Computational Intelligence

Samaneh Hosseini

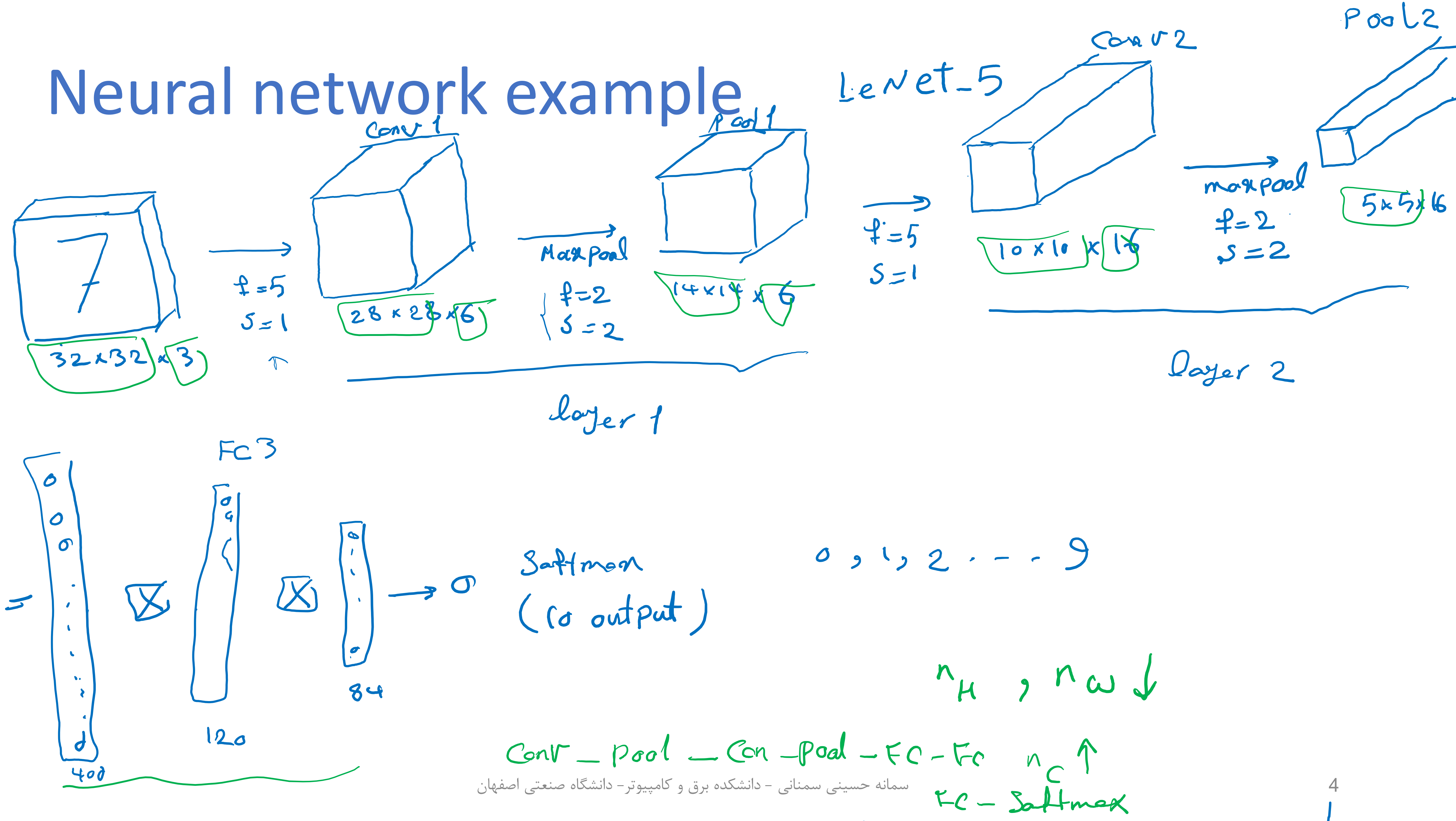
Isfahan University of Technology

Outline

- Convolutional Neural Networks
  - CNN Example
  - Why Convolutions?
- Case Studies
  - Why look at case studies?
  - Classic Networks
    - LeNet-5
    - AlexNet
    - VGG

Convolutional Neural Networks: CNN Example

Neural network example



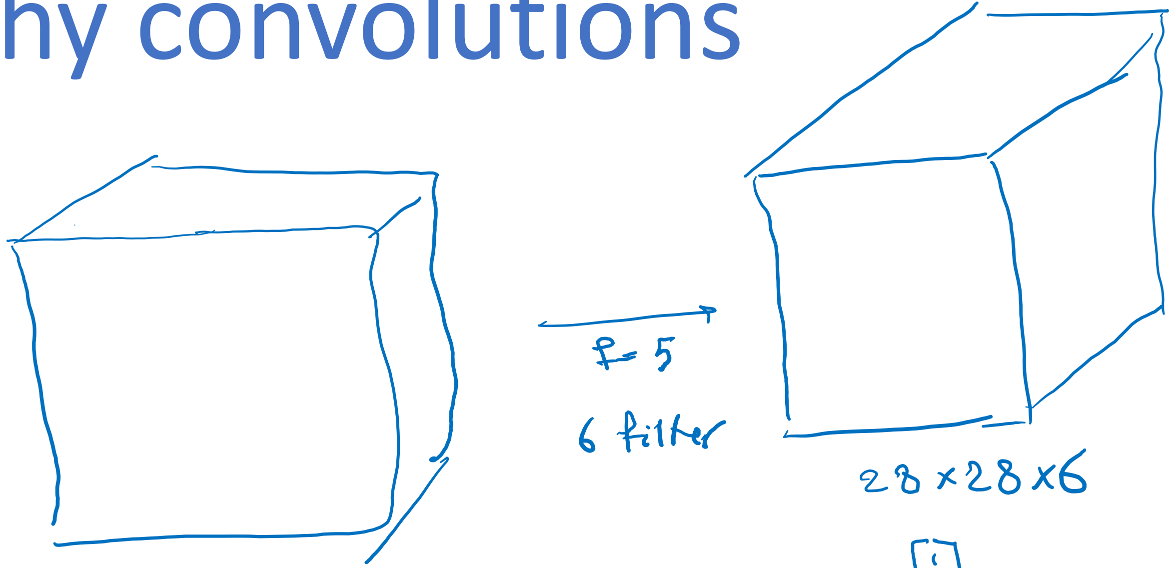
Neural network example

SL.No		Activation Shape	Activation Size	# Parameters
1.	Input Layer:	(32, 32, 3) X	3072 $a^{[0]}$	0
2.	CONV1 (f=5, s=1)	(28, 28, 8)	6272	608
3.	POOL1	(14, 14, 8)	1568	0
4.	CONV2 (f=5, s=1)	(10, 10, 16)	1600	3216
5.	POOL2	(5, 5, 16)	400	0
6.	FC3	(120, 1)	120	48120
7.	FC4	(84, 1)	84	10164
8.	Softmax	(10, 1)	10	850

Convolutional Neural Networks: Why convolutions?

Why convolutions

Conv :



f = 5

6 filter

32x32x3

$a^{[0]}$: 3072

28x28x6

$a^{[1]}$: 4,704

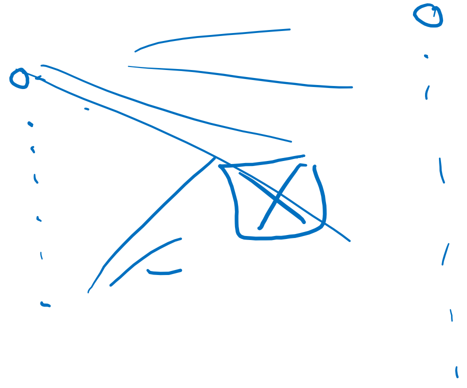
5x5 — 25

1 +

26

6 x 26 = 156 Parameters

FC :



3072

4,704

3072 x 4704 ≃ 14M

Why convolutions

translation invarince

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

6x6

*

1	0	-1
1	0	-1
1	0	-1

3x3

=

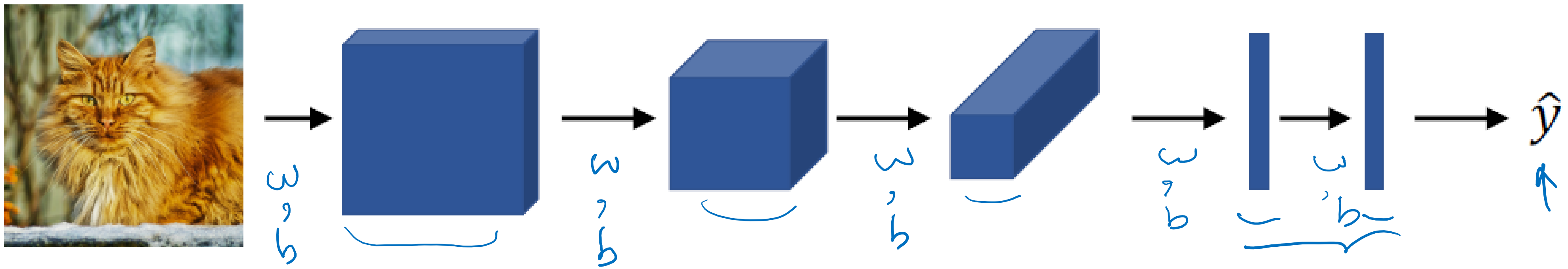
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

- **Parameter sharing**: A feature detector (such as a vertical edge detector) that's useful in one part of the image is probably useful in another part of the image.
- **Sparsity of connections**: In each layer, each output value depends only on a small number of inputs.

Putting it together

Training set $(x^{(1)}, y^{(1)}) \ldots (x^{(m)}, y^{(m)})$.

$\boxed{w, b} \rightarrow$ random



$$\text{Cost } J = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

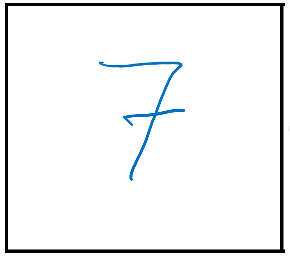
forward pass

backword pass

Use gradient descent to optimize parameters to reduce J

Case Studies:
Classic networks

LeNet - 5



32×32 ×1

gray

60K parameters

Sigmoid/tanh

Relu

after pooling

$n_H \times n_W \times n_C$

FC

FC

softmax

400

$f \times f \times n_C$

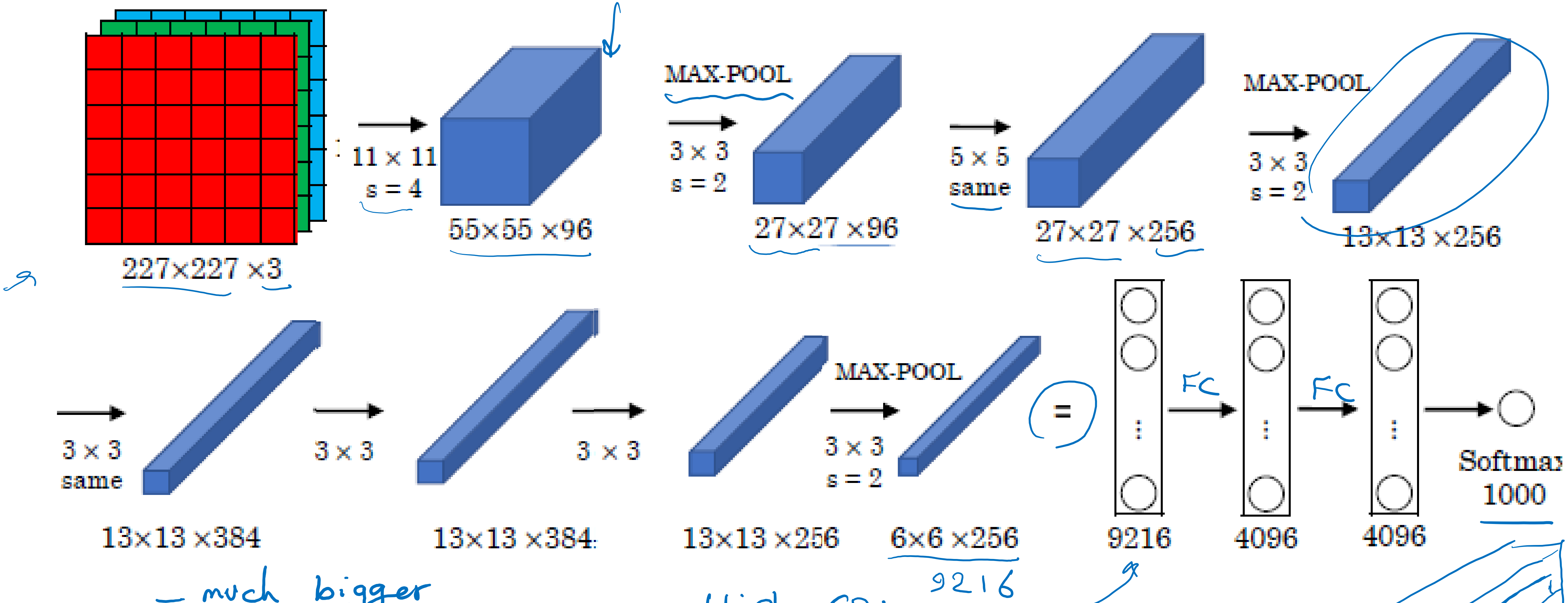
0 1 ... 15

[LeCun et al., 1998. Gradient-based learning applied to document recognition]

AlexNet - 5

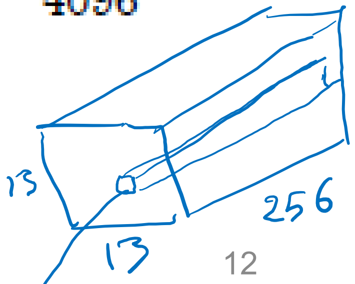
60M paramter

activation function



- much bigger
- Relu

- Multiple GPUs
- Local Response Normalization (LRN)



Normalize

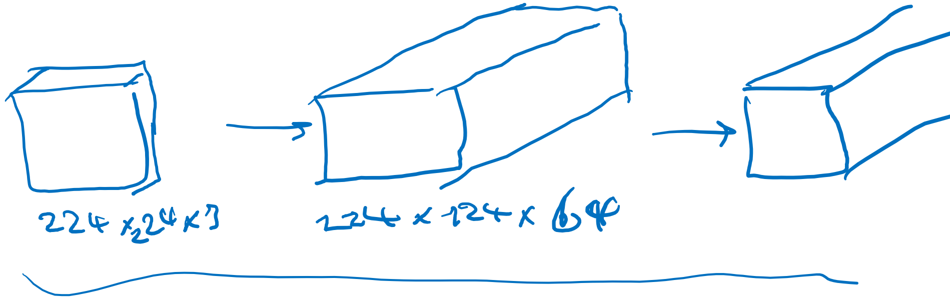
[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

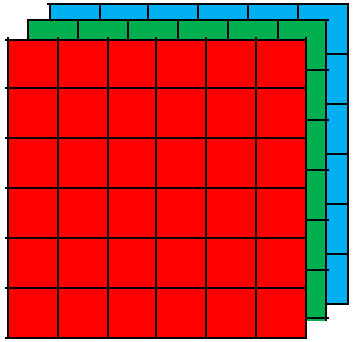
VGG - 16

VGG 19

→ CONV = 3x3 filter, S = 1, Same

→ Max pool = 2x2 S = 2





224×224 ×3

→ [CONV 64] ×2 → 224×224×64 → POOL → 112×112 ×64 → [CONV 128] ×2 → 112×112 ×128 → POOL → 56×56 ×128

→ [CONV 256] ×3 → 56×56 ×256 → POOL → 28×28 ×256 → [CONV 512] ×3 → 28×28 ×512 → POOL → 14×14×512

→ [CONV 512] ×3 → 14×14 ×512 → POOL → 7×7×512 → FC 4096 → FC 4096 → Softmax 1000

1.38 M

n_H, n_W ↓ n_C ↑

[Simonyan & Zisserman 2015. Very deep convolutional networks for large-scale image recognition] → oxford

Outline

- Convolutional Neural Networks
  - CNN Example
  - Why Convolutions?
- Case Studies
  - Why look at case studies?
  - Classic Networks
    - LeNet-5
    - AlexNet
    - VGG