

Neural Network

classmate

Date _____

Page _____

Logistic Regression - Binary Classification

Notation

$$(x, y) \in \mathbb{R}^n \quad y \in \{0, 1\}$$

m-training example : $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \dots, (x^{(m)}, y^{(m)})\}$

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

$$X\text{-shape} = (n_x, m)$$

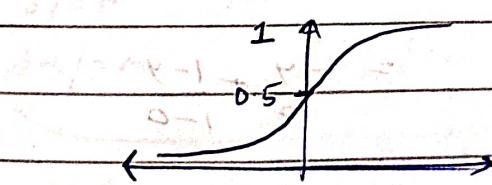
$$Y = [y^{(1)} \ y^{(2)} \ \dots \ y^{(m)}] \quad Y \in \mathbb{R}^{1 \times m}$$

$$Y\text{-shape} = (1, m)$$

Given x , $\hat{y} = P(y=1/x)$

$x \in \mathbb{R}^{n_x}$ $w \in \mathbb{R}^n$ $b \in \mathbb{R}$ \leftarrow Parameters

Output $\hat{y} = \log(\sigma(w^T x + b))$



$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Cost function we want $\hat{y}^{(i)} \approx y^{(i)}$ Given data.

Loss function $\ell(\hat{y}, y) = -(\hat{y} \log y + (1-\hat{y}) \log(1-\hat{y}))$
for single training example

Cost function $J(w, b) = \frac{1}{m} \sum_{i=1}^m \ell(\hat{y}^{(i)}, y^{(i)})$

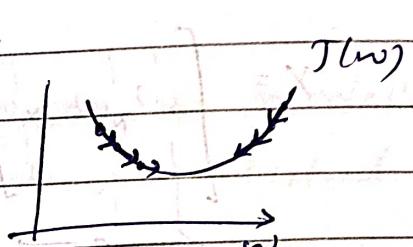
Gradient Descent

Want to find w, b that minimize $J(w, b)$
Step toward steepest gradient

Repeat {
 $w := w - \alpha \frac{d J(w)}{d w}$ $w := w - \alpha d w$

$$w := w - \alpha \frac{d J(w)}{d w}$$

$$w := w - \alpha d w$$



$$J(w, b) = w = w - \alpha d w$$

$$b := b - \alpha d b$$

$$\text{max } \hat{y} \rightarrow Y \quad [w] \quad \rightarrow [P] = Y$$

$$z = w^T x + b \rightarrow Y$$

→ for single training example

$$\hat{y} = a = \sigma(z) = \frac{1}{1+e^{-z}}$$

$$L(a, y) = -(y \log(a) + (1-y) \log(1-a))$$

$$x_1 \rightarrow z = w_1 x_1 + w_2 x_2 + b \rightarrow a = \sigma(z) \rightarrow L(a, y)$$

$$w_2 \rightarrow \frac{d z}{d a} = \frac{d L}{d a} \frac{d a}{d z}$$

$$b \rightarrow \frac{d z}{d b} = \frac{d L}{d b} \frac{d b}{d a}$$

$$= \frac{-y}{a} + \frac{1-y}{1-a}$$

$$dw_1 = x_1 \cdot dz \quad dw_2 = x_2 \cdot dz \quad db = dz$$

$$(1-p)(w_1 - \alpha d w_1, p) \rightarrow (p, \hat{p}) \text{ do not work well}$$

$$w_1 := w_1 - \alpha d w_1$$

$$b := b - \alpha d b$$

$$(1-p)(w_1 - \alpha d w_1, p) \rightarrow (1-p)(w_1 - \alpha d w_1, p) \text{ do not work well}$$

on an example (dual problem)

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m \alpha(o^{(i)}, y)$$

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

$$\frac{\partial J(w, b)}{\partial w_1} = \sum_{i=1}^m \frac{\partial \alpha^{(i)}}{\partial w_1}$$

$$J=0, \quad dw_1=0, \quad dw_2=0, \quad db=0 \quad dw = np - 300(n-x_1)$$

for $i=1 \text{ to } m$

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

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

$$(x_i) \rightarrow f(x_i) = \ln - [y^{(i)} \log \alpha^{(i)} + (1-y^{(i)}) \log (1-\alpha^{(i)})]$$

$$\frac{\partial z^{(i)}}{\partial w_1} = x_1^{(i)} \quad \frac{\partial z^{(i)}}{\partial w_2} = x_2^{(i)} \quad \frac{\partial z^{(i)}}{\partial b} = 1$$

$$dw_1 = x_1^{(i)} dz^{(i)} \quad dw_2 = x_2^{(i)} dz^{(i)} \quad db = dz^{(i)}$$

$$dw_1 = x_1^{(i)} dz^{(i)} \quad dw_2 = x_2^{(i)} dz^{(i)} \quad db = dz^{(i)}$$

$$J/m = \frac{1}{m} \sum_{i=1}^m \alpha^{(i)} = \frac{1}{m} \sum_{i=1}^m \ln - [y^{(i)} \log \alpha^{(i)} + (1-y^{(i)}) \log (1-\alpha^{(i)})]$$

$$w_1 := w_1 - \alpha dw_1$$

$$dw_1 = \frac{1}{m} \sum_{i=1}^m x_1^{(i)} dz^{(i)}$$

$$w_2 := w_2 - \alpha dw_2$$

$$dw_2 = \frac{1}{m} \sum_{i=1}^m x_2^{(i)} dz^{(i)}$$

$$b := b - \alpha db$$

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

→ Vectorization (Avoiding loop)

$$z = w^T x + b \quad (\underbrace{\begin{bmatrix} w_1 & w_2 & \dots & w_d \end{bmatrix}}_{m \times 1} w) \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}}_{d \times 1} + b = \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}}_{m \times d} \cdot \underbrace{\begin{bmatrix} x_1 & x_2 & \dots & x_d \end{bmatrix}}_{d \times 1} + b$$

Vectorized

$$z = np.dot(w, x) + b$$

$w^T x$

Non vectorized

$$\begin{aligned} p &= 1 \text{ to } n \\ z^+ &= w^{[i]} \cdot x^{[i]} + b \end{aligned}$$

$z^+ = b$

 $(u, x - \alpha)$ base-dm = wtsFaster, \approx 10x faster

eq① $u = Av$

$$u_i = \sum_j A_{ij} v_j$$

$$u = np.zeros((n, 1))$$

m at 1 = 1 ref.

$$d + \underbrace{(x^T w)}_{(n, 1) \times (1, n)} = u$$

$$u = np.dot(A, v)$$

~~$u = np.zeros(6)$~~

for i = 0 to 5:
 for j:

$$u[i] += A[i][j] * v[j]$$

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

$$v = \begin{bmatrix} e^{v_1} \\ \vdots \\ e^{v_n} \end{bmatrix}$$

import numpy as np

$$u = np.exp(v)$$

$$u = np.zeros((n, 1))$$

for i in range(n):
 $\rightarrow u[i] = \text{math.exp}(v[i])$

$$np.log(v)$$

$$np.abs(v)$$

$$np.maximum(x, 0)$$

→ Vectorization (Avoiding loop)

$$z = w^T x + b$$

$$(w_1 \ w_2 \ \dots \ w_m) w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$(w_1 x_1 + w_2 x_2 + \dots + w_m x_m) + b$$

vectorized

$$z = np.dot(w, x) + b$$

$w^T x$

Non vectorized

$$\begin{array}{l} q = 1 \text{ to } n \\ z += w^{q, i} x^{q, i} \end{array}$$

$$z += b$$

$$(w_1 x_1 + w_2 x_2 + \dots + w_m x_m) + b$$

Faster, ≈ 1000 times faster

$$eq(1) \quad u = Av$$

$$u_i = \sum_j A_{ij} v_j$$

$$u = np.zeros((n, 1))$$

$$u = np.dot(A, v)$$

~~$u = np.zeros(m)$~~

```
(i) for i = range(m):
    for j:
        u[i] += A[i][j] * v[j]
```

eq(2)

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

$$u = \begin{bmatrix} e^{v_1} \\ \vdots \\ e^{v_n} \end{bmatrix}$$

import numpy as np

$$u = np.exp(v)$$

$$u = np.zeros((n, 1))$$

```
for i in range(n):
    u[i] = math.exp(v[i])
```

$$np.log(v)$$

$$np.abs(v)$$

$$np.maximum(v, 0)$$

Qn logistic regression derivative

 $\theta = w/b$

$$J=0 \quad dw^{(1)} = np \cdot zero \cdot (h - x_1, 1) \quad (1) \quad s^T b^{(1)} x = +w/b$$

$$\text{Previous } \theta \quad (2) \quad s^T b^{(1)} x = +w/b$$

Vectorizing LR $m = \sqrt{d} \cdot \theta$ $m = \sqrt{w/b}$

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

$$\theta^{(1)} = \sigma(z^{(1)}) \quad \theta^{(2)} = \sigma(z^{(2)})$$

(step function)

$$x = \begin{bmatrix} | & | \\ x^{(1)} & x^{(2)} \\ | & | \\ \vdots & \vdots \end{bmatrix} \quad (n_x, m)$$

$$z = [z^{(1)}, z^{(2)}, \dots, z^{(m)}] = w^T x + [b, b, \dots, b]$$

$$dt^T X \cdot \theta = \sum_{i=1}^m \theta_i \cdot x_i \cdot dt_i \quad (\theta = d\theta) \quad \text{automatically}$$

$$dt^T X \cdot \theta = \sum_{i=1}^m \theta_i \cdot x_i \cdot dt_i \quad \text{if this is true} \rightarrow \theta = d\theta \quad \text{called broadcasting}$$

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

$$dZ = [dz^{(1)}, dz^{(2)}, \dots, dz^{(m)}] \quad (1) \quad s^T b^{(1)} x = +w/b$$

 $w/b \approx w \approx w$

we know

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

$$dZ = A - Y$$

$$dw = 0$$

$$dw_t = x^{(1)} dz^{(1)}$$

$$dw_t = x^{(2)} dz^{(2)}$$

$$dw = m$$

$$db \doteq b$$

$$(1, x^{(1)}) db + t = dz^{(1)} \text{ where } Q = C$$

$$db + t = dz^{(2)} \text{ (similarly)}$$

$$db = m$$

$$\begin{aligned} d + (x^T w) &= b \\ db &= (z^{(1)})_0 = \sum_{j=1}^m b_j + \frac{1}{m} \sum_{j=1}^m z^{(1)}_j \\ dw &= \frac{1}{m} X dz^t \end{aligned}$$

Implementing in LR

for iteration in range

$$[d - \dots - d] + X^T w = [z^{(1)}_0 \dots z^{(1)}_m], [z^{(1)}_0 \dots z^{(1)}_m] = z$$

$$J=0, dw_t = np \cdot zeros((n-x, 1)) \quad db=D$$

$$Z = w^T X + b$$

$$\text{for i from 0 to m-1: } d + (X, T \cdot w) \text{ take grad} \rightarrow = -np \cdot dot(w - T, x) +$$

for i=1 to (m-1)

$$A = \sigma(Z)$$

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

$$a^{(i)} = \sigma(z^{(i)}) \quad (z^{(i)})_0 = [w_0]$$

$$dz^{(i)} = A - Y$$

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

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

$$dw = \frac{1}{m} X dz^t$$

$$dw_t = x^{(i)} dz^{(i)}$$

$$db = \frac{1}{m} np \cdot sum(dw)$$

$$db_t = dz^{(i)}$$

$$db = db/m$$

$$J = J/m, dw_t = dw/m, db = db/m$$

$$Y = A - Sb$$

Broad Casting

Ex

Calories

Protein

(a) above ordering is \Rightarrow
(b) \Rightarrow ordering

Apples

Beef

Egg

Potatoes

Lemons

Carb	56.0	(0.0)	4.4	68.0	→ $A_{(3,4)}$
Protein	1.2	104.0	52.0	(28.0)	
Fat	1.8	135.0	99.0	0.9	

(row) \Rightarrow start \Rightarrow transpose

$$\text{cal} = A \cdot \text{sum}(\text{axis}=0) \quad (\text{axis}=1)$$

sum vertically sum horizontally

$$[59, 239, 155.4, 76.9]$$

percentag = $100 * A / \text{col_reshape}(1,4)$

$P = (x/p)^q$, no need (coz already $(1,4)$)

$$\begin{bmatrix} 94.9 \\ 2.03 \\ 3.05 \end{bmatrix} \xrightarrow{\text{---}} \begin{bmatrix} P \\ P \\ P \end{bmatrix} = (x/p)^q \quad 0 \leq p \leq 1$$

Ex

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} + 100 \rightarrow \begin{bmatrix} 101 \\ 102 \\ 103 \\ 104 \end{bmatrix} \quad P = (x/p)^q \quad 1 \leq p \leq V$$

$$P_{-1} = (x/p)^q \quad 0 \leq p \leq V$$

$$(P \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} + P \begin{bmatrix} 100 & 200 & 300 \\ 100 & 200 & 300 \end{bmatrix}) P^{-1} = \begin{bmatrix} 101 & 202 & 303 \\ 104 & 205 & 306 \end{bmatrix}$$

$(m,n) \rightarrow (m,n)$

General principle

$$(m,n) \xrightarrow{+x/p} (1,n) \xrightarrow{\text{---}} (m,n) \quad ()^q \quad \text{and}$$

$$* \quad (m,1) \rightarrow (m,n)$$

$$(x/p)^q \text{ for } S$$

$$(m,p) \xrightarrow{+x/p} -$$

$a = np.random.randn(5)$
 $a.shape = (5, 1)$
 "rank 1 array")

Don't use

extra 0's

instead $a = np.random.randn(5, 1)$
 $\begin{bmatrix} 0.52 \\ -0.101 \\ 2.1 \\ 0.281 \\ 8.1 \end{bmatrix}$

does

nothing

but

 $\text{assert}(a.shape == (5, 1))$

(True)

 $(a = zeros(5) + 1) == 1$

refined code $a = a.reshape((5, 1))$

(P.D.F. 1.221 PES. 1.21)

Logistic Regression Cost Function

$$\text{if } y=1 \quad p(y/x) = \hat{y}$$

$$\text{if } y=0 \quad p(y/x) = 1-\hat{y}$$

$$p(y/x) = \hat{y}^y (1-\hat{y})^{1-y}$$

$$\begin{cases} y=1 & p(y/x) = \hat{y} \\ y=0 & p(y/x) = 1-\hat{y} \end{cases}$$

$$\log p(y/x) = \log \hat{y}^y (1-\hat{y})^{1-y} = y \log \hat{y} + (1-y) \log (1-\hat{y})$$

$$= -\mathcal{L}(\hat{y}, y)$$

In example

$$\log P(\cdot) = \log \prod_{i=1}^m p(y^{(i)}|x^{(i)})$$

(row) \leftrightarrow (row)

$$= \sum \log p(y^{(i)}|x^{(i)})$$

$$-\mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

$$= - \sum_{i=1}^m \alpha(\hat{y}^{(i)}, y^{(i)})$$

Cost
(minimize) $J(w, b) = \frac{1}{m} \sum_{i=1}^m \alpha(\hat{y}^{(i)}, y^{(i)})$