Deep Generative Models

8. Generative Adversarial Networks

- 국가수리과학연구소 산업수학혁신센터 김민중

Recap

- Model Families
  - Autoregressive Models

$$p_\theta(\boldsymbol{x}) = \prod_{i=1}^{d} p_\theta(x_i | \boldsymbol{x}_{<i})$$

  - Variational Autoencoders

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$$

  - Normalizing Flow Models

$$p_X(\boldsymbol{x}; \theta) = p_Z\left(\boldsymbol{f}_\theta^{-1}(\boldsymbol{x})\right)\left|\det\left(\frac{\partial \boldsymbol{f}_\theta^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)\right|$$

Recap

- All the above families are trained by minimizing **KL divergence** $\boldsymbol{D}(p_{data} \parallel p_\theta)$ or equivalently maximizing **likelihoods** (or approximations)

Why maximum likelihood?

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{M} \log p_\theta\left(\boldsymbol{x}^{(i)}\right), \qquad \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(M)} \sim p_{data}$$

- Optimal statistical efficiency
  - Assume sufficient model capacity, such that there exists a unique $\theta^* \in \mathcal{M}$ that satisfies $p_{\theta^*} = p_{data}$
  - The convergence of $\hat{\theta}$ to θ^* when $M \to \infty$ is the "fastest" among all statistical methods when using maximum likelihood training
- Higher likelihood = better lossless compression
- Is the likelihood a good indicator of the quality of samples generated by the model?

Recap

- Model Families
  - Autoregressive Models: $p_\theta(\boldsymbol{x}) = \prod_{i=1}^{d} p_\theta(x_i | \boldsymbol{x}_{<\boldsymbol{i}})$
  - Variational Autoencoders: $p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$
  - Normalizing Flow Models: $p_X(\boldsymbol{x}; \theta) = p_Z\left(\boldsymbol{f}_\theta^{-1}(\boldsymbol{x})\right)\left|\det\left(\frac{\partial \boldsymbol{f}_\theta^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)\right|$
- All the above families are trained by minimizing KL divergence $D(p_{data} \parallel p_\theta)$ or equivalently maximizing likelihoods (or approximations)
- Today: alternative for $D(p_{data} \parallel p_\theta)$

Comparing distributions via samples

- Given a **finite set of samples** from two distributions $S_1 = \{\boldsymbol{x}{\sim}P\}$ and $S_2 = \{\boldsymbol{x}{\sim}Q\}$, how can we tell if these samples are from the same distribution? (i.e., $P = Q$?)

$S_1 = \{\boldsymbol{x}{\sim}P\}$

vs.

$S_2 = \{\boldsymbol{x}{\sim}Q\}$

Two-sample tests

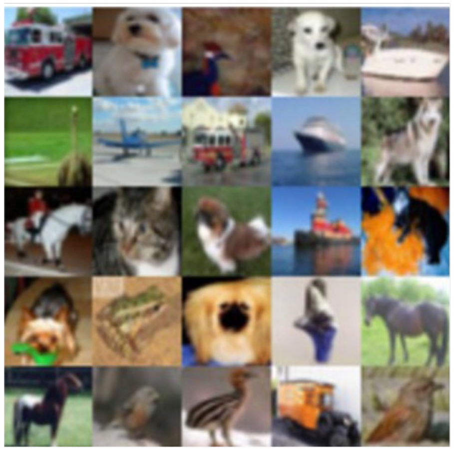
- $S_1 = \{\boldsymbol{x} \sim P\}$ and $S_2 = \{\boldsymbol{x} \sim Q\}$
- Test statistic T compares S_1 and S_2. Using T, determine $P = Q$ or not
  - E.g.,

$$T(S_1, S_2) = \left| \frac{1}{|S_1|} \sum_{\boldsymbol{x} \in S_1} \boldsymbol{x} - \frac{1}{|S_2|} \sum_{\boldsymbol{x} \in S_2} \boldsymbol{x} \right|$$

- If T is large enough, then we determine $P \neq Q$ otherwise we say $P = Q$
- Key observation: Test statistic is likelihood-free since it does not involve the densities P or Q (only samples)

Generative modeling and two-sample tests

- A priori, we assume direct access to $S_1 = D = \{\boldsymbol{x} \sim p_{data}\}$
- In addition, we have a model distribution p_θ
- Assume that the model distribution permits efficient sampling. Let $S_2 = \{\boldsymbol{x} \sim p_\theta\}$
- Alternative notion of distance between distributions:
  - Train the generative model to minimize a two-sample test objective between S_1 and S_2



vs.

$S_1 = \{\boldsymbol{x} \sim p_{data}\}$ $S_2 = \{\boldsymbol{x} \sim p_\theta\}$

Two-sample test

- In the generative model setup, we know that S_1 and S_2 come from different distributions p_{data} and p_θ respectively
- **Key idea**: Learn a statistic to automatically identify in what way the two sets of samples S_1 and S_2 differ from each other
- How? Train a classifier (called a discriminator)!

$S_1 = \{\boldsymbol{x} \sim p_{data}\}$ $S_2 = \{\boldsymbol{x} \sim p_\theta\}$

Two-sample test via a discriminator

- Any **binary classifier D_ϕ(e.g., neural network)** which tries to distinguish "real" $(y = 1)$ samples from the dataset and "fake" $(y = 0)$ samples generated from the model
- Test statistic: $-\mathrm{loss}$ of the classifier.
  - Low loss, real and fake samples are easy to distinguish (different)
  - High loss, real and fake samples are hard to distinguish (similar)
- **Goal**
  - Maximize the two-sample test statistic (in support of the alternative hypothesis $p_{data} \neq p_\theta$), or equivalently minimize classification loss

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**

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

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

- 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 $\boldsymbol{z}$ and $\boldsymbol{x}$ given by G_θ
    - Sample $\boldsymbol{z} \sim p_Z$, where p_Z is a simple prior, e.g., Gaussian
    - Set $\boldsymbol{x} = G_\theta(\boldsymbol{z})$
  - Like a flow model, but mapping G_θ need not be invertible
  - Distribution over $p_\theta(\boldsymbol{x})$ over $\boldsymbol{x}$ 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 for generator**

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

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

$$V(G, D_G^*)$$

$$= E_{\boldsymbol{x} \sim p_{data}}\left[\log \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}\right] + E_{\boldsymbol{x} \sim p_G}\left[\log \frac{p_G(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}\right]$$

$$= E_{\boldsymbol{x} \sim p_{data}}\left[\log \frac{p_{data}(\boldsymbol{x})}{\frac{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}{2}}\right] + E_{\boldsymbol{x} \sim p_G}\left[\log \frac{p_G(\boldsymbol{x})}{\frac{p_{data}(\boldsymbol{x}) + p_G(\boldsymbol{x})}{2}}\right] - \log 4$$

$$= D\left(p_{data} \parallel \frac{p_{data} + p_G}{2}\right) + D\left(p_G \parallel \frac{p_{data} + p_G}{2}\right) - \log 4$$

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

Jensen-Shannon Divergence

- Also called as the symmetric KL divergence

$$JSD(p \parallel q) = \frac{1}{2} D\left(p \parallel \frac{p+q}{2}\right) + \frac{1}{2} D\left(q \parallel \frac{p+q}{2}\right)$$

- Properties
  - $JSD(p \parallel q) \geq 0$
  - $JSD(p \parallel q) = 0$ iff $p = q$
  - $JSD(p \parallel q) = JSD(q \parallel p)$
  - $\sqrt{JSD(p \parallel q)}$ satisfies triangle inequality. I.e., it is a distance
- Optimal generator for the JSD/Negative Cross Entropy GAN

$$p_G = p_{data}$$

- For the optimal discriminator $D^*_{G^*}(\cdot)$ and generator $G^*(\cdot)$, we have

$$V(G^*, D^*_{G^*}) = -\log 4$$

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 $\boldsymbol{x}$ and is trained to make it difficult for the classifier to distinguish
- 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

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 ϕ by stochastic gradient ascent

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

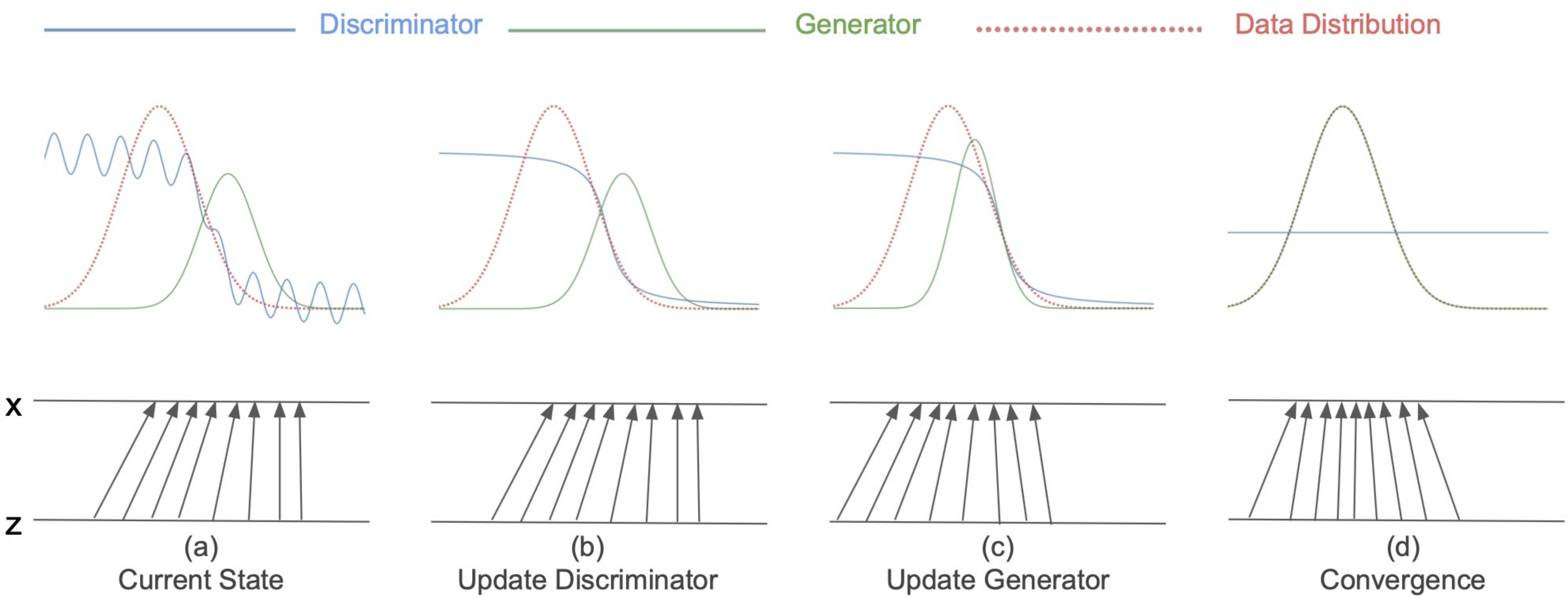
- Update the generator parameters θ by stochastic gradient descent

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

- Repeat for fixed number of epochs

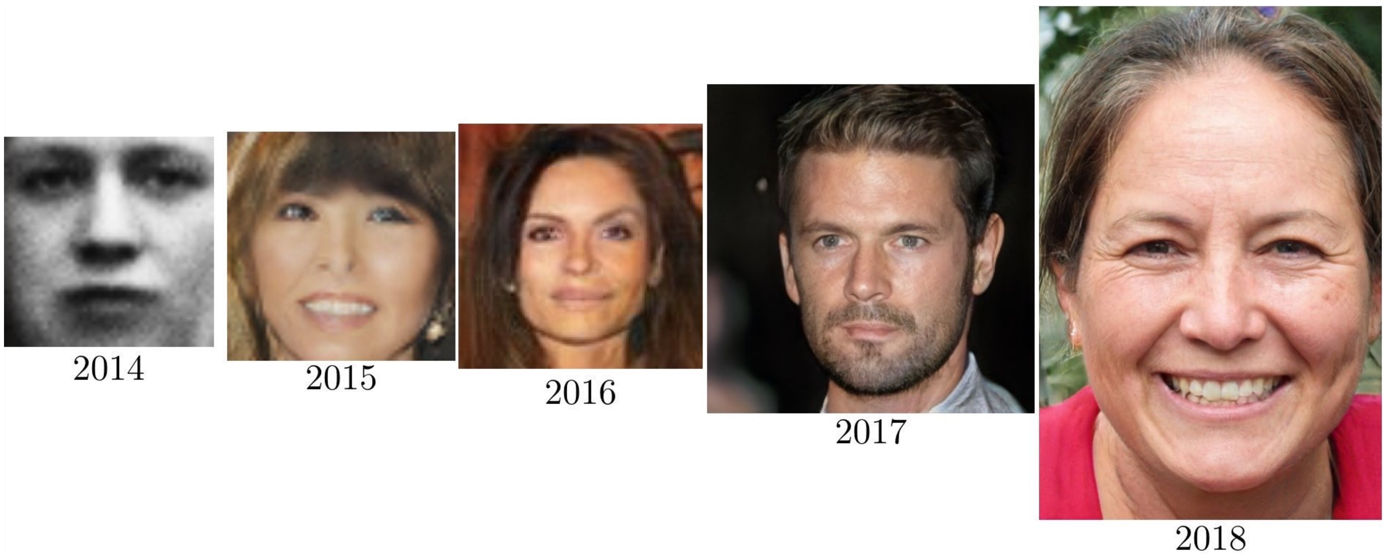
Alternating optimization in GANs

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



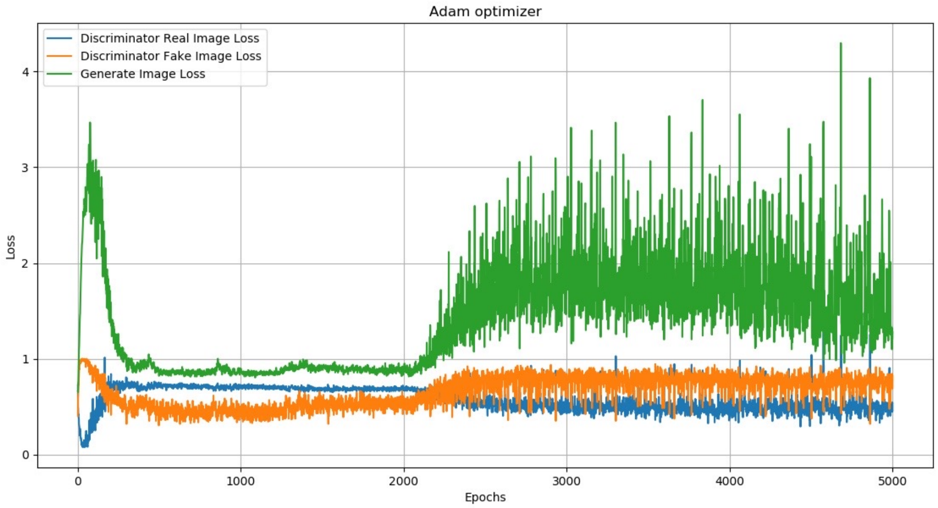
Frontiers in GAN research

- GANs have been successfully applied to several domains and tasks
- However, working with GANs can be very challenging in practice
  - Unstable optimization
  - Mode collapse Evaluation
  - Bag of tricks needed to train GANs successfully



Optimization challenges

- **Theorem (informal)**: If the generator updates are made in function space and discriminator is optimal at every step, then the generator is guaranteed to converge to the data distribution
- Unrealistic assumptions!
- In practice, the generator and discriminator loss keeps oscillating during GAN training
- No robust stopping criteria in practice (unlike MLE)



Source: Mirantha Jayathilaka

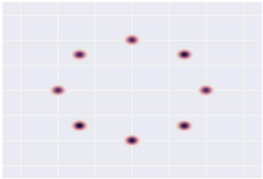
Mode Collapse

- GANs are notorious for suffering from mode collapse
- Intuitively, this refers to the phenomena where the generator of a GAN collapses to one or few samples (dubbed as “modes”)

Source: Arjovsky et at., 2017

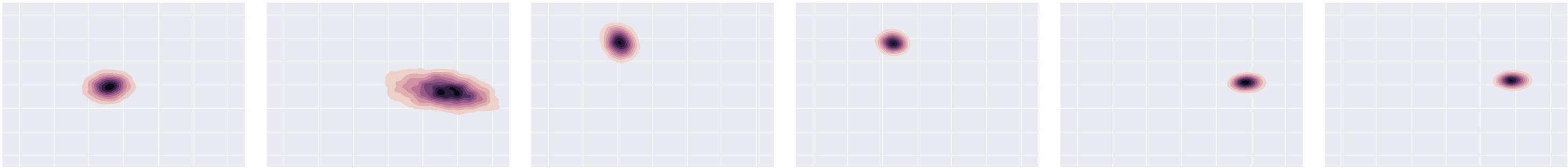
Mode Collapse

- True distribution is a mixture of Gaussians



Target

Source: Metz et at., 2017



Step 0 | Step 5k | Step 10k | Step 15k | Step 20k | Step 25k

- The generator distribution keeps oscillating between different modes

Mode Collapse

- Fixes to mode collapse are mostly empirically driven alternative architectures, alternative GAN loss, adding regularization terms, etc.
- How to Train a GAN? Tips and tricks to make GANs work by Soumith Chintala
  - https://github.com/soumith/ganhacks

Source: Metz et at., 2017

10k steps 20k steps 50K steps 100k steps

Recap

- Likelihood-free training
- Training objective for GANs

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

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

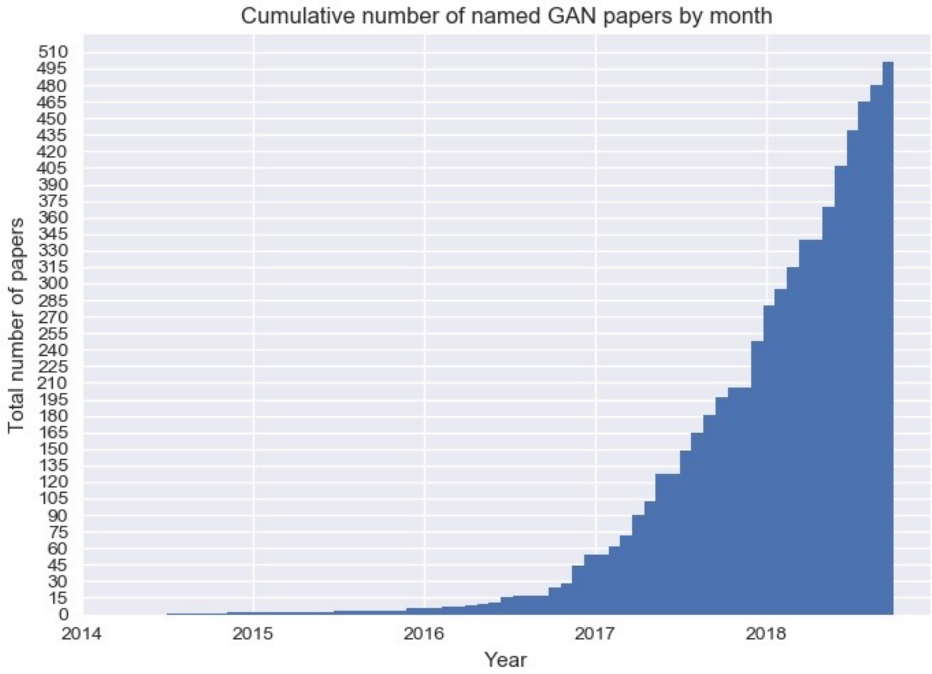
$$\min_G 2JSD(p_{data} \parallel p_G) - \log 4$$

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

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

GAN Zoo

- GAN Zoo: List of all named GANs
  - https://github.com/hinduрuravinash/the-gan-zoo



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

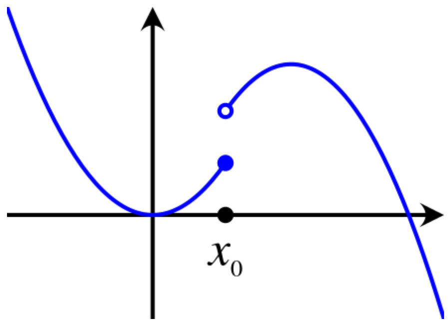
f-divergences

- 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	$\int\left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\,\mathrm{d}x$	$(\sqrt{u}-1)^2$
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 at., 2017

Thanks