Module 05 – Extra Class

SOFTMAX REGRESSION

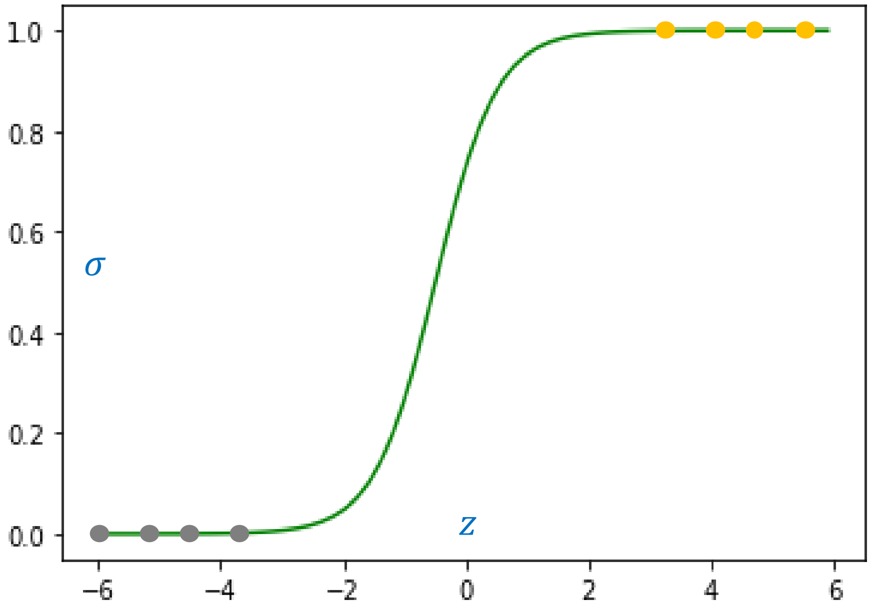
Nguyen Quoc Thai

Code - Data

Objectives

Logistic Regression (Review)

- ❖ Logistic Regression
- ❖ Sigmoid Function
- ❖ Gradient Descent



Softmax Regression

- ❖ Softmax Regression
- ❖ Softmax Function
- ❖ One Sample
- ❖ N Sample

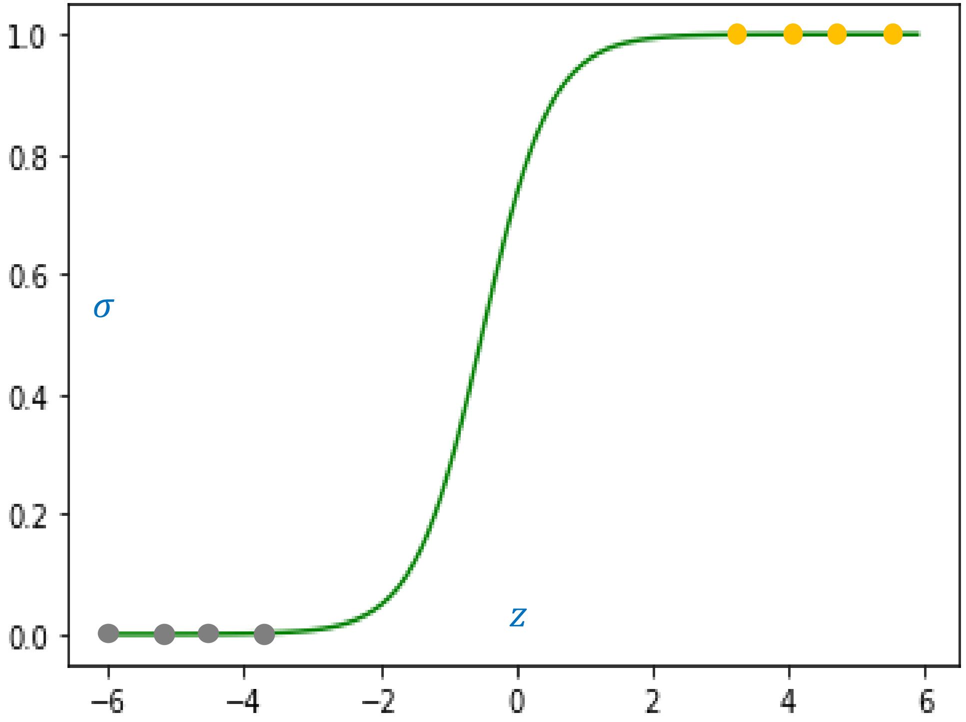
Outline

SECTION 1

Logistic Regression

SECTION 2

Softmax Regression



Logistic Regression

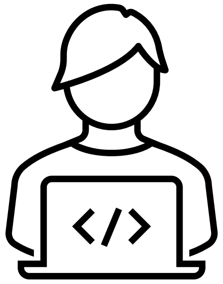
Classification Task

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	?

Prediction

Learning



Logistic Regression

Classification Task

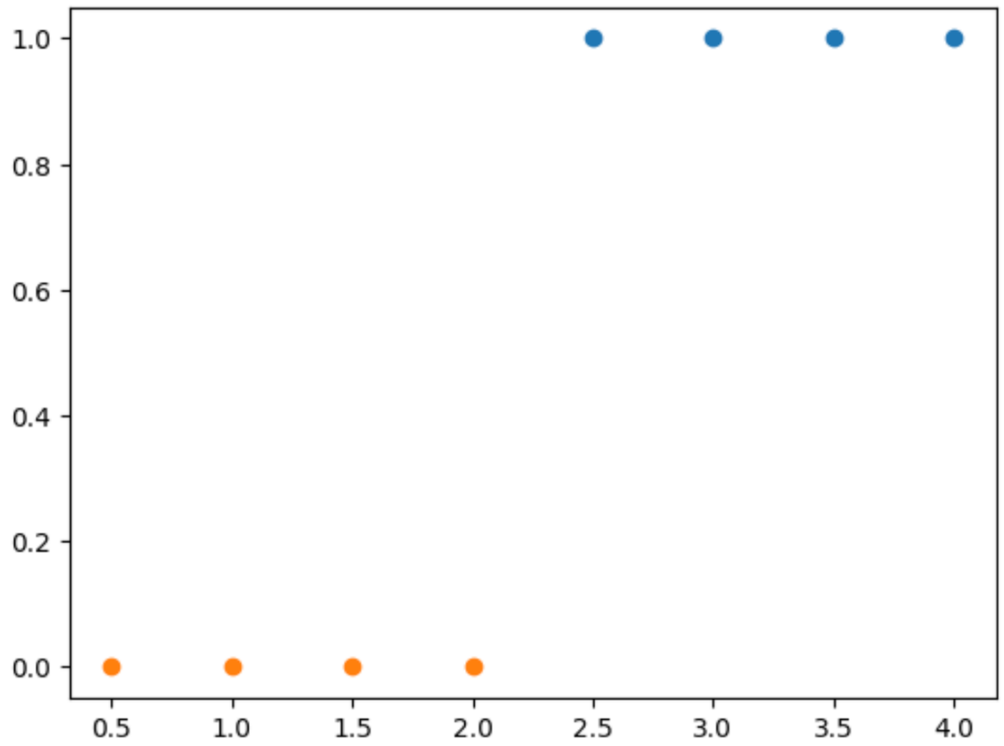
Hours	Pass
0.5	0
1.0	0
1.5	0
2.0	0
2.5	1
3.0	1
3.5	1
4.0	1

$y = f(x)$

Find a function to fit the data

Sigmoid function

Visualization



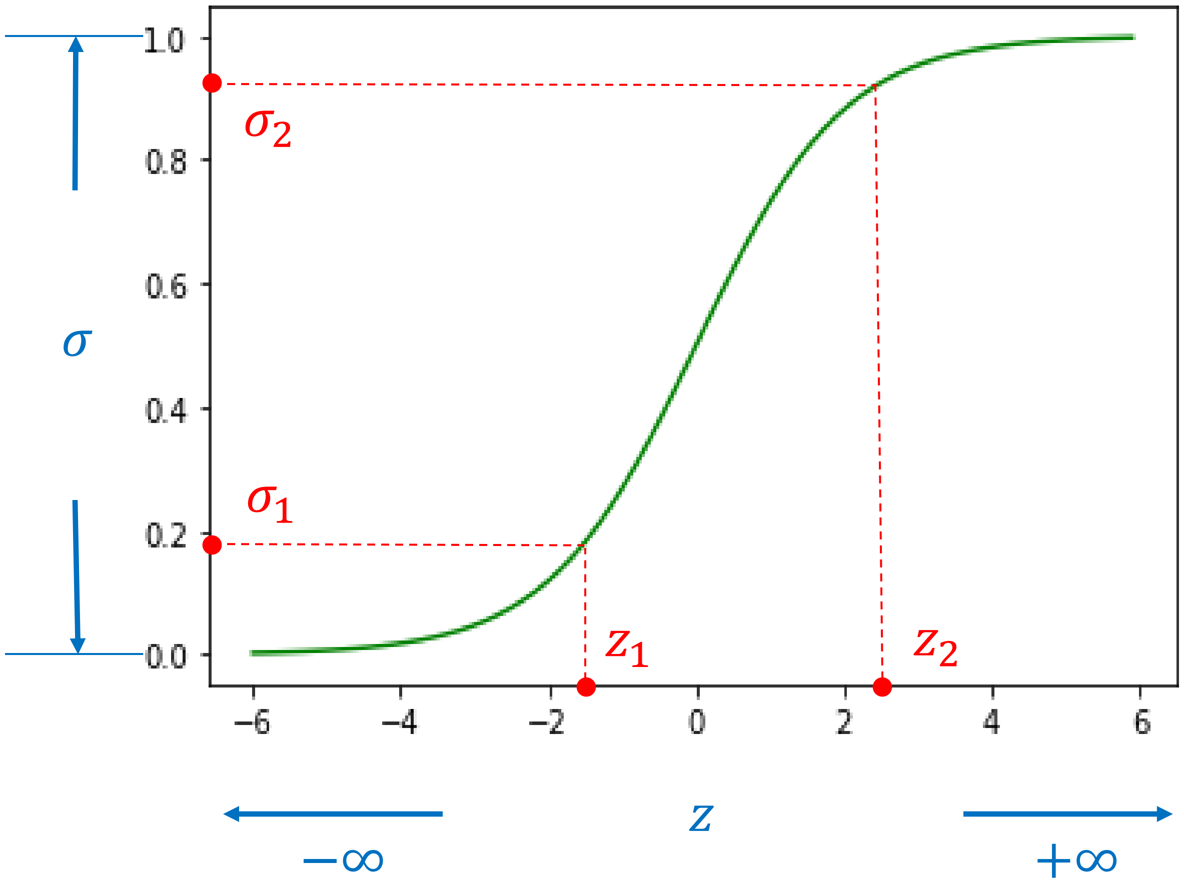
Logistic Regression

Sigmoid Function

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

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

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



Logistic Regression

Logistic Regression using Gradient Descent

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

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

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

5) Update parameters

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

η is learning rate

$$\boldsymbol{\theta} = \begin{bmatrix} \mathrm{b} \\ \mathrm{w} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$$

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

$$\boldsymbol{\eta} = 0.1$$

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

Logistic Regression

Logistic Regression using Gradient Descent

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

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

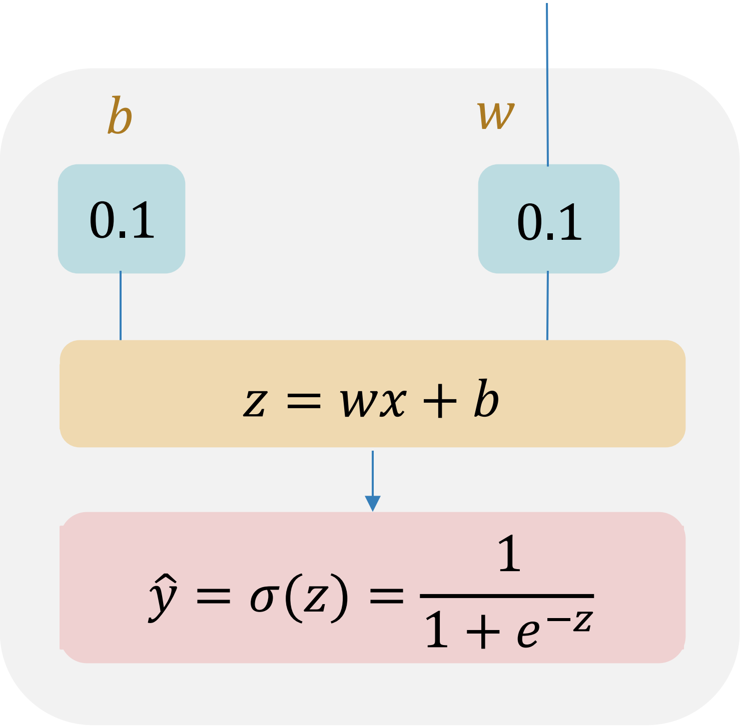
$$\nabla_{\theta}\mathrm{L} = \boldsymbol{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

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

η is learning rate

$\boldsymbol{\eta} = 0.1$ $\quad$ $\boldsymbol{\theta} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$ $\quad$ $\boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad$ $\boldsymbol{y} = [0]$



$z = 0.15$

$\hat{y} = 0.54$

Logistic Regression

Logistic Regression using Gradient Descent

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

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

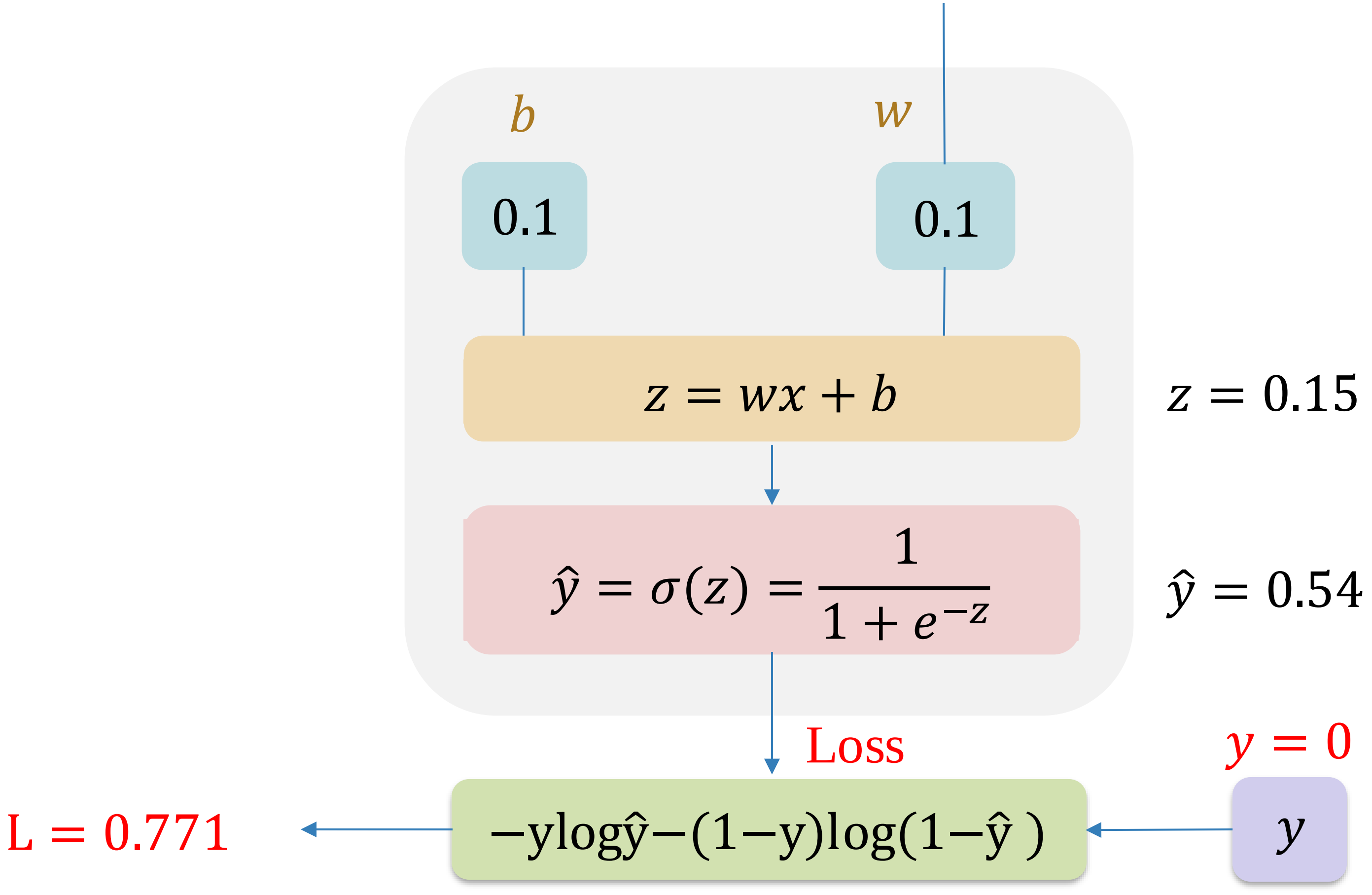
$$\nabla_{\theta}\mathrm{L} = \boldsymbol{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

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

η is learning rate

$\boldsymbol{\eta} = 0.1$ $\quad \boldsymbol{\theta} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$ $\quad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad \boldsymbol{y} = [0]$



Logistic Regression

Logistic Regression using Gradient Descent

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

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

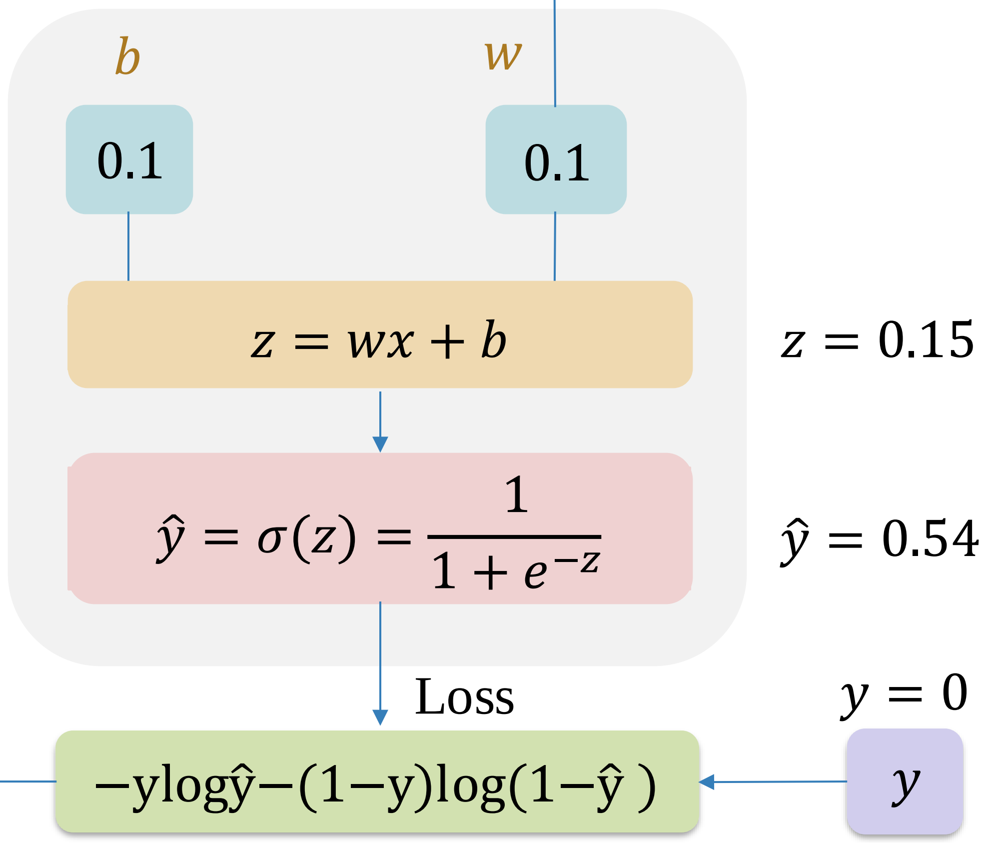
$$\nabla_\theta \mathrm{L} = \boldsymbol{x}(\hat{\mathrm{y}} - y)$$

5) Update parameters

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

η is learning rate

$\boldsymbol{\eta} = 0.1$ $\quad \boldsymbol{\theta} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$ $\quad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad \boldsymbol{y} = [0]$



$$\nabla_\theta L = \boldsymbol{x}(\hat{\mathrm{y}} - y) = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix} * 0.54 = \begin{bmatrix} 0.54 \\ 0.27 \end{bmatrix}$$

$\mathrm{L} = 0.771$

AI VIET NAM
@aivietnam.edu.vn

Logistic Regression

Logistic Regression using Gradient Descent

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

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

5) Update parameters

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

η is learning rate

$\boldsymbol{\eta} = 0.1$ $\quad \boldsymbol{\theta} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$ $\quad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad \boldsymbol{y} = [0]$

b: 0.1 $\quad w$: 0.1

$z = wx + b$ $\quad z = 0.15$

$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$ $\quad \hat{y} = 0.54$

Loss $\quad y = 0$

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

$\mathrm{L} = 0.771$

$$\nabla_{\theta} L = \boldsymbol{x}(\hat{\mathrm{y}} - y) = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix} * 0.54 = \begin{bmatrix} 0.54 \\ 0.27 \end{bmatrix}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta\nabla_{\theta}\mathrm{L} = \begin{bmatrix} 0.046 \\ 0.073 \end{bmatrix}$$

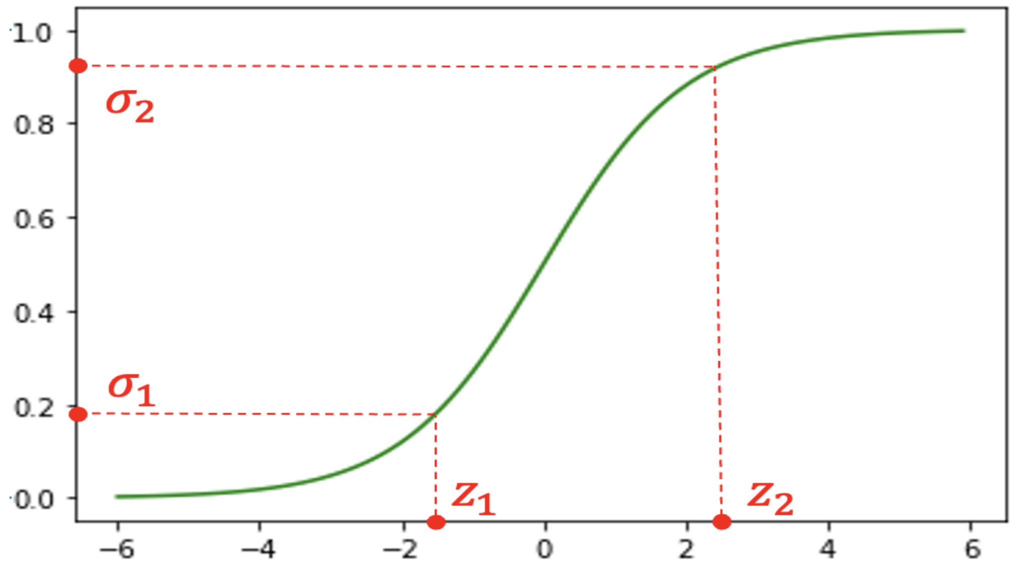
Outline

SECTION 1

Logistic Regression

SECTION 2

Softmax Regression



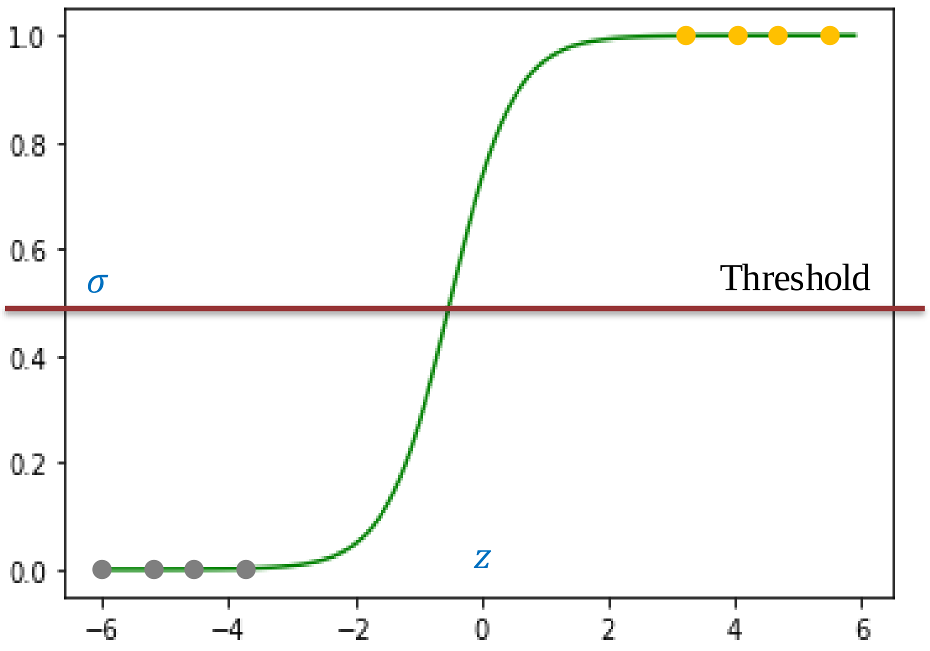
Softmax Regression

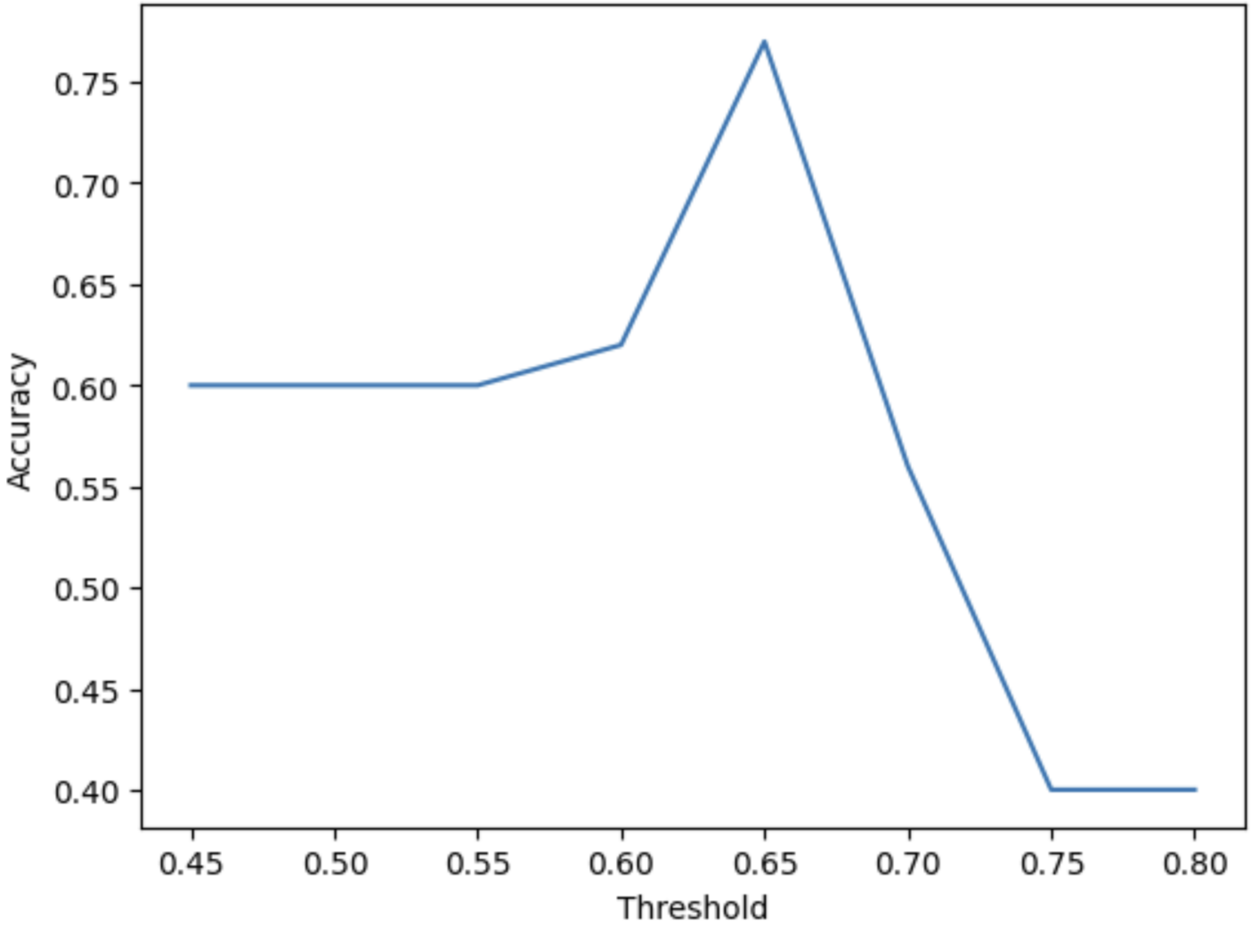
Problem

Sigmoid function

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

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





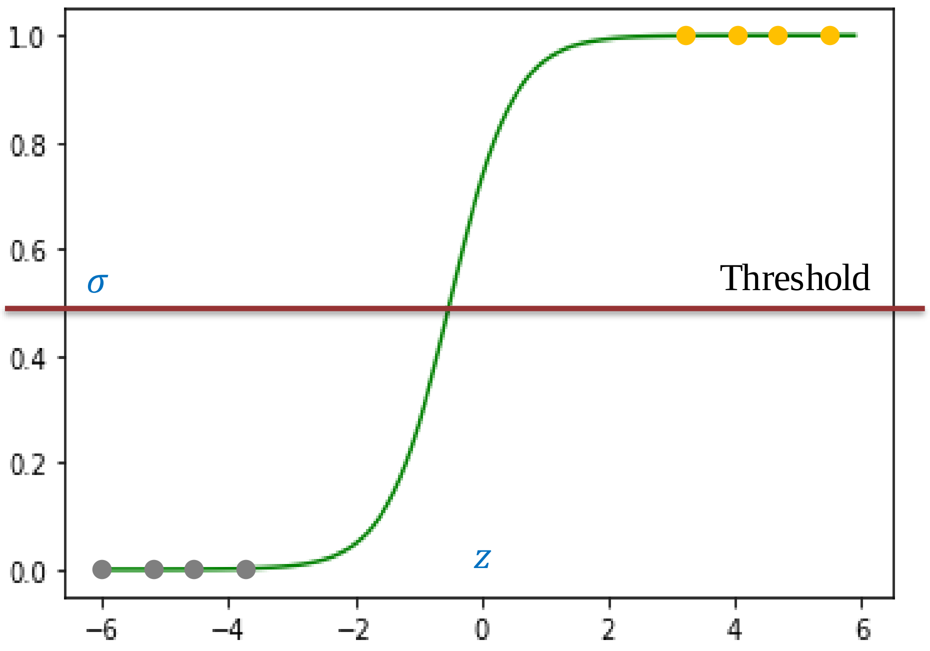
Softmax Regression

Problem

Sigmoid function

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

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



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

AI VIET NAM
@aivietnam.edu.vn

Softmax Regression

! Problem

Classes: {0, 1}
Binary Classification

Hours	Pass
0.5	0
2.0	1

Input $x = 0.5$ → Compute z → ~~Sigmoid function~~ Function $\hat{y} = f(z)$ → $P(0|X) = 0.3$; $P(1|X) = 0.7$ → 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; $P(3|X) = 0.1$

AI VIET NAM
@aivietnam.edu.vn

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

Softmax Regression

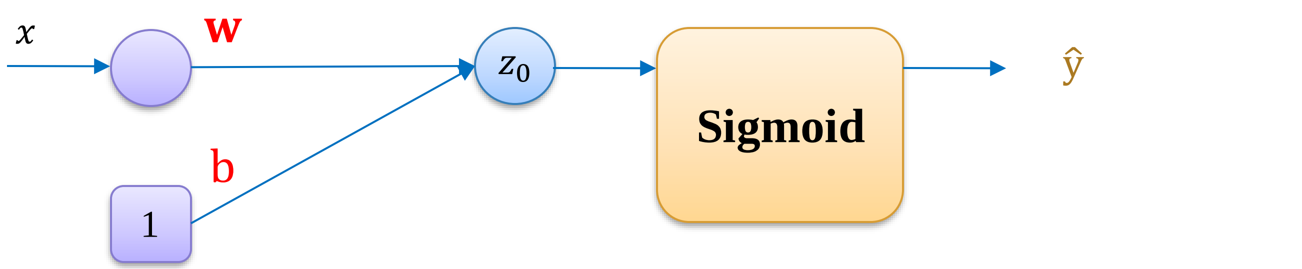
Parameters

Classes: {0, 1}
Binary Classification

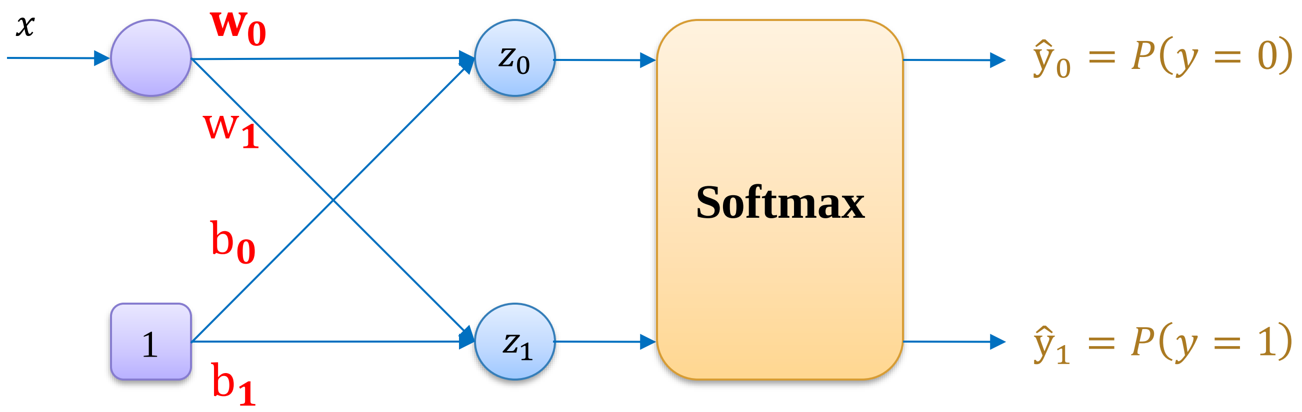
Hours	Pass
0.5	0
2.0	1

#feature: 1
#class: 2

Logistic Regression



Softmax Regression



Softmax Regression

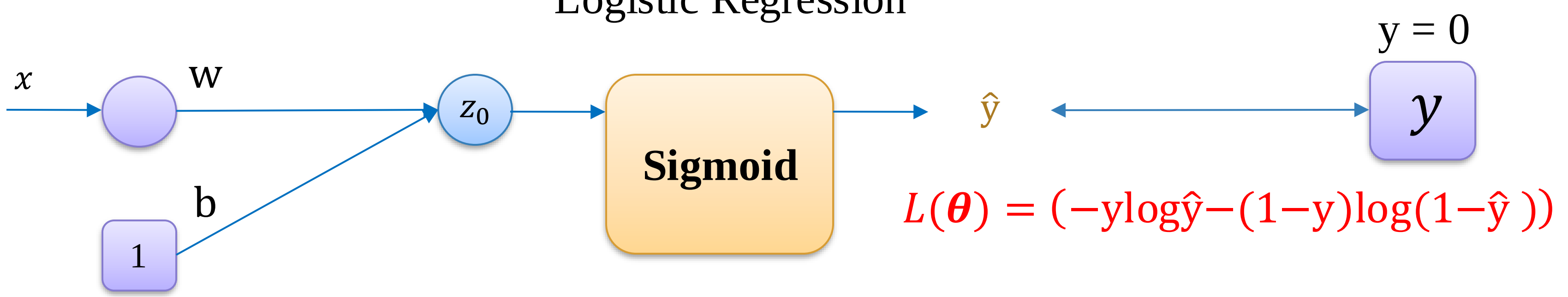
Loss Function

Classes: {0, 1}
Binary Classification

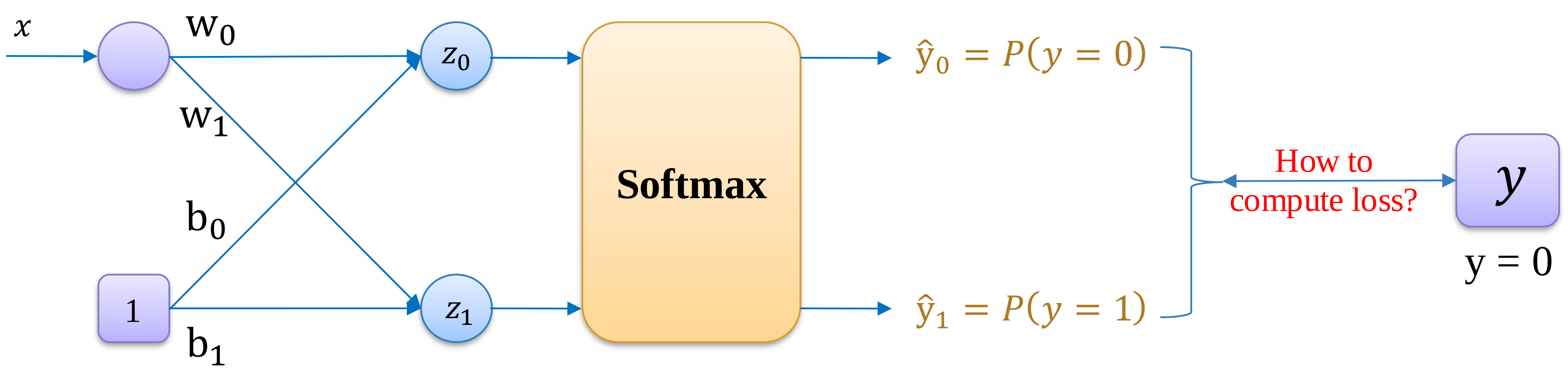
Hours	Pass
0.5	0
2.0	1

#feature: 1
#class: 2

Logistic Regression



Softmax Regression



AI VIET NAM
@aivietnam.edu.vn

Softmax Regression

One-hot Encoding

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

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

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

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

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

Softmax Regression

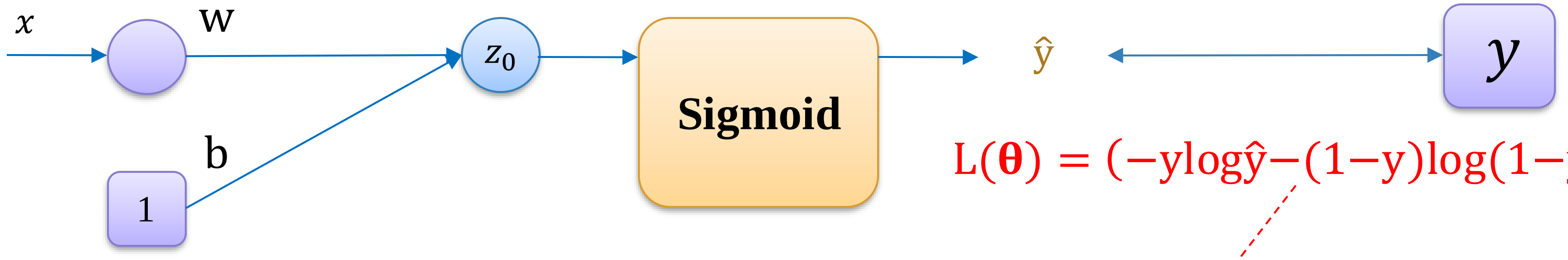
Loss Function

Classes: {0, 1}
Binary Classification

Hours	Pass
0.5	0
2.0	1

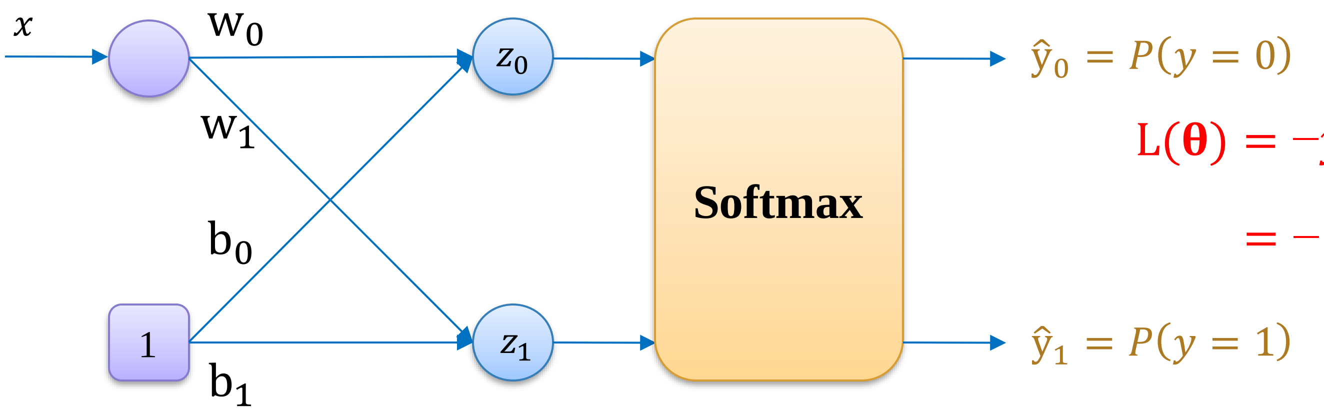
#feature: 1
#class: 2

Logistic Regression

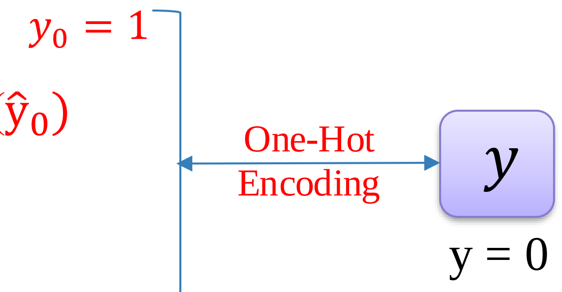


$$L(\boldsymbol{\theta}) = (-y\log\hat{y} - (1-y)\log(1-\hat{y}))$$

Softmax Regression



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



AI VIET NAM
@aivietnam.edu.vn

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

Hours	Pass
0.5	0
1.0	0
1.5	1
2.0	1

$$\boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$$

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

One-hot encoding for label

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

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

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

$$\boldsymbol{\eta} = 0.1$$

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$\boldsymbol{\eta} = 0.1$ $\quad$ $\boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad$ $\boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad$ $\boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$\boldsymbol{\eta} = 0.1$ $\quad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad \boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad \boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$

0.1 0.3 0.2 0.4

$z_0 = w_0 x_0 + b_0$ $\quad z_1 = w_1 x_1 + b_1$

$\boldsymbol{z}_0 = [0.25]$ $\quad \boldsymbol{z}_1 = [0.4]$

$$\hat{y}_0 = \frac{e^{z_0}}{\sum_{i=0}^{1} e^{z_i}} \qquad \hat{y}_1 = \frac{e^{z_1}}{\sum_{i=0}^{1} e^{z_i}}$$

$\hat{\boldsymbol{y}}_0 = [0.46]$ $\quad \hat{\boldsymbol{y}}_1 = [0.54]$

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$\boldsymbol{\eta} = 0.1$ $\quad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad \boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad \boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$

0.1 0.3 0.2 0.4

$z_0 = w_0 x_0 + b_0$ $\quad z_1 = w_1 x_1 + b_1$

$\boldsymbol{z}_0 = [0.25]$ $\quad \boldsymbol{z}_1 = [0.4]$

$$\hat{y}_0 = \frac{e^{z_0}}{\sum_{i=0}^{1} e^{z_i}} \qquad \hat{y}_1 = \frac{e^{z_1}}{\sum_{i=0}^{1} e^{z_i}}$$

$\hat{\boldsymbol{y}}_0 = [0.46]$ $\quad \hat{\boldsymbol{y}}_1 = [0.54]$

$$\mathrm{L} = -y_0 \log \hat{y}_0 - y_1 \log \hat{y}_1$$

$\mathrm{L} = [0.77]$

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$\boldsymbol{\eta} = 0.1$ $\quad$ $\boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad$ $\boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad$ $\boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$

0.1   0.3   0.2   0.4

$$z_0 = w_0 x_0 + b_0 \qquad z_1 = w_1 x_1 + b_1$$

$\boldsymbol{z}_0 = [0.25]$ $\quad$ $\boldsymbol{z}_1 = [0.4]$

$$\hat{y}_0 = \frac{e^{z_0}}{\sum_{i=0}^{1} e^{z_i}} \qquad \hat{y}_1 = \frac{e^{z_1}}{\sum_{i=0}^{1} e^{z_i}}$$

$\hat{\boldsymbol{y}}_0 = [0.46]$ $\quad$ $\hat{\boldsymbol{y}}_1 = [0.54]$

$$\mathrm{L} = -y_0 \log \hat{y}_0 - y_1 \log \hat{y}_1$$

$\mathrm{L} = [0.77]$

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

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

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$\boldsymbol{\eta} = 0.1$ $\quad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad \boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad \boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$

0.1 0.3 0.2 0.4

$$z_0 = w_0 x_0 + b_0 \qquad z_1 = w_1 x_1 + b_1$$

$\boldsymbol{z}_0 = [0.25]$ $\quad \boldsymbol{z}_1 = [0.4]$

$$\hat{y}_0 = \frac{e^{z_0}}{\sum_{i=0}^{1} e^{z_i}} \qquad \hat{y}_1 = \frac{e^{z_1}}{\sum_{i=0}^{1} e^{z_i}}$$

$\hat{\boldsymbol{y}}_0 = [0.46]$ $\quad \hat{\boldsymbol{y}}_1 = [0.54]$

$$\mathrm{L} = -y_0 \log\hat{y}_0 - y_1 \log\hat{y}_1$$

$\mathrm{L} = [0.77]$

$$\nabla_\theta \mathrm{L} = \mathbf{x}(\hat{y} - y)^T = \begin{bmatrix} -0.54 & 0.54 \\ -0.27 & 0.27 \end{bmatrix}$$

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \nabla_\theta \mathrm{L} = \begin{bmatrix} 0.105 & 0.195 \\ 0.303 & 0.397 \end{bmatrix}$$

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data
2) Compute output $\hat{y}$

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

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

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

$\emptyset$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

Hours	Pass
0.5	0
1.0	0
1.5	1
2.0	1

$$\boldsymbol{x} = \begin{bmatrix} 1.0 \\ 1.5 \end{bmatrix}$$

$$\boldsymbol{y} = [1]$$

One-hot encoding for label

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

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

$$\boldsymbol{\theta} = \begin{bmatrix} b_0 & b_1 \\ w_0 & w_1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

$$\boldsymbol{\eta} = 0.1$$

Softmax Regression

Softmax Regression using Gradient Descent

Hours	Pass
0.5	?

0.105   0.303   0.195   0.397

$z_0 = w_0 x_0 + b_0$

$z_1 = w_1 x_1 + b_1$

$\boldsymbol{z}_0 = [0.26]$

$\boldsymbol{z}_1 = [0.39]$

$$\hat{y}_0 = \frac{e^{z_0}}{\sum_{i=0}^{1} e^{z_i}}$$

$$\hat{y}_1 = \frac{e^{z_1}}{\sum_{i=0}^{1} e^{z_i}}$$

$\hat{\boldsymbol{y}}_0 = [0.47]$

$\hat{\boldsymbol{y}}_1 = [0.53]$

$y_{pred} = 1$


QUIZ TIME

AI VIET NAM
@aivietnam.edu.vn

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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
X = np.array([
     [0.5],
     [1.0],
     [1.5],
     [2.0],
     [2.5],
     [3.0],
     [3.5],
     [4.0],
])
Y = np.array([0, 0, 1, 0, 0, 1, 1, 1])
```

```python
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_onehot = convert_one_hot(Y, n_classes)
```

```python
X = np.hstack([np.ones((X.shape[0], 1)), X])
```

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$$\boldsymbol{\eta} = 0.1 \qquad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix} \qquad \boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

```
x = X[0].reshape((2,1))
y = Y_onehot[0].reshape((2,1))
x, y
```

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

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

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

AI VIET NAM
@aivietnam.edu.vn

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\oslash$ is Hadamard division

3) Compute loss (cross-entropy)

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

4) Compute derivative

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

5) Update parameters

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

η is learning rate

$$\boldsymbol{\eta} = 0.1 \qquad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix} \qquad \boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

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

```python
# compute y_hat
def predict(x, theta):
    z = theta.T.dot(x)
    y_hat = softmax_function(z)
    return z, y_hat

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

```
(array([[0.25],
        [0.4 ]]),
 array([[0.46257015],
        [0.53742985]]))
```

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$\boldsymbol{\eta} = 0.1$ $\quad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad \boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad \boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$

```python
# compute loss
def compute_loss(y_hat, y):
    loss = -np.log(y.T.dot(y_hat))
    return loss

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

```
array([[0.77095705]])
```

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$\boldsymbol{\eta} = 0.1$ $\quad \boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad \boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad \boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$

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

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

```
array([[-0.53742985,  0.53742985],
       [-0.26871492,  0.26871492]])
```

Softmax Regression

Softmax Regression using Gradient Descent

1) Pick a sample from training data

2) Compute output $\hat{y}$

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

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

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

$\emptyset$ 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

$\boldsymbol{\eta} = 0.1$ $\quad$ $\boldsymbol{x} = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$ $\quad$ $\boldsymbol{y} = 0 \rightarrow \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\quad$ $\boldsymbol{\theta} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$

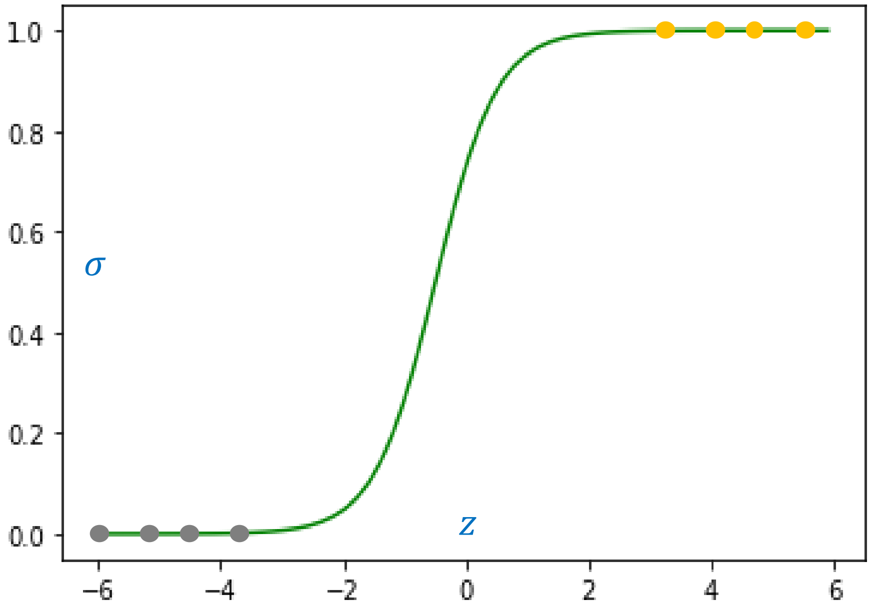
```python
# update weights
learning_rate = 0.01

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

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

```
array([[0.1053743 , 0.1946257 ],
       [0.30268715, 0.39731285]])
```

Objectives

Logistic Regression (Review)

- ❖ Logistic Regression
- ❖ Sigmoid Function
- ❖ Gradient Descent



Softmax Regression

- ❖ Softmax Regression
- ❖ Softmax Function
- ❖ One Sample
- ❖ N Sample

Thanks!

Any questions?