deeplearning.ai

One hidden layer Neural Network

Gradient descent for neural networks

Gradient descent for neural networks

Parameters: $W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}$
$(n^{[1]}, n^{[0]}) \quad (n^{[1]}, 1) \quad (n^{[2]}, n^{[1]}) \quad (n^{[1]}, 1)$

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

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

Gradient Descent:

$\rightarrow$ Repeat $\{$

$\quad \rightarrow$ Compute predicts $(\hat{y}^{(i)}, \; i=1,\ldots,m)$

$\quad \underline{dW^{[1]}} = \frac{dJ}{dW^{[1]}}, \quad \underline{db^{[1]}} = \frac{dJ}{db^{[1]}}, \; \ldots$

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

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

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

$\}$

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:

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

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

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

$$dZ^{[1]} = \underbrace{W^{[2]T} dZ^{[2]}}_{(n^{[1]}, m)} * \underbrace{g^{[1]'}(Z^{[1]})}_{element-wise \ product} \quad (n^{[1]}, m)$$

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

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

$$Y = [y^{(1)} \ y^{(2)} \cdots, y^{(m)}]$$

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

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

Andrew Ng