Generating Music Using LSTM Neural Networks

Kailash Subramanian, Daniel Angel, Nick Sangha, Keerthi Srilakshmidaran, Dr. Doug Degroot
Department of Computer Science, The University of Texas at Dallas

Abstract

Music is a repetitious art form. Musical elements, such as melodic themes, rhythmic patterns, or harmonic progressions, often appear multiple times throughout a work, meaning memory and contextual understanding is critical to music composition. Prior research in LSTM music generation has primarily focused on classical music. In this study, we investigate generating more modern genres. We selected classical Chopin and Schumann as controls, and compared the performance of LSTM architectures on jazz and anime soundtracks (a type of Japanese pop music). After conducting a survey, we found our LSTM models were able to generate modern music on par with the controls.

Background

Recurrent neural networks (RNNs) have been found to be suited for music generation due to its repetitious nature. In particular, long short-term memory (LSTM) networks, a type of RNN, have commonly been used in research in this field because they give the network the ability to process and find relationships in sequential data [1]. Several papers have found success using single-layer LSTMs trained on classical music. However, we were interested in seeing if similar architectures would still perform well for modern genres.

Data

We collected several midi datasets of Schumann pieces, Chopin mazurkas, anime soundtracks, and jazz pieces. For the Schumann and Chopin pieces, we used the piano-midi database [2]. For the anime soundtracks we scraped SheetHost [3], and for jazz we used Bushgrafts [4]. The Python library music21 was utilized to preprocess the data and isolate piano parts. The first 14,000 notes of each dataset were selected as input for our network.

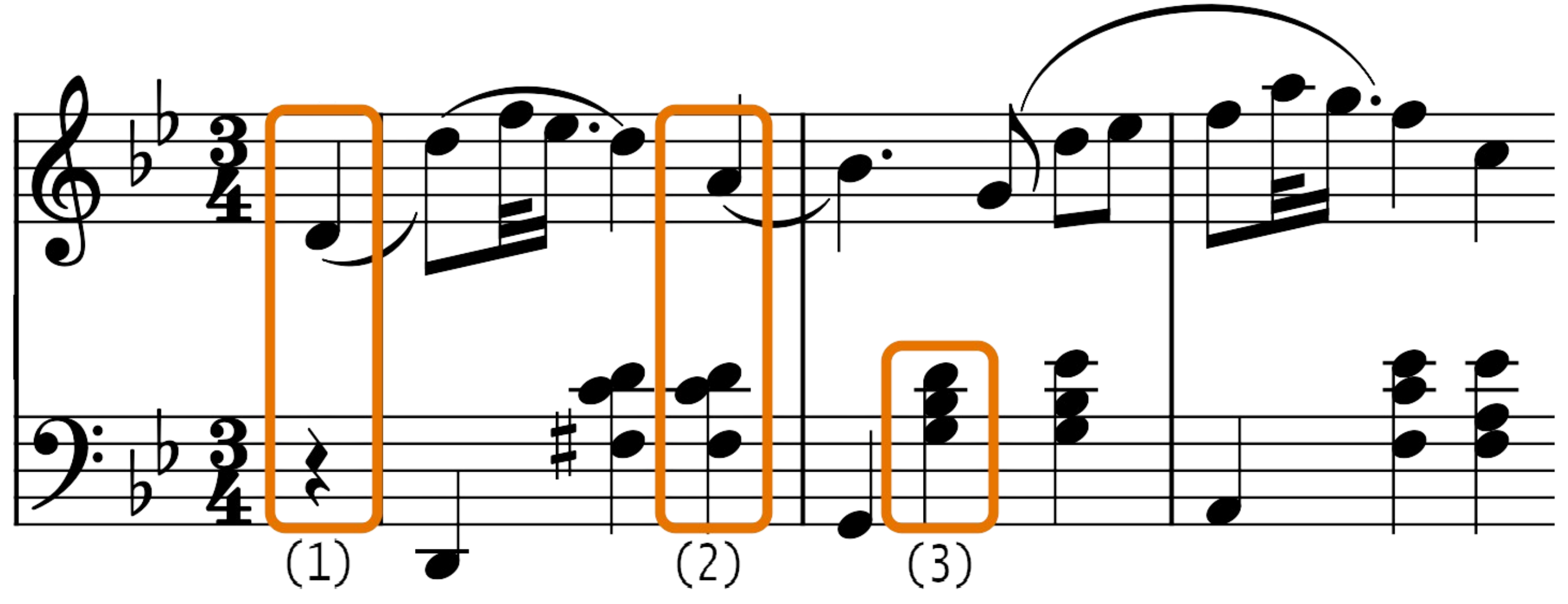

Figure 1: Example of elements being parsed by music21.

The input was transposed to C major and note lengths rounded to the nearest 0.25; we found this drastically reduced the number of categories. The music is parsed linearly based on time that a note is played in the music. In the first pass, we map every unique note and chord to an enumerated value. In the second pass, the element occurring at timestep n in the music is encoded as $j_n = [t_1, t_2, b_1, b_2]$ where t_1 and b_1 are the mapped treble and bass notes, and t_2 and b_2 are their respective lengths. For instance, timestep (2) in Figure 1 would be encoded with two notes. If a note is missing in the either clef, such as timestep (3), we extend the length of the other clef in j_{n-1} and fill any unknowns with zeroes. The input was then one-hot encoded. We chose a sequence length of 64, so the input to our network are 64 timesteps, and the output is the next timestep.

Models and Results

Architectures

We tested each of the four datasets on four model architectures:

- 1 layer LSTM (1-LSTM)
- 3 layer LSTM (3-LSTM)
- 2 layer LSTM with an Attention layer (Att. LSTM)
- 2 layer LSTM with a Bidirectional LSTM layer (Bi. LSTM)

After training each dataset and architecture combination for 200 epochs, we selected two architectures that showed the most promising results ("Best Model" and "Alternative") based on lowest training loss.

Dataset	Best Model	Loss	Accuracy	Alternative
Chopin	3-LSTM	0.1016	0.978	Att. LSTM
Schumann	1-LSTM	0.1423	0.983	Att. LSTM
Anime	1-LSTM	0.1213	0.971	Att. LSTM
Jazz	1-LSTM	0.1650	0.902	Att. LSTM

Sample Output

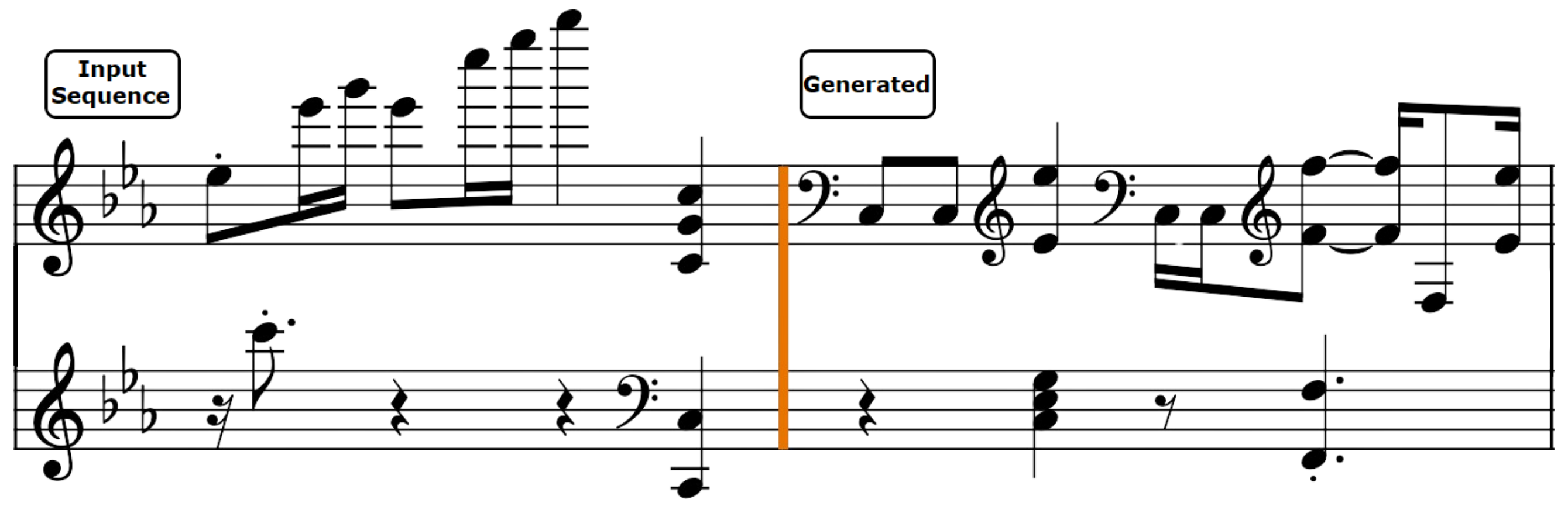

Figure 2: Output of 1-LSTM Network trained on the anime dataset.

After finding the "Best Model" and "Alternative" for each dataset, we generated samples from each. The networks were fed an input sequence from their respective dataset. We then generated midi files by un-mapping the music vectors that each model outputs.

Survey Results

To judge the musicality of each network's output, we used a survey. 41 people were given 8 clips (1 generated by "Best Model" and "Alternative") and asked to rank their favorite 4 clips.

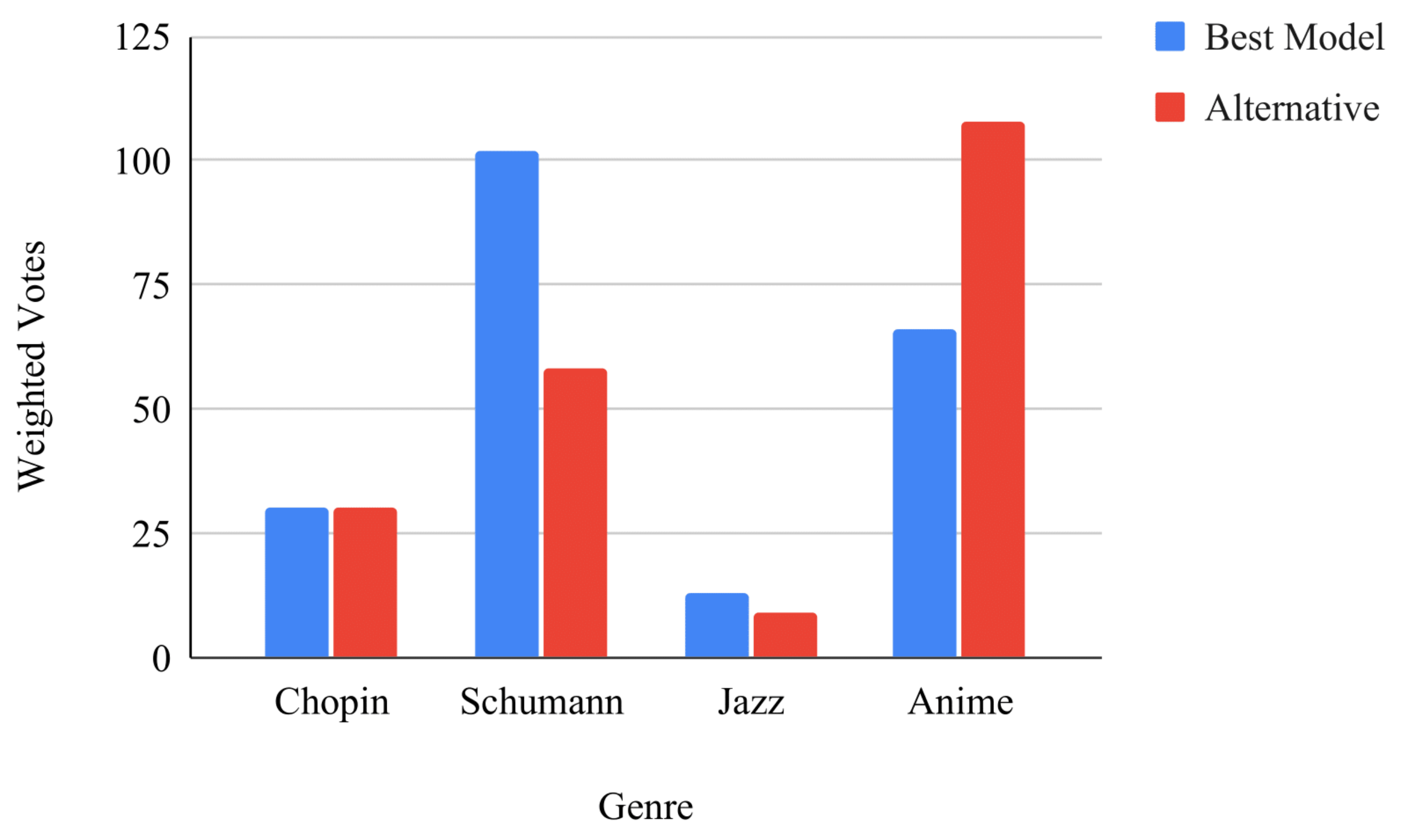

Figure 3: Weighted survey results per model.

From our survey results and analysis on the accuracy and loss of specific models in different genres, we have determined the best model to be the 2-layer LSTM with an Attention layer for the Anime genre, resulting in a weighted vote score of 108. The model coming in close second which was also the victor in first place votes, was the 1 layer LSTM for the Schumann.

Conclusion

From the results of our survey, it is clear LSTMs are capable of generating modern music just as well as classical genres. Although the networks trained on the Jazz dataset did not perform well in the poll, the models trained on Anime soundtracks had the highest average rating of all models. Additionally, the added Attention layer appears to have varying benefits per genre, as seen when comparing the Anime and Schumann models. Further, through our initial selection of models, we found that models with more parameters did not necessarily produce more accurate models during training, as the most common "Best Model" was the 1-LSTM.

Future Work

We plan to continue on improving upon our models by adding more elements to our music vectors, like time between notes, which would allow our models to understand the position of the notes relative to each other. Additionally, the pairing of certain lyrics with specific chords could entirely evolve the music generation field.

References

[1] C. Olah, "Understanding LSTM Networks" *colah's blog* [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[2] B. Krueger, "MIDI files" *Classical Piano Midi Page - MIDI files.* [Online]. Available: http://www.piano-midi.de/midi-files.htm

[3] Sheethost, "Public Sheet Music" *Sheethost.* [Online]. Available: https://sheet.host/tag/animenz

[4] D. McKenzie, "MIDI" *Doug McKenzie Jazz Piano.* [Online]. Available: https://bushgrafts.com/midi/