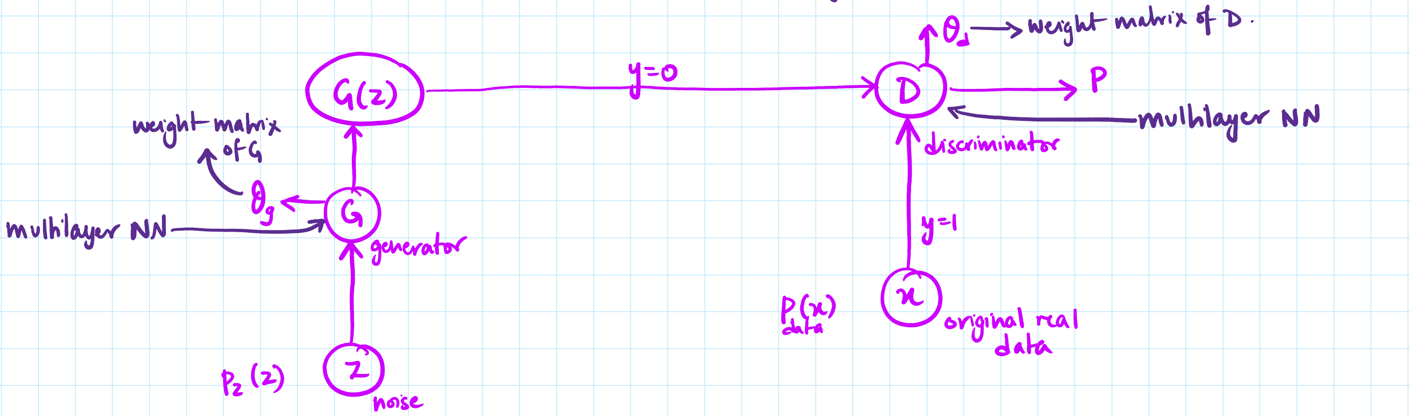
Generative Adversarial Network

→ neural networks trained in adversarial manner to generate data that mimicks some distribution.

→ Two things to understand: Basic scheme; Loss function.



Binary Cross Entropy Function: $\boxed{Z = -\sum y\ln\hat{y} + (1-y)\ln(1-\hat{y})}$

Real Data. When $y=1$, $\hat{y} = D(x)$: $L = \ln(\hat{y})$

when $y=0$, $\hat{y} = D(G(z))$ (noise): $L = \ln[1-D(G(z))]$

So for any sample: $L = \ln[D(x)] + \ln[1-D(G(z))]$

↓

Now we need this for all samples.
→ Expectation.

$$\mathcal{E}(L) = \mathcal{E}\left(\ln[D(x)]\right) + \mathcal{E}\left(\ln[1-D(G(z))]\right)$$

$$= \int P_{data}(x)\cdot \ln[D(x)]\,dx + \int P_z(z)\cdot \ln[1-D(G(z))]\,dz$$

$$V(G,D) = \mathcal{E}_{x\sim P_{data}}[\ln(D(x))] + \mathcal{E}_{z\sim P_z}[\ln(1-D(G(z)))]$$

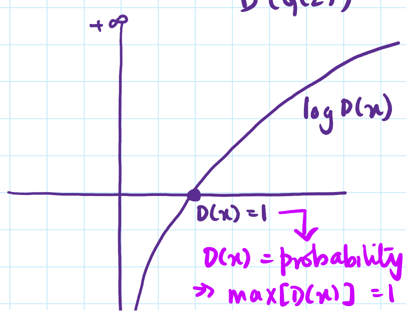
Now, the objective of the discriminator is to classify real v. fake.

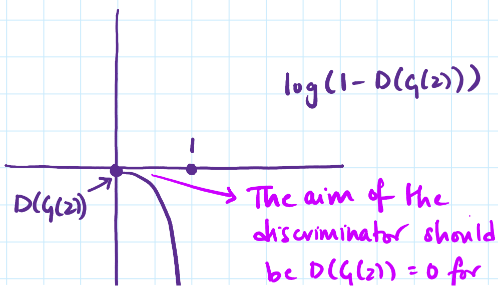
Real: $\ln(D(x))$ — A
discriminator taking real as i/p.

Fake: $\ln(1-D(G(z)))$ — B
discriminator taking data generated by generator w/ noise z as i/p

$D(x)$ = probability of a label associated w/ real data x.

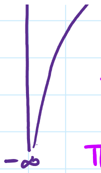
$D(G(z))$ = probability of a label associated w/ fake data z.





The aim of the discriminator should be $D(G(z)) = 0$ for

Namita Achyuthan

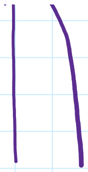


$D(x) = 1$

$D(x)$ = probability

$\Rightarrow \max[D(x)] = 1$

This is what discriminator should do for real data.



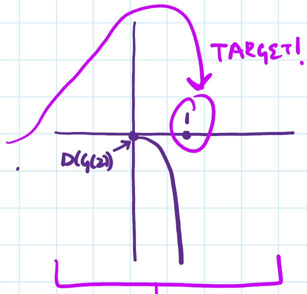
$D(G(z))$

→ The aim of the discriminator should be $D(G(z)) = 0$ for fake data.

So, $D = \max\{\ln(D(x)) + \ln(1 - D(G(z)))\}$

Discriminator: I'm going to be so perfect @ differentiating real + fake
real gets 1.
fake gets 0.

Generator: haha I'm going to fool the discriminator.
I'm going to make $G(z)$.
When I pass to discriminator, $D(G(z)) = 1$.
She'll think it's real data!



Now, the generator will achieve the target at $\boxed{-\infty}$

$\min[\log(D(x)) + \log(1 - D(G(z)))]$ — So now we're to minimize this objective function so that we get a really tiny value.

$\log(D(x))$: This guy has NO role.

$\log(1-D(G(z)))$: Here, we should be getting $D(G(z))$ as close to 1 as possible.

$\Rightarrow D(G(z)) \approx 1$

$\Rightarrow$ produce sample close to real data

So now let's put it together.

$$\min_G \max_D \{\ln(D(x)) + \ln(1 - \underbrace{D(G(z))}_{\text{get this as close to 1 as possible}})\}$$

So now when we apply over all samples, we get:

$$\min_G \max_D V(D,G) \equiv \min_G \max_D \{ \sum_{x \sim P_{data}(x)} [\log D(x)] + \sum_{z \sim p_z(z)} [1 - \log D(G(z))] \}$$

Finding the optimal discriminator:

$$V(G,D) = \int P_{data}(x) \log(D(x))\,dx + \int P_z(z) \log(1 - D(G(z)))\,dz$$

$$= \int P_{data}(x) \log(D(x))\,dx + \int P_g(x) \cdot \log(1 - D(x))\,dx.$$

How did that happen?
→ prob of random variable X is $P_x(x)$
→ can calculate PDF of $Y = G(X)$

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

In the case of a generator:

$Z \xrightarrow{P_z(z)} \boxed{G} \longrightarrow X = G(z)$

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

also assume G is invertible.

① Now we fix G and find optimal D_G^*.
→ derivative wrt $D(x)$ — discriminator network.

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

$$\frac{P_{data}(x)}{D(x)} - \frac{P_g(x)}{1 - D(x)} = 0$$

$$\frac{P_{data}(x)}{D(x)} - \frac{P_g(x)}{1-D(x)} = 0$$

$$D_G^*(x) = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}$$

② To get max value, check 2nd derivative.

$$-\frac{P_{data}(x)}{(D(x))^2} - \frac{P_g(x)}{(1-D(x))^2} < 0$$

NOW.

③ The optimal value of a generator is the REVERSE of what happens when $D = D_G^*$

$$G^* = \underset{G}{\operatorname{argmin}} (D_G^*, G)$$

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

also assume G is invertible.
$Z = G^{-1}(x)$

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

④ Atp, GAN has a unique sol^n G^* and this satisfies $P_g = P_{data}$ when...

$$D_G^* = \frac{P_{data}(x)}{P_{data}(x) + P_g(x)}$$

So now, when we substitute this value:

$$G^* = \underset{G}{\operatorname{argmin}}\, V(D_G^*, G)$$
$$= \underset{G}{\operatorname{argmin}} \left\{ \int_x P_{data}(x) \cdot \log(D^*(x)) + P_g(x) \log(1 - D_G^*(x))\, dx \right\}$$
$$= \underset{G}{\operatorname{argmin}} \left\{ \int_x P_{data}(x) \cdot \log\left(\frac{P_{data}(x)}{P_{data}(x) + P_g(x)}\right) + P_g(x) \cdot \log\left(\frac{P_g(x)}{P_{data}(x) + P_g(x)}\right) dx \right\}$$

whole bunch of math.

⑤ JS divergence for when diff prob. dist:

$$JS(P_1 \| P_2) = \frac{1}{2} \mathbb{E}_{x \sim p_1} \ln\left(\frac{P_1}{P_1 + P_2/2}\right) + \frac{1}{2} \mathbb{E}_{x \sim p_2} \ln\left(\frac{P_2}{P_1 + P_2/2}\right)$$

So, again, after all that math:

$$G^* = \underset{G}{\operatorname{argmin}} \left\{ -\log 4 + 2\, JSD(P_{data}(x) \| P_g(x)) \right\}$$

So what we did was:

1. optimize discriminator for fixed G.
2. plug that in for optimal value of generator.

Training Loop for GAN.

① fix learning of G^*

Inner loop for D:

- take m real, m fake data samples.

Namita Achyuthan

Inner loop for D:

- take m real, m fake data samples.
- update θ_d by gradient ASCENT $\xrightarrow{\text{maximize}} \frac{\partial}{\partial \theta_d} \frac{1}{m} \left[\ln[D(x)] + \ln[1 - D(G(z))] \right]$

② fix learning of D^*

- take m data samples.
- update θ_g by gradient DESCENT $\xrightarrow{\text{minimize}} \frac{\partial}{\partial \theta_g} \frac{1}{m} \left[\ln(1 - D(G(z))) \right]$