

# Classification: Logistic Regression

Hung-yi Lee

李宏毅

# 有關分組

- 作業以個人為單位繳交
- 期末專題才需要分組
- 找不到組員也沒有關係，期末專題公告後找不到組員的同學助教會幫忙湊對

# Step 1: Function Set

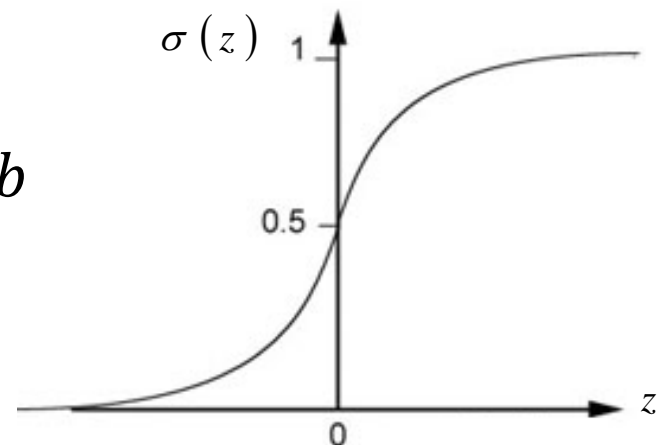
Function set: Including all different  $w$  and  $b$

$$\left\{ \begin{array}{ll} z \geq 0 & \text{class 1} \\ z < 0 & \text{class 2} \end{array} \right.$$

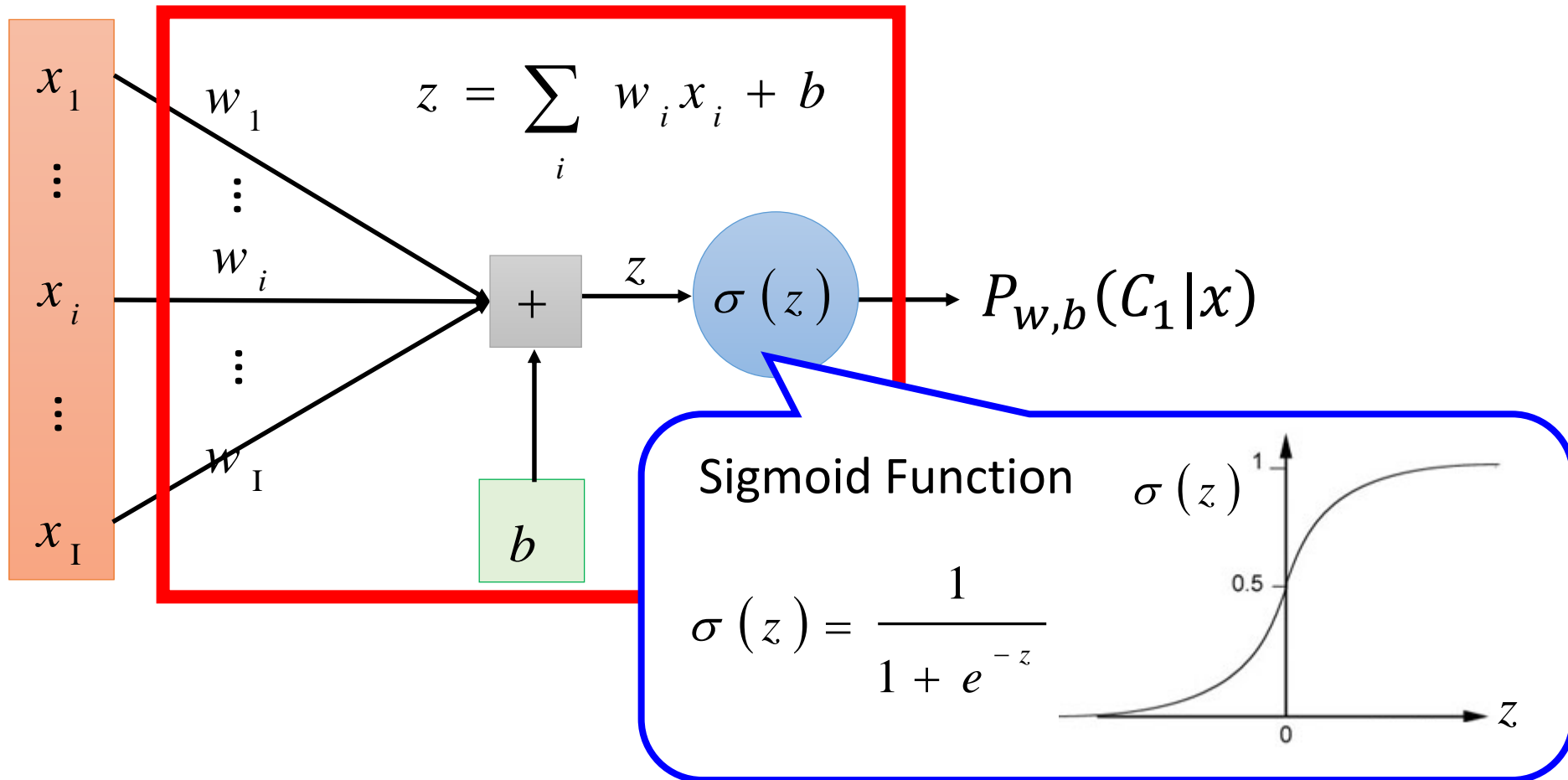
$$P_{w,b}(C_1|x) = \sigma(z)$$

$$z = w \cdot x + b = \sum_i w_i x_i + b$$

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



# Step 1: Function Set



## Step 2: Goodness of a Function

Training Data	$x^1$	$x^2$	$x^3$	...	$x^N$
	$C_1$	$C_1$	$C_2$	...	$C_1$

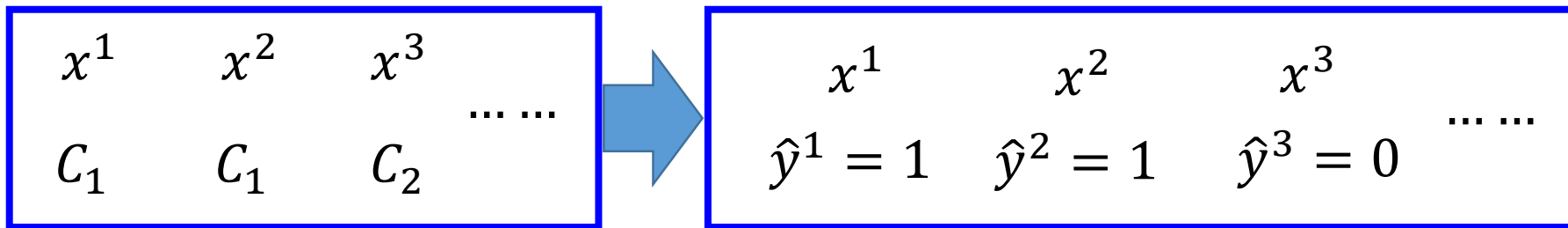
Assume the data is generated based on  $f_{w,b}(x) = P_{w,b}(C_1|x)$

Given a set of  $w$  and  $b$ , what is its probability of generating the data?

$$L(w, b) = f_{w,b}(x^1) f_{w,b}(x^2) (1 - f_{w,b}(x^3)) \cdots f_{w,b}(x^N)$$

The most likely  $w^*$  and  $b^*$  is the one with the largest  $L(w, b)$ .

$$w^*, b^* = \arg \max_{w, b} L(w, b)$$



$\hat{y}^n$ : 1 for class 1, 0 for class 2

$$L(w, b) = f_{w,b}(x^1) f_{w,b}(x^2) (1 - f_{w,b}(x^3)) \dots$$

$w^*, b^* = \arg \max_{w,b} L(w, b)$
--------------------------------------

 $=$ 

$w^*, b^* = \arg \min_{w,b} -\ln L(w, b)$
---

$$-\ln L(w, b)$$

$$= -\ln f_{w,b}(x^1) \longrightarrow -[ \boxed{1} \ln f(x^1) + \boxed{0} \ln(1 - f(x^1)) ]$$

$$-\ln f_{w,b}(x^2) \longrightarrow -[ \boxed{1} \ln f(x^2) + \boxed{0} \ln(1 - f(x^2)) ]$$

$$-\ln(1 - f_{w,b}(x^3)) \longrightarrow -[ \boxed{0} \ln f(x^3) + \boxed{1} \ln(1 - f(x^3)) ]$$

⋮

## Step 2: Goodness of a Function

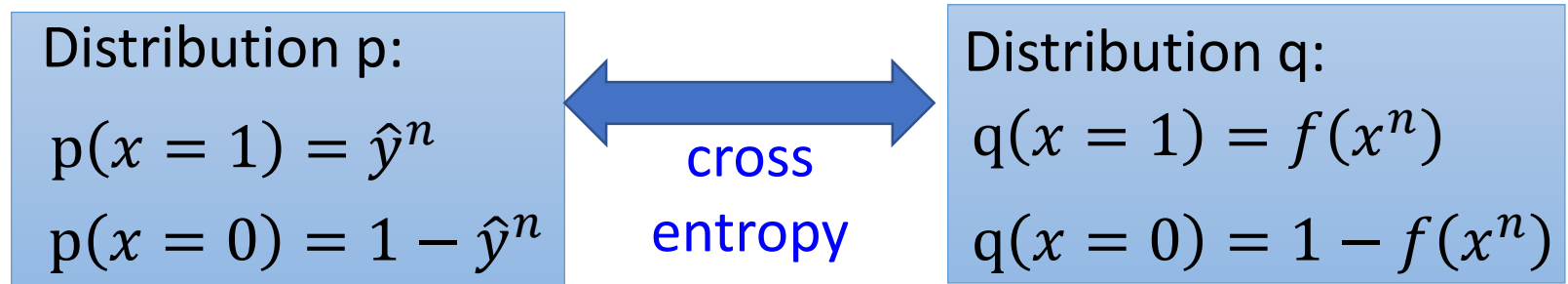
$$L(w, b) = f_{w,b}(x^1)f_{w,b}(x^2)(1 - f_{w,b}(x^3)) \cdots f_{w,b}(x^N)$$

$$-\ln L(w, b) = \ln f_{w,b}(x^1) + \ln f_{w,b}(x^2) + \ln(1 - f_{w,b}(x^3)) \cdots$$

$\hat{y}^n$ : 1 for class 1, 0 for class 2

$$= \sum_n - \left[ \hat{y}^n \ln f_{w,b}(x^n) + (1 - \hat{y}^n) \ln(1 - f_{w,b}(x^n)) \right]$$

Cross entropy between two Bernoulli distribution



$$H(p, q) = - \sum_x p(x) \ln(q(x))$$

## Step 2: Goodness of a Function

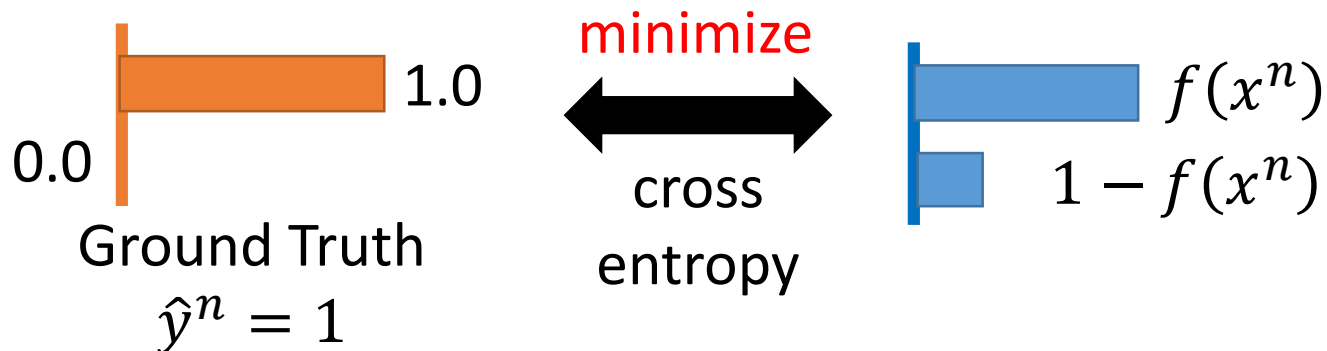
$$L(w, b) = f_{w,b}(x^1)f_{w,b}(x^2)(1 - f_{w,b}(x^3)) \cdots f_{w,b}(x^N)$$

$$-\ln L(w, b) = \ln f_{w,b}(x^1) + \ln f_{w,b}(x^2) + \ln(1 - f_{w,b}(x^3)) \cdots$$

$\hat{y}^n$ : 1 for class 1, 0 for class 2

$$= \sum_n - \left[ \hat{y}^n \ln f_{w,b}(x^n) + (1 - \hat{y}^n) \ln (1 - f_{w,b}(x^n)) \right]$$

Cross entropy between two Bernoulli distribution



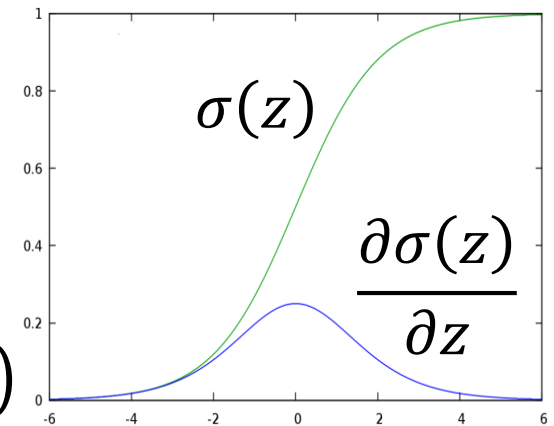


# Step 3: Find the best function

$$\frac{-\ln L(w, b)}{\partial w_i} = \sum_n - \left[ \hat{y}^n \frac{\left(1 - f_{w,b}(x^n)\right) x_i^n}{\partial w_i} + (1 - \hat{y}^n) \frac{\ln \left(1 - f_{w,b}(x^n)\right)}{\partial w_i} \right]$$

$$\frac{\partial \ln f_{w,b}(x)}{\partial w_i} = \frac{\partial \ln f_{w,b}(x)}{\partial z} \frac{\partial z}{\partial w_i} \quad \frac{\partial z}{\partial w_i} = x_i$$

$$\frac{\partial \ln \sigma(z)}{\partial z} = \frac{1}{\sigma(z)} \frac{\partial \sigma(z)}{\partial z} = \frac{1}{\cancel{\sigma(z)}} \cancel{\sigma(z)} (1 - \sigma(z))$$



$$\begin{aligned} f_{w,b}(x) &= \sigma(z) \\ &= 1 / (1 + \exp(-z)) \end{aligned}$$

$$z = w \cdot x + b = \sum_i w_i x_i + b$$

## Step 3: Find the best function

$$\frac{-\ln L(w, b)}{\partial w_i} = \sum_n - \left[ \hat{y}^n \frac{(1 - f_{w,b}(x^n)) x_i^n}{\partial w_i} + (1 - \hat{y}^n) \frac{-f_{w,b}(x^n) x_i^n}{\partial w_i} \right]$$

$$\frac{\partial \ln(1 - f_{w,b}(x))}{\partial w_i} = \frac{\partial \ln(1 - f_{w,b}(x))}{\partial z} \frac{\partial z}{\partial w_i} \quad \frac{\partial z}{\partial w_i} = x_i$$

$$\frac{\partial \ln(1 - \sigma(z))}{\partial z} = -\frac{1}{1 - \sigma(z)} \frac{\partial \sigma(z)}{\partial z} = -\frac{1}{1 - \sigma(z)} \sigma(z)(1 - \sigma(z))$$

$$\begin{aligned} f_{w,b}(x) &= \sigma(z) \\ &= 1 / (1 + \exp(-z)) \end{aligned}$$

$$z = w \cdot x + b = \sum_i w_i x_i + b$$

## Step 3: Find the best function

$$\frac{-\ln L(w, b)}{\partial w_i} = \sum_n - \left[ \hat{y}^n \frac{(1 - f_{w,b}(x^n)) x_i^n}{\partial w_i} + (1 - \hat{y}^n) \frac{-f_{w,b}(x^n) x_i^n}{\partial w_i} \right]$$

$$= \sum_n - \left[ \hat{y}^n \frac{(1 - f_{w,b}(x^n)) x_i^n}{\partial w_i} - (1 - \hat{y}^n) \frac{f_{w,b}(x^n) x_i^n}{\partial w_i} \right]$$

$$= \sum_n - \left[ \hat{y}^n - \cancel{\hat{y}^n f_{w,b}(x^n)} - f_{w,b}(x^n) + \cancel{\hat{y}^n f_{w,b}(x^n)} \right] \frac{x_i^n}{\partial w_i}$$

$$= \sum_n - \left( \hat{y}^n - f_{w,b}(x^n) \right) x_i^n$$

Larger difference, larger update

$$w_i \leftarrow w_i - \eta \sum_n - \left( \hat{y}^n - f_{w,b}(x^n) \right) x_i^n$$

## Logistic Regression + Square Error

Step 1:  $f_{w,b}(x) = \sigma \left( \sum_i w_i x_i + b \right)$

Step 2: Training data:  $(x^n, \hat{y}^n)$ ,  $\hat{y}^n$ : 1 for class 1, 0 for class 2

$$L(f) = \frac{1}{2} \sum_n (f_{w,b}(x^n) - \hat{y}^n)^2$$

Step 3:

$$\frac{\partial (f_{w,b}(x) - \hat{y})^2}{\partial w_i} = 2(f_{w,b}(x) - \hat{y}) \frac{\partial f_{w,b}(x)}{\partial z} \frac{\partial z}{\partial w_i}$$
$$= 2(f_{w,b}(x) - \hat{y}) f_{w,b}(x) (1 - f_{w,b}(x)) x_i$$

$\hat{y}^n = 1$  If  $f_{w,b}(x^n) = 1$  (close to target)   $\partial L / \partial w_i = 0$

If  $f_{w,b}(x^n) = 0$  (far from target)   $\partial L / \partial w_i = 0$

## Logistic Regression + Square Error

Step 1:  $f_{w,b}(x) = \sigma \left( \sum_i w_i x_i + b \right)$

Step 2: Training data:  $(x^n, \hat{y}^n)$ ,  $\hat{y}^n$ : 1 for class 1, 0 for class 2

$$L(f) = \frac{1}{2} \sum_n (f_{w,b}(x^n) - \hat{y}^n)^2$$

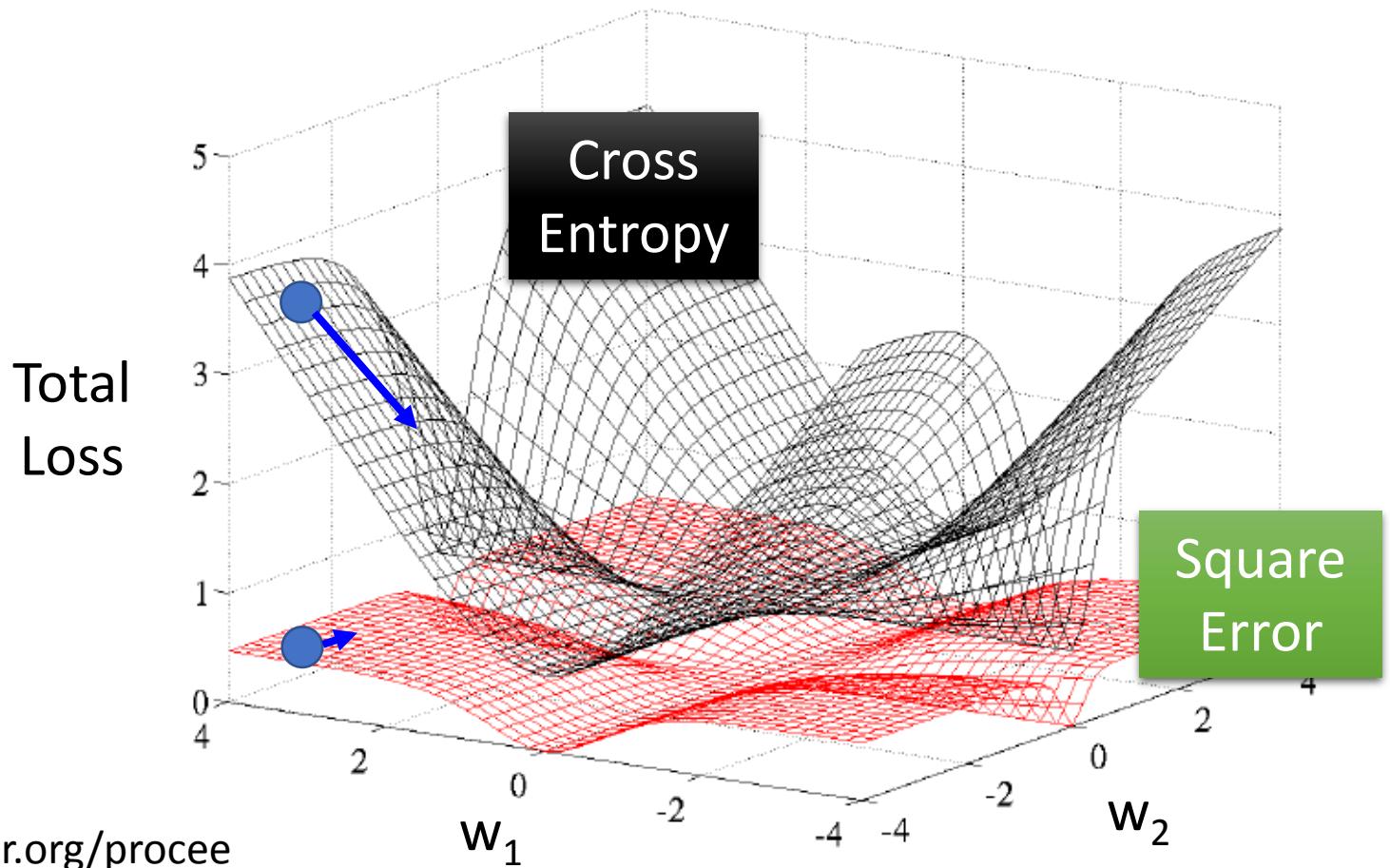
Step 3:

$$\frac{\partial (f_{w,b}(x) - \hat{y})^2}{\partial w_i} = 2(f_{w,b}(x) - \hat{y}) \frac{\partial f_{w,b}(x)}{\partial z} \frac{\partial z}{\partial w_i}$$
$$= 2(f_{w,b}(x) - \hat{y}) f_{w,b}(x) (1 - f_{w,b}(x)) x_i$$

$\hat{y}^n = 0$  If  $f_{w,b}(x^n) = 1$  (far from target)  $\Rightarrow \partial L / \partial w_i = 0$

If  $f_{w,b}(x^n) = 0$  (close to target)  $\Rightarrow \partial L / \partial w_i = 0$

# Cross Entropy v.s. Square Error



<http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf>

## **Logistic Regression**

Step 1:  $f_{w,b}(x) = \sigma \left( \sum_i w_i x_i + b \right)$

Output: between 0 and 1

## **Linear Regression**

$$f_{w,b}(x) = \sum_i w_i x_i + b$$

Output: any value

Step 2:

Step 3:

## Logistic Regression

Step 1:  $f_{w,b}(x) = \sigma \left( \sum_i w_i x_i + b \right)$

Output: between 0 and 1

Training data:  $(x^n, \hat{y}^n)$

Step 2:  $\hat{y}^n$ : 1 for class 1, 0 for class 2

$$L(f) = \sum_n l(f(x^n), \hat{y}^n)$$

## Linear Regression

$$f_{w,b}(x) = \sum_i w_i x_i + b$$

Output: any value

Training data:  $(x^n, \hat{y}^n)$

$\hat{y}^n$ : a real number

$$L(f) = \frac{1}{2} \sum_n (f(x^n) - \hat{y}^n)^2$$

Cross entropy:

$$l(f(x^n), \hat{y}^n) = -[\hat{y}^n \ln f(x^n) + (1 - \hat{y}^n) \ln(1 - f(x^n))]$$



## Logistic Regression

Step 1:  $f_{w,b}(x) = \sigma \left( \sum_i w_i x_i + b \right)$

Output: between 0 and 1

Training data:  $(x^n, \hat{y}^n)$

Step 2:  $\hat{y}^n$ : 1 for class 1, 0 for class 2

$$L(f) = \sum_n l(f(x^n), \hat{y}^n)$$

## Linear Regression

$$f_{w,b}(x) = \sum_i w_i x_i + b$$

Output: any value

Training data:  $(x^n, \hat{y}^n)$

$\hat{y}^n$ : a real number

$$L(f) = \frac{1}{2} \sum_n (f(x^n) - \hat{y}^n)^2$$

Logistic regression:  $w_i \leftarrow w_i - \eta \sum_n - \left( \hat{y}^n - f_{w,b}(x^n) \right) x_i^n$

Step 3:

Linear regression:  $w_i \leftarrow w_i - \eta \sum_n - \left( \hat{y}^n - f_{w,b}(x^n) \right) x_i^n$

# Discriminative v.s. Generative

$$P(C_1|x) = \sigma(w \cdot x + b)$$



directly find  $w$  and  $b$

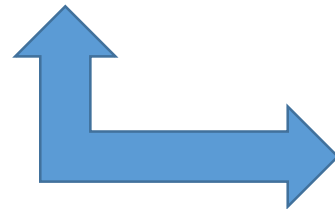


Find  $\mu^1, \mu^2, \Sigma^{-1}$

$$w^T = (\mu^1 - \mu^2)^T \Sigma^{-1}$$

$$b = -\frac{1}{2} (\mu^1)^T (\Sigma^1)^{-1} \mu^1$$

$$+ \frac{1}{2} (\mu^2)^T (\Sigma^2)^{-1} \mu^2 + \ln \frac{N_1}{N_2}$$

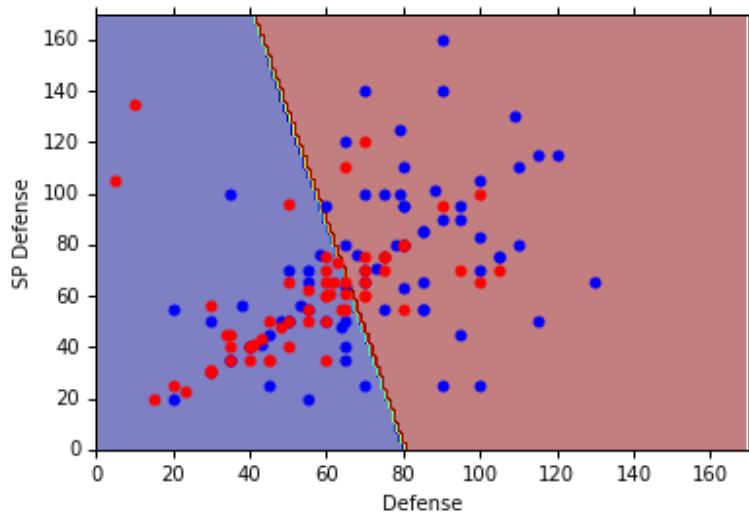


Will we obtain the same set of  $w$  and  $b$ ?

The same model (function set), but different function may be selected by the same training data.

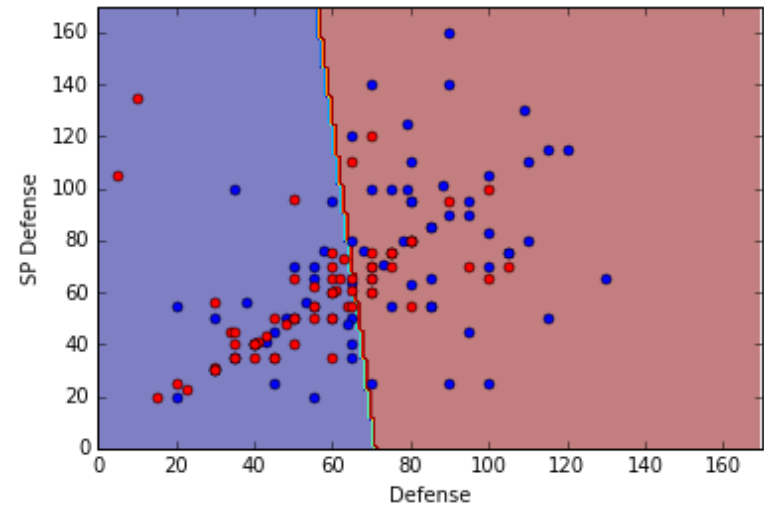
# Generative v.s. Discriminative

**Generative**



All: hp, att, sp att, de, sp de, speed  
73% accuracy

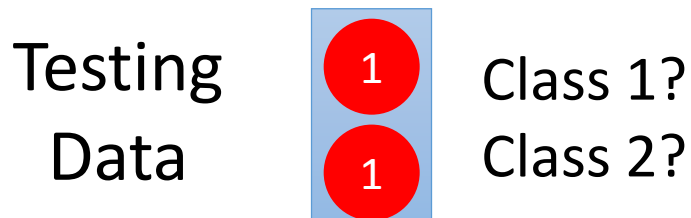
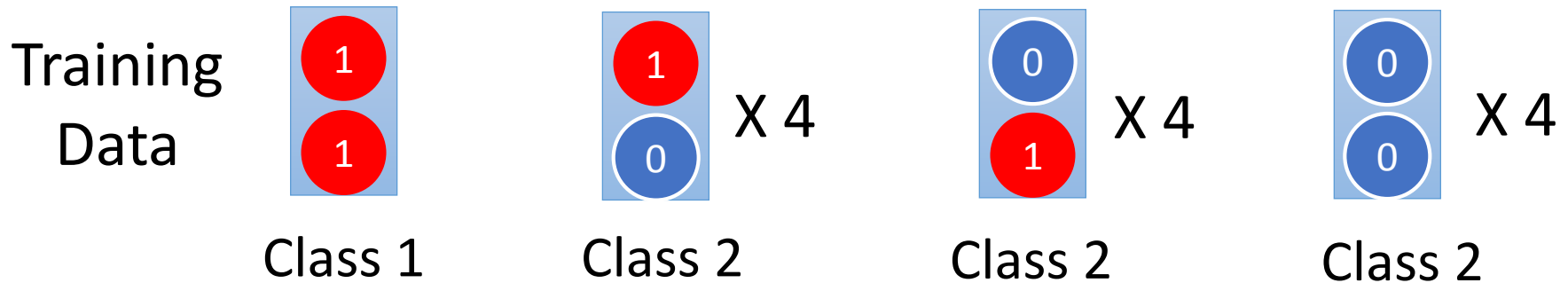
**Discriminative**



79% accuracy

# Generative v.s. Discriminative

- Example

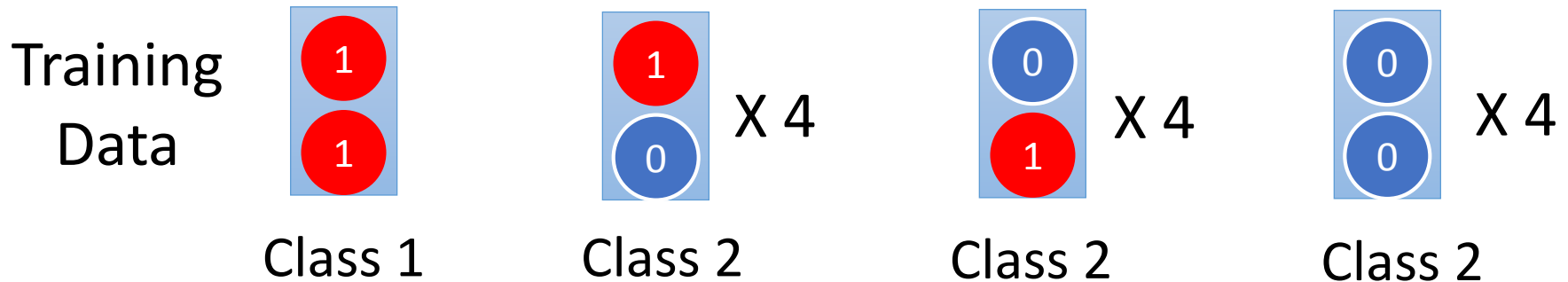


How about Naïve Bayes?

$$P(x|C_i) = P(x_1|C_i)P(x_2|C_i)$$

# Generative v.s. Discriminative

- Example



$$P(C_1) = \frac{1}{13}$$

$$P(x_1 = 1|C_1) = 1$$

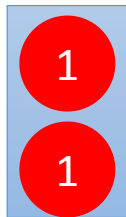
$$P(x_2 = 1|C_1) = 1$$

$$P(C_2) = \frac{12}{13}$$

$$P(x_1 = 1|C_2) = \frac{1}{3}$$

$$P(x_2 = 1|C_2) = \frac{1}{3}$$

Training  
Data

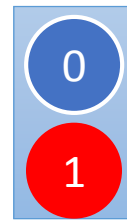


Class 1



Class 2

X 4



Class 2

X 4

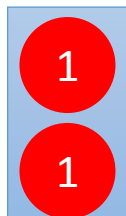


Class 2

X 4

naïve model认为第一个feature与第二个feature产生是独立的，所以在class2中是可能产生1,1的，虽然training data里没有出现过1,1

Testing  
Data



$P(C_1|x)$

<0.5

$$P(C_1|x) = \frac{P(x|C_1)P(C_1)}{P(x|C_1)P(C_1) + P(x|C_2)P(C_2)}$$

Diagram illustrating the calculation of the numerator and denominator terms:

- $P(x|C_1) = 1 \times 1$  (indicated by blue arrows from the top-left  $1 \times 1$  to the  $P(x|C_1)$  term in the fraction)
- $P(C_1) = \frac{1}{13}$  (indicated by a blue arrow from the top-right  $\frac{1}{13}$  to the  $P(C_1)$  term in the fraction)
- $P(x|C_2) = \frac{1}{3} \times \frac{1}{3}$  (indicated by blue arrows from the bottom-left  $1 \times 1$  and bottom-middle  $\frac{1}{13}$  to the  $P(x|C_2)$  term in the fraction)
- $P(C_2) = \frac{12}{13}$  (indicated by a blue arrow from the bottom-right  $\frac{12}{13}$  to the  $P(C_2)$  term in the fraction)

$$P(C_1) = \frac{1}{13}$$

$$P(x_1 = 1|C_1) = 1$$

$$P(x_2 = 1|C_1) = 1$$

$$P(C_2) = \frac{12}{13}$$

$$P(x_1 = 1|C_2) = \frac{1}{3}$$

$$P(x_2 = 1|C_2) = \frac{1}{3}$$

# Generative v.s. Discriminative

- Usually people believe discriminative model is better
- Benefit of generative model
  - With the assumption of probability distribution
    - less training data is needed
    - more robust to the noise
  - Priors and class-dependent probabilities can be estimated from different sources.

# Multi-class Classification (3 classes as example)

$$C_1: w^1, b_1 \quad z_1 = w^1 \cdot x + b_1$$

$$C_2: w^2, b_2 \quad z_2 = w^2 \cdot x + b_2$$

$$C_3: w^3, b_3 \quad z_3 = w^3 \cdot x + b_3$$

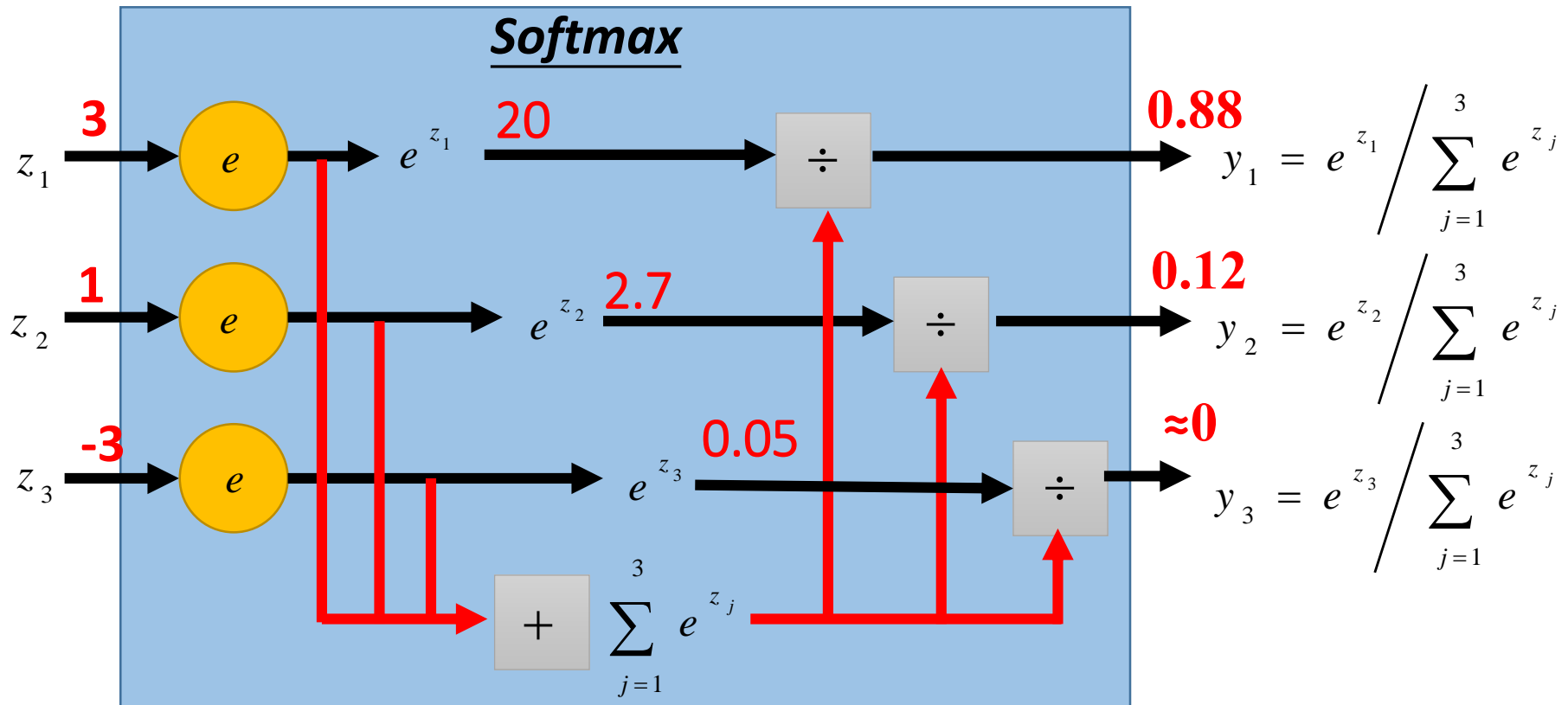
## Probability:

$$\blacksquare 1 > y_i > 0$$

$$\blacksquare \sum_i y_i = 1$$

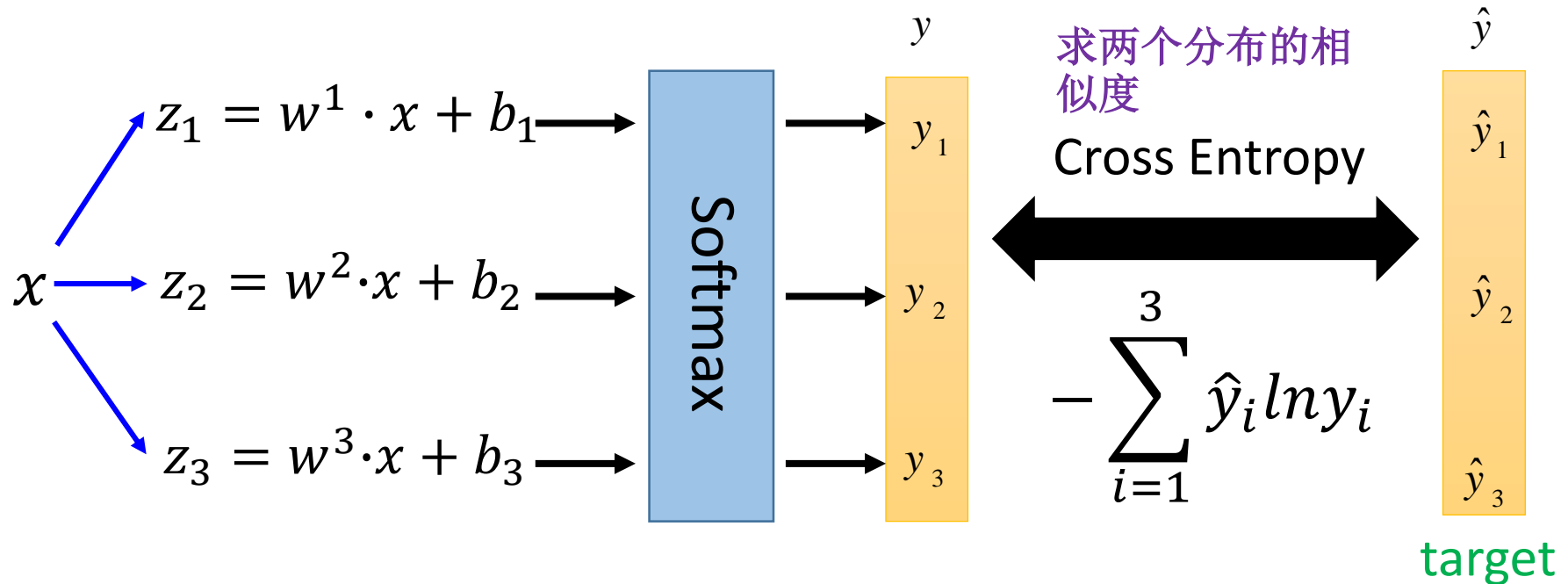
为什么多分类问题的损失函数要用 **Softmax**函数，如果我们假设 $C_1, C_2, C_3$ 都是高斯分布，共并他们共用了一个**Sigma**，那么就可以用最大似然估计推导出来。

$$y_i = P(C_i | x)$$





# Multi-class Classification (3 classes as example)



If  $x \in \text{class 1}$

$$\hat{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$-\ln y_1$$

If  $x \in \text{class 2}$

$$\hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

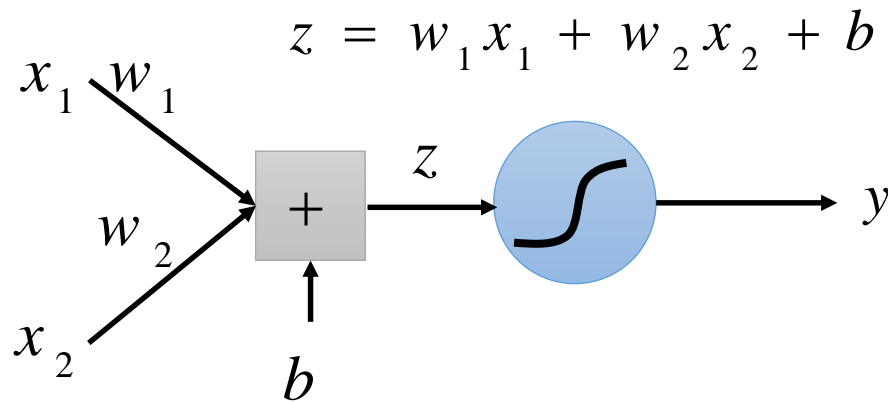
$$-\ln y_2$$

If  $x \in \text{class 3}$

$$\hat{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

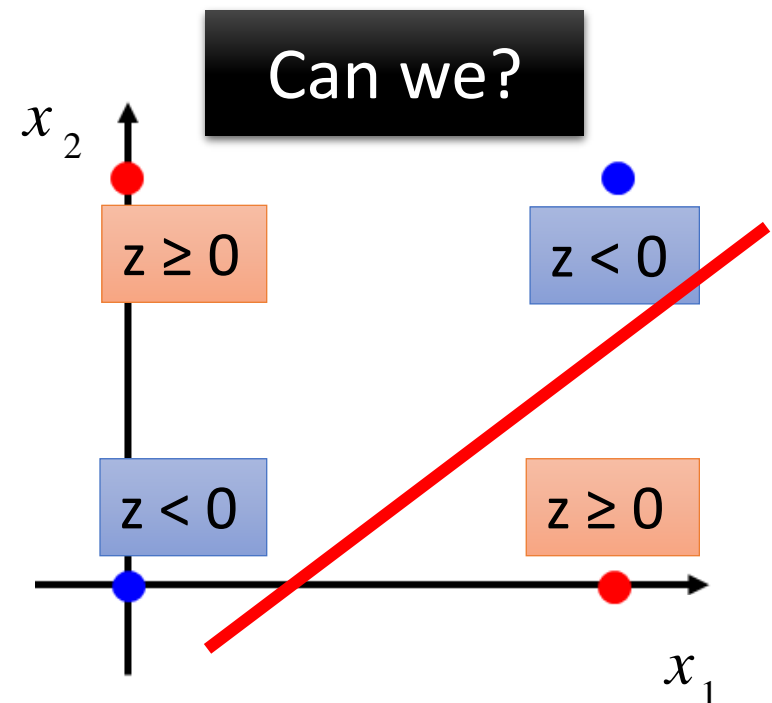
$$-\ln y_3$$

# Limitation of Logistic Regression



$$\begin{cases} \text{Class 1} & y \geq 0.5 \quad (z \geq 0) \\ \text{Class 2} & y < 0.5 \quad (z < 0) \end{cases}$$

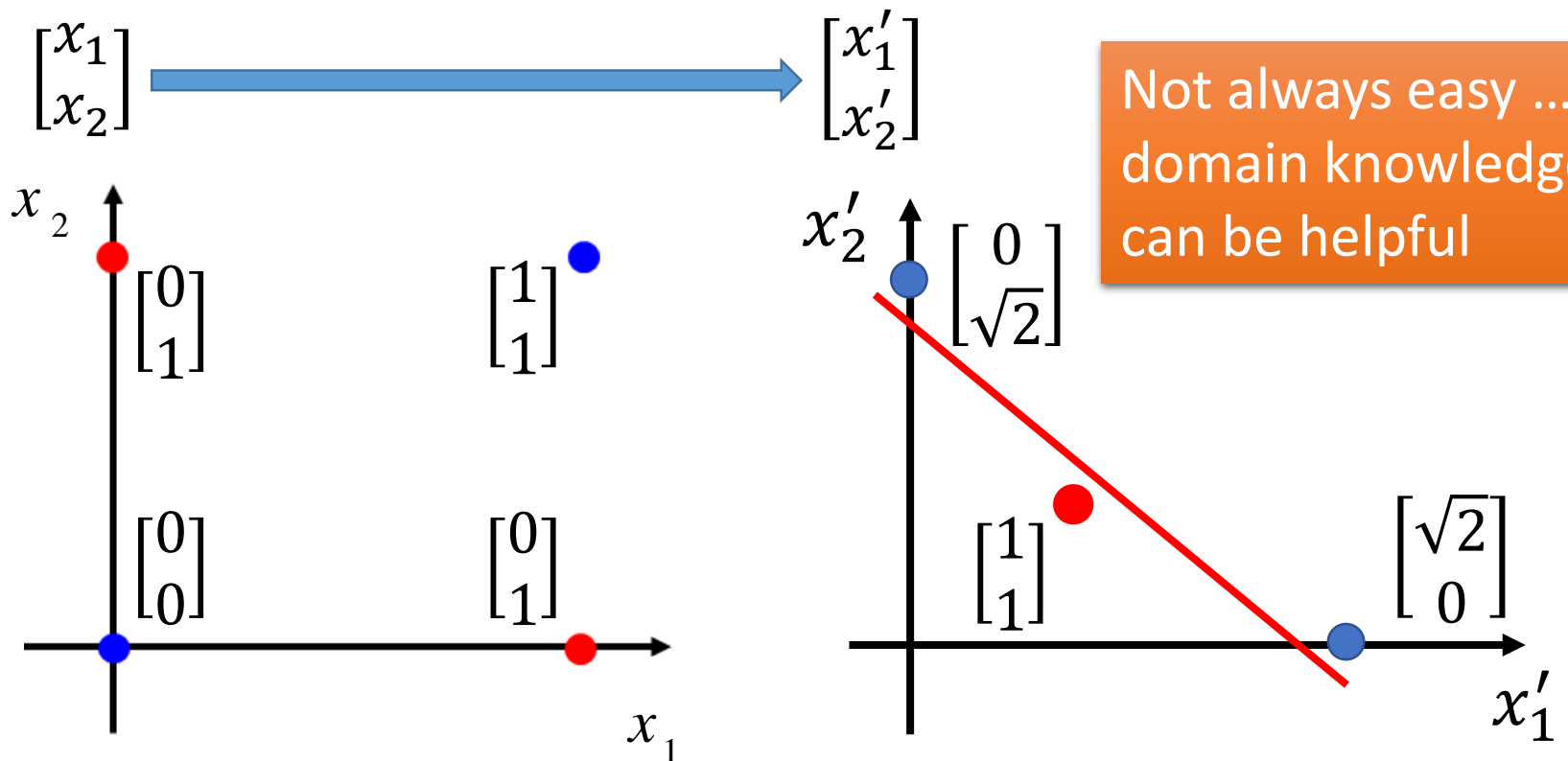
Input Feature		Label
$x_1$	$x_2$	
0	0	Class 2
0	1	Class 1
1	0	Class 1
1	1	Class 2



# Limitation of Logistic Regression

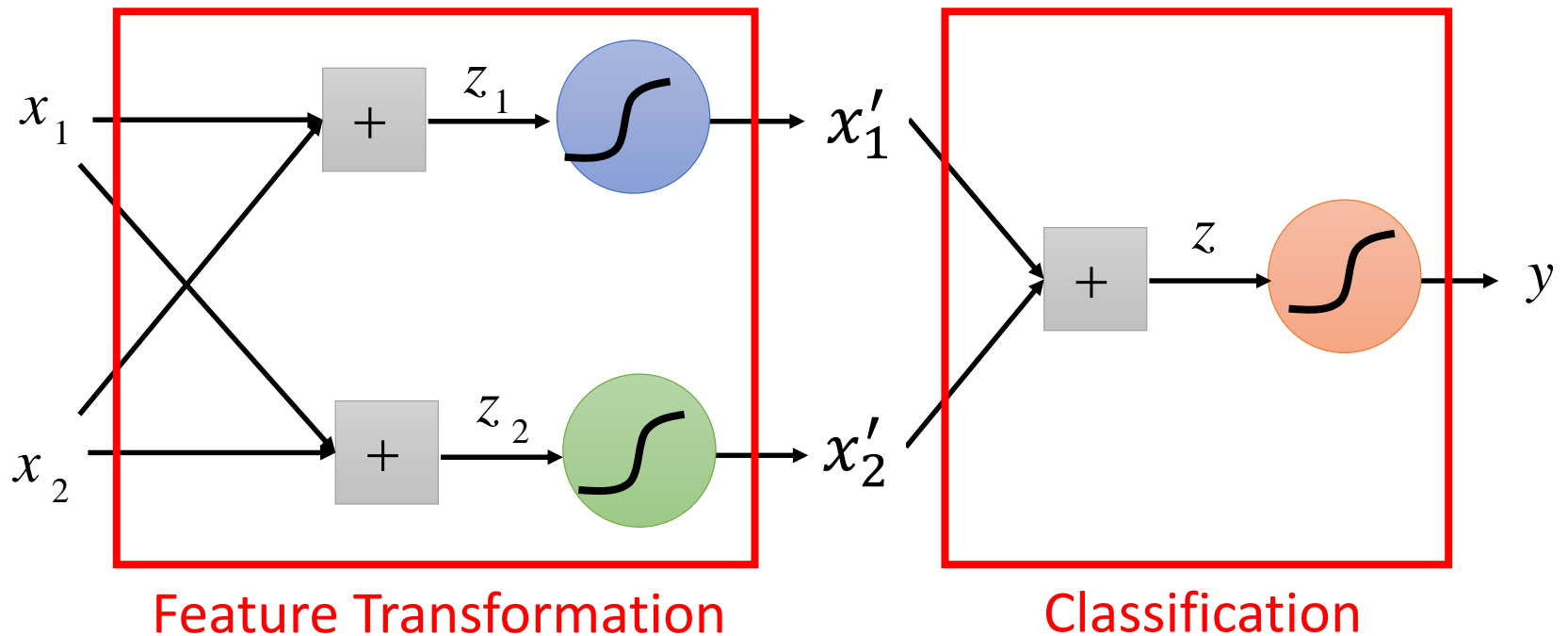
- **Feature transformation**

$x'_1$ : distance to  $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$   
 $x'_2$ : distance to  $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

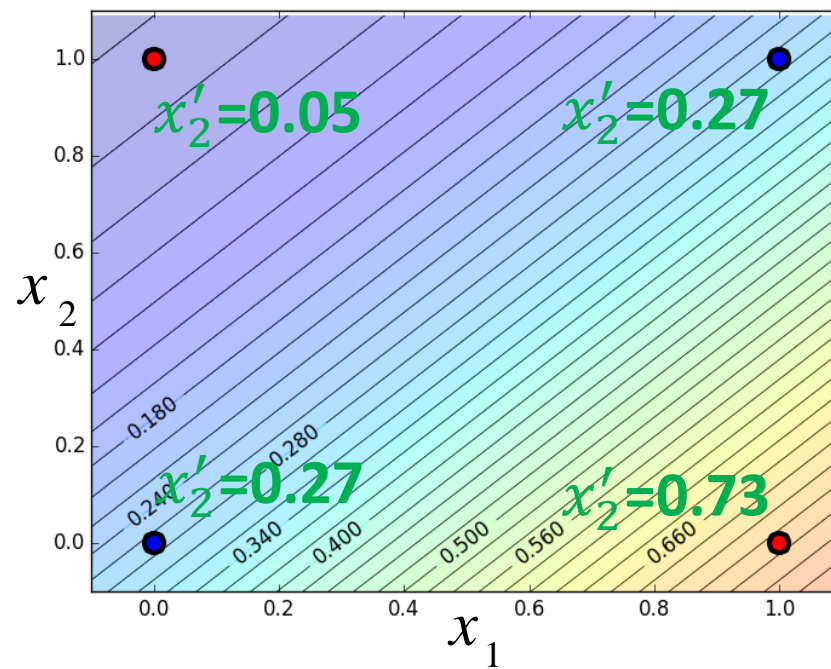
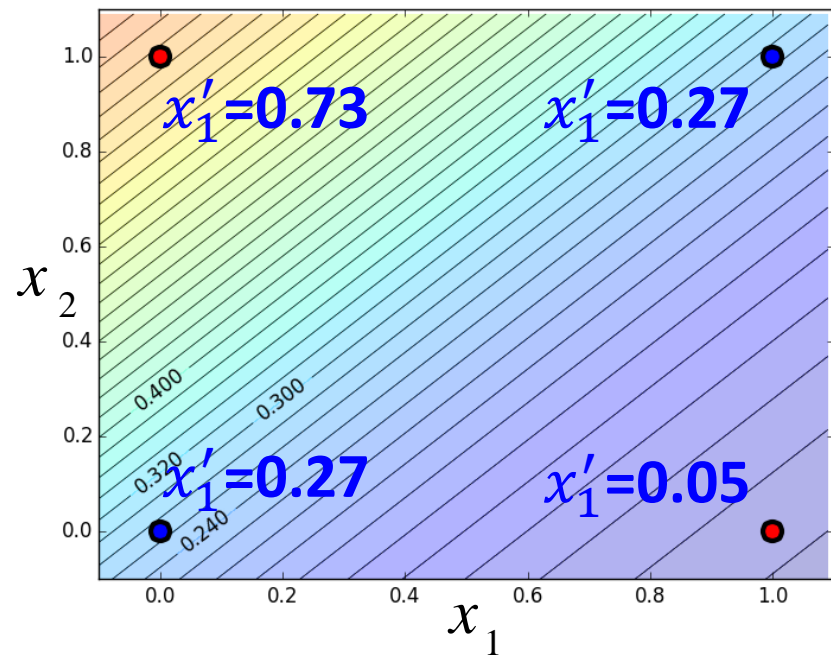
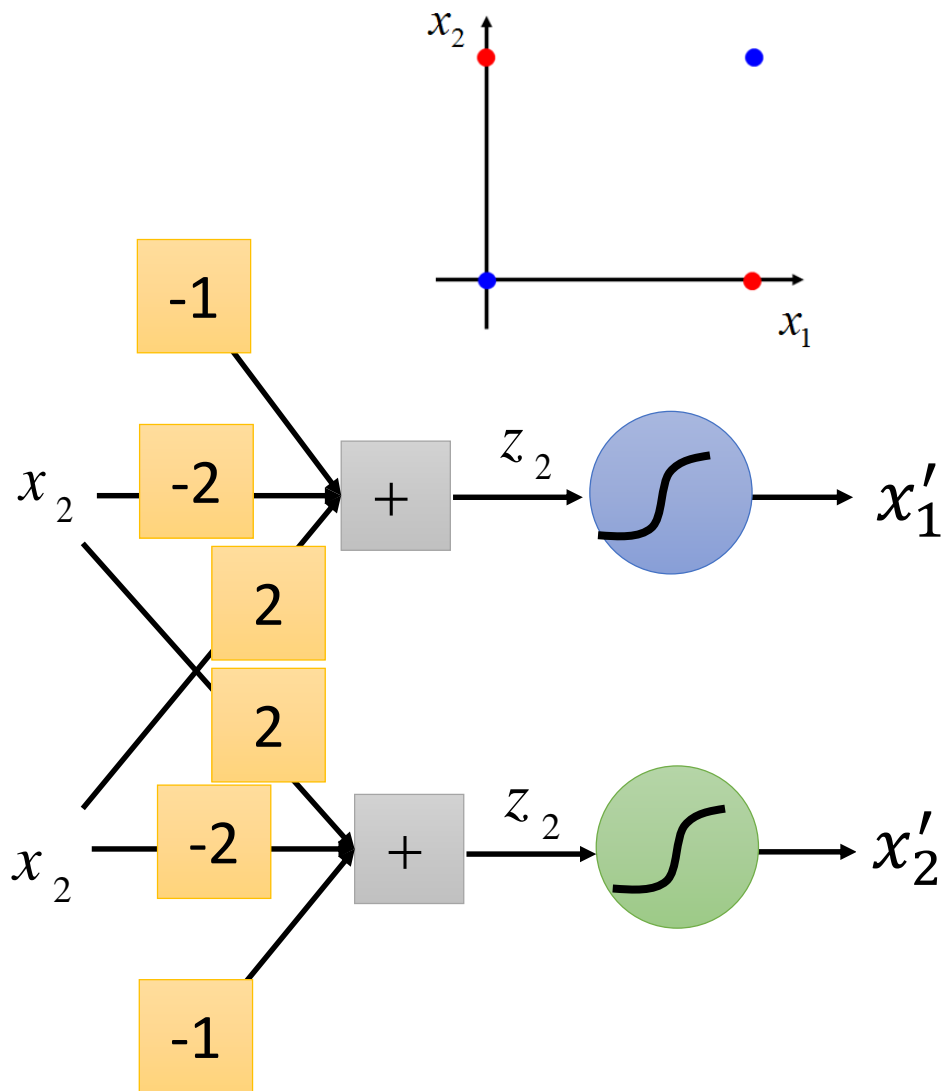


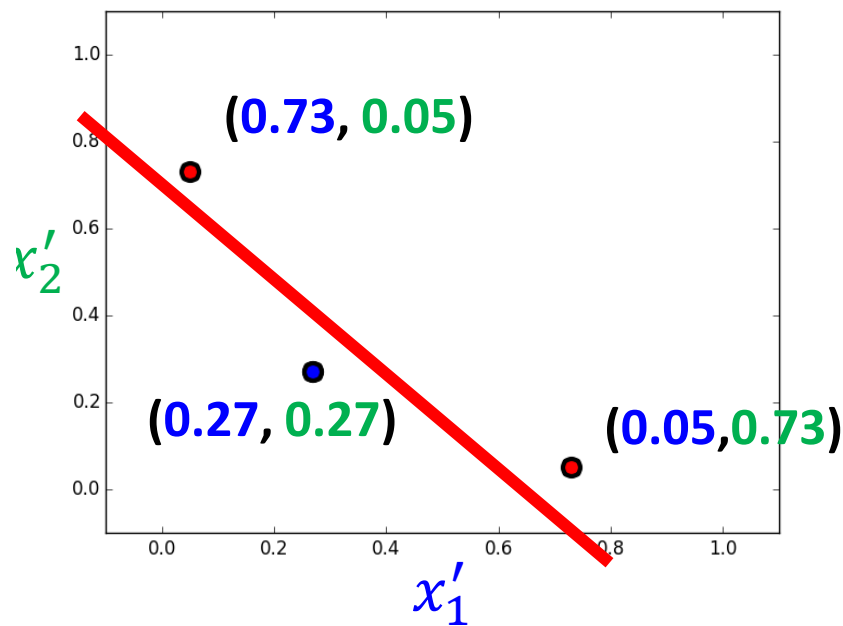
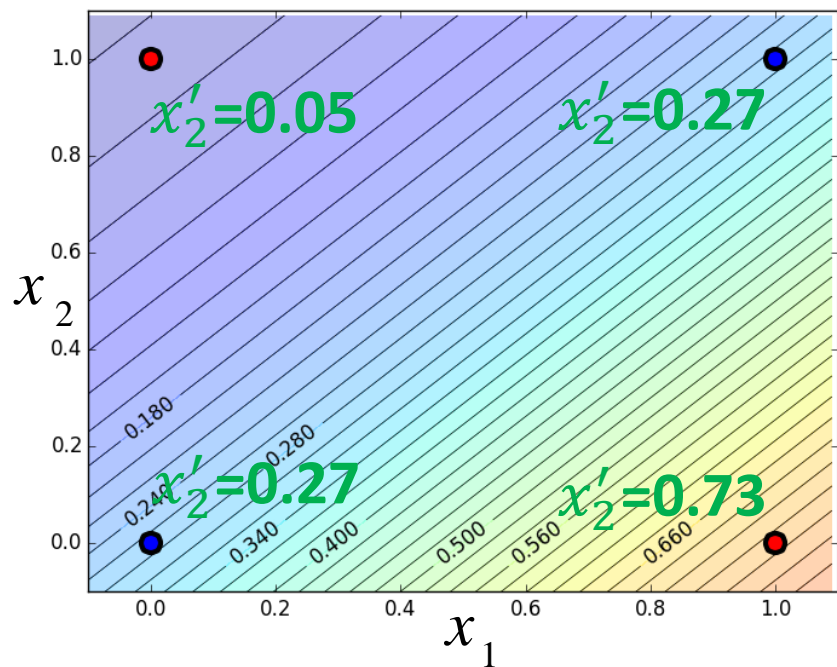
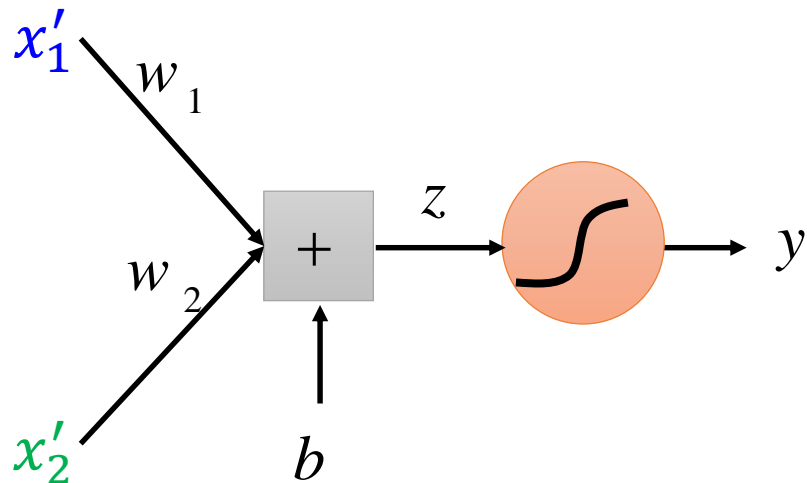
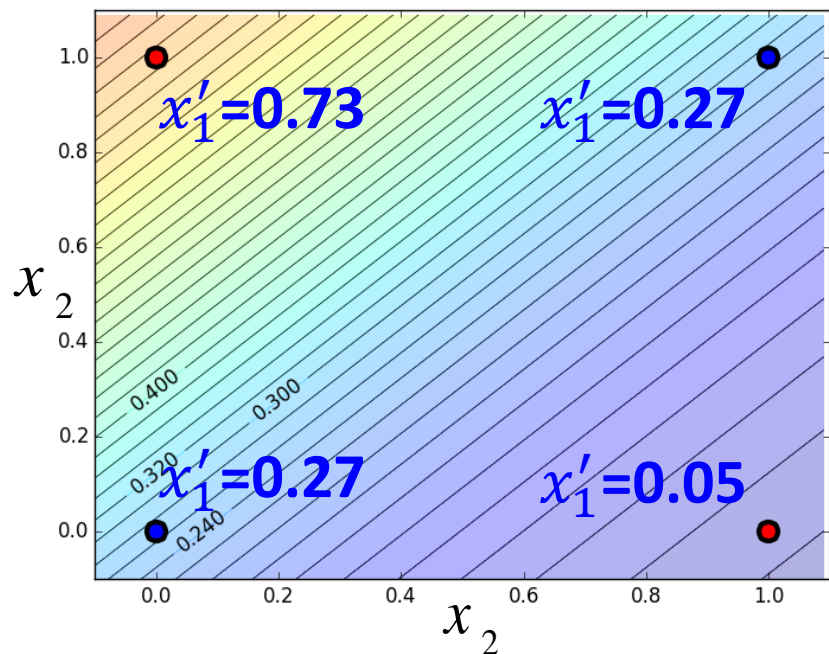
# Limitation of Logistic Regression

- Cascading logistic regression models



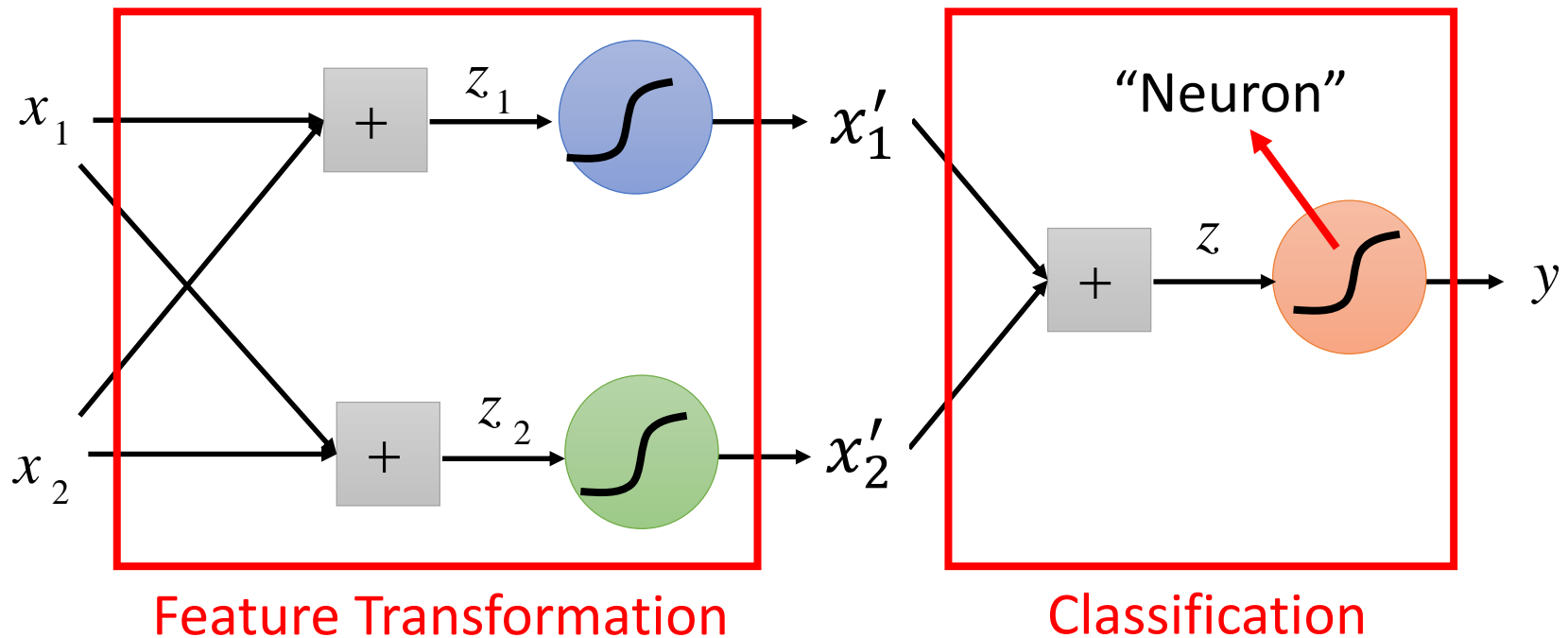
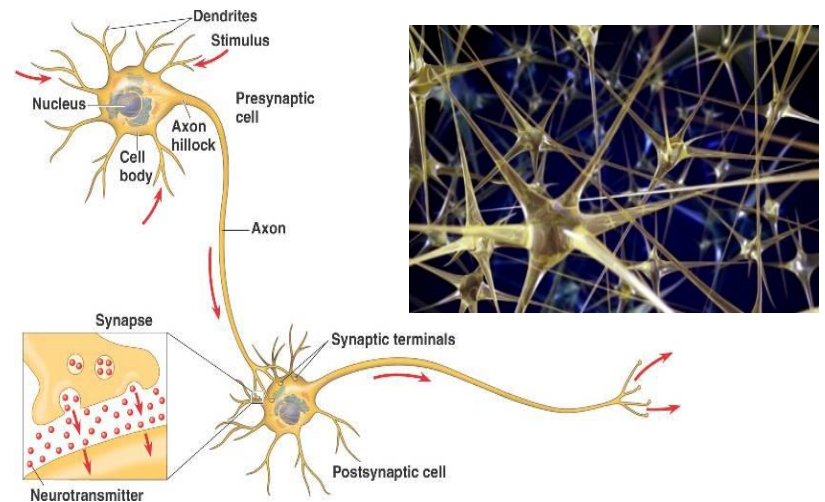
(ignore bias in this figure)





# Deep Learning!

All the parameters of the logistic regressions are jointly learned.



**Neural Network**

# Reference

- Bishop: Chapter 4.3



# Acknowledgement

- 感謝 林恩妤 發現投影片上的錯誤

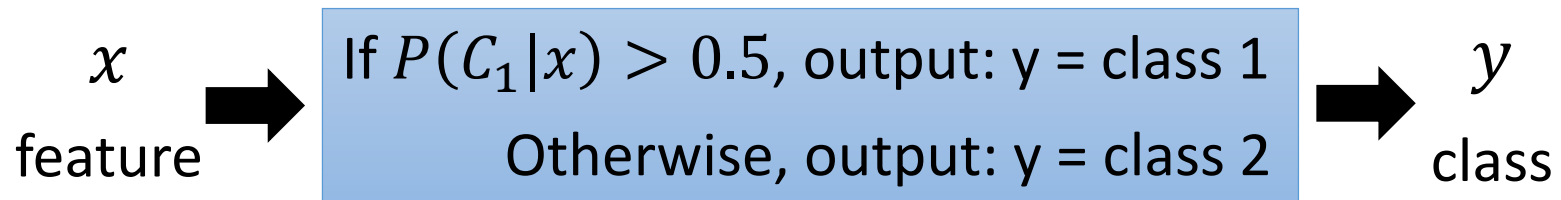
# Appendix

# Three Steps

$x^1$	$x^2$	$x^3$	$\dots \dots$	$x^n$
$\hat{y}^1$	$\hat{y}^2$	$\hat{y}^3$	$\dots \dots$	$\hat{y}^n$

$$\hat{y}^n = \text{class 1, class 2}$$

- Step 1. Function Set (Model)



$$P(C_1|x) = \sigma(w \cdot x + b)$$

$w$  and  $b$  are related to  $N_1, N_2, \mu^1, \mu^2, \Sigma$

- Step 2. Goodness of a function

$$L(f) = \sum_n \delta(f(x^n) \neq \hat{y}^n) \rightarrow L(f) = \sum_n l(f(x^n) \neq \hat{y}^n)$$

- Step 3. Find the best function: gradient descent

## Step 2: Loss function

$$f_{w,b}(x) = \begin{cases} z \geq 0 & +1 \\ z < 0 & -1 \end{cases}$$

Ideal loss:

$$L(f) = \sum_n \delta(f(x^n) \neq \hat{y}^n)$$

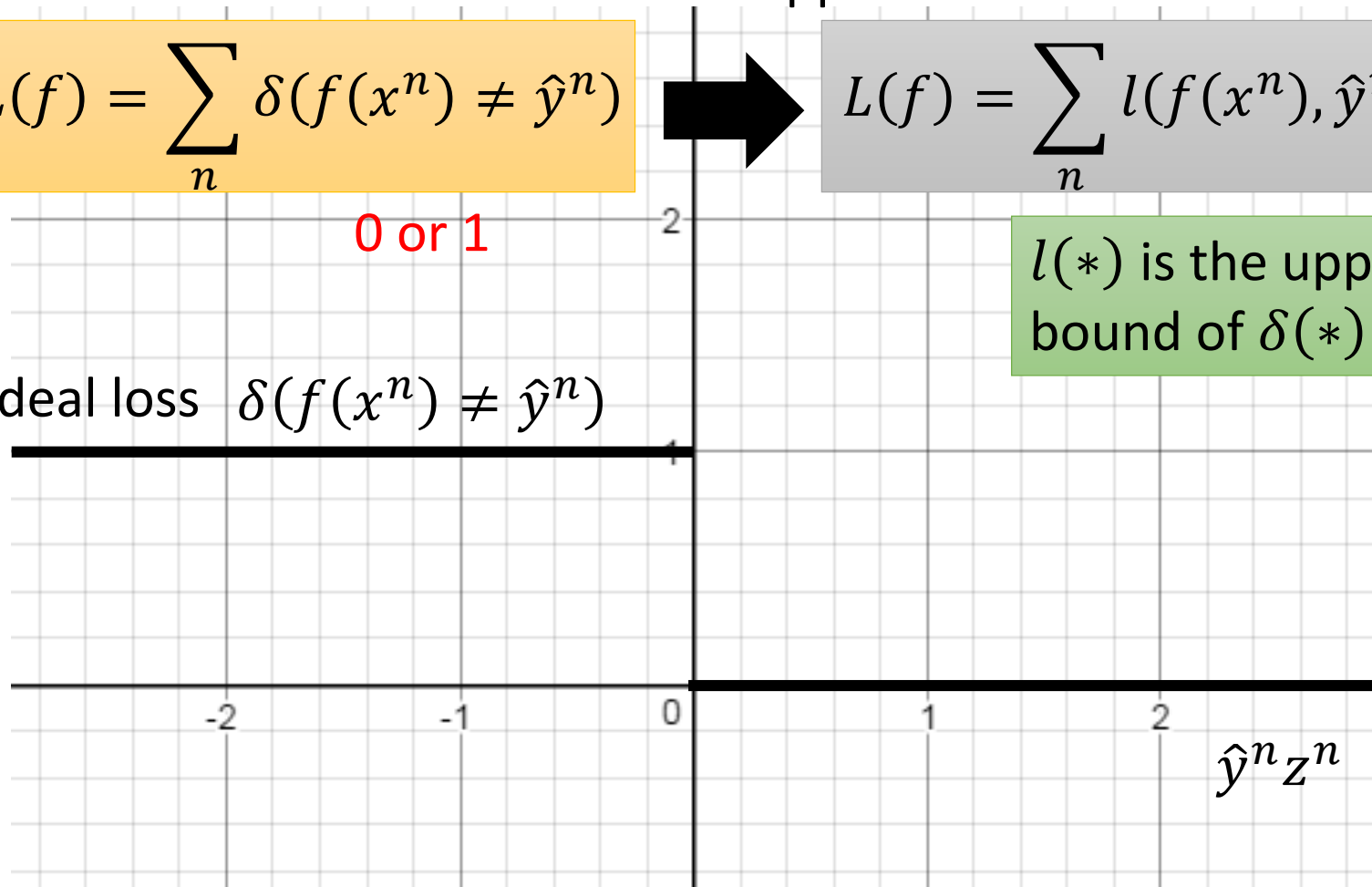
0 or 1

Approximation:

$$L(f) = \sum_n l(f(x^n), \hat{y}^n)$$

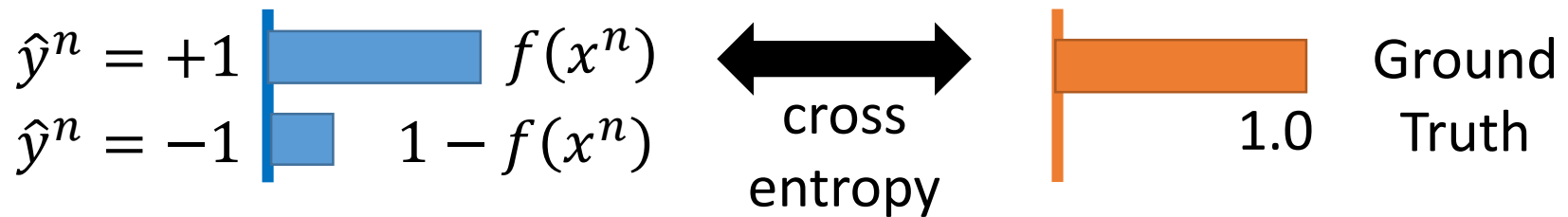
$l(*)$  is the upper bound of  $\delta(*)$

Ideal loss  $\delta(f(x^n) \neq \hat{y}^n)$



## Step 2: Loss function

$l(f(x^n), \hat{y}^n)$ : cross entropy



If  $\hat{y}^n = +1$ :

$$\begin{aligned} l(f(x^n), \hat{y}^n) &= -\ln f(x^n) = -\ln \sigma(z^n) = -\ln \frac{1}{1 + \exp(-z^n)} \\ &= \ln(1 + \exp(-z^n)) = \underline{\ln(1 + \exp(-\hat{y}^n z^n))} \end{aligned}$$

If  $\hat{y}^n = -1$ :

$$\begin{aligned} l(f(x^n), \hat{y}^n) &= -\ln(1 - f(x^n)) \\ &= -\ln(1 - \sigma(x^n)) = -\ln \frac{\exp(-z^n)}{1 + \exp(-z^n)} = -\ln \frac{1}{1 + \exp(z^n)} \\ &= \ln(1 + \exp(z^n)) = \underline{\ln(1 + \exp(-\hat{y}^n z^n))} \end{aligned}$$

## Step 2: Loss function

$l(f(x^n), \hat{y}^n)$ : cross entropy

$$l(f(x^n), \hat{y}^n) = \ln(1 + \exp(-\hat{y}^n z^n))$$

