EE-559 – Deep learning

10.2. Wasserstein GAN

François Fleuret

https://fleuret.org/ee559/

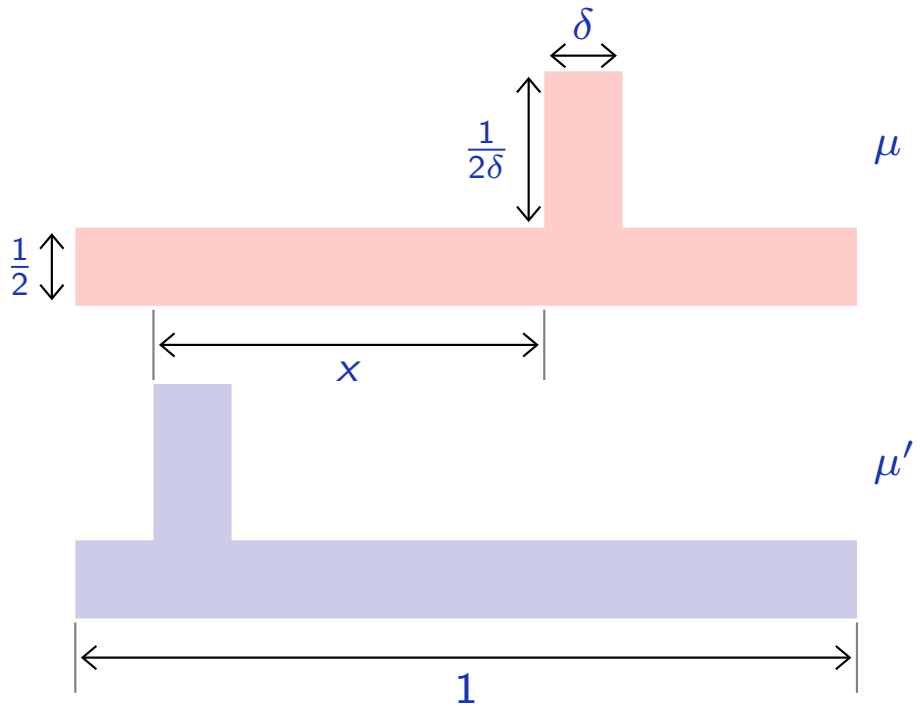
Wed Aug 29 14:57:14 UTC 2018

idiap RESEARCH INSTITUTE

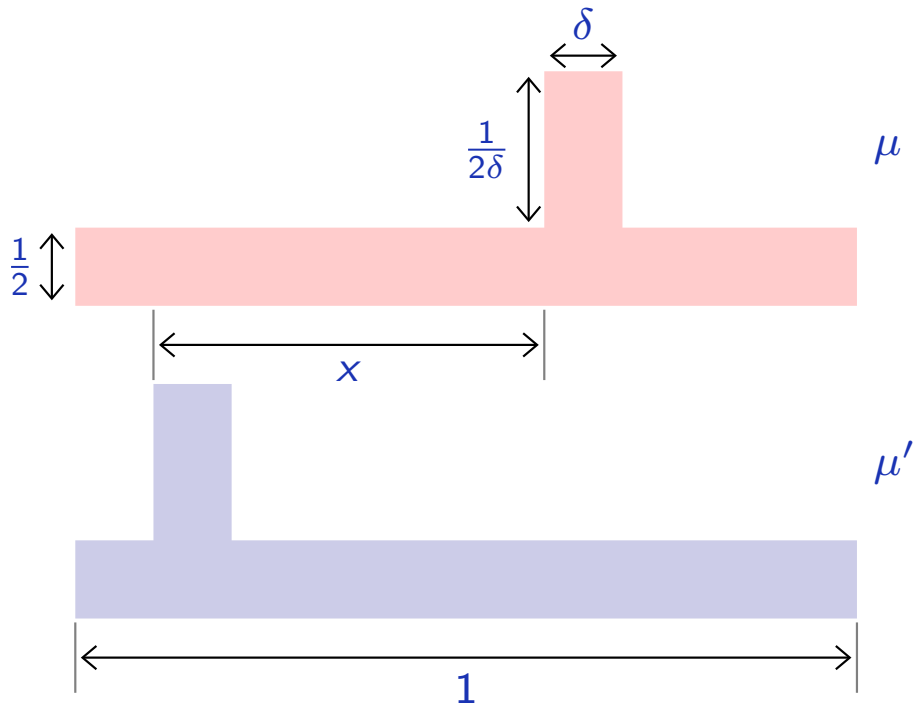
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Arjovsky et al. (2017) point out that $\mathbb{D}_{\mathrm{JS}}$ does not account [much] for the metric structure of the space.

Arjovsky et al. (2017) point out that $\mathbb{D}_{\mathrm{JS}}$ does not account [much] for the metric structure of the space.



Arjovsky et al. (2017) point out that $\mathbb{D}_{\text{JS}}$ does not account [much] for the metric structure of the space.

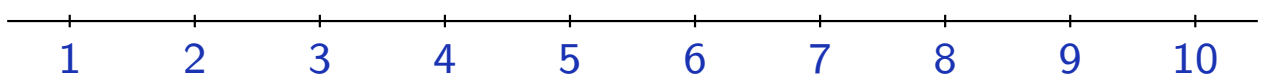


$$\mathbb{D}_{\text{JS}}(\mu, \mu') = \min(\delta, |x|) \left(\frac{1}{\delta} \log \left(1 + \frac{1}{2\delta} \right) - \left(1 + \frac{1}{\delta} \right) \log \left(1 + \frac{1}{\delta} \right) \right)$$

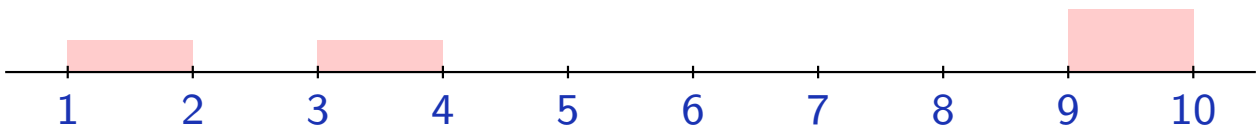
Hence all $|x|$ greater than δ are seen the same.

An alternative choice is the "earth moving distance", which intuitively is the minimum mass displacement to transform one distribution into the other.

An alternative choice is the "earth moving distance", which intuitively is the minimum mass displacement to transform one distribution into the other.

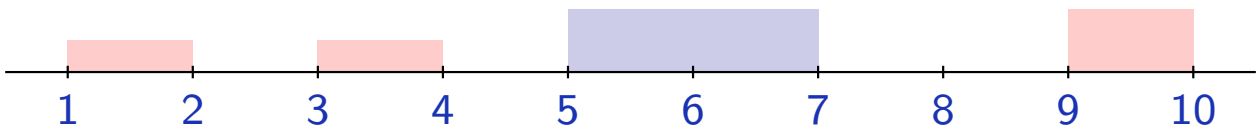


An alternative choice is the "earth moving distance", which intuitively is the minimum mass displacement to transform one distribution into the other.



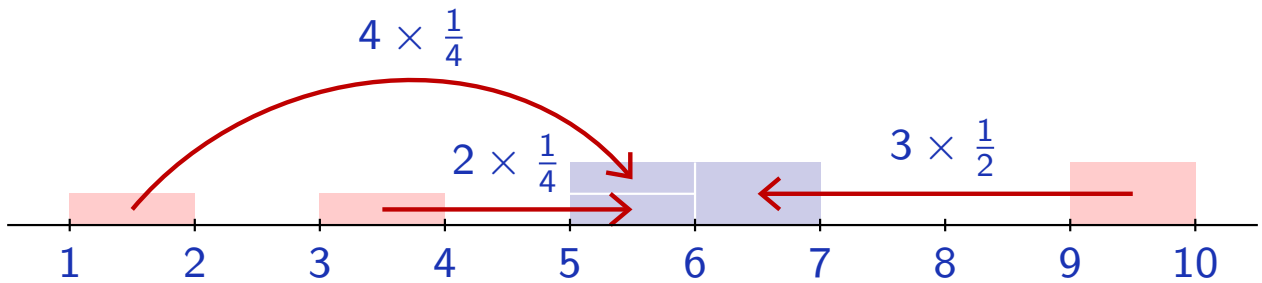
$$\mu = \frac{1}{4}\mathbf{1}_{[1,2]} + \frac{1}{4}\mathbf{1}_{[3,4]} + \frac{1}{2}\mathbf{1}_{[9,10]}$$

An alternative choice is the "earth moving distance", which intuitively is the minimum mass displacement to transform one distribution into the other.



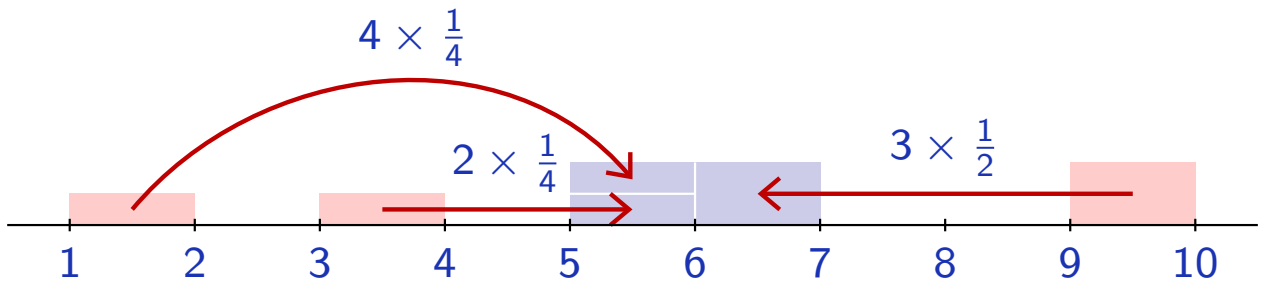
$$\mu = \frac{1}{4}\mathbf{1}_{[1,2]} + \frac{1}{4}\mathbf{1}_{[3,4]} + \frac{1}{2}\mathbf{1}_{[9,10]} \qquad \mu' = \frac{1}{2}\mathbf{1}_{[5,7]}$$

An alternative choice is the "earth moving distance", which intuitively is the minimum mass displacement to transform one distribution into the other.



$$\mu = \frac{1}{4}\mathbf{1}_{[1,2]} + \frac{1}{4}\mathbf{1}_{[3,4]} + \frac{1}{2}\mathbf{1}_{[9,10]} \qquad \mu' = \frac{1}{2}\mathbf{1}_{[5,7]}$$

An alternative choice is the "earth moving distance", which intuitively is the minimum mass displacement to transform one distribution into the other.



$$\mu = \frac{1}{4}\mathbf{1}_{[1,2]} + \frac{1}{4}\mathbf{1}_{[3,4]} + \frac{1}{2}\mathbf{1}_{[9,10]} \qquad \mu' = \frac{1}{2}\mathbf{1}_{[5,7]}$$

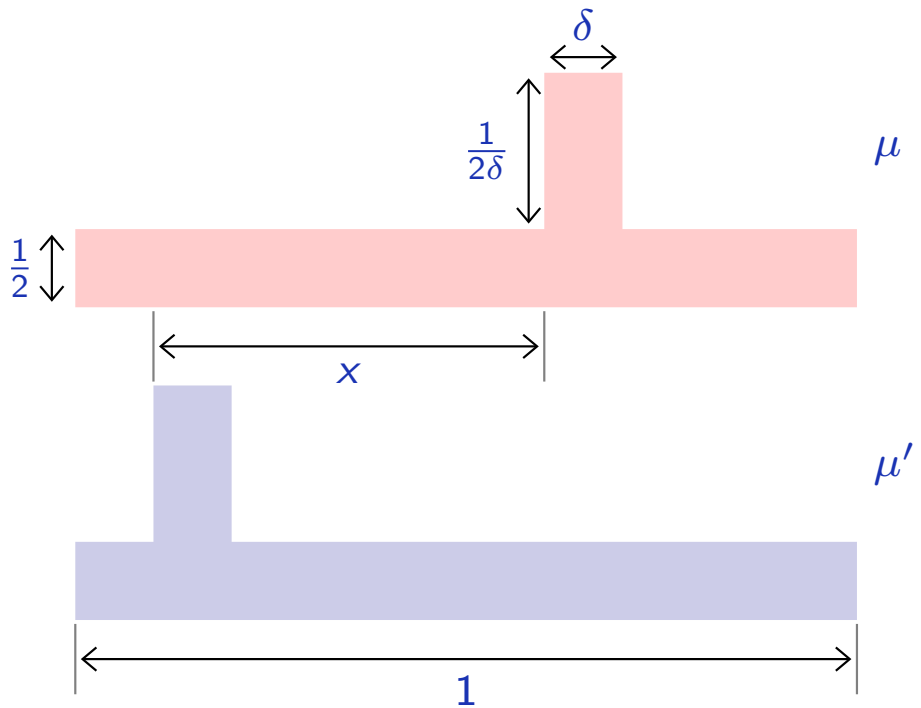
$$\mathbb{W}(\mu, \mu') = 4 \times \frac{1}{4} + 2 \times \frac{1}{4} + 3 \times \frac{1}{2} = 3$$

This distance is also known as the **Wasserstein** distance, defined as

$$\mathbb{W}(\mu, \mu') = \min_{q \in \Pi(\mu, \mu')} \mathbb{E}_{(X, X') \sim q} \Big[\|X - X'\| \Big],$$

where $\Pi(\mu, \mu')$ is the set of distributions over $\mathcal{X}^2$ whose marginals are μ and μ'.

Intuitively, it increases monotonically with the distance between modes



$$\mathbb{W}(\mu, \mu') = \frac{1}{2}|x|$$

So it would make a lot of sense to look for a generator matching the density for this metric, that is

$$\mathbf{G}^* = \operatorname*{argmin}_{\mathbf{G}} \mathbb{W}(\mu, \mu_{\mathbf{G}}).$$

So it would make a lot of sense to look for a generator matching the density for this metric, that is

$$\mathbf{G}^* = \operatorname*{argmin}_{\mathbf{G}} \mathbb{W}(\mu, \mu_{\mathbf{G}}).$$

Unfortunately, the definition of $\mathbb{W}$ does not provide an operational way of estimating it.

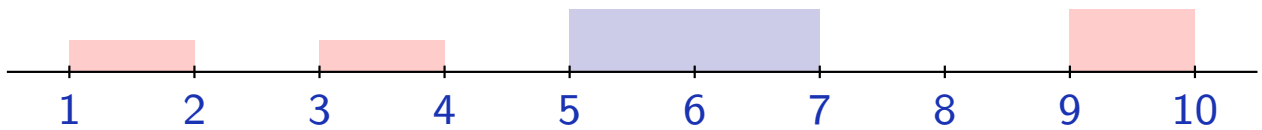
A duality theorem from Kantorovich and Rubinstein implies

$$\mathbb{W}(\mu, \mu') = \max_{\|f\|_L \leq 1} \mathbb{E}_{X \sim \mu}\Big[f(X)\Big] - \mathbb{E}_{X \sim \mu'}\Big[f(X)\Big]$$

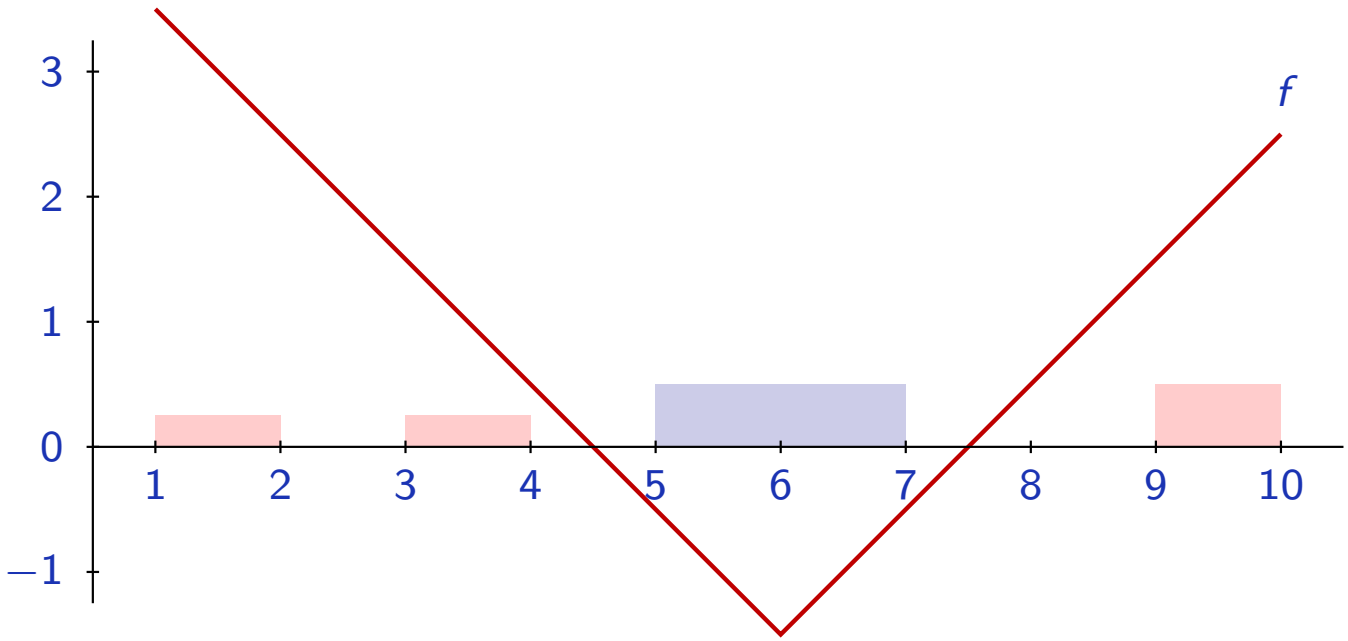
where

$$\|f\|_L = \max_{x,x'} \frac{\|f(x) - f(x')\|}{\|x - x'\|}$$

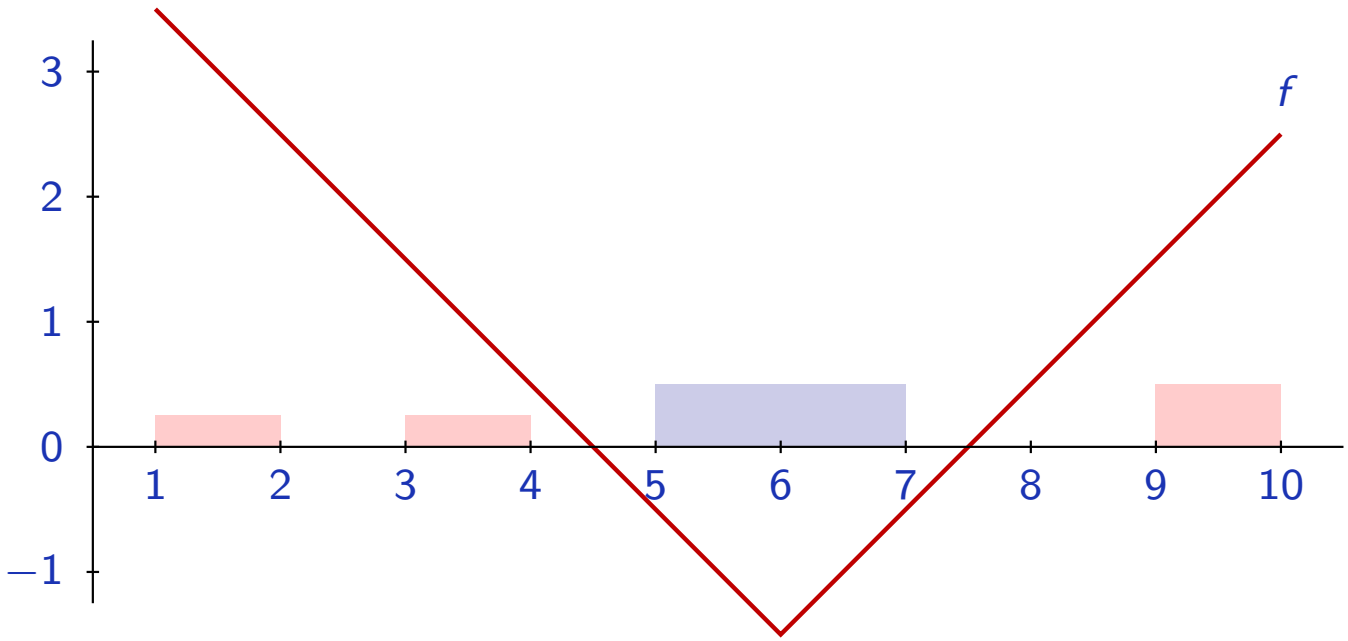
is the Lipschitz seminorm.



$$\mu = \frac{1}{4}\mathbf{1}_{[1,2]} + \frac{1}{4}\mathbf{1}_{[3,4]} + \frac{1}{2}\mathbf{1}_{[9,10]} \qquad \mu' = \frac{1}{2}\mathbf{1}_{[5,7]}$$



$$\mu = \frac{1}{4}\mathbf{1}_{[1,2]} + \frac{1}{4}\mathbf{1}_{[3,4]} + \frac{1}{2}\mathbf{1}_{[9,10]} \qquad \mu' = \frac{1}{2}\mathbf{1}_{[5,7]}$$



$$\mu = \frac{1}{4}\mathbf{1}_{[1,2]} + \frac{1}{4}\mathbf{1}_{[3,4]} + \frac{1}{2}\mathbf{1}_{[9,10]} \qquad \mu' = \frac{1}{2}\mathbf{1}_{[5,7]}$$

$$\mathbb{W}(\mu,\mu') = \underbrace{\left(3 \times \frac{1}{4} + 1 \times \frac{1}{4} + 2 \times \frac{1}{2}\right)}_{\mathbb{E}_{X\sim\mu} f(X)} - \underbrace{\left(-1 \times \frac{1}{2} - 1 \times \frac{1}{2}\right)}_{\mathbb{E}_{X\sim\mu'} f(X)} = 3$$

Using this result, we are looking for a generator

$$\begin{aligned}\mathbf{G}^* &= \operatorname*{argmin}_{\mathbf{G}} \mathbb{W}(\mu, \mu_{\mathbf{G}}) \\ &= \operatorname*{argmin}_{\mathbf{G}} \max_{\|\mathbf{D}\|_L \leq 1} \Big(\mathbb{E}_{X \sim \mu} \Big[\mathbf{D}(X) \Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}} \Big[\mathbf{D}(X) \Big] \Big),\end{aligned}$$

where the max is now an optimized predictor.

Using this result, we are looking for a generator

$$\begin{aligned}\mathbf{G}^* &= \operatorname*{argmin}_{\mathbf{G}} \mathbb{W}(\mu, \mu_{\mathbf{G}}) \\ &= \operatorname*{argmin}_{\mathbf{G}} \max_{\|\mathbf{D}\|_L \leq 1} \Big(\mathbb{E}_{X \sim \mu} \Big[\mathbf{D}(X) \Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}} \Big[\mathbf{D}(X) \Big] \Big),\end{aligned}$$

where the max is now an optimized predictor.

This is very similar to the original GAN formulation, except that the value of $\mathbf{D}$ is not interpreted through a log-loss, and there is a strong regularization on $\mathbf{D}$.

The main issue in this formulation is to optimize the network $\mathbf{D}$ under a constraint on its Lipschitz seminorm

$$\|\mathbf{D}\|_L \leq 1.$$

Arjovsky et al. achieve this by clipping $\mathbf{D}$'s weights.

The two main benefits observed by Arjovsky et al. are

- A greater stability of the learning process, both in principle and in their experiments: they do not witness "mode collapse".
- A greater interpretability of the loss, which is a better indicator of the quality of the samples.

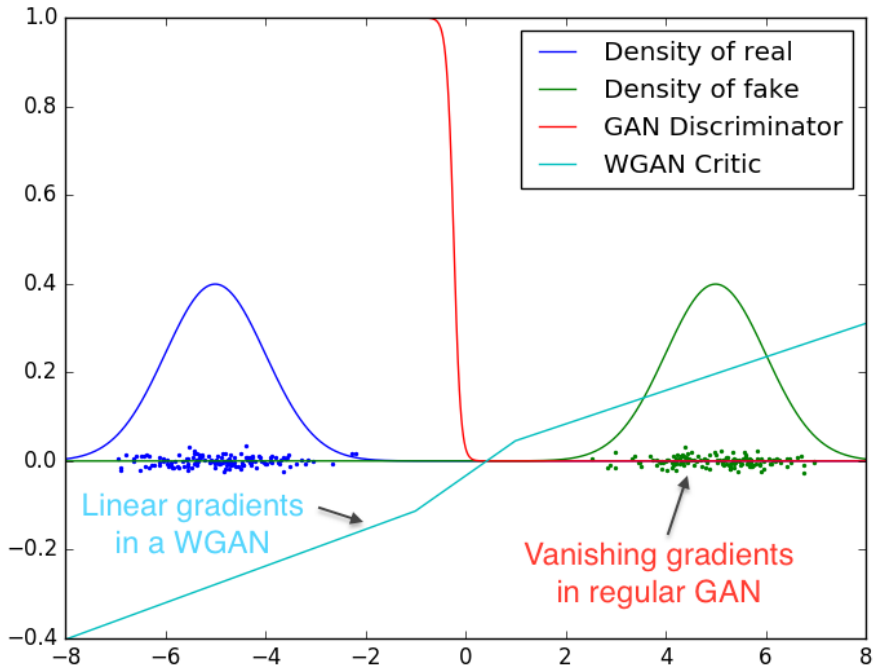


Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the traditional GAN discriminator saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

(Arjovsky et al., 2017)

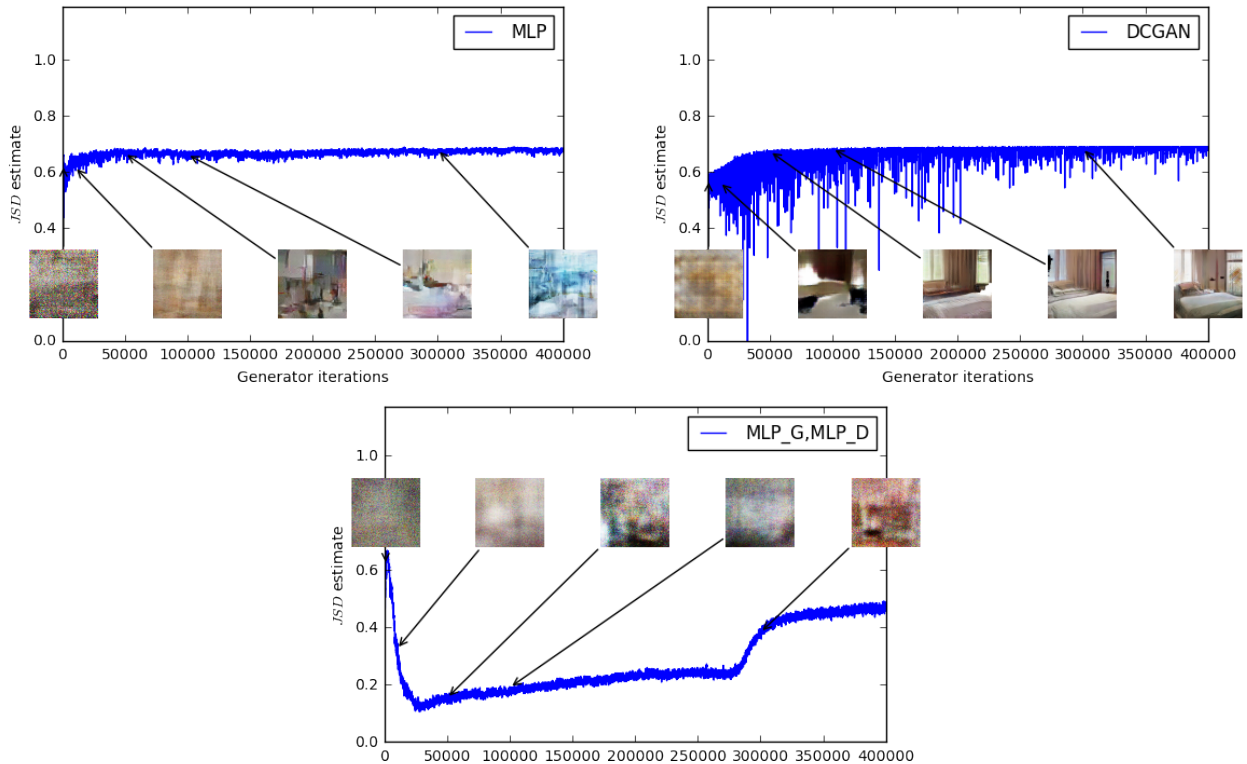


Figure 4: JS estimates for an MLP generator (upper left) and a DCGAN generator (upper right) trained with the standard GAN procedure. Both had a DCGAN discriminator. Both curves have increasing error. *Samples get better for the DCGAN but the JS estimate increases or stays constant, pointing towards no significant correlation between sample quality and loss. Bottom: MLP with both generator and discriminator. The curve goes up and down regardless of sample quality. All training curves were passed through the same median filter as in Figure 3.*

(Arjovsky et al., 2017)

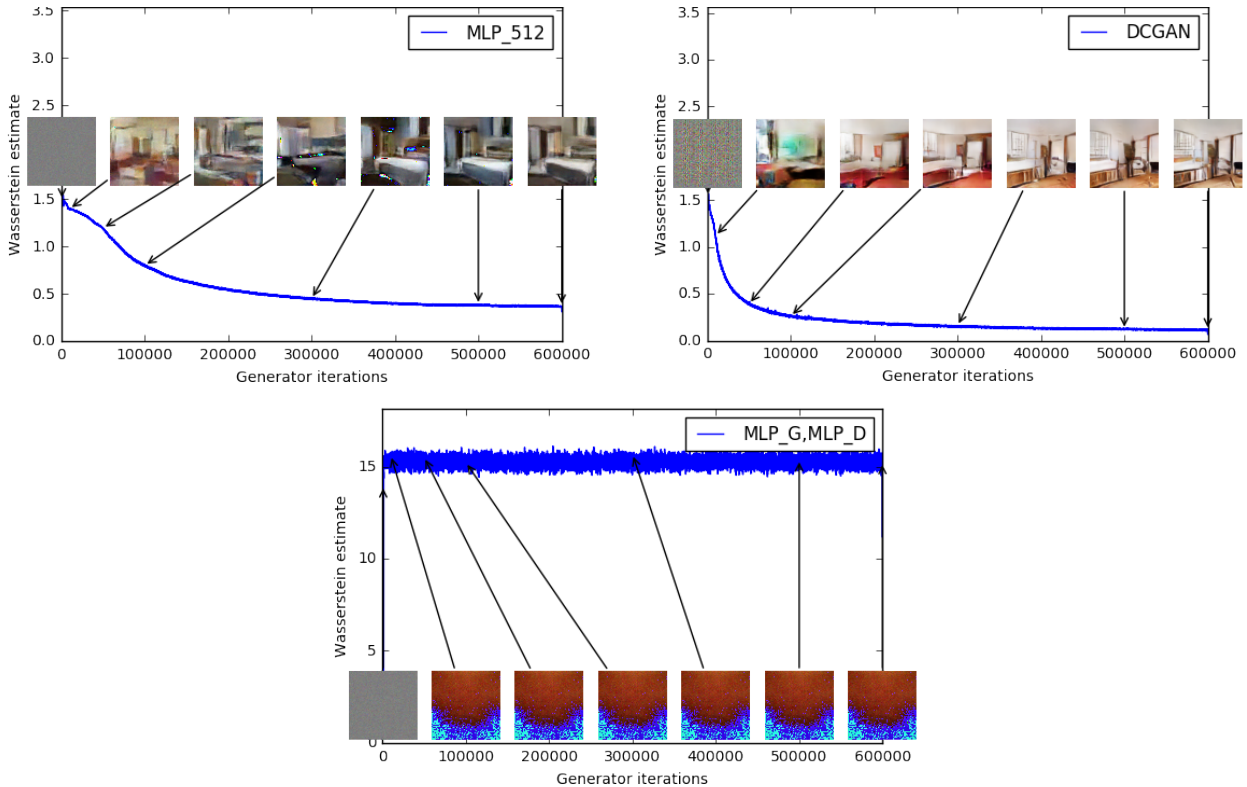


Figure 3: Training curves and samples at different stages of training. We can see a clear correlation between lower error and better sample quality. Upper left: the generator is an MLP with 4 hidden layers and 512 units at each layer. The loss decreases constistently as training progresses and sample quality increases. Upper right: the generator is a standard DCGAN. The loss decreases quickly and sample quality increases as well. In both upper plots the critic is a DCGAN without the sigmoid so losses can be subjected to comparison. Lower half: both the generator and the discriminator are MLPs with substantially high learning rates (so training failed). Loss is constant and samples are constant as well. The training curves were passed through a median filter for visualization purposes.

(Arjovsky et al., 2017)

However, as Arjovsky et al. wrote:

> "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint. If the clipping parameter is large, then it can take a long time for any weights to reach their limit, thereby making it harder to train the critic till optimality. If the clipping is small, this can easily lead to vanishing gradients when the number of layers is big, or batch normalization is not used (such as in RNNs)."
>
> (Arjovsky et al., 2017)

However, as Arjovsky et al. wrote:

> "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint. If the clipping parameter is large, then it can take a long time for any weights to reach their limit, thereby making it harder to train the critic till optimality. If the clipping is small, this can easily lead to vanishing gradients when the number of layers is big, or batch normalization is not used (such as in RNNs)."
>
> (Arjovsky et al., 2017)

In some way, the resulting Wasserstein GAN (WGAN) trades the difficulty to train **G** for the difficulty to train **D**.

However, as Arjovsky et al. wrote:

> "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint. If the clipping parameter is large, then it can take a long time for any weights to reach their limit, thereby making it harder to train the critic till optimality. If the clipping is small, this can easily lead to vanishing gradients when the number of layers is big, or batch normalization is not used (such as in RNNs)."
>
> (Arjovsky et al., 2017)

In some way, the resulting Wasserstein GAN (WGAN) trades the difficulty to train **G** for the difficulty to train **D**.

In practice, this weakness results in extremely long convergence time.

Gulrajani et al. (2017) proposed the **improved Wasserstein GAN** in which the constraint on the Lipschitz seminorm is replaced with a smooth penalty term.

Gulrajani et al. (2017) proposed the **improved Wasserstein GAN** in which the constraint on the Lipschitz seminorm is replaced with a smooth penalty term.

They state that if

$$\mathbf{D}^* = \operatorname*{argmax}_{\|\mathbf{D}\|_L \leq 1} \left(\mathbb{E}_{X \sim \mu} \left[\mathbf{D}(X) \right] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}} \left[\mathbf{D}(X) \right] \right)$$

then, with probability one under μ and $\mu_{\mathbf{G}}$

$$\|\nabla \mathbf{D}^*(X)\| = 1.$$

Gulrajani et al. (2017) proposed the **improved Wasserstein GAN** in which the constraint on the Lipschitz seminorm is replaced with a smooth penalty term.

They state that if

$$\mathbf{D}^* = \operatorname*{argmax}_{\|\mathbf{D}\|_L \leq 1} \left(\mathbb{E}_{X \sim \mu} \Big[\mathbf{D}(X) \Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}} \Big[\mathbf{D}(X) \Big] \right)$$

then, with probability one under μ and $\mu_{\mathbf{G}}$

$$\|\nabla \mathbf{D}^*(X)\| = 1.$$

This implies that adding a regularization that pushes the gradient norm to one should not exclude [any of] the optimal discriminator[s].

So instead of looking for

$$\underset{\|\mathbf{D}\|_L \leq 1}{\operatorname{argmax}} \; \mathbb{E}_{X \sim \mu}\Big[\mathbf{D}(X)\Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}}\Big[\mathbf{D}(X)\Big],$$

So instead of looking for

$$\underset{\|\mathbf{D}\|_L \leq 1}{\operatorname{argmax}} \; \mathbb{E}_{X \sim \mu}\Big[\mathbf{D}(X)\Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}}\Big[\mathbf{D}(X)\Big],$$

Gulrajani et al. propose to solve

$$\underset{\mathbf{D}}{\operatorname{argmax}} \; \mathbb{E}_{X \sim \mu}\Big[\mathbf{D}(X)\Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}}\Big[\mathbf{D}(X)\Big] - \lambda \mathbb{E}_{X \sim \mu_p}\Big[\left(\|\nabla \mathbf{D}(X)\| - 1\right)^2\Big]$$

where μ_p is the distribution of a point B sampled uniformly between a real sample X and a fake sample $\mathbf{G}(Z)$, that is $B = UX + (1-U)X'$ where $X \sim \mu$, $X' \sim \mu_{\mathbf{G}}$, and $U \sim \mathscr{U}[0,1]$.

So instead of looking for

$$\underset{\|\mathbf{D}\|_L \leq 1}{\text{argmax}} \; \mathbb{E}_{X \sim \mu}\Big[\mathbf{D}(X)\Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}}\Big[\mathbf{D}(X)\Big],$$

Gulrajani et al. propose to solve

$$\underset{\mathbf{D}}{\text{argmax}} \; \mathbb{E}_{X \sim \mu}\Big[\mathbf{D}(X)\Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}}\Big[\mathbf{D}(X)\Big] - \lambda \mathbb{E}_{X \sim \mu_p}\Big[\left(\|\nabla \mathbf{D}(X)\| - 1\right)^2\Big]$$

where μ_p is the distribution of a point B sampled uniformly between a real sample X and a fake sample $\mathbf{G}(Z)$, that is $B = UX + (1-U)X'$ where $X \sim \mu$, $X' \sim \mu_{\mathbf{G}}$, and $U \sim \mathcal{U}[0,1]$.

Note that this loss involves second-order derivatives.

So instead of looking for

$$\underset{\|\mathbf{D}\|_L \leq 1}{\text{argmax}} \; \mathbb{E}_{X \sim \mu}\Big[\mathbf{D}(X)\Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}}\Big[\mathbf{D}(X)\Big],$$

Gulrajani et al. propose to solve

$$\underset{\mathbf{D}}{\text{argmax}} \; \mathbb{E}_{X \sim \mu}\Big[\mathbf{D}(X)\Big] - \mathbb{E}_{X \sim \mu_{\mathbf{G}}}\Big[\mathbf{D}(X)\Big] - \lambda \mathbb{E}_{X \sim \mu_p}\Big[\,(\|\nabla \mathbf{D}(X)\| - 1)^2\,\Big]$$

where μ_p is the distribution of a point B sampled uniformly between a real sample X and a fake sample $\mathbf{G}(Z)$, that is $B = UX + (1-U)X'$ where $X \sim \mu$, $X' \sim \mu_{\mathbf{G}}$, and $U \sim \mathscr{U}[0,1]$.

Note that this loss involves second-order derivatives.

Experiments show that this scheme is more stable than WGAN under many different conditions.

The end

References

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *CoRR*, abs/1701.07875, 2017.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.