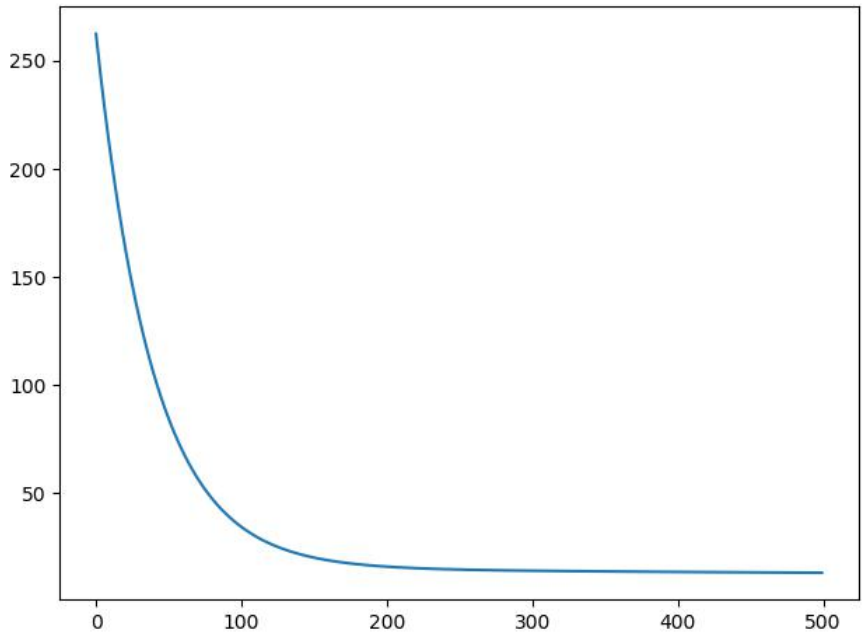
zID: z5195715
Name: Junyu Ren

COMP 9417 Homework 1

1. *You have to report the theta parameters in step 3 when you are using TV feature.*
   TV feature theta0 is approximately 10.113
   TV feature theta1 is approximately 8.272

2. *A plot, which visualises the change in cost function at each iteration.*



3.RMSE for your training set when you use TV feature.
RMSE for training set of TV feature is approximately 3.640

4.RMSE for test set, when you use TV feature.
RMSE for test set of TV feature is approximately 3.909

5. *RMSE for test set, when you use Radio feature.*
RMSE for test set of Radio feature is approximately 4.200

6. RMSE for test set, when you use newspaper feature.
RMSE for test set of newspaper feature is approximately 5.428

7. *Compare the performance of your three models and rank them accordingly.*
   Since RMSE_TV_test < RMSE_Radio_test < RMSE_Newspaper_test
   Rank: 1.TV
   2.Radio
   3.Newspaper

Code:

```
import matplotlib.pyplot as plt
import numpy as np
import math as m
import csv


def read_csv(filename):
    data = []
    with open(filename) as file:
        csv_file = csv.reader(file)
        for row in csv_file:
            data.append(row)
    return np.array(data[1:]).astype(float)


def pre_processing(data):
    for axis_num in range(1, 4):
        max_num = data[:, axis_num].max()
        min_num = data[:, axis_num].min()
        for i in range(len(data[:, axis_num])):
            data[i, axis_num] = (data[i,
axis_num]-min_num)/(max_num-min_num)
    return data


def train(train_data, axis, theta0, theta1, learning_rate, max_iteration):
    train_x = train_data[:, axis]
    train_y = train_data[:, 4]
    loss_record = []
    for i in range(max_iteration):
        y_pred = theta0 * 1 + theta1 * train_x
        error = train_y - y_pred
        loss = np.mean(error * error)
        loss_record.append(loss)
        delta_theta0 = np.mean(error * 1)
        delta_theta1 = np.mean(error * train_x)
        theta0 = theta0 + learning_rate * delta_theta0
        theta1 = theta1 + learning_rate * delta_theta1
    return theta0, theta1, loss_record
```

```
def evaluation(test_data, axis, theta0, theta1):
        test_x = test_data[:, axis]
        test_y = test_data[:, 4]
        y_pred = theta0 * 1 + theta1 * test_x
        rmse = m.sqrt(np.mean((test_y-y_pred)**2))
        return rmse


data_raw = read_csv('Advertising.csv')
data_preprocessed = pre_processing(data_raw)
data_train = data_preprocessed[:190, :]
data_test = data_preprocessed[190:, :]

theta0 = -1
theta1 = -0.5
learning_rate = 0.01
max_iteration = 500
TV = 1
Radio = 2
Newspaper = 3

theta0_TV, theta1_TV, loss_TV = train(data_train, TV, theta0, theta1, learning_rate,
max_iteration)
print("TV theta0: {}, theta1: {}".format(theta0_TV, theta1_TV))
plt.plot(loss_TV)
plt.show()

rmse_TV_train = evaluation(data_train, TV, theta0_TV, theta1_TV)
rmse_TV_test = evaluation(data_test, TV, theta0_TV, theta1_TV)
print("RMSE TV train: "+str(rmse_TV_train))
print("RMSE TV test: "+str(rmse_TV_test))

theta0_Rdo, theta1_Rdo, loss_Rdo = train(data_train, Radio, theta0, theta1,
learning_rate, max_iteration)
print("Radio theta0: {}, theta1: {}".format(theta0_Rdo, theta1_Rdo))
rmse_Rdo_test = evaluation(data_test, Radio, theta0_Rdo, theta1_Rdo)
print("RMSE Radio test: "+str(rmse_Rdo_test))

theta0_Newspr, theta1_Newspr, loss_Newspr = train(data_train, Newspaper, theta0,
theta1, learning_rate, max_iteration)
print("Newspaper theta0: {}, theta1: {}".format(theta0_Newspr, theta1_Newspr))
rmse_Newspr_test = evaluation(data_test, Newspaper, theta0_Newspr,
theta1_Newspr)
```

```
print("RMSE Newspaper test: "+str(rmse_Newspr_test))
```