Instrumental audio synthesis using GANs Progress Report

Group N
Etienne Leclerc (V00853992), Jordie Shier (V00688891), Lu Lu (V00836042),
Yangruirui Wang (V00949204), and Ziyi Feng (V00940985)

1 Problem definition

As described in our formal proposal, the goal of our project will be to implement a GAN, similar to the one proposed by Donahue et al. (2018), and train it using audio samples recorded from instrumental sounds including brass, string,reed, and mallet instruments. The project will be useful to music producers and sound effect artists, who otherwise require the use of gigantic audio libraries yet too-limited collections of samples

This problem has not altered so much as crystallized and been made more specific. We will be basing our approach on that undertaken by Donahue et al. (2018). In their paper, they describe two algorithms based on Deep Convolutional Generative Adversarial Networks (DC GAN) Gao et al. (2018): SpecGAN, which applies image transformation to spectrograms; and WaveGAN, which uses the one-dimensional structure of an audio time-series ~~more~~ directly.

We have decided to investigate the latter approach, and use Donahue et al. (2018)as a springboard from which to implement WaveGAN-like features. The code for WaveGAN is publicly available on Github; however, for our own intellectual edification we will be pursuing a parallel course, and try to independently derive our own results. In addition, we would like to implement some of our own ideas; see the following section.

If time permits, we will also attempt a stretch goal of interpolation between instrument types. Such functionality would be extremely useful to music producers, who would now have access to a dizzying array of hybrid instruments.

We will use the freely available NSynth dataset, published by Engel et al. (2018) from Googles Magenta research lab. The NSynth dataset contains over 300k four second long audio samples of labelled musical instruments. We have not yet properly explored this resource, having relied mainly on a smaller library of snare drum sounds.

2 Goals

As described in the first section, our goal in this project is to use Generative Adversarial Networks (GANs) to generate new instrumental audio from the NSynth dataset. There are two main components in a GAN: the generator network that is tasked with generating new material, and a discriminator network that is tasked with classifying input data as real or fake. The generator and discriminator are trained in parallel with the goal of creating a generator that can produce realistic audio material, which is judged by the discriminator. In our project we will train the generator and discriminator using short audio samples of solo instrumental audio with the goal of producing a generator that can create new instrumental sounds.

From our research so far, we have learned GANs are challenging to evaluate using objective measurements, although some metrics have been proposed. Donahue et al. (2018) used two objective measures:*inception score* and *nearest neighbour* comparisons and we will use both of those measurements. Additionally, the Donahue paper attempts a *Feline Turing Test*, wherein the authors cats were presented with generated bird sounds over the course of the project; the cats level of alertness increased as the quality of the samples improved. We will instead use some of our cattier friends and family. As we have completed the initial research phase and have begun development, we have been able to make our goals more specific. The specific outcome that we would like to achieve is the development of a GAN that is able to produce instrumental audio samples that are one second long at a sampling rate of 16kHz. We have successfully implemented the original DCGAN and modified it to handle one-dimensional vectors. Another specific goal is to implement and test several advancements to DCGAN proposed by Donahue et al. (2018). These advancements are:

- **Inception Score:** a measurement derived from a pre-trained Inception classifier [insert reference] which can be used to quantitatively measure performance as well as to inform early stopping during training.
- **Phase shuffling:** This is implemented to combat a known effect of GANs to produce artifacts in images, which translates to harmonic distortion in audio signals. Phase shuffling makes the discriminators job more challenging so it can be seen as a type of regularization or penalty during training.
- **Loss Function:** Improved loss function using the Wasserstein distance, as in Arjovsky et al. (2018) or Gulrajani et al. (2017)

Additionally, we have developed an idea on our own for a variation on the WaveGan model that uses ~~the~~ Mel-Frequency Cepstral Coefficients (MFCCs). This reframes the audio generation problem back into the image generation domain by using a time series of mel-scale frequency bands to create a spectrogram image. Donahue et al. experimented with a similar approach using the Short-time Fourier Transform (STFT). The STFT is linear frequency audio representation, however humans perceive frequency in a logarithmic scale. MFCCs represent audio in the frequency domain using the mel-frequency scale, which is a logarithmic based scale based on human auditory perception. While MFCCs are typically not invertible, there exist approximations allowing for transform to the time-domain. The librosa library contains such an approximation and we would like to experiment with this and compare results to the time-domain based GAN that we are also developing.

3 Plan and progress-to-date

In our project proposal we outlined six project stages: research, implementation, objective evaluation, informal subjective evaluation, stretch goal, and the final report. We have completed the research stage and have begun implementation. In the research stage, we reviewed relevant papers, textbooks, and online tutorials which informed how we started development. Now we are in the second stage, which is the implementation stage. This implementation stage will last for about two weeks, with a goal of finishing on July 20th. To begin our implementation, we started with the original DCGAN that was ~~originally~~ developed for image generation. We modified DCGAN to handle audio signals and trained it on a set of snare drum samples as a test. More details on this initial experiment is provided in the following section and a reporting of the initial results is shown in the last part of this report.

For the following plans, as we mentioned in the proposal, we would like to evaluate the model using objective and subjective methods. If we still have extra time, we will attempt our stretch goal of interpolation between different sounds. In order to track the progress of the project, we hold a Zoom meeting every Friday to share the information we have and discuss our next step. We also share code on our github repository, and use Slack to regularly converse on the project. *Dates:* June 12 - July 10

~~So far, we have performed initial research and run an initial test using a modification on the original DCGAN to produce snare drum samples.~~

Since GANs are a new topic to us, we spent a lot of time doing the initial research and readings. Since we would like every member to gain information and experience from the research stage, in the Friday Zoom meeting, we are sharing information, which we conduct independent research that might be useful to our projects. Additionally, Jordies supervisor George Tzanetakis is a valuable resource for questions regarding audio processing. The following list is the resources we have gone through which have been helpful for our project:

- Online tutorial on audio synthesis with GANs (Pasini (2019))
- Related papers on audio synthesis using GANs (Donahue et al. (2018) Engel et al. (2018))
- Deep learning textbook (LeCun et al. (2015))

- Machine Learning Mastery (this tutorial on GAN latent space interpolation could be helpful for the stretch goal of interpolating between sounds (Brownlee (2019))
- Original GAN paper (Goodfellow et al. (2014))
- Tutorial on GANs (Goodfellow (2016))

For the implementation stage we decided to start by implementing the original DCGAN first, which is what Donahue et al. based their model off of. A tutorial in the TensorFlow documentation [footnote to https://www.tensorflow.org/tutorials/generative/dcgan] provided direction for developing this initial model. We modified the DCGAN from this tutorial to handle one dimension audio signals as opposed to images and added additional convolutional layers so the output layer was a size of 16384, corresponding to one second of audio at a sampling rate of 1kHz. We then trained this model on a dataset of about 2200 snare drum samples that Jordie had used for previous research and had readily available. The model was trained over 50 epochs, which took about 2.5 hours on a MacBook Pro

3.1 FUTURE STAGE: IMPLEMENTATION AND EVALUATION

Dates: July 11 - August 1

Our future plan includes the remainder of the implementation stage, evaluation, stretch goals, and our final report. Implementation will last until July 20th, evaluation and stretch goals will last until 27th, and we will then put together our final report. We will briefly outline our plans for each of the stages here. For the next step of the implementation stage, we would like to select and prepare a subset of the NSynth dataset to train our two proposed GANs: the time domain GAN with improvements based on the WaveGan by Donahue et al., and our proposed MFCC-GAN. The specific improvements mentioned by Donahue et al. that we will be implementing during this stage are: phase shuffling in the discriminator (to eliminate periodic artifacts in the audio); inception scores for evaluation and early stopping; and an improved loss function using the Wasserstein distance. To evaluate our models, we will use the inception score, nearest neighbour metric, and set up a Turing test to distribute among friends and family.

Interpolation is a stretch goal that we would like to implement if we have time. The purpose is to provide a way to interpolate between different sound types. This idea is inspired by image GANs which are able to smoothly interpolate between different faces. In audio processing, being able to smoothly adjust and move between different sounds would be a useful feature for music producers and sound effect designers. There is a tutorial by Brownlee (2019) that might be helpful for this goal. After we wrap up the project and document results, we will be able to finish the final project report

4 TASK BREAKDOWN

Tasks that need to be done:

- Selecting and preparing a subset of the NSynth dataset
- Wasserstein metric
- Inception classifier and inception score
- Setting up a Turing test to distribute
- MFCC GAN

Task breakdown:

- Jordie: project coordinator, the guy with the whip - implementing the MFCC GAN
- Etienne: Wasserstein metric
- Ziyi: Setting up a Turing test and Phase shuffling
- Yangruirui: Nearest neighbour metric
- Lu: Selecting and preparing a subset of the NSynth dataset and Phase shuffling

5 Initial Result

So far we have implemented a baseline version of our first model, a modified version of DCGAN to handle time-domain audio signals, and trained it on a dataset of 2200 snare drum samples. We have yet to implement the inception score and nearest neighbour metrics to objectively evaluate these results, so we have performed an informal listening evaluation and produced some plots that demonstrate training progress.

Listening to the results of this baseline model indicate that the GAN is learning to create snare-drum-like sounds. The amplitude envelope is being matched quite accurately, however there is a distinct harmonic noise present that is unnatural sounding. Furthermore, any tonal quality that is present in some snare drums is not being captured; the GAN is basically producing shaped noise. The phase shuffling that was proposed by Donahue et al. will hopefully help to address the harmonic noise in the signal.

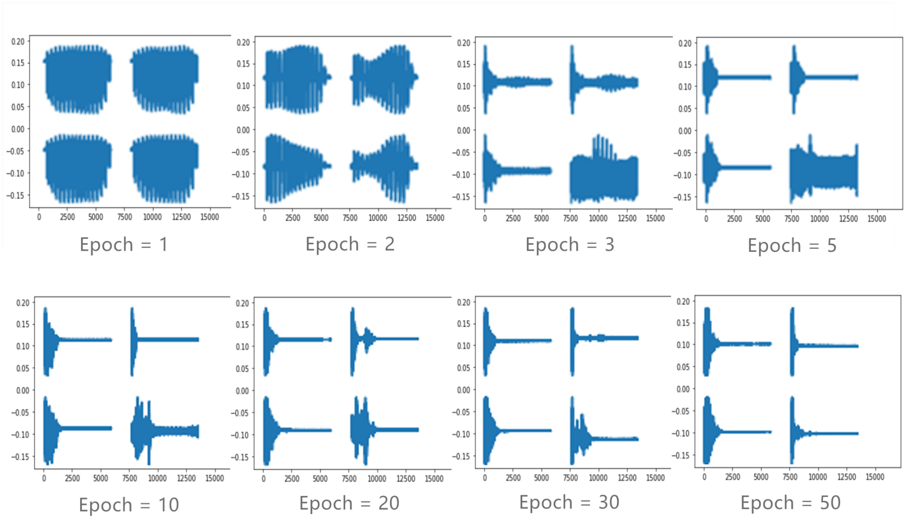

Figure 1: Audio Waveform Images on the GAN learning the envelope of the sound

Figure 1 shows the audio waveforms that the generator produced as the epoch increased from 1 to 50. When epoch is very small, the waveforms look like the noise wave, as it increases, the GAN gradually learns the envelope of the sound.

To gain insight into how sounds generated by the trained model compare to the pre-trained model as well as the training dataset, we plotted a selection of samples in two dimensions based on sound similarity. To generate this plot, 100 samples were randomly selected from the training set and 100 samples were generated from the pre-trained model and the trained model. To all three classes of audio files, we applied MFCCs (described above). For each frequency band, the mean and standard deviation was calculated. Since the results of MFCC are a 40-feature vector for each audio signal, we decided to use principal component analysis (PCA) to do dimension reduction. We reduce the dimensionality from 40 to 2, so that they can be visualized in a 2D figure. The visualization is shown below.

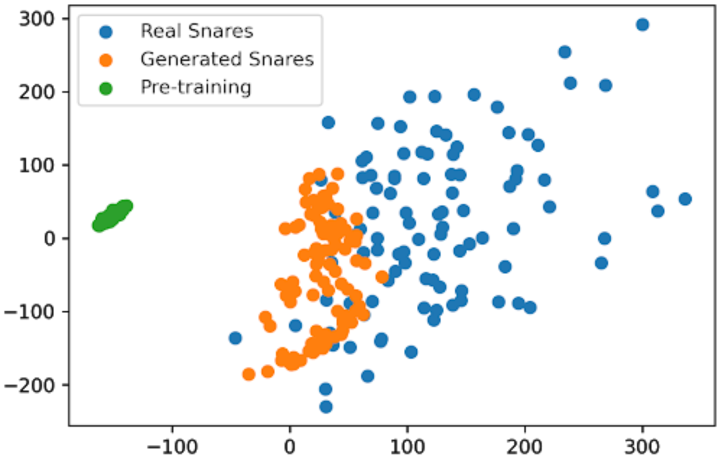

Figure 2: 2D plot of pre-training audio, generated snares audio and real snare audio

In figure2, the X-axis is the first principal component and Y-axis is the second principal component. This plot shows how all the sounds from the pre-trained model are clumped together, and then after training have expanded towards the sonic distribution of the training snare drums.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2018.

Jason Brownlee. How to explore the gan latent space when generating faces, 2019. URL `https://machinelearningmastery.com/how-to-interpolate-and-perform-vector-arithmetic-with-faces-using-a-generative-adversarial-network/`.

Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2018.

Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. In *International Conference on Learning Representations*, 2018.

Fei Gao, Yue Yang, Jun Wang, Jinping Sun, Erfu Yang, and Huiyu Zhou. A deep convolutional generative adversarial networks (dcgans)-based semi-supervised method for object recognition in synthetic aperture radar (sar) images. *Remote Sensing*, 2018.

Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Marco Pasini. Synthesizing audio with generative adversarial networks, 2019. URL `https://towardsdatascience.com/synthesizing-audio-with-generative-adversarial-networks-8e0308184edd`.