



deeplearning.ai

Case Studies

Why look at
case studies?

Outline

Classic networks:

- LeNet-5 ←
- AlexNet ←
- VGG ←

ResNet (152)

Inception

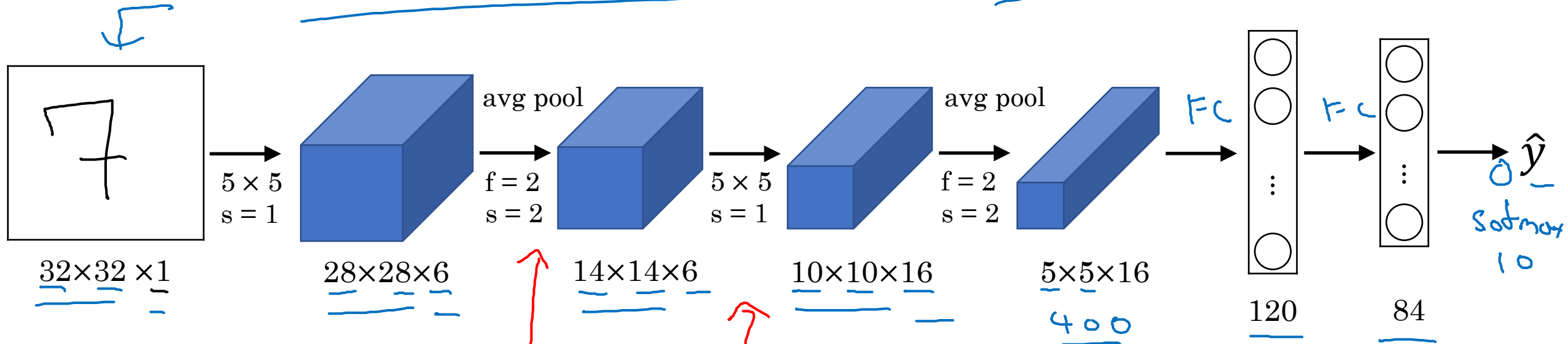


deeplearning.ai

Case Studies

Classic networks

LeNet - 5



60K parameters.

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

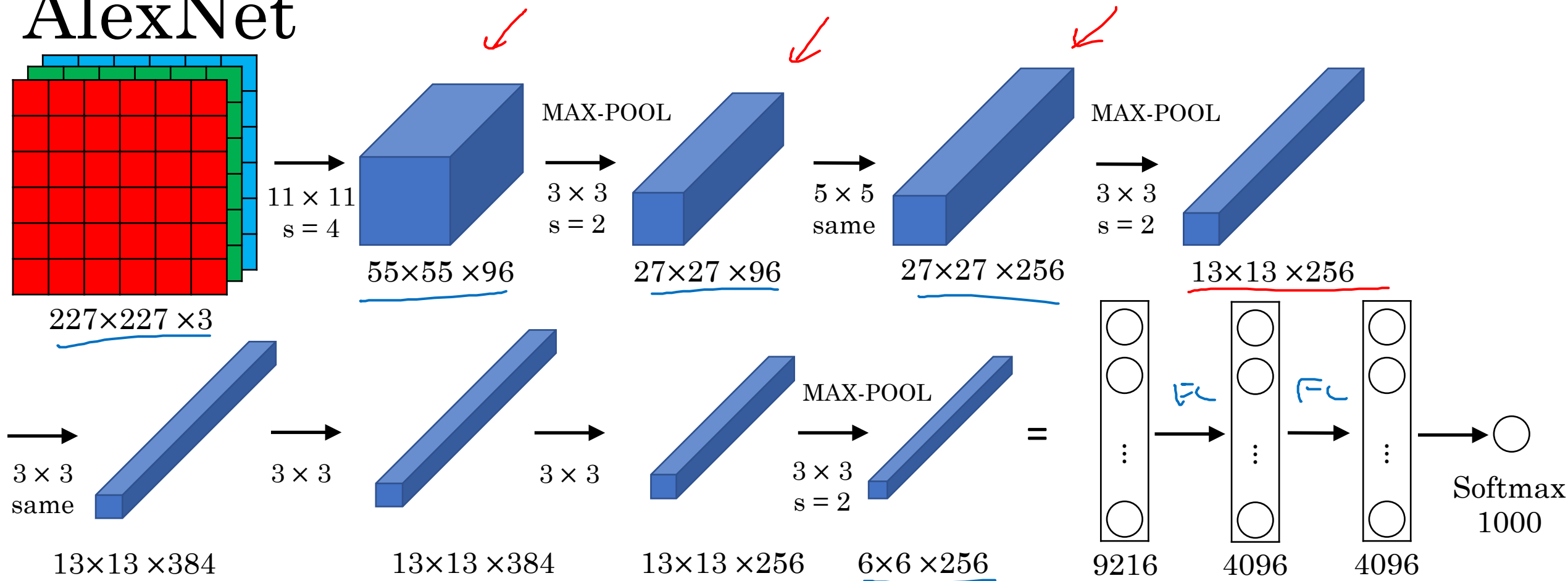
conv pool conv pool fc fc output

Advanced: sigmoid/tanh ReLU

II, III.

↓

AlexNet

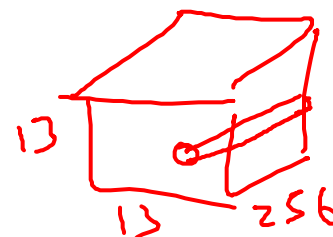


- Similar to LeNet, but much bigger.

- ReLU

- Multiple GPUs.

- Local Response Normalization (LRN)

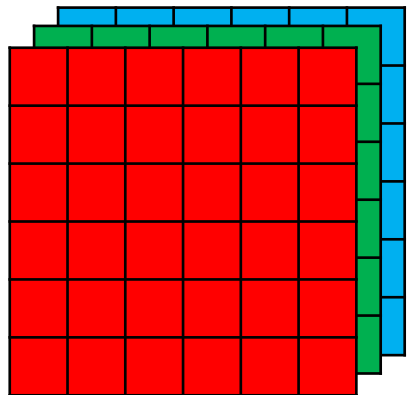


~60M parameters

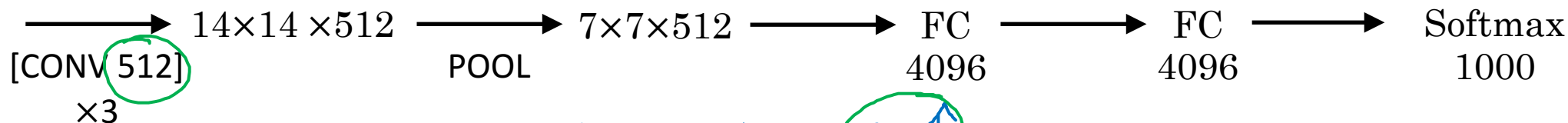
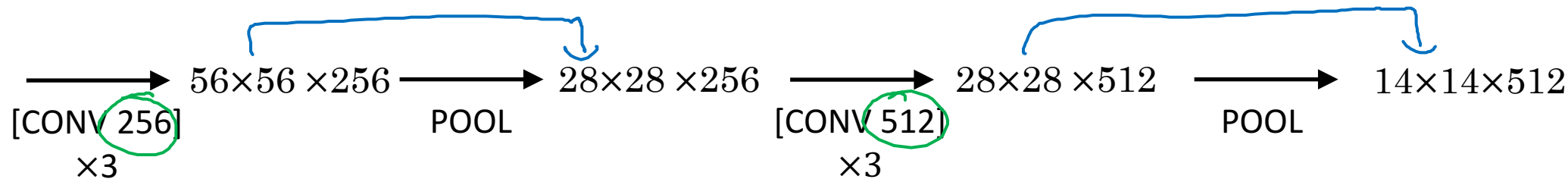
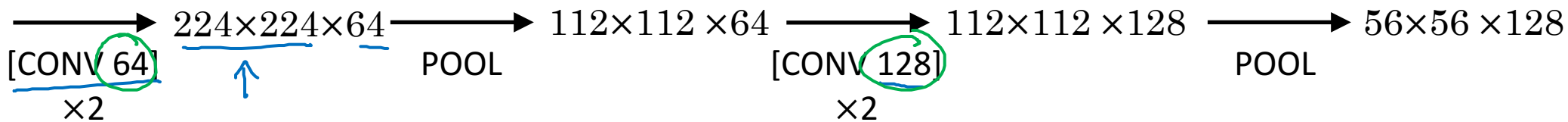
VGG - 16

CONV = 3x3 filter, s = 1, same

MAX-POOL = 2x2, s = 2



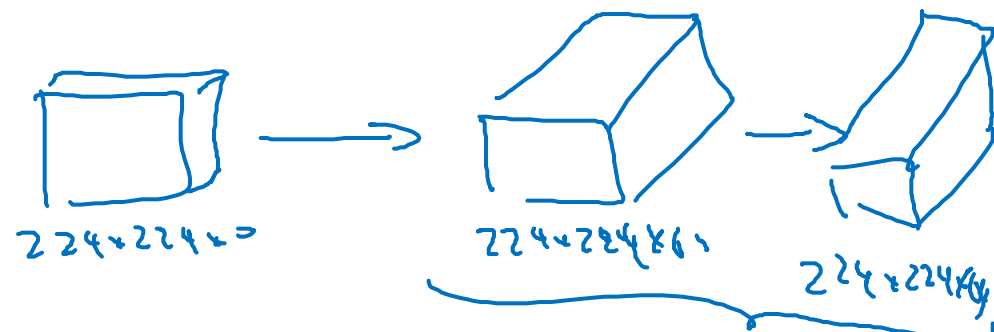
224x224x3



$n_H, n_W \downarrow$

$n_C \uparrow$

$\sim 138M$



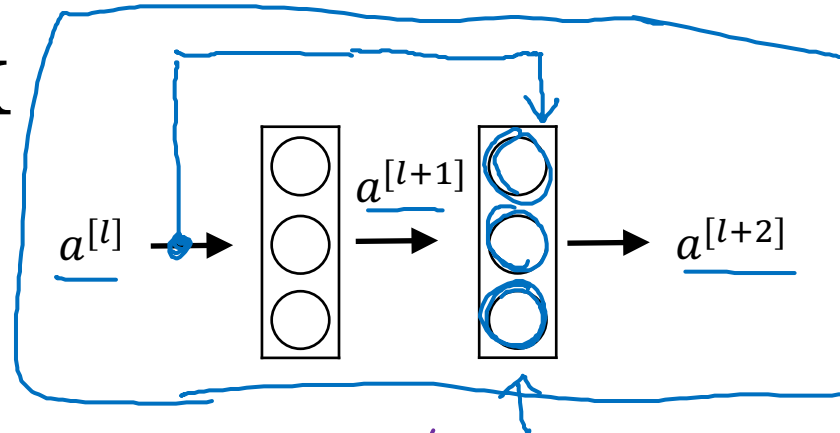


deeplearning.ai

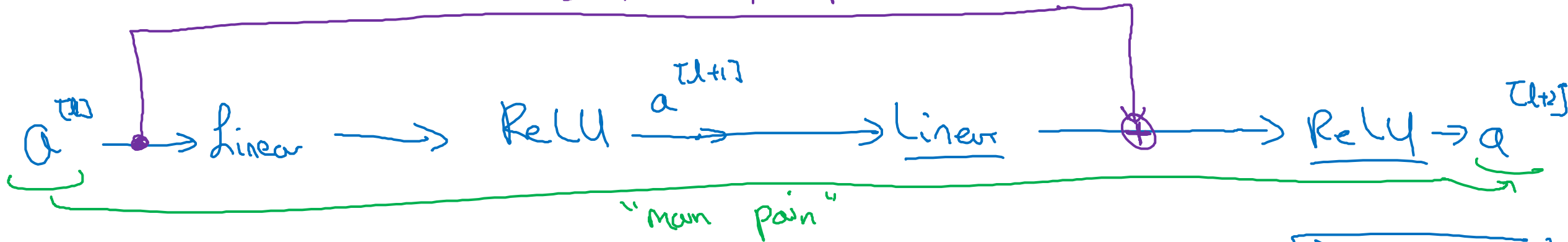
Case Studies

Residual Networks (ResNets)

Residual block



"short cut" / skip connection



$$\underline{z^{[l+1]}} = W^{[l+1]} \underline{a^{[l]}} + b^{[l+1]}$$

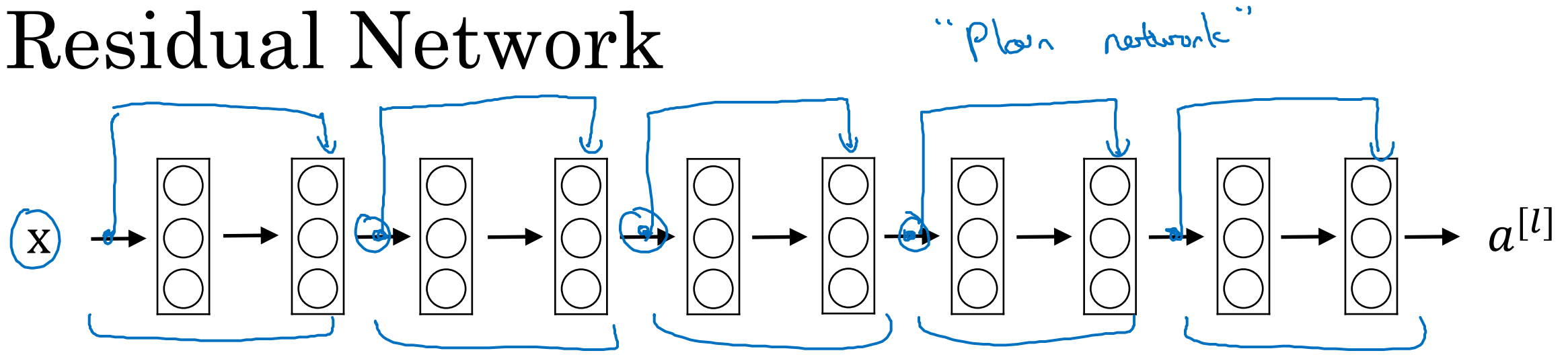
$$\underline{a^{[l+1]}} = g(\underline{z^{[l+1]}})$$

$$\underline{z^{[l+2]}} = W^{[l+2]} \underline{a^{[l+1]}} + b^{[l+2]}$$

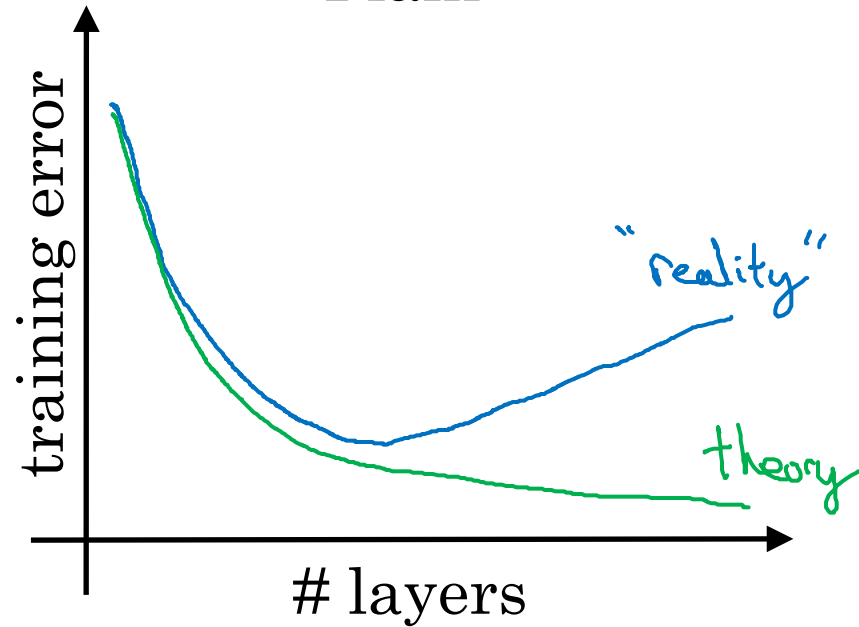
~~$$\underline{a^{[l+2]}} = g(\underline{z^{[l+2]}})$$~~

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

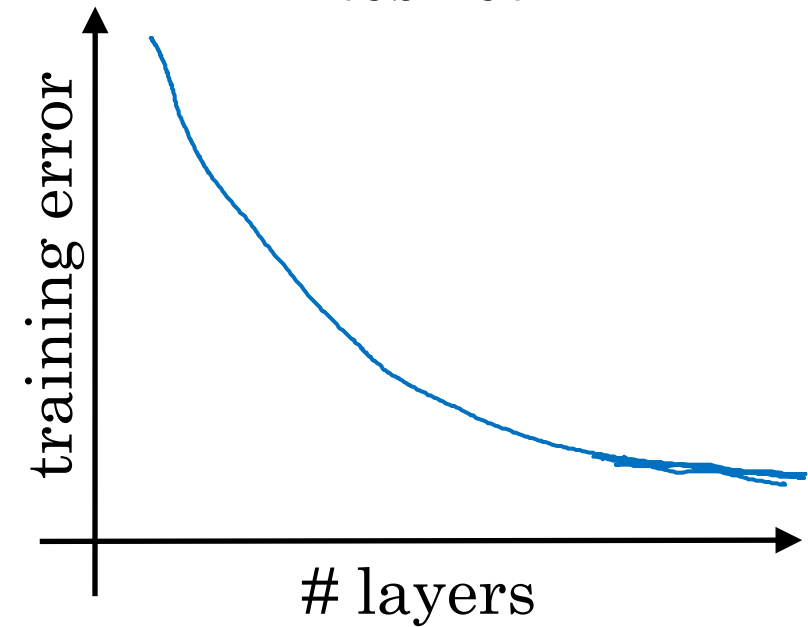
Residual Network



Plain



ResNet



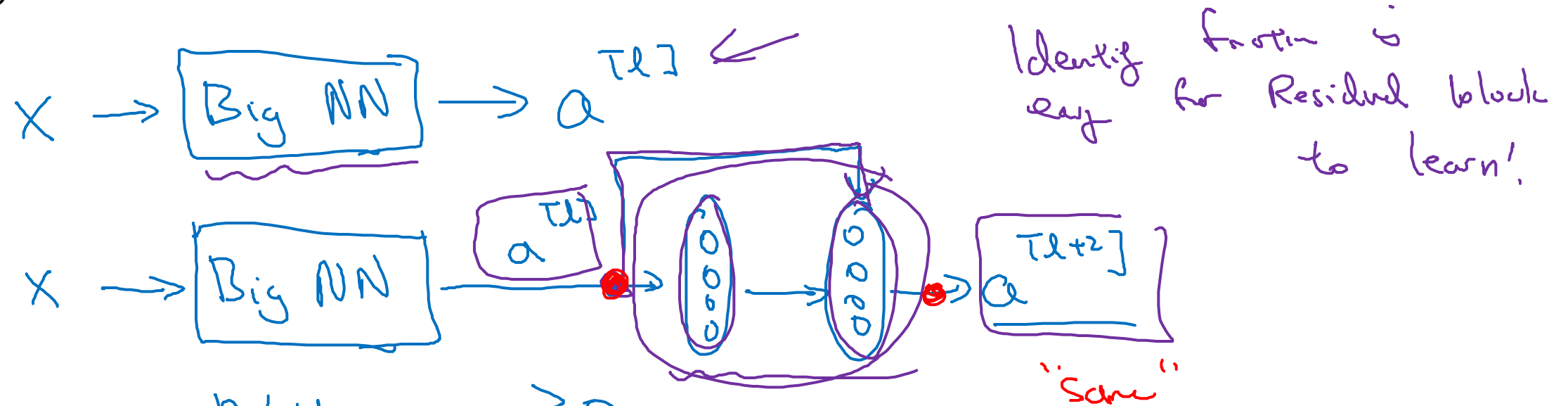


deeplearning.ai

Case Studies

Why ResNets work

Why do residual networks work?



ReLU. $a \geq 0$

$$a^{[l+2]} = g(\underbrace{z^{[l+2]} + a^{[l]}}_{\text{ReLU}})$$

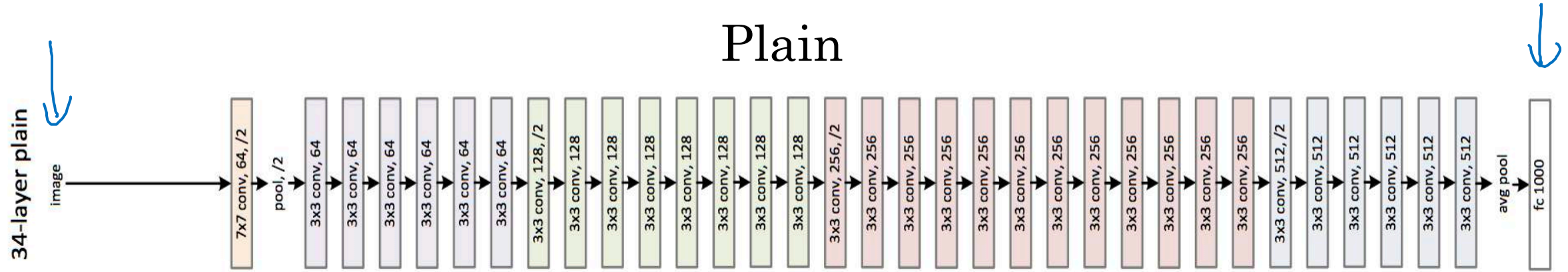
$$= g(\underbrace{W^{[l+2]} a^{[l]} + b^{[l+2]}}_{\text{If } W^{[l+2]}=0, b^{[l+2]}=0} + \underbrace{W_s a^{[l]}}_{128}) = g(a^{[l]}) = \underline{a^{[l]}}$$

256×128

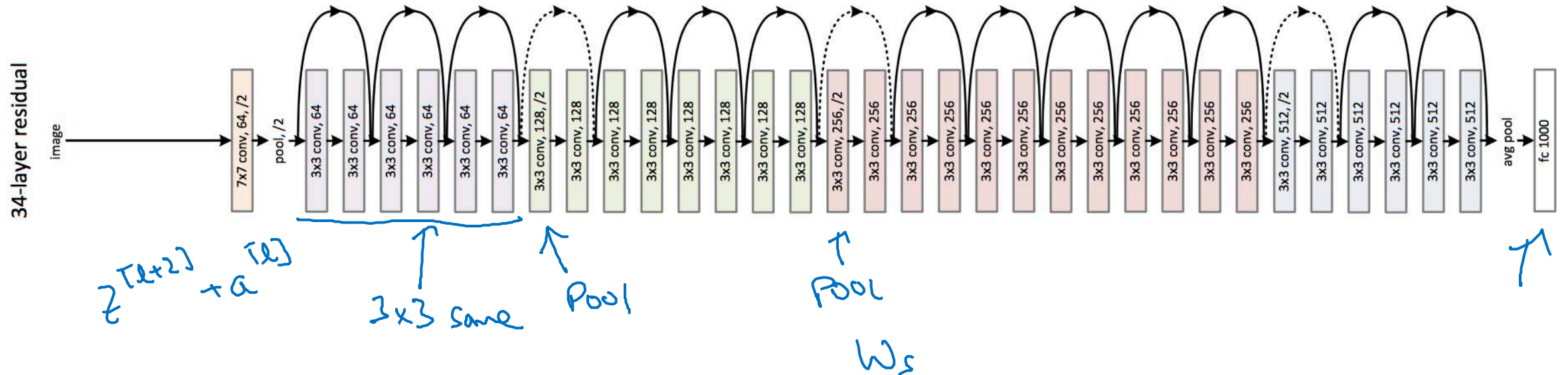
128

ResNet

Plain



ResNet





deeplearning.ai

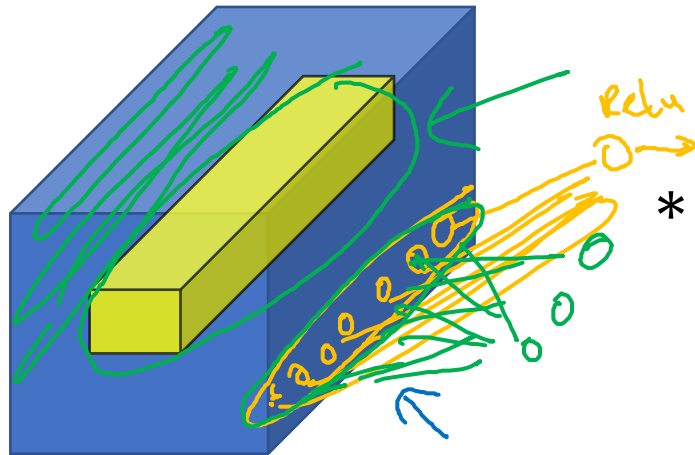
Case Studies

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

$6 \times 6 \times 1$



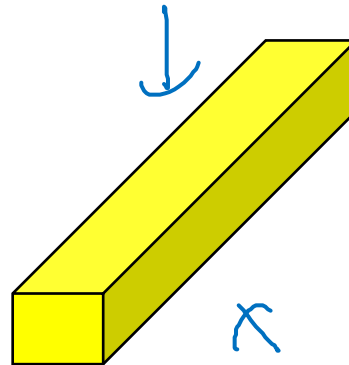
$6 \times 6 \times 32$

*

2

=

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



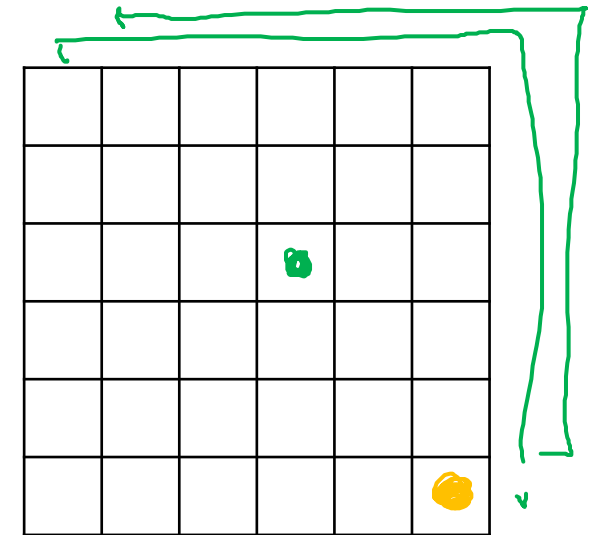
$1 \times 1 \times 32$

=

ReLU

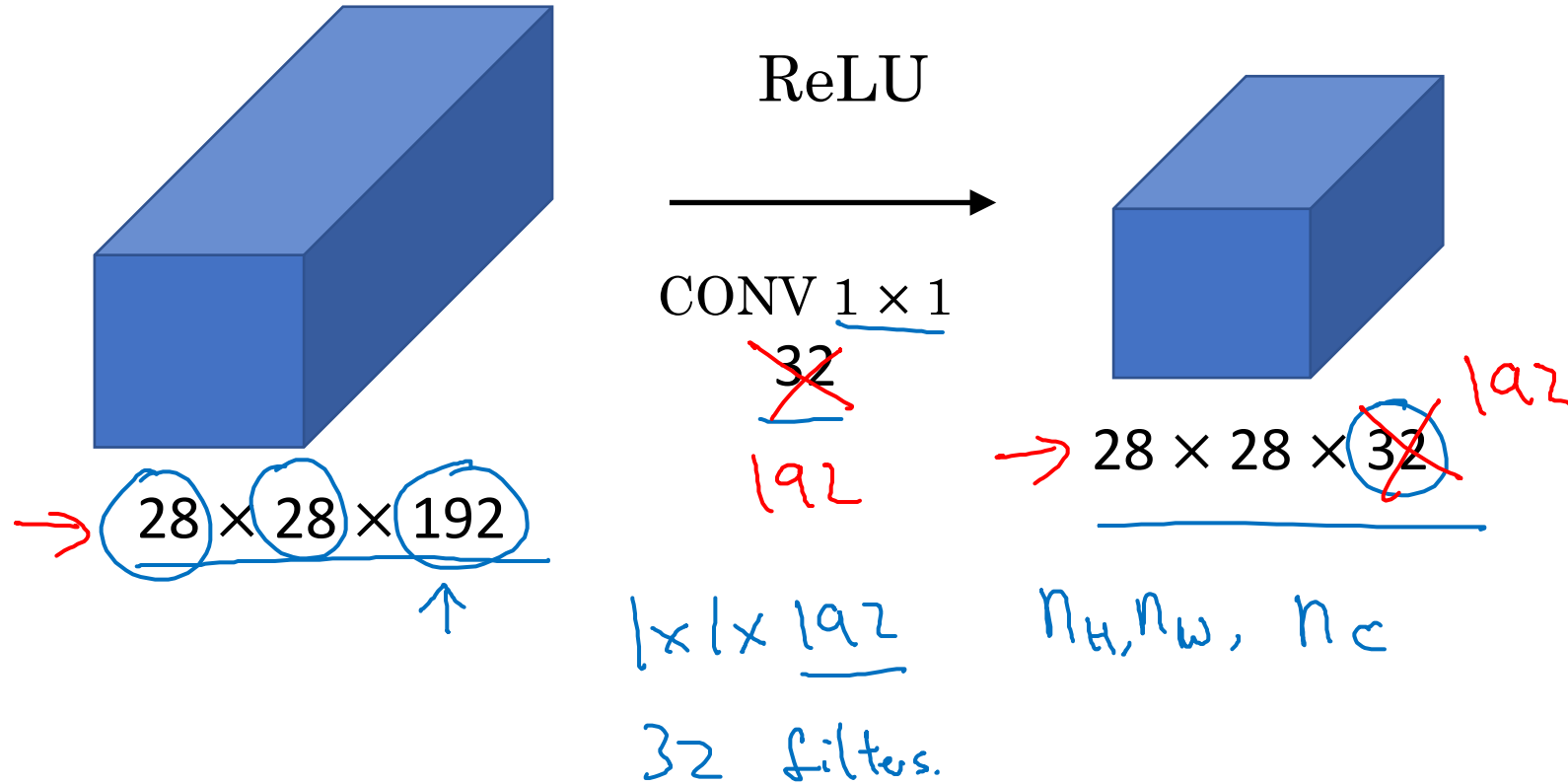
Network in
Network

2	4	6	...		



$6 \times 6 \times \# \text{ filters}$

Using 1×1 convolutions



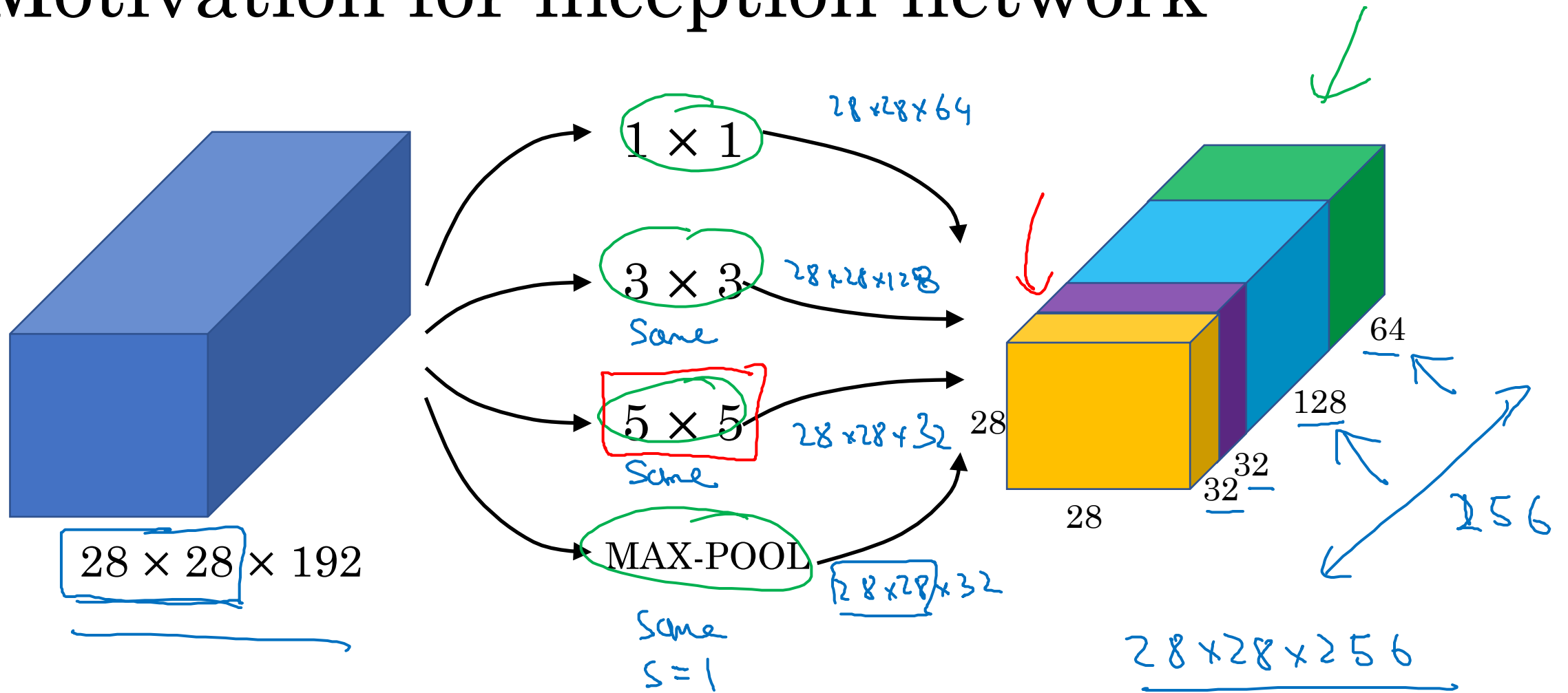


deeplearning.ai

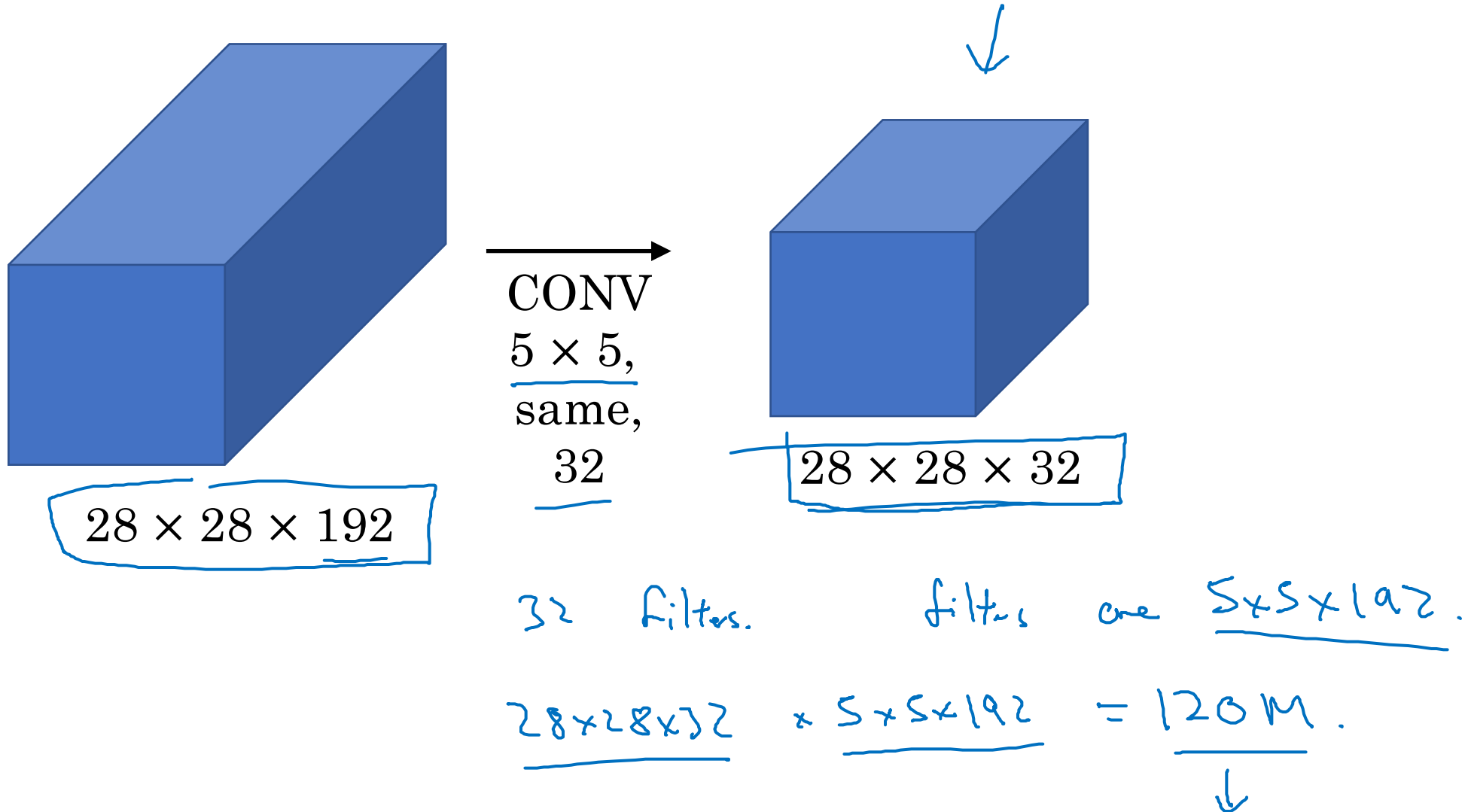
Case Studies

Inception network motivation

Motivation for inception network

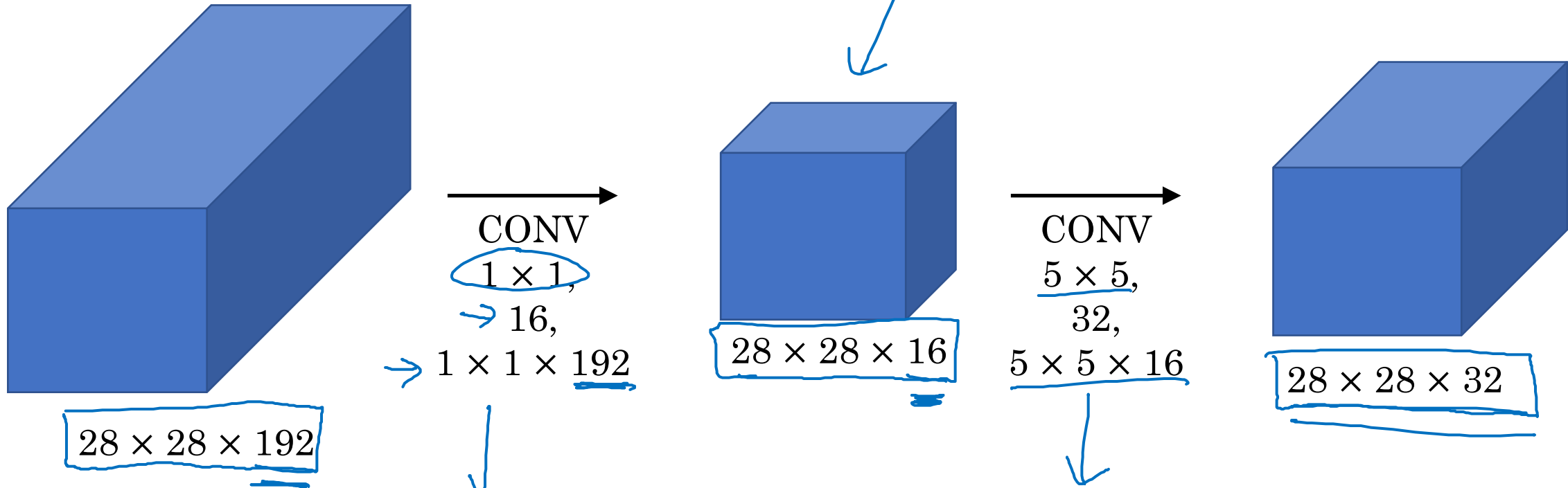
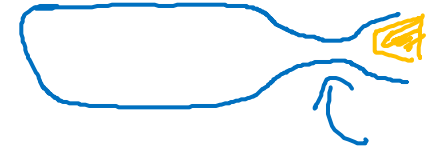


The problem of computational cost



Using 1×1 convolution

"bottleneck layers"



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

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

12.4M

120M

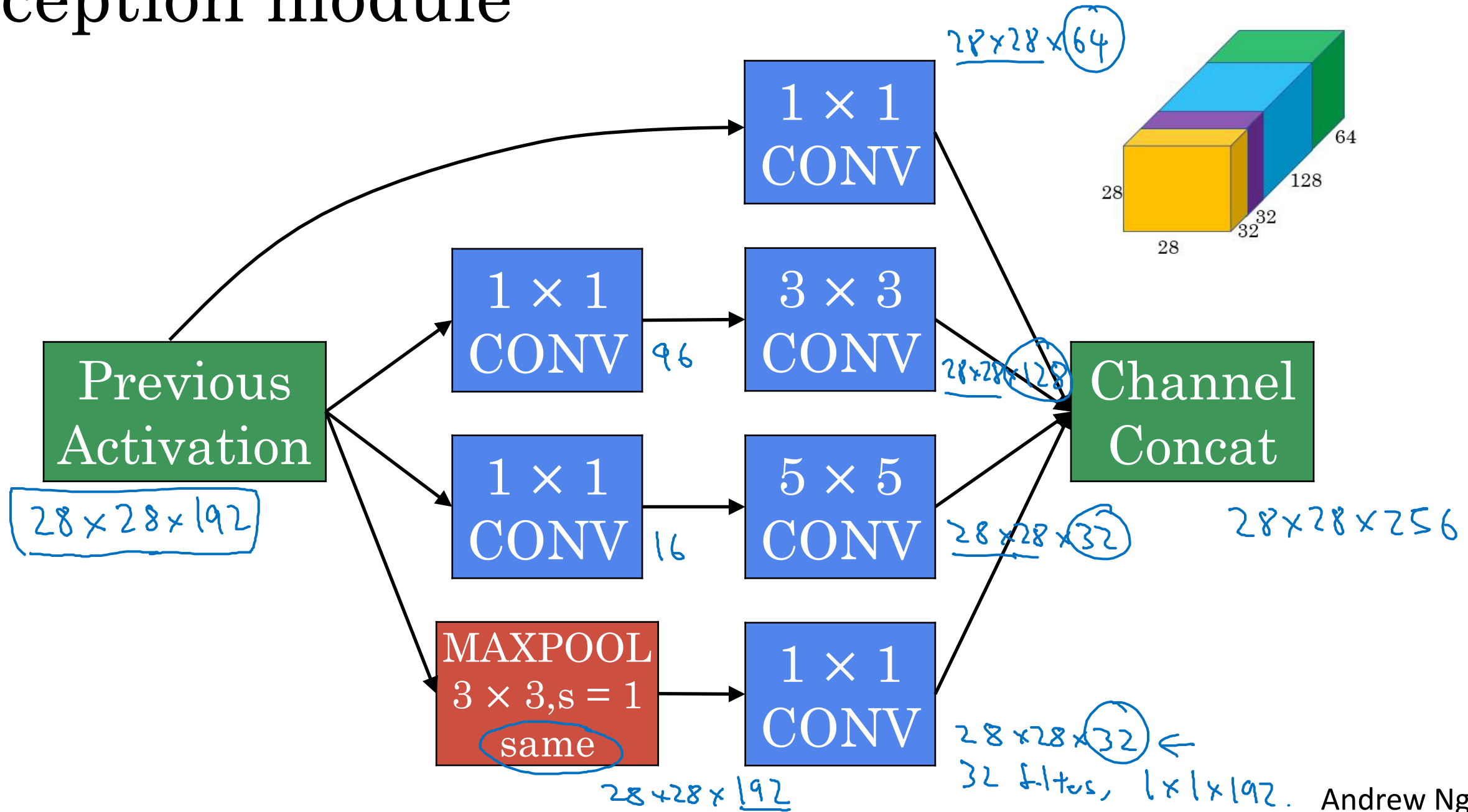


deeplearning.ai

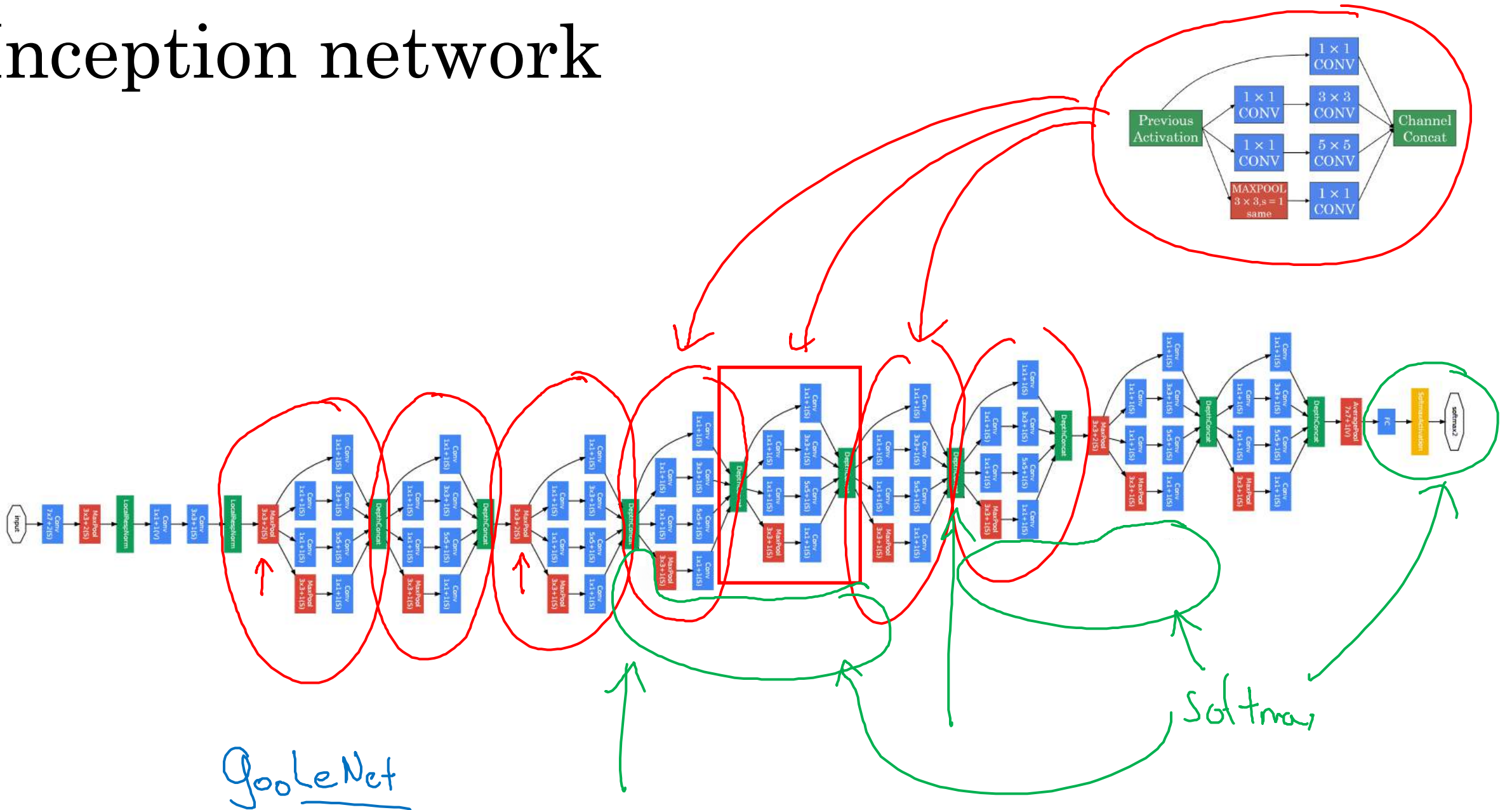
Case Studies

Inception network

Inception module



Inception network







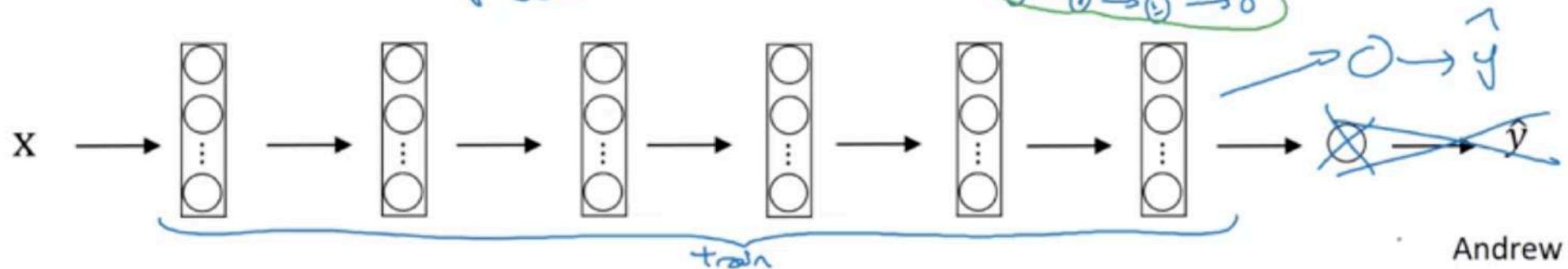
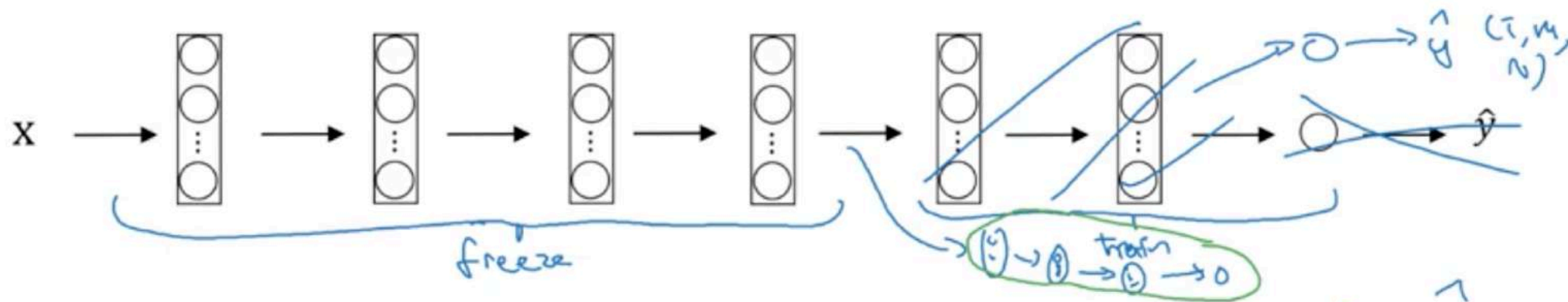
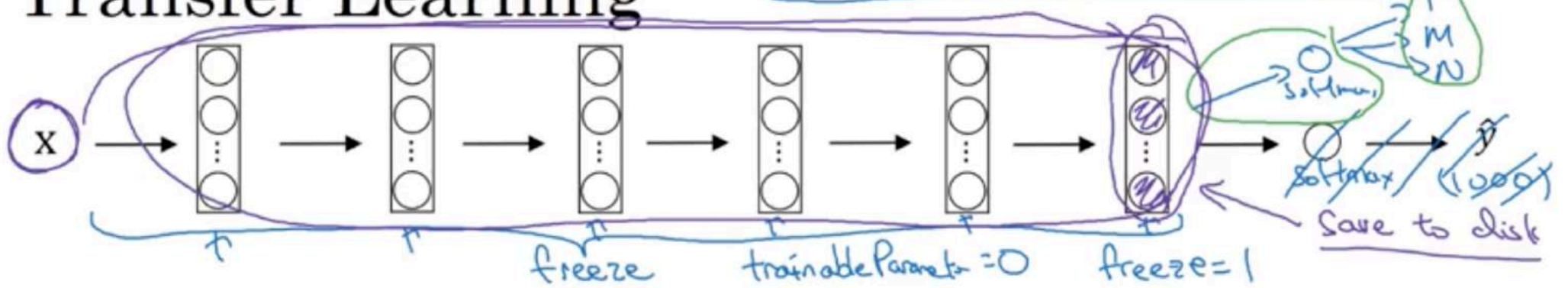
deeplearning.ai

Practical advice for using ConvNets

Transfer Learning

Transfer Learning

 Tigger
 Misty
Neither





deeplearning.ai

Practical advice for using ConvNets

Data augmentation

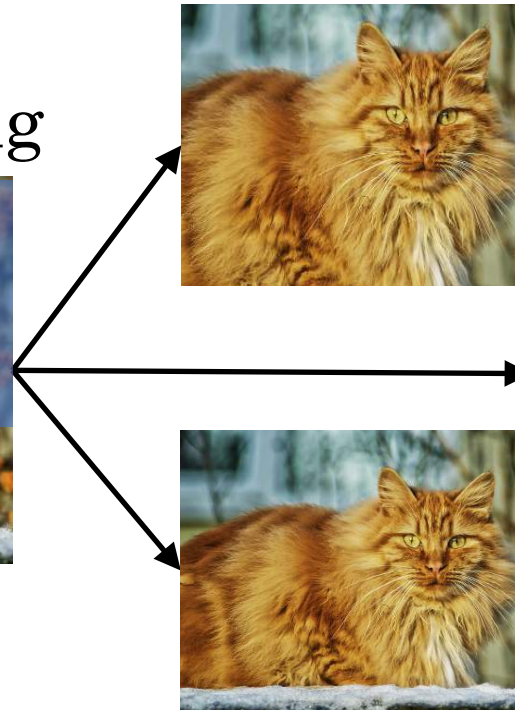
Common augmentation method

Mirroring



y

Random Cropping

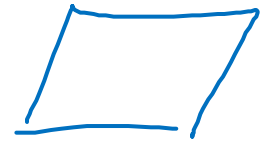


Rotation

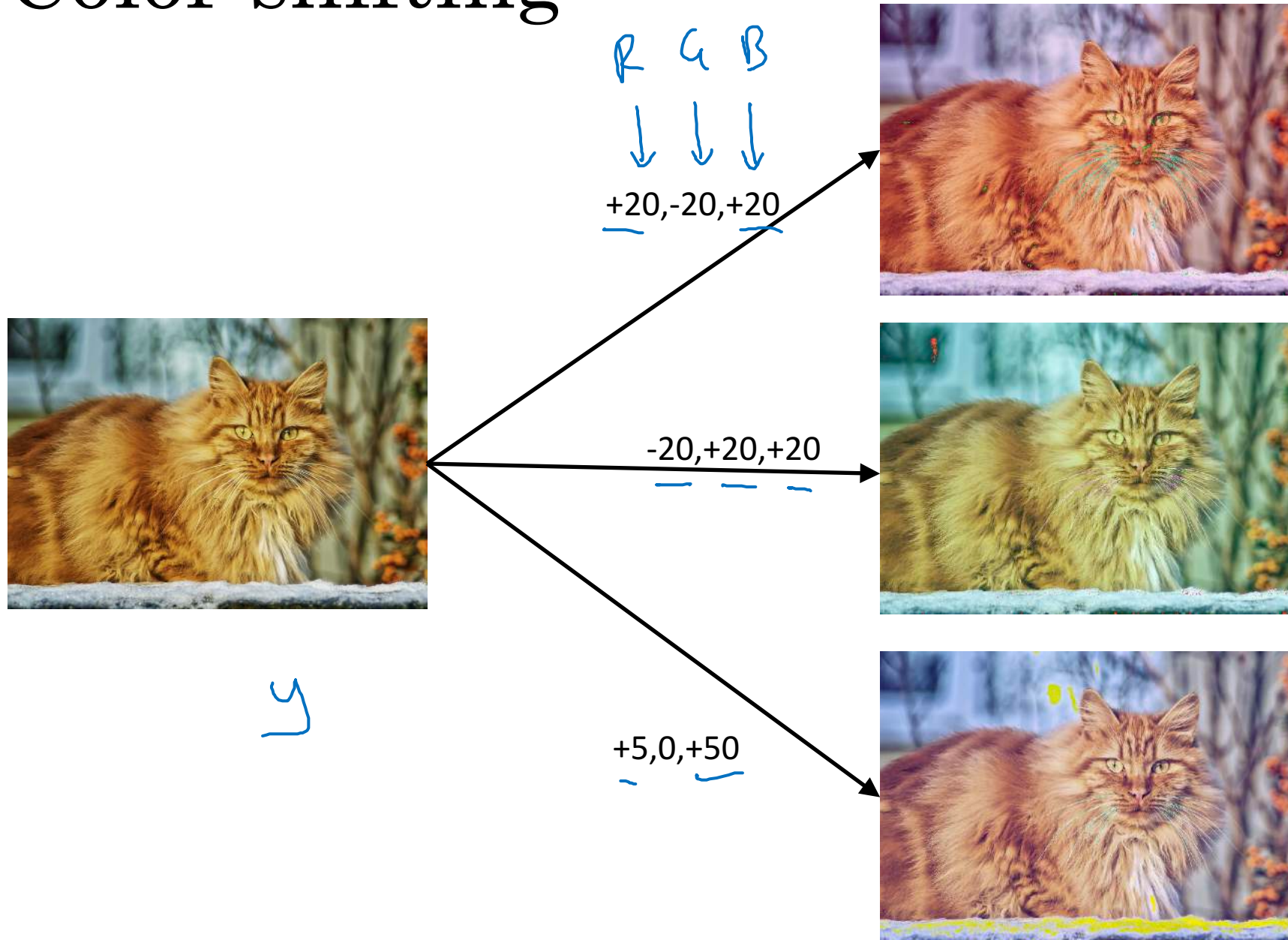
Shearing

Local warping

...



Color shifting



Advanced:

PCA

ml-class.org

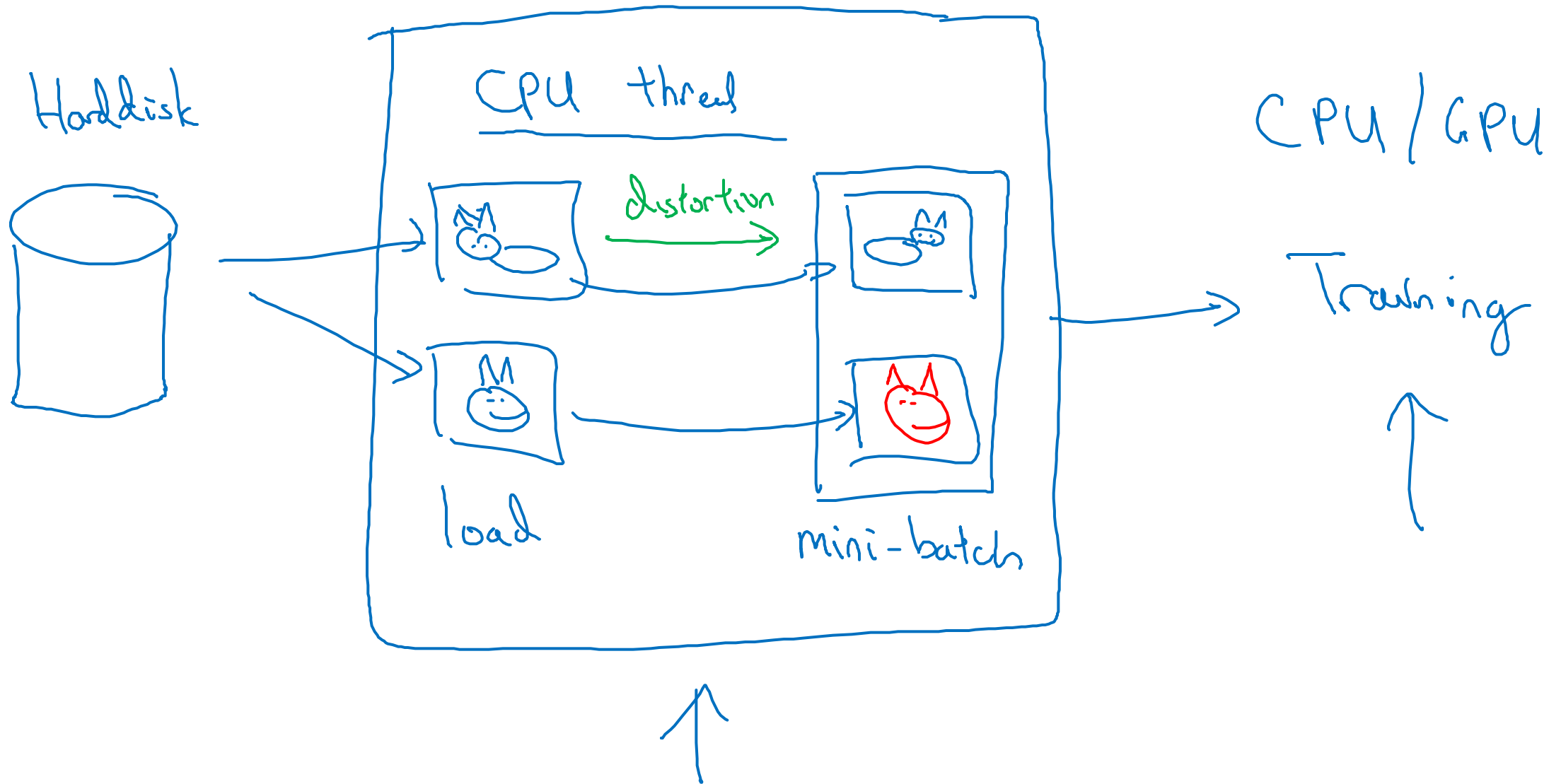
[AlexNet paper

["PCA color augmentation."

R B

G

Implementing distortions during training



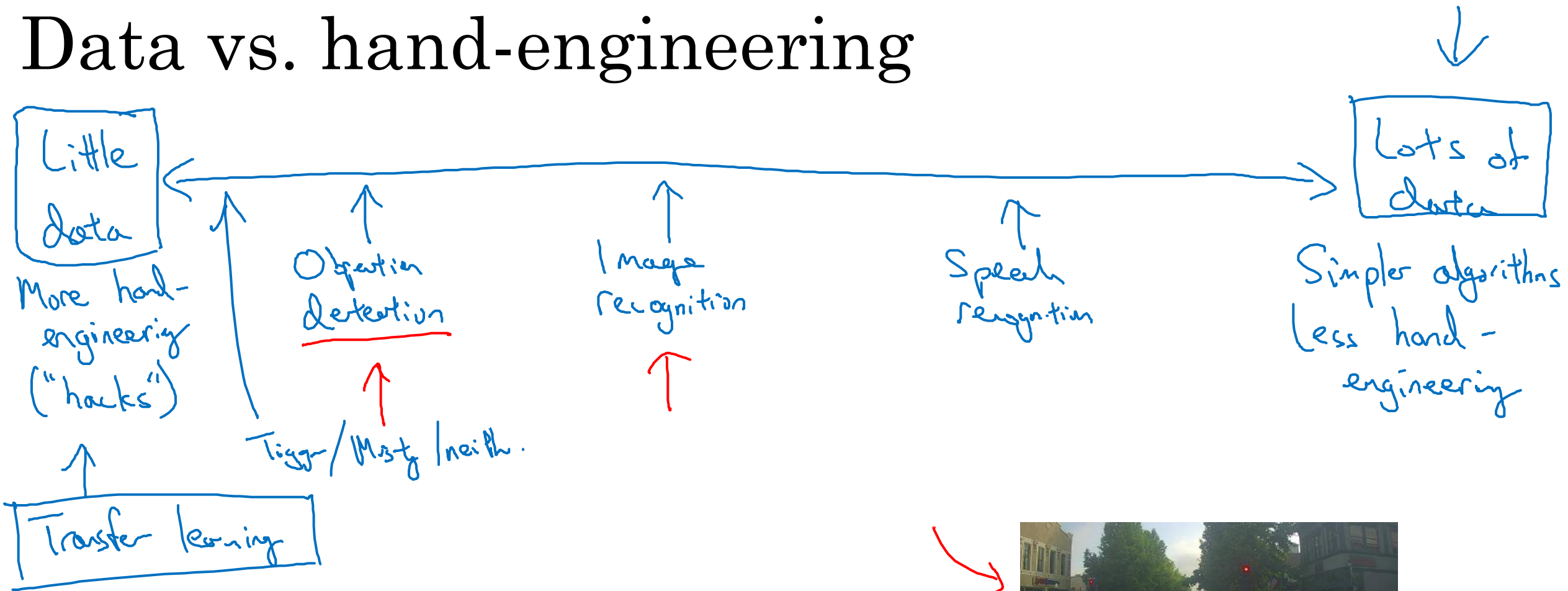


deeplearning.ai

Practical advice for using ConvNets

The state of computer vision

Data vs. hand-engineering



Two sources of knowledge

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



Tips for doing well on benchmarks/winning competitions

3-15 networks

→ \hat{y}

Ensembling

- 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



1

+



4

+



1

+



4

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