Week4

06 May 2020 18:44

Press Esc to exit full screen

deeplearning.ai

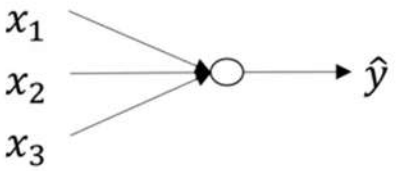
Deep Neural Networks

Deep L-layer Neural network

Welcome to the fourth week of this course.

0:00 / 5:50

What is a deep neural network?

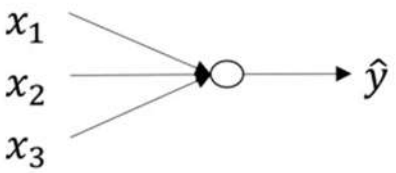


logistic regression

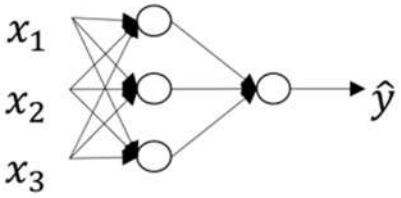
So what is a deep neural network?

Andrew Ng

What is a deep neural network?



logistic regression

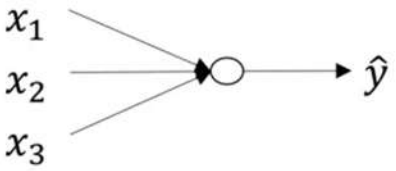


1 hidden layer

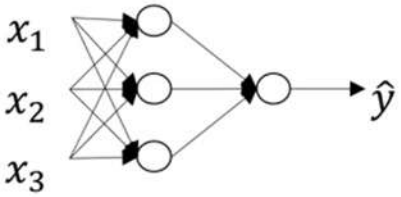
you've also seen neural networks
with a single hidden layer.

Andrew Ng

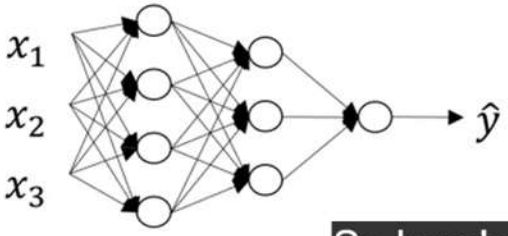
What is a deep neural network?



logistic regression



1 hidden layer

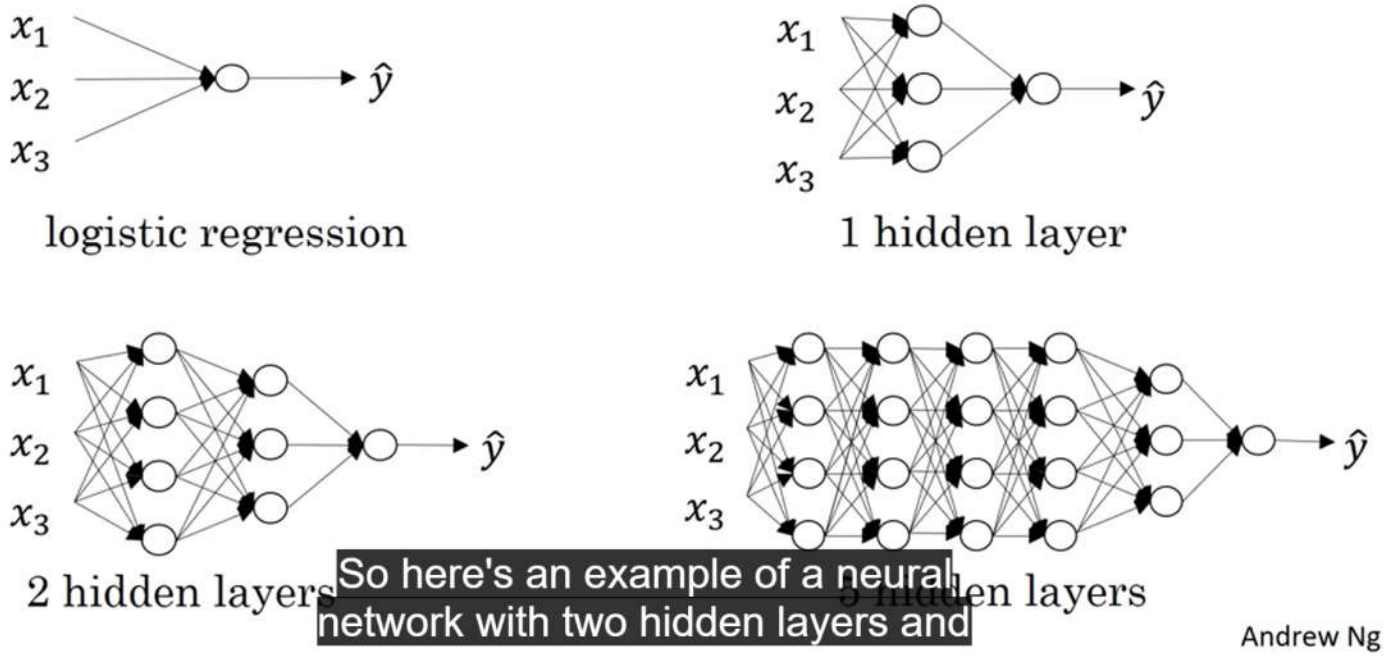


2 hidden layers

So here's an example of a neural
network with two hidden layers and

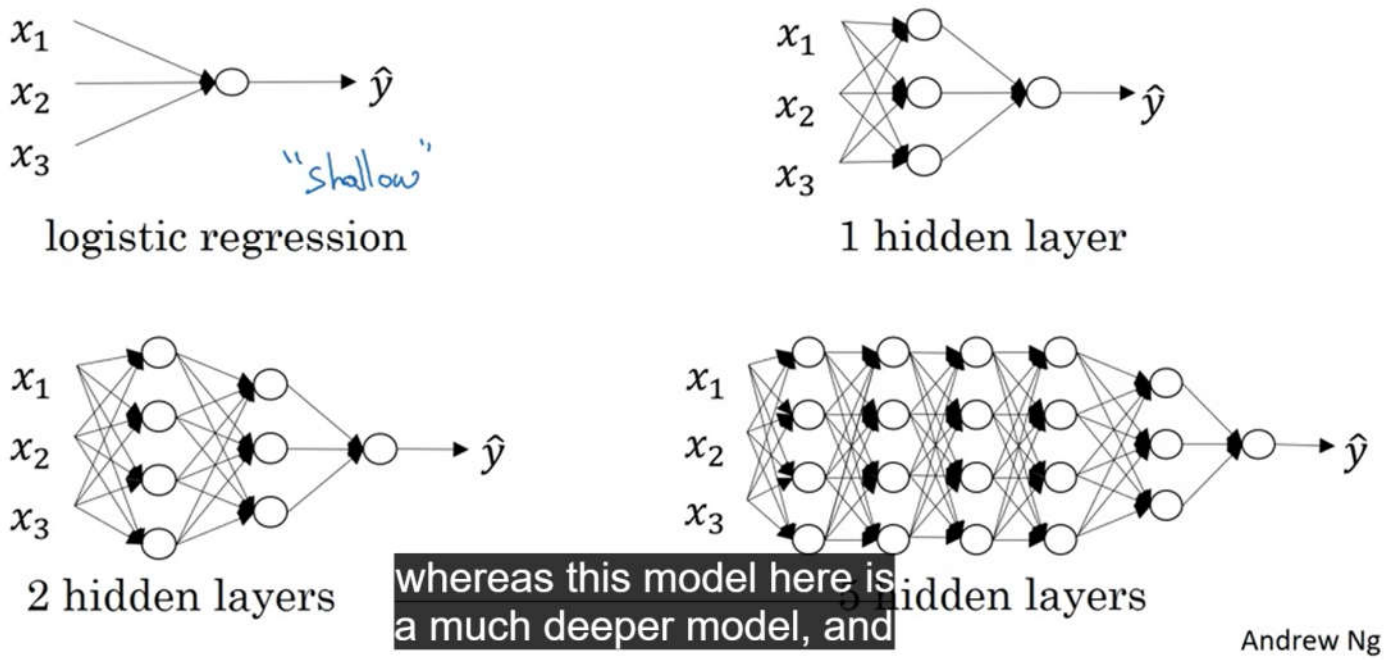
Andrew Ng

What is a deep neural network?



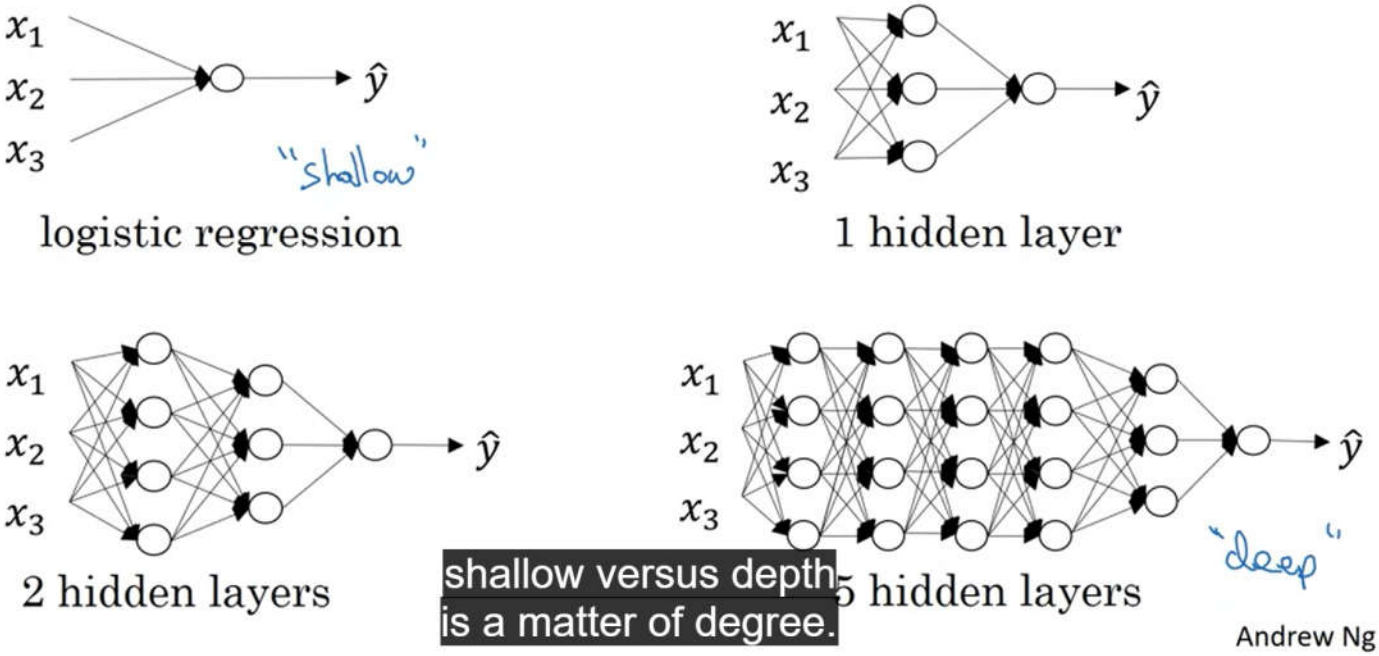
So here's an example of a neural network with two hidden layers and

What is a deep neural network?

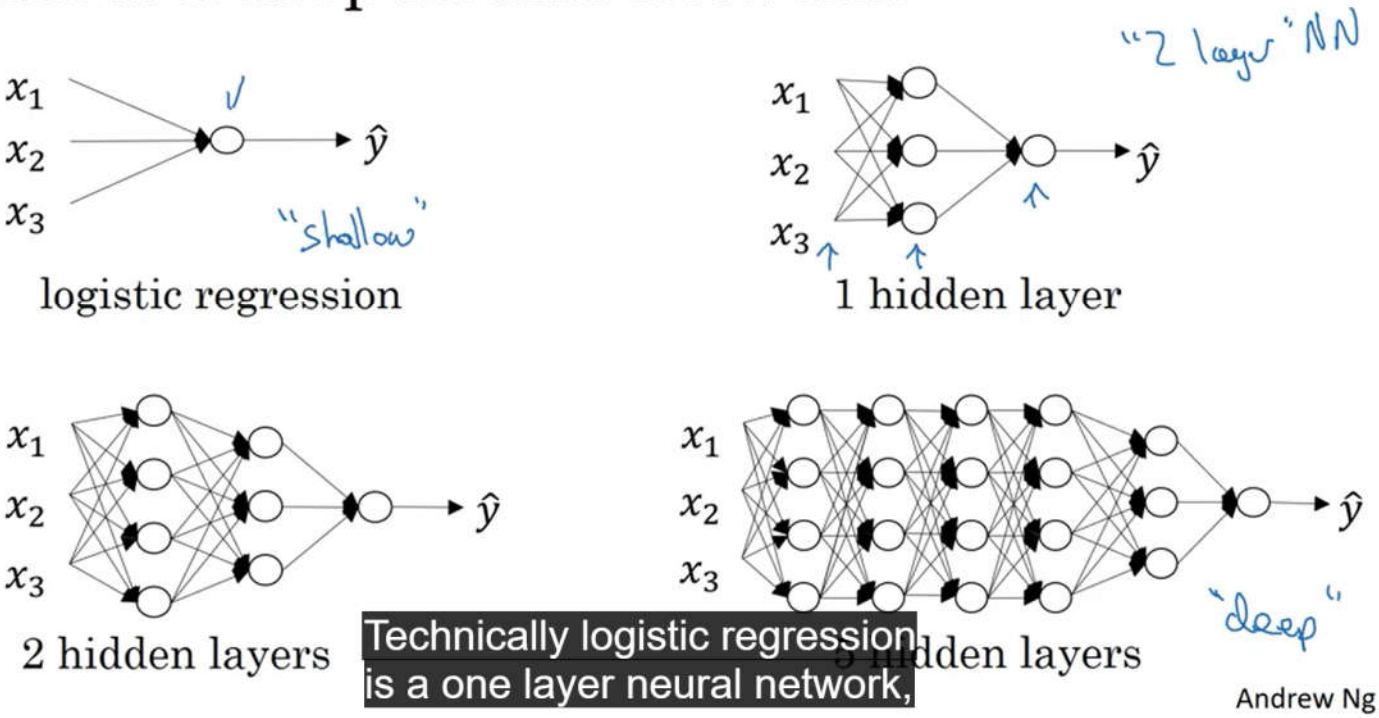


whereas this model here is a much deeper model, and

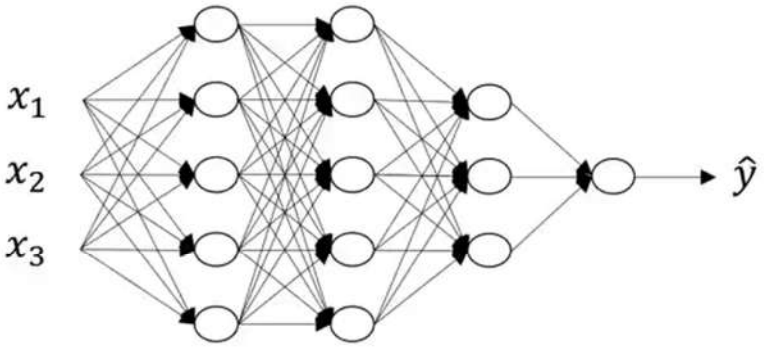
What is a deep neural network?



What is a deep neural network?



Deep neural network notation

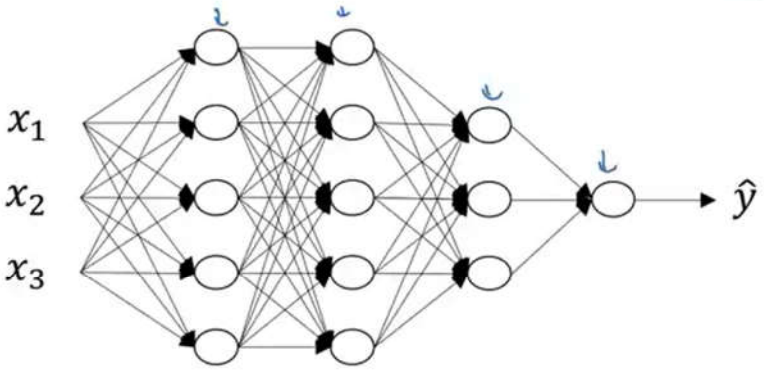


Let's now go through the notation we used to describe deep neural networks.

Andrew Ng

Deep neural network notation

4 layer NN

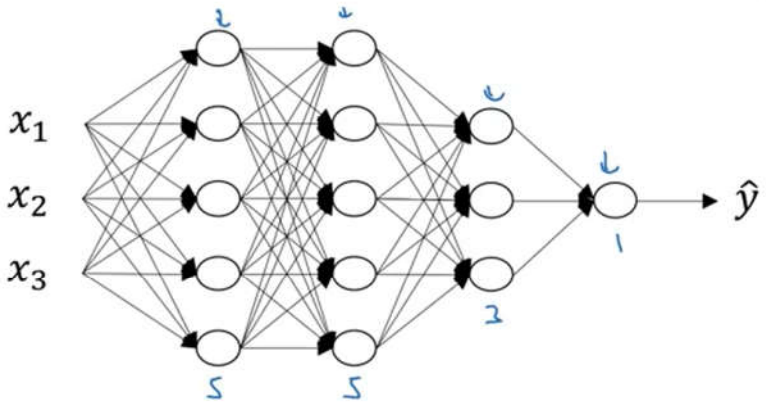


With three hidden layers, and the number of units in these hidden

Andrew Ng

Deep neural network notation

4 layer NN



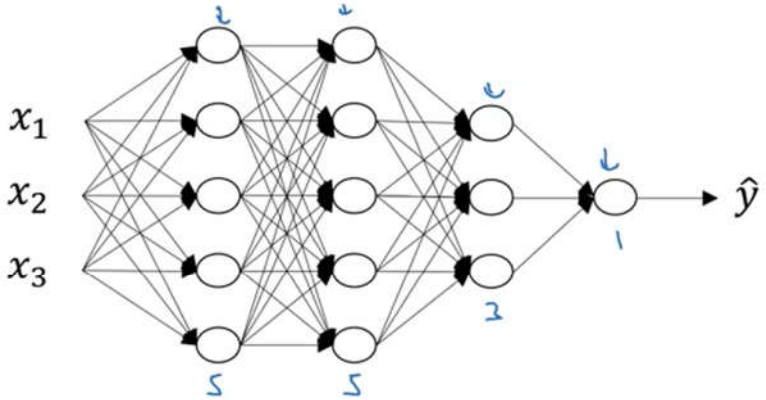
L

is going to use capital L ,to denote
the number of layers in the network.

Andrew Ng

Deep neural network notation

4 layer NN



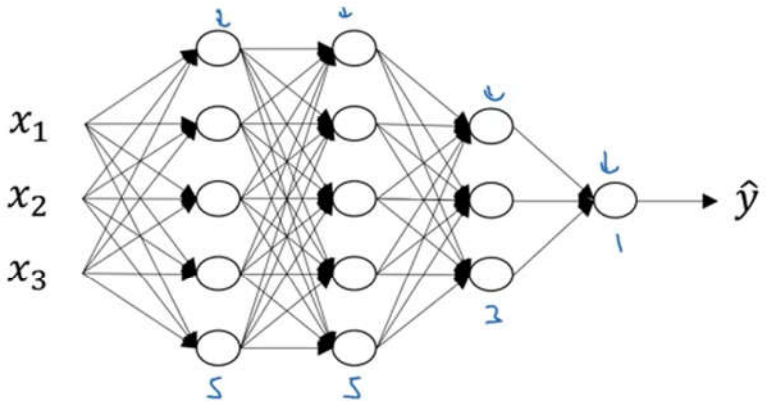
L = 4 (#layers)

So in this case, L = 4, and so
does the number of layers, and

Andrew Ng

Deep neural network notation

4 layer NN



$L = 4$ (#layers)

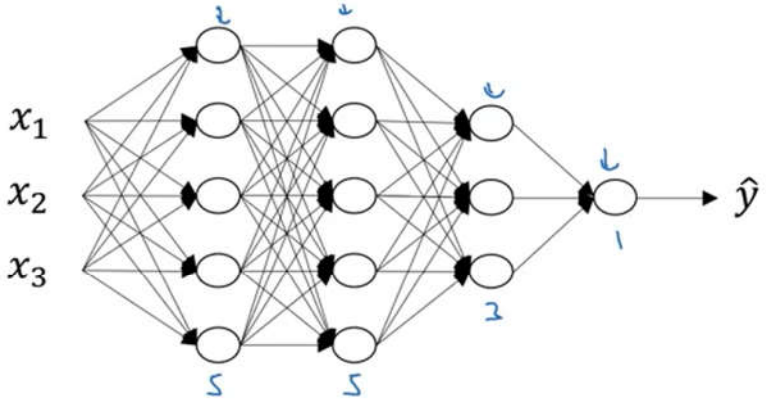
$n^{[l]} =$

we're going to use N superscript
[l] to denote the number of nodes,

Andrew Ng

Deep neural network notation

4 layer NN



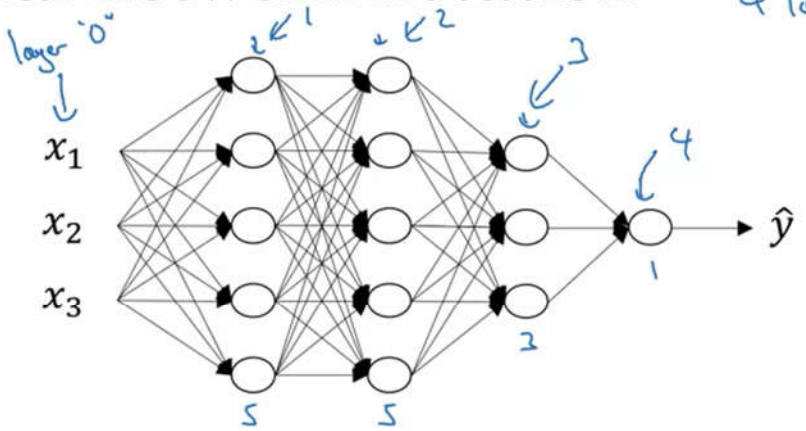
$L = 4$ (#layers)

$n^{[l]} =$ #units in layer l

or the number of units
in layer lowercase l.

Andrew Ng

Deep neural network notation



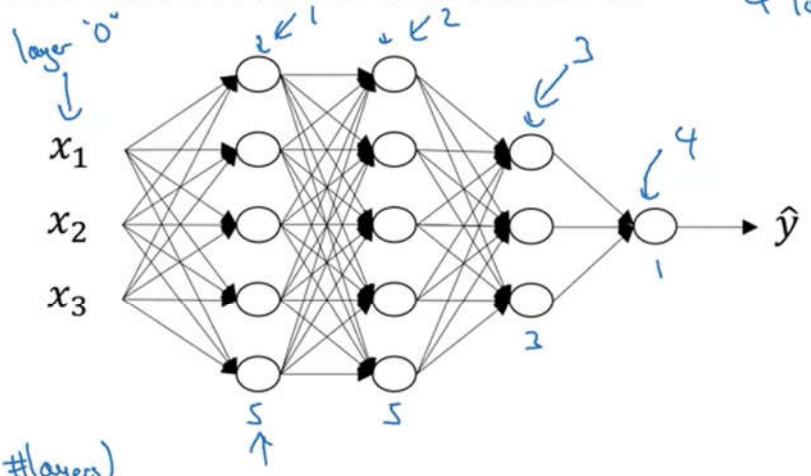
$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

This is layer 1, this is layer 2,
this is layer 3, and this is layer 4.

Andrew Ng

Deep neural network notation



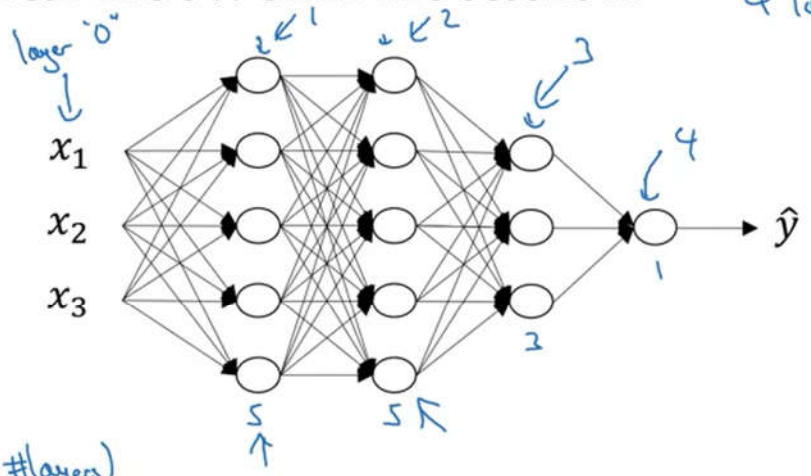
$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l $\quad n^{[1]} = 5$

the first is in there will equal 5,
because we have 5 hidden units there.

Andrew Ng

Deep neural network notation



$L = 4$ (#layers)

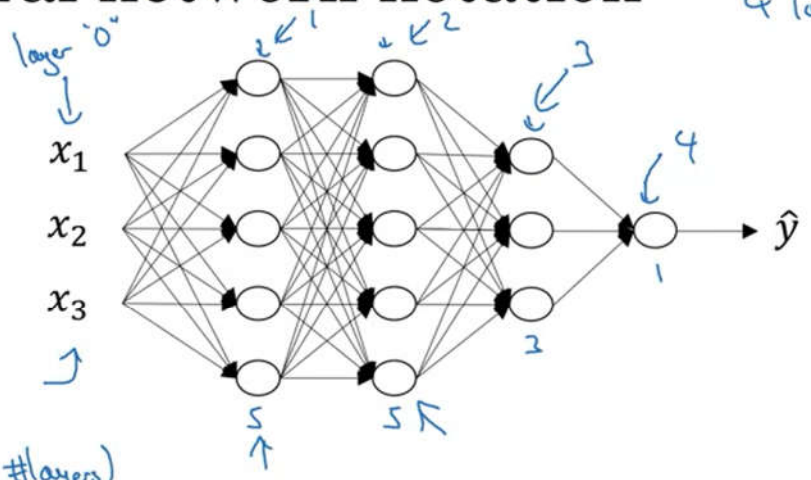
$n^{[l]}$ = #units in layer l

$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

n[4] = n[L] this number of upper units is 01,

Andrew Ng

Deep neural network notation



$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

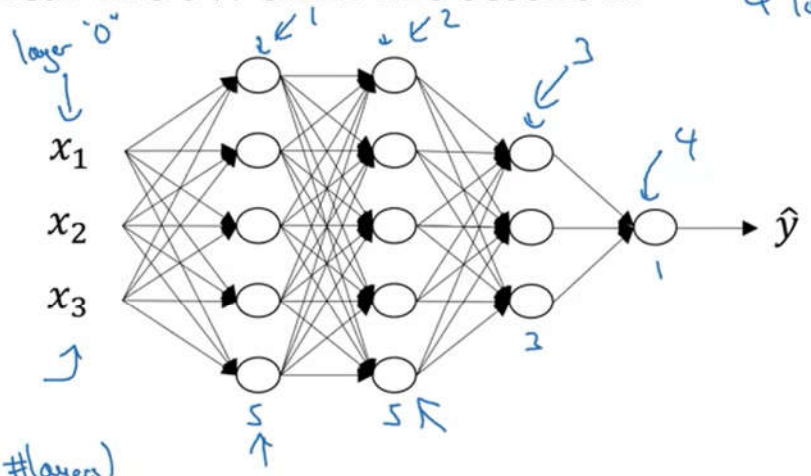
$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

$n^{[0]} = n_x = 3$

the input layer n[0] = nx = 3.

Andrew Ng

Deep neural network notation



$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

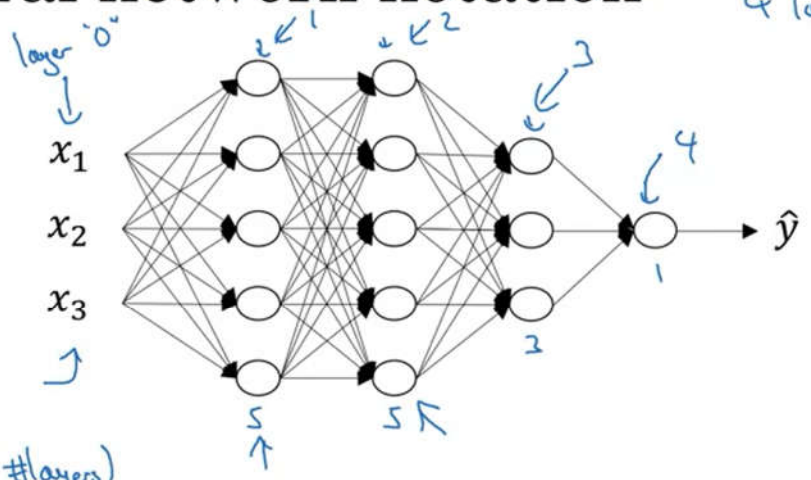
$n^{[0]} = n_x = 3$

$a^{[l]}$ = activations in layer l

a[l] to denote the activations in layer l.

Andrew Ng

Deep neural network notation



$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

$n^{[0]} = n_x = 3$

$a^{[l]}$ = activations in layer l

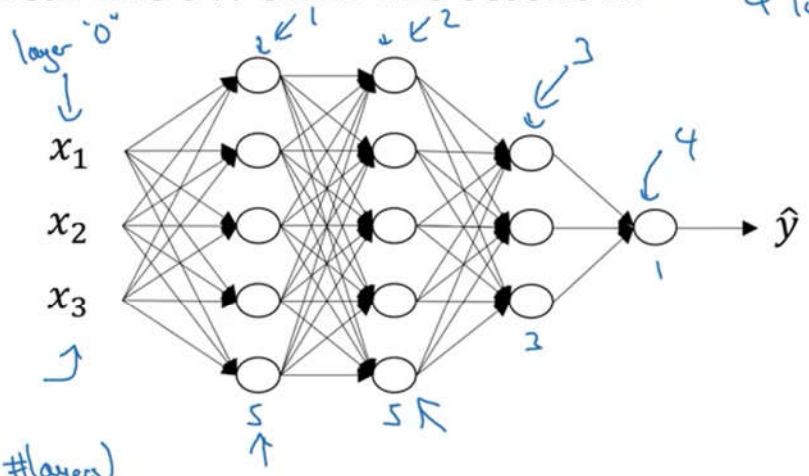
$a^{[l]} = g^{[l]}(z^{[l]})$

perhaps the activation is indexed by the layer l as well,

Andrew Ng

Deep neural network notation

4 layer NN



$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

$n^{[0]} = n_x = 3$

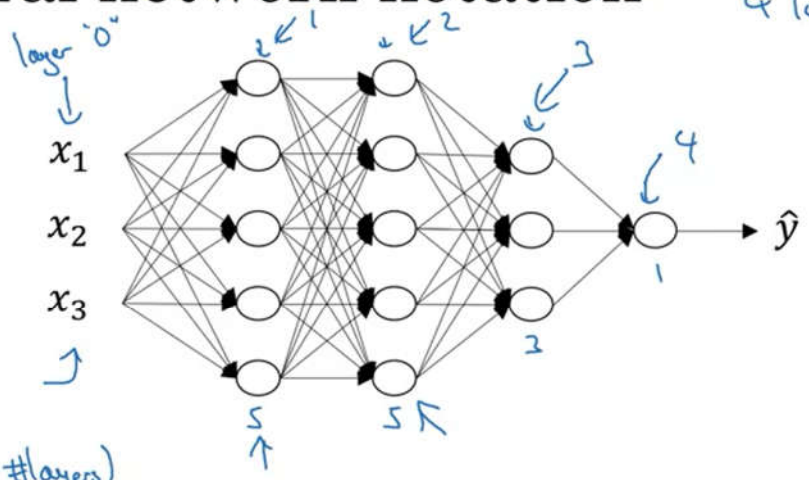
$a^{[l]}$ = activations in layer l

$a^{[l]} = g^{[l]}(z^{[l]})$, $w^{[l]}$ = weights for $z^{[l]}$

Andrew Ng

Deep neural network notation

4 layer NN



$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

$n^{[0]} = n_x = 3$

$a^{[l]}$ = activations in layer l

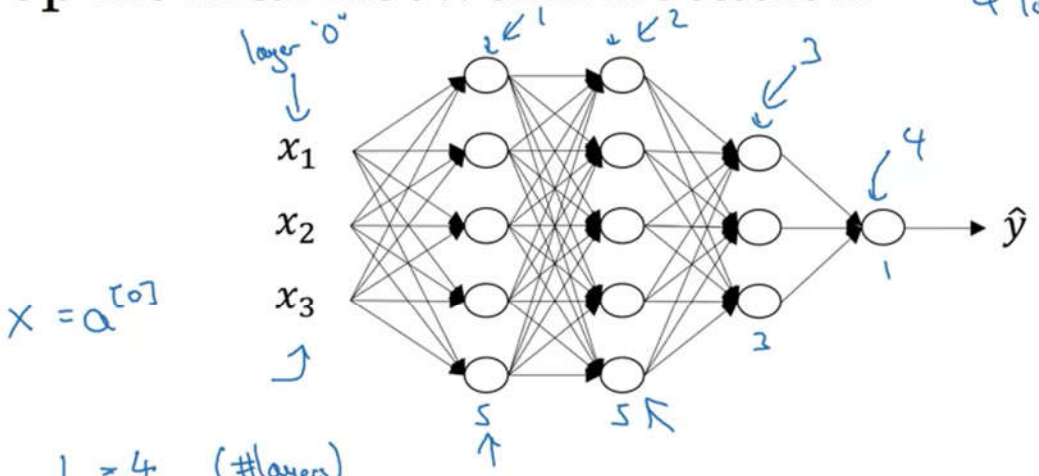
$a^{[l]} = g^{[l]}(z^{[l]})$, $w^{[l]}$ = weights for $z^{[l]}$

$b^{[l]}$

Andrew Ng

Deep neural network notation

4 layer NN



$X = a^{[0]}$

$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

$n^{[0]} = n_x = 3$

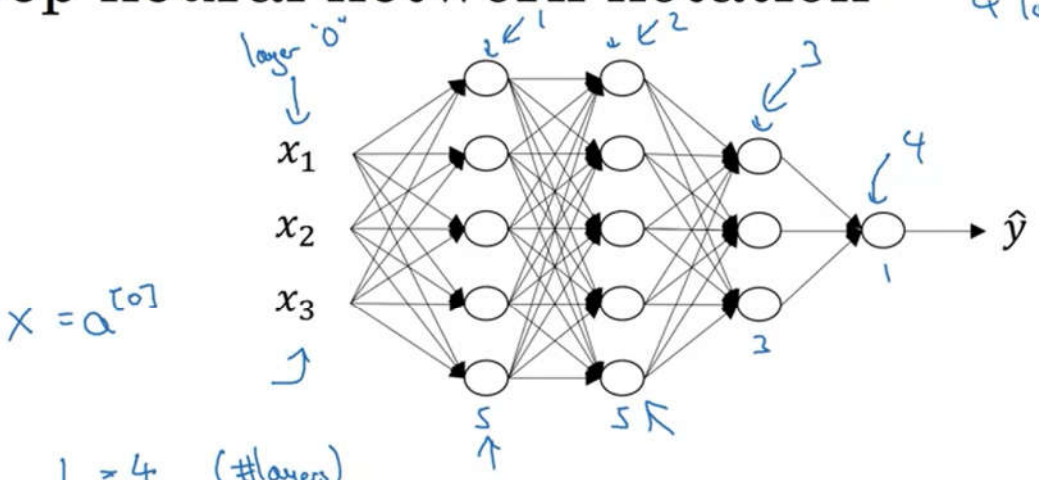
$a^{[l]}$ = activations in layer l

$a^{[l]} = g^{[l]}(z^{[l]})$, $w^{[l]}, b^{[l]}$ = weights for $z^{[l]}$

Andrew Ng

Deep neural network notation

4 layer NN



$X = a^{[0]}$

$L = 4$ (#layers)

$n^{[l]}$ = #units in layer l

$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = n^{[L]} = 1$

$n^{[0]} = n_x = 3$

$a^{[l]}$ = activations in layer l

$a^{[l]} = g^{[l]}(z^{[l]})$, $w^{[l]}, b^{[l]}$ = weights for $z^{[l]}$

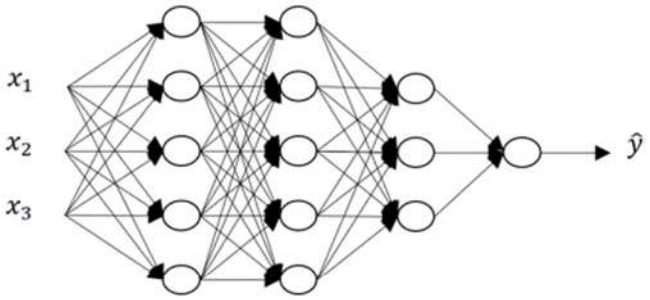
Andrew Ng

deeplearning.ai

Deep Neural Networks

Forward Propagation in a Deep Network

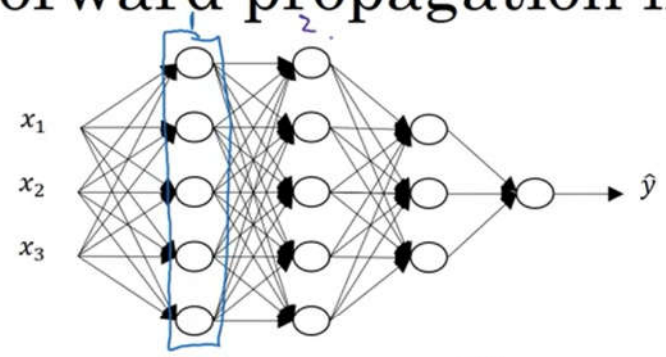
Forward propagation in a deep network



As usual, let's first go over

Andrew Ng

Forward propagation in a deep network

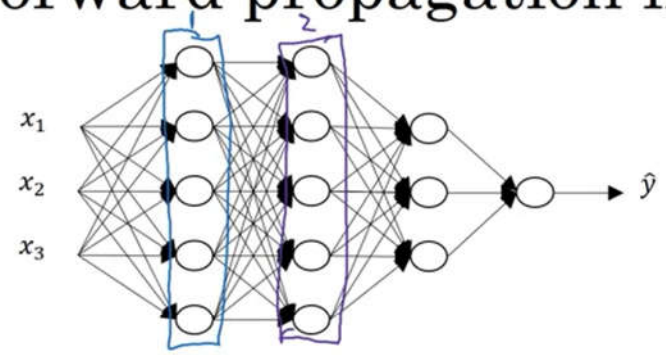


$$X: \; z^{[1]} = W^{[1]} x + b^{[1]}$$
$$a^{[1]} = g^{[1]}(z^{[1]})$$

How about layer two? Say that layer.

Andrew Ng

Forward propagation in a deep network

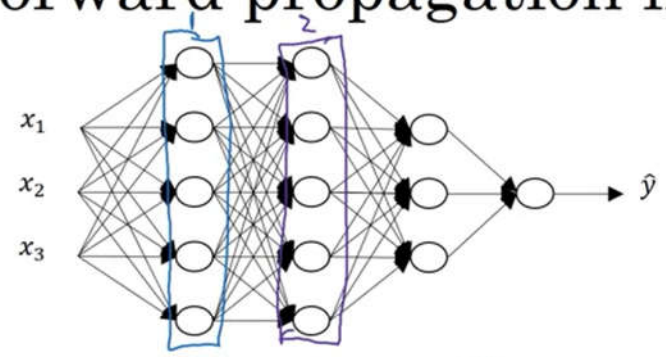


$$X: \; z^{[1]} = W^{[1]} x + b^{[1]}$$
$$a^{[1]} = g^{[1]}(z^{[1]})$$
$$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$$

Then, so the activation of layer two is the y matrix times the outputs of layer one.

Andrew Ng

Forward propagation in a deep network



$x: \; z^{[1]} = W^{[1]}x + b^{[1]}$

$a^{[1]} = g^{[1]}(z^{[1]})$

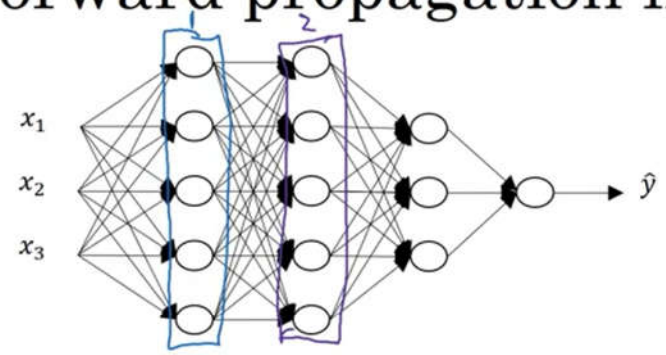
$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$

So, it's that value,

1:37 / 7:15

Andrew Ng

Forward propagation in a deep network



$x: \; z^{[1]} = W^{[1]}x + b^{[1]}$

$a^{[1]} = g^{[1]}(z^{[1]})$

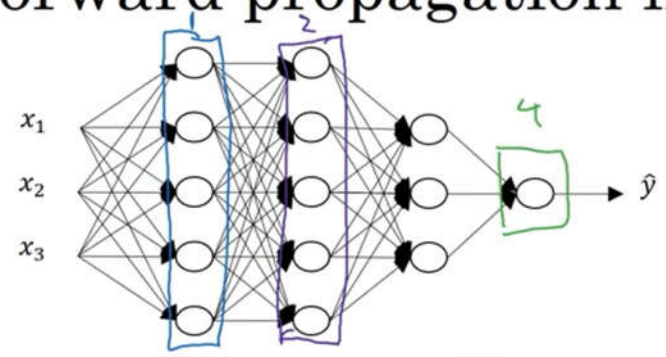
$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$

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

Then a2 equals the activation function applied to z2.

Andrew Ng

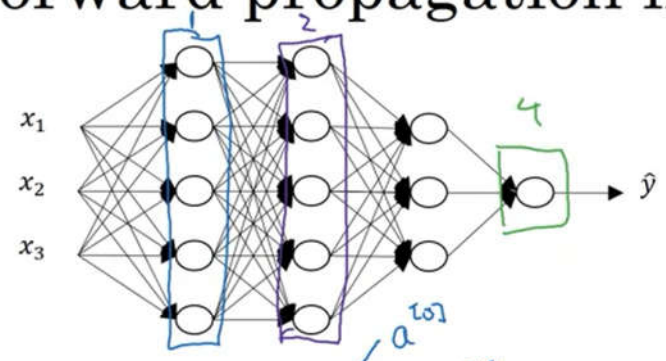
Forward propagation in a deep network



$$X: \; z^{[1]} = W^{[1]} x + b^{[1]}$$
$$a^{[1]} = g^{[1]}(z^{[1]})$$
$$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$$
$$a^{[2]} = g^{[2]}(z^{[2]})$$
$$\vdots$$
$$z^{[4]} = W^{[4]} a^{[3]} + b^{[4]}, \; a^{[4]} = g^{[4]}(z^{[4]})$$

Andrew Ng

Forward propagation in a deep network

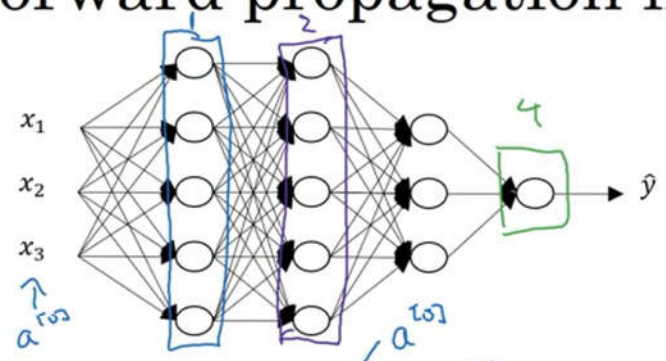


$$X: \; z^{[1]} = W^{[1]} \overset{a^{[0]}}{x} + b^{[1]}$$
$$a^{[1]} = g^{[1]}(z^{[1]})$$
$$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$$
$$a^{[2]} = g^{[2]}(z^{[2]})$$
$$\vdots$$
$$z^{[4]} = W^{[4]} a$$

because the input feature vector x is also the activations of layer zero.

Andrew Ng

Forward propagation in a deep network



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

$a^{[0]}$

X : $z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = g^{[1]}(z^{[1]})$

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

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

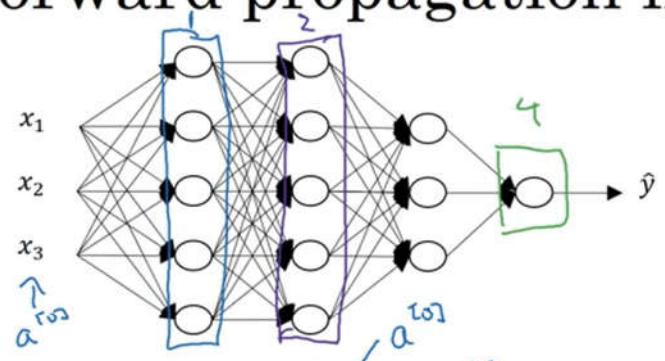
$\vdots$

$z^{[4]} = W^{[4]} a^{[3]} + b^{[4]}$

So, that's the general forward propagation equation.

Andrew Ng

Forward propagation in a deep network



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

$X = A^{[0]}$

Vectorized:

$Z^{[1]} = W^{[1]} A^{[0]} + b^{[1]}$

$A^{[1]} = g^{[1]}(Z^{[1]})$

$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$

$A^{[2]} = g^{[2]}(Z^{[2]})$

$a^{[0]}$

X : $z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = g^{[1]}(z^{[1]})$

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

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

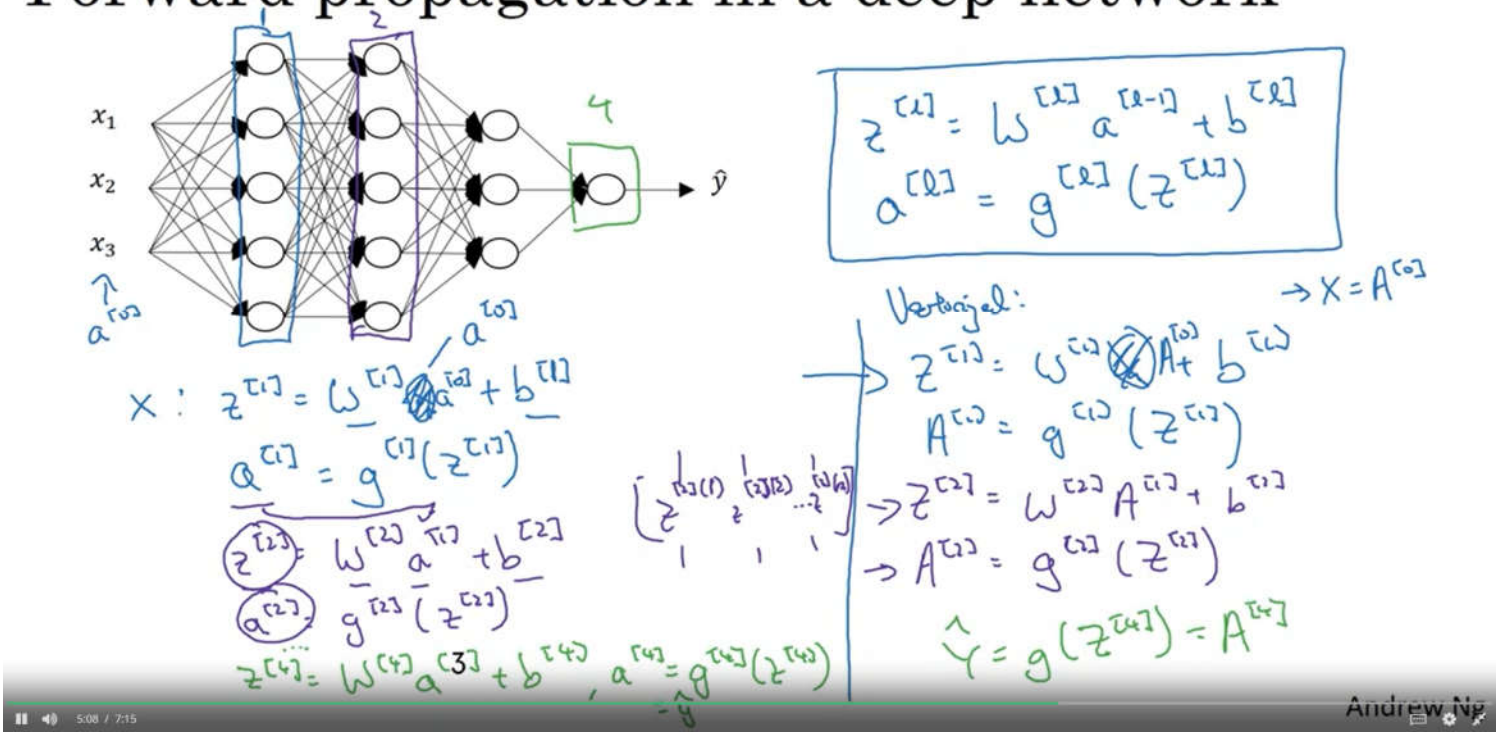
$\vdots$

$z^{[4]} = W^{[4]} a^{[3]} + b^{[4]}$

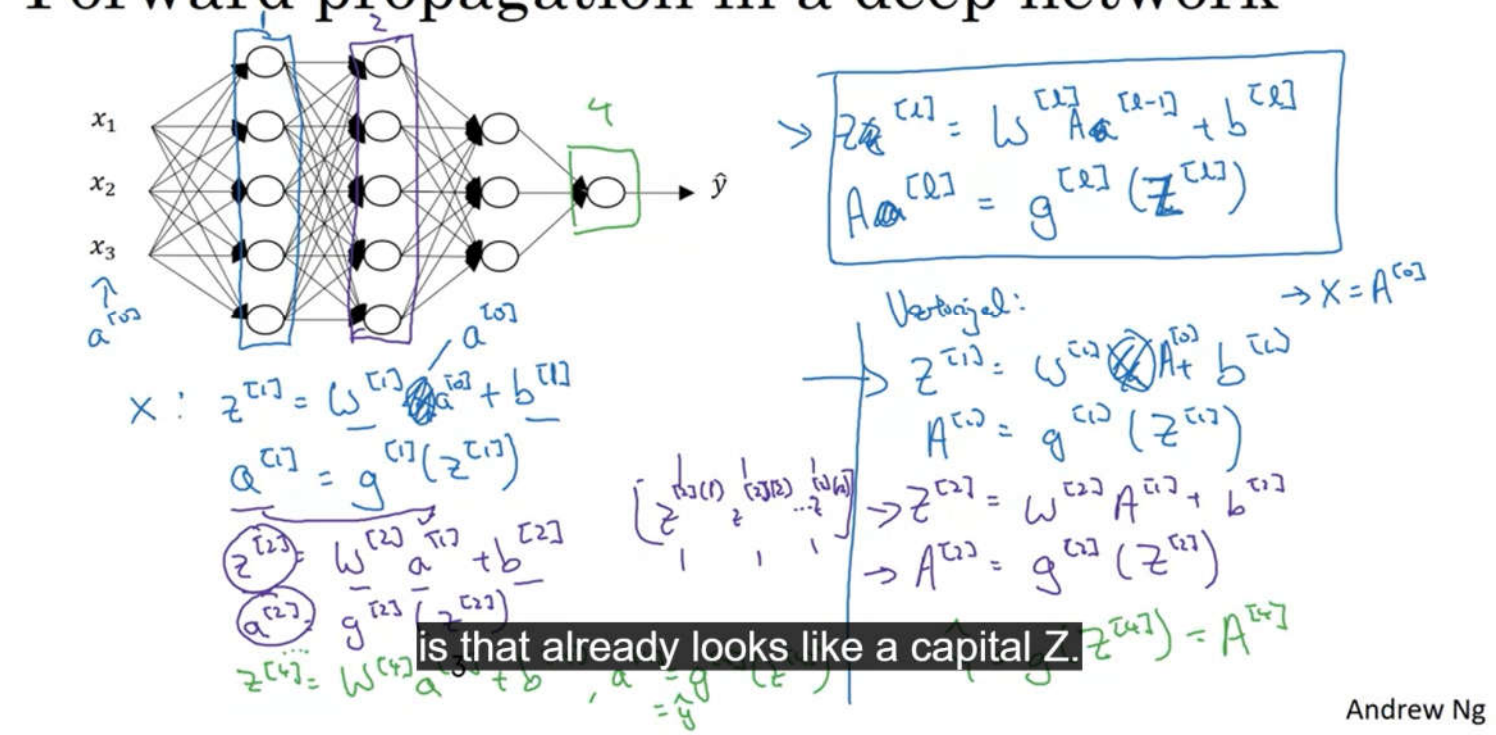
We're just taking these vectors z or a and so on,

Andrew Ng

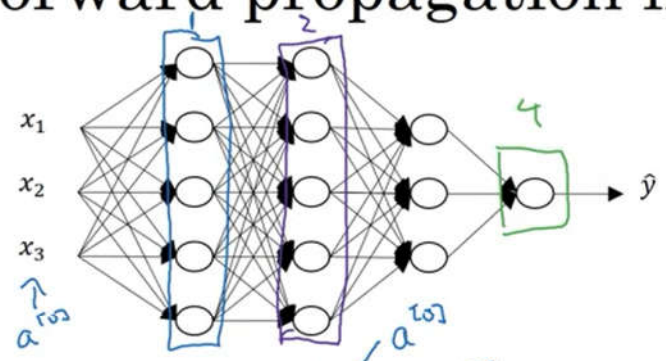
Forward propagation in a deep network



Forward propagation in a deep network



Forward propagation in a deep network



$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$
$$A^{[l]} = g^{[l]}(Z^{[l]})$$

$A^{[0]} = X$

Vertical:

$$Z^{[1]} = W^{[1]} A^{[0]} + b^{[1]}$$
$$A^{[1]} = g^{[1]}(Z^{[1]})$$
$$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$$
$$A^{[2]} = g^{[2]}(Z^{[2]})$$

for $l = 1 \ldots 4$

$X = A^{[0]}$

$X: \; Z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = g^{[1]}(z^{[1]})$

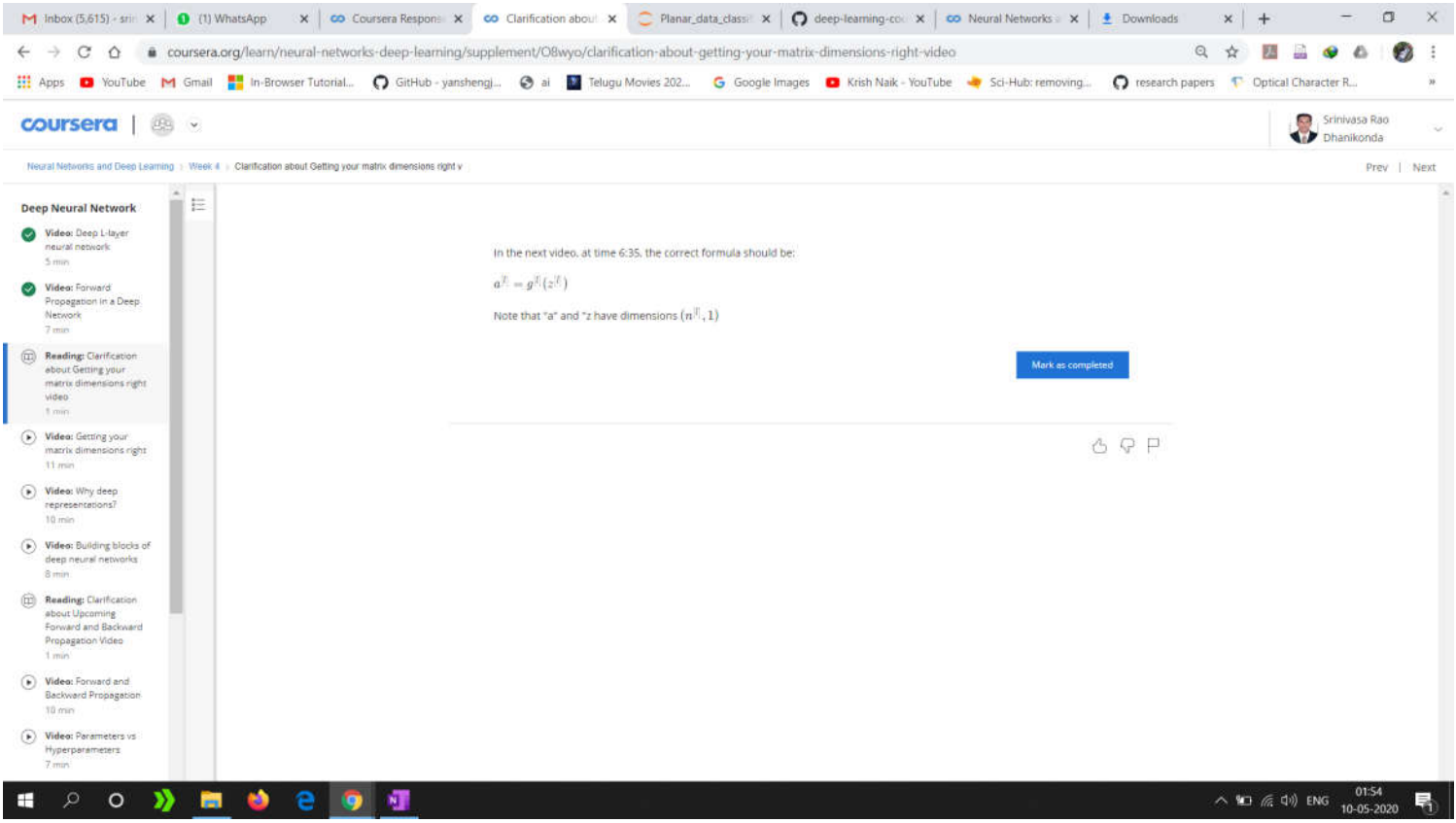
$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$

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

$z^{[4]} = \ldots = A^{[4]}$

For L equals 1 through capital L. Then you have to compute the activations for layer one,

Andrew Ng



deeplearning.ai

Deep Neural Networks

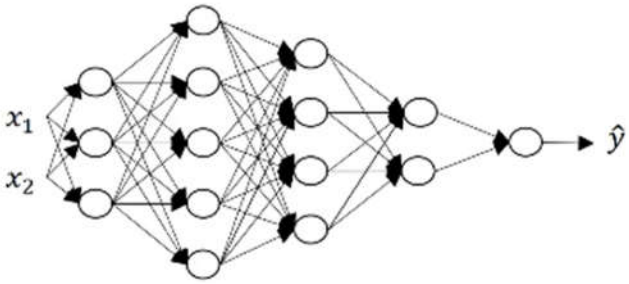
Getting your matrix dimensions right

So let me show you how to do that,
since I hope this will make it easier for

0:12 / 11:09

Parameters $W^{[l]}$ and $b^{[l]}$

$L=5$

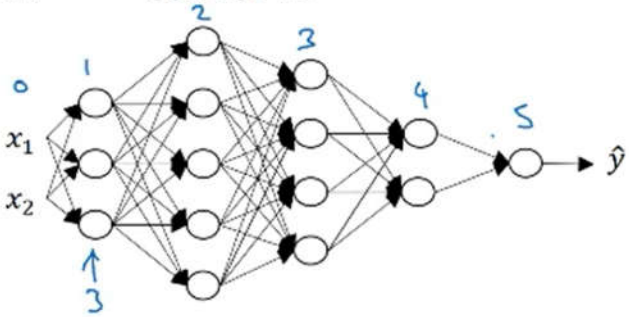


Capital L is equal to 5, right, counting
quickly, not counting the input layer,

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$



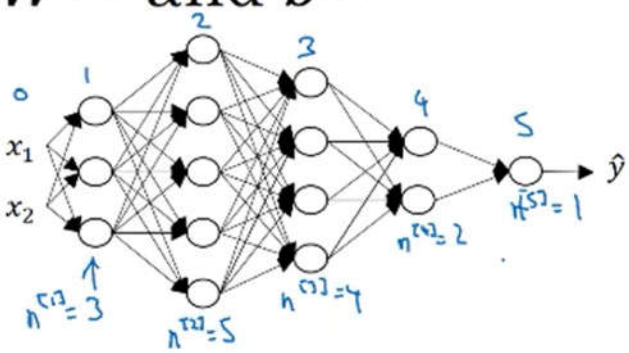
$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$

layer 1, layer 2, layer 3,
layer 4, and layer 5.

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$

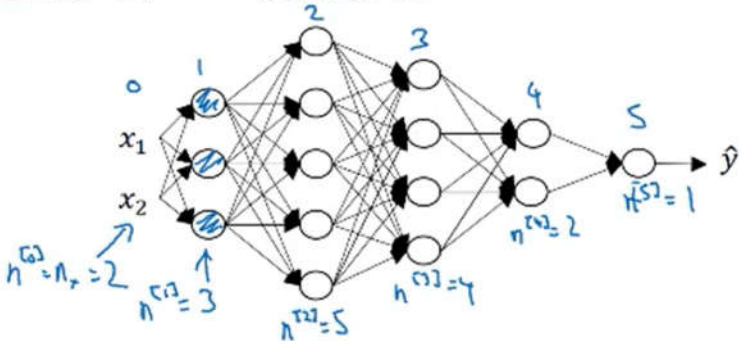


$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$

n3 is equal to 4, n4 is equal to 2,
and n5 is equal to 1.

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$



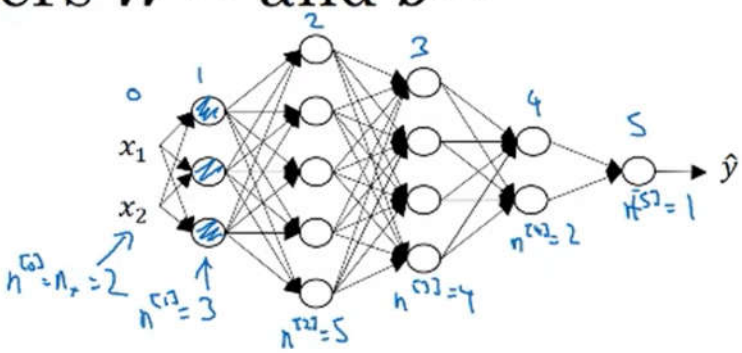
$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(3,1)$

it's going to be a 3-dimensional vector.

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(3,1)$

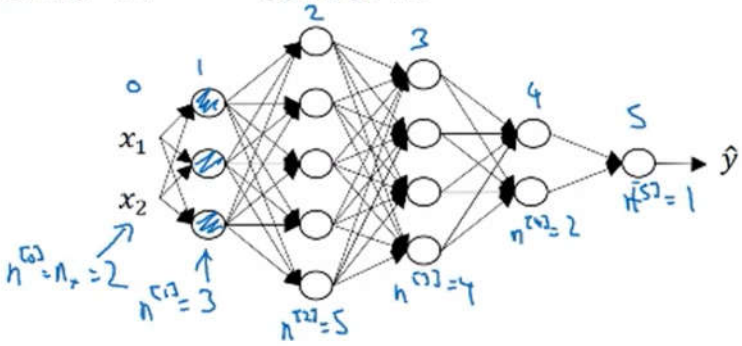
$(n^{[1]},1)$

n1 by 1-dimensional matrix,
all right, so 3 by 1 in this case.

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$



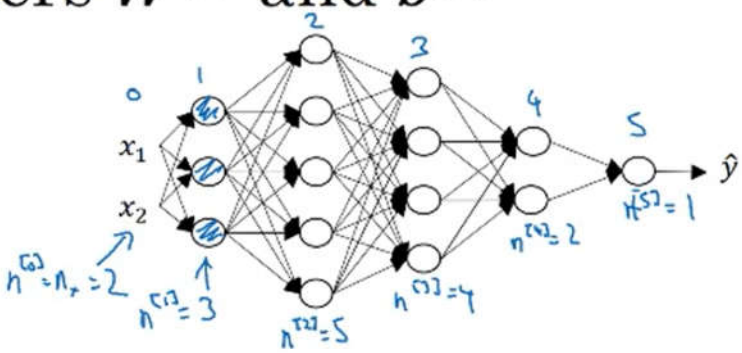
$$\underset{\substack{(3,1)\\(n^{[1]},1)}}{z^{[1]}} = W^{[1]} \cdot \underset{\substack{(2,1)\\(n^{[0]},1)}}{x} + b^{[1]}$$

So x is in this example 2 by 1, but more generally, it would be n0 by 1.

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$



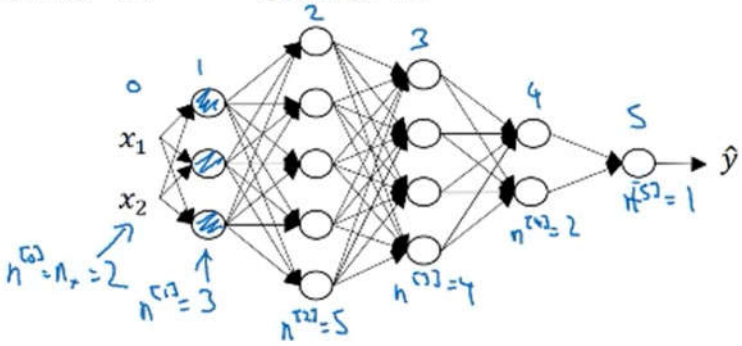
$$\underset{\substack{(3,1)\\(n^{[1]},1)}}{z^{[1]}} = \underset{(3,2)}{W^{[1]}} \cdot \underset{\substack{(2,1)\\(n^{[0]},1)}}{x} + b^{[1]}$$

$$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$$

Right, because a 3 by 2 matrix times a 2 by 1 matrix, or

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$$(3,1) \leftarrow (3,2) \quad (2,1)$$

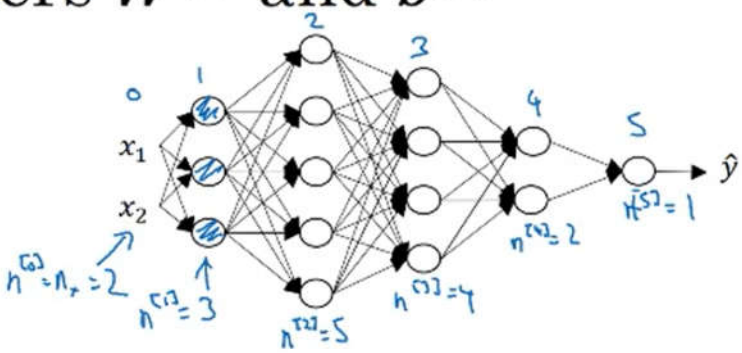
$$(n^{[1]},1) \quad (n^{[0]},1)$$

$$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$$

times the 2 by 1 vector,
that gives you a 3 by 1 vector.

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$$(3,1) \leftarrow (3,2) \quad (2,1)$$

$$(n^{[1]},1) \quad (n^{[1]},n^{[0]}) \quad (n^{[0]},1)$$

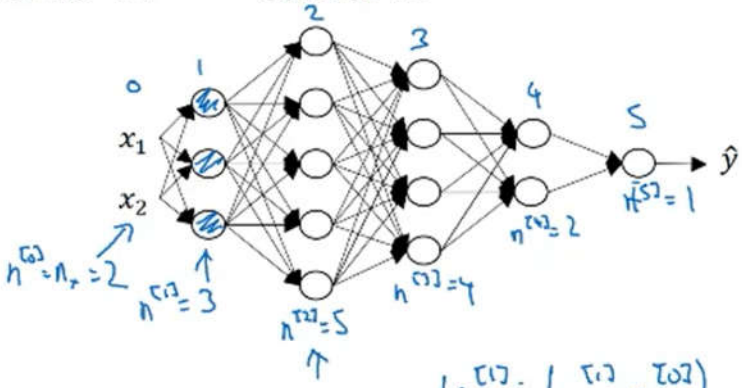
$$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$$

And more generally, this is going to
be an n1 by n0 dimensional matrix.

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(3,1) \leftarrow (3,2) \quad (2,1)$

$(n^{[1]},1) \quad (n^{[1]},n^{[0]}) \quad (n^{[2]},1)$

$$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$$

$W^{[1]}: (n^{[1]}, n^{[0]})$

$W^{[2]}: (5,3) \quad (n^{[2]}, n^{[1]})$

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

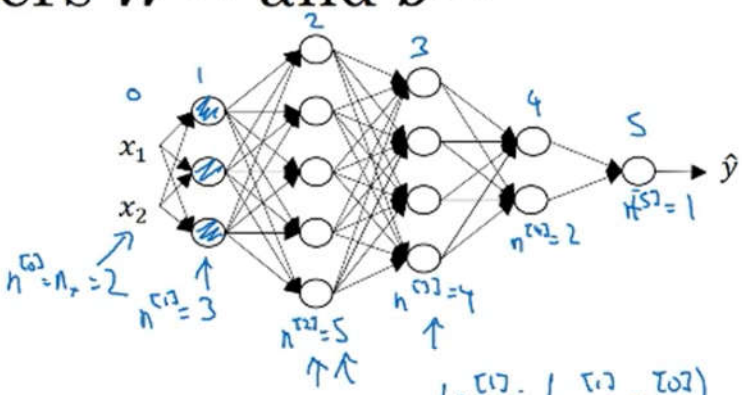
$(5,1) \quad (5,3) \quad (3,1)$

this had better be 5 by 3.

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(3,1) \leftarrow (3,2) \quad (2,1)$

$(n^{[1]},1) \quad (n^{[1]},n^{[0]}) \quad (n^{[2]},1)$

$$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$$

$W^{[1]}: (n^{[1]}, n^{[0]})$

$W^{[2]}: (5,3) \quad (n^{[2]}, n^{[1]})$

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

$(5,1) \quad (5,3) \quad (3,1)$

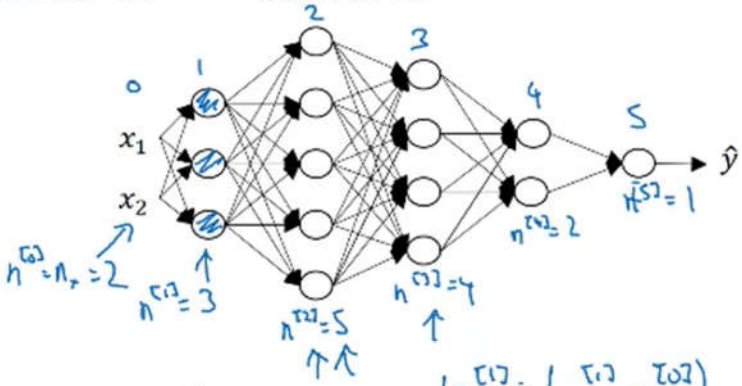
$W^{[3]}: (4,5)$

$W^{[4]}: (2,4) \quad , \quad W^{[5]}: (1,2)$

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$



$W^{[l]} : (n^{[l]}, n^{[l-1]})$

$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$

$(3,1) \leftarrow (3,2) \quad (2,1)$

$(n^{[1]}, 1) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, 1)$

$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$

$W^{[1]} : (n^{[1]}, n^{[0]})$

$W^{[2]} : (5, 3) \quad (n^{[2]}, n^{[1]})$

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

$(5,1) \quad (5,3) \quad (3,1)$

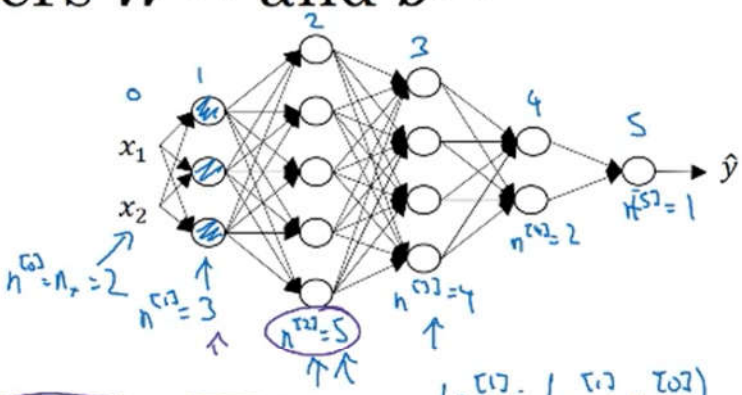
$W^{[3]} : (4, 5)$

$W^{[4]} : (2, 4) \quad , \quad W^{[5]} : (1, 2)$

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$



$W^{[l]} : (n^{[l]}, n^{[l-1]})$

$b^{[l]} : (n^{[l]}, 1)$

$z^{[1]} = \boxed{W^{[1]} \cdot x} + \boxed{b^{[1]}}$

$(3,1) \leftarrow (3,2) \quad (2,1) \quad (3,1)$

$(n^{[1]}, 1) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, 1) \quad (n^{[1]}, 1)$

$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$

$W^{[1]} : (n^{[1]}, n^{[0]})$

$W^{[2]} : (5, 3) \quad (n^{[2]}, n^{[1]})$

$z^{[2]} = \boxed{W^{[2]} \cdot a^{[1]}} + \boxed{b^{[2]}}$

$\rightarrow (5,1) \quad (5,3) \quad (3,1) \quad (5,1) \quad (n^{[2]}, 1)$

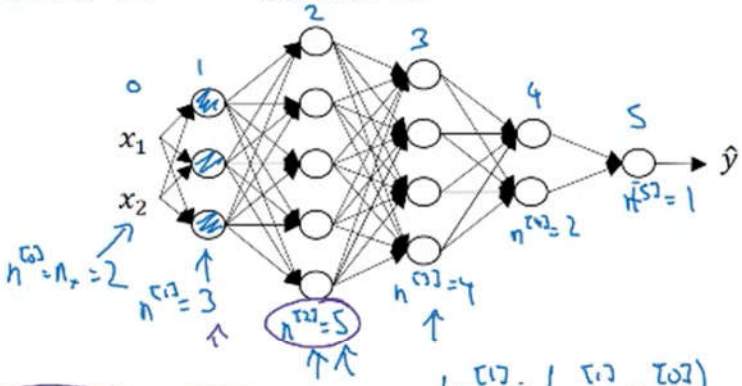
$W^{[3]} : (4, 5)$

$W^{[4]} : (2, 4) \quad , \quad W^{[5]} : (1, 2)$

Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$



$\rightarrow W^{[l]} : (n^{[l]}, n^{[l-1]})$

$\rightarrow b^{[l]} : (n^{[l]}, 1)$

$dW^{[l]} : (n^{[l]}, n^{[l-1]})$

$db^{[l]} : (n^{[l]}, 1)$

$W^{[1]} : (n^{[1]}, n^{[0]})$

$W^{[2]} : (5, 3) \quad (n^{[2]}, n^{[1]})$

$z^{[1]} = \boxed{W^{[1]} \cdot x} + \boxed{b^{[1]}}$

$(3,1) \leftarrow (3,2) \quad (2,1) \qquad (3,1)$

$(n^{[1]}, 1) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, 1) \qquad (n^{[1]}, 1)$

$z^{[2]} = \boxed{W^{[2]} \cdot a^{[1]}} + \boxed{b^{[2]}}$

$\rightarrow (5,1) \quad (5,3) \quad (3,1) \qquad (5,1)$

$(n^{[2]}, 1)$

$W^{[3]} : (4, 5)$

$W^{[4]} : (2, 4) \quad , \quad W^{[5]} : (1, 2)$

$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$

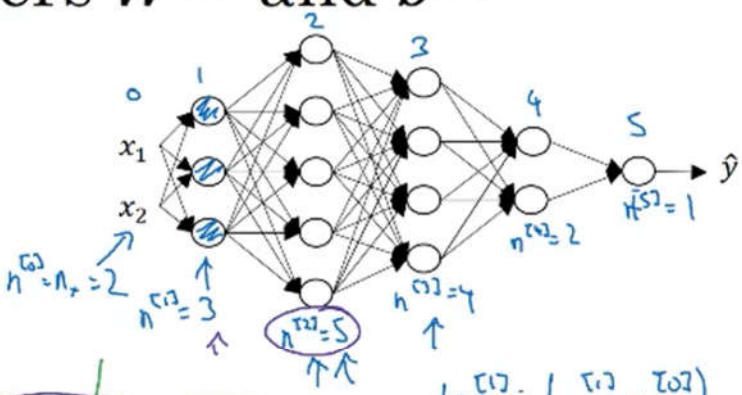
Andrew Ng

Parameters $W^{[l]}$ and $b^{[l]}$

$L = 5$

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

$a^{[l]}$



$\rightarrow W^{[l]} : (n^{[l]}, n^{[l-1]})$

$\rightarrow b^{[l]} : (n^{[l]}, 1)$

$dW^{[l]} : (n^{[l]}, n^{[l-1]})$

$db^{[l]} : (n^{[l]}, 1)$

$W^{[1]} : (n^{[1]}, n^{[0]})$

$W^{[2]} : (5, 3) \quad (n^{[2]}, n^{[1]})$

$z^{[1]} = \boxed{W^{[1]} \cdot x} + \boxed{b^{[1]}}$

$(3,1) \leftarrow (3,2) \quad (2,1) \qquad (3,1)$

$(n^{[1]}, 1) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, 1) \qquad (n^{[1]}, 1)$

$z^{[2]} = \boxed{W^{[2]} \cdot a^{[1]}} + \boxed{b^{[2]}}$

$\rightarrow (5,1) \quad (5,3) \quad (3,1) \qquad (5,1)$

$(n^{[2]}, 1)$

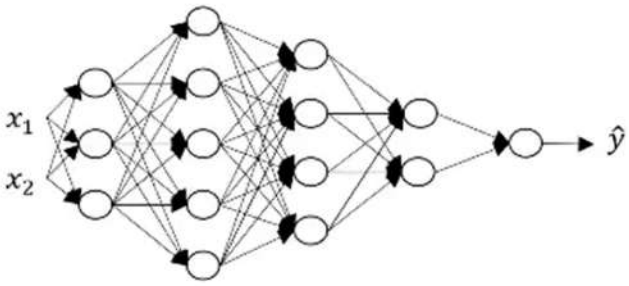
$W^{[3]} : (4, 5)$

$W^{[4]} : (2, 4) \quad , \quad W^{[5]} : (1, 2)$

$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$

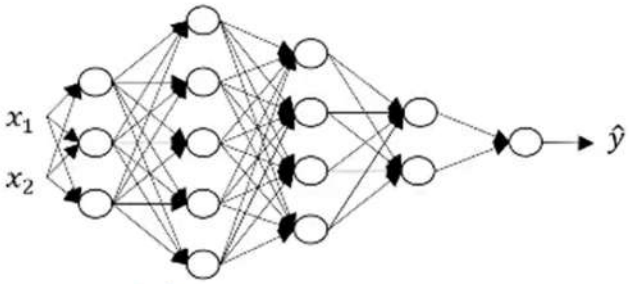
Andrew Ng

Vectorized implementation



Andrew Ng

Vectorized implementation

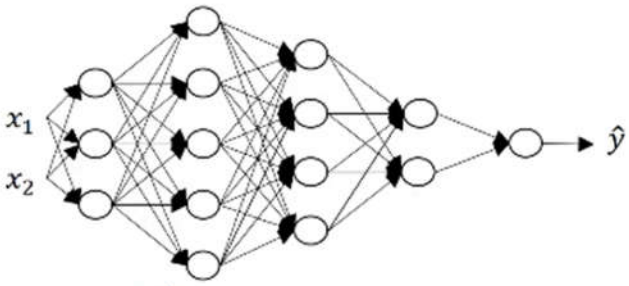


$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(n^{[1]}, 1)$ $(n^{[1]}, n^{[0]})$ $(n^{[0]}, 1)$ $(n^{[1]}, 1)$

Andrew Ng

Vectorized implementation



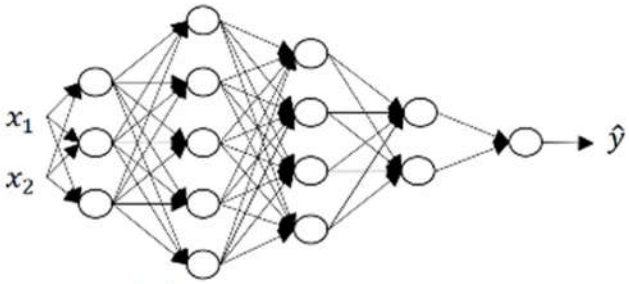
$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(n^{[1]},1)$ $(n^{[1]},n^{[0]})$ $(n^{[0]},1)$ $(n^{[1]},1)$

$$Z^{[1]} = W^{[1]} \cdot X + b^{[1]}$$

Andrew Ng

Vectorized implementation



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(n^{[1]},1)$ $(n^{[1]},n^{[0]})$ $(n^{[0]},1)$ $(n^{[1]},1)$

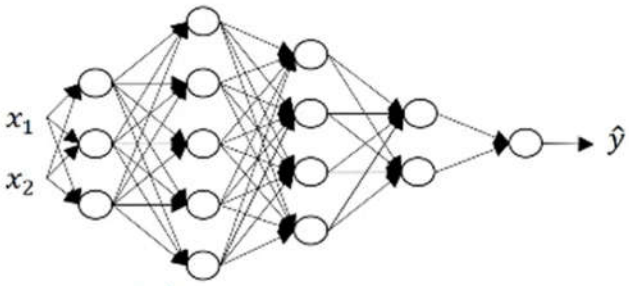
$$\left[z^{[1](1)} \; z^{[1](2)} \; \cdots \; z^{[1](m)} \right]$$

$$Z^{[1]} = W^{[1]} \cdot X + b^{[1]}$$

$(n^{[1]},m)$ $(n^{[1]},n^{[0]})$ $(n^{[0]},m)$

Andrew Ng

Vectorized implementation



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(n^{[1]}, 1) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, 1) \quad (n^{[1]}, 1)$

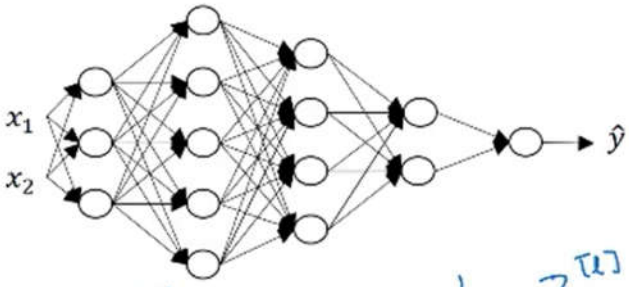
$$\left[z^{[1](1)} \; z^{[1](2)} \; \cdots \; z^{[1](m)} \right]$$

$$Z^{[1]} = W^{[1]} \cdot X + b^{[1]}$$

$(n^{[1]}, m) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, m) \quad (n^{[1]}, 1)$

Andrew Ng

Vectorized implementation



$$z^{[1]} = W^{[1]} \cdot x + b^{[1]}$$

$(n^{[1]}, 1) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, 1) \quad (n^{[1]}, 1)$

$$\left[z^{[1](1)} \; z^{[1](2)} \; \cdots \; z^{[1](m)} \right]$$

$$Z^{[1]} = W^{[1]} \cdot X + b^{[1]}$$

$(n^{[1]}, m) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, m) \quad (n^{[1]}, 1)$

$(n^{[1]}, m)$

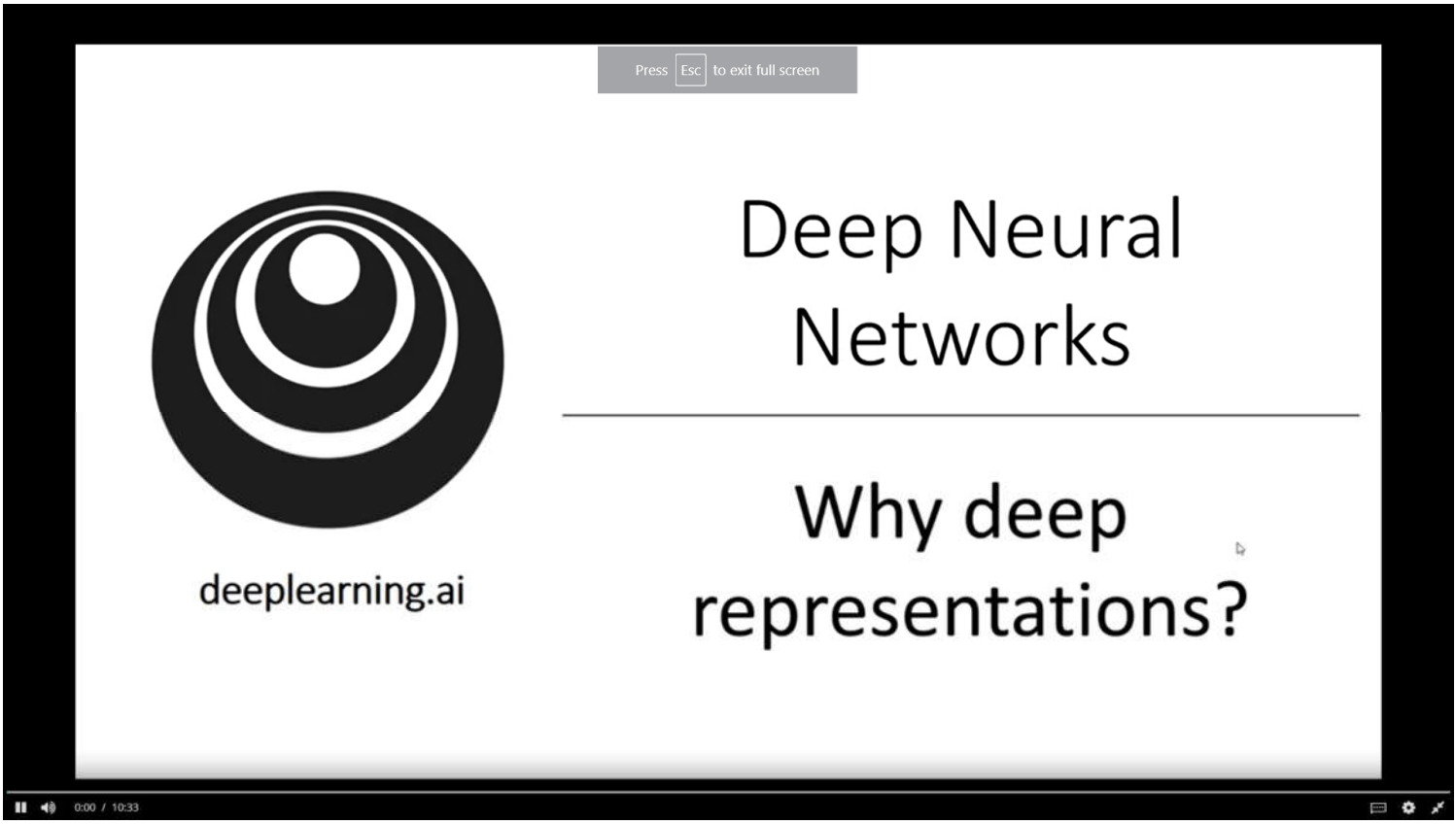
$z^{[l]}, a^{[l]} : (n^{[l]}, 1)$

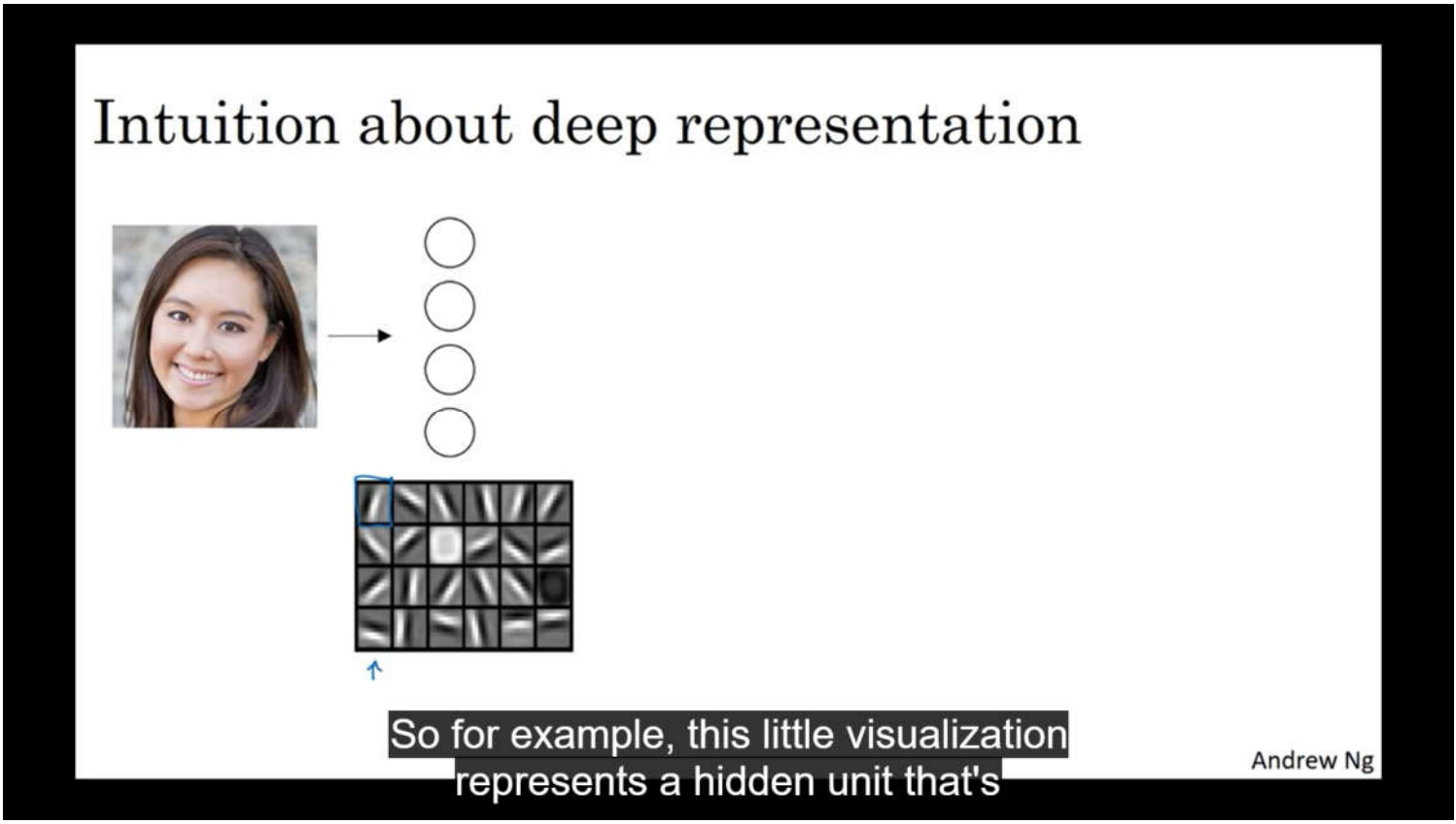
$Z^{[l]}, A^{[l]} : (n^{[l]}, m)$

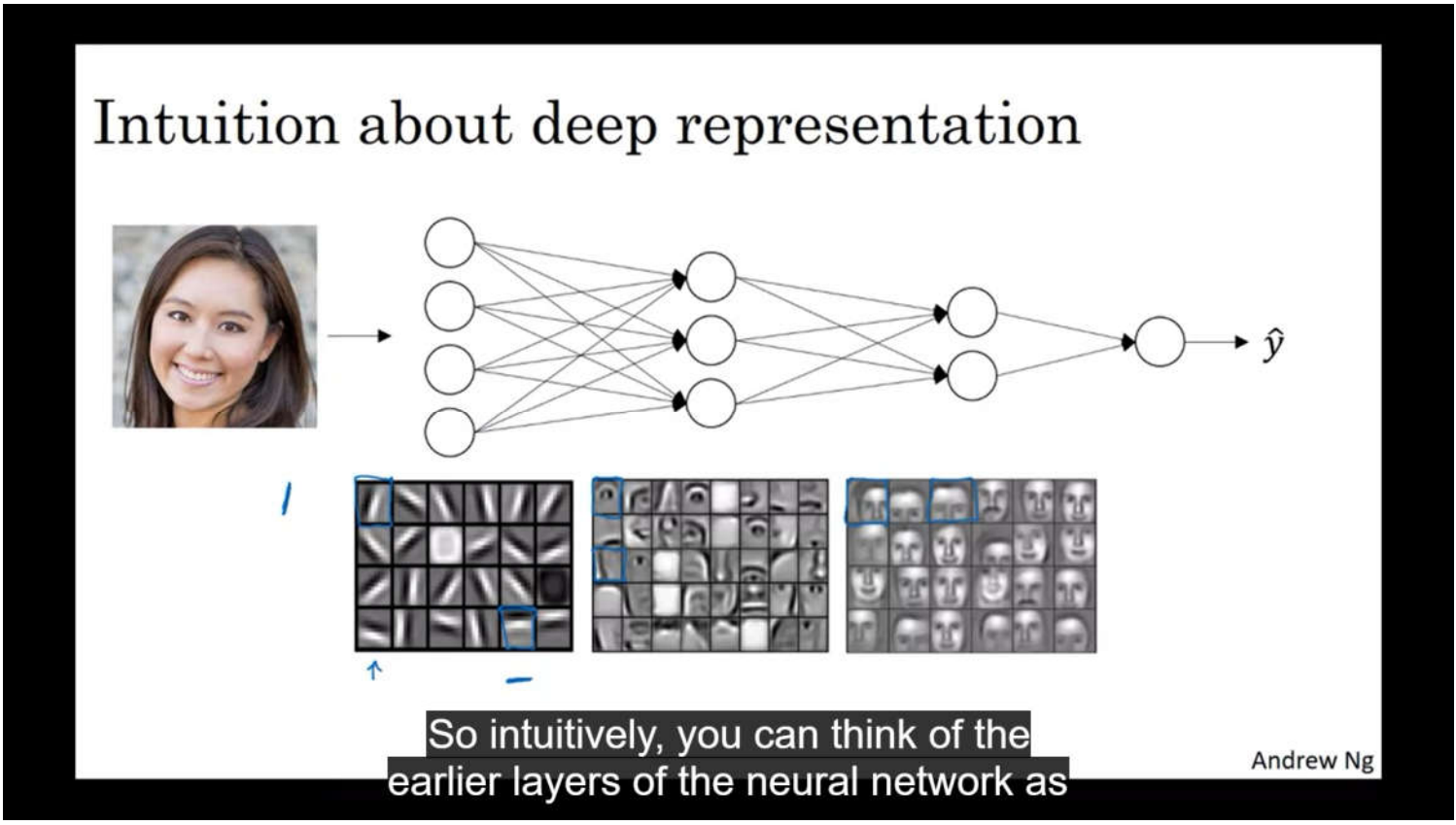
$l = 0 \quad A^{[0]} = X = (n^{[0]}, m)$

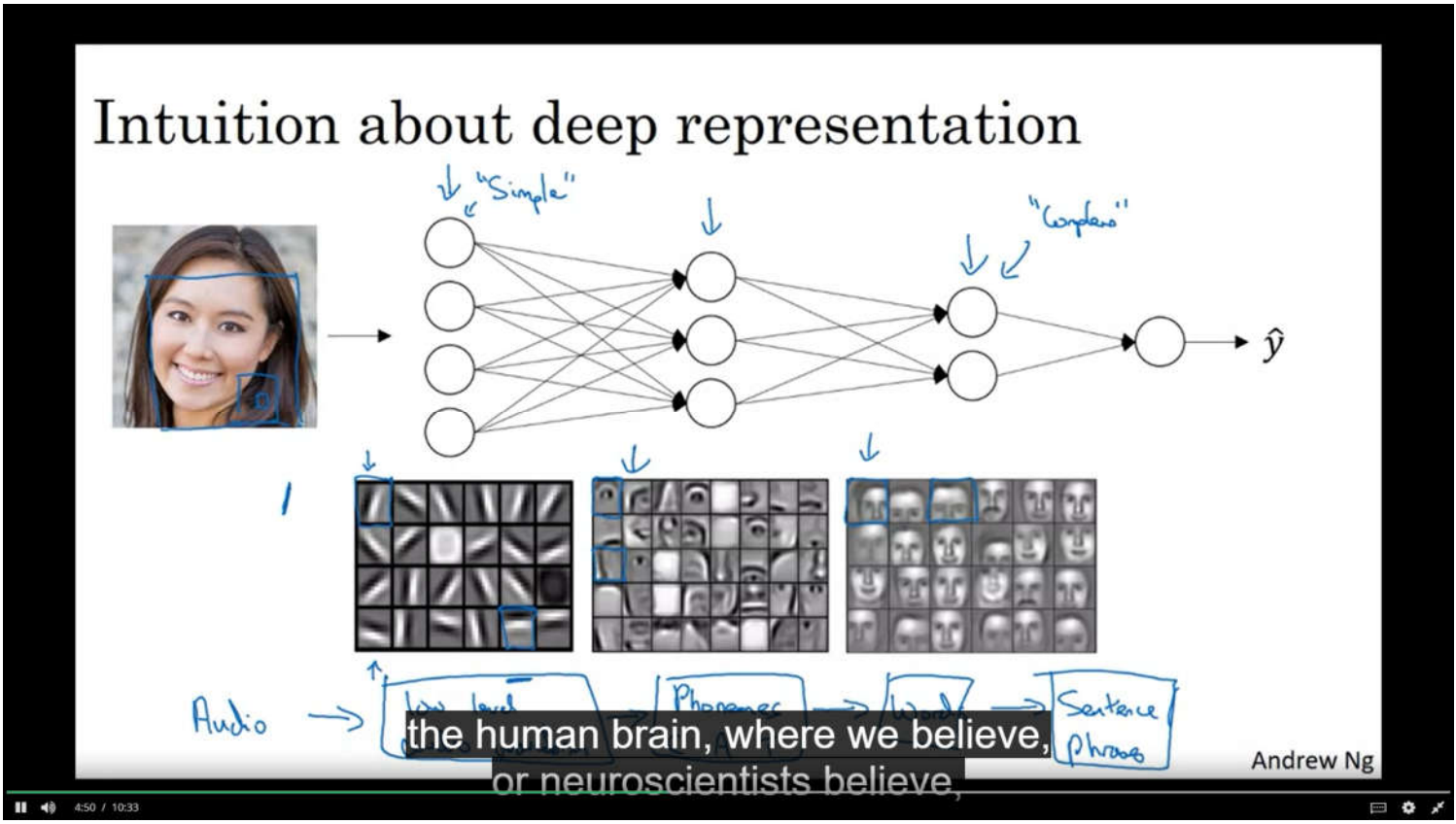
$dZ^{[l]}, dA^{[l]} : (n^{[l]}, m)$

Andrew Ng


Press Esc to exit full screen
Deep Neural Networks
Why deep representations?
deeplearning.ai
0:00 / 10:33


Intuition about deep representation
So for example, this little visualization
represents a hidden unit that's
Andrew Ng


Intuition about deep representation
$\hat{y}$
So intuitively, you can think of the
earlier layers of the neural network as
Andrew Ng


Intuition about deep representation
"Simple"
"Complex"
$\hat{y}$
Audio
Phonemes
Sentence
phrase
the human brain, where we believe,
or neuroscientists believe,
Andrew Ng
4:50 / 10:33

Circuit theory and deep learning

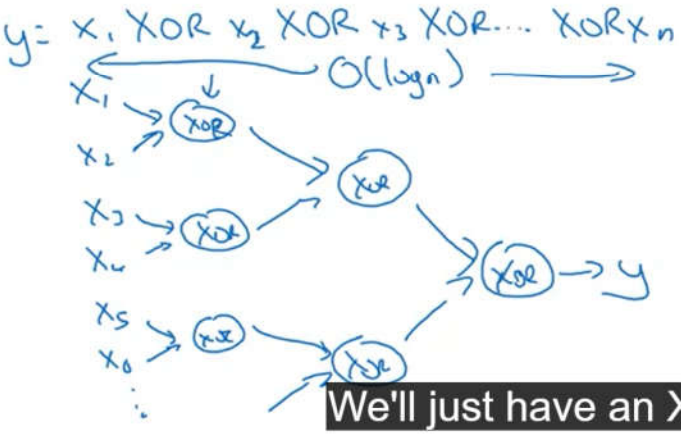
Informally: There are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute.

work well is the following.

Andrew Ng

Circuit theory and deep learning

Informally: There are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute.

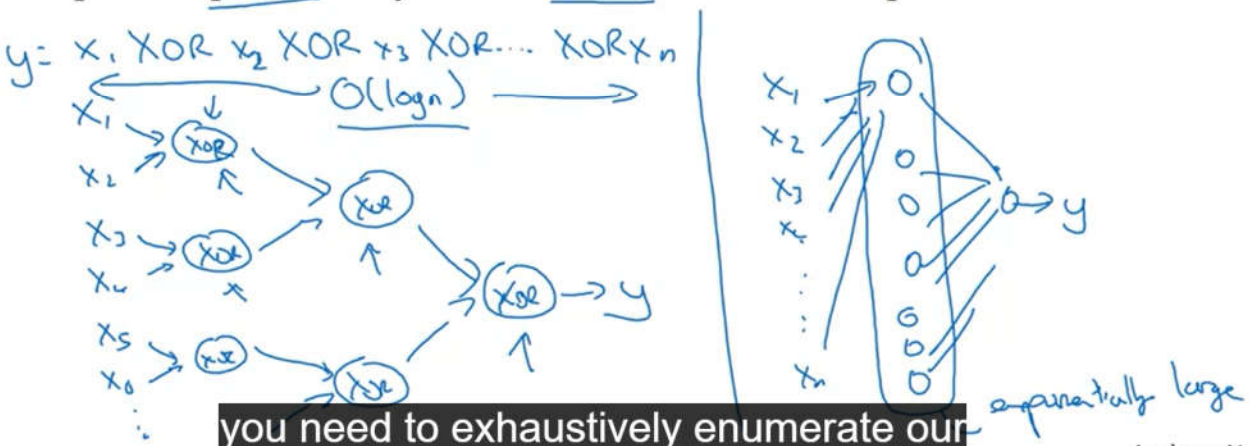


We'll just have an XOR tree.

Andrew Ng

Circuit theory and deep learning

Informally: There are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute.



you need to exhaustively enumerate our 2 to the N possible configurations.

Andrew Ng

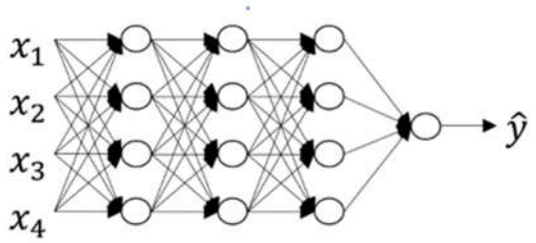
deeplearning.ai

Deep Neural Networks

Building blocks of deep neural networks

In the earlier videos from this week,

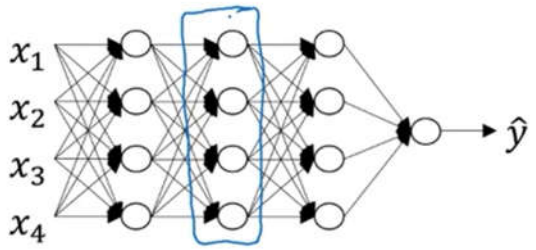
Forward and backward functions



Here's a network of a few layers.

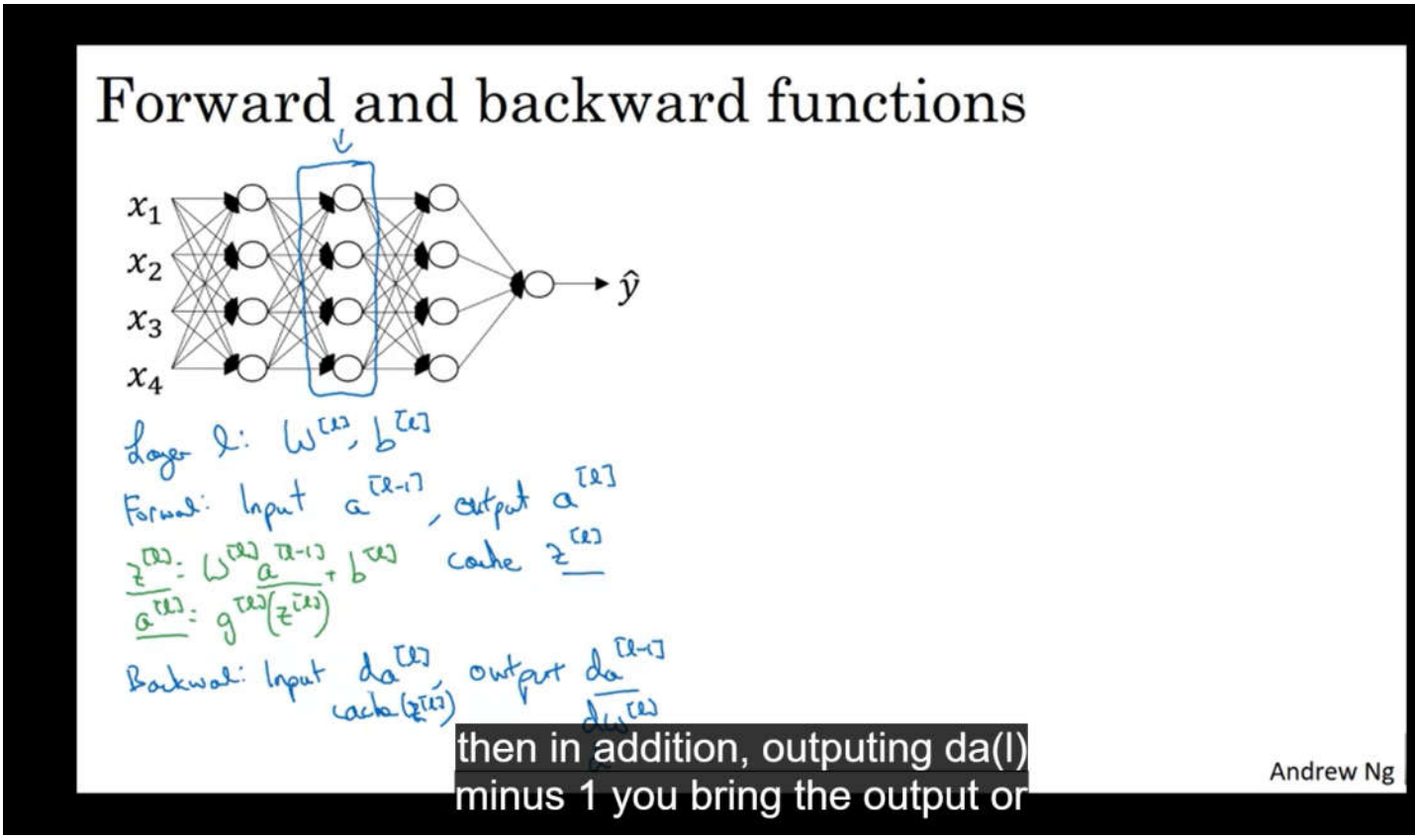
Andrew Ng

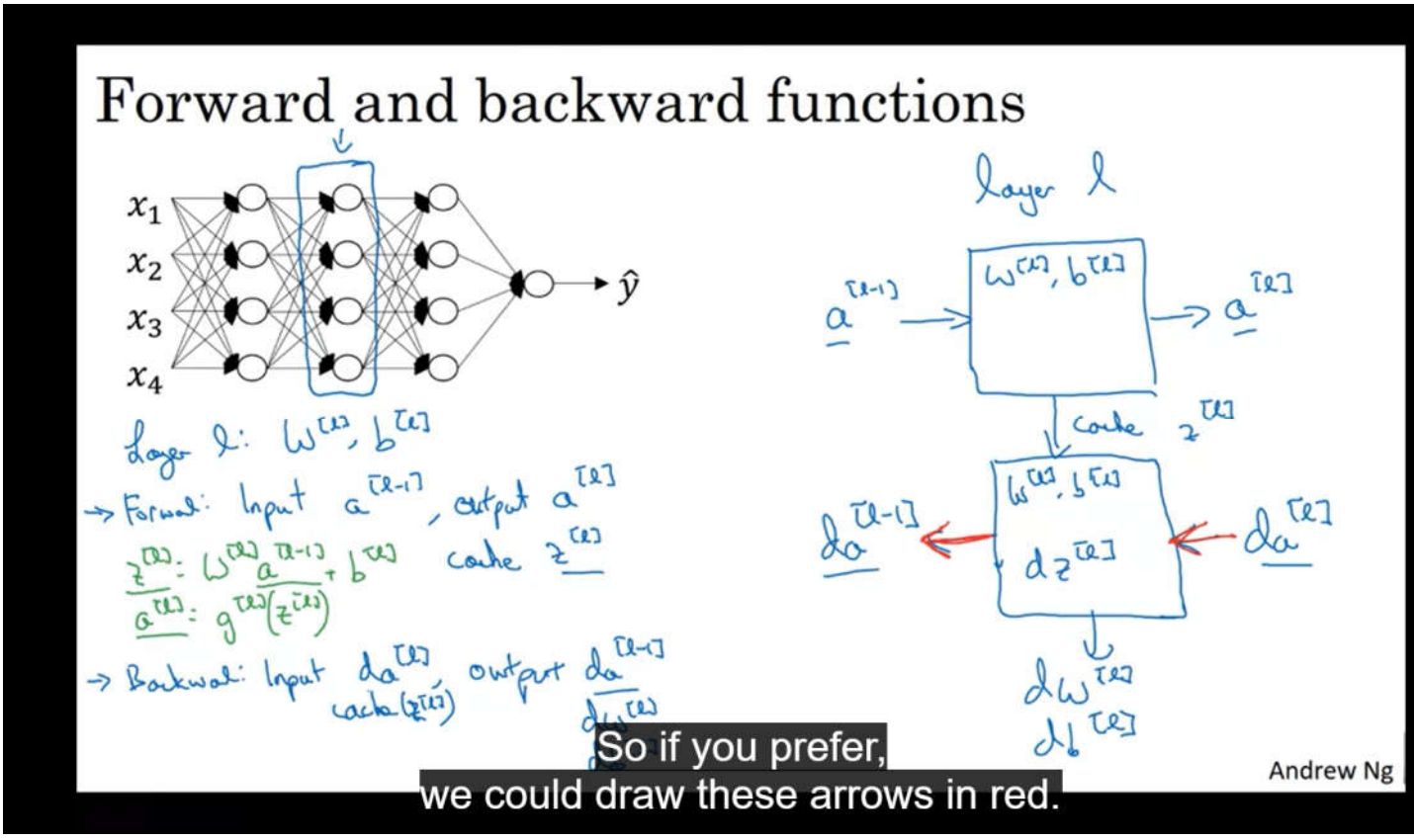
Forward and backward functions



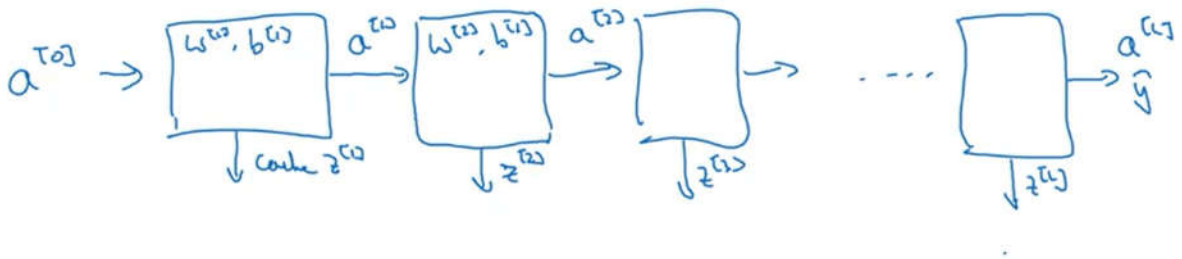
And look into the computations focusing on just that layer for now.

Andrew Ng


Forward and backward functions
x_1
x_2
x_3
x_4
$\hat{y}$
Layer l: $W^{[l]}, b^{[l]}$
Forward: Input $a^{[l-1]}$, output $a^{[l]}$
$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$ cache $z^{[l]}$
$a^{[l]} = g^{[l]}(z^{[l]})$
Backward: Input $da^{[l]}$, output $da^{[l-1]}$
cache ($z^{[l]}$)
$dW^{[l]}$
then in addition, outputing da(l)
minus 1 you bring the output or
Andrew Ng


Forward and backward functions
x_1
x_2
x_3
x_4
$\hat{y}$
Layer l: $W^{[l]}, b^{[l]}$
Forward: Input $a^{[l-1]}$, output $a^{[l]}$
$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$ cache $z^{[l]}$
$a^{[l]} = g^{[l]}(z^{[l]})$
Backward: Input $da^{[l]}$, output $da^{[l-1]}$
cache ($z^{[l]}$)
layer l
$a^{[l-1]}$
$W^{[l]}, b^{[l]}$
$a^{[l]}$
cache $z^{[l]}$
$W^{[l]}, b^{[l]}$
$da^{[l-1]}$
$dz^{[l]}$
$da^{[l]}$
$dW^{[l]}$
$db^{[l]}$
So if you prefer,
we could draw these arrows in red.
Andrew Ng

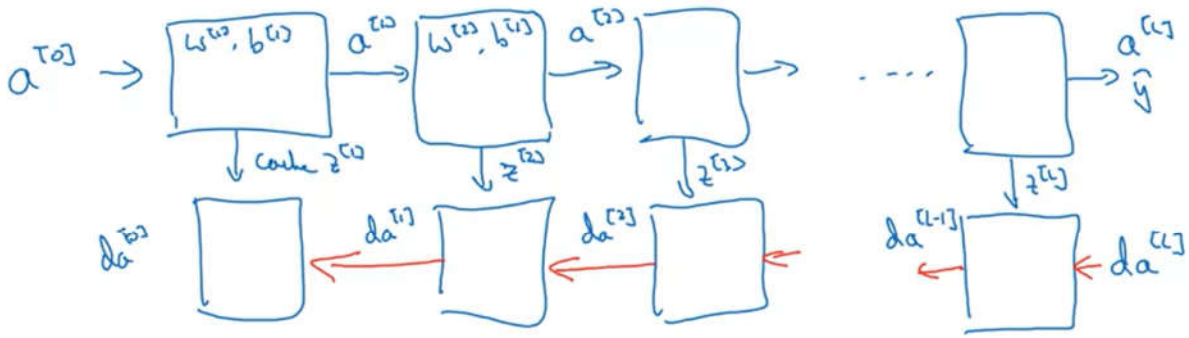
Forward and backward functions



Now, for the back propagation step, what we're going to do

Andrew Ng

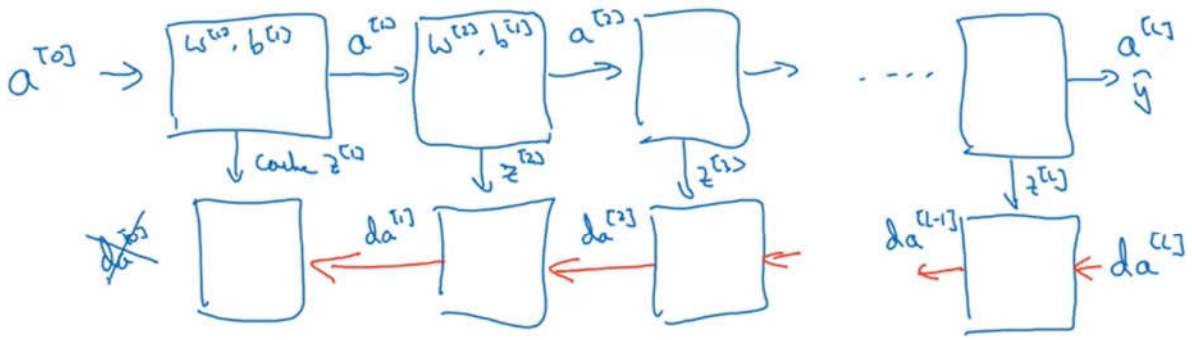
Forward and backward functions



this is derivative with respect to your

Andrew Ng

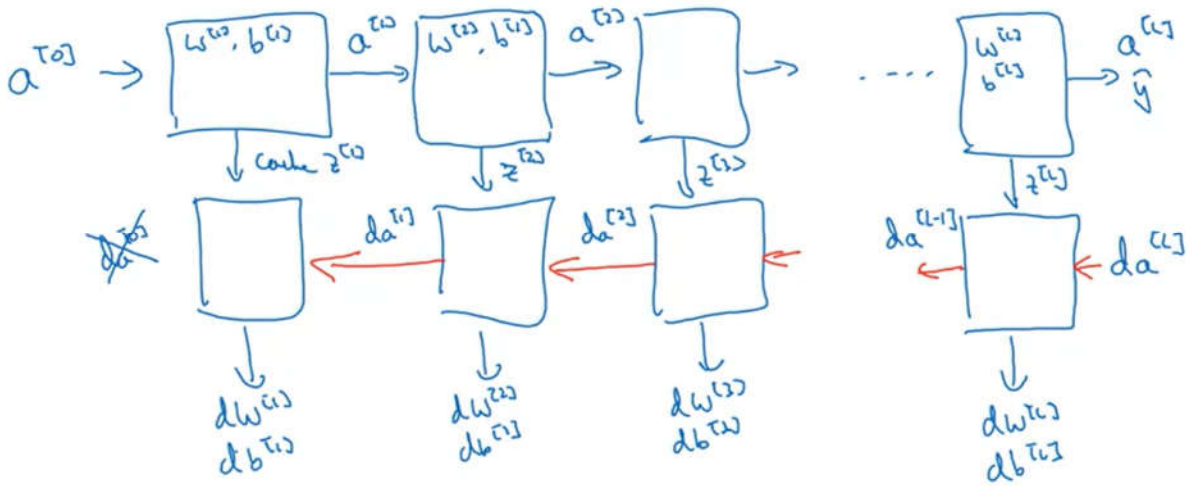
Forward and backward functions



training the weights of these supervised neural networks.

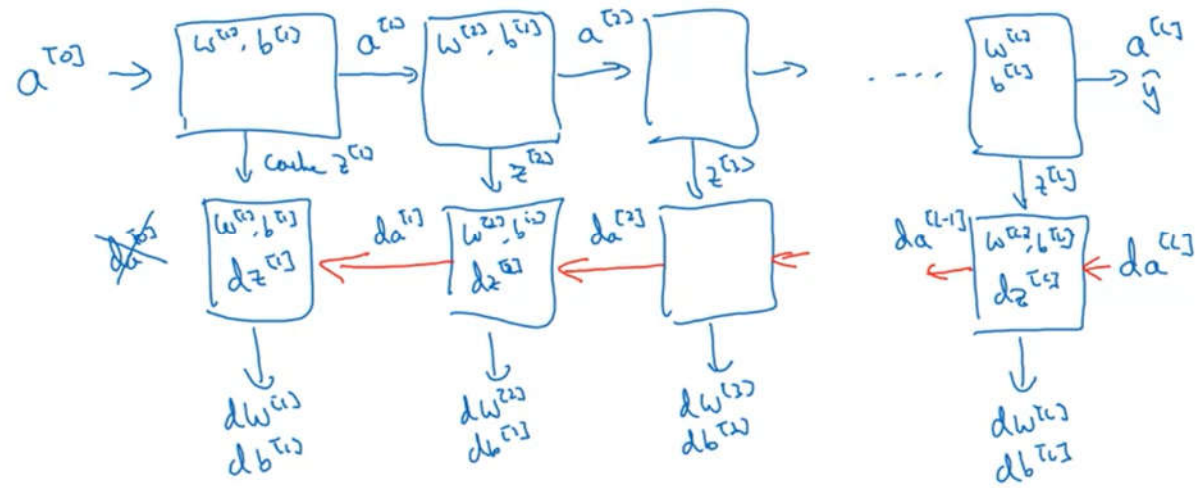
Andrew Ng

Forward and backward functions



Andrew Ng

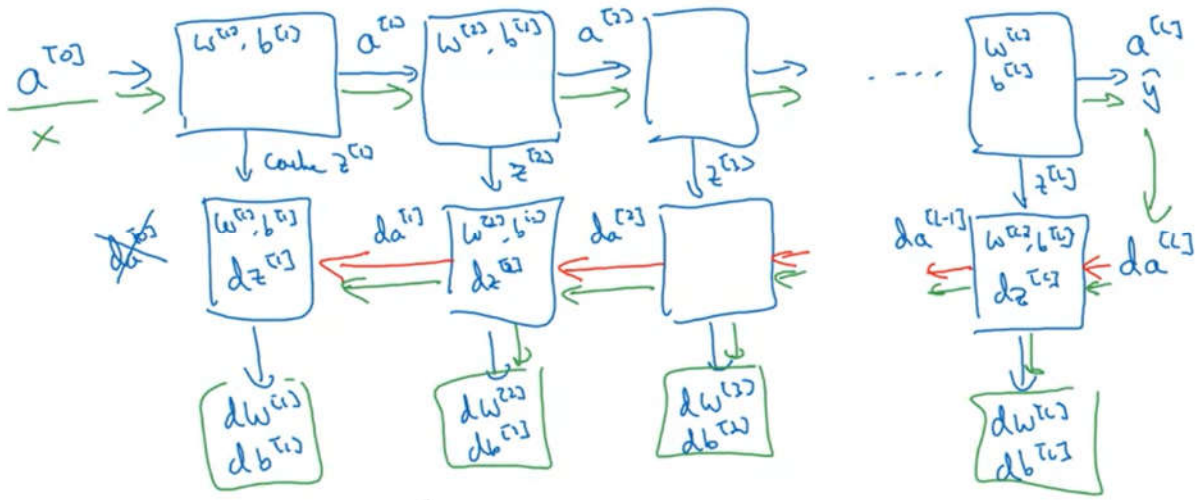
Forward and backward functions



So one iteration of training through
a neural network involves: starting with

Andrew Ng

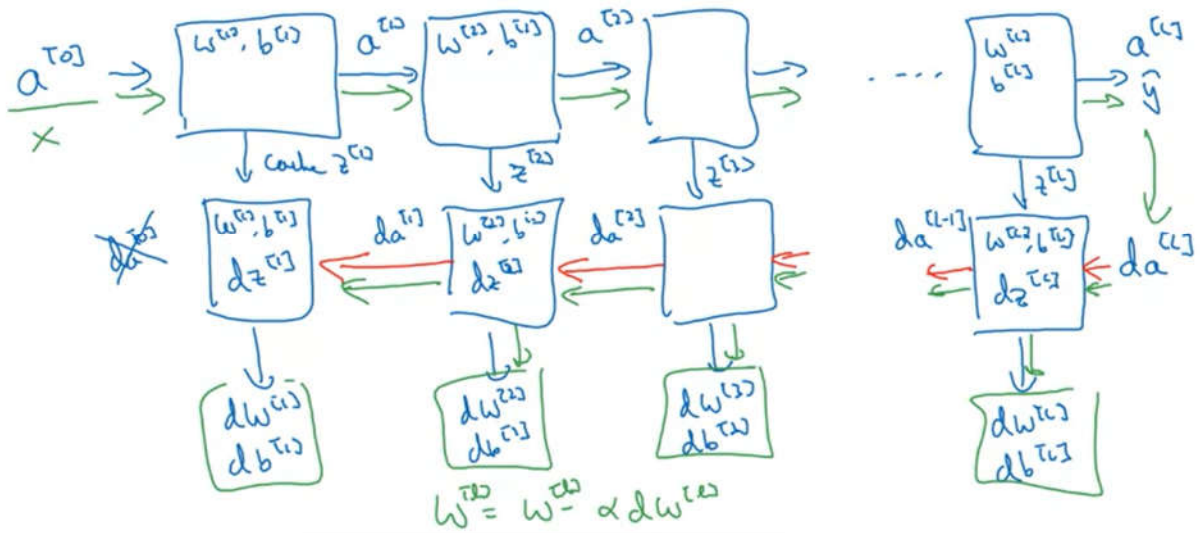
Forward and backward functions



now you have all these derivative
terms and so, you know,

Andrew Ng

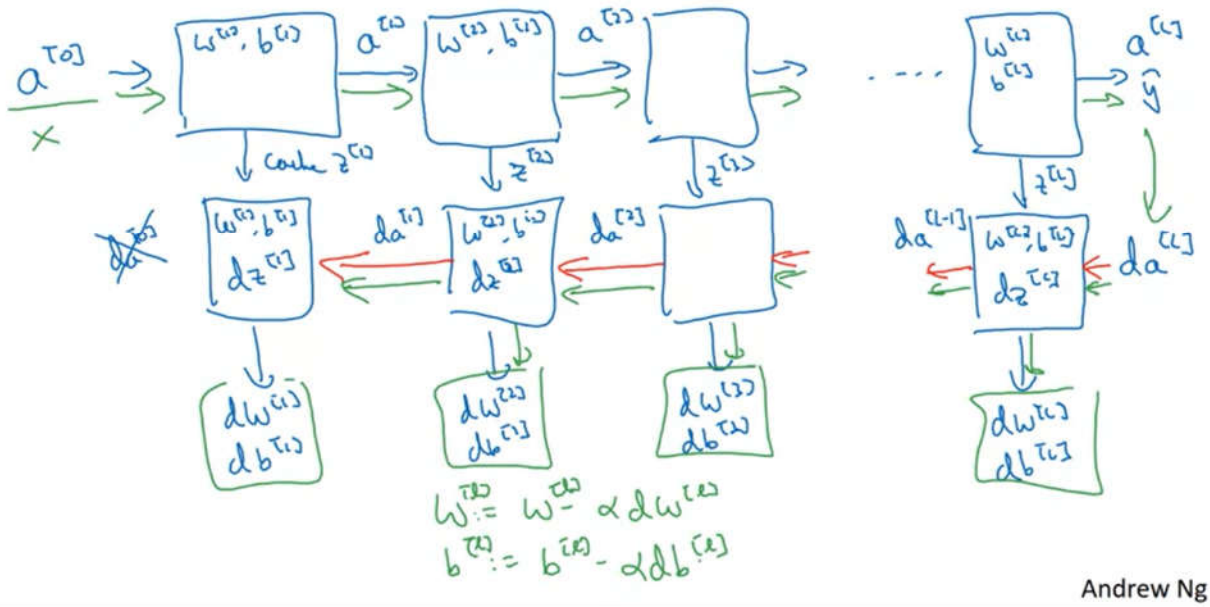
Forward and backward functions



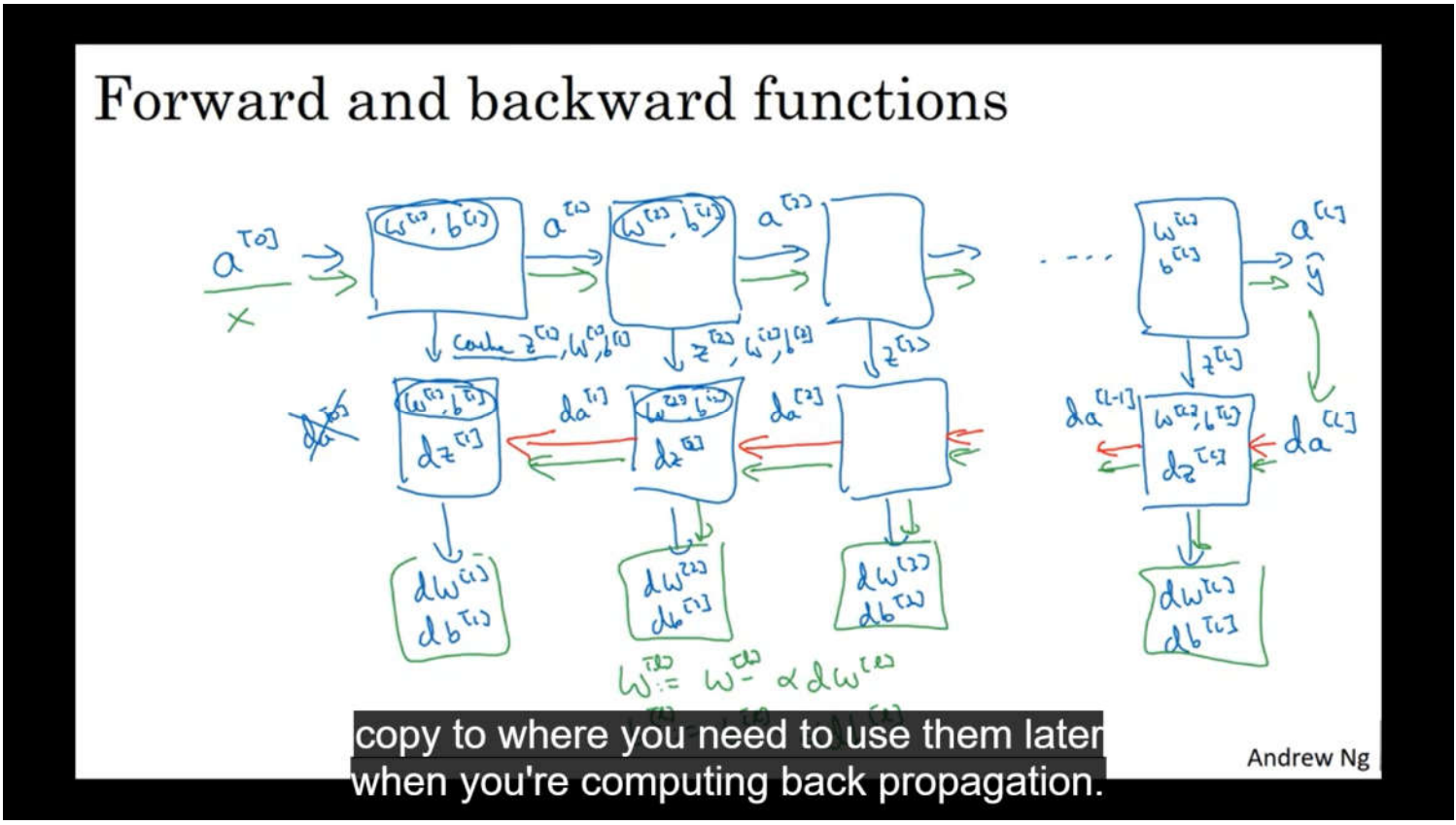
For each of the layers and similarly for b rate.

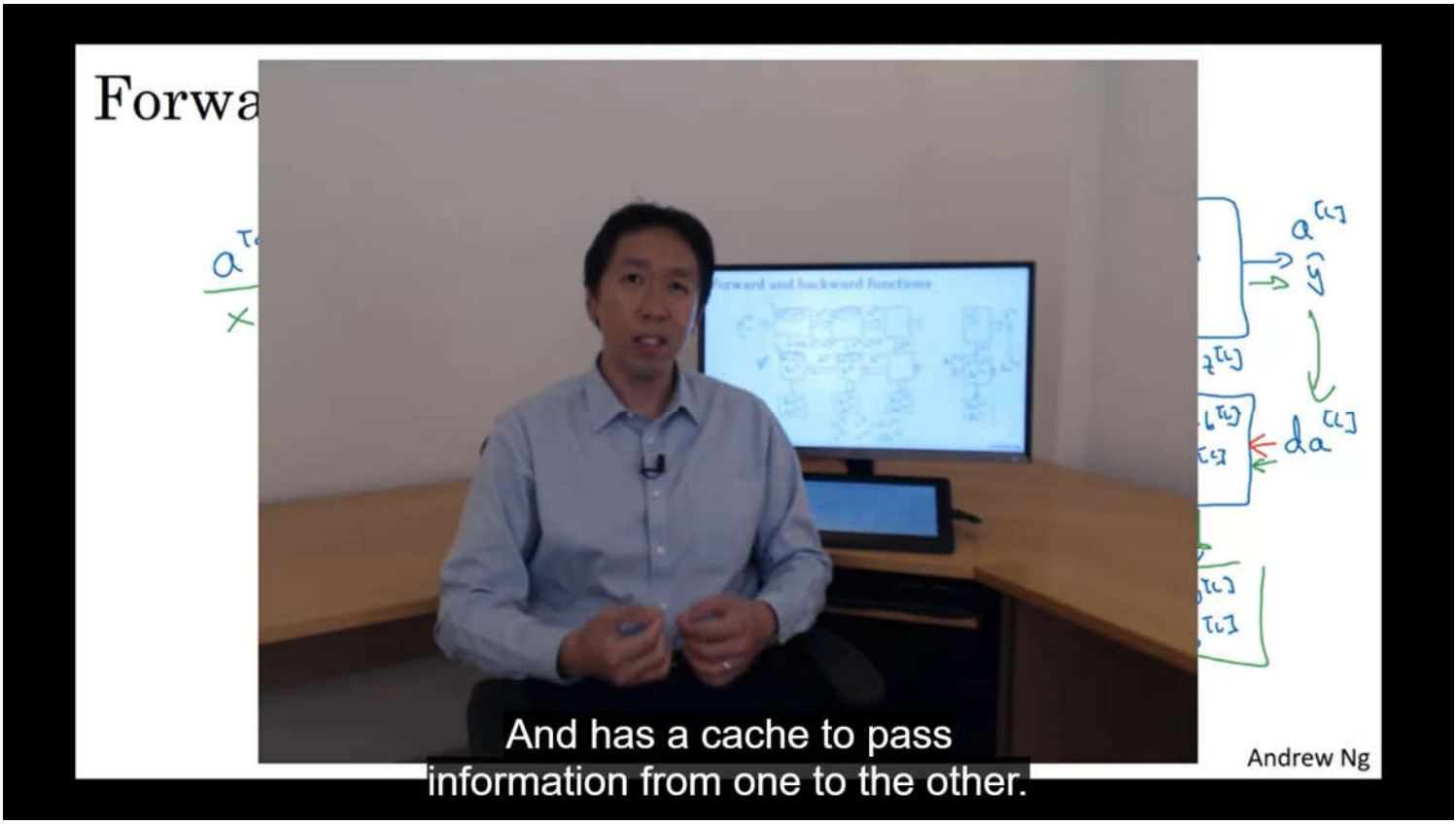
Andrew Ng

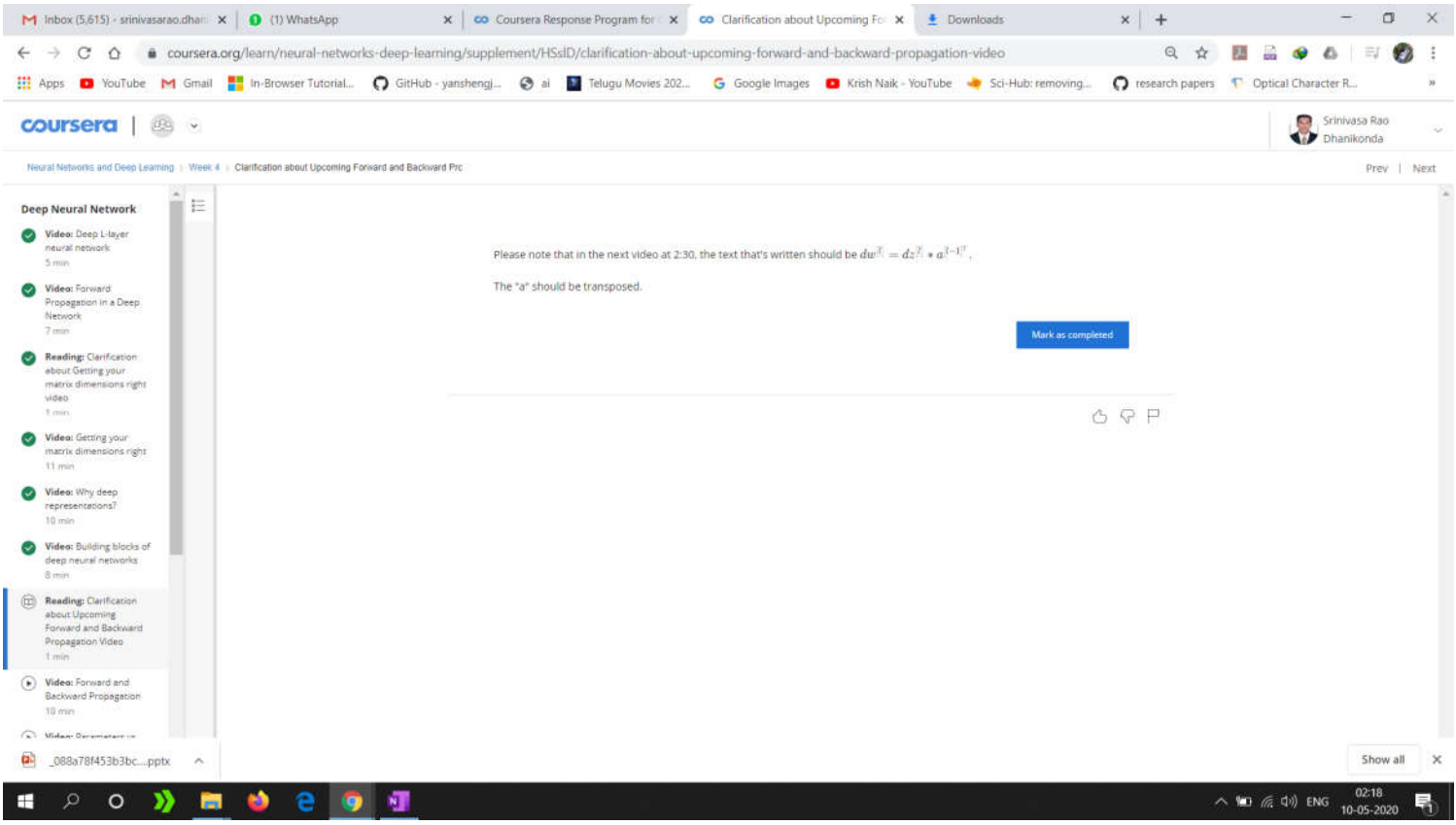
Forward and backward functions

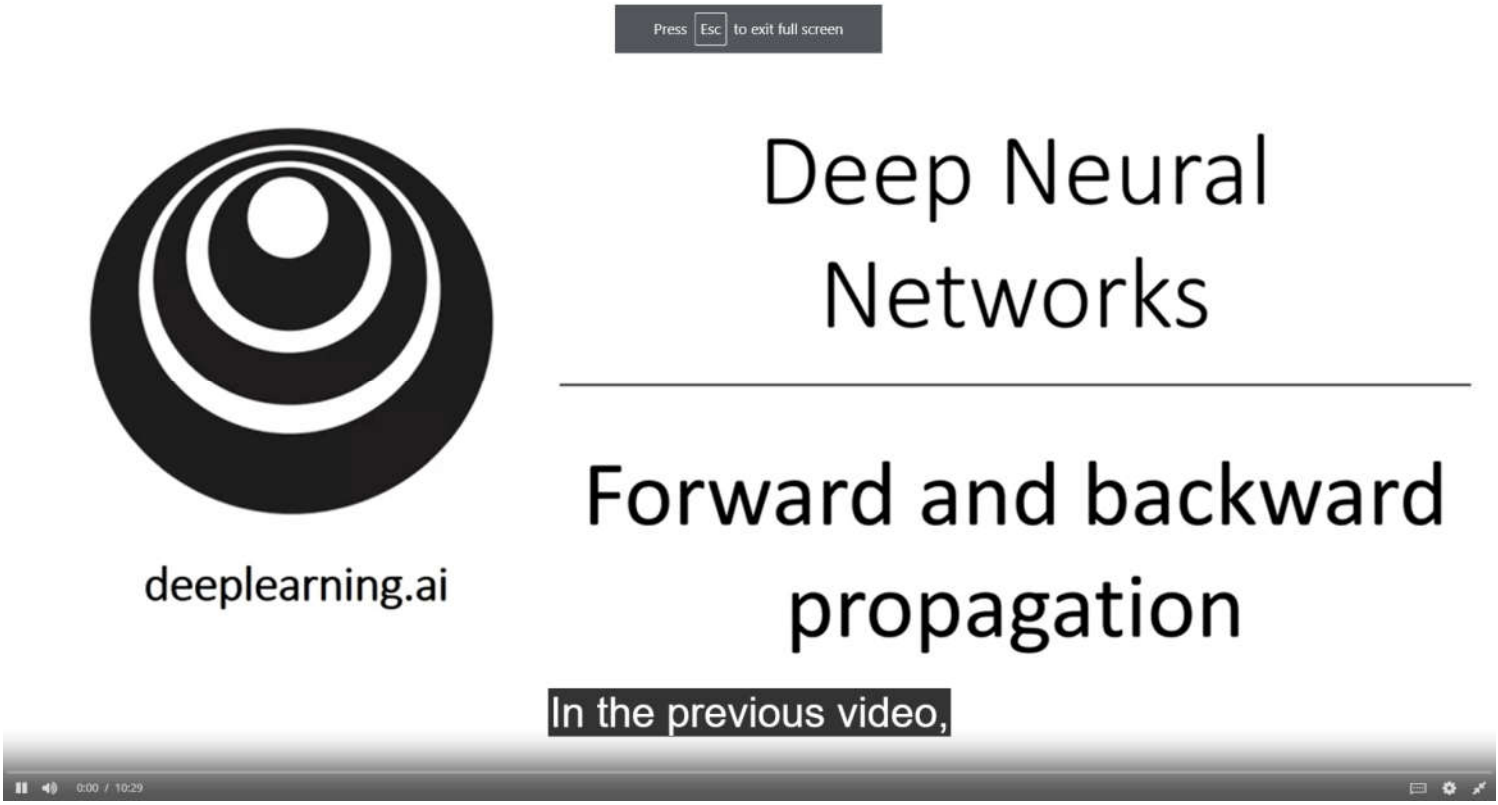


Andrew Ng


Forward and backward functions
copy to where you need to use them later
when you're computing back propagation.
Andrew Ng


And has a cache to pass
information from one to the other.
Andrew Ng


coursera.org/learn/neural-networks-deep-learning/supplement/HSsID/clarification-about-upcoming-forward-and-backward-propagation-video
coursera
Neural Networks and Deep Learning | Week 4 | Clarification about Upcoming Forward and Backward Prc
Prev | Next
Deep Neural Network
Video: Deep L-layer neural network
5 min
Video: Forward Propagation in a Deep Network
7 min
Reading: Clarification about Getting your matrix dimensions right video
1 min
Video: Getting your matrix dimensions right
11 min
Video: Why deep representations?
10 min
Video: Building blocks of deep neural networks
8 min
Reading: Clarification about Upcoming Forward and Backward Propagation Video
1 min
Video: Forward and Backward Propagation
10 min
Please note that in the next video at 2:30, the text that's written should be $dw^{[l]T} = dz^{[l]} * a^{[l-1]T}$,
The "a" should be transposed.
Mark as completed
Show all


Press Esc to exit full screen
deeplearning.ai
Deep Neural Networks
Forward and backward propagation
In the previous video,
0:00 / 10:29

Forward propagation for layer l

- Input $a^{[l-1]}$

  Output $a^{[l]}$, cache $(z^{[l]})$

Recall that what this will do is input a[l-1] and output a[l],

0:15 / 10:29

Andrew Ng

Forward propagation for layer l

→ Input $a^{[l-1]}$

→ Output $a^{[l]}$, cache $(z^{[l]})$ ← $W^{[l]}, b^{[l]}$

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

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

Vectorized:

$$Z^{[l]} = W^{[l]} \cdot A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = g^{[l]}(Z^{[l]})$$

And you remember, on the diagram for the forward step,

Andrew Ng

Forward propagation for layer l

→ Input $a^{[l-1]}$

→ Output $a^{[l]}$, cache ($z^{[l]}$)

$W^{[l]}, b^{[l]}$

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

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

Vectorized:

$Z^{[l]} = W^{[l]} \cdot A^{[l-1]} + b^{[l]}$

$A^{[l]} = g^{[l]}(Z^{[l]})$

$a^{[0]}$

$A^{[0]}$

$X = A^{[0]} \rightarrow \square \rightarrow \square \rightarrow \square \rightarrow$

compute forward propagation from left to right.

Andrew Ng

Backward propagation for layer l

→ Input $da^{[l]}$

→ Output $da^{[l-1]}, dW^{[l]}, db^{[l]}$

and output da[l-1] and dW[l] and db.

Andrew Ng

Backward propagation for layer l

→ Input $da^{[l]}$

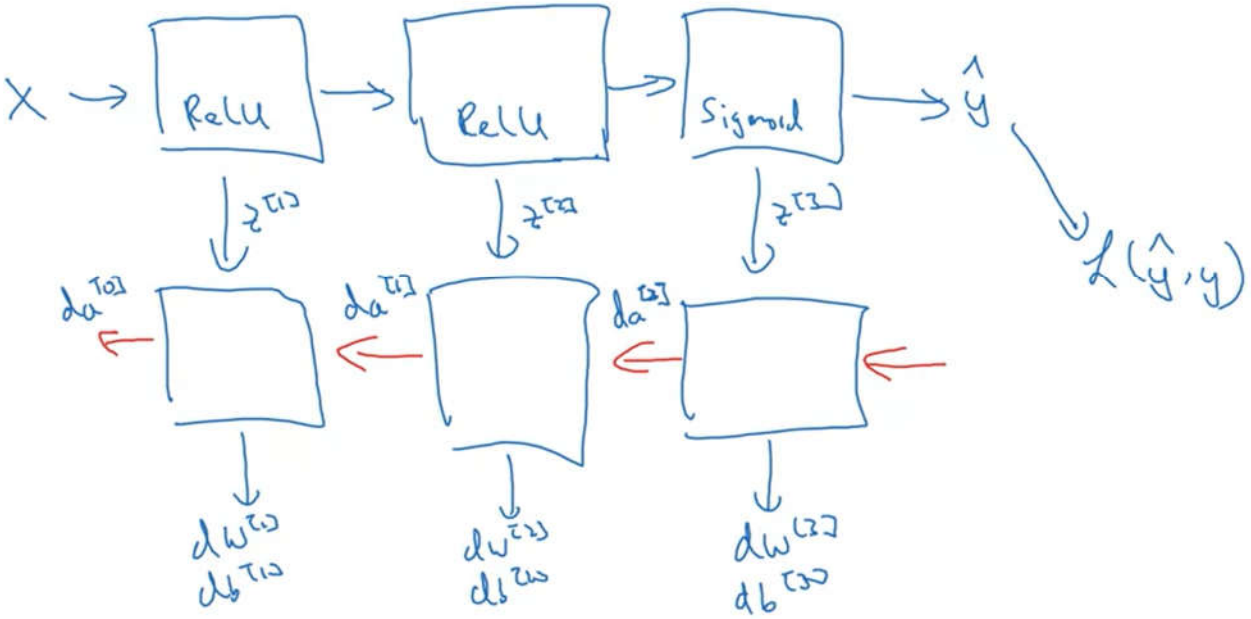
→ Output $da^{[l-1]}, dW^{[l]}, db^{[l]}$

$$dz^{[l]} = da^{[l]} * g^{[l]\prime}(z^{[l]})$$
$$dW^{[l]} = dz^{[l]} \cdot a^{[l-1]}$$
$$db^{[l]} = dz^{[l]}$$
$$da^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$$
$$dz^{[l]} = W^{[l+1]T} dz^{[l+1]} * g^{[l]\prime}(z^{[l]})$$

$$dZ^{[l]} = dA^{[l]} * g^{[l]\prime}(Z^{[l]})$$
$$dW^{[l]} = \frac{1}{m} dZ^{[l]} \cdot A^{[l-1]T}$$
$$db^{[l]} = \frac{1}{m} np.sum(dZ^{[l]}, axis=1, keepdims=True)$$
$$dA^{[l-1]} = W^{[l]T} \cdot dZ^{[l]}$$

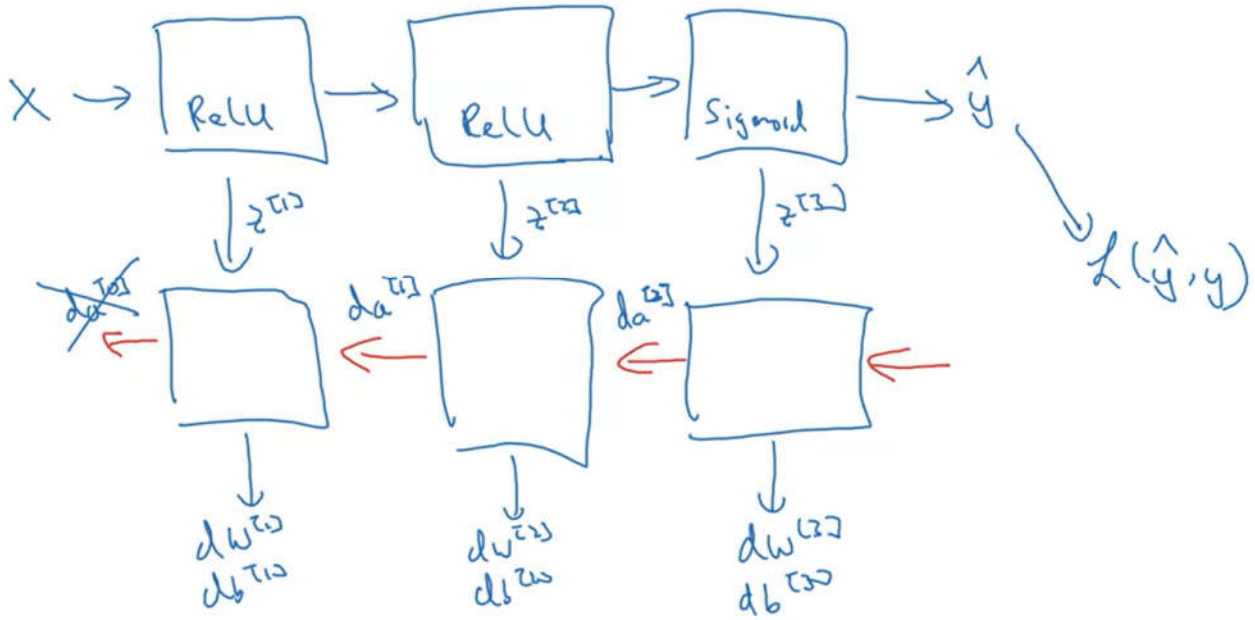
Andrew Ng

Summary



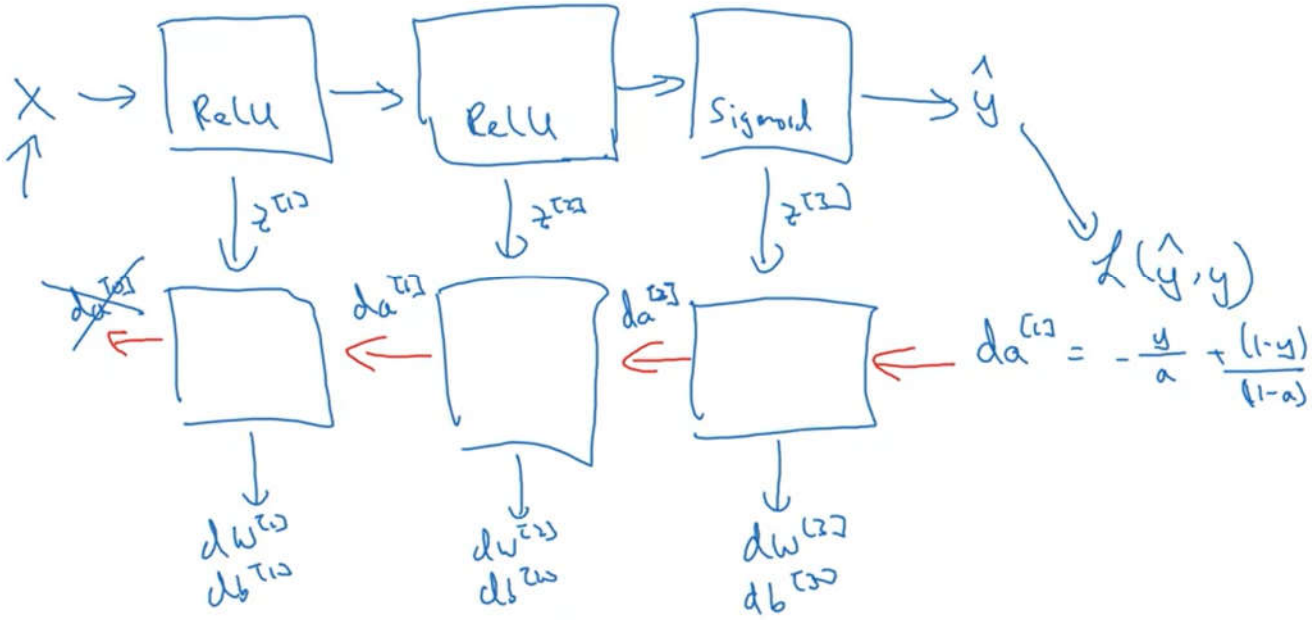
Andrew Ng

Summary



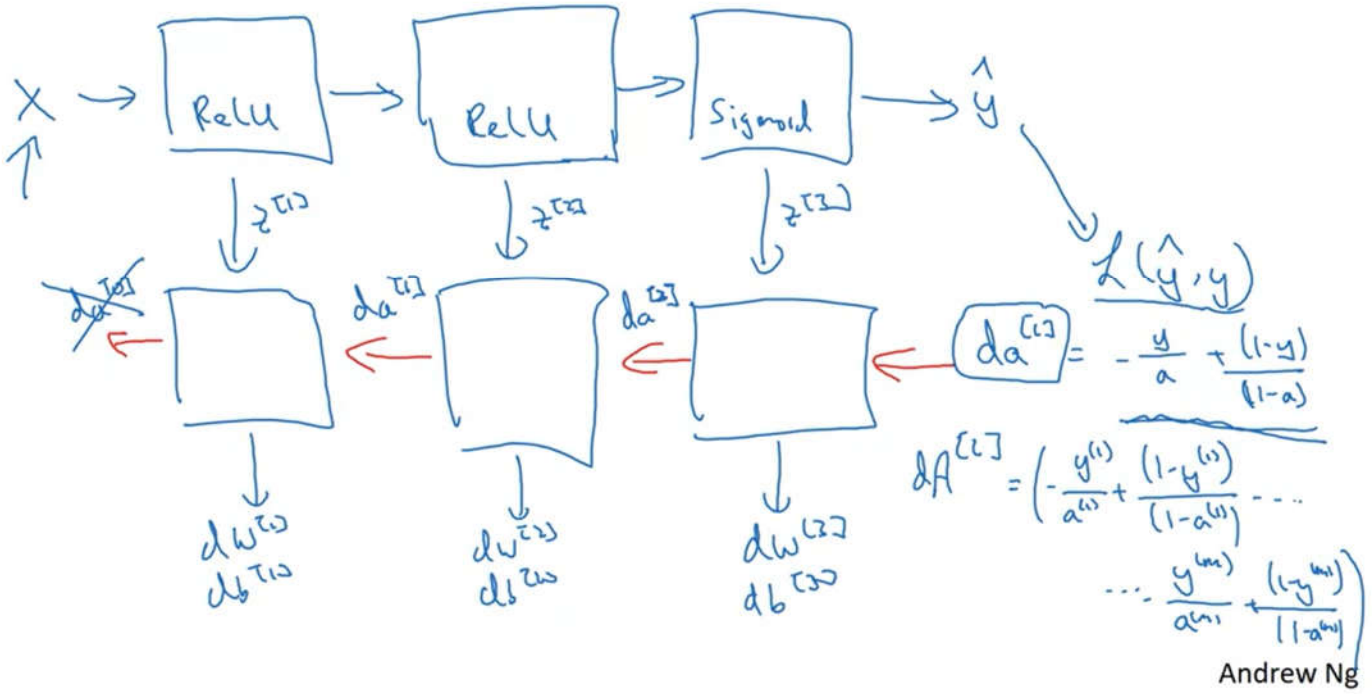
Andrew Ng

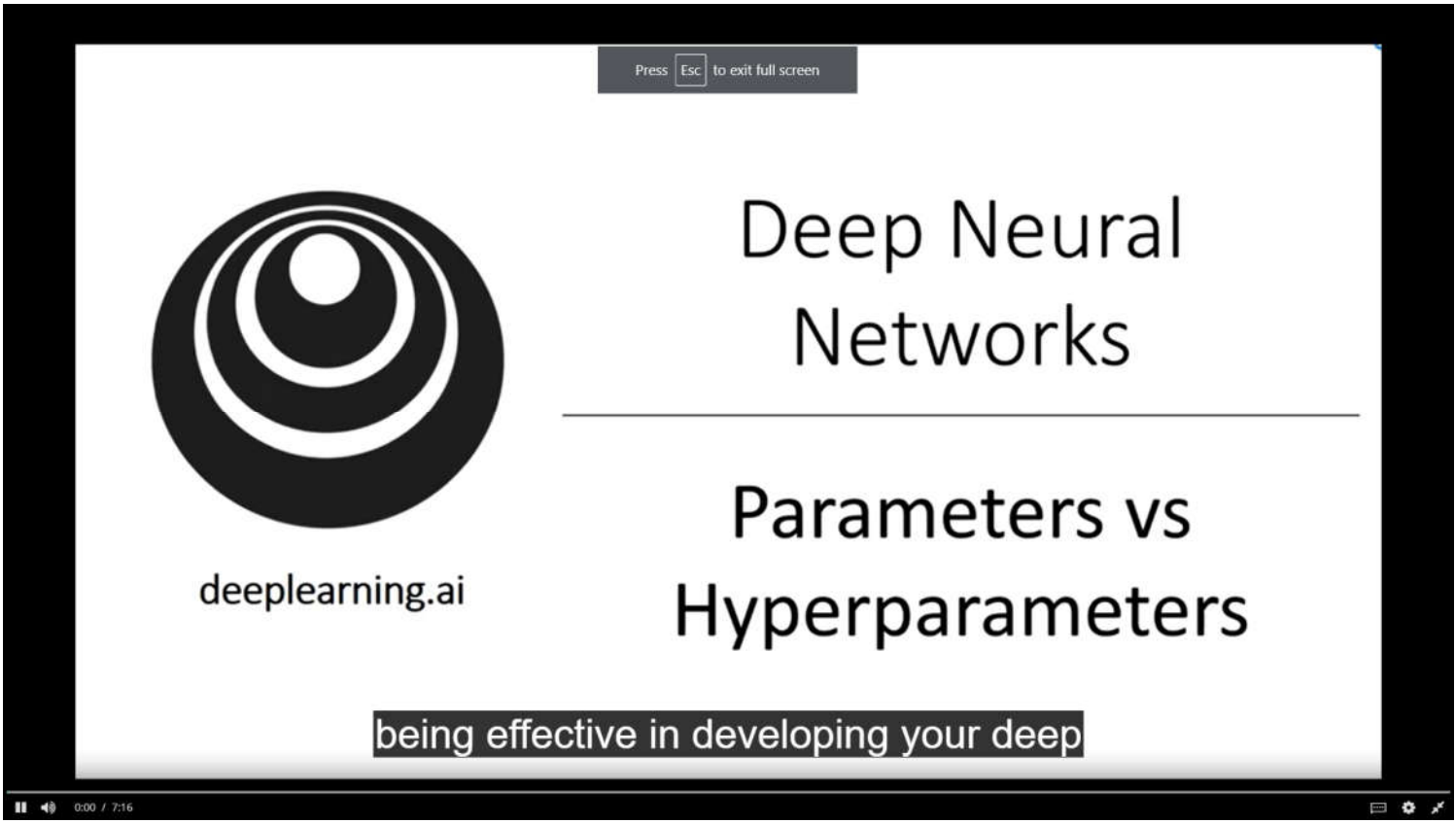
Summary



Andrew Ng

Summary


X
ReLU
ReLU
Sigmoid
ŷ
z[1]
z[2]
z[3]
da[0]
da[1]
da[2]
da[3] = -y/a + (1-y)/(1-a)
L(ŷ, y)
dW[1]
db[1]
dW[2]
db[2]
dW[3]
db[3]
dA[L] = (-y(1)/a(1) + (1-y(1))/(1-a(1)) ... - y(m)/a(m) + (1-y(m))/(1-a(m)))
Andrew Ng


Press Esc to exit full screen
Deep Neural Networks
Parameters vs Hyperparameters
deeplearning.ai
being effective in developing your deep
0:00 / 7:16

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

parameters your model are W and B and

Andrew Ng

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparamets: learning rate α

determine how your parameters evolve or

Andrew Ng

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparameters: learning rate α
#iterations

gradient descent you carry out your

Andrew Ng

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparameters: learning rate α
#iterations
#hidden layers L

capital L or the number of hidden units

Andrew Ng

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparameters: learning rate α

#iterations

hidden layers L

hidden units $n^{[1]}, n^{[2]}, \ldots$

so on and then you also have the choice

Andrew Ng

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparameters: learning rate α

#iterations

hidden layers L

hidden units $n^{[1]}, n^{[2]}, \ldots$

choice of activation

use a rel you or ten age or a sigma

Andrew Ng

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparameters: learning rate α
#iterations
hidden layers L
hidden units $n^{[1]}, n^{[2]}, \ldots$
Choice of activation function

these things below hyper parameters

Andrew Ng

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

Hyperparameters: learning rate α
#iterations
hidden layers L
hidden units $n^{[1]}, n^{[2]}, \ldots$
Choice of activation function

Later: Momentum, mini-batch

momentum term the mini batch size

Andrew Ng

What are hyperparameters?

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}, W^{[3]}, b^{[3]}$ ...

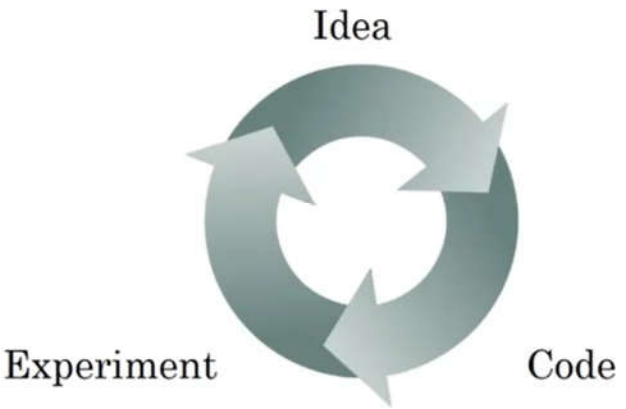
Hyperparameters: learning rate α
#iterations
#hidden layers L
#hidden units $n^{[1]}, n^{[2]}, \ldots$
choice of activation function

Later: Momentum, mini-batch size, regularizations

parameters and so on and if none of

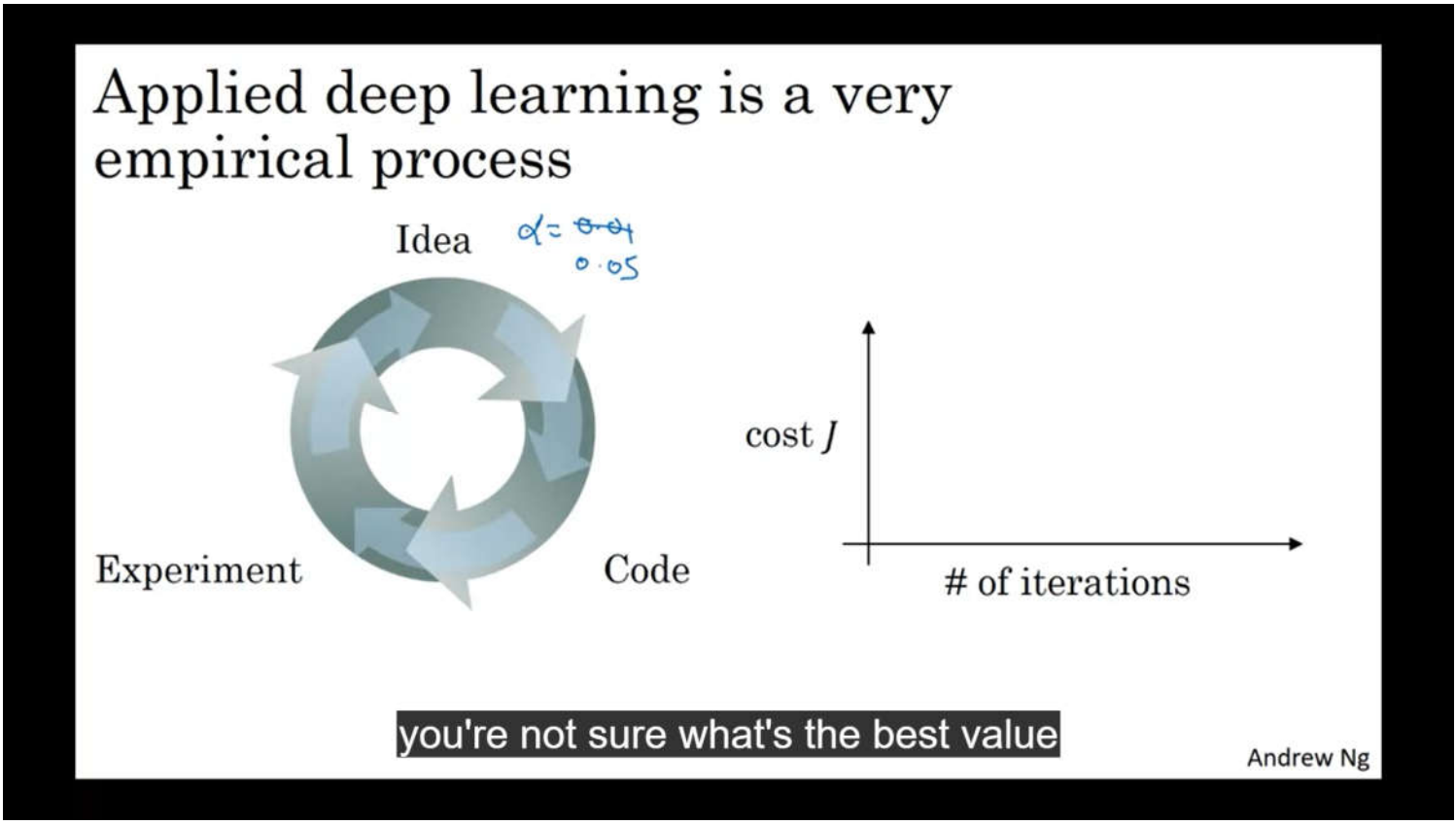
Andrew Ng

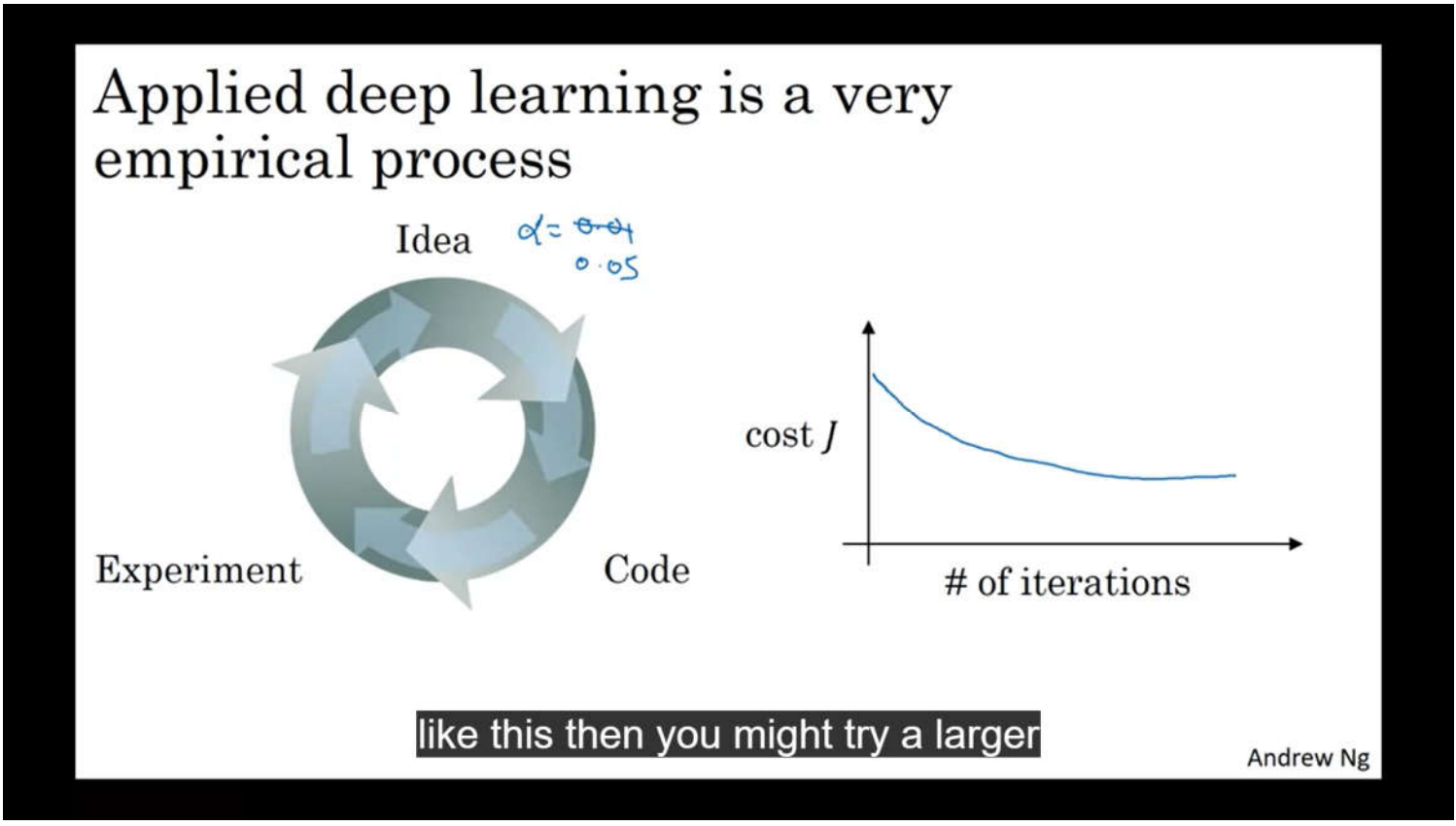
Applied deep learning is a very empirical process

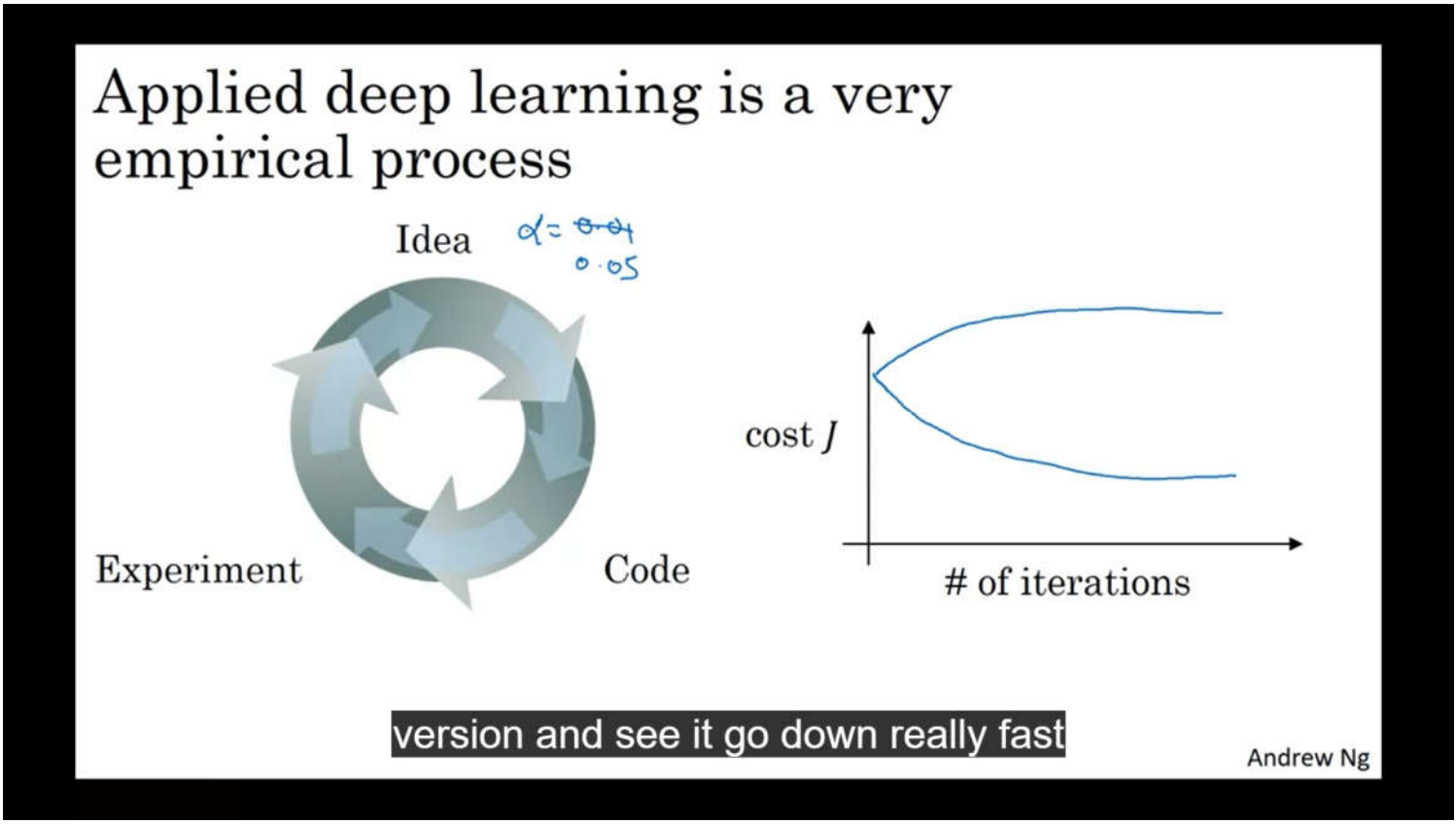


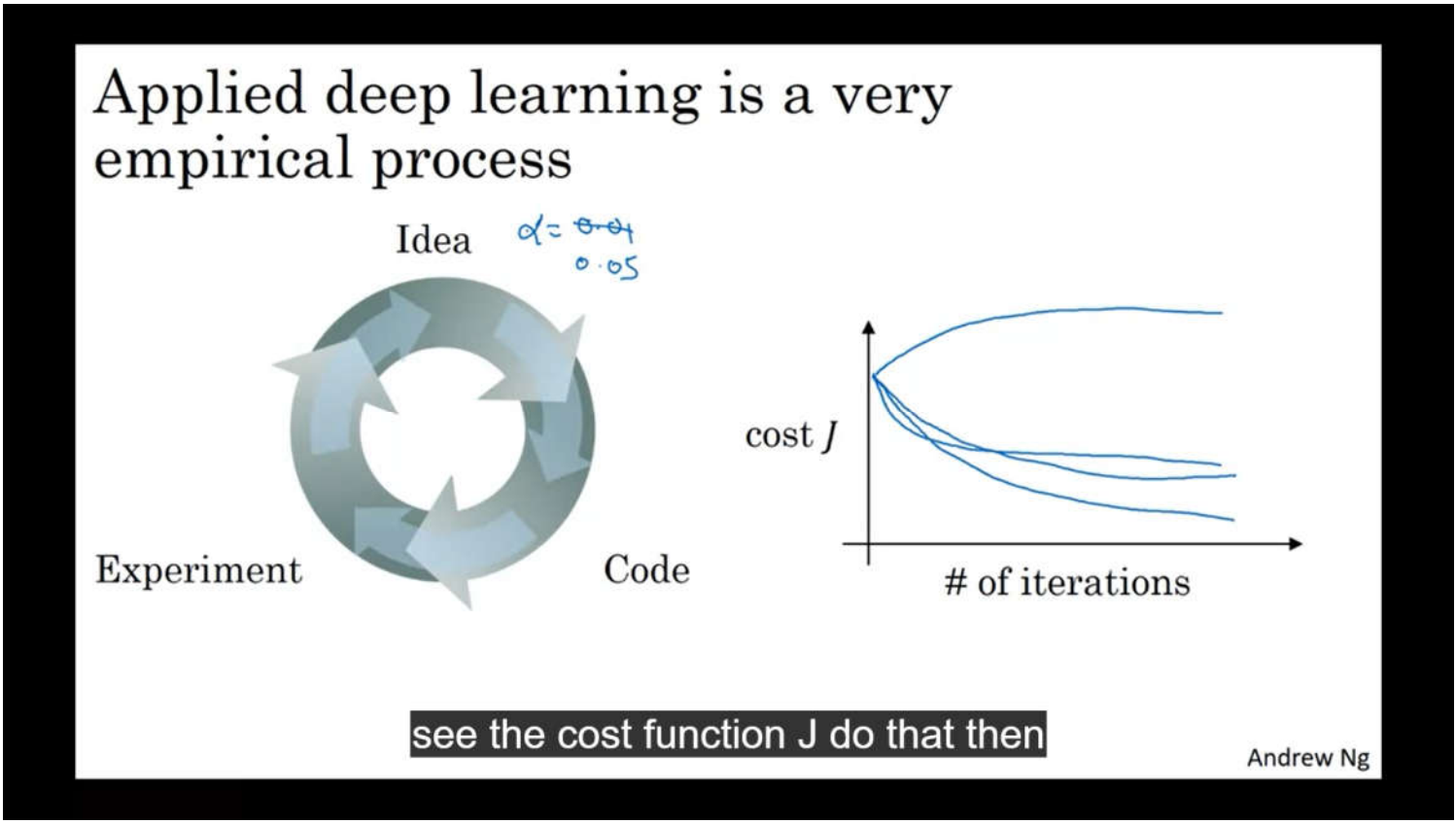
a very imperiled process where often you

Andrew Ng

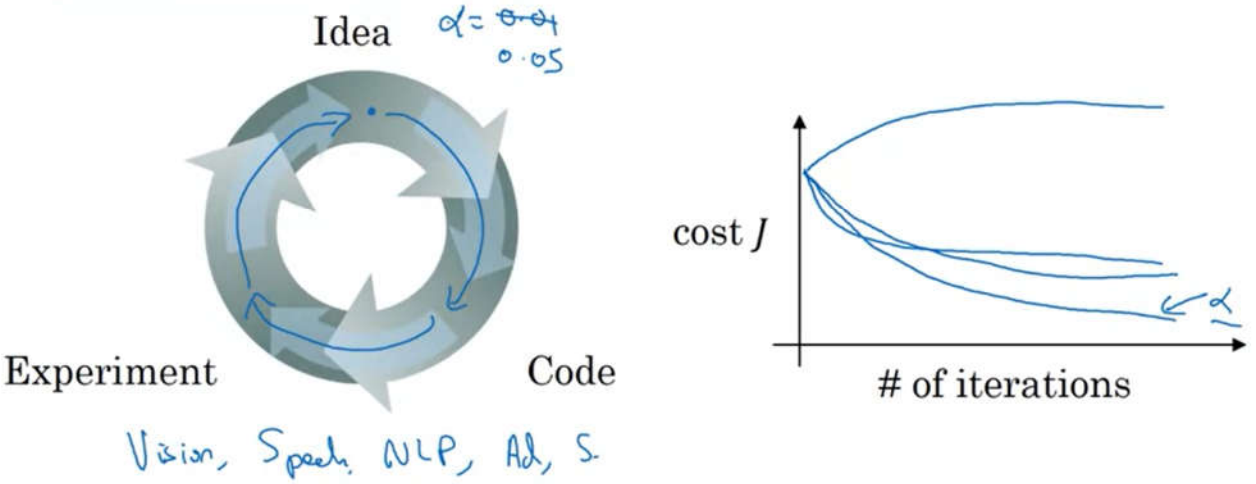

Applied deep learning is a very empirical process
Idea
α = ~~0.01~~
0.05
Experiment
Code
cost J
# of iterations
you're not sure what's the best value
Andrew Ng


Applied deep learning is a very empirical process
Idea
α = ~~0.01~~
0.05
Experiment
Code
cost J
# of iterations
like this then you might try a larger
Andrew Ng


Applied deep learning is a very empirical process
Idea
α = ~~0.01~~ 0.05
Experiment
Code
cost J
# of iterations
version and see it go down really fast
Andrew Ng


Applied deep learning is a very empirical process
Idea
α = ~~0.01~~ 0.05
Experiment
Code
cost J
# of iterations
see the cost function J do that then
Andrew Ng

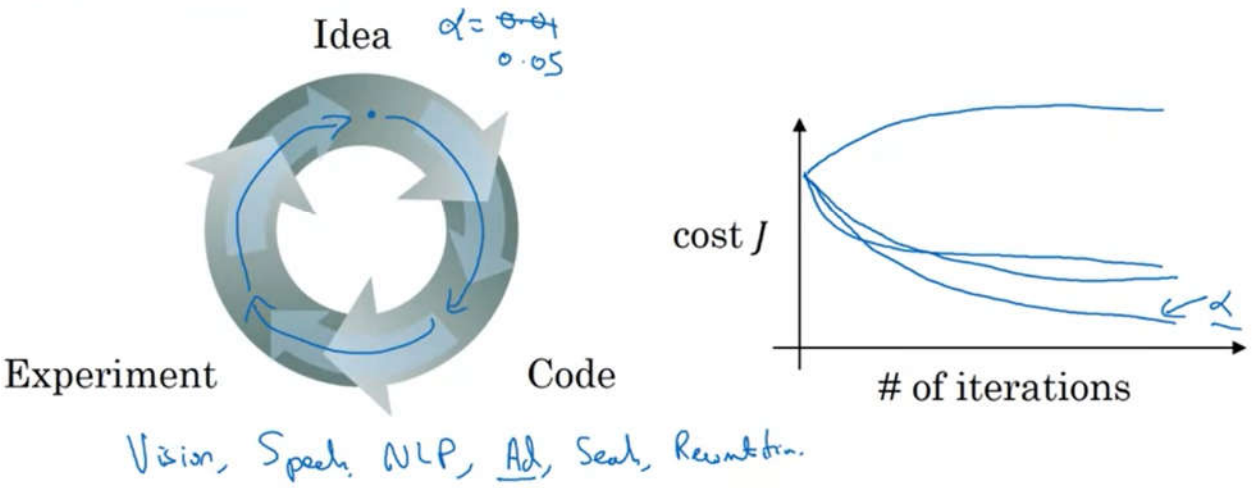
Applied deep learning is a very empirical process



maybe a online advertising or web search

Andrew Ng

Applied deep learning is a very empirical process



guidance for the best hyper parameters

Andrew Ng

coursera.org/learn/neural-networks-deep-learning/supplement/E79Uh/clarification-about-what-does-this-have-to-do-with-the-brain-video

coursera

Neural Networks and Deep Learning | Week 4 | Clarification about What does this have to do with the bra

Prev | Next

matrix dimensions right
11 min

Video: Why deep representations?
10 min

Video: Building blocks of deep neural networks
8 min

Reading: Clarification about Upcoming Forward and Backward Propagation Video
1 min

Video: Forward and Backward Propagation
10 min

Video: Parameters vs Hyperparameters
7 min

Reading: Clarification about What does this have to do with the brain video
1 min

Video: What does this have to do with the brain?
3 min

Practice Questions

Programming Assignments

Note that the formulas shown in the next video have a few typos. Here is the correct set of formulas.

$$dZ^{[L]} = A^{[L]} - Y$$

$$dW^{[L]} = \frac{1}{m} dZ^{[L]} A^{[L-1]^T}$$

$$db^{[L]} = \frac{1}{m} np.sum(dZ^{[L]}, axis = 1, keepdims = True)$$

$$dZ^{[L-1]} = W^{[L]^T} dZ^{[L]} * g'^{[L-1]}(Z^{[L-1]})$$

Note that * denotes element-wise multiplication)

$$\vdots$$

$$dZ^{[1]} = W^{[2]} dZ^{[2]} * g'^{[1]}(Z^{[1]})$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[0]^T}$$

Note that $A^{[0]^T}$ is another way to denote the input features, which is also written as X^T

$$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$$

Mark as completed

_3633684dda5ffd....pptx _088a78f453b3bc....pptx Show all

ENG 02:36 10-05-2020

deeplearning.ai

Deep Neural Networks

What does this have to do with the brain?

0:02 / 3:17

Forward and backward propagation

$$
\begin{aligned}
Z^{[1]} &= W^{[1]}X + b^{[1]} \\
A^{[1]} &= g^{[1]}(Z^{[1]}) \\
Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\
A^{[2]} &= g^{[2]}(Z^{[2]}) \\
&\vdots \\
A^{[L]} &= g^{[L]}(Z^{[L]}) = \hat{Y}
\end{aligned}
$$

$$
\begin{aligned}
dZ^{[L]} &= A^{[L]} - Y \\
dW^{[L]} &= \frac{1}{m} dZ^{[L]} A^{[L]^T} \\
db^{[L]} &= \frac{1}{m} np.\mathrm{sum}(\mathrm{d}Z^{[L]}, axis = 1, keepdims = True) \\
dZ^{[L-1]} &= dW^{[L]^T} dZ^{[L]} g'^{[L]}(Z^{[L-1]}) \\
&\vdots \\
dZ^{[1]} &= dW^{[L]^T} dZ^{[2]} g'^{[1]}(Z^{[1]}) \\
dW^{[1]} &= \frac{1}{m} dZ^{[1]} A^{[1]^T} \\
db^{[1]} &= \frac{1}{m} np.\mathrm{sum}(\mathrm{d}Z^{[1]}, axis = 1, keepdims = True)
\end{aligned}
$$

0:16 / 3:17

Andrew Ng

Forward and backward propagation

$$
\begin{aligned}
Z^{[1]} &= W^{[1]}X + b^{[1]} \\
A^{[1]} &= g^{[1]}(Z^{[1]}) \\
Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\
A^{[2]} &= g^{[2]}(Z^{[2]}) \\
&\vdots \\
A^{[L]} &= g^{[L]}(Z^{[L]}) = \hat{Y}
\end{aligned}
$$

$$
\begin{aligned}
dZ^{[L]} &= A^{[L]} - Y \\
dW^{[L]} &= \frac{1}{m} dZ^{[L]} A^{[L]^T} \\
db^{[L]} &= \frac{1}{m} np.\mathrm{sum}(\mathrm{d}Z^{[L]}, axis = 1, keepdims = True) \\
dZ^{[L-1]} &= dW^{[L]^T} dZ^{[L]} g'^{[L]}(Z^{[L-1]}) \\
&\vdots \\
dZ^{[1]} &= dW^{[L]^T} dZ^{[2]} g'^{[1]}(Z^{[1]}) \\
dW^{[1]} &= \frac{1}{m} dZ^{[1]} A^{[1]^T} \\
db^{[1]} &= \frac{1}{m} np.\mathrm{sum}(\mathrm{d}Z^{[1]}, axis = 1, keepdims = True)
\end{aligned}
$$

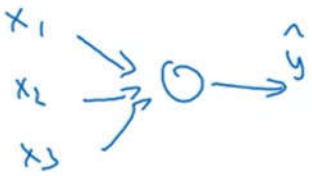
Andrew Ng

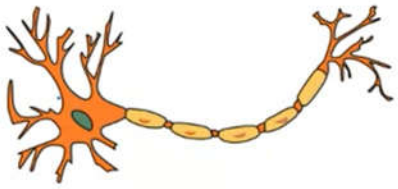
Forward and backward propagation

$$
\begin{aligned}
Z^{[1]} &= W^{[1]}X + b^{[1]} \\
A^{[1]} &= g^{[1]}(Z^{[1]}) \\
Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\
A^{[2]} &= g^{[2]}(Z^{[2]}) \\
&\vdots \\
A^{[L]} &= g^{[L]}(Z^{[L]}) = \hat{Y}
\end{aligned}
$$

$$
\begin{aligned}
dZ^{[L]} &= A^{[L]} - Y \\
dW^{[L]} &= \frac{1}{m} dZ^{[L]} A^{[L]^T} \\
db^{[L]} &= \frac{1}{m} np.\mathrm{sum}(\mathrm{d}Z^{[L]}, axis = 1, keepdims = True) \\
dZ^{[L-1]} &= dW^{[L]^T} dZ^{[L]} g'^{[L]}(Z^{[L-1]}) \\
&\vdots \\
dZ^{[1]} &= dW^{[L]^T} dZ^{[2]} g'^{[1]}(Z^{[1]}) \\
dW^{[1]} &= \frac{1}{m} dZ^{[1]} A^{[1]^T} \\
db^{[1]} &= \frac{1}{m} np.\mathrm{sum}(\mathrm{d}Z^{[1]}, axis = 1, keepdims = True)
\end{aligned}
$$

"It's like the brain."





Andrew Ng