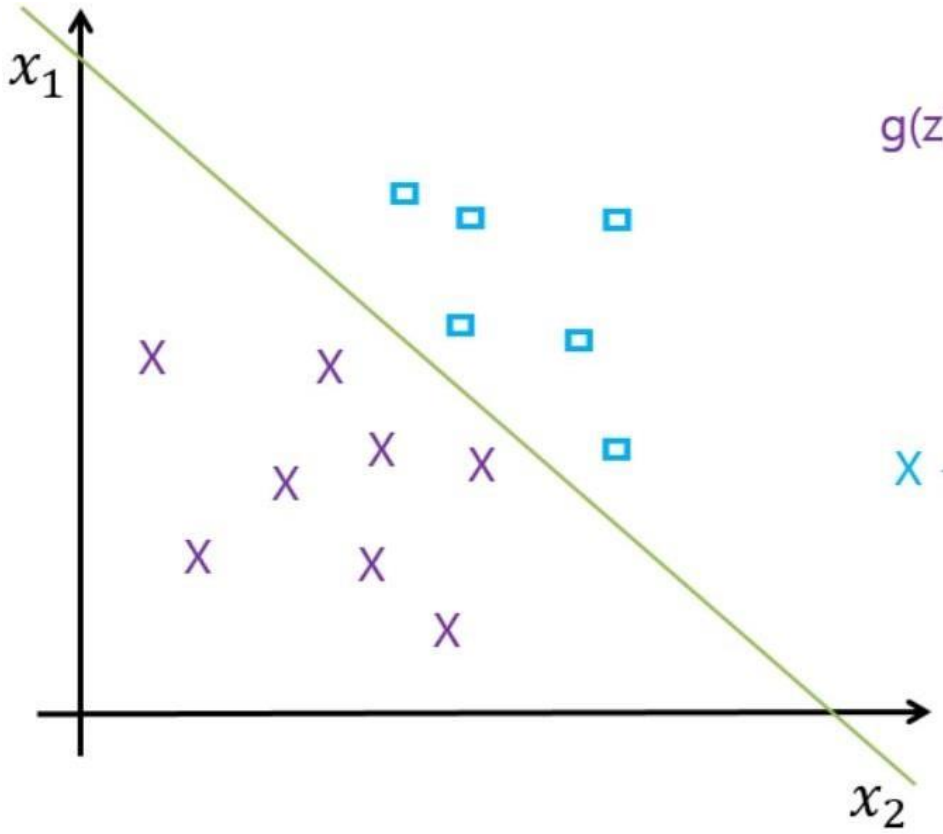
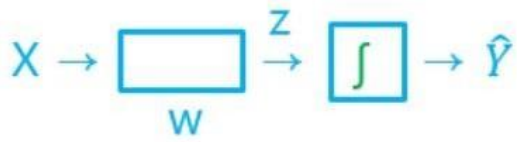
DEEP LEARNING

CROSS-ENTROPY

LOGISTIC REGRESSION

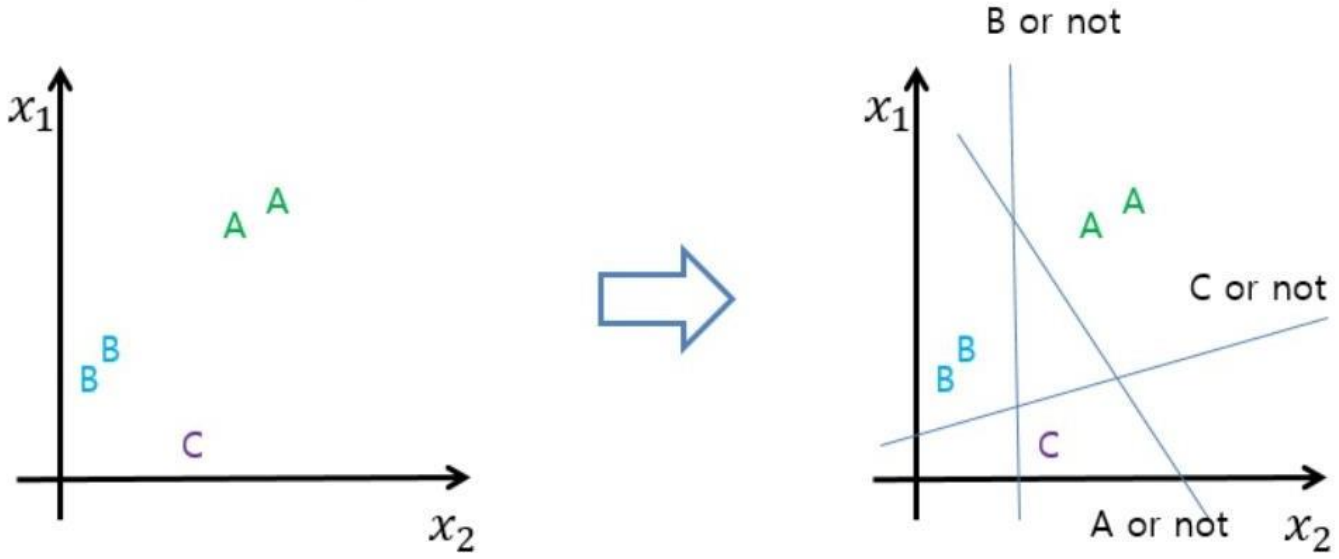


$$g(z)=\frac{1}{1+e^{-z}}$$

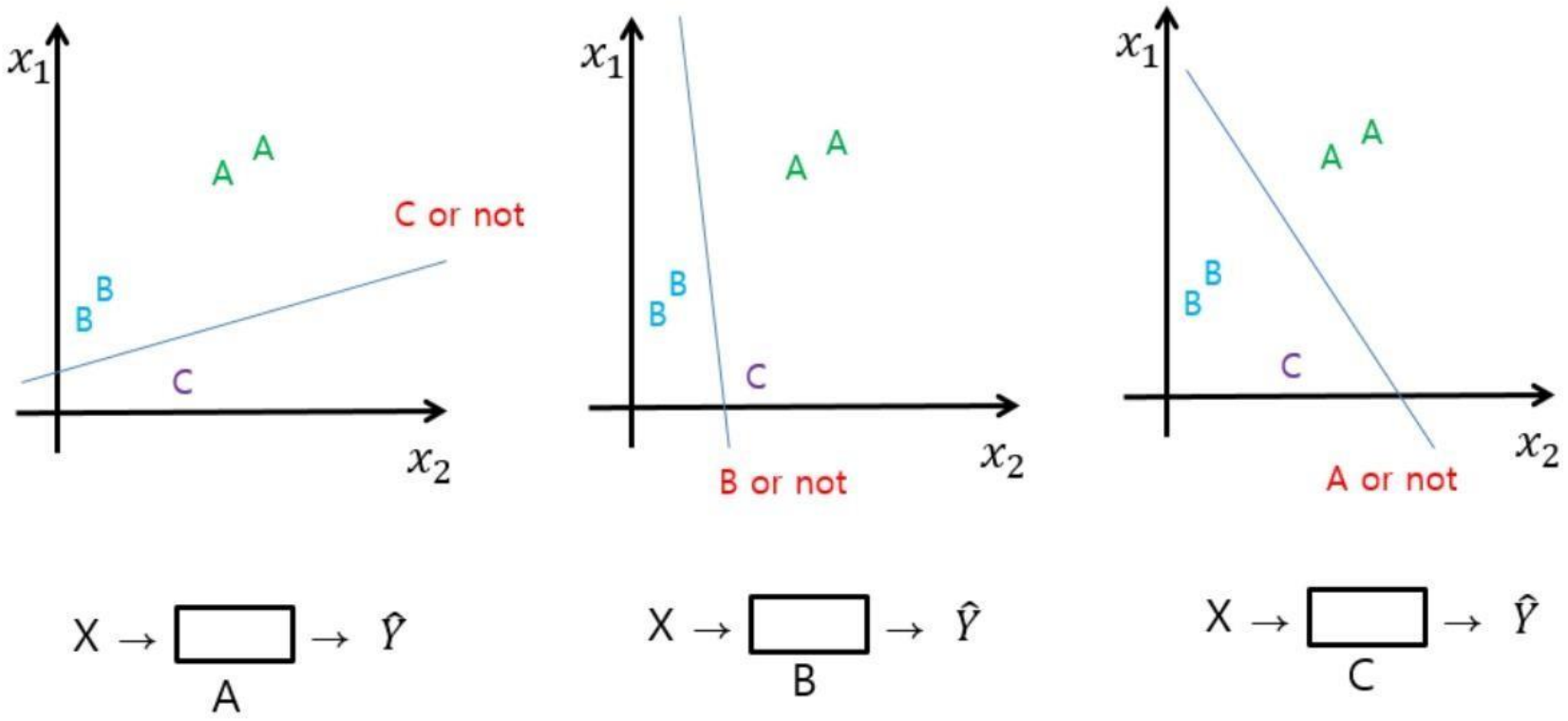


MULTINOMIAL CLASSIFICATION

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

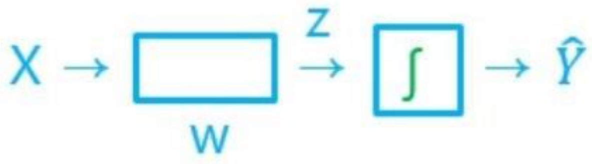


MULTINOMIAL CLASSIFICATION

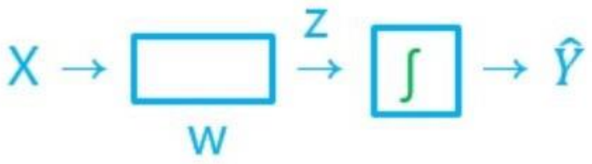


MULTINOMIAL CLASSIFICATION

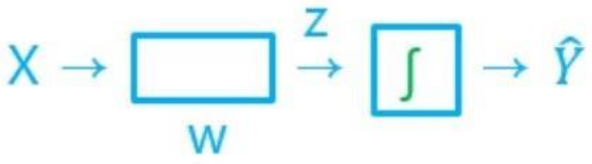
$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1 x_1 + w_2 x_2 + w_3 x_3 \end{bmatrix}$$



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



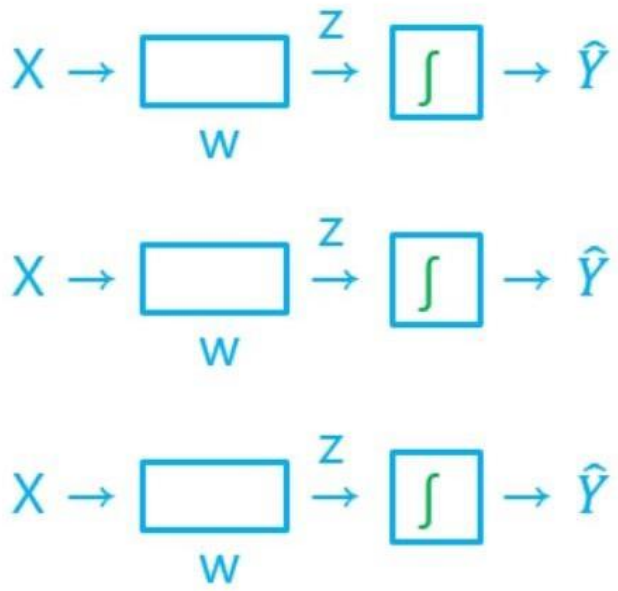
$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_1 x_1 + w_2 x_2 + w_3 x_3 \end{bmatrix}$$



MULTINOMIAL CLASSIFICATION

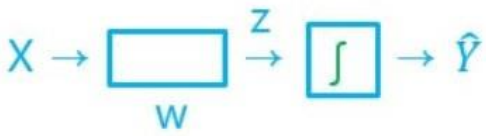
$$[w_1 \;\; w_2 \;\; w_3] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = [w_1 x_1 + w_2 x_2 + w_3 x_3]$$

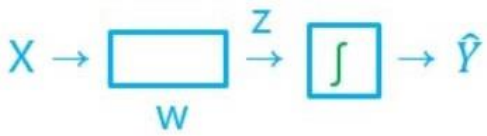
$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = ?$$

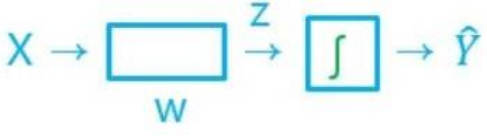


MULTINOMIAL CLASSIFICATION

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



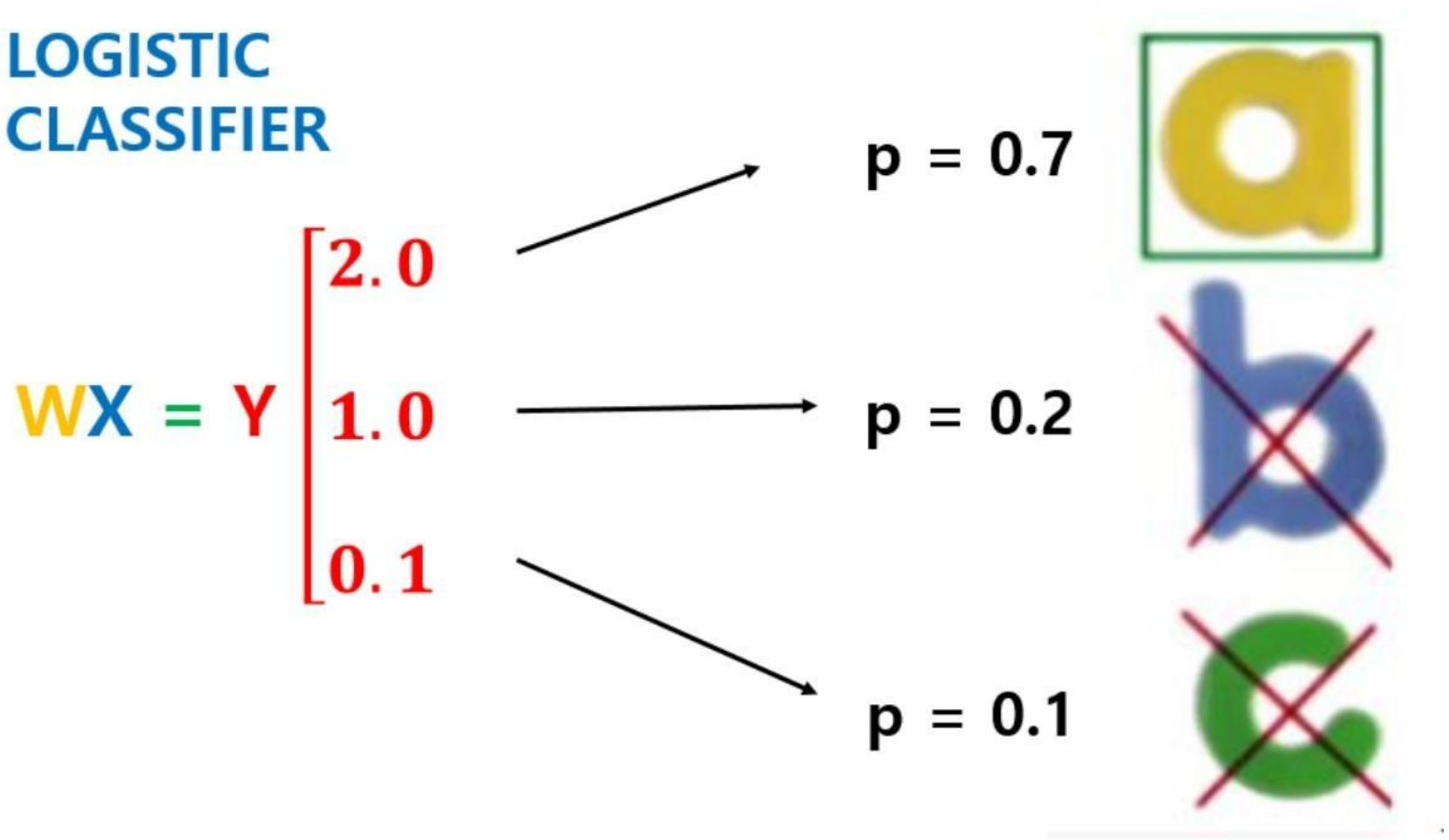




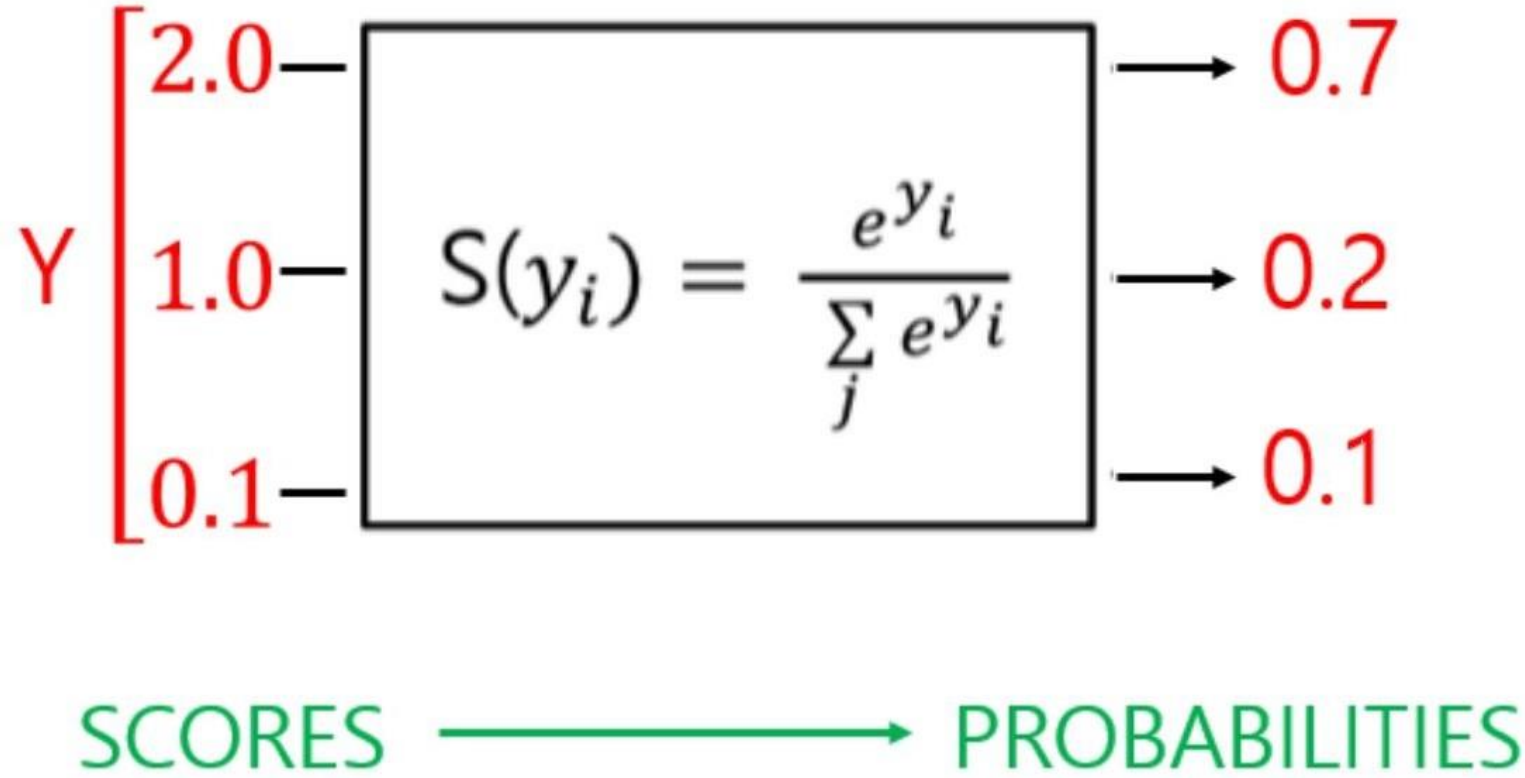
WHERE IS SIGMOID?

$$\begin{bmatrix} w_{A1} & w_{A2} & w_{A3} \\ w_{B1} & w_{B2} & w_{B3} \\ w_{C1} & w_{C2} & w_{C3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{A1}x_1 + w_{A2}x_2 + w_{A3}x_3 \\ w_{B1}x_1 + w_{B2}x_2 + w_{B3}x_3 \\ w_{C1}x_1 + w_{C2}x_2 + w_{C3}x_3 \end{bmatrix} = \begin{bmatrix} \hat{Y}_A \\ \hat{Y}_B \\ \hat{Y}_C \end{bmatrix} \begin{bmatrix} \mathbf{2.0} \\ \mathbf{1.0} \\ \mathbf{0.1} \end{bmatrix}$$

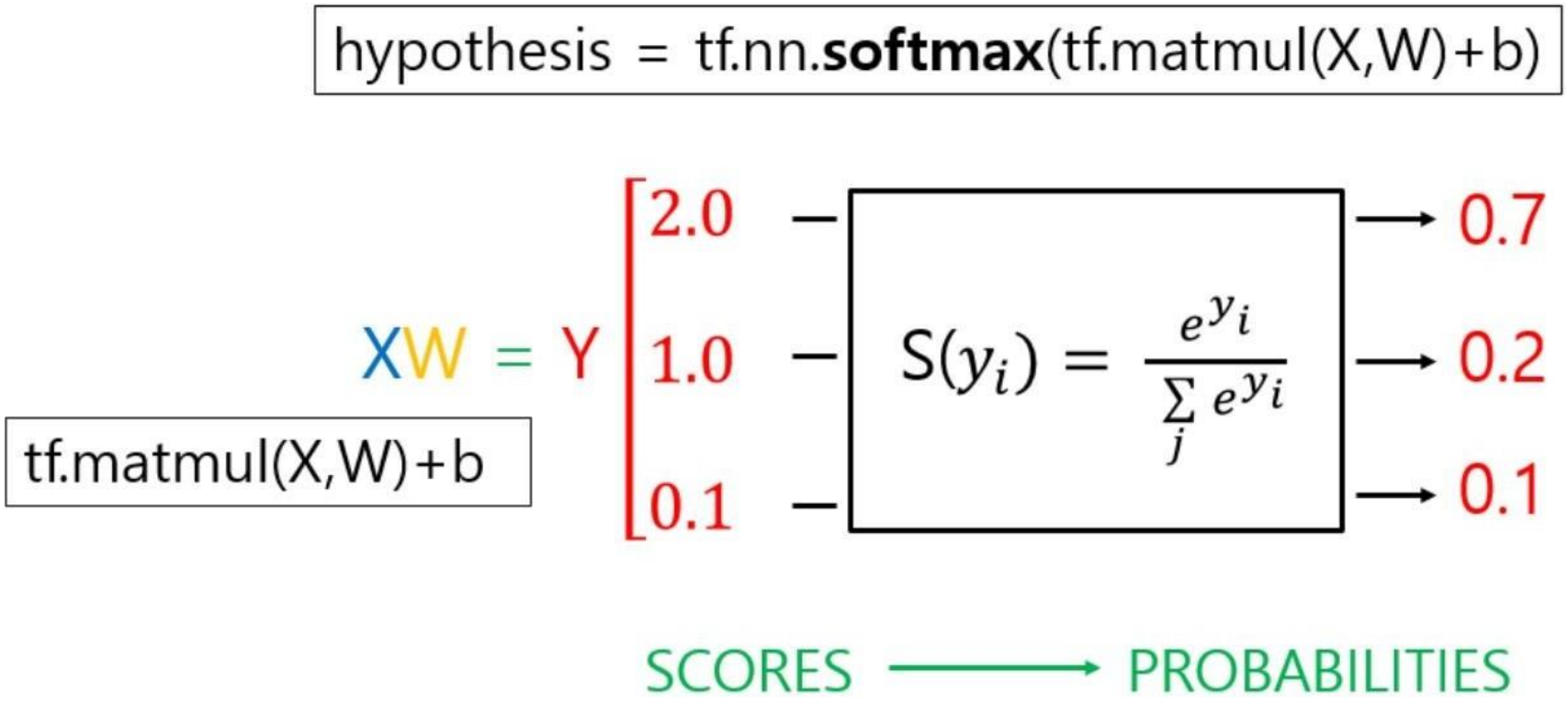
SIGMOID?



SOFTMAX



SOFTMAX


hypothesis = tf.nn.softmax(tf.matmul(X,W)+b)
XW = Y
tf.matmul(X,W)+b
2.0
1.0
0.1
$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_i}}$
0.7
0.2
0.1
SCORES
PROBABILITIES

SOFTMAX_CROSS_ENTROPY_WITH_LOGI

```
logits = tf.matmul(X, W) + b
hypothesis = tf.nn.softmax(logits)
```

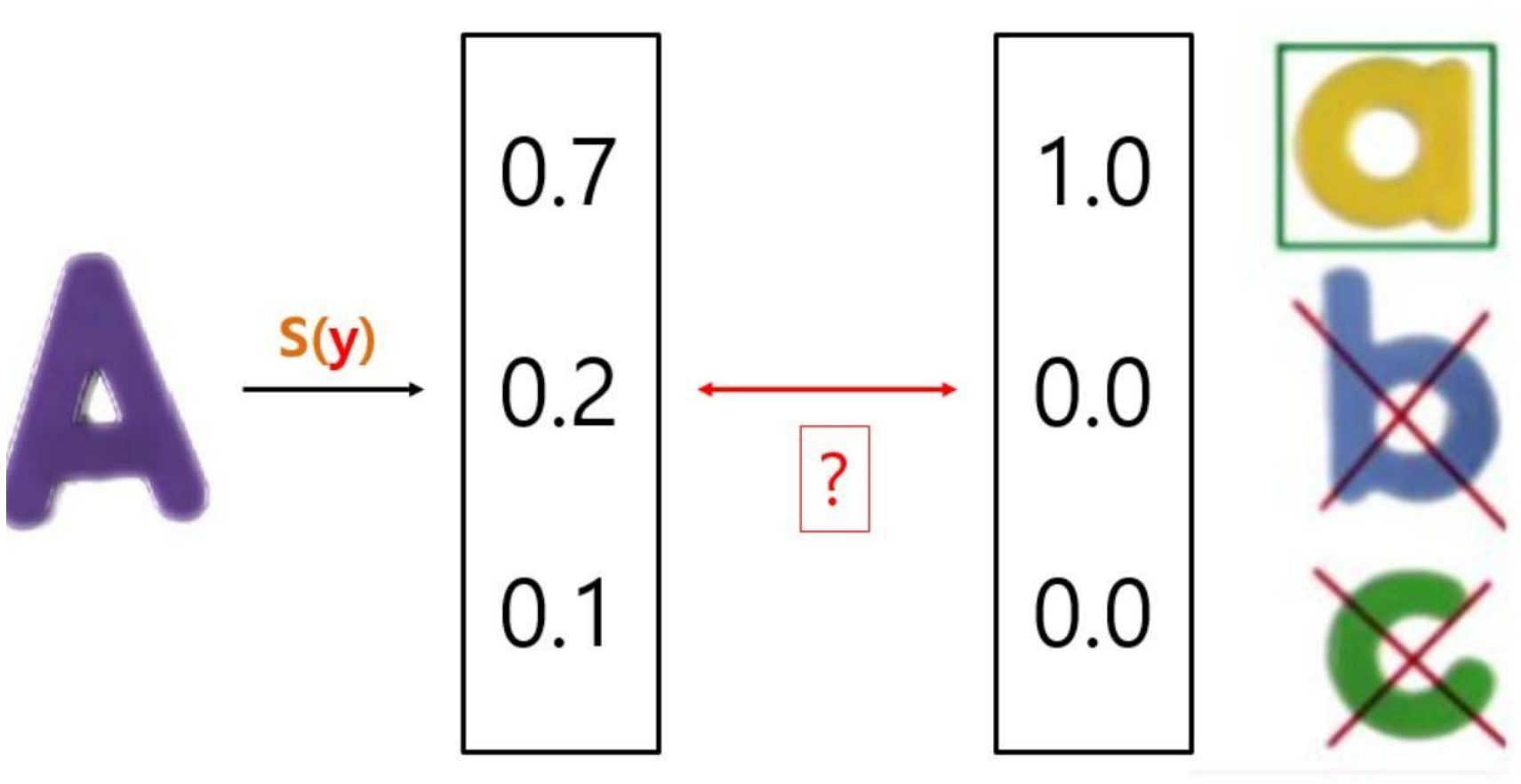
1

```
# Cross entropy cost/loss
cost = tf.reduce_mean(-tf.reduce_sum(Y * tf.log(hypothesis), axis=1))
```

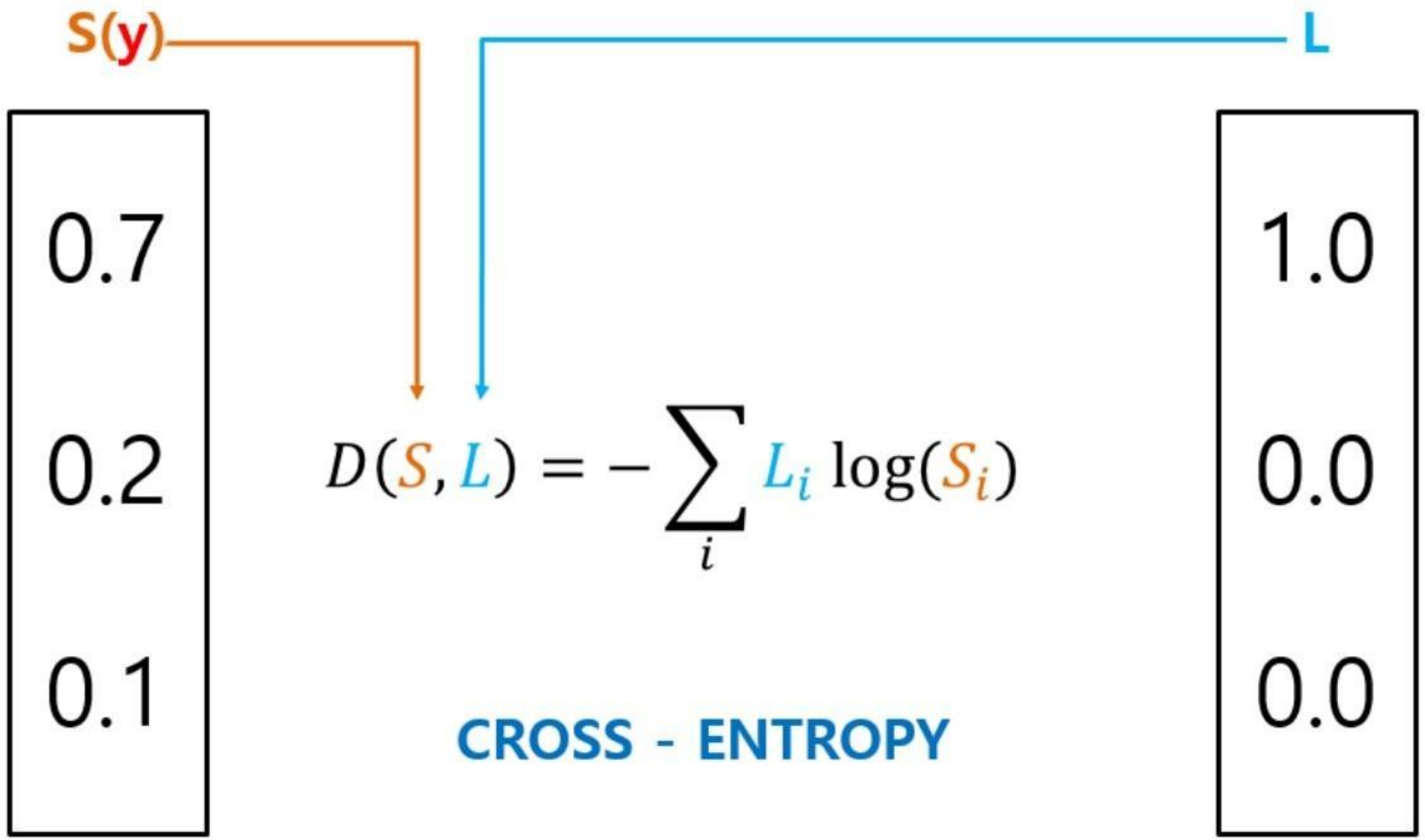
2

```
# Cross entropy cost/loss
cost_i = tf.nn.softmax_cross_entropy_with_logits(logits=logits,
                                                 labels=Y_one_hot)
cost = tf.reduce_mean(cost_i)
```

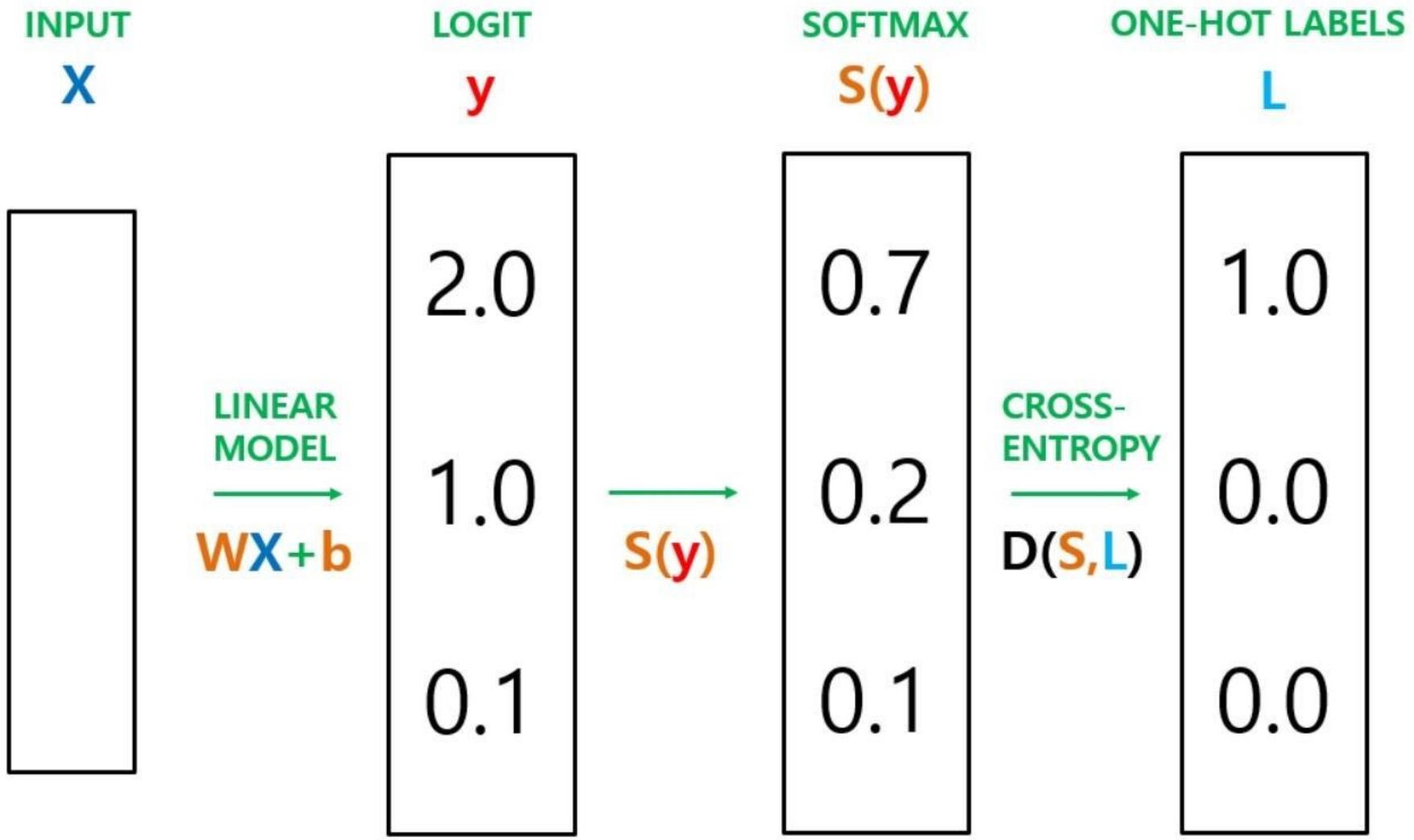
ONE-HOT ENCODING



COST FUNCTION

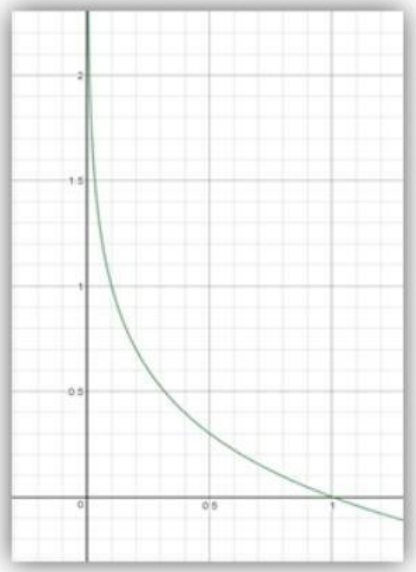


COST FUNCTION



CROSS-ENTROPY COST FUNCTION

$$-\sum_i L_i \log(S_i) \rightarrow -\sum_i L_i \log(\hat{y}_i) \rightarrow \sum_i L_i * -\log(\hat{y}_i)$$



$$L = \begin{matrix}\text{A}\\\text{B}\end{matrix}\begin{bmatrix}0\\1\end{bmatrix} = \text{B}$$

$$\hat{Y} = \begin{matrix}\text{A}\\\text{B}\end{matrix}\begin{bmatrix}0\\1\end{bmatrix} = \text{B}(0), \quad \begin{bmatrix}0\\1\end{bmatrix} * -\log\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} * \begin{bmatrix}\infty\\0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix} = 0$$

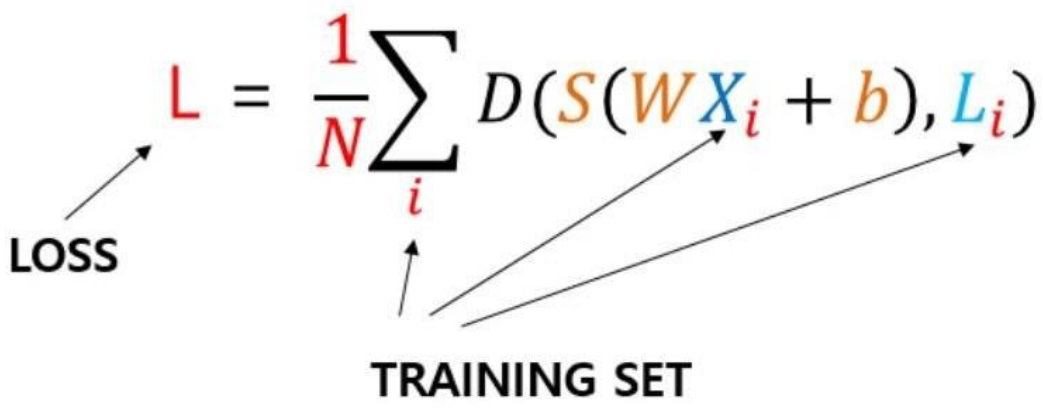
$$\hat{Y} = \begin{matrix}\text{A}\\\text{B}\end{matrix}\begin{bmatrix}1\\0\end{bmatrix} = \text{A}(\text{X}), \quad \begin{bmatrix}0\\1\end{bmatrix} * -\log\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} * \begin{bmatrix}0\\\infty\end{bmatrix} = \begin{bmatrix}0\\\infty\end{bmatrix} = \infty$$

LOGISTIC COST VS. CROSS ENTROPY

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

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

COST FUNCTION : CROSS ENTROPY




```
# Cross entropy cost/loss
cost = tf.reduce_mean(-tf.reduce_sum(Y * tf.log(hypothesis), axis=1))

optimizer = tf.train.GradientDescentOptimizer(learning_rate=0.1).minimize(cost)
```

ANIMAL CLASSIFICATION

with *softmax_cross_entropy_with_logits*