Nathon Tadeo
Student ID: 801265462
Homework 1
https://github.com/nathon-tadeo/Intro-to-ML/blob/main/homework_1_intro_to_ml.ipynb

Problem 1 (40 points)

Develop a code that run linear regression with gradient decent algorithm for each of the explanatory variables in isolation. In this case, you assume that in each iteration, only one explanatory variable (either X1, or X2, or X3) is explaining the output. Basically, you need to do three different training, one per each explanatory variable. For the learning rate, explore different values between 0.1 and 0.01 (your choice). Initialize your parameters to zero (theta to zero).

1. Report the linear model you found for each explanatory variable.

Using the linear regression example from class, the data was from D3 and each explanatory variable was separated. I followed the model from class and made functions to compute the gradient descent with the given data. Since each explanatory variable was in isolation I had the functions go through a loop, where each variable was reshaped, and stacked horizontally with the bias term. Linear regression was applied to each one and their subsequent model was calculated with the gradient descent function. The following are the linear models I found for each variable:

Linear model for x1: Y = 5.928 -2.0383*x1

Linear model for x2: Y = 0.736 + 0.558*x2

Linear model for x3: Y= 2.871 -0.520*x3

2. Plot the final regression model and loss over the iteration per each explanatory variable.

After the linear models were found, the corresponding explanatory variables were plotted with a plot regression function. This function first had the data of the variable form a scatter plot, then the linear regression model was plotted. This would form a plot containing the scatter points as well as the best linear regression line. Subsequently the function also plotted the loss over the iteration, or convergence of gradience decent. This plot took into account the cost history value obtained from the gradient decent function and plotted the Cost(J) with the number of iterations. For my model I used a learning rate of 0.05 and showcased 1500 iterations. All plots are contained in “Tadeo_Nathon 801265462 (2)Homework1” pdf.

3. Which explanatory variable has the lower loss (cost) for explaining the output (Y)?

According to the plotted convergence obtained from the above methods, it seems that x1 has the lower lost cost because it converges to the Cost(J) of 1 while x2, and x3 converge to 3.6.

4. Based on your training observations, describe the impact of the different learning rates on the final loss and number of training iteration.

I like the balance of cost and number of iterations with the learning rate of 0.05. Exploring the range of 0.01 to 0.1 learning rate, I came to discover that the higher the learning rate, the less number of iterations it takes to converge, but is less stable and could easily overshoot the minimum. Lower learning rate takes much more iterations to converge to the least cost.

Problem 2 (60 points)

This time, run linear regression with gradient descent algorithm using all three explanatory variables. For the learning rate, explore different values between 0.1 and 0.01 (your choice). Initialize your parameters (theta to zero).

1. Report the final linear model you found the best.

Using the same model from problem 1, I found the linear model using all three explanatory variables. This was achieved by reshaping each variable and horizontally stacking them together with a bias term. Theta to zeros was initialized with four parameters, then the gradient decent function was run in order to find the corresponding thetas. The following model was found using this method:
Y = 5.314 + -2.004*x1 + 0.533*x2 -0.266*x3

2. Plot loss over the iteration.

The plot for loss over the iteration was plotted similarly to Problem 1s method. I found the learning rate of 0.01 to be the most pleasant in terms of stability. Plot can be seen in "Tadeo_Nathon 801265462 (2)Homework1."

3. Based on your training observations, describe the impact of the different learning rates on the final loss and number of training iteration.

Learning rate for this problem follows the same traits as problem 1. The higher the learning rate, the less iterations to converge, but it also increases the risk of overshooting the minimum solution. If the learning rate is too small, convergence will be slow, but it is more likely to lead to a stable solution.

4. Predict the value of y for new (X1, X2, X3) values (1, 1, 1), for (2, 0, 4), and for (3, 2, 1)

Given the new values for X1, X2, and X3, they were put into an array along with the bias term. The predictions were calculated using the new_data.dot(theta) function. This function computes the dot product between the new array and the parameter vector theta. Using this method, the predictions came out to be [2.94136408 0.09179843 -0.01275555].