Ernesto Vargas Aguilar, Cong (Frank) Gu, Yunchan Clemence Lee
Yingyu Liang
CS760: Machine Learning
22 October 2018

Generating music using Multi-Categorical GAN

Generative adversarial networks are neural networks that generate their output via an adversarial game between a generator model and a discriminator model. Despite its success in the image domain, its applicability in sequential domain has been limited due to the lack of an efficient method to accurately capture and regenerate the temporal structures of the domain. For the current research, we intend to conduct a systematic study on genre-based generative models for music, by constructing 1. an efficient method to extract and encode rich variety of structural information in a music piece, 2. a generative adversarial network based on this encoding to generate high quality musical sequence, 3. an extension of this architecture to differentiate between music genre.

1 INTRODUCTION

1.1 Motivating problem

Consider a dataset of musical pieces, labeled by their respective genres. At high level, music is set of multiple sequences of notes, which contain information on its pitch, length of the note, intensity, and its relative temporal position with respect to the previous note [1]. Furthermore, the sequence of notes form a highly regularized structure that is dependent both on its short-term and long-term neighboring notes, within and between different sequences (instruments). Therefore it follows that, in order to generate music with high resemblance to real pieces, these information would need to be efficiently captured.

1.2 Background

Generative adversarial networks (GAN) are neural networks that generate their output via an adversarial game between a generator model and a discriminator model. At high level, the generator's job is to produce an artificial instance that resemble a pre-existing, real world dataset, and the job of the discriminator is to distinguish between artificial and real input instances. The architecture has seen particular success in the image domain. However, there are still limitations for GAN networks to train/output in sequential domains such as NLP [2] and music [3].

Unlike images, which can be modeled by a 3 dimensional array of bounded, continuous real values, where the first 2 dimensions denote the pixel position, and the 3rd dimension denote the color (RGB) information, sequential data contains a variable length sequence of typically discrete vectors, such as word vectors in NLP. Following traditional models for tackling such data, a number of variations of GAN architecture that replace or augment convolutional networks in GAN with RNNs have been proposed. However, the same issues that limit the effectiveness of traditional RNNs have been transferred to its application within GAN, namely how to capture the long-term and short-term structural information present in these domains. Furthermore, it has been found that typical training objectives, such as autoregression [4] used in training RNNs are inadequate when transfered to generative problems.

In an attempt to digitize and structuralize music, MIDI format has been in wide use since 1980s. Despite objection to its crudeness, the format effectively captures the sequential flow of polyphonice notes. However, as is with the case of word vectors in NLP, the format lacks the ability to summarize relations between different notes. A number of different approachs to extact such information out of MIDI files have been proposed. MusicVAE [5] is an autoencoder that learns underlying repeating short-term structures (loops), and long-term structures (progressions) via hierarchical decoder structure that control generation of short-term structures. In another work [6], a set of parallel RNNs, modeled after structure of convolutional networks, have been used to successfully generate polyphonic music. These works successfully demonstrate viability of using neural networks to generate acceptable musical sequences, possibly using multiple instruments.

On the other hand, Camino et al. [7] have recently proposed an extension of GAN architecture capable of handling multiple categories. In essence, the proposed architecture consists of multiple parallel generators, representing generator for each categorical variable, which is then concatenated using a variation of softmax to generate the final output.

2 CURRENT RESEARCH

For the purposes of the current research, we will systematically incorporate aspects of the mentioned previous works to construct a GAN capable of generating multi-categorical musical samples. In order to do so, we intend to approach the problem in two major steps.

2.1 Binary GAN based on MusicVAE

MusicVAE learns how to extract the latent structure within MIDI sequences and represent them as the intermediate data to be decoded. Furthermore they have shown that by tuning this structure, the autoencoder is capable of blending different sequences of music, and/or adding and removing specific type of structure. Taking advantage of this structure, we will attempt to construct a GAN that takes in the encoded structure as the input, in place of traditional MIDI sequences. As MusicVAE is available open source by the Magenta group [8], we plan to modify their code to be able to extract the latent vector intermediate as input for our neural network. For the neural network, we plan to test architectures proposed in previous GAN for music works, available open source [9], [10]

2.2 Extending to multi-categorical generator

Well-categorized music in MIDI have been provided through prior work [11], [12]. We will extend the binary model with a parallel discriminator structure with softmax similar to the work by Camino et al. [7] to train and distinguish between different genres of music. This will allow the network to generate music based on genre input.

3 VALIDATION

As far as we are aware, there exist no work that attempt to generate categorized music samples using machine learning. However, several metrics have been proposed to evaluate the quality of generated music in traditional GAN setting. One in particular is BLEU [13], which have used in place of log-likelihood score to quantify the quality [3]. We intend to use the same metrics to compare our model with several previous works, including SeqGan and 2 variations of models using parallel RNN [6], by training each model using dataset from single genre, as well as on existing benchmark used by SeqGan, the Nottingham [14] dataset. For the multi-categorical extension, we evaluate our model's performance on instances of the previous models trained on each genre separately.

REFERENCES

[1]O. Mogren, “C-RNN-GAN: Continuous recurrent neural networks with adversarial training,” *ArXiv161109904 Cs*, Nov. 2016.

[2]A. Graves, “Generating Sequences With Recurrent Neural Networks,” *ArXiv13080850 Cs*, Aug. 2013.

[3]L. Yu, W. Zhang, J. Wang, and Y. Yu, “SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient,” *ArXiv160905473 Cs*, Sep. 2016.

[4]F. Huszár, “How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary?,” *ArXiv151105101 Cs Math Stat*, Nov. 2015.

[5]A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music,” *ArXiv180305428 Cs Eess Stat*, Mar. 2018.

[6]D. D. Johnson, “Generating polyphonic music using tied parallel networks,” in *International Conference on Evolutionary and Biologically Inspired Music and Art*, 2017, pp. 128–143.

[7]R. Camino, C. Hammerschmidt, and R. State, “Generating Multi-Categorical Samples with Generative Adversarial Networks,” *ArXiv180701202 Cs Stat*, Jul. 2018.

[8]*Magenta: Music and Art Generation with Machine Intelligence: tensorflow/magenta*. tensorflow, 2018.

[9]O. Mogren, *Implementation of C-RNN-GAN. Contribute to olofmogren/c-rnn-gan development by creating an account on GitHub*. 2018.

[10]L. Yu, *Implementation of Sequence Generative Adversarial Nets with Policy Gradient: LantaoYu/SeqGAN*. 2018.

[11]“The Lakh MIDI Dataset v0.1.” [Online]. Available: https://colinraffel.com/projects/lmd/. [Accessed: 22-Oct-2018].

[12]C. Raffel, *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.

[13]K. Papineni, S. Roukos, T. Ward, and W. Zhu, “BLEU: a Method for Automatic Evaluation of Machine Translation,” 2002, pp. 311–318.

[14]“Nottingham.” [Online]. Available: http://www-labs.iro.umontreal.ca/~lisa/deep/data/. [Accessed: 22-Oct-2018].