Wasserstein GAN

Othmane Marfoq
othmane.marfoq@ens-paris-saclay.fr

Souhaib Attaiki
souhaib.attaiki@ens-paris-saclay.fr

Abstract

Les réseaux génératifs adversarial (GAN) [1] constituent une classe puissante de modèles génératifs qui traite la modélisation générative comme un jeu entre deux réseaux: un réseau générateur qui produit des données synthétiques en fonction d'une source de bruit et un réseau discriminant qui établit une distinction entre la sortie du générateur et les données réelles. Les GANs peuvent produire des échantillons très attrayants sur le plan visuel, mais ils sont souvent difficiles à entraîner. Dans leur article [2], *Arjovsky et al.* propose Wasserstein GAN (WGAN) pour remédier à ce problème.

1 Introduction

Les réseaux génératifs adversarial (GAN) sont un exemple de modèles génératifs. Le terme «modèle génératif» fait référence à tout modèle prenant un ensemble d'apprentissage, constitué d'échantillons tirés d'une distribution p_{data}, et apprenant à représenter une estimation de cette distribution. Le résultat est une distribution de probabilité p_{model}. Dans certains cas, le modèle estime explicitement p_{model}, tandis que dans d'autres cas, le modèle ne peut générer que des échantillons à partir de p_{model}. Certains modèles peuvent effectuer les deux. Les GANs se concentrent principalement sur la génération d'échantillons, bien qu'il soit possible de concevoir des GAN capables de faire les deux. Pourtant les GANs sont souvent difficiles à entraîner, et une grande partie des travaux récents sur le sujet a été consacrée à la recherche de moyens de stabiliser leur entraînement.

En particulier, *Arjovsky et al.* introduisent un nouvel algorithme appelé WGAN, une alternative aux réseaux génératifs adversarial traditionnelle. Dans ce nouveau modèle, ils montrent que la stabilité de l'apprentissage peut être améliorer, et qu'il est possible d'éliminer des problèmes tels que l'effondrement des modes (mode collapse) et fournir des courbes d'apprentissage utiles pour le débogage et les recherches des hyperparamètres. De plus, ils montrent que le problème d'optimisation correspondant a des bonnes caractéristiques.

Dans ce rapport, on commence par la définition des GANs et on explore leur application et leurs limites dans la **section 1.**, ensuite on introduit l'algorithme proposé par *Arjovsky et al.*, et on montre son intérêt théorique sur un exemple dans la **section 2.**. Dans la **section 3.** on étudie la convergence et les résultats de cet algorithme pour la génération d'images sur deux jeux de données: MNIST et CelebA[3] et on va les comparer avec une implémentation classique des GANs. Finalement, dans la **section 5.** ont discutera des limites de WGAN, et on introduira une version amélioré de cet algorithme (Gradient penalty[4]) avant de conclure.

2 GAN: Generatif Advarsial Network

2.1 Principe

Dans [1], *Goodfellow et al.* proposent un nouveau cadre pour estimer les modèles génératifs via un processus adversial, formé simultanément de deux modèles: un modèle génératif G qui capture la distribution des données et un modèle discriminant D qui estime la probabilité qu'un échantillon provienne des données d'apprentissage réelle et non pas généré par G. La procédure d'entraînement

de G consiste à maximiser la probabilité que D commette une erreur. Ce cadre correspond à un jeu minimax à deux joueurs. Dans l'espace des fonctions arbitraires G et D, une solution unique existe, G reproduisant la distribution des données d'apprentissage et D égal à 1/2 partout. Dans le cas où G et D sont définis par des réseaux de neurones, il peuvent être appris en utilisant la backpropagation.

L'algorithme proposé par *Goodfellow et al.* est simple à appliquer lorsque les modèles G et D sont tous deux des réseaux de neurones. Pour connaître la distribution du générateur p_g sur les données x, on définit un prior sur les variables de bruit en entrée $p_z(z)$, puis on représente une application sur l'espace de données sous la forme $G(z;\theta_g)$, où G est une fonction différentiable représentée par un réseaux de neurones avec des paramètres θ_g. On définit également un deuxième réseaux de neurones $D(z;\theta_d)$ qui produit un seul scalaire. $D(x)$ représente la probabilité que x provienne des données plutôt que de p_g. D est entraîné pour maximiser la probabilité d'attribuer la bonne étiquette à la fois aux exemples générés par G et aux exemples prévenant du vrai jeu de donnée. G est entraîné à la minimisation du $\log(1 - D(G(z)))$. En d'autres termes, D et G jouent au jeu minimax suivant à deux joueurs avec la fonction de valeur V (G, D):

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}}\left[\log D(x)\right] + \mathbb{E}_{z_z}\left[\log(1 - D(G(z)))\right]$$

Ils proposent ainsi **Algorithm 1.** pour résoudre ce problème d'optimisation

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets [1]

for Nombre d'itérations d'entraînement **do**
  for k étapes **do**
    Échantillonner $\left\{z^{(1)}, z^{(2)}, ..., z^{(m)}\right\}$ selon p_z
    Échantillonner $\left\{x^{(1)}, x^{(2)}, ..., x^{(m)}\right\}$ selon p_{data}
    Mettre à jour le discriminateur D

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right]$$

  end for
  Échantillonner $\left\{z^{(1)}, z^{(2)}, ..., z^{(m)}\right\}$ selon p_z
  Mettre à jour le générateur G

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$$

end for

2.2 Applications

On peut légitimement se demander pourquoi les modèles génératifs méritent d'être étudiés, en particulier les modèles génératifs qui ne peuvent générer que des données plutôt que de fournir une estimation de la fonction de densité. On présente dans le paragraphe suivant, quelque rasions qui peuvent motiver l'étude de tels modèles:

- Entraîner et générer à partir de modèles génératifs constituent un excellent test de notre capacité à représenter et à manipuler des distributions de probabilité de grandes dimensions. Les distributions de probabilité de grande dimension sont des objets importants dans une grande variété de domaines des mathématiques appliquées et de l'ingénierie.
- Les modèles génératifs peuvent être entraîner avec des données manquantes et peuvent fournir des prévisions sur les entrées pour lesquelles des données sont manquantes. Un cas particulièrement intéressant des données manquantes est l'apprentissage semi-supervisé, dans lequel les étiquettes pour beaucoup ou même la plupart des exemples sont manquantes. Les algorithmes modernes d'apprentissage profond nécessitent généralement un très grand

nombre d'exemples étiquetés pour pouvoir bien généraliser. Les modèles génératifs, et les GAN en particulier, sont capables d'effectuer un apprentissage semi-supervisé de manière satisfaisante.

- Super-résolution: Dans cette tâche, l'objectif est de prendre une image en basse résolution et de synthétiser un équivalent en haute résolution. Les modèles génératives sont requis car cette tâche nécessite que le modèle impute plus d'informations dans l'image que ne le faisait initialement l'entrée.
- Les applications de traduction image à image peuvent convertir des photos aériennes en cartes ou convertir des esquisses en images. Il existe une très longue file d'applications créatives difficiles à anticiper mais utiles une fois qu'elles ont été découvertes.
- Génération de l'art: Plusieurs projets récents ont démontré que les modèles génératifs, et en particulier les GANs, peuvent être utilisé pour créer des programmes interactifs qui aident l'utilisateur à créer des images réalistes correspondant à des scènes approximatives de son imagination.

2.3 Limites

La plupart des GANs soufrent de plusieurs problèmes:

- Instabilité et Non-convergence: Le paramètre du modèle oscille et ne converge jamais. Cela peut être du principalement au non équilibre en terme d'entraînement entre le générateur et le discriminateur. La solution évidente de ce problème et d'équilibre leur entraînement mais pour le moment, aucune méthode n'a montré son efficacité pour traiter ce problème.
- Mode collapse: Le générateur produit des variétés limitées d'échantillons.
- Vanishing Gradient: Le discriminateur apprend plus vite que le générateur. Ce qui fait que le générateur ne peut pas apprendre.

3 Wasserstein GAN

3.1 Différentes Distances

Les auteurs de [2] présume que la distance (au sens des distributions) que les GANs classiques ne sont pas forcément continus par rapport aux paramètres du générateur. Il s'intéressent dans leur article au mesure de la distance $\rho\left(\mathbb{P}_\theta, \mathbb{P}_r\right)$ entre la distribution du modèle et la vrai distribution provenant des données réelles. L'étude des distances entre les distributions des probabilité est motivé par le fait, que les GANs en général essaye de minimiser une telle distance. Ainsi le bon choix de celle-ci aura un impact sur la convergence et sur le rendu visuelle des images générées.

Dans cette section, on étudie plusieurs mesure de distance entre les distributions de probabilité, et on montrera sur un exemple, la motivation du choix de la distance *Earth-Mover* (ou Wasserstein-1) introduite par *Arjovsky et al.*:

- *Total Variation* (**TV**) distance
$$\delta(\mathbb{P}_r, \mathbb{P}_g) = \sup_A \left|\mathbb{P}_r(A) - \mathbb{P}_g(A)\right|$$
- *Kullback-Leibler* divergence
$$KL(\mathbb{P}_r \| \mathbb{P}_g) = \int \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x) d\mu(x)$$
Il est à noter que KL divergence n'est pas symétrique et peut être infini si $P_g(x) = 0$ et $P_r(x) > 0$
- *Jensen-Shannon* (JS) divergence
$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \| \mathbb{P}_m) + KL(\mathbb{P}_g \| \mathbb{P}_m)$$
Avec $\mathbb{P}_m = (\mathbb{P}_r + \mathbb{P}_g)/2$ Contrairement à KL divergence, JS divergence est symétrique et toujours fini.

- *Eart-Mover* (EM) distance ou Wasserstein-1

$$W(\mathbb{P}_r\,\mathbb{P}_m)) = \inf_{\gamma\in\Pi(\mathbb{P}_r\,\mathbb{P}_m))} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x-y\|\right]$$

  Où $\Pi\left(\mathbb{P}_r\,\mathbb{P}_m\right)$ est l'ensemble des distributions jointes $\gamma(x,y)$ dans les deux lois marginales sont respectivement $\mathbb{P}_r$ et $\mathbb{P}_g$. *Eart-Mover* distance entre $\mathbb{P}_g\,\mathbb{P}_r$ peut être définit informellement comme le coût minimum de transport de masse afin de transformer la distribution $\mathbb{P}_r$ en une distribution $\mathbb{P}_g$ (où le coût est la masse multipliée par la distance de transport).

Dans le paragraphe suivant, on propose une suite de distribution qui converge selon la distance EM, et qui divergent suivant les autres distances (KL-divergence, JS-divergence, TV distance).

Soit $Z\sim\mathcal{U}([0,1])$ et $\mathbb{P}_0$ la distribution du couple $(0,Z)\in\mathbb{R}^2$, et considérons $g_\theta(z)=(\theta,z)$ où θ est un paramètre scalaire. Un simple calcul montre que:

- $W(\mathbb{P}_0,\mathbb{P}_\theta)=|\theta|$
- $JS(\mathbb{P}_0,\mathbb{P}_\theta)=\begin{cases}\log 2 & \text{if } \theta\neq 0\\ 0 & \text{if } \theta=0\end{cases}$
- $KL(\mathbb{P}_\theta\|\mathbb{P}_0)=-KL(\mathbb{P}_0\|\mathbb{P}_\theta)=\begin{cases}+\infty & \text{if } \theta\neq 0\\ 0 & \text{if } \theta=0\end{cases}$
- $\delta(\mathbb{P}_0,\mathbb{P}_\theta)=\begin{cases}1 & \text{if } \theta\neq 0\\ 0 & \text{if } \theta=0\end{cases}$

On voit donc que lorsque $\theta\to 0$, $W(\mathbb{P}_0,\mathbb{P}_\theta)\to 0$, tandis que $\mathbb{P}_\theta$ ne converge pas ou diverge pour les autres distances. Il est à noter également que $W(\mathbb{P}_0,\mathbb{P}_\theta)$ est continu en fonction en fonction de θ et différentielle presque partout. Ce résultat est un résultat général comme le prouve le **Théorème 1.** de [2]. Un corollaire de ce théorème affirme que si g_θ est un réseau de neurones paramétré par θ, et si $p(z)$ est intégrable (i.e. $\mathbb{E}_{z\sim p(z)}[\|z\|]<\infty$) ce qui est le cas si on choisit que z soit un bruit gaussien ou distribué uniformément sur un compact, alors $W\left(\mathbb{P}_r,\mathbb{P}_\theta\right)$ est continue et dérivable presque partout. De plus ce résultat n'est pas vérifié par les autres distances proposées. En effet, le **Théorème 2.** de [2] compare les topologies induite par les différents ditances propsées et montre que la topologie induite par W et la plus faible, que les topologie induite par KL-divergence est la plus fort, que TV distance et JS-divergence sont équivalente, que la topologie induite par *Earth-Mover* distance et la plus faible. Cela veut dire que la convergence d'une suite de distribution pour toute distance (parmi ce proposée) implique la convergence de cette distance par *Earth-Mover* distance. De plus ce théorème établie une équivalence entre la convergence selon cette distance et la convergence en distribution des variables aléatoires.

3.2 WGAN Algorithme

Le **Théorème 2.**, même si il établie des résultats théorique de convergence selon la distance EM, il ne proposent pas d'algorithme pour trouver cette limite. Les auteurs de [2] proposent d'utiliser la dualité de Kantorovich-Rubinstein [5] afin de trouver une approximation d'*Earth-Mover* distance

$$W(\mathbb{P}_r,\mathbb{P}_\theta)=\sup_{\|f\|_L\leq 1} E_{x\sim\mathbb{P}_r}[f(x)]-E_{x\sim\mathbb{P}_\theta}[f(x)] \tag{1}$$

Un résultat plus général donne

$$W(\mathbb{P}_r,\mathbb{P}_\theta)=\frac{1}{K}\sup_{\|f\|_L\leq K} E_{x\sim\mathbb{P}_r}[f(x)]-E_{x\sim\mathbb{P}_\theta}[f(x)] \tag{2}$$

Le **Théorème 3.** de [2] montre que le problème d'optimisation (1) admet une solution f et que

$$\nabla_\theta W(\mathbb{P}_r,\mathbb{P}_\theta)=-\mathbb{E}_{z(z)}\left[\nabla_\theta f\left(g_\theta\left(z\right)\right)\right]$$

Cela donne une méthode pour résoudre $\min_\theta W(\mathbb{P}_r,\mathbb{P}_\theta)$, en utilisant une méthode de descente de gradient (SGD, RMSProp, ADAM...).

Il faut maintenant trouver une méthode efficace pour résoudre le problème d'optimisation (2), pour cela on se propose de cherche d'approcher f par un réseau de neurone f_ω paramétré par ω. Pour imposer à f_ω d'être lipshitzienne, on impose ω d'appartenir à un compact $\mathcal{W}$. Pour imposer l'appartenance de ω à un compact, les auteurs de [2] propose un méthode simple consistant à serrer les poids ω dans un ensemble de la forme $\mathcal{W} = [-c, c]^l$ après chaque mise à jour du gradient.

En regroupant tous ces résultat, *Arjovsky et al.* proposent l'**Algorithme 2.** pour approcher la distribution $\mathbb{P}_r$. Leur algorithme est proche de celui proposé dans [1]: la formule de mise à jour est modifier, car la distance qu'on minimise a changé, et il ajoute le clipping pour forcer les paramètres ω à vivre dans un compact, pour garantir la lipshitziannité de f_ω .

Algorithm 2 WGAN [2]

Require: α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: ω_0, initial critic parameters. θ_0, initial generator's parameters.
  while θ has not converged **do**
    for t = 0,...,n_{critic} **do**
      Sample $\left\{x^{(i)}\right\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
      Sample $\left\{z^{(i)}\right\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
      $g_w \leftarrow \nabla_\omega \left[\frac{1}{m} \sum_{i=1}^{m} f_\omega \left(x^{(i)} \right) \frac{1}{m} \sum_{i=1}^{m} f_\omega \left(g_\theta \left(z^{(i)} \right) \right) \right]$
      $\omega \leftarrow \omega + \alpha$ RMSProp(ω, g_w)
      $\omega \leftarrow \text{clip}(\omega, -c, c)$
    end for
    Sample $\left\{z^{(i)}\right\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f_\omega \left(g_\theta \left(z^{(i)} \right) \right)$
    $\theta \leftarrow \theta - \alpha$.RMSProp$(\theta, g_\theta)$
  end while

4 Résultats

Dans cette section, on explore le comportement du modèle proposé par [2]. Pour cela on compare les résultats (image générées et courbe de convergence) obtenue par WGAN [2] et GAN [1] sur les deux dataset de références: MNIST et celebA. Les différentes expériences sont disponibles sur notre github [6].

4.1 MNIST

MNIST est un jeux de données d'image de chiffre manuscrits, chaque image est une image en niveau de gris et contient 28×28 pixels. On utilise ce dataset comme premier test pour WGAN, vu la facilitée relative d'estimer la distribution des images de ce dataset. On a utilisé une architecture convolutionelle (similaire à celle propoosé en TP5), et on l'a entraînée avec les deux algorithme: GAN et WGAN. La **Figure 1.** montre la courbe de convergence pour les deux algorithmes. On voit bien que la convergence pour WGAN est plus stable.

Voire *Annexe A.* pour les résultats visuelles de ces deux algorithmes.

4.2 CelebA

CelebFaces Attributes Dataset (celebA) est un jeu de données d'attributs de visage à grande échelle avec plus de 200 000 images de célébrités, chacune avec 40 annotations d'attributs. Les images de cet ensemble de données couvrent de grandes variations de pose et d'encombrement en arrière-plan. CelebA possède de grandes diversités, de grandes quantités et de riches annotations, notamment

Figure 1: GAN

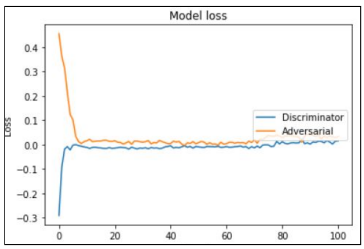


Figure 2: WGAN

Figure 3: MNIST: generator and critic loss vs epochs

- 10177 nombre d'identités
- 202 599 nombre d'images de visage

Contrairement à MNIST, la distibution de ce dataset est plus difficle à estimer, vu d'abbord la taille des images et leur diversité. On a commencé d'abord par réduire la taille des images à 64×64, pour faciliter le calcul vu qu'on était limité en ressource. On a essayé différentes architectures avec ce dataset. On a également essayer diférentes valeur de clipping pour et de nombre de critic pour améliorer le rendu visuelle et la convergence de WGAN avec ce dataset. La **Figure 2.** montre la courbe de convergence pour ces deux algorithmes. On peut constater que les images générées par l'algorithme classique sont plus belle visuellement, ceci est du fait que ce dernier peut être entraîner rapidement contrairement à WGAN, et vu nos ressources de calculs limitées, on a pas pu entraîner WGAN jusqu'au bout.

Voire *Annexe B.* pour les résultats visuelles de ces deux algorithmes.

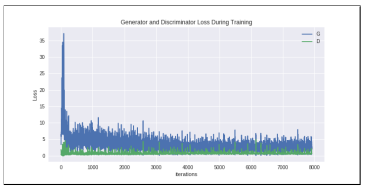


Figure 4: GAN

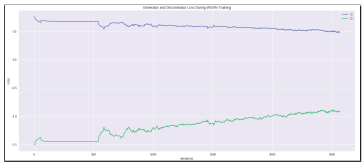


Figure 5: WGAN

Figure 6: celebA: generator and critic loss vs epochs

5 Discussion

Les auteurs de [4] explore les problèmes potentiels du à l'utilisation du clipping des poids utilisée par [2] pour imposer la lipshitzainnité de f_ω, et ils ont montré sur plusieurs exemples que peut entraîner le phénomène du vanishing ou exploding gradient, si la valeurs de c (clipping) n'est pas bien choisi. Il montre également, que l'utilisation du clipping biaise le discriminateur (critic) qui a tendance a apprendre des fonctions simples avec cette méthode. Pour remédier à ce problème, ils proposent de pénaliser la distance d'*Eart-Mover* par un terme dépendant du gradient du discriminateur, et montre que ce modèle pénalisé donne des meilleurs résultats par rapport à l'approche proposée dans [2].

6 References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Nets (2014).

[2] Martin Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein GAN (2017).

[3] Ziwei Liu and Ping Luo and Xiaogang Wang and Xiaoou Tang. Deep Learning Face Attributes in the Wild. *Proceedings of International Conference on Computer Vision (ICCV)*. December, 2015

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville. Improved Training of Wasserstein GANs. December, 2017

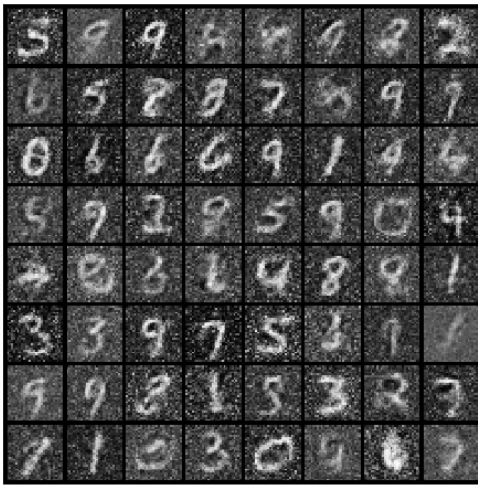
[5] Cédric Villani. Optimal Transport: Old and New. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009.

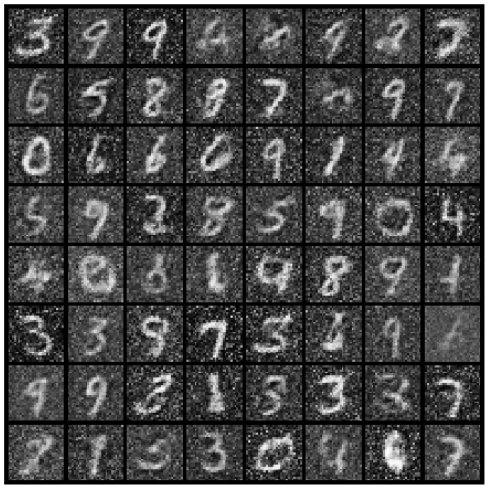
[6] https://github.com/marfoq/Wasserstein-GAN

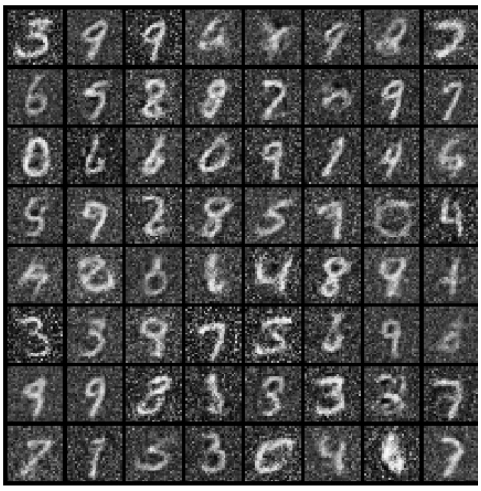
Annexe

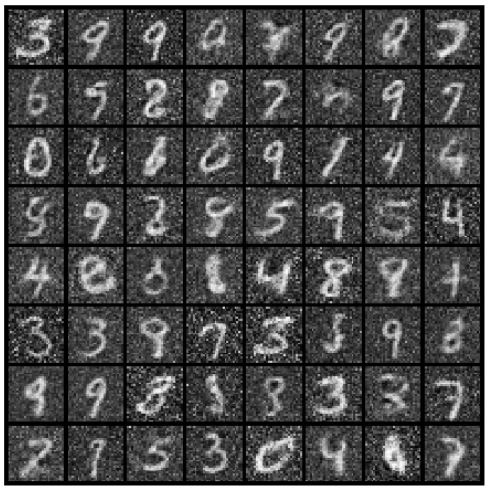
Annexe A: Résultats pour MNIST

GAN normal

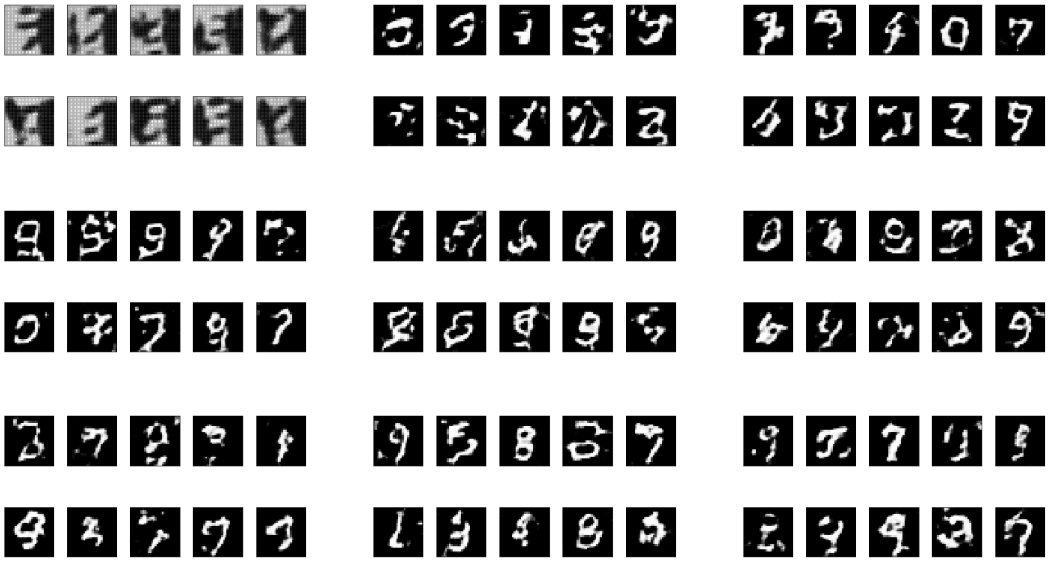




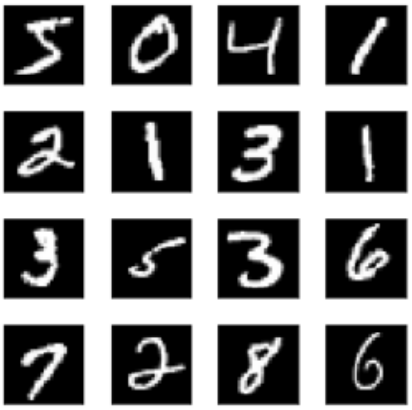




WGAN

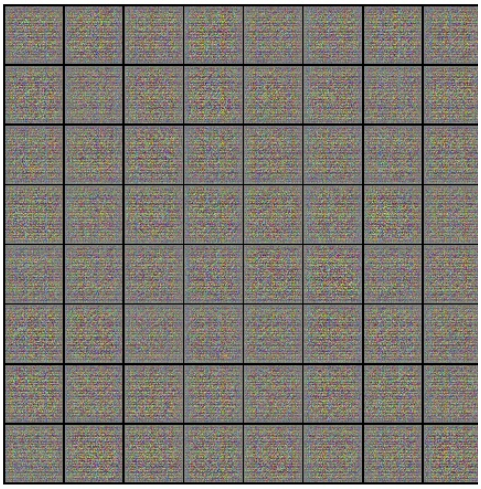


Final Result



Résultats: celebA

GAN normal



WGAN