Lecture 26, 27: Diffusion Models for Generation

Contents

1 Introduction: Generation and Test-Time Compute **2**

1.1 The Manifold Problem . . . 2

1.2 The Intuition of Diffusion . . . 2

2 The Forward Diffusion Process **2**

3 The Reverse Process: Stochastic Sampling (DDPM) **2**

3.1 The Gaussian Assumption . . . 3

3.2 Derivation of the Reverse Mean . . . 3

3.2.1 1. Bayes' Rule Application . . . 3

3.2.2 2. Analyzing the Terms . . . 3

3.2.3 3. Completing the Square . . . 3

4 Training the Diffusion Model **4**

4.1 Training Procedure . . . 4

4.2 Interpretation: Prediction Targets . . . 4

5 Deterministic Sampling (DDIM) **4**

5.1 The Deterministic Intuition . . . 5

5.2 Derivation via the "Zero" Ansatz . . . 5

5.3 ODE Perspective . . . 5

6 Summary of Algorithms **5**

1 Introduction: Generation and Test-Time Compute

Diffusion models represent a paradigm shift from "one-shot" generation (like VAEs or GANs) to iterative generation. This process can be conceptualized as applying **"test-time compute"** to generative modeling, where the model spends computational effort over time to refine a sample.

1.1 The Manifold Problem

Why not just train a classifier and optimize the input? Recall the failure mode of using a "cat classifier" to generate images. If we perform gradient ascent on a random noise image to maximize a "cat score", we obtain adversarial noise rather than a natural image.

- **Reason:** The manifold of natural images is a vanishingly small, low-dimensional subset of the high-dimensional pixel space. A standard classifier is only trained on this manifold and has undefined gradients in the vast regions of noise.
- **Solution:** We need a mechanism to navigate from the high-entropy noise space back to the data manifold.

1.2 The Intuition of Diffusion

To solve the manifold navigation problem, we bridge the gap between "clean desired samples" and "total noise" by filling the intermediate space.

1. **Forward Process (Diffusion):** We inject noise iteratively. Conceptually, imagine injecting a drop of dye into water. Thermal motion diffuses the dye until the distribution reaches equilibrium (uniform/total noise).
2. **Reverse Process (Generation):** We simulate the process in reverse. We start from the equilibrium distribution (pure noise) and iteratively concentrate probability mass back onto the data manifold.

2 The Forward Diffusion Process

We model the forward process as adding Gaussian noise over time $t \in [0, 1]$. We adopt a continuous-time perspective where we take $T \to \infty$ limit via small steps Δt.

Let x_0 be a sample from the data distribution p_{data}. The forward transition from time $t - \Delta t$ to t is modeled as:

$$x_t = x_{t-\Delta t} + \mathcal{N}(0, \sigma_q^2 \Delta t I) \tag{1}$$

where σ_q^2 is a noise variance parameter. As $t \to 1$, x_1 becomes approximately indistinguishable from pure Gaussian noise $\mathcal{N}(0, \Sigma)$.

3 The Reverse Process: Stochastic Sampling (DDPM)

The core of Denoising Diffusion Probabilistic Models (DDPM) is to learn the reverse transition density $p(x_{t-\Delta t}|x_t)$.

3.1 The Gaussian Assumption

While the exact reverse distribution for a general stochastic process is intractable, for diffusion processes with sufficiently small step size Δt and sufficiently smooth data density, the reverse step is approximately Gaussian:

$$p(x_{t-\Delta t}|x_t) \approx \mathcal{N}(x_{t-\Delta t}; \mu_t(x_t), \sigma_q^2 \Delta t I) \tag{2}$$

We assume the variance in the reverse direction matches the forward noise injection. The main challenge is to learn the mean $\mu_t(x_t)$.

3.2 Derivation of the Reverse Mean

We can derive the functional form of the reverse mean using Bayes' Rule and Taylor expansions.

3.2.1 1. Bayes' Rule Application

We start with the conditional probability: 贝叶斯公式: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

$P(X_t|X_{t-\Delta t})$: 前向加噪概率, 已知.

$$p(x_{t-\Delta t}|x_t) = \frac{p(x_t|x_{t-\Delta t})p_{t-\Delta t}(x_{t-\Delta t})}{p_t(x_t)} \tag{3}$$

Taking the logarithm to simplify the product:

$$\log p(x_{t-\Delta t}|x_t) = \log p(x_t|x_{t-\Delta t}) + \log p_{t-\Delta t}(x_{t-\Delta t}) - \underbrace{\log p_t(x_t)}_{\text{const w.r.t } x_{t-\Delta t}} \tag{4}$$

公式左边我们的变量是 $X_{t-\Delta t}$, 与 X_t 无关, 视为 Const 从而可暂时忽略.

3.2.2 2. Analyzing the Terms

The first term corresponds to the forward Gaussian transition kernel:

$$\log p(x_t|x_{t-\Delta t}) \approx -\frac{1}{2\sigma_q^2 \Delta t}||x_t - x_{t-\Delta t}||^2 + C \tag{5}$$

For the second term $\log p_{t-\Delta t}(x_{t-\Delta t})$, we use the smoothness of the density. We approximate $p_{t-\Delta t} \approx p_t$ and perform a first-order Taylor expansion of $\log p_t(\cdot)$ around x_t: 泰勒展开

$$\underbrace{\log p_{t-\Delta t}(x_{t-\Delta t})}_{\text{上一刻位置}} \approx \underbrace{\log p_t(x_t)}_{\text{当前位置}} + \overset{\text{内积}}{\langle} \nabla_x \log p_t(x_t), \underbrace{(x_{t-\Delta t} - x_t)}_{\text{位移}} \rangle + \mathcal{O}(\Delta t) \tag{6}$$

Here, the term $\nabla_x \log p_t(x_t)$ is the **Score Function**.

3.2.3 3. Completing the Square

Substituting these back and collecting terms dependent on $x_{t-\Delta t}$: 把(5)、(6)代入(4), 忽略常数项

$$\log p(x_{t-\Delta t}|x_t) \approx -\frac{1}{2\sigma_q^2 \Delta t}||x_{t-\Delta t} - x_t||^2 + \langle \nabla_x \log p_t(x_t), (x_{t-\Delta t} - x_t) \rangle \tag{7}$$

$$= -\frac{1}{2\sigma_q^2 \Delta t}\left(||x_{t-\Delta t} - x_t||^2 - 2\sigma_q^2 \Delta t \langle \nabla_x \log p_t(x_t), (x_{t-\Delta t} - x_t) \rangle\right) \tag{8}$$

By completing the square, we identify the mean of the Gaussian: 经过配方后:

在概率密度的对数公式里，加减一个常数 C，只影响概率的"高度"（归一化系数），不影响概率分布的"中心位置"（均值）和"形状"（方差）。

$$\log p(x_{t-\Delta t}|x_t) \approx -\frac{1}{2\sigma_q^2 \Delta t}||x_{t-\Delta t} - \underbrace{(x_t + \sigma_q^2 \Delta t \nabla_x \log p_t(x_t))}_{\text{Mean } \mu_t(x_t)}||^2 \tag{9}$$

对比高斯分布标准定义: 对数概率形式为 $-\frac{1}{2\sigma^2}||x-\mu||^2$

所以! $(X_t + \sigma^2 \Delta t \nabla_x \log p_t(X_t))$ 是均值!

Key Result: The Reverse Mean and Score Matching

The mean of the reverse diffusion step moves the sample in the direction of the score (the gradient of the log density):

$$\mu_t(x_t) \approx x_t + \sigma_q^2 \Delta t \nabla_x \log p_t(x_t) \tag{10}$$

To generate samples, we must learn a neural network to estimate this score $\nabla_x \log p_t(x_t)$.

4 Training the Diffusion Model

To perform generation, we train a neural network $f_\theta(x_t, t)$ to approximate the mean $\mu_t(x_t)$.

4.1 Training Procedure

Training is efficient because we can sample x_t directly without simulating the entire chain (due to Gaussian properties).

1. Sample a clean image x_0 from the training set. clean image
2. Sample a time step $t \sim \text{Uniform}[0, 1]$. 训练时，t 是乱跳的
3. Sample noise $\epsilon \sim \mathcal{N}(0, I)$.
4. Construct the noisy image x_t using the closed-form forward property (signal + noise).
5. **Loss Function:** Train $f_\theta(x_t, t)$ to predict the previous step (or equivalently, the noise). The objective is the Squared Error (MSE), which corresponds to maximizing the likelihood of the Gaussian reverse kernel.

4.2 Interpretation: Prediction Targets

Although the derivation points to predicting the mean, practically we can parameterize the network to predict different targets:

- **Predicting Noise:** Since x_t is just $x_{t-\Delta t}$ plus noise, predicting the previous step is mathematically equivalent to predicting the noise instance added.
- **Predicting x_0:** We can view $\mu_t(x_t)$ as pointing towards the clean data. The "expected negative noise" vector effectively points from x_t back to x_0.

Note: Predicting x_0 is often beneficial because the output space (valid images) is well-aligned with the inductive biases of architectures like U-Nets (e.g., convolutions), whereas pure noise is high-frequency and unstructured.

5 Deterministic Sampling (DDIM)

DDPM sampling is inherently stochastic; it injects random noise at every reverse step. However, we can achieve deterministic sampling (Denoising Diffusion Implicit Models - DDIM) by realizing that to generate valid samples, we only need to match the **marginals** $p_t(x)$, not necessarily the joint distribution of the stochastic trajectory.

5.1 The Deterministic Intuition

Consider connecting two Gaussians $X \sim \mathcal{N}(0,1)$ and $Y \sim \mathcal{N}(0,2)$:

- **Stochastic Path:** $Y = X + \epsilon$, where $\epsilon \sim \mathcal{N}(0,1)$. (Multiple Ys possible for one X).
- **Deterministic Path:** $Y = \sqrt{2}X$. (One-to-one mapping).

Both result in the correct marginal distribution for Y. DDIM seeks a deterministic mapping similar to the second case.

5.2 Derivation via the "Zero" Ansatz

$x_t \sim N(0, \sigma^2 t)$, $\mu_t(x_t) = E(x_{t-\Delta t} | x_t)$

We look for an update rule of the form:

去噪的方向 $\mu_t(x_t)$: 给定现在的噪点图 x_t, 上一刻 $x_{t-\Delta t}$ 的均值

$$x_{t-\Delta t} = x_t + \lambda(\mu_t(x_t) - x_t) \tag{11}$$

where $(\mu_t(x_t) - x_t)$ is the estimated "denoising direction". To find λ, we use a guess-and-verify method with a simple proxy distribution: a point mass $p_0 = \delta_0$. $(x_0 = 0)$

等比例: $\frac{\mu_t(x_t)}{x_t} = \frac{t-\Delta t}{t}$

1. If $p_0 = \delta_0$, then $x_t \sim \mathcal{N}(0, t\sigma^2 I)$.
2. The conditional expectation (optimal denoising) is simply linear shrinkage: $\mu_t(x_t) = \frac{t-\Delta t}{t} x_t$.
3. Substituting this into the update rule:
$$x_{t-\Delta t} = x_t + \lambda\left(\frac{t-\Delta t}{t}x_t - x_t\right) = x_t\left(1 - \frac{\lambda \Delta t}{t}\right) \tag{12}$$
4. We enforce that the variance of $x_{t-\Delta t}$ matches the diffusion schedule variance $(t-\Delta t)\sigma^2$:

$Var(kX) = k^2 Var(X)$

$$\mathrm{Var}(x_{t-\Delta t}) = \left(1 - \frac{\lambda \Delta t}{t}\right)^2 t\sigma^2 \overset{!}{=} (t-\Delta t)\sigma^2 \tag{13}$$

Solving for λ, we obtain:

$$\lambda \approx \frac{\sqrt{t}}{\sqrt{t} + \sqrt{t-\Delta t}} \tag{14}$$

This λ allows us to traverse the probability flow deterministically while maintaining the correct marginal variances at each timestep.

5.3 ODE Perspective

In the limit $\Delta t \to 0$, this deterministic update describes an Ordinary Differential Equation (ODE) known as the *Probability Flow ODE*. Visually, this creates a velocity field that transports particles from the noise distribution directly onto the data manifold along smooth trajectories (e.g., transforming a noise cloud into a spiral manifold).

6 Summary of Algorithms

Algorithm 1 DDPM Sampling (Stochastic)

```
Sample $x_1 \sim \mathcal{N}(0, \sigma_q^2 I)$
for $t = 1, 1 - \Delta t, \ldots, \Delta t$ do
    Sample noise $\eta \sim \mathcal{N}(0, \sigma_q^2 \Delta t I)$
    Predict mean using network: $\hat{x}_{prev} \leftarrow f_\theta(x_t, t)$
    Update with noise injection: $x_{t-\Delta t} \leftarrow \hat{x}_{prev} + \eta$
end for
return $x_0$
```

Algorithm 2 DDIM Sampling (Deterministic)

```
Sample $x_1 \sim \mathcal{N}(0, \sigma_q^2 I)$
for $t = 1, 1 - \Delta t, \ldots, \Delta t$ do
    Calculate step coefficient $\lambda \leftarrow \frac{\sqrt{t}}{\sqrt{t-\Delta t}+\sqrt{t}}$
    Update deterministically (no noise injection):
        $x_{t-\Delta t} \leftarrow x_t + \lambda(f_\theta(x_t, t) - x_t)$
end for
return $x_0$
```