Deep Generative Models

Lecture 11

Roman Isachenko

Moscow Institute of Physics and Technology

2023, Autumn

Recap of previous lecture

Consider Ordinary Differential Equation

$$\frac{d\mathbf{z}(t)}{dt} = f_{\boldsymbol{\theta}}(\mathbf{z}(t), t); \quad \text{with initial condition } \mathbf{z}(t_0) = \mathbf{z}_0.$$

$$\mathbf{z}(t_1) = \int_{t_0}^{t_1} f_{\boldsymbol{\theta}}(\mathbf{z}(t), t) dt + \mathbf{z}_0 = \text{ODESolve}(\mathbf{z}(t_0), f_{\boldsymbol{\theta}}, t_0, t_1).$$

Euler update step

$$\frac{\mathbf{z}(t + \Delta t) - \mathbf{z}(t)}{\Delta t} = f_{\boldsymbol{\theta}}(\mathbf{z}(t), t) \;\Rightarrow\; \mathbf{z}(t+\Delta t) = \mathbf{z}(t) + \Delta t \cdot f_{\boldsymbol{\theta}}(\mathbf{z}(t), t)$$

Residual block

$$\mathbf{z}_{t+1} = \mathbf{z}_t + f_{\boldsymbol{\theta}}(\mathbf{z}_t)$$

It is equivalent to Euler update step for solving ODE with $\Delta t = 1$!

In the limit of adding more layers and taking smaller steps we get:

$$\frac{d\mathbf{z}(t)}{dt} = f_{\boldsymbol{\theta}}(\mathbf{z}(t), t); \quad \mathbf{z}(t_0) = \mathbf{x}; \quad \mathbf{z}(t_1) = \mathbf{y}.$$

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

Recap of previous lecture

Forward pass (loss function)

$$L(\mathbf{y}) = L(\mathbf{z}(t_1)) = L\left(\mathbf{z}(t_0) + \int_{t_0}^{t_1} f_{\boldsymbol{\theta}}(\mathbf{z}(t), t) dt\right)$$
$$= L\big(\text{ODESolve}(\mathbf{z}(t_0), f_{\boldsymbol{\theta}}, t_0, t_1)\big)$$

Note: ODESolve could be any method (Euler step, Runge-Kutta methods).

Backward pass (gradients computation)

For fitting parameters we need gradients:

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

In theory of optimal control these functions called **adjoint** functions. They show how the gradient of the loss depends on the hidden state $\mathbf{z}(t)$ and parameters $\boldsymbol{\theta}$.

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

Recap of previous lecture

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

Theorem (Pontryagin)

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

Forward pass

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

Backward pass

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

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

Recap of previous lecture

Continuous-in-time normalizing flows

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

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

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

Hutchinson's trace estimator

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

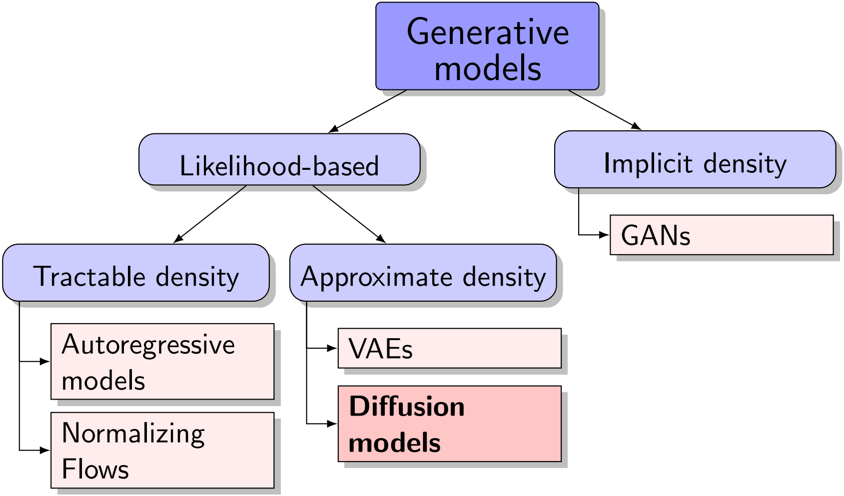
Grathwohl W. et al. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models, 2018

Outline

1. Gaussian diffusion process
   - Forward gaussian diffusion process
   - Reverse gaussian diffusion process

2. Gaussian diffusion model as VAE

Generative models zoo



Outline

Outline

Forward gaussian diffusion process

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

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

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

Statement 1

Let denote $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Then

$$\begin{aligned} \mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t = \\ &= \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-1}) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t = \\ &= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + (\sqrt{\alpha_t(1-\alpha_{t-1})}\boldsymbol{\epsilon}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t) = \\ &= \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \end{aligned}$$

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

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

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

At each step we

- scale magnitude of the signal at rate $\sqrt{1-\beta_t}$;
- add noise with variance β_t.

Statement 2

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** and **limiting** distribution:

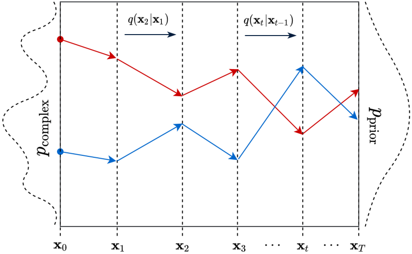
$$p_\infty(\mathbf{x}) = \int q(\mathbf{x}|\mathbf{x}')p_\infty(\mathbf{x}')d\mathbf{x}'.$$

$$p_\infty(\mathbf{x}) = \int q(\mathbf{x}_\infty|\mathbf{x}_0)\pi(\mathbf{x}_0)d\mathbf{x}_0 \approx \mathcal{N}(0, \mathbf{I})\int \pi(\mathbf{x}_0)d\mathbf{x}_0 = \mathcal{N}(0, \mathbf{I})$$

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.



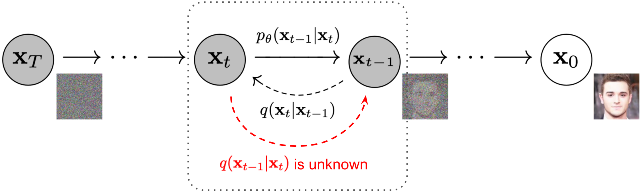
1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$, $t \geq 1$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$, where $T >>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

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

Outline

Reverse gaussian diffusion process



Forward process

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

Reverse process

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}) q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)} \approx p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})$$

- $q(\mathbf{x}_{t-1})$, $q(\mathbf{x}_t)$ are intractable.
- If β_t is small enough, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ will be Gaussian (Feller, 1949).

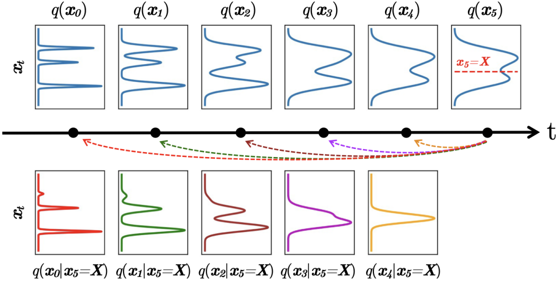
Feller W. On the theory of stochastic processes, with particular reference to applications, 1949

Reverse gaussian diffusion process

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)}$$

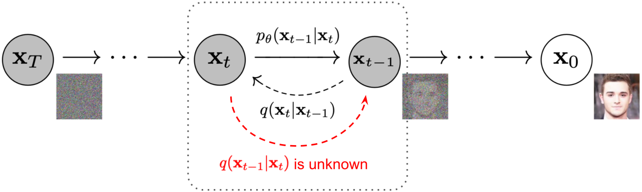
$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} = \mathcal{N}(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

- $q(\mathbf{x}_{t-1})$, $q(\mathbf{x}_t)$ are intractable.
- If β_t is small enough, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ will be Gaussian (Feller, 1949).



Xiao Z., Kreis K., Vahdat A. Tackling the generative learning trilemma with denoising diffusion GANs, 2021

Reverse gaussian diffusion process



Let define the reverse process

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

Forward process

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

Reverse process

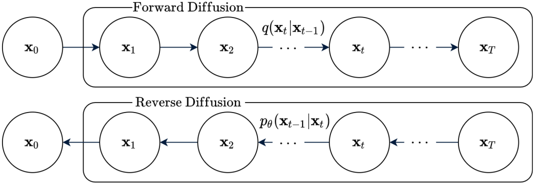
1. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$;
2. $\mathbf{x}_{t-1} = \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \cdot \boldsymbol{\epsilon} + \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$;
3. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;

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

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

Outline

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

ELBO for gaussian diffusion model

Standard ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x})} = \mathcal{L}(q, \boldsymbol{\theta}) \to \max_{q, \boldsymbol{\theta}}$$

Derivation

$$\begin{aligned}
\mathcal{L}(q, \boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_0, \mathbf{x}_{1:T}|\boldsymbol{\theta})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) \prod_{t=1}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \\
&= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)}
\end{aligned}$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} = \mathcal{N}(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

ELBO for gaussian diffusion model

Derivation (continued)

$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} =$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} =$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T) p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} =$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) + \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \sum_{t=2}^{T} \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) \right] =$$

$$= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) + \mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} +$$

$$+ \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{x}_0)} \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right)$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

ELBO for gaussian diffusion model

$$
\begin{aligned}
\mathcal{L}(q, \boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) + \mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \\
&+ \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{x}_0)} \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right) = \\
&= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) - KL\big(q(\mathbf{x}_T|\mathbf{x}_0) || p(\mathbf{x}_T)\big) - \\
&- \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \underbrace{KL\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) || p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})\big)}_{\mathcal{L}_t}
\end{aligned}
$$

- First term is a decoder distribution

$$
\log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) = \log \mathcal{N}\big(\mathbf{x}_0 | \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_1, t), \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\mathbf{x}_1, t)\big).
$$

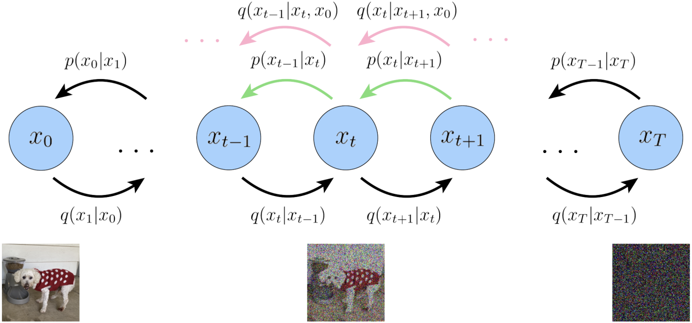
- Second term is constant ($p(\mathbf{x}_T)$ is a standard Normal, $q(\mathbf{x}_T|\mathbf{x}_0)$ is a non-parametrical Normal).

Ho J. Denoising Diffusion Probabilistic Models, 2020

ELBO for gaussian diffusion model

$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) - KL\big(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)\big) - \\ - \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \underbrace{KL\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})\big)}_{\mathcal{L}_t}$$

$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ defines how to denoise a noisy image $\mathbf{x}_t$ with access to what the final, completely denoised image $\mathbf{x}_0$ should be.



Luo C. Understanding Diffusion Models: A Unified Perspective, 2022

Summary

- 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.
- Diffusion model is a VAE model which reverts gaussian diffusion process using variational inference.
- ELBO of DDPM could be represented as a sum of KL terms.