A Machine Learning Approach for finding a latent structure in Musical Composition

Submitted in partial fulfilment of the requirements of the degree of

Bachelors of Engineering

BY

1- Kartik Agashe

7- Bhavesh Bhambhani

14- Aditya Gudal

71- Siddhant Yeole

under the guidance of

Dr. T Rajani Mangala

Department of Electronics Engineering

V.E.S. Institute of Technology

2018-19

CERTIFICATE

This is to certify that the project entitled **"A Machine Learning Approach for finding a Latent Structure in Musical Composition"** is a bonafide work of **"01- Kartik Agashe, 07- Bhavesh Bhambhani, 14- Aditya Gudal, 71- Siddhant Yeole"** submitted to the **V.E.S. Institute of Technology** in partial fulfilment of the requirement for the award of the Bachelor of Engineering in **Electronics.**

Dr. T Rajani Mangala
Supervisor/ Guide

Mrs. Kavita Tewari
Head of Department

Dr. (Mrs) J.M. Nair
Principal

Project Report Approval for B. E.

This project report entitled **"A Machine Learning Approach for finding a Latent Structure in Musical Composition" by Kartik Agashe, Bhavesh Bhambhani, Aditya Gudal, Siddhant Yeole** is approved for the degree of **Bachelor of Engineering (BE) Electronics.**

Examiners

1.----------

2.----------

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Kartik Agashe

Bhavesh Bhambhani

Aditya Gudal

Siddhant Yeole

1. INTRODUCTION

As we all know, the music industry has burgeoned considerably in recent years. A raft of new developments has catalyzed innovations in the music produced and the instruments used by the professionals. The application of Artificial Intelligence in the field of music has been propounded since the late 1970's. Artificial Intelligence aims to enhance the understanding of the music produced for the general public. In this project, we take a naïve approach towards the application of a subset of Artificial Intelligence – Learning- for the development of musical notations in compositions and predictions.

Every musical note has a distinctive sound characterized by its pitch- rudimentarily a specific frequency. Furthermore, each musical note has a certain time period for which it occurs in a musical melody. Through this project, we aim to use a machine learning algorithm which would map the diverse musical notes to their corresponding frequencies as they occur in a musical piece. Another key feature of this algorithm would be to simultaneously consider the period of occurrence of every peculiar musical note in a given melody. This project would involve working on the musical notation required for playing instruments like keyboard, guitar, violin, et al. Additionally, once the algorithm displays the notes in the output window, we can further play these notes on a virtual keyboard. On playing a sequence of notes, the algorithm which is replete with the capability of learning and has been trained on numerous samples from the repository will be able to predict the next few notes of the composition for the user.

1.1 PROBLEM STATEMENT

In modern times with advent of the internet, it is tough to make time and find a virtuoso teacher as everyone is preoccupied. In the era of Big Data it is really tedious to design a computationally efficient algorithm which can effectively represent and manipulate musical midi formatted data. Even classical machine learning models like Support Vector Machines, Logistic Regression and t-SNE could be used but are rendered ineffective when it comes to delivering a high F1 Score for model evaluation. One way of learning latent vector space representations of data ,especially musical data was proposed using deep learning models such as WaveNets, Generative Adversarial Networks and Restricted Boltzmann Machines but the most efficient and popular method was devised using Recurrent Neural Networks incorporated with Long Short Term Memory[2,3,6]. Introducing music coach, a deep learning based virtual keyboard emulated using a user interface design on a web application. This keyboard understands the notes played and then engenders new notes stochastically using maximum likelihood[4] for the forthcoming sequence of notes to be played. Not only do users get the opportunity to interact with the keyboard, but they can also feed in a song of their liking and refer to the musical notes of that song in the web application.

2. LITERATURE SURVEY

1. Qi Wang, Ruohra Zhon, Yonghing Yan- A two stage approach to note-level transcription of a specific piano- (2017)

This paper uses a two-step approach for implementing note transcription and prediction in the piano. The first step includes the usage of two Convolutional Neural Networks (CNNs). The first network works for generating the onset of a note while the second deals with the pitch of every note. We used the method given in this paper as a reference for designing or back end code that deals with the generation of notations for a musical composition. The second stage of the approach involves verification of the notes generated by the piano for which spectrogram can be used. The second step was only referenced to gain a basic idea of why verification is necessary and to understand the different parameters necessary for obtaining the highest accuracy at the output.

2. Jean Francois Paiement, Samy Bengio, Douglas Eck - Probabilistic Models for Melodic Prediction - (2009)

The choice of a particular representation for chords has a strong impact on statistical modelling of the dependence between chord symbols and the actual sequences of notes in polyphonic music. Likelihoods and Conditional prediction error rates are used as complementary measures of the quality. Exploration of chord progressions and their interactions with melodies motivates this development.

Here it is trained with the EM algorithm where the graph is given, the chords can be observed in and around the key pressed or triggered, the data is partially observed and depends on the expected value of the next note in the sequence.

3. Douglas Eck, Jasmine Lapalm- Learning Musical Structure directly from the sequences of music- (2008)

To form a particular musical structure from a global database, the combination of a LSTM recurrent neural network with an auto-correlator predictive sequence is essential. Through this paper, we understood the behavioral characteristics of a sequence which is formed using the global database.The learning process of a model involves active prediction of the next note in a set of similar musical notes in that style. Further for testing purposes, extra notes can also be generated. We applied this understanding while designing the predictive algorithm which is a part of our project.

Also, characteristics such as sensitivity levels and their temporal order subtypes were studied namely- local structure, long-timescale metrical structure and long-timescale non-metrical structure.

4. Douglas Eck, Jurgen Schmidluber – Finding Temporal Structure in Music: Blues Improvisation with LSTM recurrent Network - (2002)

The above publication details the Long Short Term Memory (LSTM) method used for finding the temporal structure in music. An important requirement of the prediction process is a memory which can store sample values that help in training the algorithm to recognize patterns and thereby predict the temporal structure in a musical piece. The methodologies mentioned in the above paper helped us understand the characteristics of storage viz. size, durability of the data &ease of access – required for implementing the algorithm in our project.

5. Matija Marolt, Alenka Kavcic, Marko Prinovic- Neural Networks for Note Onset detection in Piano- (2002)

The aforementioned paper discusses one of the important prerequisites for designing our algorithm - piano transcription. The method discussed above uses onset detection of notes in a piano to help the user understand the methodology of transcribing music for a piano.The musical file is taken and converted into a MIDI format which is suitable for use in neural network algorithms and allows ease of programming. As our algorithm necessitated the conversion of our input samples from a normal audio format to a MIDI format, we studied the different characteristics described in the given paper viz. notes, starting times, total duration and loudness. To ensure proper detection and conversion of a song, it is mandatory to maintain a balance between all the parameters mentioned above.

This paper formed the crux of the foundation of our back-end code as we had to allow the algorithm to anticipate different values of each of the parameters mentioned in the previous paragraph.

6. Michael C. Mozer- Neural Network Music Composition by Prediction – Exploring the benefits of Psychoacoustic constraints and multi-scale processing – (1994)

From the above paper, we extracted the information regarding the characteristics of musical notes- pitch, duration and harmony.

-The Transition Table approach where notes are selected sequentially and each consecutive note is predicted and expressed as a function of the current note. This method uses a probabilistic approach to determine the notes in a composition.

-The Note-by-Note composition produces each note sequentially and linearly depending on the preceding context.

We applied our understanding of the note-by-note method while designing our algorithm.

3. WORK DONE

3.1 FLOWCHART

The project was implemented chronologically in a number of steps. Initially, to devise a methodology for executing our problem statement, we surveyed relevant literary material that consisted of scholarly articles published in various journals. These papers discussed the idea of musical composition and its structure as well as certain methods of prediction that were proposed at different points in time. Subsequently, we scouted for an online repository of music samples that could be used to train our algorithm. These samples included clips of the unique tones for every key of the piano. This data was also filtered before being sent as an input to the algorithm. Once the samples were ready for use, the back end code was written in Python. This code involved various parts viz. extracting the notes from the repository, generating input and output sequences, using functionality like one-hot encoding to create a shorthand representation, conversion of the audio file to MIDI format, training of the neural model and eventually the generation of samples. Thereafter, we displayed these notes on the screen in the form of notation and finally worked on the front end that dealt with the user interface. A web page was designed to display a virtual keyboard. This keyboard would be used to play the notes generated by the algorithm. Furthermore, with the training of the model we were able to ensure the virtual keyboard predicted the next few notes of the sequence following the input given by the user. The pictorial representation of the aforementioned text is given in **Figure 3.1**

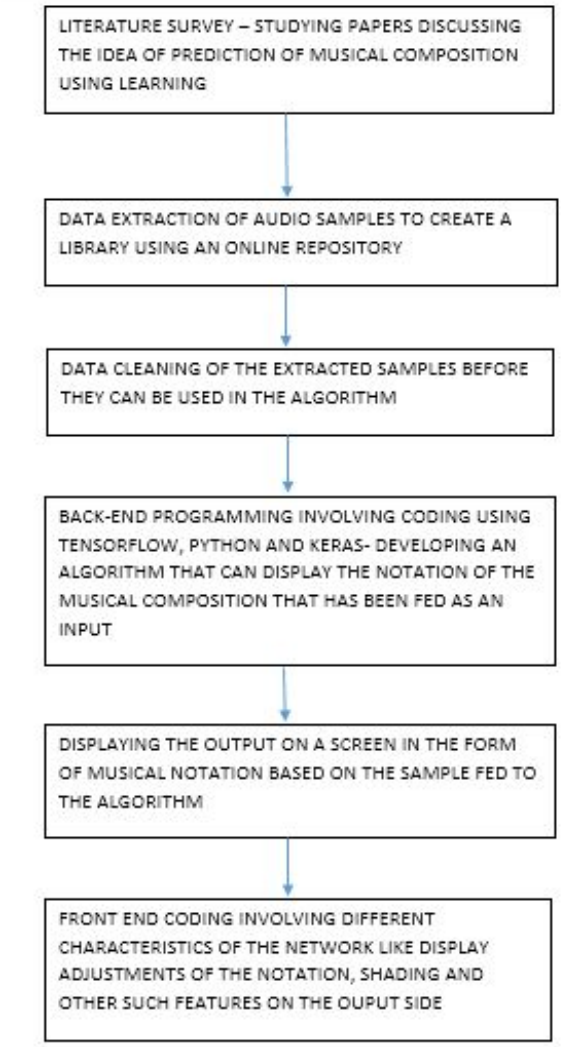


Figure 3.1 - FLOWCHART

3.2 SOFTWARE DETAILS

Softwares used for the Back End:

- Python (Version 3.5)
- Libraries of Python - glob, tensorflow, keras, numpy
- music21 API (toolkit for music converter)
- Google Magenta API

Softwares for the front end

- JAVA
- Javascript
- HTML5, CSS
- Node.js, tone.js

3.3 SOFTWARE TOOLS

Description of some of the most essential softwares from the aforementioned list is given below:

3.3.1 Google Magenta API

Magenta is a research project exploring the role of machine learning in the process of creating art and music. Primarily this involves developing new deep learning and reinforcement learning algorithms for generating songs, images, drawings, and other materials. But it's also an exploration in building smart tools and interfaces that allow artists and musicians to extend (not replace!) their processes using these models. Magenta was started by some researchers and engineers from the Google Brain Team but many others have contributed significantly to the project. The project uses TensorFlow.

Magenta.js is a JavaScript API for doing inference with Magenta models, powered by TensorFlow.js.

There are two main types of Libraries used in Magenta:

- Music : TensorFlow.js implementations and support libraries for Magenta's musical note-based models including transcription with Onsets and Frames and generation with MusicVAE, MelodyRNN, DrumsRNN, PerformanceRNN and ImprovRNN.

- Sketch: TensorFlow.js implementations and support libraries for Magenta's sketch models including SketchRNN.

In our project we use the MelodyRNN model available in the music library of the Google Magenta API.

MelodyRNN: This model applies language modeling to melody generation using an LSTM. The basic configuration acts as a baseline for melody generation with an LSTM model. It uses basic one-hot encoding to represent extracted melodies as input to the LSTM.

- Lookback RNN introduces custom inputs and labels. The custom inputs allow the model to more easily recognize patterns that occur across 1 and 2 bars. They also help the model recognize patterns related to an events position within the measure. The custom labels reduce the amount of information that the RNN's cell state has to remember by allowing the model to more easily repeat events from 1 and 2 bars ago. This results in melodies that wander less and have a more musical structure.
- Attention Configuration: In this configuration we introduce the use of attention. Attention allows the model to more easily access past information without having to store that information in the RNN cell's state. This allows the model to more easily learn longer term dependencies, and results in melodies that have longer arching themes.

3.3.2 Music21 API

Music21 is a Python-based toolkit for computer aided musicology. People use music21 to answer questions from musicology using computers, to study large datasets of music , to generate music examples, to teach fundamentals of music theory, to edit music notation, study music and the brain, and to compose music, both algorithmically and directly. Using the toolkit we found a sense of overall profile of a piece or a repertory of pieces. This system has been growing and expanding, it is an open source project nurtured by MIT music school and is perennially in development with Harvard, Smith and Mount Holyoke Colleges bolstering this toolkit and reinforcing it into a mature system. The functions are used to parse and convert notes, chords and play the melody using inherent converters and storing them into music xml format using external converters such as Lilypond.

3.3.3 Important Terminology used in the project:

a. One-hot Encoding: One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. The problem with other forms of encoding viz. label encoding is that it assumes higher the categorical value. Therefore, we use one hot encoder to perform "binarization" of the category and include it as a feature to train the model.

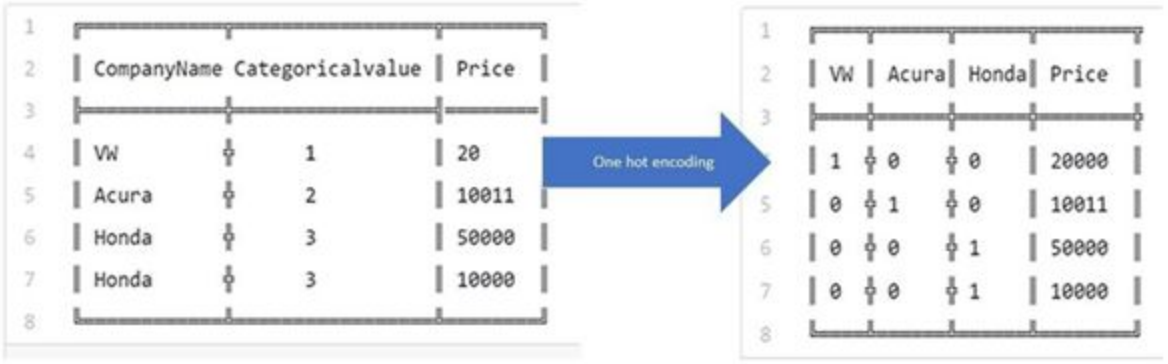


Figure 3.2. Example of One Hot Encoding

An instance of one hot encoding is given in **Figure 3.2**. Here each company has a categorical value associated with it. This categorical values cannot be read by the learning algorithm directly. For this reason categorical variables are encoded using either Label Encoding or One Hot Encoding. One Hot encoding gives each categorical value a unique vector. As there are 3 companies, with Honda repeated twice, the vector index will be "one hot" or given a numeric value of 1 wheresoever the company is indexed in the table. For instance, Acura will have the 2nd index value in the vector represented as numeric 1 , whereas Honda will have it's 3rd as well as 4th index values represented as numeric 1s which are corresponding to the row indices in the table.

b. Natural Language Processing: Commonly abbreviated as NLP , it is a subfield of computer science, information engineering and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. Challenges involving NLP are speech recognition, machine translation and natural language understanding. A corpus of musical notes are used in our project, using regular expressions and tokenizers we create a vocabulary of notes in vectorized format. Every note is capitalized , stemmers are removed and lemmatization is done to clean the midi data.

c. Word embedding: It is a collective name for a set of language modeling and feature learning techniques in NLP where words or phrases from the vocabulary are mapped to vectors of real numbers.Conceptually it involves a mathematical embedding from a space with one dimension per word to a continuous vector space with a much lower dimension. The underlying input representations of these embeddings boost performance in NLP tasks.

3.4 ALGORITHM

Step 1: The data is collected from string sounds and multisampled piano from Salamander Grand Piano V3 by Alexander Holm.The music21 converter is used to convert and parse a song in the midi format into musical notes.

Step 2: Then using an external converter, lilypond and a 3 layered basic rnn model the notes are converted into a mxl file which is further converted into a pdf format file to display the musical notation, given the pitch and duration of a sound, a pitch class is represented and saved.

Step 3: Based on the harmony each note is discretized and is imputed a particular tempo-time duration depending on its frequency. These notes are then fed to a pretrained model,here it is MelodyRNN which extracts the features.The model consists of an encoder and decoder with 3 embedding layers each and corresponding LSTM networks.

Step 4: The notes are tokenized and given a word2vec representation based on the vocabulary of notes which the network was pre-trained on.These notes are insofar one hot encoded and fed to the model which using sampling randomly picks a note and finds a mapping or a latent representation in its features so that an efficient forthcoming sequence of notes can be found.

Step 5: The one with the highest probability is chosen and through dynamic time warping keynote detection the following notes are played on the keyboard.Given the conditional independence of the preceding notes the maximum likelihood is calculated, hence as the number of notes played on the keyboard augments, the accuracy and therefore the F1 score of the model increases.

3.4.1 - A Recurrent Neural Network (RNN) is a class of artificial neural network used in deep learning applications where connections between nodes form a directed graph along a sequence. RNNs can use their internal state(memory) to process sequences of inputs. **Figure 3.3**. shows a typical RNN cell where the network "unrolls", showing at every time step an input and an activation function to fed in to the "future" nodes. Hence it is similar to a hidden markov model every output is not only dependent on the present state but also the past states of the input.

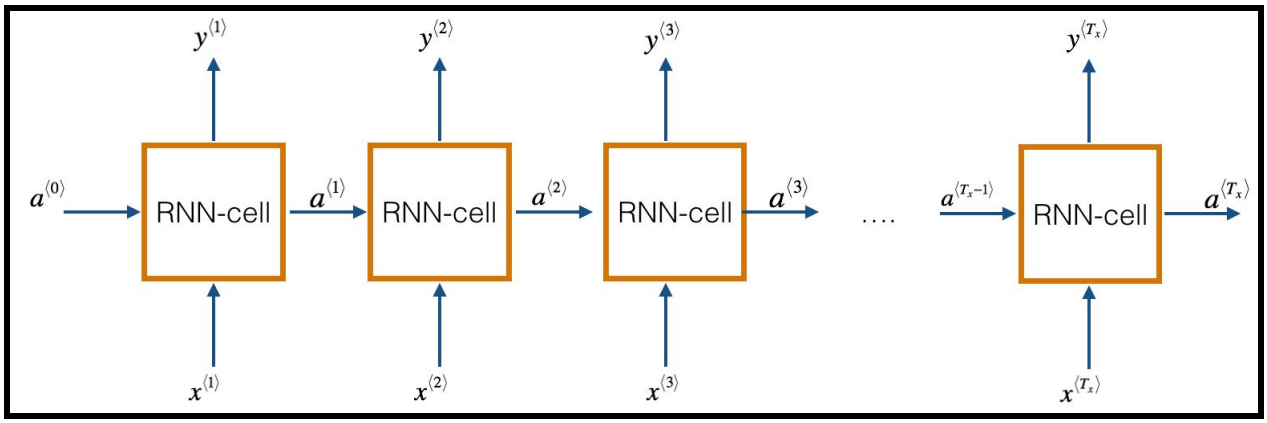


Figure 3.3 A Recurrent Neural Network (RNN) Cell

3.4.2 - Long short-term memory (LSTM) units are units of a recurrent neural network (RNN). **Figure 3.4** shows an RNN composed of LSTM units is often called an LSTM network. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. **Figure 3.5** shows the cell that remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

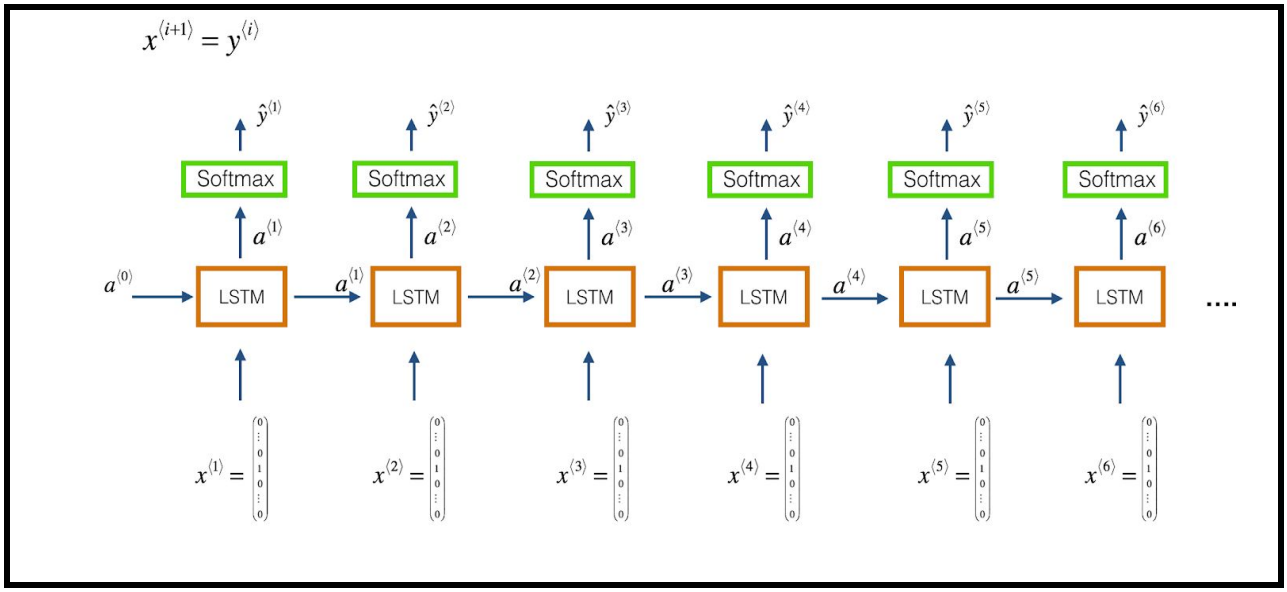


Figure 3.4.- One-to-many LSTM (Long Short Term Memory) cell

3.4.3- We use the logits- z(t) and r(t) that are linear functions in **Fig 3.5**.The memory cell hidden previous states h(t-1).Future hidden state h(t) , three multipliers and one adder that play the role of valves in the lstm cell.The leftmost multiplier is the input gate valve,the central is the forget gate valve and the rightmost is the output gate valve. Sigma and tanh are activation functions that confine the output sample space depending on their range and channelize the final output.

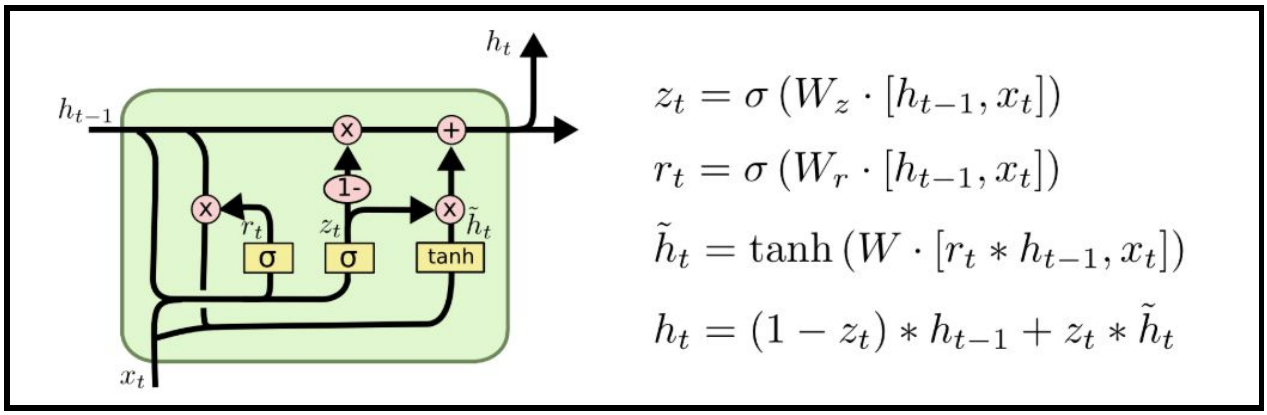


Figure 3.5- LSTM with equations

There are two approaches to feed data into the model,one is deterministic and the other is stochastic**. Figure 3.4** shows the latter method where the labels are fed in with the input data and the estimated output is given(Note that the output isn't fed back to the next time step).**Figure 3.6** shows the approach we have incorporated in our model where the labels aren't fed in, instead the estimated output is taken the and fed into the subsequent time steps.Although random or

stochastic processes sound absurd at first,they are way powerful and pragmatic as compared to deterministic or omniscient models in the real world.

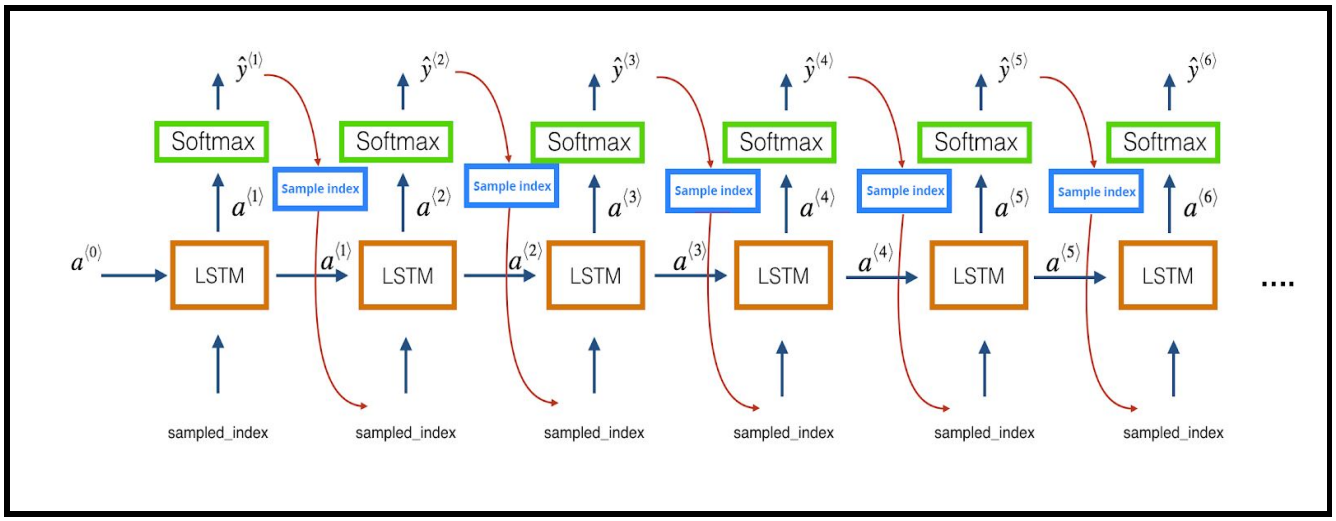


Figure 3.6- Sampling methodology in a neural network

3.4.4- Sequence-to-Sequence model as shown in **Fig 3.7** - reads a sequence (such as a sentence) as an input and produces another sequence as an output. It differs from a standard RNN in that the input sequence is completely read before the network starts producing any output. Typically, seq2seq models are implemented using two RNNs, functioning as encoders and decoders.In our model we have used the lookback RNN as the encoder and the attention RNN as the decoder.

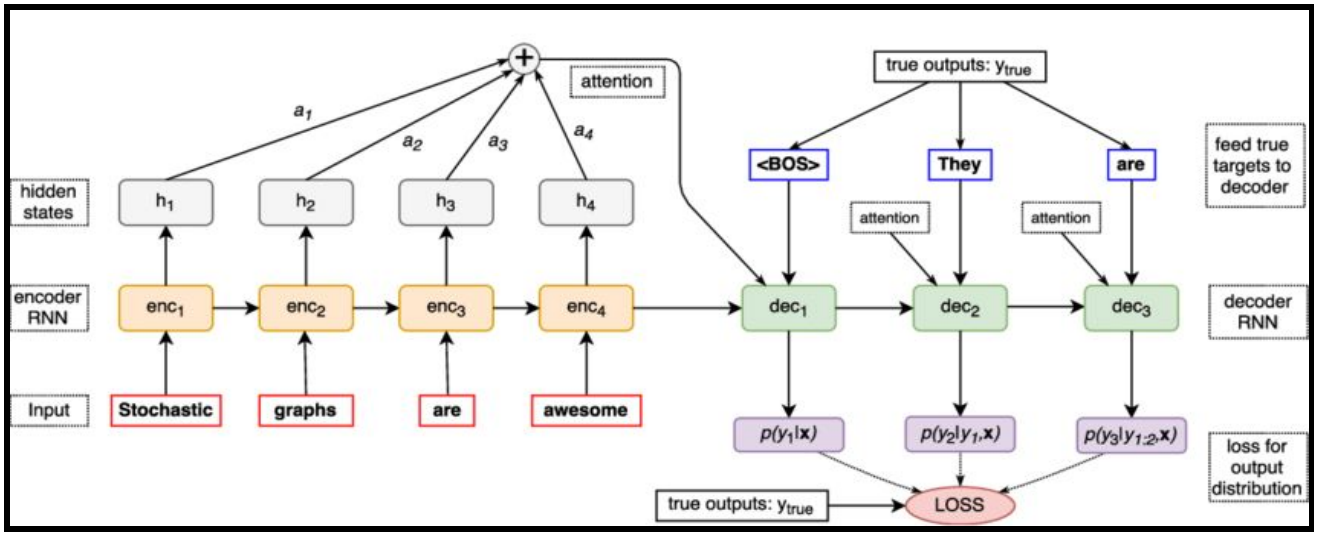


Figure 3.7- Sequence-to-sequence Model

4. RESULTS AND DISCUSSIONS

4.1: Statistical analysis results for five different songs (varying on the basis of the repetition, range of notes present and the speed)

Name of the Song	Notes played	Probability/Accuracy Of Subsequent Notes
A Little Dream	G4-F#4-G4-F#4-E4	80%
Nursery Rhyme-Mary had a little lamb	E-D-C-D	55%
Moonlight Sonata	G#-C#-E-G#-C#-E	65%
Ode to Joy-Beethoven	B-B-C-D-D	60%
Time to say Goodbye	D-G-F#-A-G	71.40%

Figure 4.1- STATISTICAL ANALYSIS OF RESULTS

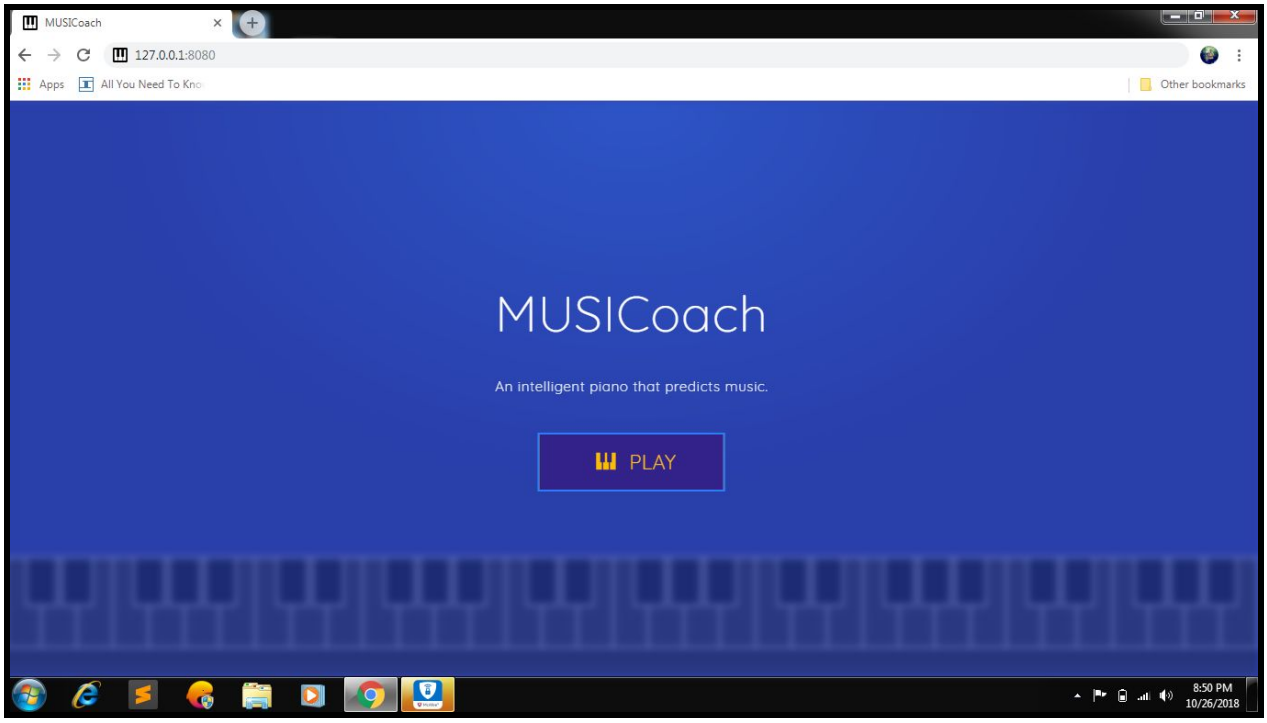
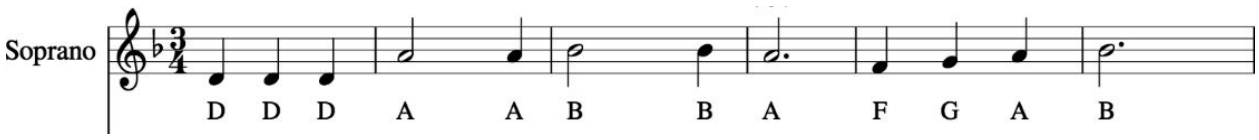


Figure 4.2- WEB PAGE- USER INTERFACE



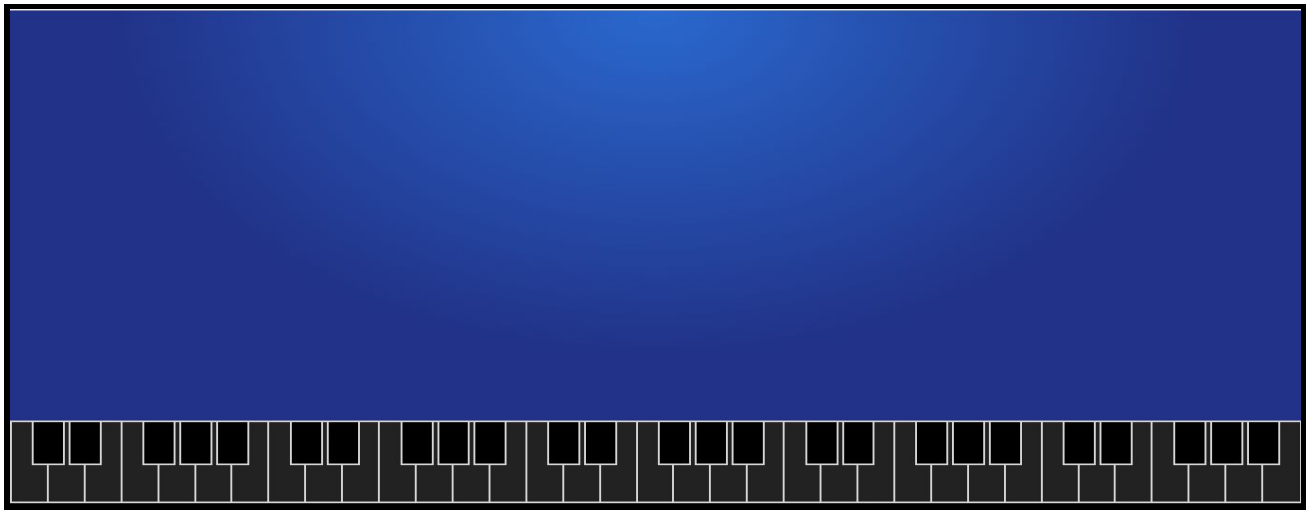


Figure 4.3- VIRTUAL KEYBOARD

4.2: Discussion- In this project, we have designed an algorithm that detects a structure in the musical composition and accordingly displays the notation of that composition. The system has been designed as a user-friendly interface where a tyro can understand the fundamentals of playing a keyboard. Using the Melody RNN model for predicting notations, we have trained the algorithm such that it can be used to feed in a song of the user's choice and the notation for the same will be displayed with an above-par accuracy (about 75-80%). This algorithm serves as a virtual instructor for the same.

5. APPLICATIONS

- The algorithm would help train students in music schools by enabling them to gain a comprehensive understanding of every musical composition with the study of musical notes involved.

- This algorithm would also lend itself to the music industry where it could serve as an assistant for music composers while developing newer harmonies/melodies as it would help the composers figure out an idiosyncratic pattern for every composition.

6. FUTURE SCOPE

One of the cardinal traits of a dexterous pianist is his/her ability to predict the notations of a musical composition which can enable them to play music on the go. This algorithm could serve to be a virtual music assistant to composers in the musical industry around the world where newer songs in diverse forms are the norm of the day. The application would further be a hassle-free, low cost substitute for nascent pianists in a world where erudite musical teachers are hard to locate. We have applied the principles of learning using the impeccable functionality of the Recurrent Neural Networks (RNN) alongside other supporting methodologies like LSTM and Sequence-to-sequence Modelling.

7. REFERENCES

[1] Qi Wang, Ruohra Zhon, Yonghing Yan- A two stage approach to note-level transcription of a specific piano- (2017)
[2] Jean Francois Paiement, Samy Bengio, Douglas Eck- Probabilistic Models for Melodic Prediction- (2009)
[3] Douglas Eck, Jasmine Lapalm- Learning Musical Structure directly from the sequences of music- (2008)
[4] Matija Marolt, Alenka Kavcic, Marko Prinovic- Neural Networks for Note Onset Detection in Piano Music- (2002)
[5] Douglas Eck & Jurgen Schmidluber- Finding Temporal Structure in Music: Blues Improvisation with LSTM recurrent Network – (2002)
[6] Michael C. Mozer -Neural Network Music Composition by Prediction – Exploring the benefits of Psychoacoustic constraints and multi-scale processing– (1994)