

Лекция 12

Variational Autoencoders

Denoising diffusion processes

Храбров Кузьма

26 ноября 2022 г.



План лекции

Autoencoders

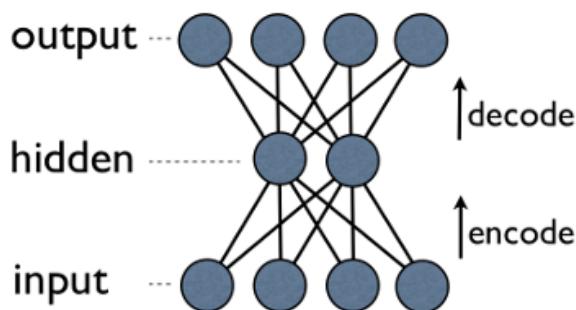
VAE

DDPM

Автокодировщики

Структура

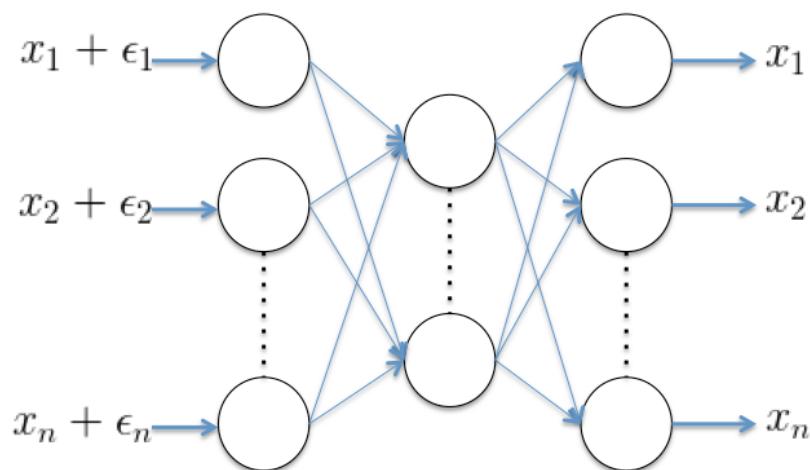
- ▶ Рассматривается сеть, обучаемая на отображении $f(x) = x$
- ▶ Внутри сети есть bottleneck слой, активации которого — представление объектов в низкоразмерном пространстве
- ▶ В сверточных сетях: pooling/stride и deconvolution / unpooling



Применение

- ▶ Выделение признаков для других алгоритмов
- ▶ Снижение размерности
- ▶ Предобучение на неразмеченных данных

Denoising autoencoder



Примеры шума

- ▶ Нормальный шум: $\mathcal{N}(\mu, \sigma^2 I)$
- ▶ Маскирующий шум: часть элементов обнуляется
- ▶ Соль и перец: часть элементов принимают максимальное/минимальное допустимое значение

Denoising autoencoder

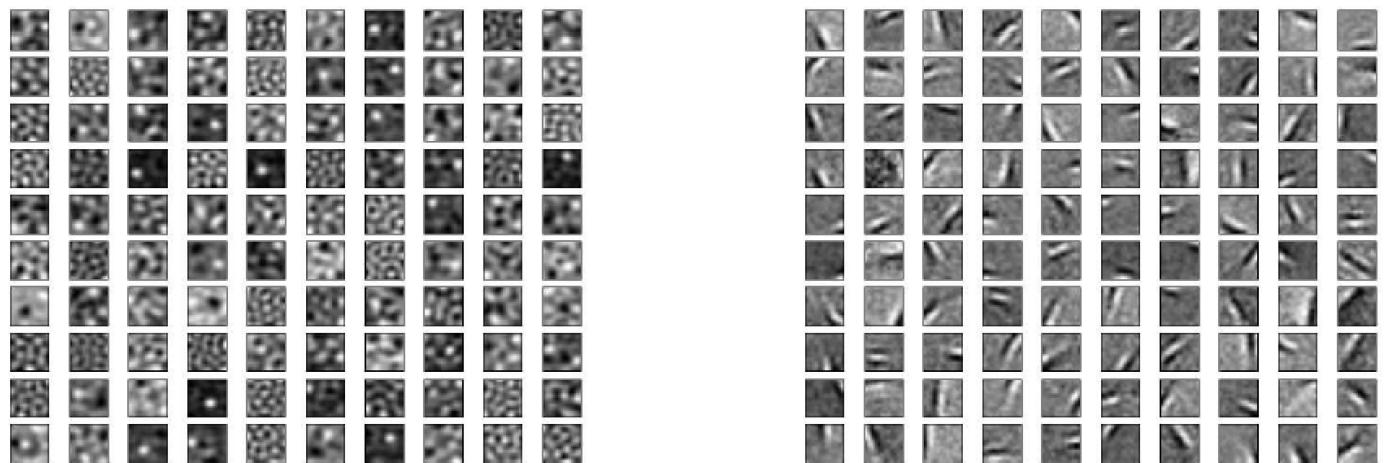
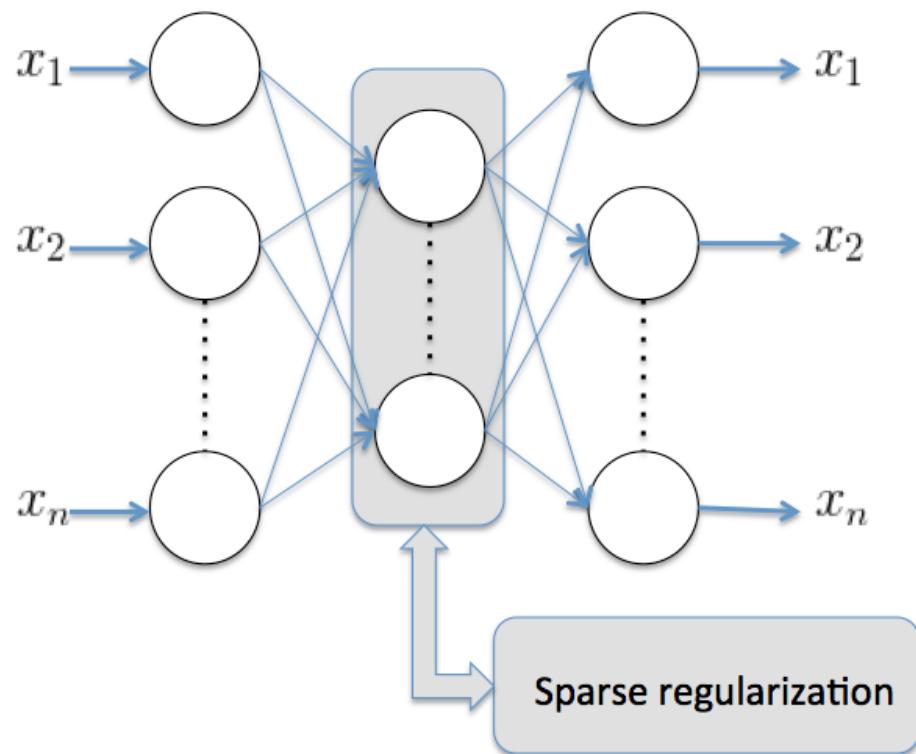


Рис.: Слева автоэнкодер, справа автоэнкодер с гауссовым шумом¹

¹Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, 2010, P. Vincent, Y. Bengio, and others

Разреженный автокодировщик



Идея: можем использовать больший скрытый слой, если введем регуляризацию

Sparse autoencoder

Регуляризатор разреженности

- ▶ Хотим, чтобы каждый нейрон в среднем активировался в ρ случаях ($\rho = 0.05$)
- ▶ Пусть средняя активация нейрона $\hat{\rho}$
- ▶ Регуляризатор: $KL(\rho \parallel \hat{\rho}) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}$

KL дивергенция

$$KL(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

- ▶ $KL(p \parallel q) \geq 0$
- ▶ $KL(p \parallel q) = 0 \Leftrightarrow p(x) = q(x)$ п.в

Sparse autoencoder²

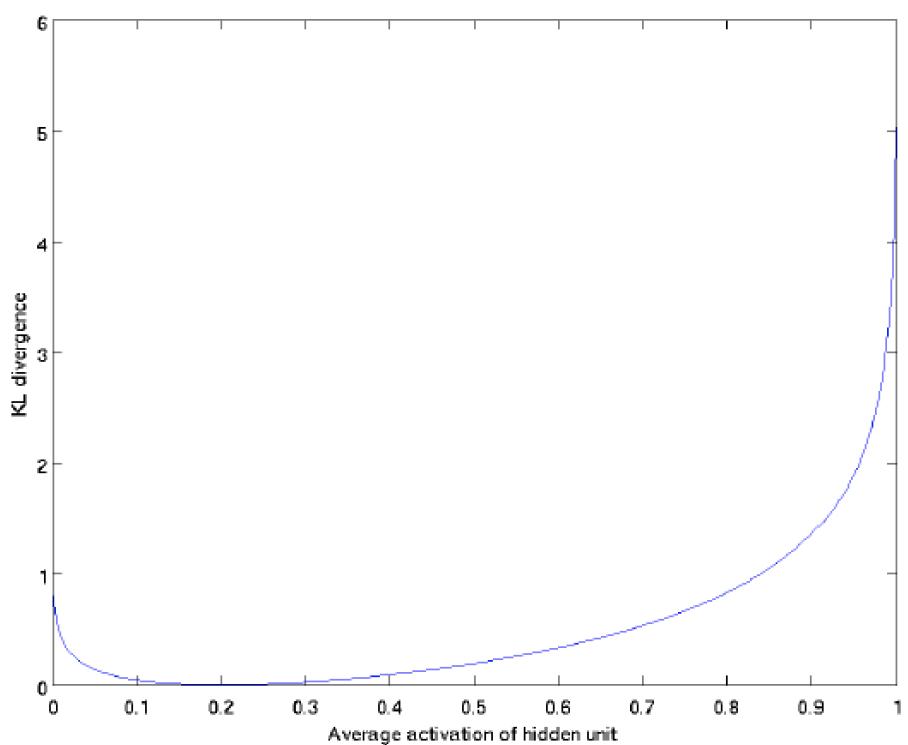


Рис.: KL достигает минимального значение в точке $\hat{\rho}_j = \rho$

²Sparse autoencoder, CS294A Lecture notes, Andrew Ng

Generative adversarial networks

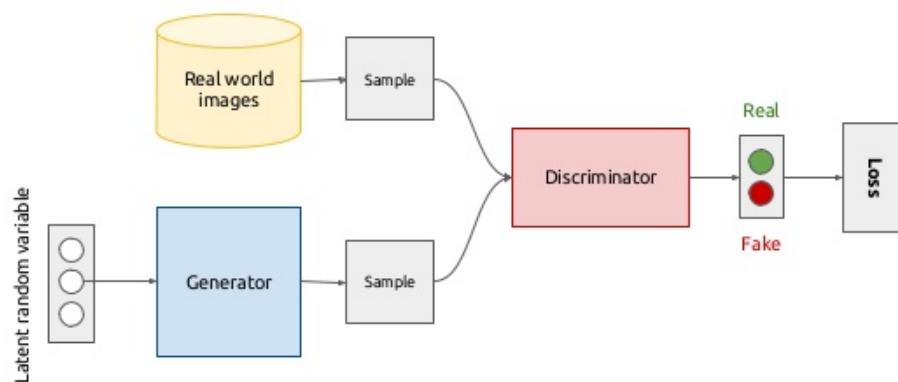
$z \sim p_z(z)$ - noise vector

$p_g(z)$ - распределение сгенерированных картинок из noise

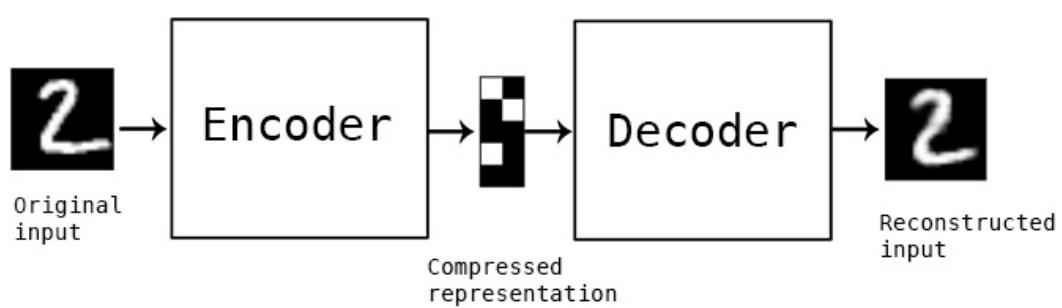
$p_{data}(x)$ - распределение настоящих картинок

$G(z)$ - генератор (генерирует картинку из z)

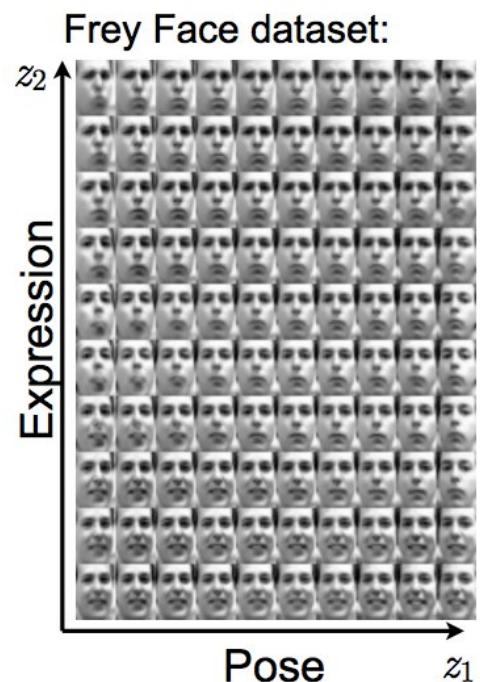
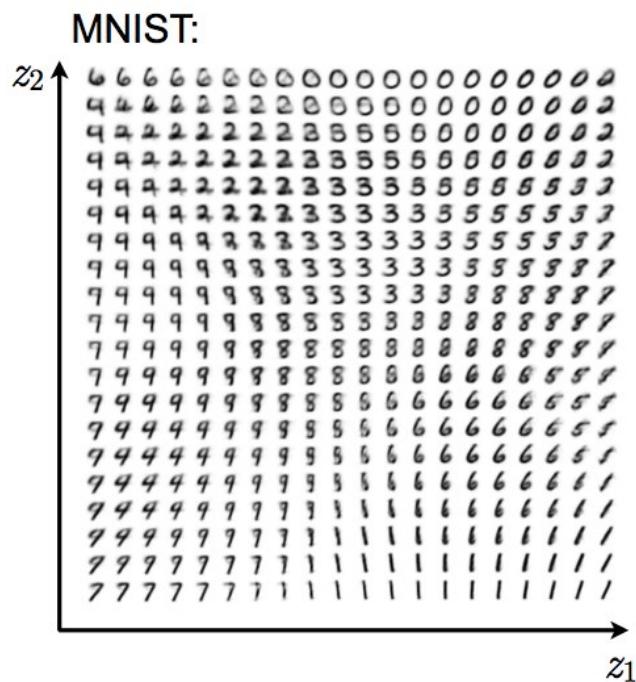
$D(x)$ - дискриминатор (отличает реальные от сгенерированных)



Autoencoder



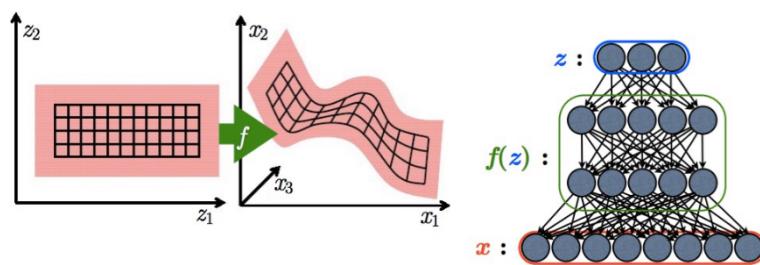
Autoencoder



Latent variable model

Найдем отображение из пространства латентных переменных (z) в распределение исходных данных (x).

$$p(x) = \int p(x, z) dz$$
$$p(x, z) = p(x|z)p(z)$$



Latent variable model

Как искать параметры сетей приближающих плотность неизвестных распределений? Методом максимума правдоподобия!

$$\theta^* = \operatorname{argmax}_\theta \log p(X)$$

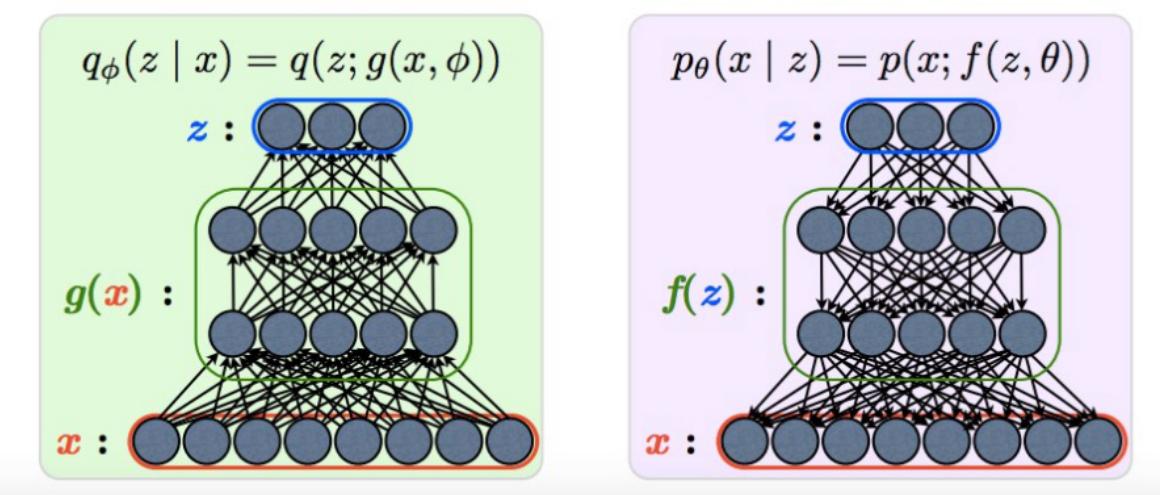
$$\begin{aligned} \log p(X) &= \sum_i \log p(x_i|\theta) = \sum_i \int_z \log p(x_i|\theta) q(z) dz \\ &= \sum_i \int_z \log \left(\frac{p(x_i, z|\theta) q(z)}{p(z|x_i, \theta) q(z)} \right) q(z) dz \geq \sum_i \int_z q(z, \phi) \log \left(\frac{p(x_i, z|\theta)}{q(z|\phi)} \right) dz \\ &= \mathcal{L}(\theta, \phi, x) \end{aligned}$$

VAE

- ▶ Откуда взять z и $p(z|x)$?
- ▶ Подход VAE: будем аппроксимировать $p_\theta(z|x)$ с помощью $q_\phi(z|x)$, оптимизируя вариационную нижнюю границу $\mathcal{L}(\theta, \phi, x)$

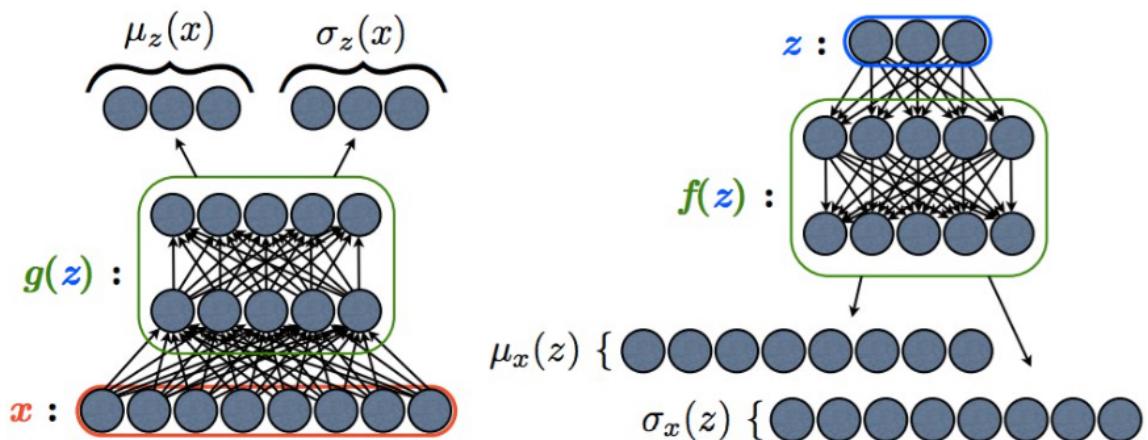
$$\begin{aligned}\mathcal{L}(\theta, \phi, x) &= \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x, z) - \log q_\phi(z | x)] \\ &= \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x | z) + \log p_\theta(z) - \log q_\phi(z | x)] \\ &= -D_{\text{KL}}(q_\phi(z | x) \| p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x | z)]\end{aligned}$$

VAE

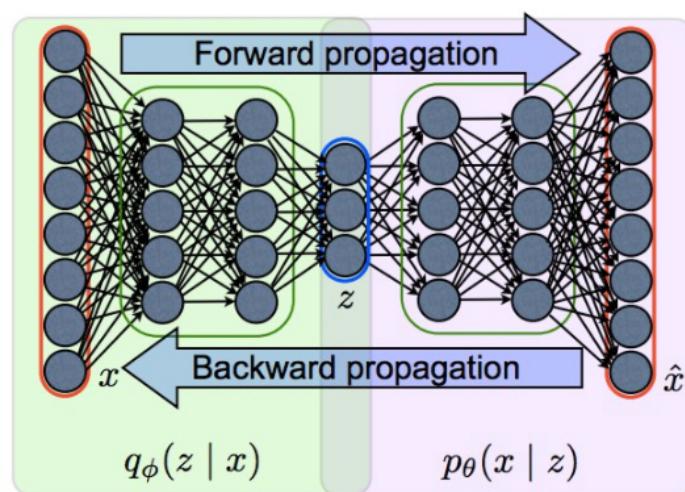


VAE

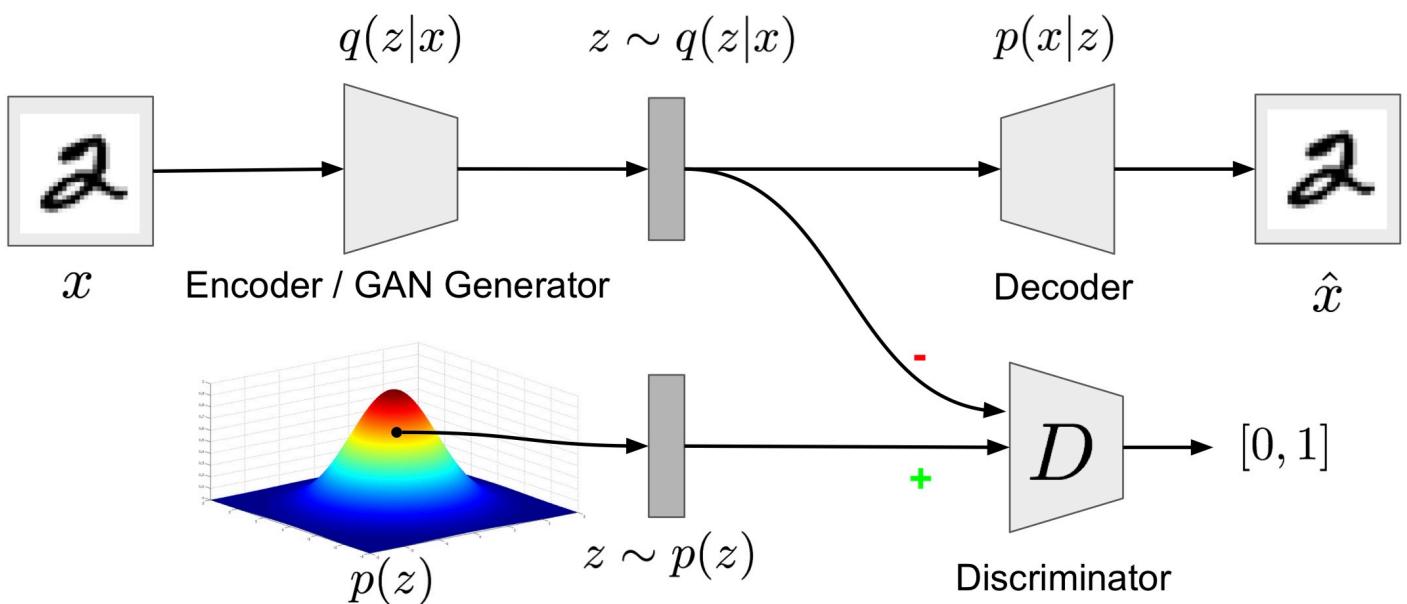
- ▶ Возьмем в качестве $q_\phi(z|x)$ нормальное распределение $\mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x))$
- ▶ Возьмем в качестве $p_\theta(x|z)$ нормальное распределение $\mathcal{N}(x; \mu_\theta(z), \sigma_\theta(z))$
- ▶ Параметризуем $z = \mu_z(x)\varepsilon_z$, где теперь $\varepsilon_z \sim \mathcal{N}(0, 1)$



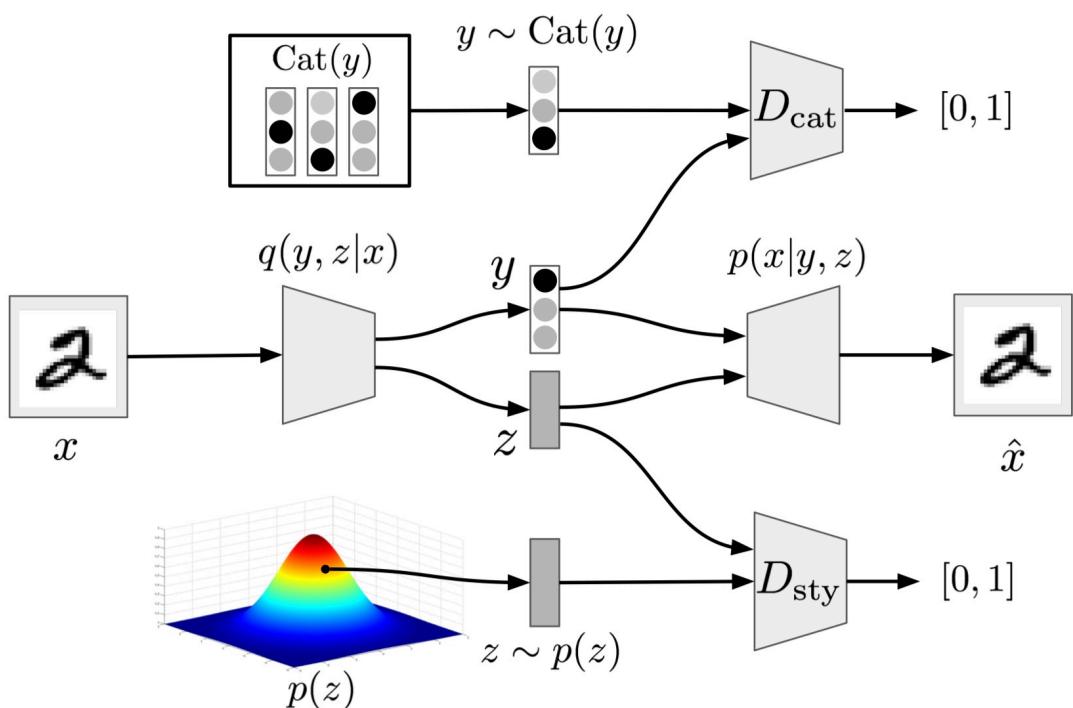
VAE



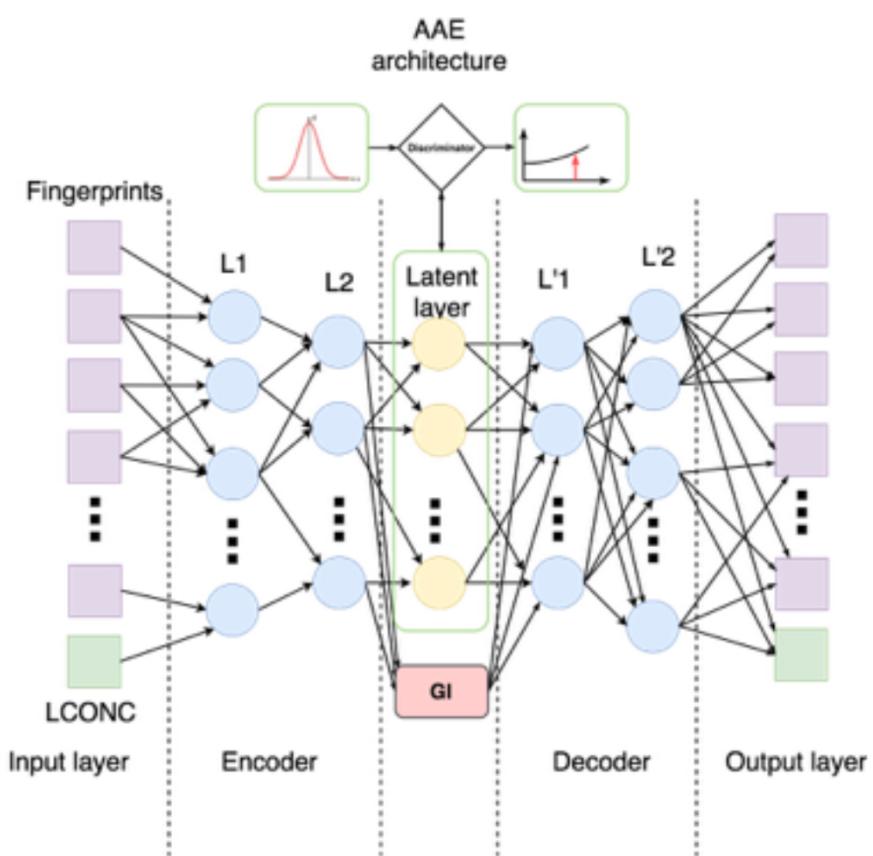
Adversarial autoencoder



AAE

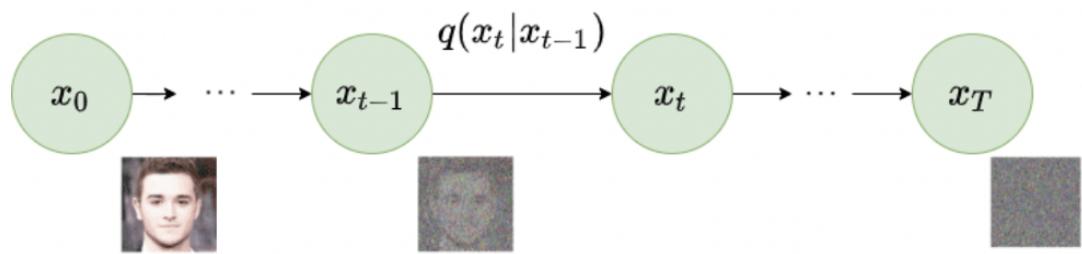


AAE

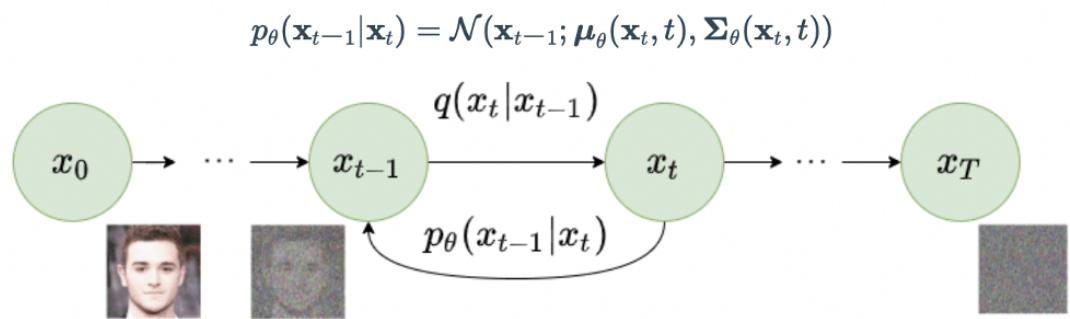


Denoising diffusion process model. Overview

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t = \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \boldsymbol{\Sigma}_t = \beta_t \mathbf{I})$$



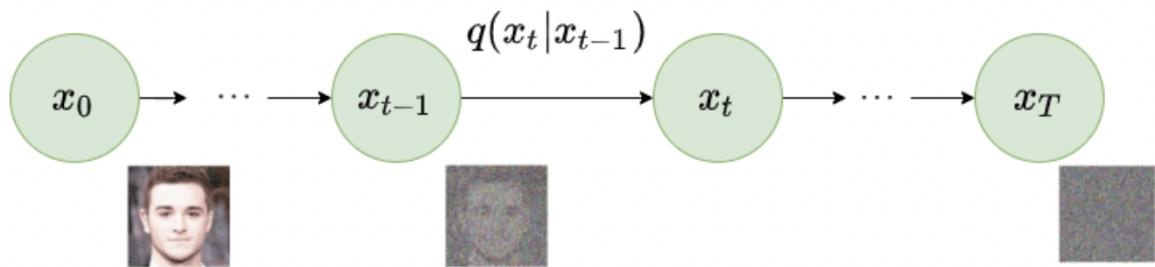
$$\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$$



Denoising diffusion process model. Forward diffusion process

Во время прямого диффузионного процесса к объекту на каждом шаге прибавляется шум. Начиная с некоторого шага T картинка фактически превращается в шум. Получаем последовательность зашумленных объектов: x_1, x_2, \dots, x_T .

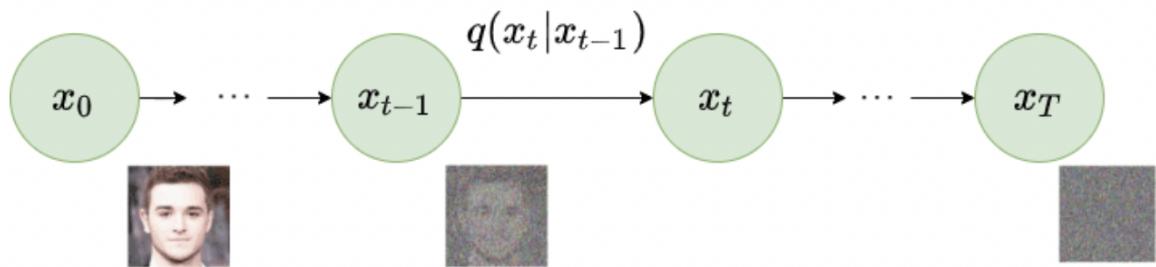
$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t = \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \boldsymbol{\Sigma}_t = \beta_t \mathbf{I})$$



Denoising diffusion process model. Forward diffusion process

Как ускорить?

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t = \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \boldsymbol{\Sigma}_t = \beta_t \mathbf{I})$$



Denoising diffusion process model. Forward diffusion process

Репараметризуем каждый шаг процесса с помощью стандартных нормальных случайных величин ε_i , где $\varepsilon_i \sim \mathcal{N}(0, I)$,

$$\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{s=0}^t \alpha_s$$

$$\begin{aligned} x_t &= \sqrt{1 - \beta_t} x_{t-1} + \sqrt{\beta_t} \varepsilon_{t-1} \\ &= \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \varepsilon_{t-1} \\ &\quad = \dots \\ &= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \varepsilon_0 \end{aligned}$$

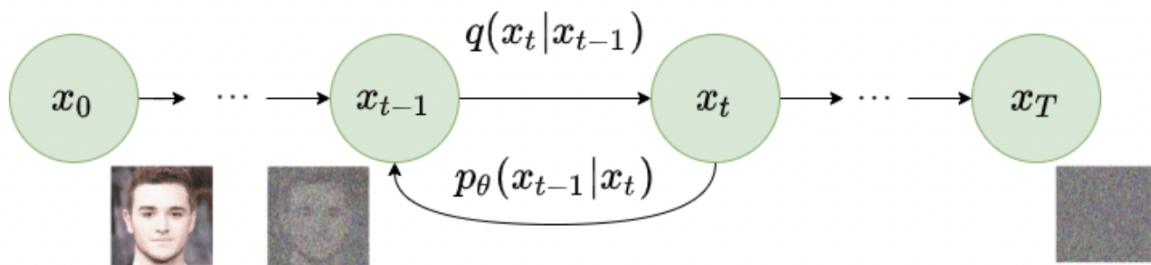
$$x_t \sim q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$$

Значения параметров β_t можно выбрать равномерно из отрезка $[0.0001, 0.02]$ или с помощью "косинусного расписания".

Denoising diffusion process model. Backward diffusion process

Для $T \gg 0$ картинка x_T стремится по распределению к $\mathcal{N}(0, I)$. Хотим выучить обратную функцию, чтобы по шуму восстановить картинку. Будем искать $q(x_{t-1}|x_t)$ в виде $\mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$, где μ_θ и Σ_θ - выходы нейронной сети.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$



Reverse diffusion process. Image modified by Ho et al. 2020

Denoising diffusion process model. Reparametrization

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

$$\begin{aligned}\tilde{\beta}_t &= 1 / \left(\frac{\alpha_t}{\beta_t} + \frac{1}{1 - \bar{\alpha}_{t-1}} \right) = 1 / \left(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t (1 - \bar{\alpha}_{t-1})} \right) = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t \\ \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) &= \left(\frac{\sqrt{\alpha_t}}{\beta_t} \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1 - \bar{\alpha}_{t-1}} \mathbf{x}_0 \right) / \left(\frac{\alpha_t}{\beta_t} + \frac{1}{1 - \bar{\alpha}_{t-1}} \right) \\ &= \left(\frac{\sqrt{\alpha_t}}{\beta_t} \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1 - \bar{\alpha}_{t-1}} \mathbf{x}_0 \right) \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t \\ &= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \mathbf{x}_0\end{aligned}$$

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_t) \\ &= \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_t \right)\end{aligned}$$

Denoising diffusion process model. Loss function

$$\begin{aligned}
-\log p_\theta(\mathbf{x}_0) &\leq -\log p_\theta(\mathbf{x}_0) + D_{\text{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)\|p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)} \right] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0) \right] \\
&= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right]
\end{aligned}$$

Let $L_{\text{VLB}} = \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0)$

$$\begin{aligned}
&= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \\
&= \mathbb{E}_q \underbrace{[D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \| p_\theta(\mathbf{x}_T))]_{L_T}}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}
\end{aligned}$$

Denoising diffusion process model. Loss function

$$\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\|_2^2} \|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{1}{2\|\boldsymbol{\Sigma}_\theta\|_2^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_t \right) - \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) \right\|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2} \|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2} \|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\|^2 \right]
\end{aligned}$$

$$\begin{aligned}
L_t^{\text{simple}} &= \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \boldsymbol{\epsilon}_t} \left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2 \right] \\
&= \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \boldsymbol{\epsilon}_t} \left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\|^2 \right]
\end{aligned}$$

Denoising diffusion process model. Training

Algorithm 1 Training

```
1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
       $\nabla_{\theta} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t)\|^2$ 
6: until converged
```

Algorithm 2 Sampling

```
1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \boldsymbol{\epsilon}_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: return  $\mathbf{x}_0$ 
```

Denoising diffusion process model. Как ускорить?

"For example, it takes around 20 hours to sample 50k images of size 32 × 32 from a DDPM, but less than a minute to do so from a GAN on an Nvidia 2080 Ti GPU."

- ▶ Strided sampling: Делаем сэмплирование с шагом > 1 .

$$\tau_1 < \tau_1 < \dots < \tau_S, S < T$$

$$q_{\sigma,\tau}(\mathbf{x}_{\tau_{i-1}}|\mathbf{x}_{\tau_t}, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{\tau_{i-1}}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \frac{\mathbf{x}_{\tau_i} - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1 - \bar{\alpha}_t}}, \sigma_t^2 \mathbf{I})$$

- ▶ Denoising diffusion implicit model. Устремим σ_t в процессе сэмплинга к 0.

Denoising diffusion process model. Classifier guidance

Пусть $f_\phi(y|x_t)$ - классификатор обученный на зашумленных объектах, можем использовать его градиенты в качестве поправки к оценке мат.ожидания x_{t-1} .

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $f_\phi(y|x_t)$, and gradient scale s .

```
Input: class label  $y$ , gradient scale  $s$ 
 $x_T \leftarrow$  sample from  $\mathcal{N}(0, \mathbf{I})$ 
for all  $t$  from  $T$  to 1 do
     $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$ 
     $x_{t-1} \leftarrow$  sample from  $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log f_\phi(y|x_t), \Sigma)$ 
end for
return  $x_0$ 
```

Denoising diffusion process model. Classifier free guidance

Добавим в качестве входа в сеть, предсказывающую шум, метку класса / эмбеддинг текста или картинки $\varepsilon = \varepsilon(x_t, t, y)$. Также будем с некоторой вероятностью подавать на вход пустой y , чтобы модель могла генерировать без условия $\varepsilon = \varepsilon(x_t, t, y = \emptyset)$.

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \\ &= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} (\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)) \\ \bar{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, y) &= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \sqrt{1-\bar{\alpha}_t} w \nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) \\ &= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) + w(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)) \\ &= (w+1)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\end{aligned}$$

Используем такой предиктор шума для сэмплирования

Denoising diffusion process model. Classifier free guidance

Algorithm 1 Joint training a diffusion model with classifier-free guidance

Require: p_{uncond} : probability of unconditional training

- 1: **repeat**
 - 2: $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$ ▷ Sample data with conditioning from the dataset
 - 3: $\mathbf{c} \leftarrow \emptyset$ with probability p_{uncond} ▷ Randomly discard conditioning to train unconditionally
 - 4: $\lambda \sim p(\lambda)$ ▷ Sample log SNR value
В обсуждённом нами варианте лямбда не сэмплируется, а вычисляется по t
 - 5: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 6: $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \epsilon$ ▷ Corrupt data to the sampled log SNR value
 - 7: Take gradient step on $\nabla_\theta \|\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \epsilon\|^2$ ▷ Optimization of denoising model
 - 8: **until** converged
-

Algorithm 2 Conditional sampling with classifier-free guidance

Require: w : guidance strength v - гиперпараметр

Require: \mathbf{c} : conditioning information for conditional sampling

Require: $\lambda_1, \dots, \lambda_T$: increasing log SNR sequence with $\lambda_1 = \lambda_{\min}$, $\lambda_T = \lambda_{\max}$

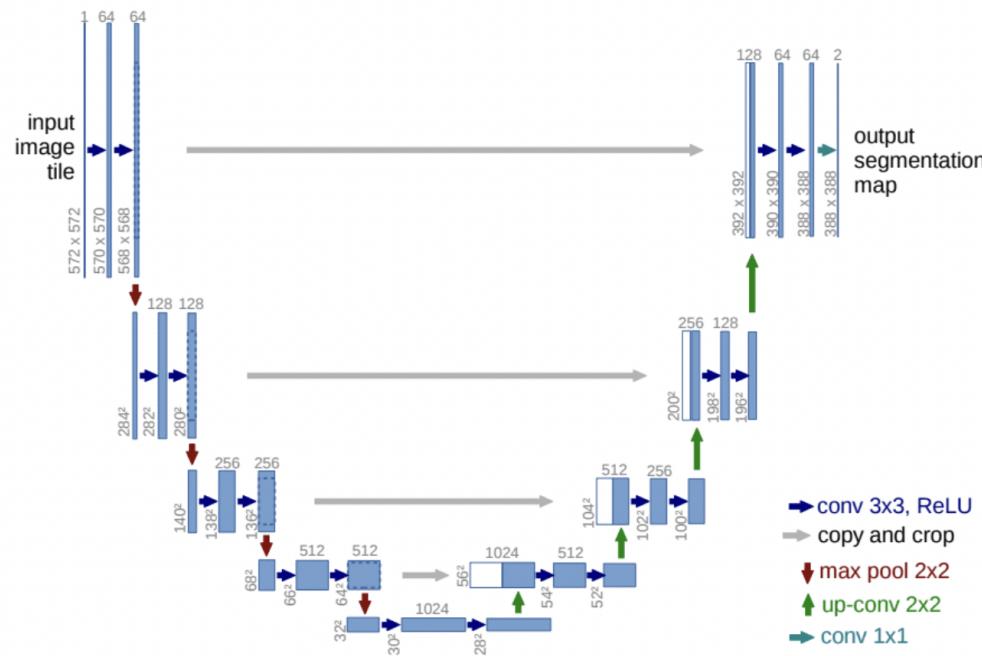
- 1: $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 2: **for** $t = 1, \dots, T$ **do**
 - ▷ Form the classifier-free guided score at log SNR λ_t
 - 3: $\tilde{\epsilon}_t = (1 + w)\epsilon_\theta(\mathbf{z}_t, \mathbf{c}) - w\epsilon_\theta(\mathbf{z}_t)$ ▷ Sampling step (could be replaced by another sampler, e.g. DDIM)
 - 4: $\tilde{\mathbf{x}}_t = (\mathbf{z}_t - \sigma_{\lambda_t} \tilde{\epsilon}_t) / \alpha_{\lambda_t}$
 - 5: $\mathbf{z}_{t+1} \sim \mathcal{N}(\tilde{\mu}_{\lambda_{t+1} | \lambda_t}(\mathbf{z}_t, \tilde{\mathbf{x}}_t), (\tilde{\sigma}_{\lambda_{t+1} | \lambda_t}^2)^{1-v} (\sigma_{\lambda_t | \lambda_{t+1}}^2)^v)$ if $t < T$ else $\mathbf{z}_{t+1} = \tilde{\mathbf{x}}_t$
 - 6: **end for**
 - 7: **return** \mathbf{z}_{T+1}
-

в простом варианте \sigma не обучается, поэтому

дисперсия детерминированная и v=1

U-Net³

В качестве denoising model (шага обратной диффузии) лучше всего себя показал аналог модели U-Net с добавлением позиционно-кодированной переменной времени *PositionalEncoding(t)*.



³<https://arxiv.org/abs/1505.04597>

Denoising diffusion process model. Latent model

Чтобы ускорить процесс диффузии, обучим автоэнкодер (энкодер(\mathcal{E})-декодер(\mathcal{D}) архитектуру) и будем применять диффузионный процесс к латентному представлению. В качестве энкодера можно использовать

- ▶ Аналог VAE-GAN
- ▶ Аналог VQ-VAE / VQGAN

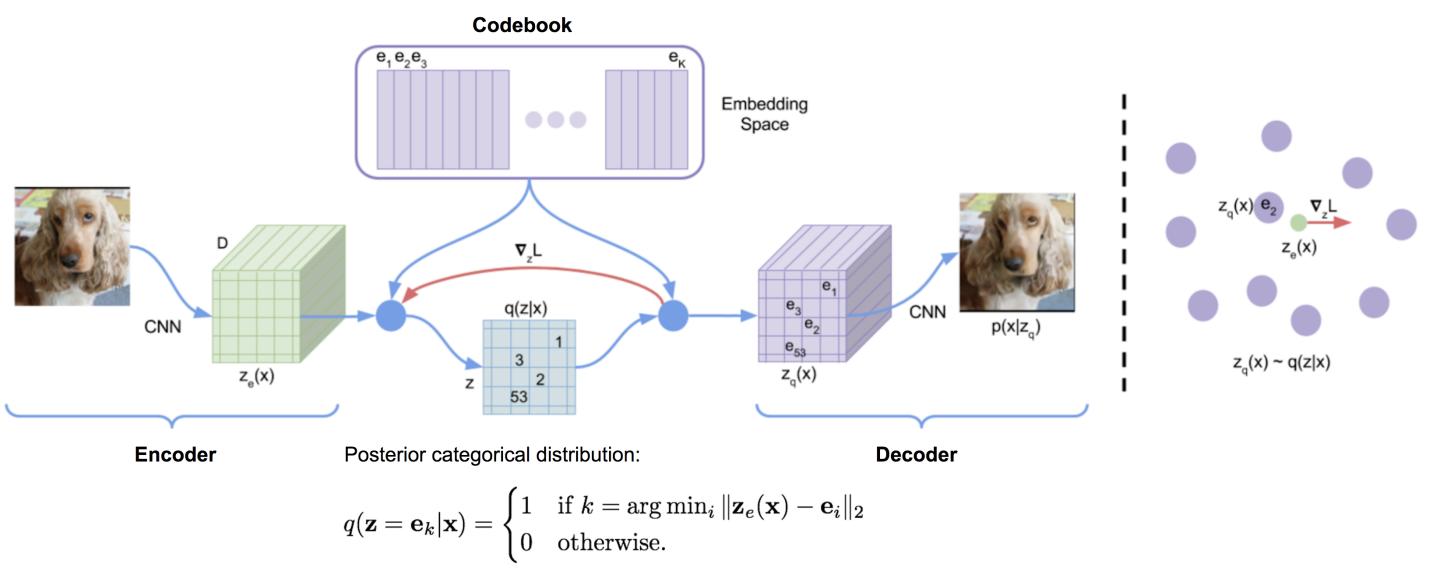
$$L_{\text{Autoencoder}} = \min_{\mathcal{E}, \mathcal{D}} \max_{\psi} \left(L_{rec}(x, \mathcal{D}(\mathcal{E}(x))) - L_{adv}(\mathcal{D}(\mathcal{E}(x))) + \log D_{\psi}(x) + L_{reg}(x; \mathcal{E}, \mathcal{D}) \right)$$

4

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x), \epsilon \sim \mathcal{N}(0, 1), t} \left[\|\epsilon - \epsilon_{\theta}(z_t, t)\|_2^2 \right]$$

⁴<https://arxiv.org/pdf/2012.09841.pdf>

VQ-VAE



Denoising diffusion process model. Latent model

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) \cdot \mathbf{V}$$

where $\mathbf{Q} = \mathbf{W}_Q^{(i)} \cdot \varphi_i(\mathbf{z}_i)$, $\mathbf{K} = \mathbf{W}_K^{(i)} \cdot \tau_\theta(y)$, $\mathbf{V} = \mathbf{W}_V^{(i)} \cdot \tau_\theta(y)$

and $\mathbf{W}_Q^{(i)} \in \mathbb{R}^{d \times d_\epsilon^i}$, $\mathbf{W}_K^{(i)}, \mathbf{W}_V^{(i)} \in \mathbb{R}^{d \times d_\tau}$, $\varphi_i(\mathbf{z}_i) \in \mathbb{R}^{N \times d_\epsilon^i}$, $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$

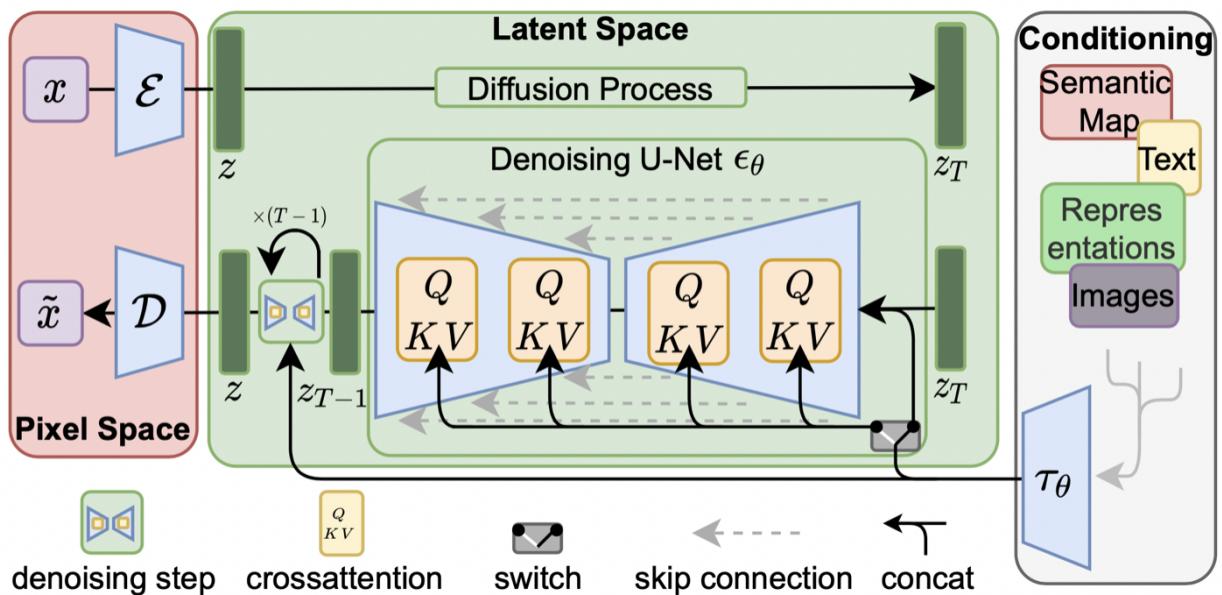


Fig. 9. The architecture of latent diffusion model. (Image source: [Rombach & Blattmann, et al. 2022](#))

Stable diffusion примеры

Безусловная генерация.



Figure 4. Samples from *LDMs* trained on CelebAHQ [39], FFHQ [41], LSUN-Churches [102], LSUN-Bedrooms [102] and class-conditional ImageNet [12], each with a resolution of 256×256 . Best viewed when zoomed in. For more samples cf. the supplement.

Stable diffusion примеры

Условная генерация по тексту

Text-to-Image Synthesis on LAION. 1.45B Model.						
'A street sign that reads "Latent Diffusion"'	'A zombie in the style of Picasso'	'An image of an animal half mouse half octopus'	'An illustration of a slightly conscious neural network'	'A painting of a squirrel eating a burger'	'A watercolor painting of a chair that looks like an octopus'	'A shirt with the inscription: "I love generative models!"'

Stable diffusion inpainting



Figure 11. Qualitative results on object removal with our *big, w/ fi* inpainting model. For more results, see Fig. 22.

$$\begin{aligned}x_{t-1}^{\text{known}} &\sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I}) \\x_{t-1}^{\text{unknown}} &\sim \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \\x_{t-1} &= m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}\end{aligned}$$

Другие модели связанные со Stable diffusion

- ▶ DALLE-2
- ▶ Imagen
- ▶ KANDINSKY 2.0

Еще ссылки

1. [Original VAE paper](#)
2. [Original VAE paper](#)
3. [Blogpost about diffusion](#)
4. [Annotated diffusion](#)
5. [Denoising Diffusion Probabilistic Models](#)
6. [Denoising Diffusion Implicit Models](#)
7. [Improved Denoising Diffusion Probabilistic Models](#)
8. [Diffusion Models Beat GANs on Image Synthesis](#)
9. [Classifier-Free Diffusion Guidance](#)
10. [GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models](#)
11. [Cascaded Diffusion Models for High Fidelity Image Generation](#)
12. [Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding](#)
13. [High-Resolution Image Synthesis with Latent Diffusion Models](#)



Вопросы