Machine Learning

Softmax Regression

Nguyen Quoc Thai

CONTENT

1 – Background

Linear Regression

Data

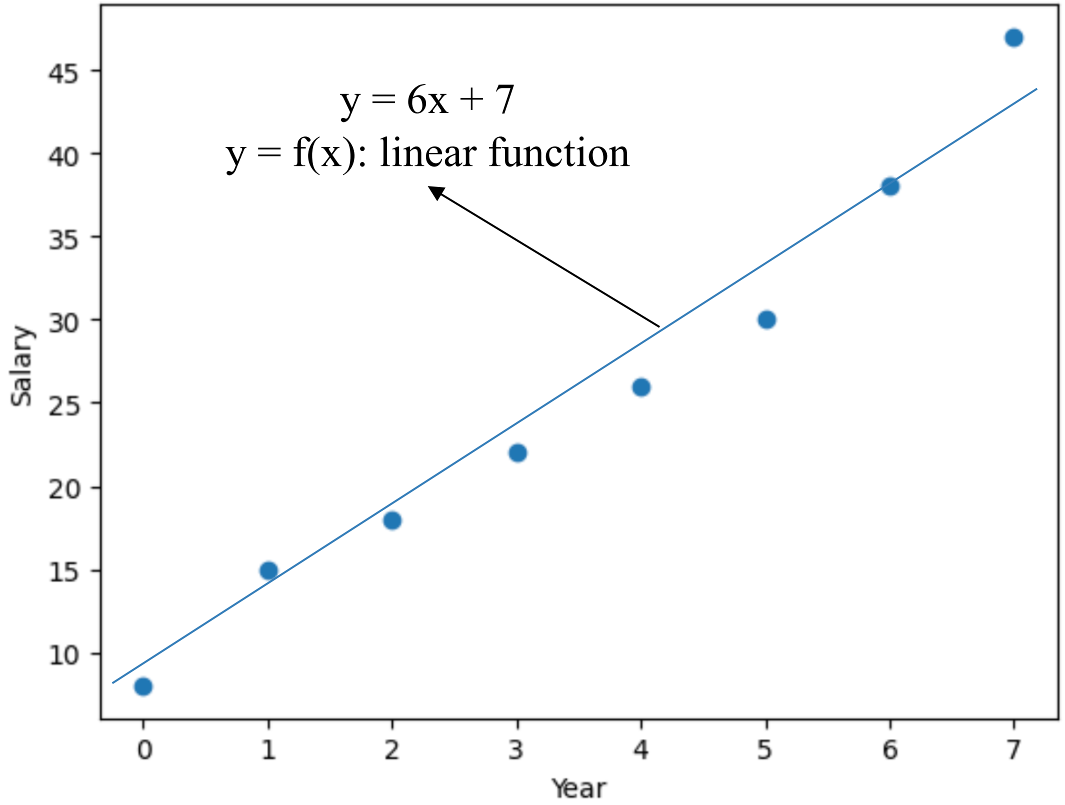
Level	Salary
0	8
1	15
2	18
3	22
4	26
5	30
6	38
7	47

Modeling

$$y = wx + b$$

Find w and b to fit the data

Visualization



1 – Background

Linear Regression

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 – Background

Logistic Regression

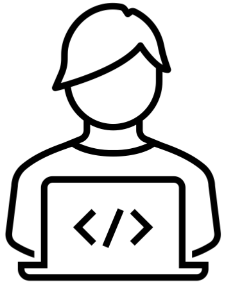
Data #1

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

Hours	Pass
0.25	???
4.5	???

Prediction

Learning



AI VIET NAM
@aivietnam.edu.vn

1 – Background

! Logistic Regression

Data #1

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

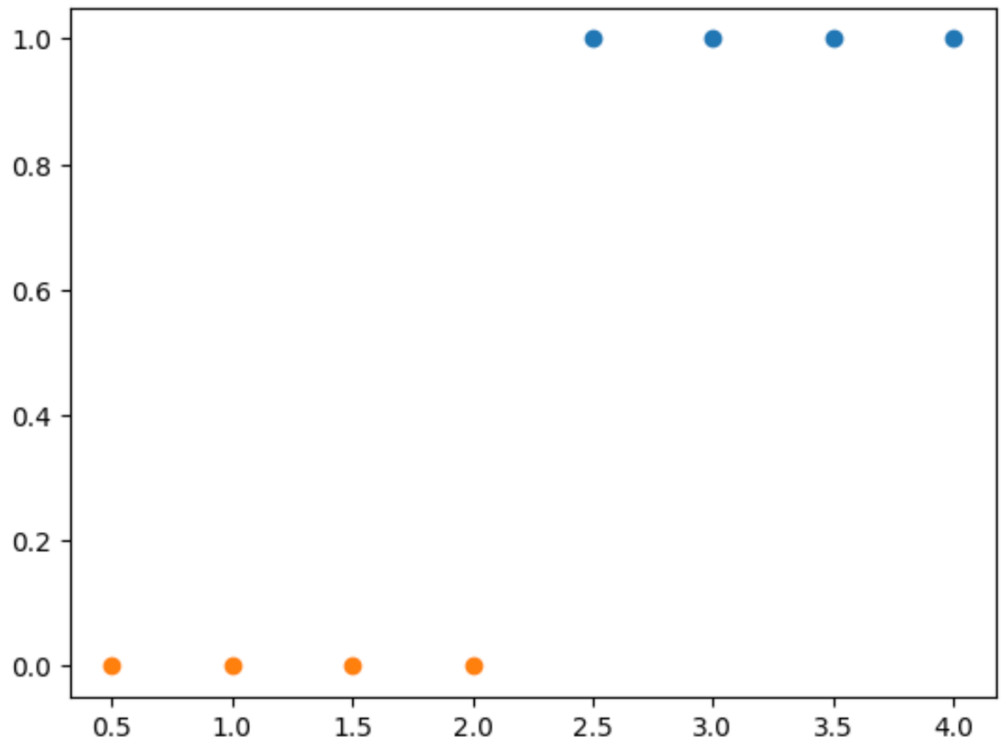
Modeling

$y = f(x)$

Find a function to fit the data

Sigmoid function

Visualization



1 – Background

Sigmoid Function

Sigmoid function

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

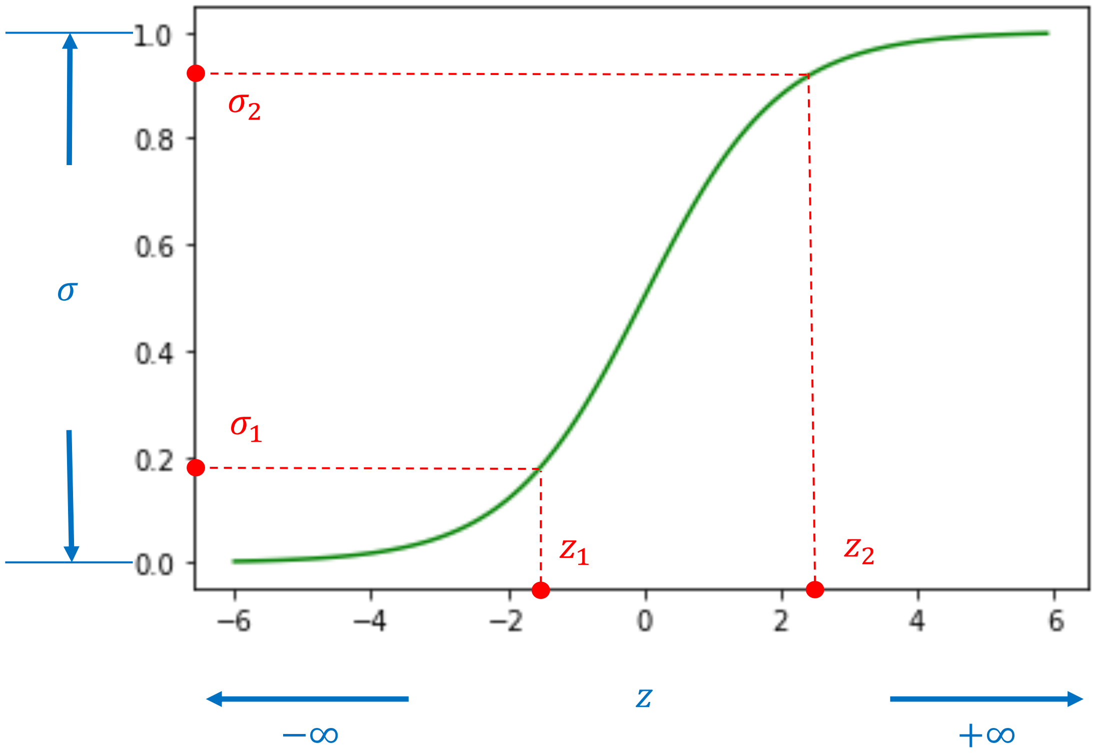
$z \in (-\infty \quad +\infty)$

$\sigma(\mathrm{z}) \in (0 \quad 1)$

Property

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

$\rightarrow \sigma(z_1) \leq \sigma(z_2)$



1 – Background

Sigmoid Function

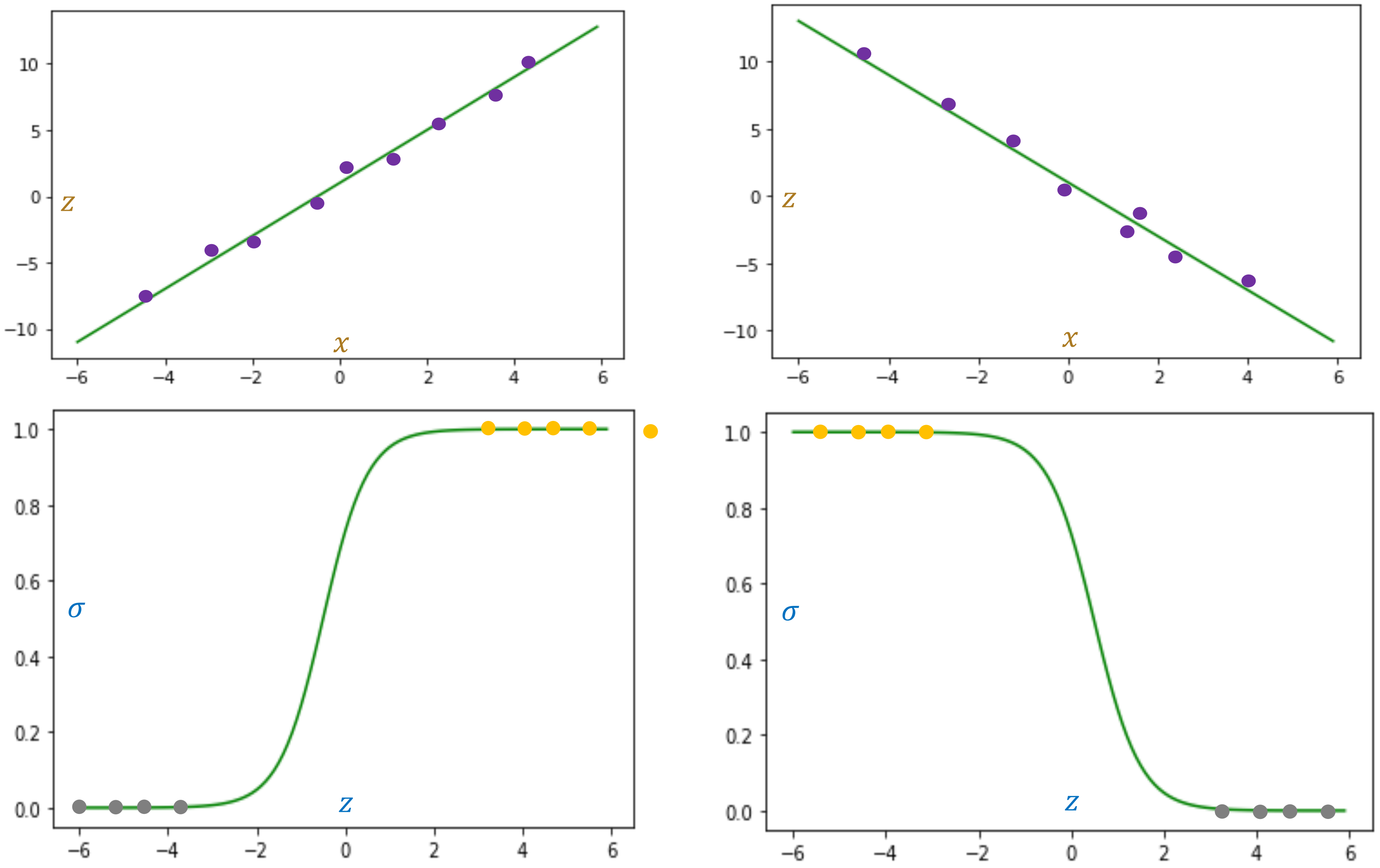
$$z = \boldsymbol{\theta}^T \boldsymbol{x}$$

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

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

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

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



AI VIET NAM
@aivietnam.edu.vn

1 – Background

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

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

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

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

$$\eta = 0.1$$

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

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

Data #1

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 – Background

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

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

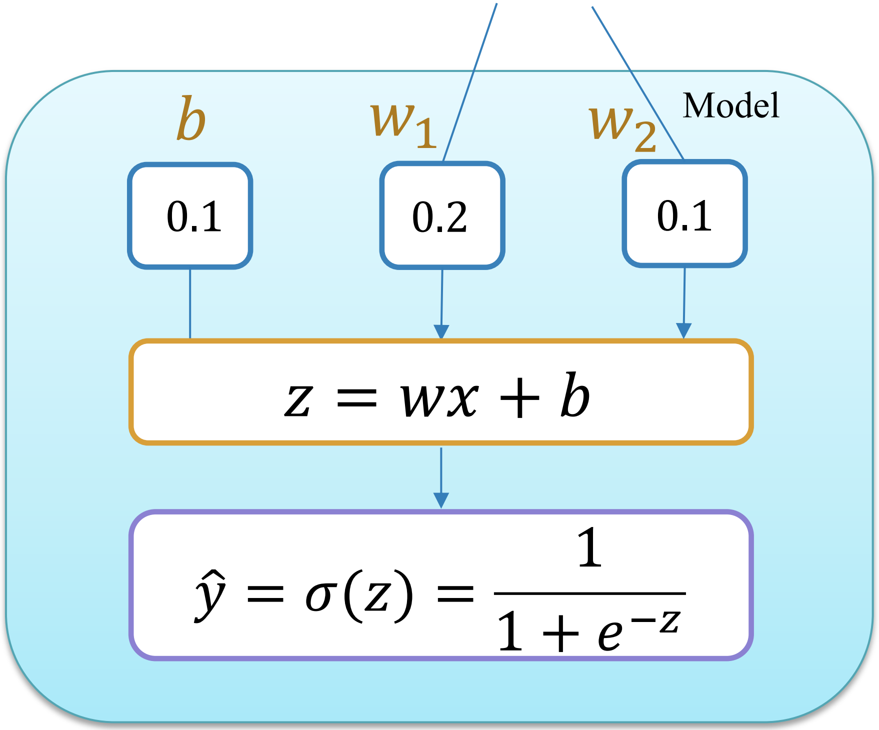
4) Compute derivative

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

5) Update parameters

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

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



$z = 0.35$

$\hat{y} = 0.59$

1 – Background

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

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

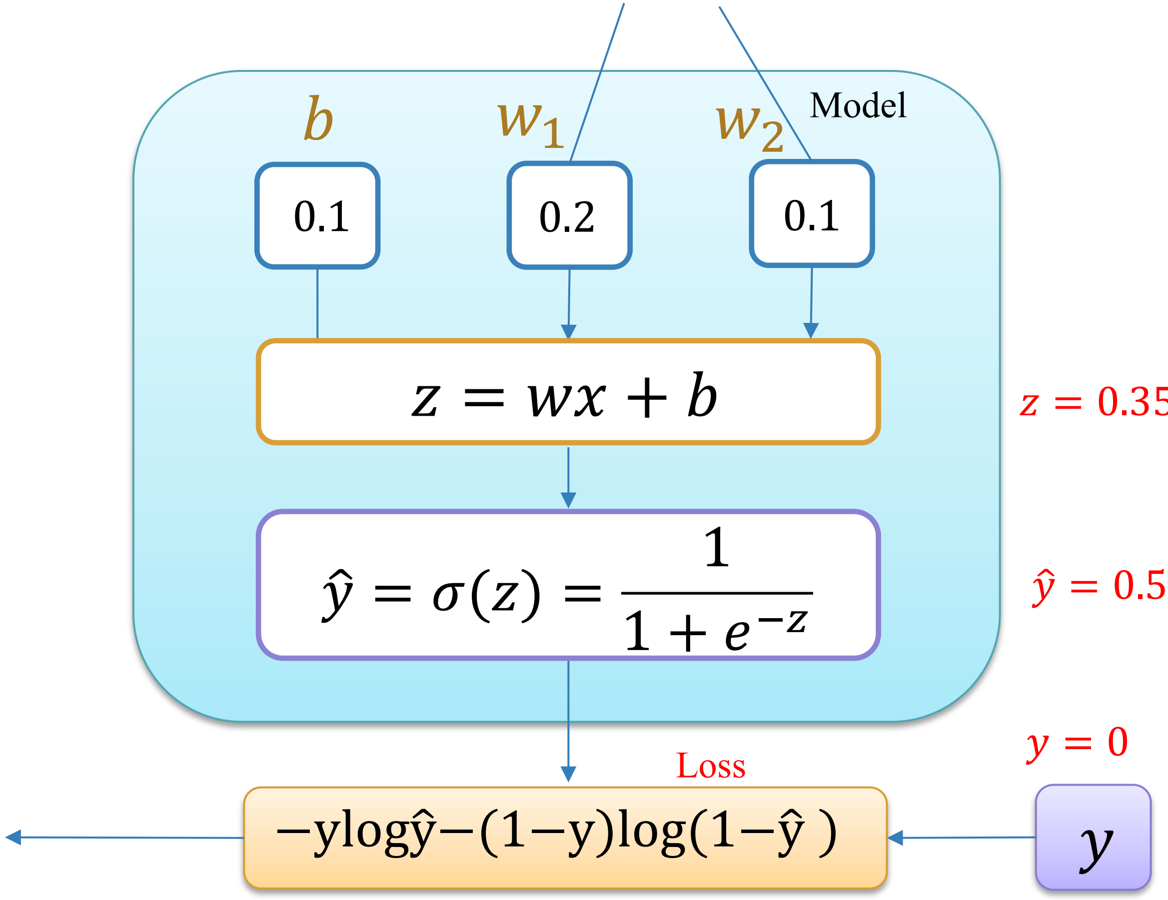
4) Compute derivative

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

5) Update parameters

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

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



$z = 0.35$

$\hat{y} = 0.59$

$y = 0$

L = 0.883

AI VIET NAM
@aivietnam.edu.vn

1 – Background

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

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

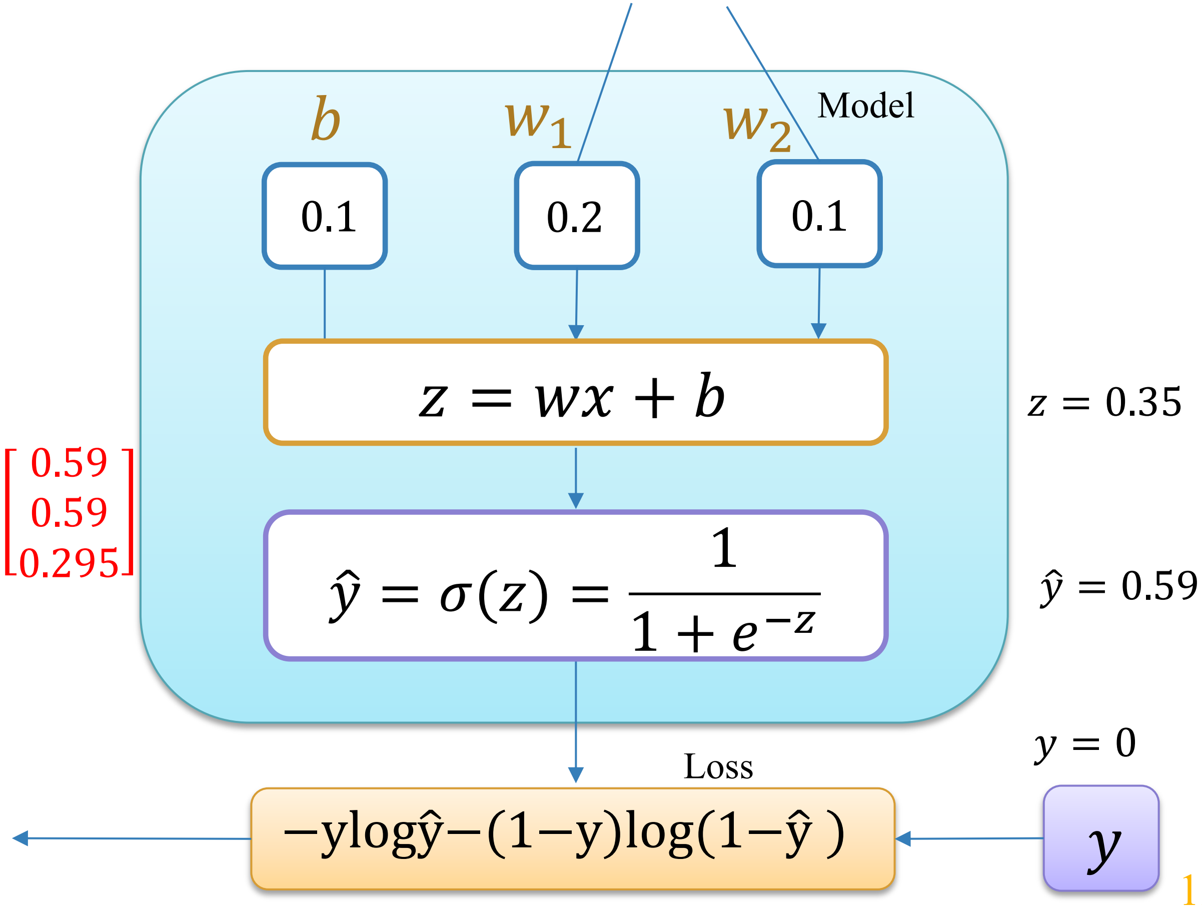
5) Update parameters

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

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

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

$\mathrm{L} = 0.883$



1 – Background

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

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

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

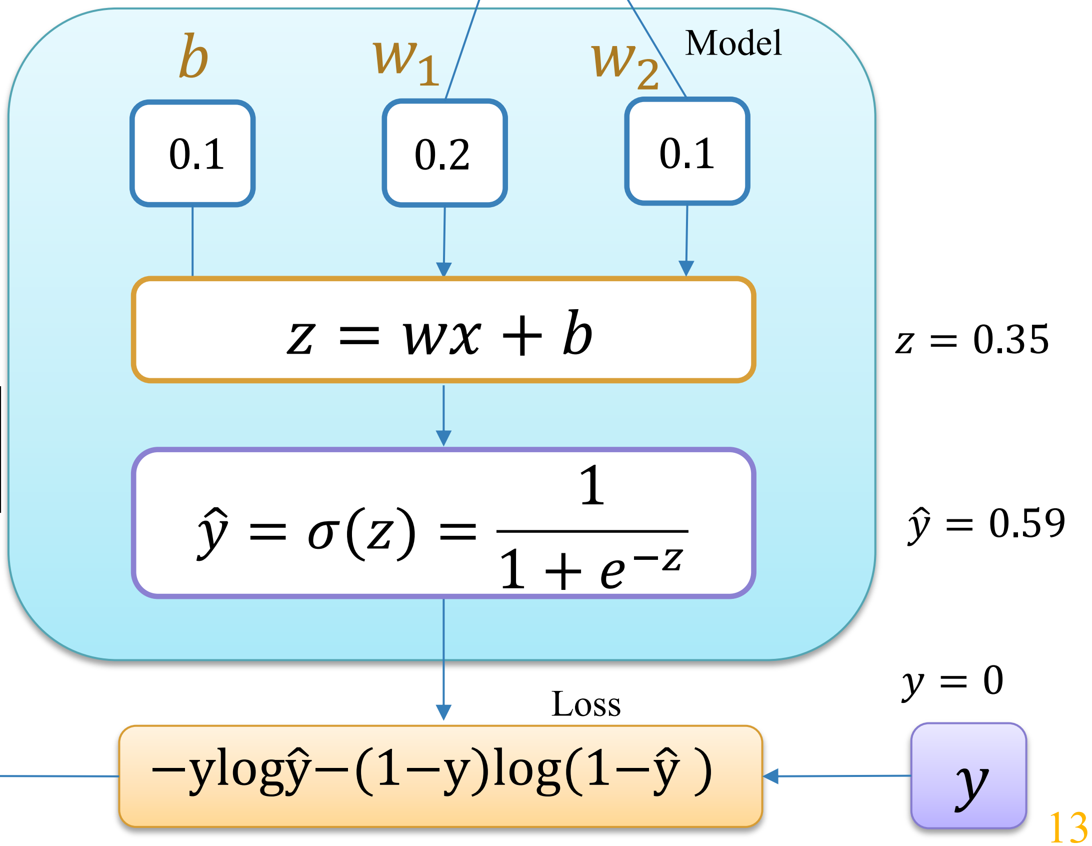
$b = 0.1 - \eta 0.59 = 0.041$

$w_1 = 0.2 - \eta 0.59 = 0.141$

$w_2 = 0.1 - \eta 0.295 = 0.0706$

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

$\mathrm{L} = 0.883$



1 – Background

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

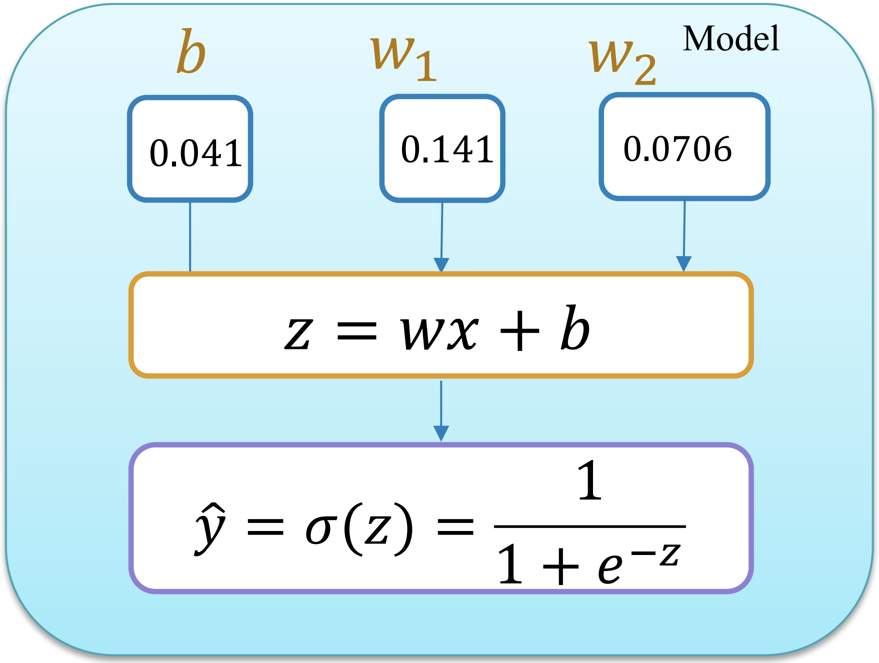
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} \qquad \eta \text{ is learning rate}$$

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

$b = 0.1 - \eta 0.59 = 0.041$

$w_1 = 0.2 - \eta 0.59 = 0.141$

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



1 – Background

Prediction

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

Prediction

Model

b 0.041 w_1 0.141 w_2 0.0706

$z = wx + b = 0.3417$

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

Threshold = 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

$z = wx + b = 0.5$

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

Threshold = 0.5

y_{pred}: 1

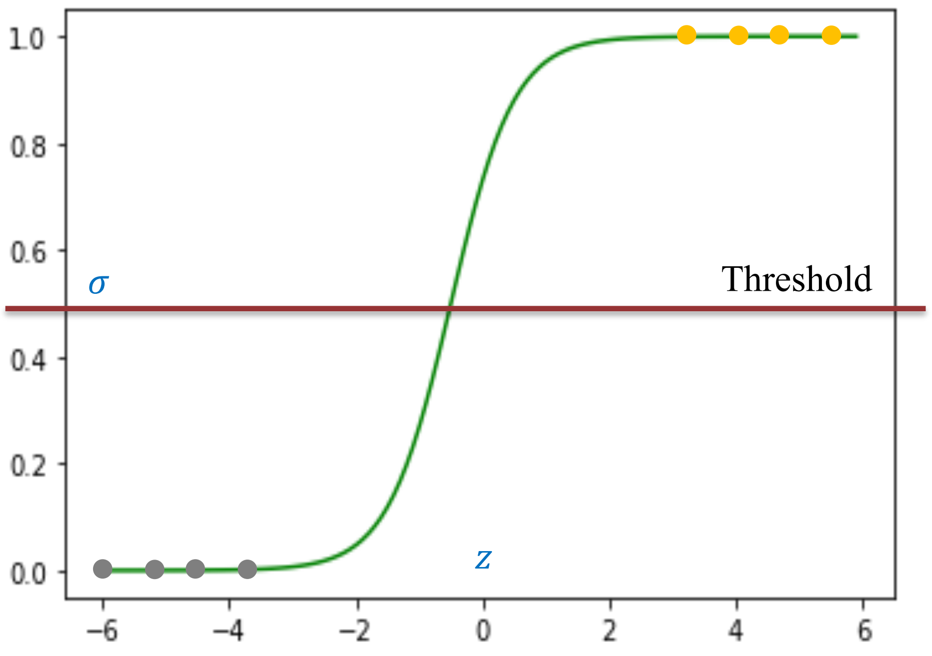
2 – Softmax Regression

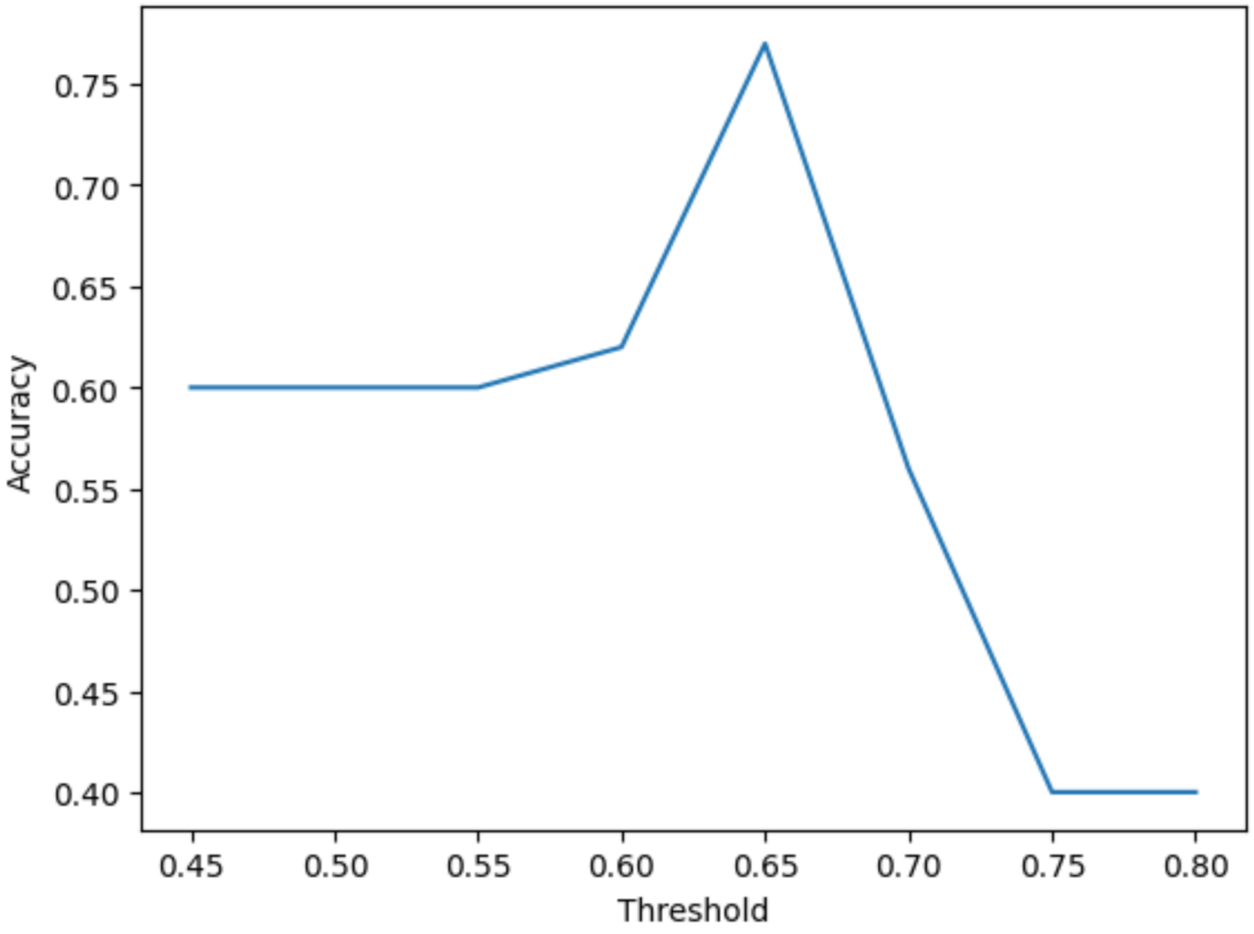
Problem

Sigmoid function

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

$z \in (-\infty \quad +\infty)$ $\qquad$ $\sigma(z) \in (0 \quad 1)$





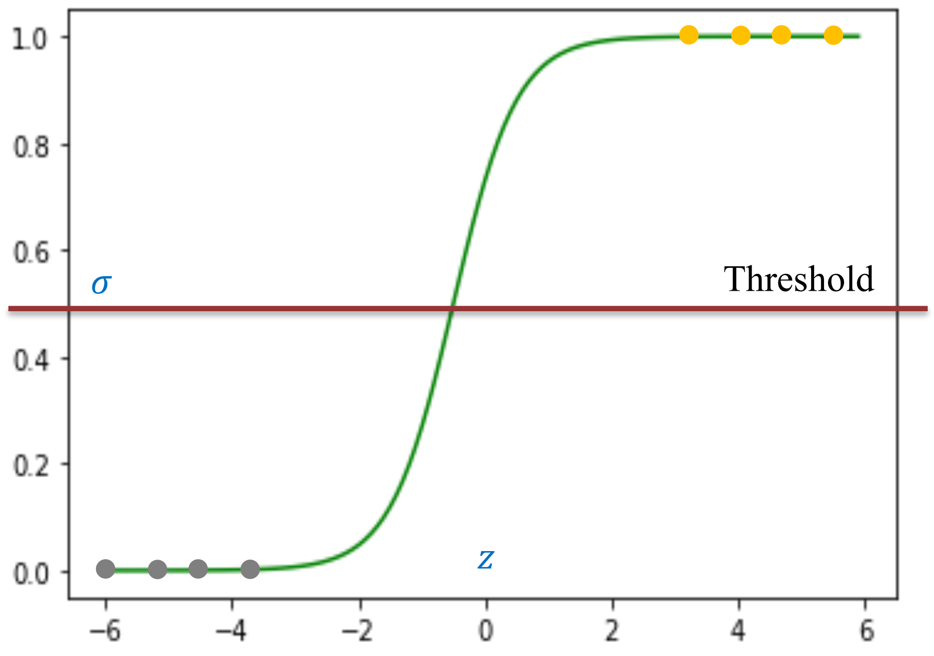
2 – Softmax Regression

Problem

Sigmoid function

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

$z \in (-\infty \quad +\infty)$ $\qquad$ $\sigma(z) \in (0 \quad 1)$



Hours	Pass
0.5	0
1.0	0
1.5	1
2.0	1

Classes: {0, 1}
Binary Classification

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

Classes: {0, 1, 2, 3}
Multi-class Classification

2 – Softmax Regression

Problem

Classes: {0, 1}
Binary Classification

Hours	Pass
0.5	0
2.0	1

Input x=0.5 → Compute z → Function $\hat{y} = f(z)$ (~~Sigmoid function~~)

→ $P(0|X) = 0.3$

→ $P(1|X) = 0.7$ → Class: 1

Softmax Function

Classes: {0, 1, 2, 3}
Multi-class Classification

Hours	Score
0.5	0
1.5	1
3.0	2
4.0	3

Input x=0.5 → Compute z → Function $\hat{y} = f(z)$

→ $P(0|X) = 0.3$

→ $P(1|X) = 0.2$

→ $P(2|X) = 0.4$ → Class: 2

→ $P(3|X) = 0.1$

2 – Softmax Regression

Softmax Function

$$P_i = f(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

$$0 \leq f(z_i) \leq 1$$

$$\sum_i f(z_i) = 1$$

Input		Probability
$z_1 = 1.0$	Softmax	$f(z_1) = 0.12$
$z_2 = 3.0$		$f(z_2) = 0.88$

Input		Probability
$z_1 = 1.0$	Softmax	$f(z_1) = 0.09$
$z_2 = 2.0$		$f(z_2) = 0.24$
$z_3 = 3.0$		$f(z_3) = 0.67$

2 – Softmax Regression

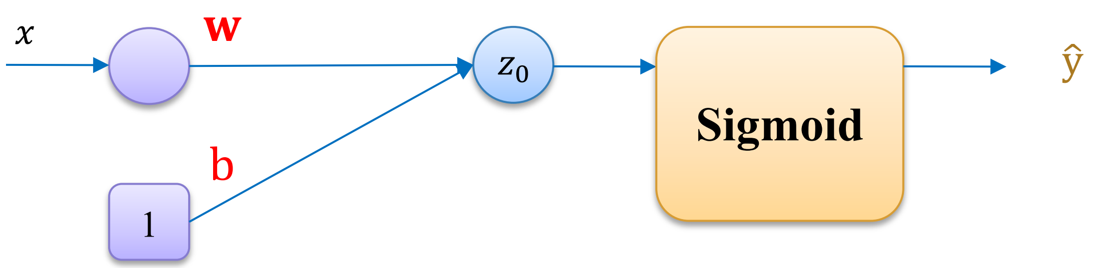
Parameters

Classes: {0, 1}
Binary Classification

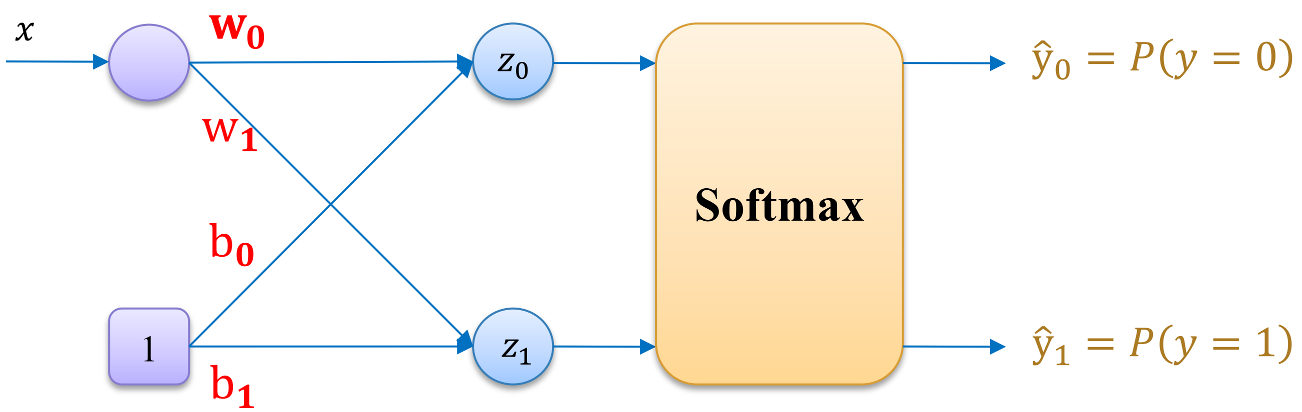
Hours	Pass
0.5	0
2.0	1

#feature: 1
#class: 2

Logistic Regression



Softmax Regression



2 – Softmax Regression

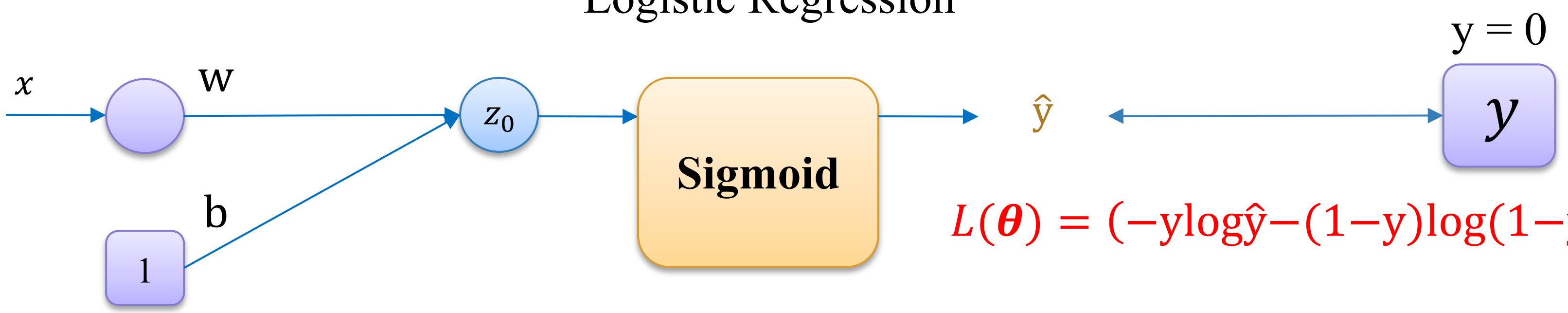
Loss Function

Classes: {0, 1}
Binary Classification

Hours	Pass
0.5	0
2.0	1

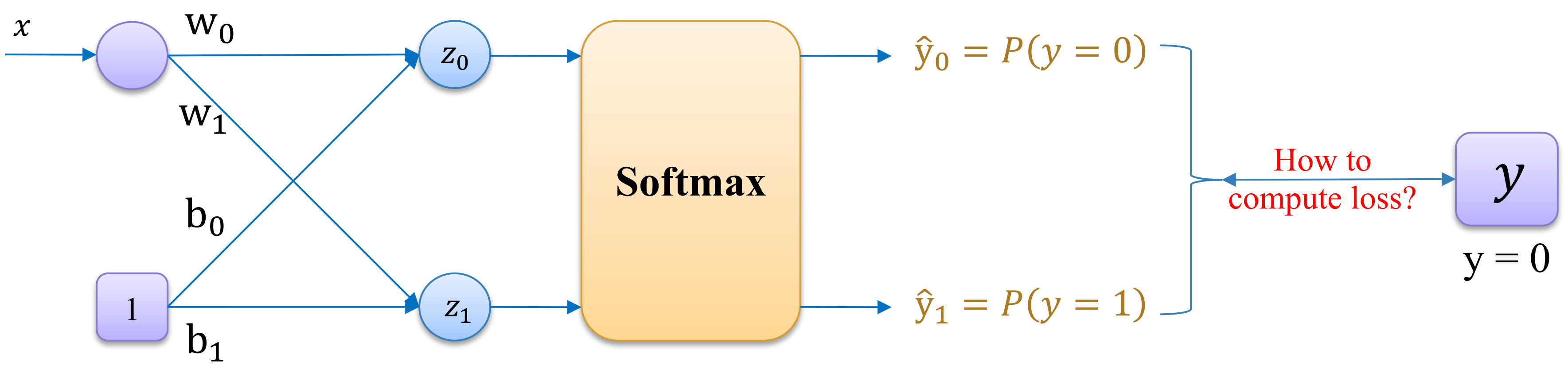
#feature: 1
#class: 2

Logistic Regression



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

Softmax Regression



2 – Softmax Regression

One-Hot Encoding

$$\boldsymbol{y} = \begin{bmatrix} y_0 \\ \dots \\ y_C \end{bmatrix} \qquad y_i \in \{0,1\} \qquad \sum_i y_i = 1 \qquad \text{C} = \#\text{classes}$$

Classes: {0, 1}
Binary Classification

Hours	Pass
0.5	0
2.0	1

#feature: 1
#class: 2

$$y = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$y = 1 \rightarrow \boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Classes: {0, 1, 2}
Multi-class Classification

Hours	Score
0.5	0
1.5	1
3.0	2

#feature: 1
#class: 3

$$y = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$y = 1 \rightarrow \boldsymbol{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$y = 2 \rightarrow \boldsymbol{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

2 – Softmax Regression

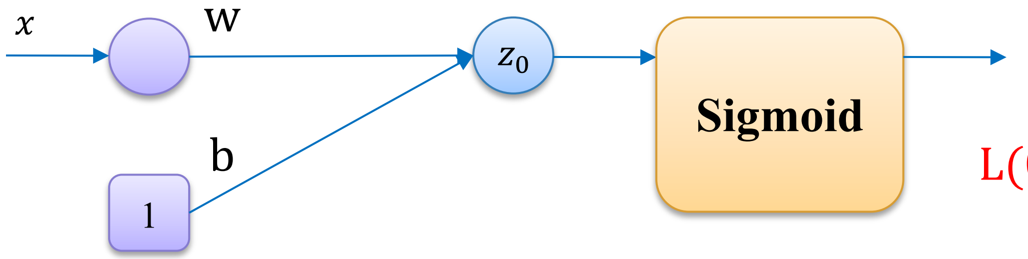
Loss Function

Classes: {0, 1}
Binary Classification

Hours	Pass
0.5	0
2.0	1

#feature: 1
#class: 2

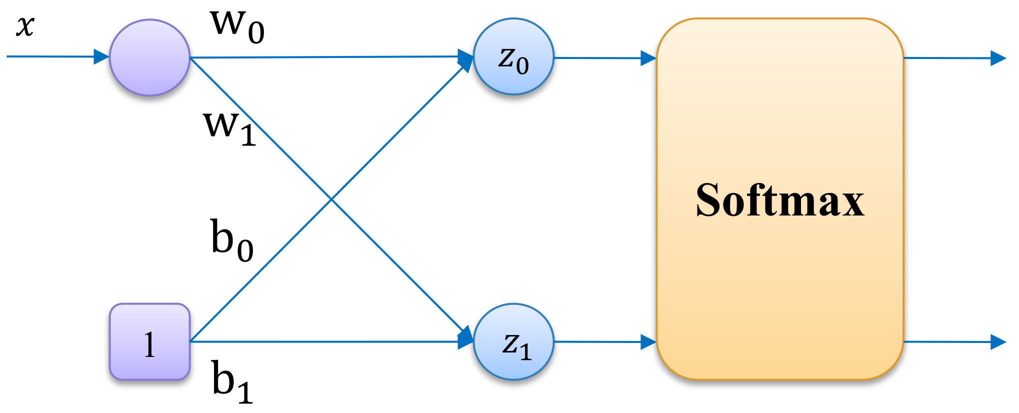
Logistic Regression



$\hat{y}$ ⟷ y y = 0

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

Softmax Regression



$\hat{y}_0 = P(y = 0)$

$\hat{y}_1 = P(y = 1)$

$$L(\boldsymbol{\theta}) = -y_1\log(\hat{y}_1) - y_0\log(\hat{y}_0)$$
$$= -\sum_i y_i \log(\hat{y}_i)$$

$y_0 = 1$
$y_1 = 0$

One-Hot Encoding

y

y = 0

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1]e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

Data #1

Hours	Pass
0.5	0
1.0	0
1.5	1
2.0	1

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

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

One-hot encoding for label

$$y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$$

$$y = 1 \rightarrow \boldsymbol{y}^T = [0 \quad 1]$$

$$\theta = \begin{bmatrix} \mathrm{b}_0 & \mathrm{b}_1 \\ \mathrm{w}_0 & \mathrm{w}_1 \end{bmatrix} \qquad \theta = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

$$\eta = 0.1$$

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \;\dots\; 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \oslash \boldsymbol{d}$$

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

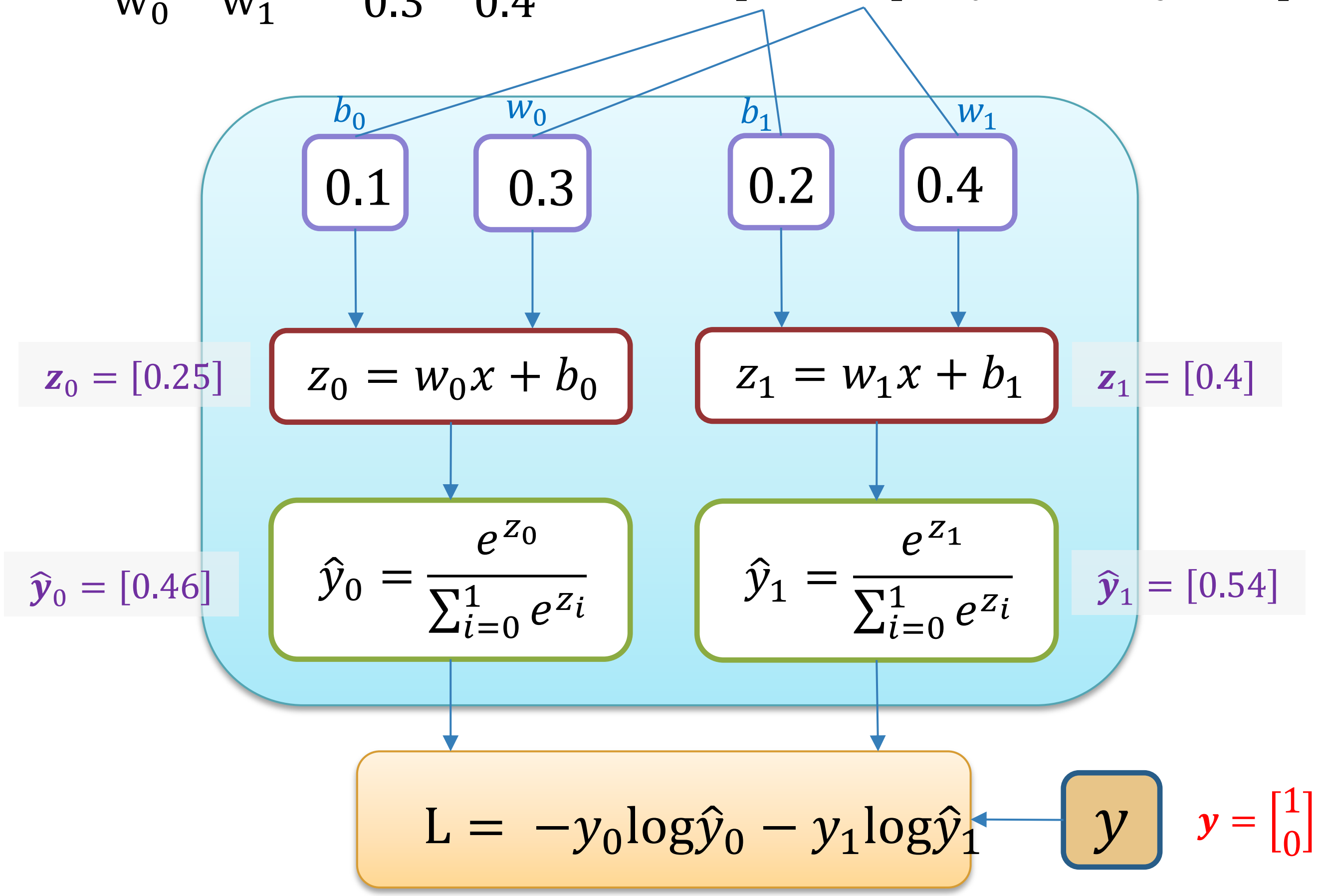
$$\nabla_{\theta} \mathrm{L} = \boldsymbol{x}(\hat{\boldsymbol{y}} - \boldsymbol{y})^T$$

5) Update parameters

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

η is learning rate

$\eta = 0.1 \quad \theta = \begin{bmatrix} b_0 & b_1 \\ w_0 & w_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 0.5] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$



2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \ldots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

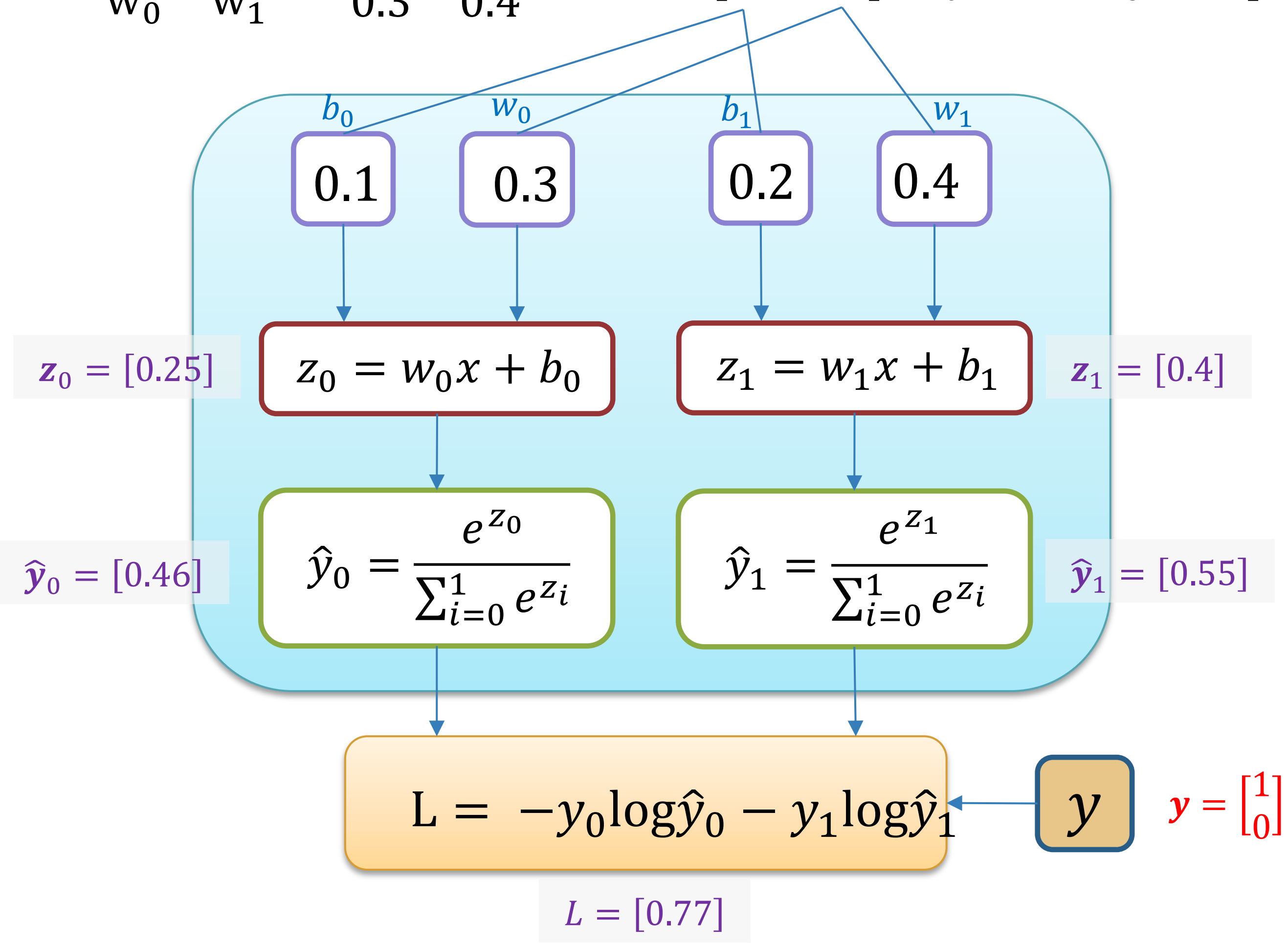
$$\nabla_{\theta} \mathrm{L} = \boldsymbol{x}(\hat{\boldsymbol{y}} - \boldsymbol{y})^T$$

5) Update parameters

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

η is learning rate

$\eta = 0.1 \quad \theta = \begin{bmatrix} b_0 & b_1 \\ w_0 & w_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 0.5] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$



AI VIET NAM
@aivietnam.edu.vn

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \ldots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \oslash \boldsymbol{d}$$

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

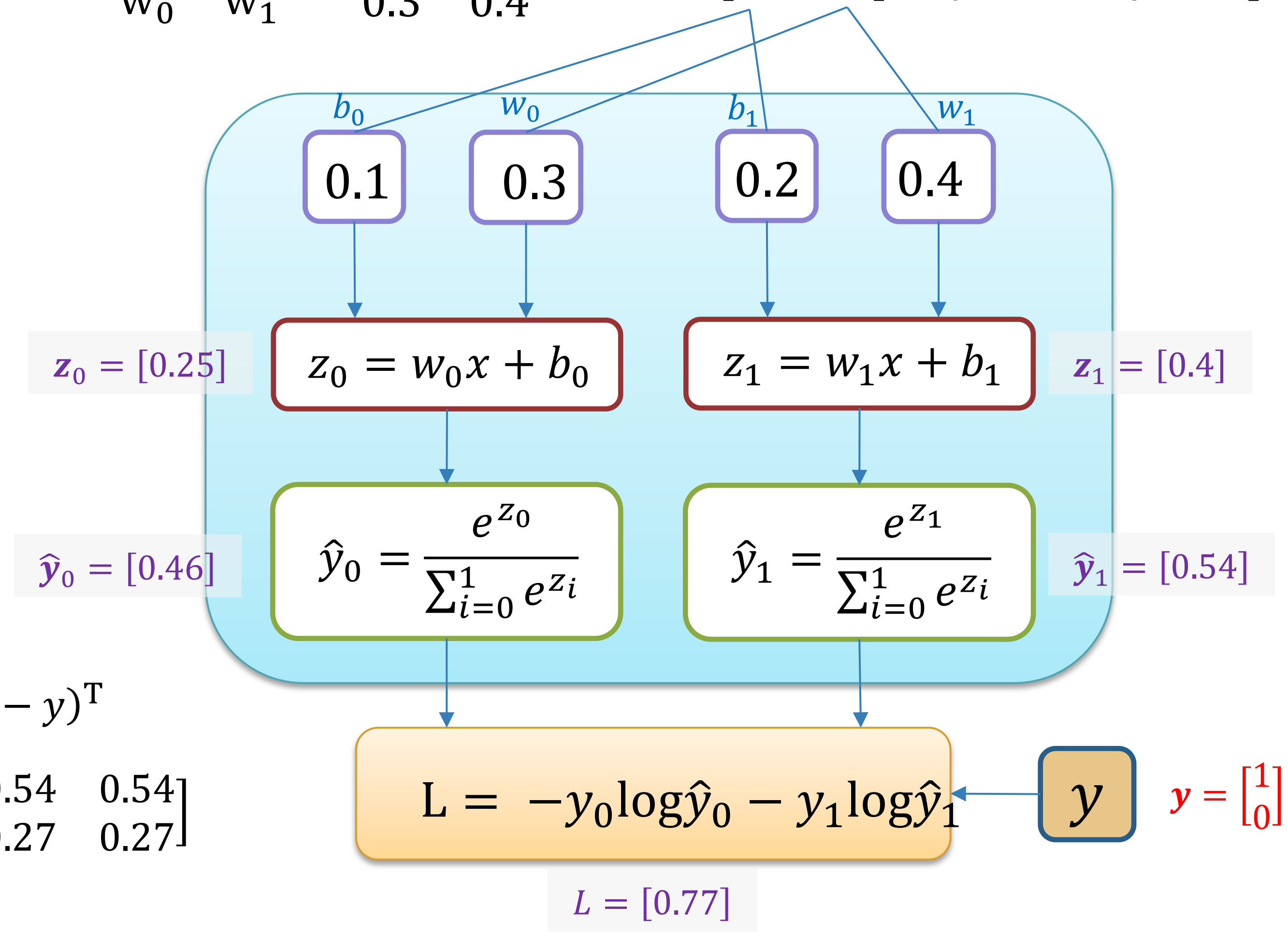
$$\nabla_{\theta} \mathrm{L} = \boldsymbol{x}(\hat{\boldsymbol{y}} - \boldsymbol{y})^T$$

5) Update parameters

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

η is learning rate

$$\eta = 0.1 \quad \theta = \begin{bmatrix} b_0 & b_1 \\ w_0 & w_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 0.5] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$$



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

$$= \begin{bmatrix} -0.54 & 0.54 \\ -0.27 & 0.27 \end{bmatrix}$$

AI VIET NAM
@aivietnam.edu.vn

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \oslash \boldsymbol{d}$$

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

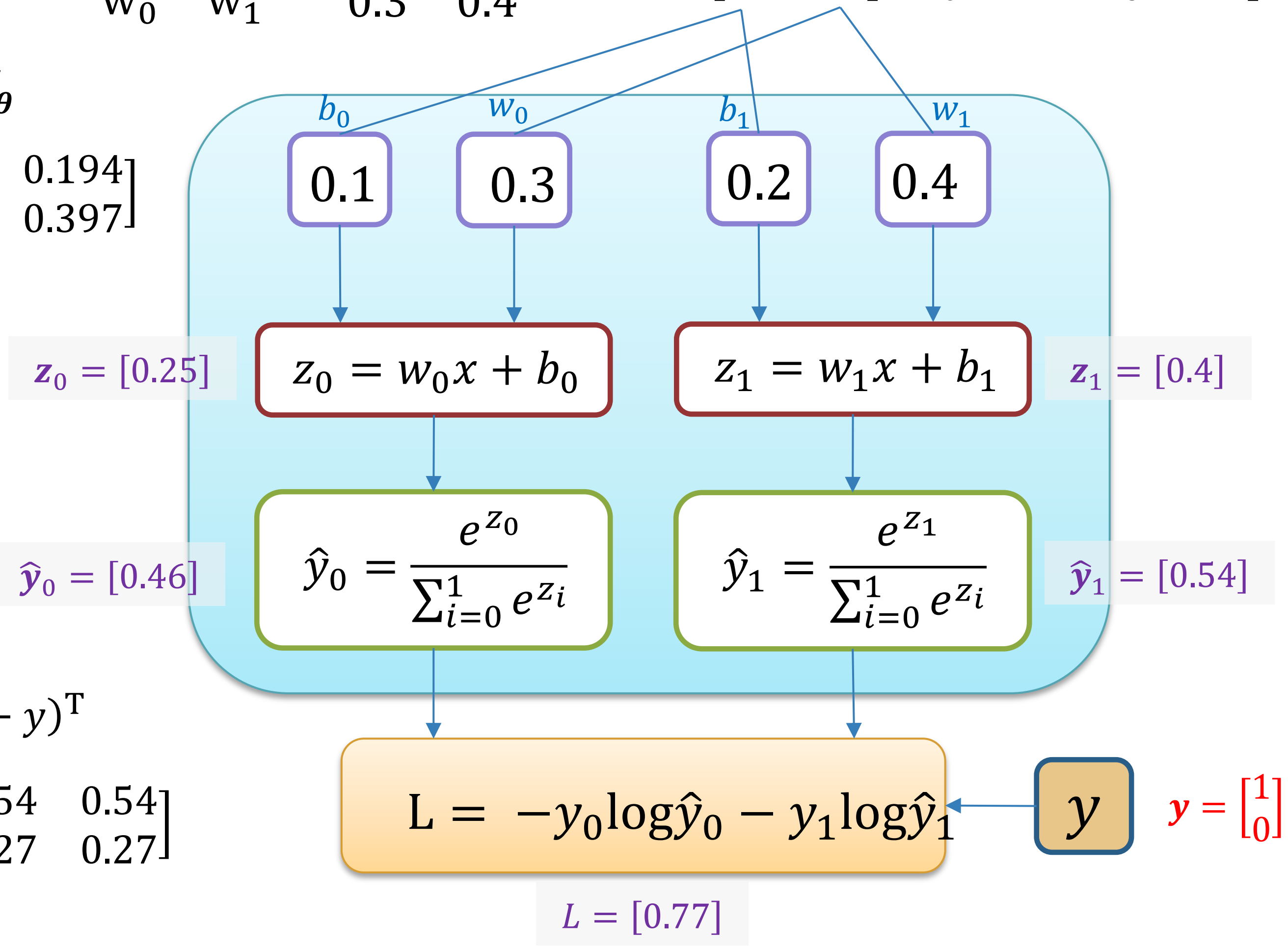
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$$

η is learning rate

$\eta = 0.1 \quad \theta = \begin{bmatrix} \mathrm{b}_0 & \mathrm{b}_1 \\ \mathrm{w}_0 & \mathrm{w}_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 0.5] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

$$= \begin{bmatrix} 0.105 & 0.194 \\ 0.302 & 0.397 \end{bmatrix}$$



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

$$= \begin{bmatrix} -0.54 & 0.54 \\ -0.27 & 0.27 \end{bmatrix}$$

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

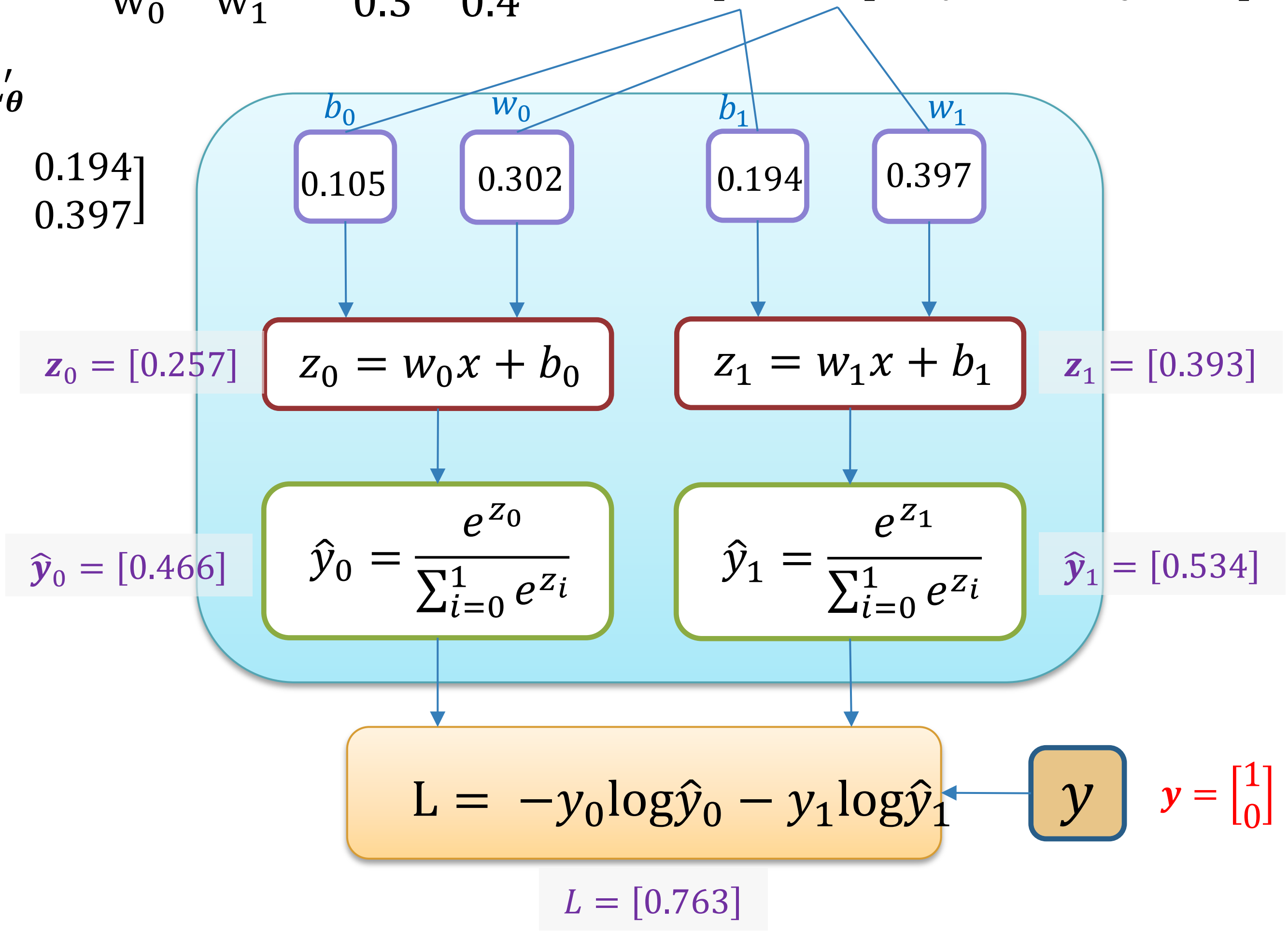
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_\theta \mathrm{L}$$

η is learning rate

$\eta = 0.1 \quad \theta = \begin{bmatrix} b_0 & b_1 \\ w_0 & w_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 0.5] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

$$= \begin{bmatrix} 0.105 & 0.194 \\ 0.302 & 0.397 \end{bmatrix}$$



2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \ldots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \oslash \boldsymbol{d}$$

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

Data #1

Hours	Pass
0.5	0
1.0	0
1.5	1
2.0	1

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

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

One-hot encoding for label

$$y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$$

$$y = 1 \rightarrow \boldsymbol{y}^T = [0 \quad 1]$$

$$\theta = \begin{bmatrix} \mathrm{b}_0 & \mathrm{b}_1 \\ \mathrm{w}_0 & \mathrm{w}_1 \end{bmatrix} \qquad \theta = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

$$\eta = 0.1$$

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \oslash \boldsymbol{d}$$

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

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

```
def convert_one_hot(y, k):
    one_hot = np.zeros((len(y), k))
    one_hot[np.arange(len(y)), y] = 1
    return one_hot

n_classes = 2
Y_one_hot = convert_one_hot(Y, n_classes)
Y_one_hot
```

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

```
y = Y_one_hot[0]
```

```
x, y
```

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

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

```
theta = np.array([[0.1, 0.2], [0.3, 0.4]])
theta
```

```
array([[0.1, 0.2],
       [0.3, 0.4]])
```

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \ldots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

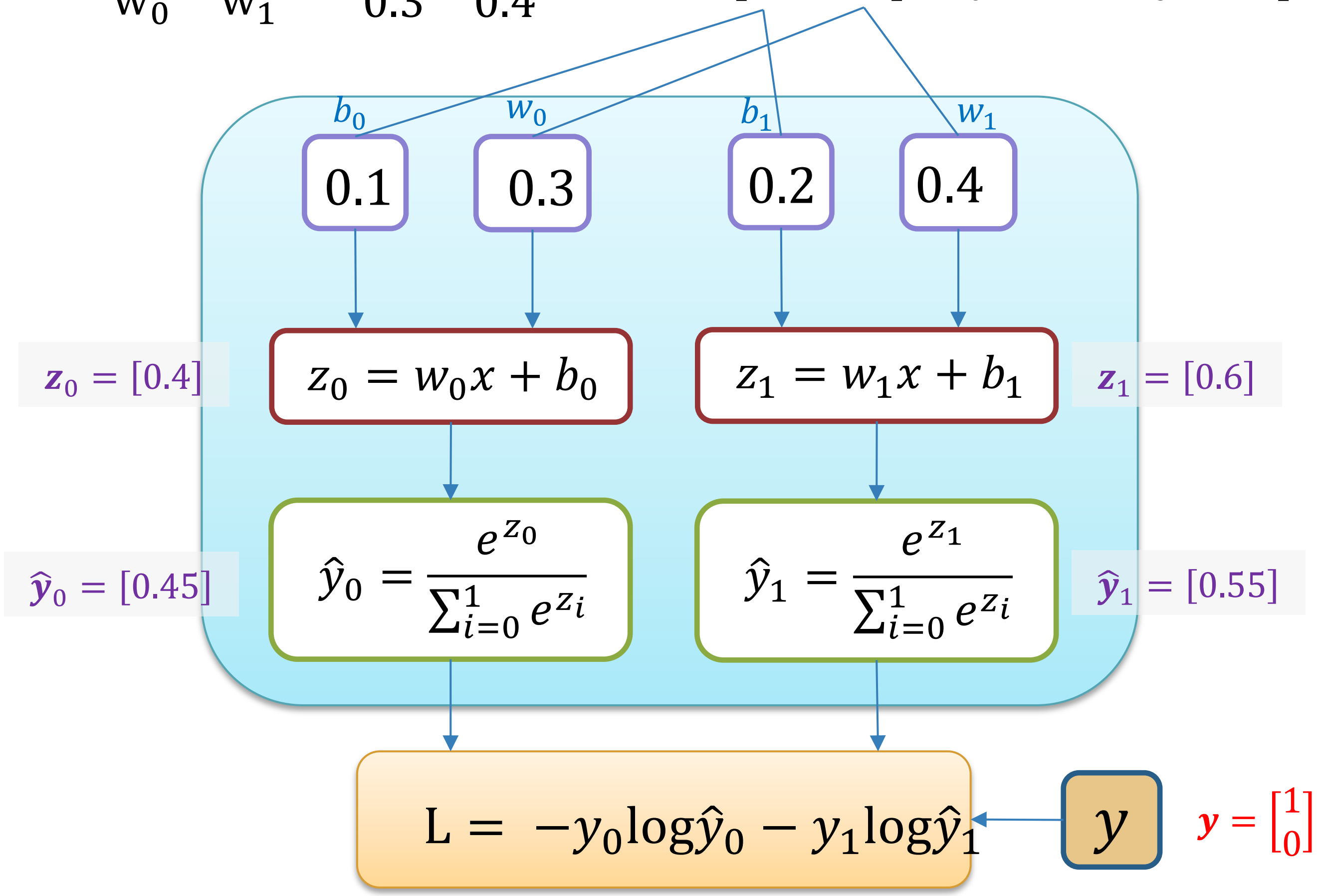
$$\nabla_{\theta} \mathrm{L} = \boldsymbol{x}(\hat{\boldsymbol{y}} - \boldsymbol{y})^T$$

5) Update parameters

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

η is learning rate

$\eta = 0.1 \quad \theta = \begin{bmatrix} b_0 & b_1 \\ w_0 & w_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 1.0] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$



2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1]e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

```
# define softmax function
def softmax_function(z):
    return np.exp(z) / np.sum(np.exp(z))
```

```
# compute y_hat
def predict(x, theta):
    z = np.dot(x , theta)
    y_hat = np.exp(z) / np.sum(np.exp(z))
    return z, y_hat

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

```
(array([[0.4, 0.6]]), array([[0.450166, 0.549834]]))
```

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \oslash \boldsymbol{d}$$

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

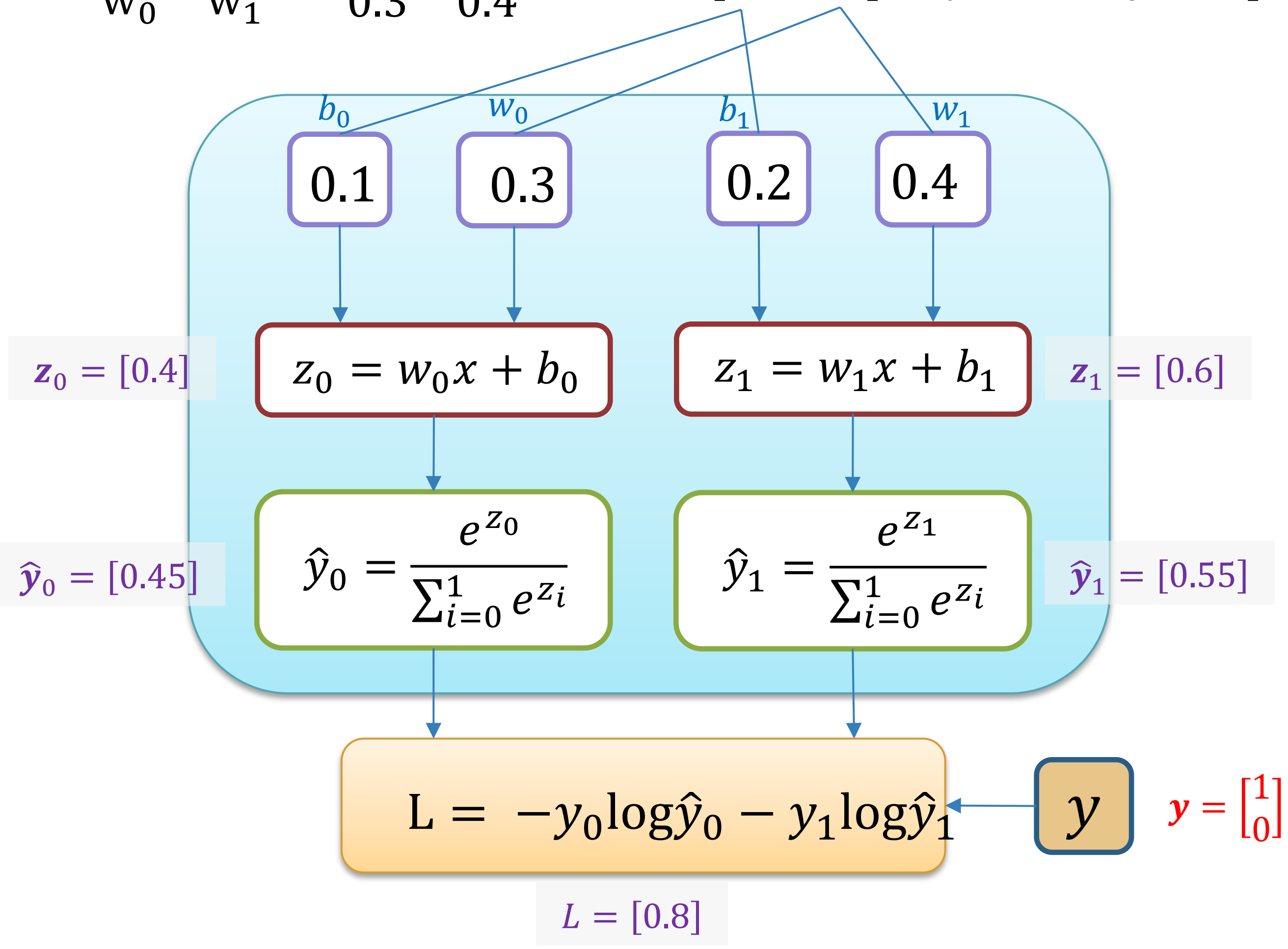
$$\nabla_{\theta} \mathrm{L} = \boldsymbol{x}(\hat{\boldsymbol{y}} - \boldsymbol{y})^T$$

5) Update parameters

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

η is learning rate

$$\eta = 0.1 \quad \theta = \begin{bmatrix} \mathrm{b}_0 & \mathrm{b}_1 \\ \mathrm{w}_0 & \mathrm{w}_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 1.0] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$$



2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \oslash \boldsymbol{d}$$

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

```
# compute loss
def compute_loss(y_hat, y):
    loss = -np.log(np.sum(y_hat*y, axis=1))
    return loss

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

```
array([0.79813887])
```

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1]e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

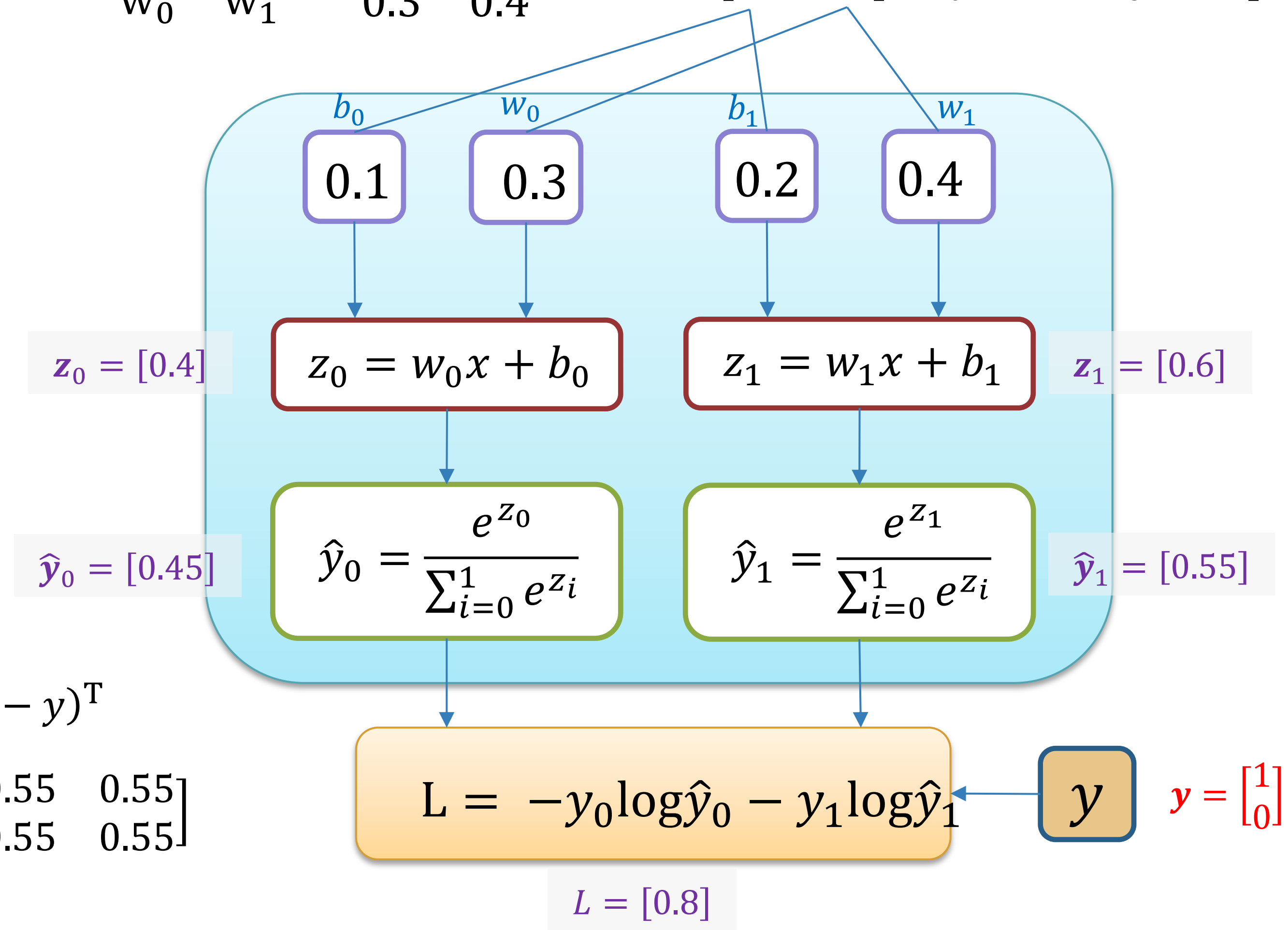
$$\nabla_\theta \mathrm{L} = \boldsymbol{x}(\hat{\mathbf{y}} - \boldsymbol{y})^T$$

5) Update parameters

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

η is learning rate

$\eta = 0.1 \quad \theta = \begin{bmatrix} \mathrm{b}_0 & \mathrm{b}_1 \\ \mathrm{w}_0 & \mathrm{w}_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 1.0] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$



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

$$= \begin{bmatrix} -0.55 & 0.55 \\ -0.55 & 0.55 \end{bmatrix}$$

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \oslash \boldsymbol{d}$$

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

```python
# compute gradient
def compute_gradient(y_hat, y, x):
    gradient = np.dot(x.transpose(), (y_hat - y))
    return gradient

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

```
array([[-0.549834,  0.549834],
       [-0.549834,  0.549834]])
```

AI VIET NAM
@aivietnam.edu.vn

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \ldots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

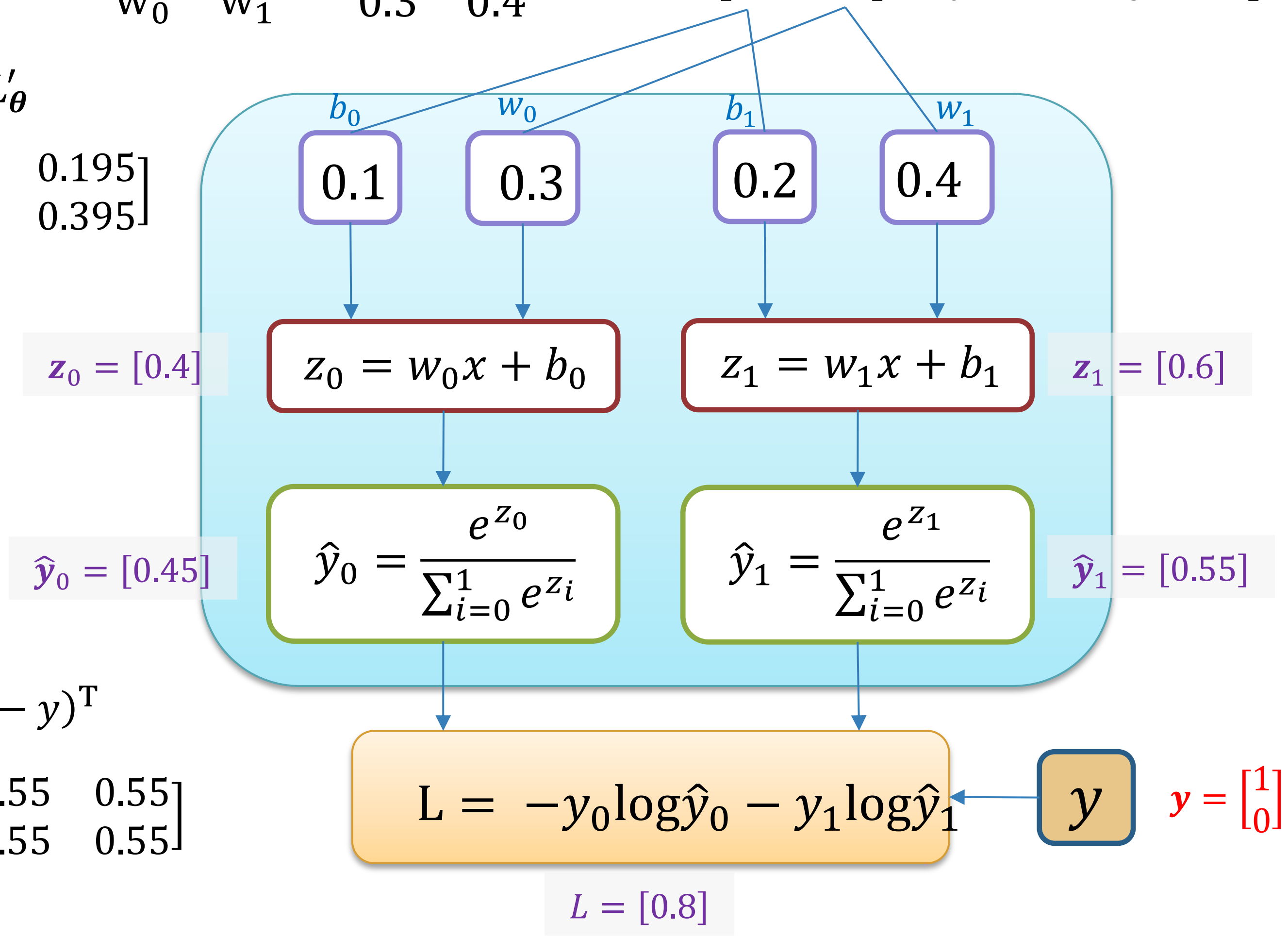
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_{\theta} \mathrm{L}$$

η is learning rate

$\eta = 0.1 \quad \theta = \begin{bmatrix} b_0 & b_1 \\ w_0 & w_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} \quad \boldsymbol{x}^T = [1 \quad 1.0] \quad y = 0 \rightarrow \boldsymbol{y}^T = [1 \quad 0]$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta L'_{\boldsymbol{\theta}}$$

$$= \begin{bmatrix} 0.105 & 0.195 \\ 0.305 & 0.395 \end{bmatrix}$$



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

$$= \begin{bmatrix} -0.55 & 0.55 \\ -0.55 & 0.55 \end{bmatrix}$$

AI VIET NAM
@aivietnam.edu.vn

2 – Softmax Regression

Softmax Regression

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

$$\boldsymbol{d} = [1 \ \dots 1] e^{\boldsymbol{z}}$$

$$\hat{\boldsymbol{y}} = e^{\boldsymbol{z}} \emptyset \boldsymbol{d}$$

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

```python
# update weights
learning_rate = 0.01

def update_weight(theta, gradient, learning_rate):
    theta -= (learning_rate * gradient)
    return theta

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

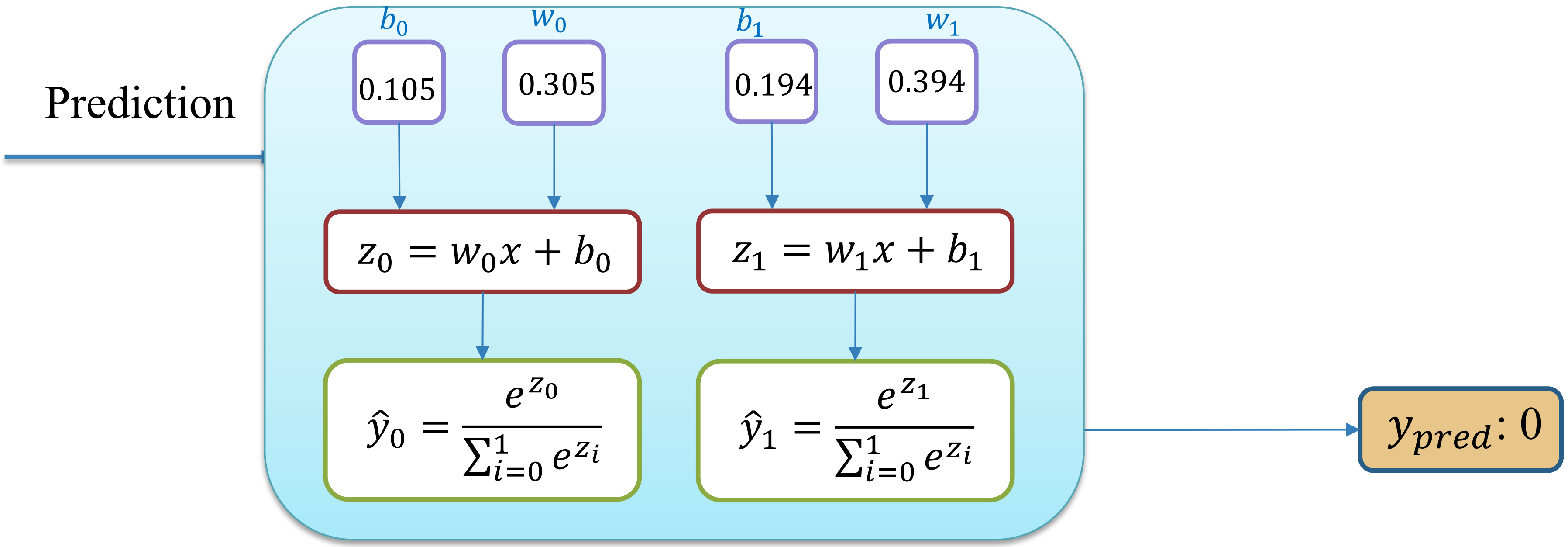
```
array([[0.10549834, 0.19450166],
       [0.30549834, 0.39450166]])
```

AI VIET NAM
@aivietnam.edu.vn

2 – Softmax Regression

Prediction

Hours	Pass
0.25	???
4.5	???



2 – Softmax Regression

Prediction

```
x_test = np.array([1.0, 0.25])
z, y_hat = predict(x_test, theta)
pred = np.argmax(y_hat)
pred
```

Thanks!

Any questions?