

Projet de Modélisation par Diffusion

Implémentation et Analyse de Modèles Génératifs

Lesech Erwann, Le Riboter Aymeric

November 29, 2025

Contents

1	Introduction	2
2	Objectifs du Projet	2
2.1	Objectifs Principaux	2
2.2	Objectifs Avancés	2
3	Méthodologie	2
3.1	Phase 1: Diffusion 2D	2
3.2	Phase 2: Images MNIST 16×16	2
3.3	Phase 3: Multi-classes MNIST	2
3.4	Phase 4: Amélioration de la Résolution	3
4	Processus de Diffusion	3
4.1	Le processus de bruitage - noising process	3
4.2	Le processus de débruitage - denoising process	3
4.3	Approche par réseaux de neurones	4
4.4	Fonction de Perte	4
4.5	Architectures de Réseaux	4
5	Métriques d'Évaluation	5
5.1	Métriques Quantitatives	5
5.2	Métriques Qualitatives	5
6	Implémentation des Algorithmes	6
6.1	Algorithme d'entraînement	6
6.2	Algorithme de génération	6
7	Références	7

1 Introduction

Les modèles de diffusion représentent une classe d’algorithmes génératifs qui ont révolutionné le domaine de la génération d’images et de données. Ces modèles apprennent à générer des échantillons en inversant un processus de bruit gaussien progressif. Ce projet vise à implémenter et analyser différentes variantes de modèles de diffusion, depuis des cas bidimensionnels simples jusqu’à la génération d’images complexes.

2 Objectifs du Projet

2.1 Objectifs Principaux

- Comprendre les fondements théoriques des modèles de diffusion
- Implémenter un modèle de diffusion sur des données 2D
- Étendre l’implémentation à la génération d’images MNIST
- Analyser les performances selon différentes architectures et hyperparamètres
- Évaluer la qualité des échantillons générés

2.2 Objectifs Avancés

- Implémenter des optimisations décrites dans la littérature récente
- Expérimenter avec des images de plus haute résolution (32×32)
- Explorer la génération d’images RGB (3 canaux)

3 Méthodologie

3.1 Phase 1: Diffusion 2D

La première phase du projet consiste à implémenter un modèle de diffusion sur des données bidimensionnelles pour visualiser et comprendre le processus.

Données: Distributions 2D simples (gaussiennes, spirales, etc.) **Visualisation:** Plots du processus de débruitage étape par étape

3.2 Phase 2: Images MNIST 16×16

Extension à la génération d’images en commençant par une seule classe (chiffre 0) du dataset MNIST redimensionné en 16×16 pixels.

Architecture: Réseau de neurones adapté aux images de petite taille **Processus:** Entraînement du modèle en bruitant progressivement les images selon $\mathcal{N}(0, 1)$

3.3 Phase 3: Multi-classes MNIST

Expansion à plusieurs classes différentes pour analyser la capacité du modèle à distinguer et générer différents types d’images.

Analyse: Étude de l’impact des différentes distributions sur la génération

3.4 Phase 4: Amélioration de la Résolution

Si le temps le permet et que l'entraînement est efficace:

- Passage à des images 32×32 pixels
- Expérimentation avec des images RGB (3 canaux)

4 Processus de Diffusion

4.1 Le processus de bruitage - noising process

Prenons une donnée x_0 . Nous allons la brouiller de manière à suivre une dynamique de Langevin:

$$\begin{aligned} x_t &= \sqrt{\sigma}x_{t-1} + \sqrt{1-\sigma}\xi \\ &\sim \mathcal{N}(\sqrt{\sigma}x_{t-1}, (1-\sigma)I) \end{aligned}$$

Nous pouvons même exprimer x_t en fonction de x_0 :

$$\begin{aligned} x_t &= \sqrt{\prod_{i=0}^{t-1} \sigma_i} x_0 + \sqrt{1 - \prod_{i=0}^{t-1} \sigma_i} \xi \\ &\sim \mathcal{N}\left(\sqrt{\prod_{i=0}^{t-1} \sigma_i} x_0, 1 - \prod_{i=0}^{t-1} \sigma_i I\right) = \mathcal{N}(\sqrt{\bar{\sigma}_t} x_0, 1 - \bar{\sigma}_t I) \end{aligned}$$

où $\bar{\sigma}_t = \prod_{i=0}^{t-1} \sigma_i$ représente le produit cumulé des coefficients de bruit et ξ est un bruit gaussien $\mathcal{N}(0, I)$.

4.2 Le processus de débruitage - denoising process

L'objectif principal du modèle de diffusion est d'inverser le processus de bruitage. Le processus inverse doit également suivre une dynamique stochastique contrôlée.

Le défi du débruitage: Contrairement au processus de bruitage qui est direct et déterministe, le débruitage présente une complexité fondamentale:

- **Bruitage (forward process):** Générer x_t selon $\mathcal{N}(\sqrt{\bar{\sigma}_t} x_0, 1 - \bar{\sigma}_t I)$ est direct car nous connaissons x_0
- **Débruitage (reverse process):** Générer x_{t-1} à partir de x_t nécessite de connaître la moyenne conditionnelle $\mu(x_t, t)$

Problématique centrale: Comment déterminer $\mu(x_t, t)$ pour le processus inverse?

Nous devons générer x_{t-1} selon une distribution $\mathcal{N}(\mu(x_t, t), \sigma^2 I)$ où $\mu(x_t, t)$ représente la moyenne conditionnelle optimale que nous devons apprendre.

Problème: La fonction $\mu(x_t, t)$ n'est pas connue analytiquement et doit être approximée.

4.3 Approche par réseaux de neurones

C'est à cette étape que nous passons de la théorie probabiliste à l'apprentissage automatique pratique. L'idée clé est d'utiliser un réseau de neurones pour approximer la fonction $\mu(x_t, t)$ inconnue.

Reformulation du problème: Plutôt que de prédire directement $\mu(x_t, t)$, nous utilisons une transformation mathématique élégante. En partant de la relation:

$$x_t = \sqrt{\sigma_t}x_0 + \sqrt{1 - \sigma_t}\xi_0$$

On peut montrer (par calcul variationnel) que la moyenne optimale s'exprime comme:

$$\mu(x_t, t) = \frac{1}{\sigma_t}x_t - \frac{1 - \sigma_t}{\sqrt{1 - \sigma_t}}\xi_0$$

Stratégie d'apprentissage: Au lieu de prédire $\mu(x_t, t)$ directement, nous entraînons un réseau $\epsilon_\theta(x_t, t)$ à prédire le bruit ξ_0 qui a été ajouté lors du processus de bruitage. Cette approche est plus stable et efficace en pratique.

Avantage: Prédire le bruit est plus simple car il s'agit d'une propriété intrinsèque du processus de diffusion, indépendante de la distribution des données originales.

4.4 Fonction de Perte

La fonction de perte utilisée pour l'entraînement est une erreur quadratique moyenne (MSE) entre le bruit réel et le bruit prédict:

$$L = \mathbb{E}_{t, x_0, \epsilon} [||\epsilon - \epsilon_\theta(x_t, t)||^2]$$

Composants de la fonction de perte:

- $\epsilon \sim \mathcal{N}(0, I)$: le bruit gaussien réellement ajouté lors du bruitage
- $\epsilon_\theta(x_t, t)$: la prédiction du réseau de neurones
- t : le pas de temps, échantillonné uniformément dans $\{1, \dots, T\}$
- x_0 : l'échantillon original de données

Interprétation: Cette fonction de perte encourage le réseau à apprendre à "deviner" quel bruit a été ajouté à chaque étape, ce qui lui permet ensuite de l'enlever lors de la génération.

4.5 Architectures de Réseaux

Choix d'architecture selon le type de données:

- **U-Net modifié:** Pour les images (MNIST 16×16 , 32×32)
 - Encoder-decoder avec connexions résiduelles
 - Intégration de l'embedding temporel à chaque couche
 - Normalisation par groupes (GroupNorm)
- **MLP avec embedding temporel:** Pour les données 2D simples

- Couche denses avec activations ReLU/GELU
- Embedding sinusoïdal pour encoder le temps t
- Normalisation et dropout pour la régularisation
- **Mécanismes d'attention:** Pour les versions avancées
 - Self-attention pour capturer les dépendances spatiales
 - Cross-attention pour conditionner sur des labels

5 Métriques d'Évaluation

L'évaluation des modèles de diffusion nécessite une approche multi-dimensionnelle combinant métriques quantitatives et qualitatives.

5.1 Métriques Quantitatives

Pour les données 2D:

- **Divergence de Kullback-Leibler (KL):** Mesure la différence entre la distribution apprise et la distribution réelle

$$D_{KL}(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

Utilisation: Comparaison directe des densités de probabilité pour les distributions 2D simples

- **Distance de Wasserstein:** Alternative robuste à la divergence KL *Avantage:* Moins sensible aux queues de distribution

Pour les images:

- **Fréchet Inception Distance (FID):** Métrique standard pour évaluer la qualité des images générées

$$FID = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

où μ_r, Σ_r et μ_g, Σ_g représentent les statistiques des features extraites par un réseau pré-entraîné

- **Inception Score (IS):** Mesure la qualité et la diversité

$$IS = \exp(\mathbb{E}_x[D_{KL}(p(y|x)||p(y))])$$

5.2 Métriques Qualitatives

Analyse visuelle systématique:

- **Inspection des échantillons:** Évaluation de la netteté et du réalisme
- **Analyse de diversité:** Vérification que le modèle ne reproduit pas toujours les mêmes échantillons (mode collapse)

- **Étude de convergence:** Suivi de l'évolution de la perte et de la qualité des échantillons pendant l'entraînement
- **Interpolation:** Test de la continuité de l'espace latent en interpolant entre échantillons

Métriques spécifiques au projet:

- Temps de convergence de l'algorithme d'entraînement
- Stabilité numérique du processus de génération
- Qualité de reconstruction pour différents nombres d'étapes de débruitage

6 Iméplementation des Algorithmes

6.1 Algorithme d'entraînement

Algorithme 1: Entrainement du modèle de diffusion

1. Sélectionner $x_0 \sim p(x_0)$
2. Tirer aléatoirement $t \sim \mathcal{U}(\{1, \dots, T\})$
3. Tirer aléatoirement $\xi_0 \sim \mathcal{N}(0, I)$
4. Calculer $x_t = \sqrt{\sigma_t}x_0 + \sqrt{1 - \sigma_t}\xi_0$
5. **while** loss > seuil **do**
6. Descente de gradient sur $\|\xi_0 - \xi_\theta(x_t, t)\|_2^2$
7. **end while**

6.2 Algorithme de génération

Algorithme 2: Génération d'échantillons via le modèle de diffusion

1. Tirer aléatoirement $x_T \sim \mathcal{N}(0, I)$
2. **for** $t \in \{T, \dots, 1\}$ **do**
3. **Si** $t > 1$, tirer aléatoirement $z \sim \mathcal{N}(0, I)$, sinon $z = 0$.
4. $x_{t-1} = \frac{1}{\sqrt{\sigma_t}} \left(x_t - \frac{1 - \sigma_t}{\sqrt{1 - \sigma_t}} \xi_\theta(x_t, t) \right) + \beta_t z$
5. **end for**

7 Références

References

- [1] Karras, T., Aittala, M., Aila, T., & Laine, S. (2022). *Elucidating the Design Space of Diffusion-Based Generative Models*. arXiv preprint arXiv:2206.00364.
- [2] Ho, J., Jain, A., & Abbeel, P. (2020). *Denoising diffusion probabilistic models*. Advances in neural information processing systems, 33, 6840-6851.
- [3] Song, J., Meng, C., & Ermon, S. (2020). *Denoising diffusion implicit models*. arXiv preprint arXiv:2010.02502.