Université Dauphine PSL

Data Science Lab 2: Generative Adversarial Network

MEDJAOURI Insaf, MAANINOU Mehdi, HUANG Zhe

November 2023

1 Introduction

Generative Adversarial Networks (GANs) have revolutionized the field of machine learning, particularly in the realm of image generation. This project explores the capabilities of GANs by training a model on the MNIST dataset, a comprehensive collection of handwritten digits ranging from 0 to 9 that serves as a benchmark for evaluating machine learning models.This report outlines our methodology, experiments, and the implications of our findings in the context of advancing GAN technology to generate 10K synthetic samples from the MNIST dataset.

2 Vanilla GAN

2.1 The structure Of vanilla GAN

Vanilla GAN is a model composed of two neural networks: the Generator (G) and the Discriminator (D). The Generator begins with an input from the latent space—a source of random noise—and attempts to generate data that mimics real data samples. These generated samples are then passed to the Discriminator alongside actual real samples from the dataset. The Discriminator's job is to distinguish between the real and fake samples, effectively learning the characteristics of the true data distribution.The Generator is optimized to produce increasingly convincing data, while the Discriminator becomes better at detecting the Generator's fakes. This process is iterative, with both networks improving through this adversarial process, driving the Generator to create data that is ever closer to the real data distribution, aiming for a point where the Discriminator can no longer tell the difference between real and generated samples.

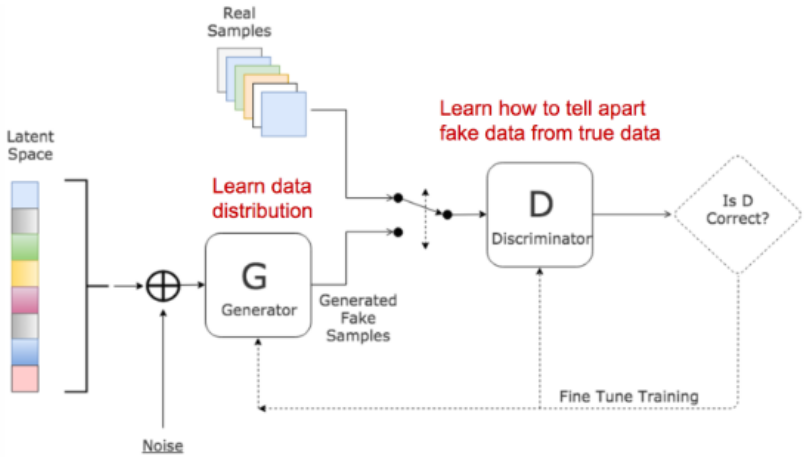

Figure 1: Structure of a Vanilla Generative Adversarial Network (GAN)

2.2 Training and results

For training the Vanilla Generative Adversarial Network, we conducted extensive hyper-parameter tuning to optimize the model's performance. We tested various combinations of hyper-parameters locally and retained the configurations that yielded the best results. The optimal settings were then committed to GitHub. Our evaluation was based on a set of metrics, including evaluation time, recall, and precision, which are crucial indicators of model efficiency and effectiveness.

- **Precision:** precision measures how many of the generated samples are realistic or of high quality. A higher precision means that a larger proportion of the images produced by the GAN are indistinguishable from real images, or meet a certain quality threshold.
- **Recall:** Recall in GANs assesses the variety or diversity of the generated samples.High recall indicates that the GAN can generate a wide variety of realistic samples, not just a few types of high-quality images.
- **Fréchet Inception Distance (FID):** FID is a metric for assessing the quality of generated images, comparing the distribution of generated images to the distribution of real images.It is defined by the equation:

$$\text{FID} = \sqrt{\sum_i (\mu_1(i) - \mu_2(i))^2 + \text{Tr}(\mathbf{C}_1 + \mathbf{C}_2 - 2\sqrt{\mathbf{C}_1\mathbf{C}_2})}$$

The optimal hyper-parameters are as follows:

- **Learning Rate**: 0.0001
- **Number of Epochs**: 200
- **Batch Size**: 128

Here are the results:

Hyper-parameters	Time(s)	FID	Precision	Recall
Default values	106.2	379.21	0.0	0.0
Optimized values	113.74	42.5	0.57	0.22

Table 1: Performance metrics for the default and optimized GAN hyper-parameters.

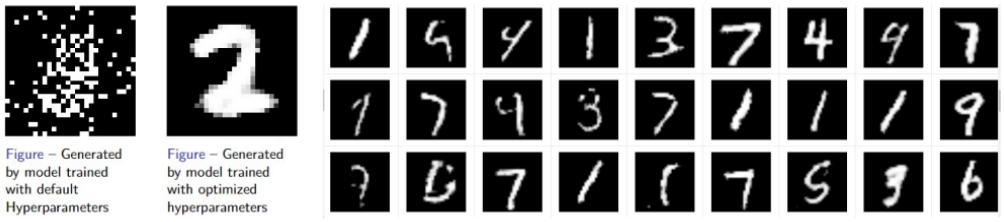

Figure 2: Sample Outputs from Optimized Vanilla GAN and Comparative Results of GAN : Default vs Optimized hyper-parameters

2.3 Challenges Encountered by Vanilla GANs

The following are common problems encountered during the training of Vanilla Generative Adversarial Networks:

- **Mode Collapse:** A frequent issue where the generator starts producing the same or similar outputs over and over again. This phenomenon, known as Mode Collapse, signifies the generator's inability to capture the diversity of the real data distribution, limiting itself to a narrow subset that deceives the discriminator.

- **Vanishing Gradient:** This problem occurs when the discriminator becomes overly proficient, labeling nearly all generated samples as fake. This leads to a loss function that approaches zero, causing the learning process to slow down. In such cases, the generator's adjustments are insufficiently significant for it to improve.

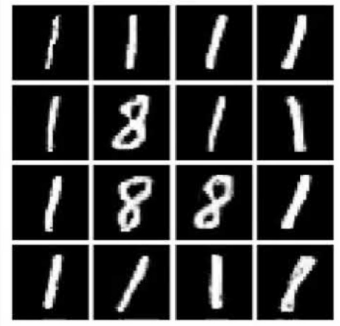

Figure 3: Mode Collapse

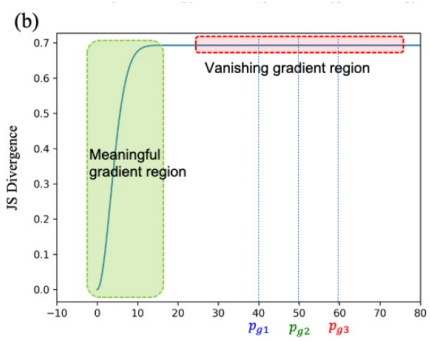

Figure 4: Vanishing Gradient

3 WGAN

One of the main problems with traditional GANs is the instability of the training process, which can result in mode collapse, vanishing gradients, and other issues. WGANs addressed this problem by using the Wasserstein distance as their loss function, which is more stable and provides more meaningful gradients to the generator.

3.1 Concept behind WGAN

WGAN introduces the Wasserstein distance as a metric for measuring the difference between the real data distribution and the generated distribution in terms of how much "work" is required to transform one distribution into the other,

$$W(P,Q) = \inf_{\gamma \in \Gamma(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|]$$

explanation of the formula:

- $\gamma \in \Gamma(P,Q)$ is the set of all joint distribution over x and y
- $(x,y) \sim \gamma$ can be seen as an amount of mass that must be moved from x to y to transform P to Q

This formulation emphasizes that the Wasserstein distance is the minimal expected cost of transporting mass from one distribution to another, considering all possible ways of achieving this transport.

The goal is to make this distance through training ass small as possible, fake data become similar to real data

3.2 What WGAN brings

From a technical point of view, Vanilla GAN and WGAN are very similar except that the critic does not have an output sigmoid function. The major difference is only on the loss:

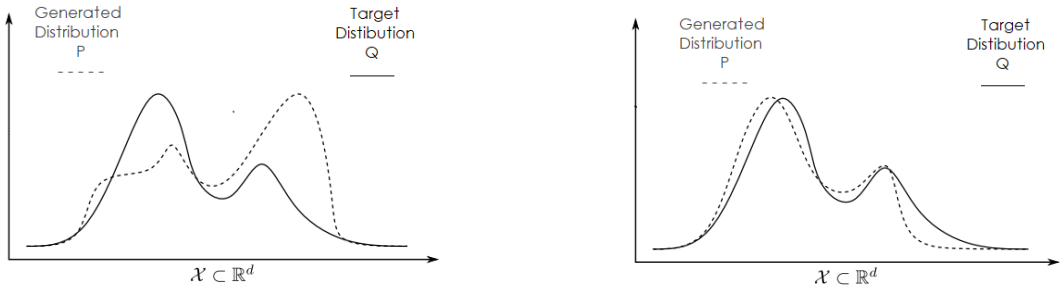

Figure 5: Evolution of generated distribution through training

- Vanilla GAN uses binary cross-entropy loss
- WGAN replaces it with Wasserstein distance

	Discriminator/Critic	**Generator**
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D(x^{(i)}) + \log\left(1 - D(G(z^{(i)}))\right)\right]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(D(G(z^{(i)}))\right)$
WGAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[f(x^{(i)}) - f(G(z^{(i)}))\right]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} f(G(z^{(i)}))$

Table 2: Loss functions for GAN and WGAN models.

We get the following optimisation problem:

$$L = \min_{g} \max_{f} \mathbb{E}_{x \sim P}[f(x)] - \mathbb{E}_{z \sim Z}[f(g(z))]$$

The generator tries to minimize the loss function while deceiving the discriminator, and the discriminator tries to maximize the loss function while correctly distinguishing real samples from generated samples.

We experimented with various combinations of hyperparameters to optimize the training of our WGAN model. Through this rigorous experimentation, we were able to achieve a significantly improved recall rate. This improvement is a clear indication that the WGAN model effectively mitigates the mode collapse probleme.

Using Wasserstein loss, offers stability during training, because to ensure that the gradients from discriminator doesn't explode the weights of the discriminator (f) are constrained to a specified range determined by the hyper-parameter c.

Algorithm 1 Update weights w using Adam and clipping

0: $w \leftarrow w - \alpha \cdot \text{Adam}(w, g_w)$ {Update using Adam}

0: $w \leftarrow \text{clip}(w, -c, c)$ {Clip the weights} =0

While weight clipping is a straightforward method for regularizing the critic in WGANs, it often leads to issues such as suboptimal image quality and convergence problems. The success of this approach hinges on the precise tuning of the hyper-parameter 'c', the clipping threshold. An inadequately chosen value can significantly impede the model's convergence and overall performance. In

practice, weight clipping has been found to be less effective, primarily due to its inherent limitations in managing the critic's weight magnitudes efficiently.

3.3 WGAN-GP

Instead of using weight clipping, researchers used Lipschitz continuity as a regularization term which penalizes loss if the value of its gradient deviates from 1. It prevents from exploding and vanishing,

$$\lambda \left(\|\nabla f(\hat{x})\|_2 - 1 \right)^2 \quad \text{where} \quad \hat{x} = \varepsilon x + (1 - \varepsilon) g(z)$$

- λ :This is a hyper-parameter that controls the strength of the regularization.
- $\nabla f(\hat{x})$: It represents the gradient of the discriminator function f with respect to its input $\hat{x}$.
- $\hat{x} = \varepsilon x + (1 - \varepsilon) g(z)$: This is a perturbed point along the straight line between a real data point x and a generated data point $g(z)$, controlled by the parameter ε.

We explored various combinations of hyperparameters and selected the optimal combination, which is documented in our Google Doc. Our WGAN-GP outperforms the standard WGAN:

Results on platform	**Time(s)**	**FID**	**Precision**	**Recall**
WGAN	116.8	139.58	0.59	0.01
WGAN-GP	110.79	36.58	0.52	0.33

Table 3: Performance metrics for our WGAN and WGAN-GP.

Figure 6: Generated Samples on the platform by our best WGAN-GP

4 Conclusion

In this project, we explored the capabilities of Generative Adversarial Networks (GANs), focusing on Vanilla GAN and WGAN-GP. Our key achievements include the successful generation of high-quality synthetic images and improved performance metrics, highlighting the potential of GANs in learning complex data distributions. We faced challenges in terms of computational resources and optimization time. Future research should explore scalability and integration with other machine learning techniques. Overall, our work contributes to a deeper understanding of GANs and their practical applications in machine learning.

References

[1] Lilian Weng. (2017). *From GAN to WGAN* [Blog Post].
https://lilianweng.github.io/posts/2017-08-20-gan/

[2] Martin Arjovsky, Soumith Chintala, Léon Bottou. (2017). *Wasserstein GAN* [Research Paper].
https://arxiv.org/abs/1701.07875

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville. (2017). *Improved Training of Wasserstein GANs* [Research Paper].
https://arxiv.org/abs/1704.00028