GANs ⇒ Generative Adversarial networks:
↳ 2 n/ws competing with one another

- neural n/w trained in adversarial manner to generate data mimicking some distribution

➤ 2 players in the game:
- Generator: produce images which look so natural that discriminator thinks that images come from real data dist.
  generative algo
  inverse CNN
  i/p = random noise
  o/p = actual image
- Discriminator: to get better and better at distinguishing b/w true images & generated (fake) images

→ G_ϕ → generator D_θ → discriminator
ϕ & θ are parameters of G & D

- neural n/w based discriminator
  i/p ∈ Real X or generated $x = G_\phi(z)$ & classify: real/fake
  o/p ⇒ $D(x)$ → Real
  ⇒ $D(G_\phi(z))$ → generated
- trained on real data
- can only 2 o/ps (1 or 0) } 1 → real, 0 → fake

– discriminator assigns a score $D_\theta(G_\phi(z))$ to image generated by generator $G_\phi(z)$
– score b/w 0 & 1 → tells us prob. of it being real/fake

Obj. f^n ⇒ maximize $\log D_\theta[G_\phi(z)]$
minimize $\log(1 - D_\theta(G_\phi(z)))$ } for single z

$p(z) = \frac{1}{N} \ \forall z$ → z drawn from uniform dist.

$$\boxed{\text{Obj. } f^n \Rightarrow \min_{\phi} \sum_{i=1}^{N} \frac{1}{N} \log\left[1 - D_\theta(G_\phi(z))\right]}$$

⇒ but z is continuous & not uniform $[z \sim N(0,1)]$

$$\therefore \Rightarrow \boxed{\min_{\phi} \underbrace{\int p(z) \log(1 - D_\theta(G_\phi(z)))}_{E_{z \sim p(z)}}}$$

- task of discriminator → 2 fold
  : assign high score to real images
  : Low score to fake images

$\underbrace{\min_{\phi} \max_{\theta}}_{comb^n}$ $\boxed{\max_{\theta} E_{x\sim p_{data}} [\log D_\theta(x)] + E_{z\sim p(z)} [\log (1 - D_\theta(G_\phi(z)))]}$

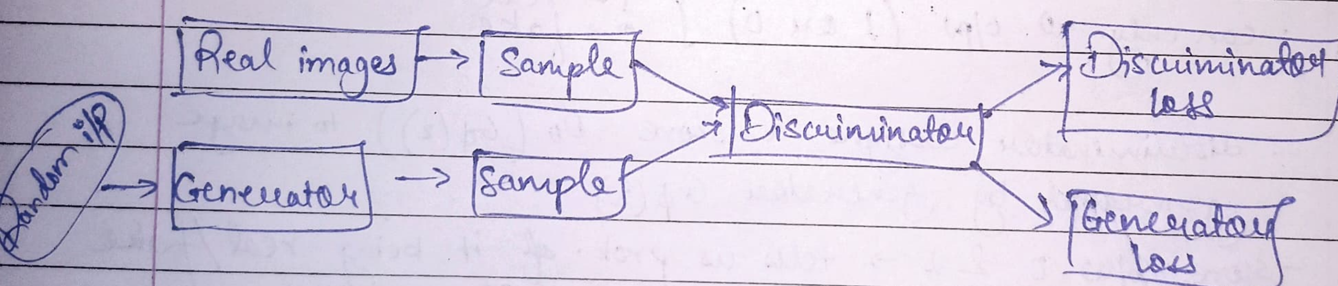
comb^n ⇒

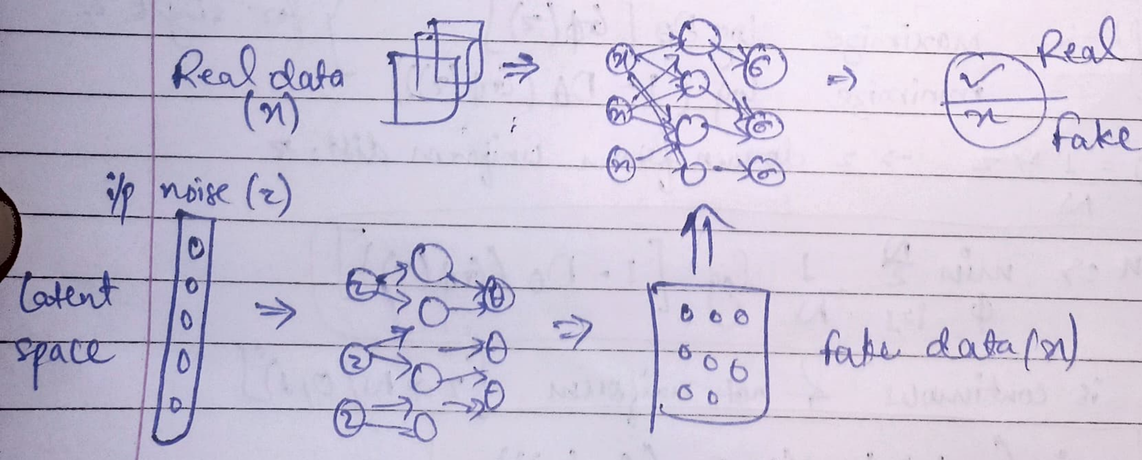
~~$\min_{\phi} \max_{\theta} [E_{x\sim p_{data}}$~~

improvement

- when sample fake, give feedback to generator using gradients
- where $D(G(z))$ is close to 0, curve of loss f^n is very flat & gradient close to 0.
- trick: instead of minimizing likelihood of discriminator being correct
  maximize likelihood of discriminator being wrong

• Architecture:





DC GANS

- Deep Convolutional GANs: popular neural n/w for Gen & Disc.
- Disc. similar to CNN with many i/p layers & 1 o/p layer
  ↳ VGG/ResNet, etc.
- Generative model G to be trained on training data x sampled from true dist. D is the one which given some std. random dist.

Architecture Guidelines: DC GANS

1) replace any pooling layers with strided convolutions (discriminator) & fractional-strided conv^n (generator)

2) use batchnorm in both gen. & disc.

3) remove fully connected hidden layers for deeper archs.

4) use ReLU actv^n in generator for all layers except for o/p → which uses tanh

5) use leaky ReLU actv^n in disc. for all layers

Issues:

1) Vanishing gradient descent problem
   → derivative of loss f^n small, update of og weights → negligible

2) mode collapse
   → when generator is able to fool discriminator with very less data

Applications:

1) Image Generation & enhancement:
- widely used to generate high-quality & realistic images, eg. creating artwork, faces, landscapes, etc.
- enhancing images: low resol^n to high, colorizing B&W pics
- img translation: satellite imgs to maps, sketches to paintings, day to night
- By generating synthetic data, researchers can augment existing datasets & improve performance

Page No.
Date

2) Deep Fake:

- GANs infamously used to create Deepfake videos,
- faces of individuals are replaced with other faces, often resulting in realistic but manipulated vids.
- facial attribute manipulⁿ, changing expressions, aging, etc.
- ethical issues: potential for misuse.

1) • Data collecⁿ: starts with collecting dataset of images of target person & source person

2) GAN is trained using these both, generates faces that resemble source person's face

3) Face swapping: during vid creation, trained GAN used to swap target person's face with source person's face.
- performed frame by frame to create a video

4) Voice manipulⁿ (optional)

5) Realistic o/p