Deep Neural Networks

Building blocks of deep neural networks
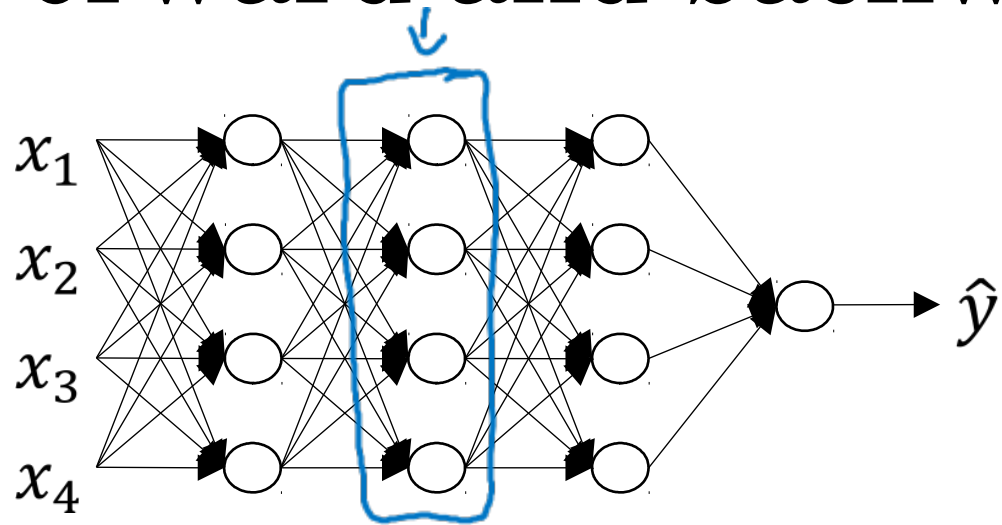
deeplearning.ai

Forward and backward functions

Layer ℓ: $W^{[\ell]}, b^{[\ell]}$

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

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

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

→ Backward: Input $da^{[\ell]}$, output $da^{[\ell-1]}$

cache $(z^{[\ell]})$

$\dfrac{dw^{[\ell]}}{db^{[\ell]}}$

layer ℓ

$a^{[\ell-1]} \rightarrow \boxed{W^{[\ell]}, b^{[\ell]}} \rightarrow a^{[\ell]}$

cache $z^{[\ell]}$

$da^{[\ell-1]} \leftarrow \boxed{\begin{array}{c} W^{[\ell]}, b^{[\ell]} \\ dz^{[\ell]} \end{array}} \leftarrow da^{[\ell]}$

$dw^{[\ell]}$

$db^{[\ell]}$

Andrew Ng

Forward and backward functions

$$W^{[\ell]} := W^{[\ell]} - \alpha \, dW^{[\ell]}$$
$$b^{[\ell]} := b^{[\ell]} - \alpha \, db^{[\ell]}$$
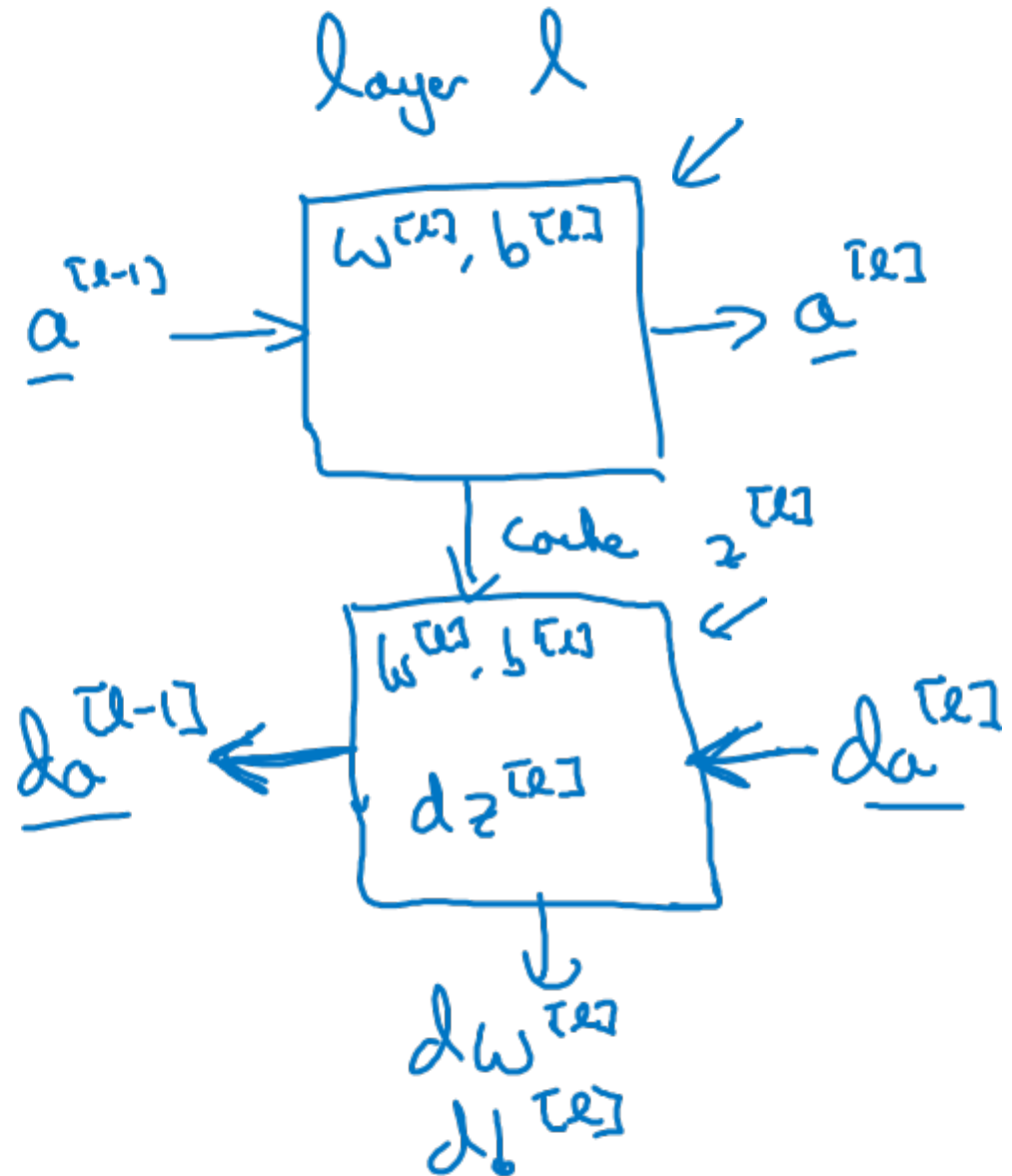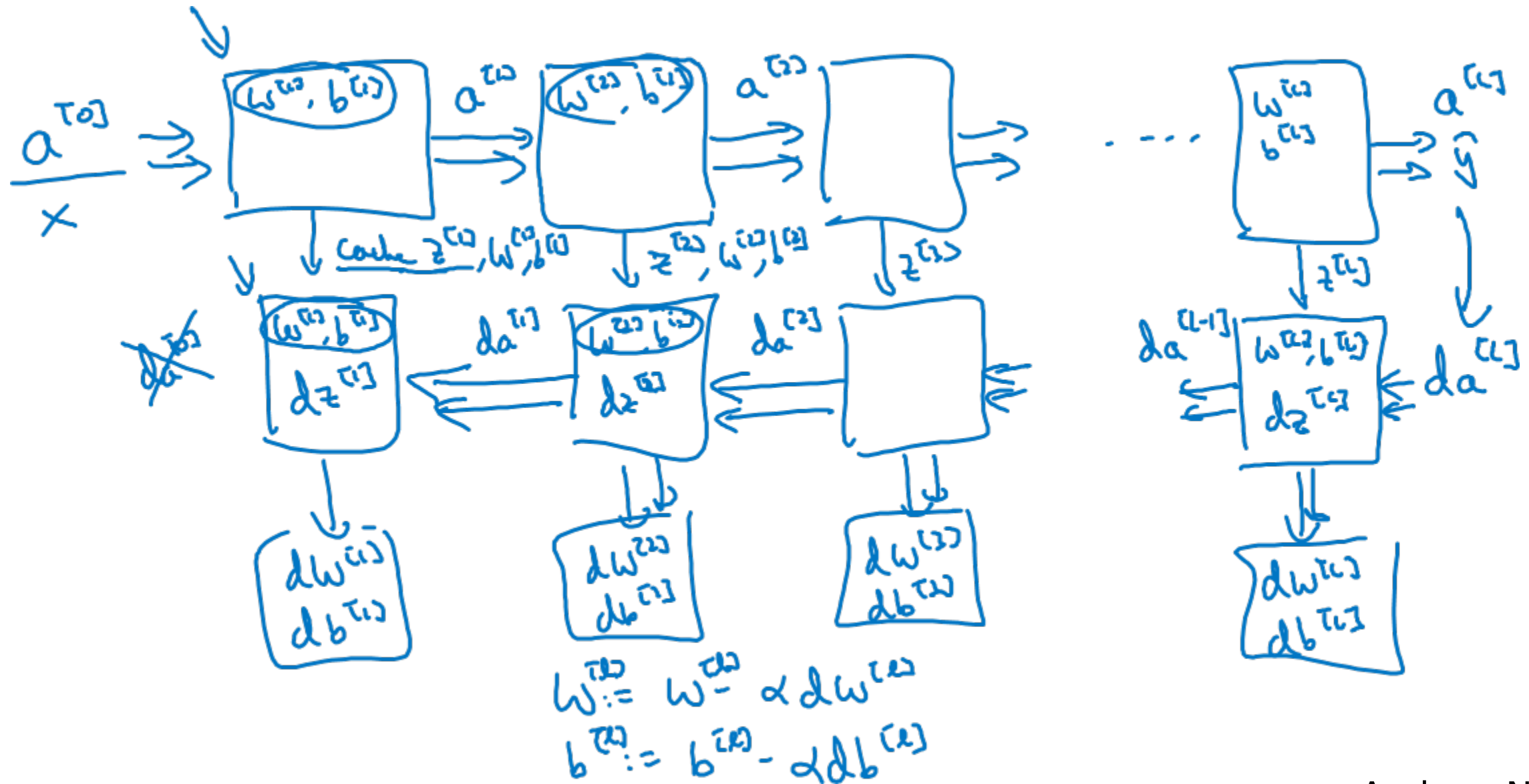
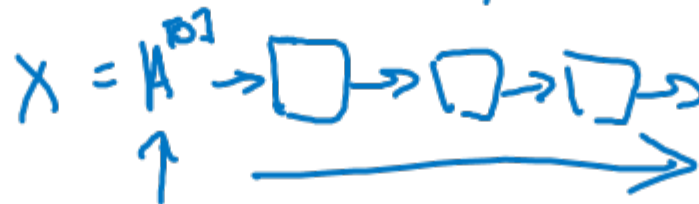# Forward propagation for layer l

→ Input $a^{[l-1]}$ ←

→ Output $a^{[l]}$, cache $(z^{[l]})$

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

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

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

$a^{[0]}$

$A^{[0]}$

Vectorial:

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

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

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

Backward propagation for layer l

→ Input: $da^{[l]}$

→ Output: $\boxed{da^{[l-1]}}$ ○, $dW^{[l]}, db^{[l]}$

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

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

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

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

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

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

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

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

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

Andrew Ng

Summary

$$da^{[L]} = -\frac{y}{a} + \frac{(1-y)}{(1-a)}$$

$$dA^{[L]} = \left(-\frac{y^{(1)}}{a^{(1)}} + \frac{(1-y^{(1)})}{(1-a^{(1)})} \cdots \cdots -\frac{y^{(m)}}{a^{(m)}} + \frac{(1-y^{(m)})}{(1-a^{(m)})}\right)$$

Andrew Ng