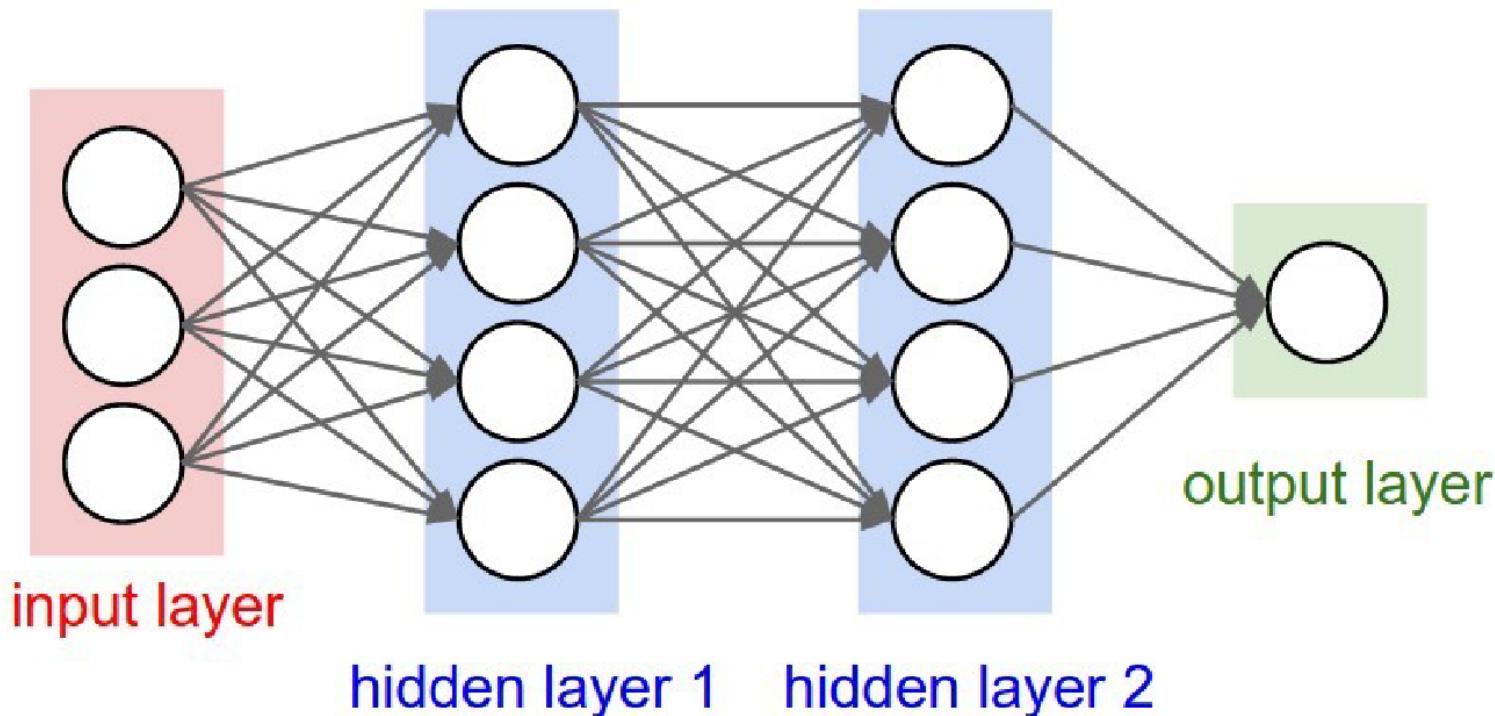


Convolutional Neural Network

Зубанов Виктор

Нейронные сети для распознавания изображений

- На вход подается изображение размера (width, height, depth)

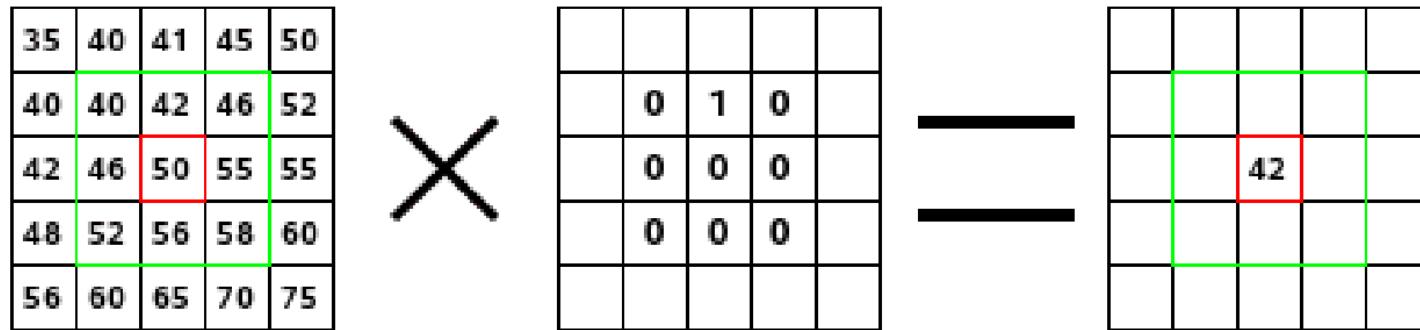


Чем плохи обычные сети

- Большой размер (для изображения 200x200x3 и одного нейрона будет 120000 связей)
- Никак не учитывается структура входных данных (изображение)
- Все это приводит к переобучению

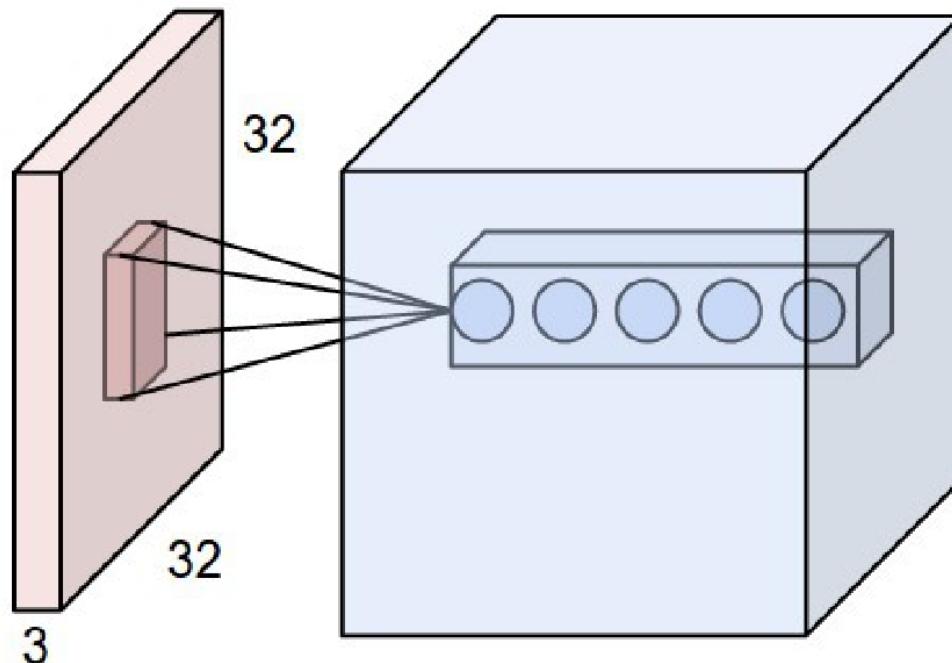
Свертка

- Ядро свертки – матрица малого размера (обычно $3 \times 3 \times d$ или $5 \times 5 \times d$, где d – глубина)
- Свертка изображения – для каждого пикселя применяем линейную комбинацию из его окрестности с весами ядра
- На входе размер (width, height, depth)
- На выходе – (width, height)



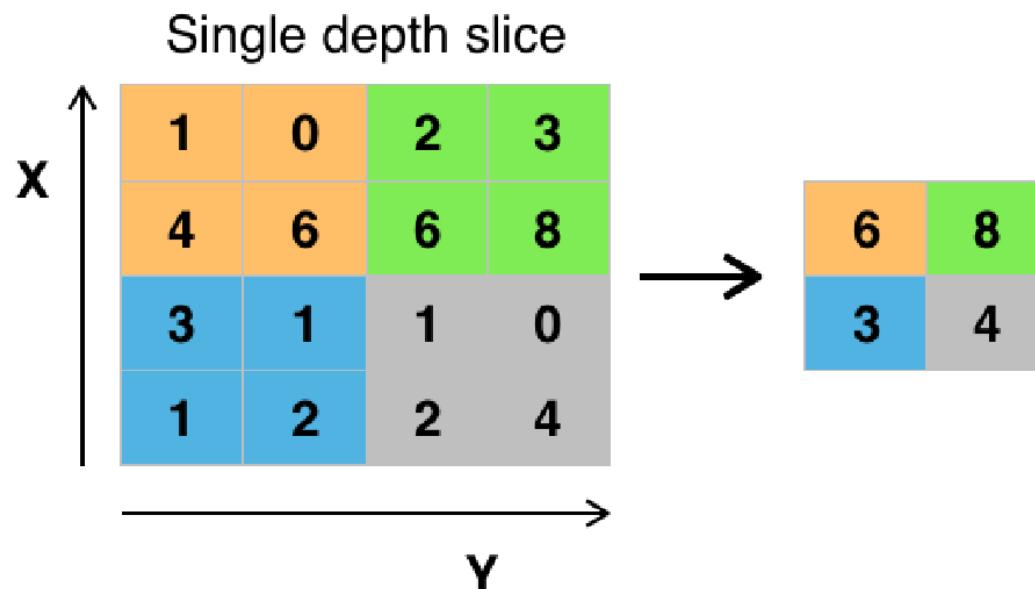
Сверточный слой

- Веса сверточных матриц общие для всех позиций
- Применяем свертку для k различных фильтров
- Количество параметров: $k \times$ размер сверточной матрицы



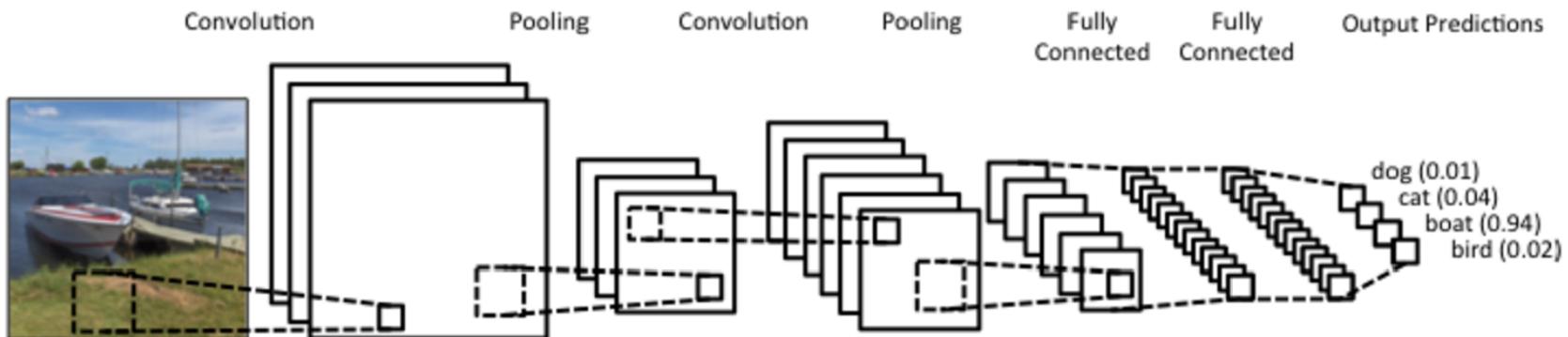
Max pooling слой

- Группа пикселей (например 2x2) уплотняется до одного
- Обычно используется функция максимума
- Уменьшает количество параметров, помогая не переобучиться



Архитектура CNN

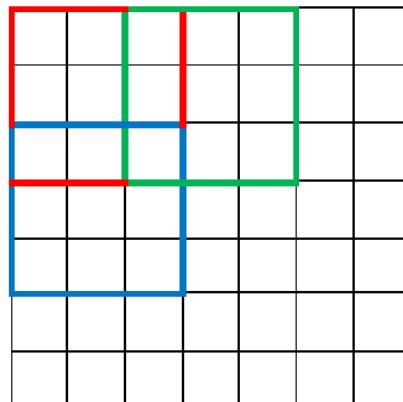
- Чередование сверточных и пулинг слоев
- Полносвязная нейросеть в конце



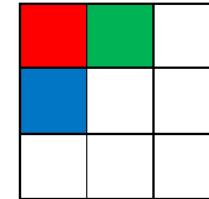
Гиперпараметры

- Количество слоев и их типы
- Для сверточного и пулинга
 - 1) Размер фильтра
 - 2) Сдвиг фильтра
- Для сверточного
 - 1) Количество фильтров

7 x 7 Input Volume

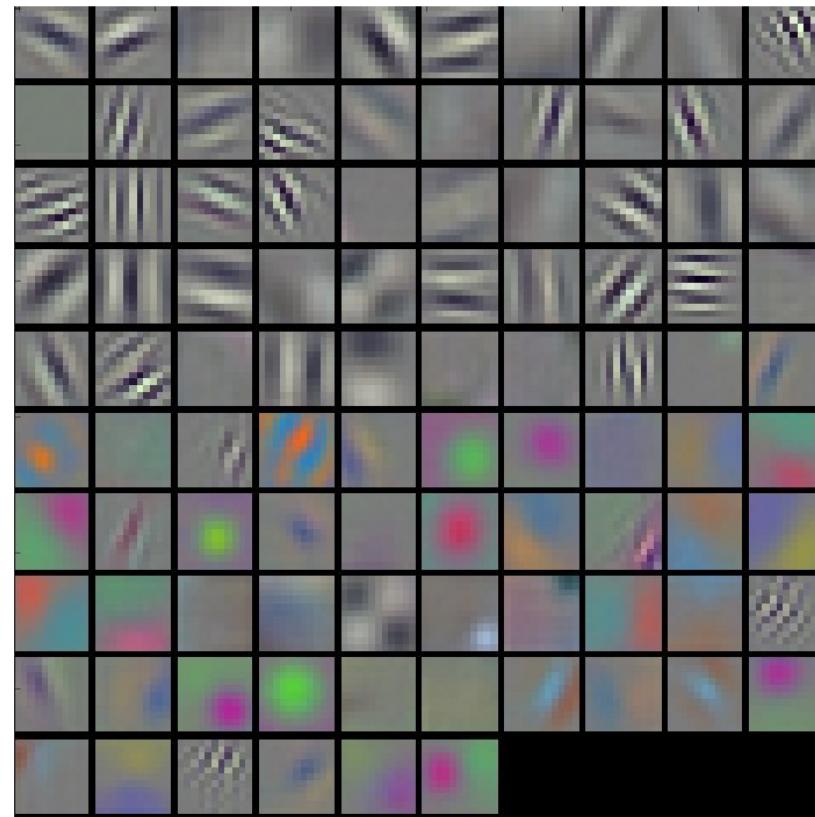


3 x 3 Output Volume



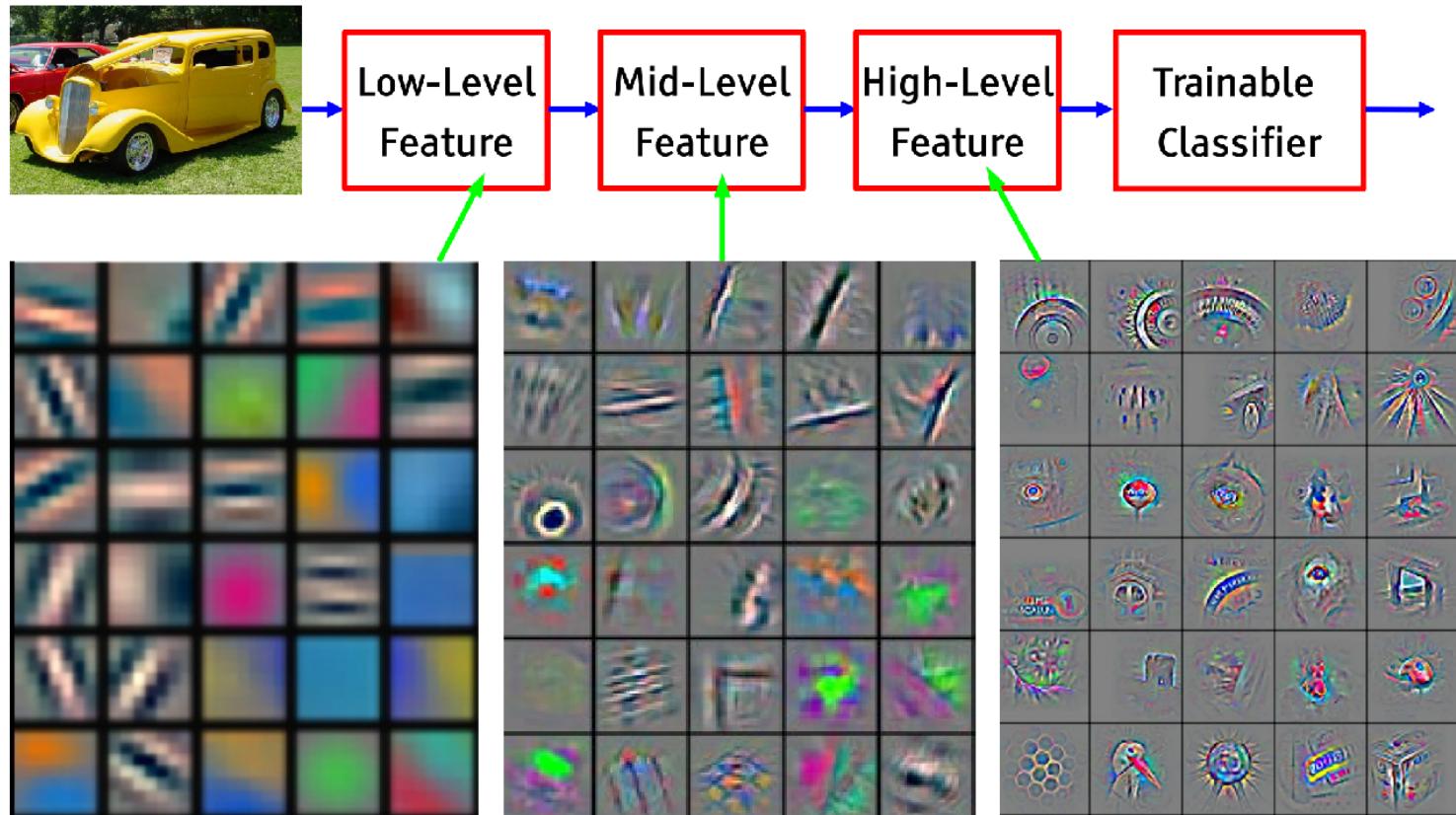
О фильтрах свертки

- Обучаясь фильтры свертки начинают реагировать на конкретные признаки
1 слой свертки



О фильтрах свертки

- Чем глубже слой, тем признаки становятся более абстрактными



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

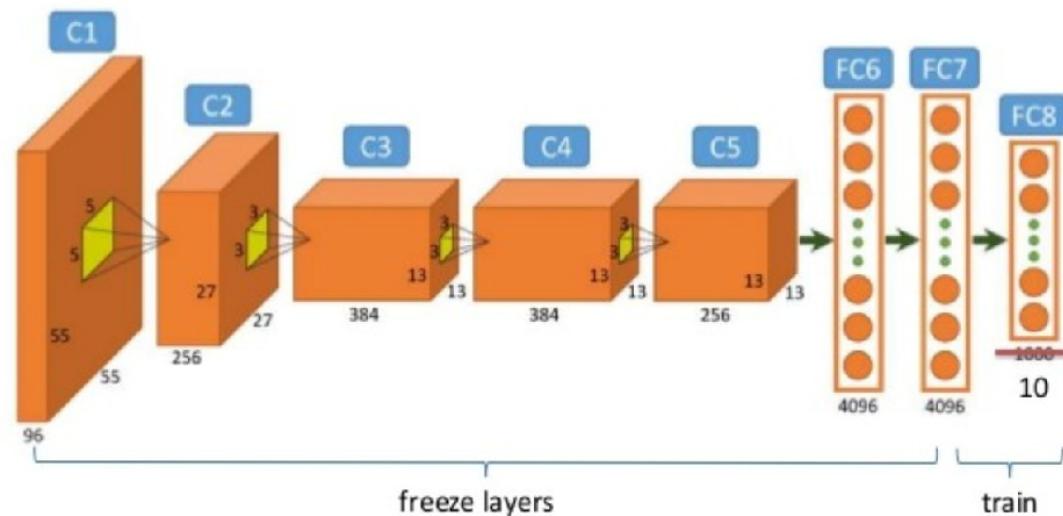
Извлечение признаков

- Нейроны на глубоких слоях реагируют на смысловые признаки
- Берем выход слоя в нейросети и с помощью него описываем объект

Дообучение (Fine-tuning)

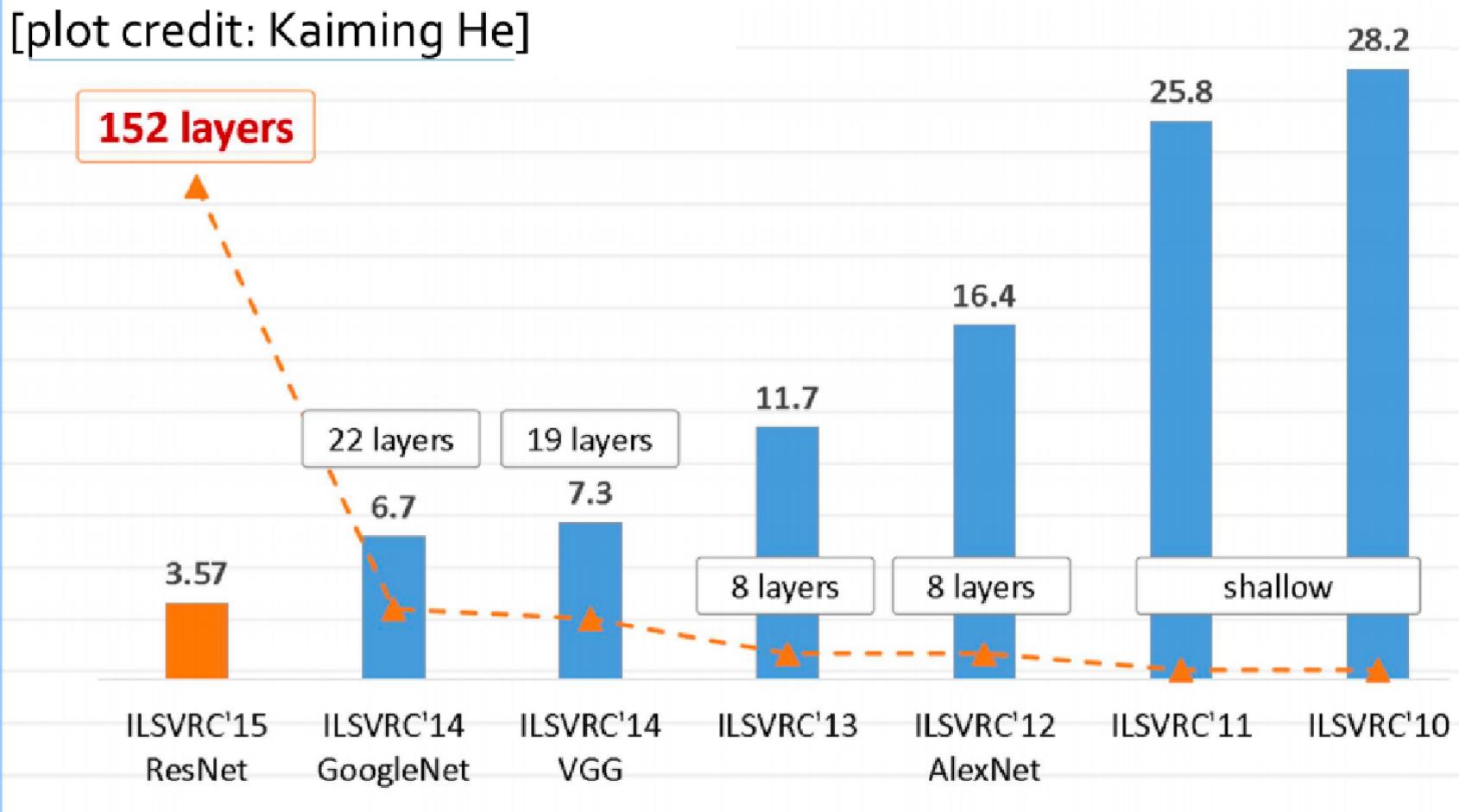
- Берем обученную нейросеть, удаляем несколько последних слоев
- Достраиваем нейросеть на полученных признаках
- Помогает избежать переобучения, когда данных мало

Fine-tuning Pretrained Network



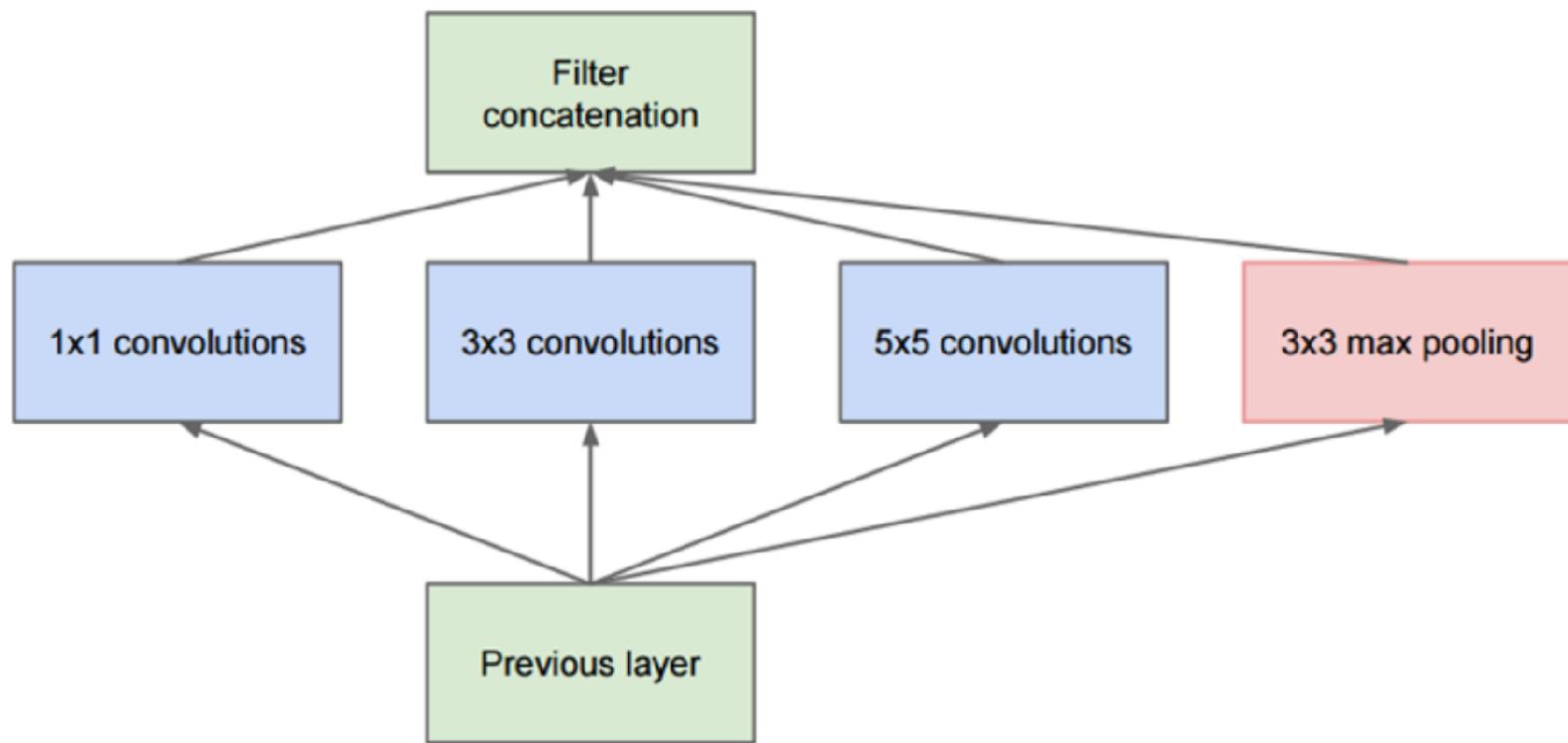
Известные архитектуры

[plot credit: Kaiming He]



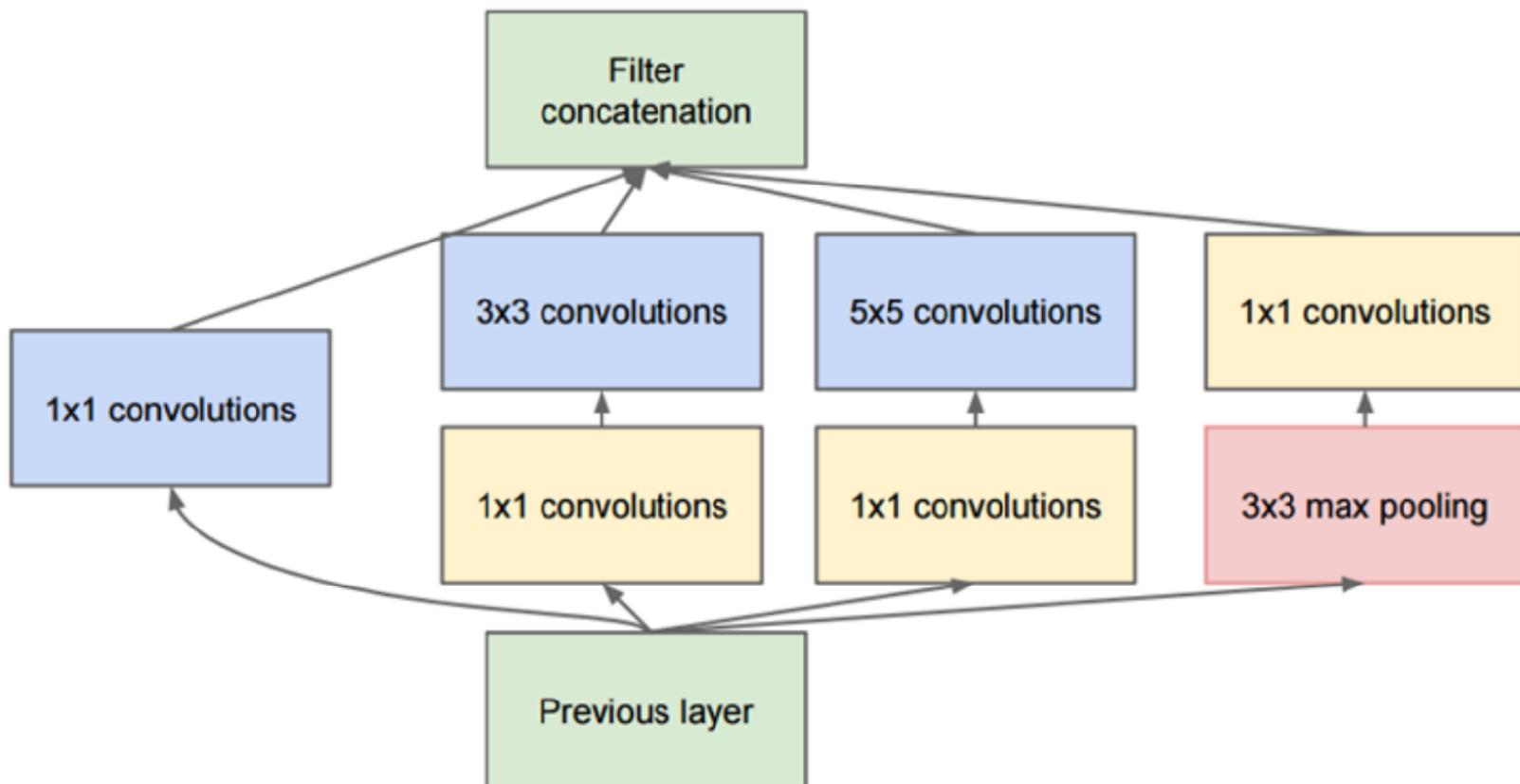
GoogleNet (Inception)

- Основная идея – параллельно сделать различные операции
- В простом виде много параметров



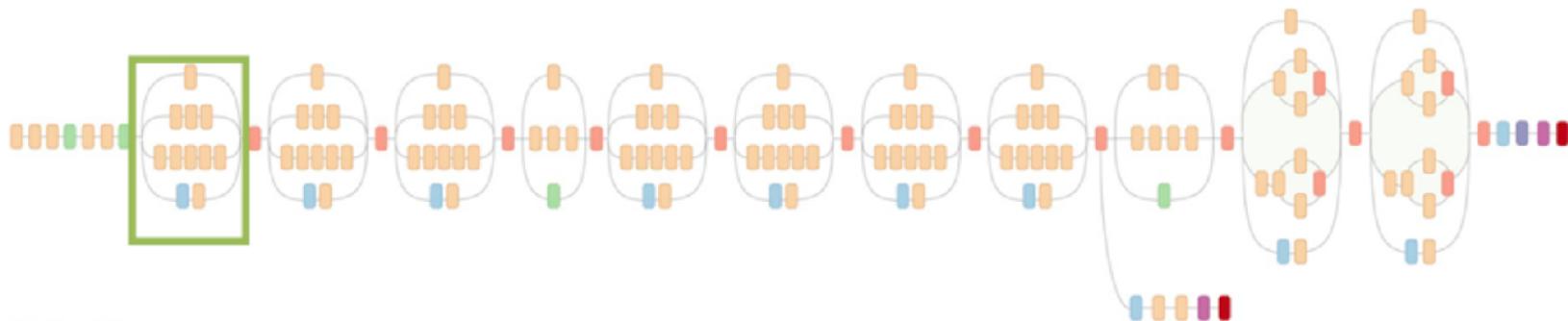
GoogleNet (Inception)

- Чтобы уменьшить количество параметров добавим фильтры 1×1



GoogleNet (Inception)

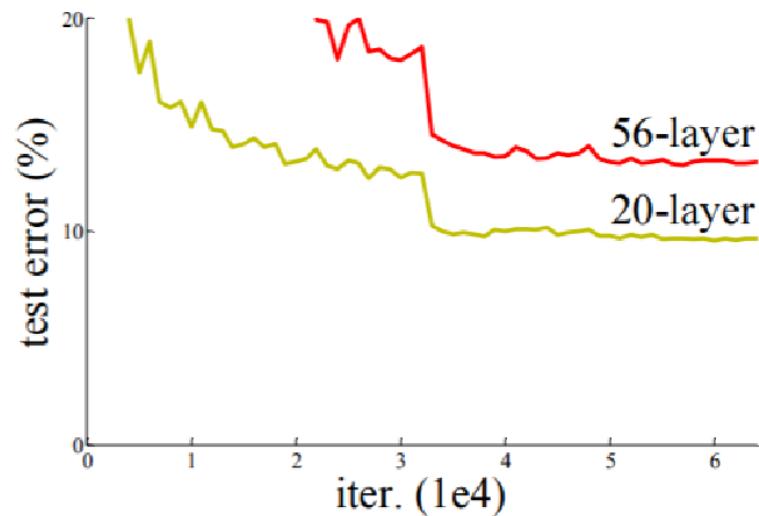
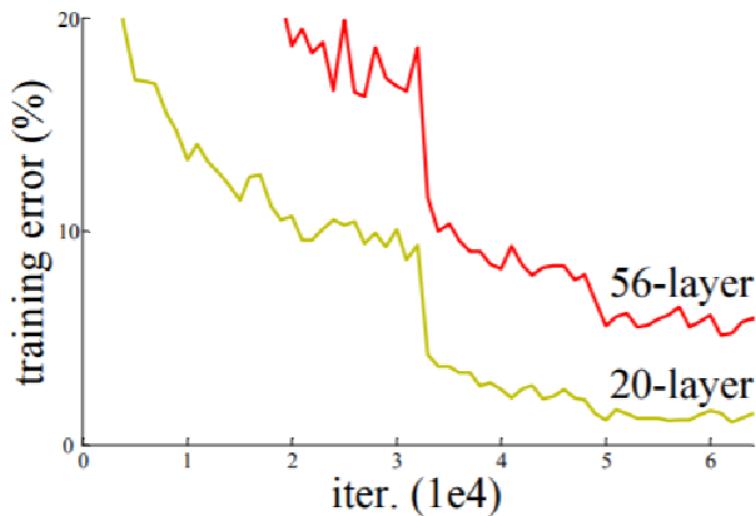
- 9 Inception блоков, в сумме более 100 слоев
- Параметров в 12 раз меньше, чем в AlexNet
- Точность 6.7%



Orange	Convolution
Blue	AvgPool
Green	MaxPool
Red	Concat
Purple	Dropout
Pink	Fully connected
Dark Red	Softmax

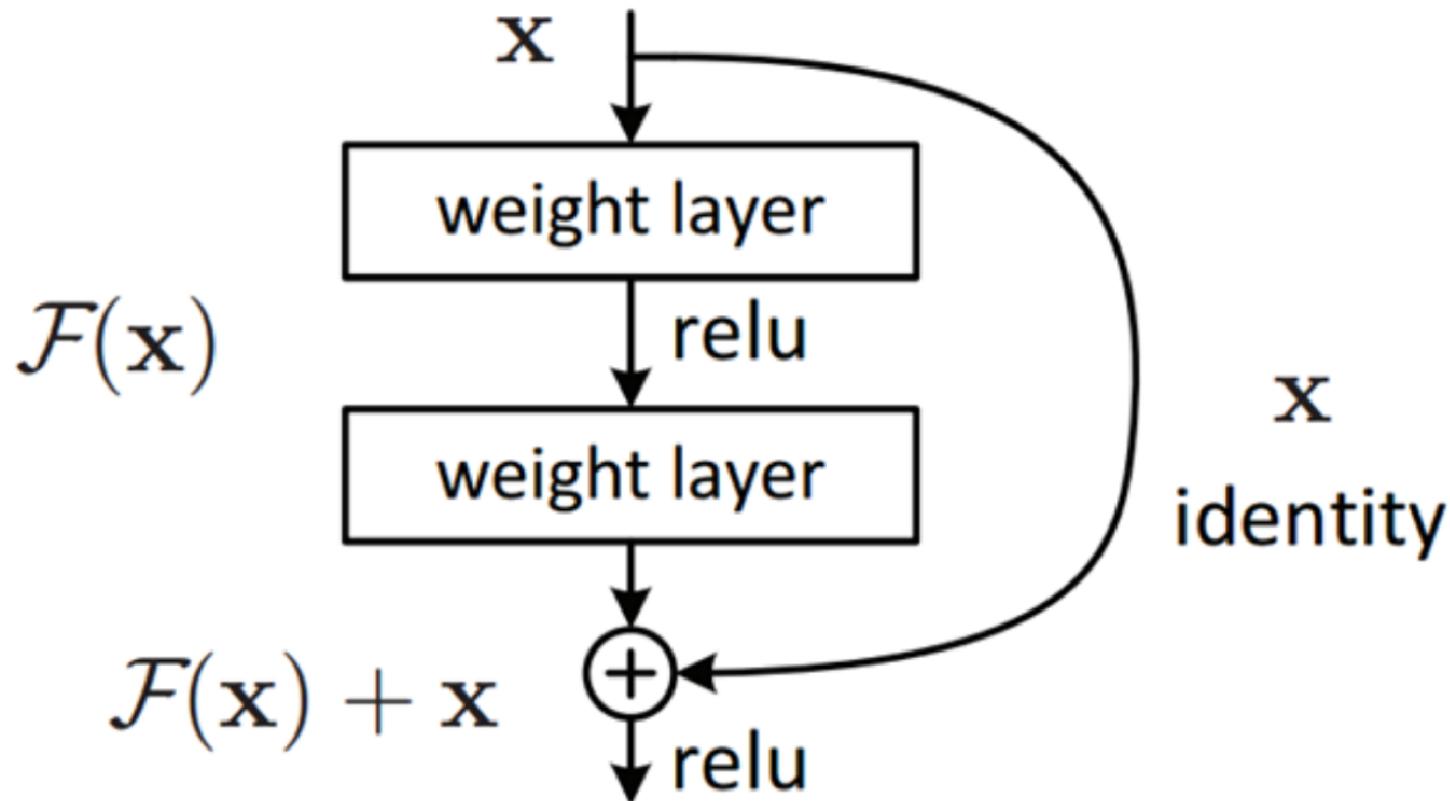
Residual Network (ResNet)

- В CNN при увеличении количества слоев точность начинает падать
- Проблема не в самих глубоких сетях, а в неумении обучать их



Residual Block

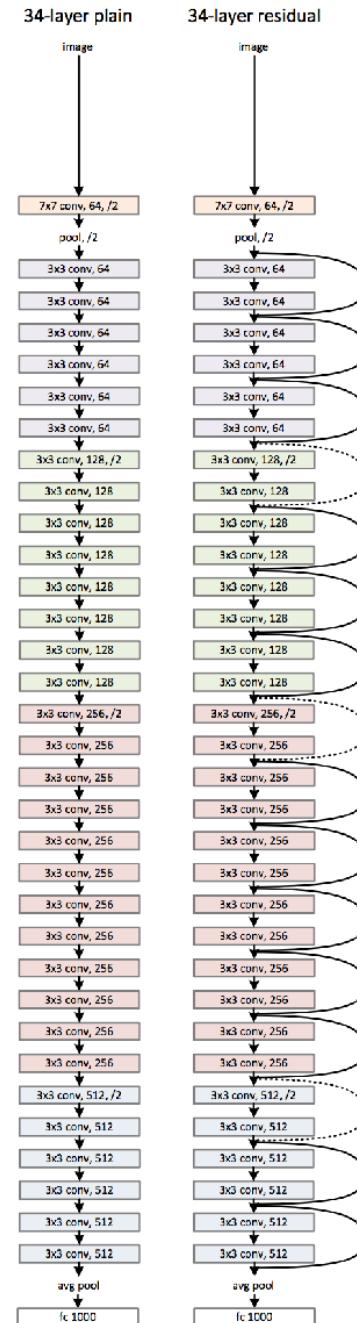
- Будем вычислять не функцию $F(x)$, а изменение x , то есть $H(x) = F(x) + x$



A residual block

Residual Network (ResNet)

- 152 слоя
- Точность 3.57%
- Пробовали сеть с 1202 слоями, но результат был хуже, предположительно, из-за переобучения



Ссылки

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

<https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html>

<https://arxiv.org/pdf/1512.03385.pdf>

https://www.youtube.com/watch?v=XuI4usfwbhU&index=6&list=PLbwKcm5vdiSYL_yEwQ6JIICBA4dMtHNxo