An Introduction to Generative Adversarial Nets

John Thickstun

Suppose we want to sample from a Gaussian distribution with mean μ and variance σ^2. If we have access to samples from a standard Gaussian $\epsilon \sim \mathcal{N}(0, 1)$, then it's a standard exercise in classical statistics to show that $\mu + \sigma\epsilon \sim \mathcal{N}(\mu, \sigma^2)$. This is a simple example of a pushforward distribution. If $g(x) = \mu + \sigma x$ and $\epsilon \sim \mathcal{N}(0, 1)$, then $g(\epsilon) \sim \mathcal{N}(\mu, \sigma^2)$; the function g "pushes forward" the distribution $\mathcal{N}(0, 1)$ on its domain to the distribution $\mathcal{N}(\mu, \sigma^2)$ on its codomain.

Generative adversarial networks [Goodfellow et al., 2014] build upon this simple idea. Suppose we want to draw samples from some complicated distribution $p(x)$. Given a latent code $z \sim q$, where q is some simple distribution like $\mathcal{N}(0, I)$, we will tune the parameters of a function $g_\theta : \mathcal{Z} \to \mathcal{X}$ so that $g_\theta(z)$ is distributed approximately like p. The function g_θ is usually taken to be a neural "generator" network.

Pushforward Distributions

Pushfoward distributions are commonly seen in numerical computing, as a means to generate samples from one distribution given samples from another distribution. For example, suppose I have a software package that provides me pseudorandom samples from the uniform distribution on $[0, 1]$, but I want samples from a distribution p with a CDF given by $F : \mathbb{R} \to [0, 1]$. Define the inverse CDF by

$$F^{-1}(u) = \inf\{x : F(x) \geq u\}. \tag{1}$$

Given $u \sim \text{Uniform}([0, 1])$, it follows that $F^{-1}(u) \sim p$:

$$\Pr(F^{-1}(u) \leq x) = \Pr(u \leq F(x)) = F(x). \tag{2}$$

This technique is called inverse tranform sampling, and is based on the observation that the push-forward of the uniform distribution through the inverse-CDF of p is the distribution p. More specialized samplers like the Box-Muller transform [Box and Muller, 1958] and the Gumbel-Max trick [Gumbel, 1954] are also based on the pushforward principle.

The general idea of pushforwards is that, given a distribution on a measurable space $\mathcal{Z}$, a measurable function $g : \mathcal{Z} \to \mathcal{X}$ induces a distribution on $\mathcal{X}$. This distribution is defined, for any measurable set $A \subset \mathcal{X}$, by

$$\Pr(A) \equiv \Pr(g^{-1}(A)). \tag{3}$$

If the distribution on $\mathcal{Z}$ is absolutely continuous then we can expand the latter probability as an integral over a density $\rho(z)$. Changing variables from z to x, we find that

$$\Pr(g^{-1}(A)) = \int_{g^{-1}(A)} \rho(z)\, dz = \int_A \rho(g^{-1}(A))|\nabla_x g^{-1}(x)|\, dx. \tag{4}$$

Therefore, the density $\rho(z)$ pushes forward to a density $p_g(x)$ defined by

$$p_g(x) \equiv \rho(g^{-1}(x))|\nabla_x g^{-1}(x)|. \tag{5}$$

Learning a Pushforward Distribution

Given a finite set of samples $x_1, \ldots, x_n \sim p$ and access to unlimited samples $z \sim q$, our goal is to learn a parameterized function $g_\theta : \mathcal{Z} \to \mathcal{X}$ such that $g_\theta(z)$ is distributed approximately like $p(x)$. From the discussion in the previous section we see that, for any parameter set θ, $g_\theta(x) \sim p_\theta$ where $p_\theta(x)$ is the pushforward distribution induced by g_θ. So another way of framing our goal is that we want to find θ such that $p_\theta \approx p$.

To talk about approximations, we need to put a topology on the space of probability measures. By far the most popular topology on probabilities is the topology of KL divergence. In this setting, our goal would be to minimize $D(p \parallel p_\theta)$. This is equivalent to maximum likelihood estimation:

$$\inf_\theta D(p \parallel p_\theta) = \inf_\theta H(p) + D(p \parallel p_\theta) = \inf_\theta \mathop{\mathbb{E}}_{x \sim p} -\log \frac{p_\theta(x)}{p(x)} p(x) = \sup_\theta \mathop{\mathbb{E}}_{x \sim p} \log p_\theta(x). \tag{6}$$

This looks promising, because we can approximate the expectation using a finite sum over samples (training data) $x_i \sim p$. But there is a problem. Recall from Equation (5) that $p_\theta(x)$ is defined in terms of $g_\theta^{-1}(x)$ and $\nabla_x g_\theta^{-1}(x)$. If g_θ is a rich family of functions (e.g. a neural network) it can be very difficult to compute the inverse image of a point and its Jacobian.

From here, there are two or three directions we could take. One option is to write down restricted function families g_θ for which we can explicitly and efficiently compute inverses and Jacobians. This approach is taken by Dinh et al. [2017], Kingma and Dhariwal [2018], and trades of expressivity in the parameterization of the model for computational tractability. Another option is to try to conquer the challenge of computing inverses and Jacobians for more general function families. This approach is less well-developed, but is partially addressed by Hand and Voroninski [2019], Ma et al. [2018]. The third route is to construct an estimate of the objective, e.g. Equation (6), and optimize with respect to this proxy estimate; this later approach is taken by the Generative Adversarial Network.

Generative Adversarial Networks

A Generative Adversarial Network (GAN) is an optimization procedure for training a pushforward distribution $p_\theta(x)$ to match samples from a target distribution $x_1, \ldots, x_n \sim p$. This is made difficult because we cannot easily evaluate $p_\theta(x)$ when this distribution is implicitly defined by a complicated pushforward function $g_\theta : \mathcal{Z} \to \mathcal{X}$. The idea of GAN is to set up a saddle point problem: in the inner optimization, we attempt to construct a good lower bound on our measure of divergence between p and p_θ (e.g. the KL-divergence). In the outer optimization, we attempt to minimize this lower bound. In this section, we derive the general form of a saddle point GAN objective for a broad class information divergences known as f-divergences [Csiszár, 1964, Ali and Silvey, 1966]. Among this class are the KL-divergence based maximum likelihood estimator (6) and the Jensen-Shannon divergence used to construct the Goodfellow GAN.

An f-divergence generalizes the KL-divergence between two probability distributions. Given a convex, lower-semicontinuous function $f : \mathbb{R} \to \mathbb{R}$ such that $f(1) = 0$, we define the f-divergence between two distributions p and q by

$$D_f(p \parallel q) \equiv \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) dx. \tag{7}$$

For example, if we take $f(x) = x \log x$ then $D_f(p \parallel q) = D(p \parallel q)$. What's interesting about f-divergences is that we can construct a lower bound on the quantity $D_f(p \parallel q)$ that doesn't require

evaluation of $q(x)$ [Nguyen et al., 2010], which allows us to circumvent the challenge of evaluating f-divergences for pushforward distributions $q = p_\theta$.

The idea is to construct a variational representation of the f-divergence using a variational representation of the function f. To construct this representation, we introduce the convex conjugate of f, defined by

$$f^*(t) \equiv \sup_x \{tx - f(x)\}. \tag{8}$$

We will exploit a basic fact about convex conjugates known as "Fenchel duality" [Rockafellar, 1970]: repeat application of the conjugate operation to a convex, lower-semicontinuous function f yields $f^{**} = f$. This allows us to write a variational expression for f:

$$f(x) = \sup_t \{tx - f^*(t)\}. \tag{9}$$

In the following proposition, we see how to convert this representation of $f(x)$ into a variational representation of $D_f(p \parallel q)$.

Proposition. *[Nguyen, Wainwright, and Jordan, 2010]*

$$D_f(p \parallel q) = \sup_{T:\mathcal{X}\to\mathbb{R}} \left[\mathop{\mathbb{E}}_{x\sim p} T(x) - \mathop{\mathbb{E}}_{x\sim q} f^*(T(x)) \right]. \tag{10}$$

Proof. Using the variational representation of f given by Equation (9),

$$\begin{aligned} D_f(p \parallel q) &= \int_{\mathcal{X}} q(x) \sup_t \left[t\frac{p(x)}{q(x)} - f^*(t) \right] dx && (11)\\ &= \int_{\mathcal{X}} \sup_t \left[tp(x) - f^*(t)q(x) \right] dx && (12)\\ &= \sup_{T:\mathcal{X}\to\mathbb{R}} \int_{\mathcal{X}} \left(T(x)p(x) - f^*(T(x))q(x) \right) dx && (13)\\ &= \sup_{T:\mathcal{X}\to\mathbb{R}} \left[\mathop{\mathbb{E}}_{x\sim p} T(x) - \mathop{\mathbb{E}}_{x\sim q} f^*(T(x)) \right]. && \square \end{aligned}$$

The f-GAN uses the variational form of the f-divergence given by Equation (10) to set up a saddle point problem [Nowozin et al., 2016]. Observe that any choice of function T in Equation (10) gives us a lower bound on the f-divergence, and moreover this lower bound can be evaluated using samples from q without explicitly evaluating $q(x)$. Using an expressive parameterized family of functions T_φ to approximate the optimal function T, we can minimize an f-divergence between p and a pushforward distribution p_θ by solving the following saddle point problem:

$$\begin{aligned} \theta_f &= \arg\inf_\theta \sup_\varphi \left[\mathop{\mathbb{E}}_{x\sim p} T_\varphi(x) - \mathop{\mathbb{E}}_{x\sim p_\theta} f^*(T_\varphi(x)) \right] && (14)\\ &= \arg\inf_\theta \sup_\varphi \left[\mathop{\mathbb{E}}_{x\sim p} T_\varphi(x) - \mathop{\mathbb{E}}_{z\sim \rho} f^*(T_\varphi(g_\theta(z))) \right]. && (15) \end{aligned}$$

The Goodfellow GAN [Goodfellow et al., 2014] is an instance of the more template GAN objective given by Equation (15). To turn the template into an actual objective, we need to specify a particular f-divergence along with the parameterizations of the pushforward function f_θ and

the variational approximator T_φ. Goodfellow et. al. use a modified Jensen-Shannon divergence objective, defined by

$$\text{GAN}(p,q) \equiv 2\text{JSD}(p,q) - \log(4) = D_{\text{KL}}\left(p \middle|\middle| \frac{p+q}{2}\right) + D_{\text{KL}}\left(p_g \middle|\middle| \frac{p+q}{2}\right) - \log(4). \tag{16}$$

The GAN objective can be expressed as an f-divergence by setting $f(x) = x\log x - (x+1)\log(x+1)$, and a straightforward computation reveals that $f^*(t) = -\log(1-e^t)$. Parameterizing $T_\varphi(x) = \log(d_\varphi(x))$, from Equation (15) we find that the Goodfellow GAN objective is given by

$$\theta_f = \arg\inf_\theta \sup_\varphi \left[\mathop{\mathbb{E}}_{x\sim p} \log d_\varphi(x) + \mathop{\mathbb{E}}_{z\sim\rho} \log(1 - d_\varphi(g_\theta(z))) \right]. \tag{17}$$

The Discriminator Perspective

If you squint at Equation (17), you may notice that it looks like a binary cross-entropy loss. Let $y \sim$ Bernoulli(.5) and consider the mixture distribution $r_\theta(x)$ defined by the conditionals $r_\theta(x|y=0) = p_\theta(x)$ and $r_\theta(x|y=1) = p(x)$. We can interpret the latent variable y as a "class label," that indicates whether x comes from the pushforward distribution $p_\theta(x)$ or the target distribution $p(x)$. Defining $p_\varphi(y|x) = \text{Bernoulli}(d_\varphi(x))$ allows us to rewrite the objective of Equation (16) as a formal, conditional cross-entropy

$$\mathop{\mathbb{E}}_{\substack{y\sim\text{Bernoulli}(.5)\\ x\sim r_\theta}} \log p_\varphi(y|x) = -H(r(y|x), p_\varphi(y|x)) \leq 0. \tag{18}$$

Therefore, we can think of $d_\varphi(x)$ as a parameterization of a classifier $p_\varphi(y|x)$ that predicts whether a given point x was sampled from the data generating distribution p, or from the pushfoward distribution p_θ. This motivates the colloquial description of the network $d_\varphi(x)$ as a "discriminator."

From Equation (18), we see that the optimal discriminator that maximizes (17) for a given generator g_θ is given by the posterior distribution $r(y|x)$. This can be expressed by Bayes' rule as

$$r(y=1|x) = \frac{r(x|y=1)r(y=1)}{r(x)} = \frac{p(x)}{p(x) + p_\theta(x)}. \tag{19}$$

Plugging the optimal discriminator into (16) and manipulating the algebra, we can show that

$$\sup_\varphi \left[\mathop{\mathbb{E}}_{x\sim p} \log d_\varphi(x) + \mathop{\mathbb{E}}_{z\sim\rho} \log(1 - d_\varphi(g_\theta(z))) \right] \tag{20}$$

$$= \mathop{\mathbb{E}}_{x\sim p} \log \frac{p(x)}{p(x) + p_\theta(x)} + \mathop{\mathbb{E}}_{z\sim\rho} \log\left(1 - \frac{p(g_\theta(z))}{p(g_\theta(z)) + p_\theta(g_\theta(z))}\right) \tag{21}$$

$$= D_{\text{KL}}\left(p \middle|\middle| \frac{p+p_g}{2}\right) + D_{\text{KL}}\left(p_g \middle|\middle| \frac{p+p_g}{2}\right) - \log 4 \tag{22}$$

$$= 2\text{JSD}(p,q) - \log(4). \tag{23}$$

This is consistent with the dual calculations performed in the previous Section.

References

Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1966. (document)

G. E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 1958. (document)

Imre Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 1964. (document)

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *International Conference on Learning Representations*, 2017. (document)

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014. (document)

E. J. Gumbel. Statistical theory of extreme values and some practical applications: a series of lectures. *US Government Printing Office*, 1954. (document)

Paul Hand and Vladislav Voroninski. Global guarantees for enforcing deep generative priors by empirical risk. *IEEE Transactions on Information Theory*, 2019. (document)

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018. (document)

Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. In *Advances in Neural Information Processing Systems*, 2018. (document)

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 2010. (document)

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, 2016. (document)

R Tyrrell Rockafellar. *Convex analysis*. Princeton university press, 1970. (document)