Important : AND Gate

In [1]:

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
tf.set_random_seed(777)  # for reproducibility
```

- Hyperparameters

In [2]:

```
learning_rate = 0.1
nb_epoch = 10000
```

- Dataset

In [3]:

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

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

In [4]:

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

In [5]:

```
x_data.shape
```

Out[5]:

```
(4, 2)
```

In [6]:

```
y_data.shape
```

Out[6]:

```
(4, 1)
```

In [7]:

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

In [8]:

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

In [9]:

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

In [10]:

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

In [11]:

```
for i in range(nb_epoch+1):
    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))
```

```
step=0 / cost=0.388042688369751
step=1000 / cost=0.05623956769704819
step=2000 / cost=0.0314212366938591
step=3000 / cost=0.02105650305747986
step=4000 / cost=0.015561504289507866
step=5000 / cost=0.012223777361214161
step=6000 / cost=0.010006972588598728
step=7000 / cost=0.008438924327492714
step=8000 / cost=0.007276789750903845
step=9000 / cost=0.0063840169459581375
step=10000 / cost=0.00567843159660697
```

Checking the results

In [12]:

```
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], :])
```

```
0 [[0.00084553]]
1 [[0.08142591]]
2 [[0.08142591]]
3 [[0.9027752]]
```