Case Studies

Why look at case studies?

deeplearning.ai

Outline

Classic networks:

- LeNet-5 $\leftarrow$

- AlexNet $\leftarrow$

- VGG $\leftarrow$

ResNet   (152)

Inception

deeplearning.ai

Case Studies

Classic networks

LeNet - 5

$32 \times 32 \times 1$ 5×5, $s = 1$ $28 \times 28 \times 6$ avg pool $f = 2$, $s = 2$ $14 \times 14 \times 6$ 5×5, $s = 1$ $10 \times 10 \times 16$ avg pool $f = 2$, $s = 2$ $5 \times 5 \times 16$ FC   120   FC   84 $\hat{y}$ Softmax 10

400

60K parameters.

$n_H, n_w \downarrow$ $n_c \uparrow$
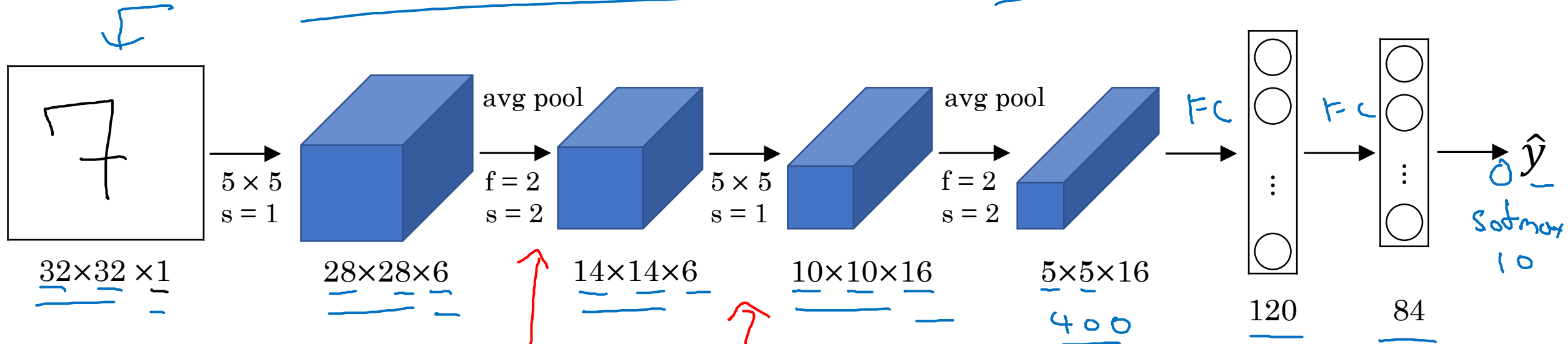
nonlinearity ofter pooling $n_H \times n_w \times n_c$ $f \times f \times n_c$

conv pool conv pool fc fc output

Advanced: Sigmoid/tanh   ReLU

$\boxed{\text{II}}$ $\underline{\text{III}}$.

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

Andrew Ng

AlexNet

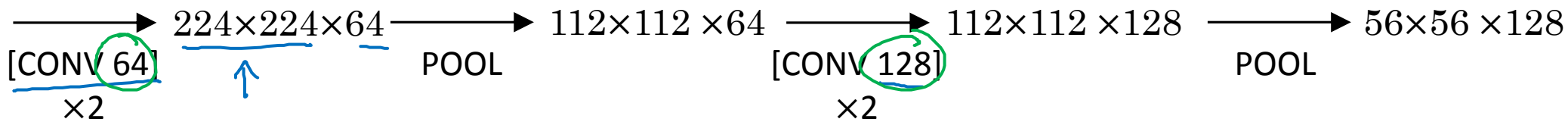$227 \times 227 \times 3$ → 11×11 $s = 4$ → $55 \times 55 \times 96$ → MAX-POOL 3×3 $s = 2$ → $27 \times 27 \times 96$ → 5×5 same → $27 \times 27 \times 256$ → MAX-POOL 3×3 $s = 2$ → $13 \times 13 \times 256$

3×3 same → $13 \times 13 \times 384$ → 3×3 → $13 \times 13 \times 384$ → 3×3 → $13 \times 13 \times 256$ → MAX-POOL 3×3 $s = 2$ → $6 \times 6 \times 256$

9216

= 9216 → FC → 4096 → FC → 4096 → Softmax 1000

- Similar to LeNet, but much bigger.
- ReLU
- Multiple GPUs.
- Local Response Normalization (LRN)

~60M parameters

13 13 256

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

Andrew Ng

VGG - 16
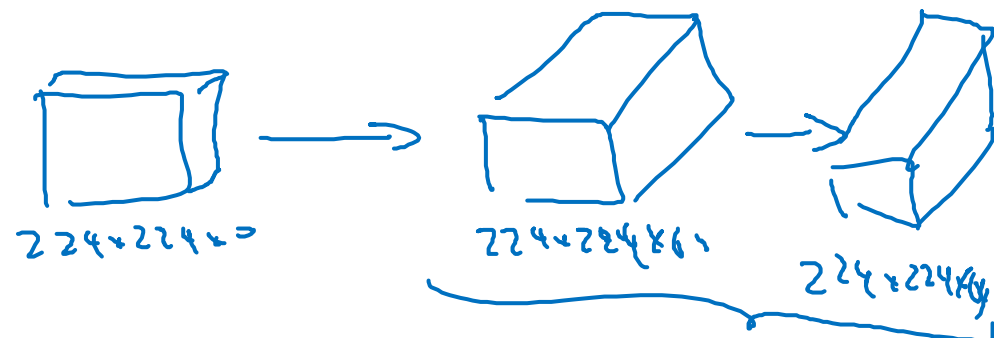
CONV = 3×3 filter, s = 1, same   MAX-POOL = 2×2 , s = 2

VGG-19

$224 \times 224 \times 3$

$224 \times 224 \times 3$ → [CONV 64] ×2 → $224 \times 224 \times 64$ → POOL → $112 \times 112 \times 64$ → [CONV 128] ×2 → $112 \times 112 \times 128$ → POOL → $56 \times 56 \times 128$

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

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

$n_H, n_W \downarrow$ $n_c \uparrow$ ~138M

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

Andrew Ng
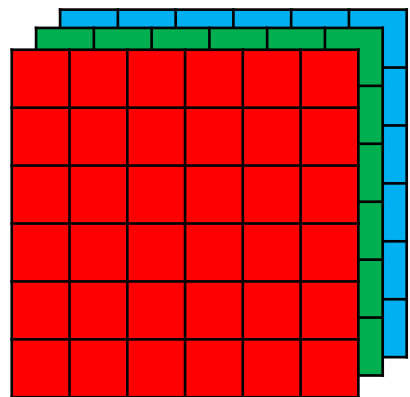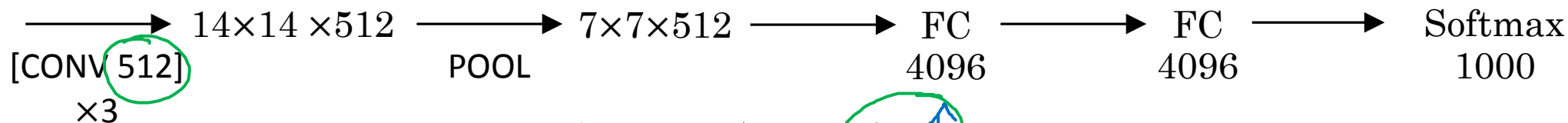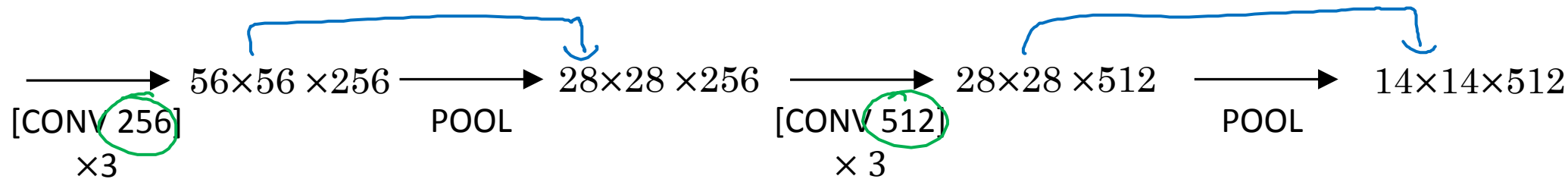
Case Studies

Residual Networks
(ResNets)

deeplearning.ai
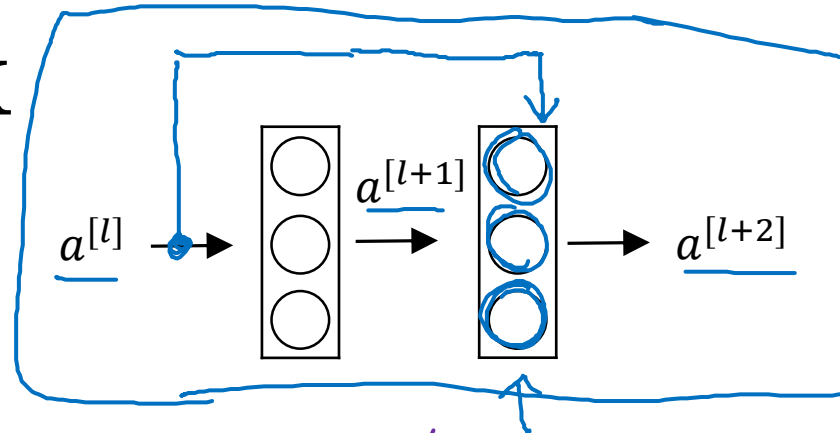
Residual block

"short cut" / skip connection

$a^{[l]}$ → Linear → → ReLU $\xrightarrow{a^{[l+1]}}$ → Linear → ⊕ → ReLU → $a^{[l+2]}$

"main pain"
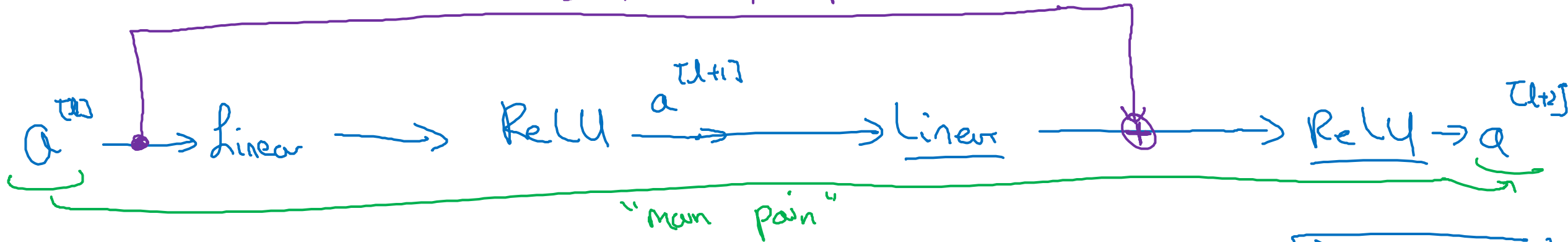
$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]}$ $a^{[l+1]} = g(z^{[l+1]})$ $z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]}$ $a^{[l+2]} = g(z^{[l+2]})$

$$a^{[l+2]} = g\left(z^{[l+2]} + a^{[l]}\right)$$

[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng

Residual Network

$$x \rightarrow \cdots \rightarrow a^{[l]}$$

Plain

ResNet

"reality"

theory

training error

layers

training error

layers

[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng
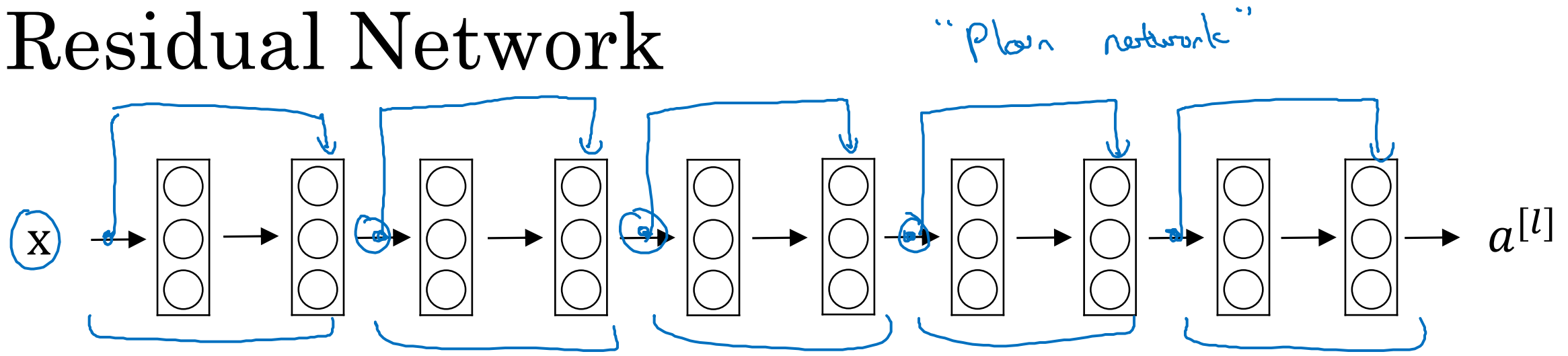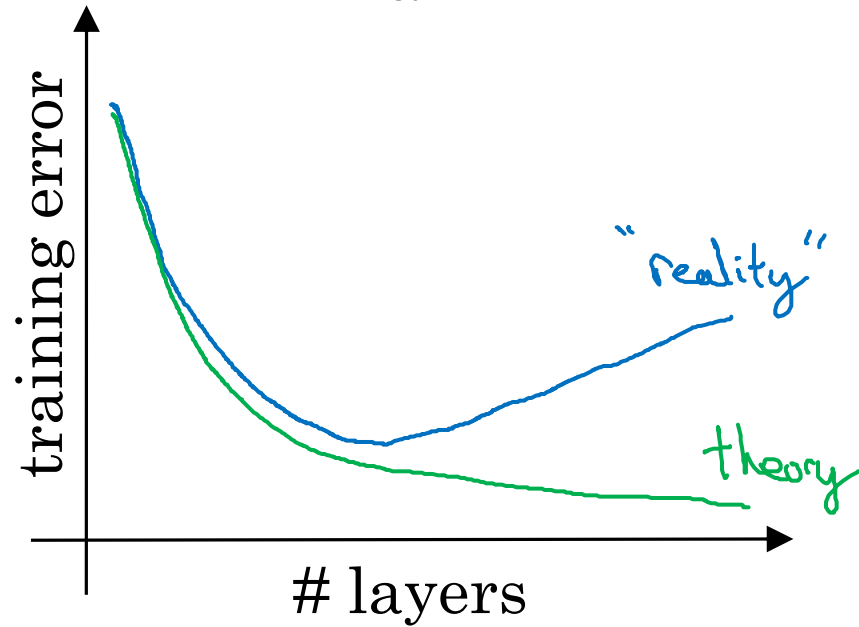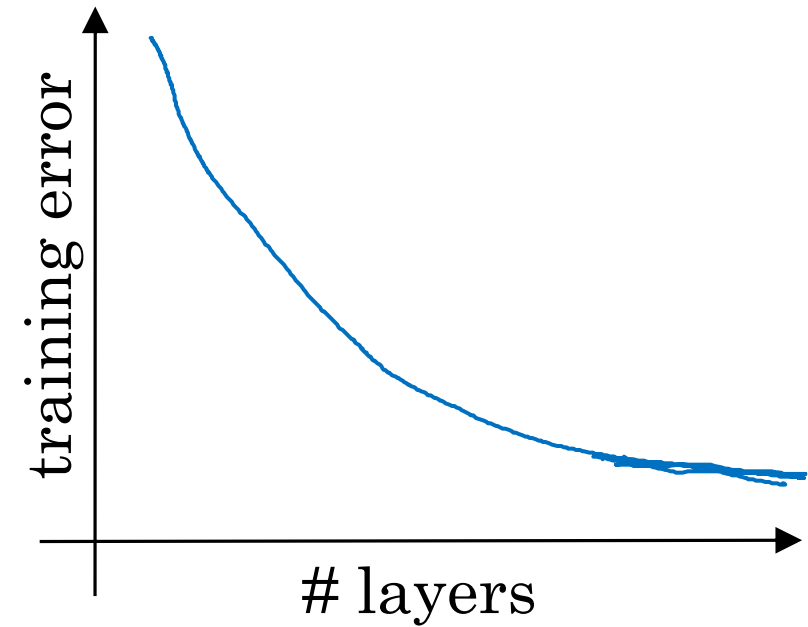
Case Studies

Network in Network
and 1×1 convolutions

deeplearning.ai

Why does a 1 × 1 convolution do?

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

6×6 × 1

* $\boxed{2}$ =

2	4	6	...		

$6 \times 6 \times 32$ * $1 \times 1 \times 32$ =

$32 \longrightarrow$ # filters. $n_c^{[l+1]}$

ReLU

Relu

Network in Network

$6 \times 6 \times$ # filters

[Lin et al., 2013. Network in network]

Andrew Ng

Using 1×1 convolutions

ReLU

CONV 1×1

~~32~~

192

$28 \times 28 \times 192$

$28 \times 28 \times$ ~~32~~ 192

$1 \times 1 \times 192$

32 filters.

n_H, n_W, n_c

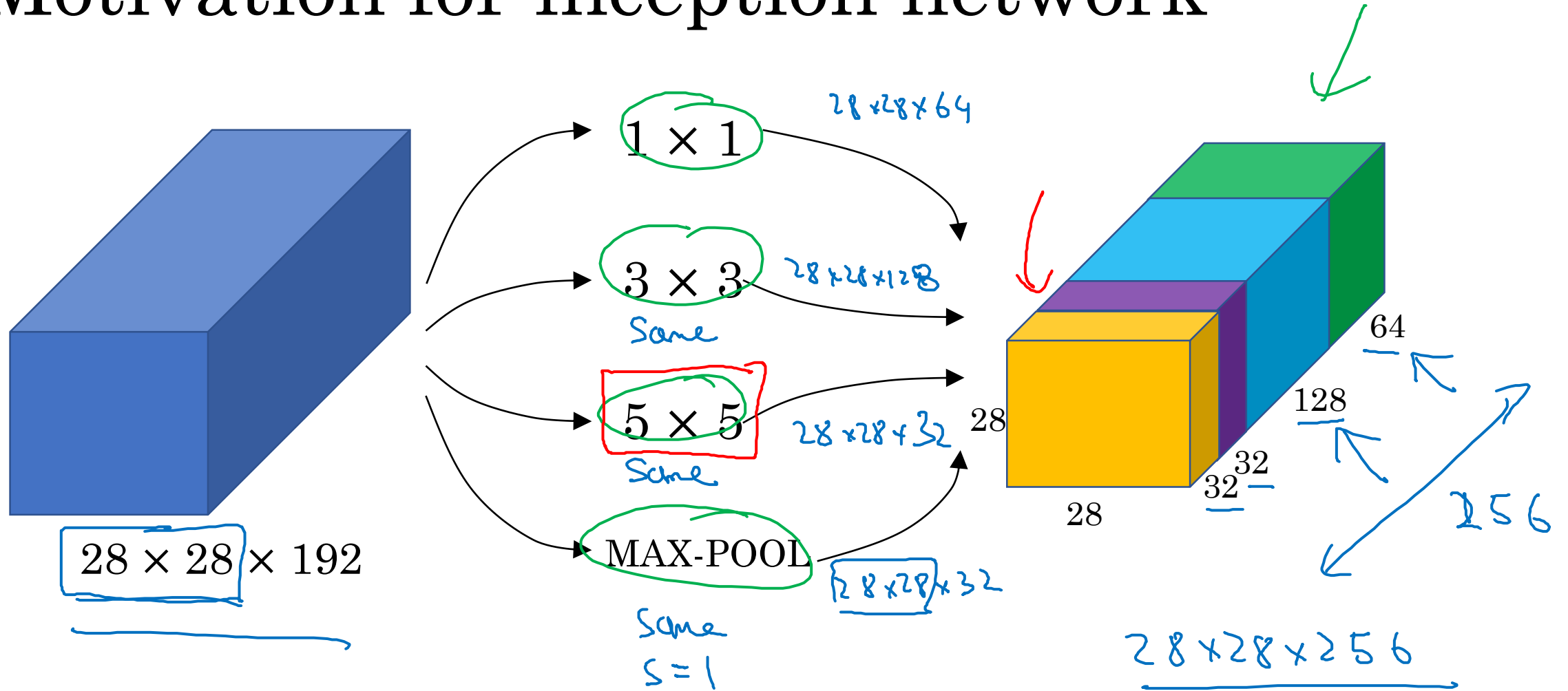[Lin et al., 2013. Network in network]

Andrew Ng
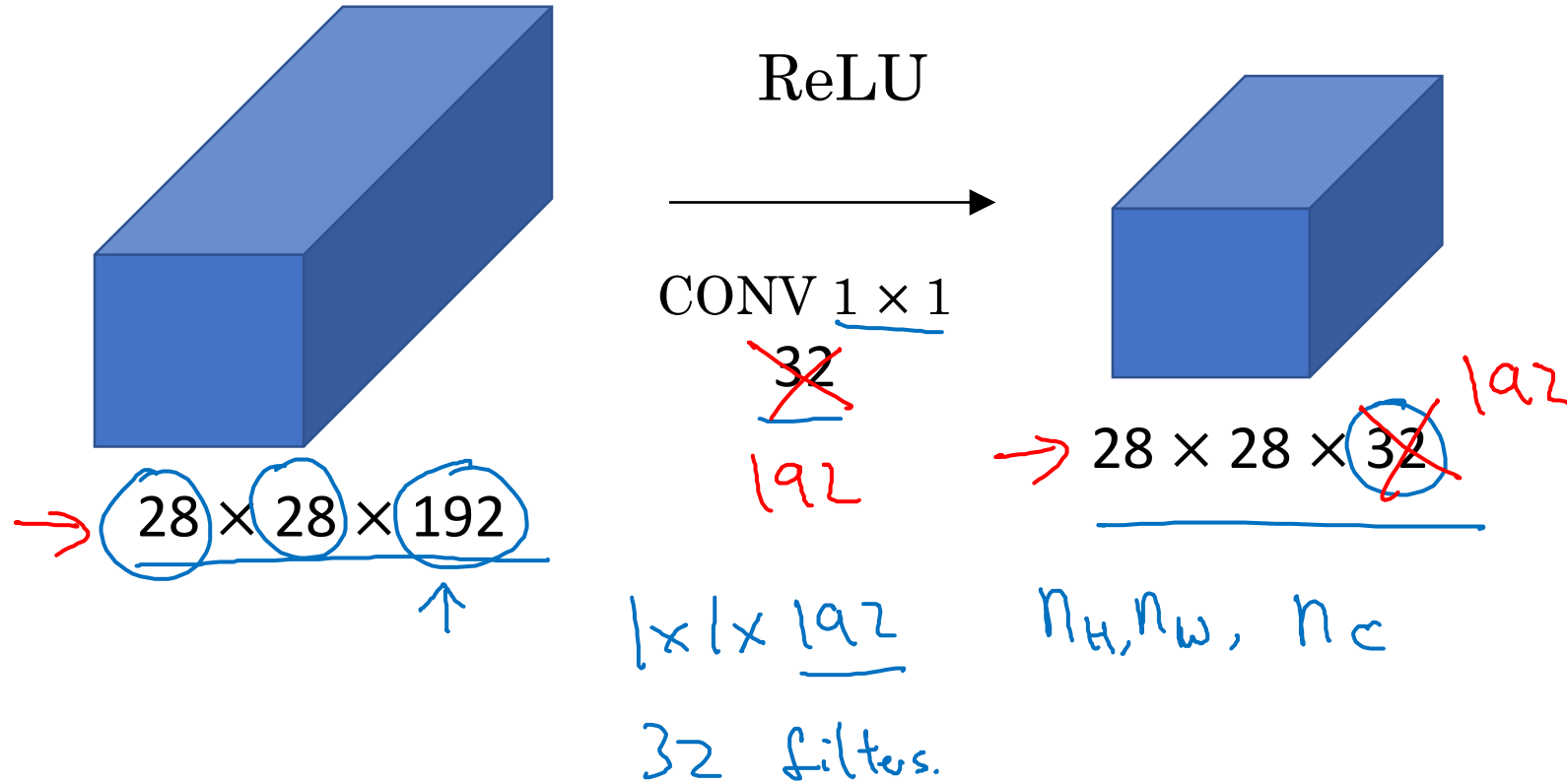
deeplearning.ai

Case Studies

Inception network motivation

Motivation for inception network

$28 \times 28 \times 192$

1×1 $28 \times 28 \times 64$

3×3 $28 \times 28 \times 128$
Same

5×5 $28 \times 28 \times 32$
Same

MAX-POOL $28 \times 28 \times 32$
Same
$S = 1$

28
28
32 32 128 64
256

$28 \times 28 \times 256$

[Szegedy et al. 2014. Going deeper with convolutions]

Andrew Ng

The problem of computational cost

CONV
5 × 5,
same,
32

28 × 28 × 192

28 × 28 × 32

32 filters.   filters are $5 \times 5 \times 192$.

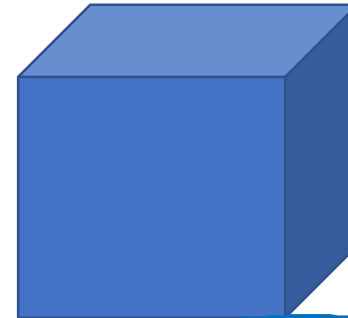$28 \times 28 \times 32 \times 5 \times 5 \times 192 = 120M$.

Andrew Ng
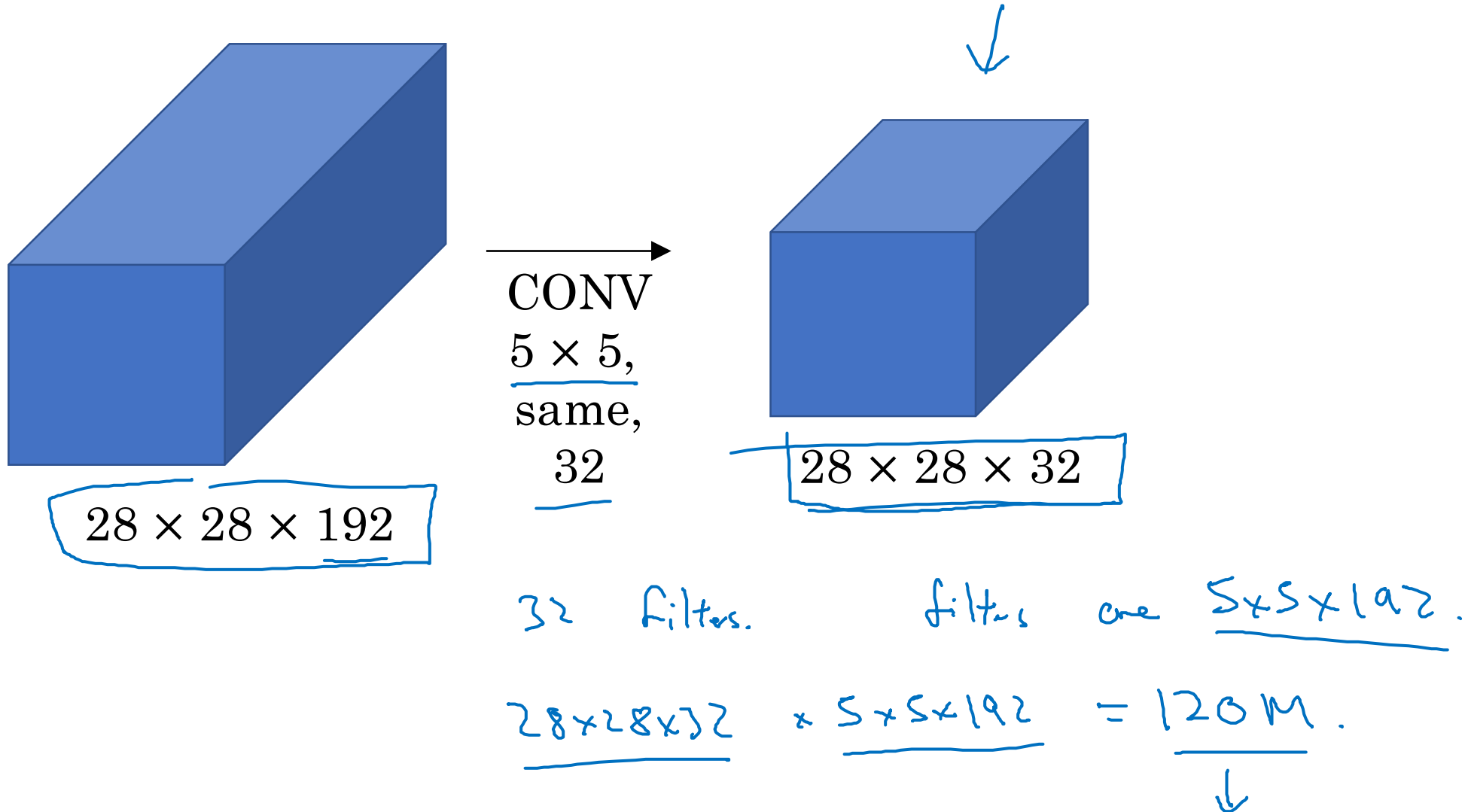
Using 1×1 convolution

"bottleneck layer"
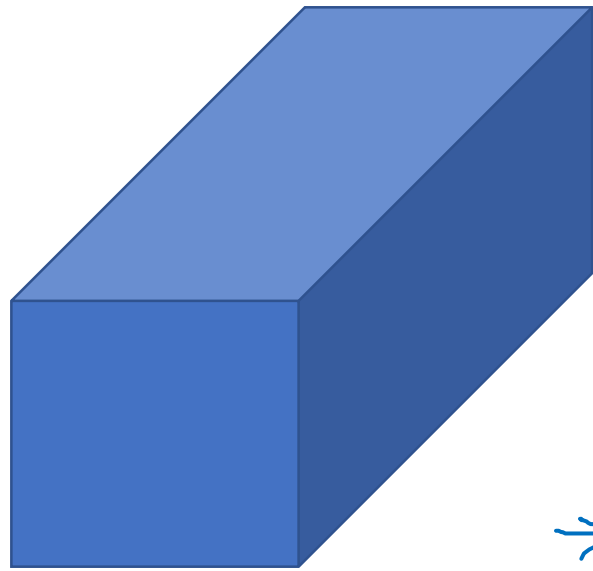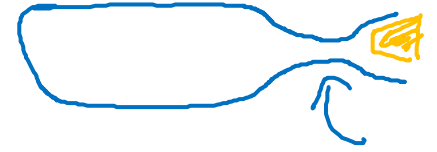
CONV
1×1,
$\rightarrow 16,$
$\rightarrow 1 \times 1 \times 192$

$28 \times 28 \times 192$

$28 \times 28 \times 16$

CONV
$5 \times 5,$
$32,$
$5 \times 5 \times 16$
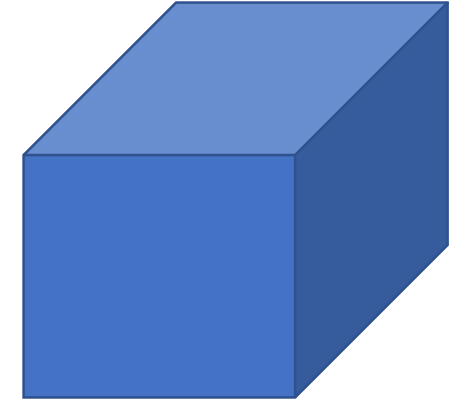
$28 \times 28 \times 32$

$28 \times 28 \times 16 \times 192 = 2.4M$

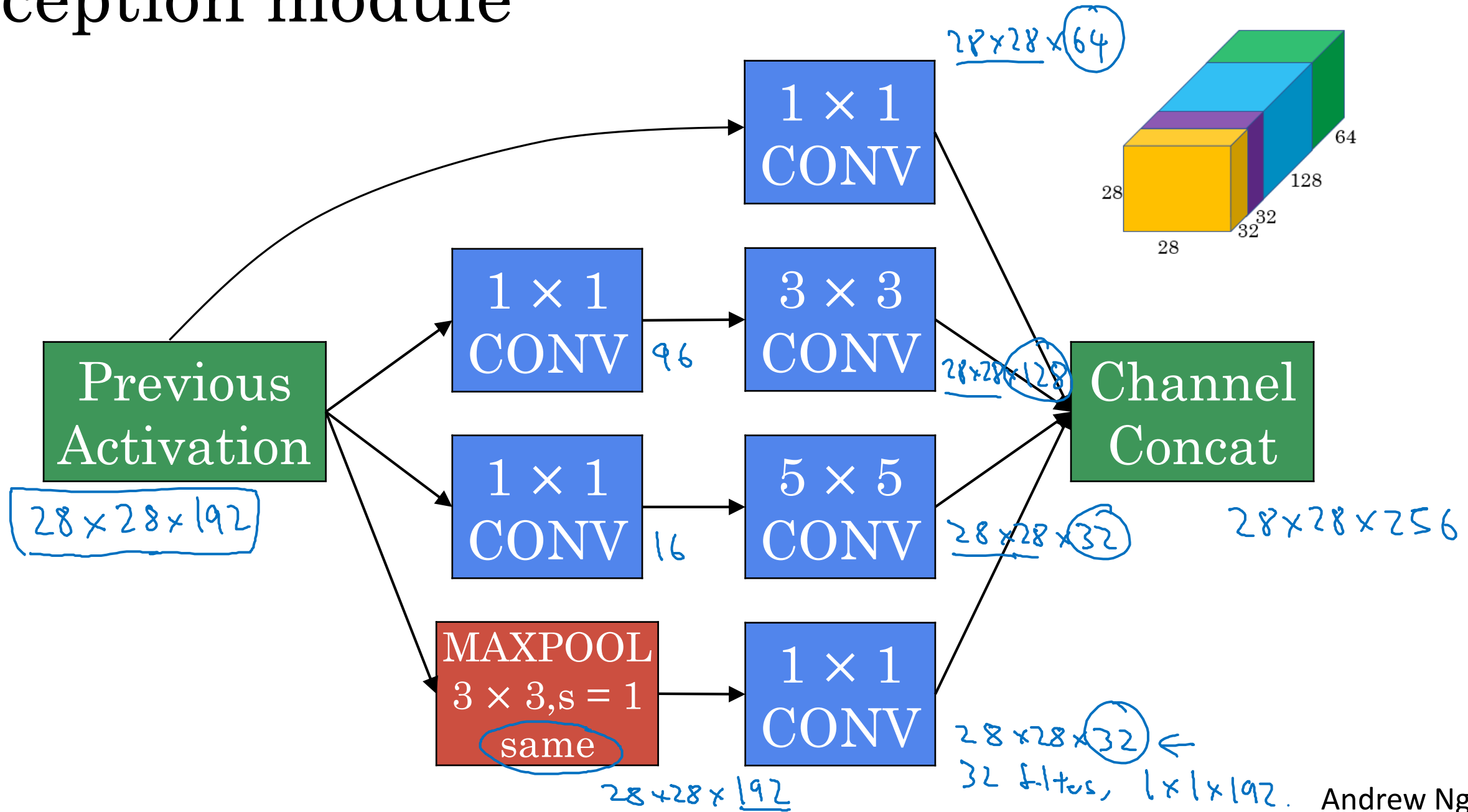$28 \times 28 \times 32 \times 5 \times 5 \times 16 = 10.0M$

$12.4M$

$120M \rightarrow$

Andrew Ng

Case Studies

Inception network

deeplearning.ai

Inception module

Previous Activation
$28 \times 28 \times 192$

1×1 CONV — $28 \times 28 \times 64$

1×1 CONV — 96 → 3×3 CONV — $28 \times 28 \times 128$

1×1 CONV — 16 → 5×5 CONV — $28 \times 28 \times 32$

MAXPOOL 3×3, s = 1, same → 1×1 CONV — $28 \times 28 \times 32$

$28 \times 28 \times 192$

32 filters, $1 \times 1 \times 192$.

Channel Concat — $28 \times 28 \times 256$

Andrew Ng

Inception network

GooLeNet

Softmax

[Szegedy et al., 2014, Going Deeper with Convolutions]

Andrew Ng

Andrew Ng