CSCI 5525 Machine Learning Fall 2019

Lecture 22: Generative Adversarial Networks

Nov 19th 2019

Lecturer: Steven Wu *Scribe: Steven Wu*

We will now introduce a different type of generative networks that do not involve evaluating the likelihood of the data under our model p_θ. This framework is called *generative adversarial networks* (GAN). There are two components in a GAN: (1) a generator and (2) a discriminator. The generator G_θ is a neural network that takes a latent vector z and deterministically maps it to sample $x = G_\theta(z)$, and the discriminator D_γ is a (probabilistic) classifier that aims to distinguish samples from the real dataset and the generator such that $D(x)$ denotes the discriminator's prediction probability of x being real.

GAN Objective. The generator and discriminator play a two player minimax game, where the generator tries to mimimc the underlying data distribution ($p_{\text{data}} = p_G$) and the discriminator tries to distinguish the samples from p_{data} versus samples from p_G. Intuitively, the generator tries to fool the discriminator to the best of its ability by generating samples that look indisginguishable from p_{data}. Formally, the GAN objective can be written as:

$$\min_\theta \max_\gamma V(G_\theta, D_\gamma) = \mathbb{E}_{x\sim p_{\text{data}}}[\ln D_\gamma(x)] + \mathbb{E}_{z\sim p_z}[\ln(1 - D_\gamma(G_\theta(z)))]$$

In this expression, the discriminator is maximizing this function V with respect to its parameters γ, where given a fixed generator G_θ it is performing binary classification to distinguish samples from p_G versus samples from p_{data}. In the homework, you will show that in this setup, the optimal discriminator is:

$$D^*_G(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}$$

where $p_G(x) = \int p_z(z)\mathbf{1}[G(z) = x]\, dz$. On the other hand, the generator minimizes this objective assuming the discriminator D_γ will best respond. And after performing some algebra, plugging in the optimal discriminator into the overall objective $V(G_\theta, D*_G(x))$ gives us:

$$2\,\text{JSD}[p_{\text{data}}, p_G] - \ln 4$$

where JSD term is the Jenson-Shannon Divergence:

$$\text{JSD}[p,q] = \frac{1}{2}\left(\text{KL}\left[p, \frac{p+q}{2}\right] + \text{KL}\left[q, \frac{p+q}{2}\right]\right)$$

JSD is a symmetric form of the KL divergence such that it satisfies all properties of the KL: $\text{JSD}(p,q) = 0$ if and only if $p = q$ and $\text{JSD}(p,q) \geq 0$ for all p, q. In addition, we get an upgrade to a symmetric form: $\text{JSD}[p,q] = \text{JSD}[q,p]$. In this case, the optimal generator for the GAN objective becomces $p_G = p_{\text{data}}$, and the optimal objective value that we can achieve with optimal generators and discriminators $G^*(\cdot)$ and $D^*_{G^*}(x)$ is $-\ln 4$. Another simple way to see this: when the generator is indeed generating samples from the distribution p_{data}, then the optimal discriminator cannot do better than predicting $D(x) = 1/2$ on every example, which gives the objective value of $-\ln 4$.

GAN Training. The training algorithm performs alternating optimization. Over iterations:

1. Sample minibatch of size m from the data set: $x^{(1)}, \ldots, x^{(m)} \sim \mathcal{D}$
2. Sample minibatch of size m of noise: $z^{(1)}, \ldots, z^{(m)} \sim p_z$
3. Take a gradient descent step on the generator parameters θ with gradient estimate:

$$\nabla_\theta V(G_\theta, D_\gamma) = \frac{1}{m} \nabla_\theta \sum_{i=1}^{m} \ln\left(1 - D_\gamma(G_\theta(z^{(i)}))\right)$$

4. Take a gradient ascent step on the discriminator parameters γ with gradient estimate:

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

Wasserstein GAN. In general, we can consider the following more general GAN objective:

$$\min_G \max_D \mathop{\mathbf{E}}_{x \sim p_X}[f(D(x))] + \mathop{\mathbf{E}}_{z \sim p_z}[f(1 - D(G(z)))] \tag{1}$$

where $f\colon [0,1] \to \mathbb{R}$ is a monotone function. For example, in standard GAN, $f(a) = \ln a$. Another popular variant of GAN is Wasserstein GAN, where $f(a) = a$. While the standard GAN objective leads to the distance of Jensen-Shannon Divergence, Wasserstein GAN leads to the *earth-mover* distance, which can be interpreted as how much mass we have to shift to convert one distribution into another.