Music Composition using AI

SoC-2022

Janaki Ram Puli

IIT Bombay

July 2022

Outline

Music Composition

Music Composition is a process of creating a new piece of music
Composition means "putting together".
Thus, music composition is something where music notes are put together in such a way that it gives pleasant sensation to our ears.
Parameters such as pitch interval, notes, chords, tempo etc. are used for composing short piece of music

About the Project

- The Project mainly focusses on music from Piano instrument
- Uses Long Short Term Memory (LSTM) , a type of Recurrent Neural Network (RNN).
- Platform: Visual Studio Code
- Language: Python 3.8
- Libraries Used: Tensorflow, Music21, Keras, NumPy, Sklearn, tqdm
- Dataset: Classical Music MIDI — Kaggle

Terminologies

- Note: This is a sound produced by a single key
- Chords: The combination of 2 or more notes is called a chord
- Octave: The distance between two notes is stated as an octave in a piano. It is specifically the gap between the two notes that share the same letter name.
- Recurrent Neural Networks (RNN)
  A recurrent neural network is a class of artificial neural networks that make use of sequential information. They are called recurrent because they perform the same function for every single element of a sequence, with the result being dependent on previous computations.
- Long Short Term Memory (LSTM)
  - LSTMs are a type of Recurrent Neural Network that can efficiently learn via gradient descent

Libraries

- Music21
  - Music21 is a Python toolkit used for computer-aided musicology
  - It allows us to teach the fundamentals of music theory, generate music examples and study music
- Keras
  - Keras is a high-level neural networks API that simplifies interactions with TensorFlow
  - It was developed with a focus on enabling fast experimentation
- TensorFlow
  - TensorFlow is a free and open-source software library for machine learning and artificial intelligence
  - It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks
- NumPy
  - NumPy is a Python library used for working with arrays
  - It also has functions for working in domain of linear algebra, Fourier transform, and matrices

Step 1

- Import all the important Libraries
- The midi file dataset has to be read using Music21 library
- "Albeniz" composed files has been used. (You can use more or less depending on your system)
- For this project, the files that contain sequential streams of Piano data has only been worked on
- All files are separated by their instruments and Piano is used only
- Piano stream from the midi file contains many data like Keys, Time Signature, Chord, Note etc.
- We require only Notes and Chords to generate music
- Lastly, the arrays of notes and chords has to be returned

Step 2

- This is done to check the number of unique notes and their distribution
- 50 is used as a threshold frequency
- Only those notes which have frequencies more than 50 have been considered
- These parameters can be changed anytime
- Two dictionaries are created where one will have notes index as a key and notes as value and other will be the reverse of the first i.e. key will be notes and value will be its respective index
- These dictionaries will be used in the next steps

Step 3

- Input and output sequences for our model are created
- A timestep of 50 has been used. So, if we traverse 50 notes of our input sequence then the 51 st note will be the output for that sequence
- As our model requires numeric data, all notes are converted to its respective index value using the "note2ind" (note to index) dictionary which has been created earlier
- Array for our model is re-shaped and the data is split into 80:20 ratio.
- 2 stacked LSTM layers with a dropout rate of 0.2 are used
- A fully connected Dense layer has been used for output
- Output dimension of the Dense Layer is taken same as the length of our unique notes along with the 'softmax' activation function (Used for multi-class classification problems)

Step 4

DropOut

Dropout refers to ignoring units (i.e. neurons) during the training phase of certain set of neurons which is chosen at random. It basically prevents overfitting while training the model, while it does not affect the inference model.

- After building the model, it is trained on the input and output data
- For this, 'Adam' optimizer is used on batch size of 128 and for total 50 epochs
- After Training, model is saved for prediction

Step 5

- Using the trained model, the notes will be predicted
- A random integer(index) is generated for our testing input array which will be our testing input pattern
- Our array is then re-shaped and the output is predicted
- Using the 'np.argmax()' function, we get the data of the maximum probability value
- This predicted index is converted to notes using 'ind2note'(index to note) dictionary
- Our next music pattern is one step ahead of the previous pattern
- This process is repeated till we generate 200 notes
- This parameter can be changed as per your requirements

Thank You