

EECS 545 Homework 5

Yuan Yin

November 28, 2018

Problem 1) Nerual Network

a.

MSE for training data is: 22.92454069780808

b.

MSE for test data is: 26.7929824

c.

The code is as follows:

```
import numpy as np
import scipy.io as sio

def activefunc_prime( input ):
    return (input > 0) * np.sign(input)
# Load the data and preprocessed the data
np.random.seed(0)
bodyfat_data = sio.loadmat('bodyfat_data.mat')
x = bodyfat_data['X']
y = bodyfat_data['y']
num,d = x.shape
y = y.reshape((1, num))
b_prime = np.ones(num)
x = np.insert(x, 0, values=b_prime, axis=1)
train_num = 150; test_num = num - 150
b = np.ones(train_num)
```

```

x_train = x[:train_num,:]
x_test = x[train_num:,:]
y_train = y[0, :train_num].reshape((1, train_num))
y_test = y[0, train_num:].reshape((1, test_num))

# Initialization
l1 = 64; l2 = 16
w1 = np.random.normal(0, 1, l1 * d).reshape((d, l1)).T
w2 = np.random.normal(0, 1, l1 * l2).reshape((l1, l2)).T
w3 = np.random.normal(0, 1, l2).reshape((1, l2))
b1 = np.zeros(l1); b2 = np.zeros(l2); b3 = 0
theta1 = np.insert(w1, 0, values=b1, axis=1)
theta2 = np.insert(w2, 0, values=b2, axis=1)
theta3 = np.insert(w3, 0, values=b3, axis=1)
delta1 = np.zeros((train_num, l1))
gamma = 10**(-7)
a1 = np.zeros((l1, train_num)); a2 = np.zeros((l2, train_num)); y_result = np.zeros((1,
    train_num))

# Neural Network
for iter in range(5000):
    ## forward pass
    for n in range(train_num):
        a1[:, n] = np.dot(theta1, x[n, :].T)
        z1 = np.maximum(a1, 0)
        z1 = np.insert(z1, 0, values=b, axis=0)
        for n in range(train_num):
            a2[:, n] = np.dot(theta2, z1[:, n].T)
            z2 = np.maximum(a2, 0)
            z2 = np.insert(z2, 0, values=b, axis=0)
            for n in range(train_num):
                y_result[0, n] = np.dot(theta3, z2[:, n].T)

    # backward pass
    delta3 = -2 * (y_train - y_result).T
    gradient3 = np.dot(z2, delta3)
    theta3 = theta3 - gamma * gradient3.T / train_num
    delta2 = np.multiply(np.dot(delta3, theta3[0, 1:].reshape((1, l2))),
        activefunc_prime(a2).T)
    gradient2 = np.dot(z1, delta2)
    theta2 = theta2 - gamma * gradient2.T / train_num
    delta1 = np.multiply(np.dot(delta2, theta2[:, 1:]), activefunc_prime(a1).T)
    gradient1 = np.dot(delta1.T, x_train)
    theta1 = theta1 - gamma * gradient1 / train_num

```

```

MSE = np.dot(y_result - y_train, (y_result - y_train).T) / train_num
iter += 1

print("MSE for training data is: ", MSE[0,0])

a11 = np.zeros((l1, test_num)); a22 = np.zeros((l2, test_num)); y_result_test = np.zeros((1,
test_num))

## test error
bb = np.ones(test_num)
for n in range(test_num):
    a11[:, n] = np.dot(theta1, x_test[n, :].T)
z11 = np.maximum(a11, 0)
z11 = np.insert(z11, 0, values=bb, axis=0)
for n in range(test_num):
    a22[:, n] = np.dot(theta2, z11[:, n].T)
z22 = np.maximum(a22, 0)
z22 = np.insert(z22, 0, values=bb, axis=0)
for n in range(test_num):
    y_result_test[0, n] = np.dot(theta3, z22[:, n].T)

MSE_test = np.dot(y_result_test - y_test, (y_result_test - y_test).T) / test_num

print("MSE for test data is: ", MSE_test[0,0])

```
