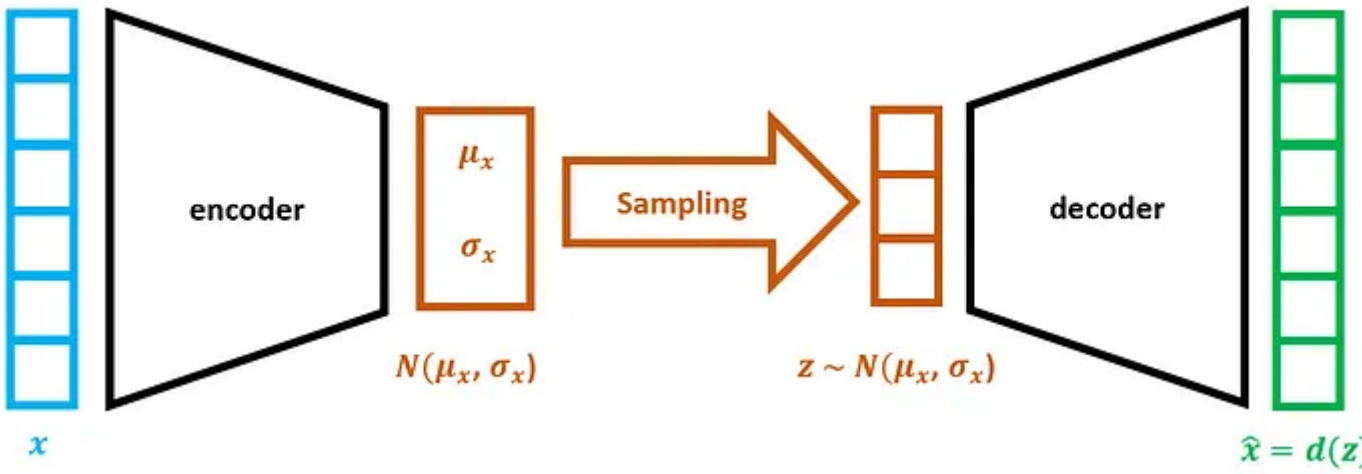
Deep Learning: Course Project

Himanshi (B20AI012)

Harsh Kumar (B20AI011)

Music Generation Using Variational Autoencoders



Objective

The objective of this project is to generate music using Variational Autoencoders (VAEs). The project utilizes the Lakh Pianoroll Dataset, which contains multitrack pianorolls derived from the Lakh MIDI Dataset. The goal is to demonstrate the ability of generative models to arrange music across different instruments and generate complex and rich music.

Introduction and Motivation

The project aims to leverage VAEs, which are autoencoders with a regularized training process, to ensure that the latent space has desirable properties for music generation. VAEs encode inputs into a distribution over latent space, allowing for continuity and completeness. Continuity ensures that similar points in the latent space produce similar outputs, while completeness ensures that sampled points from the latent space generate meaningful content.

The motivation behind using VAEs for music generation is to create a generative model that can produce new and diverse music while preserving the characteristics of the input music. By mapping the input music into latent distributions with VAEs, variation can be introduced by adding random noise to the encoded latent distribution's parameters. This process generates music outputs that are similar yet different from the original inputs, resulting in aesthetically pleasing variations.

Approach and Methodology

1. Dataset: The project utilizes the Lakh Pianoroll Dataset, specifically the LPD-5 version, which includes tracks for piano, drums, guitar, bass, and strings. The dataset consists of 21,245 MIDI files with corresponding metadata.
2. Variational Autoencoders (VAEs): VAEs are employed to encode the input music into latent distributions. The VAE architecture includes instrument-specific VAEs for piano, guitar, bass, strings, and drums. The VAEs encode the input music into a latent encoding of dimension K.
3. Latent Space Mapping: The encoded latent parameters from the piano VAE are passed through a Multi-Layer Perceptron (MLP) called MelodyNN. MelodyNN learns a mapping from the previous piano sequence's latent distribution to the next piano sequence's latent distribution. The output becomes the generated next piano output.
4. Conditional Neural Networks (CNNs): Instrument-specific VAEs are trained for the four non-piano instruments (guitar, bass, strings, drums). ConditionalNNs, which are MLPs, are trained to take the generated next-period piano latent parameters and the previous-period guitar latent parameters. These ConditionalNNs learn a mapping to generate the next-period guitar latent parameters, which are then decoded by the instrument-specific VAE's decoder to produce the next-period guitar output.These are two types of NN have been implemented :Harmony NN and Melody NN.

   The MelodyNN responsible for generating the next time-step melody and **the Conditional HarmonyNN** generating the non-piano instruments given the melody and the instrument's music

from the previous time-step. The architecture of both models includes convolutional layers, dense layers, and deconvolutional layers.

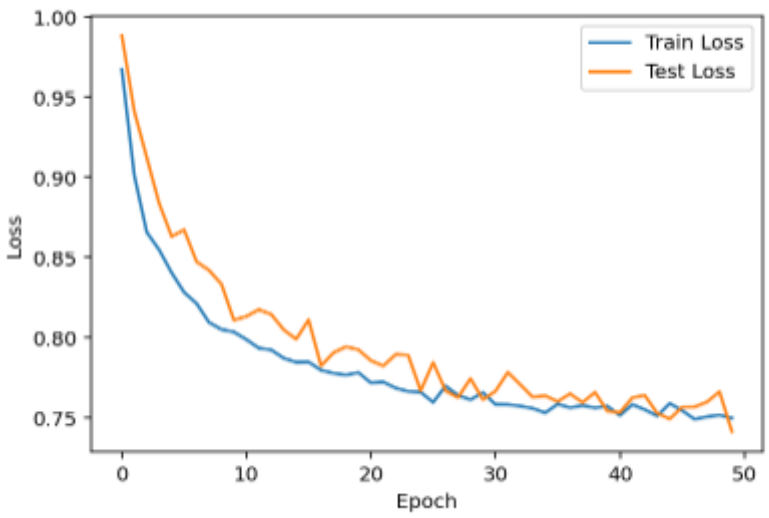
5. Latent Dimensionality and Noise: VAEs with different latent dimensionalities (8, 16, 32, and 64) are trained. A 16-dimensional latent space is used for training the conditional NNs since the music samples are relatively sparse in music space. Random noise with standard deviations between 0.5 and 1.0 is added to the encoded latent distribution's parameters to generate the desired amount of variation.

Results

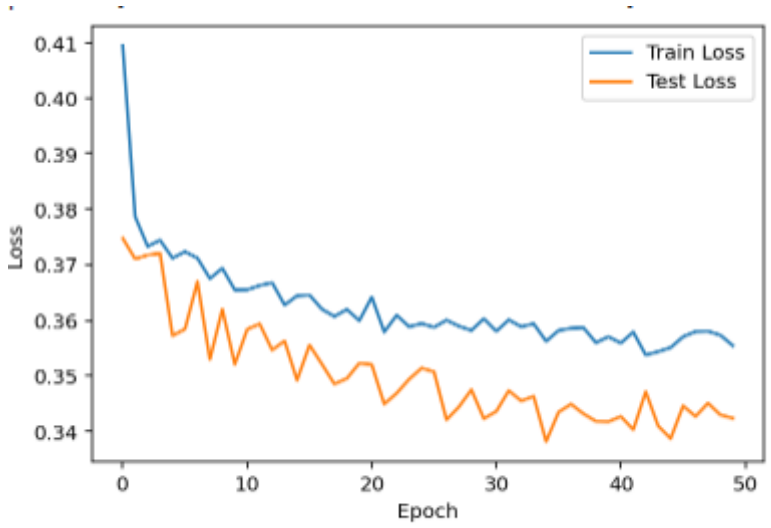
The VAE-NN approach proves successful in creating multi-instrument outputs that sound coherent and have appropriate amounts of variation. By adjusting the standard deviation of the added random noise, the generated music exhibits different levels of variation. Random noise with standard deviations between 0.5 and 1.0 is found to generate the best amount of variation.

Here you see the losss curve down below is for the melody neural networks.

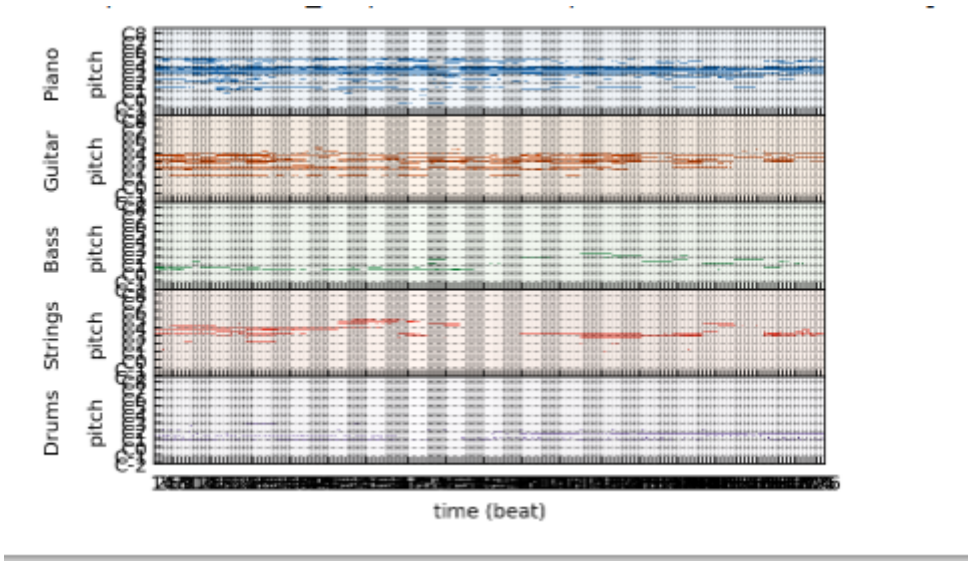
optimizer used : Adam and the Loss function : MSELoss



Here you see the loss curve below is for the conditional neural networks.



Here below you can see the final pitch vs beat graph for the outputs generated by the model.



Conclusion

In conclusion, the project of music generation using deep learning has been successfully completed. By implementing a two-part generation model with a HarmonyCNN and MelodyCNN,The architecture of the models included convolutional layers, dense layers, and deconvolutional layers, and they were able to generate high-quality music sequences with multiple instruments. The project has demonstrated the potential of deep learning techniques in generating music and can be further extended to explore more advanced models and techniques for music generation.

Code

https://github.com/harshrounder/Music-Generation-VAE

References

Producing a Chainsmokers Remix with A.I. | by Andrew Shaw | Towards Data Science

MidiNet: A Convolutional Generative Adversarial Network for...

Most existing neural network models for music generation use recurrent neural networks. However, the recent WaveNet model proposed by DeepMind shows that convolutional neural networks (CNNs) can...

https://arxiv.org/abs/1703.10847

https://github.com/ashishpatel26/Best-Audio-Classification-Resources-with-Deep-learning#dl4m-details