Logistic(Regression)Classification

Regression

- Hypothesis (가정으로 만든 직선)

$$H(x) = Wx$$

- Cost (직선 값 − 실제 관측값) "거리"

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

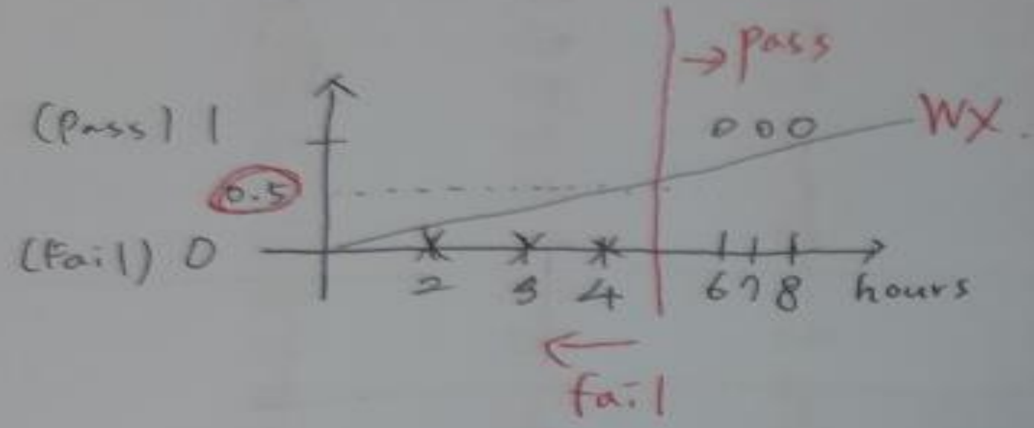
- Gradient decent
(측정해서 최소 W 찾아내기)

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

learning_rate ↳ 기울기

(binary)Classification

- 둘 중 하나를 선택하는 것
- Span Detection : Spam(1) or Ham(0)
- Facebook feed : show(1) or hide(0)
- Credit Card Fraudulent Transaction detection : legitimate(0)/fraud(1)
- Radiology, Finance



문제점) hours 커지면 합격인데
불합격으로 인식..

$H(x) = Wx + b$ ..

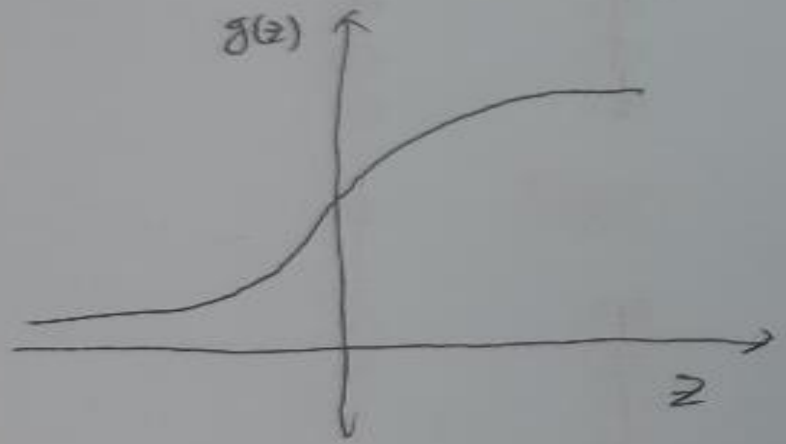
└ 0 혹은 1 외의 값 나옴..

Logistic Hypothesis

$g(z)$: 0 ~ 1 .. 중

⇒ Sigmoid : $g(z) = \frac{1}{(1+e^{-z})}$ ..

= logistic ..



① " $H(x) = g(z)$ " $= \frac{1}{1+e^{-W^T X}}$ "

z가 커지면 1로 가고
z가 작아지면 0으로 가까이 간다.

② Cost

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

↓

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

↓

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

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

(시작점에 따라 달라짐..

"Cost를 최소화

$y=1,\ H(x)=1 \rightarrow Cost(1)=0$..

$H(x)=0 \rightarrow Cost=\infty$..

$y=0,\ H(x)=0 \rightarrow Cost=0$

$H(x)=1 \rightarrow Cost=\infty$

(binary)Logistic Classification 실습

```
x_data = [[1,2], [2,3], [3,1], [4,3], [5,3], [6,2]]
y_data = [[0],[0],[0],[1],[1],[1]]

# x_data 차원
X = tf.placeholder(tf.float32, shape=[None, 2])
# y_data 차원
Y = tf.placeholder(tf.float32, shape=[None, 1])

# X_feature , Y_feature 개수
W = tf.Variable(tf.random_normal([2,1]), name='weight')
# Y_feature 개수
b = tf.Variable(tf.random_normal([1]), name='bias')
```

데이터 차원 설정

```
# H(X)
hypothesis = tf.sigmoid(tf.matmul(X, W)+b)

# Cost function
cost = -tf.reduce_mean(Y * tf.log(hypothesis) + (1-Y) * tf.log(1-hypothesis))

# Optimize
train = tf.train.GradientDescentOptimizer(learning_rate=0.01).minimize(cost)
```

```
# 예측
predicted = tf.cast(hypothesis > 0.5, dtype = tf.float32)
# 정확도
accuracy = tf.reduce_mean(tf.cast(tf.equal(predicted, Y), dtype = tf.float32))
```

Hypothesis, cost, train, 예측(0or1), 정확도 설정

(binary)Logistic Classification 실습

```
# 학습
# Launch graph
with tf.Session() as sess:
    # Initialize TensorFlow variables
    sess.run(tf.global_variables_initializer())

    for step in range(10001):
        cost_val, _ = sess.run([cost, train], feed_dict={X: x_data, Y: y_data})
        if step % 200 == 0:
            print(step, cost_val)

    # Accuracy report
    h, c, a = sess.run([hypothesis, predicted, accuracy],
                       feed_dict={X: x_data, Y: y_data})
    print("\nHypothesis: ", h, "\nCorrect (Y): ", c, "\nAccuracy: ", a)
```

Cost : 최소 값을 찾을수록 좋다

```
10000 0.13778912
```

```
Hypothesis:  [[0.02597439]
 [0.1518389 ]
 [0.28117657]
 [0.7924235 ]
 [0.9463189 ]
 [0.9824551 ]]
Correct (Y):  [[0.]
 [0.]
 [0.]
 [1.]
 [1.]
 [1.]]
Accuracy:  1.0
```

학습이 끝나면 마지막 학습 모델을 통해 hypothesis 값, hypothesis를 predict한 값(0or1), accuracy(y_data와 일치하는지 여부) 검정

결과(x1,x2 학습하여 instance 개수만큼 예측)