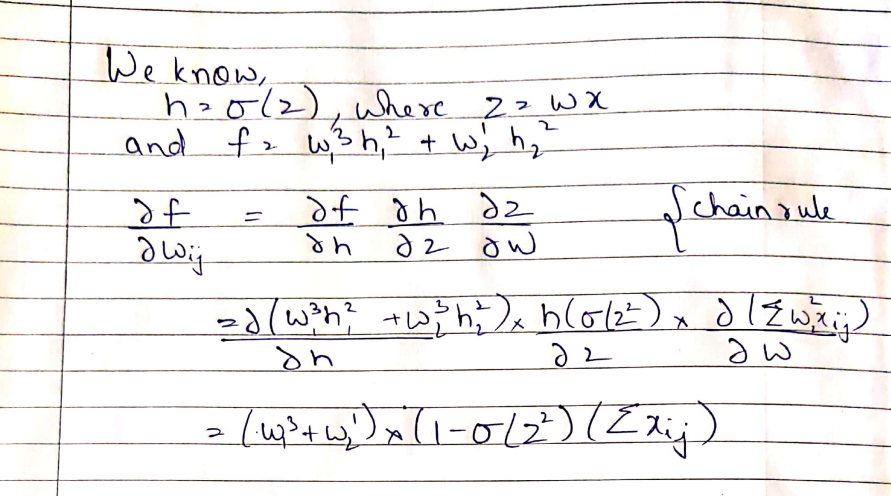
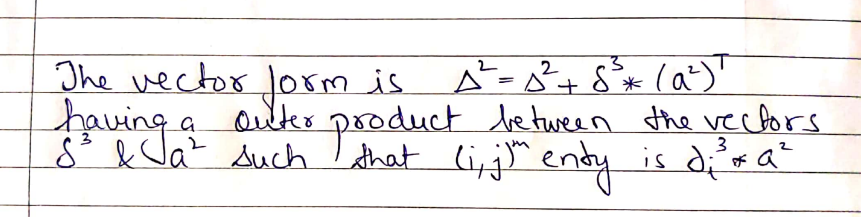
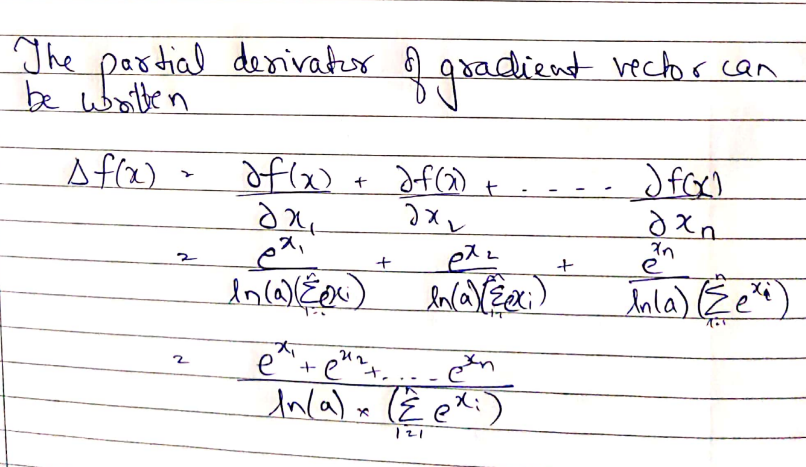
Solution 1.(a) 

Solution 1.(b) 

Solution 2. (a) & (b)

Solution 3.(a) Python code is inside the folder (SGD\_question\_3\_with\_optim.py)

Following are the result of changing learning rates without **torch optimizer**

Alpha= -0.1 , Loss: 0.000771 after 850 batches

Alpha=-0.2, Loss: 0.000547 after 9741 batches

Alpha=-0.3, Loss: 0.000562 after 2 batches

Alpha=-0.4, Never completes the program

Following are the result of changing learning rates with **torch optimizer (without momentum hyper parameter)**

Alpha= -0.01 , Loss: 0.000988 after 845 batches

Alpha=-0.1, Loss: 0.000998 after 114 batches

Alpha=-0.2, Loss: 0.000964 after 3864 batches

Alpha=-0.3, Loss: 0.000798 after 58983 batches

Alpha=-0.4, Never completes the program

Conclusion:

* Using 0.1 as learning rate, the network slowly and steadily converges in the direction of the minima. Faster in case of torch.optim
* Using 0.2 as learning rate, the gradient overshoots the minima and keeps on re-correcting its path to finally reach after too many epochs. Faster in case of torch.optim
* Using 0.3 as learning rate, the gradient by pure luck reaches the minima in its second epoch itself, indicating that the minima resides after a very sharp steep, and 0.3 learning rate helps directly reach to the bottom. In torch.optim it overshoots the minima by a further margin and takes too many epoch to run
* Using 0.4 as learning rate, the gradient keeps on missing the slope and never really finds the right direction to fall towards the slope, same happened in the case of torch.optim

Solution 3(b). Python code inside folder (SGD\_question\_3 (b).py)

Model Parameters:

* Fixed epochs at 50K
* Learning rate of 0.01
* Adam grad optimization
* Linear regression with mini batch SGD with a batch size of 5

Training result:

* Loss: 0.793707 after 50001 batches
* Learned function: y = +0.68 x^1 +0.93 x^1 +31.82

Test result:

* [39.5984], [43.2388], [48.7342]

Solution to 3(c), 4 (a), 4(b) and 4(c) are provided in jupyter notebook (Please find it in the folder itself)

Assignment\_1\_question 3.(c).ipynb

Assignment\_4(a).ipynb

Assignment\_4(b).ipynb

Assignment\_4(c) with adagrad.ipynb

Assignment\_4(c) with LBFGS.ipynb