Wasserstein proximal of GANs

Alex Tong Lin, Wuchen Li, Stanley Osher, Guido Montúfar
{atlin,wcli,sjo,montufar}@math.ucla.edu

UCLA

Introduction

- Develop deep learning techniques for mean field games and optimal transport.
- Inspired by the Hopf-Lax formula in density space, we introduce the Wasserstein proximal operator for generative adversarial networks.
- This poster presents the relaxed Wasserstein proximal operator (RWP), which is an easy to implement modification that gives faster and more stable training of generative models.

Natural Gradient

- The natural gradient is a gradient descent scheme which chooses the descent direction to be invariant to different coordinate changes.
- In deep learning, given a step-size α, a model parameterized by θ, and a loss function L, then the steepest descent is:

$$\delta^* = \operatorname{argmin}_{\{\delta : D(p_\theta \| p_{\theta+\delta}) = \alpha\}} L(\theta + \delta)$$

where D is often chosen as the Kullback–Leibler divergence

$$D(p\|q) = \int p(x) \log \frac{p(x)}{q(x)}\, dx$$

Wasserstein Proximal

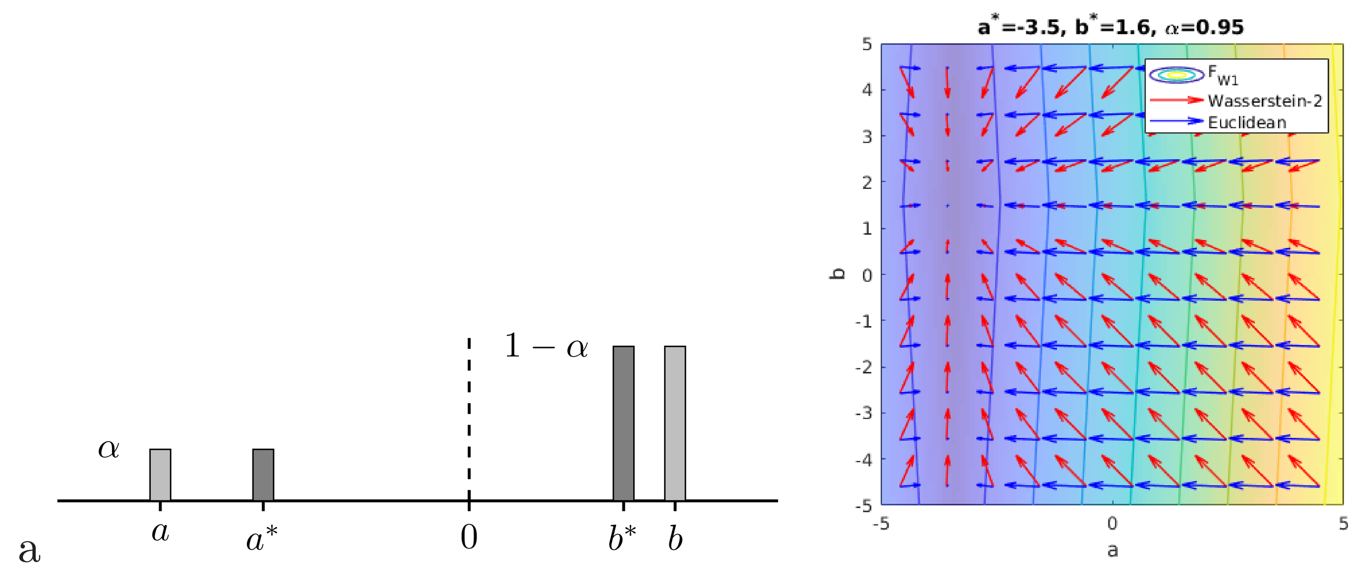
The Wasserstein proximal utilizes the natural gradient but with the Wasserstein-2 distance:

$$\delta^* = \operatorname{argmin}_{\{\delta : d_W(p_\theta \| p_{\theta+\delta})^2 = \alpha\}} L(\theta + \delta)$$

Theorem 1 (Constrained Wasserstein-2 metric)**.** *The constrained Wasserstein-2 metric function $d_W : \Theta \times \Theta \to \mathbb{R}_+$ has the following formulation:*

$$d_W(\theta_0, \theta_1)^2 = \inf \left\{ \int_0^1 \int_{\mathbb{R}^n} \|\nabla \Phi(t,x)\|^2 \rho(\theta(t), x)\, dx dt : \partial_t \rho(\theta(t), x) + \nabla \cdot (\rho(\theta(t), x) \nabla \Phi(t,x)) = 0,\ \theta(0) = \theta_0,\ \theta(1) = \theta_1 \right\}$$

where the infimum is among all feasible Borel potential functions $\Phi : [0,1] \times \mathbb{R}^n \to \mathbb{R}$ and continuous parameter paths $\theta : [0,1] \to \mathbb{R}^d$.



The Wasserstein proximal penalizes parameter steps in proportion to the mass being transported, which results in updates pointing towards the minimum of the loss function. The Euclidean proximal penalizes all parameters equally, which results in updates naively orthogonal to the level sets of the loss function.

Generative Adversarial Networks

- Generative Adversarial Networks (GANs) mimic the probability distribution from the training data. When properly trained, they can generate samples that look like the training data.
- GANs consist of two neural networks: The discriminator D_ω, and the generator G_θ. The job of of the generator is to generate "fake" samples, and the job of the discriminator is to decide whether an image is real (i.e. from the training data), or fake (i.e. from the generator).
- An example of artificial faces generated by GANs:

- (Standard) GAN loss function:

$$\max_D \min_G \ \mathbb{E}_{x\sim \text{real}}[\log D(x)] + \mathbb{E}_{z\sim N}[\log 1 - D(G(z))]$$

- In order to obtain a quantitative evaluation of generated samples, we used the Fréchet Inception Distance (FID) [2] to measure the quality of samples.

RWP Algorithm

- The RWP algorithm is meant as a drop-in regularizer. You train your GAN in the usual way with a slight modification:
- The algorithm is as follows:
  - Sample real data $\{x_i\}_{i=1}^B$ and noise data $\{z_i\}_{i=1}^B$
  - Do

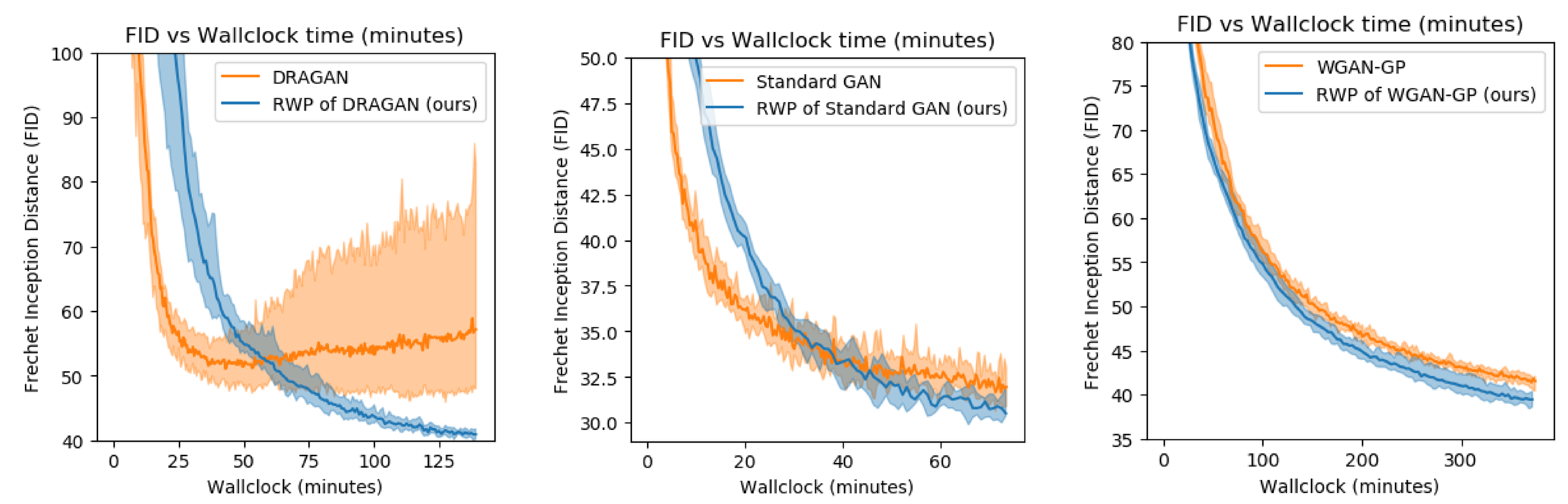
$$\omega^{k+1} \leftarrow \text{Optimizer}_\omega \left\{ \text{Loss}(D_{\omega^k}(\{x_i\}_{i=1}^B),\ D_\omega(G_\theta(\{z_i\}_{i=1}^B))) \right\}$$

  - Sample noise data $\{z_i\}_{i=1}^B$
  - Do

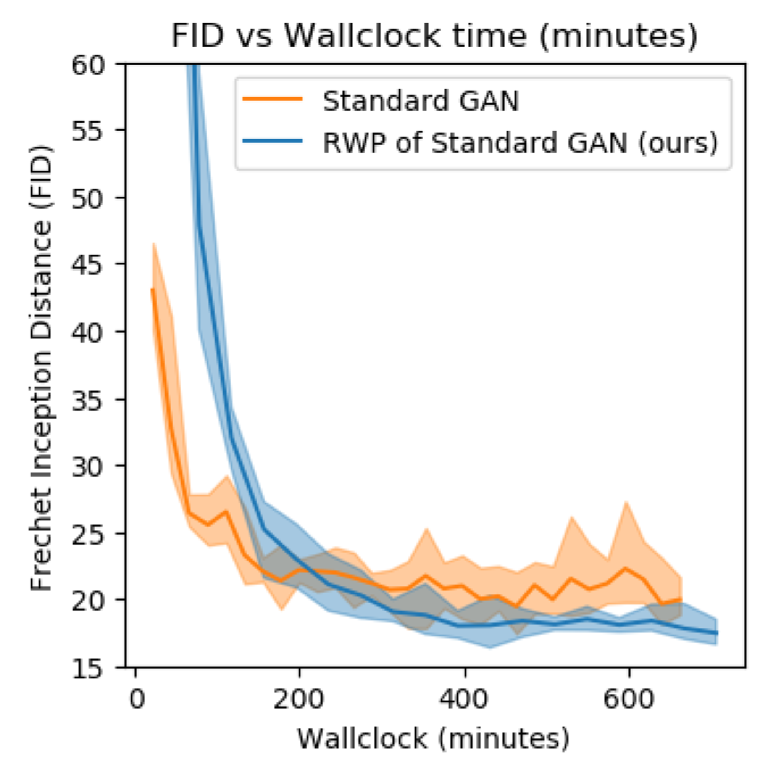
$$\theta^{k+1} \leftarrow \text{Optimizer}_\theta \left\{ Loss(G_\theta(\{z_i\}_{i=1}^N)) + \frac{1}{B} \sum_{i=1}^B \|G_\theta(z_i) - G_{\theta^k}(z_i)\|^2 \right\}$$

  - Repeat until convergence.
- Experimentally, RWP gives better and faster convergence (wallclock time).

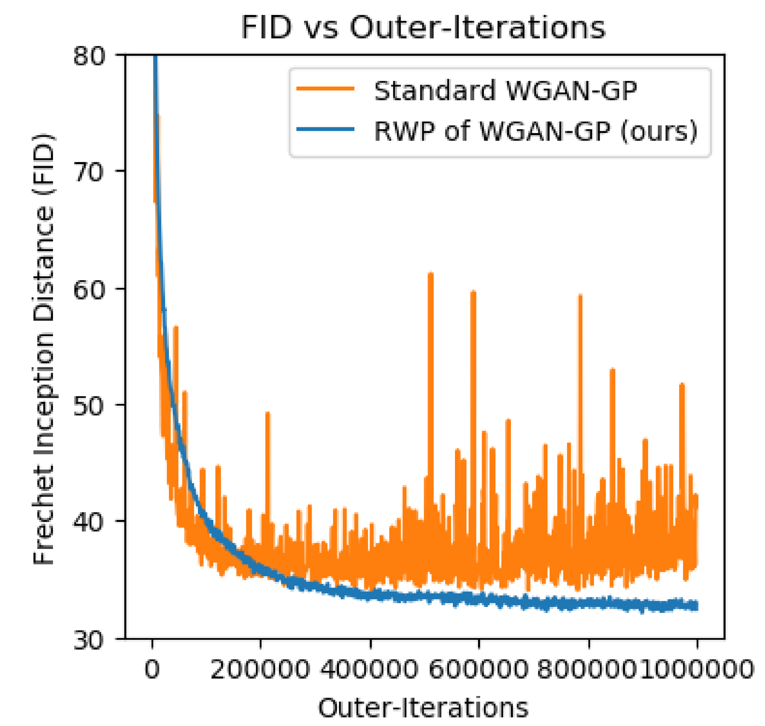
Experiments on GANs



The effect of using RWP regularization, on the CIFAR-10 dataset. The experiments are averaged over 5 runs. The bold lines are the average, and the enveloping lines are the minimum and maximum.



The effect of Relaxed Wasserstein Proximal (RWP) regularization on Standard GANs, on the CelebA dataset. The experiment was averaged over 5 runs. The bold lines are the average, and the enveloping lines are the minium and maximum. Here we see RWP regularization improves the speed (via wallclock time), and achieves a lower FID.



An experiment demonstrating the effect of performing 10 generator iterations per outer-iteration with and without RWP, where an outer-iteration is a single loop of: a number of discriminator iterations, then a number of generator iterations.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[2] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6626–6637. Curran Associates, Inc., 2017.

[3] W. Li and G. Montúfar. Natural gradient via optimal transport. *arXiv:1803.07033 [cs, math]*, 2018.

[4] W. Li and G. Montúfar. Ricci curvature for parametric statistics via optimal transport. *arXiv:1807.07095 [cs, math, stat]*, 2018.

[5] W. Li and S. Osher. Constrained dynamical optimal transport and its Lagrangian formulation. *arXiv:1807.00937 [math]*, 2018.