

## ▼ VERY Important : XOR

import tf.nn	드라이브에서 찾기	
import tf.nn	실습 모드에서 열기	
import tf.nn		
tf.set_random_seed(1)	새 Python 3 노트	
	새 Python 2 노트	
• Help	노트 열기...	Ctrl+O
learning_rate = 0.01	노트 업로드...	
nb_epoch = 1000	이름 바꾸기...	
	휴지통으로 이동	
	드라이브에 사본 저장...	
x_data = np.zeros([1000, 2])	GitHub Gist로 사본 저장...	
	GitHub에 사본 저장...	
y_data = np.zeros([1000, 1])	저장	Ctrl+S
	버전 저장 및 고정	Ctrl+M S
	업데이트 기록	
x_data = x_data.reshape([1000, 1, 2])	.ipynb 다운로드	
y_data = y_data.reshape([1000, 1, 1])	.py 다운로드	
X = tf.placeholder(tf.float32, [None, 2])	드라이브 미리보기 업데이트	
Y = tf.placeholder(tf.float32, [None, 1])		
W1 = tf.Variable(tf.random_normal([2, 1]), name='weight1')		1') #랜덤노말_무작위, 완전무작위는 X
#연결하는 뉴런의 개수	인쇄	Ctrl+P
b1 = tf.Variable(tf.random_normal([1]), name='bias1')		
layer1 = tf.nn.bias_add(tf.matmul(X, W1), b1, name='layer1')		

```
W2 = tf.Variable(tf.random_normal([2, 1]), name='weight2')
# 2개가 들어가서 1개가 나올 #히든레이어 #앞뒤의 개수를 맞추면 늘릴수있음
b2 = tf.Variable(tf.random_normal([1]), name='bias2')
hypothesis = tf.sigmoid(tf.matmul(layer1, W2) + b2) #시그모이드 아래 그래프확인
```



```
cost = tf.reduce_mean(tf.square(hypothesis - Y))
train = tf.train.GradientDescentOptimizer(learning_rate=learning_rate).minimize(cost) #옵티마이저 등
```

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

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

```
for i in range(nb_epoch+1):
    sess.run(train, feed_dict={X: x_data, Y: y_data}) #feed, 데이터들을 가져와서 먹여줌 그후 학습

    c1 = sess.run(cost, feed_dict={X: x_data, Y: y_data})
    plt.plot(i, c1, 'b.')
```



- **짧은 텍스트 HW :** 위의 코드를 변형하여 XOR 학습시 얻어진 Cost 그래프를 그리시오. Hint : List 사용

```
print(sess.run(W1))
```



## ▼ Check the results

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

    l1 = tf.sigmoid(tf.matmul(x1, W1) + b1)
    l2 = tf.sigmoid(tf.matmul(l1, W2) + b2)
    l2cast = tf.cast(l2 > 0.5, dtype=tf.float32)
    print(i, sess.run(l2))
    print(i, sess.run(l2), sess.run(l2cast), y_data[[i], :])
```



## ▼ 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)
```



```
y1 = sess.run(tf.sigmoid(x1))
print(y1)
```



```
import matplotlib.pyplot as plt
```

```
plt.plot(x1, y1)
plt.grid()
plt.title('Sigmoid')
#자연계에서 자주보이는 곡선
#특징: 0을 넣으면 0.5를 넣음 아무리 큰, 작은 수를 넣어도 0~1안의 값이 나옴
#용도: 0~1 값이 나와야하는 문제 해결시 사용
```



