Composition and production of music using Momentum LSTM

MSc Research Project
Data Analytics

Sachin Nikam
Student ID: x19198159

School of Computing
National College of Ireland

Supervisor: Dr. Rashmi Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Sachin Nikam
Student ID:	x19198159
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr. Rashmi Gupta
Submission Due Date:	16/08/2021
Project Title:	Composition and production of music using Momentum LSTM
Word Count:	7049
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sachin Nikam
Date:	10th October 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Composition and production of music using Momentum LSTM

Sachin Nikam
x19198159

Abstract

Music plays important part in the field of media and entertainment.Composing and producing music requires high practical and theoretical knowledge which needs training and high skills. Knowledge of instruments becomes necessary and key factor while composing music. This becomes difficult for non-musicians or amateurs artists. This can be fulfilled by training the machines to do so. Use of Machine learning and neural network models is used to produce music from the training data. Multiple models were used previously. LSTM and GRU recurrent neural networks were implemented. Adding momentum to the LSTM can improve the model and predict better accuracy. The paper shows the Momentum LSTM which selects the important features from the midi dataset, which contains different musical features. Purpose of the research shows the use of Momentum LSTM Recurrent Neural Network to train on the Midi files and produce new composed files.

1 Introduction

Music is one of the most important feature which plays major role in any of the media or entertainment field. Music is almost used in most of the media work including movies, games, short videos and advertisement. Machine Learning algorithms may be used by the advertising business to increase and forecast the effectiveness of their commercials, as well as execute product placements utilizing analytic and identify product connections. The same techniques are used in other mentioned industries. Out of which music industry independently or dependent on mentioned industries plays a major role for creating interactive content. Music directly or indirectly is related to media industry to gain maximum profit and revenue. Music is also used as a reference to a particular character or a movie. As an example is that how people can recognize a particular song or a movie by just listening to a soundtrack or a introductory music. A good music composition or soundtrack have a high correlation to make a movie or advertisement to make powerful scene. Generating and composing music is one of the most time consuming task and requires practical and theoretical study of the music instruments and musical languages. In field of Machine Learning and Artificial intelligence, machine have capabilities to think and work like humans. There has been great achievement integrating music technology with Machine Learning and statistical algorithms.

Multiple research shows notable improvement in the field of music generation. Use of Recurrent Neural network and other statistical approaches had been practised for many years to gain better output quality. Initially when the concept of music generation came in to existence, at that time the computation power of the computers were not that great which lacked the better performance of the output. With the increase in the technology and processing speed of the computers improved statistical and machine learning algorithms showed better performance and improvement in generating music files. Big and large companies such as Google, Spotify and Openai had launched their separate projects to solely improve work on this field. Google launched the project named "Google Magenta"[1] where team work on the special techniques to create music. In the same way Spotify too have there own labs names CTRL Labs. Openai specialised for Artificial intelligence projects have the project name "Jukebox"[2] which shows the similar work of generating music using Machine Learning algorithms and Recurrent Neural Networks. During the past few years Long Short Term Memory (LSTM) which are Recurrent Neural Networks showed much improvement in generating music.

The motivation for the research comes from the fact that some of the artist lack knowledge to create the music. There are many branches in the field of music as well. Some of the branches only deals with the technology for example practice where one need to handle Digital Audio Workstations(DAW), a tool to automatically create the music by using some internal or external plugins. Study of DAW is considered very different from learning any musical instruments. Producer need some of the knowledge of Music theory to understand the interface or compose music.Sometimes a composer is not able to start the initial work on music as artist need to write new and unique composition.Though they have great knowledge in music but still they need some reference to start with their work. This is called "creative block". This is where research is focused where generating the music by machines will be easy and does not much of the effort to have some unique files.

Research focuses on using Momentum LSTM, recurrent neural network to generate the unique music files sounding more like human composition. The files which are used in the project are Midi (Musical Instrument Digital Interface) files. These files are feed to train neural networks by using the music theory song structure "Circle of Fifths" which are used to create songs by musicians.

Figure 1 shows the structure of Circle of fifths. "Circle of fifths" [3]. Circle of fifths is a musical structure which helps musicians to manually organize 12 chromatic pitches in a form of sequence. Scales are formed by taking first note from any 12 chromatic notes and are arranged according to the majors or minors to form the scales.

[1] Google Magenta: https://magenta.tensorflow.org/

[2] Jukebox by openai: https://openai.com/blog/jukebox/

[3] https://appliedguitartheory.com/lessons/circle-of-fifth

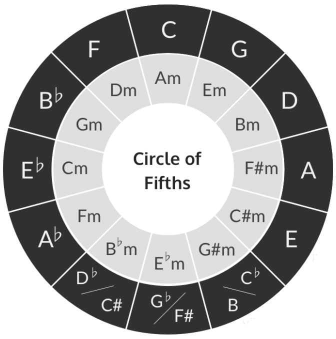


Figure 1: Circle of Fifths

Momentum LSTM will create unique music files which will be evaluated by the human participants. Participants are group of musicians and non-musicians who will response by the means of survey including questionnaires on generated musical files. This follows the research question how well the Momentum LSTM's will work on the midi dataset to generate musical files by adding Momentum to LSTM's network.

Research focuses on composing and generating audio files by using RNN's LSTM and adding Momentum optimizer from Stochastic Gradient Descent to generate music files. Research follows multiple phases and sections. The paper discusses the previous works in section.2, Methodology in section.3 which shows selection of data, pre-processing and extracting patterns and implementation of model in section 5 and evaluating result in section.6.

2 Related Work

Multiple work has been done on the generation of music using neural networks and other statistical models. Some of the research shows better output accuracy when evaluated. To conduct and understand this research in detail, previous work is critically reviewed in this section. Work is discussed on the basis of progress made on the domain and divided in the section.

Training neural network to generate music requires both statistical knowledge of Machine learning as well as music theory. Early work by (Moorer; 1972) showed the generation of short fragments of western notes. Initially to train the machines using statistical models was difficult task so output does not showed much competitive results. The problem on this task was analysed and research was concluded (Sigtia et al.; 2015). Music Language Models were combined with Recurrent Neural Network models to provide a frame by frame classifier for acoustic sounds. Study conducted by (Boulanger-Lewandowski et al.; 2014) shows architecture for hybrid model that combines a phonetic model with an arbitrary frame-level acoustic model, present fast methods for training, decoding, and sequence matching.Study by (Sturm et al.; 2016) used Deep Learning methods, particularly LSTM models,for transcription and composition, according to research. To create new transcriptions, the model was trained on a huge data of 23,000 music transcriptions. The dataset was trained, and four modes (major, minor, Dorian, and Mixolydian) were created. The dataset was ran through a "char-run" model, and the results were shared with the online community for review.

After the progress in the field of Deep Learning and statistics, statistical models and Recurrent Neural networks were largely used to improve the accuracy. Finding relevant characteristics in the data and utilizing those features to train RNNs was the initial work. The study by (Eck and Schmidhuber; 2002) demonstrates how the LSTM model was used in this area and how it was successful taking timings and count stamps. According to research, LSTM may potentially be used to compose music. To generate music, the model performed effectively in producing distinct pleasant sounds with organized time stamps on the metronome when blues kinds of music were utilized. With machines with great processing capacity, there has been tremendous progress in the area of statistics. (Conklin; 2003)created audio using a variety of statistical algorithms. In this study, a random walk and a hidden Markov model were employed. Sampling approaches such as stochastic and pattern-based sampling are also being studied. The study proposes using a complicated statistical model, and HMM (Hidden Markov model) demonstrates progress in producing some intriguing patterns throughout this investigation. (Schulze and Van Der Merwe; 2011) findings show that timestamps and time signatures were extracted more accurately. HMM can store these characteristics in memory by grouping them in order.

There isn't just one instrument in a music file. A single audio file is created by combining multiple instruments to generate polyphonic music. Deep learning was utilized to extract, identify, and categorize musical instruments in research conducted by (Li et al.; 2015). This objective is accomplished with the help of a Convolutional Network. The model was trained on a dataset of mixed recordings, and CNN classifies individual instruments such as electric bass, acoustic guitar, violin, piano, and so on into 11 different categories. In order to find the minimum loss, stochastic gradient descent is utilized. When compared to other models, this model has a higher accuracy of 82.74 percent. This study opened the doors studies with raw audio data.

Deepmind used CNNs to come up with MIDINET (Yang et al.; 2017), which was motivated by WAVENET (Oord et al.; 2016). The study was related to Google Magenta research which employed LSTMs to generate sequences in music and tales.Midi files are split into bars in MIDINET research. A Deep convolutional generative adversarial network was the primary network of MIDINET (DCGAN). To evaluate the outcome, listeners were given the produced files and asked to categorize the music into three categories. How pleasing? how real? and how interesting?. According to the findings, the music defies music theory principles, yet when the chord progressions were captured, the distinction between major and minor scales was identified. Some files were actual files, while others were made by models, according to listeners. Some listeners classify machine-generated files as originals, resulting in improved model performance.

As the research progressed, the outcomes of applying Deep Learning approaches improved. The comparison of Markov models with Deep Learning approaches was showed in a survey done by Jean-Pierre in 2017 and updated in 2019 (Briot et al.; 2017).Objectives, Representation, Architecture, Challenge, and Strategy were the five dimensions considered in the analysis. Deep Learning models had greater learning capabilities than Markov models and can create more enjoyable tunes in the output, according to a detailed examination of each of the dimensions.

Multiple tracks with various instruments organized on rhythms make up polyphonic music. A study by (Dong et al.; 2018) demonstrates the use of GAN's (Generative Adversarial Network) architecture to generate multi-track music. The model was built using real-time techniques utilized by musicians. Jamming with previously created songs on various instruments and creating songs based on musical knowledge and experience with various instruments, three models were developed. Composer, jamming, and a hybrid model Deep CNNs were used to create multi-track piano rolls. Because the files comprise many instruments, the research is focused on data pre-processing. Because the data was noisy, the analysis relied on data from piano rolls. On the data, the hybrid model produces superior results and performance. This study also led to the development of a model for video generation.

(Roberts, Engel, Oore and Eck; 2018) found that hidden characteristics, also known as latent features, may be extracted from music. Music's colors, sometimes known as ”palettes,” and tones were created in order to interpret how music is created. The model, which consists of three parts: encoder, latent vector, and decoder, was built using auto encoders. As an interface, Google MusicVAE and the NSynth instrument were utilized. Using the encoding and decoding approach, the tool developed for this study generated music. The encoder was used to input a waveform audio file, and latent vectors were recovered and transmitted for transformations such as interpolation. The output of this was given to the decoder function, which generated an raw audio file as an output. Samples were coded into pads and utilized as Synthesizer for midi files in which palettes were created based on color tones and instrument sounds were created. It was determined that better music melodies and songs generated variety of unique sounds.

The work done by Google Brain (Huang et al.; 2018) was inspired by research on utilizing encoders and decoders, which led to additional inspiration and work on Transformers to provide unique results. Long structural music compositions were created using transformers with other algorithm changes. A matrix of voices and time stamps is included in the dataset, as well as a fine and granular level pieces of data files. The distance between keys and queries was adjusted in the algorithm since the transformer model concentrates on sinusoids and provides temporal data on the sequence of music and piece. With random samples, the dataset was divided into 80,10, and 10.Magenta's RNN model (Oore et al.; 2020) was used to produce monophonic music, and its performance was compared to that of Magenta's RNN model from 2018. Using this dataset, the Relative Transformer outperforms both the LSTM and the Transformer models.

A team of Google Brain researchers conducted a quick study (Dinculescu et al.; 2019) utilizing Pre-generative music models, in which users may directly enter their Midi files into the model, and the model would attempt to imitate their structure as a latent space. TensorFlow was used to create a standalone application that allows users to make music using their browser. ”Midime” was developed taking smaller VAE model for training and latent space creation. ”Midime” composed single track music known as monophonic music for a single musical instrument also polyphonic music for numerous instruments. Because it used latent space, the majority of the notes were taken from training data, with some notes being reconstructed. It was found that notes were created from the training data with minor changes based on the patterns.

The performance and accuracy of the LSTM model were considerably improved by researchers working on it. Researchers that used the Variational Autoencoder (VAE) had much better results, although sequential data hampered their effectiveness.Hierarchical decoders were enhanced by sequential data in one case (Roberts, Engel, Raffel, Hawthorne and Eck; 2018). Sub-sequences were initially utilized to for training the data, and each of the sequence is inserted individually to produce music. Smoothing of file identification is required for files used for sequencing and modeling VAE, where the distance between two notes or data points must be quantified. The interpolation approach in latent space was used to accomplish this. To complete the assignment, three methods were employed. On the input sequence, the bidirectional LSTM network described by(Schmidhuber et al.; 1997) was utilized as an encoder method. The employment of a bidirectional recurrent encoder to execute the parametrization of the distribution of latent vector regarding the input series offers optimal long term meaning. RNN Auto-regressive models with the use of "softmax" as an activation function embedded in the output layer were used by the hierarchical decoder to decode these files. To represent these data, architecture with 3 layers was developed, which was influenced by the MusicVAE model; however, where the MusicVAE model used two levels to differentiate instruments, three layers were employed. For these layers, the 'tanh' function was used to initialize RNN. Softmax's function classified various instruments according to their kind. The demonstration was carried out on a variety of sample data, and suggestions for improvement were offered. The quality of many brief sequences recorded with various instruments was assessed. The loss function of the hierarchical model was compared to that of other models with a reduced loss function. The use of a hierarchical decoder on the MusicVAE model enhanced the sequencing in LSTM, according to the findings. The model was tested using four different types of listening tests, including melodies, trios, and drum patterns. These characteristics have variances of 37.85, 76.62, and 44.54, respectively. For all parameters, the Kruskal-Wallis H test of significance yielded a p value as 0.001. This study inspired more investigation into sequential data.

To train the partial music scores, a deep convolutional network called "COCONET" was employed (Huang et al.; 2019). To make polyphonic music, the model demonstrates exchanging notes with other notes. In order to verify the model, NADE sampling and the Gibbs Sampling method were utilized. On the piano rolls, the generative model was created in parallel.A Deep convolutional model was used to train pieces of music using the 'log' gradient loss function. When compared to NADE Sampling, Gibbs Sampling demonstrates higher performance when judged by humans. The dataset utilized for this includes Johann Sebastian Bach's scores, which motivated Google to conduct more study.

The GOOGLE MAGENTA and GOOGLE PAIR teams worked on an AI-powered Doodle named Bach Doodle based on a contribution by Johann Sebastian Bach. [4] J.Bach utilized "COCONET" to train his scores. The notes were harmonized and partial notes were produced as a constructed file using keynotes doodling.GOOGLE PAIR created a browser-based infrastructure based on TensorFlow that allows users to utilize the Doodle in real-time. (Huang, Hawthorne, Roberts, Dinculescu, Wexler, Hong, and Howcroft; 2019) research demonstrates a detailed Bach method. The data was entered using a basic music sheet with staff notes. By raising and enhancing the model's speed, it was optimized. The most challenging part was executing and implementing the model on the client-side in the browser. Working with TensorFlow as a model, the model was hosted in a .js file, and the Tensor Processing Unit was utilized to run it on Google Cloud. WebGL was utilized to improve browser performance, which also helps with model implementation and deployment. The delay of notes and implementation was reduced from 40 seconds to 2 seconds as a result of this. Several exploratory data analysis were performed on the models in response to various requests. This approach was used by millions of people to make music.Musicians actively participated in the composition of music as well as the evaluation of their work. A total of 21.6 million data instances relating to tiny music scores were made public and stored in the collection. For analysis, millions of metadata files were saved. This also contains trained samples applied to music files in order to do

[4] https://www.google.com/doodles/celebrating-johann-sebastian-bach

additional data analysis.

(Hawthorne et al.; 2018) used Neural Network to model and structured the well-known “Maestro” dataset published by Google. Transcribing, creating, and synthesizing audio data for sequential data was demonstrated in this work. The method was dubbed ”Wave2Midi2Wave.” There were 170 hours of piano scores in the collection, all of which were audio files.The goal of the project was to synthesize Midi files and shorten the distance between them. To reduce the distance, the signal processing technique Constant Q-Transform(CQT) was used. For transcription, files were first converted into Midi files frame by frame. The data was subjected to transcription models (Hawthorne et al.; 2017). The number of units in bidirectional LSTM layers was raised from 128 to 256. The transcription model was trained using the audio augmentation approach developed by (McFee and Bello; 2017) utilizing Transformers. The decoder model is used to model the dataset in the partial transformer model component. On the “Maestro” dataset, two models have been trained. This applies to both Midi and transcripted files. In the training dataset, transposing files pertaining to the usage of major and minor data sequences were employed. A piano synthesizer was used to create the trained files, which were then put into the WaveNet model (Oord et al.; 2016). The WaveNet model employed an Autoregressive approach with changes in the units given by each layer. Three models were trained, one of which was a hybrid of the two. This contains a model with Audio files, Midi files, and a mixture of the two. The losses computed after were 3.72, 3.70, and 3.84 for these models, respectively. Listening turing tests were conducted, and answers were recorded, with the goal of distinguishing between recorded files and files played on the piano. When compared to other model produced files, ground truth recorded files received the greatest number of hits. The KWH test of significance with p value as 0.001 was used to calculate variance using 67.63 as a difference between the models. Other polyphonic music datasets were used to enhance the model as a result of the research.

Research conducted by (Wu et al.; 2019) demonstrates the three-layered LSTM models termed Hierarchical Recurrent Neural Network (HRNN) to produce music utilizing its symbolic representation, which was inspired by the use of LSTM models. The model focused on bars, beats, and notes, and these three characteristics were utilized to model the network in three separate LSTM layers.By learning numerous timestamps, melodies, and patterns of music, the model is constructed at a granular level. The model learns notes that are utilized in the symbolic representation of music and are then used by artists to build lengthy musical frameworks. Beat structure and bar theory, which offers a representation of the full bar and beat using notations theory [5], is used to extract features.In the 8th and 16th bar profiles, K-means clusters are used to group the bar and beat characteristics while collecting data on probable future notes. To create sequence and melodies, the architecture is developed with three LSTM layers. Melodies incorporate outputs from the beat and bar layers with three layers to produce varying scales generated in input layer for the output layer using the softmax function. The optimizer utilized in the model was “Adam,” and the learning rate was 0.001. Three distinct HRNN models were implemented: single, double, and triple-layer. The analysis reveals time signatures as well as certain anomalies in the music creation process. When compared to musical scales, the first two demonstrates the recurrence of irregular patterns for rhythms.Listening to produced outputs and voting with hidden clicks for better audio for three HRNN models used on social media were used in the assessment. HRNN with first and third layer were explored to assess the quality of music. Listeners were

[5] https://www.studybass.com/lessons/reading-music/rhythmic-notation/

asked to determine the difference between human and machine composition using chords retrieved from files.Against identify a better performing model,compared to MusicVAE and MidiNet models with H-RNN three layered architecture. Listeners perceived 33.69 percent of generated music to be music made by humans in a music turing test. HRNN, in comparison to MusicVAE and MidiNET, performs better.

Using the Vector Quantized Variational Autoencoder (VQ-VAE) Autoregressive Neural model, the OpenAI project "Jukebox" (Dhariwal et al.; 2020) developed a model that handles on audio music that were classified and classified according to different genres of music.Different kinds of music, such as hip hop, rock, and jazz, were used to create the generated music. The model will use LTS to generate new compositions by giving creative lyrics. Three levels of abstraction were used to process the raw audio. This was motivated by work done by (Razavi et al.; 2019) on the picture dataset using the VQ-VAE model. This model was modified to function as a raw audio data model. Auto-encoders were used to train the layers individually. To manage data for modeling, three sample strategies are used: ancestral, windowed, and primed sampling. Pitch and timing are controlled to build various models for (TTS) Text to speech i.e speech synthesis from audio, which need labeled data. Long sound outputs may be created using lyrics, and these words can be converted into an audio sample of an artist from a specific genre, with the artist's natural vocals seeming natural.

Text analysis using NLP has been done in the past. Google MAGENTA and "GRAY AREA" collaborated on a project named "DearDiary"[6] in August 2020. By entering the words on the terminal or in a notebook, sounds based on emotions were produced. Special letters and punctuation marks were given specific sound notes.Notes that were generated can be replayed and listened as a melody for the keys entered. The VaderNLP and MusicVAE Machine Learning models were used to do this. VADER analyzes keystrokes and assigns a score to each note based on how joyful or sad the music is. The MusicVAE model used this feeling to create the tunes. The created output produced a nice sound with a mood that was written in a diary to help the writer or musician stay focused on their job. The research has the potential to make a breakthrough with mood categorization using music composed as evaluation metrics, while the evaluation for creating music is still the Music Turing test.

Many problems might be answered by algorithms due to neural network research. The application of Momentum LSTM on the well-known MNIST dataset is demonstrated in one of the studies given by (Nguyen et al.; 2020). This will be a unique concept for the research that has been completed. Adding momentum to transformers and decoders demonstrates a novel approach to LSTM network optimization (He et al.; 2020)). Using a different optimizer, the Momentum LSTM is compared against other Recurrent Neural Network models. Momentum is compared to other models in this study in terms of accuracy. To determine the gradient and lowest loss function, different optimization approaches such as Adam and Nesterov are examined. The MNIST dataset, which contains handwritten digits with 10 classes, and the TIMIT Dataset, which contains speech recordings, were both employed in this study. AdamLSTM, Root Mean Square Propagation (RMSprop), and Scheduled Restart LSTM were among the models tested to compare with Momentum LSTM (SRLSTM). When this optimizer was trained, it was revealed that MomentumLSTM exhibits the quickest loss in comparison to other models. The loss is likewise greatest when modeling at the language level and utilizing a text dataset.

Table 1 shows the quick overview of the important related work explained in the

[6] https://deardiary.ai/

section.

Author	Algorithm description	Dataset description
(Sigtia et al.; 2015)	Executing RNN on Hybrid Approach	MAPS Dataset
(Sturm et al.; 2016)	Using LSTM to Transcribe and Compose char-run models	Unknown
(Li et al.; 2015)	CNN's	Tracks
(Yang et al.; 2017)	Deep convolutional generative adversarial network (DCGAN)	MIDI Data
(Dong et al.; 2018)	GAN's (Generative Adversarial Network) framework	MuseGan
(Hawthorne et al.; 2018)	MusicVAE and NSYNTH	Music Scrapping
(Huang et al.; 2018)	Transformer Model	Voice data
(Hawthorne et al.; 2018)	CQT, WAVENET and Auto-Regressive models	MAESTRO Dataset
(Nguyen et al.; 2020)	Momentum RNN (Momentum LSTM)	MNIST and TIMIT

Table 1: Related Work

3 Methodology

In this research as the aspect is more of technical and research focus on more of statistical part and dataset, KDD (Knowledge Discovery in Database) approach is used as a methodology in the research. Importance is given to the type of the data used in the project. In the previous research, most of the models used raw audio files for the research. Computation difficulties have been highlighted in most situations where raw audios files were used to train models for song synthesis, where TPUs were utilized to train audio data for months at a time to achieve accurate results.To overcome this issue of computation, best logical way is to use midi audio files. As the midi files can be executed at note level it is easy to manipulate the notes of midi including tempo and signature stamps. Music with multiple instruments on a single MIDI file can be performed on distinct tracks at the same time, with different or the common instruments on different songs. As a result, MIDI may greatly simplify music learning and production.

It is very important to understand the basic music terminology to play the midi files. For all the instruments this terminology is same and the important features which are present in the terminology is as follow:

- Note: Note is considered as the basic unit of any musical or midi file. Note decides the pitch and the duration of the sound.

- Bar: Bar is considered as the time taken between two consecutive notes which contains beats played with a particular tempo or speed.

- Tempo: Tempo is considered as the speed of the music or sound which is played. Quality and sound is changed in raw audio rather than in Midi file when the speed of the tempo is increased or decreased.

- Octave: This is the gap between two notes that are 11 notes apart and the difference between the pitch and frequency of the upper note has been doubled. Eight octaves are usually seen on a physical piano and on a piano roll.

- Chord: Chord is considered as the sound produced when all the harmonic notes of frequencies and pitches are played together.

- Scale: Scales are created by choosing 7 notes of single octave or the combination of octaves. Notes are arranged in sequence which follows "Circle of Fifths" to produce scales. These are named according to the notes selected and used. For example major, minor etc are types of scales.

- Piano roll: Piano roll is the graphical representation of piano notes in Digital Audio Workstation (DAW), where Midi notes can be edited manually instead of recording the piano or keyboard output as an raw audio file.

The workflow of the model is given in the Figure 2. Workflow shows that Midi files are processed in the Digital Audio Workstation to check the data. To process these Midi files for the execution music21 library is used which is a toolkit for processing Midi files and use for implementation.

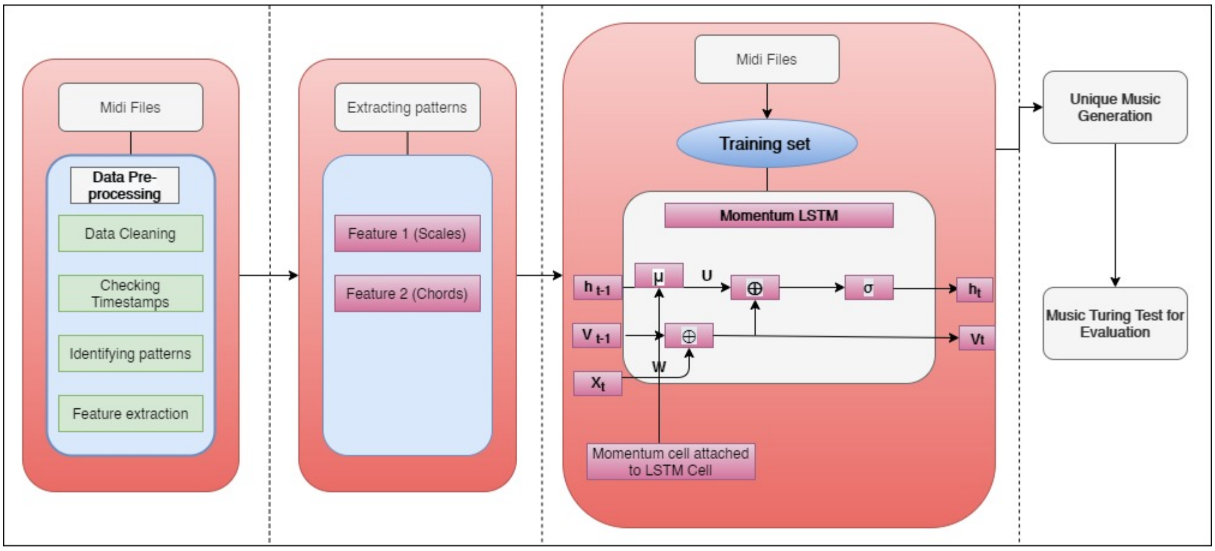


Figure 2: Workflow of a model

3.1 Dataset collection

As discussed the best format to use the data for the research will be Midi format data. This decision has a big impact on the study since it eliminates a lot of the overhead associated with utilizing audio datasets. In the same way other advantage of the using Midi data is that it is not restricted to use Midi file as an an single instrument, this can be played over DAW in any other instrument as well unlike raw audio files. As the

Midi data stores less memory and small in size compared to audio data this also reduces computational issues related to machines.

Dataset is collected from two sources and trained for the model. Initially model was trained to check with the limited data with the first data source i.e data from Feel your sound, later model was trained with large dataset i.e Google Magenta's Maestro dataset.The description of dataset are as follows:

- Feel Your Sound [7]: All 7 musical notes, known as chromatic notes or "Chromatic circle"[8] given in Figure 3, are represented by major, minor, and sharp scales and chords in this dataset. This signals are in the Midi format. There are ten patterns in total for each file, including harmonics and chords starting with scales. Each pattern has 7 different notes and note architecture to finish the loop, which follows the "Circle of Fifths" principle.

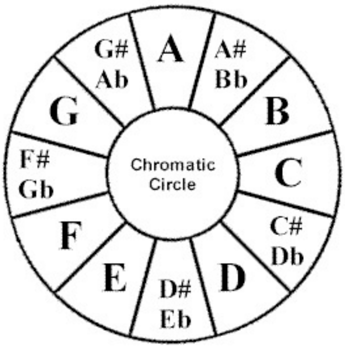


Figure 3: The Chromatic Circle

- Google Magenta (Maestro Dataset) [9]: This dataset is one of the best match for generating research using Midi files. "Maestro" was used, which is an abbreviation for MIDI and Audio Edited for Synchronous Tracks and Organization, and is a set of data of original audio virtuosic piano performances bringing the total 200 hours of audio that are finely synchronised up to 3 milliseconds precision's between their note labels and sound signals in their respective MIDI files. The dataset contains Midi files from 2004 to 2017, in which each folder contains 3000 piano notes in single performance.

3.2 Pre-processing

Processing Midi files is not much complex task unlike Audio and raw data. Midi files were played and checked in to Digital Audio Workstation for the processing. Most Midi files contains 7 notes for scales and harmonics chords. The file is independent of scales and chords which needs separation; this process may be done directly in Fl Studio or using the python package music21 if using Python. Music21 was used to parse and flatten the sheet's notes and timestamps, which were then parsed to a pickle file for easier accessibility. On each of the MIDI files in the collection, the pickle file was added with note information.

[7] https://www.feelyoursound.com/scale-chords/

[8] https://ajwalton.wordpress.com/2010/05/23/chromatic-circle/

[9] https://magenta.tensorflow.org/datasets/maestro

3.3 Extracting patterns

Once the notes are stored in the dataframe. To read all the files file name ”note” is designed where the signatures of the music and weights for the model will be used to generate the file. Generally Audio files does not contain much information unlike text and image data. As the Midi file contains note wise information, this section does not need much work for the processing. After selecting the piano track for Midi files, the timestamp and staff notes were parsed and smoothed using music21, then stored in a pickle file for easier access. On each of the MIDI files in the collection, the pickle file was added with note information.

3.4 Evaluation Method

The evaluation method used in the research is Music Directive Toy Test (MDtT). This test includes human participants. For the research musicians and non-musicians participants are included. Participants will be provided with the music produced by model and the actual original music and will be asked to answer questionnaires according to that. This will be provided in the form of Microsoft Forms. The questionnaire which are included in form are as follows:

- Are you a musician ? : The question itself states that the person is a musician is not. The non-musician category is added to the list so as the result will be non-biased.

- Which of the following option sounds more of human composition ? : After listening both the compositions participants need to answer out of both composition which music sound more of machine generated.

- Rate the composition out of 5 on the basis of melody (How pleasing tune the composition is having ?): Participant need to rate out of 5 for both the music files on how the good it is to hear the files.

- Which of the following have a unpleasant or bad sound ? : Both the files are directly compared and out of both the files, the one with the bad quality need to be selected.

- Which composition have more repetitive sound ? : The question asks out of both the files which sound has more of repetitive music in it.

- Does the music generated sound like machine generated for option 1 and option 2?: Both the sounds are compared and directly asked weather the files sounds like machine generated or not.

- Which of the composition follows good bpm ? : Out of both compositions does the sound follows good timing structure.i.e the beats per minute (bpm). The speed or tempo of the song is counted in beats per minutes.

- Can you identify the scale of the music ? (For musicians): This question is specially to identify which scale the music follows. This question is important from the musicians point of view. This will tell that the generated file follows which chromatic structure.

4 Design Specification

As the model used for the implementation is Deep Learning model, Keras and Tensorflow is used as a python framework. Momentum LSTM will be implemented using this framework. As the model contain midi files. It is important to have machine with the good processing power. To overcome this issue Google Colab is used which is a notebook provided by Google to run python code specially for Machine Learning and Neural Network models. Colab reduces training time by using GPU's and TPU's.

As the working includes Midi files, special package or music tool kit ”music21” is used, which contains useful objects and functions which specially deals with the Midi file. Pre-processing of midi files requires smoothing and flattening of the songs, which is done by music21 library. This library can plot piano rolls, analyze the track, and set the music's speed and time signature.

5 Implementation

The model was implemented by using Python language as Python supports many libraries which are used to process midi files. Tensorflow and Keras are the two most important open software libraries which are used by the python to run machine learning and deep learning models. Momentum LSTM is used as a novel model for implementation. Pickling in the method is used to parse and store the Midi files and use for flattening and smoothness. Two LSTM layers are with 512 units each. After the LSTM layers, there was a batch normalization layer and a 0.3 dropout layer. The input to a fully linked dense layer was given by this dropout layer. The outputs of this dense layer were input into a dropout and dense for normalization, and the activation function 'relu' was used. A ”softmax” activation layer was used to collect the final output. To create the momentum to the model so as to give better accuracy and follow the right direction to increase the performance and accelerate the gradient vectors. Adding momentum is the methods of optimizing the equation or algorithms. This done by importing the stochastic gradient descent (SGD) from keras optimizers. The connection is made between the Gradient descent and Recurrent Neural Network which will improve the RNN's performance.Figure 4 shows the implemented layers to compose the musical files.

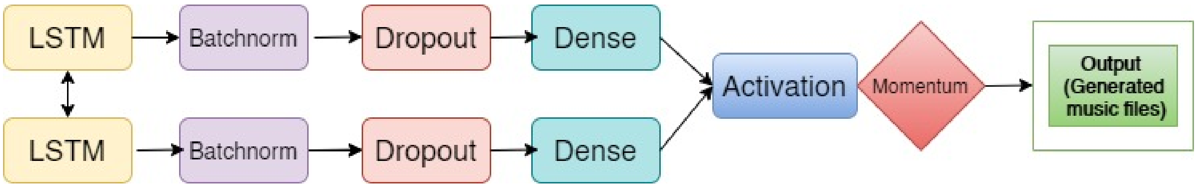


Figure 4: Workflow of a model

Initially the model was trained on the 24 midi files by taking 500 epochs with 0.1 as a learning rate and batch size of 512. As the dataset consist of 24 midi files the time taken to train the model was 1 hour. Initially Google colab was used to train the network with GPU as a session run time. Multiple combinations were tried with selecting the different Midi files. 5 Midi files were generated where the epoch with 500 was showing better generated musical files. The weights and the loss was stored and saved as .hdf5 format. These file was used for the evaluation with human participants.

To check for the long structure files, audio signals from the ”Maestro” dataset was used for the implementation. As these files were long on length and Maestro dataset contains more than 1000 audio files, it was not possible to train on all the files. Due to memory issues and long run time for the models. 50 Midi files from Year 2017 were used for the implementation. The model was implemented locally by using Jupyter Notebook. This was done by on Anaconda platform where it is easy to install packages and libraries. 30 Epochs with same learning rate of 0.1 and dropout of 0.3 with batch size of 1000 to keep the network stable. 500 notes were generated by using the model. It took 3 days to train the model. The file which was generated doesn’t showed much accuracy and there were repetitive sounds and notes.To evaluate the model, Midi from the first model were used and responses were captured.

6 Evaluation

The generated file from the model is in the format of Midi files. This file is visualized on the Digital Audio Workstation. Piano roll is the feature where Midi files can be visualised in the form of notes. Figure 5 shows the Visualized file on the Piano synthesizer and representation of generated Midi file on Digital Audio Workstation.

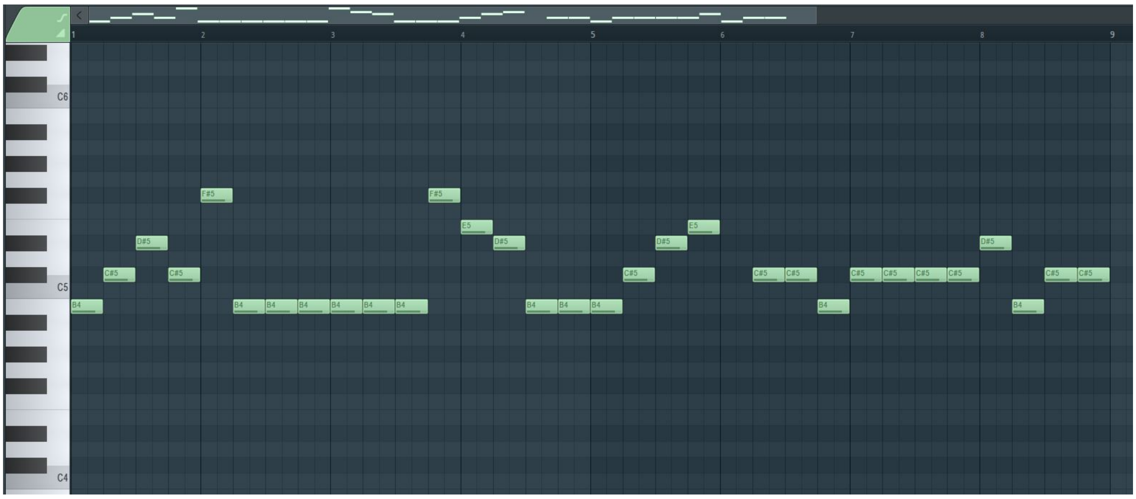


Figure 5: Representation of Midi file on DAW

6.1 Analysis of output

In this section detailed visualizations of the generated music is explained. Key or the scale for the audio is identified by using midi.plot objects from music21 library. Figure 6 shows the graphs for the notes. Maximum number of notes present in the file are B and C notes, which shows the generated file as B major chords.

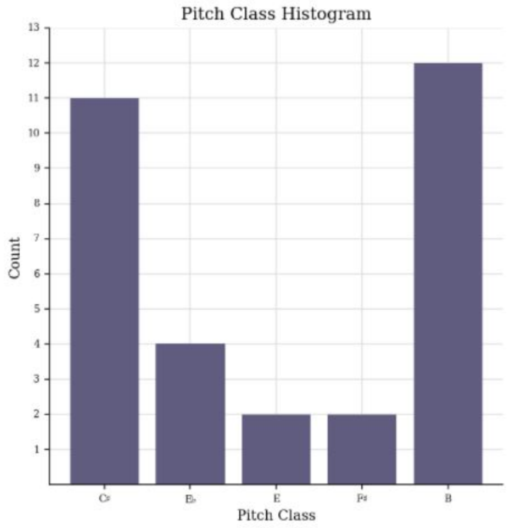
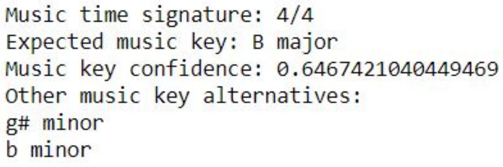


Figure 6: Histogram for notes generated

Key for the song can also be extracted by using the python. Figure 7 shows the original key signature and the alternate key signature for the generated file.



```
Music time signature: 4/4
Expected music key: B major
Music key confidence: 0.6467421040449469
Other music key alternatives:
g# minor
b minor
```

Figure 7: Key for generated music

6.2 Visualizing output in python

In Figure 8 notes from the piano rolls are visualized. Different colours shows notes for different instruments. As in research generated files consists of only single instrument. Notes with the blue dots are plotted on the graph.

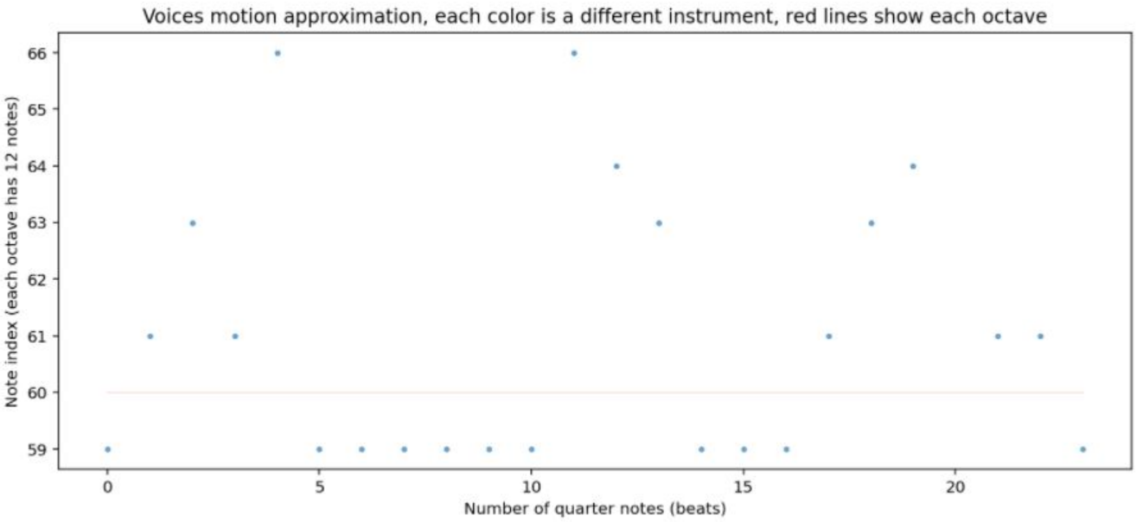


Figure 8: Visualizing notes of Piano roll

Figure 9 shows the plot for notes as a scatter plot. Repetitive notes can be find on the graph.

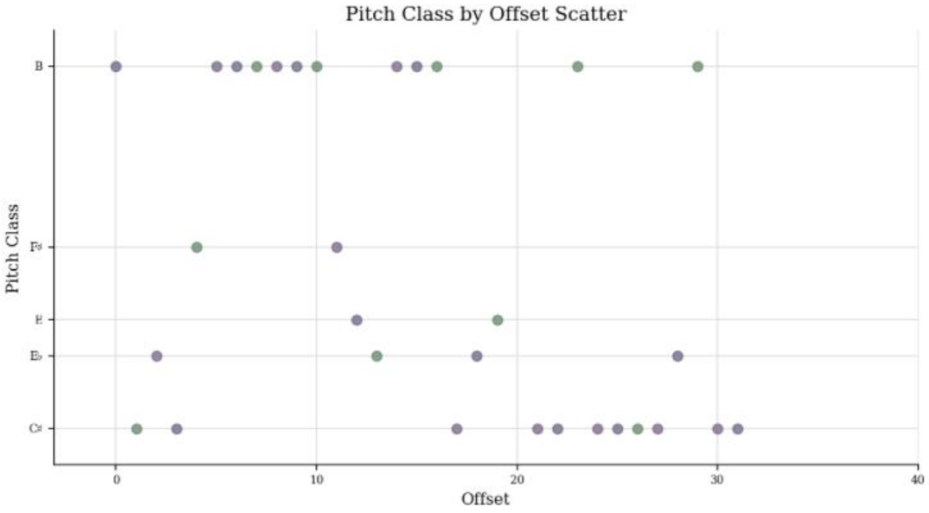


Figure 9: Scatter plot for notes of Piano roll

6.3 Response from Survey

From the analysis from section 6.1, it was found out that the generated file is B major scale. So as to sound similar B major scale Midi from original was used with the machine generated one. "Option 1" was named to the file which is generated by model and "option 2" to the original composition. Both the files were sent for the survey. The results were responded and interpreted by charts and graphs in the Microsoft Forms [10]. There were

[10] https://forms.office.com/Pages/AnalysisPage.aspx?id=wUnbbnK_6k6LP6f9CiW2jK-jRER0vMxLoVOqpsxbUTdUMDNRTzNKOEFNVUpUSzRYNUpYUFlDNkNOWS4u&AnalyzerToken=vC1A4PNzeRYT3zS6V6YzZMVBc2KlRPO4

43 responses recorded from the survey by musicians and non-musicians. Table 2 shows the detailed response from the Survey.

Questions	Option 1	Option 2
Which of the following options sounds more of Human composition ?	60%	40%
Which of the following have a unpleasant or bad sound ?	43%	57%
Which composition have more repetitive sound ?	69%	31%
Does the music generated sound like machine generated ?	57%	43%
Which of the composition follows good bpm (beats per minute) ?	62%	38%

Table 2: Response from survey for Evaluation.

From the Table 2 it can be conclude that 60% of participants selected the generated file from algorithm as a Human composition, which is considered to be a good value. 43% participants considered that file sounds unpleasant or bad. 31% people voted that audio have a repetitive notes. 57% participants voted that the music sounds like Machine generated when separate question were asked. Fot the speed and beat per minute 62% thinks that audio have good bpm. If calculated over all 60% participants think that model worked well and produced audio does sound like human composition.

6.4 Technical challenges

RNN's are considered to be greedy and require large volume of data with high quality. As the data which was used is of high quality but was not enough to train the model.To improve the quality of produced file more number of Midi files need to feed the network. The level of processing power required by RNNs is another big stumbling barrier, and the greater the dataset, the more processing power is required. Many larger and more complicated models might be tried if access to a GPU could be ensured. As the environment used was Google Colab, the session timeout for the this was less as it takes days to train the model. The memory usage for the Google Colab is also less when compared to train the models. Tuning the model's hyperparameters can also provide substantial benefits, but this is a time-consuming procedure that is frequently as much an art as a science. Momentum can also be added to the different Layers and checked for the model improvement.

7 Conclusion and Future Work

The aim behind the research is to implement MomentumLSTM and compose unique music files and to evaluate the model. From the study it can be conclude that the Momentum LSTM's showed improvement in the model. Adding Momentum as an optimizer to LSTM can improve the model better. Momentum optimizer can be used with other different layers and Models and check for the accuracy. To implement and include large files computation power of the machine need to be high as the model can take weeks to train the files. If the model is trained with high computational power and large dataset, there can be good achievement in this field. Same model can be worked with the polyphonic music where multiple instruments are included. Adding Momentum to LSTM showed good accuracy, in the future work these can be done by adding Momentum to other neural network models.Once this model is trained enough, this can be used for creating small intros for the song.

Model creates repetitive notes which doesn't give much significance to composed song. Musicians and non-musician can use the model to generate music if it is trained good.

References

Boulanger-Lewandowski, N., Droppo, J., Seltzer, M. and Yu, D. (2014). Phone sequence modeling with recurrent neural networks, *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 5417–5421.

Briot, J.-P., Hadjeres, G. and Pachet, F.-D. (2017). Deep learning techniques for music generation–a survey, *arXiv preprint arXiv:1709.01620* .

Conklin, D. (2003). Music generation from statistical models, *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, Citeseer, pp. 30–35.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A. and Sutskever, I. (2020). Jukebox: A generative model for music, *arXiv preprint arXiv:2005.00341* .

Dinculescu, M., Engel, J. and Roberts, A. (2019). Midime: Personalizing a musicvae model with user data.

Dong, H.-W., Hsiao, W.-Y., Yang, L.-C. and Yang, Y.-H. (2018). Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment, *Thirty-Second AAAI Conference on Artificial Intelligence*.

Eck, D. and Schmidhuber, J. (2002). Finding temporal structure in music: Blues improvisation with lstm recurrent networks, *Proceedings of the 12th IEEE workshop on neural networks for signal processing*, IEEE, pp. 747–756.

Hawthorne, C., Elsen, E., Song, J., Roberts, A., Simon, I., Raffel, C., Engel, J., Oore, S. and Eck, D. (2017). Onsets and frames: Dual-objective piano transcription, *arXiv preprint arXiv:1710.11153* .

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J. and Eck, D. (2018). Enabling factorized piano music modeling and generation with the maestro dataset, *arXiv preprint arXiv:1810.12247* .

He, K., Fan, H., Wu, Y., Xie, S. and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738.

Huang, C.-Z. A., Cooijmans, T., Roberts, A., Courville, A. and Eck, D. (2019). Counterpoint by convolution, *arXiv preprint arXiv:1903.07227* .

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A. M., Hoffman, M. D. and Eck, D. (2018). An improved relative self-attention mechanism for transformer with application to music generation.

Li, P., Qian, J. and Wang, T. (2015). Automatic instrument recognition in polyphonic music using convolutional neural networks, *arXiv preprint arXiv:1511.05520* .

McFee, B. and Bello, J. P. (2017). Structured training for large-vocabulary chord recognition., *ISMIR*, pp. 188–194.

Moorer, J. A. (1972). Music and computer composition, *Communications of the ACM* **15**(2): 104–113.

Nguyen, T. M., Baraniuk, R. G., Bertozzi, A. L., Osher, S. J. and Wang, B. (2020). Momentumrnn: Integrating momentum into recurrent neural networks, *arXiv preprint arXiv:2006.06919* .

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio, *arXiv preprint arXiv:1609.03499* .

Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K. (2020). This time with feeling: Learning expressive musical performance, *Neural Computing and Applications* **32**(4): 955–967.

Razavi, A., van den Oord, A. and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2, *Advances in neural information processing systems*, pp. 14866–14876.

Roberts, A., Engel, J. H., Oore, S. and Eck, D. (2018). Learning latent representations of music to generate interactive musical palettes., *IUI Workshops*.

Roberts, A., Engel, J., Raffel, C., Hawthorne, C. and Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music, *International conference on machine learning*, PMLR, pp. 4364–4373.

Schmidhuber, J., Hochreiter, S. et al. (1997). Long short-term memory, *Neural Comput* **9**(8): 1735–1780.

Schulze, W. and Van Der Merwe, B. (2011). Music generation with markov models, *IEEE MultiMedia* **18**(03): 78–85.

Sigtia, S., Benetos, E., Boulanger-Lewandowski, N., Weyde, T., Garcez, A. S. d. and Dixon, S. (2015). A hybrid recurrent neural network for music transcription, *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, pp. 2061–2065.

Sturm, B. L., Santos, J. F., Ben-Tal, O. and Korshunova, I. (2016). Music transcription modelling and composition using deep learning, *arXiv preprint arXiv:1604.08723* .

Wu, J., Hu, C., Wang, Y., Hu, X. and Zhu, J. (2019). A hierarchical recurrent neural network for symbolic melody generation, *IEEE transactions on cybernetics* **50**(6): 2749–2757.

Yang, L.-C., Chou, S.-Y. and Yang, Y.-H. (2017). Midinet: A convolutional generative adversarial network for symbolic-domain music generation, *arXiv preprint arXiv:1703.10847* .