MAIS202 - Deliverable 1 - Data Selection Proposal

Maya Scott-Lourenço

October 5th, 2020

1 Dataset

Dataset [1] will be MIDI files from the GBA/NDS era Pokémon games' original soundtrack. In total there are 837 tracks, approx. 1 minute on average, which totals just under 14 hours of Pokemon tunes a model can be trained on.Dataset could be increased to 25 hours by including the more recent mainline games and 40+hours including spin-offs, but would require converting waveforms to MIDI and dealing with the many resulting errors.

2 Methodology

2.1 Data Preprocessing

All of the data available in the MIDI files is useful for polyphonic music generation. A directory of MIDI files will be converted into "NoteSequences", a Google protocol buffer helpful for working with non-waveform file types, then "SequenceExamples" are created that contain a set of inputs and labels representing a polyphonic sequence from the set of "NoteSequences". This data is then separated into a training and evaluation set and ready for the model.

2.2 Machine learning model

The goal of the project is to generate new polyphonic Pokémon music using the Google Magenta library model PolyphonyRNN. [2] It uses Recurrent Neural Networks (RNN) with cells implementing Long-Short-Term-Memory (LSTM) to improve the RNN's ability to learn longer-term structures. This is of importance as music tends to have recurring sections and structures. Computing will be done via Google Colabs for that delicious computing power. A Concurrent Neural Network (CNN) approach like PixelCNN [3] is possible but would require representing MIDI as images and naive CNNs don't compute long range dependencies well. There is evidence to suggest that CNNs with locally connected convolutional layers could perform better than RNNs [4] but there are no available models for efficient implementation.

2.3 Metrics

Evaluating the performance of the models in polyphonic music generation is hard because music quality is subjective. There isn't a way to find an objective metric but we could look at the loss per epoch to evaluate under/overfitting.

2.4 Final Conceptualization

The music generated by the model would be integrated into a simple webapp where the user is presented with 4 music samples and need to choose which 2 were produced by AI. They can't keep playing until there's no more songs. This data would be collected (i.e how often sample-1 was thought to be an official song) and used to validate the performance of the model. I'd love it if the ML produced songs were thought to be real 60% of the time.

References

[1] MIDI files of the Pokemon FRLG, RSE, DPPt, BW, HGSS OST. URL `https://drive.google.com/file/d/1n4aIi1KS4DPMjMMaz`$_U$$jqRKyjFKwKWOY/view?usp=sharing$

[2] Magenta Team Google Brain *"Polyphony RNN, revision ca73164,"* URL `https://github.com/magenta/magenta/tree/master/magenta/models/polyphony_rnn`, 2016.

[3] Tim Salimans and Andrej Karpathy and Xi Chen and Diederik P. Kingma, *PixelCNN++: A PixelCNN Implementation with Discretized Logistic Mixture Likelihood and Other Modifications*, ICLR, (2017). URL https://arxiv.org/abs/1701.05517v1

[4] Zhihao Ouyang, Yihang Yin, Kun Yan, Jian Wu, Xiaolin Hu, Shu-Tao Xia, *Music Generation with Local Connected Convolutional Neural Network*, (2018). URL http://ouyangzhihao.com/wp-content/uploads/2018/12/MUSIC-GENERATION-WITH-LOCAL-CONNECTED-CONVOLUTIONAL-NEURAL-NETWORK.pdf