

## Basics of Neural Network Programming

Vectorizing Logistic Regression

deeplearning.ai

## Vectorizing Logistic Regression

$$\frac{z^{(1)} = w^T x^{(1)} + b}{a^{(1)}} = \sigma(z^{(1)})$$

$$\frac{z^{(2)} = w^T x^{(2)} + b}{a^{(2)}} = \sigma(z^{(2)})$$

$$\frac{z^{(3)} = w^T x^{(3)} + b}{a^{(3)}} = \sigma(z^{(3)})$$

$$\frac{z^{(3)}$$



deeplearning.ai

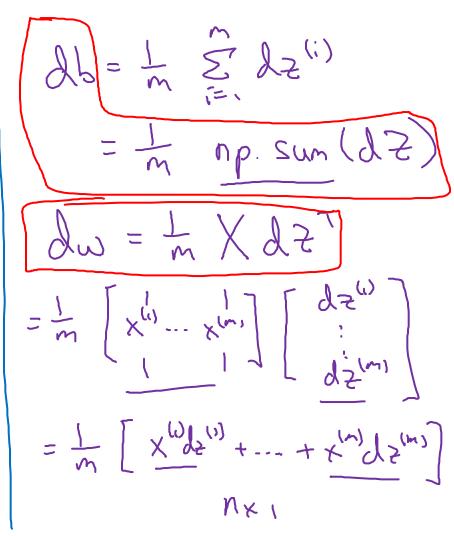
## Basics of Neural Network Programming

Vectorizing Logistic Regression's Gradient Computation

## Vectorizing Logistic Regression

$$d_{\xi}^{(i)} = a^{(i)} - y^{(i)}$$

$$d_{\xi$$



Implementing Logistic Regression

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

for  $i = 1$  to  $m$ :

 $z^{(i)} = w^T x^{(i)} + b = 0$ 
 $a^{(i)} = \sigma(z^{(i)}) = 0$ 
 $J + = -[y^{(i)} \log a^{(i)} + (1 - y^{(i)}) \log(1 - a^{(i)})]$ 
 $dz^{(i)} = a^{(i)} - y^{(i)} = 0$ 
 $dw_1 + = x_1^{(i)} z^{(i)} = 0$ 
 $dw_2 + = x_2^{(i)} z^{(i)} = 0$ 
 $dw_1 + = dz^{(i)}$ 
 $dw_2 + = dz^{(i)}$ 
 $dw_1 + = dz^{(i)}$ 
 $dw_2 + = dz^{(i)}$ 
 $dw_1 + = dz^{(i)}$ 
 $dw_2 + = dz^{(i)}$ 
 $dw_1 + = dz^{(i)}$