How Well Do WGANs Estimate the Wasserstein Metric?

Machine Learning 2021 Course

Denis Rollov
Nikolay Goncharov
Svetlana Gabdullina

Skoltech

March 24, 2021

Introduction

What are the methods for evaluating the Wassertstein-1 distance and how good are they?

Ways for computing Wasserstein-1 distance:

1. Gradient Penalty (GP)
2. Weight Clipping (WC)
3. c-transform
4. (c, ϵ)-transform
5. Lipschitz Penalty (LP)

What is Wasserstein distance?

Wasserstein Distance is a measure of the distance between two probability distributions (Earth Mover's distance).

Experiments

- Datasets: MNIST, CIFAR-10.
- Models: MLP (two hidden layers (width=128), ReLU activation), CNN (DCGAN).
- Optimizers: Adam, RMSProp.

Weight Clipping

$$\max_{\omega} \left\{ \frac{1}{N} \sum_{i=1}^{N} \varphi_\omega(x_i) - \frac{1}{N} \sum_{i=1}^{N} \varphi_\omega(y_i) \right\}. \tag{1}$$

- learning rate: 5×10^{-5}
- optimizer: RMSprop
- MLP, $\epsilon = 0.05$ and $\epsilon = 0.08$ - MNIST and CIFAR10 respectively. CNN, $\epsilon = 0.03$ and $\epsilon = 0.2$ - MNIST and CIFAR10 respectively.

Gradient Penalty

$$\max_{\omega}\left\{\frac{1}{N}\sum_{i=1}^{N}\varphi_{\omega}\left(x_i\right)-\frac{1}{N}\sum_{i=1}^{N}\varphi_{\omega}\left(y_i\right)-\right.\\ \left.\frac{\lambda}{M}\sum_{i=1}^{M}\left(1-\|\nabla_{z=z_i}\varphi_{\omega}(z)\|\right)^2\right\}. \tag{2}$$

- beta values: $(0, 0.9)$
- MLP, learning rate: 5×10^{-3}. CNN, learning rate: 8×10^{-3} and 10^{-2} - MNIST and CIFAR-10 respectively.
- optimizer: Adam
- $\lambda = 10$

Lipschitz Penalty

$$\max_{\omega} \left\{ \frac{1}{N} \sum_{i=1}^{N} \varphi_\omega(x_i) - \frac{1}{N} \sum_{i=1}^{N} \varphi_\omega(y_i) - \frac{\lambda}{M} \sum_{i=1}^{M} \max\{0, \|\nabla_{z=z_i} \varphi_\omega(z)\| - 1\}^2 \right\}. \tag{3}$$

- beta values: $(0, 0.9)$
- MLP, learning rate: 5×10^{-3}. CNN, learning rate: 8×10^{-4} and 10^{-3} - MNIST and CIFAR-10 respectively.
- optimizer: Adam
- $\lambda = 10$

c-transform

$$\varphi_\omega^c(y_i) \approx \widehat{\varphi_\omega^c}(y_i) = \min_j \{c(x_j, y_i) - \varphi_\omega(x_j)\}, \tag{4}$$

$$\max_\omega \left\{ \frac{1}{N} \sum_{i=1}^{N} \varphi_\omega(x_i) + \frac{1}{N} \sum_{i=1}^{N} \widehat{\varphi_\omega^c}(y_i) \right\}. \tag{5}$$

- learning rate: 10^{-3}
- optimizer: RMSprop
- MLP, $\epsilon = 0.05$ and $\epsilon = 0.08$ - MNIST and CIFAR10 respectively. CNN, $\epsilon = 0.03$ and $\epsilon = 0.2$ - MNIST and CIFAR10 respectively.

(c,ϵ)-transform

$$
\begin{aligned}
&\varphi_\omega^c\left(y_i\right) \approx \widehat{\varphi_\omega^{(c,\epsilon)}}\left(y_i\right) = \\
&-\epsilon \log\left(\frac{1}{N}\sum_{j=1}^{N}\exp\left(-\frac{1}{\epsilon}\left(c\left(x_j, y_i\right) - \varphi_\omega\left(x_j\right)\right)\right)\right),
\end{aligned}
\tag{6}
$$

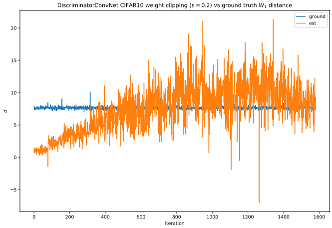
$$
\max_\omega \left\{\frac{1}{N}\sum_{i=1}^{N}\varphi_\omega\left(x_i\right) + \frac{1}{N}\sum_{j=1}^{N}\widehat{\varphi_\omega^{(c,\epsilon)}}\left(y_j\right)\right\}. \tag{7}
$$

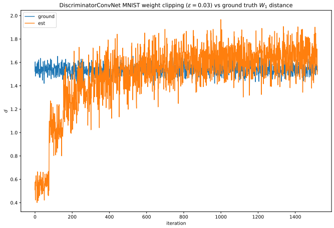
- learning rate: 10^{-4}
- optimizer: RMSprop
- MLP, CNN, $\epsilon = 12$ and $\epsilon = 1$ - MNIST and CIFAR10 respectively.

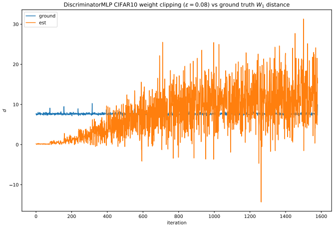
Skoltech
Skolkovo Institute of Science and Technology

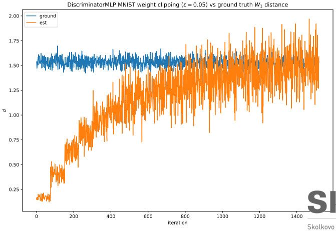
Weight clipping

CNN, MLP / CIFAR-10, MNIST



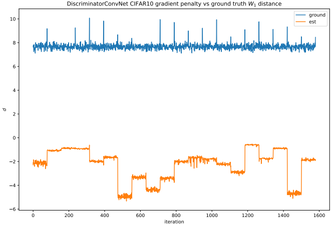


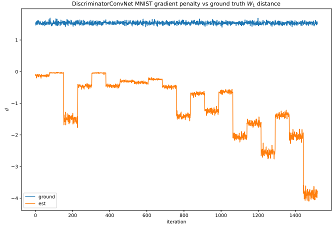


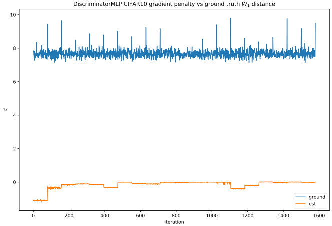


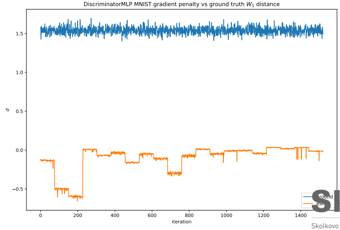
Gradient penalty

CNN, MLP / CIFAR-10, MNIST





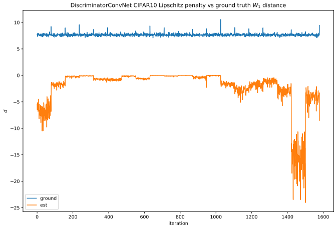


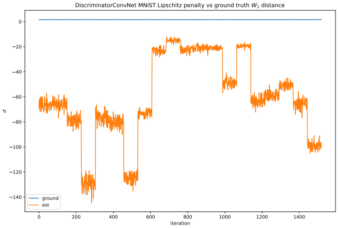


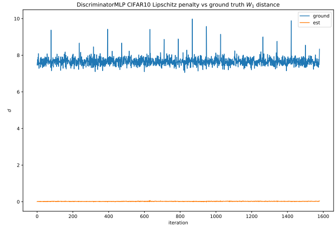
Skolkovo Institute of Science and Technology

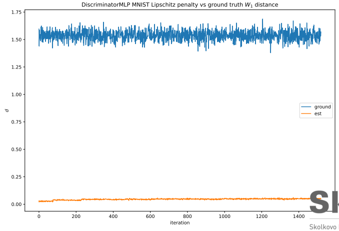
Lipschitz penalty

CNN, MLP / CIFAR-10, MNIST



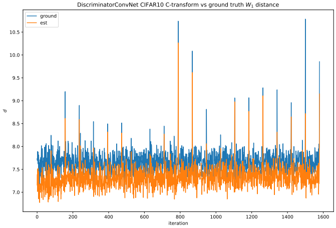


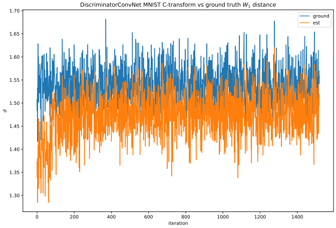


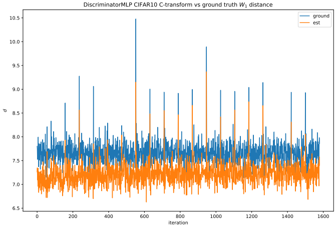


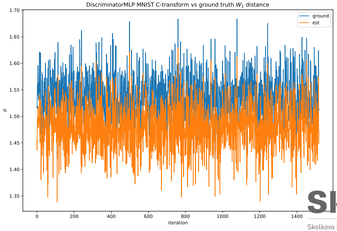
c-transform

CNN, MLP / CIFAR-10, MNIST



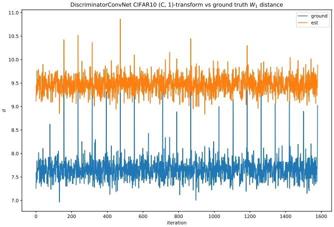


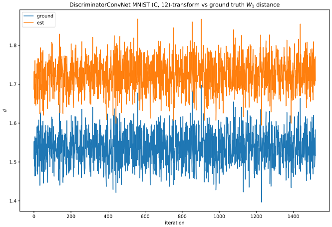


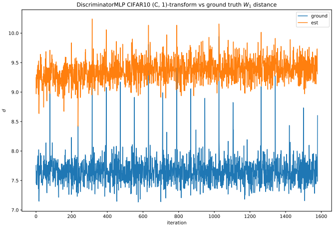


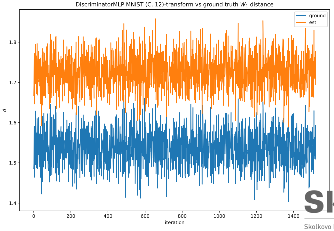
(c, ϵ)-transform

CNN, MLP / CIFAR-10, MNIST









Results

Approximation

MLP	MNIST	CIFAR-10
WC	0.407 ± 0.018	4.854 ± 0.157
GP	1.641 ± 0.009	7.819 ± 0.017
LP	1.491 ± 0.002	7.642 ± 0.011
c-transform	0.059 ± 0.001	0.448 ± 0.005
(c, ϵ)-transform	0.189 ± 0.001	1.696 ± 0.008

ConvNet	MNIST	CIFAR-10
WC	0.184 ± 0.011	3.056 ± 0.109
GP	2.612 ± 0.049	9.926 ± 0.063
LP	61.65 ± 1.659	9.761 ± 0.18
c-transform	0.065 ± 0.001	0.344 ± 0.006
(c, ϵ)-transform	0.186 ± 0.001	1.807 ± 0.007

Results

Stability

MNIST	WC	GP	LP	c-T	(c,ϵ)-T
$N=64, M=64$	0.08	0.19	0.01	1.45	1.69
$N=64, M=512$	0.06	-0.66	0.02	1.3	1.64
$N=512, M=64$	0.0	-0.38	0.01	1.44	1.65
$N=512, M=512$	0.0	-0.18	0.02	1.3	1.59
Ground truth	1.36	1.36	1.36	1.36	1.36
CIFAR-10	**WC**	**GP**	**LP**	**c-T**	**(c,ϵ)-T**
$N=64, M=64$	0.05	-0.74	0.02	7.21	9.22
$N=64, M=512$	0.04	-0.14	-0.04	6.49	9.19
$N=512, M=64$	0.0	-0.6	0.01	6.98	9.17
$N=512, M=512$	0.01	-2.9	0.01	6.3	9.13
Ground truth	6.89	6.89	6.89	6.89	6.89

Conclusion

	WC	GP	LP	c-transform	(c, ϵ)-transform
Approximation	$\checkmark$	$\times$	$\times$	$\checkmark$	$\checkmark$
Stability	$\times$	$\times$	$\times$	$\checkmark$	$\checkmark$

Further improvements: work in (c, ϵ)-transform with smaller values of ϵ without nan's.