Assignment 1

Sreemanti Dey

August 2022

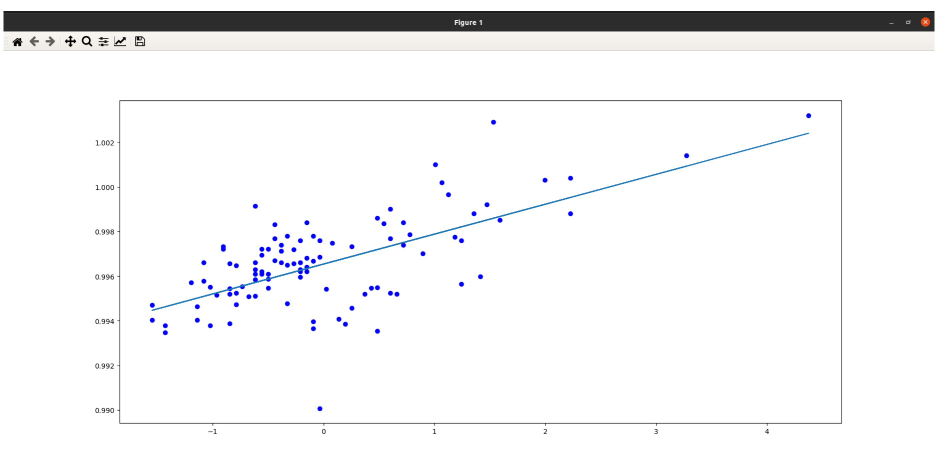
1 Part 1

1.1 Part a

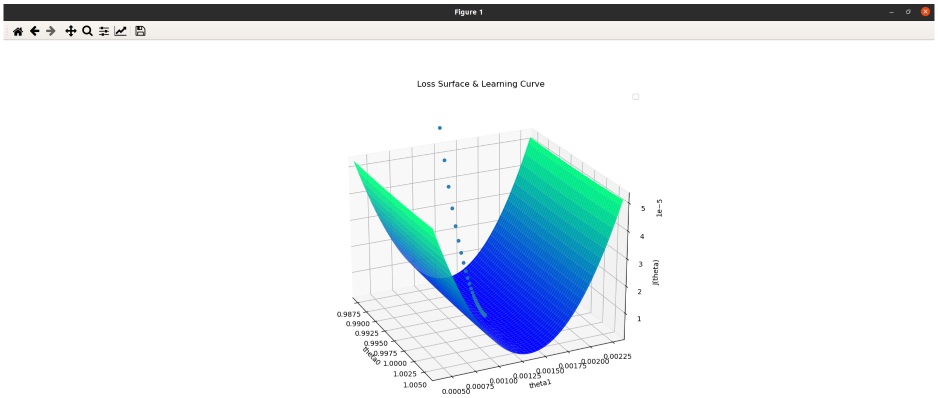
Learning rate : 0.1
Stopping criteria : absolute (previous_cost - new_cost) $\leq 10^{-9}$
Final learnt parameters :

$$\theta = \begin{bmatrix} 0.99653574 \\ 0.00134008 \end{bmatrix} \tag{1}$$

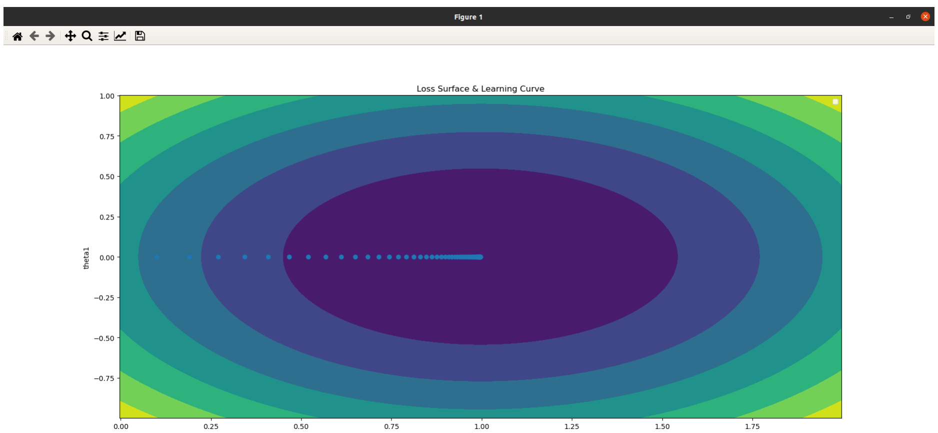
1.2 Part b



1.3 Part c

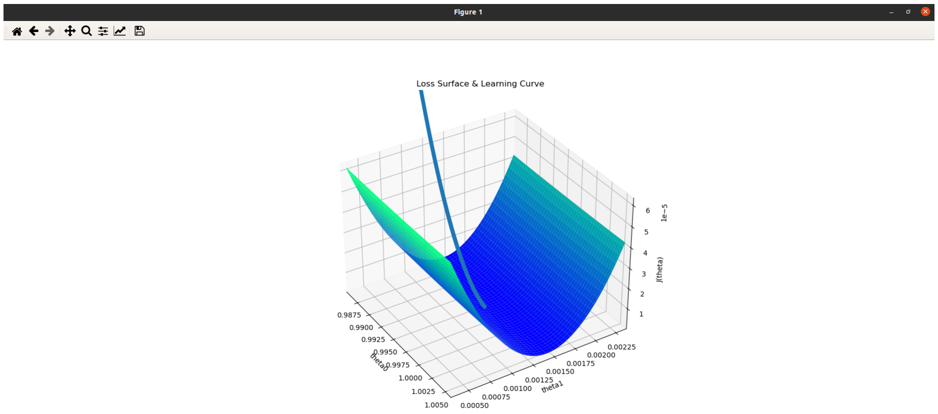


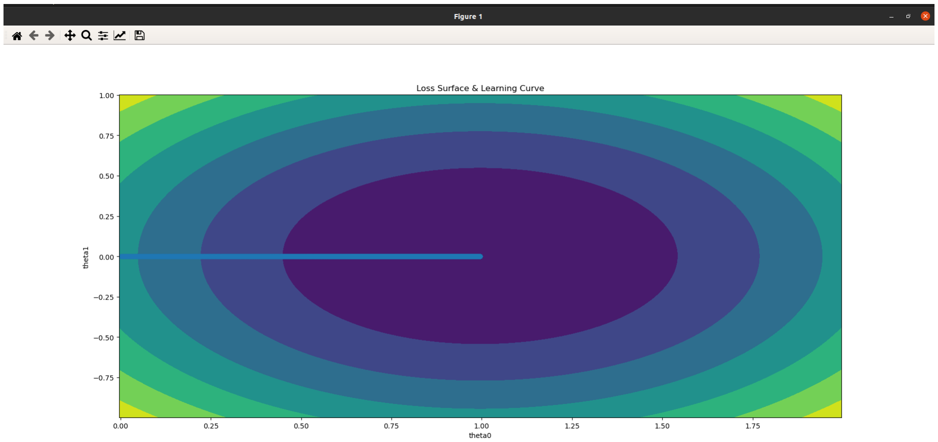
1.4 Part d



1.5 Part e

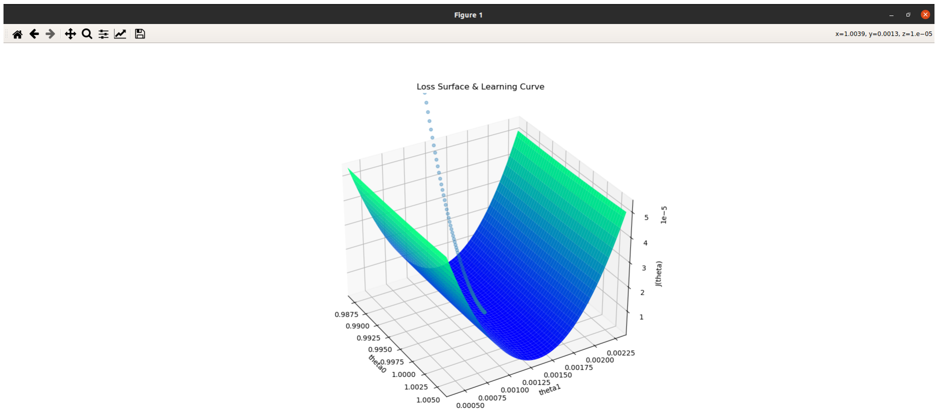
1. learning rate : 0.001

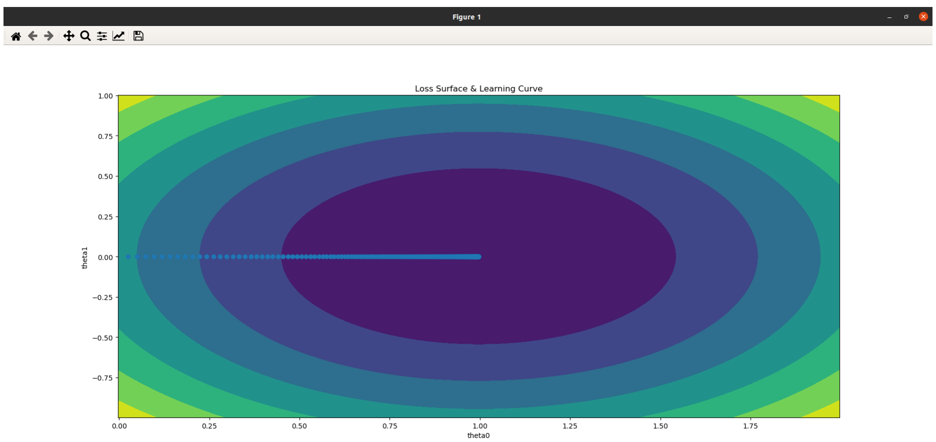




Since the learning rate is very small, we see very slow movement with each step, so the graph is dense even far away from the minima, which is not the case in other graphs.

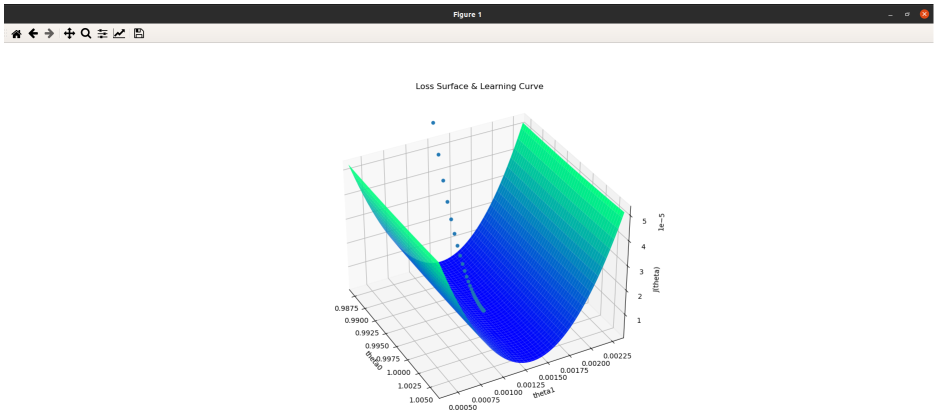
2. learning rate : 0.025

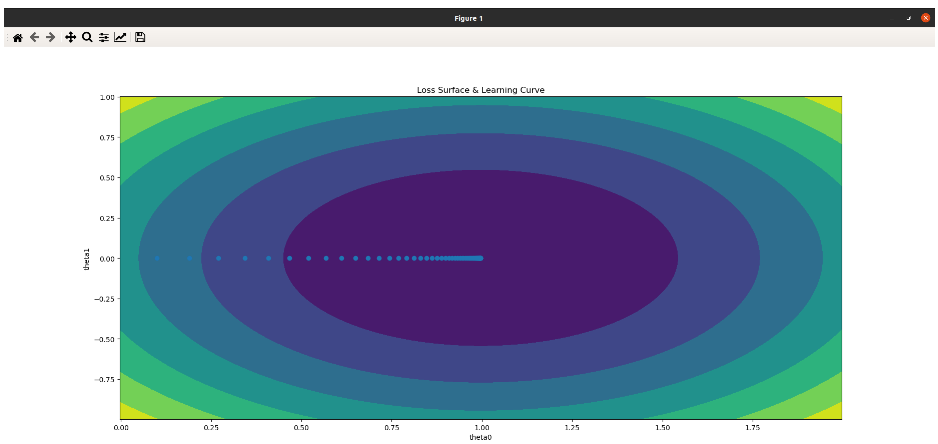




Here, the learning rate is midway, hence we see sparse (more sparse than the previous figures but less than the next) and hence this converges faster than the previous one.

3. learning rate : 0.1





Here, the learning rate is the smallest, hence we see very rapid movement with each step and hence sparse at the top, but all three converge to the same minima, this converges the fastest among all.

2 Part 2

2.1 Part a

The sampling is done.

2.2 Part b

For each batch, convergence criteria depends on 2 factors :

1. cost over last 1000 examples - This has been varied for different batch sizes

to get best fit model. We have taken the cost over the last 1000 iterations, and taken this as the main convergence criteria. This criteria has been set as different for different batch sizes to ensure convergence.

2. max iterations - This has also been varied according to different batch sizes.

Batch sizes are as follows :

1. 1
   Cost $\leq$ 0.000000001, max iterations $\leq$ 1000000

2. 100
   Cost $\leq$ 0.01, max iterations $\leq$ 100000

3. 10000
   Cost $\leq$ 0.1, max iterations $\leq$ 50000

4. 1000000
   Cost $\leq$ 0.1, max iterations $\leq$ 50000

2.3 Part c

Yes the different algorithms have converged to the same parameter values in this case. The difference is not much, as we can see the theta values obtained from different batch sizes have been written below :

1. Batch size 1 :

$$\theta = \begin{bmatrix} 2.98948772 \\ 1.04687761 \\ 2.00137386 \end{bmatrix} \tag{2}$$

   Converges the fastest - in around 1-2 sec.
   Number of iterations : 86605
   Cost wrt original hypothesis : 0.9829469215
   Cost wrt learned hypothesis : 1.0908479249900047

2. Batch size 100 :

$$\theta = \begin{bmatrix} 3.00561096 \\ 0.99588133 \\ 1.99912901 \end{bmatrix} \tag{3}$$

   Converges in around 2-3 sec.
   Number of iterations : 1000000
   Cost wrt original hypothesis : 0.9829469215
   Cost wrt learned hypothesis: 0.9841747670580564

3. Batch size 10000 :

$$\theta = \begin{bmatrix} 3.0064271 \\ 0.99801853 \\ 2.00048362 \end{bmatrix} \tag{4}$$

   Converges in around 15 sec.
   Number of iterations : 50000
   Cost wrt original hypothesis : 0.9829469215
   Cost wrt learned hypothesis : 0.9831789793953184

4. Batch size 1000000 :

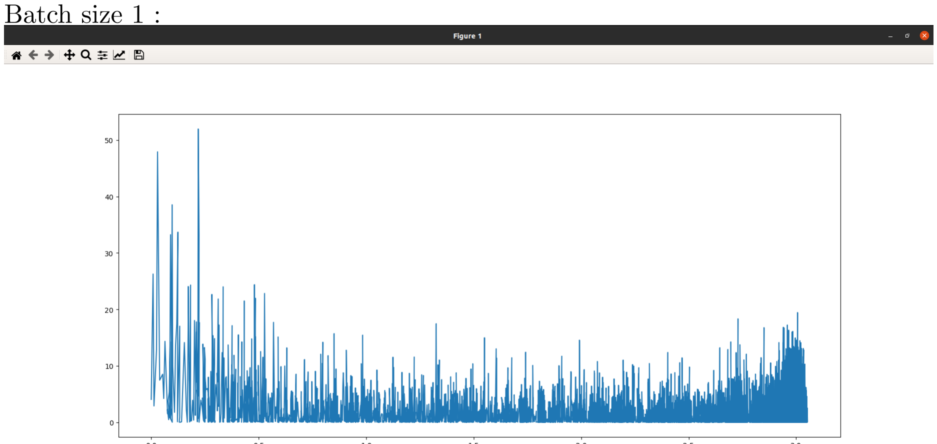
$$\theta = \begin{bmatrix} 2.9978612 \\ 1.00047121 \\ 1.99970272 \end{bmatrix} \tag{5}$$

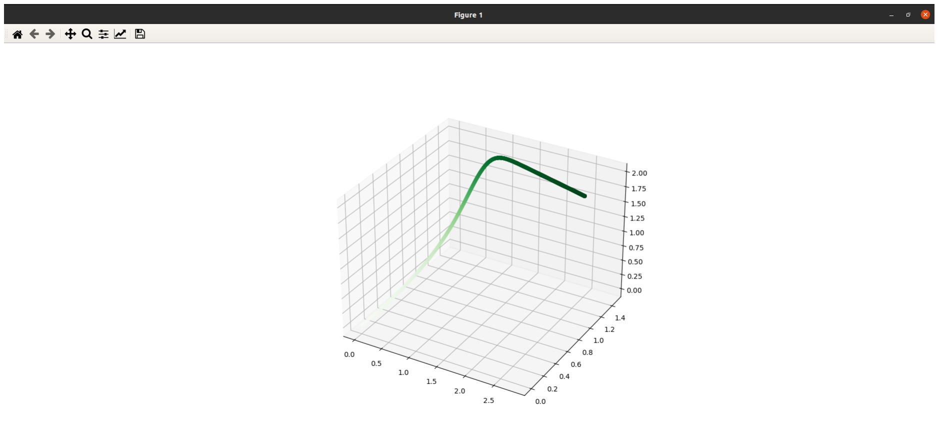
   Converges the slowest in around 23 min. Number of iterations : 50000
   Cost wrt original hypothesis : 0.9829469215
   Cost wrt learned hypothesis : 0.9829881137297302

Conclusion : The cost wrt original hypothesis is less than cost wrt learned hypothesis as expected since the learned hypothesis is obtained by stochastic gradient descent and the original hypothesis were the ones to generate the data.

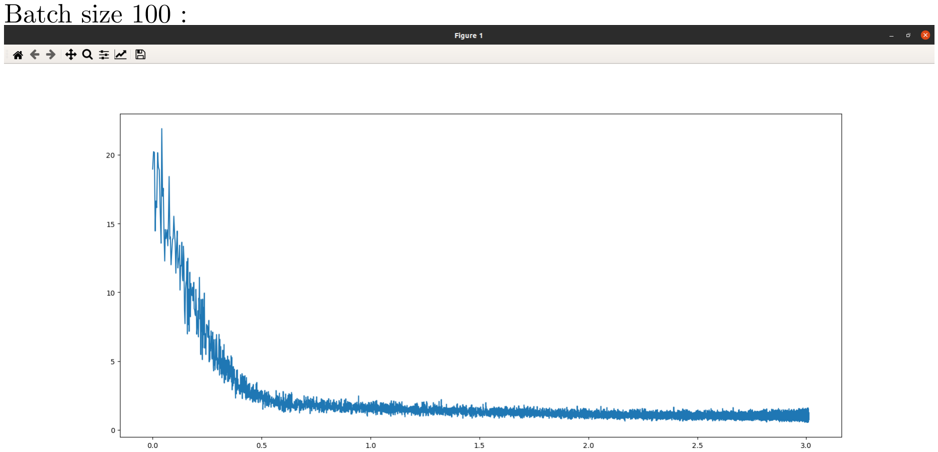
2.4 Part d

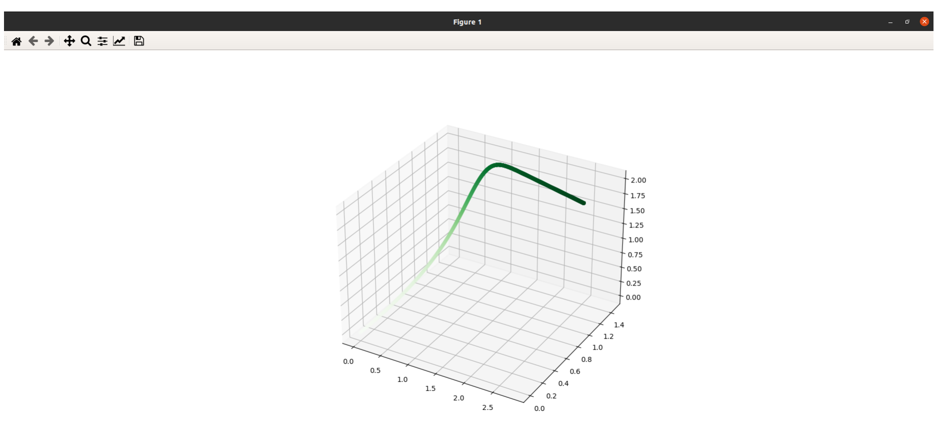
1. Batch size 1 :



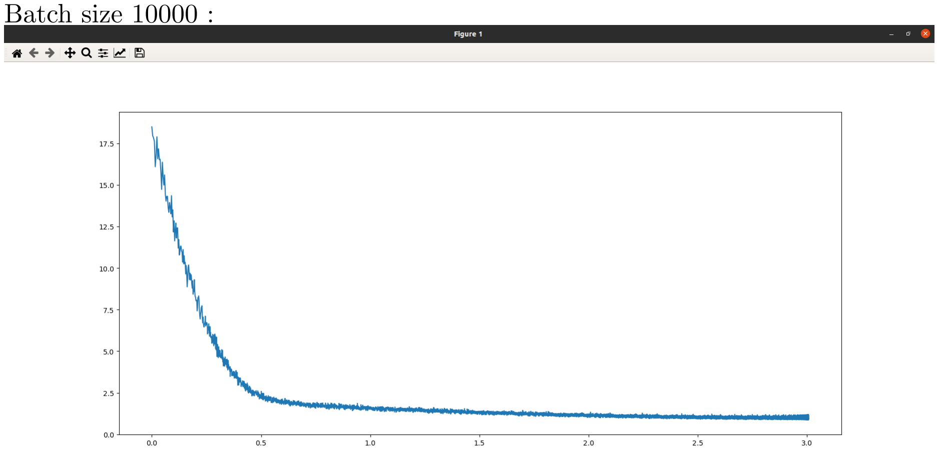


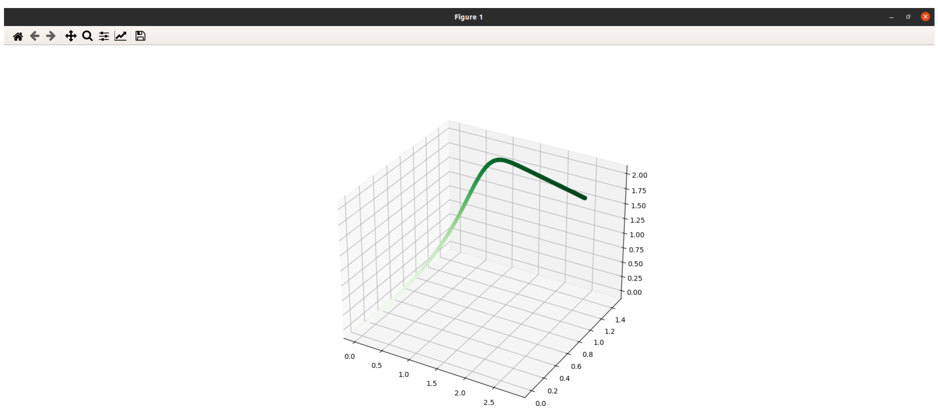
2. Batch size 100 :



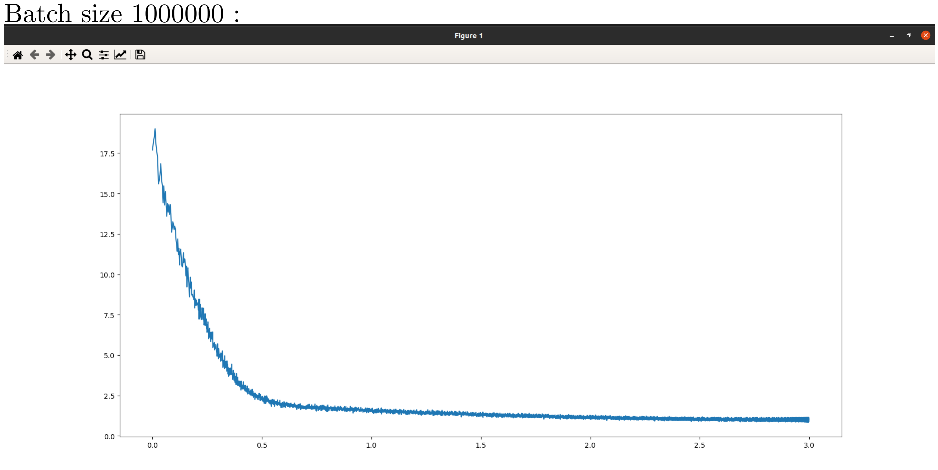


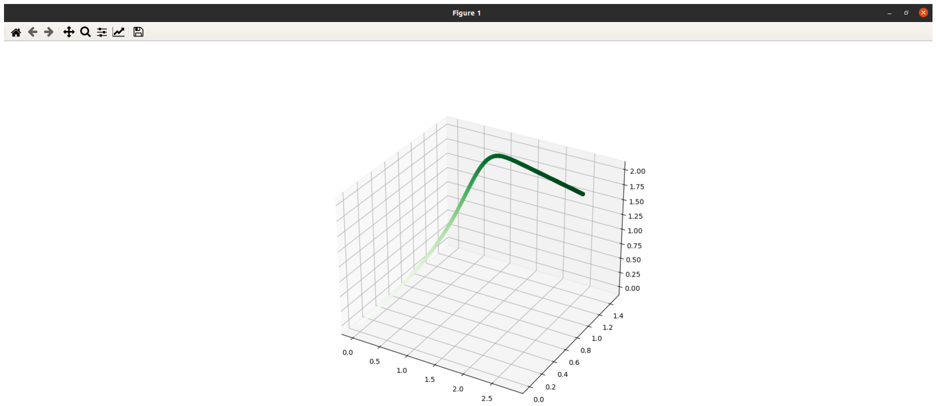
3. Batch size 10000 :





4. Batch size 1000000 :





Conclusion : For smaller batch sizes, we see the cost function moves about randomly for the initial iterations before converging finally hence the convergence curve is not smooth. But as batch sizes increase, we get a smooth descent towards the minimum cost function since higher batch size implies batch gradient descent, and the last one in the list has batch size = the total number of training examples hence it is the same as batch gradient descent where we consider all the examples together during convergence and also it takes the highest time out of all the four.

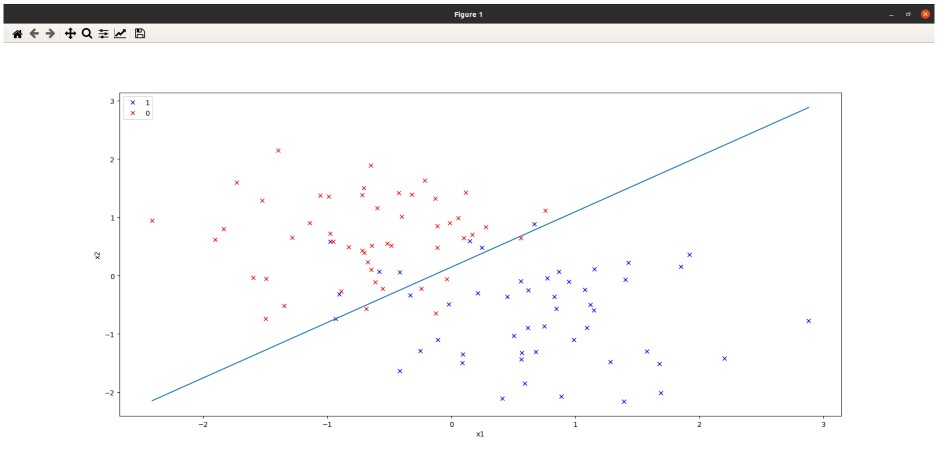
3 Part 3

3.1 Part a

Final learnt parameters :

$$\theta = \begin{bmatrix} 0.39676609 \\ 2.57387681 \\ -2.70953382 \end{bmatrix} \tag{6}$$

3.2 Part b



4 Part 4

4.1 Part a

Parameters :

$$\mu_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix} \tag{7}$$

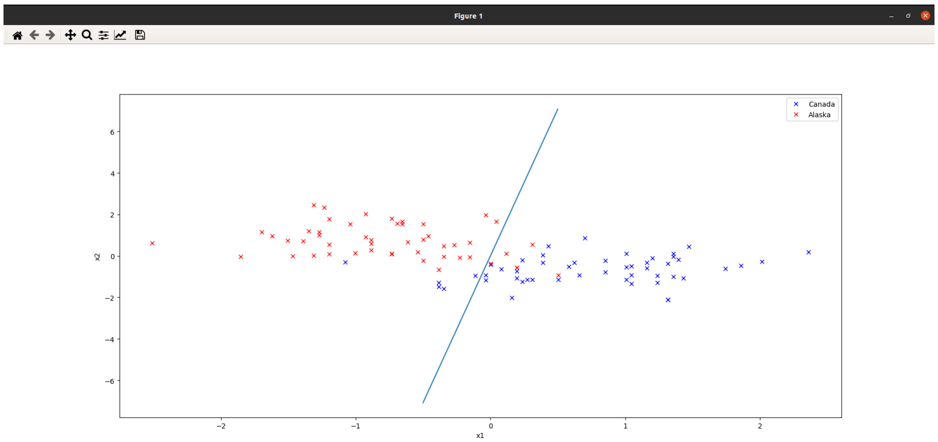
$$\tag{8}$$

$$\mu_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix} \tag{9}$$

$$\tag{10}$$

$$\sigma = \begin{bmatrix} 1.48233584 & 1.126314 \\ 1.126314 & 1.55748789 \end{bmatrix} \tag{11}$$

4.2 Part b



4.3 Part c

The linear decision boundary has been plotted in Part b above. The boundary equation is :

$$5.97x_1 - 0.42x_2 = 0 \tag{12}$$

4.4 Part d

Parameters :

$$\mu_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix} \tag{13}$$

$$\tag{14}$$

$$\mu_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix} \tag{15}$$

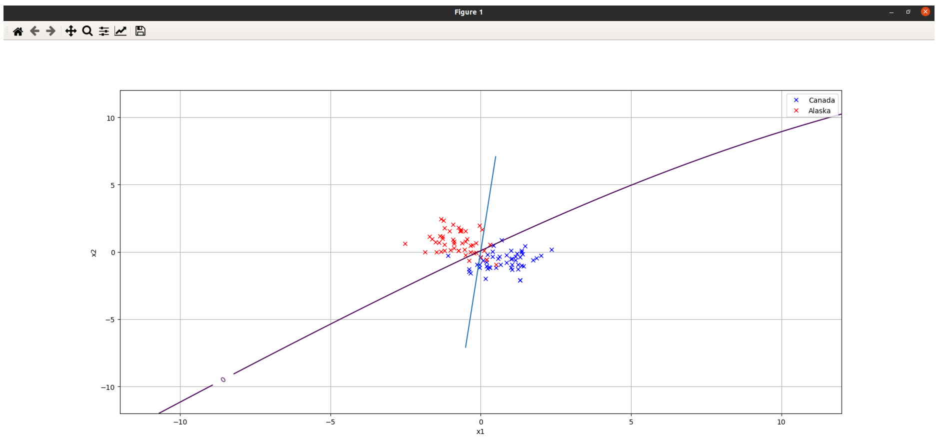
$$\tag{16}$$

$$\sigma_1 = \begin{bmatrix} 1.34811536 & 1.02753755 \\ 1.02753755 & 1.59059306 \end{bmatrix} \tag{17}$$

$$\tag{18}$$

$$\sigma_0 = \begin{bmatrix} 1.61655632 & 1.22509045 \\ 1.22509045 & 1.52438272 \end{bmatrix} \tag{19}$$

4.5 Part e



Here the brown line depicts the quadratic boundary. As we can see, it acts as a suitable decision boundary for the red and blue dots.

4.6 Part f

The quadratic boundary is a better fit for the data, since it divides more number of points correctly as compared to the linear boundary. The quadratic curve is better since the data is not known to have same variance for the red and blue points hence it provides a generalization, thus resulting in a better fit.