AssignmentReport-Group199

February 9, 2024

1 Assignment 1 Report

This is an outline for your report to ease the amount of work required to create your report. Jupyter notebook supports markdown, and I recommend you to check out this cheat sheet. If you are not familiar with markdown.

Before delivery, **remember to convert this file to PDF**. You can do it in two ways: 1. Print the webpage (ctrl+P or cmd+P) 2. Export with latex. This is somewhat more difficult, but you'll get somehwat of a "prettier" PDF. Go to File -> Download as -> PDF via LaTeX. You might have to install nbconvert and pandoc through conda; `conda install nbconvert pandoc`.

2 Task 1

2.1 task 1a)

TDT 4265 Assignment 1

$\boxed{1}$ a) Start with cost function

$$C^n(w) = -\left(y^n \ln(\hat{y}^n) + (1-y^n)\ln(1-\hat{y}^n)\right)$$

First solve

$$\frac{\partial C^n}{\partial \hat{y}^n} = -\frac{y^n}{\hat{y}^n} + \frac{1-y^n}{1-\hat{y}^n}$$

Then since $\hat{y}^n = f(x^n)$ we use the chain rule:

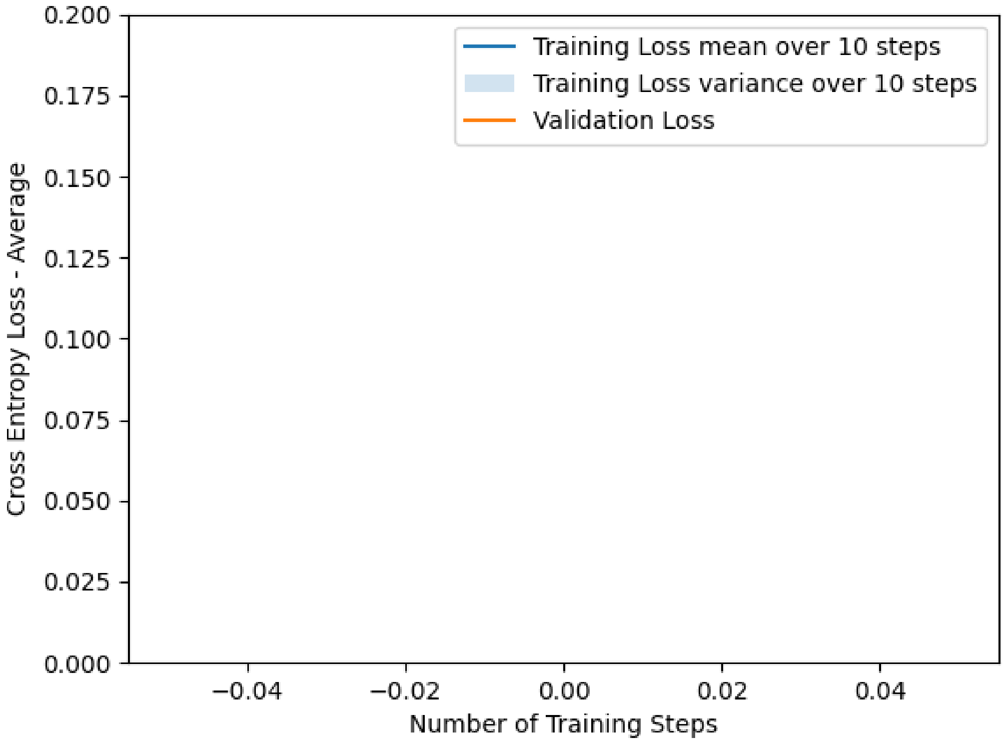
$$\frac{\partial C^n(w)}{\partial w_i} = \frac{\partial C^n}{\partial \hat{y}^n}\frac{\partial \hat{y}^n}{\partial w_i} = -\left(\frac{y^n}{\hat{y}^n} - \frac{1-y^n}{1-\hat{y}^n}\right) x_i^n \hat{y}^n(1-\hat{y}^n)$$

$$= -\left(y^n(1-\hat{y}^n) - \hat{y}^n(1-y^n)\right) x_i^n$$

$$= -(y^n - \hat{y}^n)x_i^n$$

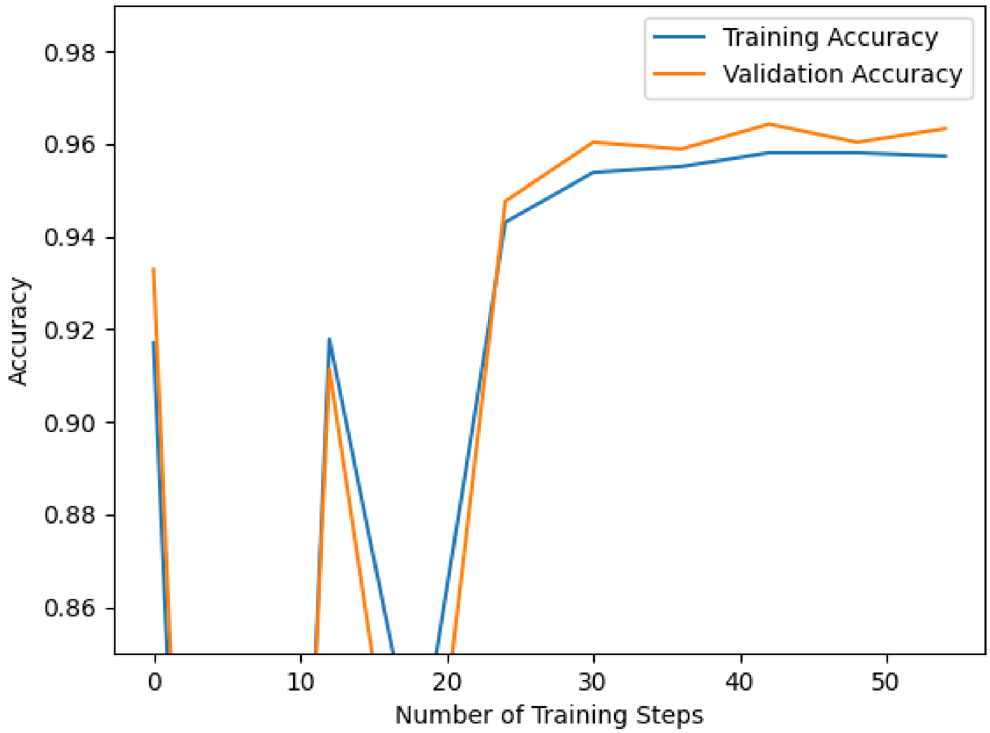
3 Task 2

3.1 Task 2b)



I have been beating my head against a wall for hours trying to figure out why i cant get this working properly, but i cant figure out why. I suspect something is wrong with either forwards, backwards, or the loss function, but whatever i try i cant seem to get it working

3.2 Task 2c)

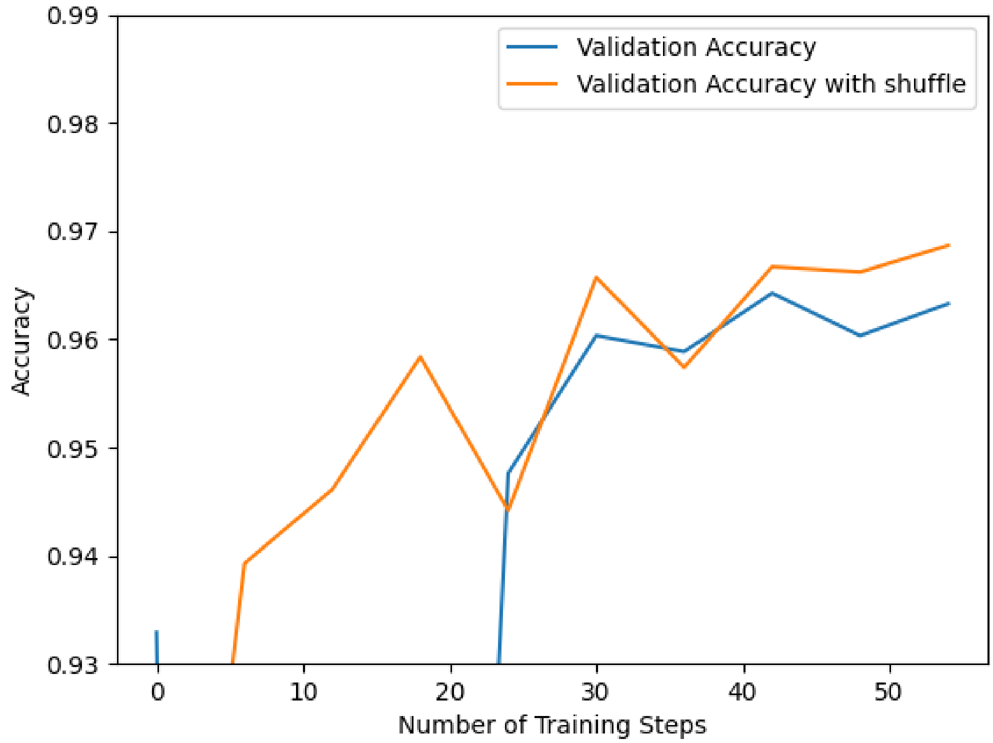


3.3 Task 2d)

It stops after only two epoch which seem low, and is propbably caused by the same problem that makes the rest of the model not work properly

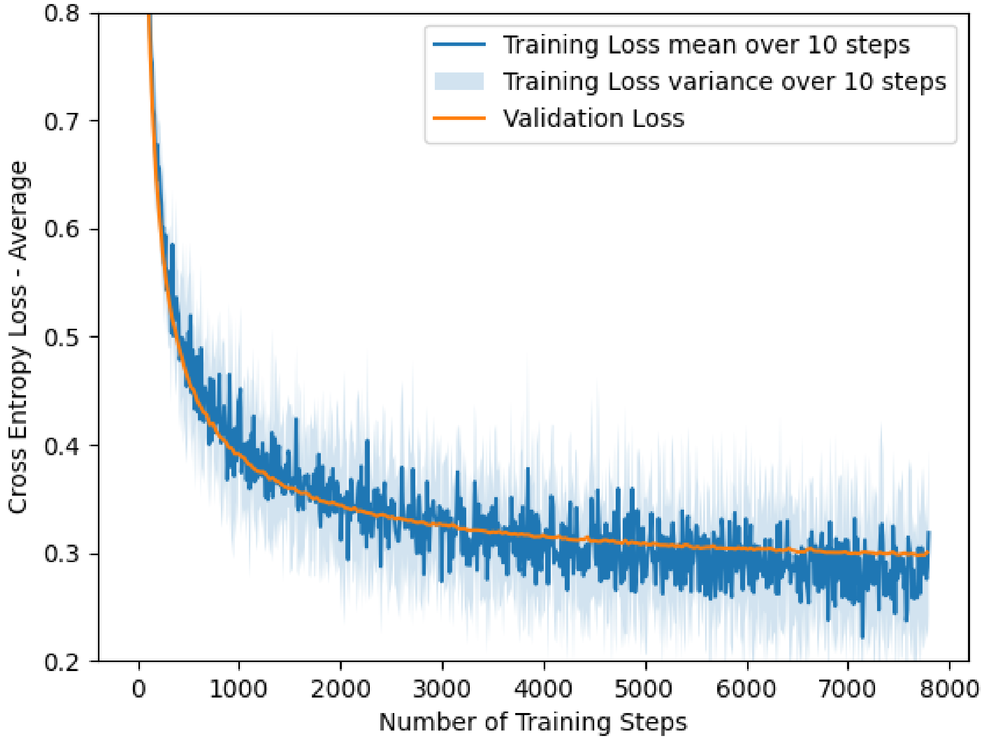
3.4 Task 2e)

Again i dont think the plot is very informative. However, i think the graph would be smoother because shuffling would mean more representative trainig batches of the entire dataset, and that the model wont learn local patterns that doesnt actually exist in the data.

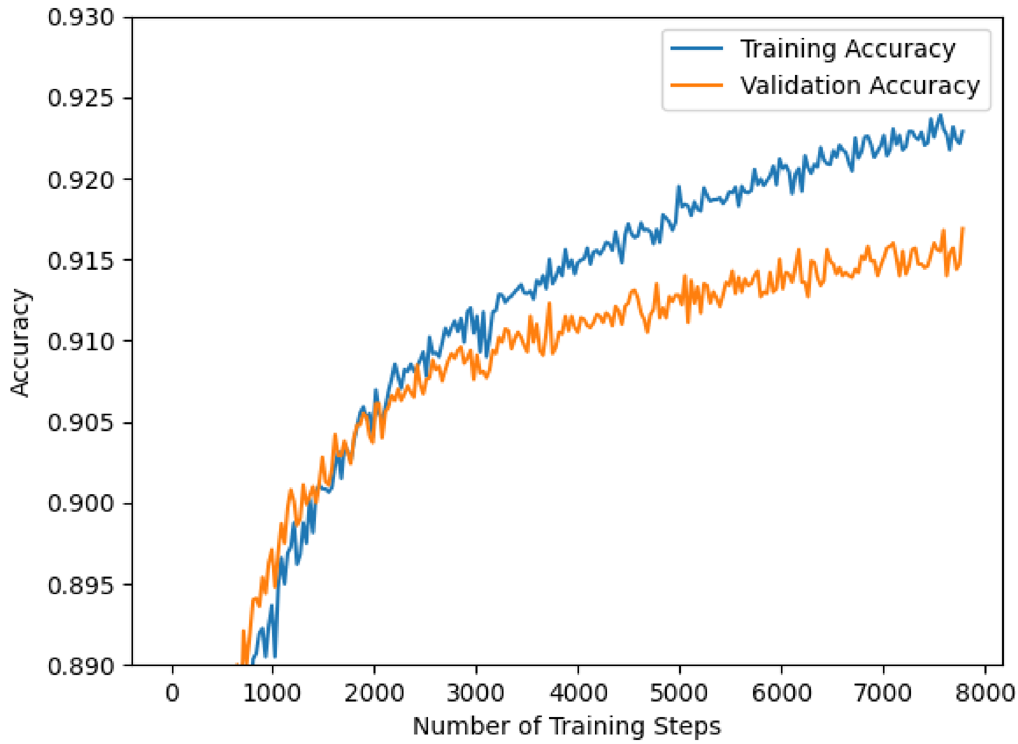

Validation Accuracy
Validation Accuracy with shuffle
Accuracy
Number of Training Steps
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0
10
20
30
40
50

4 Task 3

4.1 Task 3b)



4.2 Task 3c)



4.3 Task 3d)

The training accuracy gets significantly larger than the validation accuracy, which could be a sign of overfitting

5 Task 4

5.1 Task 4a)

The cost function with L2 regularization is

$$J(w) = C(w) + \lambda \sum ||w||^2.$$

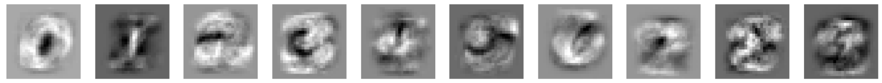
The gradient with respect to the weights is

$$\frac{\partial J(w)}{\partial w} = \frac{\partial C(w)}{\partial w} + 2\lambda w,$$

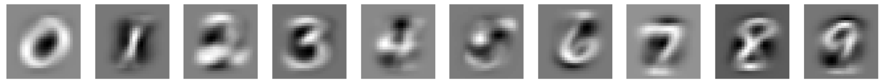
5.2 Task 4b)

A high lambda value will reduce the value of the weights, and makes the model focus on the most importan features. Therefore the visualization appear smoother

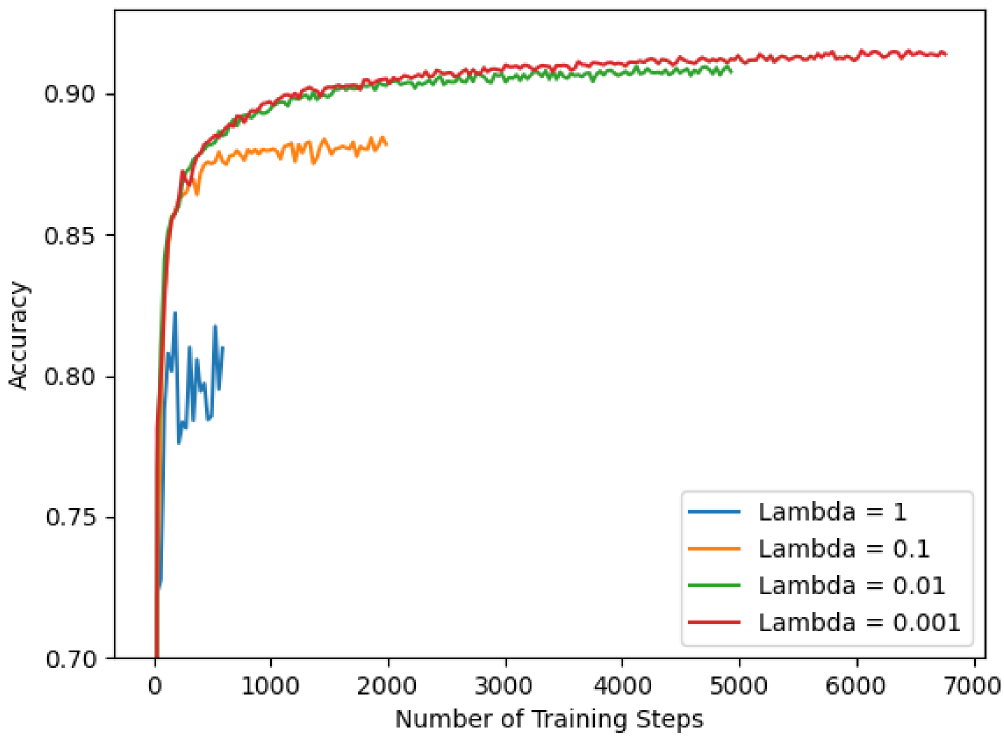
With lambda = 0



With lambda = 1



5.3 Task 4c)

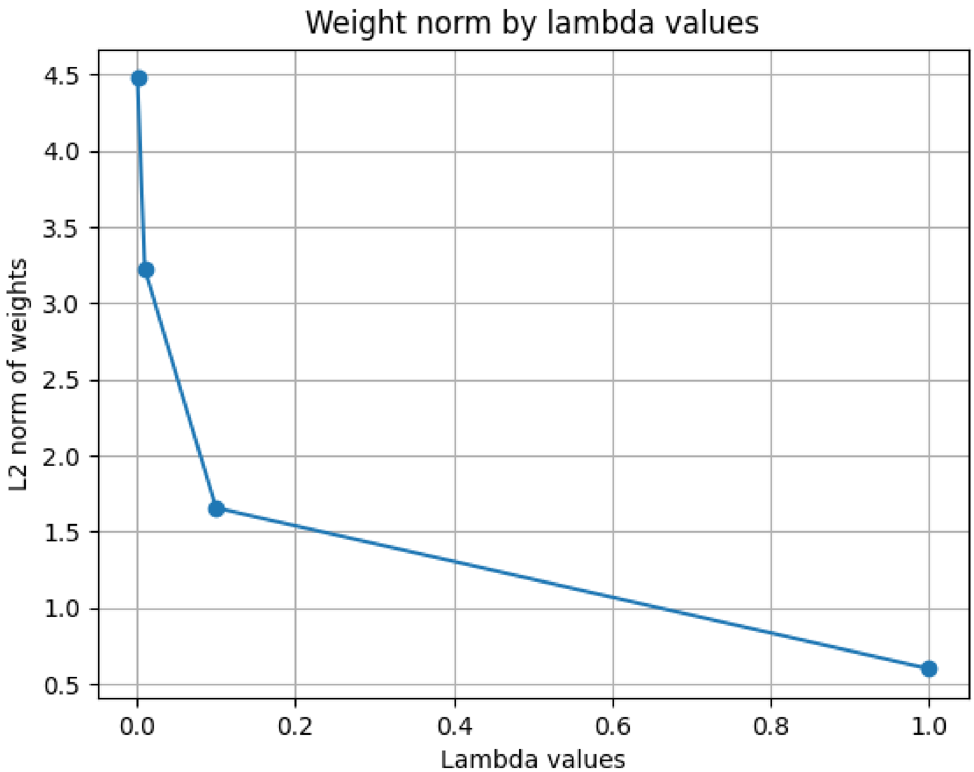


5.4 Task 4d)

I think its because large lambdas might make the model underfit, but im not sure if i really understand why

5.5 Task 4e)

There is an inverse relation between the size of lambda and the size of the weights


Weight norm by lambda values
L2 norm of weights
Lambda values
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.2
0.4
0.6
0.8
1.0