ijRASET

International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

www.ijraset.com

Call: ◎08813907089 | E-mail ID: ijraset@gmail.com

Polyphonic Music Generation

Vineet Tiwari[1], Pratheesh Shivaprasad[2], Rushikesh Pokharkhar[3], Prof. Madhura Vyawahare[4]

[1, 2, 3, 4]*Pillai College of Engineering, University of Mumbai*

Abstract: Recent advancements in neural networks have helped algorithms to generate musical sound that is comparable to sounds composed by humans. In this project, polyphonic music is generated using Deep Learning neural network. Unlike generating images and videos, generating music is a bit different; generating music is time dependent. The used algorithm for this purpose is Variational Autoencoders. By generating a sequential note of measures by already defined chord progression, our model can produce musical notes with convincing long-term structure. These algorithms have tuneable parameters and such types of algorithms yield better results and are practically useful for artists, filmmakers and many others in their creative tasks. In the future, users can provide tracks which consist of all the instruments and gives the similar track as the input. For e.g. given a specific track composed by a human, algorithms can create additional tracks similar to it.
Keywords: Music Generation, Deep Learning, LSTM, VAE(Variational Auto Encoders).

I. INTRODUCTION

Our goal with this project is to create royalty free music using Deep Learning for content creators and songwriters alike. Generating music is quite difficult and complex and requires complete domain knowledge. So, long short-term memory (LSTM) algorithm is used for generating musical sound. The music format that will be fed to the algorithm created will be MIDI which will be used for training the training module. An important point to note is that the music created will never be exactly the same as the music fed for training and learning.
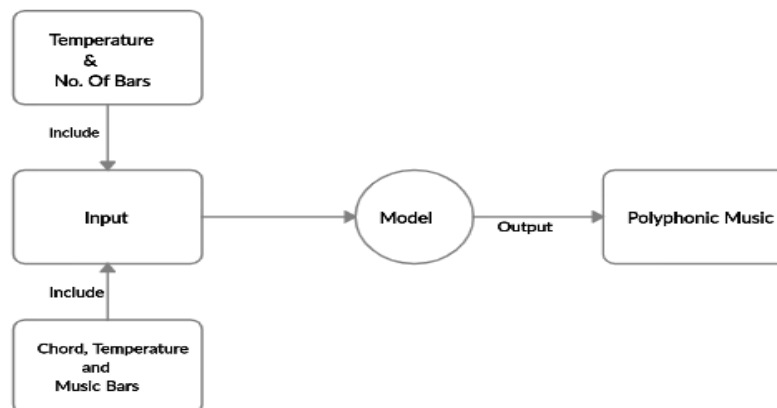

Fig.1 Basic implementation

II. LITERATURE SURVEY

Hao-Wen Dong and Yi-Hsuan Yang has explained how previously existing models needed a lot of post processing after creating the piano rolls such as Hard Thresholding (HT) or Bernoulli Sampling (BS) to obtain the final results..The advantage is that it can produce some great music but due to using GAN the overfitting can occur due to less dataset.

Li-Chia Yang, Szu-Yu Chou, Yi-Hsuan Yang has explained how RNNs were used previously for generating musical notes because RNN is considered best for sequence generation. In this the limitation of GAN overcomes due to the use of RNN, but RNN cannot train for long sequence and its main disadvantage is this factor.

Olof Mogren has explained that, investigating the feasibility of using adversarial training for a sequential model with continuous data, and evaluate it using classical music in freely available midi files'-RNN-GAN (Continuous RNN-GAN). In this both the factor of overfitting and not training the longer sequence is overcome because of combining the RNN and GAN.

Hao-Wen Dong,Wen-Yi Hsiao, Li-Chia Yang,Yi-Hsuan Yang1 has explained about generating music using CNN (which is known to be good at finding local,translation-invariant patterns).In this the great advantage is that CNN can learn great form its feature but this advantage is used mostly by images and not sure about the audio/music.

Nicholas Trieu and Robert M. Keller has explained that for the purpose of creating a jazz teaching tool in the open-source Impro-Visor (Improvisation Advisor) application, we trained JazzGAN, a generative adversarial network (GAN) using recurrent neural networks (RNN) to improvise monophonic jazz melodies over chord progressions..In this both the factor of overfitting and not training the longer sequence is overcome because of combining the RNN and GAN.

Jordi Pons,Thomas Lidy,Xavier Serra has explained that a common criticism of deep learning relates to the difficulty in understanding the underlying relationships that the neural networks are learning, thus behaving like a black-box. With these experiments we have been able to understand what some deep learning-based algorithms can learn from a particular set of data.

Huanru Henry Mao, Taylor Shin, Garrison W. Cottrell has explained about Recent advances in deep neural networks which have enabled algorithms to compose music that is comparable to music composed by humans. Evaluation of this model using human raters shows that it has improved over the Biaxial LSTM approach.

Kratarth Goel, Raunaq Vohra, and J.K. Sahoo has explained that the Authors have proposed a generic technique to model temporal dependencies and sequences using a combination of a recurrent neural network and a Deep Belief Network. They have applied this technique to the task of polyphonic music generation.

Ian Simon Jesse Engel, Adam Roberts Curtis Hawthorne, Colin Raffle Douglas has explained that the author has discovering and exploring the underlying structure of multi-instrumental music using learning-based approaches remains an open problem. They extend the recent VAE model to represent multitrack polyphonic measures as vectors in a latent space. The advantage is that it can produce different types of music like using chords, temperature, number of bars etc.

III. PROPOSED SYSTEM

The Variational Autoencoder (VAE) has proven to be an effective algorithm for producing semantically meaningful latent representations for data. However, it has thus far seen limited application to sequential data, it was found that existing recurrent Variational Autoencoder model have difficulty in modelling sequences with LSTM.
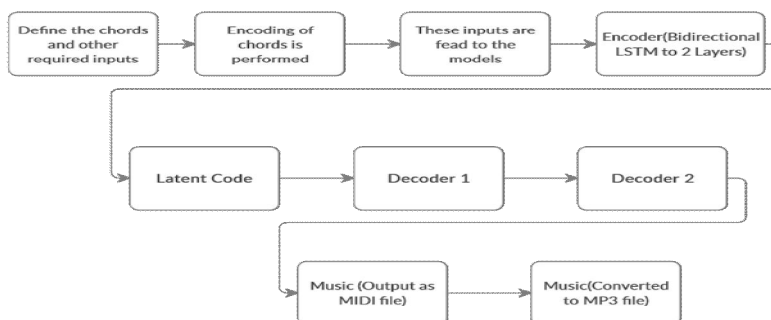

Fig.2 High Level Architecture of Polyphonic Music Generator

The high-level architecture of a Polyphonic Music Generator is depicted in Figure 2. The music generation process is performed in nine steps, each of which is handled by a separate component:

1) *Define the Chords and other Required Inputs:* Generation of music by giving some chords as input. Four different chords along with the temperature and number of bars will be given as input and based on that the model will generate music containing the specified chords.

2) *Encoding of Chords is Performed:* Once the user enters the required chords, they are then encoded using One Hot Encoding so that the model understands what chords the songs must include.

3) *Encoder:* It consists of a two - layer bidirectional LSTM network. The inputs are processed in this stage and then fed to a fully connected layer. The bidirectional LSTM layer is used so that the model can understand the long - term context of the sequence.

4) *Decoder:* The decoder in this model is a hierarchical RNN. It uses the latent code generated by the encoder and starts generating the output sequence regressively. The reason for using a hierarchical RNN is because the simple RNNs resulted in poor sampling and reconstruction for sequences due to the vanishing gradient problem.

5) *Music Output:* The output sequence is then converted to a MIDI music file so that it can be saved in the system. This MIDI format is not usually used as a normal music format and requires a specific sound font file to play, so we convert this file to an MP3 format using FluidSynth and LAME libraries.

A. Data and Training

The VAE uses the MIDI format for training. MIDI is the format in which the music is stored. There are different formats of music ie MP3, Musepack, AAC and many more. The VAE has used the MIDI files and generated some checkpoints and weights files, we have used those checkpoints and continued to build the project. This model used the monophonic music of different instruments like piano rolls, drums, flute and many more. Then this monophonic music is trained together to generate the polyphonic music. For the comparison process the generated music is compared with the MIDI files which are available.

IV. EXISTING SYSTEM AND COMPARISION

GAN basically comprises two primary networks, namely generator and discriminator. The generator generates random samples from the population and the job of the discriminator is to differentiate this random sample as real or generated. In each and every iteration, the discriminator classifies the generated sample as real or fake and then passes this information onto the generator for it to improve further. Therefore, the generator becomes better at generating samples and the differentiator becomes better at recognizing real or fake samples. After multiple iterations, the generator becomes so good at generating samples that it is technically difficult for the differentiator to recognize it as a fake sample. It can be formally said as a min-max game between the generator and the discriminator.
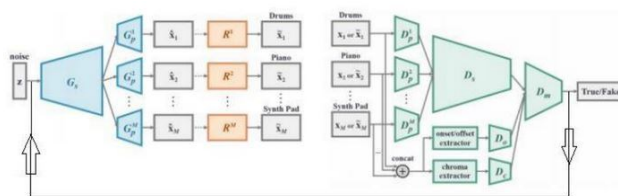

Fig.3 Existing Model [4]

The music produced by GAN was not so accurate as compared to VAE, the GAN model converts the MIDI files in text format so that model can understand the music and generate the different pattern with was never given in training process. The produced music is 0.1 seconds, so to produce music in minutes, the models needs to produce 1000's of small 0.1 seconds file and the combine those files to produce the 1 min 40 sec music, whereas the VAE produce the music in minutes. The accuracy can't be calculated based on parameters because this are all generating algorithm, but when the music was played the VAE generated better results compared to the GAN model.

V. IMPLEMENTATION

There are two types of models for music generation: chord conditional model and chord unconditional model. In this project we have used VAE (Variational Auto-Encoders) for generating music. Fig 3, below shows in-depth knowledge about the generating different types of music using VAE.
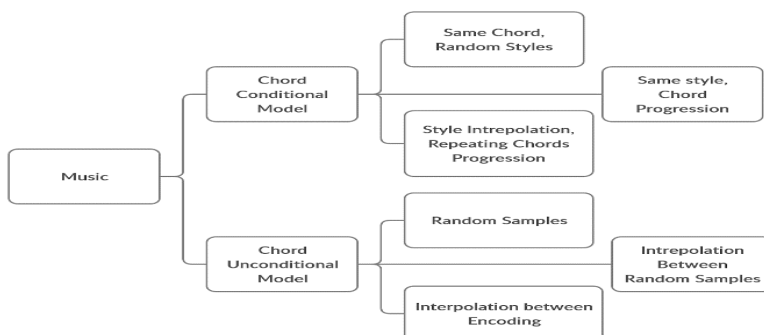

Fig.4 Methods of Generating Music

In the Chord Unconditional model, the model takes two types of inputs. The inputs are temperature and number of bars. The temperature decides the intensity of the music, like the number of instruments that will be used at same time and number of bars decides the tempo of the music, both the inputs are numeric. Once the input is provided, the model generates small sequences of music based on the input.

The other method is using chord conditional model, this model generates the music by giving some chords as input. Four different chords along with the temperature and number of bars is given as input and based on that the model generates music containing the specified chords.

To demonstrate these scenarios, a Flask web application is designed which asks for input from the user and once submitted generates music in the backend and is displayed to the user to listen and download the music file. The Landing page also consists of a set of cherry-picked music that the model has generated.

VI. RESULTS

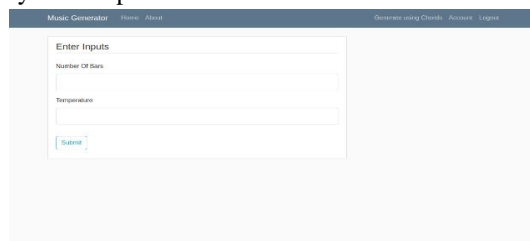1) *Input Type I:* In this type of input, only the temperature and no. of bars are entered.

Fig.5 Input Type I

2) *Input Type II:* In this type of input, along with temperature and no. of bars, chords can also be entered.
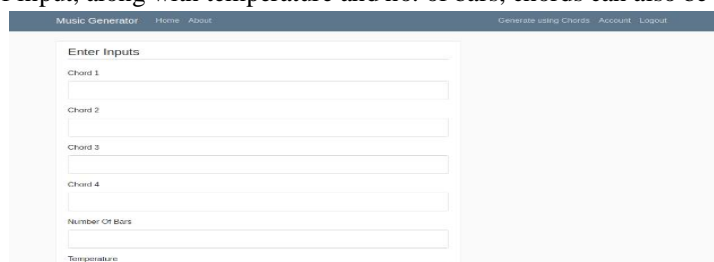

Fig.5.1 Input Type II

3) *Output:* Generated music is displayed on the output page. The user can listen to it online and download it as well.
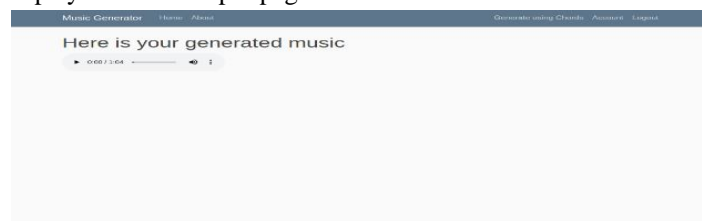

Fig.5.2 Output

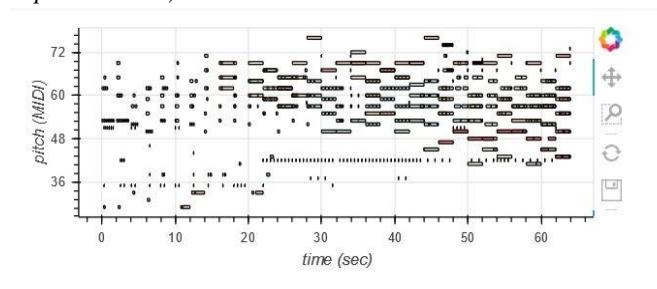4) *Produced Music (Graphical Representation)*

Fig.5.3 Produced Music (Graphical Representation)

Fig 5.3 shows graphical representation of the generated music. The pitch of the music is plotted on the Y-axis against time (in sec) on the X-axis.

The generated music samples by our project is available on http://iampratheesh.github.io/musicgenerator , all the samples all generated by VAE(model used in this project).

As mentioned previously, users can provide the inputs in two ways. The first method is easier as the user has to only provide two inputs namely Number of Bars and Temperature. The music generated using this option is generic and randomly generates sequence based on the input. In the second method, the user can mention all the chords that need to be there in the music. This is a more personalized approach as the user gets the generated music based on his/her preference.

Once the music is generated (which is in the MIDI format originally) it is then converted to mp3 format and then displayed on the browser. Users can listen to the generated music online and also have the option to download the song.

VII. CONCLUSION

In this paper, the polyphonic music generation is presented. The music is generated using VAE i.e. Variational Auto-Encoders. Various techniques for generating music like using temperature, number of bars, chords is mentioned in this paper. The dataset which is to be used is also described in this report. The dataset that is used in this project is the Multitrack music dataset. The applications of this domain are identified and presented.

VIII. FUTURE SCOPE

The architecture can be made more user friendly, so the users from non-music background can easily understand how to make Multitrack Music/Polyphonic Music. The process of generating music takes a lot of computing power which eventually takes a lot of time. This process can be further optimized so as to take less processing time further improving the user experience. Additional features like generating music based on a music file provided by the user can also be implemented to provide a personalized feel.

REFERENCES

[1] Dong, Hao-Wen, and Yi-Hsuan Yang. "Convolutional generative adversarial networks with binary neurons for polyphonic music generation." arXiv preprint arXiv:1804.09399 (2018).

[2] Yang, Li-Chia, Szu-Yu Chou, and Yi-Hsuan Yang. "Midinet: A convolutional generative adversarial network for symbolic-domain music generation using 1d and 2d conditions." arXiv preprint arXiv:1703.10847 160 (2017).

[3] Mogren, Olof. "C-RNN-GAN: Continuous recurrent neural networks with adversarial training." arXiv preprint arXiv:1611.09904 (2016).

[4] Dong, Hao-Wen, et al. "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment." Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[5] Trieu, Nicholas, and Robert M. Keller. "JazzGAN: Improvising with generative adversarial networks." Proc. of the 6th International Workshop on Musical Metacreation (MUME). 2018.

[6] Pons, Jordi, Thomas Lidy, and Xavier Serra. "Experimenting with musically motivated convolutional neural networks." 2016 14th international workshop on content-based multimedia indexing (CBMI). IEEE, 2016.

[7] Mao, Huanru Henry, Taylor Shin, and Garrison Cottrell. "DeepJ: Style-specific music generation." 2018 IEEE 12th International Conference on Semantic Computing (ICSC). IEEE, 2018.

[8] Goel, Kratarth, Raunaq Vohra, and J. K. Sahoo. "Polyphonic music generation by modeling temporal dependencies using a rnn-dbn." International Conference on Artificial Neural Networks. Springer, Cham, 2014.

[9] Choi, Keunwoo, et al. "Auralisation of deep convolutional neural networks: Listening to learned features." Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR. 2015.

[10] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." arXiv preprint arXiv:1701.07875 (2017).

[11] Dumoulin, Vincent, and Francesco Visin. "A guide to convolution arithmetic for deep learning." arXiv preprint arXiv:1603.07285 (2016)..

[12] Bretan, Mason, Gil Weinberg, and Larry Heck. "A unit selection methodology for music generation using deep neural networks." arXiv preprint arXiv:1612.03789 (2016)..

[13] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5.2 (1994): 157-166.

[14] Eck, Douglas, and Juergen Schmidhuber. "Finding temporal structure in music: Blues improvisation with LSTM recurrent networks." Proceedings of the 12th IEEE workshop on neural networks for signal processing. IEEE, 2002.

[15] Dieleman, Sander, and Benjamin Schrauwen. "End-to-end learning for music audio." 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014..

[16] Simon, Ian, et al. "Learning a latent space of multitrack measures." arXiv preprint arXiv:1806.00195 (2018).

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)