

Глубокое обучение

Свёрточные сети

План лекции

- Свёрточный слой
 - Зачем
 - Идея
 - Параметры
 - im2col
 - Receptive field
- Pooling
- Датасеты картиночной классификации
- Архитектуры свёрточных нейросетей
 - LeNet
 - AlexNet
 - VGG
 - Inception
 - ResNet
 - MobileNet

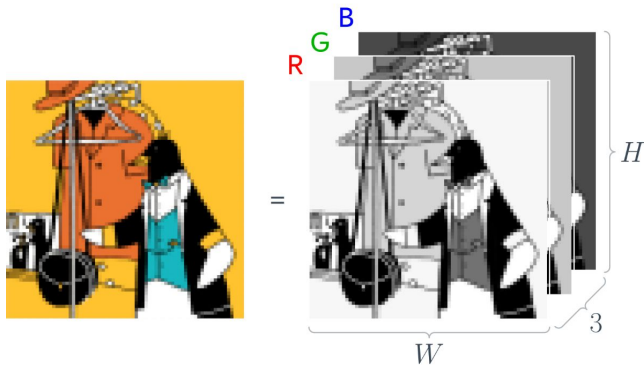
Зачем

- количество параметров

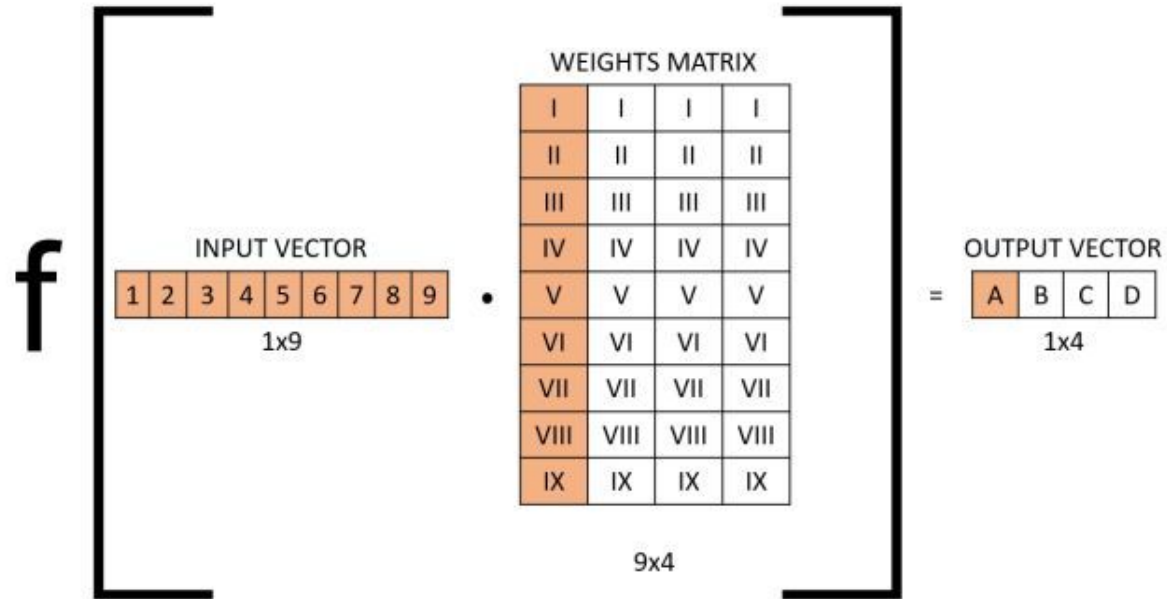
В первом линейном слое $H \times W \times C \times C_{out}$ параметров.

При этом если C_{out} сделать маленьким, то качество будет не очень, особенно на больших картинках типа 1980x1080.

Если большим, то перепараметризуем сетку, т.е. столкнёмся с переобучением, сложной оптимизацией и т.д.



Linear layer



Зачем

- количество параметров
- структура данных никак не учитывается (хотим устойчивости относительно простых изменений)



(собачка)

Зачем

- количество параметров
- структура данных никак не учитывается (хотим устойчивости относительно простых изменений)



(собачка)



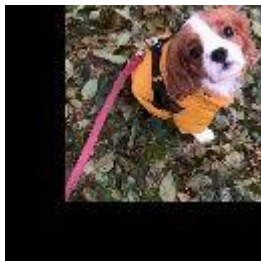
(тоже собачка)

Зачем

- количество параметров
- структура данных никак не учитывается (хотим устойчивости относительно простых изменений)



(собачка)



(тоже собачка)



(и тут)

Зачем

- количество параметров
- структура данных никак не учитывается (хотим устойчивости относительно простых изменений)



(собачка)



(тоже собачка)



(и тут)



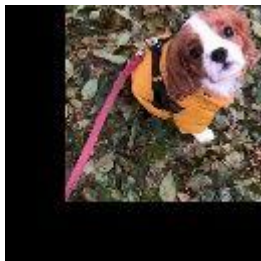
(снова она)

Зачем

- количество параметров
- структура данных никак не учитывается (хотим устойчивости относительно простых изменений)



(собачка)



(тоже собачка)



(и тут)



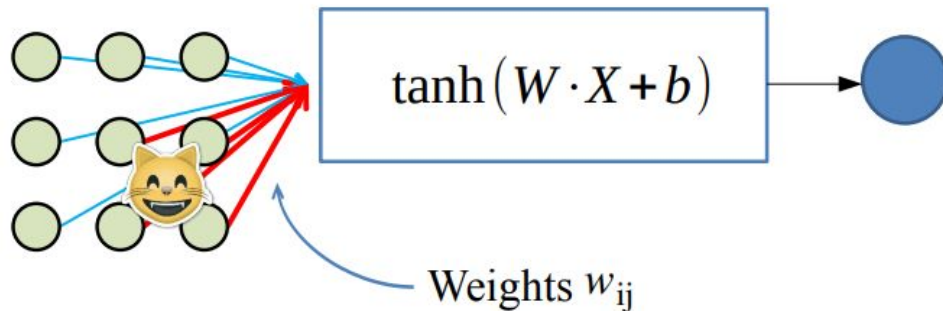
(снова она)



(и даже тут)

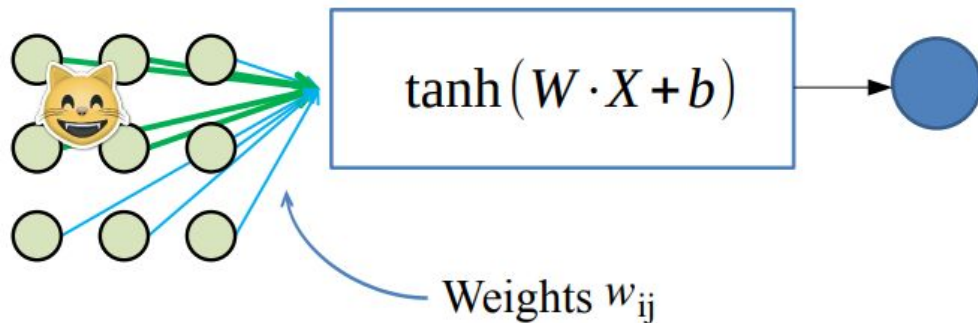
Начнём с последней проблемы

Мотивация



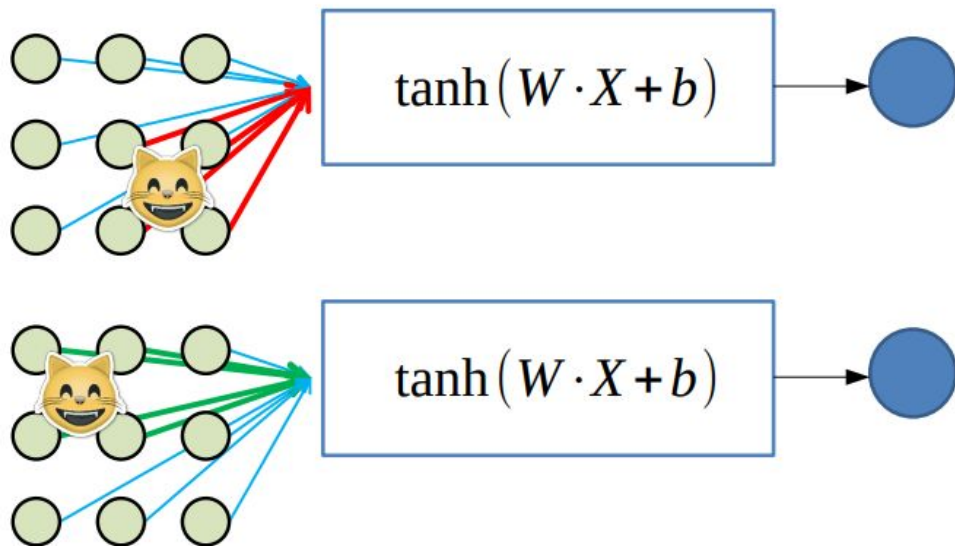
On this object, you will train **red** weights to react on cat face

Мотивация



On this object, you will train **green** weights to react on cat face

Мотивация

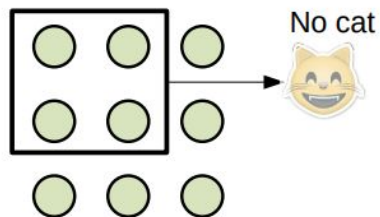
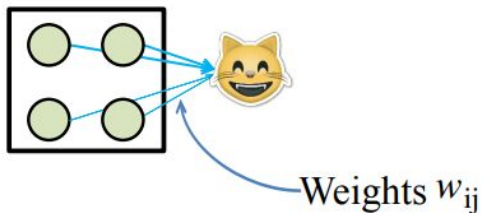


You network will have to learn those two cases separately!
Worst case: one neuron per position.

Мотивация

Same features for each spot

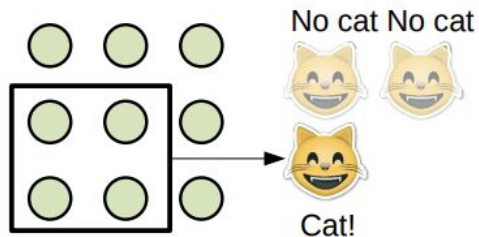
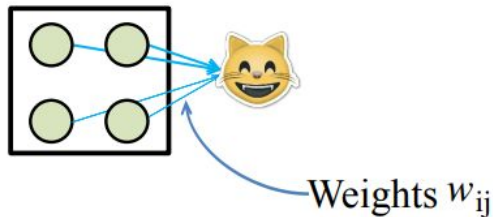
Portable cat detector pro!



Мотивация

Same features for each spot

Portable cat detector pro!



Convolution

Input image



Convolution
Kernel

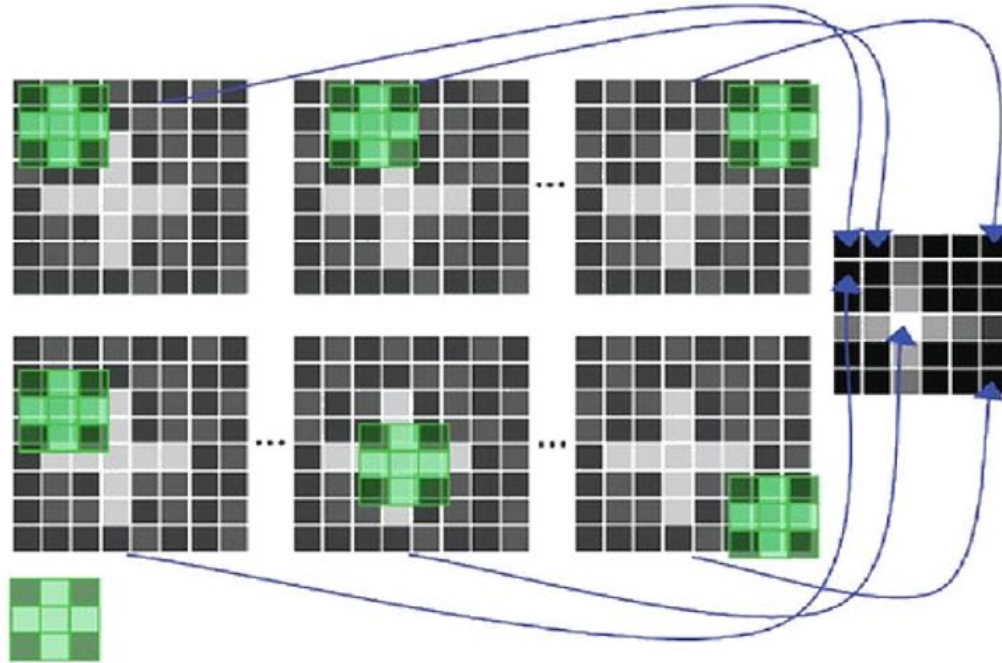
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map



Intuition: how **edge-like** is this square?

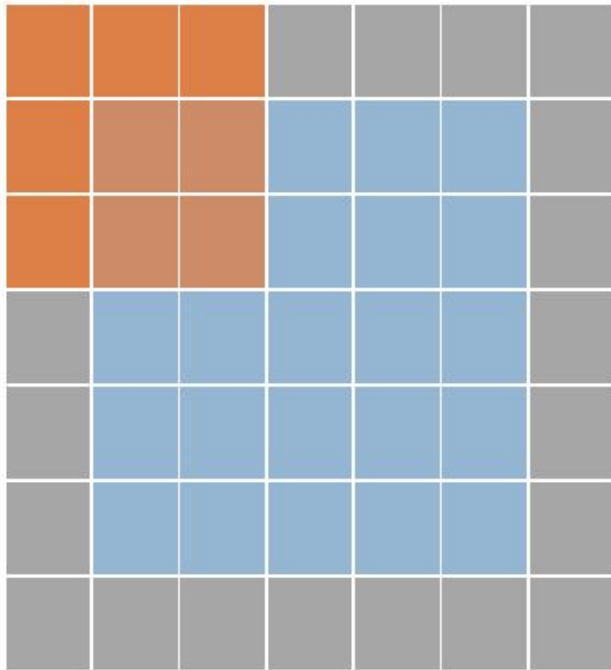
Convolution



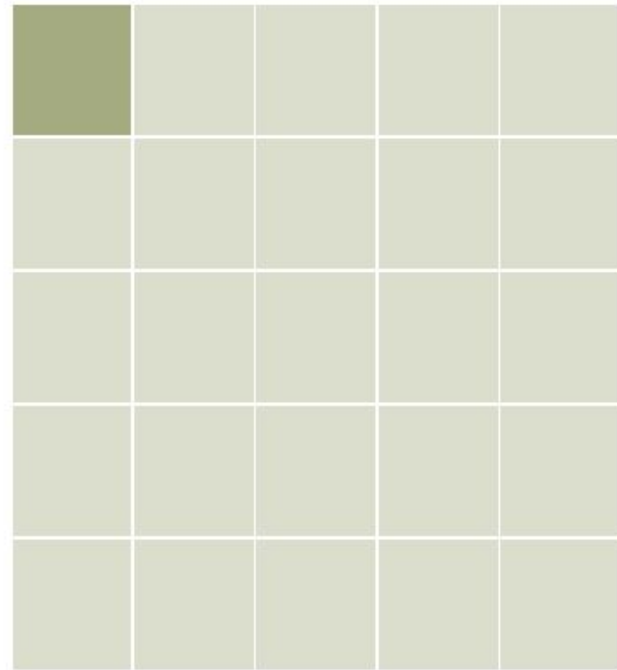
apply same filter to all patches

Convolution

Type: conv - Stride: 1 Padding: 1

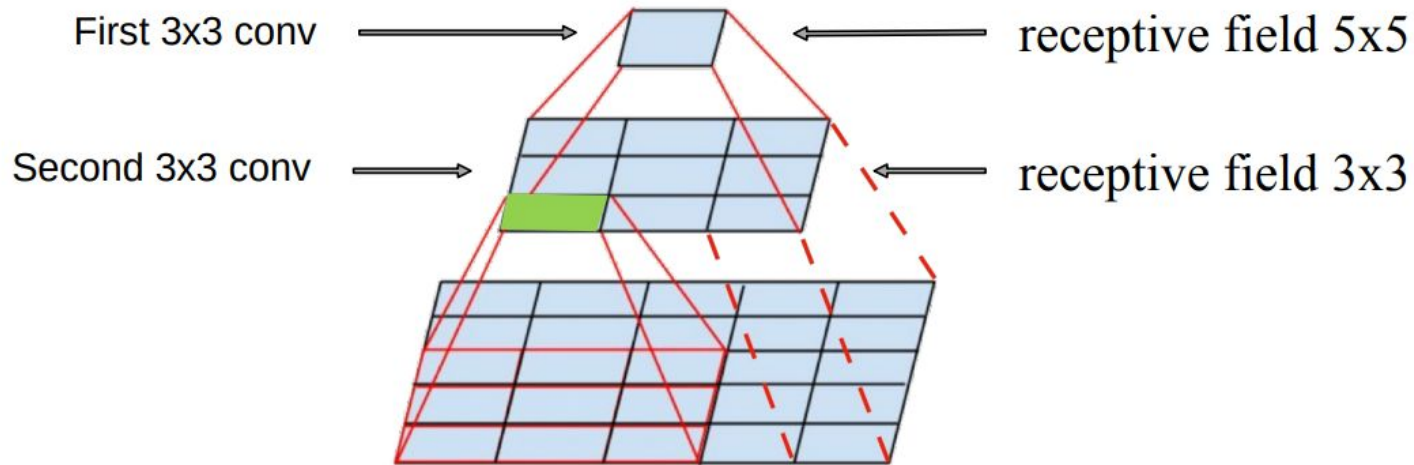


Input



Output

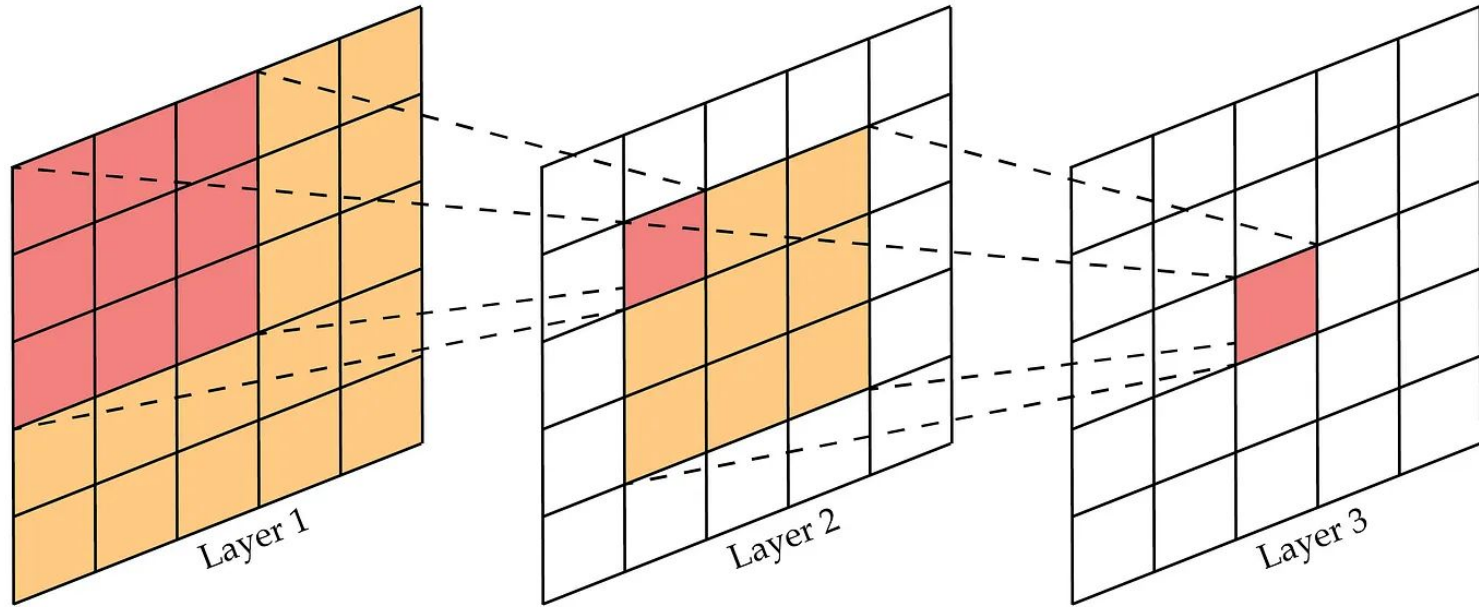
Receptive field



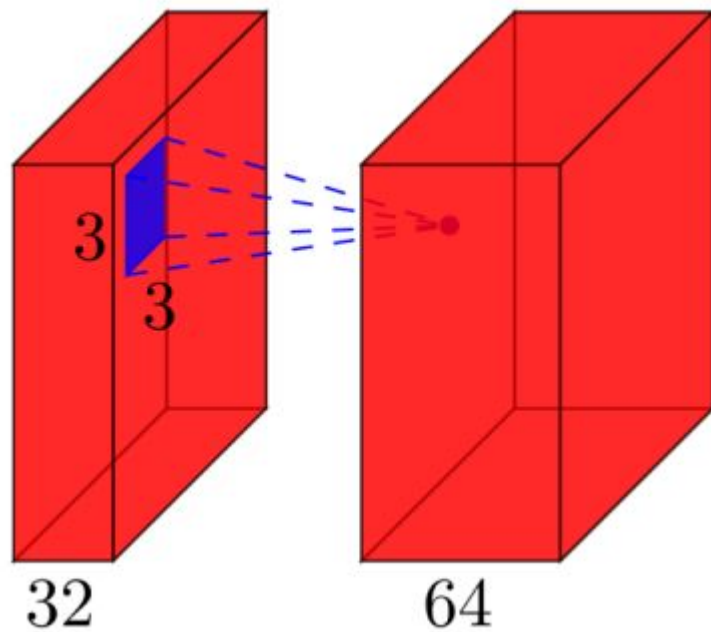
We can recognize larger objects by stacking several small convolutions!

Receptive field

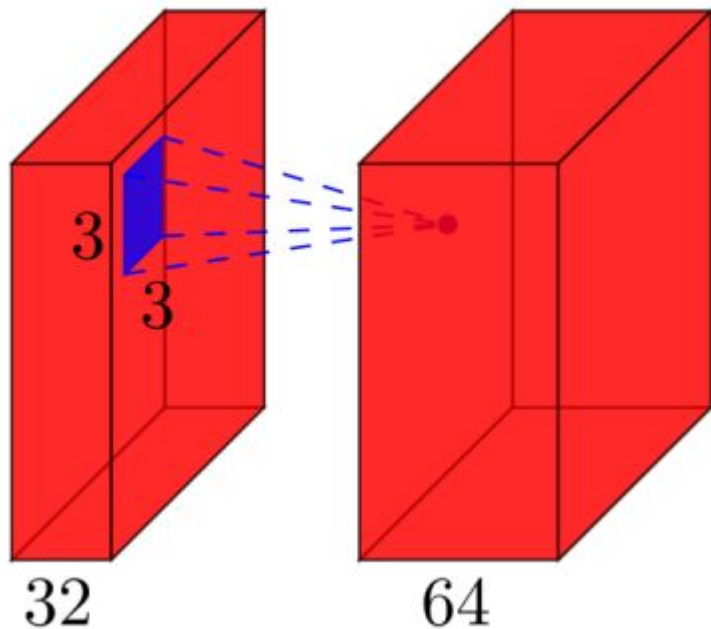
Receptive Field in Convolutional Networks



А какое количество параметров?



А какое количество параметров?



$$(3 * 3 * 32 + 1) * 64$$

Where

$3 * 3$ - kernel_size

32 - in_channels

1 - bias

64 - out_channels

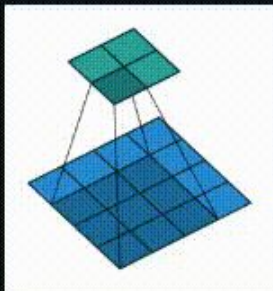
Параметры свёрточного слоя

- in_channels
- out_channels
- kernel_size

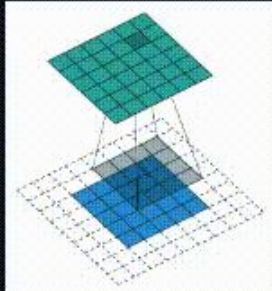
- stride
- padding
- dilation
- bias

Padding и Stride в свёрточном слое

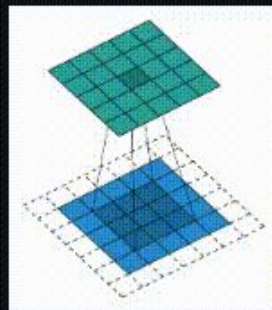
N.B.: Blue maps are inputs, and cyan maps are outputs.



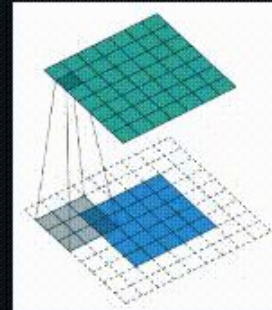
No padding, no strides



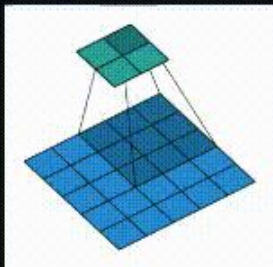
Arbitrary padding, no strides



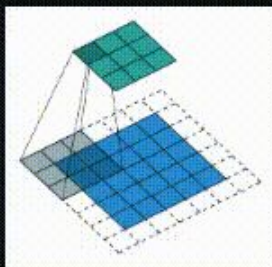
Half padding, no strides



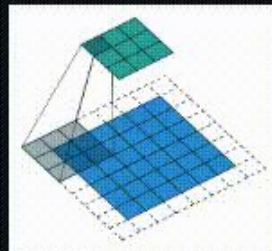
Full padding, no strides



No padding, strides



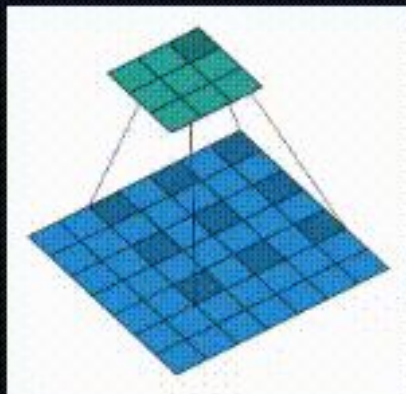
Padding, strides



Padding, strides (odd)

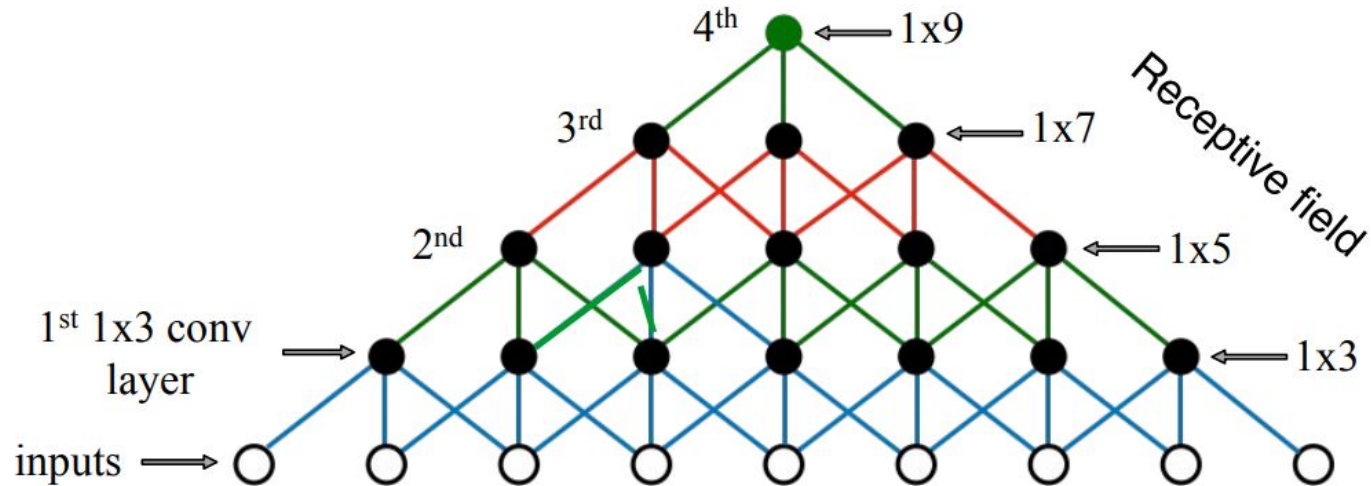
Dilation в свёрточном слое

N.B.: Blue maps are inputs, and cyan maps are outputs.



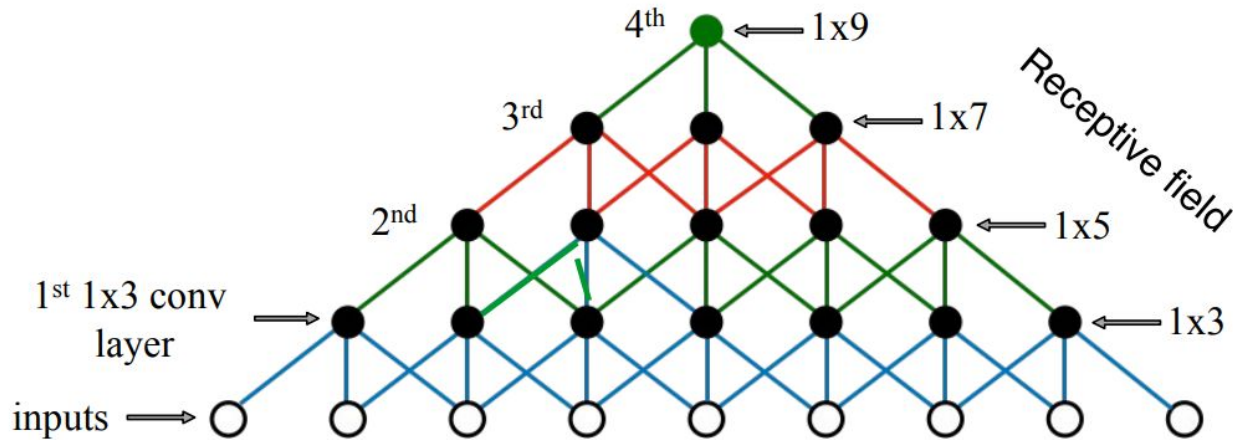
No padding, no stride, dilation

Receptive field



Q: how many 3x3 convolutions we should use to recognize a 100x100px cat

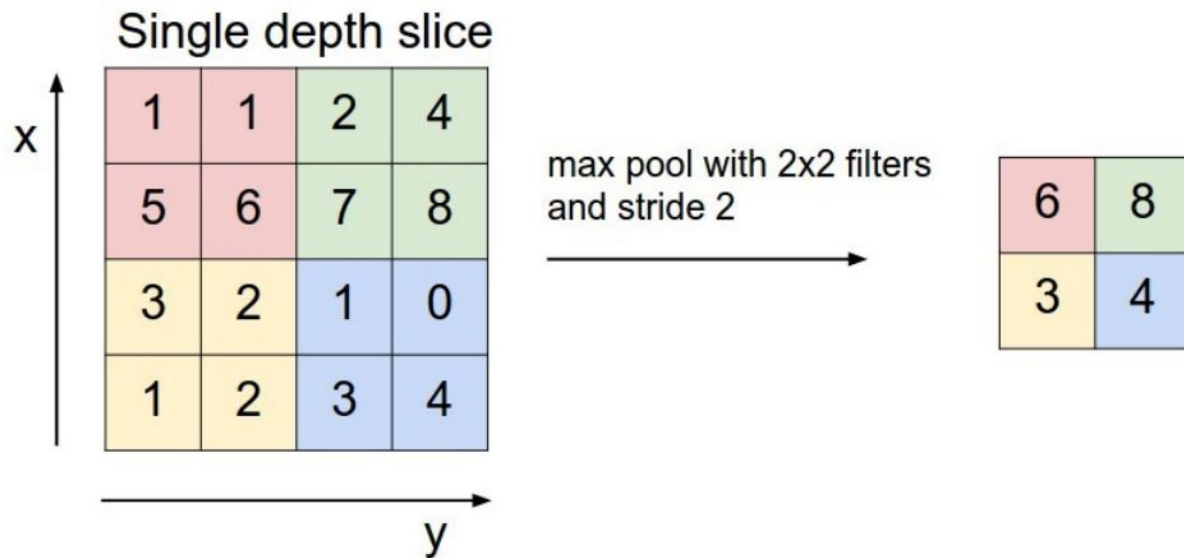
Receptive field



Q: how many 3x3 convolutions we should use
to recognize a 100x100px cat

A: around 50... we need to increase receptive field faster!

Pooling



Intuition: What is the max cat-likelihood over this area?

Pooling

Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2
pool size

100	184
12	45

Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

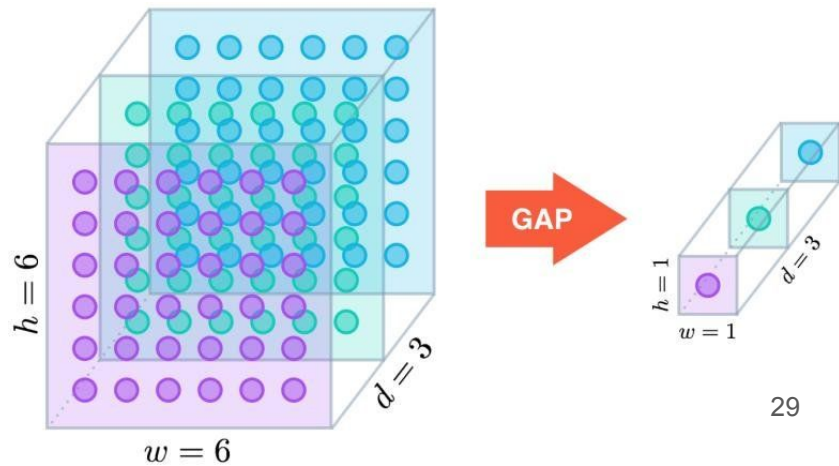
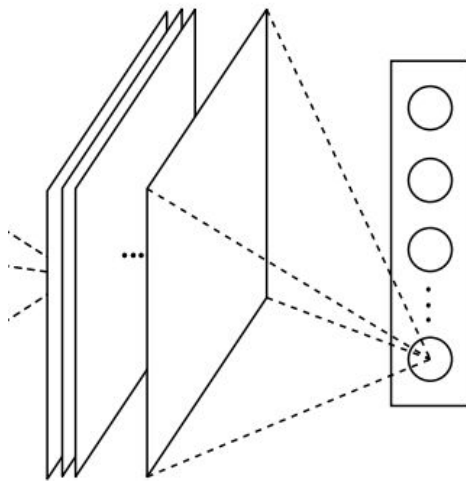
2 x 2
pool size

36	80
12	15

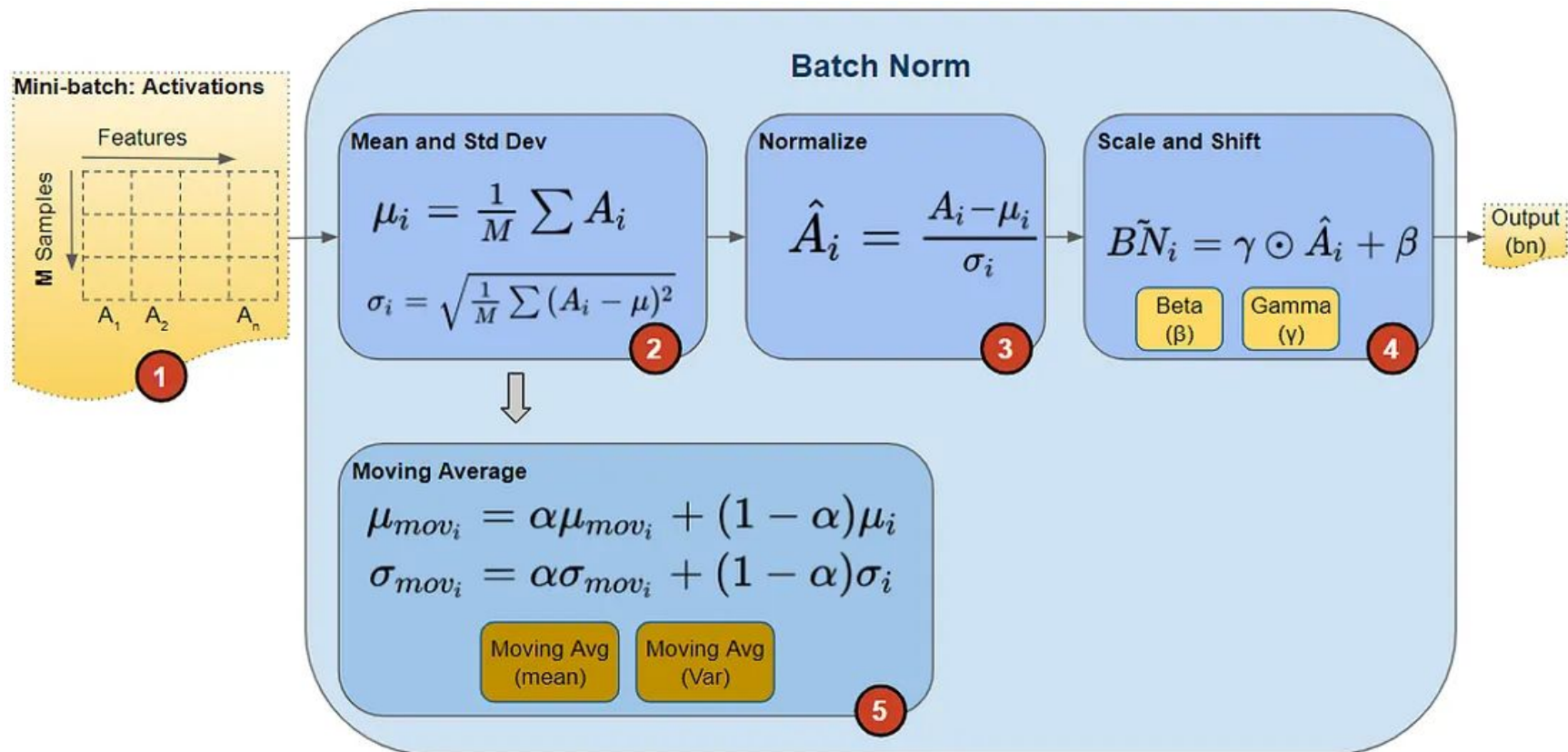
Global pooling

The idea is to generate one feature map for each corresponding category of the classification task in the last conv layer. Instead of adding fully connected layers on top of the feature maps, we take the average of each feature map, and the resulting vector is fed directly into the softmax layer.

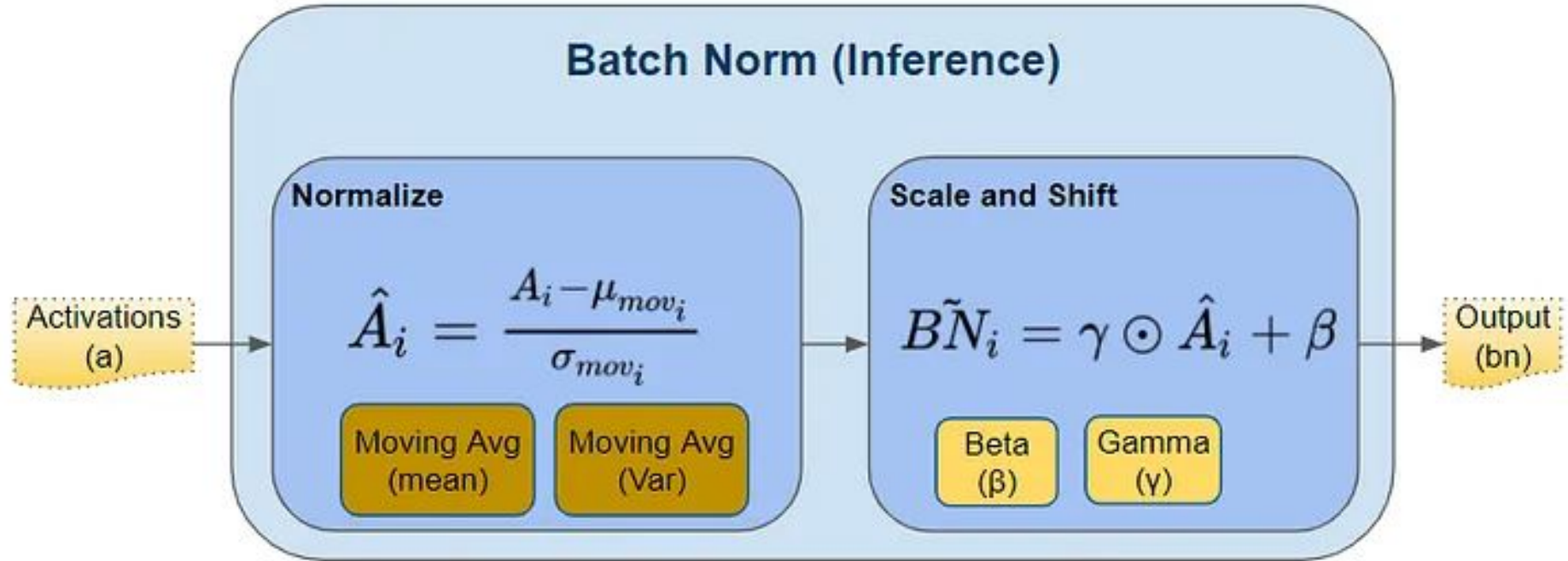
Also can be global max pooling or global average pooling.



BatchNorm

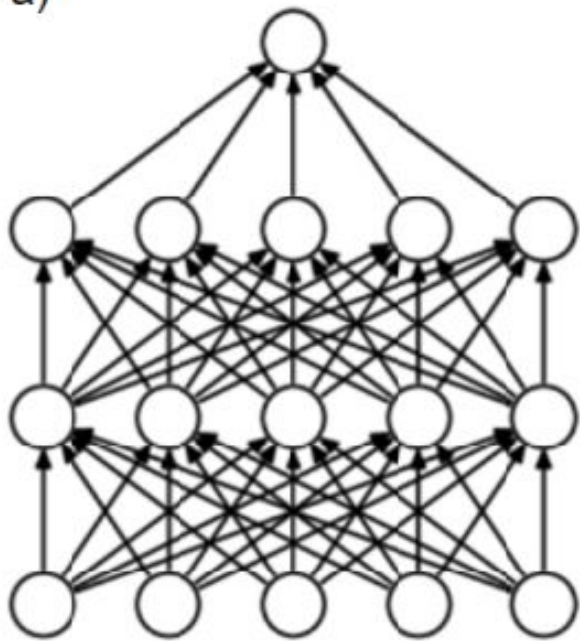


BatchNorm

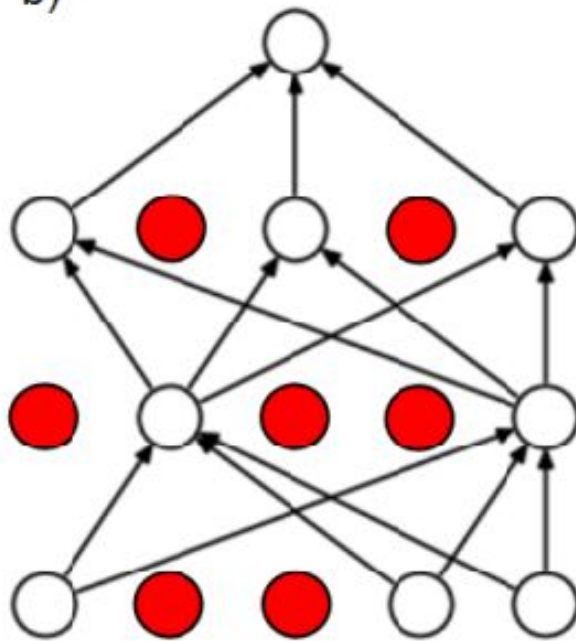


Dropout

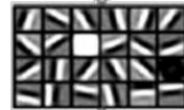
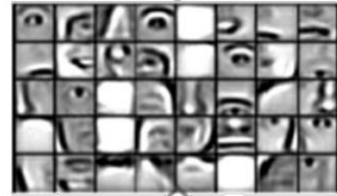
a)



b)



Stack more layers



Discrete Choices

⋮

Layer 2 Features

Layer 1 Features

Original Data

Проблемы больших нейросетей

What you sign for if you stack 1000 layers:

- `MemoryError(0x...)`
- Gradients can vanish
- Gradients can explode
- Activations can vanish
- Activations can explode
- Overfits like crazy

Проблемы больших нейросетей

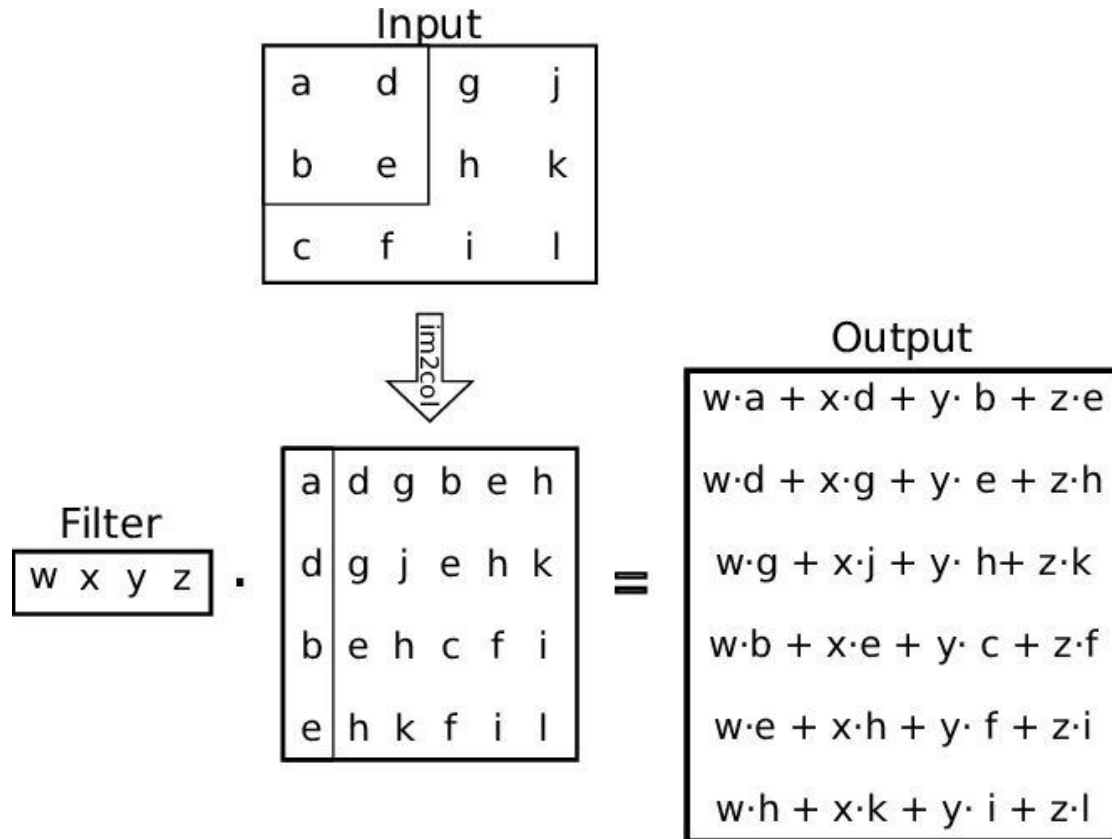
What you sign for if you stack 1000 layers:

- MemoryError(0x...)
 - **Gradients can vanish**
 - **Gradients can explode**
 - **Activations can vanish**
 - **Activations can explode**
 - Overfits like crazy
- Residual connections
- Batch normalization or similar

Де-юре это просто линейный слой (с кучей нулей)



Де-факто линейный слой с операцией im2col



Аугментации



- Idea: we can get N times more data by tweaking images.
- If you rotate cat image by 15° , it's still a cat
- Rotate, crop, zoom, flip horizontally, add noise, etc.
- Sound data: add background noises

Datasets

ImageNet

Data set Details:

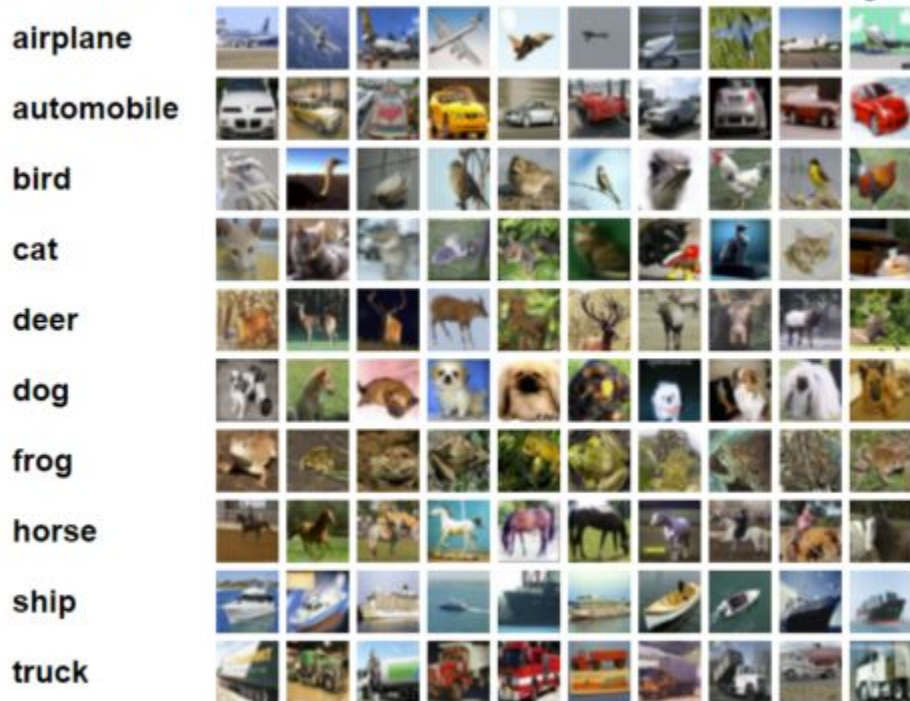
- 14,197,122 images
- 1000 classes
- 224*224



CIFAR-10

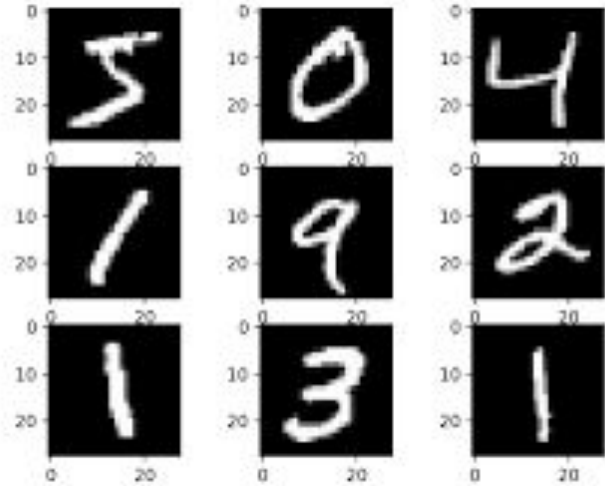
- 60,000 color images
- distributed across 10 classes
- 32*32

Here are the classes in the dataset, as well as 10 random images from each:



MNIST

- 60,000 training images and 10,000 test images of 0-9 digits
- 28*28 pixels



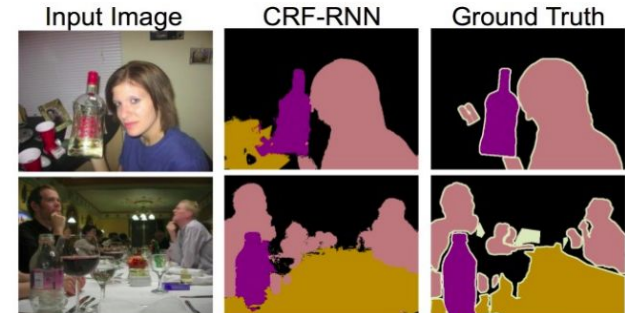
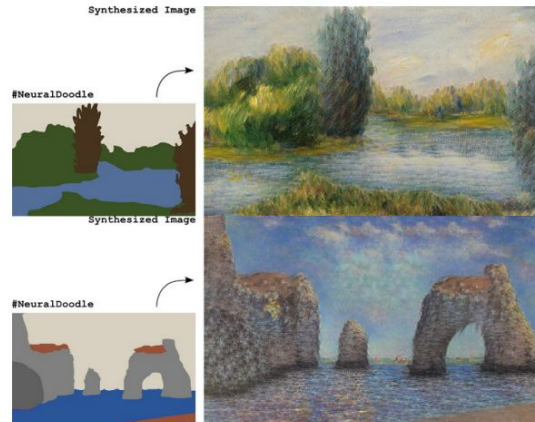
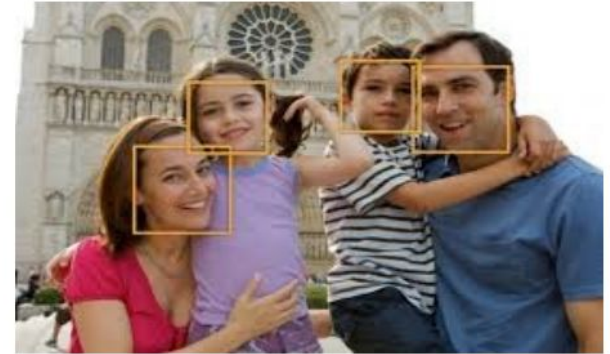
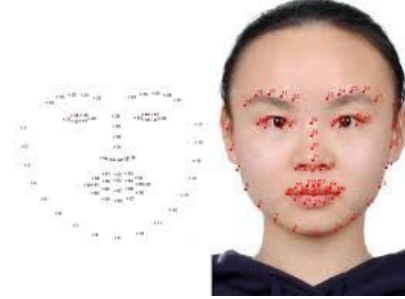
Fashion-MNIST

- 70,000 fashion images
- 28×28



Другие задачи в CV

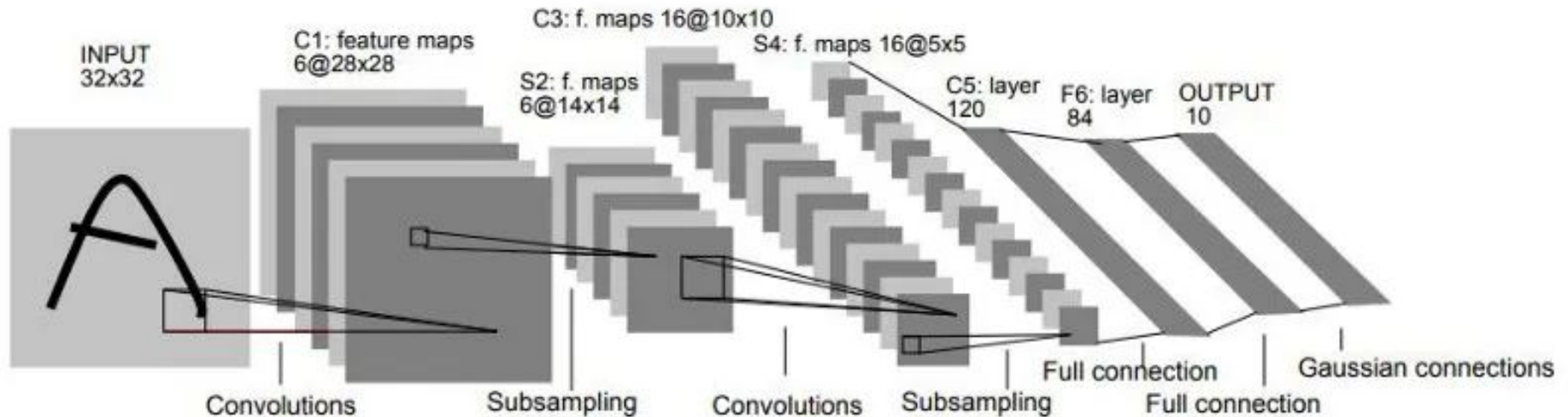
- point coordinates regression
- bounding box regression + classification
- segmentation
- style transferring
- image upscaling



Архитектуры

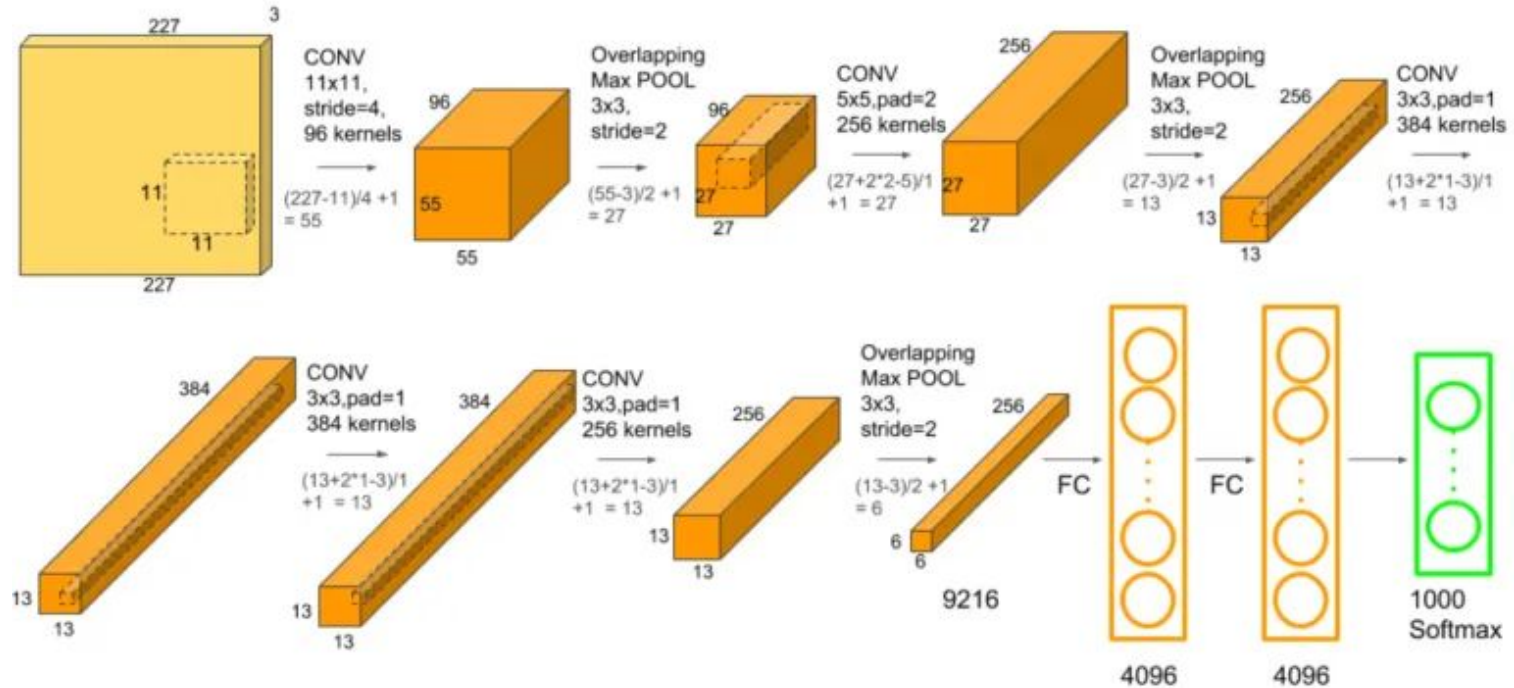
LeNet

LeNet-5, introduced by Yann LeCun and his team in the 1990s, was one of the first successful CNN architectures. Designed for handwritten digit recognition. About 60000 params

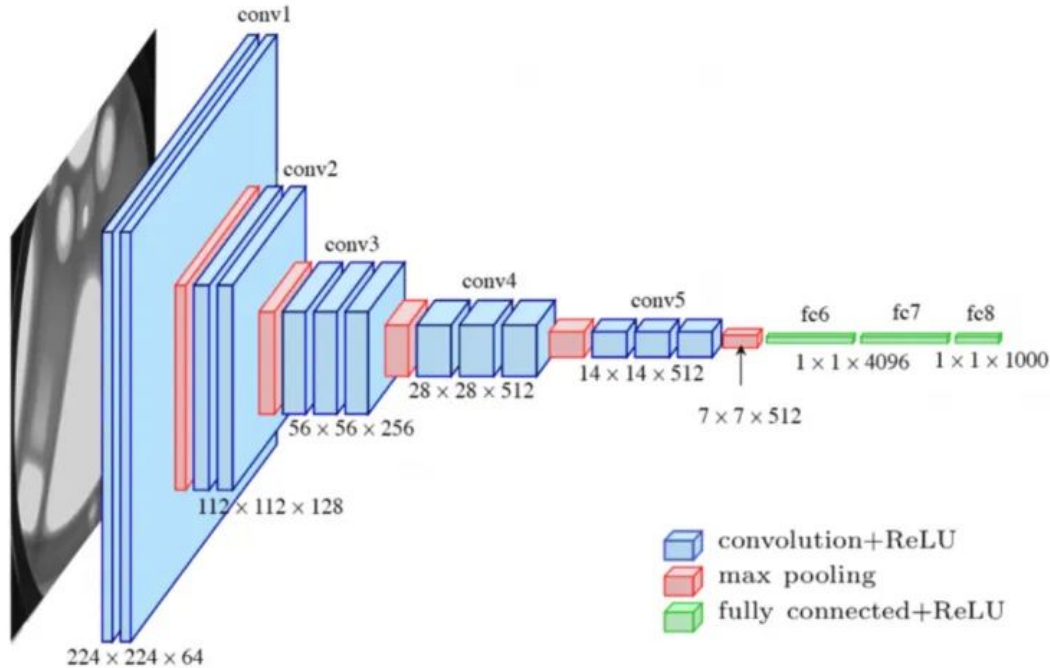


AlexNet

Alex Krizhevsky, Ilya Sutskever (ILSVRC) in 2012 was designed to be used with large-scale image datasets and it achieved state-of-the-art results at the time of its publication. 60 million params, activations - ReLU



VGGNet16



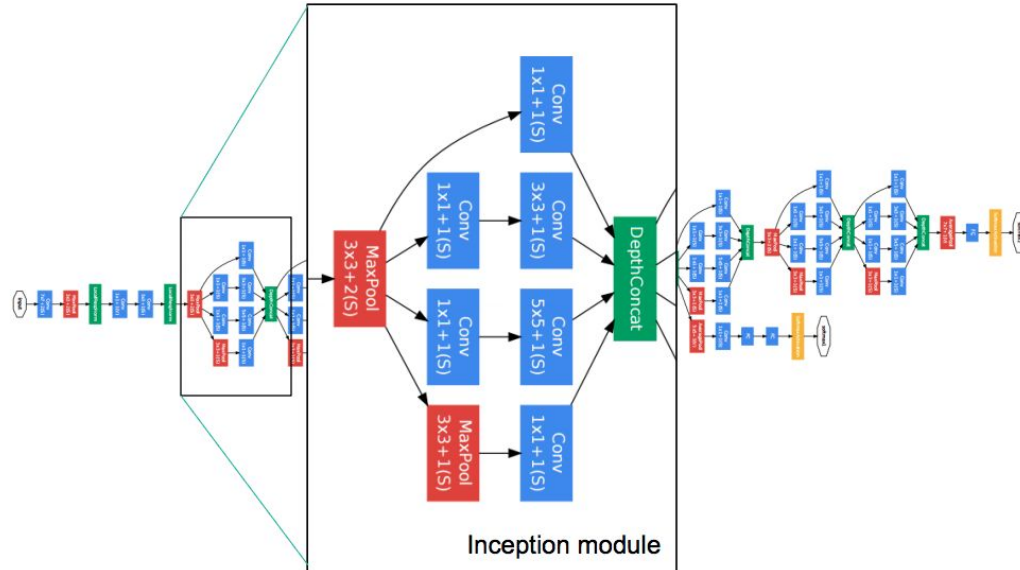
It significantly outperforms AlexNet by substituting several 3x3 kernel-sized filters for the huge kernel-sized filters.

Sota results on ImageNet 92.7 percent in 2014.

About 138 million params.

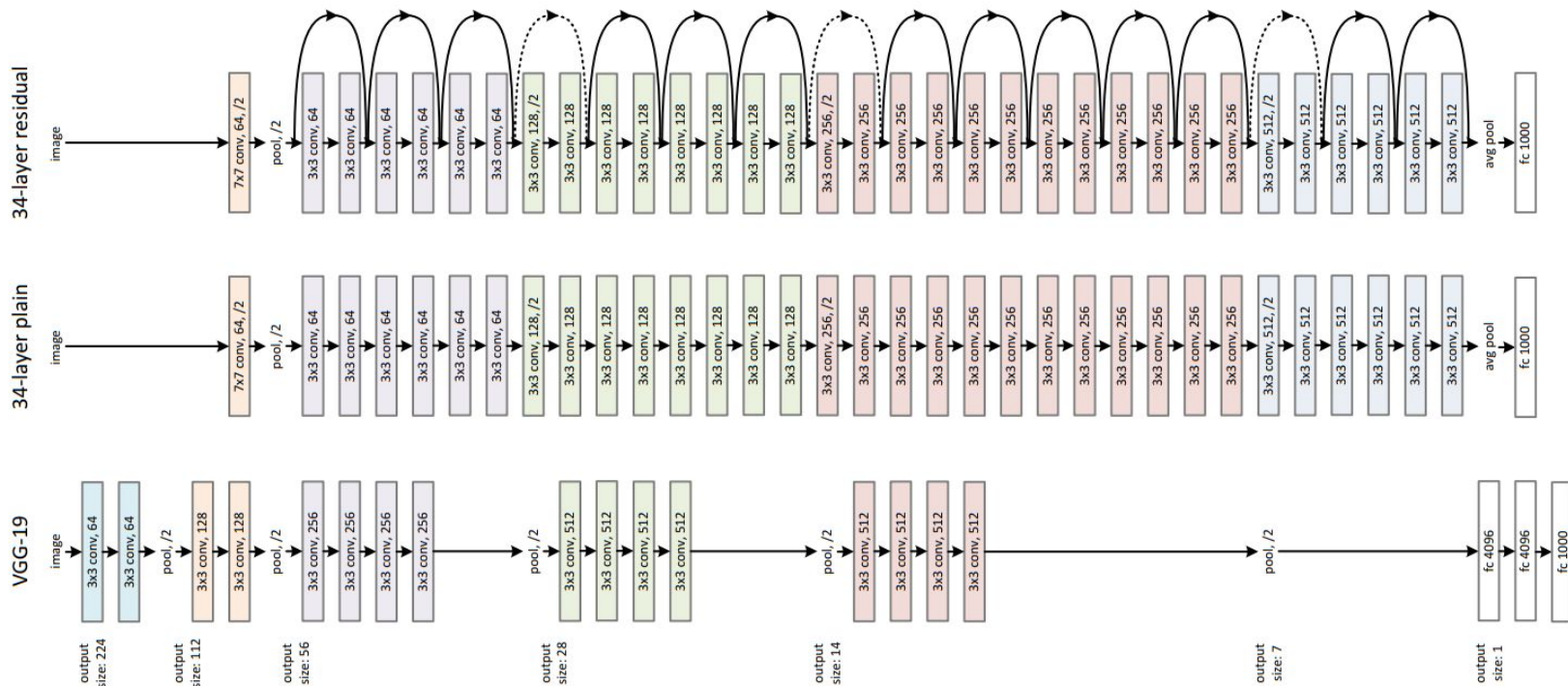
GoogLeNet (Inception)

Winner of ILSVRC 2014, introduced the Inception module, which employs parallel convolutional operations with different kernel sizes. This architecture efficiently captures features at multiple scales, promoting better generalization. Also proposed 1x1 convolution before compute-expensive operations. 7 million of params

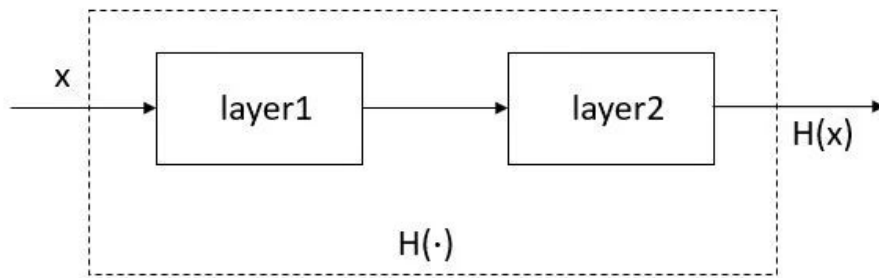


ResNet

2015-2016. Imagenet 75-79% accuracy (resnet34-resnet152). ~26 million params

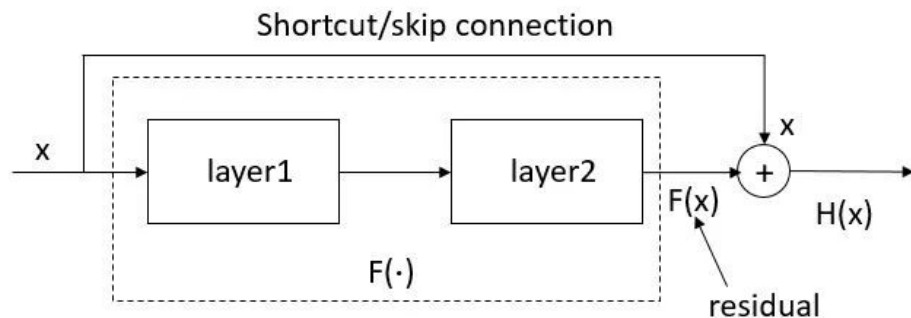


ResNet



$$\left| \frac{\partial H(x)}{\partial x} \right| \ll 1$$

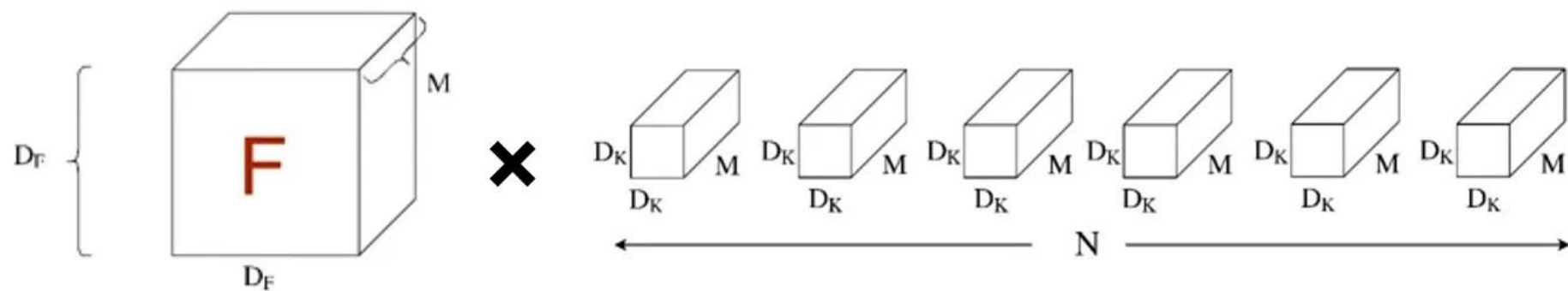
Vanishing gradient



$$\left| \frac{\partial H(x)}{\partial x} \right| = \left| \frac{\partial F(x) + \partial x}{\partial x} \right| = \left| \frac{\partial F(x)}{\partial x} \right| + 1$$

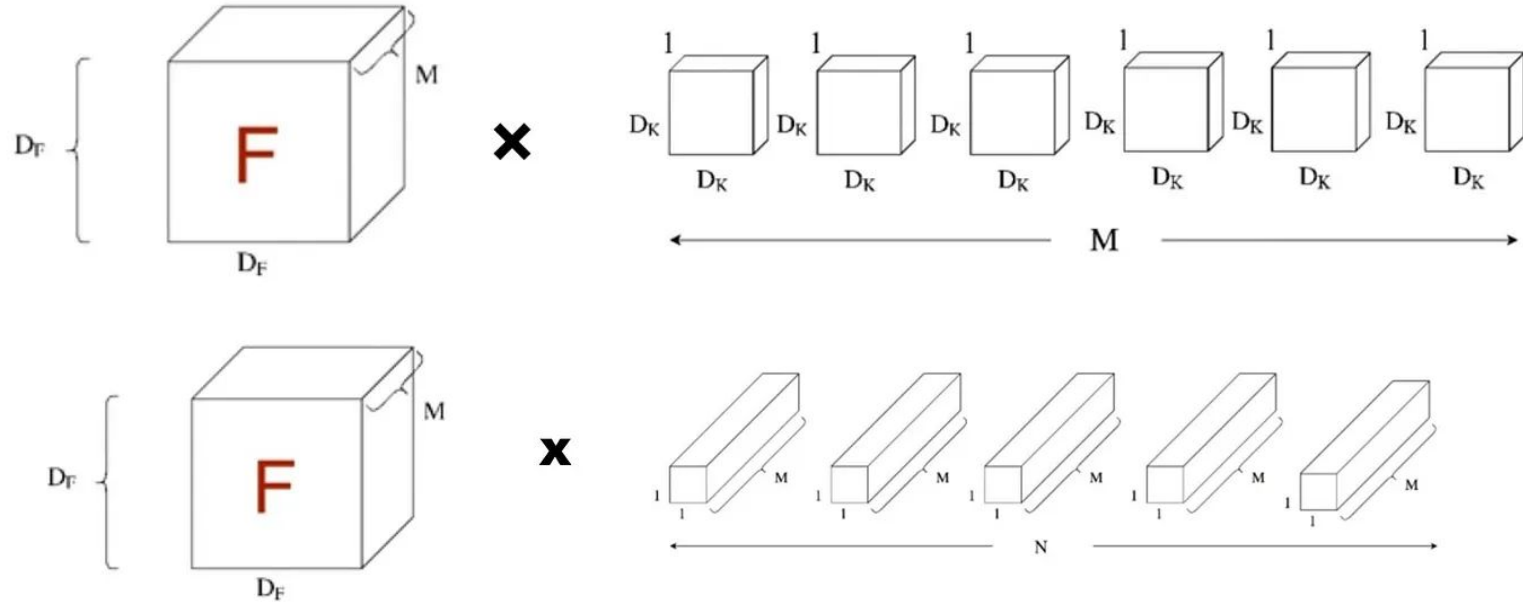
Learning residual $F(x)$ is easier than directly learning $H(x)$ since info about x can be passed through the network due to the shortcut/skip connection

MobileNet



$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$$

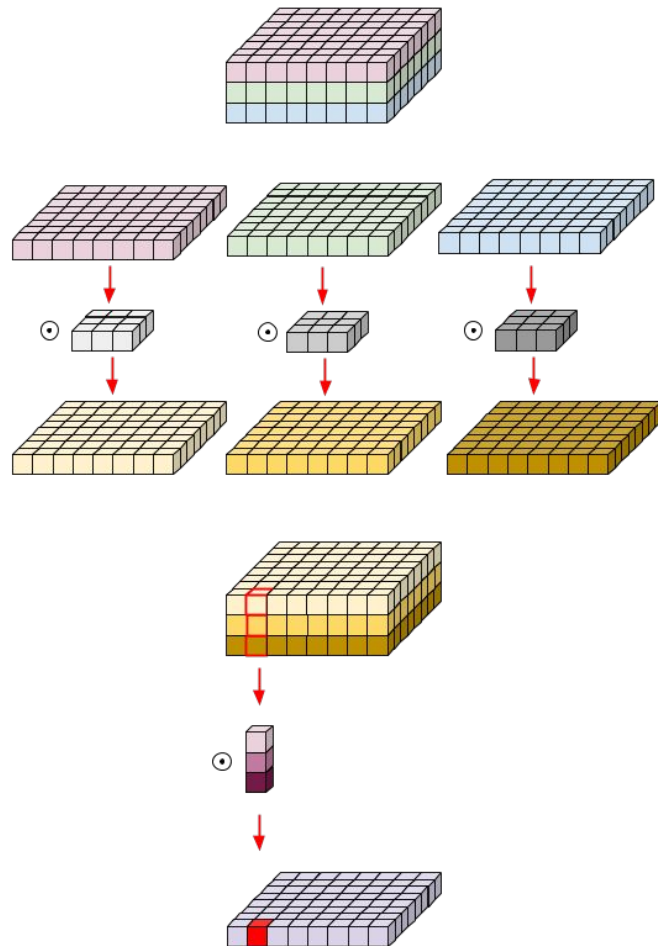
MobileNet



$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

MobileNet

Depthwise Separable Convolutional



MobileNet

2017, Google

Table 8. MobileNet Comparison to Popular Models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138