

Modèle de Diffusion

Machine Learning pour Physicien - Projet

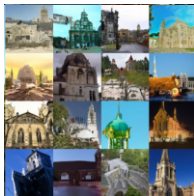
Clément, Grégoire, Eliot

January 20, 2025

Modèle de diffusion

Présentation générale

Les modèles de diffusion permettent de générer des images distribuées **de la même manière** qu'un dataset donné.



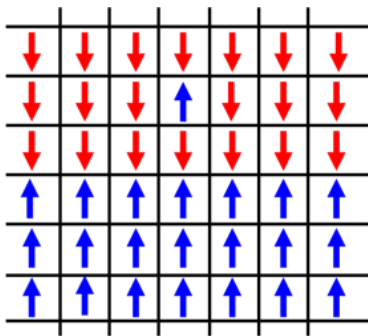
Dataset



Images générées

Figure 1: Source : Denoising Diffusion Probabilistic Models, Ho. and Al.

Exemple d'application physique : modèle d'Ising 2D



$$H(\sigma) = - \sum_{\langle i,j \rangle} J_{ij} \sigma_i \sigma_j - \mu \sum_j h_j \sigma_j$$

Exemple d'application physique : modèle d'Ising 2D

A température T , on a $p(\vec{s}) = \frac{1}{Z} \exp\left(-\frac{E(\vec{s})}{k_B T}\right)$

\implies On veut échantillonner des états probables du système

But : Trouver des schémas récurrents d'alignement de spins de matériaux magnétiques, Calculer des grandeurs moyennes caractéristiques...

Méthode : Metropolis Hasting / Diffusion

\implies Etats discret MH / Etats Continus Diffusion

Exemple d'application physique : Débruitage IRM

Les signaux en physique sont naturellement bruités :

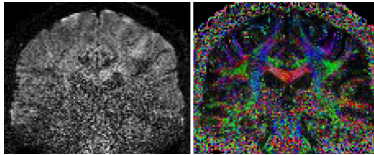


Figure 2: IRM avant débruitage

Utiliser de la diffusion permet de réduire ce bruit pour rendre les photos interprétables :

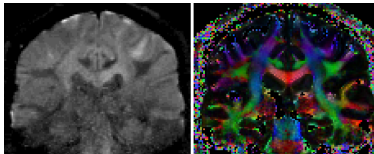


Figure 3: IRM débruité

Application en physique

Eliot si tu peux rajouter quelques trucs ici, genre des exemples avec des images

Denoising Diffusion Probabilistic Models (DDPM), théorie

Idée Générale

Considérons un ensemble de chiffres écrits à la main D . Peut-on trouver une distribution de probabilité q telle que $x \sim q(x)$?

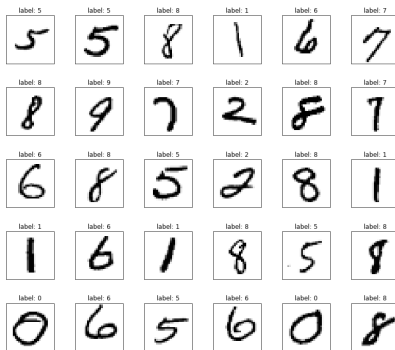
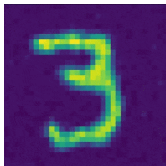


Figure 4: Source: ludwig.ai

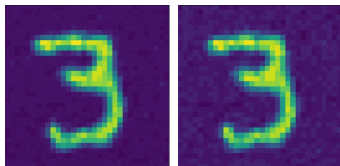
Idée Générale

Considérons l'ensemble de chiffres écrits à la main D . Est-il difficile de trouver q tel que $x \sim q(x)$, nous avons besoin d'une manière plus intelligente d'échantillonner nos chiffres écrits à la main. Examinons le procédé suivant:



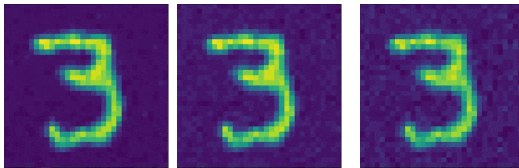
Idée Générale

Considérons l'ensemble de chiffres écrits à la main D . Est-il difficile de trouver q tel que $x \sim q(x)$, nous avons besoin d'une manière plus intelligente d'échantillonner nos chiffres écrits à la main. Examinons le procédé suivant:



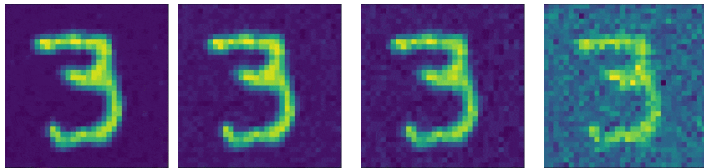
Idée Générale

Considérons l'ensemble de chiffres écrits à la main D . Est-il difficile de trouver q tel que $x \sim q(x)$, nous avons besoin d'une manière plus intelligente d'échantillonner nos chiffres écrits à la main. Examinons le procédé suivant:



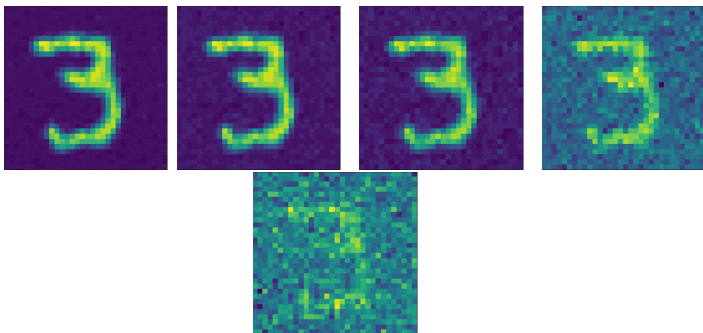
Idée Générale

Considérons l'ensemble de chiffres écrits à la main D . Est-il difficile de trouver q tel que $x \sim q(x)$, nous avons besoin d'une manière plus intelligente d'échantillonner nos chiffres écrits à la main. Examinons le procédé suivant:



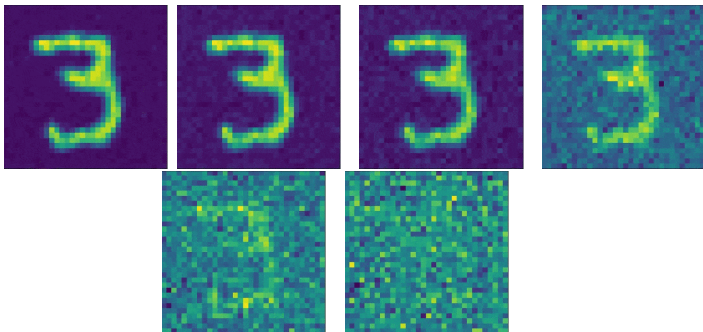
Idée Générale

Considérons l'ensemble de chiffres écrits à la main D . Est-il difficile de trouver q tel que $x \sim q(x)$, nous avons besoin d'une manière plus intelligente d'échantillonner nos chiffres écrits à la main. Examinons le procédé suivant:



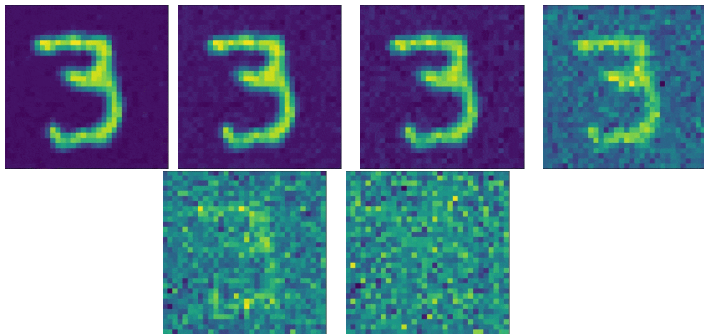
Idée Générale

Considérons l'ensemble de chiffres écrits à la main D . Est-il difficile de trouver q tel que $x \sim q(x)$, nous avons besoin d'une manière plus intelligente d'échantillonner nos chiffres écrits à la main. Examinons le procédé suivant:



Idée Générale

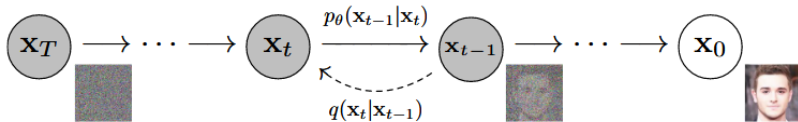
Considérons l'ensemble de chiffres écrits à la main D . Est-il difficile de trouver q tel que $x \sim q(x)$, nous avons besoin d'une manière plus intelligente d'échantillonner nos chiffres écrits à la main. Examinons le procédé suivant:



Formellement : $q(x_{t+1} | x_t) := \mathcal{N}(x_{t+1}; \sqrt{1 - \beta_t}x_t, \beta_t I)$ pour une suite $(\beta_t)_t$. Peut-on apprendre à renverser ce procédé ?

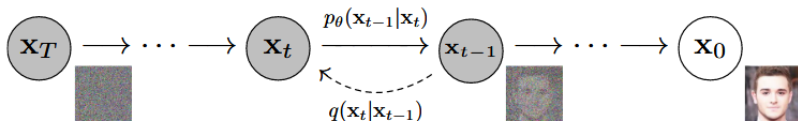
Que veut-on apprendre

En partant d'une image bruitée x_t , on entraîne un modèle pour prédire x_{t-1} .



Que veut-on apprendre

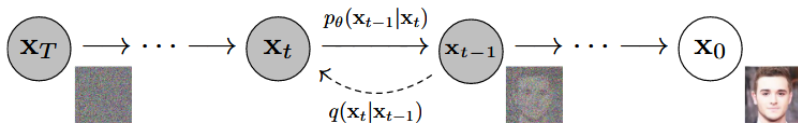
En partant d'une image bruitée x_t , on entraîne un modèle pour prédire x_{t-1} .



- En partant d'un set de données d'images x_0 , on échantillonne $(x_t)_{1:T}$ selon $q(x_{1:T} | x_0) := \prod_{t=1}^T q(x_t | x_{t-1})$,

Que veut-on apprendre

En partant d'une image bruitée x_t , on entraîne un modèle pour prédire x_{t-1} .



- En partant d'un set de données d'images x_0 , on échantillonne $(x_t)_{1:T}$ selon $q(x_{1:T} | x_0) := \prod_{t=1}^T q(x_t | x_{t-1})$,
- Soit une image bruitée x_t et t , on échantillonne selon $p_\theta(x_{t-1} | x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$.

Réduire le coût d'entraînement des données

En partant d'une image x_0 , calculer x_t prend t échantillonnage sur q . Mais une astuce simple, permet de n'en réaliser qu'un.

Réduire le coût d'entraînement des données

En partant d'une image x_0 , calculer x_t prend t échantillonnage sur q . Mais une astuce simple, permet de n'en réaliser qu'un.

Rappelons que $q(x_{t+1} \mid x_t) := \mathcal{N}(x_{t+1}; \sqrt{1 - \beta_t}x_t, \beta_t I)$. Posons $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

Réduire le coût d'entraînement des données

En partant d'une image x_0 , calculer x_t prend t échantillonnage sur q . Mais une astuce simple, permet de n'en réaliser qu'un.

Rappelons que $q(x_{t+1} | x_t) := \mathcal{N}(x_{t+1}; \sqrt{1 - \beta_t}x_t, \beta_t I)$. Posons $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

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

Réduire le coût d'entraînement des données

En partant d'une image x_0 , calculer x_t prend t échantillonnage sur q . Mais une astuce simple, permet de n'en réaliser qu'un.

Rappelons que $q(x_{t+1} | x_t) := \mathcal{N}(x_{t+1}; \sqrt{1 - \beta_t}x_t, \beta_t I)$. Posons $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

$$\begin{aligned}x_t &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\epsilon_{t-1} \\ &= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1 - \alpha_{t-1}}\epsilon_{t-2} + \sqrt{1 - \alpha_t}\epsilon_{t-1}\end{aligned}$$

Réduire le coût d'entraînement des données

En partant d'une image x_0 , calculer x_t prend t échantillonnage sur q . Mais une astuce simple, permet de n'en réaliser qu'un.

Rappelons que $q(x_{t+1} | x_t) := \mathcal{N}(x_{t+1}; \sqrt{1 - \beta_t}x_t, \beta_t I)$. Posons $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

$$\begin{aligned}x_t &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\epsilon_{t-1} \\&= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1 - \alpha_{t-1}}\epsilon_{t-2} + \sqrt{1 - \alpha_t}\epsilon_{t-1}\end{aligned}$$

Posons $G_1 \sim \mathcal{N}(0, \sigma_1^2 I)$, $G_2 \sim \mathcal{N}(0, \sigma_2^2 I)$, la somme des deux donne $g_2 \sim \mathcal{N}(0, (\sigma_1^2 + \sigma_2^2)I)$.

Réduire le coût d'entraînement des données

En partant d'une image x_0 , calculer x_t prend t échantillonnage sur q . Mais une astuce simple, permet de n'en réaliser qu'un.

Rappelons que $q(x_{t+1} | x_t) := \mathcal{N}(x_{t+1}; \sqrt{1 - \beta_t}x_t, \beta_t I)$. Posons $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

$$\begin{aligned}x_t &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\epsilon_{t-1} \\&= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1 - \alpha_{t-1}}\epsilon_{t-1} + \sqrt{1 - \alpha_t}\epsilon_{t-1} \\&= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t(1 - \alpha_{t-1}) + 1 - \alpha_t}\bar{\epsilon}_t\end{aligned}\tag{1}$$

Posons $G_1 \sim \mathcal{N}(0, \sigma_1^2 I)$, $G_2 \sim \mathcal{N}(0, \sigma_2^2 I)$, la somme des deux donne $g_2 \sim \mathcal{N}(0, (\sigma_1^2 + \sigma_2^2)I)$.

Réduire le coût d'entraînement des données

En partant d'une image x_0 , calculer x_t prend t échantillonnage sur q . Mais une astuce simple, permet de n'en réaliser qu'un.

Rappelons que $q(x_{t+1} | x_t) := \mathcal{N}(x_{t+1}; \sqrt{1 - \beta_t}x_t, \beta_t I)$. Posons $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

$$\begin{aligned}x_t &= \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\epsilon_{t-1} \\&= \sqrt{\alpha_t}\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1 - \alpha_{t-1}}\epsilon_{t-1} + \sqrt{1 - \alpha_t}\epsilon_{t-1} \\&= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t(1 - \alpha_{t-1}) + 1 - \alpha_t}\bar{\epsilon}_t \\&= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\bar{\epsilon}_t\end{aligned}\tag{1}$$

Posons $G_1 \sim \mathcal{N}(0, \sigma_1^2 I)$, $G_2 \sim \mathcal{N}(0, \sigma_2^2 I)$, la somme des deux donne $g_2 \sim \mathcal{N}(0, (\sigma_1^2 + \sigma_2^2)I)$.

Réduire le coût d'entraînement des données

En partant d'une image x_0 , calculer x_t prend t échantillonnage sur q . Mais une astuce simple, permet de n'en réaliser qu'un.

Rappelons que $q(x_{t+1} | x_t) := \mathcal{N}(x_{t+1}; \sqrt{1 - \beta_t}x_t, \beta_t I)$. Posons $\alpha_t = 1 - \beta_t$ et $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

On a $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$.

Pour l'instant, notre modèle apprend μ et Σ , c'est-à-dire qu'on échantillonne selon

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

Fixer Σ_{θ} constant donne le même résultat selon le papier, donc :

$$p_{\theta}(x_{t-1} \mid x_t) := \mathcal{N}(x_{t-1}; \mu_{\theta}(x_t, t), \sigma_t I)$$

$$p_{\theta}(x_{t-1} \mid x_t) := \mathcal{N}(x_{t-1}; \mu_{\theta}(x_t, t), \sigma_t I)$$

La probabilité pour notre modèle de générer x_0 est
 $p_{\theta}(x_0) := \int p_{\theta}(x_{0:T}) dx_{1:T}.$

$$p_{\theta}(x_{t-1} \mid x_t) := \mathcal{N}(x_{t-1}; \mu_{\theta}(x_t, t), \sigma_t I)$$

La probabilité pour notre modèle de générer x_0 est $p_{\theta}(x_0) := \int p_{\theta}(x_{0:T}) dx_{1:T}$. En utilisant le log-vraisemblance, sous certaines approximations, on cherche à minimiser

$$E_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_{\theta}(x_t, t)\|^2 \right]$$

où $\tilde{\mu}$ est la moyenne optimale qui dépend de x_0 que nous ne connaissons pas.

$$p_{\theta}(x_{t-1} \mid x_t) := \mathcal{N}(x_{t-1}; \mu_{\theta}(x_t, t), \sigma_t I)$$

La probabilité pour notre modèle de générer x_0 est $p_{\theta}(x_0) := \int p_{\theta}(x_{0:T}) dx_{1:T}$. En utilisant le log-vraisemblance, sous certaines approximations, on cherche à minimiser

$$E_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_{\theta}(x_t, t)\|^2 \right]$$

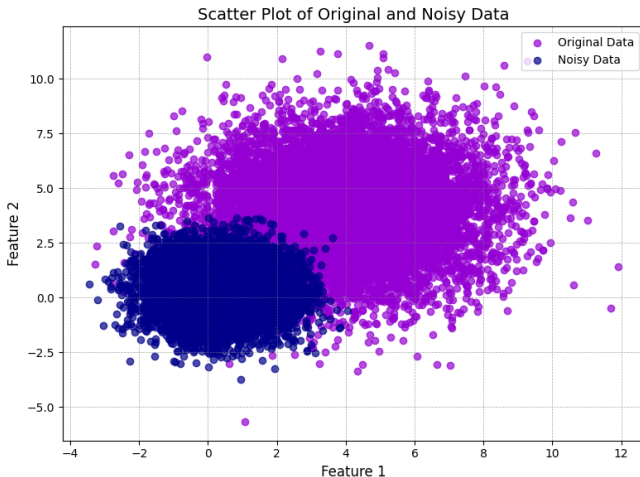
où $\tilde{\mu}$ est la moyenne optimale qui dépend de x_0 . En utilisant $x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$ nous avons une fonction de coût sur laquelle réaliser l'entraînement.

Notre premier modèle : Un simple modèle linéaire

Ici on ajoute un schéma de notre modèle, on parle bien de tout, comment on gère le temps, etc...

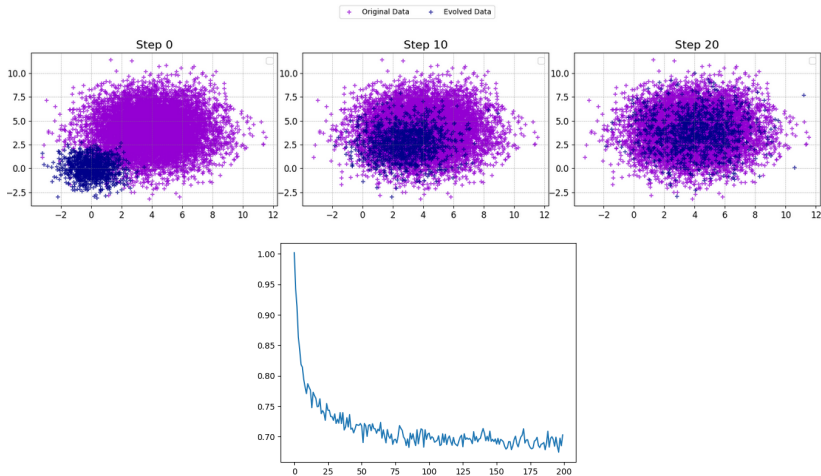
Nos résultats - Une gaussienne

On a commencé en essayant de **générer des gaussiennes**:



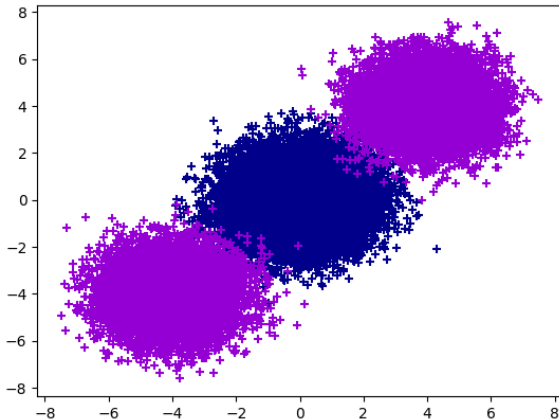
Nos résultats - Une gaussienne

On a commencé en essayant de **générer des gaussiennes** et nous avons eu des résultats encourageants:



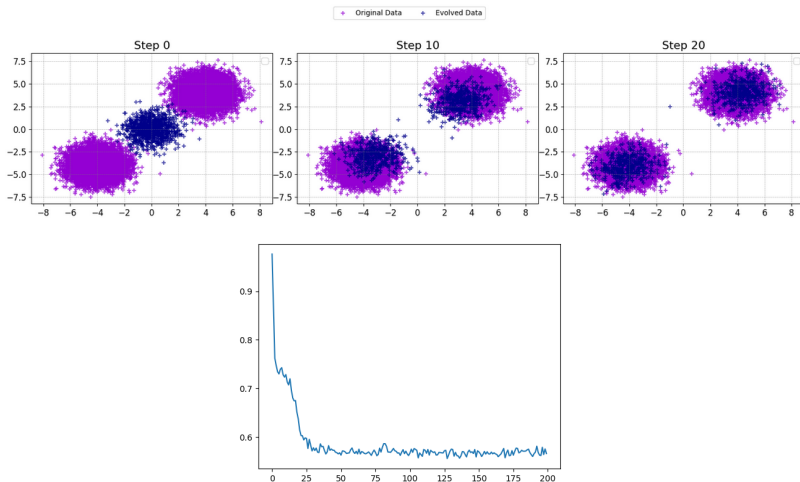
Nos résultats - Deux gaussiennes

On a ensuite essayé avec deux gaussiennes centrées à des endroits différents



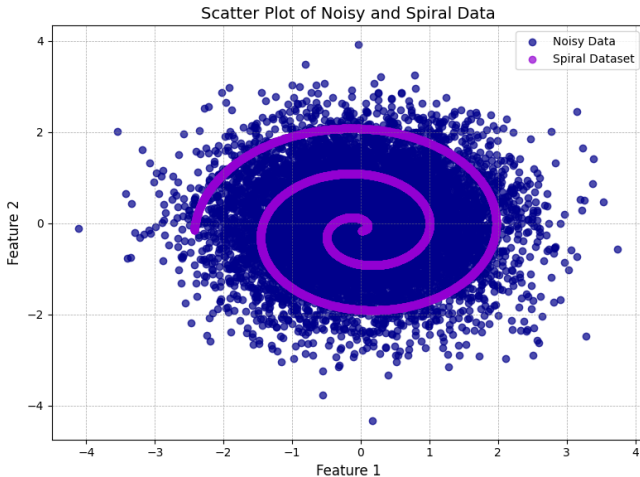
Nos résultats - Deux gaussiennes

On a ensuite essayé avec deux gaussiennes centrées à des endroits différents



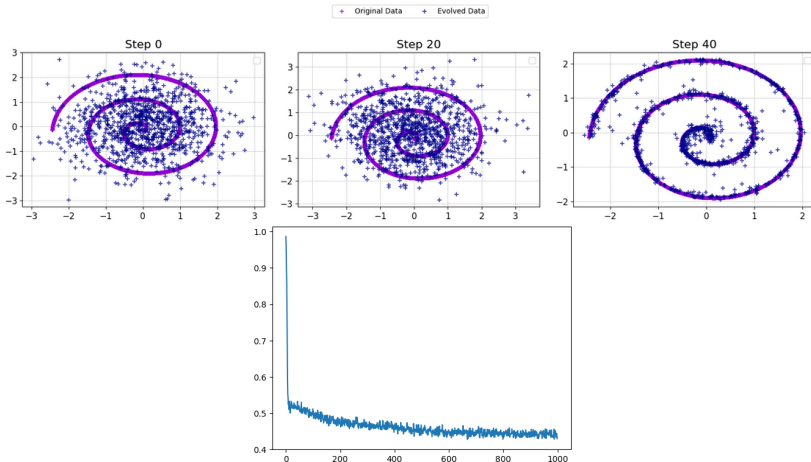
Nos résultats - Spirale

Nous avons ensuite essayé sur un ensemble de données plus compliqué : la génération de spirale.:



Nos résultats - Spirale

Nous avons ensuite essayé sur un ensemble de données plus compliqué : **la génération de spirale**. Nous avons également obtenu des résultats satisfaisants :



Notre deuxième modèle - Un UNet

UNet : Réseau de neurones spécialisé en traitement d'images
→ Fonctionne par convolutions successives

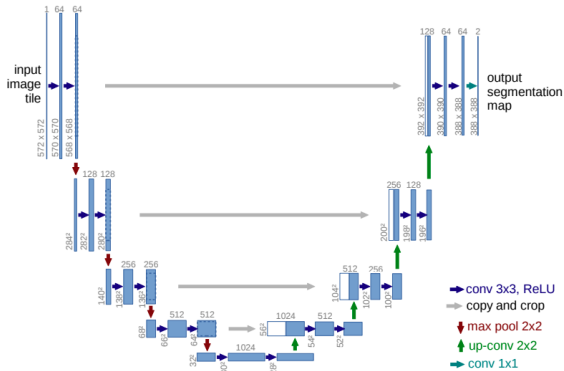


Figure 4: Structure UNet : Contracting path / Expanding path

Contracting Path

But : Capturer des informations précises

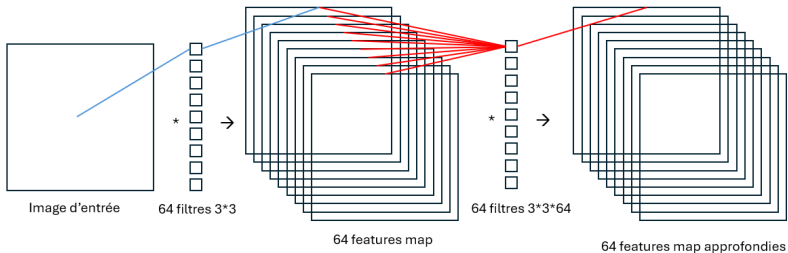


Figure 5: Contracting path, première couche

On applique ensuite un Max pooling pour réduire la résolution.

Expanding Path

But : Reconstruire l'image en mêlant détails et caractéristiques précises

- On réalise le chemin inverse
- Un "Skip connection" permet de rajouter des détails à la reconstruction

