A PIANO BOY: AN OPTIMIZED MUSIC GENERATOR

Wei Liu, Yinzhi Xi, Jie He

Abstract

Curious about the creativity of neural networks, we developed The Piano Boy, a deep neural network application which can generate piano music by learning existing pieces of piano music.

We implement a 3-LSTM-layers model with an optimized function on rhythms, which has gained the best performance among all comparing models and solved the problem of ending up repeating one note to some extent.

Challenges & Improvements

The goal is to explore a way to improve the existing AI music composers by using different deep neural network architectures or adding creative elements to models. Since this is a generative model, evaluation can rely on the feedback of listeners.

Problem Statements:
1. Could we create a music generator that can develop a sequence of piano music that is quite "realistic", which means very similar to those written by composers?
2. Generally, most models focus on solving the same problem have the problems of rhythms, could we do better?

• Improvements:
When we extracted the music, we added a node without any music(empty) to the final generated training samples. The model can also output such nodes at the time of outputs. In this way, in the music that finally generated, there will be some “empty” music on the beat. In comparison, other music generators can only generate music that every beat has music. This music generated by our model has changing rhythm, not always the same rhythm, which is more realistic.

Dataset

The datasets are a combination of midi files from the following data sources, which are all classic piano music composed by famous human composers such as Mozart and Bethoven:

Classical Piano Composer: 93 pieces
Deep Jammer: 327 pieces
Piano-midi.de: 124 pieces
MuseData: 783 pieces

Related Work

Similar applications:
AI Jukebox: One Bidirectional LSTM
Classical Piano Composer: Three LSTM Layers
Deepjazz: Two LSTM Layers
Deep Jammer: Two LSTM Layers

Description of Model(s)

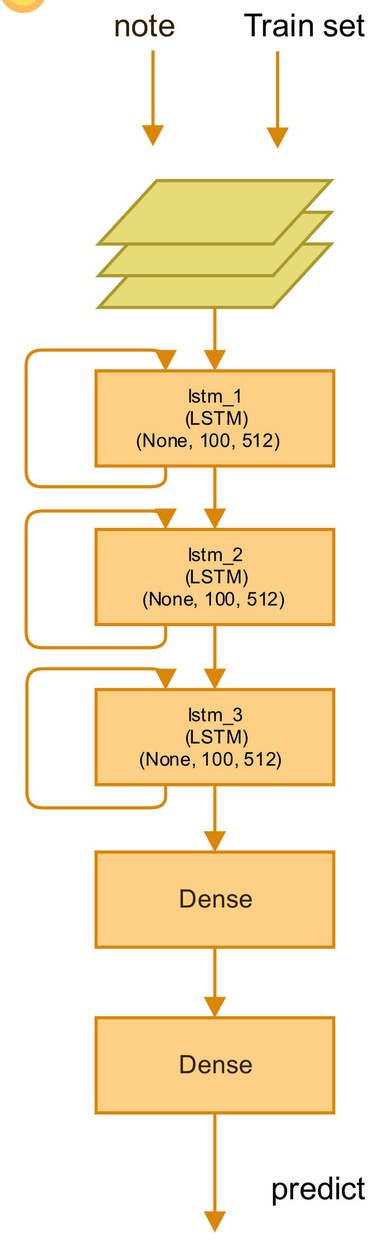
The Stacked LSTM has multiple hidden LSTM layers where each layer contains multiple memory cells and it is an extension to the basic LSTM model. We started from a 3-layer stacked LSTM model and made some modifications to it as comparison.

We tested the following variations:
Three LSTM Layers with modified feature:
Dimensions in 3 LSTM: (512->512->512)

Three LSTM Layers with different dimensionality:
Dimensions in 3 LSTM: (512->512->256)

Four LSTM Layers: Add one additional hidden LSTM layer as the fourth LSTM layer.
Dimensions in 3 LSTM: (512->512->512->512)

Model



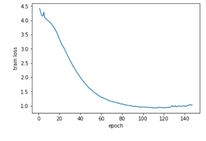
Results & Analysis

We compared our generated midi file with the outputs of other music generators which are treated as baselines. We conducted a survey to ask listeners to pick the best music product from these outputs.

Our model output is a solid leader in the voting results.

Program Name	Number of Votes
AI Jukebox	7
Classical Piano Composer	7
Deep jazz	5
Deep Jammer	4
The Piano Boy	18

We plotted the loss graph for each model we trained to observe and evaluate the relationship between cross-entropy loss and epoch over time. The graph of the best model selected is displayed:



Work Cited

McMahon, B. Music generator utilizing a Bidirectional LSTM architecture in Keras. https://github.com/cipher813/AI_Jukebox
Sigurgeirsson, S. Classical Piano Composer, https://github.com/Skuldur/Classical-Piano-Composer
Svegliato, J. Deep Jammer, https://github.com/justinsvegliato/deep-jammer
Kim, J. Deep learning driven jazz generation using Keras & Theano!, https://github.com/jisungk/deepjazz