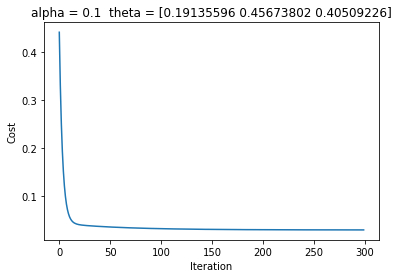
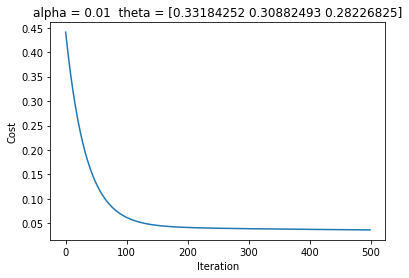
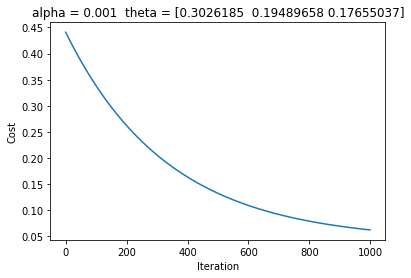
**Report on the program from Assignment 2  
Machine Learning (CS-596)**

**Name: Dhaval Harish Sharma  
Red ID: 824654344**

**Introduction:**This program takes a CSV file named “sat” containing data of students as input and calculates the student’s “GPA” based on two feature vectors called “Math SAT” and “Verb SAT” by using Multivariate Linear Regression as output.

**Working of the program:**The program starts with importing the necessary libraries for its working. “download\_data.py” file contains logic for getting the data from the sat file. Numpy is a library which provides a powerful array with necessary mathematical computations. Matplotlib is used for plotting the results of the processing. There is a file called “gradientdescent.py” which is used to find gradient descent in order to get an optimized theta vector.

The variables ALPHA and MAX\_ITER are initialized in the beginning of the program. The value of ALPHA indicates the number of steps travelled in the direction of decreasing cost function whereas the value of MAX\_ITER indicates the number of iterations to perform before ending the convergence. Let’s look at the cost functions for different values of ALPHA and MAX\_ITER as follows:   
The graph with the value of ALPHA = 0.1 and MAX\_ITER = 300 is shown in the left. As we can see from the figure, the cost reaches to minimum rapidly during the initial iterations. It just takes around 100 iterations to reach near the minimized cost function. But, this may not always be the case. If the value of ALPHA is high, it may happen that the gradient may skip the minimum and never come back to minimum. In this case, the result is 0.16954203126934056 (0.1250284686320857).  
  
Let us consider another case for ALPHA = 0.01 and MAX\_ITER = 500 as follows:  
  
The cost function decreases in a smooth fashion over time as the number of iteration increases. It takes around 200 to 300 iterations for cost function to come closer to the minimum. Also, note that the values of theta are different than before. The results are 0.147017033893325 (0.11722709952151875) which are better than the last set of ALPHA and MAX\_ITER.  
  
  
Let’s consider a last case which contains the value of ALPHA = 0.001 and MAX\_ITER = 1000 as follows:

  
This time the curve takes more iteration to get near the minimum. It is due to the fact that the graph tends to take a smaller step towards minimum as compared to larger steps taken by bigger values of ALPHA during an individual iteration. The results are 0.22406219440664307 (0.11280487318379694) which means it still needs more iterations to get to the minimum.  
  
The next step is to load the data from the CSV file in to a pandas dataframe. The function download\_data takes two parameters; first is the file name and second is the number of columns. Then, we normalize the data to make it fit better to the model. First, we find the minimum and maximum values from each column and then divide the subtraction of each element with the minimum value in its column to the range of the column. This calculates the min\_max normalization in order to properly scale the values between 0 and 1. The data is then splitted in to two parts, testing and training. The training part is used to train the model by using linear regression and the testing part is used to evaluate the accuracy of the model. The values of theta are initialized to zero and passed to the function gradientDescent to calculate the optimum values of theta which fits the model perfectly.

The gradient descent function calculates the hypothesis by multiplying the theta and transposed X matrices which results in the value of predicted GPA, according to the current theta which is a zero vector. Then, it finds the residual error by subtracting Y from hypothesis. After finding the residual error, it finds the gradient which is the change in the mean squared error. The new values of theta are found by subtracting the old values of theta with the multiplication of ALPHA and the gradient vector. Finally, the cost function is calculated by getting a sum of squares of residual errors with respect to the value of theta. This process is repeated MAX\_ITER times.

Lastly, the values of cost function and the number of iterations are plotted in a graph and the average error and standard deviation is calculated which gives us the idea about how well our model is trained and can predict the values.

**Findings:**By looking at the graphs, we can say that the value of ALPHA should be taken care of because if the value of ALPHA is high, then we may converge faster but there’s a risk that we may not get to minimum at all. Also, if the value of ALPHA is too low, then it may take a large number of iterations to get to the minimum. Thus, the value of ALPHA should be optimal, so that it’s not taking larger steps or smaller steps but instead it’s taking perfect steps. The number of iterations also matter a lot because if we keep it low then there’s a risk that we may not reach the minimum which would need more iterations or if we keep it high then if we reach to minimum early then the remaining iterations are wasted.

The gradient descent is a powerful method for finding the straight line which fits the training data perfectly. It starts with some values of theta and then eventually updates the values of theta as it gets to the minimum cost function. Also, there is a good property of gradient descent which is that we don’t need to update ALPHA values dynamically because if the cost function is far from minimum, it takes larger steps and when it’s near to the minimum it gradually reduces the step size which results in adaptive step sizes. For example, if the cost function is far from minimum then the d(F) / dΘ value is large whereas if the cost function is near to the minimum then the d(F) / dΘ value is small.

**Conclusion:**In conclusion, we can say that linear regression is a very good model to predict the values if the training data set is spread linearly such that a straight line can easily fit those values with least amount of error. The data should be properly scaled in order to get best results. The gradient descent is a powerful mechanism to generate minimum cost function without having to perform trial and error on each and every combination of theta values. In addition to this, the residual sum of squares gives an optimal parameter to find the cost function.