

# Автокодировщики. Скрытое представление

Денис Волхонский

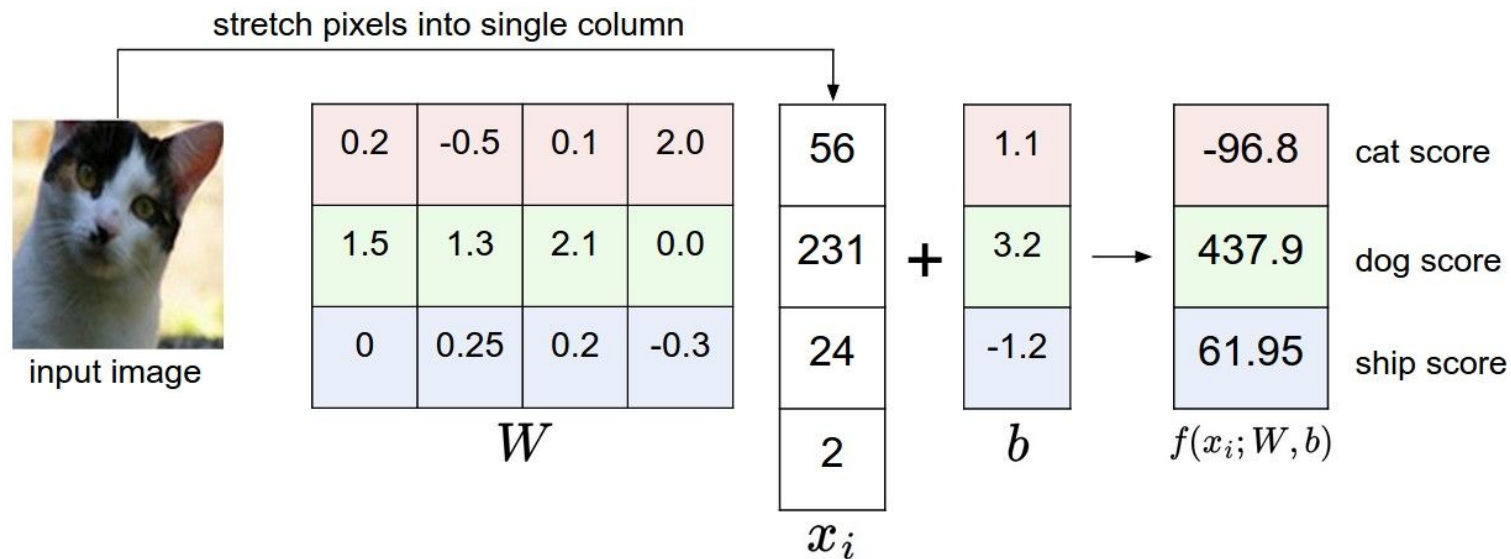
# Программа курса

- День 1: Автокодировщики
  - Семинар: автокодировщик, denoising автокодировщик, вариационный автокодировщик
- День 2: Генеративное обучение
  - Семинар: аппроксимация распределений, генерация лиц
- День 3: Основы обработки текстов
  - Семинар: генерация текста, предсказание временных рядов

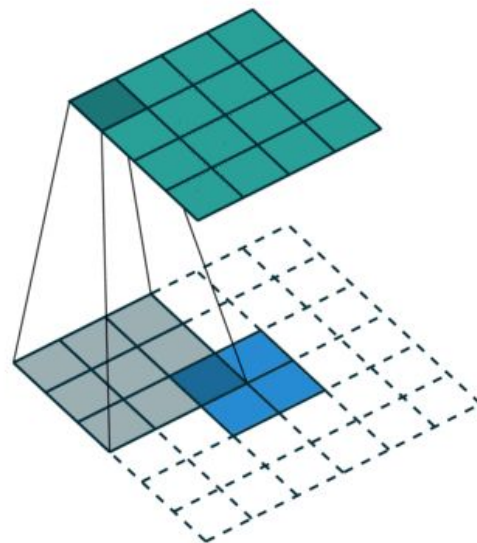
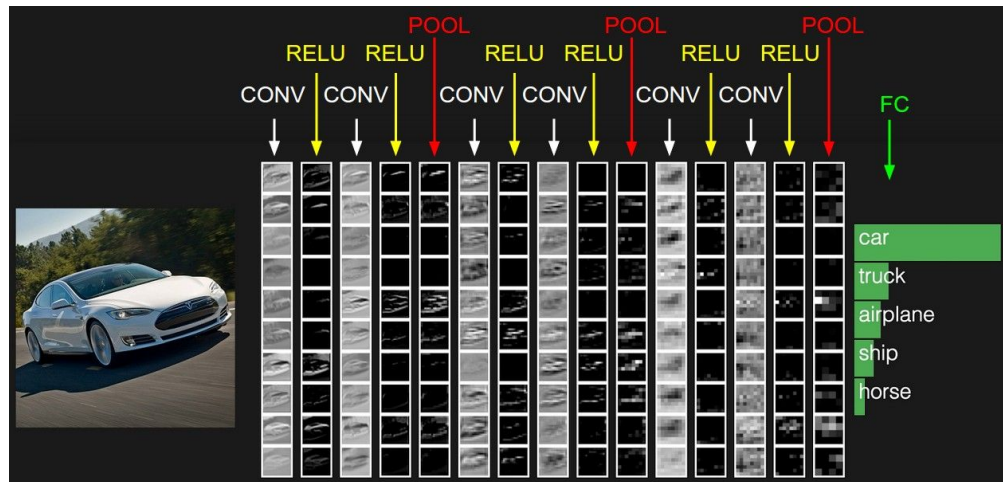
# Программа занятия

- Напоминание полносвязных и свёрточных сетей
- Autoencoders
- Denoising autoencoders
- Variational autoencoders

# Полносвязные сети



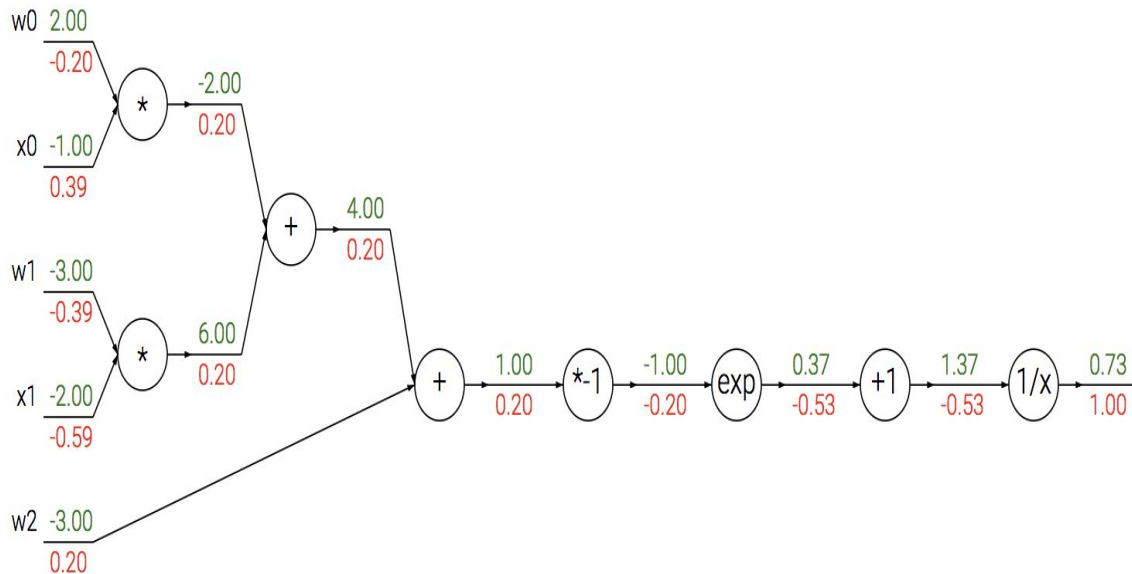
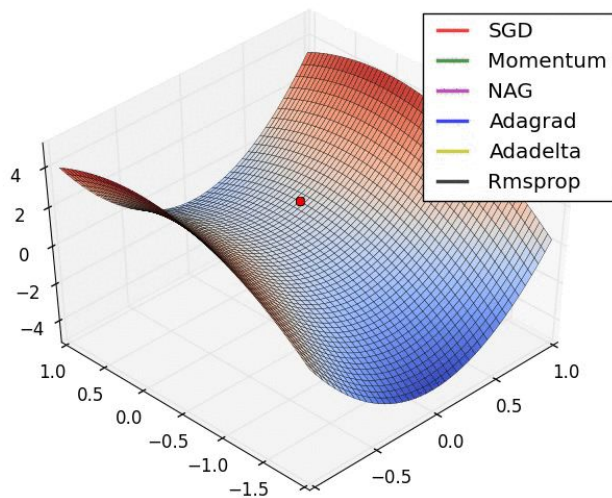
# Свёрточные сети



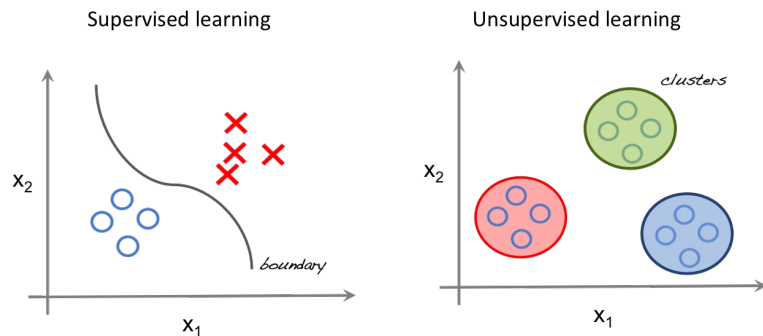
- Demo:  
<http://cs231n.github.io/convolutional-networks/>

[Source](#)

# Обучение нейросетей



# Обучение с учителем и без учителя

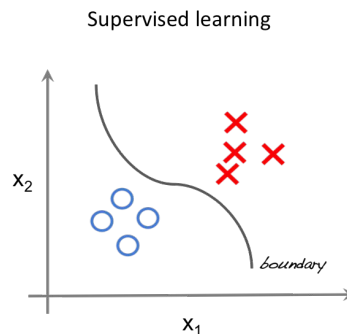


- Обучение с учителем: пары  $(x, y)$
- Обучение без учителя: только объекты  $x$

# Обучение с учителем и без учителя

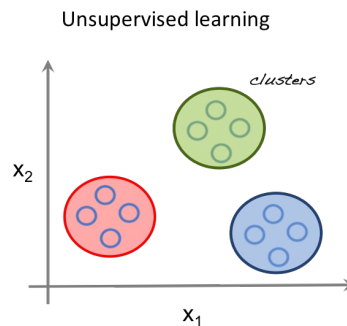
Обучение с учителем:

- Пары  $(x, y)$
- Обучение отображения  $x \rightarrow y$



Обучение без учителя:

- Только объекты  $x$
- Обучение скрытой структуры в данных

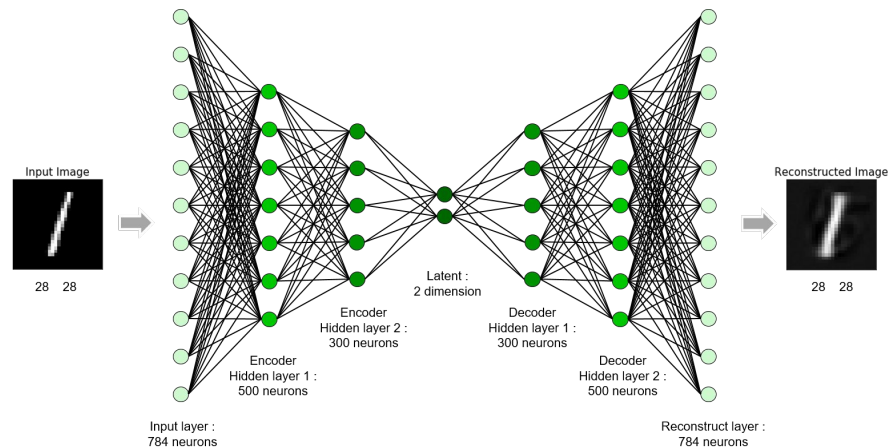
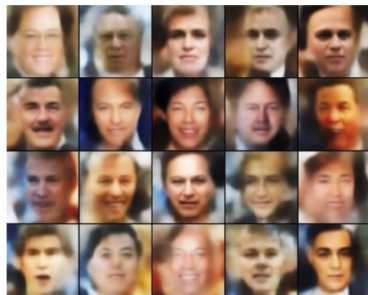


**Зачем нам обучение без учителя?**



# Зачем нам обучение без учителя

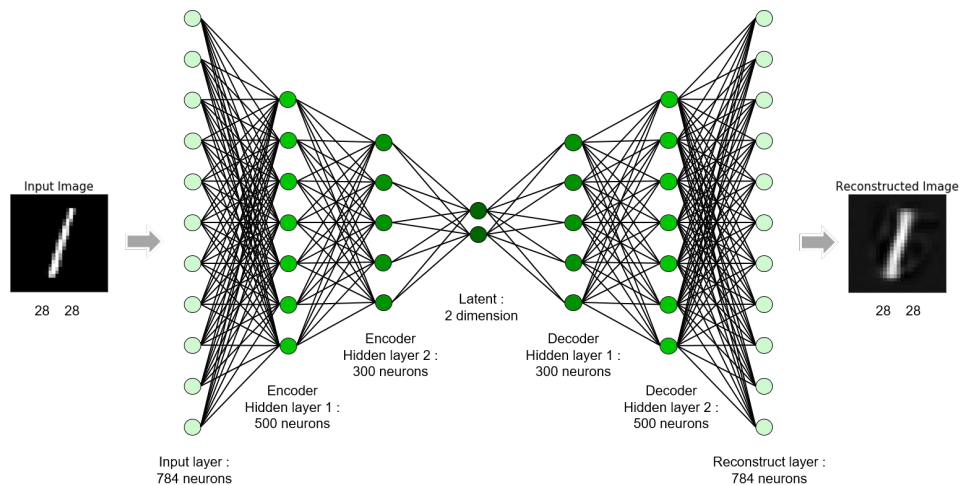
- Поиск самых подходящих признаков
- Сжатие информации
- Визуализация многомерных данных
- Поиск похожих объектов
- Генерация новых объектов



# Autoencoder

Основная идея:

- Данные из исходного (высокоразмерного) пространства
- Проецируем в другое (из которого можем восстановить их)
- Encoder: data  $\rightarrow$  hidden
- Decoder: hidden  $\rightarrow$  data
- $x \sim \text{Decoder}(\text{Encoder}(x))$

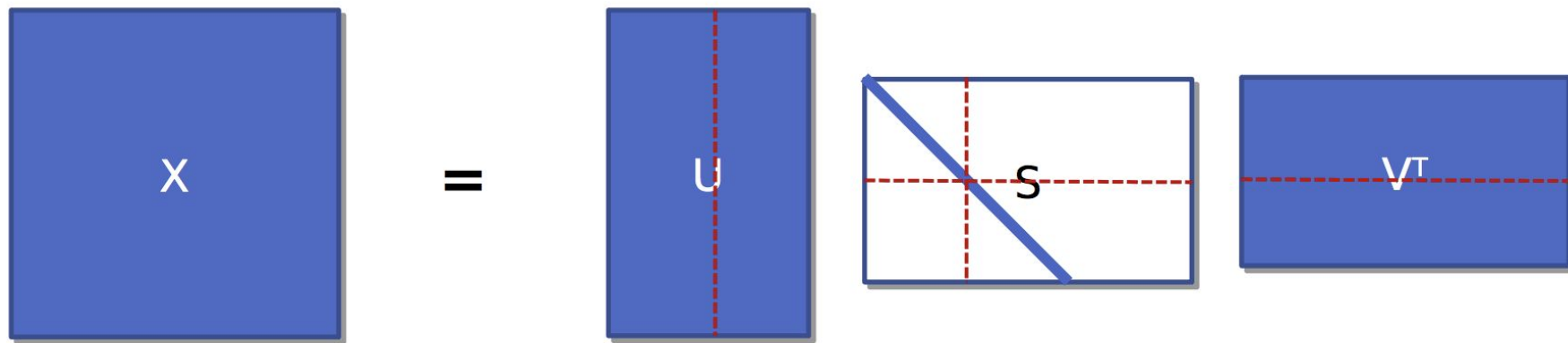


**Вопрос: зачем это может быть нужно?**

# Зачем нужны автокодировщики

- Сжатие данных
- Снижение размерности
- Обучение “хороших” признаков
- Pre-training без учителя

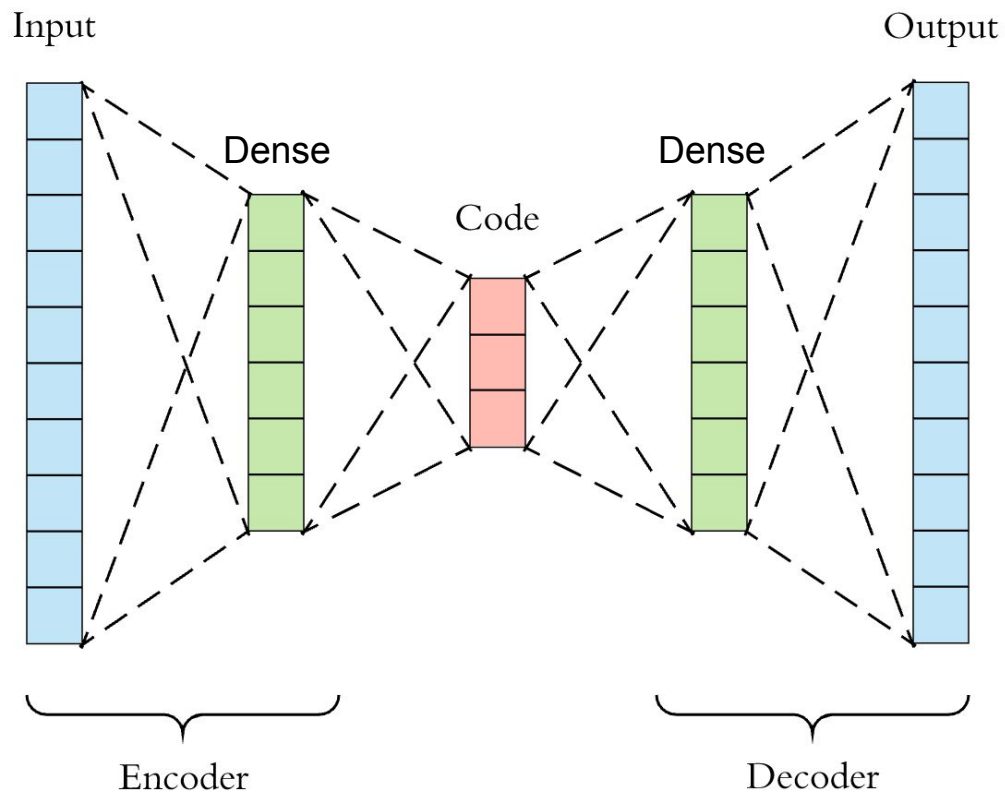
# РСА. Снижение размерности

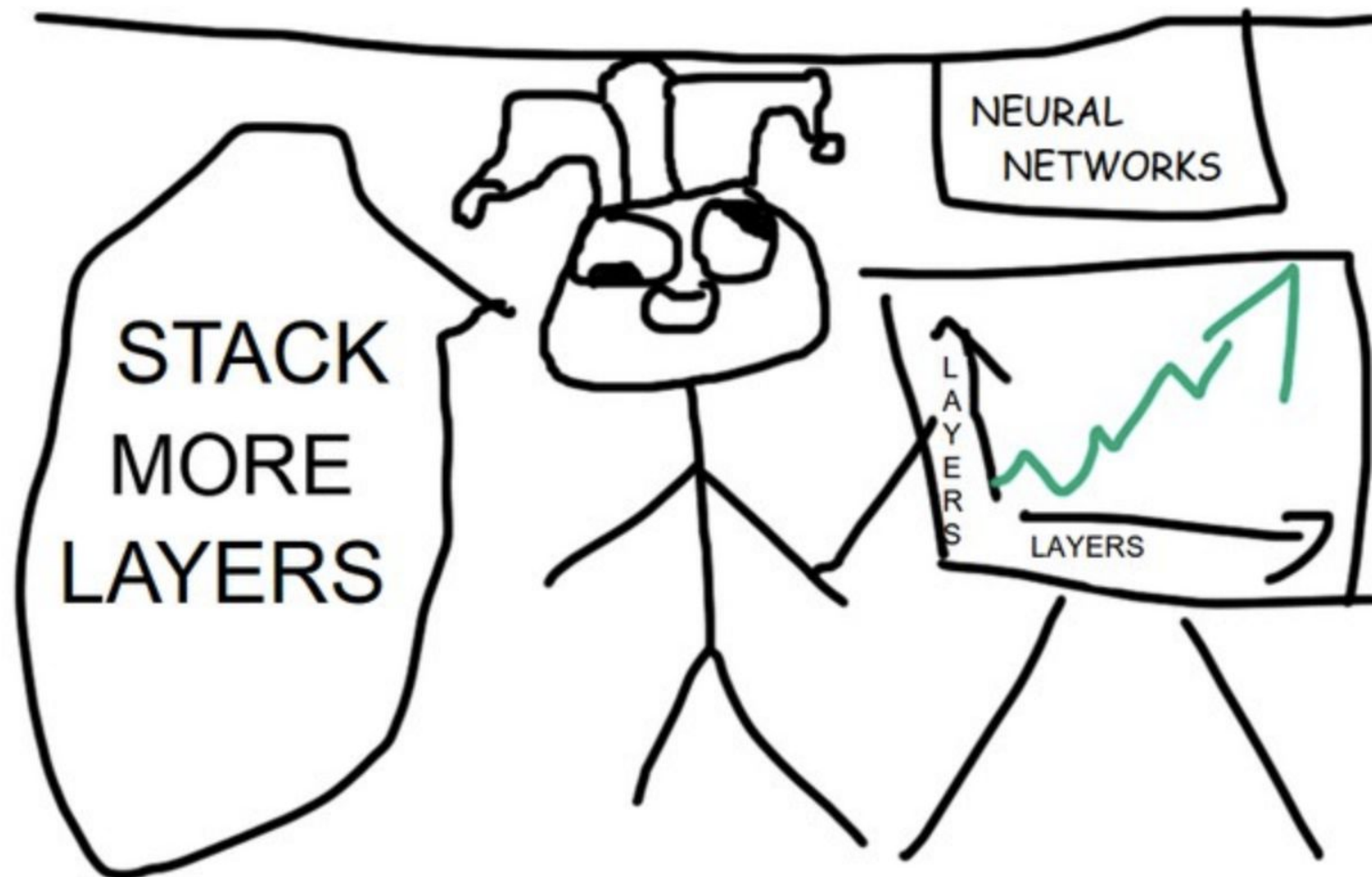


- Минимизация ошибки реконструкции

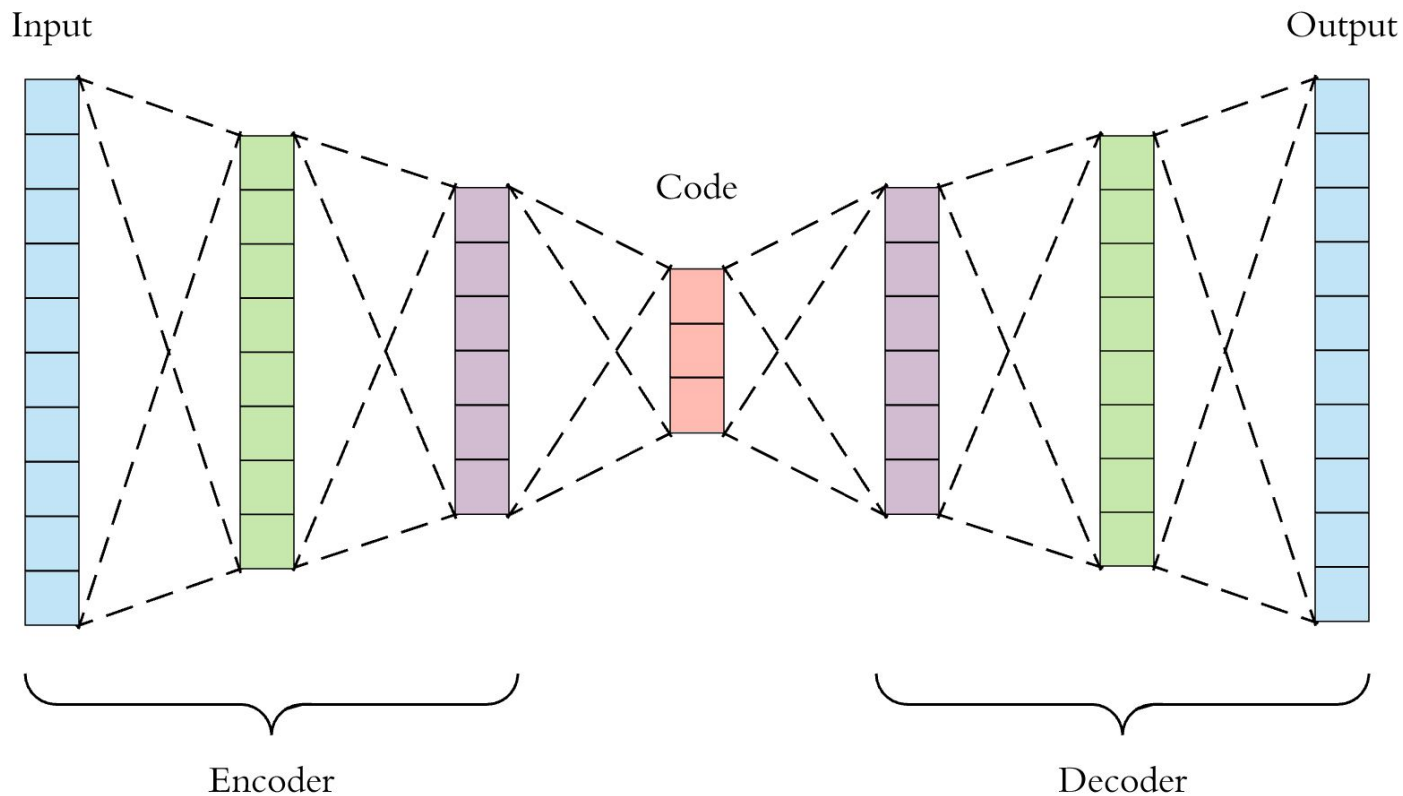
$$L = ||X - U \cdot S \cdot V^T||$$

# PCA = autoencoder

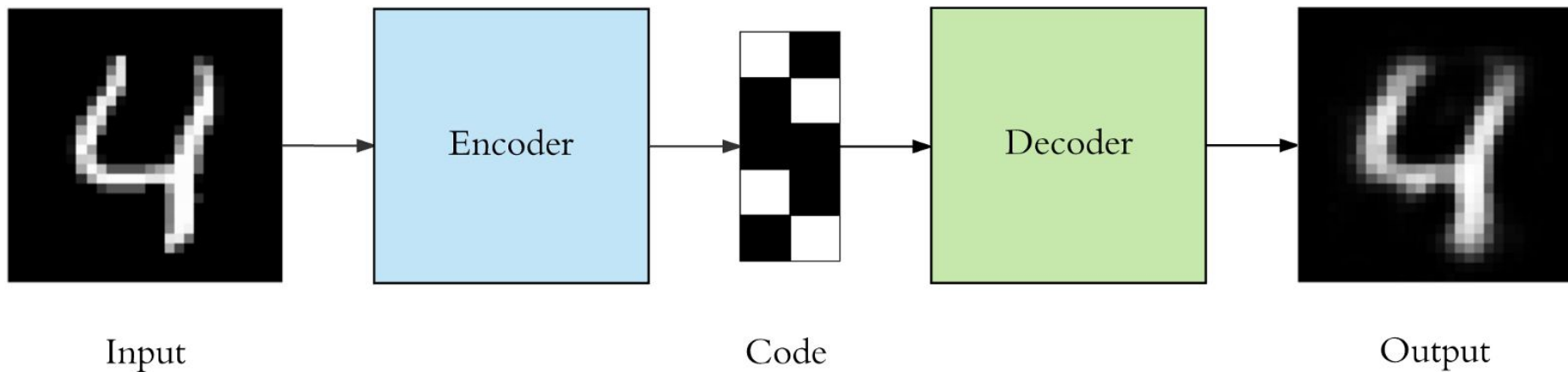




# Нужно больше слоёв!



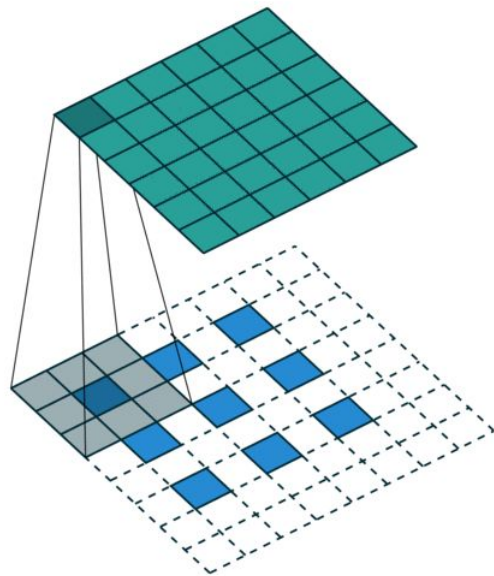
# Автокодировщик для изображений



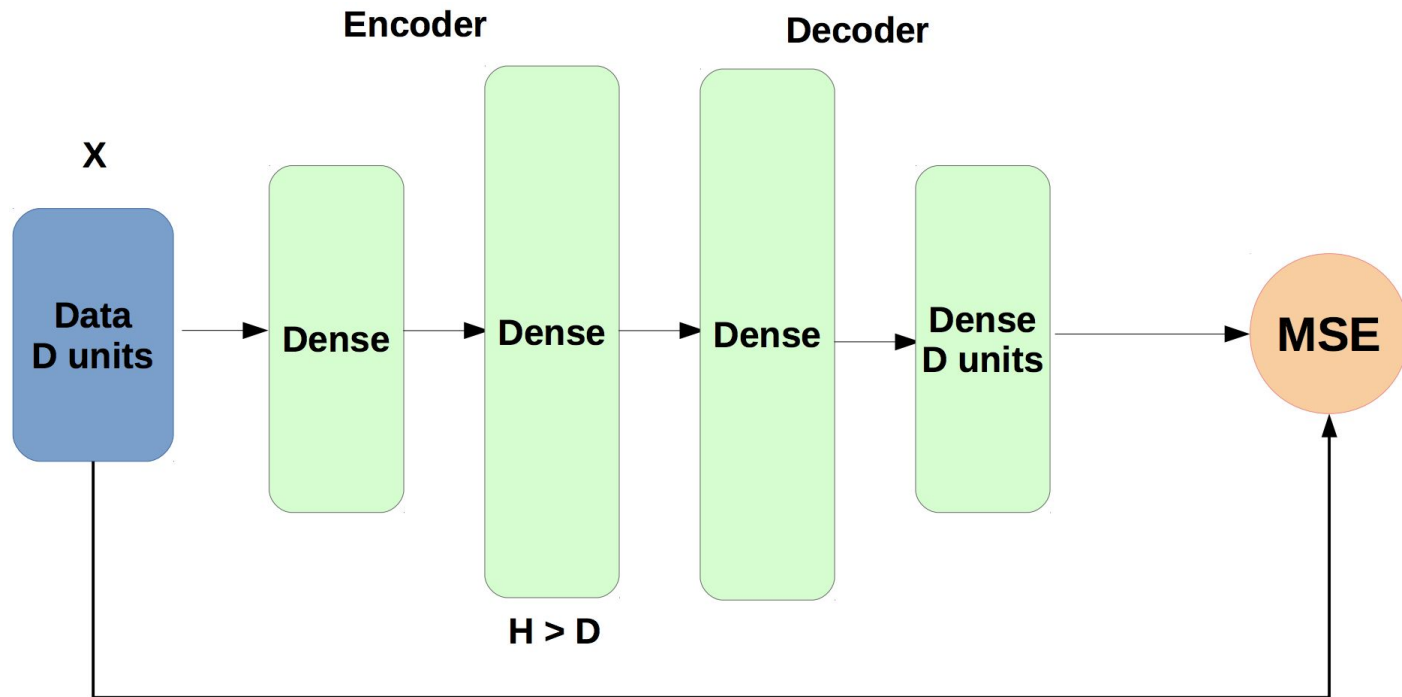


# Image upsampling

- С помощью параметров свёртки мы можем снижать/увеличивать размерность



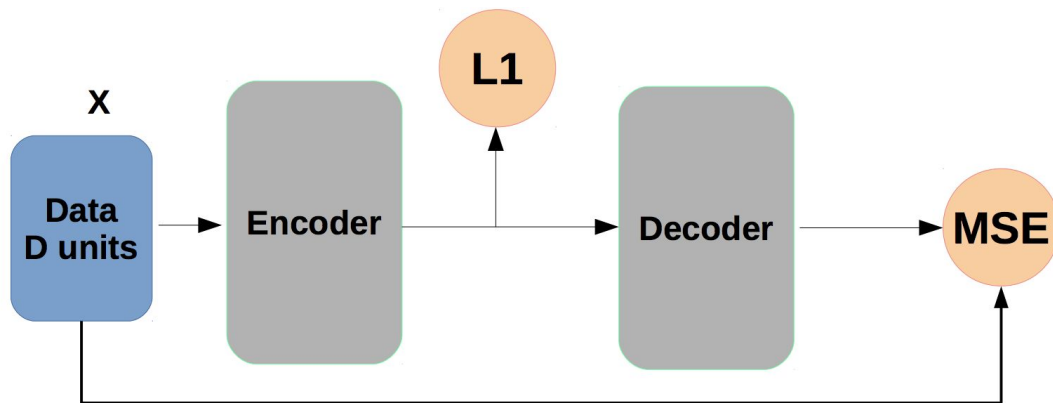
# Увеличиваем скрытое представление



Что не так с этой архитектурой?

# Увеличиваем скрытое представление

- Мы выучим тождественное отображение!
- Добавим регуляризацию

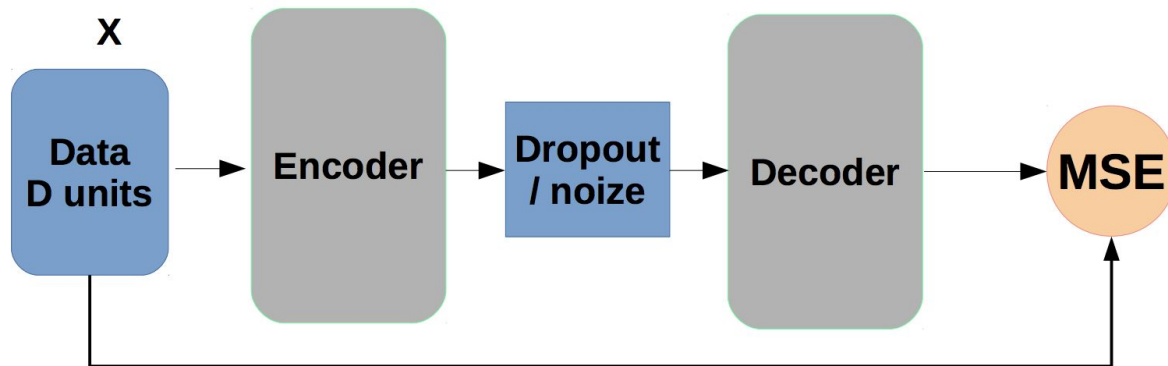


$$L = \|X - Dec(Enc(X))\|^2 + \sum_i |Enc_i(X)|$$

Какие ещё варианты?

# Увеличиваем скрытое представление

- В качестве регуляризации можем добавить шум или dropout

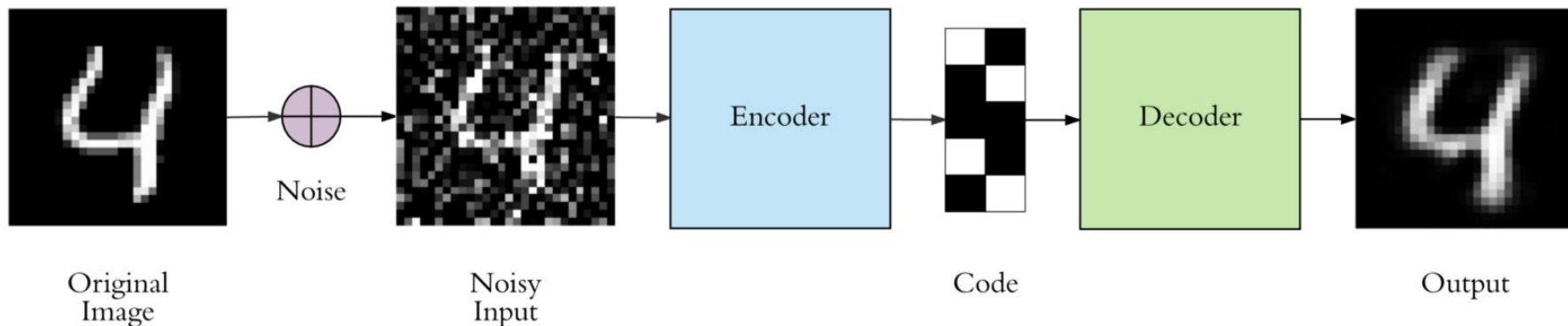


$$L = ||X - Enc(Noize(Dec(X)))||$$

Как нам сделать denoising autoencoder?

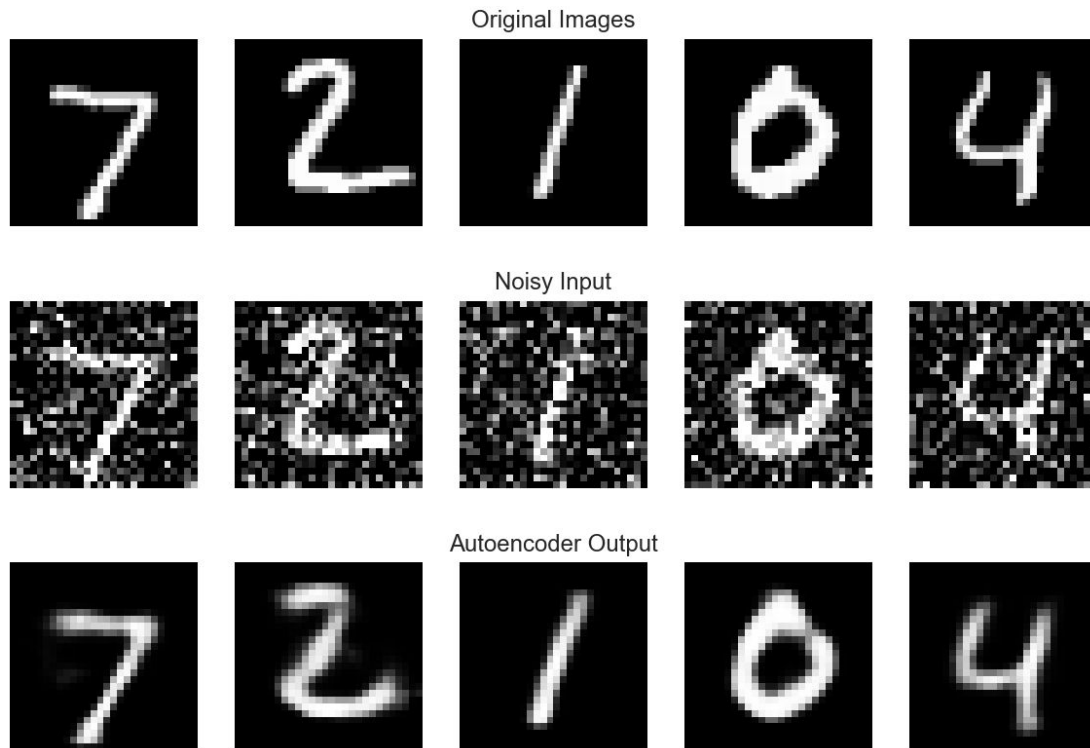
# Denoising autoencoder

- Добавляем в исходное изображение шум
- Пытаемся от него избавиться

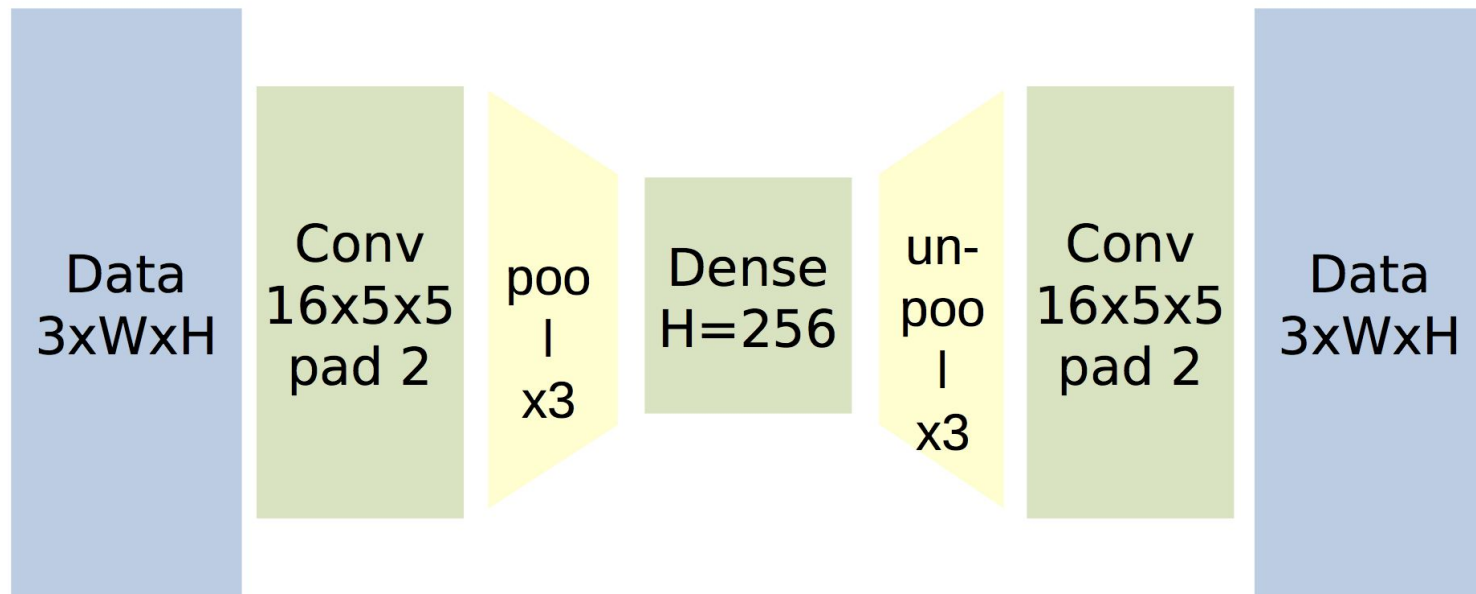


$$L = \|X - \text{Enc}(\text{Dec}(\text{Noize}(X)))\|$$

# Denoising autoencoder



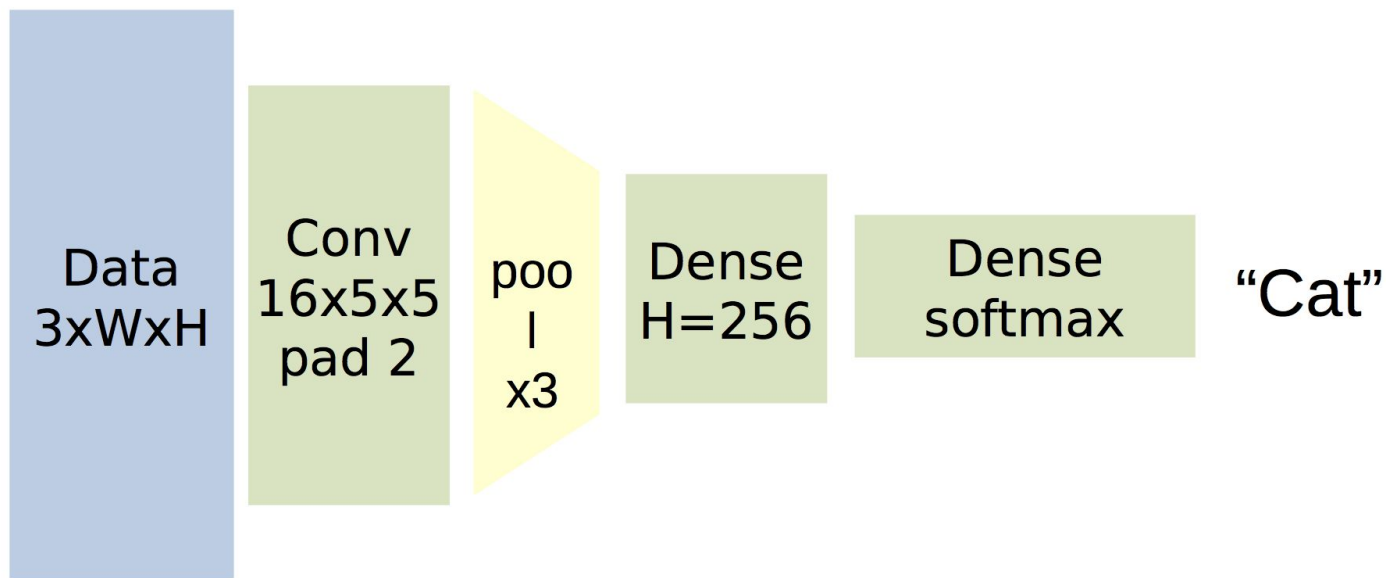
# Автокодировщик для предобучения нейросети



Как можно использовать автокодировщик для предобучения нейросети?

# Автокодировщик для предобучения нейросети

- Использовать энкодер как инициализацию



Какие преимущества от предобученного автокодировщика? Есть же Image Net



# Преимущества предобученного автокодировщика

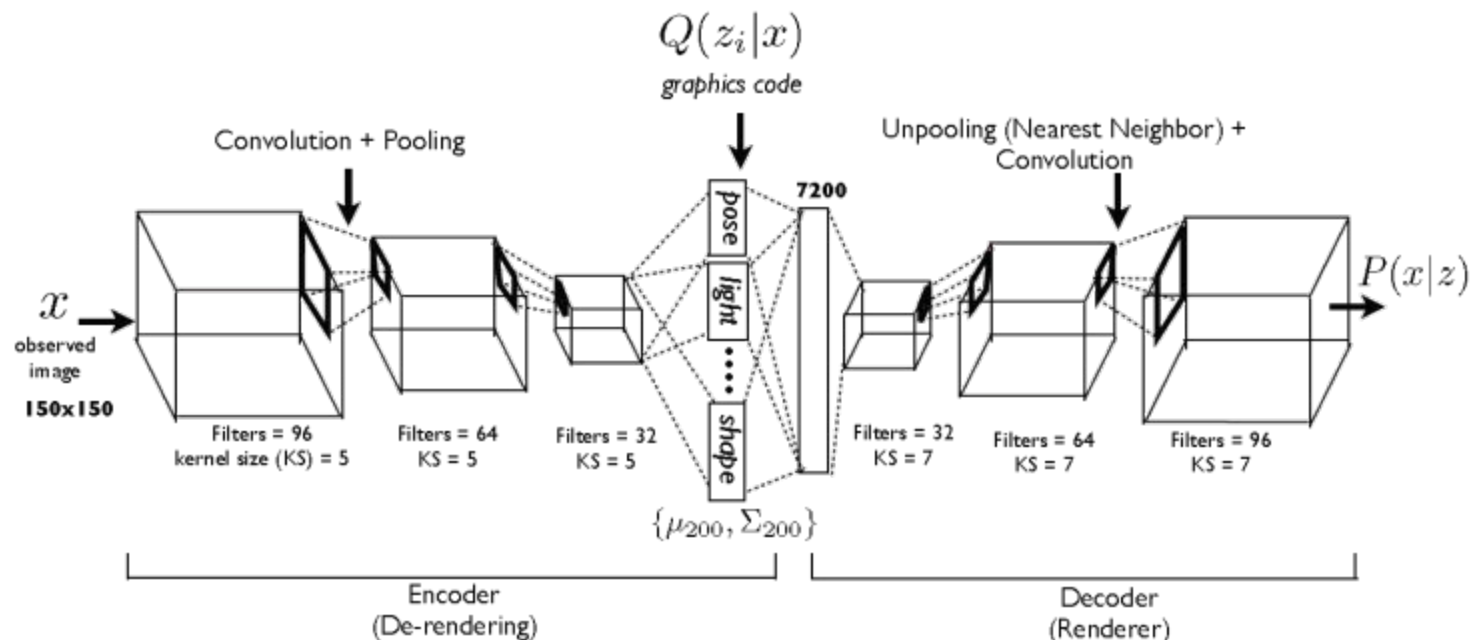
- Не нужны метки классов
- Подходит для редких задач (в отличие от Image Net)

# Deep Fake

- Обучим 2 отдельных автокодировщика для двух разных людей
- Возьмём энкодер от одного и декодер от другого
- Результат: видео  
<https://www.youtube.com/watch?v=BU9YAHigNx8>

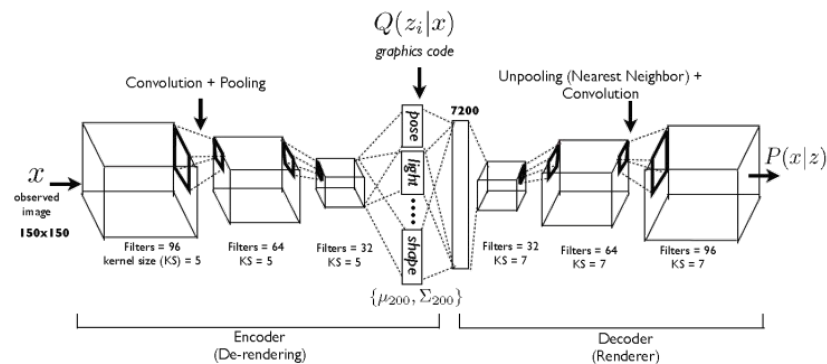


# Deep Inverse Graphics

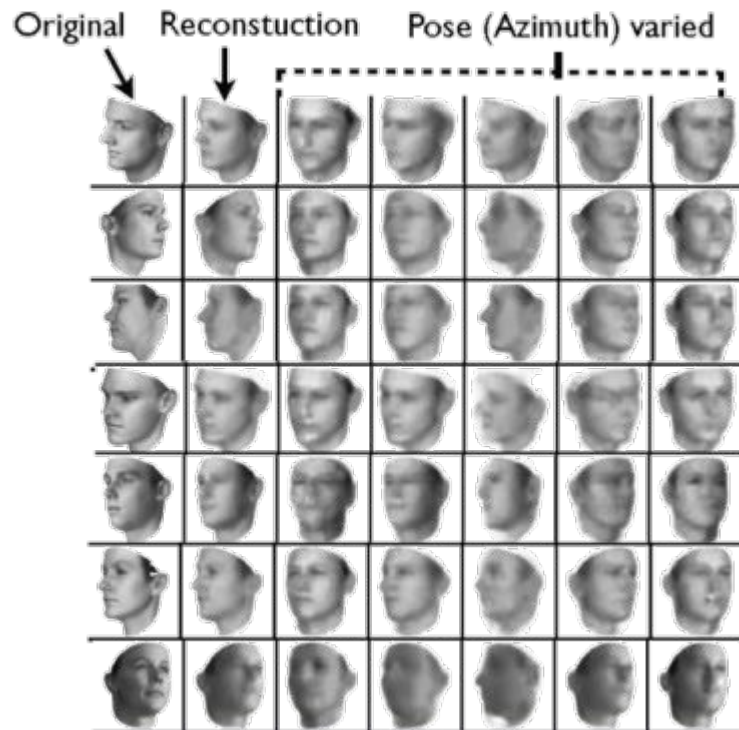
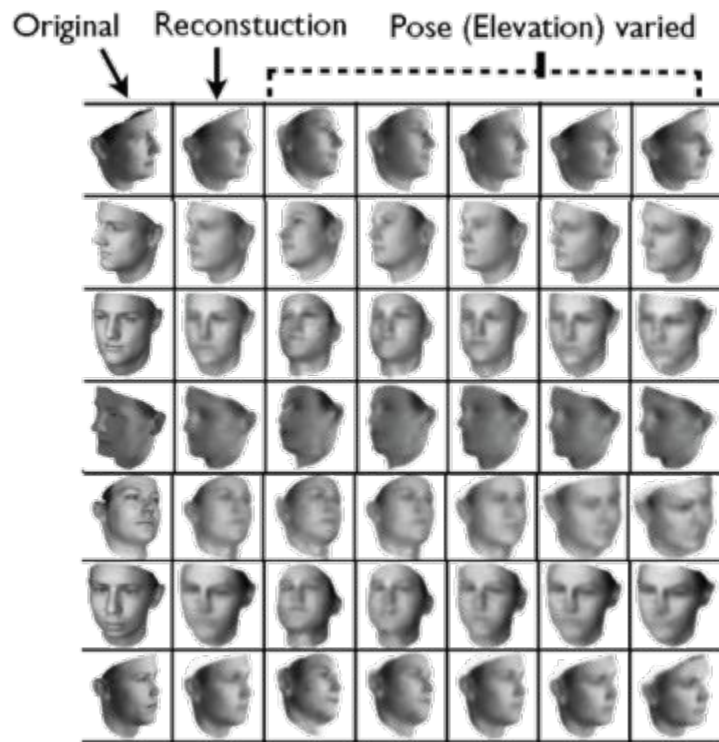


# Deep Inverse Graphics

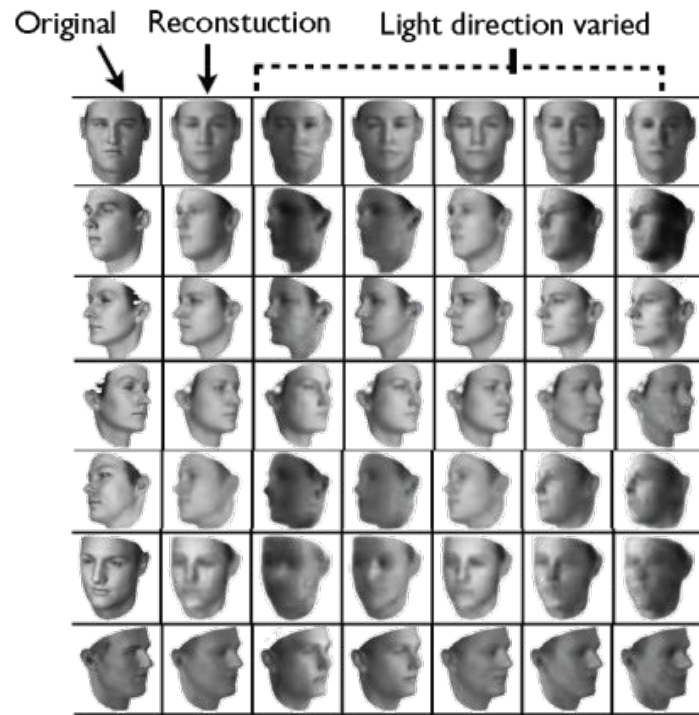
- Батч, у которого некоторый параметр фиксирован (одна поза)
- При прямом распространении заменяем  $z$  на усреднённый по батчу
- При обратном распространении добавляем штраф за отклонение  $z$  от среднего по батчу



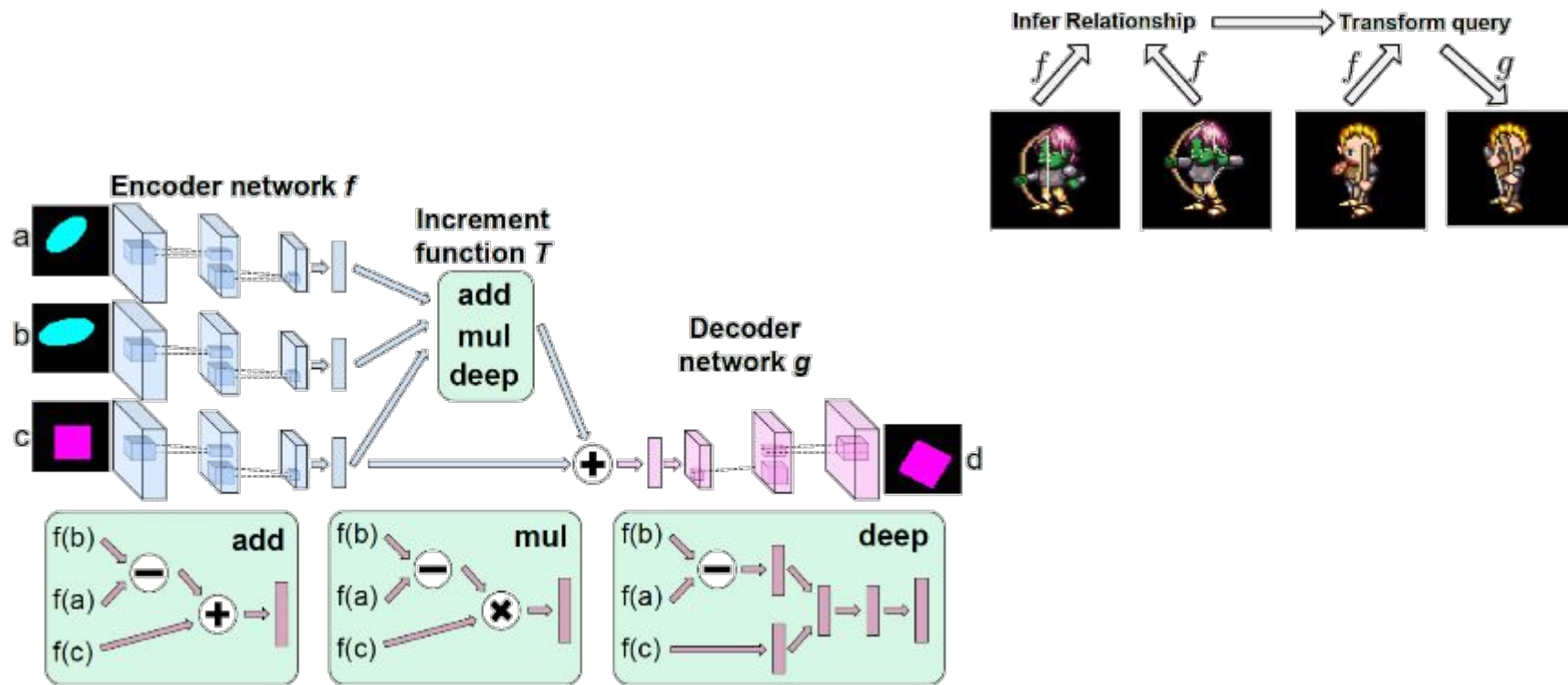
# Deep Inverse Graphics



# Deep Inverse Graphics



# Deep Visual Analogy Making



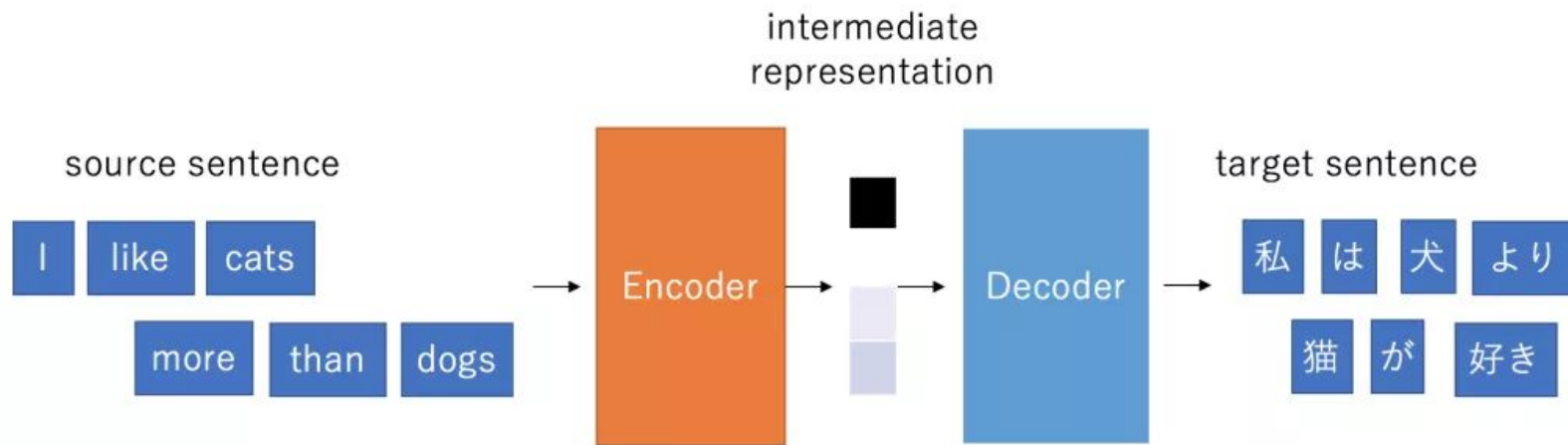
[Reed et al. NIPS 2015]

# Deep Visual Analogy Making





# Перевод текста



# Заключение

- Зачем нужны автокодировщики?

# Практика

Автокодировщик

# Семинар

- Адрес: 107.178.212.42
- git clone [https://github.com/dvolkhonskiy/pytorch\\_rusky.git](https://github.com/dvolkhonskiy/pytorch_rusky.git)
- pip install pandas==0.21 sklearn tqdm -U
- Обновить: jovyan@jupyter-den:~/pytorch\_rusky\$ git pull