SongRNN: Music Generation Through a Character Level LSTM

Brandon Szeto
Jacobs School of Engineering
University of California, San Diego
San Diego, CA 92122
`bszeto@ucsd.edu`

Darren Yu
Jacobs School of Engineering
University of California, San Diego
San Diego, CA 92122
`dmyu@ucsd.edu`

Nathaniel Thomas
Jacobs School of Engineering
University of California, San Diego
San Diego, CA 92122
`nathomas@ucsd.edu`

Abstract

With a set of musical samples in ABC notation, we can generate music with a character-level LSTM. We can accomplish this task by feeding our LSTM musical characters in a sequence, and the network would slowly learn to predict the upcoming sequence of notes. Once trained, our model could generate music samples based on a short sequence of prompted notes. The style of the music generated by our final model has a rich, mature, and elaborate feel. By modifying the network depth, layer size, and dropout rate, we could tune our model to generate a range of musical samples. After hyperparameter tuning, we found that our LSTM created more elegant musical samples than a traditional recurrent neural network. The ability of the LSTM to drop information allows it to resemble the original test samples better than a standard RNN.

Preprint. Under review.

Introduction

From classical era music to modern hip-hop, music has been ingrained in human culture for thousands of years. The idea of creating music is assumed to be a task only accomplishable by humans. But with a large sample of composed music, a recurrent network model, and some computational power, we can generate music without the interventions of humans. A long short-term memory network (LSTM) is an type of RNN that uses gates to control the flow of information in and out of the network. LSTMs are great at predicting sequences of information based on input information. With this in mind, our goal is to generate music from a short prompt of musical characters. We can accomplish this task with a character-level LSTM.

Related Works

Long Short-Term Memory

Hochreiter and Schmidhuber [1997] proposes a modified recurrent neural network (RNN) architecture designed to overcome the vanishing gradient problem in traditional RNNs. LSTM networks are capable of learning long-term dependencies in sequential data by utilizing a memory cell and a set of gates to regulate the flow of information. This paper provides deeper intuition behind why LSTM works for music generation.

Deep Learning Techniques for Music Generation

Briot et al. [2019] discusse the application of deep learning techniques, particularly recurrent neural networks (RNNs) for music generation. The paper explores various approaches for modeling musical data, including symbolic representations (e.g., MIDI) and raw audio waveforms. This paper provides deeper intuition behind why RNNs are effective at music generation.

Methods

For our baseline model, we had a simple design of a single hidden layer LSTM. Our network takes in a sequence of music notes in ABC notation one at a time. After going through the hidden layer, the output is a softmax of the probability distribution stream of the upcoming musical character.

Training network using Teacher forcing

To train our model, we went with a teacher-forcing technique. After passing in a character to the LSTM, we look at the predicted character and correct the weights if necessary. To update our weights, we used categorical cross-entropy since it best fits the scenario. We went with an Adam optimizer to create a model that does a better job at generalizing outputs based on inputs.

Song Generation

Hyper-parameter Tuning

Describe the RNN architecture you used in 5.a and briefly describe the approach you took in tuning your hyperparameters.

Feature Evaluation

Describe why it is important to do feature evaluation. Describe the approach you took in generating the Heatmap of the activation's of each neurons for each of the characters.

Results

In the Results section, you should demonstrate your baseline model's performance, performances for each of the parts of Q3, Q4, Q5 and Q6 (training, generation, hyperparameter tuning, feature evaluation). You are expected to report the following things in your submission:

a Report results for your baseline model, a 1-hidden layer LSTM with 150 neurons. (2 pt)

b Report the three generated music samples with the above-mentioned three different Temperature (T) settings. Submit their .txt and .midi files as a part of your gradescope submission. (3 pt)

c Report loss plots for hyperparameter tuning experiments namely: RNNs vs LSTMs, changing number of neurons in the hidden layer and varying dropouts. (6 pt)

d Report heatmaps with appropriate scales, for three different neurons from your best-trained model. (3 pt)

Discussion

Please discuss the following important points as well: How did the qualitative performance and loss of RNN vs LSTM models differ? Discuss the performance differences with respect to changes in the number of neurons and dropout. Also, draw insights from the heatmap of each of your neurons.

Contributions

Brandon Szeto: Data evaluation, loss graphs, diagrams, write up (related work, methods, and discussion).

Darren Yu: LSTM/RNN hyperparameter tuning, feature evaluation, and write up (abstract, introduction, and results),

Nathaniel Thomas: LSTM, SongRNN, model architecture, model training, and music generation.

References

Jean-Pierre Briot, Gaëtan Hadjeres, and François-David Pachet. Deep learning techniques for music generation – a survey, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.