deeplearning.ai

Case Studies

Inception network motivation
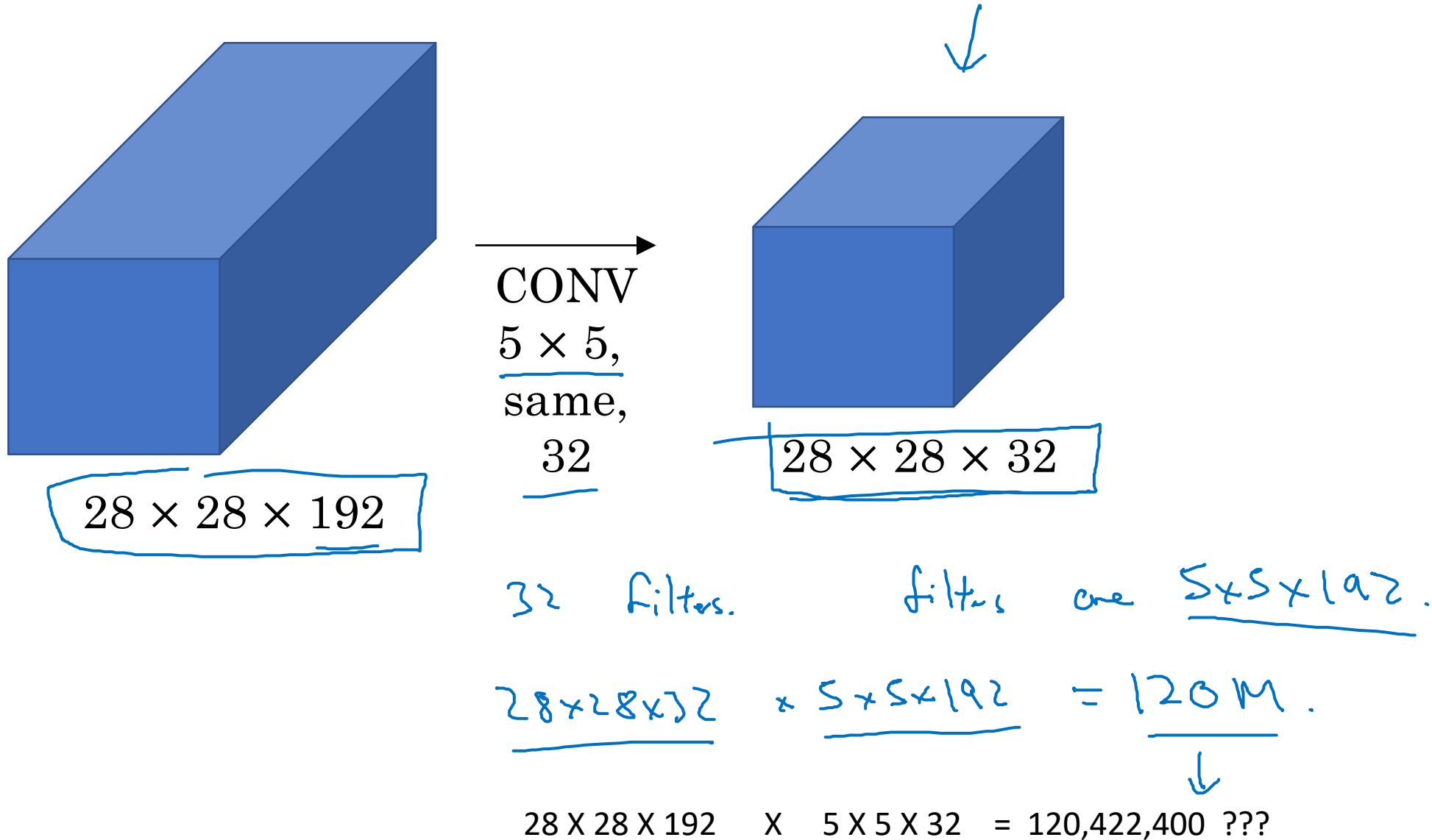
Motivation for inception network

$28 \times 28 \times 64$

1×1

3×3 $28 \times 28 \times 128$
Same

5×5 $28 \times 28 \times 32$
Same

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

$28 \times 28 \times 192$

28

28   32   128   64
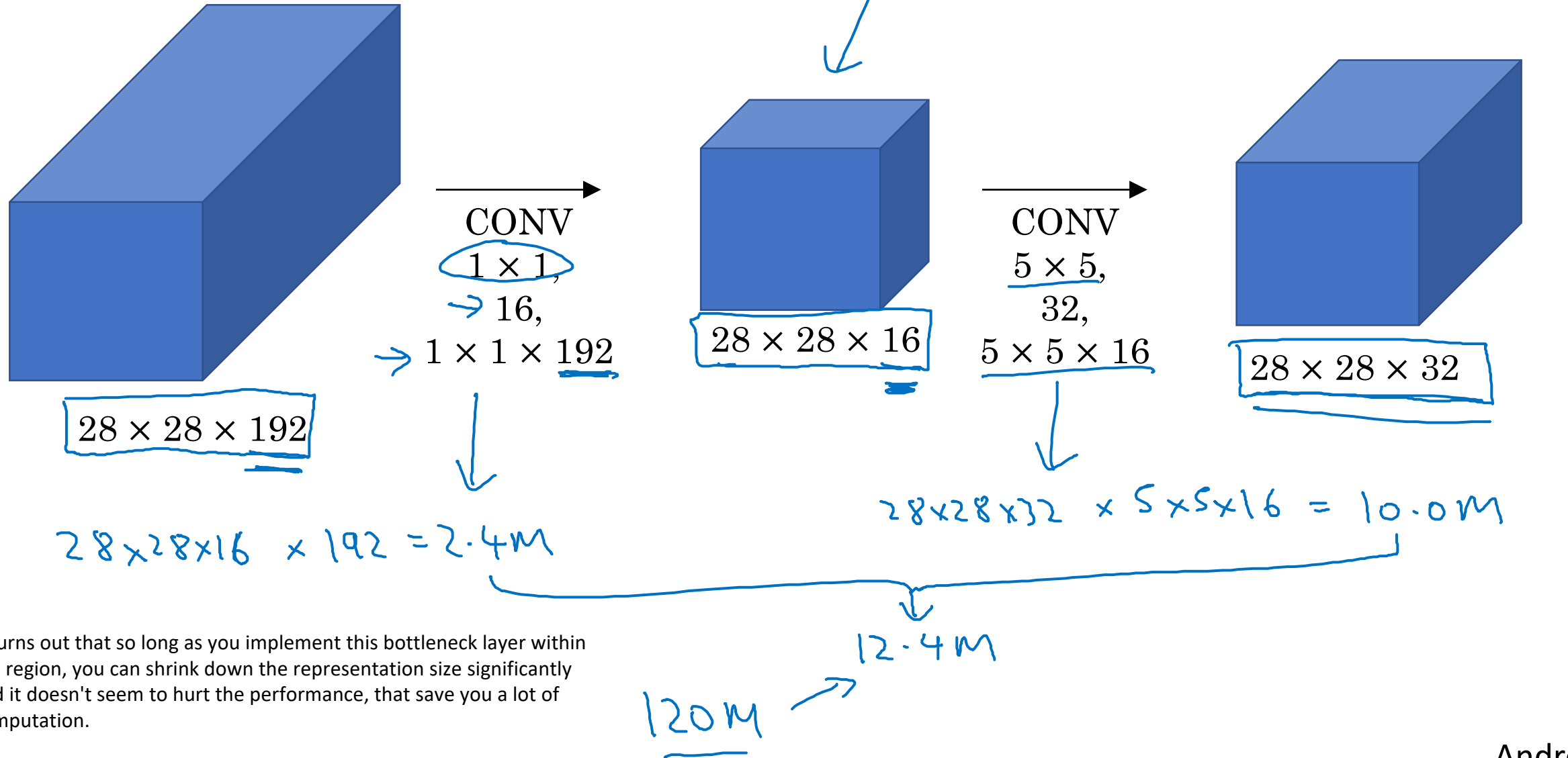32

256

$28 \times 28 \times 256$
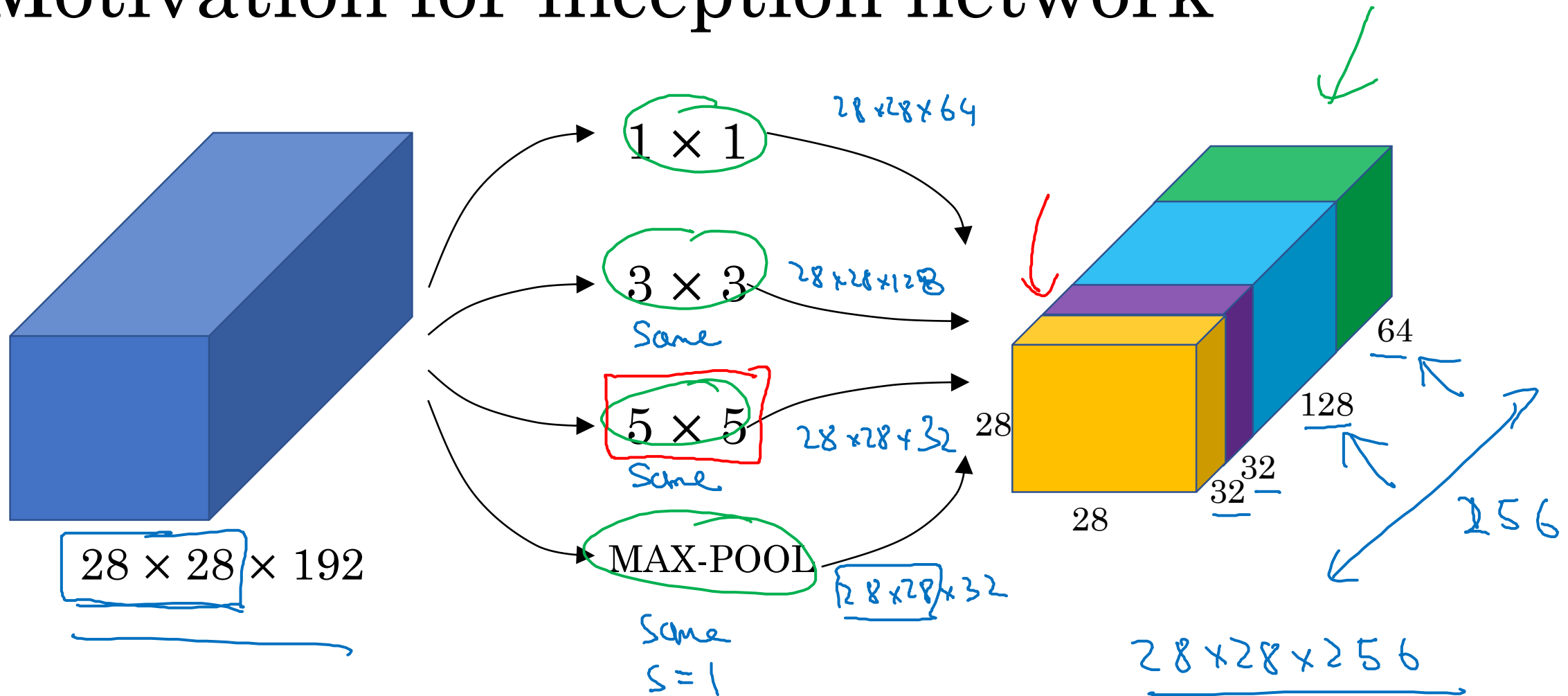
And the basic idea is that instead of you needing to pick what of these filter sizes or pooling you want and committing to that, you can do them all and concatenate all the outputs and let the network learn whatever parameters it wants to use, what are the combinations of these filter sizes it wants.

Now, it turns out that there's a problem with the inception layer as I've describe it here which is computational cost.

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

Andrew Ng

The problem of computational cost

CONV
5 × 5,
same,
32

$28 \times 28 \times 192$

$28 \times 28 \times 32$

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

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

28 X 28 X 192   X   5 X 5 X 32   = 120,422,400 ???

Andrew Ng

Using 1×1 convolution

"bottleneck layer"

CONV
1 × 1,
→ 16,
→ 1 × 1 × 192

$28 \times 28 \times 192$

$28 \times 28 \times 16$

CONV
5 × 5,
32,
5 × 5 × 16

$28 \times 28 \times 32$

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

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

$12.4M$

$120M \to$

It turns out that so long as you implement this bottleneck layer within the region, you can shrink down the representation size significantly and it doesn't seem to hurt the performance, that save you a lot of computation.

Andrew Ng