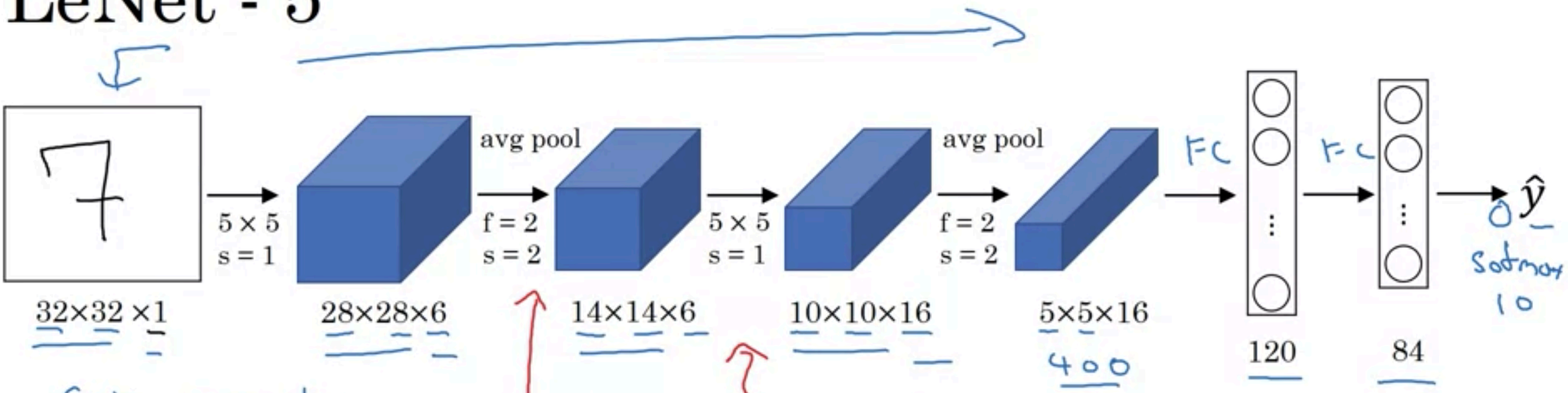
LeNet - 5



60K parameters.

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

non-linearity after pooling

$n_H \times n_W \times n_C$ $f \times f \times n_C$

conv pool conv pool fc fc output

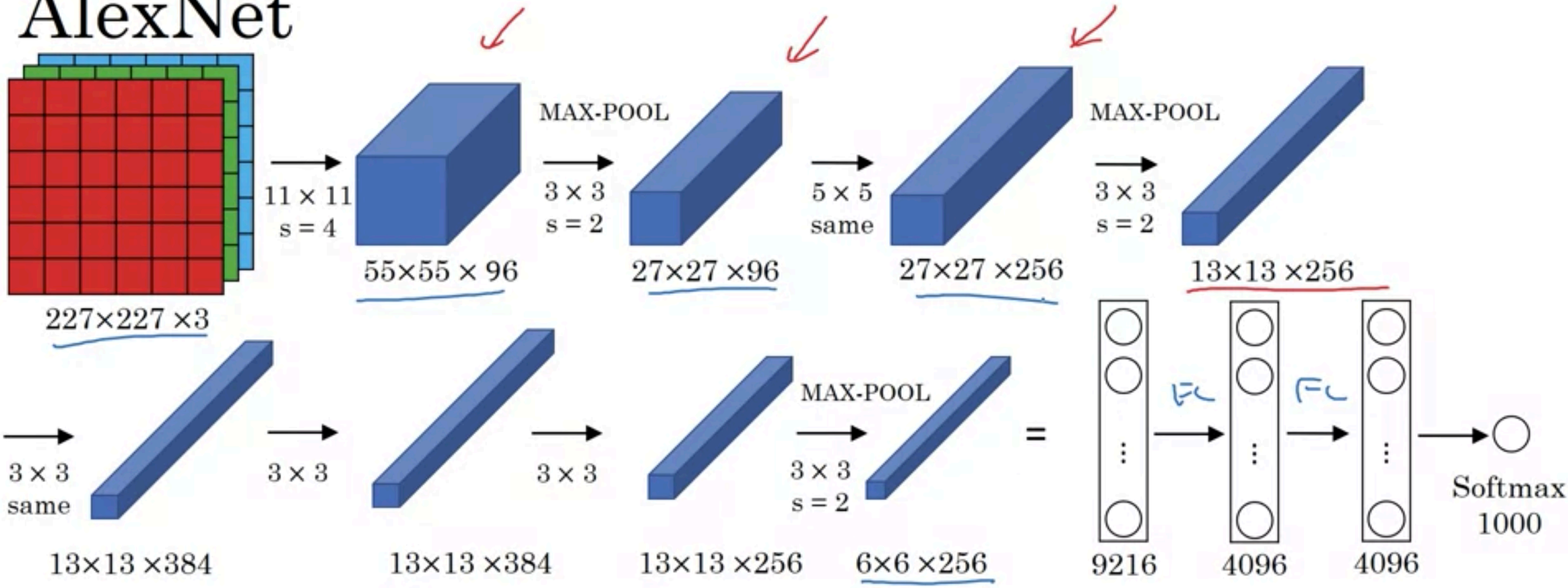
Activation: sigmoid/tanh ReLU

II III

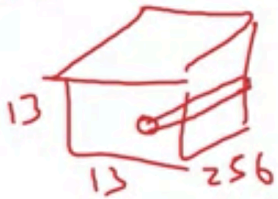
[LeCun et al., 1998. Gradient-based learning applied to document recognition]

Andrew Ng

AlexNet



- Similar to LeNet, but much bigger. 9216 60M parameters
- ReLU
- Multiple GPUs.
- Local Response Normalization (LRN)



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

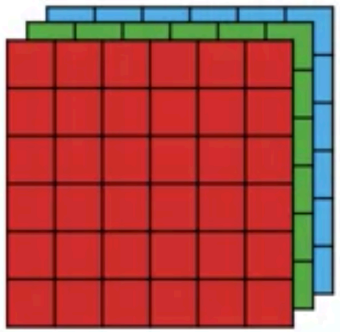
Andrew Ng

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

$n_H, n_W \downarrow$

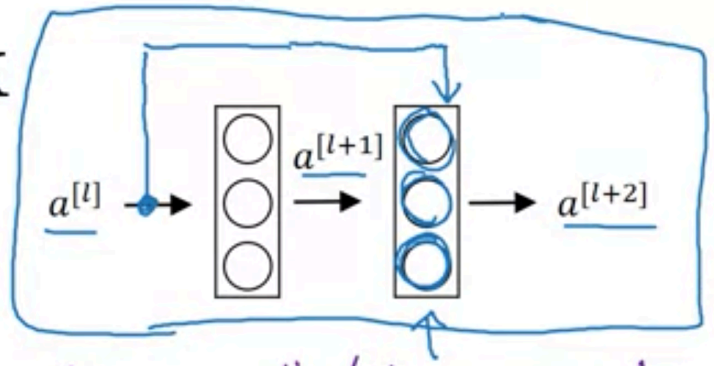
$n_c \uparrow$

~138M

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

Andrew Ng

Residual block



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

"main pain"

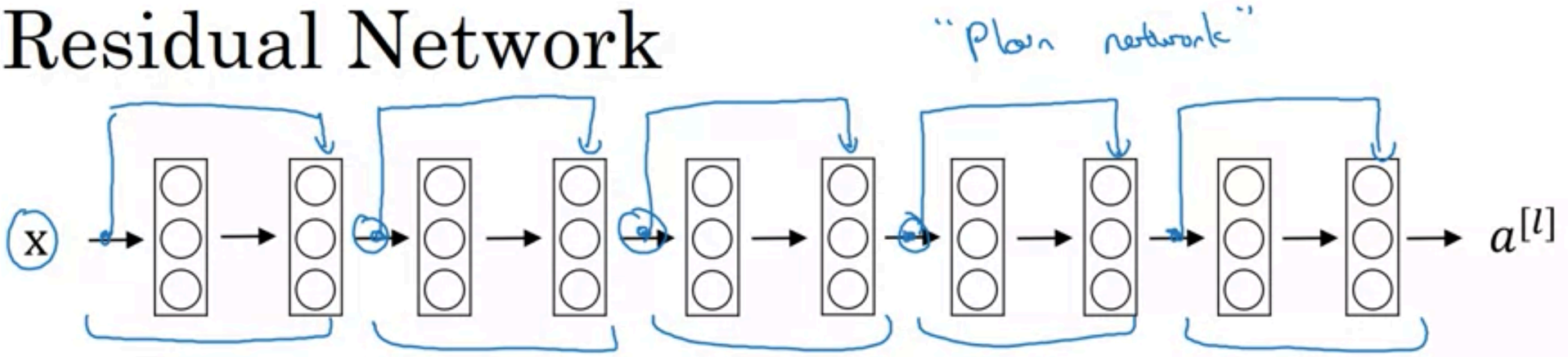
$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \qquad a^{[l+1]} = g(z^{[l+1]}) \qquad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]} \qquad \cancel{a^{[l+2]} = g(z^{[l+2]})}$$

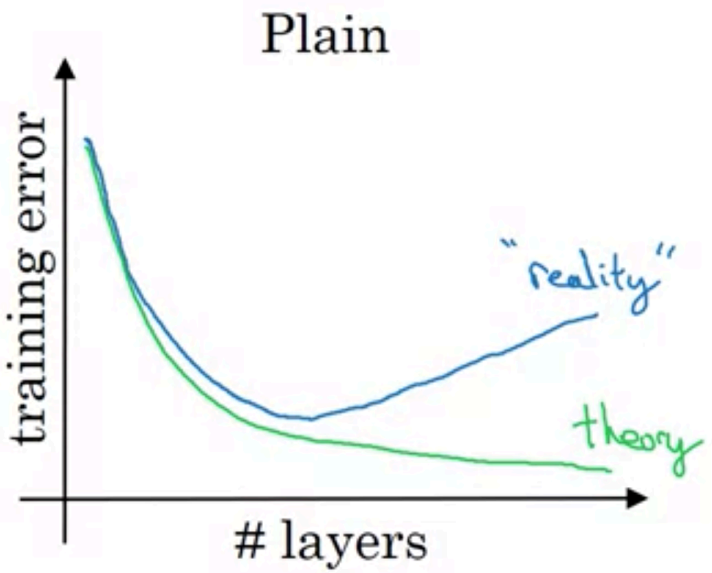
$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

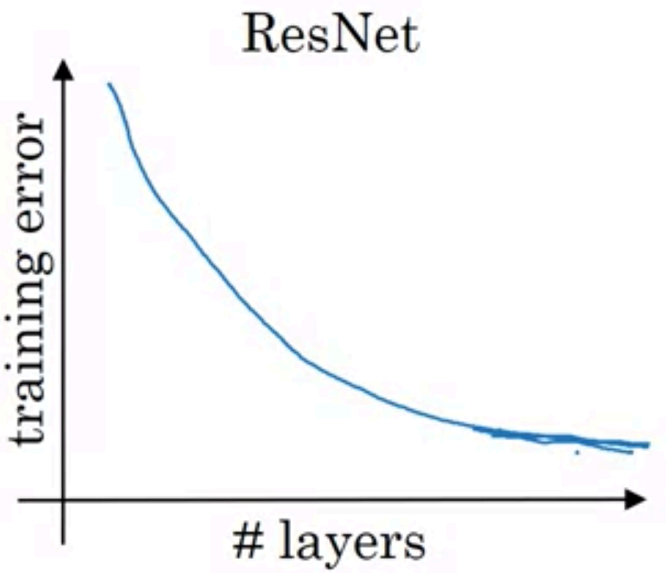
[He et al., 2015. Deep residual networks for image recognition]

Andrew Ng

Residual Network



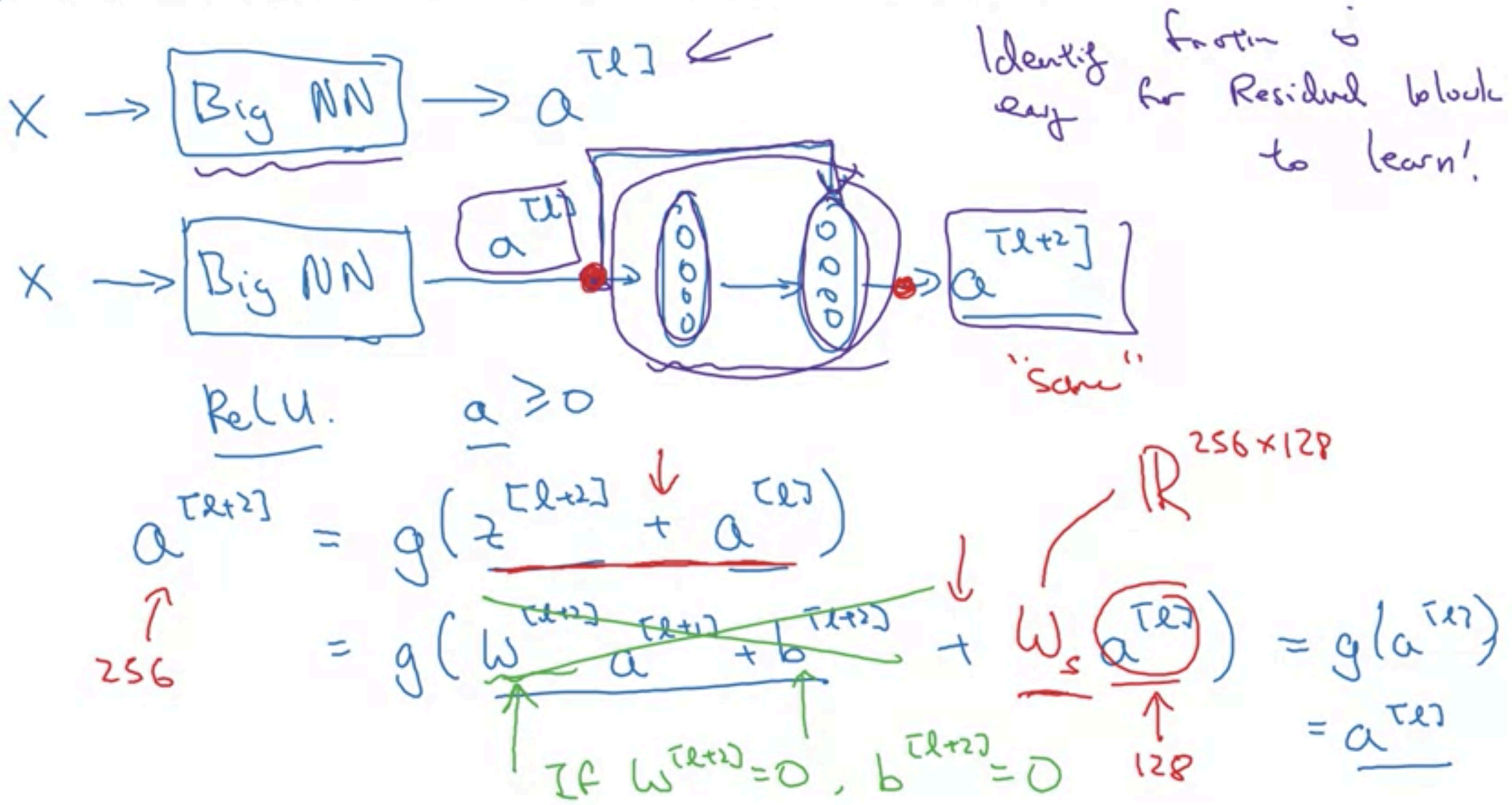




[He et al., 2015. Deep residual networks for image recognition]

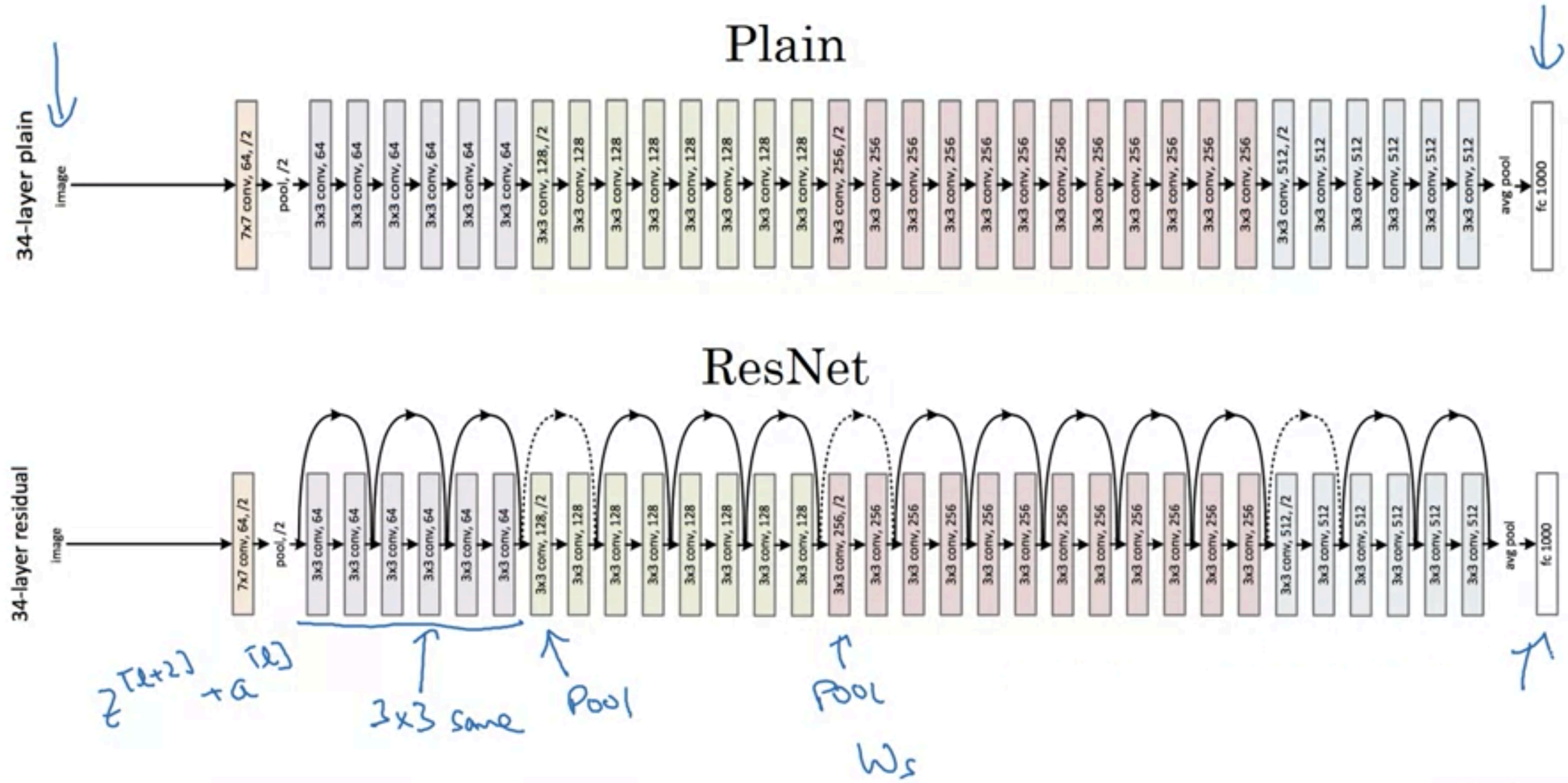
Andrew Ng

Why do residual networks work?



Andrew Ng

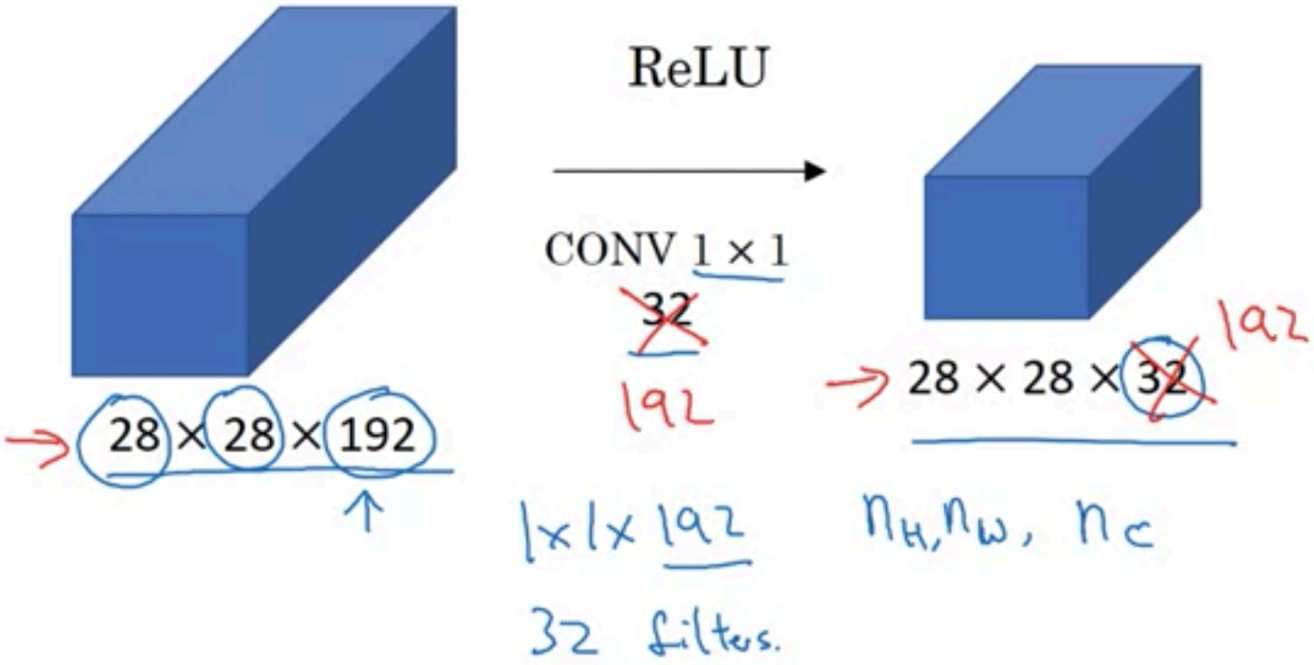
ResNet



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

Andrew Ng

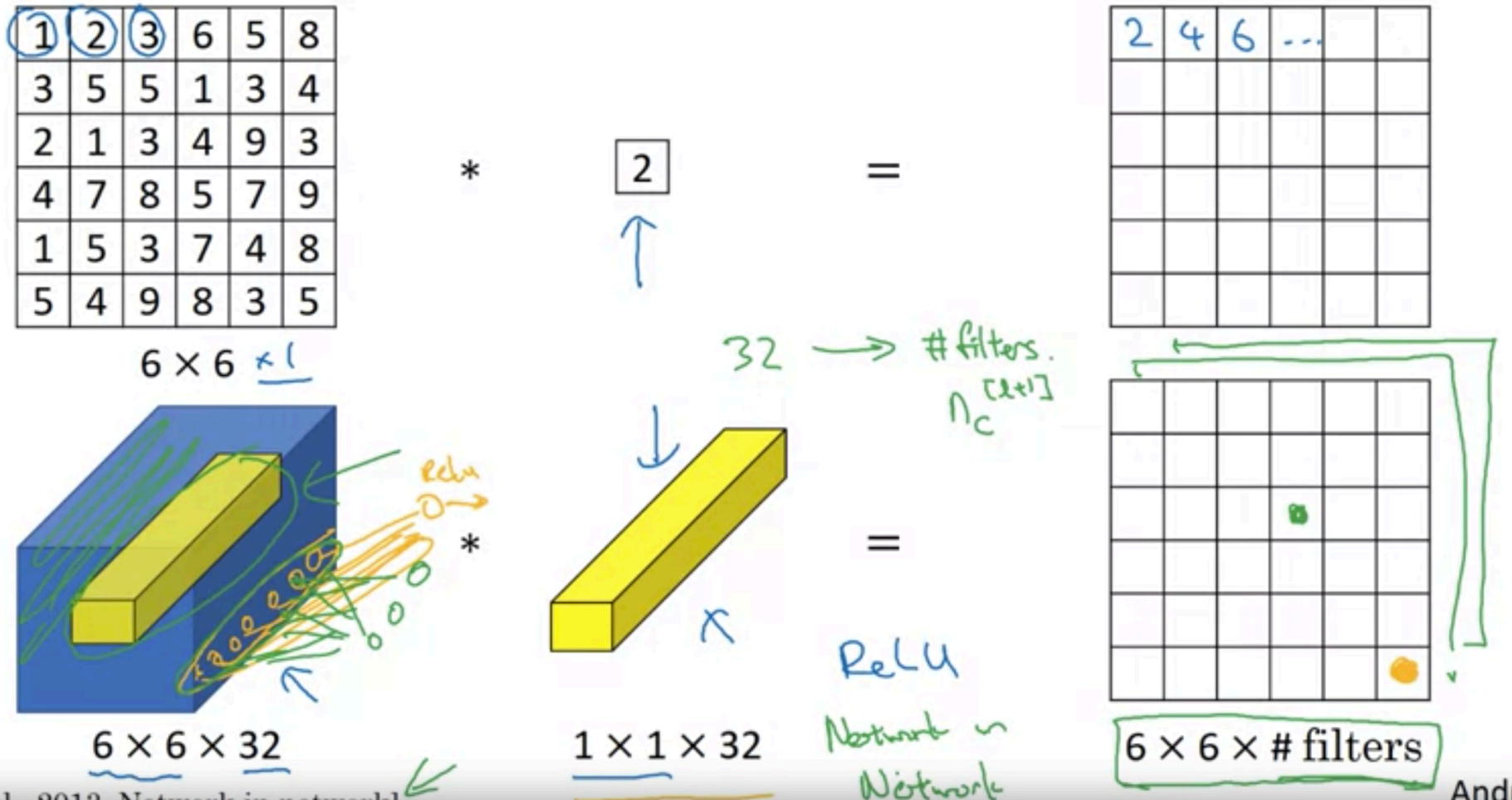
Using 1×1 convolutions



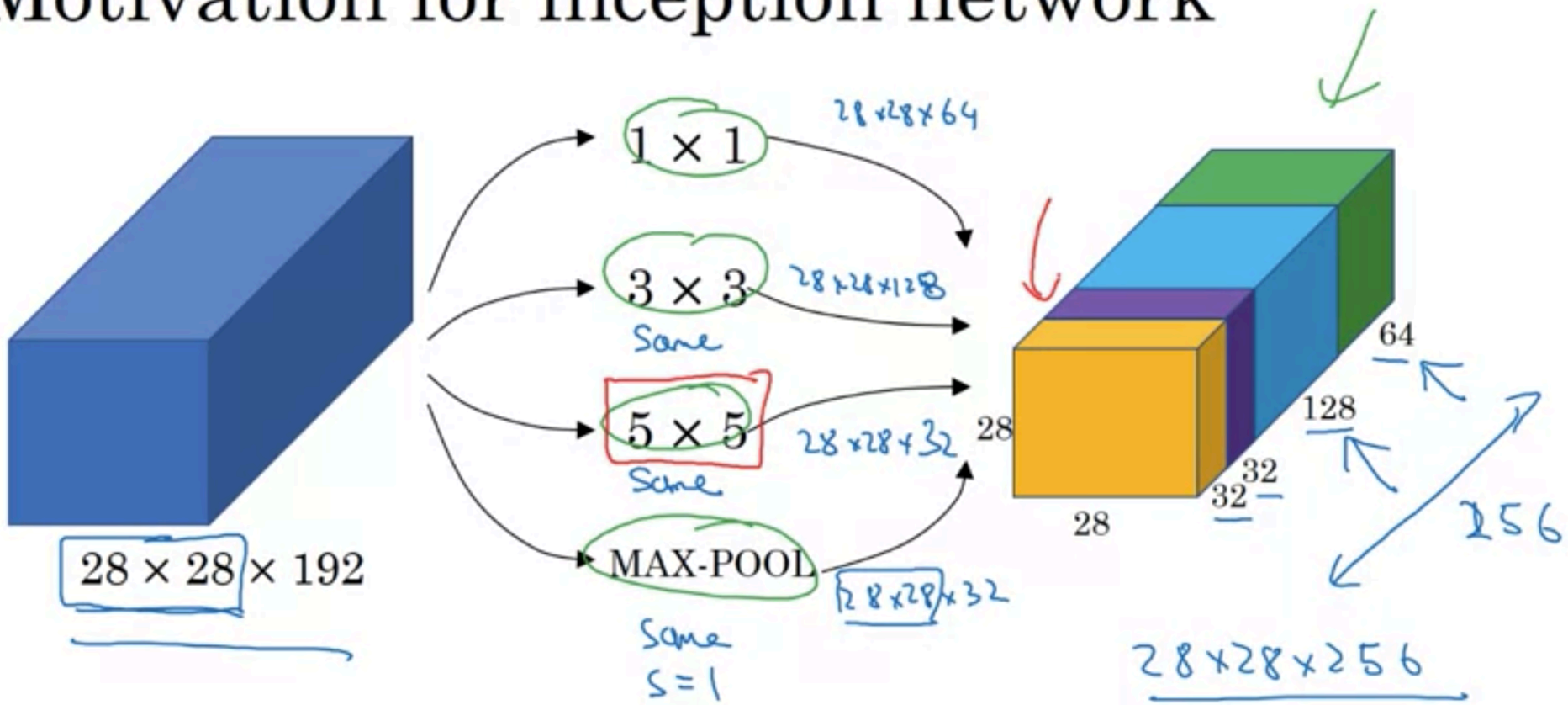
[Lin et al., 2013. Network in network]

Andrew Ng

Why does a 1 × 1 convolution do?

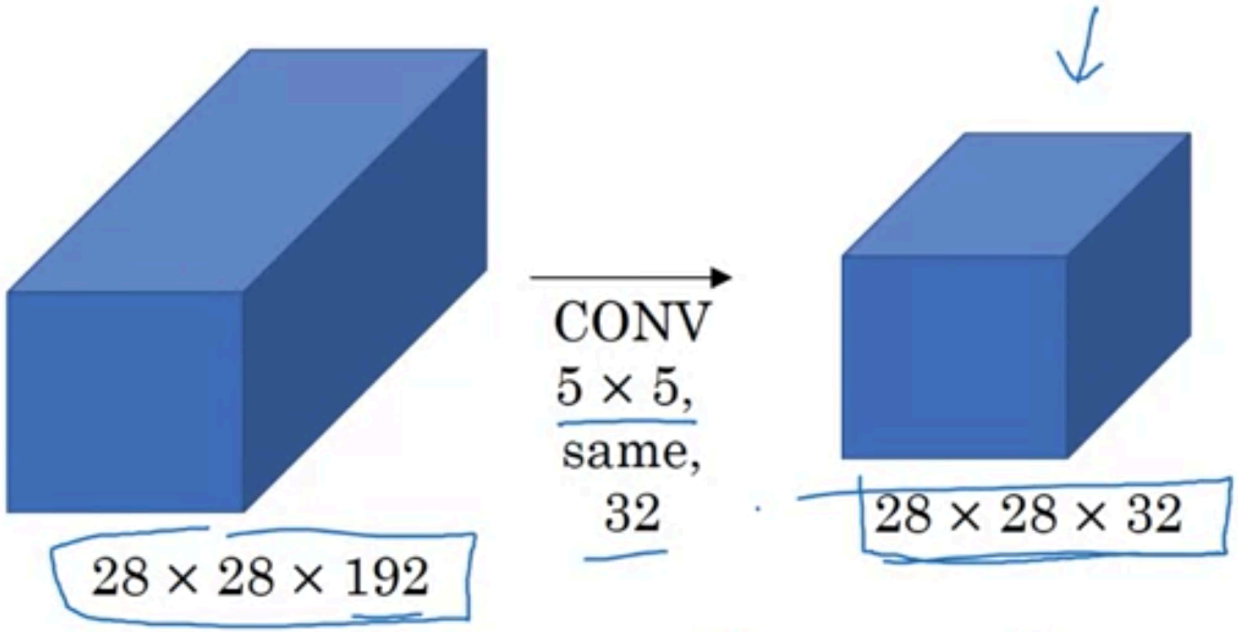


Motivation for inception network



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

The problem of computational cost



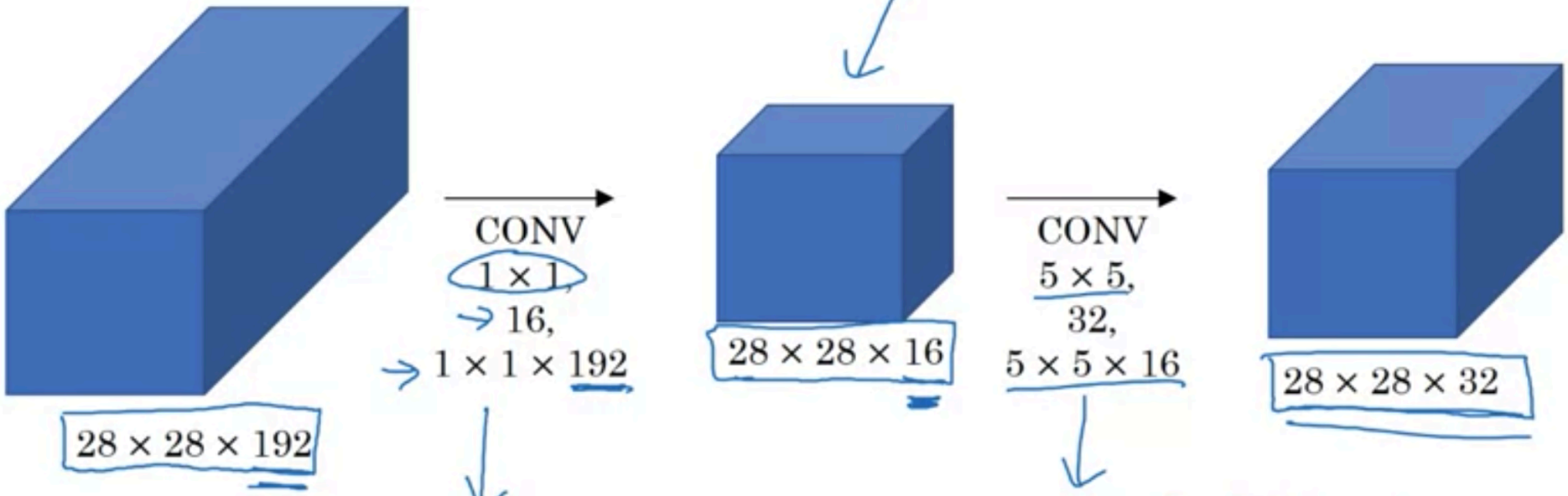
32 filters. filter one 5×5×192.

28×28×32 × 5×5×192 = 120M.

Andrew Ng

Using 1×1 convolution

"bottleneck layer"



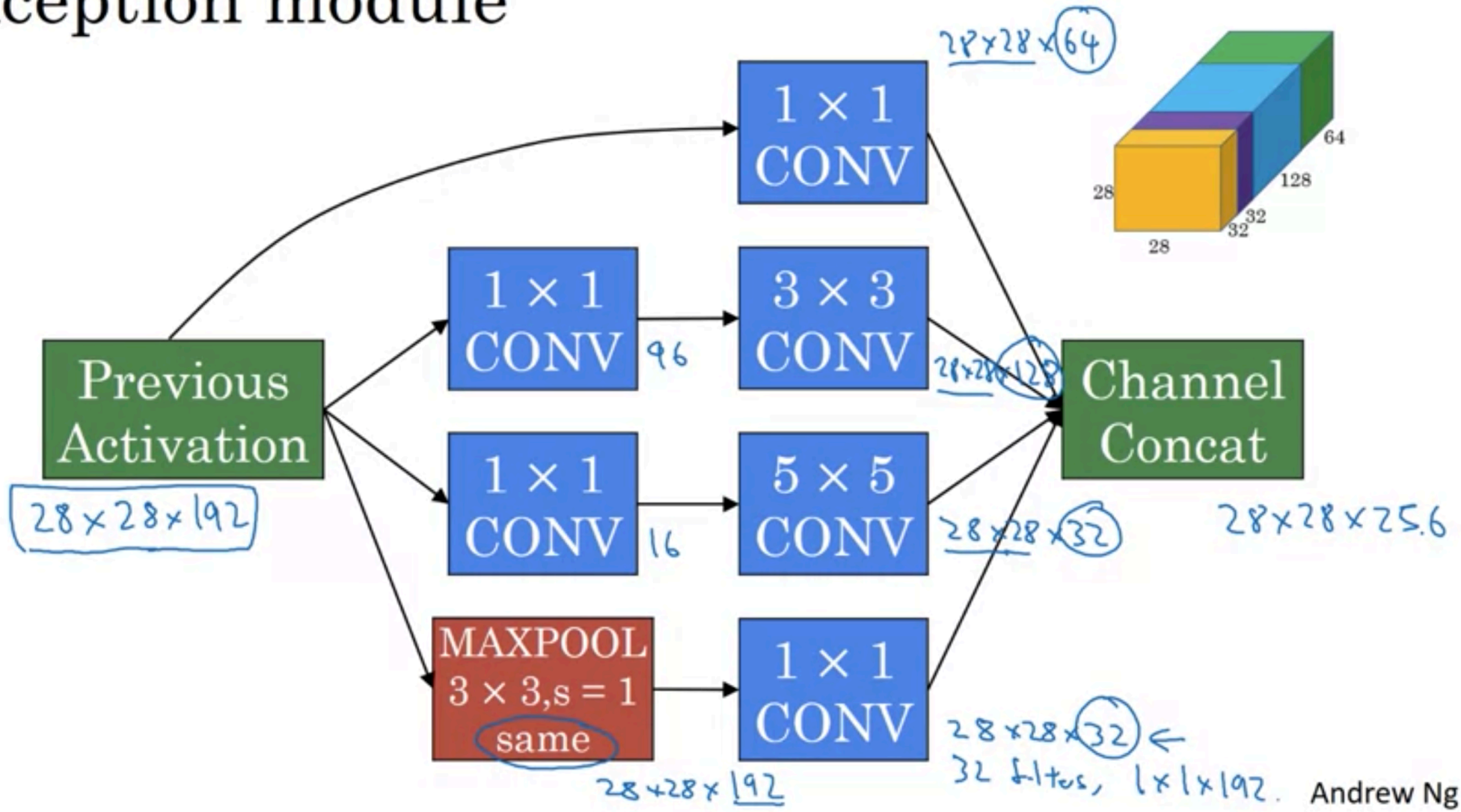
$28 \times 28 \times 16 \times 192 = 2.4\text{M}$

$28 \times 28 \times 32 \times 5 \times 5 \times 16 = 10.0\text{M}$

12.4M

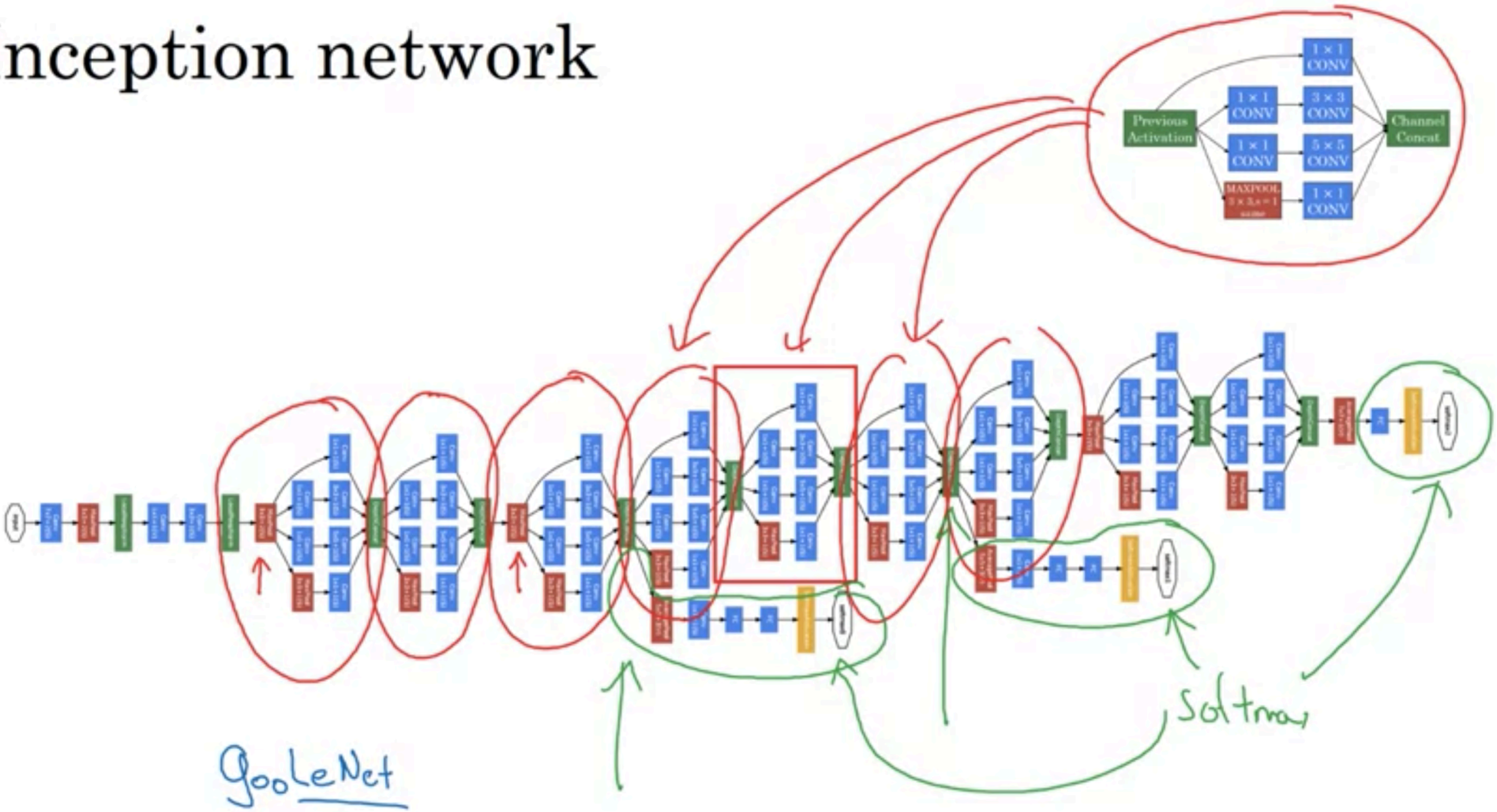
120M

Inception module



Andrew Ng

Inception network

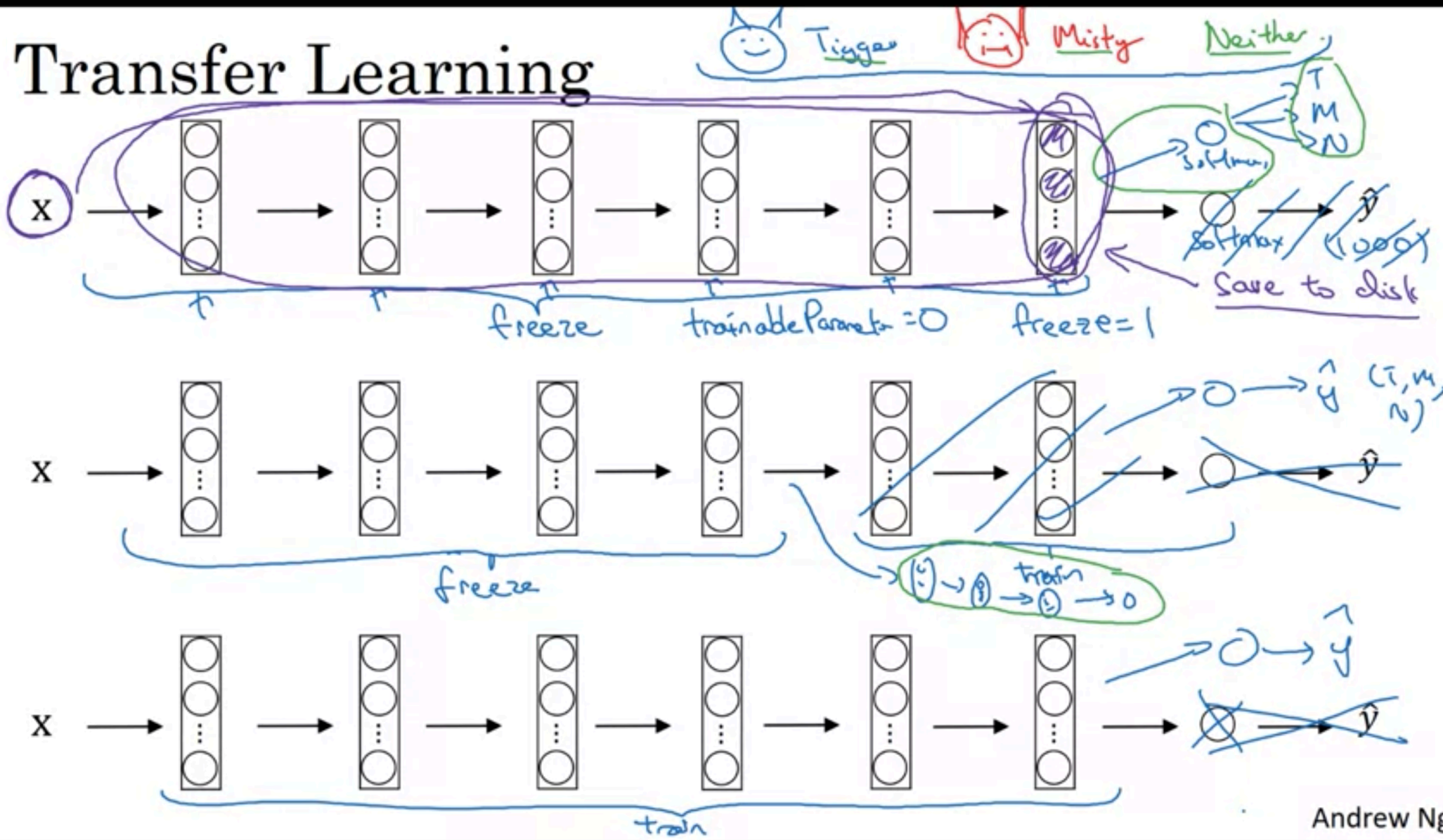


[Szegedy et al., 2014, Going Deeper with Convolutions]

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

Andrew Ng

Transfer Learning



Andrew Ng

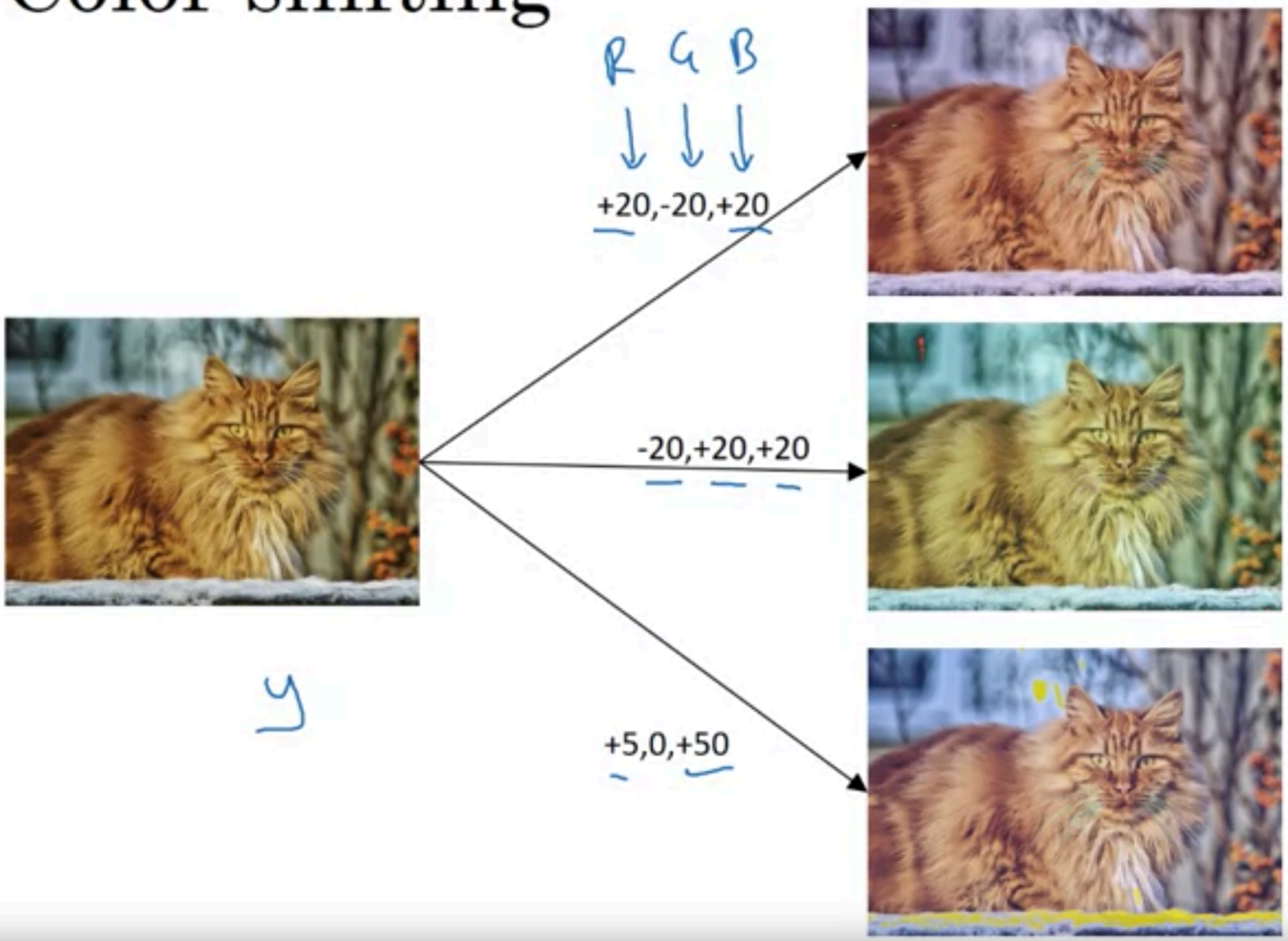
Common augmentation method

Mirroring

Random Cropping

- Rotation
- Shearing
- Local warping
- ...

Color shifting



Advanced:

PCA

ml-class.org

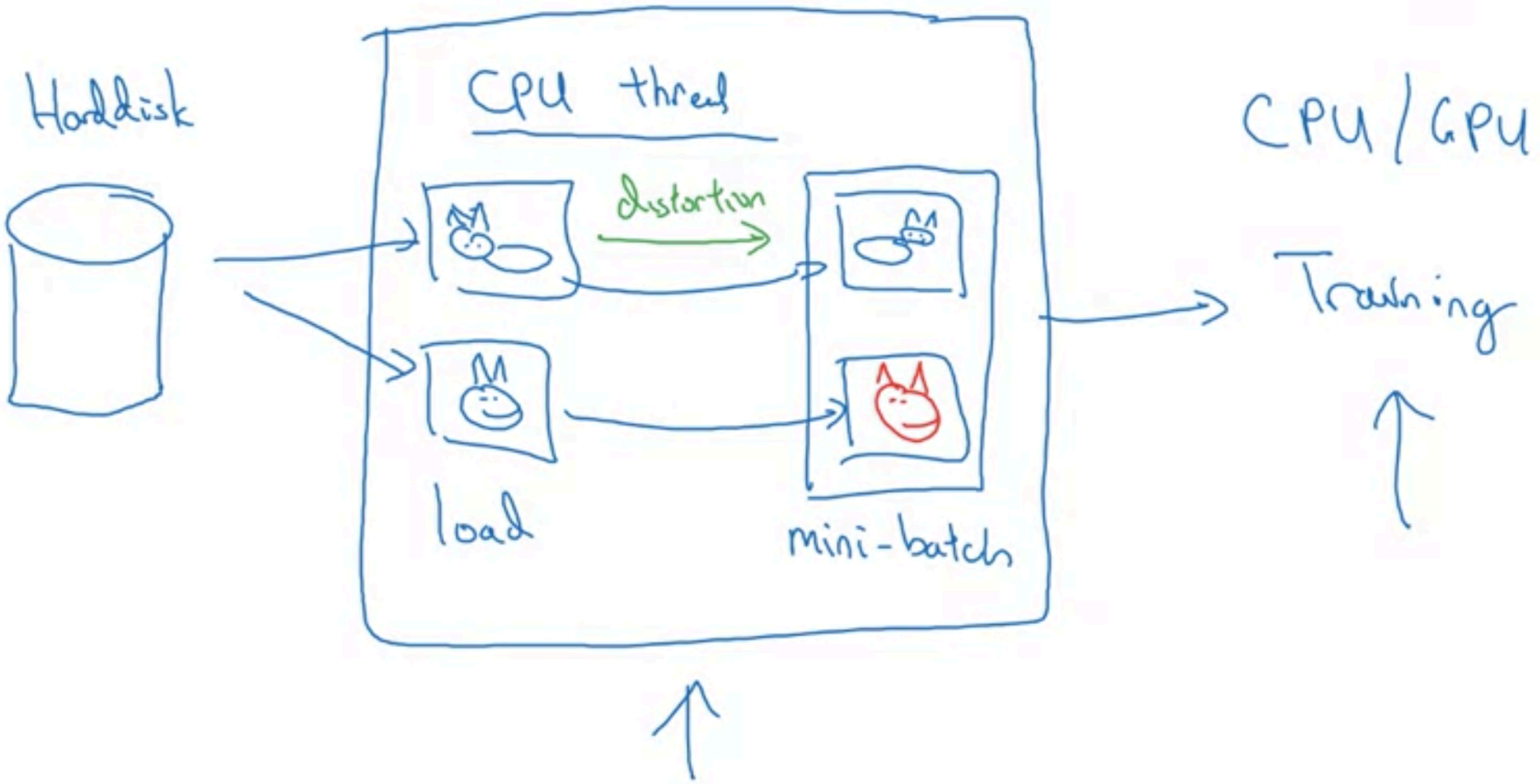
[AlexNet paper

["PCA color augmentation."

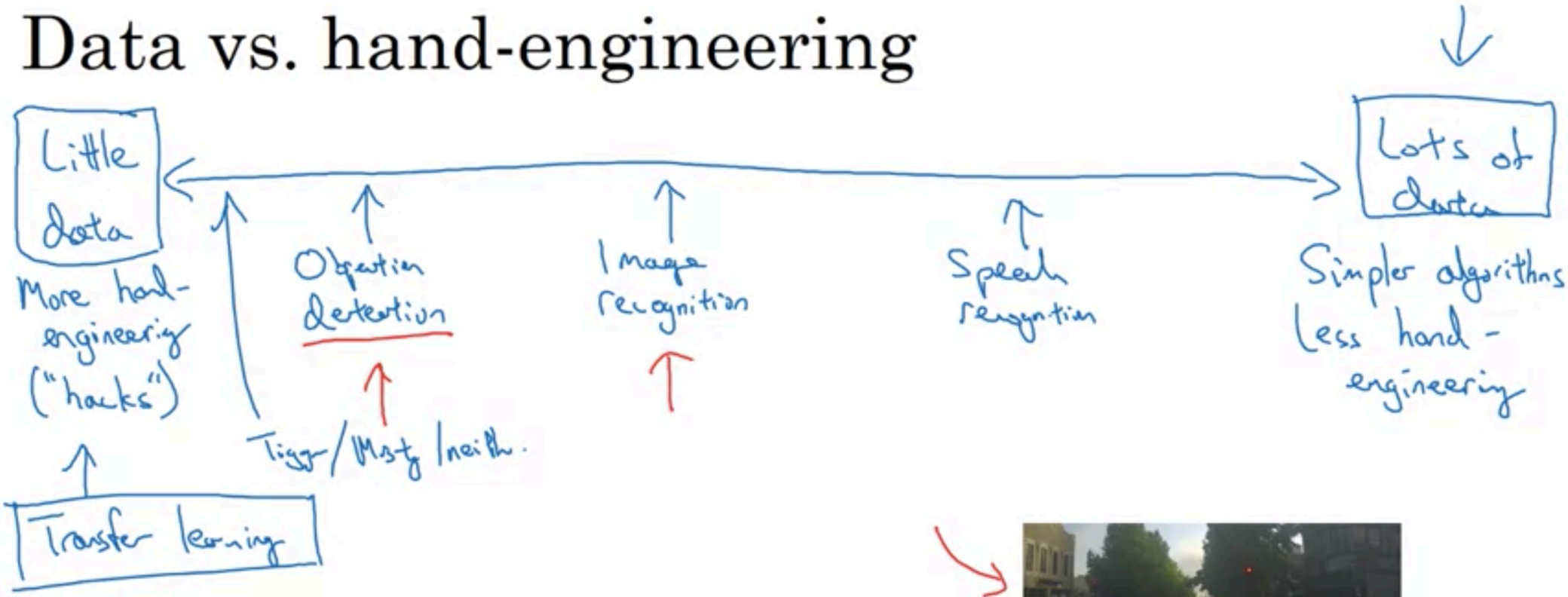
R B G

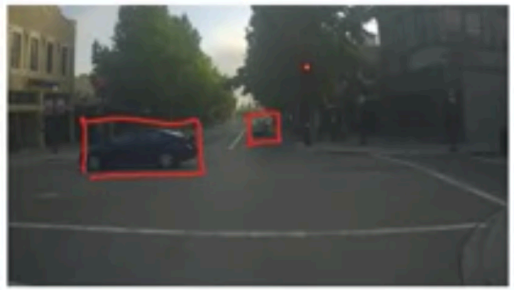
Andrew Ng

Implementing distortions during training



Data vs. hand-engineering





Two sources of knowledge

- Labeled data (x, y)
- Hand engineered features/network architecture/other components

Andrew Ng

Tips for doing well on benchmarks/winning competitions

Ensembling

3-15 networks → $\hat{y}$

- Train several networks independently and average their outputs

Multi-crop at test time

- Run classifier on multiple versions of test images and average results

10-crop

Use open source code

- Use architectures of networks published in the literature
- Use open source implementations if possible
- Use pretrained models and fine-tune on your dataset