

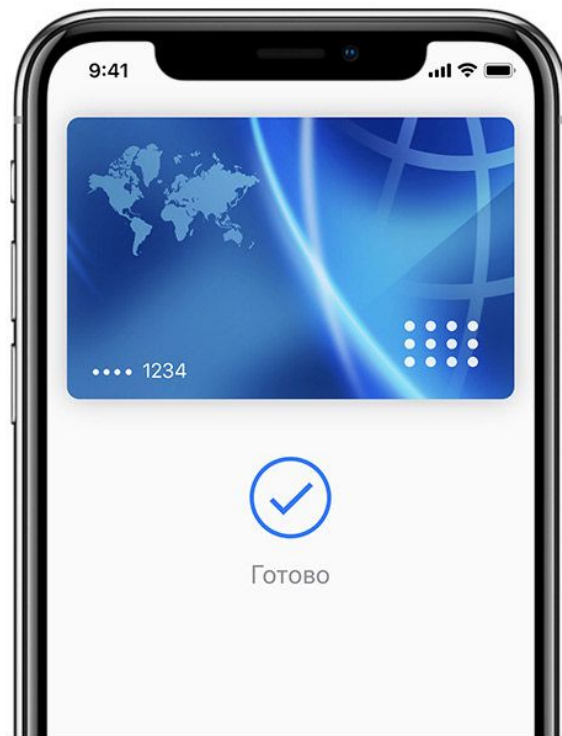
Введение в нейронные сети

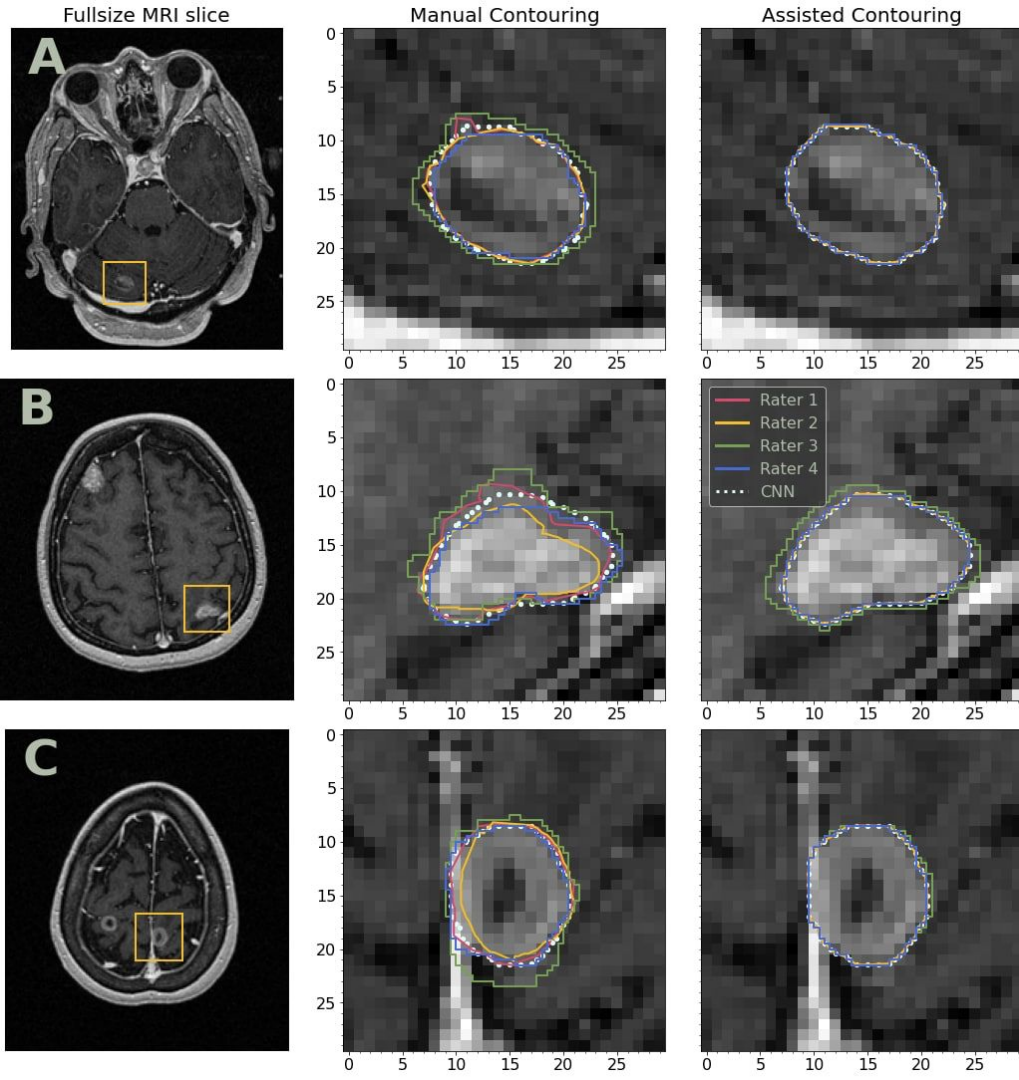
Лекция 3. Введение в сверточные НС.

Сегодня нас ждет...

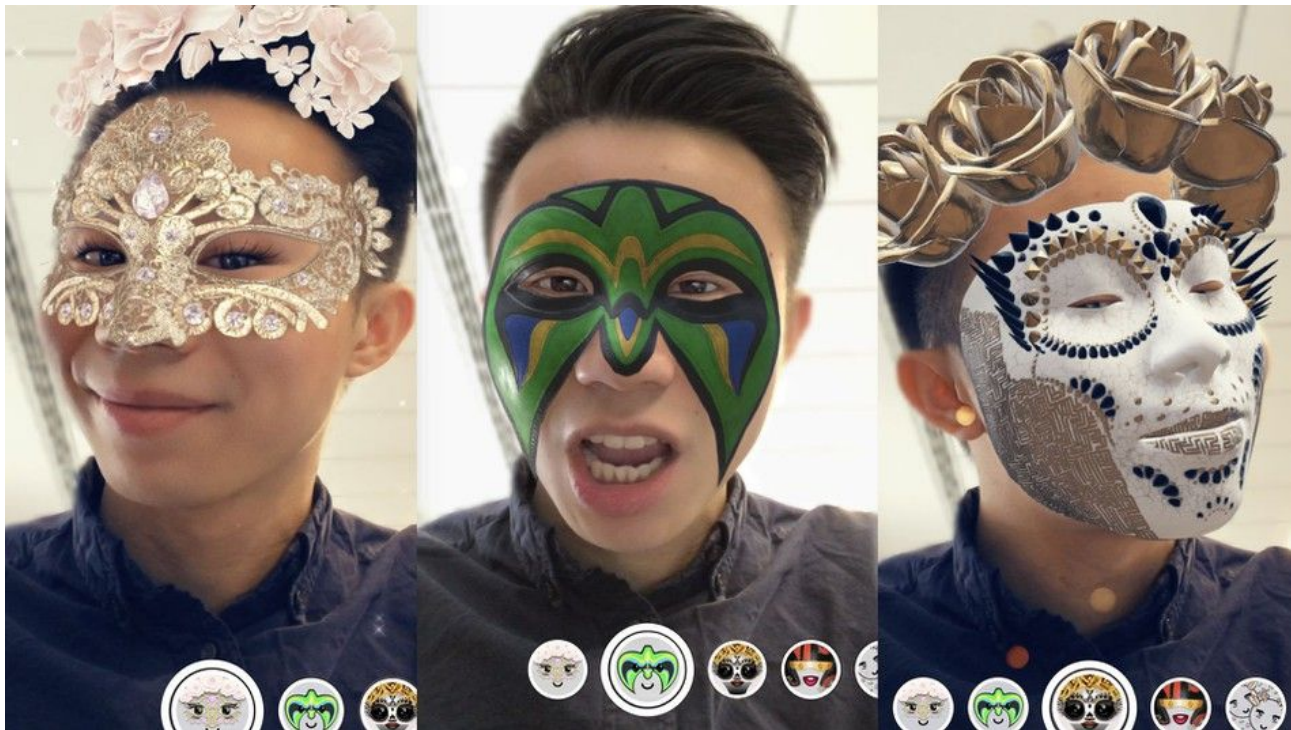
- Немного о картинках
- Сверточный слой, основные свойства и причины появления
- BatchNorm2D, Pooling
- LeNet
- Высокопроизводительные вычисления
- Реализация сверточной сети на Keras

Что можно делать с картинками?

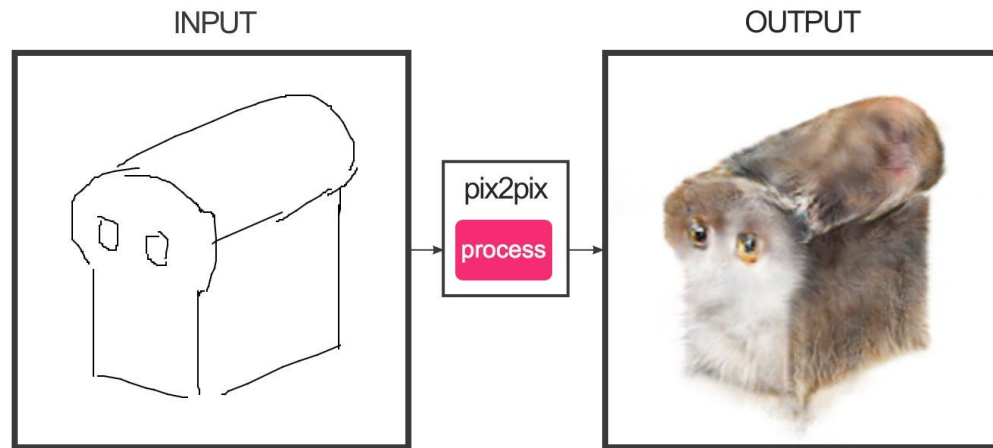
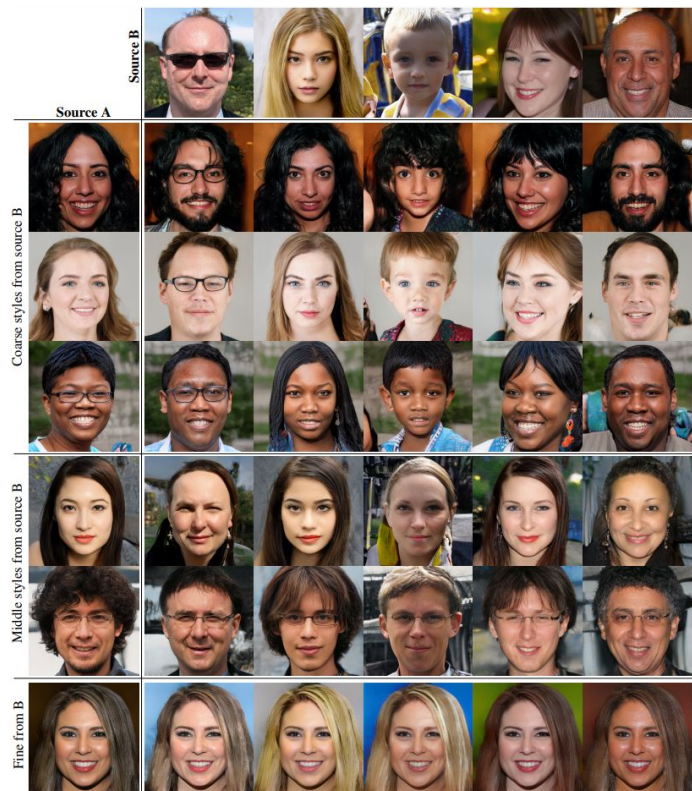




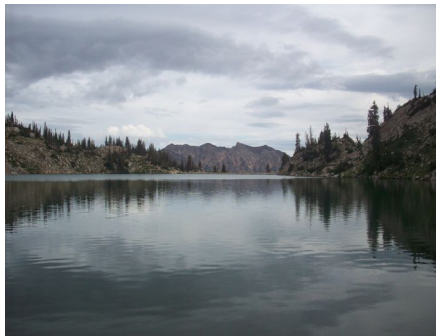
Что можно делать с картинками?



Что можно делать с картинками?



Что можно делать с картинками?



+



=



Что можно делать с картинками?



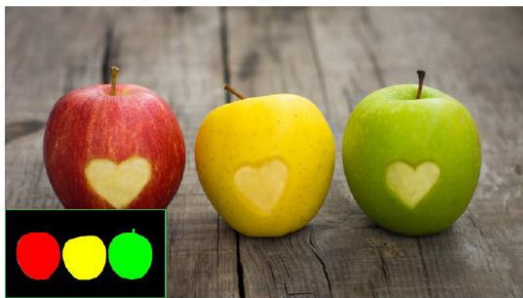
(a) Input image



(b) Reference image



(c) Our result



(d) Input image

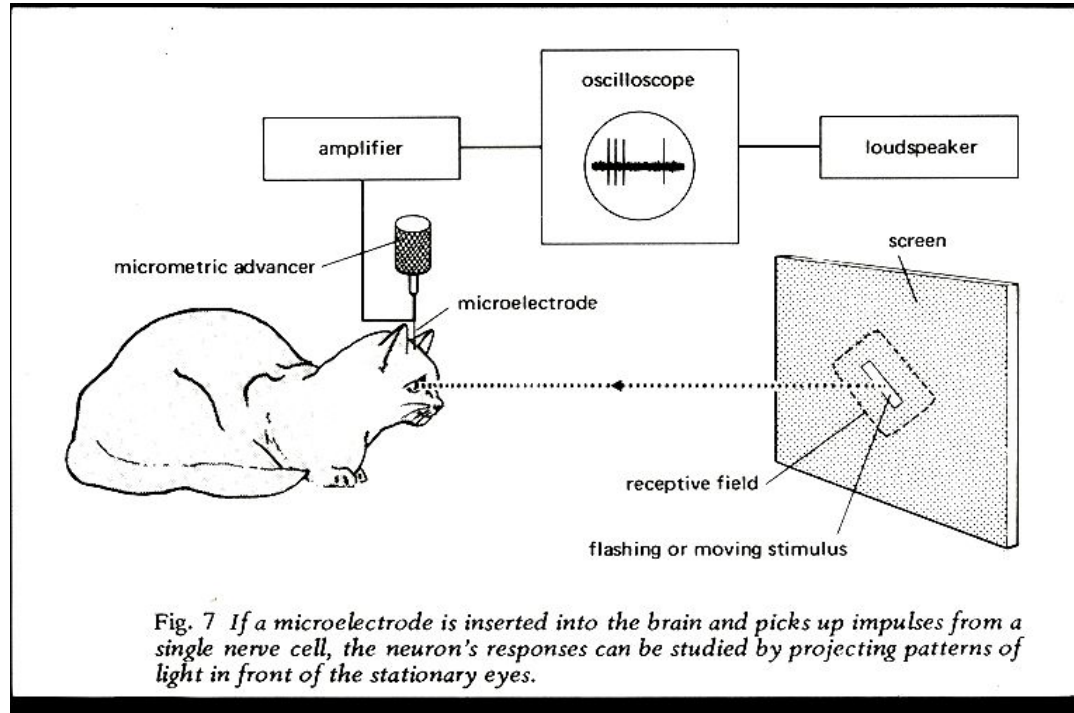


(e) Our result

Зачем нам нужны новые слои?

- Изображения представляют собой объекты с большим количеством признаков
- Изображение в формате RGB размера 640x480 будет иметь ~1млн признаков
- Число параметров полносвязной сети с внутренним слоем из 10 нейронов равно ~10млн
- **Большое число параметров** модели существенно затрудняет процесс обучения
- Полносвязный слой не даст нам **translation invariance**

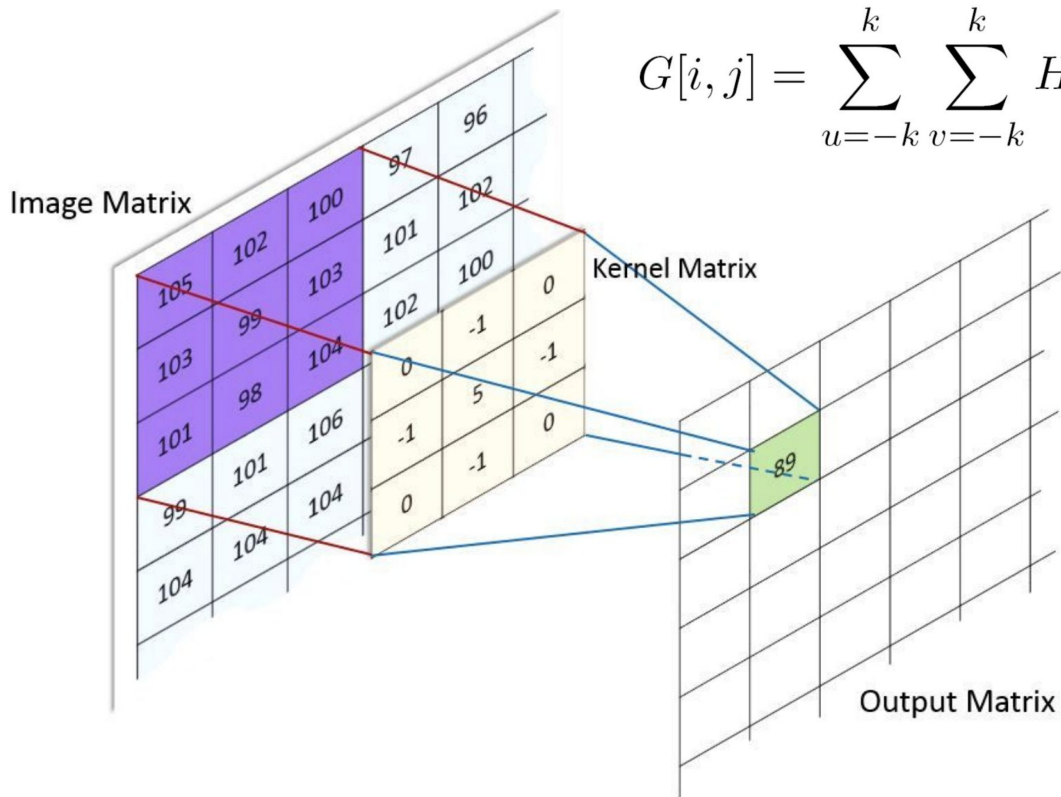
A cat perceptron



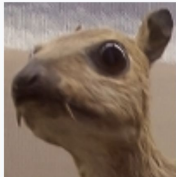
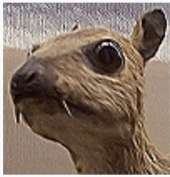
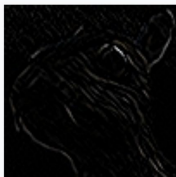
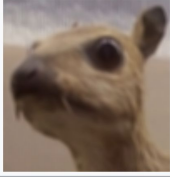
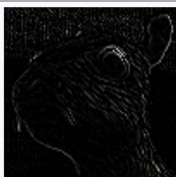
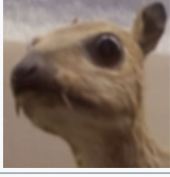
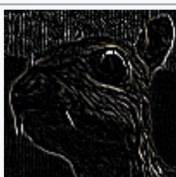
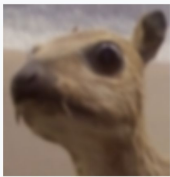
Нейрон реагирует на определенный паттерн в определенной области

Операция свертки [convolution]

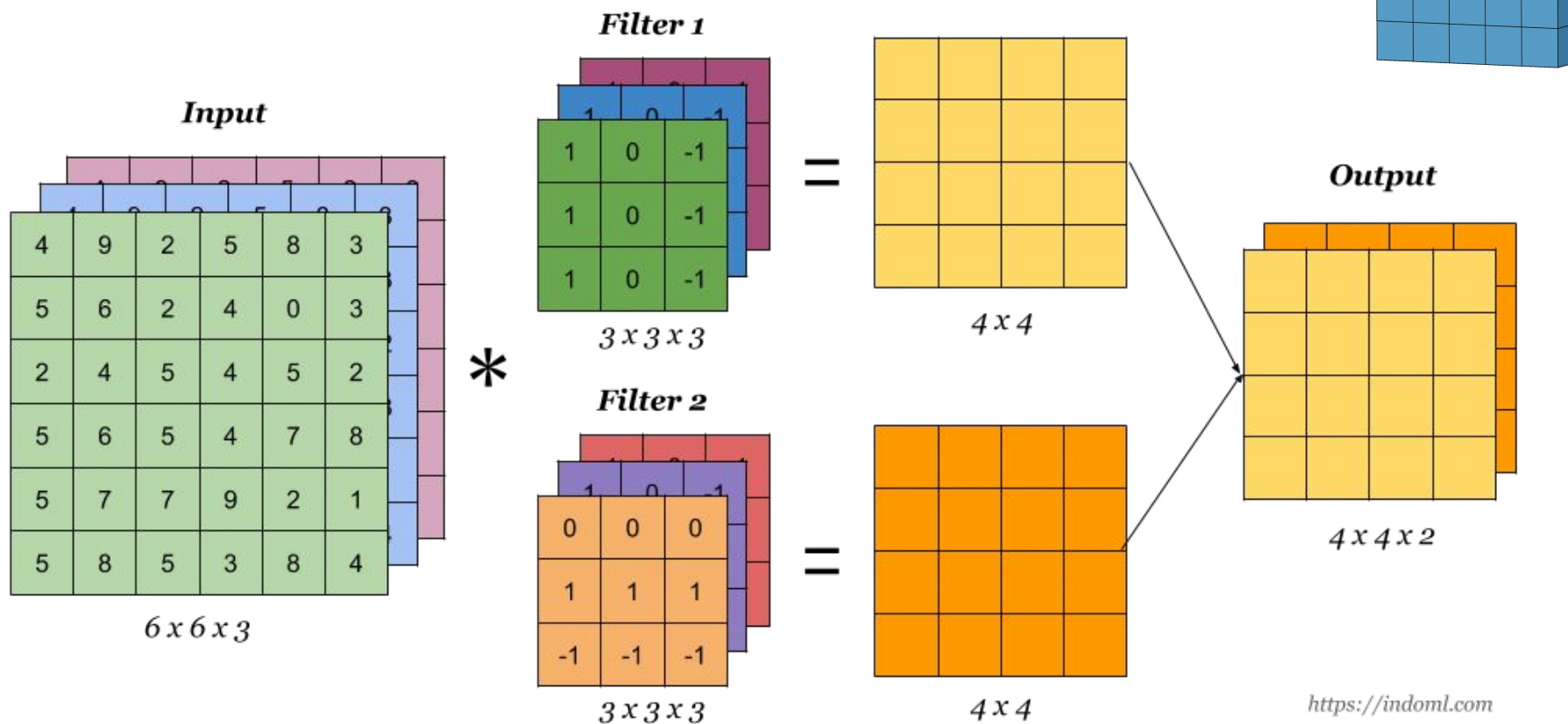
$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i - u, j - v]$$



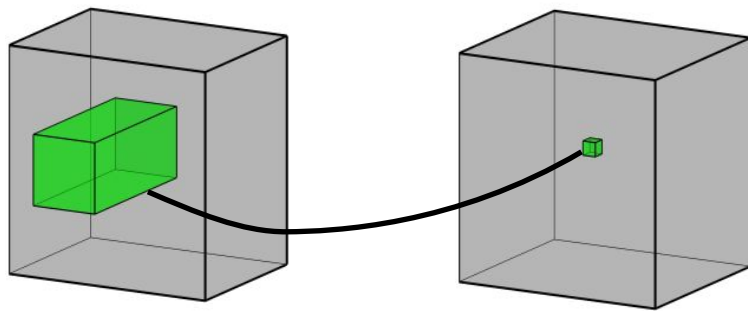
Операция свертки [convolution]

Operation	Kernel ω	Image result g(x,y)			
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$		Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$		Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$		Gaussian blur 3 × 3 (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$		Gaussian blur 5 × 5 (approximation)	$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

Операция свертки [convolution]



Операция свертки [*convolution*]

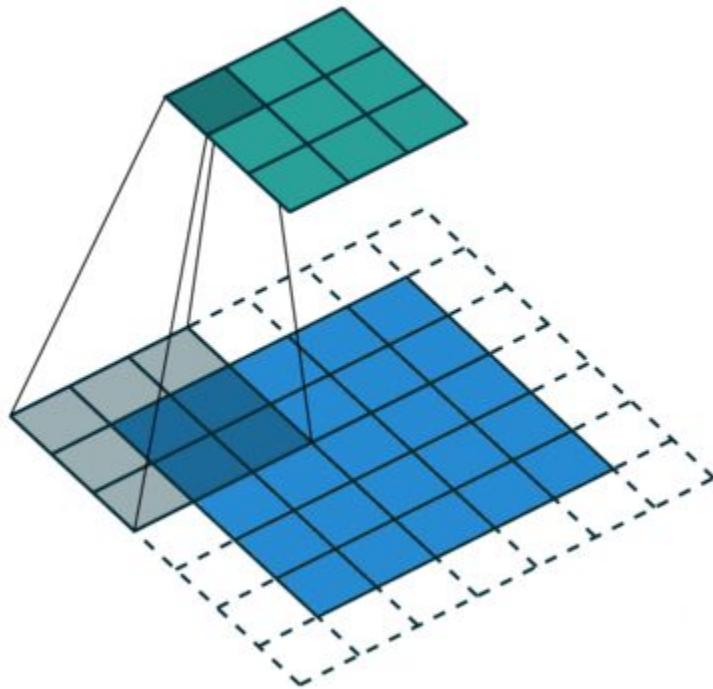


Сверточный слой

```
keras.layers.convolutional.Conv2D(filters, kernel_size,  
                                   strides=(1, 1), padding='valid',  
                                   dilation_rate=(1, 1), activation=None)
```

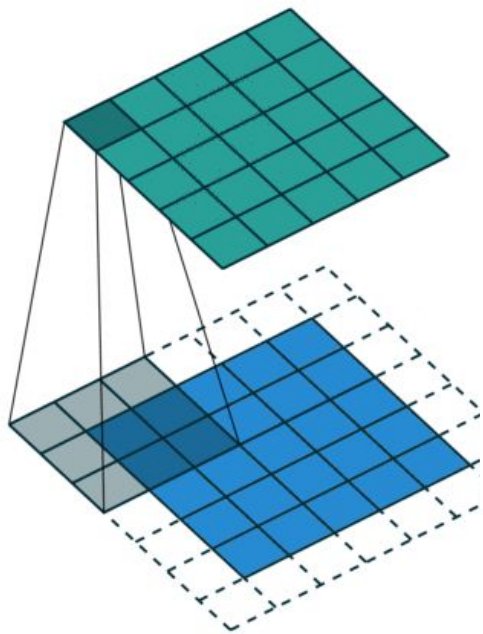
- **filters**: число фильтров в сверточном слое
- **kernel_size**: размер фильтра
- **strides**: шаг фильтра
- **padding**: отступы "valid" или "same"
- **dilation_rate**: расширение
- **activation**: функция активации, по умолчанию линейная

Stride

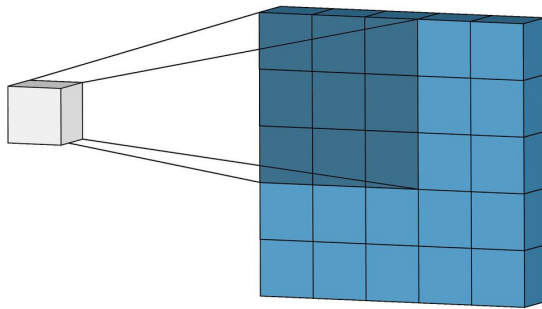


Padding

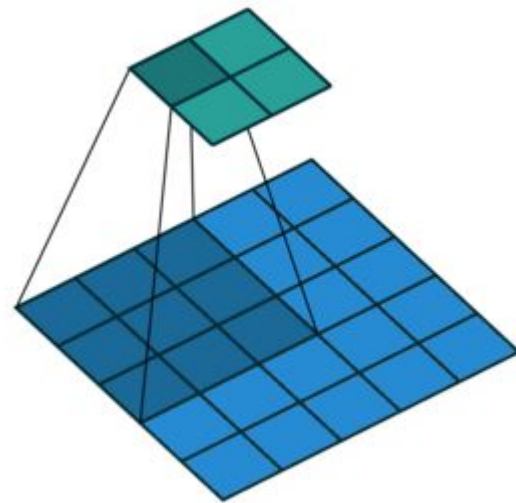
Padding



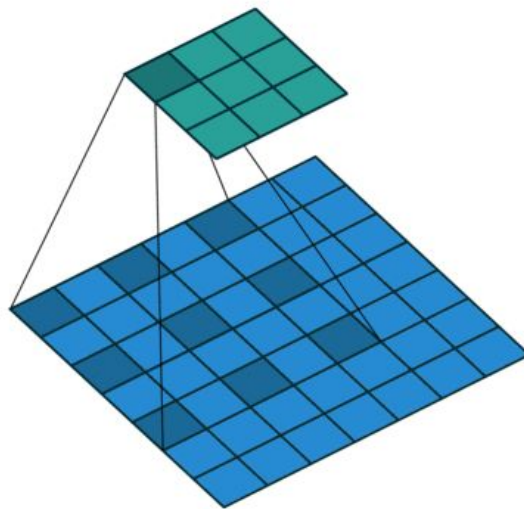
No padding



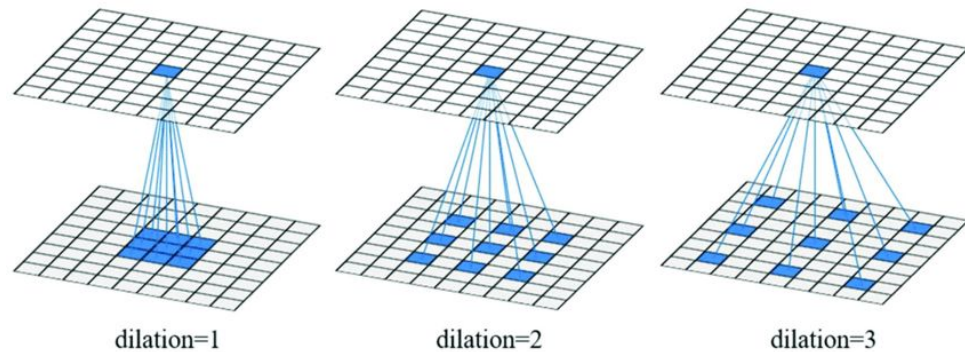
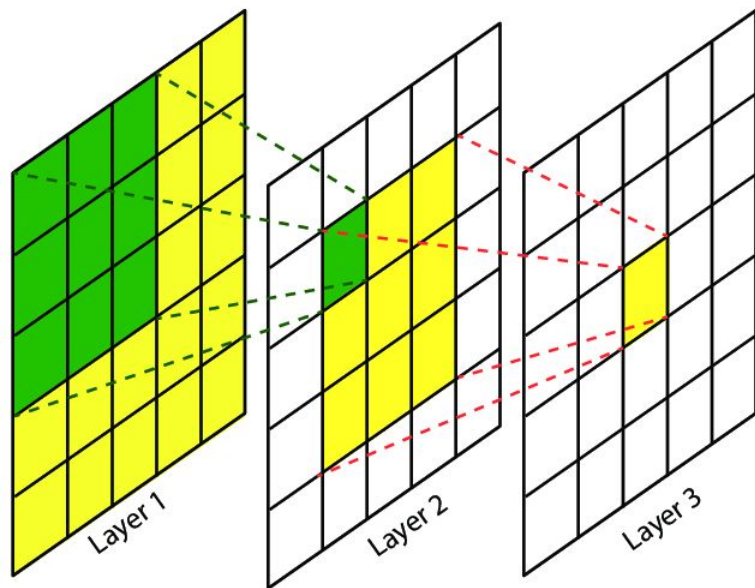
No padding + stride



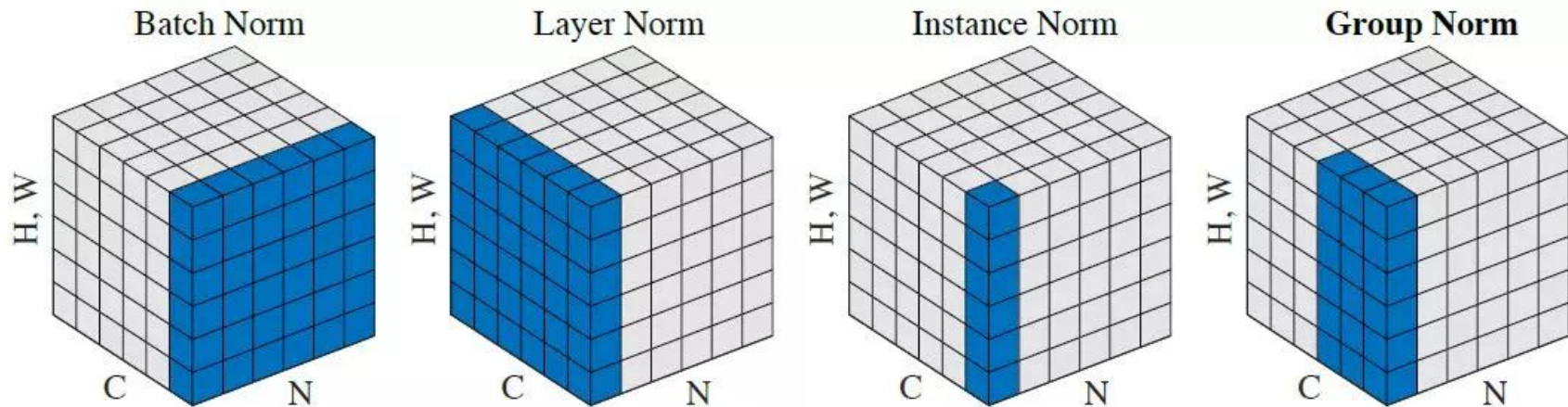
Dilation



Область восприятия [*receptive field*]



Batchnorm



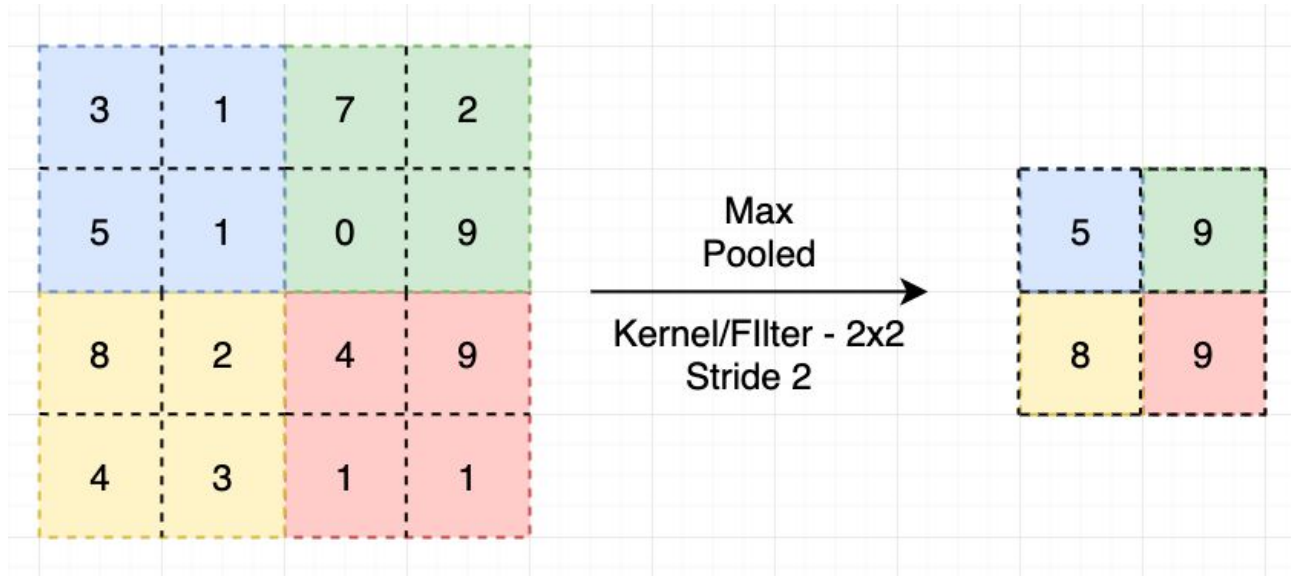
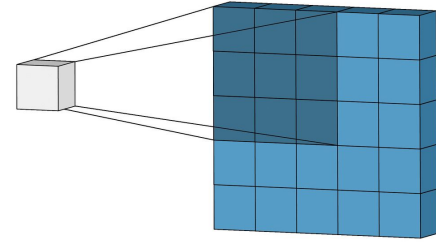
$$x = (x - \text{np.mean}(x, \text{axis}=\dots)) / \text{np.std}(x, \text{axis}=\dots)$$

Глобально хочется двух вещей:

чтобы считалось побыстрее

и receptive field был побольше

Pooling (max/average)



Pooling (max/average)

```
keras.layers.pooling.MaxPooling2D(pool_size=(2, 2), strides=None,  
padding='valid', data_format=None)
```

- **pool_size**: размер окна
- **strides**: шаг окна, по умолчанию совпадает с pool_size
- **padding**: отступ "valid" или "same"
- **data_format**: задает измерение в котором хранятся номера каналов

im2col

8	9	4	4
---	---	---	---

Flat Kernel

Original Kernel:

8	9
4	4

Original Input Matrix:

3	9	0
2	8	1
1	4	8

Im2Col

3	9	2	8
9	0	8	1
2	8	1	4
8	1	4	8

Window 1
Window 2
Window 3
Window 4

Result after Matrix Multiplication + bias:

Bias 0.06

145.06	108.06	108.06	121.06
--------	--------	--------	--------

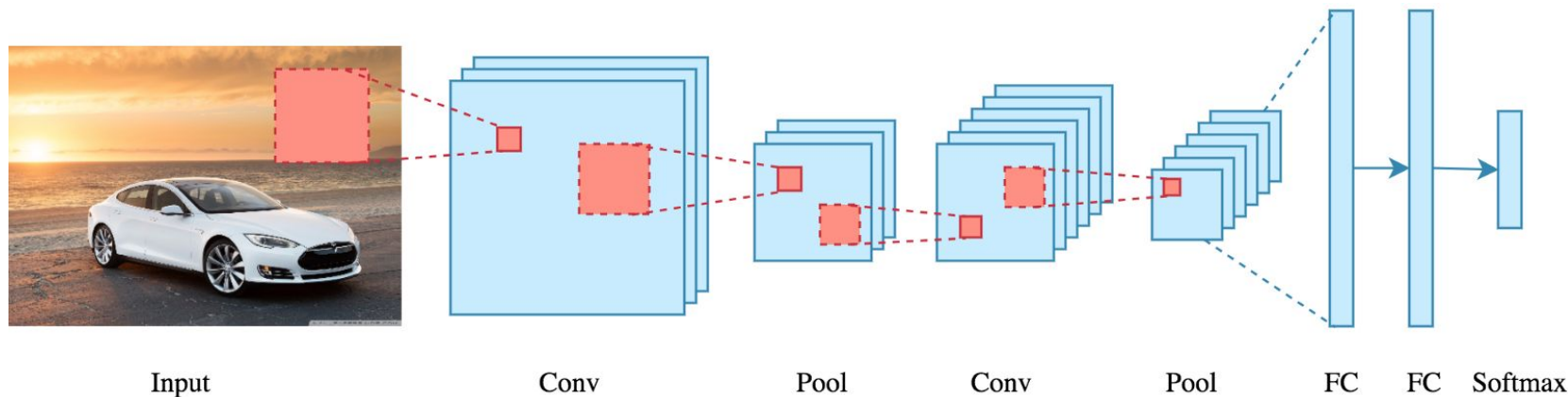
Reshaping :

145.06	108.06
108.06	121.06

Im2Col-Reshaping

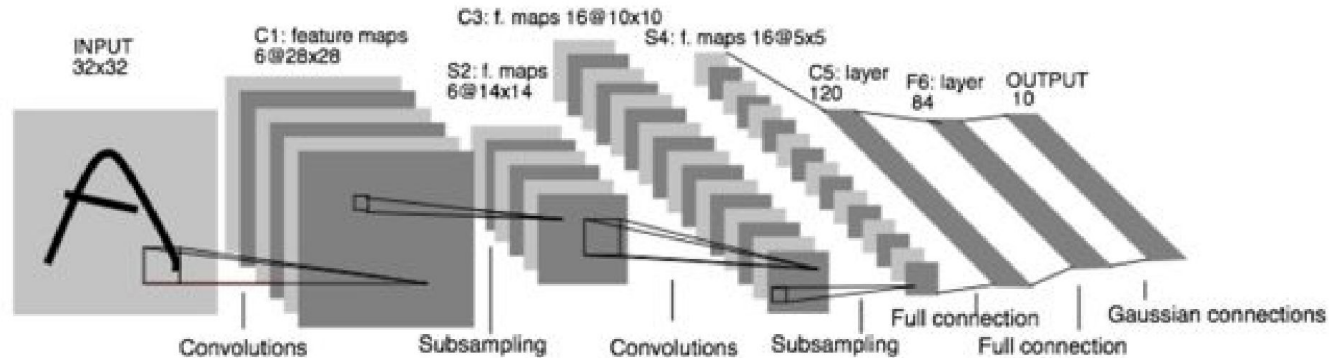
Сверточные нейронные сети (CNNs)

- В нейронных сетях для обработки изображений полносвязные слои заменяют на сверточные
- Полносвязные слои могут использоваться в выходном слое
- В сверточных сетях также используются слои понижения размера (Pooling), слои нормализации (BatchNorm) и регуляризации (Dropout)



LeNet

Convolutional Networks: 1989

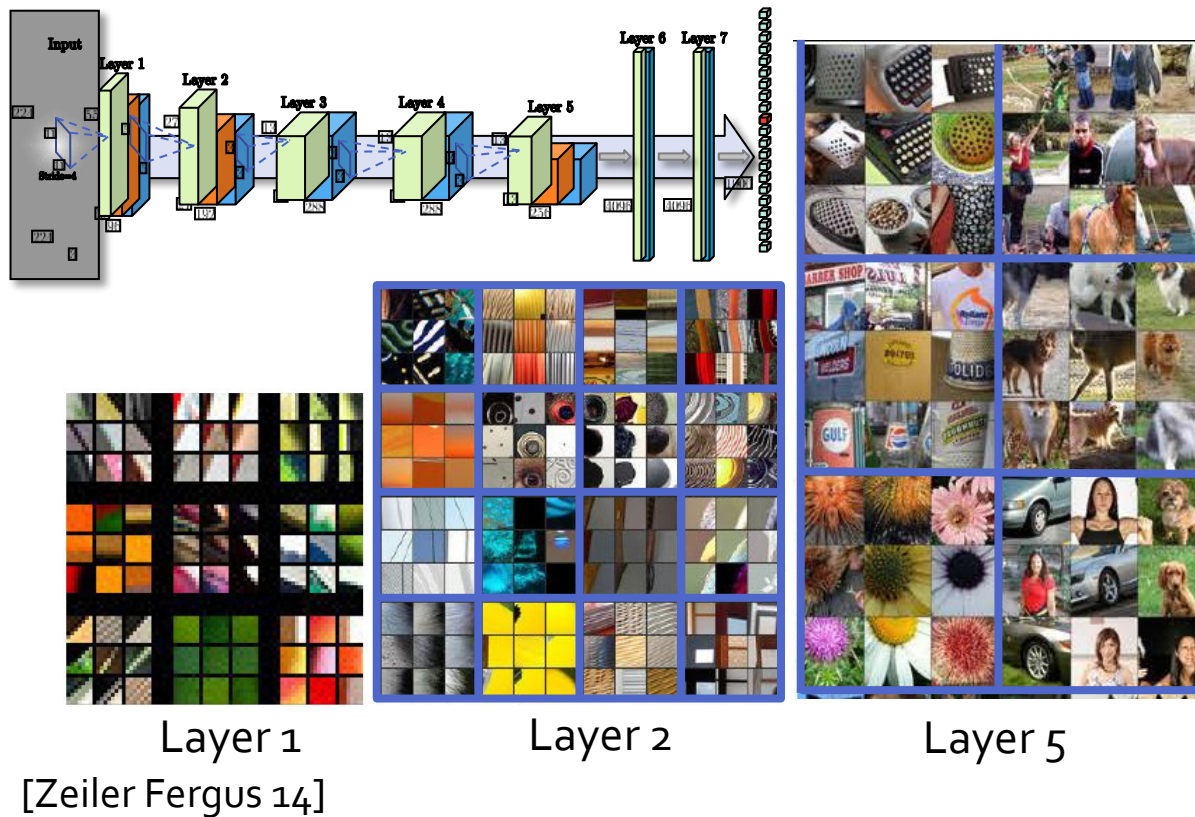


LeNet: a layered model composed of convolution and subsampling operations followed by a holistic representation and ultimately a classifier for handwritten digits. [LeNet]

LeNet

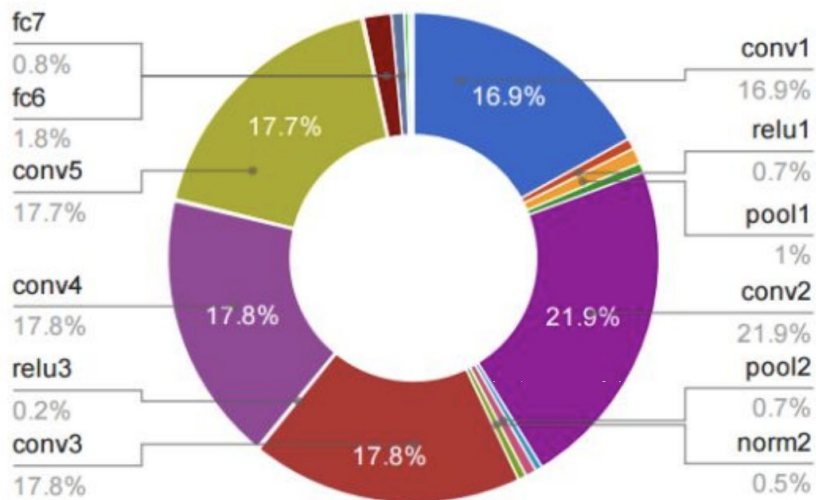
- Every convolutional layer includes three parts: convolution, pooling, and nonlinear activation functions
- Using convolution to extract spatial features (Convolution was called receptive fields originally)
- ~~• Subsampling average pooling layer~~
- ~~• tanh activation function~~
- ~~• Using MLP as the last classifier~~
- ~~• Sparse connection between layers to reduce the complexity of computation~~

Репрезентации внутри CNN

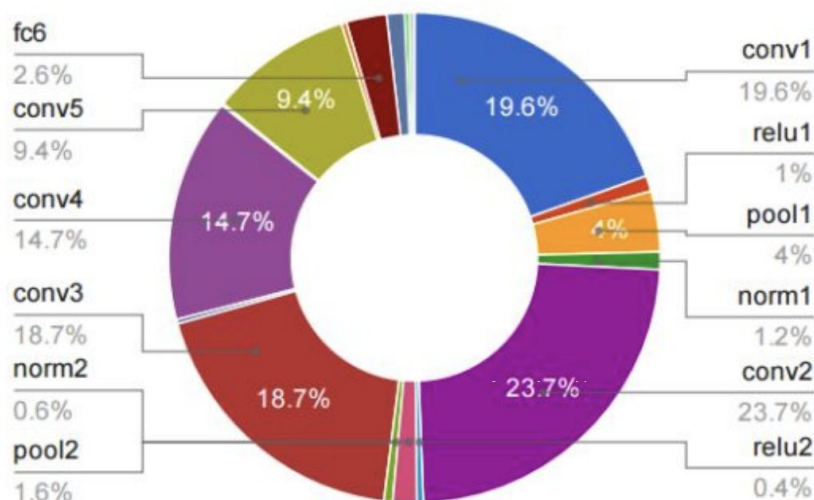


Производительность различных слоев

GPU Forward Time Distribution

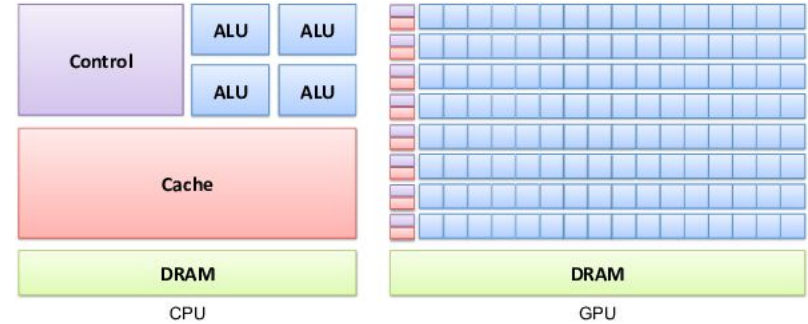
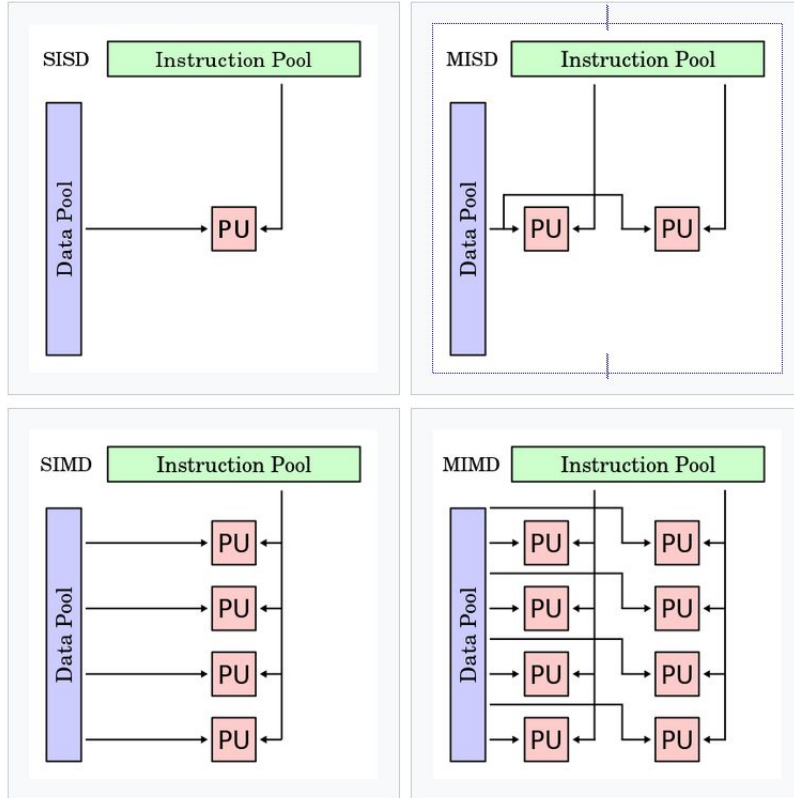


CPU Forward Time Distribution

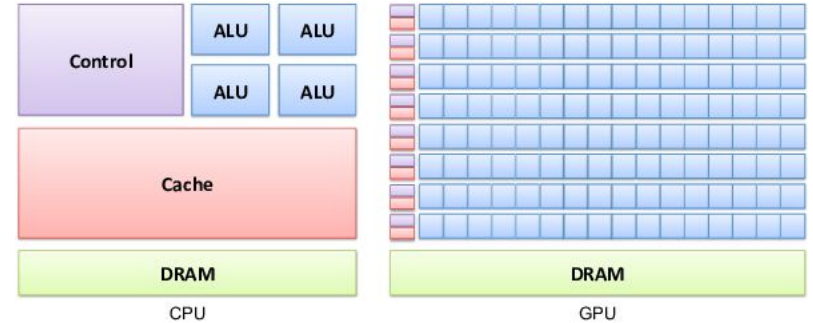
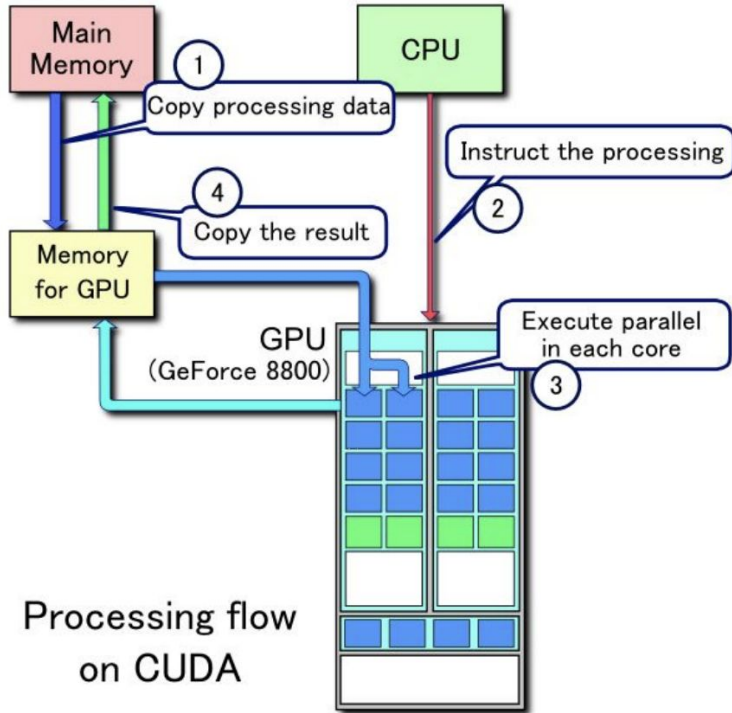


Try that yourself with kernprof or nvprof!

CPU vs. GPU

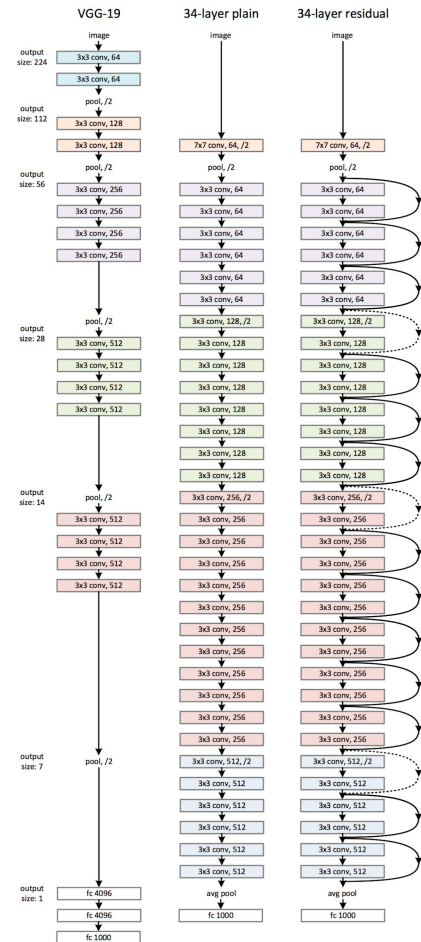
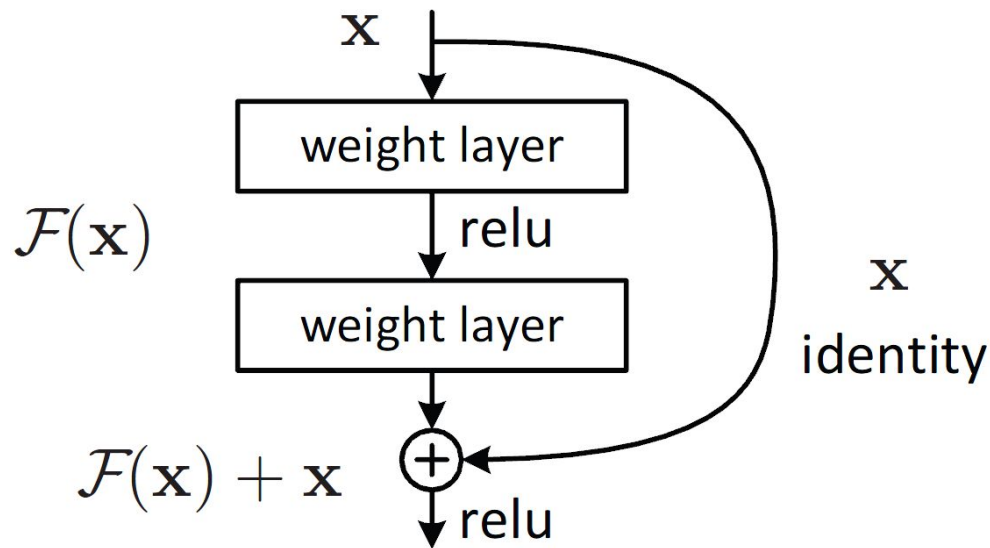


CPU vs. GPU

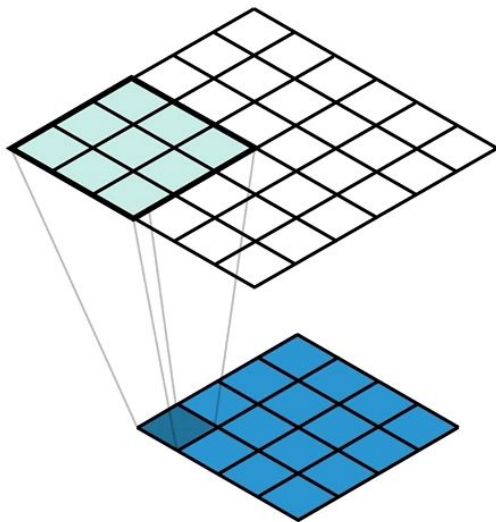


Advanced section

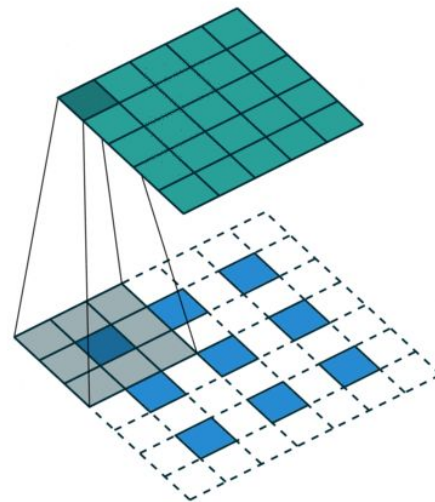
ResNet



How to upsample?



Transposed conv



Bed of nails

Доп материалы

<https://neurohive.io/ru/vidy-nejrosetej/resnet-34-50-101/>

TBD