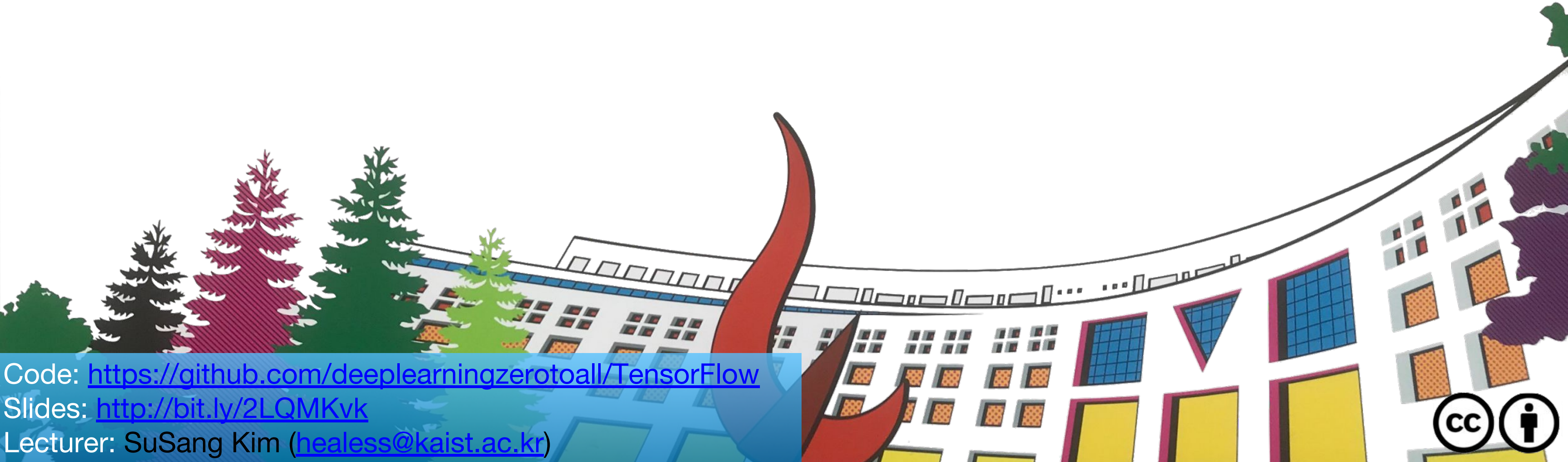
ML/DL for Everyone Season2

with TensorFlow

Lab 05-1 Logistic Regression



Code: https://github.com/deeplearningzerotoall/TensorFlow
Slides: http://bit.ly/2LQMKvk
Lecturer: SuSang Kim (healess@kaist.ac.kr)

Logistic Regression

- What is Logistic Regression?
  - Classification
  - Logistic vs Linear
- How to solve?
  - Hypothesis Representation
  - Sigmoid/Logistic Function
  - Decision Boundary
  - Cost Function
  - Optimizer (Gradient Descent)
- Codes
- Summary

Andrew Ng’s ML Class http://www.holehouse.org/mlclass/

Classification

What is Binary(Multi-class) Classification?

variable is either 0 or 1 (0:positive / 1:negative)

- Exam : Pass or Fail
- Spam : Not Spam or Spam
- Face : Real or Fake
- Tumor : Not Malignant or Malignant

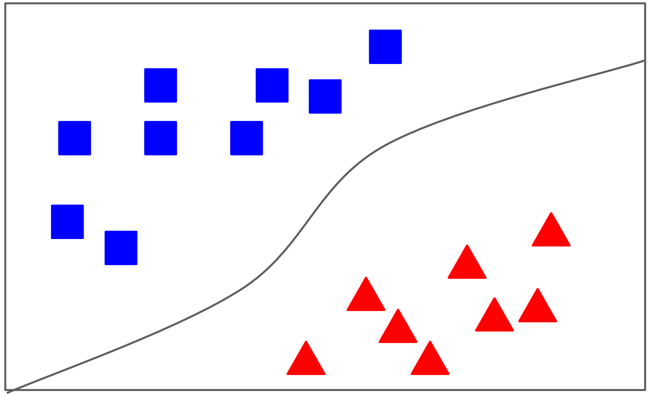
To start with machine learining, you must encode variable [0,1]

[Python Code]

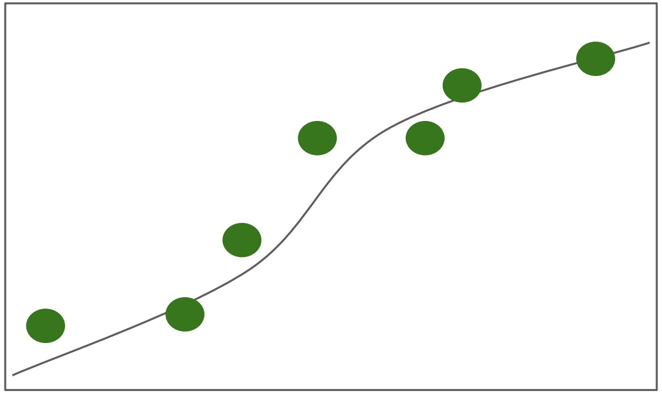
```
x_train = [[1, 2], [2, 3], [3, 1], [4, 3], [5, 3], [6, 2]]
y_train = [[0], [0], [0], [1], [1], [1]] # One Hot
```

Logistic vs Linear

What is the difference between logistic and linear?



VS



Discrete (Counted)
Shoe Size /The number of workers in a company

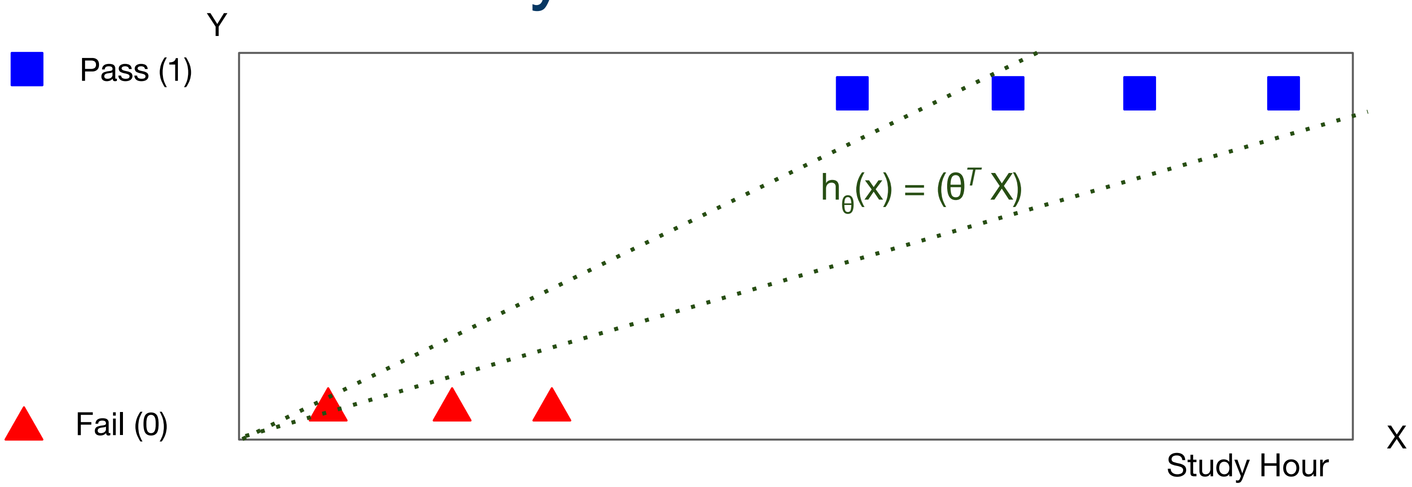
Continous (Measured)
Time / Weight / Height

[Python Code]

```
Logistic_Y= [[0], [0], [0], [1], [1], [1]] # One Hot
Linear_Y = [828.659973, 833.450012, 819.23999, 828.349976, 831.659973] # Numeric
```

Hypothesis Representation

Study Hours ⇔ Pass/Fail



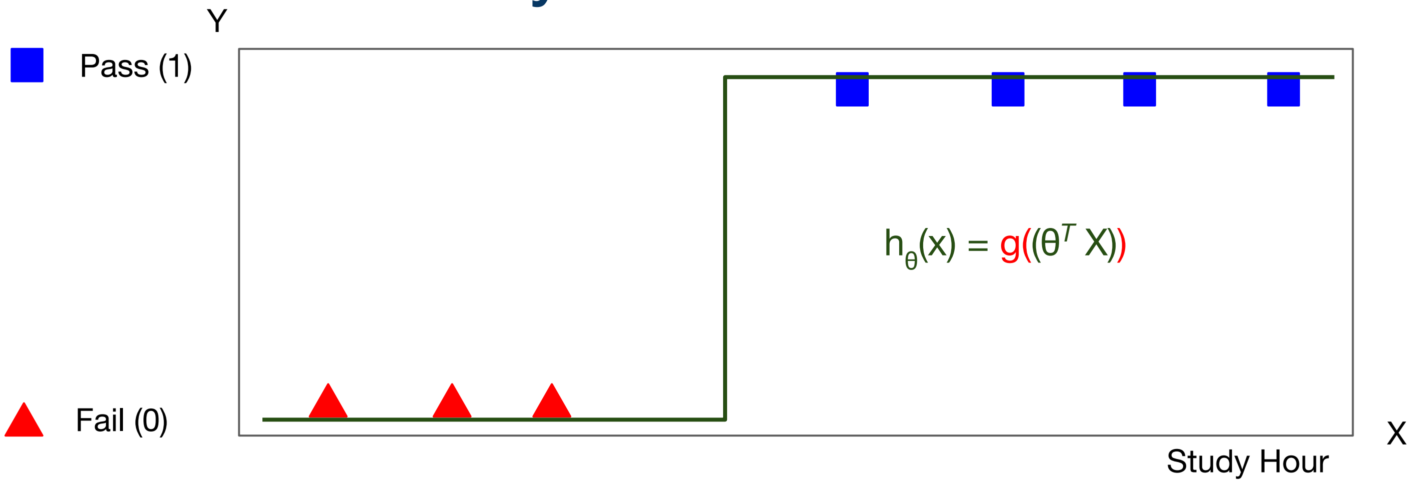
When using linear regression we did $h_\theta(x) = (\theta^T X)$
But start with binary classification, Y is only 0 or 1
So we need new function.

[Tensorflow Code]

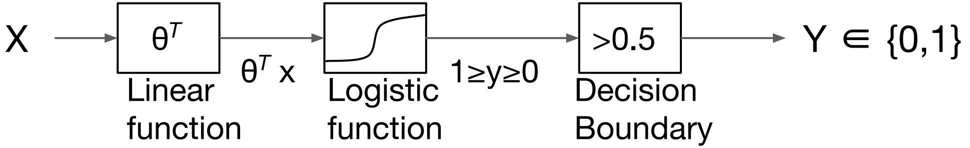
```
hypothesis  = tf.matmul(X, θ) + b #linear θ^T is an [1xn+1] matrix / θ_i are parameters
```

Hypothesis Representation

Study Hours ⇔ Pass/Fail

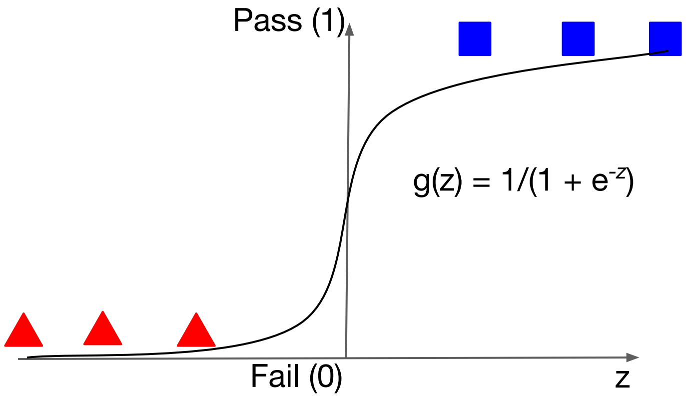


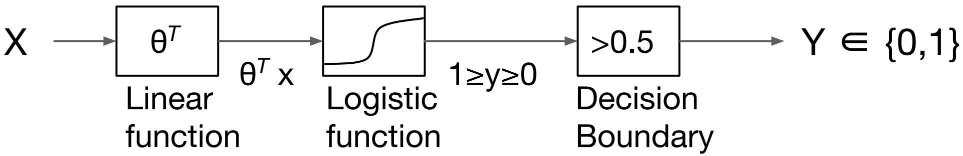
For classification hypothesis representation we do $h_\theta(x) = g((\theta^T X))$
What is g(z) Function? (linear function : $z = \theta^T X$)



Sigmoid (Logistic) function

g(z) function out value is between 0 and 1





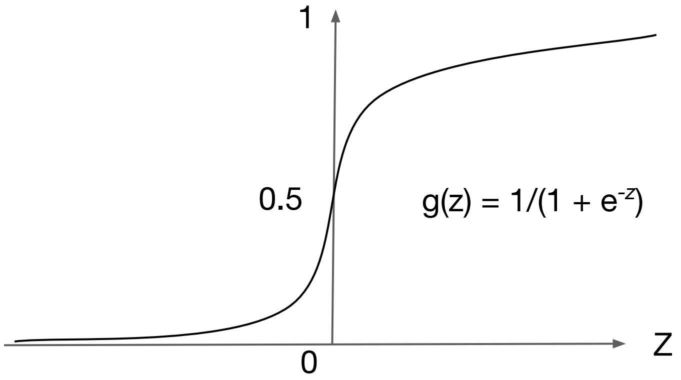
Where we define g(z) -> z is a real number -> $g(z) = e^z/(e^z + 1) = 1/(1 + e^{-z})$

[Tensorflow Code]

```
hypothesis  = tf.sigmoid(z) # z= tf.matmul(X, W) + b : θ=W
hypothesis  = tf.div(1., 1. + tf.exp(z))
```

Decision Boundary

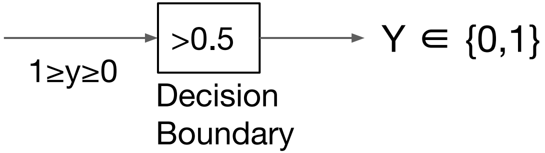
Linear / Non-linear decision boundary

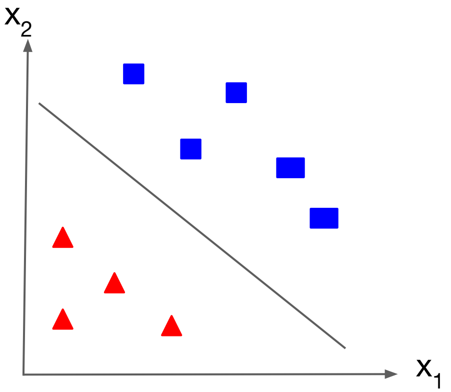


if z is positive, g(z) is greater than 0.5

θ^T x >= 0 then y = 1

θ^T x < 0 then y = 0



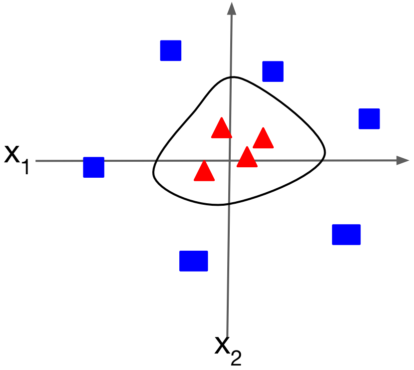


$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

θ^T is a row vector = [-3,1,1]

$-3 + x_1 + x_2 >= 0$

Decision Boundary $x_1 + x_2 = 0$



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_3 x_1^2 + \theta_4 x_2^2)$

θ^T was [-1,0,0,1,1]

$-1 + x_1^2 + x_2^2 >= 0$

Decision Boundary $x_1^2 + x_2^2 = 1$

※ θ^T is an [1 x n+1] matrix / θ_i are parameters / (h = hypothesis)

[Tensorflow Code]

```
predicted = tf.cast(hypothesis > 0.5, dtype=tf.int32)
```