Assignment-1 Report

# Introduction:

In this assignment, we tried to have an understanding about the various optimization techniques for linear regression model training.

The following methods were implemented using python,and various libraries of python like numpy and matplotlib e.t.c :

1. Gradient Descent
2. Stochastic Gradient Descent
3. GD with L1 regularization
4. GD with L2 regularization
5. Normal Equation solution

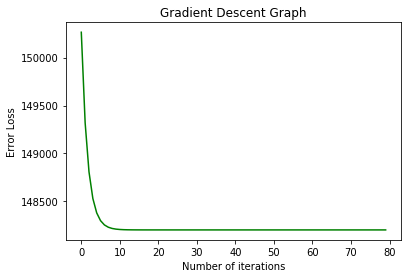
The given data consisted of 434874 rows and 2 columns (latitude and longitude). Before the model was trained, all data features were normalized by using standardization method(column-wise).

# Optimization Algorithms:

## *Gradient Descent:*

In this model, we attempted to replicate a rudimentary Gradient Descent algorithm without any regularization terms. The equation assumed to be for gradient descent fit was. The error function (cost function) associated with this was, where yi denotes the predicted value and Yi denotes the actual target value. The estimated learning rate, in this case was around 2e-6 i.e, 0.000002. The weights obtained in this case were [-5.38915579e-15,1.49685594e-01, -1.90101138e-01] which were updated after every iteration (batch size = 304412).The weights were initialized to [0,0,0] and the stopping criteria was | E(t)-E(t+1)|<=1e-5 i.e., 0.00001.The R2 and RMSE error for the testing data was 0.02501284 and 0.697849250 respectively and for the training data was 0.0249981204 and 0.6980879 respectively .The stopping criteria was |E(t)-E(t-1)|<=1e-5.The minimum sum of square of error is 148201.1244951

The graph obtained is shown below:

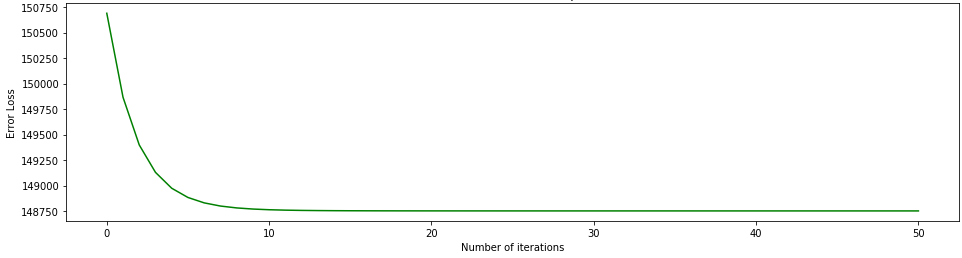


## *Stochastic Gradient Descent:*

In this model, a rudimentary SGD was applied on the given training set. The error/cost function in this as well as held to be. The weights were updated after every data point analysis (batch size = 1). The estimated learning rate, in this case was around 2e-6 i.e, 0.000002. The weights obtained in this case were [-1.08672495e-04, 1.35278690e-01, -1.68855234e-01] which were updated after every iteration (batch size =1).The weights were initialized to [0,0,0] and the stopping criteria was | E(t)-E(t+1)|=1e-5 i.e., 0.00001.The R2 and RMSE error for the testing data was 0.02578859 and 0.6969466 respectively and for the training data was 0.022718436 and 0.6990284557 respectively .The stopping criteria was |E(t)-E(t-1)|<=1e-5.The minimum sum of square of error is 148751

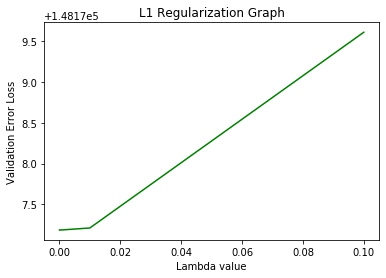
The resulting graph obtained for the SGD is shown below:

For x-axis 1unit=304412 iterations



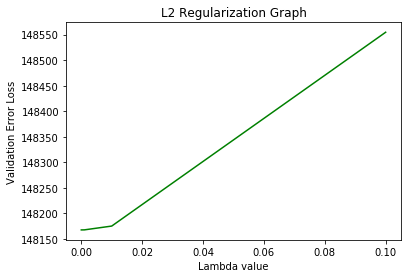
## *Gradient Descent with L1 regularization (Lasso):*

In this model, the cost function was modified to where β is the L1 regularization coefficient. This technique helps us to curb the higher values of weights. A list of L1 values ([10e-10, 10e-9, 10e-8, 10e-7, 10e-6, 10e-5,10e-4,10e-3, 10e-2, 10e-1]) were used for training .We notice that there is not much effect of L1 regularization on this model as we are using a linear regression model of degree one and there is no overfitting. that is why we notice that as the lambda value keeps on increasing the validation loss keeps on increasing as the second parameter containing the regularization coefficient keeps contributing to the loss as it increases



## *Gradient Descent with L2 regularization (Ridge):*

In this model, the cost function was modified to where β is the L2 regularization coefficient. This technique helps us to curb the higher values of weights. A list of L2 values ([10e-10, 10e-9, 10e-8, 10e-7, 10e-6, 10e-5,10e-4,10e-3, 10e-2, 10e-1]) were used for training .We notice that there is not much effect of L2 regularization on this model as we are using a linear regression model of degree one and there is no overfitting. that is why we notice that as the lambda value keeps on increasing the validation loss keeps on increasing as the second parameter containing the regularization coefficient keeps contributing to the loss as it increases

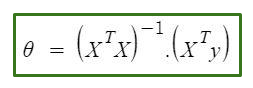


## *Solution using Normal Equations:*

In this method, a simple linear equation solution was followed. The partial derivative of the error with respect to each weight were set to 0, and the resulting three equations were solved for the three unknown weights. The linear equations obtained were as follows:



After solving the above 3 equations we get the below equation



The final weights obtained were [[-8.15573646e-15],[ 1.34566127e-01],

[-1.73478687e-01]]. The R2 and RMSE values obtained finally were 0.0256952278 and 0.7001571082955066 respectively.

# Conclusions:

* The fastest and most accurate prediction of the parameters was obtained using normal equations, as they give a straightforward mathematical solution.
* Gradient Descent converged after 60 odd iterations and the thereafter the error almost remained constant with no fluctuations.
* All the data was shuffled randomly prior to running all the above algorithms to ensure that the model trained on random points without an order. The SGD model converged after 25 odd iterations but the value kept fluctuating In addition, the training time in case of SGD was comparatively lesser than the other methods(excluding normal equations).
* As in L1 regularization we take the mod of weights we observe that the error loss increase in the L1 graph is slightly higher than the L2 regularization graph. But,still L1 and L2 regularization were not of much significance in this model as the weights were quite small and there was no over-fittingas it was a one degree equation.