Spring 2022 Optimization & Machine Learning Talk I

Wasserstein GANs Work Because They Fail

Axel G. R. Turnquist

NJIT Department of Mathematical Sciences

January 20, 2022

The Famous Generative Adversarial Network (GAN)

- Two neural networks: generator $G_\theta : \mathcal{Z} \to \mathcal{X}$ and the discriminator $D_\alpha : \mathcal{X} \to \mathbb{R}$
- Two spaces: latent space $\mathcal{Z}$ and data space $\mathcal{X}$. Usually $\mathcal{Z}$ is a multivariate Gaussian, that is $z \in \mathcal{Z}$ means $z \sim p_z$ for a multivariate Gaussian distribution p_z
- Given a value function $V\left(G_\theta, D_\alpha\right)$, the GAN problem is:

$$\min_\theta \max_\alpha V\left(G_\theta, D_\alpha\right) \tag{1}$$

Value Functions and Induced Loss Functions

How do we choose $V\left(G_\theta, D_\alpha\right)$? Make an assumption!

$$\min_\theta \max_\alpha V\left(G_\theta, D_\alpha\right) = \min_\theta F\left(G_\theta\right) \tag{2}$$

for some F. Given this assumption, if one uses vanilla GAN's value function where p^* is the real distribution:

$$V\left(G_\theta, D_\alpha\right) = \mathbb{E}_{x\sim p^*}\left[\log D_\alpha(x)\right] + \mathbb{E}_{z\sim p_z}\left[\log\left(1 - D_\alpha\left(G_\theta(z)\right)\right)\right] \tag{3}$$

induces the value function

$$\frac{1}{2}\left(\mathrm{KL}\left(p^*\middle\|\frac{p^*+p^\theta}{2}\right) + \mathrm{KL}\left(p^\theta\middle\|\frac{p^*+p^\theta}{2}\right)\right) \tag{4}$$

(Continued)

However, for vanilla GAN, $\nabla_\theta V\left(G_\theta, D_\alpha\right) \to 0$ as D_α approaches the optimal discriminator D^*. Okay, well then, we try this instead:

$$V\left(G_\theta, D_\alpha\right) = \mathbb{E}_{x \sim p^*}\left[D_\alpha(x)\right] - \mathbb{E}_{z \sim p_z}\left[D_\alpha\left(G_\theta(z)\right)\right] \tag{5}$$

Then, by the dual formulation of Optimal Transport, we get the value function

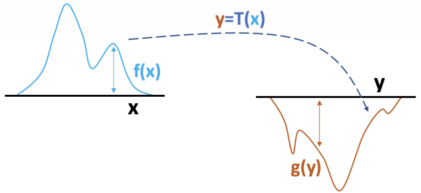
$$W_1\left(p^*, p^\theta\right) = \max_{\left\|D_\alpha\right\|_L \leq 1} V\left(G_\theta, D_\alpha\right) \tag{6}$$

Wasserstein Distance (for the Monge problem)

The Monge problem of Optimal Transport uses the **change of variables** formula from Calculus. That is, given two probability measures μ and ν and a diffeomorphic mapping T, such that $T_{\#}\mu = \nu$:

$$\int_A \mu(x) = \int_A \nu(T(x))J_T(x) \tag{7}$$

for all measurable $A \subset \Omega$ where J designates the Jacobian of the mapping T.



Wasserstein Distance (a whirlwind tour)

Now we compute the *horizontal distance* by finding the map T that takes the least amount of work to move from a point x to y, like "shoveling dirt":

$$\mathrm{dist}(\mu,\nu) = \inf_T \int_\Omega c(x,T)d\mu(x) \tag{8}$$

such that T satisfies the change of variables formula:

$$\int_A \mu(x) = \int_A \nu(T(x))J_T(x) \tag{9}$$

We get W_1 by choosing $c(x,y) = d(x,y)$.

Kantorovich-Rubinstein distance

$$W_1(\mu, \nu) := \inf_{T_{\#}\mu=\nu} \int_\Omega \|x - y\| \, d\mu(x) \tag{10}$$

1. Introduce Lagrange multipliers f and g, so this becomes a $\sup\inf$ problem.
2. Switch to an $\inf\sup$ problem (strong duality)

   $$\sup_{f(x)+g(y)\leq\|x-y\|} [\mathbb{E}_{x\sim p}[f(x)] + \mathbb{E}_{y\sim q}[g(y)]] \tag{11}$$

   where $d\mu = p(x)dx$ and $d\nu = q(y)dy$.
3. Now argue that optimizing over the class of 1-Lipschitz functions h is equal to this

   $$W_1(\mu, \nu) = \sup_{\|h\|_L\leq 1} [\mathbb{E}_{x\sim p}[h(x)] - \mathbb{E}_{y\sim q}[h(y)]] \tag{12}$$

Whistleblowers

- In Kodali et al., 2017 and Fedus et al., 2017, merely "controlling the Lipschitz constant of the discriminator may improve GAN training regardless of the statistical distance used", and "improved performance observed in WGAN-GP was simply due to the gradient penalty term and not connected to the Wasserstein distance".
- In Lucic at al., 2017 across many different GAN loss functions, there is a sensitivity to hyperparameters, and "no single loss function consistently outperforms the others. In particular, it was shown that given the right hyperparameter configuration, vanilla GAN can achieve a comparable or better performance than WGAN-GP."

Possible issues with WGAN you may or may not have noticed...

1. Assumption made above that discriminator is optimal at each step in the minimax computation
2. Computationally impossible to optimize over the set of *all* Lipschitz-1 functions
3. Do not have access to p^* and p^θ, but only finite samples (size n)

WGAN-Gradient Penalty (GP)

Introduced in (Gulrajani et al., 2017). Define:

$$\mathcal{V}\left(D_\alpha, p_n^*, p_n^\theta\right) := \mathbb{E}_{x\sim p_n^*}\left[D_\alpha(x)\right] - \mathbb{E}_{x\sim p_n^\theta}\left[D_\alpha(x)\right] \tag{13}$$

$$\mathcal{R}\left(D_\alpha, p_n^*, p_n^\theta\right) := \mathbb{E}_{x\sim\tau}\left[\left(\|\nabla_x D_\alpha(x)\| - 1\right)^2\right] \tag{14}$$

where τ is a uniform distribution sampled from the lines connecting x_i sampled from p^* and $\tilde{x}_i$ sampled from p^θ.

Then, perform N_D steps of gradient ascent with respect to $\mathcal{L}_D(\alpha) := \mathcal{V}\left(D_\alpha, p_n^*, p_n^\theta\right) - \lambda\mathcal{R}\left(D_\alpha, p_n^*, p_n^\theta\right)$ and N_G steps of gradient descent with respect to $\mathcal{L}_G(\theta) := \mathcal{V}\left(D_\alpha, p_n^*, p_n^\theta\right)$.

c-Transform WGAN

Introduced in (Mallasto et al., 2019b). Use c-transform value function:

$$V\left(G_\theta, D_\alpha\right) = \mathbb{E}_{x\sim p^*}\left[f(x)\right] + \mathbb{E}_{x\sim p^\theta}\left[f^c(x)\right] \tag{15}$$

where $f^c(x) := \sup_{y\{f(y)-\|x-y\|\}}$ and the c refers to the cost function $c(x,y) = \|x-y\|$. The algorithm is the same as for WGAN-GP, but with:

$$\mathcal{V}\left(D_\alpha, p_n^*, p_n^\theta\right) := \mathbb{E}_{x\sim p_n^*}\left[D_\alpha(x)\right] + \mathbb{E}_{x\sim p_n^\theta}\left[\hat{D}_\alpha^c(x)\right] \tag{16}$$

where $\hat{D}_\alpha^c := \min_{y\in\mathrm{supp}(p_n^\theta)} \|x-y\| - D_\alpha(y)$.

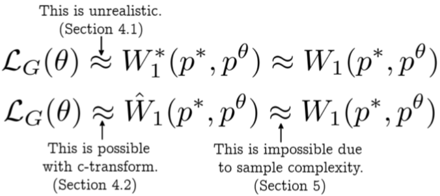
Convergence of WGAN-GP and c-transform WGAN

Oracle estimator of the Wasserstein distance:

$$W_1^*\left(p_n^*, p_n^\theta\right) = \mathbb{E}_{x\sim p_n^*}\left[f^*(x)\right] - \mathbb{E}_{x\sim p_n^\theta}\left[f^*(x)\right] \tag{17}$$

where $f^* \in \operatorname{argmax}_{\|f\|_L\leq 1}\left(\mathbb{E}_{x\sim p^*}\left[f(x)\right] - \mathbb{E}_{x\sim p^\theta[f(x)]}\right)$. Another empirical loss function, the batch estimator:

$$\hat{W}_1\left(p_n^*, p_n^\theta\right) = \max_{\|f\|_L\leq 1}\left(\mathbb{E}_{x\sim p_n^*}\left[f(x)\right] - \mathbb{E}_{x\sim p_n^\theta}\left[f(x)\right]\right) \tag{18}$$



This is unrealistic.
(Section 4.1)

$$\mathcal{L}_G(\theta) \approx W_1^*(p^*, p^\theta) \approx W_1(p^*, p^\theta)$$

$$\mathcal{L}_G(\theta) \approx \hat{W}_1(p^*, p^\theta) \approx W_1(p^*, p^\theta)$$

This is possible
with c-transform.
(Section 4.2)

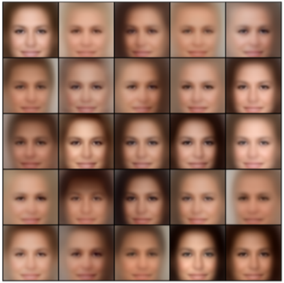
This is impossible due
to sample complexity.
(Section 5)

Proof is in the pudding...

Even though for the c-transform WGAN, we have:

$$\mathcal{L}_G(\theta) \approx \hat{W}_1\left(p^*, p^\theta\right) \tag{19}$$

and for WGAN-GP, the approximation is poor, a good approximation of the batch Wasserstein distance does not correspond to a good generative performance.



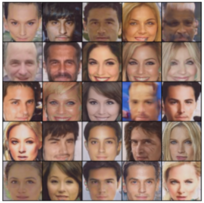
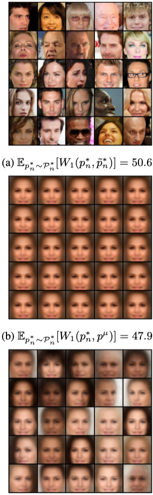


Figure 7: Samples resulting from the training with a given approximation method. c-transfrom on the top and gradient penalty on the bottom. Based on (Mallasto et al., 2019b).

I’ve fallen and I can’t get out!

Oracle estimator: $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ but it needs the true Lipschitz-1 function f^* which is prohibitively expensive. Batch estimator: $\mathcal{O}\left(n^{-1/d}\right)$.
FALSE MINIMA OF THE BATCH WASSERSTEIN DISTANCE. Experiment: expected batch Wasserstein distance:

- Two samples from the target distribution
- Target distribution and repeated means
- Target distribution and geometric k-means.



(a) $\mathbb{E}_{p_n^* \sim \mathcal{P}_n^*}[W_1(p_n^*, \tilde{p}_n^*)] = 50.67$

(b) $\mathbb{E}_{p_n^* \sim \mathcal{P}_n^*}[W_1(p_n^*, p^\mu)] = 47.91$

(c) $\mathbb{E}_{p_n^* \sim \mathcal{P}_n^*}[W_1(p_n^*, p^{\text{k-gm}})] = 39.44$

Questions?

Highlighted Resources

- Goodfellow, I. J. et al. Generative adversarial nets" Goodfellow, Ian J., 2014.
- Arjovsky, M. et al. Wasserstein Generative Adversarial Networks, 2017.
- Stanczuk, et al. Wasserstein GANs work because they rail (to approximate the Wasserstein distance, 2021.
- Kodali, N., Abernethy, J., Hays, J., and Kira, Z. On convergence and stability of GANs, 2017.
- Kodali, N. et al. On convergence and stability of GANs, 2017.

(Continued)

- Fedus, W., et al. Many paths to equilibrium; GANs do not need to decrease a divergence at every step, 2017.
- Lucic, M. et al. Are GANs created equal? A large-scale study, 2017. (Hyperparameter sensitivity)
- Gulrajani, I. et al, Improved training of Wasserstein GANs, 2017.
- Mallasto, A. et al. How well do WGANs estimate the Wasserstein metric?, 2019.

Future Talks

Next Talk:

Binan Gu