INSTRUMENTAL MELODY GENERATION USING DEEP LEARNING

Amogh G Kotabagi

amogh1741@gmail.com

Department of Computer Science, Don Bosco Institute of Technology,

Kumbalagodu Bengaluru, Karnataka, India

Bharath R

bharathgowda926@gmail.com

Department of Computer Science, Don Bosco Institute of Technology,

Kumbalagodu Bengaluru, Karnataka, India

Arun Kumar E

arunkumare986@gmail.com

Department of Computer Science, Don Bosco Institute of Technology,

Kumbalagodu Bengaluru, Karnataka, India

Chetan Chirag K H

arunkumare986@gmail.com

Department of Computer Science, Don Bosco Institute of Technology,

Kumbalagodu Bengaluru, Karnataka, India

Dr. Nagaraja P

thsn.taj@dbit.co.in

Professor

Department of Computer Science, Don Bosco Institute of Technology,

Kumbalagodu Bengaluru, Karnataka, India

Abstract

Customarily, music was treated as a simple sign and was produced physically. As of late, music is drawn to innovation, and making music is an extremely fascinating test that tests the writer's inventive limit, whether it's human or machine. The utilization of profound learning methods for producing music has acquired fame as of late attributable to the high measure of processing power being accessible and the development of profound learning models which are appropriate for gaining designs from successive information. In this paper, we want to create melodic notes that are inside the setting of the prepared example information. For this we are utilizing the Neural Network model, melodic thoughts can be procured from these calculations to make another piece of music. The utilization of profound figuring out how to take care of issues in scholarly expressions has been a new pattern that has acquired a great deal of consideration and the mechanized age of music has been a functioning region. This task manages the age of music utilizing some type of music documentation depending on different LSTM (Long Short-Term Memory) designs. An End to End associated convolution layers are utilized alongside LSTM to catch rich elements from the example train information and utilize the melodic design, for example, notes or harmonies to help realize which further expands the nature of the music created in a type of MIDI (Musical Instrument Digital Interface).

Introduction

This paper principally centers around producing tunes consequently utilizing a Recurrent Neural Network (RNN). Individuals love to pay attention to songs that continuously decrease pressure and nervousness. A Melody is a Linear Succession of Musical tones that the audience sees as a Single Entity.

Task: Our undertaking here is to a few existing information and afterward train a model utilizing the information. this model doesn't just randomize the notes however produces another tune which is respectable quality music that ought to be agreeable and great to hear.

1.Recurrent Neural Network (RNN)

A discontinuous mind network is a class of phony cerebrum networks that use progressive information. They are called dull in light of the fact that they complete a comparative job for each part of progression, with the result being dependent upon past estimations. While yields are independent of past estimations in traditional mind associations

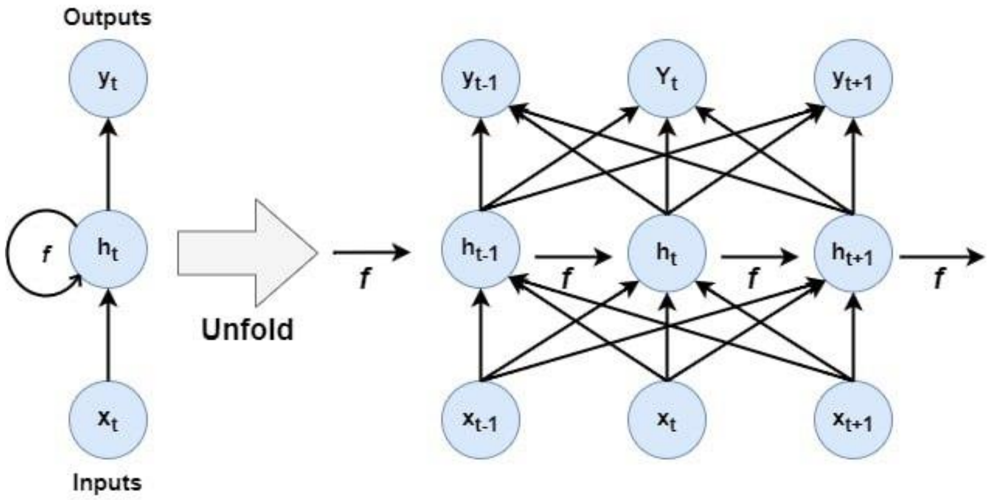


Figure 1 RNN Structure

2.Long Short-Term Memory (LSTM) organization :

In this paper, we will use a Long Short-Term Memory (LSTM) association. A sort of Recurrent Neural Network can successfully master utilizing slant plunge. Using a gating framework, LSTMs can see and encode long stretch plans. LSTMs are truly significant to handle issues where the association needs to review information for a broad stretch like the case in music and text age.

3.Music21:

Music21 is a Python device stash used for PC-upheld musicology. It grants us to show the basics of the music speculation, produce music models, and study music. The device compartment gives an essential association highlight get the melodic documentation of MIDI records. In this paper, we will use Music21 to eliminate the things in our dataset and take the cerebrum association's outcome and make an understanding of it into melodic documentation.

4.Keras :

Keras is an obvious level brain networks API that works on relationship with Tensorflow. It was made with an emphasis on drawing in fast trial and error. In this paper, we will utilize the Keras library to make and set up the LSTM model. Right when the model is set we up will incorporate it to make the melodic documentation for our music.

5.MIDI :

MIDI (Musical Instrument Digital Interface) is a specific standard that depicts a show, a mechanized place of collaboration, and connectors for interoperability between various electronic instruments, programming, and devices. MIDI passes now and again messages that decide note information(Such as pitch and velocity)as well as control signals for limits (like volume, vibrato, and clock signals). There are five kinds of messages and here we simply consider the Channel Voice type, which conveys consistent execution data over a singular channel.

Methodology

There are different to address music tunes like sheets of music documentation, ABC documentation, MIDI documentation, etc. In this endeavor, we are using MIDI documentation to address the music where it contains the quarter length, notes, and contributions and light of how it is seen by Music21. The data is separated into two thing types: Notes and Chords. Note objects contain information about the pitch, octave, and offset of the Note.

Pitch surmises the repeat of the sound, or how high or low it endlessly is tended to with the letters [A, B, C, D, E, F, G], with A being the most huge and G being the least. Octave underwear which set of pitches you use on a piano. Balance suggests where the note is coordinated in the piece.

Besides, Chord objects are a compartment for a lot of notes that are played in the meantime.

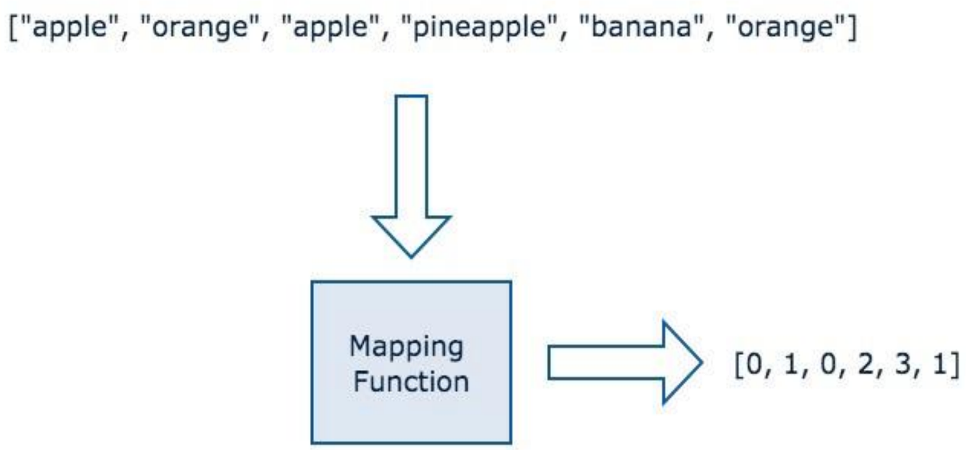
In this assignment we are using the maestro from, it is a combination of piano tune midi records from 2003 to 2018.

Data Preparation

Since we have reviewed the data and confirmed that our ideal features to use are the notes and harmonies as the data and consequence of our LSTM network the opportunity has arrived to set up the data for the association. In any case, we will stack the data into a bunch as ought to be noticeable in the code piece under:

We start by stacking each record into a Music21 stream object using the converter.parse(file) capacity. Using that stream object we get a once-over of the general huge number of notes and harmonies in the record. We add on the pitch of every single note object using its string documentation since the primary bits of the note can be repeated using the string documentation of the pitch. Likewise, we add every concordance by encoding the id of every single note as one together into a lone string, with each note being detached by a spot. These encodings grant us to decipher the outcome created by the association into the right notes and harmonies easily.

Since we have put all of the notes and harmonies into a continuous overview we can make the progressions that will go about as the commitment of our association.



When changing over from all out to mathematical information the information is switched over completely to number lists addressing where the classification is situated in the arrangement of particular qualities.

For instance apple is the essential specific worth so it advisers for 0, orange is the second so it advisers for 1, pineapple is the third so it advisers for 2, and so on.

In any case, we will make an arranging capacity to design from string-based obvious data to entire number based numerical data. This is finished because mind networks perform much better with number based numerical data than with string-based full scale data. A delineation of an obvious to numerical change ought to be noticeable in Figure 1.

Then, we want to make input groupings for the association and their different outcomes. The outcome for every data progression will be the principal note or amicability that comes after the gathering of notes in the information course of action in our once-over of notes.

In our code case, we have put the length of each gathering to be 100 notes/harmonies. This genuinely means that to expect the accompanying note in the game plan the association has the beyond 100 notes to help with making the assumption. I unequivocally recommend setting up the association using different gathering lengths to see the impact different game plan lengths can have on the music made by the association.

The last push toward setting up the data for the association is to normalize the data and one-hot encode the outcome

The last push toward setting up the information for the affiliation is to standardize the information and one-hot encode the result

We will take care of information into bunches. We will take care of a bunch of successions without a moment's delay into our RNN model. In the first place, we need to develop our batches.

We have set the accompanying parameters:

Batch Size = 64

Sequence Length = 100

We have found out that there is a total of 1052672 characters

in our data. The total number of unique characters is 269.

We have assigned a numerical index to each unique

character. We have created a dictionary where the key belongs to

a character and its value is its index. We have also created an

opposite of it, where the key belongs to the index and its value is its

character

One Hot Encoding

One hot encoding is a cycle by which obvious variables are changed over into a design that could be given to ML estimations to work on in expectation.One hot encoding can be described as the basic course of changing the outright data factors over totally to be given to machine and significant learning computations which in this way further foster conjectures as well as gathering accuracy of a model. One Hot Encoding is a run of the mill way to deal with preprocessing obvious features for AI models.

One hot encoding is a depiction of outright factors as twofold vectors. This at first expects that the absolute characteristics be intended to number characteristics. Then, every entire number worth is tended to as an equal vector that is every one of the no characteristics beside the record of the entire number, which is separate with a 1.

Network Architecture

Finally, we get to arranging the model designing. In our model we use four unmistakable sorts of layers:

LSTM layers are a Recurrent Neural Net layer that takes a gathering as a data and can return either progressions (return_sequences=True) or a structure.

Dropout layers are a regularization technique that contains setting an immaterial piece of data units to 0 at each update during the planning to thwart overfitting. The not altogether settled by the limit used with the layer.

Thick layers or totally related layers are totally related mind network layers where

every data center point is related with each outcome center point.

Since we have some data about the various layers we will utilize the time has come to add them to the network model.

For each LSTM, Dense, and Activation layer the essential limit is the quantity of centers the layer that should have. For the Dropout layer, the important limit is the unimportant piece of data units that should be dropped during planning.

For the essential layer, we want to give an exceptional limit called input_shape. The justification for the limit is to instruct the association in regards to the state with respect to the data it will plan.

The last layer should continually contain comparable proportion of center points as the amount of different outcomes our system has. This ensures that the consequence of the association will design directly to our classes.

For this paper, we will use a clear association including three LSTM layers, three Dropout layers, two Dense layers, and one institution layer. I would suggest intruding with the advancement of the relationship to check whether you can work on the possibility of the appraisals.

To figure the disaster for each accentuation of the readiness we will use hard and fast cross-entropy since all of our results basically has a spot with a solitary class and we have various classes to work with. Moreover, to further develop our affiliation we will involve a RMSprop enhancer for what it's worth generally an unbelievable decision for broken frontal cortex affiliations.

Whenever we have concluded the plan of our association the open door has shown up to start the readiness. The model. fit() ability in Keras is used to set up the association. The primary limit is the once-over of data progressions that we organized previously and the second is a summary of their specific outcomes. In our paper, we will set up the association for 200 ages (cycles), with each gathering that is spread through the association containing 64 models

To guarantee that we can stop the readiness whenever without losing the sum of our constant exertion, we will use model assigned spots. Model assigned spots give us a technique for saving the heaps of the association centers to a record after each age. This licenses us to stop running the cerebrum network at whatever point we are content with the disaster regard without worrying about losing the heaps. If not, we would have to hang on until the association has completely finished going through every one of the 200 ages before we might save the heaps to a report.

Generating Music

Since we have wrapped up preparing the network the time has come to have a great time with the network we have gone through hours preparing.

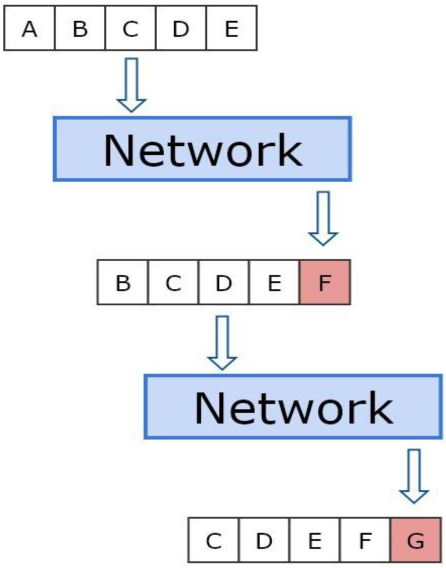
To have the choice to use the mind association to create music you ought to put it into a comparable state as previously. For straightforwardness, we will reuse code from the arrangement portion to set up the data and set up the association model in basically the same manner as previously. In any case, that rather than setting up the association we load the heaps that we saved during the planning region into the model.

By and by we can use the pre-arranged model to start making notes.

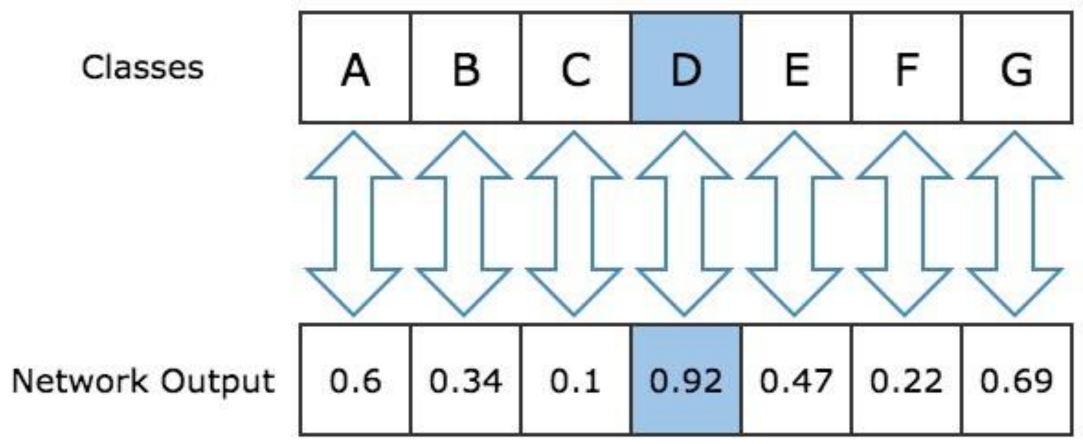
Since we have a full once-over important groupings accessible to us we will pick a sporadic record in the summary as our early phase, this licenses us to rerun the age code without changing anything and get different results predictably. In any case, If you wish to control the early phase basically displace the unpredictable ability with a request line conflict.

Here we similarly need to make an arranging capacity to decipher the consequence of the association. This capacity will design from numerical data to complete data (from numbers to notes).

We chose to deliver 500 notes using the association since that is roughly two minutes of music and provides the association with a ton of space to make a tune. For each note that we want to make we really want to introduce a gathering to the association. The primary gathering we submit is the progression of notes toward the starting record. For each following gathering that we use as data, we will dispense with the essential note of the progression and expansion the consequence of the past cycle around the completion of the course of action as ought to be noticeable in Figure 2.



To conclude the most likely assumption from the outcome from the association, we separate the document of the best worth. The value at list X in the outcome show analyzes to the probability that X is the accompanying note. Figure 3 gets a handle on this.



Then, we assemble all of the outcomes from the association into a singular show.

Since we have all of the encoded depictions of the notes and harmonies in a show we can start unraveling them and making an assortment of Note and Chord objects.

In any case, we want to choose if the outcome we are unwinding is a Note or a Chord.If the model is a Chord, we want to isolate the string into different notes. Then, we circle through the string depiction of each note and make a Note object for all of them. Then, we can make a Chord object containing all of these notes.If the model is a Note, we make a Note object using the string depiction of the contribute contained the model.

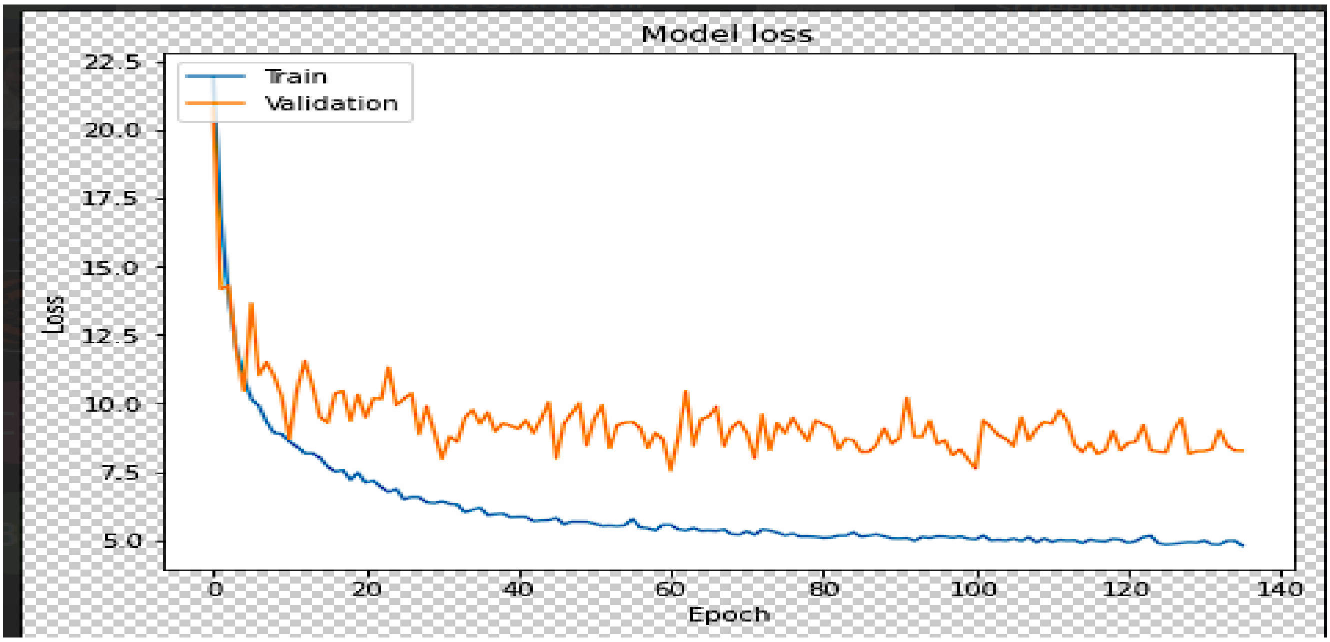
Close to the completion of each and every accentuation, we increase the offset by 0.5 (as we picked in a past fragment) and add the Note/Chord object made to a summary.

Since we have a once-over of Notes and Chords delivered by the association we can make a Music21 Stream object including the summary as a limit. Then finally to make the MIDI record to contain the music created by the association we use the form capacity in the Music21 device reserve to create the stream to a report.

Results

```
Epoch 195/200
987/987 [==============================] - 729s 728ms/step - loss: 1.3928 - categorical_accuracy: 0.7
150 - val_loss: 377.3140 - val_categorical_accuracy: 6.3379e-05
Epoch 196/200
987/987 [==============================] - 712s 722ms/step - loss: 1.2272 - categorical_accuracy: 0.7
272 - val_loss: 1202.7484 - val_categorical_accuracy: 6.3379e-05
Epoch 197/200
987/987 [==============================] - 714s 723ms/step - loss: 0.8920 - categorical_accuracy: 0.7
834 - val_loss: 1891.6295 - val_categorical_accuracy: 6.3379e-05
Epoch 198/200
987/987 [==============================] - 714s 723ms/step - loss: 0.7220 - categorical_accuracy: 0.8
034 - val_loss: 1891.6295 - val_categorical_accuracy: 6.3379e-05
Epoch 199/200
987/987 [==============================] - 714s 723ms/step - loss: 0.6920 - categorical_accuracy: 0.8
134 - val_loss: 1891.6295 - val_categorical_accuracy: 6.3379e-05
Epoch 200/200
987/987 [==============================] - 714s 723ms/step - loss: 0.6020 - categorical_accuracy: 0.8
347 - val_loss: 1891.6295 - val_categorical_accuracy: 6.3379e-05
```

Fig 4.1.1



As you can see the finally age level the absolute cross-entropy is 0.6020 and the Categorical Accuracy is around 83%and the anticipated succession is inside the setting of the Trained dataset. we could extraordinarily expand the exactness of the model by expanding the quantity of data of interest inside a specific circumstance and the GPU limit. Albeit the anticipated result is comparable to the setting of the info it is sub-par in understanding full information on the information like examples and counterbalances. We have just prepared the model with one hour of midi information from the Authors like Clementi,Mozart, Subert, and Brahms.

Conclusion

We have fabricated a device to produce tune with the MAESTRO dataset that contains piano melodies. We preprocess it, train our brain network model, then, at that point, produce the song with it. The songs are in MIDI arrangement. We utilize TensorFlow v2.0 to make it happen. I think TensorFlow v2.0 User Experience (UX) is superior to its past adaptation.

The tune produced by our model is likewise intelligible and great to hear. It can change how it plays its notes. For instance: When the generator from a note (implies it is a beginning of the music), it begins with a sluggish rhythm start.

There are a few things that we can pursue the music generator. In this article, we have explored different avenues regarding producing a solitary instrument. Imagine a scenario in which the music has various instruments. There should be a superior design to make it happen. There are numerous things that we can take a stab at trying different things with music information.