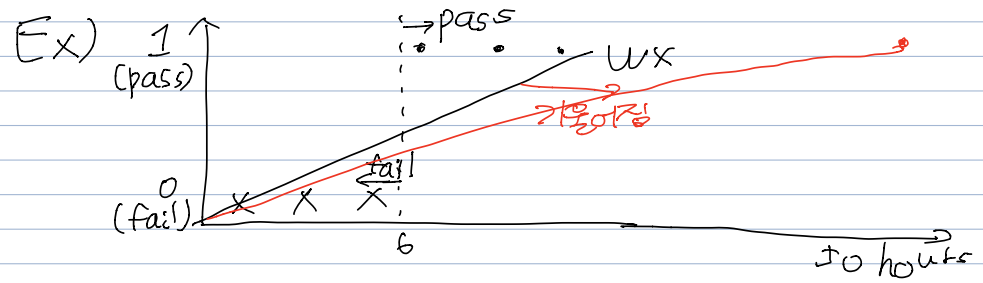
# Section 5. Logistic (Regression) Classification

## I. Logistic Classification의 가설 함수 정의

### 1. Classification

- Spam detection, facebook feed.
  Credit Card Fraudulent Transaction detection
  등에 이용됨. ⇒ 0/1 encoding 형태로 구현

Ex)




몰제점 = hypothesis는 1보다 크거나 0보다 작을 수 있음.

### 2. Logistic Hypothesis

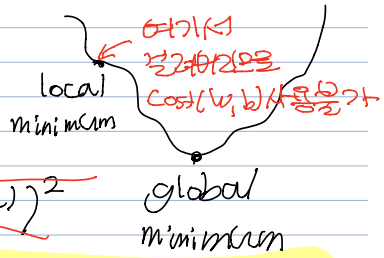
~~$H(x) = Wx + b$~~ $g(z) \rightarrow 0 \sim 1$

- Sigmoid function = S처럼 두 방향으로 꺾어짐
  = logistic function $\frac{1}{(1+e^{-z})}$

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

# II. Logistic Regression의 Cost함수

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




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

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

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

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

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

```
cost = -tf.reduce_mean(-tf.reduce_sum(Y*
    tf.log(hypothesis) + (1-Y)*tf.log(1-hypothesis)))

a = tf.Variable(0.1)
optimizer = tf.train.GradientDescentOptimizer(a)
train = optimizer.minimize(cost)
```

# III. TensorFlow로 Logistic Classification 구현

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

X = tf.placeholder(tf.float32, shape=[None, 2])
Y = tf.placeholder(tf.float32, shape=[None, 1])
W = tf.Variable(tf.random_normal([2, 1]), name='weight')
b = tf.Variable(tf.random_normal([1]), name='bias')

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

cost = -tf.reduce_mean(-tf.reduce_sum(Y *
        tf.log(hypothesis) + (1 - Y) * tf.log(1-hypothesis))

train = tf.train.GradientDescentOptimizer(learning
        rate=0.01).minimize(cost)

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

with tf.Session() as sess:
    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)
    h, c, a = sess.run([hypothesis, predicted,accuracy],
                       feed_dict={X:x_data, Y:y_data})
```

```
print ("\n Hypothesis: ", h, "\n Correct (%): ", c,
       "\n Accuracy: ", a)
```