been specially optimized and adjusted according to the nature of the music data. Our convolution uses full-channel lateral convolution, because the music is continuous and the most important is at the same time, the notes are sounded at the same time.

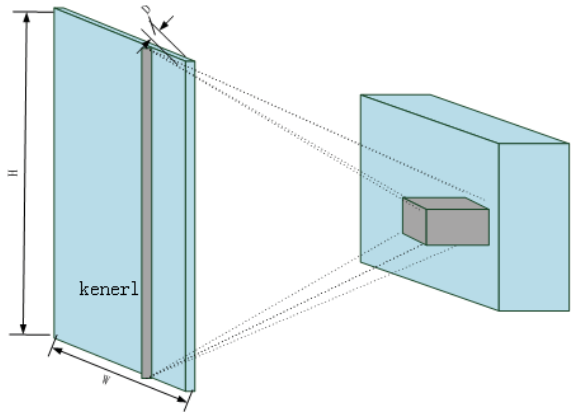


Figure 4. full-channel lateral convolution kenel in convolution of discriminator.

DCGAN almost completely uses the convolutional layer instead of the full-connection layer. The discriminator is almost symmetrical with the generator. From the above figure, we can see that there is no pooling layer and upsampling layer in the entire network. In fact, the tape is actually used. The fractional-stride convolution replaces the upsampling to increase the stability of the training.

Using the step size convolution instead of the upsampling layer, the convolution has a good effect on extracting image features and uses convolution instead of the fully connected layer. More importantly, we use the full-channel lateral convolution kernel, because the piano roller blinds as a form of music, we do not need to care too much about the longitudinal convolution, and such a convolution form allows the model to learn Be more focused on musicality and converge faster.

Use the LeakyReLU activation function in the discriminator instead of ReLU to prevent gradient sparsity [15]. Relu is still used in the generator, but the output layer uses tanh.

Training with the adam optimizer, and the learning rate is preferably 0.002. Replace the pooling layer with convolutions. (For discriminant models, the network is allowed to learn its own spatial downsampling; instead of generating a model, it is allowed to learn its own spatial upsampling).

C. GAN Model

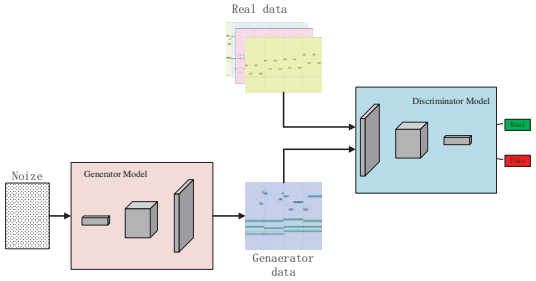


Figure 5. GAN train process.

GAN train process follows as pseudo code:

```
##start
   generator = Generator_build()
   discriminator = Discriminator_build ()
   combiner = discriminator (generator(noise_z))
   for epoch in range(epochs):
        # Train Discriminator
        # Generate a half batch of new music
         gen_musics=generator.predict(noise)
         # Train the discriminator
         discriminator.train(gen_musics)
        # Train Generator
        combiner.train()
##end
```

III. EXPERIMENT

A. Data Set

The mentioned above dataset from The Lakh MIDI Dataset[1], which is a collection of 176,581 unique MIDI files, 45,129 of which have been matched and aligned to entries in the Million Song Dataset. These MIDI files are a complete song, including musical instrument combinations of multiple tracks, such as piano, guitar, bass, drum etc. Each track contains some attributes of the instrument, including instrument name and note sequence. The sequence of notes consists of beat, pitch, velocity, and duration.

[1] https://colinraffel.com/projects/lmd/

B. Data Process

Due to the algorithm design of the paper, we are required to convert the prepared midi file into a piano roller blind and further transform it into the input data format we need to train.

First, considering that we are expected to generate some Pop music. Convert the MIDI file to piano-roll, and spilt it to five tracks: piano, guitar, bass, drum, others. Popular music is mainly composed of the above instruments, and other less common instruments are classified as others.

Second, Consideration of training model, we need to slice and standardization the midi track file. We use the 4 bars as the window unit and cut it longitudinally. If you don't have enough data, you can split the window in smaller steps. And, we separated a 4 bar and 4/4 beats into 192 copies. Since the pitch of most instruments doesn't cover 0-127, we shorten it to 0-84. Finally, the data we get after standardization such as [N * 5(track) * 192(time step) * 84(pitch)]. In particular, we will clean and filter the dirty and nonstandard data.

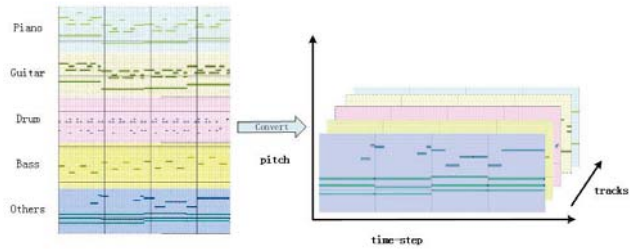


Figure 6. Convert complete piano-roll to train's input data.

C. GAN Train

The model mentioned in the paper contains two networks: generator model G, discriminator model D. This paper uses CNN's network structure for training. The network hierarchy diagram is shown below.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 32)	0
dense_1 (Dense)	(None, 80640)	2661120
leaky_re_lu_1 (LeakyReLU)	(None, 80640)	0
reshape_1 (Reshape)	(None, 192, 84, 5)	0
conv2d_1 (Conv2D)	(None, 192, 84, 32)	4032
leaky_re_lu_2 (LeakyReLU)	(None, 192, 84, 32)	0
conv2d_2 (Conv2D)	(None, 96, 42, 32)	25632
leaky_re_lu_3 (LeakyReLU)	(None, 96, 42, 32)	0
conv2d_3 (Conv2D)	(None, 48, 21, 8)	6408
leaky_re_lu_4 (LeakyReLU)	(None, 48, 21, 8)	0
conv2d_transpose_1 (Conv2DTr	(None, 192, 84, 32)	2336
leaky_re_lu_5 (LeakyReLU)	(None, 192, 84, 32)	0
conv2d_4 (Conv2D)	(None, 192, 84, 16)	12816
leaky_re_lu_6 (LeakyReLU)	(None, 192, 84, 16)	0
conv2d_5 (Conv2D)	(None, 192, 84, 8)	2056
leaky_re_lu_7 (LeakyReLU)	(None, 192, 84, 8)	0
conv2d_6 (Conv2D)	(None, 192, 84, 5)	1005

Figure 7. Generator model G-network.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	(None, 192, 84, 5)	0
conv2d_7 (Conv2D)	(None, 192, 84, 5)	630
leaky_re_lu_8 (LeakyReLU)	(None, 192, 84, 5)	0
conv2d_8 (Conv2D)	(None, 94, 40, 5)	630
leaky_re_lu_9 (LeakyReLU)	(None, 94, 40, 5)	0
conv2d_9 (Conv2D)	(None, 47, 20, 1)	81
leaky_re_lu_10 (LeakyReLU)	(None, 47, 20, 1)	0
conv2d_10 (Conv2D)	(None, 24, 10, 1)	17
leaky_re_lu_11 (LeakyReLU)	(None, 24, 10, 1)	0
flatten_1 (Flatten)	(None, 240)	0
dropout_1 (Dropout)	(None, 240)	0
dense_2 (Dense)	(None, 1)	241

Figure 8. Discriminator model D-network.

After establishing the G-network and D-network models, we started to train the network GAN and there is some tips.

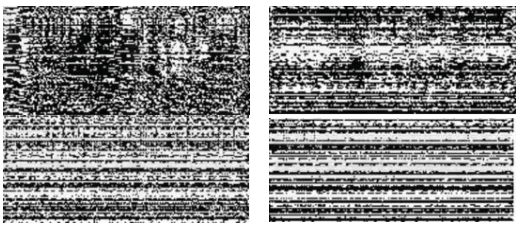
- When we train the discriminator, hold the generator values constant; and when we train the generator, hold the discriminator constant. Each should train against a static adversary.
- By the same token, pretraining the discriminator against dataset before we start training the generator will establish a clearer gradient.
- Each side of the GAN can overpower the other. If the discriminator model is overtrained, it will return values so close to 0 or 1 that the generator will struggle to read the gradient. Otherwise, it will persistently exploit weaknesses in the discriminator that lead to false negatives. This may be mitigated by the nets' respective learning rates.
- GAN model training is a quite time-consuming process, so we can consider using parallel GPU for training under conditions, otherwise, we can only do other things while training.
- 

Figure 9. The sample is generated by the model.

During the experiment, we found that we can perform as many iterations as possible, but in the training process, we can also try to generate some sample for verification. After all, in the music generation algorithm, it is not necessary to get a colossal model but a model that sounds better.

IV. EVALUATE

Music is a subject with a particularly subjective nature. At present, it is not possible to judge the quality of music without hearing. So, we use two sets of programs to evaluate our experimental results.

The first set of evaluation systems is based on statistically objective evaluations [16]. We invited some experts in the music field to evaluate and analyze some high-quality music and got some indicators from the midi data level. It is mainly divided into two categories, one is the similarity between the musical instrument orbital note sequence and the chord, because we believe that harmony is an important component of people's sense of hearing. The other is the interval relationship between notes and notes, which are well documented in both psychology and biophysics.

The second set of the evaluation system is to conduct a sample survey. We randomly selected a number of different styles of songs, mixed with some human-made music that people don't often hear, and invited our friends and colleagues to rate all the songs, although some people can tell which ones It is made by humans and which is machine-generated, but they still make a high evaluation of machine-generated music. A few people think that the music generated by the machine brings them surprises and shocks.

V. CONCLUSION

In this paper, we propose a generation model for generating note sequences under the GAN framework. We use the deep convolution neural network and optimize it according to the musical note characteristics, this optimization algorithm enables the convolution network to concentrate on learning music features and make experiments faster. At the same time, we also get some knowledge of ordinary sense to speed up the training of discriminators. Experimental data and subjective user evaluations show that proposed model can generate music-like sequence generation. Although the experimental results are below the human level, the model is theoretically mature and has ideal properties. We hope that the subsequent research can be further Improve it.

ACKNOWLEDGMENT

First of all, I would like to extend my sincere gratitude to Dr. Yin for instructive advice and effective recommendations on my thesis. I am likewise deeply indebted to the HIFIVE company providing cloud computing resources and quiet and comfortable office space. In the meanwhile, thanks to all colleagues for their professional knowledge. I should finally like to express my gratitude to my beloved Prospective wife and parents who have always been helping me out of difficulties and supporting without a word of complaint.

REFERENCES

[1] Mogren, Olof. "C-RNN-GAN: Continuous recurrent neural networks with adversarial training." arXiv preprint arXiv:1611.09904 (2016).

[2] Choi, Keunwoo, et al. "Convolutional recurrent neural networks for music classification." 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017.

[3] Eck, Douglas, and Juergen Schmidhuber. "A first look at music composition using lstm recurrent neural networks." Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale 103 (2002).

[4] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in neural information processing systems. 2014: 2672-2680.

[5] Ackley, David H., Geoffrey E. Hinton, and Terrence J. Sejnowski. "A learning algorithm for Boltzmann machines." Cognitive science 9.1 (1985): 147-169.

[6] Alain, Guillaume, et al. "GSNs: generative stochastic networks." Information and Inference: A Journal of the IMA 5.2 (2016): 210-249.

[7] Norris, James R. Markov chains. No. 2. Cambridge university press, 1998.

[8] Doersch, Carl. "Tutorial on variational autoencoders." arXiv preprint arXiv:1606.05908 (2016).

[9] Heusel, Martin, et al. "Gans trained by a two time-scale update rule converge to a nash equilibrium." arXiv preprint arXiv:1706.08500 12.1 (2017).

[10] van den Oord, Aaron, et al. "Conditional image generation with pixelcnn decoders." Advances in Neural Information Processing Systems. 2016.

[11] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

[12] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." arXiv preprint arXiv:1701.07875 (2017).

[13] Mao, Xudong, et al. "Least squares generative adversarial networks." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[14] Berthelot, David, Thomas Schumm, and Luke Metz. "BEGAN: boundary equilibrium generative adversarial networks." arXiv preprint arXiv:1703.10717 (2017).

[15] Xu, Bing, et al. "Empirical evaluation of rectified activations in convolutional network." arXiv preprint arXiv:1505.00853 (2015).

[16] Colin Raffel. "Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching". PhD Thesis, 2016.

[17] Yang, Li-Chia, Szu-Yu Chou, and Yi-Hsuan Yang. "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation." arXiv preprint arXiv:1703.10847 (2017).