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

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

Theorem (Pontryagin)

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

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

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

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

Recap of previous lecture

Continuous-in-time normalizing flows

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

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

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

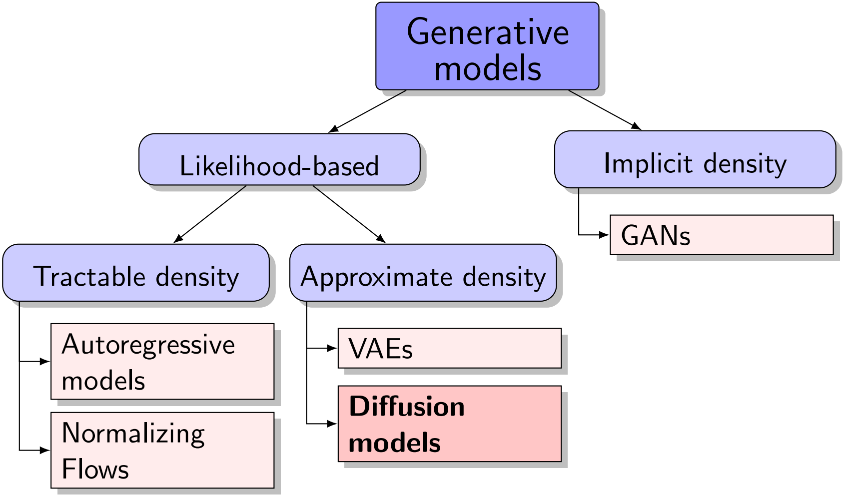
Hutchinson's trace estimator

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

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

Outline

Generative models zoo



Outline

1. Gaussian diffusion process

2. Denoising diffusion probabilistic model (DDPM)
   - Objective of DDPM
   - Reparametrization of DDPM

3. Overview of DDPM and link to score matching

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

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

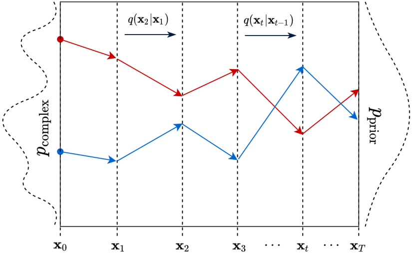
$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t | \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1 - \bar{\alpha}_t) \cdot \mathbf{I}).$$

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.

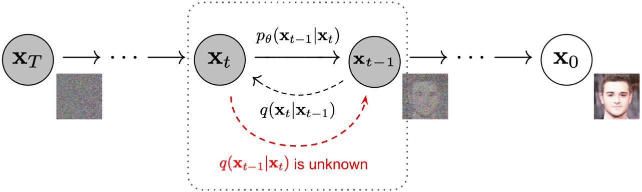


1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta} \cdot \mathbf{x}_{t-1} + \sqrt{\beta} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$, $t \geq 1$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$, where $T >> 1$.

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

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

Forward process

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

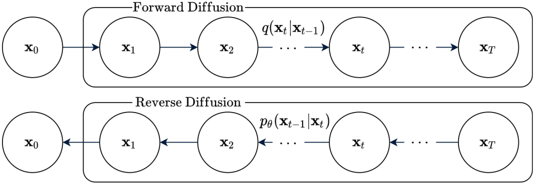
Reverse process

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

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

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta})$$

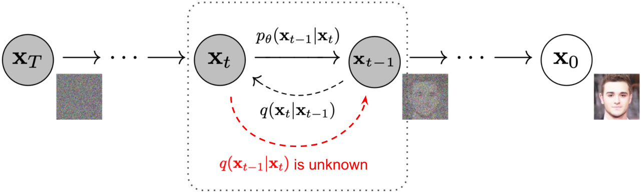
- Generative distribution and prior

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}); \quad p(\mathbf{z}|\boldsymbol{\theta}) = \prod_{t=2}^{T} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot p(\mathbf{x}_T)$$

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

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

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

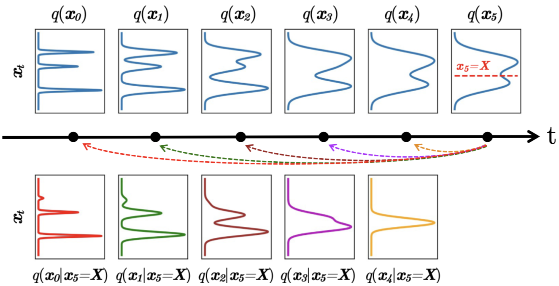
- $q(\mathbf{x}_{t-1})$, $q(\mathbf{x}_t)$ are intractable.
- If β_t is small enough, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ will be Gaussian (Feller, 1949).

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

Reverse gaussian diffusion process

Reverse process

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



Important distribution

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

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

Outline

1. Gaussian diffusion process

2. Denoising diffusion probabilistic model (DDPM)
   Objective of DDPM
   Reparametrization of DDPM

3. Overview of DDPM and link to score matching

Objective of DDPM

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

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

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

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}) q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

$$\log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right) = \log \left(\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) q(\mathbf{x}_t|\mathbf{x}_0)} \right)$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

Objective of DDPM

Derivation

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

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

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

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

$$\left. + \log p(\mathbf{x}_0|\mathbf{x}_1,\boldsymbol{\theta}) - \sum_{t=2}^{T} KL\big(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{\theta})\big)\right]$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

Objective of DDPM

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

- 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).
- $\mathcal{L}_t$ is a KL between two normal distributions:

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

$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)$ and $\tilde{\beta}_t$ have analytical formulas (we omit it) and they are both dependent on β_t.

Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

Gaussian diffusion model as VAE

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}),$$
$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{x}_{t-1}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right)$$

- Assume $\boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \tilde{\beta}_t \mathbf{I}$.
- Use KL formula between two normal distributions:

$$\begin{aligned}
\mathcal{L}_t &= KL\Big(\mathcal{N}\left(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}\right) || \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I}\right)\Big) \\
&= \mathbb{E}_{\boldsymbol{\epsilon}} \left[\frac{1}{2\tilde{\beta}_t} \left\| \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right\|^2 \right] \\
&= \mathbb{E}_{\boldsymbol{\epsilon}} \left[\frac{1}{2\tilde{\beta}_t} \left\| \frac{1}{\sqrt{1 - \beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right) - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right\|^2 \right]
\end{aligned}$$

Here we used the analytic expression for $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)$.

Reparametrization

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \frac{1}{\sqrt{1 - \beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right)$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

Reparametrization of DDPM

KL term

$$\mathcal{L}_t = \mathbb{E}_{\epsilon}\Bigg[\frac{1}{2\tilde{\beta}_t}\Bigg\|\frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) - \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_{\theta}(\mathbf{x}_t, t)\right)\Bigg\|^2\Bigg] =$$

$$\mathbb{E}_{\epsilon}\left[\frac{\beta_t^2}{2\tilde{\beta}_t(1-\beta_t)}\left\|\frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}} - \frac{\epsilon_{\theta}(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}\right\|^2\right]$$

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

At each step of reverse diffusion process we try to predict the noise ϵ that we used in forward process!

Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

Denoising diffusion probabilistic model (DDPM)

DDPM is a VAE model

- Encoder is a fixed Gaussian Markov chain.
- Latent variable is a hierarchical (in each step the dim. of the latent equals to the dim of the input).
- Decoder is a simple Gaussian model.
- Prior distribution is given by parametric Gaussian Makov chain.

Forward process

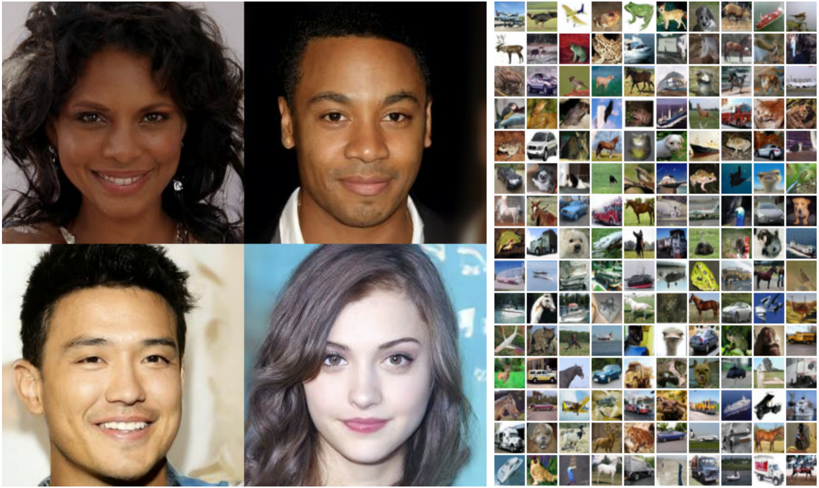
1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1 - \beta} \cdot \mathbf{x}_{t-1} + \sqrt{\beta} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$, $t \geq 1$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, 1)$.

Reverse process

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

Denoising diffusion probabilistic model (DDPM)

Samples



Ho J. Denoising Diffusion Probabilistic Models, 2020

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 is a sum of KL terms.
- At each step DDPM predicts the noise used in forward process.
- Objective of DDPM is closely related to the noise conditioned score network and score matching.