**Deep Learning For Music Review and Recreation**

# Abstract

This project explores and attempts to recreate the intersection of deep learning and music composition through the development of a neural network model capable of generating music in the style of J.S. Bach, as first seen in the paper “Deep Learning for Music” by Huang and Wu (2016). Utilizing Long Short-Term Memory (LSTM) networks, a form of recurrent neural networks known for their efficacy in learning sequences, the model was trained on a corpus of MIDI files to capture the intricacies of Bach's compositions. The training involved processing the MIDI files to extract musical elements—notes and chords—and preparing sequential data that enabled the network to learn patterns and predict subsequent musical notes. Major findings from the project include the successful identification and generation of music sequences that bear Bach's compositional signatures. The generation of music was followed by a visualization step using t-Distributed Stochastic Neighbor Embedding (t-SNE), which provided insights into the learned features and the model's internal representations of music data.

# Table of Contents

[Introduction](#_hb9p0uy3uj3n) 3

[Methodology](#_xvvu37x41qqb) 4

[Deep Learning Model](#_5dg3ebnhn2cj) 4

[Data Collection and Preprocessing Techniques](#_5dg3ebnhn2cj) 4

[Model Architecture and Parameters](#_5dg3ebnhn2cj) 4

[Training Process and Hyperparameter Tuning](#_3nu4ogbljvzy) 5

[Dataset](#_rowssuozuiu0)  6

[Experiment](#_s86knj5v5t1d) 7

[Results Discussion](#_k2jutkxdrh8n) 9

[Conclusion](#_kzdp0vb1dvul) 10

[References](#_dga3yqdcmsb3) 11

[Author Contributions](#_4nb7qn3z7s47) 11

# Introduction

The aim of this project is to develop a deep learning model, specifically using LSTM networks, to generate music that captures the essence of Bach's style by way of recreating the work performed in the paper “Deep Learning for Music” by Huang and Wu (2016). The project encompasses data preparation, model training, music generation, and analysis of the results through visualization. Ultimately, it is driven by the challenge of emulating human creativity in music composition using artificial intelligence, particularly focusing on the works of J.S. Bach, whose compositions offer a complex and structured dataset for machine learning models.

Previous efforts in music generation focused on single melody creation. Recent work has incorporated polyphonic music modeling, especially using Recurrent Neural Networks (RNN) combined with Restricted Boltzmann Machines (RBM). In their paper, Huang and Wu, however, approached this task through deep neural nets alone, specifically using Long Short Term Memory (LSTM) architecture.

This initiative has the potential to impact music creation, offering tools for composers, educational insights into music theory, and novel applications in entertainment and interactive media. It stands at the intersection of technology and art, exploring how AI can contribute to creative processes.

# Methodology

## Deep Learning Model

The project employs a Long Short-Term Memory (LSTM) network, a type of recurrent neural network (RNN) that is particularly well-suited for learning dependencies in sequence data. LSTMs excel at remembering information over long periods, which is crucial for music where the context can span across various measures.

## Data Collection and Preprocessing Techniques

The dataset consists of MIDI files of J.S. Bach's compositions. These files were parsed to extract musical elements such as notes and chords. The preprocessing involved converting these elements into a numerical format that could be processed by neural networks. Sequences of 50 notes/chords were used as input to the network with the next note/chord as the output, forming a supervised learning problem.

## Model Architecture and Parameters

The model is a Sequential model with the following architecture:

1. **LSTM Layer:** The first layer has 128 LSTM units. It takes a sequence of 50 steps as input, each step containing a single feature, representing a note or chord.
2. **Dropout Layer:** A dropout layer follows with a rate of 20%, helping to prevent overfitting by randomly setting input units to 0 at each update during training time.
3. **LSTM Layer:** Another LSTM layer with 128 units, also receiving sequences, provides the model with the ability to learn higher-order dependencies.
4. **Flatten Layer:** Flattens the output to make it suitable for input to the dense layer.
5. **Dense Layer:** A fully connected layer with 256 neurons provides learning capabilities from the flattened LSTM outputs.
6. **Dropout Layer:** An additional dropout layer with 30% rate for further regularization.
7. **Output Layer:** The final dense layer with 204 units corresponds to the size of the vocabulary (unique notes and chords). It uses softmax activation to output a probability distribution over all possible outputs with the maximum probability being the ‘chosen’ note or chord.

The total number of parameters in the model is 1,889,228, which are all trainable.

## Training Process and Hyperparameter Tuning

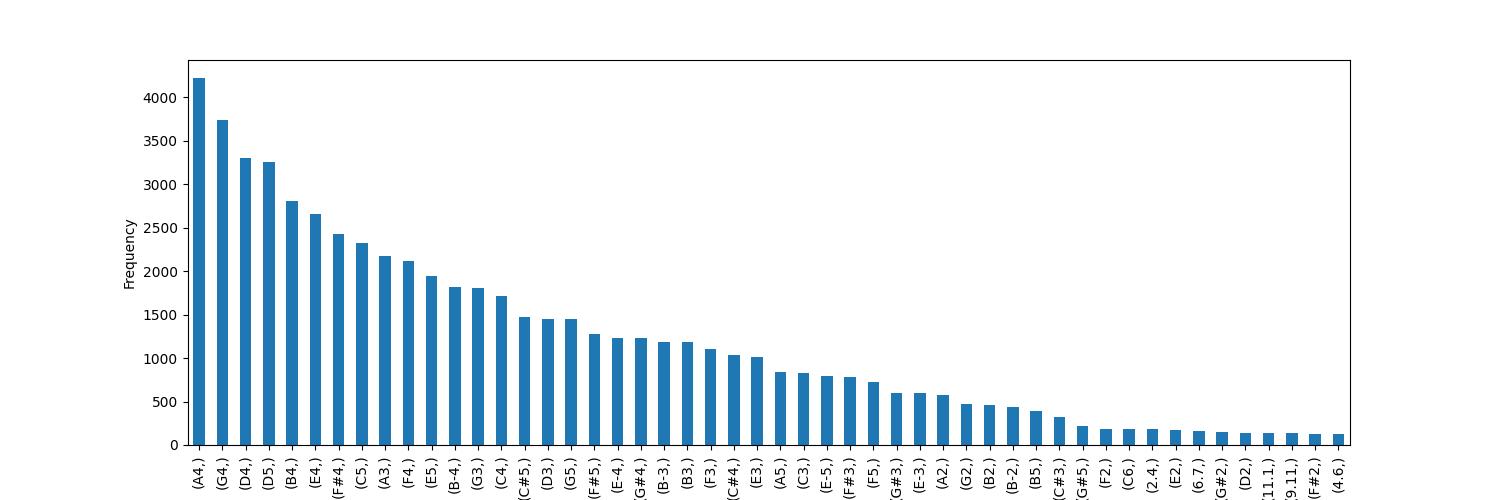
The model was trained on the prepared sequences using categorical cross-entropy as the loss function and the Adam optimizer. Hyperparameters, such as the number of LSTM units, dropout rates, and the aforementioned loss and optimizer functions, were chosen based on their use in Huang and Wu’s original paper as a consequence of the attempt to recreate their findings. The model saves the weights yielding the best validation loss across the training epochs using the ModelCheckpoint callback, indicating the end of the training process upon convergence or after a predetermined number of epochs.

The training process involved feeding the input sequences into the model and adjusting the weights through backpropagation based on the error between the predicted and actual notes/chords. The selection of 50 epochs for training implies that the entire dataset was passed forward and backward through the neural network 50 times. The batch size of 50 indicates the number of training samples to work through before the model's internal parameters are updated.

The described methodology highlights a structured approach to developing a deep learning solution for music generation, with a focus on capturing the stylistic nuances of Bach's compositions.

# Dataset

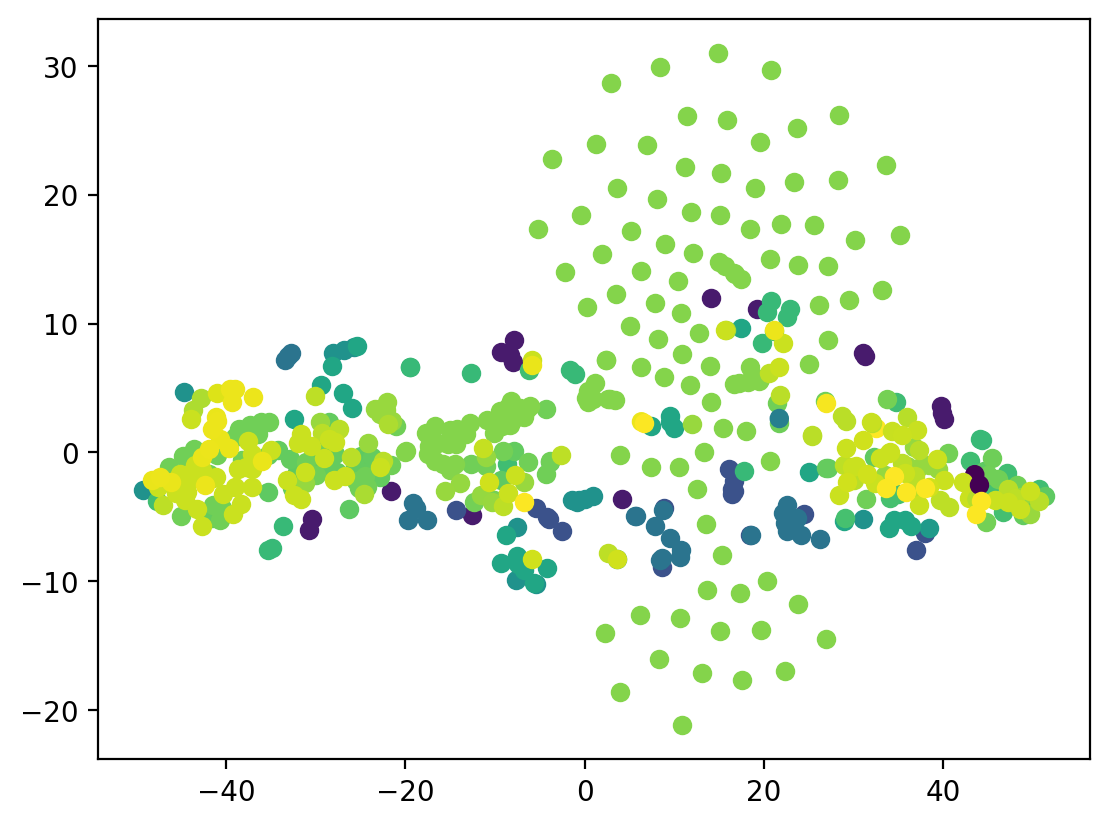
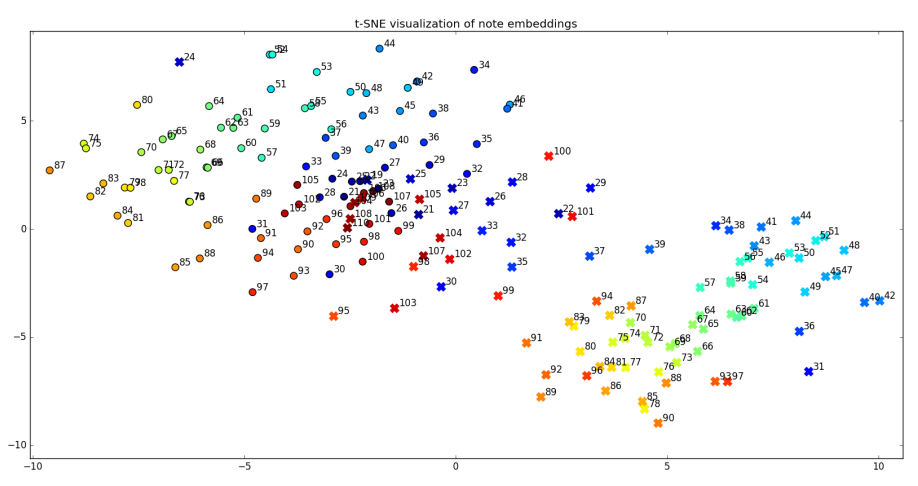
Unfortunately, the dataset is where the first limitation of recreating the original paper’s findings occurred. Since the original paper’s publishing in 2016, the datasets described in the original paper no longer exist at the websites it cites. Consequently, after discussion of whether this project remained feasible, a replacement dataset was sourced from other websites in an attempt to recreate the original to the best capabilities. This was deemed an acceptable allowance given the nature of Bach’s work as public domain, though it was noted this would introduce increased risk of not recreating the original experiment to its best accuracy.

The replacement dataset - found at bachcentral.com - contained 228 midi files of Bach’s works. Unfortunately, during preprocessing of data, it was found that 88 of these files were improperly formatted, leaving only 140 usable files for our dataset. Across these files, 62363 notes/chords were sourced across 204 unique pitches/chords. Figure 1 shows the frequencies of each note/chord, truncated to the highest 50 for graph readability. Of note from this visualization is that well over 85% of this dataset consisted of only the top-40 most pitches/chords. This was, unfortunately, substantially less than what was utilized in the dataset reported by Huang and Wu. These notes/chords were appended sequentially in the order sourced file-by-file, and this total list was used to generate a supervised training dataset of 62313 50-consecutive-element sequences which were each mapped to the note immediately following the sequence for labeling purposes.

# Experiment

The experiment was performed through a pipeline containing the dataset loading and preprocessing into sequence-note mappings, model creation, and the model training for the 50 epochs described in Huang and Wu’s paper. The training metrics used in that paper were never outlined in specifics, so categorical cross-entropy loss was monitored for determining the best training fit. Similarly, evaluation metrics for music generation were not outlined beyond a subjective listening comparison test, so the extent of evaluation was whether the midi file generated from the model post-training seemed comparable to Bach’s works. This was accomplished through an auditory evaluation by a sample of five individuals, ranking each of three short files generated from the model on a scale of 1-5 for how plausible the file could have been composed by a human being.

The original paper had, however, performed t-distributed stochastic neighbor embedding (t-SNE) visualization upon the dataset. The original visualization for the paper’s Bach dataset and the visualization for this replacement dataset can be seen in Figure 2.



# Results Discussion

Through the course of this experiment, the primary challenge was the lack of specificity to the original paper in regard to the capability of comparing the original results to the results of this experiment. The code utilized by the original, like the dataset, was not publicly found, meaning in the process of writing the code for this experiment partially from scratch, partially derived from the code accompanying the paper of “Deep Music” by Sarthak Jindal, 2018, (itself also derived and attempting to recreate the results of the original paper by Huang and Wu), it is largely unknown what, if any, procedural differences might have been included in this experiment’s process. Any such differences could go in part towards explaining the differences in the t-SNE plots generated from the original paper and this experiment, particularly the scattering of points found in the center of the experiment’s plot. Still, there are some similarities visible in the two plots. Both have two semi-concentrated clusters at opposite ends of the horizontal axis, though the plot from this experiment has less of a diagonal slant, keeping towards 0 on the vertical axis. The hues of the points also show the clusters tending towards the mid-frequencies around middle-C, just as in the original paper.

For the listening test, Table A outlines the results generated. The first file, upon listening, eventually collapsed into a repeating pattern of a single note. This did not occur in either of the other two files generated, which received, accordingly, higher rankings. While this indicates that, at least some of the time, the trained model is capable of producing music with semi-comparable complexity to Bach’s pieces, the small sample size and lack of recordings produced by the original paper’s model to perform a true comparison study prevent any further conclusions to be drawn regarding the musicality of this experiment’s model.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Person A | Person B | Person C | Person D | Person E | Average |
| File A | 3 | 2 | 4 | 1 | 2 | 2.4 |
| File B | 4 | 3 | 4 | 3 | 3 | 3.4 |
| File C | 5 | 4 | 3 | 4 | 4 | 4 |

# Conclusion

In conclusion, this experiment shows that, like the original paper it derives from, a multi-layer LSTM, character-level language model is able to learn and follow some meaningful musical structure as modeled from the compositions of the composer Bach, such that laypersons may believe the works comparable. This proves that deep neural nets, through LSTM, are capable of doing the same music generation as prior models using different network architectures.

Unfortunately, this project was limited from truly recreating and comparing the work of the original paper, “Deep Learning for Music” by Huang and Wu, due to the original paper’s lack of published code, limited description of areas of methodology, and inaccessible datasets seven years after the original paper’s publishing. While best efforts were made to recreate the findings of the original paper, these limitations did result in notable differences in the t-SNE plots used as the primary non-subjective metric for evaluation. It is recommended for future work towards this project and this topic to disregard such attempts at strict adherence to the original paper, to better attempt to verify than just recreate the findings.

# References

Huang, Allen, and Raymond Wu. “Deep Learning for Music.” *Papers WIth Code*, 15 June 2016, https://paperswithcode.com/paper/deep-learning-for-music. Accessed 15 Dec. 2023.

Jindal, Sarthak. “Deep Music.” *GitHub*, 9 Dec 2018, https://github.com/sarthak15169/Deep-Music. Accessed 15 Dec. 2023.