А.Крюков

Машинное обучение в астрофизике

Лекция 4

Светрочные сети

В предыдущих сериях

- Лекция 1. Что такое машинное обучение и сферы его применения в науке.
- Лекция 2. Основные понятия.
- Лекция 3. Метод обратного распространения ошибок.
- Лекция 4. Деревья решений.

- Лекция 5. Сверточные сети.

Анализ изображений

- Многие данные, в том числе научные, представляют из себя изображения или могут быть представлены в виде изображений.
  - Не обязательно это должны быть 2-х мерные картинки.
- Примеры научных изображений:
  - фотографии звездного неба;
  - данные с трековых детекторов частиц (3-х мерные «изображения»);
  - данные черенковских атмосферных телескопов;
  - поля температур в метеорологии

Анализ изображений

- Задачи анализа изображений
  - поиск и выделение объектов
    - например, треков частиц в детекторах.
  - извлечение физических характеристик явления
    - например, энергия космических лучей.
  - моделирование изображений
    - например, как замена Монте-Карло моделирования явлений.
  - предсказание поведения системы
    - например, фазовые переходы в физике твердого тела
  - чистка изображений от помех и/или восстановление недостающих участков
    - например, чистка случайных помех на фотографиях с длиной выдержкой в астрономии.

Полносвязанные сети и задачи анализа изображений

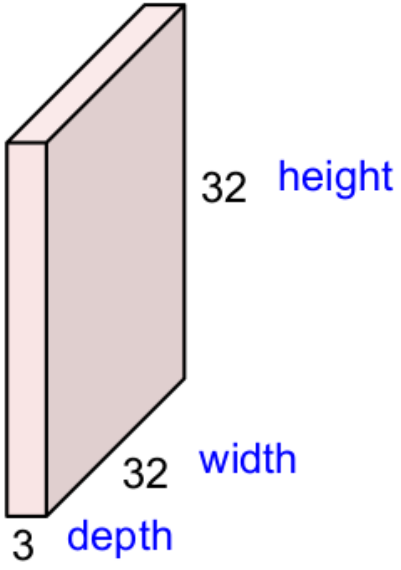
- Полносвязанные сети игнорируют информацию о расстоянии между элементами изображений.
  - как правило, соседние элементы в изображении сильнее скорелированы, чем далекие.
- Полносвязанные сети не учитывают симметрии, присущие изображениям.
  - трансляционная инвариантность - параллельный сдвиг объекта на изображении
- Глубокие и большие полносвязанные сети имеют огромное количество параметров
  - требуют большой объем оперативной памяти, повышенные требования в ЦПУ
  - повышенные требования к процедуре обучения
    - проблема недообучения и переобучения.

Сверточные сети

- Частично, все эти проблемы удается решить при помощи сверточных нейронных сетей (Convutional neural network, CNN)
- Основная идея — это применить к отдельным частям изображения одну и туже небольшую матрицу — операция свертки.
- Достоинства:
  - требуется гораздо меньше параметров
  - учитывается трансляционная инвариантность
  - повышается устойчивость обучения
- Полносвязанные слои в нейронной сети по прежнему используются в качестве классификатора или регрессора

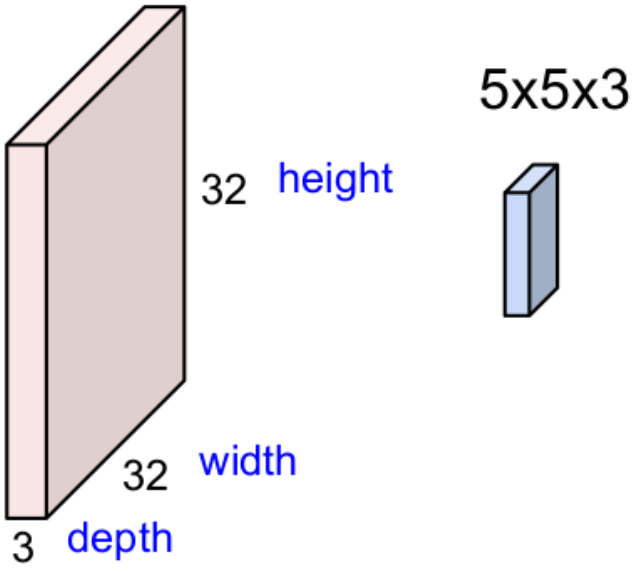
Сверточный слой

- Изображение, как правило имеет вид WxHxD матрицы, где W — ширина изображения в пикселях, H — высота, а D глубина. Например цветная VGA картинка имеет размер 640x480x3.
- Каждый пиксел — это либо целое, либо действительное число в некотором диапазоне



Сверточный слой

- Изображение, как правило имеет вид WxHxD матрицы, где W — ширина изображения в пикселях, H — высота, а D глубина. Например цветная VGA картинка имеет размер 640x480x3.
- Каждый пиксел — это либо целое, либо действительное число в некотором диапазоне
- Добавим фильтр, например 5x5x3, который мы будем двигать по изображению и выполнять операцию всертки.



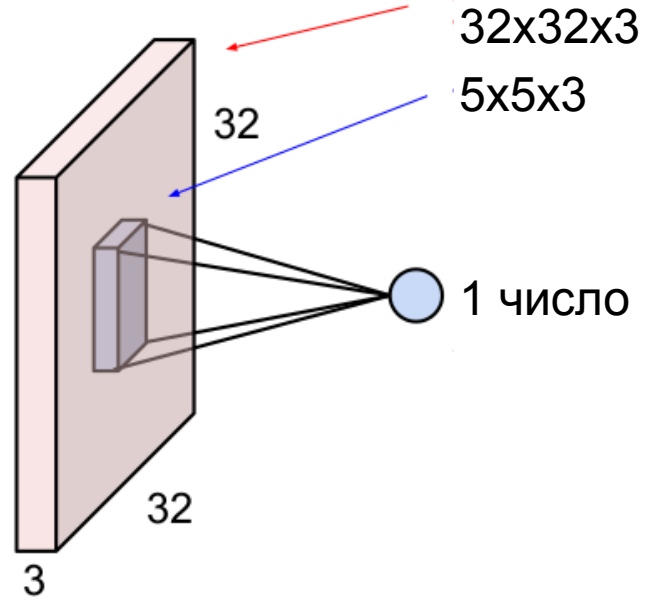
Сверточный слой

- Операция конволютивной свертки это скалярное произведение кусочка изображения и фильтра:

  $\mathbf{y=W^T*x+b}$

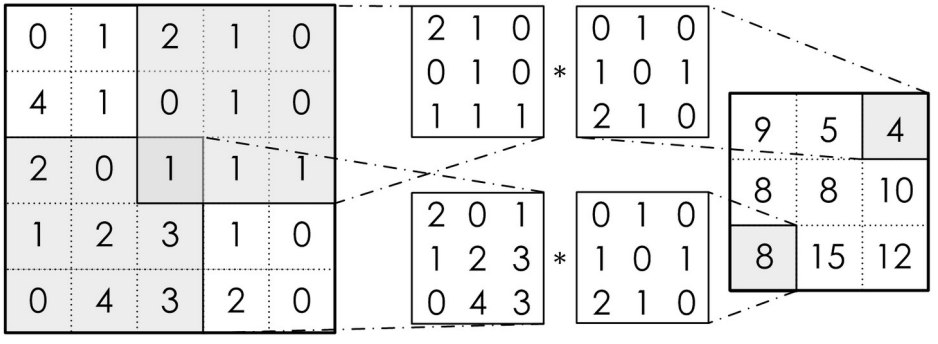
- Или покомпонентно

$$y^l_{i,j} = \sum_{-d<=a,b<=d} W_{a,b} x^l_{i+a,j+b}.$$



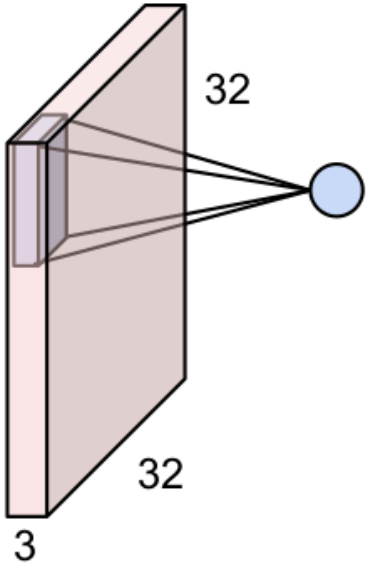
- Потом применяется нелинейн функция активации, которая почти всегда ReLU

$$z^l_{i,j} = h(y^l_{i,j}).$$



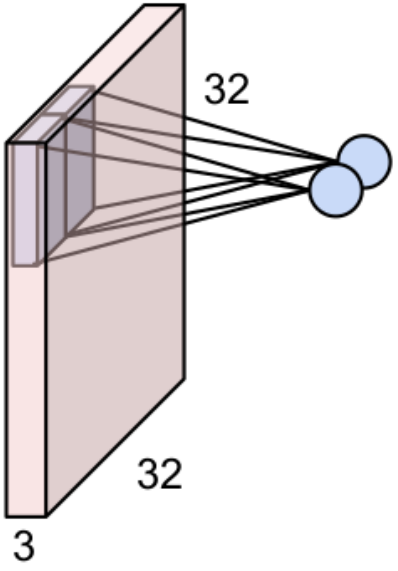
Сверточный слой

- Последовательно проходим по всему изображению



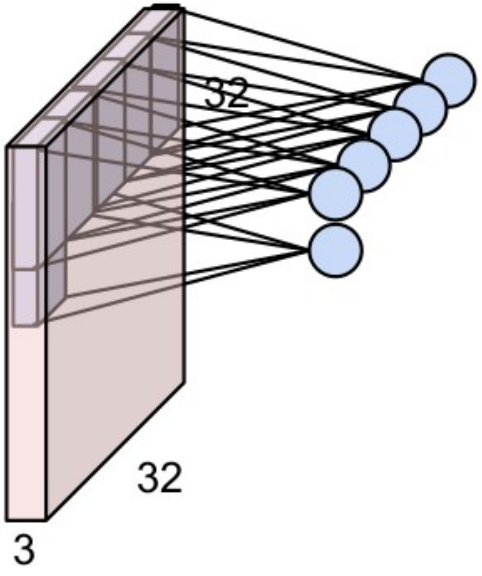
Сверточный слой

- Последовательно проходим по всему изображению



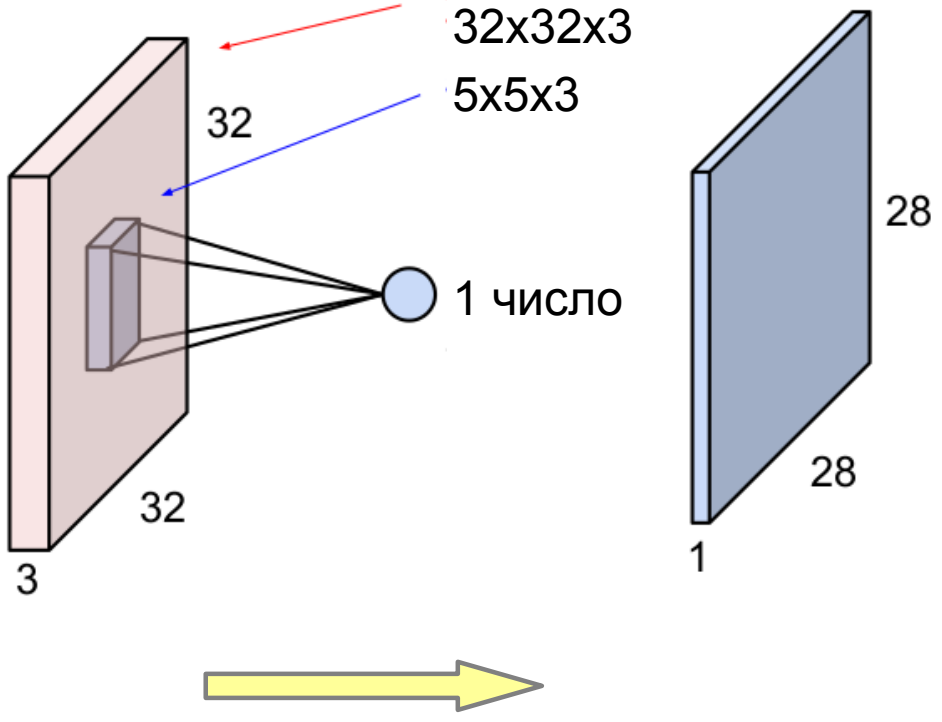
Сверточный слой

- Последовательно проходим по всему изображению



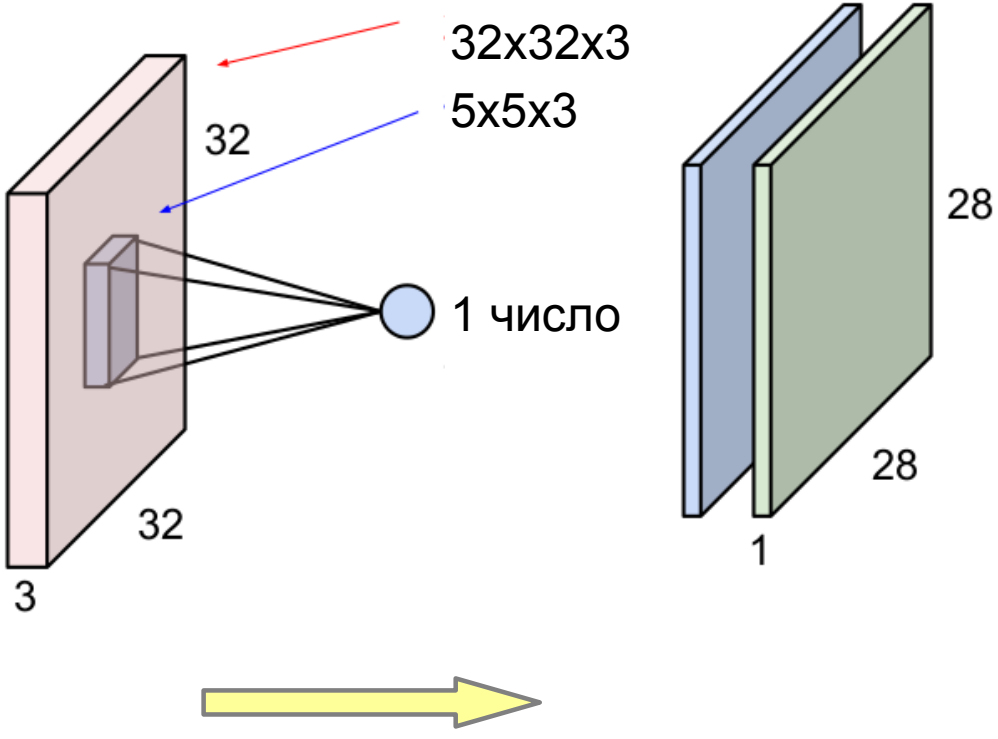
Сверточный слой

- Таким образом наше входное изображение отображается на более компактное изображение.



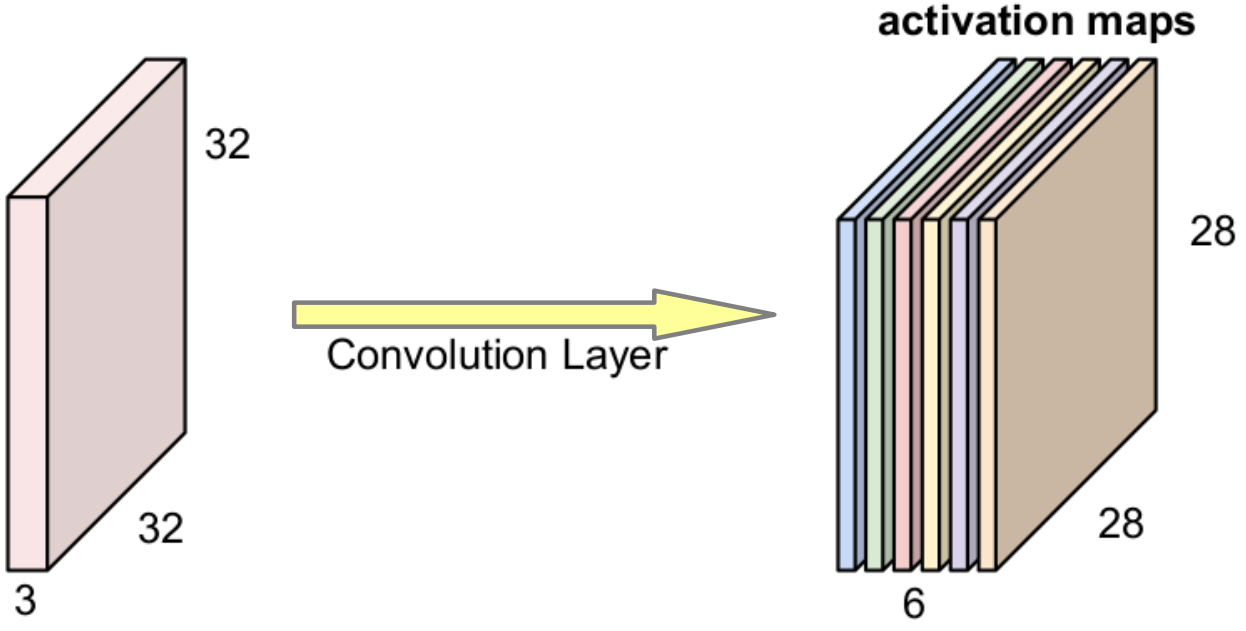
Сверточный слой

- Если мы возьмем два разных фильтра, то получим две разные карты активации.



Сверточный слой

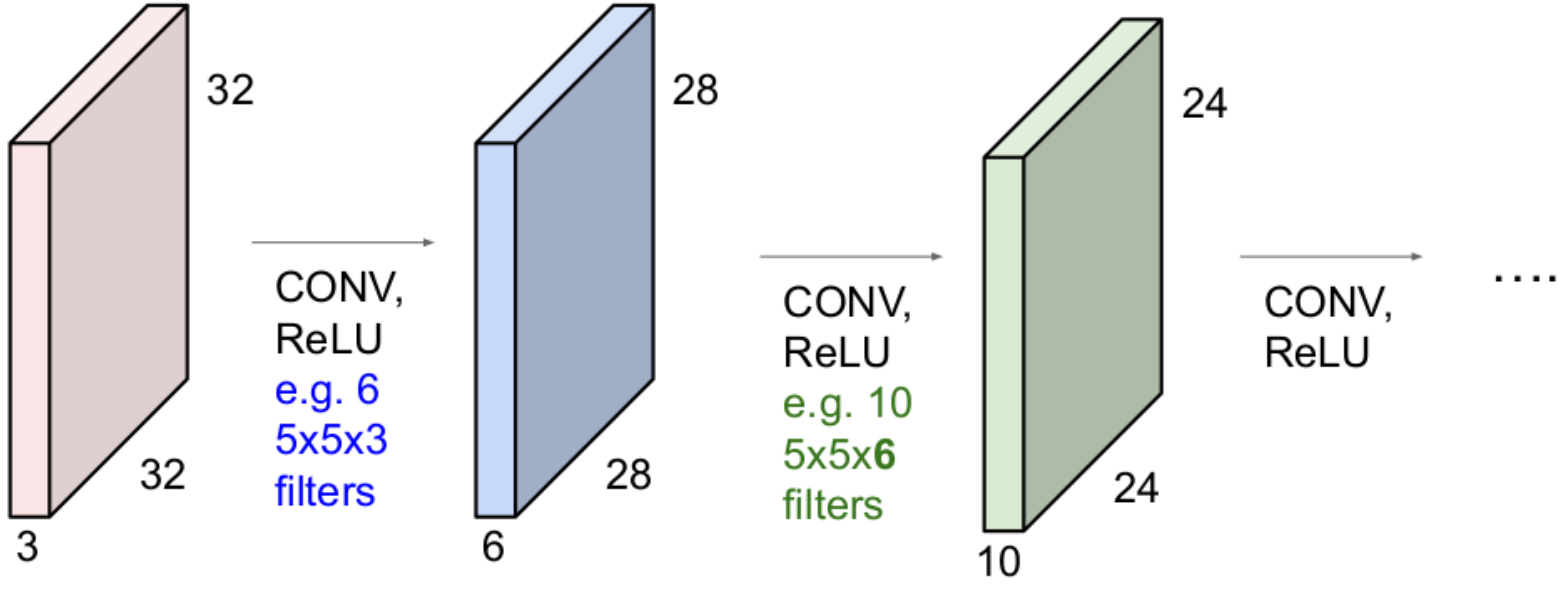
- Таких карт может быть любое количество.



Сверточный слой

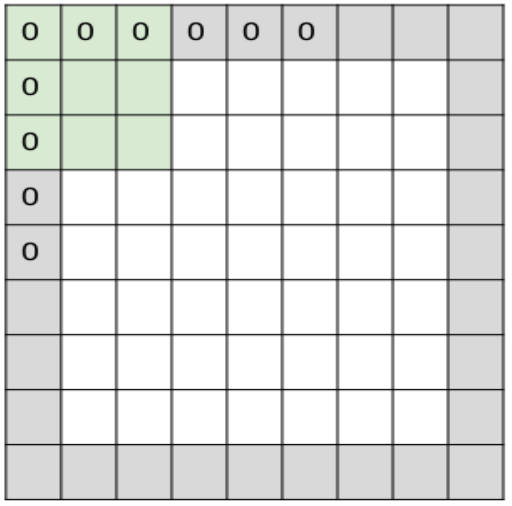
Что это нам дает?

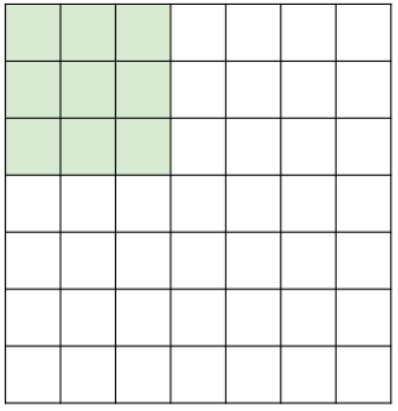
- Устойчивость к переносам (трансляционная инвариантность) и т.п.
- И радикально сокращает число весов.
- По сути, свёрточная структура – это такая радикальная форма регуляризации



Сверточный слой

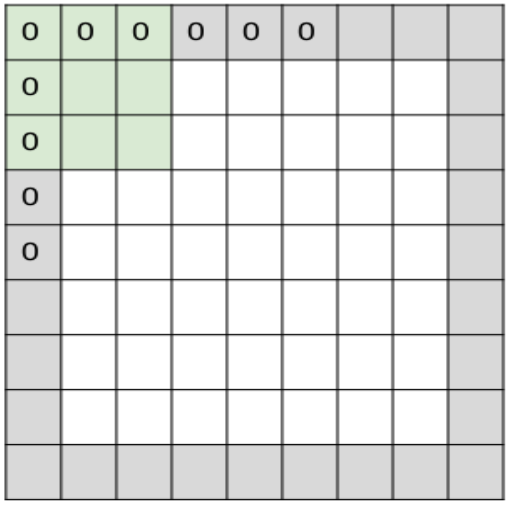
- Как сохранить размер картинки?
  - Добавим дополнительные нулевые пиксели вокруг - padding
  - Это позволит посчитать свертки и для граничных пикселей изображения.
- Какие еще есть варианты работы с фильтрами?
  - Сдвигать фильтр не на 1 пиксель, а на несколько — stride

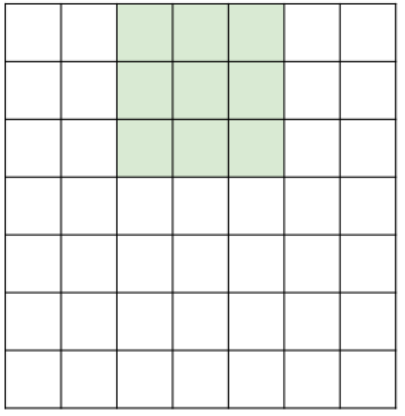




Сверточный слой

- Как сохранить размер картинки?
  - Добавим дополнительные нулевые пиксели вокруг - padding
  - Это позволит посчитать свертки и для граничных пикселей изображения.
- Какие еще есть варианты работы с фильтрами?
  - Сдвигать фильтр не на 1 пиксель, а на несколько — stride

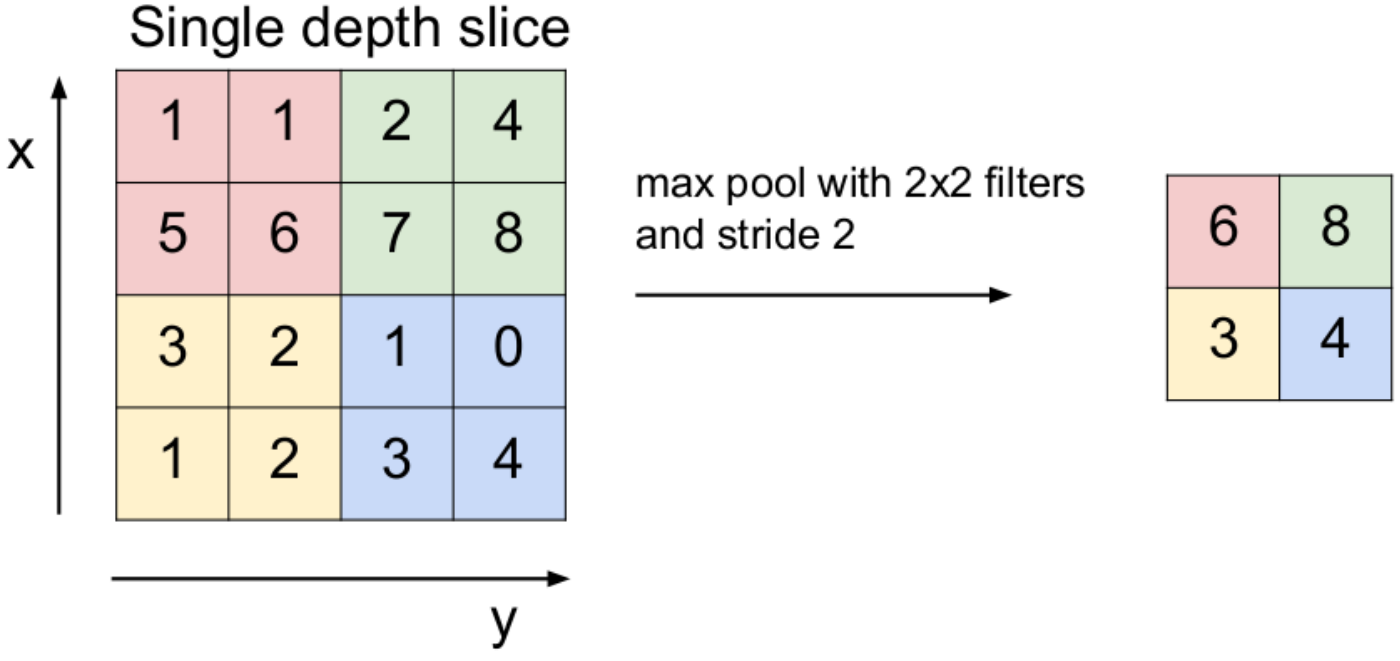




Max pooling

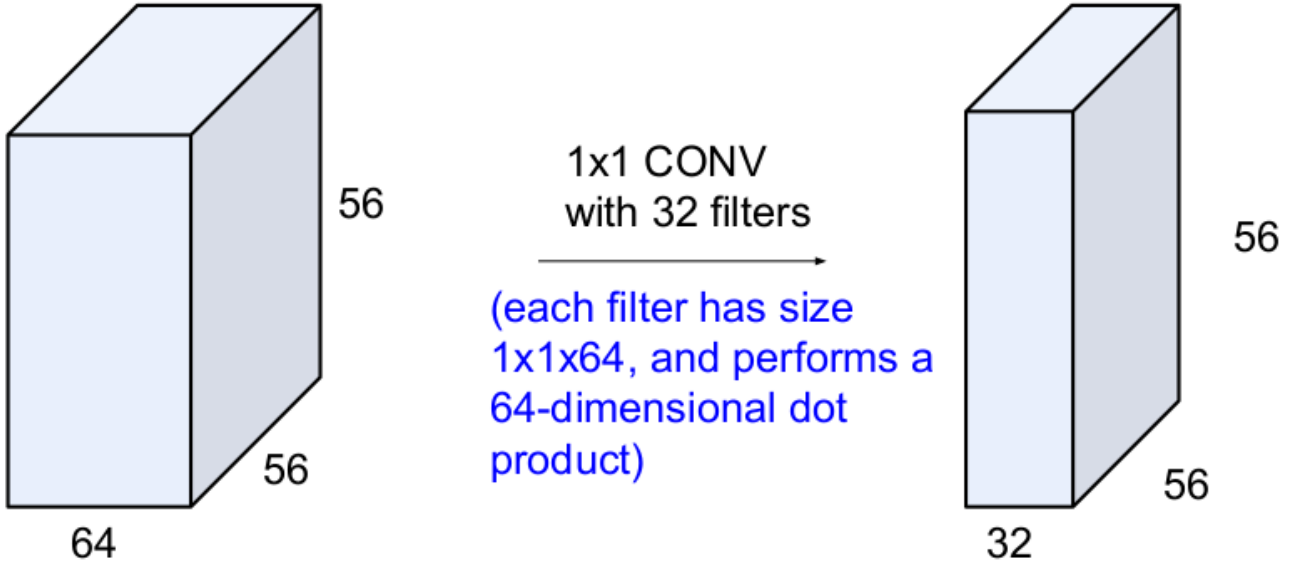
- Обычно после сверточного слоя идет слой «max pooling».
  - Его задача уменьшить размер изображения:

$$x_{i,j}^{l+1} = \max_{-d<=a,b<=d} z_{i+a,j+b}^{l}.$$



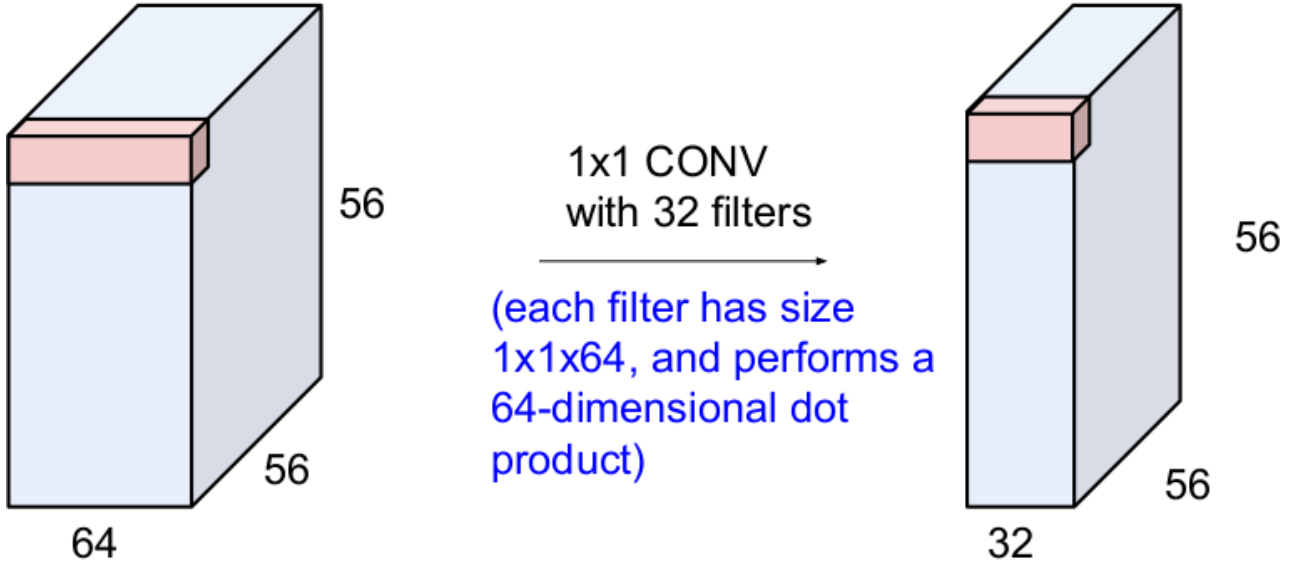
1x1 свертки

- Задача уменьшения размера изображения может быть решена с помощью 1x1 свертки



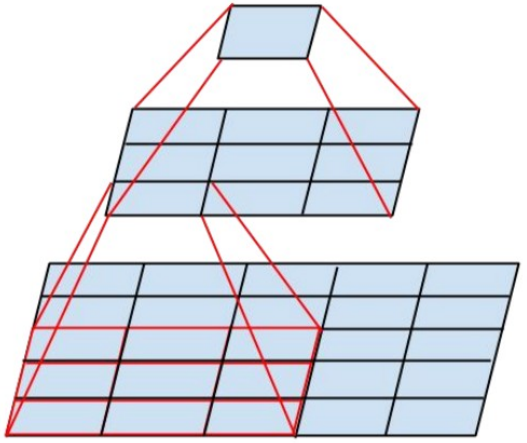
1x1 свертки

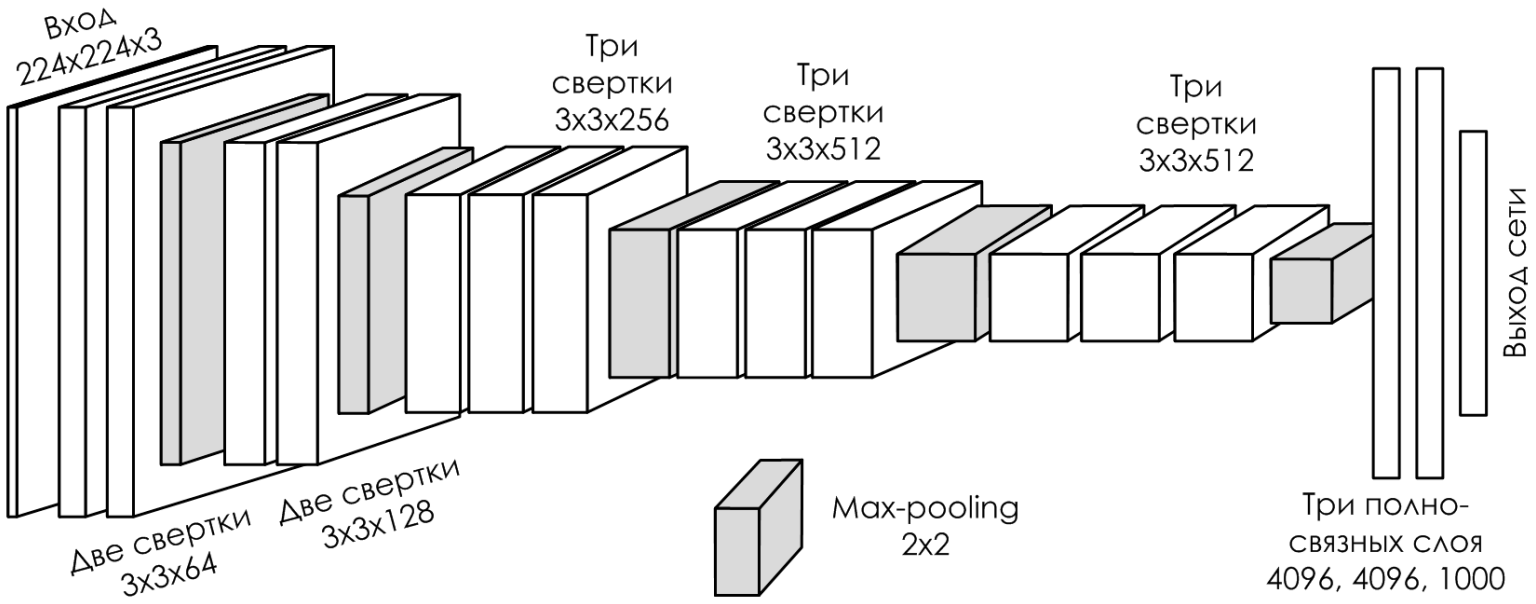
- Задача уменьшения размера изображения может быть решена с помощью 1x1 свертки



Сверточные архитектуры. VGG

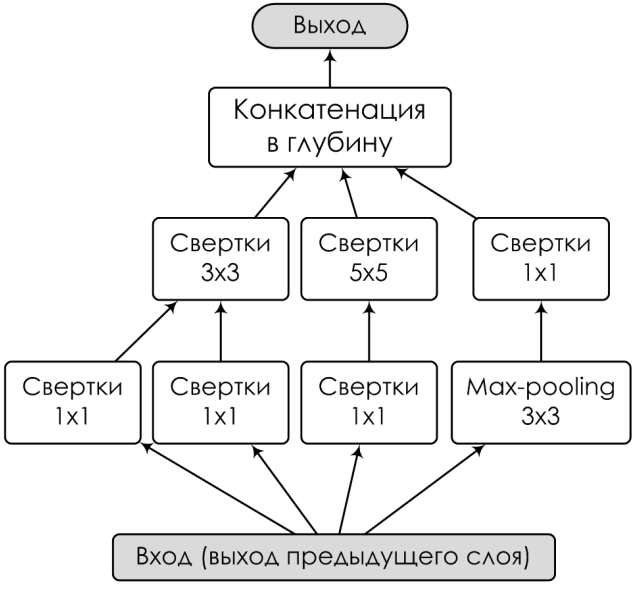
- VGG (Oxford Visual Geometry Group): большие свёртки представляются как комбинации свёрток 3 × 3.
  - снижает число весов, но делает сеть глубже.



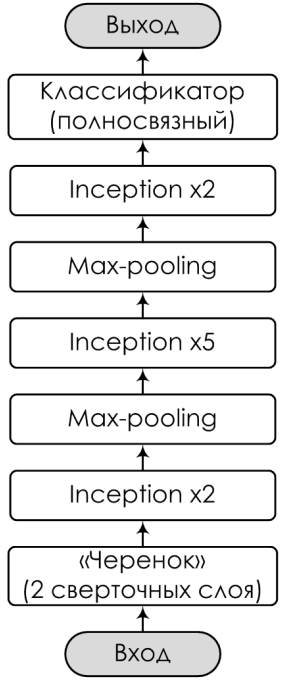


Сверточные архитектуры. Inception

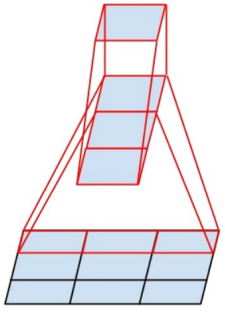
- Inception, разработанная командой GoogLeNet: “сеть в сети” (network in network)
  - сократить число весов ещё больше
  - сократить дополнительные классификаторы
- Во второй версии (2015) свёртки n×n заменили на комбинации свёрток n×1 и 1×n



(а)



(б)



Сверточные архитектуры. RESNET

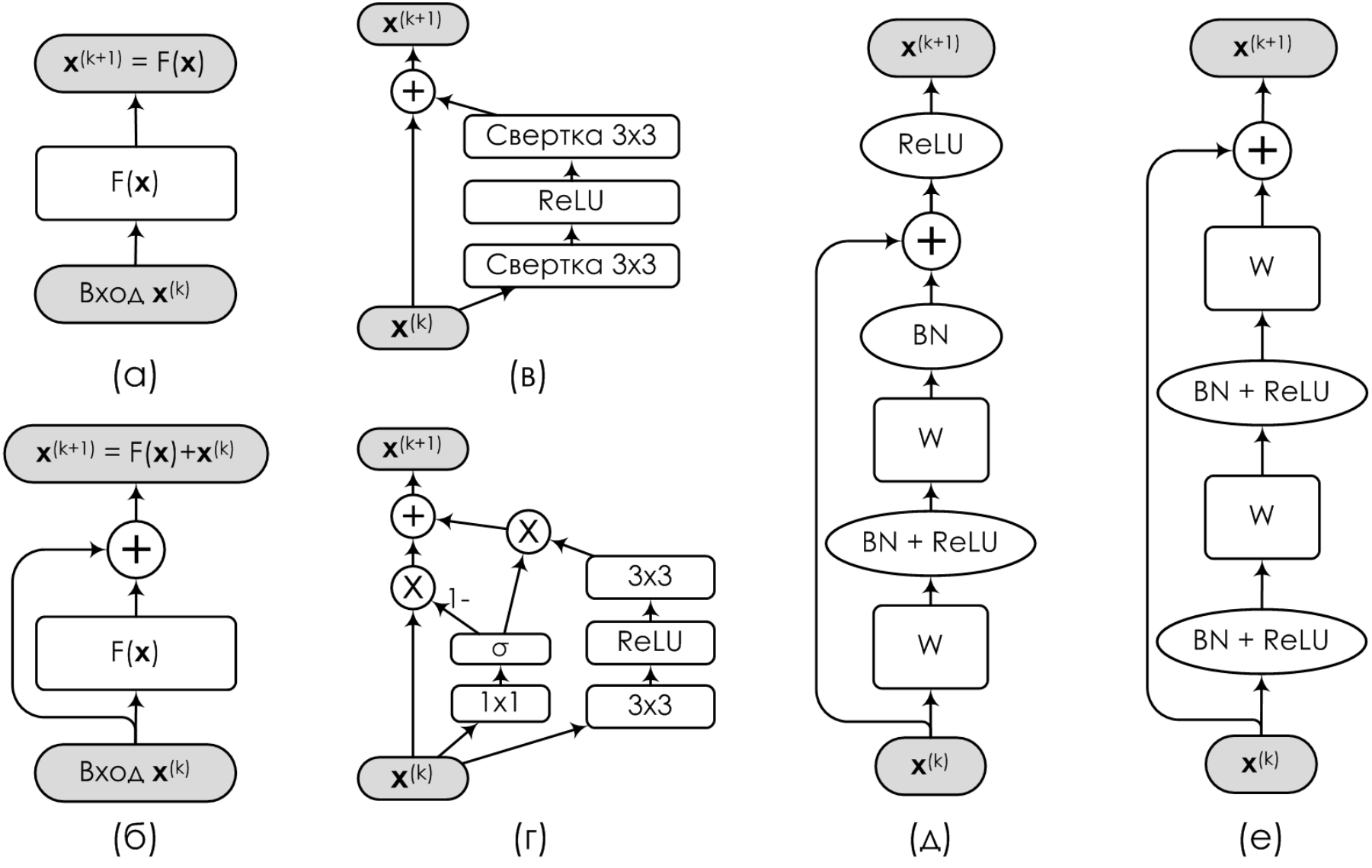
- Остаточное обучение (residual learning):
  - будем обучать разности между очередным уровнем и предыдущим.
  - Тогда градиенты смогут беспрепятственно проходить дальше.
- Функция, реализуемая остаточным блоком, выглядит как

$$\mathbf{y}^{(k)} = F(\mathbf{x}^{(k)}) + \mathbf{x}^{(k)},$$

- где $\mathbf{x}^{(k)}$ — входной вектор слоя k, $\mathbf{F}(\mathbf{x}^{(k)})$ — функция, которую вычисляет слой нейронов, а $\mathbf{y}^{(k)}$ — выход остаточного блока, который потом станет входом следующего слоя $\mathbf{x}^{(k+1)}$.
- Градиент будет проходить через этот блок беспрепятственно и не будет затухать:

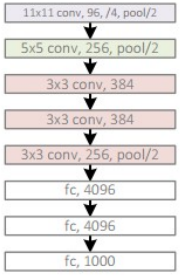
$$\frac{\partial \mathbf{y}^{(k)}}{\partial \mathbf{x}^{(k)}} = 1 + \frac{\partial F(\mathbf{x}^{(k)})}{\partial \mathbf{x}^{(k)}}.$$

Варианты RESNET

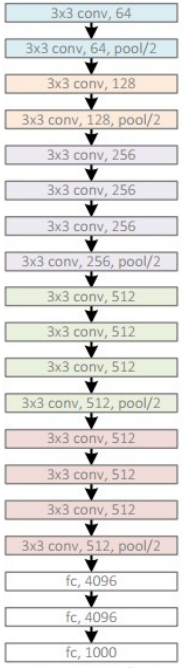


Рост глубины сверточных сетей

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

Материалы лекций:

https://theory.sinp.msu.ru/doku.php/ml_lectures