

▼ XOR 이해하기

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
```

1. AND 데이터로 바꾸어서 해보기

2. XOR 데이터로 바꾸어서 해보기

```
x_data = [[0, 0],
           [0, 1],
           [1, 0],
           [1, 1]]
```

```
y_data = [[0],
           [1],
           [1],
           [0]]
```

```
x_data = np.array(x_data, dtype=np.float32)
y_data = np.array(y_data, dtype=np.float32)
```

```
X = tf.placeholder(tf.float32, [None, 2])
Y = tf.placeholder(tf.float32, [None, 1])
```

- 안되는 네트워크 구조 예제. 되도록 고쳐봅시다.

```
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)
```



WARNING:tensorflow:From /home/seung/.env/py368keras/lib/python3.6/site-packages/tensorflow/p
Instructions for updating:
Colocations handled automatically by placer.

```
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))
```

```
sess = tf.Session()
```

```
sess.run(tf.global_variables_initializer())
```

```
sess.run(tf.global_variables_initializer())
```

```
vcost =[]
```

```
for step in range(10001):
```

```
    cost1, _ = sess.run([cost, train], feed_dict={X: x_data, Y: y_data})
```

```
    vcost.append(cost1)
```

```
    if step %1000 ==0:
```

```
        print(cost1)
```

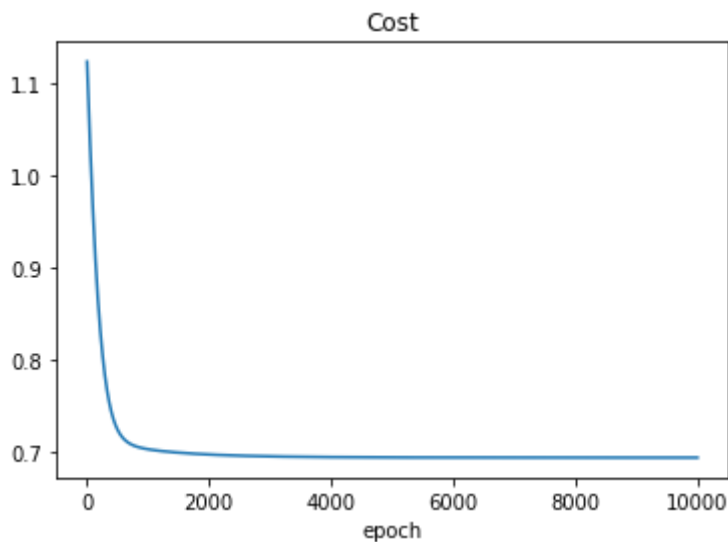
```
1.124551
0.70228124
0.69655025
0.69452083
0.6937221
0.69339466
0.6932558
0.69319546
0.6931689
0.693157
0.6931516
```

```
plt.plot(vcost)
```

```
plt.title('Cost')
```

```
plt.xlabel('epoch')
```

```
Text(0.5, 0, 'epoch')
```



```
test1 = sess.run(hypothesis, feed_dict={X:[[0.1,0]]})
```

```
print(test1[0][0])
```

```
0.49782157
```

```
test1 = sess.run(hypothesis, feed_dict={X:[[0.1,0]]})
```

