

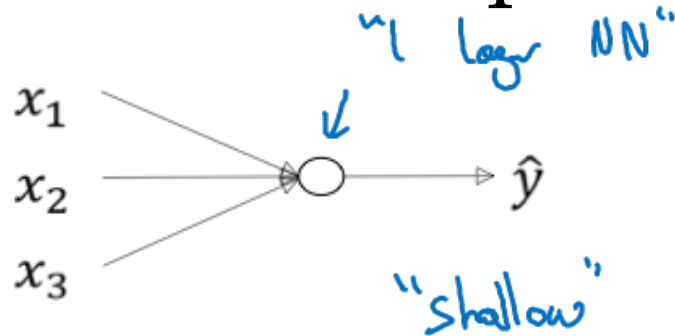


deeplearning.ai

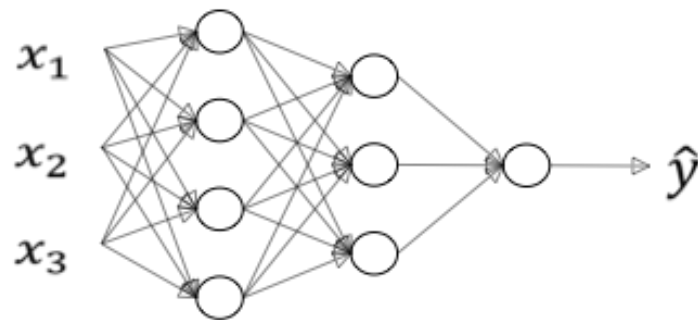
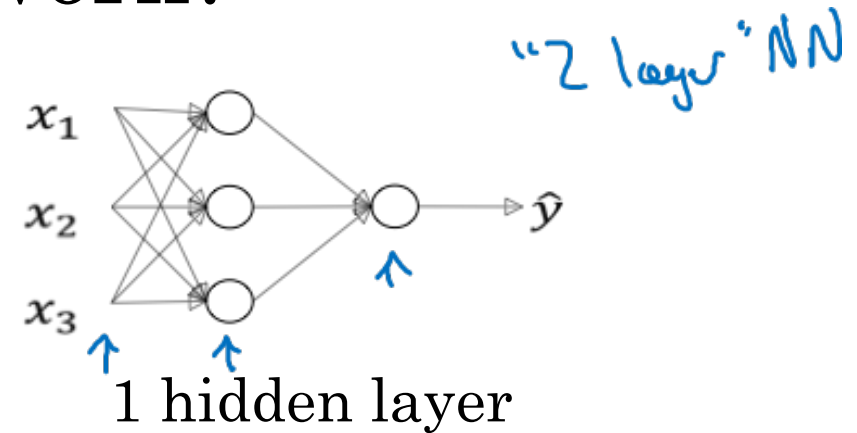
Deep Neural Networks

Deep L-layer
Neural network

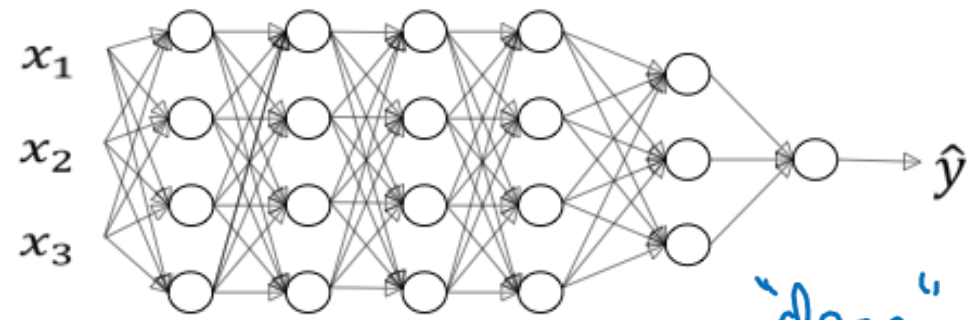
What is a deep neural network?



logistic regression



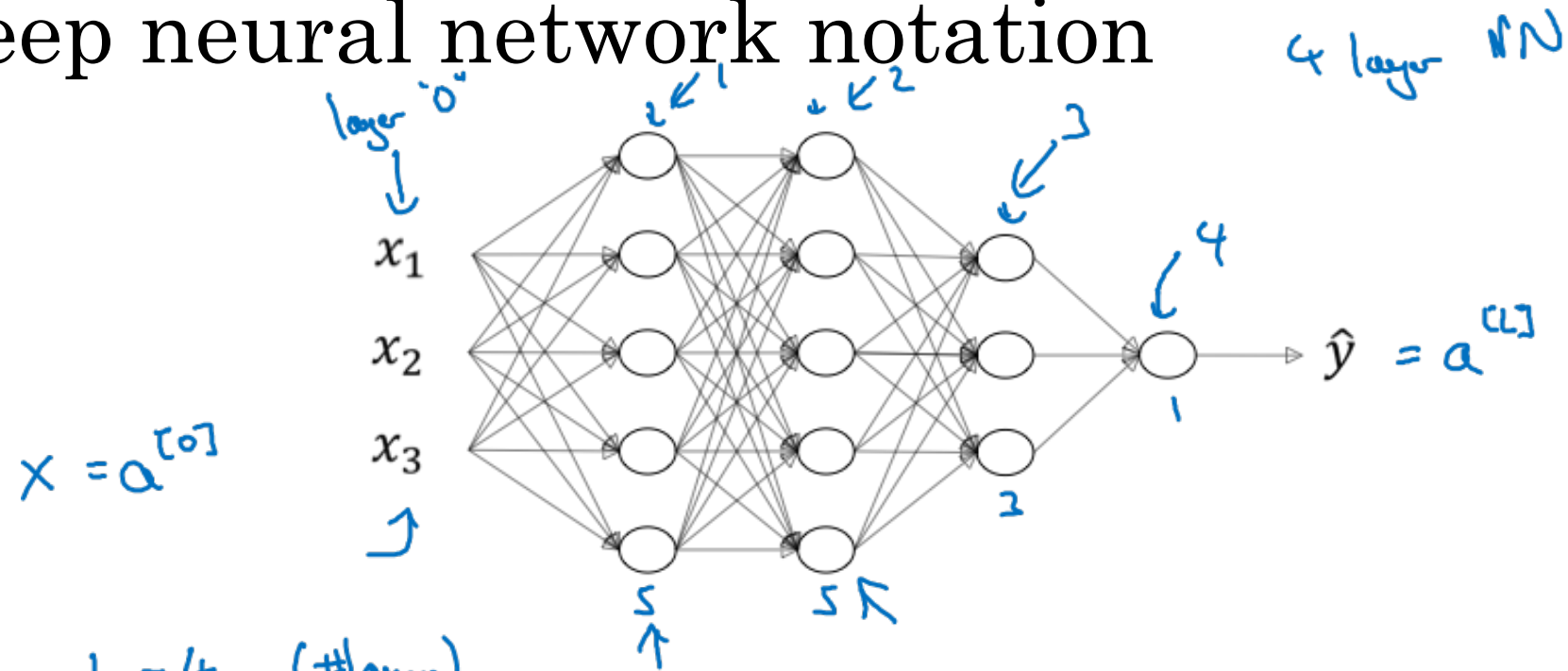
2 hidden layers



5 hidden layers

"deep"

Deep neural network notation



$L = 4$ (#layers)

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

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

$a^{[l]} = g(z^{[l]})$, $w_{ba}^{[l]} = \text{weights for } \underline{z^{[l]}}$

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

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

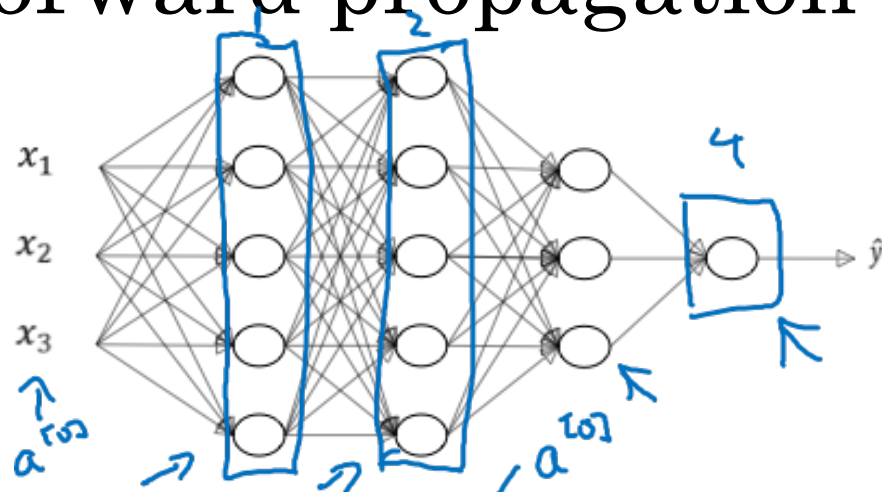


deeplearning.ai

Deep Neural Networks

Forward Propagation in a Deep Network

Forward propagation in a deep network



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

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

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

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

$$z^{[4]} = W^{[4]} a^{[3]} + b^{[4]}, a^{[4]} = g^{[4]}(z^{[4]}) = \hat{y}$$

$$\begin{aligned} z^{[l]} &= W^{[l]} A^{[l-1]} + b^{[l]} \\ A^{[l]} &= g^{[l]}(z^{[l]}) \end{aligned}$$

$A^{[0]} = X$

Vertical:

$$\begin{aligned} z^{[1]} &= W^{[1]} A^{[0]} + b^{[1]} \\ A^{[1]} &= g^{[1]}(z^{[1]}) \\ z^{[2]} &= W^{[2]} A^{[1]} + b^{[2]} \\ A^{[2]} &= g^{[2]}(z^{[2]}) \\ &\vdots \\ z^{[4]} &= W^{[4]} A^{[3]} + b^{[4]} \\ \hat{y} &= g^{[4]}(z^{[4]}) = A^{[4]} \end{aligned}$$

$\rightarrow X = A^{[0]}$
for $l=1 \dots 4$



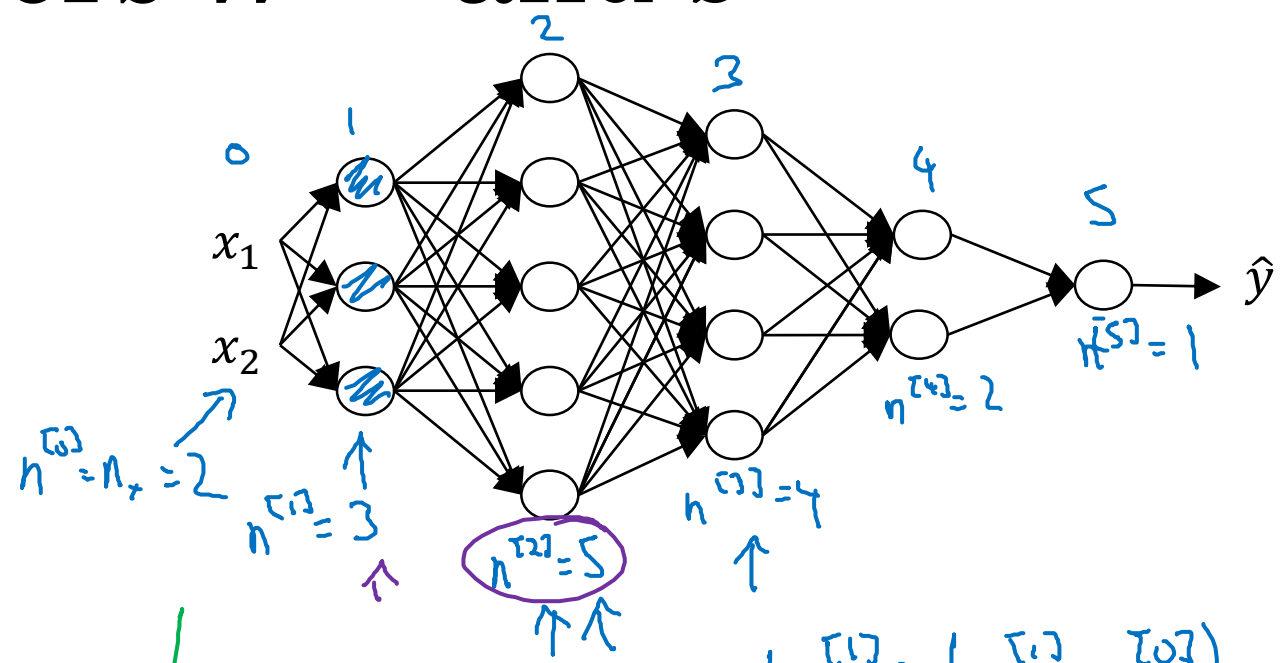
deeplearning.ai

Deep Neural Networks

Getting your matrix
dimensions right

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

\downarrow
 $z^{[L]} = g^{[L]}(a^{[L]})$
 \uparrow
 \downarrow
 $a^{[L]}$



$L=5$

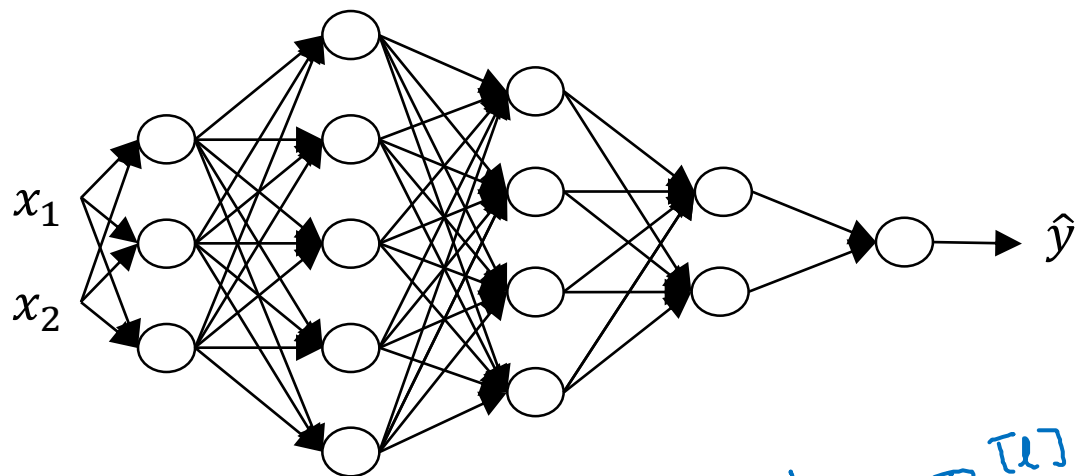
$\rightarrow W^{[L]}: (n^{[L]}, n^{[L-1]})$
 $\rightarrow b^{[L]}: (n^{[L]}, 1)$
 $\rightarrow \Delta W^{[L]}: (n^{[L]}, n^{[L-1]})$
 $\rightarrow \Delta b^{[L]}: (n^{[L]}, 1)$

\downarrow
 $z^{[1]} = \boxed{W^{[1]} \cdot x} + \boxed{b^{[1]}}$
 $(3,1) \leftarrow (3,2) \quad (2,1)$
 $(n^{[1]}, 1) \quad (n^{[1]}, n^{[0]}) \quad (n^{[0]}, 1)$
 $(3,1)$
 $(n^{[1]}, 1)$

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

$W^{[1]}: (n^{[1]}, n^{[0]})$
 $W^{[2]}: (5, 3) \quad (n^{[2]}, n^{[1]})$
 $z^{[2]} = \boxed{W^{[2]} \cdot a^{[1]}} + \boxed{b^{[2]}}$
 $\uparrow \quad \uparrow \quad \uparrow$
 $\rightarrow (5,1) \quad (5,3) \quad (3,1)$
 $(5,1)$
 $(n^{[2]}, 1)$
 $W^{[3]}: (4, 5)$
 $W^{[4]}: (2, 4)$, $W^{[5]}: (1, 2)$

Vectorized implementation



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

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

$$[z^{1} \ z^{[1](2)} \ \dots \ z^{[1](m)}]$$

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

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

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

$$z^{[2]}, A^{[2]} : (n^{[2]}, m)$$

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

$$dz^{[2]}, dA^{[2]} : (n^{[2]}, m)$$

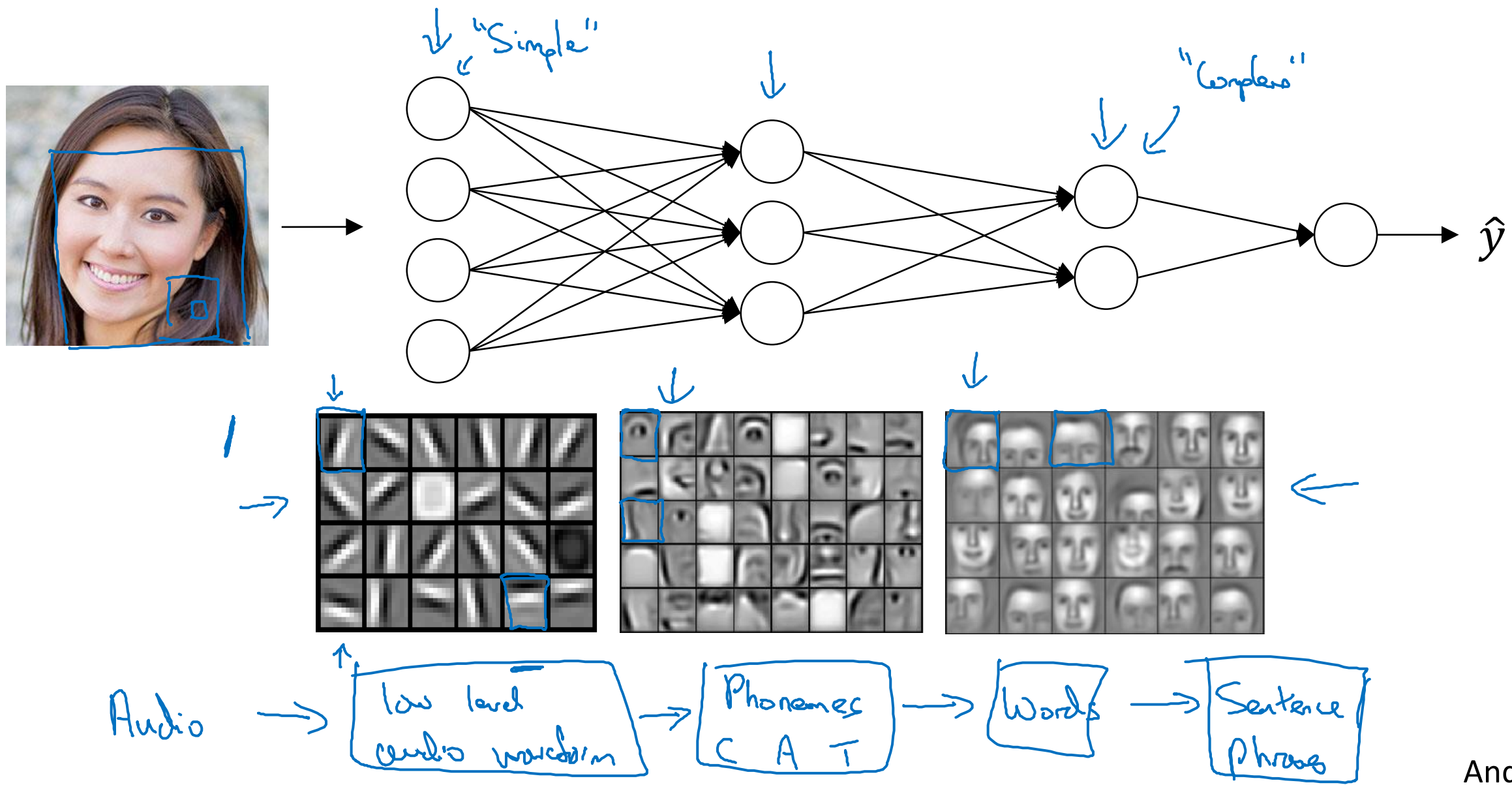


deeplearning.ai

Deep Neural Networks

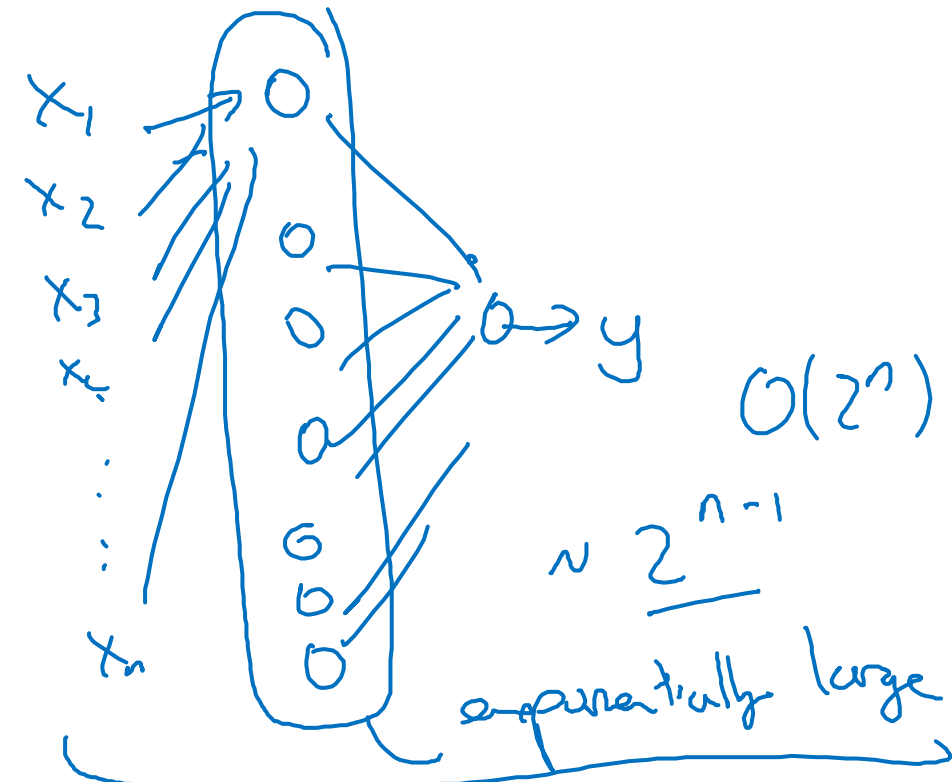
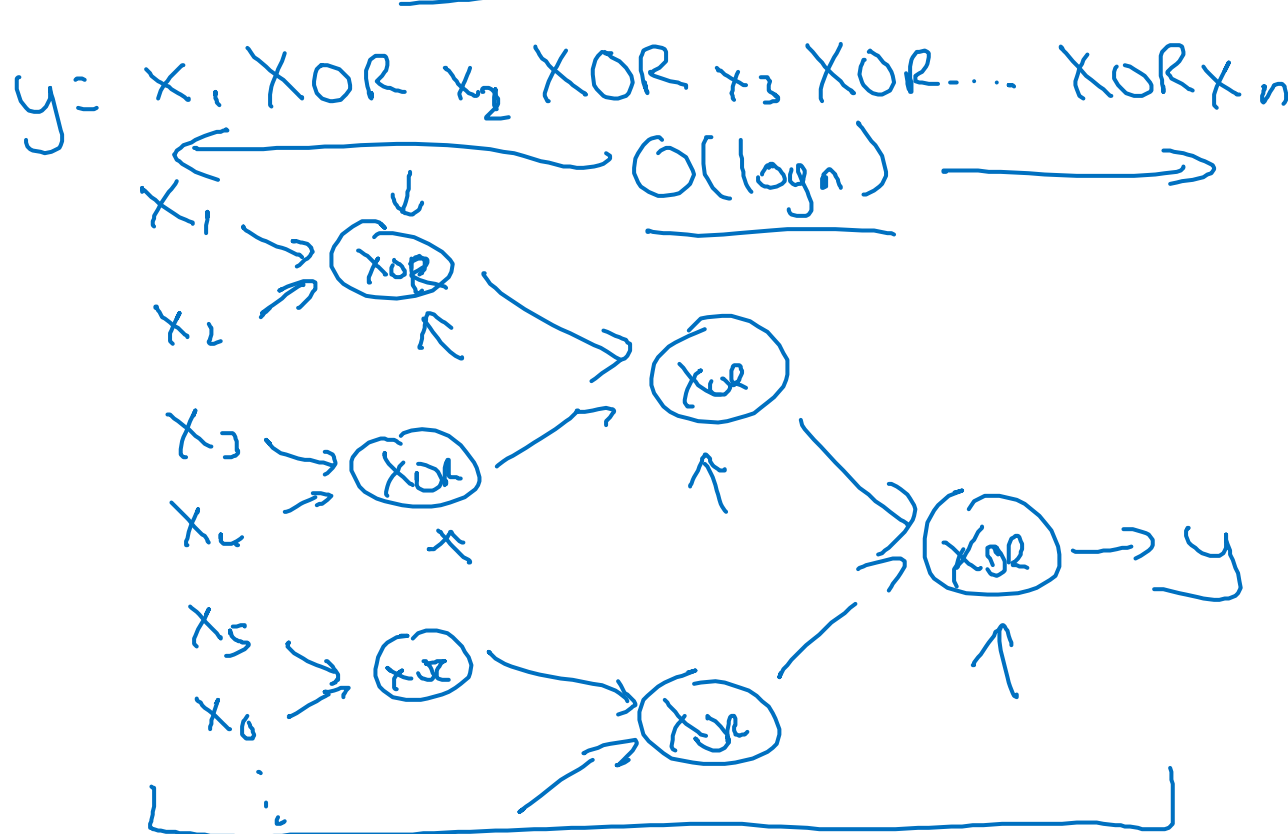
Why deep
representations?

Intuition about deep representation



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.



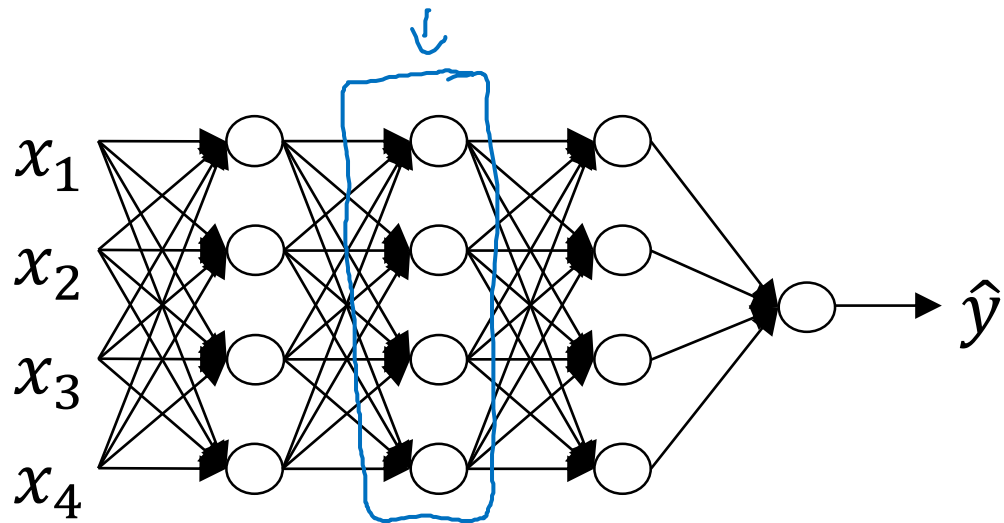


deeplearning.ai

Deep Neural Networks

Building blocks of
deep neural networks

Forward and backward functions



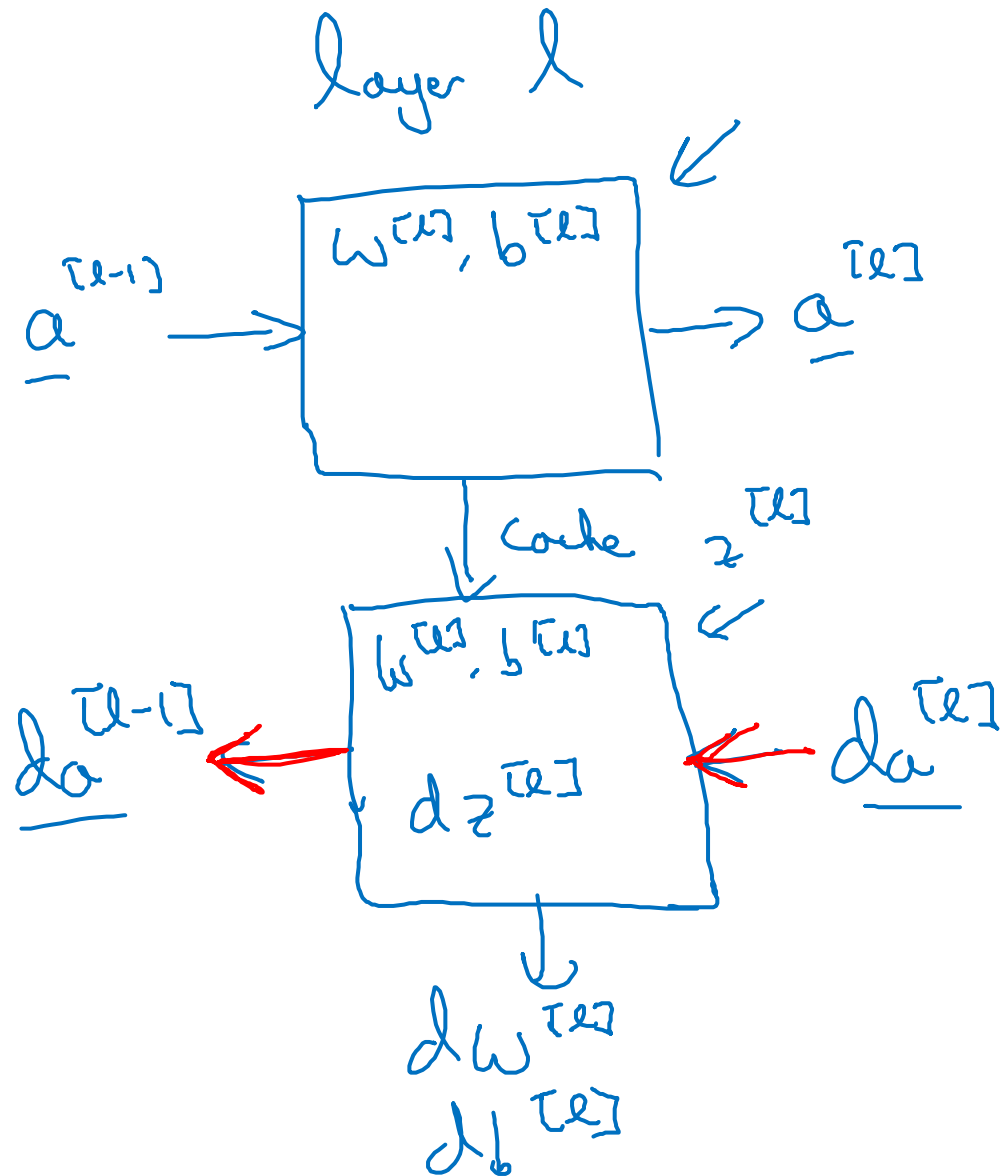
Layer l : $W^{[l]}, b^{[l]}$

→ Forward: Input $a^{[l-1]}$, output $a^{[l]}$

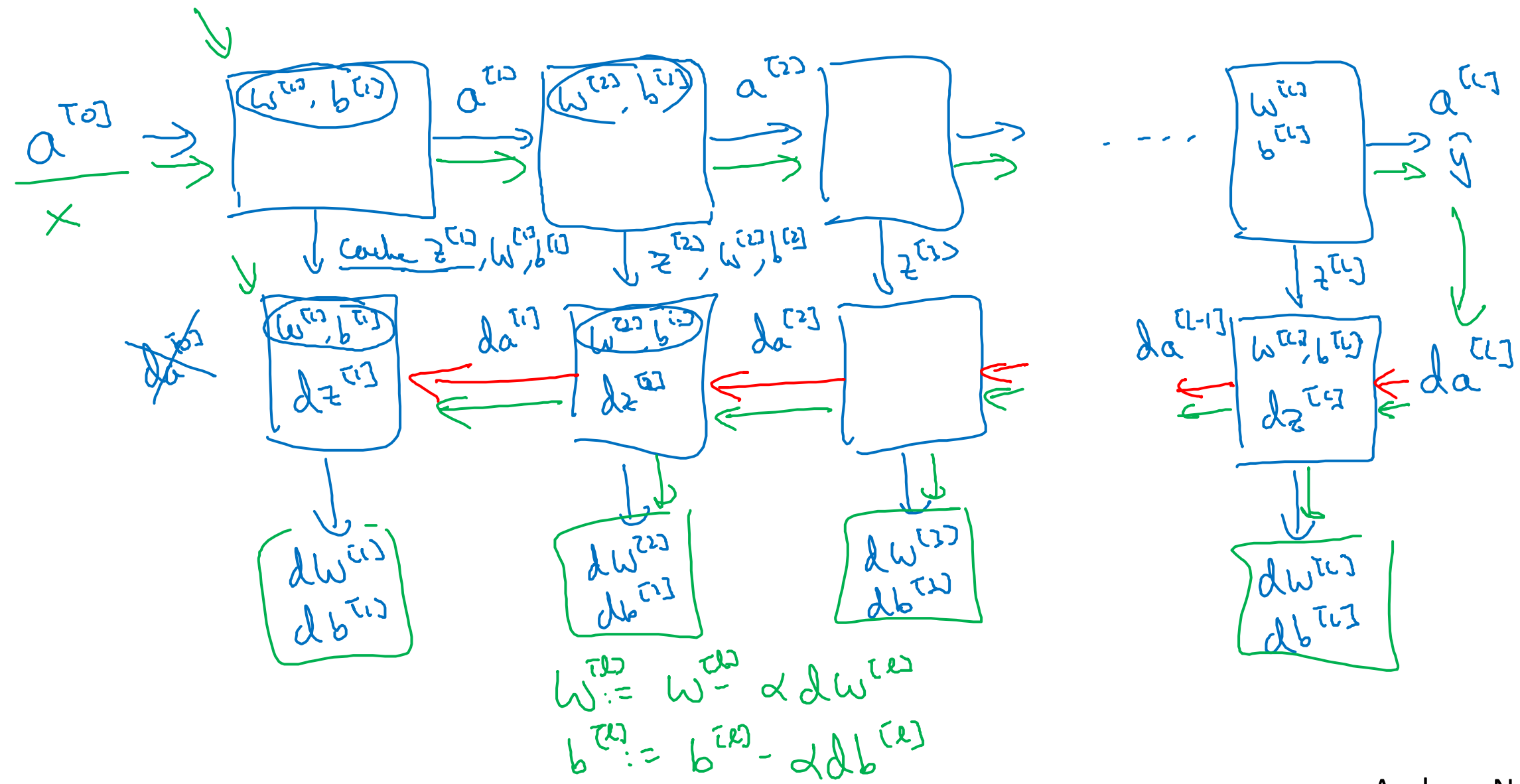
$$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$$

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

→ Backward: Input $da^{[l]}$, output $da^{[l-1]}$
 cache $z^{[l]}$
 $\frac{dw^{[l]}}{db^{[l]}}$



Forward and backward functions





deeplearning.ai

Deep Neural Networks

Forward and backward
propagation

Forward propagation for layer l

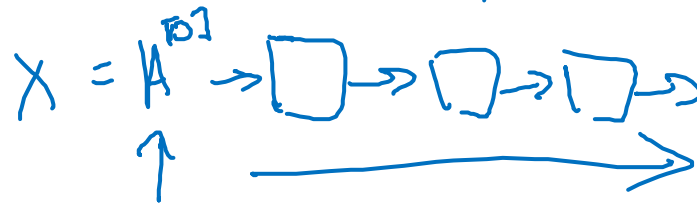
→ Input $a^{[l-1]}$ ←

→ Output $a^{[l]}$, cache ($z^{[l]}$)

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

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

$a^{[0]}$
 $A^{[0]}$



Vectorized:

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

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

Backward propagation for layer l

→ Input $da^{[l]}$

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

$$dz^{[l]} = da^{[l]} * g^{[l]'}(z^{[l]})$$

$$dW^{[l]} = dz^{[l]} \cdot a^{[l-1]}$$

$$db^{[l]} = dz^{[l]}$$

$$da^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$$

$$dz^{[l]} = W^{[l+1]T} dz^{[l+1]} * g^{[l]'}(z^{[l]})$$

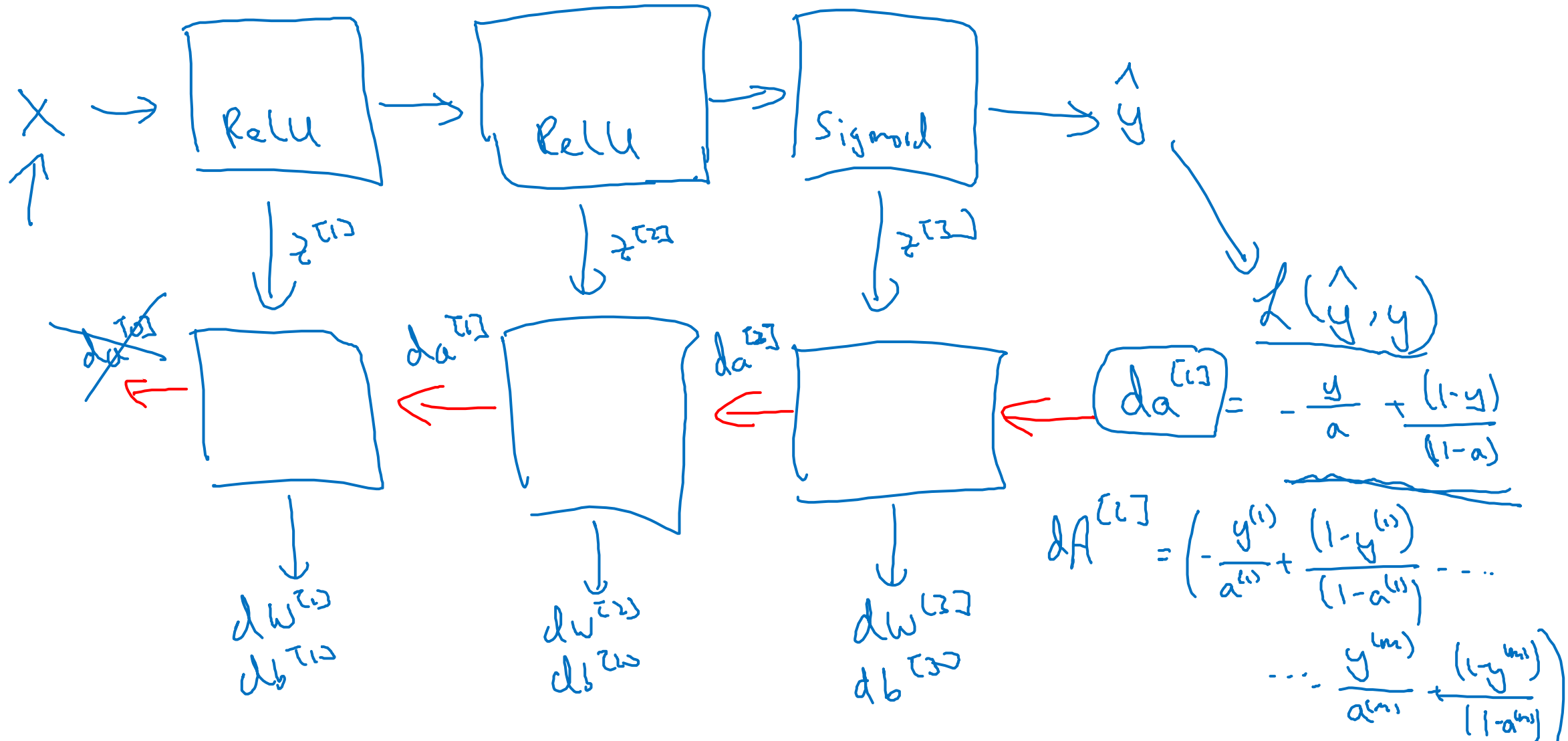
$$dz^{[l]} = dA^{[l]} * g^{[l]'}(z^{[l]})$$

$$dW^{[l]} = \frac{1}{n} dz^{[l]} \cdot A^{[l-1]T}$$

$$db^{[l]} = \frac{1}{n} \text{np.sum}(dz^{[l]}, \text{axis}=1, \text{keepdims}=\text{True})$$

$$dA^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$$

Summary





deeplearning.ai

Deep Neural Networks

Forward and backward
propagation

Forward propagation for layer l



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

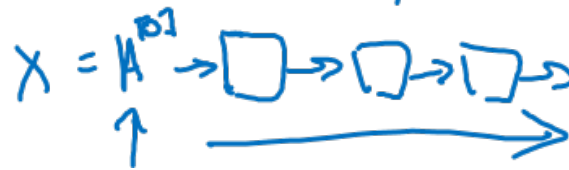
$$a^{[l]} = g^{[l]}(z^{[l]})$$

Vectorized:

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

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


$a^{[0]}$
 $A^{[0]}$



Backward propagation for layer l

→

→



$$\underline{dz^{[l]}} = \underline{da^{[l]}} * g^{[l]'}(z^{[l]})$$

$$\underline{dw^{[l]}} = \underline{dz^{[l]}} \cdot \underline{a^{[l-1]}}$$

$$\underline{db^{[l]}} = \underline{dz^{[l]}}$$

$$\underline{da^{[l-1]}} = \underline{w^{[l]T}} \cdot \underline{dz^{[l]}}$$

$$\underline{dz^{[l+1]}} = \underline{w^{[l+1]T}} \underline{dz^{[l+1]}} * g^{[l+1]'}(z^{[l+1]})$$

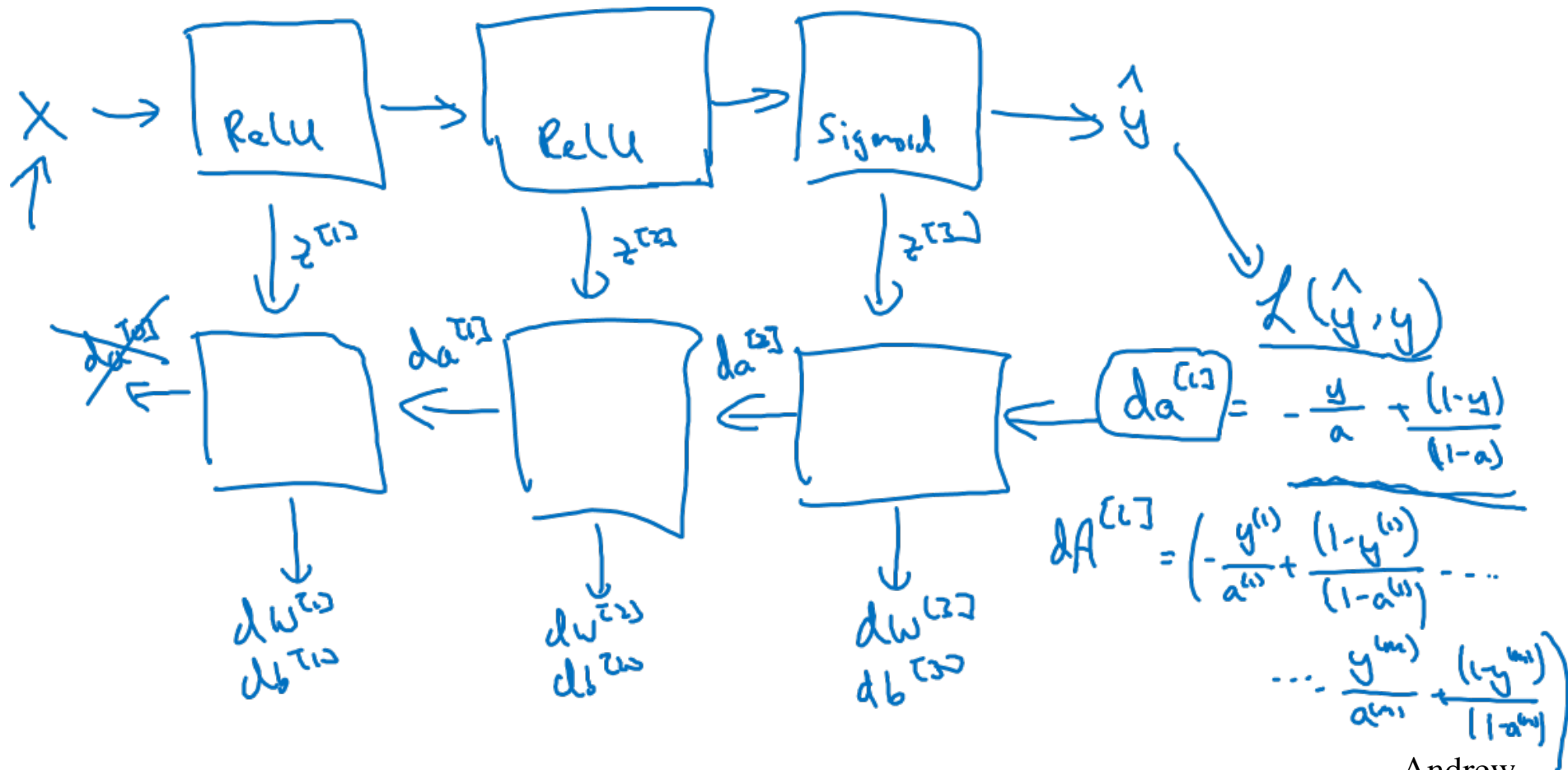
$$\underline{dz^{[l]}} = \underline{dA^{[l]}} * g^{[l]'}(z^{[l]})$$

$$\underline{dw^{[l]}} = \frac{1}{n} \underline{dz^{[l]}} \cdot A^{[l-1]T}$$

$$\underline{db^{[l]}} = \frac{1}{n} \text{np.sum}(\underline{dz^{[l]}}, \text{axis}=1, \text{keepdims}=\text{True})$$

$$\underline{dA^{[l-1]}} = \underline{w^{[l]T}} \cdot \underline{dz^{[l]}}$$

Summary





deeplearning.ai

Deep Neural Networks

Parameters vs Hyperparameters

What are hyperparameters?

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

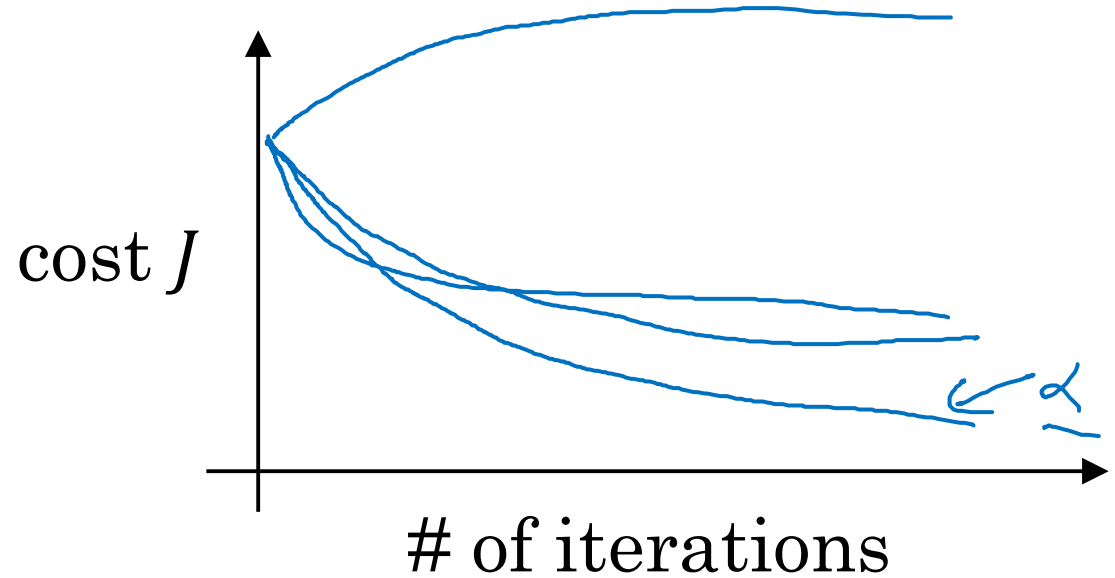
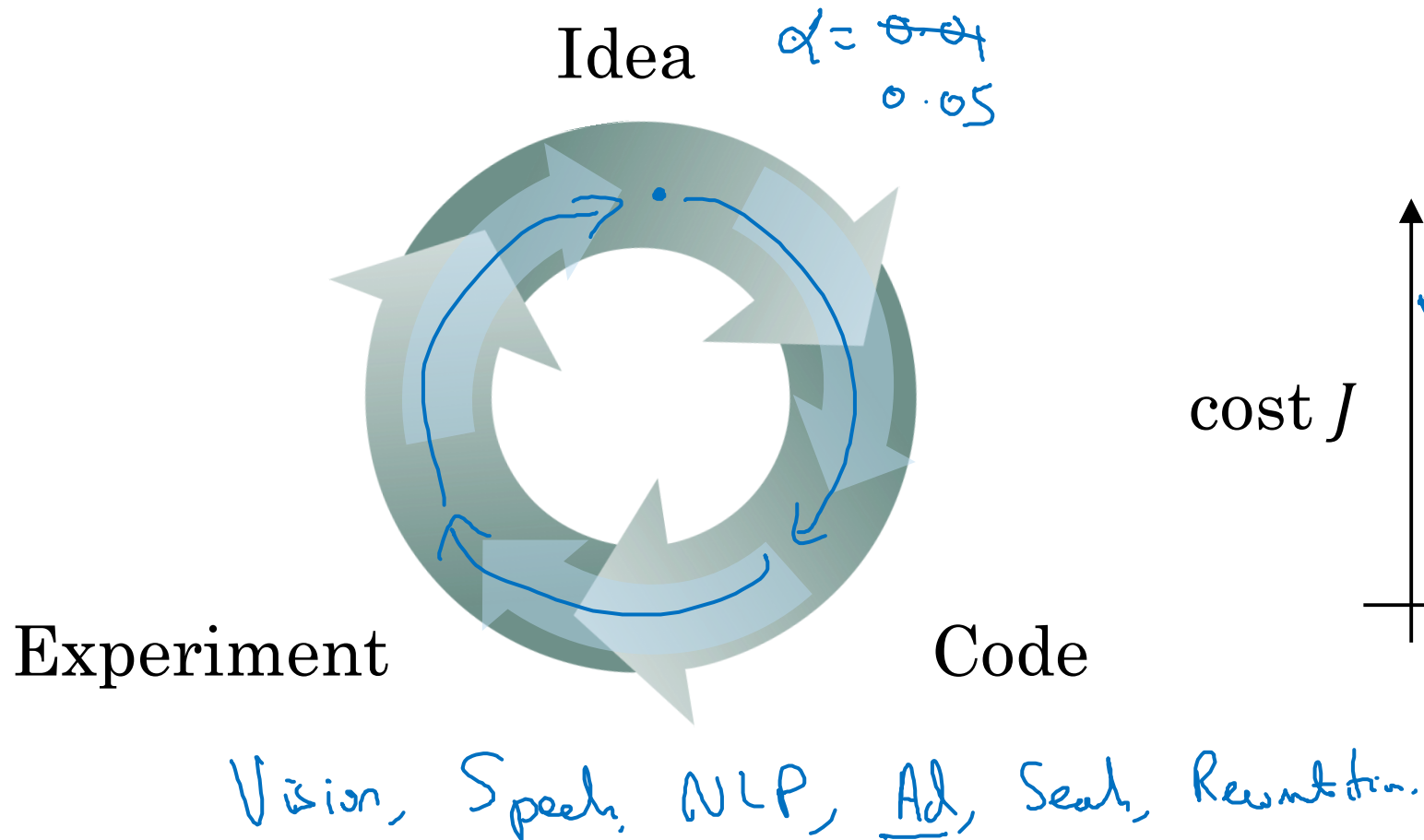


Hyperparameters:

- learning rate α
- $\frac{1}{\eta}$
- #iterations
- #hidden layers L
- # hidden units $n^{[1]}, n^{[2]}, \dots$
- choice of activation function

Later: Momentum, mini-batch size, regularizations, ...

Applied deep learning is a very empirical process





deeplearning.ai

Deep Neural Networks

What does this
have to do with
the brain?

Forward and backward propagation

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

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

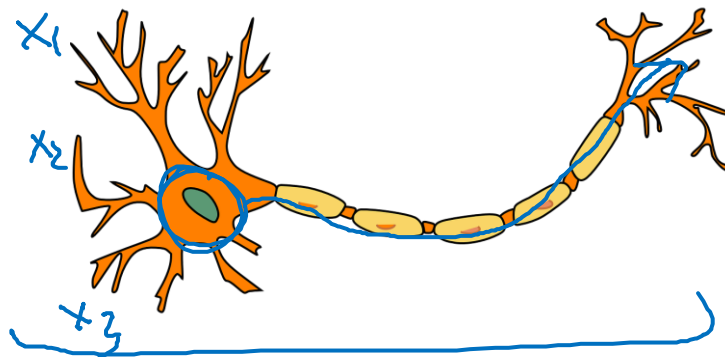
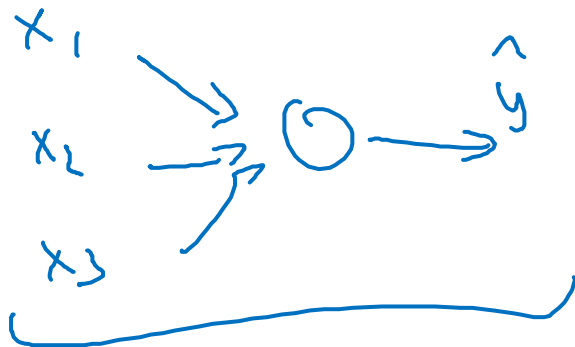
$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

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

\vdots

$$A^{[L]} = g^{[L]}(Z^{[L]}) = \hat{Y}$$

"It's like the brain"



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

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

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

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

$$\vdots$$

$$dZ^{[1]} = dW^{[L]T} dZ^{[2]} g'^{[1]}(Z^{[1]})$$

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

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