Capstone Proposal

Yijian Zong
May 16th, 2018

Proposal

Domain Background

The ability of AI to compose and categorize music is underexplored in our times. As the development of machine learning is leaping, the computer can discover the underlying musical patterns, categorize, and even compose music.

In this project, I designed a machine learning application called “AI Chopin”, which integrates the ability to distinguish Chopin’s music from other composers and the capability in a “Chopin” way. This application utilizes a CNN to train classifier and RNN to compose using the Classical Midi Page.The project was inspired my pianist identity.

Problem Statement

The goal is to design a Chopin composer and music classifier, whose tasks are the following:

1. Download and parsing the Chopin Midi data.
2. Train an RNN that can compose Chopin Style music.
3. Preprocessing the parsed data and use the most relevant features for classification.
4. Train a CNN that can categorize Chopin’s music accurately.

The ultimate adaptation can be expected to output both the sheet music and sound file of the generated music, meanwhile, the classifier can categorize the music with decent accuracy.

Metrics

For RNN, cross entropy is a decent metrics to be used on the test set to assess how accurate the model is predicting the test data in the realm of natural language processing:

$$H(T,q) = -\sum_{i=1}^{N} \frac{1}{N} \log_2 q(x_i)$$

This metric was used when evaluating the RNN generator because, through the activation function softmax, the cross-entropy loss function can capture the probabilistic character of the network and efficiently update the weights to perform better.

For CNN, since the dataset is balanced, the most efficient way to measure performance should be accuracy. Accuracy is tailored made for binary classification, taking into account both true positive and true negatives with equal weight.

Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$

Dataset and Inputs

Data Collection:

The data preparation is done by the following steps:
1. Download midi files from [the Classical Piano Midi Page](#).
2. Parsing the midi files through the Music21 library.
3. Append both chords and notes to a list called “notes”
4. a. (For composing through RNN): prepare sequences by mapping notes name to integer through dictionary and get ready for training.
b. (For classification through CNN): give the “Chopin” label to each piece and use Panda DataFrame to create a table for counts of chords, notes, duration, key signature for each song.

Data Characteristics

1.The distribution of dataset is balanced.
2. The data is made balanced by selecting 20 Chopin selections and 20 other composers selection, including Liszt, Beethoven, Bach, etc.

Input setting:

1.The input for the RNN is a list of notes and chords parsed by Music21
2.The input for CNN classifier are the columns of the DataFrame.

Reasons for choosing dataset:

1.the midi file is the very best file for analyzing the patterns underlying the compositions.
2.The midi files from the Classical Piano midi page are very organized and come in handy, which make the task easier.

Solution Statement

The solution comes in two branches:

1.Using the parsed midi file and loading into RNN for training through 100 epochs. Then examine the result of loss function and optimize the weights in hdf5 file. Watch the cross-entropy loss decreases below 3.0. The music and sheet generated should have a certain composing style and the music should sound patterned with ornaments. Then train the model several times with different architecture to optimize the model.

2.Feeding the Data converted through midi by music21 to CNN and let the neural network discover the underlying pattern. A good result would have an accuracy above 60%. The output should express whether the composer is Chopin or not. Then train the model several times with different architecture to optimize the model.

Benchmark Model:

Composing Model: three midi files with two simple layers RNN and three epochs is a good starting point to compare when we have trained our optimized model.

Classifier: Naïve Base could be a simple model to classify the dataset and give a comparable result when looking at the final optimization model of CNN

Project Design:

Strategy for composing:
1.import library->data preprocessing
2.Functions: a. generate notes b. sequence preparation c. create networks d. generate midi e. generate sheet music in xml form
3. RNN design: 7 layers (Dropouts, LSTM, Dropouts, LSTM, Dense, Dropouts, Softmax activation)
Strategy for classifying:
1.Create Panda DataFrame
2.Functions: a. append notes b. append chords 3. Create_table 4. Train Classifier
3.CNN design: transfer learning through the reuslt_transfer folder in the Git repo transfer_learning_music. Add Maxpooling2D, dropouts , Dense with activation relu, GlobaaveragePooling2D, and output layer with softmax activation.

Reference: http://cs231n.stanford.edu/reports/2017/pdfs/22.pdf