Deep Mozart

Elvis Theyo
Indian Institute of Information Technology, Gwalior
Gwalior, India
elvistheyo@gmail.com

Dr. Sunil Kumar
Indian Institute of Information Technology, Gwalior
Gwalior, India
snk@iiitm.ac.in

***Abstract*—Today, Deep Learning (DL) is being utilized to generate various contents such as images, text, etc. In addition to traditional tasks such as prediction and classification, DL is receiving growing attention as an approach to breaking down the barriers between science and the arts. For instance, DALL·E 2, the recent trending AI system that generates realistic images and art from a description in natural language has fascinated the general public and has shown the potential in usability and artistic applications of human interpretable controllable generative models. With the advent of Deep Learning, Is it then possible for a computer to learn to generate music?**

***Index Terms*—Deep Learning, LSTM, Music Generation**

I. Introduction

One of the greatest minds of the 20th century, Albert Einstein, once said, “If I were not a physicist, I would probably be a musician. I often think in music. I live my daydreams in music. I see my life in terms of music.” Music is a fascinating subject that surrounds us constantly, being a source of inspiration and canvas for imagination to many. To some, creating music is a topic worthy of dedicating one’s life to, which is a testament to the artistry and mastery involved. While music composition is an intricate form of art that requires a deep understanding of the human experience, the idea of devising a systematic or algorithmic approach to generate music is a task worth doing to understand how humans can interact with these models and get them to generate a desirable result.

II. Literature Review

Huang & Wu [1] (2016) demonstrated that a multi-layer LSTM, character-level language model applied to two separate data representations is capable of generating music that is at least comparable to sophisticated time series probability density techniques prevalent in the literature.

Keerti *et al*. [2] (2020) proposed a DL based music generation method in order to produce jazz music with rehashed melodic structures utilizing a Bi-directional Long Short Term Memory (Bi-LSTM) Neural Network with Attention.

Chen *et al*. [3] (2020) proposed Music SketchNet, a neural network framework that allows users to specify partial musical ideas guiding automatic music generation. Their approach outperformed the state-of-the-art in terms of both objective and subjective metrics.

III. Methodology

A. MIDI

Musical Instrument Digital Interface or MIDI, pronounced as “mid-ee” is a protocol designed for recording and playing back music on digital synthesizers. Rather than making sound directly, it is a series of messages like “note on,” “note off,” “note/pitch,” “pitchbend,” and many more. Since it does not contain actual audio data, MIDI files are much smaller in size as compared to regular audio files like MP3s or WAVs.

B. Dataset

The dataset used for training the model is a collection of MIDI files containing piano pieces of the prolific and influential composer of the Classical period, *Wolfgang Amadeus Mozart*. The dataset was obtained from Kaggle [4] and contains 21 MIDI files. Each MIDI file contains the notes object type and this contains information about the pitch, octave, and offset of the note.

C. Data Preprocessing

A list of all MIDI files containing Mozart pieces are parsed as a music21 stream containing chords and notes. The notes are then extracted from the data to create a corpus. The corpus is a series of notes which is essentially a list of strings where each string indicates a musical note. The unique notes in the corpus are then mapped to their indices using a dictionary. Furthermore, the corpus is split into smaller sequences of equal lengths of features and the corresponding targets. Each feature and target will contain the mapped index in the dictionary of the unique characters they signify. The features are then resized and normalized while the targets are one-hot encoded.

D. Long Short-Term Memory

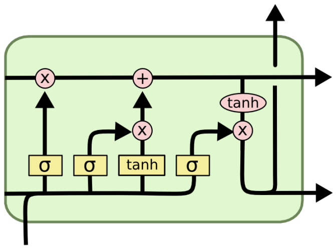

Fig. 1. LSTM Architecture.

Long Short-Term Memory (LSTM) [5] is a special kind of Recurrent Neural Network (RNN) capable of learning long-term dependencies. This network was introduced by Hochreiter & Schmidhuber in 1997.

The core idea of LSTMs is the cell state, the horizontal line running through the top of Figure 2. The cell state resembles a conveyor belt. With only a few minor linear interactions, it runs straight down the entire chain. Information can easily flow along it unmodified.

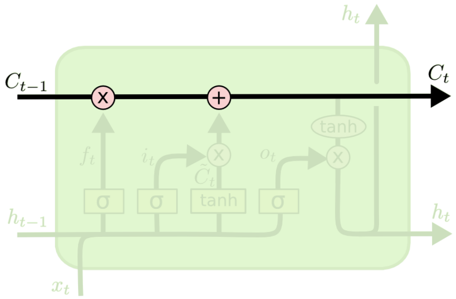

Fig. 2. Cell State in LSTM.

LSTM has the ability to remove or add information to the cell state via structures called gates. They are composed of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between 0 and 1, describing how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”. LSTM has three of these gates: forget gate, input gate, and output gate, to protect and control the cell state.

Firstly, LSTM decides what information to throw away from the cell state. This decision is made by a sigmoid layer called the “forget gate layer.” It looks at h_{t-1} and x_t and outputs a number between 0 and 1 for each number in the cell state C_{t-1}.

Now, LSTM decides what new information to store in the cell state. This has two parts. First, a sigmoid layer called the “input gate layer” decides which values to update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state.

The old cell state, C_{t-1} is now updated into the new cell state C_t. The old state is multiplied by f_t, to decide what information to forget and $i_t * \tilde{C}_t$ is added. These are the new candidate values, scaled by how much we decided to update each state value.

Finally, the output will be based on the cell state. First, the sigmoid layer decides what parts of the cell state to output. Then, the cell state is put through tanh and multiplied by the output of the sigmoid gate, so that only the required information is outputted.

IV. Implementation Details

The model was implemented using TensorFlow and Keras and was trained on Google Colaboratory. The model has the following layers:

- **LSTM layer** takes a sequence as an input and can return either sequences or a matrix.
- **Dropout layer** randomly sets input units to 0 at each step during training time, which helps prevent overfitting.
- **Dense layer** or Fully Connected layer is a fully connected neural network layer where each input node is connected to each output node.

During training, the LSTM network is fed a sequence of encoded musical features and learns to predict the probability distribution of the next feature in the sequence. The output of the LSTM network at each time step is a probability distribution over the possible values of the next musical feature. The network is trained to minimize the difference between the predicted distribution and the actual next feature in the training data using a loss function. The loss was calculated through categorical cross-entropy and optimized using the Adamax algorithm.

In order to find the optimal hyperparameters, hyperparameter tuning was performed and the model was trained a few times to analyze loss and accuracy. The final model was trained on 80% of the corpus with a batch size of 256 and a learning rate of 0.01 for 200 epochs. This model achieved lower loss and higher accuracy compared to other models.

V. Results

The model achieved a training accuracy of 96.72% and a loss of 0.0486 as shown in Figures 3 and 4 respectively. The remaining 20% of the corpus is used as seed data which is provided as input to the trained model to generate new notes.

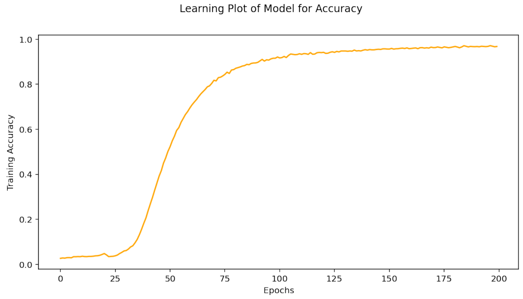

Fig. 3. Learning Plot of Model for Accuracy.

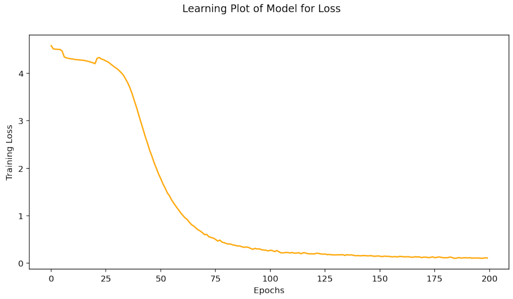

Fig. 4. Learning Plot of Model for Loss.

To generate new music, the LSTM network is initialized with a seed sequence of musical features. Since the seed data has a full list of note sequences, a random index is picked in the list as the starting point which allows the generation code to rerun without changing anything and obtain different results

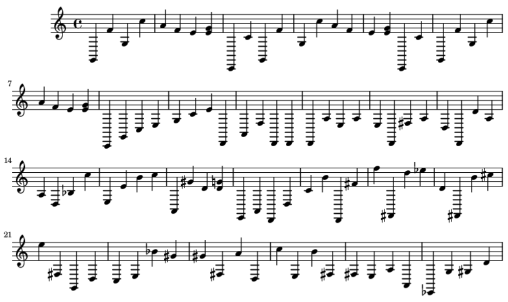

Fig. 5. Sheet music generated by the model using Mozart pieces.

every time. The LSTM network then generates a sequence of new musical features by repeatedly sampling from the probability distribution over the next set of features. The generated sequence can then be decoded back into MIDI format and played as music. Figure 5 contains a sheet music representation of the music that was generated using the model.

When evaluating the quality of the generated music, subjective experience plays a crucial role as there are no proper objective metrics that can quantify the quality of the generated music. In this paper, the quality of the generated music was evaluated based on personal opinion and subjective experience. The evaluation process involved listening to the generated music and assessing its quality based on several factors such as melody, harmony, rhythm, and overall coherence. The subjective evaluation allowed for a more nuanced understanding of the quality of the generated music, taking into account individual preferences and tastes. Although subjective evaluations can be influenced by personal biases, the results of the evaluation showed that the generated music was of good quality, and could potentially be used in a variety of contexts such as background music, soundtracks, or even as original compositions.

VI. Conclusion

This paper has demonstrated that using Deep Learning, a computer can generate music that has both harmony and melody and is considered passable as music composed by humans. While the results may not be perfect, they are pretty impressive nonetheless and shows us that Deep Learning can create music and could potentially be used to help create more complex musical pieces.

This paper's writing comes at an interesting time in the space of DL generated art. Recently, an artist, Jason Allen won the Colorado State Fair's digital arts competition by creating his art, *Théâtre D'opéra Spatial* using Midjourney, an AI program that can turn text descriptions into images [6]. This has sparked a controversial debate about AI's role in art. Although several artists aren't happy with this result, this only further validates how AI will impact our society as it continues to push science and the arts together.

References

[1] A. Huang & R. Wu, "Deep Learning for Music", 2016.

[2] G. Keerti, A N Vaishnavi, P. Mukherjee, A Sree Vidya, G. Sai Sreenithya, D. Nayab, "Attentional networks for music generation", 2020.

[3] Ke Chen, Cheng-i Wang, Taylor Berg-Kirkpatrick, Shlomo Dubnov, "Music SketchNet: Controllable Music Generation via Factorized Representations of Pitch and Rhythm", 2020.

[4] S. Rakshit, Classical Music MIDI, Kaggle, 2019.

[5] S. Hochreiter, J. Schmidhuber, "Long Short Term Memory", 1997.

[6] S. Kuta, "Art Made With Artificial Intelligence Wins at State Fair", Smithsonian Magazine, 2022.

[7] C. Olah, Understanding LSTM Networks, colah's blog, 2015.

[8] M. Phi, Illustrated Guide to LSTM's and GRU's: A step by step explanation, Towards Data Science, 2018.

[9] S. Skuli, How to Generate Music using a LSTM Neural Network in Keras, Towards Data Science, 2017.