AssignmentReport-Group1

February 6, 2021

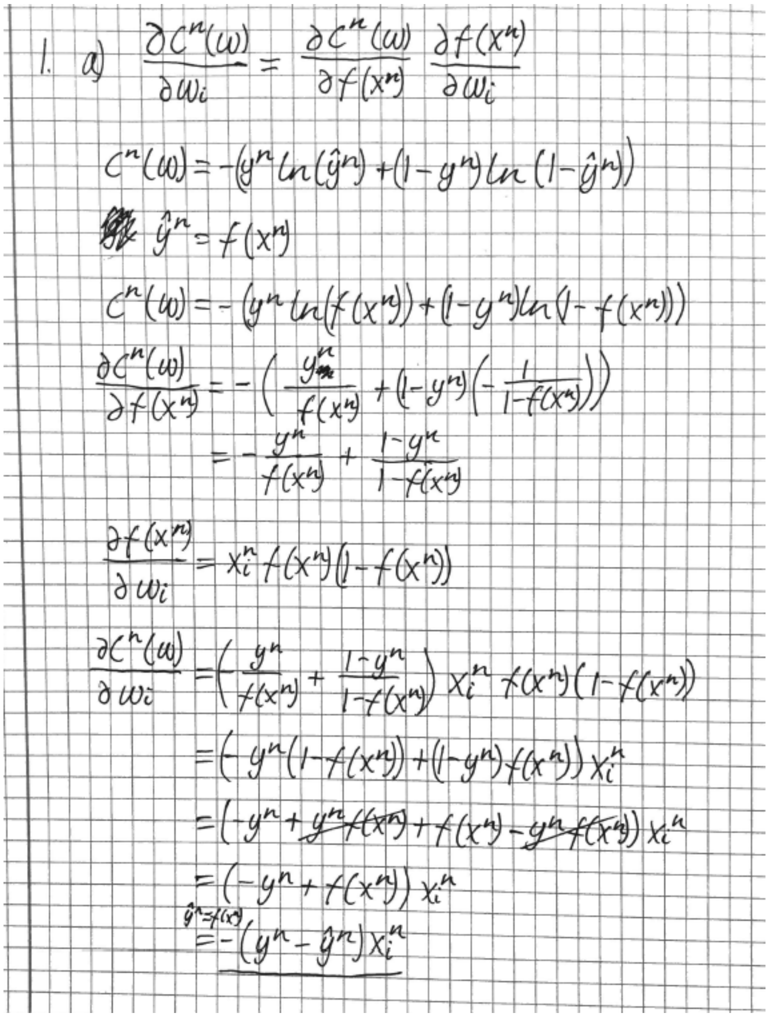
1 Assignment 1 Report

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)



1. a) $\frac{\partial C^n(w)}{\partial w_i} = \frac{\partial C^n(w)}{\partial f(x^n)} \frac{\partial f(x^n)}{\partial w_i}$

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

$\hat{y}^n = f(x^n)$

$C^n(w) = -(y^n \ln(f(x^n)) + (1-y^n)\ln(1-f(x^n)))$

$$\frac{\partial C^n(w)}{\partial f(x^n)} = -\left(\frac{y^n}{f(x^n)} + (1-y^n)\left(-\frac{1}{1-f(x^n)}\right)\right)$$

$$= -\frac{y^n}{f(x^n)} + \frac{1-y^n}{1-f(x^n)}$$

$$\frac{\partial f(x^n)}{\partial w_i} = x_i^n f(x^n)(1-f(x^n))$$

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

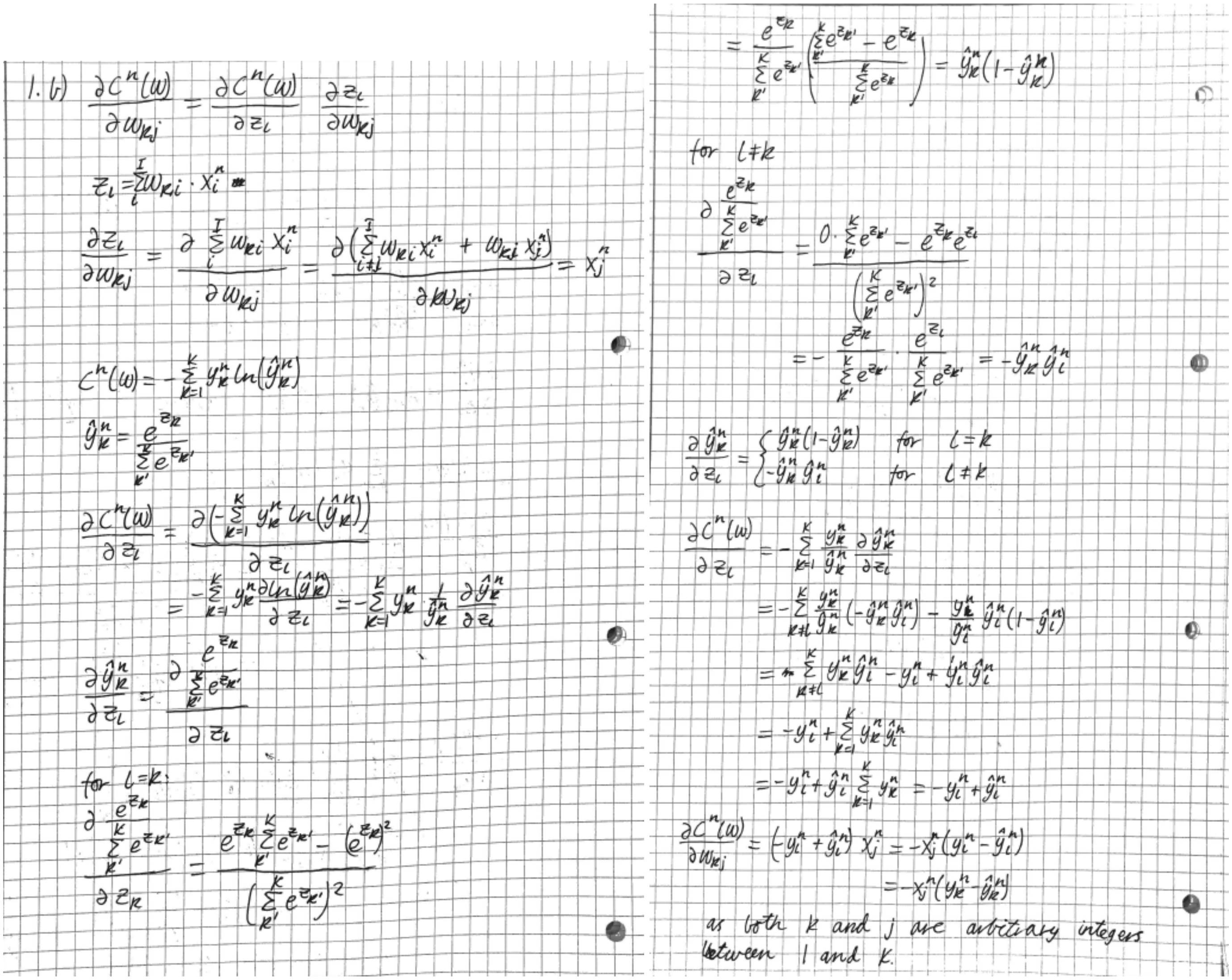
$$= (-y^n(1-f(x^n)) + (1-y^n)f(x^n))\, x_i^n$$

$$= (-y^n + \cancel{y^n f(x^n)} + f(x^n) - \cancel{y^n f(x^n)})\, x_i^n$$

$$= (-y^n + f(x^n))\, x_i^n$$

$$\overset{\hat{y}^n = f(x^n)}{=} \underline{-(y^n - \hat{y}^n)\, x_i^n}$$

2.2 task 1b)



1. b) $$\frac{\partial C^n(w)}{\partial w_{kj}} = \frac{\partial C^n(w)}{\partial z_l} \frac{\partial z_l}{\partial w_{kj}}$$

$$z_l = \sum_i^I w_{ki} \cdot x_i^n$$

$$\frac{\partial z_l}{\partial w_{kj}} = \frac{\partial \sum_i^I w_{ki} x_i^n}{\partial w_{kj}} = \frac{\partial \left(\sum_{i \neq j}^I w_{ki} x_i^n + w_{kj} x_j^n \right)}{\partial w_{kj}} = x_j^n$$

$$C^n(w) = -\sum_{k=1}^K y_k^n \ln(\hat{y}_k^n)$$

$$\hat{y}_k^n = \frac{e^{z_k}}{\sum_{k'}^K e^{z_{k'}}}$$

$$\frac{\partial C^n(w)}{\partial z_l} = \frac{\partial \left(-\sum_{k=1}^K y_k^n \ln(\hat{y}_k^n) \right)}{\partial z_l}$$

$$= -\sum_{k=1}^K y_k^n \frac{\partial \ln(\hat{y}_k^n)}{\partial z_l} = -\sum_{k=1}^K y_k^n \frac{1}{\hat{y}_k^n} \frac{\partial \hat{y}_k^n}{\partial z_l}$$

$$\frac{\partial \hat{y}_k^n}{\partial z_l} = \frac{\partial \frac{e^{z_k}}{\sum_{k'}^K e^{z_{k'}}}}{\partial z_l}$$

for $l = k$:

$$\frac{\partial \frac{e^{z_k}}{\sum_{k'}^K e^{z_{k'}}}}{\partial z_k} = \frac{e^{z_k} \sum_{k'}^K e^{z_{k'}} - (e^{z_k})^2}{\left(\sum_{k'}^K e^{z_{k'}} \right)^2}$$

$$= \frac{e^{z_k}}{\sum_{k'}^K e^{z_{k'}}} \left(\frac{\sum_{k'}^K e^{z_{k'}} - e^{z_k}}{\sum_{k'}^K e^{z_k}} \right) = \hat{y}_k^n (1 - \hat{y}_k^n)$$

for $l \neq k$

$$\frac{\partial \frac{e^{z_k}}{\sum_{k'}^K e^{z_{k'}}}}{\partial z_l} = \frac{0 \cdot \sum_{k'}^K e^{z_{k'}} - e^{z_k} e^{z_l}}{\left(\sum_{k'}^K e^{z_{k'}} \right)^2}$$

$$= -\frac{e^{z_k}}{\sum_{k'}^K e^{z_{k'}}} \cdot \frac{e^{z_l}}{\sum_{k'}^K e^{z_{k'}}} = -\hat{y}_k^n \hat{y}_l^n$$

$$\frac{\partial \hat{y}_k^n}{\partial z_l} = \begin{cases} \hat{y}_k^n (1 - \hat{y}_k^n) & \text{for} \quad l = k \\ -\hat{y}_k^n \hat{y}_l^n & \text{for} \quad l \neq k \end{cases}$$

$$\frac{\partial C^n(w)}{\partial z_l} = -\sum_{k=1}^K \frac{y_k^n}{\hat{y}_k^n} \frac{\partial \hat{y}_k^n}{\partial z_l}$$

$$= -\sum_{k \neq l}^K \frac{y_k^n}{\hat{y}_k^n} (-\hat{y}_k^n \hat{y}_l^n) - \frac{y_k^n}{\hat{y}_l^n} \hat{y}_l^n (1 - \hat{y}_l^n)$$

$$= \sum_{k \neq l}^K y_k^n \hat{y}_l^n - y_l^n + y_l^n \hat{y}_l^n$$

$$= -y_l^n + \sum_{k=1}^K y_k^n \hat{y}_l^n$$

$$= -y_l^n + \hat{y}_l^n \sum_{k=1}^K y_k^n = -y_l^n + \hat{y}_l^n$$

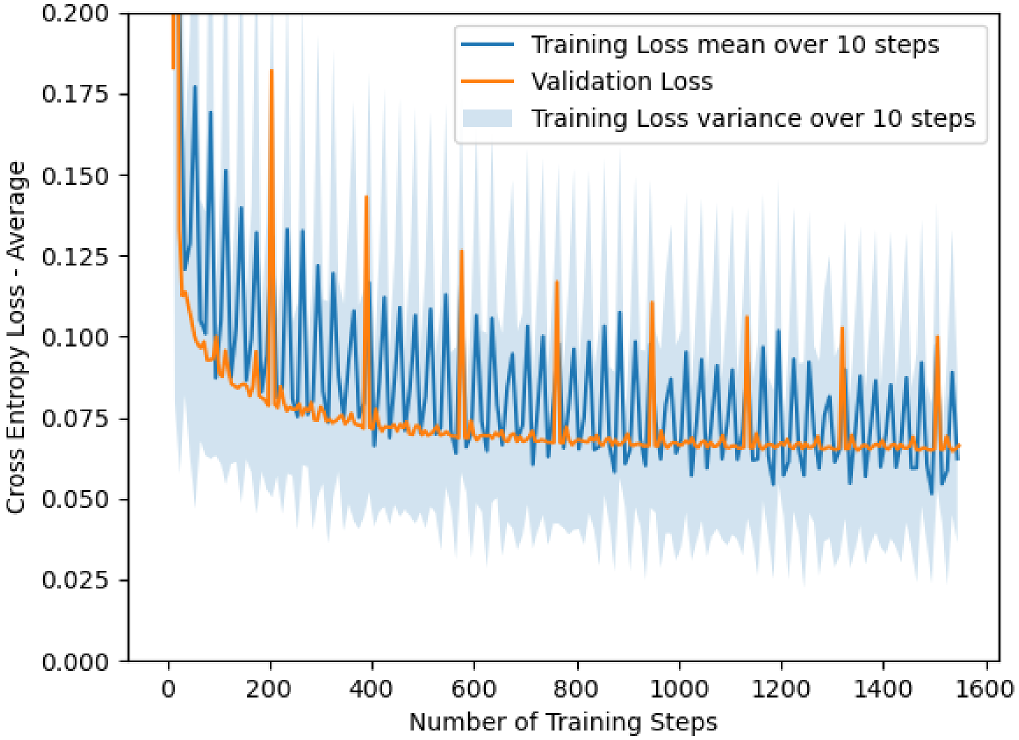
$$\frac{\partial C^n(w)}{\partial w_{kj}} = (-y_l^n + \hat{y}_l^n) x_j^n = -x_j^n (y_l^n - \hat{y}_l^n)$$

$$= -x_j^n (y_k^n - \hat{y}_k^n)$$

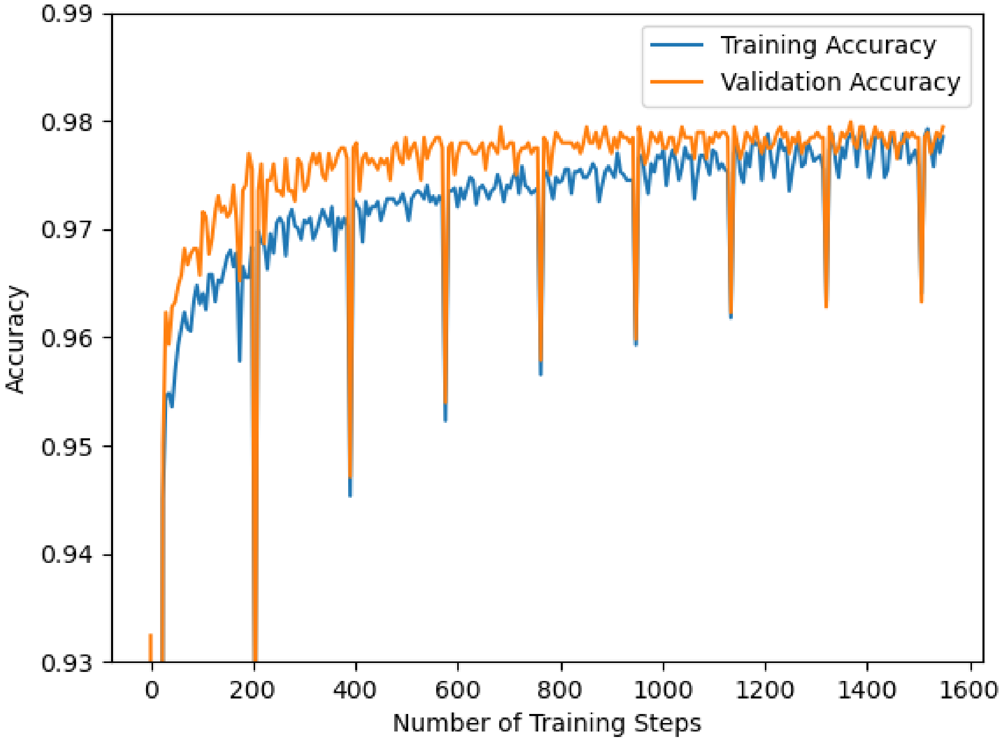
as both k and j are arbitiary integers between 1 and K.

3 Task 2

3.1 Task 2b)



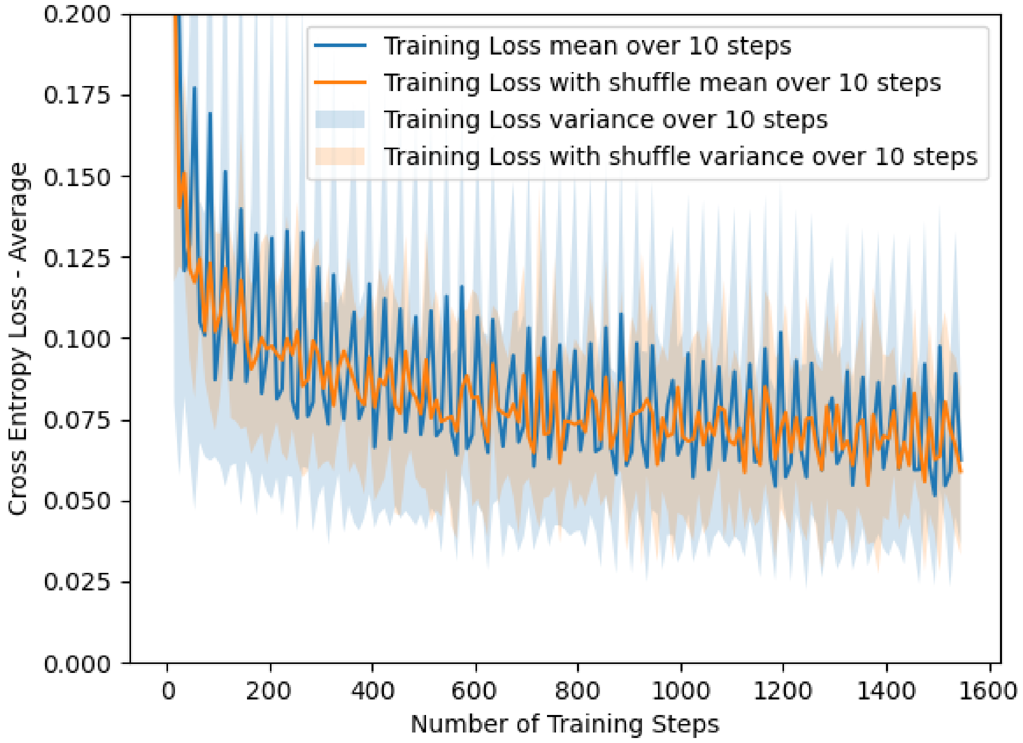
3.2 Task 2c)

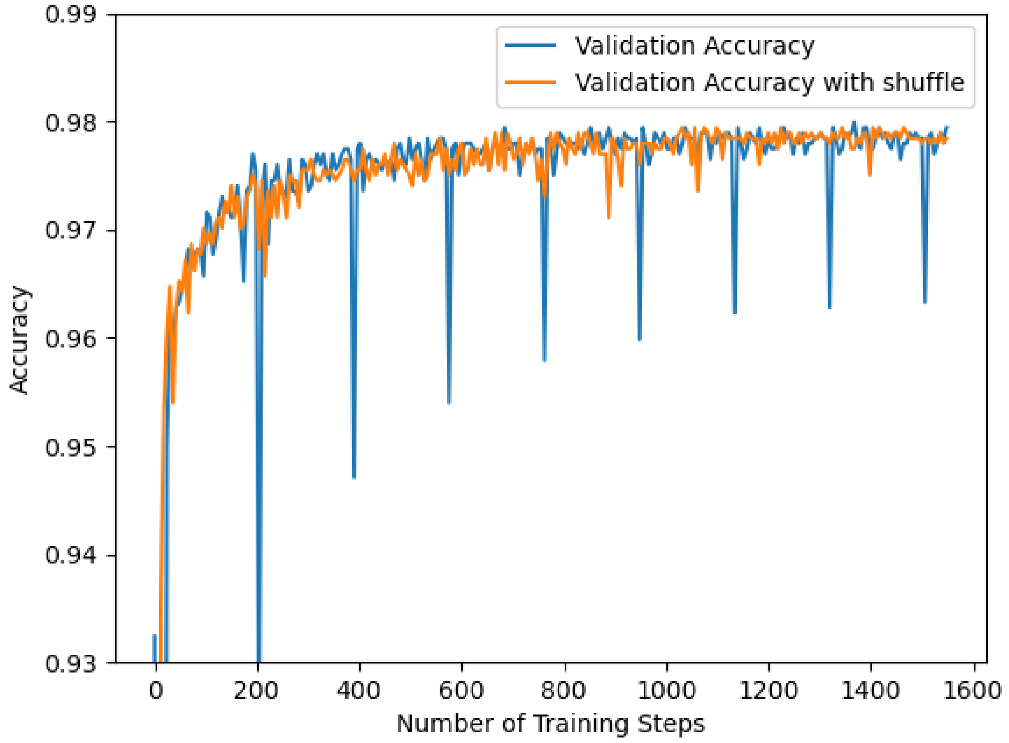


3.3 Task 2d)

Early stopping kicks in after 4 epochs.

3.4 Task 2e)


0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Cross Entropy Loss - Average
0
200
400
600
800
1000
1200
1400
1600
Number of Training Steps
Training Loss mean over 10 steps
Training Loss with shuffle mean over 10 steps
Training Loss variance over 10 steps
Training Loss with shuffle variance over 10 steps

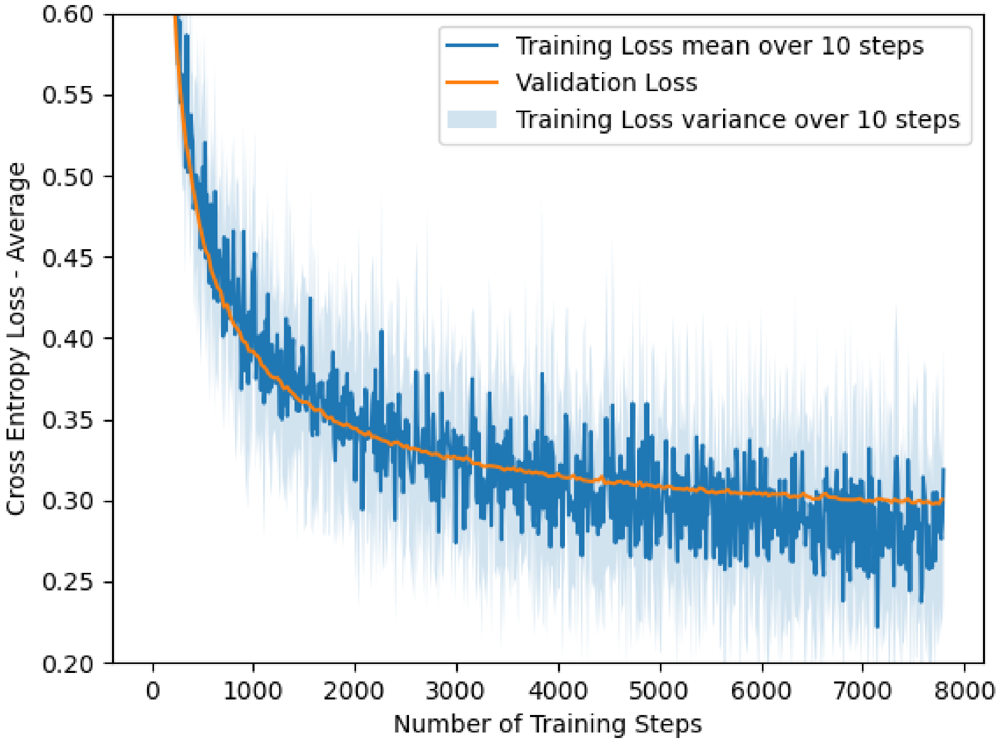

0.99
0.98
0.97
0.96
0.95
0.94
0.93
Accuracy
0
200
400
600
800
1000
1200
1400
1600
Number of Training Steps
Validation Accuracy
Validation Accuracy with shuffle

By shuffling the dataset we avoid the model being trained for only 2s at the start of every epoch, which would result in an accuracy drop when we hit the 3s until the model learns how to correctly classify both. With a shuffled dataset we encounter 2s and 3s randomly and the model learns that there exists more than one number to classify.

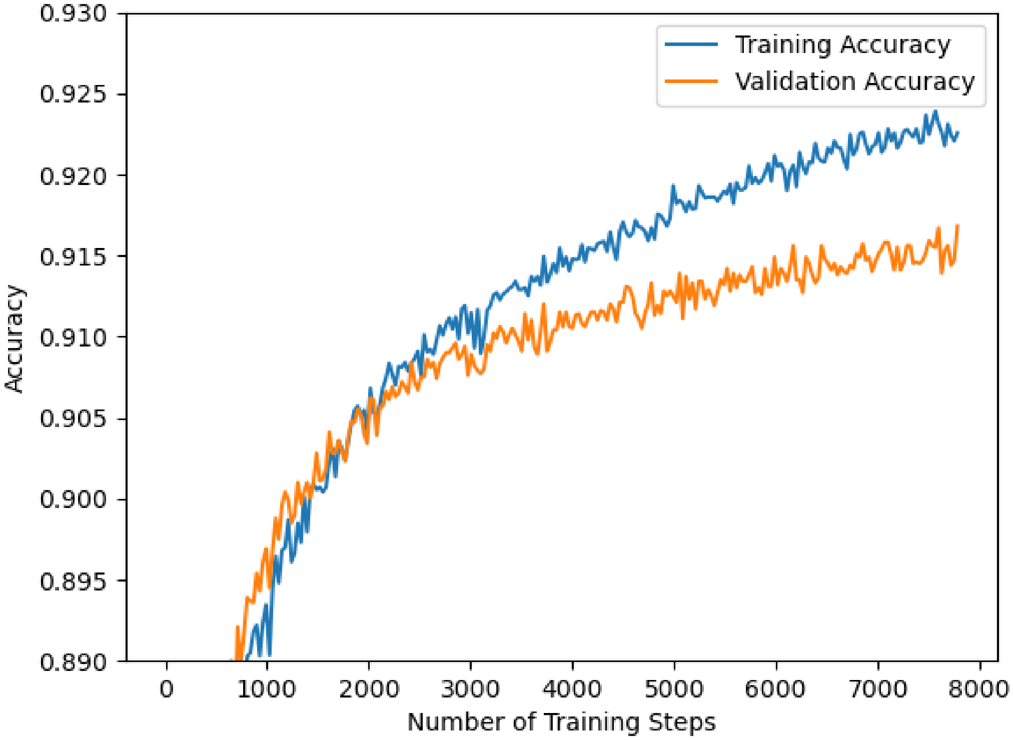
In general shuffling helps reduce variance and overfitting by ensuring the data used is a representation of the whole dataset. In the MNIST database a sorted batch of data will never include all 10 digits and thereby make the model overfitted to the digits represented and completely oblivious to other digits. While a shuffled batch of data will almost always contain all digits such that the model trains on a more general batch.

4 Task 3

4.1 Task 3b)



4.2 Task 3c)



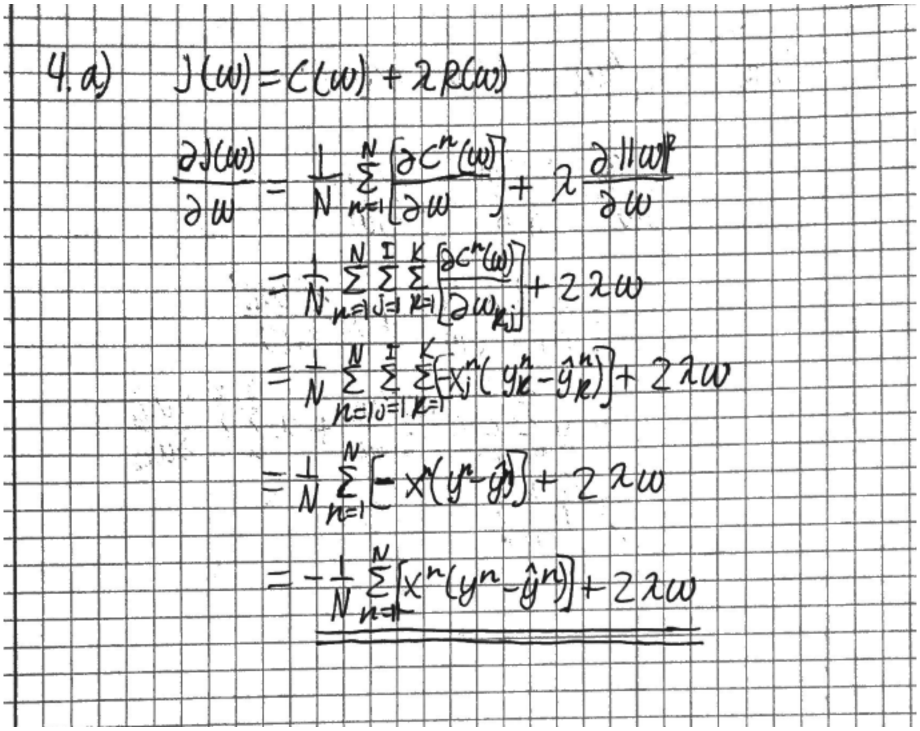
4.3 Task 3d)

The training accuracy is rising faster than the validation accuracy, indicating some overfitting. The validation accuracy is still increasing which implies more training could give a higher validation accuracy, but this higher accuracy would cost a lot of computational time compared to earlier training where the training and validation accuracies were rising almost identically.

We can also see that the validation loss is plateauing faster than the training loss, which signals the same as the plateauing in validation accuracy compared to training accuracy.

5 Task 4

5.1 Task 4a)



4.a) $J(w) = C(w) + \lambda R(w)$

$$\frac{\partial J(w)}{\partial w} = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{\partial C^n(w)}{\partial w}\right] + \lambda \frac{\partial \|w\|^2}{\partial w}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{I}\sum_{k=1}^{K}\left[\frac{\partial C^n(w)}{\partial w_{kj}}\right] + 2\lambda w$$

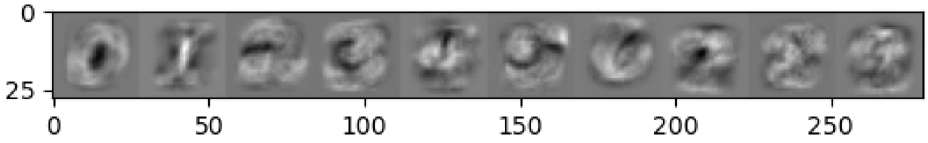
$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{I}\sum_{k=1}^{K}\left[-x_j^n(y_k^n - \hat{y}_k^n)\right] + 2\lambda w$$

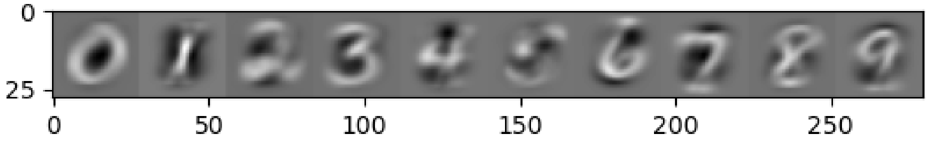
$$= \frac{1}{N}\sum_{n=1}^{N}\left[-x^n(y^n - \hat{y}^n)\right] + 2\lambda w$$

$$= \underline{\underline{-\frac{1}{N}\sum_{n=1}^{N}\left[x^n(y^n - \hat{y}^n)\right] + 2\lambda w}}$$

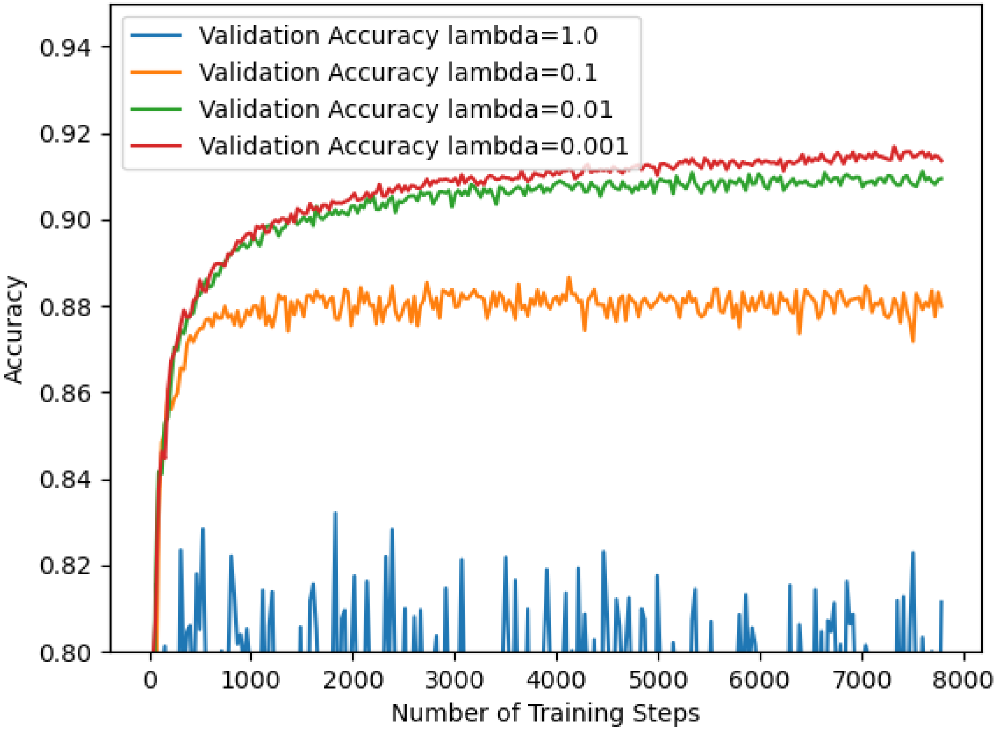
5.2 Task 4b)

The L2 regularization reduces the model complexity which results in less of the noise from the dataset leaking through the model. The reduction of model complexity increases bias but reduces noise in the bias/noise tradeoff.


0
25
0
50
100
150
200
250


0
25
0
50
100
150
200
250

5.3 Task 4c)



The zoom for this plot is a bit off, as the plot with edited range for ylim did not have time to finish running, same as task4c.

5.4 Task 4d)

More regularization reduces complexity which trades less noise for more bias, which might reduce accuracy, especially for high regularization constants where more bias is introduced, ie. lambda = 1.

5.5 Task 4e)

Could not finish running the code in time for the submission deadline, started running at about 23:00 but had not yet finished at 23:55 so that i could get the final plots into the report. Code for plotting the L2 norms is included in the code zip.

Also had to download Tex to deliver jupyter as pdf so my estimate of 5 min to be able to deliver was too short. Would have delivered 1 hr earlier if i knew code would not finish in the timeframe.

```
[ ]:
```