1. Introduction and Background

Generative adversarial networks (GANs) have been a topic of intense research interest since their 2014 introduction by Goodfellow et. al. [6]. Generative networks differ from more traditional approaches to deep learning in that, as the name suggests, they are designed to generate synthetic data which bears some resemblance to real data. More specifically, generative adversarial networks achieve this by implementing an architecture with two neural networks, a 'generator' which generates synthetic data and a 'critic' which measures how similar this synthetic data is to the real data. These networks are trained iteratively, with the generator trained to minimize the critic's ability to distinguish real and synthetic data, and the critic trained to maximize its own ability to do so [6].

A major development in the history of GANs came with Arjovsky et. al.'s introduction of a critic which uses the Wasserstein d_1-distance to differentiate real and synthetic data [4]. Wasserstein distances are defined as follows: given $p \geq 1$, a set $\Omega \subseteq \mathbb{R}^d$, and probability measures $\mu, \nu \in \mathcal{P}(\Omega)$, we say that γ is a transportation plan between μ and ν if $\gamma(A \times \Omega) = \mu(A)$ and $\gamma(\Omega \times A) = \nu(A)$ for any measurable set $A \subseteq \Omega$. Denoting the space of such transportation plans by $\Gamma(\mu, \nu)$, and making the further assumption that

$$\mu, \nu \in \mathcal{P}_p(\Omega) := \left\{ \rho \in \mathcal{P}(\Omega) \;\middle|\; \int_\Omega |x|^p d\rho(x) < \infty \right\}$$

we define the d_p-Wasserstein distance between μ and ν by

$$(1) \qquad d_p(\mu, \nu) := \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\Omega \times \Omega} |x - y|^p d\gamma(x, y).$$

Wasserstein distances come endowed with a suite of attractive geometric and mathematical properties – notably, the fact that d_p induces a true metric on $\mathcal{P}_p(\Omega)$, and Kantorovich duality, which allows us to compute $d_p(\mu, \nu)$ by maximizing a dual functional on an appropriate space of functions on Ω [14]. In the special case $p = 1$, the Kantorovich-Rubenstein theorem states that we may find $d_1(\mu, \nu)$ by maximizing a dual linear functional on the space of 1-Lipschitz functions from Ω to $\mathbb{R}$. This fact forms the basis for Wasserstein generative adversarial networks (WGANs) as defined by Arjovsky et. al. in [4]. A key advancement in the theory of WGANs occurred when Gulrajani et. al. substituted the 1-Lipschitz condition on the space of candidate functions with a gradient penalty term in the objective functional, leading to the WGAN-GP architecture of [7].

It was in this context that Milne, along with co-authors Bilocq and Nachman, developed the Trust the Critics (TTC) algorithm. In a nutshell, this algorithm begins by training a critic neural network to maximize a WGAN-GP problem, yielding what is known as an approximate Kantorovich potential. Then, the geometric information encoded in this Kantorovich potential is used to update the synthetic data μ so that it is harder to distinguish from the real data ν, circumventing the need for a generator neural network [11].

In brief, the TTC algorithm generates a sequence of synthetic data $\mu = \mu_0, \mu_1, \mu_2, ...$ which increasingly resemble the real data ν according to the following procedure described in [9, 11]. First, train a neural network u_θ to be an approximate Kantorovich potential between μ_n and ν by solving the following WGAN-GP problem:

$$\sup_\theta \left[\int_\Omega u_\theta \; d\mu_n - \int_\Omega u_\theta \; d\nu - \lambda \int_\Omega (|\nabla u_\theta| - 1)_+^2 d\sigma_n \right],$$

where $a_+ := \max(a, 0)$ and $\sigma \in \mathcal{P}(\Omega)$ is chosen depending on μ_n and ν. Then, denote the nearly-optimal u you get after training as u_n. Next, define a step size

$$\tau_n := \alpha \left[\int_\Omega u_n d\mu_n - \int_\Omega u_n d\nu \right] \approx \alpha d_1(\mu_n, \nu),$$

where $\alpha \in [0, 1]$ is a hyperparameter. Finally, update the data by defining

$$\mu_{n+1} := (\mathrm{Id} - \tau_n \nabla u_n)_\# \mu_n,$$

where the notation $T_\# \rho$ denotes the unique 'pushforward' measure on Ω such that $T_\# \rho(A) = \rho(T^{-1}(A))$ for any measurable set $A \subseteq \Omega$.

The main focus of this project, however, is to study the useful Wasserstein Rudin-Osher-Fatemi (WROF) formulation of the TTC algorithm. This formulation is inspired by the work on image denoising in [13], which was introduced by Milne in [9, Chapter 5] and studied further in [10]. In particular, [9, Proposition 5.10] states that, if $\Omega \subseteq \mathbb{R}^d$ is taken to be compact and convex, $\mu_n \ll \mathcal{L}^d$ (where $\mathcal{L}^d$ is Lebesgue measure on $\mathbb{R}^d$), and τ_n is smaller than the minimal transportation length defined in [9, Definition 4.2], then one step of the TTC algorithm is equivalent to solving the WROF problem

$$(2) \qquad \mu_{n+1} \in \operatorname*{argmin}_{\rho \in \mathcal{P}(\Omega)} \left[\frac{1}{2\tau_n} d_2^2(\mu_n, \rho) + d_1(\rho, \nu) \right].$$

This problem, it should be noted, is strikingly similar to the Jordan-Kinderlehrer-Otto scheme developed in [8] and later explained in [14, Section 8.2] and [3, Chapter 11], although the classical theory does not apply because $F(\rho) := d_1(\rho, \nu)$ fails to satisfy certain strict convexity properties which are critical for applying that theory.

2. Problem Statement and Methods

My goal for this project was to implement the WROF algorithm (2) in a simple context (albeit one to which Milne's theory does not directly apply), perform some exploratory analysis, and interpret the results to generate theoretical conclusions. As such, I used the following framework, based on Peyré's numerical implementation of the optimal transport problem in [12]. First, I modelled my space Ω as K equally-spaced points on the interval $[0, L]$, i.e. $\Omega = \{0, \frac{L}{K-1}, \frac{2L}{K-1}, \dots, \frac{(K-2)L}{K-1}, L\}$, and wrote probability measures $\rho \in \mathcal{P}(\Omega)$ in the form $\rho = \sum_{k=0}^{K-1} \rho^k \delta_{kL/K}$, with the requirement that each $\rho^k \geq 0$ and $\sum \rho^k = 1$.

Decomposing μ_n and ν in this manner, and using definition (1) of Wasserstein distance, we can write (2) as a linear program with $2K^2$ variables, $2K^2$ inequality constraints, and $3K$ equality constraints. More precisely, this is the problem of minimizing the linear function $F : \mathbb{R}^{K^2} \times \mathbb{R}^{K^2} \to \mathbb{R}$ given by

$$F(\gamma_n, \tilde{\gamma}_n) = \frac{L^2}{2K^2\tau_n} \sum_{i,j=0}^{K-1} \gamma_n^{ij}(i-j)^2 + \frac{L}{K} \sum_{i,j=0}^{K-1} \tilde{\gamma}_n^{ij}|i-j|,$$

subject to inequality constraints $\gamma_n^{ij}, \tilde{\gamma}_n^{ij} \geq 0$ for all $i, j \in \{0, ..., K-1\}$, and equality constraints $\sum_i \gamma_n^{ik} = \mu_n^k$, $\sum_j \gamma_n^{kj} = \sum_i \tilde{\gamma}_n^{ik}$, and $\sum_j \tilde{\gamma}_j^{kj} = \nu^k$ for all $k \in \{0, ..., K-1\}$. After framing (2) in this form, I numerically solved it with `cvxpy`.

3. Results and Discussion

I ran the previously decribed procedure with $(K, L) = (128, 1)$. In general, I found that, if the step size τ_n was too small, then $\mu_n = \mu$ for all $n \geq 0$ – in other words, the numerics fail to update the initial measure. However, if the step size τ_n is too large, then the numerics converge too quickly the target measure ν, avoiding the benefits of iterative procedures. I found that the most interesting results arose when the step size was on the same order of magnitude as the grid size:

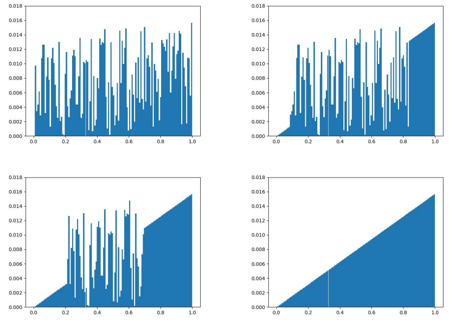

Figure 1. Implementation of the algorithm in Section 2 for a randomly generated measure μ and a target probability measure ν satisfying $\nu(\{\frac{kL}{K-1}\}) = ck$ for some constant c and all $k \in \{0, \dots, K-1\}$. The step size is $\tau = 0.005$, and the results after 0, 10, 20 and 30 steps of the WROF algorithm are shown from top left to bottom right. After 30 steps, the WROF algorithm recovers the target measure. A full animation can be found at `https://github.com/Camdav/MAT1510`.

Interestingly, in the GIF linked in the figure caption (and, to some extent, in the figure itself), there are two distinct effects which appear as we run the WROF procedure, which we will call 'translation' and 'reshaping.' Translation seems to occur when there are large-scale differences between the two measures. In our example, the source measure is roughly homogeneous when viewed at a large scale, whereas the target measure clearly distributes more mass to grid points which are further to the right, which means that a large amount of mass must be translated to the right in order to correct large-scale mass imbalances between the source measure and the target measure. Reshaping, on the other hand, occurs near places where there are no large-scale mass imbalances, and transforms the local features of the source measure into those of the target measure.

Perhaps surprisingly, the WROF algorithm appears to translate parts of a measure while reshaping other parts, but it appears to do little to no reshaping on parts of the measure which are being actively translated. In the linked GIF, it appears as if parts of the random measure μ is being translated rightward. Each iterate looks like the target measure ν near the endpoints, and like a translated version of the source μ in the center. In other words, the randomness of μ only seems to get smoothed out for mass which is already very close to its final destination. This may have implications for the numerics – i.e. we may want to halt the procedure when most mass has been translated, but before everything is reshaped, to encourage the network to generate novel samples, rather than copying the true data.

4. Future Work

It would be interesting to look into the multiscale approach in [10, Section 1.3], as opposed to the iterative regularization algorithm we implemented. We could also implement a better algorithm to solve (2), as in [1, 5], especially to mitigate the fact that such optimal transport algorithms scale poorly with dimension. Finally, it might be worthwhile to look at a particle simulation on a larger, discrete grid, as this may scale better than the implementation I used.

References

1. Jason M. Altschuler and Enric Boix-Adserà, *Wasserstein barycenters can be computed in polynomial time in fixed dimension*, J. Mach. Learn. Res. **22** (2021), Paper No. 44, 19. MR 4253737
2. ———, *Wasserstein barycenters are NP-hard to compute*, SIAM J. Math. Data Sci. **4** (2022), no. 1, 179–203. MR 4378594
3. Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré, *Gradient flows in metric spaces and in the space of probability measures*, second ed., Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, 2008. MR 2401600
4. Martin Arjovsky, Soumith Chintala, and Léon Bottou, *Wasserstein generative adversarial networks*, Proceedings of the 34th International Conference on Machine Learning (Doina Precup and Yee Whye Teh, eds.), Proceedings of Machine Learning Research, vol. 70, PMLR, 06–11 Aug 2017, pp. 214–223.
5. Jiaojiao Fan, Qinsheng Zhang, Amirhossein Taghvaei, and Yongxin Chen, *Variational Wasserstein gradient flow*, Proceedings of the 39th International Conference on Machine Learning (Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, eds.), Proceedings of Machine Learning Research, vol. 162, PMLR, 17–23 Jul 2022, pp. 6185–6215.
6. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, *Generative adversarial nets*, Advances in neural information processing systems **27** (2014).
7. Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville, *Improved training of wasserstein gans*, Neural Information Processing Systems, 2017.
8. Richard Jordan, David Kinderlehrer, and Felix Otto, *The variational formulation of the Fokker-Planck equation*, SIAM J. Math. Anal. **29** (1998), no. 1, 1–17. MR 1617171
9. Tristan Milne, *Optimal transport, congested transport, and wasserstein generative adversarial networks*, Phd thesis, University of Toronto, Toronto, Ontario, November 2022, Available at `https://hdl.handle.net/1807/125470`.
10. Tristan Milne and Adrian Nachman, *An optimal transport analogue of the rudin osher fatemi model and its corresponding multiscale theory*, 2023.
11. Tristan Milne, Étienne Bilocq, and Adrian Nachman, *Trust the critics: Generatorless and multipurpose wgans with initial convergence guarantees*, 2021.
12. Gabriel Peyré, *Optimal transport with linear programming*, `https://nbviewer.org/github/gpeyre/numerical-tours/blob/master/python/optimaltransp_1_linprog.ipynb`, Accessed: 2023-12-05.
13. Leonid I. Rudin, Stanley Osher, and Emad Fatemi, *Nonlinear total variation based noise removal algorithms*, vol. 60, 1992, Experimental mathematics: computational issues in nonlinear science (Los Alamos, NM, 1991), pp. 259–268. MR 3363401
14. Filippo Santambrogio, *Optimal transport for applied mathematicians*, Progress in Nonlinear Differential Equations and their Applications, vol. 87, Birkhäuser/Springer, Cham, 2015, Calculus of variations, PDEs, and modeling. MR 3409718