Audio Generative Adversarial Network

Chibuzor John Amadi[502623] & Kristian Perriu[505571]

[1] University of Pavia, Pavia
[2] University of Milano-Biccoca, Milan
[3] University of Milano-Statale, Milan
chibuzor.amadi01@universitadipavia.it,
kristian.perriu01@universitadipavia.it

Abstract. Generative Adversarial Networks (GANs) have gained significant attention in deep learning for their ability to generate realistic data samples. While traditionally applied to image generation, GANs have recently shown promising results in audio generation as well. This project focuses on applying GANs to the task of audio synthesis using a dataset of musical instrument sounds from FSDKaggle2018. We explore two architectures: WaveGAN, which operates directly on raw waveforms, and SpecGAN, which generates spectrogram representations of audio.The evaluation of the generated audio is conducted using both quantitative metrics, such as Mean Squared Error (MSE), and qualitative assessments, which is human judgment on the audio quality and realism.

1 Introduction

Applications for audio synthesis are numerous and include sound effects creation, speech synthesis, and music production. In order to produce high-quality audio, traditional approaches for audio synthesis frequently require manual labor, substantial domain knowledge, and intricate feature engineering. These techniques can take a lot of time, and they might not translate well to other kinds of audio data.

In recent years, deep learning techniques have revolutionized various fields by enabling automated feature extraction and generation capabilities. Among these techniques, Generative Adversarial Networks (GANs) have shown remarkable success in generating realistic data samples, particularly in the domain of image generation. GANs consist of two neural networks, a generator and a discriminator, that are trained simultaneously in a competitive framework. The generator aims to create realistic data samples, while the discriminator attempts to distinguish between real and generated samples. This adversarial process drives the generator to produce increasingly realistic outputs.

Inspired by the success of GANs in image generation, researchers have begun to explore their application in audio synthesis. GANs can potentially automate the process of audio generation, reducing the need for manual feature engineering and enabling the creation of high-quality audio samples. This project focuses on

applying GANs to the task of audio synthesis using the FSDKaggle2018 dataset, which includes various musical instrument sounds.

We investigate two distinct approaches for audio generation:

- **WaveGAN**: This approach operates directly on raw waveforms, leveraging the temporal dependencies inherent in audio signals.
- **SpecGAN**: This approach generates spectrograms, which are time-frequency representations of audio signals. SpecGAN employs the Griffin-Lim algorithm to convert the generated spectrograms back to audio waveforms.

These approaches will be expanded on in the following section. Our models are trained using a standard GAN training strategy, where the generator and discriminator are optimized using binary cross-entropy loss. The performance of the models is evaluated using both quantitative metrics, such as Mean Squared Error (MSE), and qualitative assessments, including human judgment on the audio quality and realism.

2 Model Architecture

Our model architecture consists of two main components: WaveGAN and SpecGAN. These architectures are adapted for audio generation tasks, leveraging different approaches to handle the unique characteristics of audio data.

2.1 WaveGAN

The WaveGAN architecture modifies the traditional Deep Convolutional GAN (DCGAN) to operate on one-dimensional waveforms instead of two-dimensional images. This involves using longer one-dimensional filters and removing batch normalization to stabilize training. In our implementation, we use a series of transposed convolutional layers followed by linear layers to generate audio waveforms from random noise inputs.

The generator network in WaveGAN starts with an initial transposed convolutional layer that takes in the noise vector and transforms it into a higher-dimensional feature map. This is followed by batch normalization and a LeakyReLU activation function to introduce non-linearity and improve the training dynamics. A dropout layer is added to prevent overfitting. The subsequent transposed convolutional layers continue to upscale the data, progressively increasing the temporal resolution of the feature maps. Finally, a series of linear layers further processes the upscaled data to match the desired audio shape.

The discriminator network in WaveGAN is designed to distinguish between real and generated audio waveforms. It consists of multiple linear layers with ReLU activation functions and dropout layers for regularization. The initial linear layer takes the input audio waveform and transforms it into a high-dimensional representation. Subsequent layers progressively reduce the dimensionality while maintaining important features necessary for discrimination. The final layer outputs a single value representing the probability that the input audio is real.

2.2 SpecGAN

SpecGAN generates spectrograms, which are two-dimensional time-frequency representations of audio signals. These spectrograms are then converted back to audio waveforms using the Griffin-Lim algorithm. This approach leverages the capabilities of GANs designed for image generation, treating spectrograms as images and utilizing convolutional operations to capture the time-frequency patterns.

The generator network in SpecGAN uses transposed convolutional layers to generate spectrograms from noise inputs. These layers progressively increase the spatial resolution of the feature maps, effectively generating detailed time-frequency representations. The generated spectrograms are then processed using the Griffin-Lim algorithm to reconstruct the audio waveforms. This algorithm iteratively estimates the phase information required to invert the spectrograms back to the time domain.

The discriminator network in SpecGAN is similar in structure to the WaveGAN discriminator but is adapted to operate on spectrograms. It uses convolutional layers to analyze the time-frequency patterns and determine the authenticity of the input spectrograms. By treating spectrograms as images, the discriminator can effectively learn to distinguish between real and generated spectrograms, thus guiding the generator to produce more realistic audio representations.

In our GAN models, the generator and discriminator networks are optimized using binary cross-entropy loss. The generator aims to produce samples that can fool the discriminator, while the discriminator is trained to accurately distinguish between real and generated samples. The models are trained using the Adam optimizer, with carefully chosen learning rates and beta values to ensure stable and effective training.

Overall, our approach demonstrates the potential of GANs for high-quality audio synthesis by leveraging both waveform and spectrogram representations. Each architecture exhibits unique strengths, contributing to the generation of realistic and high-fidelity audio samples.

3 Methodology

3.1 Data Collection and Preprocessing

The dataset used is FSDKaggle2018, which contains audio recordings of various musical instruments. The data is preprocessed into both waveform and spectrogram representations.

Data Download and Unpacking The dataset is downloaded from the Zenodo repository and unpacked. It includes a variety of musical instruments, allowing us to train the model on diverse audio samples.

Setting Device The models are trained using a GPU or MPS for efficient computation, ensuring that the training process is fast and can handle the large dataset effectively.

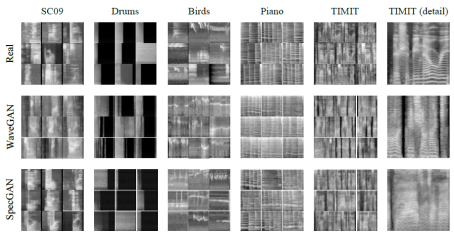

Fig. 1. Top: Random samples from each of the five datasets used in this study, illustrating the wide variety of spectral characteristics. **Middle:** Random samples generated byWaveGAN for each domain. WaveGAN operates in the time domain but results are displayed here in the frequency domain for visual comparison. **Bottom:** Random samples generated by SpecGAN for each domain.

Preprocessing Steps The raw audio data is converted to Mel Spectrograms using the Fast Fourier Transform (FFT). The spectrograms are normalized to ensure stable training. This process captures the essential features of the audio signals while reducing noise and irrelevant information.

3.2 Training Setup

The GAN models are trained on the preprocessed data using PyTorch. The training process involves alternating updates to the generator and discriminator to minimize the adversarial loss. The generator learns to create realistic audio samples, while the discriminator learns to distinguish between real and generated samples.

During the training process, the loss values for both the discriminator and generator are tracked. For example, at Epoch [8001/10000], the discriminator loss (d_loss) was 0.0049, and the generator loss (g_loss) was 4.604. These metrics help in monitoring the training progress and ensuring that the model converges appropriately.

3.3 Evaluation Metrics

The performance of the models is evaluated using both quantitative and qualitative metrics:

- Mean Squared Error (MSE) between the generated and real audio samples.
- Inception Score adapted for audio data.
- Human judgment on the quality and realism of the generated audio.

4 Results

4.1 Quantitative Results

The models' performance is evaluated on a test set, with MSE and Inception Scores calculated for both waveform and spectrogram outputs. These metrics

provide an objective measure of how closely the generated audio matches the real audio.

4.2 Visual Representations

Spectrograms of generated and real audio are compared visually to assess the fidelity of the generated samples. This helps in understanding how well the models capture the time-frequency characteristics of the audio signals.

4.3 Qualitative Analysis

Human evaluators rate the audio samples on quality, intelligibility, and resemblance to real instrument sounds. This subjective evaluation provides insights into the practical usability of the generated audio.

5 Discussion

The results demonstrate the potential of GANs for audio generation. WaveGAN and SpecGAN each have strengths and weaknesses, with WaveGAN producing better time-domain features and SpecGAN excelling in frequency-domain accuracy. Future work could involve hybrid models combining the strengths of both approaches.

6 Conclusion

This project successfully applies GANs to the challenging task of audio generation. The models can generate realistic audio samples that are qualitatively similar to real instrument sounds. Future improvements could include more advanced architectures and training techniques to further enhance the quality and diversity of generated audio.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. In Advances in Neural Information Processing Systems (pp. 2672-2680).
2. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434.
3. Donahue, C., McAuley, J., & Puckette, M. (2018). WaveGAN: Adversarial Generation of Raw Audio. arXiv preprint arXiv:1802.04208.
4. Griffin, D., & Lim, J. (1984). Signal estimation from modified short-time Fourier transform. IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(2), 236-243.

5. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of Wasserstein GANs. In Advances in Neural Information Processing Systems (pp. 5767-5777).
6. Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.
7. Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition. arXiv preprint arXiv:1804.03209.