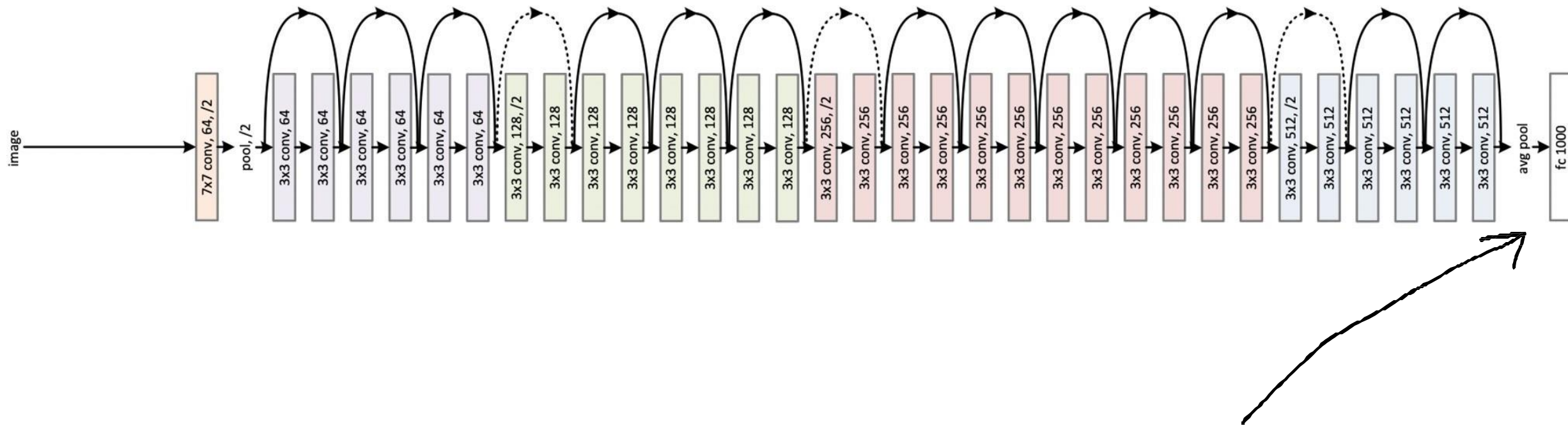


Вариационные автокодировщики

Обучение без учителя

Представления изображений

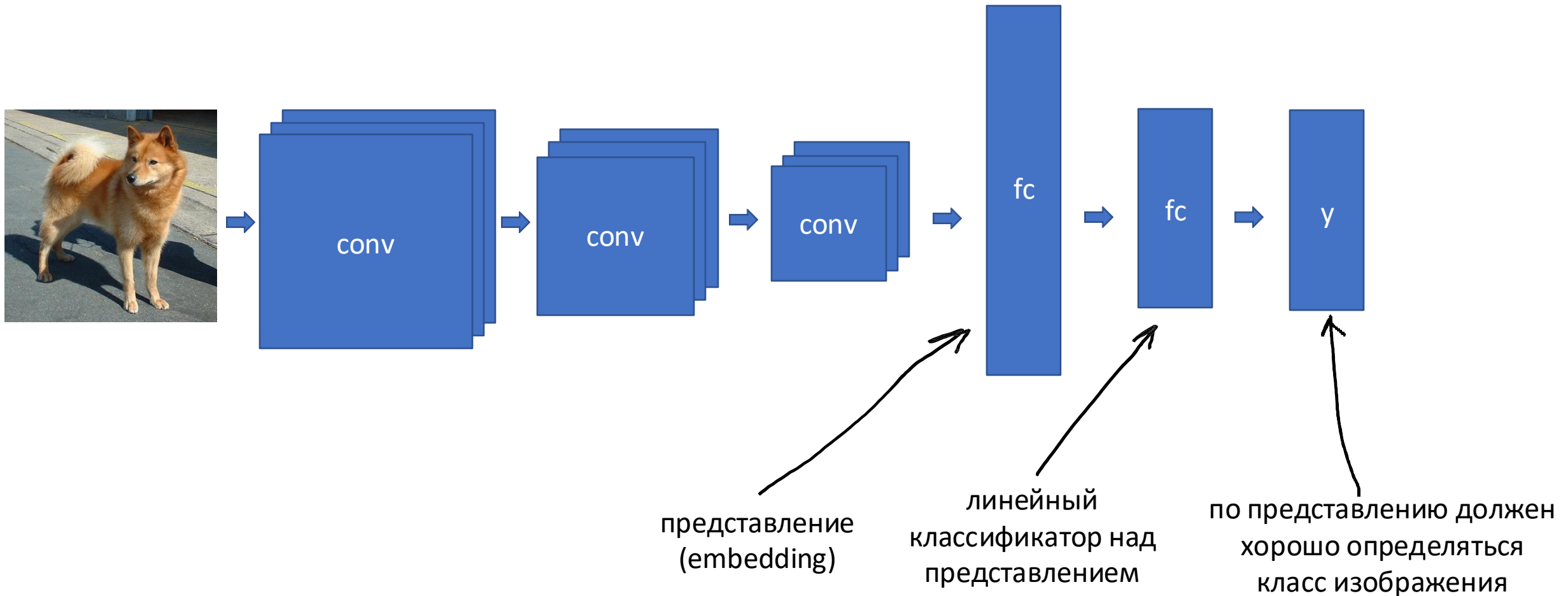


выход предпоследнего полносвязного
слоя — хорошее представления картинки

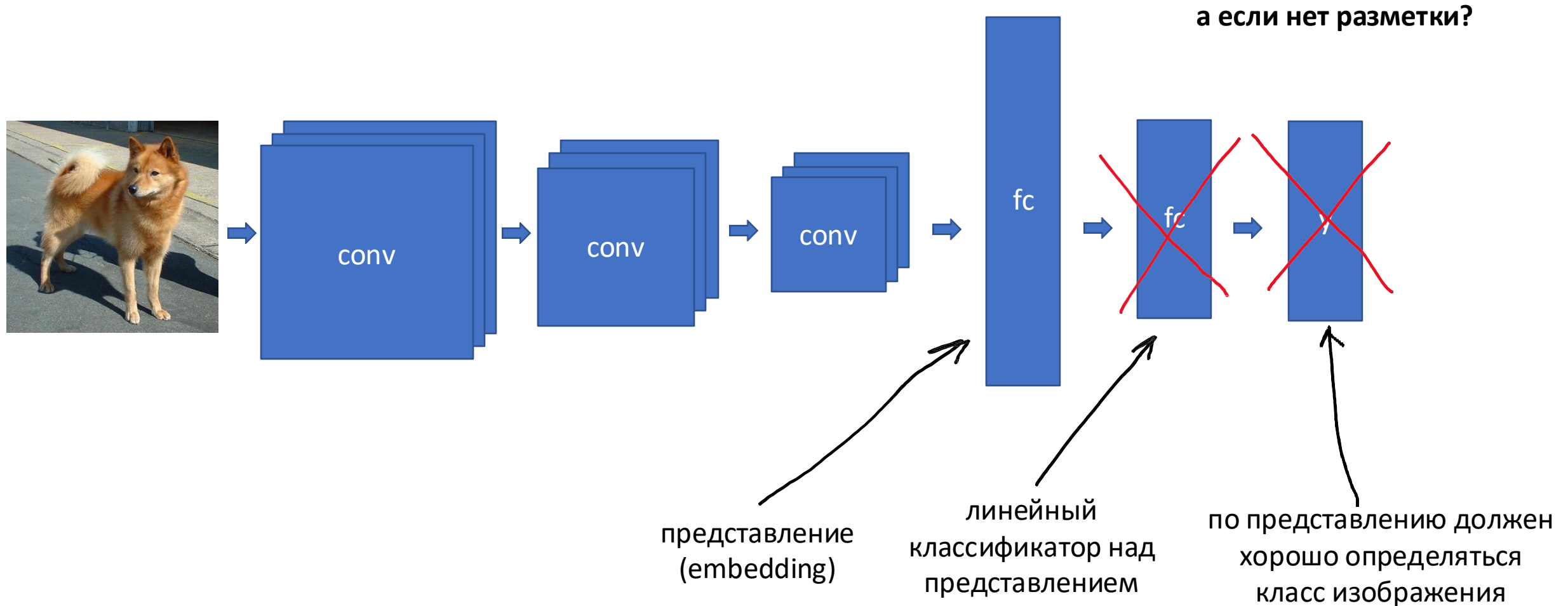
Представления изображений

- Выход предпоследнего полносвязного слоя — хорошее представления картинки
- Но для его обучения нужны изображения с разметкой
- Может, получится строить такие представления и без разметки?

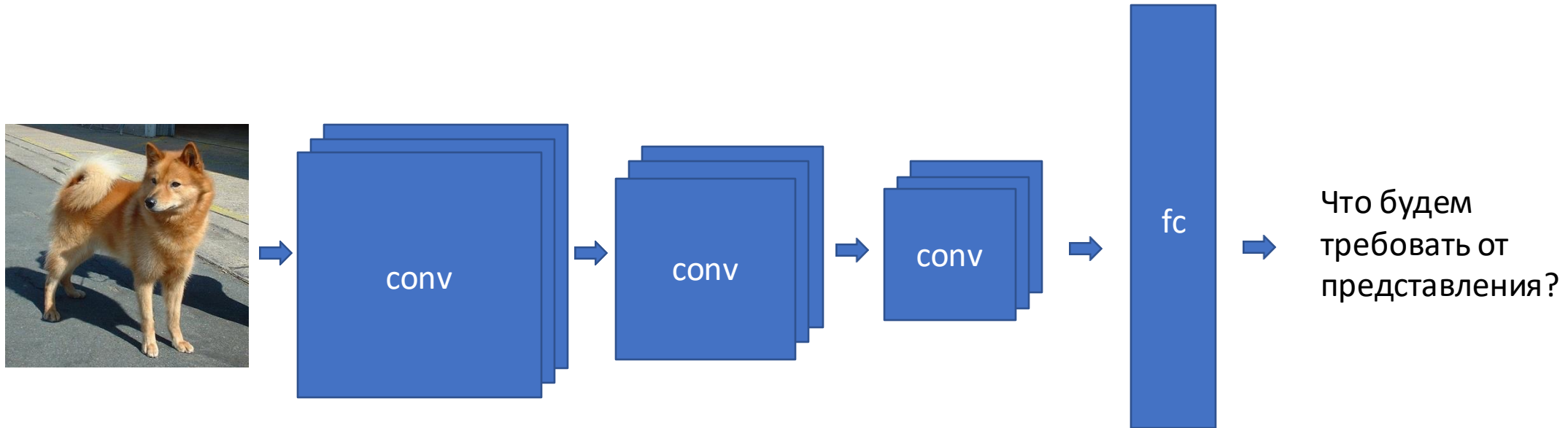
Supervised embeddings



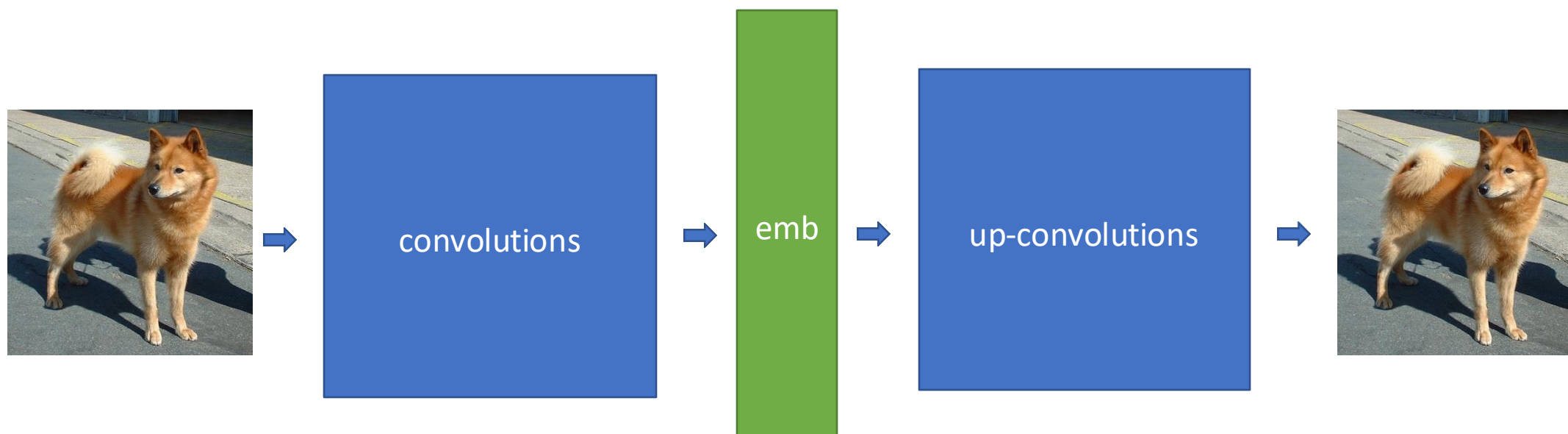
Supervised embeddings



Supervised embeddings

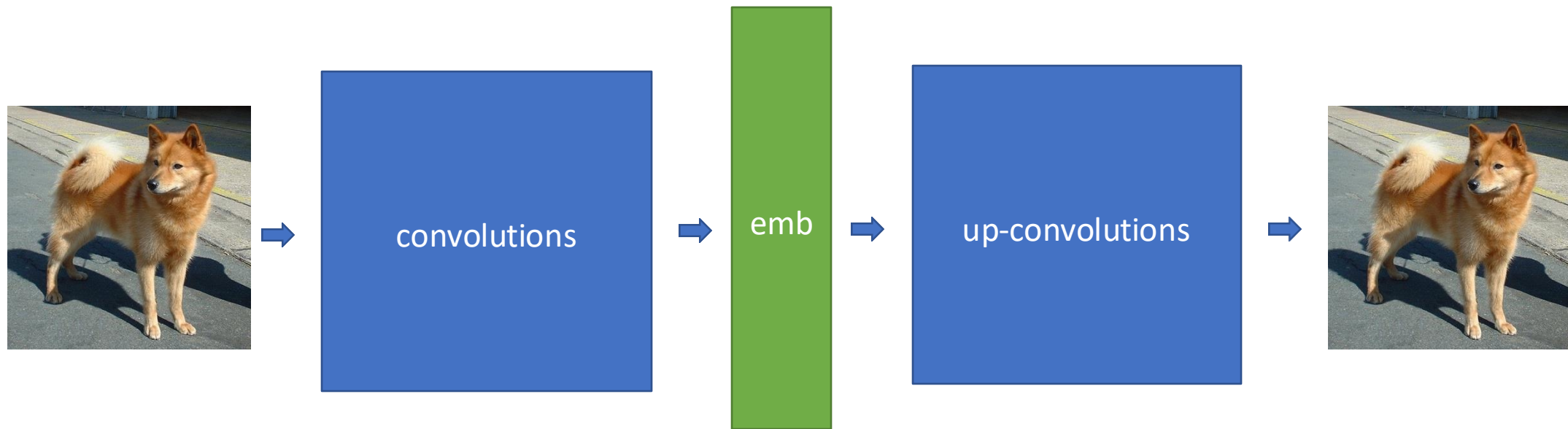


Автокодировщики



обойдёмся без полносвязных слоёв

Автокодировщики



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

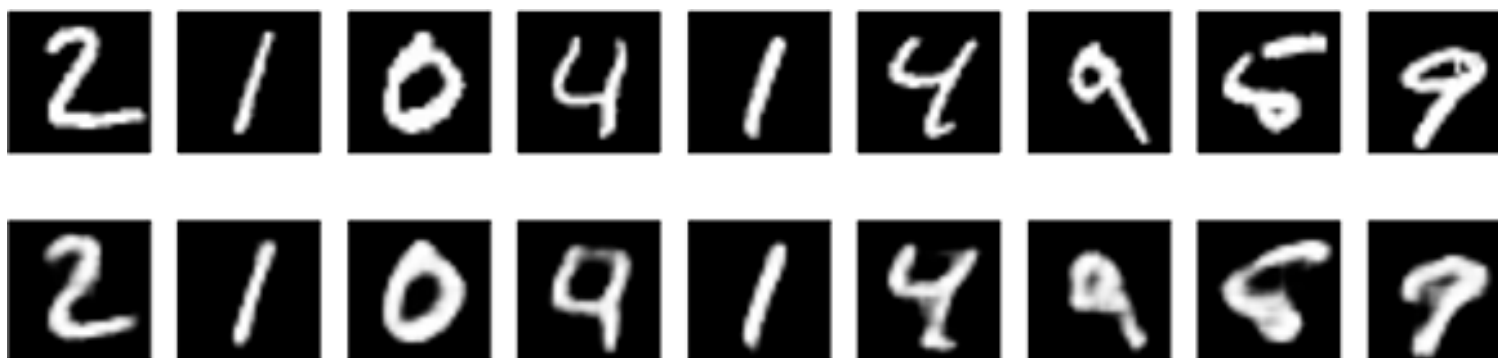
x_i — изображение

$f(x)$ — кодировщик (encoder)

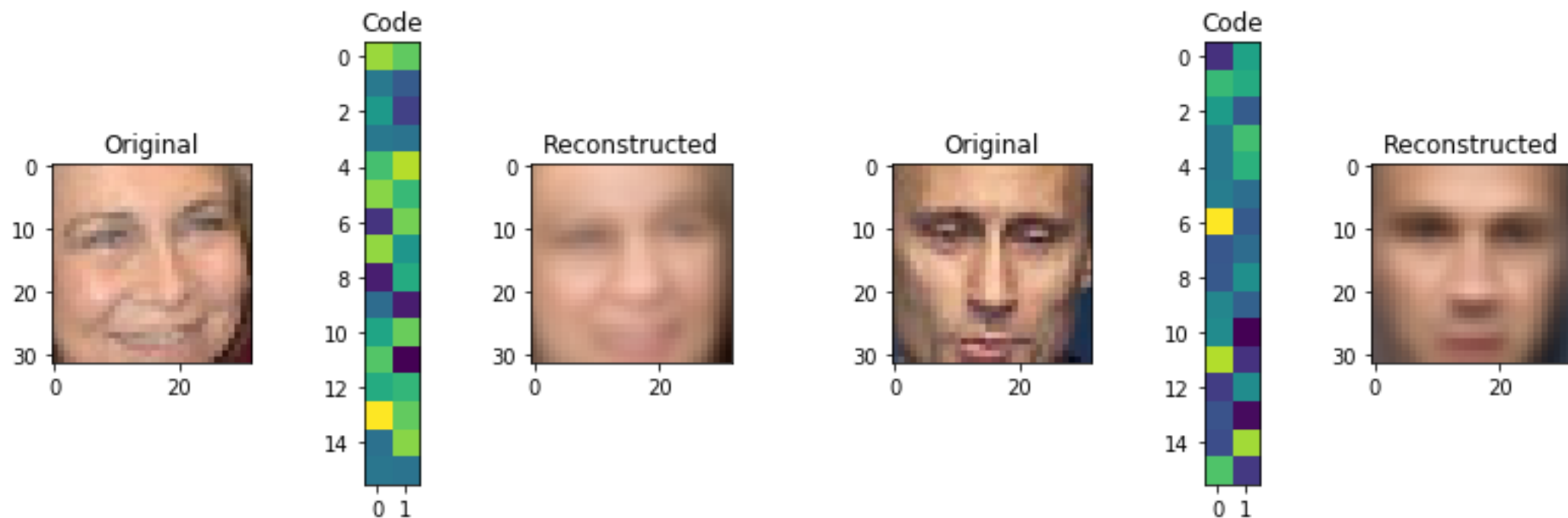
$g(z)$ — декодировщик (decoder)

$L(x, \hat{x})$ — расстояние между изображениями (например, евклидово)

Автокодировщики



Автокодировщики



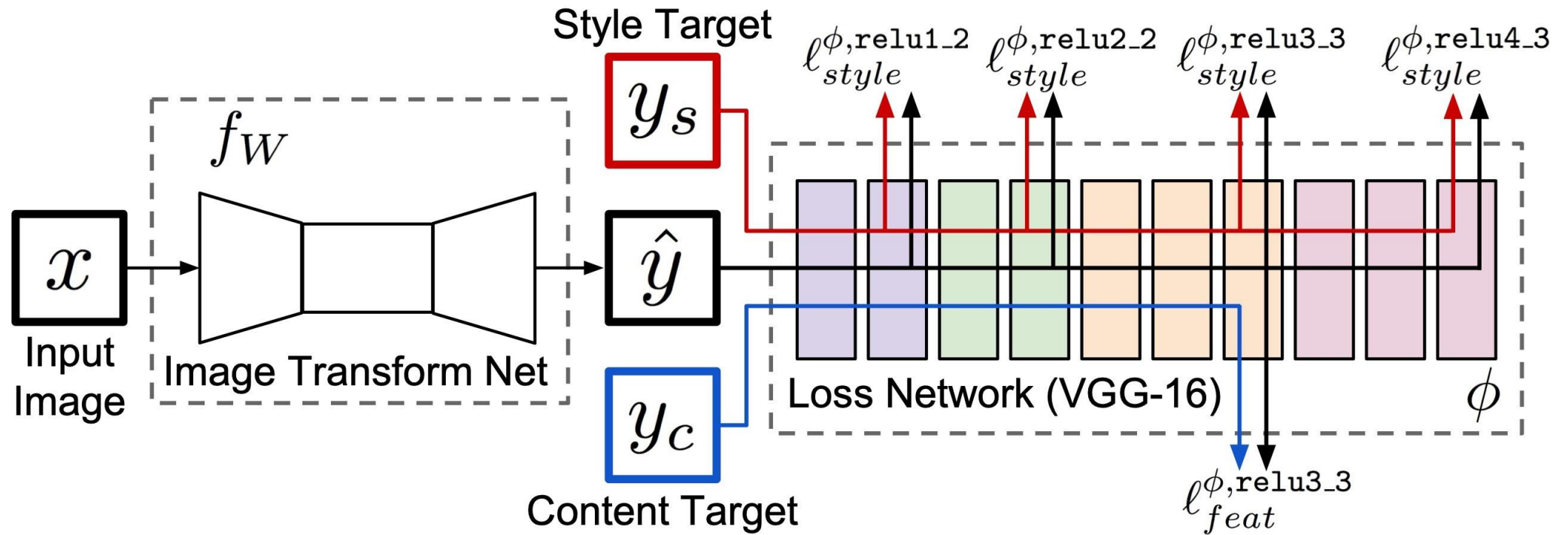
АВТОКОДИРОВЩИКИ

- Восстанавливают изображения с потерями (но это логично)
- Но при этом переобучаются
- Нужно как-то регуляризовать

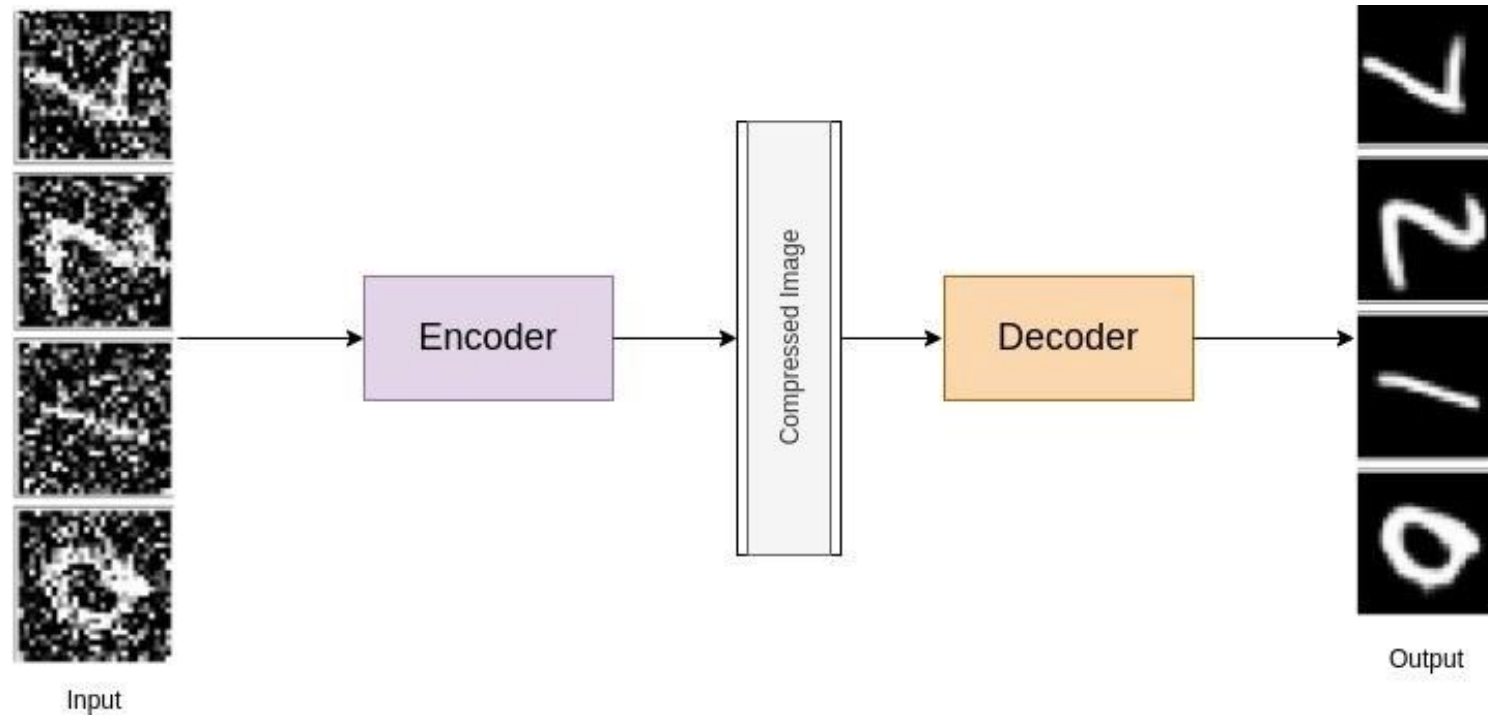
Как ещё измерять сходство картинок?

- Нам важно, чтобы сохранялся смысл, а не в точности восстанавливались пиксели

Perceptual loss (как регуляризация)



Denoising autoencoder



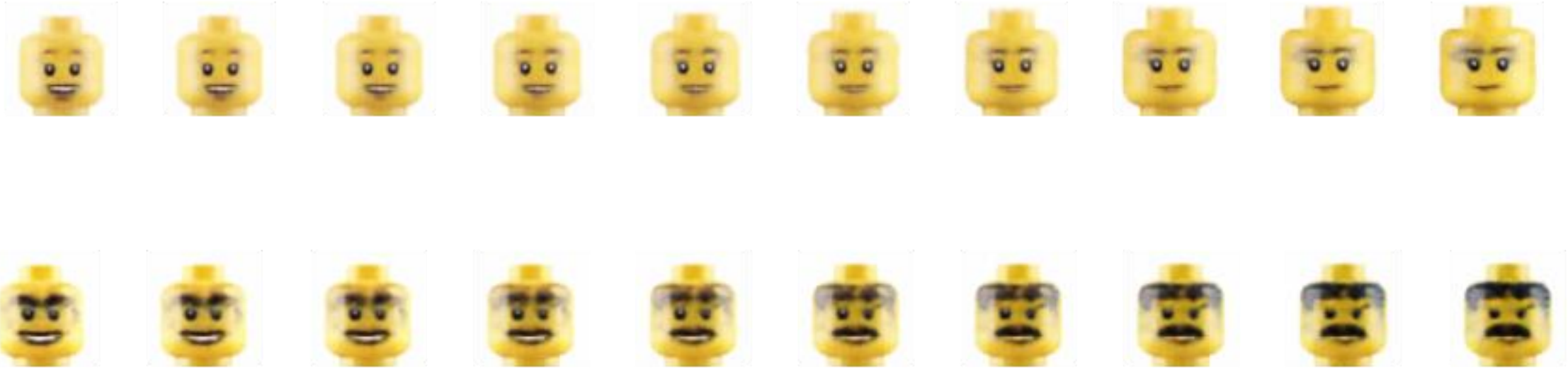
Зачем это всё?

- Сжатие данных (нелинейный аналог PCA)
- Поиск похожих изображений
- Трансформация изображений
- Генерация изображений

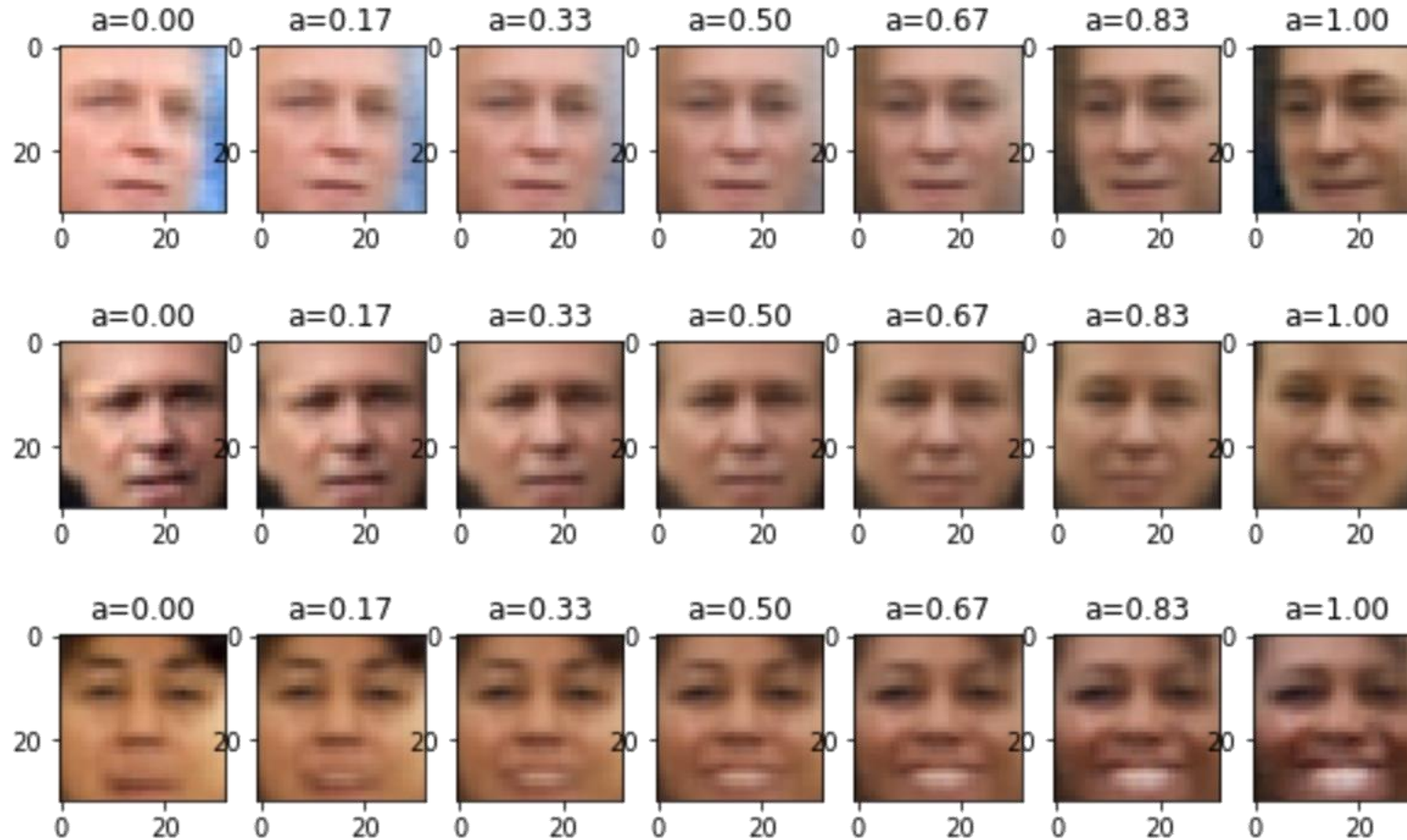
Morphing faces



Morphing faces

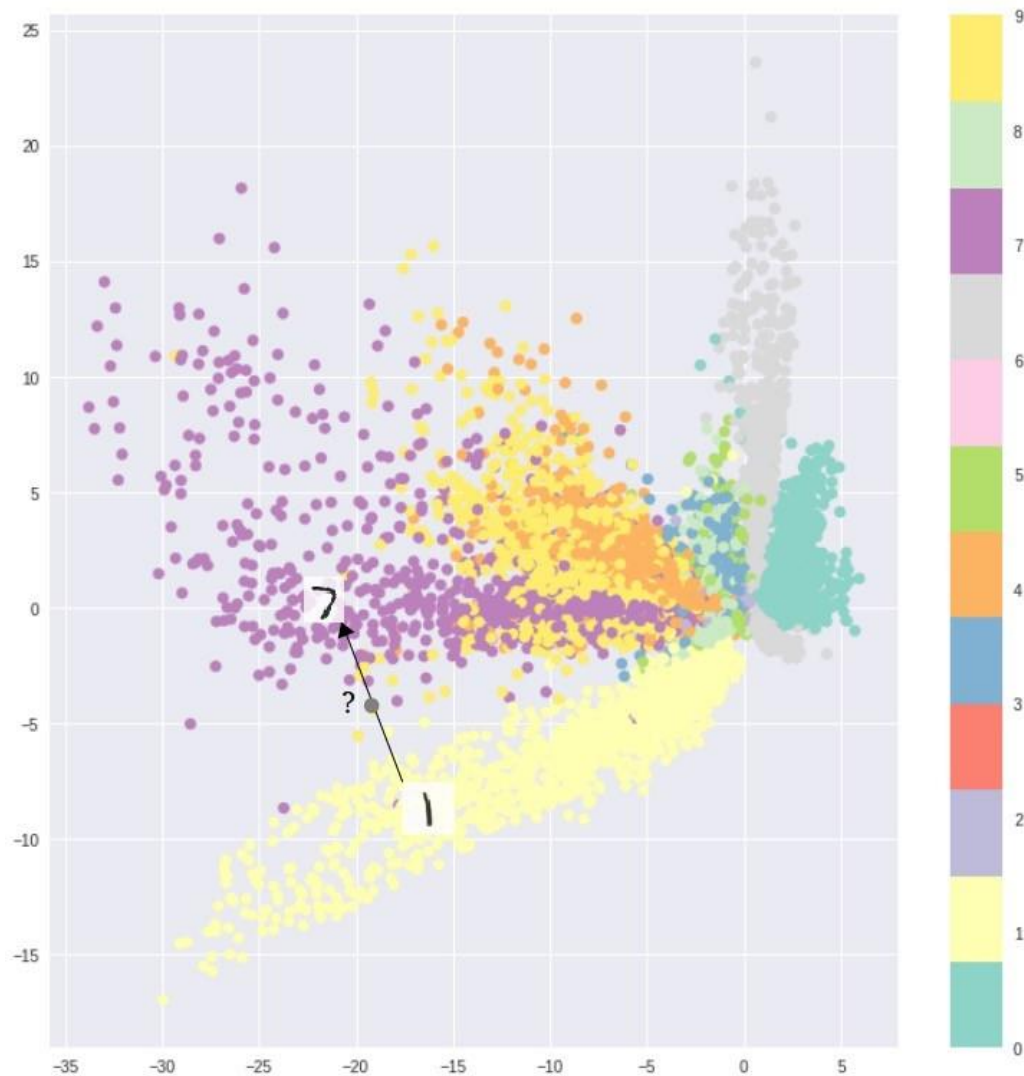


Morphing faces



Генерация изображений

Представления изображений



Цель

- Научиться генерировать изображения
- Поискать вдохновение в вероятностных методах

Вероятностный подход

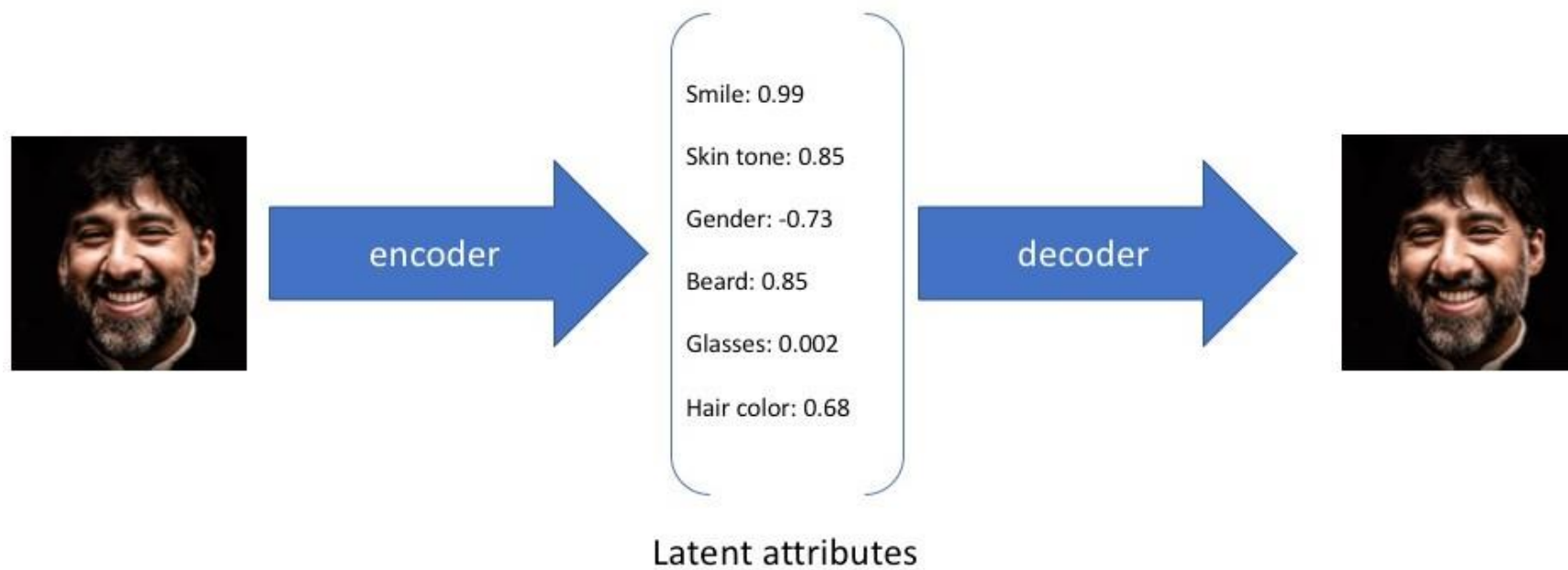
- Описываем, как генерируются ответы
- Подбираем параметры распределений так, чтобы обучающая выборка имела высокую вероятность
- Иногда полезно думать о задаче именно в терминах распределений
- Можно получать оценки неопределенности модели, а не только прогноз

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

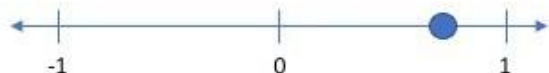
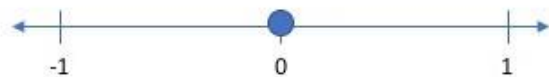
Автокодировщик



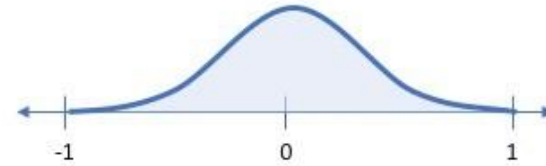
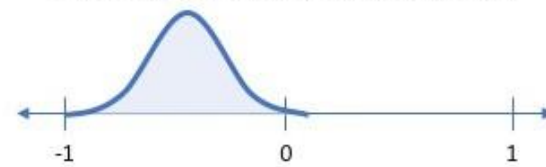
Распределения вместо точечных оценок



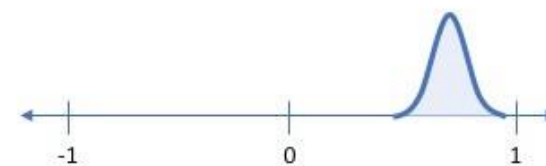
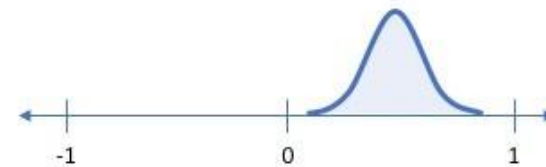
Smile (discrete value)



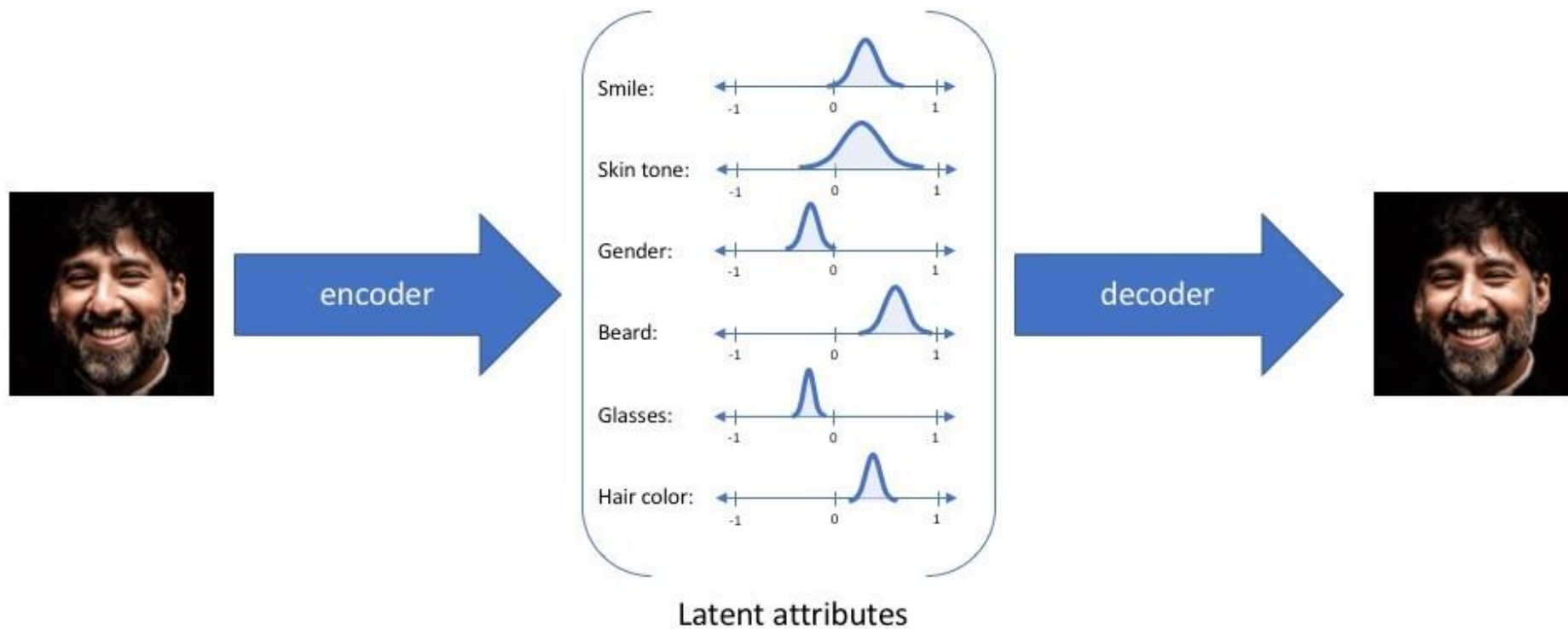
Smile (probability distribution)



vs.



Вероятностные представления



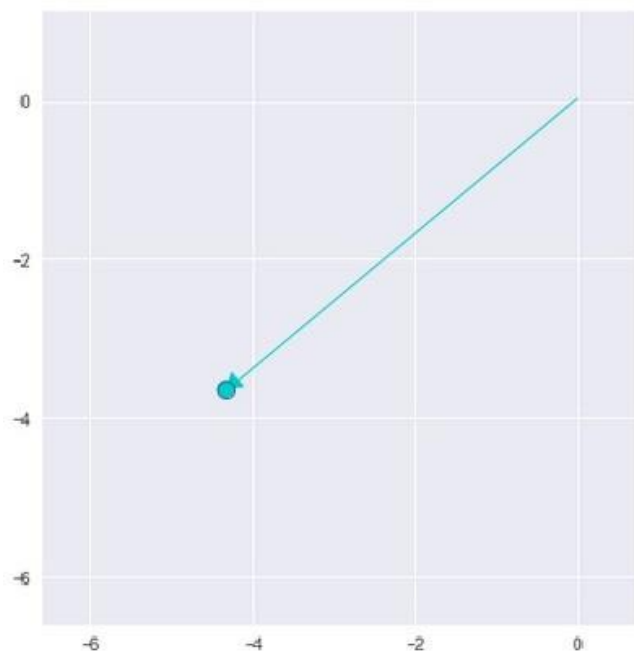
Вероятностные представления

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

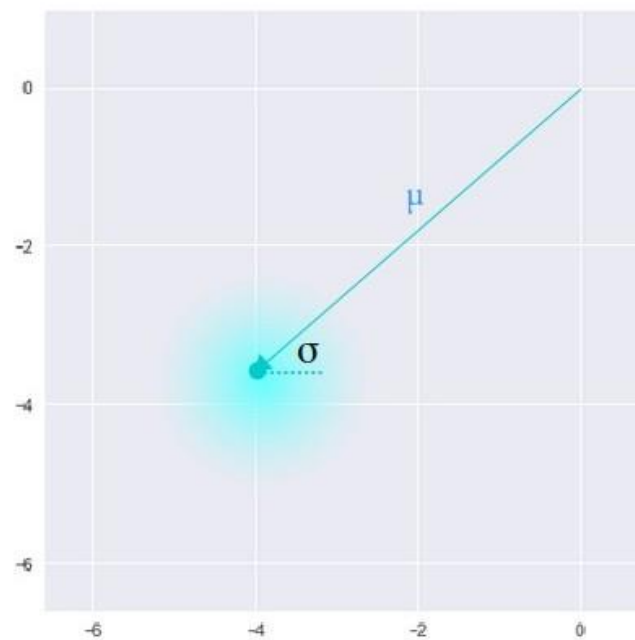
$$\text{encoder}(x) = (\mu(x), \sigma(x))$$

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

Вероятностные представления



Standard Autoencoder
(direct encoding coordinates)



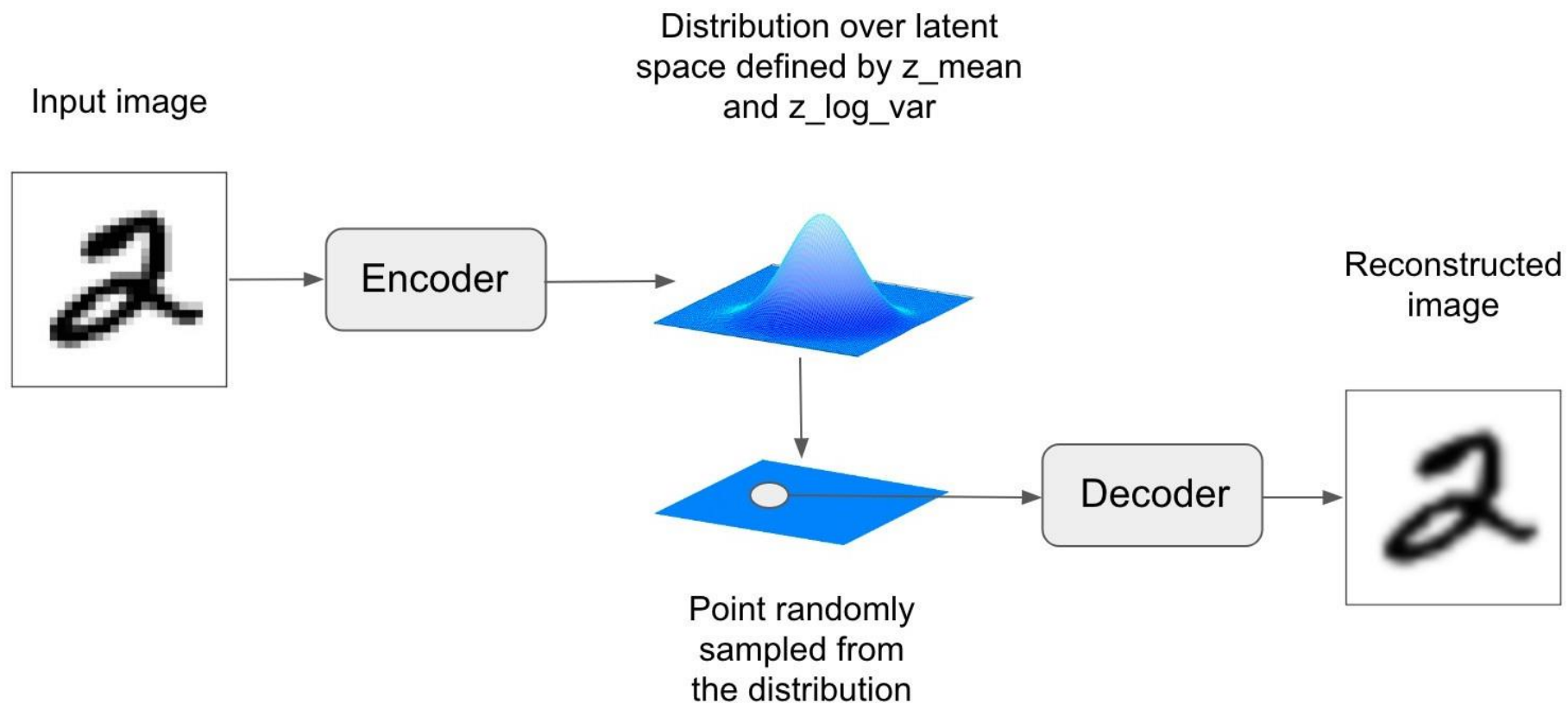
Variational Autoencoder
(μ and σ initialize a probability distribution)

Вероятностные представления

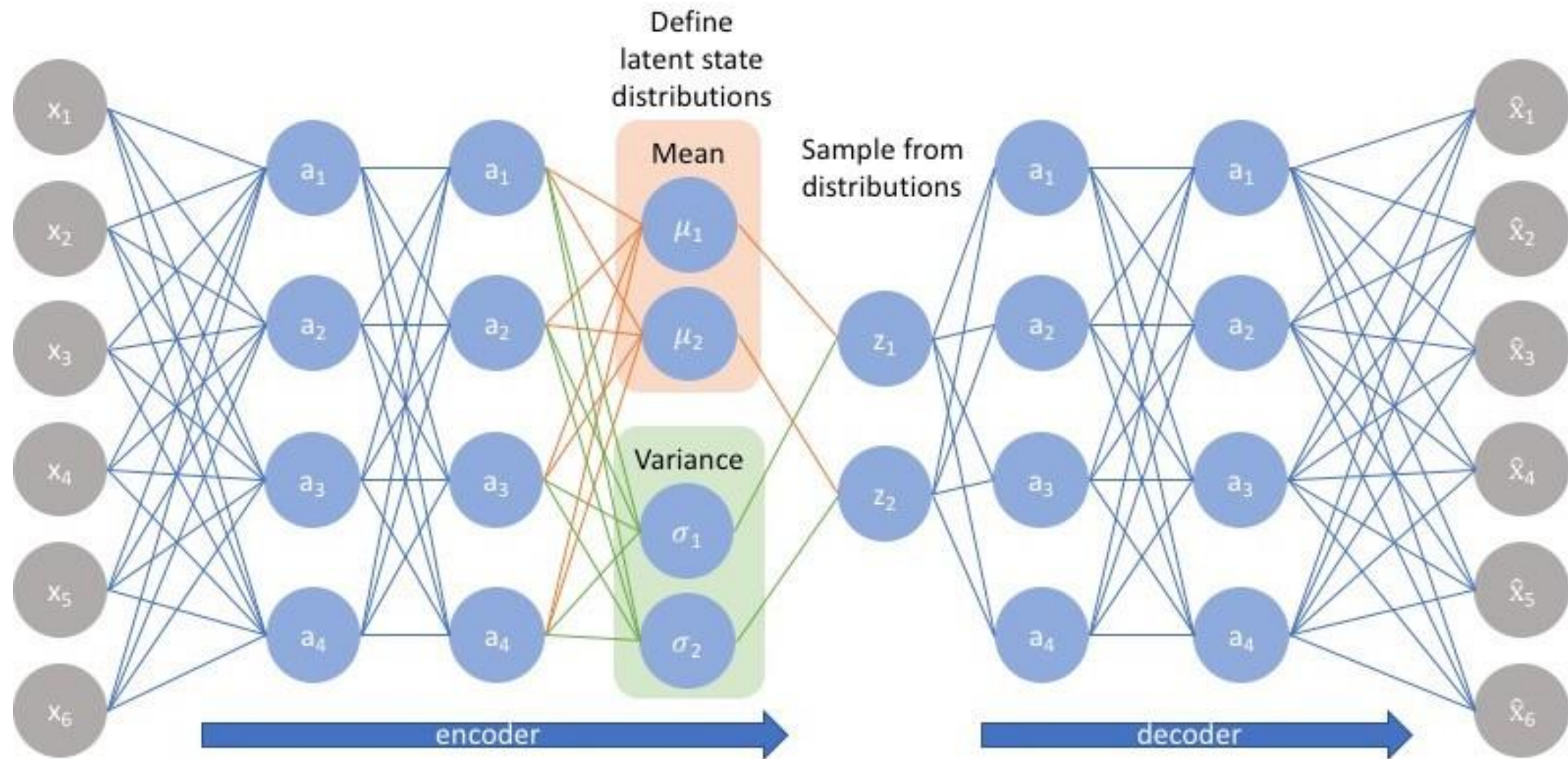
$$\text{encoder}(x) = (\mu(x), \sigma(x))$$

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

Вероятностные представления



Вероятностные представления



Вариационный автокодировщик

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

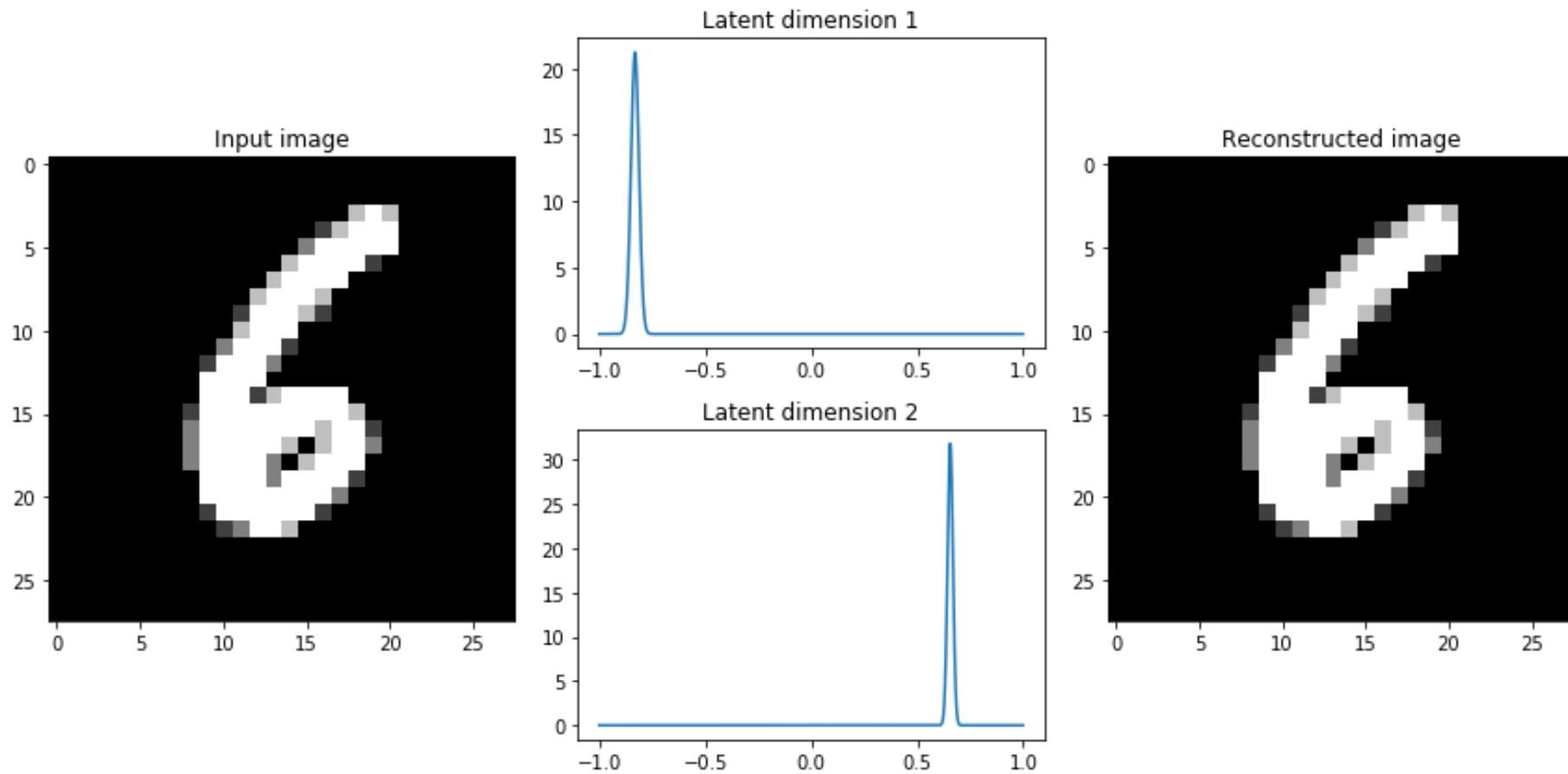
- $q(z|x)$ — кодировщик (полносвязная или свёрточная нейросеть, z — возможный эмбединг для x)
- $p(x|z) = \text{decoder}(z) + \varepsilon$ — декодировщик (полносвязная или свёрточная нейросеть)
- $\mathbb{E}_{q(z|x_i)} \log p(x_i|z)$ — как бы закодировали x_i , сгенерировали все возможные представления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

Вариационный автокодировщик

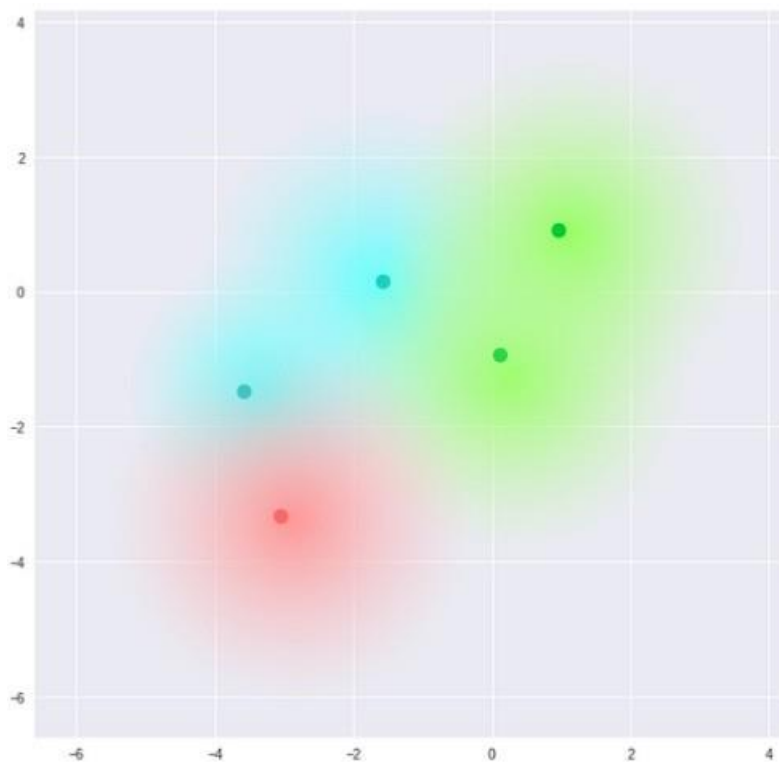
$$\sum_{i=1}^{\ell} \left(\mathbb{E}_{q(z|x_i)} \log p(x_i|z) - \text{KL}(q(z|x) \parallel \mathcal{N}(0, 1)) \right) \rightarrow \max$$

- $\text{KL}(q(z) \parallel p(z)) = \int_{-\infty}^{+\infty} q(z) \log \frac{q(z)}{p(z)} dz$ — 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

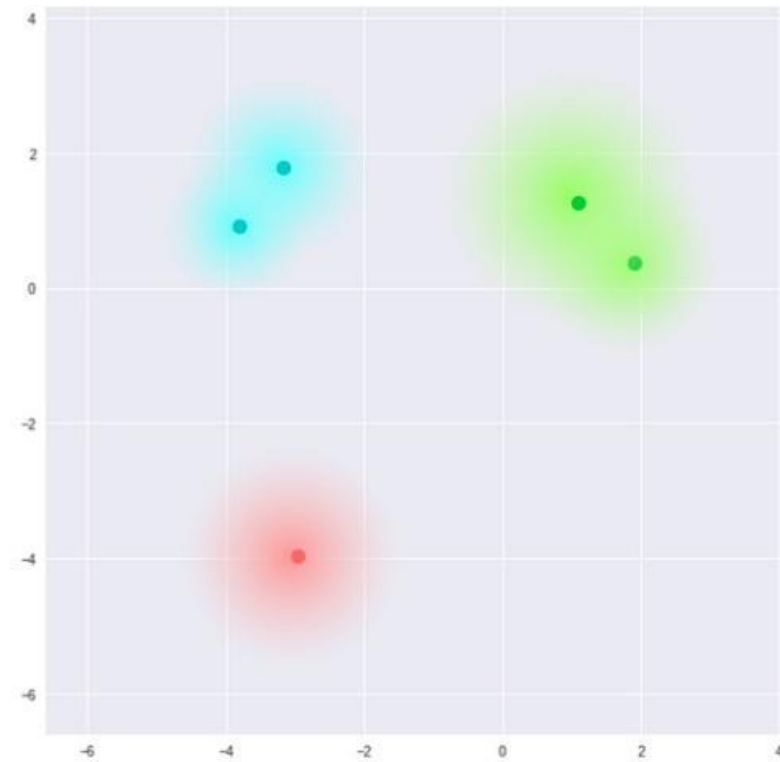
Не хотим так



Непрерывность пространства представлений



What we require



What we may inadvertently end up with

Вариационный автокодировщик

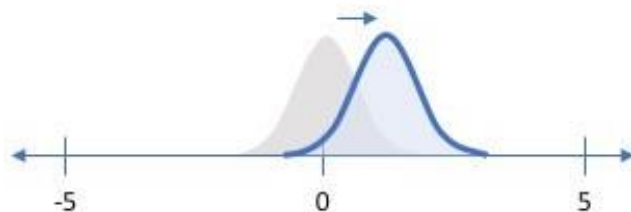
$$\sum_{i=1}^{\ell} \left(\mathbb{E}_{q(z|x_i)} \log p(x_i|z) - \text{KL}(q(z|x) \parallel \mathcal{N}(0, 1)) \right) \rightarrow \max$$

«функция потерь»
reconstruction likelihood

«регуляризатор»

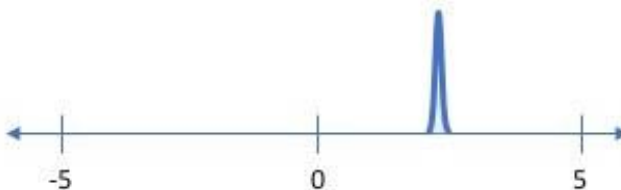
Вариационный автокодировщик

Penalizing reconstruction loss encourages the distribution to describe the input



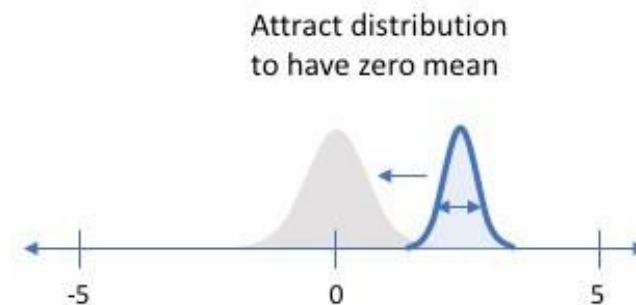
Our distribution deviates from the prior to describe some characteristic of the data

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



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

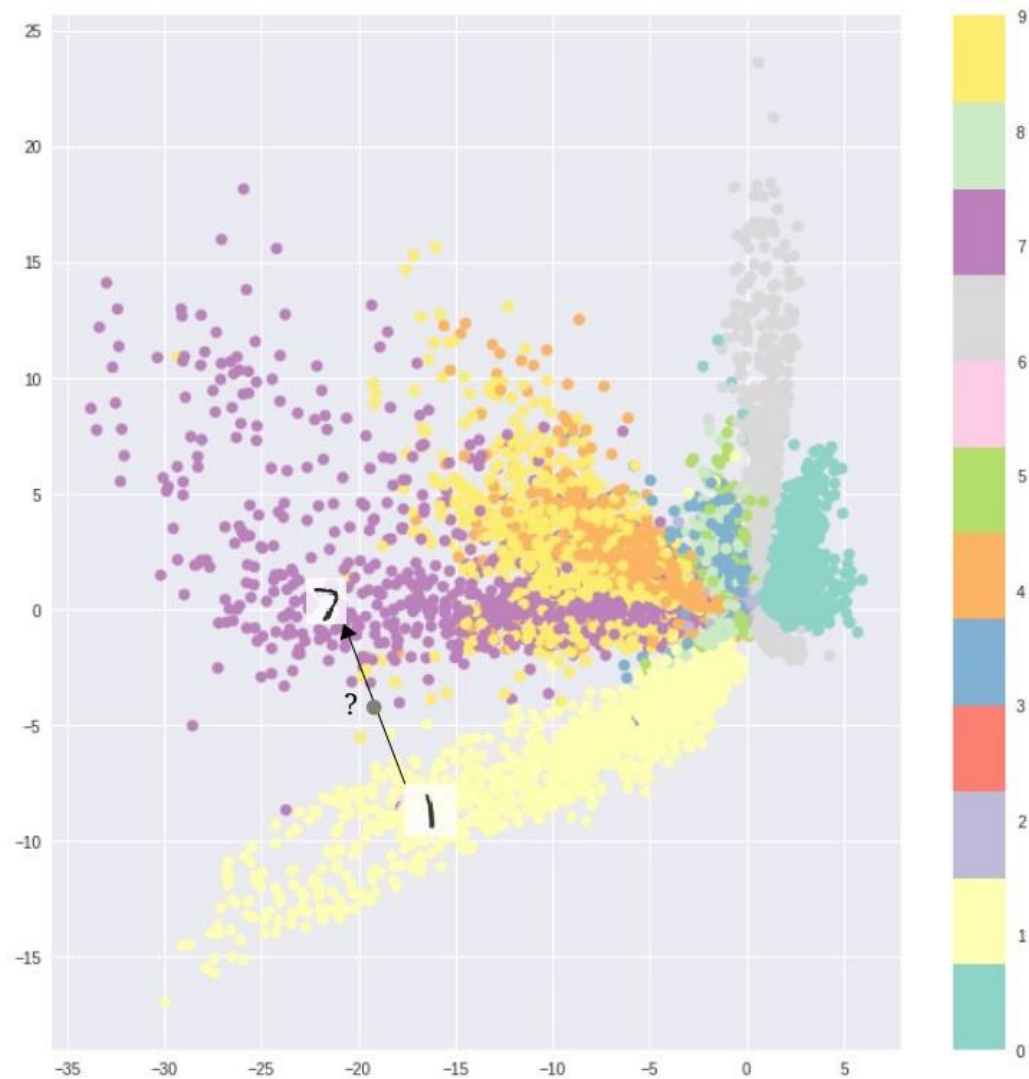
Penalizing KL divergence acts as a regularizing force



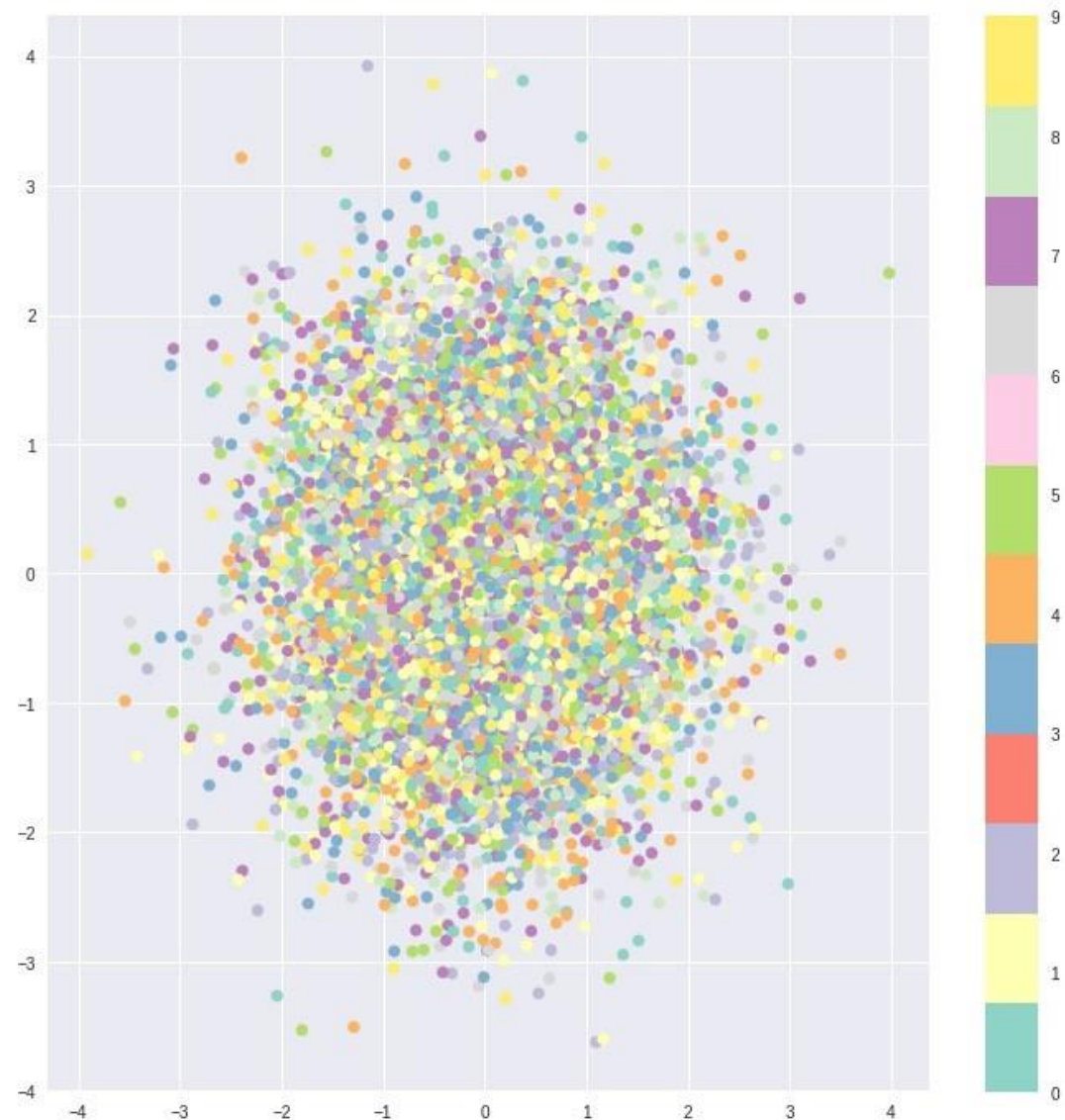
Attract distribution to have zero mean

Ensure sufficient variance to yield a smooth latent space

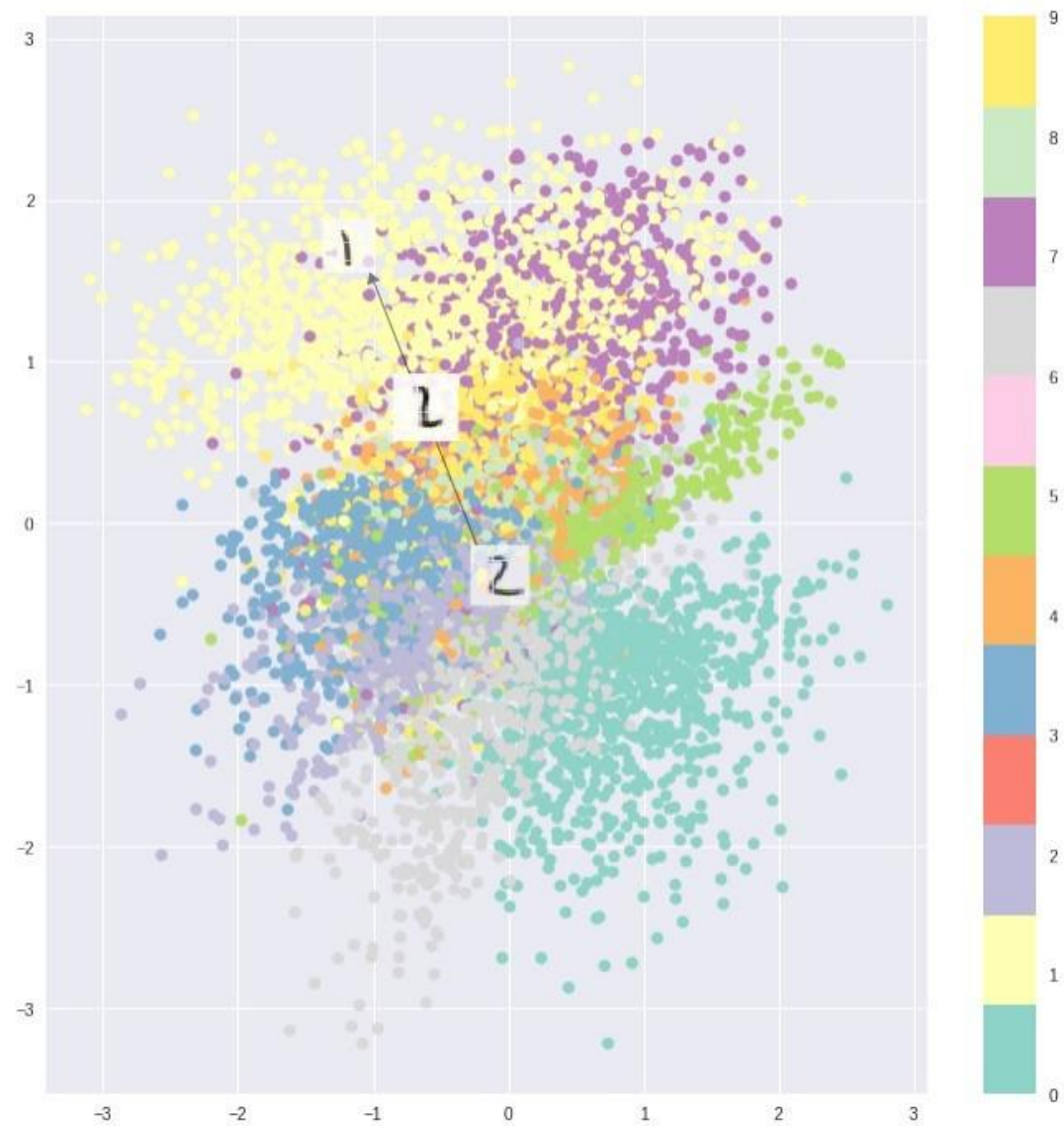
Только reconstruction likelihood



Только KL-дивергенция

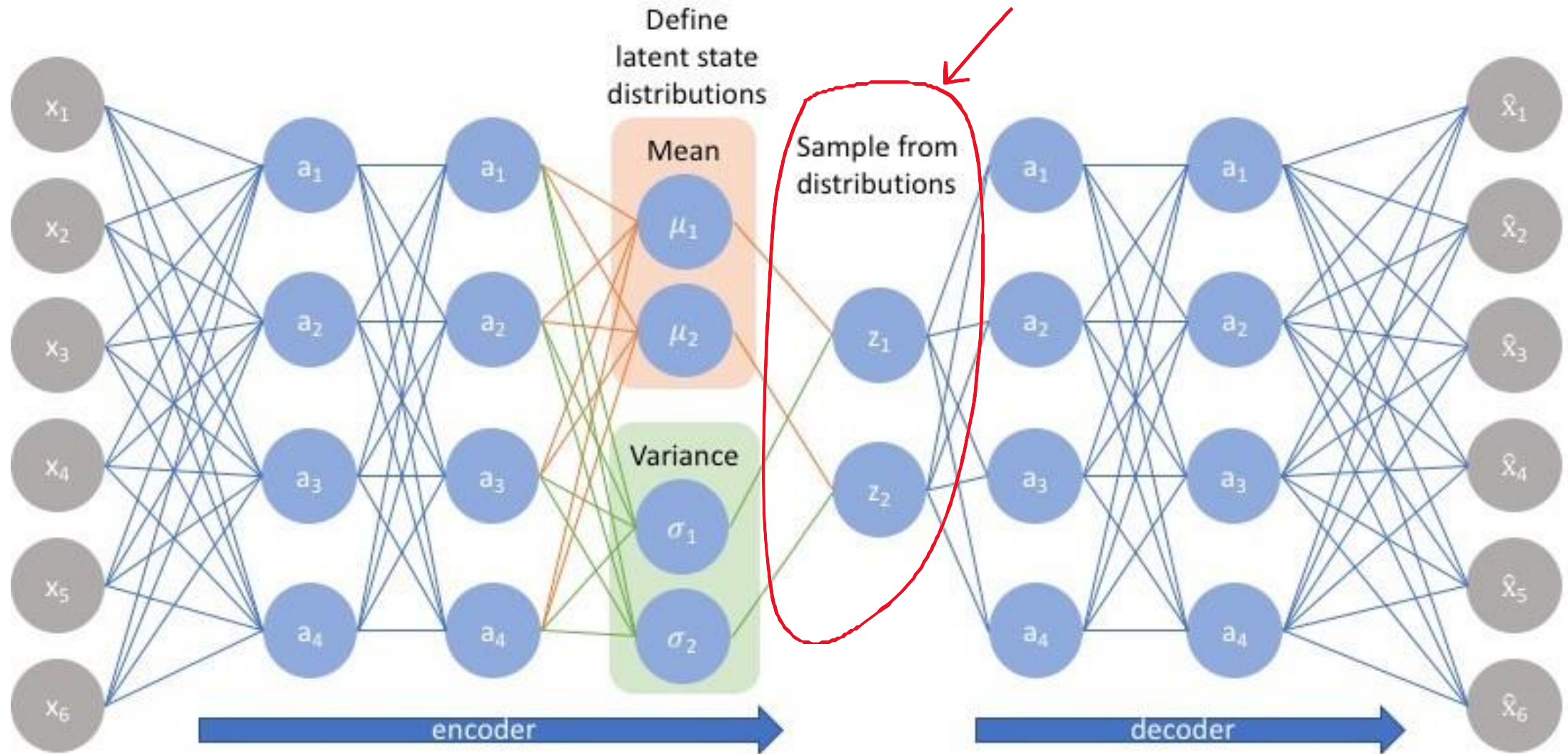


Всё вместе

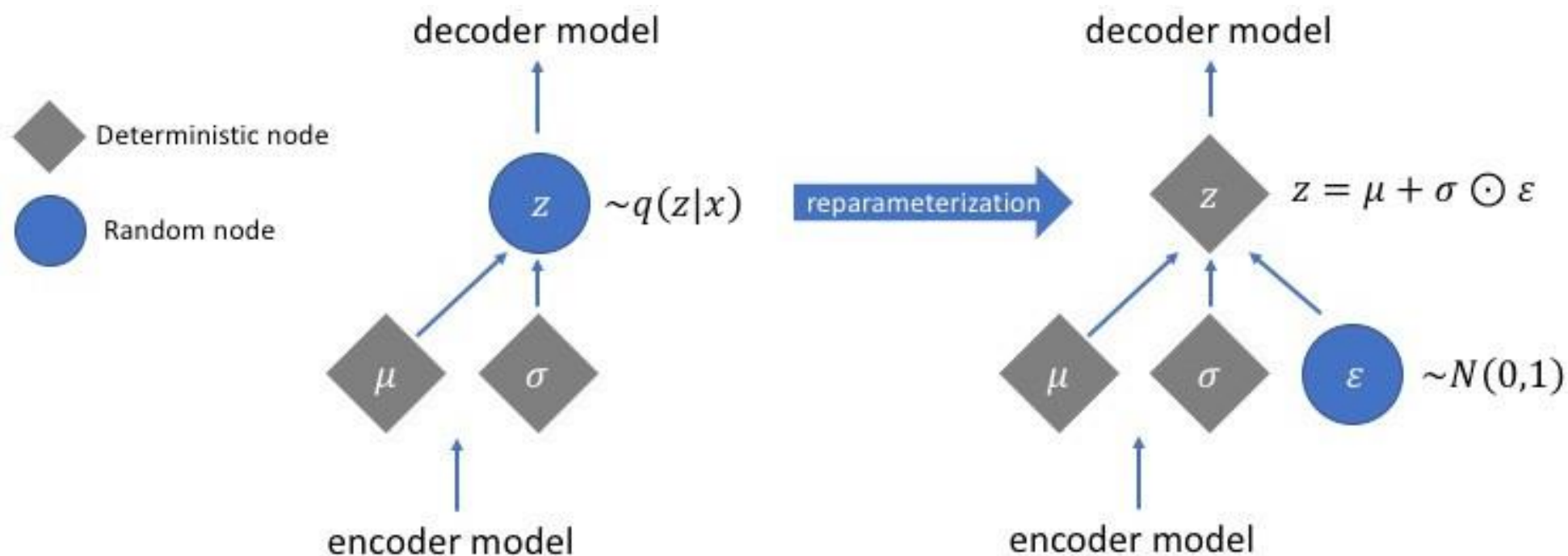


Оптимизация в VAE

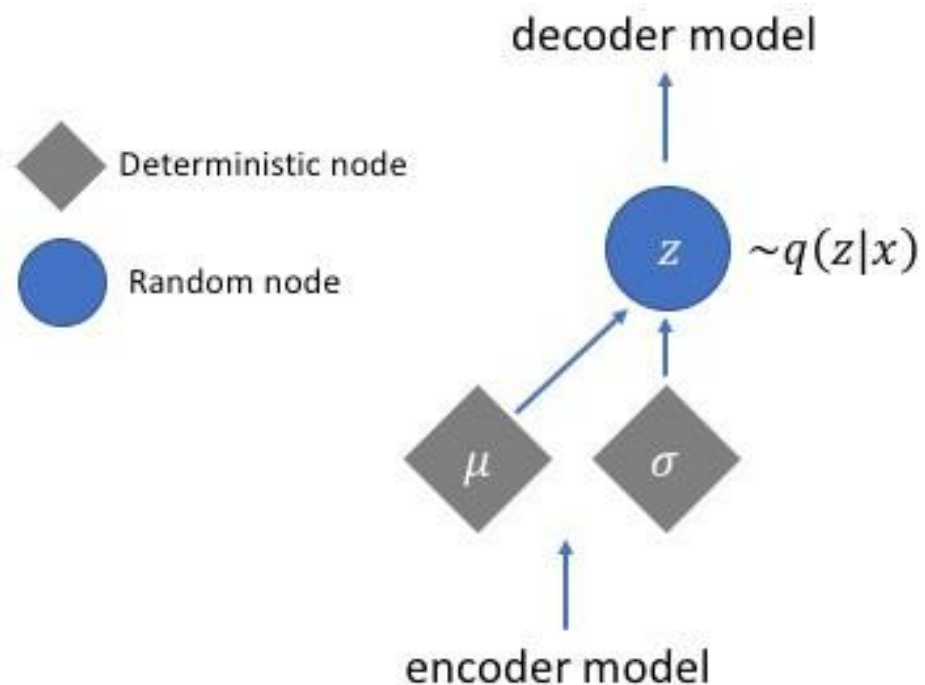
Какая-то вероятностная штука,
как считать градиенты?



Оптимизация в VAE

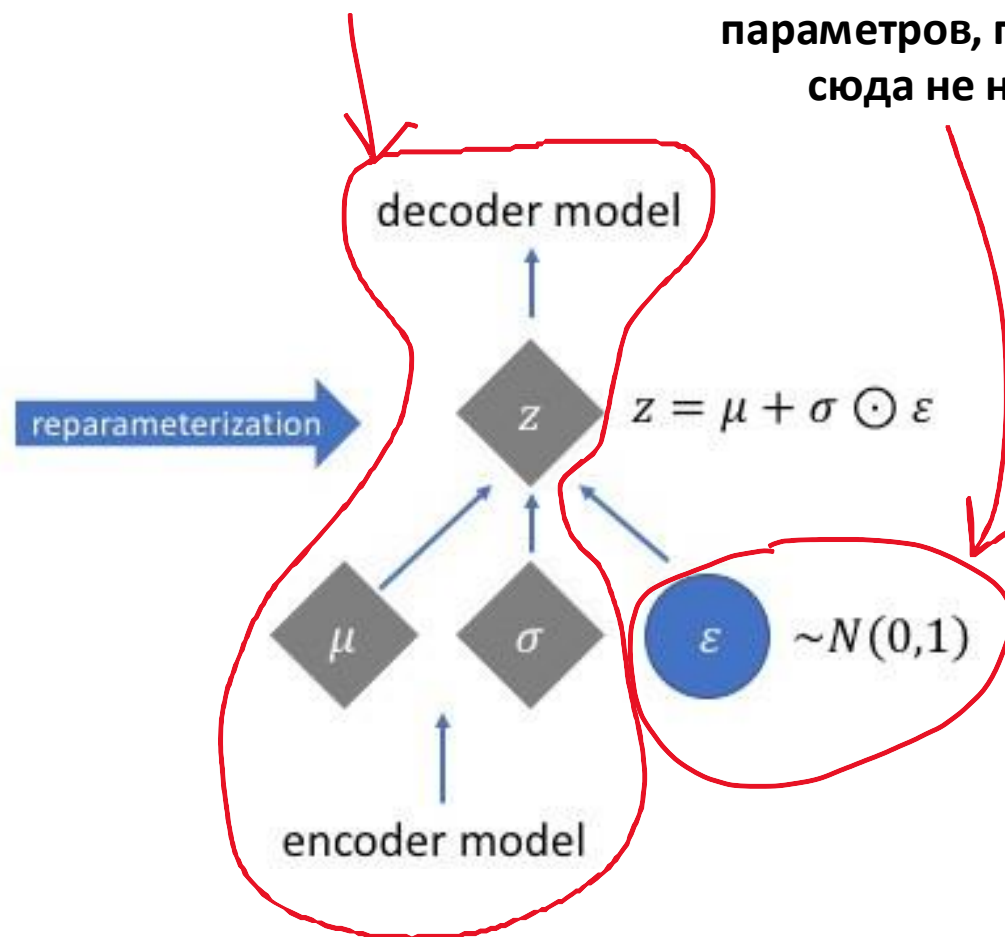


Оптимизация в VAE

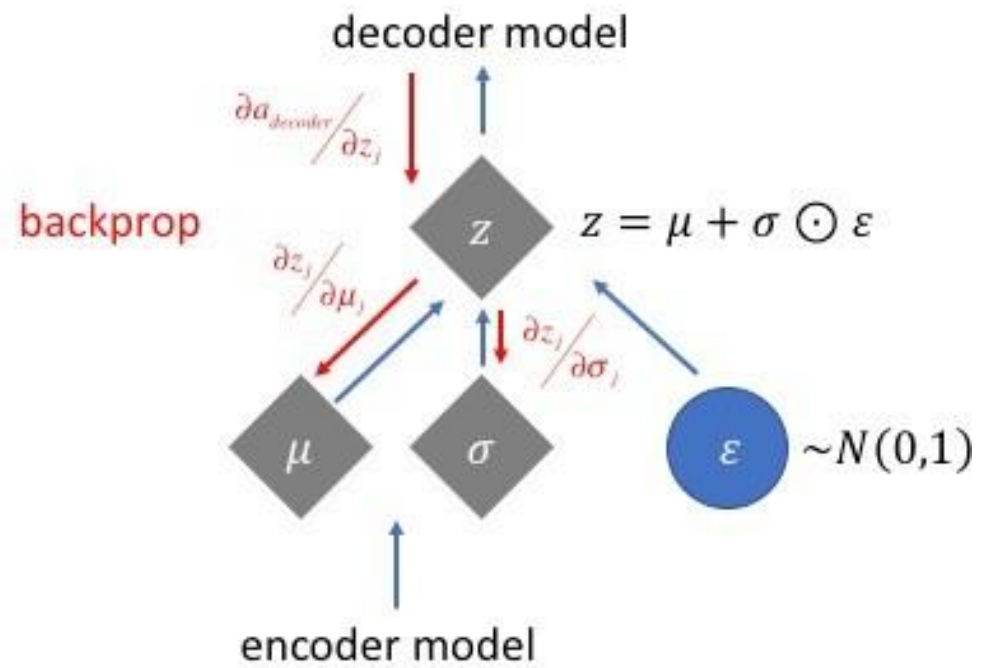


Никаких случайностей,
можем считать градиенты

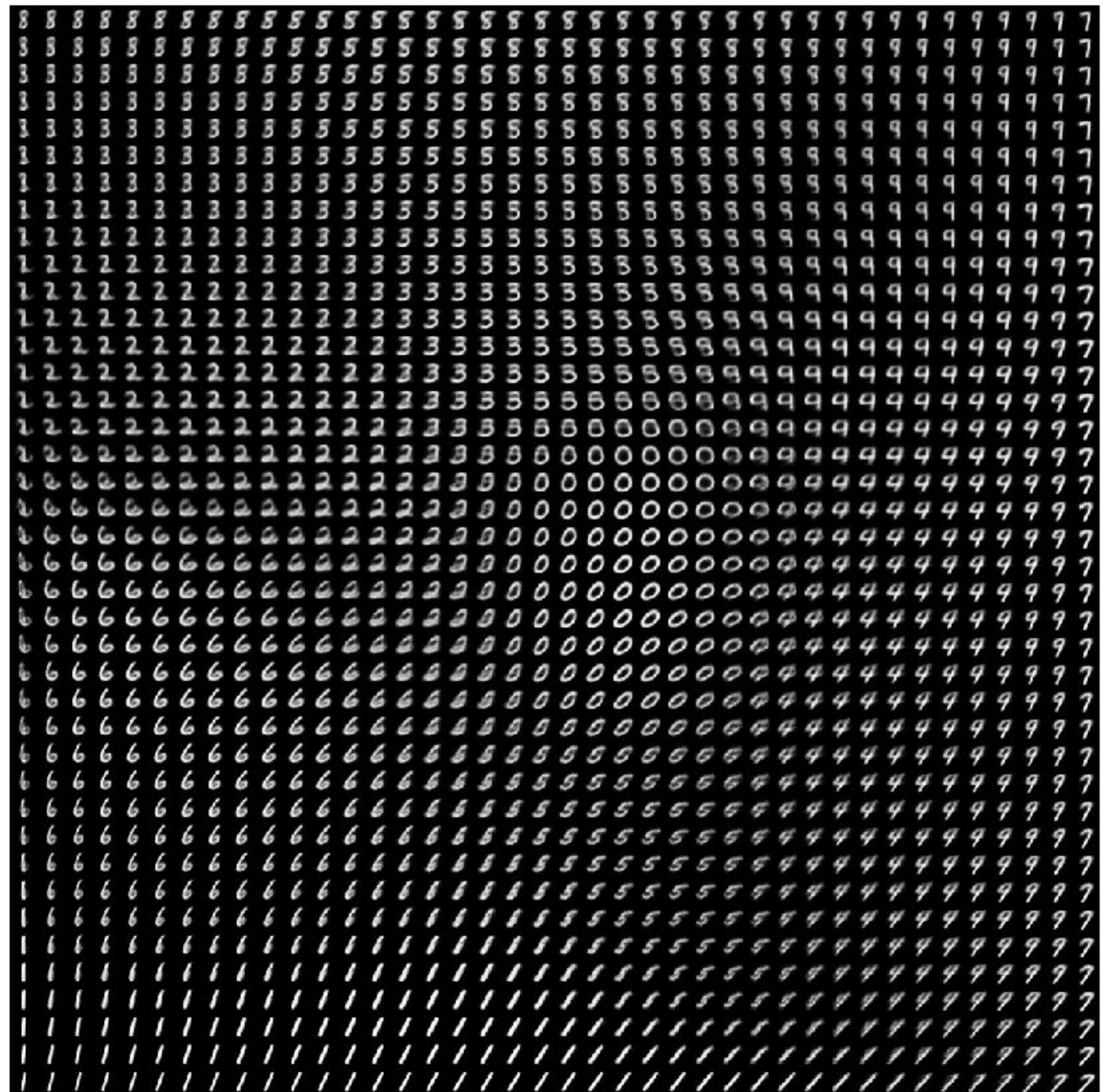
Случайность, но без
параметров, градиенты
сюда не нужны



Оптимизация в VAE

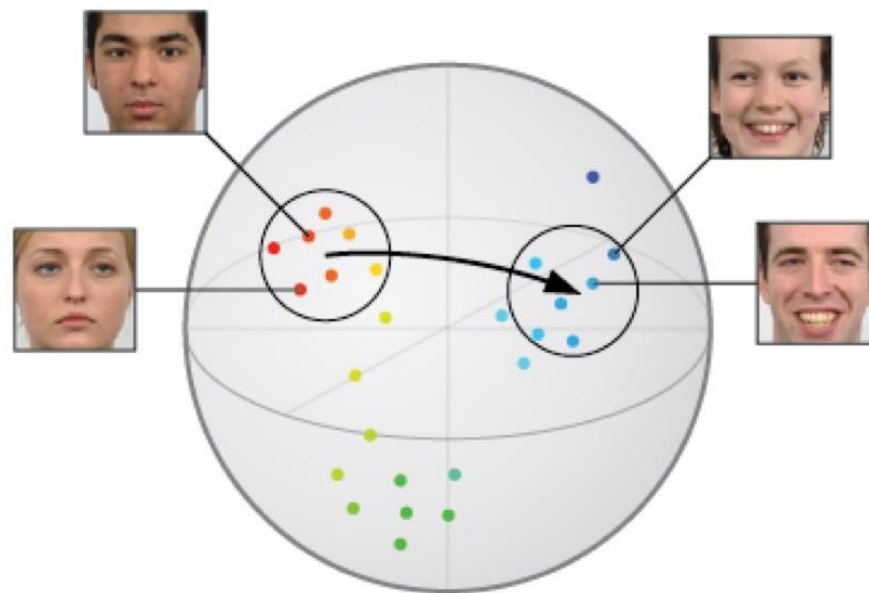


Генерация

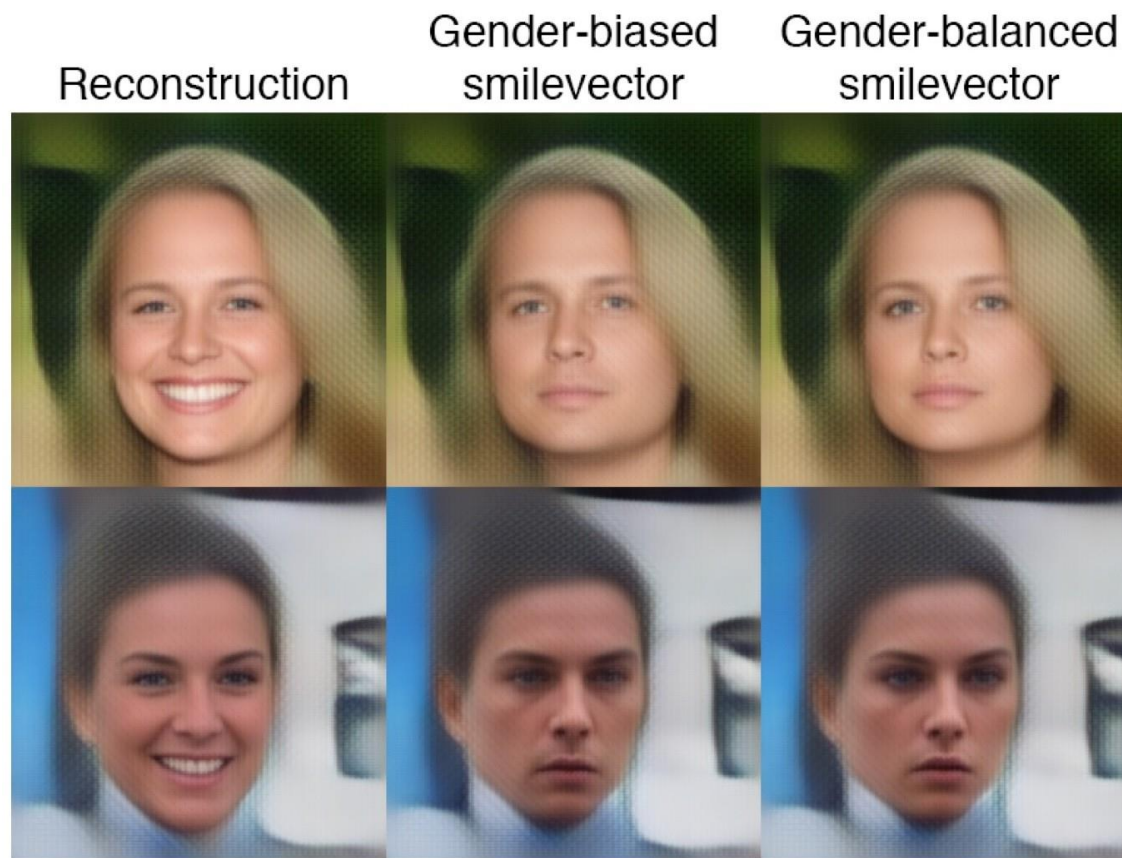


Генерация

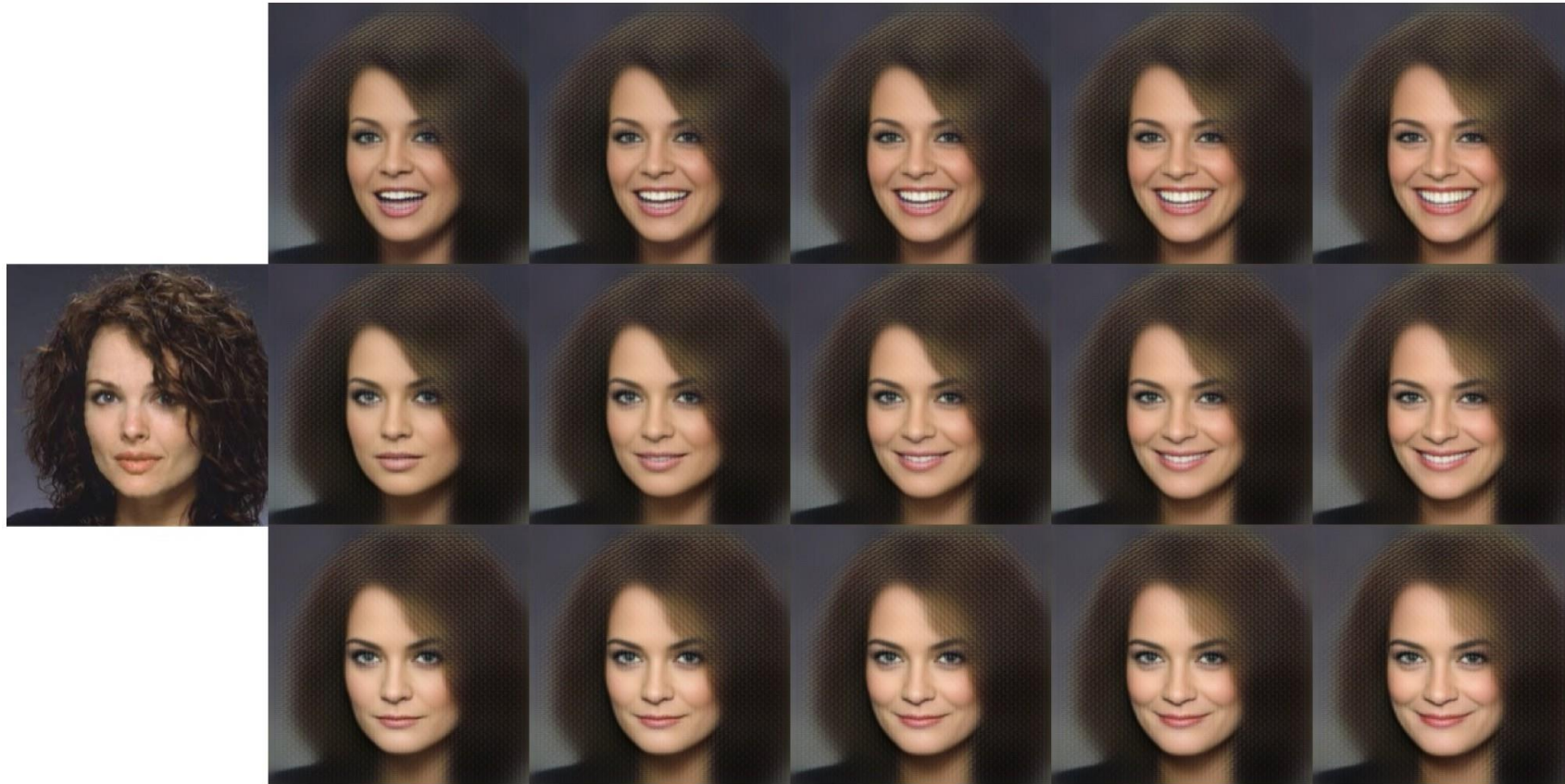
- Можем найти средний вектор разности между лицами с улыбкой и без



Генерация



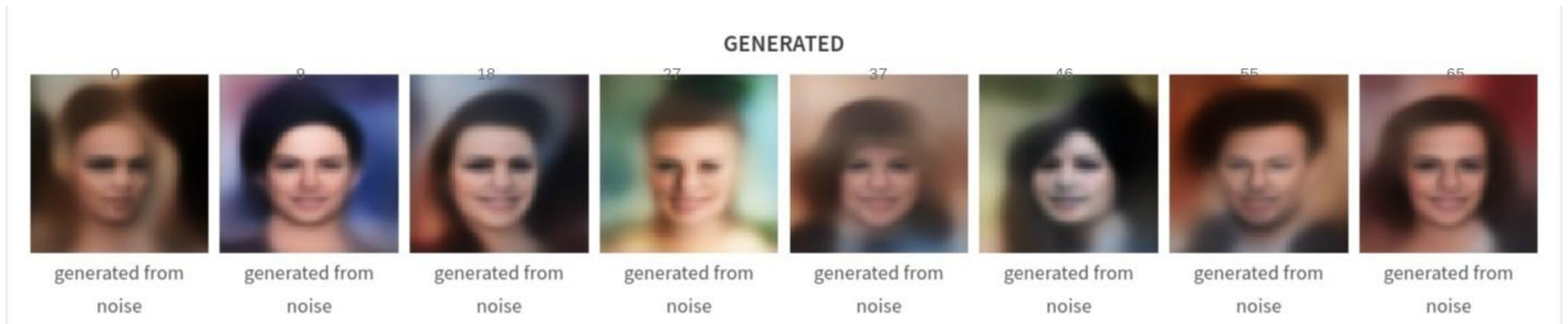
Генерация



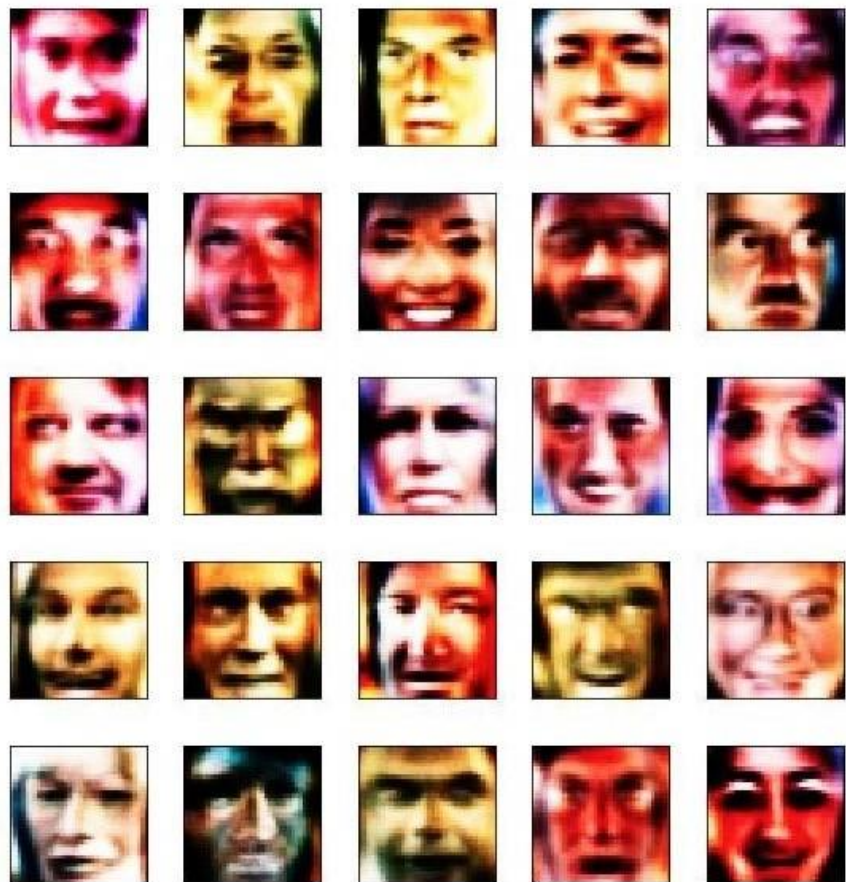
Генерация



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)

Практика

<https://colab.research.google.com/drive/16YdTvcicP1BUJAUdgBnfZNGcssqm3jhZ?usp=sharing>