Analyzing GAN Training Challenges and Experimental Insights

Khadeeja Toseef

9th September 2025

Introduction

Generative Adversarial Networks (GANs) have become a cornerstone of generative modeling, capable of synthesizing realistic data across domains such as image generation, text-to-image synthesis, and style transfer. Despite their success, training GANs remains highly unstable due to the adversarial interplay between the generator (G) and discriminator (D). This work explores fundamental challenges in GAN training, including mode collapse, vanishing gradients, and discriminator overfitting. We implement a baseline GAN on the MNIST dataset and conduct controlled experiments to observe these challenges and test mitigation strategies. Our goal is to better understand the training dynamics of GANs and the effectiveness of common stabilization techniques.

Methodology

We implement a fully connected (MLP) GAN in PyTorch and train it on the MNIST dataset for 50 epochs.

Model Architecture

- **Generator:** Three linear layers with ReLU activations and a Tanh output layer to produce images in the range $[-1, 1]$.
- **Discriminator:** Three linear layers with LeakyReLU activations and a Sigmoid output layer for real/fake classification.

Training Setup

We use binary cross-entropy loss with the Adam optimizer (learning rate $= 0.0002$, betas $(0.5, 0.999)$). Real and fake batches are alternately presented to the discriminator, followed by generator updates. We track generator and discriminator losses, and periodically sample images for qualitative inspection.

Controlled Experiments

To analyze training challenges, we design three experiments:

1. **Vanishing Gradient:** Increase discriminator learning rate so it rapidly overpowers the generator.
2. **Mode Collapse:** Increase generator learning rate, encouraging repetitive sample generation.
3. **Discriminator Overfitting:** Train on a restricted subset of 1,000 MNIST images to study memorization.

Results

Baseline Training

The baseline GAN produced recognizable digits (e.g., “1”, “7”, “9”), though with blurry edges compared to convolutional GANs. Discriminator loss quickly decreased and stabilized, while generator loss oscillated before partially stabilizing. Diversity across digits existed, but occasional bias toward specific digit modes was observed.

Vanishing Gradient

With an increased discriminator learning rate, the discriminator quickly achieved near-perfect accuracy, while the generator loss plateaued, indicating stalled learning. Applying label smoothing (real labels = 0.9) and non-saturating loss partially restored gradient flow.

Mode Collapse

With a higher generator learning rate, the generator collapsed to producing nearly identical digits (e.g., only “3s”). Strengthening the discriminator (training D more often than G) and introducing input noise perturbations improved sample variety.

Discriminator Overfitting

When trained on a reduced dataset, the discriminator memorized training samples, achieving near-zero loss. The generator began replicating memorized digits with little diversity. Adding Dropout (0.4) to the discriminator reduced overfitting, leading to more varied generated samples.

Discussion

Our experiments confirm the instability of GAN training:

- The **vanishing gradient** problem arises when the discriminator is too strong, leaving the generator unable to improve. Label smoothing is an effective mitigation strategy.
- **Mode collapse** highlights the difficulty of balancing adversaries. Without proper tuning, the generator exploits shortcuts, producing limited outputs.
- **Discriminator overfitting** reduces the model’s ability to generalize, leading to mode dropping and sample replication. Regularization techniques such as Dropout are crucial.

Comparing to Erik Linder-Norén’s PyTorch-GAN reference implementation, we find that convolutional architectures (DCGAN-style) produce sharper and more diverse digits. Differences arise from architectural capacity, latent dimension size, and longer training schedules.

Conclusion

We have demonstrated several common challenges in GAN training using a baseline MLP GAN on MNIST. Our experiments showed how vanishing gradients, mode collapse, and discriminator overfitting manifest in practice and how simple mitigation strategies (e.g., label smoothing, balanced updates, dropout) can improve training stability. Although our simple GAN produced digit-like samples, convolutional GANs remain superior in capturing spatial features. This study highlights the delicate dynamics of GAN optimization and motivates the exploration of more robust architectures such as WGAN-GP and DCGAN for future work.

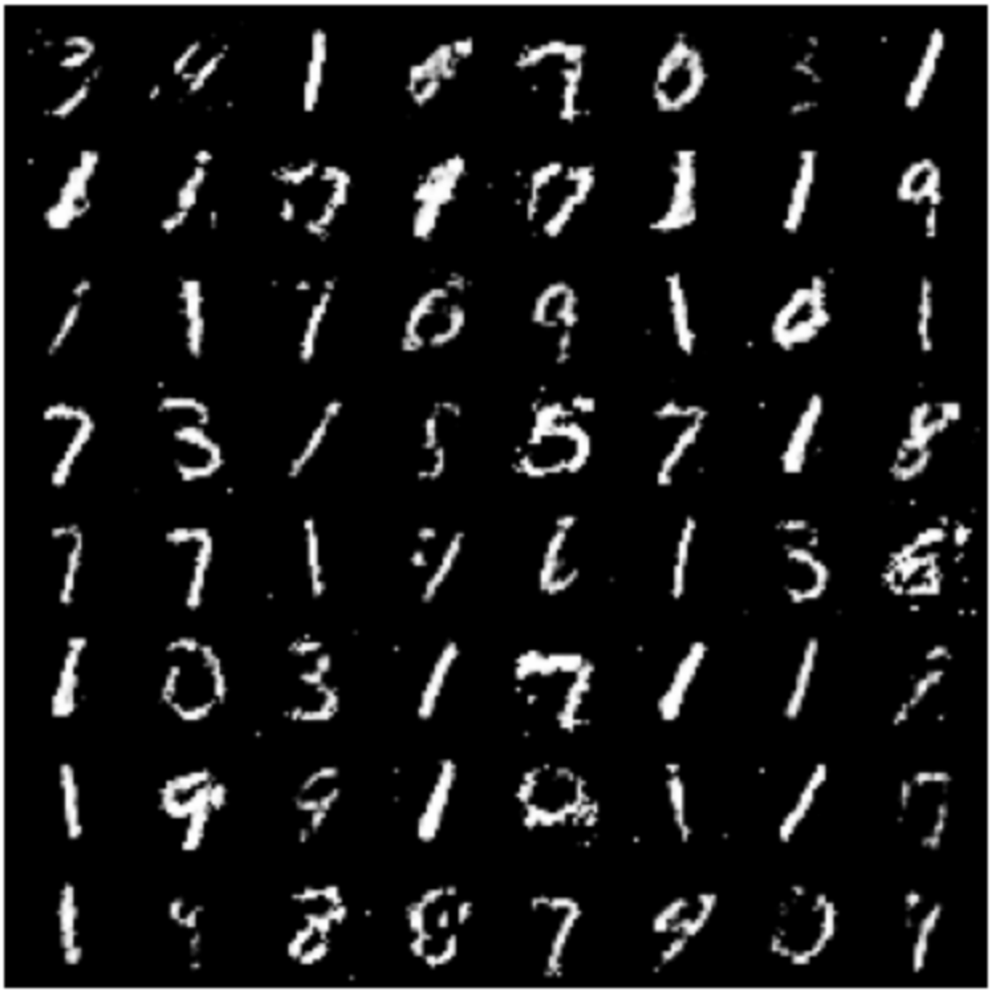

Figure 1: images generated by vanilla gan.

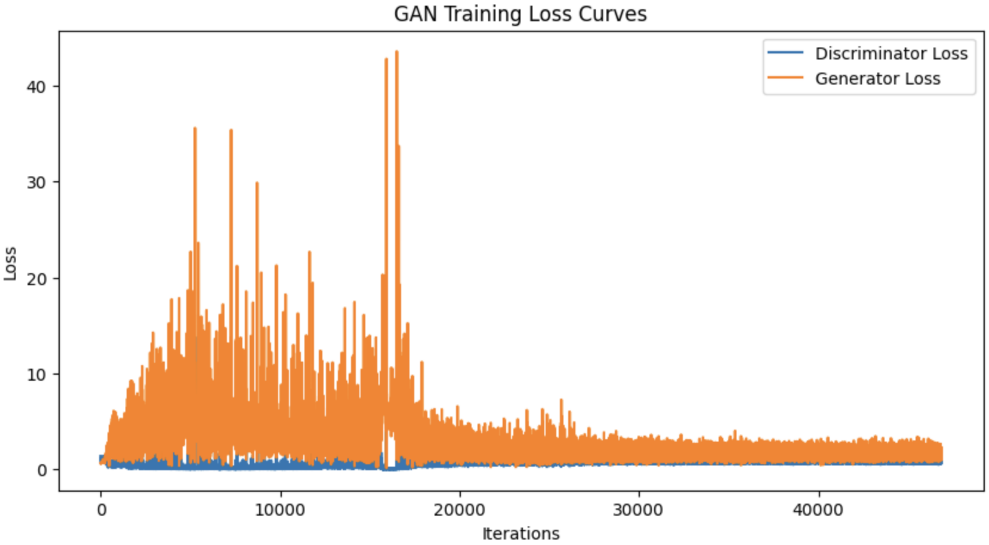

Figure 2: gan training loss curve.