

A dark blue vertical bar on the left side of the slide. A blue arrow points to the right from the bar, containing the date.

11/01/2024

cVAE GAN

Advanced Neural Net

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Ramiqi Andi

## Table des matières

Table des matières .....	1
1. Introduction.....	2
2. Rappel VAE .....	2
3. GAN.....	3
Générateur .....	3
Discriminateur .....	3
Entraînement et Convergence dans les GAN .....	3
Fonction de perte : .....	4
4. Implémentation du code : .....	5
Architecture du cVAE pour la Génération : .....	5
Intégration du Discriminateur : .....	5
Processus d'Entraînement du GAN avec cVAE .....	6
Évaluation et Ajustements .....	6
5. Résultats .....	7

## 1. Introduction

Ce rapport présente l'évolution des Conditional Variational Autoencoders (VAE) vers une nouvelle architecture combinant les Generative Adversarial Networks (GAN), nommée Conditional VAE-GAN. Pour donner suite à notre travail précédent sur les Conditional VAE, nous explorons ici comment l'intégration des GAN peut améliorer ces modèles.

En commençant par rappeler les points clés de nos recherches antérieures sur les Conditional VAE, il est ainsi possible de poser les bases pour comprendre l'ajout des GAN. Nous expliquons ensuite comment les VAE-GAN fonctionnent et pourquoi ils sont avantageux, notamment en améliorant la qualité et la diversité des données générées.

Le rapport aborde aussi les défis rencontrés lors de la combinaison des VAE et des GAN, et les solutions proposées. Les performances du Conditional VAE-GAN seront comparées avec celles des modèles classiques, en soulignant les domaines où il excelle.

L'objectif est de fournir une vue d'ensemble claire de cette technologie innovante de la génération de données et de l'apprentissage automatique.

## 2. Rappel VAE

Le rapport précédent sur les Conditional Variational Autoencoders (cVAE) s'est concentré sur leur capacité à générer des données ciblées et contrôlées, en exploitant la structure des VAE pour une génération de données plus flexible et personnalisée. En s'appuyant sur l'exemple du jeu de données MNIST, le rapport a exploré comment les cVAE peuvent non seulement reconstruire des données, mais également générer de nouvelles instances de manière significative. Il a mis en évidence l'importance des cVAE dans l'apprentissage automatique et la création générative, en se focalisant sur leur structure, mécanismes et applications variées.

Dans le rapport précédent, l'aspect technique des Conditional Variational Autoencoders (cVAE) a été abordé en détail. L'accent a été mis sur leur structure, qui combine un encodeur et un décodeur, avec une couche latente probabiliste. Cette structure permet de modéliser les données d'entrée sous forme de distributions, offrant ainsi une flexibilité accrue dans la génération de données. Des éléments clés comme la fonction de perte, composée d'un terme de reconstruction et d'une divergence KL, ont été expliqués pour montrer comment ils influencent l'apprentissage et la génération de données. La capacité des cVAE à conditionner la génération sur des informations supplémentaires a également été mise en avant, démontrant leur utilité dans des applications variées.

Dans le cadre du travail actuel sur les Conditional VAE-GAN, le Conditional Variational Autoencoder (cVAE) développé précédemment sera réutilisé comme fondement essentiel. Cette réutilisation vise à tirer profit de la structure robuste et des capacités

de génération de données du cVAE. En intégrant cette architecture éprouvée avec les Generative Adversarial Networks (GAN), nous envisageons d'améliorer significativement la qualité de la génération de données, en apportant une dimension de réalisme et de variété accrue.

### 3. GAN

Les Generative Adversarial Networks, ou GAN, sont une avancée révolutionnaire dans le domaine de l'apprentissage automatique. Un GAN se compose de deux réseaux neuronaux distincts, le générateur et le discriminateur, qui sont entraînés simultanément dans un cadre compétitif. Le générateur vise à produire des données qui ressemblent à des échantillons réels, tandis que le discriminateur essaie de distinguer les vraies données des fausses générées par le générateur. Cette dynamique crée un processus d'apprentissage continu, où chaque réseau s'améliore en réponse à l'autre, conduisant finalement à la génération de données très réalistes.

#### Générateur :

Dans un GAN, le rôle du générateur est crucial. Il est conçu pour créer des données qui imitent fidèlement la distribution des données réelles. Le générateur commence par recevoir un signal aléatoire, souvent sous la forme d'un vecteur de bruit, et le transforme en données qui ressemblent à celles de l'ensemble de données réel. L'entraînement du générateur se fait par un processus itératif où il apprend à tromper le discriminateur, s'améliorant progressivement pour produire des données de plus en plus réalistes. Ce processus est soutenu par des techniques d'optimisation qui ajustent les paramètres du générateur pour minimiser la capacité du discriminateur à distinguer les vraies données des données générées. Ainsi, le générateur devient un élément clé dans la création de nouvelles instances de données qui sont indiscernables des échantillons réels.

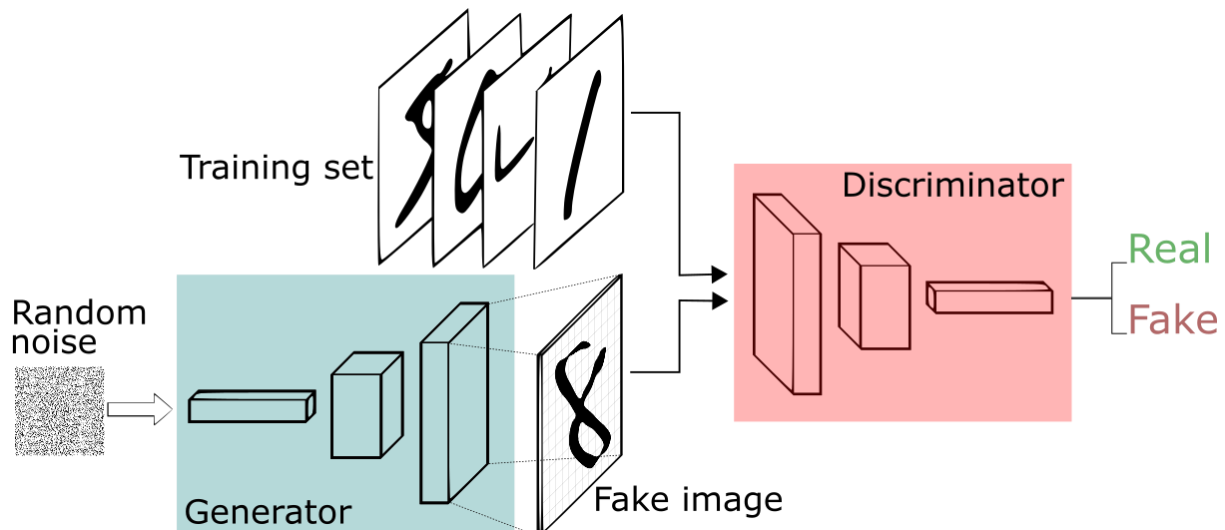
#### Discriminateur

Le discriminateur dans un GAN joue un rôle critique en évaluant les données générées par le générateur. Son objectif principal est de distinguer les vraies données de celles qui sont artificiellement générées. Fonctionnant comme un détecteur, le discriminateur est formé pour identifier les subtilités et les caractéristiques qui différencient les échantillons réels des imitations. Au fur et à mesure de l'entraînement, le discriminateur affine sa capacité à reconnaître ces différences, ce qui pousse en retour le générateur à améliorer sa qualité de génération. Ce jeu compétitif entre le générateur et le discriminateur est essentiel pour le développement d'un GAN efficace, où l'amélioration continue de l'un stimule les progrès de l'autre.

#### Entraînement et Convergence dans les GAN

L'entraînement d'un GAN est un processus délicat où le générateur et le discriminateur sont formés simultanément. Cette section se concentre sur la dynamique d'entraînement et la manière dont la convergence est atteinte. L'objectif est d'atteindre

un équilibre où le générateur produit des données suffisamment convaincantes pour que le discriminateur ne puisse pas facilement les distinguer des vraies données. Ce processus est complexe et nécessite une attention particulière pour éviter des problèmes courants comme le "mode collapse", où le générateur commence à produire une variété limitée de sorties. La convergence est atteinte lorsque le discriminateur n'a plus qu'une chance de 50% de distinguer correctement les vraies données des fausses, indiquant que le générateur a réussi à imiter parfaitement la distribution des données réelles.



### Fonction de perte :

Dans les GANs (Generative Adversarial Networks), le processus d'entraînement implique la minimisation d'une fonction de perte qui quantifie la différence entre les données générées et les données réelles. Les GANs visent à imiter une distribution de probabilité, nécessitant ainsi des fonctions de perte qui quantifient la dissimilarité entre la distribution des données générées et la distribution réelle.

Il existe plusieurs approches pour mesurer cette différence, mais c'est la fonction de perte minimax qui sera mise en place. Cette dernière décrit un scénario où le générateur vise à minimiser une fonction tandis que le discriminateur cherche à la maximiser. Elle se base sur la divergence de cross-entropie entre les distributions réelles et générées.

$$E_x [\log(D(x))] + E_z [\log(1 - D(G(z)))]$$

- $D(x)$  représente l'estimation du discriminateur de la probabilité qu'une instance de données réelle  $x$  soit réelle.
- $E_x$  est la valeur attendue sur toutes les instances de données réelles.

- $G(z)$  est la sortie du générateur lorsqu'il reçoit un bruit  $z$ .
- $D(G(z))$  est l'estimation du discriminateur de la probabilité qu'une instance fausse soit réelle.
- $E_z$  est la valeur attendue sur tous les inputs aléatoires au générateur (en effet, la valeur attendue sur toutes les fausses instances générées  $G(z)$ ).
- La formule utilise la « cross-entropy » entre les distributions réelles et générées.

## 4. Implémentation du code :

### Architecture du cVAE pour la Génération :

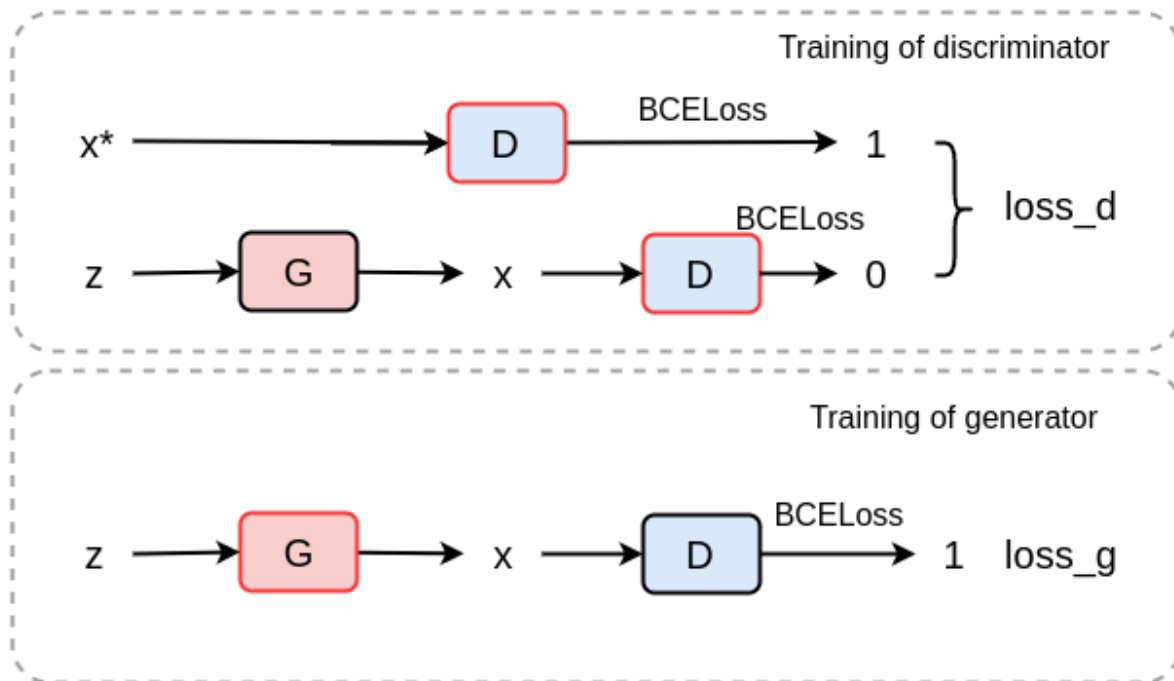
L'architecture du cVAE utilisé comme générateur dans notre GAN est cruciale pour sa performance. Le cVAE se compose de deux parties principales : l'encodeur et le décodeur. L'encodeur transforme les données d'entrée en une représentation latente, tandis que le décodeur reconstruit les données à partir de cette représentation. Pour l'adapter au rôle de générateur dans un GAN, le cVAE est modifié pour produire des données plus variées et réalistes. Des fonctions d'activation spécifiques, comme ReLU ou LeakyReLU, sont choisies pour optimiser le processus de génération. De plus, la capacité du cVAE à intégrer des conditions, comme des étiquettes ou des attributs, est exploitée pour générer des données ciblées et diversifiées. L'ensemble de ces modifications vise à améliorer la capacité du cVAE à créer des données convaincantes pour le discriminateur du GAN.

### Intégration du Discriminateur :

La mise en place du discriminateur dans notre GAN est essentielle pour juger la qualité des données générées par le cVAE. Le discriminateur est un réseau de neurones conçu pour différencier les données réelles de celles générées. Son architecture est moins complexe que celle du générateur, mais suffisamment robuste pour effectuer des évaluations précises. Il peut utiliser des couches de convolution pour traiter les données, suivies de couches denses pour la prise de décision. Dans notre cas, celui-ci sera mis en place grâce au même MLP utilisé pour le modèle cVAE. L'entraînement du discriminateur est un processus d'ajustement fin, où il apprend à reconnaître les subtilités des données générées par rapport aux données réelles. Cela crée un environnement compétitif qui stimule l'amélioration continue du générateur.

## Processus d'Entraînement du GAN avec cVAE

L'entraînement du GAN se fait en utilisant un cVAE comme générateur et un réseau discriminateur. Le processus commence par la préparation des données, où les images MNIST sont aplaties et les étiquettes sont converties en « one-hot encoding ». Une astuce employée consiste à introduire du bruit dans les étiquettes pour renforcer la robustesse du modèle.



Le discriminateur est d'abord entraîné sur des données réelles avec des étiquettes réelles, puis sur des données générées avec des étiquettes fausses. Le générateur (cVAE) est ensuite entraîné pour tromper le discriminateur. Cette étape est cruciale, car le générateur apprend à produire des données qui sont indifférenciables de données réelles pour le discriminateur.

Le processus d'entraînement est itératif, avec des ajustements des poids des réseaux à chaque époque. Les pertes du discriminateur et du générateur sont calculées et utilisées pour guider l'optimisation. Le discriminateur cherche à minimiser les erreurs de classification des données réelles et générées, tandis que le générateur vise à maximiser l'erreur du discriminateur en générant des données de plus en plus réalistes.

Ce processus d'entraînement compétitif, où le générateur et le discriminateur s'améliorent mutuellement, est au cœur de l'efficacité des GANs dans la génération de données réalistes.

## Évaluation et Ajustements

Une fois le processus d'entraînement terminé, l'évaluation de la performance du GAN est essentielle. Cette évaluation se fait en analysant les pertes moyennes du discriminateur et du générateur sur l'ensemble des époques. Un discriminateur performant aura des pertes faibles sur les données réelles et élevées sur les données

générées, tandis qu'un générateur efficace aura une perte faible, indiquant qu'il trompe efficacement le discriminateur.

Des ajustements sont apportés en fonction de ces résultats. Si le discriminateur est trop performant, cela peut empêcher le générateur de s'améliorer. Dans ce cas, des techniques comme l'introduction de bruit dans les étiquettes ou l'ajustement des taux d'apprentissage peuvent être utilisées. Inversement, si le générateur est trop dominant, le discriminateur peut nécessiter un entraînement plus intensif ou une architecture plus complexe.

## 5. Résultats

```
Epoch 90/100, Average Losses => Discriminator (Real: 0.5992477484095071, Fake: 0.808248669417428), Generator: 8655.515130211803
Epoch 91/100, Average Losses => Discriminator (Real: 0.6000429355640655, Fake: 0.8067867842945717), Generator: 8649.877182950344
Epoch 92/100, Average Losses => Discriminator (Real: 0.6006228303604288, Fake: 0.8057202684726796), Generator: 8653.996466467885
Epoch 93/100, Average Losses => Discriminator (Real: 0.6015118471086661, Fake: 0.8046842981884474), Generator: 8653.006729744136
Epoch 94/100, Average Losses => Discriminator (Real: 0.60250358547229, Fake: 0.8036067525842296), Generator: 8650.883434564066
Epoch 95/100, Average Losses => Discriminator (Real: 0.6023830686932179, Fake: 0.8028556275596497), Generator: 8650.89905904351
Epoch 96/100, Average Losses => Discriminator (Real: 0.6035358316735673, Fake: 0.8016546470269974), Generator: 8645.848483413012
Epoch 97/100, Average Losses => Discriminator (Real: 0.6038541007143602, Fake: 0.801117436463899), Generator: 8646.358237585788
Epoch 98/100, Average Losses => Discriminator (Real: 0.6049626585898369, Fake: 0.7997086991760523), Generator: 8645.709267036747
Epoch 99/100, Average Losses => Discriminator (Real: 0.6055946212206313, Fake: 0.7988611556319539), Generator: 8642.90150773754
```

Les résultats obtenus après 100 époques d'entraînement du GAN indiquent une compétition équilibrée entre le discriminateur et le générateur. Les pertes moyennes du discriminateur pour les données réelles et générées restent relativement constantes, signe que le discriminateur maintient sa capacité à différencier les données réelles des données générées. D'autre part, la perte du générateur diminue légèrement, ce qui suggère une amélioration de sa capacité à générer des données qui leurront le discriminateur. Cette tendance est positive et indique que l'entraînement du GAN progresse correctement.<sup>1</sup>



Lorsque l'on génère 10 nouveaux nombres aléatoires, ces derniers sont très différenciables et montrent que le modèle a appris à générer les différents nombres sur lesquels il s'est entraîné, et cela de différentes manières. Nous pouvons voir que les formes du chiffre « 2 » ou « 8 » ne sont pas identiques, mais toujours reconnaissables. L'ajout du GAN dans le modèle cVAE a permis d'améliorer l'efficacité dans la génération de données, permettant une diversité plus grande ainsi que des détails plus précis.

<sup>1</sup> Pour être honnête, je n'ai pas réussi à trouver la potentielle erreur qui fait que mon générateur ait une perte aussi élevée. La génération de nouvelles données se déroulant bien, je ne me suis pas attardé dessus afin de me concentrer sur le rapport...