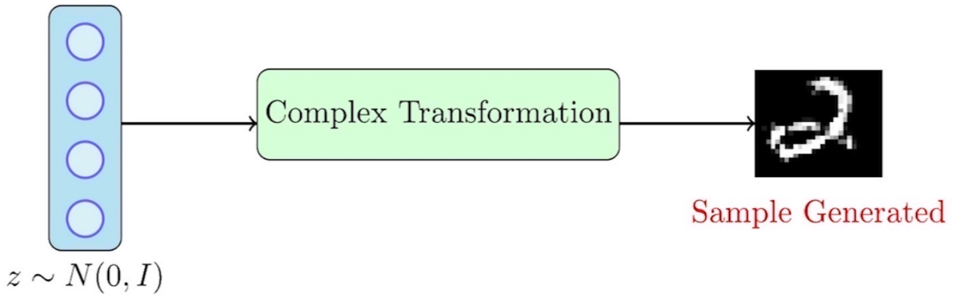
Generative Adversarial Networks (GANs)

Objective in generative models is to learn the input distribution.

Given some training data (say MNIST images), it comes from an underlying distribution.

GAN only wants to draw samples from this i/p distribution.

In other words, it wants to generate images that are similar to MNIST dataset images.



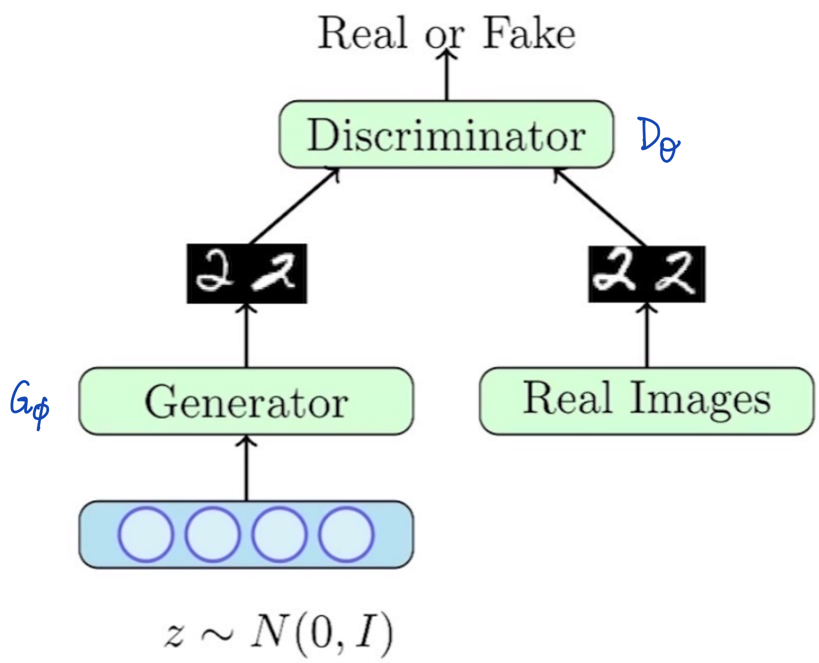
Because it is difficult to sample from the input distr., we start from a sample from normal distr. and then transform it into a sample from the input distr.

How do we get such a transformation?

- Use a Deep network and learn it.

We will use a two player game setup bet^n:

1. Generator
2. Discriminator



Job of the generator – is to produce images which look so good that the discriminator is fooled.

takes $z \sim N(0, I)$ and produce $G_\phi(z) = X \rightarrow$ image

ϕ – parameters of Generator.

Job of the discriminator – is to get better at distinguishing between true and generated images.

Takes X (image) and produces a score

$D_\theta(X) \in [0, 1]$ (Similar to Classification bet^n two Classes:

0 – Fake image

1 – True image.

Objective Function

Generator: Given some z, it wants to maximize

$$\log D_\theta(G_\phi(z))$$

$\max f(x)$
$\max \log(f(x))$

$$\max D_\theta(G_\phi(z)) \equiv \min 1 - D_\theta(G_\phi(z))$$

$$\equiv \min \log(1 - D_\theta(G_\phi(z)))$$

$$\min_\phi \mathbb{E}_{z \sim N(0,I)} \left[\log(1 - D_\theta(G_\phi(z))) \right]$$

to remove the randomness of z.

Discriminator:

$$\max_\theta \mathbb{E}_{z \sim N(0\ T)} \left[\log(1 - D_\theta(G_\phi(z))) \right] + \max_\theta \mathbb{E}_{x \sim True} [D_\theta(x)]$$

Overall Objective:

$$\boxed{\min_\phi \max_\theta \left(\mathbb{E}_{z \sim N(0\ T)} \left[\log(1 - D_\theta(G_\phi(z))) \right] + \max_\theta \mathbb{E}_{x \sim True} [D_\theta(x)] \right)}$$

for min : use gradient descent
for max : use gradient ascent