Machine Learning

Linear Regression
Logistic Regression

Nguyen Quoc Thai

CONTENT

1 – Linear Regression

Linear Regression

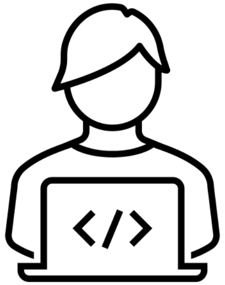
Data

Level	Salary
0	8
1	15
2	18
3	22
4	26
5	30
6	38
7	47

Level	Salary
3.5	???
10	???

Prediction

Learning



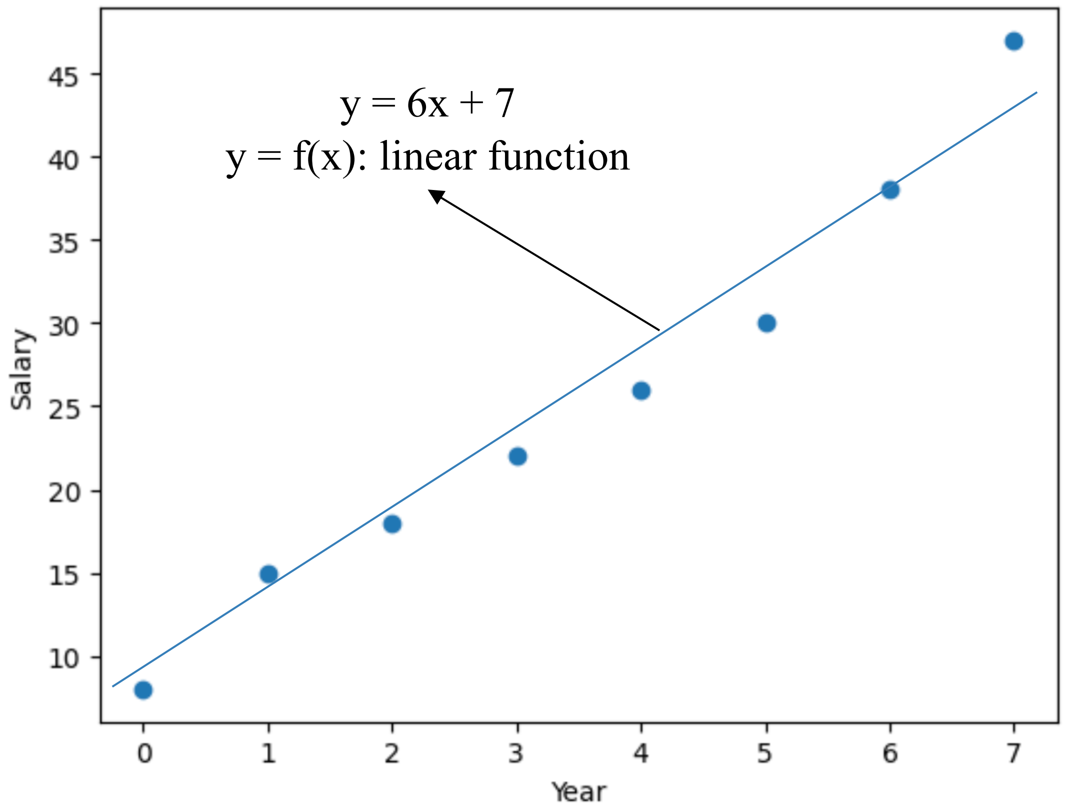
1 – Linear Regression

Linear Regression

Data

Level	Salary
0	8
1	15
2	18
3	22
4	26
5	30
6	38
7	47

Visualization



1 – Linear Regression

Linear Regression

Data

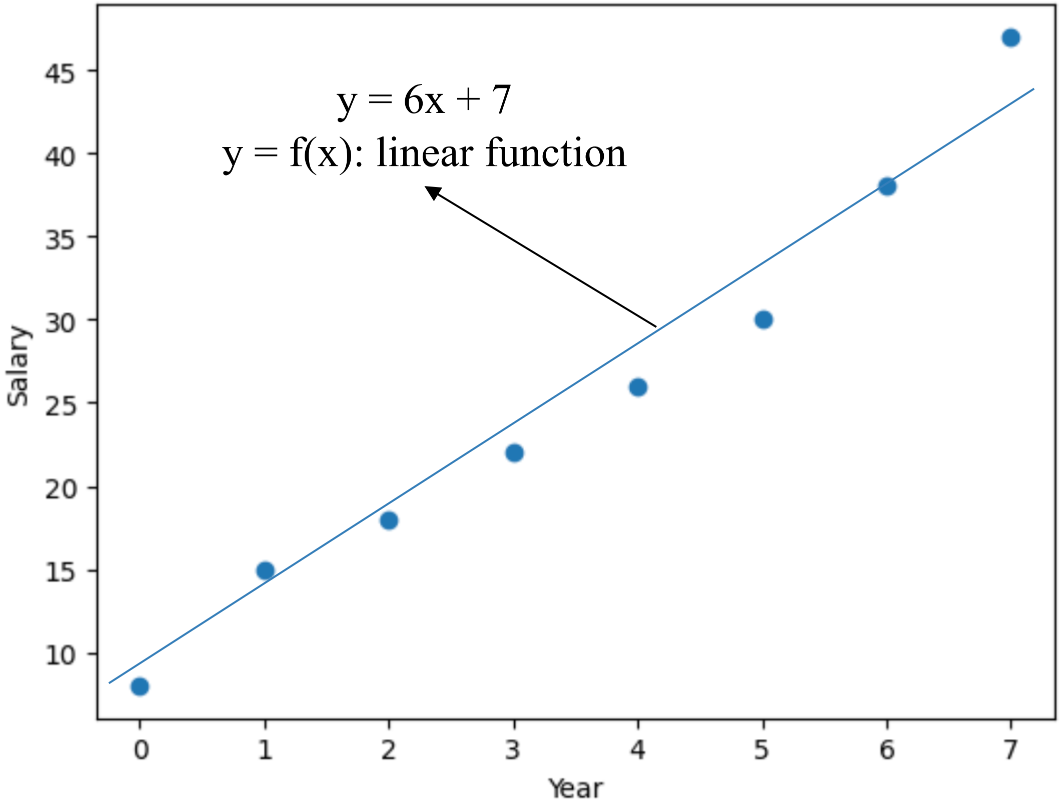
Level	Salary
0	8
1	15
2	18
3	22
4	26
5	30
6	38
7	47

Modeling

$$y = wx + b$$

Find w and b to fit the data

Visualization



1 – Linear Regression

Linear Regression using Gradient Descent

Modeling

$y = wx + b$

Init θ

lr = 0.1

$y = 2x + 2$

$x = [1 \;\; 2]$

$y = 18$

Data

$y = 2x + 2 = 6$

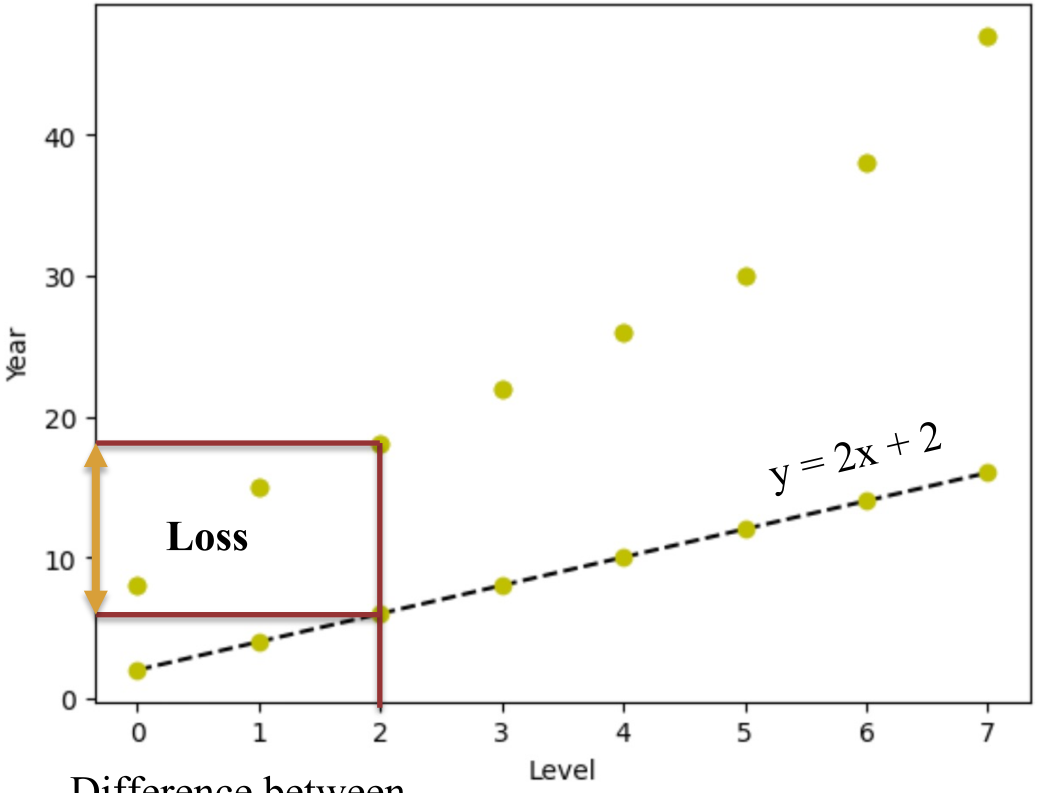
$L = (6 - 18)^2 = 144$

k = -24

$L' = \begin{bmatrix} -24 \\ -48 \end{bmatrix}$

$\theta = \begin{bmatrix} 4.4 \\ 6.8 \end{bmatrix}$

Visualization



Difference between predicted and actual value

1 – Linear Regression

Linear Regression using Gradient Descent

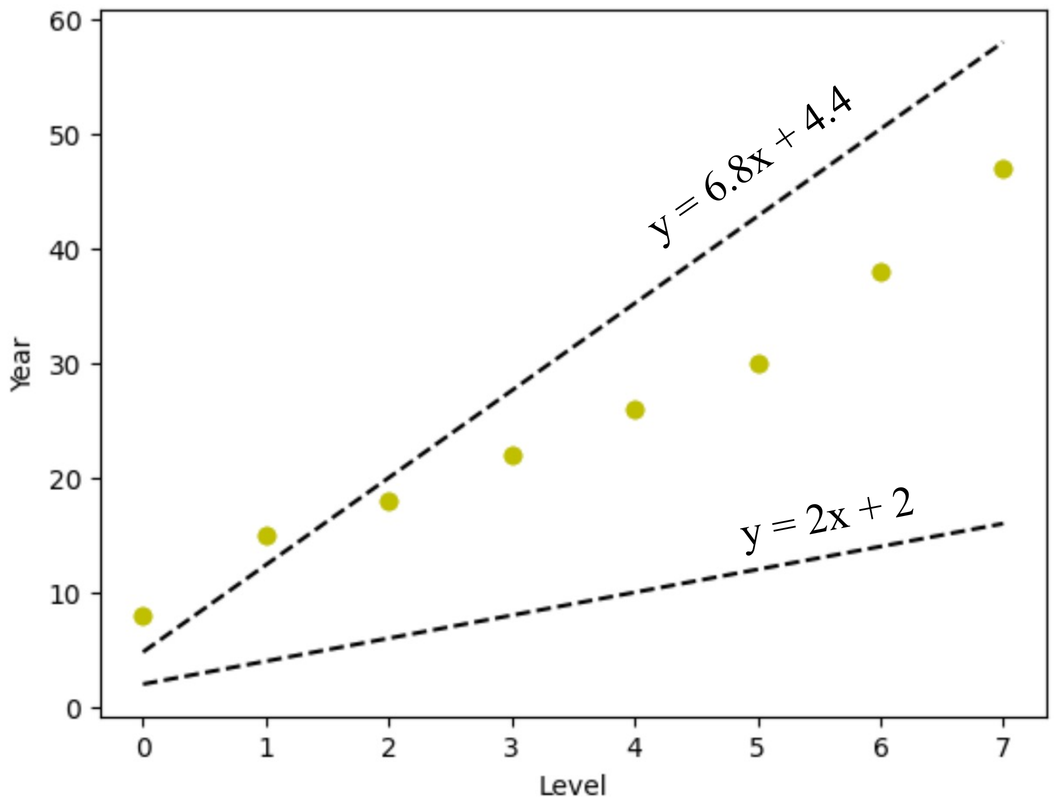
Modeling

$y = wx + b$ → $y = 2x + 2$

Updated

$y = 6.8x + 4.4$

Visualization



1 – Linear Regression

Implement for One Sample

Traditional Basic Python

1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

Vectorized Numpy

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\nabla_{\theta} \mathrm{L} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$$

η is learning rate

1 – Linear Regression

Implement using Basic Python for One Sample

1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

```
# a sample
x = 2
y = 18
```

```
# init weights
w = 2
b = 2
```

1 – Linear Regression

Implement using Basic Python for One Sample

1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

```
# forward
def predict(x, w, b):
    return x*w + b

y_hat = predict(x, w, b)
y_hat
```

```
6
```

1 – Linear Regression

Implement using Basic Python for One Sample

1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

```
# compute loss
def compute_loss(y_hat, y):
    return (y_hat-y)*(y_hat-y)

loss = compute_loss(y_hat, y)
loss
```

```
144
```

1 – Linear Regression

Implement using Basic Python for One Sample

1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

```python
# compute gradient
def compute_gradient(x, y, y_hat):
    dw = 2*x*(y_hat-y)
    db = 2*(y_hat-y)
    return dw, db

dw, db = compute_gradient(x, y, y_hat)
dw, db
```

```
(-48, -24)
```

1 – Linear Regression

Implement using Basic Python for One Sample

1) Pick a sample (x, y) from training data

2) Compute the output $\hat{y}$

$$\hat{y} = wx + b$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\frac{\partial L}{\partial w} = 2x(\hat{y} - y) \qquad \frac{\partial L}{\partial b} = 2(\hat{y} - y)$$

5) Update parameters

$$w = w - \eta \frac{\partial L}{\partial w} \qquad b = b - \eta \frac{\partial L}{\partial b}$$

η is learning rate

```
# update weights
lr = 0.1

def update_weights(w, b, dw, db, lr):
    new_w = w - lr*dw
    new_b = b - lr*db
    return new_w, new_b

new_w, new_b = update_weights(w, b, dw, db, lr)
new_w, new_b
```

```
(6.800000000000001, 4.4)
```

1 – Linear Regression

Implement using Numpy for One Sample

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\nabla_{\theta} \mathrm{L} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$$

η is learning rate

```
x = np.array([[2]])
y = np.array([18])
```

```
num_samples = x.shape[0]
num_samples
```

```
1
```

```
# append bias
x = np.hstack([x, np.ones((num_samples, 1))])
x
```

```
array([[2., 1.]])
```

```
# init weights
theta = np.array([2, 2]) # num_features: x.shape[1]
theta
```

```
array([2, 2])
```

1 – Linear Regression

Implement using Numpy for One Sample

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\nabla_{\theta} \mathrm{L} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$$

η is learning rate

```
# forward
def predict(x, theta):
    return x.dot(theta)

y_hat = predict(x, theta)
y_hat
```

```
array([6.])
```

1 – Linear Regression

Implement using Numpy for One Sample

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\nabla_{\theta} \mathrm{L} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$$

η is learning rate

```
# compute loss
def compute_loss(y_hat, y):
    return (y_hat-y)*(y_hat-y)

loss = compute_loss(y_hat, y)
loss
```

```
array([144.])
```

1 – Linear Regression

Implement using Numpy for One Sample

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\nabla_{\theta} \mathrm{L} = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$$

η is learning rate

```
# compute gradient
def compute_gradient(x, y, y_hat):
    d_theta = 2*x*(y_hat-y)
    return d_theta

d_theta = compute_gradient(x, y, y_hat)
d_theta
```

```
array([[-48., -24.]])
```

1 – Linear Regression

Implement using Numpy for One Sample

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$\hat{y} = \boldsymbol{\theta}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\theta}$$

3) Compute loss

$$L = (\hat{y} - y)^2$$

4) Compute derivative

$$\nabla_{\theta} L = 2\boldsymbol{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} L$$

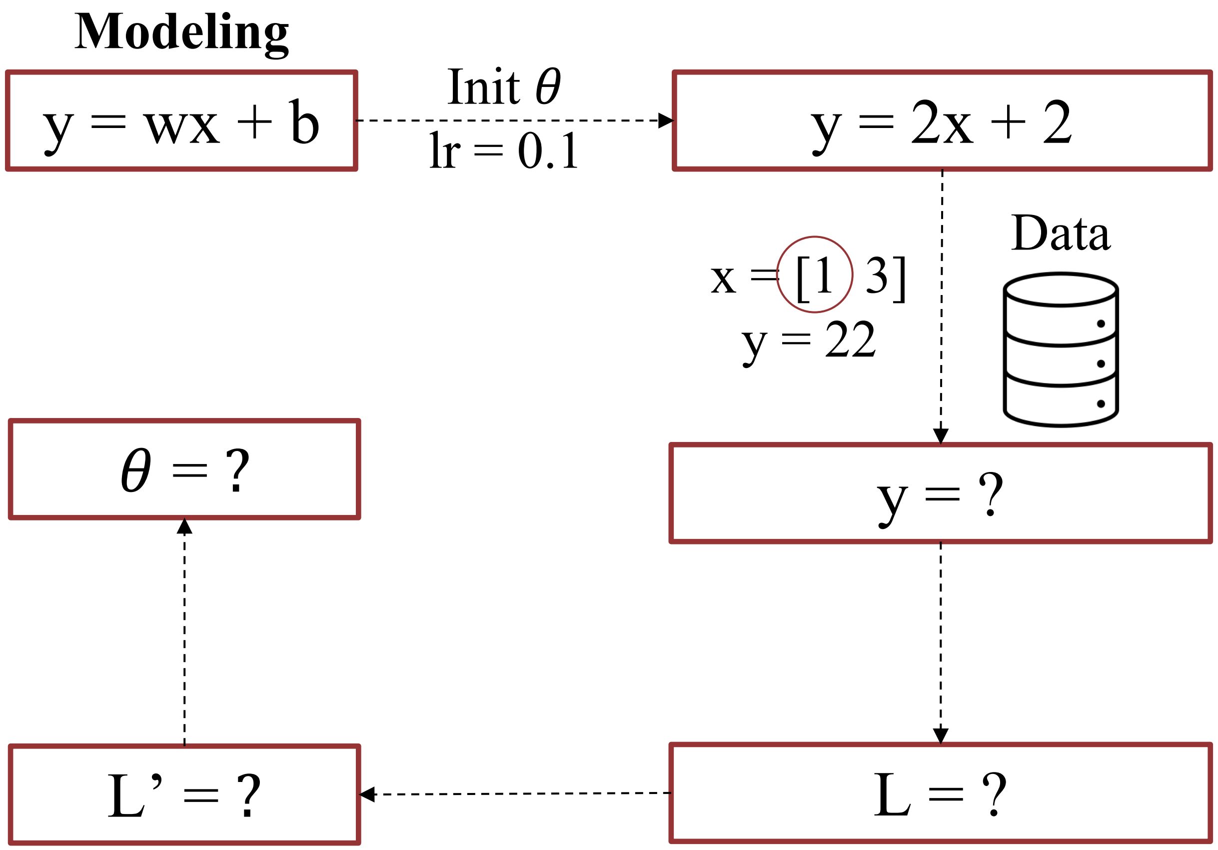
η is learning rate

```
# update weights
lr = 0.1

def update_weights(theta, d_theta, lr):
    new_theta = theta - lr*d_theta
    return new_theta

new_theta = update_weights(theta, d_theta, lr)
new_theta
```

```
array([[6.8, 4.4]])
```

1 – Linear Regression

Pratice



1 – Linear Regression

Pratice

Modeling

$y = wx + b$

Init θ
$lr = 0.1$

$y = 2x + 2$

Data

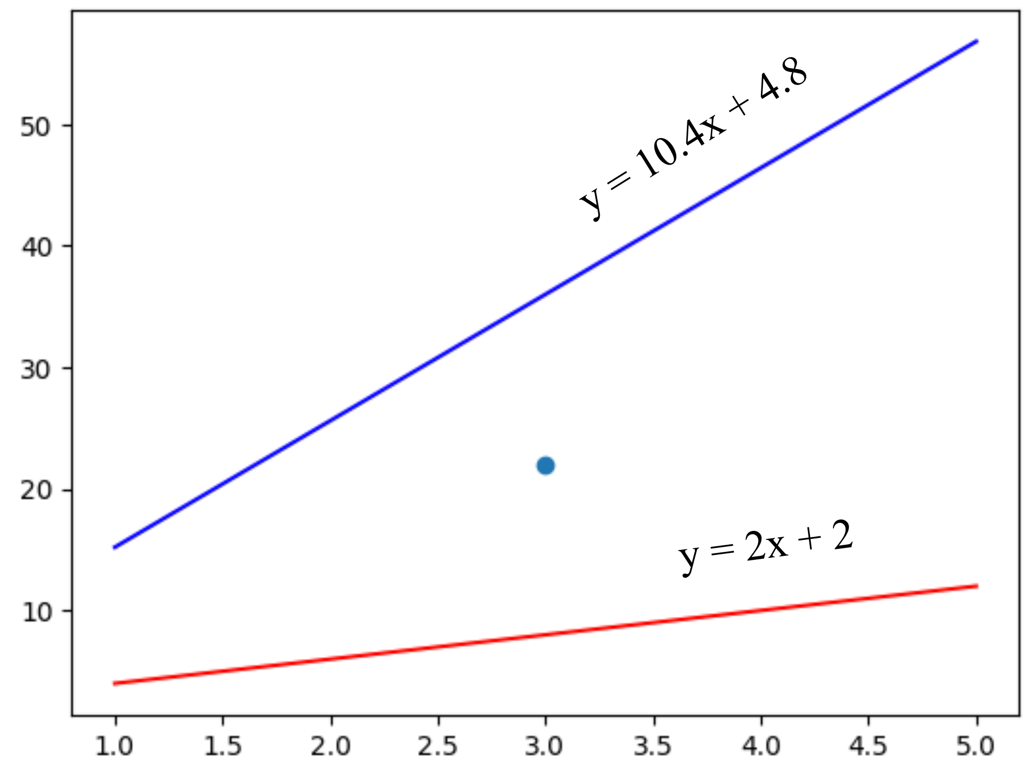
$x = [1 \ \ 3]$
$y = 22$

$y = 2x + 2 = 8$

$L = (6 - 18)^2 = 196$

$L' = \begin{bmatrix} -28 \\ -84 \end{bmatrix}$

$\theta = \begin{bmatrix} 4.8 \\ 10.4 \end{bmatrix}$



1 – Linear Regression

Problem

Data

Level	Salary
0	8
1	15
2	18
3	22
4	26
5	30
6	38
7	47

Any value

$y = ax + b$

Values: [0, 1]
Discrete values: {0, 1}

Need flexible model?

Data #1

Hours	Pass
0.5	0
1.0	0
1.5	0
2.0	0
2.5	1
3.0	1
3.5	1
4.0	1

2 – Logistic Regresion

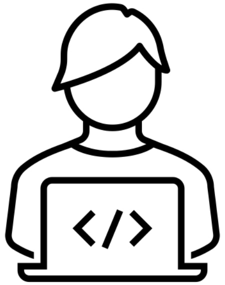
Problem

Data #1

Hours	Pass
0.5	0
1.0	0
1.5	0
2.0	0
2.5	1
3.0	1
3.5	1
4.0	1

Hours	Pass
0.25	???
4.5	???

Learning



Prediction

AI VIET NAM
@aivietnam.edu.vn

2 – Logistic Regresion

Problem

Data #1

Hours	Pass
0.5	0
1.0	0
1.5	0
2.0	0
2.5	1
3.0	1
3.5	1
4.0	1

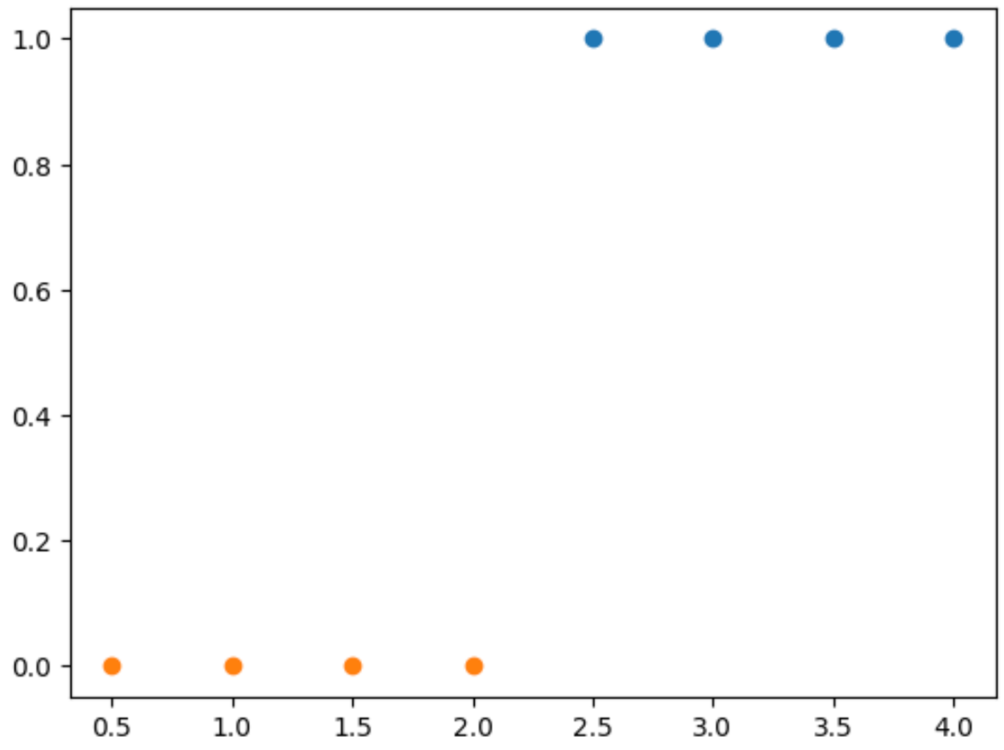
Modeling

y = f(x)

Find a function to fit the data

Sigmoid function

Visualization



2 – Logistic Regresion

Sigmoid Function

Sigmoid function

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

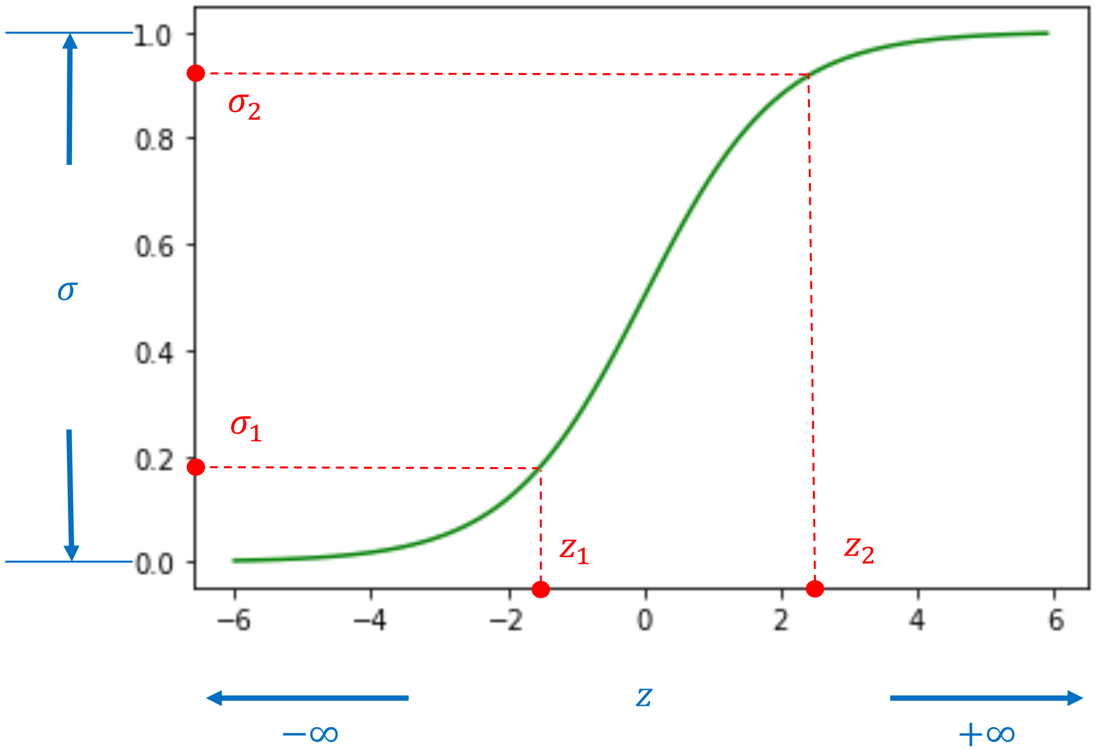
$z \in (-\infty \quad +\infty)$

$\sigma(u) \in (0 \quad 1)$

Property

$\forall z_1 z_2 \in [a \;\; b]$ and $z_1 \leq z_2$

$\rightarrow \sigma(z_1) \leq z(u_1)$



2 – Logistic Regresion

Sigmoid Function

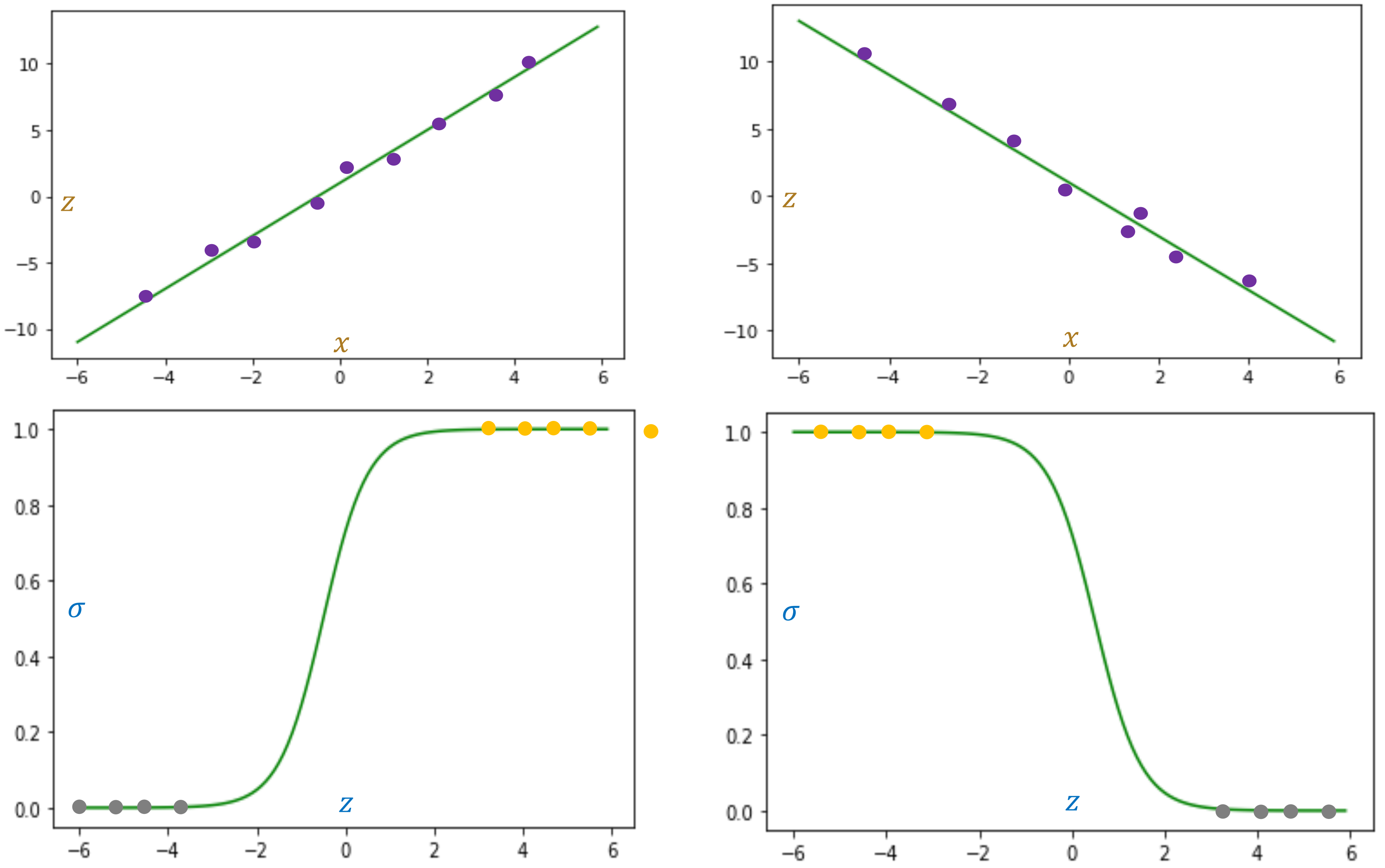
$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

$$z \in (-\infty \quad +\infty)$$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

$$\sigma(z) \in (0 \quad 1)$$



2 – Logistic Regresion

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_{\theta} \mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$ η is learning rate

$$\boldsymbol{\theta}^T = [b \quad w]$$

$$\boldsymbol{\theta}^T = [0.1 \quad 0.1]$$

$$\boldsymbol{x}^T = [1 \quad 0.5]$$

$$\boldsymbol{y} = [0]$$

$$\eta = 0.1$$

Data #1

Hours	Pass
0.5	0
1.0	0
1.5	0
2.0	0
2.5	1
3.0	1
3.5	1
4.0	1

AI VIET NAM
@aivietnam.edu.vn

2 – Logistic Regresion

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

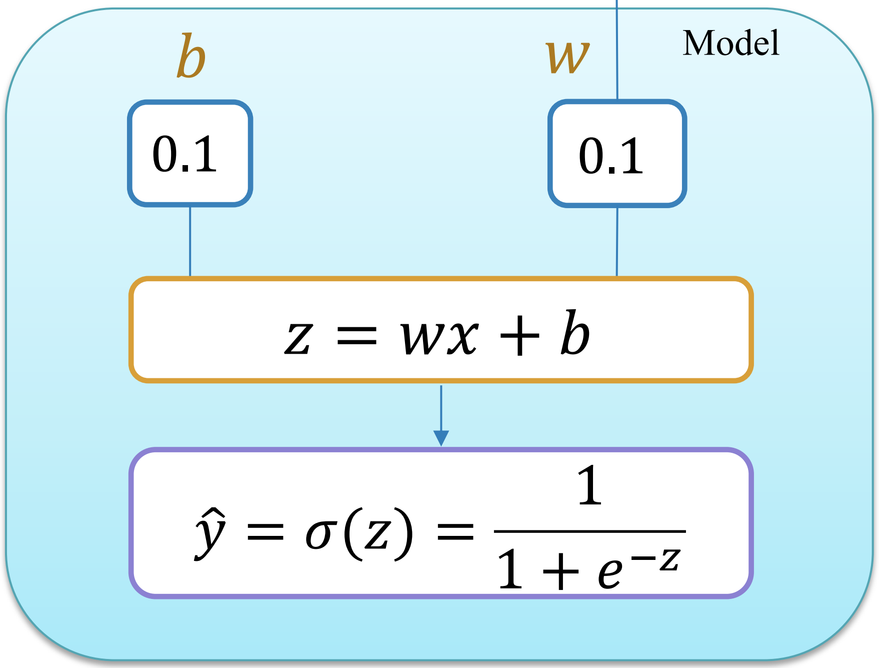
4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$



$z = 0.15$

$\hat{y} = 0.54$

2 – Logistic Regresion

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

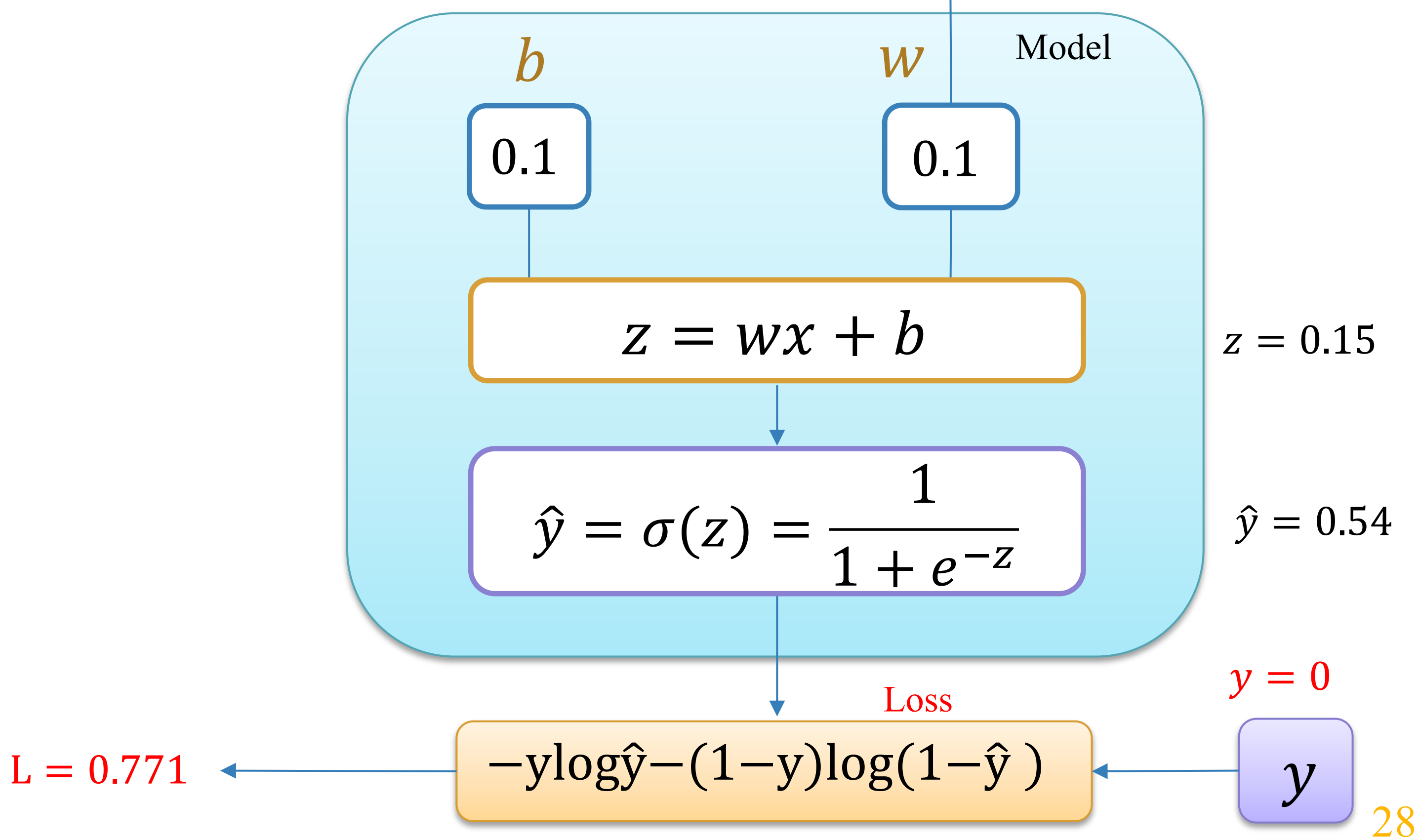
4) Compute derivative

$$\nabla_{\theta} L = \mathbf{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} L \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$



2 – Logistic Regression

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

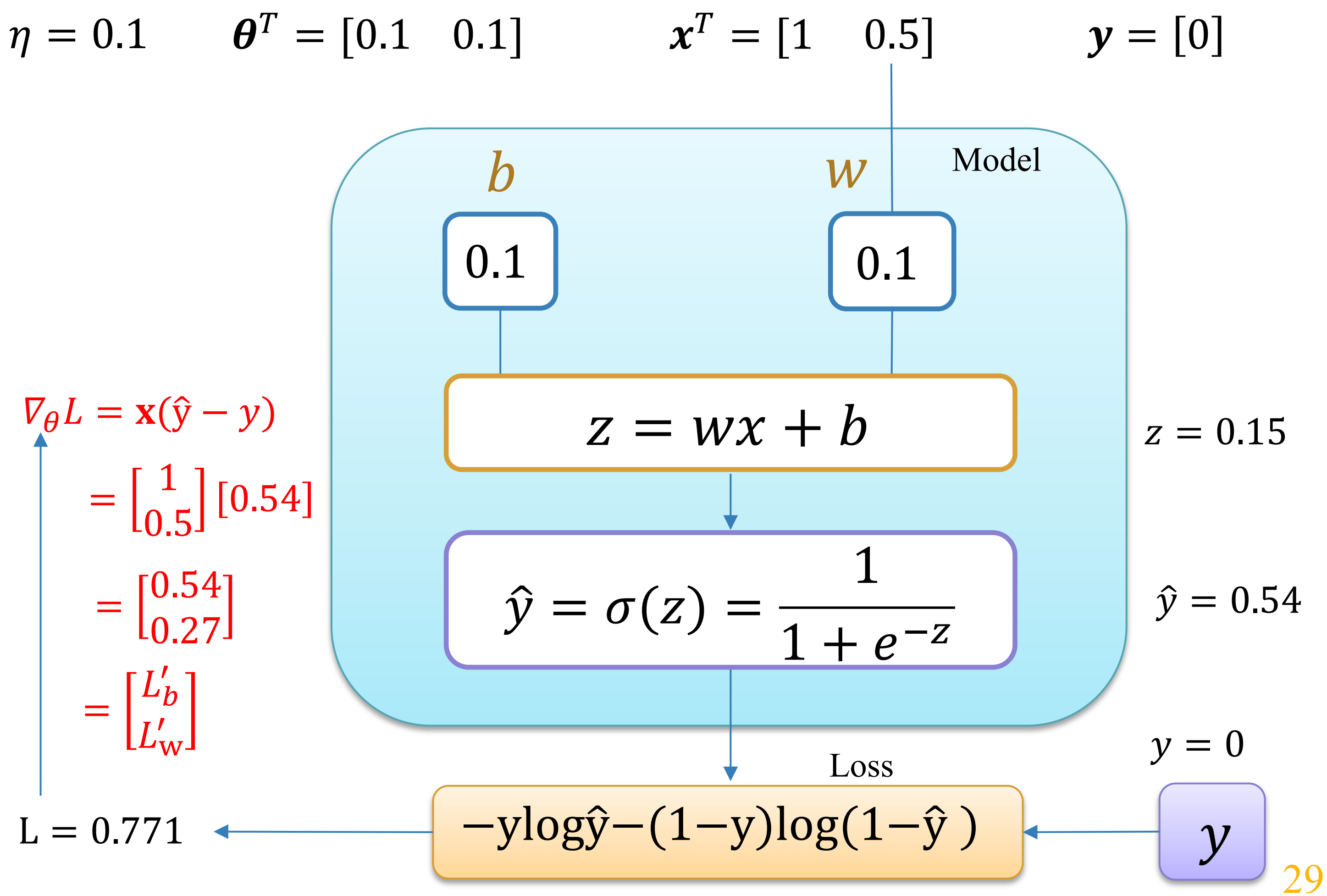
$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_\theta \mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_\theta \mathrm{L} \qquad \eta \text{ is learning rate}$$



2 – Logistic Regresion

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

$b = 0.1 - \eta 0.54$
$= 0.046$

$w = 0.1 - \eta 0.27$
$= 0.073$

$$\nabla_{\theta} L = \mathbf{x}(\hat{\mathrm{y}} - y) = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} [0.54] = \begin{bmatrix} 0.54 \\ 0.27 \end{bmatrix} = \begin{bmatrix} L'_b \\ L'_w \end{bmatrix}$$

Model

b: 0.1 $\quad$ w: 0.1

$$z = wx + b$$

$z = 0.15$

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

$\hat{y} = 0.54$

Loss

$$-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}})$$

$y = 0$ $\quad$ y

$\mathrm{L} = 0.771$

2 – Logistic Regression

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.046 \quad 0.073]$ $\quad \boldsymbol{x}^T = [1 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

$b = 0.1 - \eta 0.54$
$= 0.046$

$w = 0.1 - \eta 0.27$
$= 0.073$

Model

b: 0.046

w: 0.073

$$z = wx + b$$

$z = 0.0825$

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

$\hat{y} = 0.52$

Loss

$$-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}})$$

$y = 0$

y

Previous L $= 0.777$

L $= 0.735$

2 – Logistic Regression

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L}$$

η is learning rate

$\boldsymbol{\theta}^T = [b \quad w]$

$\boldsymbol{\theta}^T = [0.046 \quad 0.073]$

$\boldsymbol{x}^T = [1 \quad 1.0]$

$\boldsymbol{y} = [0]$

$\eta = 0.1$

Data #1

Hours	Pass
0.5	0
1.0	0
1.5	0
2.0	0
2.5	1
3.0	1
3.5	1
4.0	1

2 – Logistic Regresion

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

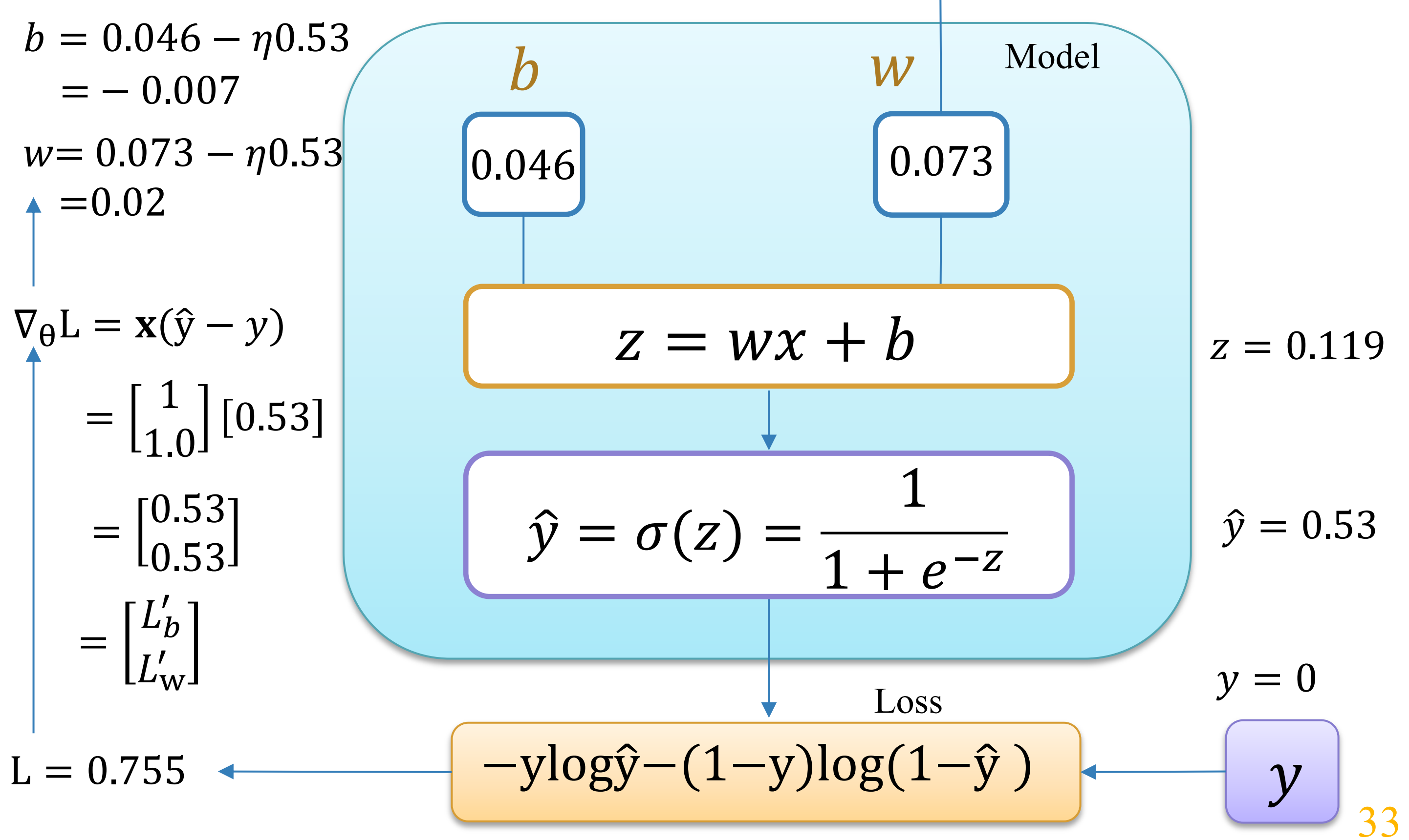
4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.046 \quad 0.073]$ $\quad \boldsymbol{x}^T = [1 \quad 1.0]$ $\quad \boldsymbol{y} = [0]$



2 – Logistic Regresion

Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

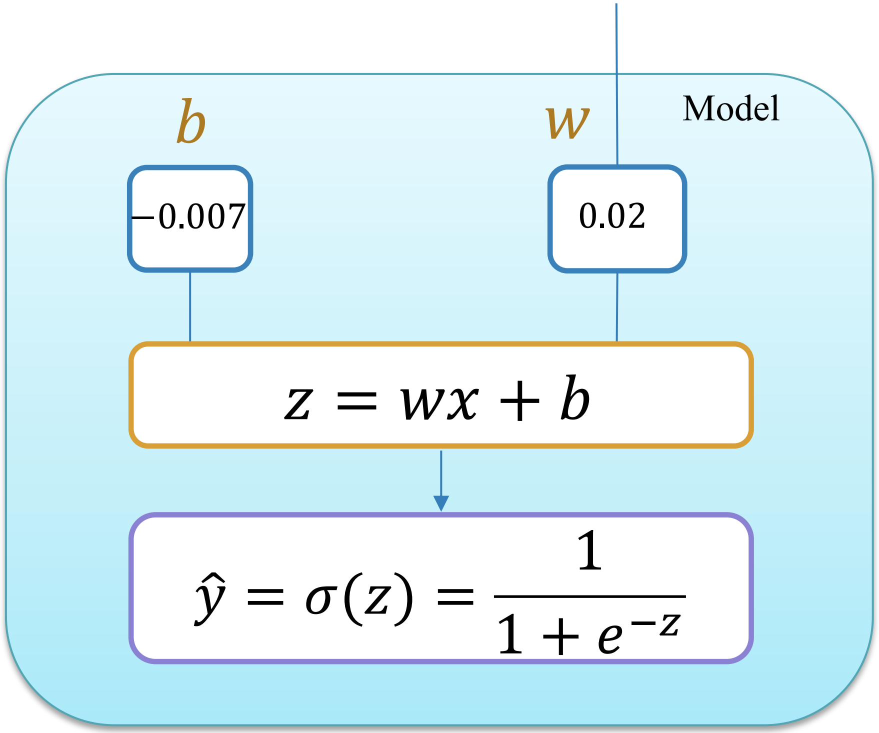
4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [-0.007 \ \ 0.02]$ $\quad \boldsymbol{x}^T = [1 \quad 1.0]$ $\quad \boldsymbol{y} = [0]$



2 – Logistic Regresion

Prediction

Data #1

Hours	Pass
0.5	0
1.0	0
1.5	0
2.0	0
2.5	1
3.0	1
3.5	1
4.0	1

Learning

Hours	Pass
0.25	???
4.5	???

Prediction

$\boldsymbol{\theta}^T = [0.046 \;\; 0.073]$

b: -0.007

w: 0.02

$$z = wx + b$$

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

2 – Logistic Regresion

Prediction

Hours	Pass
0.25	???
4.5	???

Prediction

b: -0.007 w: 0.02

$$z = wx + b = -0.002$$

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} = 0.499$$

Thresholds = 0.5

$y_{pred}: 0$

Hours	Pass
0.25	???
4.5	???

Prediction

b: -0.007 w: 0.02

$$z = wx + b = 0.083$$

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} = 0.52$$

Thresholds = 0.5

$y_{pred}: 1$

2 – Logistic Regression

Multivariable Logistic Regression

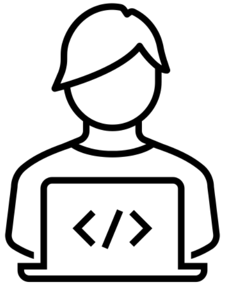
Data #2

Day	Hours	Pass
1	0.5	0
2	1.0	0
3	1.5	1
2	2.0	0
1	2.5	0
2	3.0	1
1	3.5	1
2	4.0	1

Day	Hours	Pass
2	0.25	???
1	4.5	???

Prediction

Learning



2 – Logistic Regresion

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

$$\boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$$

$$\boldsymbol{y} = [0]$$

$$\eta = 0.1$$

$$\boldsymbol{\theta}^T = [b \quad w_1 \quad w_2]$$

$$\boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$$

Data #2

Day	Hours	Pass
1	0.5	0
2	1.0	0
3	1.5	1
2	2.0	0
1	2.5	0
2	3.0	1
1	3.5	1
2	4.0	1

2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_{\theta} \mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$ $\quad$ η is learning rate

```
x = np.array([1.0, 1.0, 0.5])
y = np.array([0])
x, y
```

```
(array([1. , 1. , 0.5]), array([0]))
```

```
theta = np.array([0.1, 0.2, 0.1])
theta
```

```
array([0.1, 0.2, 0.1])
```

2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}\,))$$

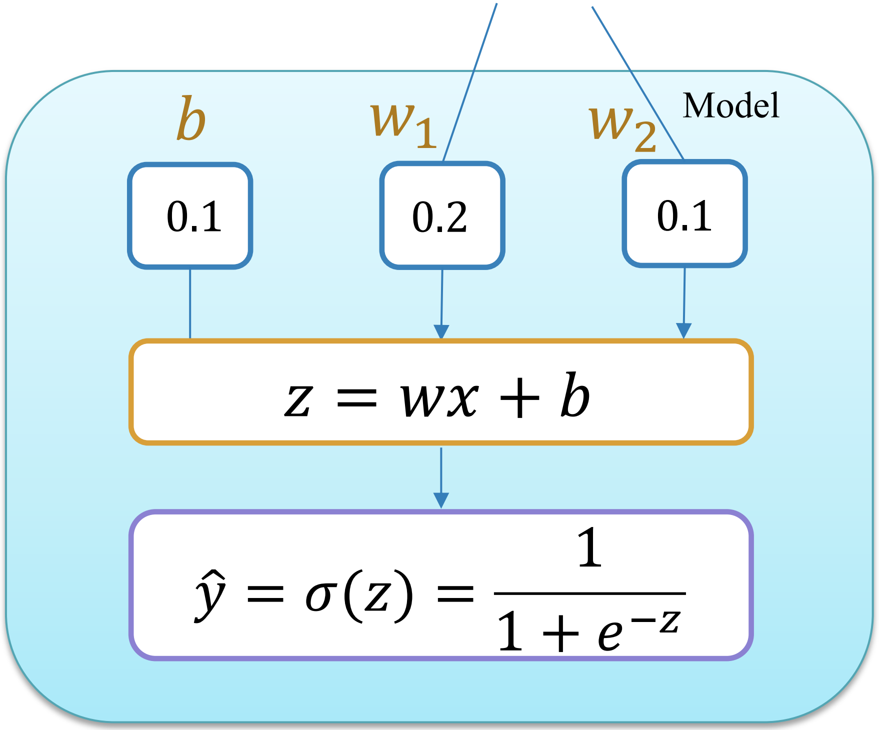
4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L}$ $\quad\eta$ is learning rate

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$



$z = 0.35$

$\hat{y} = 0.59$

2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\text{y}\log\hat{\text{y}} - (1-\text{y})\log(1-\hat{\text{y}}))$$

4) Compute derivative

$$\nabla_{\theta}\text{L} = \mathbf{x}(\hat{\text{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\text{L} \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

```
# define logistic function
def logistic_function(x):
    return 1/(1 + np.exp(-x))
```

```
# forward
def predict(x, theta):
    z = np.dot(x, theta)
    y_hat = logistic_function(z)
    return z, y_hat

z, y_hat = predict(x, theta)
z, y_hat
```

```
(0.35000000000000003, 0.5866175789173301)
```

2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

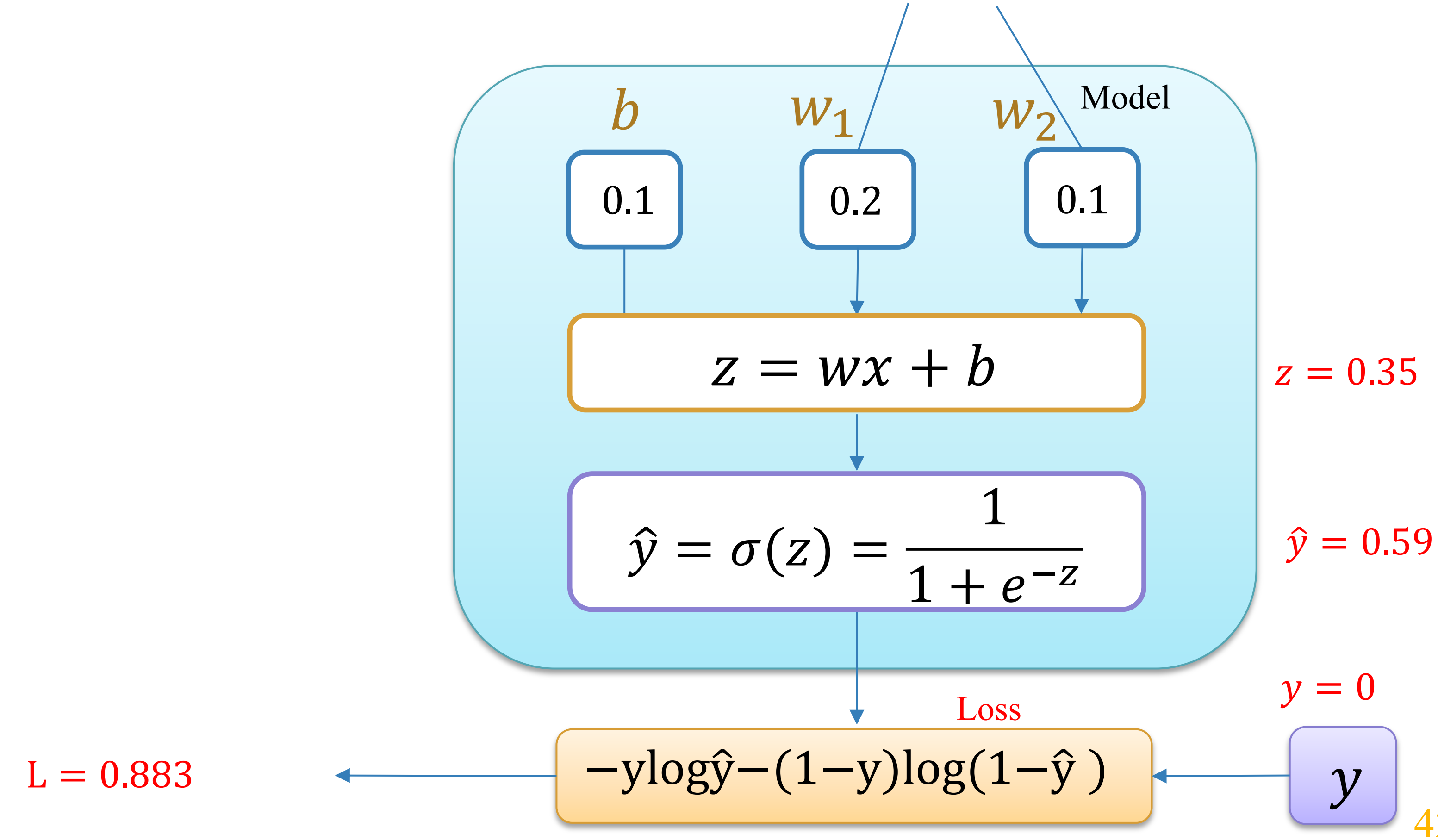
4) Compute derivative

$$\nabla_\theta L = \mathbf{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_\theta L \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$



2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1 - \mathrm{y})\log(1 - \hat{\mathrm{y}}\ ))$$

4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L}$$ η is learning rate

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

```python
# comput loss
def compute_loss(y_hat, y):
    loss = -1*((y * np.log(y_hat)) + ((1 - y) * np.log(1 - y_hat)))
    return loss

loss = compute_loss(y_hat, y)
loss
```

```
array([0.88338216])
```

2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}\,))$$

4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

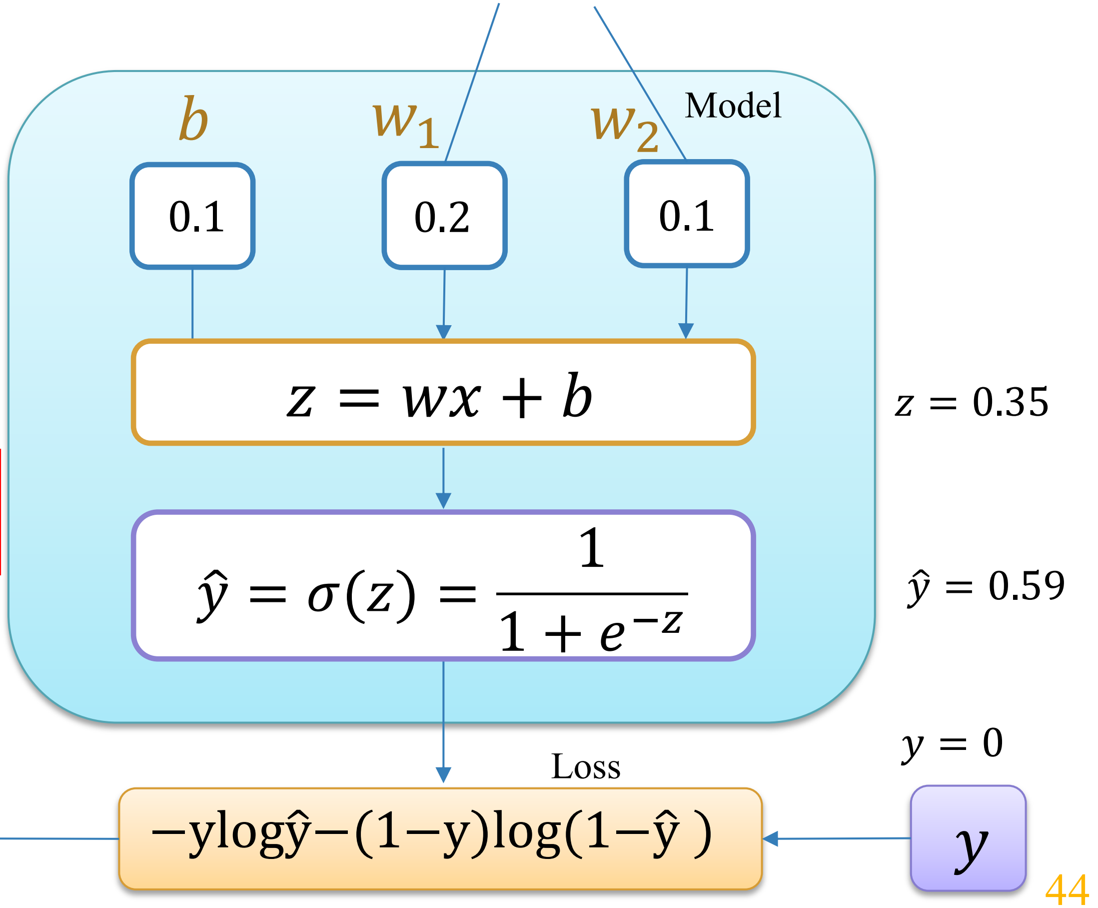
5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y) = \begin{bmatrix} 1 \\ 1.0 \\ 0.5 \end{bmatrix} [0.59] = \begin{bmatrix} 0.59 \\ 0.59 \\ 0.295 \end{bmatrix} = \begin{bmatrix} L'_b \\ L'_{w_1} \\ L'_{w_2} \end{bmatrix}$$

$\mathrm{L} = 0.883$



2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

$$\nabla_\theta L = \mathbf{x}(\hat{y} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_\theta L \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

```
# compute gradient
def compute_gradient(x, y_hat, y):
    gradient = x*(y_hat - y)
    return gradient

gradient = compute_gradient(x, y_hat, y)
gradient
```

```
array([0.58661758, 0.58661758, 0.29330879])
```

2 – Logistic Regresion

! Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

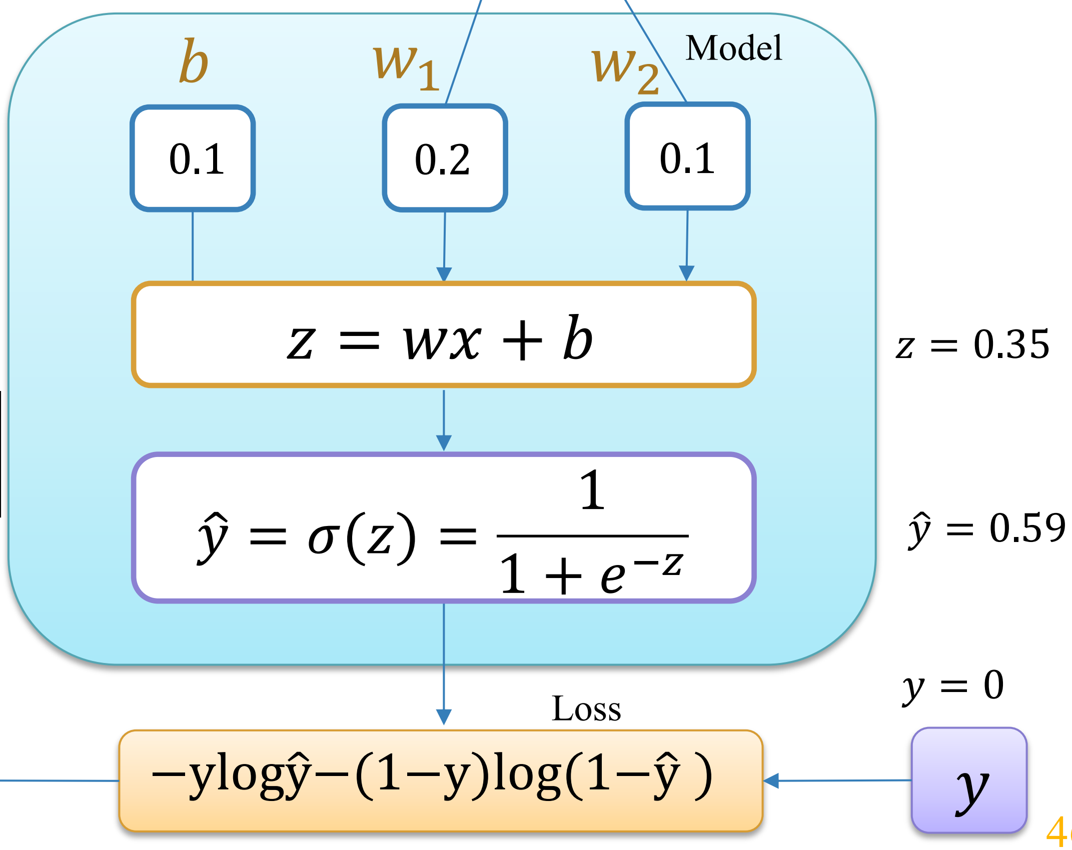
$b = 0.1 - \eta 0.59 = 0.041$

$w_1 = 0.2 - \eta 0.59 = 0.141$

$w_2 = 0.1 - \eta 0.295$
$= 0.0706$

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y) = \begin{bmatrix} 1 \\ 1.0 \\ 0.5 \end{bmatrix} [0.59] = \begin{bmatrix} 0.59 \\ 0.59 \\ 0.295 \end{bmatrix} = \begin{bmatrix} L'_b \\ L'_{w_1} \\ L'_{w_2} \end{bmatrix}$$

$\mathrm{L} = 0.883$



2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

$$\nabla_{\theta} L = \mathbf{x}(\hat{y} - y)$$

5) Update parameters

$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} L$ $\quad \eta$ is learning rate

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.1 \quad 0.2 \quad 0.1]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

```
# update weights
learning_rate = 0.1
def update_weight(gradient, theta, learning_rate):
    theta -= (learning_rate * gradient)
    return theta

theta = update_weight(gradient, theta, learning_rate)
theta
```

```
array([0.04133824, 0.14133824, 0.07066912])
```

2 – Logistic Regression

Multivariable Logistic Regression using Gradient Descent

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

4) Compute derivative

$$\nabla_{\theta} L = \mathbf{x}(\hat{y} - y)$$

5) Update parameters

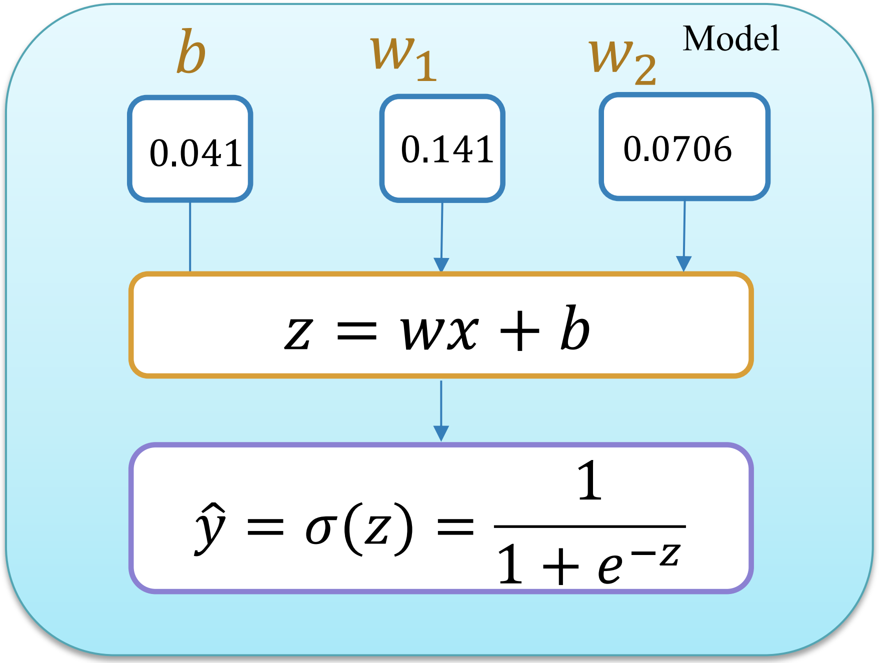
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} L \qquad \eta \text{ is learning rate}$$

$\eta = 0.1$ $\quad \boldsymbol{\theta}^T = [0.041 \quad 0.141 \quad 0.0706]$ $\quad \boldsymbol{x}^T = [1. \quad 1.0 \quad 0.5]$ $\quad \boldsymbol{y} = [0]$

$b = 0.1 - \eta 0.59 = 0.041$

$w_1 = 0.2 - \eta 0.59 = 0.141$

$w_2 = 0.1 - \eta 0.295$
$= 0.0705$



2 – Logistic Regresion

Prediction

Day	Hours	Pass
2	0.25	???
1	4.5	???

Prediction

Model

b 0.041 w_1 0.141 w_2 0.0706

$$z = wx + b = 0.3417$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} = 0.58$$

Thresholds = 0.5

y_{pred}: 1

Day	Hours	Pass
2	0.25	???
1	4.5	???

Prediction

Model

b 0.041 w_1 0.141 w_2 0.0706

$$z = wx + b = 0.5$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} = 0.622$$

Thresholds = 0.5

y_{pred}: 1

3 – Code Demo

Summary

Data #2

Day	Hours	Pass
1	0.5	0
2	1.0	0
3	1.5	1
2	2.0	0
1	2.5	0
2	3.0	1
1	3.5	1
2	4.0	1

1) Pick a sample (x, y) from training data

2) Compute output $\hat{y}$

$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

3) Compute loss

$$L(\boldsymbol{\theta}) = (-\mathrm{y}\log\hat{\mathrm{y}} - (1-\mathrm{y})\log(1-\hat{\mathrm{y}}))$$

4) Compute derivative

$$\nabla_{\theta}\mathrm{L} = \mathbf{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

Thanks!

Any questions?