Wasserstein minimization in Generative Networks

Abstract

A generative model is presented which defines an algorithm by which the minimization of the Wasserstein distance in the semi-discrete setting is guaranteed. This approach is based on two alternating steps: (a) the source of randomness of the network is a continuous distribution, hence a deterministic optimal transport map exists and is used for the Wasserstein distance; (b) given an optimal transport map between a generator network and a target distribution, thethe Wasserstein distance of the generated data and reference points is decreased through a regression step. This method is proven to minimize the Wasserstein distance with respect to both the empirical target distribution but also the underlying one. Experiments are implemented on the Kuzushiji-MNIST dataset and the Wasserstein's distance unsuitability for certain tasks is discussed.

1 Introduction

The Monge Problem: Define a cost function $c(x, y) \geq 0,\ (x, y) \in (\mathcal{X}, \mathcal{Y})$, and let $X \sim \mu,\ Y \sim \nu$ be two random variables taking values in $\mathcal{X}, \mathcal{Y}$ respectively. The Monge problem (Monge, 1781) consists of finding a map $f :\ \mathcal{X} \to \mathcal{Y}$ which transports the mass from μ to ν, while minimizing the mass transportation cost

$$\inf_f \mathbb{E}_{X\sim\mu}[c(X, f(X))],\ \text{subject to } f(X) \sim \mathcal{Y}. \tag{1}$$

While [16] originally considered the ℓ_2 norm as the cost function, we will refer to any cost function $c(x, y)$.

Kantorovich Relaxation: To make 1 always feasible, [11] relaxed the Monge problem into an minimization over couplings $(X, Y) \sim \pi$ rather than set of maps, where π has marginals equal to μ and ν,

$$\inf_\pi \mathbb{E}_{(X,Y)\sim\pi}[c(X, Y)],\ \text{subject to } X \sim \mu,\ Y \sim \nu. \tag{2}$$

This relaxation allows mass at a given point $x \in Supp(\mu)$ to be transported to several $y \in Supp(\nu)$, while the Monge problem would send the whole mass at x to a unique location $f(x)$. This formulation is a linear program and has been solved with super-cubic in the size of the supports complexity of algorithms, preventing use of OT in large-scale settings. We will make use of the above relaxation of the Mogne problem to create our basis four our setting.

A generative network g models a distribution by sampling $x \sim \mu$ from a sample distribution, and afterwards outputting $g(x)$, defining in this way a pushforward distribution $g\#\mu$. The goal of a common training procedure is to fit $g\#\mu$ to a target distribution $\hat{\nu}$, by minimizing the divergence $\mathcal{D}(g\#\mu, \hat{\nu})$ over a set of parameters defining g.

2 Semi-discrete approach to generative network training via Wasserstein minimization

[5] propose a procedure to fit generative networks to target distributions, which explicitly minimizes the Wasserstein-p distance of the the pushforward distribution $g\#\mu$ and target distribution $\hat{\nu}$. This approach alternates between two steps: given generator g_i algorithm OTS associates g_i's probability mass with that of $\hat{\nu}$ and afterwards algorithm FIT uses this association (or labeling) to find a new generator g_{i+1}, through a standard regression.

The success of the above idea relies on two properties: the generators giving rise to continuous distributions, making the process *semi-discrete*, and the fact that the generator is slowly shifted

towards the target distribution $\hat{\nu}$, making the procedure *gradual*. A key consequence of the semi-discreteness is that the underlying optimal transport can be realized with a deterministic mapping. Already established solvers have succeeded in constructing a transport between $g\#\mu$ and $\hat{\nu}$, while batch-to-batch methods of transports using samples from $g\#\mu$ are biased and do not minimize the Wasserstein distance (e.g. [3]). As for the graduality of this method, its aim is to slowly steer the generated distribution towards the targeted one.

An important theoretical guarantee is the explicit minimization of the Wasserstein distance with respect not only the target distribution $\hat{\nu}$, but also the underlying distribution ν, from which $\hat{\nu}$ was sampled. This is proved using the triangle inequality for Wasserstein distances, introducing the Wasserstein distance $W(\nu, \hat{\nu})$, between ν and $\hat{\nu}$, which is exponential in dimension [2], [20]. It is shown that when a parametric model captures the distributions well, polynomial bounds in dimension are possible.

Numerical results are presented on the Kuzushiji-MNIST dataset.

2.1 Algorithms

This work presents two alternating algorithms, which are called, as mentioned earlier, OTS and FIT, and the overall algorithm, which combines them.

The *Wasserstein-p* distance W_p of two probability measures μ', ν' and their *optimal transport cost* $\mathcal{T}_c$ in a metric space (X, d) are defined by

$$W_p(\mu', \nu') := \Big(\inf_{\gamma \in \Gamma(\mu', \nu')} \int_X d(x,y)^p d\gamma(x,y) \Big)^{1/p}, \ \mathcal{T}_c(\mu', \nu') := \int_X c(x,y) d\gamma(x,y).$$

These are related if we choose the cost function $c(x, y) := d(x, y)^p$. Here denote that $\Gamma(\mu', \nu')$ is a collection of probability measures on $X \times X$, with marginal distributions μ' and ν'.

The proposed method heavily relies on the *Kantorovich relaxation* 2 and the *Kantorovich duality*, which for our setting becomes as in [21],

$$\mathcal{T}_c(\mu', \nu') = \sup_{\phi, \psi \in \Phi_c} \int_X \phi(x) d\mu'(x) + \int_X \psi(y) d\nu'(y), \tag{3}$$

where Φ_c is a collection of $(\phi, \psi) \in L_1(d\mu') \times L_1(d\nu')$ (which means ϕ, ψ are absolutely Lebesgue integrable functions with respect to μ', ν', and $\phi(x) + \psi(y) \leq c(x, y)$, for almost all x, y).

2.2 Optimal Transport Solver (OTS)

In our generative modeling case, let $\{y_1, ..., y_N\}$ be our training set. The optimal transport cost between our pushforward measure $g\#\mu$ and the empirical measure $\hat{\nu} \sim \mathcal{U}(\{y_1, ..., y_N\})$ becomes using 3 and the analytical solution of maximizing ϕ (solved analytically due to $g\#\mu$ being continuous) $\phi(x) = \min_i(c(x, y_i) - \hat{\psi}_i)$, [9], [17]

$$\mathcal{T}_c(g\#\mu, \hat{\nu}) = \sup_{\hat{\psi} \in \mathbb{R}^N} \int_X \min_i (c(x, y_i) - \hat{\psi}_i) dg\#\mu(x) + \frac{1}{N} \sum_{i=1}^{N} \hat{\psi}_i. \tag{4}$$

The problem has now transformed into an optimization problem over a vector in $\mathbb{R}^N$. Here note that $\hat{\psi}_i := \psi(y_i)$, $\psi^c(x) := \min_i(c(x, y_i) - \hat{\psi}_i)$ and $\Phi'_{c,\hat{\psi}}$ is a collection of $\phi \in L_1(dg\#\mu)$, such that $\phi(x) + \hat{\psi}_i \leq c(x, y_i)$ for almost all x and $i = 1, ..., N$.

The algorithm OTS uses SGD to maximize 4, or minimize its negation. Given $g\#\mu$, a training dataset $y_1, ..., y_n$ corresponding to $\hat{\nu}$ sampled from ν, cost function c, batch size B and learning rate η, OTS outputs an optimal vector $\hat{\psi} = (\hat{\psi_1}, ..., \hat{\psi_N})$. Computing the minimum over the whole dataset has $\mathcal{O}(N)$ complexity, which is costly for large datasets. The trick being employed to allow us to use OTS is to calculate the gradient of $\hat{\psi}$ on a subsample (e.g. about 1%) and gradually increasing this data size to cover the whole test.

The optimal transport cost computed through 3 is the cost for the Kantorovich optimal transference plan, where mass can be split, which is a relaxation of the Monge optimal transference plan, where mass cannot be split.

To continue, a deterministic Monge OT map is important, so as to provide the regression target for FIT. In this semi-discrete setting, the Kantorovich OT plan is unique and does not split mass, becoming a Monge OT plan, under convexity and superlinearity of the cost function.

Proposition 2.1. *Let $X = \mathbb{R}^d$, $g\#\mu$ be continuous, $\hat{\nu}$ be discrete and the cost function $c(x, y) = c(x - y)$ be strictly convex and superlinear on $\mathbb{R}^d$. Then, given the optimal $\hat{\psi}$ from 4, we can derive a unique Kantorovich optimal transference plan $T(x) := y_{arg\min_i (c(x,y_i)-\hat{\psi}_i)}$ from $g\#\mu$ to ν. $T(x)$ is also a Monge optimal transference plan.*

It is important to notice that Wasserstein-p distances on ℓ_p, $p > 1$ satisfy strict convexity and superliniarity [8], while for $p = 1$ that is not the case. In practise, when $p = 1$ OTS still converges to near-optimal transference plans, creating crisper images than in other metrics (e.g. on ℓ_2, Figure 2). Furthermore, the continuity of $g\#\mu$, which is required for the existence of Monge transference plan, holds if g is a feedforward neural network with non-degenerate weights, and invertible activation function such as sigmoid, tanh, or leaky ReLU. For non-invertible activation functions such as ReLU, it is possible that $g\#\mu$ gives mass to a discrete subset of $\mathbb{R}^d$. This did not happen in our experiments (Figure 1), and can be circumvented by adding a small perturbation to g's output.

2.3 Fitting Optimal Transport Plan (FIT)

Given an initial generator g, an optimal transference plan T between $g\#\mu$ and $\hat{\nu}$ by OTS, we find a new generator g' by sampling $z \sim \mu$ and regressing the new sample $g'(z)$ to the old OT plan $T(g(z))$. Under specific assumptions seen in 3.1, this is shown to result in strictly reducing the optimal transport cost, for an exact optimal plan T, which can be approximated via OTS.

2.4 Overall Algorithm

Combining OTS and FIT in an alternating procedure is how the overall algorithm is constructed. At iteration i, OTS outputs an optimal transport plan T between the old generated distribution $g_i\#\mu$ and $\hat{\nu}$, which FIT then uses to regress g_{i+1} towards $T\#g_i\#\mu$ to obtain a lower Wasserstein distance. The importance of the alternating procedure is shown in Figure 3.

3 Theoretical Results

3.1 Optimization Guarantee: $\mathcal{T}_c(g_i\#\mu, \hat{\nu}) \to 0$.

Supposing that the costs $\mathcal{T}_c$ have β-th powers that satisfy the triangle inequality [21], we will use a scalar $\alpha \in (0, 0.5)$, whose role is to determine the precisions of OTS and FIT.

For $C_\mu(f, g) := \int c(f(x), g(x))d\mu(x)$, we assume that for every round i of the algorithm, there exist error terms $\epsilon_{ot1}, \epsilon_{ot2}, \epsilon_{fit}$, such that round i of OTS finds transport T_i satisfies:

$$\mathcal{T}_c^\beta(g_i\#\mu, \hat{\nu}) \leq C_\mu^\beta(T_i \circ g_i, g_i) \leq \mathcal{T}_c^\beta(g_i\#\mu, \hat{\nu})(1 + \epsilon_{ot1}), \text{ (approximate optimality)} \quad (5)$$

$$\mathcal{T}_c^\beta(T_i\#g_i\#\mu, \hat{\nu}) \leq \epsilon_{ot2} \leq \alpha\mathcal{T}_c^\beta(g_i\#\mu, \hat{\nu}). \text{ (approximate pushforward measure)} \quad (6)$$

Finally we require that round i of FIT finds g_{i+1} satisfies

$$C_\mu^\beta(T_i \circ g_i, g_{i+1}) \leq \epsilon_{fit} \leq \frac{1 - 2\alpha}{1 + \epsilon_{ot1}} C_\mu^\beta(T_i \circ g_i, g_i) \leq (1-2\alpha)\mathcal{T}_c^\beta(g_i\#\mu, \hat{\nu}), \text{ (progress of FIT)} \quad (7)$$

with $g_i\#\mu$ continuous to guarantee the existence of Monge transport plan. Equation 5 is satisfied by OTS by convexity, while equation 7 assures that there is progress to be made in minimizing the Wasserstein distance, otherwise the training should stop.

Furthermore, α is a tunable parameter of our algorithm for large values of which the optimality requirement of OTS is relaxed (allowing early stopping of OTS), while requiring large progress of FIT (preventing early stopping of FIT), and vise versa. This leads to creating intuitive stopping criteria.

We can now show that $\mathcal{T}_c(g_t\#\mu, \hat{\nu}) \leq e^{-t\alpha/\beta}\mathcal{T}_c(g_0\#\mu, \hat{\nu})$, thereby concluding the optimization guarantee for $t \to 0$. The inequality is achieved using the triangle inequality for the optimal transport cost on $T_i\#g_i\#\mu$, the continuity of g_{i+1}, and the assumptions regarding the error terms 5,6 and 7.

3.2 Generalization Guarantee: $\mathcal{T}_c(g_i\#\mu, \nu) \to 0$.

When talking about generalization, we want to ensure that the model fitted via training has low divergence $\mathcal{D}(g_i\#\mu, \cdot)$ from both the underlying ν and sampled distribution $\hat{\nu}$. If $\mathcal{T}_c$ satisfies the triangle inequality, then

$$\mathcal{T}_c(g_i\#\mu, \nu) \leq \mathcal{T}_c(g_i\#\mu, \hat{\nu}) + \mathcal{T}_c(\hat{\nu}, \nu),$$

and $\mathcal{T}_c(\hat{\nu}, \nu) \to 0$ by definition, as the sample size $n \to \infty$, and it remains to bound the other term. From [2, 19], we know that the sample complexity grows exponentially in the dimension. To bypass that we make two parametric assumptions about the underlying distribution ν.

We assume that the *Kantorovich potential* $\hat{\psi}$, defined on $\hat{\nu}$, is induced by a function $\psi \in \Psi$ defined on ν, with Ψ a class of function with the following generalization and approximation guarantees. Including the fixed sample size n in the notation, we require:

1. Approximation condition: For any $\epsilon > 0$, there exists a class of functions Ψ such that
$$\sup_{\psi \in L_1(\nu)} \int \psi^c d\mu + \int \psi d\nu \leq \epsilon + \sup_{\psi \in \Psi} \int \psi^c d\mu + \int \psi d\nu.$$
2. Generalization condition: Given a sample size n and function class Ψ, we suppose that there exists $D_{n,\Psi} \geq 0$ such that with probability at least $1 - \delta$ when sampling n examples from ν, every $\psi \in \Psi$ satisfies
$$\int \psi d\nu \leq D_{n,\Psi} + \int \psi d\hat{\nu}_n.$$

Both conditions can be empirically justified by fitting a neural network to approximate $\hat{\psi}$. The first condition can be mathematically justified through literature in function approximation via neural networks [7, 10, 22], e.g. by increasing the depth of the network. The second assumption is justified through the theory of neural network generalization, such as [1], e.g. using the VC dimension of neural networks.

The two above conditions can be used to prove that for every $\epsilon > 0$, with probability at least $1 - \delta$ over a draw of n samples of ν,

$$\mathcal{T}_c(g_n\#\mu, \nu) \leq D_{n,\Psi} + \epsilon + \mathcal{T}_c(g_n\#\mu, \hat{\nu}_n).$$

Using the previous discussion we have that $D_{n,\Psi} \to 0$, and $\epsilon \to 0$ as $n \to \infty$. The third term converges to 0 as proven in 3.1.

To verify the second guarantee, [5] train $\hat{\psi}$ between the fitted generating distribution $g\#\mu$ and MNIST [14] full dataset $\hat{\nu}$, then fit an MLP of 4 hidden layers, each having 512 neurons to $\hat{\psi}$ on the training dataset $\hat{\nu}_{train}$, measured both on the empirical train and test dataset. In this way, they show that the training error tends to zero, and ψ is almost identical in values with $\hat{\psi}$ when measured on $\hat{\nu}$. In a parallel work, [18] show this generalization condition without empirical testing. Namely, ψ is parametrized as a neural network in the OT algorithm.

4 Experiments

As a novelty, we present the results of the proposed overall algorithm on the Kuzushiji-MNIST dataset [6], and notice the effect of the difference in crispiness of the generated samples in the ℓ_1 (Figure 1) and ℓ_2 (Figure 2) metric spaces, as well as the importance of the alternative procedure of the overall algorithm (Figure 3), when the MLP is trained for 100 iterations.

Running the model on CIFAR10 and CelebA datasets [13, 15], [5] are able to show that while this model has the lowest Wasserstein-1 distance compared to other GAN-based models, it produces identifiable objects that are more blurry than GAN-based results. This leads them to conjecture that explicit minimization of the Wasserstein distance on pixel-wise metrics such as ℓ_1, ℓ_2 leads to a mode-collapse-free regularization effect. The objective of a low transport cost, leads such our model to cover all the modes, disregarding the sharpness of the generated samples. Thus, it is the model's objective that prevents it from collapsing, at the expense of generating blurry samples. Potential remedies include the use of a perceptual loss and the incorporation of an adversarial metric.

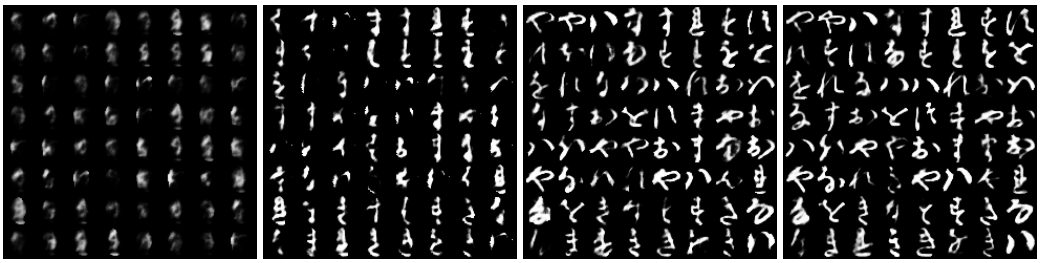

Figure 1: MLP with Wasserstein-1 distance after 1, 5, 25 and 100 iterations.

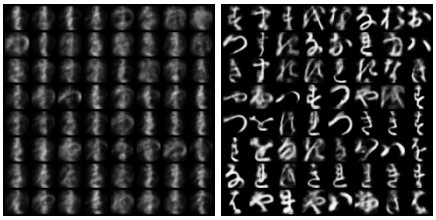

Figure 2: MLP with Wasserstein on ℓ_2 after 1 (left) and 100 (right) iterations.

Figure 3: Non-alternating procedure (left) and alternating procedure (right) after 50 iterations.

5 Explanation of the code

The dataset is loaded (Kuzushiji-MNIST, of dimension 28×28) and batch normalisation is performed for mean 0.5 and standard deviation 0.5 with batch size 64. The neural network implemented is a CNN of 4 hidden layers, each layer has 512 neurons, the activation functions used are ReLU for the first 3 layers, and tanh for the last layer. The optimization method used is Adam [12], with the learning rate of 10^{-4} and 10^{-1} for FIT and OTS respectively. Then, depending on which norm is called (ℓ_1 or ℓ_2), OTS and FIT are run in an alternating manner to train the generative model. Weights of the neural network are initialized according to a normal distribution with zero mean and variance 0.02. Lastly, the empirical stopping criterion for OTS (justified by 3.1) is that the histogram of transportation targets, which is stored in memory, is close to a uniform distribution, and specifically in the range of $[200, 320]$.

6 Conclusion and future work

The goal of [5] was to train a generator network g by minimizing the Wasserstein distance $W(g\#\mu, \nu)$ between the generated distribution $g\#\mu$ and the target distribution ν, where μ is a simple distribution such as uniform or gaussian. Motivated by the ability to compute the optimal transport plan between the generated and underlying distributions, they used it to explicitly minimize the Wasserstein distance $W(g\#\mu, \nu)$ without any adversarial procedure. It was also shown that minimizing this distance does not guarantee better results when compared to other GAN-based models. Interesting parallel work includes that of [4], where a generalization of the Wasserstein distance, the Gromov-Wasserstein discrepancy, and its implementation as a loss function in generative networks across incomparable spaces is investigated. This measure of distance compares the intra-distances of each space and introduces great flexibility into the generator, allowing it to freely alter characteristics of the generated distribution, while learning the basic structure of the reference distribution. However, to train such a network, the generator has to be constrained, the adversary regularized, while ensuring the suitability and numerical stability of the above discrepancy as a loss function.

Candidate Number: 1055360

References

[1] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge University Press, 2009.

[2] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans), 2017.

[3] Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients, 2017.

[4] Charlotte Bunne, David Alvarez-Melis, Andreas Krause, and Stefanie Jegelka. Learning generative models across incomparable spaces. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 851–861. PMLR, 2019.

[5] Yucheng Chen, Matus Telgarsky, Chao Zhang, Bolton Bailey, Daniel Hsu, and Jian Peng. A gradual, semi-discrete approach to generative network training via explicit Wasserstein minimization. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.

[6] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature, 2018.

[7] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[8] Wilfrid Gangbo and Robert J. McCann. The geometry of optimal transportation. *Acta Math.*, 177(2):113–161, 1996.

[9] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*, pages 3440–3448, 2016.

[10] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[11] Leonid V Kantorovich. On the translocation of masses. *Journal of Mathematical Sciences*, 133(4):1381–1382, 2006.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[14] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018.

[16] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

[17] Gabriel Peyré and Marco Cuturi. Computational optimal transport, 2020.

[18] Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation, 2018.

[19] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, Gert RG Lanckriet, et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

[20] Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet. On the empirical estimation of integral probability metrics. *Electron. J. Statist.*, 6:1550–1599, 2012.

[21] Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[22] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.