deeplearning.ai

Case Studies

Why look at case studies?

Outline

Classic networks:

- LeNet-5 ←
- AlexNet ←
- VGG ←

ResNet (152)

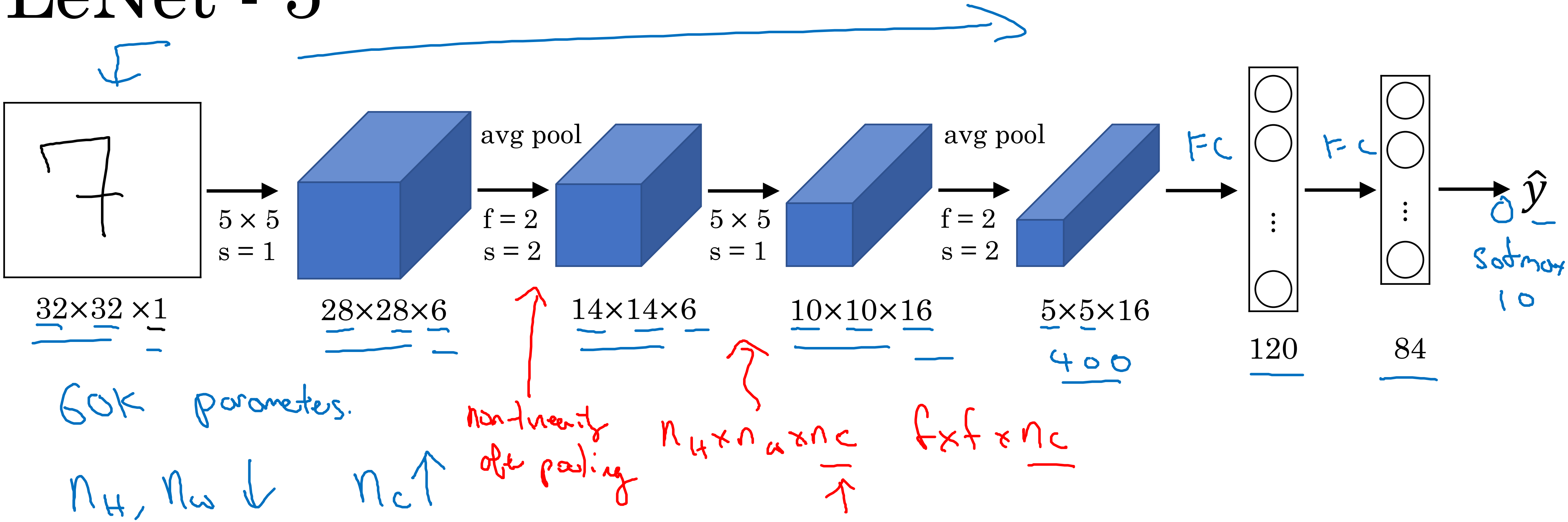
Inception

deeplearning.ai

Case Studies

Classic networks

LeNet - 5



60K parameters

$n_H, n_W \downarrow$ $\quad n_C \uparrow$

conv pool conv pool fc fc output

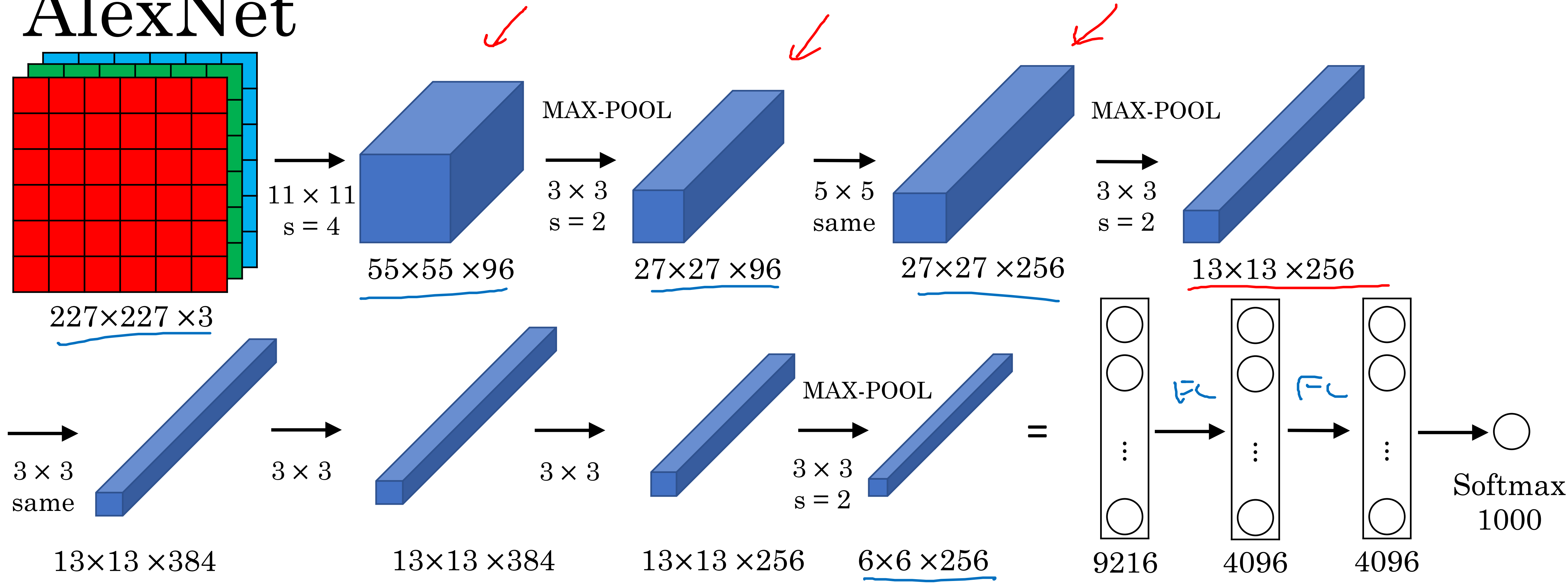
Advanced: sigmoid/tanh ReLU

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

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

Andrew Ng

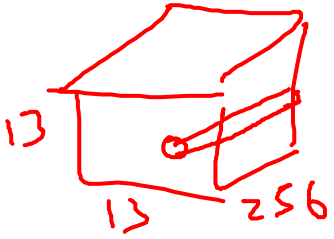
AlexNet



9216

~60M parameters

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



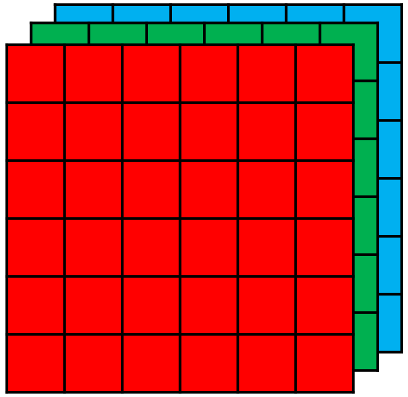
[Krizhevsky et al., 2012. ImageNet classification with deep convolutional neural networks]

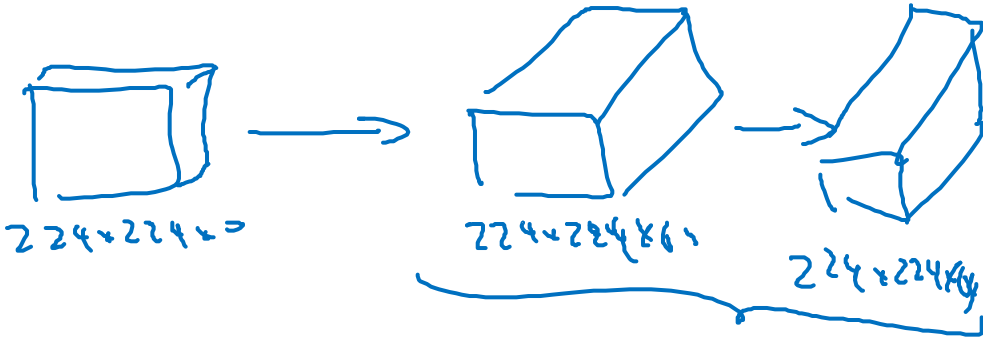
Andrew Ng

VGG - 16

VGG-19

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

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

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

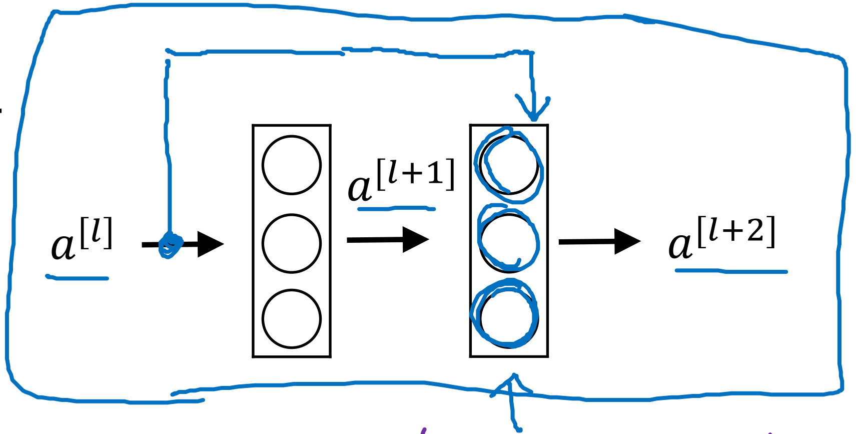
Andrew Ng

deeplearning.ai

Case Studies

Residual Networks (ResNets)

Residual block



"short cut" / skip connection

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

"main path"

$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]}$ $\quad a^{[l+1]} = g(z^{[l+1]})$ $\quad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]}$ $\quad \boxed{\cancel{a^{[l+2]} = g(z^{[l+2]})}}$

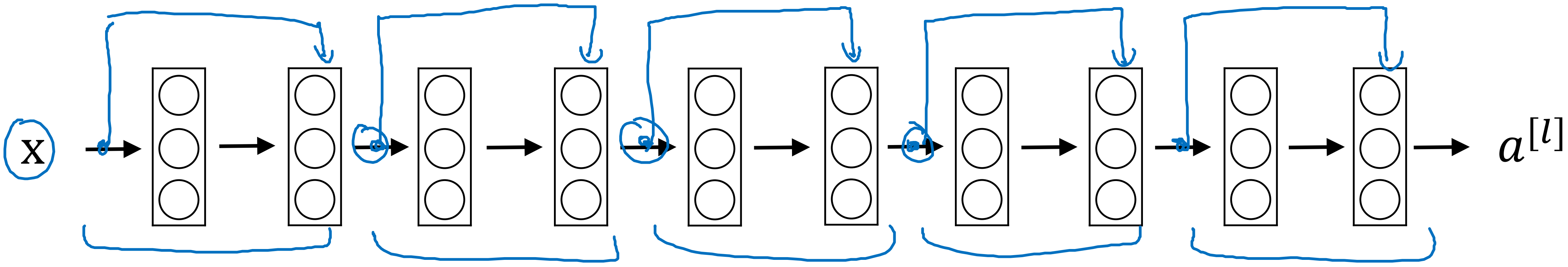
$$a^{[l+2]} = g\left(z^{[l+2]} + a^{[l]}\right)$$

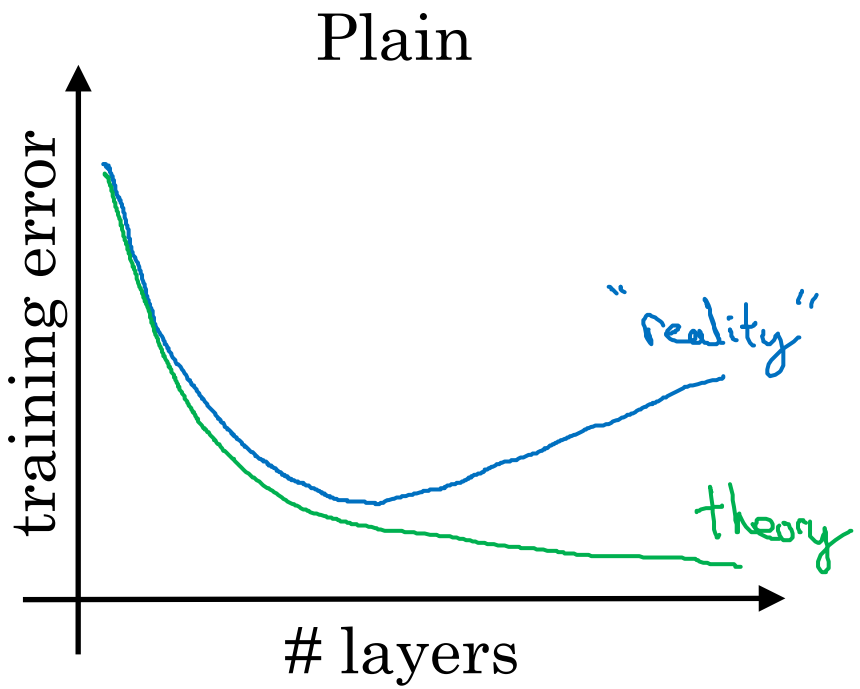
[He et al., 2015. Deep residual networks for image recognition]

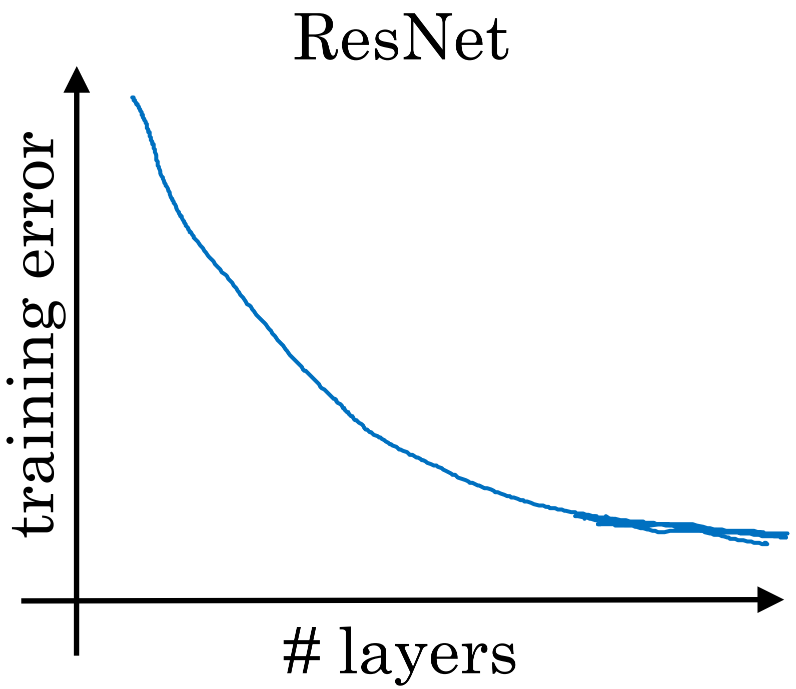
Andrew Ng

Residual Network

"Plain network"







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

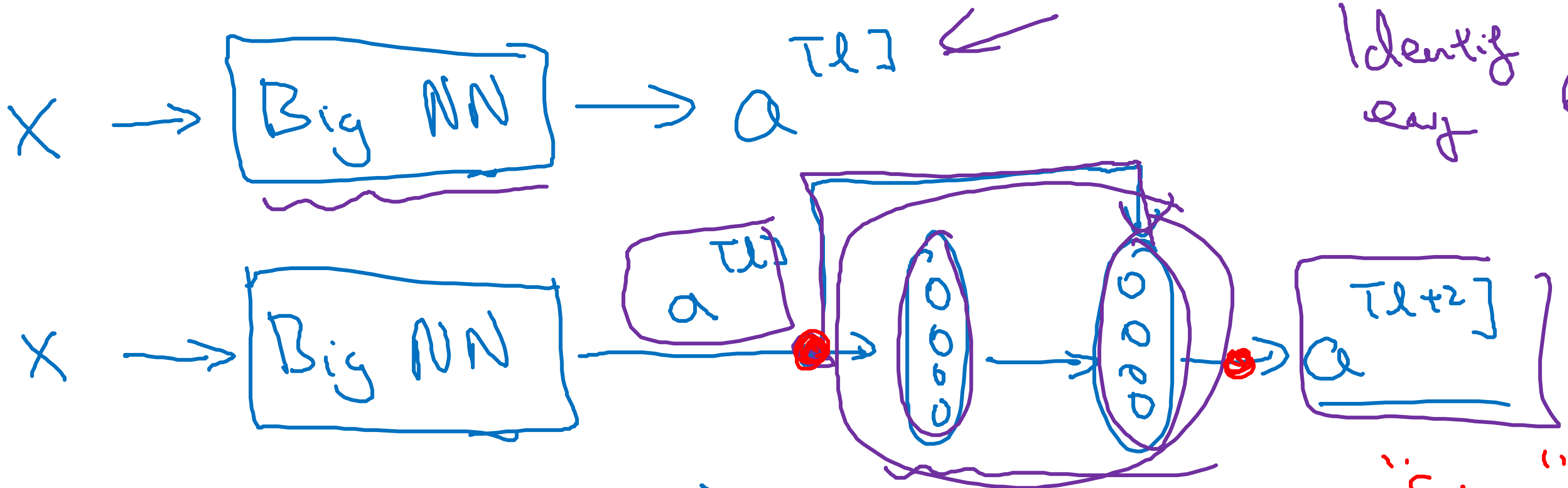
Andrew Ng

deeplearning.ai

Case Studies

Why ResNets work

Why do residual networks work?



Identity function is easy for Residual block to learn!

ReLU. $a \geq 0$

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

$$= g(\cancel{W^{[l+2]}} a^{[l+1]} + \cancel{b^{[l+2]}} + W_s a^{[l]}) = g(a^{[l]})$$

$$= a^{[l]}$$

If $W^{[l+2]} = 0$, $b^{[l+2]} = 0$

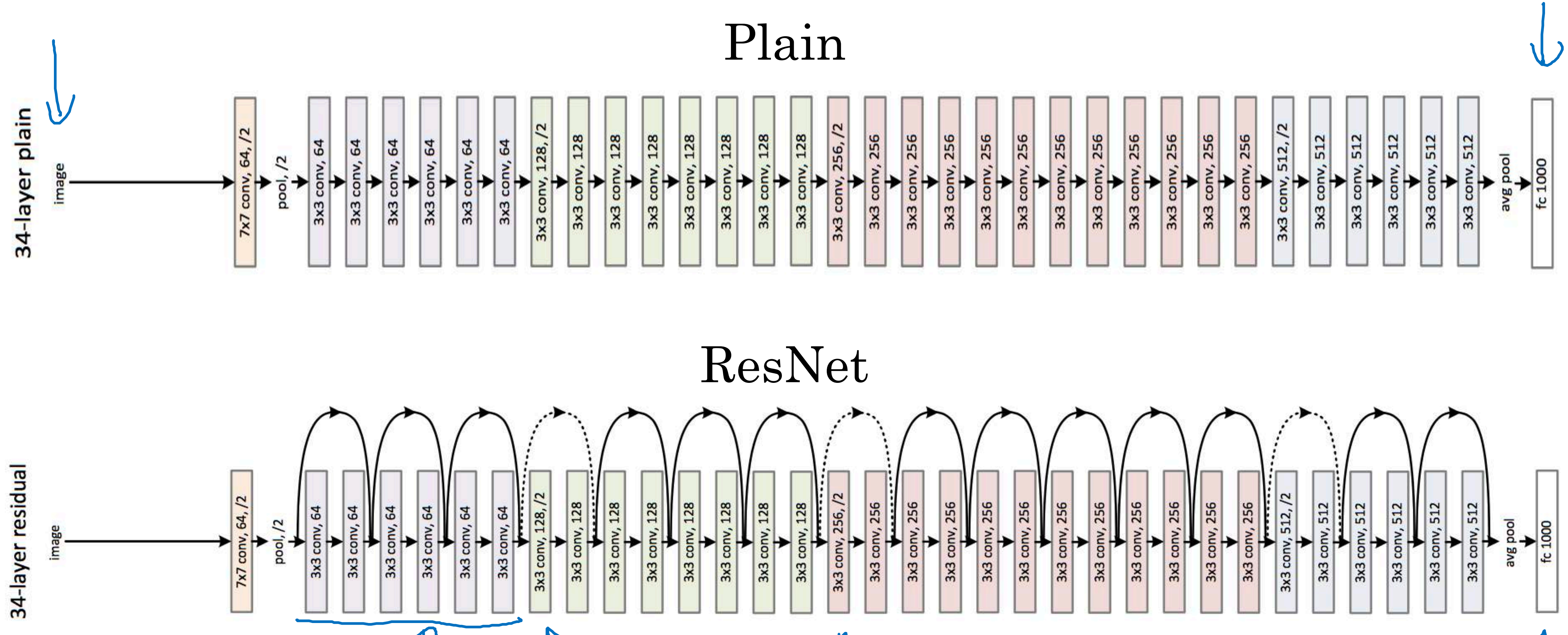
256

128

$\mathbb{R}^{256 \times 128}$

Andrew Ng

ResNet



$z^{[l+2]} + a^{[l]}$

3x3 same

Pool

Pool

W_s

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

Andrew Ng

deeplearning.ai

Case Studies

Network in Network and 1×1 convolutions

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

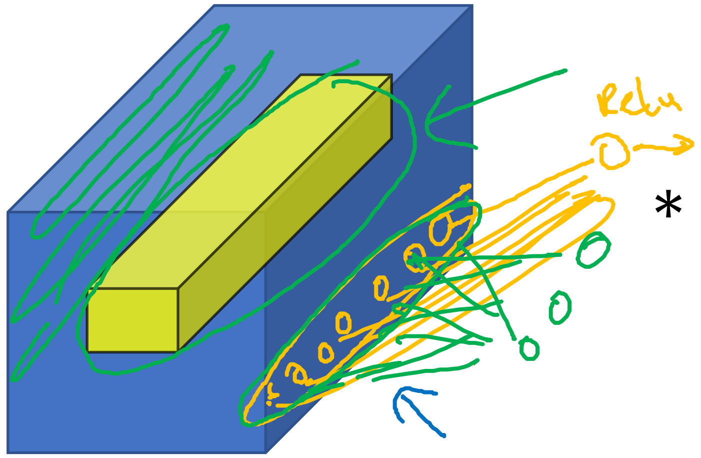
6×6 $\times 1$

$*$ $\boxed{2}$ $=$

2	4	6	...		

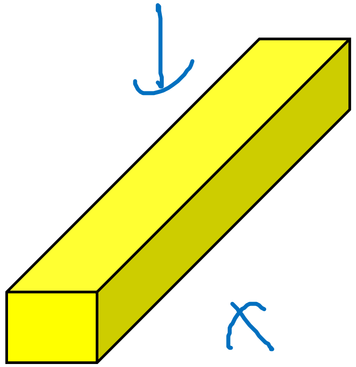
32 → # filters.

$n_c^{[l+1]}$



$6 \times 6 \times 32$

$*$

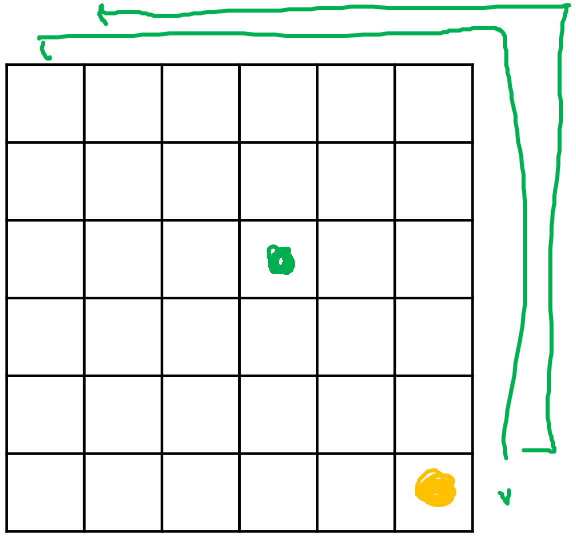


$1 \times 1 \times 32$

$=$

ReLU

Network in Network

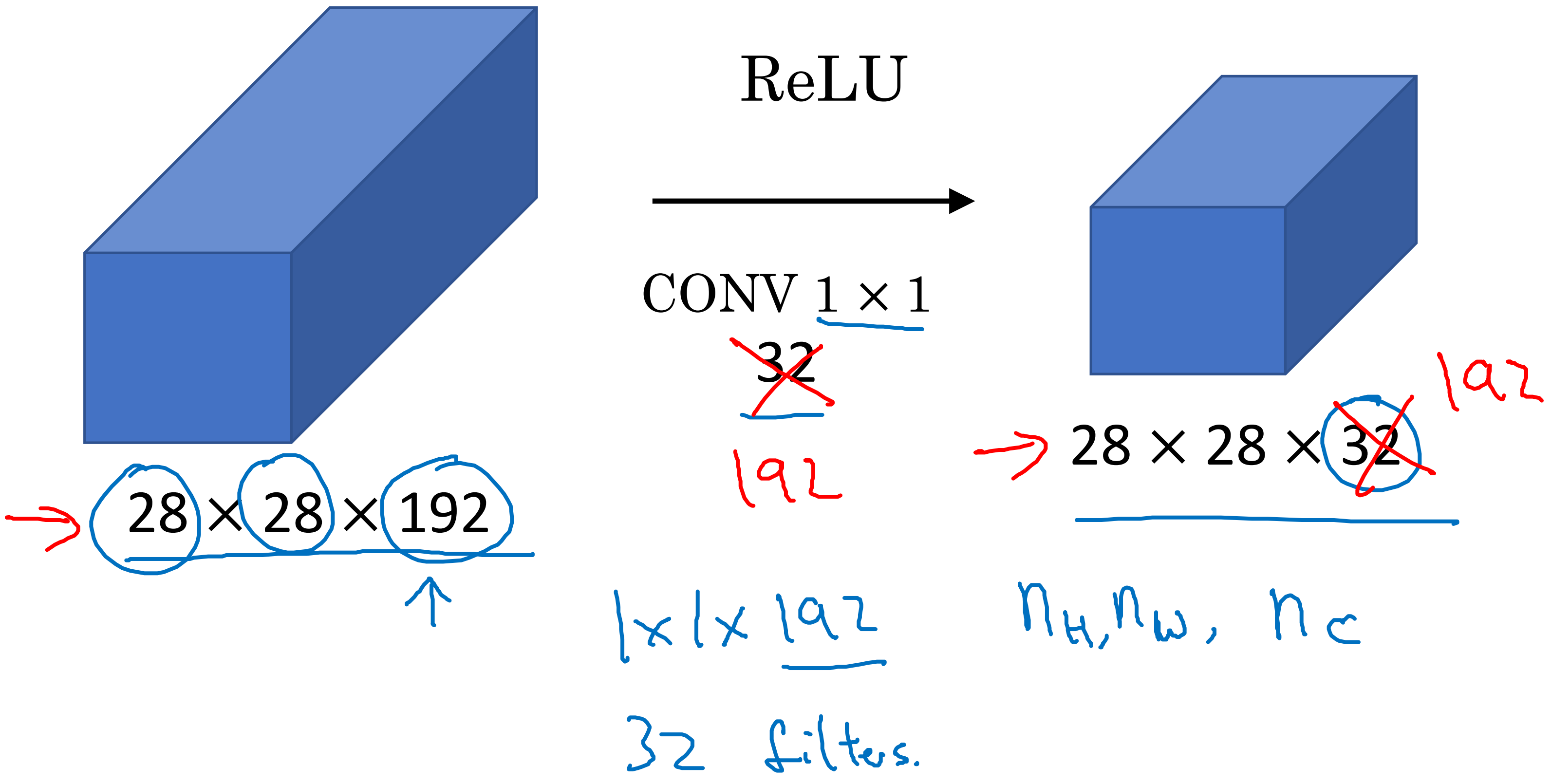


$6 \times 6 \times$ # filters

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

Andrew Ng

Using 1×1 convolutions



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

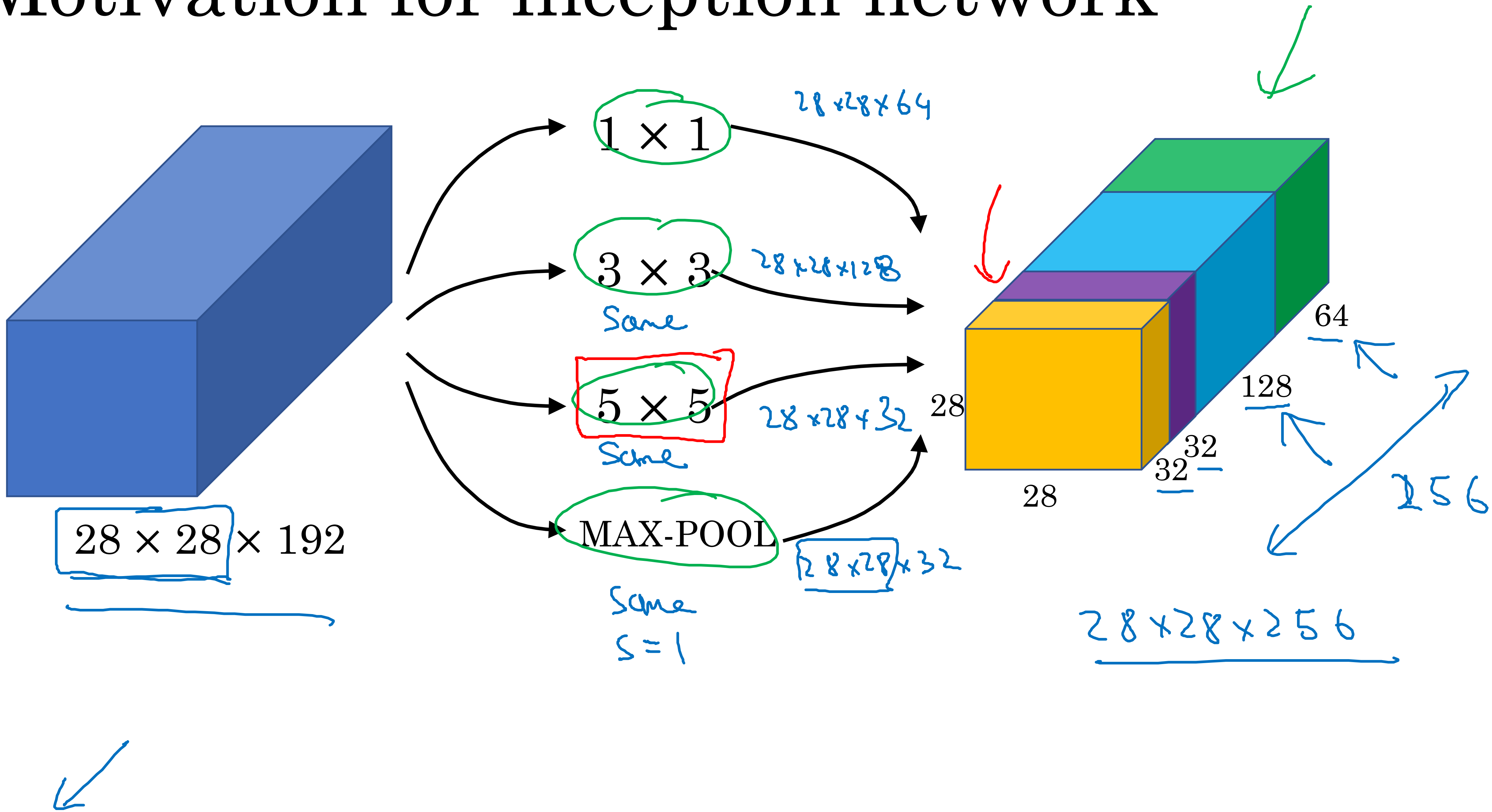
Andrew Ng

deeplearning.ai

Case Studies

Inception network motivation

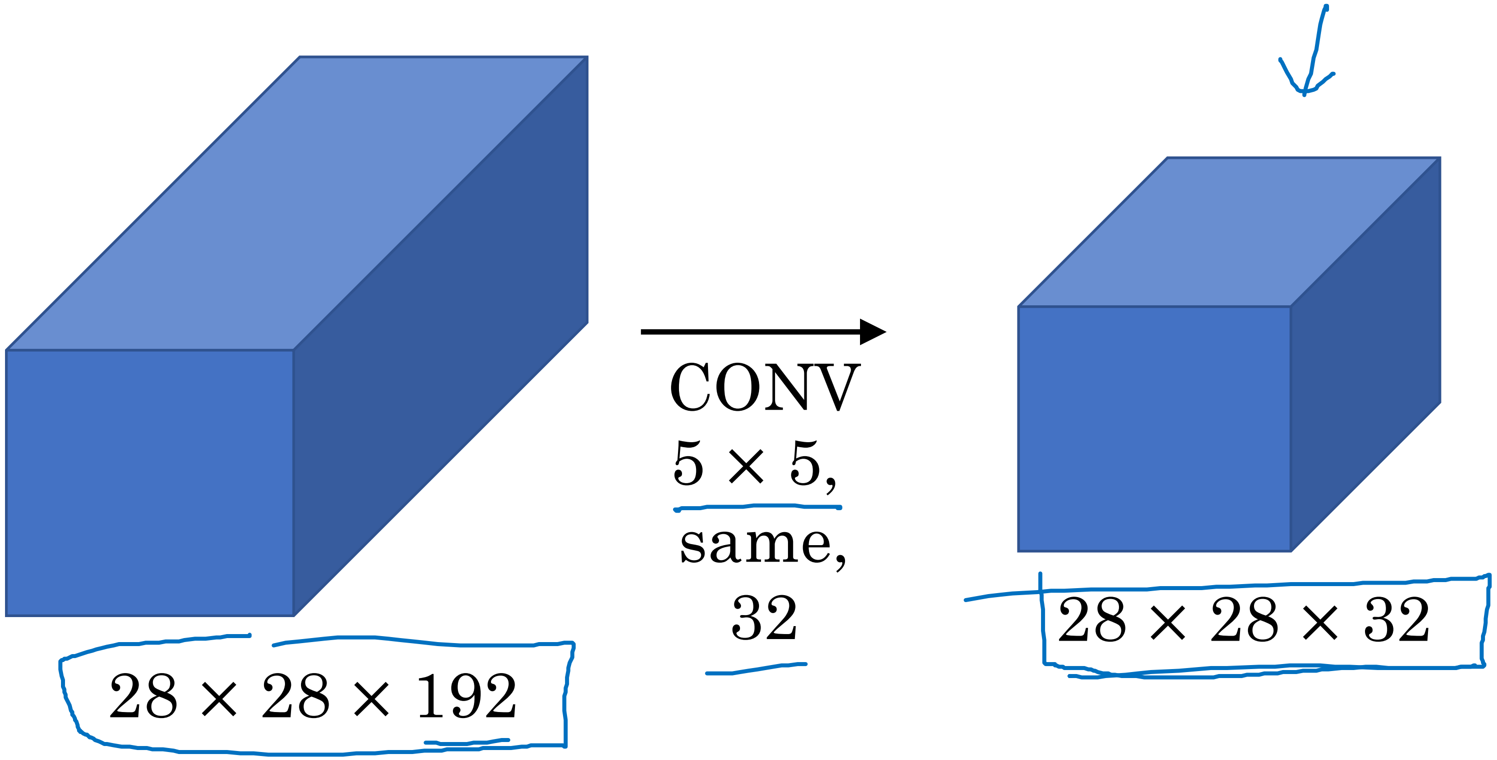
Motivation for inception network



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

Andrew Ng

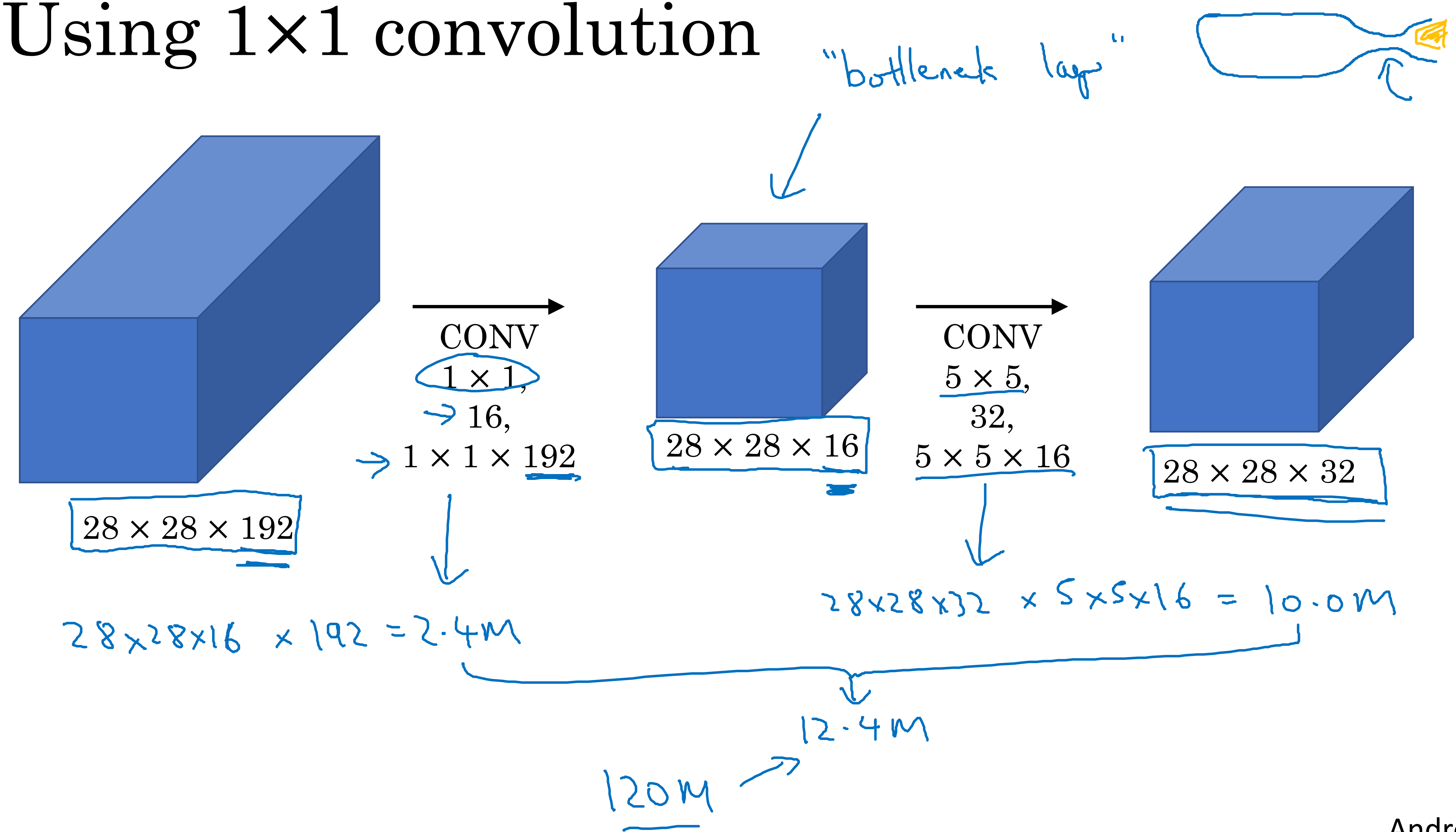
The problem of computational cost



32 filters. filters are 5×5×192.

28×28×32 × 5×5×192 = 120M.

Using 1×1 convolution

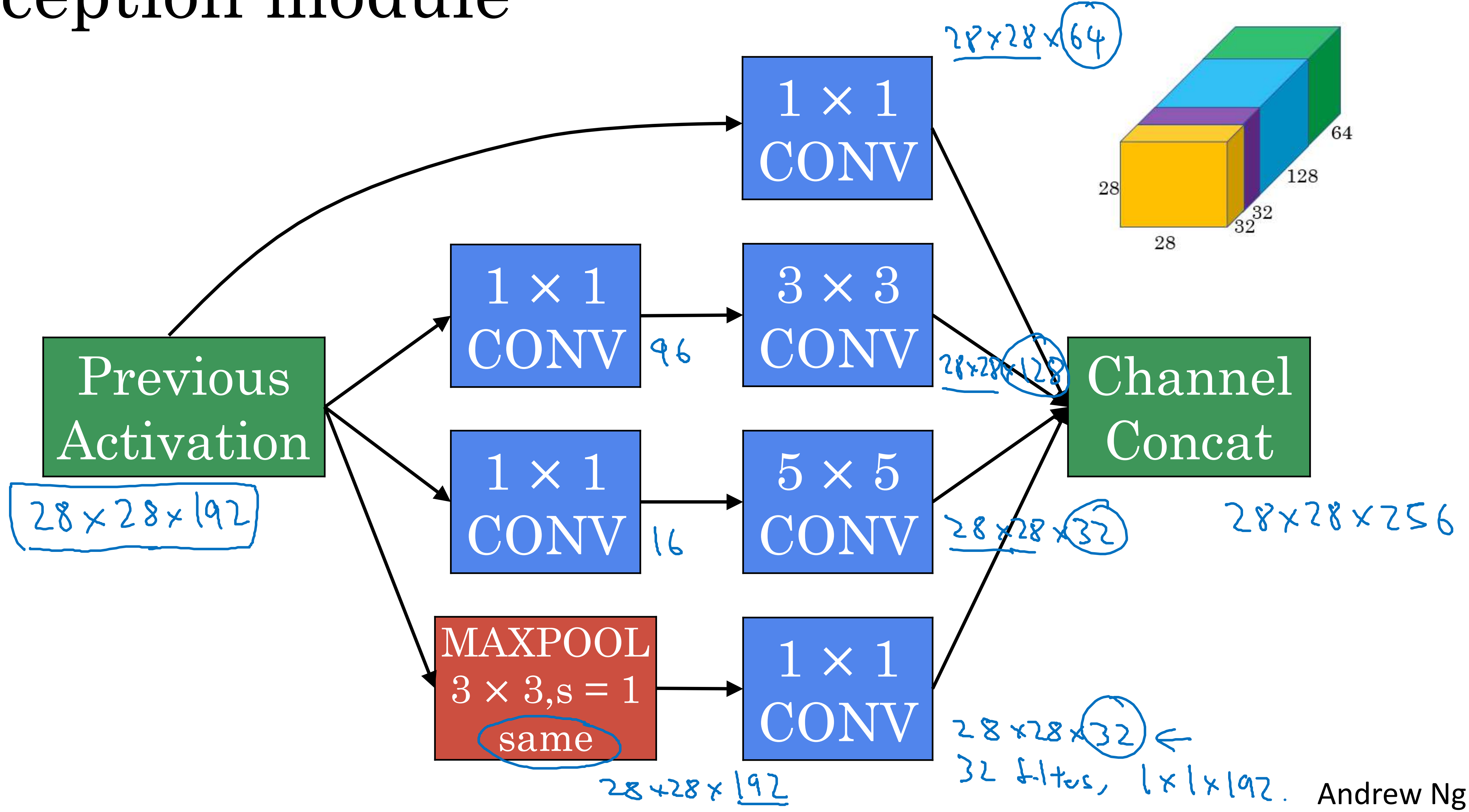


deeplearning.ai

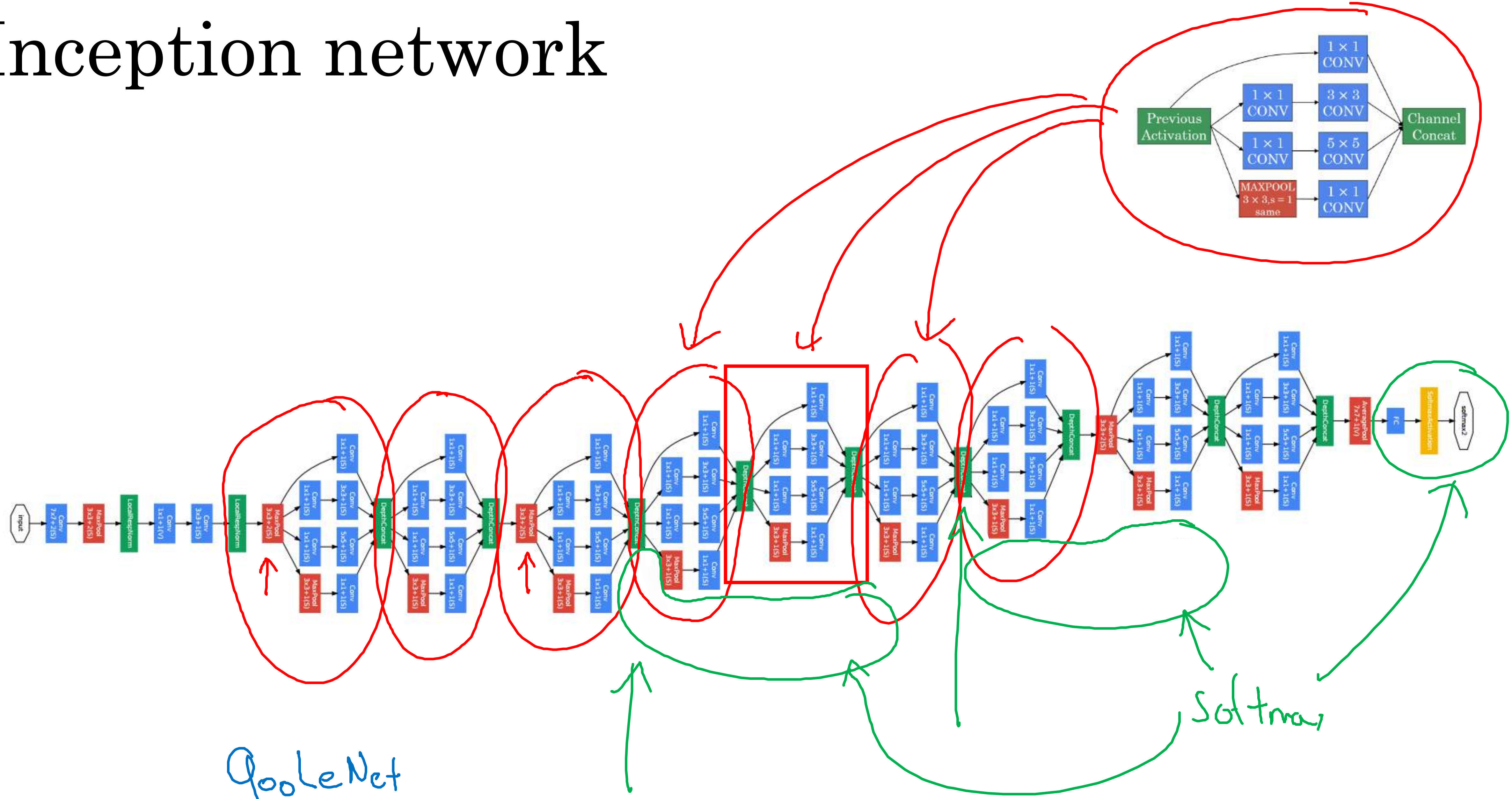
Case Studies

Inception network

Inception module



Inception network



GooLeNet

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

Andrew Ng

http://knowyourmeme.com/memes/we-need-to-go-deeper