Artificial Intelligence

How to binary classification using linear regression?

Review: Linear Regression

$$\begin{bmatrix} \ \end{bmatrix}\begin{bmatrix} \ \end{bmatrix} = H(x)$$
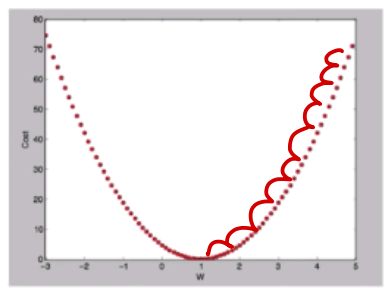
$$w_1 x_1 + w_2 x_2 = y$$

x_1 (hours)	x_2 (attendance)	y (score)
w_1	w_2	
10	5	90
9	5	80
3	2	50
2	4	60
11	1	40
10	3	?

S_1
S_2
⋮
S_5

가설 설정
- **Hypothesis:** $H(X) = WX$

예측
- **Cost:** $cost(W) = \frac{1}{m}\sum (WX - y)^2$
비용   실제값

- **Gradient decent:** $W := W - \alpha\frac{\partial}{\partial W}cost(W)$

α = learning rate

Today's Goal: Binary Classification

• Spam Detection: Spam or Ham

• Facebook feed: show or hide

• Credit Card Fraudulent Transaction detection: legitimate/fraud

: 사기거래   합법   불법

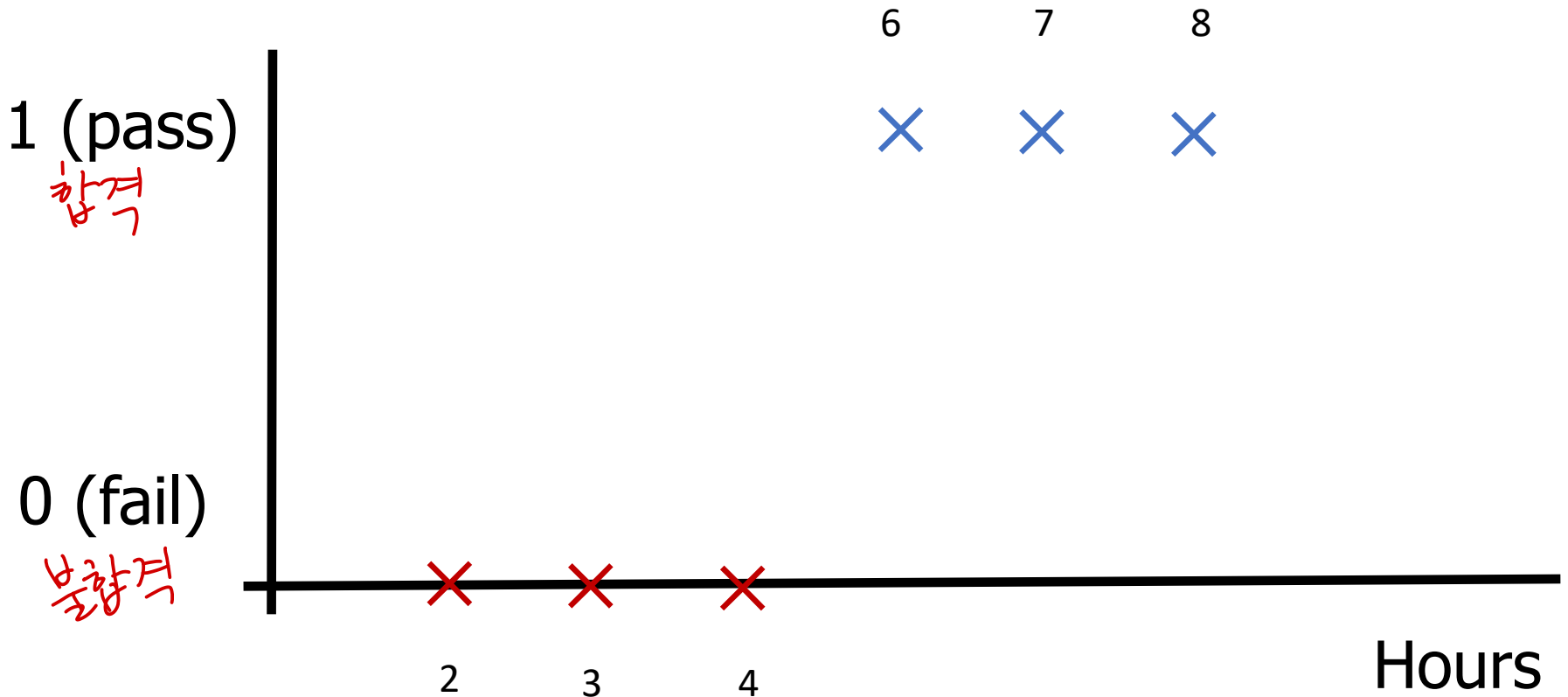# Binary Label Encoding➔ "0" or "1"   "라벨링"

- Spam Detection: Spam(1) or Ham(0)

- Facebook feed: show(1) or hide(0)

- Credit Card Fraudulent Transaction detection: legitimate(1)/fraud(0)
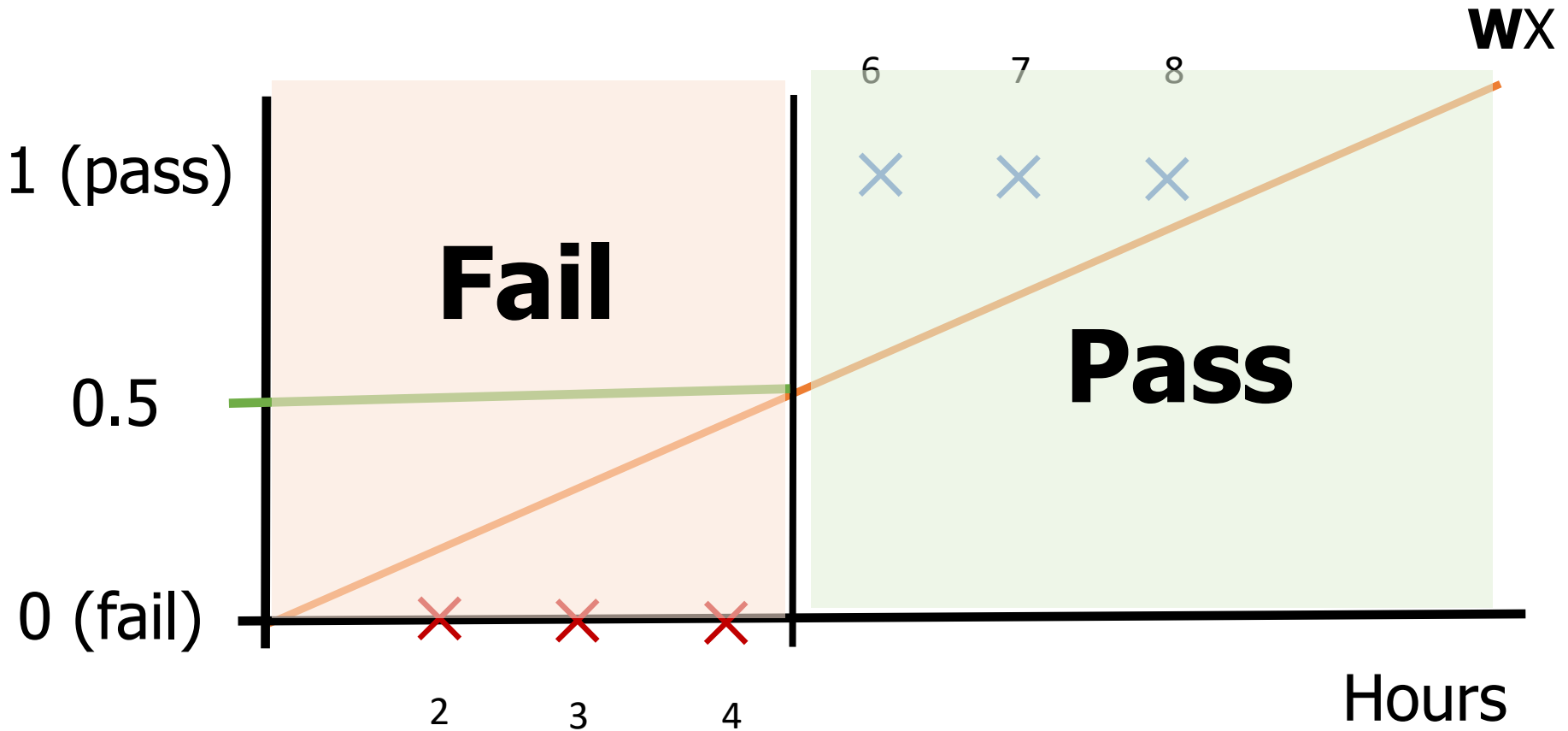
범주형 라벨 →인코딩

Y / N   →   1/0

Pass or Fail based on study hours

- Pass(1) / Fail(0)

1 (pass)
합격

0 (fail)
불합격

6   7   8

2   3   4

Hours

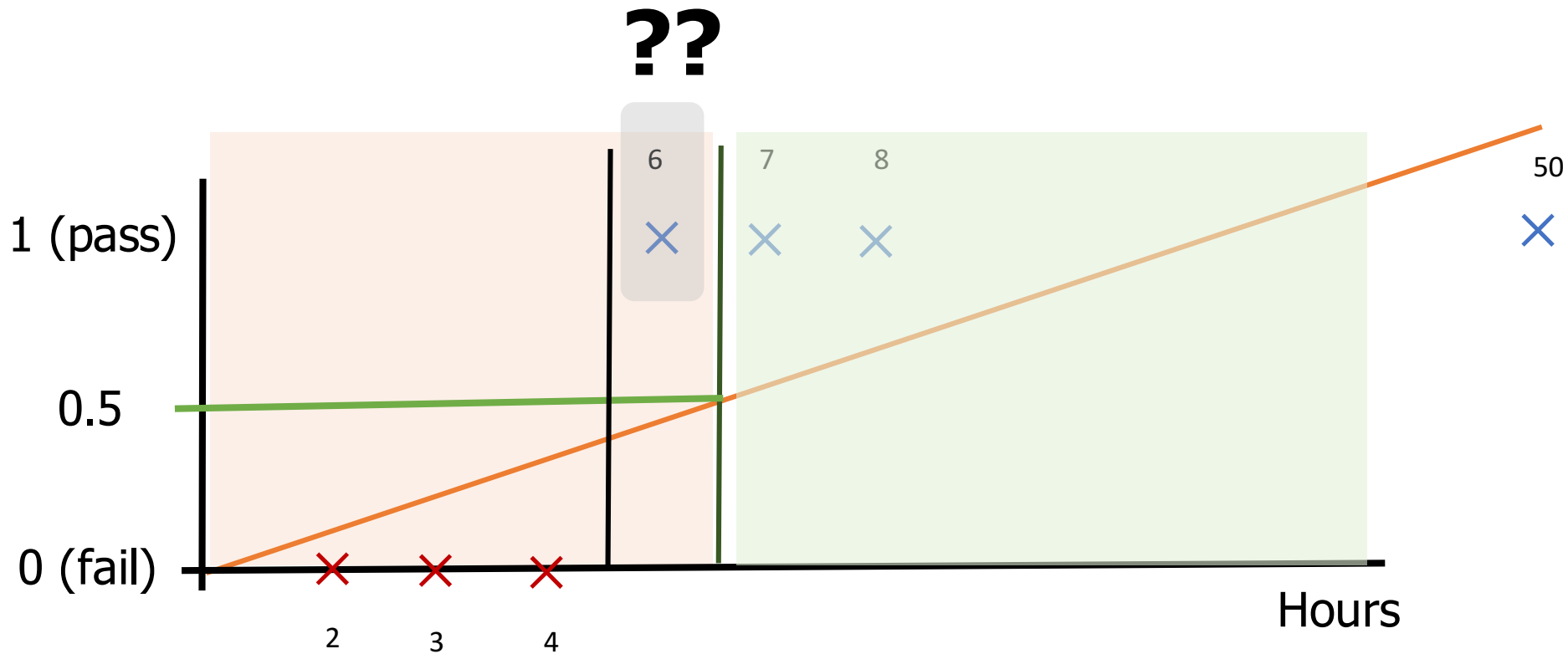# Binary Classification suing Linear Regression?

- Pass(1) / Fail(0)

Binary Classification ~~suing~~ Linear Regression?

using

- **Disadvantage #1**

Binary Classification suing Linear Regression?

- We know Y is 0 or 1

- **H**(x) = **W**x+**b**   예측
- (+) This hypothesis is simple and easy to use
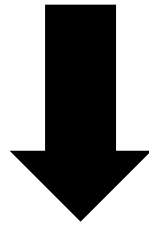- (-) This hypothesis can give values large than 1 or less than 0

- **Example)**
  x = [1;2;5;10], W=0.5, b = 0 ➔ 0 <= y <= 1
  But!!
  if x = 100, y = 50 >> 1

Logistic Hypothesis

$$H(x) = \mathbf{W}x + \mathbf{b}$$

How???

$$0 <= H(x) <= 1$$

Logistic Hypothesis

- Sigmoid: Curved in two directions, like the letter "S"

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

$$Z = WX$$

$$H(x) = \frac{1}{1+e^{-wx}}$$

Graph axes: $\sigma(z)$ vs $z = \sum w_i x_i + bias$
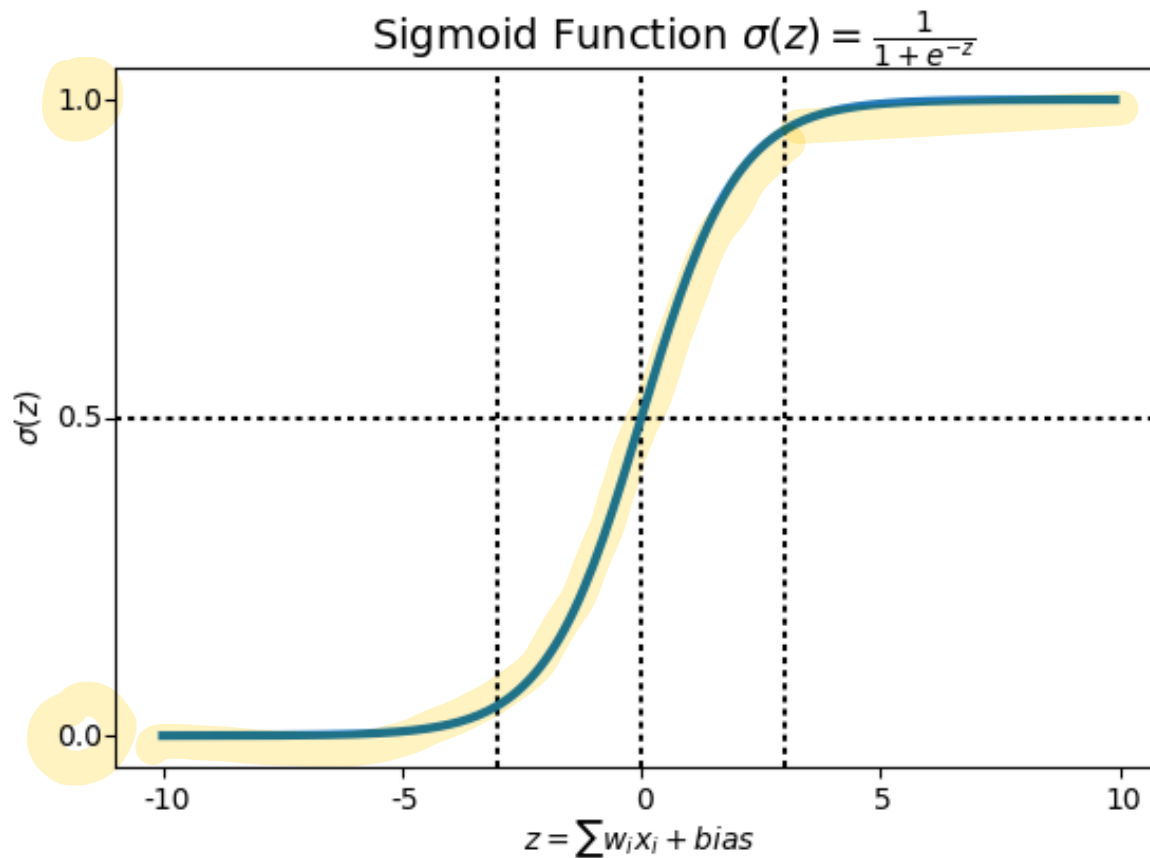
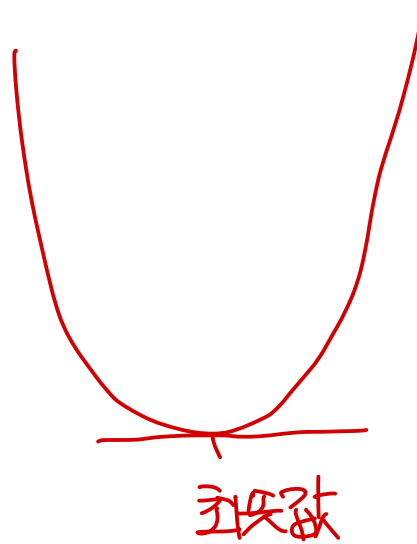**Logistic function := sigmoid function**

Logistic Hypothesis

- Sigmoid 함수 덕분에 H(x)가 Bound 되었다.

$$H(X) = \frac{1}{1 + e^{-(W^T X)}}$$

0<= H(x) <= 1
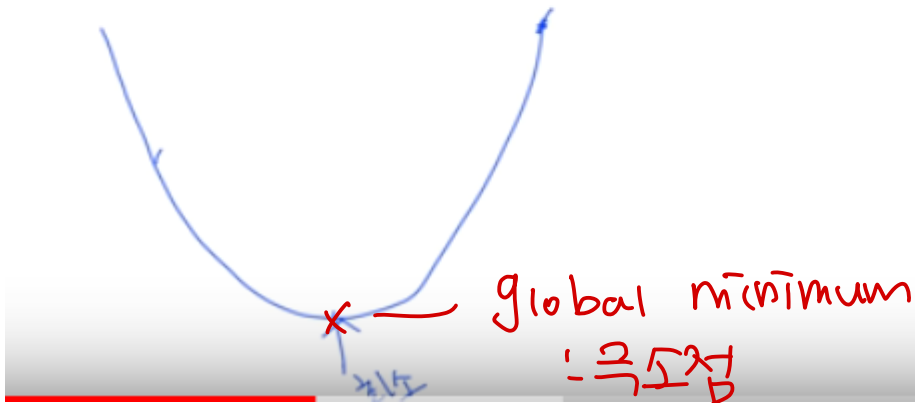
Cost function

- 이제 Cost function 에 적용해보자.

최댓값

Cost function

- 기존 linear regression cost function에 적용하니, local minima에 빠진다.

$$cost(W, b) = \frac{1}{m} \sum_{i=1}^{m} (H(x^{(i)}) - y^{(i)})^2$$

$0 < \sim < 1$

$$H(x) = Wx + b$$

$$H(X) = \frac{1}{1 + e^{-W^T X}}$$

global minimum
: 극소점

Local minimum
: 지역 극소점

Cost function

즉, 가설 설정이 변경되었으니 Cost 재설계 필요

- What is a cost function?
  - **우리의 "예측 값"(가설 값)이 얼마나 "정답"에 가까운가를 측정하는 척도!**

설계 팁:

정답에 가까워 질수록 Cost function 값 은 작고

정답에서 멀어질 수록 Cost function 값은 크게!

설계하면 된다.

Cost function for logistic regression

- Logistic regression
  - **H(x) ➜ [0, 1]**
  - if H(x) < 0.5, y = 0 else y = 1

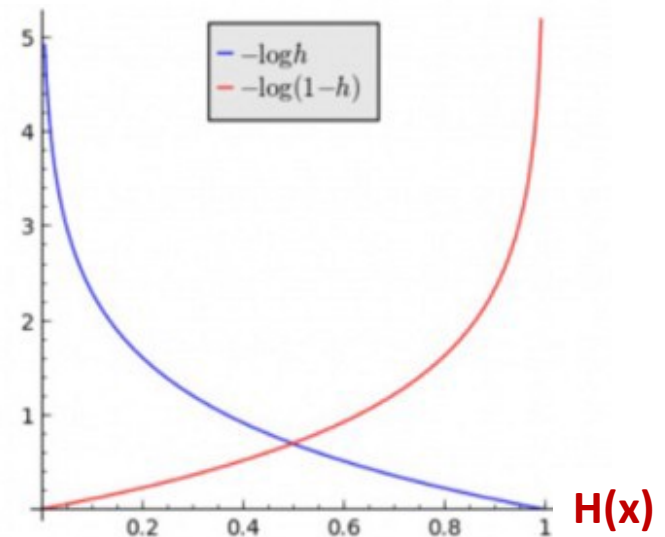 - 정답에 가까워 질수록 Cost function 값 은 작고, 정답에서 멀어질 수록 Cost function 값은 크게! 설계하면 된다.

Sigmoid

$$H(X) = \frac{1}{1 + e^{-(W^T X)}}$$

$$cost(W) = \frac{1}{m} \sum c(H(x), y)$$

$$c(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \quad \text{파} \\ -log(1 - H(x)) & : y = 0 \quad \text{반} \end{cases}$$

$y = 1 \simeq H(x) = \; ; \; cost \; \text{작} \; -log(H(x)) = 0$

C(H(x),y)

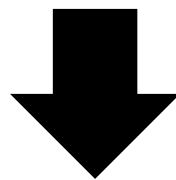**H(x)**

Cost function for logistic regression

$$cost(W) = \frac{1}{m} \sum c(H(x), y)$$

$$c(H(x), y) = \begin{cases} -log(H(x)) & : y = 1 \\ -log(1 - H(x)) & : y = 0 \end{cases}$$

하나의 equation으로 잘 표현하면

$$C(H(x), y) = -ylog\big(H(x)\big) - (1 - y)log(1 - H(x))$$

Gradient decent algorithm

$$H(X) = \frac{1}{1 + e^{-W^T X}}$$

$$cost(W) = -\frac{1}{m} \sum ylog\big(H(x)\big) + (1-y)\log(1 - H(x))$$

$$W := W - \alpha \frac{\partial}{\partial W} \text{cost}(W)$$

각자 해보자!!

(Summary) Logistic regression for binary classification
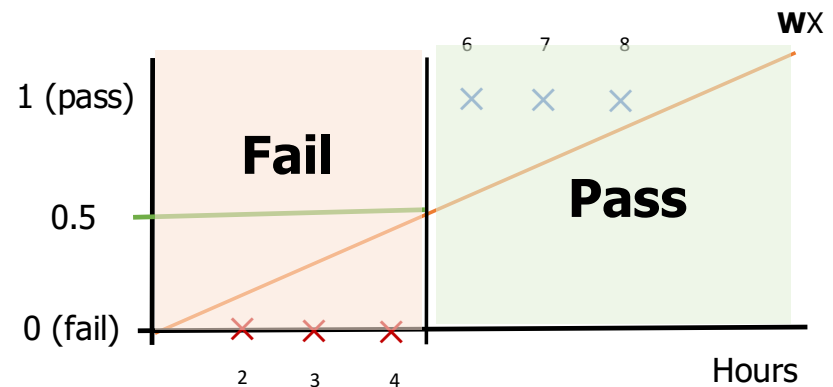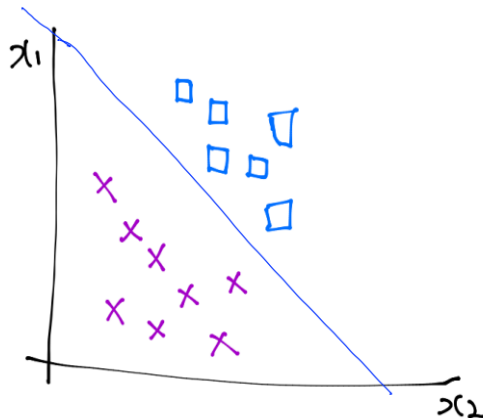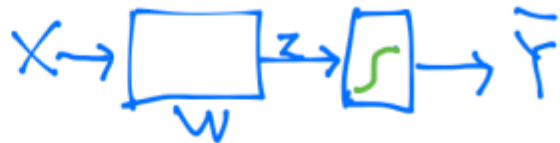
$$H_L(X) = WX$$

$$Z = H_L(X), \quad g(z)$$

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

$$H_R(x) = g(H_L(x))$$

(Summary) Logistic regression for binary classification

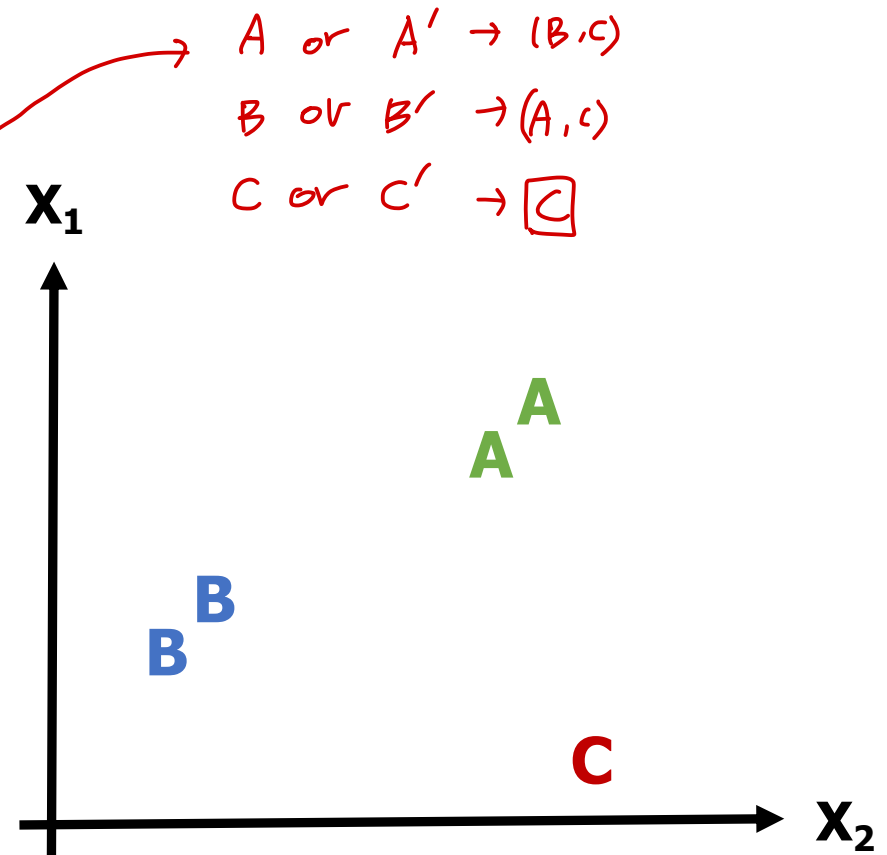$$g(z) = \frac{1}{1 + e^{-z}} \qquad H_R(x) = g(H_L(x))$$

How to multi-class classification using binary classification?

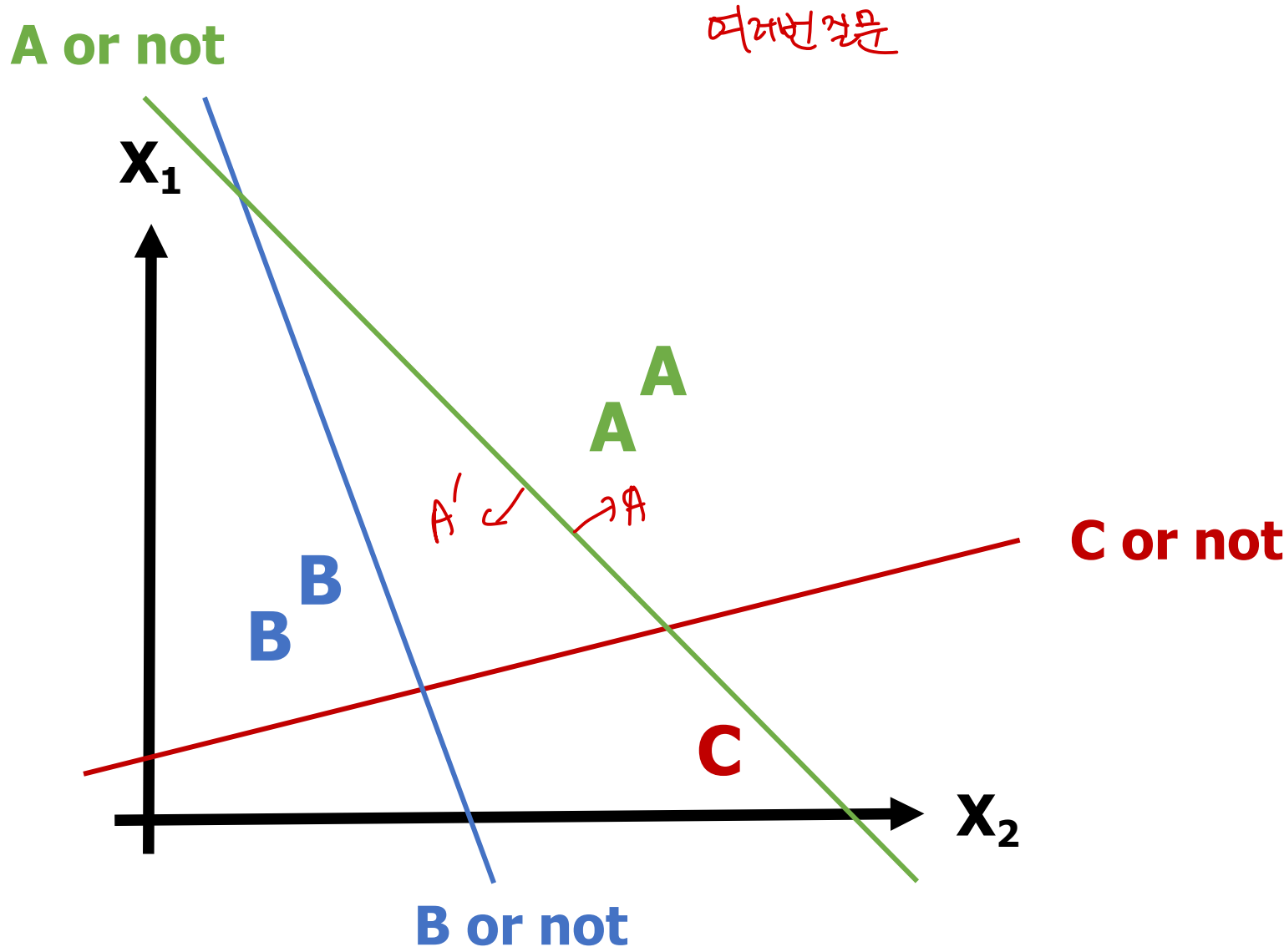*Multinomial classification := Softmax classification*

Multinomial classification (여러개의 클래스!)

A or A' → (B, C)
B or B' → (A, C)
C or C' → [C]

x1 (hours)	x2 (attendance)	y (grade)
10	5	A
9	5	A
3	2	B
2	4	B
11	1	C

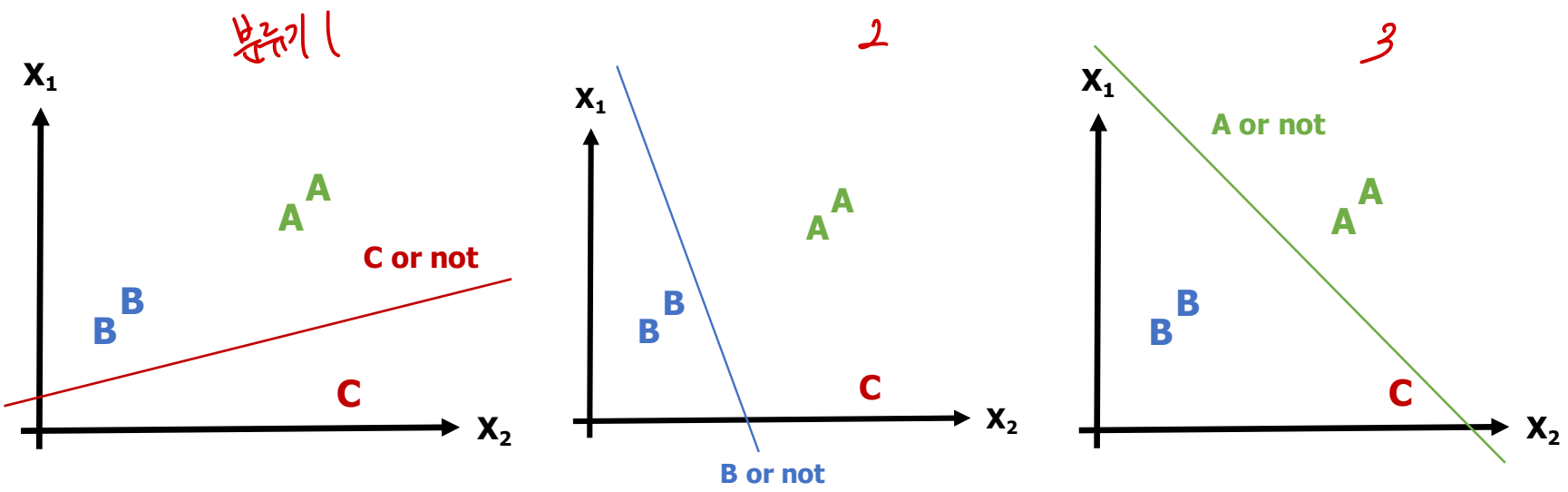10   3   ?
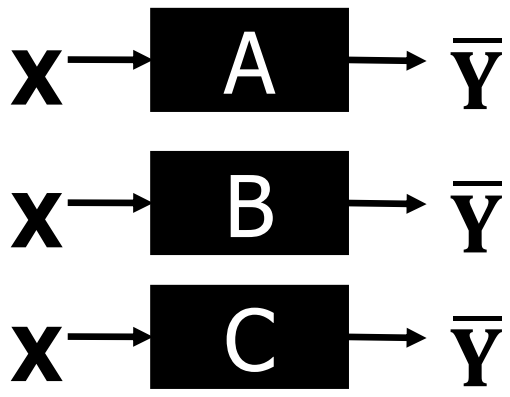
X_1

X_2

A A
B B
C

클래스 : 3개

Multinomial classification using binary classification
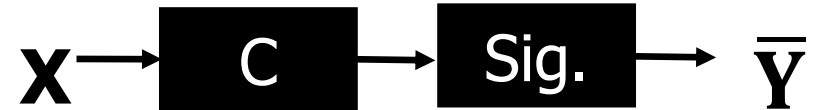
Multinomial classification using binary classification

X → [A] → $\overline{Y}$

X → [B] → $\overline{Y}$

X → [C] → $\overline{Y}$

분류기 1

X_1

A
A A

C or not

B
B B

C

X_2

2

X_1

A
A A

B
B B

C

X_2
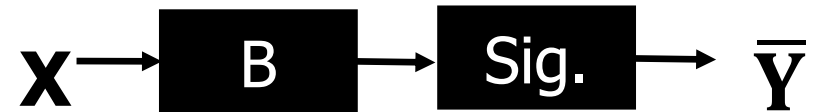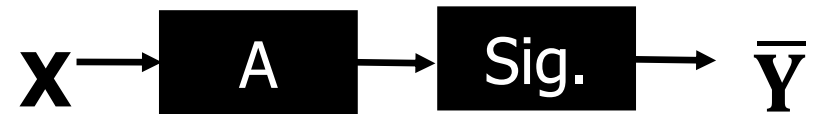
B or not

3

X_1

A or not

A
A A

B
B B

C

X_2

Multinomial classification

Using Matrix Operation

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1 x_1 + w_2 x_2 + w_3 x_3 \end{bmatrix}$$

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1 x_1 + w_2 x_2 + w_3 x_3 \end{bmatrix}$$

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1 x_1 + w_2 x_2 + w_3 x_3 \end{bmatrix}$$

X → | A | → | Sig. | → $\overline{Y}$

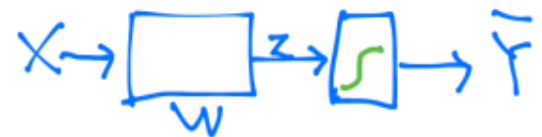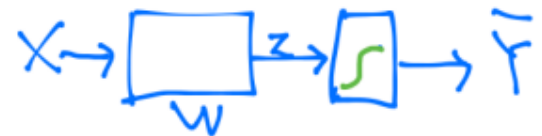**X** → | B | → | Sig. | → $\overline{Y}$

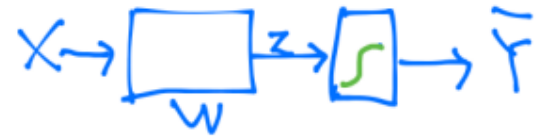**X** → | C | → | Sig. | → $\overline{Y}$

Multinomial classification
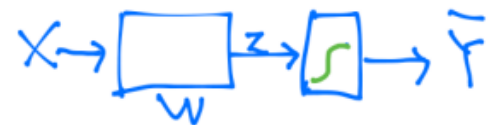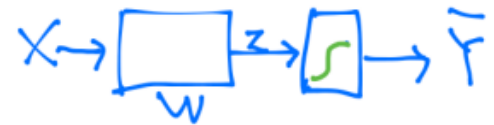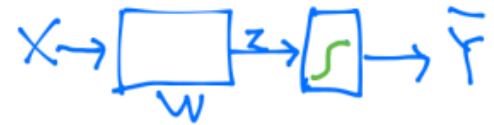
하나로 표현해서 복잡도를 내려보자. 코딩할 때도 편하게!

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1 x_1 + w_2 x_2 + w_3 x_3 \end{bmatrix}$$

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} =$$

Multinomial classification

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{A1}x_1 + w_{A2}x_2 + w_{A3}x_3 \\ w_{B1}x_1 + w_{B2}x_2 + w_{B3}x_3 \\ w_{C1}x_1 + w_{C2}x_2 + w_{C3}x_3 \end{bmatrix}$$

Multinomial classification

$$
\begin{bmatrix} W_{A1} & W_{A2} & W_{A3} \\ W_{B1} & W_{B2} & W_{B3} \\ W_{C1} & W_{C2} & W_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} W_{A1}x_1 + W_{A2}x_2 + W_{A3}x_3 \\ W_{B1}x_1 + W_{B2}x_2 + W_{B3}x_3 \\ W_{C1}x_1 + W_{C2}x_2 + W_{C3}x_3 \end{bmatrix} = \begin{bmatrix} \overline{y}_A \\ \overline{y}_B \\ \overline{y}_C \end{bmatrix}
$$

$H_A(x)$

$H_B(x)$

$H_C(x)$

$X \rightarrow \boxed{}_W \xrightarrow{Z} \boxed{\int} \rightarrow \widetilde{Y}$

$X \rightarrow \boxed{}_W \xrightarrow{Z} \boxed{\int} \rightarrow \widetilde{Y}$

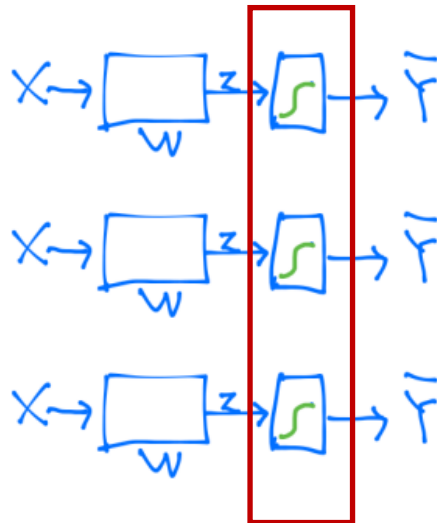$X \rightarrow \boxed{}_W \xrightarrow{Z} \boxed{\int} \rightarrow \widetilde{Y}$

Where is sigmoid?

$$\begin{bmatrix} W_{A1} & W_{A2} & W_{A3} \\ W_{B1} & W_{B2} & W_{B3} \\ W_{C1} & W_{C2} & W_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} W_{A1}x_1 + W_{A2}x_2 + W_{A3}x_3 \\ W_{B1}x_1 + W_{B2}x_2 + W_{B3}x_3 \\ W_{C1}x_1 + W_{C2}x_2 + W_{C3}x_3 \end{bmatrix} = \begin{bmatrix} \overline{y}_A \\ \overline{y}_B \\ \overline{y}_C \end{bmatrix}$$

$0 \sim 1$

$$\begin{bmatrix} 2.0 \\ 1.0 \\ 0.1 \end{bmatrix}$$

sigmoid $\rightarrow$
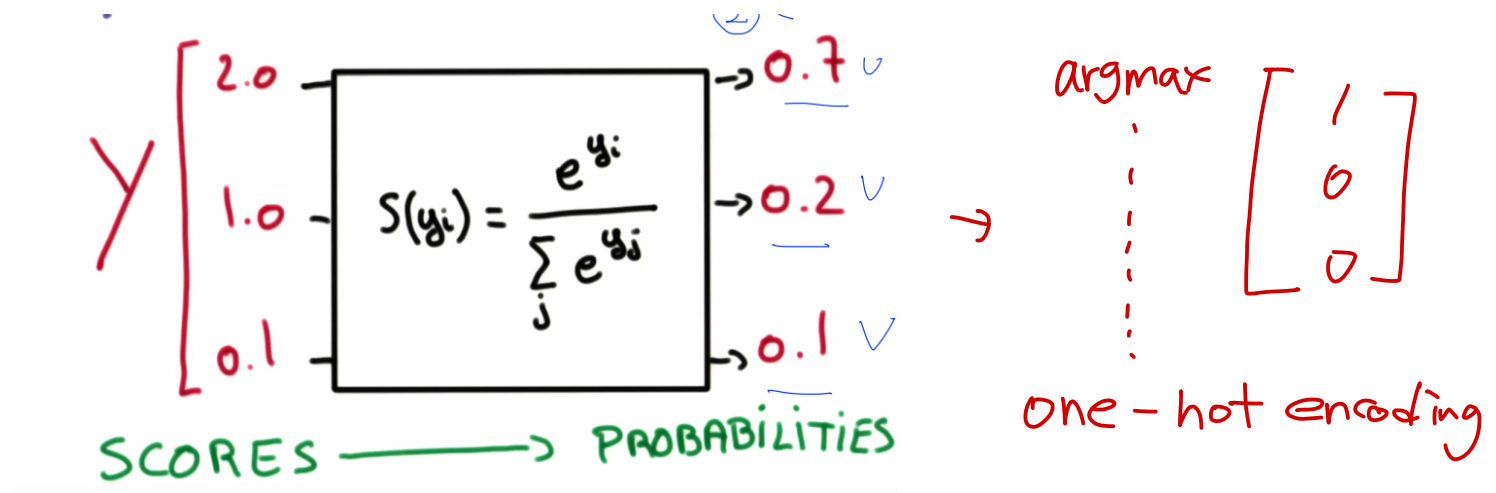
1 두개
부적합

1

0

0

1

0

0

Sigmoid?

Multiclass classification의 Hypothesis를 **효율적으로 [0,1]** 로 제한하는 방법을 알아보자.
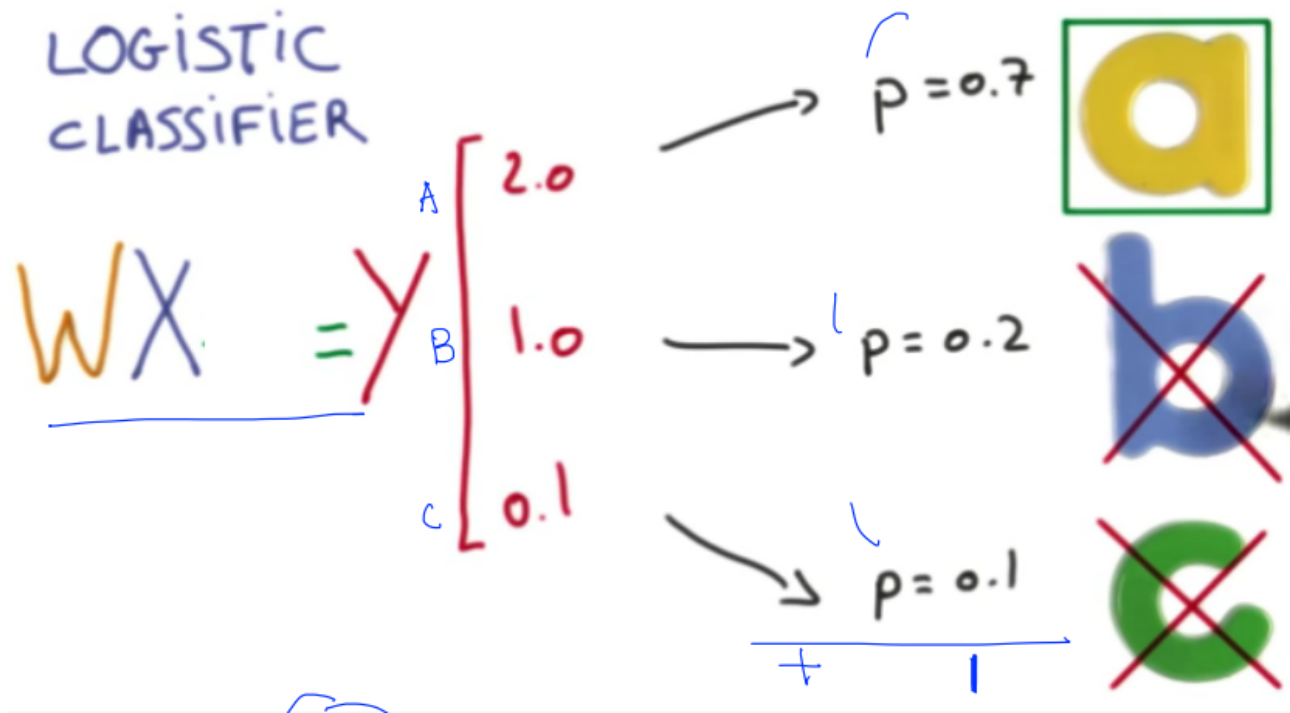
Softmax function instead of sigmoid function

Softmax function

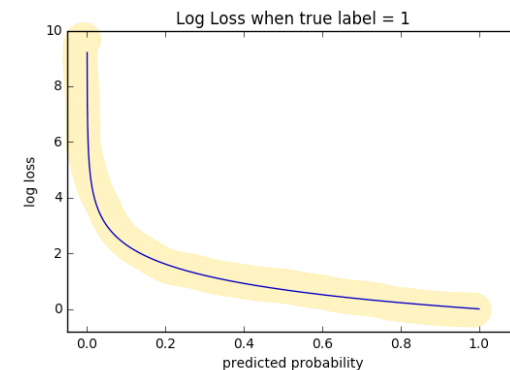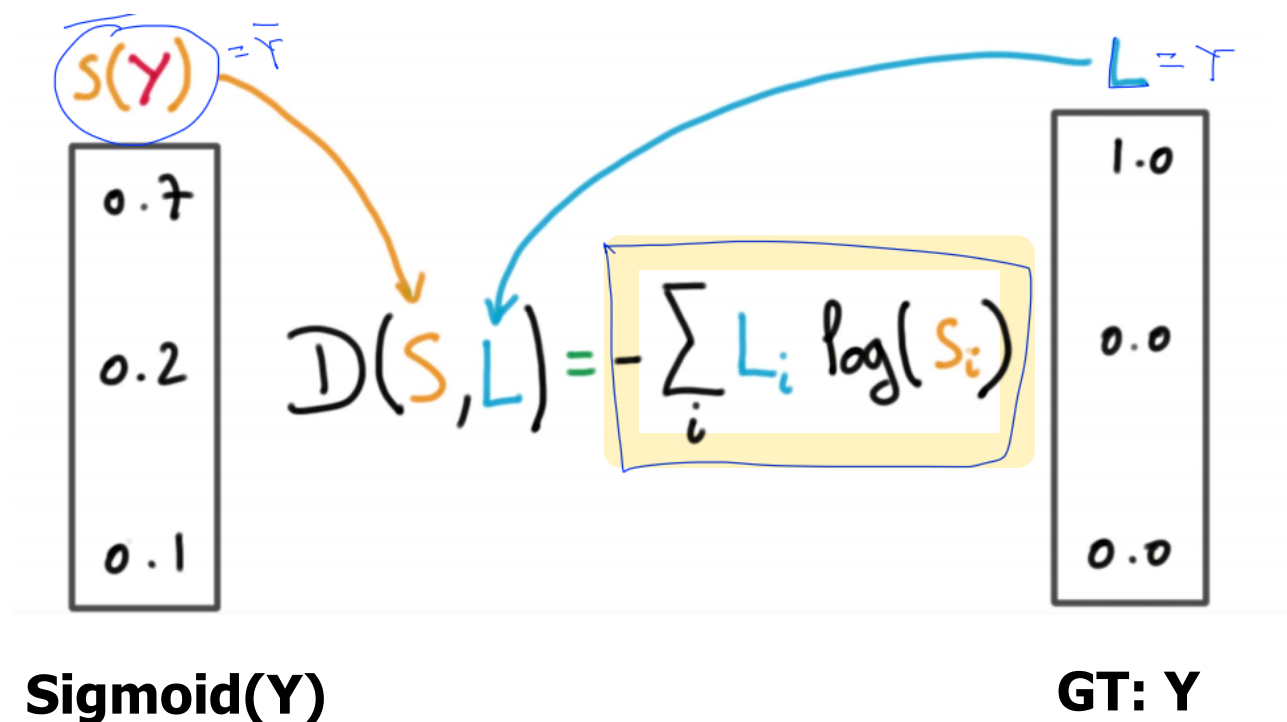**1) 모든 값이 0~1 사이**
2) 전체 합이 1 (확률 정규화!!)

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

SCORES ⟶ PROBABILITIES

2.0 → 0.7
1.0 → 0.2
0.1 → 0.1

argmax → one-hot encoding

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Softmax function (:=Probability)
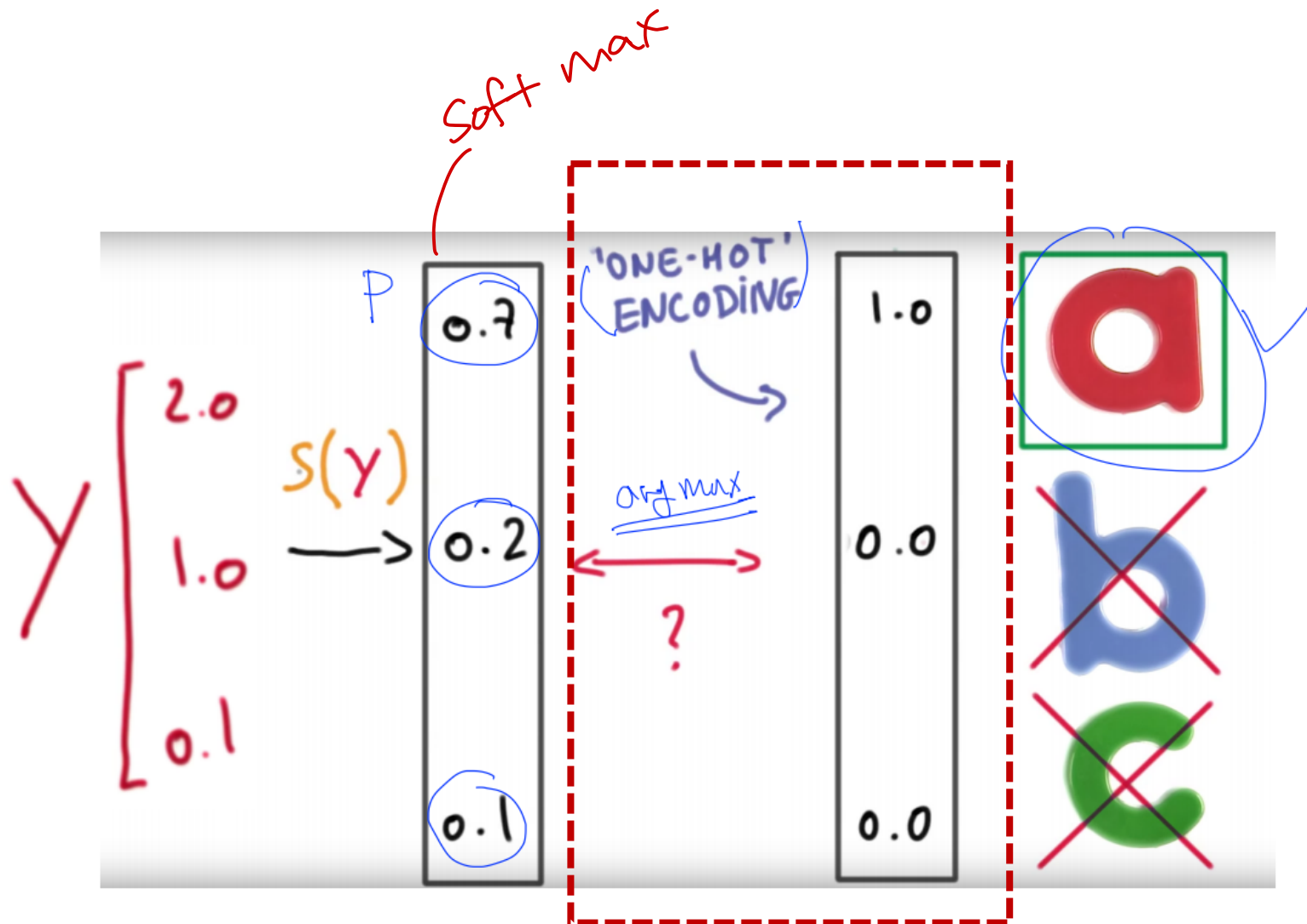
Cost function for multinomial classification

- **Cross-entropy**

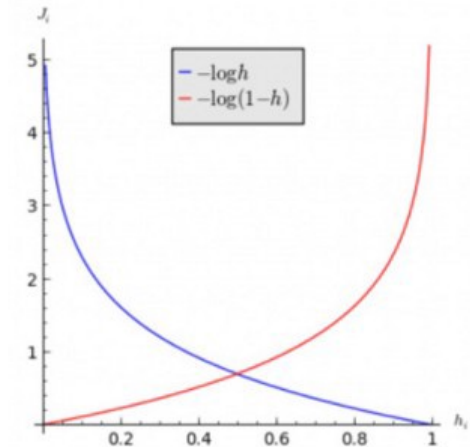**Sigmoid(Y)**   **GT: Y**

Cross-entropy cost function

$$-\sum_i L_i \log(S_i) = -\sum_i L_i \log(\bar{y}_i) = \sum_i L_i * -\log(\bar{y}_i)$$

Predict label

정답 $Y = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = B$

element-wise 곱

가설이 참 일때 $\bar{Y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = B, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot -log \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} \infty \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} => 0$
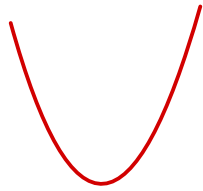
가설이 거짓일 때 $\bar{Y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = A, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot -log \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ \infty \end{bmatrix} = \begin{bmatrix} 0 \\ \infty \end{bmatrix} => \infty$
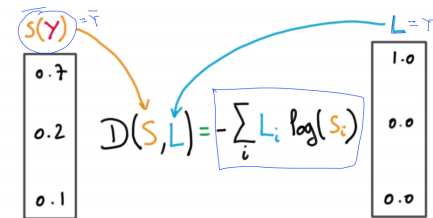
Cost function

Loss function $= \frac{1}{N}\sum_i D(S(WX_i + b), L_i)$

cross - entropy

Soft max   *GT*

$$D(S, L) = -\sum_i L_i \log(S_i)$$

: Cross entropy
= Logistic

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

: softmax

$$y = H(X) = \frac{1}{1 + e^{-W^T X}}$$

Sigmoid

$S(Y)$ $L = T$

0.7		1.0
0.2	$D(S,L) = -\sum_i L_i \log(S_i)$	0.0
0.1		0.0

$H_L(X) = WX$
$z = H_L(X)$, $g(z)$
$g(z) = \frac{1}{1 + e^{-z}}$
$H_R(X) = g(H_L(X))$