Logistic Regression

解決 Classification 的另外一個方法

Step 1: Function Set

在 Generative Probabilistic Model 中已經知道解決 Classification
問題就是要計算 Posterior Probability

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

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

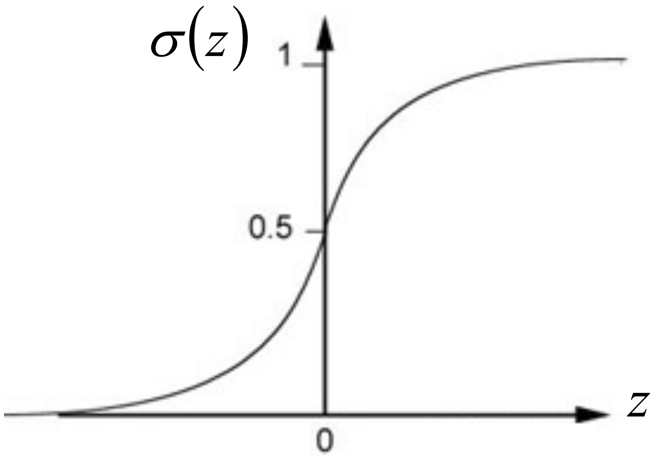
Otherwise, output C_2

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

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

經過數學推導，Posterior Probability 相當於 Sigmoid Function of Z

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



Function set: 機器學習第一步：定義一個 Function Set
在 Generative Probabilistic Model 中，是由 Prior 與 Likelihood 去計算 Posterior
在 Losgistic Regression 中，是由 z 的 w 與 b 計算 Posterior

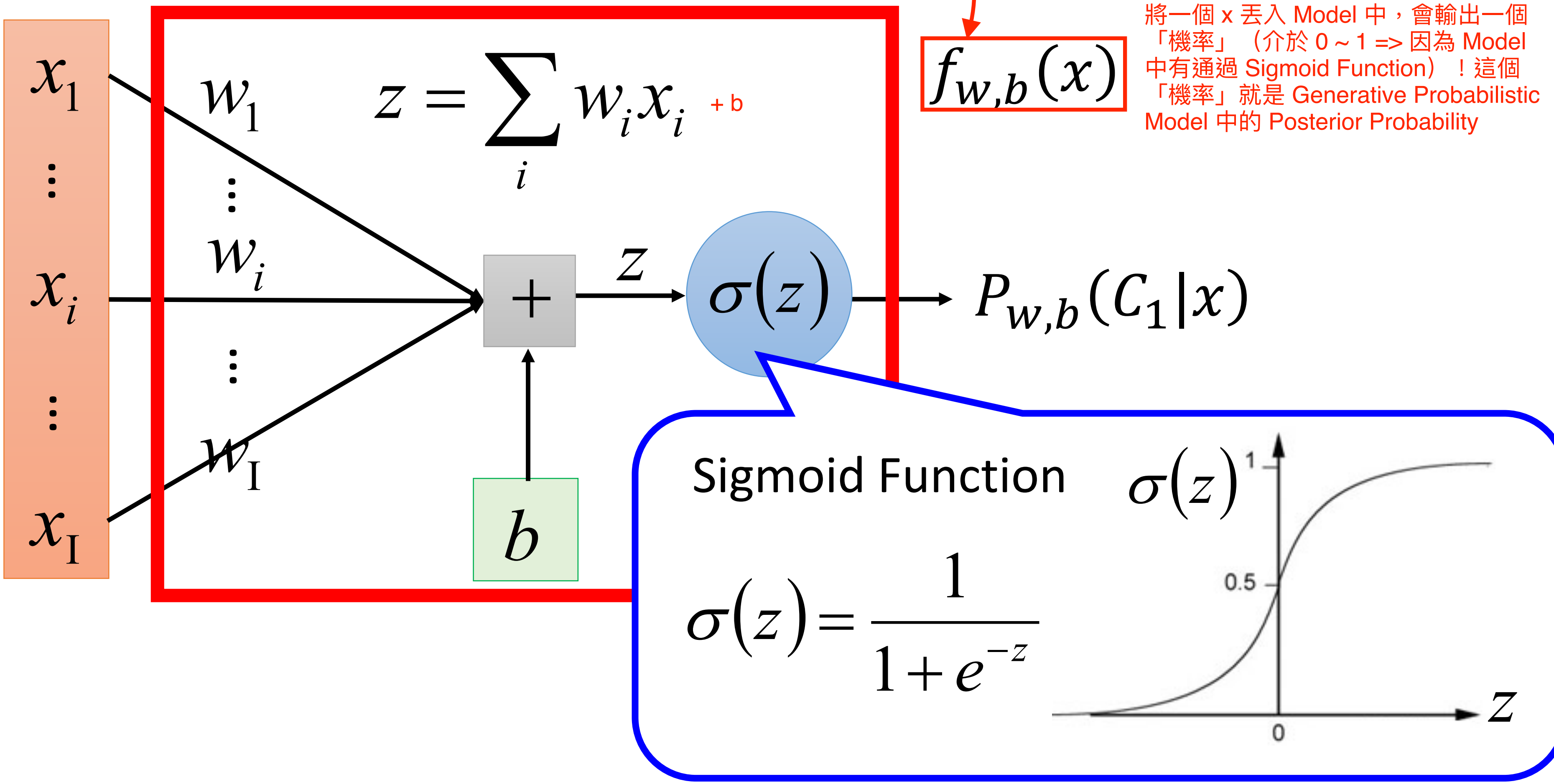
$$f_{w,b}(x) = P_{w,b}(C_1|x)$$

Including all different w and b

Step 1: Function Set

Function Set (Model) 的樣子

將一個 x 丟入 Model 中，會輸出一個「機率」（介於 0 ~ 1 => 因為 Model 中有通過 Sigmoid Function）！這個「機率」就是 Generative Probabilistic Model 中的 Posterior Probability



	Logistic Regression	***Linear Regression***
Step 1:	$f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$ Output: between 0 and 1	$f_{w,b}(x) = \sum_i w_i x_i + b$ 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?

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

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

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

$$\begin{array}{cccc} x^1 & x^2 & x^3 & \\ C_1 & C_1 & C_2 & \cdots\cdots \end{array} \Rightarrow \begin{array}{cccc} x^1 & x^2 & x^3 & \\ \hat{y}^1 = 1 & \hat{y}^2 = 0 & \hat{y}^3 = 1 & \cdots\cdots \\ & 1 & 0 & \end{array}$$

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

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

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

$$\begin{aligned} &-lnL(w,b) \\ &= -lnf_{w,b}(x^1) \\ &\quad -lnf_{w,b}(x^2) \\ &\quad -ln\left(1 - f_{w,b}(x^3)\right) \\ &\quad \vdots \end{aligned}$$

y hat   1 - y hat

$$-\left[\boxed{1}\, lnf(x^1) \cancel{+ \boxed{0}\, ln\left(1 - f(x^1)\right)}\right]$$

$$-\left[\boxed{1}\, lnf(x^2) \cancel{+ \boxed{0}\, ln\left(1 - f(x^2)\right)}\right]$$

$$-\left[\cancel{\boxed{0}\, lnf(x^3) +} \boxed{1}\, ln\left(1 - f(x^3)\right)\right]$$

Step 2: Goodness of a Function

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

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

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

$$= \sum_n \underbrace{-\left[\hat{y}^n lnf_{w,b}(x^n) + (1-\hat{y}^n) ln\left(1 - f_{w,b}(x^n)\right)\right]}_{\text{Cross entropy between two Bernoulli distribution}}$$

Distribution p:

$p(x=1) = \hat{y}^n$

$p(x=0) = 1 - \hat{y}^n$

cross entropy

Distribution q:

$q(x=1) = f(x^n)$

$q(x=0) = 1 - f(x^n)$

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

	Logistic Regression	Linear Regression
Step 1:	$f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$ Output: between 0 and 1	$f_{w,b}(x) = \sum_i w_i x_i + b$ Output: any value
Step 2:	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: 1 for class 1, 0 for class 2 $L(f) = \sum_n C(f(x^n), \hat{y}^n)$	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:

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

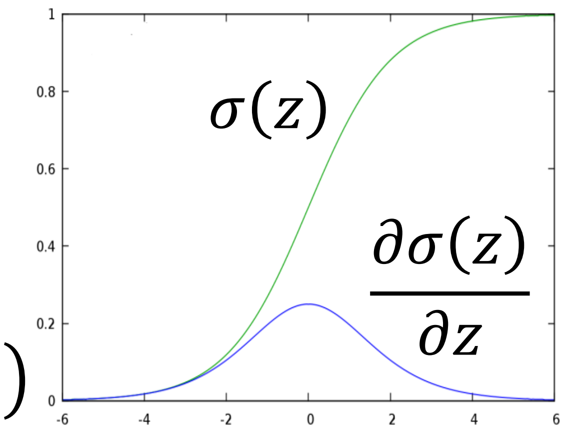
Question: Why don't we simply use square error as linear regression?

Step 3: Find the best function

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

$$\frac{\partial lnf_{w,b}(x)}{\partial w_i} = \frac{\partial lnf_{w,b}(x)}{\partial z}\frac{\partial z}{\partial w_i} \qquad \frac{\partial z}{\partial w_i} = x_i$$

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



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

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

Step 3: Find the best function

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

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

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

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

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

Step 3: Find the best function

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

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

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

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

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 | Linear Regression

Step 1:

Logistic Regression:

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

Output: between 0 and 1

Linear Regression:

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

Output: any value

Step 2:

Logistic Regression:

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

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

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

Linear Regression:

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

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

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

Step 3:

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

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

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

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

為了 Minimize Loss Function 針對 w 作微分

Step 3:

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

$$= 2\left(f_{w,b}(x) - \hat{y}\right) f_{w,b}(x)\left(1 - f_{w,b}(x)\right) x_i$$

$\hat{y}^n = 1$

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

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

Step 3:

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

$$= 2\left(f_{w,b}(x) - \hat{y}\right) f_{w,b}(x)\left(1 - f_{w,b}(x)\right) x_i$$

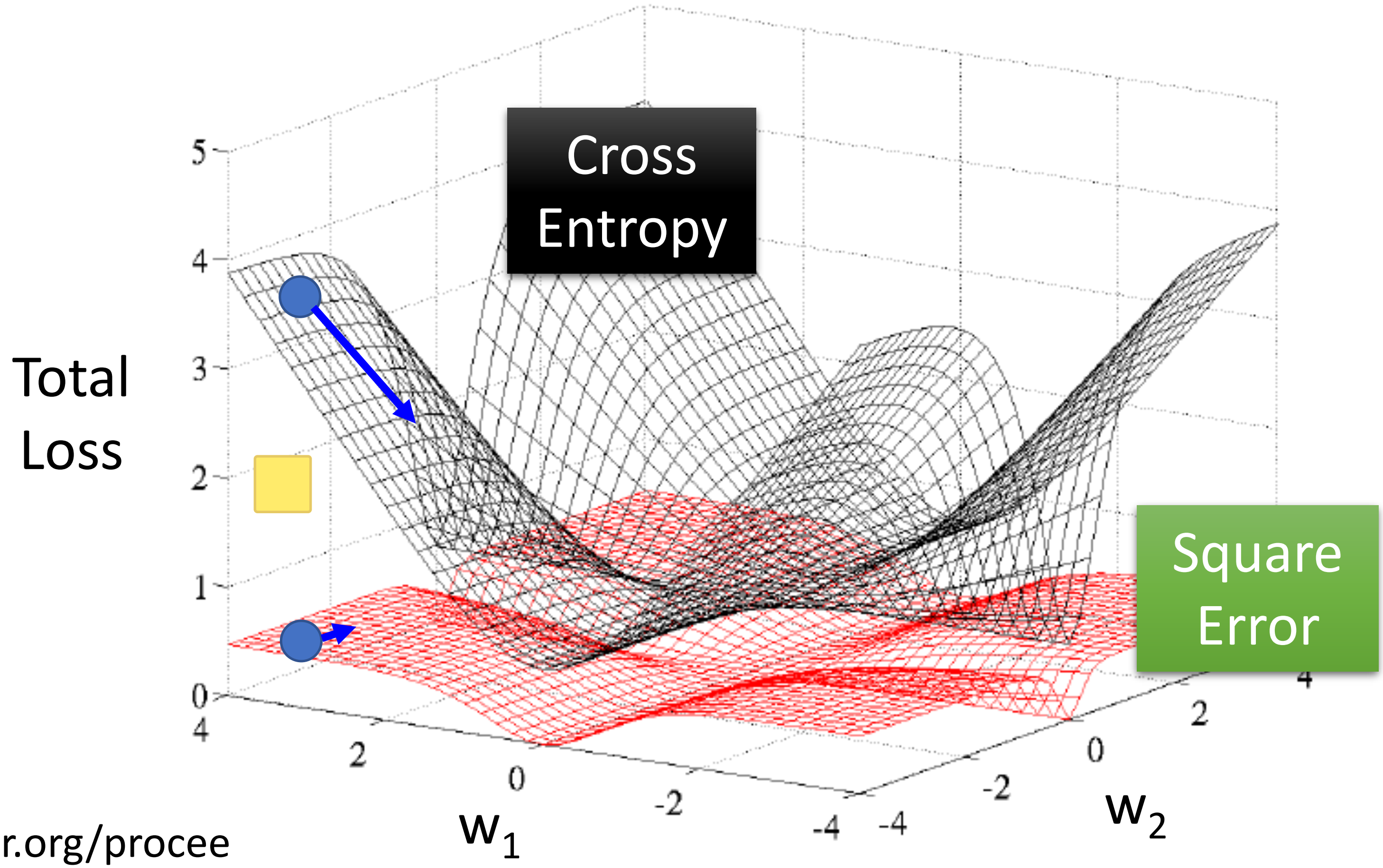
$\hat{y}^n = 0$

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) → $\partial L / \partial w_i = 0$

當實際值為 0 時也有相同情況！

Cross Entropy v.s. Square Error



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

Logistic Regression

Probabilistic Model

Discriminative v.s. Generative

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

directly find ***w*** and b

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

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

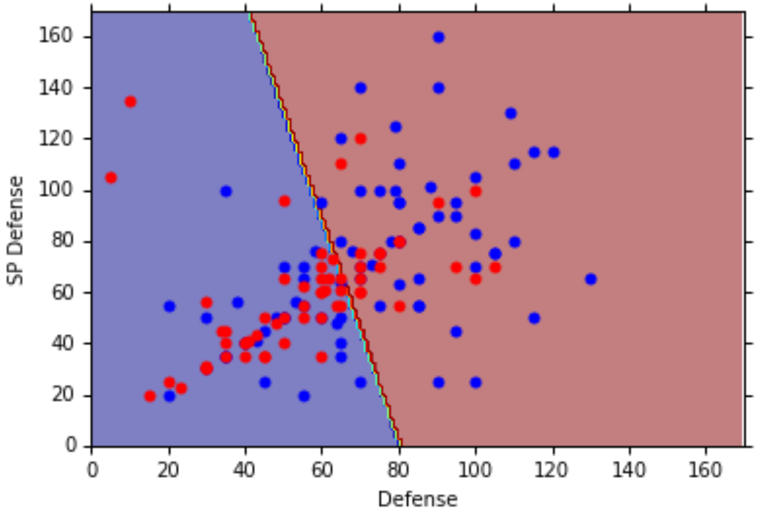
$$b = -\frac{1}{2}(\mu^1)^T(\Sigma^1)^{-1}\mu^1 + \frac{1}{2}(\mu^2)^T(\Sigma^2)^{-1}\mu^2 + ln\frac{N_1}{N_2}$$

Will we obtain the same set of w and b?

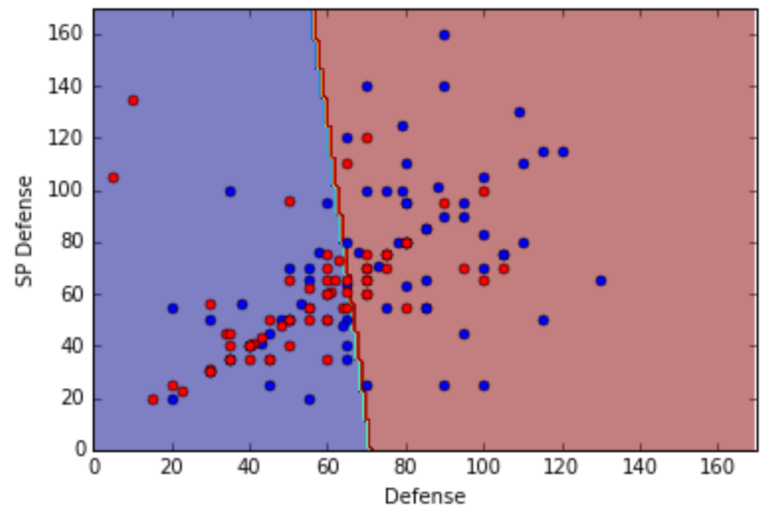
The same model (function set), but different function is selected by the same training data.

Generative v.s. Discriminative

Generative



Discriminative



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

73% accuracy

79% accuracy

解釋為什麼 Generative Model 會表現比較「差」？

Generative v.s. Discriminative

- Example

假設目前的 Training Data 中有 12 筆 Data，都有各自的 Label。每筆 Data 有兩個 Features

Training Data

Class 1	Class 2	Class 2	Class 2
1, 1	1, 0 (X 4)	0, 1 (X 4)	0, 0 (X 4)

Testing Data

1, 1

Class 1?

Class 2?

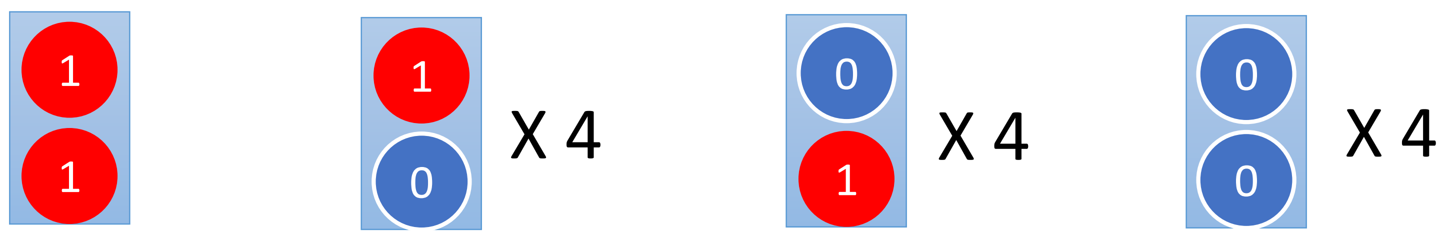
How about Naïve Bayes?

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

Generative v.s. Discriminative

- Example

Training Data

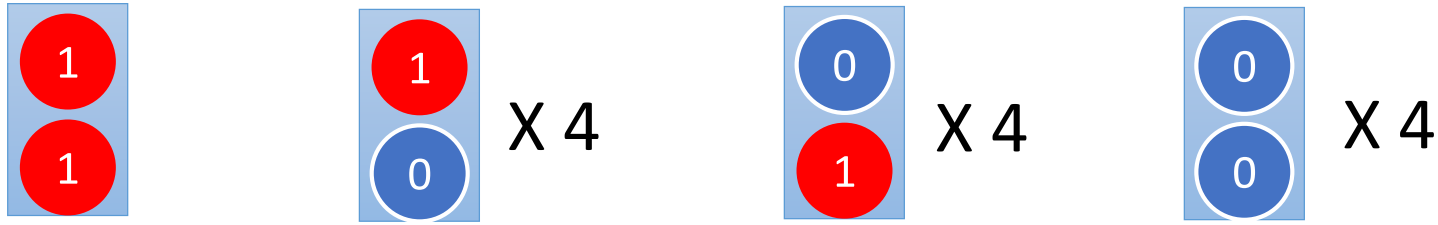


Class 1 Class 2 Class 2 Class 2

$$P(C_1) = \frac{1}{13} \qquad P(x_1 = 1|C_1) = 1 \qquad P(x_2 = 1|C_1) = 1$$

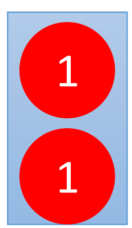
$$P(C_2) = \frac{12}{13} \qquad P(x_1 = 1|C_2) = \frac{1}{3} \qquad P(x_2 = 1|C_2) = \frac{1}{3}$$

Training Data



Class 1 Class 2 Class 2 Class 2

Testing Data



<0.5

$$P(C_1|x) = \frac{\overbrace{P(x|C_1)}^{1\times 1}\overbrace{P(C_1)}^{\frac{1}{13}}}{\underbrace{P(x|C_1)}_{1\times 1}\underbrace{P(C_1)}_{\frac{1}{13}} + \underbrace{P(x|C_2)}_{\frac{1}{3}\times\frac{1}{3}}\underbrace{P(C_2)}_{\frac{12}{13}}}$$

$$P(C_1) = \frac{1}{13} \qquad P(x_1 = 1|C_1) = 1 \qquad P(x_2 = 1|C_1) = 1$$

$$P(C_2) = \frac{12}{13} \qquad P(x_1 = 1|C_2) = \frac{1}{3} \qquad P(x_2 = 1|C_2) = \frac{1}{3}$$

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.

介紹 Logistic Regression 如何處理「多元分類」

[Bishop, P209-210]

Multi-class Classification (3 classes as example)

C$_1$: w^1, b_1

C$_2$: w^2, b_2

C$_3$: w^3, b_3

$$z_1 = w^1 \cdot x + b_1$$

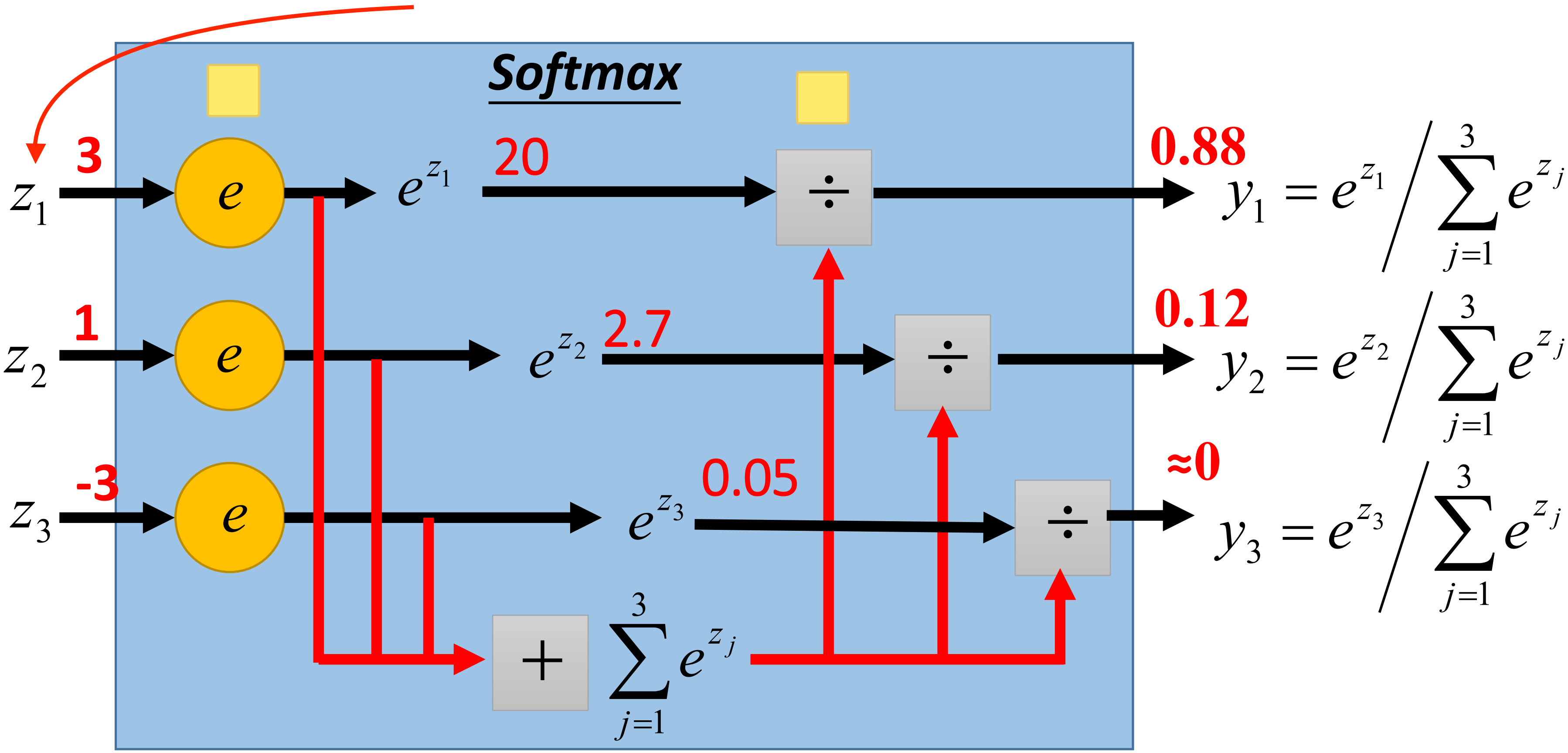
$$z_2 = w^2 \cdot x + b_2$$

$$z_3 = w^3 \cdot x + b_3$$

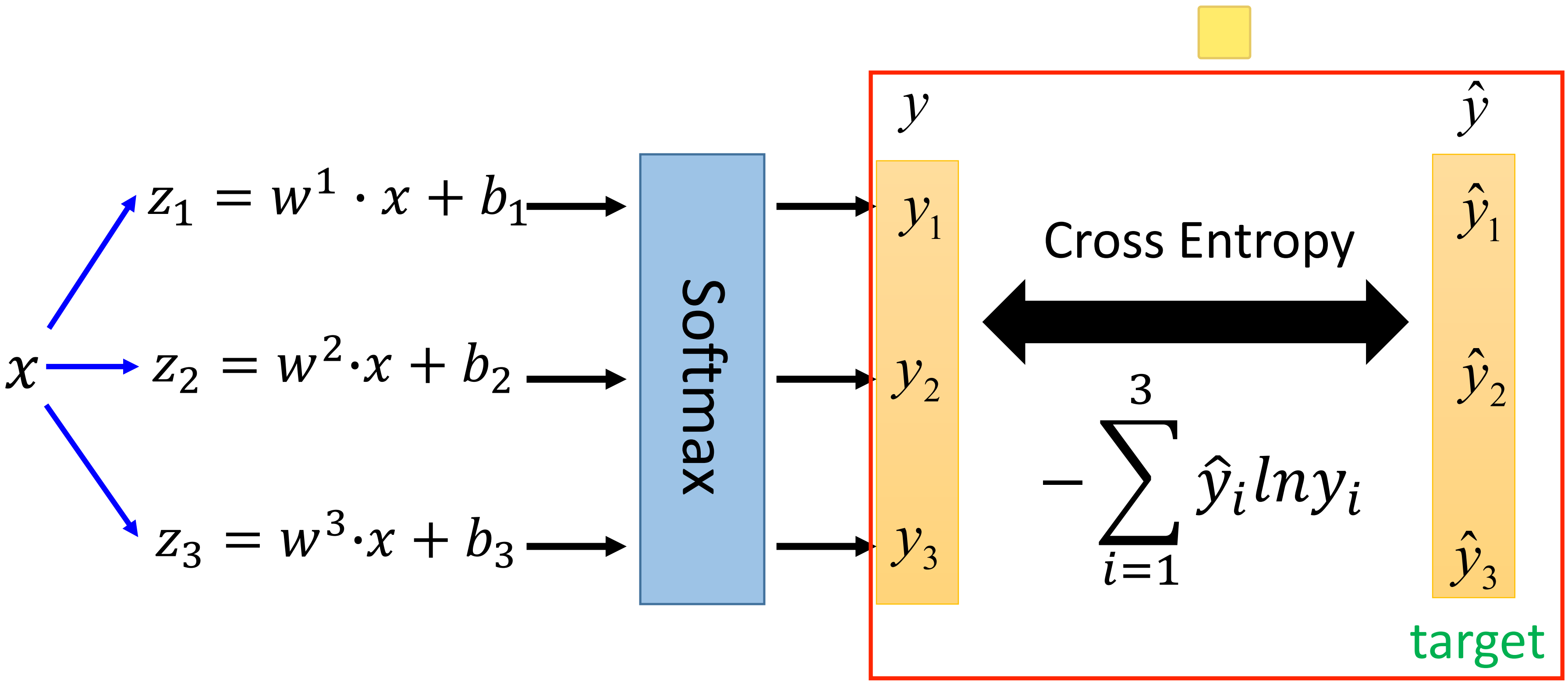
Probability:

- $1 > y_i > 0$
- $\sum_i y_i = 1$

$$y_i = P(C_i \mid x)$$



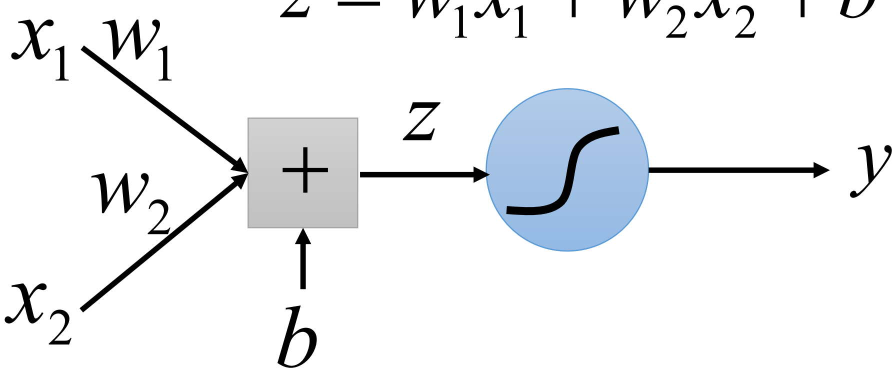
Multi-class Classification (3 classes as example)



If x ∈ class 1	If x ∈ class 2	If x ∈ class 3
$\hat{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\hat{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Limitation of Logistic Regression

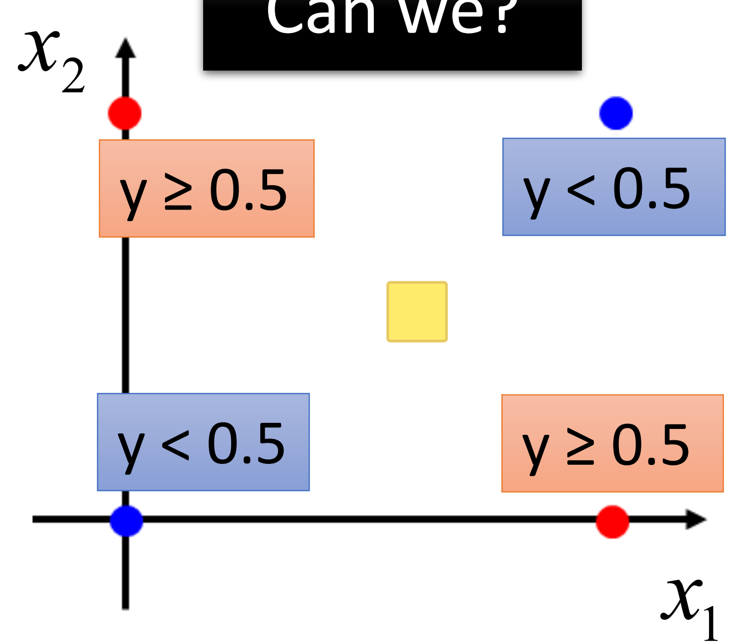
$$z = w_1 x_1 + w_2 x_2 + b$$



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

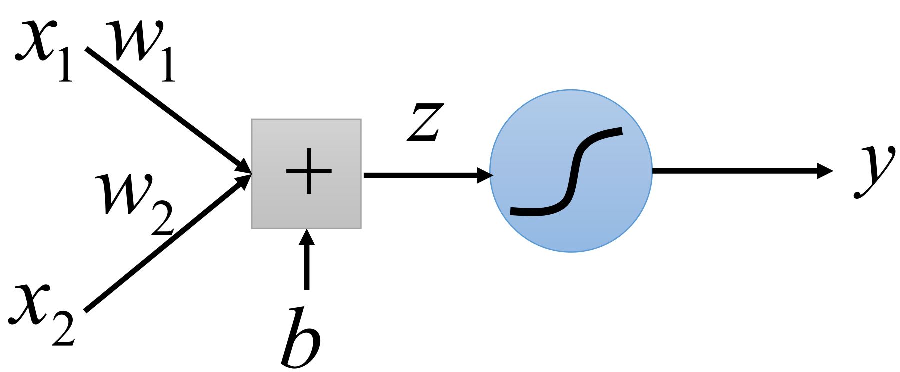
Input Feature		Label
x_1	x_2	
0	0	Class 2
0	1	Class 1
1	0	Class 1
1	1	Class 2

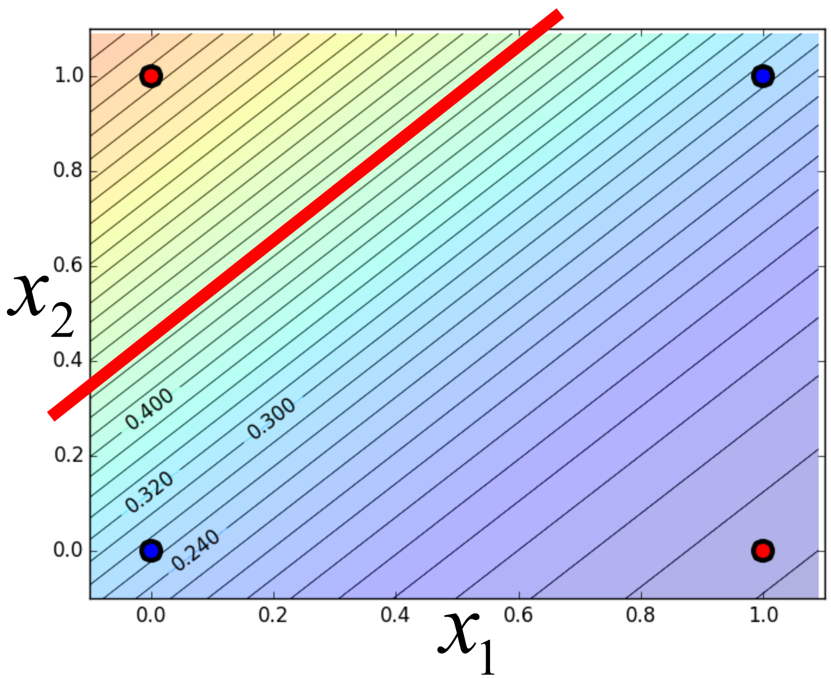
Can we?

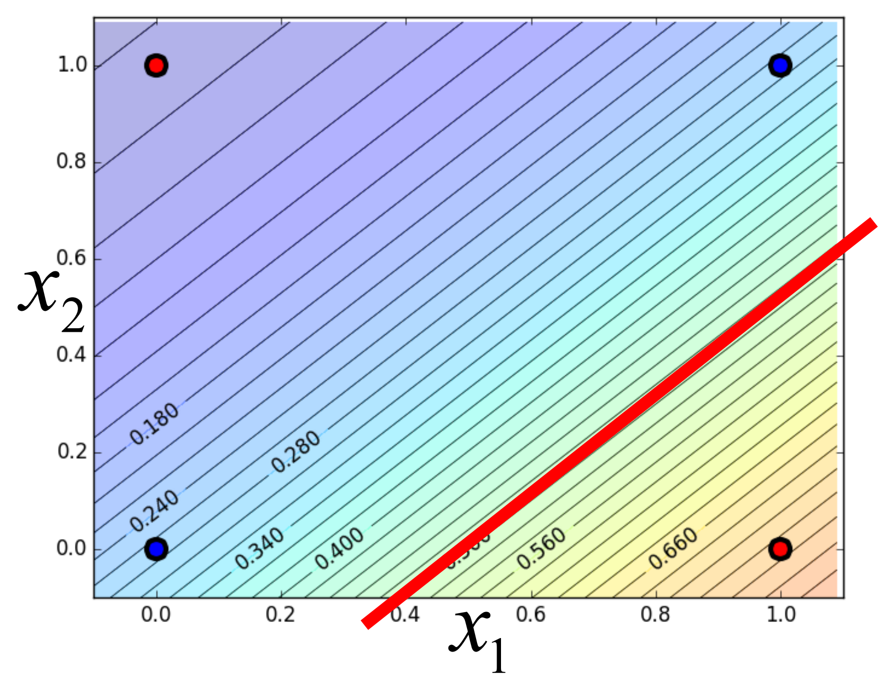


Limitation of Logistic Regression

- No, we can't ……





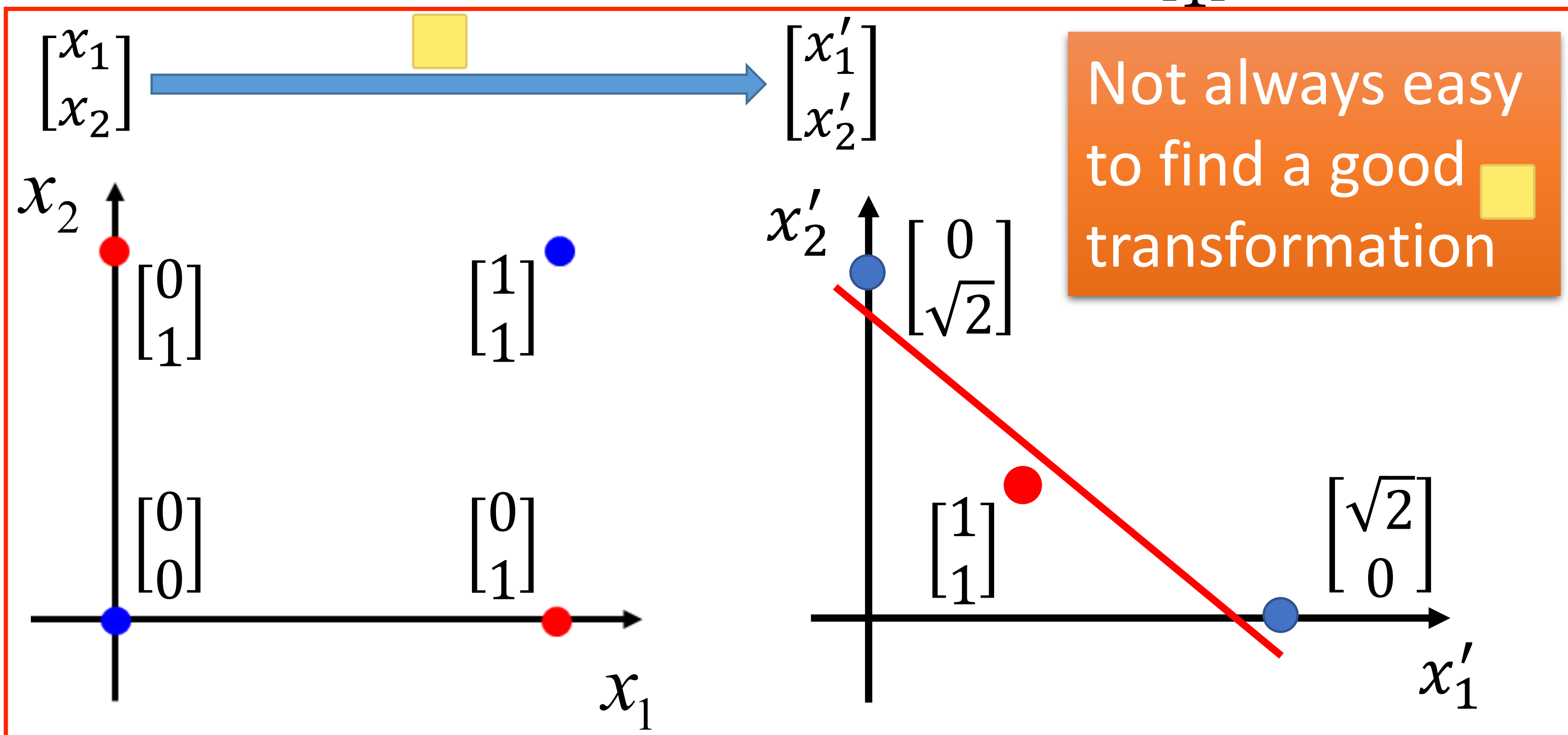


Limitation of Logistic Regression

- Feature Transformation

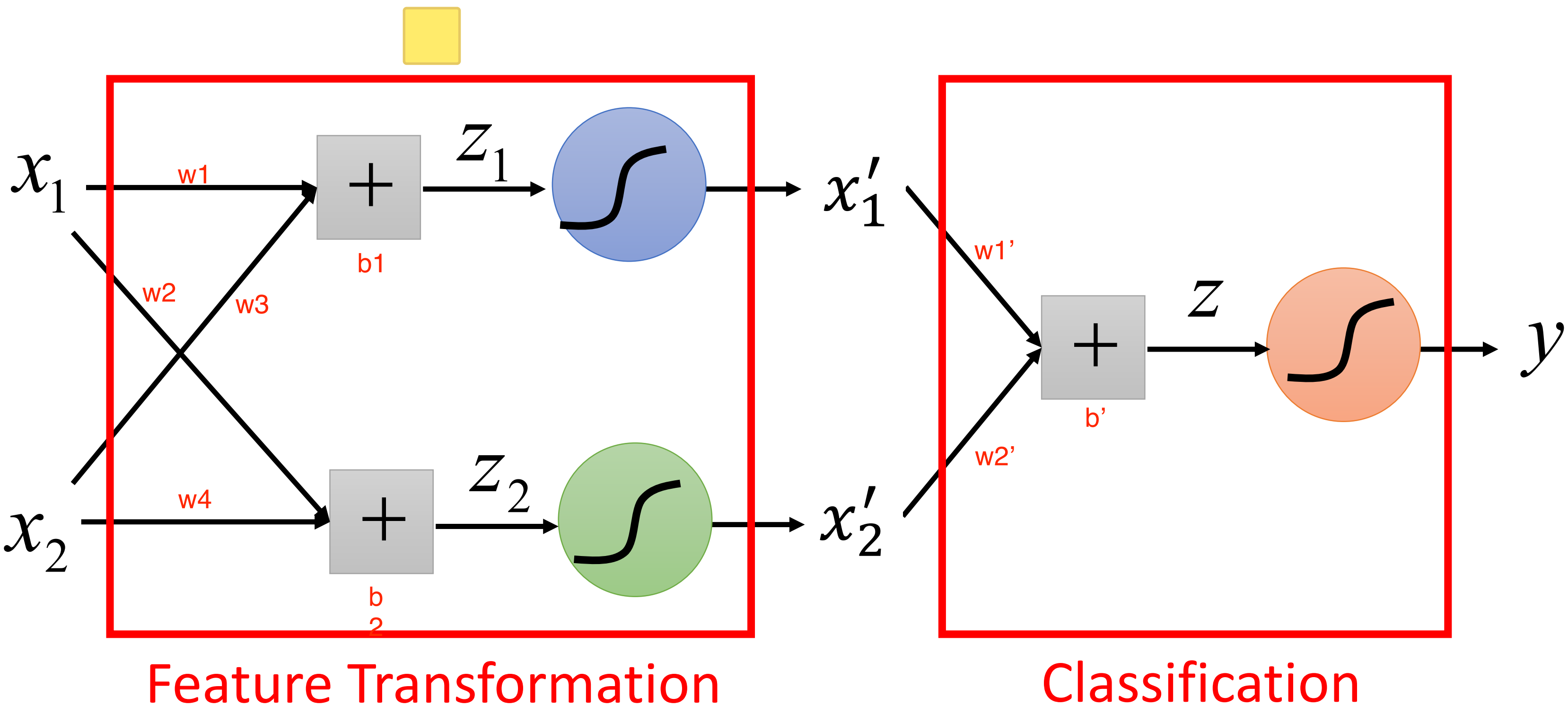
x_1': distance to $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

x_2': distance to $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

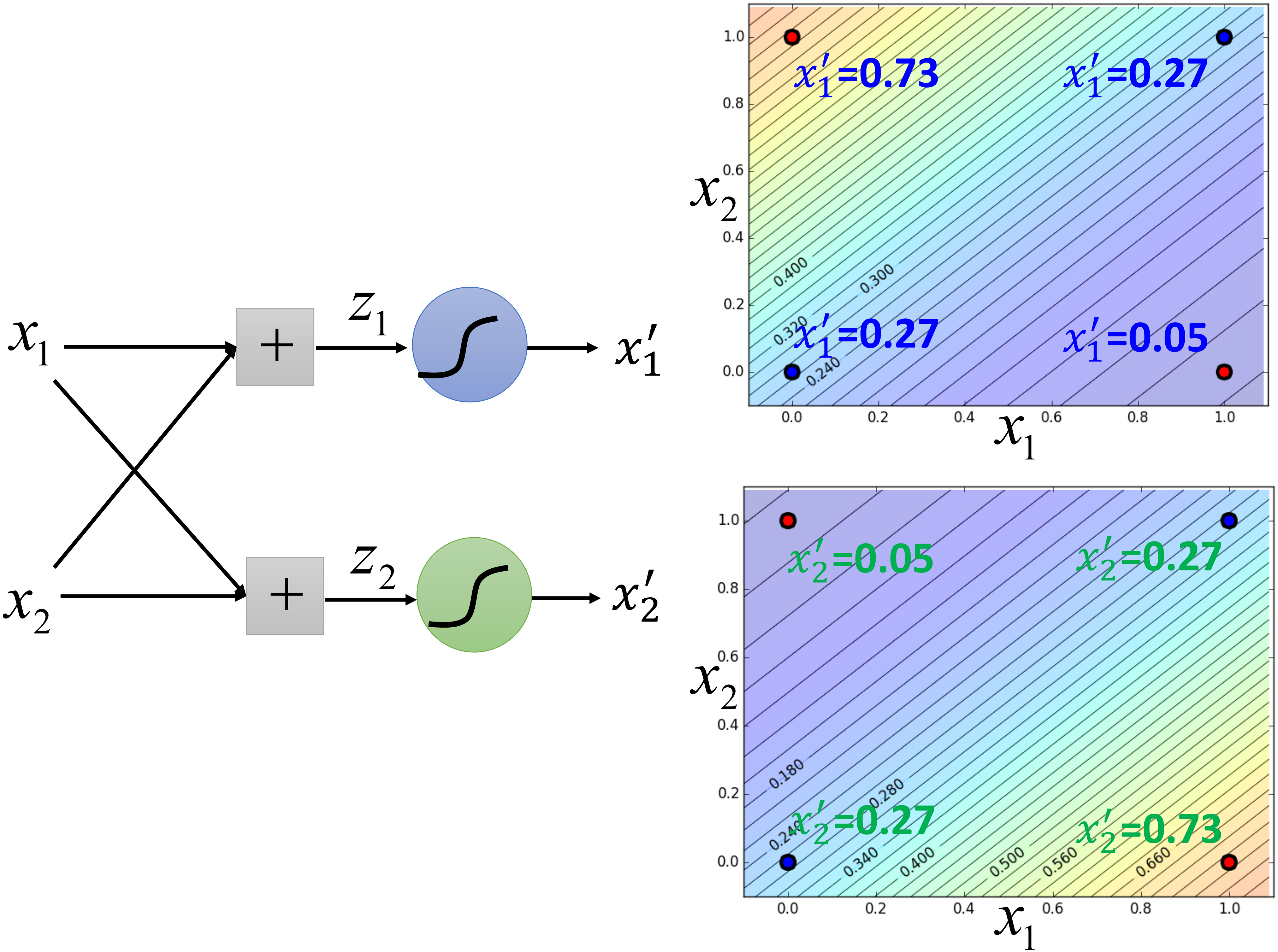


Limitation of Logistic Regression

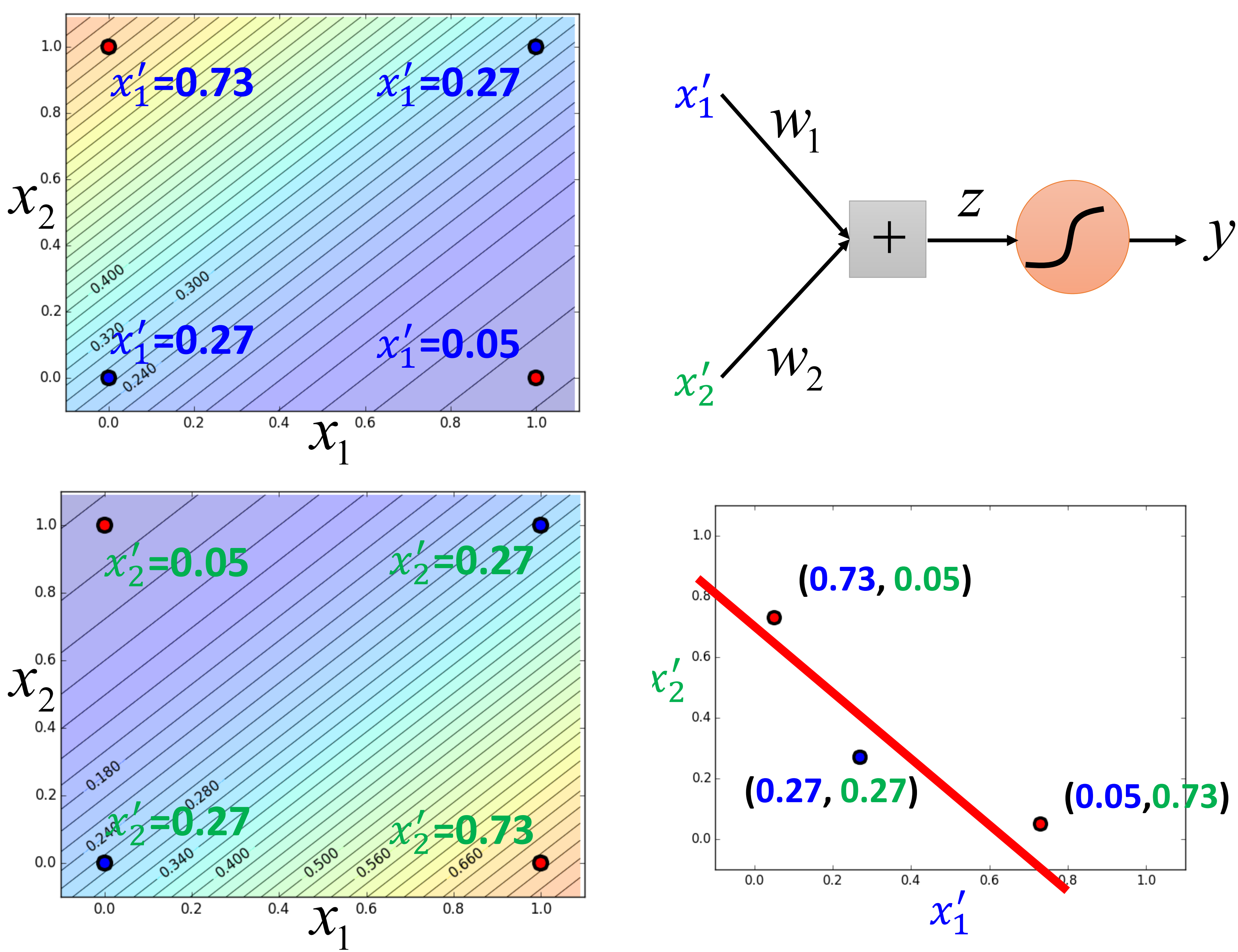
- Cascading logistic regression models



(ignore bias in this figure)

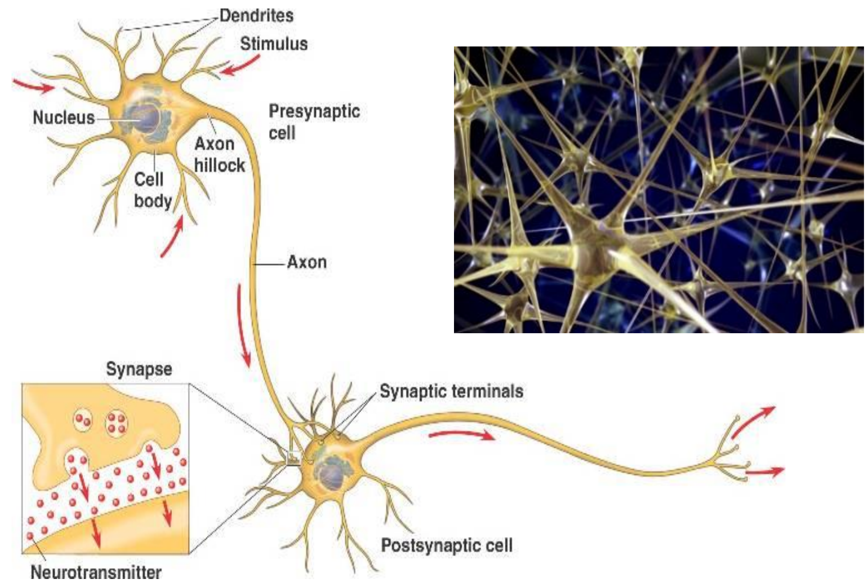


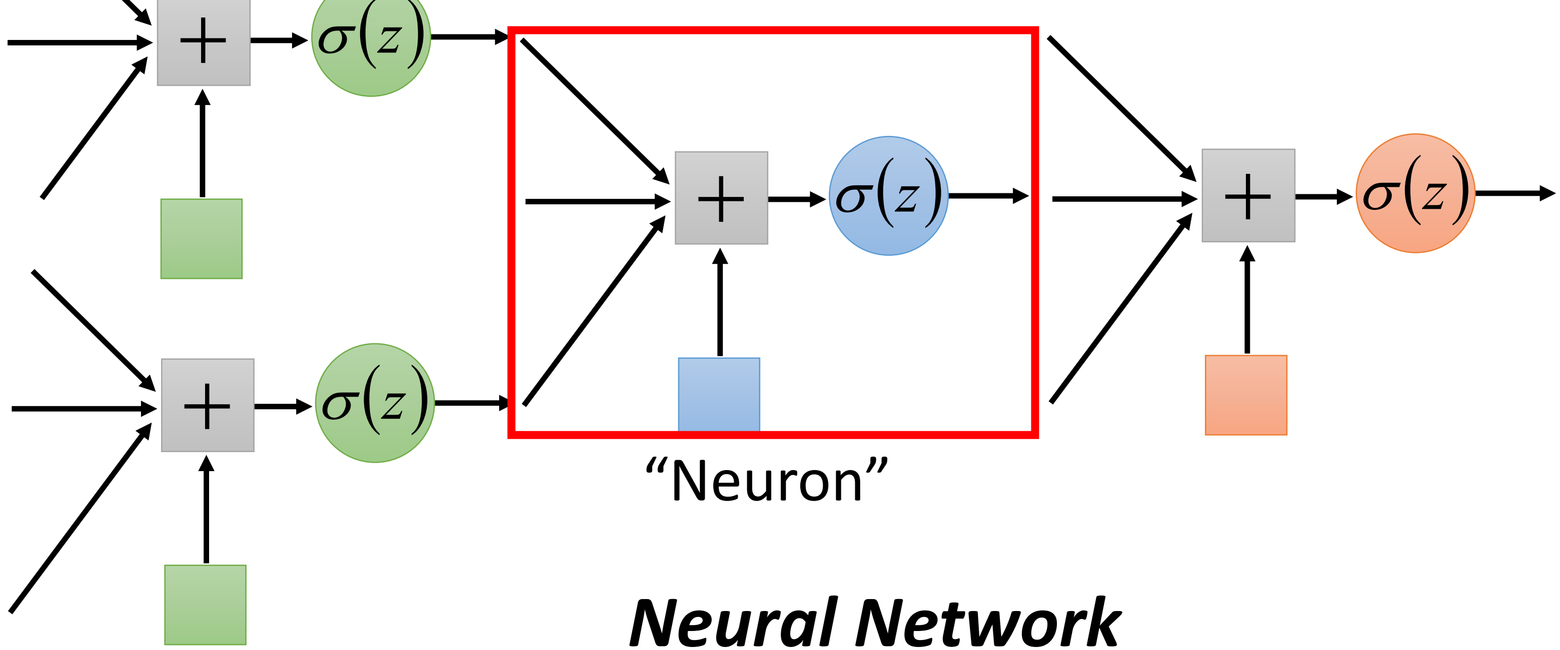
x_1
x_2
z_1
z_2
x'_1
x'_2
x'_1=0.73
x'_1=0.27
x'_1=0.27
x'_1=0.05
0.400
0.300
0.320
0.240
x'_2=0.05
x'_2=0.27
x'_2=0.27
x'_2=0.73
0.180
0.280
0.240
0.340
0.400
0.500
0.560
0.660
0.0
0.2
0.4
0.6
0.8
1.0



x_1'=0.73
x_1'=0.27
x_1'=0.27
x_1'=0.05
0.400
0.300
0.320
0.240
x_1
x_2
x_1'
x_2'
w_1
w_2
z
y
+
x_2'=0.05
x_2'=0.27
x_2'=0.27
x_2'=0.73
0.180
0.280
0.240
0.340
0.400
0.500
0.560
0.660
(0.73, 0.05)
(0.27, 0.27)
(0.05,0.73)

Deep Learning!





Neural Network