Musical Deep Learning

Nhut Minh Phan
phan92@uw.edu
University of Washington
Bothell, WA, USA

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

ABSTRACT

This paper presents a generative model of multi-instrument symbolic music.

KEYWORDS

deep learning, neural networks, sequential models, music

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 (where 0 is the lowest and 127 is the highest), along with timing, pressure and instrument identifier 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 trained either to generate raw audio, or to produce a symbolic score that must be played by a synthesizer or by humans to produce an audio music performance. This project focuses on the latter type of modeling: generative modeling of symbolic (MIDI) music to compose original musical scores.

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 for humans to listen to, but a number of recent research projects have shown promising progress in this area.

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 Long Short-Term Memory (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 [13]. This model is trained on 16-bar paragraphs of music and is capable of generating new melodies that blend two given melodies via latent space interpolation.

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]. The focus of this model is on learning long-term dependencies in music to produce longer clips of music with coherent structure. Music Transformer was trained on a dataset of Piano-e-competition performances [4] and its generated piano music received favorable qualitative (Likert scale) ratings from human listeners for its resemblance to human-composed music [5].

MuseGAN [3] is an application of Generative Adversarial Networks (GAN) to polyphonic MIDI music generation, trained on four-bar phrases of a multi-track pianoroll representation of rock songs from the Lakh Midi Dataset [11]. Like MusicVAE, MuseGAN includes a two-level generator that first samples latent codes at the phrase or bar level, then generates notes within the bars, to produce longer-term structural patterns.

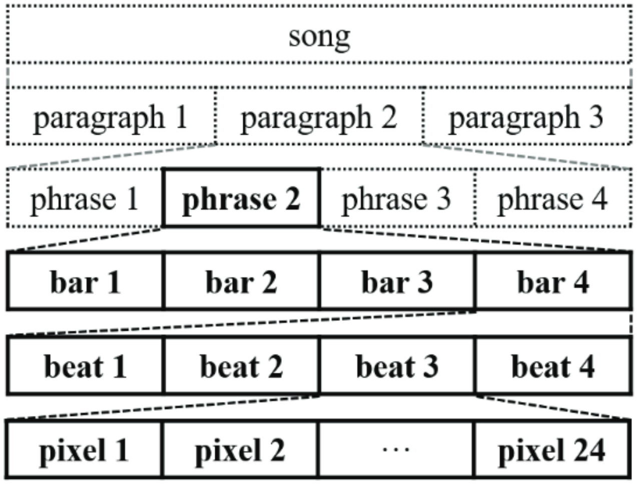

Figure 1: Diagram of the hierarchical structure of a musical composition, from the MuseGAN paper [3].

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 stylistic aspects of musical performance that are not captured by a symbolic representation, and may be specific to a particular performer, so the expressiveness of symbolic generative models is limited in comparison [7].

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 [9]. 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 [8].

Prior work points out that the division between symbolic music notes and music performances, 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 [7].

Modelling raw audio waveforms is challenging since a single second in modern recording spans thousands of timesteps. Most commonly music is recorded at 44.1 kHz (44,100 samples per second). Since a piece of music is at least a few seconds in length, capturing long-range dependencies is difficult in time domain. A model named MelNet was introduced to address this limitation of time-domain models [15]. MelNet leverages a two-dimentional time-frequency representation of audio called spectrogram, which reduces the dimentionality of the audio waveforms. This model was able to generate high-fidelity audio samples using structures at long time scales.

3 METHODS

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 [14], 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 plan to use a subset for model training and evaluation [11]. Including this second dataset may help to generalize the modeling approach beyond European classical music to include other popular genres and associated instruments.

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 that assist with the process of reading MIDI files from their binary on-disk representations into Python objects, such as pretty_midi [12], Pypianoroll [2] and music21 [1].

In order to accommodate polyphonic music, we convert each MIDI file into a pianoroll representation, as visualized in Figure 2, 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. To reduce dimensionality, I will clip the note pitch values to a narrower range, excluding the rarely used notes in the extreme high and low octaves, and encode chords (combinations of notes played simultaneously on the same instrument) as distinct tokens, so that they can be one-hot encoded

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 ticks x pitches) and stacking the training examples will produce a 4D tensor.

While we plan to model music with multiple instrument tracks, we anticipate the need to model only a fixed selection of instrument parts, similar to how MusicVAE models three-part (drum, bass and melody) [13] and MuseGAN models five-part (drum, bass, guitar, string, piano) arrangements.

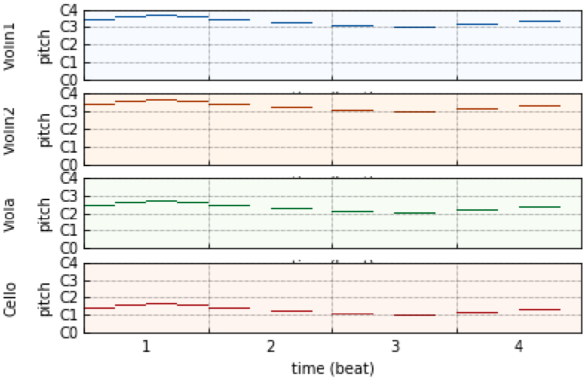

Figure 2: Pianoroll visualization of the first measure of Beethoven's Serioso String Quartet

Data augmentation is also possible and we assess its impact on the results–the literature suggests augmentation via pitch shifting each 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 [10].

The multi-stream representation discussed in [6] addresses the issue of extreme class imbalance and sparsity in the pianoroll representation.

3.3 Model Design

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 [16]. 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 used to compare against known reference distributions per musical genre [16]. Due to the cost and time requirements associated with designing a human subjects experiment, we utilize this quantitative approach to the quality assessment of generated samples.

4 RESULTS

5 DISCUSSION

REFERENCES

[1] Michael Scott Cuthbert and Christopher Ariza. [n.d.]. music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data. ([n. d.]), 6.

[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:eess.AS/1709.06298

[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:cs.SD/1810.12247

[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] Harish Kumar and Balaraman Ravindran. 2019. Polyphonic Music Composition with LSTM Neural Networks and Reinforcement Learning. arXiv:cs.SD/1902.01973

[7] 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

[8] 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

[9] 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

[10] 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

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

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

[13] 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.

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

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

[16] 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