



Rapport Generative Adversarial Networks

RDFIA - TP 9-10

Etudiant :

DJEGHRI Amine

MAMOU Idles

Numéro :

3801757

3803676

Janvier 2021

Partie 1 – Generative Adversarial Networks

1.1 Principe général

1-

$$\min_G \max_D \mathbb{E}_{\mathbf{x}^* \in \mathcal{D}_{\text{Data}}} [\log D(\mathbf{x}^*)] + \mathbb{E}_{\mathbf{z} \sim P(\mathbf{z})} [\log (1 - D(G(\mathbf{z})))] \quad (5)$$

$$\max_G \mathbb{E}_{\mathbf{z} \sim P(\mathbf{z})} [\log D(G(\mathbf{z}))] \quad (6)$$

$$\max_D \mathbb{E}_{\mathbf{x}^* \in \mathcal{D}_{\text{Data}}} [\log D(\mathbf{x}^*)] + \mathbb{E}_{\mathbf{z} \sim P(\mathbf{z})} [\log (1 - D(G(\mathbf{z})))] \quad (7)$$

Interprétation des équations 6 et 7 :

L'équation 6 : le générateur doit maximiser la probabilité que le Discriminateur se trompe et classe les données générées comme étant des données réelles.

L'équation 7 : Le discriminateur doit maximiser la probabilité qu'il classe les données réelles et générées dans leurs catégories respectives. Maximiser $\log D(\mathbf{x})$ des données réelles pour que la probabilité soit proche de 1, et maximiser $\log (1 - D(G(\mathbf{z})))$ ce qui implique la minimisation de la probabilité que les données générées sont fausses afin qu'elle soit proche de 0.

Si on utilisait seulement une des deux, on entraînerait seulement un des deux réseaux :

- On entraînerait seulement le Discriminateur D si on utilisait l'équation 7
- On entraînerait seulement le Générateur G si on utilisait l'équation 6

- 2- Idéalement, le générateur G transforme la distribution $P(\mathbf{z})$ en une distribution qui est égale à la distribution réelle des données $P(\mathbf{x})$
- 3- La "vraie" équation ici est :

$$\min_G \mathbb{E}_{\mathbf{z} \sim P(\mathbf{z})} [\log (1 - D(G(\mathbf{z})))]$$

1.2 Architectures des réseaux



Figure 1 – Visages générés après 0 , 200 et 8000 itérations avec les paramètres par défaut

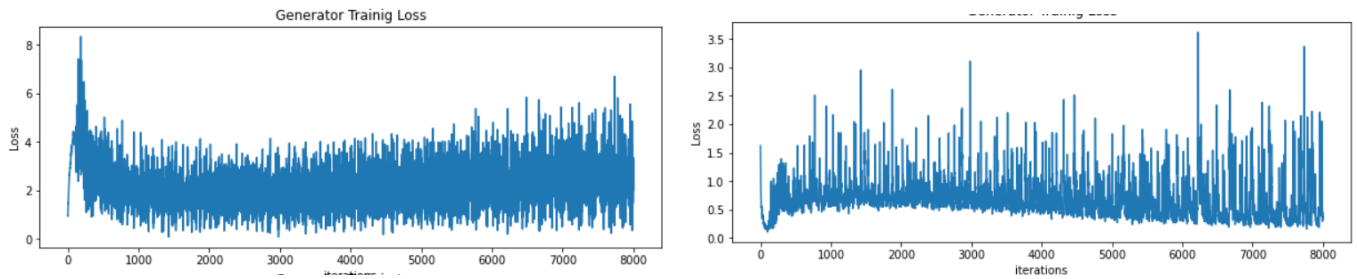


Figure 2 – L'évolution de la loss du générateur (à droite) et discriminateur (à gauche)

4- Commentaires sur l'apprentissage du GAN :

Les résultats des figures 1 et 2 sont obtenus en utilisant les paramètres par défaut.

Les visages générés avec ces hyperparamètres par défaut sont corrects et satisfaisants ceci après quelques centaines d'itérations seulement. On note également que les visages montrent des hommes et des femmes avec des races différentes, et un âge différent aussi ce qui montre la diversité des visages générés par le générateur. Les visages commencent à se former dès la 200ème itération (figure 1 image2), et continuent de devenir de plus en plus correctes et réalistes au cours de l'apprentissage jusqu'à la 8000ème itération où on obtient les résultats présentés dans la figure 1 image 3. Toutefois, les images des visages générés ne sont pas toutes correctes à cause de l'instabilité du modèle, et par conséquent on peut obtenir des images avec des visages déformés.

L'évolution de la loss :

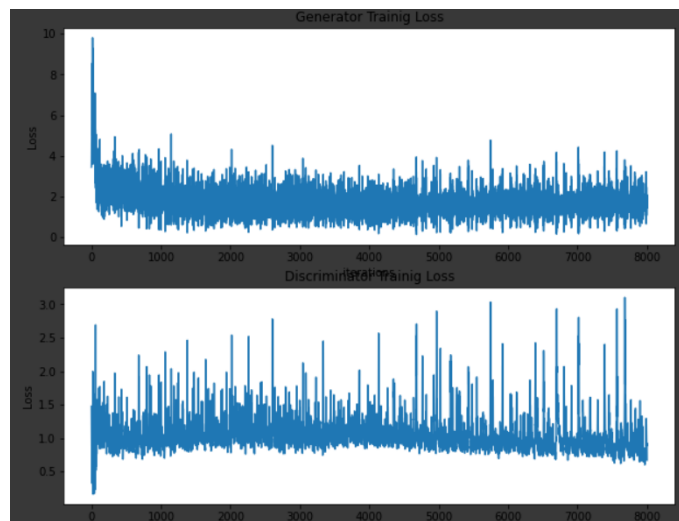
Pour le Discriminateur, on remarque que la loss descend très rapidement au début car le discriminateur arrive à distinguer les images réelles des images générées par le générateurs et qui sont très bruitées (d'où la croissance de la loss du générateur aussi) .

Ensuite, au cours de l'apprentissage, on peut voir que la loss du générateur commence à baisser et que la loss du discriminateur comment à accroître ce qui montre que le modèle générateur arrive tromper le discriminateur en générant des données de plus en plus semblables aux données réelles, le discriminateur n'arrive plus à très bien à distinguer les deux types de données ce qui fait accroître sa loss, et les modèles continuent à progresser conjointement d'où la présence des oscillations .

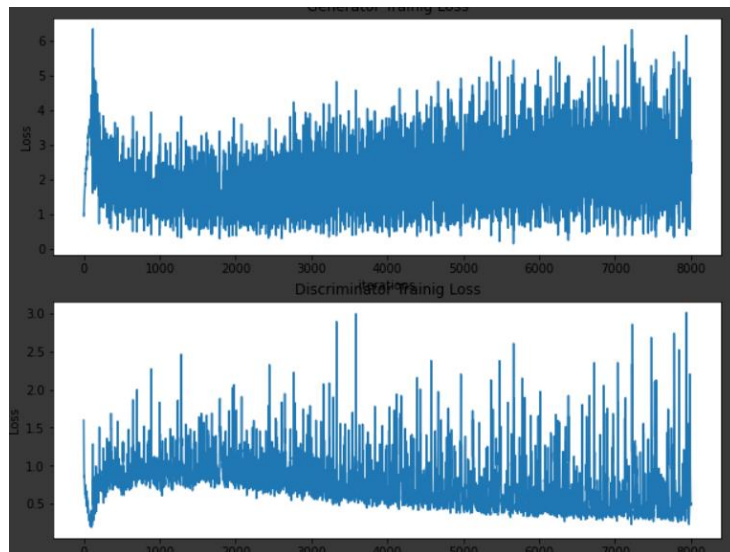
5-

- Changer le learning rate :0.001

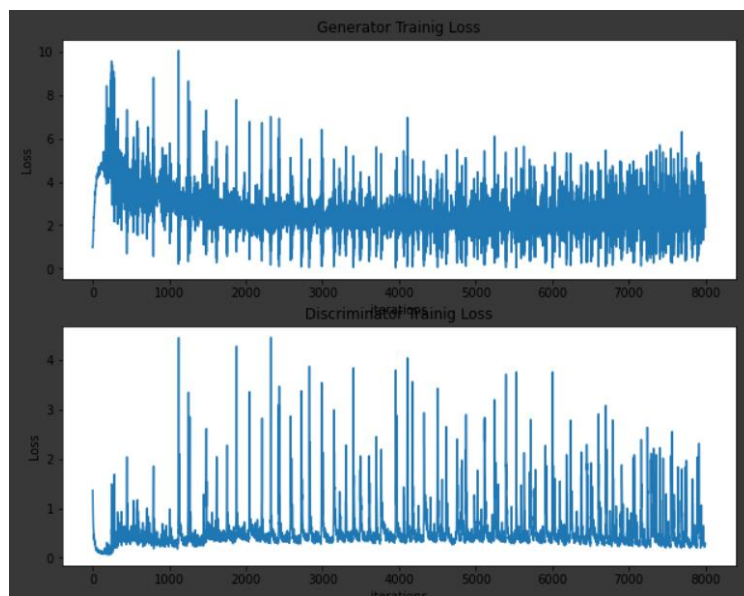
En mettant un learning rate plus grand, ceci augmente un peu la loss du discriminateur et on remarque que les modèles sont un peu moins bons lors de la génération des visages



- Mettre le momentum à 0.9 :
Les visages générés sont plus flous, le discriminateur a une loss faible et arrive à séparer les images réelles des fausses. Le générateur quant à lui a une loss plus élevée.
- $N = 10$



$N=1000$:



Avec $n=10$ et $n=1000$ on remarque que les résultats sont presque les mêmes que $n = 100$, l'instabilité est toujours présente à cause des oscillations.

Partie 2 – Conditional Generative Adversarial Networks

2.1 Principe général

6- Les équations :

$$\max_D \mathbb{E}_{x \sim \text{Data}} [\log D(x|y)] + \mathbb{E}_{z \sim P(z)} [\log (1 - D(G(z|y)))] - \dots$$

$$\max_G \mathbb{E}_{z \sim P(z)} [\log (D(G(z|y)))] \dots (7)$$

- 7- Les variables aux quels le générateur de la figure 6 peut être conditionné sont : Young, old, male , female .
- 8- Les variables aux quels le générateur de la vidéo 1 peut être conditionné sont : la saison et il peut être également conditionné à des variables liées à la météo et au changement climatique, tels que la neige, couleur de la végétation, le ciel
- 9- Les variables aux quels le générateur de la vidéo 2 peut être conditionné sont : le style de la ville (bâtiments, arbres ...ect) , actions de l'utilisateur, et les dernières images générées

2.2 Architectures cDCGAN pour MNIS

10- Commentaires des expériences DCGAN :

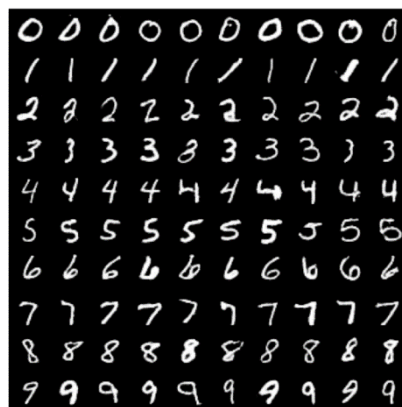
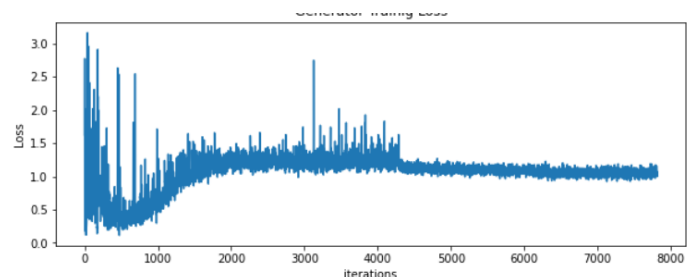
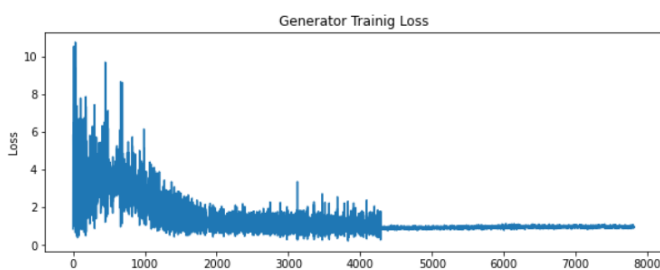


Figure - Image des chiffres générés après presque 8000 itérations

Figure – évolution de la loss du générateur (à droite) et du discriminateur (à gauche)

Les chiffres générés après presque 8000 itérations sont assez réalistes et correctes. La loss du discriminateur (malgré le fait qu'elle soit instable) a diminué au début ce qui montre la capacité du modèle à séparer les données générées des réelles, puis la loss commence à s'accroître ce qui montre que le générateur arrive à générer des données qui commencent à tromper le discriminateur, (à



l'inverse le générateur ou on voit sa loss augmenter puis à baisser). Jusqu'à l'itération 4200 ou on voit que les deux loss se stabilisent et les chiffres générés sont réalistes.

- 11- Non on ne peut pas enlever l'information Y car c'est cette information qui permet au discriminateur de détecter si le générateur génère une donnée qui ne correspond pas à ce qui a été demandé de générer.

2.3 Architectures cGAN pour MNIS

- 12- Commentaires sur le cGan :



Figure - Images des chiffres générés

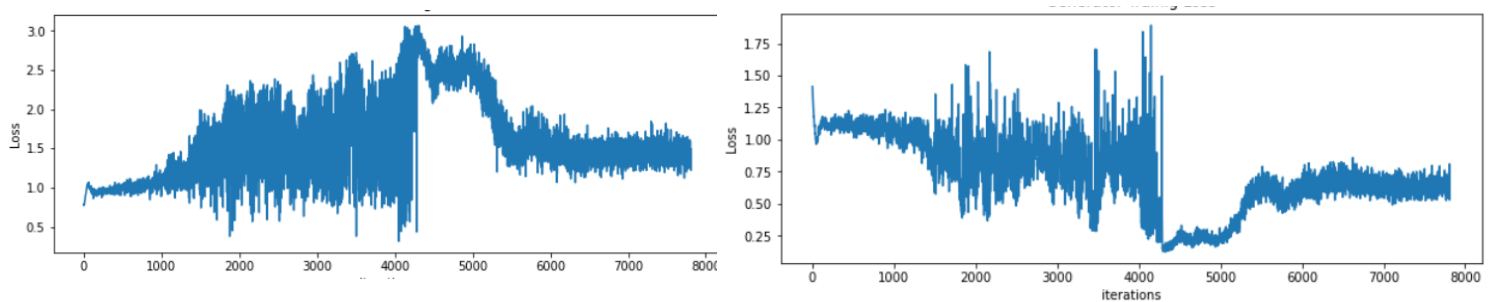


Figure – L'évolution de la loss du générateur (à droite) et discriminateur (à gauche)

Les images des chiffres générées par le cGan sont un peu bruitées et pas tous correctes contrairement au cdcGan. Même après 4200 itérations ou les deux modèles commence à se stabiliser mais n'arrivent pas à se stabiliser très bien.

- 13- Il est relativement plus difficile de générer des chiffres conditionnés avec un cGAN qu'avec un cDCGAN car c'est les couches de convolutions qui améliorent la génération dans le cDCGAN car elles capturent l'information spatiale tandis que le cGan a des couches fully connected

seulement qui ne capturent pas cette information spatiale ce qui rend la génération plus difficile.