2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST) | 978-1-6654-0267-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/IAECST54258.2021.9695533

A Symbolic-domain Music Generation Method Based on Leak-GAN

Zihan Li
Xihua Honor College,
Xihua University,
Chengdu, China,
kkli19@outlook.com

Yi Guo*
School of Electrical and Electronic Information,
Xihua University,
Chengdu, China,
lpngy@vip.163.com

Yangcheng Liu
School of Electrical and Electronic Information,
Xihua University,
Chengdu, China,
luming9704@163.com

Qianxue Zhang
School of Electrical and Electronic Information,
Xihua University,
Chengdu, China,
louxzqx@163.com

Abstract—**To efficiently train deep generative neural works, generative adversarial networks have been applied in plenty of fields including music generation. We introduce a novel way of generating symbolic-domain music via adversarial training, which allows the discriminator to leak its information to the generator for better guidance. To prove good quality of the generated pieces, we conduct comparative test on LSTM model and evaluate the randomly-sampled music on five metrics in the light of statistics and music theory. The results shows that our model is more capable of generating coherent, natural and realistic music works.**

Keywords—music generation; reinforcement learning;

I. Introduction

Music generation is commonly acknowledged as one of the most challenging but fascinating tasks over the last decades. Using a Markov model, Lejaren and Isaacson [1] succeeded in generating the first piece of music with computer in 1957. Attempts have also been made with recurrent neural networks [2,3] and long short-term memory (LSTM) [4,5]. In recent years the deep neural network such as convolutional neural network (CNN) and generative adversarial networks (GANs) have been applied in this field [6,7,8,9,10] and have shown tremendous potential in their capability for content generation and modification. Yang et al. [6] proposed a novel conditional mechanism under adversarial training to generate realistic music that is pleasant to listen to. Based on the understanding of how pop music is composed, Dong et al. [10] proposed three different methods combined with GAN to handle the interaction among tracks and generate polyphonic music with harmonic and rhythmic structure. While Midinette [6] choose to use CNN in their model, RNN has also been applied in such occasion. Mogren [11] proposed a continuous recurrent network via adversarial training in order to a highly flexible and expressive model with fully continuous sequence data for tone lengths, frequencies, intensities, and timing.

However, to our knowledge the GAN approach is still limited and suffers from exposure bias and mode collapse [12]. Moreover, music generation is still in primary stage and their work fail to completely reflex the complex relationship of musical elements of the realistic musical works in the space and time dimensions [13].

In this paper we implement symbolic-domain music melody generation experiment on a model of LeakGAN [14] based on network architecture, which allows the discriminator to leak the extracted features from higher level to the generator to help the guidance more distant, thus solving the problem of sparsity in guiding signals while training. With a novel way of representing music data, we conduct a certain number of experiments based on the idea of taking the generation problem as a sequential decision-making process [15], using evaluation metrics of both statistics and music theory. In terms of these metrics, our model shows improvements to a certain extent compared to other models. Apart from that, our model is proved to be having the ability of rapidly achieving high discriminator accuracy.

II. Model Structure

With transferring midi files to text files, the melody generation is taken as a text generation problem and then the Leak-GAN [14] generation model is used to generate text in this work. In this section we firstly introduce our representation of music and then the model we used.

A. Music Representation

In the information age, music can be stored in many ways by computer, among which we choose MIDI to be our main research format. Focusing on melody generation, we first extract the melody sound track and then calculate the distance between this MIDI's modality and the key of C to convert all notes to the key of C. To alleviate the influence of different

This work is supported by Research Funds of Sichuan Provincial Key Laboratory on Intelligent Terminals under grant SCITLAB-1021 and the Open Research Fund of Power System Wide-area Measurement and Control Sichuan Provincial Key Laboratory under Grant 2020JDS0027 .

978-1-6654-0267-5/21/$31.00 ©2021 IEEE

speed, the speed of all MIDI music is all set to 90. In the representing process, we propose a novel method which takes the notes of delay and pause into consideration.

1) Unit Time

In order to solve the problem of missing duration information in the representing process, we first get the length of the shortest note in the melody score and then we use T0 as the unit time to extract the duration information of all notes.

2) Symbolic Representation

English letters are used to represent notes. Specifically, the middle C of the piano is called C4, and the white keys on its right are C4, D4, E4, F4, G4, A4, B4, C5, D5, E5, F5, G5, A5, B5. E-4 (-: flat note) and D#4 (#: sharp note) both represents the black key between D4 and E4. Other than notes, "-" is used to represent a unit time of extension while "^" indicates the duration of the pause.

3) Cutting

As the length of one complete song may be too long for our generation model, we cut long text by piece according to analysis results of the experimental dataset.

B. LeakGAN in Music Generation

Inspired by the progressive GAN [16] approach, we choose Leak-GAN [14] model to generate our music. Composed of a discriminate net and a generate net, the Leak-GAN model allow the discriminator to leak its extracted features that is from higher level to the generator to help the guidance more distant, which effectively addresses the problem for long text-represented-music generation. Additionally, as LeakGAN model is skilled in long sequence generation, it produces promising results of adequate length as well as fine quality. Following we first introduce the general idea of generative adversarial network, then we take a closer look at the LeakGAN model.

1) Generative Adversarial Networks

Generative adversarial network [16] is a method of unsupervised learning that aims to learn by letting two neural networks play against each other. While one of the neural networks called generator (G) maps a random noise z sampled from a prior distribution to the data space, the other neural work called discriminator (D) outputs a single scalar which represents the probability of training examples coming from the data rather than the generator's distribution. Two networks train one another to exceed one another at the same time via rounds of generation and discrimination. The adversarial learning procedure can be described as a two-player minimax game between the generator and the discriminator with objective function:

$$\min_{G}\max_{D} V(D,G) = \mathrm{E}_{\mathrm{x}\sim p_{data}(x)}[\log D(x)] + \mathrm{E}_{\mathrm{z}\sim p_{z}}[1-\log(D(G(\mathrm{z})))], \tag{1}$$

where p_{data} is the real data's distribution and p_z represent the prior distribution of z.

2) Leak-GAN

On condition that we focus on symbolic-domain music generation, the sequence data of text-represented melody generation is considered as a sequential decision-making process [15]. In each time step t, the current state the action is denoted as $s_{\mathrm{t}}=(x_1,...,x_i,...,x_t)$ and x_{t+1} respectively, where x_i is one candidate token in the given vocabulary V .We train a θ-parameterized generative model G_θ to implement the action and a ϕ -parameterized discriminative model D_ϕ to guide G_θ for improvement.

With the approval of leaking discriminator D's high level feature representation to the generator who is not supposed to receive such information, the LeakGAN model addresses the problem that D's guiding signals are only available when the whole melody sequence s_T has been generated. Specifically, as the generated sentence length T gets longer, the guiding signal, which is a single scalar of the discriminator, normally contains less information. To alleviate this situation, the LeakGAN model does not only provide generator with discriminator's information, it additionally applies a hierarchical reinforcement learning architecture to the generator in order to coordinate the leaked information with generation process of G_θ.

We introduce a MANAGER module and a WORKER module of the generator, which both start from an all-zero hidden state, denoted as h_0^W and h_0^M . Fig1 is an overview of the model structure. In every step, the discriminator sent leaked feature vector f_t to the MANAGER, combined with the current state of which to produce the goal vector g_{t} as:

$$\hat{g}_t, h_t^M = M(f_t, h_{t-1}^M; \theta_m), \tag{2}$$

$$g_t = \hat{g}_t / \| \hat{g}_t \|, \tag{3}$$

where $M(\cdot\ ;\theta_m)$ represents that the LSTM [18] that MANAGER uses is with parameters θ_m and the present hidden vector is denoted as h_t^M .For the leaked feature f_t, we have

$$D_\phi(s) = \mathrm{sigmoid}(\phi_f \mathrm{F}(s;\phi_f)) = \mathrm{sigmoid}(\phi_f f), \tag{4}$$

where $\mathrm{sigmoid}(z) = 1/(1+e^{-z})$. $\mathrm{F}(\cdot\ ;\phi_f)$ represents the CNN feature extractor [17] and $f = \mathrm{F}(\mathrm{s};\phi_f)$ is the feature vector of s in the last layer of D_ϕ.

Followed by a linear transformation ψ on a sum of c goals recently generated with weight matrix W_ψ , the goal embedding vector w_t is defined as follows:

$$w_t = \psi(\sum_{i=1}^{\mathrm{c}} g_{t-i}) = W_\psi(\sum_{i=1}^{\mathrm{c}} g_{t-i}) . \tag{5}$$

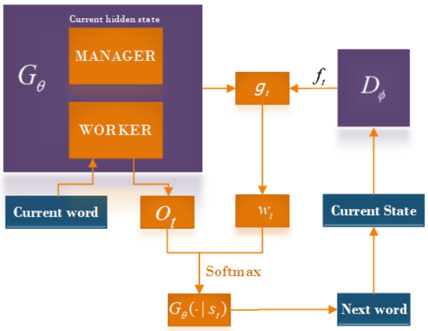

Figure 1. An overview of the model

We take x_t ,which is the current notes of generated melody, as input of our WOEKER module. To get final action space distribution of the next note to be select, we implement matrix product on w_t by O_t through a softmax under current state of the manager

$$O_t, h_t^W = W(x_t, h_{t-1}^W; \theta_w), \tag{6}$$

$$G_\theta(\cdot \mid s_t) = \mathrm{softmax}(O_t \cdot \mathrm{w}_t / \alpha), \tag{7}$$

where h_t^W is recurrent hidden vector of an LSTM and $W(\cdot\ ;\theta_w)$ represents the WORKER module. O_t denotes a matrix of the presents vector of all notes. while α is a temperature parameter illustrating the generation entropy.

III. Experiments and Results

A. Dataset

As we mainly focus on melody generation, the POP909[19] pop music dataset is the best choice at our knowledge, as its music melody track can be explicitly distinguished and directly derived. We convert the MIDI files to text files and process these files using the python library music21[20]. Specifically, we use a novel form of symbolic representation for better experiment results as previously mentioned. Firstly, we conduct normalization of speed and then transpose the keys to C. After we determine the exact duration according to unit time, all notes are represented by given characters. Based on our analysis of our data, some of the data are extremely long or short, thus we drop these highly differential data to avoid statistical inaccuracy.

Finally, the data are cut by about 150-word long pieces to seemly fit the training model. The reason we choose the length of 150 is that in our representation of pop song melodies, the verse and chorus of a song are approximately 150-word sequence.

After our processing, the dataset consists of 11792 melody data with an average length of 210-word.

B. Training Settings

1) Objective Metrics

For the generated music, we choose three mathematical statistics metrics [21], which are Wilcoxon Test, Mann-Whitney U(MWU) Test and Kruskal-Wallis H(KWH) Test respectively, to be our evaluation index. Furthermore, we take the music theory into account to test the generated data on two music theory evaluation metrics, the Smooth-saltatory progression (SSP) comparison and Note-level mode test. The specific explanation of these five metrics is presented in Table 1. The type CB means the closer the value is to number 1 the better while FB means the contrary.

TABLE I. Explanation of Metrics

Name	Explanation	Type
Wilcoxon Test	Reflection of how close the piece is to realistic music.	CB
MWU Test		CB
KWH) Test		CB
SSP comparison	Indication of if the piece is rhythmic.	CB
Note-level mode test	To check whether the notes of the generated notes are all in the C major key we stipulated.	FB

2) GAN Setting

We choose LSTM to be the architecture of both MANAGER and WORKER in the generator. For discriminator we choose CNN [17]. To solve the problem of mode collapse we adopt the interleaved training scheme [14] which alternates between adversarial training and supervised training i.e. Maximum Likelihood Estimate (MLE). In this way, not only we can reduce the problem of mode collapse, but also bring the GAN'S solution closer to that of the supervised training.

C. Results and Analysis

We randomly sample three tenths of the dataset as the test data and the remaining seven tenths as the train data, on which we run the adversarial training pretrained by MLE for 200 epochs. For the comparison experiment, LeakGAN model is mainly compared with LSTM. We train the 8 layers LSTM model with the same dataset for 500 epochs. A few samples generated by LeakGAN are illustrated in Fig. 2, which to a certain extent provide evidence for LeakGAN's ability of generating creative and harmonic music.

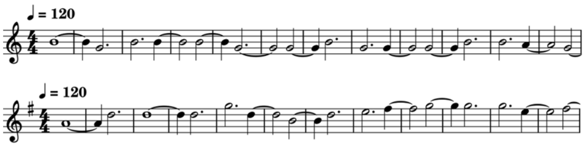

Figure 2. Two sample pieces of the generated music.

We use the five objective metrics mentioned before as evaluation scores. The results are provided in Table 2, from which we see that the first four values for LeakGAN model is relatively high and the last is lower than that of LSTM. This indicates that the music generated by the LeakGAN tends to perform well both in statistics and imitating real music. Fig.3 explicitly shows the difference in the effects between the two models, from which one can easily tell at which aspects the music generated by LeakGAN is better.

TABLE II. OBJECTIVE EVALUATION PERFORMANCE

Metrics	LeanGAN	LSTM	Type
Wilcoxon Test	**0.89105026**	0.60805510	CB
MWU Test	**0.86661874**	0.65296427	CB
KWH) Test	**0.89012412**	0.60737029	CB
SSP comparison	**0.96384517**	0.76039401	CB
Note-level mode test	**0. 87412102**	1	FB

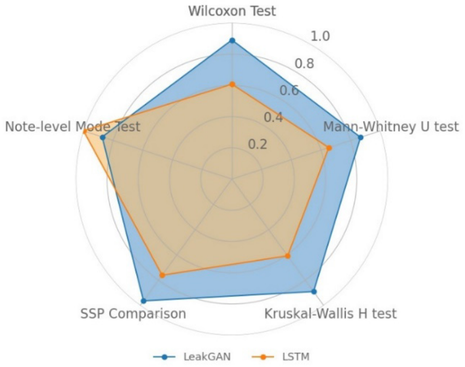

Figure 3. The comprehensive evaluation chart of LeakGAN and LSTM.

Fig.4 (a) shows the high accuracy of our model even at the very beginning, and Fig.4(b) is the training loss of D which reveals that there is a rapid decrease at the beginnings and then it saturates.

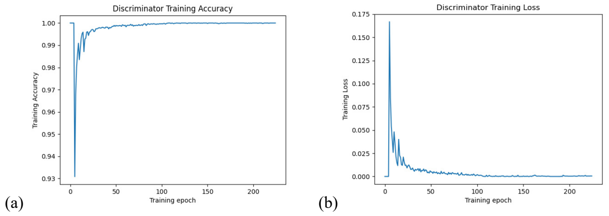

Figure 4: (a)The training accuracy of the discriminator, (b)the training loss of the discriminator.

IV. CONCLUSION

A novel generative method for melody sequence generation under the generative adversarial networks' framework is presented in this paper. Firstly, the MIDI music data has been represented into symbolic domain. And then, several objective evaluation metrics are designed to judge the quality of generated music. The results shows that the model is with high accuracy and the music created by it is of harmonic and rhythmic structure. In future work we hope to apply our model in longer music pieces and further improve the quality of the generated pieces.

ACKNOWLEGEMENTS

This work is supported by Research Funds of Sichuan Provincial Key Laboratory on Intelligent Terminals under grant SCITLAB-1021 and the Open Research Fund of Power System Wide-area Measurement and Control Sichuan Provincial Key Laboratory under Grant 2020JDS0027.

REFERENCES

[1] Feng Yin. Computer Music Technology[M]. Science Press, 2018..

[2] Bharucha, J. J., & Todd, P. M. (1989). Modeling the perception of tonal structure with neural nets. Computer Music Journal, 13(4), 44-53.

[3] Goel, K., Vohra, R., & Sahoo, J. K. (2014, September). Polyphonic music generation by modeling temporal dependencies using a rnn-dbn. In International Conference on Artificial Neural Networks (pp. 217-224). Springer, Cham.

[4] Eck, D., & Schmidhuber, J. (2002). A first look at music composition using lstm recurrent neural networks. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 103, 48.

[5] Agrawal, P., Kaushik, S., Banga, S., Pathak, N., & Goel, S. Automated Music Generation using LSTM.

[6] Yang, L. C., Chou, S. Y., & Yang, Y. H. (2017). Midinet: A convolutional generative adversarial network for symbolic-domain music generation. arXiv preprint arXiv:1703.10847.

[7] Jayawardena, D. I. D. D. (2019). Music generation for scene emotion using generative and CNN model (Doctoral dissertation).

[8] Jin, C., Tie, Y., Bai, Y. et al. A Style-Specific Music Composition Neural Network. Neural Process Lett 52, 1893–1912 (2020). https://doi.org/10.1007/s11063-020-10241-8

[9] Yu, L., Zhang, W., Wang, J., & Yu, Y. (2017, February). Seqgan: Sequence generative adversarial nets with policy gradient. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).

[10] Dong, H. W., Hsiao, W. Y., Yang, L. C., & Yang, Y. H. (2018, April). Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In Thirty-Second AAAI Conference on Artificial Intelligence.

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

[12] Shi, Z., Chen, X., Qiu, X., & Huang, X. (2018). Toward diverse text generation with inverse reinforcement learning. arXiv preprint arXiv:1804.11258.

[13] Yang Zhen. (2020). Research on Deep Learning Automatic Composition Based on MIDI Music (Master's Thesis, South China University of Technology.

[14] Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., & Wang, J. (2018, April). Long text generation via adversarial training with leaked information. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[15] Bachman, P., & Precup, D. (2015). Data generation as sequential decision making. Advances in Neural Information Processing Systems, 28, 3249-3257.

[16] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

[17] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. Advances in neural information processing systems, 28, 649-657.

[18] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[19] Wang, Z., Chen, K., Jiang, J., Zhang, Y., Xu, M., Dai, S., ... & Xia, G. (2020). Pop909: A pop-song dataset for music arrangement generation. arXiv preprint arXiv:2008.07142.

[20] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," 2010.

[21] R. V. Hogg, J. McKean, and A. T. Craig, Introduction to mathematical statistics. Pearson Education, 2005.