

deeplearning.ai

## Basics of Neural Network Programming

Vectorizing Logistic Regression

## Vectorizing Logistic Regression

$$Z^{(1)} = w^{T}x^{(1)} + b$$

$$Z^{(2)} = w^{T}x^{(2)} + b$$

$$Z^{(3)} = w^{T}x^{(3)} + b$$

$$Z^{(3)} = \sigma(z^{(3)})$$

$$Z^$$



deeplearning.ai

## Basics of Neural Network Programming

Vectorizing Logistic Regression's Gradient Computation

## Vectorizing Logistic Regression

$$\frac{dz^{(1)} = a^{(1)} - y^{(1)}}{dz^{(2)}} = \frac{dz^{(2)} = a^{(2)} - y^{(2)}}{dz^{(2)}} = \frac{dz^{(2)} - z^{($$

$$db = \frac{1}{m} \sum_{i=1}^{m} dz^{(i)}$$

$$= \frac{1}{m} \left[ x^{(i)} + x^{(i)} dz^{(i)} \right]$$

Implementing Logistic Regression

J = 0, 
$$dw_1 = 0$$
,  $dw_2 = 0$ ,  $db = 0$ 

for i = 1 to m:

$$z^{(i)} = w^T x^{(i)} + b$$

$$a^{(i)} = \sigma(z^{(i)})$$

$$J += -[y^{(i)} \log a^{(i)} + (1 - y^{(i)}) \log(1 - a^{(i)})]$$

$$dz^{(i)} = a^{(i)} - y^{(i)}$$

$$dw_1 += x_1^{(i)} dz^{(i)}$$

$$dw_2 += x_2^{(i)} dz^{(i)}$$

$$db += dz^{(i)}$$

$$J = J/m, dw_1 = dw_1/m, dw_2 = dw_2/m$$

$$db = db/m$$

iter in range (1000): 
$$C$$

$$Z = \omega^{T} X + b$$

$$= n p \cdot dot (\omega \cdot T \cdot X) + b$$

$$A = G(Z)$$

$$A$$