Exploring the Impact of Model Pruning on GANs

*Akshay Vijay Khanna
Dept. of ECE
NC State University
Raleigh, NC, US
akhanna3@ncsu.edu

*Chinmaya Srivatsa
Dept. of ECE
NC State University
Raleigh, NC, US
csrivat@ncsu.edu

I. Abstract

The goal of this project is to determine how model pruning affects image generation using Deep Convolutional Generative Adversarial Networks (DCGANs). Although DCGANs have transformed AI by creating realistic images, their computational demands remain a challenge. Reducing non-essential weights in the model through model pruning attempts to make these models less resource-intensive, which is important for deployment in limited environments. In this work, we investigate tailored DCGAN architectures using DCGAN models trained on CIFAR-10 and Fashion-MNIST. We intend to use percentage and sensitivity pruning. The effectiveness of pruning is evaluated using metrics such as model sparsity and the quality of generated images are evaluated using UQI, SSIM and MSSSIM. We aim to demonstrate that pruning can significantly reduce computational demands while maintaining acceptable performance levels.

II. Introduction

In recent years, Generative Adversarial Networks (GANs) have emerged as a transformative force in Artificial Intelligence (AI), particularly in the realm of image generation. Among these, Deep Convolutional GANs (DCGANs) have garnered significant attention for their prowess in creating high-fidelity images, fueling advancements in computer vision, content creation, and data synthesis. However, their computational demands pose significant hurdles for deployment in resource-constrained environments. This paper delves into the pivotal realm of model pruning within the domain of DCGANs, aiming to reconcile their exceptional image generation capabilities with the imperative need for computational efficiency. Model pruning, a technique focused on reducing network size by eliminating redundant or non-critical elements, stands as a promising solution to alleviate the computational burden without compromising performance.

The primary objective of this study is to comprehensively investigate the effects of various pruning methodologies on DCGANs. Through experimentation, we aim to discern the delicate balance between model size and image generation quality. While adhering to the foundational architecture of DCGANs, our exploration will venture into customized variations to assess the adaptability of pruning techniques within these structures. The choice of dataset plays a pivotal role in our investigation. While the CIFAR-10 dataset serves as our primary training ground, the flexibility to switch to the Fashion-MNIST dataset remains a contingency plan, contingent upon the availability of GPU resources. We plan to employ weight pruning techniques such as sensitivity and percentage pruning.

Our evaluation methodology encompasses a multifaceted approach, leveraging established metrics like UQI, SSIM and MSSSIM for image quality evaluation and model sparsity measurements for model size evaluation. The ultimate goal is to demonstrate that judiciously pruned DCGANs maintain satisfactory image generation prowess while significantly mitigating computational demands, thereby enhancing their viability for deployment in resource-constrained settings

III. Methodology

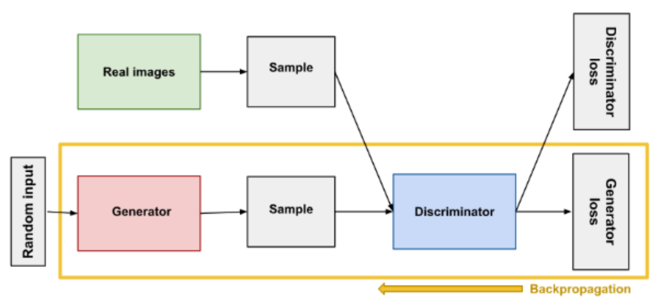

Fig. 1. High Level Architecture

A. Generative Adversarial Networks

Generative Adversarial Networks (GANs) mark a watershed moment in artificial intelligence, reshaping data creation and comprehension. This paradigm-shifting framework comprises two pivotal components—the generator and discriminator—collaborating in an adversarial dance. GANs have revolutionized fields like computer vision, content creation, and data synthesis, showcasing their unique ability to craft high-fidelity data, fundamentally altering the AI landscape.

At the core of GANs lies a dynamic interplay between the generator and discriminator. The generator, trained on a dataset, fabricates synthetic data mirroring real examples. Concurrently, the discriminator assesses this generated data, distinguishing between authentic and counterfeit samples, and offering feedback to refine the generator's output. This adversarial training fosters a competitive learning dynamic where both networks continuously improve each other. The generator evolves to create more authentic data as it refines its understanding of the dataset, aiming to fool the discriminator. Simultaneously, the discriminator enhances its ability to discern real from synthetic data. Through this adversarial process, both networks engage in a continuous learning loop, steadily advancing their respective abilities.

The adversarial nature of GANs engenders a symbiotic relationship between the generator and discriminator. As the generator becomes more proficient at crafting realistic data, the discriminator evolves to be more discerning, intensifying the learning loop. Iteration after iteration, this feedback loop propels the networks toward creating increasingly authentic outputs following the loss function in Fig 3. This intricate dance of mutual improvement characterizes GANs, leading to a perpetual evolution in data generation quality. The generator strives for realism, while the discriminator sharpens its discrimination skills. Ultimately, this continuous learning cycle fuels the progression of both networks, shaping GANs into powerful engines capable of generating data that closely emulates real-world examples.

B. Generator

The generator within a Generative Adversarial Network (GAN) serves as the creative force, fabricating synthetic data resembling real examples. Trained on a dataset, this neural network learns to generate data from scratch. It employs complex mathematical transformations to create outputs that closely emulate genuine samples. Through iterative improvement driven by adversarial feedback from the discriminator, the generator refines its ability to produce increasingly authentic data. Its primary aim is to deceive the discriminator by generating data that is indistinguishable from real instances, fostering a continuous quest for higher fidelity in the generated outputs.

C. Discriminator

The discriminator in a Generative Adversarial Network (GAN) acts as the critical evaluator, distinguishing between genuine and synthetic data. Trained to discern real from generated examples, this neural network assesses inputs and provides feedback to the generator. Its role is to classify data as either authentic or counterfeit. Through iterative learning in tandem with the generator, the discriminator refines its ability to detect increasingly subtle differences, becoming adept at identifying synthetic outputs. This network aims to accurately differentiate between real and generated data, compelling the generator to continually improve its output quality in a perpetual competition for creating more convincing data.

D. DC Generative Adversarial Networks

Deep Convolutional Generative Adversarial Networks (DCGANs) represent a specialized iteration of the traditional GAN architecture, specifically designed for handling image data by integrating convolutional layers within both the generator and discriminator networks. This convolutional integration is key to their efficiency in image-related tasks.

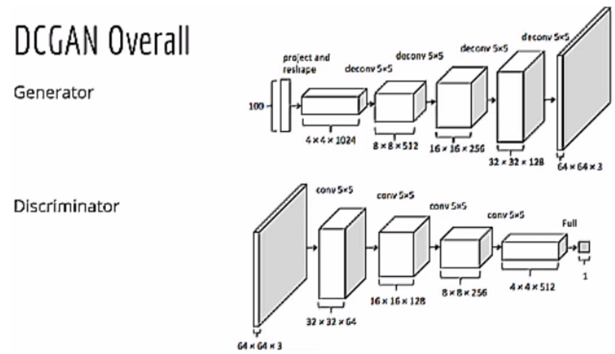

Fig. 2. DC Generative Adversarial Networks

The generator in a DCGAN transforms random noise or initial inputs into high-resolution images through a sequence of transposed convolutional layers. Unlike traditional neural networks, these layers "upsample" the input noise progressively, expanding it into complex representations that resemble genuine images. This step-by-step process allows the generator to learn intricate features and spatial hierarchies, facilitating the creation of increasingly detailed outputs. Conversely, the discriminator, functioning as a convolutional neural network, specializes in distinguishing between real images from the dataset and those generated by the generator. By employing convolutional layers, it can efficiently analyze and classify images based on their intricate visual features and structures. Through training, the discriminator refines its ability to discern the minute differences between authentic and generated images, driving the generator to produce outputs that are progressively more realistic and difficult for the discriminator to distinguish.

The iterative learning process between the generator and discriminator leads to the generation of increasingly sophisticated and authentic images. The convolutional architecture in DCGANs plays a fundamental role in their success by enabling the networks to efficiently handle image data, allowing for the creation of highly realistic images over training iterations.

$$\min_G \max_D V(D,G)$$
$$V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z))]$$

Fig. 3. DCGAN Loss Function

E. Datasets

CIFAR-10 houses 60,000 32x32 color images spanning diverse categories like airplanes, cars, and animals, serving as a benchmark for image classification. Meanwhile, Fashion MNIST comprises 28x28 grayscale fashion item images distributed among 10 categories, offering a challenging alternative to the classic MNIST dataset. Fashion MNIST, depicting various clothing and accessory types, provides a more complex and realistic dataset for evaluating image classification algorithms, acting as a substitute for MNIST in tasks requiring a more nuanced understanding of visual data. Both datasets serve as invaluable resources for benchmarking and testing machine learning models in image-related tasks.

Fig. 4. CIFAR-10 and Fashion MNIST

F. Training on CIFAR-10

For training CIFAR-10, a 30-epoch regimen is set with a Discriminator Learning Rate of 0.0002 and a Generator Learning Rate of 0.0003, utilizing the Adam optimizer with Betas of 0.5 and 0.999. These parameters orchestrate the optimization process to refine the Generative Adversarial Network's performance in image generation.

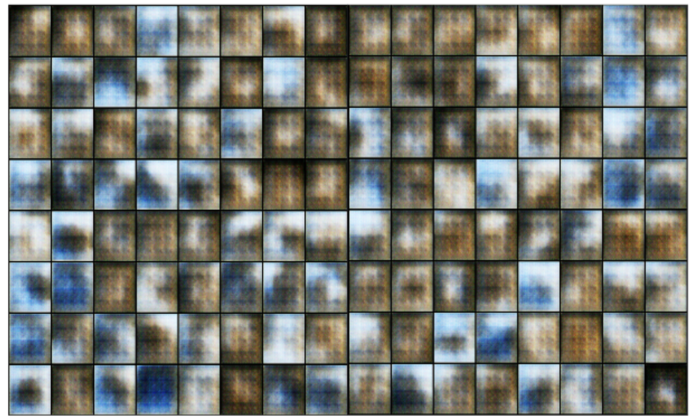

Fig. 5. Epoch 0

The close resemblance between the generated and real images indicates the success of the DCGAN in learning the intricate features and structures present in the CIFAR-10 dataset. Through this training process, the generator evolved to produce images that imitate the appearance of real data. However, by visual inspection it can be concluded that the a more complex or deeper architecture may perhaps be required to fully capture the nuances and details specific to the dataset's diverse classes, such as airplanes, cars, animals, and more.

The GAN's ability to generate images closely resembling real ones suggests that it learned to replicate the statistical patterns and visual characteristics of the CIFAR-10 dataset. Although the images generated do not visually meet the initial expectations, this achievement signifies the effectiveness of the model in creating synthetic data that captures the essence of the original dataset, showcasing the proficiency in generating high-fidelity images.

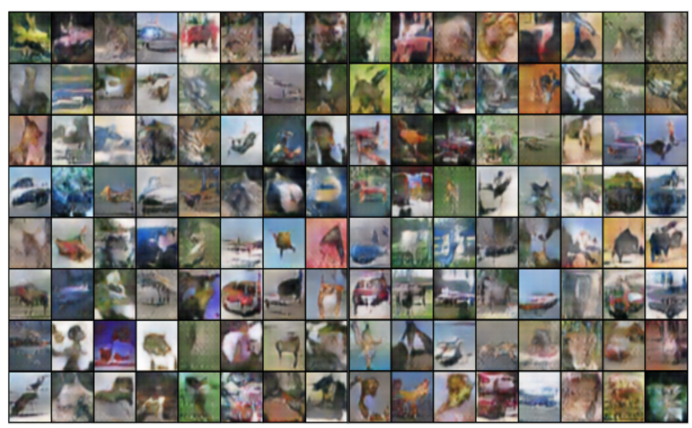

Fig. 6. Epoch 30

G. Training on Fashion MNIST

In this training setup, the DCGAN is trained for 15 epochs using a Discriminator Learning Rate of 0.0001 and a Generator Learning Rate of 0.0003. The optimization process employs the Adam optimizer with Betas set at 0.5 and 0.999, orchestrating the network's learning dynamics for improved performance in generating high-fidelity images.

Fig. 7. Epoch 0

The success of the DCGAN in the Fashion MNIST dataset echoes its proficiency in learning the dataset's nuances. Throughout the training, the generator evolved to craft images resembling real fashion items, capturing diverse apparel types like clothing, shoes, and bags. This close likeness signifies the GAN's grasp of intricate details specific to fashion items. By replicating statistical patterns and visual features, the GAN excelled in generating synthetic data closely mirroring the original Fashion MNIST dataset, demonstrating its ability to create high-fidelity fashion-related images.

Fig. 8. Epoch 15

H. Pruning

Pruning involves reducing neural network size by eliminating non-critical components, enhancing efficiency. Sensitivity pruning removes the least impactful weights (using the standard deviation of the weight distribution), while percentage pruning eliminates a fixed portion of lowest weights, both optimizing models by trimming redundant parameters, and ensuring computational efficiency without sacrificing performance in machine learning models.

Following sensitivity pruning, the model attained an average sparsity of 0.54 percent, selectively removing the least impactful weights. This targeted approach led to a marginal reduction in model complexity, allowing for minor computational gains without substantially altering the network's structure. Despite the minimal sparsity achieved, this method effectively trimmed redundant parameters, enhancing the model's efficiency while retaining its core architecture and performance. Conversely, employing percentage pruning resulted in a substantial increase in sparsity, set at 50 percent. This method systematically eliminated a fixed portion of weights, significantly reducing the model's size and computational demands. While achieving a notable sparsity level, this approach imposed a more drastic alteration to the network's structure. Despite the reduction in parameters, the model maintained a level of functionality and performance suitable for the intended tasks.

Before and after pruning, the most observable change was a slight alteration in image brightness, suggesting the model nearly achieved a dimension loss of zero. This nuanced change indicates that while sensitivity and percentage pruning significantly reduced model complexity, the alterations had a minimal impact on the visual fidelity of the generated images. This outcome underscores the effectiveness of pruning techniques in optimizing neural network efficiency without compromising the structural integrity necessary for generating high-quality images.

Fig. 9. Fashion MNIST after pruning

IV. EXPERIMENTAL ANALYSIS

UQI, SSIM, and MSSSIM are image similarity metrics. UQI evaluates overall image quality by considering luminance, contrast, and structure. SSIM measures structural similarity, indicating how well edges and textures are preserved. MSSSIM extends SSIM by analyzing multiple scales of image information, capturing finer details. These metrics assign scores indicating the likeness between generated and real images. Higher scores imply closer resemblance and better preservation of structural elements. They serve as quantitative measures to assess the fidelity of generated images compared to their real counterparts.

TABLE I
SENSITIVITY PRUNING AT S=0.75

Descriptive Statistics	UQI	SSIM	MSSSIM
Mean	0.922%	1.0+0.0j	1.0
Median	0.930%	1.0+0.0j	1.0
Mode	0.731%	1.0+0.0j	(1.0, 1.0)
Max	0.979%	1.0+0.0j	1.0
Min	0.731%	1.0+0.0j	1.0
Variance	0.001%	0.0+0.0j	0.0
Std	0.038%	0.0+0.0j	0.0

TABLE II
PERCENTAGE PRUNING AT Q=50

Descriptive Statistics	UQI	SSIM	MSSSIM
Mean	0.950%	1.0+0.0j	1.0
Median	0.954%	1.0+0.0j	1.0
Mode	0.859%	1.0+0.0j	(1.0, 1.0)
Max	0.993%	1.0+0.0j	1.0
Min	0.859%	1.0+0.0j	1.0
Variance	0.0007%	0.0+0.0j	0.0
Std	0.027%	0.0+0.0j	0.0

Tables 1 and 2 present a comparative analysis between 250 samples generated by a DCGAN, both with and without pruning techniques. Table 1 showcases the results from Sensitivity training, detailing the similarities and differences observed in the generated samples before and after pruning. Conversely, Table 2 outlines the comparisons for Percentage pruning, highlighting the alterations in the generated samples after this specific pruning method. These tables serve as comprehensive references, delineating the impact of pruning on the quality and characteristics of the generated images, and providing insights into the efficacy of different pruning techniques in maintaining image fidelity.

V. CONCLUSION

Our experiments with a standard DCGAN architecture on a simpler dataset revealed the feasibility of introducing sparsity without compromising image quality. Sensitivity and percentage pruning showcased varying degrees of sparsity, revealing a trade-off between reduced structural similarities metrics like UQI (Universal Image Quality Index) against preserved SSIM (Structural Similarity Index) and MSSSIM (Multi Scale Structural Similarity Index), emphasizing maintained structural integrity despite sparsity's introduction. Optimizing discriminator and generator learning rates independently was crucial for synchronization, ensuring stable training. Yet, managing mode collapse persisted as a challenge across different hyperparameters and kernel sizes, demanding constant monitoring during training. Unfortunately, limited GPU resources constrained our exploration to standard DCGAN architectures and restricted the analysis to smaller, less complex datasets. Consequently, assessing FID and IS scores became unfeasible due to modifications made to handle grayscale images, rendering pre-trained architectures incompatible. While our findings demonstrate the potential for sparsity in maintaining image quality, overcoming mode collapse and resource limitations remains pivotal for broader exploration. Future research could focus on addressing these challenges, potentially enabling the application of sparsity techniques in more diverse architectures and larger datasets. This endeavor would contribute significantly to understanding the interplay between sparsity, image generation quality, and model scalability, fostering advancements in efficient GAN architectures for diverse applications.

REFERENCES

[1] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems 27 (2014).
[2] Salimans, Tim, et al. "Improved techniques for training gans." Advances in neural information processing systems 29 (2016).
[3] Pytorch DCGAN - https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html
[4] Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149 (2015).