Python实验报告

实验十二

学号：2016326603039 姓名：郭 仲

学号：2016326603029 姓名：黄欣悦

学号：2016326603071 姓名：王明凯

学号：2016329600130 姓名：王宇舟

学号：2016326603046 姓名：余泓锷

实验题

实验题1：请你修改损失函数，再运行本实例，看看运行结果差异。多尝试一下，思考出现差异的原因。

In [16]:

```
%matplotlib notebook

import matplotlib.pyplot as plt
import tensorflow as tf
import tensorflow.contrib.learn as skflow
from sklearn.utils import shuffle
import numpy as np
import pandas as pd

#载入数据
df = pd.read_csv("data/boston.csv", header=0)
x_data = df[['CRIM', 'DIS', 'LSTAT']].values.astype(float) #选取其中3个比较重要的影响因素
y_data = df['MEDV'].values.astype(float) #获取y
```

In [17]:

```
#构建模型
x = tf.placeholder(tf.float32, [None,3], name = "x") # 3个影响因素
y = tf.placeholder(tf.float32, [None,1], name = "y")
with tf.name_scope("Model"):
    w = tf.Variable(tf.random_normal([3,1], stddev=0.01), name="w0")
    b = tf.Variable(1., name="b0")
    def model(x, w, b):
        return tf.matmul(x, w) + b

    pred= model(x, w, b)
```

In [18]:

```
#训练模型
#设置参数
train_epochs = 50 # 迭代次数
learning_rate = 0.01 #学习率
#定义损失函数
with tf.name_scope("LossFunction"):
    loss_function = tf.reduce_sum(tf.where(tf.greater(pred, y), (pred - y) * 10, (y - pred) * 1))
```

In [19]:

```
optimizer = tf.train.AdamOptimizer(learning_rate).minimize(loss_function)
```

In [20]:

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

In [21]:

```
tf.train.write_graph(sess.graph, 'log/boston','graph.pbtxt')
loss_op = tf.summary.scalar("loss", loss_function)
merged = tf.summary.merge_all()
```

In [22]:

```
#执行训练
sess.run(init)
```

In [23]:

```
writer = tf.summary.FileWriter('log/boston', sess.graph)
```

In [24]:

```
for epoch in range (train_epochs):
    lossv=0.0
    for xs, ys in zip(x_data, y_data):
        z1 = xs.reshape(1,3)
        z2 = ys.reshape(1,1)
        sess.run(optimizer, feed_dict={x: z1, y: z2})
        summary_str = sess.run(loss_op, feed_dict={x: z1, y: z2})
        lossv+=sess.run(loss_function, feed_dict={x: z1, y: z2})/506.00
        writer.add_summary(summary_str, epoch)
    xvalues, yvalues = shuffle(x_data, y_data)
    print (lossv)
    b0temp=b.eval(session=sess)
    w0temp=w.eval(session=sess)
    print("epoch=", epoch+1,"b=", b0temp,"w=", w0temp )
```

```
16.043088034678824
epoch= 1 b= 2.0499802 w= [[0.05825005]
 [1.3879642 ]
 [0.10720979]]
12.573579528115026
epoch= 2 b= 2.8646004 w= [[-0.01352349]
 [ 1.7239873 ]
 [ 0.23844145]]
12.178064645985856
epoch= 3 b= 3.42165 w= [[0.06998962]
 [1.693594  ]
 [0.10845822]]
12.000740554021732
epoch= 4 b= 4.057356 w= [[-0.02947857]
 [ 1.70892   ]
 [ 0.11640272]]
11.820395044658502
epoch= 5 b= 4.646777 w= [[-0.10184665]
 [ 1.6532251 ]
```

In [25]:

```
print("y=",w0temp[0], "x1+",w0temp[1], "x2+",w0temp[2], "x3+", [b0temp])
print("y=",w0temp[0], "CRIM+", w0temp[1], 'DIS+', w0temp[2], "LSTAT+", [b0temp])
```

```
y= [-0.06743419] x1+ [0.07623857] x2+ [-0.48196653] x3+ [21.210735]
y= [-0.06743419] CRIM+ [0.07623857] DIS+ [-0.48196653] LSTAT+ [21.210735]
```

出现差异的原因：

损失函数是用来估量模型的预测值pred与真实值y的不一致程度，损失函数越小，模型的鲁棒性就越好，所以争对不同的模型，使用不同的损失函数运行结果会有所不同。