e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

ORCHESTRATING MELODIES THROUGH SPECTROGRAM-DRIVEN AUTOENCODERS

Uttara Bahad[*1], Rishika Sardana[*2], Sakshi Sanjay Chalke[*3], Krutika Prakash Bhagane[*4], Ujwala Gaikwad[*5]

[*1,2,3]Student, Department Of Computer Engineering Terna Engineering College, Nerul, Maharastra, India.

[*4]Student, Department Of Computer Engineering Pillai HOC College Of Engineering And Technology, Rasayani, Maharashtra, India.

[*5]Assistant Professor, Terna Engineering College, Nerul, Maharastra, India.

DOI : https://www.doi.org/10.56726/IRJMETS45083

ABSTRACT

The thing that is common to all the people across the world, or the thing that transforms boundaries and connects us all is music. According to statistics every human hears approximately 15 minutes of music every day. Music is one category that all humans consume in all their moods, no matter if they are feeling happy, sad, angry, or lonely. Hence to cater to this need of music generating industry is a multi-billion industry. Everyday millions of songs/music all around the world is being produced and a variety of people like directors, producers, composers, singers, music-writers are included in this lengthy process. Through our research we aim to produce a mock model which will make music creating a less time consuming, lengthy, and costly process. This can be done by using AutoEncoders which are a part of unsupervised neural networks. This machine learning technique will help us generate music, which follows a specific pattern of chords. We aim to create a realistic music tone which is a synthetic data. This can be done by training VAE with Keras and by performing Short-Time/Inverse Short-Time Fourier Transformations.

Keywords: Deep Learning, Keras, Music, Neural Networks, Spectrogram, TensorFlow.

I. INTRODUCTION

Music is a big category with multiple genres but this music follows some similar observations or patters. First, music has a hierarchical structure with dependencies throughout time and is periodic in nature. It is made up of a variety of interconnected instruments that develop over time.

Since chords and melodies are arranged according to a specific pattern, they have several outputs with time steps. This makes analyzing the variation of music with time easily and further this helps in training our machine learning model with neural networks and autoencoders. This can be used by a different group of people like firstly technologists who study music or the music industry. This can also be used by machine learning and deep learning engineers as it will help them explore more about deep learning and variational autoencoders. Software engineers or audio programmers can also use this research and add their working to it as well.

Via this research we will use pre-existing sounds and apply deep learning neural networks to then train our model and then with already made sounds to generate new, unique, and fresh music.

How many types of sounds are there that can be generated using neural networks?

Some of the most common/famous sounds that can be generated are:

a) Sound Design: Sounds of raindrops, water splashing, a glass breaking are sounds that are heard frequently in actual songs, this is known as sound designed, that is sound designed for a specific object.

b) Music Notes: Single Music notes can be generated which in the actual music industry is known as samples by the producers or the composers. These single notes then further be used to develop a pattern to create a full long musical sound.

c) Speech: Speech is a sound that can be developed via texts using text-to-speech machine learning techniques.

d) Music: Recently voices cover of different singers is on a rise this can be also done using this deep learning techniques.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

II. METHODOLOGY

There are various ways to give the inputs for sound generation like

a) Conditioning generation: We can pass semantic information to train our model. For example: “Give me a sad song with happy beats” or “Give me a Nicki Minaj song without heavy rap.” So, this way by applying conditions the network can learn to identify between different moods, artists, genres, or situations of sound tunes.

b) Autonomous generation: In this the model generates sounds without any input. It hence gives a free, fresh, and unique sound with a random pattern.

c) Continuation generation: In this model we prove the training with the help of a seed, for example any random sentence said by a speaker. The model is trained to analyses the meaning of the words spoken on the sentence and to continue the music generation according to the words, mood, rhythm or flow of that line or passage.

Raw Audio or spectrograms can be used as features to train this generative model. These are also known as sound representations.

Raw Audio

Raw audio consists of a wave form that can be fitted into a network which is popularly used to be the neural network, We then train the network with this pre-existing audio and then after on the other end of the network we get a sound which is fresh, new and synthetic. This is an end to end process which requires no preprocessing or transformation of the training data.

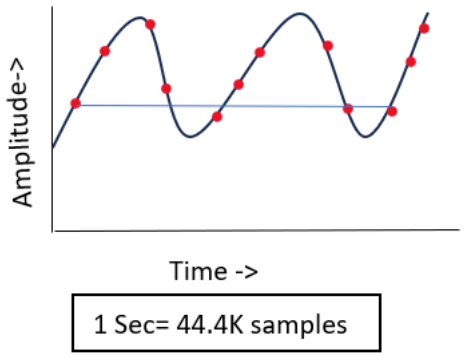


Figure 1: Soundwave with song data points(red dots)

The soundwave is a wave form that consist of an amplitude(Y-axis) that varies according to time(X-axis). It consists of various points known as the sample points along the line of the amplitude that can be referred to train models to generate new music. These are discrete points on the time series and one second of such a wave form is capable to produce 44.1K sample data. This is a very heavy process. Hence this tells us that our model should be able to handle the space and memory when we train it with raw audio.

Challenges of raw audio

a) Large, complex training data: Raw audio can make identifying long-range dependencies very tricky. These dependencies are the structures/patterns that we extract from the soundwave. These soundwaves can have various resolution dependencies like different pitch[1], melody, timbre, harmony or rhythm .So a very complex model is needed which is not feasible due to the amount of big data of the training data set. Huge amount of sample data points along the wave amplitude also causes lots of training data to the model.

b) Expensive computations: A model that uses raw audio should be suited to handle the massive training and hence computation on this data, memory, storage, processors speed makes it expensive.

c) Slow Generation: A lot of time due to training bigdata leads to a very slow generation of the music making it a lengthy and time-consuming process.

Solution: Using Spectrograms

To remove the challenges raw audio gives us we need to use a compact sound which are spectrograms. In spectrograms we use the waveform but transform it to include time(x-axis) and frequency(y-axis) domains only. A grid structure is formed. Any point that lies in this grid tells us how much energy the sound data point has at a specific frequency at a particular time. Hence a spectrogram can be used as a heat map.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

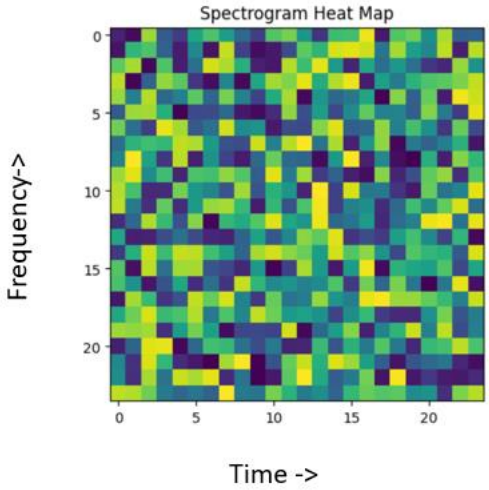


Figure 2: A spectrogram can be depicted by a heat map

The brighter the color in the heatmap the greater the energy used by that sound data point is. Hence spectrograms can be used to train the models and then create a generative sound model. While working with spectrograms we need to use preprocessing/transformations unlike raw audio.

The initial waveform(time/amplitude domain)is pre-processed and transformed by applying Short time fourier transform to it to get the spectrogram (time/frequency) domain.

Hence we train the model using the spectrogram instead of raw audio. Thus as the output we generate a spectrogram only rather than raw audio. Since humans cannot process this spectrogram audio we need to apply a reverse short time fourier transformation again to generate the output in the raw audio format which is human processible. Hence a waveform is again generated in the end.

Some various types of spectrograms[2] include:

a) Vanilla: This is the most common type of spectrogram that is mentioned above that creates a frequency-time grid heatmap and needs fourier transformation.

b) Log-amplitude: To this spectrogram we need to log transformations to the amplitude.

c) Mel spectrograms: This is also commonly used spectrograms as we can provide our models with sounds that can be perceived or identified by a human.

Therefore, a spectrogram has comparatively lot of advantages than raw-audio since it is compact, it requires less data points since the interval between two adjacent data point in the spectrogram is larger compared to that of a waveform and hence it can contain more information with less data points. This makes it a lighter, less complex, and less expensive process.

III. MODELING AND ANALYSIS

The technology stack used in this project was python as a language. For implementing deep learning technologies, we used TensorFlow and keras. For preprocessing of the dataset or applying short time/inverse fourier transformations we have used librosa which is famous for being known the audio processing library for python.

Some of the sound datasets that we have used in the project is

1) N-synth: This dataset is given by Google and it consists of a lot of musical instrumental notes.

2) MAESTRO: This dataset consists of more than 100 hours of professional piano tunes played by pianists.

Hence our model will make use of Mel Spectrograms and variational autoencoders with autonomous generation input as described in methodology above to produce music notes/music.

Autoencoders:

They are used for unsupervised machine learning and make use of unlabelled or unclassified data for training. Autoencoders make sense or automatically try and understand the data without needing of a prior information of the data feeded to it. They learn about patterns or structure in data (Representation learning).[3]

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

With autoencoders we aim to create a low dimensional representation of the original data since original waveform is huge to contain, preprocess and transform. We, make use of the bottle neck architecture/layer on the input layer to give the optimized usable outer layer.

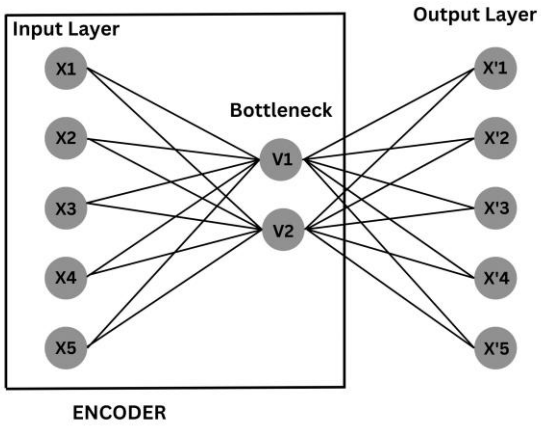


Figure 3: Encoding part (Encoder) of a Autoencoder Architecture

Autoencoder consists of encoder and decoder. The encoder is the first step before the bottleneck layer which compresses the original data into the lower dimensional representation. It encodes the data and reduces its complexity to include only the important features/attributes of the dataset and is often called as the latent space. Hence the data should be flexible to all dimensions.

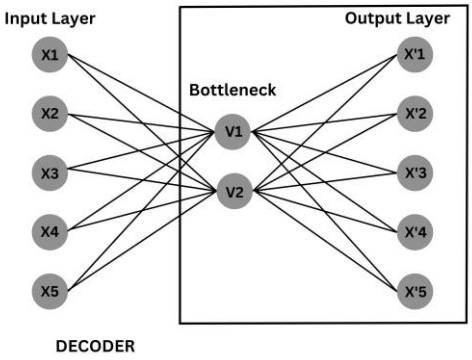


Figure 4: Decoding part (Decoder) of a Autoencoder Architecture

The decoder decompresses representation back to the original domain after processing to be able to give the output in the original resolution it came in.

We need to use the autoencoders to just reduce the complexity and the working/memory/process power of our model. We trained autoencoders (neural networks) using minimum reconstruction error to minimize the loss of data during resolution change. This, reconstruction error is the difference between the original data and reconstructed data.

The data should be sensitive enough to be able for the model to reconstruct it back and it should not be insensitive enough so that the model overfits it. Hence the autoencoders could have multiple layers before reaching the bottleneck layer to improve accuracy of the model.

Generating music using autoencoders:

1) The bottleneck should be a 2-dimensional empty space (latent space) at the start with X and Y columns.

2) Covert the raw audio to spectrogram using STFT. Feed in the spectrogram input to the encoders.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

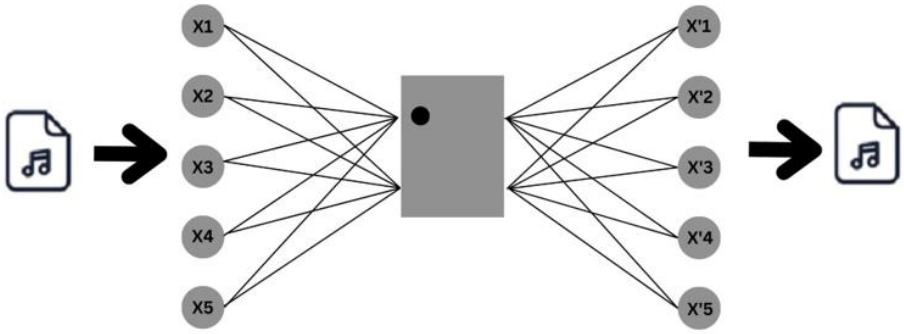


Figure 5: Spectrograms are plotted(treated) as dots in the latent space plane.

3) The autoencoder will encode the spectrogram as a point in the 2-dimensional xy plane. Multiple spectrograms will make up multiple data points in this plane.

4) The model can sample new points in this latent space using algorithms and create new data points in this latent space. This whole latent space data sets are feeded to the decoder.

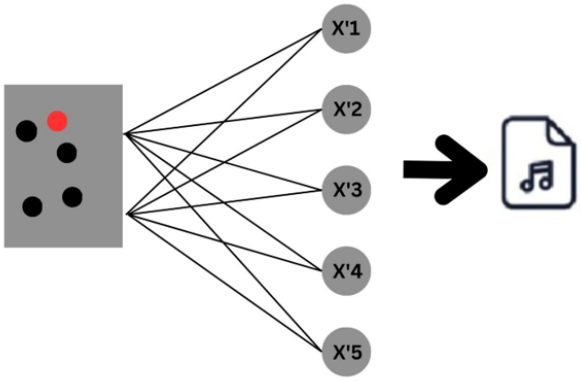
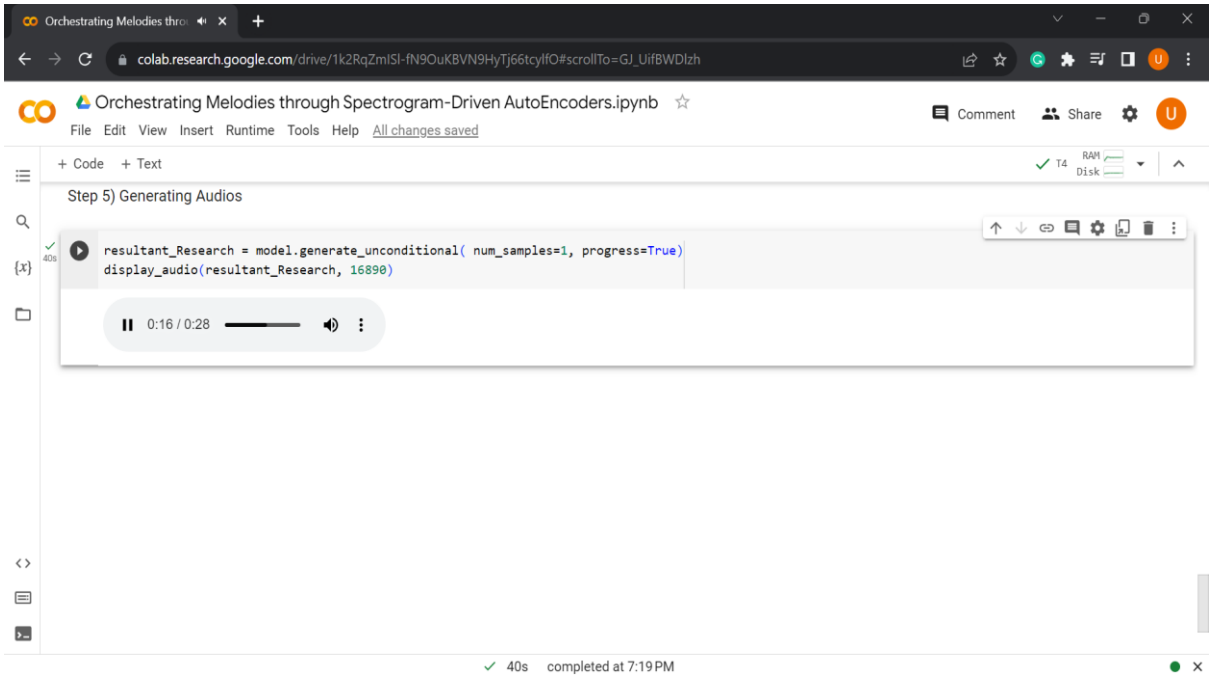


Figure 6: Model can sample new points in this latent space

4) The decoder will then be able to reconstruct this dimensionally reduced datapoints to the original spectrogram. It should be like the original input provided with minimum error.

5) Convert the output spectrogram by ISTFT process using the Griffin-Lim algorithm.

IV. RESULTS AND DISCUSSION



We make use of Google Colab (Collaboratory) which is a cloud platform because it provides free usage of processing units both graphics and tensor (GPUs/TPUs). It also allocates some RAM CPU memory which makes it easier for us to load our heavy data set that contains various sounds input files. We have made use of the Audiocraft and PyTorch. PyTorchis an open-source machine learning library for Python that provides dynamic computation graphs and tensor-based operations, facilitating efficient development and deployment of neural network models. AudioCraft is a PyTorch library designed for the purpose of facilitating deep learning research in the field of audio generation. This versatile library encompasses both inference and training components for two cutting-edge AI generative models, namely AudioGen and MusicGen. These models are renowned for their ability to produce audio of exceptional quality.

V. CONCLUSION

In this paper we have researched about the generation of music. We investigated various types of input to be given, which is the right data form that the model should be trained upon to get the right accurate input and the ways autoencoders can be used easily to generate music accurately.

This model was designed to be generate music and making the generation of music process to be simples, less computationally expensive or complex. The music being generated is clear with a good auto part and it eliminates the other noises. The music generated can be easily perceived by the human beings.

VI. REFERENCES

[1] Guo, R., Simpson, I., Magnusson, T., Kiefer, C., & Herremans, D. (2020). A variational autoencoder for music generation controlled by tonal tension. ArXiv. /abs/2010.06230

[2] Vyas, Tanishq. “A Glimpse of the Sound - Tanishq Vyas - Medium.” Medium, 28 July 2022, tanishqvyas069.medium.com/a-glimpse-of-the-sound-7c3ff5b16bf.

[3] Bank, D., Koenigstein, N., & Giryes, R. (2020). Autoencoders. ArXiv. /abs/2003.05991