Revisiting Sliced Wasserstein on Images

Florentin Coeurdoux

29/04/2022

Contents

1 Revisiting Sliced Wasserstein on Images: From vectorization to Convolution

This is a summary of the paper (Nguyen and Ho 2022)

- Classic Sliced Wasserstein (SW) is defined between two probability measure that have realizations as vectors.
- Vectorization of images has limitations:
  - Do not capture spacial structure of images.
  - Memory efficiency, since each slicing is a vector with the image dimensions

=> They propose Convolutional Sliced-Wasserstein to overcome this problem.

2 Recall

2.1 Classic SW on Tensor.

Classical SW has a complexity of $O\left(m^3 \log m\right)$ when the probability measures have at most m supports. Wasserstein distance also suffers from the curse of dimensionality, namely, its sample complexity is at the order of $O\left(n^{-1/d}\right)$ where n is the sample size.

$\mu, \nu \in \mathcal{P}_p\left(\mathbb{R}^{c \times d \times d}\right)$ for number of channels $c \geq 1$ and dimension $d \geq 1$, works as follows:

1. Vectorize x as a vector of size $\mathbb{R}^{c \cdot d^2}$
2. Stack each vectorized image on the training batch into a matrix $X \in \mathbb{R}^{n \times c \cdot d^2}$.
3. Project with L vectors from unit sphere (we sample a matrix of size $\mathbb{R}^{L \times c \cdot d^2}$)
4. Return the L projected probability measure.

2.2 Convolution operator on Tensors

First introduice in (Lecun et al. 1998) convolution operator on a tensor given :

- number of channels $c \geq 1$
- dimension $d \geq 1$
- stride size $s \geq 1$
- dilation size $b \geq 1$
- size of kernel $k \geq 1$

The convolution of a tensor $X \in \mathbb{R}^{c \times d \times d}$ with a kernel size $K \in \mathbb{R}^{c \times k \times k}$ is:

$$X \overset{s,b}{*} K = Y, \quad Y \in \mathbb{R}^{1 \times d' \times d'}$$

where $d' = \frac{d-b(k-1)-1}{s} + 1$. and has a complexity of $\mathcal{O}\left(c\left(\frac{d-b(k-1)-1}{s}+1\right)^2 k^2\right)$

Has a recall, the output height and with of a convolution operation is :

$$d_{out} = \frac{d(+padding) - k}{s} + 1$$

Exemple : if $x \in \mathbb{R}^{3 \times 28 \times 28}$ and $K \in \mathbb{R}^{3 \times 2 \times 2}$ with a stride $s = 2$ and no padding, the output will be $d_{out} = 1 + (28 - 2)/2 = 14$ and thus $x_{out} \in \mathbb{R}^{3 \times 14 \times 14}$

3 Convolution Slicer

Definition. (Convolution Slicer) For $N \geq 1$, given a sequence of kernels $K^{(1)} \in \mathbb{R}^{c^{(1)} \times d^{(1)} \times d^{(1)}}, \ldots, K^{(N)} \in \mathbb{R}^{c^{(N)} \times d^{(N)} \times d^{(N)}}$, a convolution slicer

$\mathcal{S}\left(\cdot \mid K^{(1)}, \ldots, K^{(N)}\right)$ on $\mathbb{R}^{c \times d \times d}$ is a composition of N convolution functions with kernels $K^{(1)}, \ldots, K^{(N)}$ (with stride or dilation if needed) such that:

$$\mathcal{S}\left(X \mid K^{(1)}, \ldots, K^{(N)}\right) \in \mathbb{R} \quad \forall X \in \mathbb{R}^{c \times d \times d}$$

The idea of the convolution slicer is to progressively map a given data X to a one-dimensional subspace through a sequence of convolution kernels, which capture spatial relations across channels as well as local information of the data

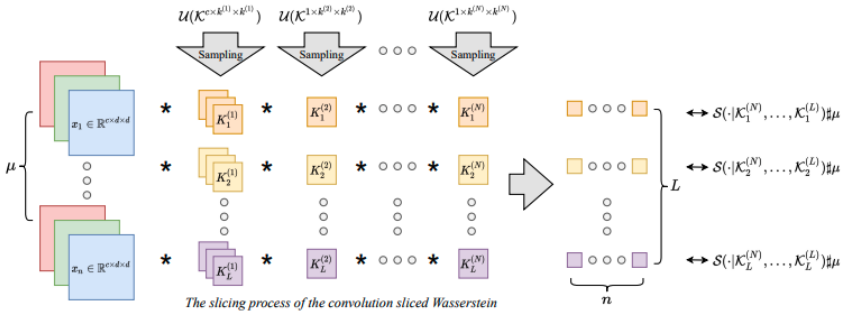

Figure 2: The convolution slicing process (using the convolution slicer). The images $X_1, \ldots, X_n \in \mathbb{R}^{c \times d \times d}$ are directly mapped to a scalar by a sequence of convolution functions which have kernels as random tensors. This slicing process leads to the convolution sliced Wasserstein (11) on images.

Figure 1: Convolutional slicing process

The paper proposed tree types of convolutional slicers :

- convolutional-base Slicer (comp and project $\mathcal{O}\left(cd^2 + d^4\right)$ and $\mathcal{O}\left(c + d^2\right)$)
- convolutional-stride Slicer ($\mathcal{O}\left(cd^2\right)$ and $\mathcal{O}\left(c + [\log_2 d]\right)$)
- convolutional-dilatation Slicer ($\mathcal{O}\left(cd^2\right)$ and $\mathcal{O}\left(c + [\log_2 d]\right)$)

3.1 Convolutional Base Slicer

Given $X \in \mathbb{R}^{c \times d \times d} (d \geq 2)$, 1. When d is even, $N = [\log_2 d]$, sliced kernels are defined as $K^{(1)} \in \mathbb{R}^{c \times \left(2^{-1}d+1\right) \times \left(2^{-1}d+1\right)}$ and $K^{(h)} \in \mathbb{R}^{1 \times \left(2^{-h}d+1\right) \times \left(2^{-h}d+1\right)}$ for $h = 2, \ldots, N-1$, and $K^{(N)} \in \mathbb{R}^{1 \times a \times a}$ where $a = \frac{d}{2^{N-1}}$. Then, the convolution-base slicer $\mathcal{CS} - b\left(X \mid K^{(1)}, \ldots, K^{(N)}\right)$ is defined as: 2. When d is odd, the convolution-base slicer $\mathcal{CS} - b\left(X \mid K^{(1)}, \ldots, K^{(N)}\right)$ takes the form:

$$\mathcal{CS} - b\left(X \mid K^{(1)}, \ldots, K^{(N)}\right) = \mathcal{CS} - b\left(X_*^{1,1} K^{(1)} \mid K^{(2)}, \ldots, K^{(N)}\right)$$

where $K^{(1)} \in \mathbb{R}^{c \times 2 \times 2}$ and $K^{(2)}, \ldots, K^{(N)}$ are the corresponding sliced kernels that are defined on the dimension $d-1$.

The idea of the convolution-base slicer in Definition 3 is to reduce the width and the height of the image by half after each convolution operator. If the width

and the height of the image are odd, the first convolution operator is to reduce the size of the image by one via convolution with kernels of size 2×2, and then the same procedure as that of the even case is applied. We would like to remark that the conventional slicing of sliced Wasserstein in Section 2 is equivalent to a convolution-base slicer $\mathcal{S}\left(\cdot \mid K^{(1)}\right)$ where $K^{(1)} \in \mathbb{R}^{c \times d \times d}$ that satisfies the constraint $\sum_{h=1}^{c} \sum_{i=1}^{d} \sum_{j=1}^{d} K_{h,i,j}^{(1)2} = 1$.

3.2 Convolutional Stride Slicer

Given $X \in \mathbb{R}^{c \times d \times d} (d \geq 2)$, 1. When d is even, $N = [\log_2 d]$, sliced kernels are defined as $K^{(1)} \in \mathbb{R}^{c \times 2 \times 2}$ and $K^{(h)} \in \mathbb{R}^{1 \times 2 \times 2}$ for $h = 2, \ldots, N-1$, and $K^{(N)} \in \mathbb{R}^{1 \times a \times a}$ where $a = \frac{d}{2^{N-1}}$. Then, the convolution-stride slicer $\mathcal{CS} - s\left(X \mid K^{(1)}, \ldots, K^{(N)}\right)$ is defined as: 2. When d is odd, the convolution-stride slicer $\mathcal{CS} - s\left(X \mid K^{(1)}, \ldots, K^{(N)}\right)$ takes the form:

$$\mathcal{CS} - s\left(X \mid K^{(1)}, \ldots, K^{(N)}\right) = \mathcal{CS} - s\left(X_*^{1,1} K^{(1)} \mid K^{(2)}, \ldots, K^{(N)}\right)$$

where $K^{(1)} \in \mathbb{R}^{c \times 2 \times 2}$ and $K^{(2)}, \ldots, K^{(N)}$ are the corresponding sliced kernels that are defined on the dimension $d - 1$.

The convolution-stride slicer reduces the width and the height of the image by half after each convolution operator. We use the same procedure of reducing the height and the width of the image by one when the height and the width of the image are odd. The benefit of the convolution-stride slicer is that the size of its kernels does not depend on the width and the height of images as that of the convolution-base slicer. This difference improves the computational complexity and time complexity of the convolution-stride slicer over those of the convolution-base slicer

3.3 Convolutional Dilatation Slicer

Given $X \in \mathbb{R}^{c \times d \times d} (d \geq 2)$, 1. When d is even, $N = [\log_2 d]$, sliced kernels are defined as $K^{(1)} \in \mathbb{R}^{c \times 2 \times 2}$ and $K^{(h)} \in \mathbb{R}^{1 \times 2 \times 2}$ for $h = 2, \ldots, N-1$, and $K^{(N)} \in \mathbb{R}^{1 \times a \times a}$ where $a = \frac{d}{2^{N-1}}$. Then, the convolution-dilation slicer $\mathcal{CS} - d\left(X \mid K^{(1)}, \ldots, K^{(N)}\right)$ is defined as:

$$\mathcal{CS} - d\left(X \mid K^{(1)}, \ldots, K^{(N)}\right) = X^{(N)}, \quad X^{(h)} = \begin{cases} X & h = 0 \\ X^{(h-1)}{}_*^{1,2} K^{(h)} & 1 \leq h \leq N-1 \\ X^{(h-1)}{}_*^{1,1} K^{(h)} & h = N, \end{cases}$$

2. When d is odd, the convolution-dilation slicer $\mathcal{CS} - d\left(X \mid K^{(1)}, \ldots, K^{(N)}\right)$ takes the form:

$$\mathcal{CS} - d\left(X \mid K^{(1)}, \ldots, K^{(N)}\right) = \mathcal{CS} - d\left(X_*^{1,1} K^{(1)} \mid K^{(2)}, \ldots, K^{(N)}\right)$$

where $K^{(1)} \in \mathbb{R}^{c \times 2 \times 2}$ and $K^{(2)}, \ldots, K^{(N)}$ are the corresponding sliced kernels that are defined on the dimension $d - 1$.

As with the previous slicers, the convolution-dilation slicer also reduces the width and the height of the image by half after each convolution operator and it uses the same procedure for the odd dimension cases. The design of kernels' size of the convolution-dilation slicer is the same as that of the convolution-stride slicer. However, the convolution-dilation slicer has a bigger receptive field in each convolution operator which might be appealing when the information of the image is presented by a big block of pixels.

4 Convolution Sliced Wasserstein

Definition. For any $p \geq 1$, the convolution sliced Wasserstein (CSW) of order $p > 0$ between two given probability measures $\mu, \nu \in \mathcal{P}_p\left(\mathbb{R}^{c\times d\times d}\right)$ is given by:

$$CSW_p(\mu,\nu) :=$$
$$\left(\mathbb{E}_{K^{(1)}\sim\mathcal{U}\left(\mathcal{K}^{(1)}\right),\ldots,K^{(N)}\sim\mathcal{U}\left(\mathcal{K}^{(N)}\right)}\left[W_p^p\left(\mathcal{S}\left(\cdot \mid K^{(1)},\ldots,K^{(N)}\right)\sharp\mu,\mathcal{S}\left(\cdot \mid K^{(1)},\ldots,K^{(N)}\right)\sharp\nu\nu\right)\right]\right)^{\frac{1}{p}}$$

where $\mathcal{S}\left(\cdot \mid K^{(1)},\ldots,K^{(N)}\right)$ is a convolution slicer with $K^{(i)} \in \mathbb{R}^{c^{(i)}\times k^{(i)}\times k^{(i)}}$ for any $i \in [N]$ and $\mathcal{U}\left(\mathcal{K}^{(i)}\right)$ is the uniform distribution with the realizations being in the set $\mathcal{K}^{(i)}$ which is defined as $\mathcal{K}^{(i)} := \left\{K^{(i)} \in \mathbb{R}^{c^{(i)}\times k^{(i)}\times k^{(i)}} \mid \sum_{h=1}^{c^{(i)}}\sum_{i'=1}^{k^{(i)}}\sum_{j'=1}^{k^{(i)}} K_{h,i',j'}^{(i)2} = 1\right\}$, namely, the set $\mathcal{K}^{(i)}$ consists of tensors $K^{(i)}$ whose squared ℓ_2 norm is 1 .

Similar to the conventional Sliced-Wasserstein the Convolution, the expectation drawn from the set $\mathcal{K}^{(1)},\ldots,\mathcal{K}^{(N)}$ and therefor is approximate using Monte Carlo method, which lead to the following :

$$\mathrm{CSW}_p(\mu,\nu) \approx \frac{1}{L}\sum_{i=1}^{L} W_p^p\left(\mathcal{S}\left(\cdot \mid K_i^{(1)},\ldots,K_i^{(N)}\right)\sharp\mu,\mathcal{S}\left(\cdot \mid K_i^{(1)},\ldots,K_i^{(N)}\right)\sharp\nu\right)$$

with a complexity of $\mathcal{O}\left(Lm\log_2 m\right)$.

- The convolution sliced Wasserstein, is symmetric, satisfies the triangle inequality, and $CSW_p(\mu,\nu) = 0 \Leftrightarrow \mu = \nu$
- $CSW_p(\mu,\nu) \leq \max -SW_p(\mu,\nu) \leq W_p(\mu,\nu)$

References

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient-Based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86 (11): 2278–2324. https://doi.org/10.1109/5.726791.

Nguyen, Khai, and Nhat Ho. 2022. "Revisiting Sliced Wasserstein on Images: From Vectorization to Convolution." arXiv.