GAN à travers le prisme des EBMs
DLLS : résultats, limites et réfléxions

Yanis FALLET

3 Janvier 2025

Résumé

Ce rapport propose une analyse approfondie de la méthode DDLS (Discriminatory Driven Latent Sampling), qui interprète les GANs (Generative Adversarial Networks) comme des modèles énergétiques (Energy-Based Models, EBM). Une idée introduite par des chercheurs de l'Université de Montréal et du Google Brain dans leur article *Your GAN is Secretly an EBM and You Should Use Discriminator Driven Latent Sampling*. La méthode DDLS est étudiée dans le cadre de l'amélioration des performances des GANs. Les limites inhérentes à cette approche sont examinées de manière critique, et des solutions innovantes sont proposées pour y remédier. Ce travail présente également une évaluation empirique de l'implémentation de cette méthode sur le dataset MNIST, mettant en évidence son efficacité ainsi que les perspectives d'amélioration.

Introduction

Les Generative Adversarial Networks (GANs) [1] se distinguent par leur capacité à accomplir des tâches variées telles que la génération d'images ou l'apprentissage semi-supervisé. Basés sur un jeu adversarial entre un générateur G et un discriminateur D, ces modèles produisent souvent des échantillons de haute qualité. Cependant, ils peuvent également générer des artefacts ou des échantillons non reconnaissables, en raison de la difficulté intrinsèque à modéliser des données en hautes dimensions et de la nature de l'optimisation adversariale.

Pour améliorer la qualité des échantillons, des techniques de rééchantillonnage telles que le *Discriminator Rejection Sampling* (DRS), le *Metropolis-Hastings GAN* (MH-GAN) et le *Discriminator Optimal Transport* (DOT) ont été proposées. Bien qu'ayant des avantages, ces méthodes exacerbent le mode collapsing ou nécessitent un coût computationel supplémentaire important.

Des travaux récents ont exploré une nouvelle perspective en interprétant les GANs comme des modèles énergétiques implicites (Energy-Based Models, EBM). Cette reformulation suggère que l'échantillonnage dans l'espace latent, par le biais de chaînes de Markov (Monte Carlo Markov Chain), pourrait contourner les limites des approches précédentes. Nous adoptons cette vision en proposant le Discriminator Driven Latent Sampling (DDLS), qui utilise le discriminateur pour corriger les biais du générateur et produire des échantillons de meilleure qualité.

1- Formalisme

1.1- Generative Adversarial Networks

Nous faisons abstraction des détails formels et du fonctionnement des GANs afin de nous concentrer sur l'essentiel de notre propos. Les Generative Adversarial Networks (GANs) constituent une classe de modèles d'apprentissage définis par un jeu minimax adversarial opposant un générateur G et un discriminateur D.

Les Wasserstein GANs (WGANs) [6] représentent une sous-catégorie de GANs qui, au lieu d'optimiser la distance de Jensen-Shannon, s'appuient sur la distance de Wasserstein-1. Cette approche présente plusieurs avantages, notamment une meilleure stabilité lors de l'entraînement.

1.2- Energy-Based Models et Langevin Dyanmics

Un modèle basé sur l'énergie (*Energy-Based Model*, EBM) est défini par une distribution de Boltzmann $p(x) = e^{-E(x)/Z}$, où $x \in X$ et X est l'espace des états, et $E(x): X \to R$ est la fonction d'énergie. Une fonction d'énergie associe à un état un niveau d'énergie qui est faible si cet état est probable et élevé si au contraire il est improbable. Les échantillons sont généralement générés à partir de $p(x)$ à l'aide d'un algorithme MCMC. Un algorithme MCMC courant dans les espaces d'états continus est la dynamique de Langevin, dont l'équation de mise à jour est :

$$x_{i+1} = x_i - \frac{\varepsilon}{2}\nabla_x E(x) + \sqrt{\varepsilon}n, \quad n \sim \mathcal{N}(0, I)$$

Une solution au problème de lenteur des chaînes de Markov consiste à effectuer l'échantillonnage dans un espace latent soigneusement conçu, en particulier dans l'espace latent.

L'intuition sous-jacente est que l'équation de la dynamique de Langevin met à jour x dans l'espace latent de manière à minimiser l'énergie. Autrement dit, elle affine x pour le rapprocher d'un état plus probable.

2- Résultats théoriques

2.1-GAN as EBM dans l'espace des pixels

Soit un GAN entraîné sur une distribution de données p_d , avec un générateur $G(z)$ produisant la distribution p_g et un discriminateur D(x). Nous supposons que p_d et p_g partagent le même support, ce qui peut être garanti par l'ajout de bruit gaussien aux deux distributions. L'entraînement des GANs est un jeu adversarial qui ne converge généralement pas vers un générateur optimal, de sorte que p_d et p_g ne correspondent pas parfaitement à la fin de l'entraînement.

Toutefois, le discriminateur fournit une estimation de cette erreur.
Supposons que le discriminateur soit proche de l'optimalité, c'est-à-dire $D(x) \approx \frac{p_d(x)}{p_d(x)+p_g(x)}$
En posant d(x) comme le logit de D(x) nous avons :

$$\frac{p_d(x)}{p_d(x)+p_g(x)} \approx \frac{1}{1+\exp\big(-d(x)\big)}$$

et donc $e^{d(x)} \approx \frac{p_d}{p_g}$ d'où $p_d(x) \approx p_g(x)e^{d(x)}$
La normalisation de $p_g(x)e^{d(x)}$ n'est pas garantie, rendant le modèle probabiliste invalide, d'où l'introduction de la constante de régularisation Z_0.
Ce modèle présente deux propriétés intéressantes : si $D = D^*$ (le discriminateur optimal), alors $p_d^* = p_d$, et il corrige le biais du générateur par un poids et une normalisation.

Le résultat important est que si nous parvenons à échantillonner à partir de cette distribution, cela devrait améliorer la qualité des échantillons générés. Cependant, deux problèmes persistent. L'échantillonnage nécessite l'utilisation d'un algorithme MCMC. Or, dans l'espace des images, cet algorithme devient très coûteux computationnellement (cf. dynamique de Langevin). De plus, l'expression explicite de la distribution induite par le générateur n'est pas connue.

2.2-GAN comme EBM dans l'espace latent

Pour résoudre ces deux problèmes, l'idée originale est d'introduire un EBM dans l'espace latent. En découle le théorème principale du papier :

Supposons que p_d soit la distribution génératrice des données et p_g la distribution du générateur induite par le générateur $G : Z \to X$, où Z est l'espace latent avec une distribution a priori $p_0(z)$. Définissons la distribution de Boltzmann $p_d^* = \frac{e^{\log p_g(x)+d(x)}}{Z_0}$, où Z_0 est une constante de normalisation.

Supposons que p_d et p_g aient le même support. De plus, soit $D(x)$ le discriminateur, et $d(x)$ le logit de D, tel que $D(x) = \sigma\big(d(x)\big)$. Nous définissons la fonction d'énergie $E(z) = -\log p_0(z) - d\big(G(z)\big)$et sa distribution de Boltzmann $p_t(z) = \frac{e^{-E(z)}}{Z}$. Alors, nous avons :

1. $p_d^* = p_d$ lorsque D est le discriminateur optimal.
2. Si $z \sim p_t$, et $x = G(z)$, alors $x \sim p_d^*$. Ainsi, on a que $G_\circ p_t = p_d^*$

Ce théorème est fondamental car il montre qu'en échantillonnant z à partir d'une distribution appropriée, la distribution induite par le générateur est égale à celle des données. De plus, étant donné que l'EBM est établi dans l'espace latent, l'algorithme MCMC sera beaucoup moins coûteux en termes de calculs.

2.3-WGAN comme EBM

Les Wasserstein GANs (WGANs) diffèrent des GANs classiques en ce qu'ils utilisent la fonction de perte de Wasserstein, ce qui confère aux WGANs une stabilité bien supérieure durant l'entraînement. Les fonctions objectifs des WGANs sont les suivantes :

$$L_D = E_{p_g}[D(x)] - E_{p_d}[D(x)] \quad , \quad L_G = -E_{p_\mathbb{0}}\big[D\big(G(z)\big)\big]$$

Le WGAN optimise approximativement la divergence Kullback-Leibler (KL) de $p_t = \frac{p_g(x)e^{-D(x)}}{Z}$ par rapport à p_d avec la contrainte que D est *K-Lipschistz*.
Cela suggère qu'on peut également effectuer une diffusion directe par descente de gradients (DDLS) dans l'espace latent du WGAN pour générer des échantillons améliorés, en utilisant une fonction d'énergie $E(z) = -\log p_0(z) - D\big(G(z)\big)$.

3- Implémentation

Sur la base des résultats théoriques présentés précédemment, nous avons dérivé une distribution dans l'espace latent, à partir de laquelle nous pouvons, théoriquement, améliorer la qualité des images générées par le générateur G, sans nécessiter de réentraînement du modèle GAN. Ce cadre permet ainsi de raffiner les échantillons générés sans reposer sur l'entraînement coûteux d'un GAN classique, ce qui en fait une approche intéressante pour optimiser les générateurs de manière plus efficace.

L'échantillonnage depuis cette distribution est réalisé en utilisant l'équation de la dynamique de Langevin. Avant de passer à l'inférence finale du générateur, il suffit d'exécuter l'algorithme pour un nombre défini d'itérations, notées N, afin d'affiner progressivement la position de z dans l'espace latent. Cela permet de déplacer z vers des zones plus « probables » ou plus adaptées à la génération d'images de qualité supérieure.

L'algorithme est relativement simple à mettre en œuvre (cf *pseudo-code*), et les hyperparamètres incluent N, qui correspond au nombre d'itérations de l'algorithme, et ϵ, la taille du pas de mise à jour qui contrôle la vitesse de convergence de l'échantillonnage. Ces hyperparamètres peuvent être ajustés pour trouver un compromis optimal entre la qualité des images générées et le coût computationnel.

Algorithm 1 Discriminator Langevin Sampling

Input: $N \in \mathbb{N}_+, \epsilon > 0$
Output: Latent code $z_N \sim p_t(z)$
Sample $z_0 \sim p_0(z)$.
for $i < N$ **do**
  $n_i \sim N(0,1)$
  $z_{i+1} = z_i - \epsilon/2\nabla_z E(z) + \sqrt{\epsilon} n_i$
  $i = i + 1$
end for

Intuitivement, si l'on imagine un espace latent gaussien de dimension 2 (cf. *schéma*), le processus commence par tirer un z initial aléatoirement dans cet espace. L'algorithme de Discriminator Langevin Sampling « déplace » ce z dans l'espace latent en fonction de la gradience de l'énergie associée à ce point, de manière à ce que ce dernier se rapproche d'une région de l'espace latent qui génère des images de meilleure qualité lorsqu'il est passé à travers le générateur G.

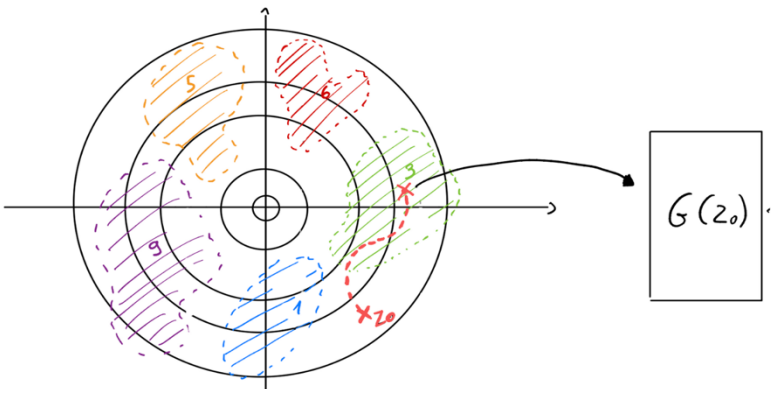


Schéma – Espace latent de dimension 2 et raffinement de z par l'algorithme DDLS

L'illustration (cf *images ci-dessous*) de cet algorithme montre clairement le déplacement de *z* dans l'espace latent, et par transposition à travers le générateur, l'évolution de l'image obtenue au fur et à mesure que le processus d'échantillonnage affine le vecteur z.

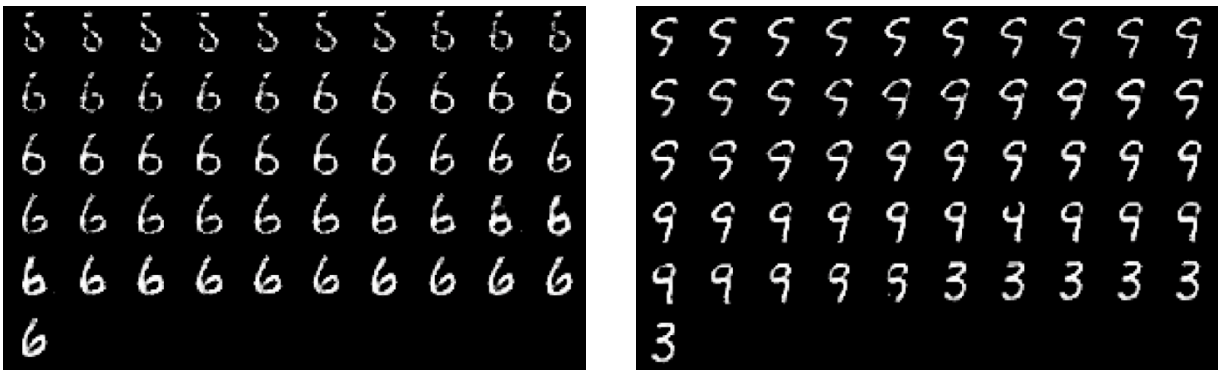


Illustration - évolution de l'image pendant le raffinement de z

4- Résultats, limites et solutions

Après l'implémentation de l'algorithme, nous avons observé un tradeoff precision recall important, c'est-à-dire que nous pouvons obtenir un haut degré de précision, mais au détriment du recall. Ce comportement est en fait attendu, car il fait partie intégrante du fonctionnement de l'algorithme. Toutefois, il est nécessaire de pouvoir ajuster les hyperparamètres de manière fine pour obtenir un meilleur équilibre entre ces deux métriques.

Malheureusement, il n'existe pas de solution évidente pour cela, et la recherche de la configuration optimale reste un défi.

La question qui se pose alors est la suivante : comment pouvons-nous atténuer ce compromis entre précision et rappel ? Quelles sont les causes possibles de cet excès de précision, et comment les résoudre ?

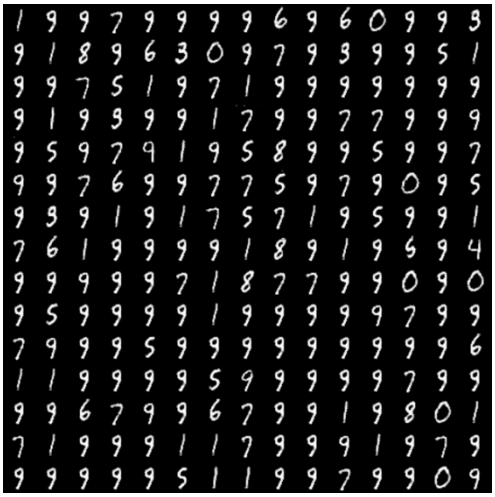


Illustration – Output WGAN-GP avec DDLS

Une explication possible réside dans le fait que notre algorithme dispose d'une trop grande liberté pour déplacer le vecteur *z* dans l'espace latent. Il se peut qu'il effectue trop de pas dans des dimensions non pertinentes, ce qui pourrait entraîner un surajustement dans l'espace latent, et par conséquent, une qualité d'échantillons moins diversifiée. Une approche pour remédier à ce problème pourrait être de contraindre le déplacement de *z* afin de prévenir une sur-optimisation excessive dans l'espace latent. Bien que cette idée

ne repose pas sur une démonstration mathématique formelle, elle s'appuie sur des intuitions et des ajustements pratiques.

Première idée : Restriction de la dimension

Pour mieux contrôler le mouvement de z, une stratégie consiste à restreindre le déplacement de z à certaines dimensions de l'espace latent, plutôt que de permettre un déplacement dans toutes les dimensions. Cela permettrait de limiter la possibilité pour z de « collapse » en un seul point de l'espace latent, ce qui pourrait résulter en une perte de diversité dans les échantillons générés.

Prenons l'exemple d'un espace latent en deux dimensions (cf. *schéma*). Si nous appliquons cette restriction, l'algorithme de Discriminator Langevin Sampling (DDLS) ne déplace z que dans un sous-espace vectoriel de l'espace latent, et non dans toutes les dimensions. En réduisant la liberté de mouvement dans l'espace latent, on évite que les vecteurs z ne convergent trop rapidement vers des régions spécifiques, ce qui permet de conserver une plus grande diversité dans les échantillons produits.

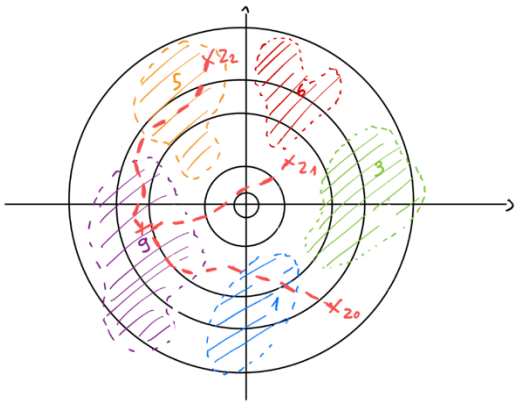


Illustration – Tous les points "collapse" en un point

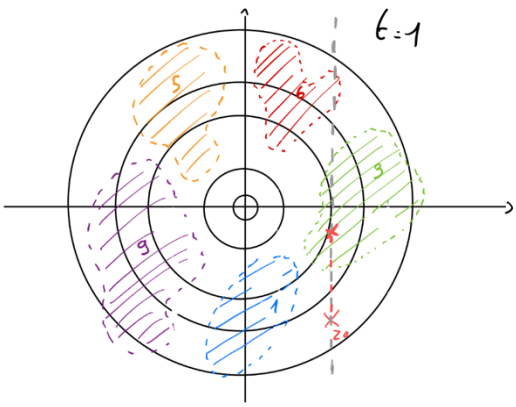


Illustration – Forcer DDLS à modifier qu'une dimension

Seconde idée : Pénalisation de la perte de diversité

Une autre approche pour atténuer le compromis entre precision et recall consiste à pénaliser la perte de diversité parmi les échantillons raffinés. En d'autres termes, à chaque itération de l'algorithme DDLS, si les vecteurs z finaux sont trop similaires entre eux, une pénalité sera introduite dans la fonction d'énergie afin de favoriser la diversité dans l'espace latent.

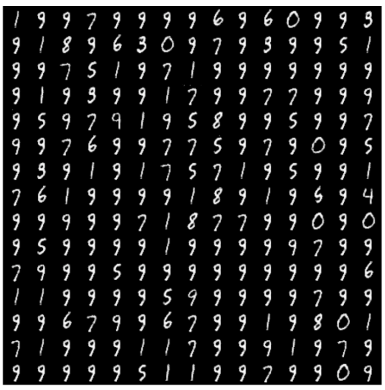
Cette pénalisation peut être formulée comme suit :

Pseudo code

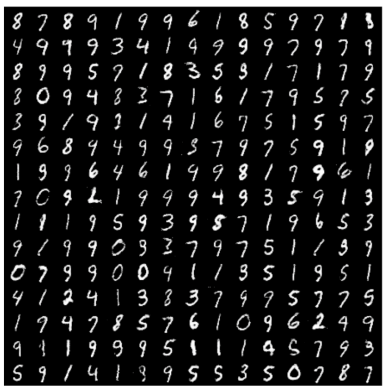
```
# Étape 1 : Calculer la moyenne des distances entre toutes les paires de points
# zs : liste des z raffinés après chaque itération
moyenne_distance_paire = moyenne(des_distances_2à2(zs, norme=2))

# Étape 2 : Calculer la perte de diversité en utilisant l'inverse de la moyenne des distances
perte_diversite = coefficient_reg_diversite / (moyenne_distance_paire + une_valeur_tres_petite)

# Étape 3 : Ajouter la perte de diversité à l'énergie totale
energie_totale += perte_diversite
```

Résultats

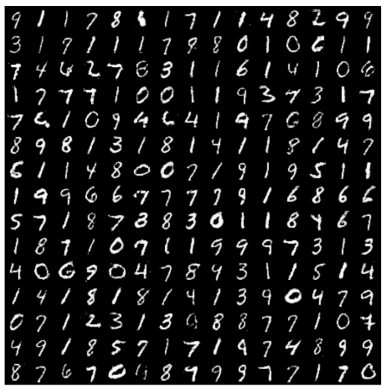
Ainsi, nous remarquons bien qu'en combinant la **restriction de dimension** et la **pénalisation de la perte de diversité**, nous parvenons à atténuer efficacement la perte de rappel associée à la méthode DDLS



WGAN-GP avec DDLS dim = 100



WGAN-GP DDLS dim = 82



WGAN-GP DDLS dim = 82 Pénalisation de la diversité

5- Conclusion

La méthode DDLS répond bien à son objectif principal, à savoir l'amélioration de la qualité des échantillons générés. Cependant, cette amélioration s'accompagne d'une perte notable en diversité. Bien que des solutions puissent être mises en place pour atténuer cet effet, il est important de souligner que ces ajustements ne résolvent pas complètement le problème. En conséquence, DDLS peut être particulièrement utile dans des contextes où la priorité est de produire des échantillons de très haute qualité, au détriment de la diversité.

6- Références

[1] Goodfellow, Ian, et al. ”Generative adversarial nets.” Advances in neural information processing systems 27 (2014). Disponible à l’adresse : https://arxiv.org/pdf/1406.2661