MAIS202 - Deliverable 3 - Final Results

Maya Scott-Lourenço

November 13th, 2020

1 Final Training Results

1.1 Comparison to Preliminary Results

Data was processed differently from the preliminary model, in an attempt to respect Colab's RAM limitations (lmfao) and increase the dataset. Instead of moving all of the different instrument tracks to the piano track, I chose to only map the first 4 and delete the rest. This still poses the same problem of not all files retaining the main melodic structure of the original music [**Figure 1**].

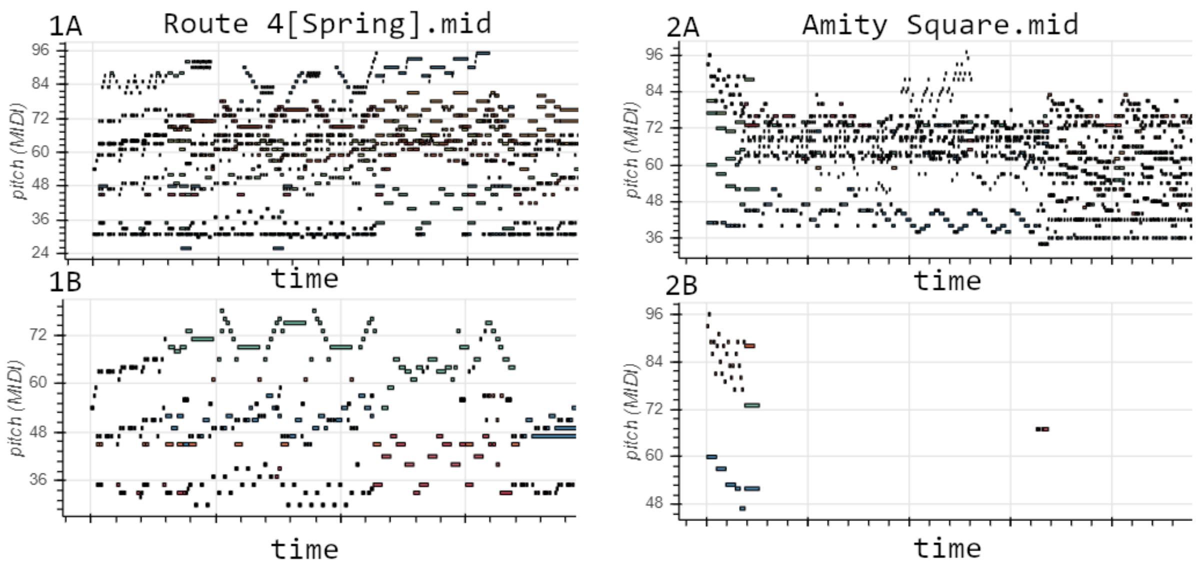

Figure 1: MIDI pitches plotted wrt time where A is the original track, B is after only the first 4 instrument tracks were kept. MIDIfile 1 retains a noticeable structure after the processing whereas 2 loses most of its notes.

I think there might be ways to solve this problem by eliminating the bass/drum tracks and then choosing the 3-4 instrument tracks with the highest quantity of notes but I was not able to find anything in the music21[1] or MIDO[2] documentation that would help me achieve this. I'm sure there is a way though if I had given myself more time to read through it.

Nonetheless it was worth trying to train the model with these samples to see if an increase in the dataset would have a positive effect on the results. This time the size of the training set was "only" 6 GB which Colab was much happier and willing to run with the following hyperparameters:

```
batch_size=32, rnn_layer_sizes=[256, 256, 256], dropout_keep_prob=0.5, learning_rate=0.001
```

Surprisingly, the outputs don't have a significant improvement over the smaller data set from the preliminary results. They don't feel better or worse they just sound like more machine learning music, absent of long-term musicality or structure. The second run results seem more sparse (i.e. less notes played) which is likely due to the inconsistent results of the data pre-processing. Considering that all the pre-trained music models available have been run on hundreds of thousands of samples, I think at this threshold of dataset size it isn't feasible to produce outputs that come close to the success of the Google Magenta music initiatives or other baselines of music generation. The corpus of work for piano performance (which is what many of these models are trained on) or Bach chorales which `polyphony-rnn` was trained on are much too large for the OST of an individual game series to contend with.

1.2 Analysis of Final Results

Cross-entropy loss on held-out data (10% of input data) was used to evaluate the performance[3]. Just like with the first run, the increase in accuracy and the decrease in loss over time suggest that the model is getting better at producing outputs that more closely resemble the inputs.

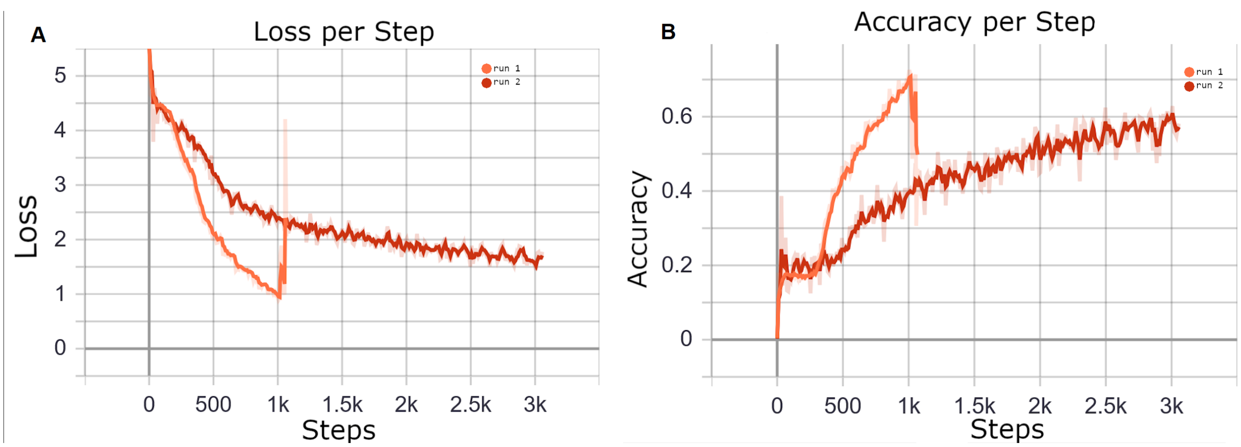

Figure 2: Loss(A) and Accuracy(B) plotted with respect to the number of steps the model has completed.

Due to the increase in the data size of the inputs the new model took many more steps to approach similar levels of accuracy and loss as the preliminary results and only after 3000 steps were these metrics for the second model close to those in the preliminary results.

The perplexity of the model can be said to be how many choices it has at each step. Since polyphony-rnn converts the sequences example to wordvectors [3] and uses the tensor flow implementation of cross entropy, we can calculate the number of choices as e^{loss} So for run 1 we have $e^{1.7} = 5$ an drun 2 $e^{2.2} = 9$. So we could say the model is as confused on the test data as if it had to choose randomly between 5 or 9 options respectively for each note.

Again, while these measures of accuracy are great for strict classification problems they don't necessarily translate well into whether the goal of producing Pokemon music is being met. The overall musical quality of the outputs certainly increased as the model conitnued to train and the loss decreased but their quality as standalone songs that could work as elements of a video game soundtrack is questionable.

2 Final demonstration proposal

2.1 Stacks and Technologies

The final product is evisioned to be a simple webapp with a React frontend and Flask backend. The Flask backend will allow for users to generate new music instances from the saved bundle file of the trained model which will be one page of the website. The other page will present two audio samples: one of which is pre-processed input data and the other will be a piece generated by the model. Users can then guess which they think was generated by the model. I will also attempt to collect statistics on the percentage of songs that were correctly guessed to be produced using machine learning (It's gonna be 100% :$\hat{)}$) in order to gauge the efficacy of the model.

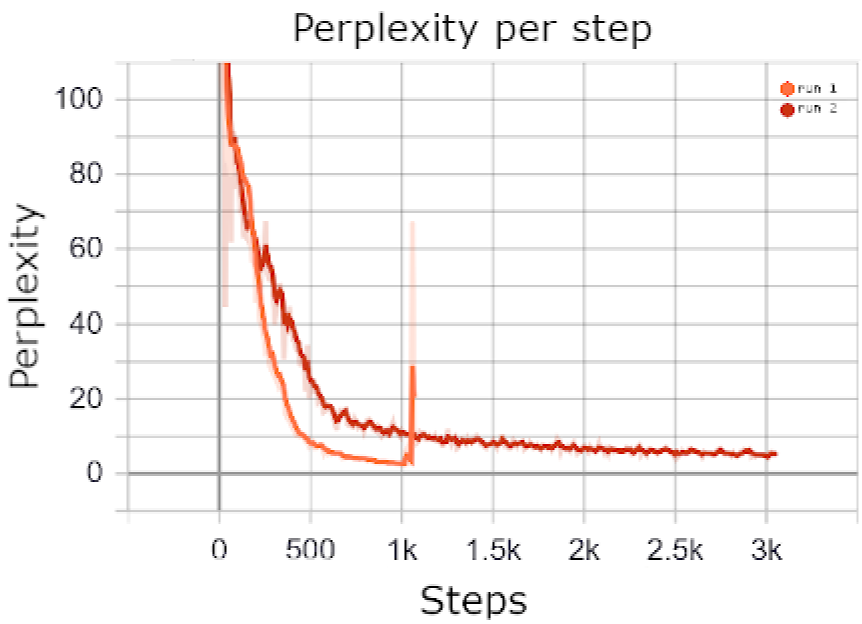

Figure 3: Perplexity of run1 and run2 plotted wrt the number of steps completed.

I have no experience with React, Flask or collecting statistics from a webapp but I will nonetheless attempt all three. I'm confident I can get the backend and frontend to work using the proposed frameworks it just might not be super pretty. The statistics can likely be done with webforms and then saving the guesses to a database. This will be incorporated if time and braincells permit.

All of my knowledge will be (and historically has been) from online resources such as tutorials, discussion boards and documentation. I will also ask the TMPs for help since they seem experienced in these frameworks!

References

[1] Michael Scott Cuthbert and Christopher Ariza. "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data." In: *ISMIR*. Ed. by J. Stephen Downie and Remco C. Veltkamp. International Society for Music Information Retrieval, 2010, pp. 637–642. ISBN: 978-90-393-53813. URL: `http://dblp.uni-trier.de/db/conf/ismir/ismir2010.html#CuthbertA10`.

[2] Rapolas Binkys Ole Martin Bjørndalen. *Mido - MIDI Objects for Python*. 2013. URL: `https://github.com/mido/mido/tree/stable#readme`.

[3] Feynman Liang et al. "Automatic Stylistic Composition of Bach Chorales with Deep LSTM". In: *18th International Society for Music Information Retrieval Conference*. Oct. 2017. URL: `https://www.microsoft.com/en-us/research/publication/automatic-stylistic-composition-of-bach-chorales-with-deep-lstm/`.