Deep Generative Models

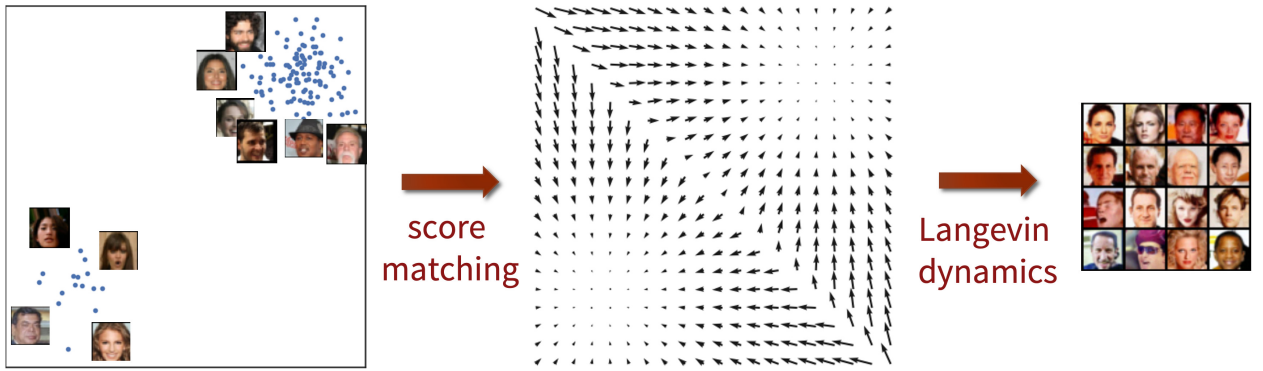
Lecture 13

Roman Isachenko

AI Masters

2024, Spring

Recap of previous lecture



Theorem (implicit score matching)

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

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 (Hutchinson's trace estimation)**.

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

Recap of previous lecture

Let perturb original data by normal noise $p(\mathbf{x}'|\mathbf{x}, \sigma) = \mathcal{N}(\mathbf{x}'|\mathbf{x}, \sigma^2\mathbf{I})$

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

Then the solution of

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

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

Theorem (denoising score matching)

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

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

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

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

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

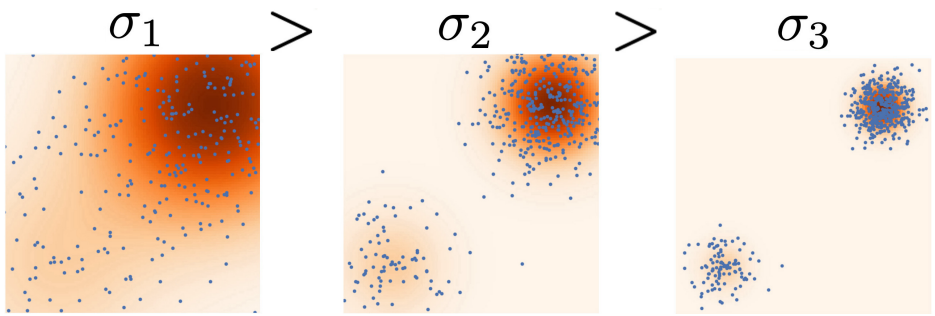
Recap of previous lecture

Noise conditioned score network

- Define the sequence of noise levels: $\sigma_1 > \sigma_2 > \cdots > \sigma_L$.
- Train denoised score function $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}', \sigma)$ for each noise level:

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

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



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

Recap of previous lecture

NCSN training

1. Get the sample $\mathbf{x}_0 \sim \pi(\mathbf{x})$.
2. Sample noise level $l \sim U[1, L]$ and the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.
3. Get noisy image $\mathbf{x}' = \mathbf{x}_0 + \sigma_l \cdot \boldsymbol{\epsilon}$.
4. Compute loss $\mathcal{L} = \|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}', \sigma_l) + \frac{\boldsymbol{\epsilon}}{\sigma_l}\|^2$.

NCSN sampling (annealed Langevin dynamics)

- Sample $\mathbf{x}_0 \sim \mathcal{N}(0, \sigma_1 \mathbf{I}) \approx \pi(\mathbf{x}|\sigma_1)$.
- Apply T steps of Langevin dynamic

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \frac{1}{2}\eta_l \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}, \sigma_l) + \sqrt{\eta_l}\boldsymbol{\epsilon}_t.$$

- Update $\mathbf{x}_0 := \mathbf{x}_T$ and choose the next σ_l.

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

Recap of previous lecture

NCSN objective

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

DDPM objective

$$\mathcal{L}_t = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})} \left[\frac{(1-\alpha_t)^2}{2\tilde{\beta}_t \alpha_t} \left\| \frac{\boldsymbol{\epsilon}}{\sqrt{1-\bar{\alpha}_t}} - \frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}} \right\|^2 \right]$$

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

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\cdot\mathbf{x}_0}{1-\bar{\alpha}_t} = -\frac{\boldsymbol{\epsilon}}{\sqrt{1-\bar{\alpha}_t}}.$$

Let reparametrize our model:

$$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}.$$

Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

1. Diffusion and Score matching SDEs

2. Guidance
   - Classifier guidance
   - Classifier-free guidance

3. The worst course overview

Outline

1. Diffusion and Score matching SDEs

2. Guidance
   - Classifier guidance
   - Classifier-free guidance

3. The worst course overview

Score matching SDE

Denoising score matching

$$\mathbf{x}_l = \mathbf{x} + \sigma_l \cdot \boldsymbol{\epsilon}_l, \quad p(\mathbf{x}_l|\mathbf{x}, \sigma_l) = \mathcal{N}(\mathbf{x}_l|\mathbf{x}, \sigma_l^2\mathbf{I})$$
$$\mathbf{x}_{l-1} = \mathbf{x} + \sigma_{l-1} \cdot \boldsymbol{\epsilon}_{l-1}, \quad p(\mathbf{x}_{l-1}|\mathbf{x}, \sigma_{l-1}) = \mathcal{N}(\mathbf{x}_{l-1}|\mathbf{x}, \sigma_{l-1}^2\mathbf{I})$$

$$\mathbf{x}_l = \mathbf{x}_{l-1} + \sqrt{\sigma_l^2 - \sigma_{l-1}^2} \cdot \boldsymbol{\epsilon}, \quad p(\mathbf{x}_l|\mathbf{x}_{l-1}, \sigma_l) = \mathcal{N}(\mathbf{x}_l|\mathbf{x}_{l-1}, (\sigma_l^2 - \sigma_{l-1}^2)\cdot\mathbf{I})$$

Let turn this Markov chain to the continuous stochastic process $\mathbf{x}(t)$ taking $L \to \infty$:

$$\mathbf{x}(t + dt) = \mathbf{x}(t) + \sqrt{\frac{\sigma^2(t + dt) - \sigma^2(t)}{dt}} dt \cdot \boldsymbol{\epsilon} = \mathbf{x}(t) + \sqrt{\frac{d[\sigma^2(t)]}{dt}} \cdot d\mathbf{w}$$

Variance Exploding SDE

$$d\mathbf{x} = \sqrt{\frac{d[\sigma^2(t)]}{dt}} \cdot d\mathbf{w}$$

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Diffusion SDE

Denoising Diffusion

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

Let turn this Markov chain to the continuous stochastic process taking $T \to \infty$ and taking $\beta(\frac{t}{T}) = \beta_t \cdot T$

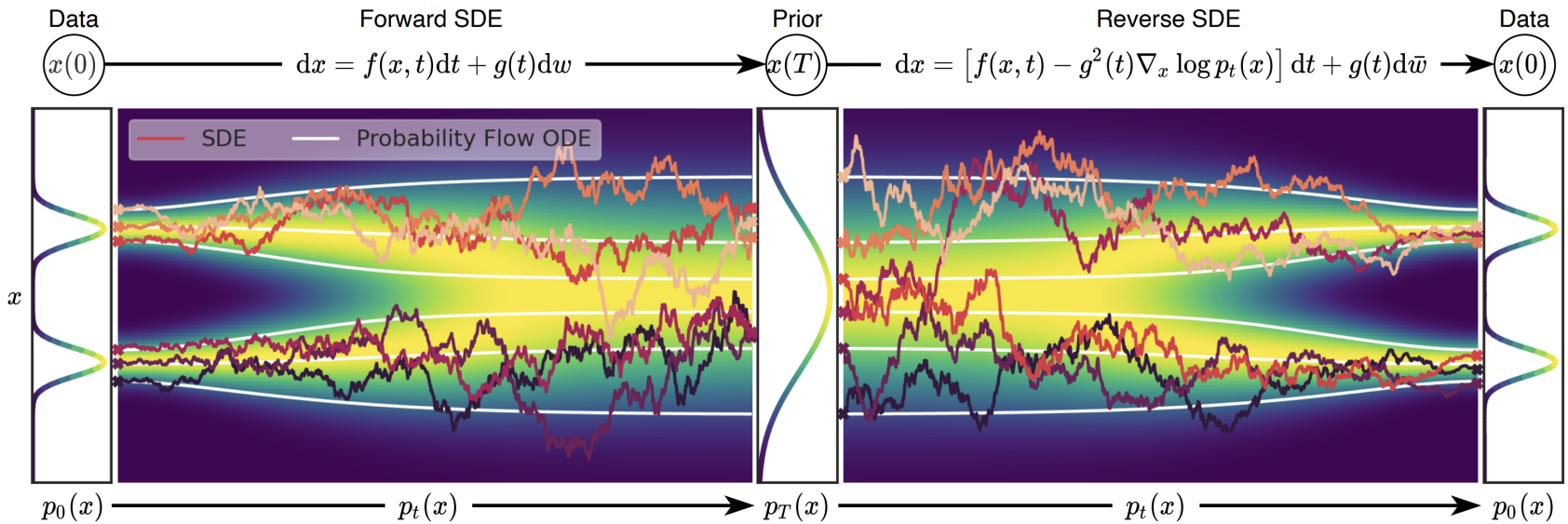
$$\begin{aligned}
\mathbf{x}(t) &= \sqrt{1-\beta(t)dt}\cdot\mathbf{x}(t-dt) + \sqrt{\beta(t)dt}\cdot\boldsymbol{\epsilon} \approx \\
&\approx (1 - \frac{1}{2}\beta(t)dt)\cdot\mathbf{x}(t-dt) + \sqrt{\beta(t)dt}\cdot\boldsymbol{\epsilon} = \\
&= \mathbf{x}(t-dt) - \frac{1}{2}\beta(t)\mathbf{x}(t-dt)dt + \sqrt{\beta(t)}\cdot d\mathbf{w}
\end{aligned}$$

Variance Preserving SDE

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}(t)dt + \sqrt{\beta(t)}\cdot d\mathbf{w}$$

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Diffusion SDE



Variance Exploding SDE (NCSN)

$$d\mathbf{x} = \sqrt{\frac{d[\sigma^2(t)]}{dt}} \cdot d\mathbf{w}$$

Variance Preserving SDE (DDPM)

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}(t)dt + \sqrt{\beta(t)} \cdot d\mathbf{w}$$

Song Y., et al. Score-Based Generative Modeling through Stochastic Differential Equations, 2020

Outline

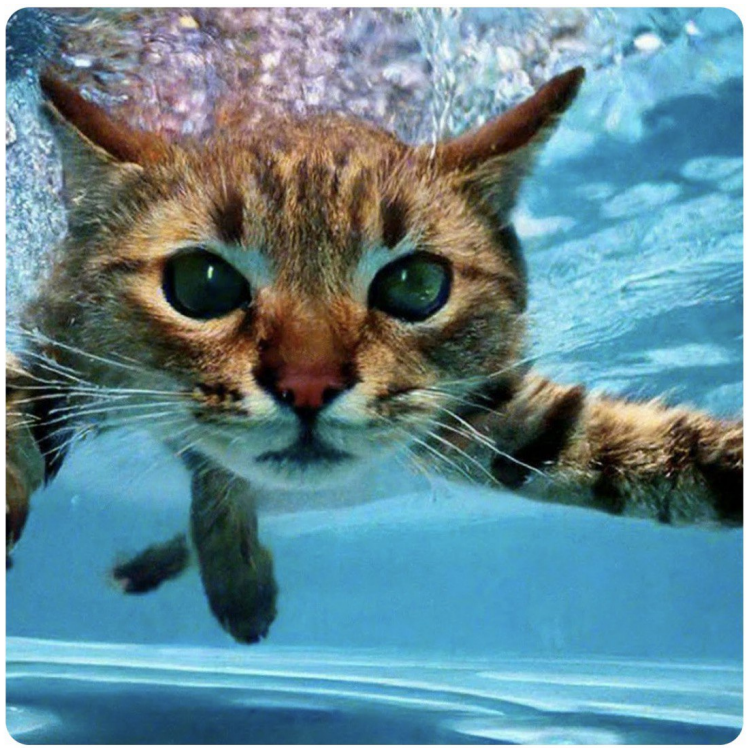
1. Diffusion and Score matching SDEs

2. Guidance
   - Classifier guidance
   - Classifier-free guidance

3. The worst course overview

Guidance

- Throughout the whole course we have discussed unconditional generative models $p(\mathbf{x}|\boldsymbol{\theta})$.
- In practice the majority of the generative models are **conditional**: $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$.
- Here $\mathbf{y}$ could be the class label or **text** (for text-to-image models).



Кот ныряет в бассейн, как ребенок на обложке альбома Nevermind, реалистично

рука человека с пятью пальцами, ни четыремя, ни шестью, а с 5 (пять) пальцами

Guidance

How to make conditional model $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$?

- If we have **supervised** data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$ we could theat $\mathbf{y}$ as additional model input:
  - $p(x_j|\mathbf{x}_{1:j-1}, \mathbf{y}, \boldsymbol{\theta})$ for AR;
  - Encoder $q(\mathbf{z}|\mathbf{x}, \mathbf{y}, \boldsymbol{\phi})$ and decoder $p(\mathbf{x}|\mathbf{z}, \mathbf{y}, \boldsymbol{\theta})$ for VAE;
  - $G_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{y})$ for NF and GAN;
  - $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta})$ for DDPM.
- If we have **unsupervised** data $\{\mathbf{x}_i\}_{i=1}^m$ we need to create the way to convert unconditional model $p(\mathbf{x}|\boldsymbol{\theta})$ to the conditional.

DDPM **unsupervised** guidance

- Let imagine we are given the distribution $q(\mathbf{y}|\mathbf{x}_0)$.
- Since we have already defined Markov chain, we have $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}) = q(\mathbf{x}_t|\mathbf{x}_{t-1})$.
- Let try to find reverse $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y})$.

Label guidance

Label: Ostrich (10th ImageNet class)

VQ-VAE (Proposed) **BigGAN deep**

Razavi A., Oord A., et al. Generating Diverse High-Fidelity Images with VQ-VAE-2, 2019

Text guidance

Prompt: a stained glass window of a panda eating bamboo
Left: $\gamma = 1$, Right: $\gamma = 3$.

Nichol A., et al. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models, 2022

Outline

Classifier guidance

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

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

Conditional distribution

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

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

$$= q(\mathbf{y}|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1}|\mathbf{x}_t) \cdot \text{const}(\mathbf{x}_{t-1}).$$

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot \text{const}(\mathbf{x}_{t-1}).$$

- $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})$ - our unsupervised diffusion model.
- $p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})$ - classifier for noised samples $\mathbf{x}_{t-1}$

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

Conditional distribution

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta}, \phi) = p(\mathbf{y}|\mathbf{x}_{t-1}, \phi) \cdot p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot \text{const}(\mathbf{x}_{t-1})$$

$$\log p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta}, \phi) = \log p(\mathbf{y}|\mathbf{x}_{t-1}, \phi) + \log p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) + \text{const}$$

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

$$\log p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = -\frac{\|\mathbf{x}_{t-1} - \boldsymbol{\mu}\|^2}{2\boldsymbol{\sigma}^2} + \text{const}(\mathbf{x}_{t-1})$$

Taylor expansion

$$\log p(\mathbf{y}|\mathbf{x}_{t-1}, \phi) \approx \log p(\mathbf{y}|\mathbf{x}_{t-1}, \phi)|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}} +$$
$$+ (\mathbf{x}_{t-1} - \boldsymbol{\mu}) \cdot \nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{y}|\mathbf{x}_{t-1}, \phi)|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}} =$$
$$= (\mathbf{x}_{t-1} - \boldsymbol{\mu}) \cdot \mathbf{g} + \text{const}(\mathbf{x}_{t-1}),$$

where $\mathbf{g} = \nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{y}|\mathbf{x}_{t-1}, \phi)|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}}$.

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

$$\log p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = (\mathbf{x}_{t-1} - \boldsymbol{\mu}) \cdot \mathbf{g} - \frac{\|\mathbf{x}_{t-1} - \boldsymbol{\mu}\|^2}{2\boldsymbol{\sigma}^2} + \text{const}(\mathbf{x}_{t-1})$$
$$= -\frac{\|\mathbf{x}_{t-1} - \boldsymbol{\mu} - \boldsymbol{\sigma} \odot \mathbf{g}\|^2}{2\boldsymbol{\sigma}^2} + \text{const}(\mathbf{x}_{t-1})$$
$$= \log \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \mathbf{g}, \boldsymbol{\sigma}^2) + \text{const}(\mathbf{x}_{t-1})$$

Guided sampling

1. Sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
2. Compute mean of $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I})$:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t).$$

3. Compute $\mathbf{g} = \nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}}$.
4. Get denoised image $\mathbf{x}_{t-1} = (\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) + \sqrt{\tilde{\beta}_t} \cdot \mathbf{g}) + \sqrt{\tilde{\beta}_t} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

Theorem (denoising score matching)

$$\mathbb{E}_{q(\mathbf{x}_t)} \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) \right\|_2^2 =$$

$$= \mathbb{E}_{\pi(\mathbf{x}_0)} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0) \right\|_2^2 + \text{const}(\boldsymbol{\theta})$$

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) \approx -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}} = \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t).$$

Conditional distribution

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \nabla_{\mathbf{x}_t} \log \left(\frac{p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\phi}) p(\mathbf{x}_t|\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\phi})} \right)$$

$$= \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\phi}) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta})$$

$$= \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\phi}) - \frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}$$

Classifier-corrected noise prediction

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \sqrt{1-\bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\phi})$$

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

Classifier-corrected noise prediction

$$\hat{\epsilon}_\theta(\mathbf{x}_t, t) = \epsilon_\theta(\mathbf{x}_t, t) - \sqrt{1-\bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \phi)$$

$$\hat{\epsilon}_\theta(\mathbf{x}_t, t) = \epsilon_\theta(\mathbf{x}_t, t) - \gamma \cdot \sqrt{1-\bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \phi)$$

Here we introduce guidance scale γ that controls the magnitude of the classifier guidance.

Conditional distribution

$$\frac{\hat{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}} = \frac{\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}} - \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \phi)$$

$$\nabla^\gamma_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \theta, \phi) = \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \phi) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta)$$

$$p(\mathbf{x}_t|\mathbf{y}, \theta, \phi) = \frac{p(\mathbf{y}|\mathbf{x}_t, \phi)^\gamma p(\mathbf{x}_t|\theta)}{Z}$$

Check that $\nabla_{\mathbf{x}_t} \log Z \neq 0$.

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

Guided sampling

1. Sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
2. Compute "corrected" $\hat{\epsilon}_{\theta}(\mathbf{x}_t, t)$:

$$\hat{\epsilon}_{\theta}(\mathbf{x}_t, t) = \epsilon_{\theta}(\mathbf{x}_t, t) - \gamma \cdot \sqrt{1 - \bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \phi)$$

3. Compute mean of $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta) = \mathcal{N}(\mu_{\theta}(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I})$:

$$\mu_{\theta}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \hat{\epsilon}_{\theta}(\mathbf{x}_t, t)$$

4. Get denoised image $\mathbf{x}_{t-1} = \mu_{\theta}(\mathbf{x}_t, t) + \sqrt{\tilde{\beta}_t} \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Outline

Classifier-free guidance

Previous methods require training the additional classifier model $p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\theta})$ on the noisy data. Let try to avoid this requirement.

$$\nabla^{\gamma}_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\phi}) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) =$$

$$= \gamma \cdot \nabla_{\mathbf{x}_t} \log \left(\frac{p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) p(\mathbf{y}|\boldsymbol{\phi})}{p(\mathbf{x}_t|\boldsymbol{\theta})} \right) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) =$$

$$= \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) - \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) =$$

$$= \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) + (1 - \gamma) \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta})$$

What will we get if $\gamma = 1$?

Classifier-free-corrected noise prediction

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \gamma \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y}, t) + (1 - \gamma) \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$$

In practice we could train the single model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y}, t)$ on **supervised** data alternating with real conditioning $\mathbf{y}$ and empty conditioning $\mathbf{y} = \emptyset$.

Ho J., Salimans T. Classifier-Free Diffusion Guidance, 2022

Outline

1. Diffusion and Score matching SDEs

2. Guidance
   - Classifier guidance
   - Classifier-free guidance

3. The worst course overview

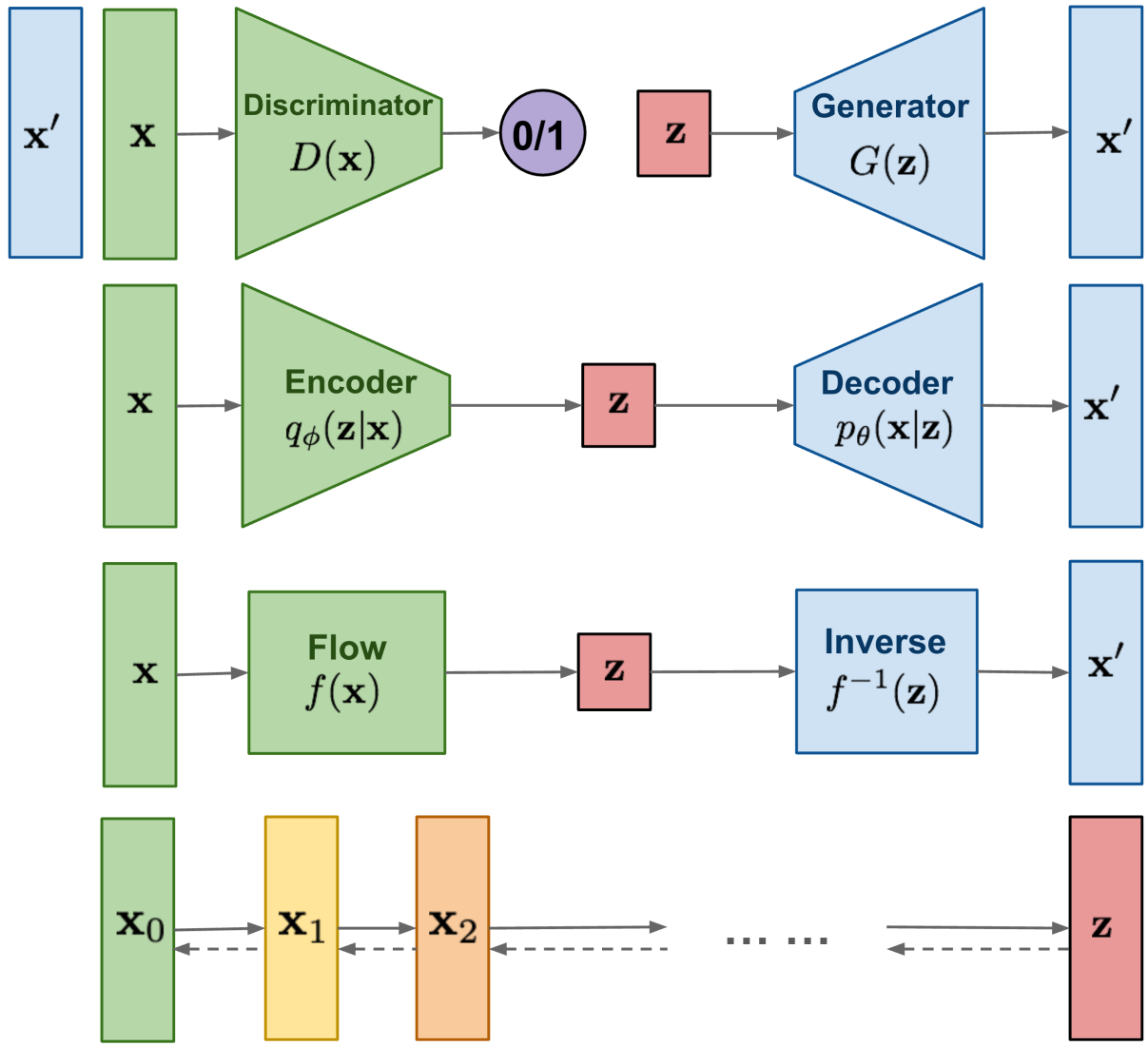
The worst course overview :)

GAN: Adversarial training

VAE: maximize variational lower bound

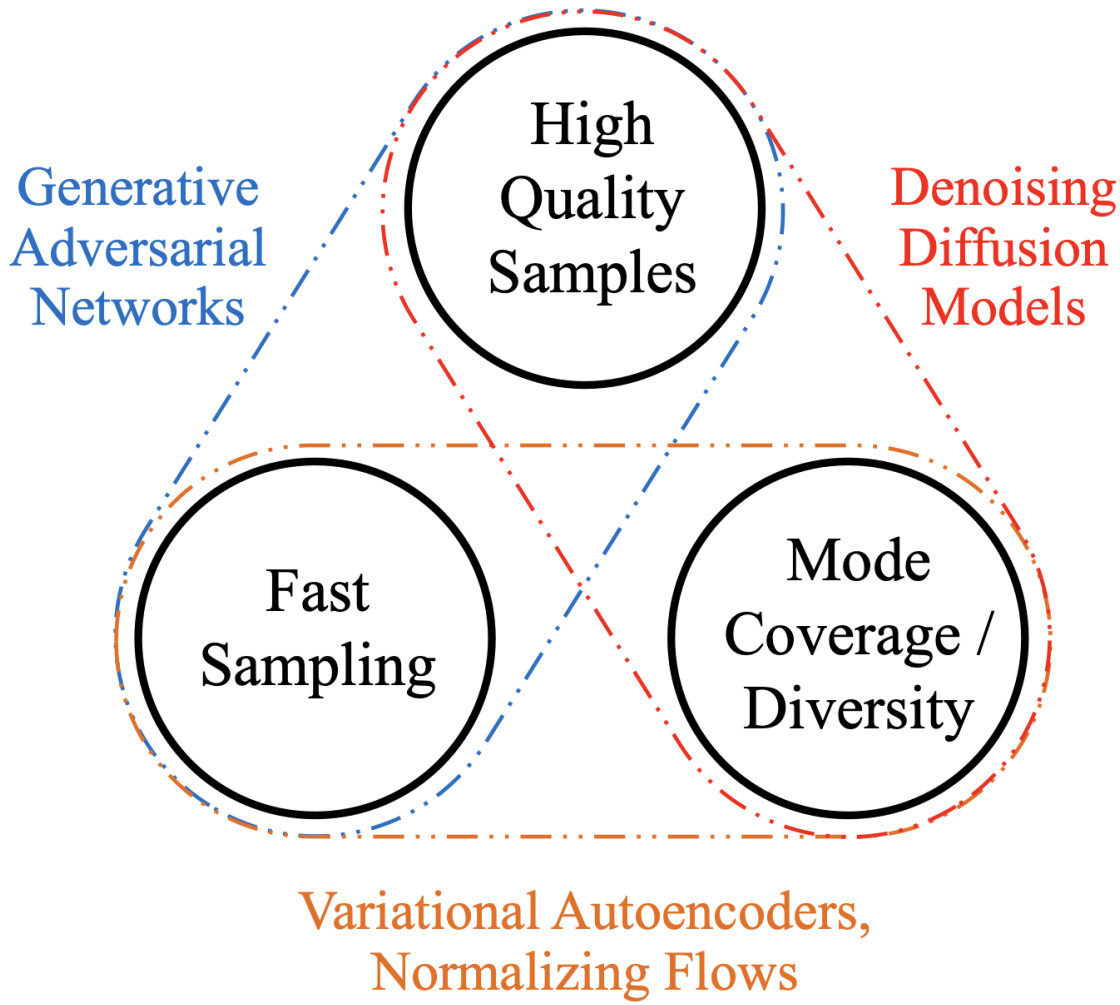
Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse



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

The worst course overview :)



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

Summary

- Score matching (NCSN) and diffusion models (DDPM) are the discretizations of the SDEs (variance exploding and variance preserving).
- Conditional models use labels $\mathbf{y}$ as the additional input. Majority of the modern generative models are conditional.
- Classifier guidance is the way to turn the unconditional model to the conditional one via the training additional classifier on the noisy data.
- Classifier-free guidance allows to avoid the training additional classifier to get the conditional model. It is widely used in practice.