GAN and WGAN Training

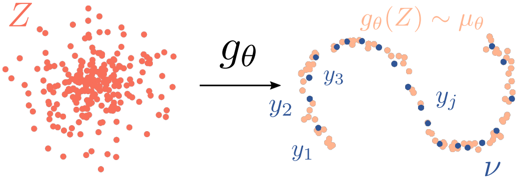
Arthur Leclaire

MVA Generative Modeling
January, 16th, 2024

About Course Validation

- Assignment given in Session 5 (February, 6th)
  Due for Session 8 (February, 27th)
- **Projects**
  Project list given at Session 8 (February, 27th)
  Choice of group and subject for March, 5th
  Project defense: March 25th to 29th
- Attending the practical sessions is **mandatory** for course validation

Learning a Generative Network



GOAL: Estimate a generative model that fits a database $(y_j)_{1 \leq j \leq J}$ of images

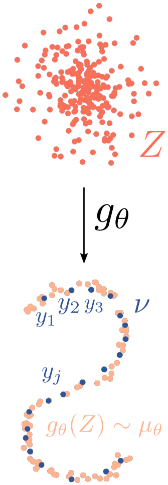
Loss function for Generative Modeling

Learning a Generative Network consists in solving

$$\inf_{\theta \in \Theta} \mathcal{L}(\mu_\theta, \nu)$$

where

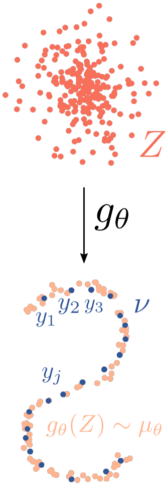
- $\mathcal{L}$ is a loss function between probability distributions μ, ν on $\mathcal{X}, \mathcal{Y} \subset \mathbf{R}^d$
- ... which (sometimes) depends on a "ground cost" $c : \mathcal{X} \times \mathcal{Y} \to \mathbf{R}$ (e.g. $c(x, y) = \|x - y\|_2^2$)
- μ_θ is a probability on a compact $\mathcal{X} \subset \mathbf{R}^d$:
  Often, $g_\theta(Z) \sim \mu_\theta$ with g_θ neural network and $Z \sim \zeta$ input noise
- The generator is parameterized by a θ in a open set $\Theta \subset \mathbf{R}^q$
- ν is a probability on a compact $\mathcal{Y} \subset \mathbf{R}^d$:
  Often, ν is the empirical distribution of the data



Outline

In this session, we will study two approaches for learning generative models:

- Generative Adversarial Networks (GANs)
  based on the Jensen-Shannon divergence $\mathrm{JS}(\mu_\theta, \nu)$
  [Goodfellow et al., 2014]
- Wasserstein Generative Adversarial Networks (WGANs)
  based on the optimal transport cost $W(\mu_\theta, \nu)$
  [Arjovsky et al., 2017]



Adversarial training is related to a *dual formulation* of the loss function.

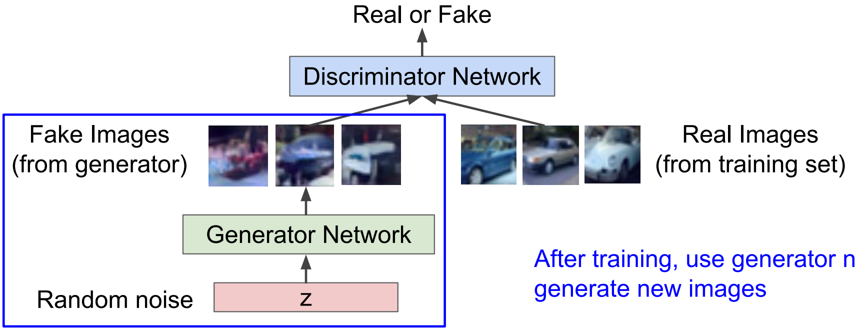
The dual variable is interpreted as a discriminator between real and fake points.

In practice, it will be parameterized by a neural network.

The chosen loss function imposes different constraints on the dual variable.

Adversarial training can be implemented with an alternate algorithm.

Generator v.s. Discriminator



After training, use generator network to generate new images

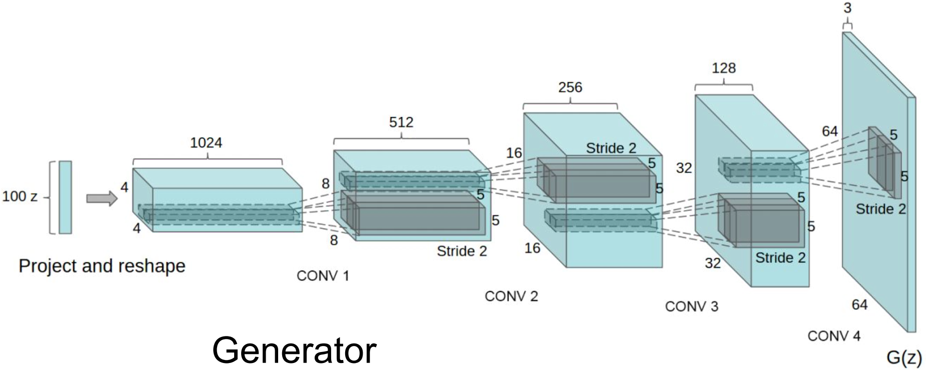
Neural Network architecture

Input noise Z has often distribution uniform $\mathcal{U}([0,1]^p)$ or Gaussian $\mathcal{N}(0, \mathrm{Id})$.

Generator and discriminator networks can have various layers:

- Fully connected layers
- Upsampling or Subsampling layers
- Convolution (with stride)
- Transposed convolution (with stride)
- Activation functions: RELU, leakyRELU, sigmoid, etc
- BatchNorm
- ...

A glimpse on a Generative Architecture



Generator

DCGAN [Radford et al., 2016]

Plan

The Gist of Adversarial Training

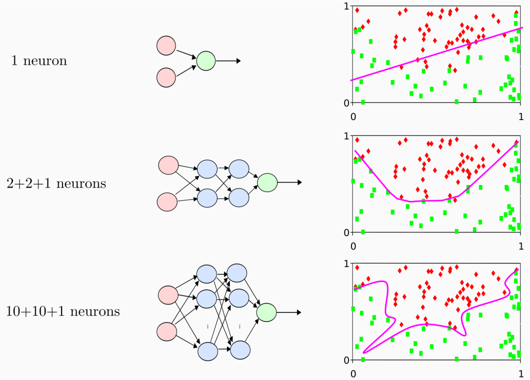
- Train simultaneously a generator g_θ and a discriminator D with alternating updates:

→ Push the discriminator $D : \mathbf{R}^d \rightarrow [0, 1]$ to discriminate between real and fake samples:
$D(g_\theta(z))$ should be close to 0 for any z
$D(y_j)$ should be close to 1 for any data point y_j

→ Push the generator g_θ to fool the discriminator
i.e. push $D(g_\theta(z))$ closer to 1 for any z

Classification of fake points vs data points

For a fixed generator, updating *D* is a kind of classification problem



Discriminator learning

- The discriminator solves a binary classification problem between real and fake images:

$$\max_{D \in \mathcal{D}} \mathbb{E}[\log D(Y)] + \mathbb{E}[\log(1 - D(g_\theta(Z))]$$

  where $\mathcal{D}$ is a (parametric) set of measurable functions $D : \mathbf{R}^d \to [0, 1]$. ($\log 0 = -\infty$.)

- Based on a finite sample $(x^{(i)})$ of real and fake points, this is a logistic regression with labels
  $\ell^{(i)} = 1$ if $x^{(i)}$ is one of the data points (y_j),
  $\ell^{(i)} = 0$ if $x^{(i)}$ is a generated point $g_\theta(Z)$.
  On a finite sample, this loss is called binary cross-entropy (`BCELoss` in PyTorch):

$$\max_{D} \sum_{i=1}^{N} \left[\ell^{(i)} \log D(x^{(i)}) + (1 - \ell^{(i)}) \log \left(1 - D(x^{(i)}) \right) \right]$$

- Finally, adversarial training can be seen as a **min-max** two-player game:

$$\min_{\theta \in \Theta} \max_{D \in \mathcal{D}} \mathbb{E}[\log D(Y)] + \mathbb{E}[\log(1 - D(g_\theta(Z))]$$

Training Algorithm

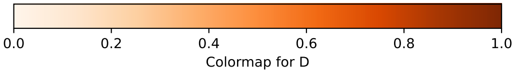
- In practice, g_θ and D are parameterized by neural networks.
  D must have values in $[0, 1]$: take last layer as sigmoid activation $\sigma(x) = \frac{1}{1+e^{-x}}$.
  (Alternately, use `BCEWithLogitsLoss` in PyTorch.)
- The GAN training algorithm alternates between
  - Ascent step(s) on $D \mapsto \mathbb{E}[\log D(Y)] + \mathbb{E}[\log(1 - D(g_\theta(Z))]$
  - Descent step(s) on $\theta \mapsto \min_\theta \mathbb{E}[\log(1 - D(g_\theta(Z))]$

    (or on $\theta \mapsto \mathbb{E}[\log(D(g_\theta(Z))]$; *non-saturating loss*)
- For each step, use stochastic gradient-based updates (SGD, ADAM, ...).
  Each step requires to take samples of $g_\theta(Z)$ and Y

Illustration with a 2D example

Question: can you imagine a good discriminator for the following configuration?

- Dark blue: data points $(y_j)_{1 \leq j \leq J}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ



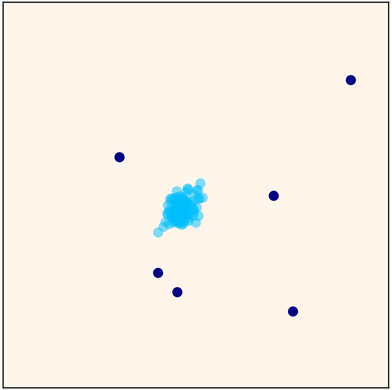
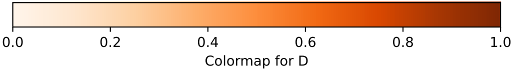


Illustration with a 2D example

Question: can you imagine a good discriminator for the following configuration?

- Dark blue: data points $(y_j)_{1 \le j \le J}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \le k \le 100}$ of μ_θ



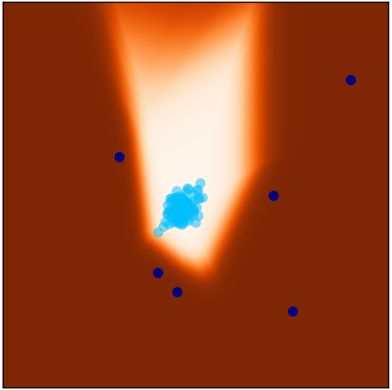
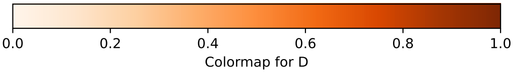
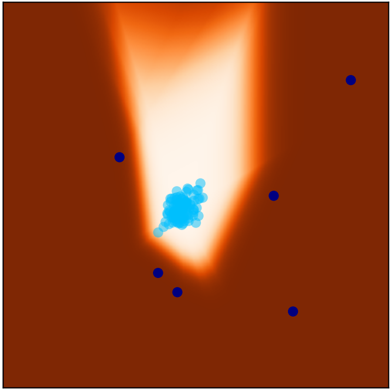


Illustration with a 2D example

Question: can you imagine a good discriminator for the following configuration?

- Dark blue: data points $(y_j)_{1 \leq j \leq J}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ



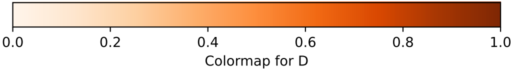


Problem: D is close to 1 on Supp(μ_θ) $\rightarrow$ **"vanishing gradients" issue** (on ∇_θ)

Illustration with a 2D example

And now a tougher example...

- Dark blue: data points $(y_j)_{1 \leq j \leq J}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ



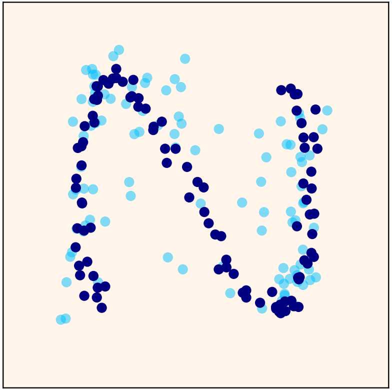
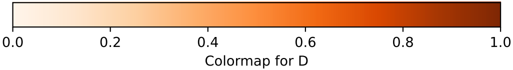
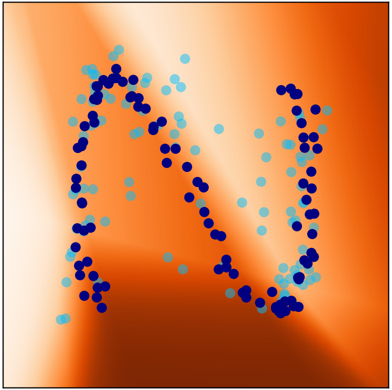


Illustration with a 2D example

And now a tougher example...

- Dark blue: data points $(y_j)_{1 \leq j \leq J}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ





Optimal Discriminator

Let us fix θ. Assume that there is a measure M such that μ_θ and ν have densities w.r.t. M:

$$d\mu_\theta = p_\theta dM \quad \text{and} \quad \nu = qdM \quad \text{(for example, take } M = \mu_\theta + \nu\text{)}.$$

Let

$$L(\theta, D) = \int \log(D) d\nu + \int \log(1 - D) d\mu_\theta.$$

Let $\mathcal{D}_\infty$ the set of measurable functions from $\mathbf{R}^d$ to $[0, 1]$. Remark that

$$0 \geqslant \sup_{D \in \mathcal{D}_\infty} L(\theta, D) \geqslant L(\theta, \tfrac{1}{2}) = -\log 4.$$

Proposition

We have

$$\sup_{D \in \mathcal{D}_\infty} L(\theta, D) = L(\theta, D^*_\theta) \quad \textit{with} \quad D^*_\theta = \frac{q}{q + p_\theta}.$$

Remark: The optimal discriminator is unique as soon as $p_\theta > 0$, M-.a.e. [Biau et al., 2018].

Relation with Jensen-Shannon divergence

Recall the definition of the Kullback-Leibler divergence between probability measures μ, ν:

$$\mathrm{KL}(\mu|\nu) = \begin{cases} \int \log(\frac{d\mu}{d\nu}) d\mu & \text{if } \frac{d\mu}{d\nu} \text{ exists,} \\ +\infty & \text{otherwise.} \end{cases}$$

Recall that $\mathrm{KL}(\mu, \nu) \geq 0$ with equality if and only if $\mu = \nu$.

Also, $\mathrm{KL}(\mu_n, \mu) \to 0$ implies $\mu_n \to \mu$ in total variation (Pinsker inequality, see [Tsybakov, 2008]).

The Jensen-Shannon divergence is defined by

$$\mathrm{JS}(\mu, \nu) = \frac{1}{2} \mathrm{KL}(\mu, \tfrac{\mu+\nu}{2}) + \frac{1}{2} \mathrm{KL}(\nu, \tfrac{\mu+\nu}{2}).$$

Proposition

We have

$$\sup_{D \in \mathcal{D}} L(\theta, D) = L(\theta, D_\theta^*) = 2\, \mathrm{JS}(\mu_\theta, \nu) - \log 4.$$

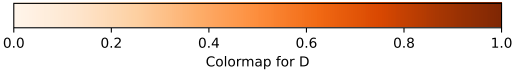
Insufficiency of the Jensen-Shannon divergence

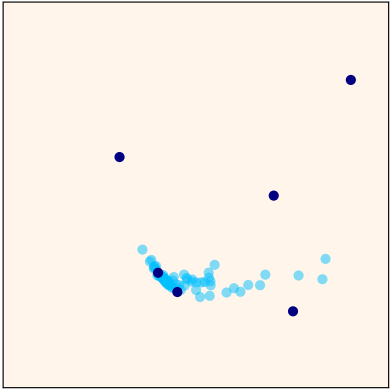
- If there exists A such that $\mu_\theta(A) = 0$ and $\nu(A^c) = 0$,
  then there is an optimal D^*_θ such that $D^*_\theta = 0$ on A^c and $D^*_\theta = 1$ on A.
  Therefore, $L(\theta, D^*_\theta) = 0$, i.e. $\text{JS}(\mu_\theta, \nu) = \log 2$.
  Problem: This does not depend on how "close" the supports are.
- When ν is the empirical data distribution, it has finite support $A = \mathcal{Y}$.
  Assume that $\mu_\theta(A) = 0$ (true as soon as μ_θ has a density).
  Then D^*_θ is ≈ 0 around fake points, and ≈ 1 around data points.
  Problem: With D^*_θ, the gradient w.r.t. θ is not informative (*vanishing gradients*)
- Why does it work then?
  $\rightarrow$ Because the parameterized discriminator is in practice smoother than D^*_θ.

What did you expect?

Final configuration. What is the final discriminator?

- Dark blue: data points $(y_j)_{1 \leq j \leq 6}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ

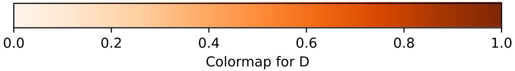


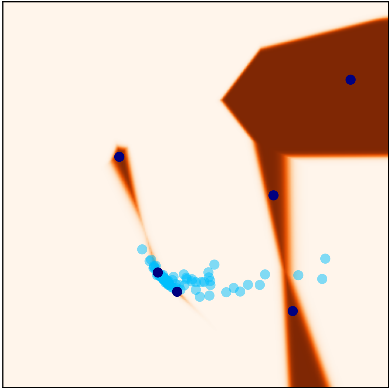


What did you expect?

Final configuration. What is the final discriminator?

- Dark blue: data points $(y_j)_{1 \leq j \leq 6}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ

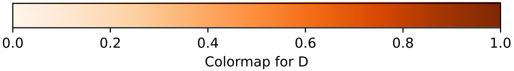


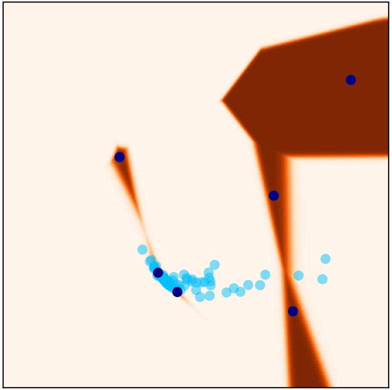


What did you expect?

What happens if we update only the generator?

- Dark blue: data points $(y_j)_{1 \leq j \leq 6}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ

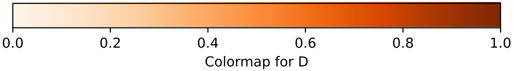


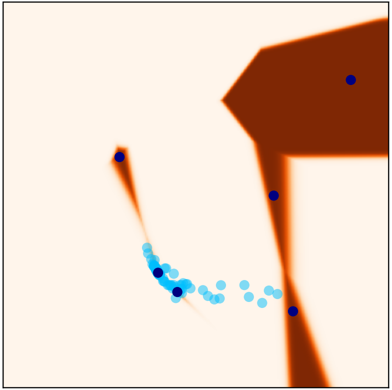


What did you expect?

What happens if we update only the generator?

- Dark blue: data points $(y_j)_{1 \leq j \leq 6}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ

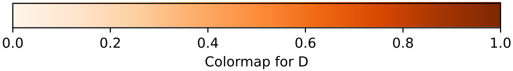


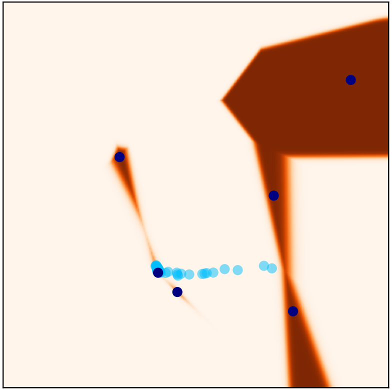


What did you expect?

What happens if we update only the generator?

- Dark blue: data points $(y_j)_{1 \leq j \leq 6}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ

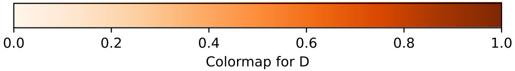


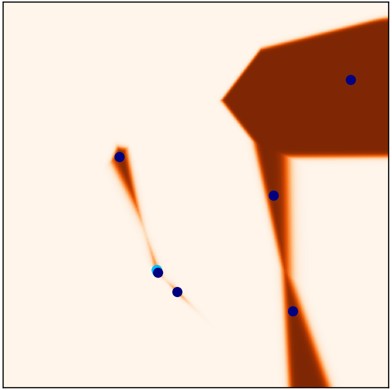


What did you expect?

What happens if we update only the generator?

- Dark blue: data points $(y_j)_{1 \leq j \leq 6}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ

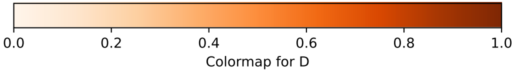


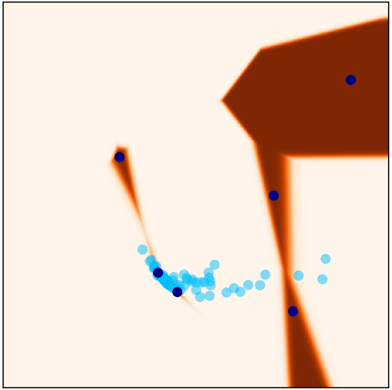


What did you expect?

And if we retrain the discriminator?

- Dark blue: data points $(y_j)_{1 \leq j \leq 6}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ

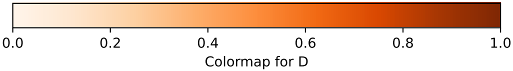


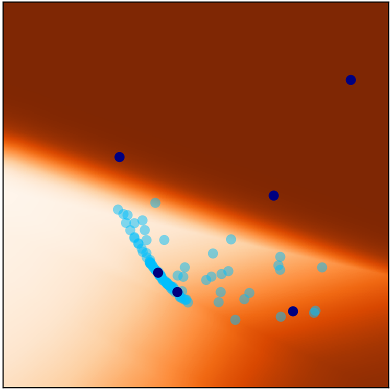


What did you expect?

And if we retrain the discriminator?

- Dark blue: data points $(y_j)_{1 \leq j \leq 6}$
- Light blue: 100 samples $(g_\theta(z_k))_{1 \leq k \leq 100}$ of μ_θ





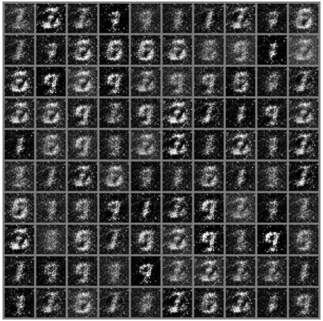
GAN Training for MNIST digits (next week)

Training with MNIST (60 000 images)

- Adam optimizer
- Learning rate 0.0002 for both the discriminator and the generator

Real images

Fake images, epoch 1



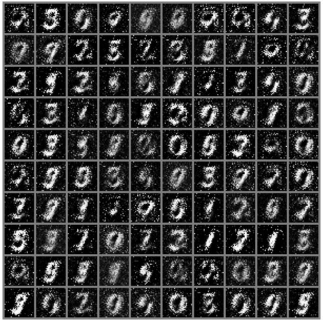
GAN Training for MNIST digits (next week)

Training with MNIST (60 000 images)

- Adam optimizer
- Learning rate 0.0002 for both the discriminator and the generator

Real images

Fake images, epoch 2



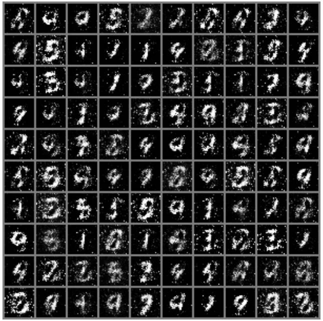
GAN Training for MNIST digits (next week)

Training with MNIST (60 000 images)

- Adam optimizer
- Learning rate 0.0002 for both the discriminator and the generator

Real images

Fake images, epoch 3



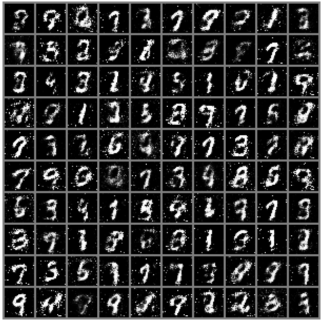
GAN Training for MNIST digits (next week)

Training with MNIST (60 000 images)

- Adam optimizer
- Learning rate 0.0002 for both the discriminator and the generator

Real images

Fake images, epoch 10



GAN Training for MNIST digits (next week)

Training with MNIST (60 000 images)

- Adam optimizer
- Learning rate 0.0002 for both the discriminator and the generator

Real images

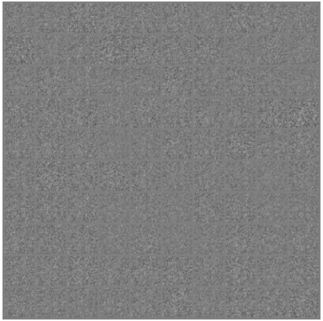
Fake images, epoch 100

GAN Training for MNIST digits (next week)

Training GANs is quite unstable!
The generator can suffer from *mode collapse*:
i.e. it always produces the same image (one mode only).
Example: same as before **but with SGD instead of Adam**.

Real images

Fake images, epoch 1



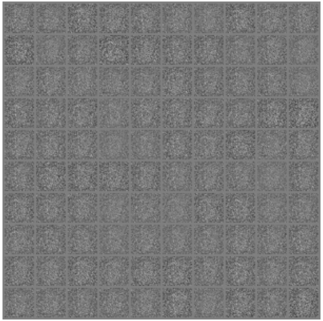
GAN Training for MNIST digits (next week)

Training GANs is quite unstable!

The generator can suffer from *mode collapse*:
i.e. it always produces the same image (one mode only).
Example: same as before **but with SGD instead of Adam**.

Real images

Fake images, epoch 2



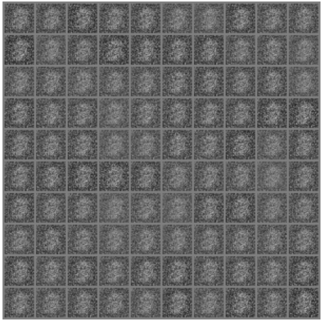
GAN Training for MNIST digits (next week)

Training GANs is quite unstable!

The generator can suffer from *mode collapse*:
i.e. it always produces the same image (one mode only).
Example: same as before **but with SGD instead of Adam**.

Real images

Fake images, epoch 3

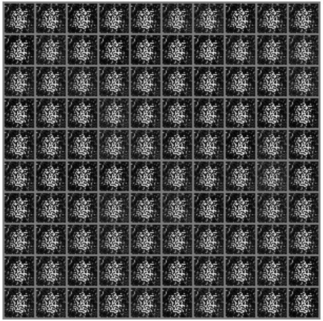


GAN Training for MNIST digits (next week)

Training GANs is quite unstable!
The generator can suffer from *mode collapse*:
i.e. it always produces the same image (one mode only).
Example: same as before **but with SGD instead of Adam**.

Real images

Fake images, epoch 10



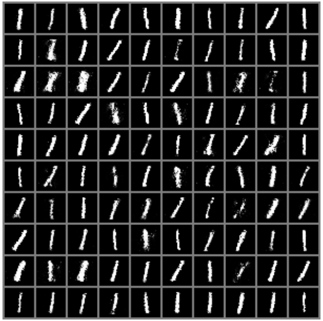
GAN Training for MNIST digits (next week)

Training GANs is quite unstable!

The generator can suffer from *mode collapse*:
i.e. it always produces the same image (one mode only).
Example: same as before **but with SGD instead of Adam**.

Real images

Fake images, epoch 100



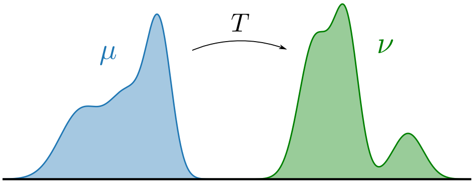
Plan

Optimal Transport (see G. Peyré's or Villani's books)

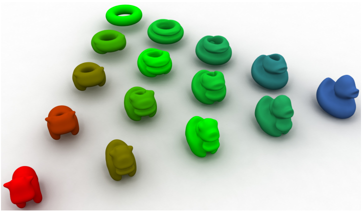
For μ, ν probability measures on $\mathbf{R}^d$, let

$$\mathrm{OT}(\mu, \nu) = \min_T \int_{\mathbf{R}^d} c(x, T(x))d\mu(x)$$

where T should send μ onto ν.



COLOR TRANSFER



SHAPE INTERPOLATION

Two OT formulations

Let μ, ν two probability distributions supported in $\mathcal{X}, \mathcal{Y} \subset \mathbf{R}^d$.

OPTIMAL TRANSPORT COST WITH MONGE FORMULATION:

$$\mathrm{OT}(\mu, \nu) = \min_{T\sharp\mu=\nu} \int_{\mathbf{R}^d} c(x, T(x)) d\mu(x) \qquad \text{(OT-Monge)}$$

where $T\sharp\mu(A) = \mu(T^{-1}(A))$ for all A.

OPTIMAL TRANSPORT COST WITH KANTOROVICH FORMULATION:

$$W(\mu, \nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x, y)\, d\pi(x, y) \qquad \text{(OT-Kanto)}$$

where $\Pi(\mu, \nu)$ is the set of distributions π on $\mathcal{X} \times \mathcal{Y}$ with marginals μ, ν.

NB: If T solves (OT-Monge), then the law of $(X, T(X))$ (with $X \sim \mu$) solves (OT-Kanto).
Also, under weak regularity assumptions on μ, $\mathrm{OT}(\mu, \nu) = W(\mu, \nu)$ [Santambrogio, 2015].

Metric Properties

For $c(x,y) = \|x-y\|^p$, $p \in [1, \infty)$, the p-Wasserstein cost is defined by

$$W_p(\mu,\nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \|x-y\|^p \, d\pi(x,y).$$

Theorem (See e.g. Chap 6 of [Villani, 2009])

Let $\mathcal{P}_p$ the set of probability measures μ on $\mathbf{R}^d$ such that $\int \|x\|^p d\mu(x) < \infty$.

- $W_p^{\frac{1}{p}}$ *is a distance on $\mathcal{P}_p$.*
- $\mu_n \xrightarrow[n\to\infty]{W_p} \mu$ *if and only if* $\begin{cases} \forall \varphi \in \mathscr{C}_b(\mathbf{R}^d), & \int \varphi d\mu_n \to \int \varphi d\mu \\ \int \|x\|^p d\mu_n(x) \to \int \|x\|^p d\mu(x) \end{cases}$.

Dual Optimal Transport

Theorem

If μ, ν are supported in $\mathcal{X}, \mathcal{Y}$ compact and if c is continuous on $\mathcal{X} \times \mathcal{Y}$, then

$$W(\mu, \nu) = \sup_{\varphi, \psi} \int \varphi(x) d\mu(x) + \int \psi(y) d\nu(y),$$

where $\varphi \in \mathscr{C}(\mathcal{X}), \psi \in \mathscr{C}(\mathcal{Y})$ are such that $\varphi(x) + \psi(y) \leqslant c(x, y)$ for all $x \in \mathcal{X}, y \in \mathcal{Y}$.

For fixed ψ, the optimal φ is the ***c*-transform** defined by

$$\psi^c(x) = \min_{y \in \mathcal{Y}} c(x, y) - \psi(y).$$

Theorem

If μ, ν are supported in $\mathcal{X}, \mathcal{Y}$ compact and if c is continuous on $\mathcal{X} \times \mathcal{Y}$, then

$$W(\mu, \nu) = \sup_{\psi \in \mathscr{C}(\mathcal{Y})} \int \psi^c(x) d\mu(x) + \int \psi(y) d\nu(y),$$

Duality - sketch of proof

Let $\mathcal{M}_+(\mathcal{X} \times \mathcal{Y})$ the set of non-negative measures on $\mathcal{X} \times \mathcal{Y}$.

We put the constraint in the functional by noticing

$$\sup_{\varphi,\psi} \int \varphi d\mu + \int \psi d\nu - \int (\varphi(x) + \psi(y)) d\pi(x,y) = \begin{cases} 0 & \text{if} \quad \pi \in \Pi(\mu,\nu) \\ +\infty & \text{otherwise} \end{cases}.$$

We get the problem

$$\inf_{\pi \in \mathcal{M}_+(\mathcal{X} \times \mathcal{Y})} \sup_{\varphi,\psi} \int c(x,y) d\pi(x,y) + \int \varphi d\mu + \int \psi d\nu - \int (\varphi(x) + \psi(y)) d\pi(x,y).$$

Using Fenchel-Rockafellar duality, we can exchange inf-sup and get

$$\sup_{\varphi,\psi} \Big(\int \varphi d\mu + \int \psi d\nu + \underbrace{\inf_{\pi \in \mathcal{M}_+(\mathcal{X} \times \mathcal{Y})} \int (c(x,y) - \varphi(x) - \psi(y)) d\pi(x,y)}_{= \begin{cases} 0 & \text{if } \varphi(x) + \psi(y) \leqslant c(x,y) \ d\mu(x) d\nu(y) \text{ a.e} \\ -\infty & \text{otherwise} \end{cases}} \Big).$$

Regularity of dual solutions

Proposition

Assume that c is L-Lipschitz. Then for any $\psi \in \mathscr{C}(\mathcal{Y})$, ψ^c is L-Lipschitz.

Consequence for $c(x, y) = \|x - y\|$ **on** $\mathcal{X} = \mathcal{Y}$:
There exist 1-Lipschitz solutions with $\psi^c = -\psi$. Therefore,

$$W_1(\mu, \nu) = \sup_{\psi \in \text{Lip}_1(\mathcal{Y})} -\int \psi(x) d\mu(x) + \int \psi(y) d\nu(y)$$

Wasserstein Generative Networks (WGAN)

Learning a Wasserstein WGAN consists in solving

$$\operatorname*{Argmin}_{\theta \in \Theta} W(\mu_\theta, \nu),$$

For any groundcost c, we can use the c-transform formulation:

$$W(\mu_\theta, \nu) = \sup_{\psi \in \mathscr{C}(\mathcal{Y})} \mathbb{E}[\psi(Y)] + \mathbb{E}[\psi^c(g_\theta(Z))].$$

For $c(x, y) = \|x - y\|$, we get the usual WGAN formulation [Arjovsky et al., 2017]:

$$W_1(\mu_\theta, \nu) = \sup_{D \in \text{Lip}_1} \mathbb{E}[D(Y)] - \mathbb{E}[D(g_\theta(Z))].$$

Advantage of the Wasserstein cost over KL: it is sensitive to the groundcost!
(and thus to the distance between the supports of μ_θ and ν)

Recall Loss functions

- Loss function for **"Vanilla" GAN:**

$$\sup_{D \in \mathcal{D}_\infty} \mathbb{E}[\log D(Y)] + \mathbb{E}[\log(1 - D(g_\theta(Z)))]$$

- Loss function for **WGAN** (for the 1-Wasserstein cost):

$$\sup_{D \in \mathrm{Lip}_1} \mathbb{E}_{Y \sim \nu}[D(Y)] - \mathbb{E}_{Z \sim \zeta}[D(g_\theta(Z))].$$

  We just got rid of the $\log$ and $D(x)$ is not in $[0, 1]$... but we now have a constraint "$D \in \mathrm{Lip}_1$".

- The WGAN training algorithm alternates between
  · Ascent step(s) on $D \mapsto \mathbb{E}[D(Y)] - \mathbb{E}[D(g_\theta(Z)]$
  · Descent step(s) on $\theta \mapsto \min_\theta \mathbb{E}[-D(g_\theta(Z))]$
- But, we have to constrain $D \in \mathrm{Lip}_1$ along the way...

Learning Lipschitz discriminators

- The original WGAN paper [Arjovsky et al., 2017] uses weight clipping to restrict the Lipschitz constant:

```
for p in D.parameters():
    p.data.clamp_(-c, c)
```

- Alternately, [Gulrajani et al., 2017] proposed to change the discriminator loss in order to penalize the Lipschitz constant of D.
- This requires to estimate the Lipschitz constant of D.

Practical estimation of a Lipschitz constant

From points $(x_i), (y_j)$, we can sample the segments $[x_i, y_j]$:

$$a_{ij} = (1 - u_{ij}x_i) + u_{ij}y_j \quad \text{with} \quad u_{ij} \sim \mathcal{U}(0,1),$$

and then compute $\nabla D(a_{ij})$ by automatic differentiation:

```
def lipconstant(D,x,y):
    m = x.shape[0]
    n = y.shape[0]
    u = torch.rand((m,n,1))
    xy = (u * y[None,:,:] + (1 - u) * x[:,None,:]).flatten(end_dim=1)
    xy.requires_grad_()

    Dxy = D(xy)
    gradout = torch.ones(Dxy.size())
    gradients = torch.autograd.grad(outputs=Dxy, inputs=xy, grad_outputs=gradout,
       create_graph=True, retain_graph=True)[0]

    gradients_norm = torch.sqrt(torch.sum(gradients ** 2, dim=1))

    return torch.mean(gradients_norm)
```

NB: For sufficiently large batches $(x_i), (y_i)$ of same size, you can just use the points

$$a_i = (1 - u_i x_i) + u_i y_i \quad \text{with} \quad u_i \sim \mathcal{U}(0,1).$$

The Gradient Penalty

- Actually, Gulrajani et al. propose to use a finer property of W_1:
the optimal dual potential φ satisfies $\|\nabla \phi\| = 1$ on segments joining samples from μ_θ and ν.
(see e.g. [Santambrogio, 2015], and also a remark later in these slides)
- Therefore, they proposed to include a "gradient penalty" in the loss:

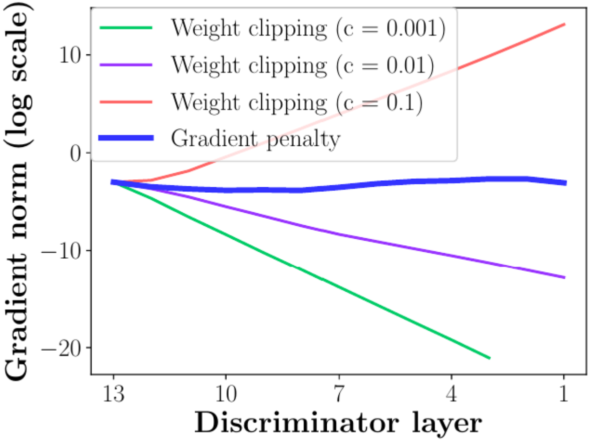
$$\mathrm{GP}(D) = \mathbb{E}[(\|\nabla D(X)\| - 1)^2] \quad \text{where} \quad X \sim \mathcal{U}([g_\theta(Z), Y]).$$

Warning: the gradient is with respect to the variable x and not the parameters θ.
- This leads to the **WGAN-GP** discriminator loss (with penalty weight $\lambda > 0$):

$$\sup_D \mathbb{E}[D(Y)] - \mathbb{E}[D(g_\theta(Z))] - \lambda \mathbb{E}[(\|\nabla D(X)\| - 1)^2].$$

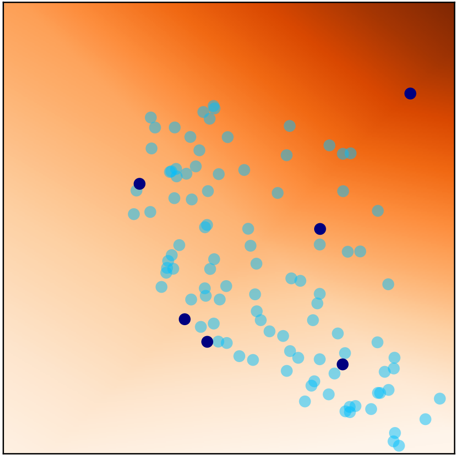
- We could also do a unilateral penalty $\mathbb{E}[(\|\nabla D(X)\| - 1)_+^2]$.

WGAN: Gradient Penalty v.s. Weight clipping

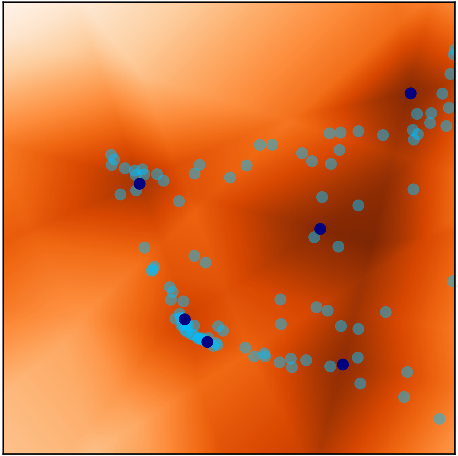


(source: [Gulrajani et al., 2017])

Example of WGAN training



WGAN-WC



WGAN-GP

WGAN Stability

WGAN-GP is a more stable way to train deep convolutional generators/discriminators.
But the results still depend highly on the optimization strategy and on the networks architectures.

Figure 2: Different GAN architectures trained with different methods. We only succeeded in training every architecture with a shared set of hyperparameters using WGAN-GP.

(source: [Gulrajani et al., 2017])

Plan

WGAN in the semi-discrete case

The rest of the section is devoted to WGAN learning with **semi-discrete optimal transport**.

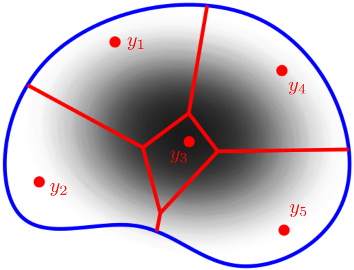
Semi-discrete Optimal transport is the case where

- μ **has a density on $\mathbf{R}^d$**
- ν **has finite support** i.e. $\mathcal{Y}$ finite

More generally, we will also have in mind the case where μ has a density on a subspace (or submanifold) of $\mathbf{R}^d$.

In the semi-discrete case, we will see that

- we know the form of the OT map
- we can use the c-transform for stable WGAN learning



Example:
μ is a density in graylevels
ν is uniform on $\mathcal{Y} = \{y_j\}$

Laguerre Diagram

[Aurenhammer et al., 1998], [Kitagawa et al., 2017]

In this semi-discrete case, we will look for solutions of (OT-Monge) under the form

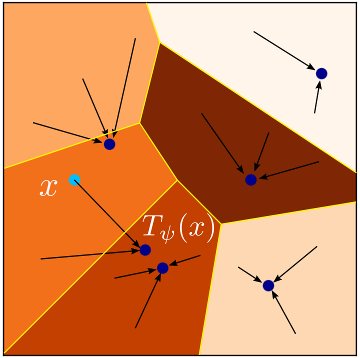
$$T_\psi(x) = \underset{y \in \mathcal{Y}}{\text{Argmin}}\ c(x, y) - \psi(y)$$

where $\psi \in \mathbf{R}^{\mathcal{Y}}$. Here, $\psi = (\psi(y_1), \ldots, \psi(y_J))$.

The preimages of T_ψ form a **Laguerre diagram**.

$\mathbb{L}_\psi(y) = T_\psi^{-1}(y)$ is called the Laguerre cell of y.

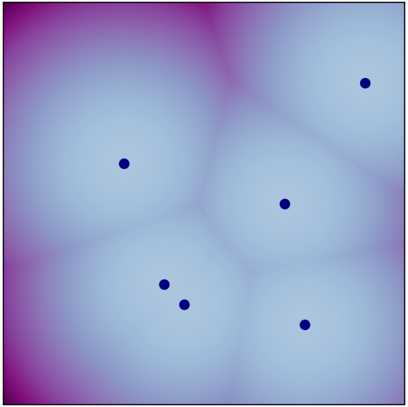
- Very simple parameterization
- Stochastic Algorithm to compute ψ (wait for it...)



$$\mu = \mathcal{U}([0,1]^2) \longrightarrow \nu = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \delta_y$$

Let's look at c-transforms for the quadratic cost

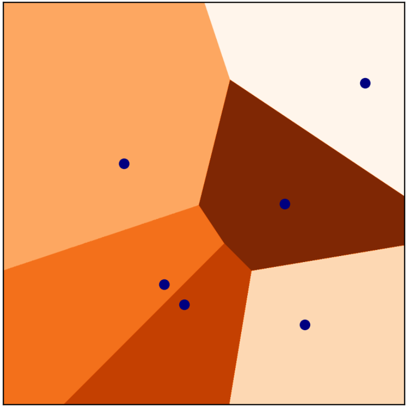
Suppose that we want to compute the optimal transport from $\mu = \mathcal{U}([0,1]^2)$ to $\nu = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \delta_y$.



$\psi^c(x) = \min_j \|x - y_j\|^2$ with $\psi = 0$

Let's look at c-transforms for the quadratic cost

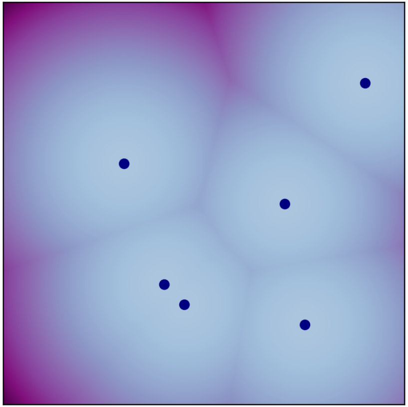
Suppose that we want to compute the optimal transport from $\mu = \mathcal{U}([0,1]^2)$ to $\nu = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \delta_y$.



Voronoi diagram ($\psi = 0$)

Let's look at c-transforms for the quadratic cost

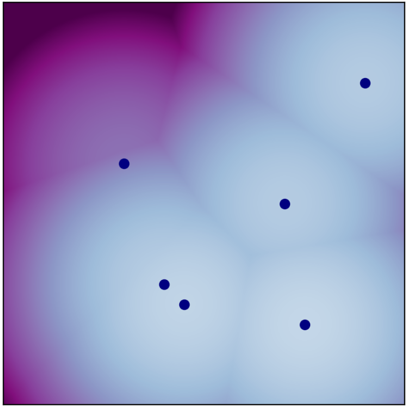
Suppose that we want to compute the optimal transport from $\mu = \mathcal{U}([0,1]^2)$ to $\nu = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \delta_y$.



$\psi^c(x) = \min_j \|x - y_j\|^2$ with $\psi = 0$

Let's look at c-transforms for the quadratic cost

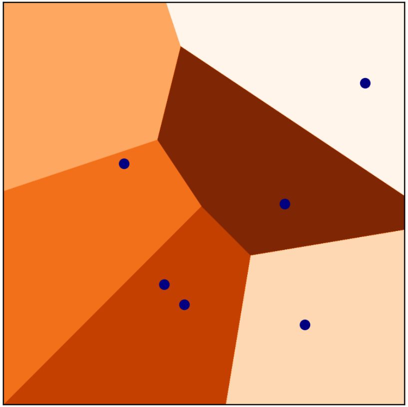
Suppose that we want to compute the optimal transport from $\mu = \mathcal{U}([0,1]^2)$ to $\nu = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \delta_y$.



$\psi^c(x) = \min_j \|x - y_j\|^2 - \psi(y_j)$ with optimal ψ

Let's look at c-transforms for the quadratic cost

Suppose that we want to compute the optimal transport from $\mu = \mathcal{U}([0,1]^2)$ to $\nu = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \delta_y$.



Laguerre diagram with optimal ψ

Optimality of T_ψ

Proposition

T_ψ is an optimal mapping between μ and $m := (T_\psi)_\sharp \mu$.

Proof.

Let $T : \mathcal{X} \to \mathcal{Y}$ measurable such that $T_\sharp \mu = m$.
Using the definition of T_ψ and integrating,

$$\int \Big(c(x, T_\psi(x)) - \psi(T_\psi(x))\Big) d\mu(x) \leqslant \int \Big(c(x, T(x)) - \psi(T(x))\Big) d\mu(x)$$

But since $m = (T_\psi)_\sharp \mu = T_\sharp \mu$ we have

$$\int \psi(T_\psi(x)) d\mu(x) = \int \psi(T(x)) d\mu(x) = \int \psi(y) dm(y)$$

and thus

$$\int c(x, T_\psi(x)) d\mu(x) \leqslant \int c(x, T(x)) d\mu(x).$$

□

Towards a finite-dimensional concave problem

In the semi-discrete setting, ν has finite support $\mathcal{Y} = \{y_1, \ldots, y_J\}$. Writing $v_j = \psi(y_j)$ and $\nu_j = \nu(\{y_j\})$, we have

$$\int \psi d\nu = \sum_{j=1}^{J} \psi(y_j)\nu(\{y_j\}) = \sum_j \nu_j v_j.$$

We thus have to maximize the function

$$H(v) = \int_X \left(\min_j c(x, y_j) - v_j \right) d\mu(x) + \sum_j \nu_j v_j \qquad (v \in \mathbf{R}^J).$$

Dual Problem

Theorem ([Kitagawa et al., 2019])

Assume that μ has a density w.r.t. Lebesgue measure λ on $\mathbf{R}^d$, and that ν has finite support $\mathcal{Y}$. Assume also that

$$\forall y, z \in \mathcal{Y}, \forall t \in \mathbf{R}, \quad \lambda(\{\, x \,|c(x,y) - c(x,z) = t\}) = 0.$$

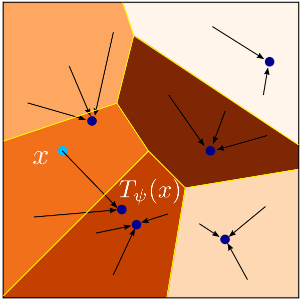
Then, a solution to (OT) is given by T_ψ where $v = (\psi(y_j)) \in \mathbf{R}^J$ maximizes the $\mathcal{C}^1$ concave function

$$H(v) = \int_{\mathbf{R}^d} \big(\min_j \|x - y_j\|^2 - v_j\big)\, d\mu(x) + \sum_j \nu_j v_j,$$

whose gradient is given by $\dfrac{\partial H}{\partial v_j} = -\mu(\mathbb{L}_\psi(y_j)) + \nu_j$.

NB: *H* is not strictly concave in general.

Semi-discrete OT and Mass constraints



$$\mu = \mathcal{U}([0,1]^2) \longrightarrow \nu = \frac{1}{|\mathcal{Y}|}\sum_{y\in\mathcal{Y}} \delta_y$$

Corollary

The following statements are equivalent

- *v is a global maximizer of H*
- *T_v is an optimal transport map between μ and ν*
- $(T_v)_\sharp \mu = \nu$

Consequence: Solving semi-discrete OT from μ to ν amounts to finding a Laguerre diagram $(\mathbb{L}_\psi(y))_{y\in\mathcal{Y}}$ that divides the μ-mass according to the target masses ν:

$$\forall j, \quad \mu(\mathbb{L}_\psi(y_j)) = \nu(\{y_j\}).$$

Remark linked to the Gradient Penalty

Consider the c-transform for the 1-Wasserstein cost:

$$\psi^c(x) = \min_j \|x - y_j\| - \psi(y_j).$$

On $\mathbb{L}_\psi(y_j)$, we have $T_\psi(x) = y_j$ and $\psi^c(x) = \|x - y_j\| - \psi(y_j)$ and then, if $x \neq y_j$,

$$\nabla \phi(x) = \nabla \psi^c(x) = \nabla \|x - y_j\| = \frac{x - y_j}{\|x - y_j\|}.$$

In particular, $\|\nabla \phi(x)\| = 1$, justifying the GP term of [Gulrajani et al., 2017].

Question: Is this still true for the 2-Wasserstein cost? (i.e. with $c(x, y) = \|x - y\|^2$)

ASGD Algorithm for Semi-Discrete OT

The optimal dual variable v for $W(\mu, \nu)$ can be found via a stochastic algorithm. Indeed, write

$$W(\mu,\nu) = \max_v H(v) = \max_v \mathbb{E}_{X\sim\mu_\theta}\left[\tilde{H}(v,X)\right] \quad \text{with} \quad \tilde{H}(v,x) = v^c(x) + \int v d\nu$$

with *Averaged Stochastic Gradient Descent* (ASGD): [Genevay et al., 2016]

$$\forall k \in \mathbb{N}^*, \quad \begin{cases} \widetilde{v}_k & = \widetilde{v}_{k-1} + \frac{\gamma}{\sqrt{k}}\left(\frac{1}{|B_k|}\sum_{x\in B_k} \partial_v \tilde{H}(\widetilde{v}_{k-1}, x)\right) \\ v_k & = \frac{1}{k}(\widetilde{v}_1 + \cdots + \widetilde{v}_k), \end{cases}$$

where $\gamma > 0$ is the learning rate, and the (B_k) are batches of samples of μ_θ.

Proposition

- *$H(\cdot)$ is a concave function*
- *We have the convergence guarantee in expectation (w.r.t. the batches B_k)*

$$\mathbb{E}[H(v_*) - H(v_k)] = \mathcal{O}\left(\frac{\log k}{\sqrt{k}}\right),$$

Exercise 1

On $\mathbf{R}^2$ we consider the groundcost $c(x, y) = \|x - y\|$ (Euclidean distance).
Compute $\mathrm{JS}(\mu, \nu)$ and $W_1(\mu, \nu)$ for the following measures on $\mathbf{R}^2$:

- μ uniform on the square of vertices $(0, \pm 1)$, $(\pm 1, 0)$.
- $\nu = \frac{1}{2}\delta_{y_1} + \frac{1}{4}\delta_{y_2} + \frac{1}{4}\delta_{y_3}$ with

$$y_1 = (2, 0), \quad y_2 = (-1, 1) \quad y_3 = (-1, -1).$$

Exercise 2

Consider

- μ_θ the uniform distribution on the segment $[a, b]$ with $\theta = (a, b) \in \Theta = (\mathbf{R}^2)^2$,
- $\nu = \frac{1}{2}\delta_{y_1} + \frac{1}{2}\delta_{y_2}$ with $y_1 = (-1, 0)$ and $y_2 = (1, 0)$,
- $c(x, y) = \|x - y\|^2$.

1) For any $\theta \in \Theta$, compute $W(\mu_\theta, \nu)$.

2) Solve $\min_{\theta\in\Theta} W(\mu_\theta, \nu)$.

The Gradient formula

Let us write

$$h(\theta) := W(\mu_\theta, \nu) = \max_{\psi \in \mathscr{C}(\mathcal{Y})} H(\psi, \theta) \quad \text{where} \quad H(\psi, \theta) = \int_{\mathcal{X}} \psi^c d\mu_\theta + \int_{\mathcal{Y}} \psi d\nu.$$

Proposition ([Arjovsky et al., 2017])

Let θ_0 and ψ_0 satisfying $h(\theta_0) = H(\psi_0, \theta_0)$.
If h and $\theta \mapsto H(\psi_0, \theta)$ are both differentiable at θ_0, then

$$\nabla h(\theta_0) = \nabla_\theta H(\psi_0, \theta_0). \tag{Grad-OT}$$

Problem : there are cases where no such couple (ψ_0, θ_0) exists.
(Exercise: find such a case.)

A sufficient condition for (Grad-OT)

Theorem ([Houdard et al., 2023])

Suppose that Card$(\mathcal{Y}) = J < \infty$ and c Lipschitz and $\mathscr{C}^1$ in x. Suppose also that

- *$\forall \theta \in \Theta$, the optimal ψ_* for $W(\mu_\theta, \nu)$ is unique up to additive constants.*
- *$\forall \theta \in \Theta$, $\forall \psi \in \mathbf{R}^J$, μ_θ does not charge the interface of the Laguerre diagram of ψ,*

$G(\Theta)$: *$\forall \theta_0 \in \Theta$, there is a neighborhood V of θ_0 and $K \in L^1(\zeta)$ such that $g(\cdot, Z)$ is a.s. $\mathscr{C}^1$ on V and*

$$\forall \theta \in V, \quad \zeta\text{-a.s.}. \quad \|g(\theta, Z) - g(\theta_0, Z)\| \leqslant K(Z)\|\theta - \theta_0\|.$$

Then $h_0(\theta) = W_0(\mu_\theta, \nu)$ is differentiable at any $\theta \in \Theta$ and (Grad-OT) holds:

$$\nabla h_0(\theta) = \nabla_\theta H_0(\psi_*, \theta) = \mathbb{E}\left[D_\theta g(\theta, Z)^T \nabla \psi_*^c(g_\theta(Z))\right].$$

Proposition

Assume also that the input noise is integrable, that is, $\mathbb{E}[\|Z\|] < \infty$.
Hypothesis $G(\Theta)$ is true for g_θ a neural network with $\mathscr{C}^1$ and Lipschitz activation functions

Alternate algorithm for semi-discrete WGAN learning

The semi-discrete WGAN cost writes as

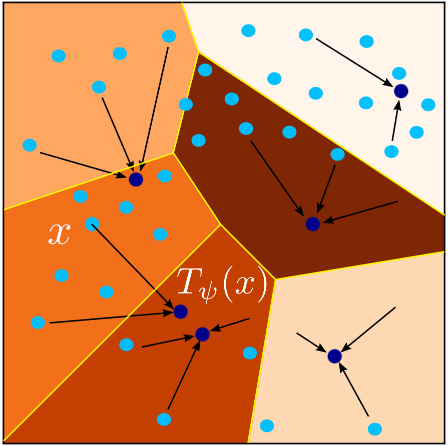
$$\min_{\theta} h(\theta) = \min_{\theta} \max_{\psi} H(\psi, \theta)$$

Initialization : θ (random)
For $n = 1, \dots, N$
- $\psi \approx \text{Argmax}\, H(\cdot, \theta)$ (ASGD)
- $\theta \approx \text{Argmin}\, H(\psi, \cdot)$ (ADAM)

Output: Model μ_θ

NB: Both steps rely on samples of μ_θ.

$$\nabla_\theta H(\psi, \theta) = \mathbb{E}\left[\nabla_\theta \left(\psi^c(g(\theta, Z))\right)\right],$$
$$\nabla \psi^c(x) = \nabla_x c(x, T_\psi(x)).$$

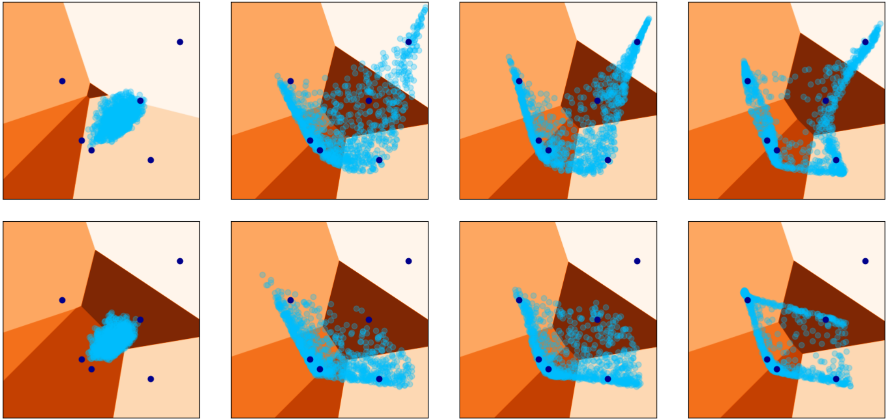


Dark blue: points of ν
Light blue: samples of μ_θ
Orange partition: Laguerre diagram of T_ψ

Example of semi-discrete WGAN

$K = 100$

$K = 2$



$n = 0$ $n = 50$ $n = 100$ $n = 200$

K : number of iterations in ASGD subloop

Comment: Semi-discrete WGAN learning is even more stable, but requires visiting the whole $\mathcal{Y}$ at each iteration.

Take-home Messages

SUMMARY AND COMMENTS:

- We introduced GANs and Wasserstein GANs
- Connection between Adversarial training and Dual expression of the loss
- Alternate algorithm for adversarial training
- Some constraints (Lipschitz) help to make training more stable
- Semi-discrete OT gives a parameterization of one dual variable by a *c*-transform. It makes training even more stable but is limited to relatively small datasets.
- Results also depend on the generator/discriminator architectures and the optimization strategy

✗ The adopted losses do not measure if the generated images are photo-realistic. How to assess the quality of a generative model for large-scale image synthesis?
$\rightarrow$ Let's discuss that next Tuesday! (among other things)

THANK YOU FOR YOUR ATTENTION!

References I

- Arjovsky, M., Chintala, S., and Bottou, L. (2017).
  Wasserstein generative adversarial networks.
  In *International Conference on Machine Learning*, pages 214–223.
- Biau, G., Cadre, B., Sangnier, M., and Tanielian, U. (2018).
  Some theoretical properties of gans.
  arXiv preprint arXiv:1803.07819.
- Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016).
  Stochastic optimization for large-scale optimal transport.
  In *Advances in neural information processing systems*, pages 3440–3448.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
  Generative adversarial nets.
  In *Advances in neural information processing systems*, pages 2672–2680.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017).
  Improved training of Wasserstein gans.
  In *Advances in neural information processing systems*, pages 5767–5777.

References II

Houdard, A., Leclaire, A., Papadakis, N., and Rabin, J. (2023).
On the gradient formula for learning generative models with regularized optimal transport costs.
Transactions on Machine Learning Research.

Kitagawa, J., Mérigot, Q., and Thibert, B. (2019).
Convergence of a newton algorithm for semi-discrete optimal transport.
Journal of the European Mathematical Society, 21(9):2603–2651.

Radford, A., Metz, L., and Chintala, S. (2016).
Unsupervised representation learning with deep convolutional generative adversarial networks.
In Bengio, Y. and LeCun, Y., editors, *Proceedings of ICLR*.

Santambrogio, F. (2015).
Optimal transport for applied mathematicians.
Birkäuser, NY.

Tsybakov, A. (2008).
Introduction to Nonparametric Estimation.
Springer Series in Statistics. Springer New York.

References III

Villani, C. (2009).
Optimal transport: old and new, volume 338.
Springer.