Deep learning in automatic music generation

Yizhen Zhang
Fu Foundation School of Engineering and Applied Science, New York City, NY 10027, US.

yz4401@columbia.edu

Abstract. Music is a grouping of musical tones from various frequencies. While artists composed through a deliberate arrangement of different notes, nowadays, A.I. programs learn to automatically generate short music through a machinal sequence of distinct notes. This essay compared the utility and efficiency of traditional machine learning (Regression Model) and deep learning methods (LSTM). This research only focused on instrumental classical music and used the MusicNet collection as the primary dataset. The comprehensive experiments are conducted from these two models, which suggests two results. Firstly, the LSTM model generates melodies that better fit the training styles. Secondly, models are better fitted on single music data than on the entire dataset.

Keywords: Deep learning, Automatic Music Generation, LSTM, Regression Model.

1. Introduction

Though the theoretical foundation for their genres, canonical rules in music composition issue substantial challenges for composers' musical imagination. Inspired by David Cope, a precedent in music AI, I intend to incorporate machine learning into music composition to relieve mechanical labor, such as writing countermelody for the cantus firmus and embracing artistic creativity. Excited by the potentiality of algorithmic composition, I am driven to develop music software embedded with machine learning algorithms that enables collaboration between computer programs and composers, the interaction design I envision as the future of computer-generated music [1].

Traditional machine learning methods like regression models could complete such composition procedures. Meanwhile, as the deep learning technique developed in the 21st century, many scientists used the algorithmic capacity of deep learning architectures to train programs to learn musical styles from a massive number of classical compositions [2]. In my research, I investigated the utility of traditional machine learning and deep learning in automatic music generation. Both have their advantages and drawbacks, which I will discuss in detail in the conclusion paragraphs.

2. Literature Review

A new research topic in the 21st century, the field of automatic music generations, has its state-of-the-art methods distributed in four categorizations: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Variational Autoencoders (VAEs), and the Regression Model. First, in the category of the RNN model, researcher Skuli experimented with LSTMS to generate single-instruments music in 2017 [3]. Following up on this scholarship, scientist Nelson, in 2020, used LSTM to compose lo-fi music, a music quality that treated elements as imperfections in the context of a recording. Second, for CNN Model, Researcher Yang (2017) created MidiNet, which generates multi-instrument music sequences [4]. In the same year, Scientist Dong (2017) used MuseGAN, which utilizes multiple generators to achieve synthetic multi-instrument music that respects dependencies between instruments [5]. Then, researcher Tham (2021) introduced a new VAE-based architecture to generate novel musical samples [6]. Finally, musician Chen employed Random Forest Regression Model for accessible music generation in 2020 [7].

3. Methods

3.1. Introduction of dataset

First, the dataset for all model training is only constituted of instrumental classical music from the Baroque Period, Classical Period, or Romantic Period. For more details, I used the MusicNet dataset from Kaggle, posted by Dr. Sara Gomes [8]. MusicNet dataset consists of 330 freely licensed classical music recordings from over 1 million annotated labels. Those labels recorded the precise time of each note in every recording, the instrument that plays each note, and the note's position in the metrical structure of the composition. Moreover, there are three sub-datasets in the MusicNet dataset: the data with the general information of the databases (metadata, Table 1), the test data with information per music (test data, Table 2), and the train data with information per music (train data, Table 3). Such divisions facilitate the data pre-processing process and help us classify datasets for model training.

Table 1. Meta Data.

	id	composer	composition	movement	ensemble	source	transcriber	catalog_name	seconds
0	1727	Schubert	Piano Quintet in A major	2. Andante	Piano Quintet	European Archive	http://tirolmusic.blogspot.com/	OP114	447
1	1728	Schubert	Piano Quintet in A major	3. Scherzo: Presto	Piano Quintet	European Archive	http://tirolmusic.blogspot.com/	OP114	251
2	1729	Schubert	Piano Quintet in A major	4. Andantino - Allegretto	Piano Quintet	European Archive	http://tirolmusic.blogspot.com/	OP114	444
3	1730	Schubert	Piano Quintet in A major	5. Allegro giusto	Piano Quintet	European Archive	http://tirolmusic.blogspot.com/	OP114	368
4	1733	Schubert	Piano Sonata in A major	2. Andantino	Solo Piano	Museopen	Segundo G. Yogore	D959	546

Table 2. Test Data.

	start_time	end_time	instrument	note	start_beat	end_beat	note_value	id
0	9694	11742	41	61	4.875	0.108333	Thirty Second	2106
1	11742	34270	41	62	5.000	0.975000	Quarter	2106
2	34270	53725	42	54	6.000	0.975000	Quarter	2106
3	34270	53725	43	50	6.000	0.975000	Quarter	2106
4	34270	53725	41	57	6.000	0.975000	Quarter	2106

Table 3. Train Data.

	start_time	end_time	instrument	note	start_beat	end_beat	note_value	id
0	113630	208862	1	44	0.00	0.989583	Quarter	2575
1	113630	140254	1	56	0.00	0.239583	Sixteenth	2575
2	113630	208862	1	60	0.00	0.989583	Quarter	2575
3	140766	168414	1	51	0.25	0.239583	Sixteenth	2575
4	168926	187870	1	56	0.50	0.239583	Sixteenth	2575

3.2. Data Analysis

Second, to better understand the data, I investigated Dr. Gomes's categorization on MusicNet. I focused on the ensemble and composer of each composition work. As a result, in the ensemble analysis (Figure 1), the top three popular ensemble types are: solo piano (47.3%), string quartet (17.3%), and accompanied violin (6.7%). In the composer analysis (Figure 2), the top three famous composers are: Beethoven (47.6%), Bach composed (20%), and Schubert composed (9%). Based on the analysis, I will select works composed by these top three composers in the type of solo piano and accompanied violin. Since string quartet composition is not solo music, this type might raise confusion in our final model training. Therefore, the string quartet compositions are removed from our dataset.

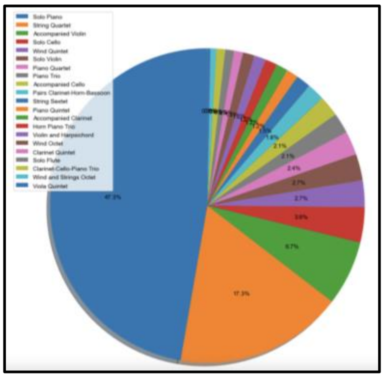

Figure 1. Ensemble Analysis.

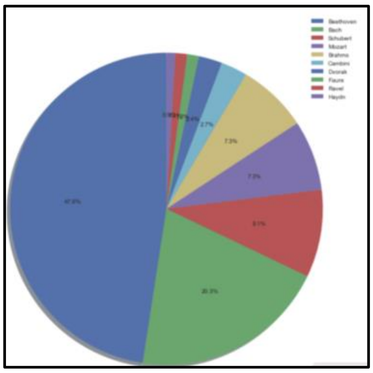

Figure 2. Composer Analysis.

3.3. Model Architecture Introduction

Then, I investigated two model architectures: the LSTM (Long Short-term Memory) Model from deep learning implemented by Scientist Aravind Pai and the Regression Model by Dr. Sara Gomes. Created by Hochreiter and Schmidhuber in 1997, LSTM Model is a variant of RNN that captures the long-term dependences of the input sequences. At each time step, the Long Short Term Memory cell (LSTM cell) receives an input of an amplitude value. Next, the LSTM cell calculates the undercover vector, which it then passes onto the subsequent timesteps (Figure 3). In such a way, the LSTM model can hold on to connecting the information as the gap between relevant information grows.

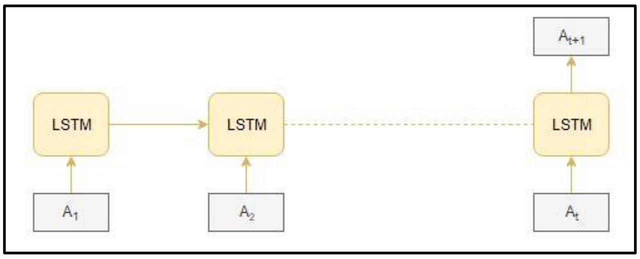

Figure 3. LSTM Model.

Then, for the implementation of the Regression Model, Dr. Sara Gomes selected Decision Tree Regressor and Linear Regressor. A decision tree implements regression and classification architectures organized in the form of tree structures. Such a model breaks down a database into smaller subsets while gradually constructing a linked decision tree [9]. Next, a more basic linear regression model fits a linear function or surface that minimizes the differences between the actual outputs and the predictions. I aim to compare the efficiency and utility of deep learning and traditional machine learning in automatic music generation with these two models.

4. Implementations

4.1. Data Preprocessing

The classical music dataset was preprocessed separately to better train the model. First, for the LSTM model, Pai prepared new musical files that only have frequent top notes, which took a minimalist approach to refine the dataset. Later, to prepare the input-output series for the LSTM method, Pai sequentialized music notes with a unique integer and prepared the "integer-sequences" for input and output data. Second, for Regression Model, Dr. Gomes worked with the whole dataset and selected "the composer," "duration," "the note," "start and end time," and "start and end beat" as relevant features in predictions. Finally, based on the results of previous data analysis, Dr. Gomes prepared the data to perform separate predictions on solo piano. They accompanied violin, the two most popular ensemble genres from the dataset.

4.2. Model Training

For the LSTM Model, Scientist Pai trained the model using a cluster of 128 for 50 epochs. The currently hidden vector is computed based on the current input and formerly hidden vector ht-1 at timestep ht. This way, the LSTM Model grabs the sequential information presented in the input sequence. After successful model training, Pai converted the predictions into MIDI files using the music21 library designed for digital music analysis by a team from MIT. Finally, with the assistance of the music21 library, Pai created note and chord objects founded on the LSTM model's predictions.

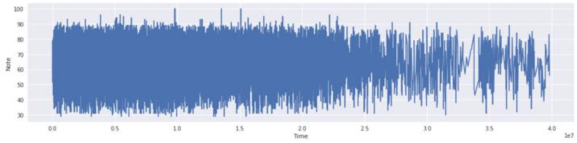

Figure 4. Solo Piano.

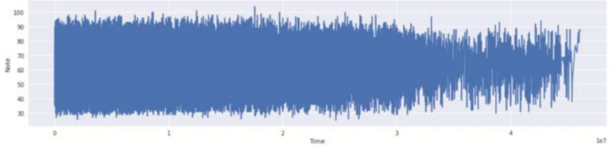

Figure 5. Violin.

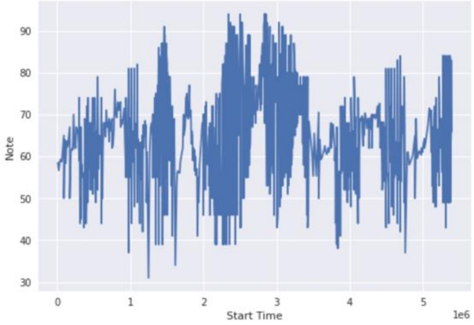

Figure 6. #1759 Music.

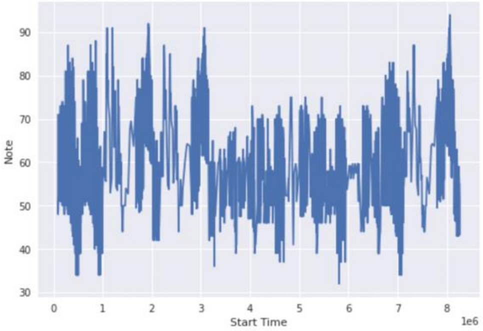

Figure 7. #2628 Music.

Next, for the regression model, Dr. Sara Gomes implemented the regression model first on the entire dataset and then on two random individual instances (#1759 music and #2628 music) to predict music by solo piano and violin, respectively. In both circumstances, the decision tree models tend to give higher accuracy in their training results. Figures 4 (solo piano) and 5 (violin) show the predicted notes versus start time for the entire data testing set. Figures 6 (#1759 music) and 7 (#2628 music) show the results run on individual instances.

4.3. Model Evaluation

Dr. Gomes used R-Squared Metrics to evaluate the model accuracy. *R-squared* is a statistical standard representing the goodness of fit for regression models. The closer the r-square value to 1, the ideal value, the fitter the regression model is [10]. In Dr. Gomes's research, as R-Squared Metrics (Table 4) suggested, both regressors perform poorly on the entire dataset, with R2 values ranging from around 0.01 to 0.22. However, both regressors perform well on individual cases with an average R2 of around 0.52 to 0.80. For LSTM Model, I did not use evaluation metrics to scale the model since the predictions gave auditory music results, and it is hard to standardize sensual enjoyment numerically. However, in my view, the model performs well in generating solo piano music, consisting of well-reasoned melodies despite its simplistic structures.

Table 4. R-Squared Metrics.

	Model Evaluati on: Solo Piano	Model Evaluatio n: Accompanied Vi olin	Prediction Evaluat ion: Solo Piano (#175 9)	Prediction Evaluation: Accompanied Violin (#262 8)
Tree Regression	0.217872	0.158875	0.791103	0.525985
Linear Regression	0.010397	0.021409	/	/

5. Conclusion

In short, the regression model is easier to evaluate than the LSTM Model. More specifically, for the regression model, Tree Regression presents better fits than Linear Regression, and Models were fitted to single music and showed promising results. However, the Regression model is less efficient in translating mathematical prediction into auditory results (the actual music) than the LTSM Model. To improve both models, future researchers can either increase the size of the dataset or refine the model itself. An increment in the training data size raises the efficiency of deep learning models while refining the pre-trained model helps to build a more robust system that gives more comprehensive predictions.

References

[1] Zachary. (November, 2020). “How I Built a Lo-Fi Hip-Hop Music Generator.” Medium. Artificial Intelligence in Plain English. https://ai.plainenglish.io/building-a-lo-fi-hip-hop-generator-e24a005d0144.

[2] Pai, Aravindpai. (January, 2021). “Automatic Music Generation: Music Generation Deep Learning.” Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/01/how-to-perform-automatic-music-generation/.

[3] Skúli, Sigurður. (December, 2017). “How to Generate Music Using a LSTM Neural Network in Keras.” Medium. Towards Data Science.

[4] Yang, Li-Chia, Szu-Yu Chou, and Yi-Hsuan Yang. (July, 2017). “Midinet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation.” arXiv.org.

[5] Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. (November, 2021). “Musegan: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment.” arXiv.org. https://arxiv.org/abs/1709.06298.

[6] Tham, Isaac. (August, 2021). “Generating Music Using Deep Learning.” Medium. Towards Data Science. https://towardsdatascience.com/generating-music-using-deep-learningcb5843a9d55e.

[7] Chen, Vivian, Jackson DeVico, Arianna Reischer, Leo Stepanewk, Ananya Vasireddy, Nicholas Zhang, and Sabar Dasgupta. (2020). “Random Forest Regression of Markov Chains for Accessible Music Generation.” 2020 IEEE MIT Undergraduate Research Technology Conference (URTC).

[8] Gomes, Sara. (November, 2021). “Music Generation Based on Classics .” Kaggle. Kaggle. https://www.kaggle.com/code/smogomes/music-generation-based-on-classics.

[9] T, Smitha., and V. Sundaram. (2012). “Classification Rules by Decision Tree for Disease Prediction.” International Journal of Computer Applications 43, no. 8: 6–12. https://doi.org/10.5120/6121-8323.

[10] Rácz, Bajusz, and Héberger. (August, 2019). “Multi-Level Comparison of Machine Learning Classifiers and Their Performance Metrics.” Molecules 24, no. 15 (2019): 2811. https://doi.org/10.3390/molecules24152811.