Alarm Sound Recommendation Based on Music Generating System

Wenhan Han[1,2(✉)] and Xiping Hu[1,2]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Beijing, China
{wh.han,xp.hu}@siat.ac.cn
[2] The Chinese University of Hong Kong, Sha Tin, Hong Kong

Abstract. In this paper, we propose an alarm sound recommendation system based on music generation. The recommendation system will be integrated with an application named iSmile, which is a sleep analysis and depression detection application built by the authors in previous work. We use a music generating algorithm based on GAN (Generative Adversarial Nets) as the core of the recommendation system. To the best of our knowledge, it is the first application recommending real-time generated music rather than existing music. In the following part of the paper, we detail the algorithm, the experiment we conducted and the result analysis. The result shows that the recommendation system can effectively generate and recommend proper alarm sound according to the emotion prediction.

Keywords: Music generation · Alarm sound recommendation

1 Introduction

"5G" is a popular word nowadays. It is expected to have explosive bandwidth increment and much higher efficiency [1]. From an economic point of view, as the technology and devices upgraded, the cost of using wireless network can be much reduced. "5G" must change people's lives from a lot of aspects. For examples, people's demand for WIFI can be greatly reduced, the devices around people can keep online all the time and applications can provide more creative services. "5G" means not only the improvement of hardware technology but also the applications on mobile platforms. Benefited from the "5G" network, applications driven by big data will be more common in people's lives.

Previously, we have conducted a research about the impact of digital alarm sound to human emotions and constructed an application named iSmile. ISmile includes an Android application and an alarm sound recommendation system on the cloud. The Android application is used to collect users' sleep data, provide alarm service and request the feedback of users. The cloud accepts user data, building context profiles of users and pushing proper alarm sounds to users.

V. C. M. Leung et al. (Eds.): 5GWN 2019, LNICST 278, pp. 86–95, 2019.
https://doi.org/10.1007/978-3-030-17513-9_7

In the previous work, the alarm sound recommendation system in the back-end mainly consists of two parts, an alarm sound library, and a sleep data analysis model. The alarm sound library uses music genre classification [18] to classify existing music, then the sleep analysis model analyzes the uploaded data and selects proper alarm sounds to push. Many sounds or music recommendation systems are constructed in this way, for example, Hu et al. proposed a music recommendation system in [13], which aimed at helping drivers adjust their emotions and making a safe driving environment. However, recommending existing music to users is not flexible enough to satisfy the users due to the diverse preferences of individuals. Now, we think about doing something new. To make the sounds recommended more personalized, We alter the back-end of the application from recommending existing alarm sound to generating new music.

After GAN (Generative Adversarial Network) was proposed [10], many derivations of GAN have sprung up. GAN is applied to various fields and achieves a lot of amazing results, for examples, seqGAN [22], IRGAN [19] and AE-GAN [16]. It can be used for generating pictures, texts, etc. Recently, applying GAN to music generating becomes a popular topic. Dong et al. proposed MuseGAN [6–8], a GAN model which can generate 4-bar multi-track music phrases. Before that, there are many trials to make computers generate music automatically. Wolfram used his theory "a new kind of science" [21] to generate music, but it is more like a sequence of chaotic notes rather than music, although it has some potential regularity. Sturm et al. used RNNs to generate monophonic melodies [17]. Hadjeres et al. also used RNNs, and proposed a model to generate four-voice chorales [12]. RNN-RBM [4] was able to generate polyphonic single-track pianorolls. Using hierarchical RNNs to coordinate three tracks, Song from PI [5] can generate a lead sheet with an additional monophonic drums track. C-RNN-GAN [14] were able to generate music as a series of note events. Compared with MuseGAN, all those previous works can only generate single-track music or have limitations in performance. Our model built in the back-end of iSmile is inspired by MuseGAN actually.

The contributions are as follows:

- We use a GAN-based model to alter the previous alarm sound library, which can generate new alarm sounds according to the user context information.
- We conduct experiments to evaluate the results of the new recommendation system.

In Sect. 2, we introduce some related works. We detail the method of building the recommendation system in Sect. 3. Section 4 is mainly about the experiments we conduct. Finally, we conclude the work we do in Sect. 5.

2 Related Work

2.1 GAN

GAN [10] consists of a generator and a discriminator. The generator tries to generate fake data to fool the discriminator, while the discriminator tries to

distinguish the fake data from the true data. The generator and the discriminator play a minimax game and compete with each other. Finally, the generator and the discriminator will both obtain better performances.

2.2 MuseGAN

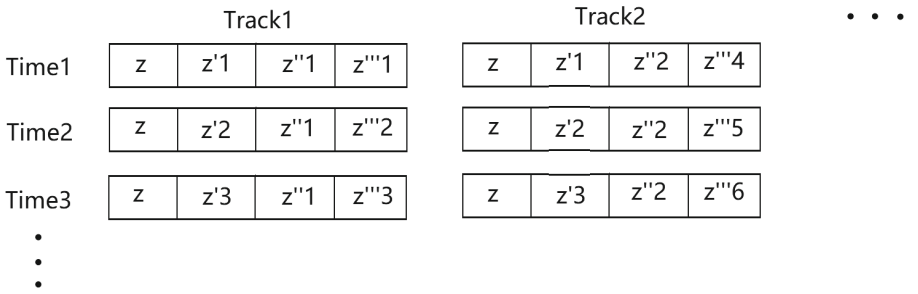

Fig. 1. The input format of the generator.

MuseGAN has a great performance in generating multi-track music. The model in [7] is designed to generate 5-track music with 4-bar length. Naturally, MuseGAN consists of two parts, a generator and a discriminator. In the generator, each track has a network to generate single-track music, which has the same network structure with each other. In order to simulate the regularity and randomness of music, the input of those 5 tracks is set to have 4 parts: the inter-track time-independent random vector, the intra-track time-independent random vector, the inter-track time-dependent random vector, and the intra-track time-dependent random vector. Figure 1 shows the format of the input, where the horizontal axis denotes the tracks, the vertical axis denotes the time steps, and the rectangles denote the input vectors.

2.3 iSmile

ISmile is the previous work of our team. It includes the front-end, an Android application, and the back-end, which is built on the cloud. Users can set the time when they will get up before they sleep using iSmile installed on their smartphones, then their smartphones will record and upload their sleep status data. Before the time when they are supposed to be woken up, the cloud will have analyzed their sleep data and push proper alarm sounds to their smartphones. The analyzer on the cloud is based on an approach named Emotion-Aware Smart Tips (EAST), which combines multivariate regression, random forest, and neural network to quantify the relations between sleep patterns and emotional states [11]. In the experiments we have conducted, the alarm sounds pushed by the cloud are selected from the existing music library.

3 Method

3.1 Model Design

Inspired by MuseGAN, we use an analogous network structure to generate alarm sounds. The alarm sounds generated by the network are 4-bar 8-track music. The input of each track in the generator is a 128-dimension random tensor, which is formatted as what mentioned in Sect. 3.2. With the time solution 24 (one beat), the output of each track in the generator is a $4 \times 96 \times 84$ tensor, where the 84 denotes 84 kinds of pitches. After merging whole 8 tracks' outputs as one tensor, it is input into the discriminator, whose output is a float between 0 and 1, indicating how it looks like the music created by human composers. Figure 2 shows the general structure of the whole model, where G denotes a single-track generator, and D is the discriminator. The model has 8 tracks, generating 8 music tracks of different instruments, which are drums, piano, guitar, bass, ensemble, reed, synth lead and synth pad. To save space, Fig. 2 just depicts 3 tracks. Figure 3 gives the detail about the single-track generator G. Similarly, to save space, the number of the layers is not the same as the figure. Actually, we use 7 layers of transposed CNN [9] to generate a single-track pianoroll.

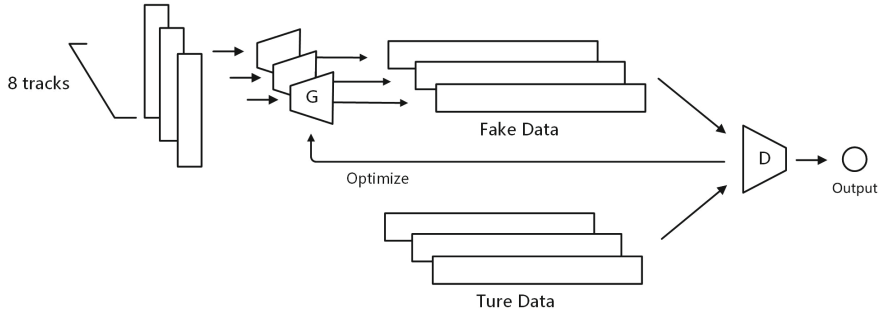

Fig. 2. The general structure of the GAN model.

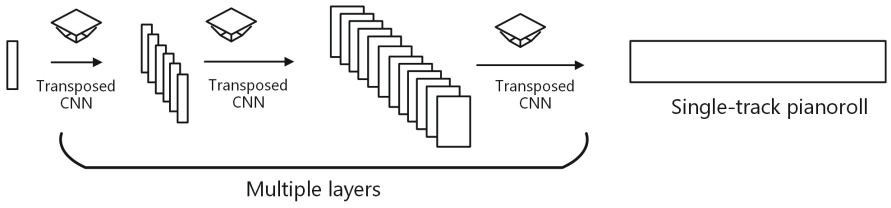

Fig. 3. The general structure of the single-track generator.

The whole generator acts as an interpreter. Before the generator, there is an encoding model, who encodes the result of emotion prediction made by the EAST system. The encoding model is formed with 3 layers of transposed CNN. It controls the inter-track time-independent random vector z in the input of the generator, which decides the base style of the whole music.

3.2 System Structure

In the previous work, iSmile has the structure as Fig. 4. All users have the same sound recommendation model. It limits the variety and personality of the sounds that are pushed to users' smartphones. Although the process of predicting emotions takes users' context information in consideration, the results of the sound recommendation system reacting to the same emotion predictions never change.

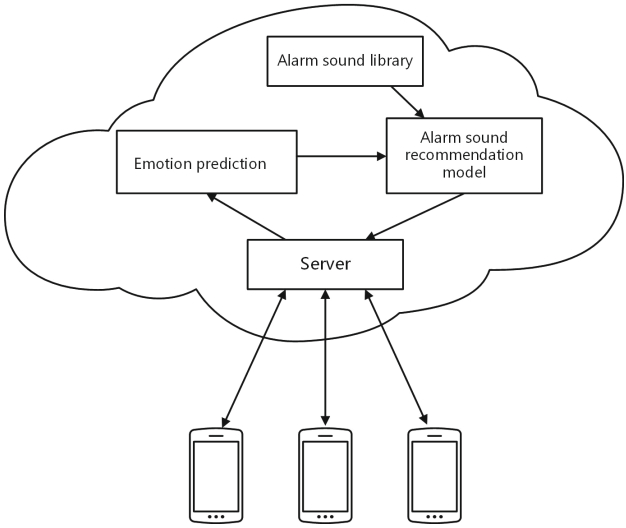

Fig. 4. Previous structure of iSmile.

After we apply the new recommendation system into iSmile, each user will have a private alarm sound generator. During users using iSmile, their own models will be trained to adapt to themselves. For example, a person who likes quietness will receive some light music when his emotion prediction result is looked happy, while another one may receive some passionate music although they have the same prediction result (Fig. 5).

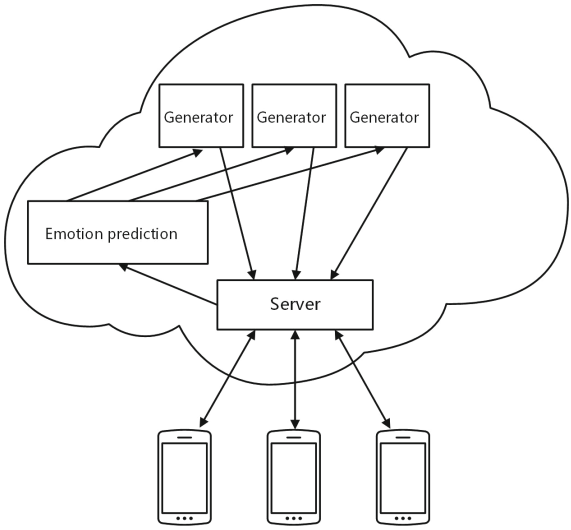

Fig. 5. New structure of iSmile.

4 Experiment

4.1 GAN Model Training

Original GAN is difficult to train. The reason is that when the support of the true set and the fake set generated by the generator is a low dimension manifold in high dimension, the possibility is almost 1 that the intersection measure of the true set and the fake set is 0 [2,3]. The performance of the generator and the discriminator must be in a relative balance. Excessively great performance of the discriminator will make the generator learn nothing. So we strictly control and dynamically adjust the optimization times of the generator and the discriminator in an iteration period when training the GAN model. The data set is Lakh Pianoroll Dataset built by Dong et al. [7,15].

4.2 Encoding Model Training

The purpose of the GAN model is to make the music generated be more like music created by human composers. The encoding model decides the style and emotion of the music. The result of the emotion prediction is a 2-D tensor, which includes arousal value and valence value (both are 0–10) as our previous work [11]. We set those two values to extremes respectively in order to evaluate the performance of the encoding model. We give feedback (arousal value and valence value) that how it fits the emotion we expect after the generator interprets the output of the encoding model. Policy gradient based REINFORCE learning [20] is utilized to train the encoding model. The encoding model adjusts the parameters according to the feedback we give, increasing the possibility of outputs that can make we give high scores.

4.3 Result Evaluation

Figures 6, 7 and 8 give the visualized result samples during training. To evaluate the authenticity of the results compared with the true music, we find 20 people to do a test. All test takers are found from the Internet by random, where 10 are male and 10 are female. Their ages are in range 18–24. We randomly selected 10 pieces of generated music and 10 pieces of true music from the dataset and mix them up, then ask our test takers to score for the authenticity of those pieces of music. Figure 9 shows the mean scores of those pieces of music, where the black points denote true ones and white points denote fake ones. The vertical positions of the points are randomly selected to avoid overlapping. Generally, the fake music and true music are not be distinguished clearly, but some generated pieces still get much lower scores than the trues.

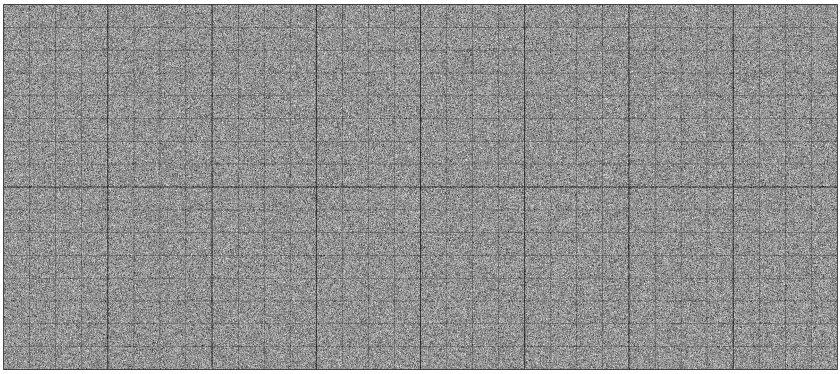

Fig. 6. The sample of results at the beginning

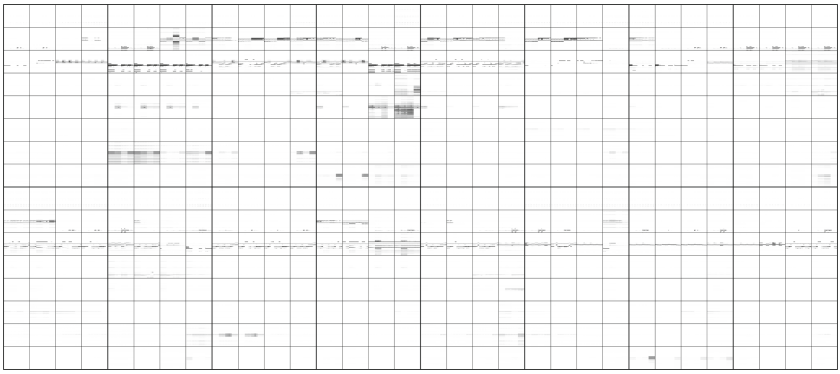

Fig. 7. The sample of results at the 3000th iteration

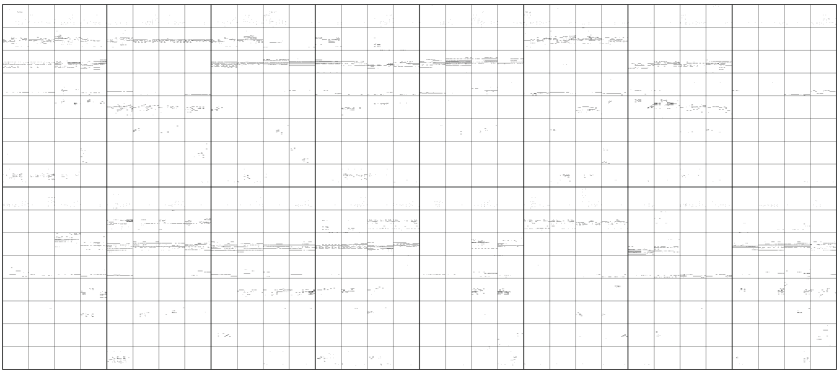

Fig. 8. The sample of results at the 20000th iteration

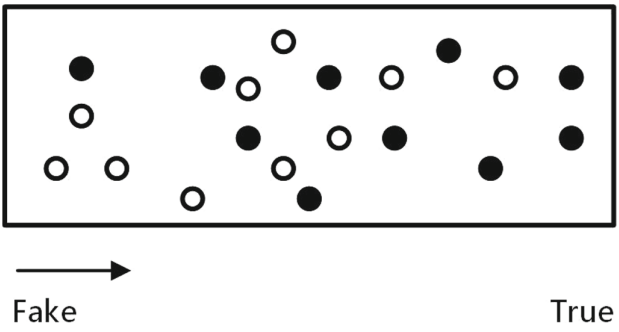

Fig. 9. The authenticity test

We also make a test about the emotion of the generated music. Arousal value and valence value are used to evaluate the emotion. We set both the arousal and valence values to 10 for the generator to produce music, and ask test takers to score for their moods and feelings after hearing these pieces of music. Figure 10 shows the mean scores, where black points are positive ones and white points are negative ones. To avoid overlapping, all positions of the points are handled by randomly jittering in range −0.5 to 0.5.

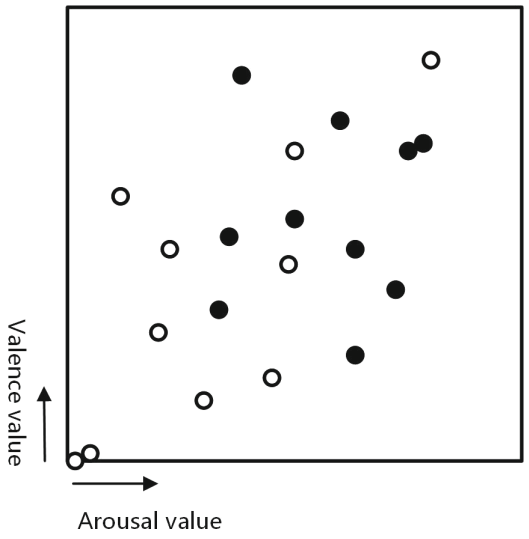

Fig. 10. The emotion test

5 Conclusion

We propose and construct an alarm sound recommendation system based on music generating. The music generating model combines GAN and REINFORCE learning. To the best of our knowledge, it is the first sound recommendation system recommending real-time generated music rather than existing music. The experiments we conducted show that the system has pretty good performance. However, the generated music still has a little distance to the true pieces, and the model is not very stable. Due to the limitation of time, the experiments are not very comprehensive. We need more user data to optimize and evaluate the recommendation system. More details will be shown in our full paper version soon.

References

1. Andrews, J.G., et al.: What will 5G be? IEEE J. Sel. Areas Commun. **32**(6), 1065–1082 (2014)
2. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862 (2017)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223 (2017)
4. Boulanger-Lewandowski, N., Bengio, Y., Vincent, P.: Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. arXiv preprint arXiv:1206.6392 (2012)
5. Chu, H., Urtasun, R., Fidler, S.: Song from pi: a musically plausible network for pop music generation. arXiv preprint arXiv:1611.03477 (2016)

6. Dong, H.W., Hsiao, W.Y., Yang, L.C., Yang, Y.H.: MuseGAN: demonstration of a convolutional gan based model for generating multi-track piano-rolls. In: Proceedings of International Society of Music Information Retrieval Conference (2017)
7. Dong, H.W., Hsiao, W.Y., Yang, L.C., Yang, Y.H.: MuseGAN: multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: Proceedings of AAAI Conference on Artificial Intelligence (2018)
8. Dong, H.W., Yang, Y.H.: Convolutional generative adversarial networks with binary neurons for polyphonic music generation. arXiv preprint arXiv:1804.09399 (2018)
9. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285 (2016)
10. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
11. Guo, Y., et al.: Poster: emotion-aware smart tips for healthy and happy sleep. In: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, pp. 549–551. ACM (2017)
12. Hadjeres, G., Pachet, F., Nielsen, F.: DeepBach: a steerable model for bach chorales generation. arXiv preprint arXiv:1612.01010 (2016)
13. Hu, X., et al.: SAFeDJ: a crowd-cloud codesign approach to situation-aware music delivery for drivers. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **12**(1s), 21 (2015)
14. Mogren, O.: C-RNN-GAN: continuous recurrent neural networks with adversarial training. arXiv preprint arXiv:1611.09904 (2016)
15. Raffel, C.: Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching. Columbia University (2016)
16. Shen, S., Jin, G., Gao, K., Zhang, Y.: AE-GAN: adversarial eliminating with GAN. arXiv preprint arXiv:1707.05474 (2017)
17. Sturm, B.L., Santos, J.F., Ben-Tal, O., Korshunova, I.: Music transcription modelling and composition using deep learning. arXiv preprint arXiv:1604.08723 (2016)
18. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Trans. Speech Audio Process. **10**(5), 293–302 (2002)
19. Wang, J., et al.: Irgan: a minimax game for unifying generative and discriminative information retrieval models. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 515–524. ACM (2017)
20. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3–4), 229–256 (1992)
21. Wolfram, S.: A New Kind of Science, vol. 5. Wolfram Media, Champaign (2002)
22. Yu, L., Zhang, W., Wang, J., Yu, Y.: SeqGAN: sequence generative adversarial nets with policy gradient. In: AAAI, pp. 2852–2858 (2017)