EECS 545 Project Progress Report: Tempo-Lite: Drum Beats Generated from Other Instruments

Congni Shi, Li Ye, Muye Jia, Jerry Cheng
{congni, yeliqd ,muyej, chengjry}@umich.edu

February 2, 2022

1 Problem Statement

In this project, the group would like to investigate some of the existing Deep Neural Networks (DNN) for music composition/generation. More particularly, the group would like to produce a drum beats generator model, which that the model would take the complete, previously recorded tracks from the other musical instruments, and try to generate a drum track sequentially based on these provided other tracks.

2 Significance

Comparing to the copious deep learning researches in vision, acoustic generative networks are less studied. The group believes that it would be a lucrative field since amateur composers may not be able to write out scores for all musical instruments — they are likely to be relatively more familiar to one or few musical instruments than others. In that case, our model can be a beneficial resource towards the amateur composers with their compositions. Moreover, since the aim of the group is to provide a computationally-light training model, it can be easily re-trained towards different musical instruments, even novel ones such as theremin or Kazoo with contemporary styles.

3 Related Works/Novelty

Although the field of music generation is not as well developed as the field of image generation, there are still some rather impressive works that can generate drum beats or music in general. The model that the group is most interested in is the MuseGAN [1] model. It is a multi-track sequential generative adversarial network. It the state-of-the-art temporal model at its time for symbolic music generation, and is one of the only known models for generating polyphonic music.

In this course project, we are expecting to utilize the Hybrid model in Figure 2 from MuseGAN. In this architecture, each segment GAN has one local Gaussian random vector input z_i and a global Gaussian random vector input z. z_i's are most likely different for each segment generator, and z would remain the same for each segment generators, serving as an artificial inductive bias for all generator to produce more style-aligned music for a single song. Then the GAN is expected to generate a segment of track using these two random vectors.

However, the goal of the MuseGAN model diverges from this course project, since it focuses on generating all tracks from scratch using a relatively large-scale GAN. In this course project, the main focus was to generate drum tracks from other instruments by using a lighter model. To adapt this objective, the team would first like to perform a classification/encoding task on the existing tracks from other instruments, and serve this encoded vector as the inductive bias to further facilitate each segment GAN. Moreover, the team is also expecting to explore different loss implementation and discriminator architectures to further simplify the model while remaining the same performance.

4 Proposed Method/approach

Inspired by the related reading materials, the group has decided that the generative networks of the project's model should also be a temporal model, which it would have information related to both the entire track and the local segments of the music for pattern recognition. A coarse architecture of the model is demonstrated in Figure 1.

4.1 Pre-Generator Ablations

First, the completed tracks of other instruments are collected and feed into a classification/encoding algorithm. There are many different music styles such as Jazz, Blues and Rock. Each of these styles has its certain nuances, phrases, techniques, and sounds which are associated with the style of drumming that accompanies it. thus, music genre classification is critical. The algorithm would produce a drum genre space that will be utilized as the global inductive bias vector z for the segmented generators. After reading some related materials upon music genre classifications [**?**, **?**, **?**, **?**], the group proposed three methods for the classification/encoding algorithm to implement this part of the network.

4.1.1 CNN Encoders

For the first option, the group would like to implement a conventional CNN encoder for the classification/encoding task. It is expected to be the most computationally intensive implementation but the most mature encoding structure available as of today. The group would probably inspect both the CNN encoder with freezed implementation during generation task or continuously fine-tuning the CNN encoder while training for the generation task.

4.2 Subspace Learning

For the second option, the group would like to test the classification task with the subspaces learning technique. Subspace learning should be much faster in inference comparing to the CNN encoder.

For the music genre classification task of this project, the team currently plans to first group the training data into different music genres, and then use SVD to obtain an orthonormal basis for each genre space. Some related training can be seen in [7]. After having learned the genre subspaces, classification can be done by comparing the projections of one particular test data onto the orthogonal complement of the genre subspaces.

4.2.1 Perceiver

In the last option, one of the student in the group would like to implement an attention-based model, namely the Perceiver [5]. Since the output space of the perceiver model would be a logits vector, it can also serve as a traditional classifier/encoder structure. This part of the project would potentially be adopted from one of the student's EECS-542, Advanced Computer Vision course project.

4.3 Discriminator Ablations

The group would also like to do a study upon the ablation of the discriminator architecture, which the group has two variations in mind.

4.3.1 No Discriminator

The most natural and intuitive idea the group has raised and would like to test on is completely removing the discriminator architecture from the original MuseGAN model. Thus instead of using a discriminator, the group would like to see if the model would solely work with L2-norm loss between the generated and the labels.

To evaluate the similarity between the generated vector and the vector in the data sample – i.e. to check whether the model-generated drum beats fit with provided music notes-we propose (as an initial proposition) to use Euclidean distance for measuring the similarity between two vectors. Specifically, the group is considering L1 or L2 distance metric. The error calculation works as the following:

$$J(w) = \sum_{i=1}^{N} \left\| G(\mathbf{z}) - x^{(i)} \right\| \tag{1}$$

where $G(\mathbf{z})$ represents generated drum beat vector, and $x^{(i)}$ represents the data sample in training data set. The discriminator will denote the generated drum beat as adequate as long as the above error is lower than a threshold value.

4.3.2 Traditional Discriminator

The second intuitive method that came to the group was to use a traditional discriminators in GANs [3] instead of the probabilistic Wasserstein distance discriminator in the MuseGAN model.

5 Evaluations

For evaluations, the group would compare the model against its own variations listed above in the Approach section in Section 4. The group would compare the performance of the model and analyze why certain variants would perform better than the others. The group would also like to compare the results against some other similar models that has been proposed by others if the time permits, such as the NeuralBeats [6] and other DNN based drum generations [4].

6 Appendix

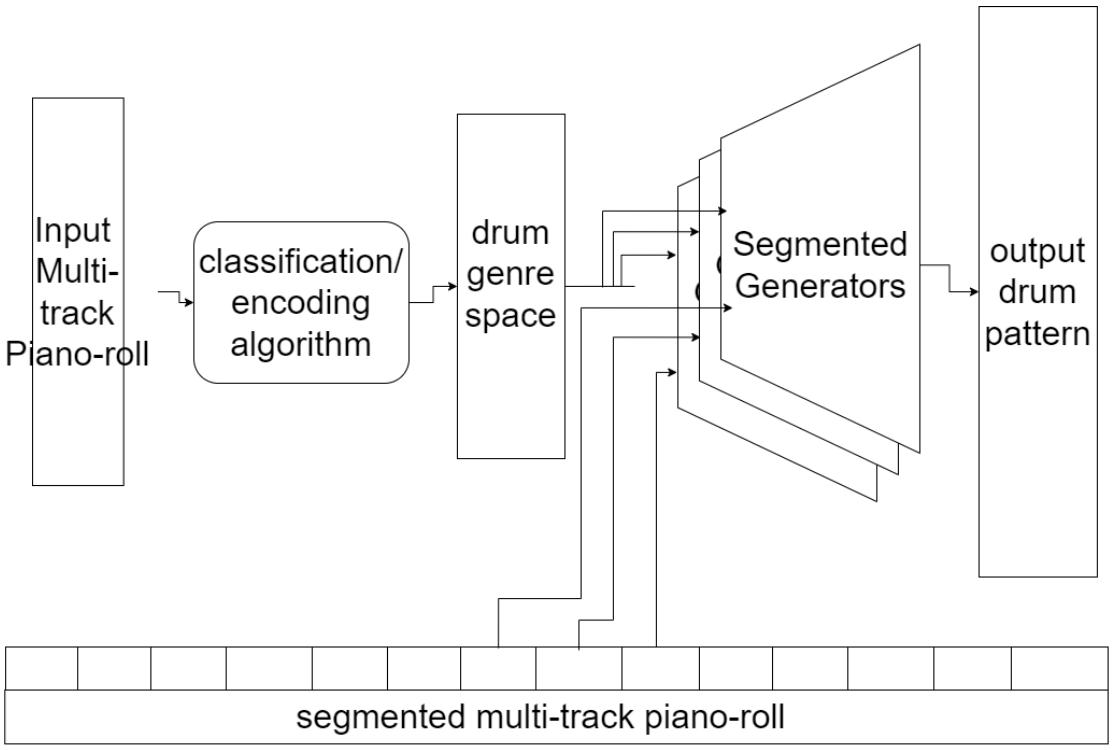
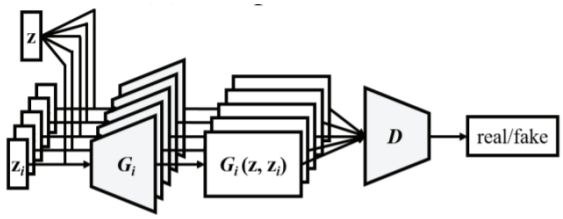


Figure 1: Course Architecture of the model



(c) Hybrid model

Figure 2: MuseGAN: Hybrid model

References

[1] 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, 2017.

[2] H.-W. Dong and Y.-H. Yang. Convolutional generative adversarial networks with binary neurons for polyphonic music generation, 2018.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[4] T. Huang. Neural networks generated lamb of god drum tracks, Mar 2019.

[5] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira. Perceiver: General perception with iterative attention, 2021.

[6] S. Nikolov. Neuralbeats: Generative techno with recurrent neural networks, Apr 2016.

[7] A. Ritchie, C. Scott, L. Balzano, D. Kessler, and C. S. Sripada. Supervised principal component analysis via manifold optimization. In *2019 IEEE Data Science Workshop (DSW)*, pages 6–10, 2019.