



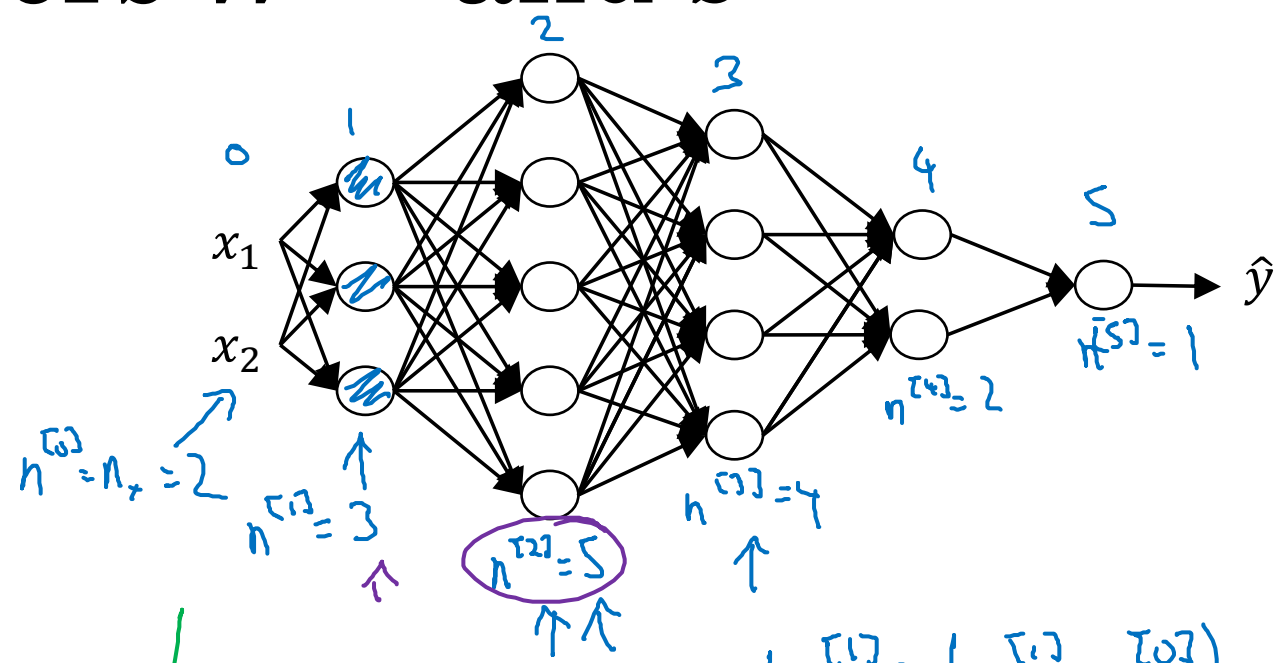
deeplearning.ai

Deep Neural Networks

Getting your matrix
dimensions right

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

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



$L=5$

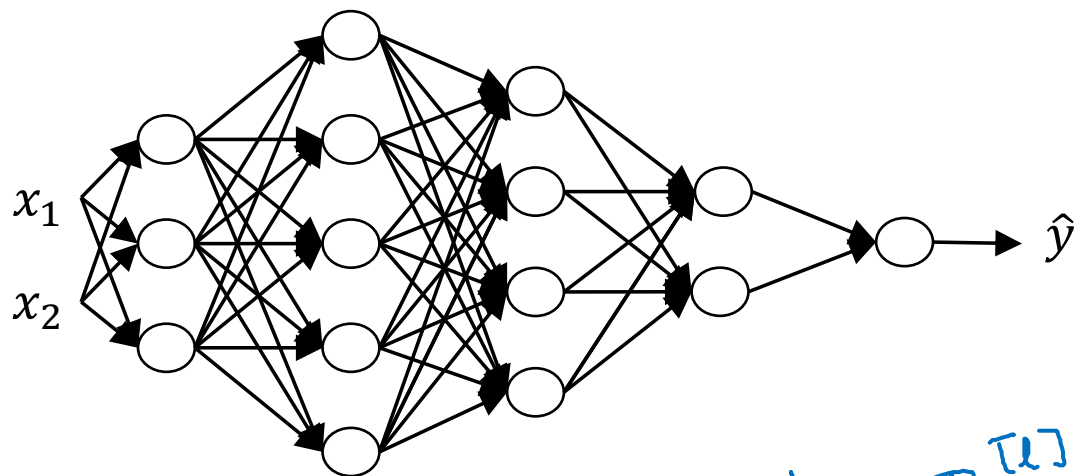
$\rightarrow W^{[L]}: (n^{[L]}, n^{[L-1]})$
 $\rightarrow b^{[L]}: (n^{[L]}, 1)$
 $\rightarrow \Delta W^{[L]}: (n^{[L]}, n^{[L-1]})$
 $\rightarrow \Delta b^{[L]}: (n^{[L]}, 1)$

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

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

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

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

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

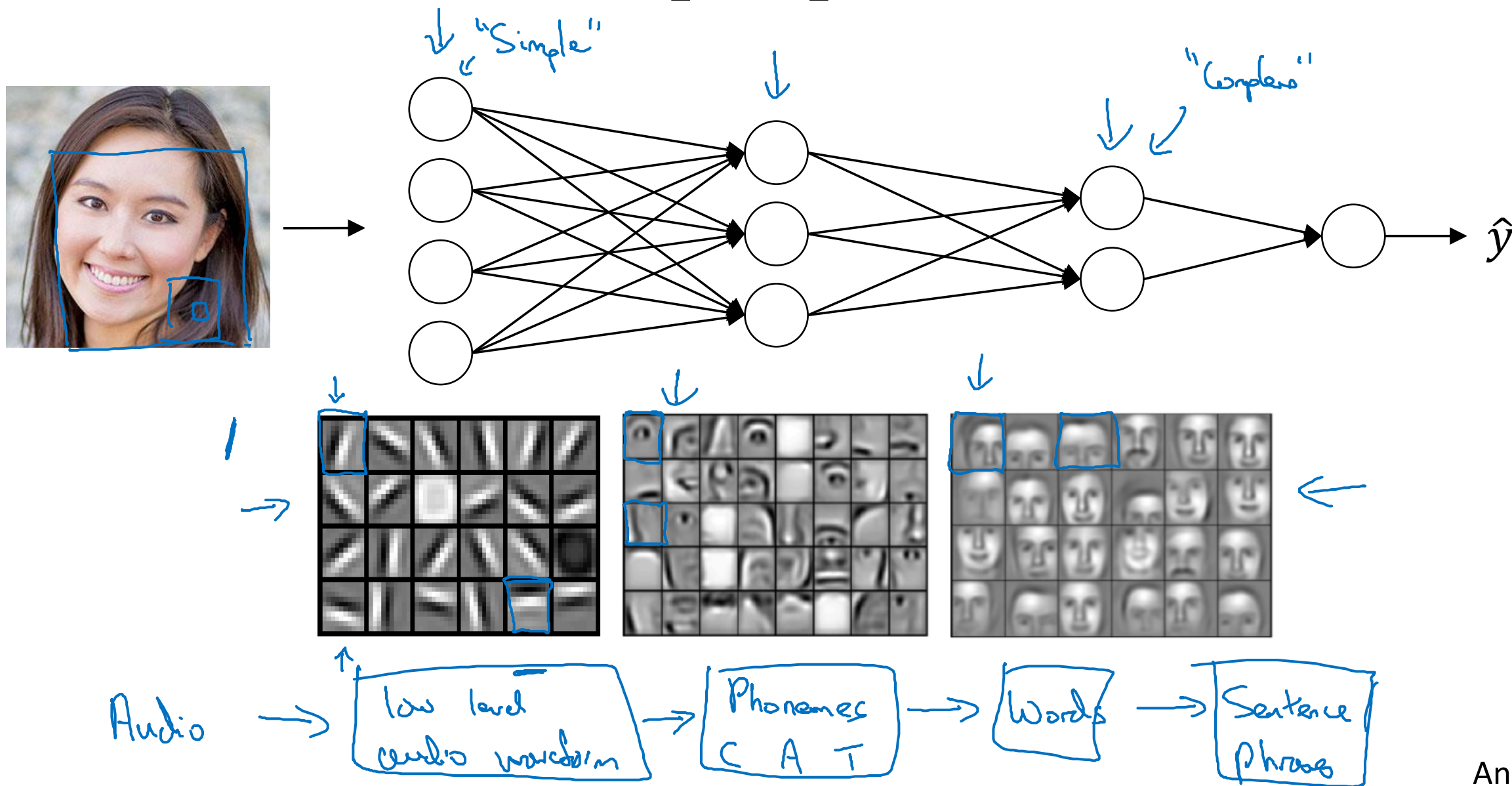


deeplearning.ai

Deep Neural Networks

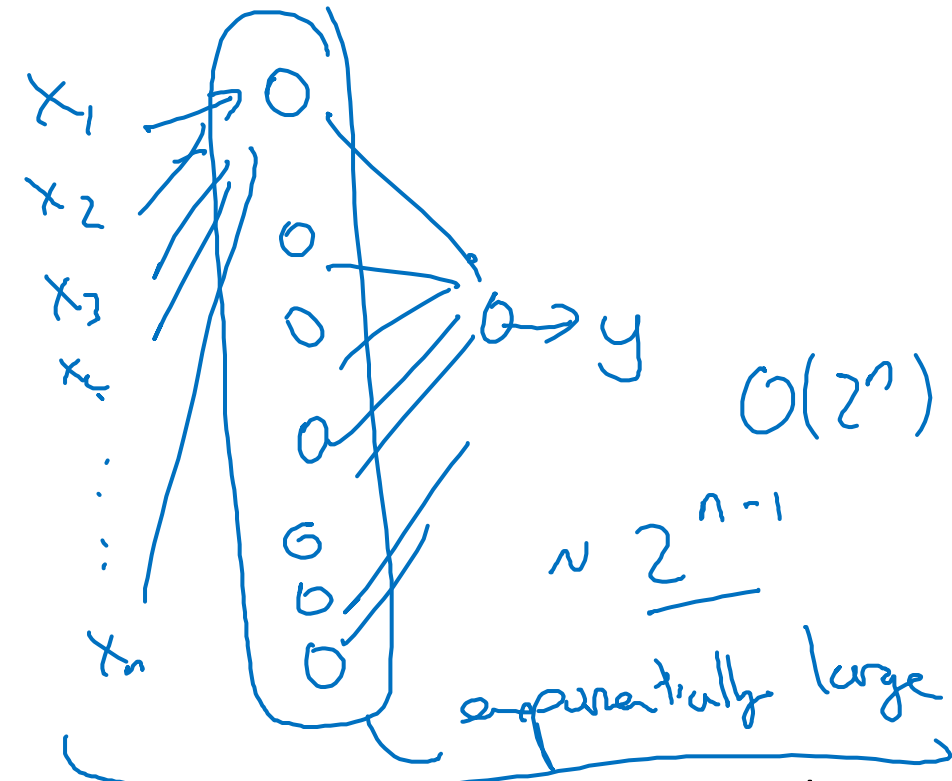
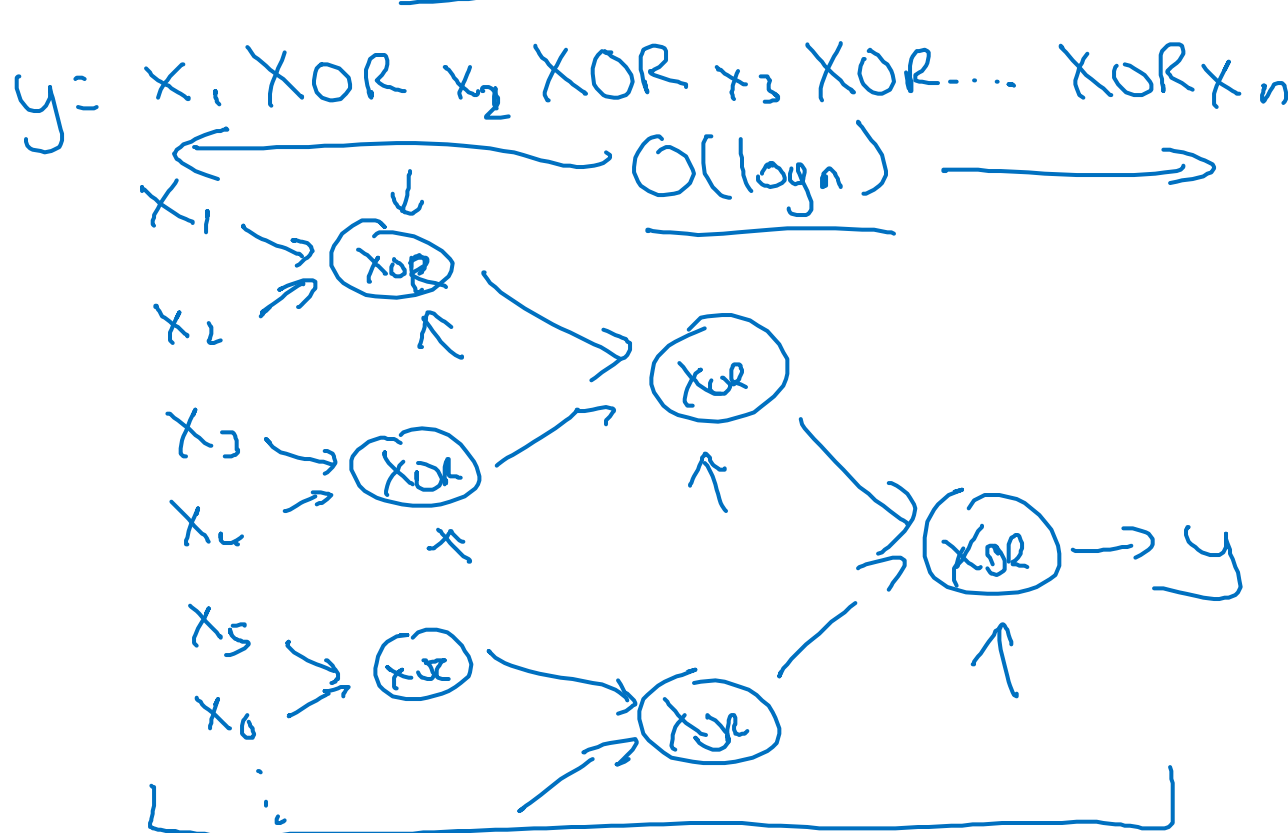
Why deep
representations?

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.



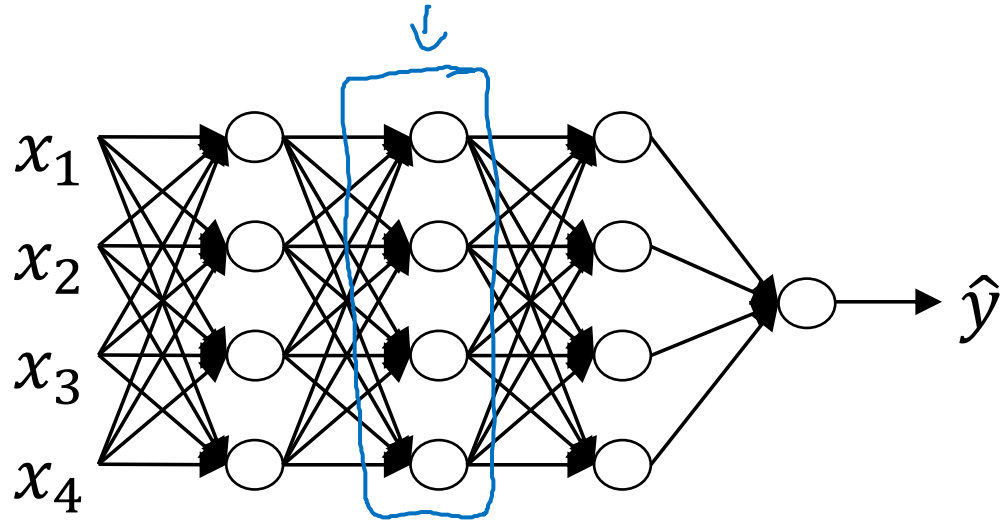


deeplearning.ai

Deep Neural Networks

Building blocks of
deep neural networks

Forward and backward functions



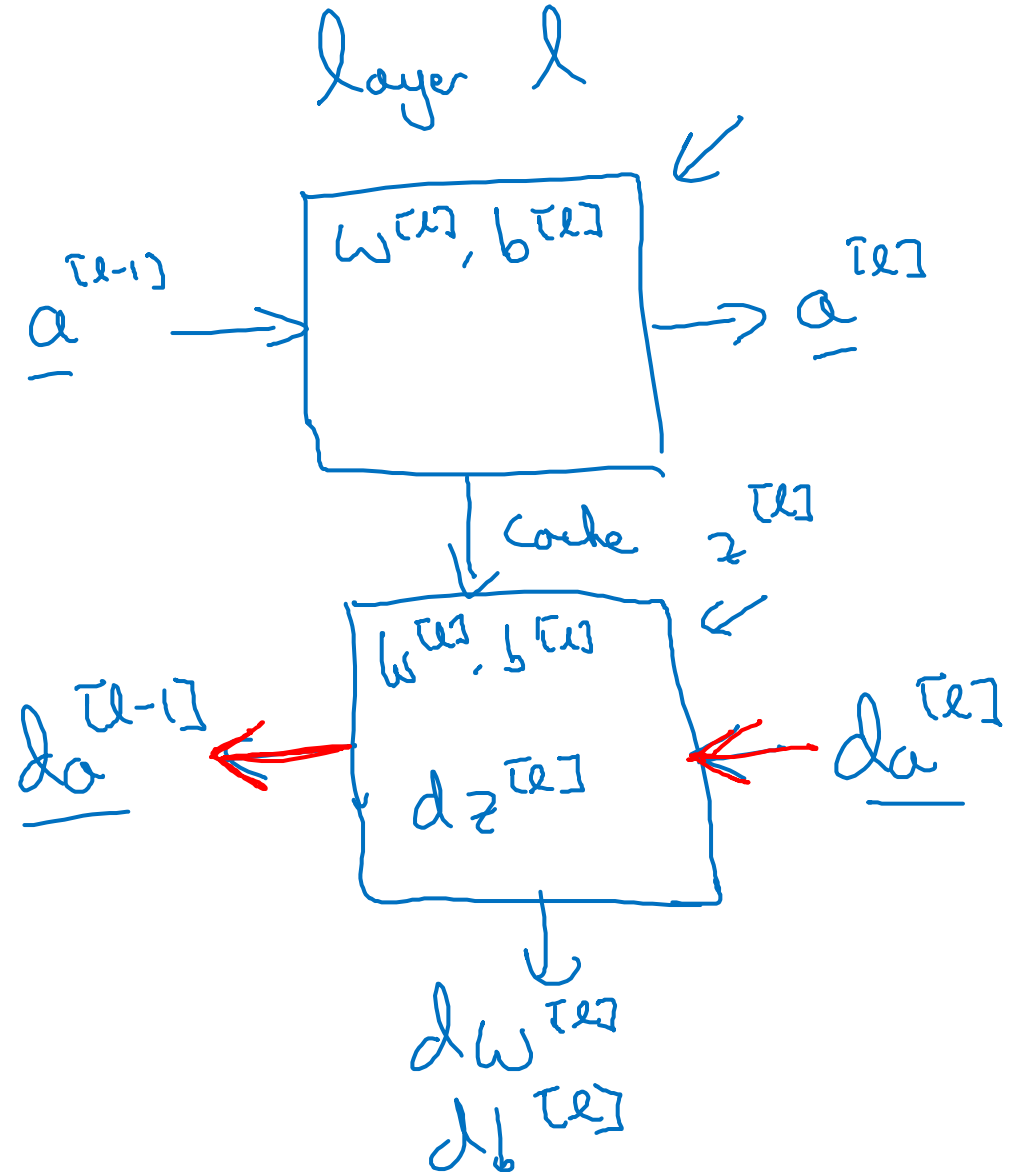
Layer l : $W^{[l]}, b^{[l]}$

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

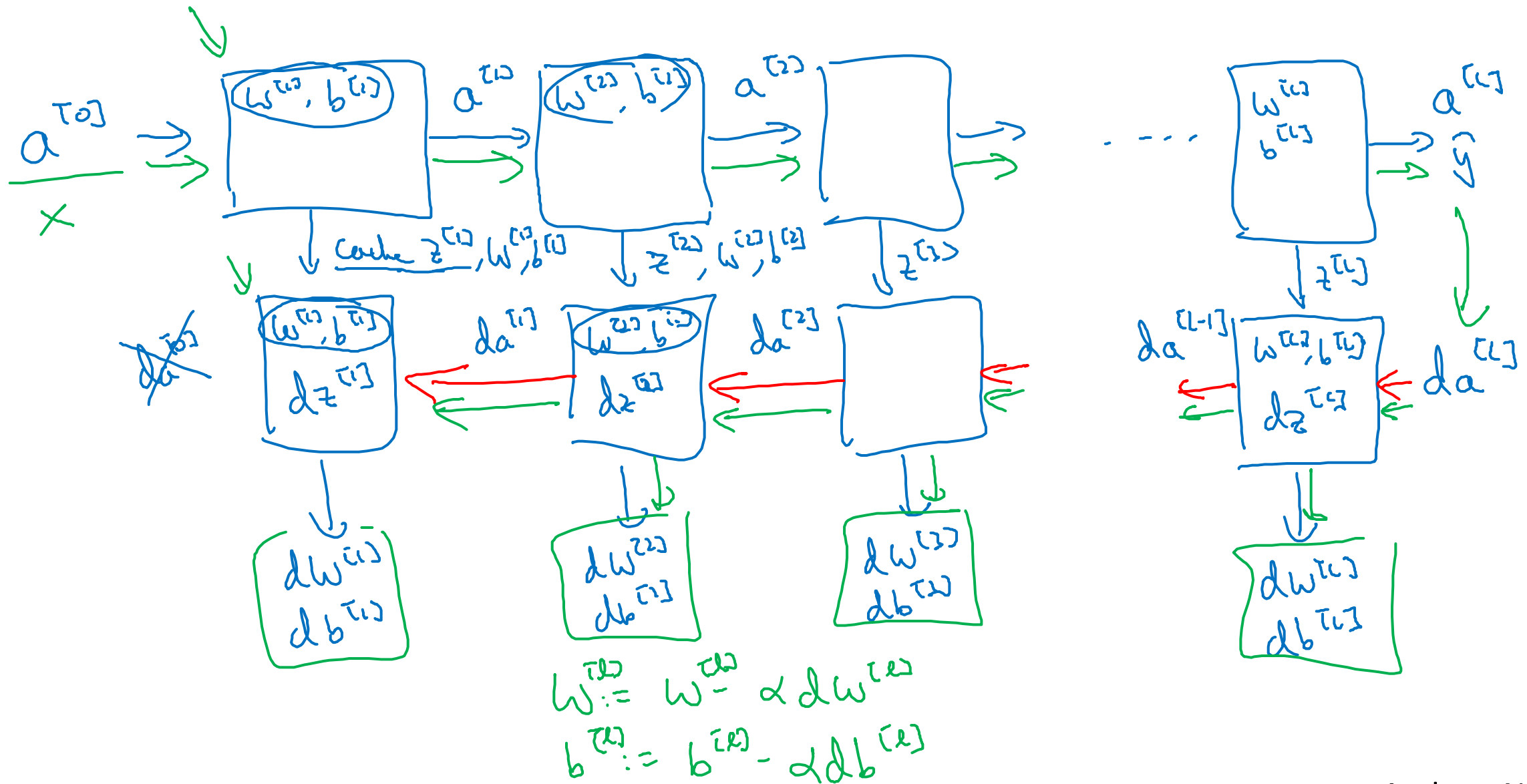
$$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]} \quad \text{cache } z^{[l]}$$

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

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



Forward and backward functions





deeplearning.ai

Deep Neural Networks

Forward and backward
propagation

Forward propagation for layer l

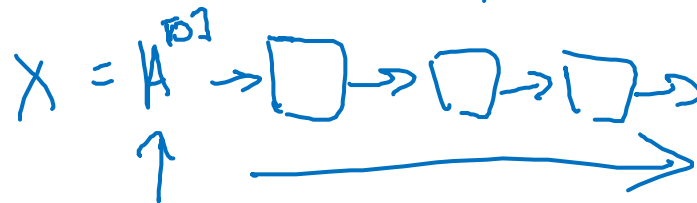
→ Input $a^{[l-1]}$ ←

→ Output $a^{[l]}$, cache ($z^{[l]}$)

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

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

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



Vectoriel:

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

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

Backward propagation for layer l

→ Input $da^{[l]}$

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

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

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

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

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

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

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

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

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

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

Summary

