جامعة الأمير مقرن بن عبد العزيز
University of Prince Mugrin

Department of Artificial Intelligence

College of Computer and Cyber Sciences

Introduction to Deep Learning

Generative Adversarial Network (GAN) for Image Generation

1. Learning Objectives

By the end of this lab, students will:

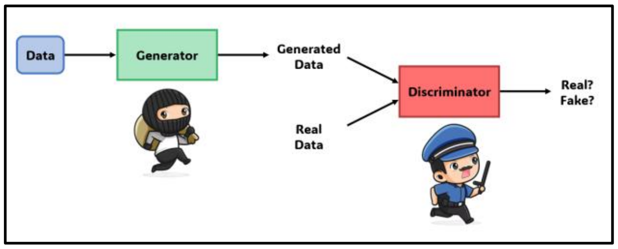
- Understand the fundamental concepts of Generative Adversarial Networks (GANs), including the roles of the **Generator** and **Discriminator**.
- Learn how to **build and train** both the Generator and Discriminator networks using PyTorch.
- Implement a GAN using PyTorch to **generate images** based on a given dataset.

2. Explanation of Key Concepts

– **Generative Adversarial Networks (GANs):**

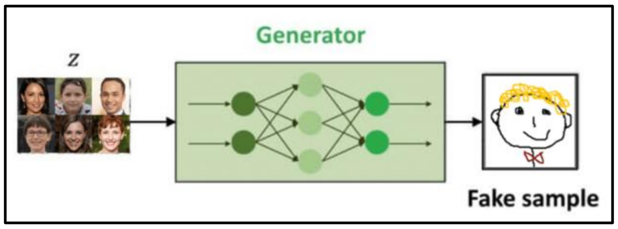
GANs are type of machine learning model composed of two neural networks: a Generator and a Discriminator.

- The Generator creates fake data (e.g., images) based on random noise.
- The Discriminator tries to distinguish between real and fake data.



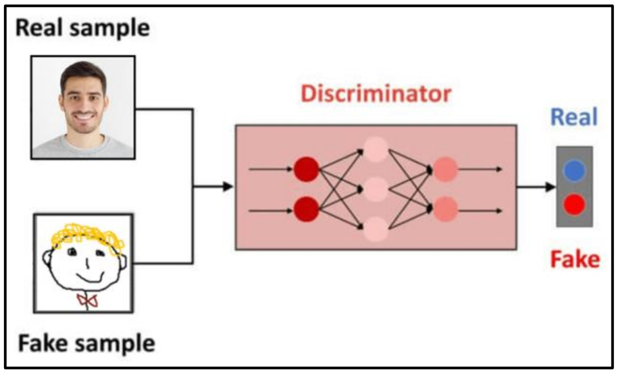
– **Generator:**

Generator in GANs is a neural network that generate realistic data samples similar to training data.

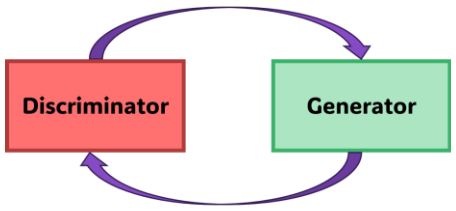


– **Discriminator:**

Discriminator in GANs is a neural network that is trained to distinguish between real data and generated (fake) data.



– **Training GANs:**



- Step 1:

Train the **discriminator** on real data and fake data.

The goal is to minimize classification loss.

- Step 2:

Train the **generator** on real data only.

The **generator** is trained to maximize the **discriminator** loss.

Simply …

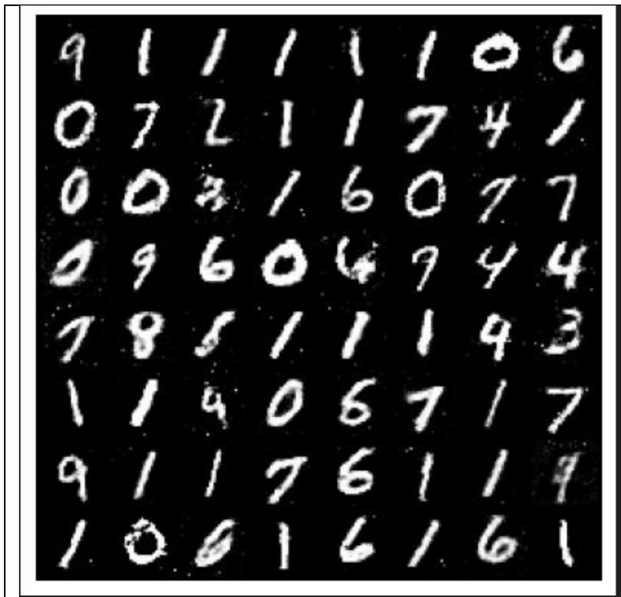
Train the **generator** to beat the **discriminator**.

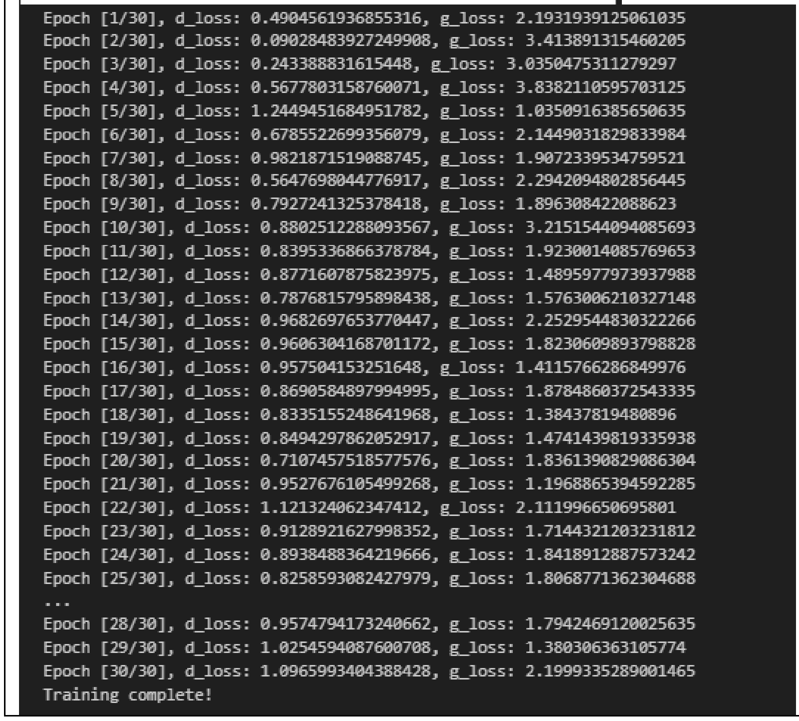
Train the **discriminator** to beat the **generator**.

3. Activities

➢ Exercise 1: Building a GAN Model for MNIST Image Generation.

In this exercise, we will build a Generative Adversarial Network (GAN) to generate images from the MNIST dataset using PyTorch. We will build a simple GAN with a Generator and a Discriminator to generate realistic MNIST images.





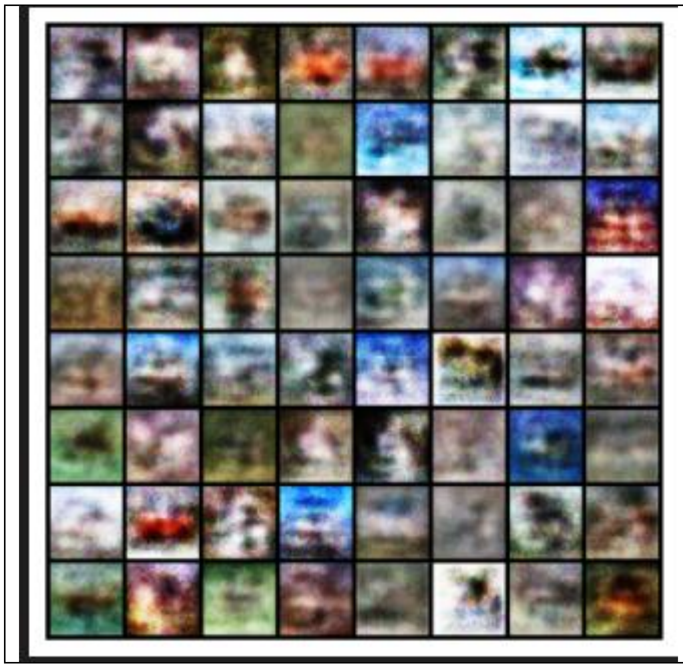
```
Epoch [1/30], d_loss: 0.4904561936855316, g_loss: 2.1931939125061035
Epoch [2/30], d_loss: 0.09028483927249908, g_loss: 3.413891315460205
Epoch [3/30], d_loss: 0.243388831615448, g_loss: 3.0350475311279297
Epoch [4/30], d_loss: 0.5677803158760071, g_loss: 3.8382110595703125
Epoch [5/30], d_loss: 1.2449451684951782, g_loss: 1.0350916385650635
Epoch [6/30], d_loss: 0.6785522699356079, g_loss: 2.1449031829833984
Epoch [7/30], d_loss: 0.9821871519088745, g_loss: 1.9072339534759521
Epoch [8/30], d_loss: 0.5647698044776917, g_loss: 2.2942094802856445
Epoch [9/30], d_loss: 0.7927241325378418, g_loss: 1.896308422088623
Epoch [10/30], d_loss: 0.8802512288093567, g_loss: 3.2151544094085693
Epoch [11/30], d_loss: 0.8395336866378784, g_loss: 1.9230014085769653
Epoch [12/30], d_loss: 0.8771607875823975, g_loss: 1.4895977973937988
Epoch [13/30], d_loss: 0.7876815795898438, g_loss: 1.5763006210327148
Epoch [14/30], d_loss: 0.9682697653770447, g_loss: 2.2529544830322266
Epoch [15/30], d_loss: 0.9606304168701172, g_loss: 1.8230609893798828
Epoch [16/30], d_loss: 0.957504153251648, g_loss: 1.4115766286849976
Epoch [17/30], d_loss: 0.8690584897994995, g_loss: 1.8784860372543335
Epoch [18/30], d_loss: 0.8335155248641968, g_loss: 1.38437819480896
Epoch [19/30], d_loss: 0.8494297862052917, g_loss: 1.4741439819335938
Epoch [20/30], d_loss: 0.7107457518577576, g_loss: 1.8361390829086304
Epoch [21/30], d_loss: 0.9527676105499268, g_loss: 1.1968865394592285
Epoch [22/30], d_loss: 1.121324062347412, g_loss: 2.111996650695801
Epoch [23/30], d_loss: 0.9128921627998352, g_loss: 1.7144321203231812
Epoch [24/30], d_loss: 0.8938488364219666, g_loss: 1.8418912887573242
Epoch [25/30], d_loss: 0.8258593082427979, g_loss: 1.8068771362304688
...
Epoch [28/30], d_loss: 0.9574794173240662, g_loss: 1.7942469120025635
Epoch [29/30], d_loss: 1.0254594087600708, g_loss: 1.380306363105774
Epoch [30/30], d_loss: 1.0965993404388428, g_loss: 2.1999335289001465
Training complete!
```

4. Tasks

➢ Task 1:

In this task, you will modify the GAN from Exercise 1 to generate images from CIFAR-10 dataset instead of MNIST.

<u>Submit your jupyter notebook & Add a screenshot of the result:</u>



➢ Task 2:

Solve the following questions to test your knowledge on what we learned in the lab:

1. What is the purpose of the Generator class in GAN?
A) To generate random noise for the discriminator
B) To generate images from a random noise vector
C) To classify images as real or fake
D) To calculate the loss between real and fake images

2. What is the purpose of the Discriminator class in GAN?
A) To generate random noise for the discriminator
B) To generate images from a random noise vector
C) To classify images as real or fake
D) To calculate the loss between real and fake images

3. In the code, what is the purpose of the tanh activation function in the Generator?
A) To normalize the output values to the range [0, 1]
B) To prevent overfitting during training
C) To convert the output to a binary format
D) To ensure the output values are in the range [-1, 1]

4. What is the output of the Discriminator in this code?
A) A vector of image probabilities
B) A binary value indicating if an image is real or fake
C) A 28x28 image
D) A latent vector used by the Generator

5. What does z_dim represent in the Generator class?
A) The size of the image being generated
B) The size of the input noise vector
C) The number of layers in the Generator
D) The number of channels in the output image

6. What does the term "Adversarial" refer to in the context of a Generative Adversarial Network (GAN)?
A) The components of the network act independently, where one generates data and the other passively evaluates it without competition.
B) The Generator and Discriminator work as collaborative components, improving together without any competition.
C) The two networks in the GAN setup engage in a competitive process where one aims to create realistic data and the other aims to detect it.
D) The Generator and Discriminator act as partners, where the goal is for the Generator to help the Discriminator classify fake data correctly.

7. True or False: In a GAN, the Generator and the Discriminator work independently without influencing each other to improve the overall performance of the model.
False

8. True or False: The Generator in the provided code produces images by transforming random noise into data that resembles real images from the MNIST dataset.
True

5. References

pytorch-mnist-GAN/pytorch-mnist-GAN.ipynb at master · lyeoni/pytorch-mnist-GAN · GitHub

PyTorch GAN: Generating MNIST digits

Generating MNIST Digit Images using Vanilla GAN with PyTorch