Introduction to Machine and Deep Learning Theory

Diffusion models

Aleksandr Petiushko

Lomonosov MSU
Faculty of Mechanics and Mathematics

March 22, 2023

Content

AP

Introduction[1]

- Let's remember Variational Auto Encoder (VAE)
- The concept of VAE is based on the definition of "latent" information z, which we don't observe
- What we observe is actually the samples from the joint distribution $p(x, z)$
- In order to get the marginal probability $p(x)$ we need to either:
    - Integrate out all possible latents: $p(x) = \int p(x, z)dz$, or
    - Use the chain rule of probability: $p(x) = \frac{p(x,z)}{p(z|x)}$

[1] Hereafter we'd refer for any details to C. Luo. "Understanding diffusion models: A unified perspective", 2022

AP

Introduction (2)

- $p(x) = \int p(x,z)dz$ vs $p(x) = \frac{p(x,z)}{p(z|x)}$?
- Neither of these approaches are tractable: we cannot practically integrate over the whole latent space and we don't have an access to a ground truth latent encoder $p(z|x)$
- The approach to mitigate the problem: let's use the known (and learned) family of posterior distribution $q_\phi(z|x)$ and maximize $p(x)$ through maximization of so-called Evidence Lower Bound (ELBO): $\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right]$
- Using the definition of ELBO, we can establish[2] the following relation:

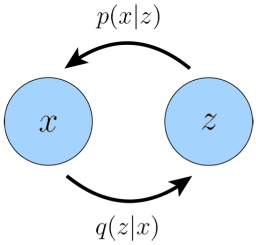
$$\log p(x) \geq \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right]$$

[2] **Exercise**: Prove it

AP

VAE

- For VAE, we can rewrite[3] the ELBO as the sum of "reconstruction" and "prior matching" terms:
  $\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right] = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$
- The following design choices are commonly used:
  $q_\phi(z|x) = N(\mu_\phi(x), \sigma_\phi^2 I)$
  $p(z) = N(0, I)$



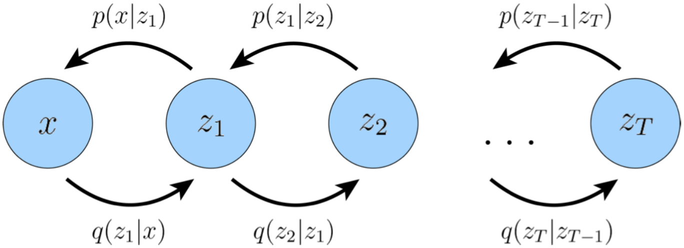
- Let's introduce hierarchy to latents!
- Even more, let's work for Markovian case, where decoding each latent z_t is dependent only on the previous latent z_{t+1}.

[3] **Exercise**: Prove it

MHVAE

- In case of Markovian Hierarchical VAE we can rewrite[4] the ELBO:

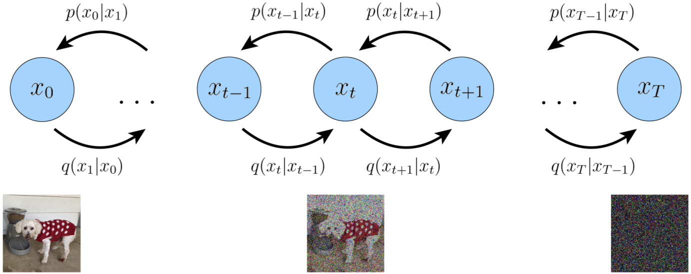
$$\mathbb{E}_{q_\phi(z_{1:T}|x)}\left[\log \frac{p(x, z_{1:T})}{q_\phi(z_{1:T}|x)}\right] = \mathbb{E}_{q_\phi(z_{1:T}|x)}\left[\log \frac{p(z_T)p_\theta(x|z_1)\prod_{t=2}^{T} p_\theta(z_{t-1}|z_t)}{q_\phi(z_1|x)\prod_{t=2}^{T} q_\phi(z_t|z_{t-1})}\right]$$



[4] **Exercise**: Prove it

Variational Diffusion Models (VDM)[5]

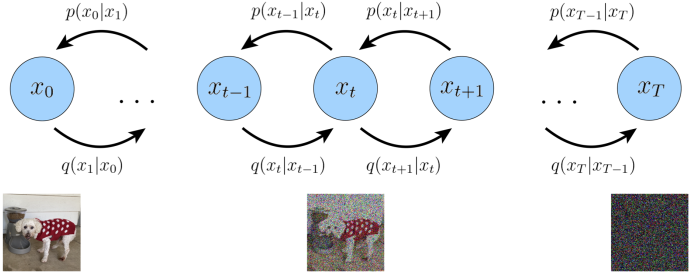
- Variational Diffusion Models (VDM) are MHVAE with the following properties:
  - **Dimension** of latents z is exactly the **same** as dimension of observed data x
  - Latent encoder is a **linear Gaussian** process (it means a Gaussian distribution centered around the output from the previous timestamp t)
  - Parameters of encoders vary so as the distribution of the final latent (at timestamp T) is a **standard** Gaussian $N(0, I)$



[5] D. Kingma et al. "Variational diffusion models", 2021

Diffusion process: notations

- Why **Diffusion**? Aside from SDE/PDE parallel, we can think of adding step by step some portion of noise as a diffusion analogy
- **Forward** diffusion process: adding noise by $q(x_t|x_{t-1})$. Also known as *encoding*
- **Reverse** diffusion process: de-noising by $p(x_{t-1}|x_t)$. Also known as *decoding*



VDM: important remarks

- Let's use the notation for **both** the latents **and** data as x_t: $x_0 = x, x_{t,t>0} = z_t$
  - For simplification – because the dimensionality of z and x is the same
  - Then the posterior distribution: $q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$
- Let's the "atomic" encoder (recall, a linear Gaussian) has a distribution $q(x_t|x_{t-1}) = N(a_t x_{t-1}, \sigma_t^2 I)$
  - We would like to choose coefficients so as the **variance** of latents is **preserved**
  - In this case the variance of $x_t = a_t x_{t-1} + \sigma_t \epsilon$, where $\epsilon \sim N(0, I)$, is: $Var(x_t) = a_t^2 + \sigma_t^2$
  - If we additionally assume re-normalization so as the preserved variance should be equal to I, then $\sigma_t = \sqrt{1 - a_t^2}$
  - Finally, if we define $a_t = \sqrt{\alpha_t}$ (for later calculations simplification), then $\sigma_t = \sqrt{1 - \alpha_t}$ and

$$q(x_t|x_{t-1}) = N(\sqrt{\alpha_t} x_{t-1}, (1 - \alpha_t) I)$$

- The joint distribution for a decoder is $p(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$, where the prior $p(x_T) = N(0, I)$
- Note, that here we only learn decoder params θ

VDM: analytical $q(x_t|x_0)$

- We have the variance preserving $q(x_t|x_{t-1}) = N(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)$
- What about $q(x_t|x_0)$?
  - Let us roll-out the equations based on linearity of Gaussians, where any $\epsilon_t \sim N(0, I)$:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} = \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-2}) + \sqrt{1-\alpha_t}\epsilon_{t-1} =$$

$$= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1} =$$

$$= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon_{t-2} = \dots = \sqrt{\prod_{i=1}^{t}\alpha_i}x_0 + \sqrt{1-\prod_{i=1}^{t}\alpha_i}\epsilon_0$$

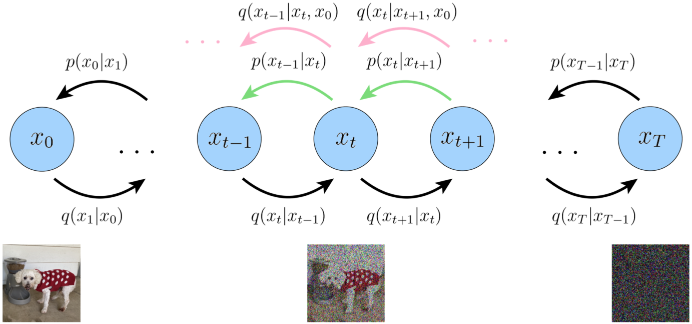
  - So, if we define $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$, then $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_0$
- As a result, we get $q(x_t|x_0) = N(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$

Exercise: Prove the transition between lines 2 and 3 of equations above.

AP

VDM: on a ground-truth denoising step

- The (learned) denoising step is described by $p_\theta(x_{t-1}|x_t)$
- How we could approximate it with some ground-truth distribution?
- The $q(x_{t-1}|x_t)$ is hard to get in the analytical form, but after **adding conditioning** on x_0 – $q(x_{t-1}|x_t, x_0)$ – it is possible!



AP

VDM: analytical $q(x_{t-1}|x_t, x_0)$

- Using chain rule and Markovian assumption, we get the following:

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_{t-1}, x_t|x_0)}{q(x_t|x_0)} = \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)} = \frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

- Note, that we know distributions of $q(x_t|x_{t-1})$, $q(x_t|x_0)$ and $q(x_{t-1}|x_0)$
- Let's substitute:

$$q(x_{t-1}|x_t, x_0) = \frac{N(\sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I)N(\sqrt{\bar{\alpha}_{t-1}}x_0, (1-\bar{\alpha}_{t-1})I)}{N(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)} \propto$$

$$\propto \exp\left\{-\left[\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{2(1-\alpha_t)} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{2(1-\bar{\alpha}_{t-1})} - \frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{2(1-\bar{\alpha}_t)}\right]\right\}$$

AP

VDM: analytical $q(x_{t-1}|x_t, x_0)$ (cont.)

- Let's simplify[6]:

$$q(x_{t-1}|x_t, x_0) \propto \exp\left\{-\frac{\left(x_{t-1} - \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}\right)^2}{2\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}\right\} \propto$$

$$\propto N\left(\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}, \Sigma_q(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}I\right)$$

- Let's denote the scalar multiplying factor $\sigma_q^2(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \Rightarrow \Sigma_q(t) = \sigma_q^2(t)I$

[6] **Exercise**: Prove this simplification.

VDM: ELBO (1)

- Now we are fully equipped to calculate the ELBO (and, correspondingly, the loss function)
- Here we also will use Bayes rule and Markovian property for $q(x_t|x_{t-1}, x_0)$

$$\log p(x_0) \geq \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)p_\theta(x_0|x_1)\prod_{t=2}^{T} p_\theta(x_{t-1}|x_t)}{q(x_1|x_0)\prod_{t=2}^{T} q(x_t|x_{t-1})}\right] =$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)p_\theta(x_0|x_1)\prod_{t=2}^{T} p_\theta(x_{t-1}|x_t)}{q(x_1|x_0)\prod_{t=2}^{T} q(x_t|x_{t-1}, x_0)}\right] =$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)p_\theta(x_0|x_1)}{q(x_1|x_0)} + \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1}, x_0)}\right]$$

AP

VDM: ELBO (2)

$$\log p(x_0) \geq \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)p_\theta(x_0|x_1)}{q(x_1|x_0)} + \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{\frac{q(x_{t-1}|x_t,x_0)q(x_t|x_0)}{q(x_{t-1}|x_0)}}\right] =$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)p_\theta(x_0|x_1)}{q(x_1|x_0)} + \log \frac{q(x_1|x_0)}{q(x_T|x_0)} + \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\right] =$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)p_\theta(x_0|x_1)}{q(x_T|x_0)} + \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\right] =$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log p_\theta(x_0|x_1)\right] + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)}{q(x_T|x_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\right]$$

AP

VDM: ELBO (3)

After removing (line 1) unnecessary random variables and re-writing (line 3) the expectation[7]:

$$\log p(x_0) \geq$$

$$\geq \mathbb{E}_{q(x_1|x_0)}\left[\log p_\theta(x_0|x_1)\right] + \mathbb{E}_{q(x_T|x_0)}\left[\log \frac{p(x_T)}{q(x_T|x_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(x_t,x_{t-1}|x_0)}\left[\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\right] =$$

$$= \mathbb{E}_{q(x_1|x_0)}\left[\log p_\theta(x_0|x_1)\right] - D_{KL}(q(x_T|x_0)||p(x_T)) -$$

$$- \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\left[D_{KL}(q(x_{t-1}|x_t,x_0)||p_\theta(x_{t-1}|x_t))\right]$$

[7] **Exercise**: Formally prove it

VDM: Loss terms

- *Reconstruction* term: $\mathbb{E}_{q(x_1|x_0)}\left[\log p_\theta(x_0|x_1)\right]$. Estimate through Monte Carlo
- *Prior matching* term: $D_{KL}(q(x_T|x_0)||p(x_T))$. We can omit it hereafter as it has no trainable params
- *Denoising matching* term: $\mathbb{E}_{q(x_t|x_0)}\left[D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))\right]$. Major loss term, where we learn the transition step $p_\theta(x_{t-1}|x_t)$ as an approximation to a tractable, ground truth denoising transition step $q(x_{t-1}|x_t, x_0)$ – but without the access to the initial data example x_0!

Note: When $T = 1$, the denoising matching term is absent, and the reconstruction and prior matching terms are exactly the same as for VAE ($x_0 = x, x_T = z$).

VDM: Denoising matching term

- Consider the KL-divergence $D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$
- In order to match learned denoising transition step $p_\theta(x_{t-1}|x_t)$ to ground truth denoising transition step $q(x_{t-1}|x_t, x_0)$ as closely as possible, we also **model** p as **Gaussian**: $p_\theta(x_{t-1}|x_t) \sim N(\mu_\theta, \Sigma_\theta)$
- KL-divergence between two Gaussians for data of dimension d[8]:

$$D_{KL}(N(\mu_q, \Sigma_q)||N(\mu_\theta, \Sigma_\theta)) = \frac{1}{2}\left[\log \frac{|\Sigma_\theta|}{|\Sigma_q|} - d + tr(\Sigma_\theta^{-1}\Sigma_q) + (\mu_\theta - \mu_q)\Sigma_\theta^{-1}(\mu_\theta - \mu_q)\right]$$

- Let's set the **variance** of p exactly the **same** as of q: $\Sigma_\theta = \Sigma_q$

[8] **Exercise**: Prove it

VDM: Denoising matching term (cont.)

- Then $D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) =$
$\frac{1}{2}\left[\log\frac{|\Sigma_q|}{|\Sigma_q|} - d + tr(\Sigma_q^{-1}\Sigma_q) + (\mu_\theta - \mu_q)\Sigma_q^{-1}(\mu_\theta - \mu_q)\right] = \frac{1}{2\sigma_q^2(t)}\left\|\mu_\theta - \mu_q\right\|_2^2$
- The proposal is for $\mu_\theta(x_t, t)$ to match $\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}$ by using the following re-parameterization:

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

- where $\hat{x}_\theta(x_t, t)$ is our trained neural network that is trying to predict x_0 from the noisy x_t (and having the information about the time t)

VDM: Denoising matching term – interpretation I ($\hat{x}$)

- Substituting expressions for $\mu_q(x_t, x_0)$ and $\mu_\theta(x_t, t)$, we get the following[9]:
  $D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) = \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \|\hat{x}_\theta(x_t, t) - x_0\|_2^2$
- Maximizing ELBO $\Rightarrow$ minimizing the Denoising matching term over all timestamps:

$$\arg\min_\theta \mathbb{E}_{t\sim U[2,T]} \left[\mathbb{E}_{q(x_t|x_0)} \left[\frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \|\hat{x}_\theta(x_t, t) - x_0\|_2^2 \right] \right]$$

- It is known as interpretation I[10]

[9] **Exercise**: Prove it

[10] J. Ho, A. Jain, and P. Abbeel. "Denoising diffusion probabilistic models", 2020

VDM: Denoising matching term – interpretation II ($\hat{\epsilon}$)

- Taking into account $q(x_t|x_0) = N(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$ we can express $x_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_0}{\sqrt{\bar{\alpha}_t}}$, $\epsilon_0 \sim N(0, I)$ and then substitute into $\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t} = \frac{1}{\sqrt{\alpha_t}}x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\epsilon_0$[11]
- Then we can re-parameterize our approximate denoising transition mean in order to predict the noise $\hat{\epsilon}_\theta(x_t, t)$ by the appropriate neural net:

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

- Corresponding denoising matching term becomes: $D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) = \frac{1}{2\sigma_q^2(t)} \|\mu_\theta - \mu_q\|_2^2 = \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \|\epsilon_0 - \hat{\epsilon}_\theta(x_t, t)\|_2^2$

[11] **Exercise**: Prove it

VDM: Denoising matching term – interpretation III (s_θ)

- Based on Tweede's Formula[12], we get $x_0 = \frac{x_t + (1-\bar{\alpha}_t)\nabla \log p(x_t)}{\sqrt{\bar{\alpha}_t}}$, and then substitute into $\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t} = \frac{1}{\sqrt{\alpha_t}}x_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla \log p(x_t)$[13]
- Then we can re-parameterize our approximate denoising transition mean in order to predict the score function $s_\theta(x_t, t)$ by the appropriate neural net:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}x_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}s_\theta(x_t, t)$$

- Corresponding denoising matching term becomes: $D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) = \frac{1}{2\sigma_q^2(t)}\left\|\mu_\theta - \mu_q\right\|_2^2 = \frac{1}{2\sigma_q^2(t)}\frac{(1-\alpha_t)^2}{\alpha_t}\left\|s_\theta(x_t, t) - \nabla \log p(x_t)\right\|_2^2$
- Note, that the score function is the scaled, opposite direction than the noise: $\nabla \log p(x_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_0$

[12] Efron, Bradley. "Tweedie's formula and selection bias", 2011

[13] **Exercise**: Prove it

AP

Some design choices

- In the seminal work of DDPM[14] the authors used the following parameters:
  - Task: image generation
  - $T = 1000$
  - $\beta_t = 1 - \alpha_t$, and β_t is increasing linearly from $\beta_1 = 10^{-4}$ to $\beta_T = 2 \cdot 10^{-2}$ so as *Prior matching* term $D_{KL}(q(x_T|x_0)||p(x_T)) \approx 0$
  - Decoder $p_\theta(x_{t-1}|x_t, t)$ is the same U-Net[15] based on a Wide ResNet[16] for all timestamps with time fed as a sinusoidal position encoding ("embedding")

[14] J. Ho, A. Jain, and P. Abbeel. "Denoising diffusion probabilistic models", 2020

[15] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation", 2015.

[16] S. Zagoruyko and N. Komodakis. "Wide residual networks", 2016

Enhancements of VDM

- Instead of predefined noise parameters α_t/β_t, they can be learned[17]
- The process of noise addition $q(x_t|x_{t-1}, x_0)$ can be non-Markovian (DDIM)[18]
- To explicitly learn[19] the variance of reversed diffusion process (variance of decoder) Σ_θ
- Improve the sampling procedure either by optimal sampling sub-trajectory (by dynamic programming)[20] or Progressive Distillation[21]
- A significant amount of works representing the Score-Based Generative models and Langevin dynamics

[17] D. Kingma et al. "Variational diffusion models", 2021

[18] J. Song, C. Meng, and S. Ermon. "Denoising diffusion implicit models", 2020

[19] F. Bao et al. "Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models", 2022

[20] D. Watson et al. "Learning fast samplers for diffusion models by differentiating through sample quality", 2021

[21] T. Salimans and J. Ho. "Progressive distillation for fast sampling of diffusion models", 2022

Conditioning

- How to have more targeted generation? E.g. text conditioning in text2image generation, or a low resolution image to make high-resolution in the process of super-resolution?
- It is done via conditioning on a signal y: $p(x) \to p(x|y)$
- For our decoders it means $p(x_{0:T}|y) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, y)$
- A simple approach to add a new input y is not very effective as decoder can sometimes (or even often!) just ignore this input[22]
- The solution is to use the so-called mechanism of “guidance” in order to control the generation process more explicitly (and at the same time at the cost of diversity)

[22] P. Dhariwal and A. Nichol. "Diffusion models beat gans on image synthesis", 2021)

AP

Classifier guidance[23]

- Let's refer to the Interpretation III (learning score function s_θ to approximate $\nabla \log p(x)$) with the help of Bayes rule:
$\nabla \log p(x_t|y) = \nabla \log \frac{p(y|x_t)p(x_t)}{p(y)} = \nabla \log p(x_t) + \nabla \log p(y|x_t) - \nabla_{x_t} \log p(y) =$
$\nabla \log p(x_t) + \nabla \log p(y|x_t)$
- So we can decompose into learning a standalone unconditional model in parallel to some classifier taking noisy image and trying to predict the label y: only during the inference we combine them
- For the fine-grained control of conditioning, we can introduce the scalar coefficient $\gamma \in [0,1]$: $\nabla \log p(x_t|y) = \nabla \log p(x_t) + \gamma \nabla \log p(y|x_t)$
- Starting from $\gamma = 0$ (unconditional generation) we can increase γ to use conditioning more explicitly
- Obvious drawback: 3rd-party classifier $p(y|x_t)$ should tackle all the noisy levels $\Rightarrow$ usually we need to train this classifier synchronously with the diffusion model as well

[23] Y. Song et al. "Score-based generative modeling through stochastic differential equations", 2020

Classifier-free guidance[24]

- Idea: to use conditioning – but without the standalone 3rd-party classifier
- Let's substitute $\nabla \log p(y|x_t) = \nabla \log p(x_t) - \nabla \log p(x_t|y)$ into $\nabla \log p(x_t|y) = \nabla \log p(x_t) + \gamma \nabla \log p(y|x_t)$:

$$\nabla \log p(x_t|y) = \gamma \nabla \log p(x_t|y) + (1-\gamma) \nabla \log p(x_t)$$

- Using values $\gamma > 1$ we move in the direction away from unconditional score function
- Approach: to learn a single (instead of two different) conditional model where the "unconditional" behavior is modeled through the fixed conditioning input (such as zeros): random dropout

[24] J. Ho and T. Salimans. "Classifier-free diffusion guidance", 2022

Takeaway notes

- 2 of 3 diffusion model interpretations: Markovian Hierarchical VAE
- Latents are not interpretable (in comparison to VAE)
- The analogy of the diffusion process in our brain is still under question
- We have NLL (in comparison to GAN)
- Obvious drawbacks:
  - expensive sampling (although there are multiple approaches to mitigate this issue),
  - latents are of the same dimension as input data (Stable/Latent diffusion still has the same issue – they just use as an input already low-dimensional latents from other models)

Thank you!