

# Logistic Regression

# Step 1: Function Set

We want to find  $P_{w,b}(C_1|x)$

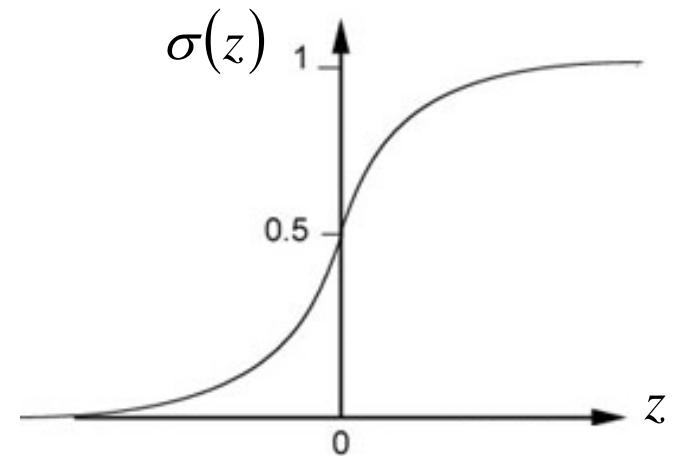
If  $P_{w,b}(C_1|x) \geq 0.5$ , output  $C_1$

Otherwise, output  $C_2$

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

$$z = w \cdot x + b$$

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

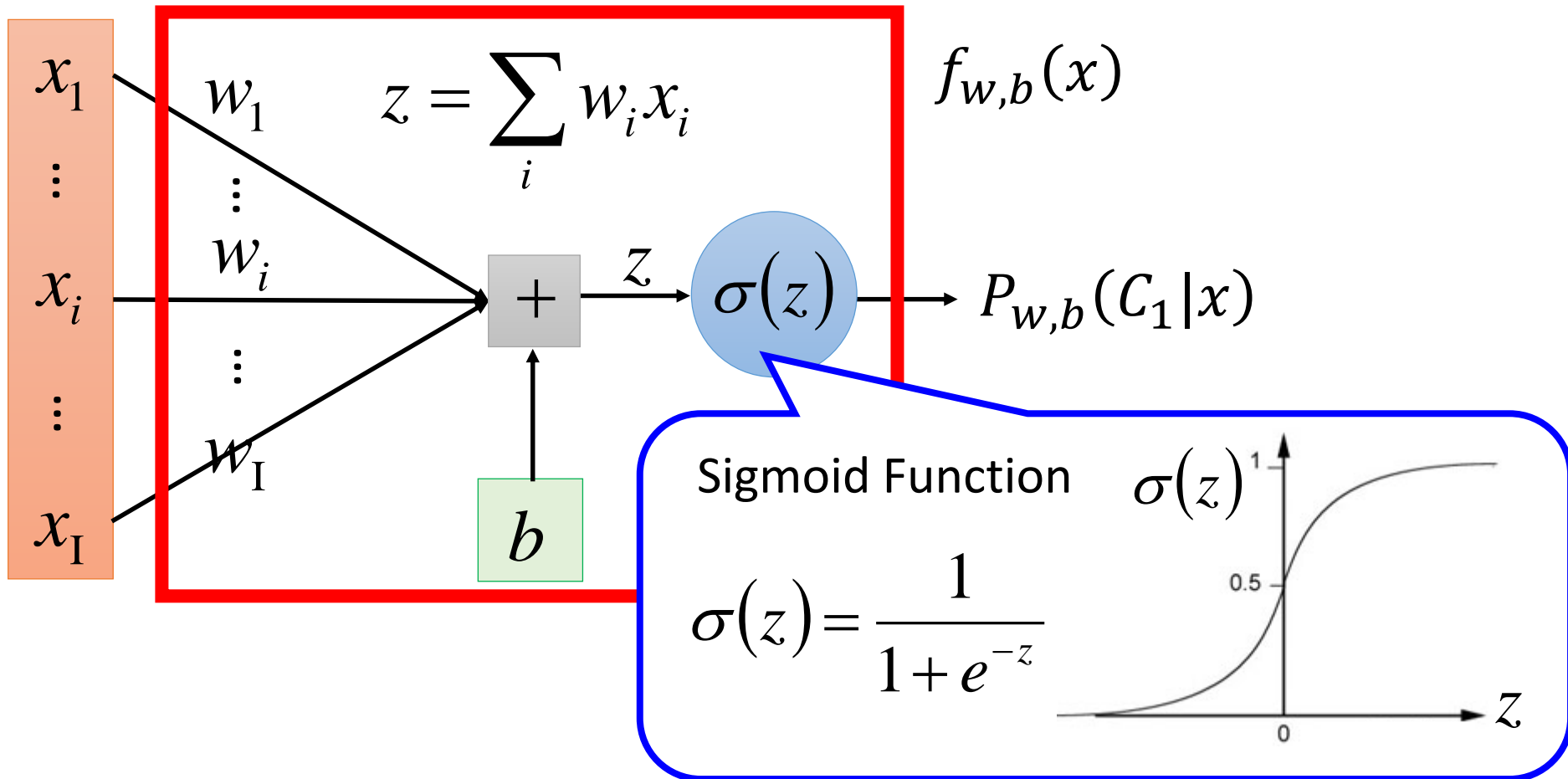


Function set:

$$f_{w,b}(x) = P_{w,b}(C_1|x)$$

Including all  
different  $w$  and  $b$

# Step 1: Function Set



## **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:

## Step 2: Goodness of a Function

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

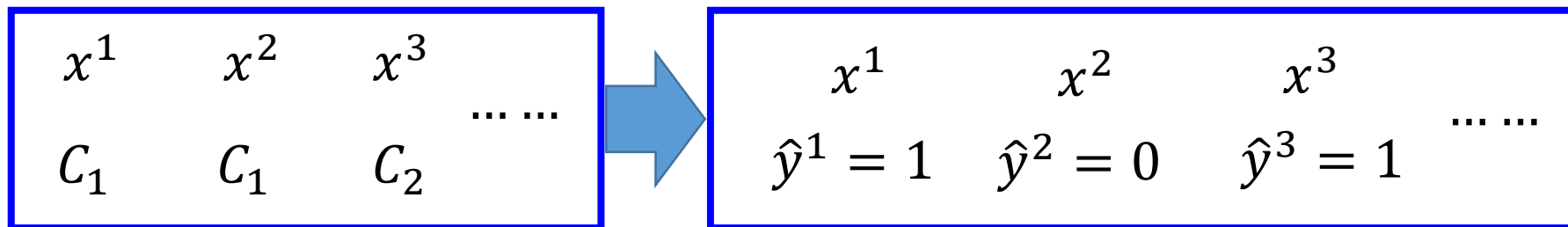
Assume the data is generated based on  $f_{w,b}(x) = P_{w,b}(C_1|x)$

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

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

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

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

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

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

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

⋮

## Step 2: Goodness of a Function

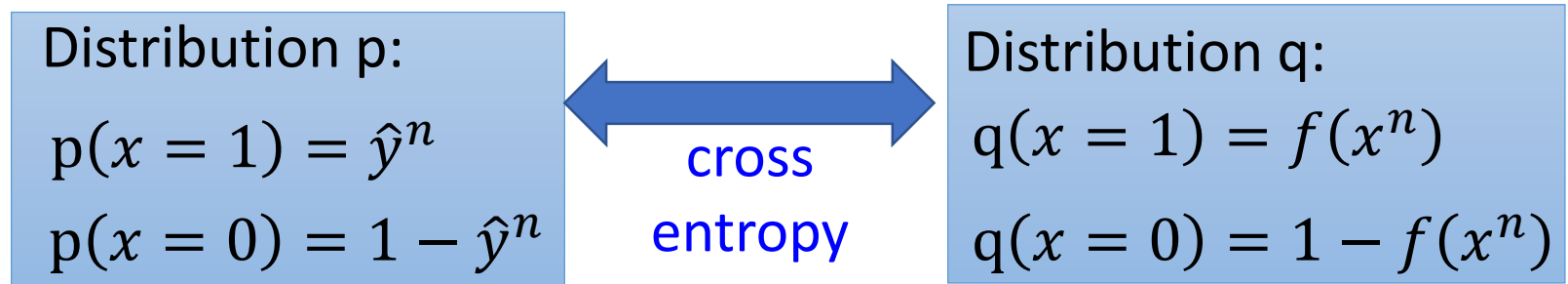
$$L(w, b) = f_{w,b}(x^1)f_{w,b}(x^2)(1 - f_{w,b}(x^3)) \cdots f_{w,b}(x^N)$$

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

$\hat{y}^n$ : 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



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

## 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

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

## 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:

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

Question: Why don't we simply use square error as linear regression?

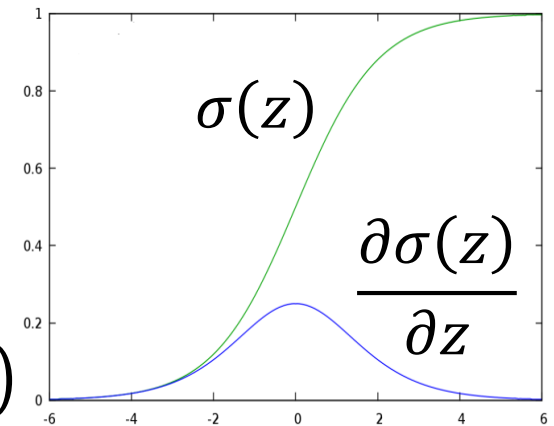


# Step 3: Find the best function

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

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

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



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

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

## 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

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

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

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

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

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

## 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

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

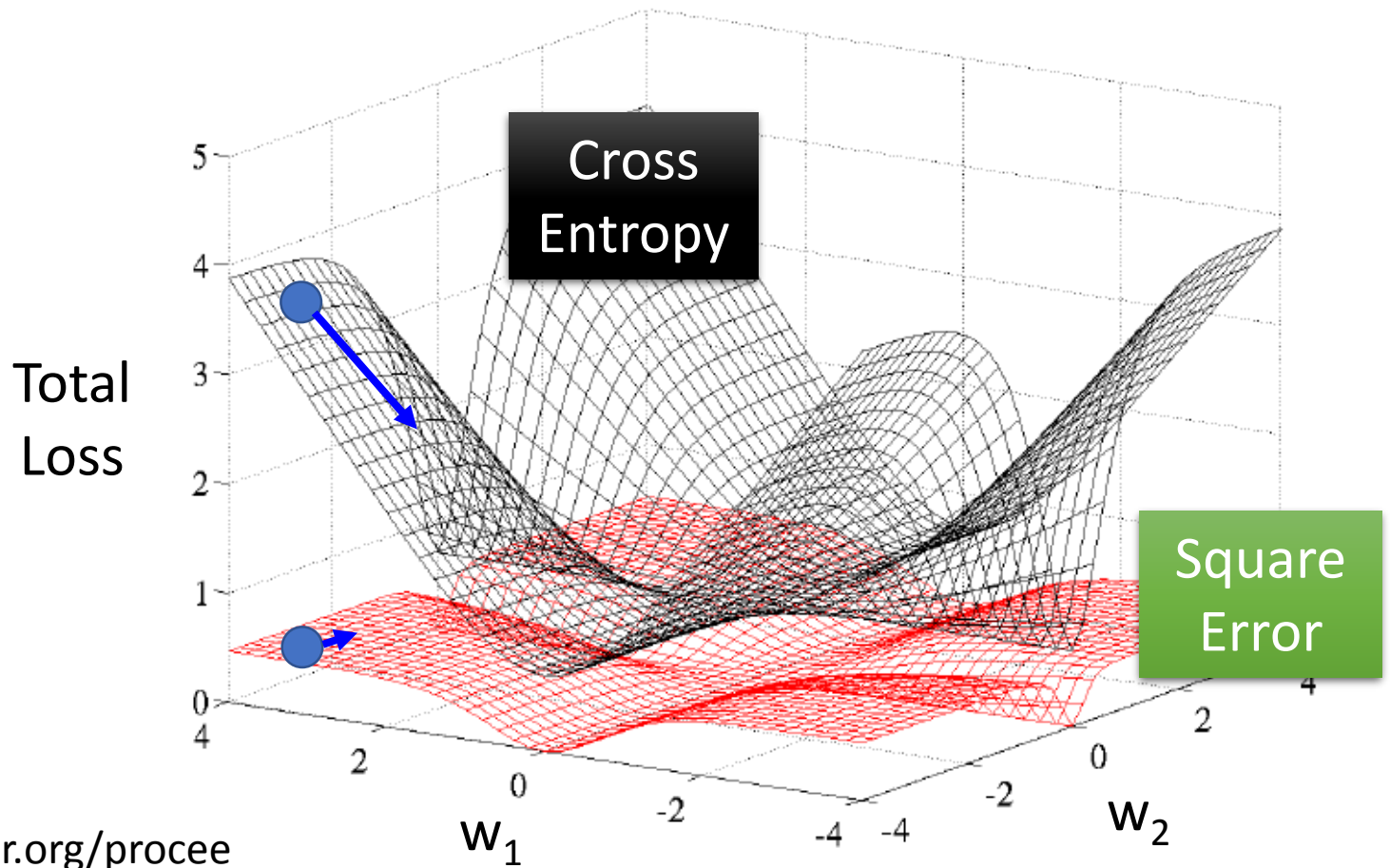
Step 3:

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

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

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

# Cross Entropy v.s. Square Error



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

# Discriminative v.s. Generative

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



directly find  $w$  and  $b$

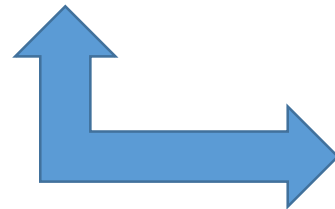


Find  $\mu^1, \mu^2, \Sigma^{-1}$

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

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

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



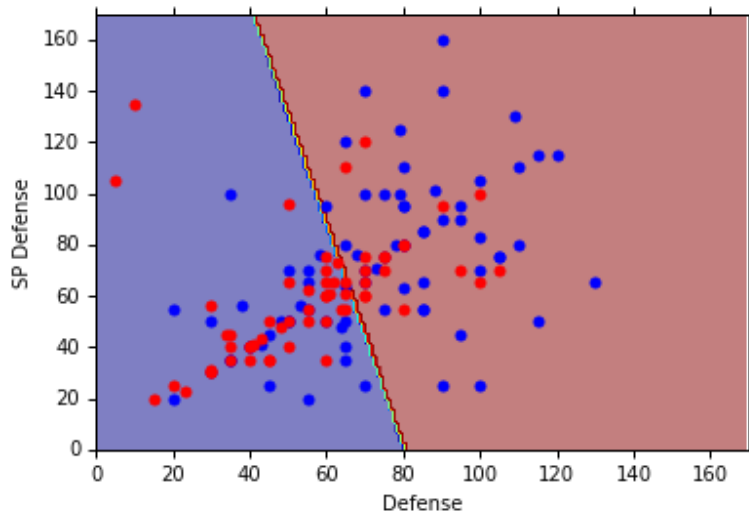
Will we obtain the same set of  $w$  and  $b$ ?

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



# Generative v.s. Discriminative

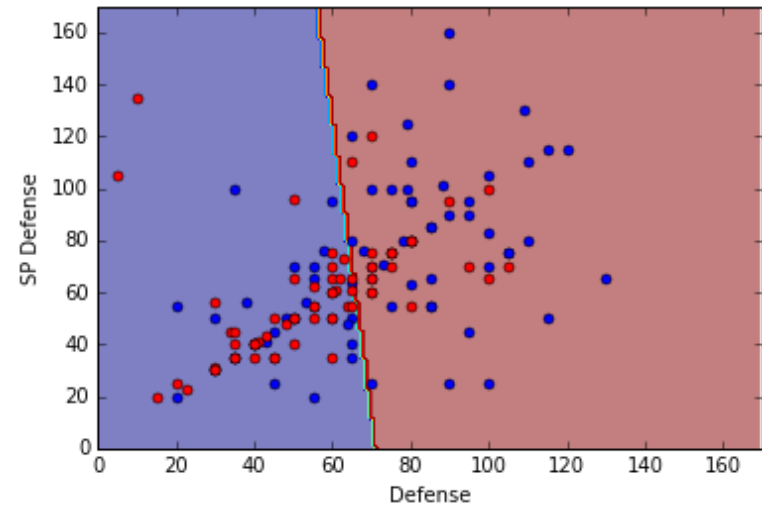
**Generative**



All: total, hp, att, sp att, de, sp de, speed

73% accuracy

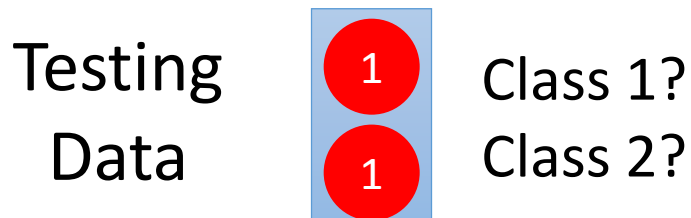
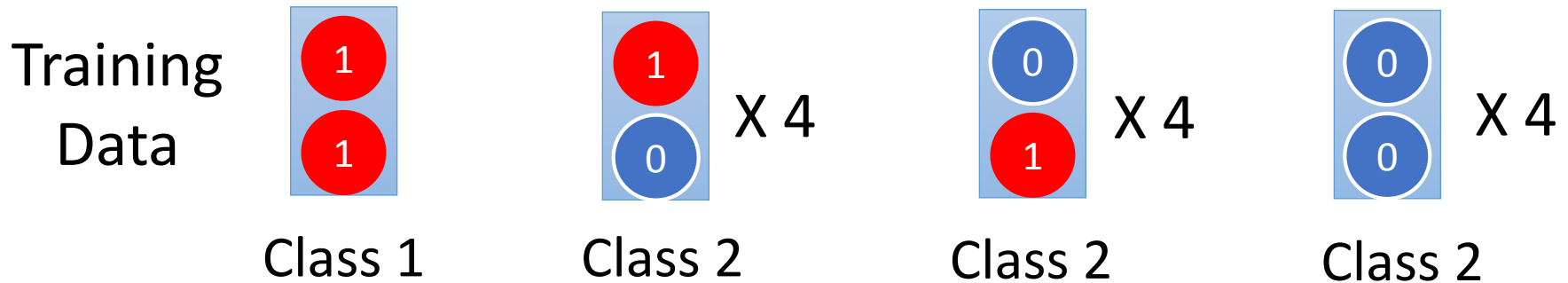
**Discriminative**



79% accuracy

# Generative v.s. Discriminative

- Example

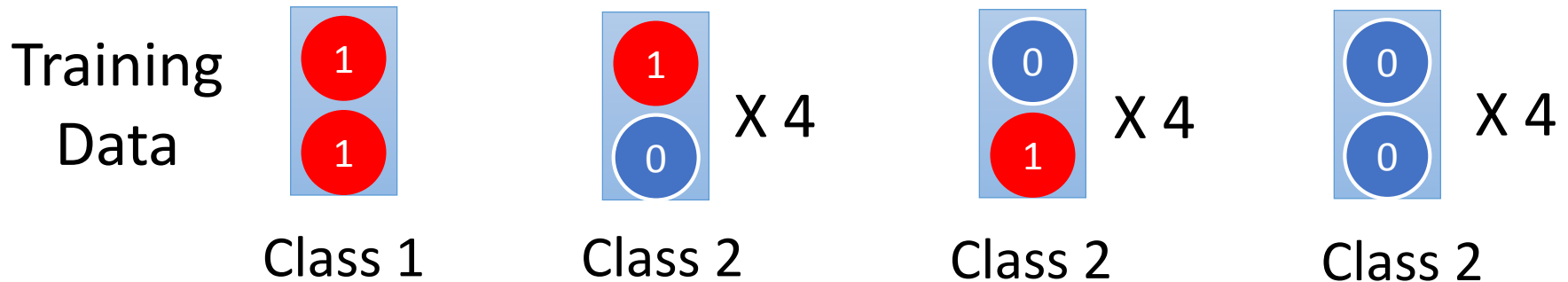


How about Naïve Bayes?

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

# 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

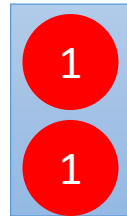
X 4



Class 2

X 4

Testing  
Data



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

<0.5

Diagram illustrating the calculation of  $P(C_1|x)$  for testing data (x<sub>1</sub>=1, x<sub>2</sub>=1):

- $P(x|C_1)P(C_1) = 1 \times 1$  (Top path)
- $P(x|C_2)P(C_2) = \frac{1}{3} \times \frac{1}{3}$  (Bottom path)
- $P(C_1) = \frac{1}{13}$  (Top path)
- $P(C_2) = \frac{12}{13}$  (Bottom path)

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

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

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

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

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

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

# Generative v.s. Discriminative

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

# Multi-class Classification (3 classes as example)

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

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

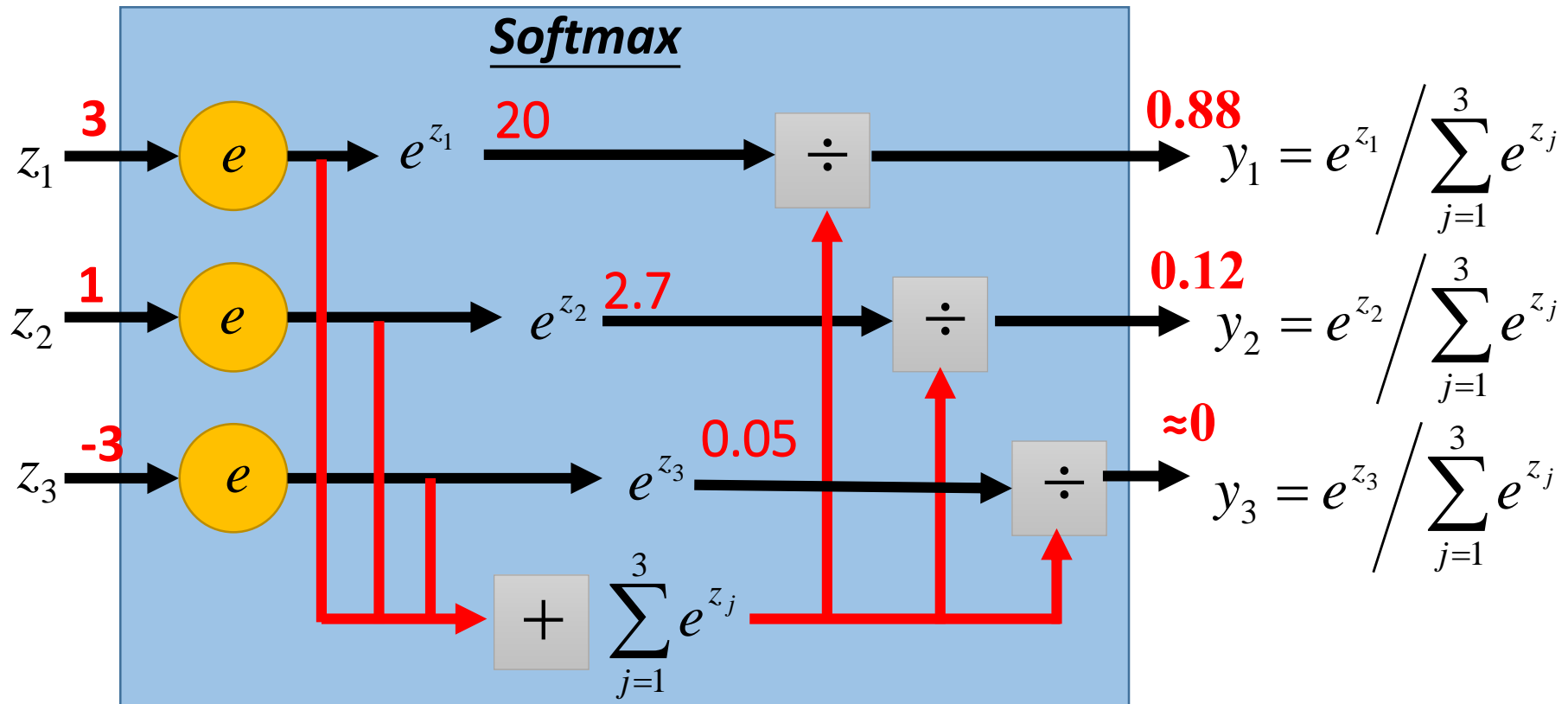
$$C_3: w^3, b_3 \quad z_3 = w^3 \cdot x + b_3$$

## Probability:

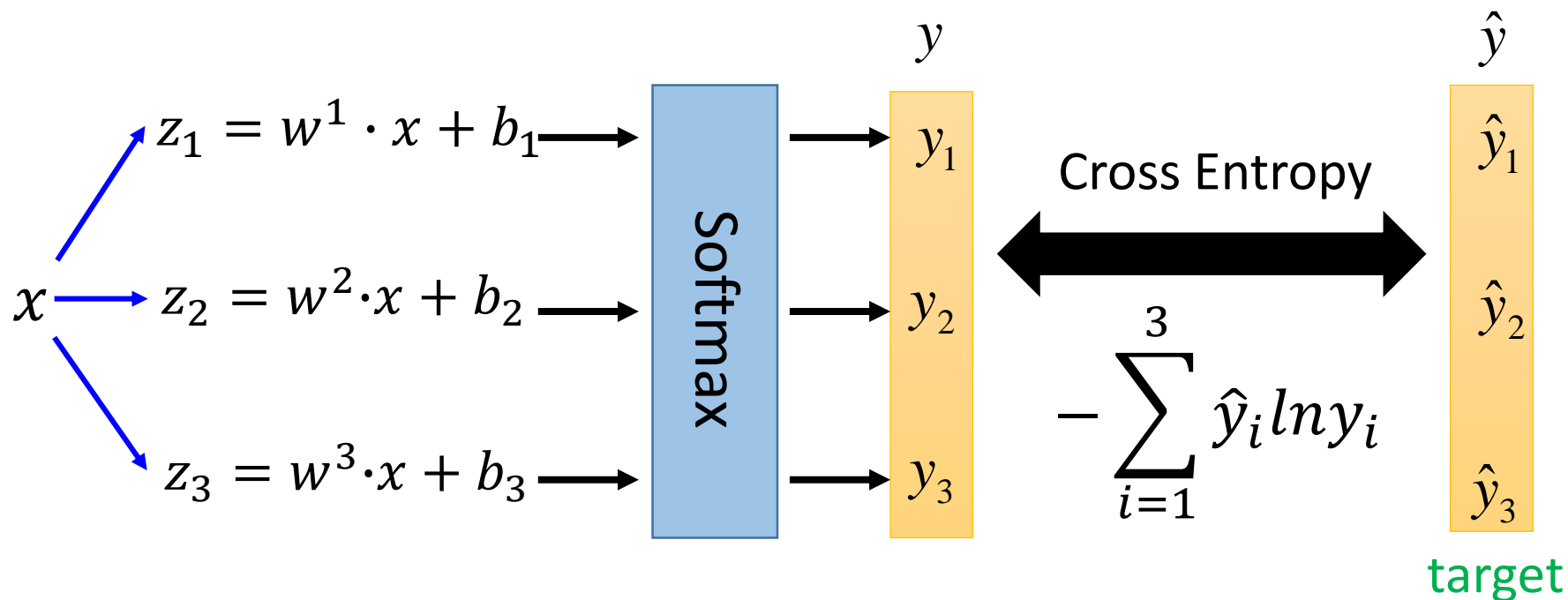
$$\blacksquare 1 > y_i > 0$$

$$\blacksquare \sum_i y_i = 1$$

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



# Multi-class Classification (3 classes as example)



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

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

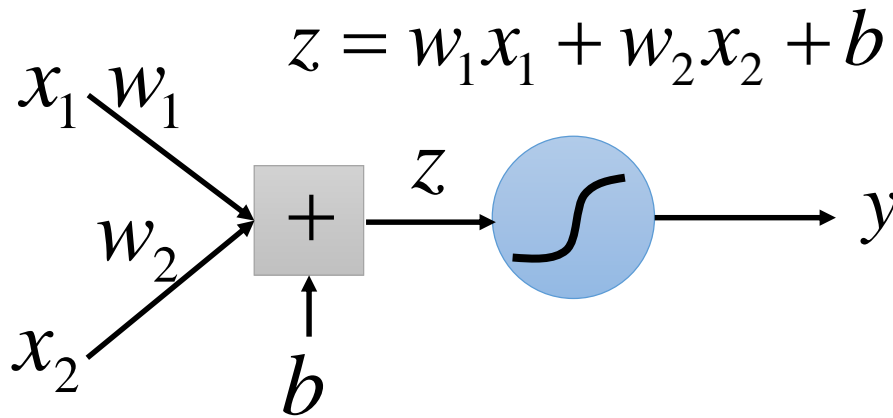
If  $x \in \text{class 2}$

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

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

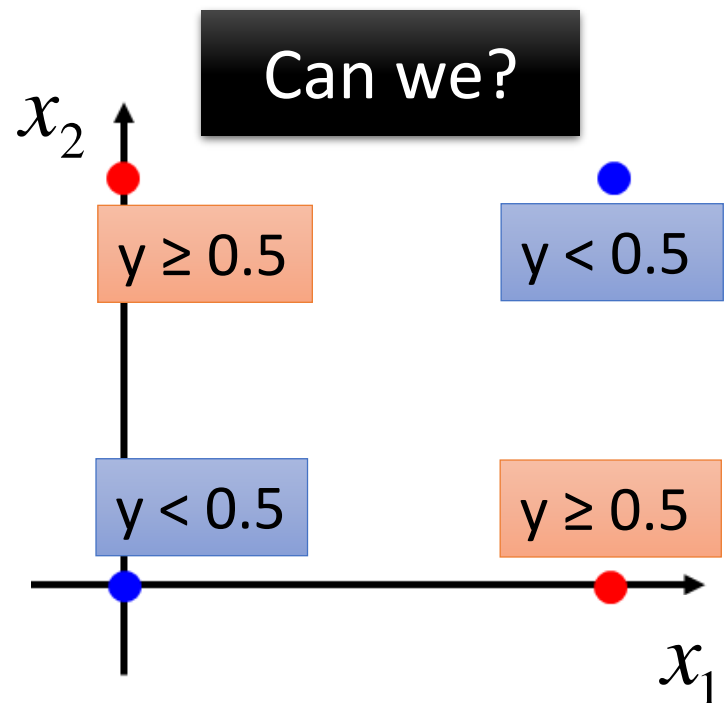
$$\hat{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

# Limitation of Logistic Regression



$$\begin{cases} \text{Class1} & y \geq 0.5 \\ \text{Class2} & y < 0.5 \end{cases}$$

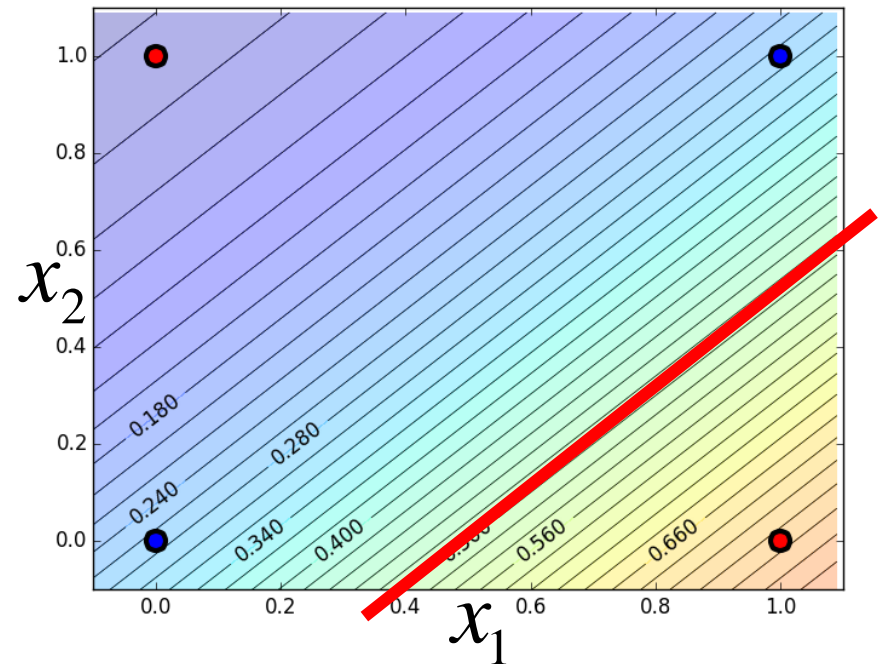
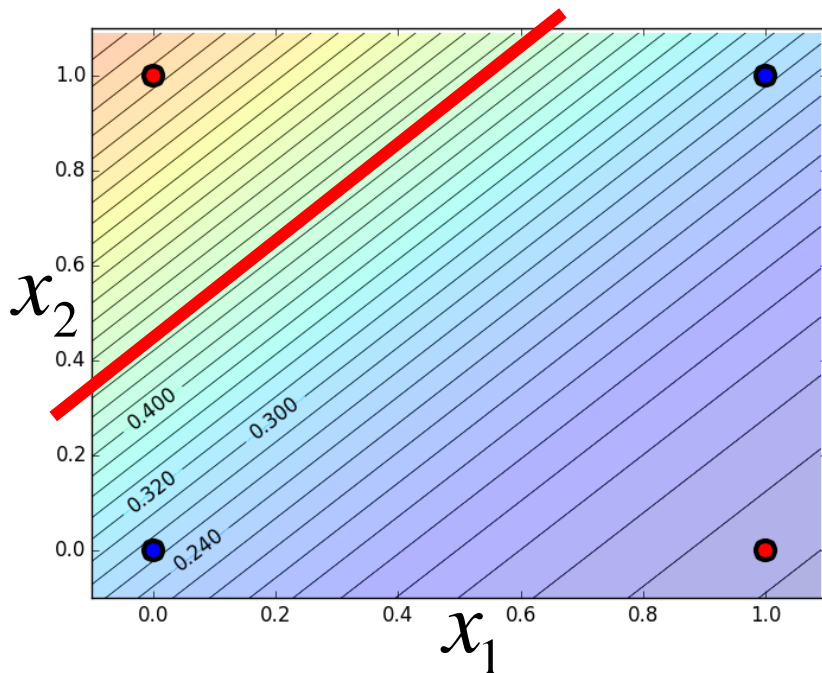
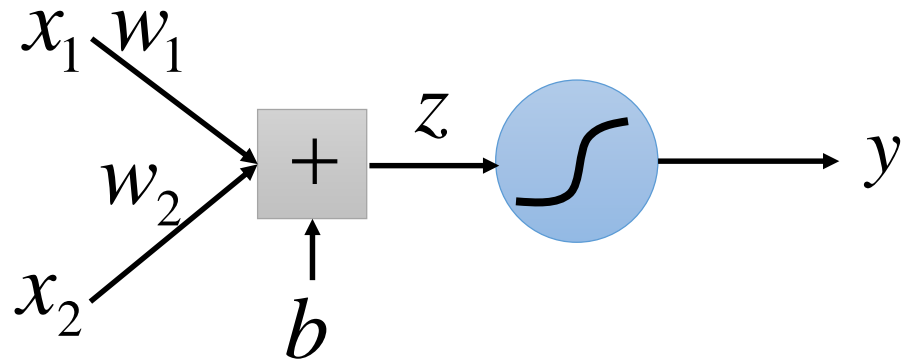
Input Feature		Label
$x_1$	$x_2$	
0	0	Class 2
0	1	Class 1
1	0	Class 1
1	1	Class 2





# Limitation of Logistic Regression

- No, we can't .....

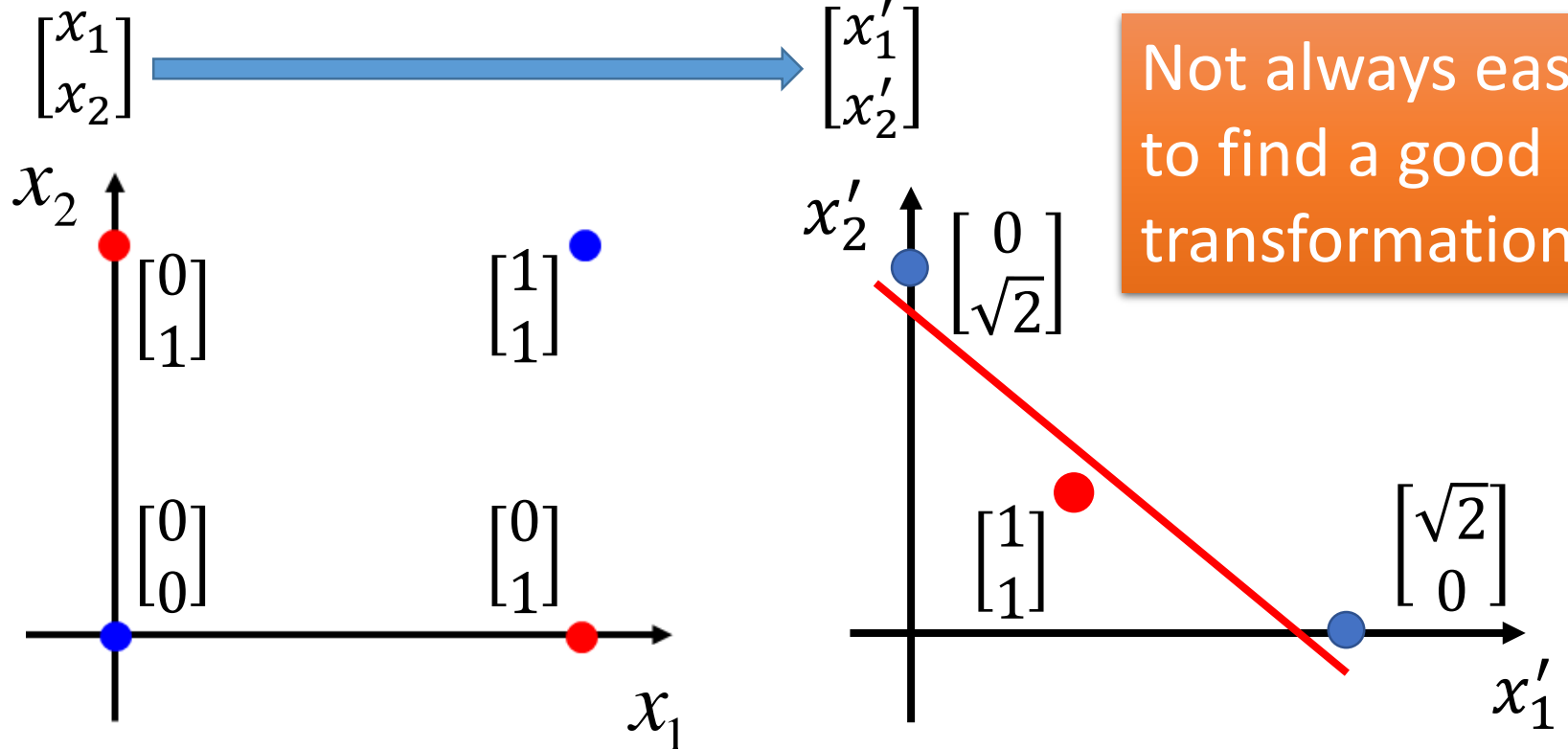


# Limitation of Logistic Regression

- Feature Transformation

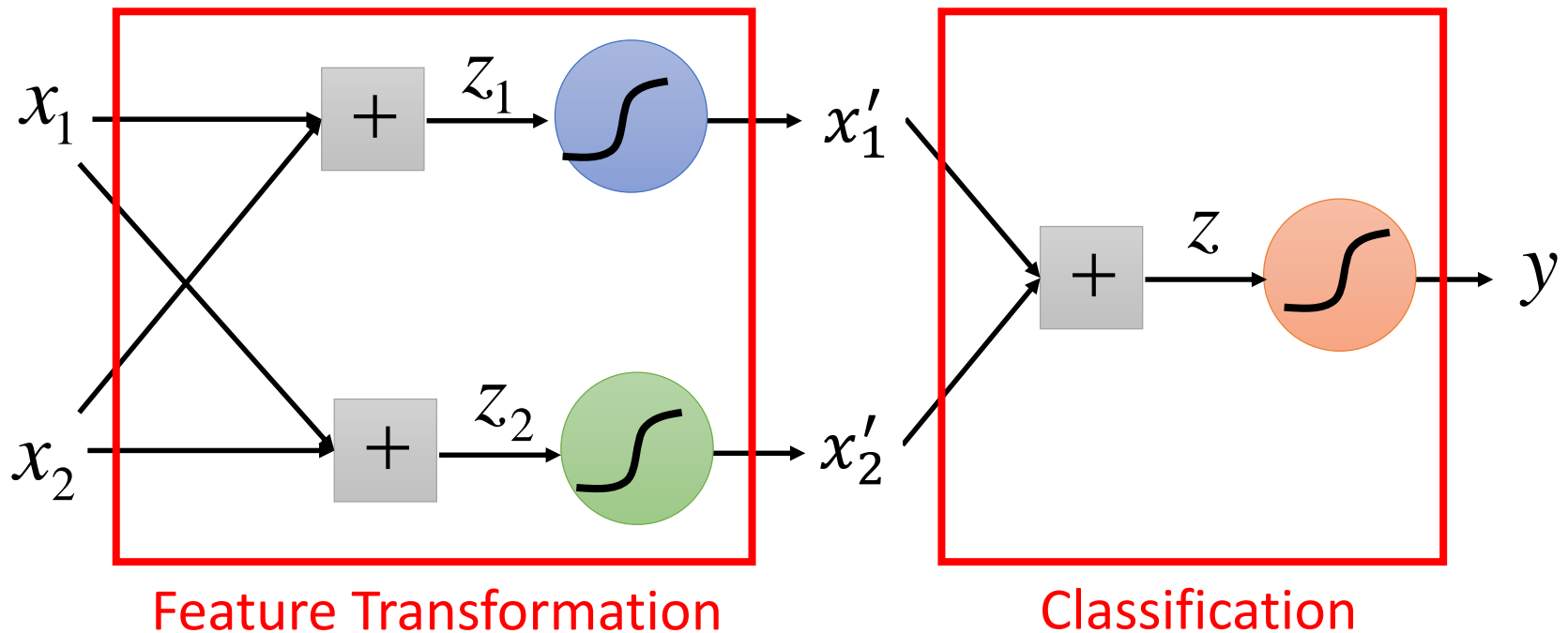
$x'_1$ : distance to  $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

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

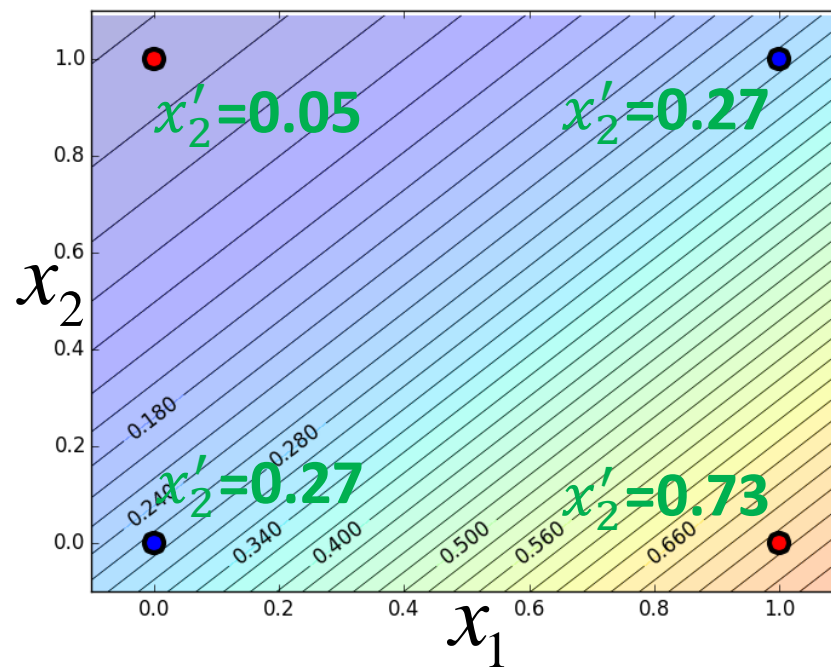
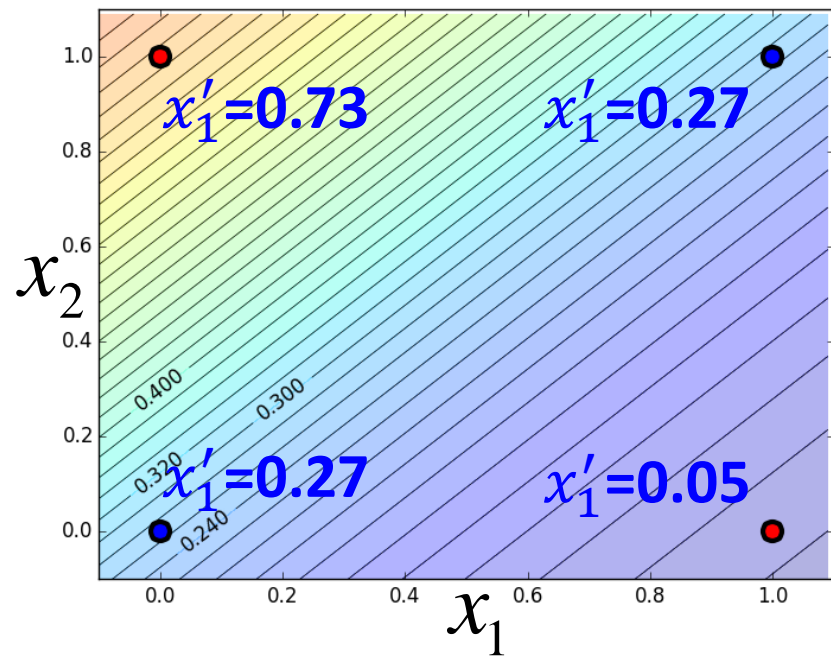
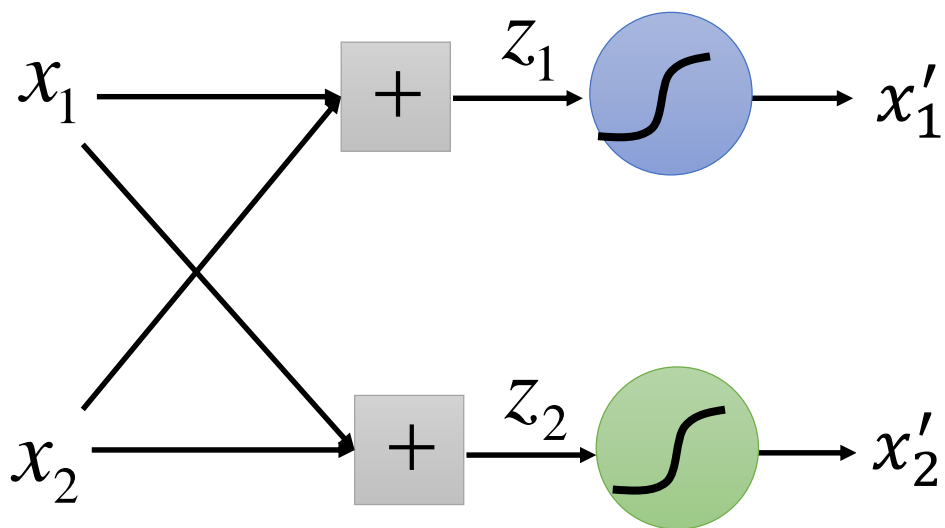


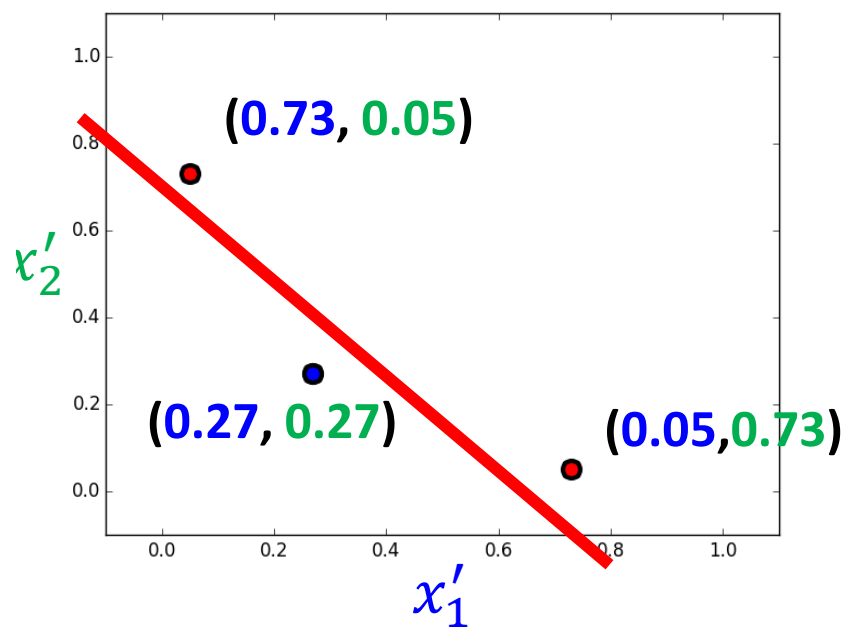
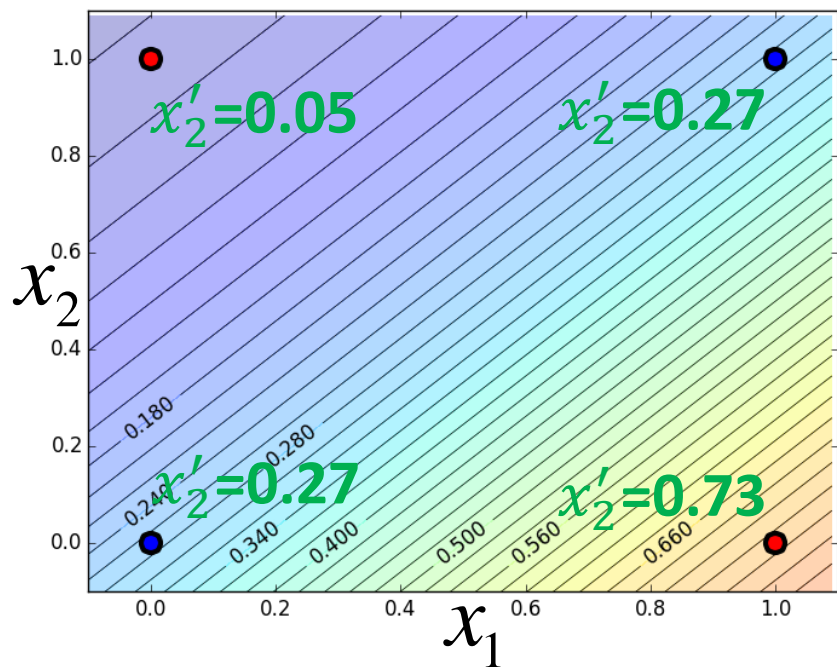
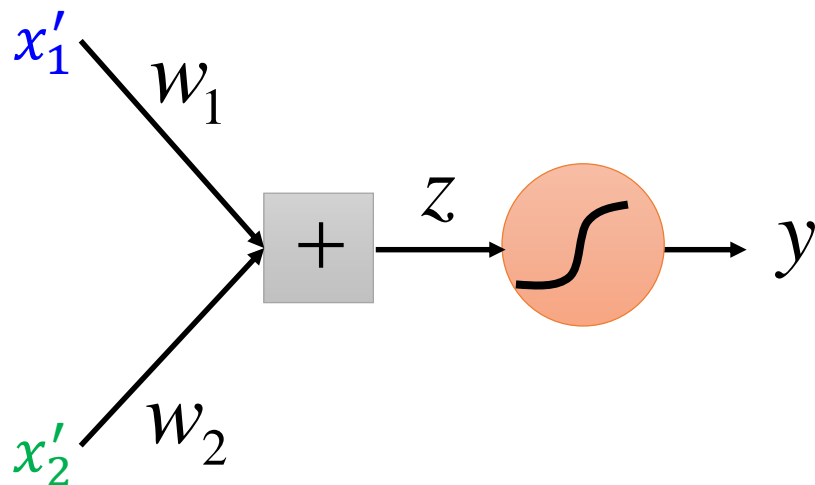
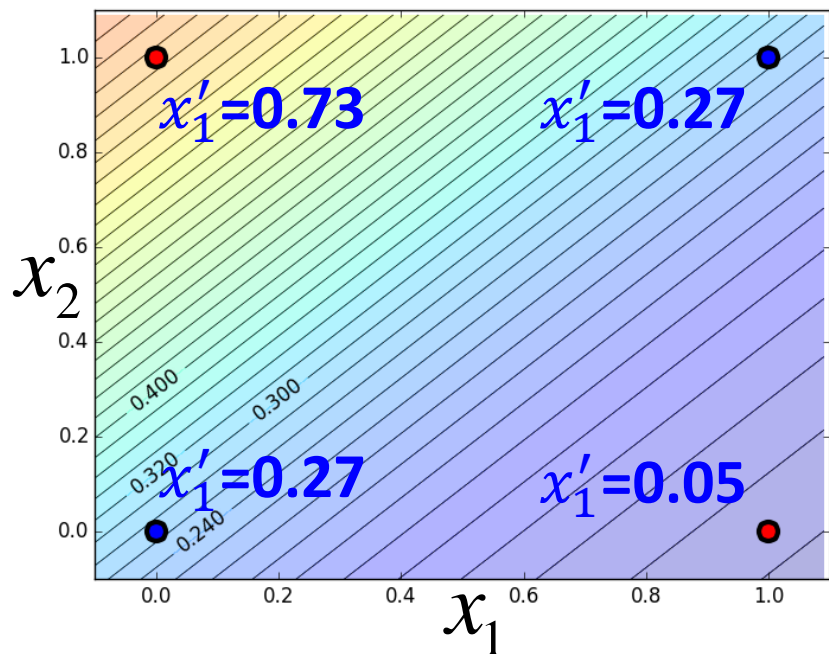
# Limitation of Logistic Regression

- Cascading logistic regression models

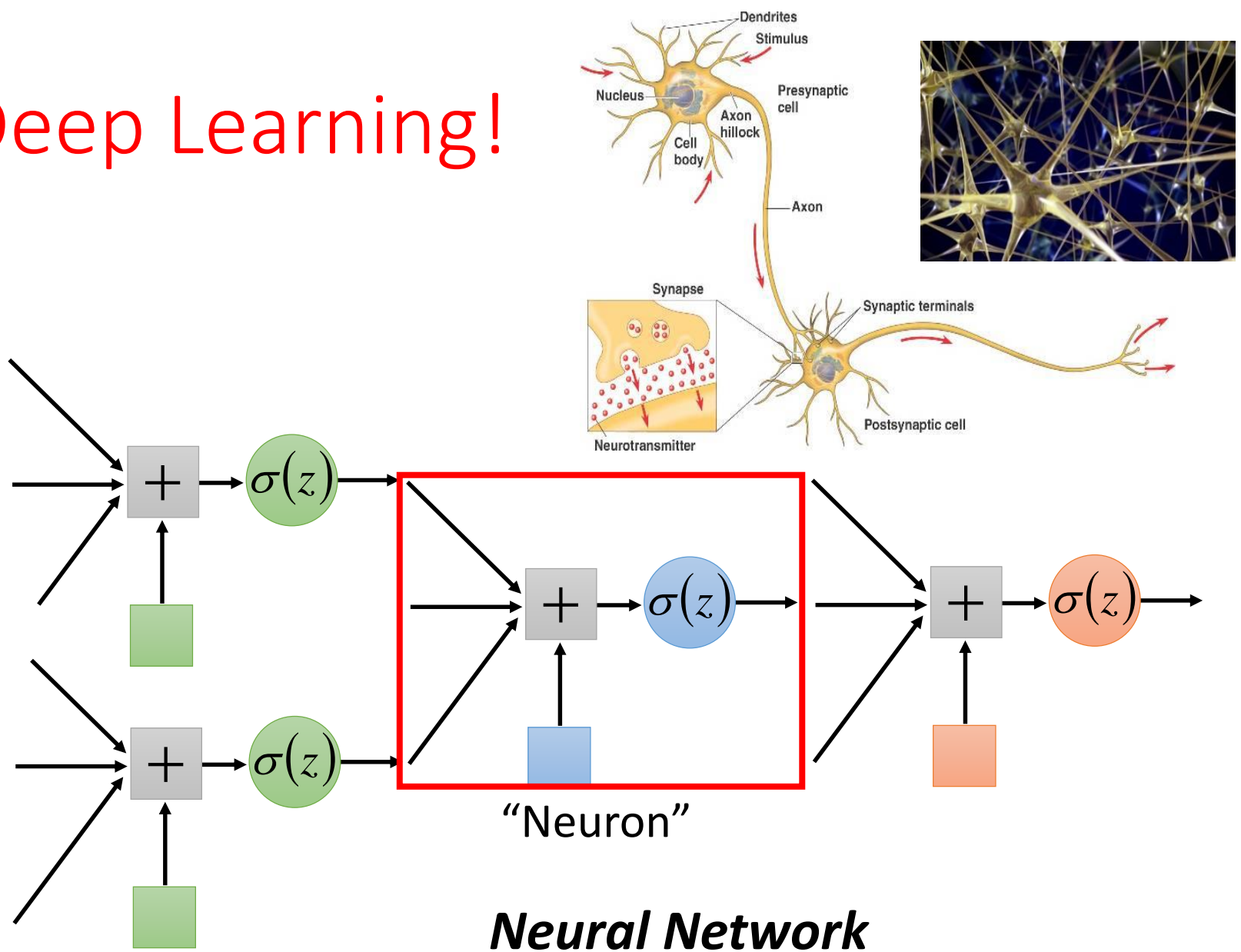


(ignore bias in this figure)





# Deep Learning!



# Reference

- Bishop: Chapter 4.3