One hidden layer
Neural Network

Backpropagation
intuition (Optional)

deeplearning.ai

Computing gradients

Logistic regression

x

w ⟶ $z = w^T x + b$ ⟶ $a = \sigma(z)$ ⟶ $\mathcal{L}(a, y)$

b
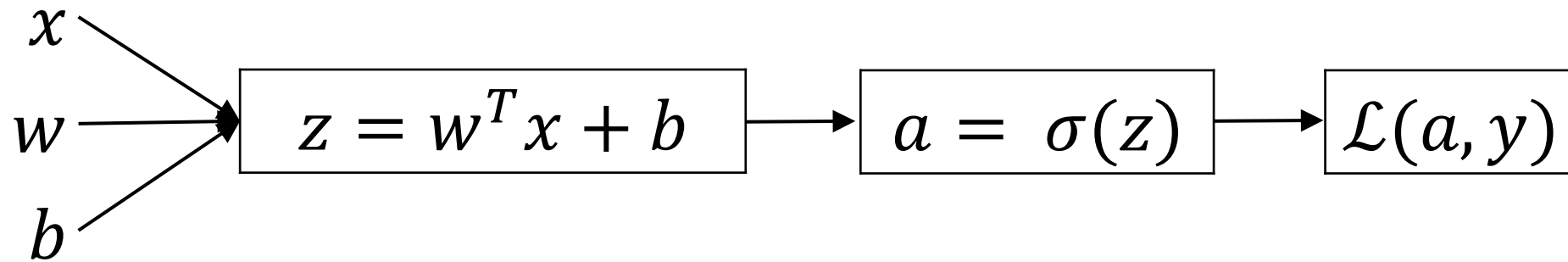
Neural network gradients

$$W^{[2]}$$

$$b^{[2]}$$

$$x$$

$$W^{[1]}$$

$$b^{[1]}$$

| $z^{[1]} = W^{[1]}x + b^{[1]}$ | $a^{[1]} = \sigma(z^{[1]})$ | $z^{[2]} = W^{[2]}x + b^{[2]}$ | $a^{[2]} = \sigma(z^{[2]})$ | $\mathcal{L}(a^{[2]}, y)$ |
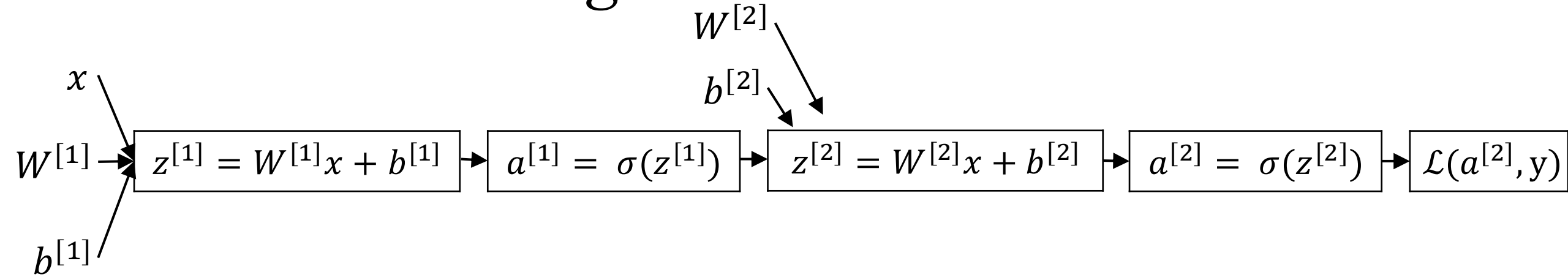
Summary of gradient descent

$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]}a^{[1]^T}$$

$$db^{[2]} = dz^{[2]}$$

$$dz^{[1]} = W^{[2]^T}dz^{[2]} * g^{[1]'}(\mathrm{z}^{[1]})$$

$$dW^{[1]} = dz^{[1]}x^T$$

$$db^{[1]} = dz^{[1]}$$

Summary of gradient descent

$$dz^{[2]} = a^{[2]} - y$$

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = dz^{[2]}a^{[1]^T}$$

$$dW^{[2]} = \frac{1}{m}dZ^{[2]}A^{[1]^T}$$

$$db^{[2]} = dz^{[2]}$$

$$db^{[2]} = \frac{1}{m}np.sum(dZ^{[2]}, axis = 1, keepdims = True)$$

$$dz^{[1]} = W^{[2]^T}dz^{[2]} * g^{[1]'}(z^{[1]})$$

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

$$dW^{[1]} = dz^{[1]}x^T$$

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

$$db^{[1]} = dz^{[1]}$$

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