Standard cost function for the discriminator is based on binary classification:

$$J^{(D)}(\theta^{(D)}, \theta^{(G)}) = -\frac{1}{2} \mathbb{E}_{x \sim p_{data}} \log D(x) - \frac{1}{2} \mathbb{E}_{z} \log (1 - D(G(z)))$$

(cross-entropy cost when learning a binary classifier with sigmoid output.

Note the classifier is trained on two minibatches of data:

→ one is dataset where label is 1 for all examples

→ one is from generator where label is 0 for all examples.

By training the discriminator, we obtain estimate of ratio:

$$\frac{p_{data}(x)}{p_{model}(x)}$$

Estimating this ratio enables computing various divergences:

Rather than lower bounds like ELBO, here GAN approximation is based on using supervised learning to estimate ratio of two densities.

[since we're using supervised learning, may have overfitting / underfitting]

What about cost function for generator?

Simplest kinds of games are zero-sum games, where sum of all players' costs is ~~zero~~ always zero, so

$$J^{(G)} = -J^{(D)}$$

so entire game can be summarized by value function specifying the discriminator's payoff.

$$V(\theta^{(D)}, \theta^{(G)}) = -J^{(D)}(\theta^{(D)}, \theta^{(G)})$$

Since zero-sum games are minimax games.

$$\theta^{(G)*} = \underset{\theta^{(G)}}{\operatorname{argmin}} \max_{\theta^{(D)}} V(\theta^{(D)}, \theta^{(G)})$$

Further, under this minimax formulation, Goodfellow et al. showed the game essentially minimizes Jensen-Shannon divergence between data and model distribution.

An alternative that seems to work better in practice (but not so theoretically motivated)

→ continue to use cross-entropy minimization for discriminator, but instead of flipping sign on discriminator's cost for generator, we flip the target used to construct the cross-entropy cost

$$J^{(G)} = -\frac{1}{2} \mathbb{E}_z \log D(G(z))$$

So generator maximizes log-probability of discriminator being mistaken rather than minimizing log-probability of discriminator being correct.

<u>maximum likelihood</u>

MLE can be thought of as minimizing KL divergence

$$\theta^* = \underset{\theta}{\operatorname{argmin}} D_{KL}(p_{data}(x) \| p_{model}(x; \theta))$$

if the discriminator is optimal, then one can minimize ↑, to obtain MLE when

$$J^{(G)} = \frac{-1}{2} \mathbb{E}_z \left(\exp\left(\sigma^{-1}(D(G(z))) \right) \right)$$

where σ is logistic sigmoid function.

[Equivalent in expectation; Goodfellow et al. (2014)]

minimax $\Rightarrow$ Jensen-Shannon divergence

ML $\Leftrightarrow$ K-L divergence.

What about other divergences?

Goal of discriminator is to minimize

$$J^{(D)}(\theta^{(D)}, \theta^{(G)}) = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}} \log D(x) - \frac{1}{2}\mathbb{E}_z \log\left(1 - D(G(z))\right)$$

with respect to $\theta^{(D)}$.

Supposing discriminator can be optimized in function space so value of $D(x)$ is specified independently for every value of x.

→ Assume both p_{data} and p_{model} are non-zero everywhere

To minimize $J^{(D)}$ with respect to D, we write the functional derivatives with respect to a single entry $D(x)$, then set equal to zero.

$$\frac{\partial}{\partial D(x)} J^{(D)} = 0$$

By solving, we get:

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_{model}(x)}$$

Estimating this ratio is key approximation mechanism used by GANs.

ML in GAN framework

(original minimax formulation corresponds to Jensen-Shannon divergence.)

→ derive a cost function that yields approximate maximum likelihood within GAN framework.

Goal: to design $J^{(D)}$ so if we assume discriminator optimal, then expected gradient of $J^{(G)}$ is matched to expected gradient of $D_{KL}(p_{data} \| p_{model})$

ie. $J^{(G)} = \mathbb{E}_{x \sim p_g} f(x)$ is form of solution, find $f(x)$.

---

take derivative of KL divergence with respect to parameter θ:

$$\frac{\partial}{\partial \theta} D_{KL}(p_{data} \| p_g) = -\mathbb{E}_{x \sim p_{data}} \frac{\partial}{\partial \theta} \log p_g(x).$$

now find f to make $E_{x \sim p_g} f(x)$ have derivative that matches

so take derivative of $\mathbb{E}_{x \sim p_g} f(x)$:

$$\frac{\partial}{\partial \theta} J^{(G)} = E_{x \sim p_g} f(x) \frac{\partial}{\partial \theta} \log p_g(x).$$

To obtain this, needed two assumptions

(1) assume $p_g(x) > 0$ everywhere, ie. we can use identity $p_g(x) = \exp(\log p_g(x))$

(2) assume we can use Leibniz rule to exchange order of differentiation and integration.

so we see derivatives of $J^{(G)}$ close to what we want,

but problem is expectation is computed by drawing samples from p_g. when we would want it to be computed using samples from p_{data}.

use importance sampling to reweight things by

choosing $f(x) = \frac{p_{data}(x)}{p_g(x)}$

GAN is a minimax game between two neural networks

- Generator
- Discriminator

Most people think a GAN is working well if:

① generator can reliably generate data that fools the discriminator.

② generator creates samples that are as diverse as the distribution of the real-world samples.

Mode collapse: when generator fails at ② and all generated samples are similar or the same.

Generator learns to map several different values of z to same "golden" point that fools the discriminator

Partial mode collapse

mode collapse may arise because maximin solution to GAN game is different from minimax solution.

$G^* = \min_G \max_D V(G,D)$, here G^* draws samples from the data distribution.

when we exchange the order and find

$G^* = \max_D \min_G V(G,D)$ instead,

the min w.r.t. generator now lies in inner loop of optimization

→ the generator is asked to map every z value to the single x coordinate that the discriminator believes is most likely to be real rather than false.

simultaneous gradient descent doesn't clearly privilege minimax over maximin.

[hope is it'll do minimax, but often does maximin].

Ways of addressing mode collapse.

unrolling: update the generator's loss function to back propagate through k steps of gradient updates for discriminator. This lets generator see k steps into future to encourage more diverse samples.

packing: modify discriminator to make decisions on several samples of the same class, either real or artificial.

→ several samples that the same are more likely to be artificial.