Learning and Implementing Generative Adversarial Networks (GANs)

Comparative Study of Basic and Improved GAN Using PyTorch

Purpose: Gaining Hands-on Experience and Understanding of GANs through Coding Implementation with the MNIST Dataset

Zhiqi Pang

2023.9.27

Basic GAN Overview

- **Objective**: To generate data that's similar to some real data.
- **Architecture**:

  Utilizes Multi-Layer Perceptron (MLP) with fully connected layers.
  - **Generator** (creates images) :
    - Linear Layers: [100, 256], [256, 512], [512, 784]
    - Activation: LeakyReLU, Tanh
  - **Discriminator** (evaluates them):
    - Linear Layers: [784, 512], [512, 256], [256, 1]
    - Activation: LeakyReLU, Sigmoid

- **Training:**
  - Learning Rate: 0.0002
  - Batch Size: 100
  - Number of Epochs: 100
  - Optimizer: Adam
  - Loss: Binary Cross Entropy Loss

- **Steps for Coding Implementation**:

  1: Import Necessary Libraries
  2: Set Hyperparameters
  3: Load the MNIST dataset
  4: Construct the architecture of the Generator
  5: Develop the architecture of the Discriminator
  6: Initialize Models and Optimizers
  7: Implement the training loop
  8: Assess the model's performance using FID and IS
  9: Inspect the quality of images generated

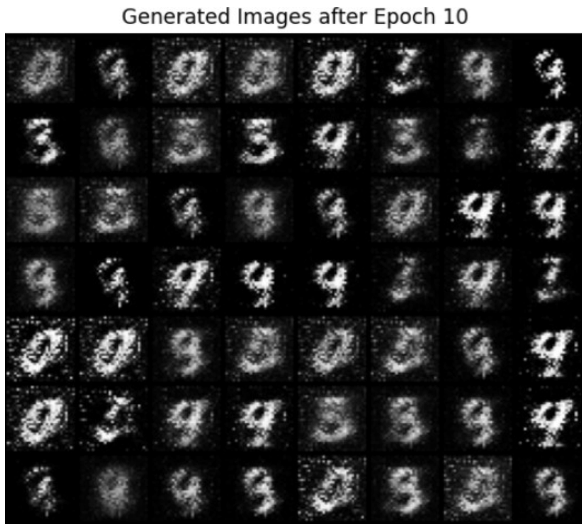
Basic GAN Overview

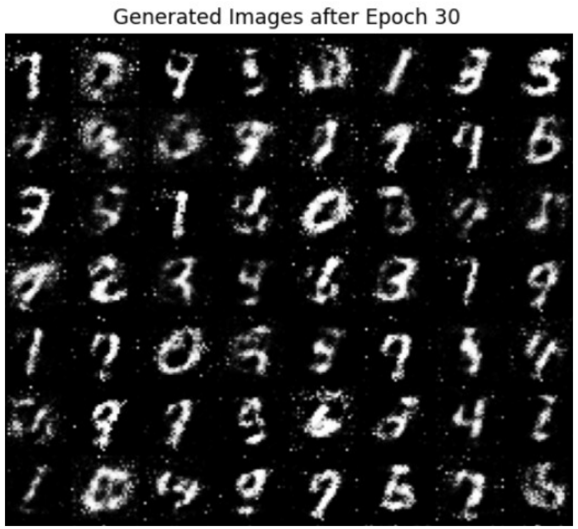
- **Architecture**:

  Utilizes Multi-Layer Perceptron (MLP) with fully connected layers.

  - **Generator** (creates images) :
    - Linear Layers: [100, 256], [256, 512], [512, 784]
    - Activation: LeakyReLU, Tanh
  - **Discriminator** (evaluates them):
    - Linear Layers: [784, 512], [512, 256], [256, 1]
    - Activation: LeakyReLU, Sigmoid

- **Training:**
  - Learning Rate: 0.0002
  - Batch Size: 100
  - Number of Epochs: 100
  - Optimizer: Adam
  - Loss: Binary Cross Entropy Loss





Convolutional GAN

- **Objective**: Enhance Basic GAN for refined image generation and increased model complexity via CNN incorporation, aiming for clearer and more detailed outputs.
- **Improvements**:
  - Employ **Convolutional Layers** in both Generator and Discriminator.
  - Incorporate **Batch Normalization** in Generator.
  - Apply **Dropout** & **Label Smoothing** in Discriminator.
- **Architecture**:
  - **Convolutional Generator** :
    - ConvTranspose2D layers
    - ReLU Activations
    - Batch Normalization
    - Output: Tanh Activation
  - **Convolutional Discriminator**:
    - Conv2D layers with Dropout: 0.3
    - LeakyReLU Activations
    - Output: Sigmoid Activation
- **Training:**
  - Learning Rate: 0.0002
  - Batch Size: 100
  - Number of Epochs: 100
  - Optimizer: Adam ($\beta1 = 0.5$, $\beta2 = 0.999$)
  - Loss: Binary Cross Entropy Loss
  - Real Labels: 0.9 (for Label Smoothing)

Convolutional GAN

- **Architecture**:
  - **Convolutional Generator** :
    - ConvTranspose2D layers
    - ReLU Activations
    - Batch Normalization
    - Output: Tanh Activation
  - **Convolutional Discriminator**:
    - Conv2D layers with Dropout: 0.3
    - LeakyReLU Activations
    - Output: Sigmoid Activation

- **Training:**
  - Learning Rate: 0.0002
  - Batch Size: 100
  - Number of Epochs: 100
  - Optimizer: Adam (β1 = 0.5, β2 = 0.999)
  - Loss: Binary Cross Entropy Loss
  - Real Labels: 0.9 (for Label Smoothing)

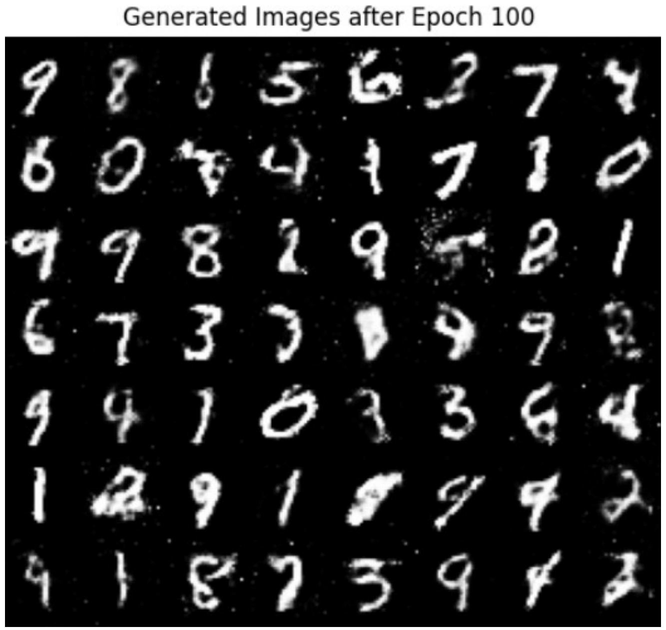

Generated Images after Epoch 10


Generated Images after Epoch 30

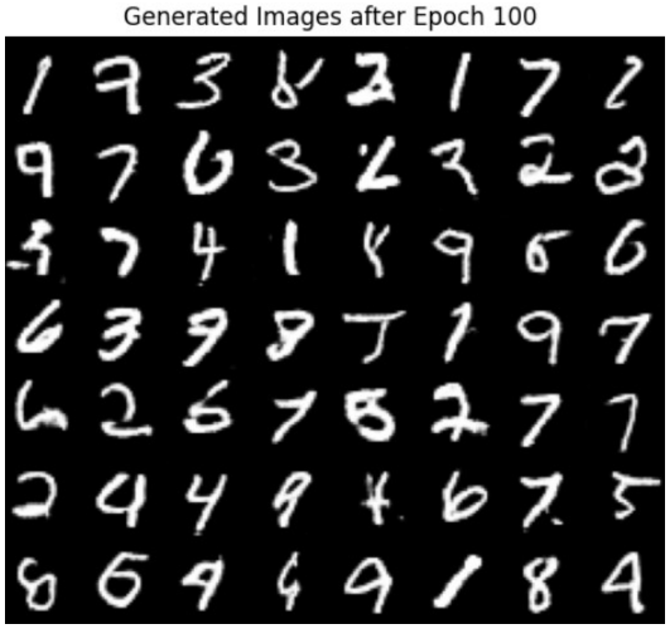

Generated Images after Epoch 70

Comparative Analysis

- **Resolution Comparison**:
  - The generated images after 100 epochs show a marked difference in clarity between the Basic and Convolutional GANs.
  - It is evident that the Convolutional GAN produces images with sharper and clearer details compared to its Basic counterpart.



Basic GANs



Convolutional GANs

Analyzing Training Dynamics GANs

- **Key Metrics**:
  - **D(x):** Discriminator’s prediction on real images

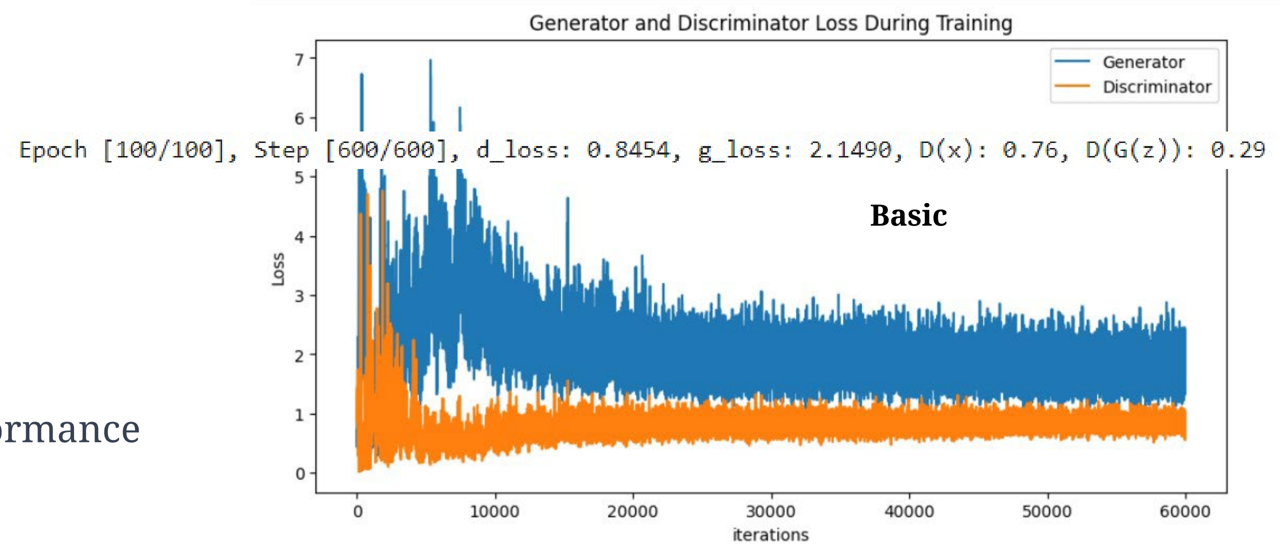
    Closer to 1 is ideal.
  - **D(G(z)):** Discriminator’s prediction on generated images

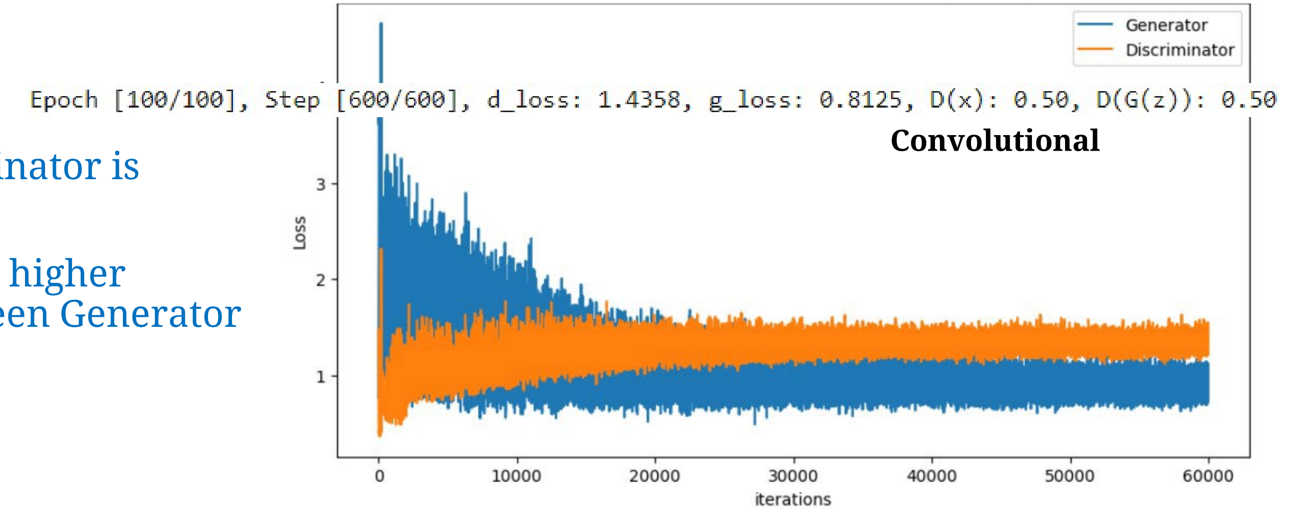
    Closer to 1 indicates better Generator.
  - **g_loss** & **d_loss**

    Lower values are better, indicating improved model performance

    Balancing is crucial to avoid model collapse.

- **Observations**:
  - **Basic GANs** seem to have a disparity where the discriminator is dominating the generator.
  - **Convolutional GANs** exhibit more balanced values, but higher Discriminator loss, indicating a better competition between Generator and Discriminator.



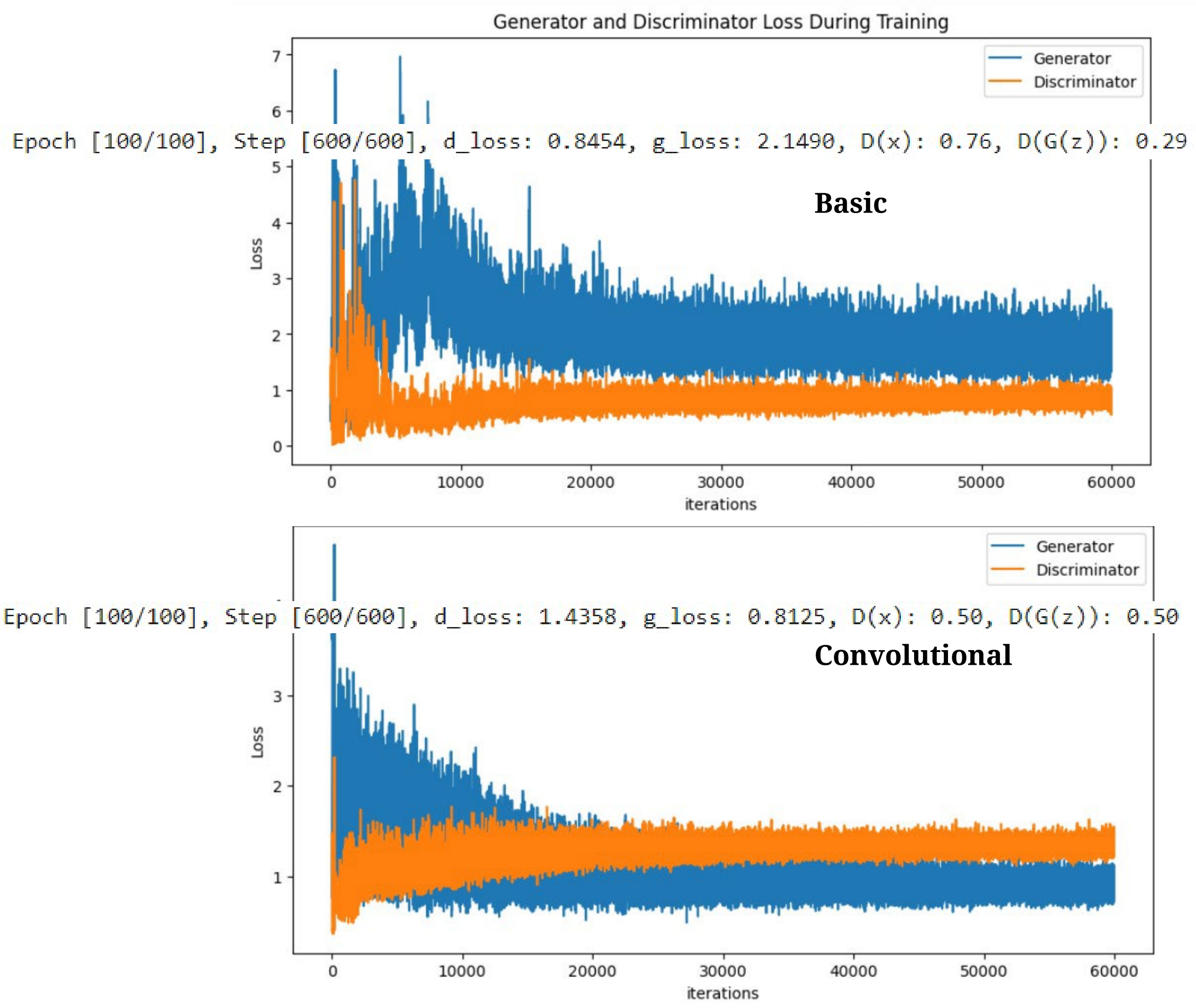


Comparative Analysis

- **Generator Performance:**
  - In Basic GANs, the generator struggles more, reflected by higher g_loss, implying less convincing generation of images.
  - Convolutional GANs have lower g_loss, suggesting the generator can create more believable images.

- **Discriminator Performance:**
  - The Basic GAN’s discriminator is more proficient at differentiating between real and fake, evidenced by its lower d_loss and higher D(x).
  - the discriminator in Convolutional GANs struggles more, with a higher d_loss and D(G(z)) values nearing 0.5, suggesting a more balanced competition between the Generator and the Discriminator.

- **Model Balance:**
  - Convolutional GANs illustrate a more balanced competition, which is crucial for the sustained learning of both the generator and the discriminator.



Evaluating GAN models:

- **Frechet Inception Distance (FID) :**
  - Measures the similarity between generated and real images, taking into account the feature distribution in the image representation space.
  - Lower FID indicates better model performance and more realistic generated images.
- **Inception Score (IS) :**
  - Evaluates the quality and diversity of generated images by computing the KL divergence between the conditional label distribution and the marginal label distribution of generated samples.
  - Higher IS implies better image quality and diversity.

- **Basic GAN:**
  - FID: 281.92
  - Inception Score: 1.46 (±0.13)
- **Convolutional GANs:**
  - FID: 162.17
  - Inception Score: 2.11 (±0.26)

- **Comparative Insight:**
  - The Convolutional GAN outperforms the Basic GAN, evident by a lower FID and a higher Inception Score, indicating that the images generated by the Convolutional GAN are of higher quality and more realism.
  - The higher standard deviation in IS for the Convolutional GAN also suggests more diversity in the generated images.

Thank you

feedback, suggestions, continuous learning