Generative Adversarial Networks Report

Bohao Zou 917796070
Bingdao Chen 917781027

STA 221 Question 2

University of California, Davis

May 16, 2020

Kullback-Leibler Divergence

Kullback-Leibler (KL) divergence is defined as $D_{KL}(P\|Q) = \mathbb{E}_{x\sim P}\left[log\frac{P(x)}{Q(x)}\right]$. It is used to measure how different two distributions if two separate probability distributions $P(x)$ and $Q(x)$ over the same random variable x. The KL divergence is non-negative and asymmetry, so $\mathbb{E}_{x\sim P}\left[log\frac{P(x)}{Q(x)}\right]$ and $\mathbb{E}_{x\sim Q}\left[log\frac{Q(x)}{P(x)}\right]$ is different. Therefore, the results are different if minimizing $D_{KL}(p\|q)$ and $D_{KL}(q\|p)$. For the first situation, $q(x)$ are more likely to put high probability mass on all of modes of $p(x)$. For the second situation, $q(x)$ is more focused on one single mode of $p(x)$, avoiding putting probability mass in the low probability areas between modes of $p(x)$.

KL Divergence and MLE

For an unknown data-generating distribution $p_{data}(x)$, we want to find a good estimator $p_{model}(x;\theta)$ for the true probability $p_{data}(x)$. Maximum Likelihood Estimation (MLE) is one method, and it is defined as $\theta_{MLE} = \text{argmax}_\theta \prod_{i=1}^{m} p_{model}(x^{(i)};\theta)$ After log-transformation and re-scaling, it can be expressed as an expectation with respect to the empirical distribution $\hat{p}_{data}$: $\theta_{MLE} = \text{argmax}_\theta \mathbb{E}_{x\sim\hat{p}_{data}} \text{log}p_{model}(x;\theta)$ .Besides finding the maximum likelihood θ_{MLE}, another way to view this problem is to minimize the dissimilarity between the empirical distribution $\hat{p}_{data}$ and the model distribution p_{model}, which can be measured by KL divergence as follows. $D_{KL}(\hat{p}_{data}\|p_{model}) = \mathbb{E}_{x\sim\hat{p}_{data}}\left[\text{log}\hat{p}_{data}(x) - \text{log}p_{model}(x)\right]$ .Considering $\text{log}\hat{p}_{data}(x)$ is fixed, so this problem is equivalent to minimizing $-\mathbb{E}_{x\sim\hat{p}_{data}}[\text{log}p_{model}(x)]$, which is same as the above MLE equation.In order to obtain a good estimator, we can either maximize the likelihood or minimize KL divergence. However, in practice, minimizing KL divergence is more desired for its non-negative property.

Issues with Explicit Density Models

(1) Long running time: For example, FVBNs cannot generate samples in parallel, whose running time is proportional to the dimensions of x.
(2) Too many restrictions: For instance, only a few probability distributions admit tractable Markov chain sampling in Boltzmann machines. Generator must be invertible and the latent code z must have the same dimension as the samples x in non-linear ICA.
(3) The need of Markov chains: For instance, Boltzmann machines and GSNs. Because there is no clear way to give the convergence of the results, the algorithms might not convergence.
(4) The need of variation bound and asymptotically consistent: For instance, some VAEs. The consistency issue will occurs if we accidentally select some poor posterior or prior distribution.
(5) Because the explicit density models are not as flexible as implied density model and people can not give a exactly correct density for data distribution. So, the quality of samples produced by explicit density models is worse than some implied model like GAN.

Main idea of Generative Adversarial Networks

The framework of GANs can be interpreted as a game between two players. One is called discriminator and one is called generator. The discriminator examines samples from generator whether they are real or fake, and the generator is trained to fool the discriminator. In this process, the generator can learn to create samples that are drawn from the same distribution as the training data.Formally, GANs contains latent variables z and observed variables x. Function D and G stands for the discriminator and generator and use $\theta^{(D)}$ and $\theta^{(G)}$ as parameters respectively. The purpose of both two is to minimize its own cost function $J^{(D)}(\theta^{(D)},\theta^{(G)})$ and $J^{(G)}(\theta^{(D)},\theta^{(G)})$ respectively. Both of them can only control its own parameters, and during this process, the solution is called Nash equilibrium which is $(\theta^{(D)},\theta^{(G)})$. It is a local minimum of $J^{(D)}$ with respect to $\theta^{(D)}$ and a local minimum $J^{(G)}$ with respect to $\theta^{(G)}$.

Drawbacks of GANs

(1) It needs huge data set and takes a long time to train. It also needs prodigious computational resource.
(2) Training a GAN need to find a Nash equilibrium of a game. However, sometimes gradient descent can do this, sometimes it can not. We do not have a good method for finding equilibrium.
(3) It's hard to learn to generate discrete data, like text.
(4) GANs are difficult to guess another pixel value based on one pixel value. It losses spatial correlation.