```python
import tensorflow as tf

x_data = [[1,1,1], [2,2,2], [3,3,3], [4,4,4], [5,5,5], [6,6,6], [7,7,7]]
y_data = [[0], [0], [0], [1], [1], [1], [1]]
X = tf.placeholder(tf.float32, shape=[None, 3])
Y = tf.placeholder(tf.float32, shape=[None, 1])
W = tf.Variable(tf.random_normal([3,1]))
b = tf.Variable(tf.random_normal([1]))

model = tf.sigmoid(tf.add(tf.matmul(X,W),b))
cost = tf.reduce_mean((-1)*Y*tf.log(model) + (-1)*(1-Y)*tf.log(1-model))
train = tf.train.GradientDescentOptimizer(0.01).minimize(cost)

prediction = tf.cast(model > 0.5, dtype=tf.float32)
accuracy = tf.reduce_mean(tf.cast(tf.equal(prediction, Y), dtype=tf.float32))

with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    # Training
    for step in range(10001):
        cost_val, train_val = sess.run([cost, train], feed_dict={X: x_data, Y: y_data})
        print(step, cost_val)
    # Testing
    h, c, a = sess.run([model, prediction, accuracy], feed_dict={X: x_data, Y: y_data})
    print("₩nModel: ", h,"₩nCorrect: ", c, "₩nAccuracy: ", a)

```