In [0]:

```
from IPython.display import Image
from tqdm import tqdm_notebook
```

로지스틱 함수

- 로지스틱 모형 식은 독립 변수가 [-∞,∞]의 어느 숫자이든 상관 없이 종속 변수 또는 결과 값이 항상 범위 [0,1] 사이에 있도록 한다.
- 이는 오즈비(odds ratio)를 로짓(logit) 변환을 수행함으로써 얻어진다
- 함수를 sigmoid로 제공해준다.

In [0]:

```
import tensorflow as tf

tf.set_random_seed(777)

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(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.cast(tf.equal(predicted, Y), dtype=tf.float32))
```

In [0]:

```
with tf.Session() as sess :
    sess.run(tf.global_variables_initializer())

    for step in tqdm_notebook(range(10001)) :
        cost_val, _ = sess.run([cost, train], feed_dict={X: x_data, Y: y_data})
        if step % 200 == 0 or step < 10 :
            print("Step : {} Wt Cost : {}".format(step, cost_val))


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

```
Step : 0        Cost : 3.363095283508301
Step : 1        Cost : 3.3407506942749023
Step : 2        Cost : 3.318399667739868
Step : 3        Cost : 3.2960402965545654
Step : 4        Cost : 3.273649215698242
Step : 5        Cost : 3.251307725906372
Step : 6        Cost : 3.228942632675171
Step : 7        Cost : 3.206615686416626
Step : 8        Cost : 3.184260606765747
Step : 9        Cost : 3.161933183670044
Step : 200      Cost : 0.5485365986824036
Step : 400      Cost : 0.5159369111061096
Step : 600      Cost : 0.49320507049560547
Step : 800      Cost : 0.47366562485694885
Step : 1000     Cost : 0.45606350898742676
Step : 1200     Cost : 0.4397983253002167
Step : 1400     Cost : 0.42456284165382385
Step : 1600     Cost : 0.41018804907798767
Step : 1800     Cost : 0.39657315611839294
Step : 2000     Cost : 0.3836517632007599
Step : 2200     Cost : 0.37137570977211
Step : 2400     Cost : 0.35970571637153625
Step : 2600     Cost : 0.34860798716545105
Step : 2800     Cost : 0.33805158734321594
Step : 3000     Cost : 0.3280077874660492
Step : 3200     Cost : 0.3184490203857422
Step : 3400     Cost : 0.309349000453949
Step : 3600     Cost : 0.30068260431289673
Step : 3800     Cost : 0.2924259901046753
Step : 4000     Cost : 0.284556120634079
Step : 4200     Cost : 0.27705150842666626
Step : 4400     Cost : 0.2698913514614105
Step : 4600     Cost : 0.2630561888217926
Step : 4800     Cost : 0.25652769207954407
Step : 5000     Cost : 0.25028857588768005
Step : 5200     Cost : 0.2443222850561142
Step : 5400     Cost : 0.2386135309934616
Step : 5600     Cost : 0.2331480234861374
Step : 5800     Cost : 0.2279120534658432
Step : 6000     Cost : 0.22289299964904785
Step : 6200     Cost : 0.21807891130447388
Step : 6400     Cost : 0.2134586125612259
Step : 6600     Cost : 0.20902161300182343
Step : 6800     Cost : 0.20475810766220093
Step : 7000     Cost : 0.20065884292125702
Step : 7200     Cost : 0.19671528041362762
Step : 7400     Cost : 0.19291919469833374
Step : 7600     Cost : 0.18926306068897247
Step : 7800     Cost : 0.1857396960258484
Step : 8000     Cost : 0.182342529296875
Step : 8200     Cost : 0.17906510829925537
Step : 8400     Cost : 0.1759016513824463
Step : 8600     Cost : 0.17284655570983887
Step : 8800     Cost : 0.16989459097385406
Step : 9000     Cost : 0.1670408993959427
Step : 9200     Cost : 0.16428081691265106
Step : 9400     Cost : 0.16161005198955536
Step : 9600     Cost : 0.1590244024991989
Step : 9800     Cost : 0.1565200239419937
Step : 10000    Cost : 0.15409328043460846
```