Basics of Neural Network Programming

deeplearning.ai

Vectorizing Logistic Regression

Vectorizing Logistic Regression

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

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

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

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

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

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

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

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

$1 \times m$

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

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

$\rightarrow dw = 0$

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

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

$\vdots$

$dw \mathrel{/}= m$

$\cancel{dw_1}$

$\cancel{dw_2}$

$\vdots$

$db = 0$

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

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

$\vdots$

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

$db \mathrel{/}= m.$

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

$$= \frac{1}{m} \; np.sum(dZ)$$

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

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

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

$n \times 1$

Implementing Logistic Regression

J = 0, $\mathrm{d}w_1$ = 0, $\mathrm{d}w_2$ = 0, db = 0

for i = 1 to m:

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

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

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

$$\mathrm{d}z^{(i)} = a^{(i)} - y^{(i)}$$

$$\mathrm{d}w_1 \mathrel{+}= x_1^{(i)} \mathrm{d}z^{(i)}$$

$$\mathrm{d}w_2 \mathrel{+}= x_2^{(i)} \mathrm{d}z^{(i)}$$

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

$$\mathrm{db} \mathrel{+}= \mathrm{d}z^{(i)}$$

J = J/m, $\mathrm{d}w_1$ = $\mathrm{d}w_1$/m, $\mathrm{d}w_2$ = $\mathrm{d}w_2$/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