Part II

DIFFUSION

Table of contents

Forward Diffusion Process

Data-to-noise process

Diffusion models define a *fixed forward process* that gradually destroys structure in the data (Ho et al., 2020; Sohl-Dickstein et al., 2015). In the continuous-time view, this process is described by a stochastic differential equation (Song et al., 2021). Contrary to flow matching notation, here the forward direction is usually

$$X_0 \sim p_{\mathrm{data}}, \qquad X_1 \approx \mathcal{N}(0, I).$$

Continuous-time formulation

- A common choice is the variance-preserving SDE: $dX_t = -\frac{1}{2}\beta(t)X_t\,dt + \sqrt{\beta(t)}\,dW_t.$
- Its marginal corruption has the closed form

$$X_t = \alpha(t)X_0 + \sigma(t)\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \qquad \text{with} \qquad \alpha(t) = \exp\left(-\frac{1}{2}\int_0^t \beta(s)\,ds\right), \quad \sigma^2(t) = 1 - \alpha^2(t).$$

- Note that a common non-variance-preserving choice is the *variance-exploding* SDE, where the signal is not contracted: $\alpha(t) = 1$.

Reverse-Time Generative Dynamics

Forward marginals

The forward corruption is chosen by design: $q_t(x_t \mid x_0) = \mathcal{N}\big(x_t; \alpha(t)x_0, \sigma^2(t)I\big)$. It induces a noisy marginal distribution $p_t(x_t) = \int q_t(x_t \mid x_0)\, p_{\mathrm{data}}(x_0)\, dx_0$.

Why the reverse process must be learned

Generation reverses the noising path:

$$X_1 \sim \mathcal{N}(0, I), \qquad X_1 \longrightarrow X_0 \sim p_{\mathrm{data}}.$$

But the true reverse transition $p(x_{t-\Delta t} \mid x_t)$ depends on the unknown marginal p_t, hence it is not available explicitly.

Reverse-time dynamics

For a forward SDE

$$dX_t = f(t, X_t)\, dt + g(t)\, dW_t,$$

the reverse dynamics use the score $s_t(x) = \nabla_x \log p_t(x)$:

$$dX_t = \big[f(t, X_t) - g^2(t) s_t(X_t)\big]\, dt + g(t) d\overline{W}_t, \qquad t : 1 \to 0.$$

Score Matching Formulation

What the score means

The score $s_t(x) = \nabla_x \log p_t(x)$ points toward directions where the noisy density p_t increases. In the reverse process, it tells the sampler how to move a noisy point toward more likely, more data-like regions.

Denoising score matching

- Since the marginal score $s_t(x)$ is unknown, use the conditional corruption

$$X_t = \alpha(t)X_0 + \sigma(t)\varepsilon.$$

- The conditional score is known:

$$\nabla_{x_t} \log q_t(x_t \mid x_0) = -\frac{x_t - \alpha(t)x_0}{\sigma^2(t)} = -\frac{\varepsilon}{\sigma(t)}.$$

- Train a neural score network by regression:

$$\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathbb{E}_{t,x_0,\varepsilon}\left[\lambda(t)\left\|s_\theta(t, X_t) + \frac{\varepsilon}{\sigma(t)}\right\|^2\right].$$

Sampling Algorithms

Algorithmic view

Training learns a family of denoisers or scores indexed by time. Sampling uses them sequentially, starting from Gaussian noise and progressively removing noise until a data-like sample is obtained.

Algorithm 1 Ancestral DDPM sampling

Require: trained noise predictor ε_θ

1: sample $X_T \sim \mathcal{N}(0, I)$
2: for $t = T, T-1, \ldots, 1$
3: compute

$$\mu_\theta(X_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(X_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\varepsilon_\theta(t, X_t)\right)$$

4: sample $z \sim \mathcal{N}(0, I)$, except $z = 0$ at the last step
5: set $X_{t-1} = \mu_\theta(X_t, t) + \widetilde{\sigma}_t z$
6: return X_0

References for this part I

- Ho, J., Jain, A., & Abbeel, P. (2020).**Denoising diffusion probabilistic models.** *Advances in Neural Information Processing Systems*.
- Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015).**Deep unsupervised learning using nonequilibrium thermodynamics.** *International Conference on Machine Learning*.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021).**Score-based generative modeling through stochastic differential equations.** *International Conference on Learning Representations*.