Polyphonic Music Generation from MIDI Performances Using RNNs

Sean Metzger (seanmetz@stanford.edu) & Daniel Gallegos (dgabot@stanford.edu)

Project Goal and Motivation

Composing music is one of humanity’s most amazing accomplishments. We thus sought to generate high quality classical music using an LSTM. Although many LSTMs can generate decent music, we sought to create an LSTM that could not only support multiple notes at once, but also encode dynamics, making the music sound less robotic and more human generated.

We trained our LSTM on 100-note sequences of classically played music with dynamics generated by musicians on the Yamaha E-Piano Competition Dataset. The database contains performance of the greatest classical pieces ever by Bach, Chopin, Beethoven and more.

Encoding MIDI Files

Each vector x represents a single time step from our data. Each vector y represents the desired output for one step of the RNN.

Choosing an Encoding

Choosing the right encoding is critical to our neural network’s performance. We allowed each note to have a different encoding, which enabled the network to learn the different characteristics of each note. This approach, known as “one-hot encoding,” was effective in capturing the nuances of each note in the music.

Representing Polyphony

Each network's initial weights were randomized. This allowed the network to start with a diverse set of initial conditions, which in turn led to a more diverse set of generated outputs. This approach was successful, as the network was able to generate music that was both accurate and expressive.

Results and Future Steps

Listening to our results & post-processing our model's output

1. Modifying the encoding so that we can create notes that are held for multiple beats. This can be combined with a little post-processing to create notes that can take any length.
2. Modifying the encoding to create notes that can play more loudly and softly. This combined with the first step can be used to create a program that can generate the full extent of sounds able to be played on the piano.
3. We could alter the loss function to take into account harmonies and phasing a note that doesn’t cause dissonance. This would help the model more true and generate music that more closely imitates pianos from the baroque, classical, and romantic eras.
4. Create a more cohesive dataset so that it can generate music better of a more specific type instead of general "classical" music.

References & Acknowledgements

1. https://huggingface.com/gpt2
2. https://www.tensorflow.org/api_guides/python/tf
3. https://www.tensorflow.org/api_guides/python/tf
4. https://www.tensorflow.org/api_guides/python/tf
5. https://www.tensorflow.org/api_guides/python/tf
6. The Stanford golf golf

Acknowledgements

CS 230 – Winter 2018 – Stanford

seanmetz@stanford.edu and dgabot@stanford.edu