Improving GANs using Optimal Transport

Allouche Dan
ENSAE Paris
Palaiseau, France
dan.allouche@ensae.fr

Dahan Nicolas
ENSAE Paris
Palaiseau, France
nicolas.dahan@ensae.fr

I. Generative Modeling and Optimal Transport

Richard Feynman said : "*What I cannot create, I do not understand*". With this quote we can understand why one of the recent aspirations in machine learning research is to study the problem of how to learn models that generate images, audio, video, text or other data. Generative modeling are providing the means to learn from data in order to model a complex data distribution. We evaluate these models by their ability to generate data that is similar to the input data distribution.

Generative adversarial nets (GANs) are one of the most famous approaches to this problem. In this approach, two neural networks are trained : a generative model G that captures the data distribution and a discriminative model D that distinguishes between generated data and training data. D defines a distance between the model distribution and the data distribution which G can optimize to produce data that more closely resembles the training data.

The authors of the paper, use the contribution of optimal transport in primal form to measure the distance between the distributions. After several research works (in order to avoid problems related : to the tractability of the problem, to biased estimators, to the fact that we use mini-batch in NN and to high dimensional data More precisely) they created a new metric they called *mini-batch energy distance* :

$$D^2(p,g) = 2\mathbb{E}\left[\mathcal{W}_{c_\eta}(\mathbf{X},\mathbf{Y})\right] - \mathbb{E}\left[\mathcal{W}_{c_\eta}(\mathbf{X},\mathbf{X}')\right] - \mathbb{E}\left[\mathcal{W}_{c_\eta}(\mathbf{Y},\mathbf{Y}')\right],$$

where :

- p is the data distribution
- q is the generator distribution
- $\mathbf{X}, \mathbf{X}'$ are independent mini-batches from data distribution p
- $\mathbf{Y}, \mathbf{Y}'$ are independent mini-batches from data distribution q
- $\mathcal{W}_{c_\eta}(\mathbf{X},\mathbf{Y}) = \inf_{M\in\mathcal{M}} Tr[MC^T]$,
- $\mathcal{M}$ the set of matrices with all positive entries, with rows and columns summing to one.
- $C_{i,j} = c_\eta(\mathbf{x}_i, \mathbf{y}_j)$
- $c_\eta(\mathbf{x},\mathbf{y}) = 1 - \frac{v_\eta(\mathbf{x})\cdot v_\eta(\mathbf{y})}{\|v_\eta(\mathbf{x})\|_2 \|v_\eta(\mathbf{y})\|_2}$
- v_η is a deep neural network.

II. Implementation

We implement the OT-GAN described in the paper using pytorch. The architectures of the neural networks are given in the following tables.

operation	activation	kernel	stride	output shape
z				100
linear	GLU			16384
reshape				$1024 \times 4 \times 4$
2x NN upsample				$1024 \times 8 \times 8$
convolution	GLU	5×5	1	$512 \times 8 \times 8$
2x NN upsample				$512 \times 16 \times 16$
convolution	GLU	5×5	1	$256 \times 16 \times 16$
2x NN upsample				$256 \times 32 \times 32$
convolution	GLU	5×5	1	$128 \times 32 \times 32$
convolution	tanh	5×5	1	$3 \times 32 \times 32$

Fig. 1. Generator Architecture

operation	activation	kernel	stride	output shape
convolution	CReLU	5×5	1	$256 \times 32 \times 32$
convolution	CReLU	5×5	2	$512 \times 16 \times 16$
convolution	CReLU	5×5	2	$1024 \times 8 \times 8$
convolution	CReLU	5×5	2	$2048 \times 4 \times 4$
reshape				32768
l2 normalize				32768

Fig. 2. Discriminative Architecture

We train the OT-GAN on the MNIST dataset. We obtain the following generated images:

We recognize with more or less difficulty the numbers.