Generative modeling project : Neural Optimal Transport

Paul Barbier[1] **and Bastien Le Chenadec**[1]

[1]École des Ponts ParisTech, Master MVA

Contribution Statement

1 Introduction

Optimal Transport (OT) is a mathematical framework that aims to find the most efficient way to transport a distribution of mass to another. This framework has been used extensively in the context of generative models, for instance as a loss function in the training of Generative Adversarial Networks (GANs) or by learning a mapping between two distributions. In this project, we aim to study the paper "Neural Optimal Transport" (Korotin, 2023) [1] which introduces an algorithm to train a neural network to learn the optimal transport between two distributions.

2 Background on Optimal Transport

Let μ and ν be two probability distributions on $\mathcal{X}$ and $\mathcal{Y}$ respectively (typically $\mathcal{X}, \mathcal{Y} = \mathbb{R}^n, \mathbb{R}^m$). To give a meaning to "efficiently" transporting mass, we need to define a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that quantifies the cost of transporting a unit of mass in $\mathcal{X}$ to one in $\mathcal{Y}$. The (Monge) optimal transport problem consists in finding a **transport map** $T^* : \mathcal{X} \to \mathcal{Y}$ such that :

$$T^* \in \underset{T\#\mu=\nu}{\text{Argmin}} \int_{\mathcal{X}} c(x, T(x)) d\mu(x) \tag{1}$$

where $T\#\mu$ is the pushforward distribution of μ by T, defined by $(T\#\mu)(A) = \mu(T^{-1}(A))$ for any measurable set $A \subset \mathcal{Y}$. This formulation calls for a deterministic mapping from $\mathcal{X}$ to $\mathcal{Y}$, which is not always desirable or feasible under general assumptions. Kantorovich introduced a more general OT problem that aims at finding a **transport plan** $\pi^* \in \Pi(\mu, \nu)$ in the set of joint distributions on $\mathcal{X} \times \mathcal{Y}$ with marginals μ and ν such that :

$$\pi^* \in \underset{\pi\in\Pi(\mu,\nu)}{\text{Argmin}} \int_{\mathcal{X}\times\mathcal{Y}} c(x, y) d\pi(x, y) \tag{2}$$

In general the solution to the Kantorovich problem is stochastic, but in some cases it may be deterministic in which case it is also a solution to the Monge problem. Following this idea of stochasticity in the solution, weak OT was introduced as a relaxation of the Kantorovich problem, where the cost function is of the form $C : \mathcal{X} \times \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$. In this case the weak OT problem writes :

$$\pi^* \in \underset{\pi\in\Pi(\mu,\nu)}{\text{Argmin}} \int_{\mathcal{X}} C(x, \pi(\cdot|x)) d\pi(x) \tag{3}$$

where $\pi(\cdot|x)$ is the conditional distribution of π given x and $d\pi(x)$ is the marginal distribution of π on $\mathcal{X}$.

Building on this framework, (Korotin, 2023) [1] introduce **stochastic maps** $T : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ where $\mathcal{Z}$ is a latent space corresponding to the randomness in the transport. They show that the weak optimal transport problem can be reformulated and solved by a SGAD algorithm. This approach is particularly interesting in the context of generative modeling, as it allows to learn a stochastic mapping between two distributions.

3 Experiments

For this project, we conducted experiments using the code provided by the author. You can find our scripts at https://github.com/bastienlc/NOT. We've done a first experiment with generated synthetic data and a second one with a more realistic use-case using a large dataset.

3.1 Synthetic data

3.2 Real dataset

Authors made experiments using faces from animes and real persons and the results were funny so we decided to make a similar one with a different dataset. More precisely, we took the CelebA dataset [2] which comprises 200k images of celebrity faces with attribute annotations. For the second dataset, we chose CartoonSet100K, a dataset introduced in [3]. It's a large-scale dataset containing 100k of 2D generated cartoon avatar images.

Figure 1: Images samples from CartoonSet100K

For this large-scale experiment we used a cloud instance with a NVidia A100 GPU with 40GB of memory. We increased the batch size per dataset from 64 to 128 and we used 128×128 images in place of 64×64 to get generated images with a nicer resolution. Increasing the batch size lead to better memory utilisation and quicker convergence as we'll see in the next paragraphs.

On the figure below, you can see how the loss function evolves throughout the training. Unusually, the loss function here is fluctuating around the constant value 0 and that's expected: explain why it's expected TODO.

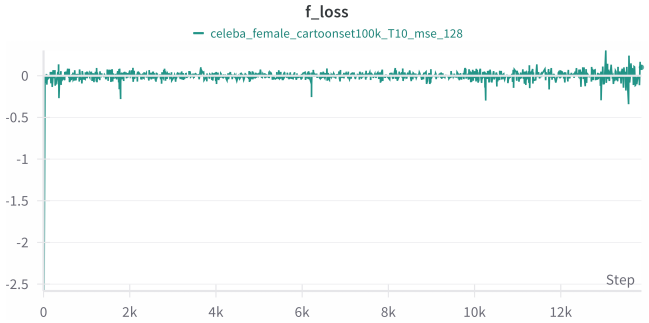
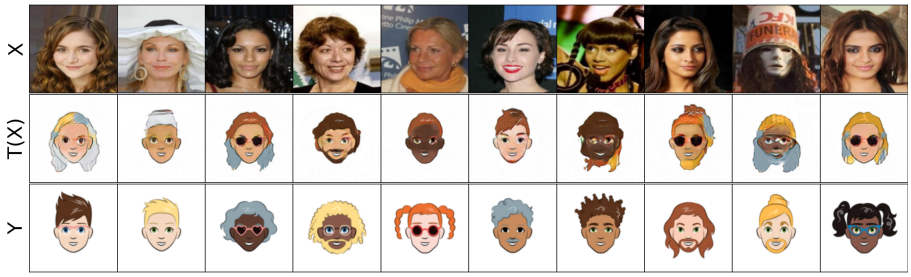
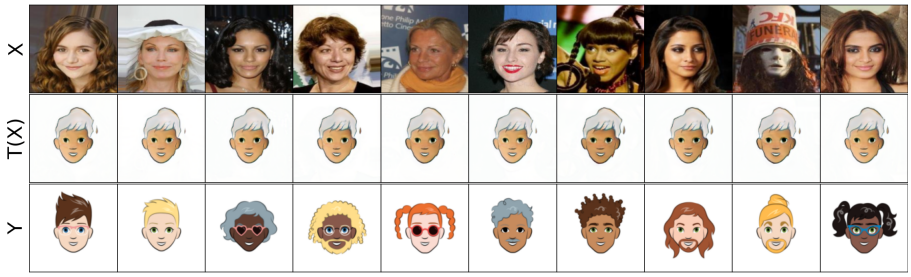


Figure 2: Loss function

(a) Iteration 1



(b) Iteration 5000



(c) Iteration 13800

Figure 3: Test images

Those experiments are quite long in time (about a day for $\tilde{1}$5k iterations with the parameters given above) so we restrict ourselves to only train the strong formulation model and the results for a fixed sample of the test set are given in 3.

3.2.1 Observations

Figure 3a lets us see the model initialisation.

Furthermore, we witnessed realistic mappings of the real faces onto the cartoon avatar distribution around iteration 5000, see 3b. This is far less than what the authors observed in their experiments, this number was closer to 40k iterations for convergence. However, our images have an empty background so it might be easier in comparison with anime face images which contain a complex surrounding as they are extracted from animes. One can see that the color of the skin and the hair map well. However, the model struggles when there are glasses on the target image. It translates glasses into dark black holes. It's very surprising to see new avatars thanks to those mappings.

At this point, we thought we were far from the end and we decided to continue the training for a night. Unfortunately, few thousands iterations later mode collapses appeared with the same avatar being mapped for a given batch. Surprisingly, the constant predicted face is changing with training without changing the mode collapse.

4 CONCLUSION

Random sentence for the conclusion in the meantime.

REFERENCES

[1] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. *arXiv (Cornell University)*, 1 2022. doi: 10.48550/arxiv.2201.12220. URL https://arxiv.org/abs/2201.12220.

[2] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[3] Amelie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, and Kevin Murphy. XGAN: unsupervised image-to-image translation for many-to-many mappings. *CoRR*, abs/1711.05139, 2017. URL http://arxiv.org/abs/1711.05139.

APPENDIX

A FIGURES