# *Creating Music with Deep Learning*

*LSTMs vs GANs*

## BACKGROUND

In the past decade, generative models have been applied to a multitude of tasks and have found great success in tasks like text generation [1], speech synthesis [2], and raw audio generation [3]. While deep learning has only recently been used for artistic purposes, it has shown great potential, particularly in the areas of music and visual art generation. Still, art generation is still a relatively new application of deep learning, and a wide variety of models are actively being developed.

While many different methods for representing music have been proposed, we aim to use symbolic representations of music, such as those used in several public projects [4, 5], to train 2 types of generative models and compare their ability to generate realistic music. With a thorough comparison of 2 methods, we can better understand their strengths and weaknesses for the given problem formulation.

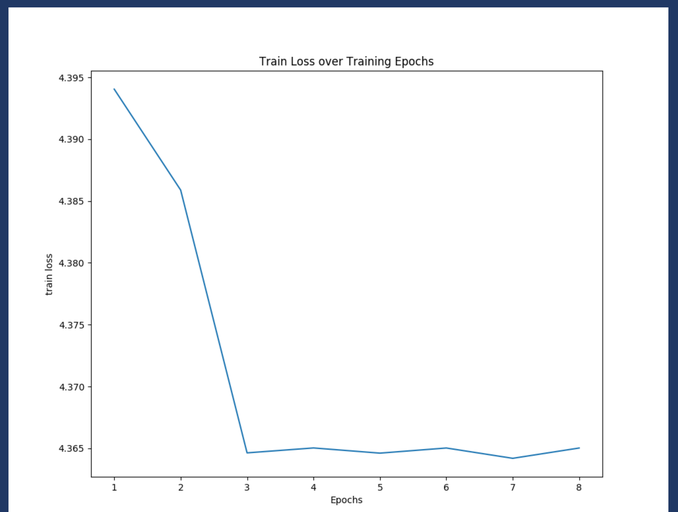
## METHODS

1. Collected examples of classical piano music in the form of MIDI files. Segmented into 100-note sequences.
2. Two models trained to generate music:
   1. LSTM – Train by predicting next note in sequence.
   2. GAN [6]
      – Discriminator = LSTM + FC layers
      – Generator = FC layers with seeded input
      Train both using the loss from the discriminator predicting real vs generated music.
3. Generate samples of new music from both and listen to the results to evaluate.

## RESULTS

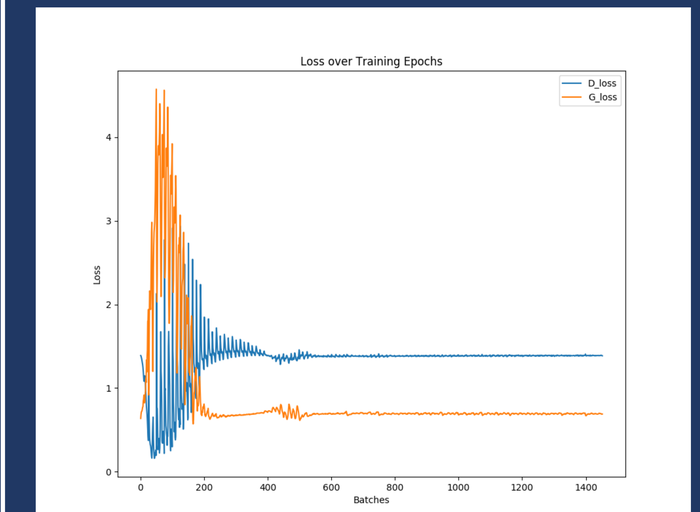
- LSTM outputs sounded dissonant, lacked any theme.
- GAN outputs sounded pretty good. Not many sub-themes, but clearly learned some repetitions and harmonic notes.

The **LSTM** was **unable** to learn to generate even basic music. Notes sounded totally dissonant and random. From the figure below, it is clear it didn't learn much as it was trained.




**The loss over training epochs barely decreased for the LSTM model, despite several different optimizers and learning rates attempted. Future work should investigate this to create a better setup for training. To start with, the model should be simplified, possibly by reducing the size of the fully connected layers or the hidden state vector, to see if it can learn anything when there are fewer parameters.**

The **GAN** was **able** to generate music that sounded passable to a human listener. A simple strategy it used was repeating a single note as a baseline and adding other relatively harmonic notes at semi-regular intervals.




**Initially, the discriminator is stronger than the generator, so its loss decreases while that of the generator increases. At some point, the generator starts doing better and so this pattern reverses. Ultimately, the 2 losses stabilize, and they train against each other for a while. Music produced in the early stages sounded like a series of completely random notes, while music at the end had steady notes and some harmony.**

## AMMO BAR

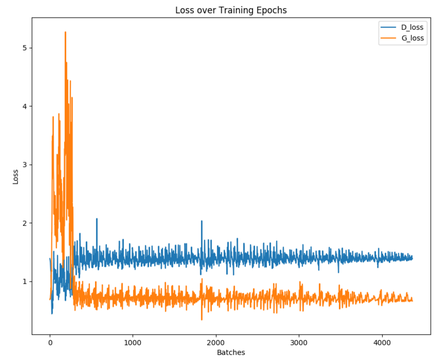



Figure illustrating training loss when training on both piano and drum music.


**References**

[1] Hu, Zhiting, et al. "Toward controlled generation of text." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

[2] Oord, Aaron van den, et al. "Parallel wavenet: Fast high-fidelity speech synthesis." *arXiv preprint arXiv:1711.10433* (2017).

[3] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499* (2016).

[4]https://github.com/RichardYang40148/MidiNet/tree/master/v1

[5]https://github.com/Skuldur/Classical-Piano-Composer

[6] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems*. 2014.



Gavin Mischler, Alex Chang, Darian Low, Matthew Ost

JOHNS HOPKINS UNIVERSITY