· Likelihood and Sample Quality

1. high log-likelihood, poor samples

$p_\theta(x) = 0.01 \cdot p_{data}(x) + 0.99\, p_{noise}(x)$

99% samples are noise

$$\log p_\theta(x) = \log[0.01\, p_{data}(x) + 0.99\, p_{noise}(x)]$$
$$\geq \log 0.01\, p_{data}(x)$$
$$= \log p_{data}(x) - \log 100$$

lower bound: $E_{x\sim p_{data}}[p_\theta(x)] \geq E_{x\sim p_{data}}[\log p_{data}(x)] - \log 100$

upper bound: $KL(p_{data} \| p_\theta) = E_{x\sim p_{data}}[\log p_{data}(x)] - E_{x\sim p_{data}}[\log p_\theta(x)] \geq 0$

$E_{x\sim p_{data}}[p_\theta(x)] \leq E_{x\sim p_{data}}[\log p_{data}(x)]$

$E_{x\sim p_{data}}[\log p_{data}(x)]$

$\downarrow$

$E_{x\sim p_{data}}[p_\theta(x)]$

100

$\uparrow$

$E_{x\sim p_{data}}[\log p_{data}(x)] - 100$

as we increase the dimensionality of x, $\log p_{data}(x)$ gets bigger in absolute value than in the sense of relative difference, $E_{p_{data}}[\log p_\theta(x)] \ll E_{p_{data}}[\log p_{data}(x)]$

high log-likelihood does not imply good sampling

2. good samples, low likelihoods $p_\theta(x_{sample})$

just memorize the training set

· Two-Sample Test

given $S_1 = \{x \sim P\}$ $S_2 = \{x \sim Q\}$, a two-sample test consider the following hypothesis

null hypothesis: $H_0 : P = Q$

alternate hypothesis: $H_1 : P \neq Q$

Test statistic T compares S_1 and S_2 (eg. difference in mean or variance)

if T is less than a threshold α, accept H_0

Test statistic T is likelihood-free since it does not involve $P(x)$ or $Q(x)$

Assume direct access to $S_1 = D = \{x \sim P_{data}\}$

have a model distribution P_θ which is easy to sample from, Let $S_2 = \{x \sim P_\theta\}$

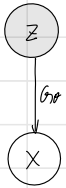
train the generative model to minimize a two-sample test objective between S_1 and S_2

Test statistic: Learn a statistic that maximize a suitable notion of distance between S_1 and S_2

Generator Adversarial Network

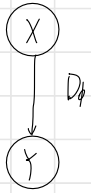
generator

1. directed latent variable model with a deterministic mapping from z to x given by G_θ
2. minimizes a two-sample test objective (in support of null hypothesis $P_{data} = P_\theta$)



discriminator:

1. any function (eg. neural network) which tries to distinguish "real" samples from the dataset and "fake" samples from generator
2. maximize the two-sample test objective (in support of the alternate hypothesis $P_{data} \neq P_\theta$)



discriminator: $\max_D V(G, D)$

$$V(G, D) = E_{x \sim P_{data}}[\log D(x)] + E_{x \sim P_G}[\log(1 - D(x))]$$

want a large log probability for samples from data distribution

want a low log probability for samples from generator distribution

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

$$\frac{dV}{dD_\theta(x)} = P_{data}(x) \cdot \frac{1}{D(x)} - P_G(x) \frac{1}{1 - D(x)} = 0$$

optimal discriminator: $D^*_\theta(x) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)}$

$$D^*_\theta(x) = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)}$$

if $P_{data}(x) \approx P_G(x)$, then the discriminator cannot tell whether x is from P_{data} or P_G

generator: $\min_G V(G,D)$

$\max_D E_{x\sim P_G}[\log(1-D(x))]$

for the optimal discriminator $D_G^*(\cdot)$. we have

$$V(G, D_G^*) = E_{x\sim P_{data}}\left[\log D_G^*(x)\right] + E_{x\sim P_G}\left[\log(1-D_G^*(x))\right]$$

$$= E_{x\sim P_{data}}\left[\log \frac{P_{data}(x)}{P_{data}(x)+P_G(x)}\right] + E_{x\sim P_G}\left[\log \frac{P_G(x)}{P_{data}(x)+P_G(x)}\right]$$

$$= E_{x\sim P_{data}}\left[\log \frac{P_{data}(x)}{\frac{P_{data}(x)+P_G(x)}{2}}\right] + E_{x\sim P_G}\left[\log \frac{P_G(x)}{\frac{P_{data}(x)+P_G(x)}{2}}\right] - \log 4$$

$$= KL\left[P_{data} \,\|\, \frac{P_{data}+P_G}{2}\right] + KL\left[P_G \,\|\, \frac{P_{data}+P_G}{2}\right] - \log 4$$

2X Jenson-Shannon Divergence

$$= 2 \times D_{JSD}(P_{data} \| P_G) - \log 4$$

the optimal generator approximate $P_G = P_{data}$

for the optimal generator $G^*(\cdot)$ and $D_{G^*}^*(\cdot)$, we have $V(G^*, D_{G^*}^*) = -\log 4$

· Jenson-Shannon Divergence

$$D_{JSD}(P\|Q) = \frac{1}{2}\left[KL\left(P\|\frac{P+Q}{2}\right) + KL\left(Q\|\frac{P+Q}{2}\right)\right]$$

also called symmetric KL divergence

$D_{JSD}(P\|Q) \geq 0$ $=0$ iff $P=Q$

$D_{JSD}(P\|Q) = D_{JSD}(Q\|P)$

D_{JSD} is not a measure of distance since it does not satisfy the triangle inequality, but $\sqrt{D_{JSD}}$ does

· GAN Training Algorithm

$$V(G,D) = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{x\sim p_g}[\log(1-D(x))]$$

Sample $x^{(1)} \cdots x^{(m)}$ from D

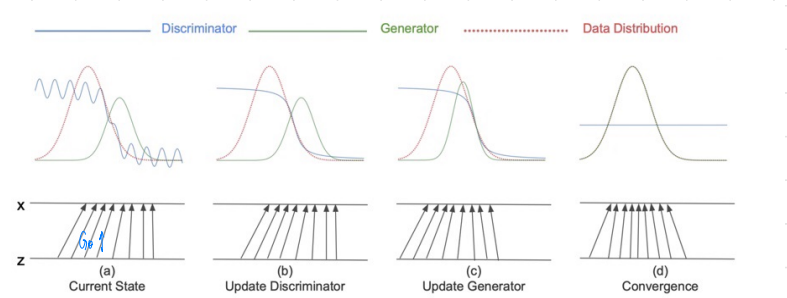
Sample $z^{(1)} \cdots z^{(m)}$ from noise p_z

Update the generator parameters θ by stochastic gradient descent

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

Update the discriminator parameters ϕ by stochastic gradient ascent

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



· Optimization Challenges

Unrealistic assumption: we specify a function and only make update in parameter space

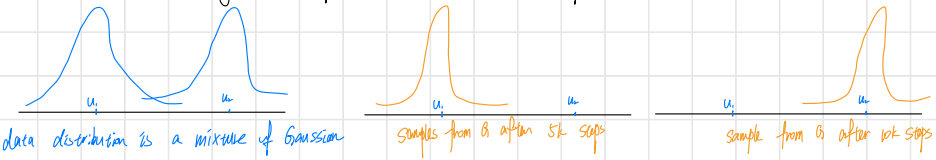
If the generator updates are made in function space, and discriminator is optimal at every step then the generator guaranteed to converge to the data distribution — unrealistic

no robust stopping criteria in practice (unlike MLE)

· Mode Collapse

GANs are notorious for suffering from mode collapse

the phenomena where the generator collapses to one or several examples



Fixing mode collapse are mostly empirically driven:
alternate architectures, adding regularization, injecting small noise perturbation https://github.com/soumith/ganhacks