Basics of Wasserstein GAN

June 28, 2022

Useful papers - [ACB17]

1 Introduction

This paper discusses about Wasserstein metric in the problem of the data distribution approximation. Let P_Θ - model distribution, P_{data} - real distribution, our aim is to find Θ^* such that $P_{\Theta^*} \approx P_{data}$. It is possible to highlight two most common approaches:

$$\begin{aligned} &1)\ \mathbb{KL}(P_{data}||P_\Theta) = \int dX\, P_{data}(X) \log \frac{P_{data}(X)}{P_\Theta(X)} \to \max_\Theta \\ &2)\ \mathbb{E}_{P_{data}(x)} \log D(x) + \mathbb{E}_{P(z)} \log(1 - D(G(z))) \to \min_G \max_D \end{aligned} \tag{1}$$

The first one is VAE approach and the second one is GAN. However, both approaches suffer from same problem (we highlight only one, however it is possible to consider many other disadvantages). The real data (in case of images) lies in manifold much smaller dimension than dimension of images. Thus, the supports of $P_{data}(X)$ and $P_\Theta(X)$ practically do not intersect. Let's show this with an example (Figure 1). We

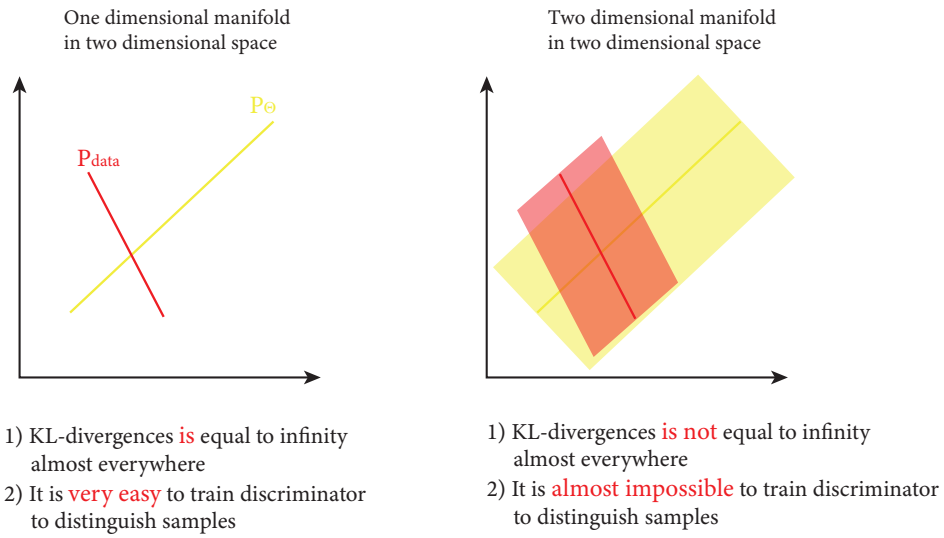

Figure 1: Manifold example.

show an example of one dimensional manifold in two dimensional space (left) and two dimensional manifold (right). As can be seen from the left figure, it is unlikely that P_{data} and P_Θ are intersected. Because of that we have a problem with both approaches. 1) KL divergence is equal to ∞ almost everywhere (because $P_\Theta = 0$ and $P_{data} \neq 0$, while integrating by P_{data} support); 2) Discriminator trains very easy and $D(X) = 1, D(G(Z)) = 0$, is easy to distinguish samples. So, the gradients of D is zero (as can be seen from Figure 2), thus the generator does not train. One approach to

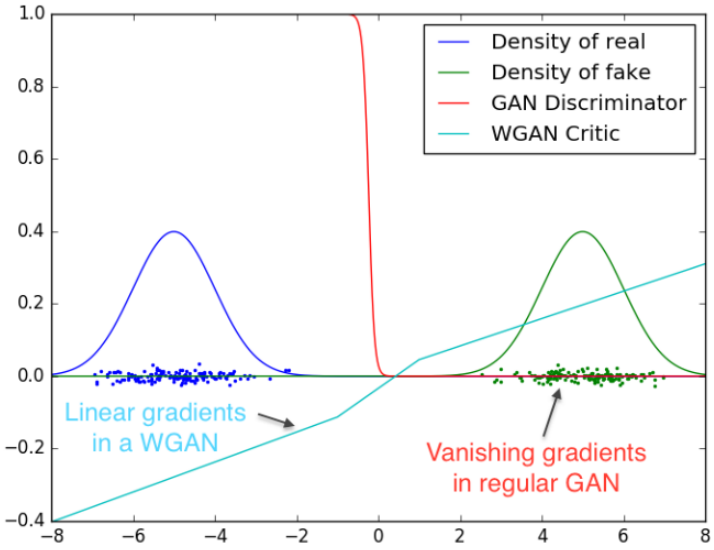

Figure 2: Gradient of the disrciminator

fix it: add noise to data. It can do manifold more widely, i.e. more closer to the right figure. The authors of the paper are trying to find more suitable distance instead of KL-divergence and Jenses-Shannon divergence (it is GAN).

The authors consider the Wasserstein distance:

$$\mathbb{W}(P, Q) = \inf_{\gamma \in \mathcal{G}(P,Q)} \mathbb{E}_{(x,y)\sim\mathcal{G}} \, ||x - y|| = \inf_{\gamma \in \mathcal{G}(P,Q)} \int \gamma(x, y)||x - y||dxdy \tag{2}$$

Looks scary. Lets consider in more detail. Here γ - transport plan, i.e. how much probability mass should be transferred from point x to point y. The Wasserstein distance shows how much effort does it take to make another distribution from one, that is, the amount of mass per distance. Note that we can do it in different ways (because we have different transport plans $\mathcal{G}(P, Q)$). Also, we have the following normalizing rules:

$$\begin{gathered} \int \gamma(x, y)dx = Q(y) - \text{if we did plan for } x \\ \text{then we have to obtain distribution } Q \text{ in } y \text{ point.} \\ \int \gamma(x, y)dy = P(x) - \text{if we did plan for } y \\ \text{then we have to obtain distribution } P \text{ in } x \text{ point.} \end{gathered} \tag{3}$$

Consider a simple discrere example (Figure 1). Here we have two discrete distribution

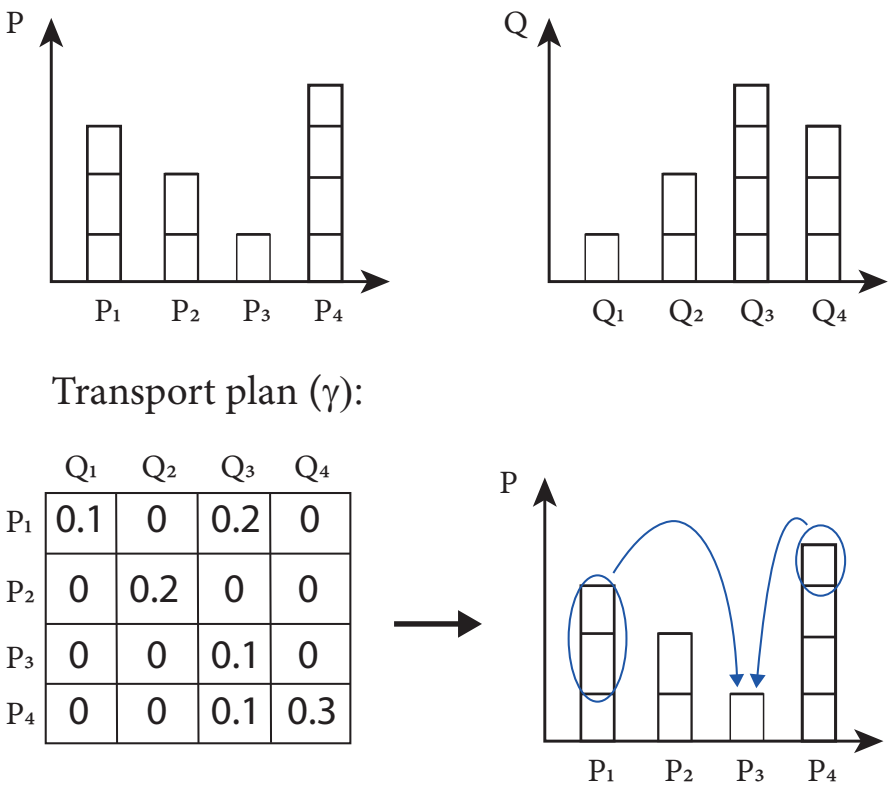

Figure 3: Simple disrcete example

and moves P towards Q. Transport plan is presented in table. Lets calculate the Wasserstein metric using this plan and assuming that distance between P_i and Q_i is equal to zero.

$$\mathbb{W}(P,Q) = \sum_x \sum_y \gamma(x,y)D(x,y) = 0.1*0 + 0.2*2 + 0.2*0 + 0.1*0 + 0.1*1 + 0.3*0 = 0.5 \tag{4}$$

Lets have a look to another example illustrating advantages of the Wasserstein distances compared to other metrics in case of disjoint supports 1. We have $P(x,y) = (0, \mathbb{U}(0,1)), Q(x,y) = (\Theta, \mathbb{U}(0,1))$.

$$1)\ \mathbb{KL}(P||Q) = \sum_{x=0}^{0} \int dy P(X=0,y) \log \frac{P(X=0,y)}{Q(X=0,y)} = \int dy 1 \log \frac{1}{0} = \infty$$

$$2)\ \mathbb{KL}(Q||P) = \sum_{x=\Theta}^{\Theta} \int dy Q(X=\Theta,y) \log \frac{Q(X=\Theta,y)}{P(X=\Theta,y)} = \int dy 1 \log \frac{1}{0} = \infty$$

$$3)\mathbb{JSD}(P||Q) = \frac{1}{2}\mathbb{KL}(P||\frac{P+Q}{2}) + \frac{1}{2}\mathbb{KL}(Q||\frac{P+Q}{2}) = \log 2$$

$$\frac{1}{2}\mathbb{KL}(P||\frac{P+Q}{2}) = \frac{1}{2}\sum_{x=0}^{0} \int dy P(X=0,y) \log \frac{P(X=0,y)}{\frac{P(X=0,y)+Q(X=0,y)}{2}} = \frac{1}{2}\int_0^1 dy 1 \log 2 = \frac{1}{2}\log 2$$

$$4)W(P||Q) = |\Theta| \tag{5}$$

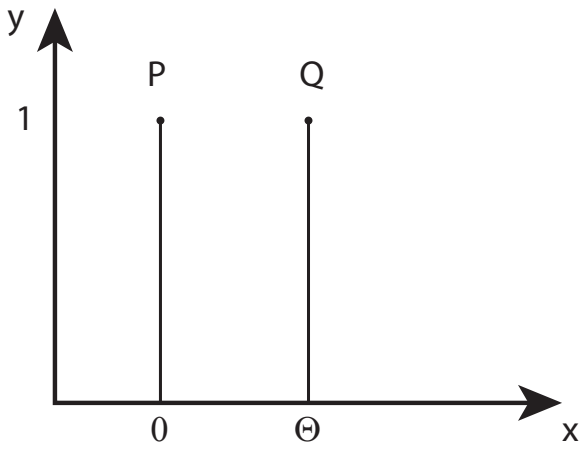

Figure 4: Simple example of distributions with disjoint supports

As can be seen, we have problems with all distances (as a consequence problems with gradients) with the exception of the Wasserstein metric.

However, it is unclear how to calculate Wasserstein metric on practice, we have some infinum over transport plans... But we can use Kantorovich-Rubinstein duality

$$W(P||Q) = \frac{1}{K} \max_{||f||_L \leq K} \left[\mathbb{E}_{P(X)} f(X) - \mathbb{E}_{Q(X)} f(X)\right] \tag{6}$$

where $f : \mathcal{X} \to \mathbb{R}$, we assume that this function is k-lipschitz, i.e. $|f(X_1) - f(X_2)| \leq K||X_1 - X_2||, \forall X_1, X_2 \in \mathcal{X}; K-$constant. In other words, we need that first derivative of f. So, it is look much better, however we have a maximum over k-lipschitz function. As always if we have some unknown function let it be a neural network, i.e. $f : \mathcal{X} \times \Theta \to \mathbb{R}$ (it calls critic in the paper). Also, we need a k-lipschitz from neural network. It is known that neural network is composition of function:

$$f_\Theta(X) = \sigma(W_n \sigma(W_{n-1} ... \sigma(W_0 X))) \tag{7}$$

where $\Theta = (W_n, ..., W_0)$, σ - activation function. It is known that composition of lipschitz functions is also lipschitz function. Thus, we have to consider the lipschitz property only for $\sigma(W_0 X)$ (single operation). Let's do it intuitively (I can't do it any other way:( ). If Θ lies in a compact set (a closed and bounded set, whatever that means), for example, $\Theta \in [1, 1]^m$, after the single operation we remain in a closed set, i.e. $\sigma(W_0 X)$ also lies in compact set. It is clear from the following facts: in cases of neural networks $\mathcal{X}$ is a compact because we always normalize our data (for example to $[0, 1]$); $W-$ compact by definition; function σ does not violate the compactness property (take ReLU, for example, and think about it). So, if $\sigma(W_0 X)$ is a compact, then it is limited, i.e. $||\sigma(W_0 X)|| \leq C$, then $||\sigma(W_0 X_1) - \sigma(W_0 X_0)|| \leq ||\sigma(W_0 X_1)|| + ||\sigma(W_0 X_0)|| = C_1 + C_2 = C$. We can product both parts by $||X_1 - X_0||$, that is, $||\sigma(W_0 X_1) - \sigma(W_0 X_0)|| \cdot ||X_1 - X_0|| \leq C||X_1 - X_0||$. And because $\mathcal{X}$ is a compact set $||X_1 - X_0|| \leq C_2$, so we can choose C in a way that $||\sigma(W_0 X_1) - \sigma(W_0 X_0)|| \cdot C_2 \leq C||X_1 - X_0||$. I think it is a shit proof, but i have tried.

Finally, if the weights of neural network lies in a compact set, then neural network satisfies the k-lipschitz property. In the paper authors do it by clipping of parameters

to some interval ($[0.01, 0.01]^d$). Everything is prepared for training the WGAN. We have two stages training procedure:

1. First of all, we have to approximate the Wasserstein distance (6). In other words, as can be seen from the formula, we have to find f that provides maximum of $\left[\mathbb{E}_{P(X)} f(X) - \mathbb{E}_{Q(X)} f(X)\right]$. So, we need high accuracy approximation before we minimize this metric. Thus, we will do several steps of gradient descent:

$$\Theta_{i+1} = \Theta_i + \nabla_\Theta \left[\mathbb{E}_{P_{data}(X)} f_\Theta(X) - \mathbb{E}_{P(Z)} f_\Theta(g_\Phi(Z))\right] \tag{8}$$

   Here we have moved to GAN notation, i.e. $P_{data}-$ distribution of data, $P(Z) = \mathbb{N}(0, I)-$distribution of latent variables, $g_\Phi-$ generator (also neural network). We use Monte-Carlo approximation of the expectations (to do stochastic optimization)

$$\Theta_{i+1} = \Theta_i + \nabla_\Theta \left[\frac{1}{n}\sum_{i=1}^{n} f_\Theta(x_i) - \frac{1}{n}\sum_{i=1}^{n} f_\Theta(g_\Phi(z_i))\right] \tag{9}$$

2. After the Wasserstein distance approximation, we are ready to minimize it itself, i.e. approximate the real data distribution.

$$\Phi_{i+1} = \Phi_i + \nabla_\Phi \frac{1}{n}\sum_{i=1}^{n} f_\Theta(g_\Phi(z_i)) \tag{10}$$

As can be seen, it is very similar (I would say completely same) to the classical GAN with the exception of critic instead of discriminator. So, why all this theory, if we could just replace the classifier in GAN (discriminator) with an arbitrary neural network (critic)? :). It is a joke. Below the algorithm from the original paper is provided.

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.
while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^m \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^m f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

Figure 5: WGAN algorithm

Lets summarize the most important points. In this paper authors propose an approach that replace the discriminator in GAN to an arbitrary neural network, i.e. critic. Such move help to solve the main problem of the GAN: we can not train discriminator til optimally!!! (although the gan theory asks for it). This is due to the fact that the

data (images) lies in manifold much lower dimension, thus the supports of the distribution practically do not intersects. So, it is very easy to build a hyperplane between them and as a consequence distinguish samples. In this cases gradients of the discriminator is vanished and it is became impossible to train generator. The critic does not suffer from this problem (as can be seen from Figure 2) and it is very nice.

References

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.