deeplearning.ai

Basics of Neural Network Programming

Vectorizing Logistic Regression

Vectorizing Logistic Regression

$z^{(1)} = \boxed{w^T x^{(1)} + b}$ $\quad$ $z^{(2)} = \boxed{w^T x^{(2)} + b}$ $\quad$ $z^{(3)} = w^T x^{(3)} + b$

$\boxed{a^{(1)}} = \sigma(z^{(1)})$ $\quad$ $\boxed{a^{(2)}} = \sigma(z^{(2)})$ $\quad$ $\boxed{a^{(3)}} = \sigma(z^{(3)})$

$X = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix}$ $\quad$ (n_x, m) $\quad$ $\mathbb{R}^{n_x \times m}$

$[\text{———}]$ w^T $\begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix}$

$Z = \boxed{[z^{(1)} \; z^{(2)} \cdots z^{(m)}]} = \underbrace{w^T X + \underbrace{[b \; b \cdots b]}_{1 \times m}} = \left[\boxed{w^T x^{(1)} + b} \; \boxed{w^T x^{(2)} + b} \cdots w^T x^{(m)} + b\right]$

$1 \times m$

$\rightarrow Z = \text{np.dot}(w.T, X) + b$ $\quad$ $(1,1)$ $\quad$ $\mathbb{R}$ $\quad$ "Broadcasting"

$A = [a^{(1)} \; a^{(2)} \cdots a^{(m)}] = \sigma(Z)$

Andrew Ng

deeplearning.ai

Basics of Neural Network Programming

Vectorizing Logistic Regression's Gradient Computation

Vectorizing Logistic Regression

$dz^{(1)} = a^{(1)} - y^{(1)}$ $\quad$ $dz^{(2)} = a^{(2)} - y^{(2)}$ $\quad \cdots$

$$\boxed{dZ = [dz^{(1)} \; dz^{(2)} \cdots dz^{(m)}]}$$

$1 \times m$

$A = [a^{(1)} \cdots a^{(m)}]$. $\quad Y = [y^{(1)} \cdots y^{(m)}]$

$\rightarrow dZ = A - Y = [a^{(1)} - y^{(1)} \quad a^{(2)} - y^{(2)} \quad \cdots]$

$\cancel{dw_1}$
$\cancel{dw_2}$
$\vdots$

$\rightarrow dw = 0$

$dw \mathrel{+}= x^{(1)} dz^{(1)}$

$dw \mathrel{+}= x^{(2)} dz^{(2)}$

$\vdots$

$dw /= m$

$db = 0$

$db \mathrel{+}= dz^{(1)}$

$db \mathrel{+}= dz^{(2)}$

$\vdots$

$db \mathrel{+}= dz^{(m)}$

$db /= m.$

$$\boxed{db = \frac{1}{m} \sum_{i=1}^{m} dz^{(i)} = \frac{1}{m} \text{np.sum}(dZ)}$$

$$\boxed{dw = \frac{1}{m} X \, dZ^T}$$

$$= \frac{1}{m} \begin{bmatrix} | & & | \\ x^{(1)} & \cdots & x^{(m)} \\ | & & | \end{bmatrix} \begin{bmatrix} dz^{(1)} \\ \vdots \\ dz^{(m)} \end{bmatrix}$$

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

$n \times 1$

Andrew Ng

Implementing Logistic Regression

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

$\boxed{\texttt{for i = 1 to m:}}$

$$z^{(i)} = w^T x^{(i)} + b$$
$$a^{(i)} = \sigma(z^{(i)})$$
$$J \mathrel{+}= -[y^{(i)} \log a^{(i)} + (1 - y^{(i)}) \log(1 - a^{(i)})]$$
$$dz^{(i)} = a^{(i)} - y^{(i)}$$
$$\left.\begin{array}{l} dw_1 \mathrel{+}= x_1^{(i)} dz^{(i)} \\ dw_2 \mathrel{+}= x_2^{(i)} dz^{(i)} \end{array}\right\} \; dw \mathrel{+}= x^{(i)} * dz^{(i)}$$
$$db \mathrel{+}= dz^{(i)}$$

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

$db = db/m$

for iter in range(1000):

$$Z = w^T X + b$$
$$= np.dot(w.T, X) + b$$
$$A = \sigma(Z)$$
$$dZ = A - Y$$
$$dw = \frac{1}{m} X dZ^T$$
$$db = \frac{1}{m} np.sum(dZ)$$
$$w := w - \alpha dw$$
$$b := b - \alpha db$$

Andrew Ng