

Soutenance du projet signal : Modèles génératifs de diffusion

Ayoub Choukri - Axel Olougouna - Yannis Mac - Hicham Laghmam - Maxime Moshfeghi

June 26, 2024

Plan

1 Introduction

2 Modèles génératifs de diffusion : Fondements mathématiques

- Idée principale
- Modèle de diffusion
- Analogie avec les VAE
- Réseaux de neurones U-Net

3 Application à la génération d'images MNIST

- Avec le réseau UNET
- Réseaux de neurones débruiteurs préentraînés

4 Conclusion

Introduction

Contexte

- Les modèles génératifs de diffusion sont des modèles qui permettent de générer à partir d'une base de données, de nouvelles données qui y ressemblent.
- Ces modèles sont utilisés dans plusieurs domaines comme la génération de texte, d'images, de sons, etc.
- Les modèles génératifs de diffusion sont des modèles récents qui ont montré des performances très intéressantes dans la génération de données.

Objectif

- Avoir une compréhension minimale du fondement mathématique des modèles génératifs de diffusion.
- Implémenter un modèle génératif de diffusion, et l'appliquer à la génération de nouvelles images MNIST.

Introduction: Exemple de génération d'images



Figure: Exemple de génération d'images à partir d'un modèle génératif de diffusion

Plan

1 Introduction

2 Modèles génératifs de diffusion : Fondements mathématiques

- Idée principale
- Modèle de diffusion
- Analogie avec les VAE
- Réseaux de neurones U-Net

3 Application à la génération d'images MNIST

- Avec le réseau UNET
- Réseaux de neurones débruiteurs préentraînés

4 Conclusion

Modèles génératifs de diffusion : pourquoi un tel nom ?

Idée principale

- Dans le contexte de la modélisation statistique, un modèle de Diffusion est le processus de transformation d'une distribution de probabilité assez complexe (Distribution des images de chats, des images de chiens, etc.) en une distribution de probabilité plus simple (Prédefinie, et généralement gaussienne).
- Considérons, par exemple, comment on pourrait modéliser la distribution de toutes les photos naturelles. Chaque image est un point dans l'espace de toutes les images, et la distribution des photos naturelles est un "nuage" dans cet espace. En ajoutant répétitivement du bruit aux images, ce nuage se diffuse dans le reste de l'espace des images, jusqu'à devenir pratiquement indiscernable d'une distribution gaussienne $\mathcal{N}(0, 1)$.
- Un modèle capable d'inverser approximativement cette diffusion peut alors être utilisé pour échantillonner à partir de la distribution originale. Cette approche est étudiée dans le cadre de la thermodynamique hors équilibre, car la distribution initiale n'est pas en équilibre, contrairement à la distribution finale.

Modèles génératifs de diffusion : pourquoi un tel nom ?

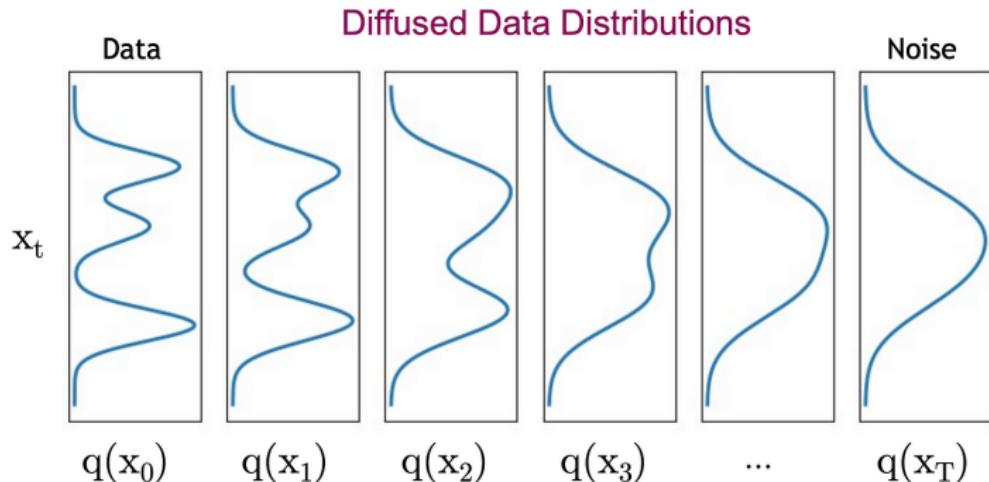


Figure: Diffusion d'une distribution complexe vers une distribution gaussienne

Source : <https://lih-verma.medium.com/diffusing-the-mathematical-equations-of-diffusion-modelling-9a64565fd430>

Forward diffusion model

- On note, pour $t \in \llbracket 0, T \rrbracket$, x_t l'image à l'instant t , et x_0 l'image initiale.
- On souhaite définir un modèle forward de diffusion qui va nous permettre de transformer cette image (point probable selon la distribution des images) en une image plus simple (point probable selon une distribution gaussienne).
- En d'autres termes, il nous faut définir une transformation qui va de plus en plus ajouter du bruit gaussien à notre image.
- L'équation principale de notre forward diffusion est la suivante :

$$x_{t+1} = \sqrt{\alpha}x_t + \sqrt{1 - \alpha}\epsilon_t \quad (1)$$

Où $\epsilon_t \sim \mathcal{N}(0, I)$, et α est un paramètre de diffusion.

On pourrait se demander pourquoi une telle équation va permettre de transporter notre distribution complexe vers une distribution gaussienne.

Forward diffusion model

- Soit $\alpha, \beta \in \mathbb{R}$, cherchons sous quelle conditions la distribution de x_t à la limite est une gaussienne. Avec $x_{t+1} = \sqrt{\alpha}x_t + \sqrt{\beta}\epsilon_t$.
- En effet, par une récurrence simple, on peut prouver que :

$$x_t = \alpha^{\frac{t}{2}} x_0 + \sum_{i=0}^{t-1} \alpha^{\frac{i}{2}} \sqrt{\beta} \epsilon_{t-i}$$

- Ainsi, si $\alpha < 1$, et comme $\epsilon_t \stackrel{iid}{\sim} \mathcal{N}(0, I)$, alors :

$$\alpha^{\frac{t}{2}} \xrightarrow[t \rightarrow +\infty]{} 0$$

et

$$\sum_{i=0}^{t-1} \alpha^{\frac{i}{2}} \sqrt{\beta} \epsilon_{t-i} \xrightarrow[t \rightarrow +\infty]{} \mathcal{N}\left(0, \frac{\beta}{1-\alpha}\right)$$

car

$$\text{Var}\left(\sum_{i=0}^{t-1} \alpha^{\frac{i}{2}} \sqrt{\beta} \epsilon_{t-i}\right) = \sum_{i=0}^{t-1} \alpha^i \beta = \frac{1-\alpha^t}{1-\alpha} \beta \xrightarrow[t \rightarrow +\infty]{} \frac{\beta}{1-\alpha}$$

- Ainsi, pour que x_t converge vers une gaussienne, il faut que $\alpha < 1$ et $\beta + \alpha = 1$.

Forward diffusion model

On rappelle l'équation de diffusion forward :

$$x_{t+1} = \sqrt{\alpha}x_t + \sqrt{1 - \alpha}\epsilon_t$$

- On comprend ainsi, que le paramètre α de notre équation Forward de diffusion permet de contrôler la vitesse de diffusion de notre distribution complexe vers une distribution gaussienne.
- En effet, plus α est petit (β est grand) plus la diffusion est rapide, car on aura tendance à plus vite oublier l'image initiale.
- Cependant, dans la pratique, on utilise plutôt une graduation linéaire de α pour permettre une meilleure convergence de notre distribution vers une gaussienne.
- Pour cela, on se définit, une valeur minimale et maximale de β , et l'équation de diffusion devient :

$$x_t = \sqrt{\alpha_t}x_{t+1} + \sqrt{1 - \alpha_t}\epsilon_t$$

Ainsi, l'équation qui lie x_t directement à x_0 est la suivante :

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sum_{i=0}^{t-1} \sqrt{\alpha_{t-i}}\sqrt{1 - \alpha_{t-i}}\epsilon_{t-i} \quad \text{Où} \quad \bar{\alpha}_t = \prod_{i=0}^{t-1} \alpha_i$$

Backward diffusion model

- L'idée du modèle backward est de pouvoir partir à partir d'un bruit gaussien X_T , et de débruiter cette image petit à petit pour arriver à une image plus complexe.

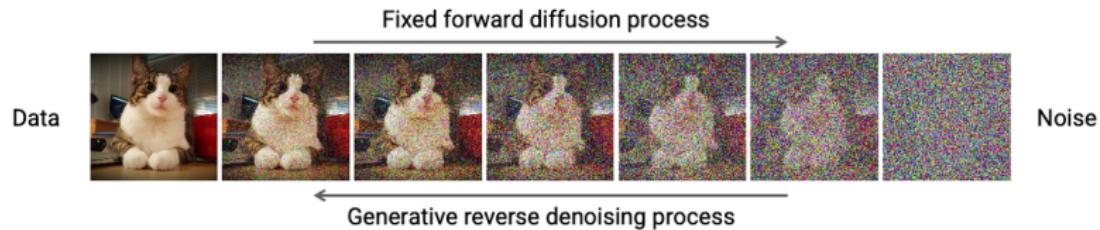


Figure: Idée du modèle backward de diffusion

Backward diffusion model

- L'équation principale du backward diffusion est la suivante :

$$x_t = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta \right) + \sqrt{(\beta_t)} \times \epsilon_t \quad (2)$$

Où ϵ_θ est le bruit prédit par un réseau de neurone débruiteur.

Afin de voir d'où vienne cette équation, il faut vraiment faire l'analogie avec les VAE.

Analogie avec les VAE

- Les Auto Encodeurs variationnels sont des réseaux de neurones artificiels qui ont été introduits en 2013 par D.Kingma et M.Welling.
- L'idée principale des VAE est de pouvoir approcher une distribution de probabilité $P(X)$ par une distribution paramétrée P_θ ayant pour paramètre θ .
- Pour cela, on va chercher à minimiser la divergence de Kullback-Leibler entre $P(X)$ et P_θ .
- On rappelle la divergence de Kullback-Leibler entre deux distributions P et Q est définie par :

$$KL(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx = \mathbb{E}_{X \sim P} [\log \frac{P(X)}{Q(X)}]$$

Analogie avec les VAE

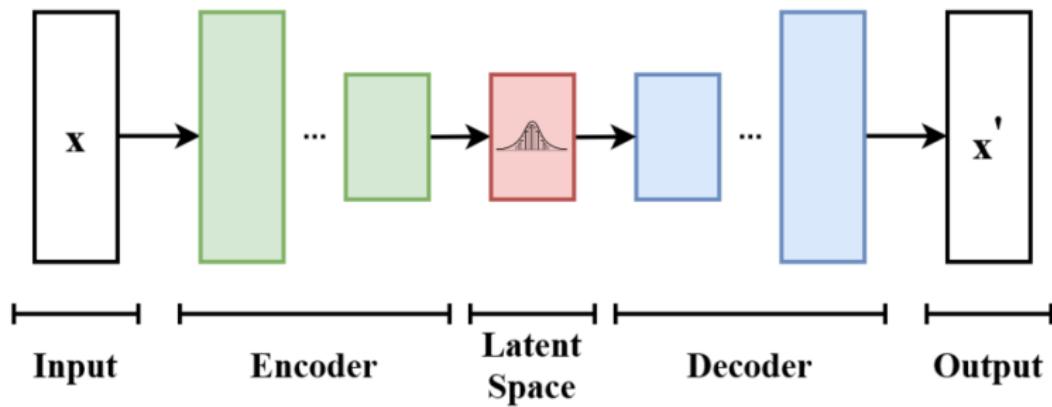
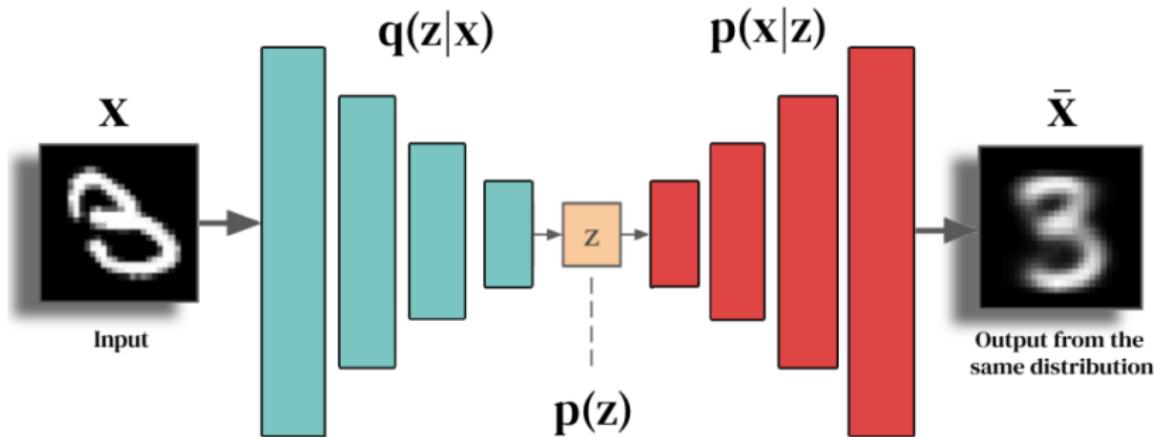


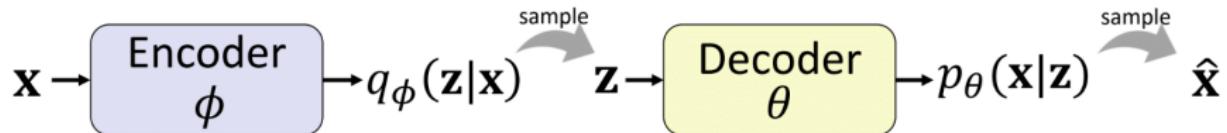
Figure: Le schéma de base d'un auto-encodeur variationnel. Le modèle reçoit x comme entrée. L'encodeur le comprime dans l'espace latent. Le décodeur reçoit en entrée les informations prélevées dans l'espace latent et produit x' aussi semblable que possible à x .

Analogie avec les VAE

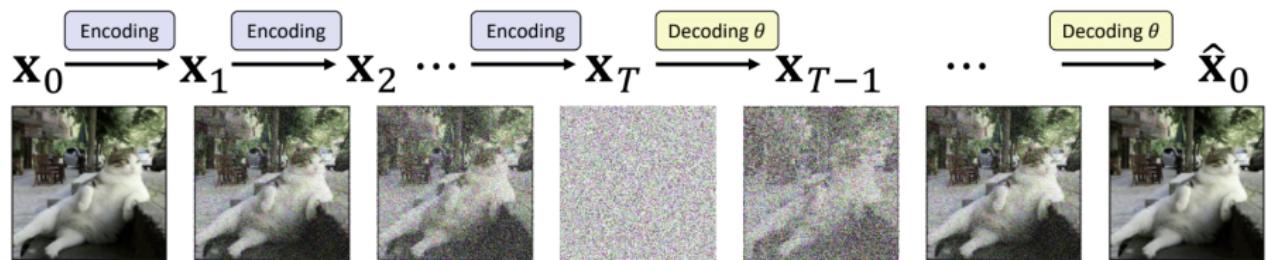


Analogie avec les VAE

Variational Autoencoder (VAE)

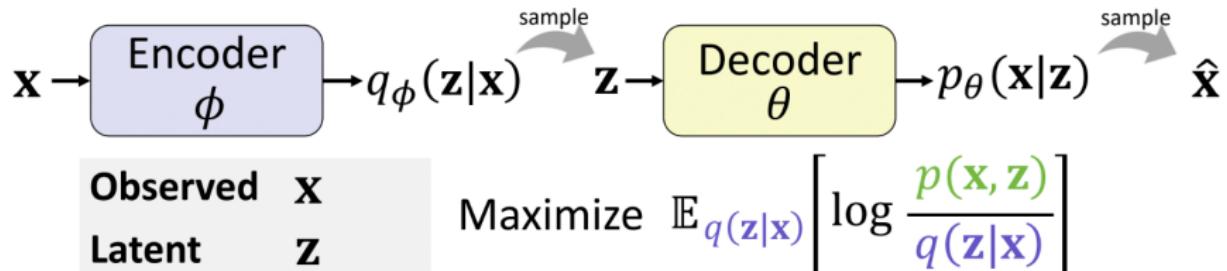


Diffusion models

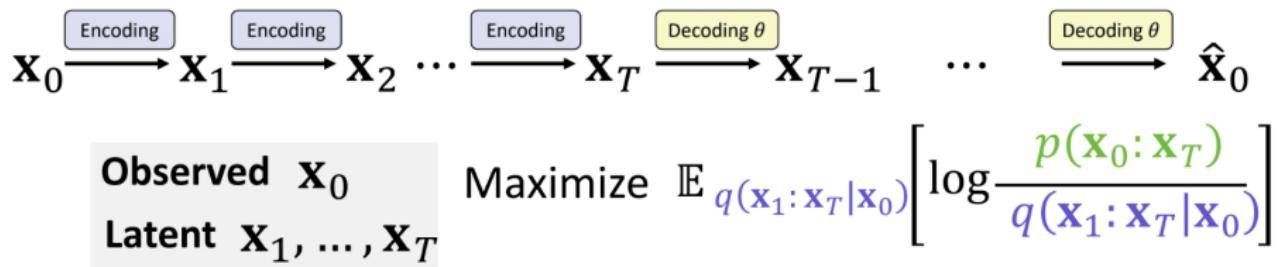


Analogie avec les VAE

Variational Autoencoder (VAE)



Diffusion models



ELBO : Evidence Lower Bound

Voyons voir, pourquoi on maximise $\mathbb{E}_{q(x_1:x_T|x_0)}[\log \frac{P(x_0:x_T)}{q(x_1:x_T|x_0)}]$ dans le modèle de diffusion.

- On rappelle que le but des VAE est de minimiser la divergence de Kullback-Leibler entre $P(X)$ et P_θ .

$$\begin{aligned} KL(P||P_\theta) &= \int P(x) \log \frac{P(x)}{P_\theta(x)} dx \\ &= \int P(x) \log P(x) dx - \int P(x) \log P_\theta(x) dx \\ &= \mathbb{E}_{X \sim P}[\log P(X)] - \mathbb{E}_{X \sim P}[\log P_\theta(X)] \\ &= -H(P) + L_{P_\theta}(X) \end{aligned}$$

Avec $H(P)$ l'entropie de P , et $L_{P_\theta}(X)$ la log-vraisemblance de X selon P_θ .

- Ainsi minimiser la divergence de Kullback-Leibler revient à maximiser la log-vraisemblance de X selon P_θ .

ELBO : Evidence Lower Bound

- Or on remarque que :

$$\begin{aligned}\log p_\theta(x_0) &= \log \int p_\theta(x_0, x_1, \dots, x_T) dx_1 \dots dx_T \\ &= \log \int p_\theta(x_0 : x_T) \frac{q(x_1 : x_T | x_0)}{q(x_1 : x_T | x_0)} dx_1 \dots dx_T \\ &= \log \mathbb{E}_{q(x_1 : x_T | x_0)} \left[\frac{p_\theta(x_0 : x_T)}{q(x_1 : x_T | x_0)} \right] \\ &\geq \mathbb{E}_{q(x_1 : x_T | x_0)} \left[\log \frac{p_\theta(x_0 : x_T)}{q(x_1 : x_T | x_0)} \right]\end{aligned}$$

- Ainsi, au lieu de maximiser la log-vraisemblance, on maximise plutôt ce q'on appelle l'ELBO (Evidence Lower Bound) qui est une borne inférieure de la log-vraisemblance.
- l'ELBO donne une estimation du pire cas de la log-vraisemblance.
- Essayons maintenant d'utiliser le fait qu'on est en train de faire face à des chaines de Markov.

ELBO : Evidence Lower Bound

- On a :

$$\log \frac{p_\theta(x_0 : x_T)}{q(x_1 : x_T | x_0)} = \log \frac{p(x_T) \prod_{t=1}^T p(x_{t+1} | x_t)}{q(x_1 | x_0) \prod_{t=2}^T p(x_t | x_{t-1}, x_0)}$$

- Or le dénominateur peut se réécrire comme suit :

$$\begin{aligned} q(x_1 | x_0) \prod_{t=2}^T q(x_t | x_{t-1}, x_0) &= q(x_1 | x_0) \prod_{t=2}^T \frac{q(x_{t-1} | x_t, x_0) q(x_t | x_0)}{q(x_{t-1} | x_0)} \\ &= \prod_{t=2}^T q(x_{t-1} | x_t, x_0) \times q(x_1 | x_0) \times \prod_{t=2}^T \frac{q(x_t | x_0)}{q(x_{t-1} | x_0)} \\ &= q(x_T | x_0) \prod_{t=2}^T q(x_{t-1} | x_t, x_0) \end{aligned}$$

ELBO : Evidence Lower Bound

- Ainsi, on a :

$$\begin{aligned}\log \frac{p_\theta(x_0 : x_T)}{q(x_1 : x_T | x_0)} &= \log \frac{p(x_T) \prod_{t=1}^T p(x_{t+1} | x_t)}{q(x_T | x_0) \prod_{t=2}^T q(x_{t-1} | x_t, x_0)} \\ &= \log \frac{P(x_T) P_\theta(x_0 | x_1) \prod_{t=2}^T P_\theta(x_{t-1} | x_t)}{q(x_T | x_0) \prod_{t=2}^T q(x_{t-1} | x_t, x_0)} \\ &= \log \frac{p(x_T)}{q(x_T | x_0)} + \log p_\theta(x_0 | x_1) + \sum_{t=2}^T \log \frac{P_\theta(x_{t-1} | x_t)}{q(x_{t-1} | x_t, x_0)}\end{aligned}$$

- On rappelle que ces trois termes, etaient sous l'espérance de $q(x_1 : x_T | x_0)$.

- Ainsi, on a :

$$\log \frac{p_\theta(x_0 : x_T)}{q(x_1 : x_T | x_0)} = \log \frac{p(x_T)}{q(x_T | x_0)} + \log p_\theta(x_0 | x_1) + \sum_{t=2}^T \log \frac{P_\theta(x_{t-1} | x_t)}{q(x_{t-1} | x_t, x_0)}$$

- $\log \frac{p(x_T)}{q(x_T | x_0)}$ est presque égale à 0, car par construction, $q(x_T | x_0)$ est une gaussienne et donc sera proche de $p(x_T)$.
- $\log p_\theta(x_0 | x_1)$ représente l'erreur de reconstruction de notre image x_0 à partir de x_1 .
- Le troisième termes est plutôt intéressant, car il représente à quelle point notre modèle décodeur (débruitage) est capable de retrouver l'image x_{t-1} à partir de x_t .

- Voyons voir maintenant que vaut $q(x_{t-1} \mid x_t, x_0)$.
- En effet, ceci représente la densité de l'image bruitée à l'étape $t - 1$ sachant qu'on connaît x_t et x_0 !

$$\begin{aligned}
q(x_{t-1} \mid x_t, x_0) &= \frac{q(x_{t-1}, x_t, x_0)}{q(x_t, x_0)} \\
&= \frac{q(x_t \mid x_{t-1}, x_0)q(x_{t-1}, x_0)}{q(x_t, x_0)} \\
&= \frac{\mathcal{N}(\sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I) \times \mathcal{N}(\sqrt{\bar{\alpha}_{t-1}}x_0, (1 - \alpha_{t-1}^-)I)}{\mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1 - \alpha_{t-1}^-)I)} \\
&= \mathcal{N}(\mu_q(x_t, x_0), \Sigma_q(x_t, x_0))
\end{aligned}$$

- Où :

$$\mu_q(x_t, x_0) = \frac{(1 - \alpha_{t-1}^-)\sqrt{\alpha_t}x_t + (1 - \alpha_t)\sqrt{\alpha_{t-1}^-}x_0}{1 - \bar{\alpha}_t}$$

Et

$$\Sigma_q(x_t, x_0) = \frac{(1 - \alpha_t)(1 - \alpha_{t-1}^-)}{1 - \bar{\alpha}_t} \times I$$

- Ainsi, on a :

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(\mu_q(x_t, x_0), \Sigma_q(x_t, x_0))$$

Où

$$\mu_q(x_t, x_0) = \frac{(1 - \bar{\alpha}_{t-1})\sqrt{\alpha_t}x_t + (1 - \alpha_t)\sqrt{\alpha_{t-1}}x_0}{1 - \bar{\alpha}_t}$$

Et

$$\Sigma_q(x_t, x_0) = \frac{(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \times I$$

- Or, comme on sait que notre processus Backward:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$$

- Si on fixe $\Sigma_\theta(x_t) = \Sigma_q(x_t, x_0)$, alors minimiser la divergence de Kullback-Leibler revient à minimiser la distance entre $\mu_\theta(x_t)$ et $\mu_q(x_t, x_0)$.

$$KL(q(x_{t-1} \mid x_t, x_0) \parallel p_\theta(x_{t-1} \mid x_t)) = \frac{1 - \bar{\alpha}_t}{2(1 - \alpha_t)(1 - \bar{\alpha}_{t-1})} \|\mu_q(x_t, x_0) - \mu_\theta(x_t)\|^2$$

- Or comme, $\mu_q(x_t, x_0)$ est une combinaison linéaire de x_t et x_0 , et comme :

$$x_0 = \frac{x_t - \sqrt{1 - \hat{\alpha}_t} \epsilon}{\sqrt{\hat{\alpha}_t}}$$

- Si on injecte $x_0 = \frac{x_t - \sqrt{1 - \hat{\alpha}_t} \epsilon}{\sqrt{\hat{\alpha}_t}}$ dans l'expression de $\mu_q(x_t, x_0)$, on obtient :

$$\mu_q(x_t, x_0) = \frac{x_t}{\sqrt{\alpha_t}} + \frac{(1 - \alpha_t)\sqrt{1 - \bar{\alpha}_t} \epsilon}{1 - \bar{\alpha}_t}$$

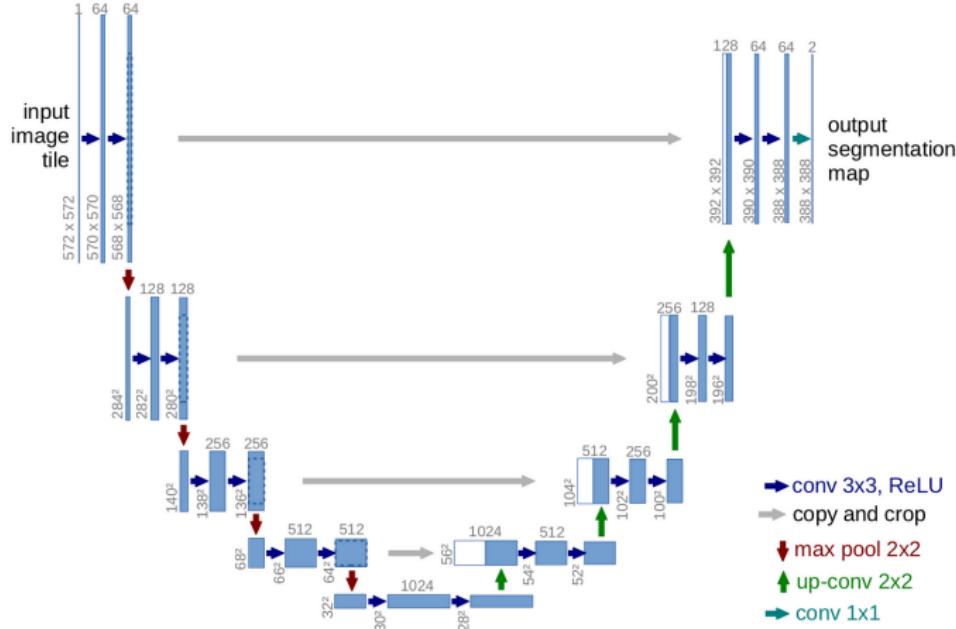
De la même manière, on supposera donc que :

$$\mu_\theta(x_t) = \frac{x_t}{\sqrt{\alpha_t}} + \frac{(1 - \alpha_t)\sqrt{1 - \bar{\alpha}_t} \epsilon_\theta}{1 - \bar{\alpha}_t}$$

- Donc on peut réécrire la loss de KL en fonction de $\|\epsilon_\theta - \epsilon\|^2$.
- On rappelle que $\Sigma_\theta(x_t) = \Sigma_q(x_t, x_0) = \frac{(1 - \alpha_t)(1 - \alpha_{t-1})}{1 - \bar{\alpha}_t} \times I$
- Il est naturel de considérer comme équation Backward de diffusion :

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta \right) + \sqrt{\beta_t} \times \epsilon_t$$

Architecture du U-Net

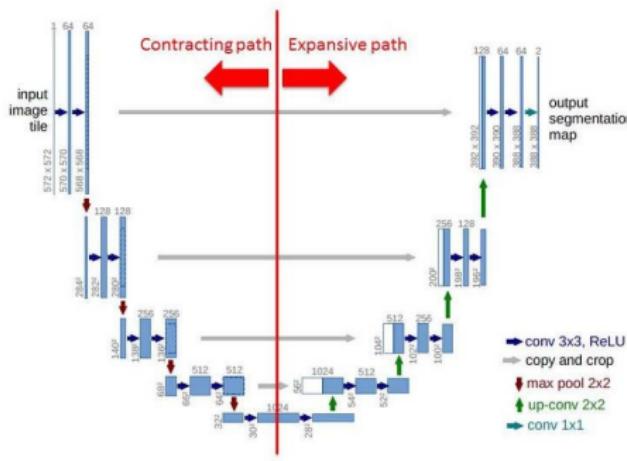


Caractéristiques du U-Net

Il est constitué de deux parties :

- Un encodeur(chemin de contraction) qui sert à capturer le contexte de l'image.
- Un décodeur(chemin d'expansion) qui permet une localisation précise grâce à la convolution transposée.

Network Architecture



Plan

1 Introduction

2 Modèles génératifs de diffusion : Fondements mathématiques

- Idée principale
- Modèle de diffusion
- Analogie avec les VAE
- Réseaux de neurones U-Net

3 Application à la génération d'images MNIST

- Avec le réseau UNET
- Réseaux de neurones débruiteurs préentraînés

4 Conclusion

Données MNIST

- Le dataset MNIST est un dataset de 60000 images de chiffres écrits à la main.
- Il a été très utilisé pour tester les performances des modèles de machine learning.
- Chaque image est de taille 32×32 pixels.

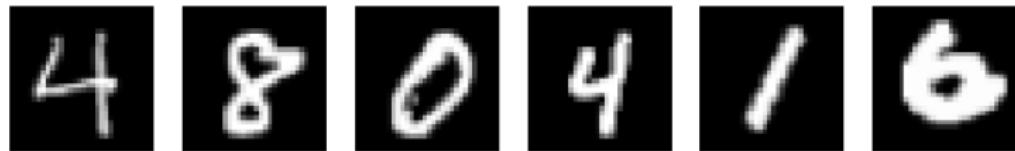


Figure: Exemple d'images du dataset MNIST

Notre Forward

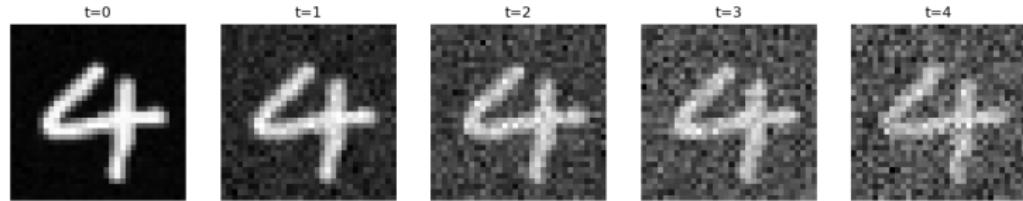


Figure: Images bruités avec le modèle de diffusion forward

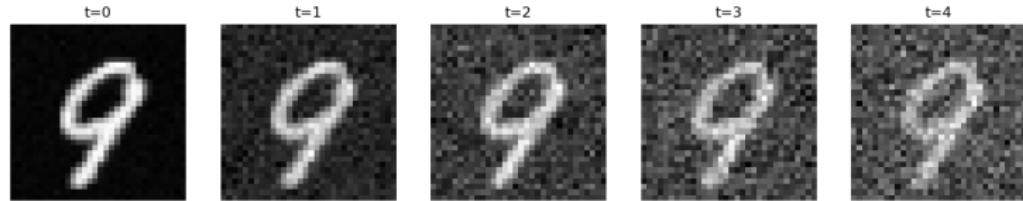
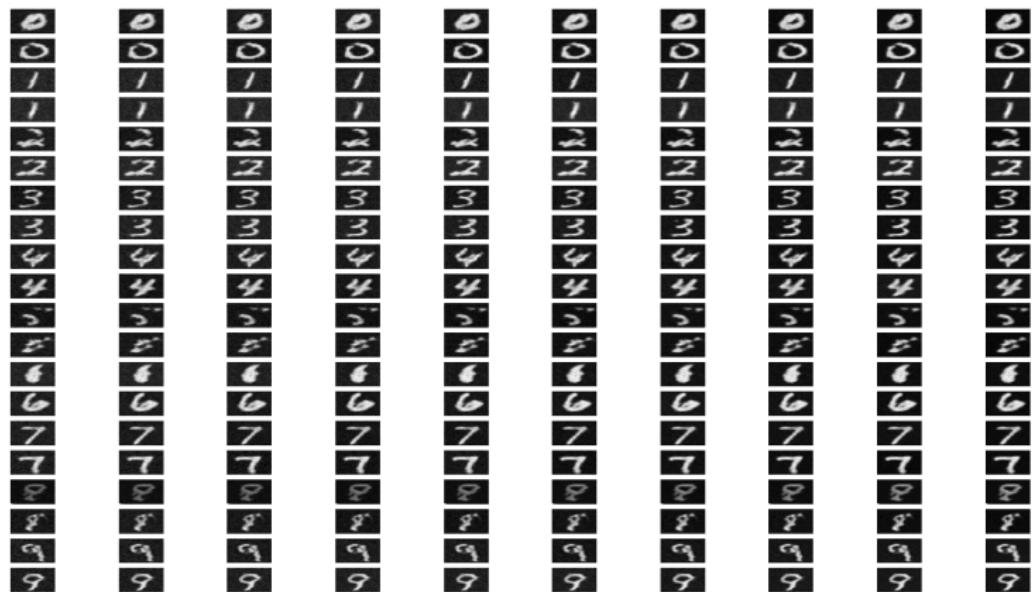


Figure: Images bruités avec le modèle de diffusion forward

Images générées avec le Backward diffusion

Voici quelques images générées en partant d'un bruit blanc et en débruitant au fur et à mesure:



On peut maintenant se poser la question de savoir si ces images sont totalement nouvelles ou si elles ressemblent à des images déjà existantes.

Images générées avec le Backward diffusion

Voici quelques images générées en partant d'un bruit blanc et en débruitant au fur et à mesure:



On peut maintenant se poser la question de savoir si ces images sont totalement nouvelles ou si elles ressemblent à des images déjà existantes.

Comparaison des images générées avec des images du dataset



Figure: Images générées à gauche, et images les plus proches du dataset à droite

Expérimentation sur un modèle de débruitage préentraîné

- On a essayé d'utiliser, un modèle de débruitage préentraîné nommé KBNet : Kernel Basis Network for Image Restoration..
- C'est un modèle qui renvoie une image débruitée contrairement à notre réseau UNET qui renvoie une estimation du bruit dans une image.
- Ainsi afin d'estimer le bruit, on a essayé de soustraire l'image débruitée à l'image bruitée.

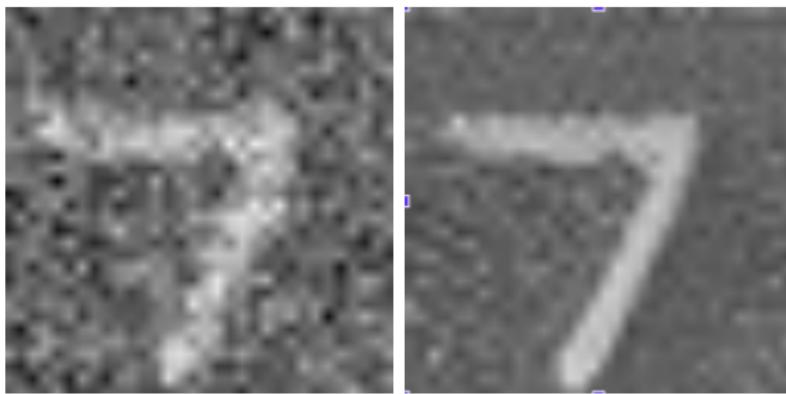
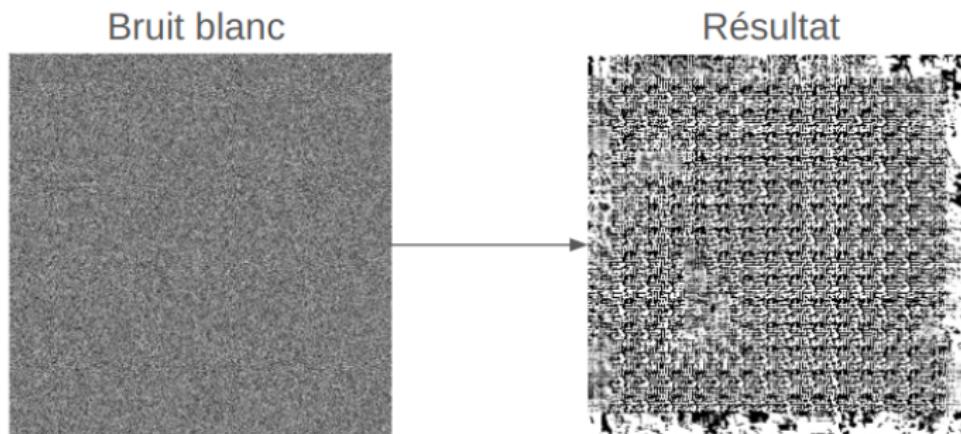


Figure: Image bruitée à gauche, et image débruitée à droite

Expérimentation sur un modèle de débruitage préentraîné

- Nous avons ensuite implémenté l'algorithme permettant de générer une image à partir d'une estimation. Pour cela, nous avons réalisé un script qui réalise des successions de bruitage et de débruitage sur l'image.



- Nous n'avons donc pas obtenu ici, des résultats satisfaisants. Pour expliquer cela, nous pensons que notre algorithme de débruitage estime finalement un bruit global de l'image à la différence d'UNET. (une normalisation de bruit n'est ici pas suffisante.)

Plan

1 Introduction

2 Modèles génératifs de diffusion : Fondements mathématiques

- Idée principale
- Modèle de diffusion
- Analogie avec les VAE
- Réseaux de neurones U-Net

3 Application à la génération d'images MNIST

- Avec le réseau UNET
- Réseaux de neurones débruiteurs préentraînés

4 Conclusion

- Les modèles de diffusions étudiés dans ces projets fonctionnent assez bien sur notre jeu de données MNIST.
- Avantages :
 - Le modèle de diffusion arrive bien à générer des images qui ne se trouvaient pas initialement dans notre Dataset d'origine.
 - Le modèle de diffusion arrive bien à apprendre la distribution de données sur lequel on l'entraîne.
- Inconvénients :
 - Le modèle de diffusion mélange parfois des chiffres pendant la génération.
 - Le modèle de diffusion peut mettre beaucoup de temps à converger selon la complexité de la distribution qu'on veut apprendre.