



deeplearning.ai

Basics of Neural Network — Programming — Vectorizing Logistic Regression

Vectorizing Logistic Regression

$$\begin{aligned} \rightarrow z^{(1)} &= w^T x^{(1)} + b \\ \rightarrow a^{(1)} &= \sigma(z^{(1)}) \end{aligned}$$

$$\begin{aligned} z^{(2)} &= w^T x^{(2)} + b \\ \underline{a^{(2)}} &= \sigma(z^{(2)}) \end{aligned}$$

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

$$\underline{\underline{X}} = \begin{bmatrix} x^{(1)} & x^{(2)} & \dots & x^{(n)} \\ | & | & & | \\ | & | & & | \\ | & | & & | \end{bmatrix}$$

$$\frac{(n_x, m)}{\mathbb{R}^{n_x \times m}}$$

$$\vec{1} = \omega \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix}$$

$$\underline{z} = \begin{bmatrix} \underline{z}^{(1)} & \underline{z}^{(2)} & \dots & \underline{z}^{(m)} \end{bmatrix} = \underbrace{w^T X}_{1 \times m} + \underbrace{[b \ b \dots \ b]}_{1 \times m} = \boxed{w^T x^{(1)} + b} \quad \boxed{w^T x^{(2)} + b} \quad \dots \quad \boxed{w^T x^{(m)} + b}$$

$$\rightarrow \underline{z = np.dot(w.T, x) + \frac{b}{n}}_{(1,1)} \quad \mathbb{R}$$

$$A = [a^{(1)} \ a^{(2)} \ \dots \ a^{(n)}] = \sigma(z)$$



Basics of Neural Network

Programming
Vectorizing Logistic

Regression's Gradient
Computation

deeplearning.ai

Vectorizing Logistic Regression

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

$$\underline{dz} = [\underline{dz^{(1)}} \quad \underline{dz^{(2)}} \quad \dots \quad \underline{dz^{(m)}}] \quad \leftarrow$$

$1 \times m$

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

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

$$\begin{aligned} \rightarrow \underline{dw} &= 0 \\ \underline{dw} &+ = \underline{x^{(1)} dz^{(1)}} \\ \underline{dw} &+ = \underline{x^{(2)} dz^{(2)}} \\ &\vdots \\ \underline{dw} &= m \end{aligned}$$

~~dw_1~~
 ~~dw_2~~
 ~~\vdots~~

$$\begin{aligned} \underline{db} &= 0 \\ \underline{db} &+ = \underline{dz^{(1)}} \\ \underline{db} &+ = \underline{dz^{(2)}} \\ &\vdots \\ \underline{db} &+ = \underline{dz^{(m)}} \\ \underline{db} &= m. \end{aligned}$$

$$\begin{aligned} \underline{db} &= \frac{1}{m} \sum_{i=1}^m dz^{(i)} \\ &= \frac{1}{m} \text{np.sum}(\underline{dz}) \end{aligned}$$

$$\begin{aligned} \underline{dw} &= \frac{1}{m} X \underline{dz} \\ &= \frac{1}{m} \begin{bmatrix} x^{(1)} & \dots & x^{(m)} \\ 1 & & 1 \end{bmatrix} \begin{bmatrix} dz^{(1)} \\ \vdots \\ dz^{(m)} \end{bmatrix} \\ &= \frac{1}{m} \left[\underline{x^{(1)} dz^{(1)}} + \dots + \underline{x^{(m)} dz^{(m)}} \right] \\ &\quad n \times 1 \end{aligned}$$

Implementing Logistic Regression

$$J = 0, \quad dw_1 = 0, \quad dw_0 = db = dk_0 = 0$$

for i = 1 to m:

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

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

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

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

$$\left[\begin{array}{l} dw_1 += x_1^{(i)} dz^{(i)} \\ dw_2 += x_2^{(i)} dz^{(i)} \end{array} \right] \quad dw += x^{(i)} * dz^{(i)}$$

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

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

$$db = db/m$$

```

for iter in range(1000):
    Z = w.T * X + b
    = np.dot(w.T, X) + b
    A = sigma(Z)
    dz = A - Y
    dw = 1/m * X * dz.T
    db = 1/m * np.sum(dz)

    w := w - alpha * dw
    b := b - alpha * db
    
```