



**deeplearning.ai**

# One hidden layer Neural Network

---

Gradient descent for  
neural networks

# Gradient descent for neural networks

Parameters:  $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}$   
 $(n^{[1]}, n^{[2]})$   $(n^{[2]}, 1)$

$$n_x = n^{[0]}, \quad n^{[1]}, \quad \underline{n^{[2]} = 1}$$

$$\text{Cost function: } J(W^{[1]}, b^{[1]}, \underset{\uparrow}{W^{[2]}}, \underset{\uparrow}{b^{[2]}}) = \frac{1}{m} \sum_{i=1}^n \ell(\underset{\uparrow a^{[2]}}{\hat{y}}, y)$$

Gradient descent:

→ Repeat {

→ Compute predictions  $(\hat{y}^{(i)}, i=1, \dots, m)$

$$\frac{dW^{[1]}}{dW^{[1]}} = \frac{dJ}{dW^{[1]}}, \quad \frac{db^{[1]}}{db^{[1]}} = \frac{dJ}{db^{[1]}}, \dots$$

$$W^{[1]} := W^{[1]} - \alpha \frac{dW^{[1]}}{dW^{[1]}}$$

$$b^{[1]} := b^{[1]} - \alpha \frac{db^{[1]}}{db^{[1]}}$$

$$W^{[2]} := \dots \quad b^{[2]} := \dots$$

# Formulas for computing derivatives

Forward propagation:

$$z^{[1]} = w^{[1]}x + b^{[1]}$$

$$A^{[1]} = g^{[1]}(z^{[1]}) \leftarrow$$

$$z^{[2]} = w^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = g^{[2]}(z^{[2]}) = \sigma(z^{[2]})$$

Back propagation:

$$dz^{[2]} = A^{[2]} - Y \leftarrow$$

$$dw^{[2]} = \frac{1}{n} dz^{[2]} A^{[1]T}$$

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

$$dz^{[1]} = \underbrace{w^{[2]T} dz^{[2]}}_{(n^{[1]}, m)} \times \underbrace{g^{[1]'}(z^{[1]})}_{\text{element-wise product}} \underbrace{(n^{[1]}, m)}_{(n^{[1]}, m)}$$

$$dw^{[1]} = \frac{1}{n} dz^{[1]} x^T$$

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

reshape  $\uparrow$

$$Y = [y^{(1)} y^{(2)} \dots, y^{(n)}]$$

$$(n^{[2]}) \leftarrow$$

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