IMPLEMENTATION OF GAN'S AND CGAN'S

VELAMALA RAHUL(SC21B127)

05-05-2024

Abstract

This project explores the application of Generative Adversarial Networks (GANs) and Conditional GANs (cGANs) in synthesizing data for two distinct domains: handwritten digits and anime faces. Through extensive experimentation, we demonstrate the ability of these models to generate realistic samples resembling the characteristics of the respective datasets.

Using the MNIST dataset, GANs and cGANs generate plausible handwritten digits, showcasing their potential for data augmentation in computer vision tasks. Additionally, leveraging an anime face dataset, GANs produce high-quality synthetic anime faces with diverse characteristics, promising applications in creative industries.

The project involves rigorous experimentation, including hyperparameter tuning and performance evaluation, ensuring the robustness of the generated samples. We discuss potential applications and future directions for GAN-based approaches in data augmentation and creative content generation.

Overall, this project contributes to the growing research on GANs, highlighting their versatility and impact in various domains, from computer vision to creative arts.

1 Introduction

The generation of realistic synthetic data is a crucial task in various fields, ranging from computer vision to creative content creation. In this project, we address this challenge by employing Generative Adversarial Networks (GANs) and Conditional GANs (cGANs) to generate synthetic data in two distinct domains: handwritten digits and anime faces.

The importance of generating realistic synthetic data lies in its utility for tasks such as data augmentation, model training, and creative content generation. In domains like computer vision, where labeled data may be scarce or expensive to obtain, synthetic data generation offers a cost-effective solution for enlarging datasets and improving model performance. Similarly, in creative industries, such as animation and character design, the ability to generate diverse and lifelike synthetic content can streamline the production process and foster creativity.

Our results demonstrate the effectiveness of GAN-based models in generating high-quality synthetic data that closely resembles the characteristics of the original datasets. Through extensive experimentation, we showcase the ability of these models to produce realistic handwritten digits and anime faces, highlighting their potential for various applications. Furthermore, we discuss the implications of our findings and potential avenues for future research in leveraging GANs for synthetic data generation.

Overall, this project addresses the pressing need for realistic synthetic data generation and underscores the significance of GAN-based approaches in advancing various fields.

2 Related Work

Previous research has extensively explored the use of Generative Adversarial Networks (GANs) for synthetic data generation, particularly in domains such as handwritten digits and anime face synthesis. Studies have utilized architectures like DCGANs and con-

ditional GANs to generate realistic samples resembling MNIST digits and anime-style facial images.

Our approach shares similarities with existing work, leveraging GANs for synthetic data generation. However, we extend beyond by exploring both handwritten digits and anime faces within the same framework. Additionally, we emphasize the importance of conditional information for fine-grained control over generated attributes. Through rigorous experimentation, we optimize model performance, contributing to the understanding of best practices in GAN training for synthetic data generation

3 Data

For the anime face generation aspect of our project, we utilized a dataset sourced from Kaggle, comprising 63,632 "high-quality" anime faces. This diverse dataset encompasses various anime characters and facial expressions, providing a rich source for training our GAN model. Prior to training, we standardized the image size and normalized pixel values to ensure consistency and facilitate model convergence.

In contrast, for the handwritten digit generation component, we employed the classical MNIST dataset, consisting of 60,000 training images and 10,000 testing images of handwritten digits (0 through 9). Each grayscale image is 28x28 pixels in dimension. Preprocessing steps included normalizing pixel values to the range [0, 1] and reshaping images to a suitable format for model input.

These preprocessing steps were crucial to ensure uniformity and consistency in both datasets, enhancing the performance and convergence of our GAN models. By standardizing image size and normalizing pixel values, we facilitated effective training of our models for synthetic data generation in the respective domains of anime faces and handwritten digits.

4 Methods

Our approach to solving the problem of synthetic data generation in the domains of anime faces and handwritten digits revolves around the utilization of Generative Adversarial Networks (GANs) and Conditional GANs (cGANs). These deep learning architectures are well-suited for generating realistic synthetic data by learning the underlying distribution of the training data and generating samples that resemble it.

1. Anime Face Generation:

For anime face generation, we employed a GAN architecture consisting of a generator and a discriminator network. The generator generates synthetic anime faces from random noise, while the discriminator distinguishes between real and synthetic images. Through adversarial training, the generator learns to generate increasingly realistic anime faces, while the discriminator learns to distinguish between real and fake images.

2. Handwritten Digit Generation:

Similarly, for handwritten digit generation, we utilized a GAN architecture. The generator generates synthetic handwritten digits, while the discriminator differentiates between real and synthetic digits. By training the GAN adversarially, the generator learns to produce realistic digit samples, while the discriminator improves its ability to discriminate between real and synthetic digits.

Justification and Comparison:

Our approach using GANs and cGANs is appropriate for several reasons:

GANs have demonstrated success in generating high-quality synthetic data across various domains, making them a suitable choice for our task. Conditional GANs allow for control over the attributes of the generated samples, enabling us to generate specific anime faces with desired characteristics or handwritten digits with particular attributes. The adversarial training process encourages the generator to produce samples that closely resemble the distribution of the training data, resulting in realistic synthetic data. Compared to alternative methods such as traditional image synthesis techniques or rule-based approaches, GANs offer several advantages:

GANs do not rely on explicit rules or heuristics for generating samples, allowing for more flexible and adaptable generation of diverse data. GANs can capture complex patterns and correlations in the training data, leading to the generation of more realistic and

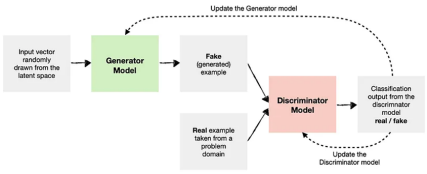

Figure 1: Architecture of GAN

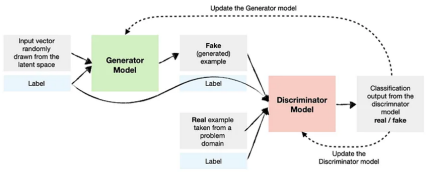

Figure 2: Architecture of Conditional GAN

diverse samples. GANs are data-driven and learn directly from the training data, making them suitable for tasks where the underlying data distribution is complex or unknown.

4.1 Error Function in GANs

In GANs, the training objective involves a minimax game between two neural networks: the generator G and the discriminator D. The generator tries to generate samples that are indistinguishable from real data, while the discriminator aims to differentiate between real and fake samples. The objective function for GANs can be formulated as:

$$\min_G \max_D V(D,G) =$$

$$\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

Here, $p_{\text{data}}(x)$ represents the distribution of real data, $p_z(z)$ is the distribution of noise vectors sampled from a prior distribution, $G(z)$ is the generated sample, and $D(x)$ is the discriminator's output indicating the probability that x comes from real data.

4.2 Error Function in cGANs

In cGANs, additional conditional information y is provided to both the generator and discriminator. This conditioning allows for control over the generated samples based on specific attributes. The objective function for cGANs can be formulated as:

$$\min_G \max_D V(D,G) =$$

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

Here, $p_{\text{data}}(x,y)$ represents the joint distribution of real data and its corresponding conditional labels y, and $G(z,y)$ is the generated sample conditioned on y.

4.3 Training Process

During training, both Generative Adversarial Networks (GANs) and Conditional GANs (cGANs) utilize an iterative optimization process. This process involves updating the parameters of the generator and discriminator networks alternately to minimize the objective function, typically a form of adversarial loss.

Generator Update:

- The generator aims to generate synthetic samples that are indistinguishable from real data. It updates its parameters to minimize the discrepancy between the distribution of generated samples and real samples, as perceived by the discriminator.
- The generator's objective is to minimize the generator's loss function, which penalizes it for generating samples that the discriminator can easily identify as fake.

Discriminator Update:

- The discriminator aims to accurately distinguish between real and fake samples. It updates its parameters to maximize its ability to differentiate between real and generated samples.
- The discriminator's objective is to maximize its loss function, which penalizes it for misclassifying real samples as fake and vice versa.

Adversarial Training:

- The training process alternates between updating the generator and discriminator networks. The generator generates synthetic samples, which are then fed into the discriminator along with real samples.

- The discriminator provides feedback to the generator by assessing the realism of the generated samples. This feedback guides the generator's updates to produce more realistic samples in the next iteration.

- Similarly, the discriminator's parameters are updated based on its ability to distinguish between real and generated samples accurately.

Convergence:

- The iterative optimization process continues until the generator produces samples that are indistinguishable from real data, and the discriminator cannot differentiate between real and generated samples effectively.

- At convergence, the generator has learned to capture the underlying distribution of the training data, producing realistic synthetic samples.

In summary, both GANs and cGANs employ iterative optimization, where the generator and discriminator networks are updated alternately to minimize the objective function. This adversarial training process enables the generator to produce realistic synthetic samples while guiding the discriminator to accurately differentiate between real and fake samples.

5 Experiments

Score was Epoch for 400 epochs of MNIST class and Score was Epoch for 25 epochs of Anime class was represented as a graph in this section.And the hyper parametes were choosen such that the discriminator is as low as possible and the generator loss is also not compramised.

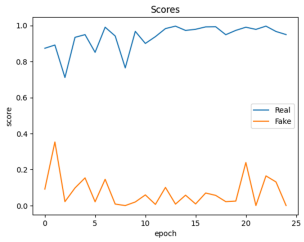

Figure 3: Score vs Epoch (In Anime Face DataSet)

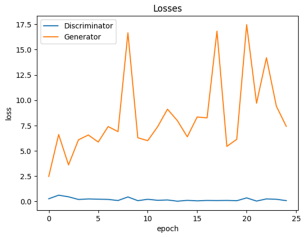

Figure 4: Losses vs Epoch (In Anime Face DataSet)

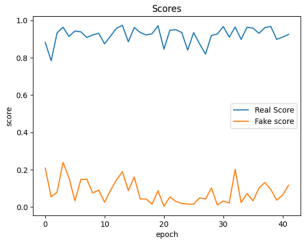

Figure 5: Score vs Epoch (In MNIST DataSet)

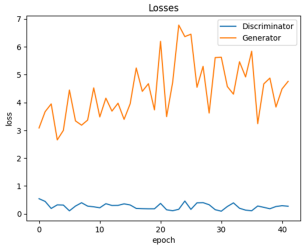

Figure 6: Score vs Epoch (In MNIST DataSet)

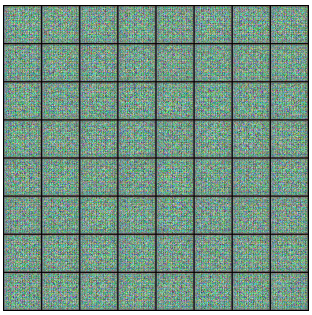

Figure 7: First Epoch in Anime Face Dataset

Figure 8: Final Epoch in Anime Face

6 Conclusion

The above images show the images generated in first epoch and final epoch showing the improvment of the generated as the epochs are increasing.

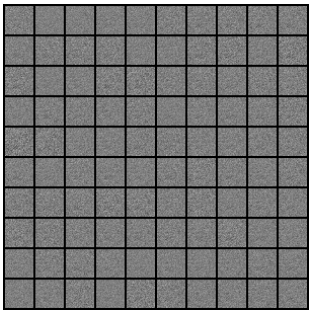

Figure 9: First Epoch in MNIST

Figure 10: Final Epoch in MNIST

Figure 11: Conditional GAN output