Deep Generative Models

Lecture 14

Roman Isachenko

Moscow Institute of Physics and Technology

2022 – 2023

Recap of previous lecture

$$\mathbf{a}_{\mathbf{z}}(t) = \frac{\partial L(\mathbf{y})}{\partial \mathbf{z}(t)}; \quad \mathbf{a}_{\boldsymbol{\theta}}(t) = \frac{\partial L(\mathbf{y})}{\partial \boldsymbol{\theta}(t)} \text{– adjoint functions.}$$

Theorem (Pontryagin)

$$\frac{d\mathbf{a}_{\mathbf{z}}(t)}{dt} = -\mathbf{a}_{\mathbf{z}}(t)^T \cdot \frac{\partial f(\mathbf{z}(t), t, \boldsymbol{\theta})}{\partial \mathbf{z}}; \quad \frac{d\mathbf{a}_{\boldsymbol{\theta}}(t)}{dt} = -\mathbf{a}_{\mathbf{z}}(t)^T \cdot \frac{\partial f(\mathbf{z}(t), t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}.$$

Forward pass

$$\mathbf{z}(t_1) = \int_{t_0}^{t_1} f(\mathbf{z}(t), t, \boldsymbol{\theta}) dt + \mathbf{z}_0 \quad \Rightarrow \quad \text{ODE Solver}$$

Backward pass

$$\left.\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{\theta}(t_0)} &= \mathbf{a}_{\boldsymbol{\theta}}(t_0) = -\int_{t_1}^{t_0} \mathbf{a}_{\mathbf{z}}(t)^T \frac{\partial f(\mathbf{z}(t), t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}(t)} dt + 0 \\
\frac{\partial L}{\partial \mathbf{z}(t_0)} &= \mathbf{a}_{\mathbf{z}}(t_0) = -\int_{t_1}^{t_0} \mathbf{a}_{\mathbf{z}}(t)^T \frac{\partial f(\mathbf{z}(t), t, \boldsymbol{\theta})}{\partial \mathbf{z}(t)} dt + \frac{\partial L}{\partial \mathbf{z}(t_1)} \\
\mathbf{z}(t_0) &= -\int_{t_1}^{t_0} f(\mathbf{z}(t), t, \boldsymbol{\theta}) dt + \mathbf{z}_1.
\end{aligned}\right\} \Rightarrow \text{ODE Solver}$$

Chen R. T. Q. et al. Neural Ordinary Differential Equations, 2018

Recap of previous lecture

Continuous-in-time normalizing flows

$$\frac{d\mathbf{z}(t)}{dt} = f(\mathbf{z}(t), t, \boldsymbol{\theta}); \quad \frac{d \log p(\mathbf{z}(t), t)}{dt} = -\text{tr} \left(\frac{\partial f(\mathbf{z}(t), t, \boldsymbol{\theta})}{\partial \mathbf{z}(t)} \right).$$

Theorem (Picard)

If f is uniformly Lipschitz continuous in $\mathbf{z}$ and continuous in t, then the ODE has a **unique** solution.

Forward transform + log-density

$$\begin{bmatrix} \mathbf{x} \\ \log p(\mathbf{x}|\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ \log p(\mathbf{z}) \end{bmatrix} + \int_{t_0}^{t_1} \begin{bmatrix} f(\mathbf{z}(t), t, \boldsymbol{\theta}) \\ -\text{tr} \left(\frac{\partial f(\mathbf{z}(t), t, \boldsymbol{\theta})}{\partial \mathbf{z}(t)} \right) \end{bmatrix} dt.$$

Hutchinson's trace estimator

$$\log p(\mathbf{z}(t_1)) = \log p(\mathbf{z}(t_0)) - \mathbb{E}_{p(\boldsymbol{\epsilon})} \int_{t_0}^{t_1} \left[\boldsymbol{\epsilon}^T \frac{\partial f}{\partial \mathbf{z}} \boldsymbol{\epsilon} \right] dt.$$

Grathwohl W. et al. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, 2018

Recap of previous lecture

SDE basics

Let define stochastic process $\mathbf{x}(t)$ with initial condition $\mathbf{x}(0) \sim p_0(\mathbf{x})$:

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w},$$

where $\mathbf{w}(t)$ is the standard Wiener process (Brownian motion)

$$\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}(0, (t-s)\mathbf{I}), \quad d\mathbf{w} = \boldsymbol{\epsilon} \cdot \sqrt{dt}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

Langevin dynamics

Let $\mathbf{x}_0$ be a random vector. Then under mild regularity conditions for small enough η samples from the following dynamics

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta \frac{1}{2} \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}).$$

will comes from $p(\mathbf{x}|\boldsymbol{\theta})$.

The density $p(\mathbf{x}|\boldsymbol{\theta})$ is a **stationary** distribution for the Langevin SDE.

Welling M. Bayesian Learning via Stochastic Gradient Langevin Dynamics, 2011

Stochastic differential equation (SDE)

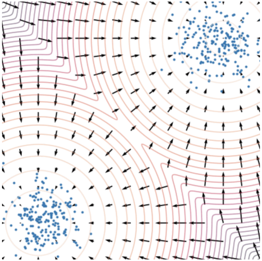
Statement

Let $\mathbf{x}_0$ be a random vector. Then samples from the following dynamics

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta \frac{1}{2} \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1).$$

will come from $p(\mathbf{x}|\boldsymbol{\theta})$ under mild regularity conditions for small enough η and large enough t.

The density $p(\mathbf{x}|\boldsymbol{\theta})$ is a **stationary** distribution for this SDE.



Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021

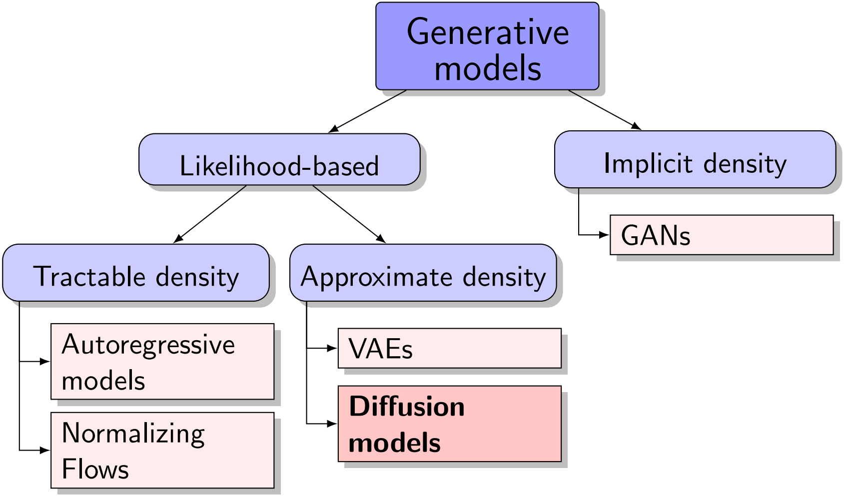
Outline

1. Score matching
2. Noise conditioned score network
3. Gaussian diffusion process

Outline

1. Score matching
2. Noise conditioned score network
3. Gaussian diffusion process

Generative models zoo



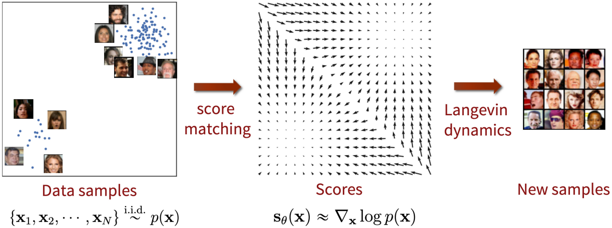
Score matching

We could sample from the model using Langevin dynamics if we have $\nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$.

Fisher divergence

$$D_F(\pi, p) = \frac{1}{2}\mathbb{E}_\pi \big\| \nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \big\|_2^2 \to \min_{\boldsymbol{\theta}}$$

Let introduce **score function** $\mathbf{s}(\mathbf{x}, \boldsymbol{\theta}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$.



Problem: we do not know $\nabla_{\mathbf{x}} \log \pi(\mathbf{x})$.

Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021

Score matching

Theorem (implicit score matching)

Under some regularity conditions, it holds

$$\frac{1}{2}\mathbb{E}_\pi\left\|\mathbf{s}(\mathbf{x},\boldsymbol{\theta})-\nabla_\mathbf{x}\log\pi(\mathbf{x})\right\|_2^2=\mathbb{E}_\pi\left[\frac{1}{2}\|\mathbf{s}(\mathbf{x},\boldsymbol{\theta})\|_2^2+\mathrm{tr}\big(\nabla_\mathbf{x}\mathbf{s}(\mathbf{x},\boldsymbol{\theta})\big)\right]+\text{const}$$

Proof (only for 1D)

$$\mathbb{E}_\pi\left\|s(x)-\nabla_x\log\pi(x)\right\|_2^2=\mathbb{E}_\pi\left[s(x)^2+(\nabla_x\log\pi(x))^2-2[s(x)\nabla_x\log\pi(x)]\right]$$

$$\begin{aligned}\mathbb{E}_\pi[s(x)\nabla_x\log\pi(x)] &= \int\pi(x)\nabla_x\log p(x)\nabla_x\log\pi(x)dx\\ &= \int\nabla_x\log p(x)\nabla_x\pi(x)dx=\pi(x)\nabla_x\log p(x)\Big|_{-\infty}^{+\infty}\\ &- \int\nabla_x^2\log p(x)\pi(x)dx=-\mathbb{E}_\pi\nabla_x^2\log p(x)=-\mathbb{E}_\pi\nabla_x s(x)\end{aligned}$$

$$\frac{1}{2}\mathbb{E}_\pi\left\|s(x)-\nabla_x\log\pi(x)\right\|_2^2=\mathbb{E}_\pi\left[\frac{1}{2}s(x)^2+\nabla_x s(x)\right]+\text{const.}$$

Hyvarinen A. Estimation of non-normalized statistical models by score matching, 2005

Score matching

Theorem (implicit score matching)

$$\frac{1}{2}\mathbb{E}_{\pi}\Big\|\mathbf{s}(\mathbf{x}, \boldsymbol{\theta}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x})\Big\|_2^2 = \mathbb{E}_{\pi}\Big[\frac{1}{2}\|\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})\|_2^2 + \text{tr}\big(\nabla_{\mathbf{x}}\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})\big)\Big] + \text{const}$$

Here $\nabla_{\mathbf{x}}\mathbf{s}(\mathbf{x}, \boldsymbol{\theta}) = \nabla^2_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$ is a Hessian matrix.

1. The left hand side is intractable due to unknown $\pi(\mathbf{x})$ – **denoising score matching**.
2. The right hand side is complex due to Hessian matrix – **sliced score matching**.

Sliced score matching (Hutchinson's trace estimation)

$$\text{tr}\big(\nabla_{\mathbf{x}}\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})\big) = \mathbb{E}_{p(\boldsymbol{\epsilon})}\Big[\boldsymbol{\epsilon}^T \nabla_{\mathbf{x}}\mathbf{s}(\mathbf{x}, \boldsymbol{\theta})\boldsymbol{\epsilon}\Big]$$

Song Y. Sliced Score Matching: A Scalable Approach to Density and Score Estimation, 2019

Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021

Denoising score matching

Let perturb original data $\mathbf{x} \sim \pi(\mathbf{x})$ by random normal noise

$$\mathbf{x}' = \mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1), \quad p(\mathbf{x}'|\mathbf{x}, \sigma) = \mathcal{N}(\mathbf{x}'|\mathbf{x}, \sigma^2 \mathbf{I})$$

$$\pi(\mathbf{x}'|\sigma) = \int \pi(\mathbf{x}) p(\mathbf{x}'|\mathbf{x}, \sigma) d\mathbf{x}.$$

Then the solution of

$$\frac{1}{2} \mathbb{E}_{\pi(\mathbf{x}'|\sigma)} \left\| \mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, \sigma) - \nabla_{\mathbf{x}'} \log \pi(\mathbf{x}'|\sigma) \right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

satisfies $\mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, \sigma) \approx \mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, 0) = \mathbf{s}(\mathbf{x}, \boldsymbol{\theta})$ if σ is small enough.

Vincent P. A Connection Between Score Matching and Denoising Autoencoders, 2010

Denoising score matching

Theorem

$$\mathbb{E}_{\pi(\mathbf{x}'|\sigma)} \left\| \mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, \sigma) - \nabla_{\mathbf{x}'} \log \pi(\mathbf{x}'|\sigma) \right\|_2^2 =$$
$$= \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{p(\mathbf{x}'|\mathbf{x}, \sigma)} \left\| \mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, \sigma) - \nabla_{\mathbf{x}'} \log p(\mathbf{x}'|\mathbf{x}, \sigma) \right\|_2^2 + \text{const}(\boldsymbol{\theta})$$

Gradient of the noise kernel

$$\nabla_{\mathbf{x}'} \log p(\mathbf{x}'|\mathbf{x}, \sigma) = \nabla_{\mathbf{x}'} \log \mathcal{N}(\mathbf{x}'|\mathbf{x}, \sigma^2 \mathbf{I}) = -\frac{\mathbf{x}' - \mathbf{x}}{\sigma^2}$$

- The RHS does not need to compute $\nabla_{\mathbf{x}'} \log \pi(\mathbf{x}'|\sigma)$ and even $\nabla_{\mathbf{x}'} \log \pi(\mathbf{x}')$.
- $\mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, \sigma)$ tries to **denoise** a corrupted sample $\mathbf{x}'$.
- Score function $\mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, \sigma)$ parametrized by σ. How to make it?

Vincent P. A Connection Between Score Matching and Denoising Autoencoders, 2010

Outline

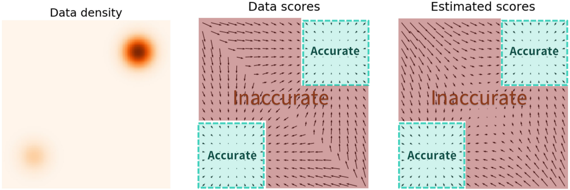
1. Score matching

2. Noise conditioned score network

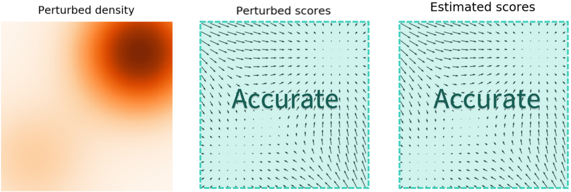
3. Gaussian diffusion process

Denoising score matching

- If σ is **small**, the score function is not accurate and Langevin dynamics will probably fail to jump between modes.



- If σ is **large**, it is good for low-density regions and multimodal distributions, but we will learn too corrupted distribution.



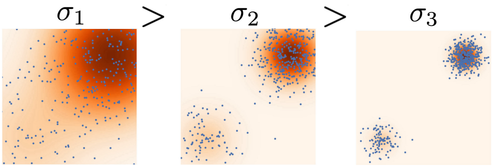
Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021

Noise conditioned score network

- Define the sequence of noise levels: $\sigma_1 > \sigma_2 > \dots > \sigma_L$.
- Perturb the original data with the different noise level to get $\pi(\mathbf{x}'|\sigma_1), \dots, \pi(\mathbf{x}'|\sigma_L)$.
- Train denoised score function $\mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, \sigma)$ for each noise level:

$$\sum_{l=1}^{L} \sigma_l^2 \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{p(\mathbf{x}'|\mathbf{x}, \sigma_l)} \left\| \mathbf{s}(\mathbf{x}', \boldsymbol{\theta}, \sigma_l) - \nabla'_{\mathbf{x}} \log p(\mathbf{x}'|\mathbf{x}, \sigma_l) \right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

- Sample from **annealed** Langevin dynamics (for $l = 1, \dots, L$).



Song Y. et al. Generative Modeling by Estimating Gradients of the Data Distribution, 2019

Noise conditioned score network

Training: loss function

$$\sum_{i=1}^{L} \sigma_l^2 \mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{\epsilon} \left\| \mathbf{s}_l + \frac{\epsilon}{\sigma_l} \right\|_2^2,$$

Here

- $\mathbf{s}_l = \mathbf{s}(\mathbf{x} + \sigma_l \cdot \epsilon, \boldsymbol{\theta}, \sigma_l)$.
- $\nabla_{\mathbf{x}'} \log p(\mathbf{x}'|\mathbf{x}, \sigma) = -\frac{\mathbf{x}' - \mathbf{x}}{\sigma^2} = -\frac{\epsilon}{\sigma_l}$.

Inference: annealed Langevin dynamic

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^L, \epsilon, T$.

```
Initialize x̃_0
for i ← 1 to L do
    α_i ← ε · σ_i^2 / σ_L^2        ▷ α_i is the step size.
    for t ← 1 to T do
        Draw z_t ~ N(0, I)
        x̃_t ← x̃_{t−1} + (α_i / 2) s_θ(x̃_{t−1}, σ_i) + √α_i z_t
    end for
    x̃_0 ← x̃_T
end for
return x̃_T
```

Samples

Song Y. et al. Improved Techniques for Training Score-Based Generative Models, 2020

Outline

1. Score matching

2. Noise conditioned score network

3. Gaussian diffusion process

Forward gaussian diffusion process

Let $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$, $\beta \in (0, 1)$. Define the Markov chain

$$\mathbf{x}_t = \sqrt{1-\beta} \cdot \mathbf{x}_{t-1} + \sqrt{\beta} \cdot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1);$$

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t | \sqrt{1-\beta} \cdot \mathbf{x}_{t-1}, \beta \cdot \mathbf{I}).$$

Statement 1

Applying the Markov chain to samples from any $\pi(\mathbf{x})$ we will get $\mathbf{x}_\infty \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$. Here $p_\infty(\mathbf{x})$ is a **stationary** distribution:

$$p_\infty(\mathbf{x}) = \int q(\mathbf{x}|\mathbf{x}') p_\infty(\mathbf{x}') d\mathbf{x}'.$$

Statement 2

Denote $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$. Then

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$$

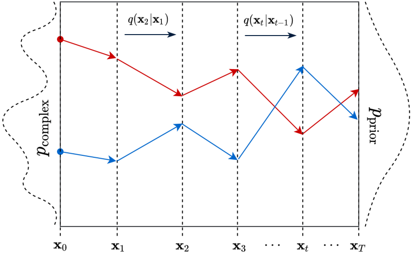
$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1 - \bar{\alpha}_t) \cdot \mathbf{I}).$$

We could sample from any timestamp using only $\mathbf{x}_0$!

Sohl-Dickstein J. Deep Unsupervised Learning using Nonequilibrium Thermodynamics, 2015

Forward gaussian diffusion process

Diffusion refers to the flow of particles from high-density regions towards low-density regions.

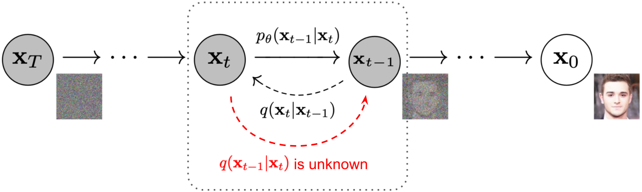


1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta} \cdot \mathbf{x}_{t-1} + \sqrt{\beta} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$, $t \geq 1$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$, where $T >> 1$.

If we are able to invert this process, we will get the way to sample $\mathbf{x} \sim \pi(\mathbf{x})$ using noise samples $p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$.
Now our goal is to revert this process.

Das A. An introduction to Diffusion Probabilistic Models, blog post, 2021

Reverse gaussian diffusion process



Let define the reverse process

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{x}_{t-1}|\boldsymbol{\mu}(\mathbf{x}_t, \boldsymbol{\theta}, t), \boldsymbol{\sigma}^2(\mathbf{x}_t, \boldsymbol{\theta}, t)\right)$$

Forward process

1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta} \cdot \mathbf{x}_{t-1} + \sqrt{\beta} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$, $t \geq 1$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$.

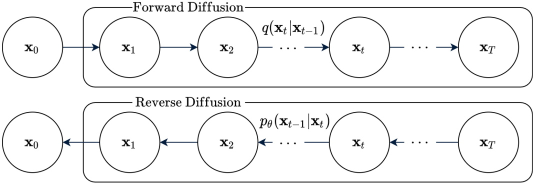
Reverse process

1. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$;
2. $\mathbf{x}_{t-1} = \boldsymbol{\sigma}(\mathbf{x}_t, \boldsymbol{\theta}, t) \cdot \boldsymbol{\epsilon} + \boldsymbol{\mu}(\mathbf{x}_t, \boldsymbol{\theta}, t)$;
3. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;

Note: The forward process does not have any learnable parameters!

Weng L. What are Diffusion Models?, blog post, 2021

Gaussian diffusion model as VAE



- Let treat $\mathbf{z} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ as a latent variable (**note:** each $\mathbf{x}_t$ has the same size).
- Variational posterior distribution (**note:** there is no learnable parameters)
$$q(\mathbf{z}|\mathbf{x}) = q(\mathbf{x}_1, \dots, \mathbf{x}_T|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}).$$
- Probabilistic model
$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta})$$
- Generative distribution and prior
$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}); \quad p(\mathbf{z}|\boldsymbol{\theta}) = \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot p(\mathbf{x}_T)$$

Das A. An introduction to Diffusion Probabilistic Models, blog post, 2021

Summary

- Score matching proposes to minimize Fisher divergence to get score function.
- Sliced score matching and denoising score matching are two techniques to get scalable algorithm for fitting Fisher divergence.
- Noise conditioned score network uses multiple noise levels and annealed Langevin dynamics to fit score function.
- Gaussian diffusion process is a Markov chain that injects special form of Gaussian noise to the samples.
- Reverse process allows to sample from the real distribution $\pi(\mathbf{x})$ using samples from noise.