

ASSIGNMENT-4

Name : Arsh Pratap

Roll No. : 2018-IMT-021

Course – Machine Learning (ITIT 4103)

Deadline – 18th October 2021

Problem Statement

Suppose you are the CEO of a restaurant franchise and are considering different cities for opening a new outlet. The chain already has trucks in various cities and you have data for profits and populations from the cities. You would like to use this data to help you select which city to expand to next.

The file ex1data1.txt contains the dataset for our linear regression problem. The first column is the population of a city and the second column is the profit of a food truck in that city. A negative value for profit indicates a loss.

1. Use a scatter plot to visualize the data, since it has only two properties to plot (profit and population).
2. Consider a simple linear model with two parameters and one input variable and mean square error cost function to implement the gradient descent algorithm to find the intercepts. Assume a suitable terminating condition.
3. Plot the model alongside the scatter plot to show the fit model.
4. Perform steps 1,2,3 in batch mode for varying values of alpha, learning rate and plot the results.
5. For each of the experiments performed above in steps 1,2,3,4 with varying learning rates visualize the cost function as a contour plot as well as plot the values of parameters to visualize the stepwise traversal of the parameters on this contour plot.

Note - The reader can find the code for the assignment at the following link :

<https://github.com/arshPratap/ML-Assignment/blob/main/Assigment4.ipynb>

Steps followed are as follows :

- Use pandas to read the required the data
- Plot x and y (as shown)



- Cost Function(Mean Squared Error)

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

- Linear regression formula

$$h_{\theta}(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

- We build gradient descent model using the equation

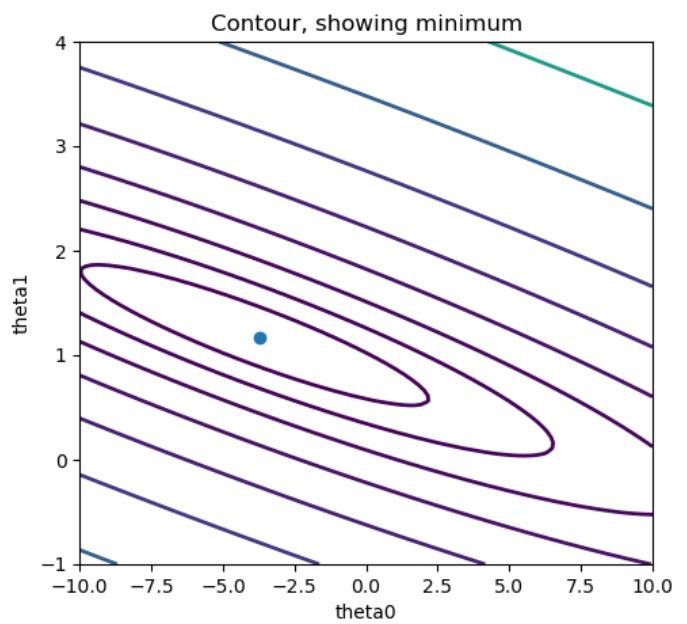
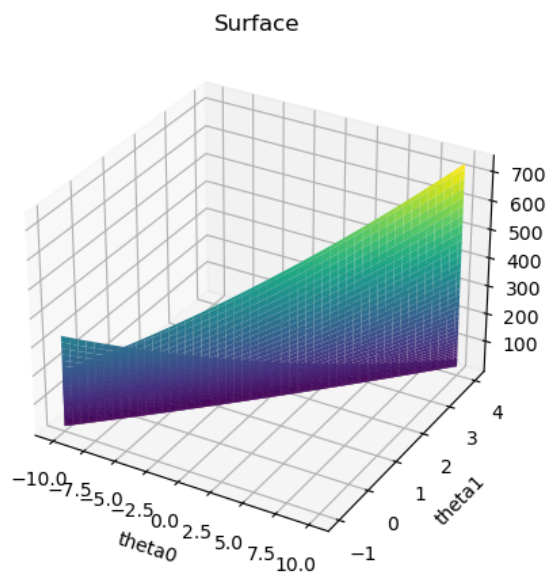
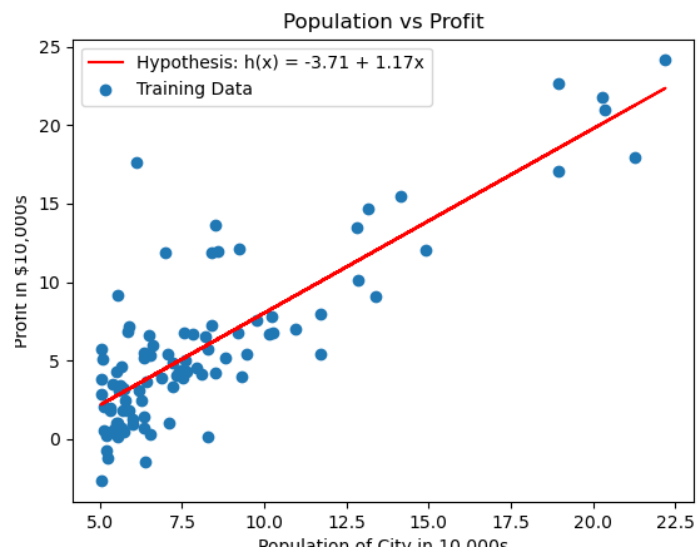
$$\theta_j = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad \text{simultaneously update } \theta_j \text{ for all } j$$

The following steps are done for theta 0 and theta 1 for n steps,(n=1500) :

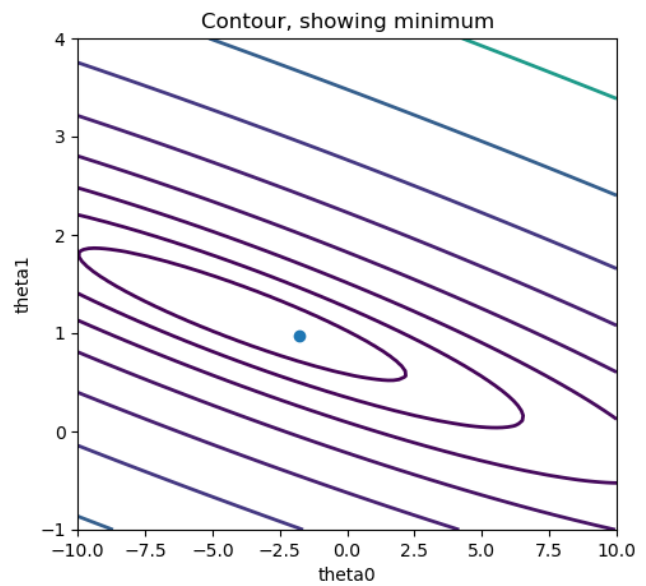
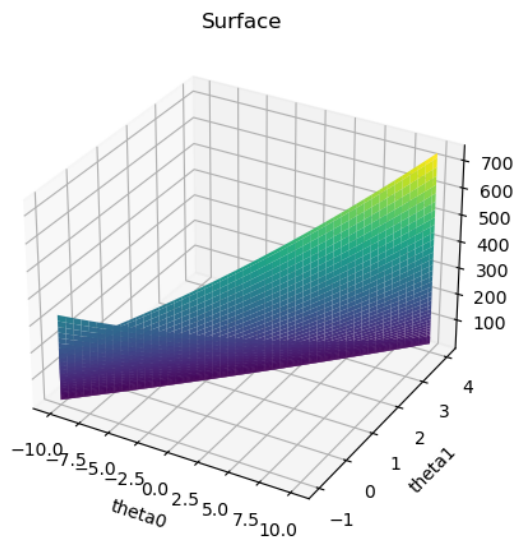
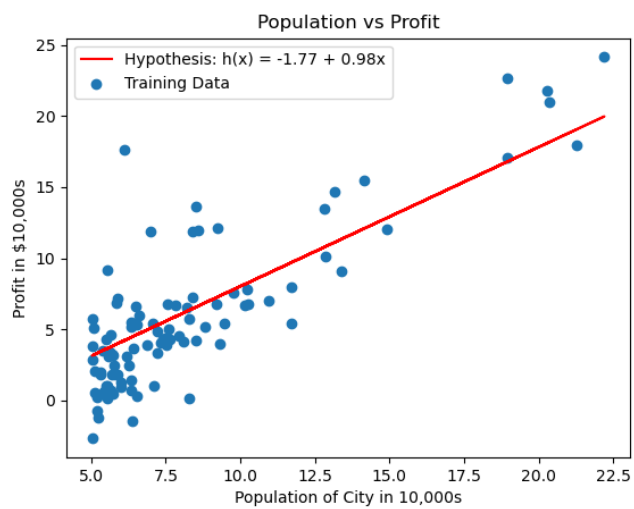
- Contour built using matplotlib plt contour function
- For tita 0 from range -10,10 and tita 1 from range -1,4 cost is calculated and these 3 variables are passed to contour function

Result

Learning rate **0.01**



Learning rate 0.001



Learning rate 0.0001

