**第七周作业**

**姓名：李英**

**学号：2017051611007**

**最终代码：**

import tensorflow as tf

import matplotlib.pyplot as plt

import numpy as np

bsdh = tf.keras.datasets.boston\_housing

(x,y),(test\_x,test\_y) = bsdh.load\_data()

x = x.astype(np.float32)

y = y.astype(np.float32)

for i in range(13):

    x[:,i] = (x[:,i]-np.min(x[:,i]))/(np.max(x[:,i])-np.min(x[:,i]))

def model(x,w,b):

    return tf.matmul(x,w) + b

w = tf.Variable(tf.random.normal([13,1],mean=0,stddev=1.0,dtype=tf.float32))

b = tf.Variable(tf.zeros(1),dtype=tf.float32)

training\_epochs = 50 #迭代次数

learning\_rate = 0.001 #学习率

batch\_size = 10 #批量训练一次的样本数

def loss(x,y,w,b):

    err = model(x,w,b) - y

    squared\_err = tf.square(err)

    return tf.reduce\_mean(squared\_err)

def grad(x,y,w,b):

    with tf.GradientTape() as tape:

        loss\_ = loss(x,y,w,b)

    return tape.gradient(loss\_,[w,b])

optimizer = tf.keras.optimizers.SGD(learning\_rate)

loss\_list = []

total\_step = int(len(y)/batch\_size)

x1 = x.astype(np.float32)

print(tf.matmul(x1,w))

for epoch in range(training\_epochs):

    for step in range(total\_step):

        xs = x[step\*batch\_size:(step+1)\*batch\_size,:]

        ys = y[step\*batch\_size:(step+1)\*batch\_size]

        grads = grad(xs,ys,w,b)

        optimizer.apply\_gradients(zip(grads,[w,b]))

    loss\_train = loss(x,y,w,b).numpy()

    loss\_list.append(loss\_train)

    print("epoch={:3d},train\_loss={:.4f}".format(epoch+1,loss\_train))

print("w:",w)

print("b:",b)

plt.figure()

plt.rcParams["font.sans-serif"]="SimHei"

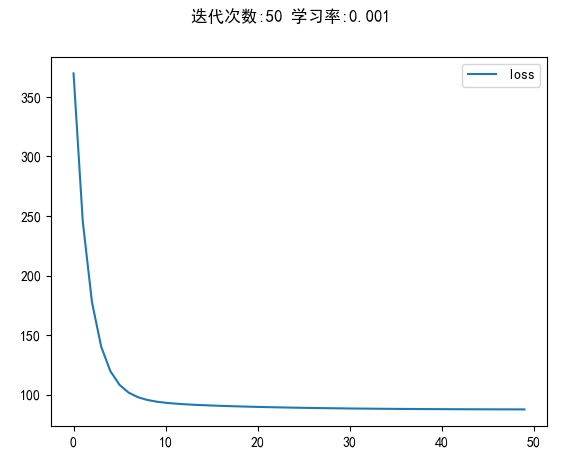
plt.plot(loss\_list,label="loss")

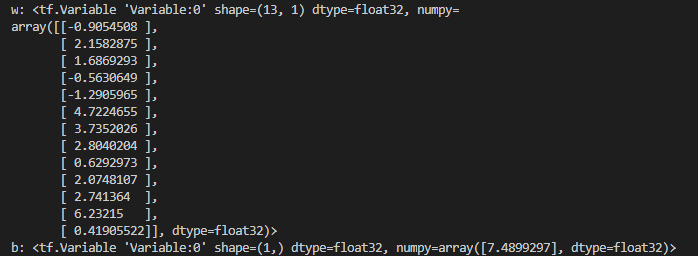
plt.legend()

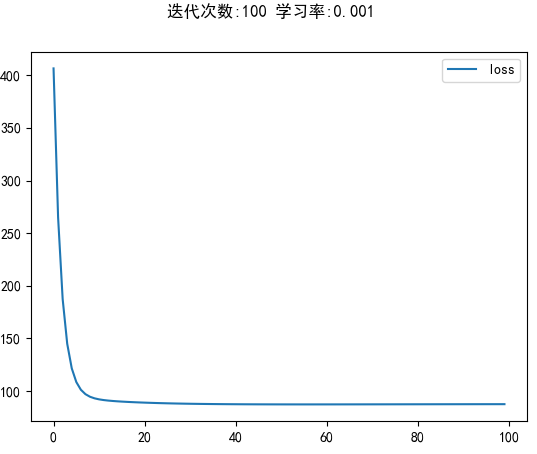
plt.suptitle("迭代次数:"+str(training\_epochs)+" 学习率:"+str(learning\_rate))

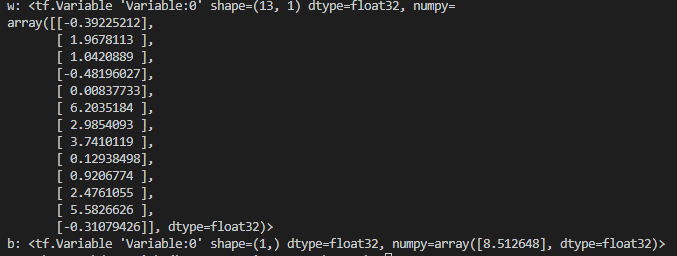
plt.show()

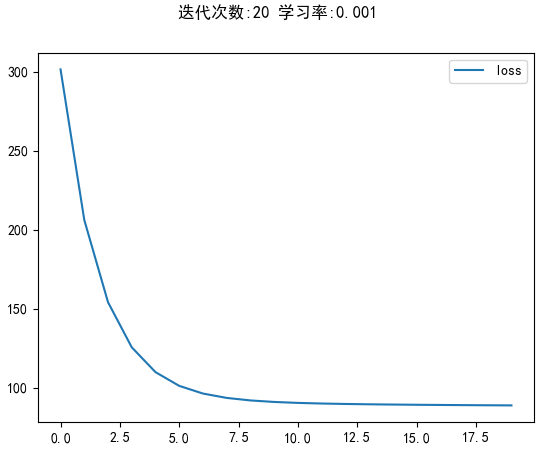
**不同超参数分别运行结果：**

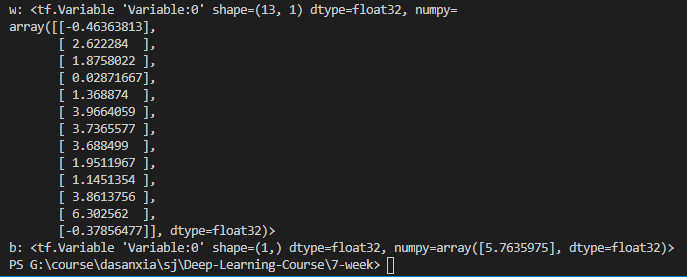


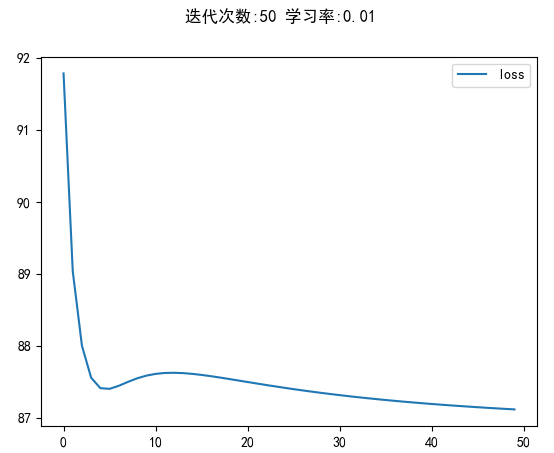


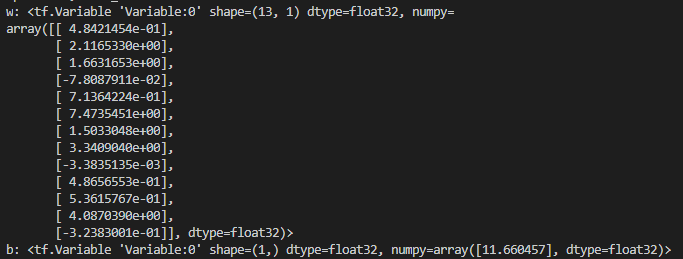


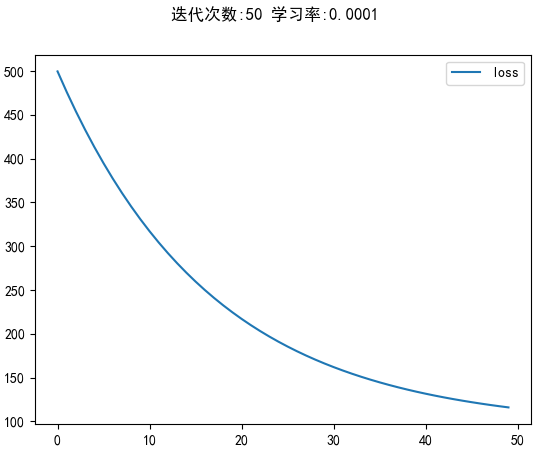


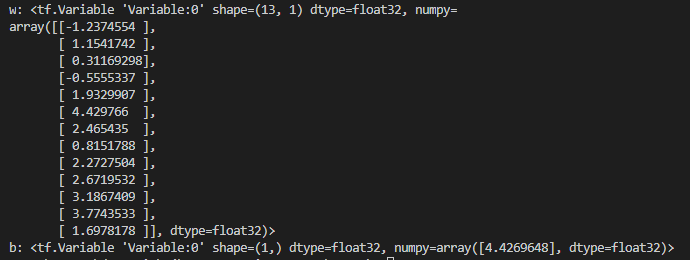












**GitHub地址：**[**https://github.com/ying1024/Deep-Learning-Course.git**](https://github.com/ying1024/Deep-Learning-Course.git)