Optimal Transport GAN

Thi Hai Yen VU
haiyen96.hp@gmail.com
Master Data Science

Abstract

This report introduces OT-GAN [1], a variant of Generative Adversarial Networks (GAN) based on a metrics combining an optimal transport distance and an energy distance defined on the adversarial feature space. This new approach yields more stable training and achieves state-of-the-art results in image generation (at the time of its publication).

1 Introduction

Generative Adversarial Networks (GAN) [2], since its appearance, have gained a lot of attentions from the machine learning community, particularly in the field of computer vision as it enables to generate very realistic images by learning from a given dataset. The idea of GAN is simple yet brilliant. It dually trains two networks whose objectives are opposite. One network, *discriminator*, takes a given image as input and must tries to tells whether it is *real* or *fake*. The other one, called *generator*, tries to generate fake images which are as real as possible so as to fool the discriminator. The well trained GAN model is then able to produce generally good results.

Later, the GAN framework is re-interpreted in terms of Optimal Transport (OT) theory [3] using the *Wassertein-1 distance*. Following this work, many other work then started to investigate and exploit the relationships between GAN and OT. These works mainly rely on using different OT distances for training the GAN. This paper [1] is also based on this idea and introduces a so-called *mini-batch energy distance* which is claimed to obtain statistically consistent objective and unbiased gradients. In addition, it is also claimed to have a stronger discriminative power and a more stable generative modeling.

The next section will introduce briefly the relation between GAN and OT, then the theoretical formulations of the model OT-GAN. Section 3 presents my experimental results of the OT-GAN model for a simple MNIST dataset.

2 Theoretical formulations

2.1 GAN and Optimal Transport

As discussed previously, the original GAN model makes use of a generator g and a discriminator d to train a generative model by solving the following min-max problem:

$$L = \inf_g \sup_d \mathbb{E}_{\mathbf{x}\sim p} \log[d(\mathbf{x})] + \mathbb{E}_{\mathbf{y}\sim g} \log[1 - d(\mathbf{y})] \tag{1}$$

where $\mathbf{x}$ denotes a real image drawn from the distribution of training data $\mathbf{x} \sim p(\mathbf{x})$, $\mathbf{y}$ is a fake image generated by the generator $\mathbf{y} = g(\mathbf{z})$ where $\mathbf{z}$ is a noise drawn from some prior $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ ($p_{\mathbf{z}}$ can be $\mathcal{N}(0, 1)$ for example). Here we write $\mathbf{y} \sim g(\mathbf{y})$ instead of $\mathbf{y} = g(\mathbf{z}), \mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$. The maximization of d ensures that the discriminator distinguishes well the real and fake images, while the minimization of g encourages the generator to produce images as real as possible.

From an optimal transport point of view, the generative modeling can be seen as a distribution approximation problem, in which we try to minimize *certain distance* between the two distributions $p(\mathbf{x})$ and $g(\mathbf{y})$. These distances can be defined in several ways using optimal transport theory.

The paper [3] proposes using the classic Wassertein-1 distance, or Earth-Mover distance, as follows:

$$D_{EMD}(p, g) = \inf_{\gamma \in \Pi(p,g)} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\gamma} c(\mathbf{x}, \mathbf{y}) \tag{2}$$

where $\Pi(p, q)$is the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ with marginals $p(\mathbf{x}), g(\mathbf{y})$, and $c(\mathbf{x}, \mathbf{y})$ is a cost function which is taken to be the Euclidean distance. Generally, D_{EMD} is a metric with appropriate choice of the cost function c. However the equation defined in (2) is impossible to compute (intractable), hence it is formulated under the dual form:

$$D_{EMD}(p, g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{\mathbf{x}\sim p} f(\mathbf{x}) - \mathbb{E}_{\mathbf{y}\sim g} f(\mathbf{y}), \tag{3}$$

This problem is in general still intractable but can be approximated by using the class of the discriminator d as described above. Using this class of GAN discriminators, equation (3) becomes very similar to the original equation (1) of GAN.

Later, [4] proposes using a generalized version of the Wassertein-1 distance, known as *Sinkhorn distance* [5], instead of the EMD:

$$D_{Sinkhorn}(p, g) = \inf_{\gamma \in \Pi(p,g)} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\gamma} c(\mathbf{x}, \mathbf{y}) - \epsilon \mathbf{E}(\gamma) \tag{4}$$

where $\mathbf{E}(\gamma)$ is the entropic regularization of the distribution γ. This distance can then be approximated on mini-batches of data $\mathbf{X}, \mathbf{Y}$ using Sinkhorn algorithm given in [5], and is denoted $\mathcal{W}_c(\mathbf{X}, \mathbf{Y})$. Despite the tractability, this method has a disadvantage that it does not compute unbiased estimators of the gradients. For this reason, another distance, called *Energy Distance*, or *Cramer Distance*, is introduced in [6]:

$$D_{ED}(p, g) = \sqrt{2\mathbb{E}[\|\mathbf{x} - \mathbf{y}\|] - \mathbb{E}[\|\mathbf{x} - \mathbf{x}'\|] - \mathbb{E}[\|\mathbf{y} - \mathbf{y}'\|]} \tag{5}$$

where $\mathbf{x}, \mathbf{x}'$ are independent samples from p, and $\mathbf{y}, \mathbf{y}'$ are independent samples from g.

2.2 Mini-batch energy distance

This paper [1] introduces another distance called *Mini-batch energy distance*, which is in fact a combination of the Sinkhorn distance and the Energy Distance defined in (4) and (5):

$$D_{MED}(p, g) = \sqrt{2\mathbb{E}[\mathcal{W}_c(\mathbf{X}, \mathbf{Y})] - \mathbb{E}[\mathcal{W}_c(\mathbf{X}, \mathbf{X}')] - \mathbb{E}[\mathcal{W}_c(\mathbf{Y}, \mathbf{Y}')]}. \tag{6}$$

This distance use mini-batches $\mathbf{X}, \mathbf{X}', \mathbf{Y}, \mathbf{Y}'$ instead of point samples $\mathbf{x}, \mathbf{x}', \mathbf{y}, \mathbf{y}'$ and therefore permits a more stable training. In addition, it also computes unbiased mini-batch gradients by introducing a term $-\mathcal{W}_c(\mathbf{Y}, \mathbf{Y}')$ to the formulation, as for the Energy Distance.

2.3 Optimal Transport GAN (OT-GAN)

The OT-GAN method makes use of the mini-batch energy distance defined in 6 to train the network similarly to other GAN methods. One thing to note is that this method utilises the following cosine distance cost function:

$$c_\eta(\mathbf{x}, \mathbf{y}) = 1 - \frac{v_\eta(\mathbf{x}) \cdot v_\eta(\mathbf{y})}{\|v_\eta(\mathbf{x})\|_2 \|v_\eta(\mathbf{y})\|_2} \tag{7}$$

where $v_\eta(\mathbf{x}), v_\eta(\mathbf{y})$ are the output feature vectors of the discriminator applied to $\mathbf{x}$ and $\mathbf{y}$ respectively.

The OT-GAN algorithm is thus presented in Algorithm 1.

3 Experiments

3.1 Implementation details

The OT-GAN model is implemented in Pytorch, where the back-propagation is eased with autograd. To compute the Sinkhorn distance between the mini-batches, I follow this repository[1] and adapt it to the cost given in (7).

[1] https://github.com/gpeyre/SinkhornAutoDiff

Algorithm 1 Optimal Transport GAN (OT-GAN) training algorithm with step size α, using mini-batch SGD for simplicity

Require: n_{gen}, the number of iterations of the generator per critic iteration
Require: η_0, initial critic parameters. θ_0, initial generator parameters
for $t = 1$ to N **do**
  Sample $\mathbf{X}, \mathbf{X}'$ two independent mini-batches from real data, and $\mathbf{Y}, \mathbf{Y}'$ two independent mini-batches from the generated samples
  $\mathcal{L} = \mathcal{W}_c(\mathbf{X}, \mathbf{Y}) + \mathcal{W}_c(\mathbf{X}, \mathbf{Y}') + \mathcal{W}_c(\mathbf{X}', \mathbf{Y}) + \mathcal{W}_c(\mathbf{X}', \mathbf{Y}') - 2\,\mathcal{W}_c(\mathbf{X}, \mathbf{X}') - 2\,\mathcal{W}_c(\mathbf{Y}, \mathbf{Y}')$
  if $t \bmod n_{gen} + 1 = 0$ **then**
    $\eta \leftarrow \eta + \alpha \cdot \nabla_\eta \mathcal{L}$
  else
    $\theta \leftarrow \theta - \alpha \cdot \nabla_\theta \mathcal{L}$
  end if
end for

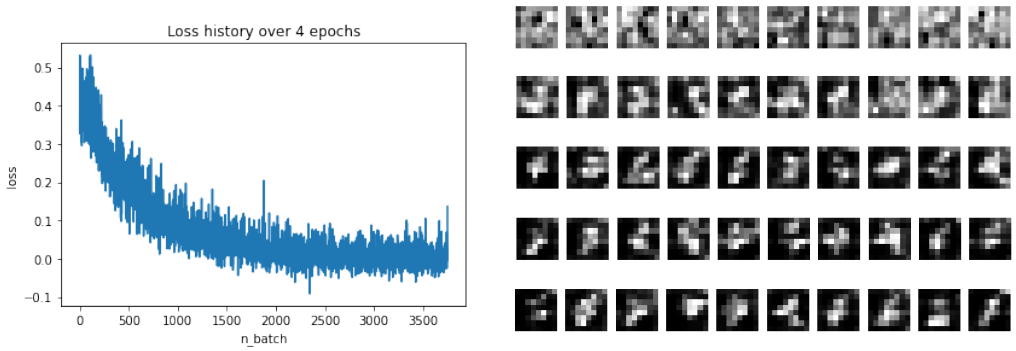
I perform the OT-GAN model with the classic MNIST dataset. However, as the algorithm takes a lot of time to train (even on GPU), the images are resized to have a size 8×8. The training data consists of $60,000$ images.

The generator and discriminator are either simple MLP networks or CNN networks with 2 convolutional blocks. The dimensionalities of the hidden noise space $\mathbf{z}$ and the output space of the discriminator d are taken to be 16, and the ϵ parameter of the regularized OT distance is taken to be 1. The Sinkhorn algorithm is performed with 50 iterations for each run. For training, the batch size (of $\mathbf{X}, \mathbf{X}', \mathbf{Y}, \mathbf{Y}'$) is taken to be 32 and the generator is optimized 5 times more frequent than the discriminator, this is to avoid the cost c to become degenerate. I use Adam optimizer with a learning rate of 10^{-4} for training.

The training is performed on google colab with a GPU, and takes on average 3.5-4s for a batch (of size 32) and roughly 1 hour for one full epoch.

3.2 OT-GAN with MNIST

I first try to use the CNN architectures for the generator and the discriminator and train the model for 4 epochs. The results are illustrated in figure 1.



(a) Training loss history

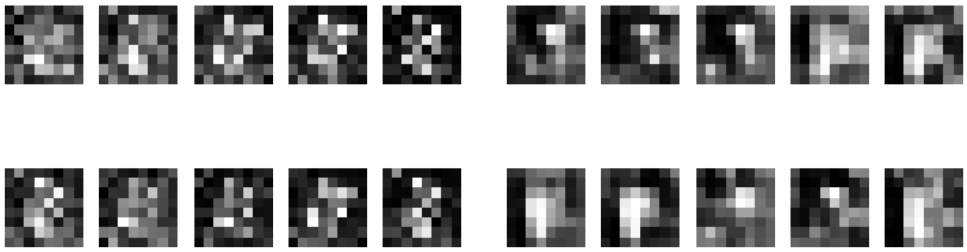
(b) Sampled images over 5 epochs (first row is epoch 0)

Figure 1: Results of OT-GAN for MNIST images with CNN Generator and Discriminator

We can see that the loss decreases well during training. The loss however has a large variation due to the opposite objectives of the generator and the discriminator, and this help to improve the model. Consider the generated images, we observe a great improvement during the first two epochs, the images started to look like digits, this also corresponds to the low loss during training. We thus can deduce that the value of the loss is somewhat coherent with the quality of the generated images.

3.3 Comparison of CNN and MLP networks

I also train a model using MLP discriminator and generator and compare qualitatively the results obtained by MLP and CNN networks.



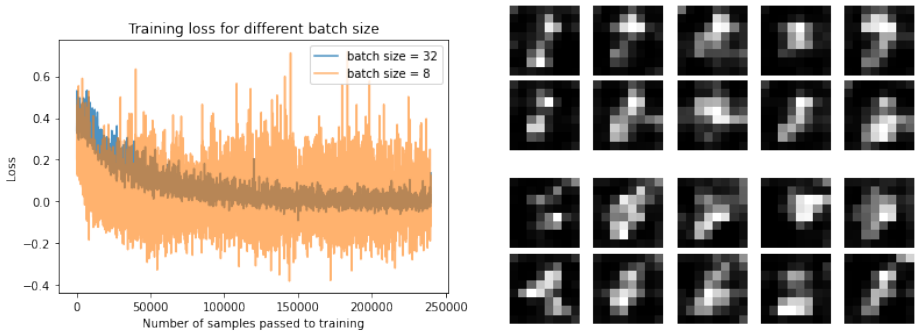
(a) MLP (b) CNN

Figure 2: Comparison of images generated by MLP and CNN networks (after one epoch of training)

We observe that the images obtained by the MLP network is more noisy, compared to smoother ones obtained with the CNN network. This is expected as the MLP network does not capture any spatial relation between the pixels, while the CNN network has a large receptive field, permitting it to generate smoother and more realistic images.

3.4 The effect of batch size

The use of mini-batches has been claimed to have a more stable training. I verify this by training the CNN network for a smaller batch size of 8, this indeed increases the number of batches per epoch by 4, but each batch takes only 0.35s instead of 4s. Finally it takes roughly 0.5 hour to finish one epoch. The results are shown in figure 3.



(a) Training loss history

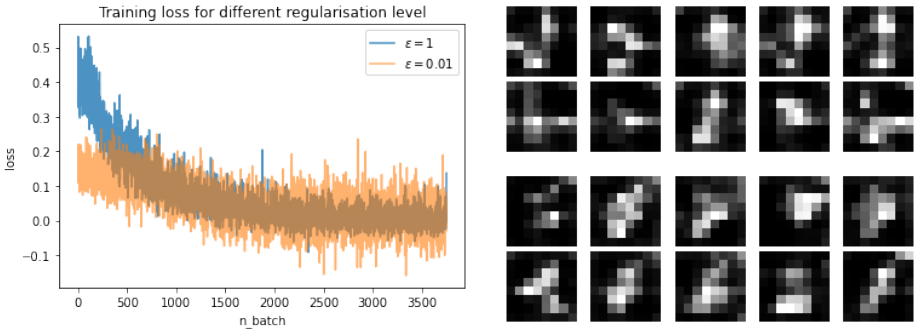
(b) Sampled images for models trained with batch size 8 (top) and 32 (bottom)

Figure 3: Comparison of models trained with different batch size

We see clearly that by using a larger batch size, we obtain more stable training, while using small batch size creates large variance. Consider the generated images, we see that the model trained with smaller batch size create less various samples than the model trained with larger batch size.

3.5 The influence of the regularisation parameter

Finally, we try to use a different regularisation parameter $\epsilon = 0.01$ instead of $\epsilon = 1$. This is expected to yield a more accurate approximation of the Wassertein-1 distance by the Sinkhorn distance, and thus can be expected to have a better distribution approximation. The results are shown in figure 4.



(a) Training loss history

(b) Sampled images for $\epsilon = 0.01$ (top) and $\epsilon = 1$ (bottom)

Figure 4: Comparison of models trained with different batch size

We see that using $\epsilon = 0.01$ yields smaller loss at the beginning, but then it is similar to the case $\epsilon = 1$, even the training has a larger variance than our first case. The images generated using smaller regularisation parameter are not better than larger regularisation parameter as expected. This may be because I haven't train enough the models.

4 Conclusions

I presented OT-GAN, a variant of the GAN model using an OT metrics to minimize the energy distance between the generator distribution and the data distribution. By using mini-batches, it attains a consistent training process with unbiased gradients, resulting in a strong discriminative power and a robust generation ability. I also achieved to implement the network on the MNIST dataset using Pytorch and obtained quite satisfying results.

References

[1] Tim Salimans, Han Zhang, Alec Radford, and Dimitris N. Metaxas. Improving gans using optimal transport. *CoRR*, abs/1803.05573, 2018.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[4] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences, 2017.

[5] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances, 2013.

[6] Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *CoRR*, abs/1705.10743, 2017.