**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

Jnana Sangama, Belgaum- 590 018

**A Project Report On**

# "INSTRUMENTAL MELODY GENERATION USING DEEP LEARNING "

*Submitted in the partial fulfilment of the requirements for the award of the Degree of*

**Bachelor of Engineering in Computer Science and Engineering**

*submitted by*


**AMOGH G KOTABAGI [1DB18CS013]**

**BHARATH R [1DB18CS030]**

**ARUN KUMAR E [1DB18CS021]**

**CHETAN CHIRAG K H [1DB18CS041]**


*Under the guidance of*


**Prof. Dr. Nagaraja P**

Department of CSE

**Don Bosco Institute of Technology**

**Mysore Road, Kumbalgodu, Bengaluru-560074**


**2021-2022**

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DON BOSCO INSTITUTE OF TECHNOLOGY**

**Mysore Road, Kumbalgodu, Bengaluru-560074**

# CERTIFICATE

This is to certify that the project Phase-II entitled **“INSTRUMENTAL MELODY GENERATION USING DEEP LEARNING”** is carried out by **Amogh G Kotabagi [1DB18CS013], Bharath R[1DB18CS030], Arun Kumar E[1DB18CS021], Chetan Chirag KH [1DB18CS041]** are bonafide students of **Don Bosco Institute of Technology, Bangalore** in partial fulfillment for the award of the degree of **Bachelor of Engineering in Computer science and Engineering** of **Visvesvaraya Technological University, Belagavi** during the academic year **2021-22**. The project Phase -II report has been approved as it satisfies the academic requirements in respect of the Project Phase II prescribed for the Bachelor of Engineering Degree.

| | | |
|---|---|---|
| **Dr. Nagaraja P** | **Dr. K B Shiva Kumar** | **Prof. B S Umashankar** |
| Professor | Professor & Head | Principal |
| Guide, Department of CSE | H.O.D., Department of CSE | Principal, D.B.I.T |
| D.B.I.T., Bangalore-74 | D.B.I.T., Bangalore-74 | Bangalore-74 |

**External Viva**

Name of Examiner Signature with date

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DON BOSCO INSTITUTE OF TECHNOLOGY**

Mysore Road, Kumbalgodu, Bengaluru-560074

# DECLARATION

We, **Amogh G Kotabagi [1DB18CS013], Bharath R [1DB18CS030], Arun Kumar E [1DB18CS021], and Chetan Chirag KH[1DB18CS041]** students of eighth semester B.E, at Department of Computer Science and Engineering, Don Bosco Institute of Technology, Bengaluru declare that the project phase-II entitled "**INSTRUMENTAL MELODY GENERATION USING DEEP LEARNING**" has been carried out by us and submitted in partial fulfillment of the course requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering discipline of Visvesvaraya Technological University, Belagavi during the academic year **2021-22**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree.

**Place: Bangalore**

**Date:**

# ACKNOWLEDGEMENT

The satisfaction and euphoria the successful completion of any project is incomplete without the mention of people who made it possible, whose constant guidance and encouragement made our effort fruitful.

First and foremost, we ought to pay our due regard to this institute, which provided us a platform and gallowed displayingour skills through the medium of project work. We express our heartfelt thanks to our beloved principal **Prof. B S UMASHANKAR**, **Don Bosco Institute of Technology, Bangalore** for his encouragement all through our graduation life and providing us with the infrastructure.

We express our deep sense of gratitude and thanks to **Dr. K B SHIVA KUMAR, Head of the Department, Computer Science and Engineering** for extending his valuable insight and suggestions offered during the course.

It is our utmost pleasure to acknowledge the kind help extended by our guide **Dr. NAGARAJA P**, **Professor**, **Department of Computer Science and Engineering**, for guidance and assistance which consequently resulted in getting the project phase II work completed successfully.

Last but not the least I would like to thank teaching and non-teaching staff for their cooperation extended during the completion of the project Phase-II.

**Amogh G Kotabagi [1DB18CS013]**

**Bharath R [1DB18CS030]**

**Arun Kumar E [1DB18CS021]**

**Chetan Chirag KH [1DB18CS041]**

# ABSTRACT

Traditionally, music was treated as an analog signal and was generated manually. In recent years, music is attracted to technology that can generate a suite of music automatically without any human intervention. To accomplish this task, we need to overcome some technical challenges which are discussed descriptively in this project. The focus of this project is used to generate musical notes using Recurrent Neural Networks (RNN) with the help of Long Short-Term Memory (LSTM) networks.

A model is designed to execute this algorithm where data is represented with the help of a musical instrument digital interface (MIDI) file format for easier access and better understanding. Preprocessing of data before feeding it into the model used in this paper is used to learn the sequences of polyphonic musical notes over a single-layered LSTM network. When the model was thoroughly analyzed, it produces stellar results in composing new melodies.

# TABLE OF CONTENTS


| | |
|---|---|
| **Acknowledgment** | **I** |
| **Abstract** | **II** |
| **Table of Contents** | **III** |
| **List of Figures** | **IV** |

| | | **CHAPTERS** | **Page No** |
|---|---|---|---|
| **1.** | | **INTRODUCTION** | **1** |
| | 1.1 | Overview | |
| | 1.2 | Existing System | |
| | 1.3 | Proposed System | |
| | 1.4 | Objectives | |
| | 1.5 | Problem Statement | |
| **2.** | | **LITERATURE SURVEY** | |
| **3.** | | **DESIGN AND IMPLEMENTATION** | |
| | 3.1 | (Architecture / block diagram ) | |
| | 3.2 | (Algorithms can be included) | |
| | 3.3 | Software requirement specification | |
| **4.** | | **RESULT AND PERFORMANCE ANALYSIS** | |
| **5.** | | **CONCLUSION** | |
| | | **REFERENCES** | |

# LIST OF FIGURES


| FIGURE NO. | TITLE | Page No |
|---|---|---|
| 1.1 | Neural Network | |
| 1.3 | Recurrent Neural Network | |
| 3.1 | System Archietecture and design | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# INSTRUMENTAL MELODY GENERATION USING DEEP LEARNING

# Chapter 1

# INTRODUCTION

## 1.1 OVERVIEW

**Machine Learning and Deep Learning**

ML is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithm to imitate the way that humans learn, gradually improving its accuracy.

Deep learning is a subset of Machine learning that imitates the way neurons in the human brain process. The1 main process of deep learning is that it takes input in the input layer and process them in hidden layer and then gives the desired output in output layer.

There are 3 types of the neural network:

1. Artificial Neural Network (ANN).
2. Convolutional Neural Network (CNN).
3. Recurrent Neural Network (RNN).

In this paper, we are going to be focusing on Recurrent Neural Network (RNN).

ANN is mainly used to solve a problem related to Tabular data, Time data, and Text data.

RNNs are mainly used to solve the problem related to Time series data, Text data, and Audio data.

CNN is mainly used to solve problems related to Image classification, Image recognition.

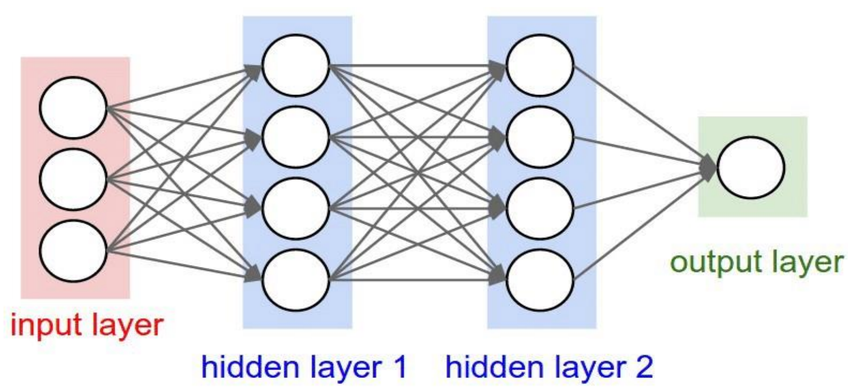



Fig 1.1 Neural Network

Our Idea mainly focuses on Generation Melody automatically using a Recurrent Neural Network (RNN). Nowadays people love to hear a melody, which gradually reduces stress and anxiety and helps them to continue their daily routine and also keeps them indulging in their day-to-day lifestyles.

## 1.2 EXISTING SYSTEM

Melody generation is a topic that has been studied in much detail in the research industry in the past We have a convolution GAN model that directly creates binary-valued instrumental rolls by using binary neurons, but it had low efficiency because every iteration process of it has a delay. In addition, due to lack of context, it is difficult to get the coherence and deep-seated rhythm information.

## 1.3 PROPOSED SYSTEM

To overcome the deficit in the existing system we are using RNN that too specifically LSTM architecture-based RNN. RNN is used as it exhibits temporal dynamic behavior. RNNs can use internal memory to process a sequence of input of any length. Unlike CNN it uses feedback connections, such that it could process 'entire' sequences of data. This helps in remembering the context of the given sequence.

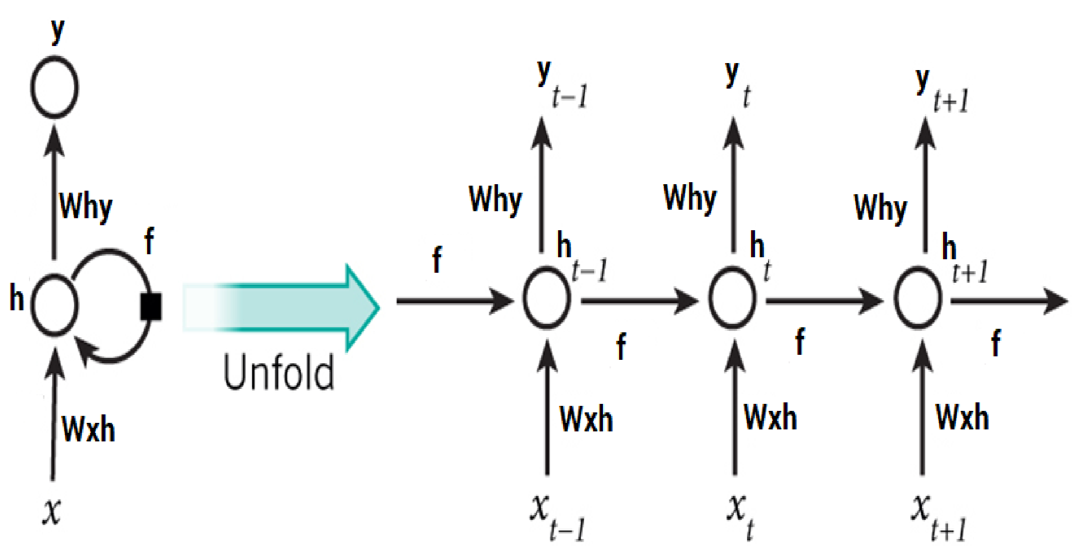



**Fig 1.3 Recurrent Neural Network**

## 1.4 OBJECTIVES

To generate musical melodies using Deep Learning without any Human Intervention
A model is designed where data is represented with the help of Musical Instrument Digital Interface (MIDI) file format for easier access and better understanding.

## 1.5 PROBLEM STATEMENT

Create a Model which gives out tunes that are Rhythmic and Melodious to listen to, where the output of the model should be dependent on the input tune given by the user. Also, this model shouldn't need any human interaction in-between the processing.

# CHAPTER 2

# LITERATURE SURVEY

| Sl No. | Title | Author | Work Description | Deficiency |
|---|---|---|---|---|
| 1 | WAVENET: A GENERATIVE MODEL FOR RAW AUDIO | Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu 2016, eprint arXiv:1609.03499 | This paper introduces WaveNet, a deep neural network for generating raw audio waveforms. The model is fully probabilistic and autoregressive, with the predictive distribution for each audio sample conditioned on all previous ones. | The user did have had any control over generation music like pitch ,tones time interval, such as interpolation, attribute vector |
| 2 | Compress to Create: Autoencoder | Jean-Pierre Briot,2020, Laboratoire d'Informatique de Paris 6 | The motivation is in using the capacity of modern deep learning architectures and associated training and generation techniques to automatically learn styles from arbitrary corpora and then generate samples from the estimated distribution, with some degree of control over the generation. | The user had more control over such as interpolation, attribute vector arithmetics, recursion and objective optimization, as will be illustrated by various examples. But took up too much time to generate |

| | | | | |
|---|---|---|---|---|
| 3 | Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation | Hao-Wen Dong, Yi Hsuan Yang 2018, ISMIR | In this paper, we study whether we can have a convolutional GAN model that directly creates binary-valued piano rolls by using binary neurons. Specifically, we propose to append to the generator an additional refiner network, which uses binary neurons at the output layer. | Had low efficiency because every iteration process of it has a delay. In addition,due to the lack of context, it is difficult to get the coherence and deep-seated rhythm information. |

# CHAPTER 3

# DESIGN AND IMPLEMENTATION

## 3.1 Architecture / Block Diagram

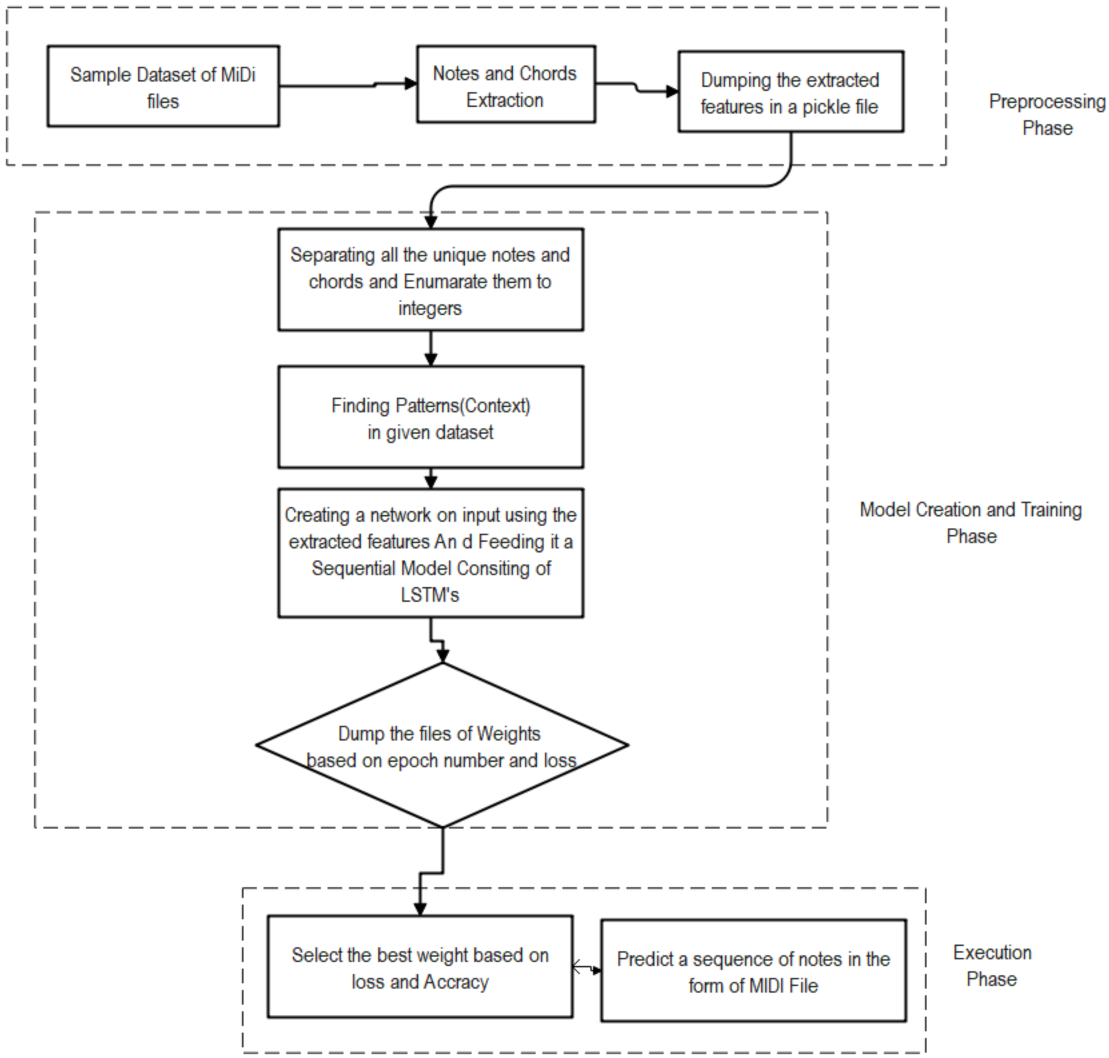



**fig 3.1 .1system architecture and design**

We use a sample Dataset consisting of songs in MIDI files. These files are then parsed using a Music21 library to get the Notes and Chords of each song. Now we dump this extracted content into a pickle file. This was Preprocessing phase. By separating all the unique notes and chords and Enumerating them into integers then we are going to find the patterns in the given dataset. So basically following this process will give us the model with its weights which are based on epoch number and loss.

Finally, we select the best weights based on loss and accuracy and predict a sequence of notes in the form of a MIDI file.

```
lstm_1 (LSTM)                (None, 100, 512)          2099200

lstm_2 (LSTM)                (None, 512)               2099200

batch_normalization (BatchN  (None, 512)               2048
ormalization)

dropout (Dropout)            (None, 512)               0

dense (Dense)                (None, 256)               131328

activation (Activation)      (None, 256)               0

batch_normalization_1 (Batc  (None, 256)               1024
hNormalization)

dropout_1 (Dropout)          (None, 256)               0

dense_1 (Dense)              (None, 83)                21331

activation_1 (Activation)    (None, 83)                0

=================================================================
Total params: 5,406,803
Trainable params: 5,405,267
```

Fig 3.1.2 Model summary

In our model we use four different types of layers:

**LSTM layers** are a Recurrent Neural Net layer that takes a sequence as an input and can return either sequence (return_sequences=True) or a matrix.

**Dropout layers** are a regularisation technique that consists of setting a fraction of input units to 0 at each update during the training to prevent overfitting. The fraction is determined by the parameter used with the layer.

**Dense layers** or fully connected layers are a fully connected neural network layers where each input node is connected to each output node.

**The Activation layer** determines what activation function our neural network will use to calculate the output of a node.

## 3.2 Algorithm

**Input**: European folksong dataset, New unlabelled MIDI files (MIDI File Format)

**Output**: Sequence of nodes which are melodies and in context with the input melody (MIDI Files).

**Step 1**: Pre-Process the dataset

Read the folk from MIDI File format.

Remove Songs that have nonacceptable durations.

Encode the songs with music time series representation.

Save the song to a text file.

**Step 2**: Generate training sequence

Sorts all the possible input to one variable and Its respective target to another variable.

Encoded music will be converted to an integer.

Split the dataset into training and testing datasets.

Perform one-hot encoding.

**Step 3**: Build an RNN which implements LSTM

Using Keras/TensorFlow.

**Step 4**: Evaluate the model with the Testing set

Its Accuracy is not acceptable to go to step 2.

**Step 5**: Pre-process the output from the model

Decode the output to an array of integers.

Convert integer back to symbols and export them back as the midi file format.

**Step 6**: You can play the midi file using midi player such as MUSESCORE.

## 3.3 Software Requirement Specification

**HARDWARE REQUIREMENTS:**

- Processor : Inteli5 or more
- Processor speed : 2.4GHz
- RAM : 4 GB
- GPU : 4GB NVIDIA1050 TI
- Hard Disk Space: 128GB

**SOFTWARE REQUIREMENTS:**

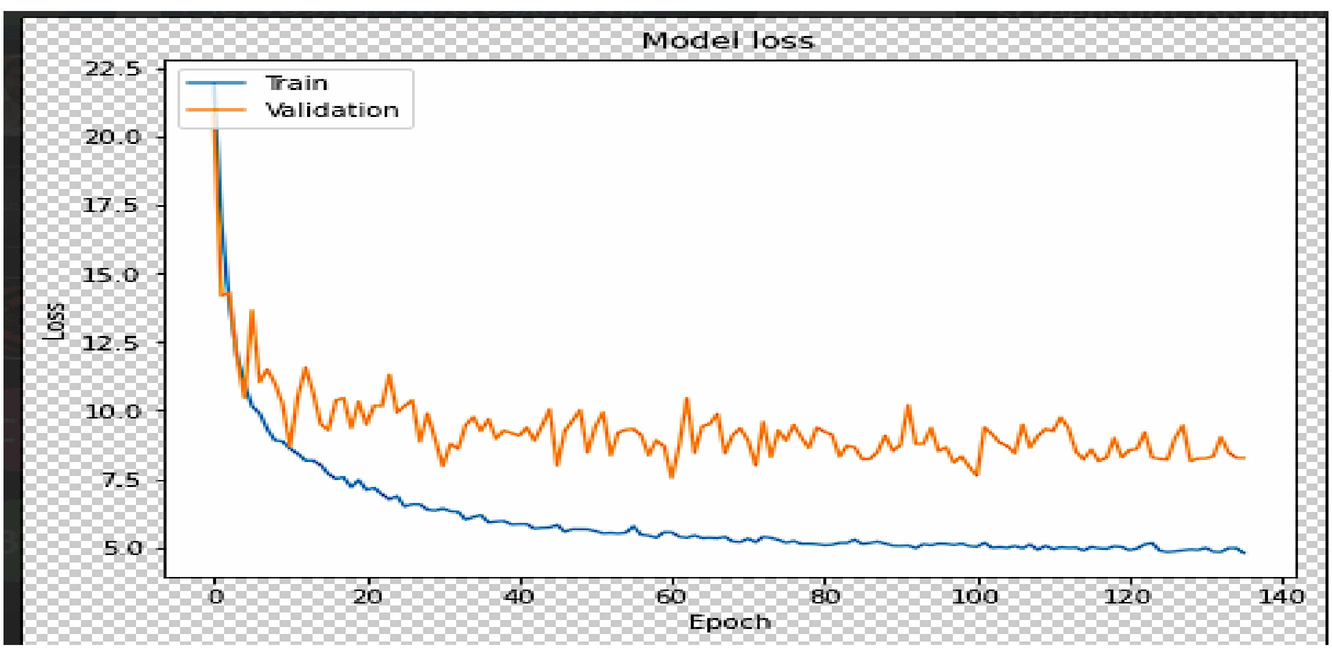
- Operating system: Windows7/8/10
- Coding Language: Python 3.0
- Software Tool:Jupyter Notebook.

## CHAPTER 4

# RESULTS AND PERFORMANCE ANALYSIS

```
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 at last epoch level the categorical cross-entropy is 0.6020 and the Categorical is around 83%and the predicted sequence is within the context of the Trained dataset. we could greatly increase the accuracy of the model by increasing the number of data points within a context and the GPU capacity. Although the predicted output is on par with the context of the input it is inferior in comprehending full knowledge of the input like patterns and offsets. We have only trained the model with one hour of midi data from the Authors like Clementi,Mozart, Subert, and Brahms.

# CHAPTER 5

# CONCLUSION

We have built a tool to generate melody with the MAESTRO dataset that contains piano songs. We preprocess it, train our neural network model, then generate the melody with it. The melodies are in MIDI format. We use TensorFlow v2.0 to do it. I think TensorFlow v2.0 User Experience (UX) is better than its previous version.

The melody generated by our model is also coherent and good to hear. It can adjust how it plays its notes. For example: When the generator from a note (means it is a start of the music), it starts with a slow tempo start.

There are some things that we can try for the music generator. In this article, we have experimented with generating a single instrument. What if the music has multiple instruments? There needs to be a better architecture to do it. There are multiple things that we can try on experimenting with music data.

# REFERENCES

**[1]** MUSIC GENERATION USING DEEP LEARNING -Dr.P.SHOBHA RANI1, S.V.PRANEETH2, VATTIGUNTA REVANTH KUMAR REDDY3, M.J.SATHISH

**[2]** LSTM Based Music Generation System -Sanidhya Mangal, Rahul Modak, Poorva Joshi

**[3]** LSTM-based Music Generation (2019) -Falak Shah, Twisha Naik, Nisarg Vyas,InFoCusp Innovations Private Limited, Ahmedabad, India

**[4]** NeuralFunk - Combining Deep Learning with Sound Design | by Max Frenzel, PhD | Towards Data Science

**[5]** Wikifonia Dataset." http://www.synthzone.com/files/Wikifonia/Wikifonia.zip. [Online; accessed 19-July-2019].

**[6]** "Summaries and Tensorboard." https://www.tensorflow.org/guide/summaries and tensorboard. [Online; accessed 19-July-2019].

**[7]** "Magenta Documentation." https://magenta.tensorflow.org. [Online; ac-cessed 30-July-2019].

**[8]** Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley,"Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training," arXiv preprint arXiv:1907.04868, 2019.

**[9]** ''MIDI Documentation." https://mido.readthedocs.io/en/latest/aboutmidi.html. [Online; accessed 19-July-2019]

**[10]** F. Furukawa, "Fundamentals of Music & Properties of Sound – Music Theory Lesson," 3 August 2015. [Online]. Available: https://popliteral.com/fundamentals-of-music-properties-of-sound-music- theory-lesson/. [Accessed 7 November 2018]G.

**[11]** A. N. V. A. S. J. A. K. A. Joshi, "A Comparative Analysis of Algorithmic Music Generation on GPUs and FPGAs," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 229-232, 2018.

**[12]** G. a. F. N. Hadjeres, "Interactive Music Generation with Positional Constraints using Anticipation-RNNs," arXiv preprint arXiv:1709.06404, 2017.

**[13]** P. Y. Nikhil Kotecha, "Generating Music using an LSTM Network," arXiv.org, vol. arXiv:1804.07300, 2018.

**[14]** S. Mangal, "Music Research," 13 November 2018. [Online]. Available: https://gitlab.com/sanidhyamangal/music_research.

**[15]** "Musical_Matrices," 10 July 2016. [Online]. Available: https://github.com/dshieble/Musical_Matrices/tree/master/PopMusic_Midi. [Accessed 2 November 2018]

# DETAILS OF THE STUDENT

**BATCH – 2018-2022**

| | | | | |
|---|---|---|---|---|
| NAME: | | | | |
| | | | | |
| USN: | | | | |
| | | | | |
| PHONE NUMBER: | | | | |
| | | | | |
| EMAIL ID: | | | | |
| | | | | |
| PLACED – IN: | | | | |
| | | | | |
| | | | | |
| PERMANENT ADDRESS | | | | |