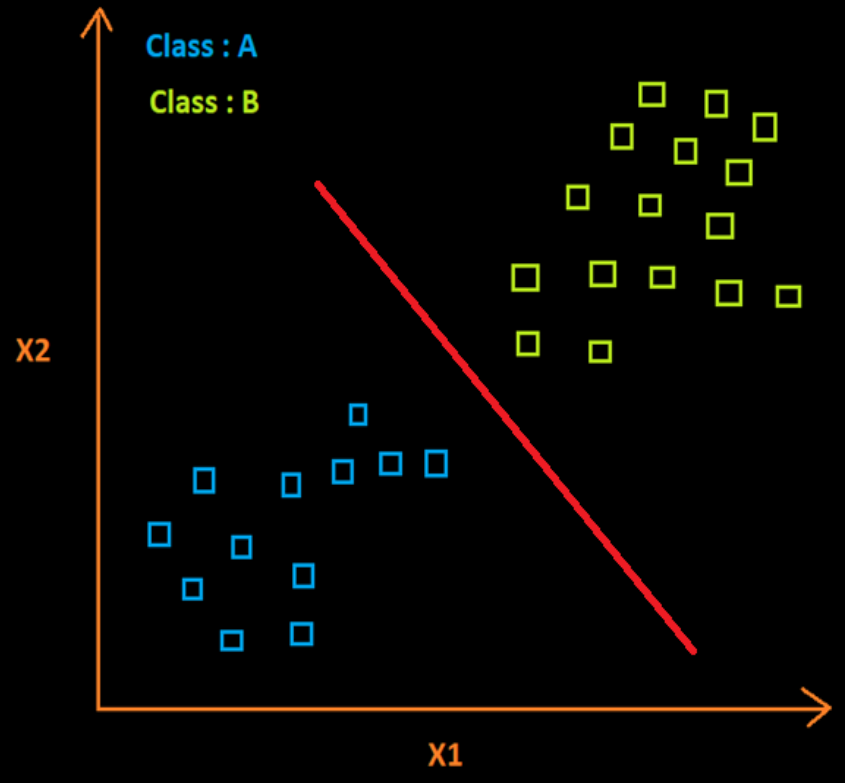
Generative Adversarial Network
(GAN)

Topics to be covered

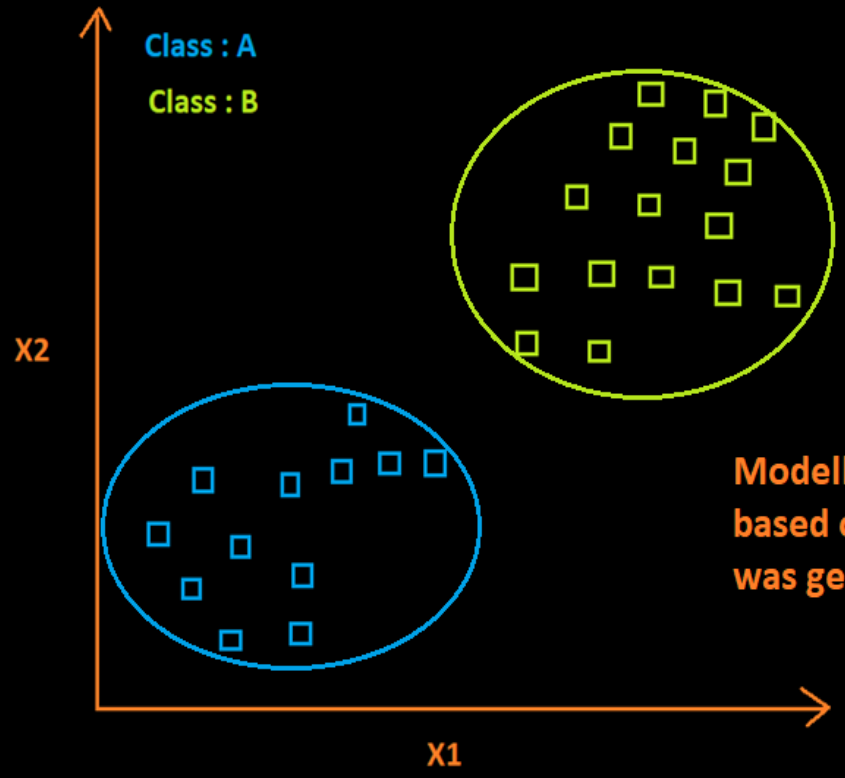
- Discriminative and Generative Models
- Working of GANs
- Project

Generative and Discriminative Models

Generative models focus on explaining how the data was generated, especially the distribution data belongs to, while discriminative models focus on predicting the labels of the data by drawing boundaries in the data space.

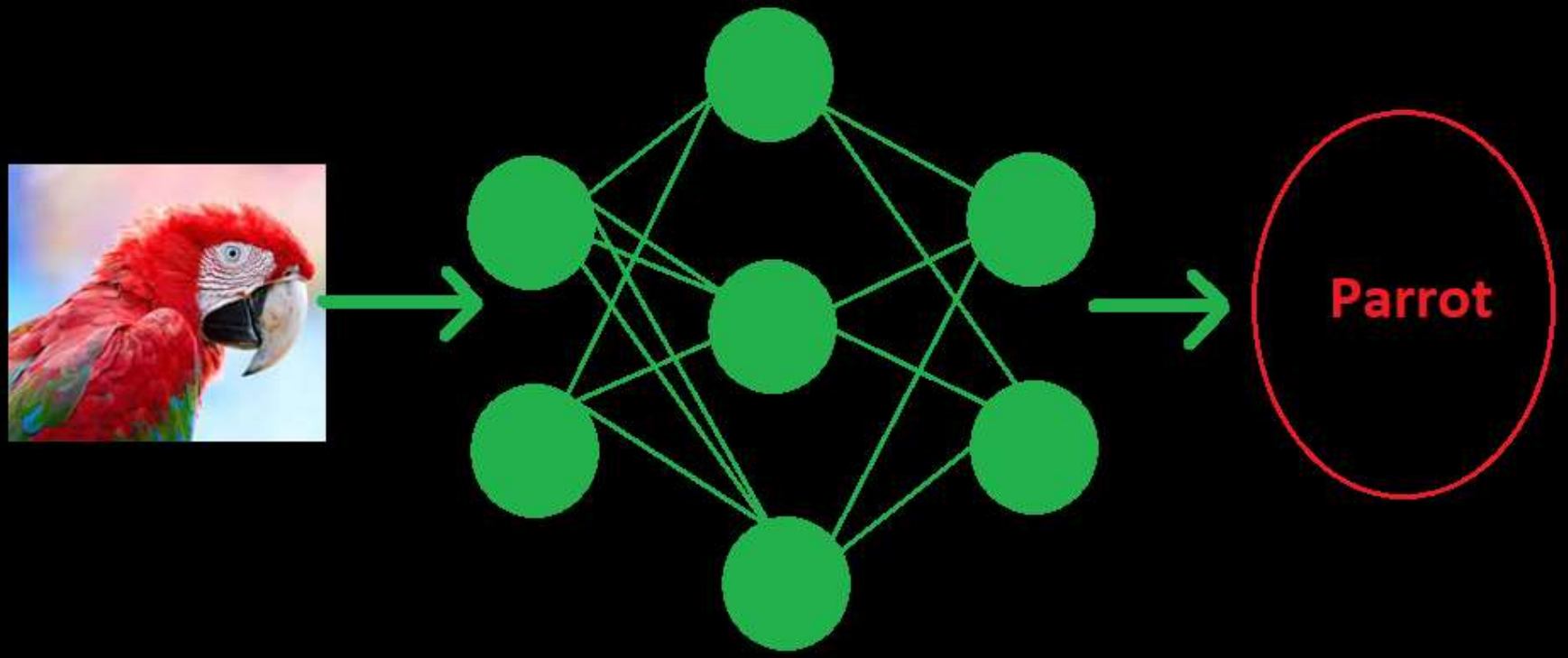


Descriminative Model

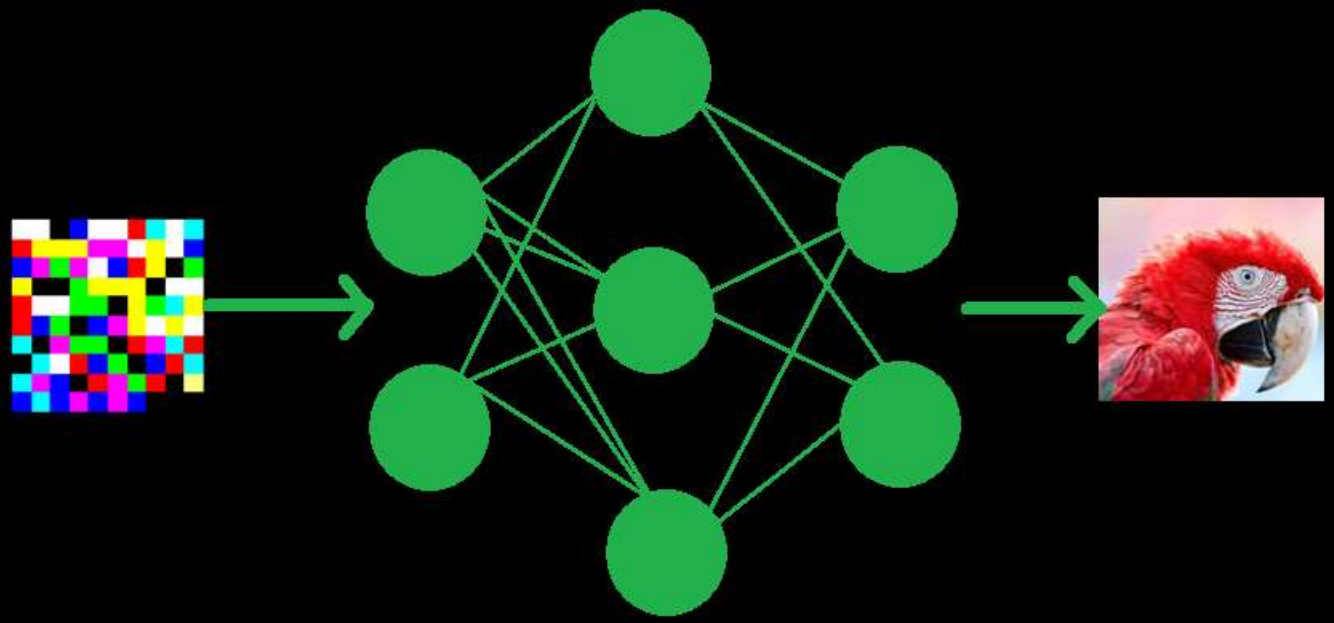


Modelling the data based on how the data was generated

Generative Model



Discriminative Model



Generative Models

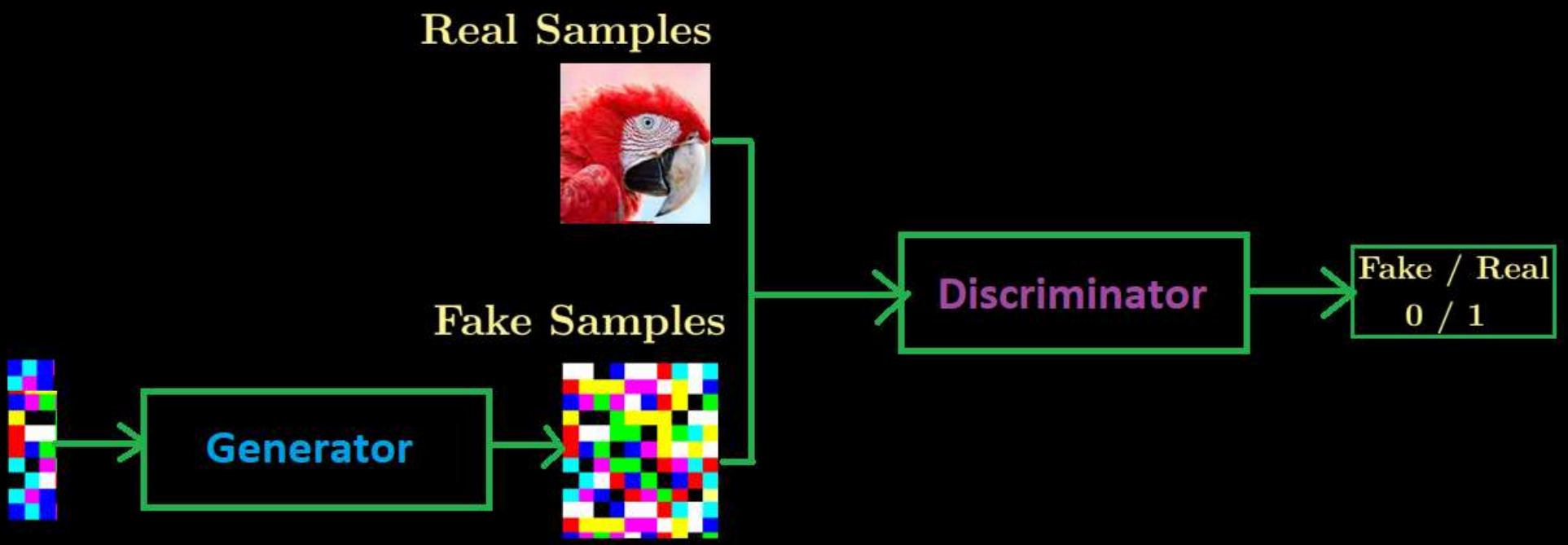
Training of GAN

Generative Adversarial Network (GAN)

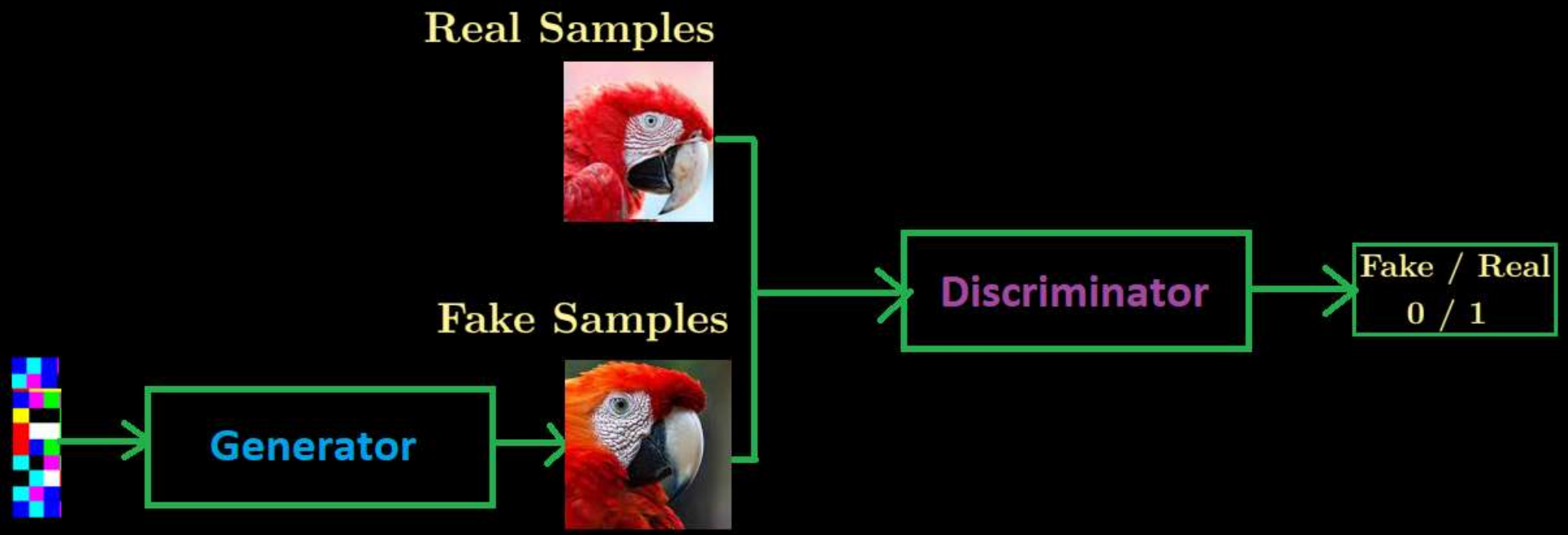
- Adversarial means competing each other. In GANs, two models i.e, generative and descriminative models are competing each other.
- Generative Model tries to generate fake data from noise and discriminative model find the difference between real and fake data.
- The generative model starts with random weights and generate a fake image and thinks that he will fool the discriminator. Then the discriminative model comes along and says that this is not a real image. Then generative has to update the weights so the discriminative model says yes.

Training Steps of GAN

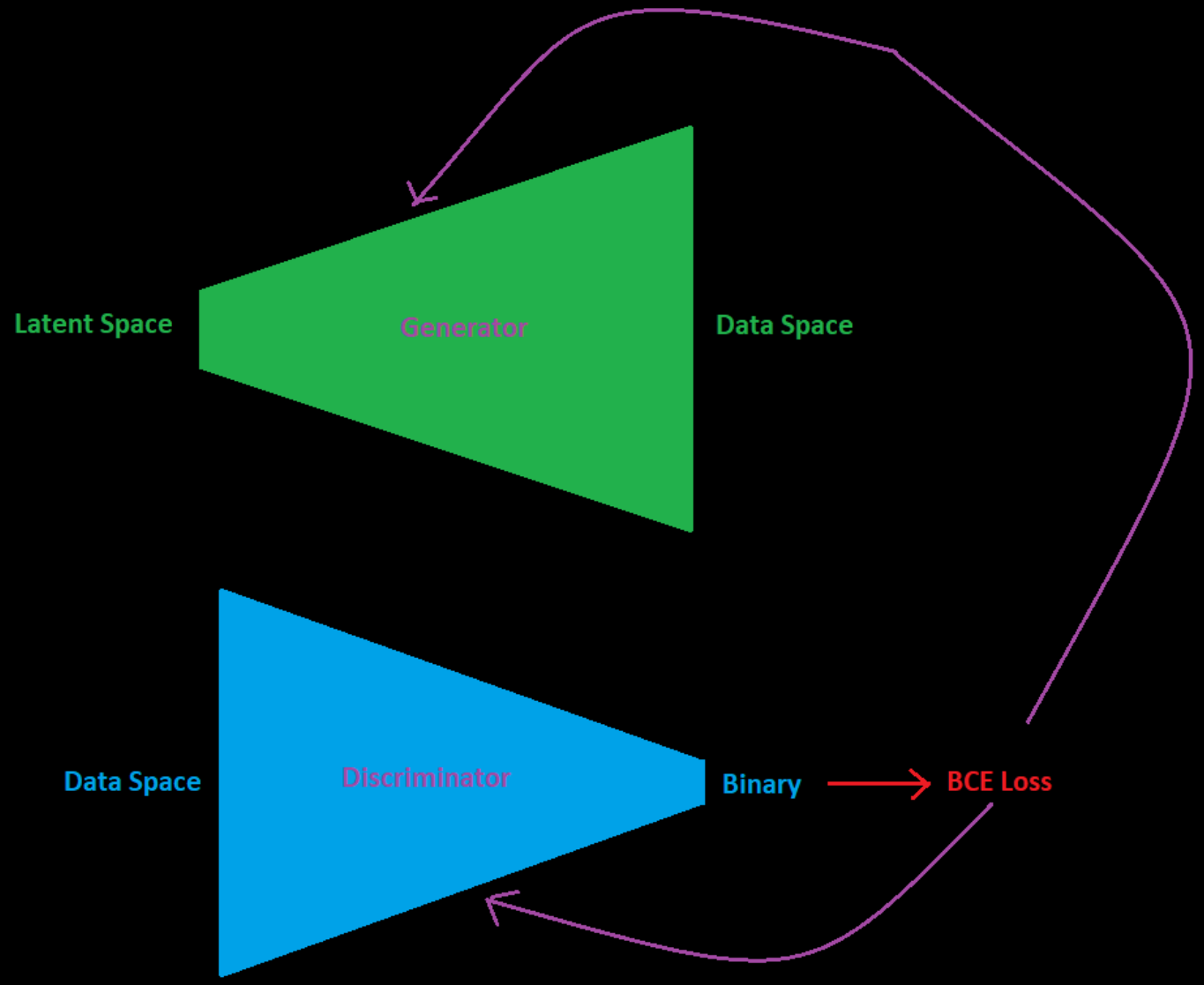
- First we train the discriminative model of the GAN. Discriminative model has two steps of training.
- In the first step of training the discriminator, we pass all real images to discriminator and compare the predictions of discriminator with real labels (all ones). The difference between the real and predicted labels is our loss which we have to minimize.
- In the second step of training the discriminator, we pass all fake images to discriminator but we want discriminator to identify them as real. When discriminator identify them as fake then this is the error which we propagate back to the generator.
- The total loss of the discriminator is the sum of the losses in both steps of training.



Training of GAN

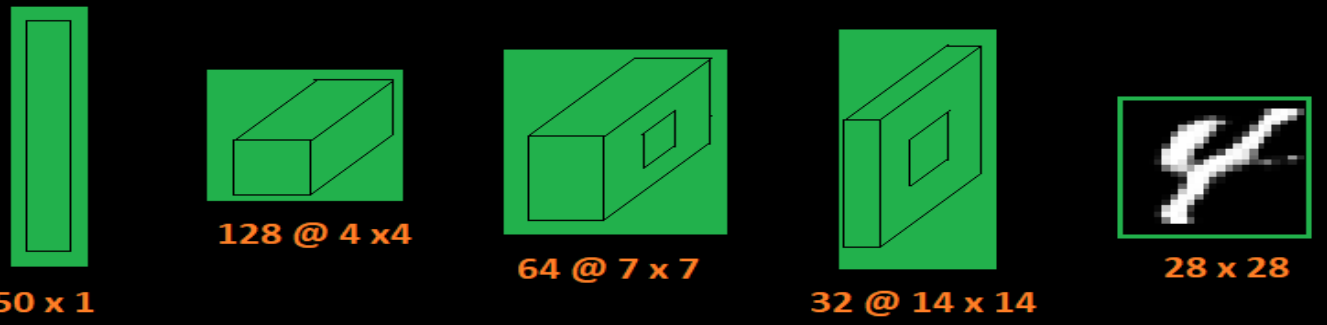


Training of GAN

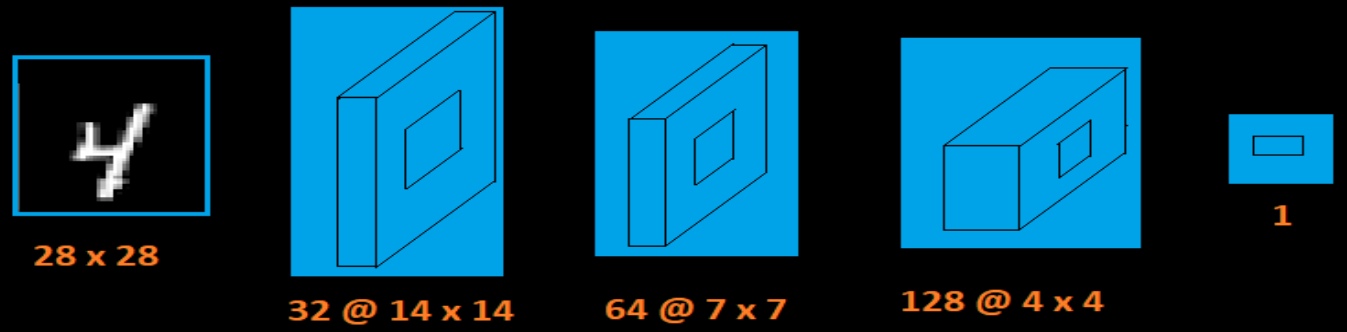

Latent Space
Generator
Data Space
Data Space
Discriminator
Binary
BCE Loss

Deep Convolutional GAN (DCGAN)

Generator



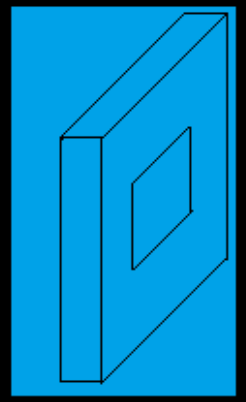
Discriminator



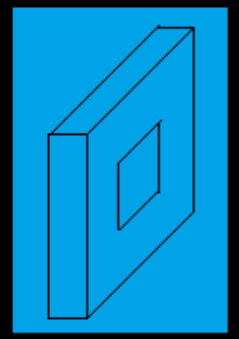
DCGAN Descriminator for 28 x 28 Images

Discriminator

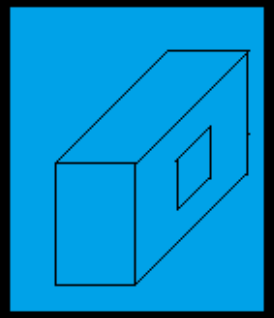
28 x 28



32 @ 14 x 14



64 @ 7 x 7



128 @ 4 x 4

1

First Convolutional Layer

$$W_{in} = 28$$

$$m = 4$$

$$padding = 1$$

$$stride = 2$$

$$W_{out} = \frac{28 - 4 + 2 * 1}{2} + 1$$

$$\boxed{W_{out} = 14}$$

Output of Conv Layer

$$W_{out} = \frac{W_{in} - m + 2 * padding}{stride} + 1$$

Second Convolutional Layer

$$W_{in} = 14$$

$$m = 4$$

$$padding = 1$$

$$stride = 2$$

$$W_{out} = \frac{14 - 4 + 2 * 1}{2} + 1$$

$$\boxed{W_{out} = 7}$$

Third Convolutional Layer

$$W_{in} = 7$$

$$m = 3$$

$$padding = 1$$

$$stride = 2$$

$$W_{out} = \frac{7 - 3 + 2 * 1}{2} + 1$$

$$\boxed{W_{out} = 4}$$

Fourth Convolutional Layer

$$W_{in} = 4$$

$$m = 4$$

$$padding = 0$$

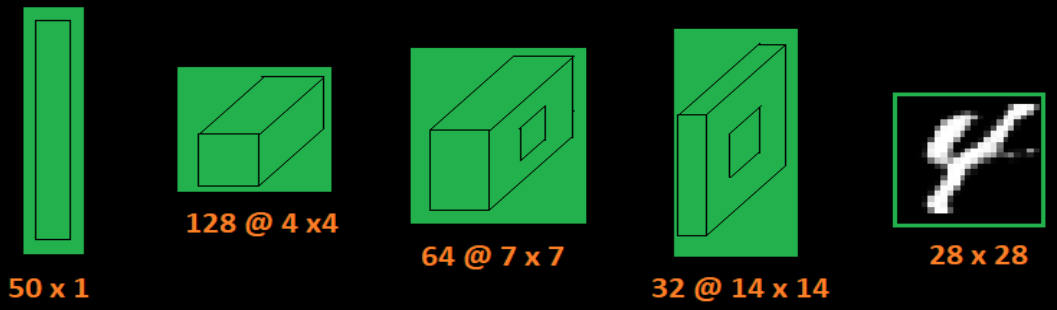
$$stride = 1$$

$$W_{out} = \frac{4 - 4 + 2 * 0}{1} + 1$$

$$\boxed{W_{out} = 1}$$

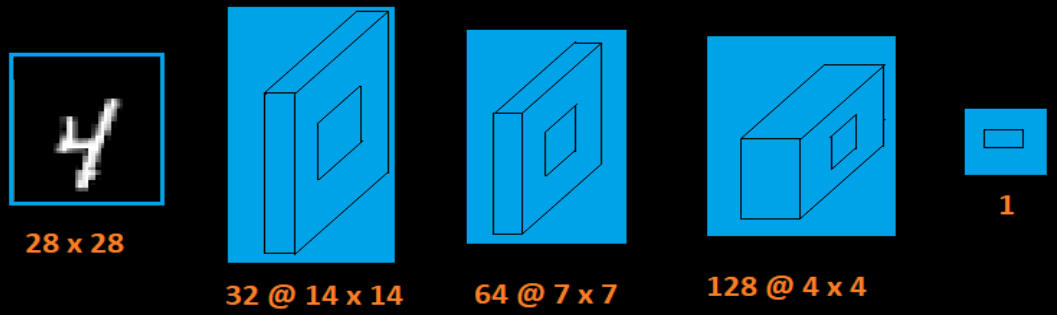
```python
class generator(nn.Module):
  def __init__(self):
    super().__init__()

    self.conv1 = nn.ConvTranspose2d(50, 128, 4, 1, 0)
    self.conv2 = nn.ConvTranspose2d(128, 64, 3, 2, 1)
    self.conv3 = nn.ConvTranspose2d( 64, 32, 4, 2, 1)
    self.conv4 = nn.ConvTranspose2d( 32, 1,  4, 2, 1)
```

Generator



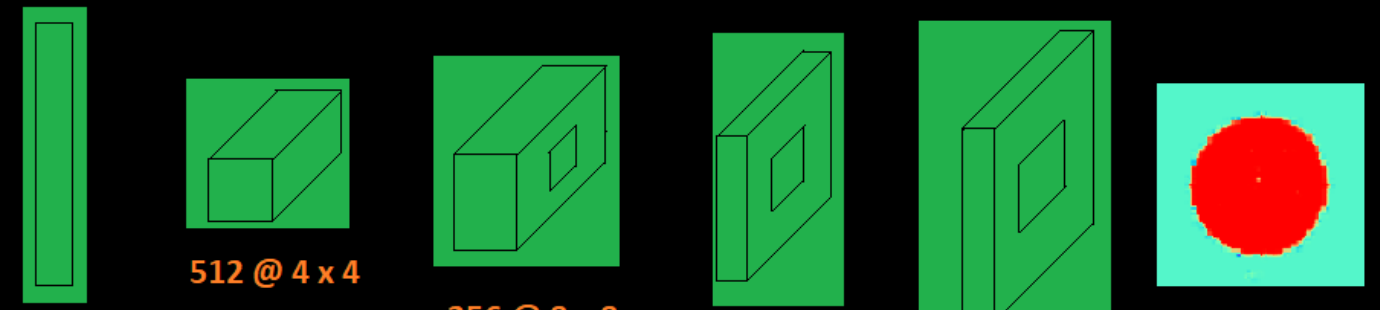

```python
class discriminator(nn.Module):
  def __init__(self):
    super().__init__()

    self.conv1 = nn.Conv2d(  1, 32, 4, 2, 1)
    self.conv2 = nn.Conv2d( 32, 64, 4, 2, 1)
    self.conv3 = nn.Conv2d(64, 128, 3, 2, 1)
    self.conv4 = nn.Conv2d(128, 1,  4, 1, 0)
```

Discriminator

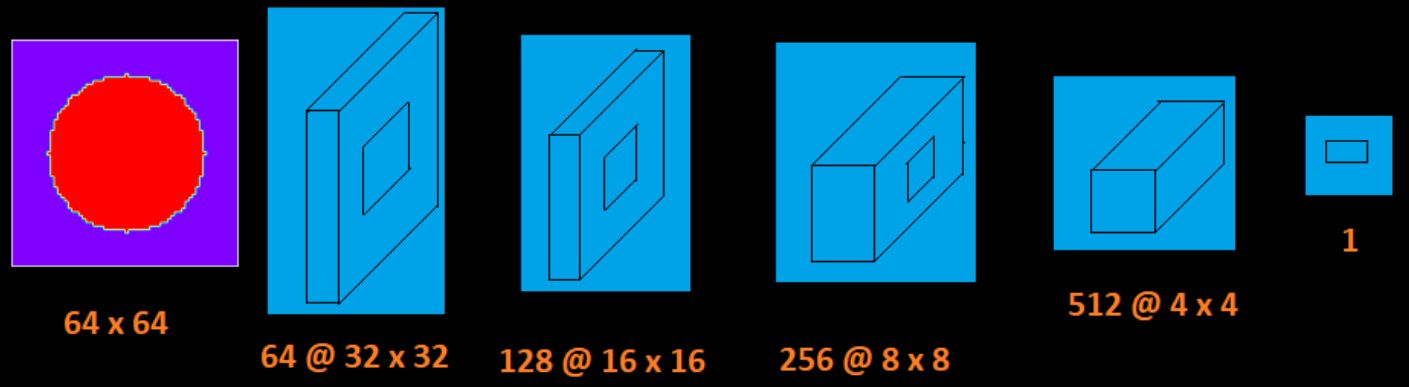


DCGAN Descriminator for 64 x 64 Images

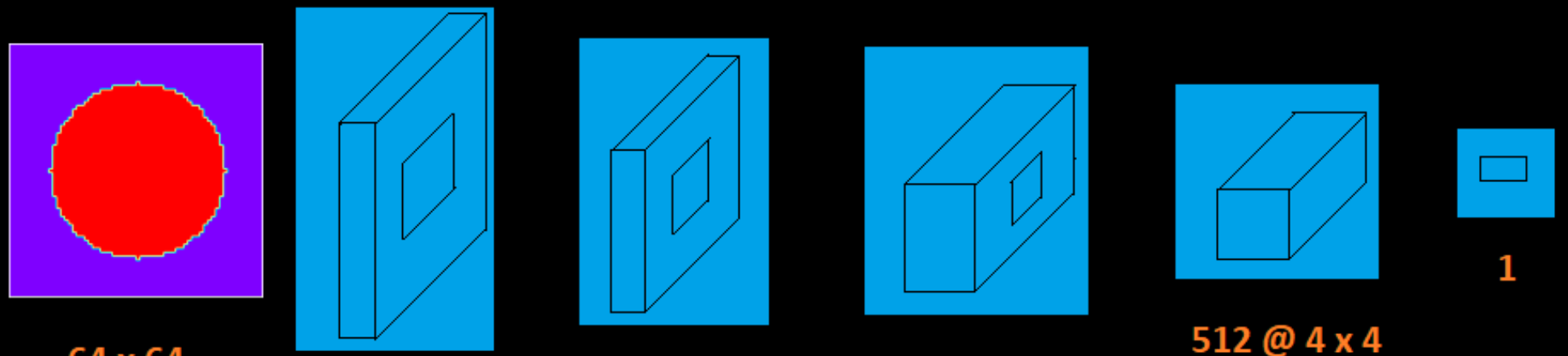
Generator



Discriminator



Discriminator



First Convolutional Layer

$$W_{in} = 64$$

$$m = 4$$

$$padding = 1$$

$$stride = 2$$

$$W_{out} = \frac{64 - 4 + 2 * 1}{2} + 1$$

$$\boxed{W_{out} = 32}$$

Second Convolutional Layer

$$W_{in} = 32$$

$$m = 4$$

$$padding = 1$$

$$stride = 2$$

$$W_{out} = \frac{32 - 4 + 2 * 1}{2} + 1$$

$$\boxed{W_{out} = 16}$$

Third Convolutional Layer

$$W_{in} = 16$$

$$m = 4$$

$$padding = 1$$

$$stride = 2$$

$$W_{out} = \frac{16 - 4 + 2 * 1}{2} + 1$$

$$\boxed{W_{out} = 8}$$

Fourth Convolutional Layer

$$W_{in} = 8$$

$$m = 4$$

$$padding = 1$$

$$stride = 2$$

$$W_{out} = \frac{8 - 4 + 2 * 2}{2} + 1$$

$$\boxed{W_{out} = 4}$$

Fifth Convolutional Layer

$$W_{in} = 4$$

$$m = 4$$

$$padding = 0$$

$$stride = 1$$

$$W_{out} = \frac{4 - 4 + 2 * 0}{1} + 1$$

$$\boxed{W_{out} = 1}$$