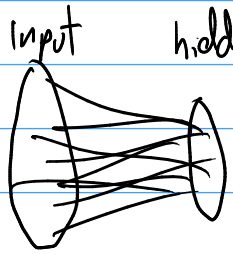
Convolutional Neural Networks

Problems beyond FC layers

Image classification

Input: Image 1 Megapixel $\approx 10^6$ dimensional



$10^6 \cdot 10^3 = 1$ billion params

GPT-2 $\approx$ 1.5 B params

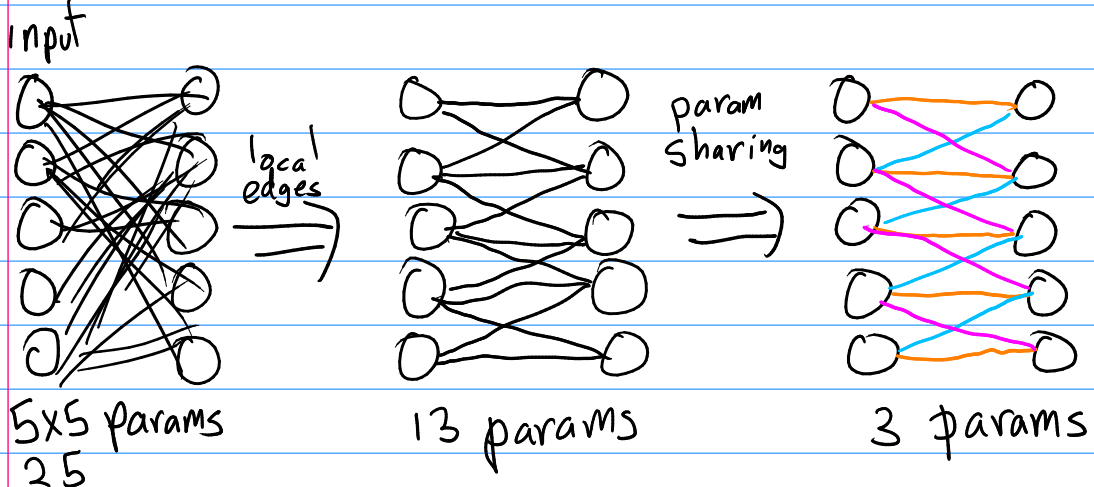
So?

- Huge cost to train
- Need lots of data
- Difficulties w/optimization
- Likely overfit

1. Translation invariance
2. Locality

Principles: parameter sharing, only local edges

Simple e.g.:



Convolutions

$$y(i) = (x * w)(i) = \int x(t)\, w(i-t)\, dt$$

$$= \sum x[t]\, w[i-t]$$

Signal processing

Use "cross-correlation" operator.

Input

0	1	2
3	4	5
6	7	8

$*$ kernel / Map filter

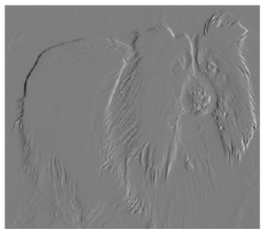
0	1
2	3

$=$

19	25
37	43

$0 \cdot 0 + 1 \cdot 1 + 3 \cdot 2 + 4 \cdot 3 = 19$

$4 \cdot 0 + 5 \cdot 1 + 2 \cdot 7 + 8 \cdot 3 = 43$



-1	1

Convolutional Neural Nets

Input: Images

3D objects (tensor: generalization of matrix)

width, height, depth ← channels

3 for color RGB, 1 for B+W

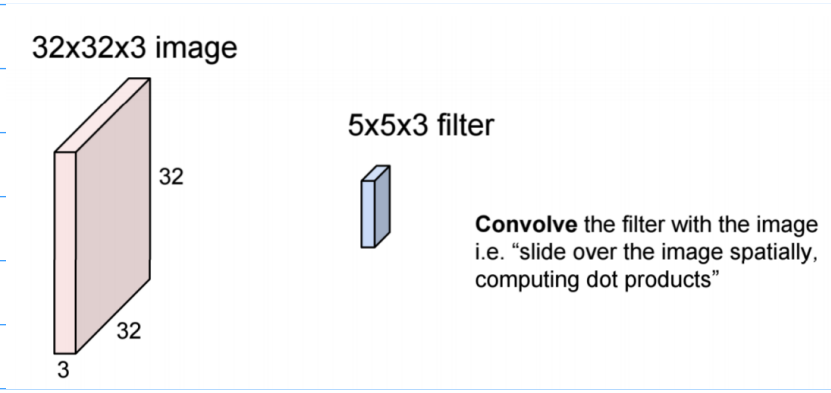
More than 3 channels in later layers

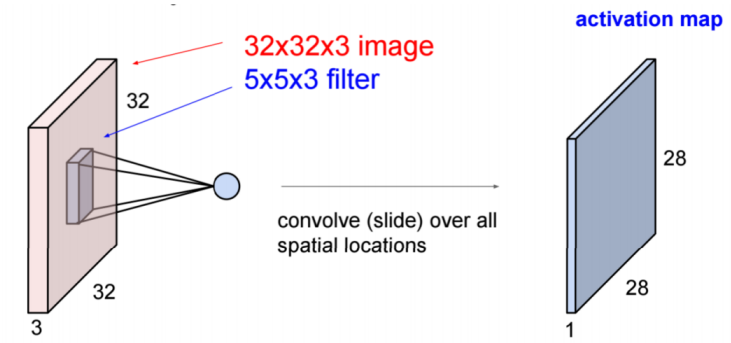
MNIST: 28x28x1

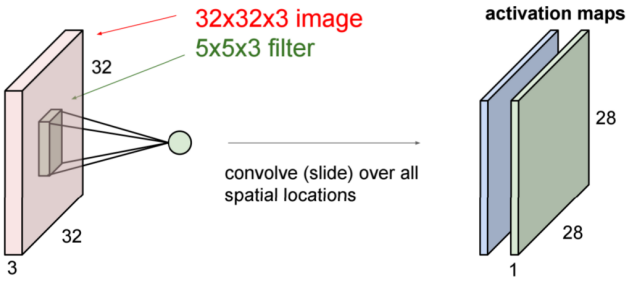
CIFAR-10: 32x32x3

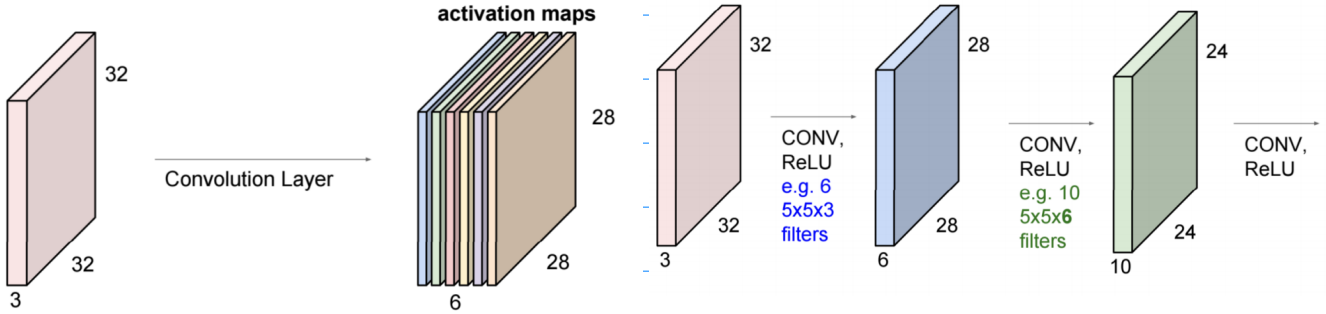
Output: 1x1xC ← # of classes

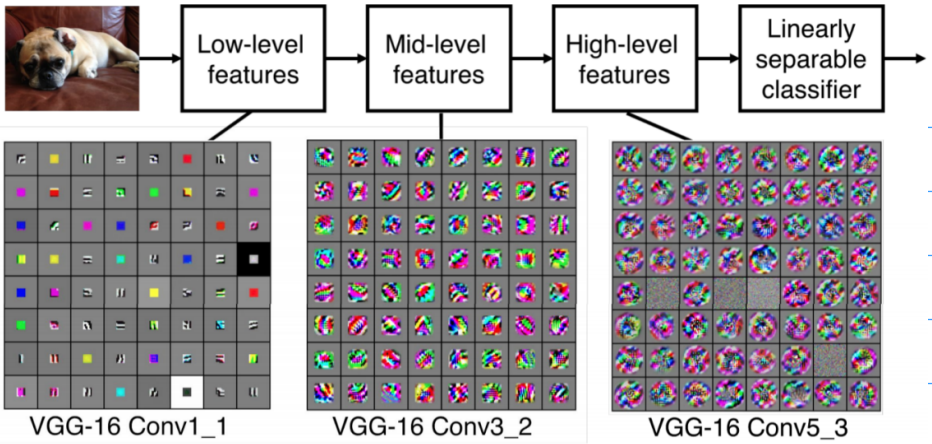
3 Main layer types
- Fully connected
- Convolutional
- Pooling





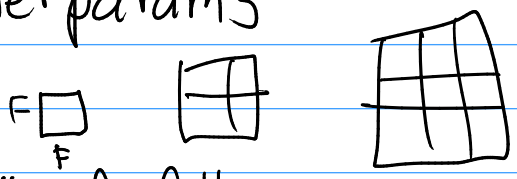




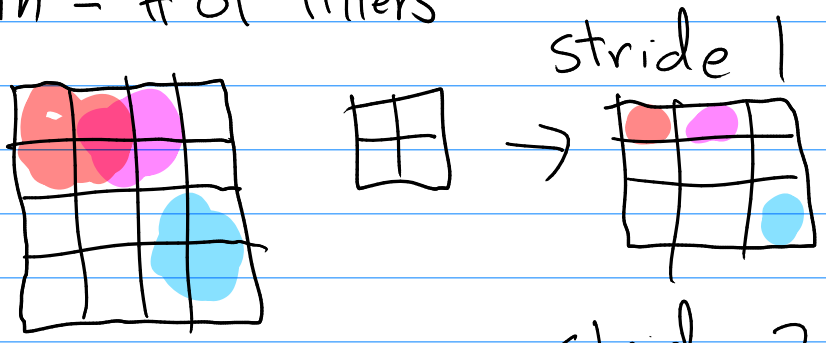


Conv layer hyperparams

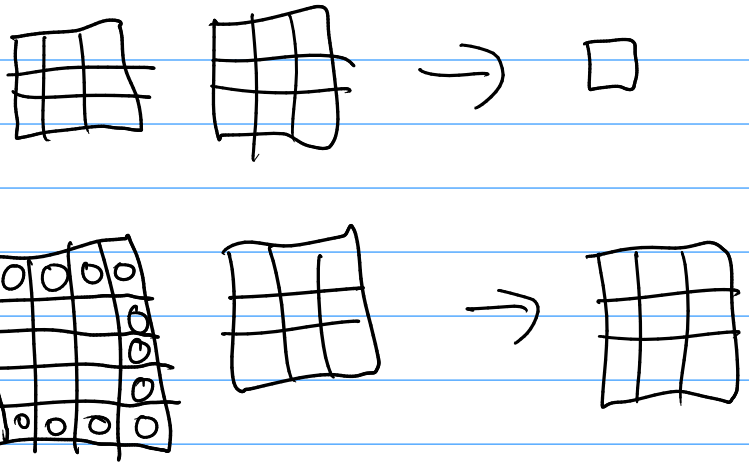
- Filter size

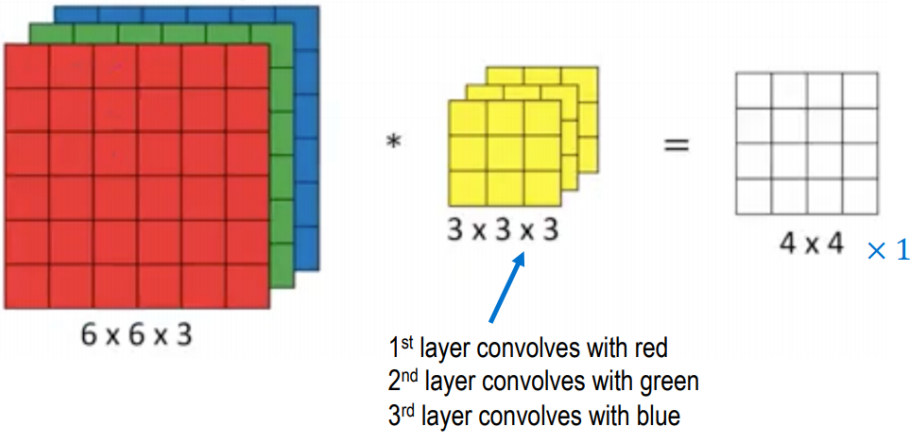


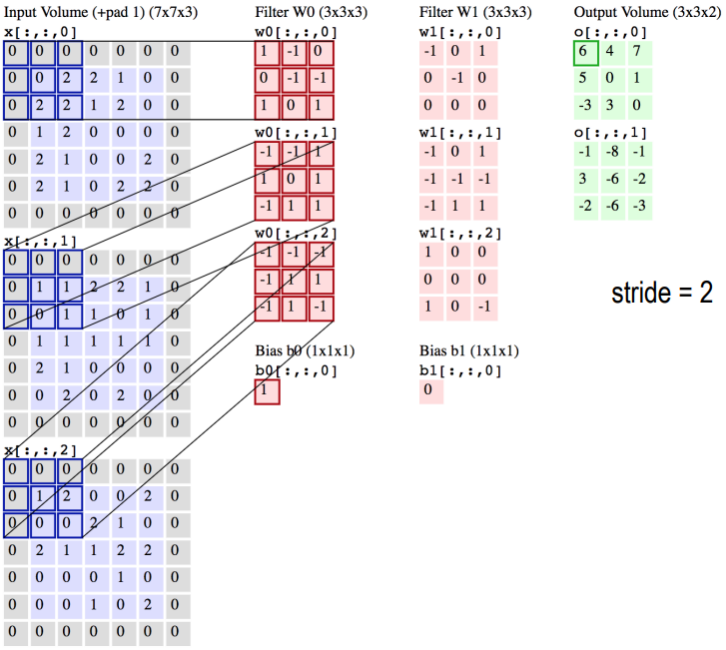
- Output depth = # of filters
- Stride



- Zero padding







Conv Layer Summary:

Input: $W_1 \times H_1 \times D_1$

Four hyper params:

- # of filters K
- width/height F ($F \times F \times D_1$)
- Stride S
- Zero padding P

Output: $W_2 \times H_2 \times D_2$

- $W_2 = (W_1 - F + 2P)/S + 1$
- $H_2 = (H_1 - F + 2P)/S + 1$
- $D_2 = K$.

# of params:

K filters, $F \times F \times D_1$ Params

$F^2 D_1 K$ weights

K biases

Example computation:

Input: 227x227x3.

Use: 96 filters, 11x11x3, stride 4, no padding.

Output:

$W_2 = (227 - 11 + 2 \cdot 0)/4 + 1 = \frac{216}{4} + 1 = 55.$

$H_2 = 55$

$D_2 = 96$

$(227, 227, 3) \Rightarrow (55, 55, 96)$

# of params: $11^2 \cdot 3 \cdot 96 + 96 = 34944$

FC layer: (input size + 1)(output size) $\approx$ 45 bill.

No weight sharing, only local connections

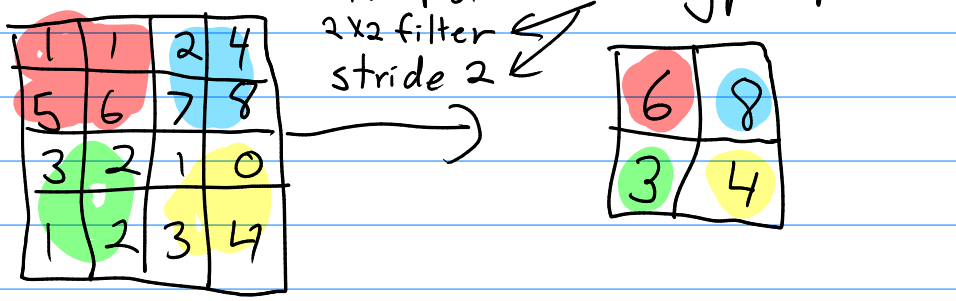
$(11 \cdot 11 \cdot 3 + 1)(55 \cdot 55 \cdot 96) = 106$m params

param/filter

Pooling Layer

Max pool
Avg pool
L2-norm pool

- No learnable params
- Only 1 filter



Pool Summary

Input: $W_1 \times H_1 \times D_1$

2 Hyperparams:
- Width/height F
- Stride S

Output $W_2 \times H_2 \times D_2$

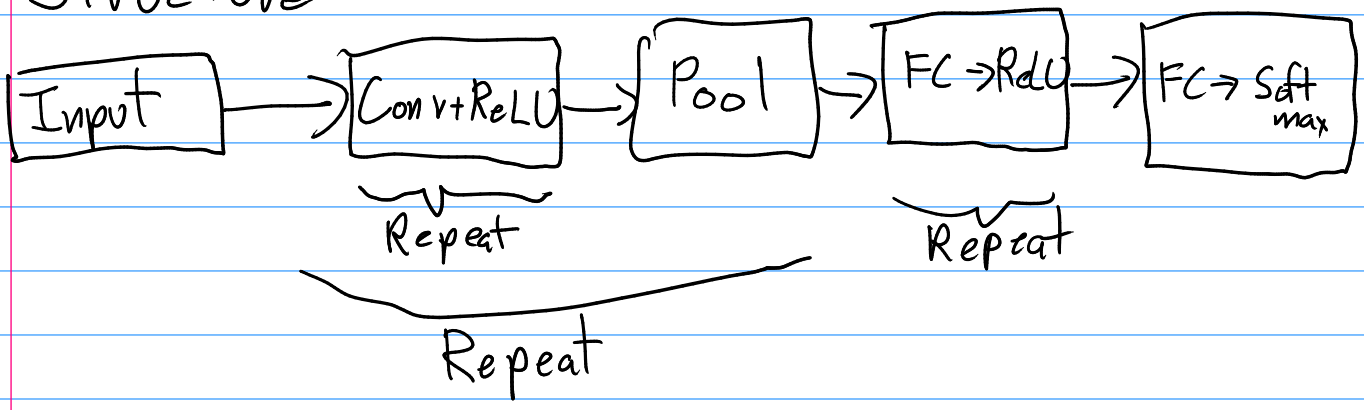
$W_2 = (W_1 - F)/S + 1$

$H_2 = (H_1 - F)/S + 1$

$D_2 = D_1$

No learnable params

Structure



Datasets: MNIST

60k train, 10k test

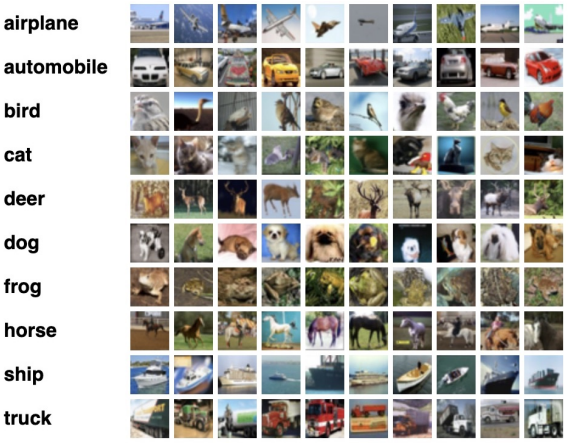
10 classes

28×28×1 . 99.9%

CIFAR-10:

32x32x3

50k train, 10k test 10 classes



ImageNet: 14m images

~400x500x3

10's of thousands of classes

ILSVRC dataset

2012-2017

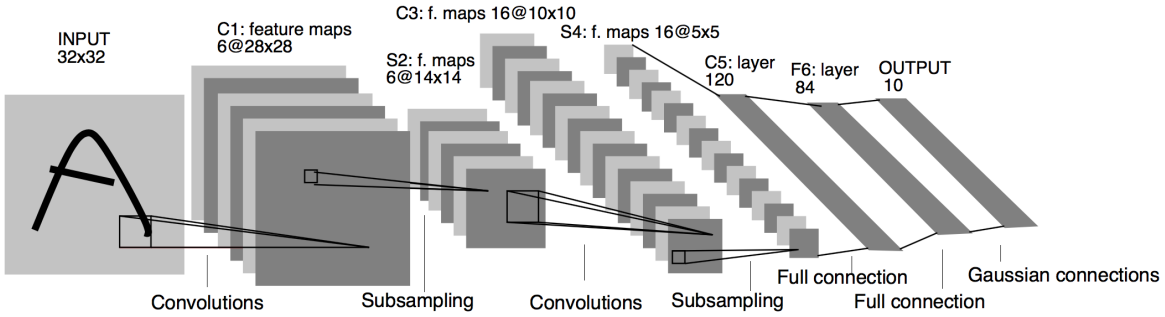
1.2m training

50k test set

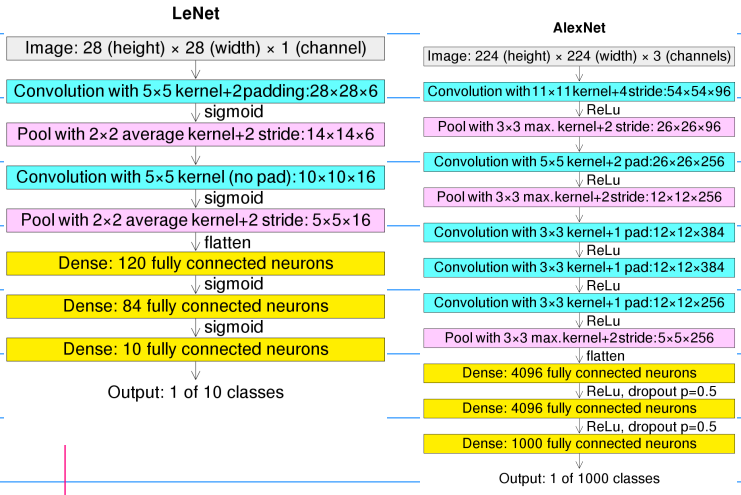
1000 classes

Architectures

LeNet (1998)



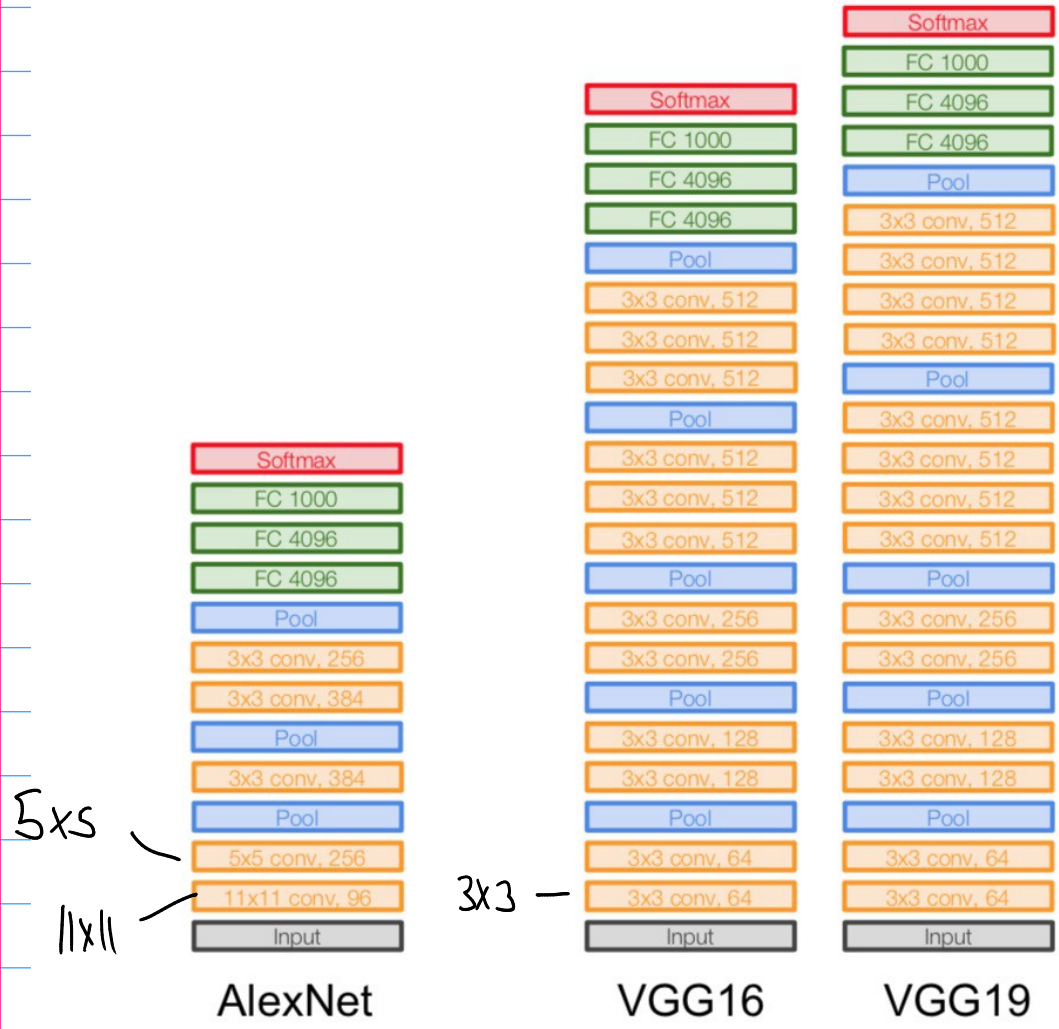
AlexNet (2012)

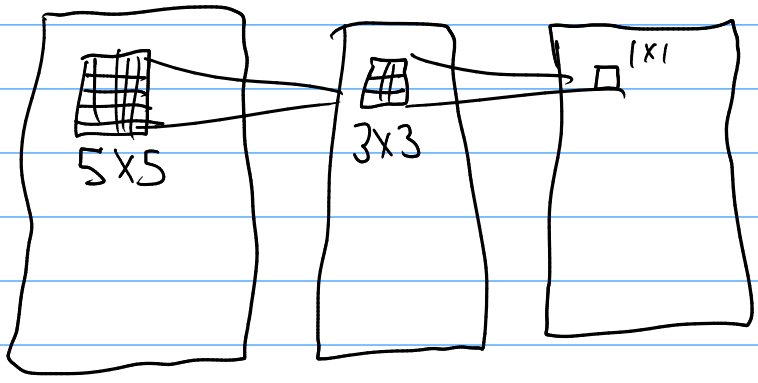


Beat competition in ILSVRC 12

Top 5 error of 15% vs 25% for best alt

VGGNet (2014)

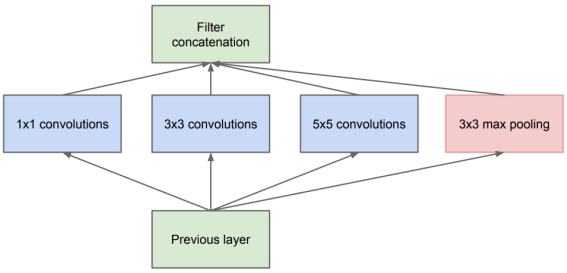




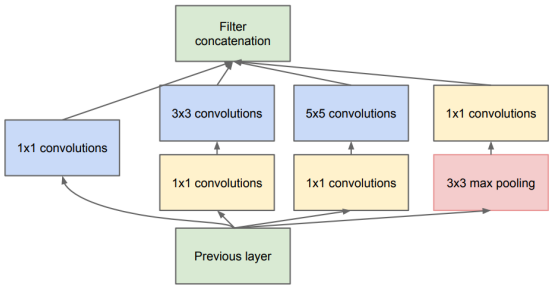
More layers + small filters
better than few layers + large filters

GoogLeNet (2014)

Inception Module



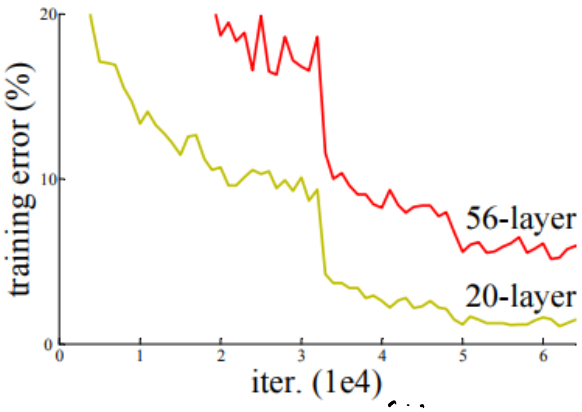
(a) Inception module, naïve version

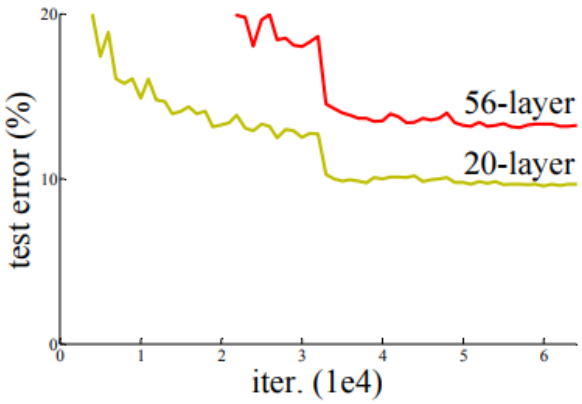


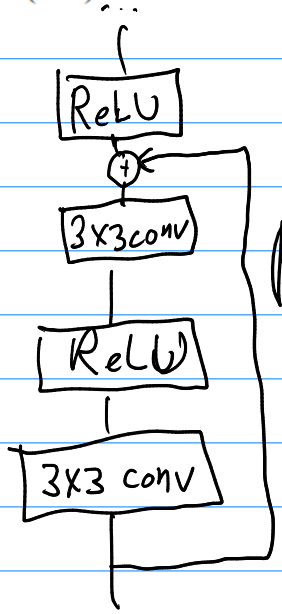
(b) Inception module with dimension reductions

W x H x D input
k filters (1x1)
W x H x k output

ResNet (2015)







Residual connection


softmax2
SoftmaxActivation
FC
AveragePool 7x7+1(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
softmax1
AveragePool 5x5+3(V)
softmax0
LocalRespNorm
Conv 7x7+2(S)
input