E0 306: Deep Learning: Theory and Practice 26th March 2019

Scribe Notes
Generative Adversarial Networks

Lecturer: Amit Deshpande *Scribe*: Vishal Tripathy

In this lecture, some problems such as "mode collapse" with training of GANs and suggest some techniques to overcome this problem. In mode collapse generator not generate does not really generate the true data distribution but ends up with other distribution which does not generate some parts of the data. One of the paper followed in this lecture is "Towards Principles methods for Training Generative Adversarial Networks[1]" by Martin Arjovsky and Leon Bottou. This is about some principled method for training GANs. This pointed out some problems with the objective functions in GANs and resulting instability and slow training caused by it.
The above doesn't introduce any algorithm, rather it is more about understanding the training dynamics of GANs. There are 2 sections:

- problems due to instability and saturations which arises during training GANs
- Different variants of GANs to overcome this problem

Definition

A distribution mapped to X,

- $P_g(x)$: generator distribution over data x
- $P_z(z)$: input noise or random seed

There are 2 things at play here.

1. generator(G) $G : (\mathbb{R}^k, \theta_g) \rightarrow \mathbb{R}^d$
2. discriminator(D) $D : (\mathbb{R}^d, \theta_d) \rightarrow \{0, 1\}$

where θ_g, θ_d are model parameters of generator discriminator respectively. Also, the actual distribution is also from space $\mathbb{R}_d$. So, both output of generator and actual distribution are in same domain(may be in same or different part of the domain).
We train D to maximize the probability of assigning the correct label to be both training example and sample from G.

The loss function associated with is,

$$\min_G \max_D \mathop{\mathbb{E}}_{x \sim P_{data}()x} [\log(D(x))] + \mathop{\mathbb{E}}_{z \sim P_z(z)} [1 - \log(D(G(z))]$$

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

Actually the goal of discriminator to separate the true data from the generated one. So, if we keep G fixed then the loss function decreases with increase in D(x) and increases with increase in D(G(z). And the goal of generator is to maximize the loss with respect to actual data and maximize with respect to the generated sample.

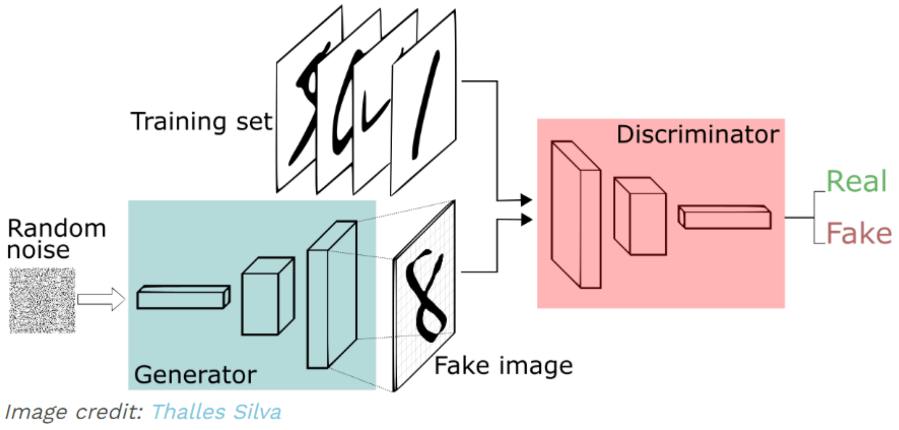

Figure 1: demonstration of the basic idea behind GAN

If we keep the generator fixed,then

$$\operatorname*{argmax}_D V(G, D) \Rightarrow D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

However,

$$\min_G (V(G, D^*)) = -log(4) + 2 * JS(p_{data}, p_g)$$

where JS is Jensen-Shannon divergence

$$\min_G JS(p_{data}, p_g) \text{ is convex problem}$$

The above minimization is convex problem in generator distribution **but may not be in generator parameters**. But in practice we hope that that convex in distribution is also implies convex on each parameters. In practice we do training by

- finding optimal discriminator
- and for that discriminator find the optimal generator(convex)

The above method iterates till stationary point for both generator and discriminator is reached. Since, optimizing generator after discriminator is convex in theory(minimizing Jensen-Shannon divergence). Hence, it is one step minimization step in theory but not in practice. This is because it is not convex in parameters. This is just **minimizing distance between true data distribution and genrated distributions**.
Suppose, we had access to any distribution. If space of all generated distribution is same as the space of all distributions. Then we start with the generator and look the the optimal discriminator then optimize the generator in step.
In practice, we do gradient update for both discriminator and generator. we can use other heuristics of distance between distributions as well. One distance heuristics which is popular is KL divergence(Kullback-Leibler divergence). KL-divergence:

$$KL(p_{data}||p_g) = \int_{\mathbb{X}} p_{data}(x) \log \frac{p_{data}(x)}{p_g(x)} dx$$

- if $p_{data} > p_g$, then x is a point with higher probability of coming from the data than being a generated sample. This is the core of the phenomenon commonly described as mode dropping: when there are large regions with high values of p_{data}, but small value of p_g. If $p_{data} > 0$ and $p_g \to 0$ then KL divergence explodes quickly. So, KL divergence penalizes the generator for not covering the data distribution.

- on the other hand if $p_{data} < p_g$, then x is more likely to be generated by the generator than to be a data point. So, the case where $p_{data} \to 0$ and $p_g > 0$. KL doesn't penalize generator resulting in fake images that doesn't look real.

So, in order to go for a middle ground by using Jensen-Shannon divergence,

$$JSD(p_{data}||p_g) = \frac{1}{2}(KL(p_{data}||p_m) + \frac{1}{2}KL(p_g||p_m)))$$

where $p_m = \frac{p_{data}+p_g}{2}$
So, Jensen-Shannon divergence is average of KL of p_{data} to average and p_g to average.
Success of GAN above other previous papers is due to use of JS rater than KL.
GAN also has drawbacks like, the after some iterations of training the updates become very small due to small gradients.
Instead of using $\log[1 - D(G(z))]$ it was proposed to use $\log(D(G(z))$. This improves gradient a lot. But, keep in mind **with this change this is no longer Jensen-Shannon Divergence**. But, now instead of minimizing convex function we maximize concave function.
Even after this change update still gets worse after certain point.

So, why do updates go worse?
One of the reason might be that the data distribution lies in a low dimensional manifold(like MNIST). So, it is highly likely that generator and discriminator lie in different manifold with zero measure intersection.
In the above scenario, at either one of p_g or p_{data} is zero almost everywhere. This is because the measure of both of them to be nonzero is is zero. For, example the MNIST data set lie in a domain of 784 dimmensions. But the 10 numbers together form a small dimensional manifold inside the 784 dimensions.
Just a sketch of idea, When you perform non-linearity on a one dimensional space

it transforms in a manifold. So, if you preform same non-linearity on n different one dimensional spaces the resulting space will be contained in union of less than n manifolds. (And linear transformations have no effect)

In this case,

$$JSD(p_{data}||p_g) = \frac{1}{2}(KL(p_{data}||p_m) + \frac{1}{2}KL(p_g||p_m)))$$

where $p_m = \frac{p_{data}+p_g}{2}$

$$\Rightarrow JSD(p_{data}||p_g) = KL(p_{data}||p_{data})\text{or}KL(p_g||p_g)$$

In both the cases, it is zero. So, it won't make any progress.

Theorem 1 *If two distributions $\mathbb{P}_r$ and $\mathbb{P}_g$ have support contained on two disjoint compact subsets $\mathbb{S}_1$ and $\mathbb{S}_2$ respectively, then there is a smooth optimal discriminator $\ni D^* : \mathbb{X} \to [0,1]$ that has accuracy 1 and $\nabla_x D^*(x) = 0$, $\mathbb{S}_1 \cup \mathbb{S}_2 \in \mathbb{X}$*

Proof: Our loss function is,

$$\underset{x\sim\mathbb{P}_r}{E}[\log D(x)] + \underset{x\sim\mathbb{P}_\mathfrak{d}}{E}[\log(1 - D(x))]$$

Since, these 2 sets and compact and disjoint in Hausdoff space. Hence, from Urisohn's lemma there is smooth function $D^*(x) :\to [0,1]$ such that $D*(\mathbb{S}_1) = 1$ and $D^*(\mathbb{S}_2) = 0$.So, this discriminator will produce accuracy of 1.

Theorem 2 *Vanishing gradient on the generator Let $g_\theta : \mathbb{Z} \to \mathbb{X}$ be a differentiable function that induces a distribution $\mathbb{P}_\mathfrak{d}$. Let $\mathbb{P}_r$ be the real data distribution. Let D be a differentiable discriminator. If the condition of previous theorem satisfiea, $||D - D^*|| < \epsilon$,and $\underset{z\sim p(z)}{\mathbb{E}}[||J_\theta g_\theta(z)||_2^2] \leq M^2$, then*

$$||\nabla_\theta \underset{z\sim p_z}{\mathbb{E}}[\log(1 - D(g_\theta(z)))]||_2 < M\frac{\epsilon}{1-\epsilon}$$

Towards Softer Metrics And Distributions

As we have seen earlier the instability and vanishing gradient is partly because of non-existing overlap between range of actual data and generator. We will try to fix that using noise and smoothening the distribution of the probability mass. Hence increasing the stability.

Theorem 3 *If X has a distribution $\mathbb{P}_X$ with support on $\mathbb{M}$ and ϵ is an absolute continous random variable with density P_ϵ, then $\mathbb{P}_{X+\epsilon}$ is absolutely continous with probability density of*

$$P_{X+\epsilon}(x) = \int_{\mathbb{M}} P_\epsilon(x-y)d\mathbb{P}_X(y)$$

With this noise being added the optimal discriminator changes to,

$$D^*(x) = \frac{p_{r+\epsilon}}{p_{r+\epsilon+p_{g+\epsilon}}}$$

Earlier gradient update was,

$$2 * \nabla_\theta JSD(p_{data}, p_{g_\theta})$$

Now it becomes,

$$\nabla_\theta JSD(p_{data+\epsilon, p_{g_\theta+\epsilon}})$$

Theorem 4 *Let $\mathbb{P}_r$ and $\mathbb{P}_g$ be two distributions wih support on $\mathbb{M}$ and $\mathbb{P}$ repectively, with $\epsilon \in \mathbb{N}(0, \sigma^2\mathbb{I})$. Then, the gradient passed to the generaor has the form*

$$\begin{aligned}
&\underset{z\sim\mathbb{P}_z}{E}[\nabla_\theta \log(1 - D^*(g_\theta(z)))] \\
&\quad = \underset{z\sim\mathbb{P}_z}{E}[a(z)\int_{\mathbb{M}} P_\epsilon(g_\theta(z)-y)\nabla_\theta||g_\theta(z)-y||^2 d\mathbb{P}_r(y) \\
&\quad - b(z)\int_{\mathbb{P}} P\epsilon(g_\theta(z)-y)\nabla_\theta||g_\theta(z)-y||^2 d\mathbb{P}_g(y)]
\end{aligned}$$

where a(z),b(Z) is positive with $b(z) > a(z)$ iff $P_{r+\epsilon} > P_{g+\epsilon}$,

This theorem proves that we will drive our samples $g_\theta(z)$ towards points along the data manifold, weighted by their probability and the distance from our samples.

References

[1] Martin Arjovsky and Léon Bottou. Towards Principled Methods for Training Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1701.04862, Jan 2017.