

Lecture 4

Logistic (Regression) classification

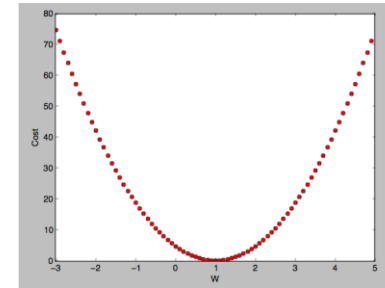
Dong Kook Kim

Regression : Recap

x1 (hours)	x2 (attendance)	y (score)
10	5	90
9	5	80
3	2	50
2	4	60
11	1	40

- Hypothesis: $H(X) = WX$

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



- Gradient decent:

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

Step I: Binary Classification Data

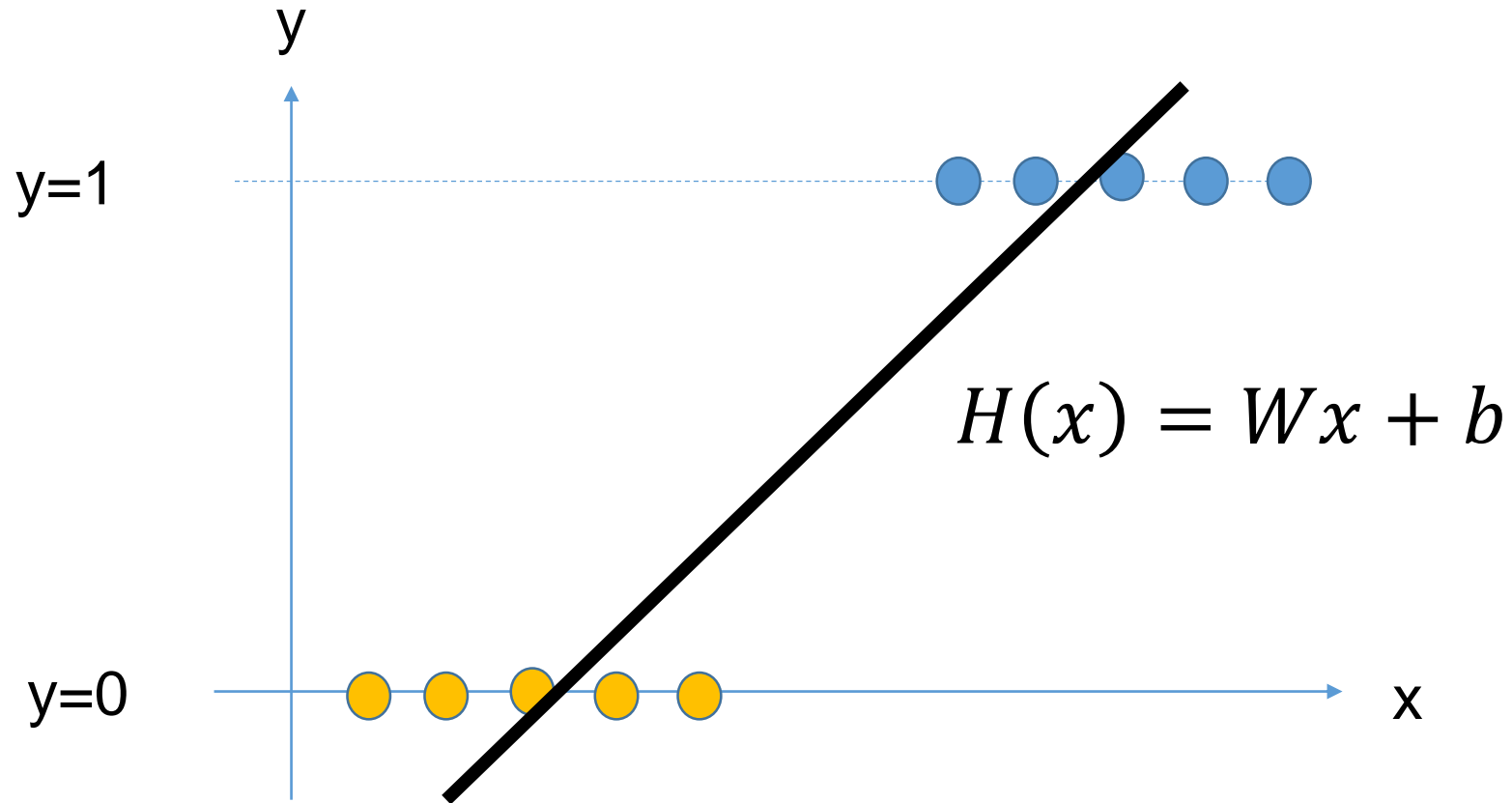
- Target : 0 or 1 (Binary)

x1 (hours)	x2 (attendance)	y (P or F)	y (target)
10	5	Pass (90)	1
9	5	Pass (80)	1
3	2	Fail (50)	0
2	4	Fail (60)	0
11	1	Fail (40)	0

Target : 0, 1 encoding

- Target : 0 or 1 (binary)
- Spam Detection: Spam (1) or Ham (0)
- Tumor Detection: Malignant(1) or Not (0)
- Facebook feed: show(1) or hide(0)
- Credit Card Fraudulent Transaction detection: legitimate(0) or fraud (1)

Linear Regression Model



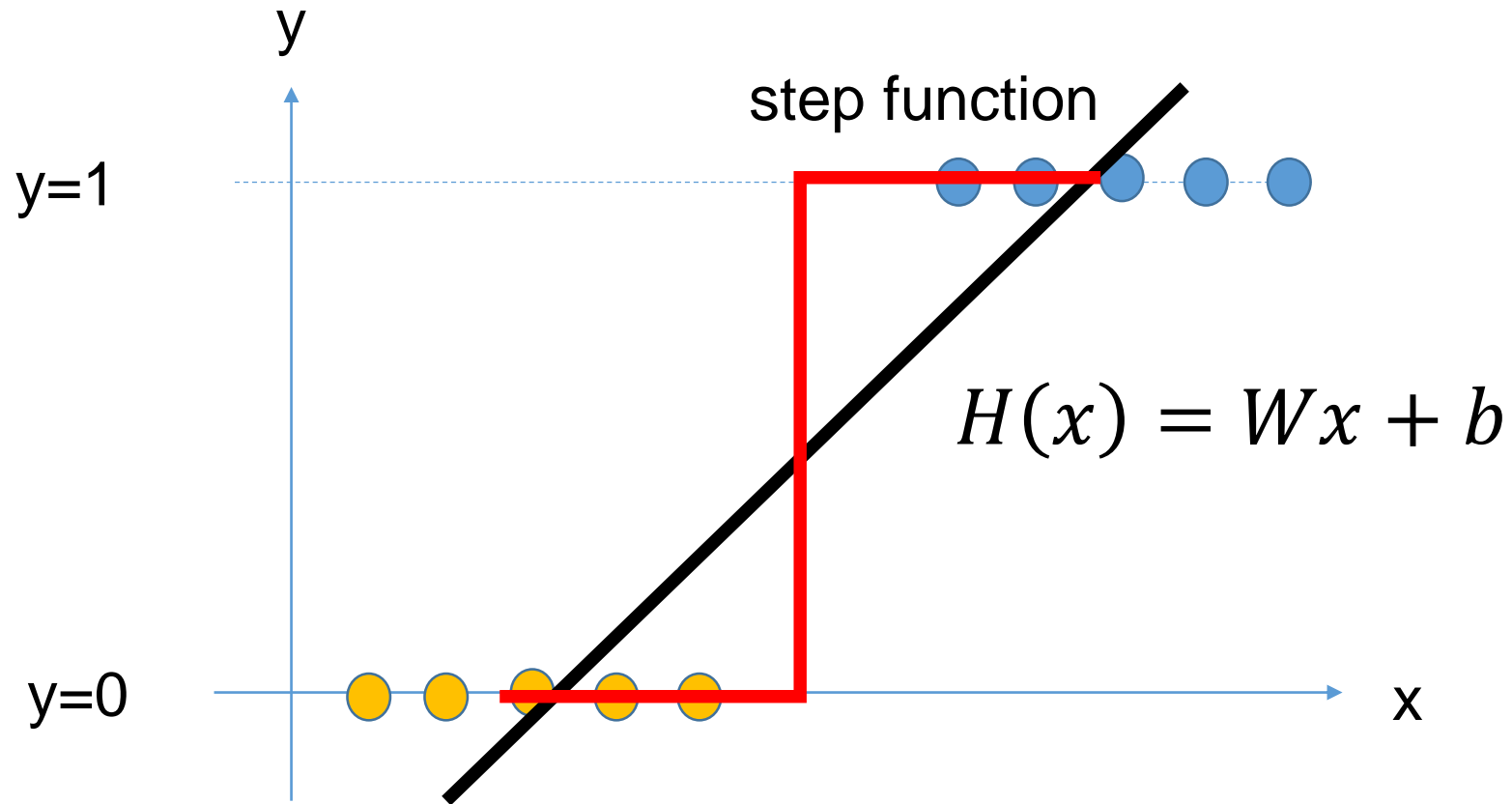
Logistic Regression

- We know y is 0 or 1

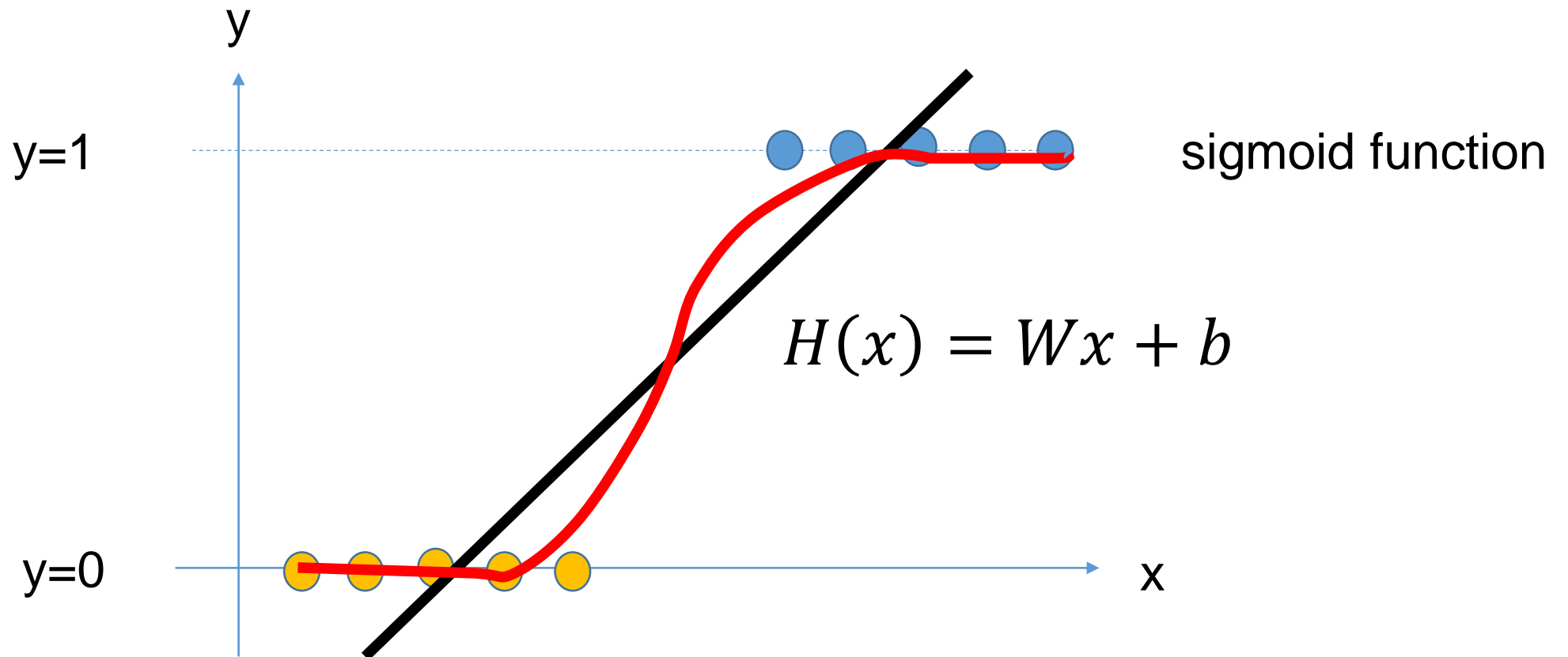
$$H(x) = Wx + b : \text{linear regression}$$

- Hypothesis can give values larger than 1 or less than 0
- A **nonlinear function** is need to represent 0 or 1 hypothesis

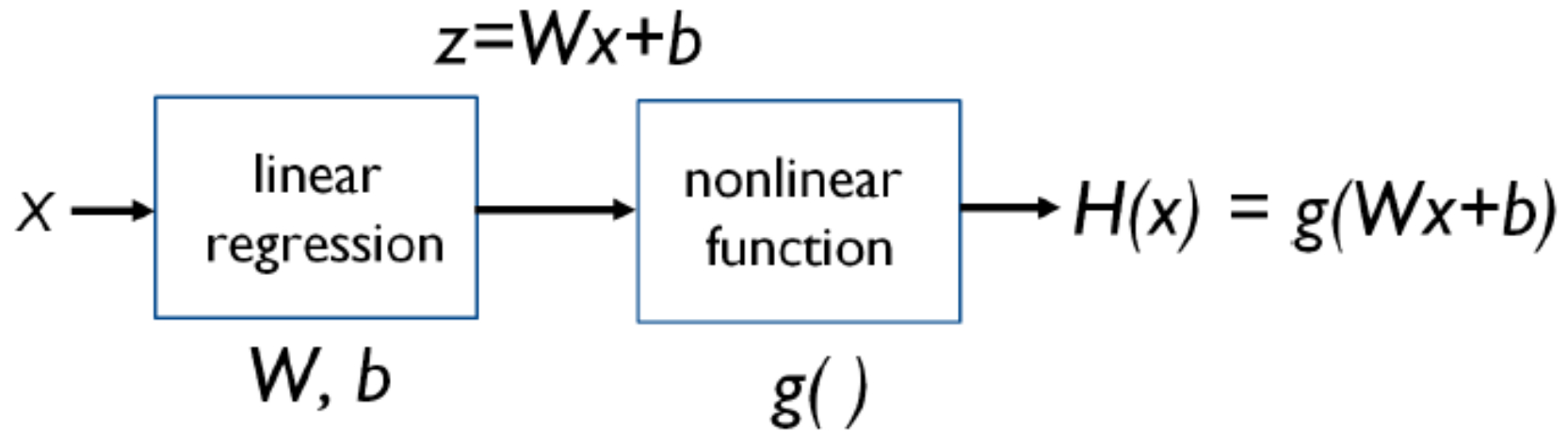
Nonlinear function : Step function



Nonlinear function : Sigmoid function



Step 2: Logistic Regression Model



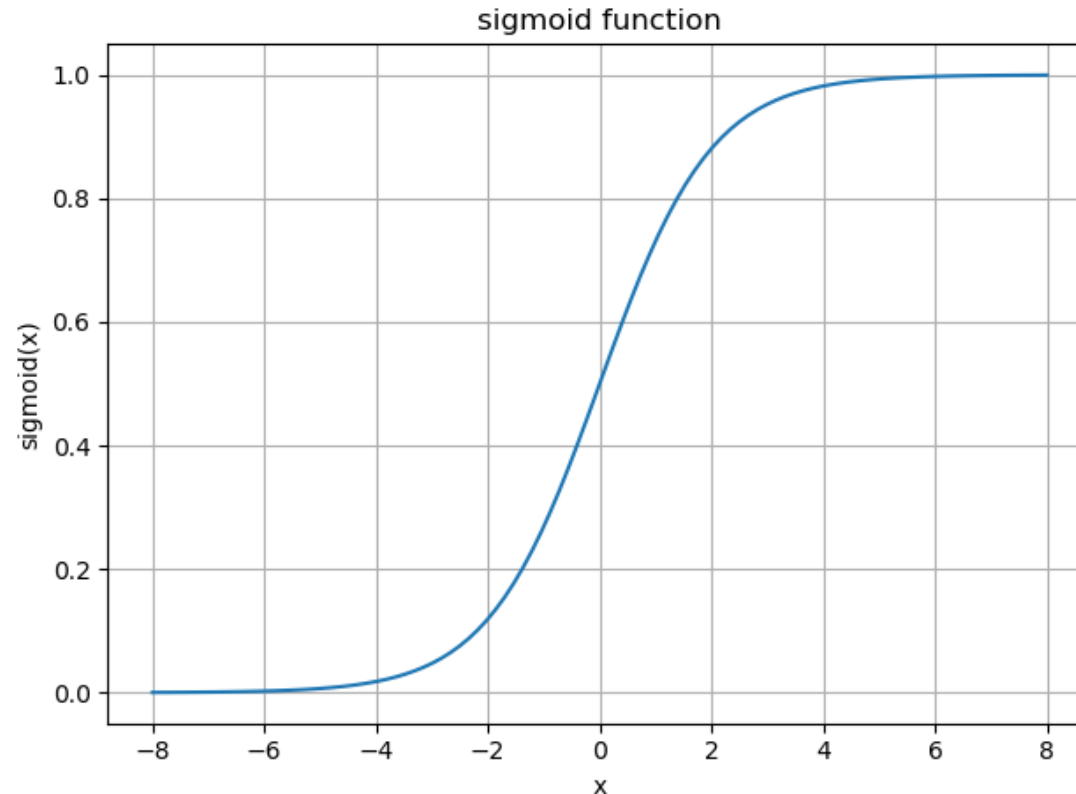
$$z = Wx + b$$

$$H(x) = g(z)$$

Sigmoid Function

- Logistic function or Sigmoid function
- Curved in two directions, liked the letter 'S'

$$y = \sigma(x) = \frac{1}{1 + e^{-x}}$$

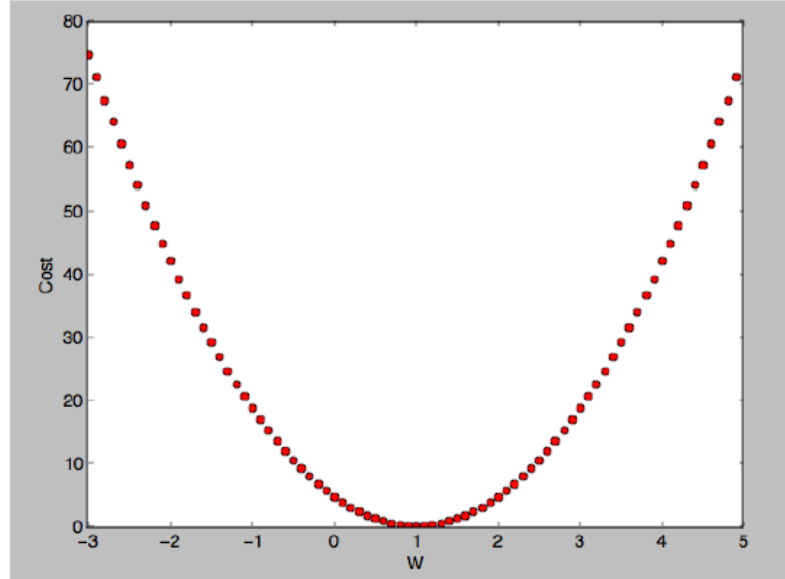


Step 2: Logistic Regression Model

$$H(X) = \sigma(Wx + b) = \frac{1}{1 + e^{-(Wx+b)}}$$

Loss Function : LR

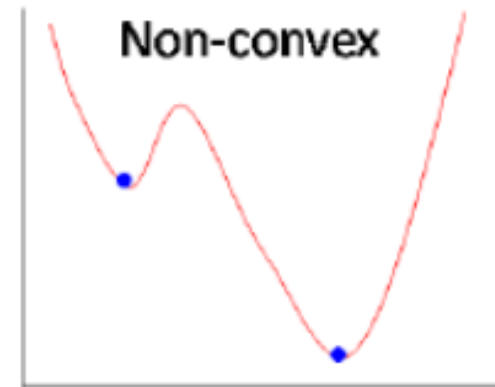
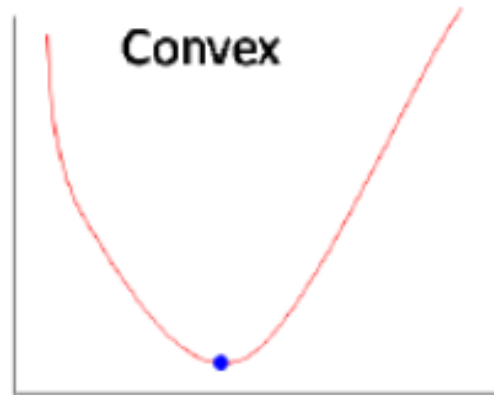
$$cost(W, b) = \frac{1}{m} \sum_{i=1}^m (H(x^{(i)}) - y^{(i)})^2 \text{ when } H(x) = Wx + b$$



Loss Function : MSE

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

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



Step 3: New Loss Function for Logistic

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

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

Cross – entropy function

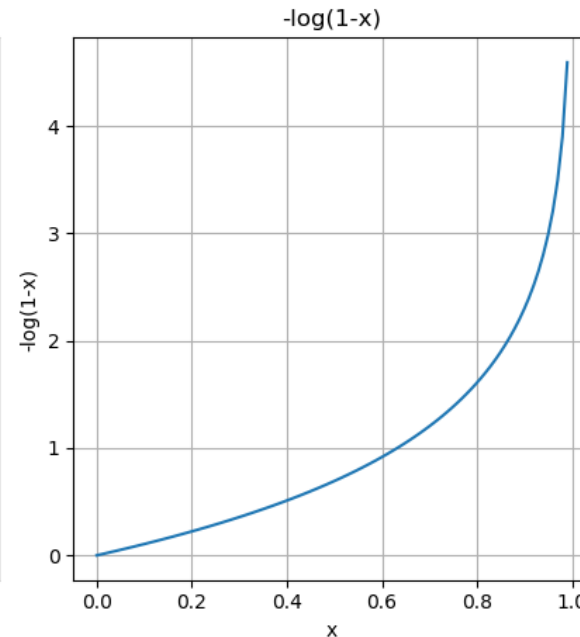
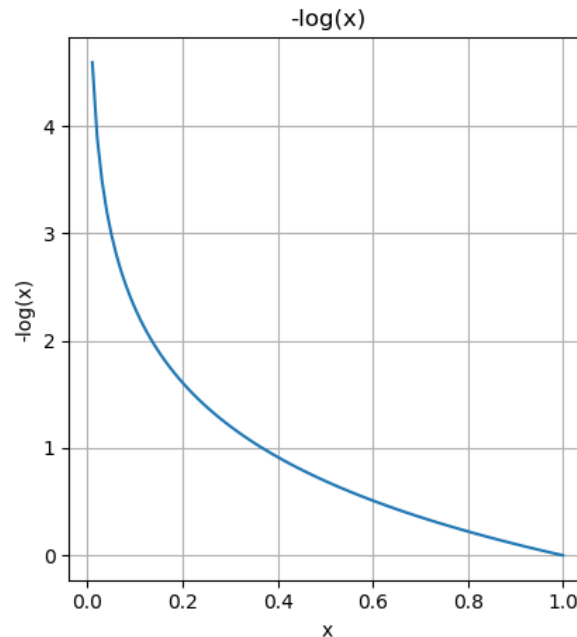
Understanding Loss Function

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

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

If $y = 0$, $C(H(x), y) = -\log(1 - H(x))$

$H(x) = 1 \rightarrow cost = 0$
 $H(x) = 0 \rightarrow cost = \infty$



$H(x) = 1 \rightarrow cost = \infty$
 $H(x) = 0 \rightarrow cost = 0$

Step 3: Loss Function - Cross-Entropy

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

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

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

Step 4: Optimization – GD algorithm

$$\text{cost}(W, b) = -\frac{1}{m} \sum y \log(H(x)) + (1 - y) \log(1 - H(x))$$

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

- Derivative cost(W) wrt W

$$\frac{\partial}{\partial W} \text{cost}(W, b) = \sum (y - H(x))x$$

Optimization – GD algorithm

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

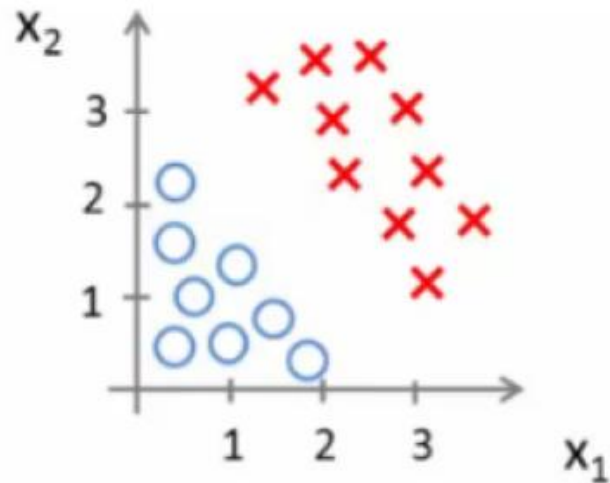
$$W := W - \alpha \frac{\partial}{\partial W} cost(W)$$

- Keras : `tensorflow.keras.losses.binary_crossentropy`
- `model.compile(loss='binary_crossentropy', optimizer=sgd)`

Step 5: Testing - Decision Boundary

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

Decision : if $H(x) > 0.5, y = 1$, else $y = 0$



Decision Boundary

$$H(x) = 0.5 \quad x_2$$

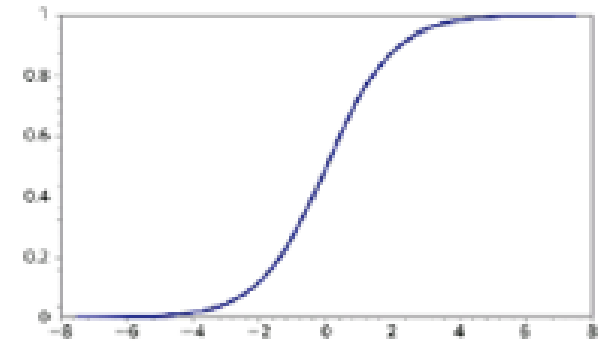
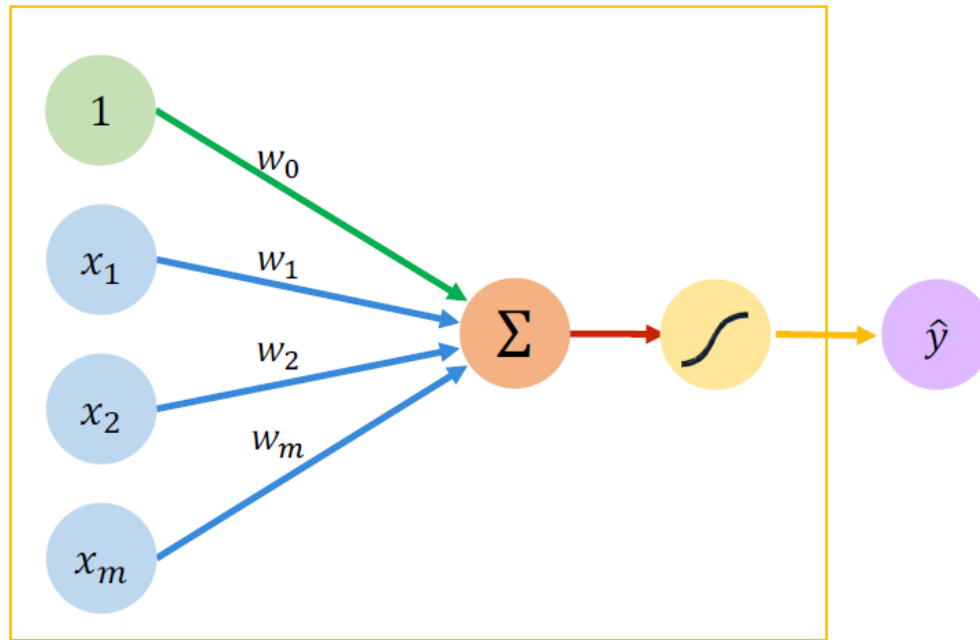


Graph for Logistic Regression

- x : n-dim vector, y : scalar
- W : n-dim vector, b : scalar

$$y = H(X) = \sigma(Wx + b)$$

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



- Sigmoid function

Keras : `Dense(l, input_dim=n, activation='sigmoid')`

Logistic Regression : Summary

- **Muti-variate Input/Scalar Target:**

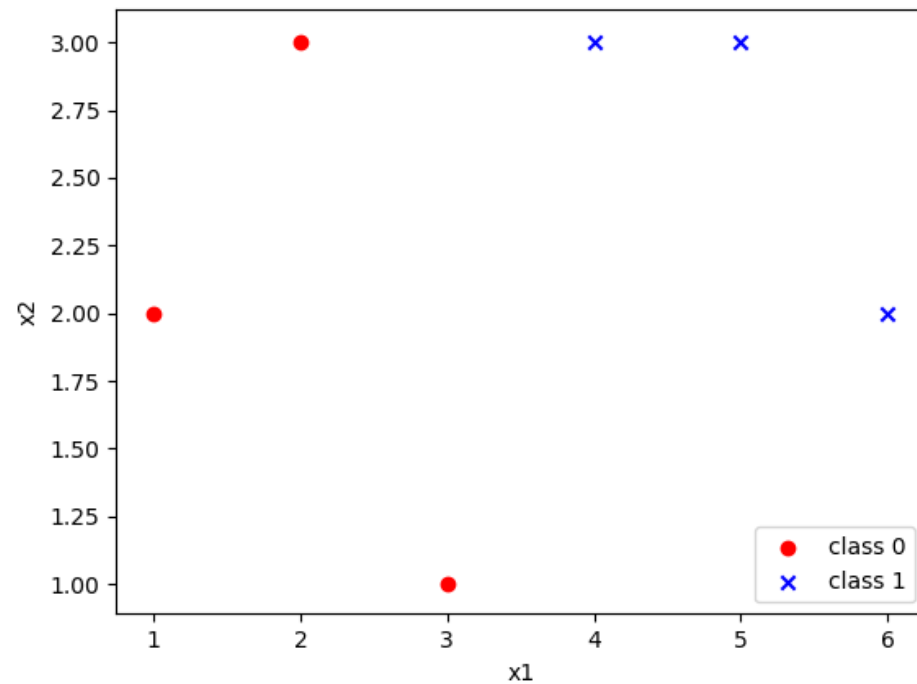
Data Set :	$x^{(i)}, y^{(i)}, i = 1, \dots, m$	Input vector & scalar target (0 or 1)
Model :	$H(X) = \frac{1}{1 + e^{-(Wx+b)}}$	Linear model W : weight vector b : bias
Cost Function	$\text{cost}(W, b) = -\frac{1}{m} \sum y \log(H(x)) + (1 - y) \log(1 - H(x))$	CE
Optimization	$W := W - \alpha \frac{\partial \text{cost}(W, b)}{\partial W}$	GD
Testing	<i>Given x & W, b decide $y = 1$ if $H(x) > 0.5$, else $y = 0$</i>	Metric : Accuracy

Exercise 03-1.

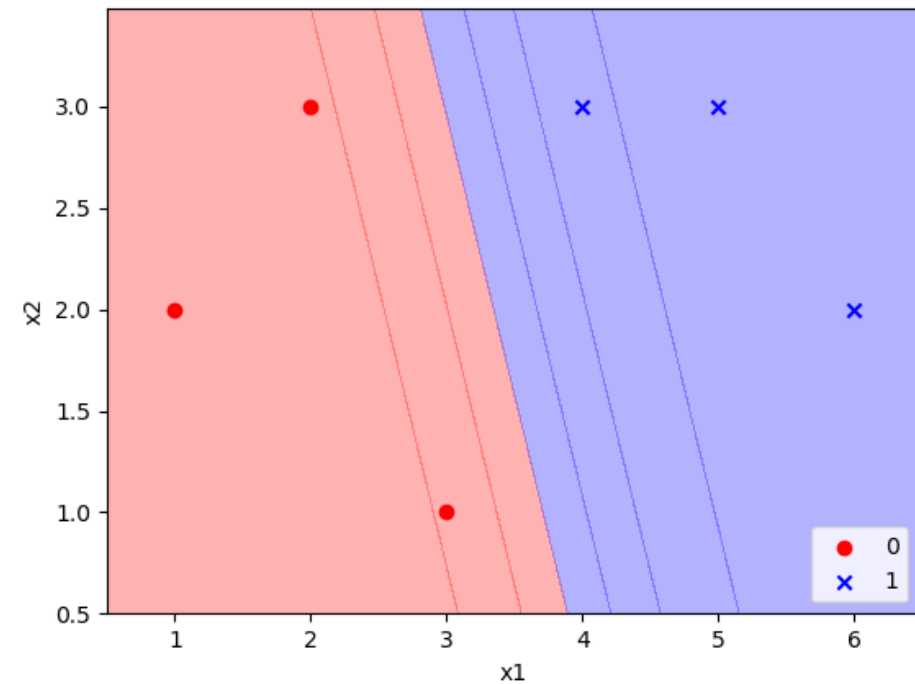
`tf2-03-1-logistic_classification.py`

Exercise 03-I.

- Data: input and target



- Classification results



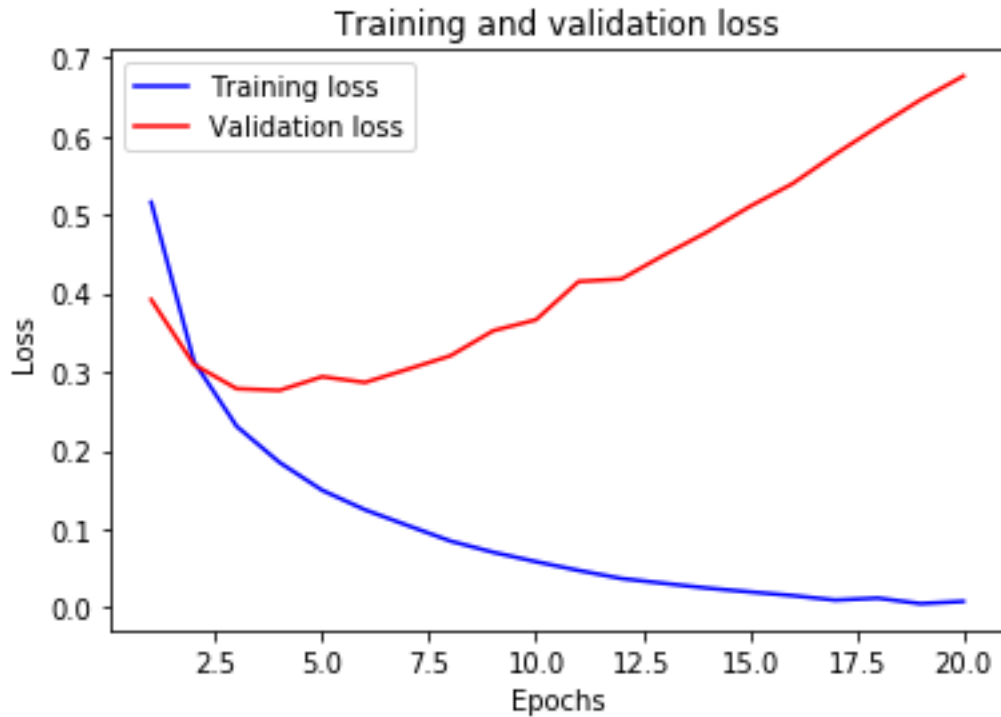
Exercise 03-2.

tf2-03-2-imdb_classification.py

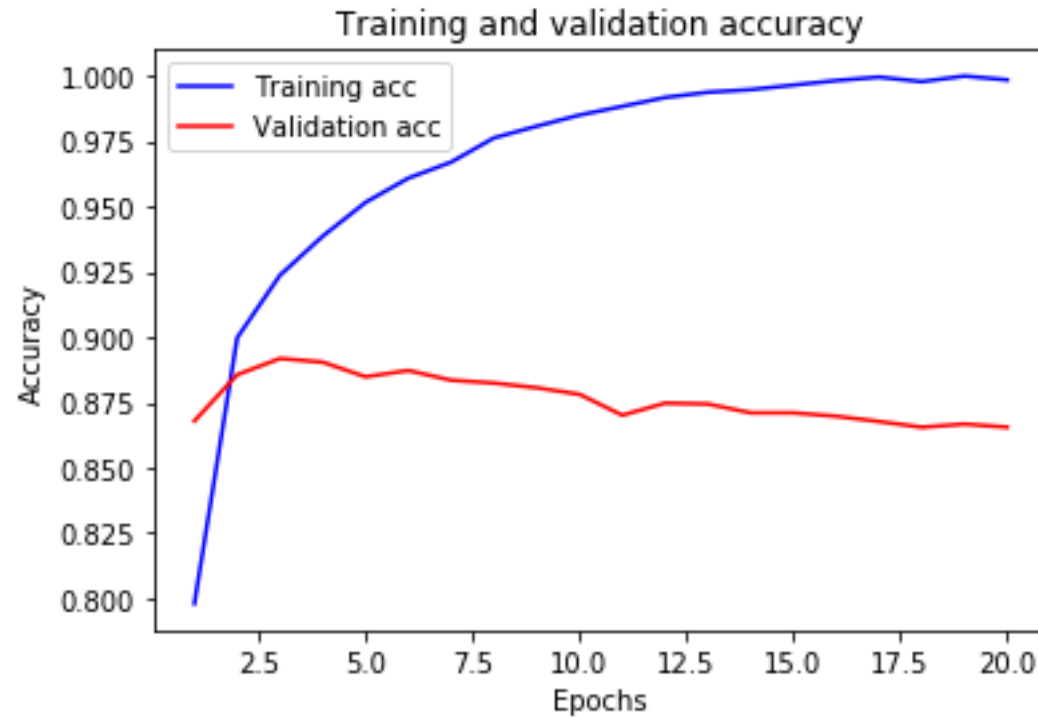
- movie review classification

Exercise 03-2.

Training & validation loss



Training & validation accuracy



Test accuracy : 85.03%