Assignment-2 Report

# Introduction:

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

The following methods were implemented using python, numpy and matplotlib:

1. Gradient Descent for Degree 2
2. Gradient Descent for Degree 3
3. Gradient Descent for Degree 4
4. Gradient Descent for Degree 5
5. Gradient Descent for Degree 6
6. Gradient Descent for Degree 6-L1 Regularization
7. Gradient Descent for Degree 6-L2 Regularization

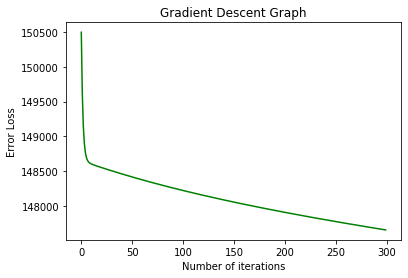
The given data consisted of 434874 rows and 2 columns (latitude and longitude). Before the model was trained, all data features were standardized (column wise), shuffled and split into training (70%) and test data (30%).

# Polynomial Models:

## *Gradient Descent-Degree 2:*

In this model, we attempted to replicate a Gradient Descent algorithm for Degree 2 polynomial without any regularization terms. The equation assumed for gradient descent fit was of 6 terms. 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 0.0000009 and the model converged to a minimum squared error of 147658 after around 300 iterations. The weights obtained in this case were [[-3.01063259e-01, -2.81602896e-01, 1.80715129e-01, -1.42598610e-15, 2.71455408e-01, 8.79852490e-02]]

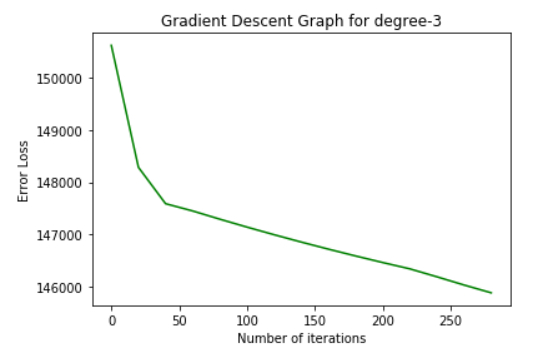
which were updated after every iteration (batch size = 304412). The stopping criteria was kept |E(t)-E(t-1)|<=3



The weights were initialized by zero initially. The R2 and RMSE error in this case for Train data were 0.02987787 and 0.720505781072 respectively. The R2 and RMSE error in this case for Test data were 0.02480629 and 0.7005258 respectively.

## *Gradient Descent-Degree 3:*

In this model, we attempted to replicate a Gradient Descent algorithm for Degree 3 polynomial without any regularization terms. The equation assumed for gradient descent fit was of 10 terms . 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 0.0000001 and the model converged to a minimum squared error of 146319 after around 300 iterations. The weights obtained in this case were [[-5.92249284e-01, -1.20849654e-01, 2.82348054e-01, -7.12918512e-02, -9.61326354e-02, -7.96597936e-02, 3.23800338e-01, 8.04714150e-15, 3.69210031e-01, -8.81300893e-02]] which were updated after every iteration (batch size = 304412).The stopping criteria was |E(t)-E(t-1)|=3.5



The weights were initialized by zero initially. The R2 and RMSE error in this case for Train data were 0.0423857947and 0.69195888 respectively. The R2 and RMSE error in this case for Test data were 0.038932484 and 0.69501085 respectively.

## *Gradient Descent-Degree 4:*

In this model, we attempted to replicate a Gradient Descent algorithm for Degree 4 polynomial without any regularization terms. The equation assumed for gradient descent fit was of 15 terms. 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 0.0000005 and the model converged to a minimum squared error of 144682.04774792

after around 300 iterations. The weights obtained in this case were

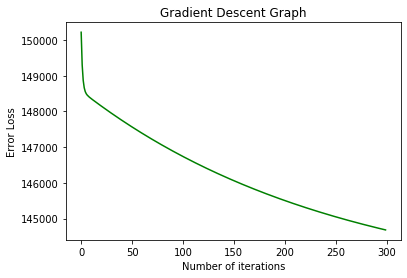
[[-4.75250136e-01,-2.15752746e-01, 1.91623364e-02, 2.05694081e-01,

-5.81946921e-02 ,-2.05069146e-01, 3.37450161e-02 , 2.30318491e-01,

-6.29305974e-02 , 4.93245026e-02, -6.77253024e-02 , 2.57292006e-01,

1.68208847e-15 , 2.86556629e-01, -7.25782390e-02]]

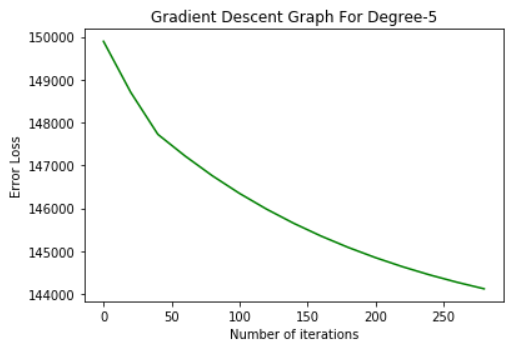
which were updated after every iteration (batch size = 304412).The stopping criteria was |E(t)-E(t-1)|<=4



The weights were initialized by zero initially. The R2 and RMSE error in this case for Train data were 0.04942957 and 0.689523952 respectively. The R2 and RMSE error in this case for Test data were 0.0462876 and 0.68934388 respectively.

## *Gradient Descent-Degree 5:*

In this model, we attempted to replicate a Gradient Descent algorithm for Degree 5 polynomial without any regularization terms. The equation assumed for gradient descent fit was of 21 terms . 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 0.0000007 and the model converged to a minimum squared error of 143991.24962784 after around 300 iterations. The weights obtained in this case were [[-3.53701926e-01, -2.04975600e-01, -6.77427858e-02,5.31837890e-02,1.39192752e-01,-4.94920692e-02,-1.99574598e-01,-6.09272374e-02,6.25130735e-02,1.54576715e-01,-5.22956206e-02,-5.37275651e-02, 7.24266562e-02, 1.71458391e-01, -5.51333565e-02, 8.29325815e-02, -5.80049430e-02 ,1.89848381e-01, -6.56704932e-15, 2.09685353e-01, -6.09100441e-02]] which were updated after every iteration (batch size = 304412).The stopping criteria was |E(t)-E(t-1)|<=5



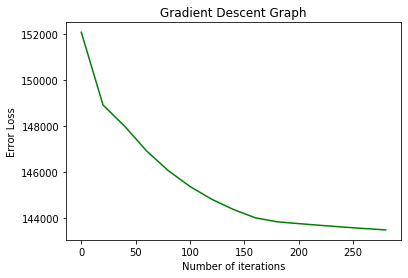
The weights were initialized by zero initially. The R2 and RMSE error in this case for Train data were 0.05396816 and 0.6878171 respectively. The R2 and RMSE error in this case for Test data were 0.054786966 and 0.6886091 respectively.

## *Gradient Descent-Degree 6:*

In this model, we attempted to replicate a Gradient Descent algorithm for Degree 6 polynomial without any regularization terms. The equation assumed for gradient descent fit was of 28 terms . 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 0.0000002 and the model converged to a minimum squared error of 143416.44957055

after around 300 iterations. The weights obtained in this case were [[-2.63826691e-01 -1.73004234e-01 -8.81271866e-02 -1.09447779e-02 5.47410426e-02 9.42075789e-02 -4.31920634e-02 -1.70103483e-01 -8.45655392e-02 -6.40590540e-03 6.09891478e-02 1.04249632e-01 -4.49805201e-02 -8.08279833e-02 -1.63493262e-03 6.76082074e-02 1.15291409e-01 -4.67903215e-02 3.37327022e-03 7.46034030e-02 1.27367484e-01 -4.86212618e-02 8.19757770e-02 -5.04731331e-02 1.40477311e-01 1.68616080e-15 1.54568056e-01 -5.23457261e-02]]

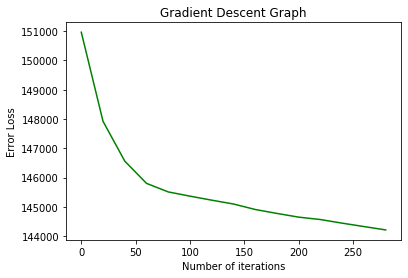
which were updated after every iteration (batch size = 304412).The stopping criteria was |E(t)-E(t-1)|<=5



The weights were initialized by zero initially. The R2 and RMSE error in this case for Train data were 0.05774463 and 0.68631867 respectively. The R2 and RMSE error in this case for Test data were 0.0556978561 and 0.6864069 respectively.

## *Gradient Descent-Degree 6 L1:*

In this model, we attempted to replicate a Gradient Descent algorithm for Degree 3 polynomial without any regularization terms. The equation assumed for gradient descent fit was of 28 terms . 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 0.0000001 and the value of the regularization coefficient was 0.002 and the model converged to a minimum squared error of 144102.9788097 after around 300 iterations. The weights obtained in this case were [[-2.14051431e-01 -1.39833491e-01 -7.01647753e-02 -6.59920891e-03 4.52173885e-02 7.58236685e-02 -4.11915145e-02 -1.37142418e-01 -6.69016174e-02 -2.54828633e-03 5.04979264e-02 8.42457042e-02 -4.24568536e-02 -6.34953869e-02 1.56543775e-03 5.60708018e-02 9.34737231e-02 -4.37379643e-02 5.79776037e-03 6.19384940e-02 1.03532087e-01 -4.50346962e-02 6.80999582e-02 -4.63468971e-02 1.14415525e-01 -2.01943505e-16 1.26075087e-01 -4.76744141e-02]] which were updated after every iteration (batch size = 304412). The stopping criteria was |E(t)-E(t-1)|<=5

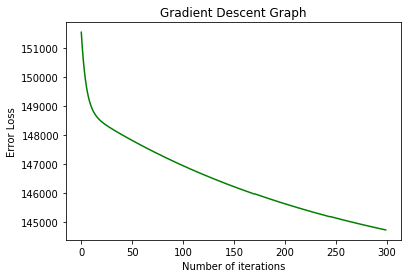


The weights were initialized by zero initially. The R2 and RMSE error in this case for Train data were 0.0539969248 and 0.687751 respectively. The R2 and RMSE error in this case for Test data were 0.0529794682 and 0.68964887 respectively.

## *Gradient Descent-Degree 6 L2:*

In this model, we attempted to replicate a Gradient Descent algorithm for Degree 3 polynomial without any regularization terms. The equation assumed for gradient descent fit was of 28 terms . 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 0.0000001 and the value of regularization coefficient is 0.002 and the model converged to a minimum squared error of 144735.70344159 after around 300 iterations. The weights obtained in this case were [[-1.68858927e-01 -1.10761127e-01 -5.61028693e-02 -6.11277666e-03 3.64653212e-02 6.09688249e-02 -3.92843863e-02 -1.08440396e-01 -5.33068534e-02 -2.60652554e-03 4.12206852e-02 6.84836957e-02 -4.01861375e-02 -5.03951409e-02 1.05453237e-03 4.62302117e-02 7.67102105e-02 -4.10995451e-02 4.87271140e-03 5.14956346e-02 8.56713907e-02 -4.20245001e-02 5.70155582e-02 -4.29608924e-02 9.53646910e-02 -2.75729186e-15 1.05749612e-01 -4.39086116e-02]]

which were updated after every iteration (batch size = 304412).The stopping criteria was |E(t)-E(t-1)| <=5



The weights were initialized by zero initially. The R2 and RMSE error in this case for Train data were 0.048821323 and 0.6896298 respectively. The R2 and RMSE error in this case for Test data were0.04791285342 and 0.68699534 respectively.

# Conclusions:

* We notice that the r2\_score keeps on increasing for both train and the test data as the degree of the equation increases.Ideally if there was overfitting for the higher degrees then the test data r2\_score should have decreased.But we don’t observe that indicating that there is no overfitting even for the higher degrees this may be due to the huge size of the dataset.
* We see that the r2\_score for the degree 6 is more than the L1,L2 regularization of the degree 6.As there is no overfitting in degree 6 we do not see any increase in the r2\_score over its regularization counter parts.Therefore there is no need of regularization on degree 6 as there is no overfitting for degree 6 model.
* We see that L1,L2 regularization for the degree 6 on test data gives r2\_score similar to that of the degree 4 and degree 5.This is because there is no overfitting in degree 6 without regularization