GANs and Their Generalization

Shirin Goshtasbpour
Matteo Zecchin

Thursday 30th April, 2020

Generative Process

Goal: Given samples from true distribution $P(x)$ draw $x \sim P$
Latent variable model

- Latent variable $z \in \mathcal{Z}$
- Observables $x \in \mathcal{X}$

$$\underset{p_z}{z} \xrightarrow{p(x|z)} x$$

- Marginal Likelihood

$$p(x) = \int p_z(z)p(x|z)dz$$

- Latent variable prior $z \sim \mathcal{N}(0, \mathbf{I})$, $z \in \mathbb{R}^m$
- Generator $G : \mathbb{R}^m \to \mathbb{R}^d$ $\quad x \sim \underbrace{\mathcal{N}(G(z), \sigma^2\mathbf{I})}_{p_G(x|z)}$ $\quad$ for$\sigma \to 0$

Forget about the likelihood: Find G that implicit $p_G(x) \approx p(x)$

Adversarial Training

MLE is intractable, use a binary classifier to get training signal

- Discriminator $D(x)$: distinguish between data & fake samples

$$p(x, y) = \frac{1}{2} y p(x) + \frac{1}{2}(1 - y) p_G(x) \qquad y \in \{0, 1\}$$

- Saddle-point problem with logistic likelihood

$$\min_G \max_D \mathbb{E}_{x, y \sim p(x, y)} [\log D(x) + \log(1 - D(G(z)))]$$

Theoretical results for alternating optimization

- Optimal discriminator $D^*(G)$

$$\mathbb{P}(D^*(G)(x) = 1|x) = \frac{p(x)}{p(x) + p_G(x)}$$

- Optimal generator G^* minimizes $\text{JS}(p||p_G)$
- Optimal discriminator $\mathbb{P}(D^*(G^*)(x) = 1|x) = 1/2$

GAN in Practice

Parameterized models $G(.;\theta)$ and $D(.;\phi)$

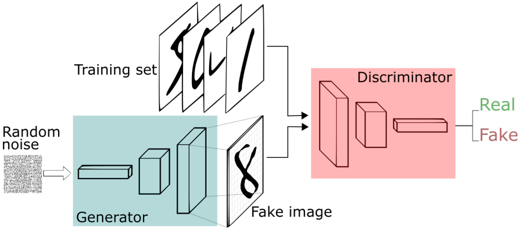


Figure: https://sthalles.github.io/intro-to-gans/

- Suboptimal solution
- SGD with backpropagation (curriculum and synchronous learning)
- No measure to assess convergence
- Generator update is adjusted to solve $\max_\theta \log D(G(z;\theta);\phi)$ in outer loop

Evaluation Methods

Marginal likelihood $p_G(x)$ is not defined

- Kernel Density Estimation (KDE) $$\widehat{p}(x) = \frac{1}{n\sigma}\sum_{i=1}^{n} K\left(\frac{x - x_i}{\sigma}\right)$$
- Annealed importance sampling
- Variational lower bound

Proxy performance measures

- Inception Score (IS)
- Frechet Inception Distance (FID)
- Precision and recall

f-divergence

- The objective is to obtain neural samples whose output distribution $p_G(x)$ resembles a target distribution $p(x)$.
- An f-divergence provides a notion of similarity between two distributions.

$$D_f(p_G|p) = \int_{\mathcal{X}} p(x) f\left(\frac{p_G(x)}{p(x)}\right) dx$$

- However working directly with f-divergences is not possible because it cannot be expressed as an empirical average of quantities that we can compute

Variational lower bound

- It is possible to obtain a lower bound on an f-divergence that can be computed using samples of p and p_G.

$$D_f(p_G|p) \geq \sup_{T \in \mathcal{T}} \mathop{\mathbb{E}}_{x \sim p}[T(x)] - \mathop{\mathbb{E}}_{x \sim p_G}[f^*(T(x))]$$

  where $\mathcal{T} : \mathcal{X} \to \mathbb{R}$ is an arbitrary class of functions and f^* is the Fenchel conjugate of f.
- The bound is tight if $f'\left(\frac{p_G(x)}{p(x)}\right) \in \mathcal{T}$

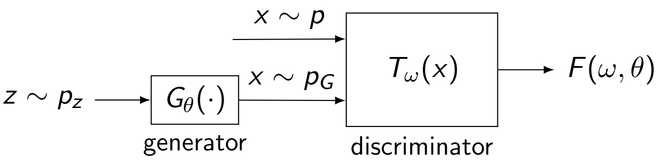
Variational divergence minimization

If we replace T with a neural network T_ω parametrized by ω.

$$D_f(p_G|p) \geq \max_{\omega \in \Omega} \mathop{\mathbb{E}}_{x \sim p}[T_\omega(x)] - \mathop{\mathbb{E}}_{x \sim p_G}[f^*(T_\omega(x))]$$

this needs to be:

- maximized w.r.t. ω to make the bound non vacuous.
- minimized w.r.t. θ to drive down the divergence.



Issue with divergence

Toy example: data lies in a 1D segment and we want to learn θ^*.

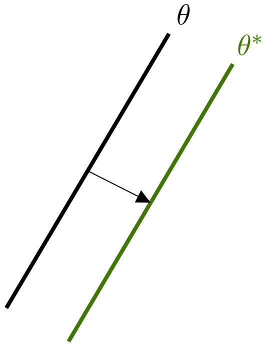
- Jensen-Shannon (GANs)

$$JS(p_G, P_{\theta^*}) = \begin{cases} 0 & \text{if } \theta = \theta^* \\ \log(2) & \text{if } \theta \neq \theta^* \end{cases}$$

- Kullback-Leibler

$$KL(p_G, P_{\theta^*}) = \begin{cases} 0 & \text{if } \theta = \theta^* \\ \infty & \text{if } \theta \neq \theta^* \end{cases}$$

Both not continuous in θ.

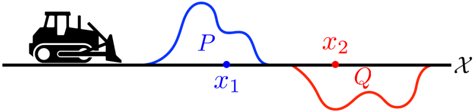


W-GANs

Minimize the Wasserstein metric

$$W(p, p_G) = \inf_{\gamma \in \Gamma(p, p_G)} \mathop{\mathbb{E}}_{(x,y)\sim\gamma} [\|x - y\|]$$

where $\Gamma(p, p_G)$ is the set of distributions with marginals p and p_G.



$\gamma(x_1, x_2)$: amount of mass moved from x_1 to x_2

Taking the infimum we are seeking for the most efficient way to transport P over Q.

Minimize the Wasserstein metric

The Wasserstein metric is a proxy to compare p and p_G, no need to find the optimal transport plan.

Dual formulation

$$W(p, p_G) = \sup_{||f||_L < 1} \left\{ \mathop{\mathbb{E}}_{x \sim p} [f(x)] - \mathop{\mathbb{E}}_{x \sim p_G} [f(x)] \right\}$$

Once again circumvent the problem of directly computing the metric (or divergence) using duality.

Optimization procedure

Replacing function f with a neural network T_ω of parameter ω

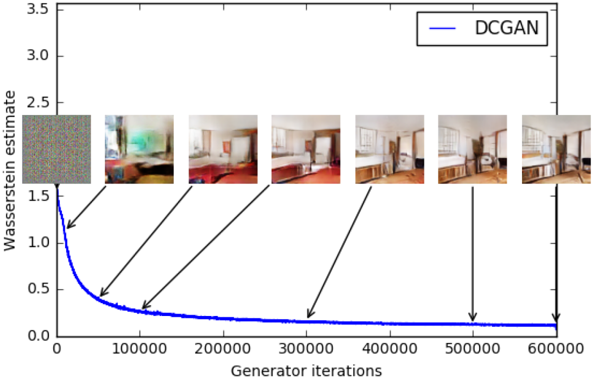
$$W_\omega(p, p_G) = \max_{\omega \in \Omega} \left\{ \mathop{\mathbb{E}}_{x \sim p} [T_\omega(x)] - \mathop{\mathbb{E}}_{x \sim p_G} [T_\omega(G_\theta(x))] \right\}$$

Alternating optimization:

- Optimize ω until convergence to obtain the value of the Wasserstein metric.
- Compute the gradient of $W_\omega(p, p_G)$ the w.r.t. θ to optimize the generator $G_\theta(\cdot)$.

Results

The Wasserstein metric strongly correlates with the quality of the generated samples.



Results

Training prodecure is more stable, the discriminator is trained till optimality providing better gradient estimates to the generator.

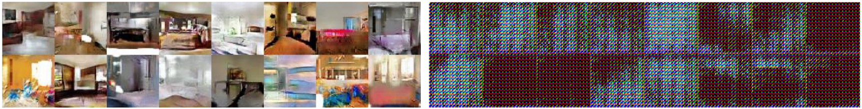


Figure: W-GAN vs GAN without batch normalization

References

- Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.
- Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka. "f-gan: Training generative neural samplers using variational divergence minimization." Advances in neural information processing systems. 2016.
- Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein gan." arXiv preprint arXiv:1701.07875 (2017).

Cool examples of GANs

Check out some stunning example applications of GANs

- https://thispersondoesnotexist.com/
- https://www.youtube.com/watch?v=JTploft1ZAI
- https://www.youtube.com/watch?v=cQ54GDm1eL0
- https://www.youtube.com/watch?v=p5U4NgVGAwg
- https://www.youtube.com/watch?v=gLoI9hAX9dw
- https://www.youtube.com/watch?v=9reHvktowLY
- https://www.youtube.com/watch?v=BHACKCNDMW8
- https://www.youtube.com/watch?v=tOUM2s3UN6Q (edited)