International Conference on Advances in Materials, Computing and Communication Technologies
In Association with International Journal of Scientific Research in Science and Technology
Volume 9 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

Study of Fundamental Approaches in Regards with Automatic Music Generation using AI

Vincy Kaushik[1], Pravin Kumar Mishra[2]
[1]Bharat Institute of Technology, Meerut, Uttar Pradesh, India
[2]Assistant Professor, Bharat Institute of Technology, Meerut, Uttar Pradesh, India

ABSTRACT
Computational creativity is an interdisciplinary topic in which computers attempt to achieve creative behaviors. One of the prolifest areas of music generation, which employs computer methods to make music, is known as algorithmic composition or music meta creation. It is often difficult to determine specific objectives and to monitor issues that state-of-the-art systems can deem addressed and what additional advancements will instead be necessary. In this survey, we attempt to provide people who want to study computer creativity and music production with a thorough introduction. We examine the state-of-the-art systems of Music Generation by providing instances of the primary techniques to creating music and identifying the open issues mentioned in earlier studies. We mention works that have offered answers to each of these issues and that describe what still needs to be done and suggested guidelines for additional study. This paper combined my two passions – music and deep learning – to create an automatic music generation model. We are thrilled to share our approach with you, to enable you to generate your music! We will first quickly understand the concept of automatic music generation before diving into the different approaches we can use to perform this. Finally, we will fire up Python and design our automatic music generation model.

Keywords – Automatic Music Generation, Stochastic Music, CNN, LSTM, WaveNet, Deep Learning architectures, WaveNet

I. INTRODUCTION

The first computer music was published in 1957. The tune, called 'The Silver Scale,' was 17 seconds long by its composer Newman Guttman. It was created by the program Mathews at the Bell Laboratories for sound synths, called Music I. [1] In the same year, the first soundtrack created by a computer was "The Illiac Suite." Lejaren A. Hiller and Leonard M. Isaacson, both musicians and scientists, were human "meta-composers." It used stochastic models (Markov chain) as an early example of algorithmic composition, as also rules for filtering material according to desired qualities.[2]

The release in 1983 of the DX 7 synthesizer Yamaha based the Chowning model on a synthesis based on frequency modulation was a breakthrough in the field of sound synthesis (FM).

In the same year, the MIDI6 interface was released to interact with different software and instruments. Another milestone in the processing environment

utilized for real-time synthesis and interactive achievement was the creation of Puckette at the Max/MSP IRCAM.[3]

In the early 1960s, Iannis Xenakis examined the notion of stochastic composition concerning algorithmic composition. 7[20], called 'Atrees' in his 1962 piece. The approach involves calculating numerous options from several possibilities established by the composer utilizing computer-fast calculations to produce samples of the selected musical compositions. In another way, the grammars and rules used to describe the style of a certain corpus or, more generally, of tonal music theory, following the starting direction of "Illic suite." The creation of a 4-part chorale in the 80s is an example of Ebcioglu's composition program CHORAL. [4]

The method known as Experiments in Musical Intelligence (EMI) by David Cope at the end of the 1980s increased that approach to the ability of a group of composer scores to build their grammar and database of rules[18]. Perhaps we're not a physicist like Mr. A.R Rahman, but his ideas on music are entirely our support! When we haven't opened up my music player, we can't recall one day.

We've always dreamed of music creating, but haven't got the instrument hang. It was till we found profound knowledge. We were able to construct our original musical composition using specific approaches and frames without learning the theory of music.

A. What is Automatic Music Generation?

Music is defined as a collection of tones of various frequencies. The Generation of Automatic Music is a technique of writing a brief piece of music with minimal human interaction.

B. What could be the simplest form of generating music?

It all began with the random choice of sounds and their combination to create a piece of music. Mozart offered these random sound options in a dice game in 1787. He has painstakingly created around 272 tones! Then, depending on the total of 2 dices he picked a tonne. [5]

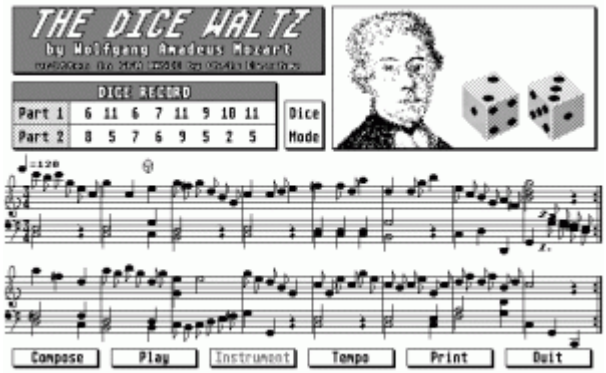

Figure 1: Another interesting idea was to make use of musical grammar to generate music.

Iannis Xenakis employed statistics and probability principles in the early 1950s to write music, which was commonly known as Stochastic Music. He described music as a string of components (or sounds) happening incidentally. Therefore, the theory he formulated was stochastic. His random choice of components depended entirely on mathematical ideas.

Deep Learning architectures have recently become the cutting edge for the generation of automatic music. This article discusses two distinct techniques in WaveNet and LSTM (Long Short Term Speed) networks to automated music composition.

- Must-Read Tutorial to Learn Sequence Modeling
- Essentials of Deep Learning: Introduction to Long Short Term Memory (LSTM)
- A Comprehensive Tutorial to learn Convolutional Neural Networks (CNNs) from Scratch

C. What are the Constituent Elements of Music?

The music is mostly made up of chords and notes. From the standpoint of the piano instrument we will explain these terms:

- ***Note:*** A single key sound is termed a note Note Note

- ***Chords:*** Simultaneously a chord is considered the sound generated by two or more keys. Most chords usually have a minimum of 3 keynotes.
- ***Octave:*** The octave is referred to as a repeating pattern. There are 7 black and white keys for each octave.

II. DIFFERENT APPROACHES TO AUTOMATIC MUSIC GENERATION

Two Deep Learning-based systems for automated music generation — WaveNet and LSTM – are discussed in detail. But why only designs of deep learning? Deep Learning is a field that is inspired by a brain structure. These networks automatically extract the features from the dataset and can train any non-linear function. For this reason, neural networks are called:

A. Universal Functional Approximations.

Therefore, Deep Learning models are the cutting-edge model in many areas such as Natural Language Processing (NLP) and Computer Vision. Let's look at how these music composition models may be built. [7]

Method 1: WaveNet Usage

WaveNet's main goal is to produce fresh samples of the original data distribution. Therefore, the generative model is known.

Wavenet is like an NLP linguistic model.

The model predicts the next word, given a series of words, in a language model. In WaveNet, as in the language model, a sequence of samples is predicted for the following sample.

Method 2: Model for Long Short Term (LSTM)

Long Short Term Memory Model is a version of Recurrent Neural Nets (RNNs), also known as LSTM, which may capture longer-term input dependencies. LSTM offers a wide range of sequence-to-sequence modeling applications, such as speech recognition, text summary, video classification, etc.

III. TRAINING OF OUR MODEL

Let me explore in detail how we may use these two techniques to train our model.

Wavenet: The phase of training

See how we may construct sequences for input and output.

WaveNet Inputs

The raw audio wave chunk is used as the input for WaveNet. Raw audio wave refers to the time series domain representation for a wave.

An audio wave is represented inside the time-series domain, in the form of amplitude values recorded at various time intervals:

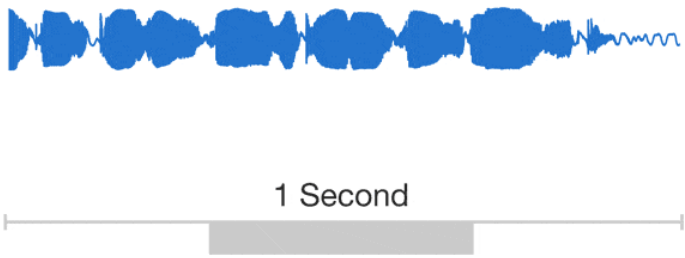

Figure 2: Audio wave

The output of the WaveNet:

Given the amplitude sequence, the following amplitude values are predicted by WaveNet. With the assistance of an example, let us grasp it. Consider a 5-second audio wave with 16,000 samples (that is 16,000 samples per second). Now, for 5 seconds we have captured 80,000 samples at different intervals. Bring the sound into equal-size pieces, say 1024

(which is a hyperparameter). [10] The following graphic shows the sequences for the model input and output:

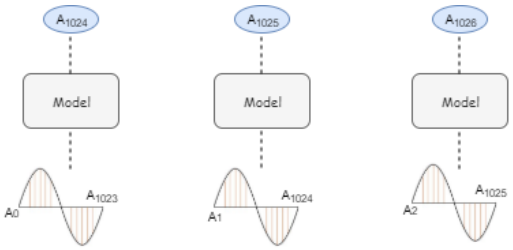

Figure 3: Input and Output of first 3 chunks

For the rest of the pieces, we may follow a similar approach. From the foregoing, we can deduce that the output of each chunk solely depends on past information (i.e. prior times) and not on future times. This task is thus known as an automated task and the model is called an automatic model.[12]

Phase of deduction

We will attempt to create additional samples throughout the inference phase. Let's see how this is done:

1. Choose a random sample value array from a model starting point
2. The model now displays the distribution of probability over all samples choose the value with the maximum probability and append it to an array of samples
3. Delete the first element and pass it as an input for the next iteration
4. Repeat steps 2 and 4 for a certain number of iterations

IV. UNDERSTANDING THE WAVENET ARCHITECTURE

Causal Dilated 1D Convolution layers are the building elements of WaveNet. First, let us grasp the relevance of the ideas involved. One of the principal reasons is that the characteristics of an input should be extracted. When image processing, for example, we use a characteristic map to combine the picture with a filter. [8]

Convolution is a combining 2 functions mathematical procedure. Convolution is a linear mix of specific picture portions with the kernel when processing the image.

Figure 4: Dilated 1D Convolution layers

You can browse through the below article to read more about convolution: Architecture of Convolutional Neural Networks (CNNs) Demystified

What is 1D Convolution?

The objective of 1D convolution is similar to the Long Short Term Memory model. It is used to solve similar tasks to those of LSTM. In 1D convolution, a kernel or a filter moves along only one direction:

The result of the conversion varies according to kernel size, input form, padding type, and step. Now I'll go over several padding kinds to explain the relevance of utilizing a 1D Convolution Dilated Causal Layer. [11]

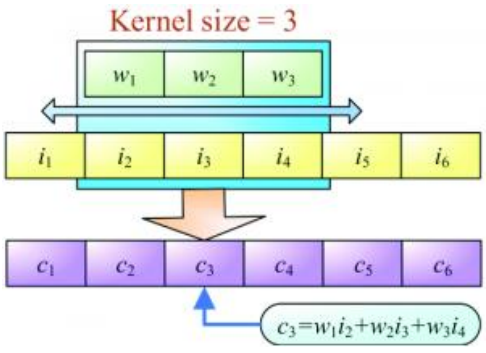

Figure 5: 1D convolution

The input and output sequences differ by length when we set the padding to be valid. The output length is smaller than the input length:

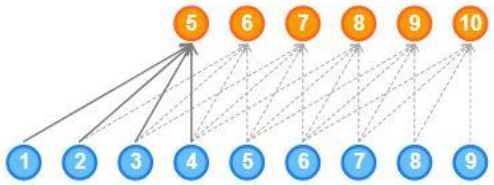

Figure 6: Dilated Causal 1D Convolution layers

When we set the padding to the same, zeroes are padded on either side of the input sequence to make the length of input and output equal:

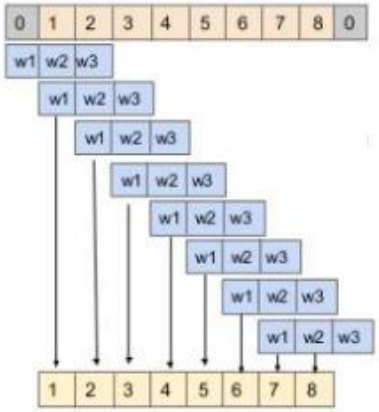

Figure 7: Dilated Causal 1D Convolution layers

1D Convolution pros:

- Collects the sequential information in the input sequence
- Training is significantly quicker than GRU or LSTM due to the lack of recurring connections.

Convolution's adverse effects:

- The output is torn into the previous t-1 and the future t+1 time stages when the padding is set to equal.
- It, therefore, contradicts the concept of self-regression
- When padding is configured to be effective to calculate residual connections in the input and output sequences differ in length (which will be covered later)

What is 1D Causal Convolution?

In simpler terms, normal and causal convolutions differ only in padding. In causal convolution, zeroes are added to the left of the input sequence to preserve the principle of autoregressive:

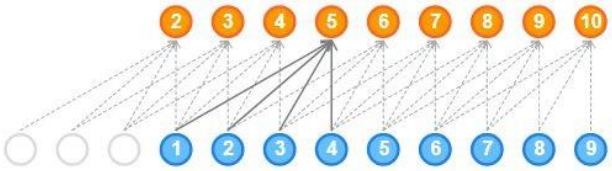

Figure 8: Principle Of Autoregressive

Pros of Causal 1D convolution:

- Causal Convolution does not take into consideration the future periods, a condition for the construction of the Generative model

Cons of Causal 1D convolution:

- Causal convolution cannot look back into the past or the timesteps that occurred earlier in the sequence. Hence, causal convolution has a very low receptive field. The receptive field of a network refers to the number of inputs influencing an output:

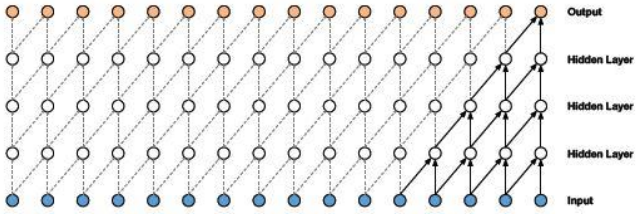

Figure 9: Causal 1D convolution

The outcome is only impacted by 5 inputs, as you can see. Therefore, the network's reception field is 5, which is quite low. A big kernel can also contribute to the receptive field of a network, but bear in mind that computer complexity is increasing. This leads us to the wonderful Dilated 1D Causal Convolution idea. [13]

What is Dilated 1D Causal Convolution?

Dilated 1D convolution is characterized as a causal 1D layer with gaps or voids between kernel values.

The dilation rate specifies the number of spaces to be added. It specifies a network's reception area. A size kernel and dilation rate d have d-1 holes between all kernel values.

In addition to a 7* 7 input with dilation rate 2, the reception area of a 3*3 kernel with the same dilation rate 2 is 5*5. [14]

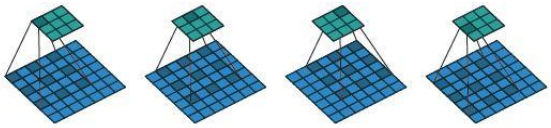

Figure 10: Causal 1D convolution layer

Pros of Dilated 1D Causal Convolution:

- The dilated 1D convolution network increases the receptive field by exponentially increasing the dilation rate at every hidden layer:

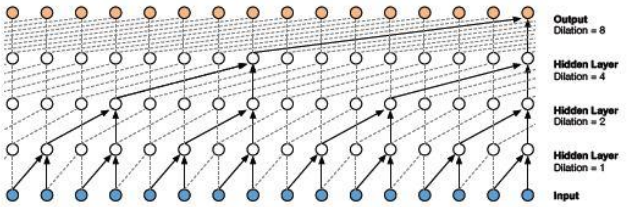

Figure 11: Dilated 1D Causal Convolution

As you can see here, the output is influenced by all the inputs. Hence, the receptive field of the network is 16. [15]

Residual Block of WaveNet:

A building block contains Residual and Skip connections which are just added to speed up the convergence of the model:

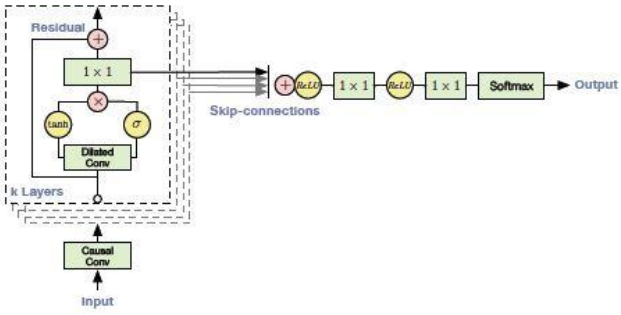

Figure 12: Residual Block of WaveNet

The Workflow of WaveNet:

- Input is fed into a causal 1D convolution
- The output is then fed to 2 different dilated 1D convolution layers with sigmoid and tanh activations
- The element-wise multiplication of 2 different activation values results in a skip connection
- And the element-wise addition of a skip connection and output of causal 1D results in the residual

Long Short Term Memory (LSTM) Approach

The Long Short Term Memory (LSTM) model is another technique for automated music production. Input and output sequences are prepared like WaveNet. Amplitude value is entered into the Long Short Term Memory cell at each time, which then calculates the vector concealed and transmits it to the next times. [16]

The currently hidden vector at timestep $\mathbf{h_t}$ is computed based on the current input $\mathbf{a_t}$ and previously hidden vector $\mathbf{h_{t-1}}$. This is how the sequential information is captured in any Recurrent Neural Network:

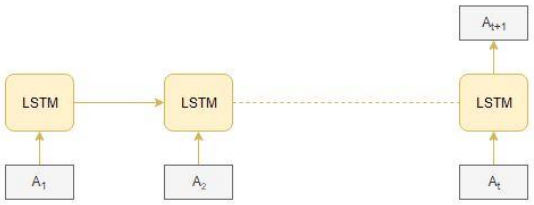

Figure 13: Recurrent Neural Network

Pros of LSTM:

- Captures the sequential information present in the input sequence

Cons of LSTM:

- It consumes a lot of time for training since it processes the inputs sequentially

V. RESULT

From the pattern above, we may deduce that the majority of the notes are extremely low. Let us thus preserve the high-frequency sounds and discard those that are low frequency. The threshold is defined as 50, here. However, it is possible to modify the

parameter. As you can see above, there are about 170 notes often present. Now let us produce fresh music files including only the most common notes. Fantastic, okay? But it doesn't halt your studying here. Remember that a baseline model has been created.

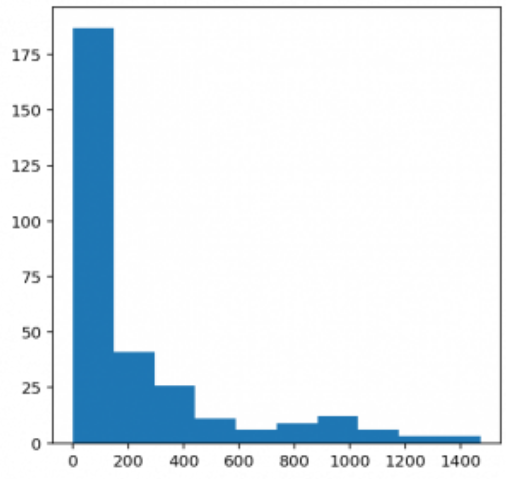

Figure 14: Result

The performance of the model may still be improved in many ways:

- As the size of the training dataset is small, we can fine-tune a pre-trained model to build a robust system
- Collect as much training data as you can since the deep learning model generalizes well on the larger datasets

VI. CONCLUSION

Deep Learning has a wide range of applications in our daily life. The key steps in solving any problem understand the problem statement, formulating it, and defining the architecture to solve the problem. We had a lot of fun (and learning) while working on this project. Music is a passion of mine and it was quite intriguing combining deep learning with that. We are looking forward to hearing your approach to the problem in the comments section. And if you have any feedback on this article or any doubts/queries, kindly share them in the comments section below and We will get back to you.

VII. REFERENCES

[1]. Amabile, T. M. (1983a). "A consensual technique for creativity assessment," in The Social Psychology of Creativity, Springer Series in Social Psychology, ed T. M. Amabile (New York, NY: Springer), 37–63.

[2]. Amabile, T. M. (1983b). The social psychology of creativity: a componential conceptualization. J. Pers. Soc. Psychol. 45, 357–376. doi: 10.1037/0022-3514.45.2.357

[3]. Amabile, T. M., Conti, R., Coon, H., Lazenby, J., and Herron, M. (1996). Assessing the work environment for creativity. Acad. Manage. J. 39, 1154–1184. doi: 10.5465/256995

[4]. Anders, T., and Miranda, E. R. (2011). Constraint programming systems for modeling music theories and composition. ACM Comput. Surv. 43, 1–38. doi: 10.1145/1978802.1978809

[5]. Anderson, C., Eigenfeldt, A., and Pasquier, P. (2013). "The generative electronic dance music algorithmic system (GEDMAS)," in Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE'13) Conference (Palo Alto, CA), 4.

[6]. Ariza, C. (2009). The interrogator as a critic: the turning test and the evaluation of generative music systems. Comput. Music J. 33, 48–70. doi: 10.1162/comj.2009.33.2.48

[7]. Baer, J., and McKool, S. S. (2009). "Assessing creativity using the consensual assessment technique," in Handbook of Research on Assessment Technologies, Methods, and Applications in Higher Education (Hershey, PA: IGI Global), 65–77.

[8]. Bell, C. (2011). Algorithmic music composition using dynamic Markov chains and genetic algorithms. J. Comput. Sci. Coll. 27, 99–107.

[9]. Bidlack, R. (1992). Chaotic systems as simple (but complex) compositional algorithms.

Compu. Music J. 16, 33–47. doi: 10.2307/3680849

[10]. Biles, J., Anderson, P., and Loggi, L. (1996). "Neural network fitness functions for a musical IGA," in Proceedings of the Soft Computing Conference (Reading, UK), 11.

[11]. Biles, J. A. (1994). "GenJam: a genetic algorithm for generating jazz solos," in ICMC, Vol. 94 (Ann Arbor, MI), 131–137.

[12]. Biles, J. A. (2001). "Autonomous GenJam: eliminating the fitness bottleneck by eliminating fitness," in Proceedings of the 2001 Genetic and Evolutionary Computation Conference Workshop Program (San Francisco, CA), 7.

[13]. Boden, M. A. (1998). Creativity and artificial intelligence. Artif. Intell. 103, 347–356. doi: 10.1016/S0004-3702(98)00055-1

[14]. Boden, M. A. (2004). The Creative Mind: Myths and Mechanisms. London: Routledge.

[15]. Bodily, P. M., and Ventura, D. (2018). "Musical metacreation: past, present, and future," in Mume 2018 (Salamanca: University of Salamanca), 5.

[16]. Brèmaud, P. (2013). Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues, Vol 31. New York, NY: Springer Science & Business Media.

[17]. Bringsjord, S., Bello, P., and Ferrucci, D. (2003). "Creativity, the turing test, and the (better) Lovelace test," in The Turing Test, ed J. H. Moor (Dordrecht: Springer), 215–239.

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

[19]. Briot, J.-P., Hadjeres, G., and Pachet, F.-D. (2020). Deep Learning Techniques for Music Generation. Computational Synthesis and Creative Systems. Basel: Springer International Publishing.

[20]. Briot, J.-P., and Pachet, F. (2018). Deep learning for music generation: challenges and directions. Neural Comput. Appl. 32, 981–993. doi: 10.1007/s00521-018-3813-6