Génération d'images avec *Sliced Wasserstein*

Samuel Boïté, Mathias Grau

11 mars 2025

Résumé

Nous illustrons les résultats de *Tanguy* [4] sur la convergence de l'algorithme de descente de gradient stochastique (SGD) avec la perte *Sliced Wasserstein* (SW_2), en l'appliquant à des données synthétiques, à MNIST et à CelebA.

1 Introduction théorique

1.1 Contexte

En transport optimal, la distance de Wasserstein permet de comparer des distributions de probabilité en tenant compte de la géométrie de l'espace sous-jacent. Elle est définie comme suit pour deux distributions de probabilité μ et ν sur $\mathbb{R}^d$ ayant un second moment fini :

$$\mathrm{W}_2^2(\mu,\nu) \coloneqq \inf_{\gamma\in\Pi(\mu,\nu)} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|x-y\|^2 \mathrm{d}\gamma(x,y),$$

où $\Pi(\mu,\nu)$ est l'ensemble des mesures sur $\mathbb{R}^d\times\mathbb{R}^d$ ayant μ et ν comme marginales respectives. Cependant, le calcul de cette distance souffre de la malédiction de la dimension, en raison de la complexité en $\mathcal{O}(n^{-1/d})$ pour n échantillons.

Une alternative plus efficace est la distance de *Sliced Wasserstein*, introduite par *Rabin et al.* [3]. Cette méthode consiste à projeter les distributions selon des directions aléatoires, puis à calculer la distance de Wasserstein classique dans le cas particulier unidimensionnel. Plus précisément, pour deux mesures discrètes

$$\mu_X \coloneqq \frac{1}{n}\sum_{k=1}^n \delta_{x_k} \qquad \text{et} \qquad \mu_Y \coloneqq \frac{1}{n}\sum_{k=1}^n \delta_{y_k},$$

la distance de Wasserstein entre les projections de ces mesures sur une direction $\theta\in\mathbb{R}^d$ vaut

$$\mathrm{W}_2^2(P_\theta\#\mu_X, P_\theta\#\mu_Y) = \frac{1}{n}\sum_{k=1}^n \left(\theta^\top x_{\sigma(k)} - \theta^\top y_{\tau(k)}\right)^2,$$

où σ et τ sont des permutations de x et y respectivement, permettant de trier les projections de x et y dans l'ordre croissant. La distance *Sliced Wasserstein* est alors définie comme la moyenne de cette distance sur toutes les directions $\theta\in\mathcal{U}(\mathbb{S}^{d-1})$:

$$\mathrm{SW}_2^2(\mu_X,\mu_Y) \coloneqq \mathbb{E}_{\theta\sim\mathcal{U}(\mathbb{S}^{d-1})}\mathrm{W}_2^2(P_\theta\#\mu_X, P_\theta\#\mu_Y).$$

Cette distance, que l'on peut approcher par une méthode de Monte-Carlo, ne souffre pas de la malédiction de la dimension. Cependant, elle souffre d'un défaut appelé *projection*

complexity : dans des epsaces de grande dimension, la plupart des directions sont peu informatives, et seules certaines projections capturent véritablement les différences entre distributions.

Pour pallier cela, une extension naturelle consiste à considérer la direction la plus discriminante, i.e. celle qui maximise la distance de Wasserstein 1D. Cela définit une nouvelle distance, notée max-SW_2, définie par : [1]

$$\text{max-SW}_2(\mu_X, \mu_Y) \coloneqq \max_{\theta \in \mathbb{S}^{d-1}} \mathrm{W}_2^2(P_\theta \# \mu_X, P_\theta \# \mu_Y),$$

où μ_ω et ν_ω sont les projections des mesures μ et ν sur la direction ω.

En *deep learning*, on peut utiliser la distance SW pour entraîner des réseaux de neurones génératifs. L'objectif est de minimiser la perte $\mathrm{SW}_2^2(T_u \# \mu_X, \mu_Y)$, où $T_u \# \mu_X$ est l'image de la mesure μ_X par un réseau de neurones T_u, et μ_Y est la mesure cible. L'entraînement peut être effectué en utilisant l'algorithme de descente de gradient stochastique (SGD).

1.2 Résultats de convergence

Cette sous-partie synthétise l'approche et les résultats théoriques de Tanguy [4] sur la convergence de l'algorithme SGD sur la perte SW_2.

Lorsque l'on minimise une fonction F par descente de gradient, on suit en réalité une version discrétisée du *flot de gradient*

$$\dot{u}(s) = -\nabla F(u(s)).$$

Cependant, en *deep learning*, où les fonctions de perte peuvent être complexes et les activations non lisses, F n'est pas différentiable partout. On introduit alors la notion de sous-gradient de Clarke $\partial_C F(u)$, qui généralise la notion de gradient pour une fonction localement lipschitzienne : si F est différentiable en u, alors $\partial_C F(u) = \{\nabla F(u)\}$, et si F n'est pas différentiable en u, $\partial_C F(u)$ est un ensemble convexe non vide qui regroupe toutes les directions de descente « plausibles » obtenues par des approximations locales de F :

$$\partial_C F(x) = \left\{ \xi \in \mathbb{R}^n, \quad \langle \xi, v \rangle \leq \limsup_{y \to x, h \downarrow 0} \frac{f(y + hv) - f(y)}{h} \right\}.$$

Dans ce cadre non-lisse, on considère donc la discrétisation du flot de sous-gradient, c'est-à-dire la dynamique définie par l'inclusion différentielle

$$\dot{u}(s) \in -\partial_C F(u(s)).$$

L'algorithme SGD génère alors une suite de points $u(0)$, $u(1)$, ... via l'itération

$$u(t+1) = u(t) - \alpha \varphi\left(u(t), z(t+1)\right),$$

où $\varphi(u, z)$ représente une forme d'approximation du sous-gradient basé sur une perte $f(u, z)$ issue d'un échantillonnage. Pour comparer cette trajectoire aux solutions continues du flot de sous-gradient, on définit une trajectoire interpolée linéairement $u_\alpha(s)$ par

$$u_\alpha(s) = u(t) + \frac{s - t\alpha}{\alpha}\left(u(t+1) - u(t)\right) \qquad \text{pour } s \in [t\alpha, (t+1)\alpha[.$$

On définit alors l'ensemble $S_{-\partial_C F}(K)$ des trajectoires absolument continues qui satisfont le flot de sous-gradient $\dot{u}(s) \in -\partial_C F(u(s))$ pour presque tout s, et qui vérifient $u(0) \in K$, où K est un ensemble compact de conditions initiales.

On se donne alors les hypothèses suivantes :

1. La loi de l'*input* μ_X et les données μ_Y sont supposées avoir un support compact ;
2. Le réseau de neurones $T(u, x)$ est supposé $\mathcal{C}^2$ par morceaux pour x fixé et lipschitzien par rapport à u et x, et ses paramètres évoluent dans un domaine borné. En pratique, cela vaut pour la plupart des réseaux utilisés en pratique : les activations peuvent être $\mathcal{C}^2$-lisses (ex. sigmoïde, tanh...), semi-algébriques localement lipschitziennes (ex. ReLU, LeakyReLU...), ou polynomiales par morceaux.

Alors :

Théorème 1. *Si le* learning rate *diminue, les trajectoires de SGD se rapprochent des flots de sous-gradients, lesquels convergent à leur tour vers des points critiques de F.*

Pour mieux comprendre ce résultat, l'auteur précise que si F était $\mathcal{C}^2$ et avait un nombre fini de points critiques, alors on aurait la convergence de la solution $u(s)$ de cet *Algorithme 1* vers un point critique de F.

Pour obtenir un résultat de convergence encore plus fort, l'article considère une version modifiée du SGD où du *bruit additif* est ajouté à chaque itération, favorisant ainsi une meilleure exploration de l'espace, et où du *weight clipping* est appliqué pour garantir que les paramètres u restent bornés.

Sous ces hypothèses plus strictes, et en supposant également que la loi de l'*input* μ_X et celle des données μ_Y sont discrètes, ce qui est naturel pour y mais moins pour x :

Théorème 2. *Les trajectoires de ce SGD bruité et projeté convergent, lorsque le* learning rate *tend vers 0, vers un ensemble de points critiques généralisés $\mathcal{C}_r$ de F.*

Si l'on projette les paramètres du réseau sur $\bar{B}(0, r)$, $\mathcal{C}_r$ est en fait l'ensemble des points critiques de F, à l'exception du cas pathologique où les paramètres sont de norme $\|u\|_2 = r$, ce qui n'arrive jamais en pratique, pour peu que la borne r choisie sur les poids soit suffisamment grande. On a donc, en pratique, convergence de cet *Algorithme 2* vers des points critiques de la fonction de perte SW_2.

Ces résultats théoriques illustrent les observations pratiques selon lesquelles l'entraînement de réseaux Sliced-Wasserstein GAN avec SGD fonctionne bien.

1.3 Notre expérience

Afin d'illustrer ces propriétés de convergence, nous appliquerons le premier algorithme (SGD avec la perte SW_2) à une distribution synthétique en dimension 2 ainsi qu'au jeu de données MNIST, permettant ainsi d'évaluer empiriquement la convergence et la qualité des échantillons générés.

Pour des grands jeux de données (MNIST et CelebA), au lieu de calculer directement la distance SW_2 entre les images générées et réelles, nous l'évaluons sur les sorties des dernières couches du discriminateur. Ces couches produisent un ensemble de caractéristiques de plus haut niveau de l'image. Nous forçons ainsi le discriminateur à produire des représentations similaires pour les images réelles et les images *fake*. Cette approche est suggérée par Deshpande et al. et permet d'obtenir des résultats significativement meilleurs [2].

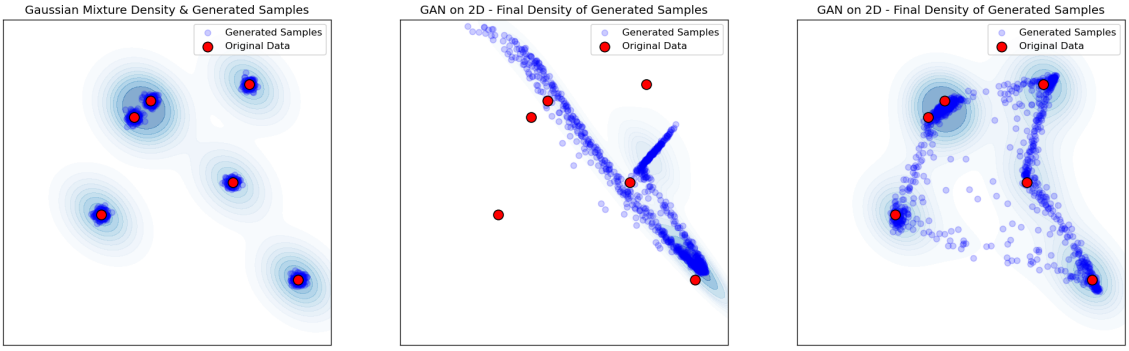
2 Simulation numérique

2.1 Jeu de données synthétique 2D

Nous commençons par un cas synthétique comme nous l'avions fait en cours : une distribution cible en dimension 2 pouvant représenter, par exemple, un mélange gaussien. Notre générateur $G_\phi(z)$ transforme un bruit $z \sim \mathcal{N}(0, I)$ en un échantillon dans $\mathbb{R}^2$.

Comparaison visuelle GAN vs. Sliced Wasserstein

Dans un cadre GAN standard, on min max pour entrainer de manière adversariale le générateur et le discriminateur, tandis qu'avec le *Sliced Wasserstein GAN* (SW-GAN), on remplace l'objectif du générateur afin que les images générées aient une distribution similaire à celle des images réelles grace à la fonction de perte $\mathrm{SW}_2^2(G_\phi \# \mu, \nu)$, approximée avec un certain nombre de directions aléatoires. Visuellement, on observe que l'approche SW a tendance à mieux couvrir l'ensemble des modes, avec moins de *mode collapse*.



(a) GMM cible (b) GAN standard (c) SW-GAN

FIGURE 2.1 – Expérience 2D : distribution cible vs. échantillons GAN standard et SW-GAN après entraînement (optimiseur SGD).

Impact de l'optimiseur (SGD vs Adam)

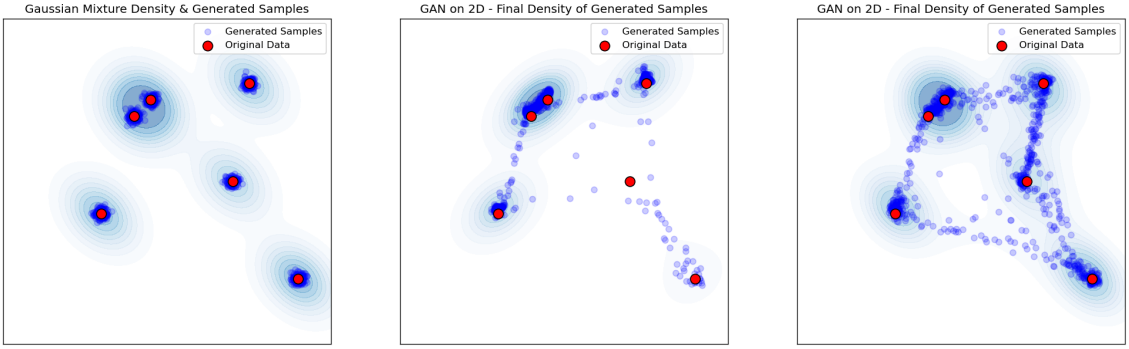
Nous avons reproduit l'expérience avec Adam comme optimiseur, ce qui peut accélérer la convergence, mais n'élimine pas pour autant les phénomènes de couverture incomplète des modes en GAN standard. En revanche, pour SW-GAN, on note une bonne répartition des points générés.

2.2 MNIST

Nous avons ensuite continué avec des données réelles : le jeu de données MNIST (28x28). Nous comparons à nouveau un GAN standard et l'approche *Sliced Wasserstein* (SW-GAN). On observe que pour un même réseau de neurone et exactement les mêmes paramètres et optimiseurs, le GAN standard est absolument incapable de générer des images, quel que soit l'optimiseur utilsé. A contrario, pour la perte SW évaluée sur la dernière couche du discriminateur, dont la formule est donnée par

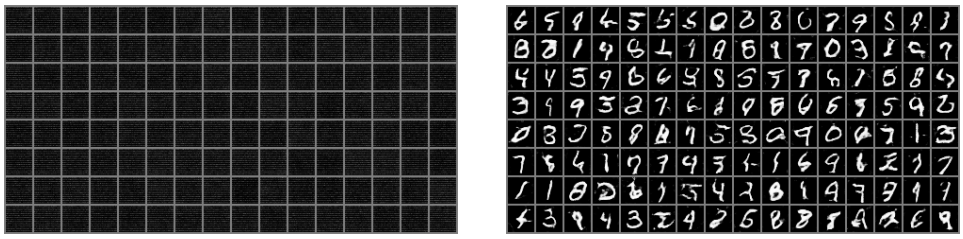
$$\mathrm{SW}_2\left(D(G(z)), D(x_{\text{réel}})\right), \qquad z \sim \mathcal{N}(0, I),$$

on observe des données générées très cohérentes sans *mode collapse*.



(a) GMM cible (b) GAN standard (c) SW-GAN

FIGURE 2.2 – Expérience 2D : comparaison avec Adam comme optimiseur.

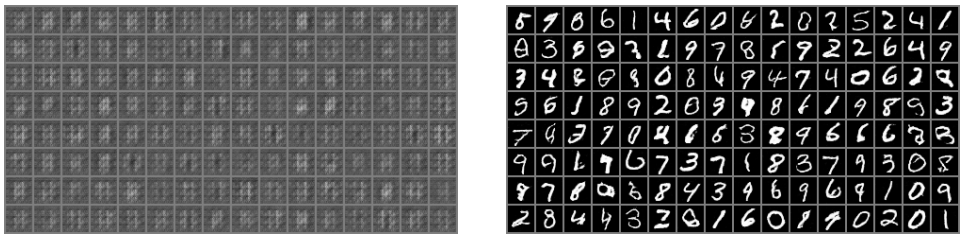


(a) GAN standard (SGD) (b) SW-GAN (SGD)

FIGURE 2.3 – Chiffres générés après plusieurs itérations de SGD (MNIST).

Résultats visuels

En utilisant Adam, la stabilité peut s'avérer meilleure dès le début, mais la tendance générale reste la même.



(a) GAN standard (Adam) (b) SW-GAN (Adam)

FIGURE 2.4 – Chiffres générés avec Adam (MNIST).

Notons toutefois que la comparaison entre le GAN et le SW-GAN a été réalisée en utilisant le même réseau de neurones, la même initialisation, les mêmes hyperparamètres et les mêmes optimiseurs. Une optimisation spécifiquement adaptée au GAN aurait peut-être permis d'obtenir de meilleurs résultats, mais au prix d'un temps d'entraînement nettement plus long et d'une qualité finale malgré tout inférieure.

2.3 CelebA et Max-SW

Enfin, nous appliquons la méthode sur le jeu de données CelebA (100 000 images de visages), de plus grande dimension et plus grande diversité. Ici, on teste plutôt la distance max-SW pour améliorer la discrimination entre les images. En pratique, on résoud à chaque

calcul de la distance un problème d'optimisation visant à maximiser la distance de Wasserstein 1D en la direction de projection ω.

(a) max-SW (SGD) (b) max-SW (Adam)

FIGURE 2.5 – Exemples de visages générés (CelebA) via une approche max-SW. L'optimiseur n'influence pas fondamentalement la qualité, mais la stabilité peut être améliorée par Adam.

En pratique, la distance max-SW se révèle parfois plus robuste pour distinguer des distributions très complexes en haute dimension. Par ailleurs, dans le cas de Celeba, la projection des gradients sur une boule de rayon fixé s'est avérée déterminante pour la qualité des images générées avec l'optimiseur SGD.

3 Conclusion

La distance *Sliced Wasserstein* constitue une alternative moins coûteuse que la distance de Wasserstein, et permet d'atteindre plus de stabilité qu'un GAN standard, avec moins de *mode collapse*. Les résultats de *Tanguy* [4] garantissent la convergence de l'algorithme SGD vers un ensemble de points critiques de la perte *SW* sous des hypothèses raisonnables (réseau lisse par morceaux, mesures bornées, etc.). Expérimentalement, on observe une bonne couverture des modes et une stabilité accrue, aussi bien sur des données synthétiques qu'en imagerie réelle (MNIST, CelebA).

Il reste à évaluer d'autres méthodes : le deuxième algorithme proposé avec bruit et *weight clipping*, les variantes $p \neq 2$ de la distance SW_p, et leur application à des tâches d'apprentissage plus complexes.

Le code est disponible sur GitHub : https://github.com/mathias-grau/Sliced-Wasserstein-GAN

Références

[1] Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander Schwing. Max-Sliced Wasserstein Distance and its use for GANs, April 2019. arXiv :1904.05877 [cs].

[2] Ishan Deshpande, Ziyu Zhang, and Alexander Schwing. Generative Modeling using the Sliced Wasserstein Distance, March 2018. arXiv :1803.11188 [cs].

[3] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein Barycenter and Its Application to Texture Mixing. In Alfred M. Bruckstein, Bart M. ter Haar Romeny, Alexander M. Bronstein, and Michael M. Bronstein, editors, *Scale Space and Variational Methods in Computer Vision*, pages 435–446, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[4] Eloi Tanguy. Convergence of SGD for Training Neural Networks with Sliced Wasserstein Losses, March 2024. arXiv :2307.11714 [cs].