Analysis Of the Principle and Application of Machine Learning for Music Composition

Yingkai Sun

Beijing International Bilingual Academy, Beijing, China

* Corresponding Author Email: 2024asun@biba-student.org

Abstract. Contemporarily, machine learning and artificial intelligence are undergoing rapid developments in the field of arts. This paper explores the principles and applications of machine learning in music composition, tracing back to its inception in 1950s, and taking a brief look at the first works in the field of computer-generated music. It delves into the key principles of music composition using machine learning, and discusses the theory behind major models of recurrent neural networks (RNNs) and Generative Adversarial Networks (GANs) and how they are utilized in music composition. This paper also discusses about the real-world applications and potentials of current models, and finally analyze key limitations present in this field and points out a few directions for future works. This paper gives a brief and simple overview of the current status of machine learning in music composition, and provides an assistance to individuals interested in exploring this field and a valuable insight for future innovations.

Keywords: Machine Learning, Music Composition, computer-generated music, recurrent neural networks (RNNs), Generative Adversarial Networks (GANs).

1. Introduction

In recent years, artificial intelligence has been primarily focused on replacing human creativity and imagination, which was once thought impossible. Though in just a few years, computer generated visual arts have experienced an evolutionary growth with the emergence of machine-learning models such as Stable Diffusion or Midjourney. In tandem with these visual breakthroughs, AI has been quietly revolutionizing another form of artistic expression: music composition. Though the quest for computer generated music began around half a century ago. The first piece of music generated from an electronic computer is generally credited to Lejaren Hiller and Leonard Isaacson's "The Illiac Suite", a four-movement string quartet created in 1957 [1]. The main composition technique that Hiller and Isaacson used was stochastic algorithms, which was solely based on the principles of probability. Elements of music, including rhythm, pitch, interval, dynamics, and articulation are all stochastically generate based on pre-set parameters. Hiller and Isaacson could tweak the parameters and influence the result but could not determine the final product.

After the groundbreaking achievement of "The Illiac Suite" in 1957, computer-generated music began to develop rapidly. One of the significant developments that followed was the adoption of more sophisticated algorithms in the field of computer-generated music. These algorithms aimed to not only generate music out of randomness, but also to introduce elements of creativity and to generate acceptable and conventional music that mimics human compositions, such as Markov models, Cellular Automata, or Generative Grammar. Up until the 1980s, the main approach to computer generated music was using the method of algorithmic composition that is based upon probabilities and rules. At around 1980, David Cope introduced his groundbreaking computer work "Experiments in Musical Intelligence" (EMI) [2]. Many would agree that this was the pioneer in utilizing machine learning in musical composition. David Cope initially approached this experiment by setting the rules manually for a rule-based system, though he quickly found that the output generated was dull and uninteresting. He realized that having a human being in the middle, setting rules and limitations, was unnecessary [3].

This led him to create an algorithm that can create compositions based on a database of music of famous composers. It analyzes patterns, elements, and other musical components, and recombine

them into a new creation. David himself called it "recombination", however, it resembles the basic logic of machine learning and stands as one of the first instances of utilizing neural network in music generation. Then, from 1980s to early 21st century, people started to experiment by using neural networks to generate music [4]. Right now, machine learning has become the most widely used, efficient, and successful method used in music generation.

In recent years, the field of computer-generated music have experienced rapid growth with the use of machine learning. Projects like jukebox [5], DeepBach [6], and many other projects all exhibits promising results, producing quality compositions using machine learning. Deepbach is a generative model that is capable of creating music that imitates the style of the composer Johann Sebastian Bach. It can generate convincing baroque style, four-voiced choruses, with coherent yet original harmonization of melody lines. Bach's music is known to be extremely intricate and complex; to tackle this problem, Deepbach adopts a method of pseudo-Gibbs sampling, which allows Deepbach to be trained with specific, categorized, and labeled samples. For example, two of the four voices may be separated and trained, or it could only be programmed to predict a single voice. In the end, Deepbach exhibited promising results. An experiment was conducted and around half of the 1272 subjects, most with music experience or background, though that the generated music from Deepbach is composed by Bach. Jukebox, introduced in 2020, is also one of the latest projects in the field of AI music. It is a project powered by OpenAI and can create music of different styles such as jazz, rock, and classical; notably, it is also able to generate lyrics for specific artists or genre. Jukebox is built upon VQ-VAE. In essence, the model first encodes the data, representing them using discrete codes. This breaks the data into "building blocks"; these could be elements of music such as pitch, rhythm and so on, Then, it generates an output by rearranging these codes to create a composition [7]. With that, jukebox now can generate minute long pieces in the style of many different artists, along with the ability to compressing and representing music with minimized data loss. Right now, the field of computer-generated music is experiencing rapid advancements. AI and other machine learning models are now capable of generating long, coherent music in various styles, creating endless possibilities and injecting creativity into the music industry.

With the rapid development of the use of machine learning in mimicking human creativity, the field of music is not to be overlooked, though music very well may be the hardest of the arts to tackle. However, through more than 80 years of exploring in the field of music generation, people have already great success in music generation from the use of machine learning. This paper takes a detailed look at the application of machine learning in music composition by covering the basic theories of machine learning, principles for music composing, common models used in machine learning, the practical application of machine learning, and an analysis of limitations and outlook of using machine learning in generating music.

2. Basic Description of Machine Learning

As a branch of artificial intelligent, machine learning has found its way as the most influential field in modern technology. It is being applied in so many fields and aspects such as search recommendations, ad placements, and weather predictions [8]. In essence, machine learning enables computers to learn, adapt, and make decisions from existing data without explicit instructions and programs given by human [9]. Machine learning is usually categorized into three types: supervised learning, unsupervised learning, and reinforced learning. The most frequently used type of learning in generating music is supervised learning. In supervised learning, the input data that the network receives are all labeled. A machine is provided a training set of data, with the inputs and outputs correctly identified, and is trained to classify data and predict outputs [10]. The most widely used and developed model are the neural network models, whether in the music industry or any places. Neural networks are constructed from layers of neurons. Each neuron is connected from all the neurons from the preceding layer via nodes, each of which carries a specific weight. A neuron's value is calculated by summing the products of these weights and the values from the connected neuron on the previous

layer. A bias, a constant term, is also added to introduces flexibility and offsets the result. Mathematically, the value of each neuron is denoted by:

$$value = b + \Sigma_i x_i w_i \quad (1)$$

where b is the bias, x_i is the value of each neuron connected, and w_i is the corresponding weight. Since neural network is generally supervised, meaning that both the inputs and the outputs are labeled, one could calculate how far the output from the network is to the target value, or the error of the training sample, and the function for that is called the cost function. The main objective of training this network lies at minimizing the cost function. This objective drives the iterative process of adjusting the network's learnable parameters, which is the weights and biases, after each sample trained. By fine-tuning each parameter, the network can enhance its accuracy of prediction as error decreases. In the music industry, machine learning has a huge advantage for its ability of pattern recognition. Neural networks can analyze vast amount of data that includes patterns of melody, harmonies, rhythms and so on. This allows the generated music to adhere to the conventional music, creating desired outcomes.

3. Principle for Music Composing

Music composition is a complicated process, where differences in composer, style, and time influences the final product by a massive scale. There are several musical elements that are constant through any style of music. In music theory, the four main elements of music are melody, harmony, rhythm, and timbre. To make a computer generate something with all the elements in a coherent, balanced, and conventional manner is a challenging task [11].

The exploration for generating a monophonic melody line began just as people started to explore the possibilities of technology in music generation. The initial idea of approach, as mentioned earlier, was a stochastic approach, which incorporates structured randomness in generating melody [12]. Even though in recent years, melody generation is all based on machine learning, it still resembles a similar idea, which is exploring motifs and patterns to create a coherent and well-structured melody line [13]. The generation for harmonies is somewhat the same. The initial approach was rule-based models, where the models' generations are restrained from predefined rules that follows the music theory [14]. However, machine learning approaches, including neural networks, have become prevalent in harmony generation as well. These models can analyze existing harmonic progressions, analyze, and identify rules and patterns from harmonic progressions, and creates harmony to complement a given melody [15]. The principles for rhythm generation is more focused on patterns, since rhythms often exists intertwined and inherently with the melody [16]. Still, machine learning models, including recurrent neural networks and Markov models, have been used to capture rhythmic structures from existing music and apply them to generate new rhythms. Lastly, the timbre of a music is closely related to the specific genre of the music, such as specific instruments for certain styles. Therefore, it is mostly pre-decided based on the style chosen. The most challenging part, however, would be to combine these elements into a coherent piece of music. The approach by Deepbach, a model able to generate four-voiced Bach fugues, separates each of the four voices while training, allows it to generate singular voices on top of three existing voices as well as a complete four-voiced composition [6]. Even with other approaches, nearly all computer-based music-generation models are built upon machine learning models.

4. Models

There are three main models that are being used in music compositions today. Three separate papers will be analyzed and used to represent a model used. The first model that is commonly used is Recurrent neural networks (RNNs). RNN is a neural network that specializes in dealing with sequential data, such as notes in a melody line. Unlike ordinary feed-forward neural networks, RNNs have an extra loop of neural network that allows data from previous inputs to be stored as "memory".

Hence, one could obviously see why RNNs are so commonly used in music generations, especially in melody generation. As melody is essentially ordering notes in a sequence, RNNs would be suitable as it allows past notes in a melody to be considered and used to generate output. However, if simply using the traditional RNN structure, the user may run into problems of vanishing and exploding gradient [17]. This means that the gradient, or rate of change of cost function, either changes too slowly during back-propagation, meaning some parts of the network learns too slowly or not at all, or the gradient changes too fast that the network fails to converge. To prevent vanishing and exploding gradient, an LSTM, long short-term memory, model is often implemented instead of a simple hidden state in traditional RNNs. An LSTM model, different from traditional RNNs, contains an input gate, forget gate, memory cell, and lastly an output gate. In an LSTM model, the memory cell can store information and retain it for a long time, while traditional RNNs can only store short term information (seen from Fig. 1) [18].

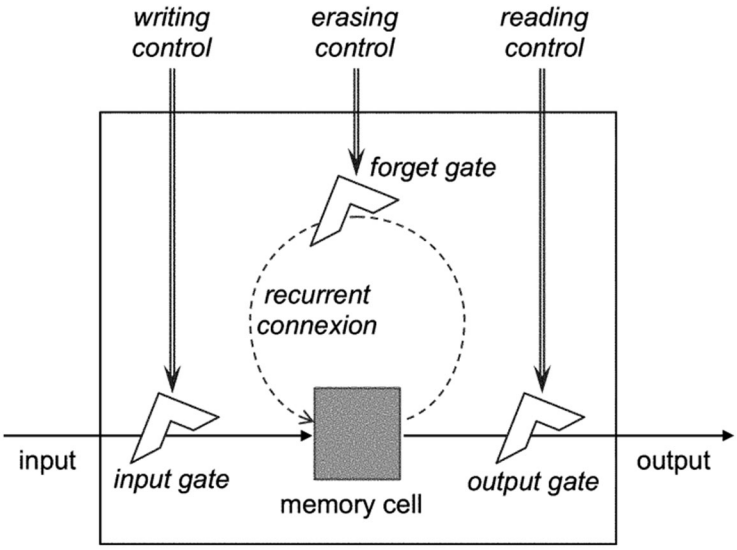

Figure 1. A simplified LSTM model diagram [18].

Each of the three gates, input, forget, and output, are assigned a value inside the range of 0 and 1 using the sigmoid activation function, where 0 means no information will be passed through the gates and 1 means all the information will be passed through the gates. The input gates controls what information is traveled to the memory cell, or what will be stored. The ability to selective store or block information allows the magnitude of the gradient to be controlled during back-propagation. It could stop excess information from being stored in memory, preventing the input from causing gradient explosions, but also ensuring that the gradient don't get too small to prevent vanishing gradient. The forget gate is the gate that decides whether to discard or keep the information stored previously. The forget gate is extremely important, because if long-term information is stored forever in the memory cell, it could cause the whole system to break down. Lastly, the output gate controls which information is to be outputted from the LSTM, and also decides the information of the next hidden state. Together, these three gates work together to resolve the vanishing and exploding gradient problem as it offers a way to adjust and control the information traveling though the unit as well as the gradient through back-propagation.

Another common form of model used in computer generated music is Generative Adversarial Network (GAN) [19]. GAN is a network that is capable of training two competitive models simultaneously. These two models in a typical GAN are called generator and discriminator [20]. The discriminator is a neural network that is trained and optimized to be able to identify real samples sent to it. Then, the role of the other model called generator will produce synthetic, or 'fake' samples. These samples are then sent to the discriminator. The discriminator's job is to identify whether the sample received from the generator is a fake sample or not. After that, the fact that the sample is synthetic is revealed to both models (seen from Fig. 2).

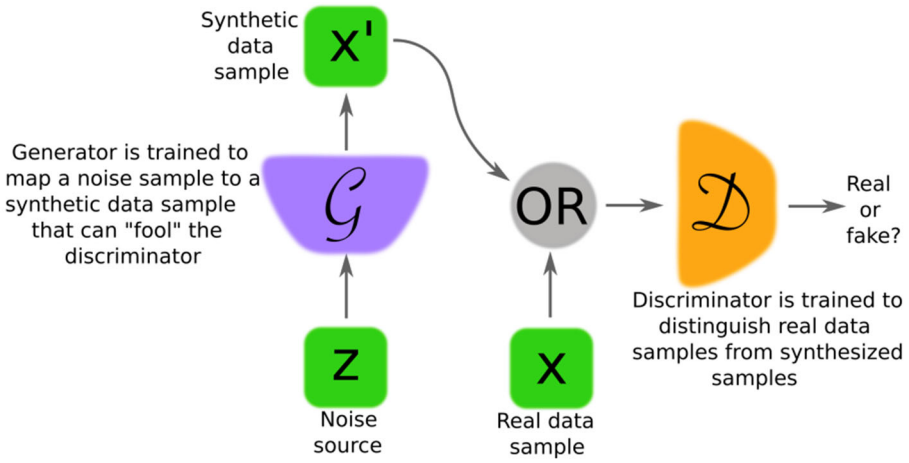

Figure 2. A figure of the Generator and Discriminator in a Generative Adversarial Network in Creswell [20].

If the discriminator fails to identify that the sample is a generated sample, then the discriminator will be updated. However, if the discriminator successfully identifies that the received sample is not a sample from real-life, meaning it could identify that the sample is generated from the generator, then the generator will have to be optimized. This process will be iterated many times until the generate is capable of generating near-perfect samples that will fool the discriminator.

5. Applications

Though experiencing rapid development in computer-generated music, actual real-world application for these models is somewhat limited, or in other words, premature. The main focus of research in this field is still focused on creating a model rather than finding and implementing models as an active and available application for users. Many researchers have expressed their desired aimed for these models to be a useful creation for both musicians and non-musicians [5, 6, 21], but even with the success of most of these models, rarely any are made easily accessible to the common population or friendly for users [22]. Nevertheless, there is still a few examples where computer-generated music has been put into applications. One such example is the Google's Magenta Project. The Googles Magenta Project is an open-source project created by Google and introduced in 2016. The magenta project revolves around the use of machine learning in imitating human creativity such as music and art. It is developed based on Google's TensorFlow framework and aims to explore the use of machine learning in music and art creation. The main applications for such technology lie in two categories: aiding musicians in music creating and for replacing the need for humans in music composition. First, machine learning models could aid musicians in creating music. It is used as a tool to expand the creativity of musicians, allowing human-machine co-creation in generating of musical production. Some musicians also express that machine learning models doesn't necessarily mean to create the best compositions and replace human creativity, but rather as a tool to speed up the process of creating music [22], eliminating extra efforts needed to refine compositions. The other main use for these models is for commercial use. For example, short commercials that requires a short, style-specific music could reduce its cost of hiring a musician or purchasing copyrights by using machine learning based models that are capable of generating specific style, mood, and genre of music based on user requirements. This, in the future, could also stem across movie, games and other art forms. Though there are already cases of this happening, there is still a huge room for exploration. Lastly, another major use for generative musical models can be used to encourage non-musicians to walk into the world of music creations. With simpler interactive ways and tools that could create professional sounding music from simple inputs, music-enthusiasts without formal musical training

could also wield the ability to express messages and emotions through music. This also stands as the main goal for many different researchers in this field.

6. Limitations and Future Outlooks

Music generation through machine learning has indeed shown its potential by producing captivating melody lines and well-harmonized voices, though a few problems still pose against the current landscape of music generation. A major challenge that machines learning faces in music generation is the ability to sustain the quality of the composition through a longer timescale. While machine learning exceeds at generating short passages, it fails to produce a long piece while having a well-structured long-term relationship within the piece, lacking a structure that spans across the whole piece.

A common limitation is the massive amount of training time. A minute long product could take around an hour of training [5]. Longer and more sophisticated music, like Bach fugues, could take more than a day of continuous training [11]. these intricate compositions often include complex musical elements that requires an extensive time to learn and refine for algorithms, sometimes still fails to capture the essence, details, and nuances in a music.

Another major limitation of machine learning is overfitting, where a machine learning model learns too well to the point that all the noises and idiosyncrasies of the training data are learned. Overfitting poses many issues. One, the unwanted imperfections and noises in training data will be learned and produced in the product. Furthermore, it could lead to a lack of creativity and originality as it focuses too much on capturing every element in the training data, as well as repeating cliche patterns and melody. This also leads to a lack of coherence inside the music as sequences of notes, melodies, and harmonies may not fit well together because the model could be overly fixed on mimicking an individual training data.

As a growing field, machine learning in music generation is developing rapidly. However, there is still a lot to be improved. The focus of future works lies at creating better long-term music compositions. Currently, researchers have not been hugely successful with creating a well-structured and coherent piece that don't lose its quality over long periods of time, though it's likely that, with this pace of development, this problem will be solved in the near future. Researchers should also aim at making these models user friendly and putting them into real-world uses. As mentioned above, there is still a huge potential for these technologies in the world today, whether it is towards aiding musicians in their creative processes, non-musicians in lowering the threshold, or to create appropriate music for commercial uses.

7. Conclusion

To sum up, machine learning is undergoing an unprecedented pace of development. Beginning in the 1950s, computer generated music has come from using random note generation to machine learning. Right now, the introduction of models such as RNN and GAN stands as a foundation for the current development of music generation. This also creates great possibilities for real-world applications of computer music generation in the future. However, there is still work remains to be done in the future, such as focusing on generating long and well-structured music or lowing the accessibility of such models. Overall, despite the developments in computer-generated music composition, there is still great potential and innovation to be discovered in the future.

References

[1] L. A. Hiller and L. M. Isaacson, L.M, *Musical composition with a high-speed digital computer*. (The MIT Press, Cambridge, 1997) pp. 9–22.

[2] D. Cope, Journal of New Music Research, **18(1-2)**, 117-139 (1987).

[3] D. Cope, Experiment Ucsc.edu., retrieved from: http://artsites.ucsc.edu/faculty/cope/experiments.html.

[4] J. J. Bharucha and P. M. Todd, Computer Music Journal, **13(4)**, 44–53 (1989).

[5] P. Dhariwal, H. Jun, C. Payne, et al. arXiv preprint arXiv:2005.00341 (2020).

[6] G. Hadjeres, F. Pachet and F. Nielsen, "Deepbach: a steerable model for bach chorales generation", In International conference on machine learning (2017) pp. 1362-1371.

[7] P. Dhariwal, H. Jun and C. M. Payne, Jukebox. OpenAI. Retrieved from: https://openai.com/research/jukebox.

[8] P. Domingos, Communications of the ACM, 55(10), 78-87 (2012).

[9] J. Daintith and E. Wright, *A dictionary of computing* (Oxford University Press, Oxford, 2010).

[10] What is Supervised Learning? | IBM. (2023). Ibm.com. Retrieved from https://www.ibm.com/topics/supervised-learning.

[11] N. Kotecha and P. Young, arXiv preprint arXiv:1804.07300 (2018).

[12] F. Attneave, The Journal of Aesthetics and Art Criticism, **17(4)**, 503–510 (1959).

[13] P. Lu, X. Tan, B. Yu, et al., arXiv preprint arXiv:2208.14345 (2022).

[14] K. Ebcioğlu, Computer Music Journal, **12(3)**, 43–51 (1988).

[15] Y. C. Yeh, W. Y. Hsiao, S. Fukayama, et al. Journal of New Music Research, **50(1)**, 37-51 (2021).

[16] D. Herremans, C. H. Chuan and E. Chew, ACM Computing Surveys (CSUR), **50(5)**, 1-30, (2017).

[17] B. L. Sturm, J. F. Santos, O. Ben-Tal, I. Korshunova, arXiv preprint arXiv:1604.08723 (2016).

[18] J. P. Briot, G. Hadjeres, and F. D. Pachet, arXiv preprint arXiv:1709.01620 (2017).

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., Communications of the ACM, **63(11)**, 139-144 (2020),

[20] A. Creswell, T. White, V. Dumoulin, et al., IEEE signal processing magazine, **35(1)**, 53-65 (2018).

[21] K. Choi, G. Fazekas and M. Sandler, arXiv preprint arXiv:1604.05358 (2016).

[22] M. Civit, J. Civit-Masot, F. Cuadrado and M. J. Escalona, Expert Systems with Applications, **1**, 118190 (2022).