HOMEWORK1 Report

Full name: RONG ZHEN zID: z5225226

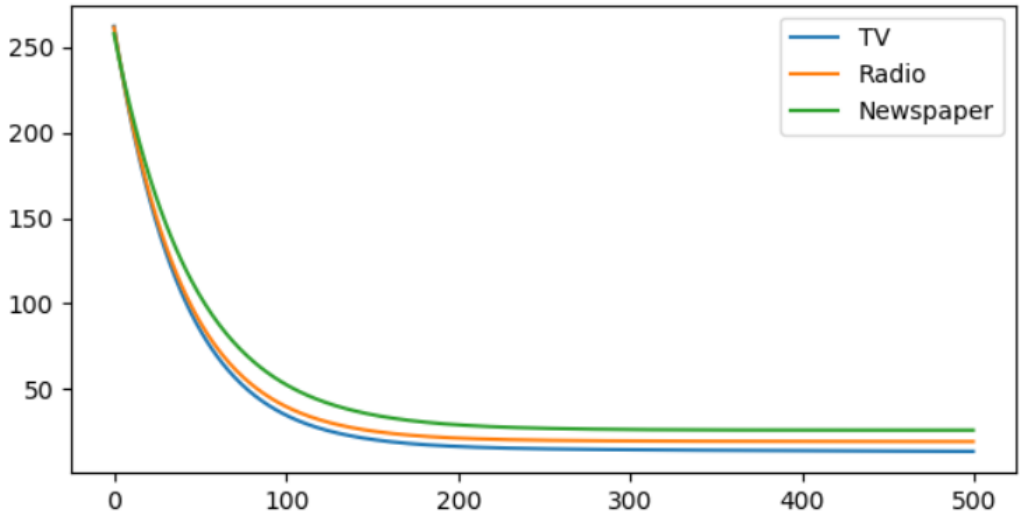
1. **You have to report the θ parameters in step 3 when you are using TV feature.**

   Answer: θ_0 = 10.11283406777275

   θ_1 = 8.271831294479673

2. **A plot, which visualizing the change in cost function J(θ) at each iteration.**

   Answer:



3. **RMSE for your training set when you use TV feature.**

   Answer: RMSE(TV training set) = 3.640 (3 digits after the decimal point)

4. **RMSE for test set, when you use TV feature.**

   Answer: RMSE(TV testing set) = 3.909 (3 digits after the decimal point)

5. **RMSE for test set, when you use Radio feature.**

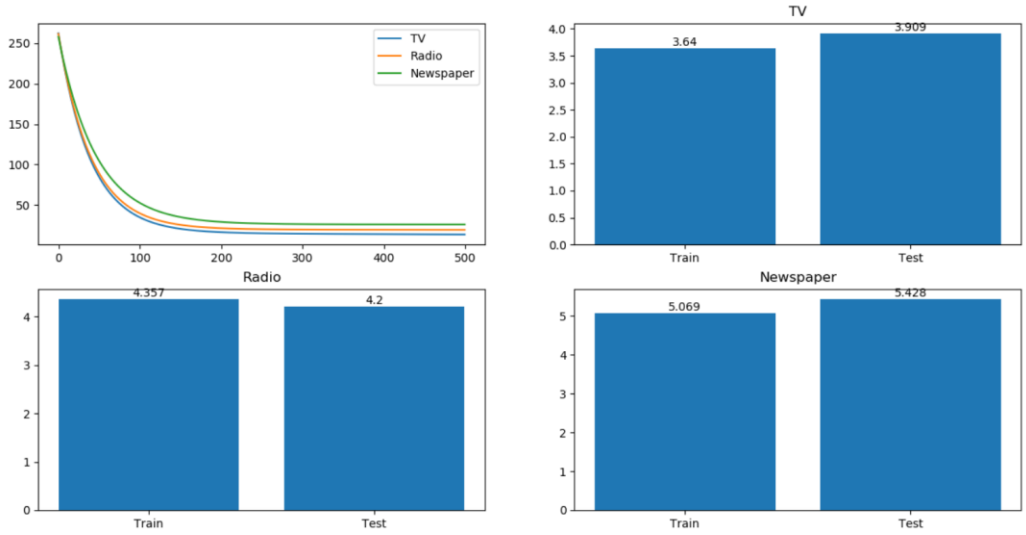
   Answer: RMSE(Radio test set) = 4.200 (3 digits after the decimal point)

6. **RMSE for test set, when you use newspaper feature.**

Answer: RMSE(Newspaper test set) = 5.428 (3 digits after the decimal point)

7. **Compare the performance of your three models and rank them accordingly.**

Answer:



I compared three models using the graph above. The RMSE of TV is the smallest.

Rank: (with loss) TV Radio Newspaper

(with RMSE) TV Radio Newspaper

Code:

```
1. import csv
2. import math
3. import matplotlib.pyplot as plt
4. x1 = []
5. x2 = []
6. x3 = []
7. y = []
8. with open('Advertising.csv','r') as csvfile:
```

```
    reader = csv.DictReader(csvfile)
    for row in reader:
        x1.append(float(row['TV']))
        x2.append(float(row['Radio']))
        x3.append(float(row['Newspaper']))
        y.append(float(row['Sales']))

 #pre-processing
def normalisation(xlist):
     xmin = min(xlist)
     xmax = max(xlist)
     for i in range(len(xlist)):
         xlist[i] = (xlist[i] - xmin)/(xmax - xmin)
     return xlist

x1 = normalisation(x1)
x2 = normalisation(x2)
x3 = normalisation(x3)

#creating test and training set
training_list = [x1[:190],x2[:190],x3[:190],y[:190]]
test_list = [x1[190:201],x2[190:201],x3[190:201],y[190:201]]

#Gradient descent
def gradientDescent(theta0,theta1,xlist,ylist):
    x0 = 1
    alpha = 0.01
    num = 0
    J_theta = []
    while(num<500):
        new_list0 = []
        new_list1 = []
        new_list2 = []
        h_x = []
        for i in xlist:
            h_x.append(theta0 * x0 + theta1 * i)
        for j in range(len(xlist)):
            new_list0.append((ylist[j] - h_x[j]))
            new_list1.append((ylist[j] - h_x[j])*xlist[j])
            new_list2.append((ylist[j] - h_x[j])**2)
        theta0 = theta0 + (sum(new_list0)/190)*alpha
        theta1 = theta1 + (sum(new_list1)/190)*alpha
        J_theta.append(sum(new_list2)/190)
        num = num + 1
```

```
    return theta0,theta1,J_theta

theta0 = -1
theta1 = -0.5
theta0_0,theta0_1,graph1 = gradientDescent(theta0,theta1,training_list[0],training_list[3])
theta1_0,theta1_1,graph2 = gradientDescent(theta0,theta1,training_list[1],training_list[3])
theta2_0,theta2_1,graph3 = gradientDescent(theta0,theta1,training_list[2],training_list[3])

#visualization
plt.figure()
plt.subplot(2,2,1)
plt.plot(graph1,label = 'TV')
plt.plot(graph2,label = 'Radio')
plt.plot(graph3,label = 'Newspaper')
plot.legend()
#plt.show()

#evaluation
def RMSE(xlist,ylist,theta0,theta1):
    h_theta = []
    L = []
    for i in xlist:
        h_theta.append(theta0 + theta1 * i)
    #print(h_theta)
    length = len(xlist)
    for j in range(length):
        L.append((ylist[j] - h_theta[j])**2)
    rmse = round(math.sqrt(sum(L)/length),3)
    return rmse

rmse0 = [RMSE(training_list[0],training_list[3],theta0_0,theta0_1),RMSE(test_list[0],test_list[3],theta0_0,theta0_1)]
rmse1 = [RMSE(training_list[1],training_list[3],theta1_0,theta1_1),RMSE(test_list[1],test_list[3],theta1_0,theta1_1)]
rmse2 = [RMSE(training_list[2],training_list[3],theta2_0,theta2_1),RMSE(test_list[2],test_list[3],theta2_0,theta2_1)]

print(rmse0)
print(rmse1)
print(rmse2)
```

```
namelist = ['Train','Test']
a = plt.subplot(222)
a.set_title('TV')
plt.bar(x = namelist,height = rmse0)
for x, y in enumerate(rmse0):
    plt.text(x, y, '%s' % y, ha='center', va='bottom')
b = plt.subplot(223)
b.set_title('Radio')
plt.bar(x = namelist,height = rmse1)
for x, y in enumerate(rmse1):
    plt.text(x, y, '%s' % y, ha='center', va='bottom')
c = plt.subplot(224)
c.set_title('Newspaper')
plt.bar(x = namelist,height = rmse2)
for x, y in enumerate(rmse2):
    plt.text(x, y, '%s' % y, ha='center', va='bottom')
plt.show()
```