```
import tensorflow as tf
import numpy as np
import m
             드라이브에서 찾기
                                                          력 #재현가능
tf.set_i
             실습 모드에서 열기
learnin
            새 Python 3 노트
Hyperpa
            새 Python 2 노트
#datase
                                                  Ctrl+0
            노트 열기...
x_data
            노트 업로드...
            이름 바꾸기...
y_data
             휴지통으로 이동
             드라이브에 사본 저장...
            GitHub Gist로 사본 저장...
            GitHub에 사본 저장...
                                                   Ctrl+S
            저장
x_data
                                                 Ctrl+M S
            버전 저장 및 고정
             업데이트 기록
y_data
            .ipynb 다운로드
            .py 다운로드
             드라이브 미리보기 업데이트
x data
y_data
                                                   Ctrl+P
             인쇄
x_data.shape
y_data.shape
X = tf.placeholder(tf.float32, [None, 2]) # placeholder 자리잡기
Y = tf.placeholder(tf.float32, [None, 1])
W1 = tf.Variable(tf.random_normal([2, 1]), name='weight1')
b1 = tf.Variable(tf.random_normal([1]), name='bias1')
hypothesis = tf.sigmoid(tf.matmul(X, W1) + b1)
cost = tf.reduce_mean(tf.square(hypothesis - Y)) # 평균/ 정답이랑 내 가설이랑 비교 # sum/n=mean
train = tf.train.GradientDescentOptimizer(|earning_rate=|earning_rate).minimize(|cost)
```

```
# Launch graph
sess = tf.Session()

# TensorFlow 변수들(variables) 초기화 (Initialization)
sess.run(tf.global_variables_initializer())

for i in range(10001):
    sess.run(train, feed_dict={X: x_data, Y: y_data})
    if i % 1000 ==0:
        c1 = sess.run(cost, feed_dict={X: x_data, Y: y_data})
        print('step={} / cost={}'.format(i, c1))
```

▼ 결과 확인하기

```
for i in range(4):
    x1 = x_data[[i], :]

    l1 = tf.sigmoid(tf.matmul(x1, W1) + b1)

    print( i, sess.run(|1))
    #print( i, sess.run(|2), sess.run(|2cast), y_data[[i], :])
```

• HW: 위의 코드를 변형하여 XOR 학습시 얻어진 Cost 그래프를 그리시오. Hint: List 사용

▼ 참고: Sigmoid

```
y1 = 1.0
y2 = sess.run(tf.sigmoid(y1))
print('{} --> {}'.format(y1, y2))

다

Sigmoid를 그려볼까요?

x1 = np.arange(-10,10, 0.5)
print(x1)
```

 Γ

```
for i in range(len(x1)):
    y1 = x1[i]
    y2 = sess.run(tf.sigmoid(y1))
    print('{} →> {}'.format(y1, y2))
```

```
plt.plot(x_data)
plt.plot(y_data)
```

С⇒