Convexity Regularizer for Neural Optimal Transport

Koffivi Gbagbe [1] **Bintang A.S.W.A.M** [1] **Nikita Gushchin (TA)** [1]

Abstract

We inserted a convexity regularizer in the loss of the neural optimal transport algorithm proposed in Korotin et al. (2023) to compute the optimal transport map (plan) for strong transport cost and investigate the quality of the inverse mapping. We also evaluate the performance of our proposed algorithm on an unpaired image-to-image translation problem using a colored images of MNIST digits. Theoretical-based computational intensive and implications of the results are discussed.

Keywords: convexity regularizer, OT map, inverse target-to-input mapping, adversarial training, convex optimal transport.

Github repo: bin-koff

1. Introduction

Using neural networks to solve continuous optimal transport problem is a promising approach especially for unpaired style-transfer problem (see figure 1). The idea behind this method is to learn a one-to-one mapping (OT map) between the source and target data distributions. The proposed method by Korotin et al. (2023) uses adversarial training similar to Generative Adversarial Networks (GANs), which is not very stable. However, unlike GANs, this method's optimal "discriminator" must be convex, and its gradient can be used for inverse mapping from the target distribution to the source distribution. To address the issue of stability, we find necessary to insert a convexity regularizer (kind of gradient penalty in WGAN-GP) in the loss of the neural optimal transport algorithm to improve its stability during the training phase while improving the high-quality of the inverse target-to-input mapping.

General notations:
$\mathcal{X}$ and $\mathcal{Y}$ are some polish spaces; $\mathcal{P}(\mathcal{X})$ and $\mathcal{P}(\mathcal{Y})$ are sets of probability defined respectively on $\mathcal{X}$ and $\mathcal{Y}$; For a measurable map $T : \mathcal{X} \longrightarrow \mathcal{Y}$ the operator $T_{\#}$ denotes the so called push-forward operator.

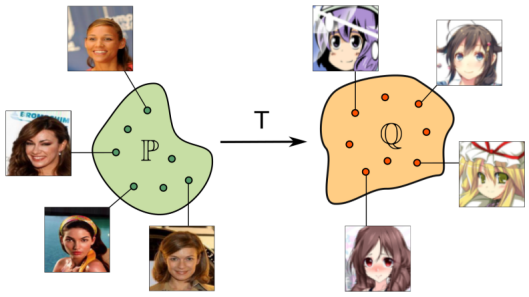

Figure 1. Example of unpaired image-to-image style translation. Inputs: two unpaired dataset sample empirically from some probability distribution function $\mathbb{P}$ and $\mathbb{Q}$; Output: a Style translation map $T : \mathcal{X} \longrightarrow \mathcal{Y}$ such that $T_{\#}\mathbb{P} = \mathbb{Q}$.

2. Preliminaries

2.1. Monge's Optimal Transport formulation

For $\mathbb{P} \in \mathcal{P}(\mathcal{X}), \mathbb{Q} \in \mathcal{P}(\mathcal{Y})$ and a cost function $c : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$, Monge's primal formulation of the cost is

$$\textbf{Cost}(\mathbb{P}, \mathbb{Q}) \stackrel{\text{def}}{=} \inf_{T_{\#}\mathbb{P}=\mathbb{Q}} \int_{\mathcal{X}} c(x, T(x)) d\mathbb{P}(x) \qquad (1)$$

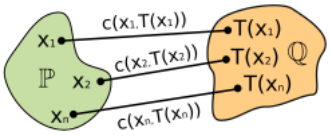

Figure 2. Monge's OT formulation.

The minimum is taken over the set of measurable functions called transport maps $T : \mathcal{X} \longrightarrow \mathcal{Y}$ that maps the marginal distribution $\mathbb{P}$ to $\mathbb{Q}$. The optimal transport map T^* is called the OT map. It is important to note that the equation (1) is not symmetric and for some $\mathbb{P}, \mathbb{Q}$ it may happen that their is no T such that $T_{\#}\mathbb{P} = \mathbb{Q}$(does not allow mass splitting) therefore Kantorovitch (1958) proposed the following formulation

Machine Learning 2023 Course [1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Koffivi Gbagbe <koffivi.gbagbe@skoltech.ru>, Bintang A.S.W.A.M <BintangAlamSemesta.WisranAm@skoltech.ru>, Nikita Gushchin (TA) <Nikita.Gushchin@skoltech.ru>.

Final Projects of the Machine Learning 2023 Course, Skoltech, Moscow, Russian Federation, 2023.

2.2. Strong optimal transport formulation

To overcome the mass splitting problem Kantorovitch (1958) proposed the following reformulation

$$\textbf{Cost}(\mathbb{P},\mathbb{Q}) \stackrel{\text{def}}{=} \inf_{\pi\in\Pi(\mathbb{P},\mathbb{Q})} \int_{\mathcal{X}\times\mathcal{Y}} c(x,T(x))d\pi(x,y) \quad (2)$$

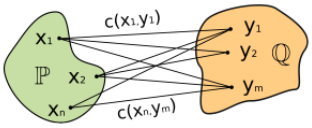

Figure 3. Strong OT formulation

and the minimum is taken over all the transport plan π which represent all distributions on $\mathcal{X}\times\mathcal{Y}$ whose marginals are $\mathbb{P}$ and $\mathbb{Q}$ and the optimal plan denoted by π^*. Lastly, an example of strong OT cost for $\mathcal{X}=\mathcal{Y}=\mathcal{R}^D$, is the ($p$-th power of) Wasserstein-p distance $\mathcal{W}_p$, i.e. formulation (2) in which $c(x,y)=\frac{\|x-y\|^p}{2}$, $(p=2)$.

3. Related work

We overview the latest research on both OT maps and OT costs. For OT maps (OT plans), this topic is still evolving but challenging, because there is only a few number of scalable methods have progressively been developed in response to it. According to Korotin et al. (2023), to compute the OT map, such methods shall be mathematically expressed in terms of dual and primal formulation. In dual formulation, corresponding methods are created for OT map with the quadratic cost, i.e. Wasserstein-p distance($p=2$) while also have simpler optimization procedures. Aside from that, this formulation generates in a version of unbalanced optimal transport that leads to sparse solutions, which either hinders the interpretability for machine learning tasks or they do not perform well in practice. Conversely, such methods based on primal formulation using generative models and yields complex optimization objectives with several adversarial regularizers, in constrast for practical applications it is quite tough to setup since they need careful selection of hyper-parameters.

On the other hand, OT costs are basically treated as the loss function to learn generative models, the most notable example of OT costs in large-scale machine learning is Wasserstein GANs (Arjovsky et al. (2017); Gulrajani et al. (2017)).

4. Algorithm for learning the OT plan

4.1. Neural optimal transport (NOT)

Korotin et al. (2023) proposed a deep neural networks based novel algorithm to compute deterministic and stochastic OT plans which is designed for both weak and strong optimal transport costs.

Algorithm 1 Neural optimal transport (NOT)

Input: distributions $\mathbb{P},\mathbb{Q},\mathbb{S}$ accessible by samples mapping network T_θ; potential network $f_w:\mathbb{R}^Q\mapsto\mathbb{R}$; number of inner iterations K_T; (strong) cost $C:\mathcal{X}\times\mathcal{P}(\mathcal{Y})\mapsto\mathbb{R}$; empirical estimator $\hat{C}(x,T(x,Z))$ for the cost
Output: learned stochastic OT map T_θ representing an OT plan between distributions $\mathbb{P},\mathbb{Q}$
repeat
  Samples batches $Y\sim\mathbb{Q}, X\sim\mathbb{P}$; for each $x\in X$ sample batch $Z_x\sim\mathbb{S}$;
  $\mathcal{L}_f \leftarrow \frac{1}{|X|}\sum_{x\in X}\frac{1}{|Z_x|}\sum_{z\in Z_x} f_w(T_\theta(x,z)) - \frac{1}{|Y|}\sum_{y\in Y} f_w(y)$;
  Update w by using $\frac{\partial\mathcal{L}_f}{\partial w}$
  for $k_T=1$ **to** K_T **do**
    Sample $X\sim\mathbb{P}$, for each $x\in X$ sample batch $Z_x\sim\mathbb{S}$;
    $\mathcal{L}_f \leftarrow \frac{1}{|X|}\sum_{x\in X}\Big[\hat{C}(x,T_\theta(x,Z_x)) - \frac{1}{|Z_x|}\sum_{z\in Z_x} f_w(T_\theta(x,z))$
    Update θ by using $\frac{\partial\mathcal{L}_T}{\partial\theta}$
  end for
until *converged*

For more details about Algorithm 1 see Korotin et al. (2023)

4.2. Regularized Neural optimal transport (Reg-NOT)

In order to add a regularization term to the loss of the previous algorithm we used the following steps:

- From the dual optimization problem we deduce that the potential
$$f(x)=f^*(x)\frac{|X|}{2} \quad (3)$$
- By posing $f(x)=\frac{||x||^2}{2}-\psi(x)$ and equating with equation (3) we obtain
$$\psi(x)=-f^*\frac{|X|}{2}+\frac{\|x\|^2}{2} \quad (4)$$
- We desire ψ to be a convex function which satisfies the Jensen's inequality
$$\psi(tx_1+(1-t)x_2)-t\psi(x_1)-(1-t)\psi(x_2)\le 0,\ t\in[0,1] \quad (5)$$

- Then, we define and penalize the loss function to hold Jensen's inequality

$$\mathcal{L}(X_1, X_2) = \psi(tX_1 + (1-t)X_2) \\ -t\psi(X_1) - (1-t)\psi(X_2) \leq 0 \tag{6}$$

- To make loss positive we may apply the **ReLU** function; hence let define

$$\mathrm{R}(X_1, X_2) = \mathbf{ReLU}(\mathcal{L}(X_1, X_2)) \tag{7}$$

- Finally we uses the mean of $\mathrm{R}(X_1, X_2)$ as a regularization term which can be multiplied by a given learning rate.

Following these steps we modified the previous algorithm as follow:

Algorithm 2 Regularized Neural optimal transport (Reg-NOT)

Input: distributions $\mathbb{P}, \mathbb{Q}, \mathbb{S}$ accessible by samples mapping network T_θ; potential network $f_w : \mathbb{R}^Q \mapsto \mathbb{R}$; number of inner iterations K_T; (strong) cost $C : \mathcal{X} \times \mathcal{P}(\mathcal{Y}) \mapsto \mathbb{R}$; empirical estimator $\hat{C}(x, T(x, Z))$ for the cost
Output: learned stochastic OT map T_θ representing an OT plan between distributions $\mathbb{P}, \mathbb{Q}$
repeat
  Samples batches $Y \sim \mathbb{Q}, X \sim \mathbb{P}$; for each $x \in X$ sample batch $Z_x \sim \mathbb{S}$;
  $Y^{'} \leftarrow Y$ and divide $Y^{'}$ into two equal samples Y_1 and Y_2
  $R(y_1, y_2) \leftarrow \mathrm{ReLU}(\psi(ty_1 + (1-t)y_2) - t\psi(y_1) - (1-t)\psi(y_2)); t \in [0,1], y_1 \in Y_1, y_2 \in Y_2$
  $\mathcal{L}_f \leftarrow \frac{1}{|X|}\sum_{x\in X}\frac{1}{|Z_x|}\sum_{z\in Z_x} f_w(T_\theta(x,z)) - \frac{1}{|Y|}\sum_{y\in Y} f_w(y) + \alpha\frac{1}{|Y_1|}\sum_{(y_1,y_2)\in Y_1\odot Y_2} R(y_1, y_2)$;
  Update w by using $\frac{\partial \mathcal{L}_f}{\partial w}$
  for $k_T = 1$ **to** K_T **do**
    Sample $X \sim \mathbb{P}$, for each $x \in X$ sample batch $Z_x \sim \mathbb{S}$;
    $\mathcal{L}_f \leftarrow \frac{1}{|X|}\sum_{x\in X}\Big[\hat{C}(x, T_\theta(x, Z_x)) - \frac{1}{|Z_x|}\sum_{z\in Z_x} f_w(T_\theta(x,z))$
    Update θ by using $\frac{\partial \mathcal{L}_T}{\partial \theta}$
  end for
until *converged*

Here the set $Y_1 \odot Y_2 = \{(y_1, y_2)$ s.t y_1 and y_2 have the same index in Y_1 and $Y_2\}$ and α is the learning rate multiplied by the regularization term.

5. Experiments and Results

In this section, we will represent our experiment on toy dataset, i.e. color-MNIST, here X represents colored images of MNIST digit 2 whereas Y represents colored images of MNIST digit 3. Our task is to map digit 2 to the digit 3 of corresponding color for unpaired style transfer image-to-image translation. Furthermore, we have eventually been running a single case which is regularization with learning rate $\alpha = 0.01$. On the flip side, it requires to introduce loss function (denoted as f-loss) and Frechet Inception Distance (FID). Loss function is basically to measure how far an estimated value is from its true value, in which it maps decisions to their associated costs. Acorrding to Borji (2018) , FID is a metric for evaluating the quality of generated images and specifically developed to evaluate the performance of generative adversarial networks. A lower FID indicates smaller distances between synthetic and real data distributions, in short produces higher-quality images; conversely, a higher FID correlates higher distances between real and synthetic data distribution, in short produces lower-quality images.

5.1. Generator and Discriminator architecture

For our experiment we modified the implementation of algorithm 1 from the Github repository iamalexkorotin. The code used as Generator UNet which architecture contains two paths: a contraction path called encoder which is just a stack of convolutional and max pooling layers and is used to capture the context in an image. The second path is known as decoder and is the symmetric expanding path and is used to enable precise localization using transposed convolutions. UNet only contains Convolutional layers and does not contain any Dense layer hence it can accept image of any size. On the other hand the code uses as Discriminator ResNet (Residual Network) which is based on the the concept called residual blocks and allows the model to skip layers without affecting performance.

5.2. Source-to-Target Map of Regularized Neural Optimal Transport (Reg-NOT) with $\alpha = 0.01$

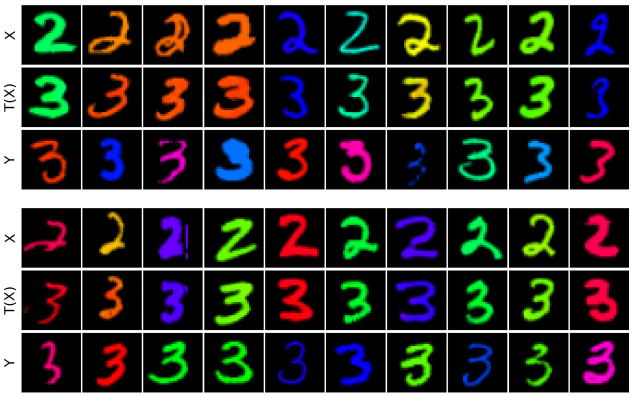

Figure 4. Fixed and Random test image after convergence

We have been running our training using GPU supported by the computing server, Tesla P1000-PCIE-12GB which consumes around one and half day for performing regularization task with $\alpha = 0.01$. Figure 4 represents the results of the training phase at the $53900-$th iteration. Visualy we can see that the model successffuly learned to transfert the style of digits '2' to digits '3'. Bellow, the figure 5 depicts the FID score of the training which slightly alleviated to final value in between zero and ten and hence indicates that the quality of images generated by the generator is similar to the sample ones (both fixed and random images) during training process of direct map. Meanwhile the loss function with regularization irregularly risen and fall as well as had asymptotically been declining to zero-value at the end of iteration (figure 5).

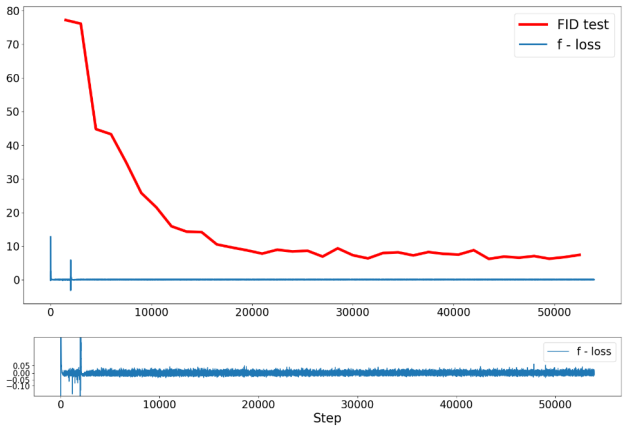

Figure 5. FID and Loss-function at Reg-NOT training

5.3. Target-to-Source Map with the Discriminator of Reg-NOT

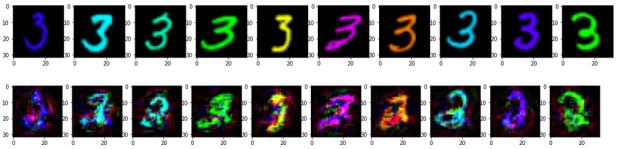

Figure 6. Inverse mapping from digits '3' to '2'

Despite the fact of our proposed algorithm converged, there is still small computational issues to conduct the inverse map task. However, it existed a few of colors (figure 6) appeared in preserved state. The only issue is explicitly lying on blurred state due to the emergence of particular noise.

5.4. Comparing NOT and Reg-NOT on Inverse Map Problem

Note: The models used are saved at the 7499 step

Using Reg-NOT

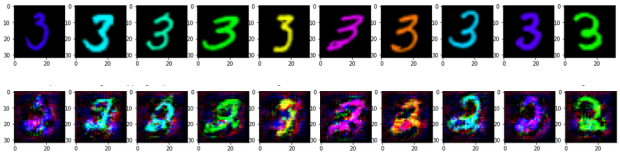

Figure 7. Inverse mapping "3" to "2" using Reg-NOT

Using NOT

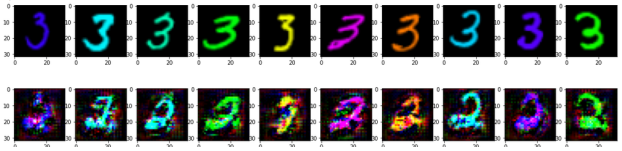

Figure 8. Inverse mapping "3" to "2" using NOT

From both experiments, it vividly seems that neither NOT nor reg-NOT is eventually mapped the digits three 3 correctly to the digits two 2.

6. Conclusion and Future Work

In particular, we managed to retain from our experiments by adding a convexity regularizer to the Neural Optimal Transport algorithm, the algorithm is still converged and able to completely learn for the source-to-target OT map. However, it remains challenging to resolve target-to-source OT map including the issue of stability. However, we definitely consider this as our upcoming research investigation by taking into account the stability of Wasserstein-GP and its improvement.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

Borji, A. Pros and cons of gan evaluation measures, 2018.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.

cc/paper_files/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf.

Kantorovitch, L. On the translocation of masses. *Management Science*, 5:1–4, 1958.

Korotin, A., Selikhanovych, D., and Burnaev, E. Neural optimal transport, 2023.

A. Team member's contributions

We both did almost everything together hence it is meaningless to explicitly list the same thing for both of us.