

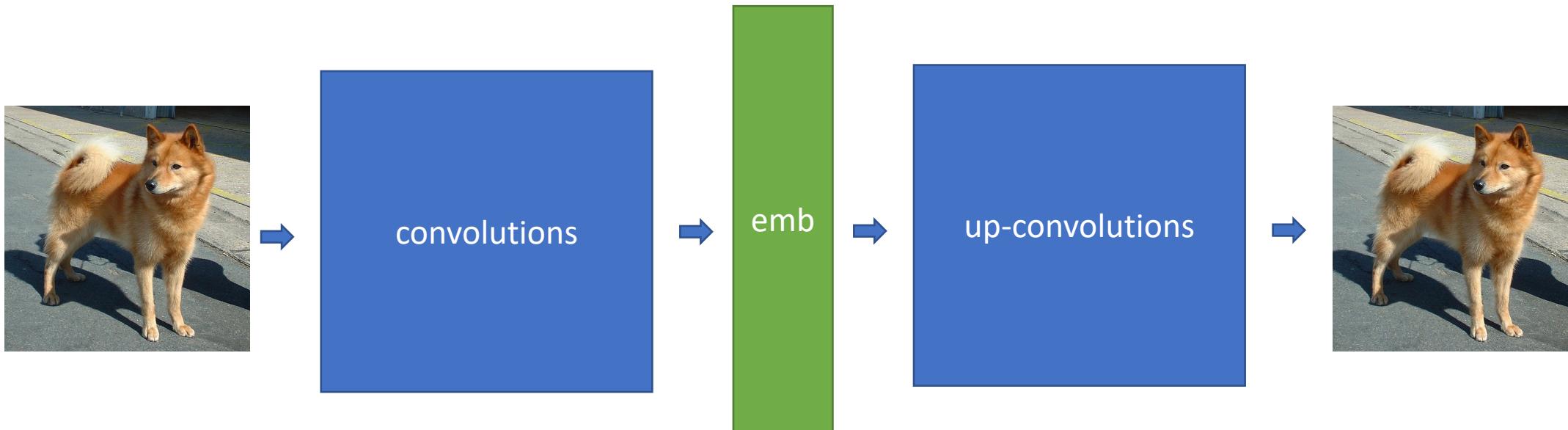
Прикладные задачи анализа данных

Лекция 2
Вариационные автокодировщики

Евгений Соколов
esokolov@hse.ru

НИУ ВШЭ, 2021

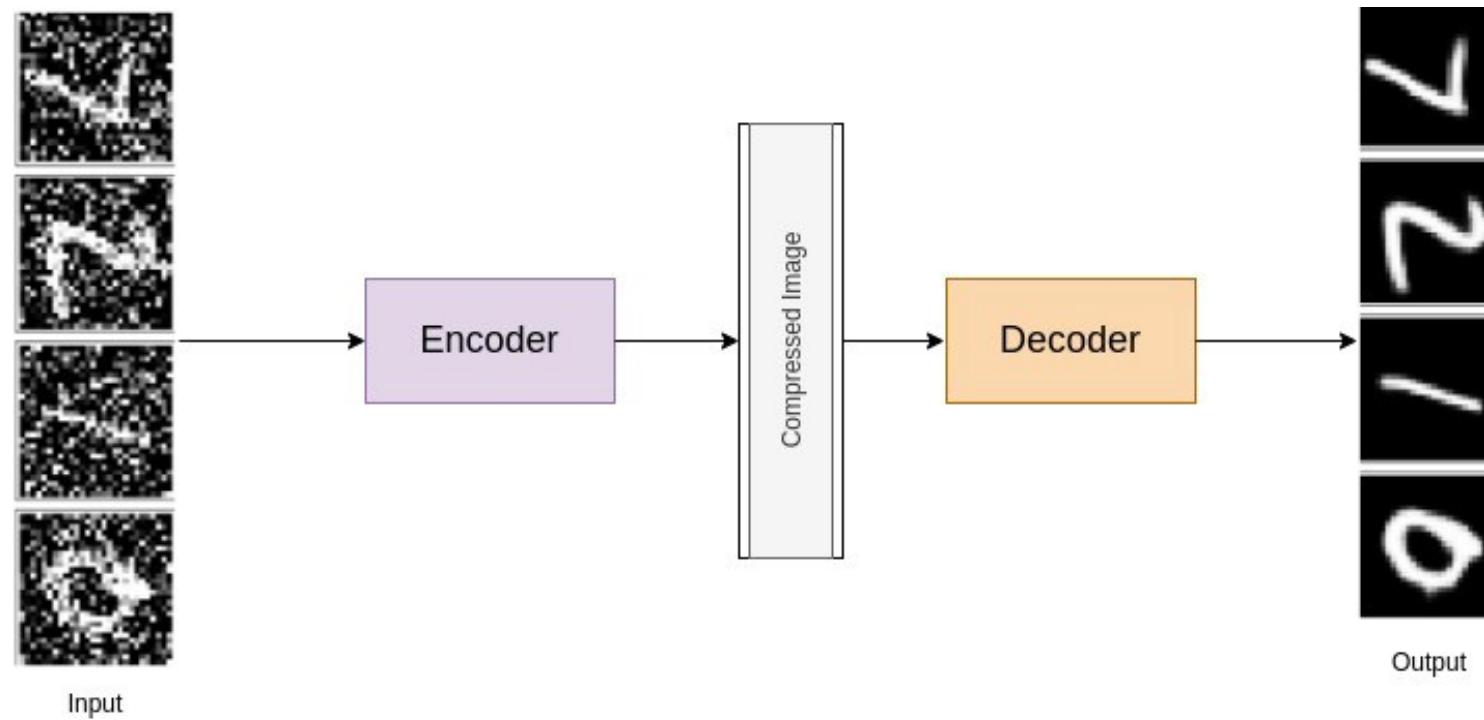
Авто кодировщики



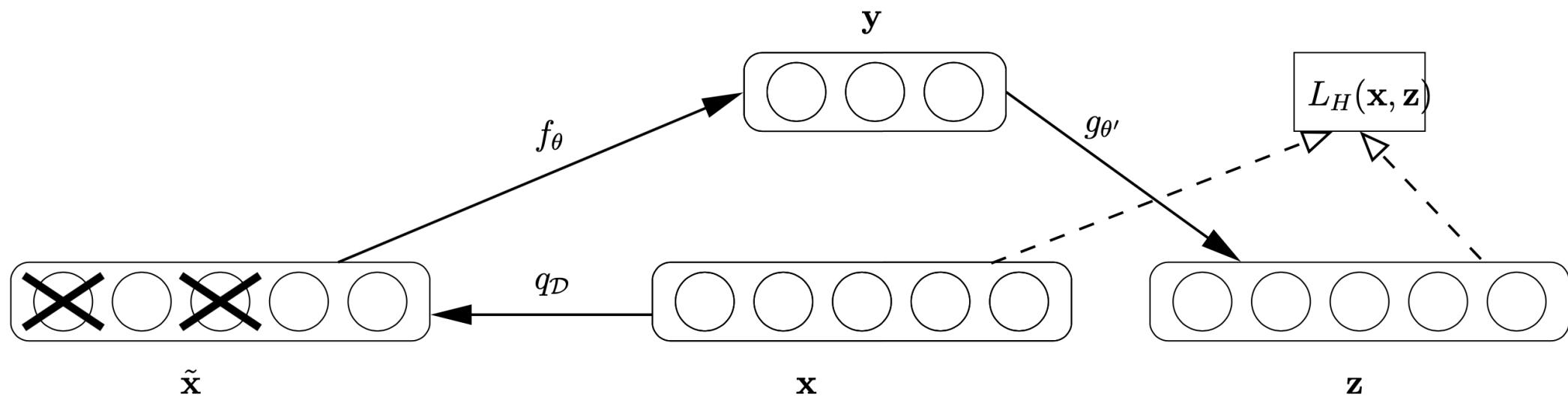
$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(x_i, g(f(x_i))) \rightarrow \min$$

x_i — изображение
 $f(x)$ — кодировщик (encoder)
 $g(z)$ — декодировщик (decoder)
 $L(x, \hat{x})$ — расстояние между изображениями (например, евклидово)

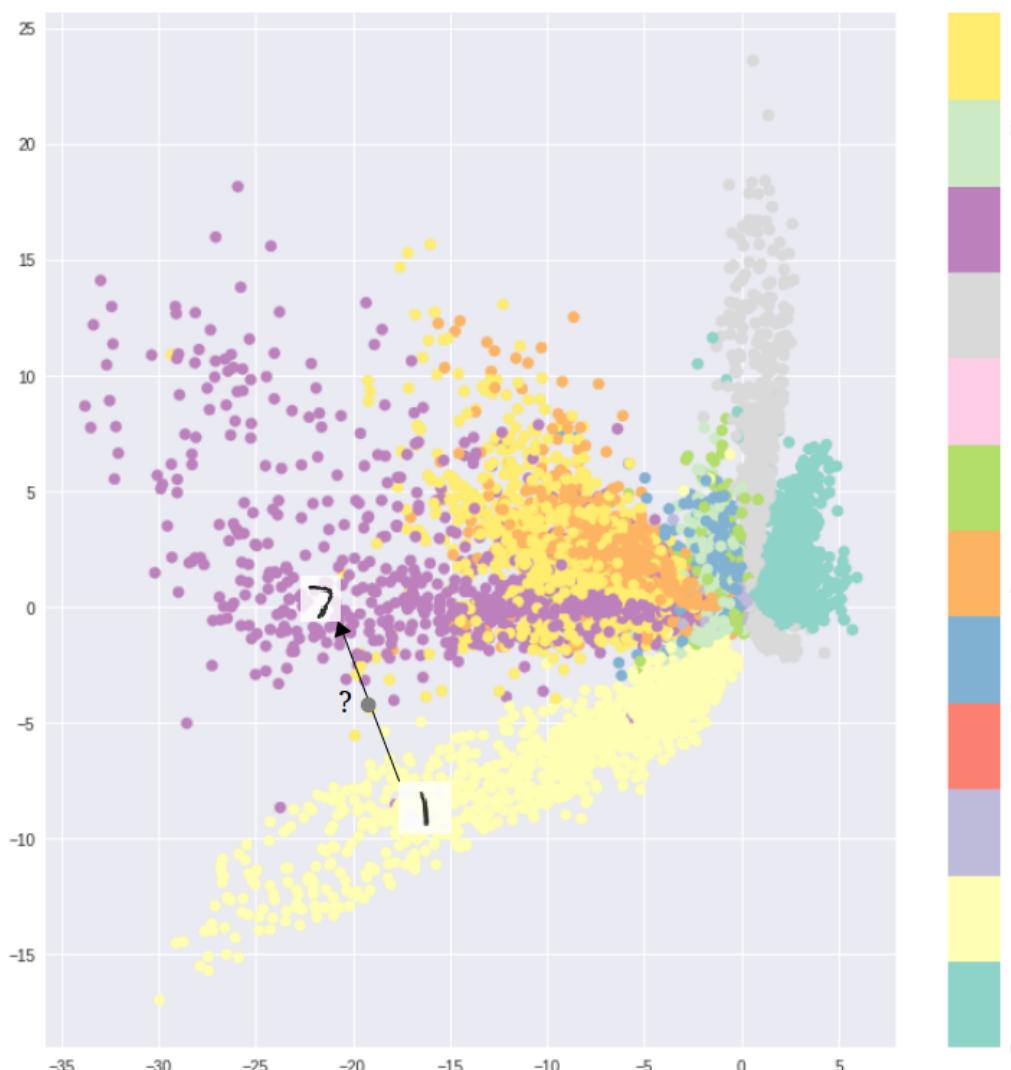
Denoising autoencoder



Denoising autoencoder



Представления изображений



Цель

- Научиться генерировать изображения
- Поискать вдохновение в вероятностных методах

Вероятностный подход

- Описываем, как генерируются ответы
- Подбираем параметры распределений так, чтобы обучающая выборка имела высокую вероятность
- Иногда полезно думать о задаче именно в терминах распределений
- Можно получать оценки неопределенности модели, а не только прогноз

Тематическое моделирование

- Данные — набор текстов, каждый представлен мешком слов
- У нас T тем
- Каждая тема t — распределение на словах $\phi_t = (\phi_{t1}, \dots, \phi_{tN})$
- Каждый текст d — распределение на темах $\theta_d = (\theta_{d1}, \dots, \theta_{dT})$
- Процесс генерации текста при заданных θ_d :
 - Выбираем тему
 - Выбираем слово из этой темы
 - Добавляем это слово в текст

Анализ экзаменационных вариантов

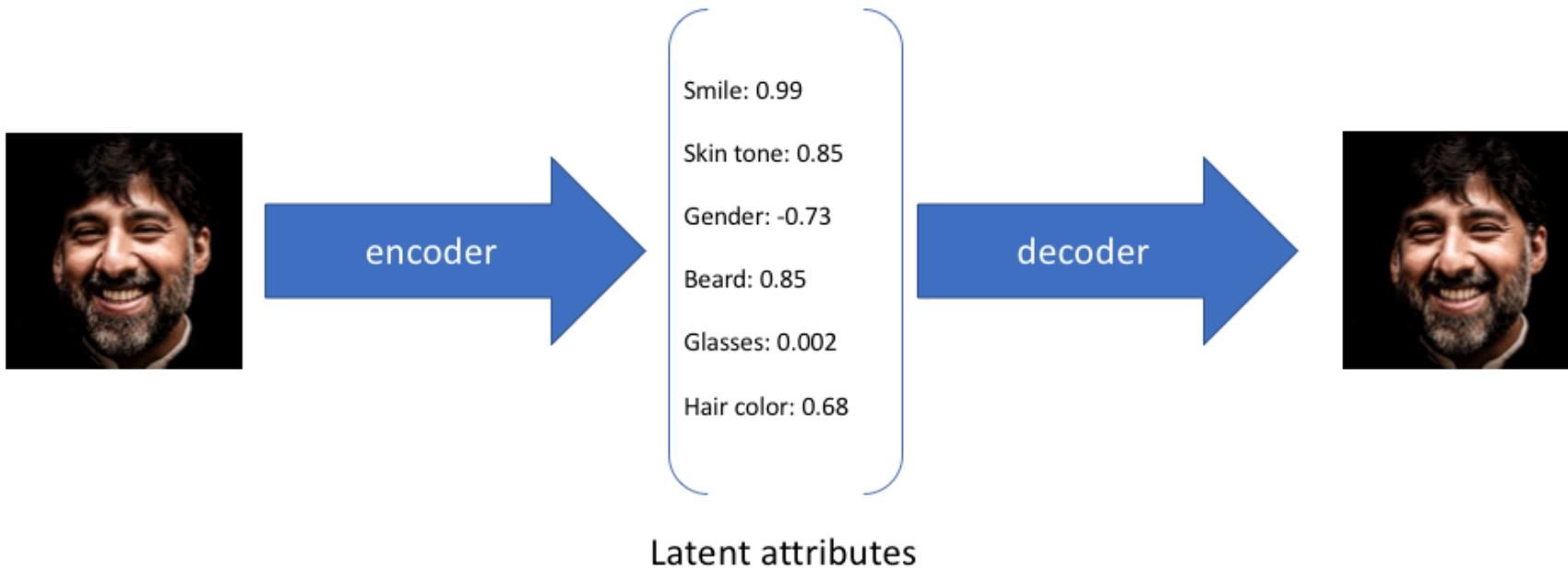
- Данные — результаты экзамена
- Студент s получил за задачу t оценку g_{st}
- У каждого студента своя сила $\alpha_s \in (-\infty, +\infty)$
- У каждого задания своя сложность $\frac{1}{\beta_t} \in [0, +\infty)$
- Вероятность решения задачи: $\frac{1}{1+\exp(-\alpha_s \beta_t)}$

Вариационные автокодировщики

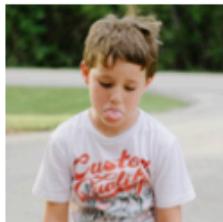
Источники

- <https://www.jeremyjordan.me/variational-autoencoders/>
- <https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf>
- <https://arxiv.org/abs/1312.6114>
- Весь материал взят из этих статей

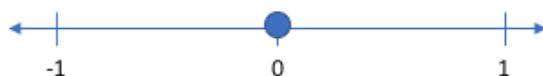
Авто кодировщик



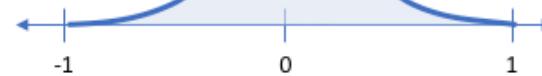
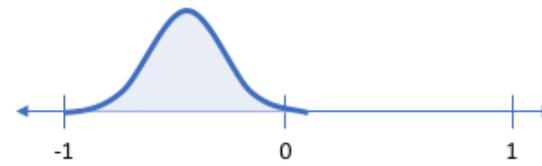
Распределения вместо точечных оценок



Smile (discrete value)

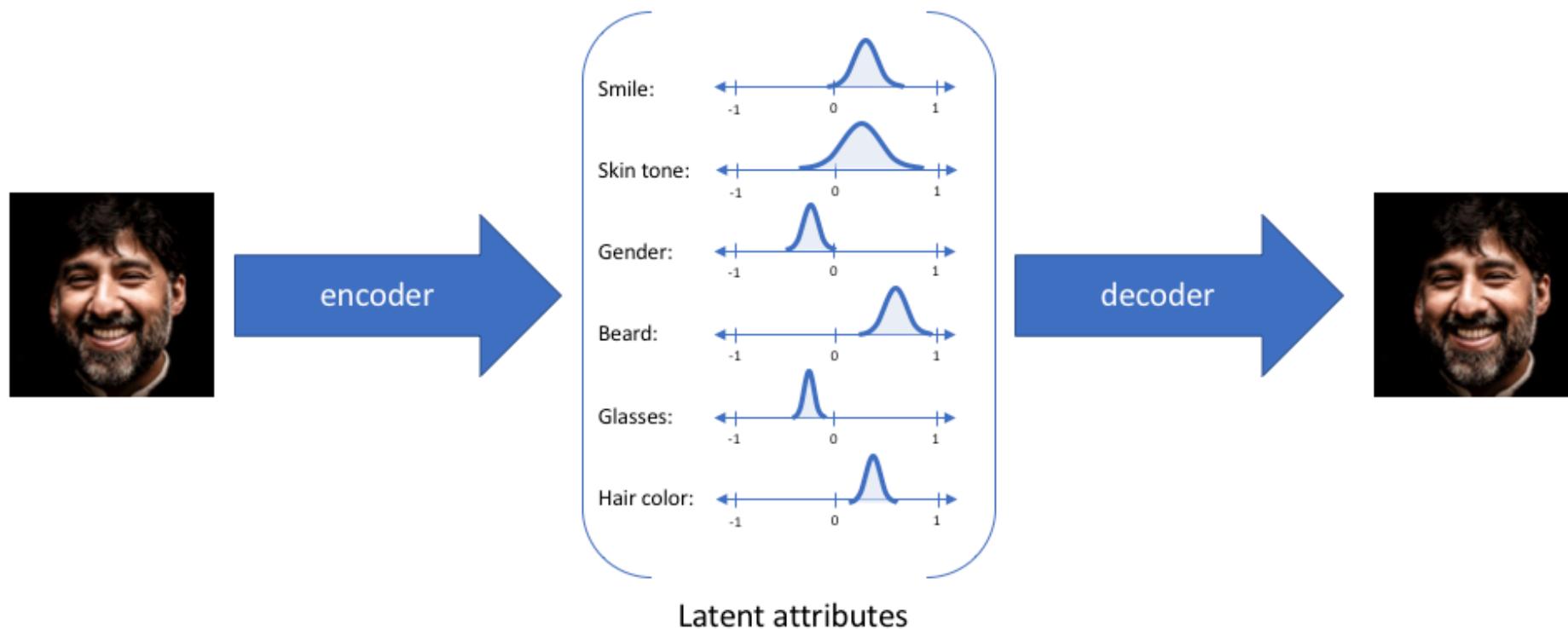


Smile (probability distribution)



vs.

Вероятностные представления



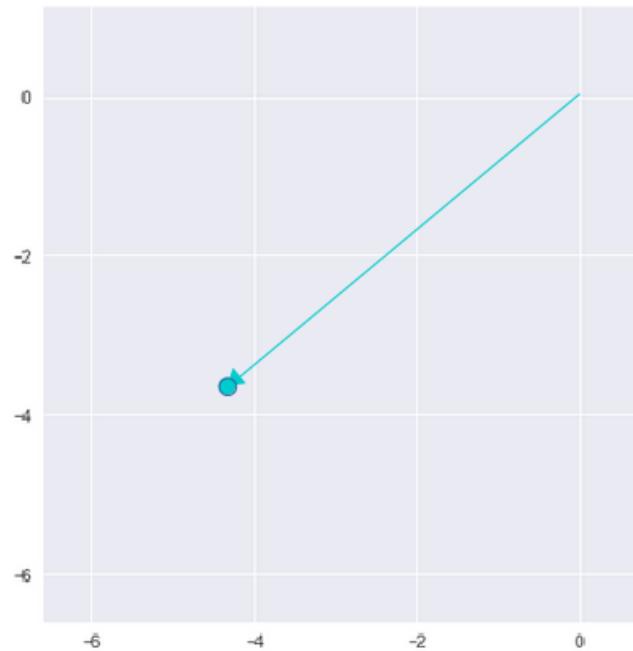
Вероятностные представления

- Хотим построить пространство представлений (\mathbb{R}^d)
- Картинка соответствует *распределению* в этом пространстве
- Пусть это будет нормальное распределение

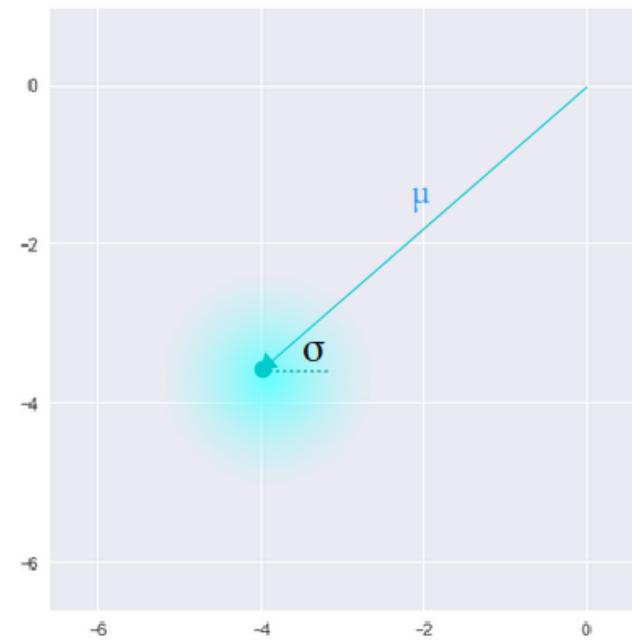
$$\text{encoder}(x) = (\mu(x), \sigma(x))$$

- $\mu(x), \sigma(x)$ — векторы размера d
- Среднее и дисперсия по каждому измерению

Вероятностные представления



Standard Autoencoder
(direct encoding coordinates)



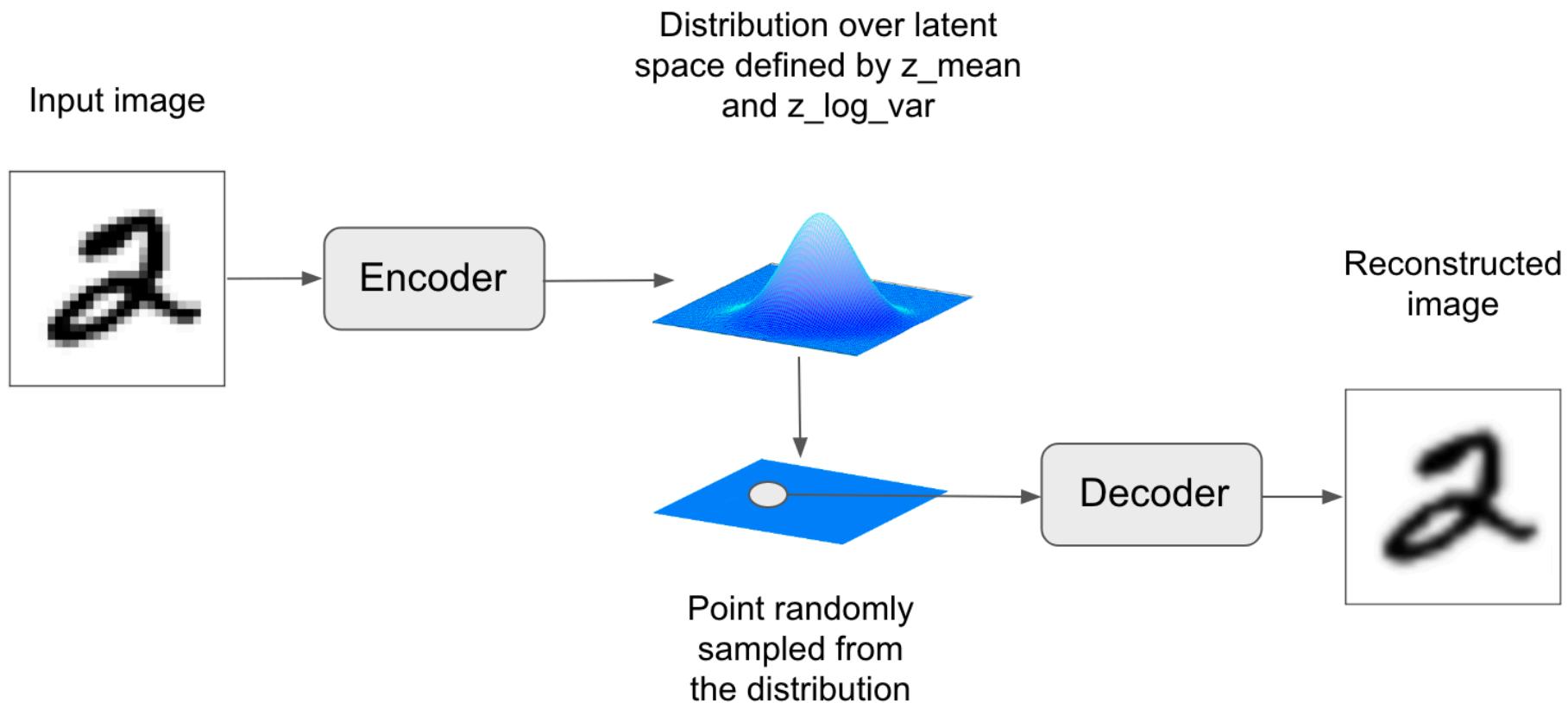
Variational Autoencoder
(μ and σ initialize a probability distribution)

Вероятностные представления

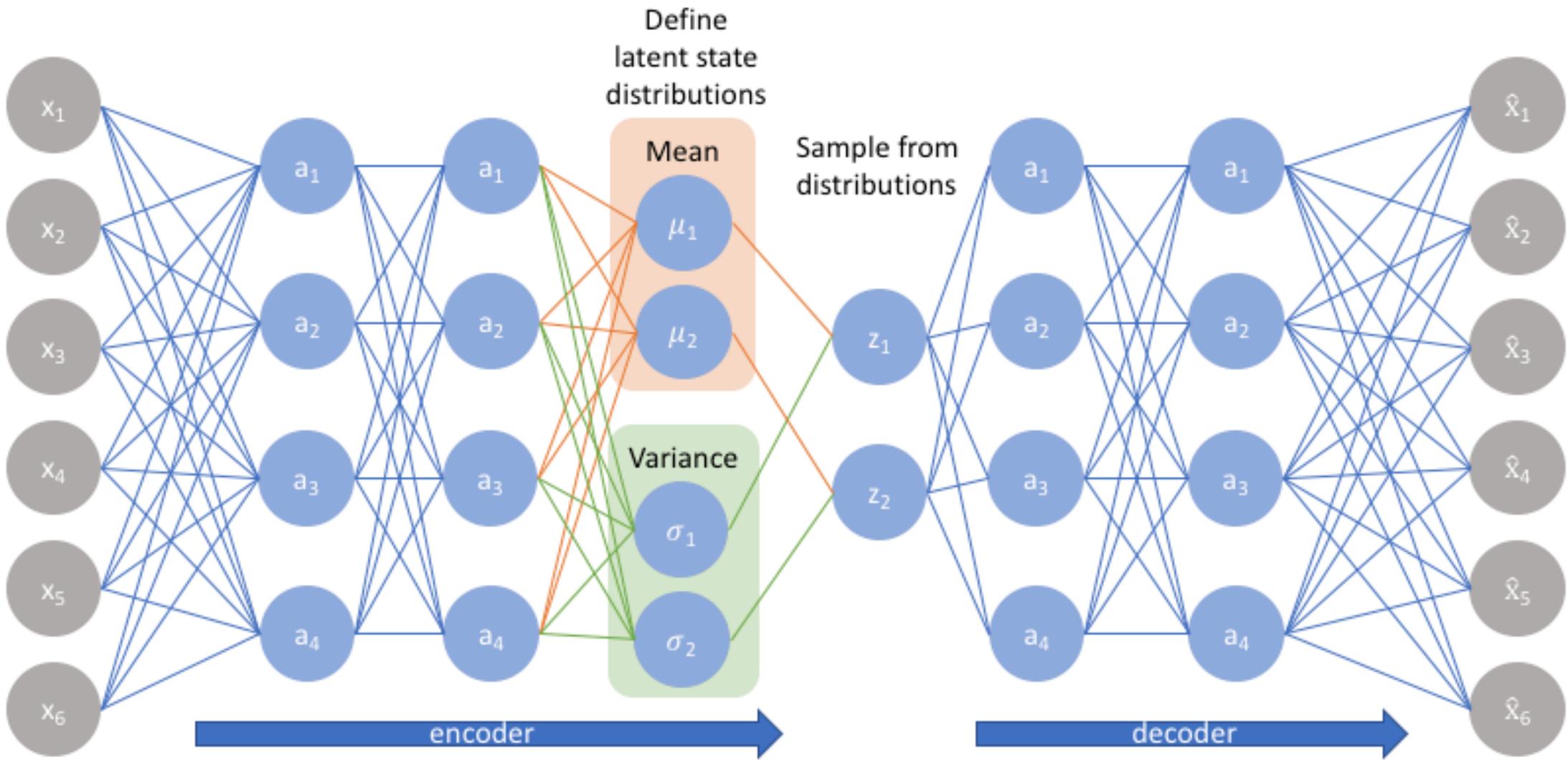
$$\text{encoder}(x) = (\mu(x), \sigma(x))$$

- Семплируем вектор z из такого распределения
- Вектор раскодируется: $\text{decoder}(z) = \tilde{x}$
- Раскодированная картинка \tilde{x} должна быть похожа на исходную x (для любого вектора, семплированного из распределения)

Вероятностные представления



Вероятностные представления



Вариационный автокодировщик

$$\sum_{i=1}^{\ell} \left(\mathbb{E}_{q(z|x_i)} \log p(x_i|z) - \text{KL}(q(z|x) \parallel \mathcal{N}(0, 1)) \right) \rightarrow \max$$

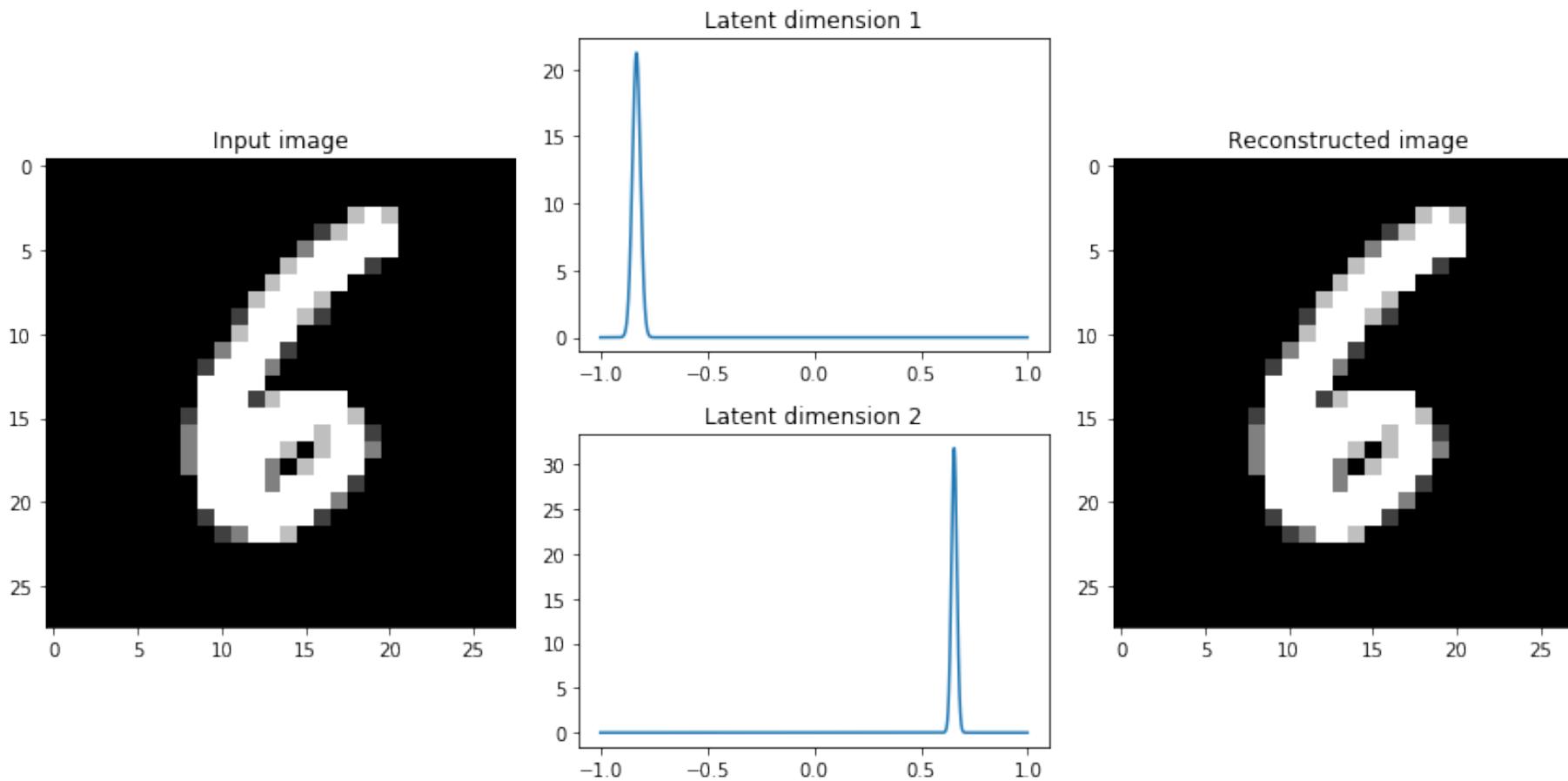
- $q(z|x)$ — кодировщик (полносвязная или свёрточная нейросеть, z — средние и дисперсии)
- $p(x|z) = \text{decoder}(z) + \varepsilon$ — декодировщик (полносвязная или свёрточная нейросеть)
- $\mathbb{E}_{q(z|x_i)} \log p(x_i|z)$ — как бы закодировали x_i , сгенерировали все возможные представления и посчитали среднюю ошибку реконструкции

Вариационный автокодировщик

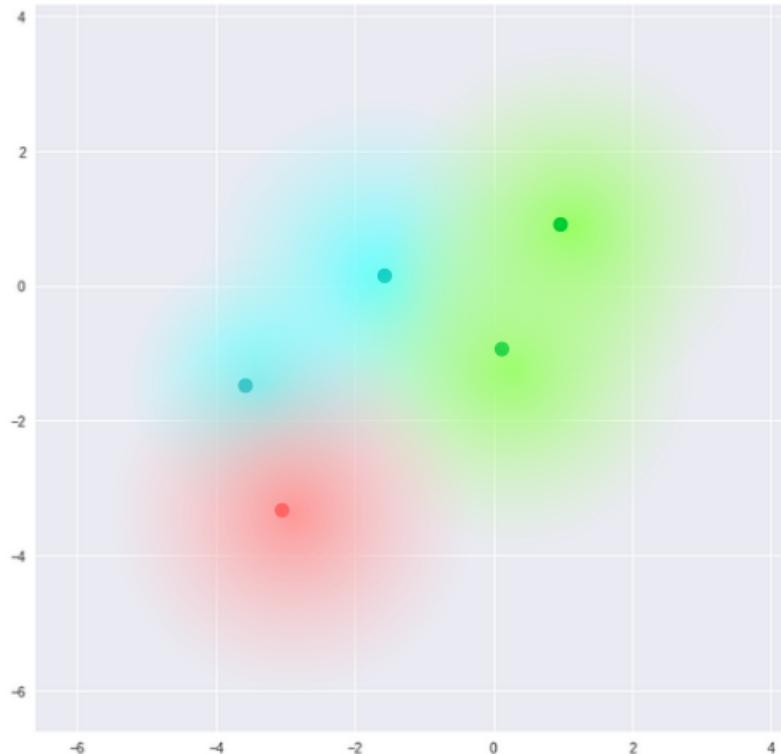
$$\sum_{i=1}^{\ell} \left(\mathbb{E}_{q(z|x_i)} \log p(x_i|z) - \text{KL}(q(z|x) \parallel \mathcal{N}(0, 1)) \right) \rightarrow \max$$

- $\text{KL}(q(z) \parallel p(z)) = \int_{-\infty}^{+\infty} q(z) \log \frac{q(z)}{p(z)} dz$ — дивергенция Кульбака-Лейблера (расстояние между распределениями)
- $\text{KL}(q(z|x) \parallel \mathcal{N}(0, 1))$ — требуем, чтобы $q(z|x)$ было как можно более похоже на стандартное нормальное
- Если $q(z|x)$ вырожденное, то это переобучение

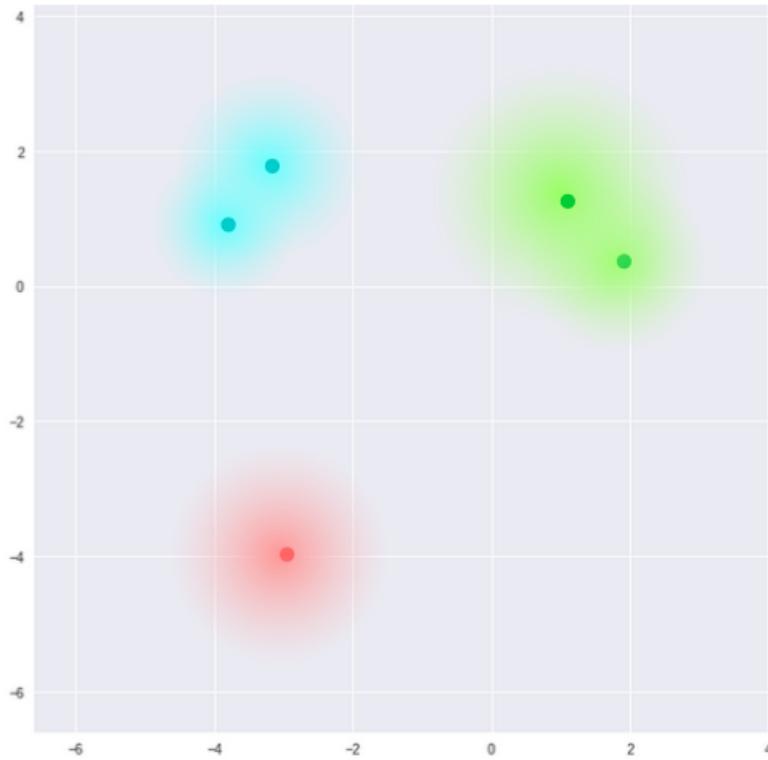
Не хотим так



Непрерывность пространства представлений



What we require



What we may inadvertently end up with

Вариационный автокодировщик

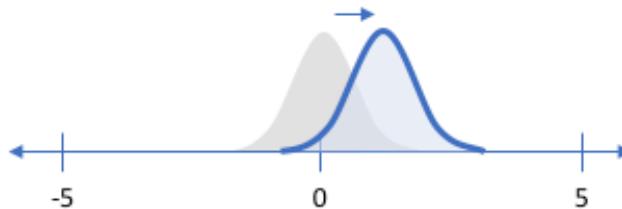
$$\sum_{i=1}^{\ell} \left(\mathbb{E}_{q(z|x_i)} \log p(x_i|z) - \text{KL}(q(z|x) \parallel \mathcal{N}(0, 1)) \right) \rightarrow \max$$

«функция потерь»
reconstruction likelihood

«регуляризатор»

Вариационный автокодировщик

Penalizing reconstruction loss
encourages the distribution to
describe the input



Our distribution
deviates from the
prior to describe
some characteristic
of the data

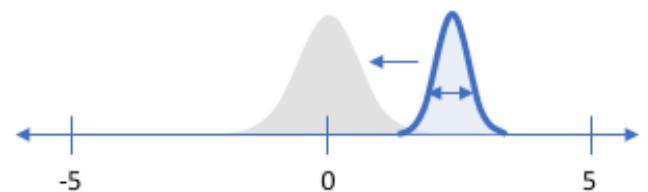
Without regularization, our
network can “cheat” by learning
narrow distributions



With a small
enough variance,
this distribution is
effectively only
representing a
single value

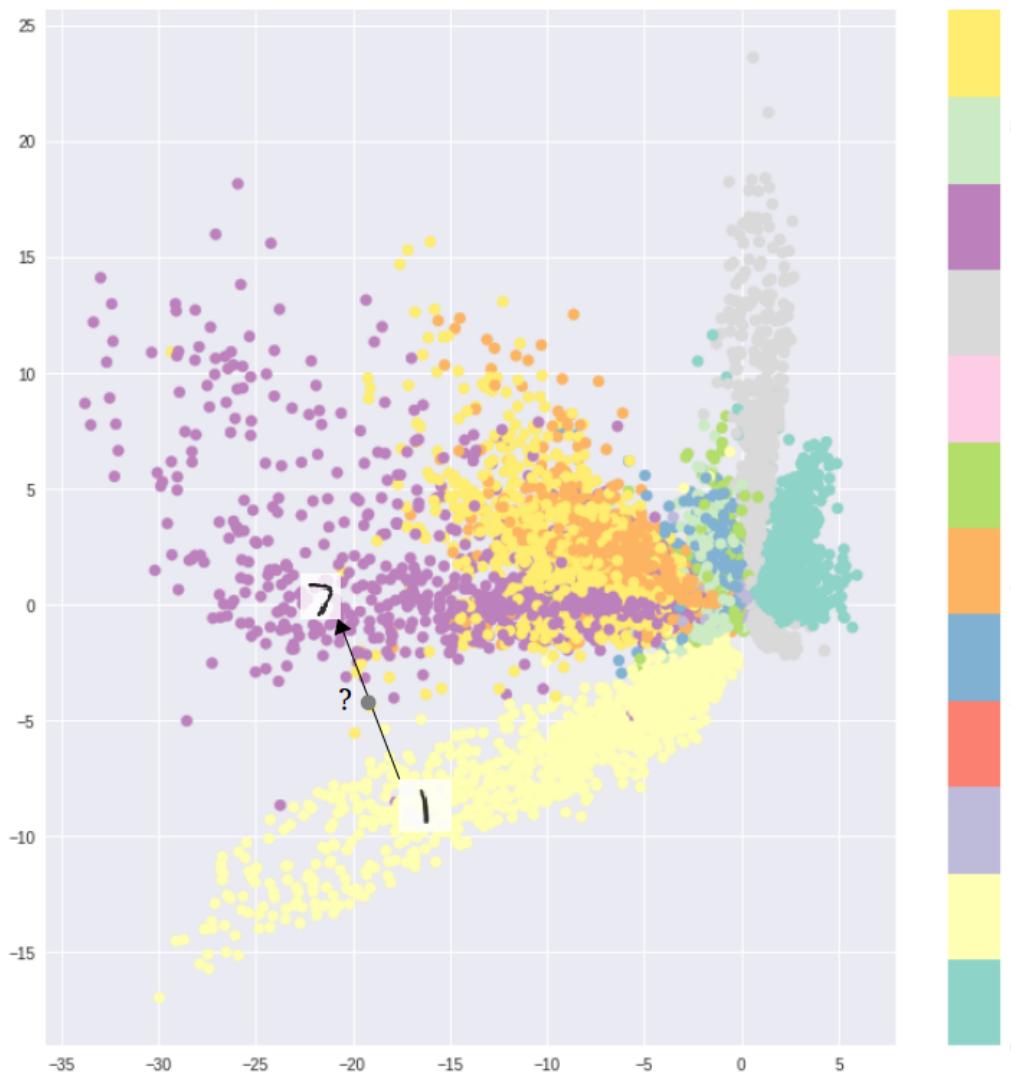
Penalizing KL divergence
acts as a regularizing force

Attract distribution
to have zero mean

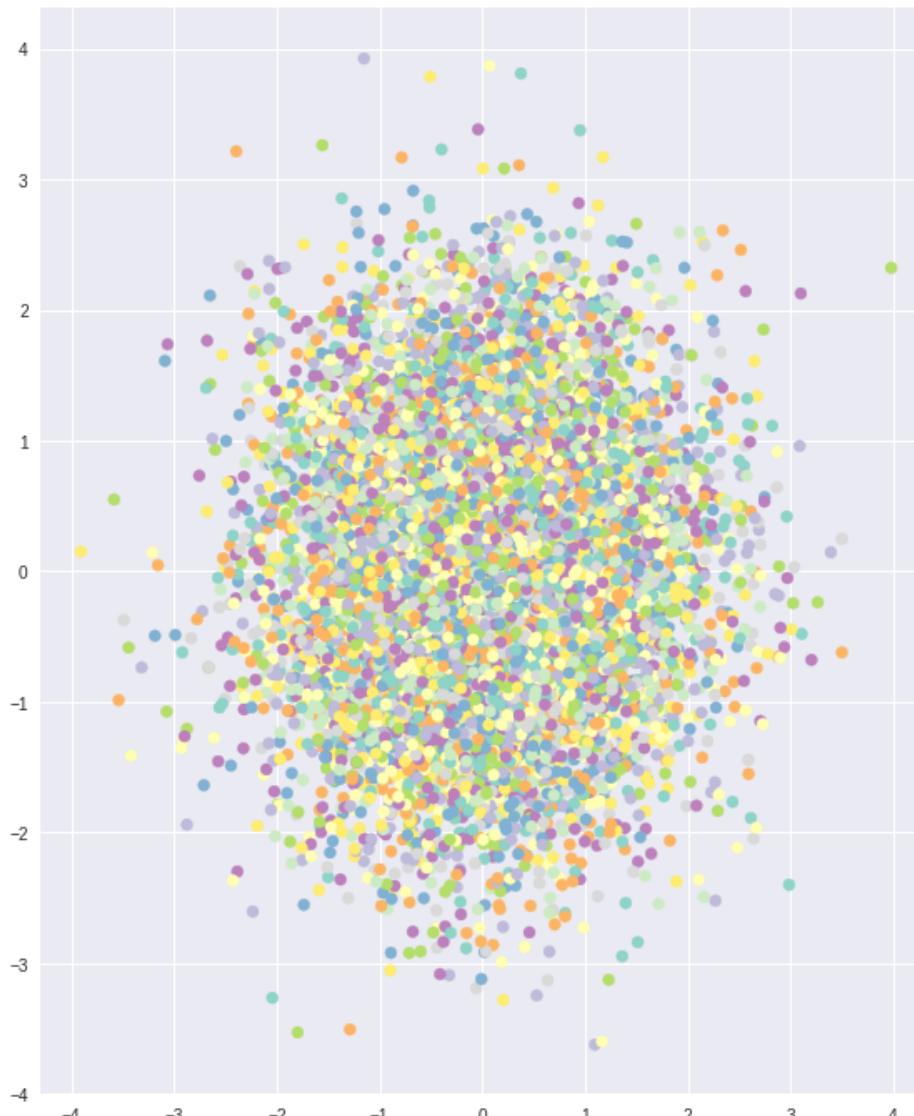


Ensure sufficient variance to
yield a smooth latent space

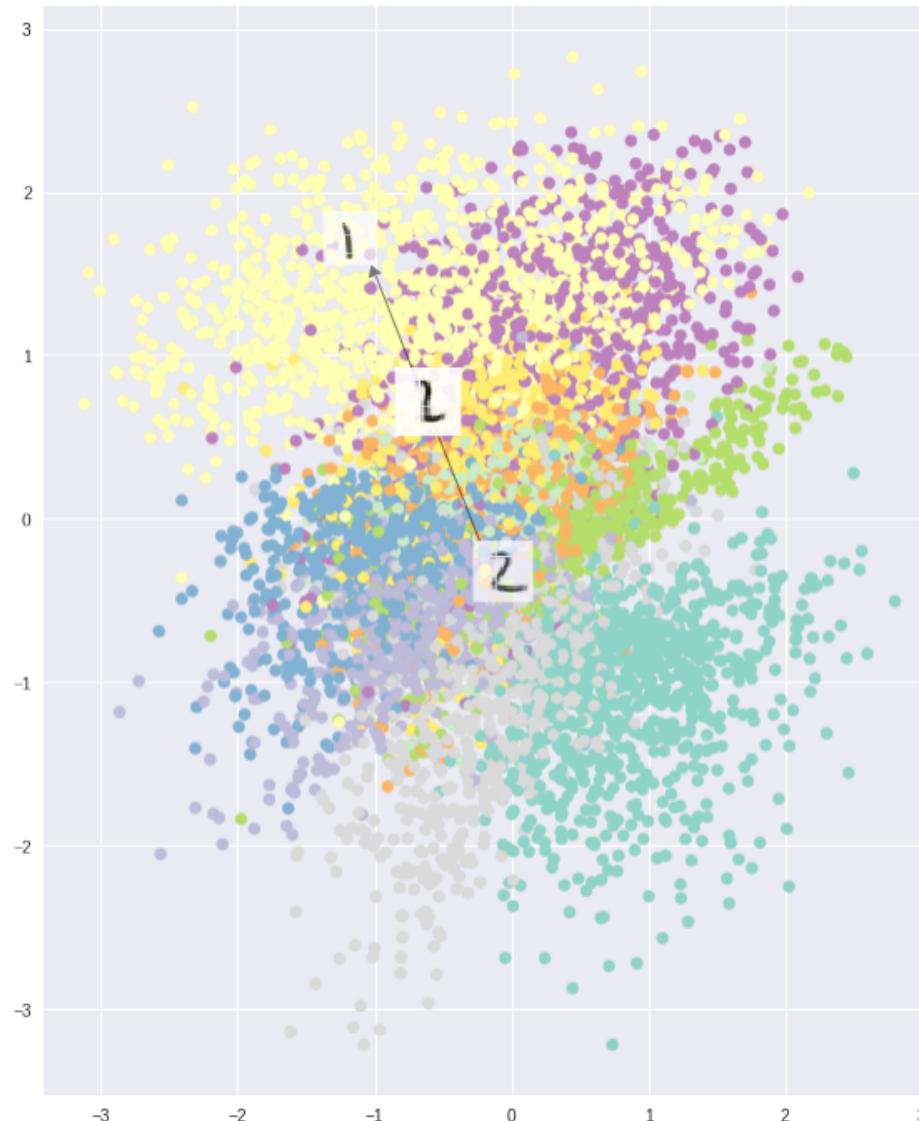
Только reconstruction likelihood



Только KL-дивергенция

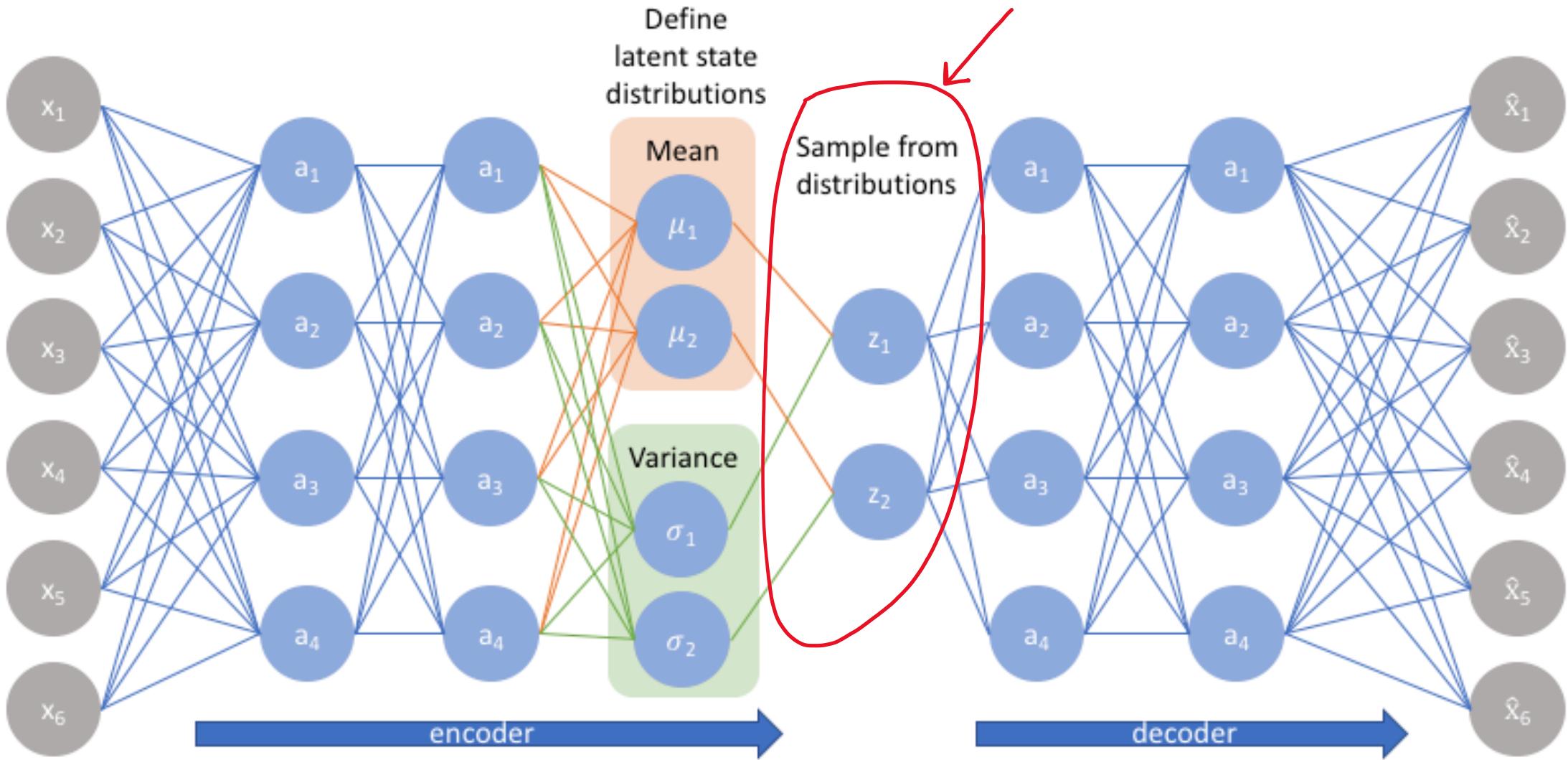


Всё вместе

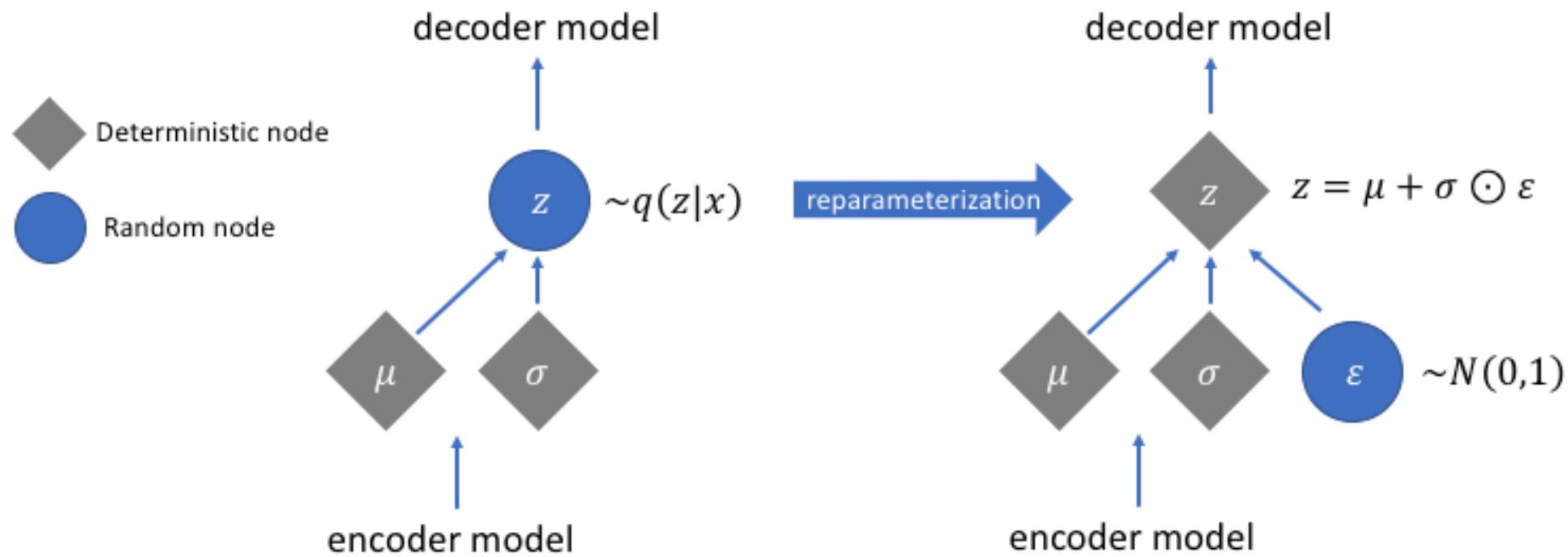


Оптимизация в VAE

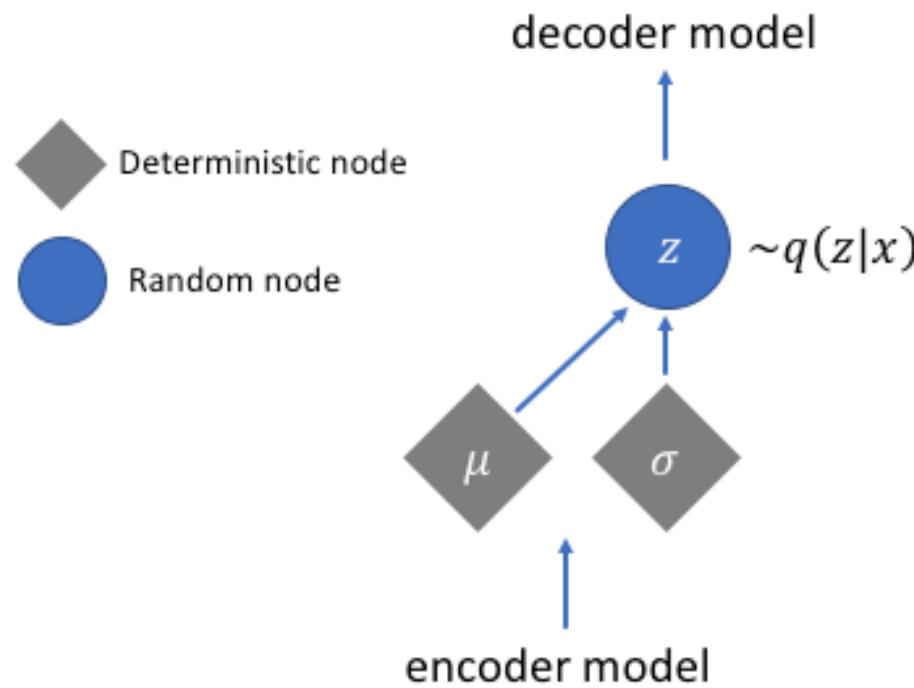
Какая-то вероятностная штука,
как считать градиенты?



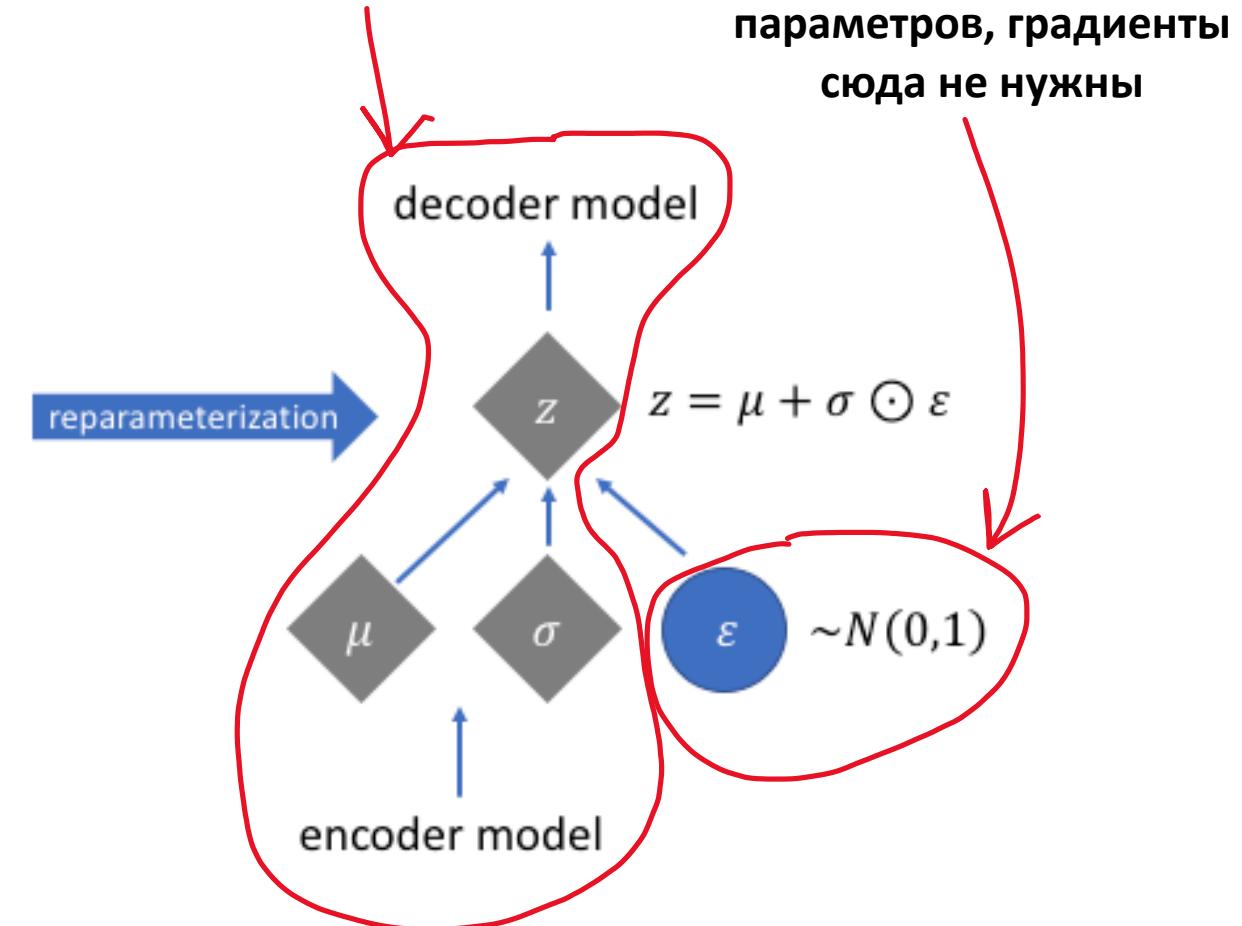
Оптимизация в VAE



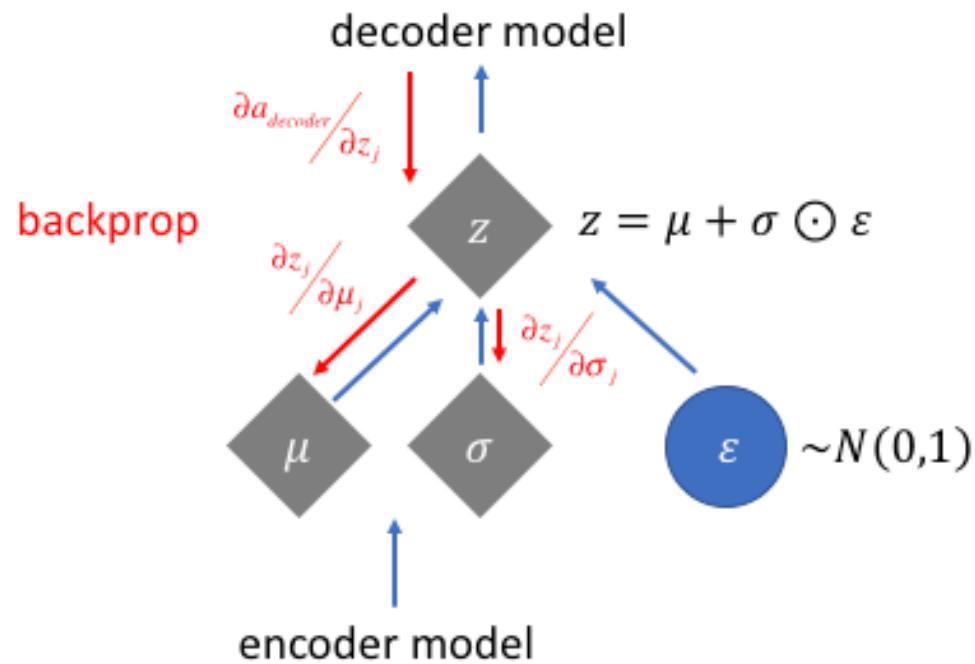
Оптимизация в VAE



Ни каких случайностей,
можем считать градиенты



Оптимизация в VAE

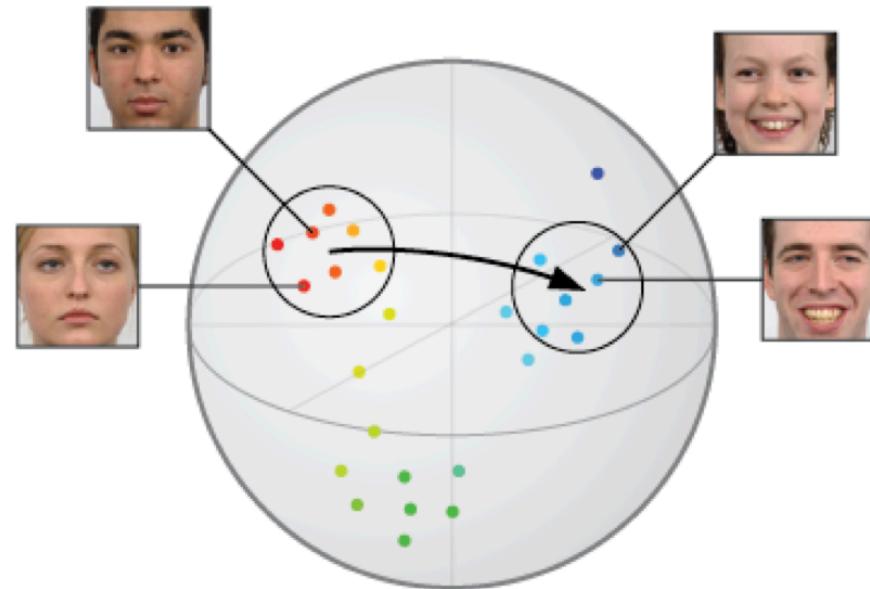


Генерация

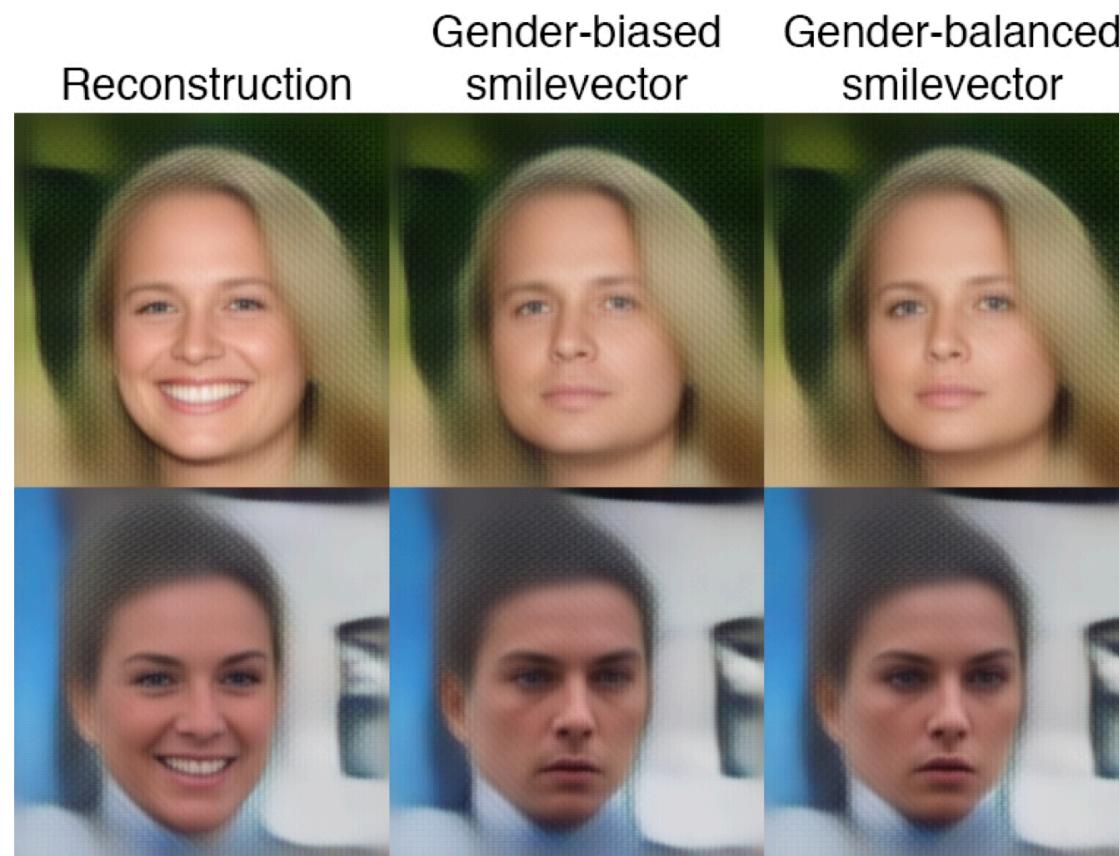
The image displays a massive grid of binary digits (0s and 1s), arranged in a regular pattern. The pattern consists of several parallel diagonal lines that slope upwards from left to right. These lines are composed of alternating 0s and 1s. The spacing between these main diagonal lines varies, creating a stepped or staircase-like effect across the entire grid. The overall appearance is that of a digital signal or a complex binary code visualization.

Генерация

- Можем найти средний вектор разности между лицами с улыбкой и без



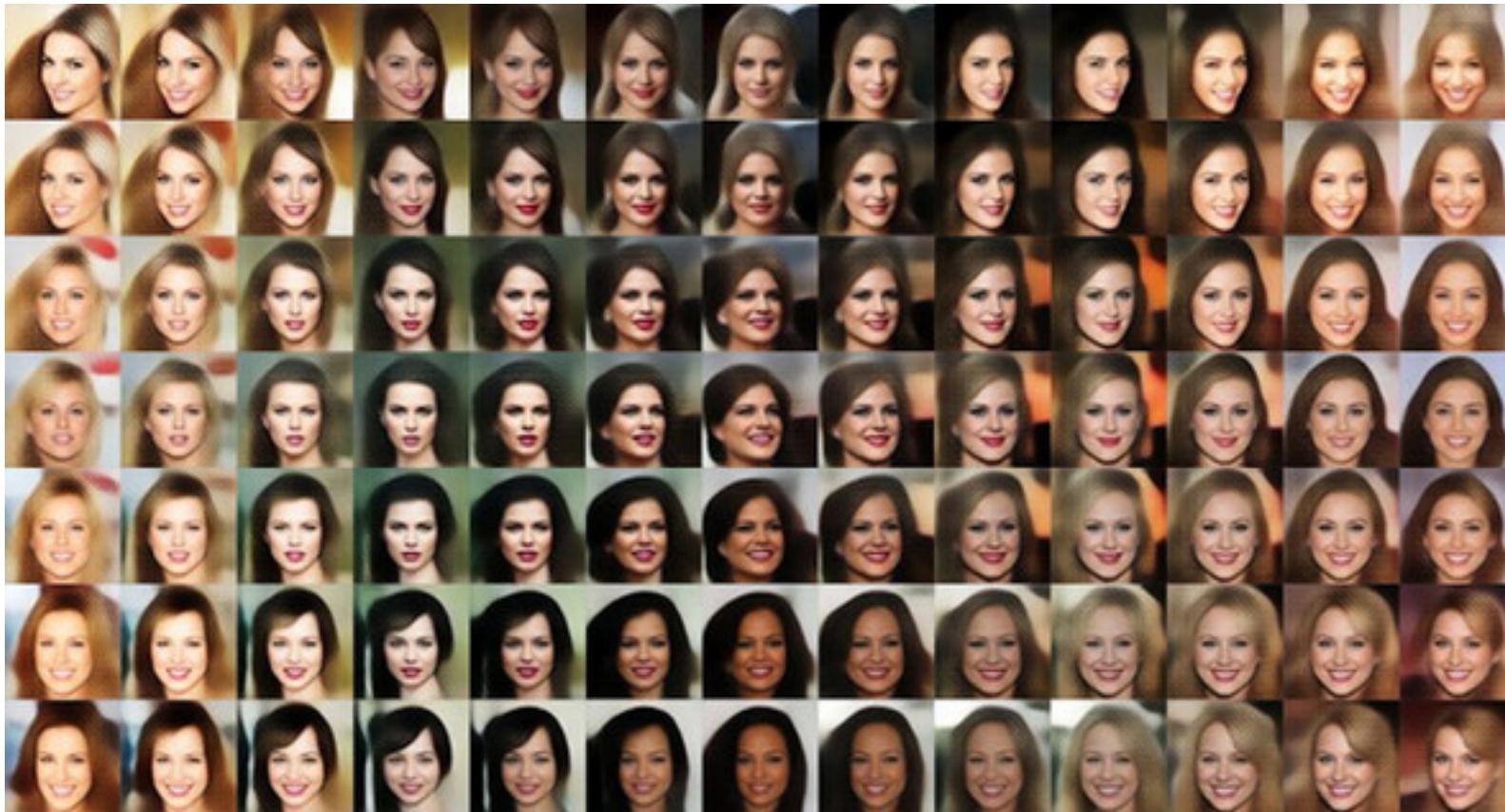
Генерация



Генерация



Генерация



Резюме

- Вариационный автокодировщик пытается построить такое пространство представлений, что каждая точка соответствует какому-то разумному изображению
- Выводится из вероятностных соображений
- Позволяет генерировать изображения (и не только)