Progress Report for CSS 586 Course Project: Modeling Latent Patterns in Music

Alex Kyllo
akyllo@uw.edu
University of Washington
Bothell, WA, USA

ABSTRACT

This report explores recent research in symbolic music modeling with deep learning and provides a progress report on a course project to train a generative model of classical and popular music on the MusicNet and Lakh MIDI datasets.

KEYWORDS

deep learning, sequence learning, generative modeling, recurrent neural networks, music, MIDI

1 INTRODUCTION

Machine learning models of music have interesting applications in music information retrieval and creative tools for musical artists and educators. Generative models can create accompaniments for music, transfer styles between clips of music, and even generate entirely new music. Music is challenging to model because it exhibits a complex hierarchy of recurring patterns and long-range temporal dependencies, and because both musical scores and performances have multiple possible digital representations.

Depending on the task, machine learning models of music may be trained on the audio signal itself, either in a time domain or a frequency domain representation, or they may be trained on a digital symbolic representation of music, the most common of which is MIDI (Musical Instrument Digital Interface) notation. MIDI is an encoding of music as streams of bytes in one or more tracks or channels, each representing a sequence of 128 possible pitch values, along with timing, pressure and instrument values. A music transcription model may transcribe an audio signal as a MIDI score, which can easily be converted into other symbolic representations such as sheet music for human performers to read from, while a synthesizer model can convert MIDI representations into audio signals. A generative music model can be either trained to generate raw audio, or trained to produce a score that must be played by either a synthesizer or by humans to produce an audio music performance.

2 RELATED WORK

The state of the art in music generation has a long way to go before it can consistently generate music scores or performances that would be enjoyable and popular to listen to. There are a number of interesting, recent papers and projects in the area of music machine learning, several of which focus on learning long-term dependencies in music to produce longer clips with coherent structure.

Google's Magenta is an umbrella project for music deep learning research and development of software tools to expose these models for use by creative artists and students.

MusicVAE, part of the Magenta project, is a variational LSTM autoencoder for MIDI that incorporates a novel hierarchical structure using a "conductor" recurrent layer in its decoder model to better capture structure at multiple levels and avoid the problem of "posterior/mode collapse" whereby a generative model learns to ignore its latent code and rely on autoregression [12].

Another Magenta model called Music Transformer is a generative model that borrows its approach from the Natural Language Processing (NLP) domain, using a self-attention network to model MIDI music as a sequence of discrete tokens with relative positional dependencies [5]. This model was trained on the MAESTRO dataset of Piano-e-competition performances [4] and its generated piano music received favorable qualitative (Likert scale) ratings from human listeners [5].

MuseGAN [3] is an application of Generative Adversarial Networks to polyphonic MIDI music generation, trained on four-bar phrases of a multi-track pianoroll representation of rock songs from the Lakh Midi Dataset [10].

A major advantage of working with the symbolic representation of music is that it is of far lower dimensionality than the raw audio waveforms of a recorded performance, which makes it less computationally expensive. However, there are many aspects of musical performance that are not captured by a symbolic representation, so the expressiveness of symbolic generative models is constrained [6].

Other research has focused on modeling raw audio waveforms directly. WaveNet is a causal convolutional neural network for generating raw audio waveforms, developed by Google DeepMind, which achieves state of the art performance in generating natural sounding speech from text, but is also capable of generating short, realistic snippets of audio music [8].

Another model named SampleRNN generates raw audio waveforms using a three-tier hierarchy of gated recurrent units (GRU) to model recurrent structure at multiple temporal resolutions [7].

Jukebox by OpenAI utilizes a vector-quantized variational autoencoder (VQ-VAE) to compress raw audio into a sequence of discrete codes and models these sequences using autoregressive transformers to generate music of various popular genres, including singing and lyrics [1].

Audio data can also be modeled in the frequency domain through the use of Fourier analysis. The recent MelNet model is trained on spectrograms and can learn musical structures such as melody and harmony and variations in volume, timbre and rhythm [14].

Prior work points out that the division between symbolic music notes and the sounds of music is analogous to the division between symbolic language and utterances in speech, which may inspire ideas for combining the two approaches [4]. A paper from Boston

University describes an effort to combine the symbolic and waveform approaches to music modeling, by training an LSTM to learn melodic structure of different styles of music, then providing generations from this model as conditioning inputs to a WaveNet-based raw audio generator [6].

3 PLANNED METHODS

This research project will focus on the generative modeling of symbolic music using MIDI data as inputs, because of the advantages of symbolic music models in representing long-term structure in musical compositions to produce generations with coherent structure and use of repetition over long time scales.

3.1 Datasets

Two primary datasets will be used for this research project: MusicNet, which is a collection of 330 freely licensed European classical music recordings with aligned MIDI scores [13], and the Lakh MIDI Dataset, which includes a collection of 45,129 Creative Commons licensed MIDI files of popular music songs that have been matched and aligned to MP3 recordings, and of which we will use a subset for model training and evaluation [10].

3.2 Data Preprocessing

Several choices must be made in how to preprocess binary MIDI files into training examples for a neural network. There are multiple open-source Python packages capable of reading MIDI files into NumPy array-based representations, such as pretty_midi [11] and Pypianoroll [2]. In order to accommodate polyphonic music, we will convert each MIDI file into a pianoroll representation, wherein each instrument track is a sparse matrix that multi-hot encodes the velocity values for each of 128 possible pitch levels at each timestep. Because songs are typically at least a few minutes long and of varying length, it will not be feasible to train with entire songs as examples, so we will crop songs into phrases of equal numbers of measures, to use as training data.

The result of this preprocessing is that each training example will be a 3D tensor of shape (tracks x notes x ticks) and stacking the training examples will produce a 4D tensor.

Data augmentation is also possible and we plan to test its impact on our results–the literature suggests augmentation via pitch shifting the entire training example up or down by up to six semitones, and increasing or reducing the speed by up to 10% in order to create additional training examples and reduce overfitting [9].

3.3 Model Fitting

We will explore several modeling approaches to generating symbolic music:

- Sliding window sequence prediction with RNNs (LSTM/GRU)
- Sliding window sequence prediction with Transformers
- Latent space interpolation with Sequential VAEs
- Latent space interpolation with Sequential GANs

3.4 Model Evaluation

Evaluation of generative models is challenging because there is no equivalent of an accuracy metric like what is used in supervised learning. For autoencoder models we can measure how accurately the model can reconstruct its own inputs, but this does not tell us the quality of the interpolated examples. Generative models are typically evaluated using a combination of qualitative metrics whereby human judges rate the quality of the generated examples (essentially a Turing test), and quantitative metrics that assess the differences in the parametric distributions of generated and real examples. Yang and Lerch (2020) proposes a set of metrics informed by music theory, for probabilistically evaluating how similar the generations are to known sample distributions of real music [15]. These metrics include counts, ranges, histograms and transition matrices of pitches and note lengths, then the Kullback-Leibler divergence and overlapping area of the probability density functions are use to compare against known reference distributions per musical genre [15]. Due to the cost and time requirements associated with designing a human subjects experiment, we plan to utilize this quantitative approach to generation quality assessment.

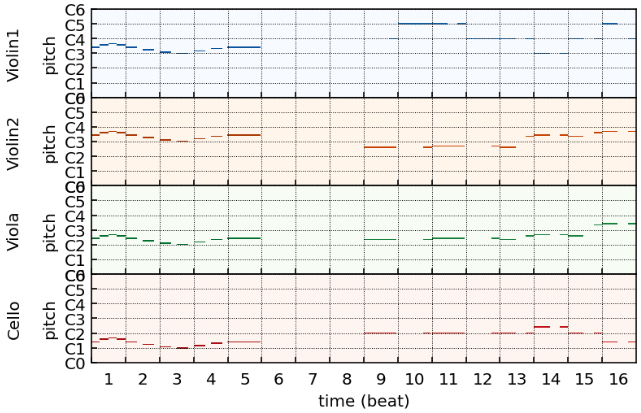

Figure 1: Pianoroll visualization of the first four measures of Beethoven's Serioso String Quartet

4 PROGRESS AND REMAINING WORK

REFERENCES

[1] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A Generative Model for Music. arXiv:2005.00341 [eess.AS]

[2] Hao-Wen Dong and Wen-Yi Hsiao. 2018. Pypianoroll: Open Source Python Package for Handling Multitrack Pianoroll. (2018), 2.

[3] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. 2017. MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. arXiv:1709.06298 [eess.AS]

[4] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. arXiv:1810.12247 [cs.SD]

[5] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2018. Music Transformer. (2018). arXiv:1809.04281 http://arxiv.org/abs/1809.04281

[6] Rachel Manzelli, Vijay Thakkar, Ali Siahkamari, and Brian Kulis. 2018. Conditioning Deep Generative Raw Audio Models for Structured Automatic Music. *CoRR* abs/1806.09905 (2018). arXiv:1806.09905 http://arxiv.org/abs/1806.09905

[7] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. 2017. SampleRNN: An Unconditional End-to-End Neural Audio Generation Model. (2017). arXiv:1612.07837 http://arxiv.org/abs/1612.07837

[8] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. (2016). arXiv:1609.03499 http://arxiv.org/abs/1609.03499

[9] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. 2018. This Time with Feeling: Learning Expressive Musical Performance. (2018). arXiv:1808.03715 http://arxiv.org/abs/1808.03715

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

[11] Colin Raffel and Daniel P.W. Ellis. 2018. Intuitive Analysis, Creation and Manipulation of MIDI Data With pretty_midi. (2018), 2.

[12] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. In *Proceedings of the35th International Conference on Machine Learning*. 10.

[13] John Thickstun, Zaid Harchaoui, and Sham Kakade. 2017. Learning Features of Music from Scratch. arXiv:1611.09827 [stat.ML]

[14] Sean Vasquez and Mike Lewis. 2019. MelNet: A Generative Model for Audio in the Frequency Domain. arXiv:1906.01083 [eess.AS]

[15] Li-Chia Yang and Alexander Lerch. 2020. On the evaluation of generative models in music. 32, 9 (2020), 4773–4784. https://doi.org/10.1007/s00521-018-3849-7