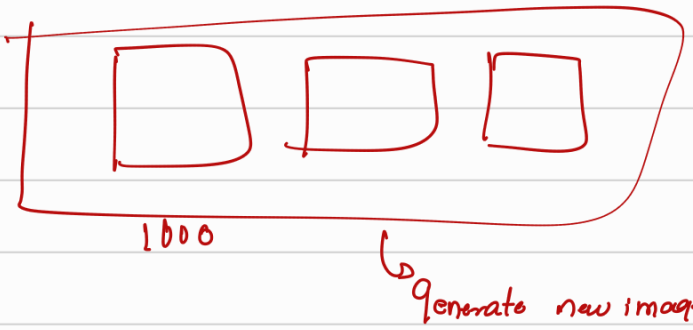
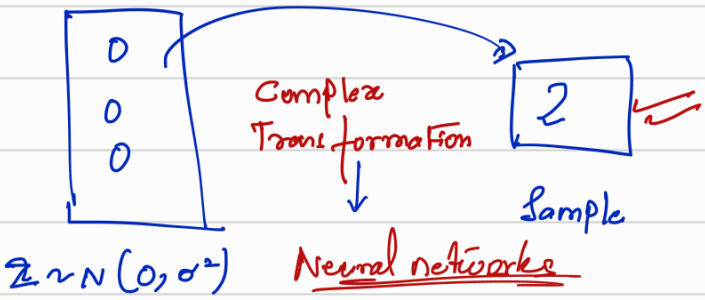
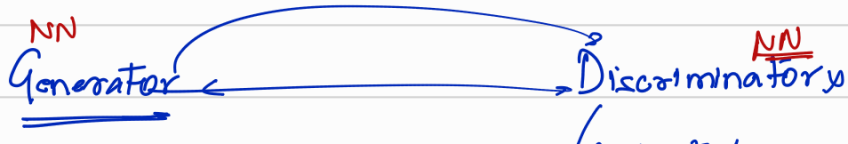
[Generative Adversarial Networks] (GANs)



(n ~~1000~~ images) → (generate new images)



given n number of mnist images, can I learn to generate more images from a simple normal dist.

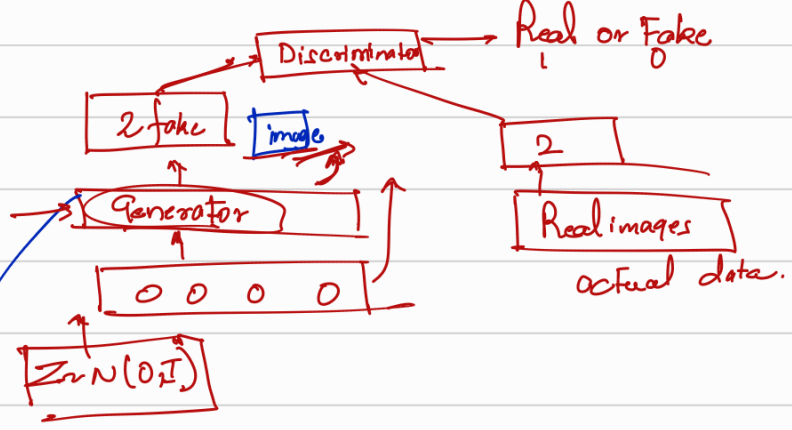


generating the images from the normal dist

differentiating the image → real or fake.

⊙ generate images such that discriminator finds it hard to discriminate

⊙ to distinguish b/w fake and real in the best possible way.



Objective function of the generator

Neural network model

image → $G_\phi(z)$ → Parameters

NN

$D_\theta(G_\phi(z))$ → Parameters

Generator's objective ⇒ $\max \left[\log\left(D_\theta(G_\phi(z))\right)\right]$

$\max \int_z p(z) \times \log D_\theta G_\phi(z)$

$\min_\theta \int_z p(z) \times \left[1 - \log\left(D_\theta G_\phi(z)\right)\right]$

(Discriminator's objective:)

$\max E \log\left(D_\theta(x)\right)$

x data → Real data

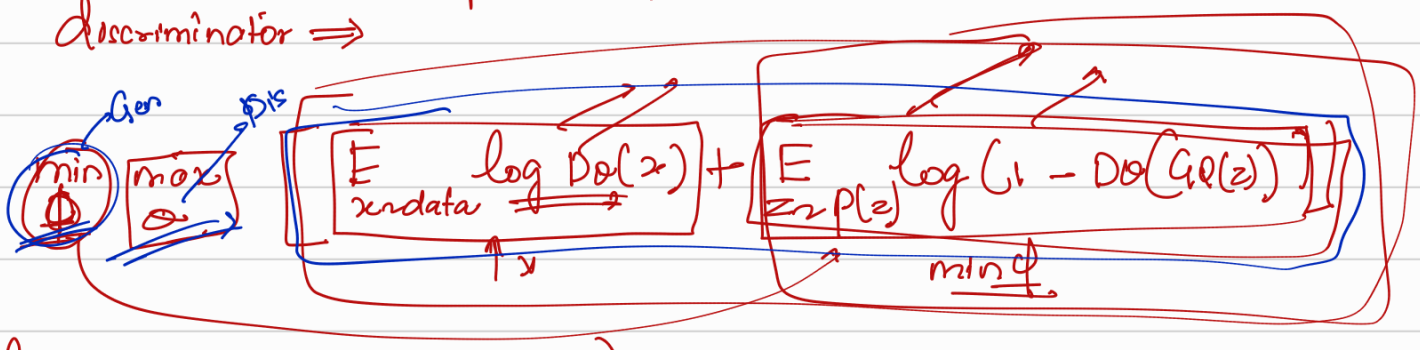
$\min \left[\log D_\theta G_\phi(z)\right)$

$\max E_{z \sim p(z)} \left[1 - \log D_\theta G_\phi(z)\right]$

$$\max_\theta \left[E_{x \sim data} \log D_\theta(x) + E_{z \sim p(z)} \log\left(1 - D_\theta G_\phi(z)\right) \right]$$

$$\max \left[\log D_\theta G_\phi(z)\right) = \min \left[1 - \log D_\theta G_\phi(z)\right]$$

Combined objective function of the Generator & the discriminator ⟹



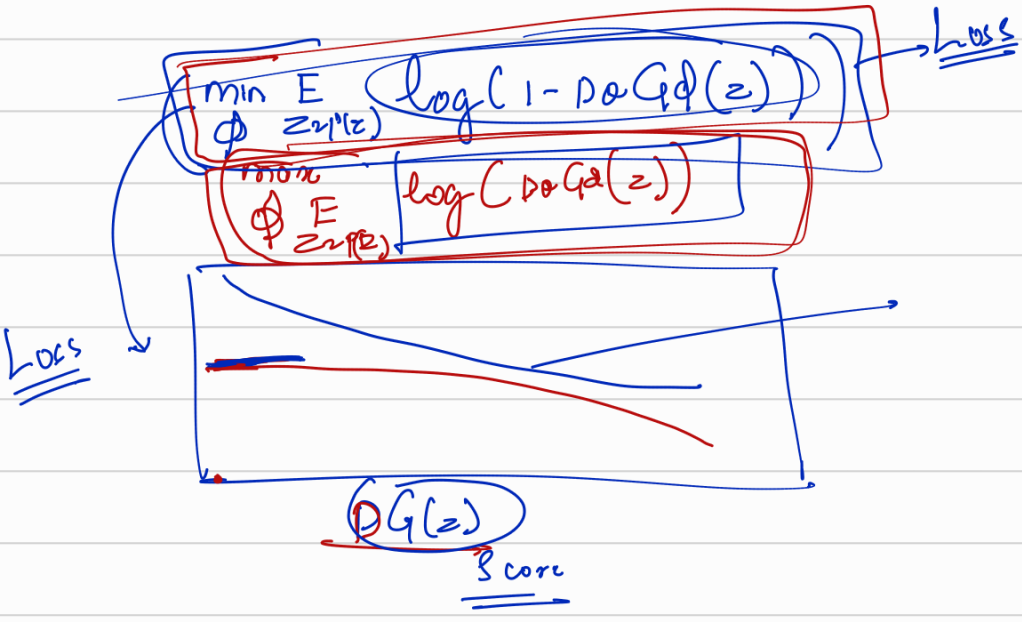
(how do we train such a model).

Step 1:

① Gradient ascent on the discriminator:

→ ① $\max_{\theta} \left[E_{x \sim p_{data}} \log D_\theta(x) + E_{z \sim p(z)} \log(1 - D_\theta G_\phi(z)) \right]$

-Gradient descent on the generator.



Procedure for GAN training.

for number of training iterations do:

for k steps:

- Sample minibatch of m noise samples $\{z^{(1)} \dots z^{(m)}\}$.
- Sample minibatch of m real examples $\{x^{(1)} \dots x^{(m)}\}$.
- Update the discriminator by ascending the sg:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_\theta(x^{(i)}) + \log\left(1 - D_\theta G_\phi(z^{(i)})\right) \right]$$

(Real) (Fake)

end for

- Sample minibatch of m noise samples $\{z^{(1)} \dots z^{(m)}\}$.

Update the generator by stochastic gradient ascent:

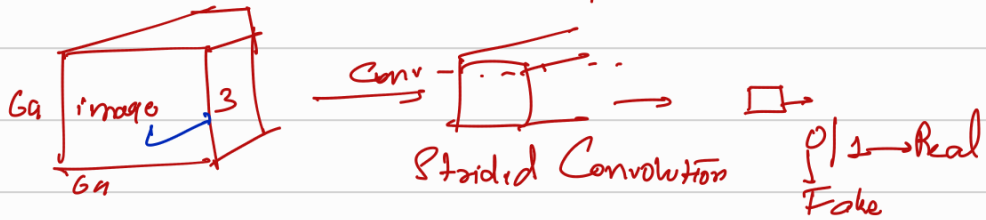
$$\nabla_{\phi} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_\theta G_\phi(z^{(i)}) \right]$$

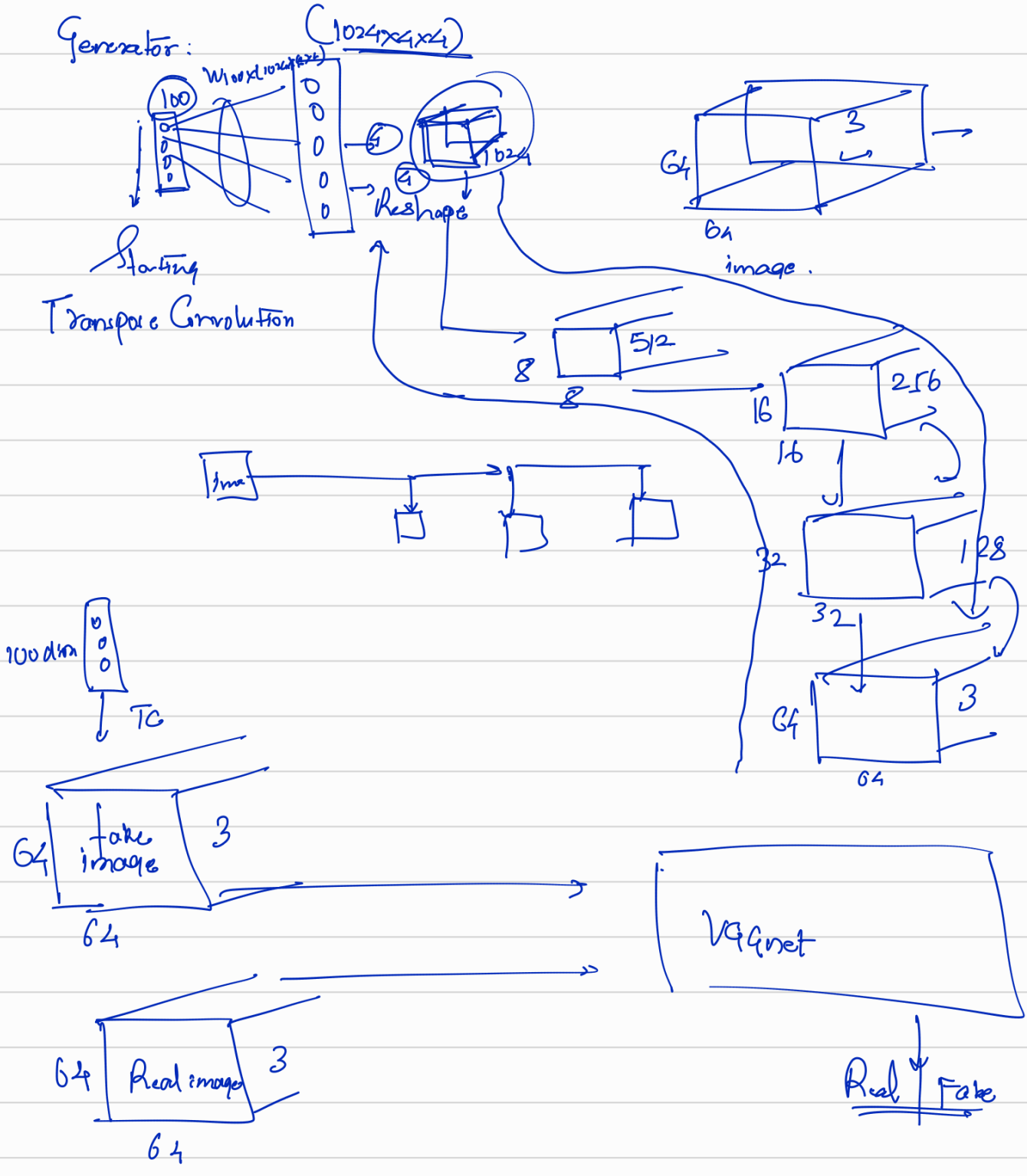
end for

DCGAN

Deep Convolution GANs

Discriminator architecture: (VGGnet / Resnet)

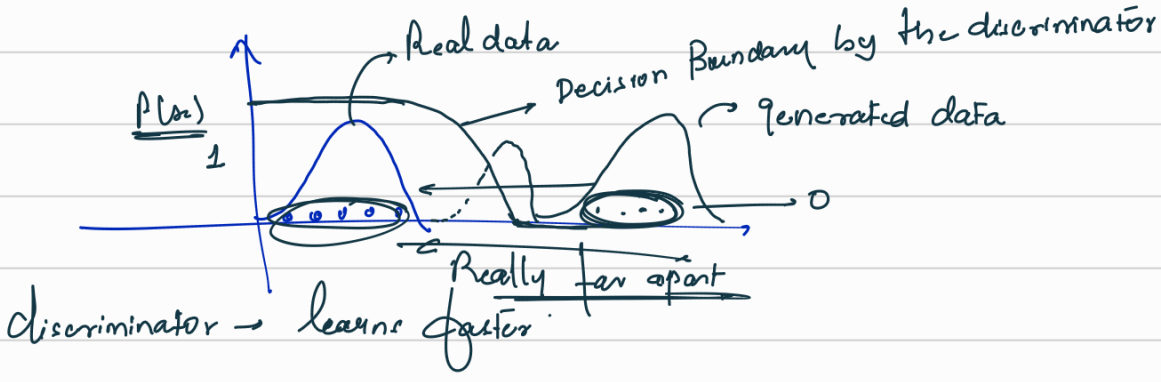




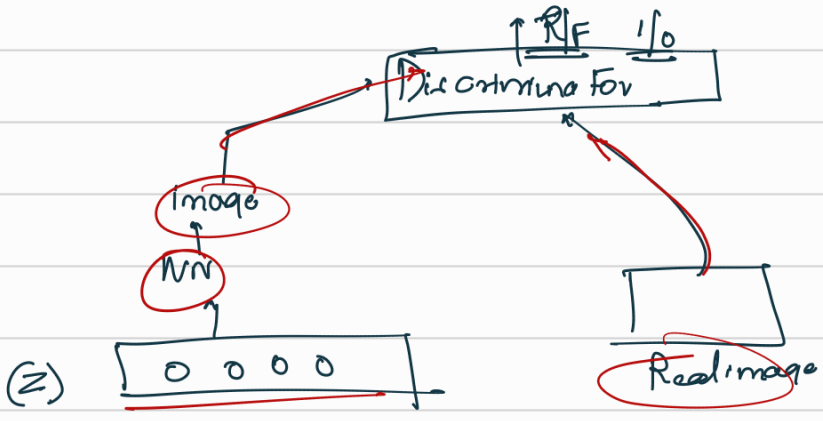
- ① any pooling layer has to be replaced with strided Convolution
- ① Use batchnorm for both the generator and the discriminator
- ① Remove any fully connected hidden layers →
- ① Use Relu activation in generator for all layers and the output is tanh
- ① Use leaking Relu activation in the discriminator for all layers.

Improved GAN training

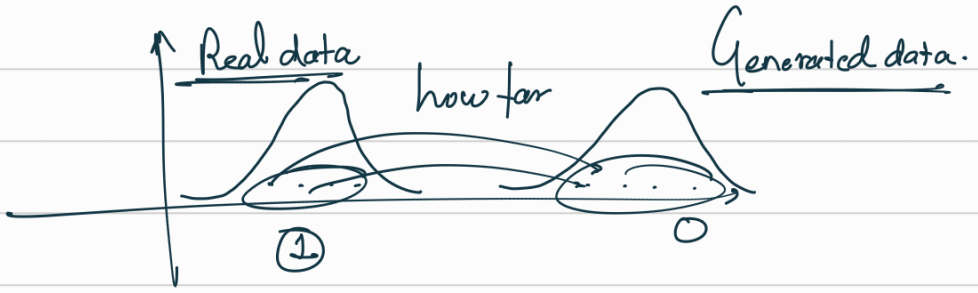
Wasserstein GAN

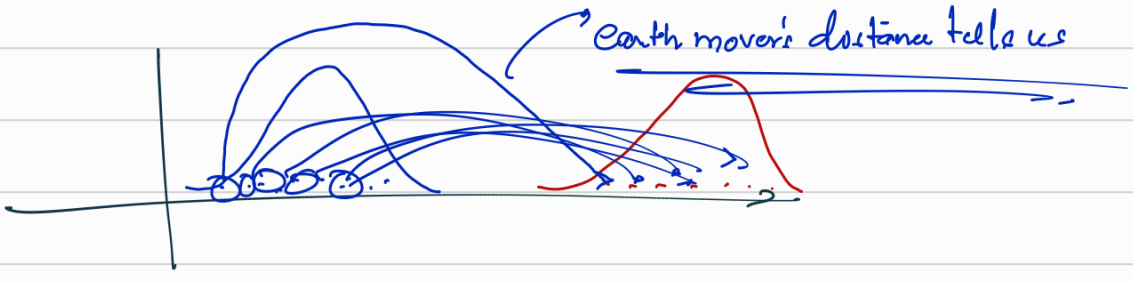


WGANs



Earth mover's distance





Changes implemented in WGAN compared to normal GAN:

① Discriminator final layer (Sigmoid) ~~was~~ removed and instead we used a linear layer.

② implementation of earth movers distance

Cool theorem → Kantorovich - Rubinstein Duality.

Lipschitz constant

$$W(p_{data}, p_g) = \left| E_{p_{data}}[f(x)] - E_{p_g(x)}[f(x)] \right|$$

$$\frac{1}{n}\sum f(x) - 1/n \sum \frac{f(x)}{p_g(x)}$$

Real data

Discriminator neural network

Discriminator Neural network

1- Lipschitz Scalar function

$$|f(x) - f(y)| \leq |x - y|$$

If I change some data in my real image, the output does not change more than 1 times of the change in the image.

$x = 2x \rightarrow$ 1 unit

Lipschitz = 2

$y_1 = 2x_1$

$y_2 = 2x_2$

$$|y_2 - y_1| = 2|x_2 - x_1|$$

$$\frac{1}{n}\left[f_\theta(x_{real})\right] - \lambda\left(\left\|\nabla_{\hat{x}} f_\theta(\hat{x})\right\|_2 - 1\right)^2 - \frac{1}{n} f_\theta[G(x)]$$

the change in the discriminator output value based on a unit change in the image norm

WGAN →

Style GANs

Code