

```

import tensorflow as tf

import time

x_ = tf.placeholder(tf.float32, shape=[4,2], name = 'x-input')
y_ = tf.placeholder(tf.float32, shape=[4,1], name = 'y-input')

Theta1 = tf.Variable(tf.random_uniform([2,2], -1, 1), name = "Theta1")
Theta2 = tf.Variable(tf.random_uniform([2,1], -1, 1), name = "Theta2")

Bias1 = tf.Variable(tf.zeros([2]), name = "Bias1")
Bias2 = tf.Variable(tf.zeros([1]), name = "Bias2")

with tf.name_scope("layer2") as scope:
    A2 = tf.sigmoid(tf.matmul(x_, Theta1) + Bias1)

with tf.name_scope("layer3") as scope:
    Hypothesis = tf.sigmoid(tf.matmul(A2, Theta2) + Bias2)

with tf.name_scope("cost") as scope:
    cost = tf.reduce_mean((( y_ * tf.log(Hypothesis)) +
        ((1 - y_) * tf.log(1.0 - Hypothesis)) ) * -1)

with tf.name_scope("train") as scope:
    train_step = tf.train.GradientDescentOptimizer(0.01).minimize(cost)

XOR_X = [[0,0],[0,1],[1,0],[1,1]]
XOR_Y = [[0],[1],[1],[0]]

init = tf.initialize_all_variables()

sess = tf.Session()

writer = tf.train.SummaryWriter("./logs/xor_logs", sess.graph_def)

```

```
sess.run(init)
```

```
t_start = time.clock()
```

```
for i in range(100000):
```

```
    sess.run(train_step, feed_dict={x_: XOR_X, y_: XOR_Y})
```

```
    if i % 1000 == 0:
```

```
        print('Epoch ', i)
```

```
        print('Hypothesis ', sess.run(Hypothesis, feed_dict={x_: XOR_X, y_: XOR_Y}))
```

```
        print('Theta1 ', sess.run(Theta1))
```

```
        print('Bias1 ', sess.run(Bias1))
```

```
        print('Theta2 ', sess.run(Theta2))
```

```
        print('Bias2 ', sess.run(Bias2))
```

```
        print('cost ', sess.run(cost, feed_dict={x_: XOR_X, y_: XOR_Y}))
```

```
t_end = time.clock()
```

```
print('Elapsed time ', t_end - t_start)
```

https://github.com/StephenOman/TensorFlowExamples/blob/master/xor%20nn/xor_nn.py