Deep Generative Models

9. Generative Adversarial Networks

- 국가수리과학연구소 산업수학혁신센터 김민중

Two-sample test via a discriminator

- **Training objective for discriminator**

$$\max_{D_\phi} V(p_\theta, D_\phi) = \max_{D_\phi} E_{\boldsymbol{x} \sim p_{data}}\left[\log D_\phi(\boldsymbol{x})\right] + E_{\boldsymbol{x} \sim p_\theta}\left[\log\left(1 - D_\phi(\boldsymbol{x})\right)\right]$$

$$\approx \max_{D_\phi} \sum_{x \in S_1} \log D_\phi(\boldsymbol{x}) + \sum_{x \in S_2} \log\left(1 - D_\phi(\boldsymbol{x})\right)$$

- For a fixed generative model p_θ, the discriminator is performing binary classification with the cross-entropy objective
  - Assign probability 1 to true data points $\boldsymbol{x} \sim p_{data}$ (in set S_1)
  - Assign probability 0 to fake samples $\boldsymbol{x} \sim p_\theta$ (in set S_2)

Two-sample test via a discriminator

- **Training objective for discriminator**

$$\max_{D_\phi} V(p_\theta, D_\phi) = \max_{D_\phi} E_{\boldsymbol{x}\sim p_{data}}\left[\log D_\phi(\boldsymbol{x})\right] + E_{\boldsymbol{x}\sim p_\theta}\left[\log\left(1 - D_\phi(\boldsymbol{x})\right)\right]$$

$$\approx \max_{D_\phi} \sum_{x\in S_1} \log D_\phi(\boldsymbol{x}) + \sum_{x\in S_2} \log\left(1 - D_\phi(\boldsymbol{x})\right)$$

- For a fixed generative model p_θ, the optimal discriminator is given by

$$D_\theta^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_\theta(\boldsymbol{x})}$$

- If $p_\theta = p_{data}$, classifier cannot do better than chance ($D_\theta^*(\boldsymbol{x}) = 1/2$)

Generative Adversarial Networks

- A two-player minimax game between a generator and a discriminator
- **Generator**
  - Directed latent variable model with a deterministic mapping between $\mathbf{z}$ and $\mathbf{x}$ given by G_θ
    - Sample $\mathbf{z} \sim p_Z$, where p_Z is a simple prior, e.g., Gaussian
    - Set $\mathbf{x} = G_\theta(\mathbf{z})$
  - Like a flow model, but mapping G_θ need not be invertible
  - Distribution over $p_\theta(\mathbf{x})$ over $\mathbf{x}$ is implicitly defined (no likelihood!)
  - Minimizes a two-sample test objective (in support of the null hypothesis $p_{data} = p_\theta$)

Example of GAN objective

- **Training objective**

$$\min_{G} \max_{D} V(G, D) = \min_{G} \max_{D} E_{\boldsymbol{x} \sim p_{data}}[\log D(\boldsymbol{x})] + E_{\boldsymbol{x} \sim p_G}\left[\log\left(1 - D(\boldsymbol{x})\right)\right]$$

- For the optimal discriminator $D_G^*(\cdot)$, we have

$$\begin{aligned}
&V(G, D_G^*) \\
&= E_{\boldsymbol{x} \sim p_{data}}\left[\log \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}\right] + E_{\boldsymbol{x} \sim p_G}\left[\log \frac{p_G(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}\right] \\
&= E_{\boldsymbol{x} \sim p_{data}}\left[\log \frac{p_{data}(\boldsymbol{x})}{\frac{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}{2}}\right] + E_{\boldsymbol{x} \sim p_G}\left[\log \frac{p_G(\boldsymbol{x})}{\frac{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}{2}}\right] - \log 4 \\
&= D\left(p_{data} \parallel \frac{p_{data} + p_G}{2}\right) + D\left(p_G \parallel \frac{p_{data} + p_G}{2}\right) - \log 4 \\
&= 2JSD(p_{data} \parallel p_G) - \log 4
\end{aligned}$$

The GAN training algorithm

- Sample minibatch of n training points $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(n)}$ from p_{data}
- Sample minibatch of n noise vectors $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \cdots, \boldsymbol{z}^{(n)}$ from p_Z
- Update the discriminator parameters $\boldsymbol{\phi}$ by stochastic gradient ascent

$$\nabla_{\boldsymbol{\phi}} V(G_\theta, \boldsymbol{D}_{\boldsymbol{\phi}}) = \frac{1}{n} \nabla_{\boldsymbol{\phi}} \sum_{i=1}^{n} \left[\log \boldsymbol{D}_{\boldsymbol{\phi}}(\boldsymbol{x}^{(i)}) + \log\left(1 - \boldsymbol{D}_{\boldsymbol{\phi}}\left(G_\theta(\boldsymbol{z}^{(i)})\right)\right)\right]$$

- Update the generator parameters $\boldsymbol{\theta}$ by stochastic gradient descent

$$\nabla_{\boldsymbol{\theta}} V(\boldsymbol{G}_{\boldsymbol{\theta}}, D_\phi) = \frac{1}{n} \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log\left(1 - D_\phi\left(\boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i)})\right)\right)$$

- Repeat for fixed number of epochs

Recap of GANs

- Choose $d(p_{data}, p_\theta)$ to be a two-sample test statistic
  - Learn the statistic by training a classifier (discriminator)
  - Under ideal conditions, equivalent to choosing $d(p_{data}, p_\theta)$ to be $JSD(p_{data} \parallel p_\theta)$
- Generator G_θ(e.g., neural network) is a mapping that generates $\boldsymbol{x}$ from the latent variable $\boldsymbol{z}$ and is trained to make it difficult for the classifier to distinguish

Recap of GANs

- Pros:
  - Loss only requires samples from p_θ (No likelihood needed!)
  - Lots of flexibility for the neural network architecture, any G_θ defines a valid sampling procedure
  - Fast sampling (single forward pass)
- Cons: very difficult to train in practice

Summary of GANs

- Likelihood-free training
- Training objective for GANs

$$V(G,D) = E_{x\sim p_{data}}[\log D(\boldsymbol{x})] + E_{x\sim p_G}\left[\log\left(1 - D(\boldsymbol{x})\right)\right]$$

- With the optimal discriminator D_G^*, we see GAN minimizes a scaled and shifted Jensen-Shannon divergence

$$\min_G 2JSD(p_{data} \parallel p_G) - \log 4$$

- Parameterize D by ϕ and G by θ
- Prior distribution p_Z

$$\min_\theta \max_\phi E_{x\sim p_{data}}[\log D_\phi(\boldsymbol{x})] + E_{z\sim p_Z}\left[\log\left(1 - D_\phi\left(G_\theta(\boldsymbol{z})\right)\right)\right]$$

- I.e.,

$$V(G_\theta, D_\phi) = \frac{1}{n}\sum_{i=1}^{n}\left[\log D_\phi\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D_\phi\left(G_\theta\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$$

Beyond KL and Jenson-Shannon Divergence

- What choices do we have for $d(\cdot)$?
  - KL divergence: Autoregressive models, Flow models
  - (scaled and shifted) Jensen-Shannon divergence (approximately): original GAN objective

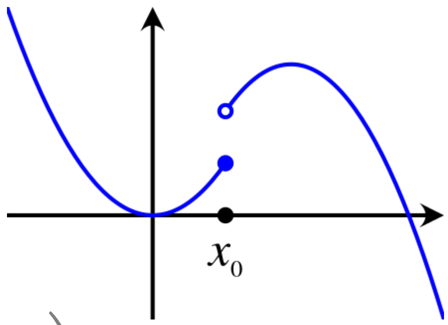
f-divergences

- What choices do we have for $d(\cdot)$?
- Given two densities p and q, the f-divergence is given by

$$D_f(p,q) = E_{\boldsymbol{x}\sim q}\left[f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)\right]$$

- Where f is any convex, lower-semicontinuous function with $f(1) = 0$
- Convex: Line joining any two points lies above the function
- Lower-semicontinuous

$$\liminf_{x\to x_0} f(x) \geq f(x_0)$$



- for any point x_0
- Jensen's inequality

$$E_{\boldsymbol{x}\sim q}\left[f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)\right] \geq f\left(E_{\boldsymbol{x}\sim q}\left[\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right]\right) = f\left(\int p(\boldsymbol{x})d\boldsymbol{x}\right) = f(1) = 0$$

- Example: KL divergence with $f(u) = u \log u$

f-divergences

Name	$D_f(P\|Q)$	Generator $f(u)$
Total variation	$\frac{1}{2}\int \lvert p(x)-q(x)\rvert \,\mathrm{d}x$	$\frac{1}{2}\lvert u-1\rvert$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$	$u\log u$
Reverse Kullback-Leibler	$\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$	$(u-1)^2$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x$	$\frac{(1-u)^2}{u}$
Squared Hellinger	$\int \left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\,\mathrm{d}x$	$\left(\sqrt{u}-1\right)^2$
Jeffrey	$\int (p(x)-q(x))\log\left(\frac{p(x)}{q(x)}\right)\,\mathrm{d}x$	$(u-1)\log u$
Jensen-Shannon	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$	$-(u+1)\log\frac{1+u}{2}+u\log u$
Jensen-Shannon-weighted	$\int p(x)\pi\log\frac{p(x)}{\pi p(x)+(1-\pi)q(x)}+(1-\pi)q(x)\log\frac{q(x)}{\pi p(x)+(1-\pi)q(x)}\,\mathrm{d}x$	$\pi u\log u-(1-\pi+\pi u)\log(1-\pi+\pi u)$
GAN	$\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x-\log(4)$	$u\log u-(u+1)\log(u+1)$
α-divergence ($\alpha\notin\{0,1\}$)	$\frac{1}{\alpha(\alpha-1)}\int\left(p(x)\left[\left(\frac{q(x)}{p(x)}\right)^\alpha-1\right]-\alpha(q(x)-p(x))\right)\mathrm{d}x$	$\frac{1}{\alpha(\alpha-1)}\left(u^\alpha-1-\alpha(u-1)\right)$

Source: Nowozin et al., 2017

Training with f-divergences

- Given p_{data} and p_θ, we could minimize $D_f(p_{data}, p_\theta)$ or $D_f(p_\theta, p_{data})$ as learning objectives. Non-negative and zero if $p_\theta = p_{data}$
- However, it depends on the density ratio which is **unknown**

$$D_f(p_\theta, p_{data}) = E_{\boldsymbol{x} \sim p_{data}}\left[f\left(\frac{p_\theta(\boldsymbol{x})}{p_{data}(\boldsymbol{x})}\right)\right]$$

$$D_f(p_{data}, p_\theta) = E_{\boldsymbol{x} \sim p_\theta}\left[f\left(\frac{p_{data}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})}\right)\right]$$

- To use f-divergences as a two-sample test objective for likelihood-free learning, we need to be able to estimate the objective using only samples (e.g., training data and samples from the model)

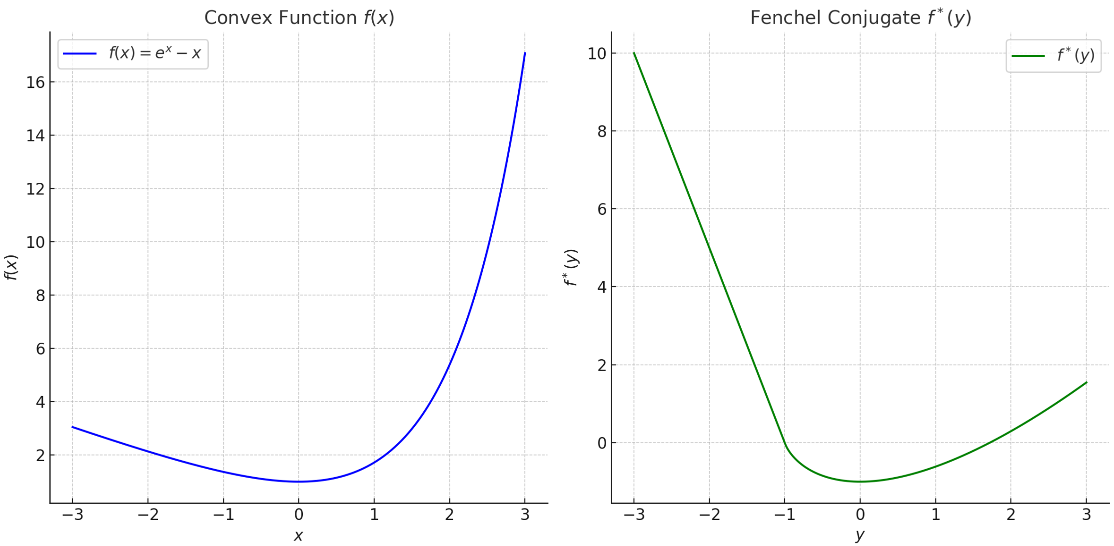
Towards Variational Divergence Minimization

- Fenchel conjugate: for any function $f(\cdot)$, its convex conjugate is

$$f^*(t) \coloneqq \sup_{u \in dom_f} (ut - f(u))$$

where dom_f is the domain of the function f

- f^* is convex (pointwise supremum of convex functions is convex) and lower semi-continuous



Towards Variational Divergence Minimization

- Let f^{**} be the Fenchel conjugate of f^*
$$f^{**}(u) := \sup_{t \in dom_{f^*}} \left(tu - f^*(t)\right)$$
- $f^{**} \leq f$. Proof: By definition, for all t, u
$$f^*(t) \geq ut - f(u) \text{ or equivalently } f(u) \geq ut - f^*(t)$$
$$f(u) \geq \sup_{t \in dom_{f^*}} \left(ut - f^*(t)\right) = f^{**}(u)$$
- Strong Duality: $f^{**} = f$ when $f(\cdot)$ is convex and lower semicontinuous

f-GAN: Variational Divergence Minimization

- We obtain a lower bound to an f-divergence via Fenchel conjugate

$$D_f(p,q) = E_{\boldsymbol{x}\sim q}\left[f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)\right] = E_{\boldsymbol{x}\sim q}\left[f^{**}\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)\right]$$

$$= E_{\boldsymbol{x}\sim q}\left[\sup_{t\in dom_{f^*}}\left(t\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} - f^*(t)\right)\right]$$

$$\geq \sup_{T\in\mathcal{T}}\int_{\mathcal{X}}\left[T(\boldsymbol{x})p(\boldsymbol{x}) - f^*\big(T(\boldsymbol{x})\big)q(\boldsymbol{x})\right]d\boldsymbol{x}$$

$$= \sup_{T\in\mathcal{T}}\Big(E_{\boldsymbol{x}\sim p}[T(\boldsymbol{x})] - E_{\boldsymbol{x}\sim q}[f^*(T(\boldsymbol{x}))]\Big)$$

- where $\mathcal{T}: \mathcal{X} \to \mathbb{R}$ is an arbitrary class of functions
- **Note**: Lower bound is likelihood-free w.r.t. p and q

f-GAN: Variational Divergence Minimization

- Variational lower bound

$$D_f(p,q) \geq \sup_{T\in\mathcal{T}}\left(E_{x\sim p}[T(\boldsymbol{x})] - E_{\boldsymbol{x}\sim q}[f^*(T(\boldsymbol{x})]\right)$$

- Choose an f-divergence
- Let $p = p_{data}$ and $q = p_G$
- Parameterize T by ϕ and G by θ
- Consider the following f-GAN object

$$\min_{\theta}\max_{\phi} F(\theta,\phi) = \min_{\theta}\max_{\phi} E_{\boldsymbol{x}\sim p_{data}}\left[T_\phi(\boldsymbol{x})\right] - E_{\boldsymbol{x}\sim p_{G_\theta}}\left[f^*\left(T_\phi(\boldsymbol{x})\right)\right]$$

- Generator G_θ tries to minimize the divergence estimate and discriminator T_ϕ tries to tighten the lower bound
- Substitute any f-divergence and optimize the f-GAN objective
- Prior distribution p_Z

$$\min_{\theta}\max_{\phi} E_{\boldsymbol{x}\sim p_{data}}\left[T_\phi(\boldsymbol{x})\right] - E_{\boldsymbol{z}\sim p_Z}\left[f^*\left(T_\phi(G_\theta(\boldsymbol{z}))\right)\right]$$

Example: Univariate Mixture of Gaussians

- p_{data}: a mixture of Gaussians
- Model Q_θ using linear transformation of a standard normal $z \sim N(0,1)$ and outputs $G_\theta(z) = \mu + \sigma z$, where $\theta = (\mu, \sigma)$

	KL	KL-rev	JS	Jeffrey	Pearson
$D_f(P\|\|Q_{\theta^*})$	0.2831	0.2480	0.1280	0.5705	0.6457
$F(\hat{\omega}, \hat{\theta})$	0.2801	0.2415	0.1226	0.5151	0.6379
μ^*	1.0100	1.5782	1.3070	1.3218	0.5737
$\hat{\mu}$	1.0335	1.5624	1.2854	1.2295	0.6157
σ^*	1.8308	1.6319	1.7542	1.7034	1.9274
$\hat{\sigma}$	1.8236	1.6403	1.7659	1.8087	1.9031

train \ test	KL	KL-rev	JS	Jeffrey	Pearson
KL	**0.2808**	0.3423	0.1314	0.5447	0.7345
KL-rev	0.3518	**0.2414**	0.1228	0.5794	1.3974
JS	0.2871	0.2760	**0.1210**	0.5260	0.92160
Jeffrey	0.2869	0.2975	0.1247	**0.5236**	0.8849
Pearson	0.2970	0.5466	0.1665	0.7085	**0.648**

Table 3: Gaussian approximation of a mixture of Gaussians. Left: optimal objectives, and the learned mean and the standard deviation: $\hat{\theta} = (\hat{\mu}, \hat{\sigma})$ (learned) and $\theta^* = (\mu^*, \sigma^*)$ (best fit). Right: objective values to the true distribution for each trained model. For each divergence, the lowest objective function value is achieved by the model that was trained for this divergence.

Source: Nowozin et al., 2017

Beyond KL and Jensen-Shannon Divergence

- What choices do we have for $d(\cdot,\cdot)$?
  - KL divergence: Autoregressive Models, Flow models
  - (scaled and shifted) Jensen-Shannon divergence (approximately): via the original GAN objective
  - Any other f-divergence (approximately): via the f-GAN objective

$$\min_{\theta} \max_{\phi} F(\theta, \phi) = \min_{\theta} \max_{\phi} E_{\boldsymbol{x} \sim p_{data}}\left[T_{\phi}(\boldsymbol{x})\right] - E_{\boldsymbol{x} \sim p_{G_{\theta}}}\left[f^{*}\left(T_{\phi}(\boldsymbol{x})\right)\right]$$

Wasserstein GAN: beyond f-divergence

- The f-divergence is defined as

$$D_f(p,q) = E_{x\sim q}\left[f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)\right]$$

- The support of q must cover the support of p. Otherwise, discontinuity arises in f-divergences
- E.g.,
  - Let $p(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases}$ and $q_\theta(x) = \begin{cases} 1, & x = \theta \\ 0, & x \neq \theta \end{cases}$
  - $D_{KL}(p, q_\theta) = \begin{cases} 0, & \theta = 0 \\ \infty, & \theta \neq 0 \end{cases}$
  - $D_{JS}(p, q_\theta) = \begin{cases} 0, & \theta = 0 \\ \log 2, & \theta \neq 0 \end{cases}$
  - We need a "smoother" distance $\mathrm{D}(p,q)$ that is defined when p and q have disjoint supports

Wasserstein (Earth-Mover) distance

- Introduced by Leonid Vaseršteĭn(Russia)
- 1st Wasserstein distance

$$D_W(p,q) := \inf_{\gamma \in \Gamma(p,q)} \int_{K \times K} |\boldsymbol{x} - \boldsymbol{y}| d\gamma(\boldsymbol{x}, \boldsymbol{y})$$
$$= \inf_{\gamma \in \Gamma(p,q)} \sum_{\boldsymbol{x},\boldsymbol{y}} |\boldsymbol{x} - \boldsymbol{y}| \gamma(\boldsymbol{x}, \boldsymbol{y})$$

- where $\Gamma(p,q)$ contains all joint distributions of $(\boldsymbol{x}, \boldsymbol{y})$ where the marginal of $\boldsymbol{x}$ is $p(\boldsymbol{x})$ and the marginal of $\boldsymbol{y}$ is $q(\boldsymbol{y})$
- $\gamma(\boldsymbol{y}|\boldsymbol{x})$: a probabilistic earth moving plan that warps $p(\boldsymbol{x})$ to $q(\boldsymbol{y})$
- Let $p(x) = \begin{cases} 1, & x = 0 \\ 0, & x = 1 \end{cases}$ and $q_\theta(x) = \begin{cases} 1, & x = \theta \\ 0, & x \neq \theta \end{cases}$
- $D_W(p, q_\theta) = |\theta|$

Wasserstein GAN (WGAN)

- Kantorovich-Rubinstein duality
$$D_W(p,q) = \sup_{\|f\|_L \le 1} E_{\boldsymbol{x}\sim p}[f(\boldsymbol{x})] - E_{\boldsymbol{x}\sim q}[f(\boldsymbol{x})]$$
- $\|f\|_L \le 1$ means the Lipschitz constant of $f(\boldsymbol{x})$ is 1. I.e.,
$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \le \|\boldsymbol{x} - \boldsymbol{y}\|_1 \quad \forall \boldsymbol{x}, \boldsymbol{y}$$
- Intuitively, f cannot change too rapidly
- Wasserstein GAN with discriminator $D_\phi(\boldsymbol{x})$ and generator $G_\theta(\boldsymbol{z})$
$$\min_{\theta} \max_{\phi} E_{\boldsymbol{x}\sim p_{data}}\left[D_\phi(\boldsymbol{x})\right] - E_{\boldsymbol{z}\sim p_Z}\left[D_\phi\left(G_\theta(\boldsymbol{z})\right)\right]$$
- Lipschitzness of $D_\phi(\boldsymbol{x})$ is enforced through **weight clipping** or **gradient penalty** on $\nabla_x D_\phi(\boldsymbol{x})$
- To enforce Lipschitz constraint, clip the weights of the critic to lie within a compact space $[-c, c]$. The set of functions satisfying this constraint is a subset of the K-Lipschitz functions for some $K(c)$
- If we replace $\|f\|_L \le 1$ for $\|f\|_L \le K$, then we end up with $K \cdot D_W(p,q)$

Wasserstein GAN Gradient Penalty

- Wasserstein GAN-GP

$$\min_{\theta}\max_{\phi} E_{x\sim p_{data}}\left[D_{\phi}(x)\right] - E_{z\sim p_Z}\left[D_{\phi}\left(G_{\theta}(z)\right)\right]$$
$$- \lambda E_{\hat{x}\sim p_{\hat{X}}}\left[\left(\left\|\nabla_{\hat{x}} D_{\phi}(\hat{x})\right\|_2 - 1\right)^2\right]$$

Proposition 1. *Let $\mathbb{P}_r$ and $\mathbb{P}_g$ be two distributions in $\mathcal{X}$, a compact metric space. Then, there is a 1-Lipschitz function f^* which is the optimal solution of $\max_{\|f\|_L \le 1} \mathbb{E}_{y\sim\mathbb{P}_r}[f(y)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)]$. Let π be the optimal coupling between $\mathbb{P}_r$ and $\mathbb{P}_g$, defined as the minimizer of: $W(\mathbb{P}_r,\mathbb{P}_g) = \inf_{\pi\in\Pi(\mathbb{P}_r,\mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\pi}[\|x-y\|]$ where $\Pi(\mathbb{P}_r,\mathbb{P}_g)$ is the set of joint distributions $\pi(x,y)$ whose marginals are $\mathbb{P}_r$ and $\mathbb{P}_g$, respectively. Then, if f^* is differentiable[‡], $\pi(x=y)=0$[§], and $x_t = tx + (1-t)y$ with $0 \le t \le 1$, it holds that $\mathbb{P}_{(x,y)\sim\pi}\left[\nabla f^*(x_t) = \frac{y-x_t}{\|y-x_t\|}\right] = 1$.*

Corollary 1. *f^* has gradient norm 1 almost everywhere under $\mathbb{P}_r$ and $\mathbb{P}_g$.*

Wasserstein GAN Gradient Penalty

- Wasserstein GAN-GP

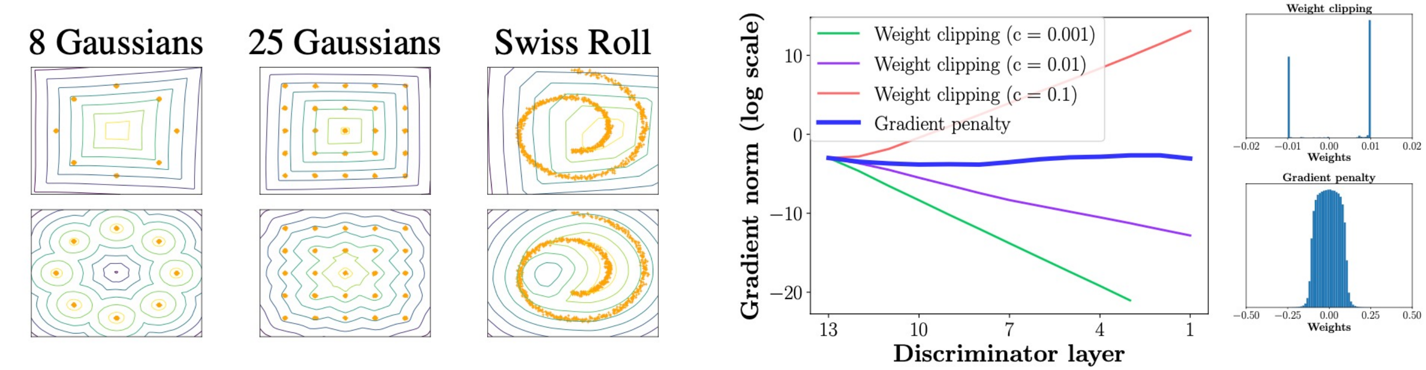
$$\min_{\theta}\max_{\phi} E_{\boldsymbol{x}\sim p_{data}}\left[D_{\phi}(\boldsymbol{x})\right] - E_{\boldsymbol{z}\sim p_{Z}}\left[D_{\phi}\left(G_{\theta}(\boldsymbol{z})\right)\right] - \lambda E_{\widehat{\boldsymbol{x}}\sim p_{\widehat{X}}}\left[\left(\left\|\nabla_{\widehat{\boldsymbol{x}}} D_{\phi}(\widehat{\boldsymbol{x}})\right\|_2 - 1\right)^2\right]$$

- **Sampling distribution** $p_{\widehat{X}}$: uniformly along straight lines between pairs of points sampled from the data distribution and the generator distribution

Wasserstein GAN Gradient Penalty

- Wasserstein GAN-GP

$$\min_{\theta}\max_{\phi} E_{x\sim p_{data}}\left[D_{\phi}(\boldsymbol{x})\right] - E_{z\sim p_Z}\left[D_{\phi}\left(G_{\theta}(\boldsymbol{z})\right)\right]$$
$$- \lambda E_{\hat{x}\sim p_{\hat{X}}}\left[\left(\left\|\nabla_{\hat{x}} D_{\phi}(\hat{\boldsymbol{x}})\right\|_2 - 1\right)^2\right]$$



(a) Value surfaces of WGAN critics trained to optimality on toy datasets using (top) weight clipping and (bottom) gradient penalty. Critics trained with weight clipping fail to capture higher moments of the data distribution. The 'generator' is held fixed at the real data plus Gaussian noise.

(b) (left) Gradient norms of deep WGAN critics during training on the Swiss Roll dataset either explode or vanish when using weight clipping, but not when using a gradient penalty. (right) Weight clipping (top) pushes weights towards two values (the extremes of the clipping range), unlike gradient penalty (bottom).

Figure 1: Gradient penalty in WGANs does not exhibit undesired behavior like weight clipping.

Inferring latent representations in GANs

- The generator of a GAN is typically a directed, latent variable model with latent variables $\boldsymbol{z}$ and observed variables $\boldsymbol{x}$
- How can we infer the **latent feature representations** in a GAN?
- Unlike a normalizing flow model, the mapping $G: \boldsymbol{z} \rightarrow \boldsymbol{x}$ need not be invertible
- Unlike a variational autoencoder, there is no inference network $q(\cdot)$ which can learn a variational posterior over latent variables

Inferring latent representations in GANs

- **Solution 1**: For any point x, use the activations of the prefinal layer of a **discriminator** as a feature representation
- **Intuition**: Like supervised deep neural networks, the discriminator would have learned useful representations for x while distinguishing real and fake

Inferring latent representations in GANs

- If we want to directly infer the latent variables $\boldsymbol{z}$ of the generator, we need a **different learning algorithm**
- A regular GAN optimizes a two-sample test objective that compares samples of $\boldsymbol{x}$ from the generator and the data distribution
- **Solution 2**: To infer latent representations, we will compare samples of $\boldsymbol{x}$, $\boldsymbol{z}$ from the joint distributions of observed and latent variables as per the model and the data distribution
- For any $\boldsymbol{x}$ generated via the model, we have access to $\boldsymbol{z}$ (sampled from a simple prior $p(\boldsymbol{z})$)
- For any $\boldsymbol{x}$ from the data distribution, $\boldsymbol{z}$ is however unobserved (latent). Need an encoder

Summary of Generative Adversarial Networks

- **Key observation**: Samples and likelihoods are not correlated in practice
- Two-sample test objectives allow for learning generative models only via samples (likelihood-free)
- Wide range of two-sample test objectives covering f-divergences and Wasserstein distances (and more)

Thanks