GANs (Generative Adversarial Network) → (Ian Goodfellow + Yoshua Bengio) → reference 2, 2.

↳ (conflict or opposition)

GAN are deep neural net arch comprised of two neural networks, competing one against the other (i.e., why adversarial).

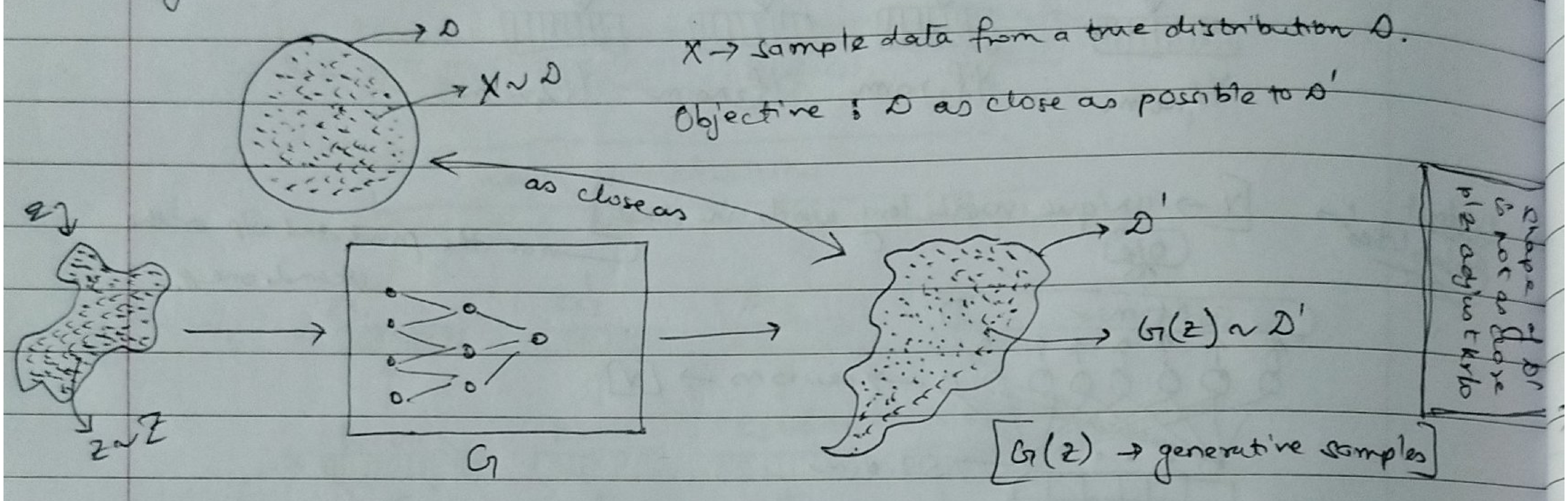
→ GAN are Neural networks that are trained in an adversarial manner, to generate data mimicking some distribution.

→ Two class of models in machine learning :-

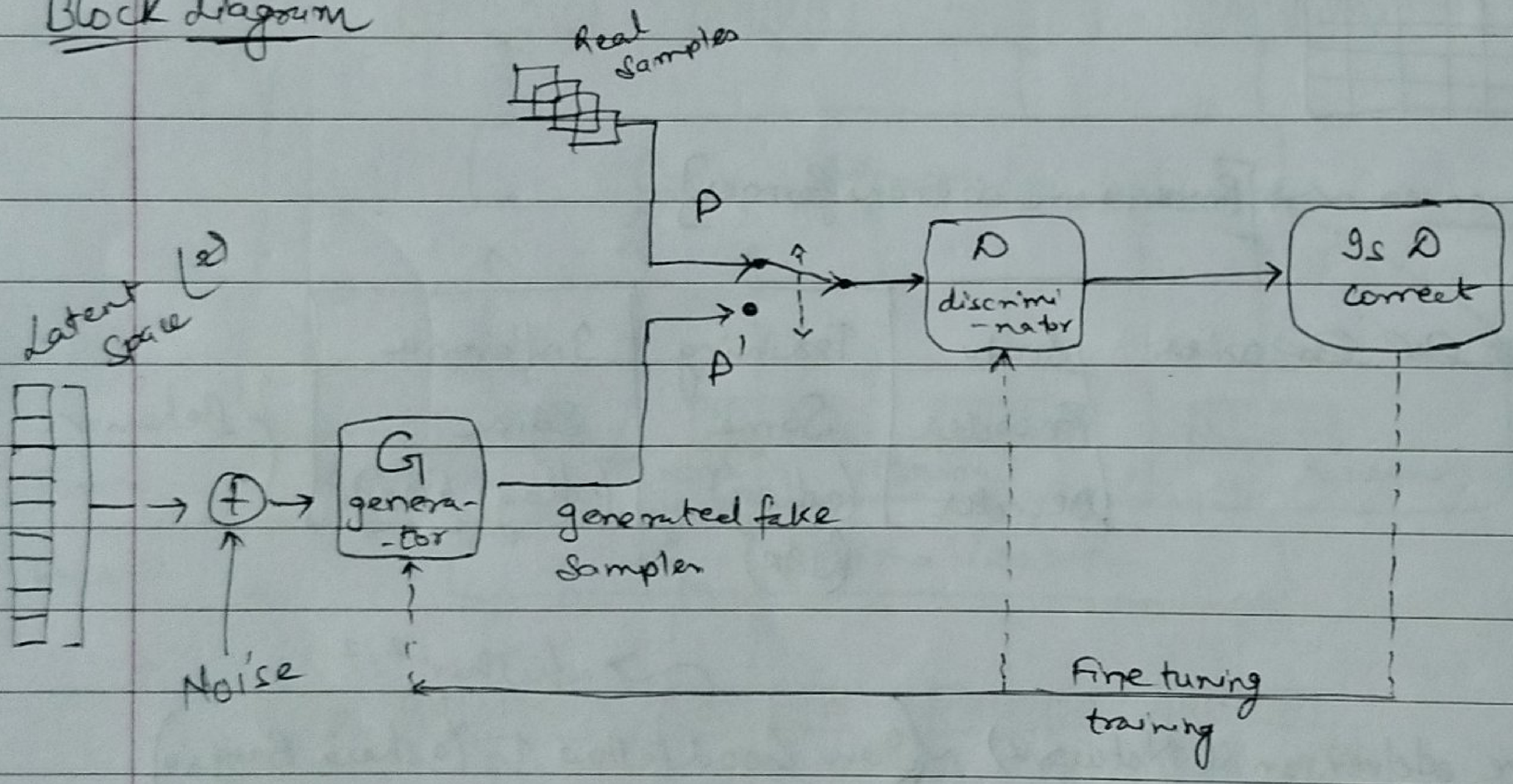
(a) Discriminative model :→ It is the one that discriminates b/w two different classes of data.

(b) Generative model :- A generative model G to be trained on training data X sampled form some true distribution D is the one which, given some standard random distribution Z produces a distribution D' which is close to D according to some closeness metric, mathematically,

$Z \sim Z$ maps to a sample $G(Z) \sim D'$



Block diagram



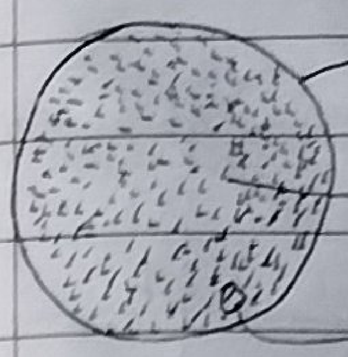
While training, we will not train our discriminator (w, b), we can't make our discriminator to strong to beat.

- Loss Function of GAN

Discriminator :- Role is to distinguish b/w the actual & fake data.

Generator :- Role is to create data in such a way so that it can fool the generator.

Page No.
Date



$p(x)$

$[\bar{z} = z]$ → small(z)

→ Specific instantiation $[X = x]$

$D(x;\theta)$ → $D(x)$ = probability that x is coming from the original dataset

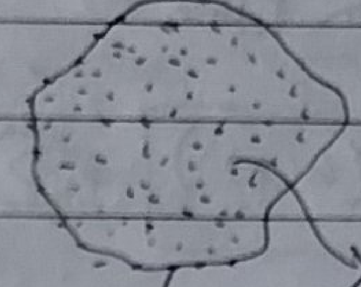
$D(x) \in [0,1]$

Random data dist.

as close as

$G_1(z;\theta_1)$

$p_z(z)$



$G_1(z) = \hat{x}$ → generated space

$p_g(x)$

$\{G, D \rightarrow$ are diff. functions$\}$

z = sample dist from Random sample

$\Rightarrow$ The closeness will come from many num of iterations

$p_g(x) = p_{data}(x)$

$\Rightarrow$ if G is fed into the D then, $D(G(z)) \in [0,1]$

The conventions :- that are used vigorously
$D(x)$, $D(G(z))$, $p_z(z)$, $p_{data}(x)$

Loss function → Binary Cross Entropy :-

$$L(\hat{y}, y) = [y \log \hat{y} + (1-y) \log(1-\hat{y})]$$

The label for the data coming from $p_{data}(x)$ is $y = 1$ & $\hat{y} = D(x)$ so putting this down will be :-

$$L(D(x), 1) = \log(D(x)) \quad \text{— (A)}$$

& for data coming from generator, the label is $y = 0$, & $\hat{y} = D(G(z))$ so in that case :-

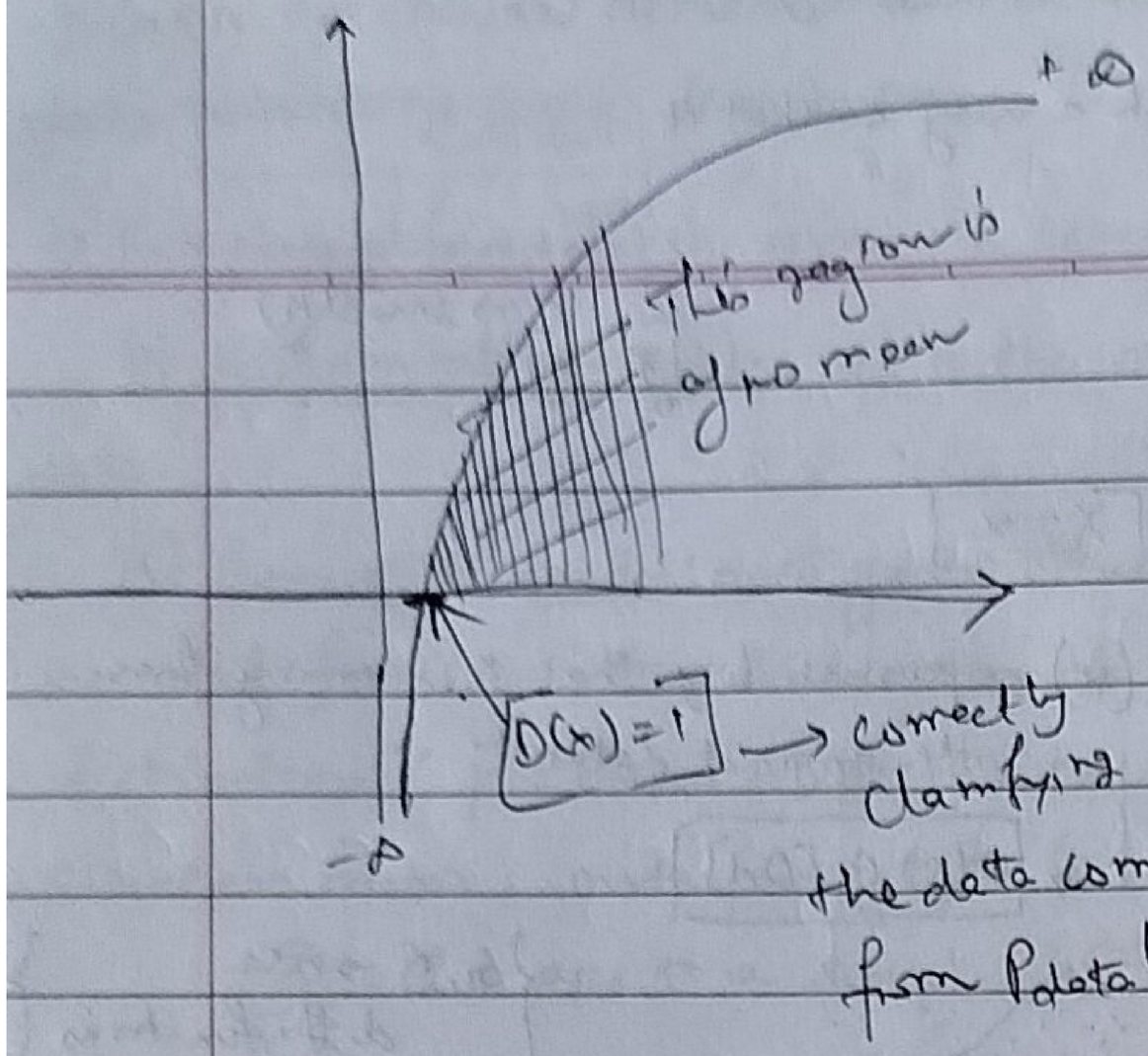
$$L(D(G(z)), 0) = (1-0) \log(1 - D(G(z))) = \log(1 - D(G(z))) \quad \text{— (B)}$$

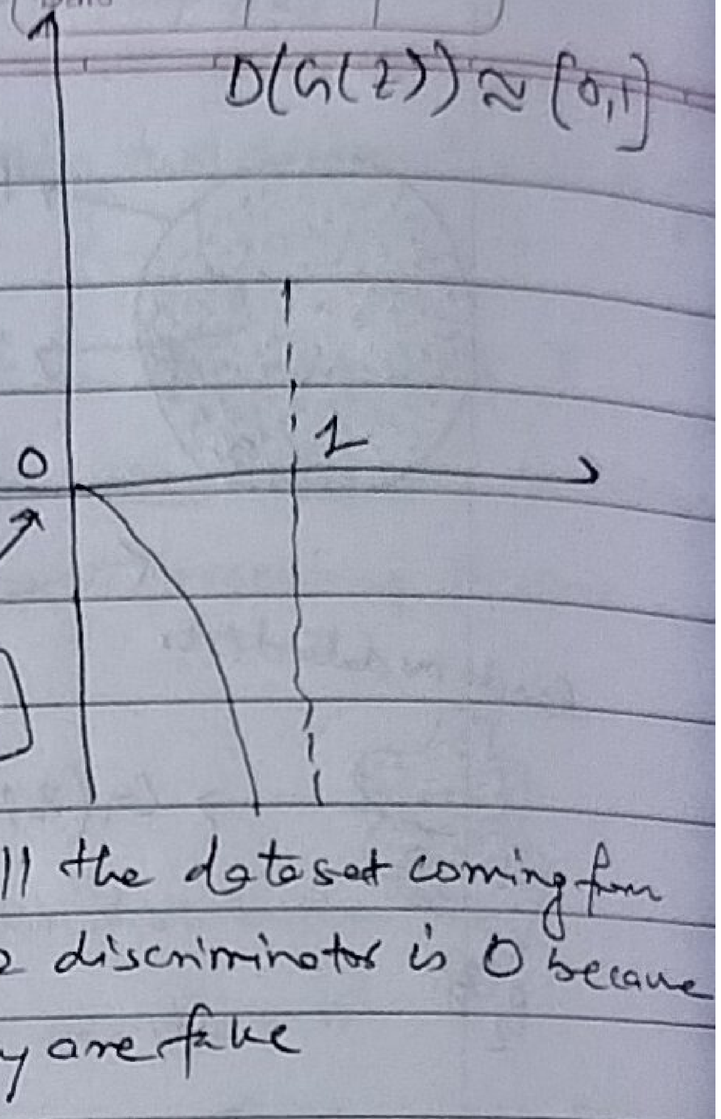
- Objective of the discriminator is to correctly classify fake vs the Real dataset.

For this (A) & (B) should be maximized

A → $[\log(D(x))]$ $[\log(1 - D(G(z))] \leftarrow B$

we need to maximize these two terms



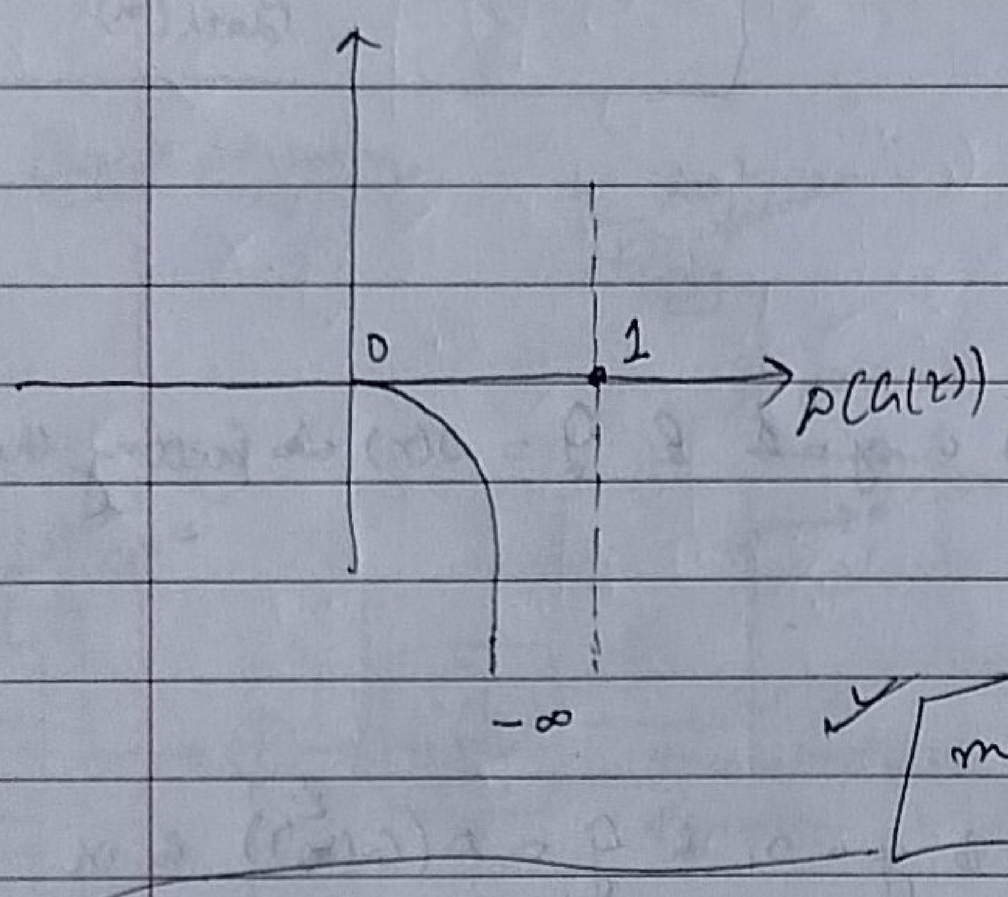


→ All the dataset coming from the discriminator is 0 because they are fake

$$\max \log(D(x)) + \log(1 - D(G(z))) \quad \text{(D)}$$

- Objective of the generator is to fool the discriminator.

$D(G(z)) = 1$ → Real Samples. , we all need to focus on the term $\log(1 - D(G(z)))$ because there is the term of generator, whereas the $\log(D(x))$ hasn't any generator term



At, $D(G(z)) = 1$ we can fool the discriminator.

We know, at $D(G(z)) = 1$, the discriminator is being fooled by generator. So at point (1) the value of $D(G(z))$ is $-\infty$ that is very less than 0. So we have to min the objective function

$$\min \left[\log(D(x)) + \log(1 - D(G(z)))\right]$$

↳ Here $(D(x))$ has no role however we are writing just to complete it. //

∴ So by minimizing we are forcing $[\log(1 - D(G(z)))]$ to reach at the value of 1, By this it can fool the discriminator.

$$\min_G \max_D \left\{\left[\log(D(x)) + \log(1 - D(G(x)))\right]\right\}$$

↳ Objective Function

↳ for only 1 instance of X.

So for all data, we need the Expected Value the final Objective func is :-

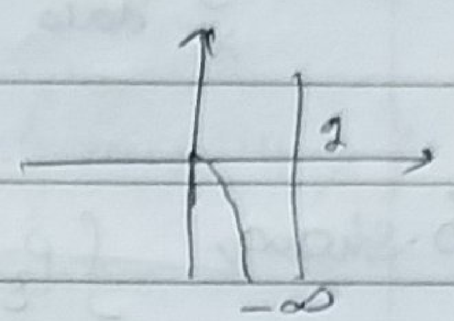
$$\min_G \max_D V(D,G) = \left\{ \underset{x \sim p_{data}(x)}{E} [\log D(x)] + \underset{z \sim p_z(z)}{E} [\log(1 - D(G(z)))] \right\}$$

Page No.
Date

(*) Generator from scratch :→

$y = 0, \hat{y} = D(G(z))$

$L(D(G(z)), 0) = (1) \log [1 - D(G(z))]$



$D(G(z)) = 1 \rightarrow$ Our main goal to fool the discriminator

$\rightarrow \left[\min \log [1 - D(G(z))] \right]$

Understanding Loss function :-

$$\min_G \max_D V(D,G) = \min_G \max_D \left[\underset{x \sim p_{data}(x)}{E} [\log D(x)] + \underset{z \sim p_z(z)}{E} [\log(1 - D(G(z)))] \right] \rightarrow (A)$$

The major thing is :- How to find the best discriminator among all the discriminator that are possible for this optimization.

According to paper :- Finding the best discriminator :-

Proposition :- For G fixed, the optimal discriminator D is :-

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

Proof :- The training criterion for the discriminator D, given any generator G is to maximize eq :→ (A)

Hence, the optimal discriminator for a given G is denoted as D_G^*.

$$\therefore \boxed{D_G^* = \underset{D}{\text{argmax}}\ V(D,G)}$$

Please Note :- $\boxed{E_{p(x)}[x] = \int_x x\, p_x(x)\, dx}$ — (1) → This is imp.

$D_G^* =$ argmax$_D$ $\left\{ \underset{x \sim p_{data}(x)}{E} [\log(D(x))] + \underset{z \sim p_z(z)}{E} [\log(1 - D(G(x)))] \right\}$ — (2)

What value D will max the argument // that's why argmax

$= \underset{D}{\text{argmax}} \{ V(G,D) \}$

Applying (1) to (2) we get

$$V(G,D) = \int_x p_{data}(x) \log(D(x))\,dx + \int_z p_z(z) \log(1 - D(G(z)))\,dz$$

In the paper it is written as :-

$$= \int_x p_{data}(x) \log(D(x))\,dx + \int_x p_g(x) \log(1 - D(x))\,dx \quad ??$$

To show, $\int_z p_z(z) \log(1 - D(G(z)))\,dz = \int_x p_g(x) \log(1 - D(x))\,dx$.

To do this we need to Recall some concepts in probability.

Recall Probability :-

If the probability density function of a random variable X is given as $P_X(x)$ it is possible to calculate the probability density function of some variable $Y = G(X)$. This is called as "change of variable" & is defined as

Pic rep

$$P_Y(y) = P_X(G^{-1}(y)) \left| \frac{d}{dy}(G^{-1}(y)) \right|$$

X, $P(x)$ → G → $Y(G(x))$, $P_Y(y)$??

→ G is the ~~transformation~~ function //

In our case

Z, $P(z)$ → G → $X = G(Z)$

$$P_g(x) = P_z(G^{-1}(x)) \frac{d}{dx}(G^{-1}(x))$$

mapping all the x are replaced by z, and all the y are replaced by x

To prove/show. $\int_z p_z(z) \log(1 - D(G(z)))\,dz = \int_x p_g(x) \log(1 - D(x))\,dx$

$z \sim p_z(z)$ → G → $x = G(z)$

assume G is invertible $\Rightarrow z = G^{-1}(x)$

Let's start from LHS

$$\text{or, } \int_z P_z(z) \log(1 - D(G(z)))\, dz = \int_x P_z(G^{-1}(x)) \log(1 - D(x))\, dG^{-1}(x)$$

Page No.
Date

$$= \int_x P_z(G^{-1}(x)) \log(1 - D(x)) \frac{dG^{-1}(x)}{dx} \cdot dx$$

Using relation $P_g(x) = P_z(G^{-1}(x)) \frac{d}{dx} G^{-1}(x)$ {derived earlier}

$$= \int_x P_z(G^{-1}(x)) \log(1 - D(x)) \frac{dG^{-1}(x)}{dx} dx \quad \{\text{rewriting}\}$$

$$= \int_x \left\{ P_z(G^{-1}(x)) \frac{dG^{-1}(x)}{dx} \right\} \log(1 - D(x))\, dx$$

$$= \int_x P_g(x) \log(1 - D(x)) \cdot dx \quad \text{— (3)} \quad \underline{\text{Proved}}$$

So, as we derived this which has the same equation in the paper

$$\text{i. } V(G, D) = \int_x P_{data}(x) \log(D(x))\, dx + \int_x P_g(x) \log(1 - D(x))\, dx$$

Clubbing both integrals now,

$$V(G, D) = \int_x \left[P_{data}(x) \log(D(x)) + P_g(x) \log(1 - D(x)) \right] dx$$

The optimal D^* for a given G is obtained by maximizing the argument $V(G, D)$.

So, we will find maximum of the integrand & choose the value of at maximum value to be optimal value of D given G, i.e. D_G^*

$$\text{So, } \frac{d}{dD(x)} \left[P_{data}(x) \log(D(x)) + P_g(x) \log(1 - D(x)) \right] = 0$$

$$\Rightarrow \left\{ \frac{P_{data}(x)}{D(x)} - \frac{P_g(x)}{1 - D(x)} = 0 \right\} \quad \text{— (4)}$$

$$\boxed{D_G^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}} \quad \boxed{\text{proved}} \quad \text{(optimal value)}$$

We got the optimal one, now we need to get that whether it is maximum or not!!! to get that we need to do one more derivative 4

To prove it is a maximum value, we differentiate eqn (4)

$$\frac{-p_{data}(x)}{(D(x))^2} - \frac{p_g(x)}{(1-D(x))^2} < 0$$ Hence maximum

value is achieved for D given G as,

$$D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$ optimal as well as the maximum value //

↳ we have calculated the value of D, tho it is not practically possible but mathematically it is important. Why not practically? Because the $p_{data}(x)$ is non-apnon'.

↓

not based on theory, but rather on exp & observation

} It is mathematically important, because its existence enables to prove an optimal value of generator. //

The role of generator is to reverse of that of D. i.e. min so the optimal G. that minimize the loss function occurs when $D = D^*_G$. Also we get optimal G^* as

$$G^* = \arg\min_G V(D^*_G, G)$$

(At this point, we must show that the optimization problem stated in (A) has a unique solution G^* & this solution satisfies $p_g = p_{data}$.

So, our optimal $D^*_G = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$

By substituting optimal value of D^*_G in $G^* = \arg\min_G V(D^*_G, G)$ we obtain.

$$= \arg\min_G \left\{ \int_x \{ p_{data}(x) \log(D^*_G(x)) + p_g(x) \log\{1 - D^*_G(x)\} \} dx \right\}$$

$$= \arg\min_G \left[\int_x \left\{ p_{data}(x) \log\left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right) + p_g(x) \log\left(\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right) \right\} dx \right]$$

Add & subtract $(\log 2)\, p_{data}(x)$ & $(\log(2))\, p_g(x)$ in above eqn

we obtain :-

$$G^* = \int_x \left[\{(\log 2 - \log 2)\, p_{data}(x) + p_{data}(x) \log \left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) \right. +$$

$$\left. (\log 2 - \log 2)\, p_g(x) + p_g(x) \log \left(\frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) \right\} dx \Bigg]$$

$$G^* = \int_x \left[-\log 2 \left(p_{data}(x) + p_g(x) \right) + p_{data}(x) \left\{ \log 2 + \log \left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) \right\} \right.$$

$$\left. + p_g(x) \left\{ \log 2 + \log \left(\frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) \right\} dx \right]$$

$$G^* = -\log_2 \int_x \left[p_g(x) + p_{data}(x) \right] dx + \int_x p_{data}(x) \left[\log 2 + \log \left(\frac{p_{data}(x)}{p_g(x) + p_{data}(x)} \right) \right]$$

$$+ \int_x p_g(x) \left[\log 2 + \log \left(\frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) \right] dx.$$

(*) $\int_x p(x)\, dx = 1$, $\log \left(\frac{p_{data}(x)}{\left(p_{data}(x) + p_g(x) \right)/2} \right)$, $\log \left(\frac{p_g(x)}{\left(p_g(x) + p_{data}(x) \right)/2} \right)$

$$G^* = -\log 2 (1+1) + \underbrace{\int_x p_{data}(x) \log \left(\frac{p_{data}(x)}{\left(p_{data}(x) + p_g(x) \right)/2} \right)}_{KL\ Divergence} +$$

$$\underbrace{\int_x p_g(x) \log \left(\frac{p_{data}(x)}{\left(p_{data}(x) + p_g(x) \right)/2} \right)}_{KL\ Divergence}.$$

$$G^* = \arg\min_G \left\{ -\log 4 + KL \left[p_{data}(x) \,\middle\|\, \frac{p_{data} + p_g}{2} \right] + \right.$$

$$\left. KL \left[p_g(x) \,\middle\|\, \frac{p_{data} + p_g}{2} \right] \right\}.$$

must be wondering there is no G at RHS !, But p_g is dependent on G. see (prob.'s derivation)

$$G^* = \arg\min_G \{ -\log 4 + 2\, JSD(p_{data}^{(x)} \,||\, p_g^{(x)}) \}$$

↳ Jensen Shanon Divergence

we know, $p_g(x) = \int_x \{ p_z(G^{-1}(x)) \frac{d G^{-1}(x)}{dx} \}$ ∴ $p_g(x)$ is independent on G

Here, Jensen Shanon Divergence is :-

$$JSD = \frac{1}{2}\{ KL(P||M) + KL(Q||M) \} \text{ where } M = \frac{P+Q}{2}$$

∴ As we are minimizing that term if $p_g(x) = p_{data}^{(x)}$ then the JSD will be 0 (whole term)//

min value achieved by generator will be :- $[-\log(4)]$

$$G^* = \arg\min_G \{ -\log 4 + 2\, JSD(p_g(x) \,||\, p_{data}^{(x)}) \}$$

$$\Big\| \; 0 \rightarrow \text{if } p_g(x) = p_{data}^{(x)}$$

So, JSD is 0 only when $p_g(x) = p_{data}^{(x)}$ which minimizes the argument & the value obtain is $-\log 4$ //

Theorem :- The global minimum of the criterion $G^* = \arg\min_G V(D_G^*, G)$ is achieved if & only if $p_g = p_{data}$, at that point G^* value is $-\log 4$

Proof :- the goal of GAN process is to have $p_g(x) = p_{data}^{(x)}$, so substituting these values for optimal D results in $D_G^* = 1/2$

NOTE :- while we were deriving we got the solution, $p_g(x) = p_{data}^{(x)}$, We didn't said that directly //

Let's go with Goodfellow idea //

Recalling,

$$V(G,D) = \underset{G}{\operatorname{argmin}} \left\{ \int_x \left[p_{data}(x) \log(D(x)) + p_g(x) \log(1 - D(x)) \right] \right\} dx$$

Page No.
Date / /

Substituting, $D(x) = 1/2$ we get,

$$V(G, D_G^*) = \int_x \left[p_{data}(x) \log 1/2 + p_g(x) \log 1/2 \right] dx = -\log 2 \int_x \left[p_{data}(x) + p_g(x) \right] dx$$

$$V(G, D_G^*) = -2\log 2 = -\log 4 \quad \left[\because p_{data}(x) = p_g(x) \text{ & } \left\{ \int_x p(x)\,dx = 1 \right\} \right]$$

Optimization of Loss function

Lets Recall the loss function:→

$$\mathcal{L}_2 \min_G \max_D \left\{ \underset{x \sim p(x)_{data}}{E} [\log D(x)] + \underset{z \sim p_z(z)}{E} [\log(1 - D(G(z)))] \right\}$$

In paper:→ Algorithm 1 → they showed that they are using Mini batch Gradient Descent

① Taking m fake sample (y=0) & m real samples (y=1)

② Update the discriminator, to update the W & B of the discriminator network i.e. the neural network (by ascending)

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

θ_d → Parameters involved in discriminator

$\frac{1}{m}\sum_{i=1}^{m}$ → Expectation over m samples

$x^{(i)} \to p_{data}(x)$

$z^{(i)} \to p_g(z)$

These are coming from two different data, one from $p_{data}(x)$ & one from $p_g(z)$

③ Train the generator seperately.

↳ again, take m noise samples from $p_g(z)$

↳ Update the generator (by descending) using stochastic Gradient

$$\nabla_{\theta_g} \left(\frac{1}{m} \sum_{i=1}^{m} \left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right)$$

θ_g → Parameters involved in Generator

→ Expectation over (m) samples.

Overall

$$\max_D \left\{ \underset{x \sim p(x)_{data}}{E} [\log D(x)] + \underset{z \sim p_z(z)}{E} [\log(1 - D(G(z)))] \right\}$$

$$\min_G \left\{ \underset{z \sim p_z(z)}{E} [\log(1 - D(G(z)))] \right\}$$

Vanishing Gradient problem

As the weights and bias of Discriminator and generator has their individual.

Page No.
Date

so, we will represent the parametes of discriminator as ϕ and the parametes of the generator as θ.

ods by derivatives if (the loss func) gets to the θ, so lets see :→

$$\mathcal{L} = \min_{G} \max_{D} \; \mathbb{E}_{x \sim p_{data}(x)} \left[\log D_\phi(x)\right] + \mathbb{E}_{z \sim p_z(z)} \left[\log\left(1 - D(G_\theta(z))\right)\right]$$

↳ depends on generator only

$$\nabla_\theta \mathcal{L} = \nabla_\theta \left[\mathbb{E}_{z \sim p_z(z)} \log\left(1 - D_\phi(G_\theta(z))\right)\right]$$

$p_z(z)$ → Random distribution

$$= \mathbb{E}_{z \sim p_z(z)} \nabla_\theta \left[\log 1 - \left(D_\phi(G_\theta(z))\right)\right]$$

$$= \mathbb{E}_{z \sim p_z(z)} \left[\frac{\partial G_\theta(z)}{\partial \theta} \cdot \frac{\partial D_\phi(G_\theta(z))}{\partial G_\theta(z)} \left\{\frac{1}{\left[D(G_\theta(z)) - 1\right]}\right\}\right]$$

$$= \mathbb{E}_{x \sim p_g(x)} \quad x = G(z)$$

$$= \mathbb{E}_{x \sim p_g(x)} \left[\frac{\partial x}{\partial \theta} \frac{\partial D(x)}{\partial x} \cdot \frac{1}{(D(x) - 1)}\right]$$

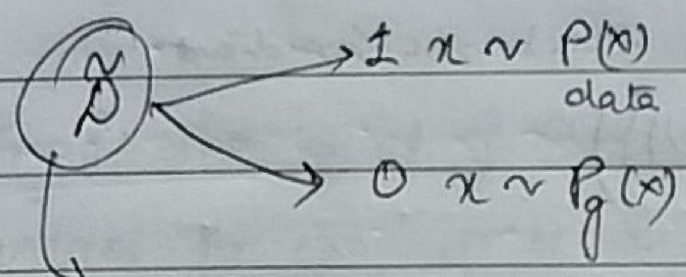
$$= \mathbb{E}_{x \sim p_g(x)} \left[\frac{1}{(D(x) - 1)} \cdot \frac{\partial D(x)}{\partial x} \cdot \frac{\partial x}{\partial \theta}\right] \quad \{\text{Rewriting}\}$$

$\nabla_x = \frac{\partial}{\partial x}$

$\nabla_\theta = \partial/\partial\theta$

$$= \mathbb{E}_{x \sim p_g(x)} \left\{\frac{1}{(D(x) - 1)} \cdot \nabla_x D(x) \cdot \nabla_\theta x\right\}$$

We are training the generator, after the training of the discriminator.



if D tends to $\tilde{D}$ & $x \sim p_g$ (fake samples) then,

$$\lim_{D \to \tilde{D}} D(x) = 0$$

$$\lim_{D \to \tilde{D}} \nabla_x D(x) = 0$$

Lets put this into our eqn

$$\nabla_\theta \mathcal{L} = \lim_{D \to \tilde{D}} \mathbb{E}_{x \sim p_g(x)} \left\{\frac{\nabla_x D(x) \; \nabla_\theta x}{(D(x) - 1)}\right\} = 0$$ (Vanishing Gradient)

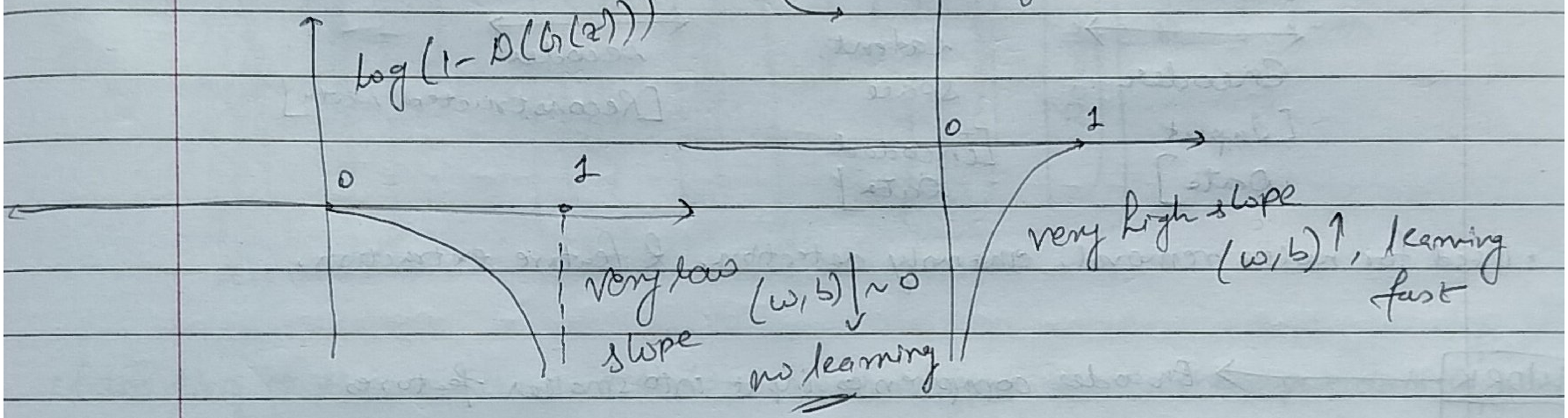
This is more of the mathematical Reasoning for our discussion of Vanishing Gradients. as we seen it turn out to be 0, which will not help in the back propagation on the w & b of the generator and the generator will produce fake samples and it will not learn at all (which are not close as real sample)

So to solve this just see the cost function :-

$$\Rightarrow \min_G \; E_{z \sim P_z(z)} \left[\log \left(1 - D_\theta G_\theta(z) \right) \right]$$

↳ To avoid this situation we will change the func.

$$\max_G \; E_{z \sim P_z(z)} \left[\log \left(D(G(z)) \right) \right]$$



- Mode Collapse :- During the training process, the generator may collapse to a setting where it always produces same output. This is called mode collapse.

Code :→ GitHub