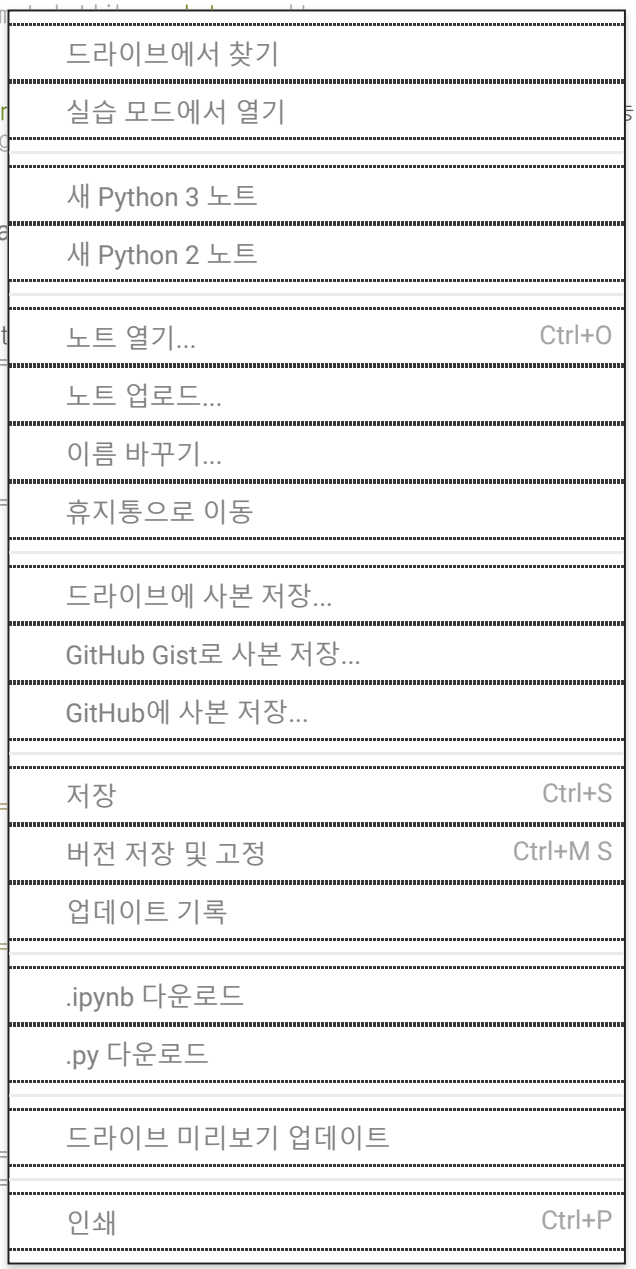
```
import tensorflow as tf
import numpy as np
import m
```




```
tf.set_r                                                    력 #재현가능
learning

Hyperpa

#dataset
x_data =

y_data =

'''
x_data =

y_data =

'''

x_data =
y_data =

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(learning_rate=learning_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(l1))
   #print( i, sess.run(l2), sess.run(l2cast), 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))
```

Sigmoid를 그려볼까요?

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

```
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)
```