

Глубокое обучение. Работа с изображениями. Сверточные сети. Механизм внимания в сверточных сетях. Сегментация. GAN

Санкт-Петербургский государственный университет
Кафедра статистического моделирования

Глубокое обучение: Введение

В чем разница между классическим машинным обучением и глубоким обучением?

Модели глубокого обучения — **нейронные сети**.

Th Хорника: для любой непрерывной функции найдется нейронная сеть с линейным выходом, аппроксимирующая функцию с заданной точностью.

Слой нейронной сети — преобразование $\sigma(\mathbf{XW} + b)$, где

- 1 \mathbf{X} — входные данные слоя, строки — индивиды, столбцы — признаки.
- 2 \mathbf{W}, b — веса слоя (обучаемые параметры).
- 3 σ — нелинейная функция. Требования: непрерывная, монотонная, желательно дифференцируемая.

Classic ML:

- Модели имеют не более одного нелинейного слоя, не требуют много вычислительных ресурсов.
- Feature extraction: звлечение признаков из неструктурированных данных требует сложной аналитической работы.
- Feature engineering: сложная аналитическая работа.

Deep Learning:

- 1 Модели имеют более одного нелинейного слоя. Требуют значительные вычислительные ресурсы.
- 2 Feature extraction: не требует много трудозатрат.
- 3 Feature engineering: происходит автоматически ввиду нелинейных преобразований.

Глубокое обучение: Архитектуры

Гибкость архитектуры нейронной сети: любое число выходов.

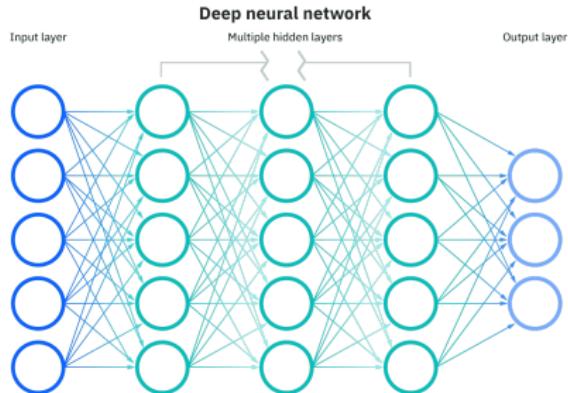


Рис.: Архитектура нейронной сети

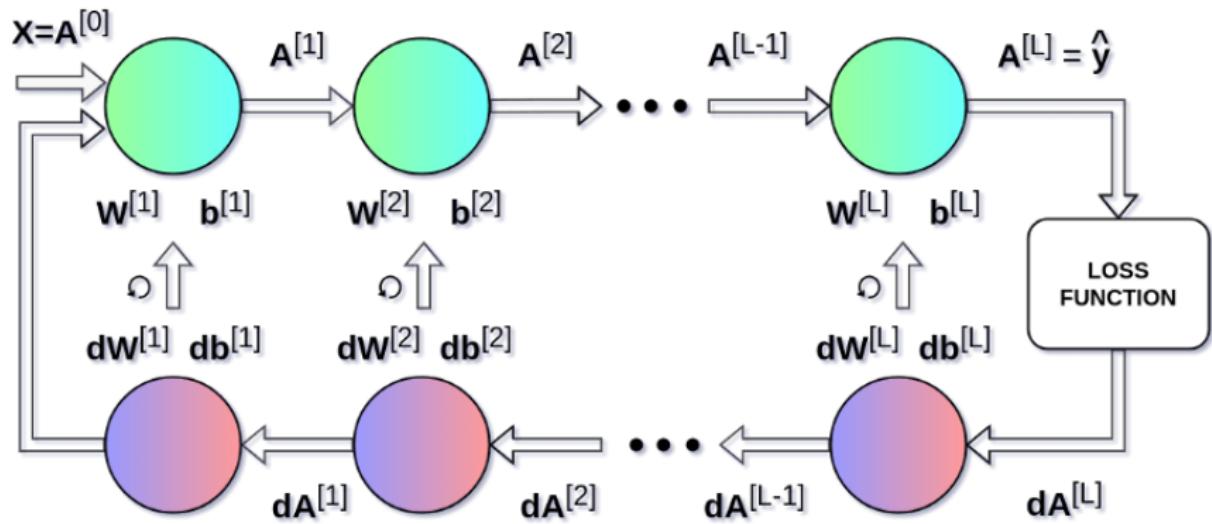
Это означает, что можно решать задачу как многомерной регрессии, так и многоклассовую классификацию:

$$\text{softmax}(\mathbf{X}W_j + b_j) = \frac{\exp(\mathbf{X}W_j + b_j)}{\sum_{i=1}^K \exp(\mathbf{X}W_i + b_i)}$$

Нейронные сети: обучение

Нейронная сеть обучается с помощью **прямого** прохода и **обратного** распространения ошибки (forward/backward propagation). Используется градиентный спуск.

FORWARD PROPAGATION



Затухающий градиент (vanishing gradient):

- При больших по модулю значениях аргумента функций активации \tanh , sigmoid производная функции активации стремится к нулю.
- Таким образом градиент не распространяется.

Решение: Использование других функций активации; использование residual blocks.

Взрывающийся градиент (exploding gradient):

- Накопление больших по модулю градиентов.
- Наблюдается, например, при использовании sigmoid в скрытых слоях: градиент растет экспоненциально.

Решение: Регуляризация и **Gradient clipping**:

$$\nabla_c = \begin{cases} \nabla, & \|\nabla\| < T, \\ \frac{T}{\|\nabla\|} \nabla, & \text{otherwise} \end{cases}$$

Регуляризация:

- 1 L1, L2 штрафы к функции потерь
- 2 Dropout (2014) — как добиться того, чтобы модель использовала все свои параметры? Случайное зануление доли весов.
- 3 BatchNorm (2015) — автоматическая стандартизация входа слоя.

Прореживание (pruning) — сжатие предобученной сети за счет устранения части параметров.

- Самые популярные техники прореживания (**OBS, OBD**) основаны на разложении функции потерь в ряд Тейлора: устранием веса с самым низким ростом функции потерь.

Работа с изображениями

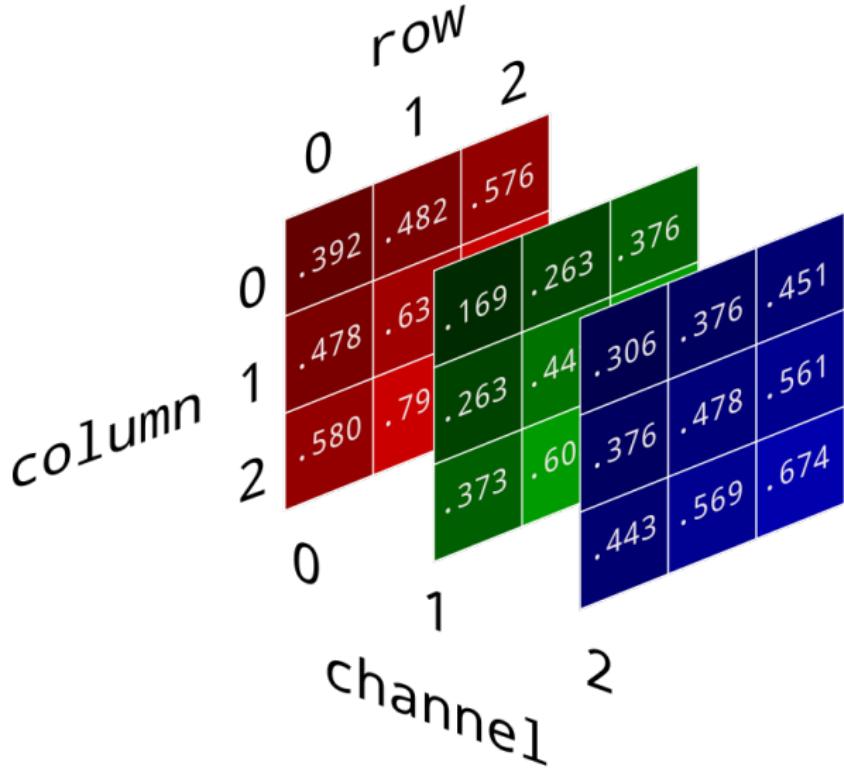


Рис.: Представление изображения в памяти

Распознавание цифр (MNIST)

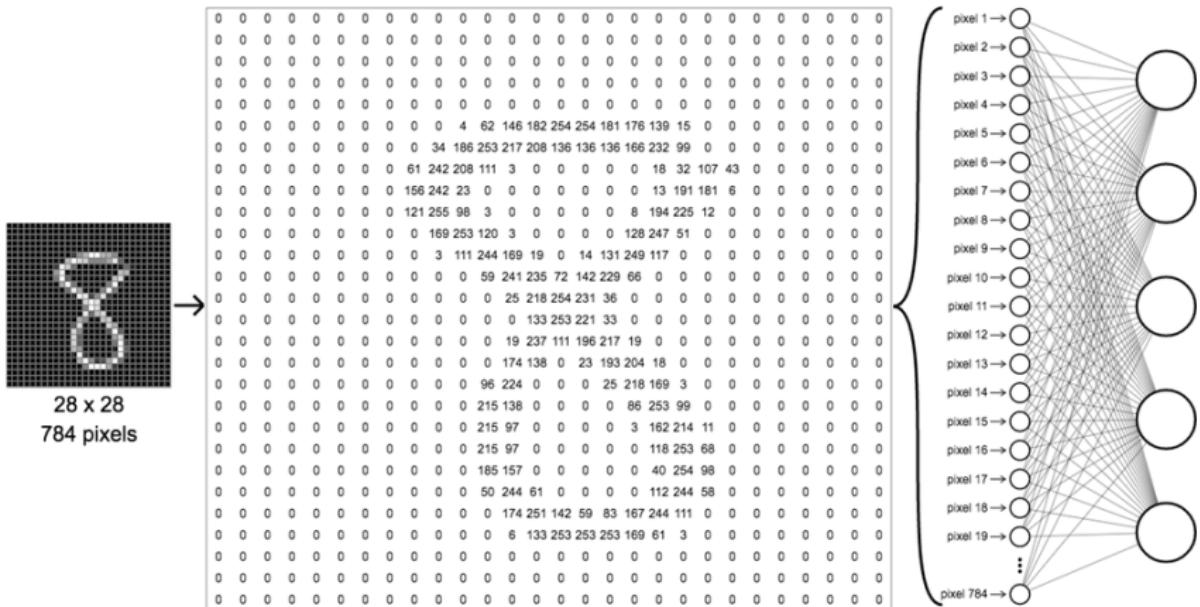


Рис.: Первые архитектуры ANN для распознавания цифр

Недостаточная точность, большое количество параметров

Первые архитектуры сверточных сетей

Самая первая широко известная архитектура — LeNet 5 (1995).
Input: 32x32 gray-scale, свертки 5x5, 60К параметров.

Convolutions on RGB image

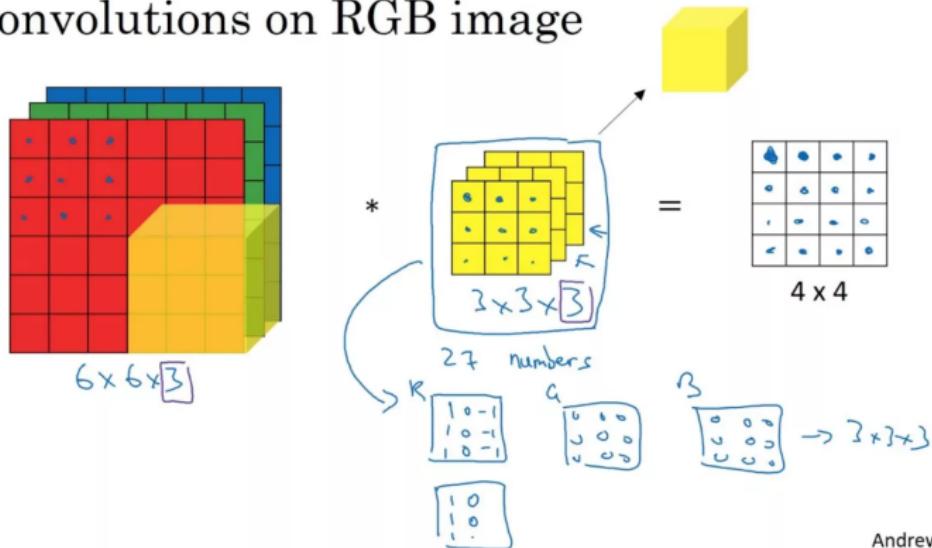


Рис.: Convolutional layer

Andrew Ng

Первые архитектуры сверточных сетей

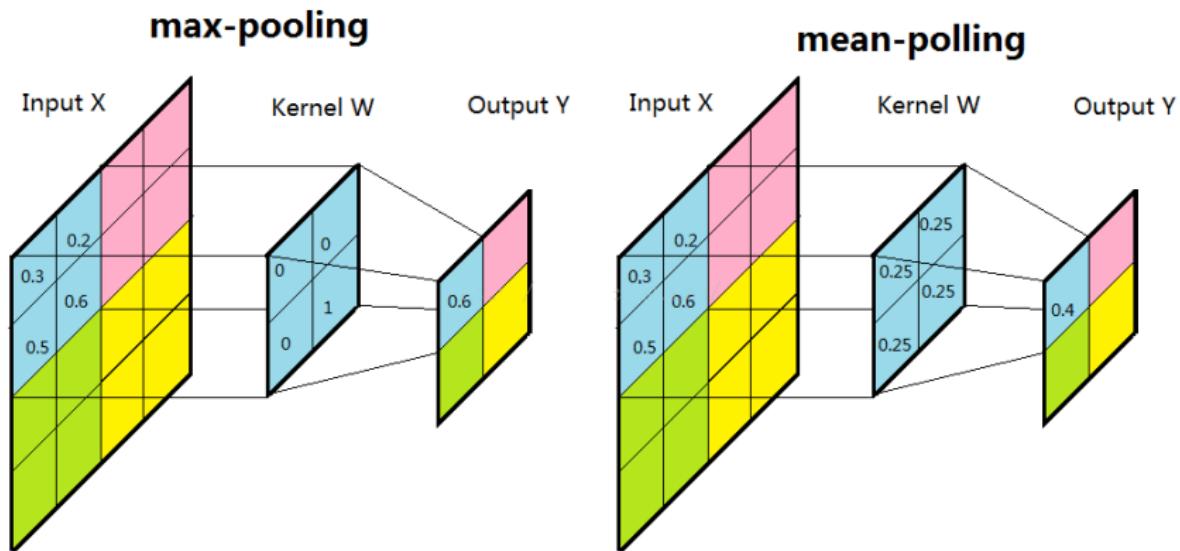


Рис.: Max-Pooling layer

Первые архитектуры сверточных сетей

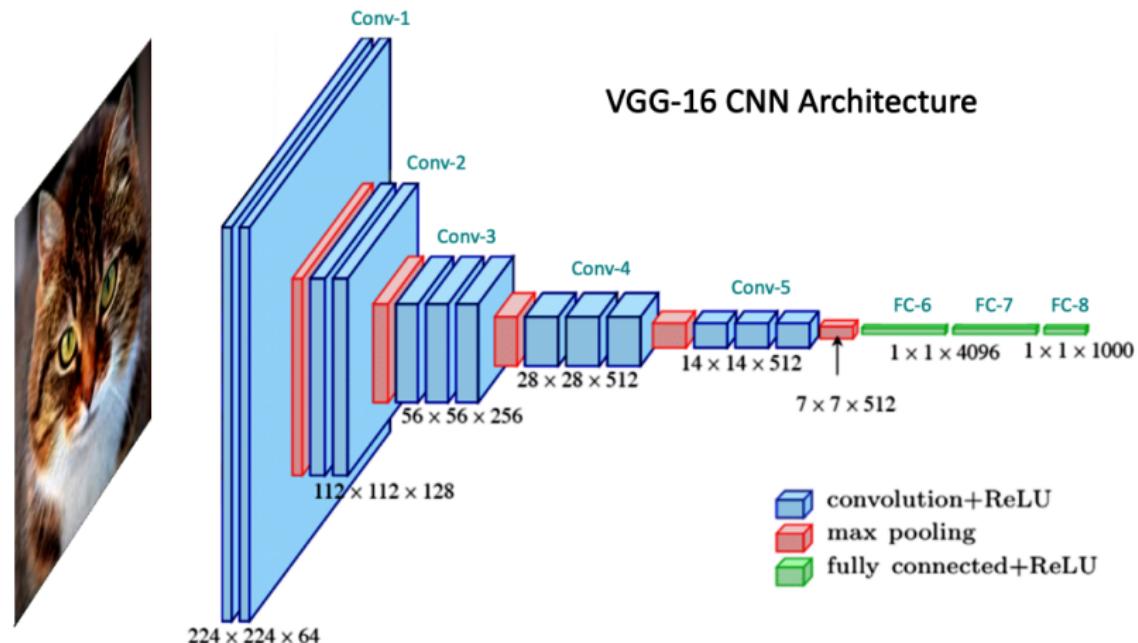
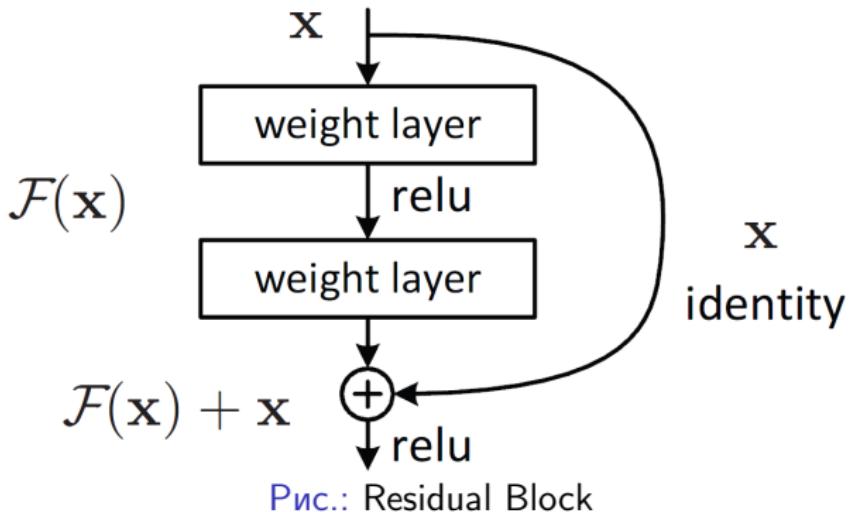


Рис.: Первые архитектуры сверточных сетей

Первые архитектуры сверточных сетей

- 1 LeNet 5 (1995). Input: 32x32 gray-scale, свертки 5x5, 60К параметров.
- 2 AlexNet (2012) Input: 227x227x3, свертки 11x11, 5x5, 3x3, 60М параметров. Победа в ImageNet и технологический прорыв.
- 3 VGG-16 (2015) Input: 224x224x3, использование двойных сверток, 138М параметров.
- 4 ResNet (2015): использование residual blocks — технологический прорыв.



- Добавление residual blocks не сильно вредит сети.
- Справляется с проблемой затухающего градиента.
- В каждой современной сверточной сети есть residual blocks.

Inception (2014)

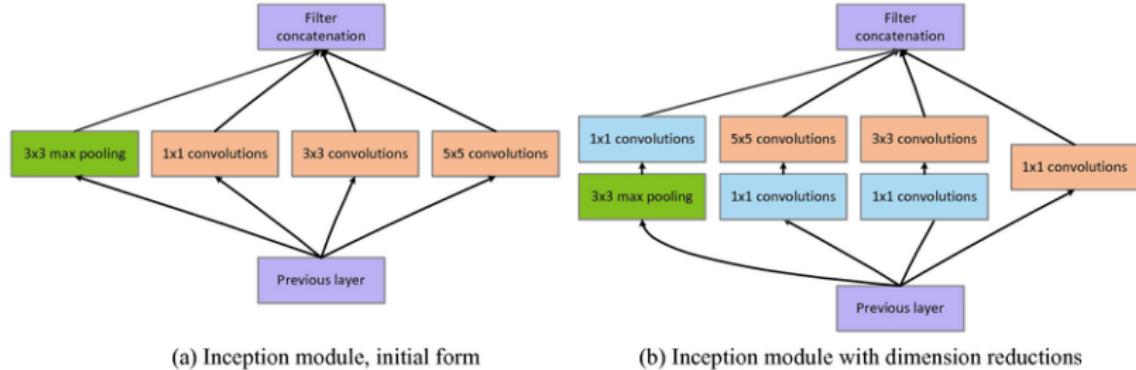


Рис.: Inception v1 blocks

- Идея нескольких выходов и составная Loss function позволили справиться с проблемой затухающего градиента.
- Bottleneck layer: свертки 1x1 позволяют уменьшать количество каналов.

Inception (2014)

convolution

max pooling

channel concatenation

channel-wise normalization

fully-connected layer

softmax

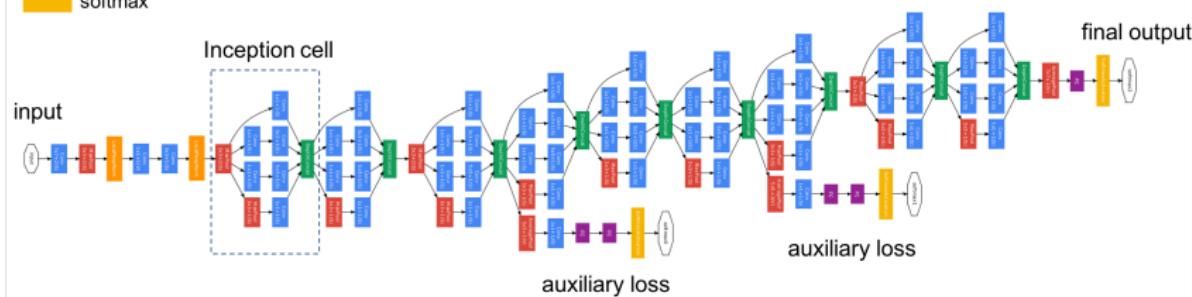


Рис.: Inception v1

Преимущества сверточных сетей

- В сверточных блоках параметров в разы меньше, чем в полносвязных.
- Сверточные блоки позволяют извлекать сложные признаки.
- Transfer learning: сверточная сеть, обученная на широком домене, может быть дообучена для более узкого домена.
- Можно визуализировать карты признаков из сверточных блоков (Layer Heatmap)

Layer Heatmap

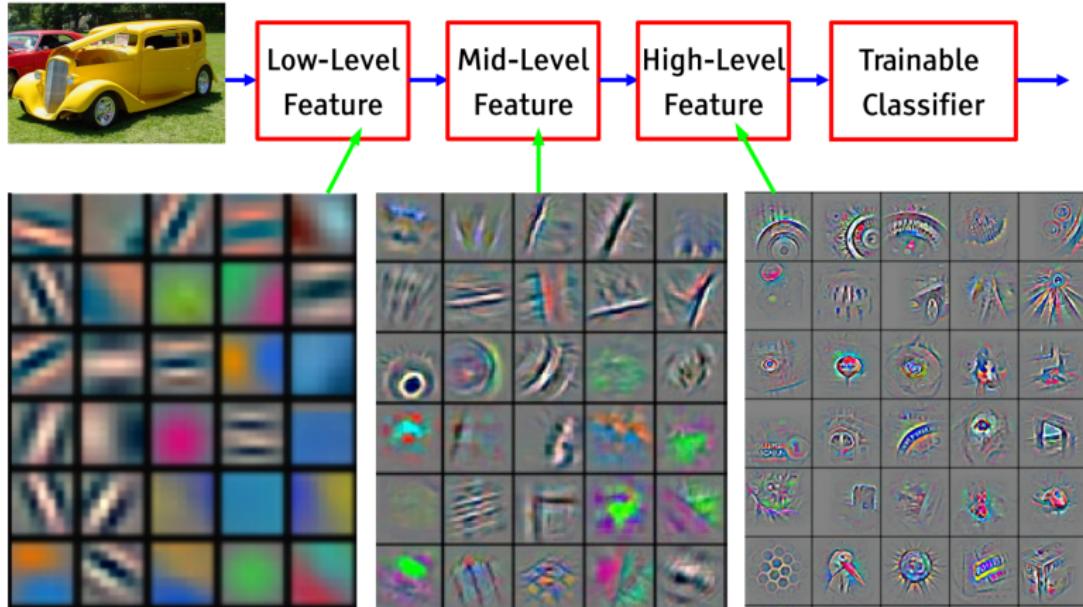


Рис.: Classifier Conv layer Heatmap

Механизм внимания в сверточных сетях

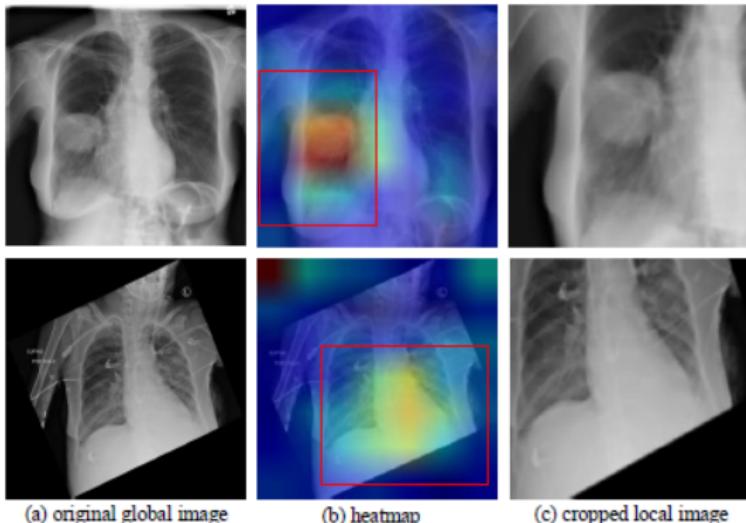


Рис.: Attention in Conv Networks

- Базовая идея: не все части входной карты признаков важны, можно сфокусироваться на некоторых регионах.
- Реализация: считаем *attention weights* (некоторый слой + softmax) \Rightarrow домножаем карту признаков на *attention weights*.

Сегментация: базовая идея

- Задача сегментации: необходимо создать маску, которая будет разделять изображение на несколько классов.

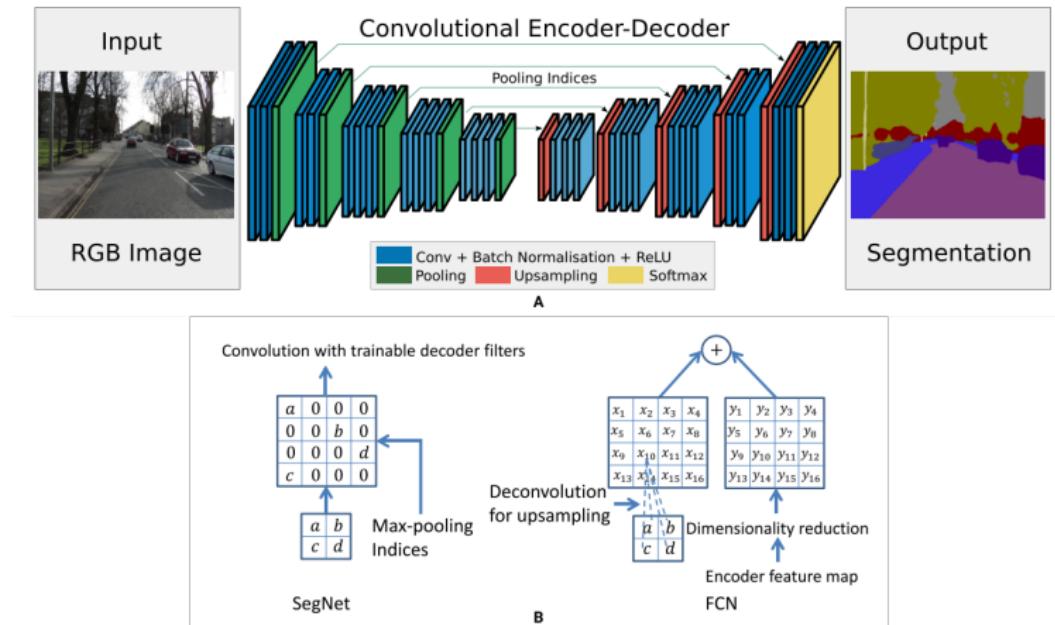


Рис.: Segmentation Architecture

Сегментация: Unet (2015)

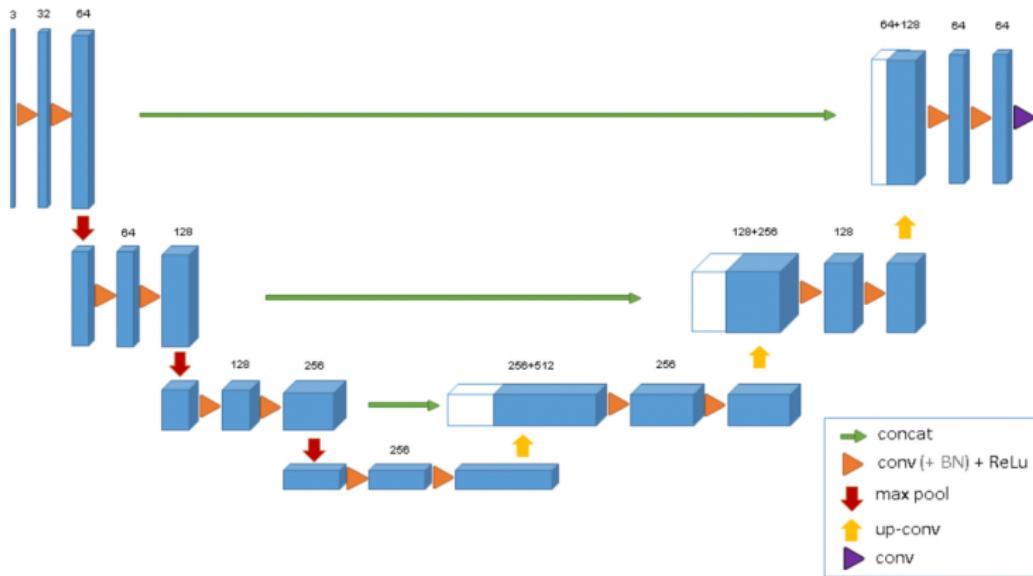


Рис.: Unet (2015) Architecture

- UpConv слой: возвращение размерности выходной карты признаков к размерности входной.

- Генеративные сети — нейронные сети, которые генерируют реалистичные образцы данных, подобные тем, на которых они обучались.

Составляющие GAN:

- 1 Генератор — модель, которая учится из нормального распределения получать изображение.
- 2 Дискриминатор — модель, которая классифицирует выход генератора как настоящее изображение или ненастоящее.
- 3 Генератор и Дискриминатор обучаются попеременно.

GAN: Введение

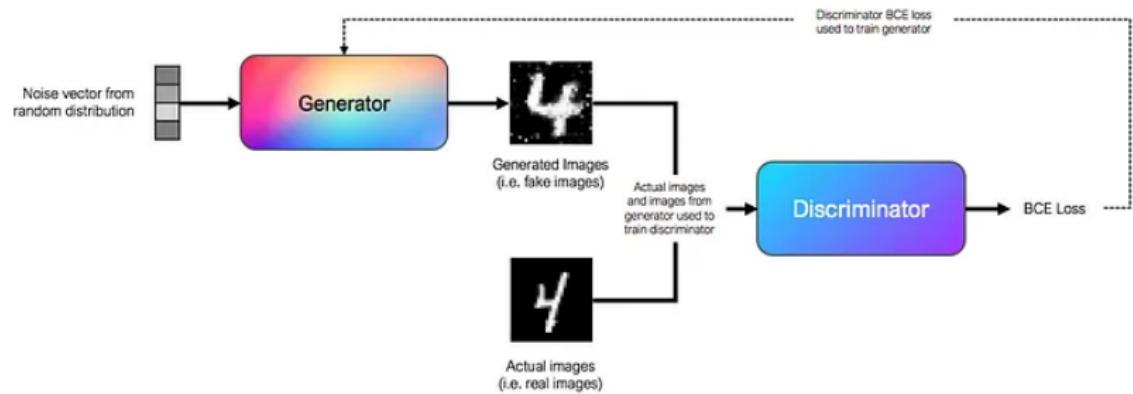


Рис.: GAN Architecture

GAN: История создания

- DCGAN (2015): вектор из нормального распределения размера 100 и Deep CNN.



Рис.: DCGAN images

GAN: История создания

- Stable Diffusion: text-to-image модель

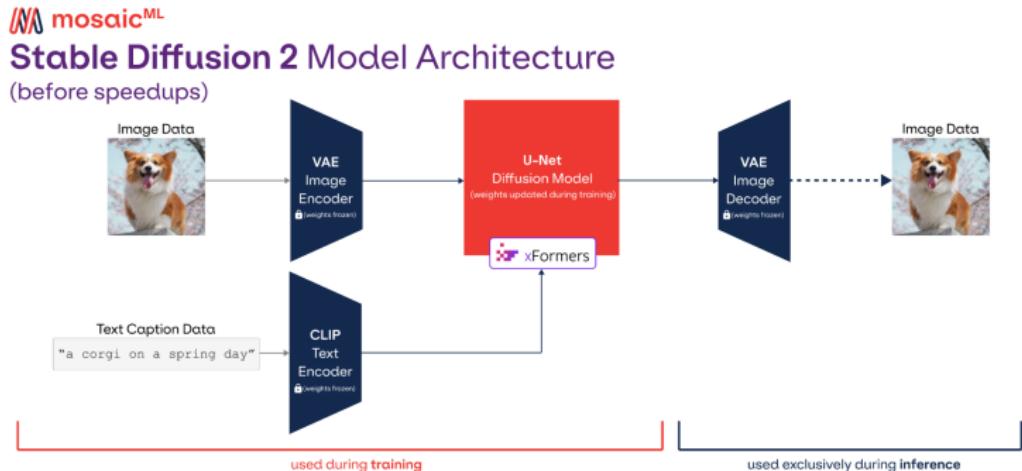


Рис.: Stable diffusion architecture

GAN: артефакты