Laboratory 2

Multivariable Linear Regression

Due Date: Beginning of Week 4 Lab

Concepts:

- Simple Linear regression
- Multivariable Linear Regression

Total Points: 100 points

Objectives:

The gradient descent is a basic technique to estimate linear discriminant functions. You will use the gradient descent method to implement linear regression.

Files Needed:

- Lab2_Part1.mlx
- Lab2_Part2.mlx
- car_mpg.csv
- car_mpg_testingData.csv

Assignment:

Part I: Simple Regression for Automobile mpg Data

You will implement linear regression with one variable to predict mpg for vehicles using the automotive dataset. The data is originally collected to build the model to predict city-cycle fuel consumption in miles per gallon using multiple variables. In Part I, you will build a simple linear model which concerns city-cycle fuel consumption in miles per gallon with one variable.

As we learned in the class, the gradient descent algorithm is a method of updating weights, $\boldsymbol{w_0}$ and $\boldsymbol{w_1}$ to reduce the mean squared error cost function. In this lab exercise, you will implement following steps:

- Load data from a file (text file or csv file).
- Create a scatter plot of data for visualization
- Data normalization
- Fit a simple linear model using the gradient descent method
- Evaluate the linear regression model –mean square error calculation

- Plot the linear fit with the test data

1. **Load the data from a file**
   - The file "car_mpg.csv" contains the dataset for the mpg linear regression problem. Since the file is a CSV file (comma-separated-values), you can use Matlab ***load()*** function. The data file contains attributes as below:

     1. mpg: continuous
     2. cylinders: multi-valued discrete
     3. displacement: continuous
     4. horsepower: continuous
     5. weight: continuous
     6. acceleration: continuous
     7. model year: multi-valued discrete
     8. origin: multi-valued discrete

   - In this simple regression model, the 4th column (horse power) is used as input and the first column (mpg) is used as output.
   - Complete the lines of code for data loading and feature extraction in the section Part1-1 in the provided file, "Lab2_Part1.mlx".

2. **Visualize the data**
   - Before starting on any prediction model, it is often useful to understand the data by plotting it. Using the **plot()** function, plots the vector Y(mpg) versus the vector X(horse power) .
   - To learn more about the plot command, you can type help plot at the MATLAB command prompt or to search online for plotting documentation.
   - Complete the lines of code in section Part1-2.

3. **Data normalization**
   - In machine learning, data normalization is a procedure that **makes input features are in the similar ranges.**
   - Another reason why the data normalization is applied is that gradient descent converges much faster with feature scaling than without feature scaling.
   - Complete the lines of code for data normalization in section Part1-3 for the given input feature

4. **Add Bias term**
   - The "**bias term**" or "intercept parameter" allows us to move the linear model along the y-axis. It is **common practice to put the bias term of the linear regression** in the input vector.
   - Complete the lines of code in section Part1-4.

5. **Find parameters using gradient descent**

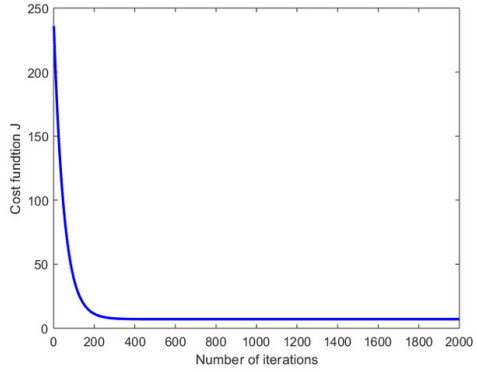
You will fit the linear regression parameters, $\boldsymbol{w_0}$ and $\boldsymbol{w_1}$ for the provided dataset using the gradient descent method. The objective of the training is to find the parameters, $\boldsymbol{w_0}$ and $\boldsymbol{w_1}$ that minimize the cost function:

$$\underset{w_0, w_1}{\text{Minimize}} \quad J(w_0, w_1) = \frac{1}{2m} \sum_{i=1}^{m} ((\boldsymbol{w_0} + \boldsymbol{w_1 x^i}) - \boldsymbol{y^i})^2$$

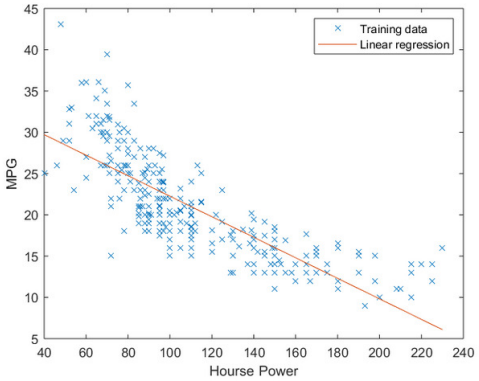
- Complete the lines of code to implement the gradient descent method in section Part1-5.
- At the end of the for loop iteration, you have the final MSE error and the final parameters, $\boldsymbol{w_0}$ and $\boldsymbol{w_1}$.

- **Question 1:** what are the final MSE error and the final parameter values for $\boldsymbol{w_0}$ and $\boldsymbol{w_1}$?

6. **Plot the MSE cost convergence graph.** Generate a plot that shows how the MSE cost changes over the iterations. The plot of the MSE cost versus the iteration looks like below:



7. **Plot the linear regression model over the data.** Use the final weights, $\boldsymbol{w_0}$ and $\boldsymbol{w_1}$ to plot the simple linear regression model over the training data. The plot should be similar with the below figure:



To plot the linear regression model over the data, you need to plot the data first. Using the command "***hold on***", you can keep this plot visible to the next consecutive plot. Then, plot the linear model.

8. **Test the prediction model with a new dataset.** Load the testing data file "car_mpg_testingData.csv". The data file format is the same with the training data. To make the predictions for the new testing data you need to do the following steps:
   - **E**xtract the input feature and the output from the new data. The output data will be used to evaluate the prediction model on the new testing data.
   - Normalize the input feature. **You have to use the same mean and the same standard deviation which are calculated with training data** in section Part1-3.
   - Add the intercept term (Bias term) in the input vector. A new input matrix is generated by concatenating the bias term vector and the new normalized input feature vector.
   - Make the predictions for the new testing data using the linear regression model you trained in Part1-5.
   - **Calculate the MSE cost for the new testing data**
   - **Question 2:** what is the MSE error of the new testing data?

Question3: Conduct an experiment with three different learning rates: ∂ =0.001, 0.01, and 0.1. Generate the MSE cost convergence plots with three different learning rates. Write your observations of the effect of the learning rate in machine learning.

Part II: Linear regression with multiple variables

A multivariable linear regression models the relationship between two or more input features (variables) and an output variable by fitting a linear equation. You will implement the linear regression with multiple variables to predict mpg for vehicles using the automotive dataset. The input features used in the prediction model are cylinders in the 2nd column, displacement in 3rd column, and horsepower in 4th column. The lines of code in Part1 will used in Part II with very few modifications in feature extraction and data visualization parts. Open the provided live script file, "Lab2_Part2.mlx" and do the followings:

1. Load the data from the file "car_mpg.csv" and feature extraction:
   - The input features are cylinders in the 2nd column, displacement in 3rd column, and horsepower in 4th column.
   - The output, mpg is in the first column.
2. Data visualization
   - Generate three plots of each input feature versus the output y (mpg)
   - Utilize the subplot command to generate three subplots in a row.
3. Data normalization and add a bias term

4. Find parameters using gradient descent
5. Plot the MSE cost convergence graph
6. Test the prediction model with a new testing dataset. Load the data file "car_mpg_testingData.csv". Do the steps in Part1-8.
   Question 4: what is the MSE error of the multivariable linear regression model with the new testing data?

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  - **A report (10 pts): contains answers for Question1 –Question4.**
  - Part 1 (45pts): The live script file, " Lab2_Part1.mlx", in that all outputs are generated either inline or on the right side.
  - Part 2 (45pts): The live script file, "Lab2_Part2.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab2_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab2_Green_Smith.zip