HW4 Report

John Pascoe

Github: [CPSC-8430-Deep-Learning/HW4 at main · jpascoe32fb/CPSC-8430-Deep-Learning (github.com)]

Introduction

General Adversarial Networks (GANs) are a class of artificial intelligence algorithms used in unsupervised machine learning, known for their ability to generate new data instances that mimic training data. They are implemented by a system of two neural networks, the generator and discriminator, contesting each other in an attempt to improve data generation capabilities. This report delves into three specific variations of GANs: Auxiliary Classifier GAN (ACGAN), Wasserstein GAN (WGAN), and Deep Convolutional GAN (DCGAN). Each type of GAN model addresses specific challenges in training and stability.

Implementations

Deep Convolutional GAN (DCGAN)

DCGAN's are a direct extension of GANs, primarily using convolutional and convolutional-transpose layers in the generator and discriminator respectively. This structure helps in learning deep representations without pooling layers, using strided convolutions for down-sampling. Batch normalization is used extensively to help maintain a normal distribution in the inputs of each layer. Initial challenges faced when implementing this model were model instability and mode collapse. Overall, the DCGAN architecture promotes the stability of training GANs on high-resolution images and provides a robust framework for unsupervised learning.

Wasserstein GAN (WGAN)

WGAN introduces a novel cost function using the Wasserstein distance, addressing the problem of unstable GAN training, mode collapse, and the failure to converge. The WGAN modifies the traditional GAN by clipping weights to a compact space and using a critic instead of a discriminator. The key issue with my initial WGAN setup was related to the use of weight clipping to enforce the Lipschitz constraint, which could lead to gradient vanishing or exploding problems. This was mitigated through further optimization of the weight clipping extension. Overall, this alteration improves training stability and provides more meaningful learning gradients throughout the training process.

Auxiliary Classifier GAN (ACGAN)

ACGAN extends the GAN architecture by adding a secondary task of classifying images into classes alongside the primary task of generating realistic images. This model leverages labeled data to enforce that generated samples are not only realistic but also classifiable. One of the main difficulties was ensuring that both loss functions for classification and real/fake determination were balanced. If the classification loss dominated, the discriminator could ignore the authenticity of the images. Balancing these losses required careful tuning of the loss contributions. Additionally, ensuring that the generator produces diverse and class-accurate images involves fine-tuning the interplay between the noise and label inputs. Overall, the dual objective of ACGAN's lead to higher-quality, discriminative image synthesis.

Analysis

DCGAN

Figure 1: The training loss of both the generator and discriminator over 50 epochs.

Figure 2: The 10 best generated images by my model after training.

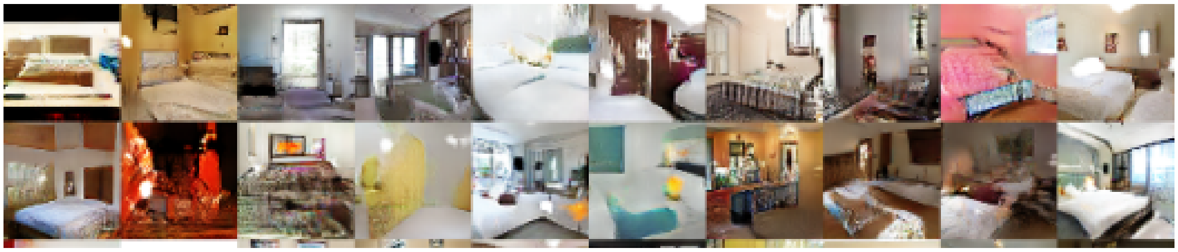

Figure 3: Example images of existing DCGAN generated images.

The training process of the DCGAN model, as reflected in the loss graph, exhibits typical adversarial behavior. The generator loss, represented in orange, initially starts at a higher value and demonstrates significant variance throughout the training epochs. This behavior is indicative of the generator learning and trying to adapt its parameters to better fool the discriminator. The gradual downward trend suggests an improvement in the generator's ability to produce more convincing images as it learns the distribution of the training data over time.

Conversely, the discriminator loss, shown in blue, also displays considerable fluctuation, starting high and eventually converging to a lower, but not stable, value. This pattern is expected in the adversarial training dynamics where the discriminator continuously improves at distinguishing real images from fake ones generated by the generator. The lack of stability towards the end of training may point to a continued challenge for the discriminator in confidently classifying images, which could be due to the increasing quality of the generated images or due to the discriminator reaching its capacity in distinguishing nuances between real and fake data.

The ten best generated images by the DCGAN model post-training display a variety of colors and textures, which suggests that the generator has learned a diverse representation of the dataset features. While some images show areas of coherence that hint at the presence of object-like structures, the overall clarity and recognizability of the images are still limited. The generated images lack the resolution and definition needed to clearly discern the subjects and categories they are meant to represent from the CIFAR-10 dataset. The results indicate that the model has captured some basic aspects of the image distribution but requires further refinement to generate high-quality images. This could involve additional training epochs, tweaks to the model architecture, or hyperparameter optimization.

Based on a visual comparison of Figures 2 and 3, although my model was able to pick up various colors and shapes from training, it wasn’t able to produce clear and identifiable objects like existing networks are able to.

WGAN

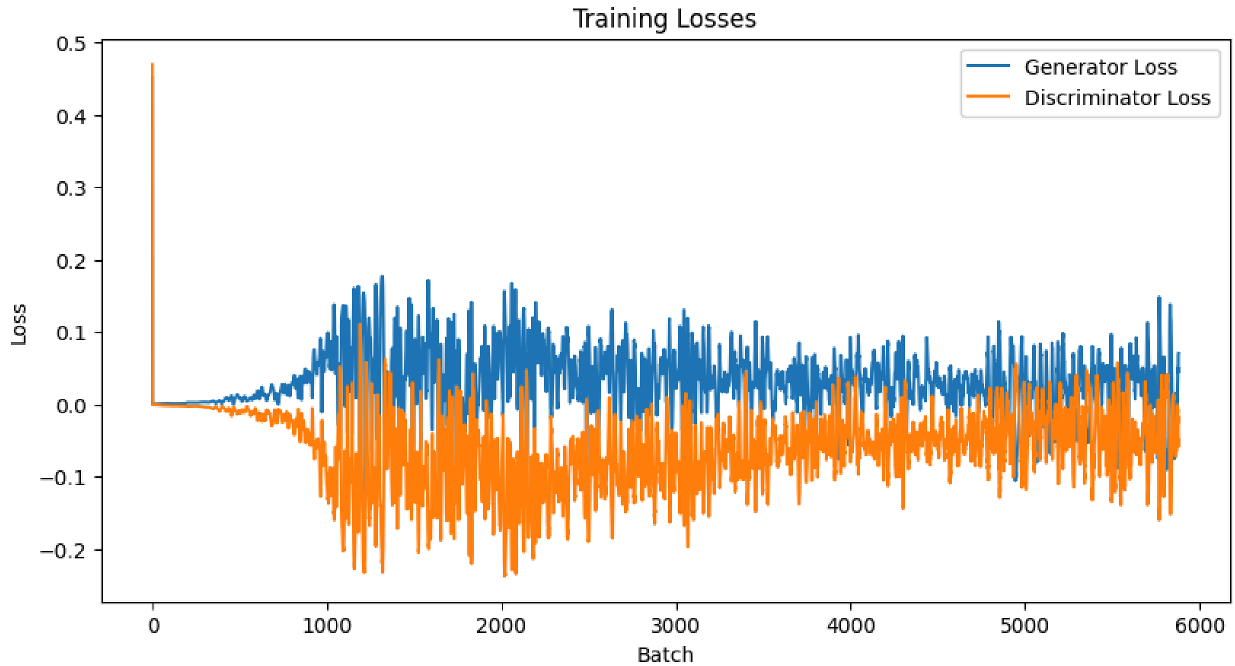

Figure 4: The training loss of both the generator and discriminator over 30 epochs.

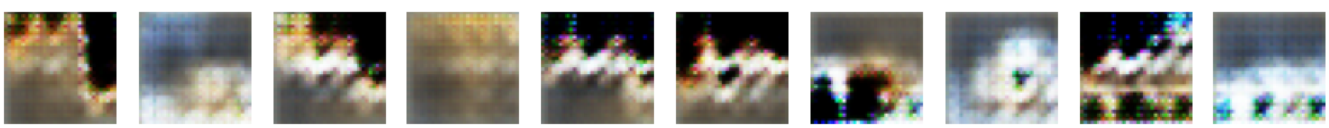

Figure 5: The 10 best generated images by my model after training.

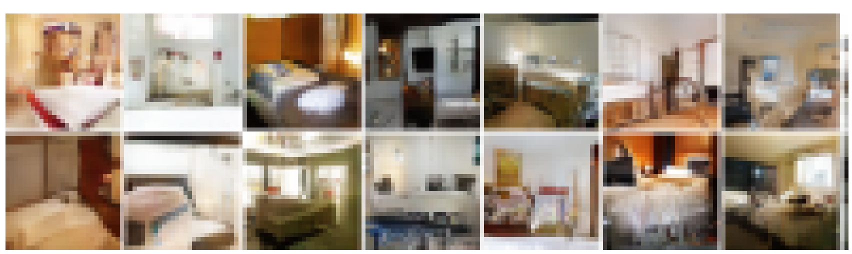

Figure 6: Example images of existing WGAN generated images.

The training loss graph of the WGAN model over 30 epochs shows a more stable convergence compared to a typical DCGAN. The generator loss and discriminator loss are plotted in blue and orange, respectively. Both losses fluctuate within a narrow band, which is a characteristic behavior of Wasserstein GANs due to the use of the Wasserstein distance as a loss function. This metric tends to provide more stable and reliable gradient information during training. In contrast to many GAN variants, the discriminator loss (also known as the criterion in the context of

WGANs) does not converge to zero but rather stabilizes around a value that indicates a balance between the generator and discriminator. This behavior is desirable and indicates that the generator is continuously improving in step with the discriminator, rather than overwhelming it or failing to learn.

The ten best generated images by the WGAN model display a mix of clear patterns and less defined forms, reflecting the ongoing learning process of the generator. The images show some structure and varied texture, suggesting that the generator has learned aspects of the data distribution. However, they still lack the clarity and definition of real-world objects, with some images showing signs of mode collapse or repetitive patterns. Despite this, the images showcase a discernible improvement over random noise, indicating that the WGAN is learning to capture the data distribution. Yet, it's evident that further tuning is needed to enhance the resolution and diversity of the generated images.

Based on the visual comparison of Figures 5 and 6, my model was unable to properly learn colors and image clarity like the existing models. My model showcased the ability to pick up shapes and patterns, but was unable to create images of good visual clarity.

ACGAN

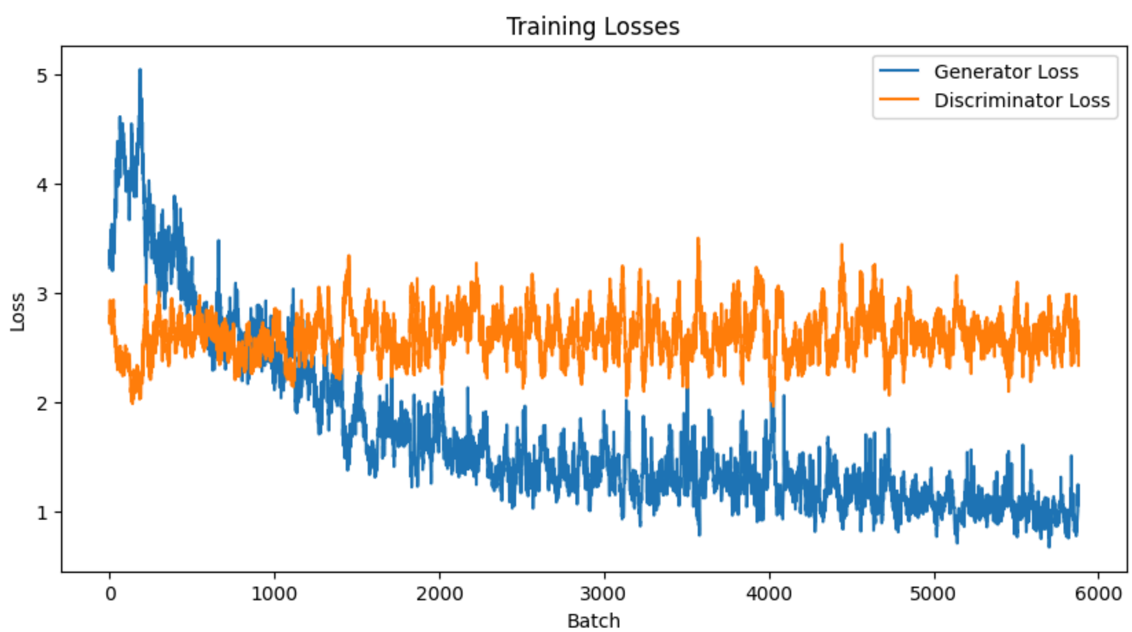

Figure 7: The training loss of both the generator and discriminator over 30 epochs.

Figure 8: The 10 best generated images by my model after training.

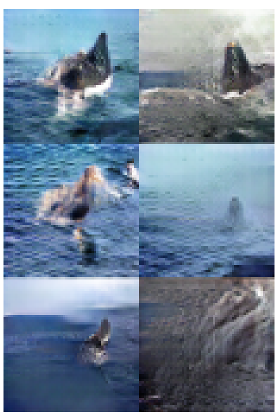

Figure 9: Example images of existing WGAN generated images.

The ACGAN model's training losses show a distinct trend over 30 epochs. The generator loss, shown in orange, and the discriminator loss, depicted in blue, indicate the adversarial dynamic characteristic of GANs. Unlike traditional GANs, ACGAN incorporates class label information, aiming to improve the quality of generated images. In Figure 5, both losses demonstrate an erratic yet contained pattern, suggesting that the model has not yet reached a point of equilibrium where the generator consistently fools the discriminator. However, the generator's loss does not trend upwards, nor does the discriminator's loss plummet to zero, which can often signify a failure mode in GAN training.

The ten best generated images from the ACGAN model present a mixture of defined and ambiguous elements. The images show a variety of shapes and color palettes, indicating that the network has learned a diverse set of features from the data. Despite this, the images still lack clear, recognizable objects and the coherence we might expect from a fully trained model. This is a common challenge with ACGANs, as the model not only needs to generate convincing images but also align them with specific class labels. The presence of some structure in the images does suggest that the conditional generation aspect of the ACGAN is functioning, as it appears to guide the generative process to some extent. However, further training, architectural refinement, or hyperparameter optimization would likely be necessary to achieve crisper and more accurate class-conditional image generation.

Based on visual comparison of Figures 8 and 9, like existing models, my ACGAN model was able to learn color palettes and shapes of varying degrees. Unlike existing models, my model was unable to generate clear images with objects that are obvious to discern.

Conclusion

This project highlighted the practical challenges and theoretical considerations in training GANs. Each GAN variant addressed specific issues like training stability, mode collapse, and image diversity. While DCGAN provided a good starting point with its simplicity, WGAN brought robustness in training, and ACGAN introduced the utility of labeled data for enhanced specificity in generation tasks. Several strategies were employed to optimize the training of these models such as Gradient Penalty, Label Smoothing, and Batch Normalization. Across all models, several optimization strategies were pivotal. Gradient Penalty emerged as a powerful technique for enforcing the Lipschitz constraint, proving instrumental in the success of WGAN training. Label Smoothing was another technique utilized, aimed at making the discriminator less confident in its assessments, thereby preventing it from overpowering the generator. Batch Normalization was also widely used to normalize the input to layers within the networks, promoting faster learning and higher overall stability. The iterative nature of this work underscored the experimental aspect of machine learning research. With each training cycle, new insights were gleaned—some reinforcing established theories, others challenging preconceptions, all contributing to the growing body of knowledge around generative models.