THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

Introduction to Diffusion Models

MIAO Zhou
The Hong Kong Polytechnic University
January 6, 2025

Contents

Brief Introduction to Diffusion Models

Diffusion models are a class of generative models that progressively transform simple noise into complex data distributions through a series of learned denoising steps.

- These models are inspired by physical diffusion processes, where data is iteratively corrupted by noise and then reconstructed.
- They have shown state-of-the-art performance in image generation, speech synthesis, and other generative tasks.

In this presentation, we will cover the **basic concepts** of typical models and **illustration of code**.

Contents

Definition of ELBO

Generative Model Objective: The primary goal of generative models is to approximate the data distribution $p(\mathbf{x})$, allowing us to model and generate data. The optimization objective is to maximize the log-likelihood of the observed data:

$$\log p(\mathbf{x})$$

Challenge: Directly optimizing $\log p(\mathbf{x})$ is often intractable due to the complexity of marginalizing over latent variables $\mathbf{z}$:

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z})\, d\mathbf{z}.$$

Solution: Evidence Lower Bound (ELBO): To overcome this challenge, we introduce a variational approximation $q(\mathbf{z} \mid \mathbf{x})$ to the true posterior and derive a lower bound:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\frac{\log p(\mathbf{x}, \mathbf{z})}{\log q_\phi(\mathbf{z} \mid \mathbf{x})} \right].$$

The ELBO can be rewritten as:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x} \mid \mathbf{z}) \right] - \mathrm{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z}) \right),$$

where:

- $\log p(\mathbf{x} \mid \mathbf{z})$: Reconstruction term, measuring how well the latent variable explains the data.
- $\mathrm{KL}(q \| p)$: Regularization term, ensuring $q(\mathbf{z} \mid \mathbf{x})$ is close to the prior $p(\mathbf{z})$.

Comparing VAE and Diffusion Models via ELBO

Variational Autoencoders (VAE):

- VAE optimizes the Evidence Lower Bound (ELBO) of the log-likelihood:

$$\text{ELBO}(\mathbf{x}) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log \overbrace{p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})}^{\text{a Gaussian}}]}_{\text{how good your decoder is}} - \underbrace{\mathbb{D}_{\text{KL}}(\overbrace{q_\phi(\mathbf{z} \mid \mathbf{x})}^{\text{a Gaussian}} \parallel \overbrace{p(\mathbf{z})}^{\text{a Gaussian}})}_{\text{how good your encoder is}}.$$

Diffusion Models:

- Diffusion models also optimize a variational lower bound but frame the process as a series of gradual denoising steps:

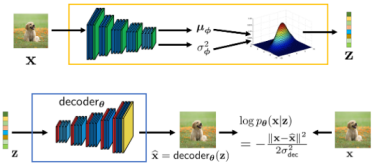
$$\text{ELBO}_{\phi,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{x}_1|\mathbf{x}_0)}[\log \underbrace{p_{\boldsymbol{\theta}}(\mathbf{x}_0 \mid \mathbf{x}_1)}_{\text{how good the initial block is}}] - \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{x}_{T-1}|\mathbf{x}_0)}[\underbrace{\mathbb{D}_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{x}_T \mid \mathbf{x}_{T-1}) \| p(\mathbf{x}_T))}_{\text{how good the final block is}}]$$

$$- \sum_{t=1}^{T-1} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{x}_{t-1},\mathbf{x}_{t+1}|\mathbf{x}_0)}[\underbrace{\mathbb{D}_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \| p_{\boldsymbol{\theta}}(\mathbf{x}_t \mid \mathbf{x}_{t+1}))}_{\text{how good the transition blocks are}}].$$

Architectures of VAE and Diffusion Model

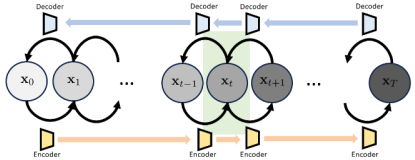
Variational Autoencoders (VAE):

- VAE Architecture:



Diffusion Models:

- Diffusion Model Architecture:



Contents

Introduction to Parameters in DDPM Model

Key Parameters in DDPM:

- T: Total number of timesteps in the diffusion process.
  - Controls the granularity of the diffusion process.
  - A larger T allows for more gradual noise addition and removal but increases computational cost.
- β_t: Noise schedule parameter for each timestep t.
  - Represents the variance of the added noise at timestep t.
  - Typically increases linearly or quadratically over time to ensure smooth noise corruption.
  - Values are bounded: $0 < \beta_t < 1$.
- α_t: Derived from β_t as $\alpha_t = 1 - \beta_t$.
  - Represents the retained signal at timestep t.
  - $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ accumulates the retained signal across timesteps, helping to express intermediate distributions.

Formulas of forward and backward process

Conditional distribution $q_\phi(\mathrm{x}_t \mid \mathbf{x}_0)$. The conditional distribution $q_\phi(\mathbf{x}_t \mid \mathbf{x}_0)$ is given by

$$q_\phi(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t \mid \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \quad (1-\bar{\alpha}_t)\,\mathbf{I}\right)$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.
The distribution $q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$ takes the form of

$$q_\phi(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0), \boldsymbol{\Sigma}_q(t)\right)$$

where

$$\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{(1-\alpha_t)\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}\mathbf{x}_0$$

$$\boldsymbol{\Sigma}_q(t) = \frac{(1-\alpha_t)(1-\sqrt{\bar{\alpha}_{t-1}})}{1-\bar{\alpha}_t}\mathbf{I} \stackrel{\text{def}}{=} \sigma_q^2(t)\mathbf{I}.$$

Training and Inference of DDPM

The loss function for a denoising diffusion probabilistic model:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{t=1}^{T} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2\,\bar{\alpha}_{t-1}}{(1-\bar{\alpha}_t)^2} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\left\| \widehat{\mathbf{x}}_{\boldsymbol{\theta}}\left(\mathrm{x}_t\right) - \mathbf{x}_0 \right\|^2 \right]$$

- **Training algorithm**
- Pick a random time stamp $\ell \sim$ Uniform $[1, T]$.
- Draw a sample $\mathbf{x}_t \sim \mathcal{N}\left(\mathbf{x}_t \mid \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_\iota)\,\mathbf{I}\right)$, i.e.,

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{(1-\bar{\alpha}_t)}\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}).$$

- Take the gradient $\nabla_{\boldsymbol{\theta}} \left\| \widehat{\mathbf{x}}_{\boldsymbol{\theta}}\left(\mathbf{x}_t\right) - \mathrm{x}_0 \right\|^2$ to update $\widehat{\mathbf{x}}_{\boldsymbol{\theta}}$

Training and Inference of DDPM

- Inference on a Deniosing Diffusion Probabilistic Model. (Version: Predict image)
- You give us a white noise vector $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
- Repeat the following for $t = T, T-1, \ldots, 1$.
- We calculate $\widehat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t)$ using our trained denoiser.
- Update according to

$$\mathbf{x}_{t-1} = \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{(1-\alpha_t)\sqrt{\alpha_{t-1}}}{1-\bar{\alpha}_t}\widehat{\mathbf{x}}_{\boldsymbol{\theta}}(\mathbf{x}_t) + \sigma_q(t)\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

NON-MARKOVIAN of DDIM

Let us consider a family $\mathcal{Q}$ of inference distributions, indexed by a real vector $\sigma \in \mathbb{R}_{\geq 0}^{T}$:

$$q_\sigma\left(x_{1:T} \mid x_0\right) := q_\sigma\left(x_T \mid x_0\right) \prod_{t=2}^{T} q_\sigma\left(x_{t-1} \mid x_t, x_0\right)$$

where $q_\sigma\left(\boldsymbol{x}_T \mid x_0\right) = \mathcal{N}\left(\sqrt{\alpha_T} x_0, \left(1-\alpha_T\right) \boldsymbol{I}\right)$ and for all $t > 1$,

$$q_\sigma\left(x_{t-1} \mid x_t, x_0\right) = \mathcal{N}\left(\sqrt{\alpha_{t-1}} x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t} x_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 I\right).$$

The mean function is chosen to order to ensure that $q_\sigma\left(x_t \mid x_0\right) = \mathcal{N}\left(\sqrt{\alpha_t} x_0, \left(1-\alpha_t\right) \boldsymbol{I}\right)$ for all t (see Lemma 1 of Appendix B), so that it defines a joint inference distribution that matches the "marginals" as desired. The forward process [3] can be derived from Bayes' rule:

$$q_\sigma\left(x_t \mid x_{t-1}, x_0\right) = \frac{q_\sigma\left(x_{t-1} \mid x_t, x_0\right) q_\sigma\left(x_t \mid x_0\right)}{q_\sigma\left(x_{t-1} \mid x_0\right)}$$

Inference of DDIM

Training function of DDIM is the same as DDPM, sampling function one can generate a sample $\boldsymbol{x}_{t-1}$ from a sample $\boldsymbol{x_t}$ via:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right)}_{\text{"predicted } \boldsymbol{x}_0\text{"}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{"direction pointing to } \boldsymbol{x}_t\text{"}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is standard Gaussian noise independent of $\boldsymbol{x}_t$, and we define $\alpha_0 := 1$. Different choices of σ values results in different generative processes, all while using the same model ϵ_θ, so re-training the model is unnecessary. When $\sigma_t = \sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{1-\alpha_t/\alpha_{t-1}}$ for all t, the forward process becomes Markovian, and the generative process becomes a DDPM.

Contents

Introduction to Langevin Dynamics

The Langevin dynamics for sampling from a known distribution $p(\mathbf{x})$ is an iterative procedure for $t = 1, \ldots, T$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \tau \nabla_{\mathbf{x}} \log p\left(\mathbf{x}_t\right) + \sqrt{2\tau}\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$$

where τ is the step size which users can control, and $\mathbf{x}_0$ is white noise.

Without the noise term $\sqrt{2\tau}\mathbf{z}$ at the end, the Langevin dynamics is literally gradient descent. The descent direction $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ is carefully chosen that $\mathbf{x}_t$ will converge to the distribution $p(\mathbf{x})$.

Loss Function of Score Matching

- **Loss Function of Exact Score Matching (ESM):**

$$\min_{\theta} \mathcal{J}_{\mathrm{ESM}}(\theta) := \mathbb{E}_{p(t,X)} \left[\| s_\theta(t, X) - \nabla \log p(t, X) \|^2 \right]$$

  which expands to:

$$\mathcal{J}_{\mathrm{ESM}}(\theta) = \| s_\theta(t, X) \|^2 - 2\nabla \cdot s_\theta(t, X)$$

- **Loss Function of Sliced Score Matching (SSM):**

$$\min_{\theta} \tilde{\mathcal{J}}_{\mathrm{SSM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}(0,T)} \mathbb{E}_{v \sim \mathcal{N}(0,I)} \mathbb{E}_{p(t,X)} \left[\lambda(t) \left(\| s_\theta(t, X) \|^2 + 2v^\top \nabla (v^\top s_\theta(t, X)) \right) \right]$$

Loss Function of Score Matching

- **Loss Function of Denoising Score Matching(DSM):**

$$\min J_{\mathrm{DSM}}(\boldsymbol{\theta}) \stackrel{\mathrm{def}}{=} \mathbb{E}_{q(\mathbf{x},\mathbf{x}')}\left[\frac{1}{2}\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} q\left(\mathbf{x} \mid \mathbf{x}'\right)\right\|^2\right]$$

which expands to:

$$\mathbb{E}_{q(\mathbf{x}')}\left[\frac{1}{2}\left\|\mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}' + \sigma\mathbf{z}\right) + \frac{\mathbf{z}}{\sigma}\right\|^2\right]$$

Contents

Training and Inference of SDE Diffusion

- Forward Diffusion.

$$d\mathbf{x} = \underbrace{\mathbf{f}(\mathbf{x}, t)}_{\text{drift}}\, dt + \underbrace{g(t)}_{\text{difusion}}\, d\mathbf{w}.$$

- Reverse Diffusion.

$$d\mathbf{x} = [\underbrace{\mathbf{f}(\mathbf{x}, t)}_{\text{drift}} - g(t)^2 \underbrace{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}_{\text{score function}}]dt + \underbrace{g(t)d\overline{\mathbf{w}}}_{\text{reverse-time diffusion}}$$

- Forward Diffusion function of DDPM.

$$d\mathbf{x} = \underbrace{-\frac{\beta(t)}{2}\mathbf{x}}_{=\mathbf{f}(\mathbf{x},t)}\, dt + \underbrace{\sqrt{\beta(t)}}_{=g(t)}\, d\mathbf{w}.$$

- Reverse Diffusion function of DDPM.

$$d\mathbf{x} = -\beta(t)\left[\frac{\mathbf{x}}{2} + \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + \sqrt{\beta(t)}d\overline{\mathbf{w}}$$

Contents

Reference

- **Articles:**
  - Tutorial on Diffusion Models for Imaging and Vision arxiv.
  - SCORE-BASED DIFFUSION MODELS VIA STOCHASTIC DIFFERENTIAL EQUATIONS – A TECHNICAL TUTORIAL arxiv.
- **Github Repositories:**
  - Useful project of DDPM and DDIM based on pytorch Github.
  - Useful project of score matching diffusion based on pytorch Github.
  - All illustration code and PPT pdf Github.
- **Youtubers:**
- Dhiraj Madan Youtube.

Good Luck!

- Thank you for your attention!
- We hope this introduction has been helpful for understanding the basics.
- Wishing you success in your journey with diffusion models!

THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

Opening Minds • Shaping the Future • 啟迪思維 • 成就未來