Exploring Generative music techniques, its applications and use case scenarios with RNN, Latent Space and Gan models

Claudiu Andrei

Queen Mary University of London

Abstract. This project aims at experimenting, studying, and analyzing various generative music techniques in order to learn and become familiar with this topic of the computational creativity field. In more detail, the architectures and methods analyzed include recurrent neural networks, how they are employed in generative music tasks and what advantages they bring for the generation of musical sequences. Latent spaces and their use in tasks where the creation of new music pallets is desired are explored. In generative audio research and creation of new and original sounding instruments is also subject of interest in generative music, this project aims at using a technology called GANSynth in order to produce individual instrument notes using GAN models.

1 Introduction

[1] Computationally automated generation of music is a concept which started being investigated and gaining interest from relevant communities in the late 70's (Hedges, 1978). Thanks to modern advancements in the field of artificial intelligence and widely spread use of deep learning techniques, state of the art systems are now produced which are able to generate personalized, expressive, inspiring and original music, defining not only technical achievements but also cultural relevance and value [2]. As this project is aimed at generative music techniques using deep learning it is important to comprehend its 5 main dimensions:

Objective: The objective of this coursework is to explore generative music techniques while experimenting with various architectural models when it comes to deep learning. The type of content in this project is mostly standalone pieces ranging on the spectrum from monophonic melodies with a single instrument to multivoice polyphony melodies containing multiple instruments playing potentially multiple notes at the same time. The context in which the generated artifacts are be performed is sequencer software, or a player that processes the music.

Representation: Data for this project is represented with audio files, midi and wav.

Architecture: As the aim of this project is to explore a variety of generative music techniques, a number of models and different architecture types are employed: mainly the objective is to produce artifacts with the use of three main network types; recurrent neural networks with LSTM based language for generating new music that aims at continuing sequences of notes; in addition, Generative Adversarial Networks used to learn to produce individual instrument notes. These are discussed in more detail in subsequent sections of this document.
Challenge: Some of the challenges and limitations of this explorative project are linked to the datasets used for the generation of audio artifacts. Evaluation metrics are a tricky part for this kind of project as it is not possible to obtain comparable results from each of the systems and it is therefore more favorable to not compare the outputs but rather discuss these without necessarily indicating Some limitations of the project include the extent to which tests can be carried out in different scenarios: because of time constrains it is not feasible to alter the structure of these architectures in detail as some of these are areas of expertise that can take years to become totally familiar with
Strategy: Given the still highly experimental state that generative music is in, it is good practice to maintain flexibility and experiment with different configuration option. Experimentation is also undertaken to improve the system and address to address higher level issues in computational creativity.

2 Background

The working style of RNNs is suited for this kind of task due to their nature of considering sequential information as input and providing information in sequential format rather than accepting a static input and providing the same stable output with each execution. The model used in this project applies language modelling to melody generation by using LSTM (long short-term memory). The aim here is to input a sequence of notes and obtain a continuation of the melody. Furthermore, in this project, the concept of palettes is explored: data is summarized by an encoder and a decoder is used to attempt to generate the original audio files. The system employed for this, Music VAE, can be configured to alter the number of steps of a sequence as well as its resemblance to the original excerpt and its randomicity. Generative adversarial networks are an approach to generative music modeling using deep learning techniques such as CNNs. The system utilized to experiment with this architecture is known as GANSynth, a model built using the Magenta library which is able to interpolate between random instruments or 2 chosen instruments in order to generate new sound flavor. The use of GANs for generative purposes is an interesting method to research as it allows for a clever way of training models by framing an unsupervised type task as a supervised problem divided in two sub-models, where the generator model creates new examples and the discriminator classifies these as either appertaining to the domain or generated.

3 System Description, Dataset and Data Representation

Data and datasets for this project differ based on the system, as there are multiple models being used here, the type of musical excerpt they work with as input and type of artifact generated as output differ slightly in each of the models explored. Firstly, the model Melody RNN[7], continues melodies by predicting next note given all the previous notes. Here the distribution of probable notes is sampled, and the chosen note is fed back into the model for the next step. The raw data for this model that is to be used as input is a collection of MIDI files, however these need to be converted into a format that Tensorflow can understand using the "SequenceGenerator" interface provided with MelodyRNN. The MelodyRNN system has predefined default configurations therefore making it easy to test and experiment with different models. For this project the configurations explored are Basic and Attention: this is going to provide the grounds for analyzing the effect of introducing attention to the model. Figure 1 below portrays the structure of a basic RNN.

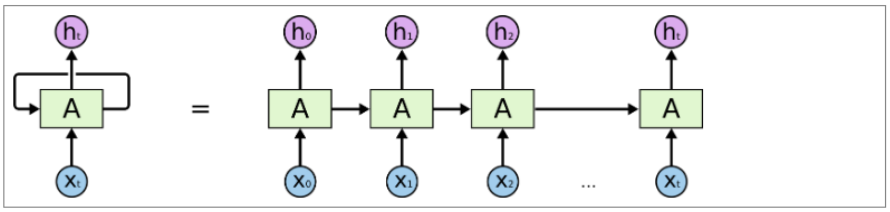
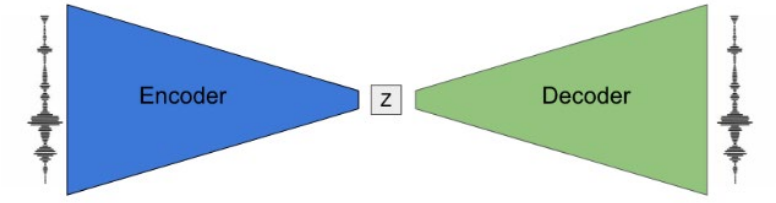


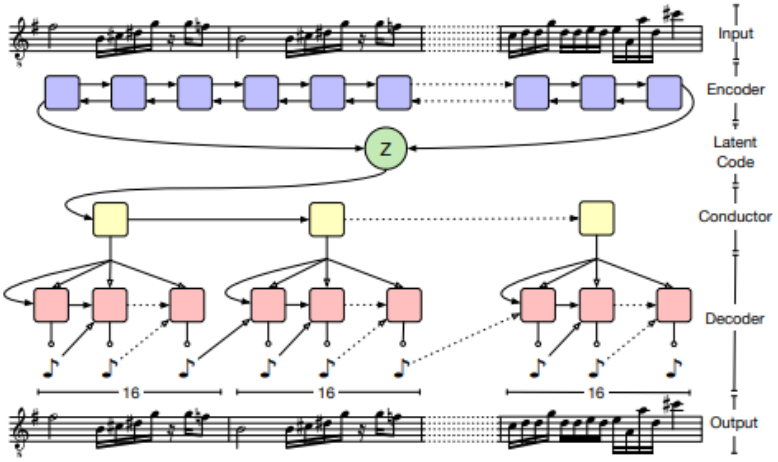
Figure 1

When experimenting with MusicVAE[https://arxiv.org/pdf/1803.05428.pdf] the concept of latent spaces is introduced, here the data used follows the same format as the previous System, Sequences converted from MIDI files. As previously mentioned MusicVAE is able to create music in different modes: Random Sampling from prior distributions and interpolation between two sequences therefore in this project both are being employed and tested. In order to learn latent representations, the system uses an AE (autoencoder) which is able to compress or encode samples into a vector of numbers which is passed through a bottleneck reducing the dimensionality of the vector, forcing the system to learn a compression scheme. The vector is then reproduced or decoded and during the process the qualities which appear to be more common are selected. Figure 2 below portrays an AE that has learned a latent space of

timbre in musical notes.



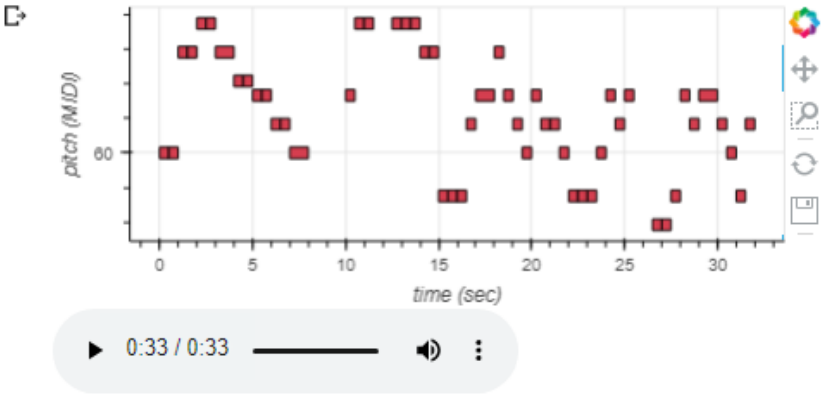
The reason why latent spaces need to be learned is so that artifacts can be generated that don't sound random: exploring melodies by enumerating all the different variations is not achievable and most of these will also not sound good. If looking for example at a 2-bar sequence in 4/4 time the number of possible combinations of events is 90^{32} (88 keys + release + rest)[9] which makes evident the need for latent spaces: if melodies are generated by only taking into account essential qualities, the number of "random" and unusable artifacts decreases greatly. Figure 3 below showcases the schematic of MusicVAE [5][6]



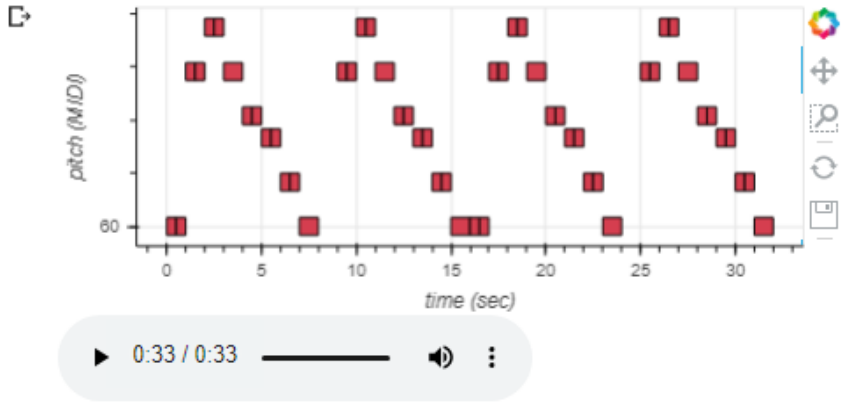
Generative Adversarial Networks have been successfully employed in image generation and are considered to be a state-of-the-art system for generating this type of artifacts. In this project a recently developed method is employed: Adversarial Neural Audio Synthesis, a model first revealed by the GoogleAI team in 2019 [3][4]. The progressive GAN architecture is used to up sample with convolution from a single layer into the full sound and an STFT is used to compare a frame of signal to many different frequencies.[9]

4 Experiments and Results

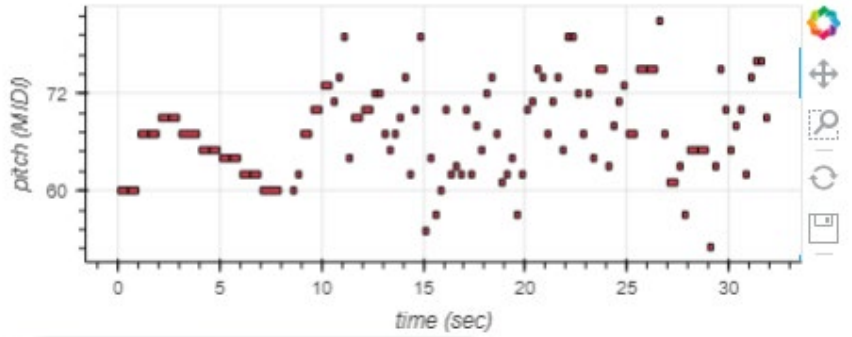
When experimenting with MelodyRNN two configurations are tested and below are examples of some outputs obtained.



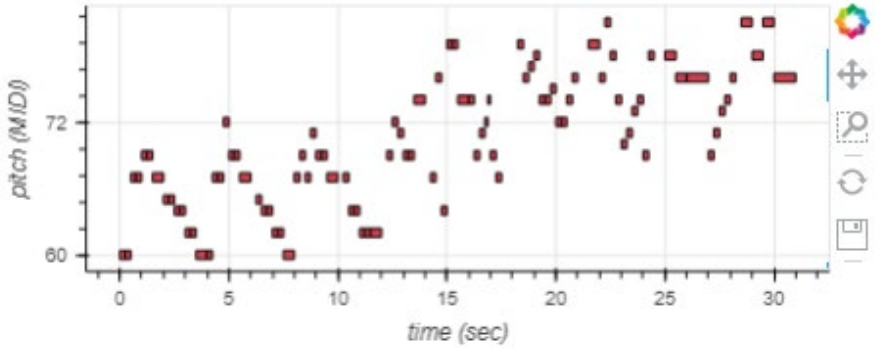
In the above image a sequence of the melody Twinkle Twinkle Little Star is generated to continue the original 9 sec file. This is the most basic type of RNN is the configuration and therefore not much regard is given to features of the song, making it sound very artificial. In order to improve on this aspect, the Attention_RNN has been tested afterwards and in this case the out obtained is displayed below:



Since the new model introduces the concept of attention it is very good at capturing features of the melody however the creativity aspect does not appear to be very present here since the attention loops the original melody over and over. In the next iteration the configuration is altered by modifying the temperature parameter in order to obtain a more varied result:



This time the system behaves more as expected and produces an interesting melody which doe does not sound out of place.
Next MusicVAE is tested:



Here two sequences are interpolated, and the output is an interesting melody that resembles both the Twinkle Twinkle melody and the Teapot melody. This model is good at retaining consistency on different outputs therefore they only sound slightly different on each run.

GANSynth:

download.wav

The GANSynth system here has created a new version of the input file by making changes to the timbre of each of the instruments and generating new sounds for each of them at every step of the midi file's execution. This cannot be displayed in this report however the file is attached above for the reader to analyze the output.

5 Discussion and Higher Level Computational Creativity Issue

The architectures tested have yielded interesting results and after curating the configuration of the models, outputs are obtained which allow the systems to be analyzed with more scrutiny. Performance of the systems varies depending on the architecture employed and furthermore the models used for this project are pretrained with powerful hardware at Google Data centers, making it complex to comment on the runtime of various models as the results cannot be replicated by everyone willing to test them because of time constraints as well as lack of adequate equipment. It can be commented however, that from the experiments carried out, the GAN system took the longer to execute, however it appears to be the most pleasing output amongst the systems; this is however subjective. It can me argued that the objective of these models do not align with the purposes of computational creativity's core meaning: generating new and original artifacts by using AI. If the output is in part conditioned by pieces composed by humans, it can be implied that new and original content is not being produced, instead the AI uses mathematical algorithms to alter the original input therefore not being exactly creative in the way a human would be.

6 Conclusions and Future Work

In conclusion it can be said that the aim of the project of exploring generative music techniques has been met, allowing the for the critical analysis and discussion of the different models employed. In future work it could be of interest to explore these models further and analyse their inner working in more detail, however this was not possible in this project as the Magenta library is a huge collection of generative music methods, offering almost countless options for customisation and different configurations. The models and architectures analysed can be used for future projects in the generative music field and this project provides a good foundation for future work on the topic by whoever finds it of interest. The creativity of the systems has been criticised, providing the grounds for addressing the higher-level issues in computational creativity more in depth, which was not possible in this project due to word limit and time constraints.

References

1. Meade, N., Barreyre, N., Lowe, S.C. and Oore, S., 2019. Exploring conditioning for generative music systems with human-interpretable controls. arXiv preprint arXiv:1907. 04352..
2. S. Colton, Lecture 17+18, Computational Creativity (ECS7022P), 2022 Generative Music
3. Engel, J. H.; Agrawal, K. K.; Chen, S.; Gulrajani, I.; Donahue, C. & Roberts, A. (2019), GANSynth: Adversarial Neural Audio Synthesis., in 'ICLR (Poster)', OpenReview.net, .
4. Karras, T.; Aila, T.; Laine, S. & Lehtinen, J. (2017), 'Progressive Growing of GANs for Improved Quality, Stability, and Variation'.
5. Roberts, A.; Engel, J.; Raffel, C.; Hawthorne, C. & Eck, D. (2018), 'A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music', cite arxiv:1803.05428.

6. Brock, A.; Donahue, J. & Simonyan, K. (2018), 'Large Scale GAN Training for High Fidelity Natural Image Synthesis.', CoRR abs/1809.11096 .
7. https://magenta.tensorflow.org/2016/06/10/recurrent-neural-network-generation-tutorial
8. https://magenta.tensorflow.org/gansynth
9. https://magenta.tensorflow.org/music-vae