Generative Improv-Theremin

CSCI1470 - Reflection

Jaehyun Jeon, Junewoo Park, Min Jean Cho, Oh Joon Kwon
{jjeon5, jpark49, mcho5, ok1}@cs.brown.edu

Introduction.

Generative Improv-Theremin is a creative deep learning project for generating sequentially and temporally relevant music from human agent interaction. We plan to modify a published model from Google called thee Piano Genie.

Piano Genie(https://magenta.tensorflow.org/pianogenie) is an intelligent controller which allows non-musicians to improvise on the piano. Using an unsupervised learning approach and a bidirectional RNN architecture, the model's encoder learned to map 88-key piano sequences to 8-button sequences, and its decoder learned to map the button sequences back to piano music. At performance time, the user's input replaces the encoder's output. Then the decoder is evaluated in real-time.

Another unique aspect of our implementation is that a camera-based hand segmentation system will be used instead of the 8 button controller such that the user input is through hand motion instead of a button controller. The hand segmentation system will output the same one-hot 1-by-8 vector that serves identical purpose as the 8 button controller.

Challenges.

Since we had no prior knowledge in bidirectional stacked LSTM architecture, we had to spend a considerable amount of time understanding the model. Moreover, quantization was challenging to understand since the paper did not provide any mathematical formulations. From the released code from Github, we still struggled to understand TF 1.x implementations (nested `with tf.variable_scope`). The paper defines three different losses, namely reconstruction (cross-entropy loss), marginal, and contour loss. While reconstruction loss is easy to implement, marginal and contour losses could be challenging to implement. The biggest challenge would be translating TF 1.x architecture into Keras based TF 2.x architecture.

Other than the architecture, we also needed to worry about user interface, as the image segmentation could not be as efficient as we initially thought. We may need to change the interface to keyboard (discussed in Plan section). The data processing part is also problematic as well, as we will need to preprocess MIDI data type. We plan to overcome this challenge by using Google's Magenta MIDI data processing utilities, along with the Maestro dataset used in the paper and other music generation projects by the Magenta group.

Insights.

We had to spend some time understanding the architecture, which were done in Google collab (https://colab.research.google.com/drive/1GWR9gUzJhl_l8icNJt8tgtna1SaptZx2). We are still in the process of completing the model, which we plan to do so soon, as we now have some understanding of the model. The model written so far is on the project Github (https://github.com/pulpiction/CS1470-Final-Project-J5). We will be testing the model with MAESTRO dataset once the model is completed.

Plan.

We will need to dedicate our time into completeing the architecture and fine-tuning the model with the preprocessed dataset. We may switch from our initial hand segmentation system to keyboard input of 1-8.