Diffusion Model

Clément Rouvroy
ENS Paris
rouvroy@phare.normalesup.org

Grégoire Le Corre
ENS Paris
gregoire.le.corre@ens.psl.eu

Nathan Boyer
ENS Paris
nathan.boyer@ens.psl.eu

January 19, 2025

1 Introduction and motivation

Given a dataset of vectors $\mathcal{X}$, the goal of diffusion is to have an algorithm that allows sampling $x \sim p(x)$ such that p is near the distribution of $\mathcal{X}$. As the distribution of $\mathcal{X}$ is unknown, it is a hard problem (can you give the distribution of a set of human faces ?). This problem is interesting but can also be crucial in some applications. For example, in Data generation [Tra+23], which is needed in medical Deep Learning because of the cost to produce a small amount of data [Tor+24].

2 Diffusion Probabilistic Model

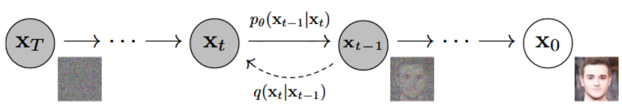

Figure 1: Illustration of the diffusion process

DDPM, presented in [HJA20], is a solution to the problem presented above. The main idea, presented on Fig. 1, is to build for any $x_0 \in \mathcal{X}$ a sequence $(x_t)_{0\leq t\leq T}$ such that for any $t \in [\![2; T]\!]$, x_t is obtained from x_{t-1} by adding a Gaussian noice to it: $\mathcal{N}(x_{t+1}; \sqrt{1-\beta_t}x_t, \beta_t I)$, where $(\beta_t)_{1\leq t\leq T}$ is a fixed sequences to be fixed. Using this process, rather than learning to generate an element of $\mathcal{X}$ from a white noise, they learn a model p_θ that given (x_t, t) learns to predict x_{t-1}. They parametrisze p_θ to be $\mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$ and fixed Σ_θ to a constant to ease the computations. Using maths and Σ_θ constant, they found that they can just learn ϵ (the noise added from x_{t-1} to x_t) and optimize the loss (where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=0}^{t} \alpha_i, \beta_t = \prod_{i=0}^{t} \beta_i$):

$$E_q\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|\right] \tag{1}$$

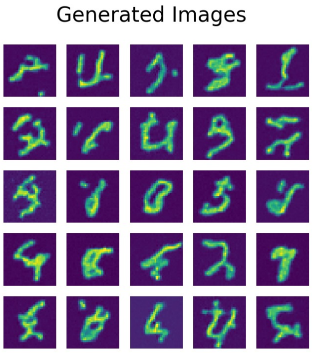

Figure 2: Our generation of hand-written digits

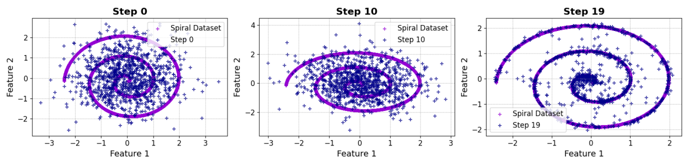

Figure 3: Diffusion visualization on Spirale

We have implemented (https://github.com/crdevio/DiffusionModel) this model, based on the code of Dataflowr (Marc Lelarge), and tested it on three distributions: Gaussian, Spirale, and MNIST. One can see on Fig. 2 our generation of hand-written digits and on Fig. 3 a visualization of our diffusion implementation.

3 Ameliorations

Some datasets need a $T >> 1000$ to perform well, but you may want to have a trade-off between quality and cost of generation. An easy trick is to see that our model is learning to generate ϵ_t such that $x_t = x_0 + \sigma_{t,0}\epsilon_t$ (reparametrization trick in [HJA20]). From this equation, one can get an approximation of x_0, $\tilde{x_0} = x_t - \sigma_{t,0}\epsilon_t$ and generate $\tilde{x}_{t-k} = \tilde{x_0} + \epsilon_{t-k}$. Hence, by fixing $k \geq 1$ one can generate an image in T/k steps.

In [ND21] and [DN21], OpenAI explored many upgrades to the standard DDPM, one of the main ideas is to let Σ_θ be learned between β_t and $\tilde{\beta}_t$ (the two limits found by [HJA20]).

Diffusion was proposed as a replacement for GANs (though GANs are still used, as witnessed by the Test of Time award from NeurIps) and GANs have access to labeled data to train. Hence, in [DN21], labels were used to train Diffusion models. The idea is to sample according to a Gaussian that now also depends on $\nabla \log p(y \mid x_t)$ where y is the desired label. The intuition behind that comes from Langevin Dynamics, which allows one to do a Markov process depending on $\nabla \log p(x)$ to sample from $p(x)$ without knowing $p(x)$. Here, a pre-trained classifier can estimate $\nabla \log p(x)$. However, for some data it is too hard to train a classifier, hence [HS22] introduces a classifier-free guidance that can perform the same results without needing a classifier, it relies on training simultaneously, two diffusion models, one that knows the label of data and one that does not.

Using the building block of classifier-free guidance, OpenAI introduced CLIP [Rad+21] that don't take labels as input but prompts. The main idea is to replace the label with a vector of representation (it is called Representation Learning, [BCV14]) and to also have a model that takes an image andreturnsn a vector of the same dimension. Then they train a model to give the distance between two learned vectors, and they use its gradient to direct the image generation to a prompt.

State-of-the-art models, such as ControlNet [ZRA23] accept more inputs, such as sketch, or canye edges to draw from.

References

[BCV14] Yoshua Bengio, Aaron Courville, and Pascal Vincent. *Representation Learning: A Review and New Perspectives*. 2014. arXiv: 1206.5538 [cs.LG]. URL: https://arxiv.org/abs/1206.5538.

[DN21] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. June 2021. DOI: 10.48550/arXiv.2105.05233. arXiv: 2105.05233 [cs]. URL: http://arxiv.org/abs/2105.05233 (visited on 01/01/2025).

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. Dec. 2020. DOI: 10.48550/arXiv.2006.11239. arXiv: 2006.11239 [cs]. URL: http://arxiv.org/abs/2006.11239 (visited on 01/01/2025).

[HS22] Jonathan Ho and Tim Salimans. *Classifier-Free Diffusion Guidance*. July 2022. DOI: 10.48550/arXiv.2207.12598. arXiv: 2207.12598 [cs]. URL: http://arxiv.org/abs/2207.12598 (visited on 01/01/2025).

[ND21] Alex Nichol and Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. Feb. 2021. DOI: 10.48550/arXiv.2102.09672. arXiv: 2102.09672 [cs]. URL: http://arxiv.org/abs/2102.09672 (visited on 01/09/2025).

[Rad+21] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[Tor+24] Adrian Tormos et al. *Data Augmentation with Diffusion Models for Colon Polyp Localization on the Low Data Regime: How much real data is enough?* 2024. arXiv: 2411.18926 [cs.CV]. URL: https://arxiv.org/abs/2411.18926.

[Tra+23] Brandon Trabucco et al. *Effective Data Augmentation With Diffusion Models*. 2023. arXiv: 2302.07944 [cs.CV]. URL: https://arxiv.org/abs/2302.07944.

[ZRA23] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. *Adding Conditional Control to Text-to-Image Diffusion Models*. Nov. 2023. DOI: 10.48550/arXiv.2302.05543. arXiv: 2302.05543 [cs]. URL: http://arxiv.org/abs/2302.05543 (visited on 01/01/2025).