COL774 – Machine Learning ( Assignment -1)

Rajat Singh (2020CSZ8507)

**Question 1: Linear Regression**

**Part A:**

Text

Description automatically generated

**Part B:**

Stoping criteria : abs cost difference < 0.0000001 or epoch>10000

Learning rate : 0.001

Number of epochs : 67

Final Theta :

[[0.00133904]

[0.9957635 ]]

Chart, scatter chart

Description automatically generated

Learning rate : 0.01

Number of epochs : 2

Final Theta :

[[0.0013402]

[0.9966201]]

Chart, scatter chart

Description automatically generated

**Part C** :

Stoping criteria : abs cost difference < 0.0000001 or epoch>10000

Learning rate : 0.001

Number of epochs : 67

Final Theta :

[[0.00133904]

[0.9957635 ]]

Chart

Description automatically generated

**Part D :**

Stoping criteria : abs cost difference < 0.0000001 or epoch>100

Learning rate : 0.001

Number of epochs : 67

Final Theta :

[[0.00133904]

[0.9957635 ]]

Diagram

Description automatically generated

Part E :

Diagram

Description automatically generatedDiagram

Description automatically generatedChart, scatter chart

Description automatically generated

Above are the plot of the error function at different learning rate. From these plots we can see that if we increase the learning rate after a certain threshold then the cost function will start diverging instead of converging. Since learning rate controls the size of the steps we are moving towards the minimizing the cost function. Thus smaller learning rate leads to small step size and thus slowly and steadily we will reach the minima and vice versa is also Ture but increasing the learning function after a certain threshold leads to overshoot of the cost function.

Thus we have to choose the learning rate accordingly as small learning rates leads to very small step thus will take long time to converge but inversely large learning rate leads to overshoot of the cost function and leads to no result.

**Question 2: Sampling and Stochastic Gradient Descent**

Part A:

1 million samples are

Part B:

For learning rate 0.001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Batch size | Absolute cost difference threshold | Iterations | Time (s) | Learned theta |
| 1 | 1e-2 | 2 | 29.65 | [3.,1.,2.] |
| 100 | 1e-2 | 2 | 2.58 | [3.,1.,2.] |
| 10000 | -- | -- | -- | Overshoot |
| 1000000 | -- | -- | -- | Overshoot |

Here from this table we can see that for 10000 batch size or above the theta is overshooting at learning rate 0.001.

Part C:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Batch size | Absolute cost difference threshold | Iterations | Learning rate | Time (s) | Learned theta |
| 1 | 1e-5 | 2 | 0.001 | 29.65 | [3.,1.,2.] |
| 100 | 1e-5 | 2 | 0.0005 | 2.6 | [3.,1.,2.] |
| 10000 | 12-2 | 6 | 0.000001 | 12.26 | [2.9244957, 1.00891407, 1.99703892] |
| 1000000 | --- | --- | 0.000001 | 22.46 | For as small learning rate it is overshooting. |
|  |  |  |  |  |  |

Error for the original theta : 0.9829469214999997

For batch\_size : 1 , learning\_rate : 0.001 , threshold : 1e-05

(1000000, 3)

Number of epochs : 2

Final Theta :

[[3.]

[1.]

[2.]]

Error : 0.9829469215000003

For batch\_size : 100 , learning\_rate : 0.0005 , threshold : 1e-05

(1000000, 3)

Number of epochs : 2

Final Theta :

[[3.]

[1.]

[2.]]

Error : 0.9829469215

For batch\_size : 10000 , learning\_rate : 1e-06 , threshold : 0.01

(1000000, 3)

Number of epochs : 6

Final Theta :

[[2.92454725]

[1.00885616]

[1.9970328 ]]

Error : 0.9895588908248337

For batch\_size : 1000000 , learning\_rate : 1e-06 , threshold : 0.01

(1000000, 3)

Number of epochs : 21

Final Theta :

[[ 2.57835931e+28]

[ 1.96593104e+29]

[-6.51436004e+28]]

Error : 2.2069729622422818e+60

Here in this part I took different values of the learning rate and check the convergence of the theta using stochastic gradient descent. We can see from the above reading that for 1000000 at a small value of learning rate the theta is not converging and it overshoots. For batch size =1 it took only two passes to converge theta for a given error threshold. It could be possible that theta converges to the required value in between the epoch but I didn’t consider that case and check the deviation after the full epoch pass. For batch size 1000000 the SGD overshoot the theta value for small learning rate.

Part D:

Chart

Description automatically generated

Chart

Description automatically generated

Chart

Description automatically generated

Chart, radar chart

Description automatically generated

Here from the above plots we can see that when our batch size was 1 then the theta was changing in a zig zag way which matches our intuition of SGD . For batch size 1 we are getting the optimal value of theta but the direction of the theta is very zig zag and is not constant. For batch size 100 the same scenario can be observed. But for the batch size 10000 the direction is very smooth. For batch size 1000000 we are not getting the correct answer as it is not converging for the optimal value of theta.

**Question 3: Logistic Regression**

Part A:

Final Theta :

[ 0.40125316]

[ 2.5885477 ]

[-2.72558849]

Epochs : 7

Part B:

Chart, scatter chart

Description automatically generated

Question 4: Gaussian Discrmimant Analysis

Part A :

Phi : 0.5

mew0 : [[ 0.75529433] [-0.68509431]]

mew1 : [[-0.75529433] [ 0.68509431]]

Covariance matrix :

[ 0.42953048 -0.02247228]

[-0.02247228 0.53064579]

Part B:

Chart, scatter chart

Description automatically generated

Part C:

Chart, scatter chart

Description automatically generated

Part D:

Phi : 0.5

Mew\_0:[[ 0.75529433][-0.68509431]]

Mew\_1:[[-0.75529433][ 0.68509431]]

covariance matrix 0 :

[0.47747117 0.1099206 ]

[0.1099206 0.41355441]

covariance matrix 1 :

[ 0.38158978 -0.15486516]

[-0.15486516 0.64773717]

Part E:

Chart, scatter chart

Description automatically generated

Part F:

From comparing the plots above we can state that the quadratic boundary fits the data little better than that of the linear boundary.