# 

CS-405 Deep Learning Fall 2023

Lab 12

**Generative Adversarial Networks (GAN) - PyTorch Tutorial**

|  |  |
| --- | --- |
| NAME | RAJA HAIDER ALI |
| CMS ID | 346900 |
| GROUP | 2 |

Date: 17th Dec 2023

# Report:

**Question 1: What does the latent\_dim variable used for in the cell below?**

**Answer:** The generator model in the GAN architecture takes a point from the latent space as input and generates a new image. The latent space itself has no meaning. Typically it is a 100-dimensional hypersphere with each variable drawn from a Gaussian distribution with a mean of zero and a standard deviation of one.

**Question 2: Please explain the architecture of the Generator part of our GAN in the cell below. Is the Generator convolutional or fully-connected?**

**Answer:** The Generator of this GAN is fully-connected, as it consists of linear layers. You can see this in the block function of the GAN i.e. layers = [nn.Linear(input\_features, output\_features)]. It takes a random latent vector as input and generates an image as output through a series of linear transformations, batch normalization, and LeakyReLU activations.

**Question 3: Please explain the architecture of the Discrinimator part of our GAN in the cell below. Is the discriminator convolutional or fully-connected?**

**Answer:** The Discriminator part is also fully-connected as the layers used are linear layers i.e. nn.Linear(int(np.prod(img\_shape)), 512) etc.

**Question 4: Why is the adversarial loss binary cross-entropy used here?**

**Answer:** Binary Cross Entropy loss is used in binary classification task. As in GAN model distinguish between real and generated samples, so BCE loss is used to effectivelty calculate the loss.

**Question 5: Answer the following questions in the cell below:**

**(i) What are the "valid" and "fake" variables initialized for?**

**(ii) What does "real\_imgs = imgs.type(Tensor)" do and why are we doing it?**

**(iii) What is the tensor z used for in the line: z = Tensor(np.random.normal(0, 1, (imgs.shape[0],latent\_dim)))**

**(iv) In the code below is the Generator updated first or the discriminator?**

**For discriminator, you can see it is a sum of two losses "real\_loss" and "fake\_loss". What do these two losses signify?**

**Answer:**

1. **valid and fake:** These are target labels for the Discriminator during training. valid variable is used when the Discriminator is learning about real images, aiming for its output to be close to 1.0. fake variable is used when the Discriminator is learning about generated images, wanting its output to be close to 0.0.
2. **real\_imgs = imgs.type(Tensor):** It is a conversion ensuring that the images (imgs) are in the right format (data type and device) for further processing in the neural network.
3. **z in z = Tensor(np.random.normal(0, 1, (imgs.shape[0], latent\_dim))):** z is a set of random numbers that are used as input for the Generator. It adds an element of randomness, ensuring the Generator produces diverse images.
4. **Generator or Discriminator update order:** The Generator is updated first. It tries to generate images that look real to confuse the Discriminator.
5. **real\_loss and fake\_loss:** These are components of the Discriminator's training. real\_loss measures how well the Discriminator recognizes real images, and fake\_loss measures its ability to detect generated images as fake. Both losses are averaged to update the Discriminator's parameters effectively.

**Question 6: How many images did the Generator produce?**

**Answer:** The generator produced 64 images because we fed it a sample\_z vector of size (64,100). This is illustrated by the below code snippet.

