# 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 LSTM musical characters in a sequence, and the network would slowly learn to predict the upcoming sequence of notes. Once trained, a 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. A point of comparison could be made against a standard RNN. By modifying the network layer size and dropout rate, we could tune our model to generate a range of musical samples. After 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 in three different areas allows it to resemble the original samples better than a standard RNN. Our LSTM trained faster and created better music than our 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

For song generation, we incorporated a hyperparameter called temperature. Temperature introduces a level of randomness by influencing the probability distribution of categories. Instead of taking the argmax of the probabilities like in a normal softmax layer, we pick the following character by rolling an n-sided dice of n character outputs. By doing this, we allow the music generation to be unique and non-deterministic. To begin, we feed our model an input prompt of characters. After the prompt is finished, we allow the model to generate a character by itself and feed the next character as input. This continues until it reaches an endpoint of fixed length.

### Hyper-parameter Tuning

To have some kind of comparison, we created an RNN with a single hidden layer that is affected by a dropout rate. A dropout rate is the probability any neuron in the hidden layer gets dropped or is ignored. By including a dropout rate in an RNN, we help reduce the chances of an exploding and vanishing gradient. The size of the hidden layer and the dropout rate are controlled as hyperparameters.

### Feature Evaluation

After generating a musical sample, we can visualize how each generated character affects the neurons in an LSTM. To do this, we first perform a forward pass of each character in the generated sample. Because we are only viewing the neuron activation, we will not call a back pass to avoid updating the gradient. After we perform a pass of the whole sample, we can create a heat map to better visualize how each character affects each neuron.

## Results

### LSTM Results

### Music Generation from LSTM

### RNN and LSTM Loss Plots

### LSTM Neuron Activation Heatmaps

## 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**: RNN/SongRNN, RNN hyperparameter tuning, loss plots, write up (related work and discussion).

**Darren Yu**: Feature evaluation, heatmaps, and write up (abstract, introduction, methods, and results),

**Nathaniel Thomas**: LSTM/SongRNN, model training, music generation, and inference.

## 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.