



deeplearning.ai

# Basics of Neural Network Programming

---

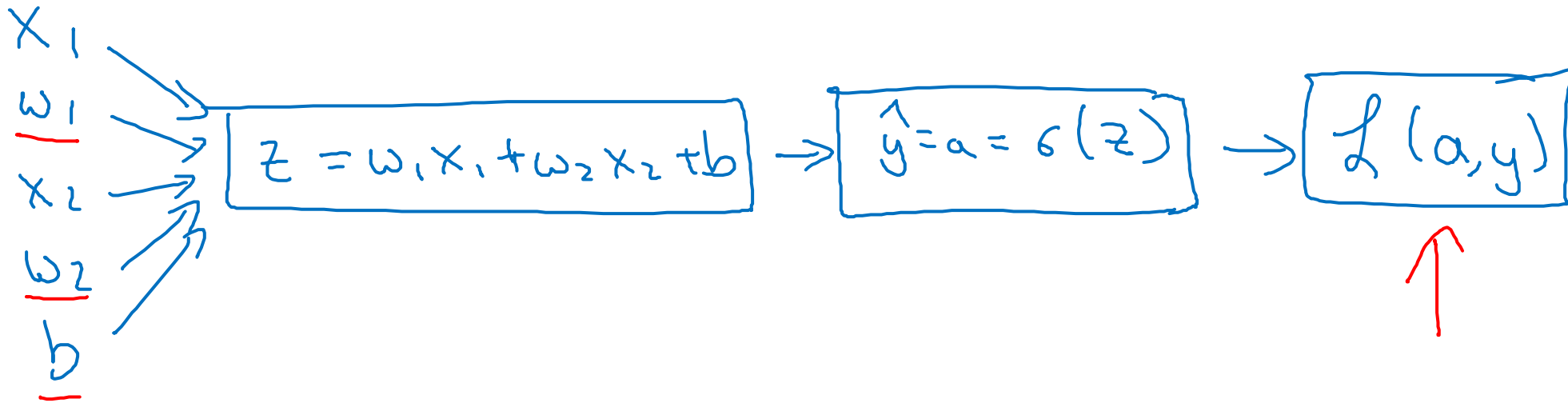
Logistic Regression  
Gradient descent

# Logistic regression recap

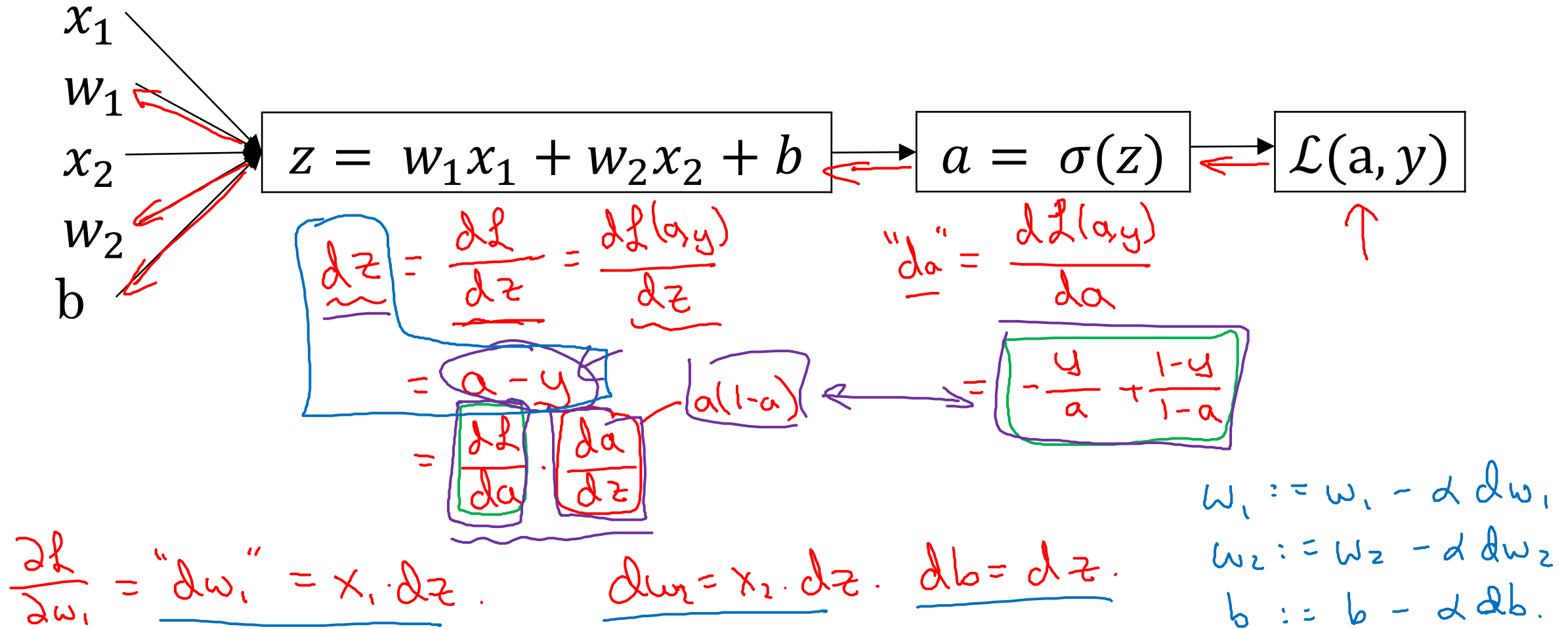
→  $z = w^T x + b$

→  $\hat{y} = a = \sigma(\underline{z})$

→  $\mathcal{L}(a, y) = -(y \log(a) + (1 - y) \log(1 - a))$



# Logistic regression derivatives





deeplearning.ai

# Basics of Neural Network Programming

---

Gradient descent  
on *m* examples

# Logistic regression on $m$ examples

$$\underline{J(w, b)} = \frac{1}{m} \sum_{i=1}^m \ell(a^{(i)}, y^{(i)})$$

$$\rightarrow a^{(i)} = \hat{y}^{(i)} = \sigma(z^{(i)}) = \sigma(w^T x^{(i)} + b)$$

$$(x^{(i)}, y^{(i)})$$

$$\underline{dw_1^{(i)}}, \underline{dw_2^{(i)}}, \underline{db^{(i)}}$$

$$\underline{\frac{\partial}{\partial w_1} J(w, b)} = \frac{1}{m} \sum_{i=1}^m \underbrace{\frac{\partial}{\partial w_1} \ell(a^{(i)}, y^{(i)})}_{\underline{dw_1^{(i)}} - (x^{(i)}, y^{(i)})}$$

# Logistic regression on $m$ examples

$$J=0; \underline{dw_1}=0; \underline{dw_2}=0; \underline{db}=0$$

→ For  $i=1$  to  $m$

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

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

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

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

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

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

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

$n=2$

$dw_3$   
 $\vdots$   
 $dw_n$

$J /= m \leftarrow$

$$\underset{\uparrow}{dw_1} /= m; \quad \underset{\uparrow}{dw_2} /= m; \quad \underset{\uparrow}{db} /= m. \quad \leftarrow$$

$$dw_1 = \frac{\partial J}{\partial w_1}$$

$$w_1 := w_1 - \alpha \underline{dw_1}$$

$$w_2 := w_2 - \alpha \underline{dw_2}$$

$$b := b - \alpha \underline{db}$$

Vectorization