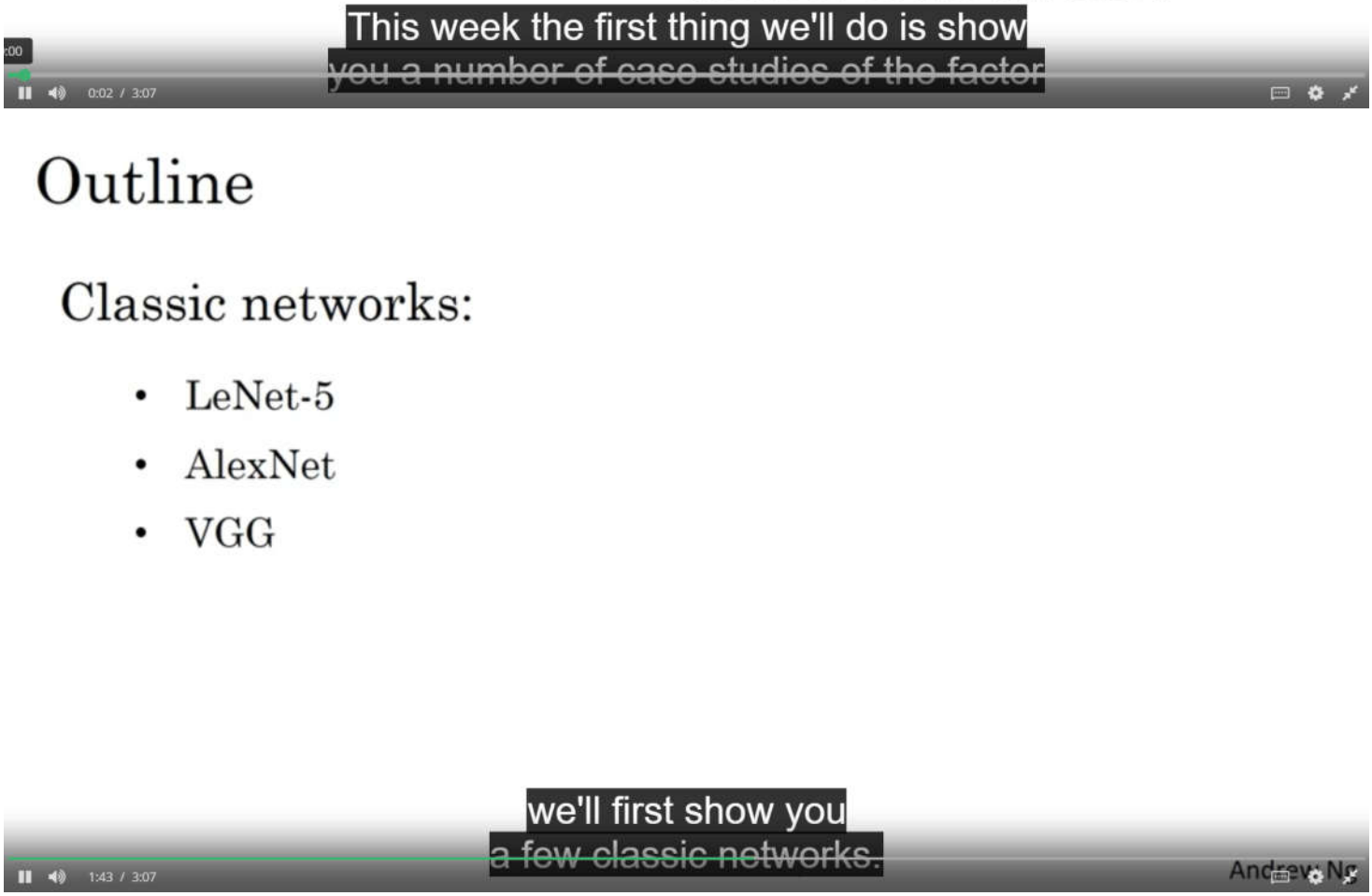
CNN WEEK 2

29 April 2020 09:41



Outline

Classic networks:

- LeNet-5 ←
- AlexNet ←
- VGG ←

ResNet

Then I want to show you the ResNet or conv residual network and

Andrew Ng

Outline

Classic networks:

- LeNet-5 ←
- AlexNet ←
- VGG ←

ResNet (152)

Inception

And then finally you also see a case study of the Inception neural network.

Andrew Ng

Press Esc to exit full screen

deeplearning.ai

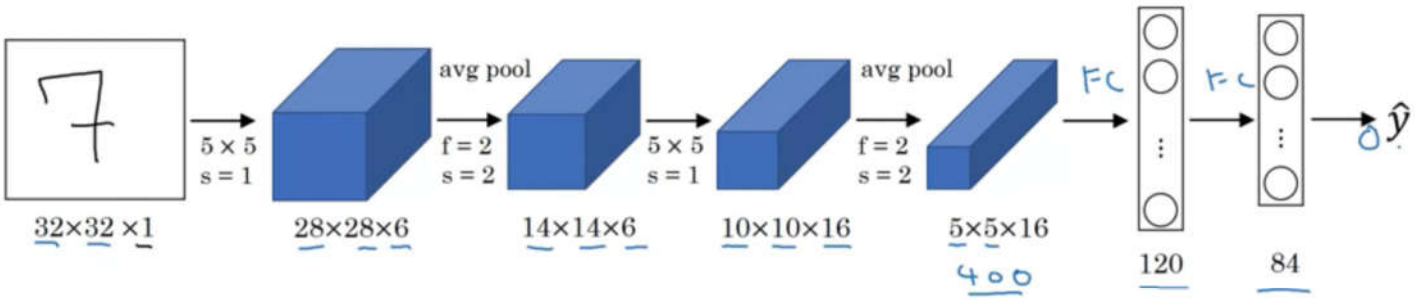
Case Studies

Classic networks

In this video, you'll learn about some of

0:00 / 18:18

LeNet - 5

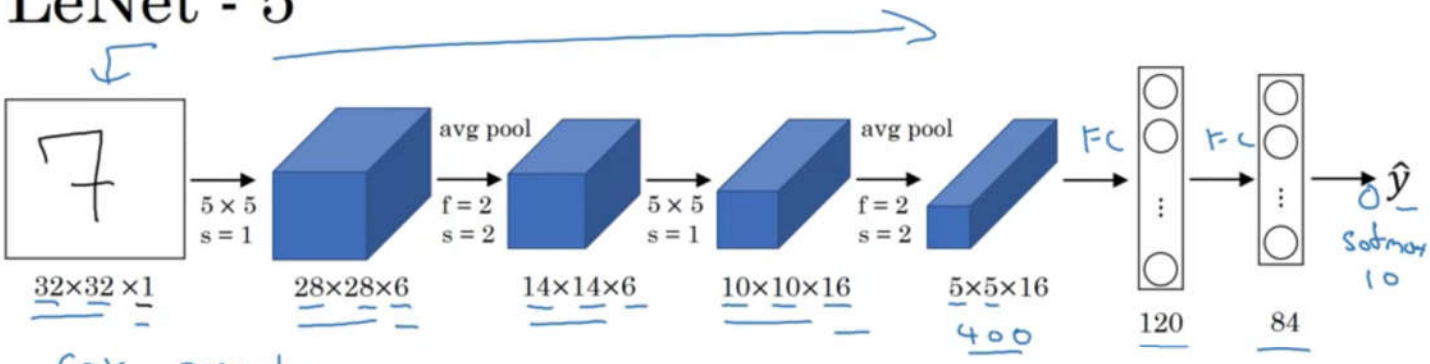


I guess you could draw one more node here to make a prediction for ŷ.

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

Andrew Ng

LeNet - 5



60K parameters.

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

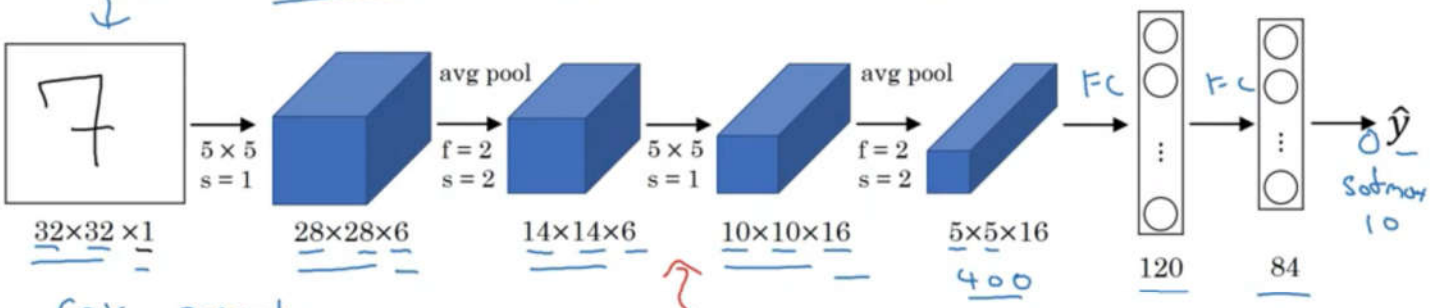
conv pool conv pool fc fc output

So this type of arrangement of layers is quite common.

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

Andrew Ng

LeNet - 5



60K parameters.

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

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

conv pool conv pool fc fc output

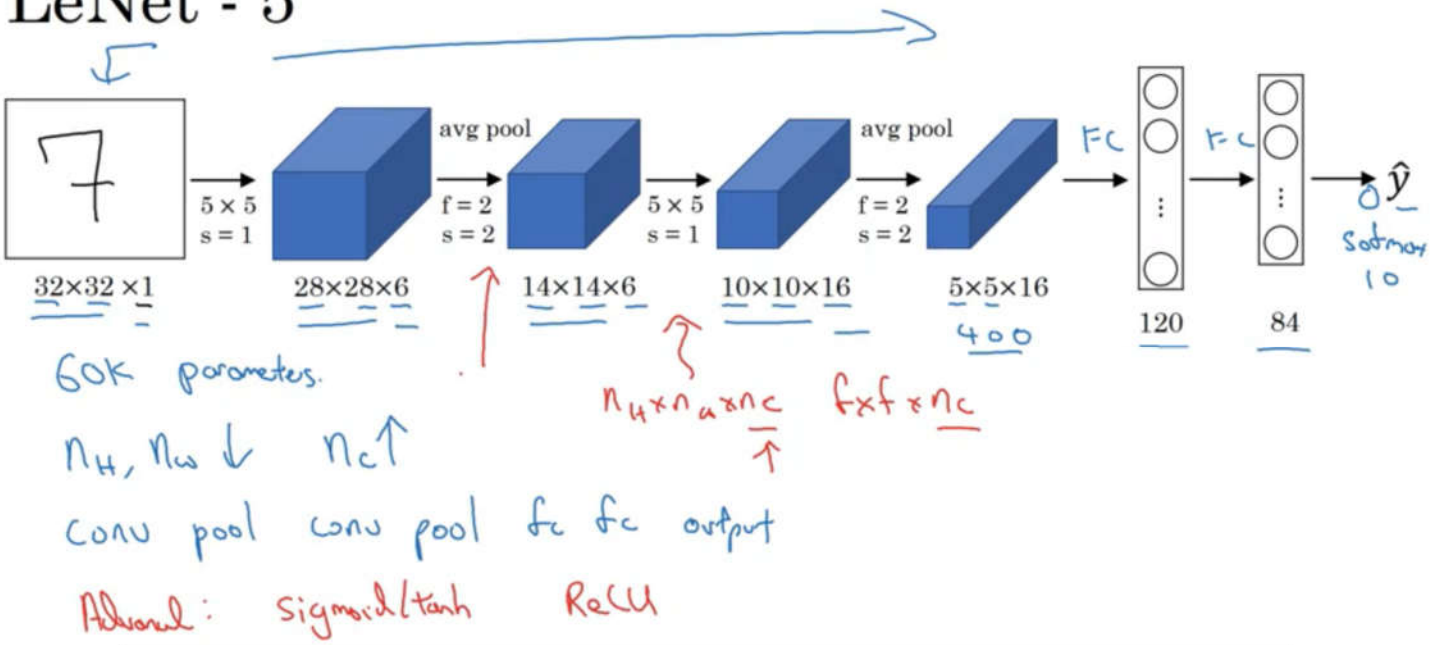
Activation: sigmoid/tanh ReLU

But back then, computers were much slower.

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

Andrew Ng

LeNet - 5



now is that the original LeNet-5 had a non-linearity after pooling,

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

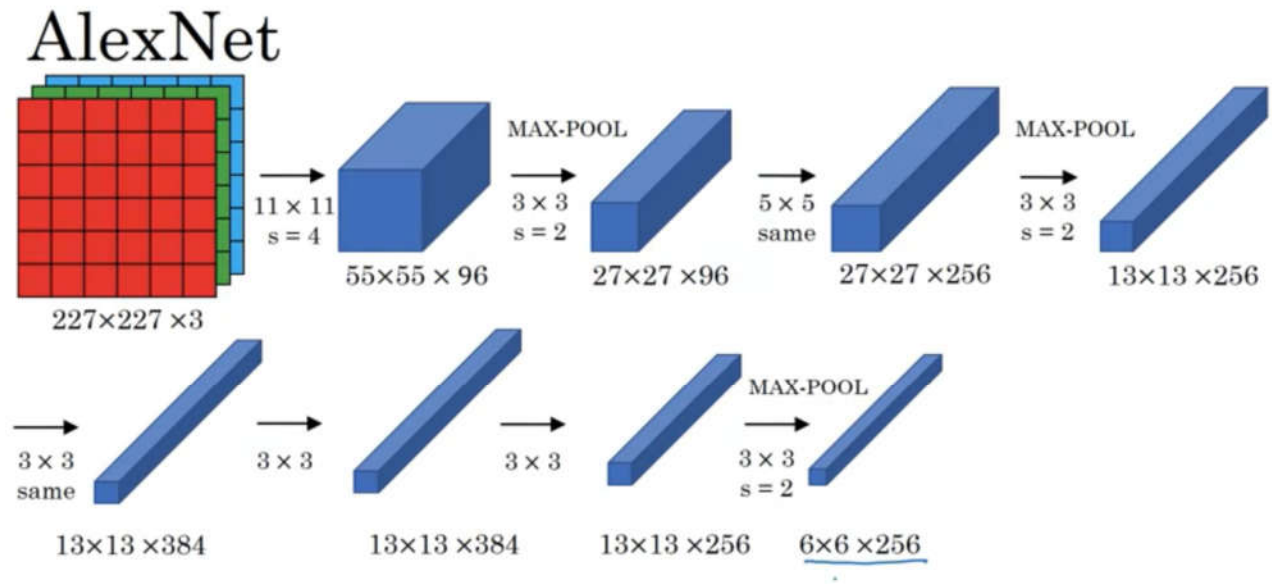
Andrew Ng

AlexNet

The other author's were Ilya Sutskever and Geoffrey Hinton.

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

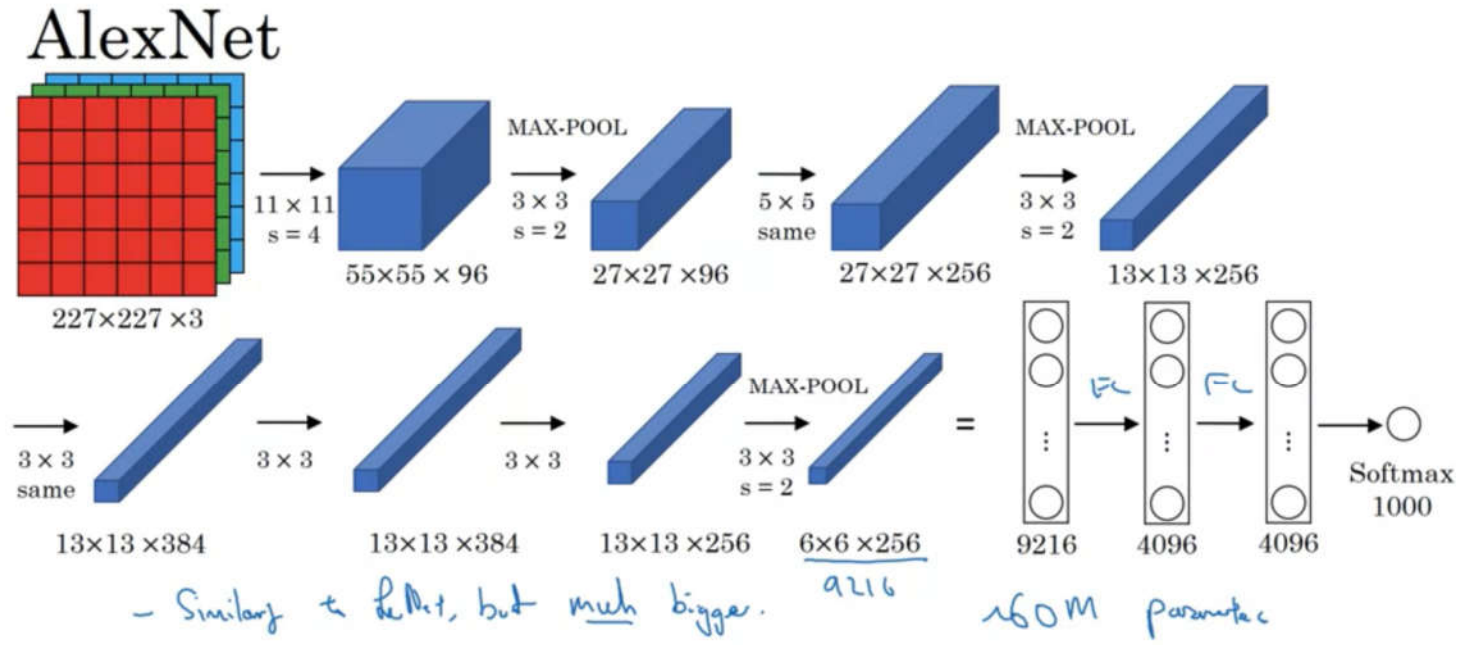
Andrew Ng



If you multiply all these numbers,6 times 6 times 256, that's 9216.

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

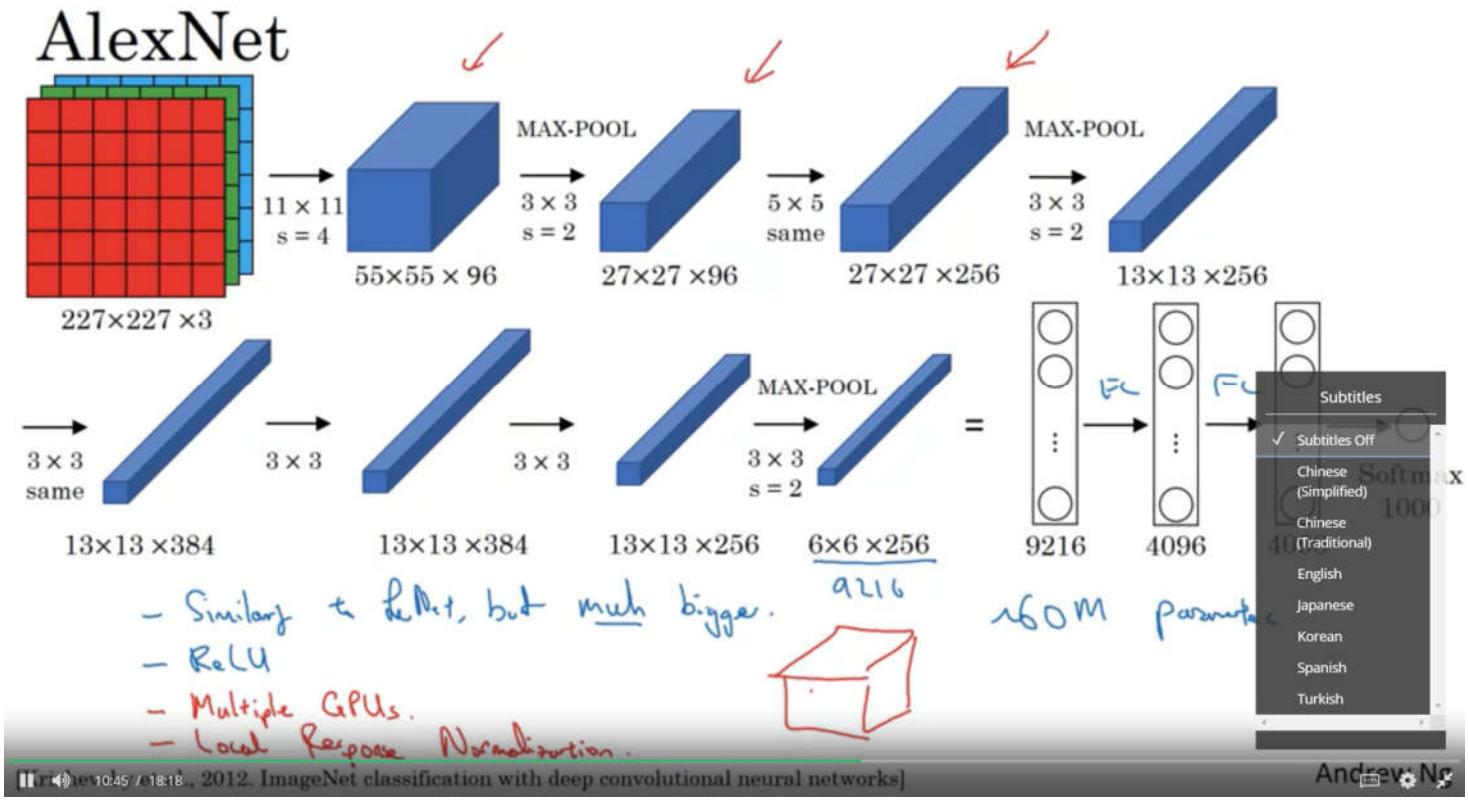
Andrew Ng



allowed it to have a just remarkable performance.

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

Andrew Ng



VGG - 16

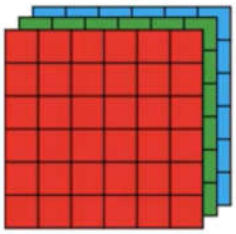
CONV = 3×3 filter, s = 1, same MAX-POOL = 2×2 , s = 2

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

Andrew Ng

VGG - 16

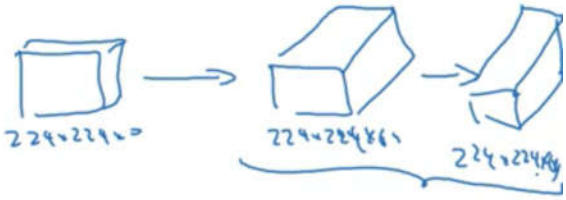
CONV = 3×3 filter, s = 1, same MAX-POOL = 2×2 , s = 2



224×224 ×3

→ 224×224×64

[CONV 64]
×2

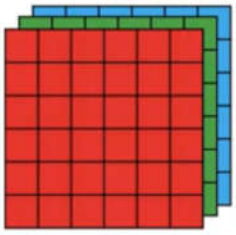


[Simonyan & Zisserman 2015. Very deep convolutional networks for large-scale image recognition]

Andrew Ng

VGG - 16

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



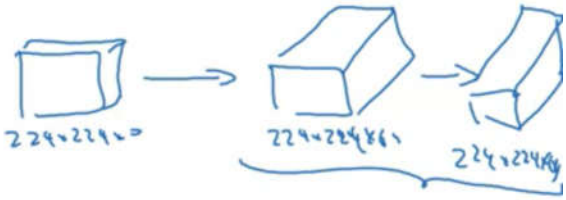
224×224 ×3

→ 224×224×64 → 112×112 ×64 →

[CONV 64]
×2

POOL

[CONV 128]
×2

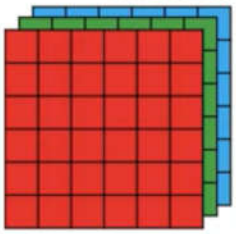


[Simonyan & Zisserman 2015. Very deep convolutional networks for large-scale image recognition]

Andrew Ng

VGG - 16

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

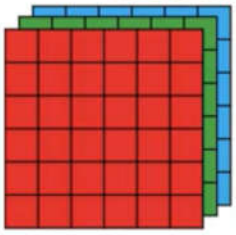
→ [CONV 512] ×3 → 14×14 ×512 → POOL → 7×7×512 → FC 4096 → FC 4096 → Softmax 1000

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

Andrew Ng

VGG - 16

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

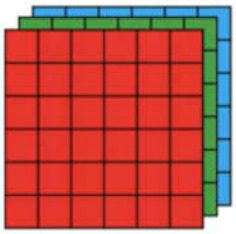
~138M

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

Andrew Ng

VGG - 16

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

~138M

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

Andrew Ng

Press Esc to exit full screen

deeplearning.ai

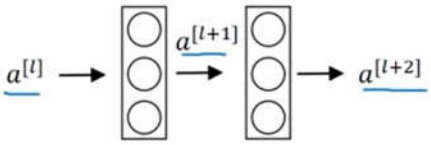
Case Studies

Residual Networks (ResNets)

Very, very deep neural networks are difficult to train because

0:00 / 7:07

Residual block



$a^{[l]} \longrightarrow$ Linear $\longrightarrow$

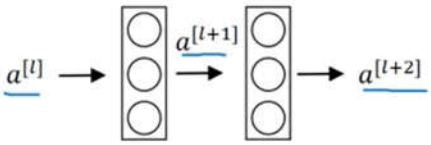
$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]}$$

So you go from a[l] to compute z[l

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

Andrew Ng

Residual block



$a^{[l]} \longrightarrow$ Linear $\longrightarrow$ ReLU $\xrightarrow{a^{[l+1]}}$

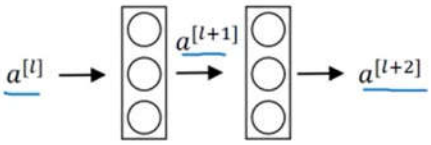
$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \qquad a^{[l+1]} = g(z^{[l+1]})$$

And that's governed by this equation where a[l+1] is g(z[l+1]).

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

Andrew Ng

Residual block



$a^{[l]} \longrightarrow$ Linear $\longrightarrow$ ReLU $\xrightarrow{a^{[l+1]}}$ Linear

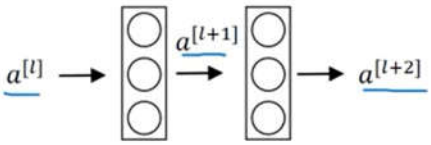
$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \qquad a^{[l+1]} = g(z^{[l+1]}) \qquad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]}$$

So this is quite similar to this equation we saw on the left.

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

Andrew Ng

Residual block



$a^{[l]} \longrightarrow$ Linear $\longrightarrow$ ReLU $\xrightarrow{a^{[l+1]}}$ Linear $\longrightarrow$ ReLU

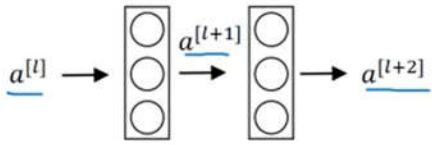
$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \qquad a^{[l+1]} = g(z^{[l+1]}) \qquad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]} \qquad a^{[l+2]} = g(z^{[l+2]})$$

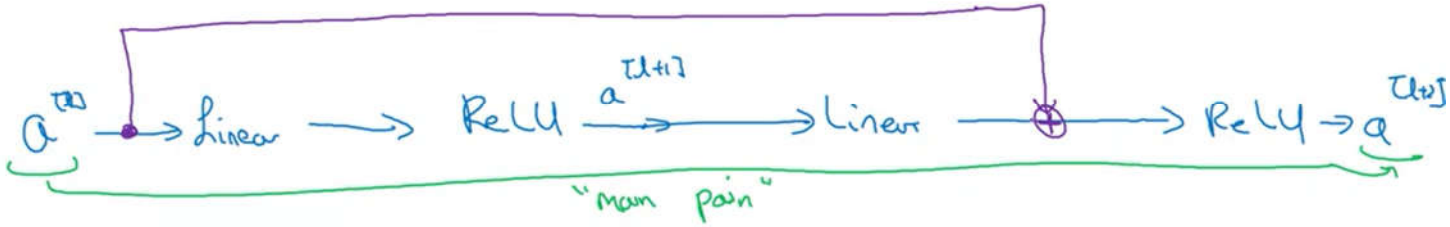
now governed by that equation where G here would be the ReLU nonlinearity.

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

Andrew Ng

Residual block





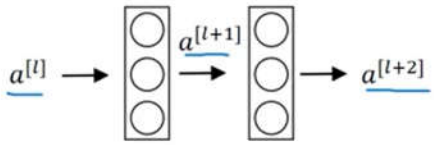
$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \qquad a^{[l+1]} = g(z^{[l+1]}) \qquad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]} \qquad a^{[l+2]} = g(z^{[l+2]})$$

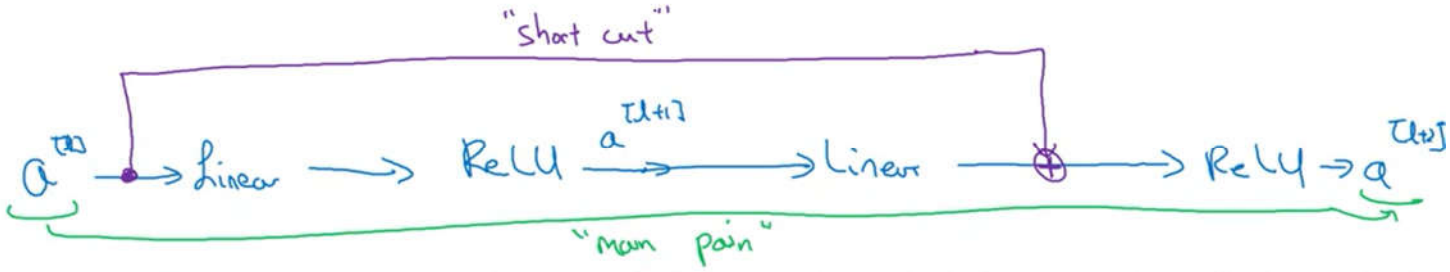
before applying to non-linearity, the ReLU non-linearity.

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

Andrew Ng

Residual block





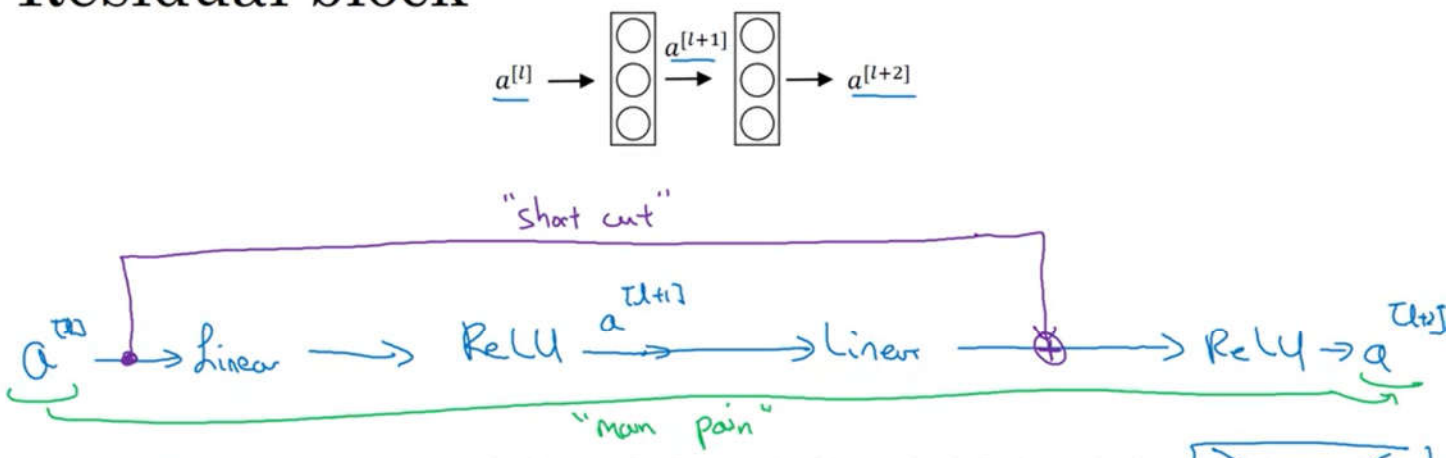
$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]} \qquad a^{[l+1]} = g(z^{[l+1]}) \qquad z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]} \qquad a^{[l+2]} = g(z^{[l+2]})$$

So rather than needing to follow the main path,

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

Andrew Ng

Residual block



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

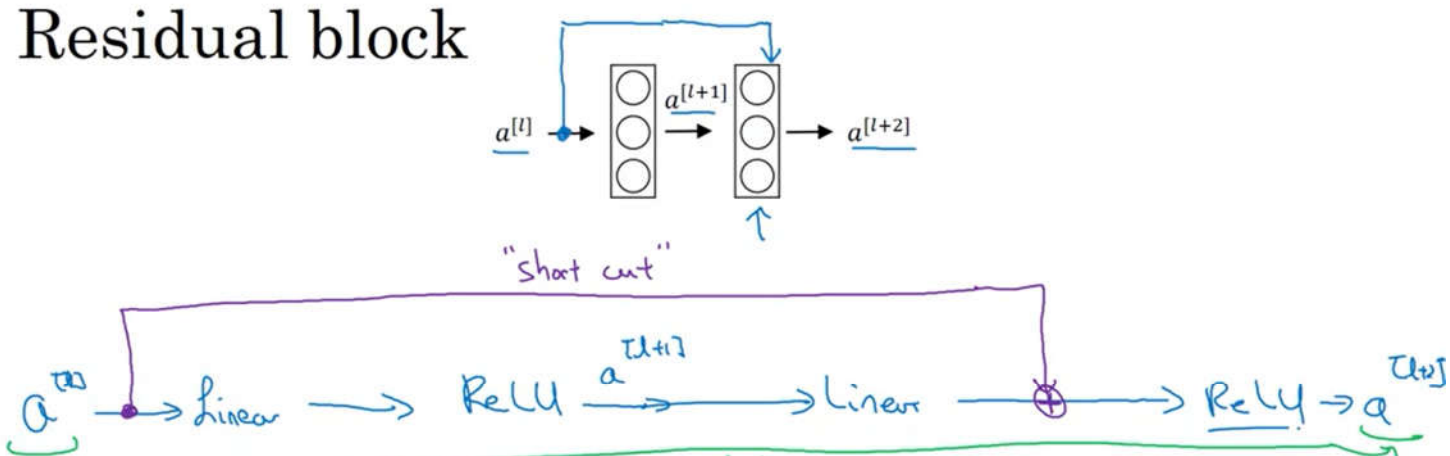
$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

So, the addition of this a[l] here,

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

Andrew Ng

Residual block



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

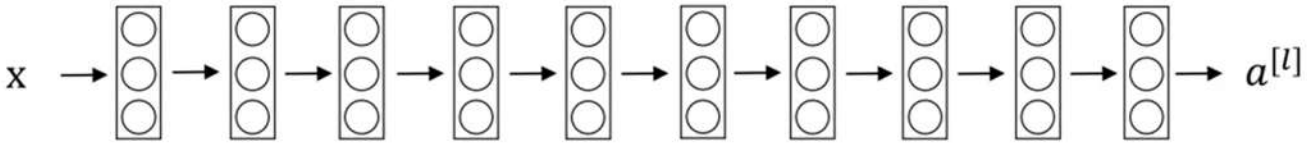
$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

because the short cut is actually added before the ReLU non-linearity.

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

Andrew Ng

Residual Network

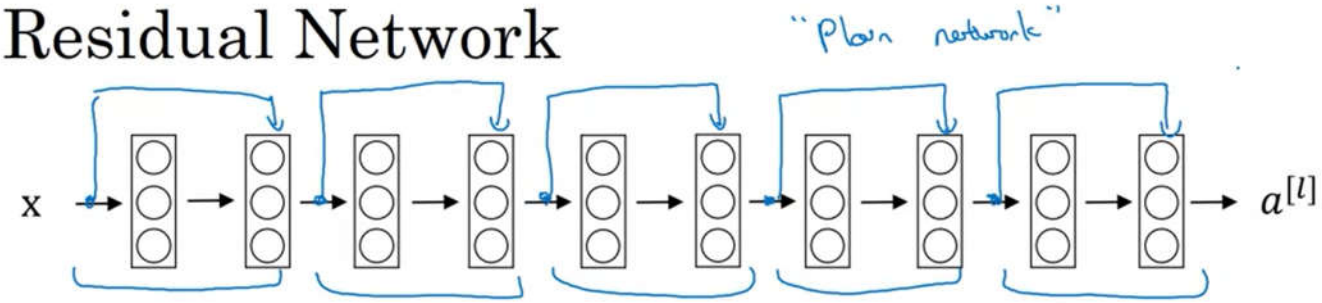


This is not the residual network,

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

Andrew Ng

Residual Network

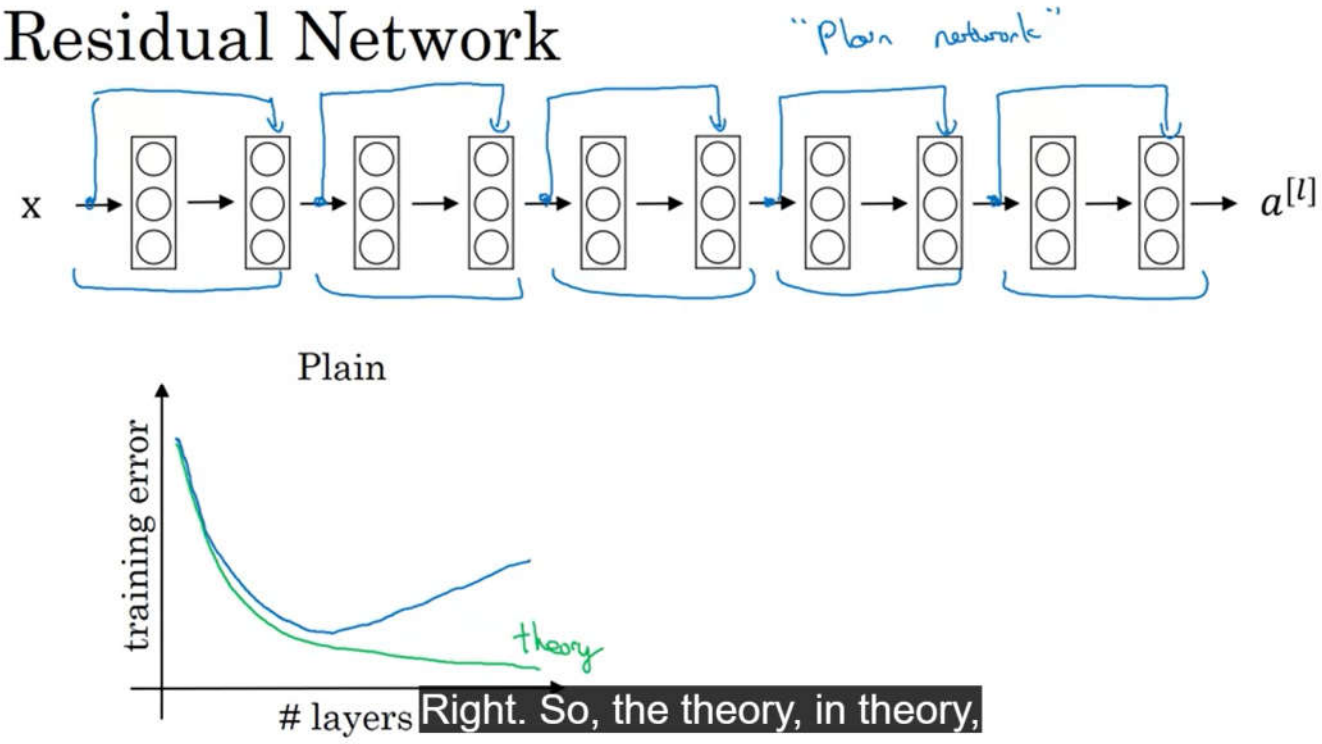


and this is a residual network.

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

Andrew Ng

Residual Network

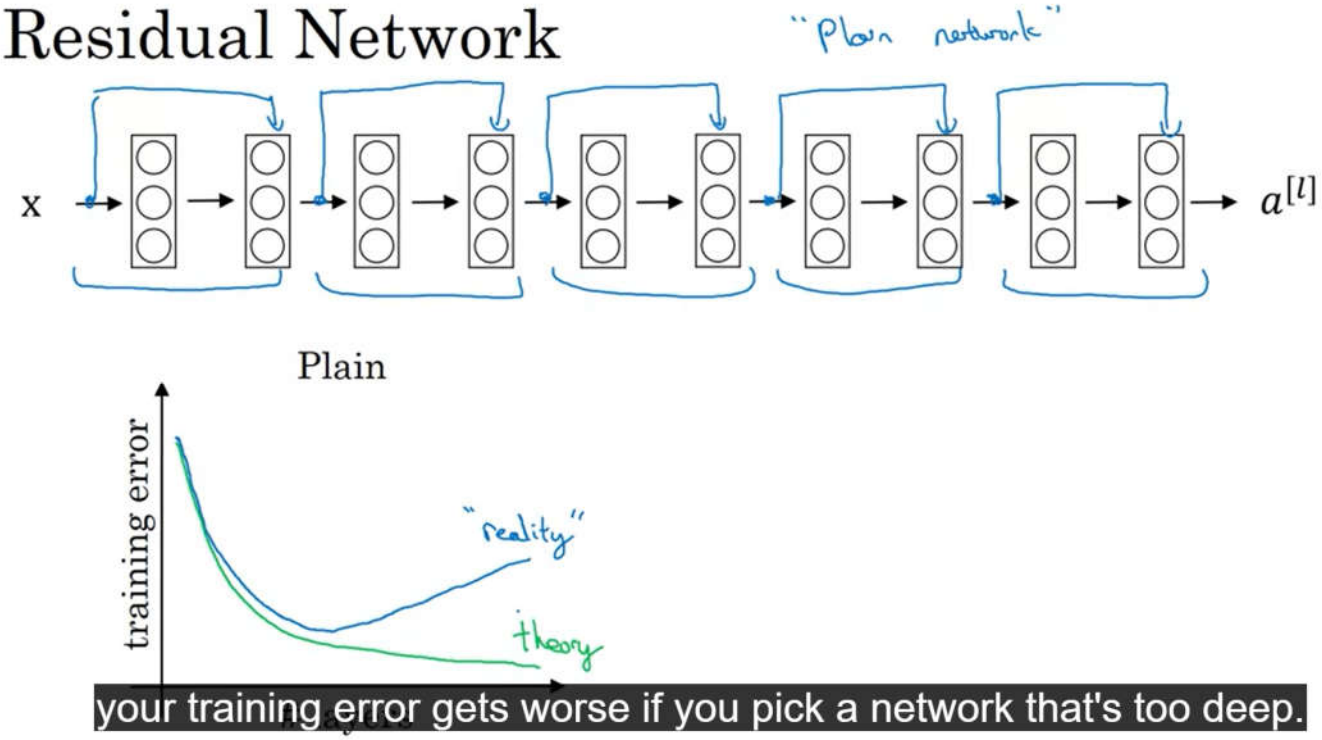


Right. So, the theory, in theory,

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

Andrew Ng

Residual Network

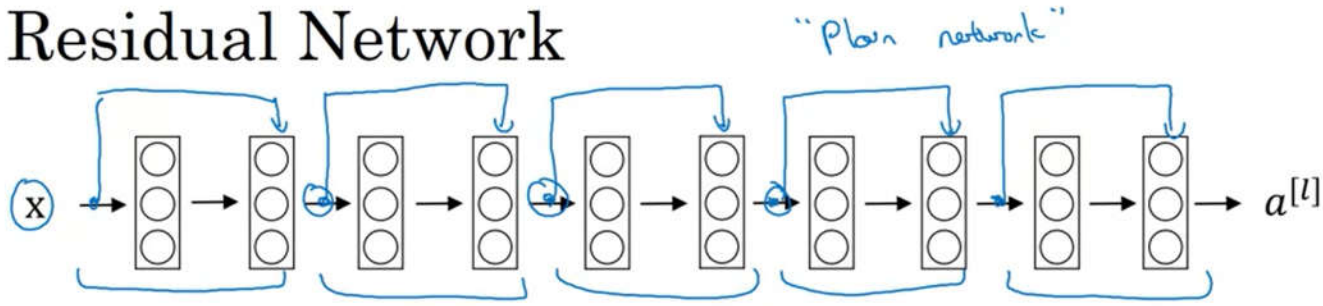


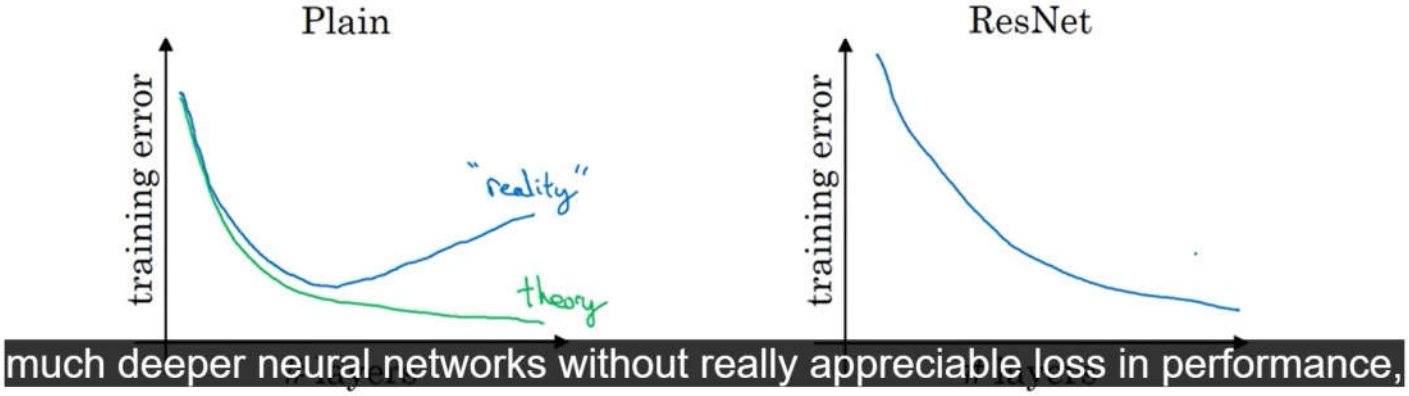
your training error gets worse if you pick a network that's too deep.

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

Andrew Ng

Residual Network





much deeper neural networks without really appreciable loss in performance,

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

Andrew Ng

Press Esc to exit full screen

deeplearning.ai

Case Studies

Why ResNets work

So, why do ResNets work so well?

0:00 / 9:12

Why do residual networks work?

What we saw on the last video was that if you make a network deeper,

Andrew Ng

Why do residual networks work?

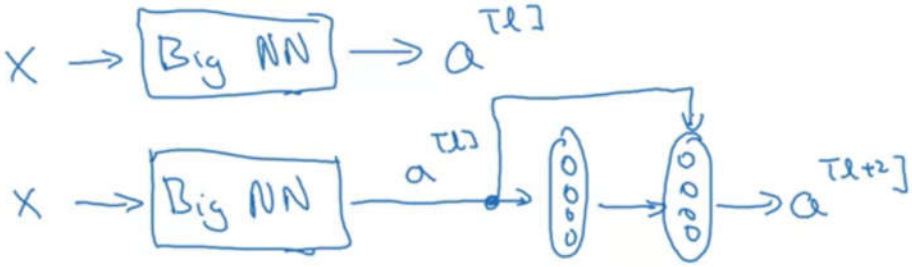
$$x \rightarrow \boxed{\text{Big NN}} \rightarrow a^{[l]}$$

$$x \rightarrow$$

the neural network to make it a little bit deeper.

Andrew Ng

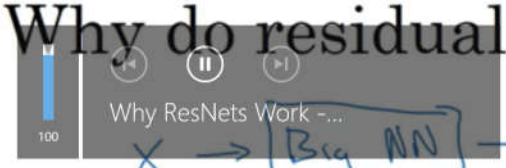
Why do residual networks work?

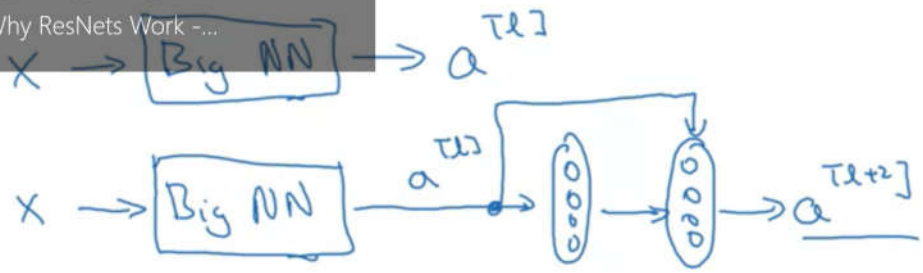


a residual block with that extra short cut.

Andrew Ng

Why do residual networks work?





ReLU. $a \geq 0$

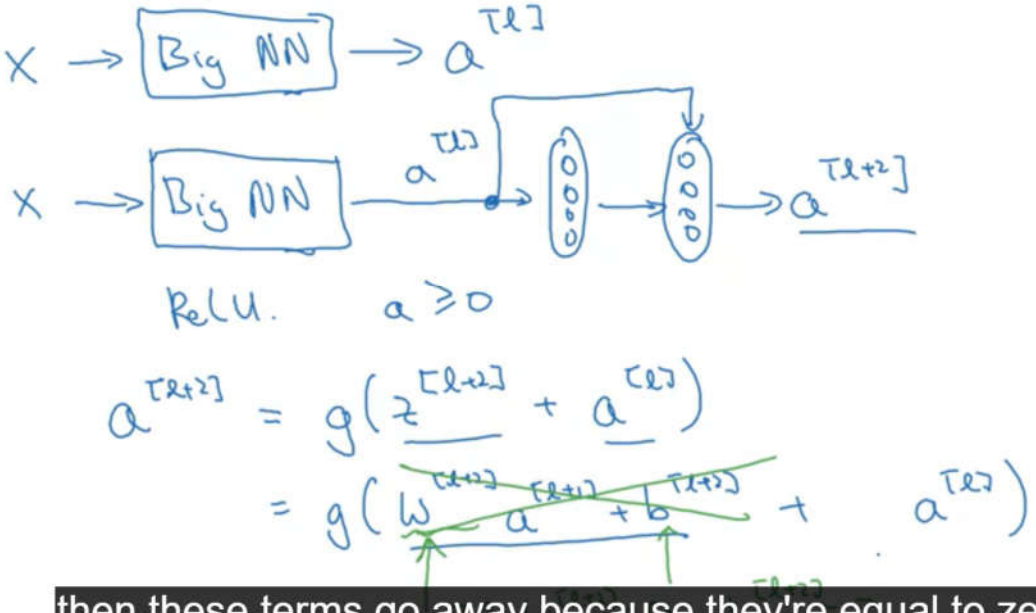
$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

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

but W is really the key term to pay attention to here.

Andrew Ng

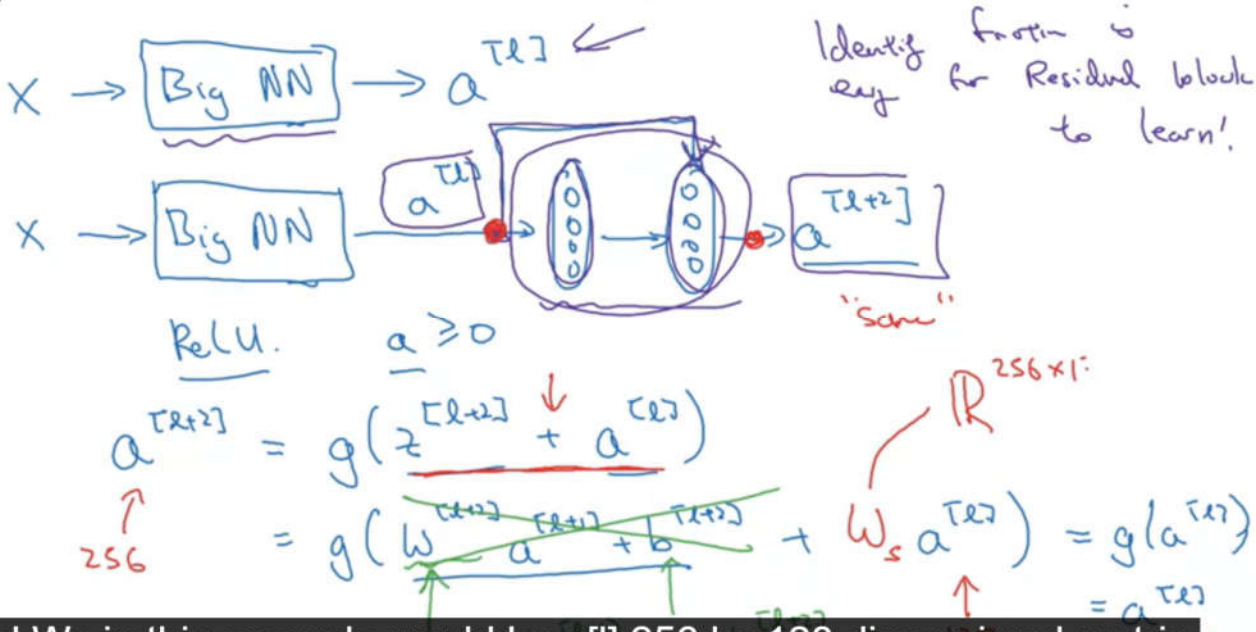
Why do residual networks work?



then these terms go away because they're equal to zero,

Andrew Ng

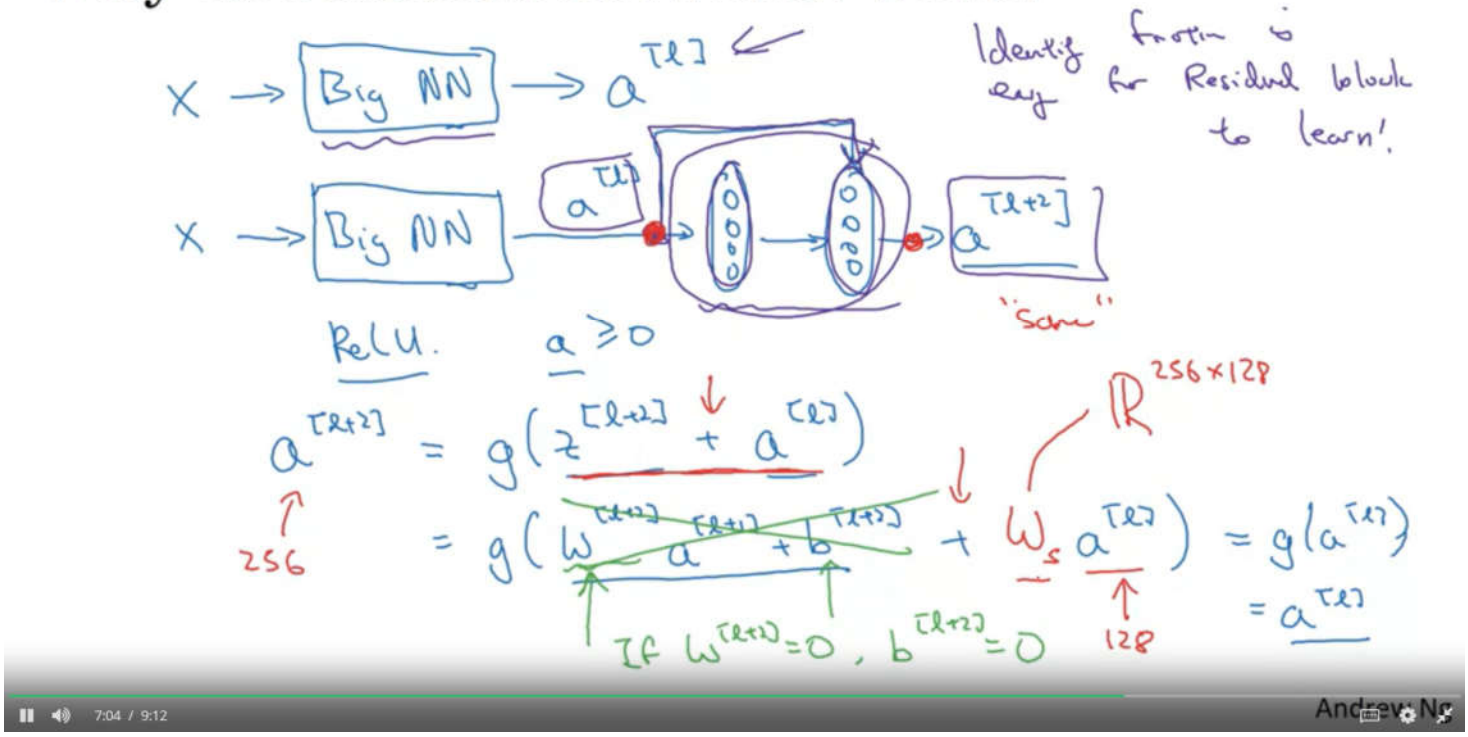
Why do residual networks work?



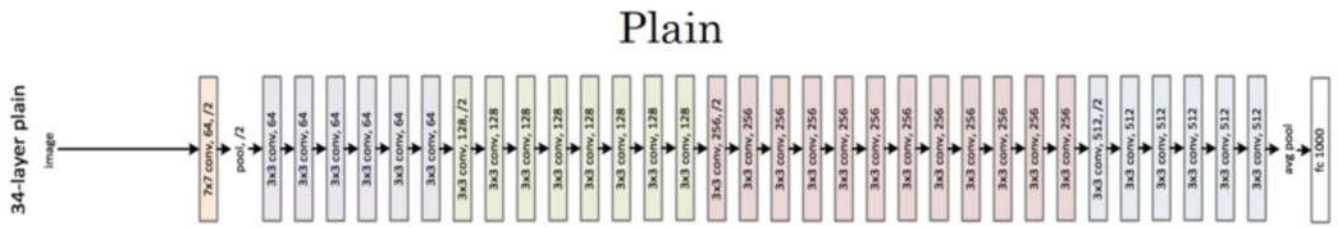
and Ws in this example would be a[l] 256 by 128 dimensional matrix.

Andrew Ng

Why do residual networks work?



ResNet

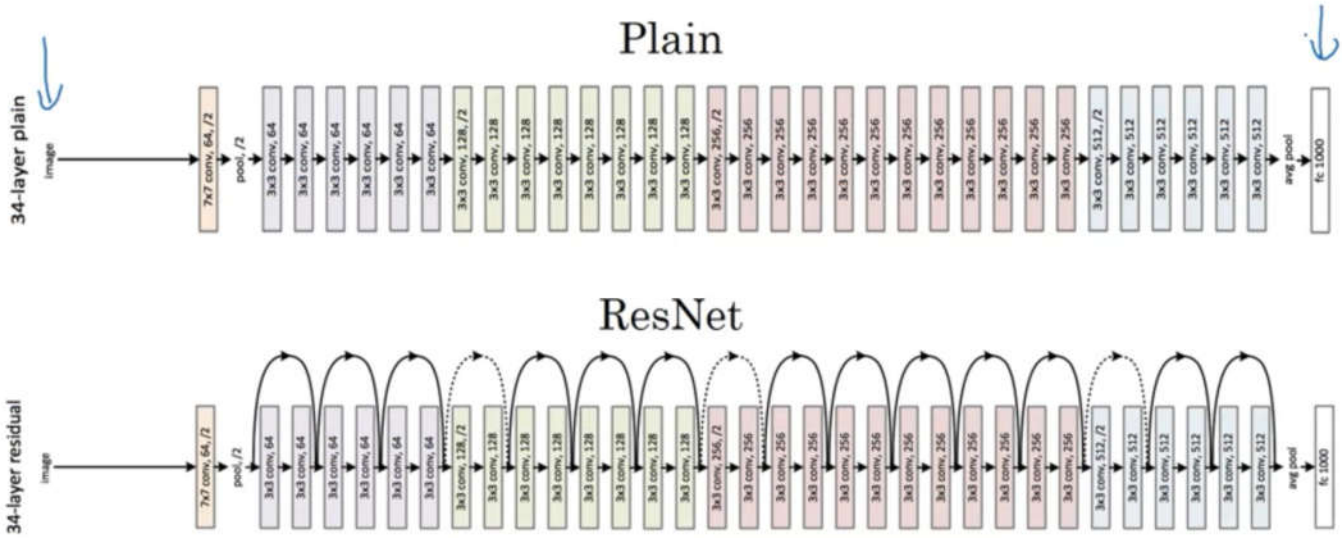


This is an example of a plain network and in which you input an image

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

Andrew Ng

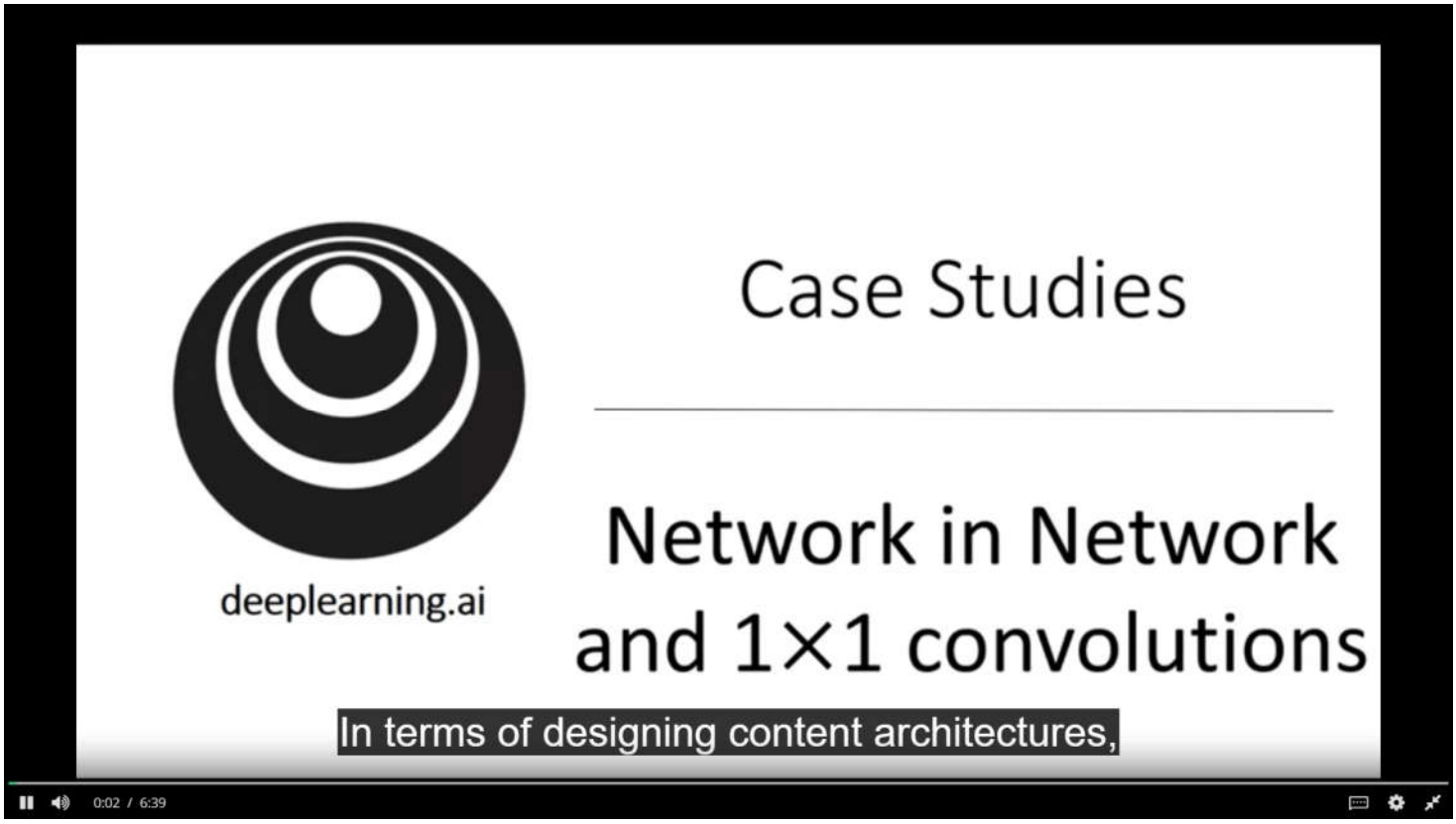
ResNet

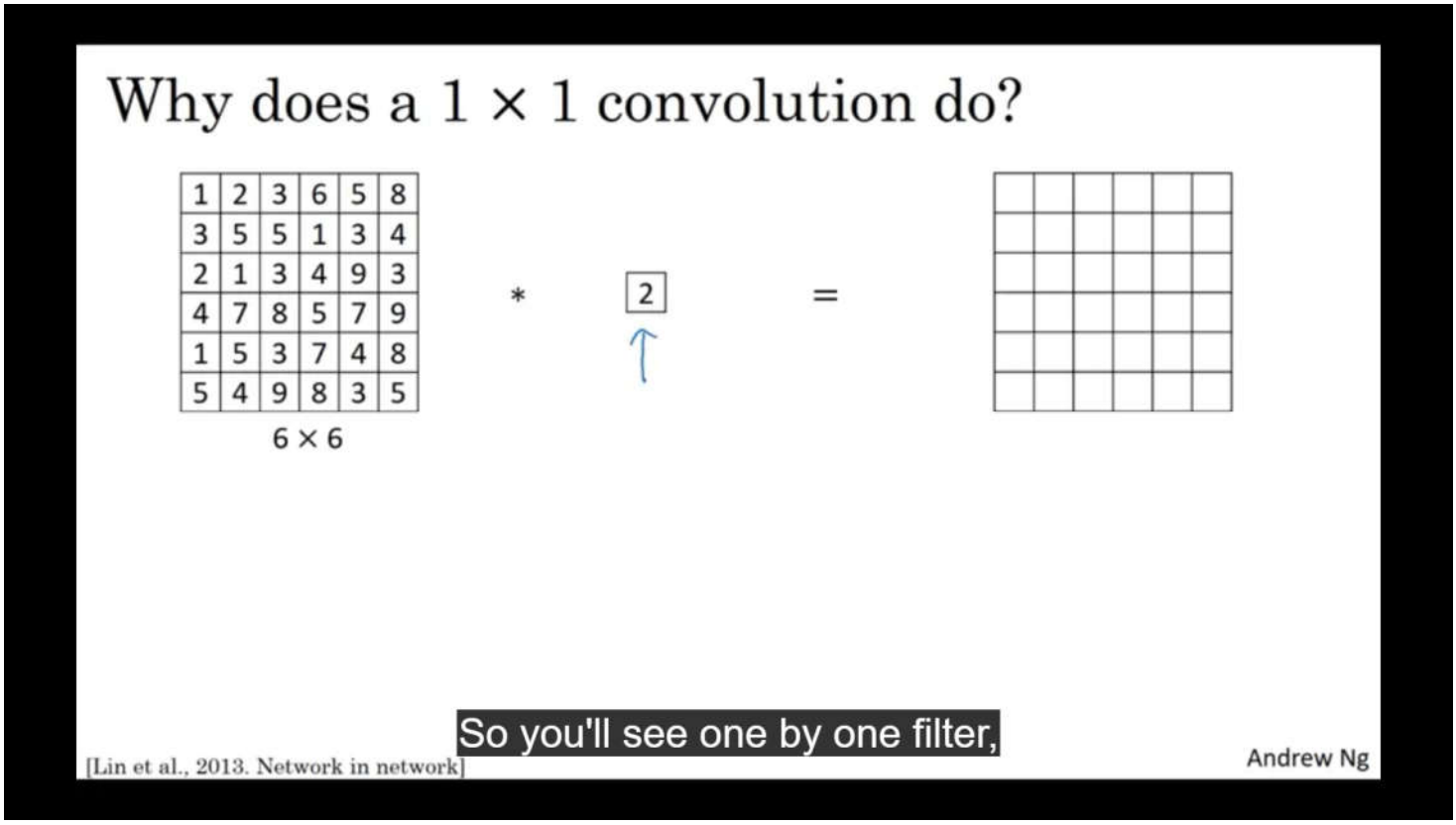


you add those extra skip connections.

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

Andrew Ng


Case Studies
deeplearning.ai
Network in Network
and 1×1 convolutions
In terms of designing content architectures,
0:02 / 6:39


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
*
2
=
So you'll see one by one filter,
[Lin et al., 2013. Network in network]
Andrew Ng

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

* $\boxed{2}$ =

2	4	6	...		

And so, a convolution by a one by one filter,

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

Andrew Ng

Why does a 1 × 1 convolution do?

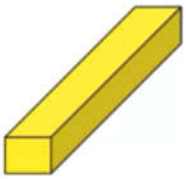
1	2	3	6	5	8
3	5	5	1	3	4
2	1	3	4	9	3
4	7	8	5	7	9
1	5	3	7	4	8
5	4	9	8	3	5

6 × 6 × 1

* $\boxed{2}$ =

2	4	6	...		

6 × 6 × 32

*  =

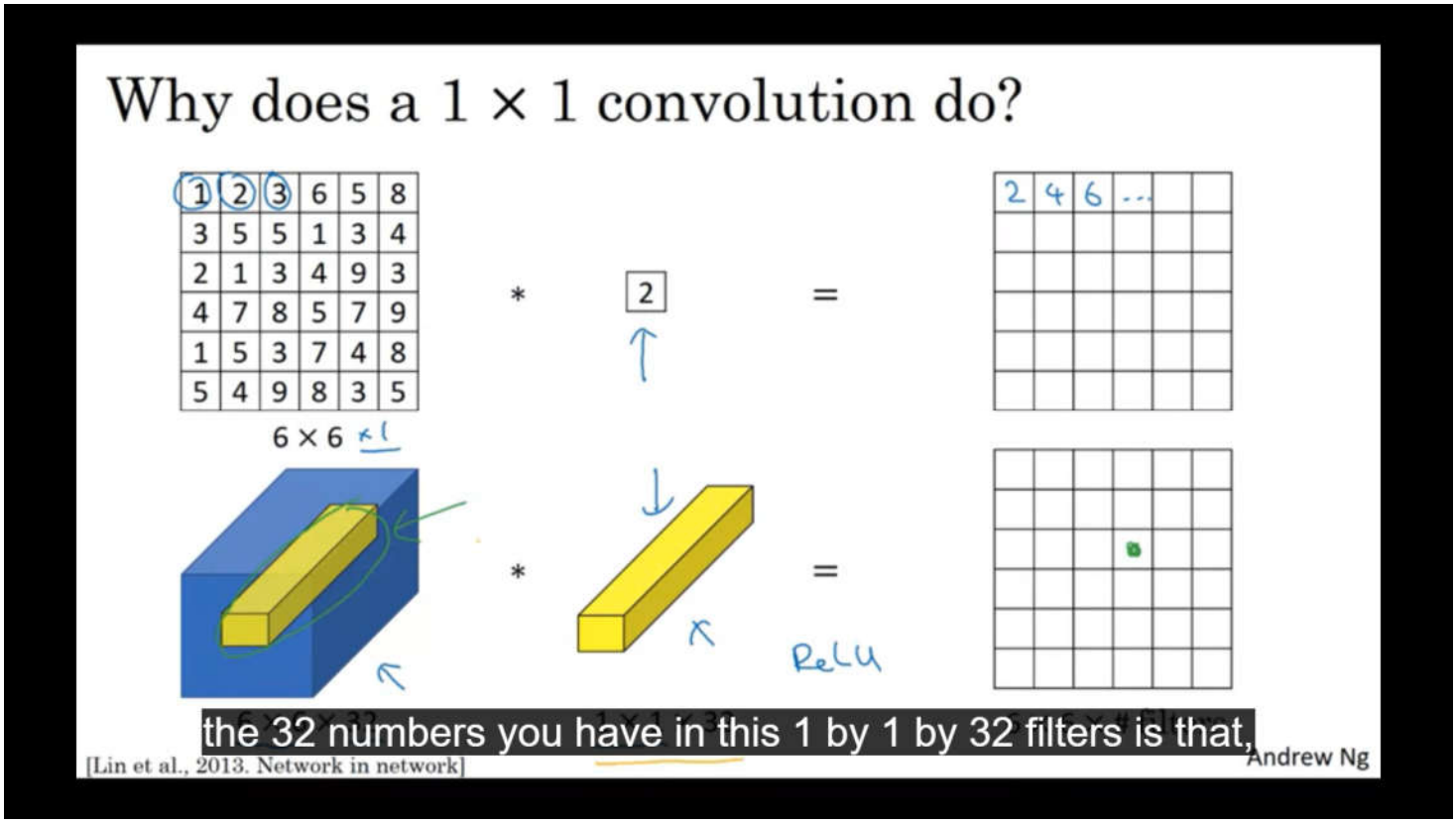
1 × 1 × 32

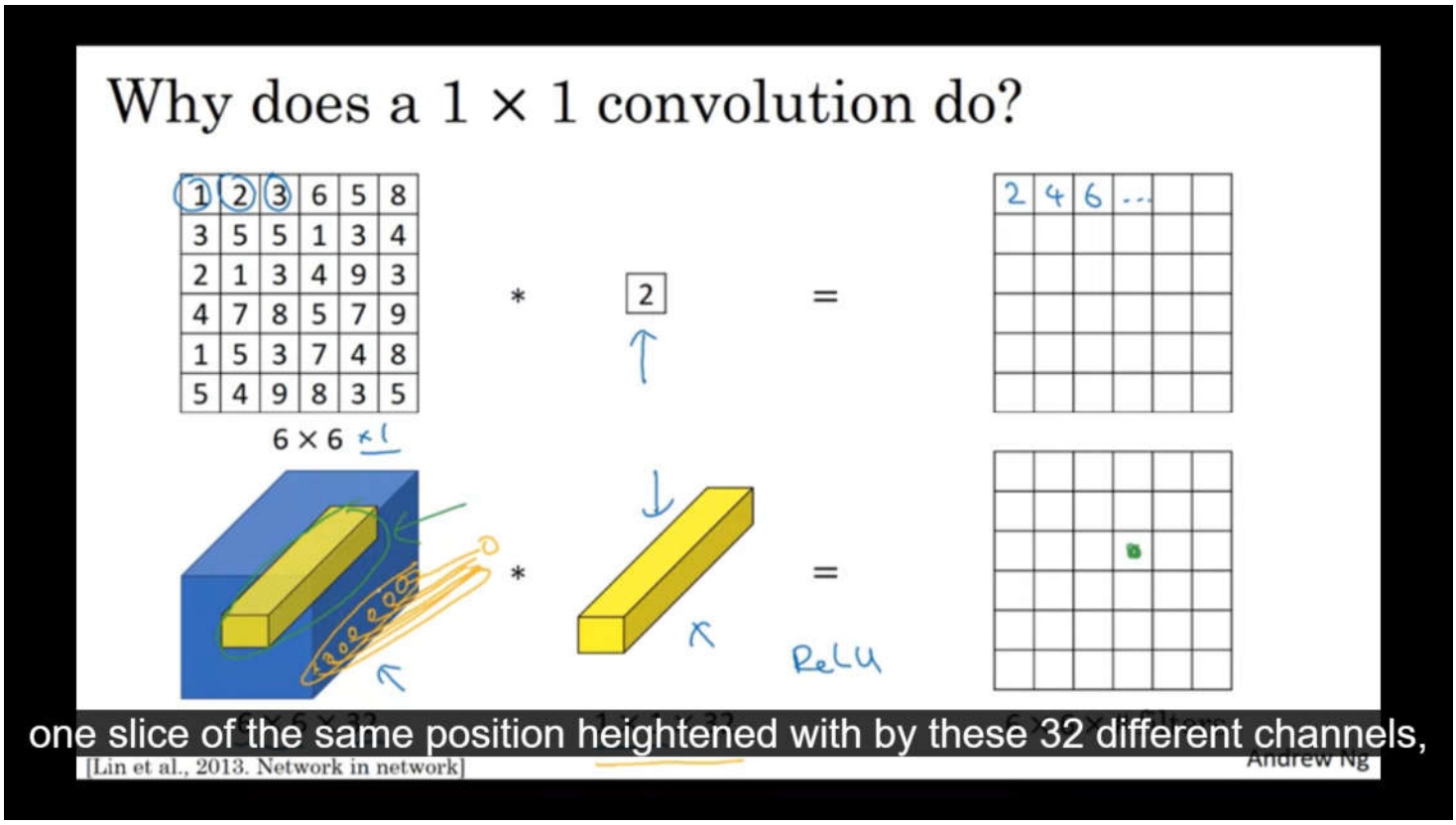
6 × 6 × # filters

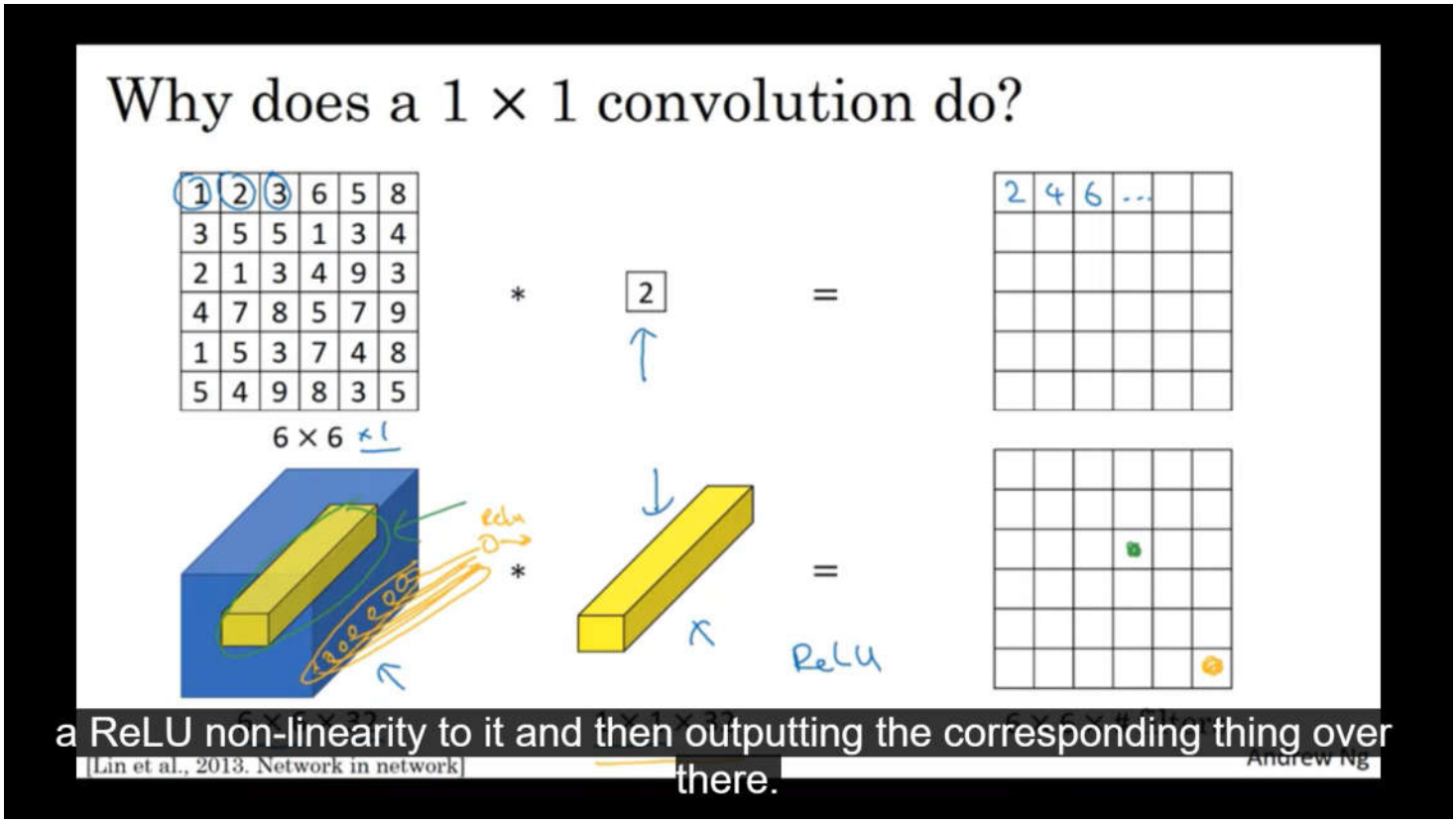
[Lin et al., 2013. Network in network]

Andrew Ng

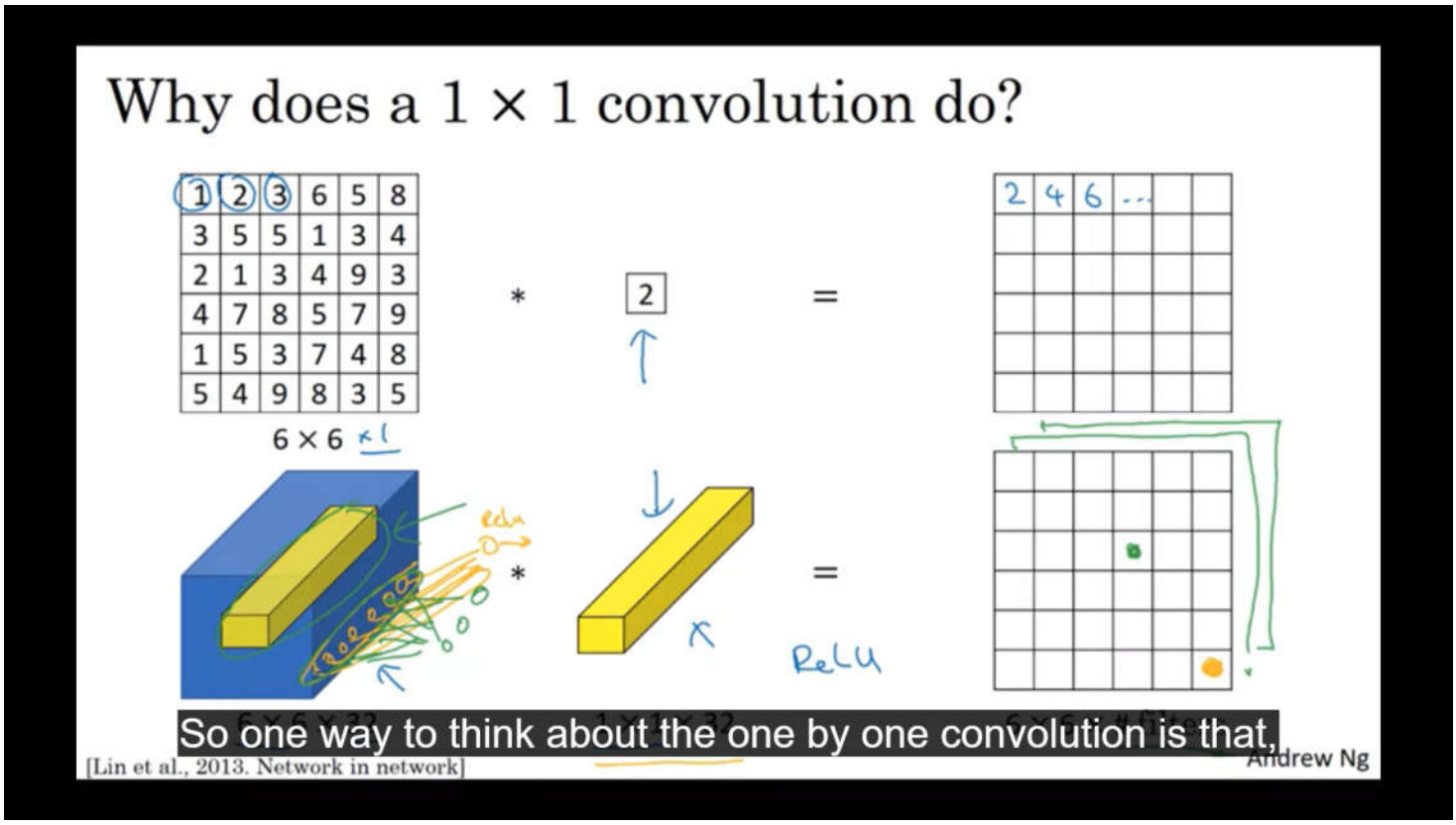
1:10 / 6:39


Why does a 1 × 1 convolution do?
6 × 6 × 1
ReLU
the 32 numbers you have in this 1 by 1 by 32 filters is that,
[Lin et al., 2013. Network in network]
Andrew Ng

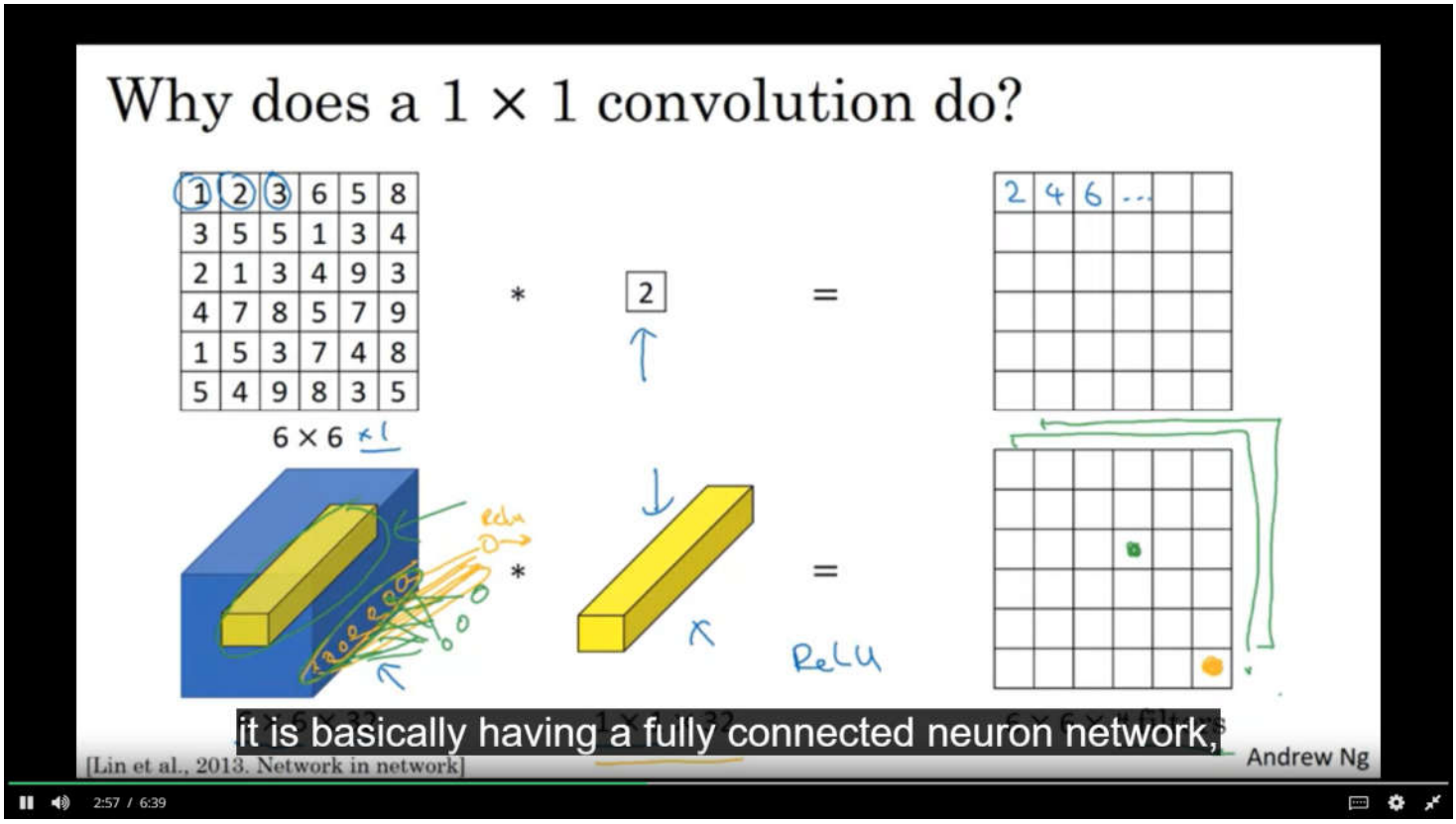

Why does a 1 × 1 convolution do?
6 × 6 × 1
ReLU
one slice of the same position heightened with by these 32 different channels,
[Lin et al., 2013. Network in network]
Andrew Ng

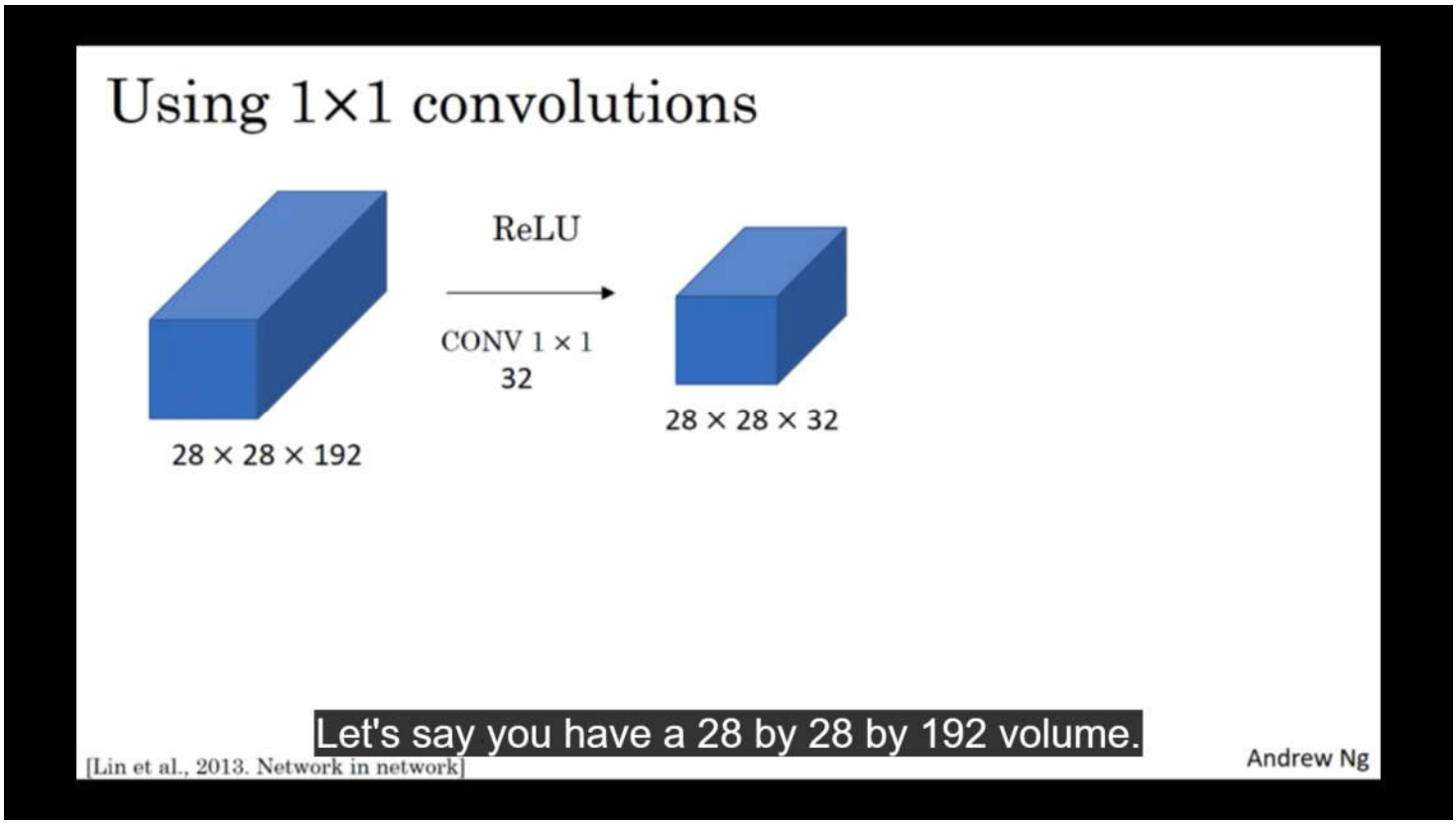


a ReLU non-linearity to it and then outputting the corresponding thing over there.

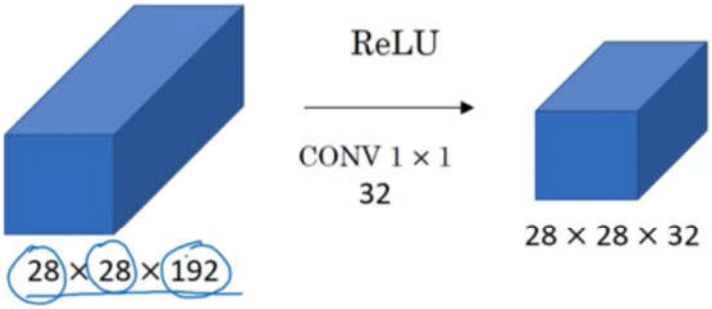


So one way to think about the one by one convolution is that,


Why does a 1 × 1 convolution do?
6 × 6 × 1
ReLU
it is basically having a fully connected neuron network,
[Lin et al., 2013. Network in network]
Andrew Ng
2:57 / 6:39


Using 1×1 convolutions
ReLU
CONV 1 × 1
32
28 × 28 × 192
28 × 28 × 32
Let's say you have a 28 by 28 by 192 volume.
[Lin et al., 2013. Network in network]
Andrew Ng

Using 1×1 convolutions

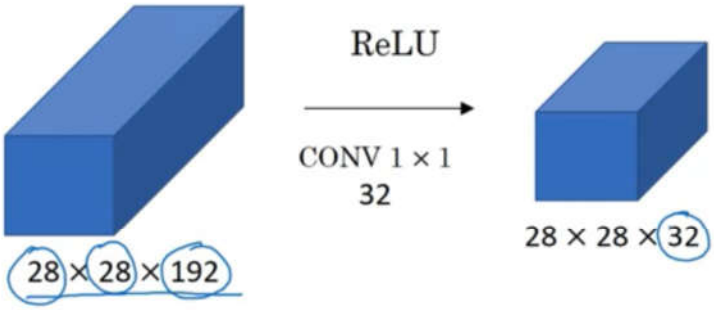


But one of a number of channels has gotten too big and we want to shrink that.

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

Andrew Ng

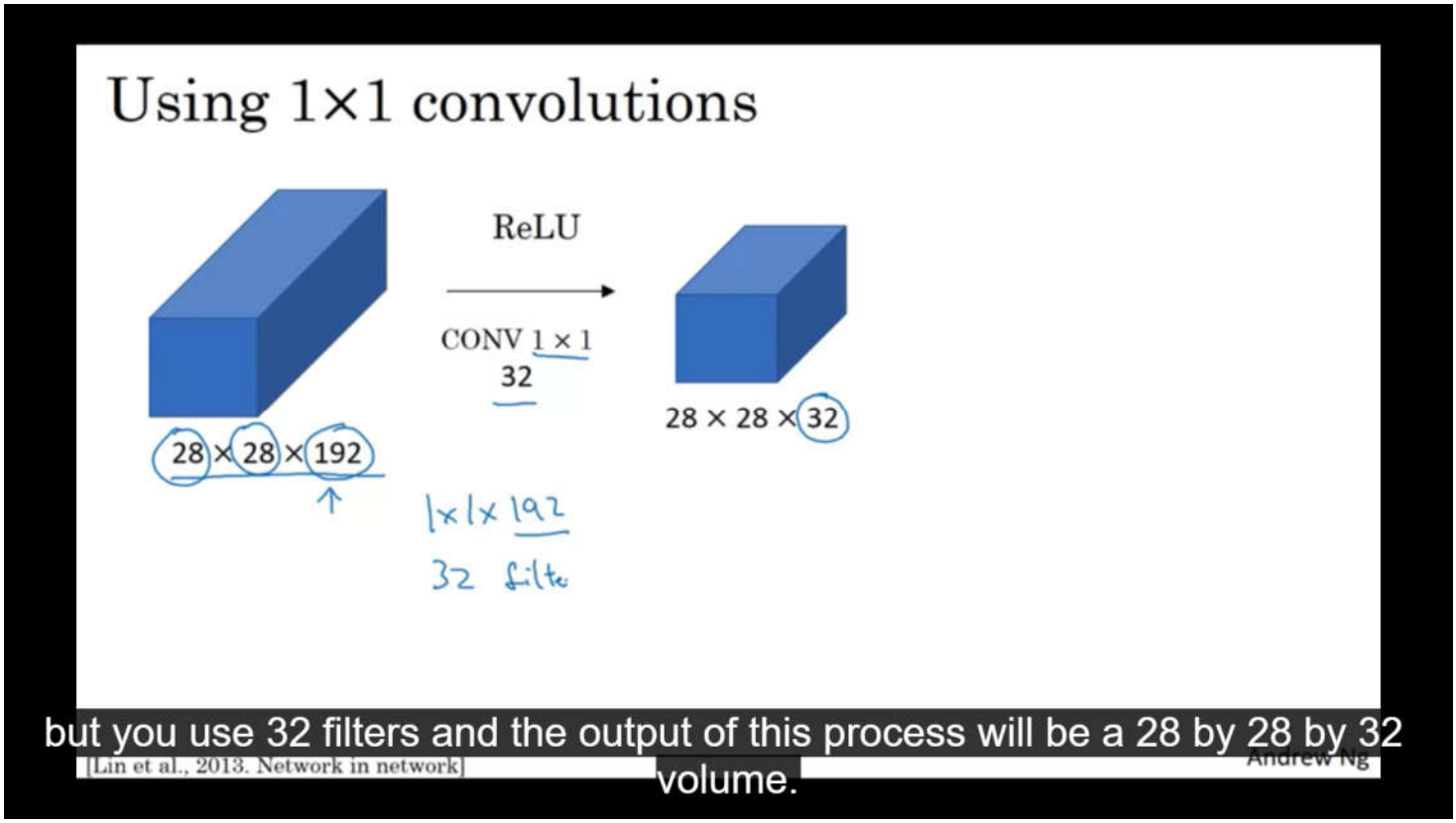
Using 1×1 convolutions

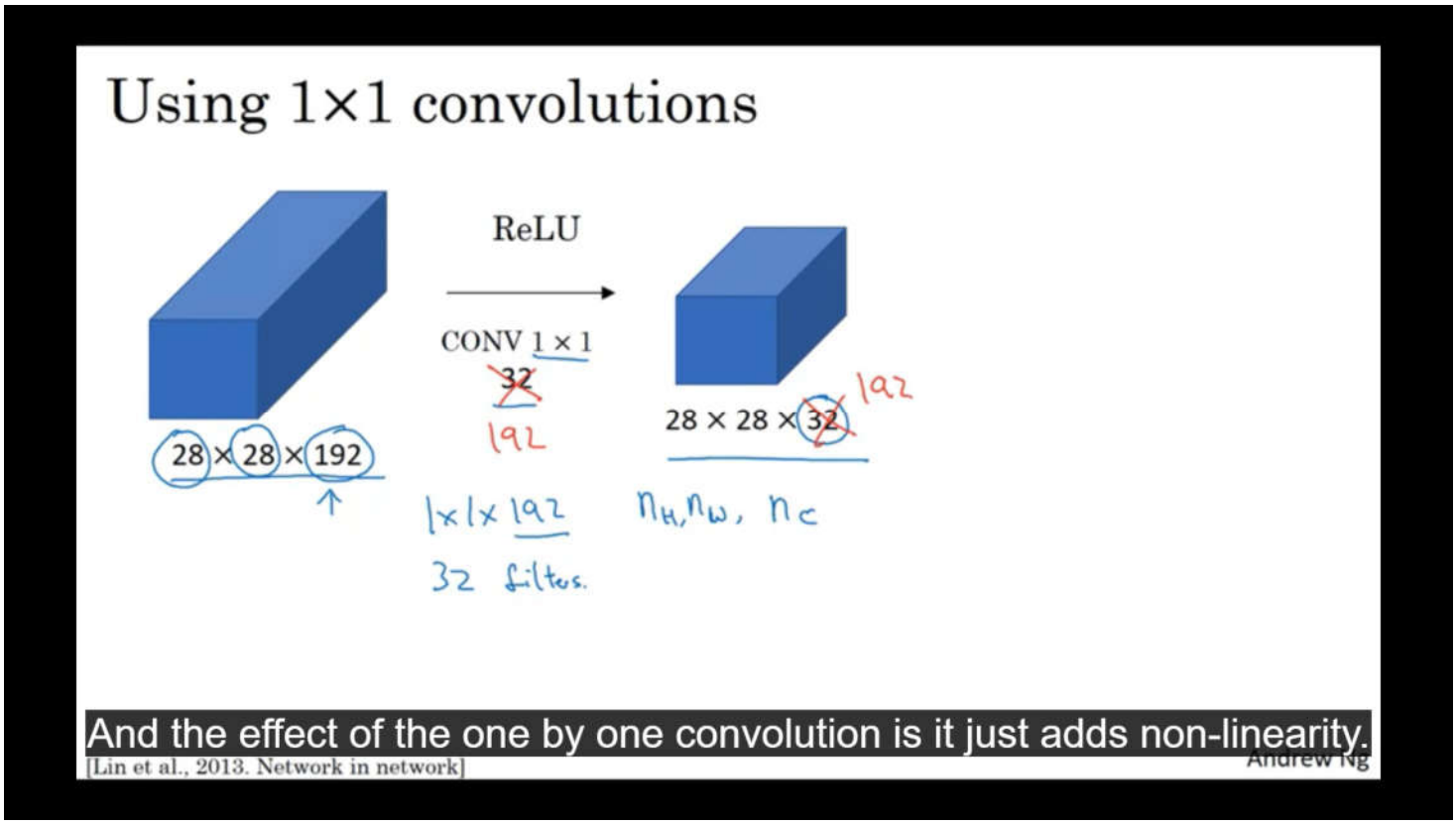


How do you shrink it to a 28 by 28 by 32 dimensional volume?

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

Andrew Ng


Using 1×1 convolutions
ReLU
CONV 1 × 1
32
28 × 28 × 192
28 × 28 × 32
1×1×192
32 filte
but you use 32 filters and the output of this process will be a 28 by 28 by 32 volume.
[Lin et al., 2013. Network in network]
Andrew Ng


Using 1×1 convolutions
ReLU
CONV 1 × 1
32
192
28 × 28 × 192
28 × 28 × 32
192
1×1×192
32 filters.
nH, nW, nc
And the effect of the one by one convolution is it just adds non-linearity.
[Lin et al., 2013. Network in network]
Andrew Ng

Press Esc to exit full screen

deeplearning.ai

Case Studies

Inception network motivation

When designing a layer for a ConvNet, you might have to pick,

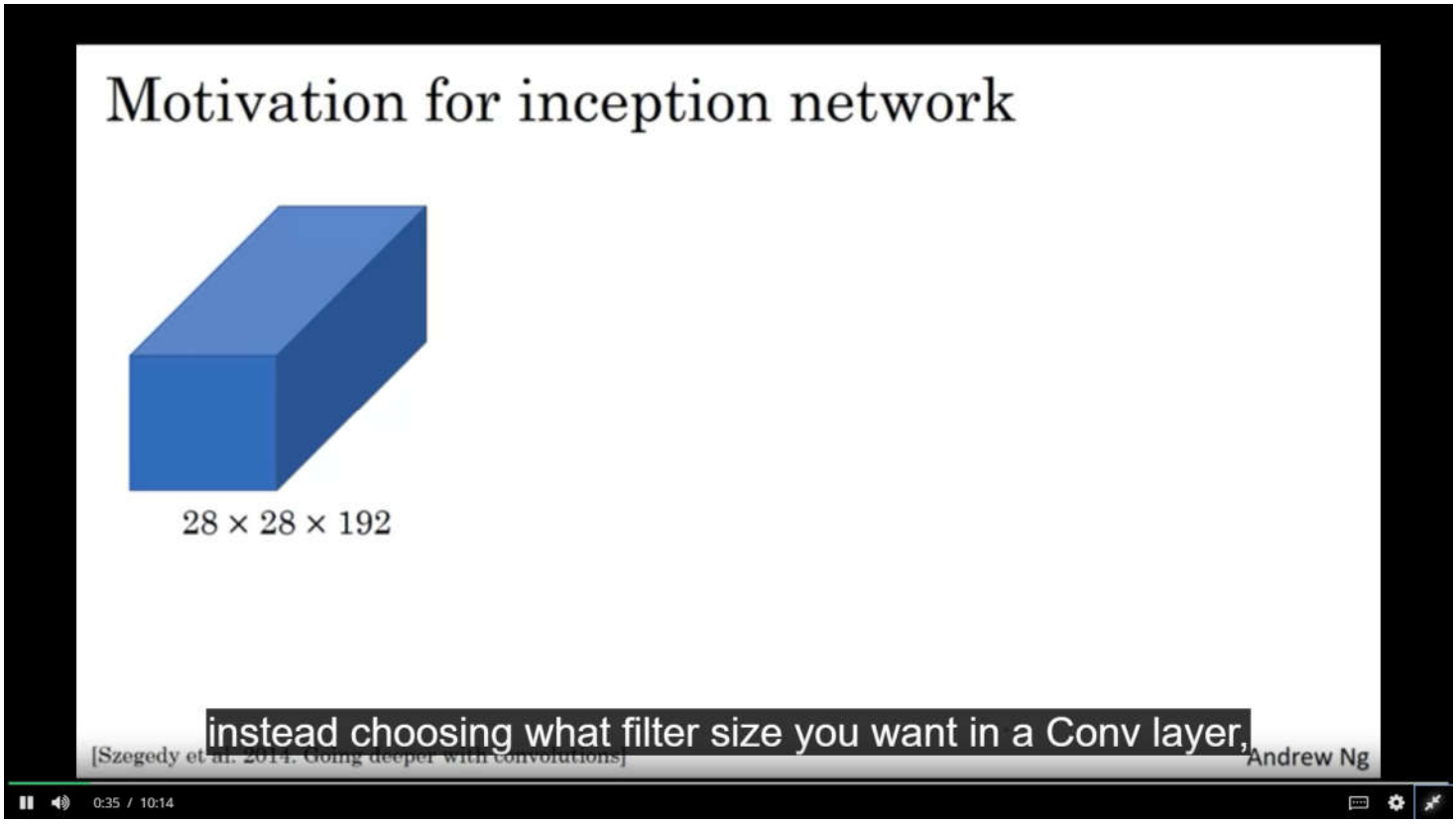
Motivation for inception network

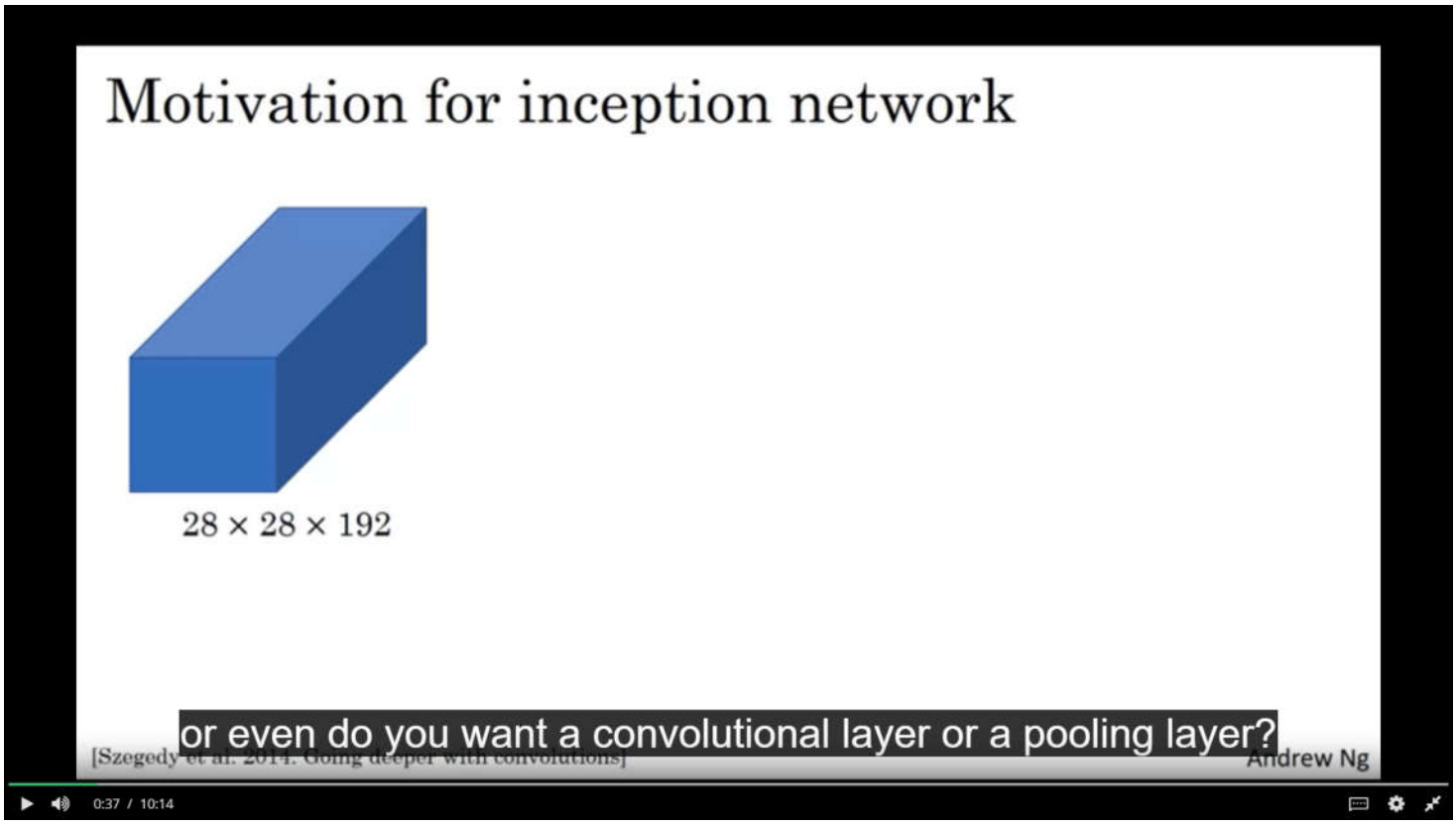
$28 \times 28 \times 192$

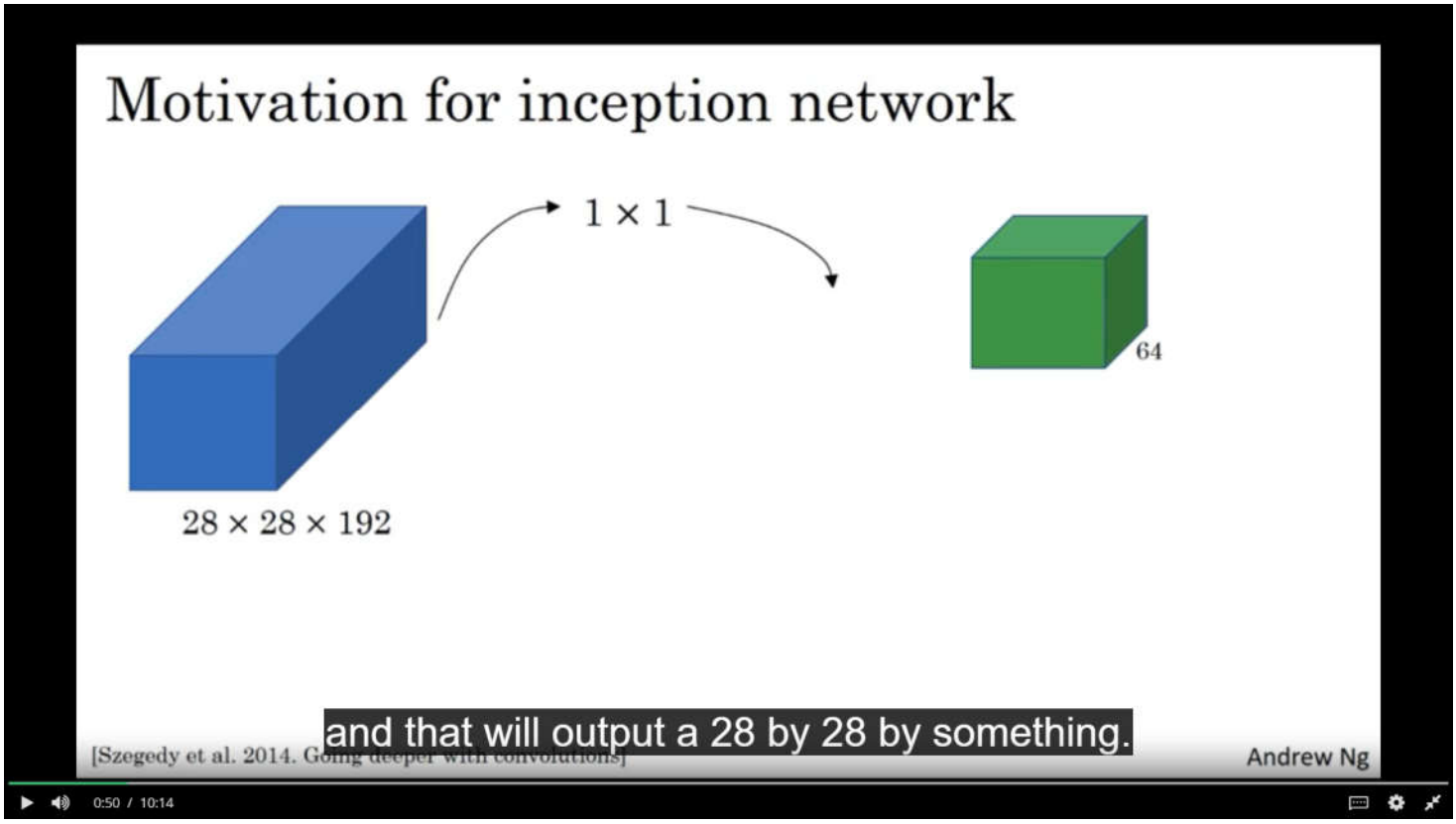
So what the inception network or what an inception layer says is,

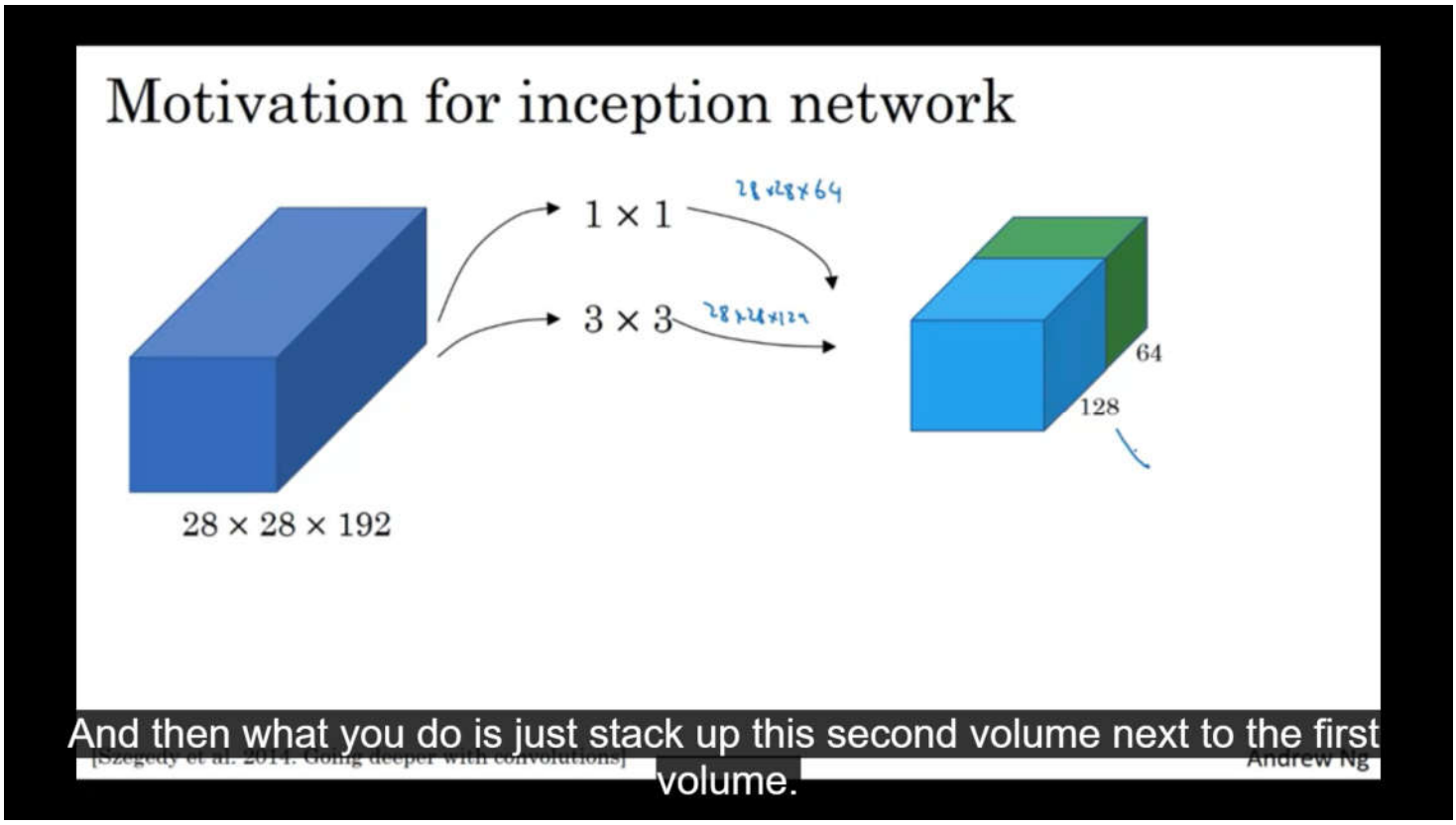
[Szegedy et al. 2014. Going deeper with convolutions]

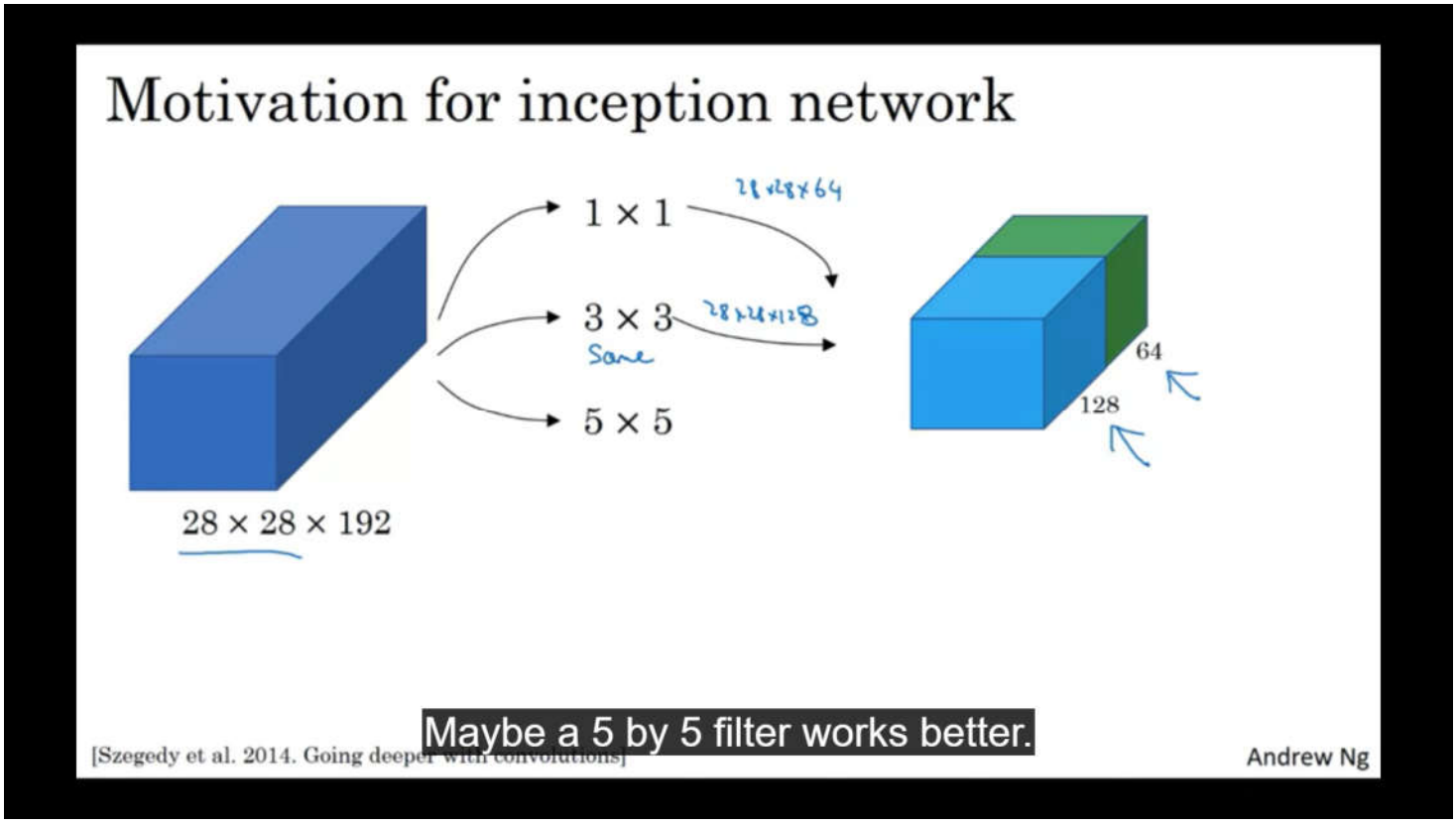
Andrew Ng

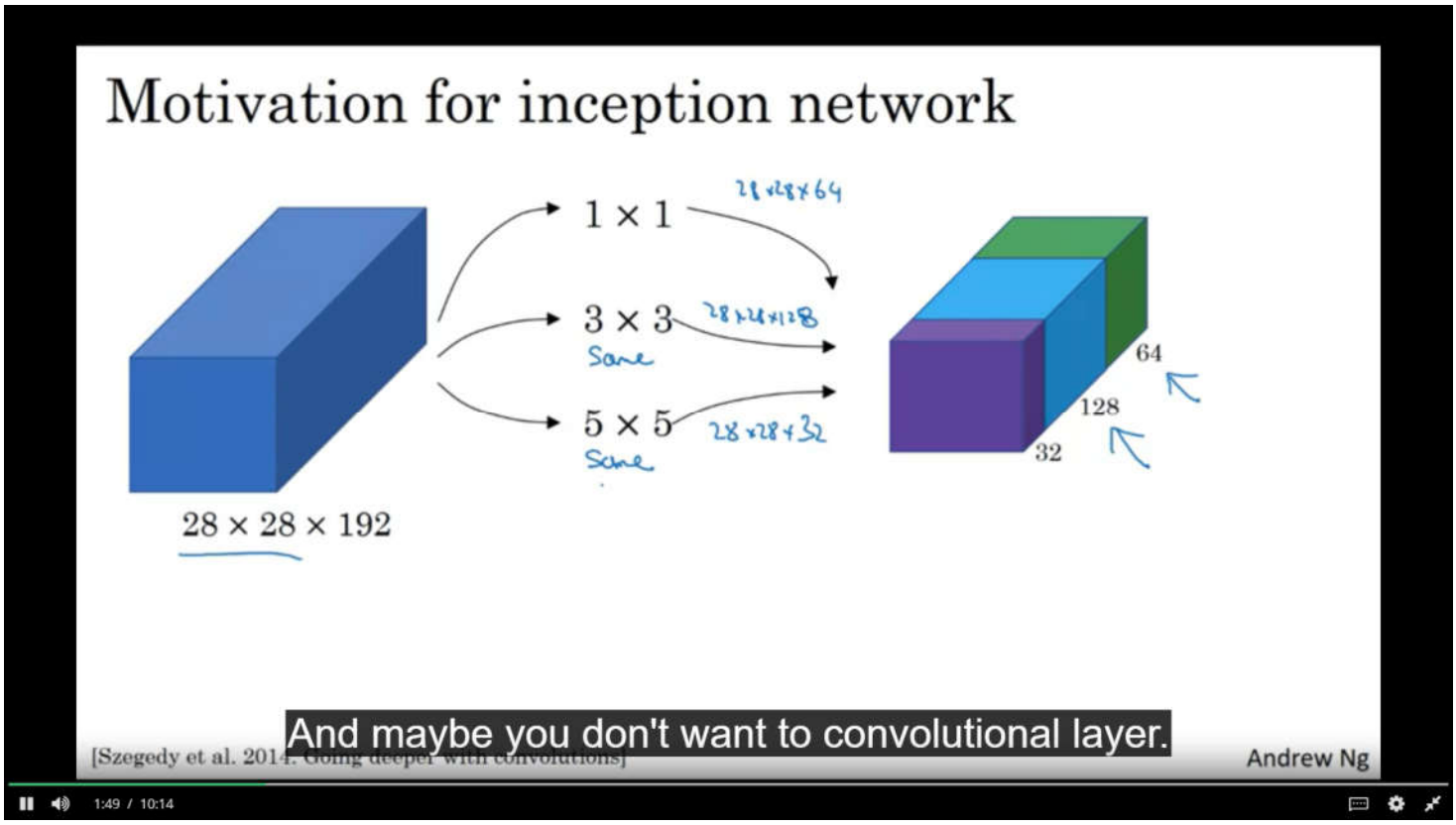

Motivation for inception network
28 × 28 × 192
instead choosing what filter size you want in a Conv layer,
[Szegedy et al. 2014. Going deeper with convolutions]
Andrew Ng
0:35 / 10:14


Motivation for inception network
28 × 28 × 192
or even do you want a convolutional layer or a pooling layer?
[Szegedy et al. 2014. Going deeper with convolutions]
Andrew Ng
0:37 / 10:14

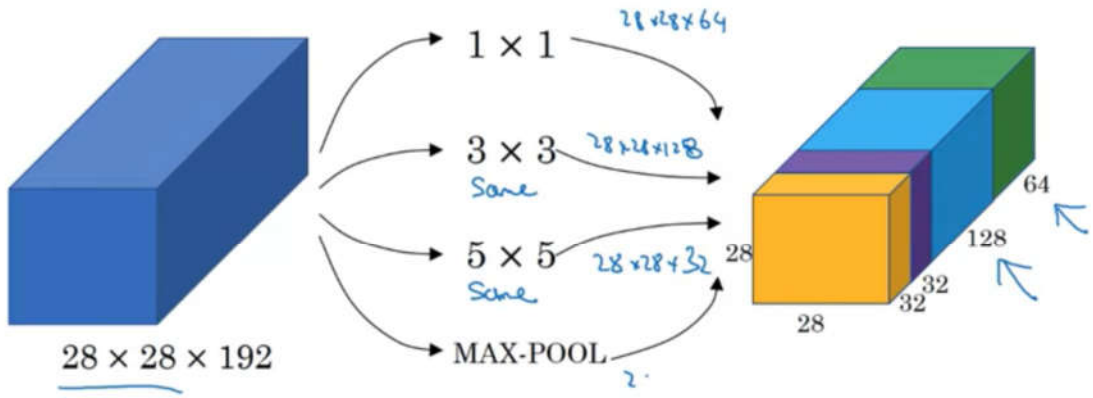

Motivation for inception network
1 × 1
64
28 × 28 × 192
and that will output a 28 by 28 by something.
[Szegedy et al. 2014. Going deeper with convolutions]
Andrew Ng
0:50 / 10:14


Motivation for inception network
1 × 1
28×28×64
3 × 3
28×28×128
64
128
28 × 28 × 192
And then what you do is just stack up this second volume next to the first volume.
[Szegedy et al. 2014. Going deeper with convolutions]
Andrew Ng


Motivation for inception network
1 × 1
28×28×64
3 × 3
28×28×128
Same
5 × 5
64
128
28 × 28 × 192
Maybe a 5 by 5 filter works better.
[Szegedy et al. 2014. Going deeper with convolutions]
Andrew Ng


Motivation for inception network
1 × 1
28×28×64
3 × 3
28×28×128
Same
5 × 5
28×28×32
Same
64
128
32
28 × 28 × 192
And maybe you don't want to convolutional layer.
[Szegedy et al. 2014. Going deeper with convolutions]
Andrew Ng
1:49 / 10:14

Motivation for inception network

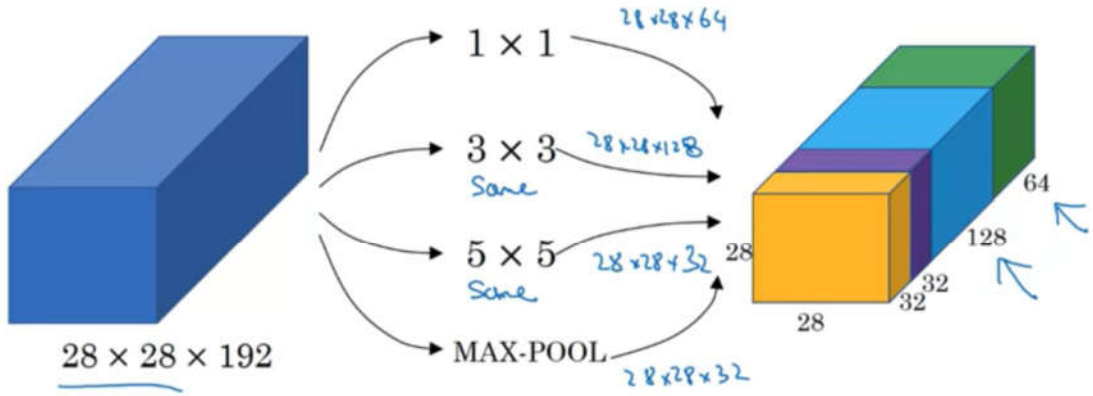


And here pooling outputs 28 by 28 by 32.

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

Andrew Ng

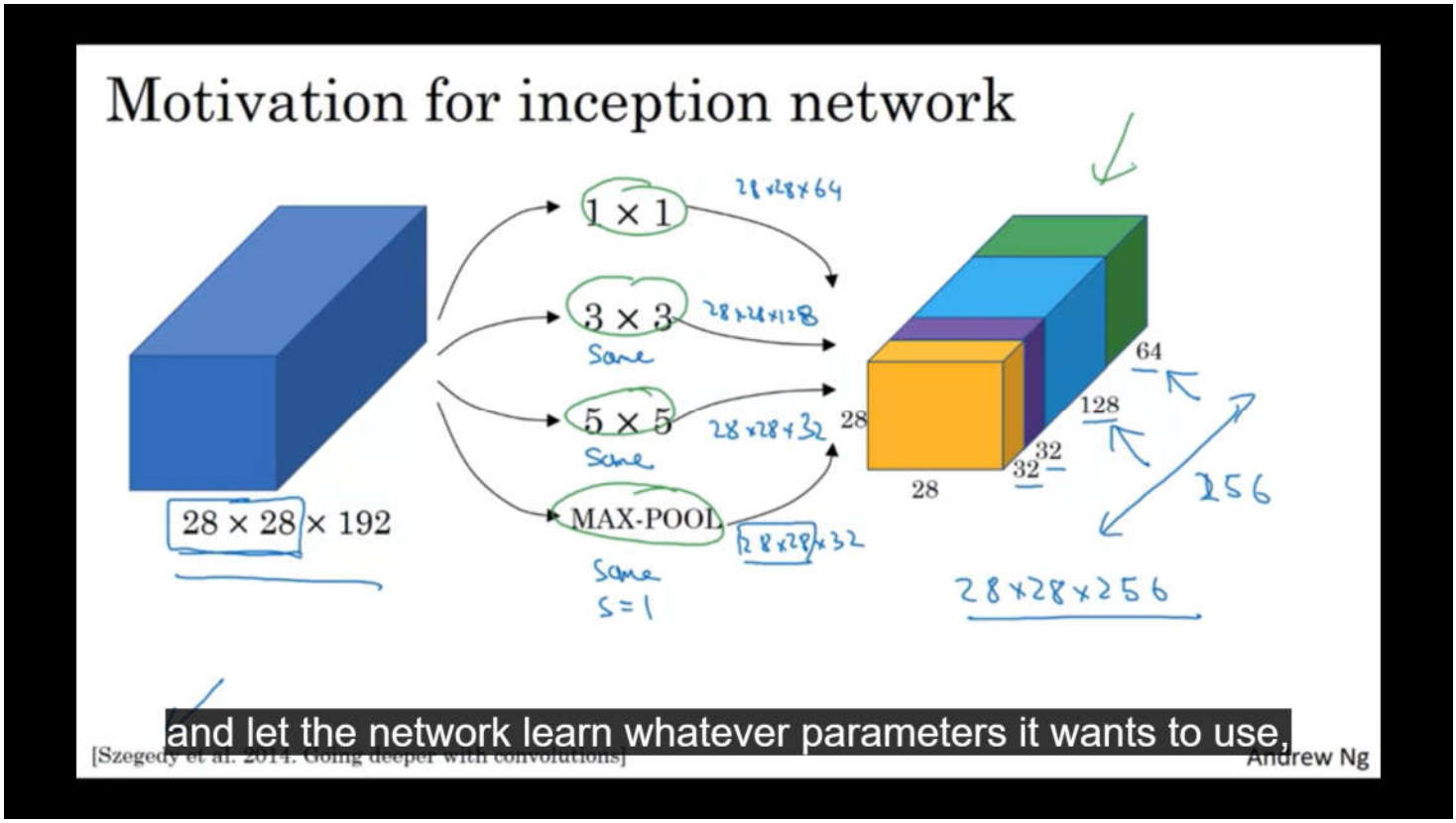
Motivation for inception network

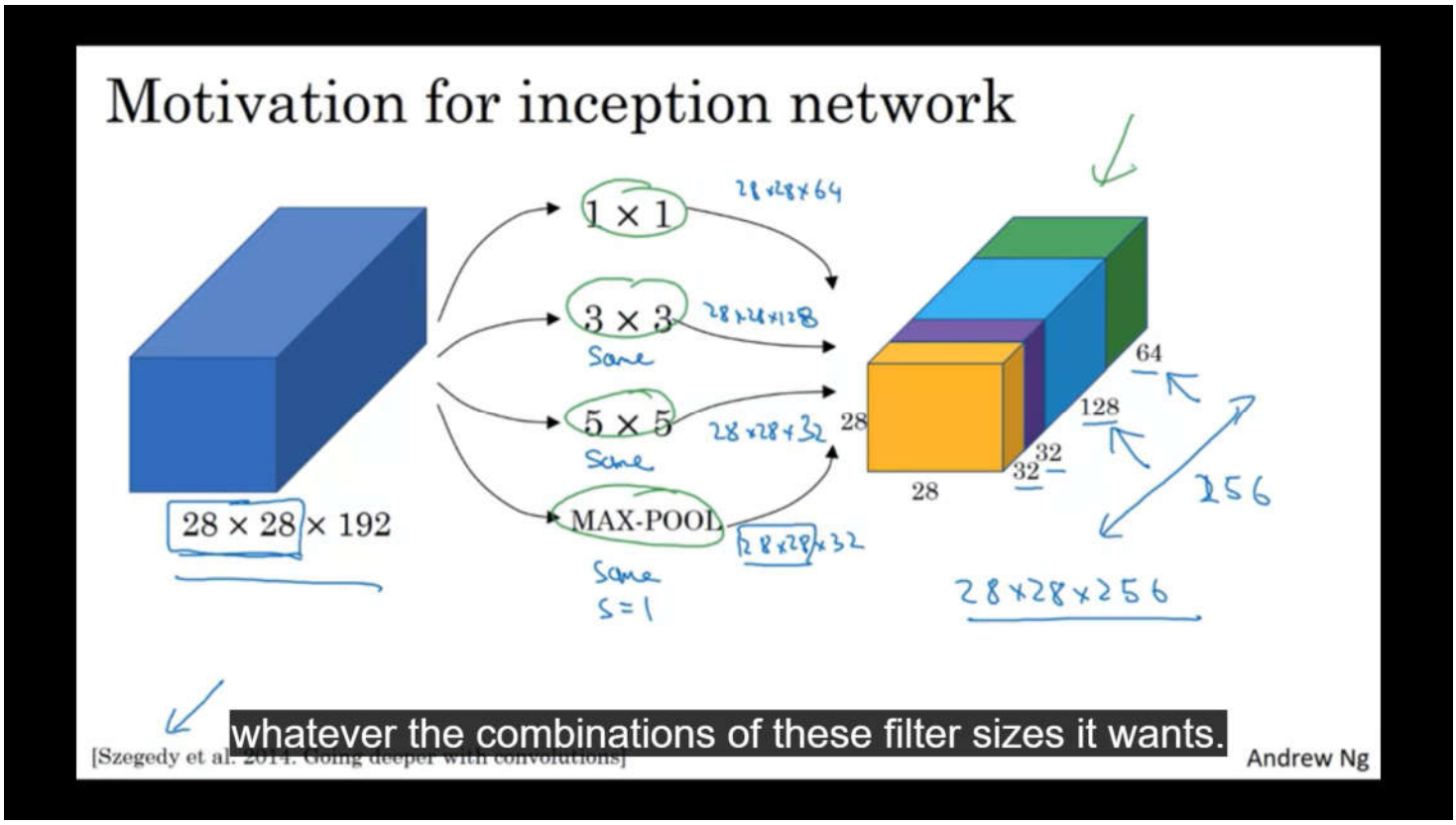


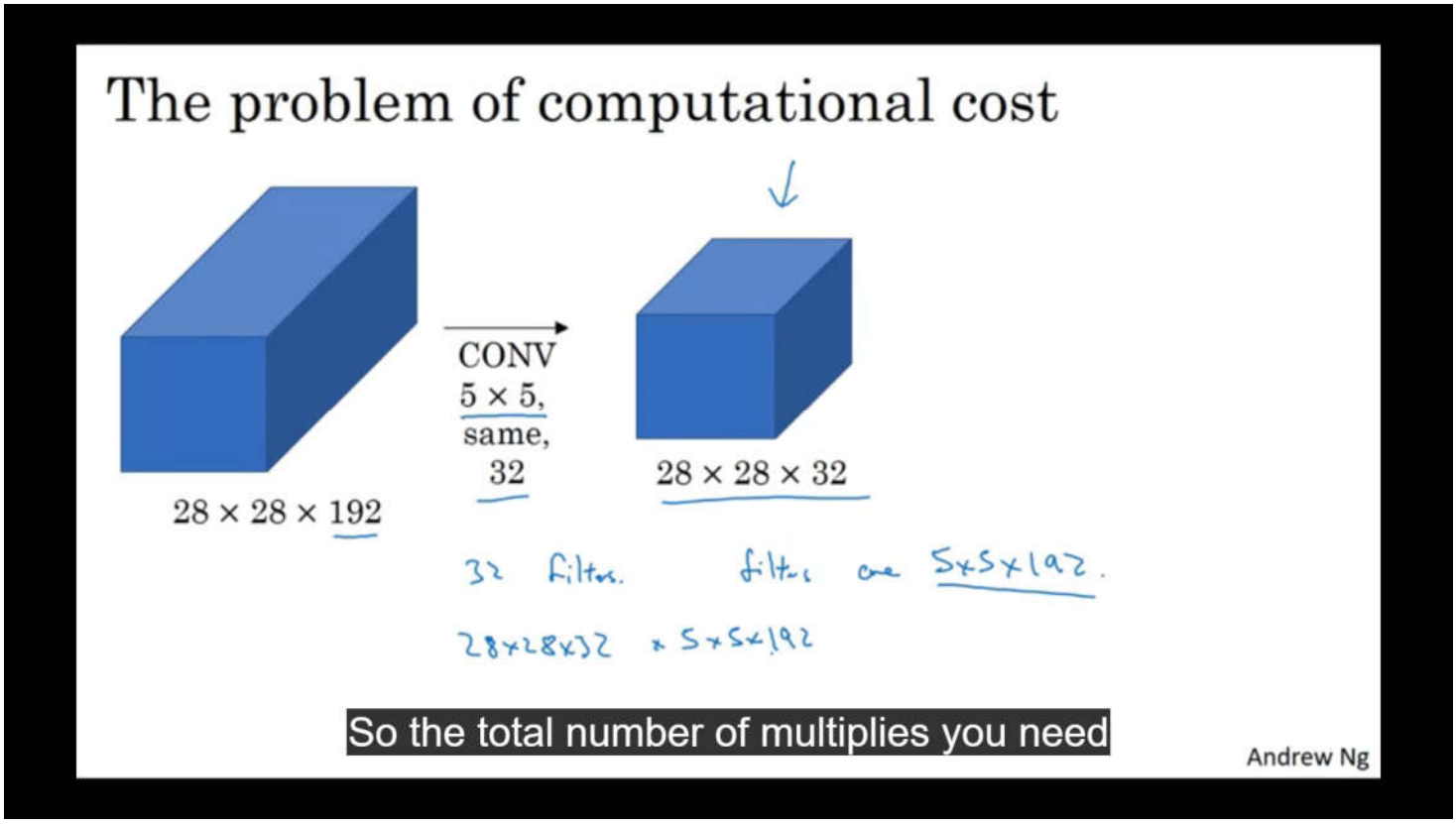
Now in order to make all the dimensions match,

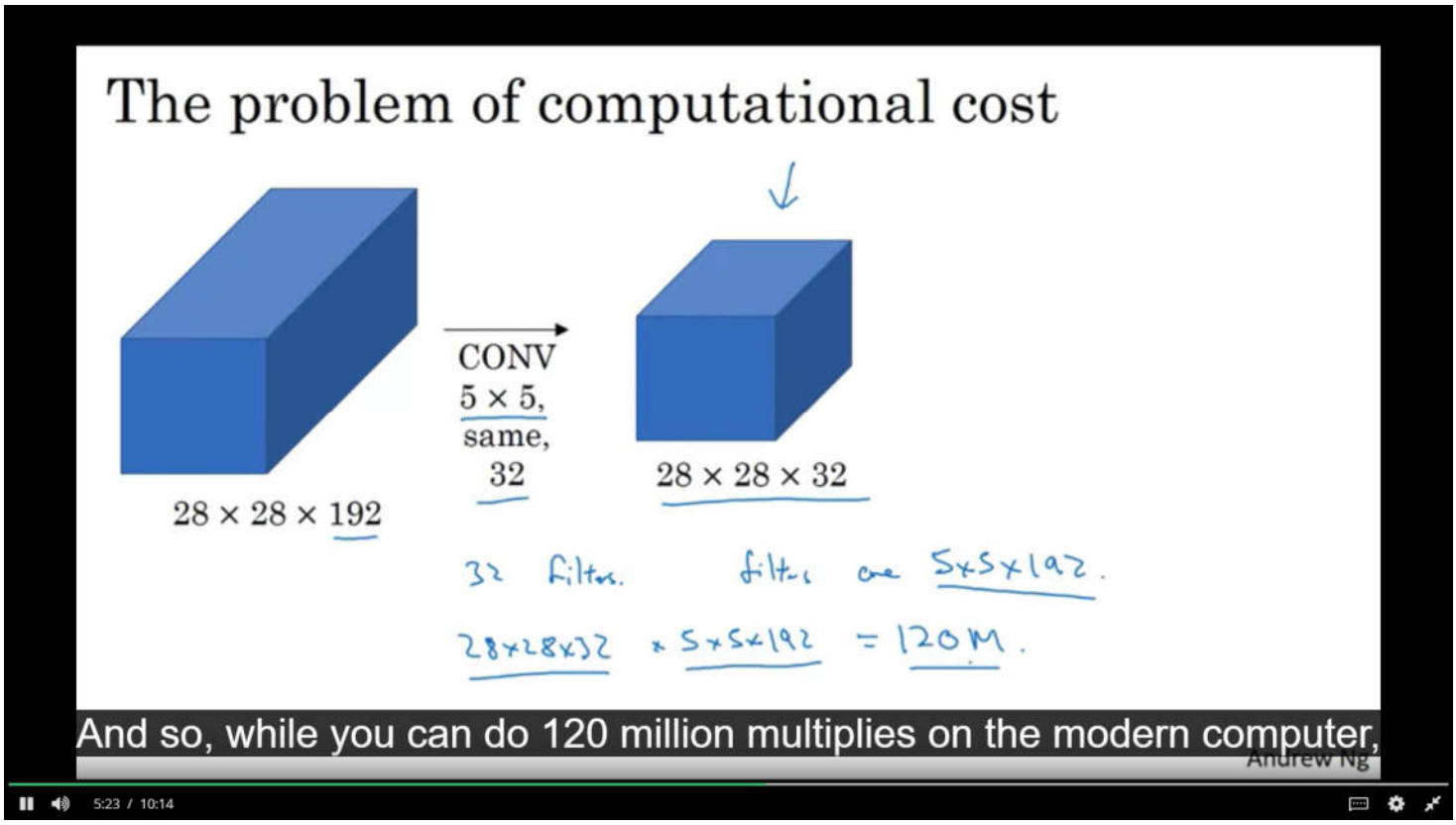
[Szegedy et al. 2014. Going deeper with convolutions]

Andrew Ng

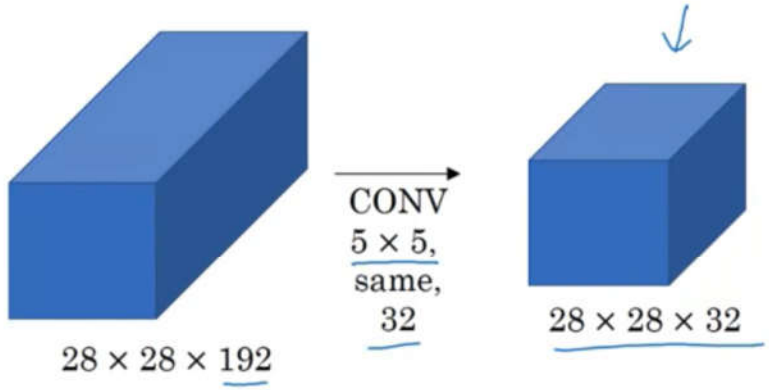

Motivation for inception network
1 × 1
3 × 3
5 × 5
MAX-POOL
28 × 28 × 192
28×28×64
28×28×128
28×28×32
28×28×32
Same
Same
Same
S=1
28
28
64
128
32
32
256
28×28×256
and let the network learn whatever parameters it wants to use,
[Szegedy et al. 2014. Going deeper with convolutions]
Andrew Ng


Motivation for inception network
1 × 1
3 × 3
5 × 5
MAX-POOL
28 × 28 × 192
28×28×64
28×28×128
28×28×32
28×28×32
Same
Same
Same
S=1
28
28
64
128
32
32
256
28×28×256
whatever the combinations of these filter sizes it wants.
[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 5×5×192.
28×28×32 × 5×5×192
So the total number of multiplies you need
Andrew Ng


The problem of computational cost
CONV
5 × 5,
same,
32
28 × 28 × 192
28 × 28 × 32
32 filters. filters are 5×5×192.
28×28×32 × 5×5×192 = 120M.
And so, while you can do 120 million multiplies on the modern computer,
Andrew Ng
5:23 / 10:14

The problem of computational cost



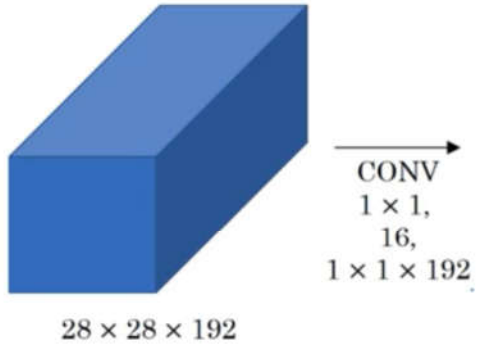
32 filters. filters are 5×5×192.

28×28×32 × 5×5×192 = 120M.

Here is an alternative architecture for inputting 28 by 28 by 192,

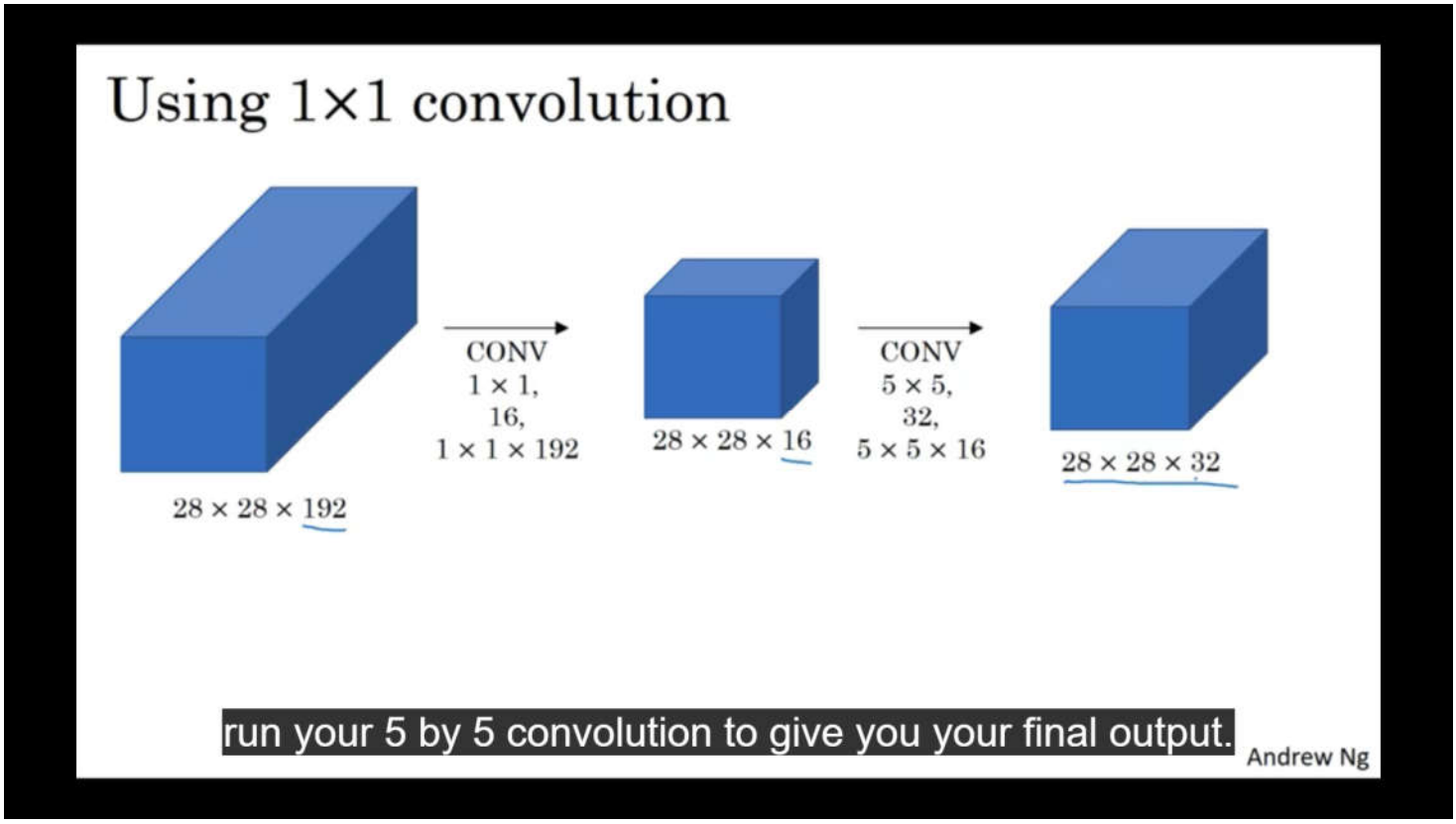
Andrew Ng

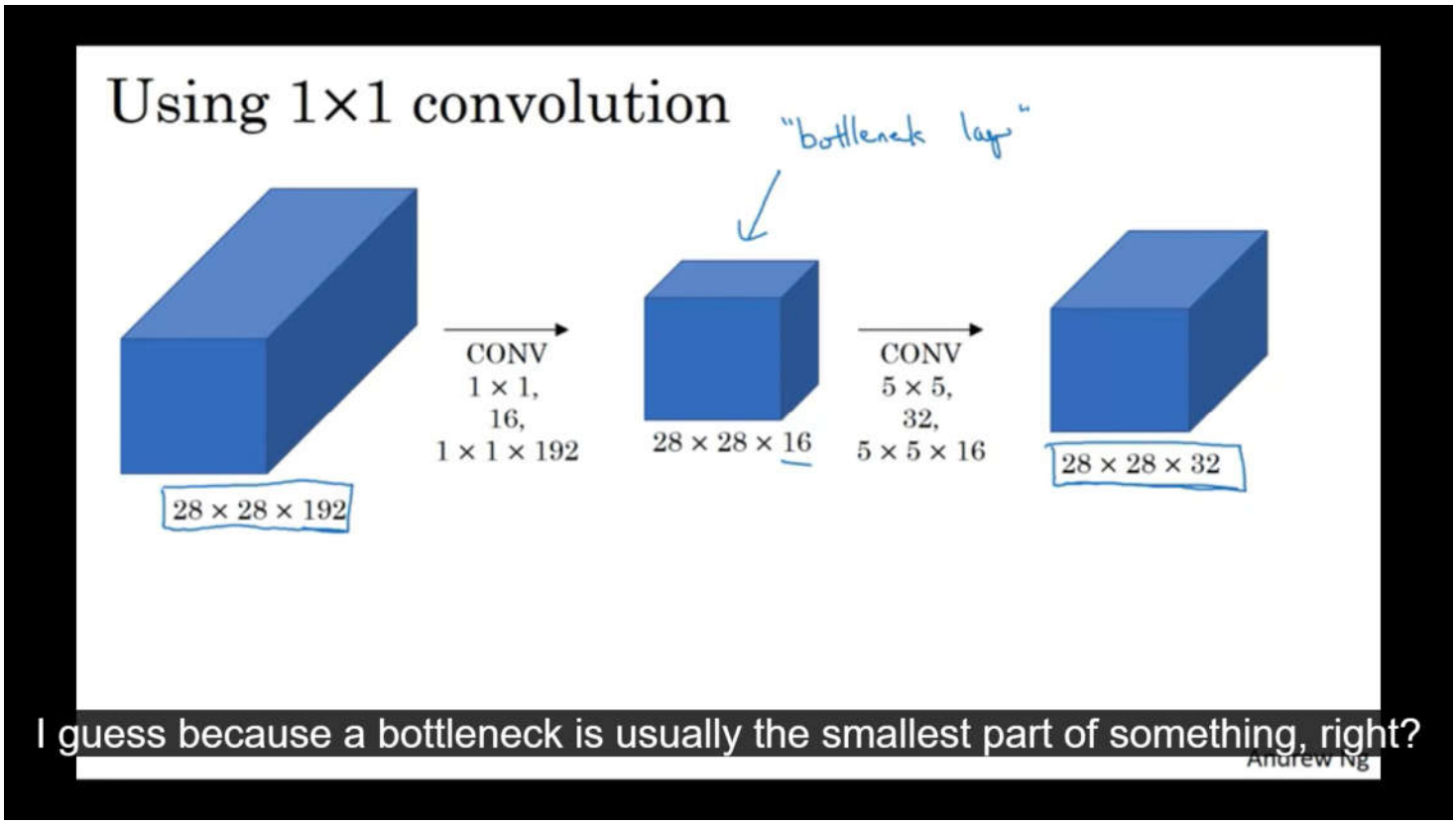
Using 1×1 convolution

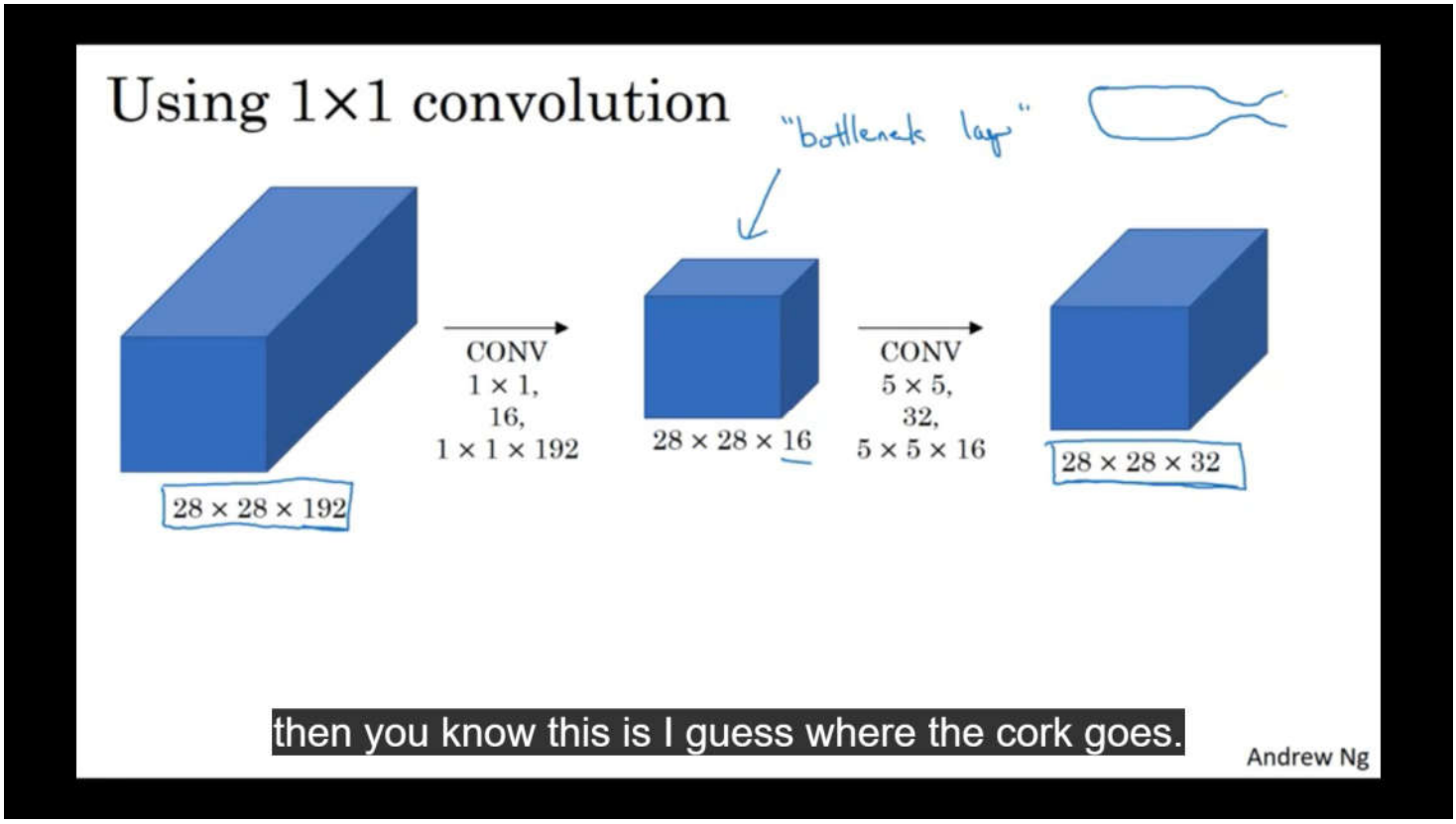


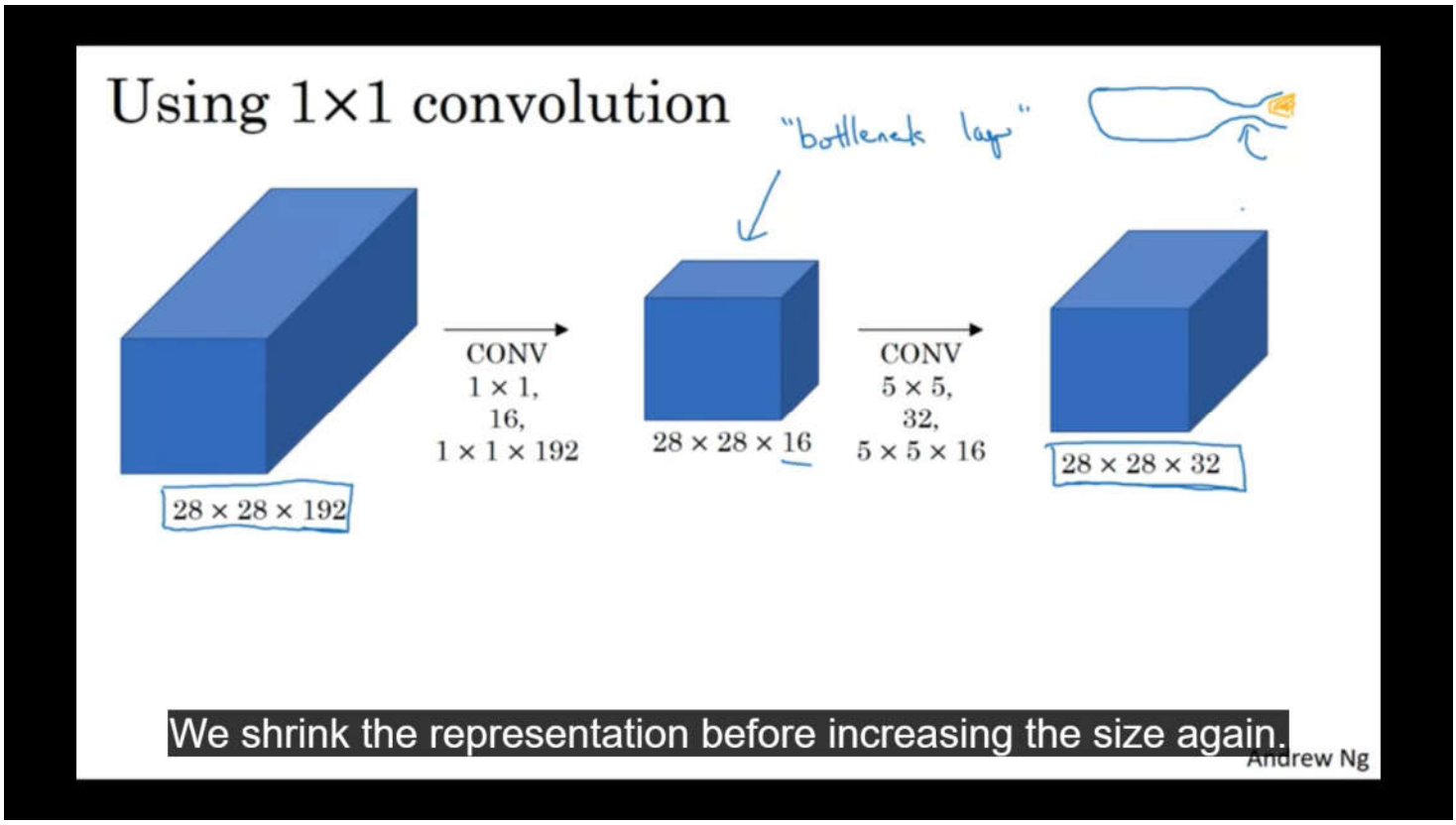
use a 1 by 1 convolution to reduce the volume to 16 channels instead of 192 channels,

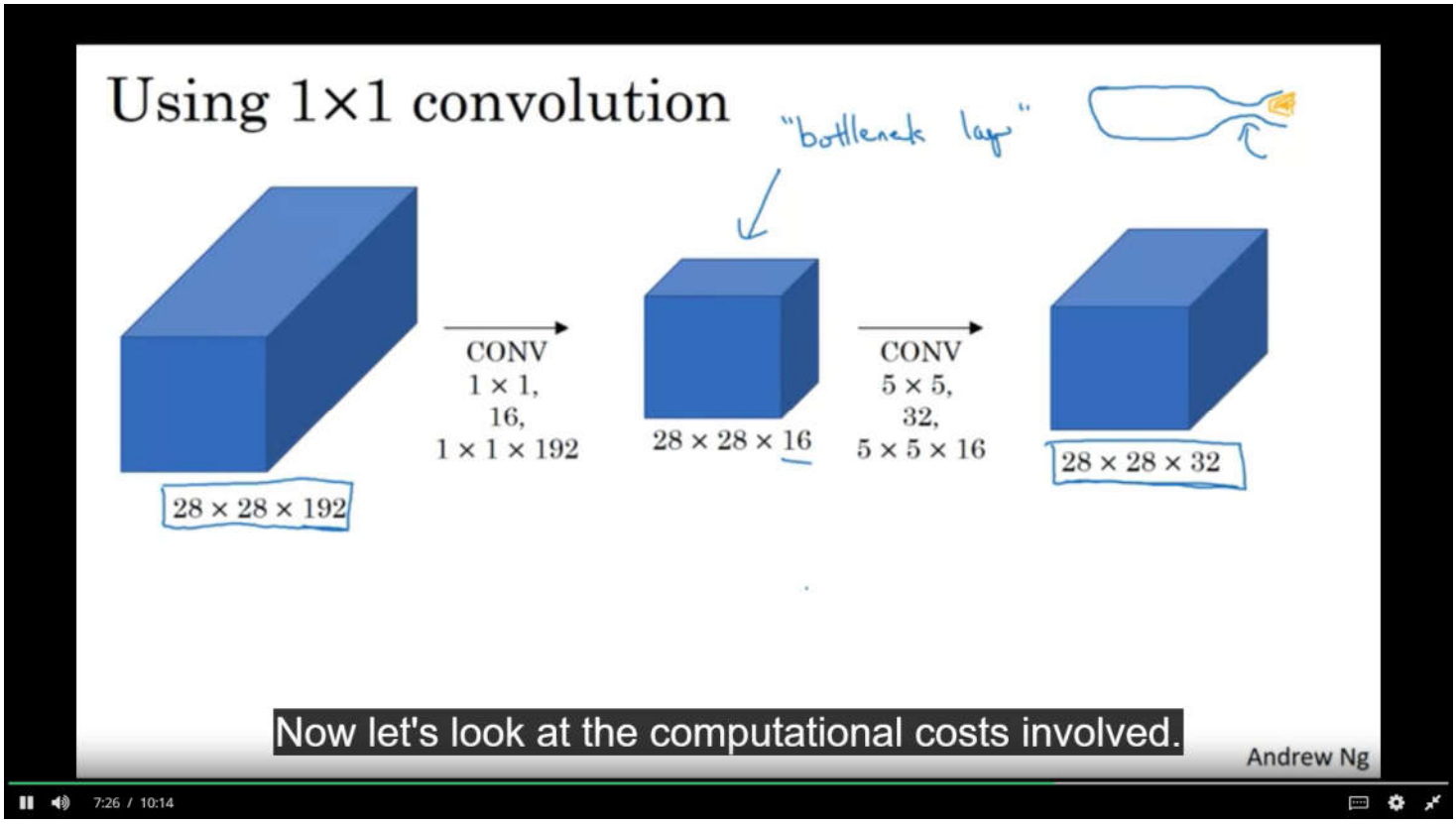
Andrew Ng

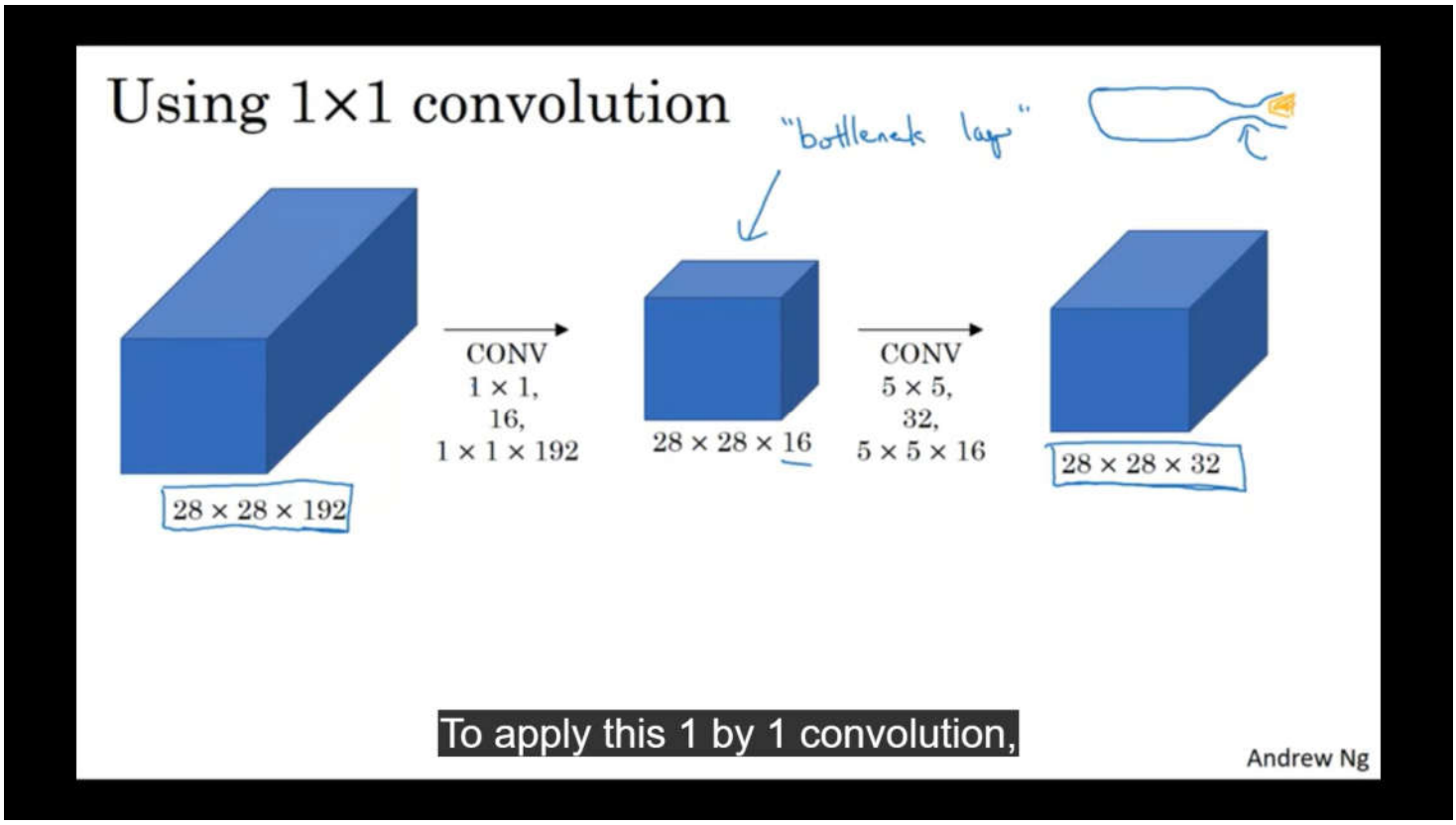

Using 1×1 convolution
28 × 28 × 192
CONV
1 × 1,
16,
1 × 1 × 192
28 × 28 × 16
CONV
5 × 5,
32,
5 × 5 × 16
28 × 28 × 32
run your 5 by 5 convolution to give you your final output.
Andrew Ng

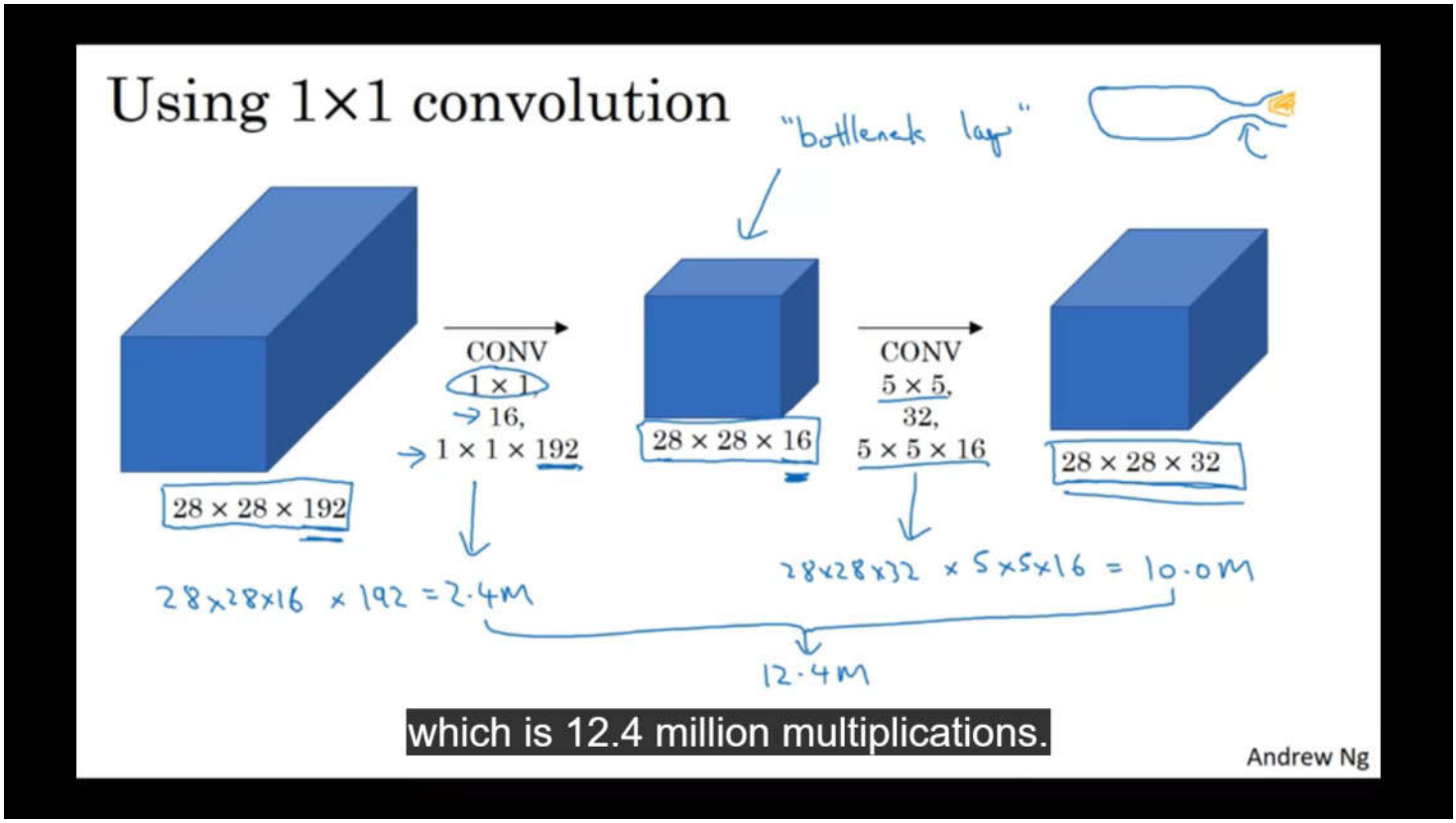

Using 1×1 convolution
"bottleneck layer"
28 × 28 × 192
CONV
1 × 1,
16,
1 × 1 × 192
28 × 28 × 16
CONV
5 × 5,
32,
5 × 5 × 16
28 × 28 × 32
I guess because a bottleneck is usually the smallest part of something, right?

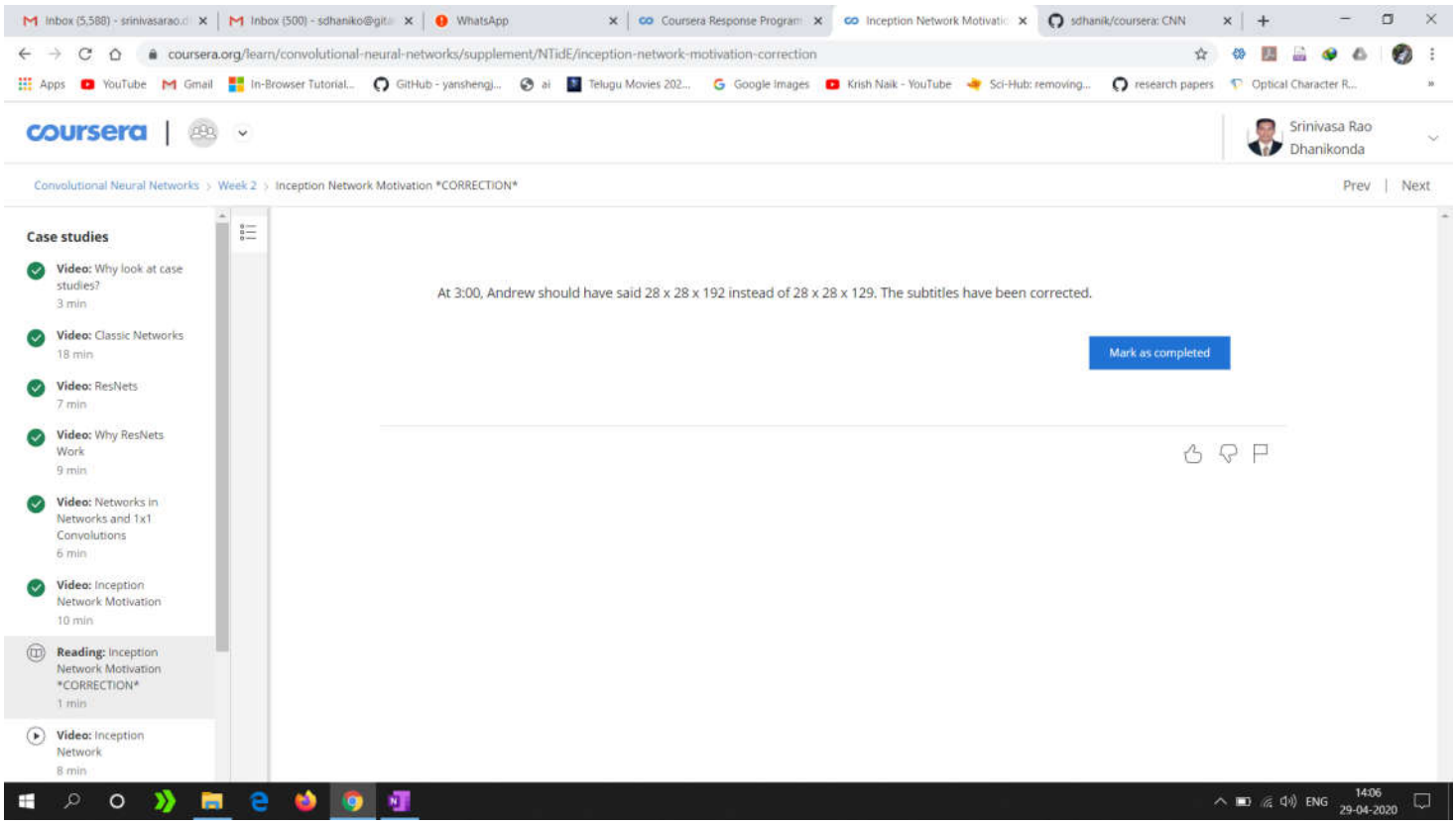

Using 1×1 convolution
"bottleneck layer"
CONV
1 × 1,
16,
1 × 1 × 192
28 × 28 × 192
28 × 28 × 16
CONV
5 × 5,
32,
5 × 5 × 16
28 × 28 × 32
then you know this is I guess where the cork goes.
Andrew Ng


Using 1×1 convolution
"bottleneck layer"
CONV
1 × 1,
16,
1 × 1 × 192
28 × 28 × 192
28 × 28 × 16
CONV
5 × 5,
32,
5 × 5 × 16
28 × 28 × 32
We shrink the representation before increasing the size again.
Andrew Ng


Using 1×1 convolution
"bottleneck layer"
CONV
1 × 1,
16,
1 × 1 × 192
28 × 28 × 192
28 × 28 × 16
CONV
5 × 5,
32,
5 × 5 × 16
28 × 28 × 32
Now let's look at the computational costs involved.
Andrew Ng
7:26 / 10:14


Using 1×1 convolution
"bottleneck layer"
CONV
1 × 1,
16,
1 × 1 × 192
28 × 28 × 192
28 × 28 × 16
CONV
5 × 5,
32,
5 × 5 × 16
28 × 28 × 32
To apply this 1 by 1 convolution,
Andrew Ng


Using 1×1 convolution
"bottleneck layer"
CONV
1 × 1,
16,
1 × 1 × 192
28 × 28 × 192
28 × 28 × 16
CONV
5 × 5,
32,
5 × 5 × 16
28 × 28 × 32
28x28x16 x 192 = 2.4M
28x28x32 x 5x5x16 = 10.0M
12.4M
which is 12.4 million multiplications.
Andrew Ng

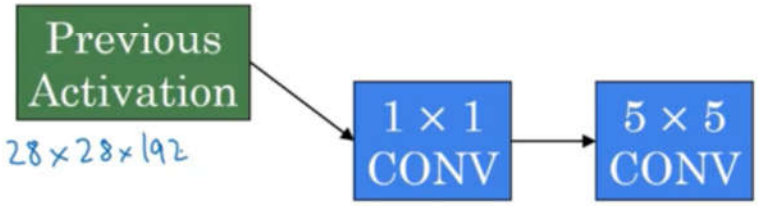

coursera.org/learn/convolutional-neural-networks/supplement/NTidE/inception-network-motivation-correction
coursera
Srinivasa Rao Dhanikonda
Convolutional Neural Networks › Week 2 › Inception Network Motivation *CORRECTION*
Prev | Next
Case studies
Video: Why look at case studies?
3 min
Video: Classic Networks
18 min
Video: ResNets
7 min
Video: Why ResNets Work
9 min
Video: Networks in Networks and 1x1 Convolutions
6 min
Video: Inception Network Motivation
10 min
Reading: Inception Network Motivation *CORRECTION*
1 min
Video: Inception Network
8 min
At 3:00, Andrew should have said 28 x 28 x 192 instead of 28 x 28 x 129. The subtitles have been corrected.
Mark as completed

deeplearning.ai

Case Studies

Inception network

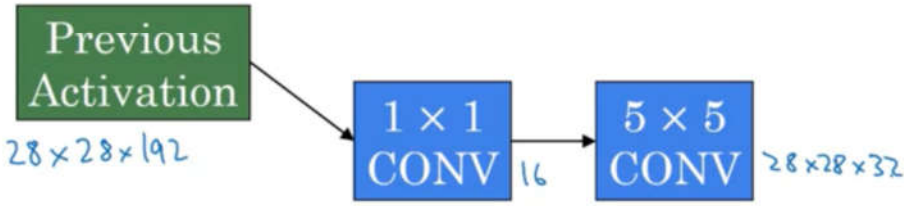
Inception module



The example we worked through in depth was the 1 by 1 followed by 5 by 5.

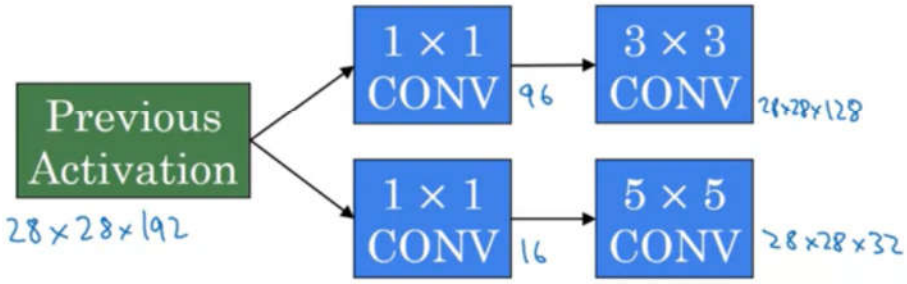
Andrew Ng

Inception module

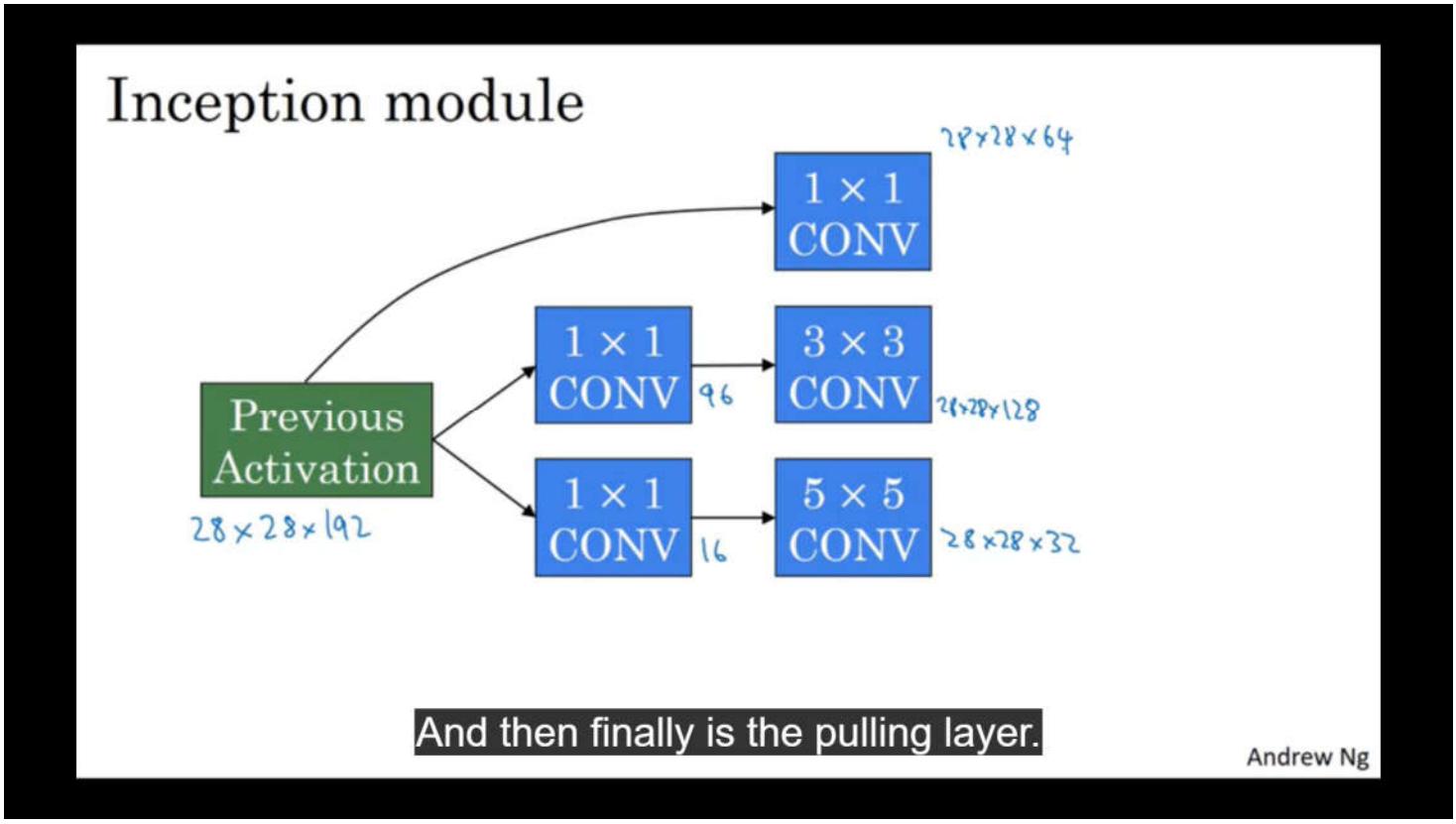


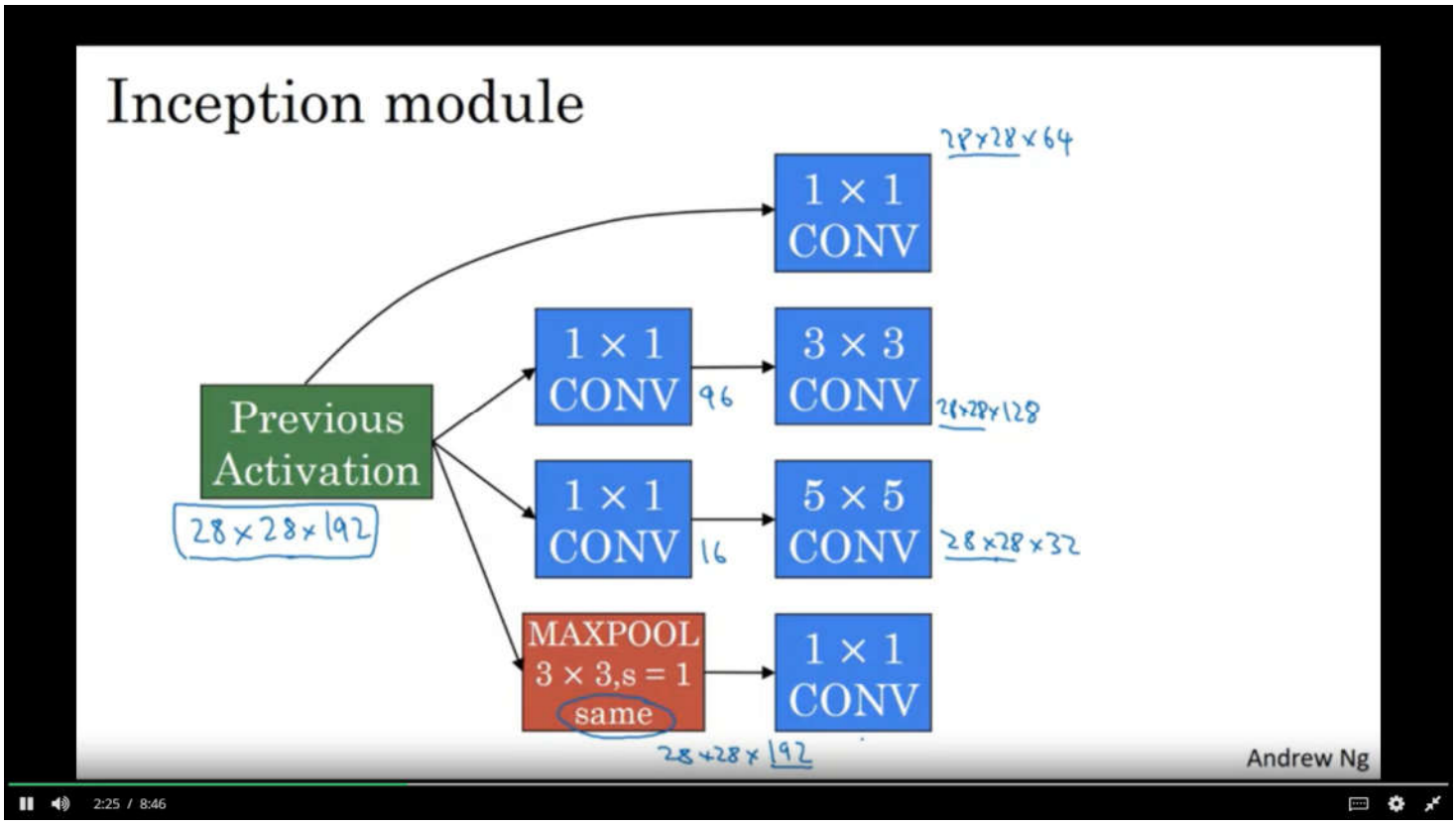
Andrew Ng

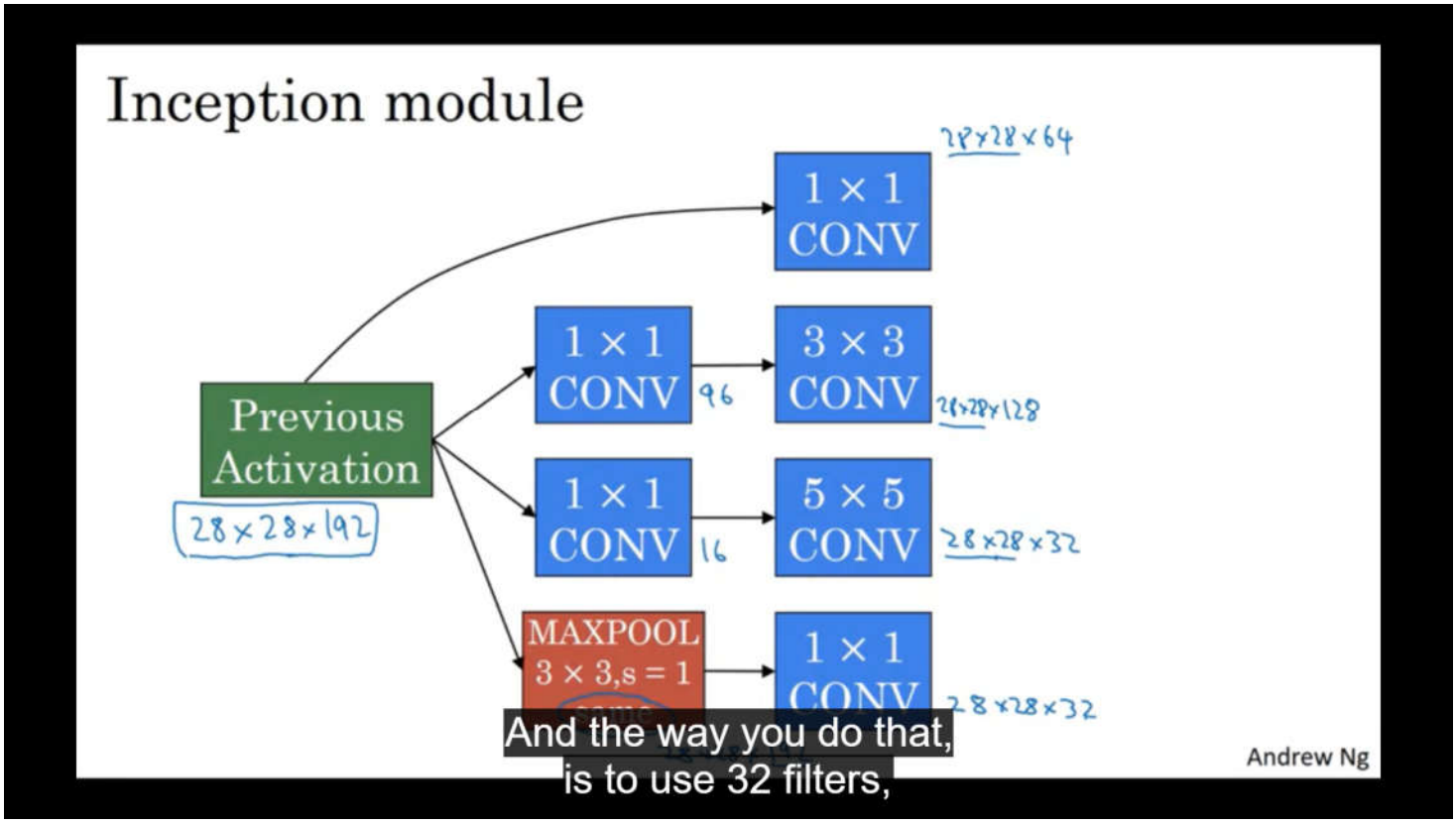
Inception module



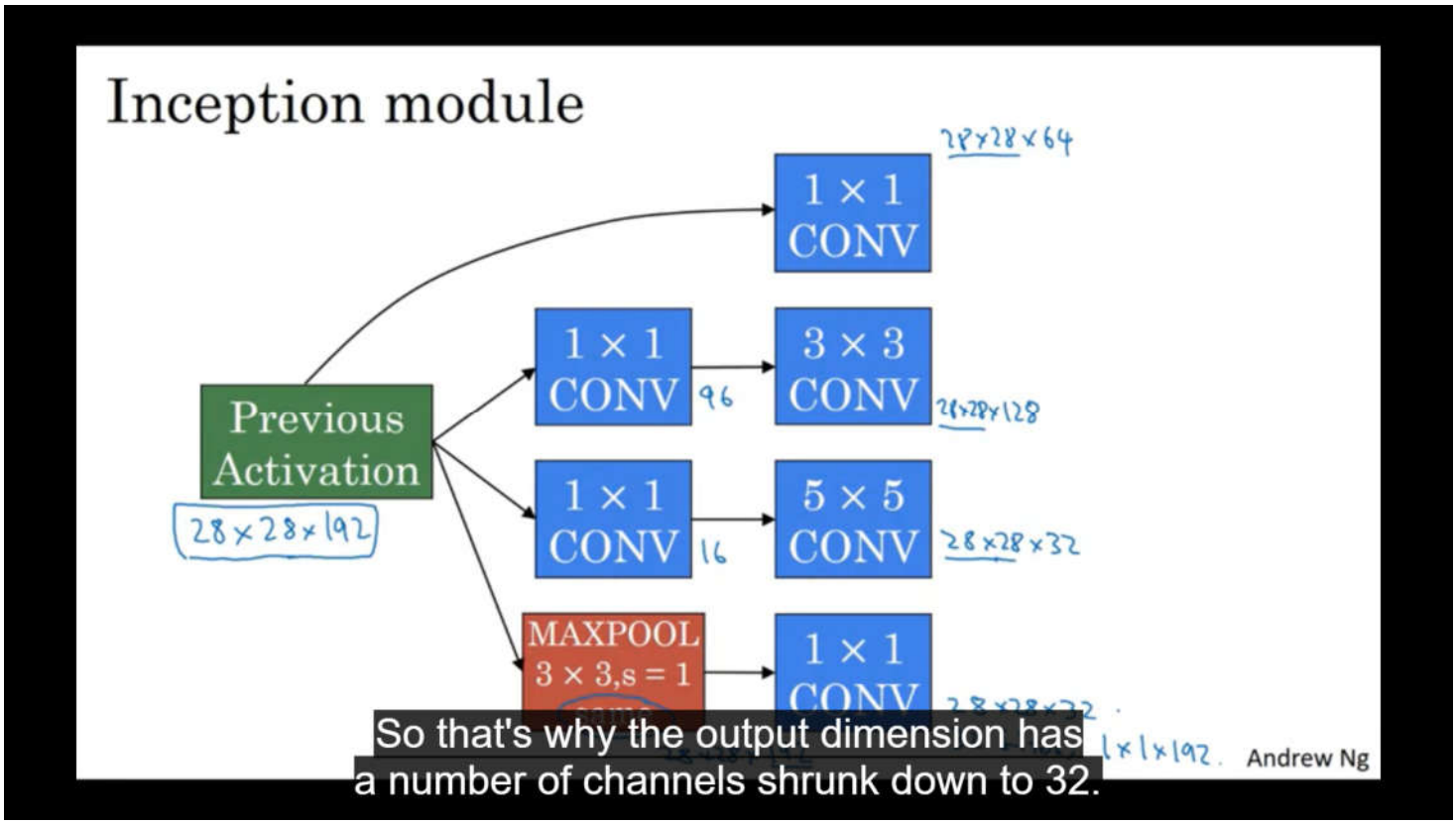
Andrew Ng


Inception module
28×28×64
1 × 1 CONV
1 × 1 CONV
96
3 × 3 CONV
28×28×128
Previous Activation
28×28×192
1 × 1 CONV
16
5 × 5 CONV
28×28×32
And then finally is the pulling layer.
Andrew Ng

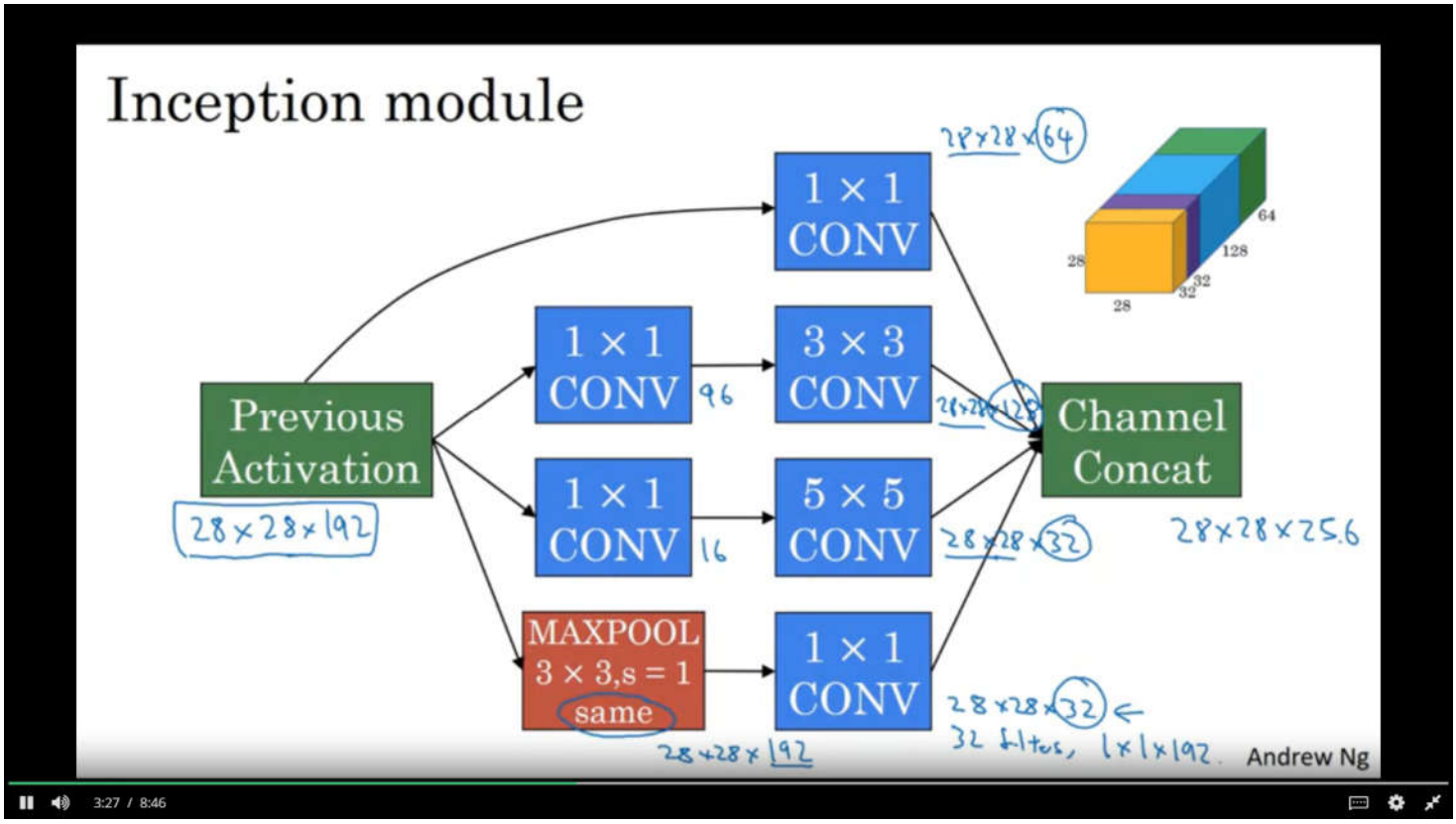

Inception module
28×28×64
1 × 1 CONV
1 × 1 CONV
96
3 × 3 CONV
28×28×128
Previous Activation
28×28×192
1 × 1 CONV
16
5 × 5 CONV
28×28×32
MAXPOOL
3 × 3,s = 1
same
28×28×192
1 × 1 CONV
Andrew Ng
2:25 / 8:46

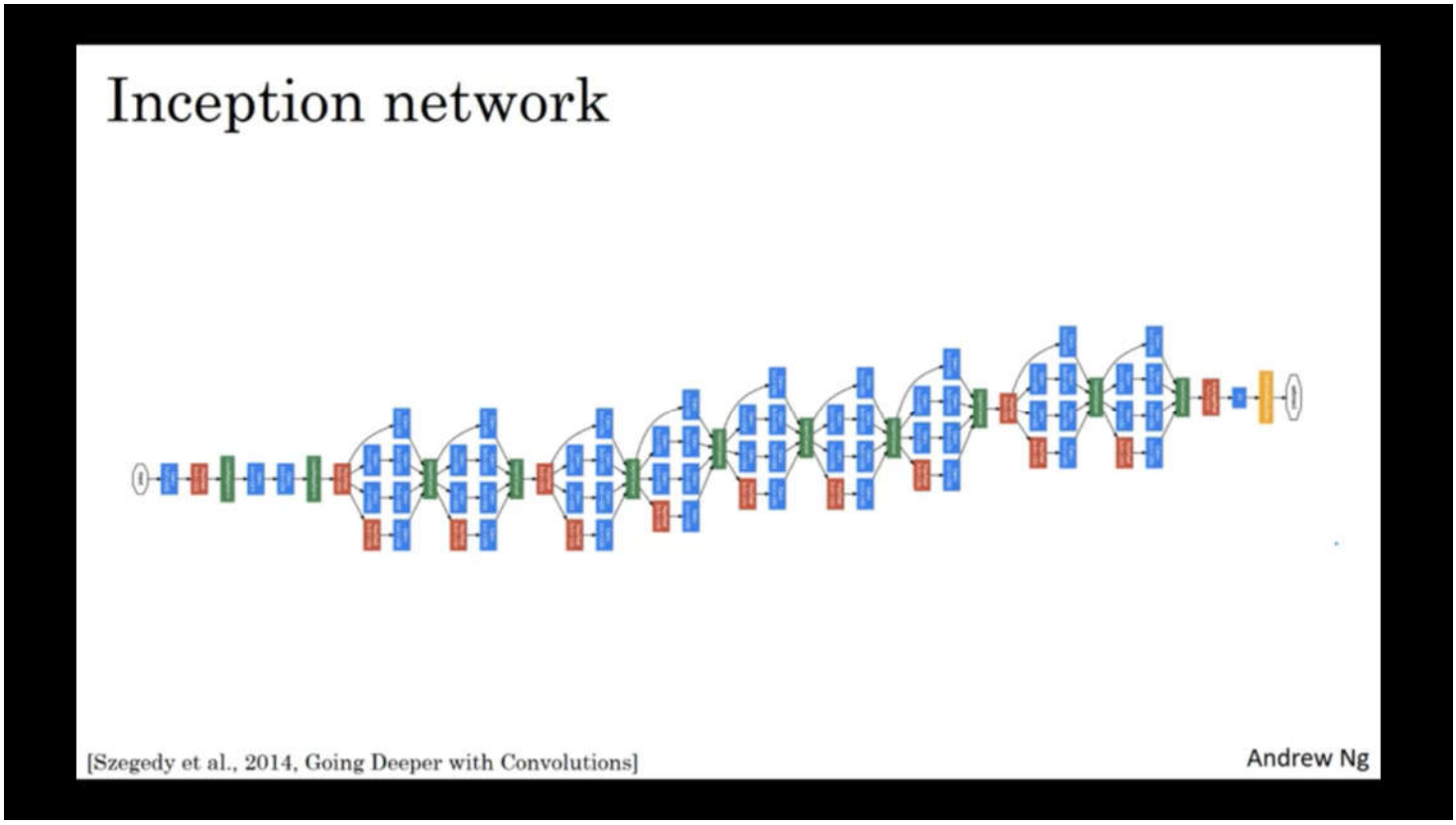


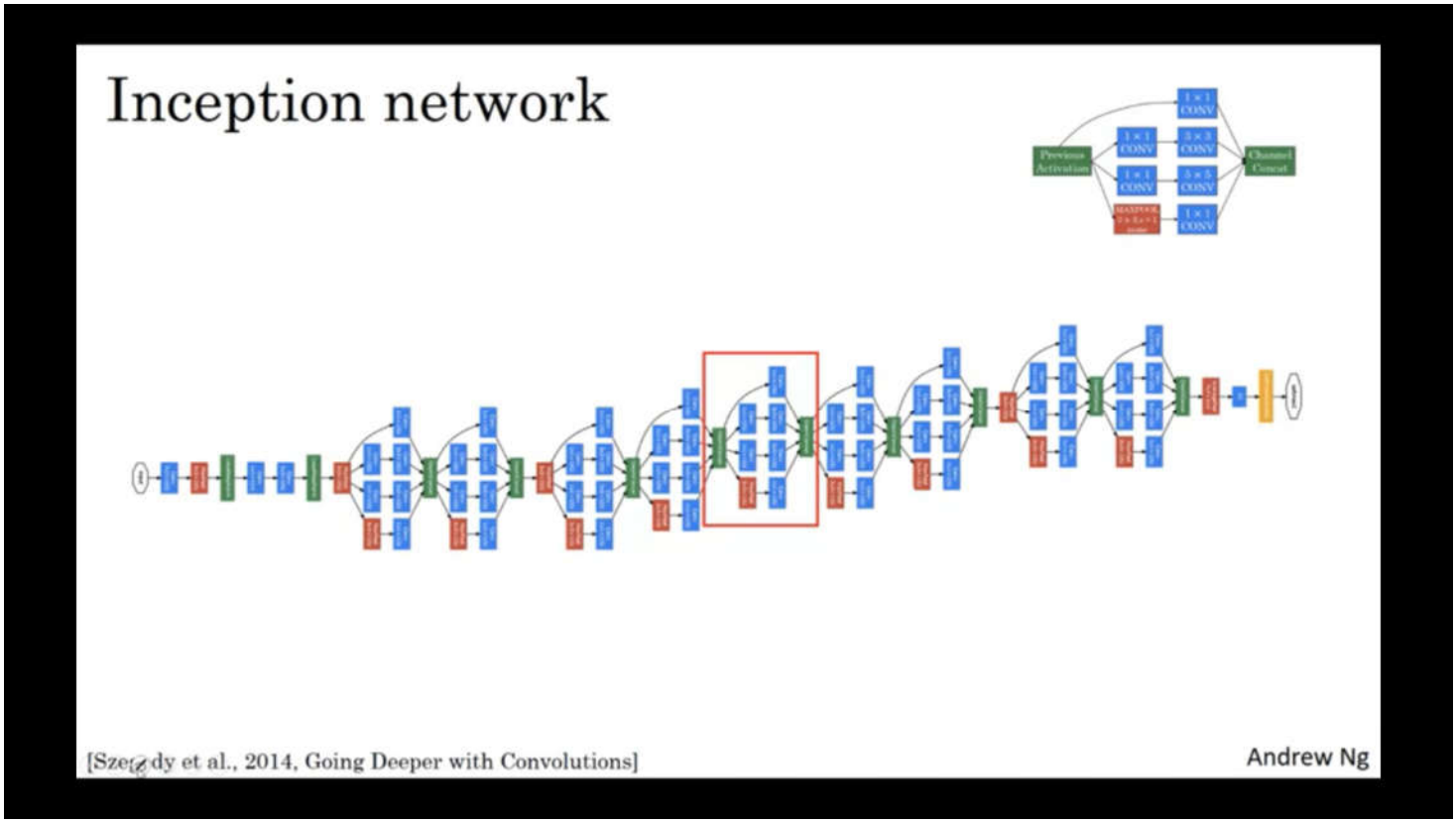
And the way you do that,
is to use 32 filters,

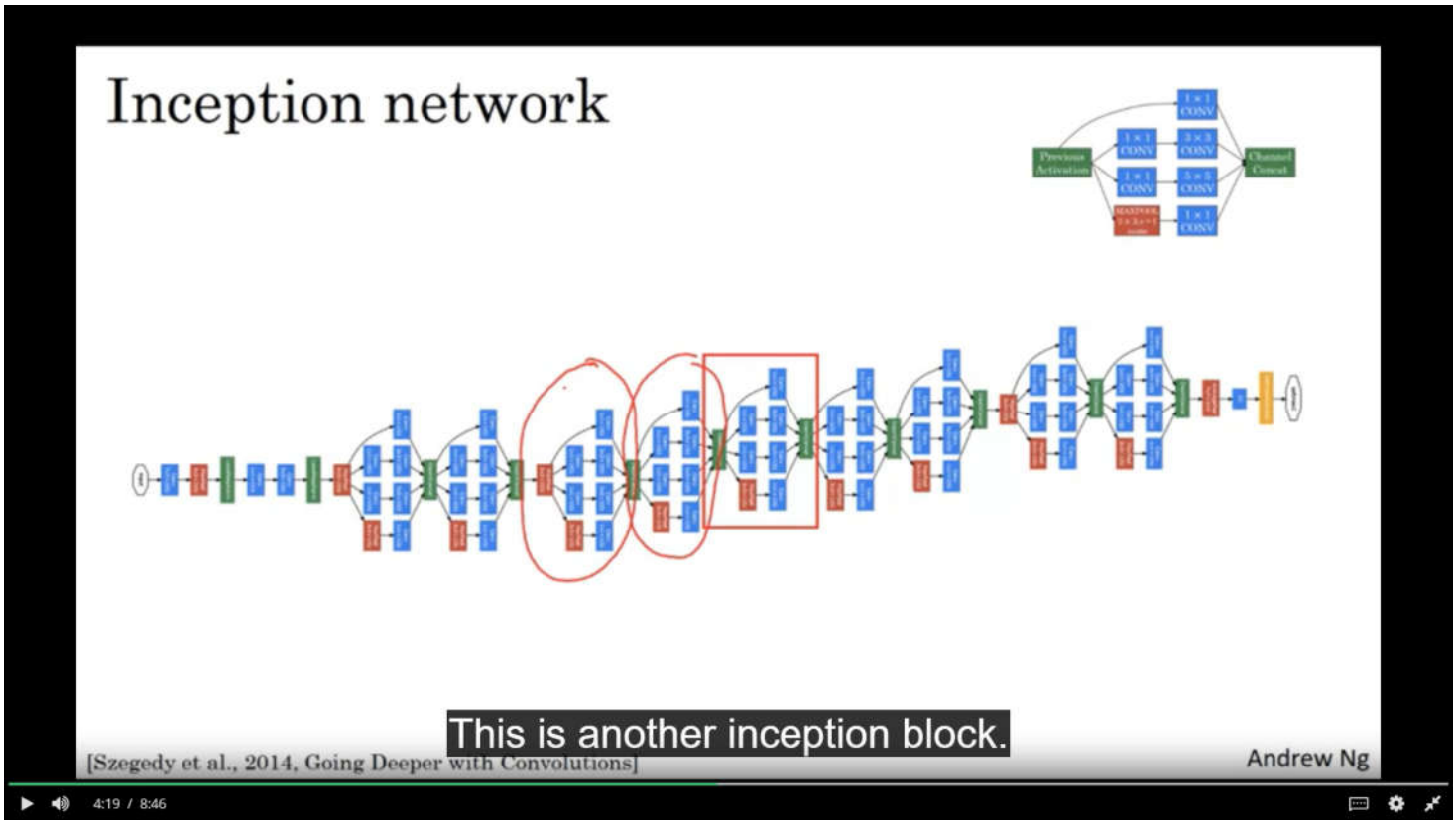


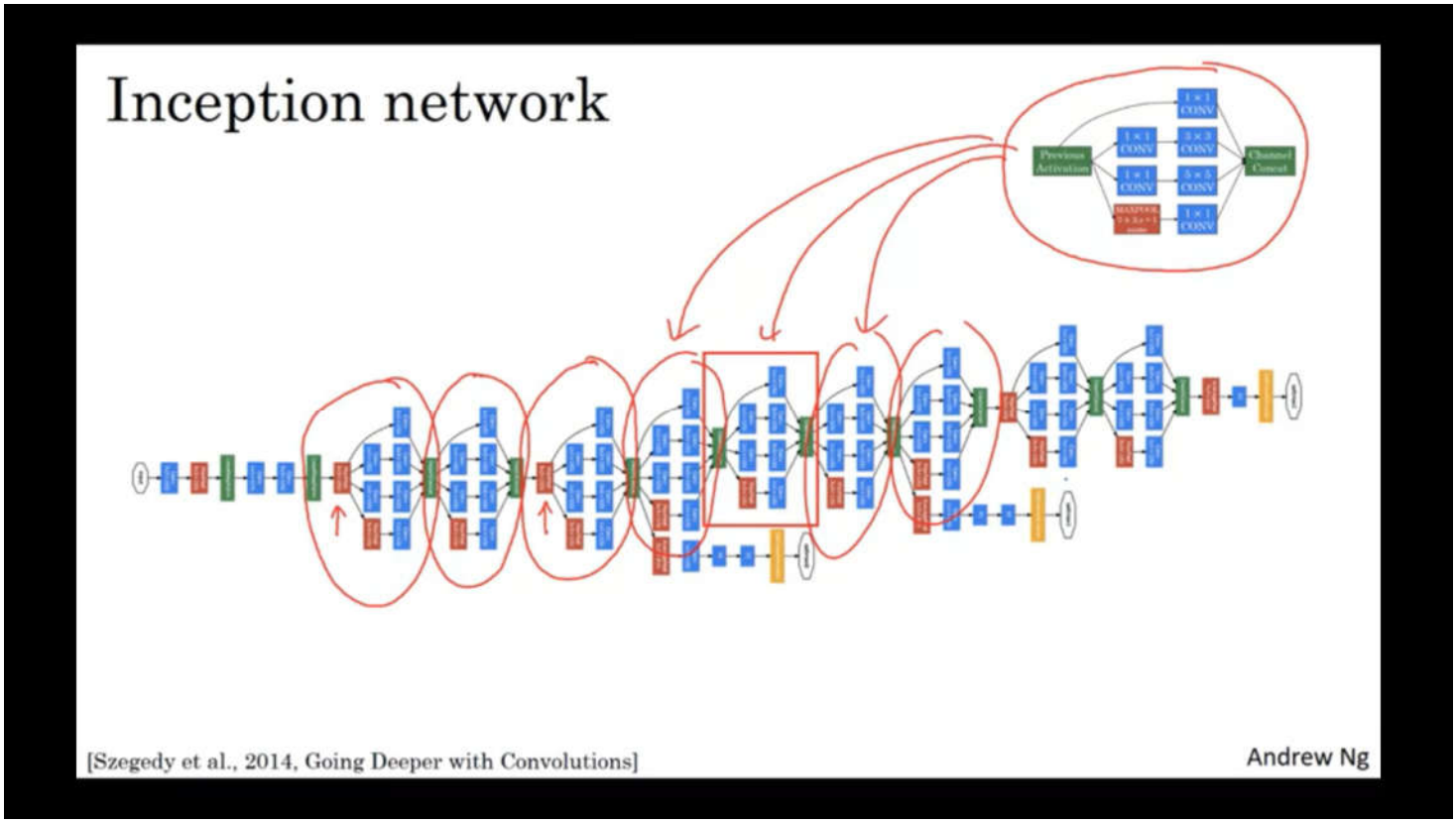
So that's why the output dimension has
a number of channels shrunk down to 32.

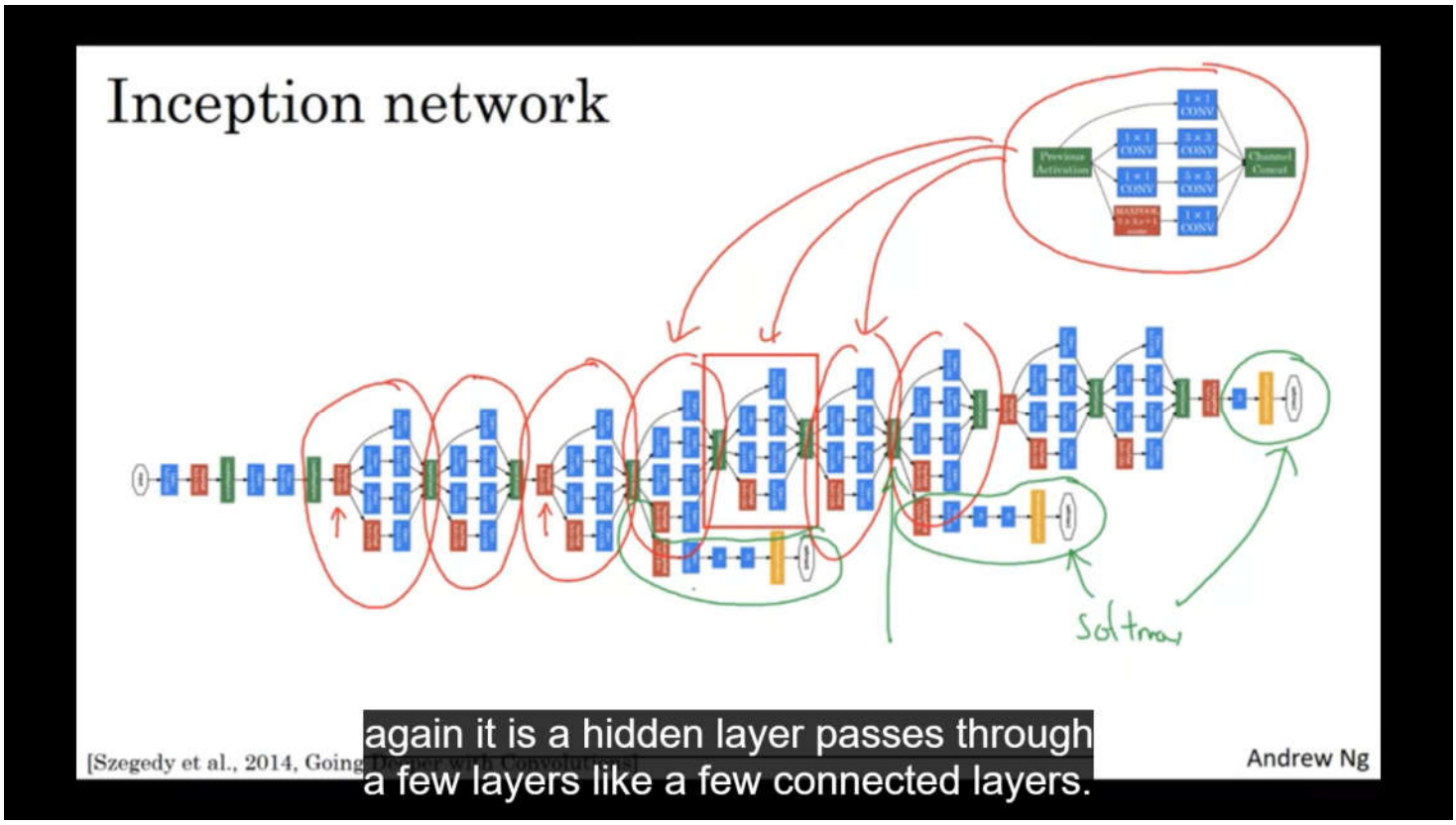

Inception module
Previous Activation
28×28×192
1 × 1 CONV
1 × 1 CONV
96
3 × 3 CONV
1 × 1 CONV
16
5 × 5 CONV
MAXPOOL 3 × 3,s = 1 same
28×28×192
1 × 1 CONV
28×28×64
28×28×128
28×28×32
28×28×32
32 filters, 1×1×192.
Channel Concat
28×28×256
64
128
32
32
28
28
Andrew Ng
3:27 / 8:46


Inception network
[Szegedy et al., 2014, Going Deeper with Convolutions]
Andrew Ng

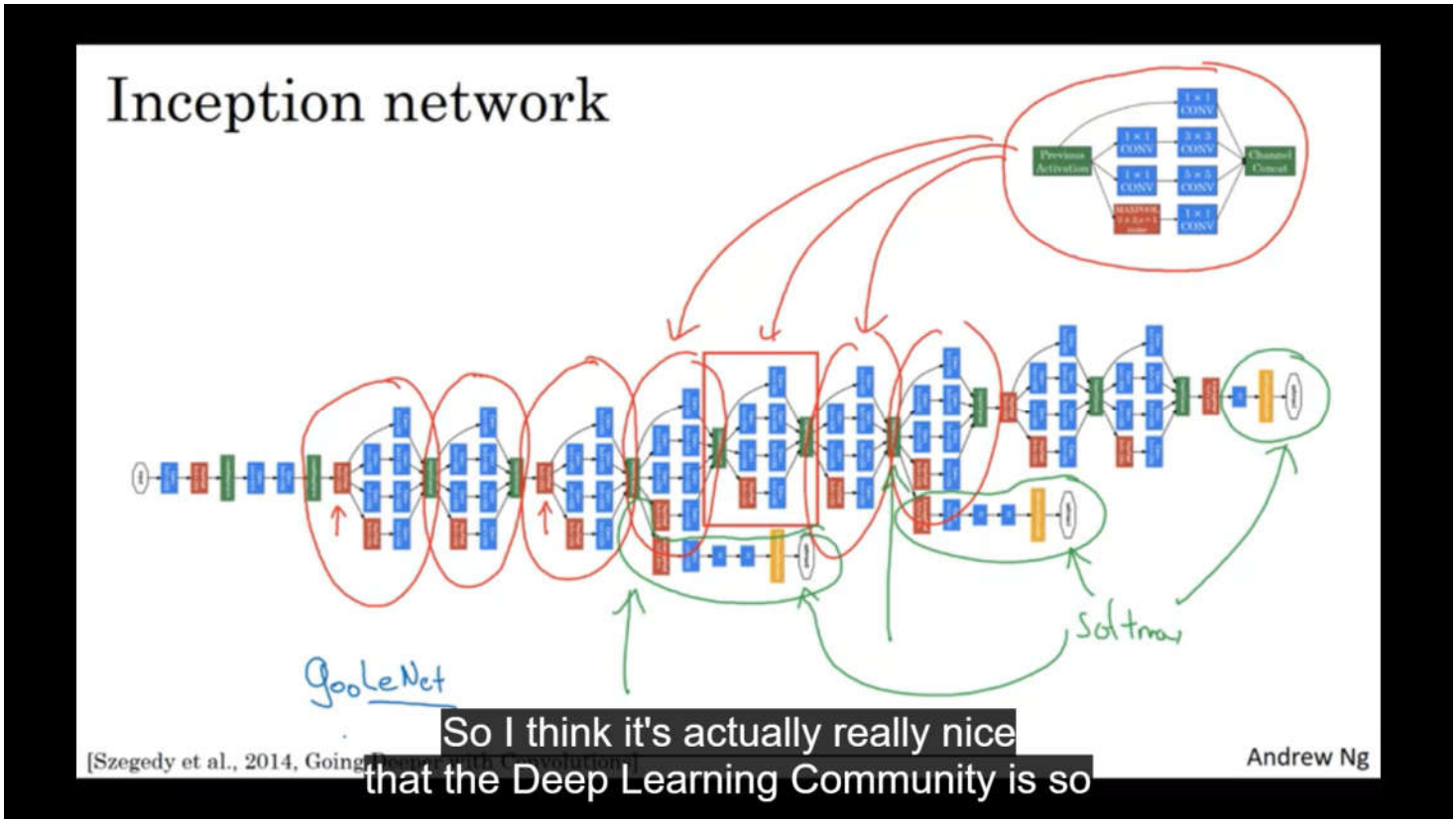

Inception network
[Szegedy et al., 2014, Going Deeper with Convolutions]
Andrew Ng


Inception network
This is another inception block.
[Szegedy et al., 2014, Going Deeper with Convolutions]
Andrew Ng
4:19 / 8:46



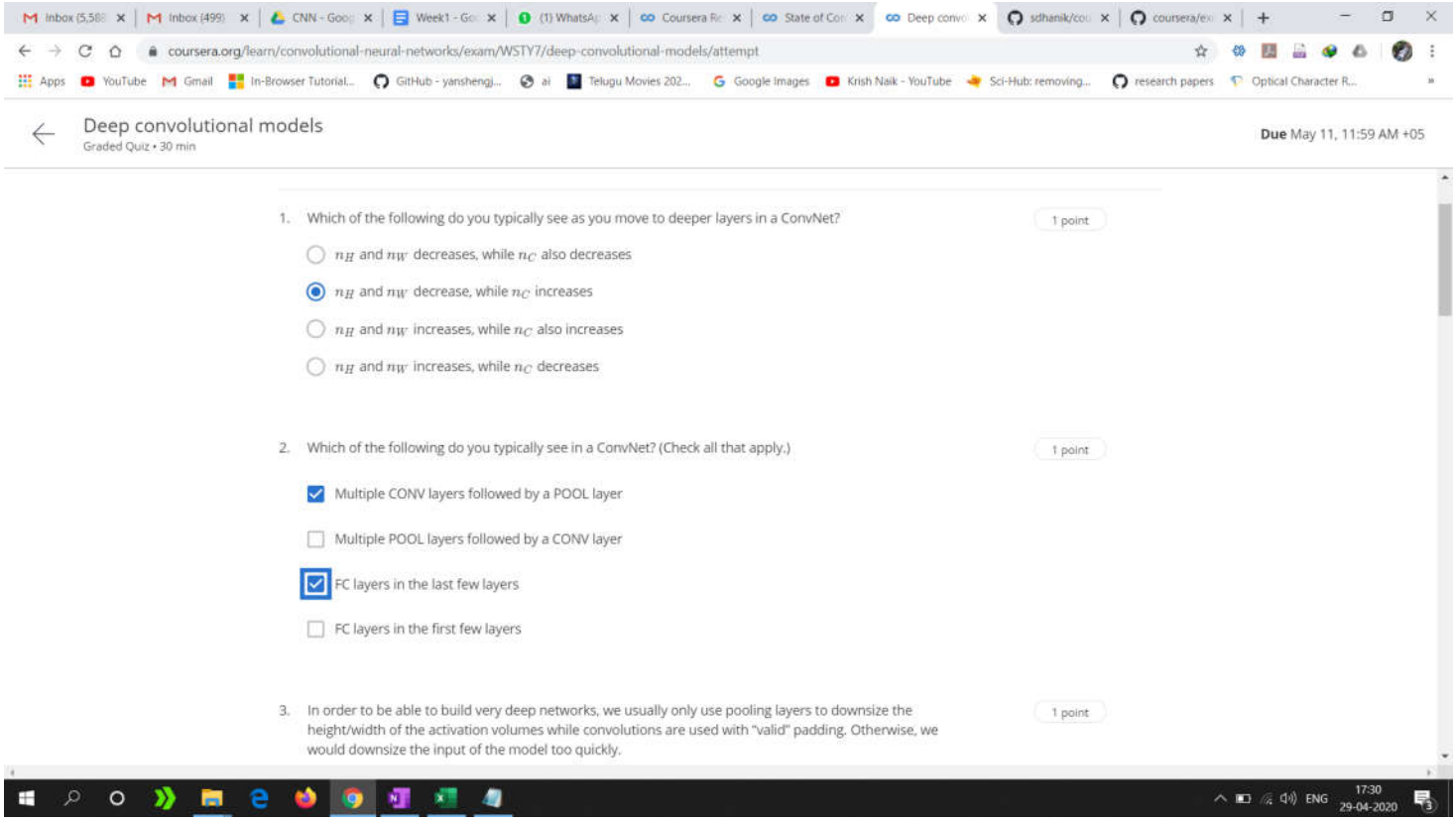


again it is a hidden layer passes through
a few layers like a few connected layers.



So I think it's actually really nice
that the Deep Learning Community is so

The inception paper actually cites
this meme for we need to go deeper.



coursera.org/learn/convolutional-neural-networks/exam/W5TY7/deep-convolutional-models/attempt

Deep convolutional models
Graded Quiz • 30 min

Due May 11, 11:59 AM +05

1. Which of the following do you typically see as you move to deeper layers in a ConvNet? (1 point)
   - ○ n_H and n_W decreases, while n_C also decreases
   - ◉ n_H and n_W decrease, while n_C increases
   - ○ n_H and n_W increases, while n_C also increases
   - ○ n_H and n_W increases, while n_C decreases

2. Which of the following do you typically see in a ConvNet? (Check all that apply.) (1 point)
   - [x] Multiple CONV layers followed by a POOL layer
   - [ ] Multiple POOL layers followed by a CONV layer
   - [x] FC layers in the last few layers
   - [ ] FC layers in the first few layers

3. In order to be able to build very deep networks, we usually only use pooling layers to downsize the height/width of the activation volumes while convolutions are used with "valid" padding. Otherwise, we would downsize the input of the model too quickly. (1 point)

coursera.org/learn/convolutional-neural-networks/exam/WSTY7/deep-convolutional-models/attempt

Deep convolutional models

Graded Quiz • 30 min

Due May 11, 11:59 AM +05

3. In order to be able to build very deep networks, we usually only use pooling layers to downsize the height/width of the activation volumes while convolutions are used with "valid" padding. Otherwise, we would downsize the input of the model too quickly. (1 point)

- ○ True
- ◉ False

4. Training a deeper network (for example, adding additional layers to the network) allows the network to fit more complex functions and thus almost always results in lower training error. For this question, assume we're referring to "plain" networks. (1 point)

- ○ True
- ◉ False

5. The following equation captures the computation in a ResNet block. What goes into the two blanks above? (1 point)

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

- ○ $z^{[l]}$ and $a^{[l]}$, respectively
- ○ 0 and $a^{[l]}$, respectively
- ○ 0 and $z^{[l+1]}$, respectively
- ◉ $a^{[l]}$ and 0, respectively

17:31 29-04-2020

coursera.org/learn/convolutional-neural-networks/exam/WSTY7/deep-convolutional-models/attempt

Deep convolutional models

Graded Quiz • 30 min

Due May 11, 11:59 AM +05

6. Which ones of the following statements on Residual Networks are true? (Check all that apply.) (1 point)

- [x] Using a skip-connection helps the gradient to backpropagate and thus helps you to train deeper networks
- [ ] The skip-connection makes it easy for the network to learn an identity mapping between the input and the output within the ResNet block.
- [ ] A ResNet with L layers would have on the order of L^2 skip connections in total.
- [x] The skip-connections compute a complex non-linear function of the input to pass to a deeper layer in the network.

7. Suppose you have an input volume of dimension 64x64x16. How many parameters would a single 1x1 convolutional filter have (including the bias)? (1 point)

- ○ 4097
- ○ 2
- ○ 1
- ◉ 17

17:32 29-04-2020

coursera.org/learn/convolutional-neural-networks/exam/WSTY7/deep-convolutional-models/attempt

Deep convolutional models

Graded Quiz • 30 min

Due May 11, 11:59 AM +05

8. Suppose you have an input volume of dimension $n_H \times n_W \times n_C$. Which of the following statements you agree with? (Assume that "1x1 convolutional layer" below always uses a stride of 1 and no padding.) — 1 point

- [ ] You can use a pooling layer to reduce n_H, n_W, but not n_C.
- [x] You can use a pooling layer to reduce n_H, n_W, and n_C.
- [ ] You can use a 1x1 convolutional layer to reduce n_H, n_W, and n_C.
- [x] You can use a 1x1 convolutional layer to reduce n_C but not n_H, n_W.

9. Which ones of the following statements on Inception Networks are true? (Check all that apply.) — 1 point

- [ ] Making an inception network deeper (by stacking more inception blocks together) should not hurt training set performance.
- [x] Inception blocks usually use 1x1 convolutions to reduce the input data volume's size before applying 3x3 and 5x5 convolutions.
- [ ] Inception networks incorporates a variety of network architectures (similar to dropout, which randomly chooses a network architecture on each step) and thus has a similar regularizing effect as dropout.
- [x] A single inception block allows the network to use a combination of 1x1, 3x3, 5x5 convolutions and pooling.

coursera.org/learn/convolutional-neural-networks/exam/WSTY7/deep-convolutional-models/attempt

Deep convolutional models

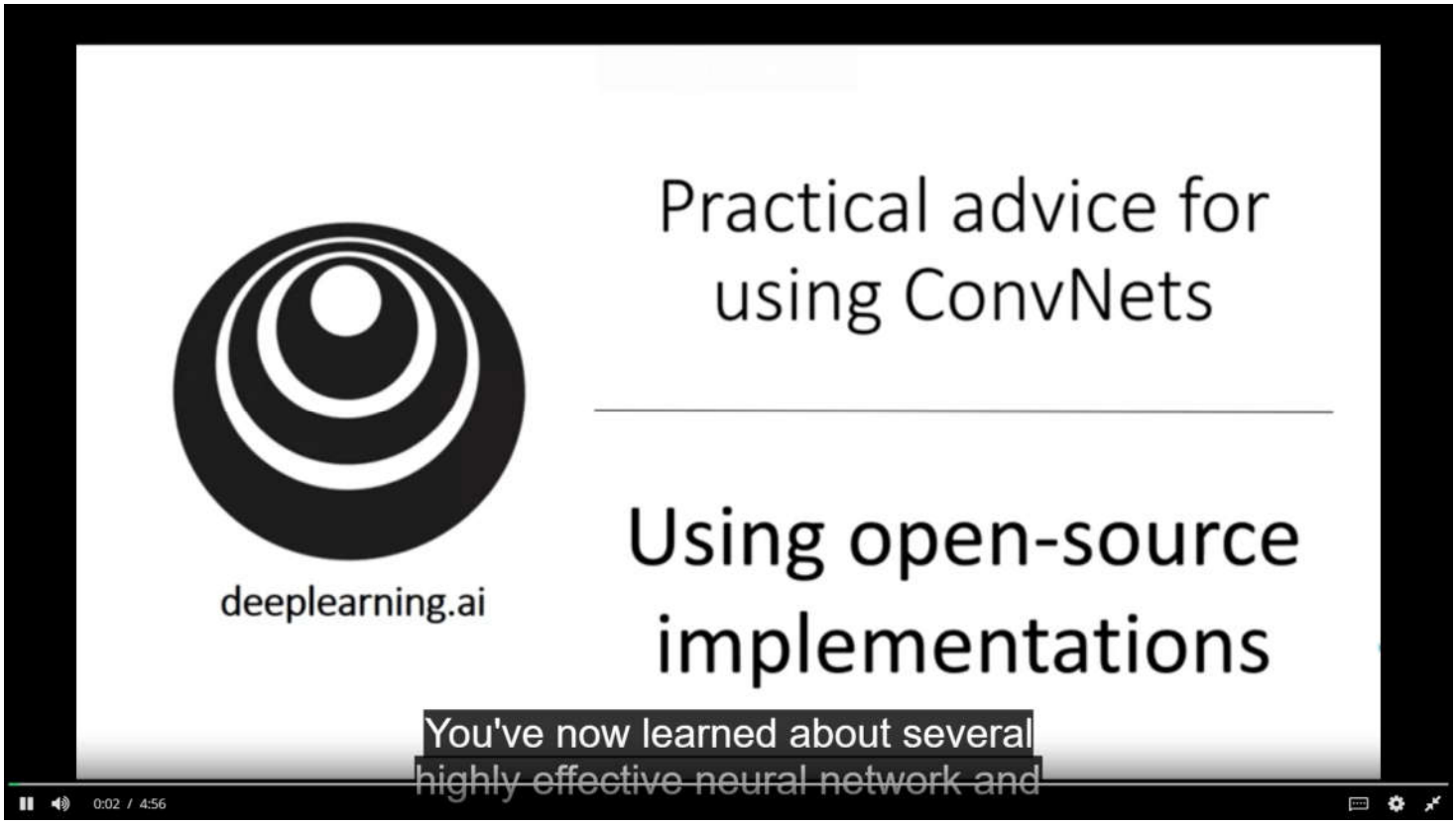
Graded Quiz • 30 min

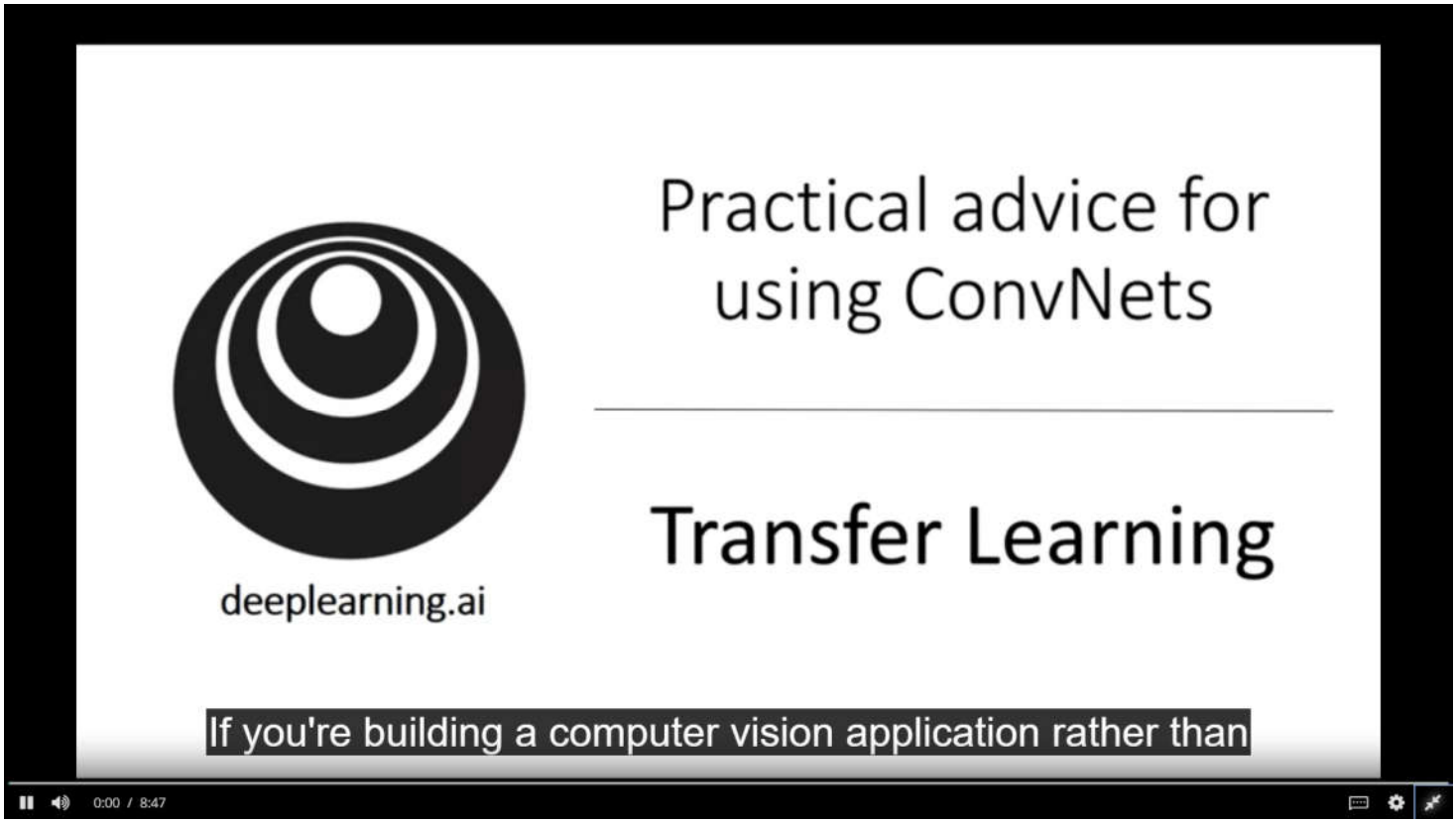
Due May 11, 11:59 AM +05

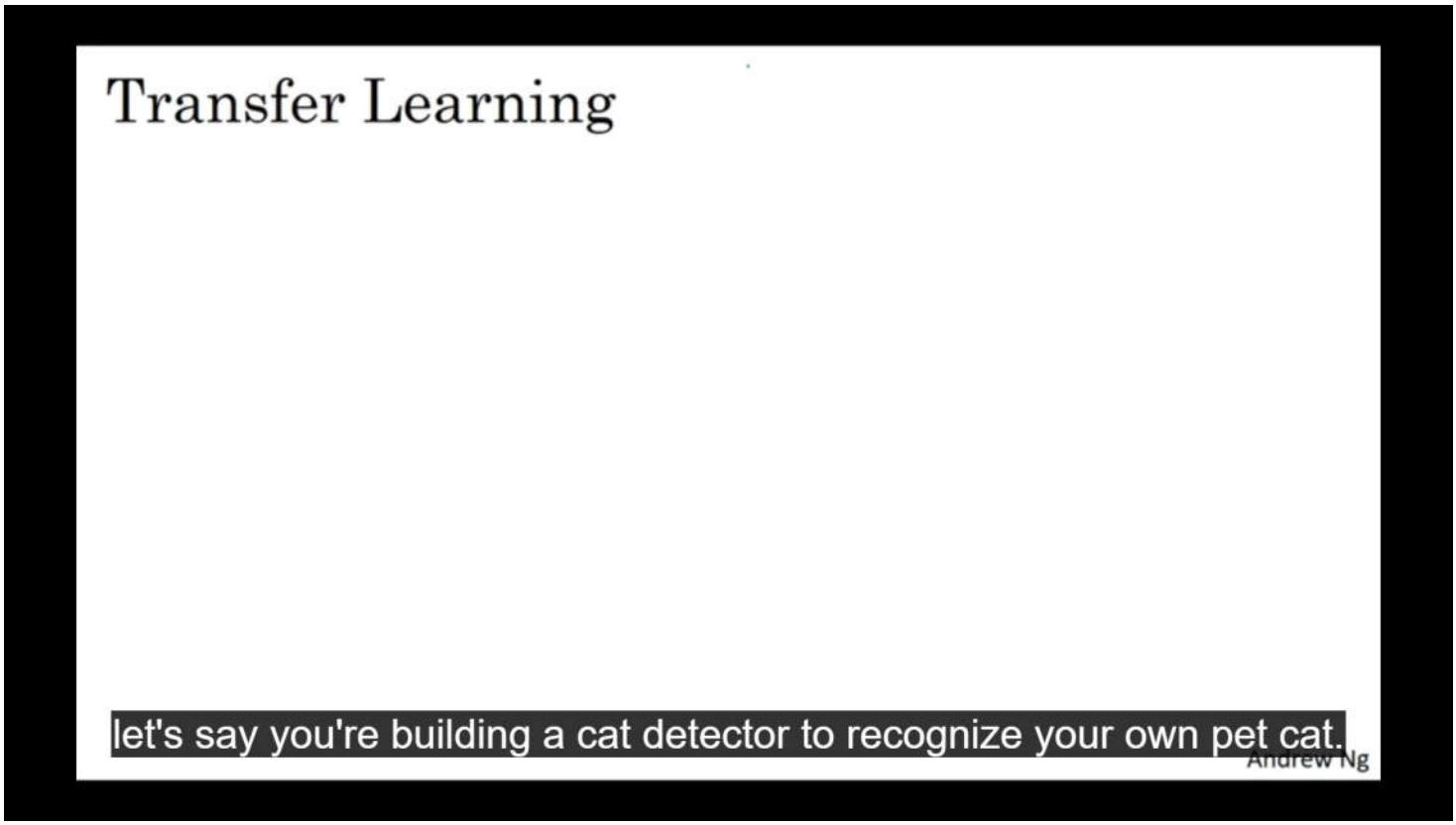
- [x] Inception blocks usually use 1x1 convolutions to reduce the input data volume's size before applying 3x3 and 5x5 convolutions.
- [ ] Inception networks incorporates a variety of network architectures (similar to dropout, which randomly chooses a network architecture on each step) and thus has a similar regularizing effect as dropout.
- [x] A single inception block allows the network to use a combination of 1x1, 3x3, 5x5 convolutions and pooling.

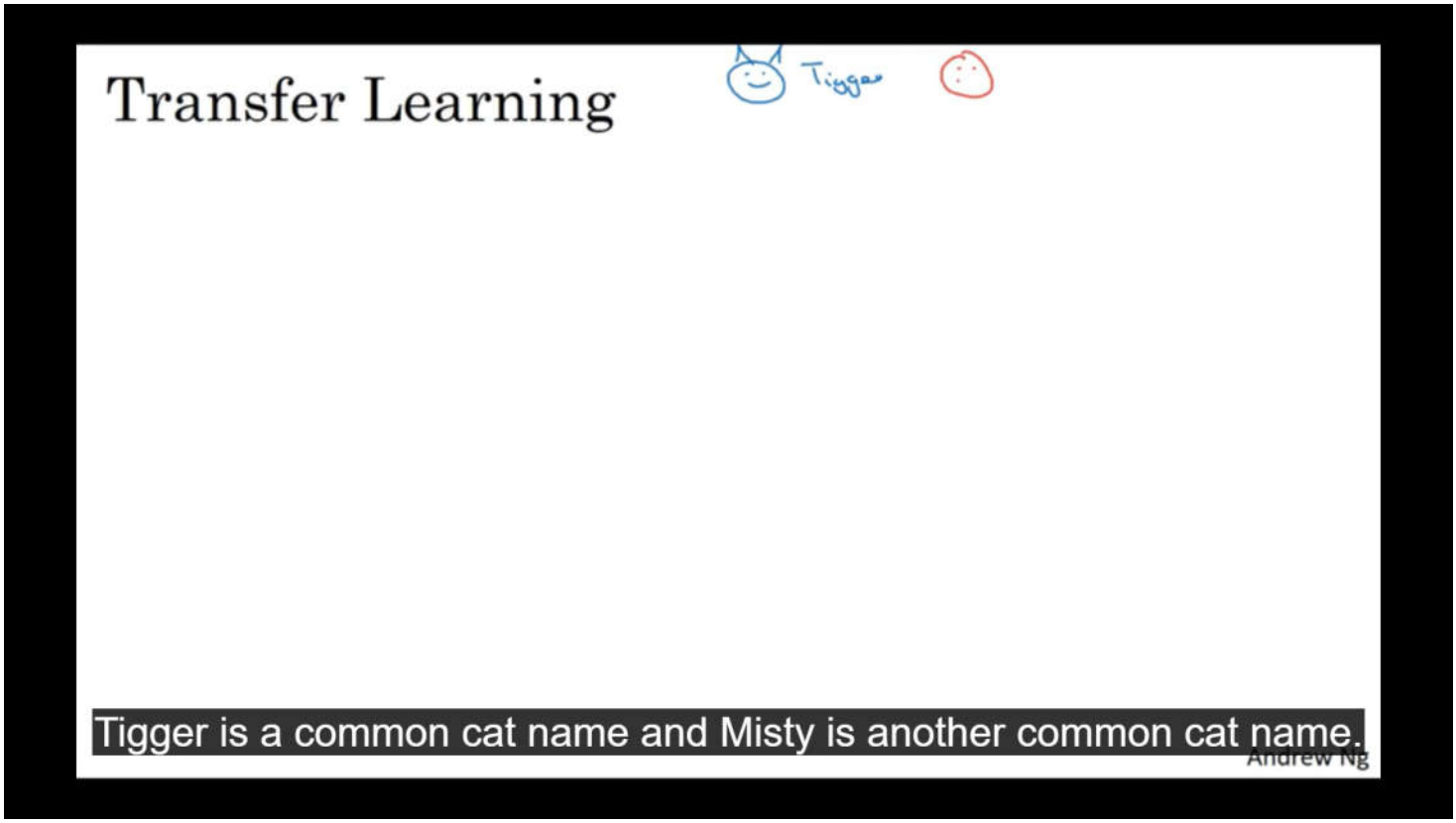
10. Which of the following are common reasons for using open-source implementations of ConvNets (both the model and/or weights)? Check all that apply. — 1 point

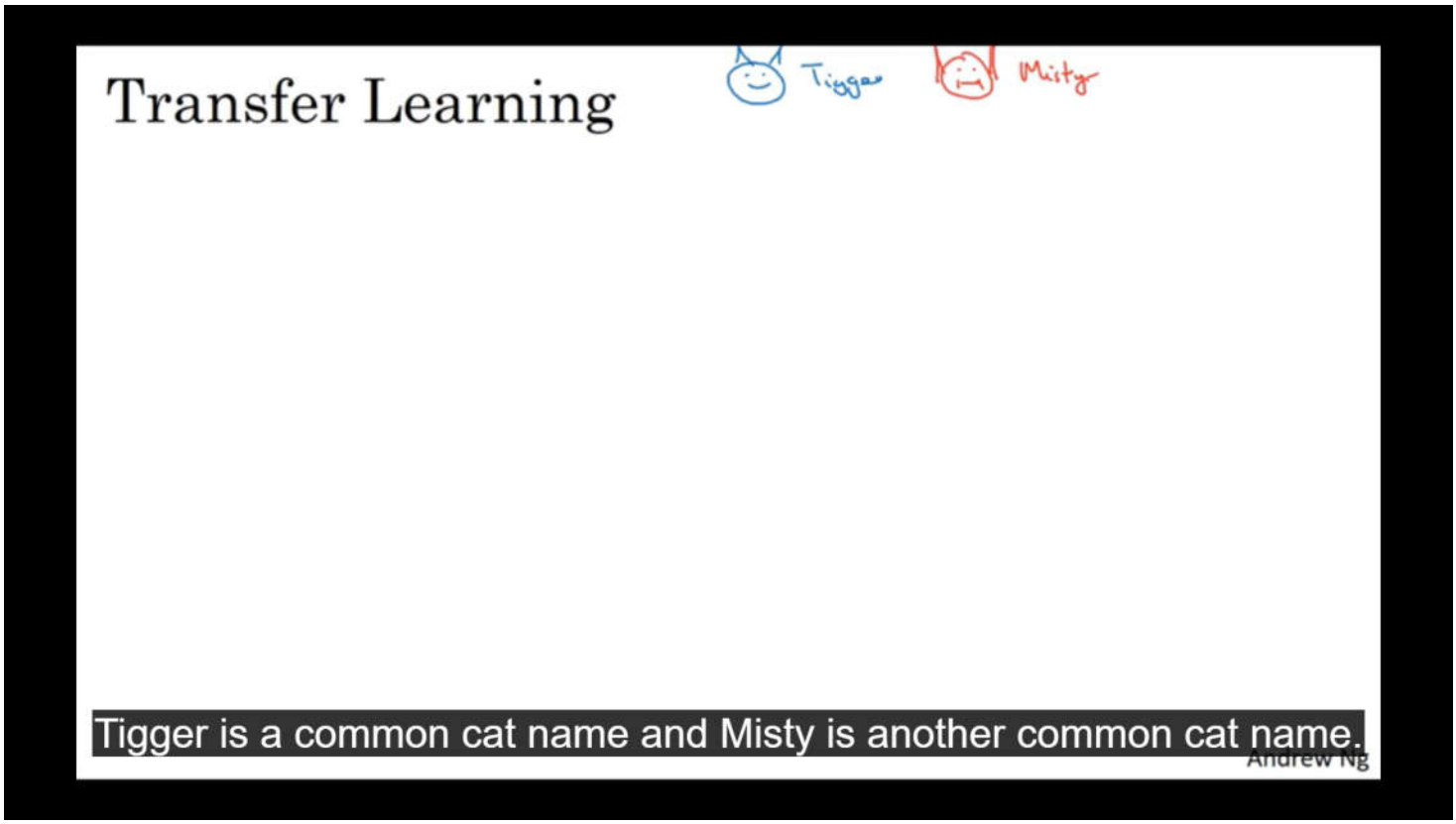
- [x] Parameters trained for one computer vision task are often useful as pretraining for other computer vision tasks.
- [x] It is a convenient way to get working an implementation of a complex ConvNet architecture.
- [ ] The same techniques for winning computer vision competitions, such as using multiple crops at test time, are widely used in practical deployments (or production system deployments) of ConvNets.
- [ ] A model trained for one computer vision task can usually be used to perform data augmentation even for a different computer vision task.

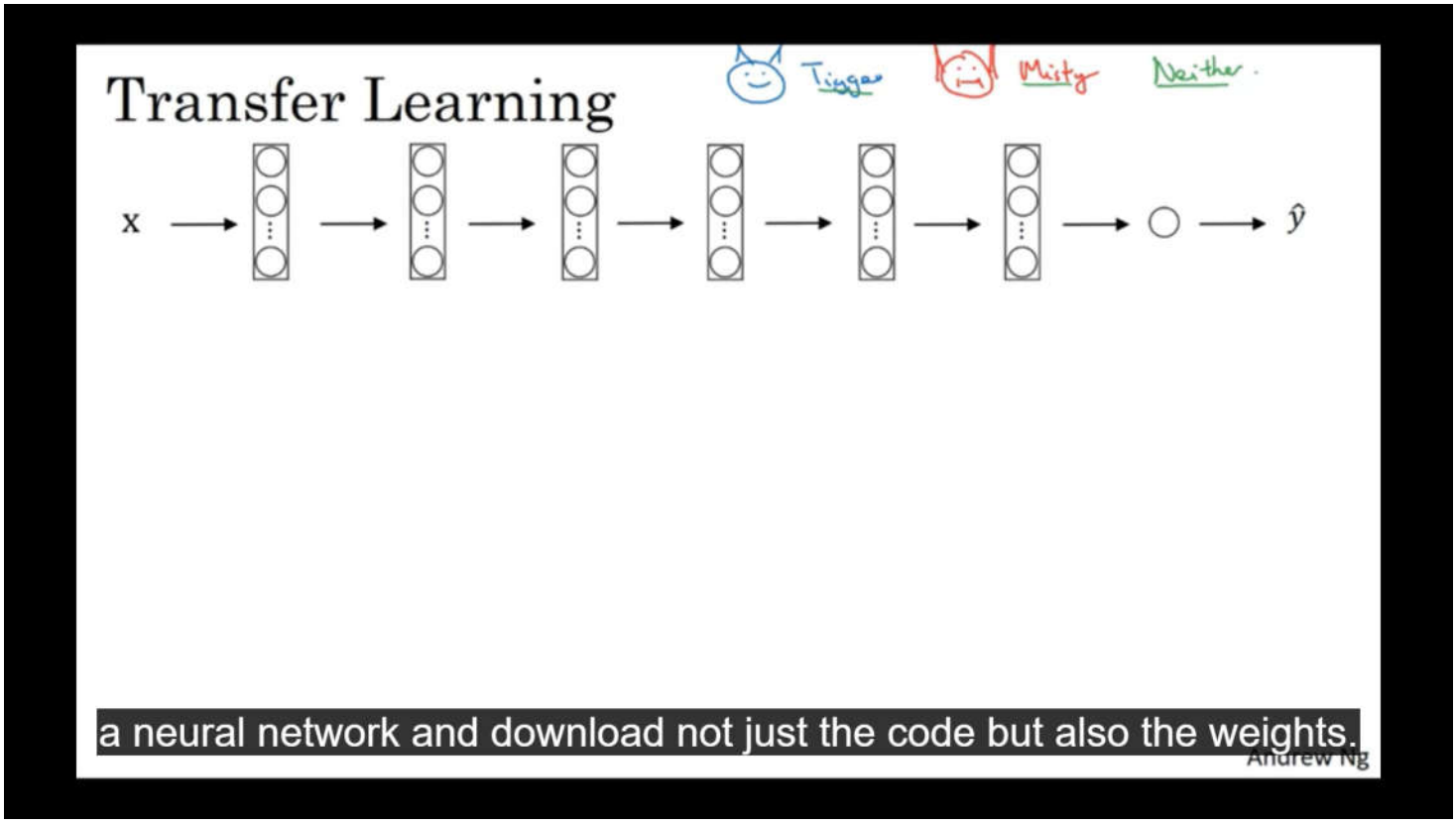

Practical advice for using ConvNets
Using open-source implementations
deeplearning.ai
You've now learned about several
highly effective neural network and
0:02 / 4:56

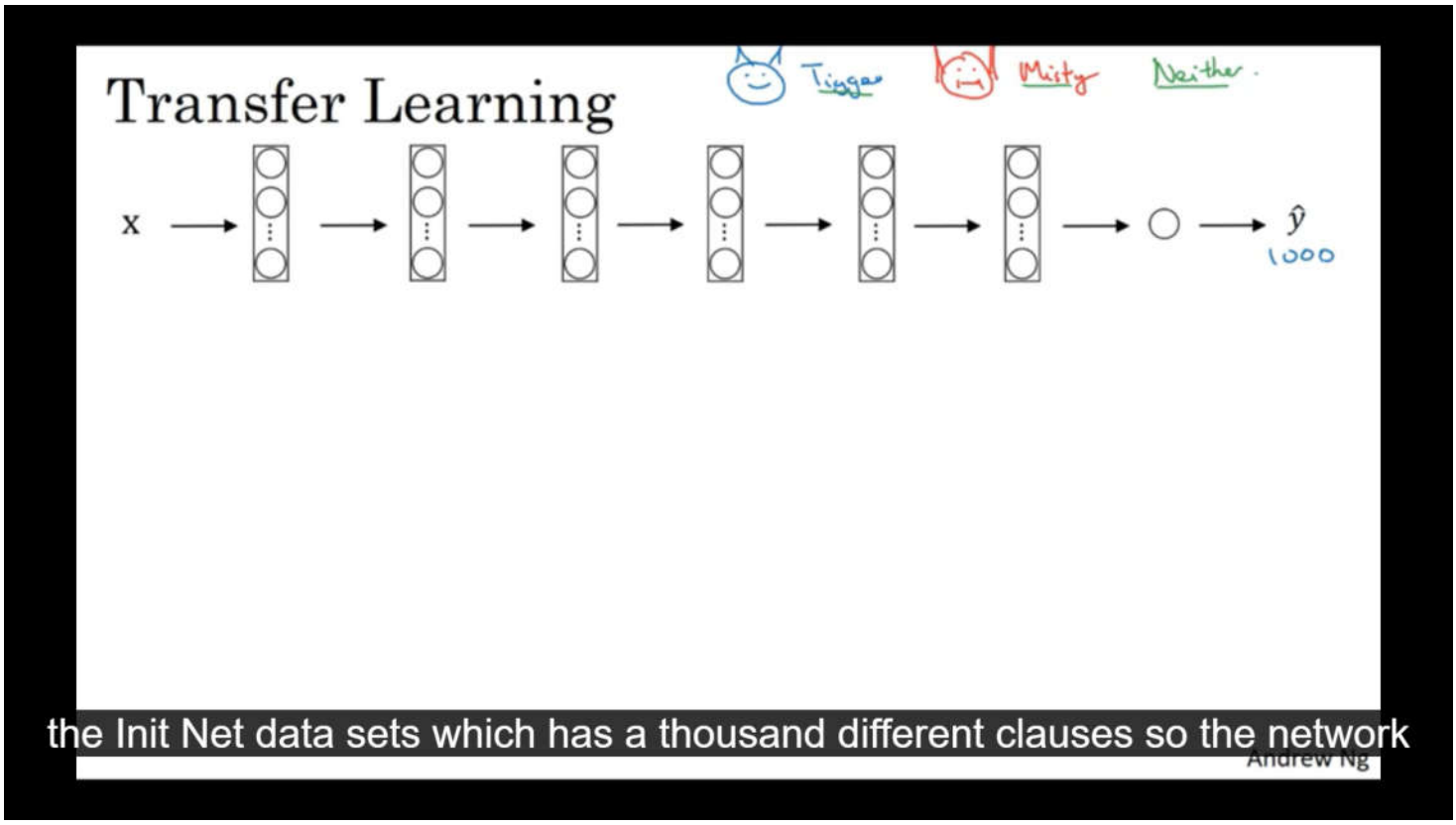

Practical advice for using ConvNets
Transfer Learning
deeplearning.ai
If you're building a computer vision application rather than
0:00 / 8:47

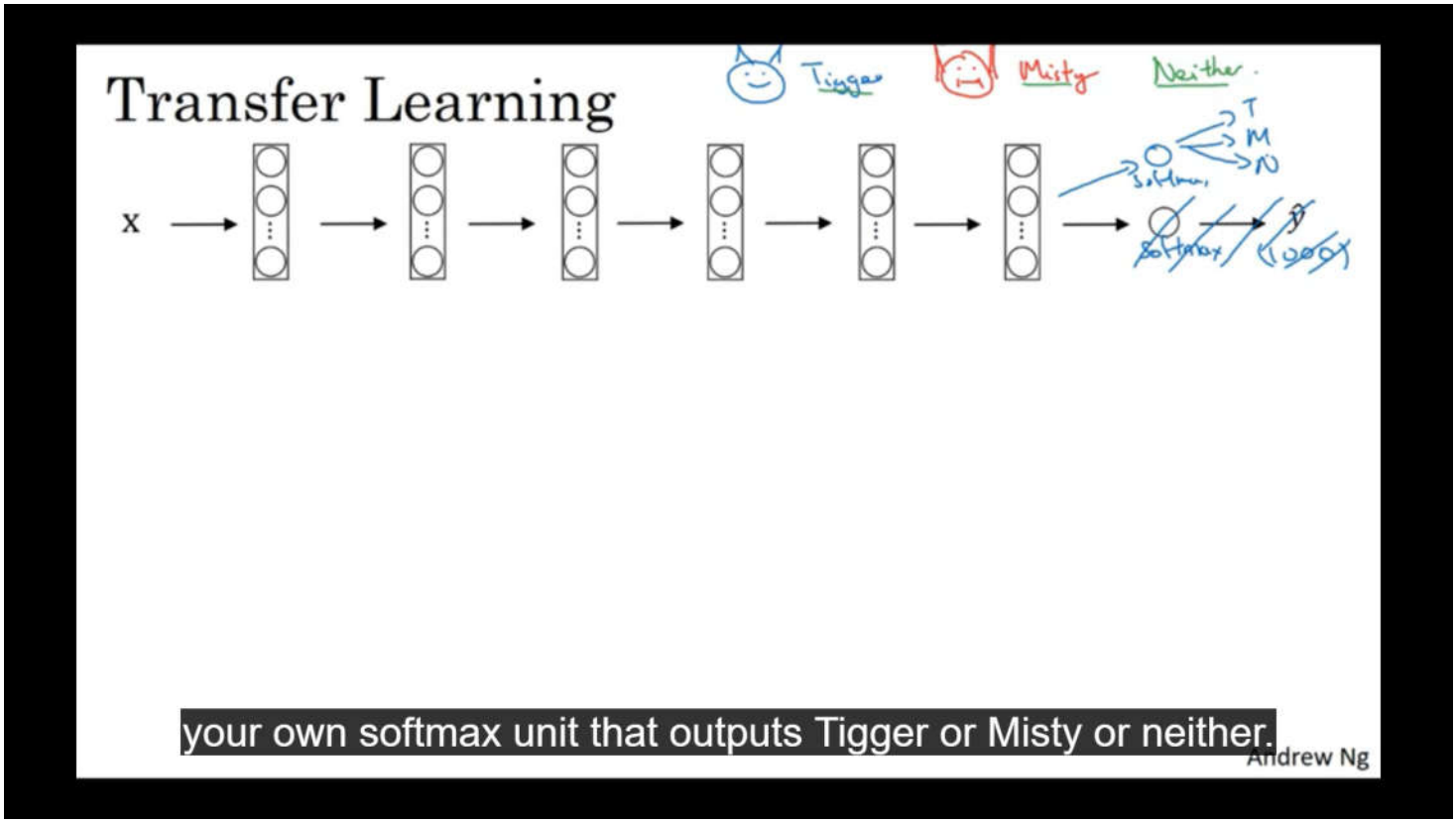

Transfer Learning
let's say you're building a cat detector to recognize your own pet cat.
Andrew Ng


Transfer Learning
Tigger
Tigger is a common cat name and Misty is another common cat name.
Andrew Ng


Transfer Learning
Tigger
Misty
Tigger is a common cat name and Misty is another common cat name.
Andrew Ng

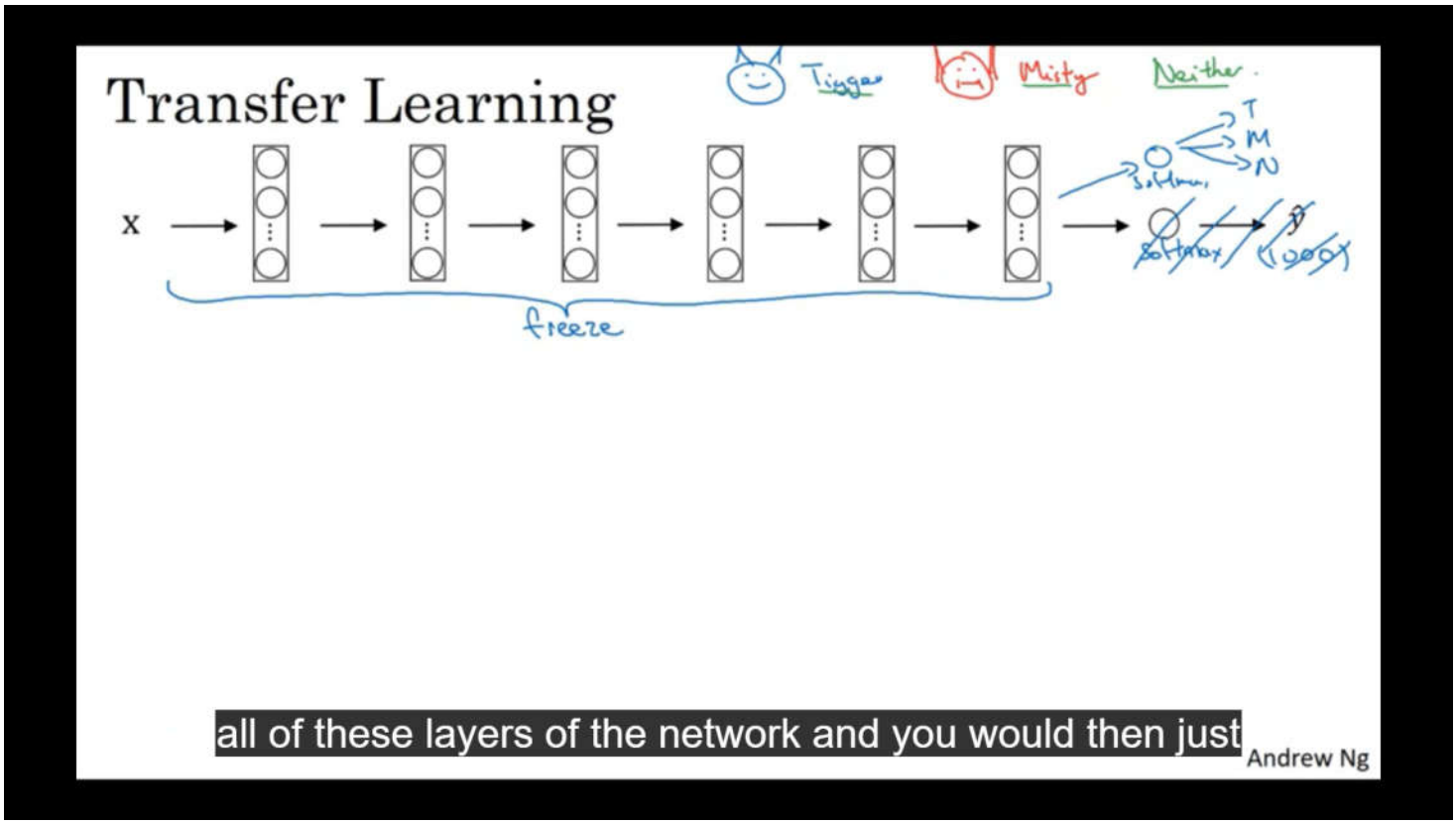

Transfer Learning
Tigger
Misty
Neither
x
ŷ
a neural network and download not just the code but also the weights.
Andrew Ng


Transfer Learning
Tigger
Misty
Neither
x
ŷ
1000
the Init Net data sets which has a thousand different clauses so the network
Andrew Ng



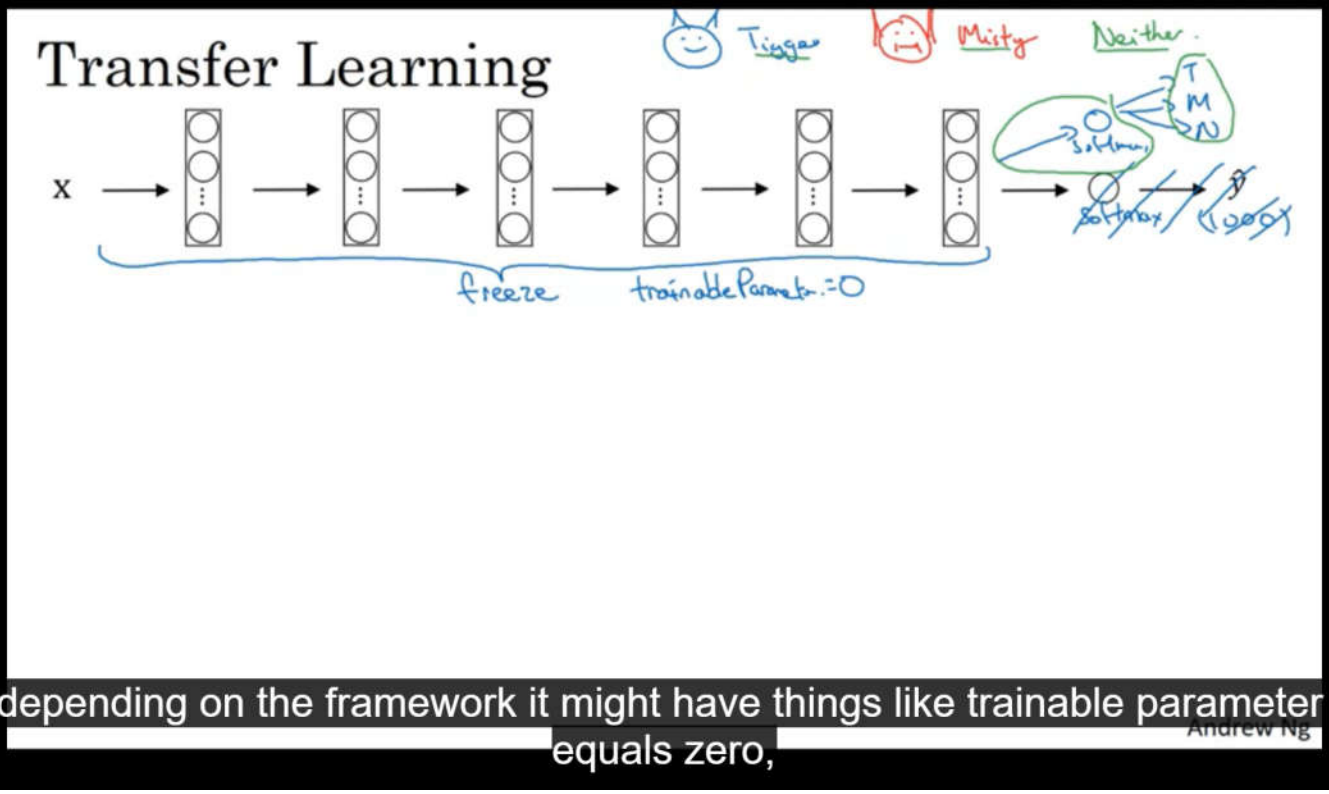
your own softmax unit that outputs Tigger or Misty or neither.

Andrew Ng

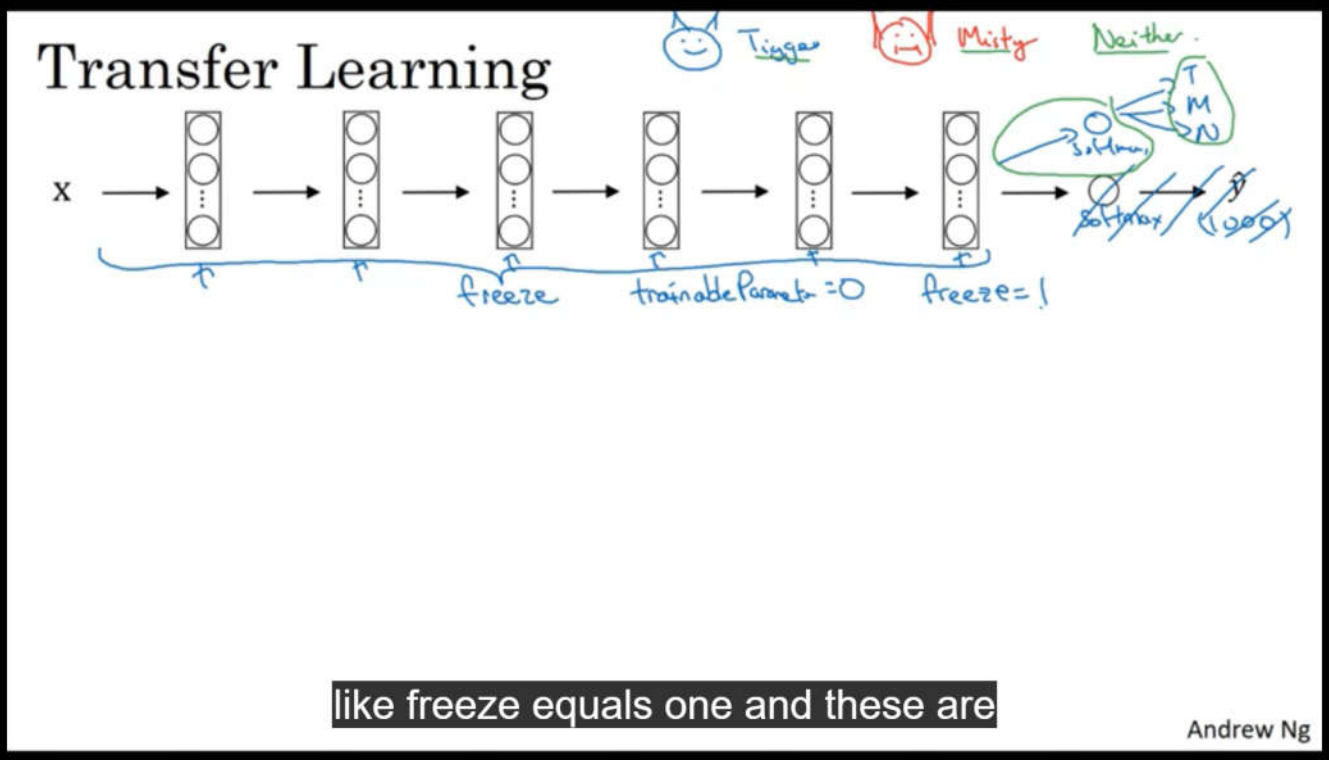


all of these layers of the network and you would then just

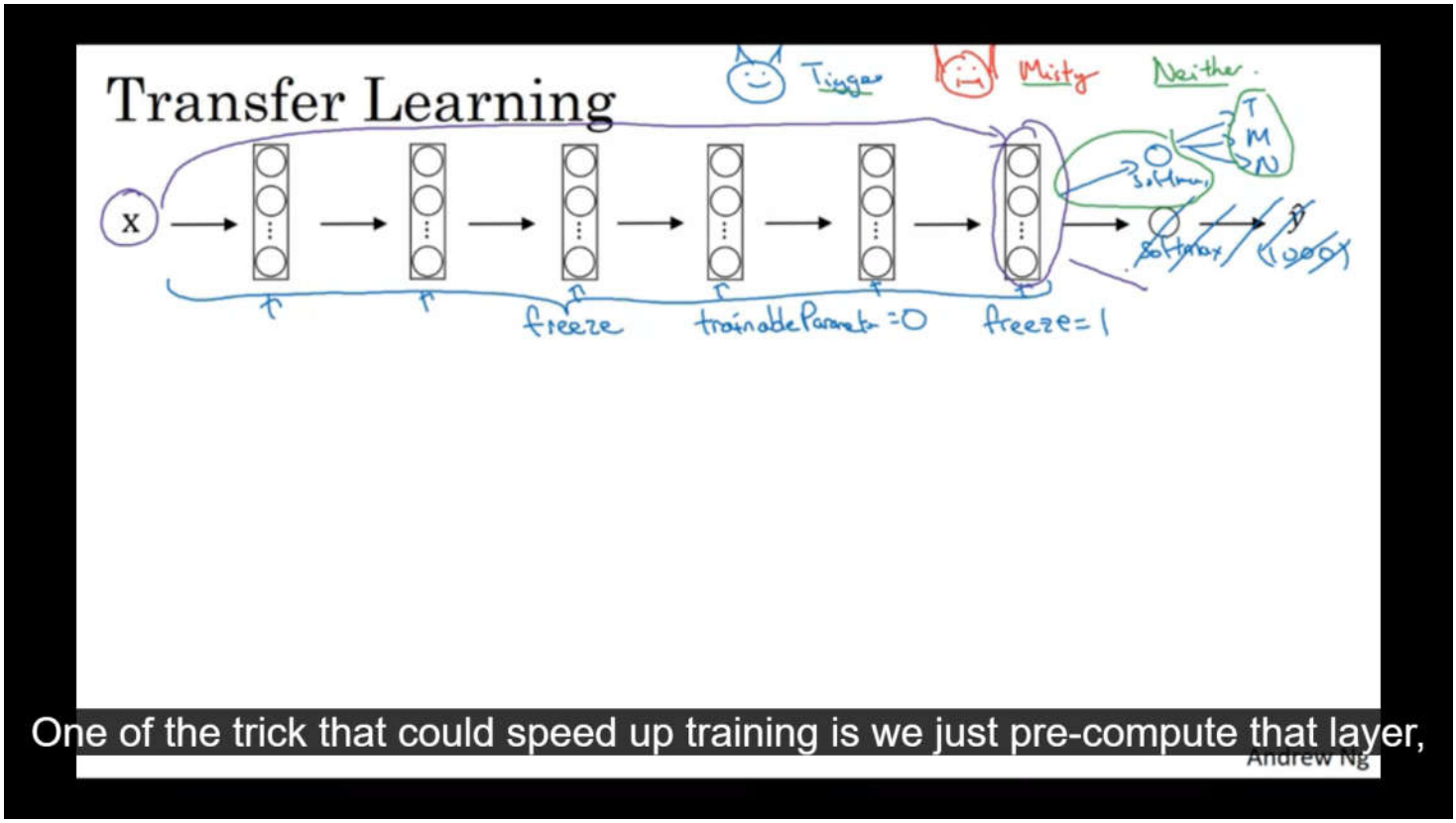
Andrew Ng



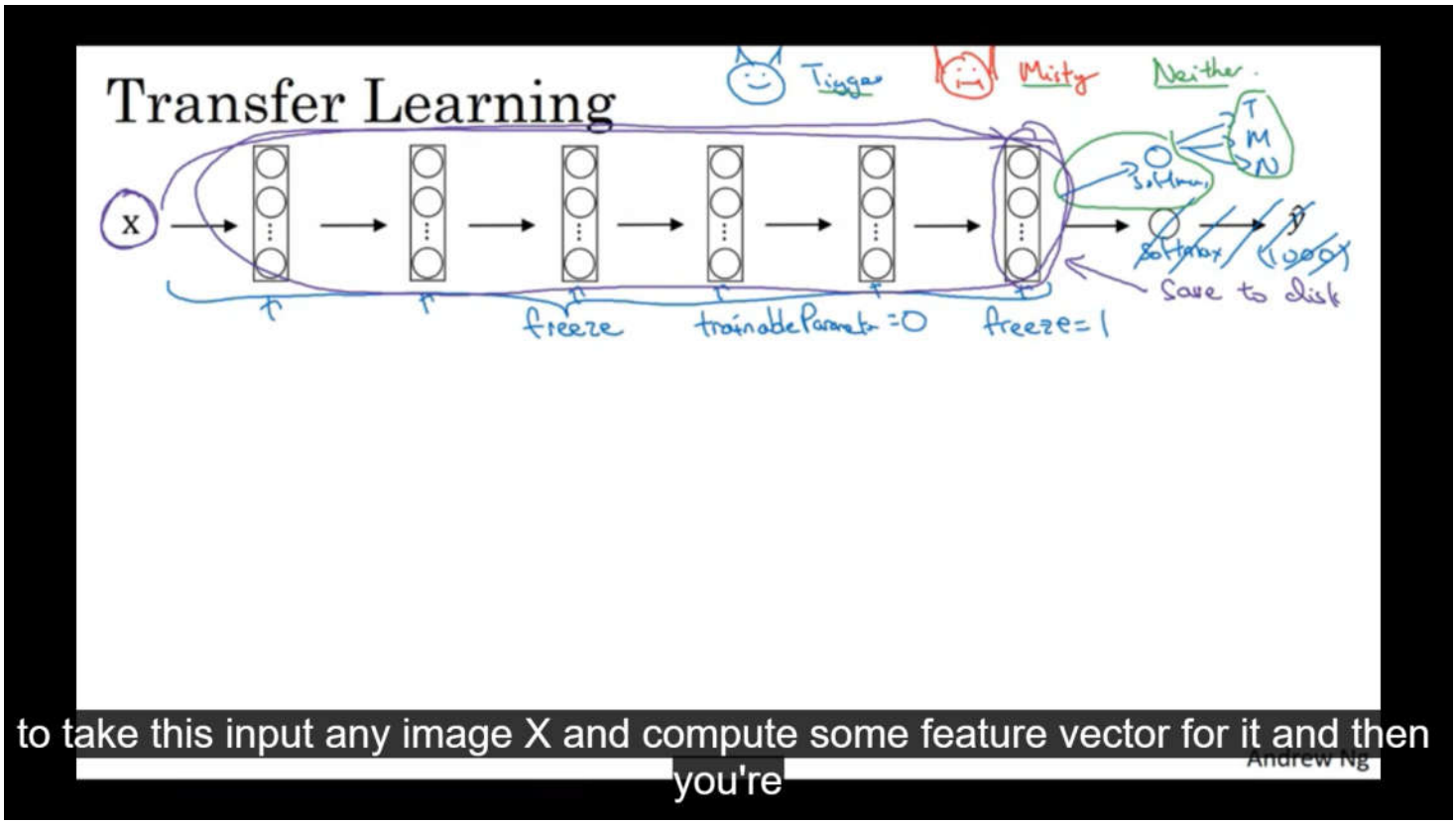
depending on the framework it might have things like trainable parameter equals zero,



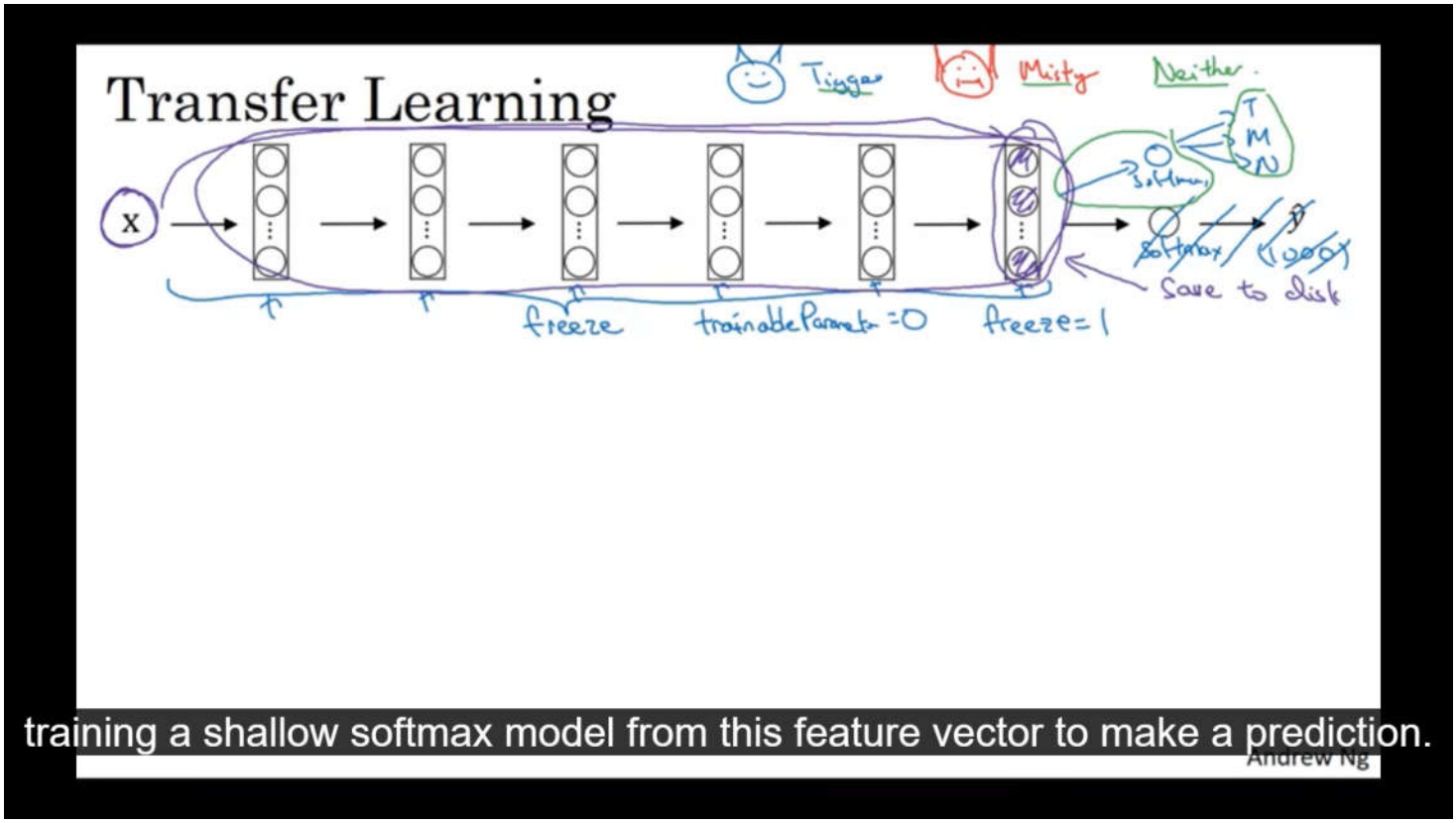
like freeze equals one and these are



One of the trick that could speed up training is we just pre-compute that layer,

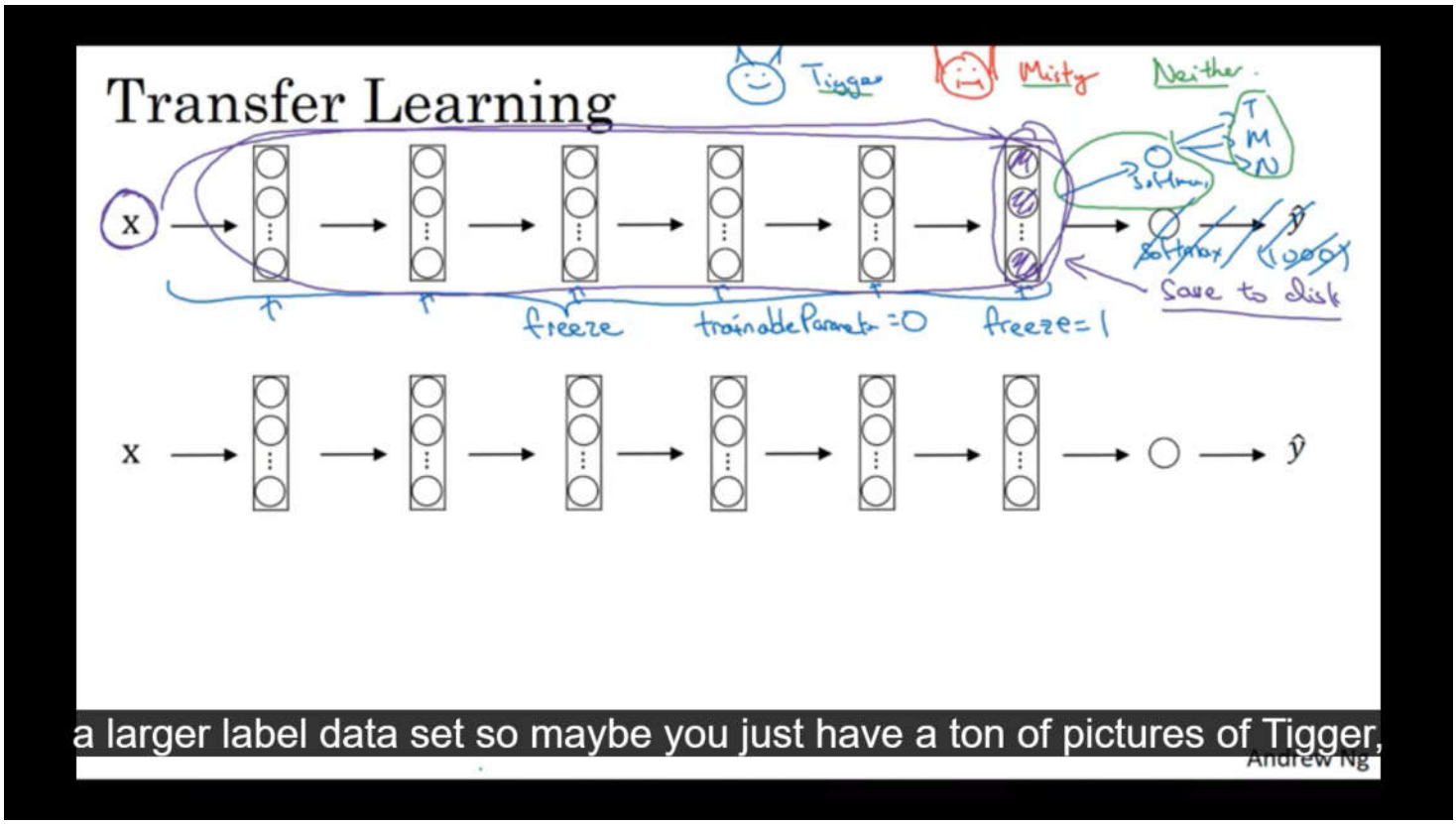


to take this input any image X and compute some feature vector for it and then you're



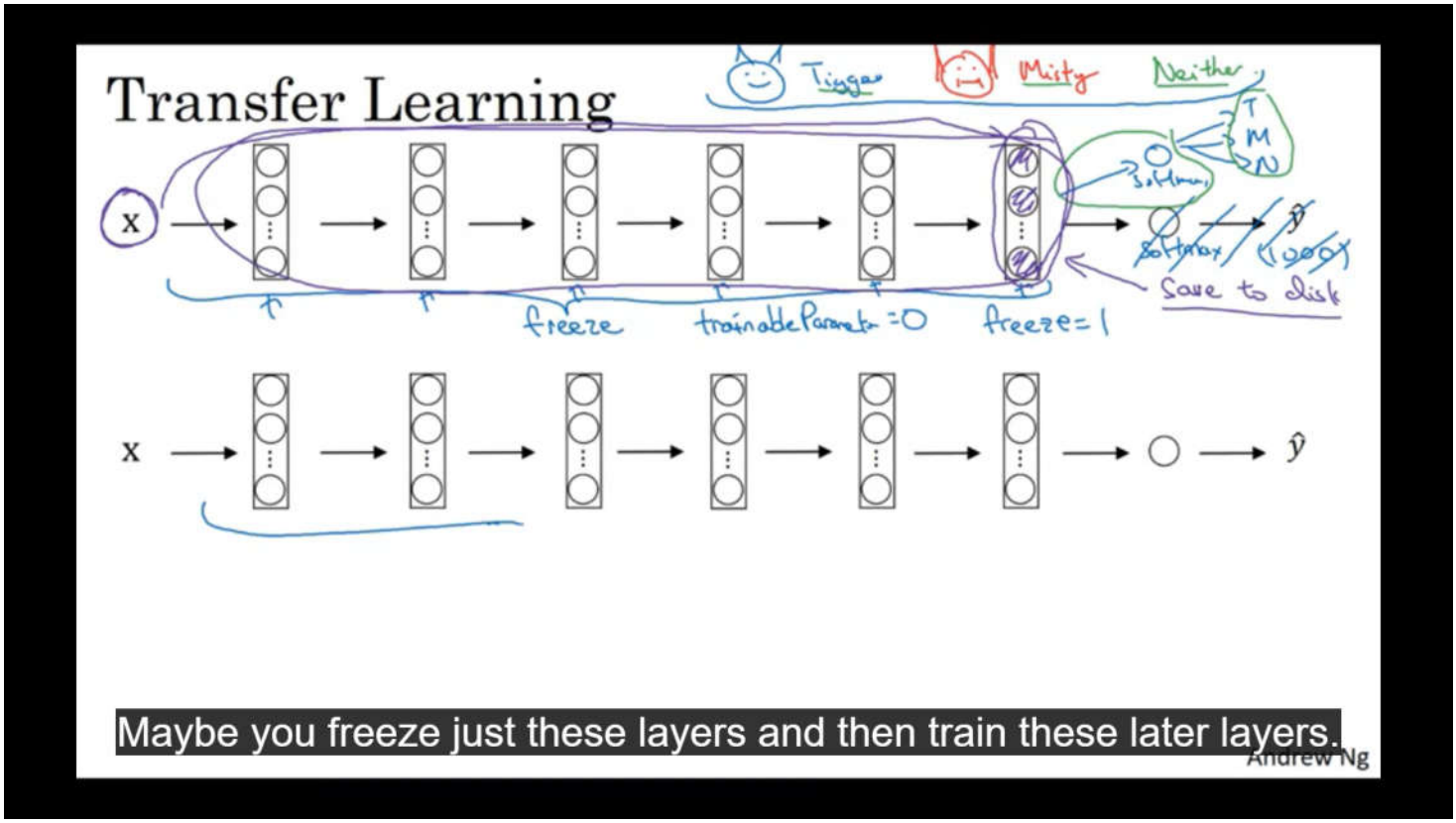
training a shallow softmax model from this feature vector to make a prediction.

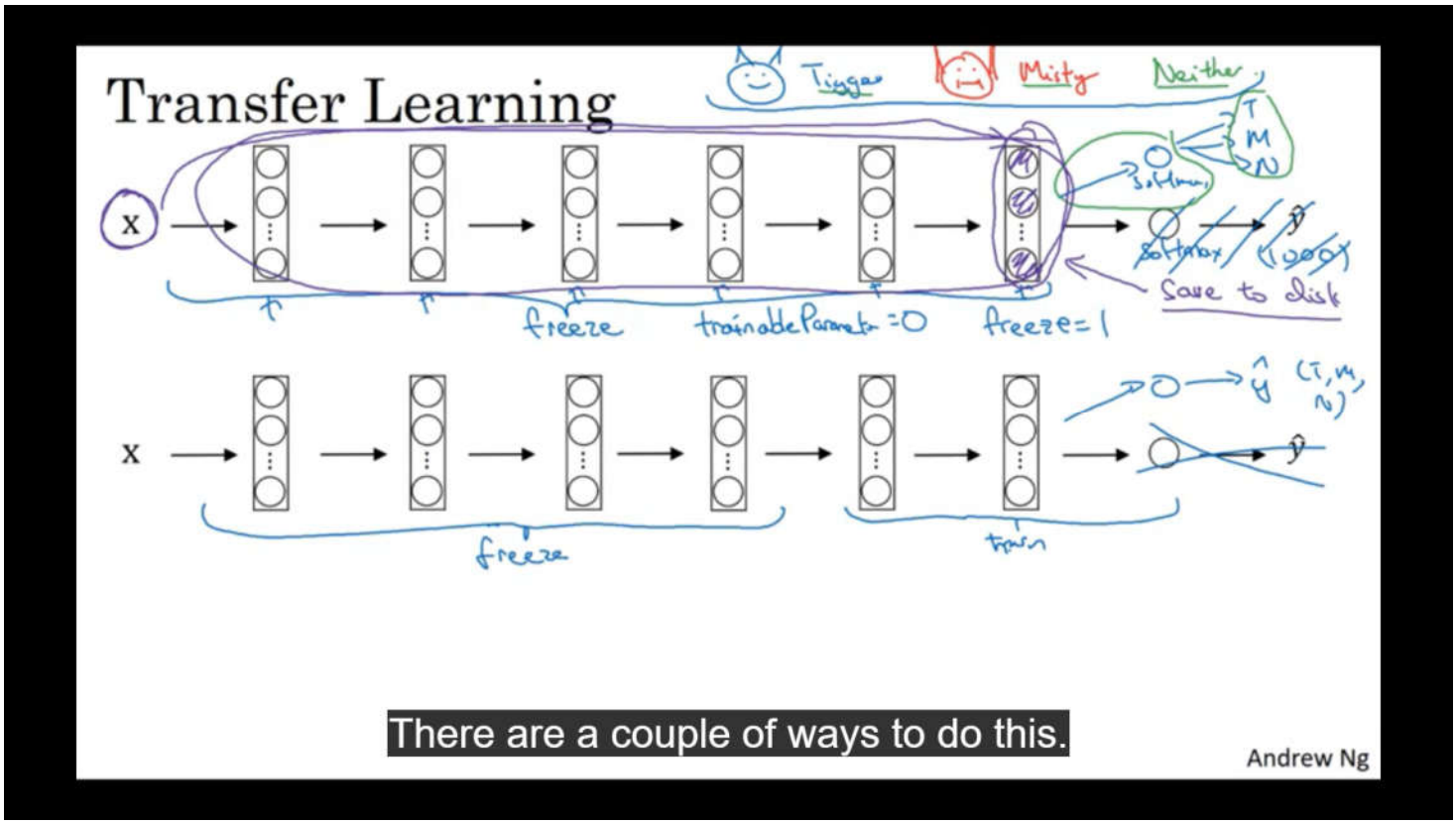
Andrew Ng

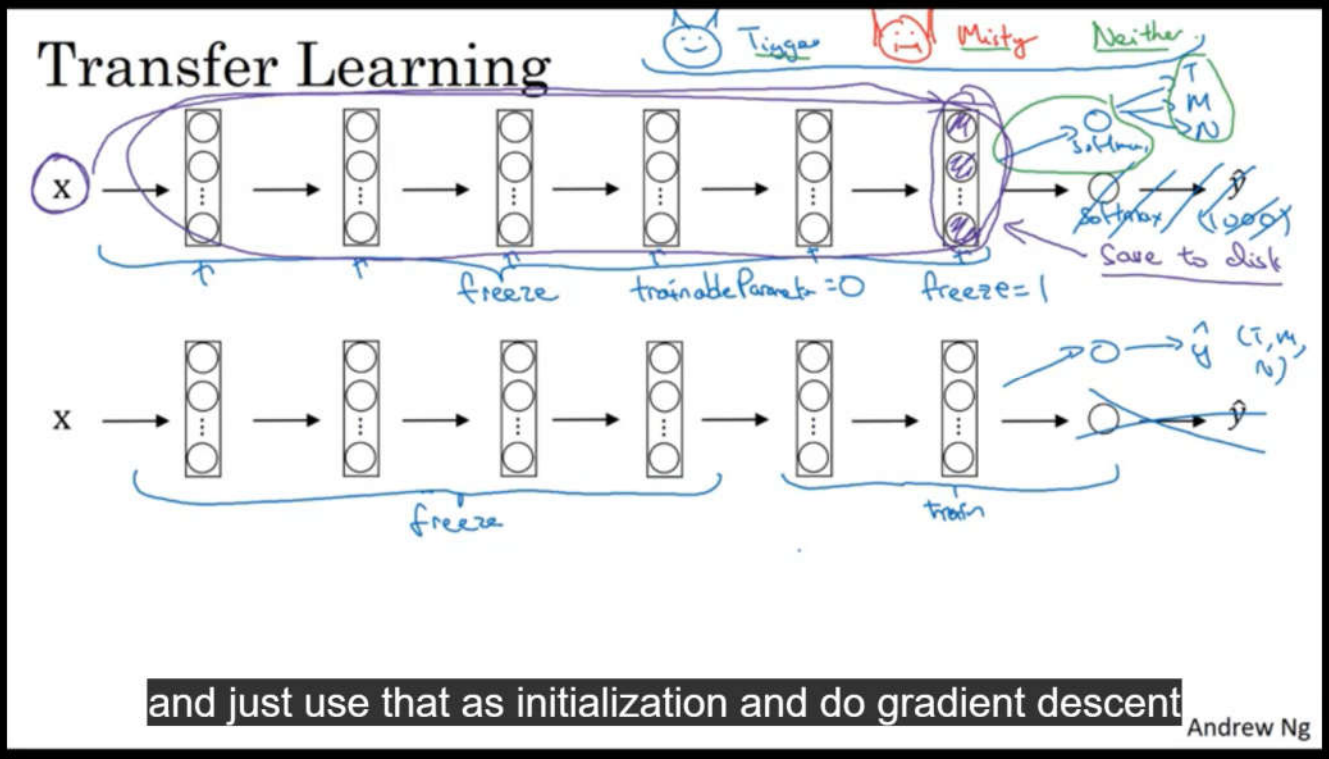


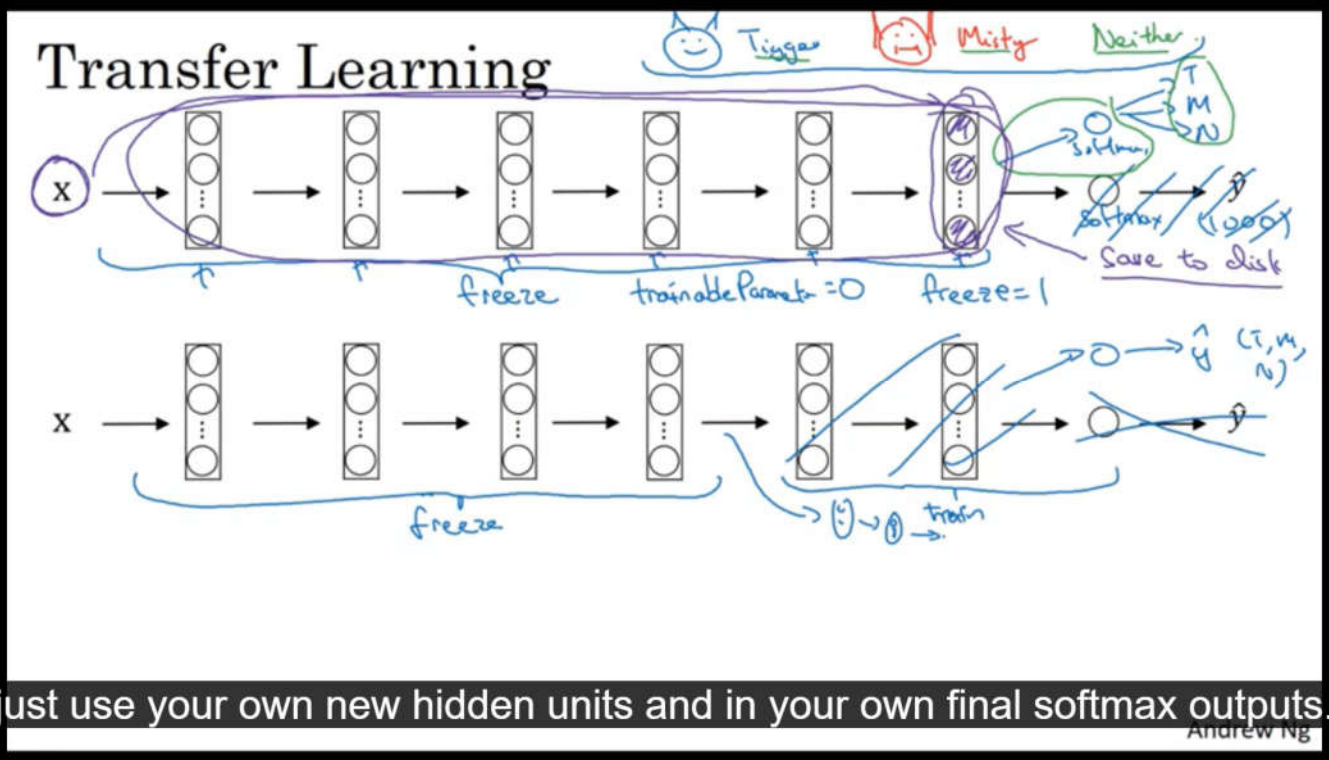
a larger label data set so maybe you just have a ton of pictures of Tigger,

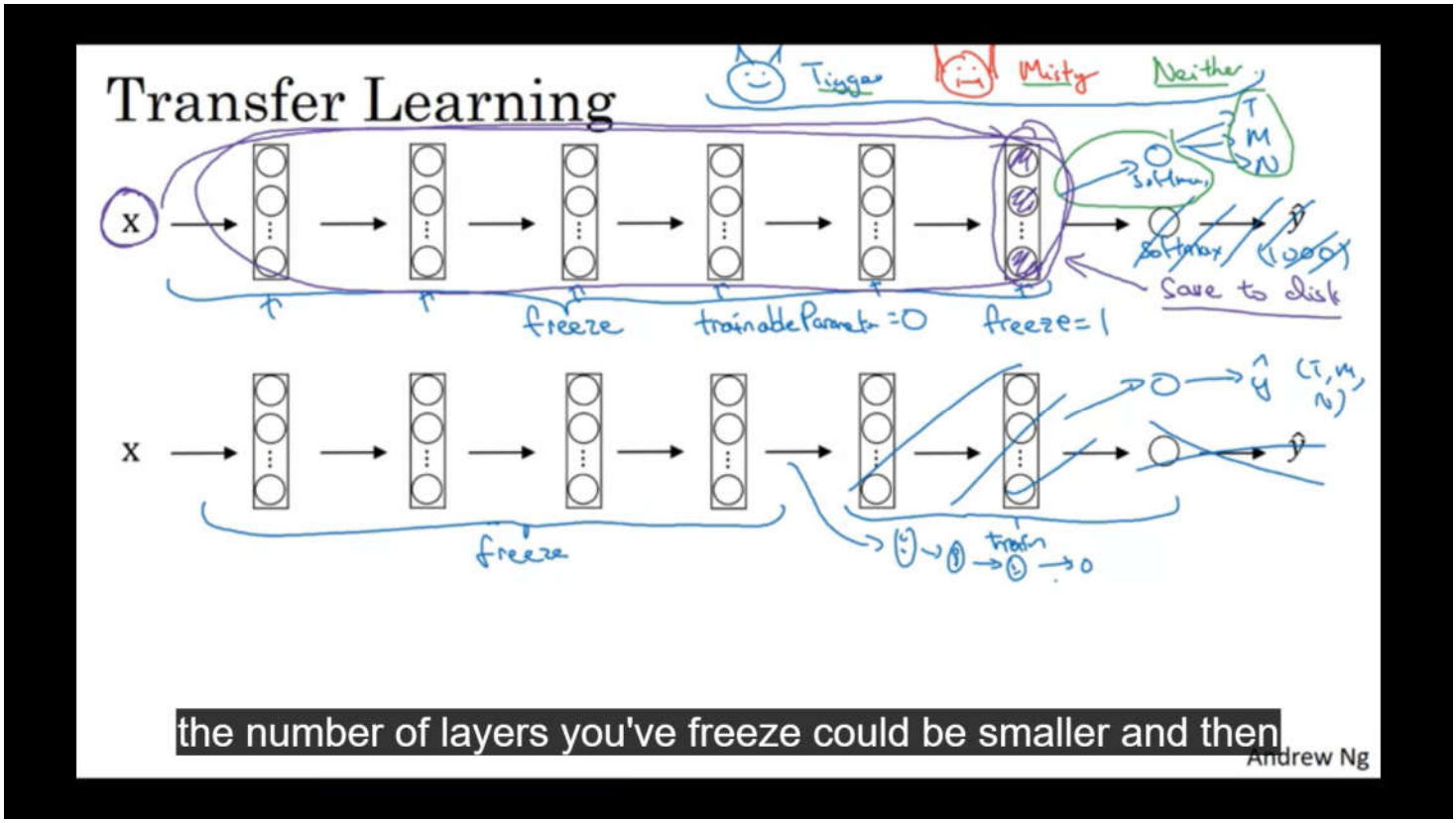
Andrew Ng

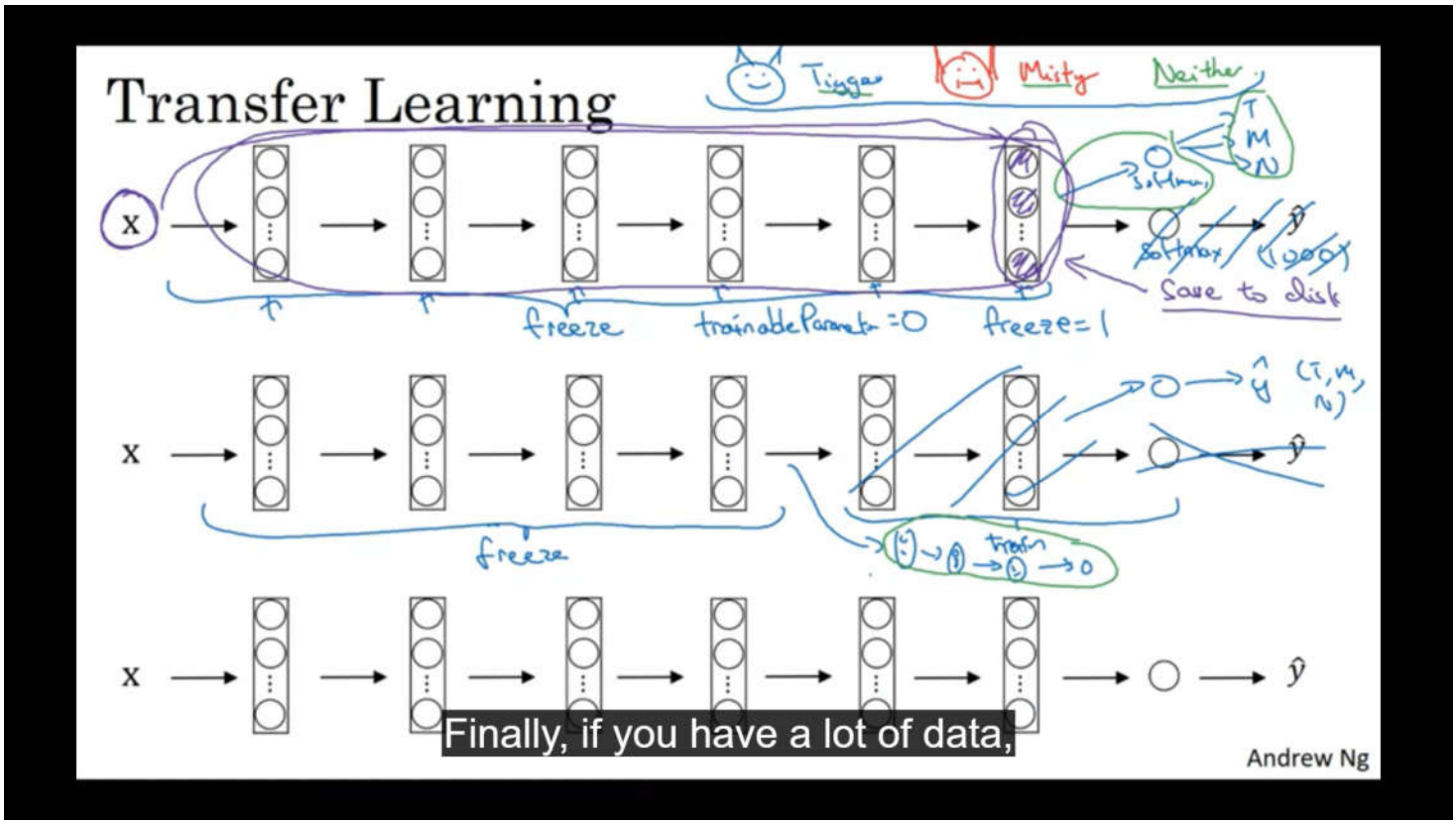

Transfer Learning
Tiggo
Misty
Neither
T
M
N
softmax
x
ŷ
softmax
Save to disk
freeze
trainableParameter = 0
freeze = 1
x
ŷ
Maybe you freeze just these layers and then train these later layers.
Andrew Ng

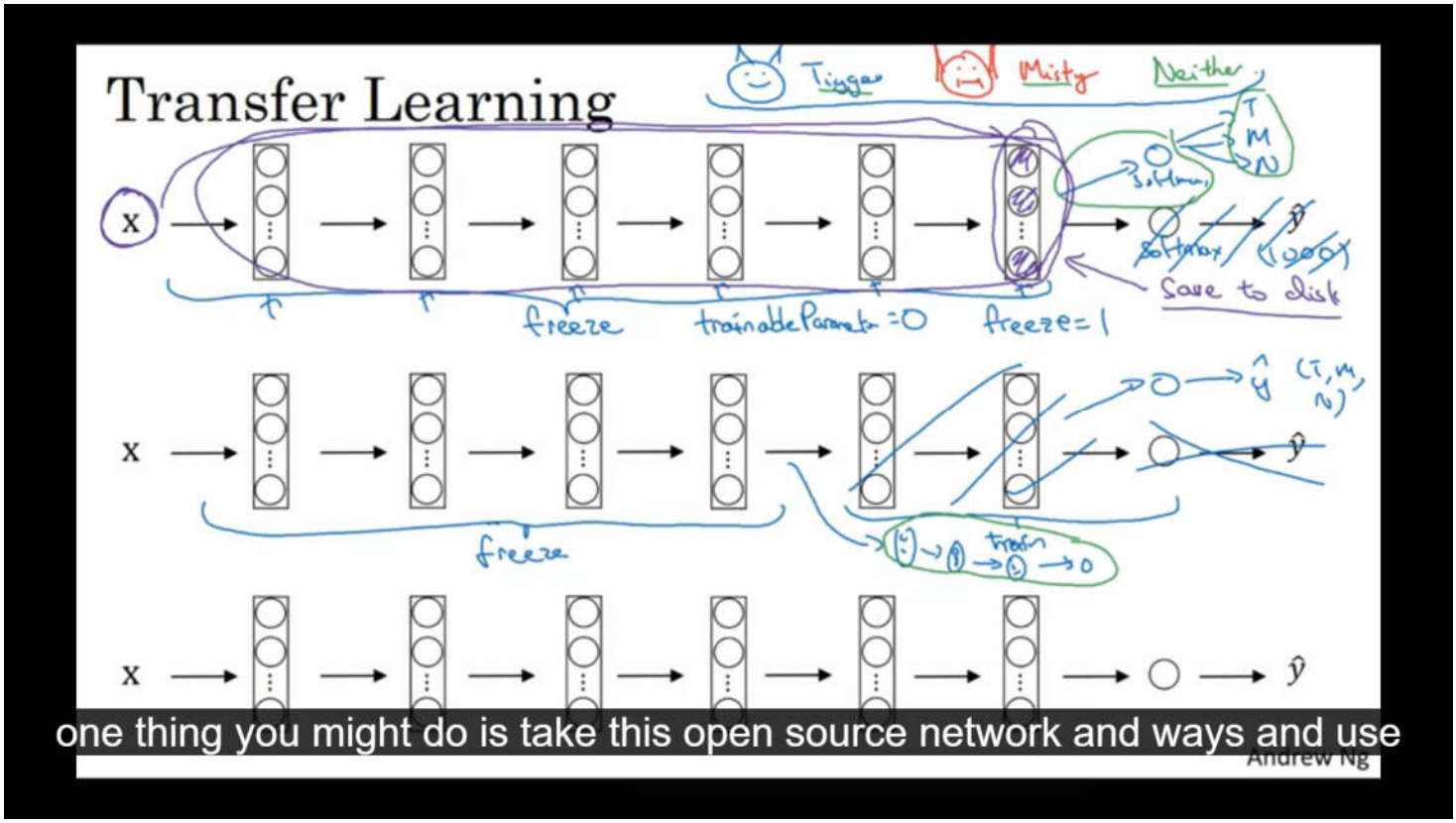

Transfer Learning
Tiggo
Misty
Neither
T
M
N
softmax
x
ŷ
softmax
Save to disk
freeze
trainableParameter = 0
freeze = 1
x
ŷ
(T, M, N)
ŷ
freeze
train
There are a couple of ways to do this.
Andrew Ng


Transfer Learning
Tigger
Misty
Neither
T
M
N
softmax
x
Save to disk
freeze
trainable Parameter = 0
freeze = 1
x
freeze
train
and just use that as initialization and do gradient descent
Andrew Ng

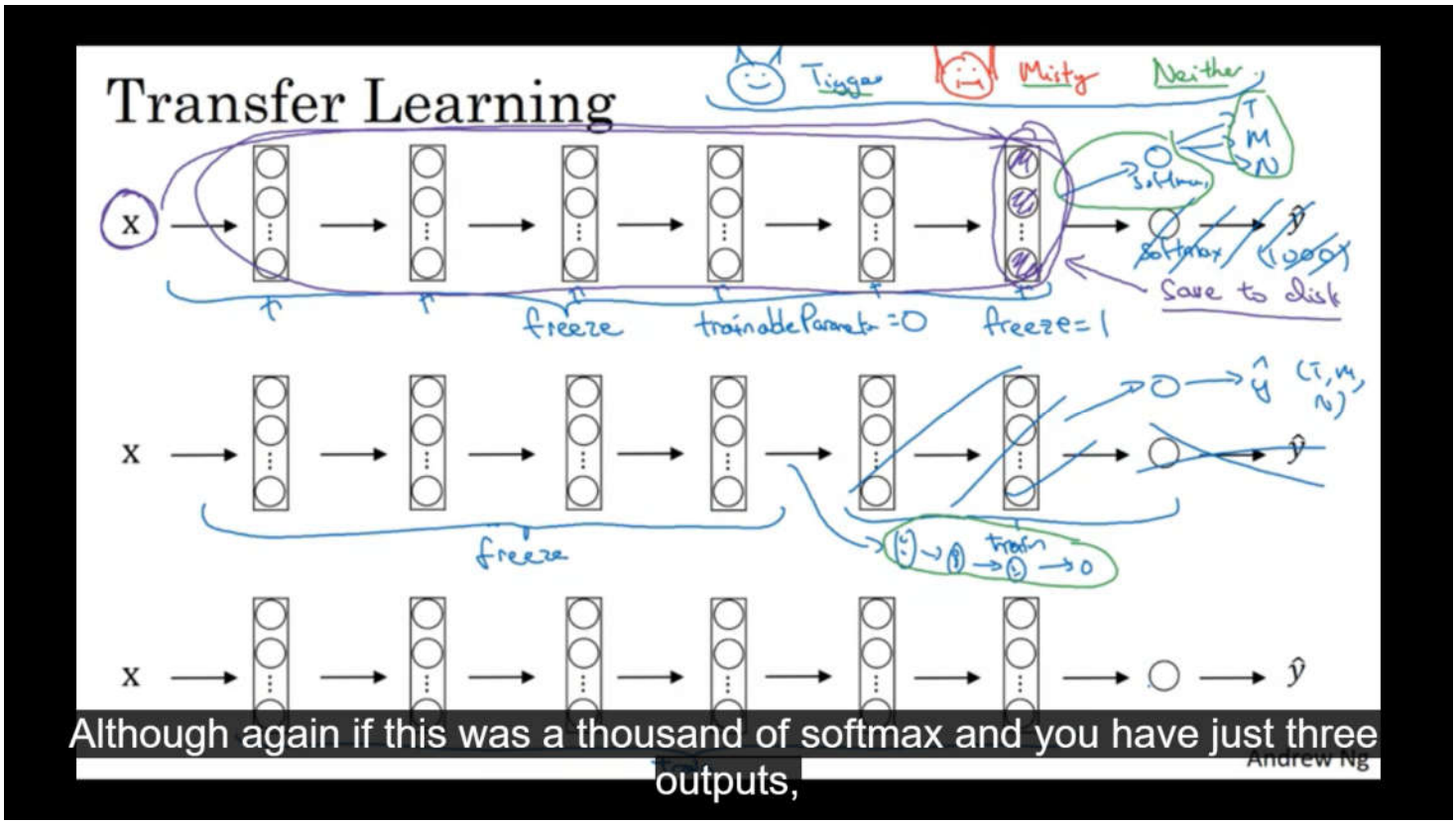

Transfer Learning
Tigger
Misty
Neither
T
M
N
softmax
x
Save to disk
freeze
trainable Parameter = 0
freeze = 1
x
freeze
train
just use your own new hidden units and in your own final softmax outputs.
Andrew Ng


Transfer Learning
Tigger
Misty
Neither
T
M
N
softmax
Save to disk
freeze
trainableParameter =0
freeze=1
x
freeze
train
the number of layers you've freeze could be smaller and then
Andrew Ng

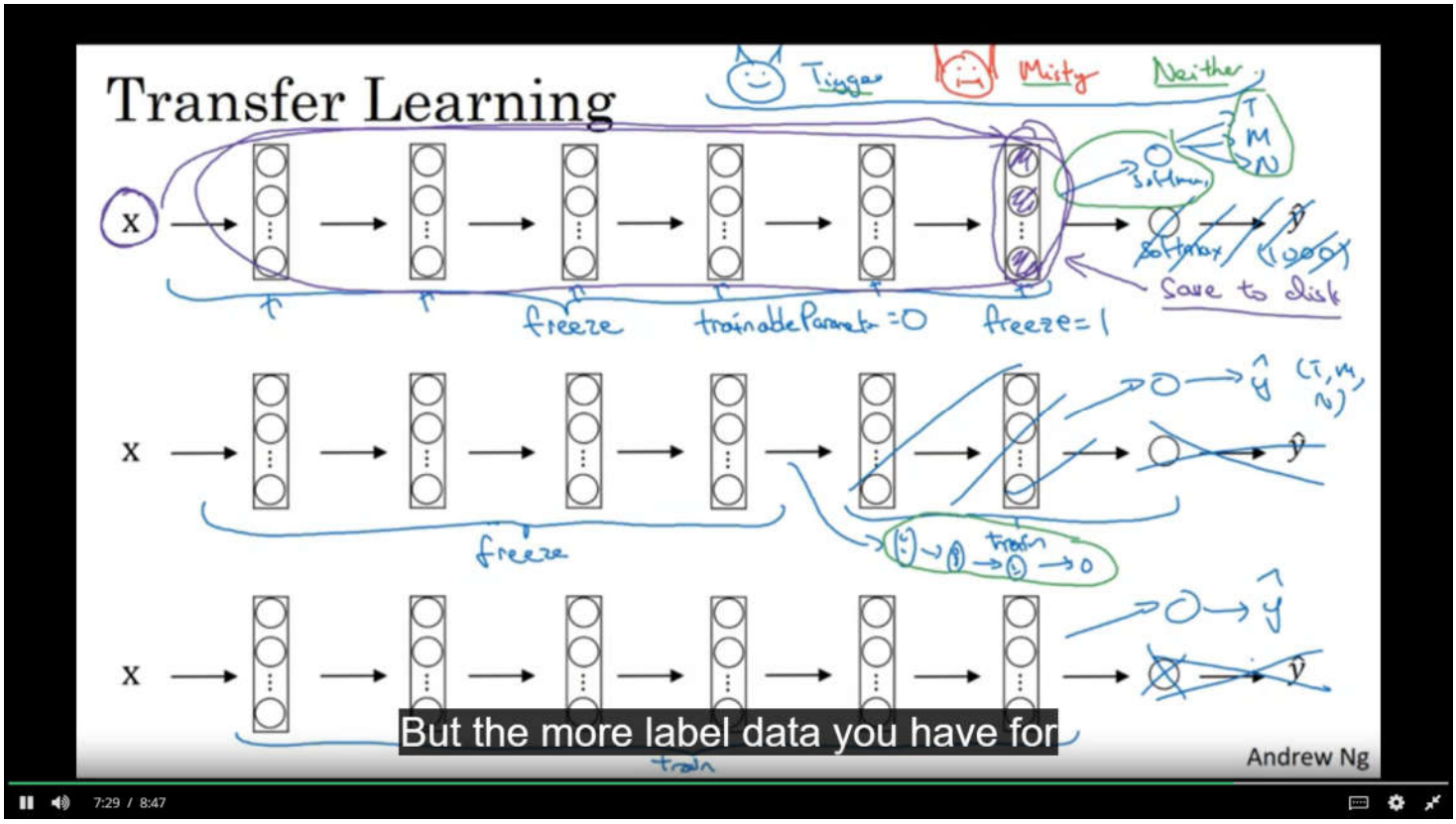

Transfer Learning
Tigger
Misty
Neither
T
M
N
softmax
Save to disk
freeze
trainableParameter =0
freeze=1
x
freeze
train
Finally, if you have a lot of data,
Andrew Ng

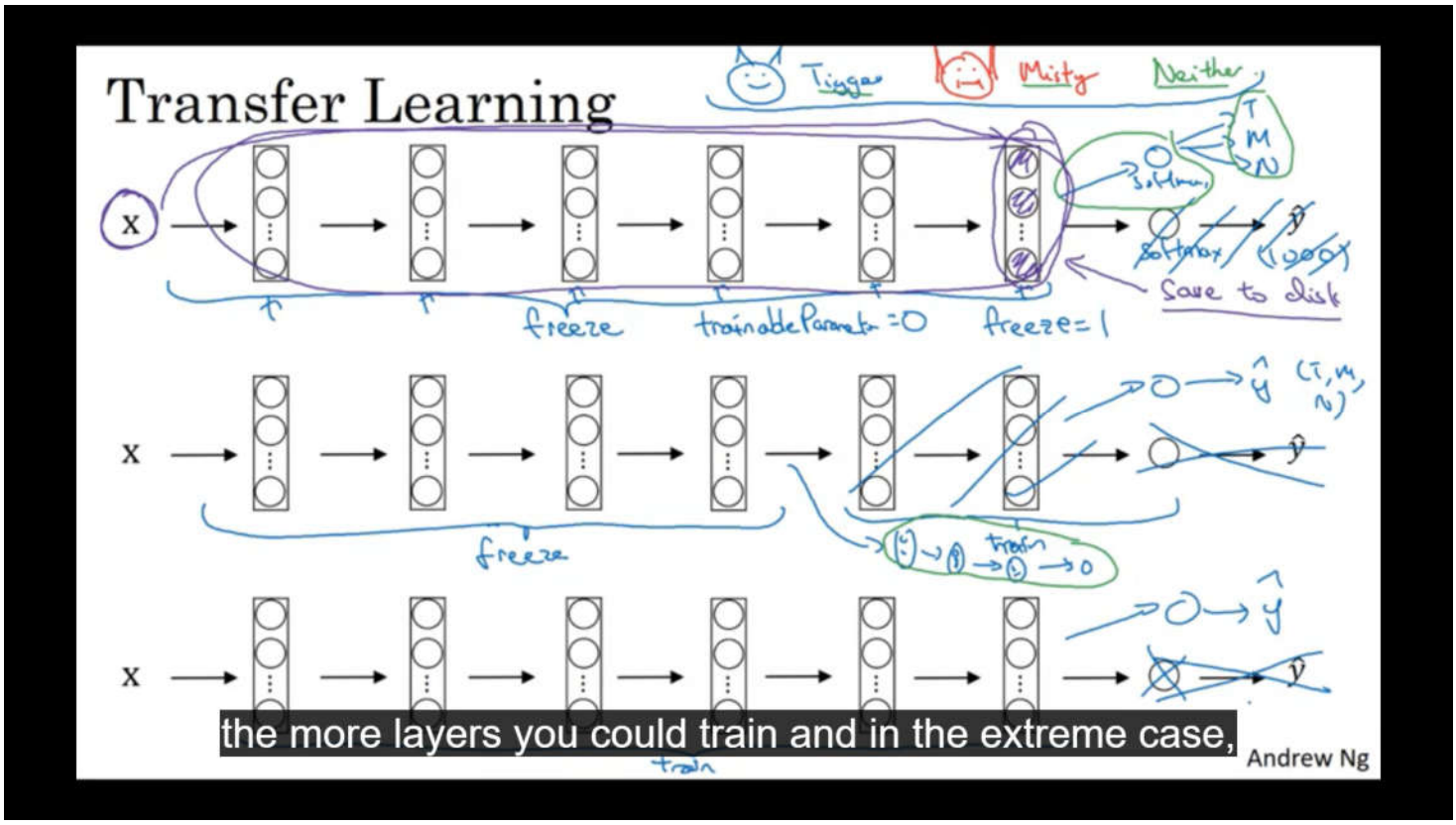


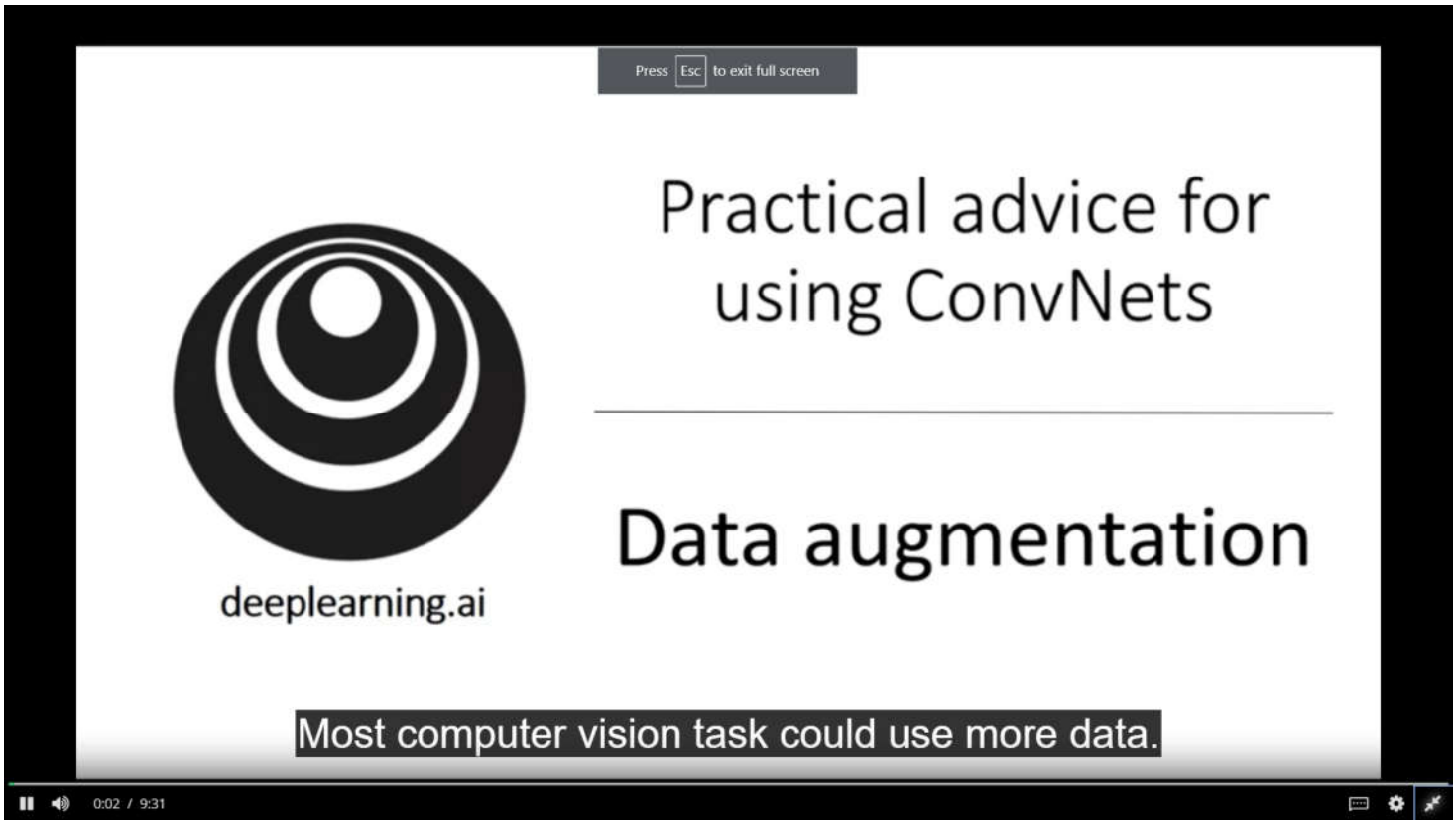
one thing you might do is take this open source network and ways and use

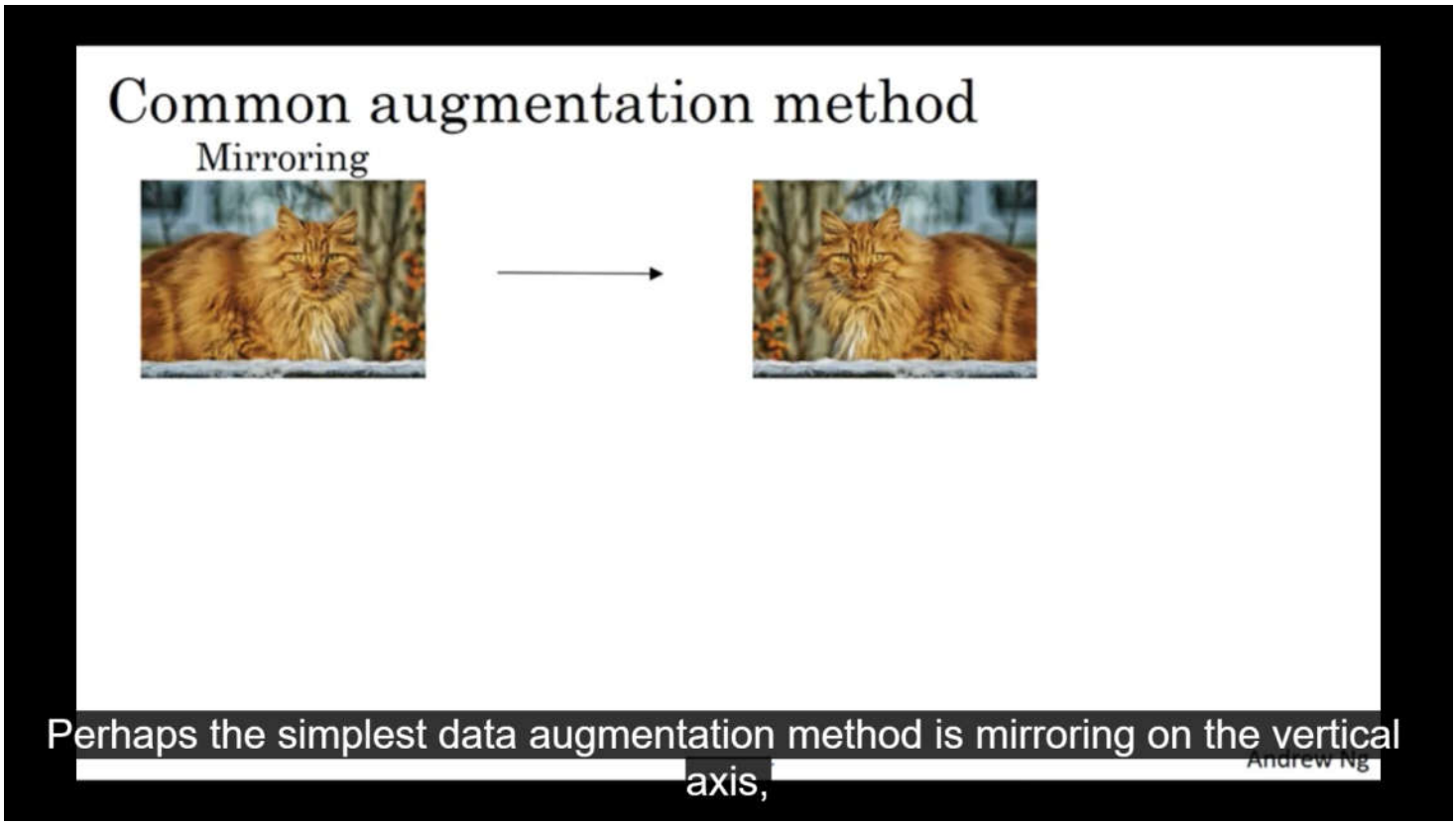


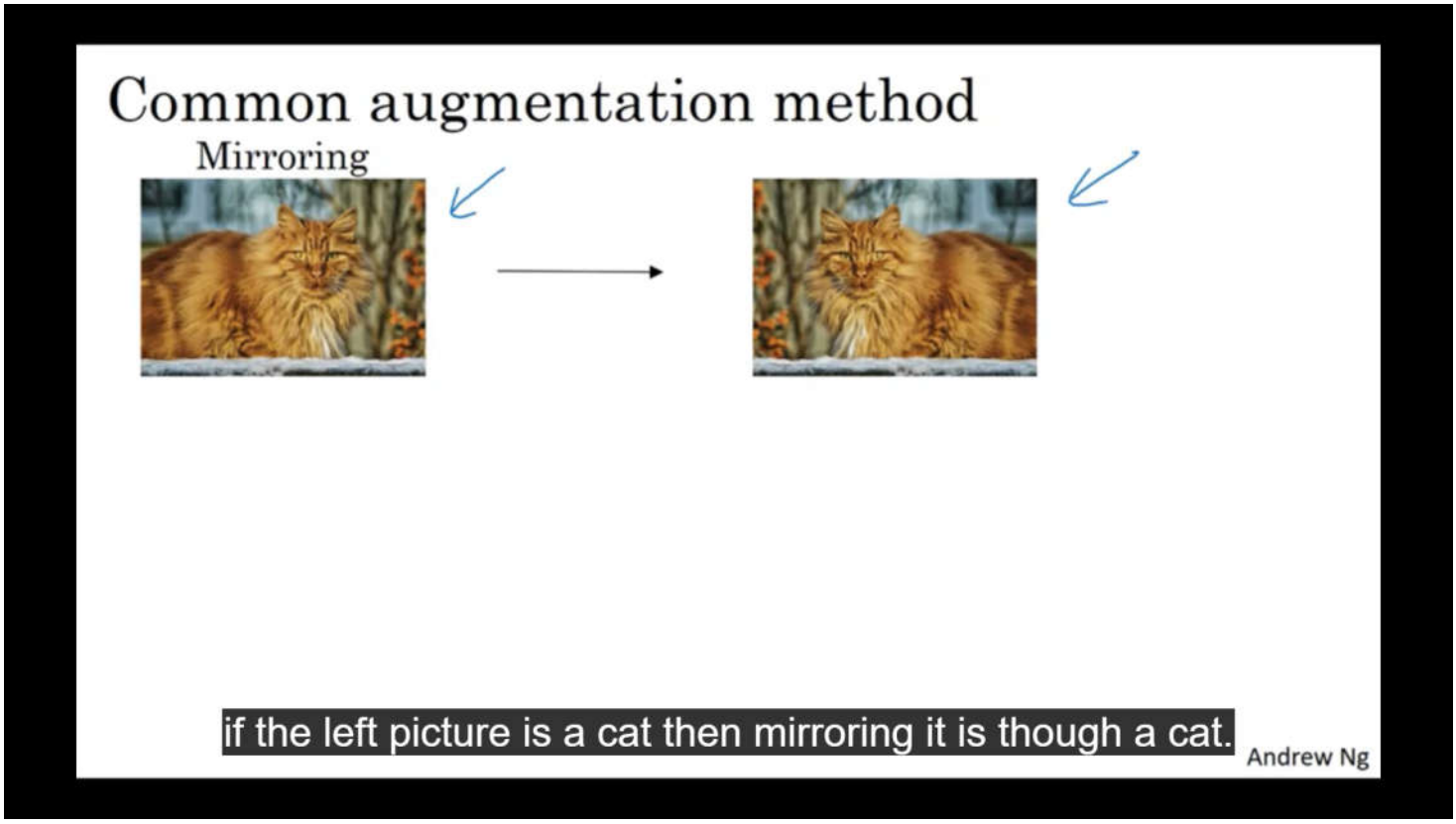
Although again if this was a thousand of softmax and you have just three outputs,

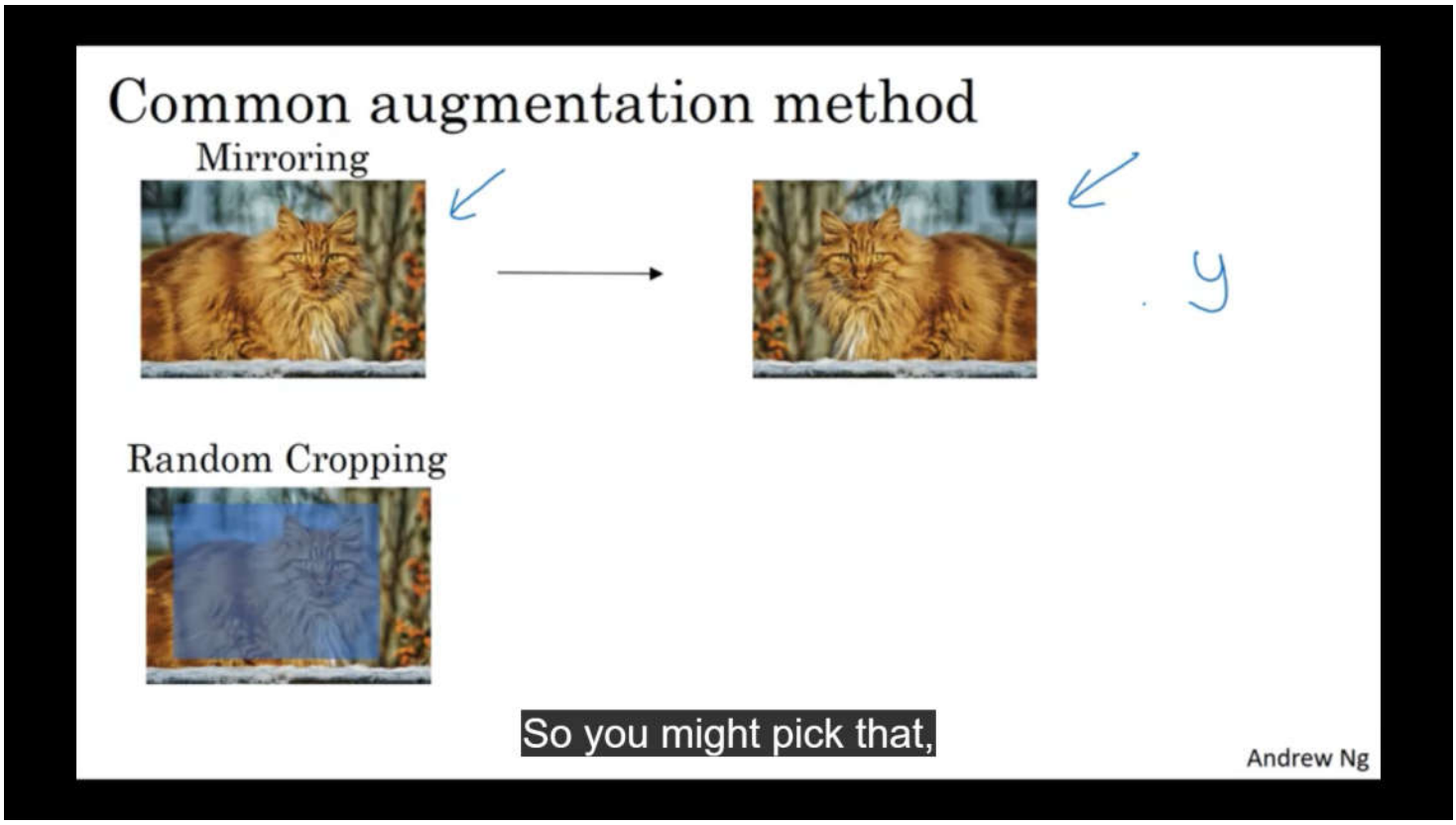

Transfer Learning
Tigger
Misty
Neither
T
M
N
softmax
Save to disk
freeze
trainableParameter =0
freeze=1
x
freeze
train
But the more label data you have for
Andrew Ng
7:29 / 8:47

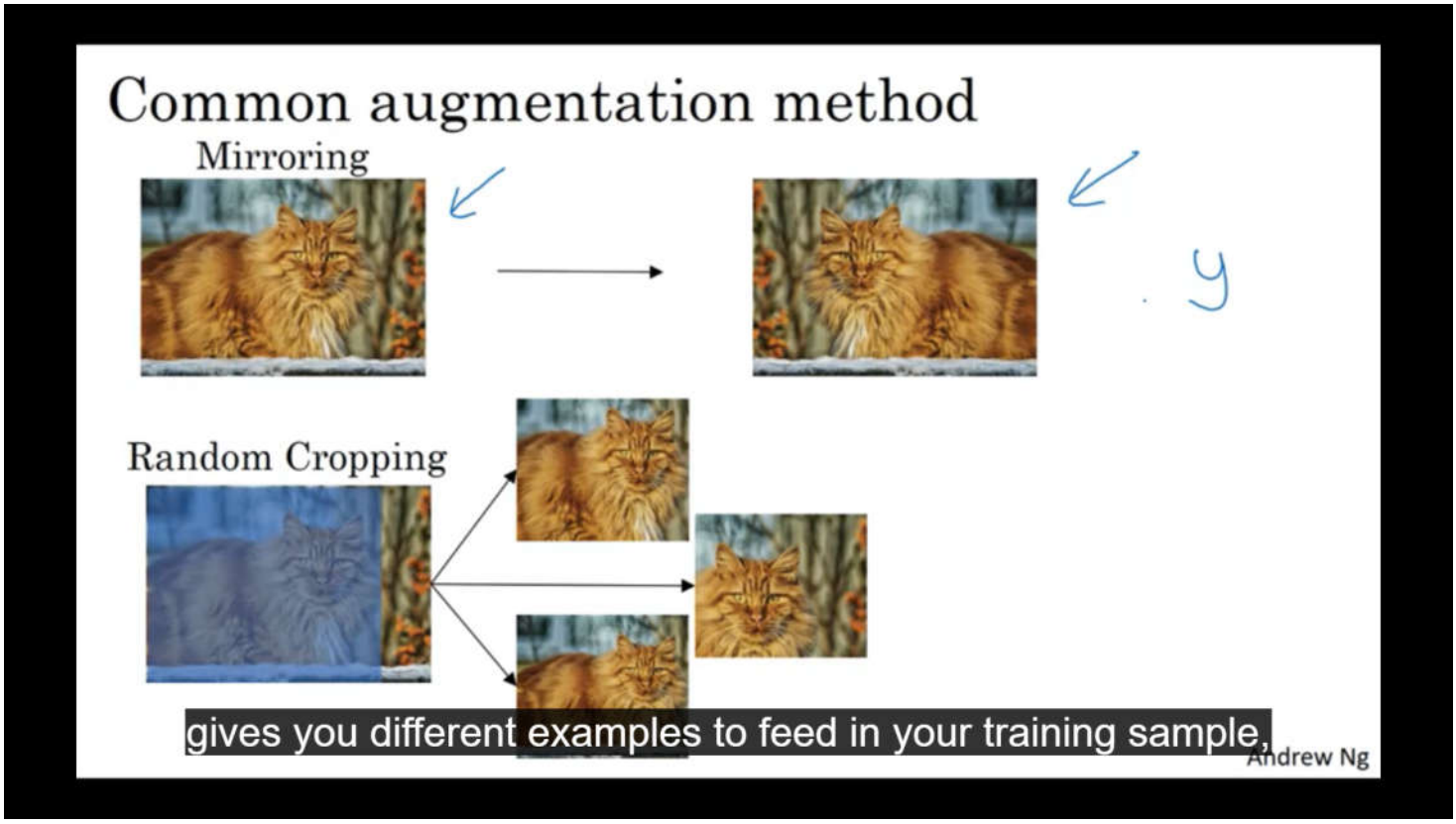

Transfer Learning
Tigger
Misty
Neither
T
M
N
softmax
Save to disk
freeze
trainableParameter =0
freeze=1
x
freeze
train
the more layers you could train and in the extreme case,
Andrew Ng

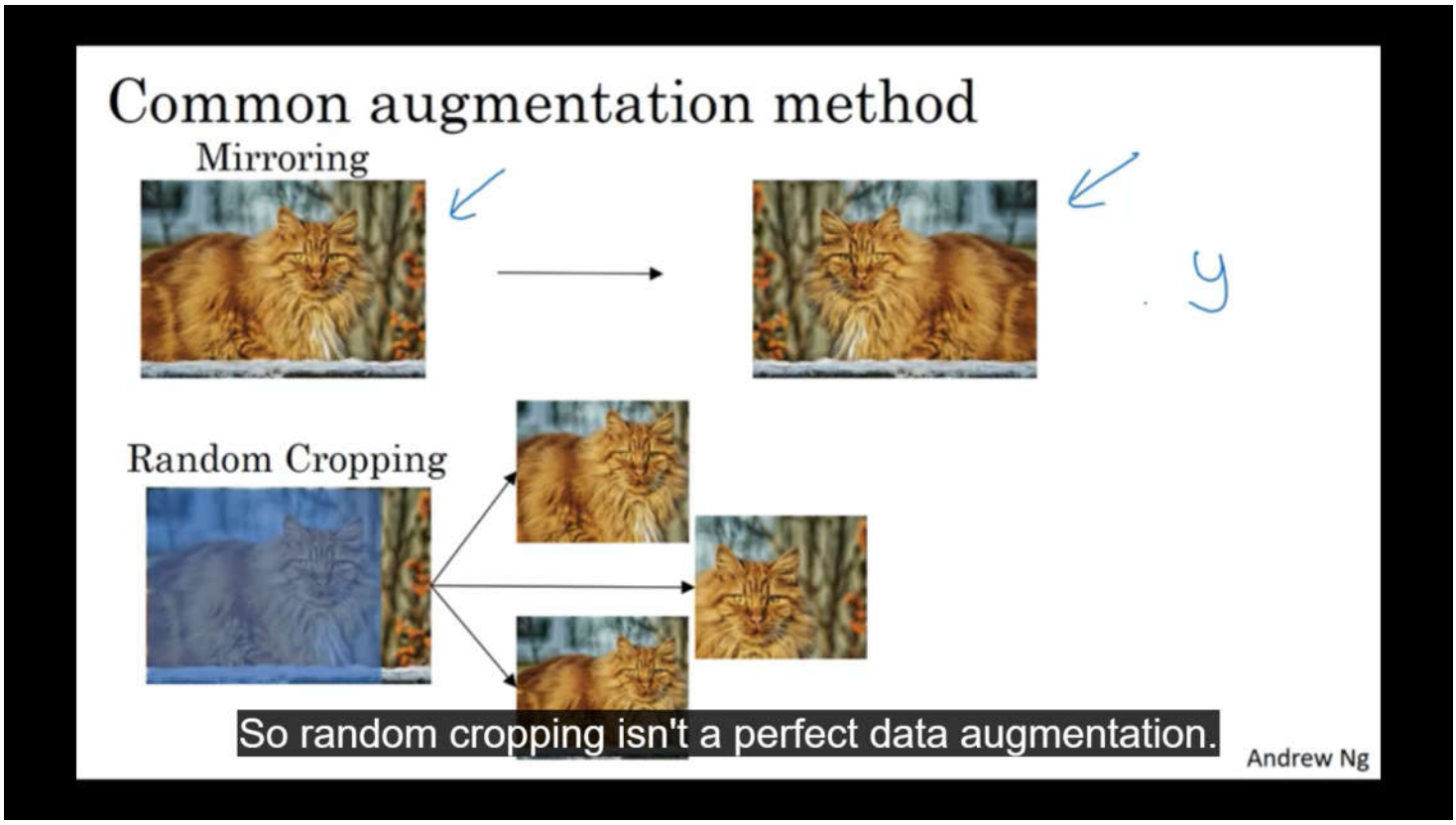

Press Esc to exit full screen
Practical advice for using ConvNets
deeplearning.ai
Data augmentation
Most computer vision task could use more data.
0:02 / 9:31

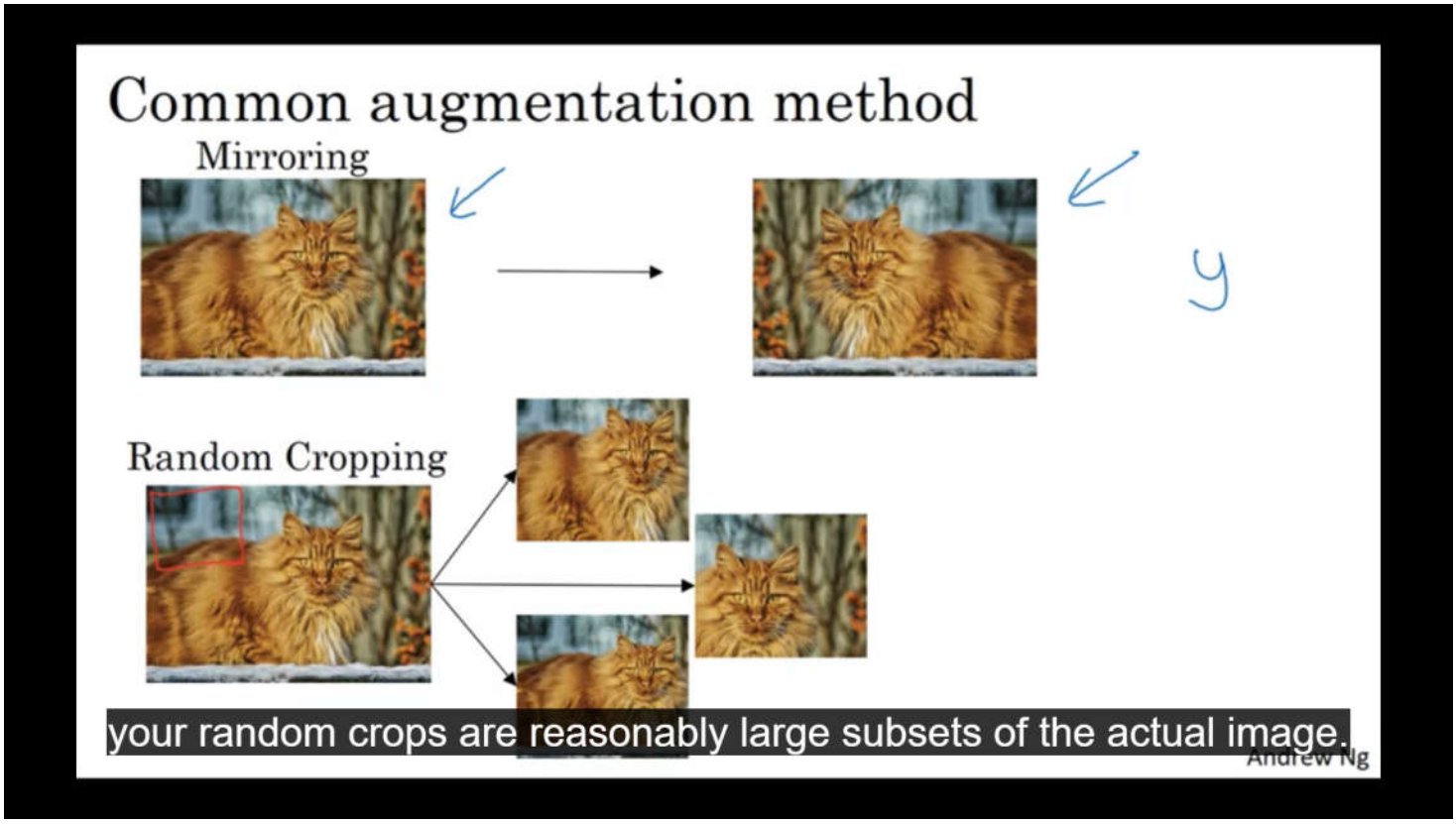

Common augmentation method
Mirroring
Perhaps the simplest data augmentation method is mirroring on the vertical axis,
Andrew Ng

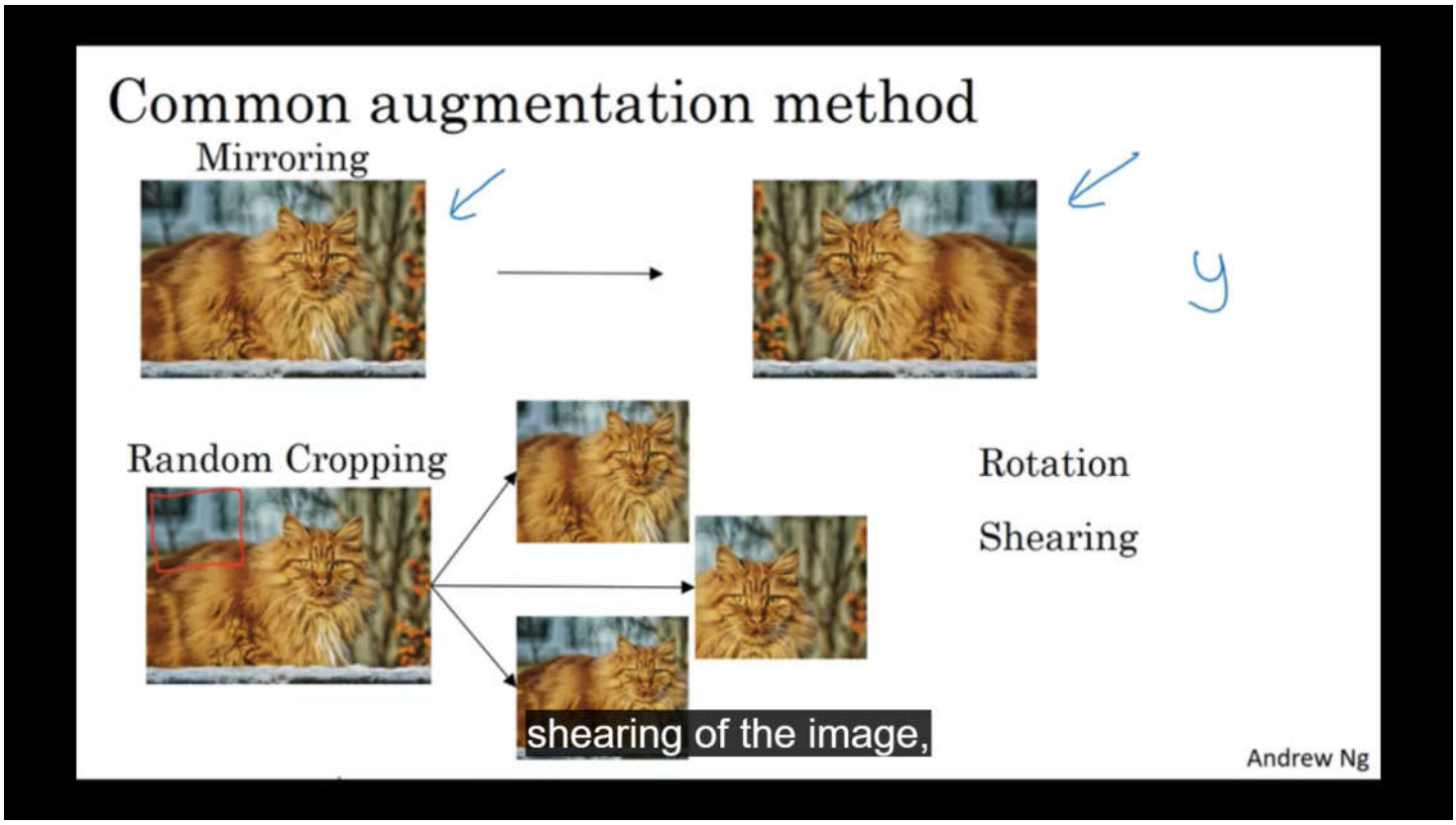

Common augmentation method
Mirroring
if the left picture is a cat then mirroring it is though a cat.
Andrew Ng

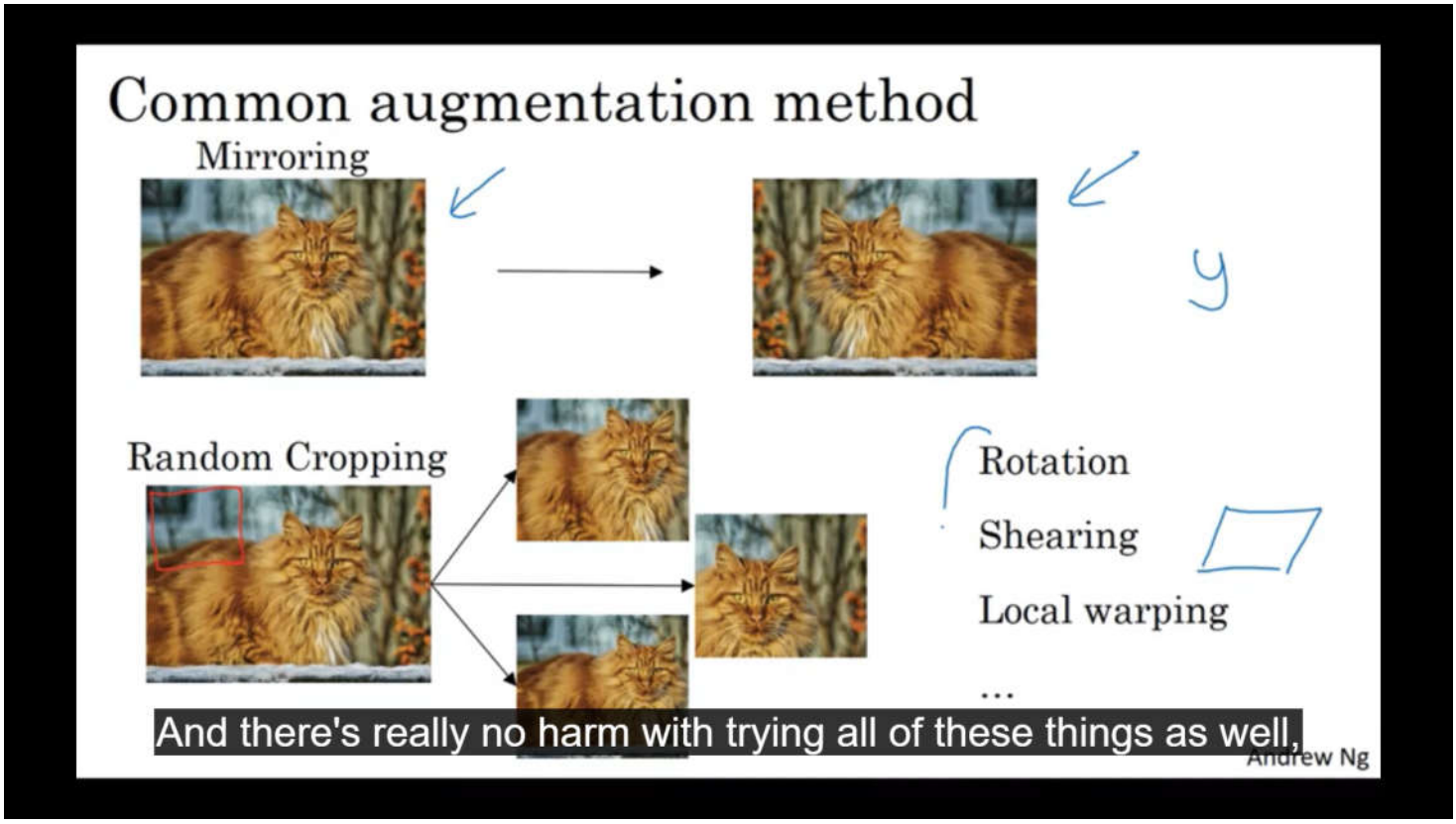

Common augmentation method
Mirroring
y
Random Cropping
So you might pick that,
Andrew Ng

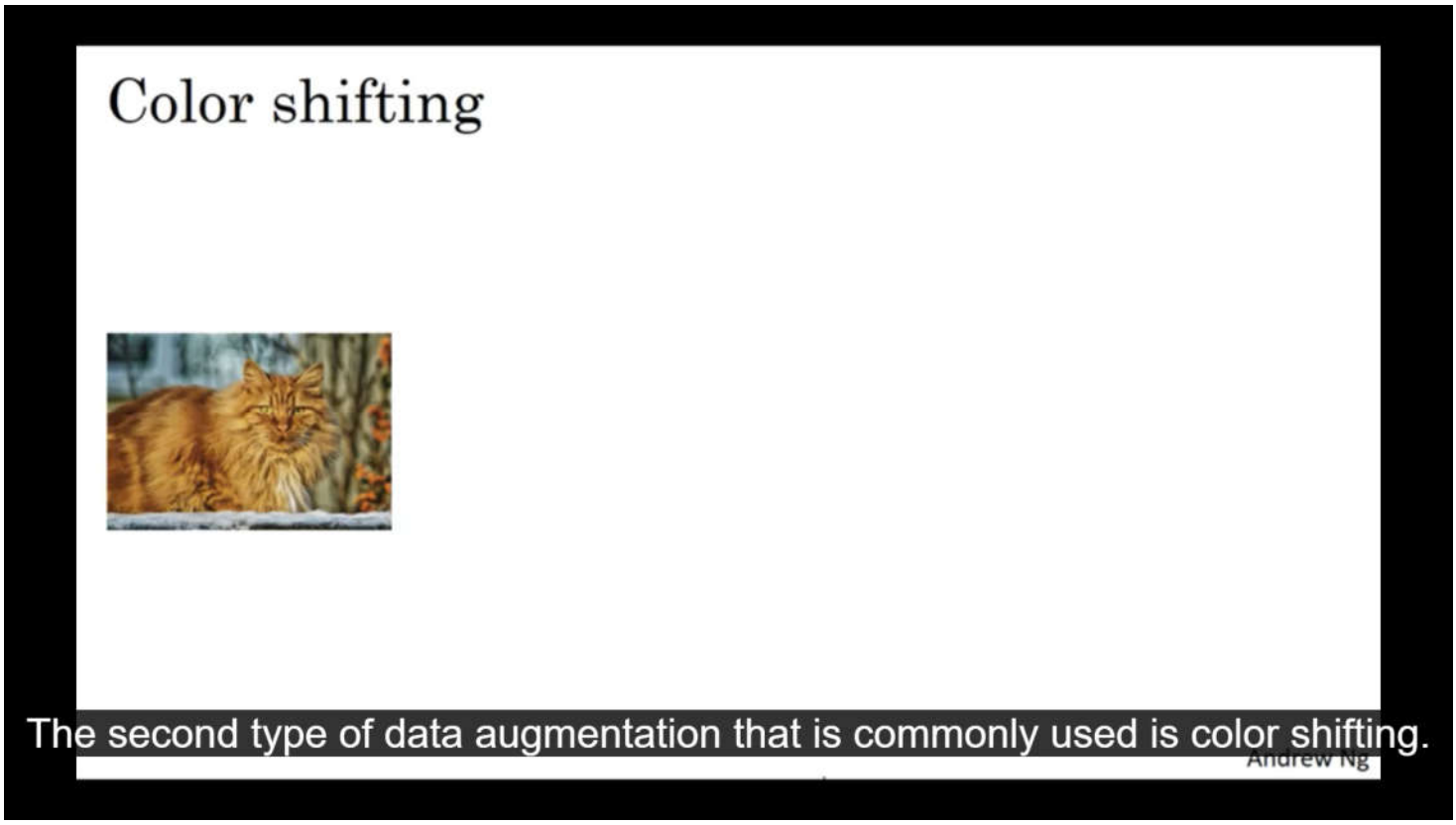

Common augmentation method
Mirroring
Random Cropping
y
gives you different examples to feed in your training sample,
Andrew Ng

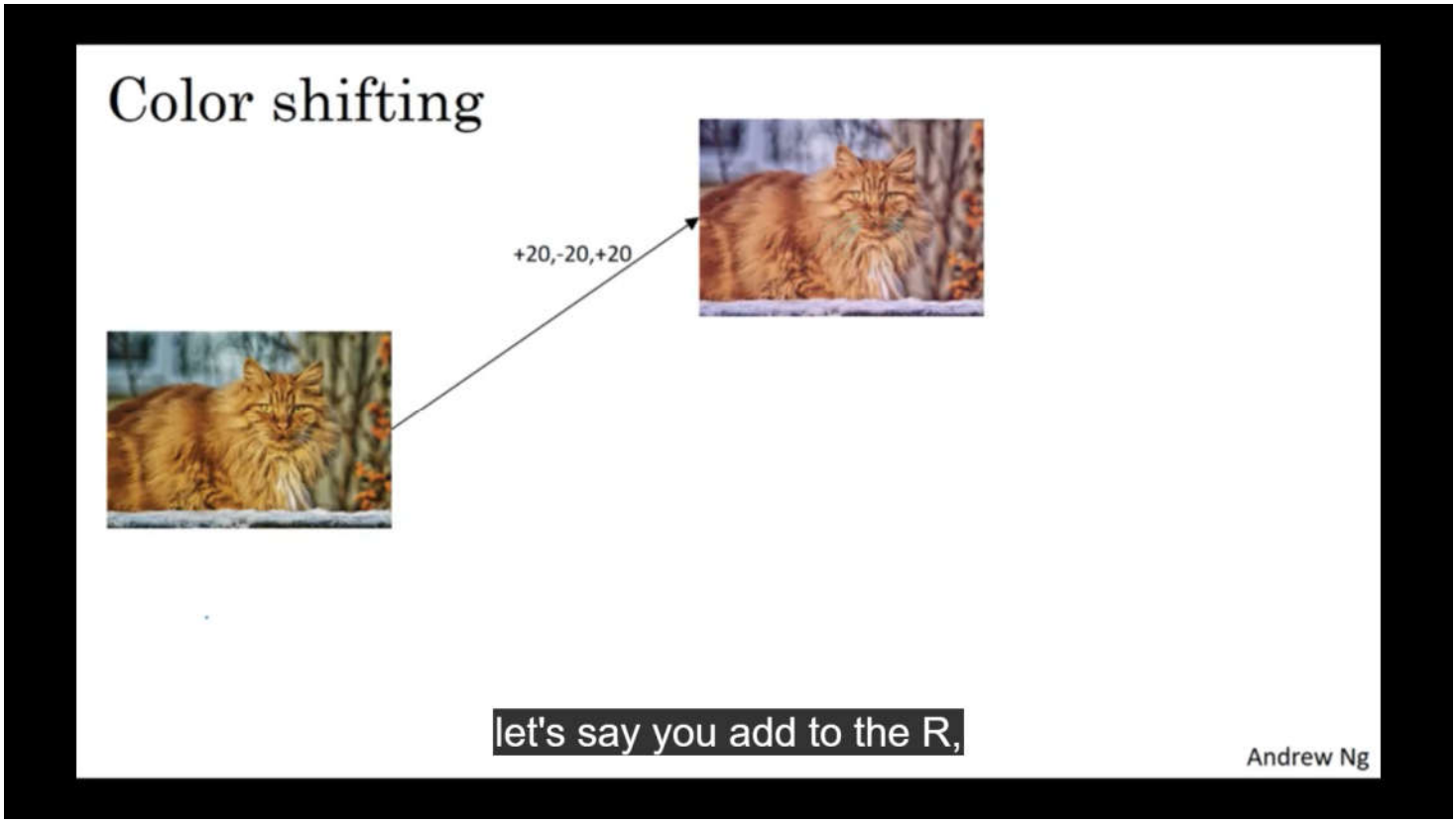

Common augmentation method
Mirroring
Random Cropping
y
So random cropping isn't a perfect data augmentation.
Andrew Ng

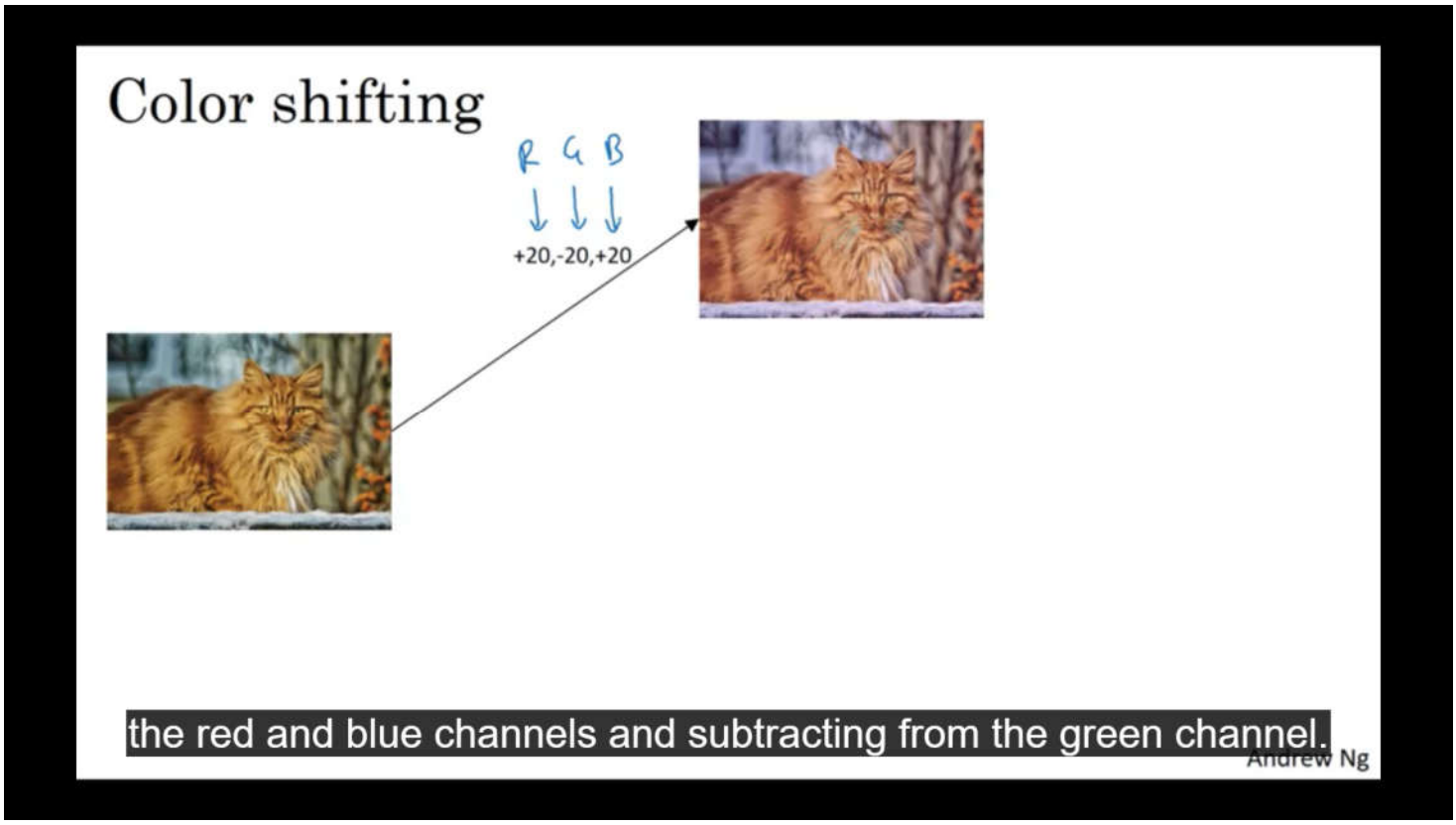

Common augmentation method
Mirroring
Random Cropping
y
your random crops are reasonably large subsets of the actual image.
Andrew Ng

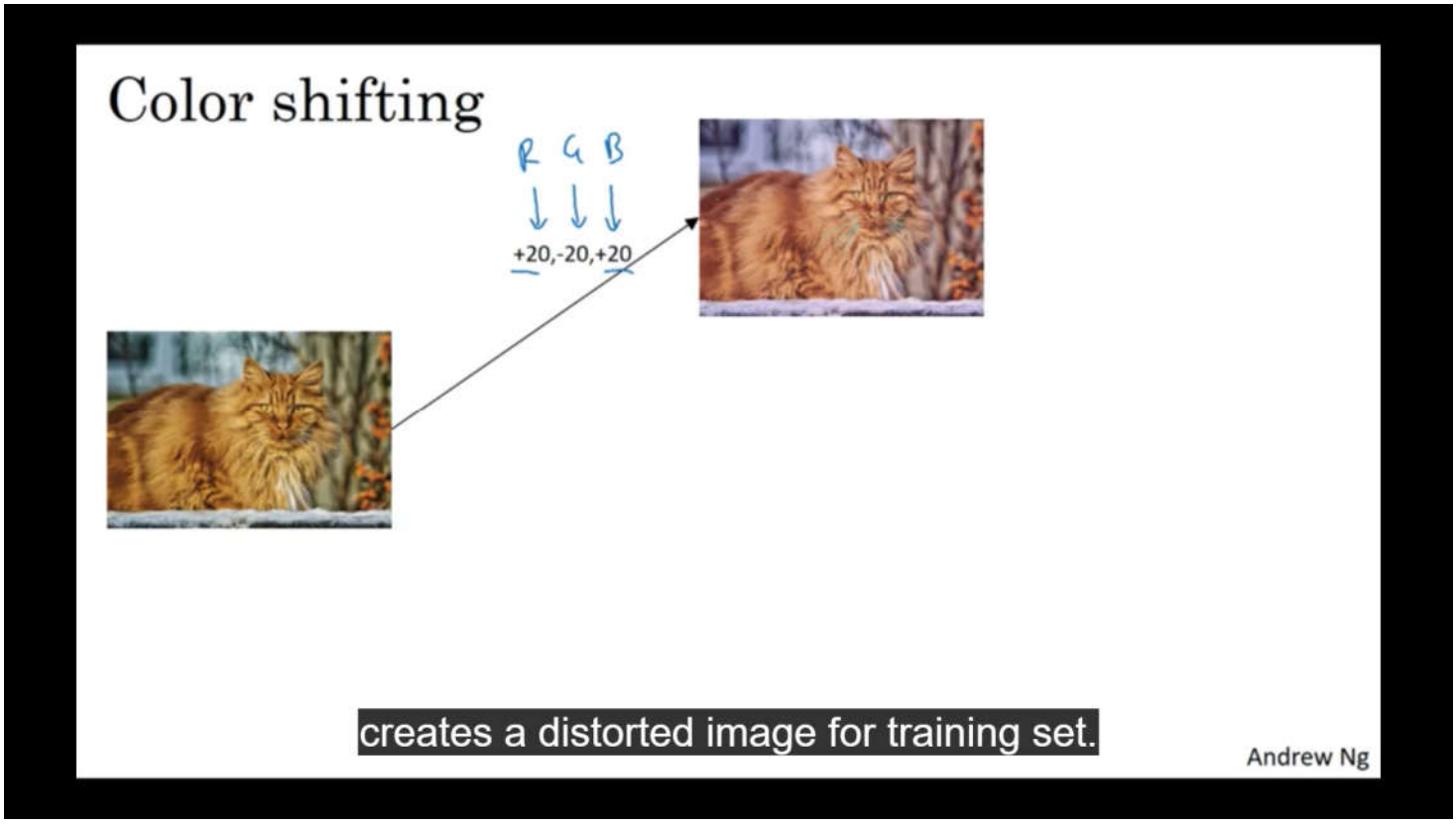

Common augmentation method
Mirroring
Random Cropping
Rotation
Shearing
y
shearing of the image,
Andrew Ng

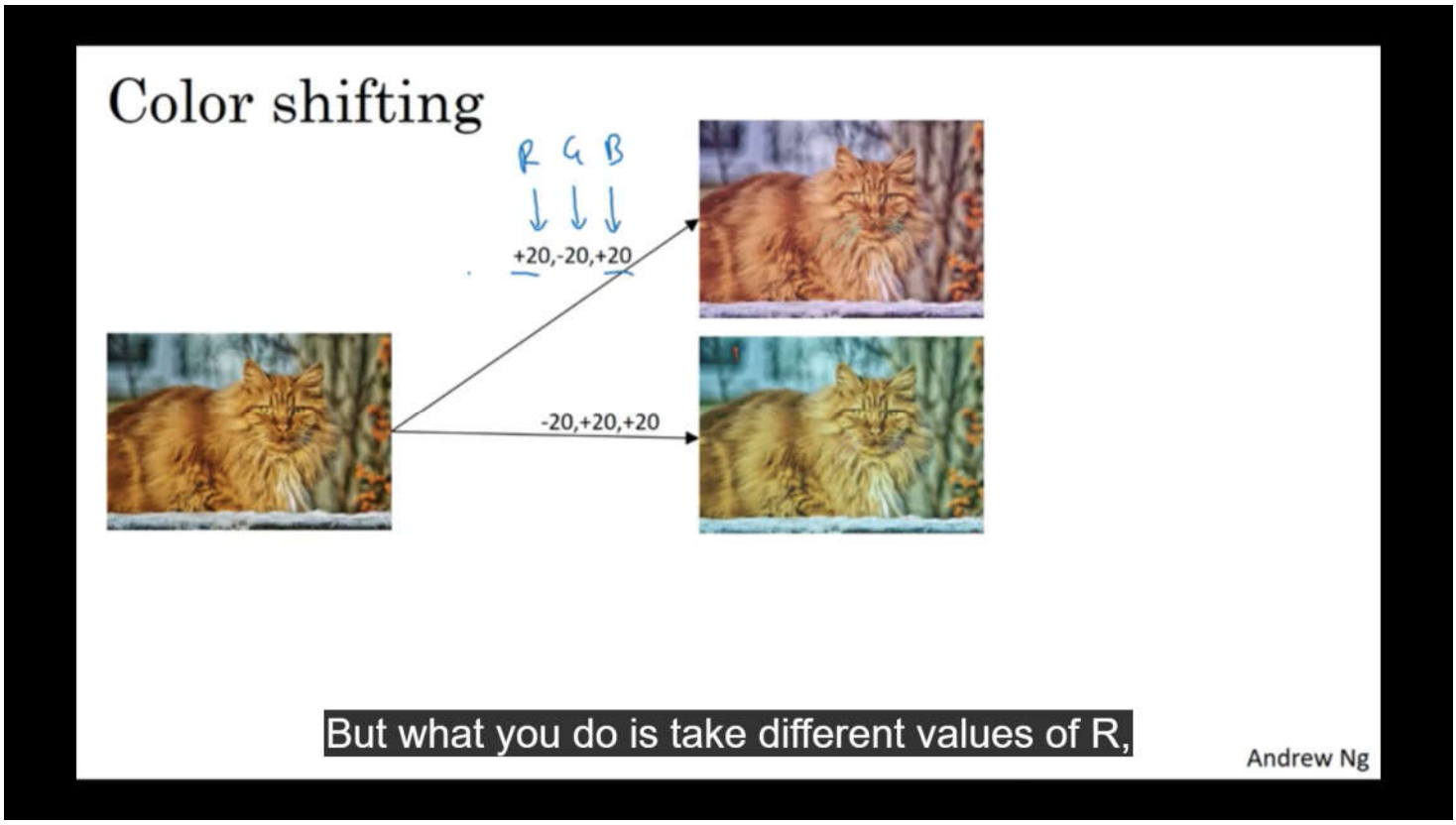

Common augmentation method
Mirroring
Random Cropping
Rotation
Shearing
Local warping
...
And there's really no harm with trying all of these things as well,
Andrew Ng

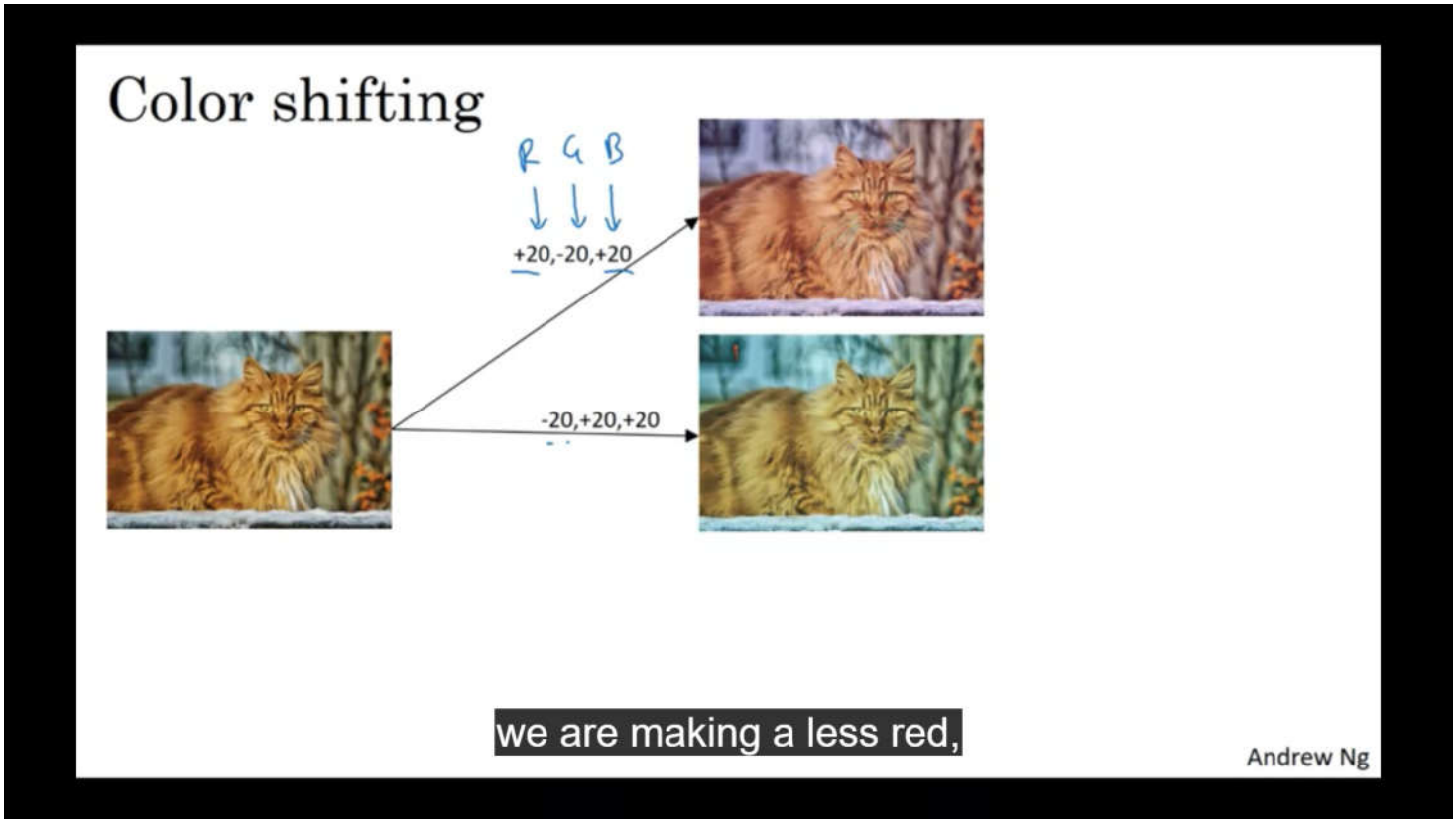

Color shifting
The second type of data augmentation that is commonly used is color shifting.
Andrew Ng

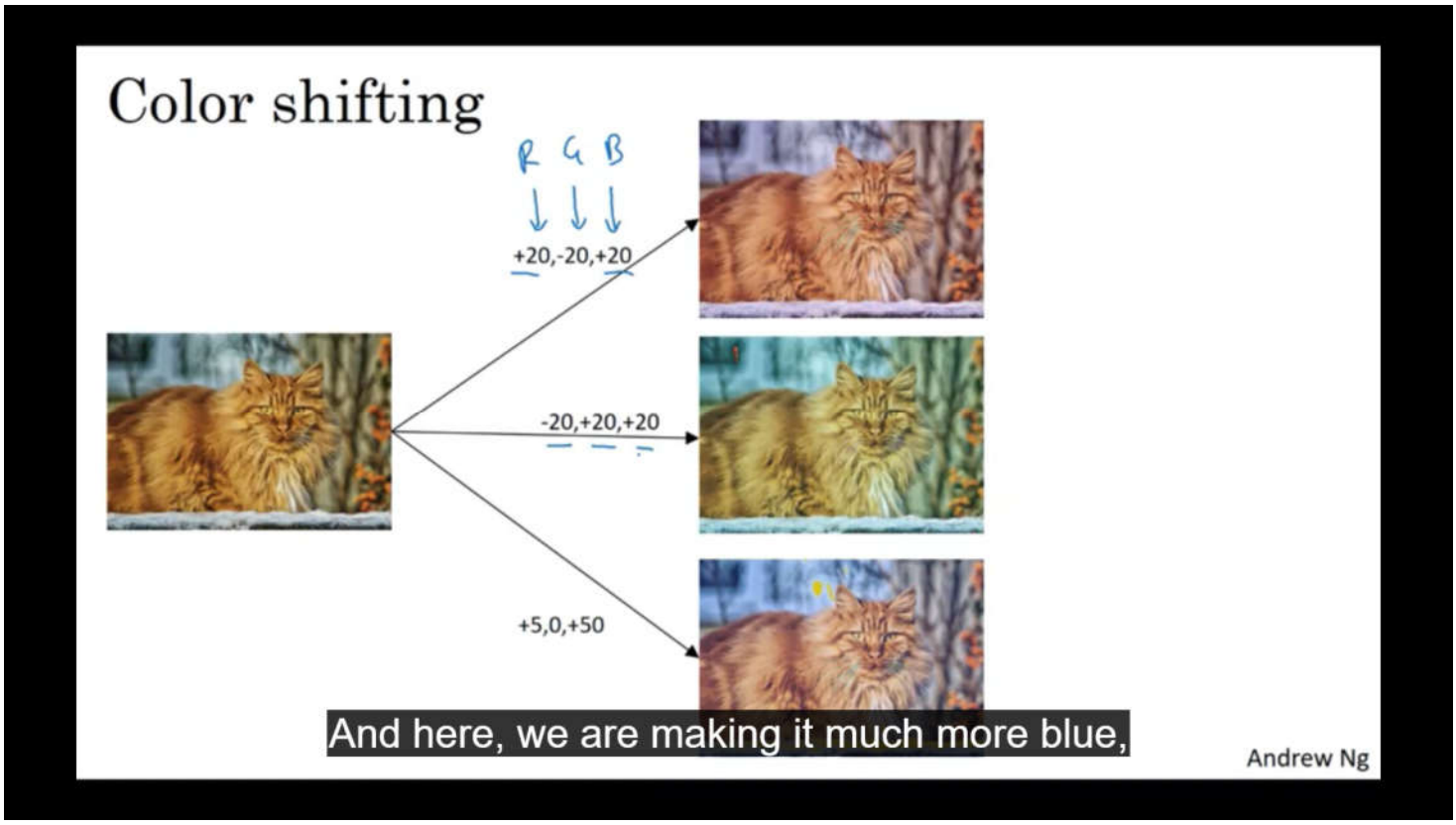

Color shifting
+20,-20,+20
let's say you add to the R,
Andrew Ng

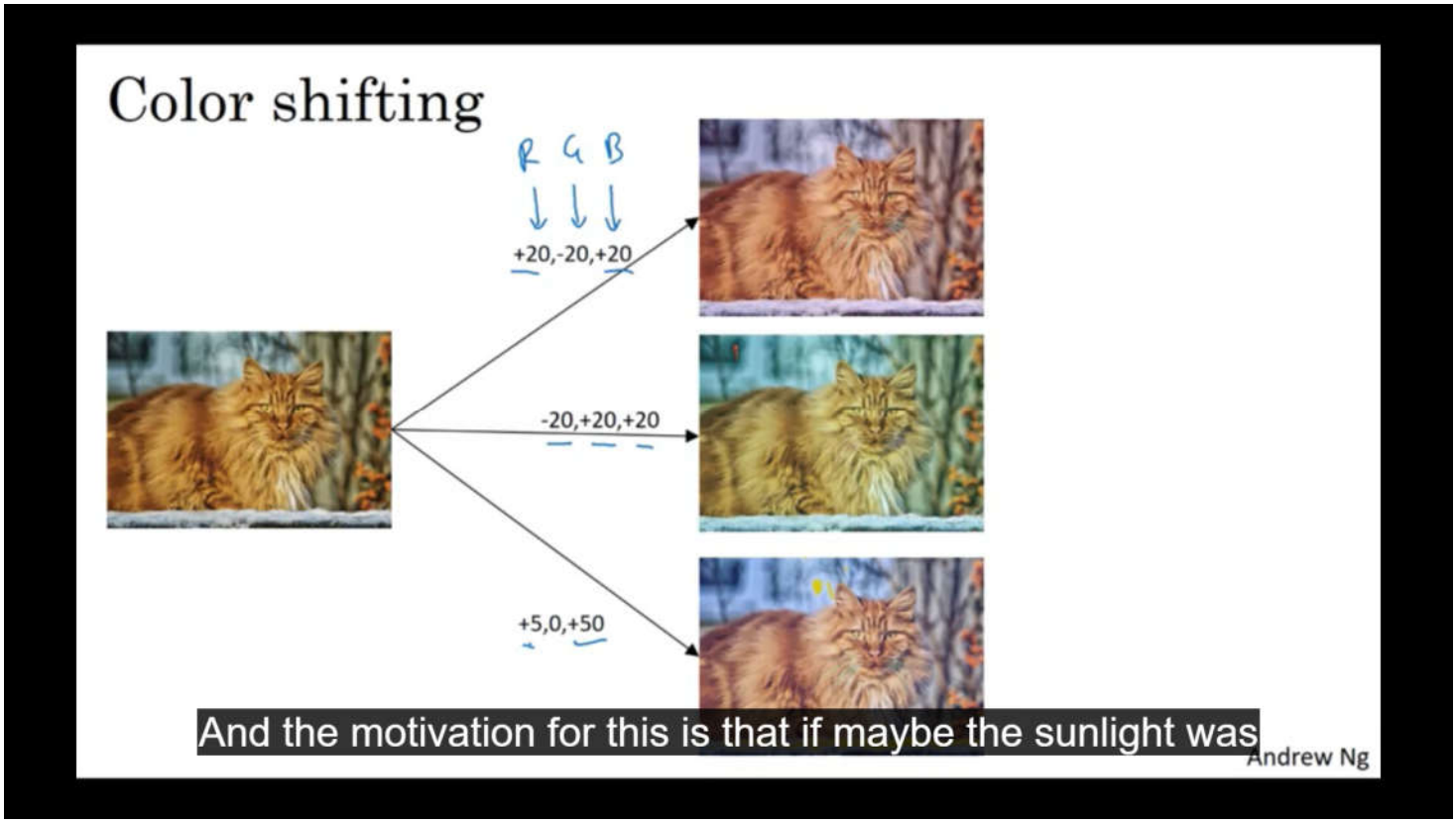

Color shifting
R G B
+20,-20,+20
the red and blue channels and subtracting from the green channel.
Andrew Ng

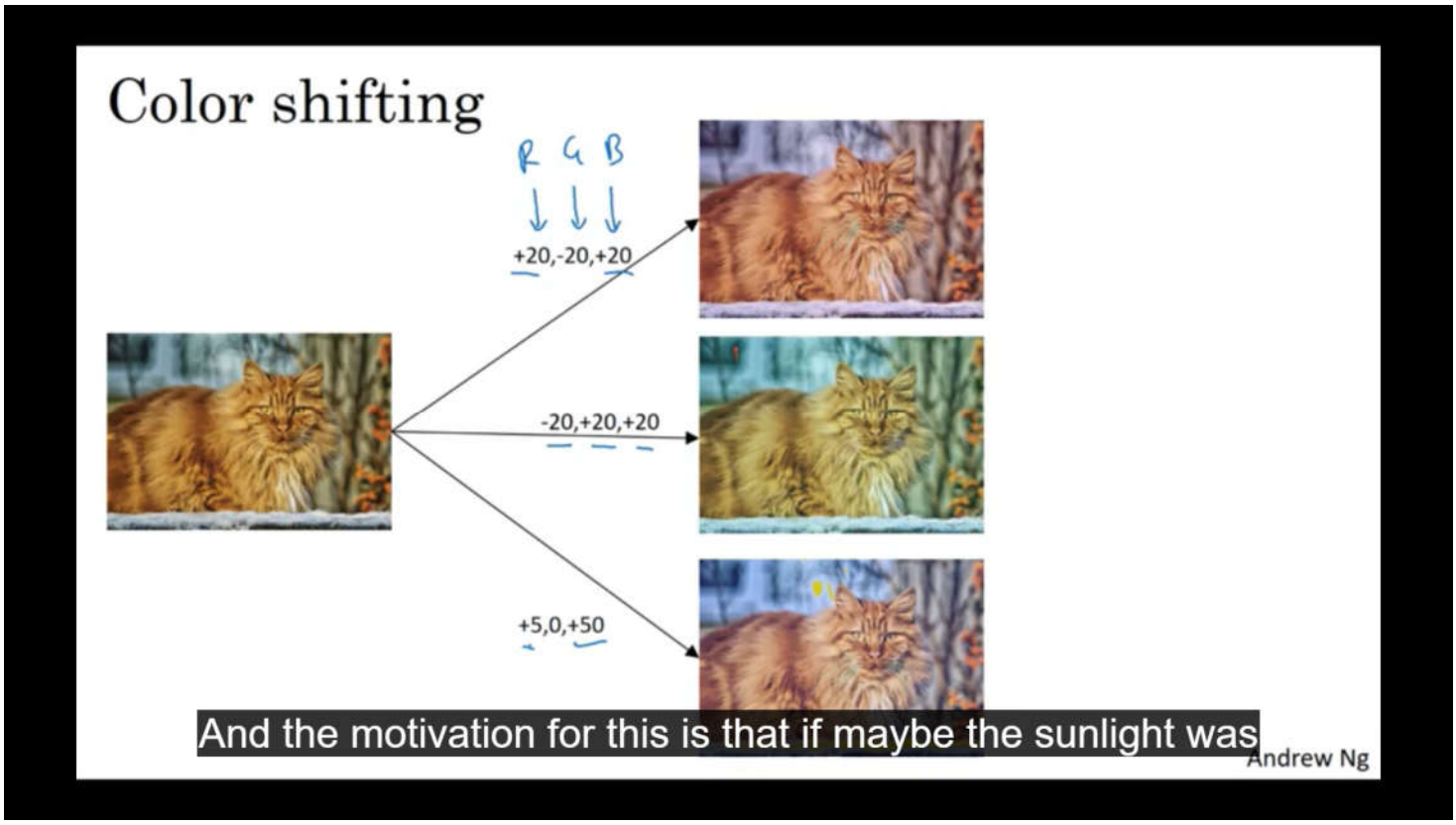

Color shifting
R G B
+20,-20,+20
creates a distorted image for training set.
Andrew Ng

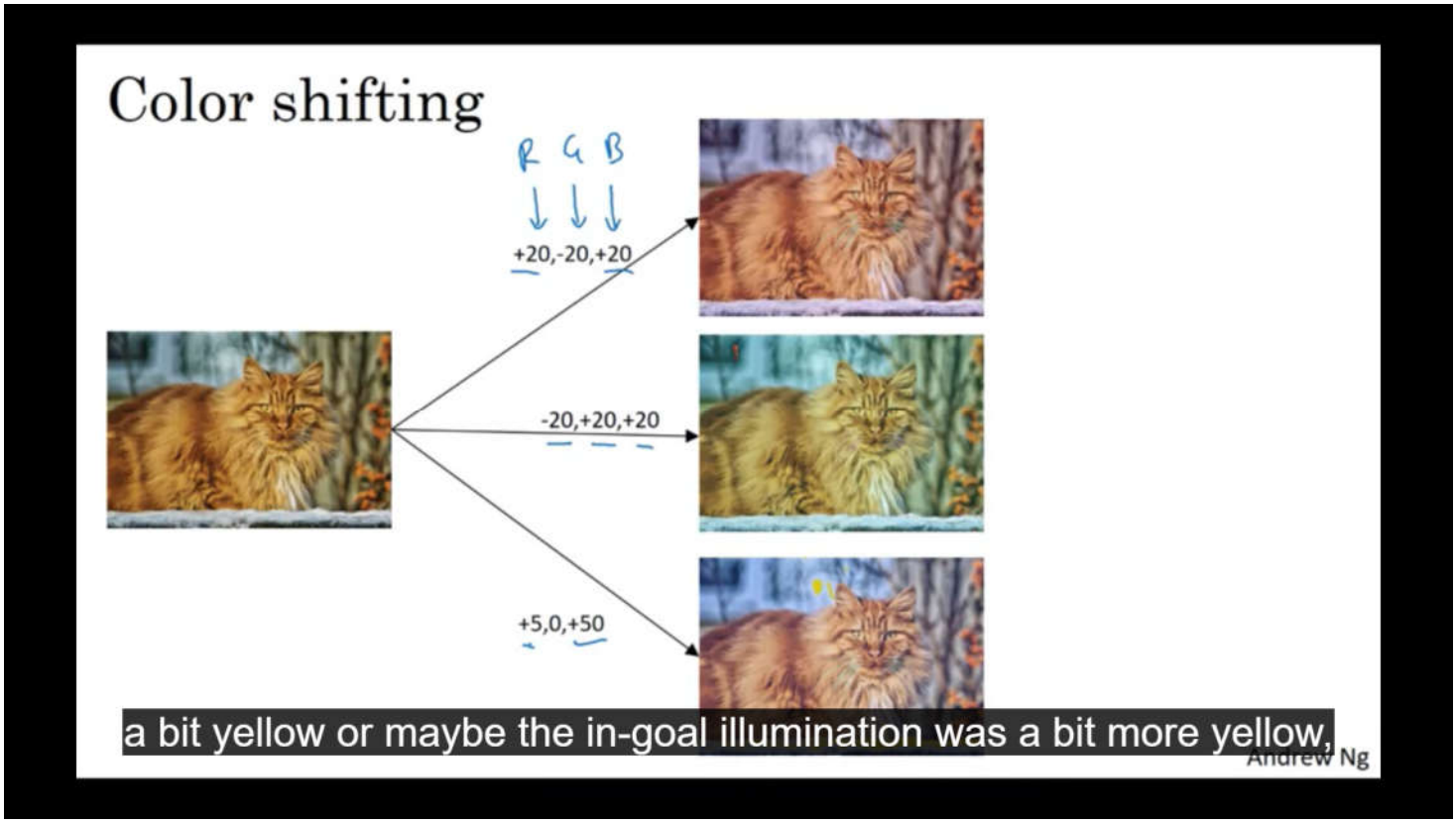

Color shifting
R G B
+20,-20,+20
-20,+20,+20
But what you do is take different values of R,
Andrew Ng

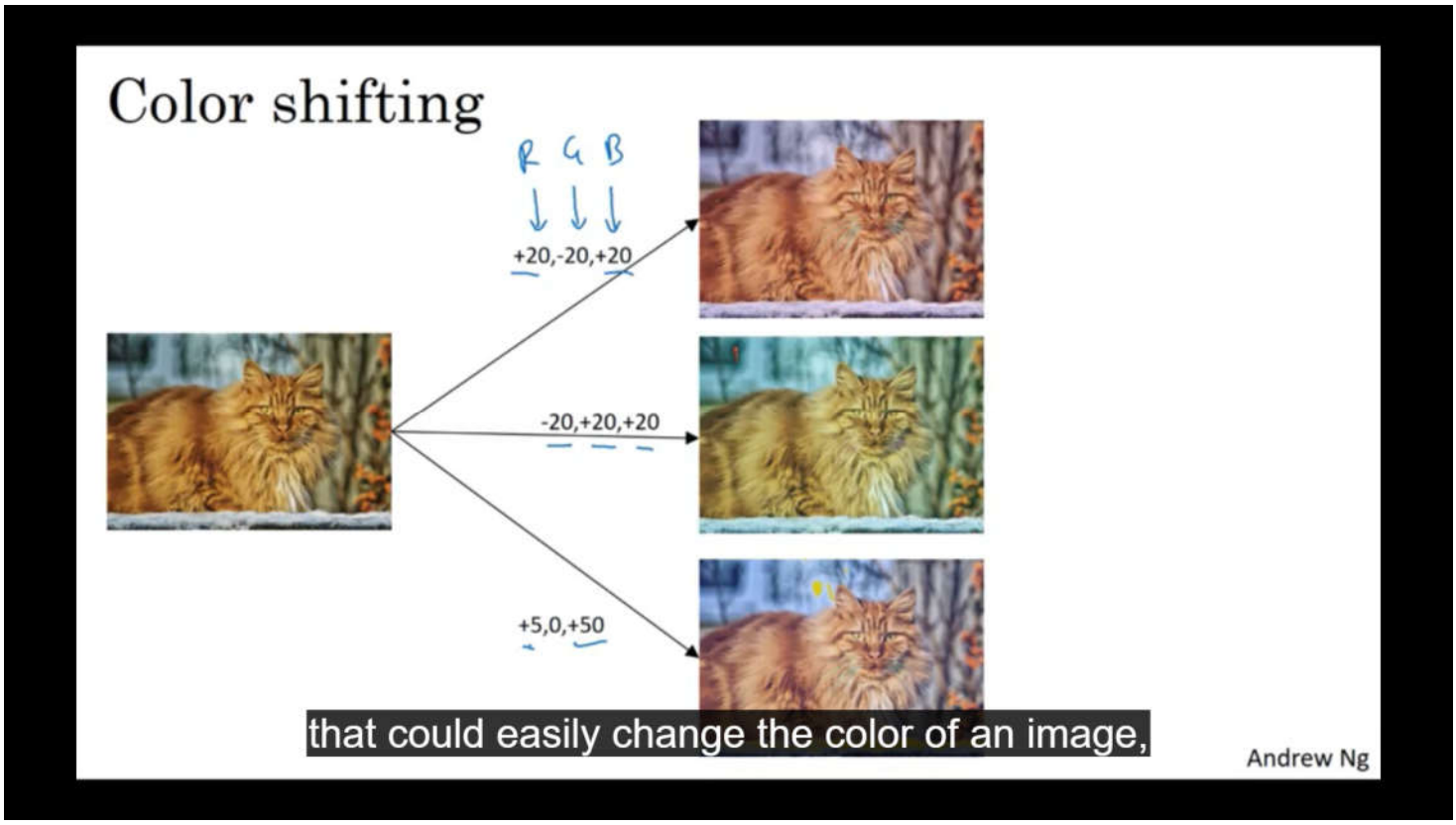

Color shifting
R G B
+20,-20,+20
-20,+20,+20
we are making a less red,
Andrew Ng

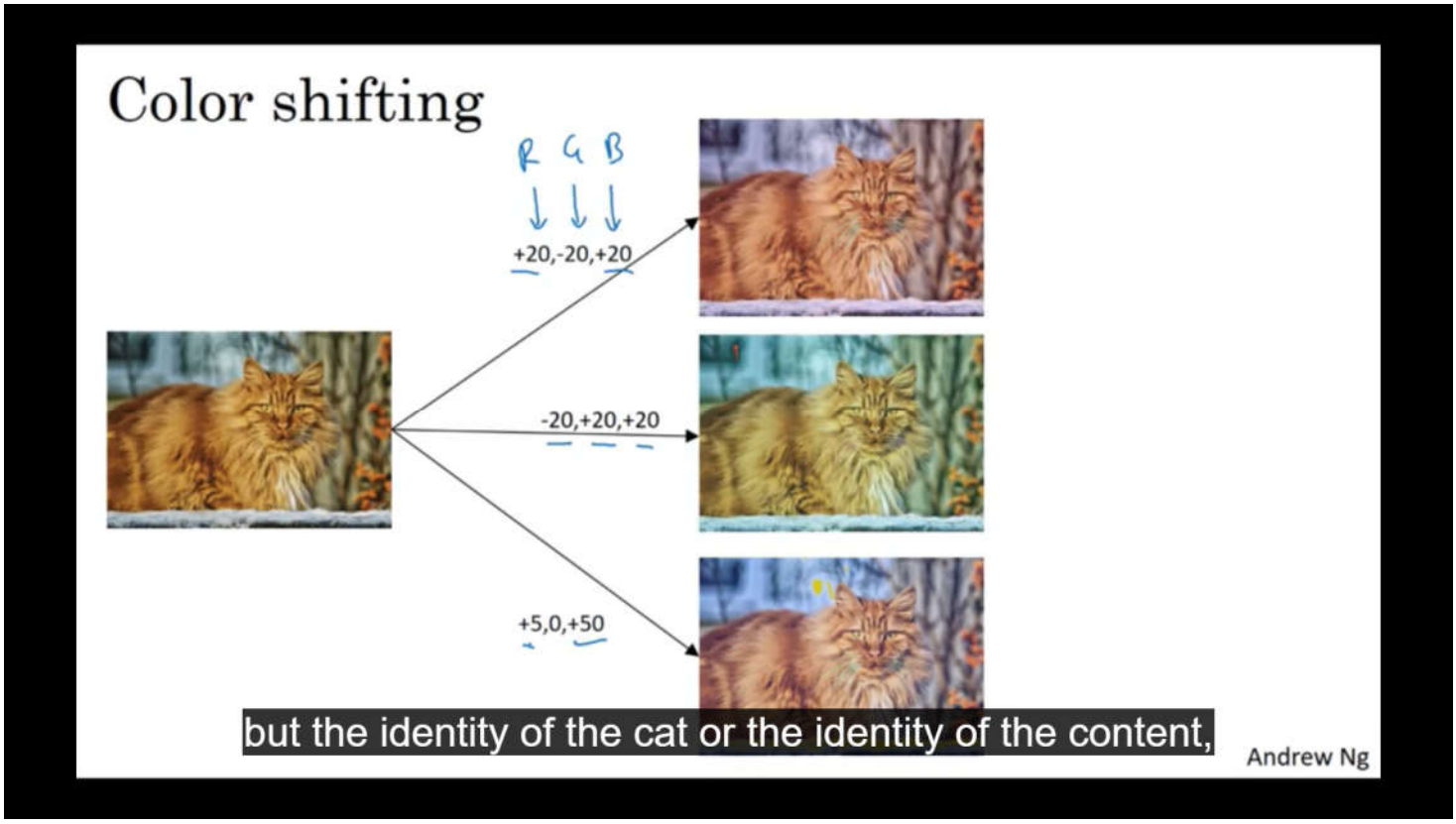

Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
And here, we are making it much more blue,
Andrew Ng


Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
And the motivation for this is that if maybe the sunlight was
Andrew Ng

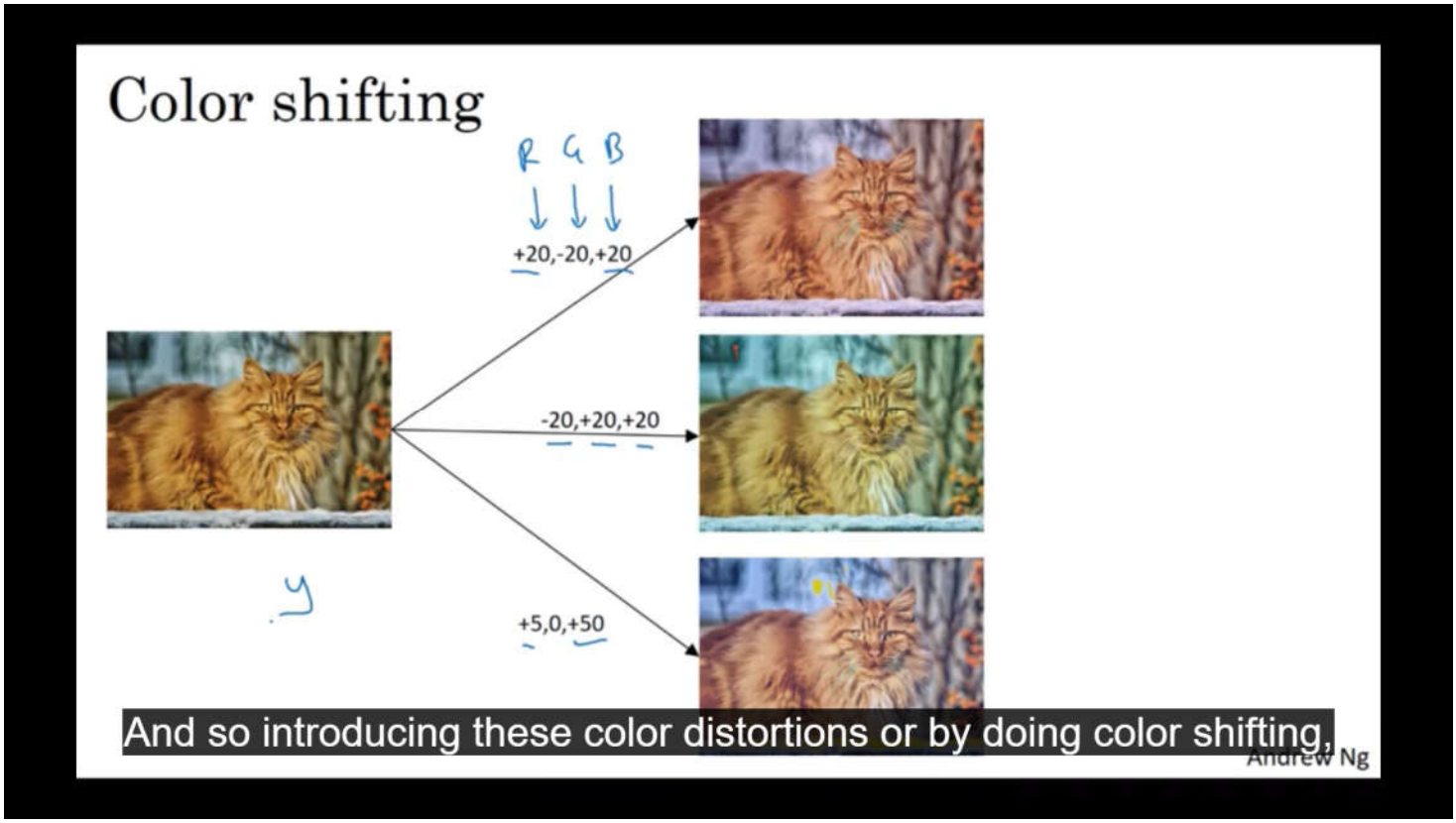

Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
And the motivation for this is that if maybe the sunlight was
Andrew Ng

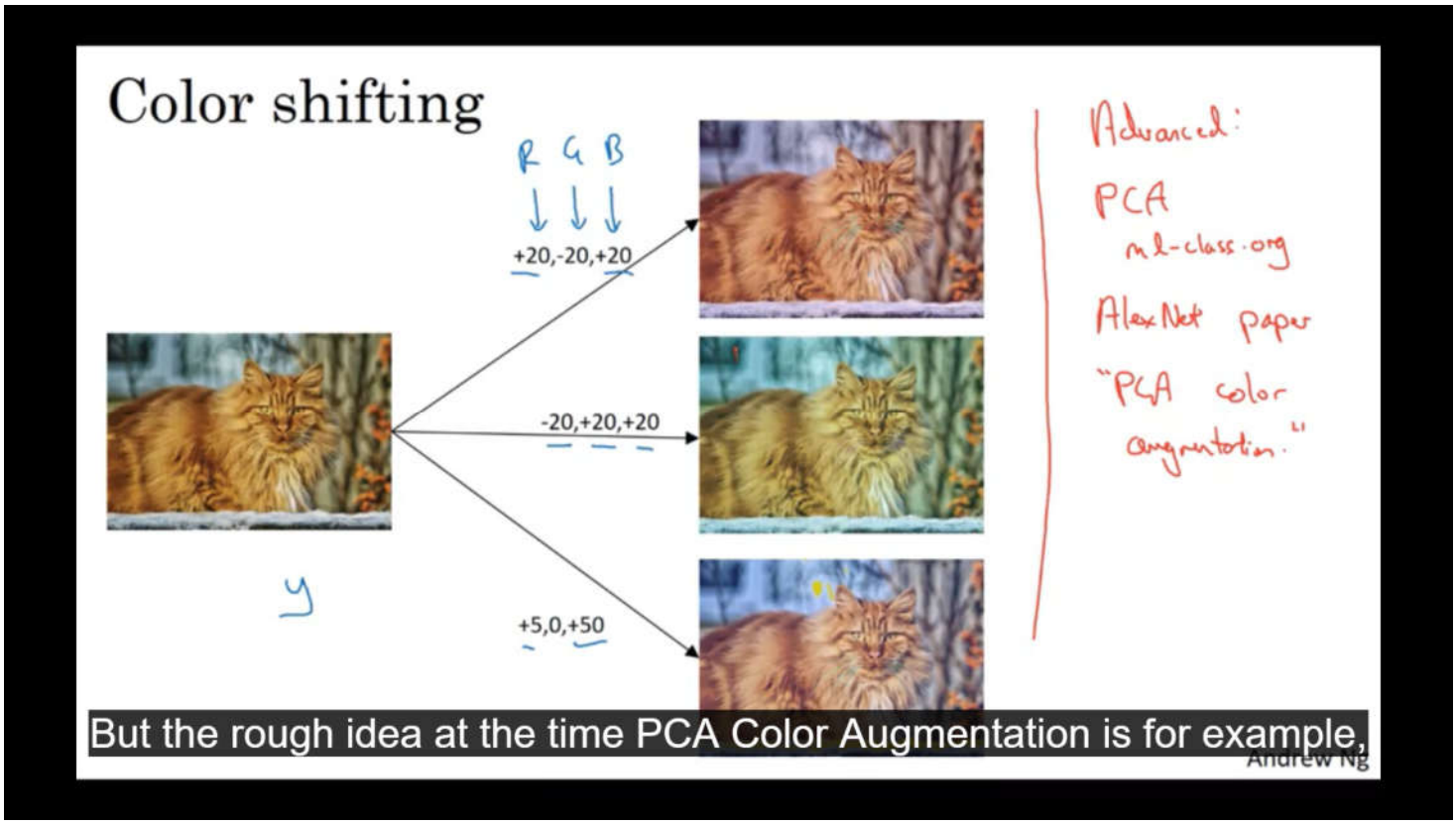

Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
a bit yellow or maybe the in-goal illumination was a bit more yellow,
Andrew Ng

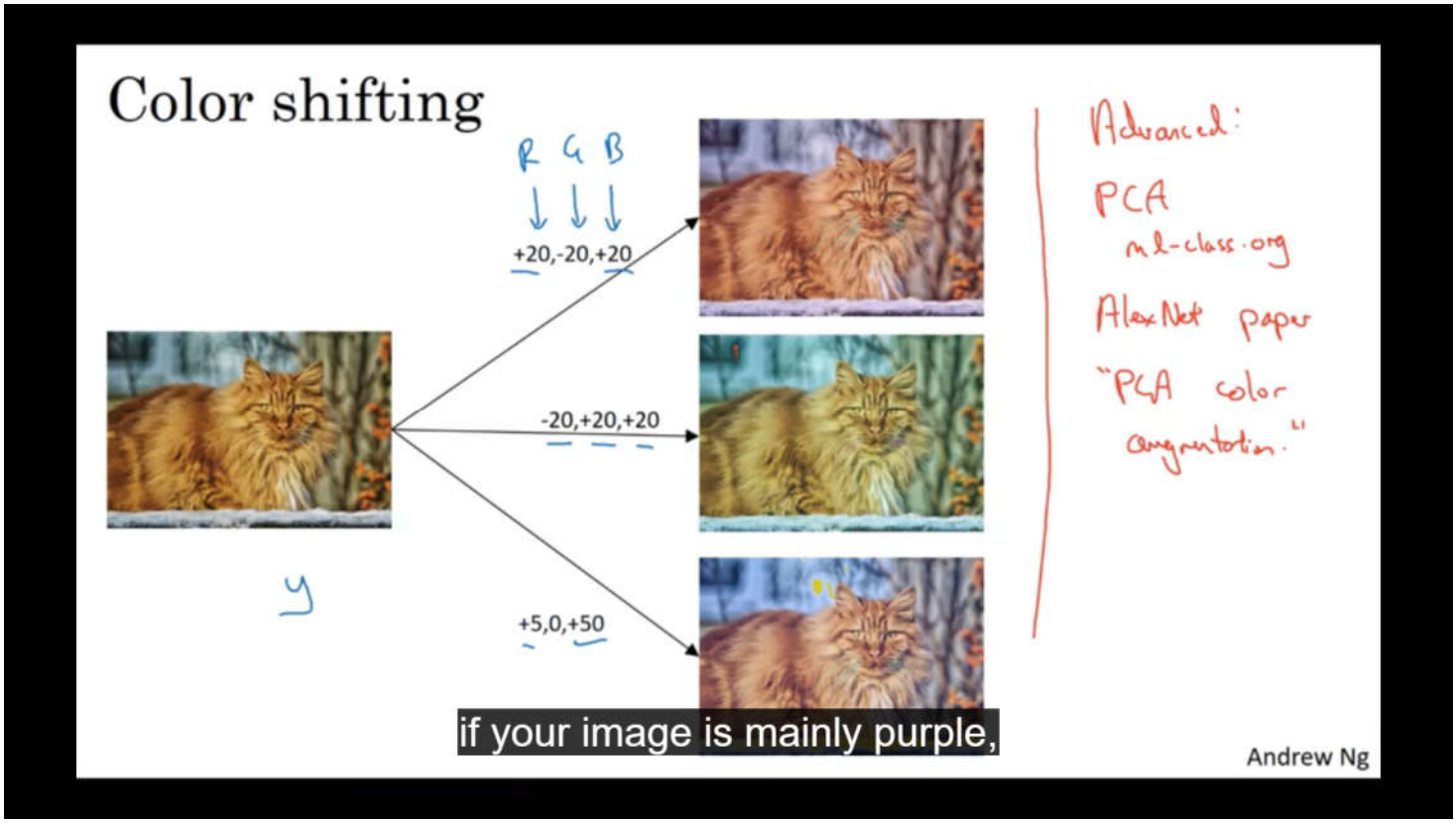

Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
that could easily change the color of an image,
Andrew Ng

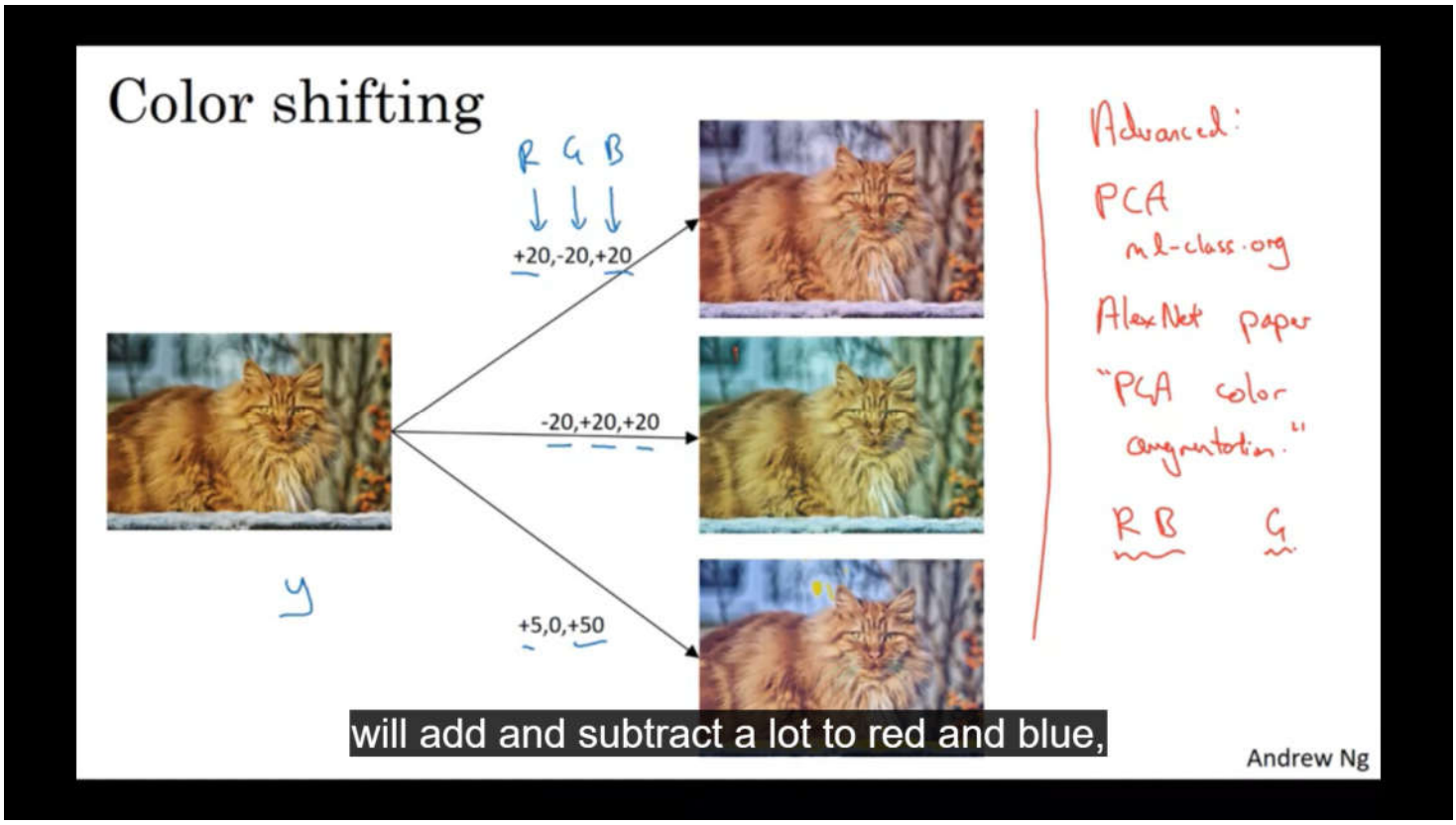

Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
but the identity of the cat or the identity of the content,
Andrew Ng

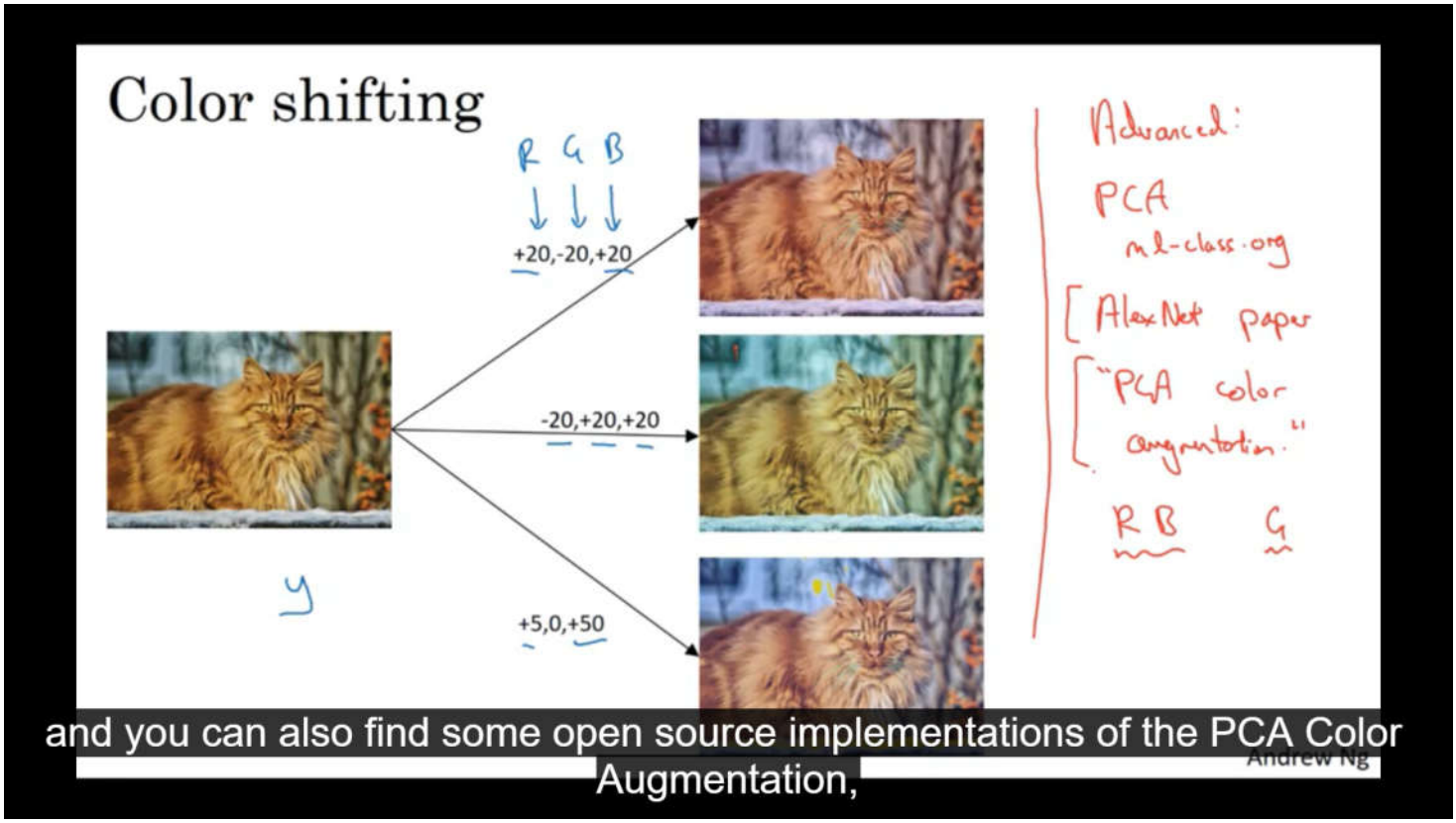
Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
y
the label y, just still stay the same.
Andrew Ng


Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
y
And so introducing these color distortions or by doing color shifting,
Andrew Ng

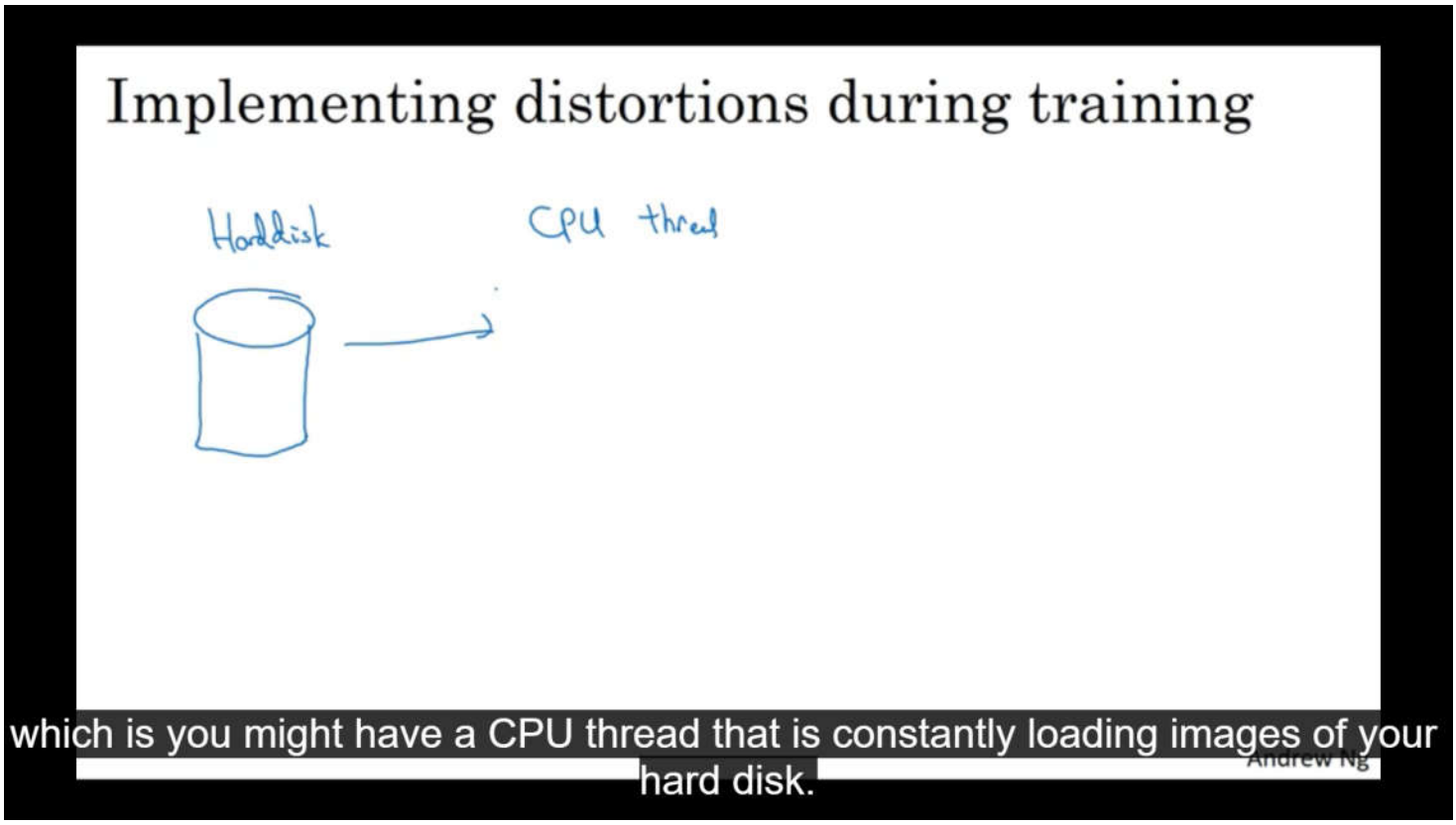

Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
y
Advanced:
PCA
ml-class.org
AlexNet paper
"PCA color augmentation."
But the rough idea at the time PCA Color Augmentation is for example,
Andrew Ng


Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
y
Advanced:
PCA
ml-class.org
AlexNet paper
"PCA color augmentation."
if your image is mainly purple,
Andrew Ng

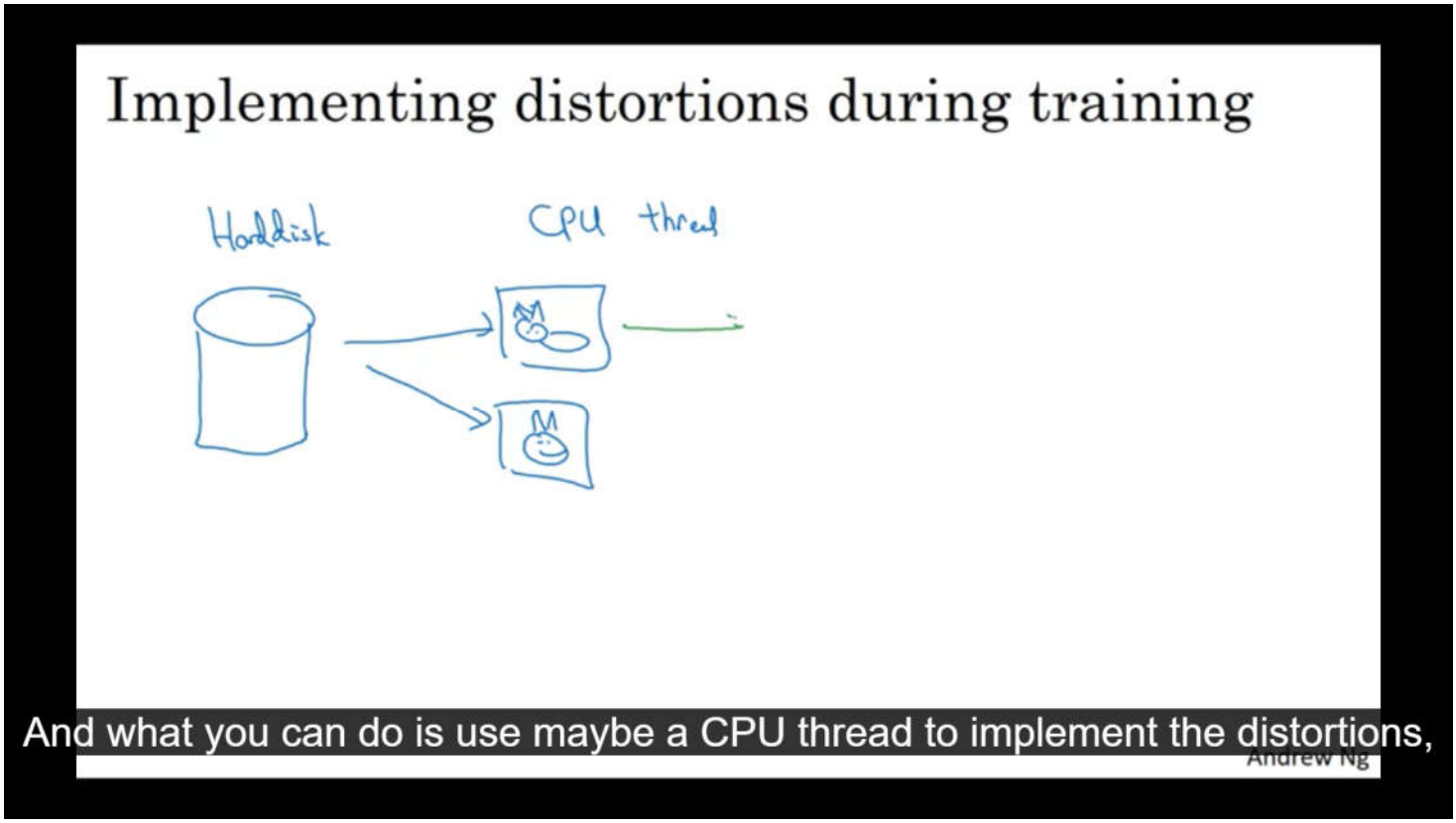

Color shifting
R G B
+20,-20,+20
-20,+20,+20
+5,0,+50
y
Advanced:
PCA
ml-class.org
AlexNet paper
"PCA color augmentation."
R B
G
will add and subtract a lot to red and blue,
Andrew Ng

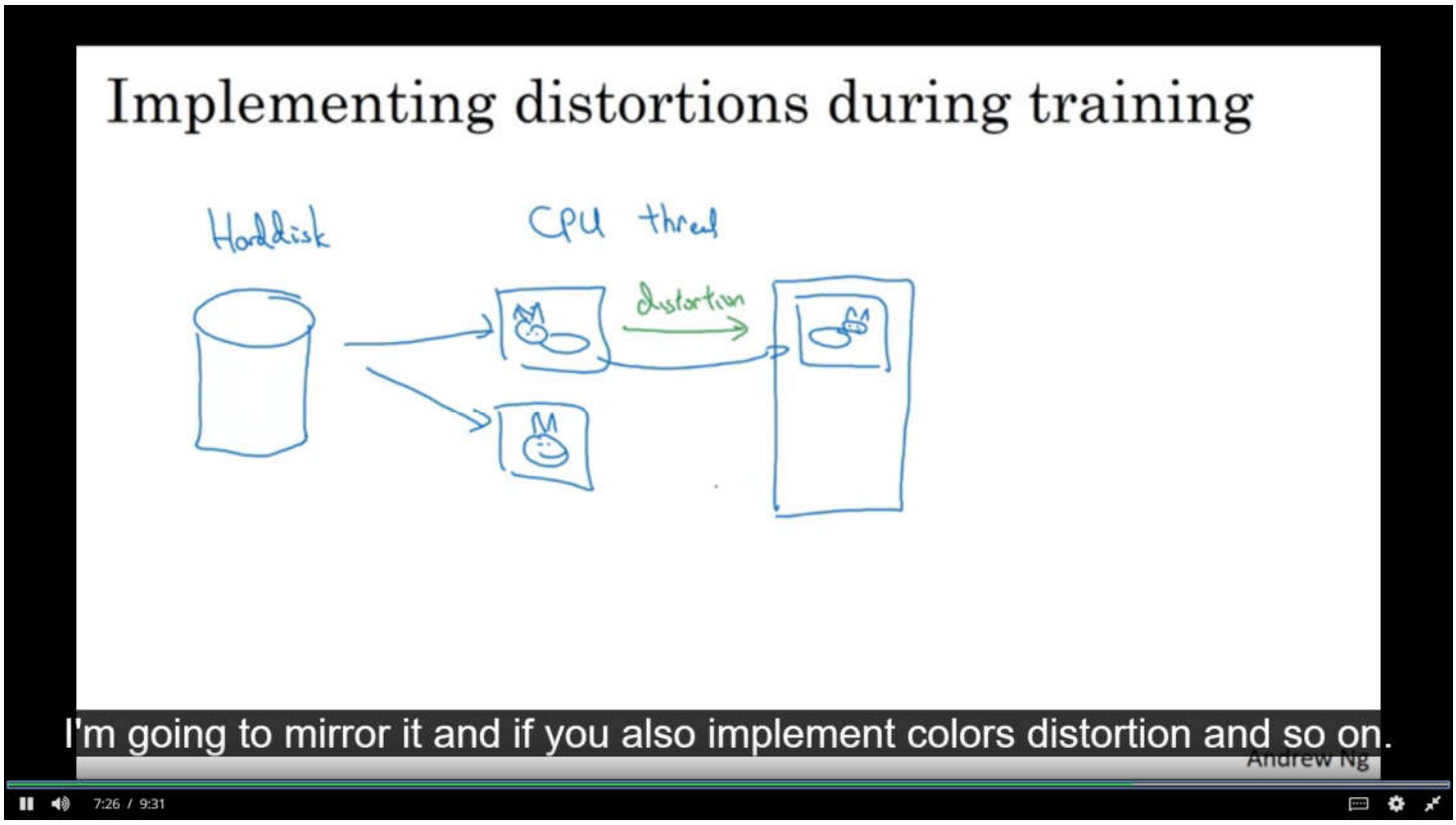


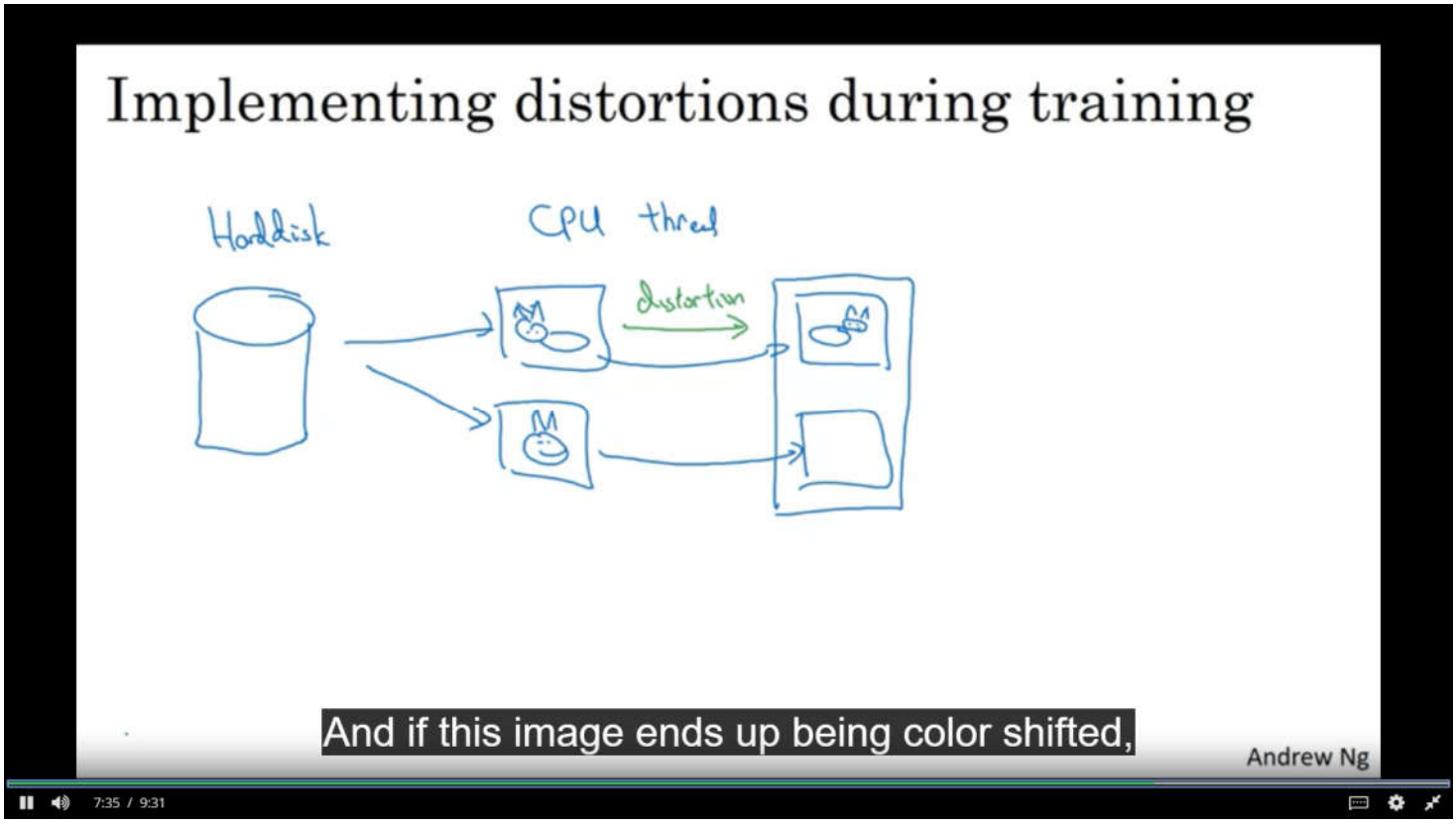
and you can also find some open source implementations of the PCA Color Augmentation,

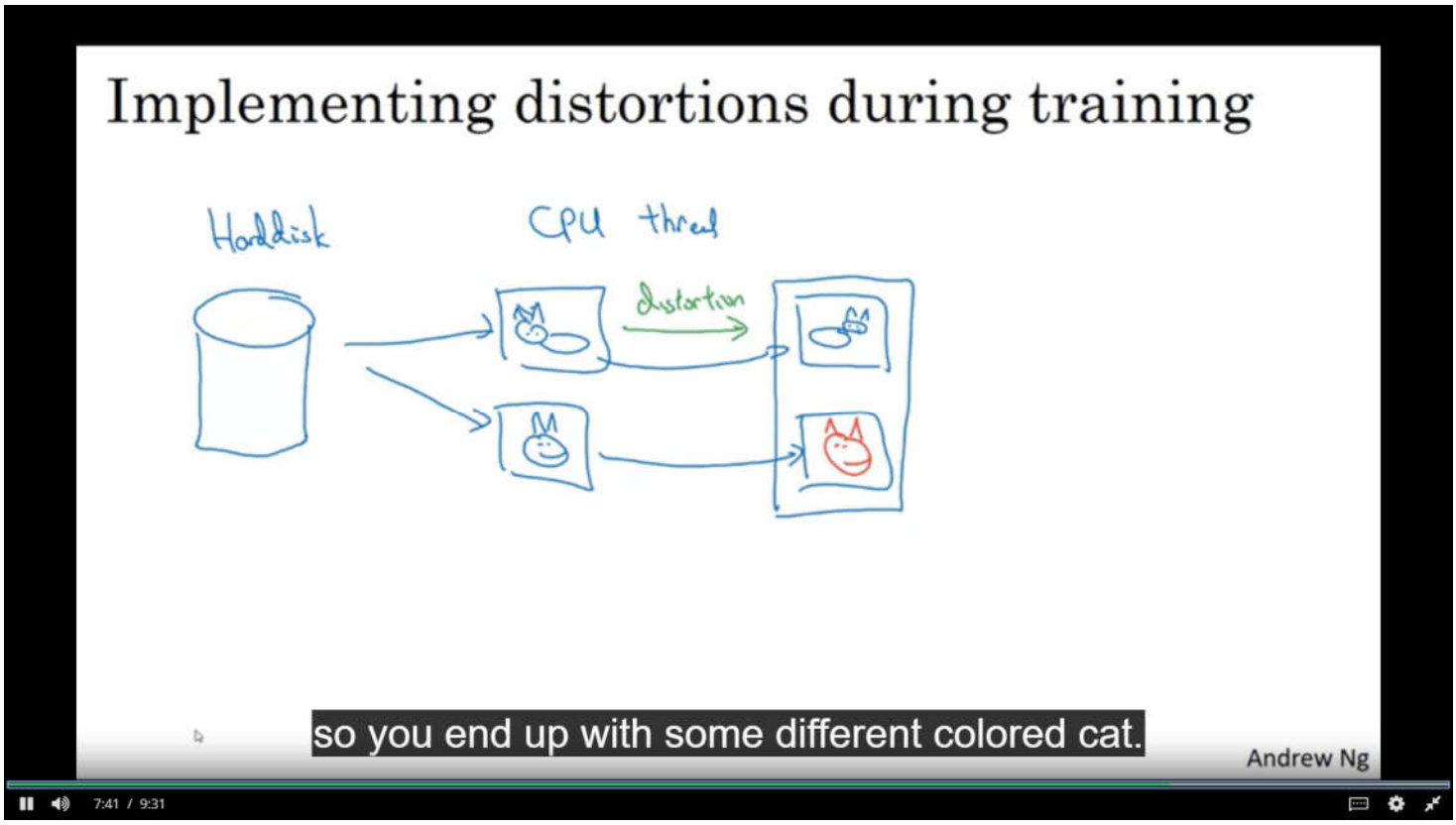


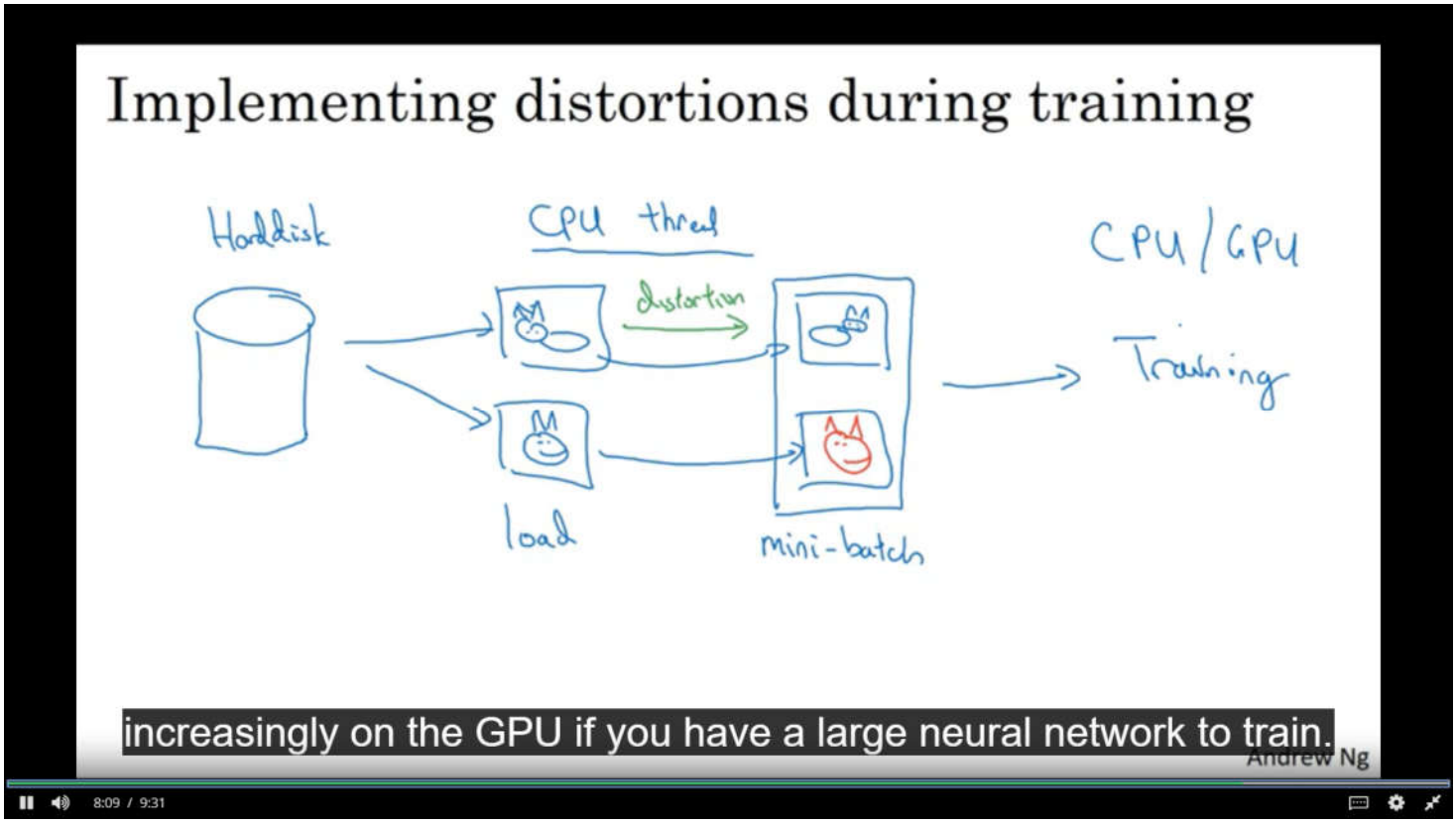
which is you might have a CPU thread that is constantly loading images of your hard disk.

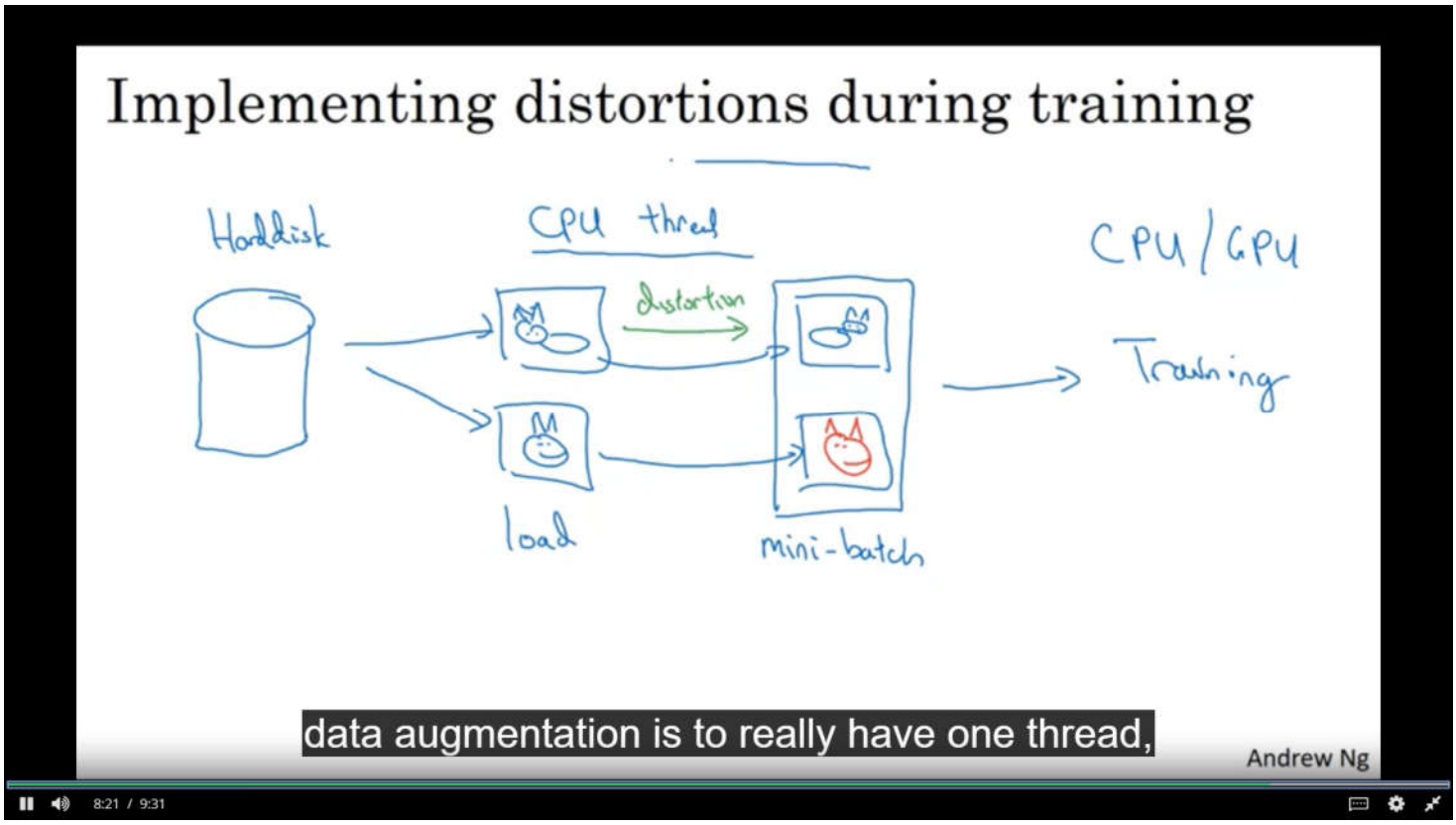

Implementing distortions during training
Harddisk
CPU thread
And what you can do is use maybe a CPU thread to implement the distortions,
Andrew Ng


Implementing distortions during training
Harddisk
CPU thread
distortion
I'm going to mirror it and if you also implement colors distortion and so on.
Andrew Ng
7:26 / 9:31


Implementing distortions during training
Harddisk
CPU thread
distortion
And if this image ends up being color shifted,
Andrew Ng
7:35 / 9:31


Implementing distortions during training
Harddisk
CPU thread
distortion
so you end up with some different colored cat.
Andrew Ng
7:41 / 9:31


Implementing distortions during training
Harddisk
CPU thread
distortion
CPU / GPU
Training
load
mini-batch
increasingly on the GPU if you have a large neural network to train.
Andrew Ng
8:09 / 9:31


Implementing distortions during training
Harddisk
CPU thread
distortion
CPU / GPU
Training
load
mini-batch
data augmentation is to really have one thread,
Andrew Ng
8:21 / 9:31

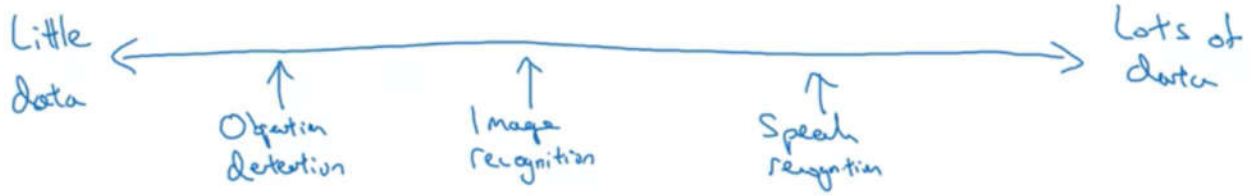
deeplearning.ai

Practical advice for using ConvNets

The state of computer vision

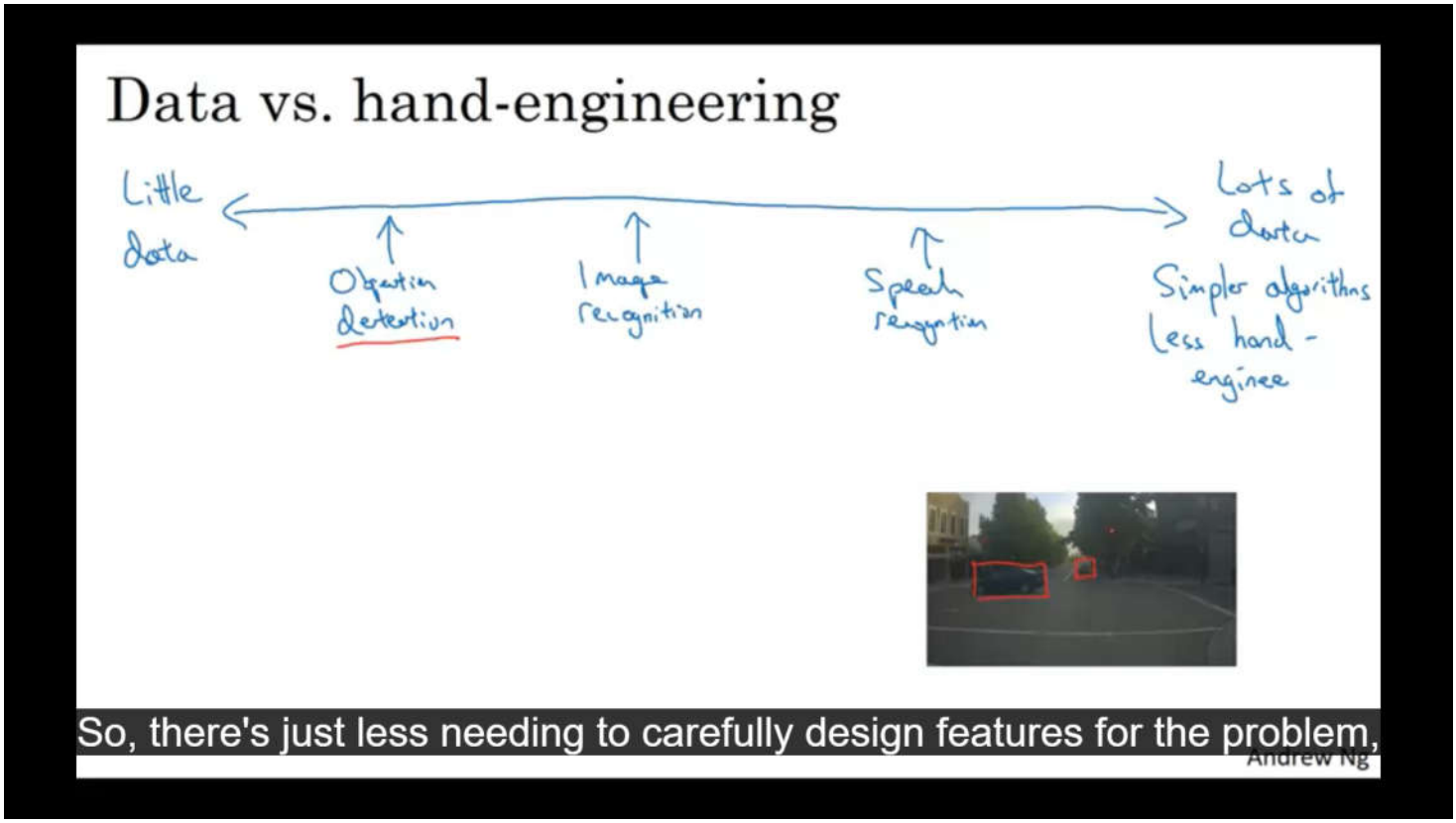
and how you build these systems yourself for computer vision.

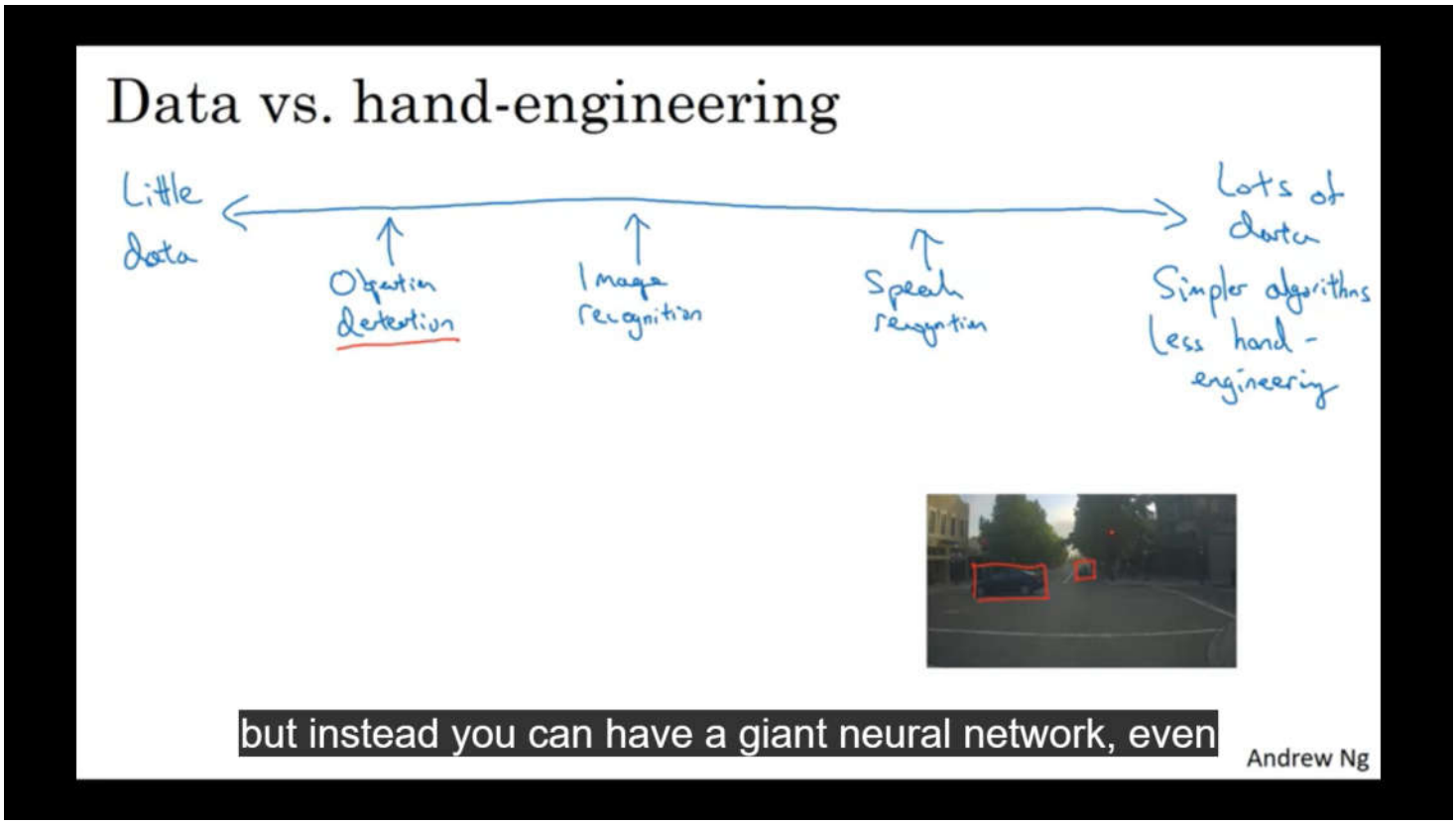
Data vs. hand-engineering

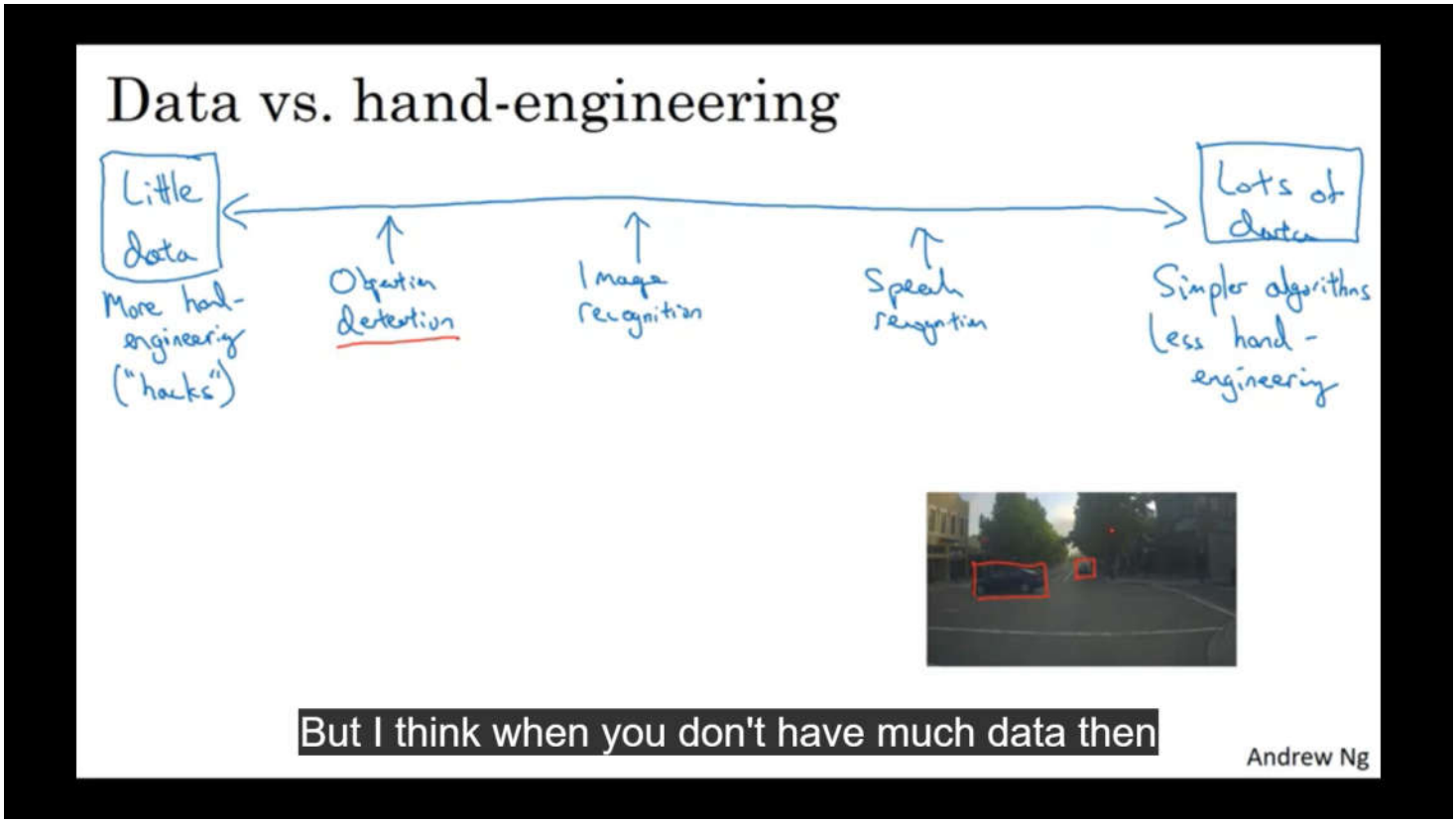


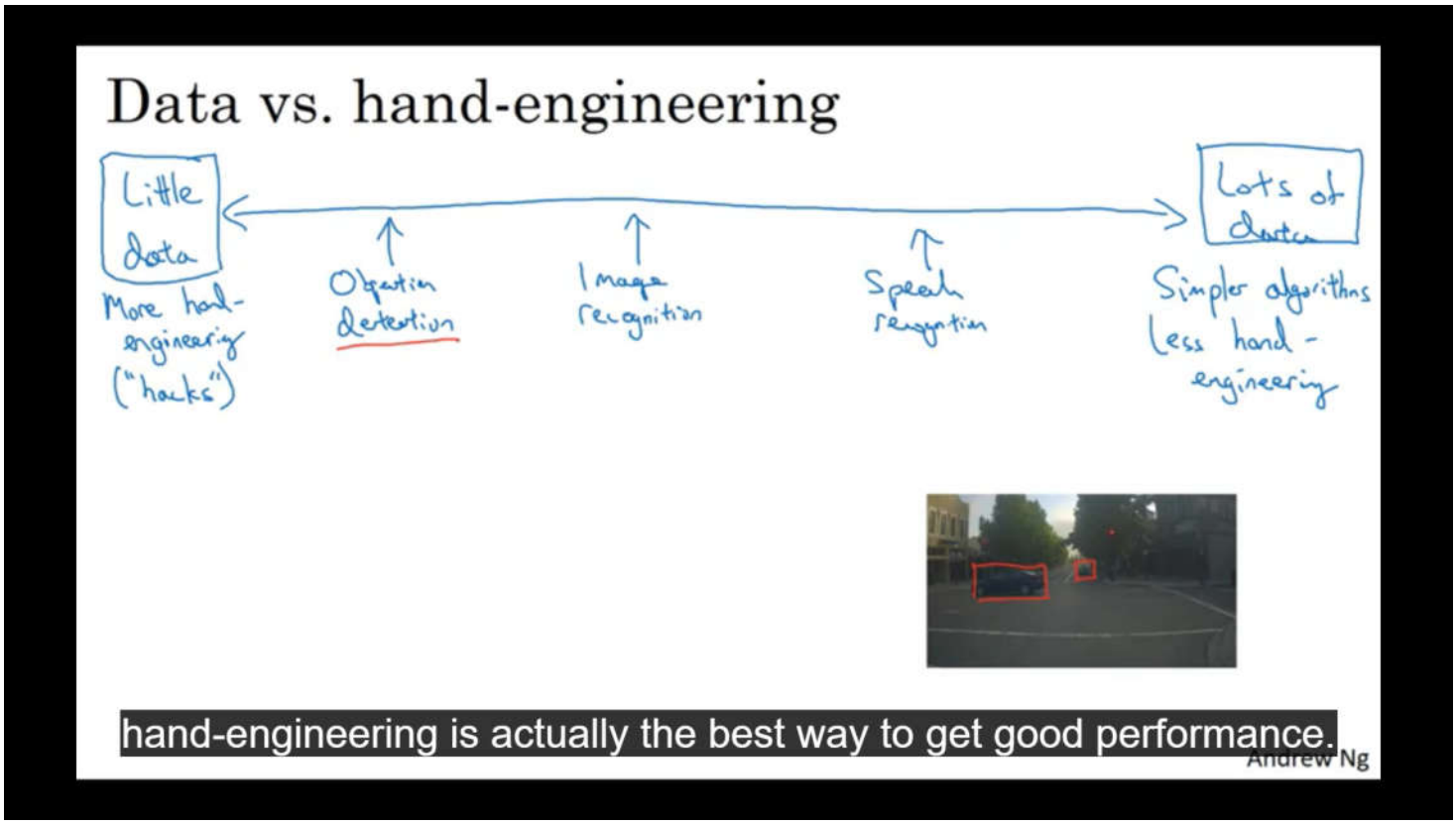
Whereas object detection is look in the picture and actually you're putting

Andrew Ng


Data vs. hand-engineering
Little data
Lots of data
Object detection
Image recognition
Speech recognition
Simpler algorithms
Less hand-engineering
So, there's just less needing to carefully design features for the problem,
Andrew Ng


Data vs. hand-engineering
Little data
Lots of data
Object detection
Image recognition
Speech recognition
Simpler algorithms
Less hand-engineering
but instead you can have a giant neural network, even
Andrew Ng


Data vs. hand-engineering
Little data
Lots of data
More hand-engineering ("hacks")
Object detection
Image recognition
Speech recognition
Simpler algorithms
Less hand-engineering
But I think when you don't have much data then
Andrew Ng


Data vs. hand-engineering
Little data
Lots of data
More hand-engineering ("hacks")
Object detection
Image recognition
Speech recognition
Simpler algorithms
Less hand-engineering
hand-engineering is actually the best way to get good performance.
Andrew Ng

Data vs. hand-engineering

Little data

More hand-engineering ("hacks")

Object detection

Image recognition

Speech recognition

Lots of data

Simpler algorithms
Less hand-engineering

Two sources of knowledge

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

Andrew Ng

3:20 / 12:37

Tips for doing well on benchmarks/winning competitions

standardized benchmark data sets and on winning competitions.

Andrew Ng

Tips for doing well on benchmarks/winning competitions

But you also see in the papers people do things that allow you to do well on a benchmark,

Andrew Ng

Tips for doing well on benchmarks/winning competitions

Ensembling

- Train several networks independently and average their outputs

train several neural networks independently and average their outputs.

Andrew Ng

Tips for doing well on benchmarks/winning competitions

Ensembling

3-15 networks

$\rightarrow \hat{y}$

- Train several networks independently and average their outputs

Multi-crop at test time

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

So, what I mean by that is you've seen how you can do data augmentation.

Andrew Ng

Tips for doing well on benchmarks/winning competitions

Ensembling

3-15 networks

$\rightarrow \hat{y}$

- Train several networks independently and average their outputs

Multi-crop at test time

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

So, for example let's see a cat image

Andrew Ng

Tips for doing well on benchmarks/winning competitions

Ensembling

3-15 networks

- Train several networks independently and average their outputs → $\hat{y}$

Multi-crop at test time

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

10-crop

and run that through the crossfire.

Andrew Ng

Use open source code

- Use architectures of networks published in the literature
- Use open source implementations if possible

And you can use an open source implementation if possible,

Andrew Ng

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

And finally someone else may have spent weeks training a model

Andrew Ng

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

But of course if you have the compute resources and the inclination,

Andrew Ng