

# Convolutional Neural Networks

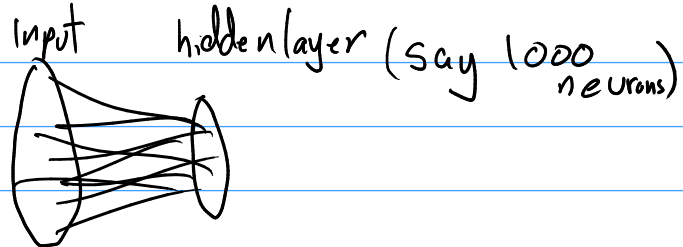
Problems beyond FC layers

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



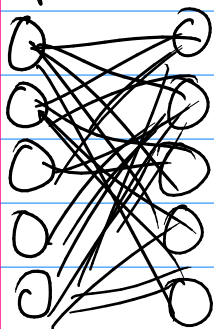
1. Translation invariance

2. Locality

Principles: parameter sharing, only local edges

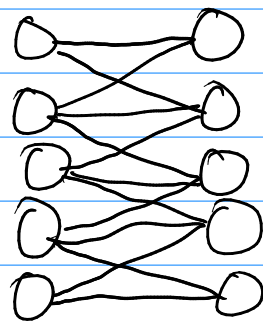
Simple e.g.:

input



5x5 params  
25

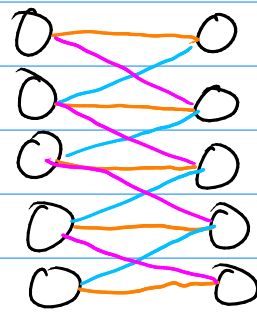
local edges



13 params

param sharing

$\Rightarrow$



3 params

## 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

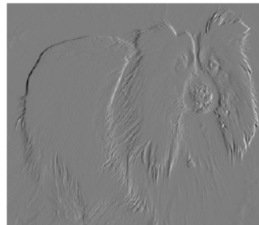
0	1
2	3

=

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

MNIST:  $28 \times 28 \times 1$

CIFAR-10:  $32 \times 32 \times 3$

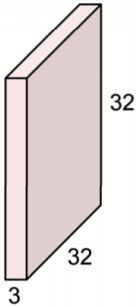
More than 3 channels in later layers

Output:  $1 \times 1 \times C \leftarrow$  # of classes

# 3 Main layer types

- Fully connected
- Convolutional
- Pooling

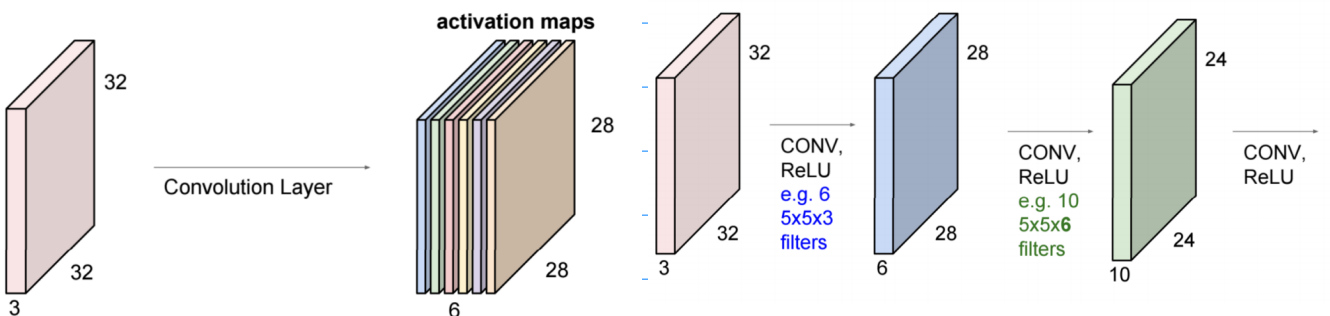
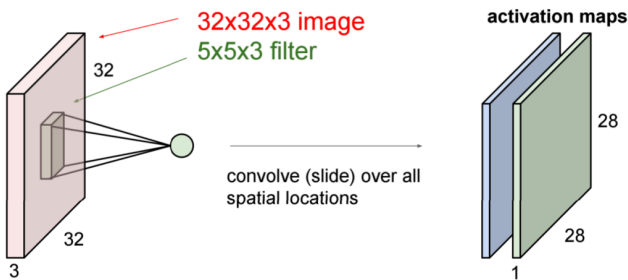
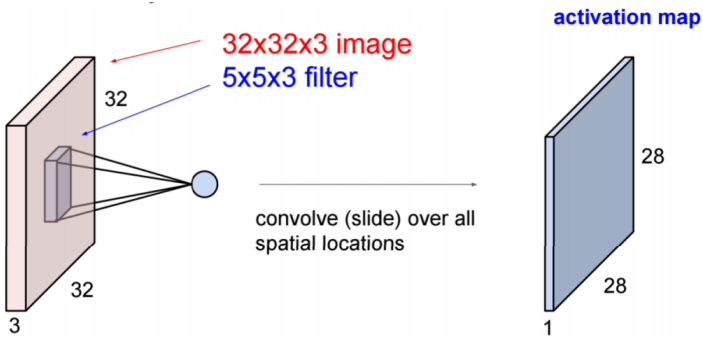
32x32x3 image

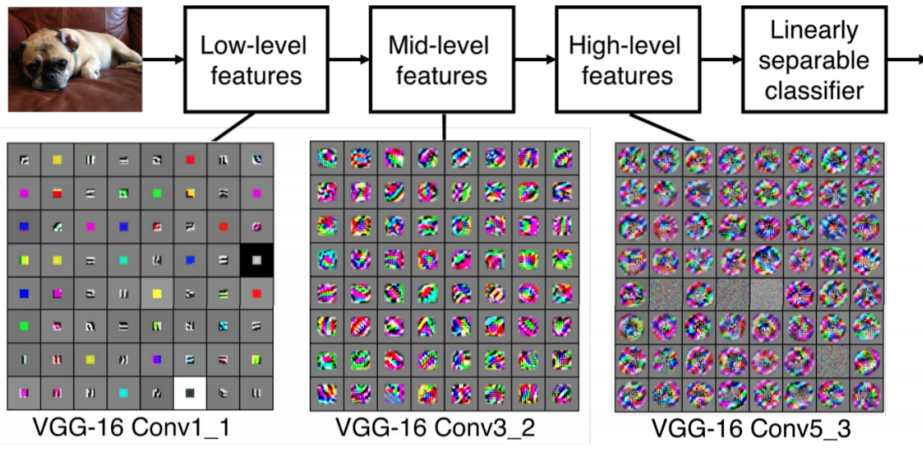


5x5x3 filter



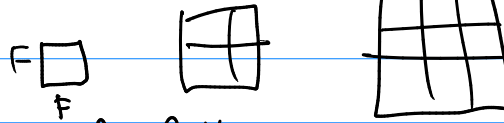
**Convolve** the filter with the image  
i.e. "slide over the image spatially,  
computing dot products"





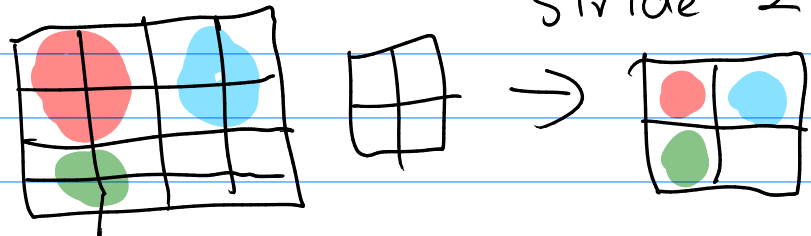
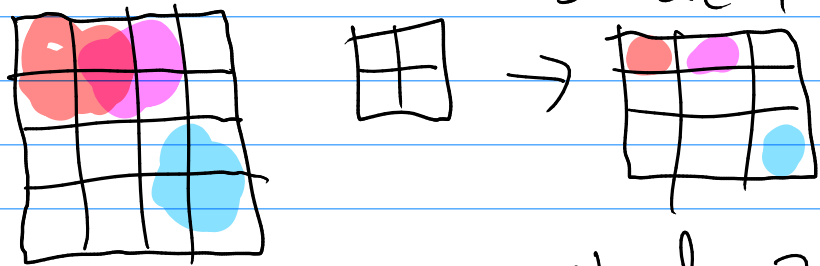
## Conv layer hyperparams

- Filter size

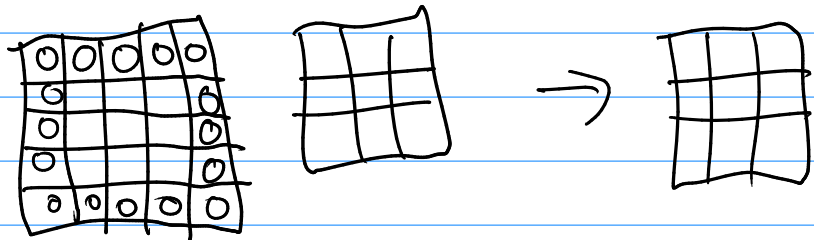
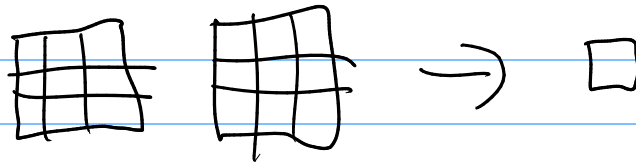


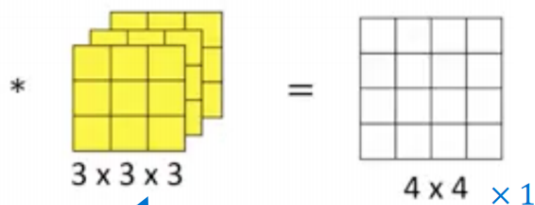
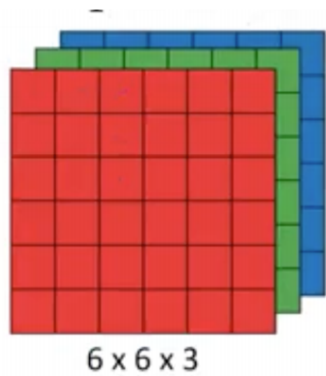
- Output depth = # of filters

- Stride



- Zero padding





1<sup>st</sup> layer convolves with red  
 2<sup>nd</sup> layer convolves with green  
 3<sup>rd</sup> layer convolves with blue

Input Volume (+pad 1) (7x7x3)	Filter W0 (3x3x3)	Filter W1 (3x3x3)	Output Volume (3x3x2)
$x[:, :, 0]$ 0 0 0 0 0 0 0 0 0 2 2 1 0 0 0 2 2 1 2 0 0 0 1 2 0 0 0 0 0 2 1 0 0 2 0 0 2 1 0 2 2 0 0 0 0 0 0 0 0	$w0[:, :, 0]$ 1 -1 0 0 -1 -1 1 0 1 $w0[:, :, 1]$ -1 -1 1 1 0 1 -1 1 1 $w0[:, :, 2]$ -1 -1 1 -1 1 1 -1 1 -1	$w1[:, :, 0]$ -1 0 1 0 -1 0 0 0 0 $w1[:, :, 1]$ -1 0 1 -1 -1 -1 -1 1 1 $w1[:, :, 2]$ 1 0 0 0 0 0 1 0 -1	$o[:, :, 0]$ 6 4 7 5 0 1 -3 3 0 $o[:, :, 1]$ -1 -8 -1 3 -6 -2 -2 -6 -3
$x[:, :, 1]$ 0 0 0 0 0 0 0 0 1 1 2 2 1 0 0 0 1 1 0 1 0 0 1 1 1 1 1 0 0 2 1 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0	Bias $b0$ (1x1x1) $b0[:, :, 0]$ 1	Bias $b1$ (1x1x1) $b1[:, :, 0]$ 0	
$x[:, :, 2]$ 0 0 0 0 0 0 0 0 1 2 0 0 2 0 0 0 0 2 1 0 0 0 2 1 1 2 2 0 0 0 0 0 1 0 0 0 0 0 1 0 2 0 0 0 0 0 0 0 0			

stride = 2

## 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
- $F^2 P, K$  weights
- $K$  biases

Example computation:

Input:  $227 \times 227 \times 3$ .

Use: 96 filters,  $11 \times 11 \times 3$ , 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:  $(\text{input size} + 1)(\text{output size}) \approx 45 \text{ bill.}$

No weight sharing, only local connections

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

# param/filter

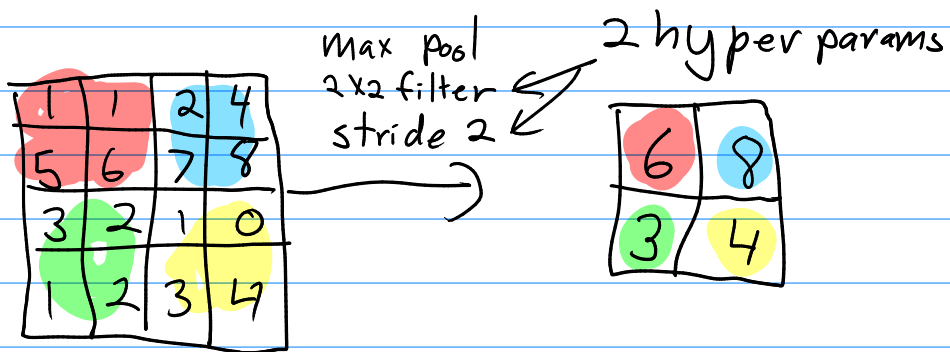
Pooling Layer

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 Hyper params:

- Width/height  $F$
- Stride  $S$

Output  $W_2 \times H_2 \times D_2$

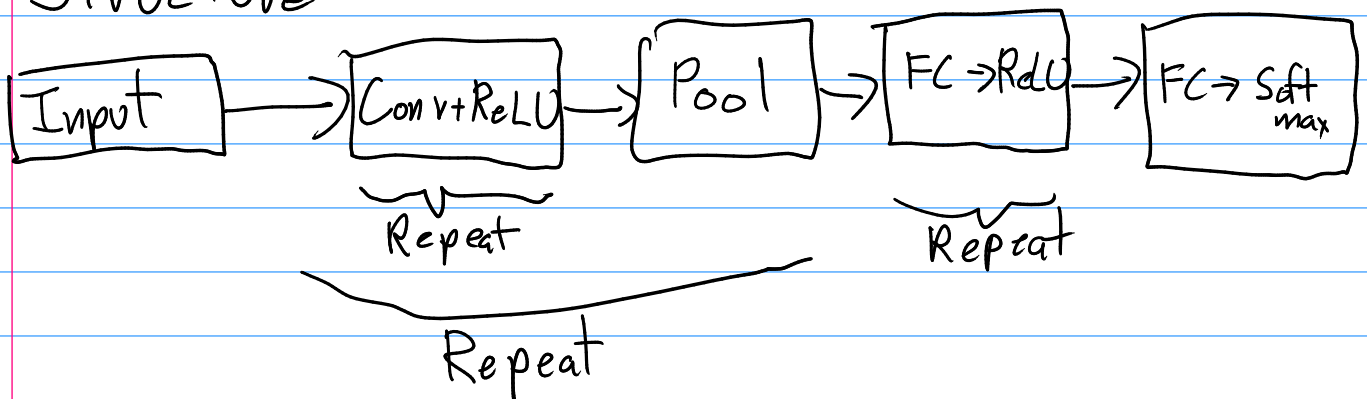
$$W_2 = (W_1 - F) / S + 1$$

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

$$D_2 = D_1$$

No learnable params

## Structure



Datasets: MNIST

60k train, 10k test

10 classes

28x28x1, 99.9%

3	4	2	1	9	5	6	2	1	8
8	9	1	2	5	0	0	6	6	4
6	7	0	1	6	3	6	3	7	0
3	7	7	9	4	6	6	1	8	2
2	9	3	4	3	9	8	7	2	5
1	5	9	8	3	6	5	7	2	3
9	3	1	9	1	5	8	0	8	4
5	6	2	6	8	5	8	8	9	9
3	7	7	0	9	4	8	5	4	3
7	9	6	4	7	0	4	9	2	3

CIFAR-10:

$32 \times 32 \times 3$

50k train, 10k test 10 classes

airplane

automobile

bird

cat

deer

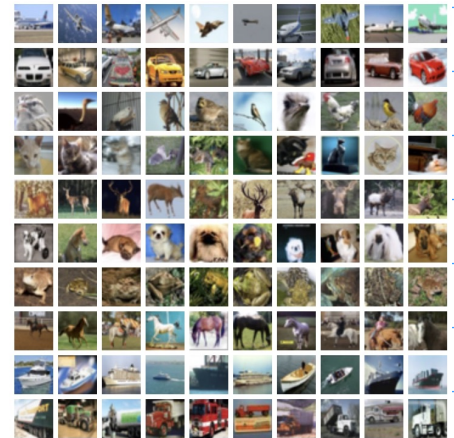
dog

frog

horse

ship

truck



ImageNet: 14m images

$\sim 400 \times 500 \times 3$

10's of thousands of classes

LSVRC dataset  
2012-2017

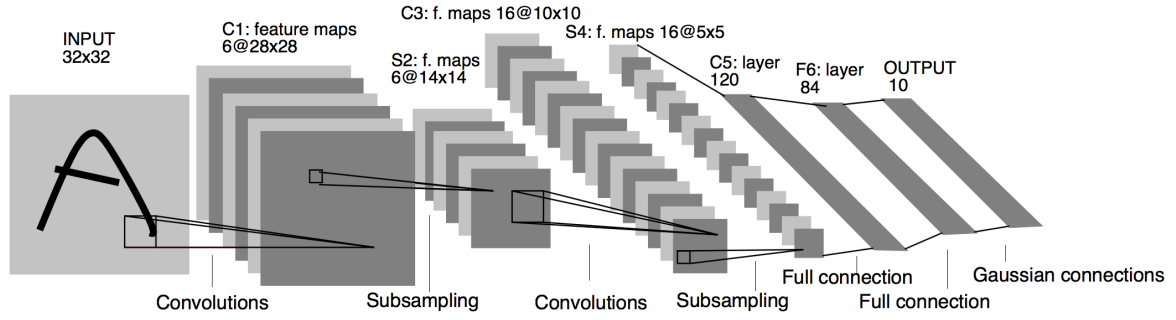
1.2m training  
50k test set  
1000 classes



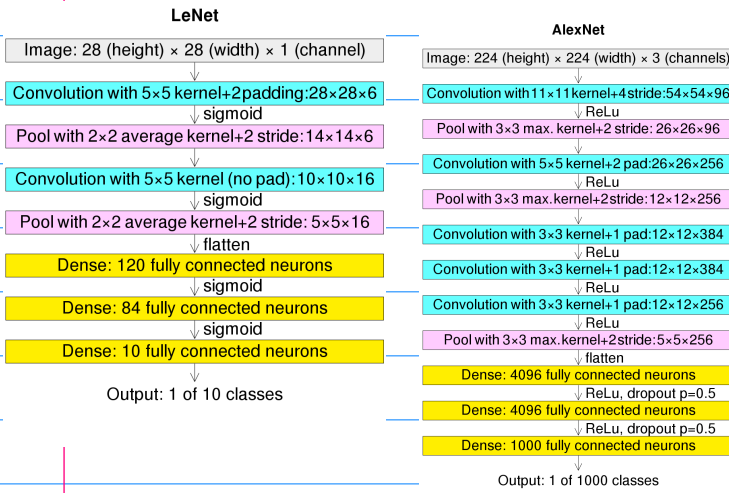


# Architectures

## LeNet (1998)

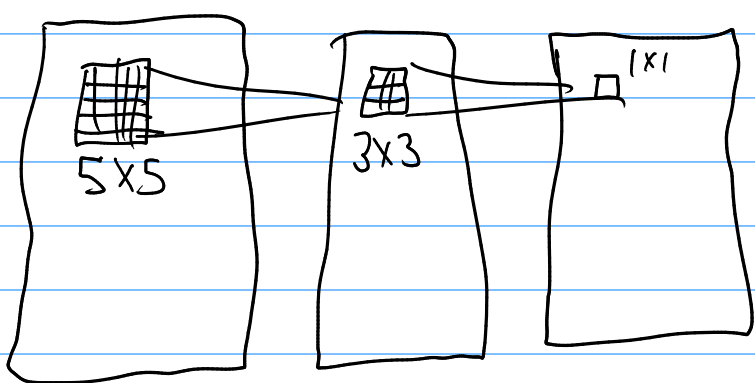
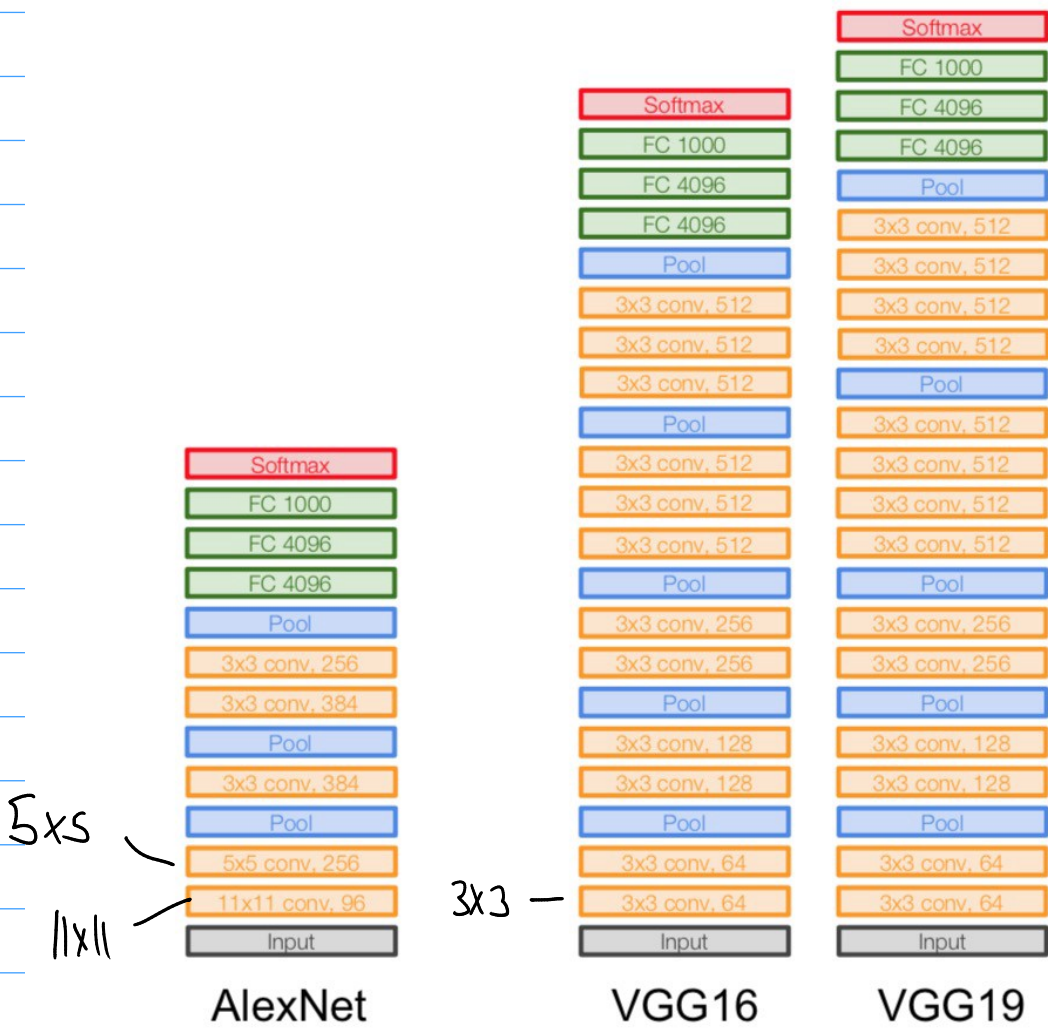


## AlexNet (2012)



Beat competition  
in ILSVRC 12  
Top 5 error of 15%  
vs 25% for  
best all

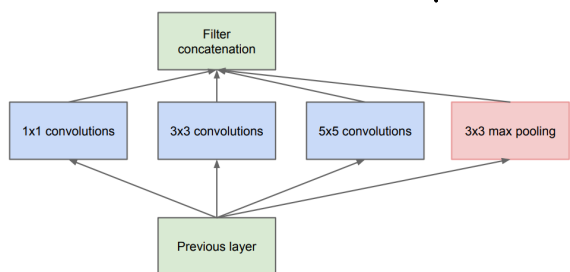
## VGGNet (2014)



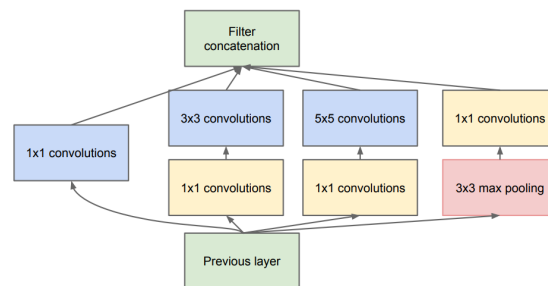
More layers + small filters  
better than few layers + large filters

# GoogLeNet (2014)

## Inception Module



(a) Inception module, naïve version



(b) Inception module with dimension reductions

$W \times H \times D$  input  
 $k$  filters ( $1 \times 1$ )  
 $W \times H \times k$  output

# ResNet (2015)

