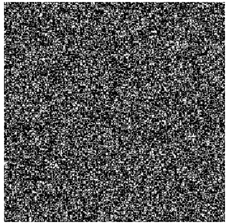
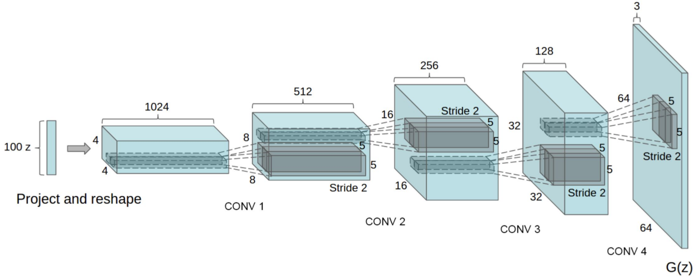
Generative Adversarial Networks

Presented by Yi Zhang

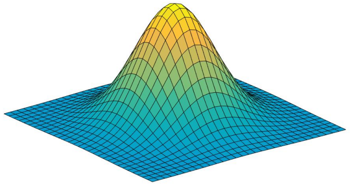
Deep Generative Models

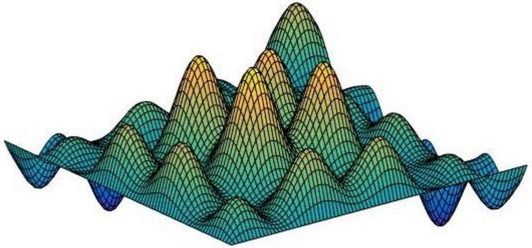


N(O, I)



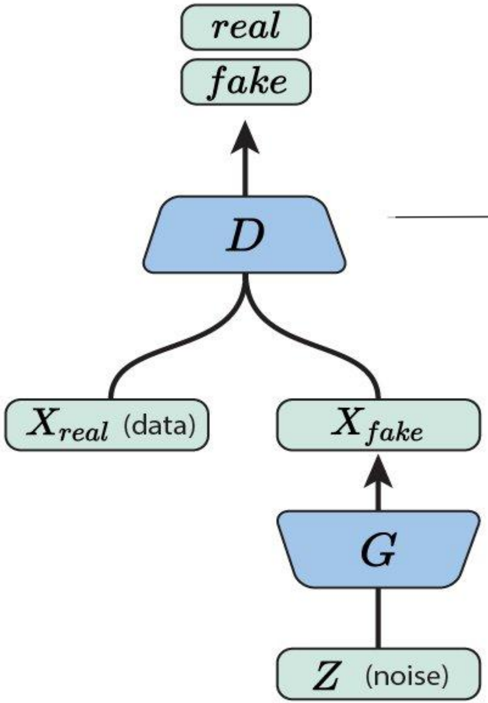
Variational Auto-Encoders
GANs





Unreasonable Effectiveness of GANs

GANs



Discriminator
tries to distinguish genuine data from fake ones

Generator
turns noise into imitations of data, trying to fool the Discriminator

$$\min_G \max_D \; E_{x \sim P_{real}}[f(D(x))] + E_h[f(1 - D(G(h)))].$$

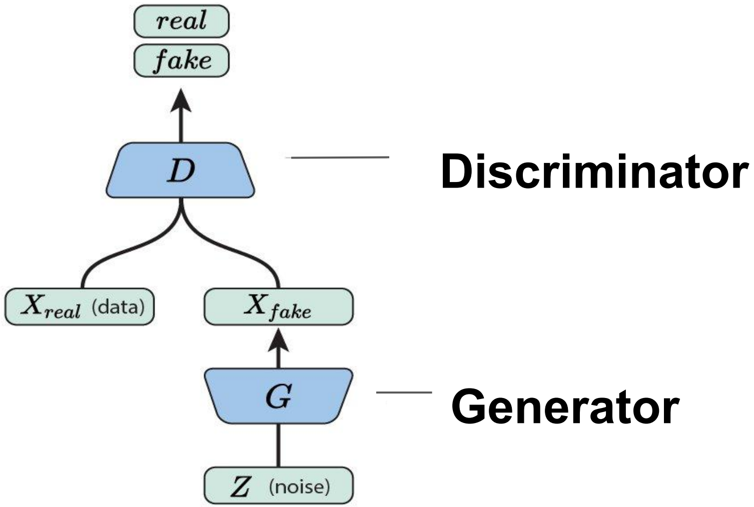
- f(x) = log(x) for ordinary GANs

Idea 1: *deep nets are good at recognizing images, then let it judge of the outputs of a generative model*

Idea 2: *If a good discriminator net has been trained, use it to provide "gradient feedback" that improves the generative model.*

Idea 3: *Turn the training of the generative model into a game of many moves or alternations.*

GANs



Training a GAN ⟹ playing a two-player game:

$$\min_{\theta} \max_{\phi} F(\theta, \phi)$$

where $F(\cdot, \cdot)$ is the payoff function:

$$F(\theta, \phi) = \mathbf{E}_{x \sim P_{real}}[f(D_\phi(x))] + \mathbf{E}_{z \sim \mathcal{N}}[f(1 - D_\phi(G_\theta(z)))]$$

where θ and ϕ are the parameters of the generators and discriminators.

Some open questions

1. Does an equilibrium exist?

   Pure equilibrium may not exist in general.
   i.e, the rock/paper/scissor game

2. *Suppose the equilibrium exists.*
   Does the generator win at the equilibrium?

   It seems to be the case in practice, with various training techniques.

3. *Suppose the generator wins.*
   What does this say about whether or not generator's distribution is close to the real distribution?

   Most GANs' research focus on this area.
   i.e design objective functions.

Objective functions of GANs

Original GANs $f(x) = log(x)$

$$\min_G \max_D \quad E_{x\sim P_{real}}[f(D(x))] + E_h[f(1 - D(G(h)))].$$

The optimal Discriminator is $D(x) = P_{real}(x)/(P_{real}(x) + P_{synth}(x)).$

Then the objective function becomes the Jenson-Shannon Divergence!

$$JSD[P\|Q] = JSD[P\|Q] = \frac{1}{2}KL\left[P\middle\|\frac{P+Q}{2}\right] + \frac{1}{2}KL\left[Q\middle\|\frac{P+Q}{2}\right].$$

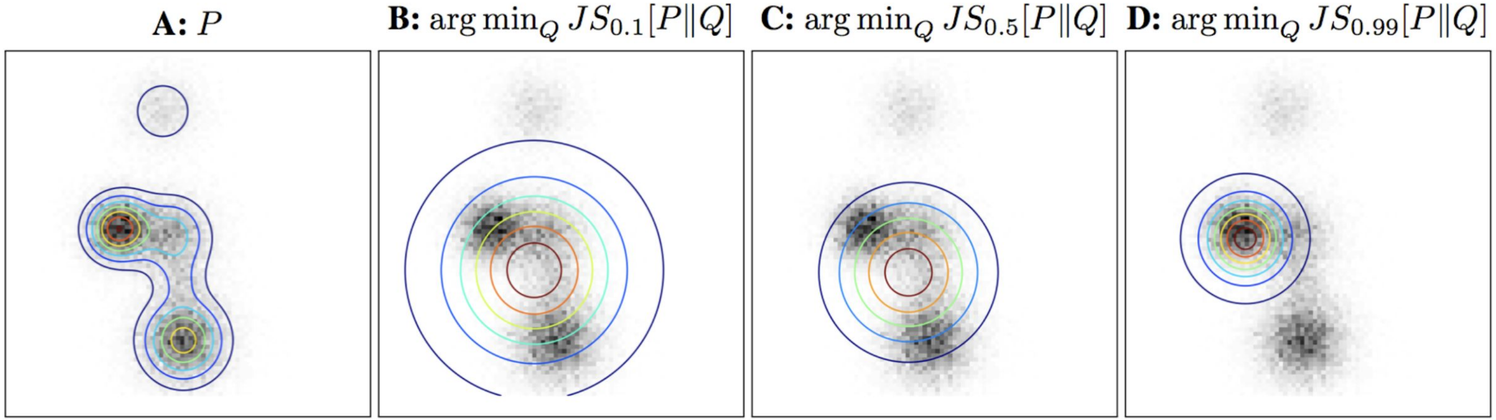
Objective functions of GANs

Original loss function $f(x) = \log(x)$

Consider a generalized family of JSD

$$JS_\pi[P\|Q] = \pi \cdot KL[P\|\pi P + (1-\pi)Q] + (1-\pi)KL[Q\|\pi P + (1-\pi)Q].$$

(Pi = 0.5 recovers the ordinary JSD)



Objective functions of GANs

Wasserstein GANs $\boldsymbol{f(x) = x}$

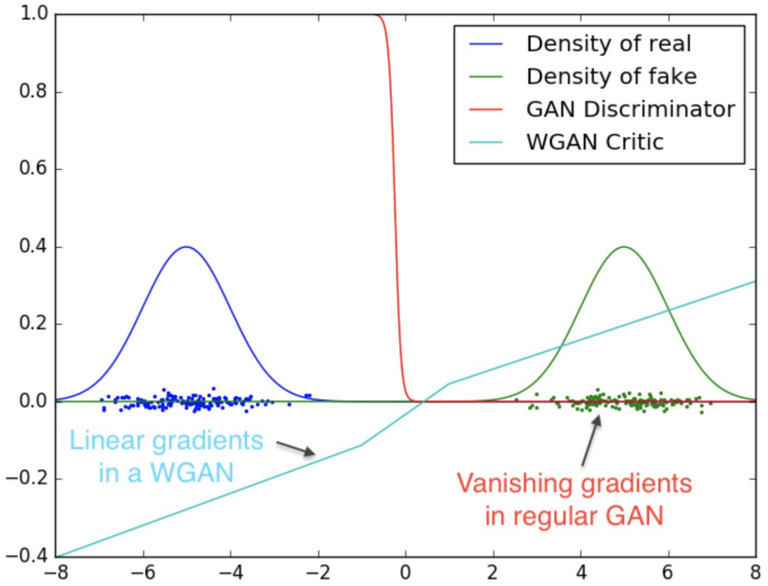
$$\min_G \max_D \; E_{x\sim P_{real}}[f(D(x))] + E_h[f(1 - D(G(h)))].$$

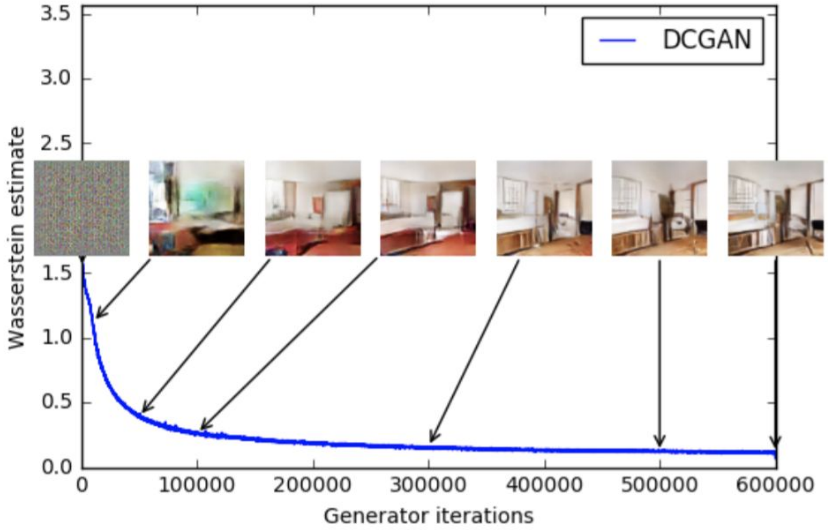
Wasserstein Distance: $$W(P||Q) = \sup_{||D||_L \le 1} \mathop{\mathbb{E}}_{x\sim P}[D(x)] - \mathop{\mathbb{E}}_{x\sim Q}[D(x)]$$

the amount of probability mass that must be transported from P to Q

Objective functions of GANs

Wasserstein GANs $\boldsymbol{f(x) = x}$





Objective functions of GANs

Wasserstein GANs $f(x) = x$

$$\min_G \max_D \; E_{x \sim P_{real}}[f(D(x))] + E_h[f(1 - D(G(h)))].$$

Wasserstein Distance: $$W(P||Q) = \sup_{||D||_L \leq 1} \mathop{\mathbb{E}}_{x \sim P}[D(x)] - \mathop{\mathbb{E}}_{x \sim Q}[D(x)]$$

the amount of probability mass must be transported from P to Q

However…JSD and Wasserstein Distance don't generalize !!!

Theorem 3.1. *Let μ, ν be uniform Gaussian distributions $\mathcal{N}(0, \frac{1}{d}I)$. Suppose $\hat{\mu}, \hat{\nu}$ are empirical versions of μ, ν with m samples. Then when $\log m \ll d$, we have with high probability*

$$d_{JS}(\mu, \nu) = 0, d_{JS}(\hat{\mu}, \hat{\nu}) = \log 2.$$
$$d_W(\mu, \nu) = 0, d_W(\hat{\mu}, \hat{\nu}) \geq 1.1.$$

Further, let $\tilde{\mu}, \tilde{\nu}$ be the convolution of $\hat{\mu}, \hat{\nu}$ with a Gaussian distribution $N(0, \frac{\sigma^2}{d}I)$, as long as $\sigma < \frac{c}{\sqrt{\log m}}$ for small enough constant c, we have with high probability

$$d_{JS}(\tilde{\mu}, \tilde{\nu}) > \log 2 - 1/m.$$

However…JSD and Wasserstein Distance don’t generalize !!!

The training objective **<u>should not</u>** be interpreted as minimizing these two distances between distributions !

Neural-Network Divergence

$$d_{NN}(P||Q) = \sup_{D \in NN} \mathbf{E}_{x \sim P}[D(x)] - \mathbf{E}_{x \sim Q}[D(x)]$$

where NN is a class of (small) neural networks.

If the class NN is small, then we have generalization guarantee:

Suppose n is the number of parameters of the discriminator D, m is the number of samples in $\hat{P}$ and $\hat{Q}$. If $m \geq Cn\log(n)/\epsilon^2$, then

$$|d_{NN}(P||Q) - d_{NN}(\hat{P}||\hat{Q})| \leq \epsilon$$

with high probability.

But here is the bad news….

Suppose n is the number of parameters of the discriminator D, m is the number of samples in $\hat{P}$ and $\hat{Q}$. If $m \geq Cn\log(n)/\epsilon^2$, then

$$|d_{NN}(P||Q) - d_{NN}(\hat{P}||\hat{Q})| \leq \epsilon$$

with high probability.

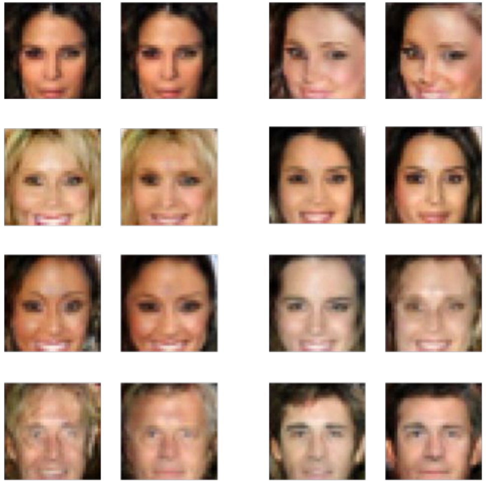
$Cn\log(n)/\epsilon^2$ is usually small compared to the of modes of real-life data distribution, i.e. # of faces

The generator can fool the discriminator by memorizing a small number of images.

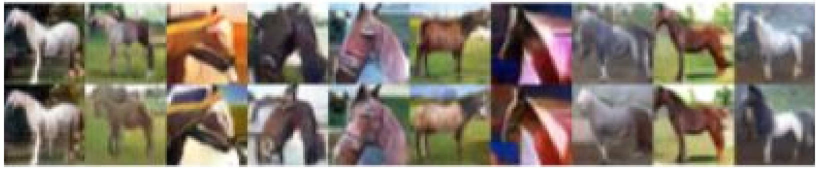
But here is the bad news….

$Cn\log(n)/\epsilon^2$ is usually small compared to the of modes of real-life data distribution, i.e. # of faces

The generator can fool the discriminator by memorizing a small number of images.



evidence of lack of diversity



Equilibrium in GANs

GAN is a two-player game with the following payoff function

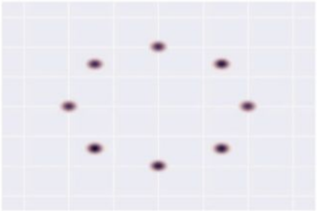
$$F(u,v) = \mathop{\mathbb{E}}_{x \sim \mathcal{D}_{real}} [f(D_v(x))] + \mathop{\mathbb{E}}_{h \sim \mathcal{D}_h} [f(1 - D_v(G_u(h)))].$$

where u, v are the parameters of the generator G and the discriminator D.

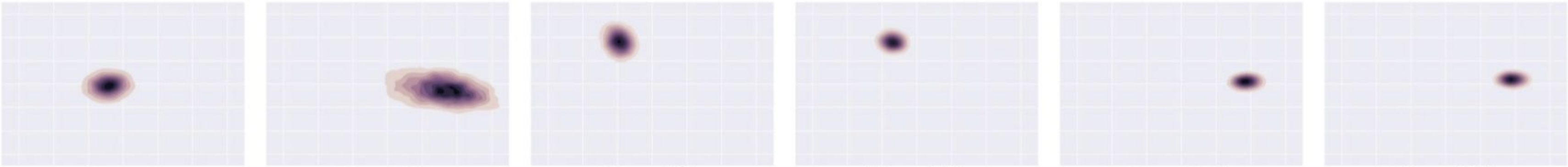
Equilibrium may not exist, for pure strategies!!!
(rock/paper/scissors)

Equilibrium in GANs

Equilibrium may not exist for pure strategies!!!
(rock/paper/scissors)



Target



Step 0 Step 5k Step 10k Step 15k Step 20k Step 25k

Equilibrium in GANs

Equilibrium may not exist for pure strategies!!!
(rock/paper/scissors)

[von Neumann's Min-Max Theorem]

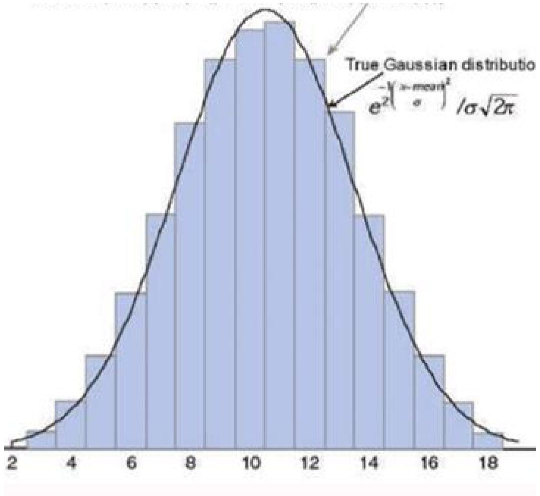
There exists an equilibrium, if players are allowed to play *mixed-strategies*.

i.e. infinite mixture of generators
infinite mixture of discriminators

Equilibrium in GANs

Then the equilibrium exists.
But does the (infinite mixture of) generator(s) **win** the game at it?

Yes!!!!



Infinite mixture of generators

Each generator is responsible for generating a single image

and in real life…..
eps-approximate if we use a mixture of size O(n log(n) / eps^2)

Conclusion

- Distinguish between JSD/Wasserstein Distance and Neural-Net Divergence
- Neural-Net Divergence ensures generalization, but doesn’t encourage diversity
- Pure equilibrium may not exist; we need to allow mixed strategies
- Then the equilibrium exists, and the generators win the game
- In practice, we can approximate the equilibrium using a finite mixture