Extending Neural Optimal Transport for 3D

Sergei Kholkin, Maksim Bobrin, Anastasia Batsheva, Artem Basharin
Skolkovo Institute of Science and Technology, Skolkovo, Russia

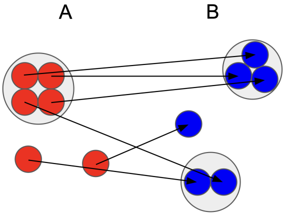
Introduction to Machine Learning, 2023

24.04.2022

Outline

What is transport?

Suppose we have two bunch of points A and B. And we want to **transport** (aka turn) A into B. In the easiest case we can just pull each of A's point to one of B:



$$a_i \to b_j$$

Transport cost: $c(a_i, b_j) = \text{for example} = \|a_i - b_j\|_2^2$

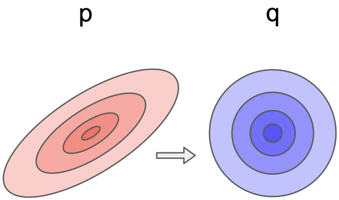
Total transport cost: $C(A, B) = \sum_{a_i \in A} \sum_{b_j \in B} c(a_i, b_j) \underbrace{T(a_i, b_j)}_{\text{plan}}$

Transport plan: $T(a_i, b_j) = w$ - how many points we transport from a_i to b_j.

Condition:

$$\sum_{b_j \in B} T(a^i, b^j) = w_a(a^i), \quad \sum_{a_i \in A} T(a^i, b^j) = w_b(b^j)$$

where $w_a(a_i), w_b(b_j)$ - numbers of A's points in a_i and B's points in b_j respectively.



Distribution transport from $p(x)$ to $q(x)$:

Condition: $\int T(x_p, x_q)\ dx_p = p(x_q), \quad \int T(x_p, x_q)\ dx_q = q(x_p)$

Total transport cost: $C(p, q) = \int \int c(x_p, x_q) T(x_p, x_q)\ dx_p\ dx_q$

$\boxed{\textbf{Optimal transport plan}: \arg\min_T C(p, q)}$

How to find an OT plan: related works

1. Kullback–Leibler divergence - fails when support of distributions is not the same.
2. Wasserstein distance - out of scope - computes only OT costs, not OT plans.

How to find an OT plan: proposed method

1. Find OT map $f : X \to Y$ minimized an expected transportation cost (Monge):

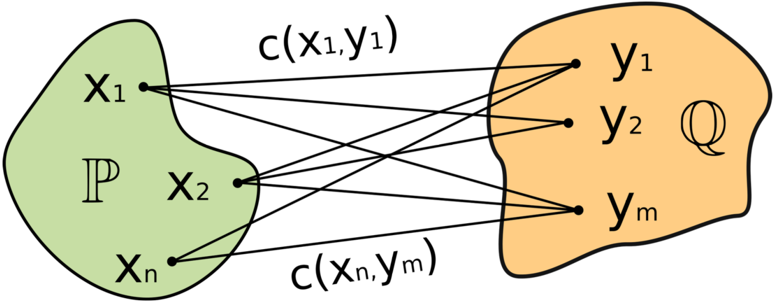
$$M(\mu, \nu) = \inf_T \{ \int_X c(x, T(x)) d\mu : T_{\#}\mu = \nu \}$$

2. Monge formulation is not suitable for particle-like distributions, this is when Kantorovich relaxation helps!
3. Idea: split mass!

$$K(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \{ \int_{X \times Y} c(x, y) d\pi(x, y) \}$$

Main idea of Neural Optimal Transport: Let Neural Network decide optimal transporation plan!

- NOT paper introduced generalization, namely weak OT cost (completely different what WGAN makes)



Related work

OT problem has two subproblems - strong (deterministic) OT and weak OT. Existing methods are designed only for a strong OT.

One can solve OT problem in either primal or dual formulation.

For primal formulation using GANs, careful selection of hyperparameters is needed.

Dual problem can be formulated as follows:

$$\text{Cost}(\mathbb{P}, \mathbb{Q}) = \sup_{f} \int_{\mathcal{X}} f^{C}(x) d\mathbb{P}(x) + \int_{\mathcal{Y}} f(y) d\mathbb{Q}(y)$$

So, most of the methods are using dual formulation and are usually using W_2 as cost function. Existing methods include

- ICNN (input convex neural networks), which have strong theoretical bounds, but are useless for large scale problems
- Entropy regularization OT. Downside: hard to sample.
- MM:R (Maximin reversed). Has less downsides, but still limited to determenistic approach

Dim	2	4	8	16	32	64	128	256
[MMv1]	0.99	0.99	0.99	0.99	0.98	0.97	0.99	0.99
[MM]	0.99	0.99	0.99	0.99	0.99	0.99	0.99$\looparrowright$	0.99$\looparrowright$
[MM:R]	0.99	1.00	1.00	0.99	1.00	0.98	$\nrightarrow$	$\nrightarrow$
[MMv2]	0.99	0.99	0.99	0.99	0.99	0.96$\looparrowright$	0.99$\looparrowright$	0.99$\looparrowright$
[MMv2:R]	0.99	1.00	0.97	0.96	0.99	0.97	0.99	1.00
[W2]	0.99	0.99	0.99	0.99	0.99	0.97	1.00	1.00
[W2:R]	0.99	1.00	0.98	0.98	0.99	0.97	1.00	1.00
[MM-B]	0.99	1.00	0.98	0.96	0.96	0.94	0.93	0.93
[LS]	0.94	0.86	0.80	0.80	0.81	0.83	0.82	0.81
[L]	0.75	0.80	0.73	0.73	0.76	0.75	0.77	0.77

Methods

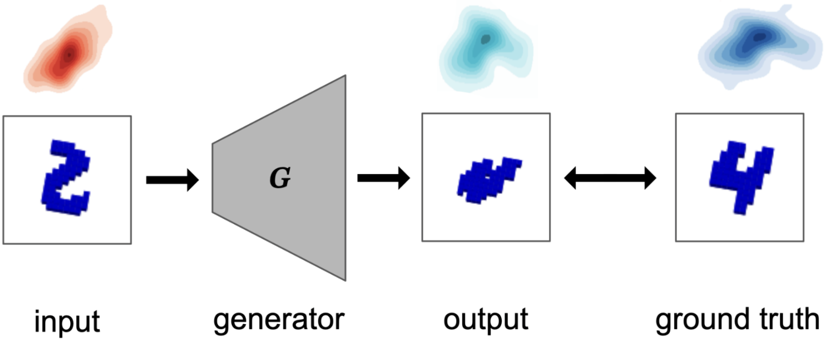
Our model:

$$Cost(\mathbb{P}, \mathbb{Q}) = {}^{sup}_{f}{}^{inf}_{T} \mathcal{L}(f, T)$$

$$\mathcal{L}(f, T) = \int_{\mathcal{Y}} f(y) d\mathbb{Q}(y) + \int_{\mathcal{X}} c(x, T(x)) d\mathbb{P}(x) - \int_{\mathcal{Y}} f(T(x)) d\mathbb{P}(x)$$

T - transport plan
f - potential function
MaxiMin optimization objective

Our model:



$T : \mathbb{R}^{3\times16\times16} \to \mathbb{R}^{3\times16\times16}$ U-net with 2D convolutions replace by 3D convolutions
$f : \mathbb{R}^{3\times16\times16} \to \mathbb{R}$ ResNet architecture with replaced 2D convolutions as well

Our distributions:

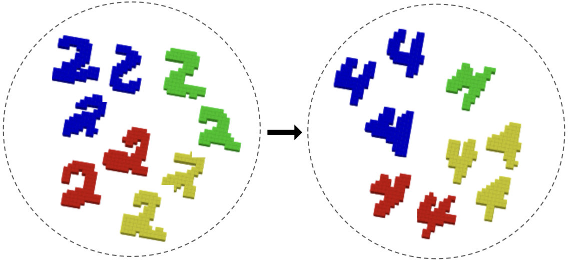


Figure: 3D-MNIST

Our task: to find an optimal transport which turn twos into fours.

Our results: $2 \rightarrow 4$

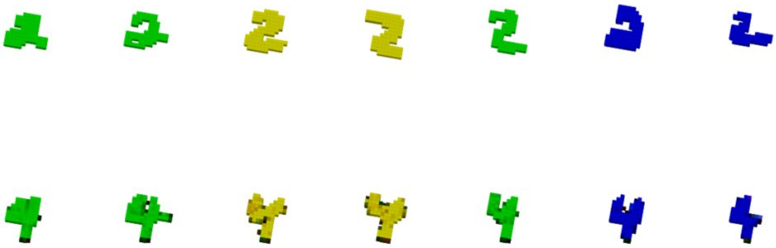


Figure: Here are the original twos at the top and the generated (transported) fours at the bottom.

Our results: $4 \to 2$

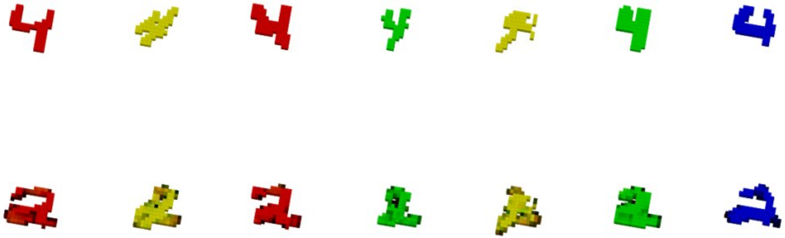


Figure: Here are the original fours at the top and the generated (transported) twos at the bottom.

Conclusions

Within the project, we ...

- builded a transport plan by neural network modeling
- showed that colors of digits don't change
- obtained results very similar to the ground truth

Team contributions

- Sergei Kholkin: Implementing code
- Anastasia Batsheva: Presentation/Report
- Maksim Bobrin: Running experiments, refactoring code
- Artem Basharin: Presentation and documentation

Thank you for attention!

Questions? Link to Github