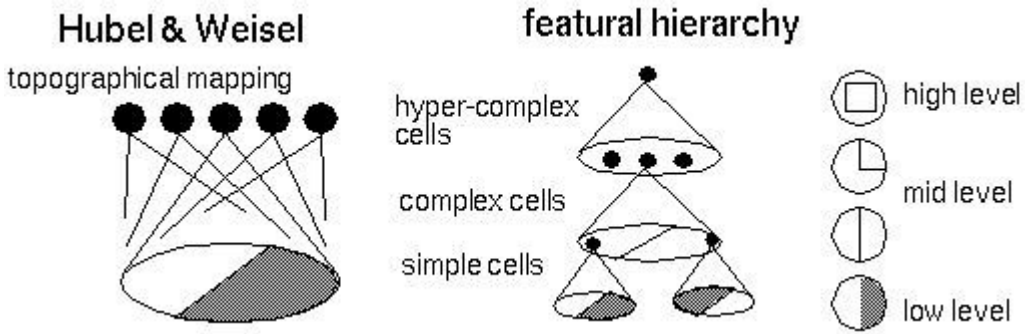
Сверточные нейронные сети (CNN)

Федоров Игорь
НИУ ВШЭ
01.11.2019

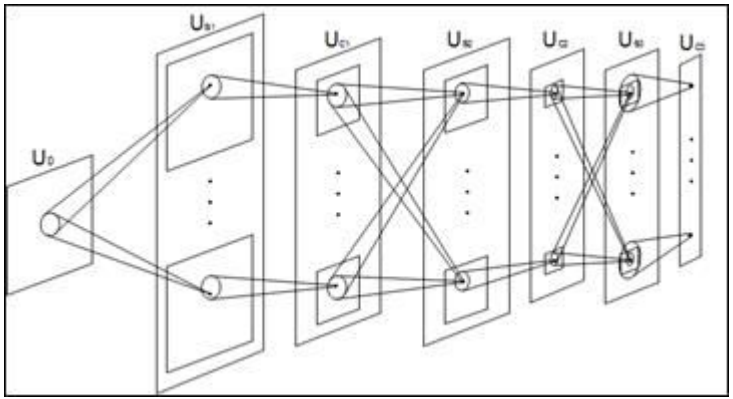
Для начала… биология

- Первичная зрительная кора - каскад простых и сложных клеток
- Эти клетки образуют иерархическую структуру
- Соседние нейроны обрабатывают соседние области изображения
- Создается пространственная карта поля зрения



Когнитрон и неокогнитрон (1980)

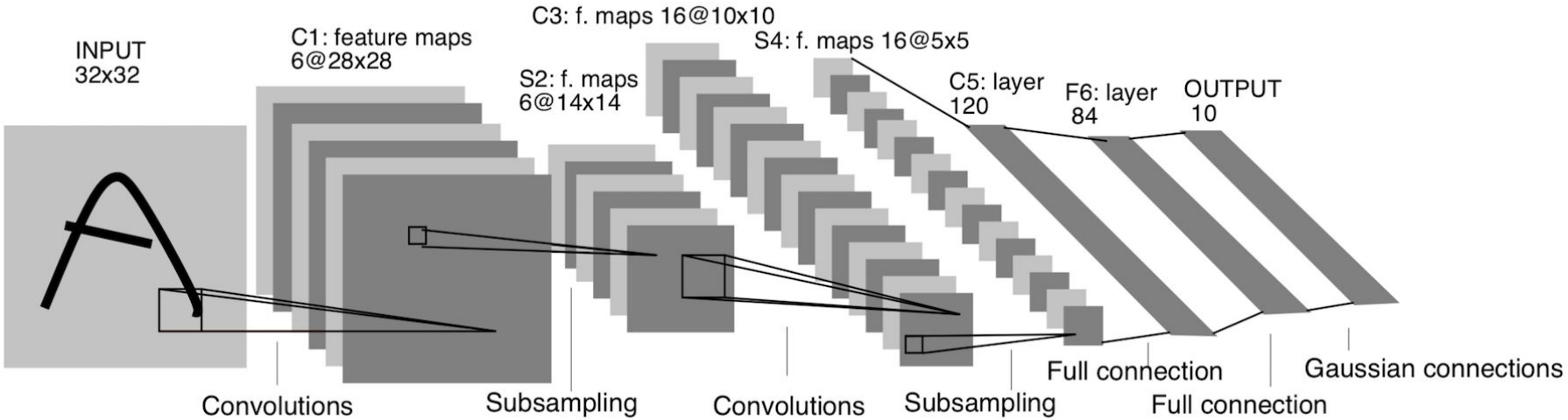
- Попытка математически реализовать структуру зрительной коры
- Задача распознавания образов
- Обучение без учителя



Ян ЛеКун и архитектура LeNet (1998)

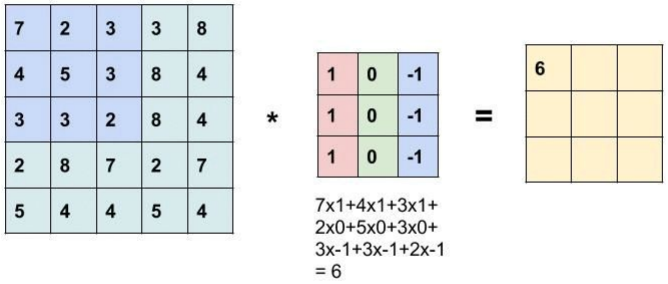
- Создавалась для задачи распознавания рукописных цифр - MNIST
- Обучение с учителем - метод обратного распространения ошибки
- LeNet-5 - ~60k параметров

LeNet-5


INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

Сверточные слои (convolution)

- Аналог “простых клеток” зрительной коры - распознавание признака
- На первом шаге используются простые примитивы
- На последующих - более высокоуровневые образы
- Ядро 3x3, 5x5, 7x7… Может быть многоканальным
- Шаг не более половины размера
- Создание рамки (padding)

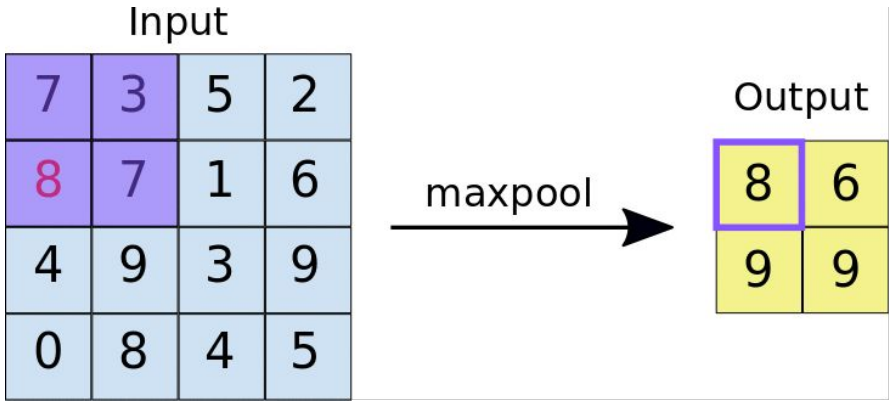


$$\frac{W - F + 2P}{S} + 1$$

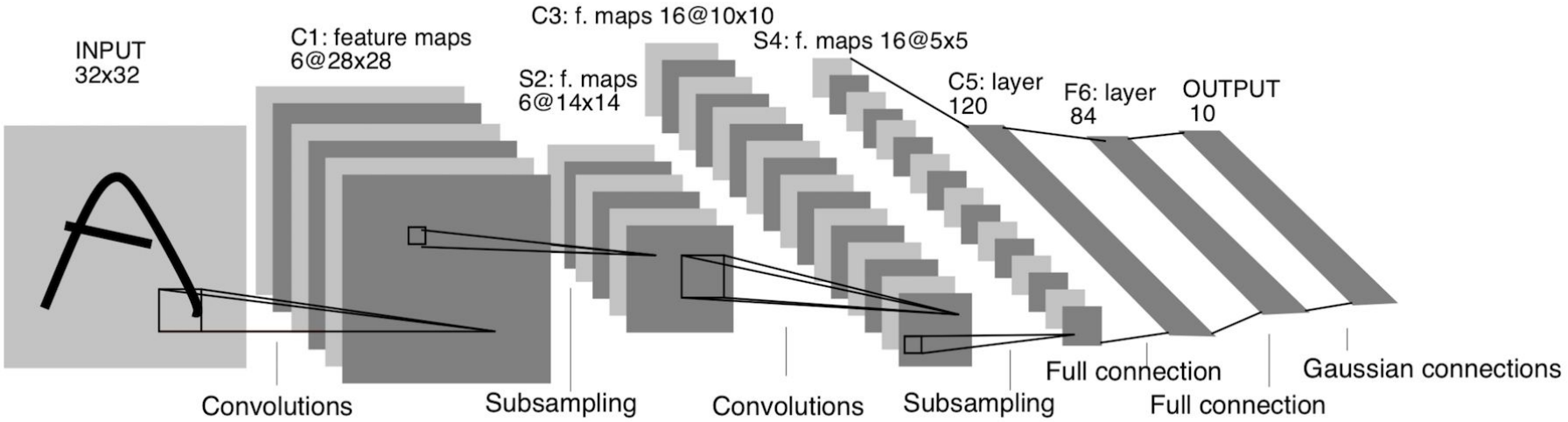
Субдискретизирующие слои (pooling)

- Аналог “сложных клеток” - уплотнение признаковой карты
- Работает благодаря локальной корреляции пикселей
- Шаг не больше размера окна

- MaxPool > AvgPool



Как это работало в LeNet?



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

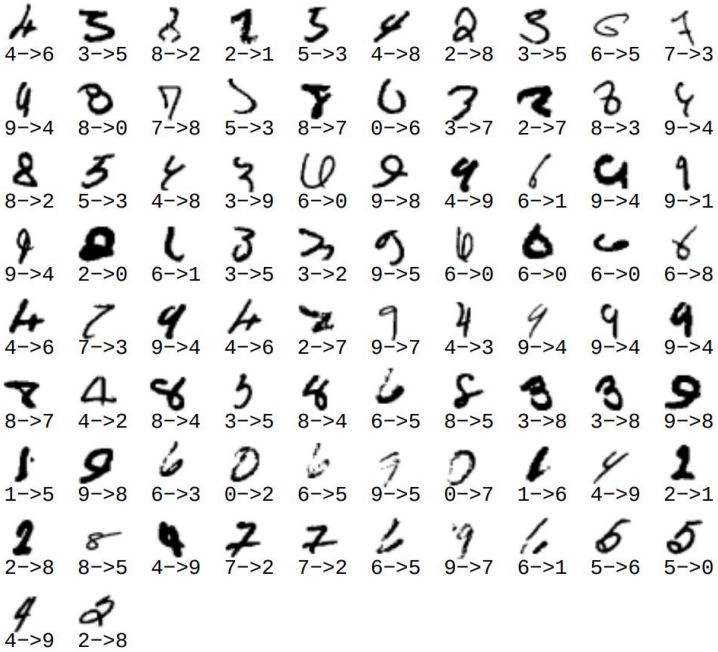
FC & Softmax Layers

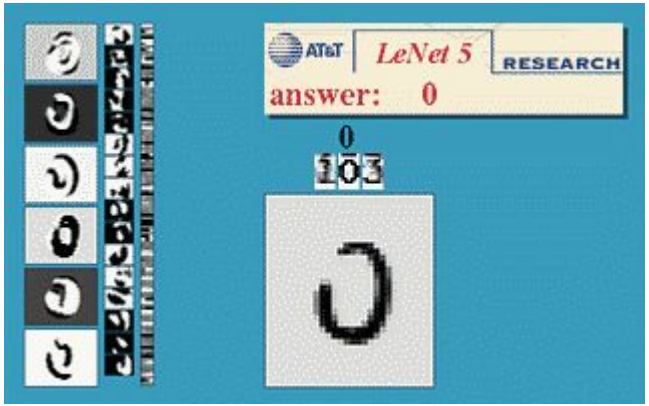
- В самом конце сети нас ждут полносвязные слои (один или несколько)
- Softmax - получение вероятностей принадлежности к классам
- Впрочем, ЛеКун использовал Euclidean Radial Basis Function

$$y_i = \sum_j (x_j - w_{ij})^2$$

А результаты можно глянуть?

- Ошибка на тестовой части MNIST для LeNet5 - 0.95%



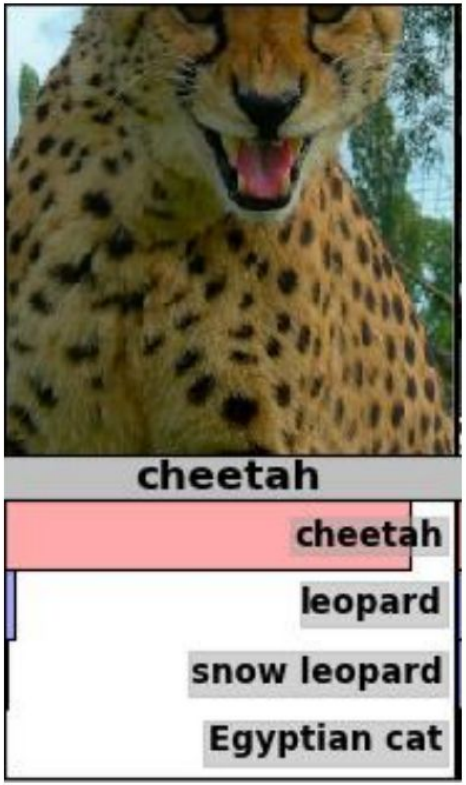


ImageNet Large Scale Visual Recognition Challenge

- >1.2M hi-res изображений

- Задача 1 - угадать, что на них изображено (1000 классов)
- Метрики - top-1 и top-5 accuracy

- Задача 2 - на каждом изображении выделить область с описываемым объектом
- Метрика - размер перекрытия


cheetah
cheetah
leopard
snow leopard
Egyptian cat


bullet train
bullet train
passenger car
subway train
electric locomotive


hand glass
scissors
hand glass
frying pan
stethoscope

AlexNet(2012) - идеи Крижевского

- Топология - как у LeNet, но размер гораздо больше (~ в 1000 раз)
- Обучение на GPU
- Аугментация
- ReLU
- Dropout
- MaxPooling с перекрыванием

Зачем GPU?

- GPU созданы для ускоренного расчета графики (шейдеры, etc.)
- FLOPS, Bandwidth, Parallelism
- Обучение нейросети - это примерно то же самое
- TPU, NPU… - еще быстрее

- AlexNet обучалась на двух GPU - 60M параметров

Почему важна аугментация?

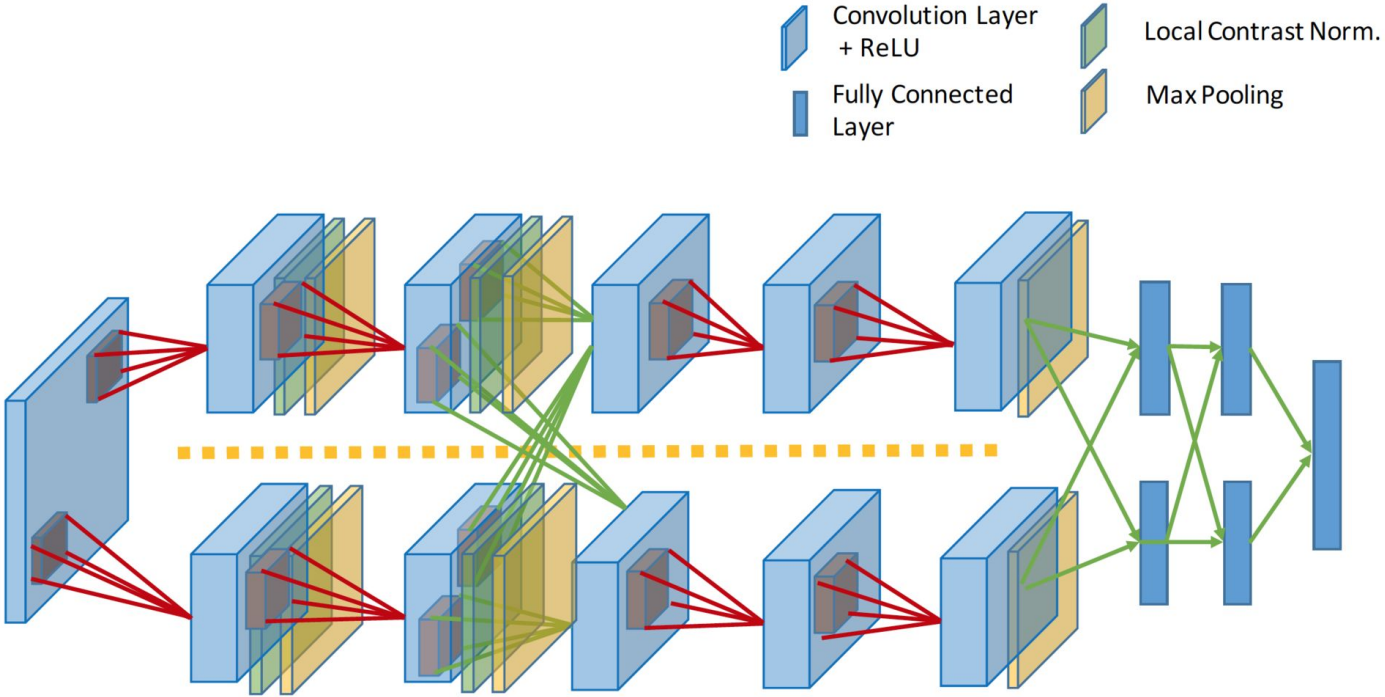
- Вместо картинки 256x256 - 10 картинок 224x224
- Color deformation
- Простой способ увеличить количество данных и глубже обучить сеть

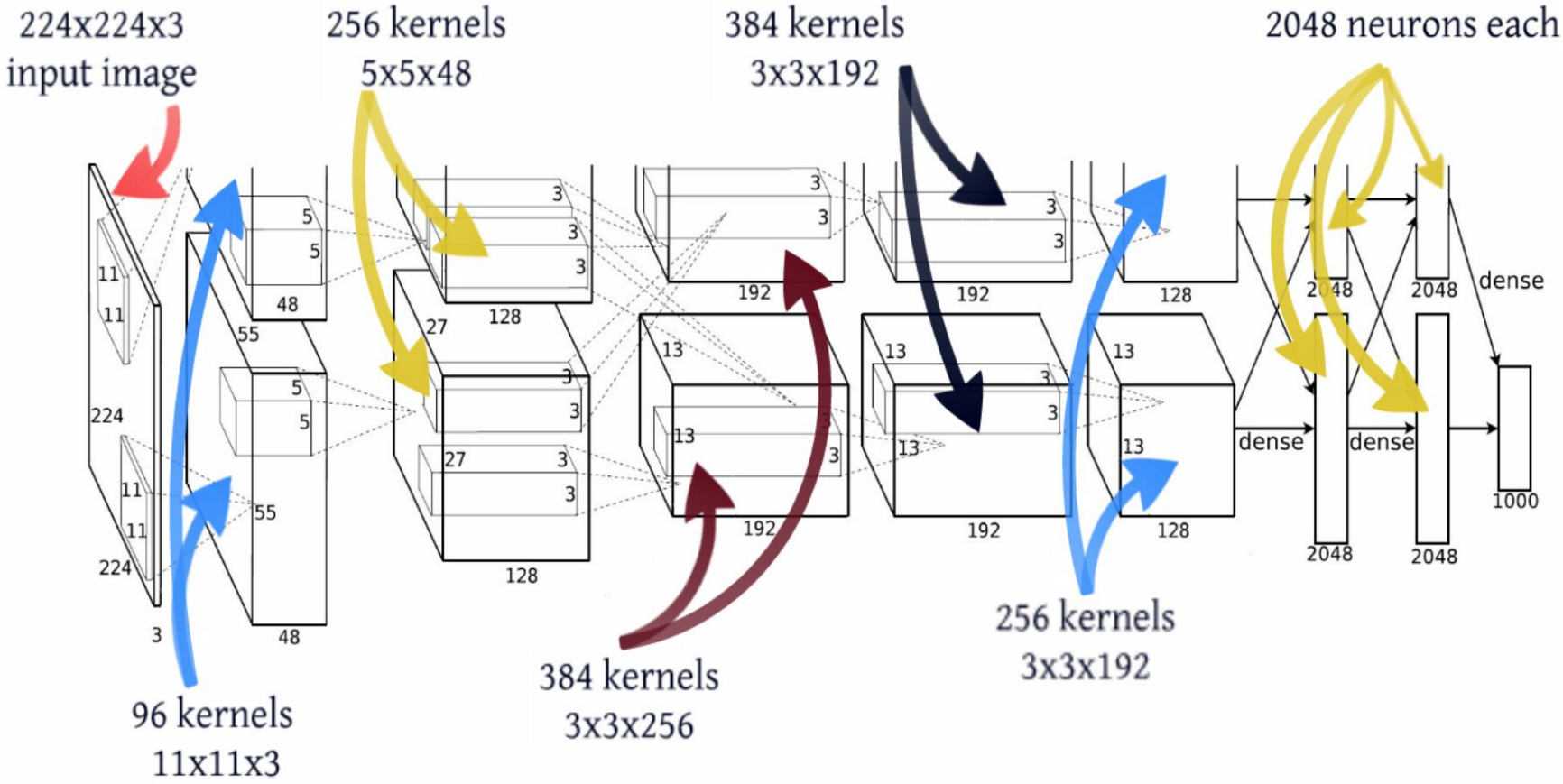

Crop
Symmetry
Rotation
Scale
Original
Noise
Hue
Obstruction
Blur

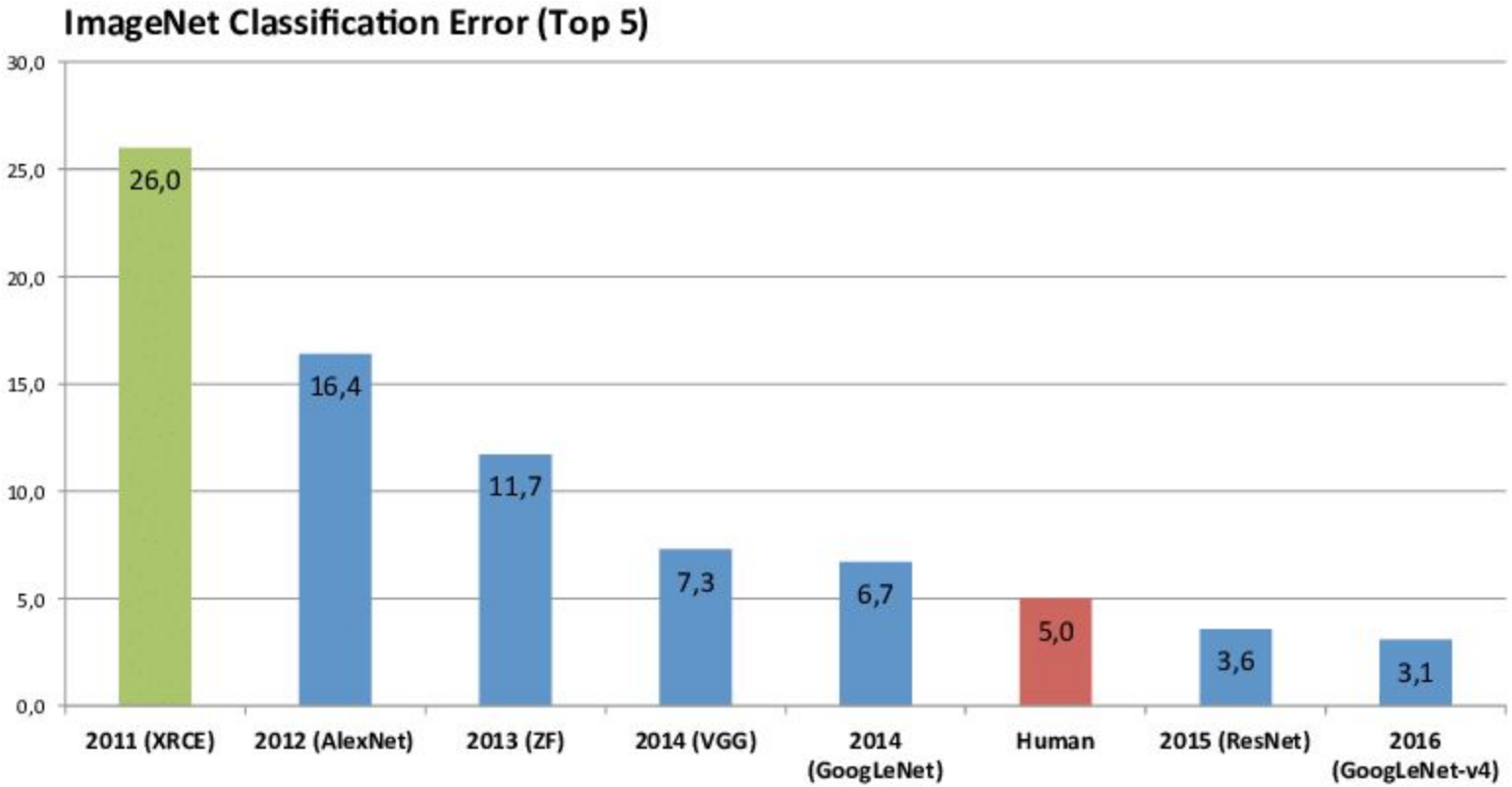
Взглянем же на этого красавца!

У меня есть статья с более чем 40k цитирований, а у тебя нет

Ну и сеть его тоже оценим

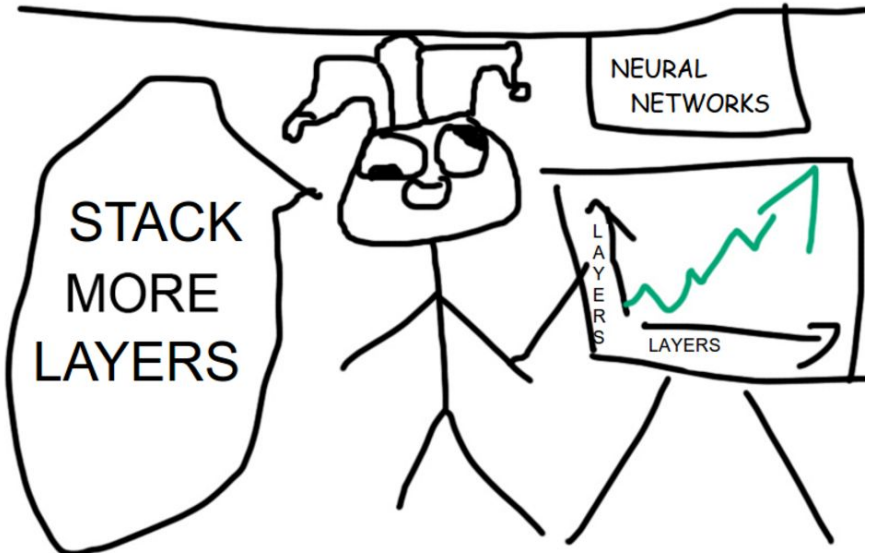


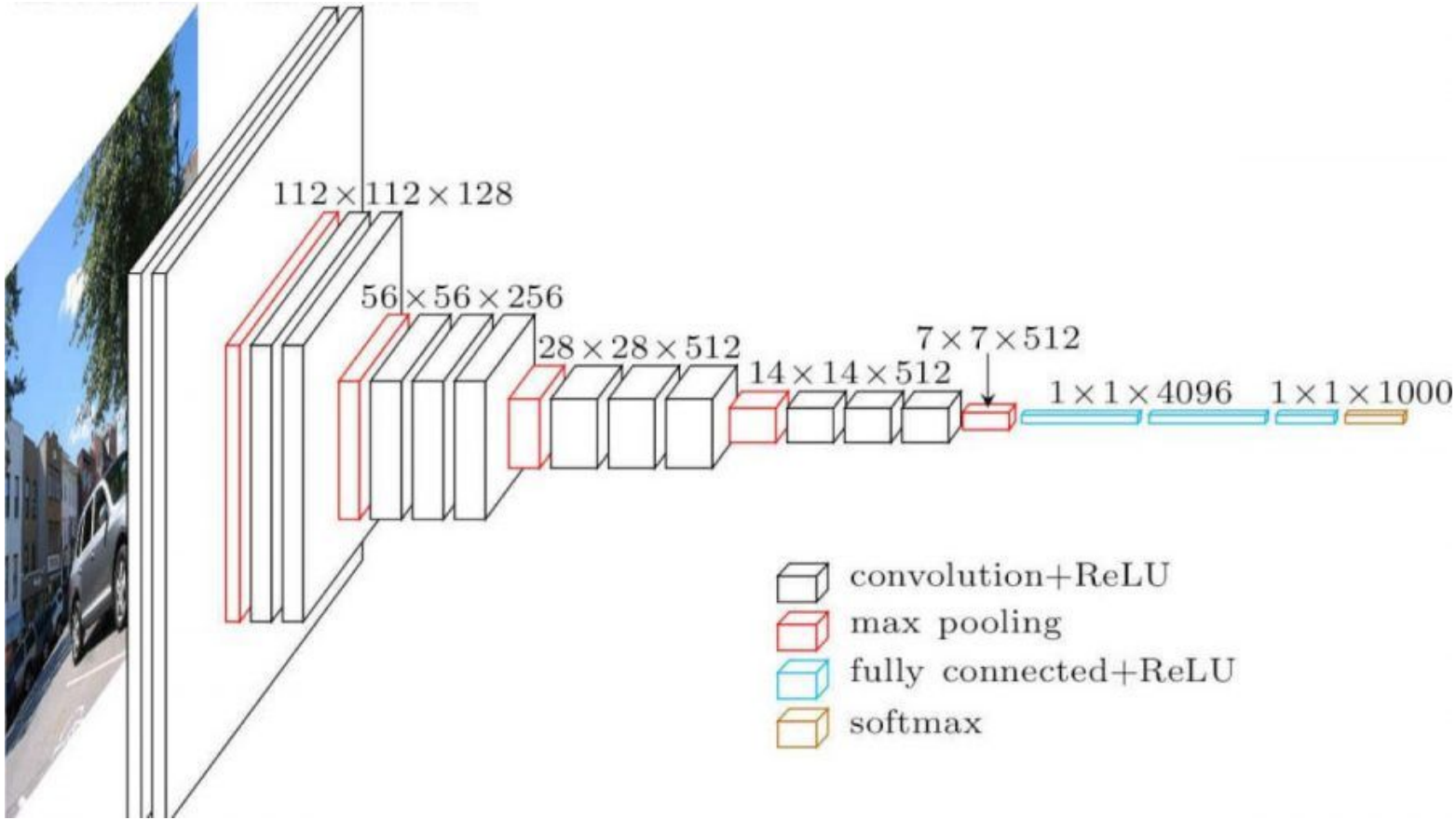

224x224x3
input image
256 kernels
5x5x48
384 kernels
3x3x192
2048 neurons each
96 kernels
11x11x3
384 kernels
3x3x256
256 kernels
3x3x192
dense
dense
dense
1000


ImageNet Classification Error (Top 5)
30,0
25,0
20,0
15,0
10,0
5,0
0,0
26,0
16,4
11,7
7,3
6,7
5,0
3,6
3,1
2011 (XRCE)
2012 (AlexNet)
2013 (ZF)
2014 (VGG)
2014 (GoogLeNet)
Human
2015 (ResNet)
2016 (GoogLeNet-v4)

VGG (2014)

- Свертка 5x5 - 25 параметров
- Последовательность сверток 3x3 - 18 параметров
- Последняя работа, использующая “стандартную топологию”
- VGG19 - 144M параметров



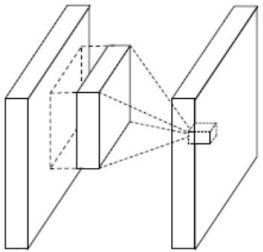

112×112×128
56×56×256
28×28×512
14×14×512
7×7×512
1×1×4096
1×1×1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

Проблемы стандартной топологии

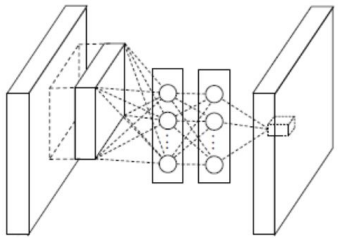
- Увеличение в ширину - вероятнее переобучаемся
- Увеличение в глубину - затухающие градиенты
- Качество уже не растет
- Пора придумать что-то новое!

Network In Network - MLPConv

- Сверточный слой - по сути линейная модель над признаками, работающая за счет избыточного представления
- Заменим линейный фильтр на многослойный перцептрон
- Cascaded Cross Channel Parametric Pooling (CCCP Pooling)
- CNN, у которой сверточные слои - CNN



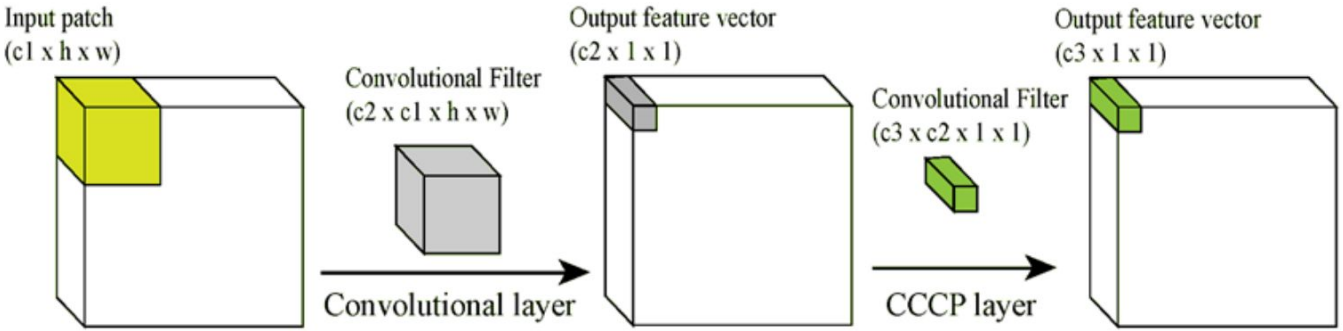
Linear Convolutional Layer



mlpconv Layer

$$f_{i,j,k} = \max(w_k^T x_{i,j}, 0).$$

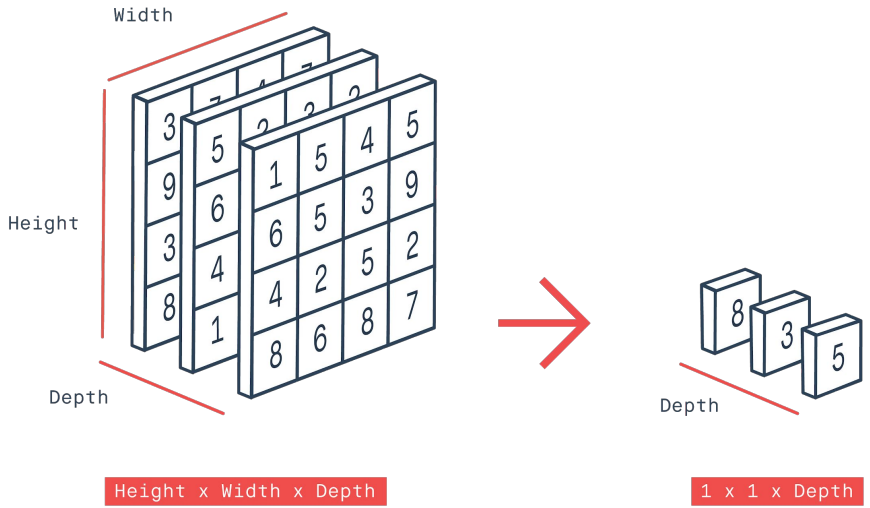
$$\begin{aligned} f^1_{i,j,k_1} &= \max({w^1_{k_1}}^T x_{i,j} + b_{k_1}, 0). \\ &\vdots \\ f^n_{i,j,k_n} &= \max({w^n_{k_n}}^T f^{n-1}_{i,j} + b_{k_n}, 0). \end{aligned}$$



Efficient implementation of CCCP

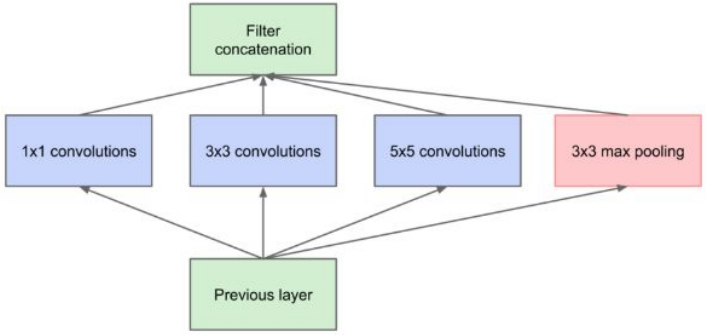
Network In Network - Global Average Pooling

- В сетях стандартной топологии 80% вычислений уходят на сверточные слои, а 80% памяти - на полносвязные
- Давайте от них избавимся

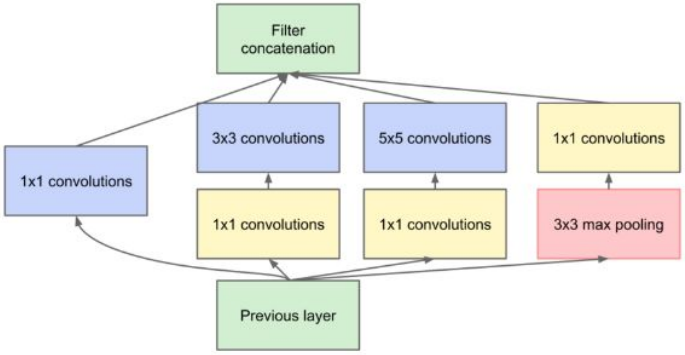


Inception / GoogLeNet (2014)

- Наследие идей NIN - 1x1 свертки и GAP
- Что если у признаков разный масштаб?
- Применим свертки разного размера параллельно

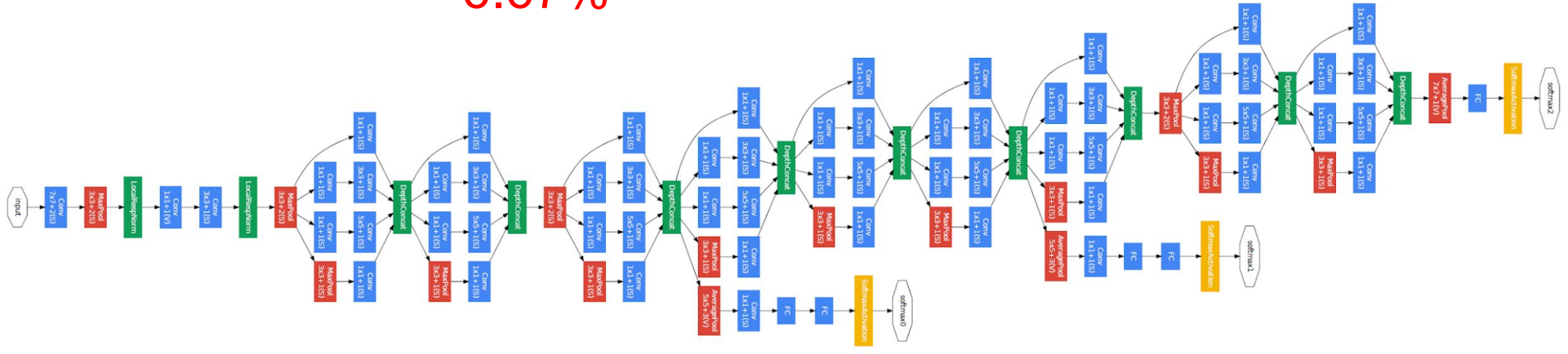


(a) Inception module, naïve version



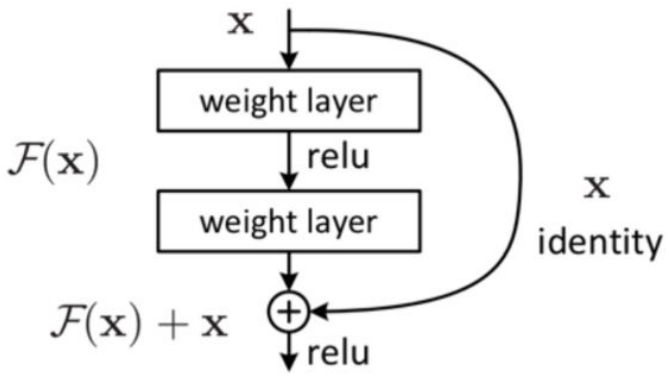
(b) Inception module with dimension reductions

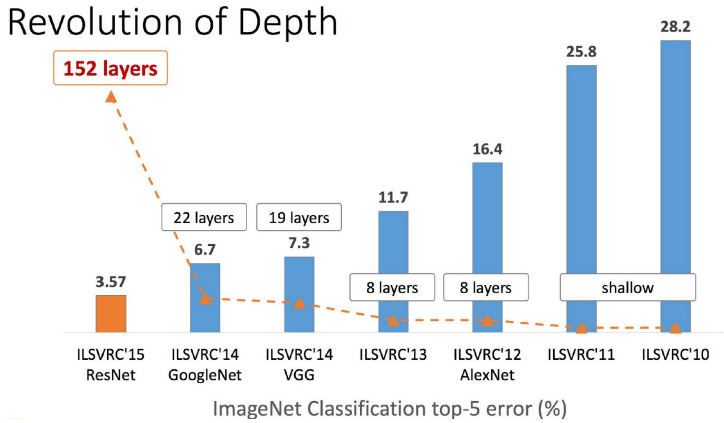
6.67%



ResNet

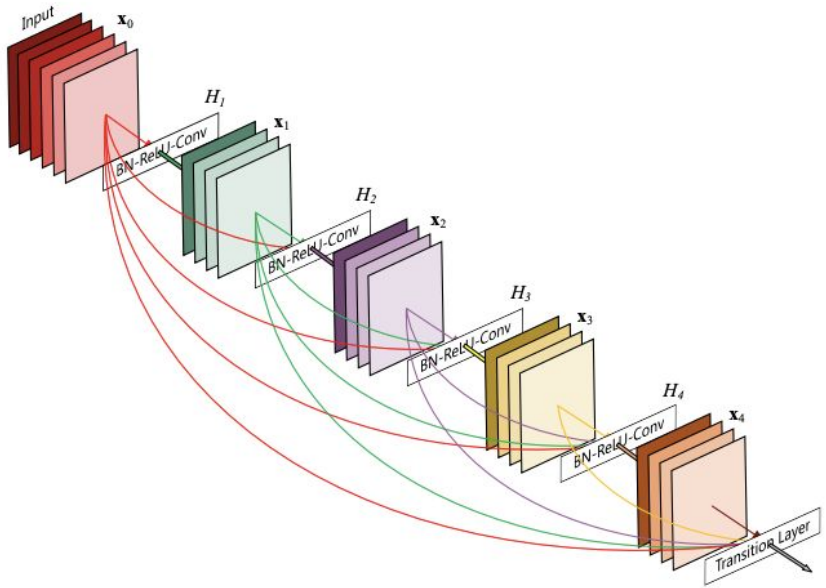
- Нейросеть может аппроксимировать почти любую H(x)
- Также она может аппроксимировать F(X) = H(x) - x
- Целевая функция будет равна F(x) + x





DenseNet

- А что если соединить каждый слой с каждым?
- Параметров меньше, слоев больше, качество лучше



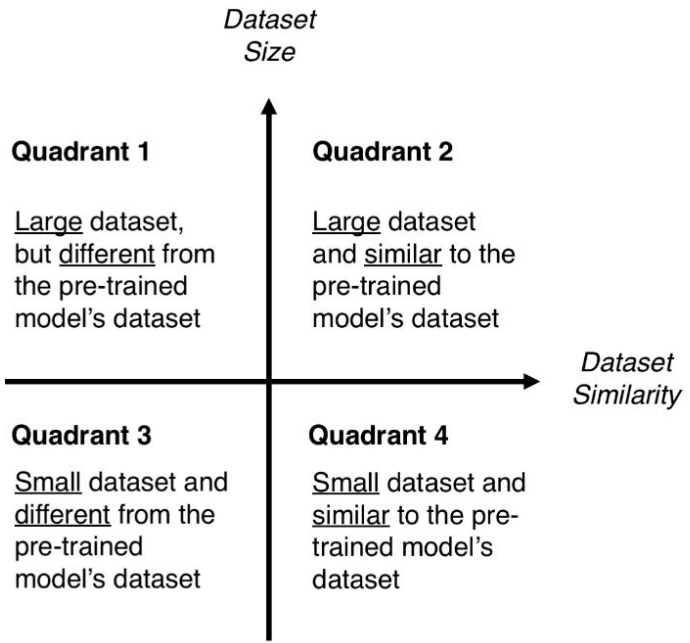
Хорошо, но как этим пользоваться?

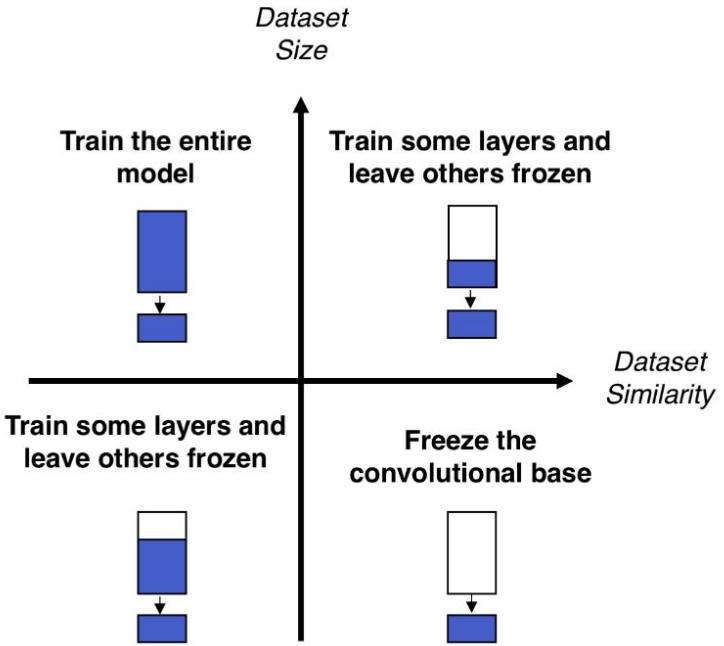
- Готовые CNN-модели есть во всех популярных фреймворках
- С нуля обучать часто не нужно - fine-tuning
- Не забывайте про GPU

Documentation for individual models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

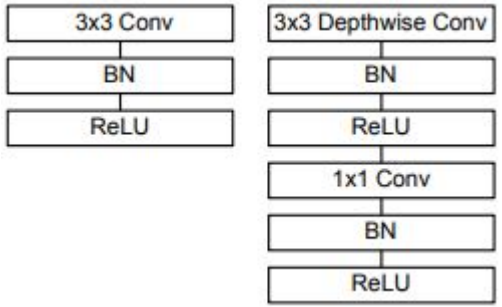
Transfer learning





MobileNet

- SOTA-результаты достигнуты, время заняться оптимизацией
- Depthwise separable convolution - сначала сворачиваем каждый канал 3x3 сверткой, затем меняем глубину 1x1 сверткой
- Множитель ширины и множитель разрешения
- Размеры позволяют работать на телефоне



Вопросы

1. Размерность входа сверточного слоя - 32x32x1. Размерность выхода - 28x28x6. Сколько в этом слое было применено сверточных фильтров, с какой размерностью и шагом?

2. Опишите идею и работу блока Inception.

3. Почему в большинстве задач, решаемых с помощью CNN, рекомендуется использовать предобученные модели?

Источники

- Gradient-Based Learning Applied to Document Recognition - LeCun et al., 1998
- ImageNet Classification with Deep Convolutional Neural Networks - Krizhevsky et al., 2012
- Network in Network - Lin et al., 2014
- Going deeper with convolutions - Szegedy et al., 2014
- Deep residual learning for image recognition - He et al., 2015

Convolutional Neural Networks (CNN)

Федоров Игорь
НИУ ВШЭ
01.11.2019