Master 2 Machine Learning for Data Science

Deep learning : apprentissage semi-supervisé sur MNIST

Réalisé par:

Amira KOUIDER - 21904040 FI

Nadia RADOUANI – 21911973 FI

Enseignant: M Blaise HANCZAR

Année universitaire: 2020-2021

Table des matières

Table des figures

I. Introduction

Dans le deep learning, l'apprentissage supervisé a été le paradigme d'apprentissage le plus populaire et le centre de la plupart des recherches. Cependant, il requiert des données volumineuses d'où la nécessité de créer des modèles capables d'apprendre à partir de moins de données.

Dans cette optique, l'apprentissage semi-supervisé est une technique dans laquelle des données étiquetées et non étiquetées sont utilisées pour former un classifieur. Ce type de classifieurs prend une infime partie des données étiquetées et une quantité beaucoup plus importante de données non étiquetées (provenant de la même base). L'objectif est de combiner ces sources pour former un réseau neuronal et lui apprendre à classer un nouveau point de données.

Dans cette perspective, les GAN (*Generative Adversarial Network*) introduits en 2014 dans l'article Generative Adversarial Networks par Goodfellow et ses collaborateurs, offrent une approche prometteuse qui se repose sur la mise en compétition de deux réseaux au sein d'un framework.

Dans cette étude, nous implémentons un algorithme d'apprentissage semi-supervisé sur la base de données MNIST pour la reconnaissance de données manuscrites. Nous implémenterons une variante du DCGAN (Deep convolutional Generative Adversial Network) avec seulement 100 observations labellisées.

II. GAN (Generative Adversarial Network)

De nombreux systèmes d'apprentissage machine examinent une entrée compliquée par exemple une image, et produisent une sortie simple (une étiquette comme "voiture"). En revanche, un modèle génératif prend un petit morceau d'entrée, peut-être quelques nombres aléatoires, et produit une sortie complexe, comme l'image d'un visage d'apparence réaliste.

Pour cette étude, nous aurons recours au modèle génératif DCGAN (*Deep Convolutional Generative Adversarial Network*) pour faire de l'apprentissage semi-supervisé avec seulement 100 labels utilisés lors de la phase d'apprentissage. Mais pour pouvoir le comprendre, il faut d'abord comprendre le GAN.

Une manière intuitive de comprendre les GAN est d'imaginer un faussaire essayant de fabriquer des fausses montres Rolex. On l'appelle le générateur. Ce dernier mélange ses fausses Rolex avec des vraies et les présente à un marchand qui a des connaissances basiques en identification de montres contrefaites. On l'appelle discriminateur.

Le marchand fait une évaluation de l'authenticité de chaque montre : le discriminateur renvoie 1 si la montre est authentique, et 0 si elle ne l'est pas et donne un retour au faussaire sur ce qui fait ressembler une «Rolex» à une Rolex.

Avec la backpropagation, le faussaire retourne dans son atelier pour fabriquer de nouvelles Rolex (il va mettre à jour les poids et le biais). Au fil du temps, le faussaire devient de plus en plus compétent pour imiter le style de Rolex – En maximisant la probabilité que sa montre ressemble à une vraie Rolex – et le marchand devient de plus en plus expert en repérage de contrefaçon (en maximisant la probabilité de repérer une Rolex et celle de repérer une contrefaçon).

Les GANs sont donc constitués de deux réseaux de neurones, l'un formé pour générer des données et l'autre formé pour distinguer les fausses données des données réelles (d'où la nature contradictoire du modèle).

Le générateur apprend à générer des données plausibles. Les cas générés deviennent des exemples d'apprentissage pour le discriminateur : le générateur essaie de tromper le discriminateur, il essaie de maximiser la probabilité de faire en sorte que le discriminateur confonde ses entrées avec la réalité.

Tandis que le discriminateur apprend à distinguer les fausses données du générateur des données réelles : il essaie de ne pas être dupé. Il pénalise le générateur pour avoir produit des résultats non plausibles et il le guide pour produire des images plus réalistes.

Pour résumer, le générateur et le discriminateur sont tous les deux des réseaux de neurones. La sortie du générateur est directement connectée à l'entrée du discriminateur. Grâce à la rétropropagation, la classification du discriminateur fournit un signal que le générateur utilise pour mettre à jour ses poids.

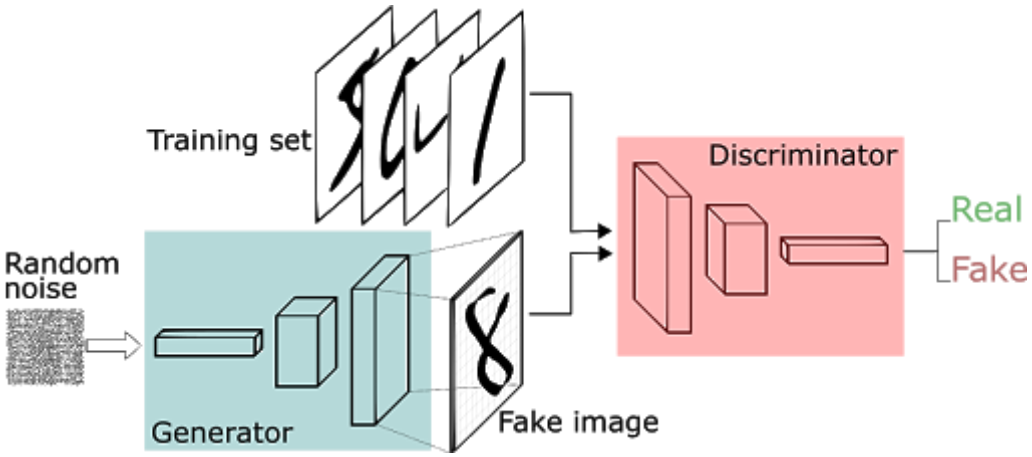

Figure 1 - Architecture des deux réseaux

Pour la phase d'apprentissage, l'architecture GAN implique l'apprentissage simultané de deux modèles: le générateur et le discriminateur.

1) Le discriminateur apprend pendant n epochs en maintenant le générateur constant. Il analyse uniquement les données réelles pour apprendre à les comprendre.
2) Le générateur apprend à produire des données contrefaites pendant m epochs en maintenant le discriminateur constant.
3) Les étapes 1 et 2 sont répétées pour l'apprentissage du générateur et du discriminateur. Grâce à cette mise en compétition, les deux réseaux se développent et s'améliorent en précision et en efficacité : le réseau générateur apprend à générer des ensembles de données toujours plus réalistes, et le réseau discriminateur apprend à identifier comme faux des ensembles de données contrefaites extrêmement trompeurs.

Au fur et à mesure que le générateur s'améliore avec l'apprentissage, les performances du discriminateur se détériorent car celui-ci ne peut pas faire la différence entre les données réelles et les fausses. Si l'apprentissage du générateur est réussi, alors le discriminateur a une précision de 50%.

Le retour d'information des discriminateurs perd de son importance avec le temps. Si le GAN continue à s'entraîner au-delà du moment où le discriminateur donne un retour d'information complètement aléatoire, alors le générateur commence à s'entraîner sur le retour d'information indésirable, et sa qualité se détériore.

D’un point de vue mathématiques, les GANs sont formulées de telle manière d’avoir comme objectif de minimiser la fonction de coût (qui calcule la distance entre l'observation réelle et l’observation artificielle) décrite ci-dessous :

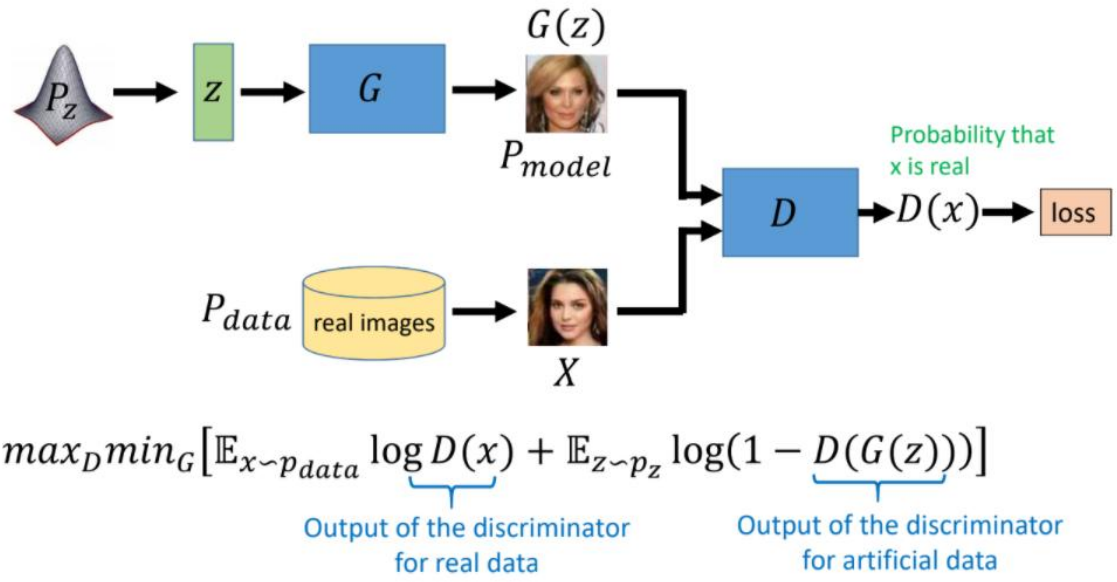

Figure 2 - Schéma explicatif et formule mathématique de la fonction de coût

Le discriminateur cherche à maximiser la probabilité que l’image réelle soit réelle, donc maximiser sa sortie pour les données réelles D(x), et minimiser la probabilité que l’image artificielle soit réelle (1- D(G(z))).

III. DCGAN

Les DCGANs sont l'une des conceptions de réseau les plus populaires et les plus réussies du GAN. Ils se composent principalement de couches de convolution sans max pooling ni de couches entièrement connectées. Ils utilisent des pas de convolution et une convolution transposée pour le sous-échantillonnage et le sur-échantillonnage. La figure ci-dessous représente la conception du réseau pour le générateur et discriminateur :

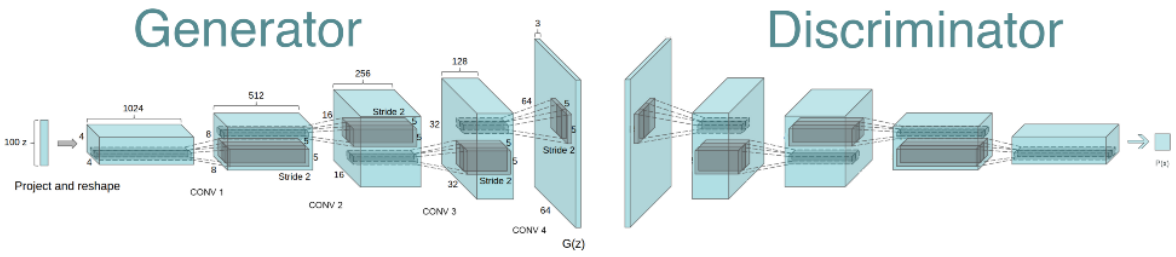

Figure 3 - La conception du réseau pour le générateur et le discriminateur

Pour résumer les DCGANs :

- Ils remplacent toutes les couches max pooling par une couche de convolution.
- Ils utilisent la convolution transposée pour le sur-échantillonnage.
- Ils éliminent les couches entièrement connectées.
- Ils utilisent la normalisation par lots (Batch normalization) sauf pour la couche de sortie du générateur et la couche d'entrée du discriminateur.
- Ils utilisent ReLU comme fonction d'activation dans le générateur sauf pour la sortie qui utilise tanh. Tandis que dans le discriminateur ils utilisent LeakyReLU.

IV. Implémentation

Pour faire notre apprentissage avec seulement 100 labels, nous procéderons avec deux modèles: Un modèle d'apprentissage supervisé, et un modèle d'apprentissage semi supervisé. Notre objectif sera toujours de former un modèle qui prend *X* en entrée et génère *y* en sortie. Cependant, tous nos exemples d'apprentissage ne portent pas d'étiquette *y*. Les 100 premières observations ont été choisies pour faire l'apprentissage du modèle.

Figure 4 - Les 100 labels utilisées pour l'apprentissage

Nous avons eu les résultats suivants :

Figure 5 -Quelques images générées par le générateur pour chaque epoch

Nous pouvons examiner des échantillons d'images générées. Comme on pouvait s'y attendre, les échantillons d'images générés avant l'époque 30 sont de qualité relativement médiocre. A partir de l'époque 30, on obtient des échantillons relativement clairs.

Nous pouvons s'appuyer ainsi que le graphique de la fonction coût (figure suivante)

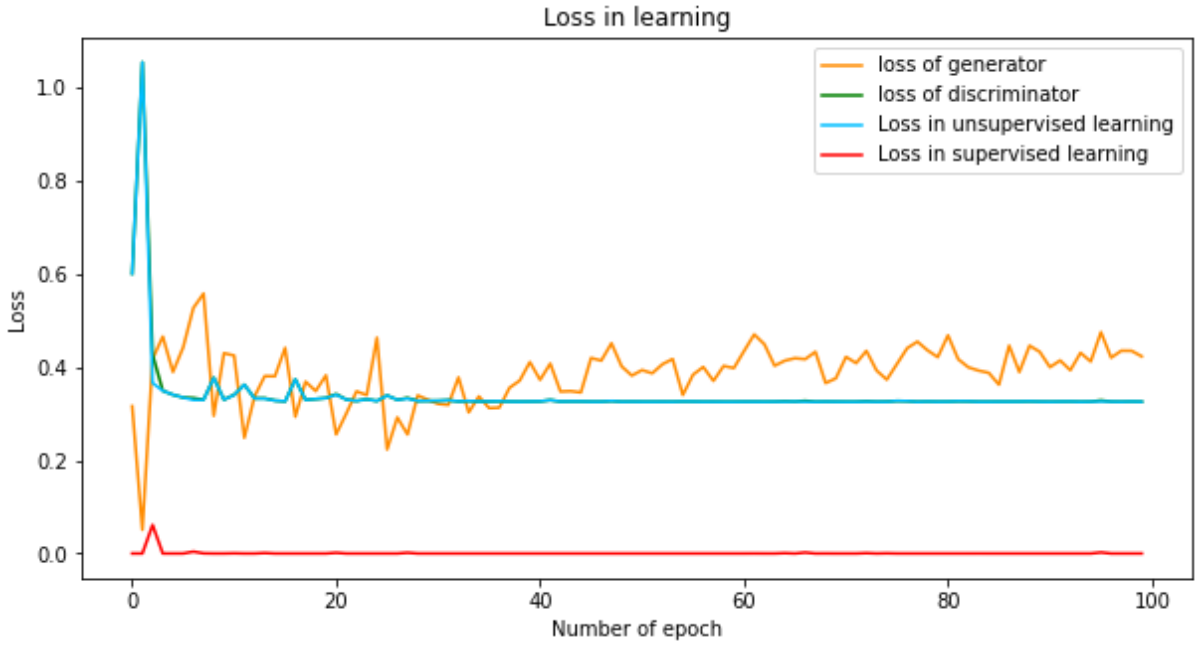

Figure 6 - Graphique fonction de coût

En ce qui concerne la perte de discriminateur, on peut la diviser en deux parties : La perte non supervisée qui représente le problème du GAN où le discriminateur doit faire la différence entre les vraies images provenant de l'ensemble d'apprentissage et les fausses images provenant du générateur, et la perte supervisée qui calcule les probabilités individuelles réelles des classes.

Il s'agit donc d'une classification binaire : nous voulons une valeur de probabilité proche de 1 pour les images réelles et proche de 0 pour les images artificielles. Nous avons donc utiliser la fonction d'entropie croisée sigmoïde (sigmoid cross entropy function) pour calculer cette perte.

Pour la perte supervisée, nous devions utiliser les logits du discriminateur. Comme il s'agit d'un problème de classification multi-classes, nous pouvons utiliser la fonction d'entropie croisée softmax (softmax cross entropy function) avec les véritables étiquettes dont nous disposions.

La perte discriminatoire est donc la somme de la perte supervisée et de la perte non supervisée. En outre, comme nous prétendons ne pas avoir la plupart des étiquettes, nous devons les ignorer dans la perte supervisée.

Pour la perte du générateur, nous utilisons le feature matching : c'est le concept de pénalisation de l'erreur moyenne absolue entre la valeur moyenne d'un ensemble de caractéristiques sur les données d'apprentissage et les valeurs moyennes de cet ensemble de caractéristiques sur les échantillons générés. Autrement dit, le générateur sera entraîné à faire correspondre les valeurs attendues des caractéristiques sur une couche intermédiaire du discriminateur.

Bien que la perte de feature matching fonctionne bien dans le cadre d'un apprentissage semi-supervisé, les images produites par le générateur ne sont pas très bonnes.

V. Conclusion

De nombreux chercheurs considèrent l'apprentissage non supervisé comme le chaînon manquant aux systèmes généraux d'intelligence artificielle.

Pour briser ces obstacles, il est essentiel de tenter de résoudre des problèmes déjà établis en utilisant des données moins étiquetées. Dans ce scénario, les GANs constituent une véritable alternative pour apprendre des tâches compliquées avec des échantillons moins étiquetés.

Pourtant, il existe encore un écart de performance entre l'apprentissage supervisé et semi-supervisé. Mais nous pouvons certainement nous attendre à ce que cet écart se réduise à mesure que de nouvelles approches entrent en jeu.