Image Synthesis With Generative Adversarial Networks (GANs) Using CIFAR-10
CS-GY 6953 / ECE-GY 7123 Deep Learning Project Report — Spring 2024

Ark Pandey[1*], **Durga Avinash Kodavalla**[1*], **Priyangshu Pal**[2*]

[1]New York University, Tandon School of Engineering, Department of Electrical and Computer Engineering (ECE)
[2]New York University, Tandon School of Engineering, Department of Computer Science and Engineering (CSE)
Project GitHub Repository: click here

Abstract

This project thoroughly studies image synthesis with Generative Adversarial Networks (GANs)[7] using CIFAR-10[1] dataset as the benchmark. GAN architecture trained with a discriminator and generator together, augmented with Wasserstein loss and gradient penalty (WGAN-GP)[4] for improved stability and convergence. Fréchet Inception Distance (FID) and Inception Score (IS)[5] as evaluation metrics, indicating the quality and diversity of generated images. With an FID score of 12.06 and an IS score of 1.0, the generated images closely resemble real CIFAR-10 samples. This project underscores the proficiency of GANs in image synthesis, mainly in conjunction with WGAN-GP. Our findings contribute to advancing and providing valuable insights for future research in this domain.

Introduction

The demand for generating realistic images is increasing across various industries, driven by the growing use of technologies like augmented reality, virtual environments, and digital content creation. This prompts the need for advanced deep-learning solutions. This project leverages Generative Adversarial Networks (GANs) to synthesize diverse and photorealistic images from scratch trained on the CIFAR-10 dataset. The project aims to gain a deeper understanding of GAN models, their training dynamics, and the challenges in generating high-quality images. The outcome of this project will contribute to understanding the image synthesis technology and have practical applications in various fields, meeting the current need for realistic image generation and paving the way for future innovations.

Literature Survey

Ian J. Goodfellow et al.'s paper "Generative Adversarial Nets,"[2] published in 2014, introduced the groundbreaking framework of Generative Adversarial Networks (GANs). GANs introduce a game-theoretic framework where two neural networks, a generator and a discriminator, are trained simultaneously in an adversarial manner. Despite their success, traditional GAN training methods often face challenges such as mode collapse, training instability, and limited image quality, particularly when generating high-resolution images. Karras et al. proposed the "Progressive Growing of GANs" technique[3], which aims to cope with these challenges by incrementally growing both the generator and discriminator networks during training. This technique offers a promising solution to longstanding challenges in generating high-resolution, realistic images.

Dataset

The CIFAR-10 dataset contains 60000 32x32 color images. The dataset has 10 classes, each having 6000 images and these classes are completely mutually exclusive. Training a GAN on CIFAR-10 requires the generator to learn patterns and textures to generate realistic images without demanding a lot of compute resources, making it a suitable benchmark for evaluating the capabilities of GAN architectures for the project.

Architecture

Generative Adversarial Networks (GANs) are employed in unsupervised machine learning, which is a result of two neural networks competing with each other in a zero-sum game fashion. This involves two models: a generative model which is the generator that captures the data distribution and a discriminative model which is the discriminator that estimates the probability that the sample came from the dataset rather than the generator. The generator has a goal of generating outputs that are good enough to deceive the discriminator.

Model	Trainable Parameters
Generator	3,448,576
Discriminator	2,637,312

Table 1: Number of trainable parameters

I. GENERATOR

Architecture:

- The first step of our Generator model is a vector input with a latent space that is processed through a number of transposed convolutional layers. The following layer

*These authors contributed equally.

of the network upscales the image dimension while reducing the depth, which gradually moves from a deep, spatially small representation to a shallow, spatially large one.

- The network has batch normalization and ReLU activations after every transposed convolution except the last one, where the Tanh function is applied for scaling the outputs to range [-1, 1] in the same way as the CIFAR-10 images were preprocessed.

Reason for Use:

- This architecture is suitable for generating the images because it gradually builds the spatial structure, improving the image as layers are added. This method is especially useful when the depth of the network increases because it creates more fine details.
- Batch normalization is a technique that is applied to the input layer in order to stabilize the learning process by normalizing the input by adjusting and scaling activations.

Advantages:

- The particular architecture allows to perform the learning and upscaling of the features in a latent space to a full 32x32 image. It is in fact best for producing small images such as those in CIFAR-10.
- It can create various images having a low level of artifacts because of the careful architecture design.

II. Discriminator

Architecture:

- The Discriminator uses a standard convolutional neural network architecture, which is progressively downscaling the image, increasing the depth to produce a high-dimensional representation of the input image.
- It uses Leaky ReLU activations which are responsible for the small gradient when the unit is not active. This could be a potential reason for gradients to flow easily through the architecture. It applies batch normalization for most layers to maintain the distribution of the inputs.

Reason for Use:

- This model is successful in differentiating between real and fake images because it utilizes deep and powerful networks that can capture fine details.
- LeakyReLU helps in the solving of the "dying ReLUs" issue, which happens in networks with plain ReLU activations.

Advantages:

- It offers robust training suitability as the discriminator should be powerful enough to guide the generator toward generating realistic images.
- It can effectively handle the complexity of real-world image distributions like those found in the CIFAR-10 dataset

In general, the architectural decisions for the generator and the discriminator are suitable for training GANs on the CIFAR-10 dataset, which has a lot of complexity and diversity. The architectures enable the models with the ability to deal with the features essential for the generation of detailed small images.

Methodology

I. LOSS FUNCTION

Discriminator Loss (Wasserstein GAN with Gradient Penalty - WGAN-GP): The Wasserstein loss is used as the loss function in the discriminator (critic) of our model, and it measures the Earth mover's distance between the distributions of real data and generated data. This loss function is more stable and more meaningful than traditional GAN loss functions through training. Specifically, the discriminator's loss is computed as:

$$\text{Discriminator loss} = D(x) - D(G(z)) + \lambda_{gp} \times \text{Gradient Penalty}$$

Here, D(x) is the discriminator's score for the real images and D(G(z)) is the score for the fake images generated by the generator. The gradient penalty (WGAN-GP) imposes a Lipschitz constraint, which is important for the theoretical properties of the Wasserstein GAN.

Generator Loss: The generator's task is to deceive the discriminator by producing images that look like real images.

$$\text{Generator Loss} = -mean(D(G(z)))$$

In this case, the generator loss is the negative mean of the discriminator's predictions on the fake images produced by the generator. With this configuration, the generator will try to produce images that get higher scores from the discriminator which will then make those images look more realistic.

This loss configuration creates a feedback loop in which the discriminator is motivated to improve in judging real from fake images, and the generator is pushed to generate more convincing fakes. The gradient penalty used in the discriminator loss (WGAN-GP) also helps in stabilizing the training process by enforcing the Lipschitz constraints that is essential for the critic to guide the generator correctly.

II. GRADIENT PENALTY

- **Implementation Details:** The gradient penalty is an important part of the training process of the Wasserstein GANs. It aids in the enforcement of the Lipschitz constraint which the critic (discriminator in other GAN types) needs. It does this by calculating the gradient of the discriminator's scores with respect to interpolations between real and fake images.
- **Gradient Computation:** Interpolated images are produced by blending real and fake images together with a random epsilon value (alpha). The discriminator assesses the interpolations, and the gradients of these scores with respect to the interpolated images are calculated.

III. HYPERPARAMETERS

- **Batch size:** 64
- **Learning rate (lr):** 0.0002 for both generator and discriminator.
- **Latent size:** 100, which refers to the size of the random noise vector input to the generator.
- **Epochs:** 100
- **Lambda for Gradient Penalty (λ_{gp}):** 10, which controls the weight of the gradient penalty in the loss function.

IV. DATA AUGMENTATION

The CIFAR-10 dataset is augmented during preprocessing to include:

- Random horizontal flips to provide mirror-image invariance.
- Random rotations of up to 10 degrees to provide orientation invariance.
- Random crops with padding of 4 pixels on every side and then crop them back to their original size. This provides a simulation of translation and thus the system can tolerate slight changes in the object's position.

V. LEARNING RATE SCHEDULERS

Schedulers for both G and D: We implement a step learning rate scheduler for the generator and discriminator. The learning rate is reduced by a factor of 0.1 every 30 epochs. This feature works such that the initial larger steps in the landscape can be followed by smaller, more accurate ones as training proceeds, which results in a better and more stable convergence.

VI. OTHER TRAINING DETAILS

- **Optimizers:** Adam optimizer chosen for both generator and discriminator with betas set at (0. 5, 0. 999) which controls the decay rates of moving averages of the gradients and their squares.
- **Loss Logging and Model Saving:** After each epoch, the losses for both, the generator and the discriminator, are logged and model weights are saved after every 10 epochs. This is a kind of process that allows monitoring of training progress and restart of training from the checkpoint.
- **Image Generation:** A set of images is produced at each checkpoint that is generated by the current state of the generator to see the quality and the way the images are developing over the training period.
- Regular Clearing of CUDA Cache: This is done for the optimization of GPU memory while training.

VII. FRECHÉT INCEPTION DISTANCE (FID)

- **Definition:** FID (or Fréchet Inception Distance) is a metric that shows the difference between the feature vectors calculated for the real and fake images. These feature vectors are obtained by a ResNet18[6] network that has been pre-trained on the CIFAR-10 dataset. The FID is computed by comparing the mean and covariance of the feature vectors between the real and generated images, which are measured with the Fréchet (or Wasserstein-2) distance.
- **Interpretation:** Lower FID values represent the fact that the generated pictures are more alike the real pictures, which means better image quality and variety.

VIII. INCEPTION SCORE (IS)

- **Definition:** Inception Score takes the Inception model to classify each generated image into one of the CIFAR-10 classes and then analyzes the distribution of these classes. It measures the clarity and diversity of the generated images based on two main aspects:
  1. The entropy of labels distribution for each image – a low entropy means that the model is confident (clear) in the class of each image.
  2. The fact that there are different predictions in the dataset—a high entropy of the distribution of predicted classes along all the images means that there is good diversity.
- **Interpretation:** The higher the IS values are, the more likely the generated images are meaningful (to the classifier) and diverse.

Results

The Generative Adversarial Network (GAN) model has performed well. The negative values of the discriminator loss indicate that the discriminator can effectively distinguish between real and fake images. The generator loss values indicate the ability of the generator to produce images that are realistic enough to deceive the discriminator. Loss values stabilize towards the later epochs, indicating that the model has reached a stable state. In Figure 1, we plot the Generator and Discriminator loss curves against each epoch. We have

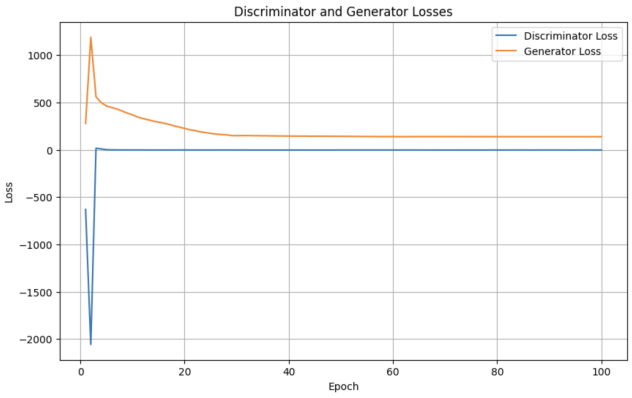

Figure 1: Generator and Discriminator loss curves

used The Fréchet Inception Distance (FID) metric is used to measure the similarity between the distributions of features extracted from real images and generated images. Our model achieved a score of Fréchet Inception Distance 12.05. We

have also used the Inception Score metric to assess the quality and diversity of images created by our generative model. Our model achieved an Inception Score of 1.0. In Figure 2,

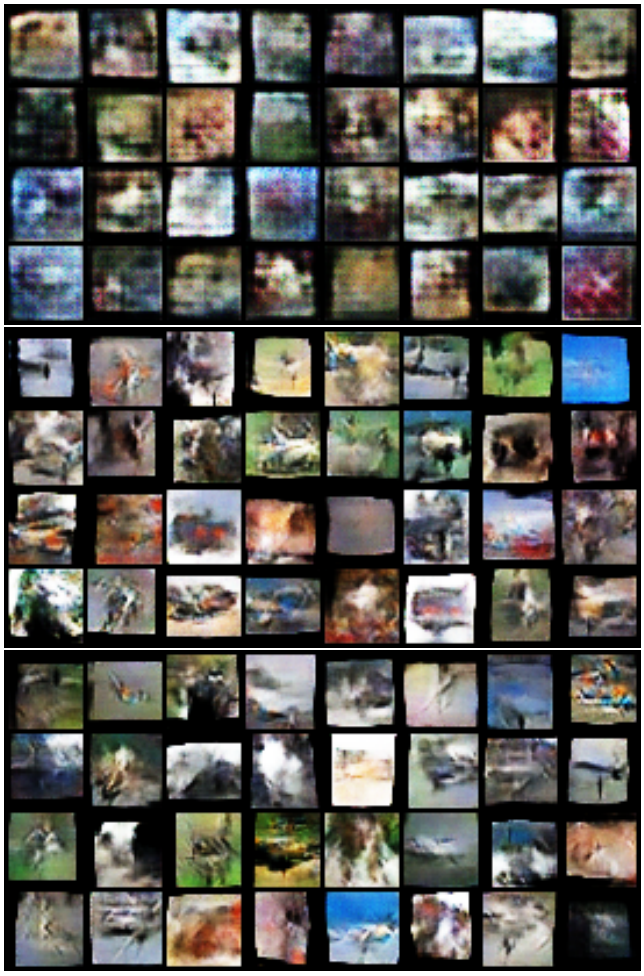

Figure 2: Images generated by the GAN model

we can see the images generated by the GAN model after training for 10, 50, and 100 epochs.

Conclusion

This project focused on the application of advanced techniques, including Wasserstein GANs with Gradient Penalty (WGAN-GP) useful for stabilizing, and converging the model during the training process. Visual inspection of generated images improved with an increase in training epochs of the generator. Loss plots provided insights into training dynamics, showing a steady decrease in generator loss. Based on evaluation metrics, FID and IS scores reveal that the generator can synthesize images relatively close to real images in terms of visual features, good quality, and a high degree of diversity in terms of their content and appearance. The successful task of generating credible images from CIFAR-10 highlights the potential of GANs in tasks requiring realistic image synthesis.

References

[1] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. (2014). Generative Adversarial Networks.

[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen, Progressive Growing of GANs for Improved Quality, Stability, and Variation. 2018.

[4] E. Gallagher and B. O'Sullivan, "Reconstructing Dystopian Urban Scenes using Generative Adversarial Networks," in Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science, 2019.

[5] Y. Chong, J. Lee, D. E. Carlson, and A. W. Rivenson, "Effectively Unbiased FID and Inception Score and Where to Find Them," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[6] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv:1512.03385, 2015.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," arXiv:1406.2661, 2014.