Generative Improv-Theremin

CSCI1470 - Final Project

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.

The two options we are considering in re-implementation are:

1. Create a Transformer based architecture that completely replaces the bi-directional RNN implementation of the paper.
2. Re-implement the model using PyTorch instead of Tensorflow

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.

Related Work

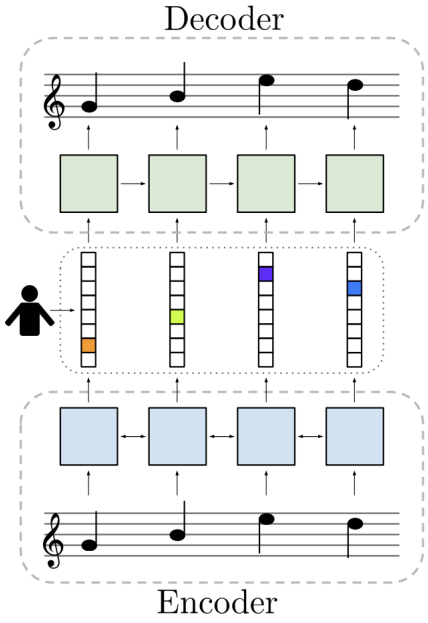
Piano Genie receives real-time user-provided time-dependent sequence of 8-buttons to create melodies that respect rhythm and melodic contours. Simply speaking, it learns a nondeterministic mapping from 8-buttons to 88-keys on the keyboard from unsupervised data set.

Some prior works include manipulation of contours using pre-programmed melodies, non-real time user-controlled generator that generates a whole song from melodic contours, and supervised learning of melodies from labeled data (e.g. gestures of professional musicians).

Data

Dataset used in the project will be drawn from International Piano e-Competition. The competition website (http://www.piano-e-competition.com) provides MIDI file submission from the past competitions. Because of the structure of the model, we will only be using this dataset for training.

Methodology



The architecture of the Piano Genie model is as follows: LSTM Encodere and LSTM Decoder. For each input piano note, the encoder outputs a real-valued scalar which is then discretized by quantizing it to k = 8 buckets equally spaced between 1 and 1. The decoder is very similar to a language model in which the next note is predicted from a previous note. Since we are aiming for a realtime improvisation, in order to allow the network to factor in timing into its predictions, we add in a ΔT feature to the input, representing the amount of time since the previous note quantized into 32 buckets evenly spaced between 0 and 1 second.

Piano Genie paper suggests the combination of reconstruction, marginal, and contour losses. Let us denote each loss as L_R, L_M, and L_C, respectively. L_R is simply the conventional model entropy loss on the approximated probability distribution of the model. L_M and L_C come from the quantization method, specifically Integer Quantization Autoencoder (IQAE). L_M ensures that the output will be mapped in $[-1, 1]$. L_C compares the finite differences of the inputs and of those of the encoder outputs. This encourages the encoder to produce output contours that match the input contours (increasing tonal inputs should produce increasing tonal outputs). This loss will be the primary objective function to be optimized.

$$L = L_R + L_M + L_C$$

$$\begin{aligned} L_R &= -\sum_{\mathbf{x}} \log \mathbb{P}_{\text{dec}}(\mathbf{x} \mid \text{enc}(\mathbf{x})) \\ L_M &= \sum_{\mathbf{x}} \max(|\text{enc}_s(\mathbf{x})| - 1, 0)^2 \\ L_C &= \sum_{\mathbf{x}} \max(1 - \Delta\mathbf{x} \cdot \Delta\text{enc}_s(\mathbf{x}), 0)^2 \end{aligned}$$

Metrics

Qualitative

The paper suggested the following three reasonable criteria for evaluating user experience.

1. Did you enjoy the experience of performing this instrument?
2. Did you enjoy the music that was produced while you played?
3. Were you able to control the music that was produced?

We also plan to compare our model with other models such as deterministic major scale model or models that do not consider melodic contour controlled by users unlike Piano Genie.

Quantitative

As suggested in the paper, we plan to measure the performance of the model in the following ways: model perplexity, ratio of contour violations, and latency. We can view music as a generalization of language model, so perplexity would be a good quantitative measure

of success. The model perplexity would be simply measured by $\exp(L_R)$. Ratio of contour violation measures the proportion of timesteps when the signs of input interval and the output interval disagree (we want the model to produce an increasing sequence when provided with an increasing sequence of inputs).

The quantitative measurement will also be focusing on comfortable user experience, specifically in terms of latency. We define user latency as time discrepancy between user input and model output. According to papers on human auditory perception (https://mp.ucpress.edu/content/36/1/109), people can perceive difference between 10 ms and 20 ms, but not between 0 ms and 10 ms. We plan to implement model with less than 10 $\pm$3 ms user latency. That is, time spent in forward pass for a single test input should take less than 10 $\pm$3 ms, including other systematic implementations such as interface.

Ethics

We will be using a public dataset as described above; currently there is no concern on ethical issue of the dataset. Additionally, the model architecture aims to be minimal and performance-centric in order to decrease the user latency. While it is possible that we may be consuming some significant amount of energy in training and hyperparameter testing, we will aim primarily to decrease environmental impact in terms of power usage.

Often times, improvisation is an area that differentiates musicians from one another. The musical interpretation and representation inside each musician's mind is the unique factor that allows such different expressions. However, the user input of our improv-theremin only considers the operational uniqueness of each user. There is only one machine-learned representational interpretation of music that is applied to all users. So an interesting question that this project suggests is whether this can be considered true improvisation or not.

Division of Labor

`jjeon5` : Image Segmentation / Piano Genie Architecture
`jpark49` : Piano Genie Architecture / Quantization
`mcho5` : Piano Genie Architecture / Data Collection
`ok1` : Piano Genie Architecture / Data Cleaning