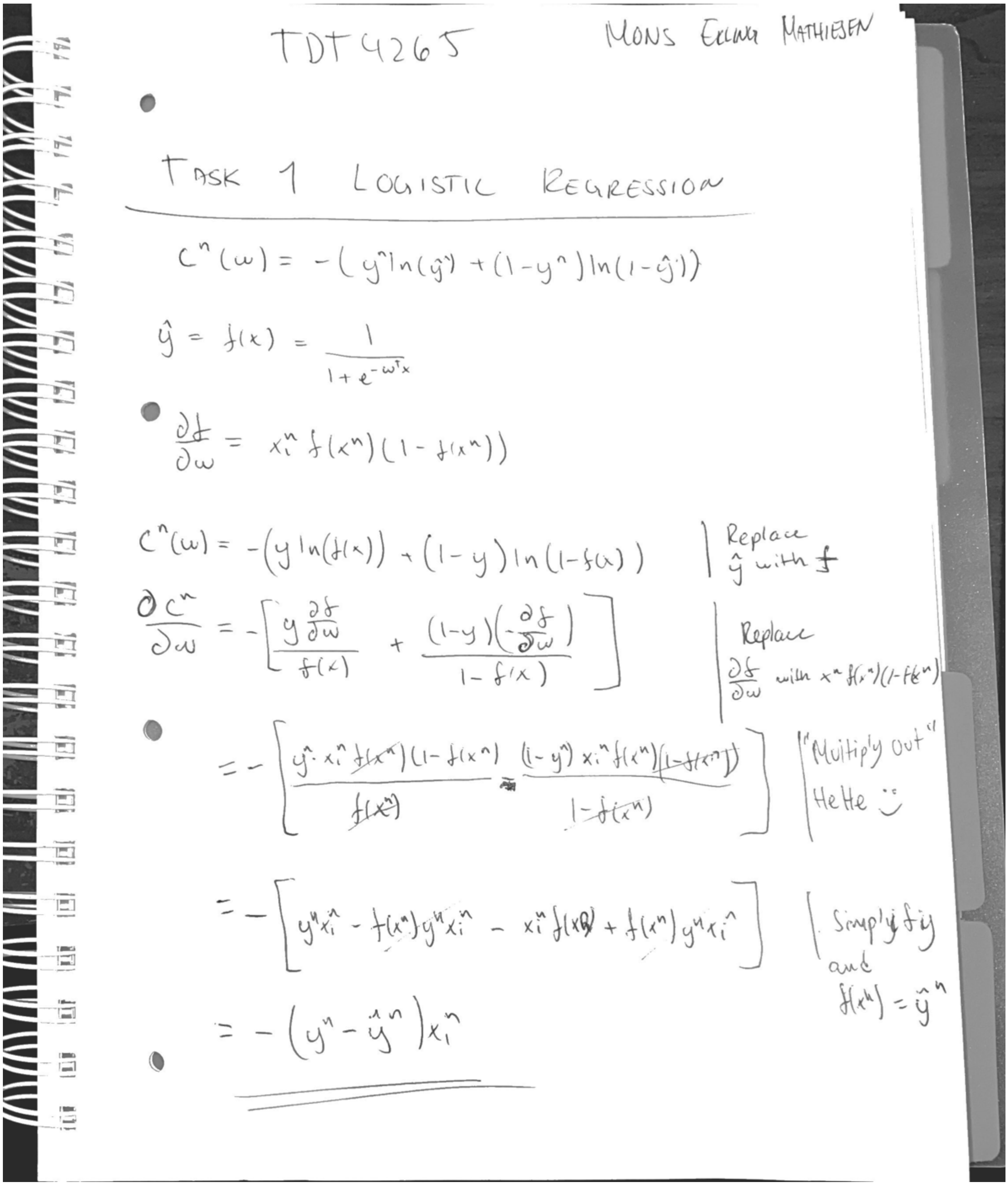
Assignment 1 Report

Task 1

Task 1a)



TDT4265 Mons Erling Mathiesen

Task 1 Logistic Regression

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

$$\hat{y} = f(x) = \frac{1}{1+e^{-w^T x}}$$

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

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

Replace $\hat{y}$ with f

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

Replace $\frac{\partial f}{\partial w}$ with $x^n f(x^n)(1-f(x^n))$

$$= -\left[\frac{y^n x_i^n \cancel{f(x^n)}(1-f(x^n))}{\cancel{f(x^n)}} - \frac{(1-y^n)x_i^n f(x^n)\cancel{(1-f(x^n))}}{\cancel{1-f(x^n)}}\right]$$

"Multiply out" Hehe :)

$$= -\left[y^n x_i^n - f(x^n)y^n x_i^n - x_i^n f(x^n) + f(x^n)y^n x_i^n\right]$$

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

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

Task 1b)

SOFTMAX

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{k'}^{K} e^{z_{k'}}} \quad , \text{where } z_k = w_k^T \cdot x$$

$$= \sum w_{ki} \cdot x_i$$

$\frac{\partial \hat{y}_k}{\partial z_k}$ depends on $k = k'$

$$\Sigma = \sum_{k'}^{K} e^{z_{k'}}$$

① if $k = k'$

$$\frac{\partial \hat{y}_k}{\partial z_k} = \frac{e^{z_k}\Sigma - e^{z_k}e^{z_k}}{\Sigma^2} = \frac{e^{z_k}(\Sigma - e^{z_k})}{\Sigma^2}$$

$$= \hat{y}_k^n(1 - \hat{y}_k^n)$$

② if $k \neq k'$

$$\frac{\partial \hat{y}_k^n}{\partial z_k} = \frac{0 - e^{z_k}e^{z_{k'}}}{\Sigma^2} = -\hat{y}_k^n \hat{y}_{k'}^n$$

CROSS ENTROPY LOSS

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

$$\frac{\partial C^n}{\partial z_k} = -\sum y_k^n \frac{1}{\hat{y}_k^n} \frac{\partial \hat{y}_k^n}{\partial z_k}$$

[illegible] Split into two cases (①,②)

$$\frac{\partial C^n}{\partial z_k} = -\frac{y_k^n}{\cancel{\hat{y}_k^n}}\cancel{\hat{y}_k^n}(1-\hat{y}_k^n) - \sum_{k\neq k'}^{K}\frac{y_k^n}{\cancel{\hat{y}_k^n}}(-\cancel{\hat{y}_k^n}\hat{y}_{k'}^n)$$

$$= -y_k^n(1-\hat{y}_k^n) + \sum_{k\neq k'}^{K} y_k^n\hat{y}_{k'}^n$$

$$= y_k^n\hat{y}_k^n - y_k^n + \sum_{k\neq k'}^{K} y_k^n\hat{y}_{k'}^n$$

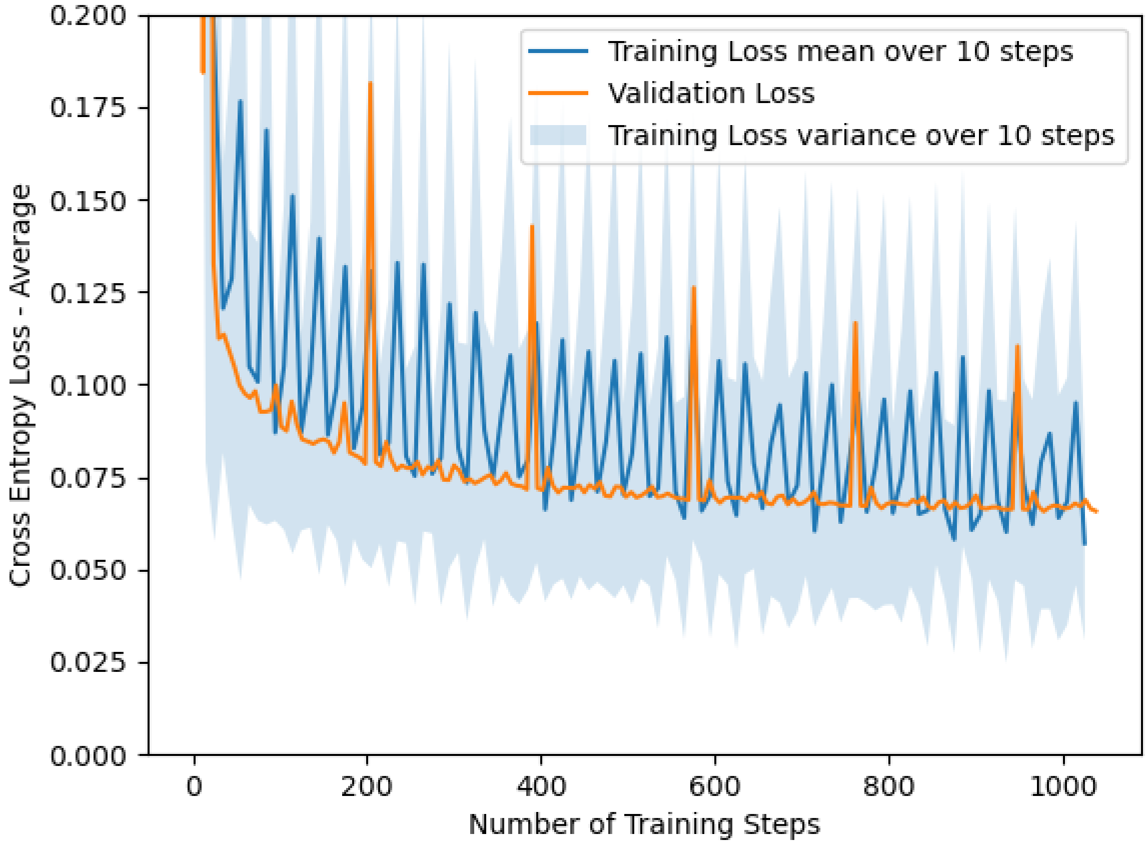
$$= -y_k^n + \underbrace{\sum_{k=1}^{K} y_k^n}_{=1}\hat{y}_{k'}^n = \hat{y}_{k'}^n - y_k^n$$

$$\frac{\partial z_k}{\partial w_{kj}} = x_j$$

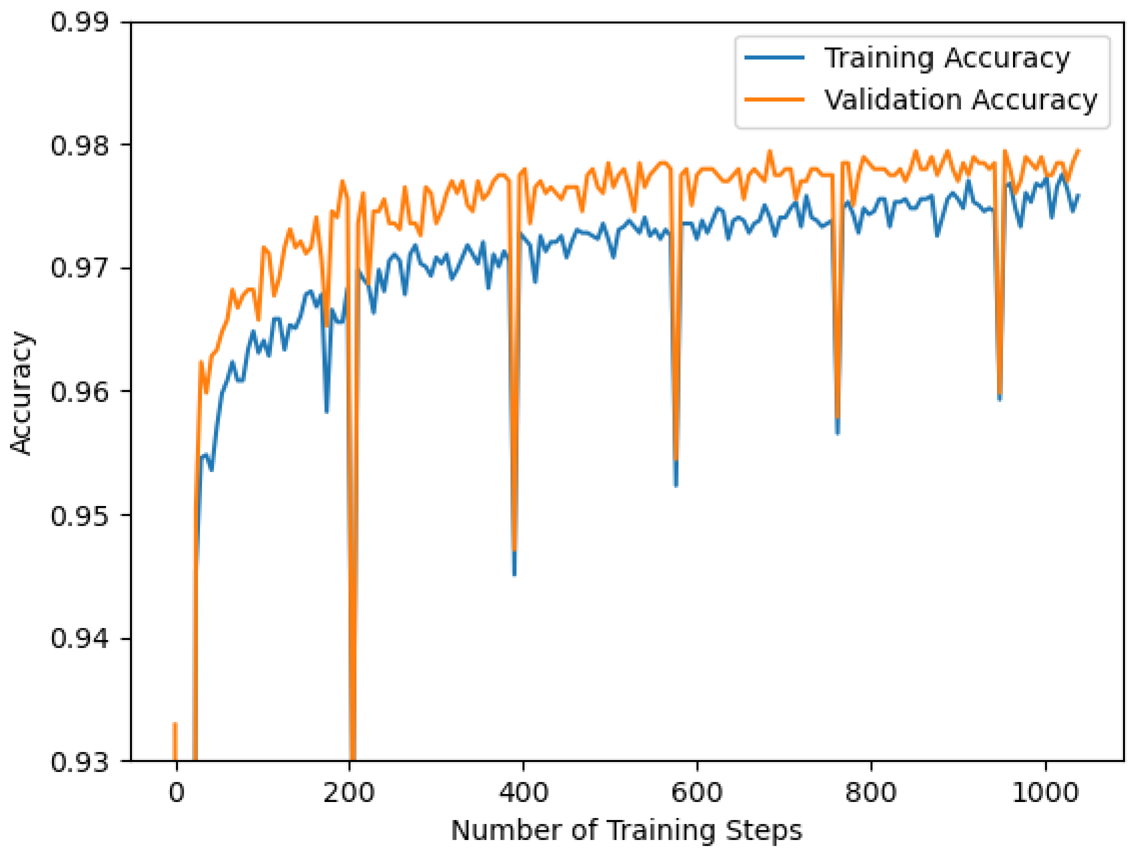
$$\frac{\partial C^n(w)}{\partial w_{kj}} = \frac{\partial z_k}{\partial w_{kj}} \cdot \frac{\partial C^n}{\partial z_k} = -x_j^n(y_k^n - \hat{y}_k^n)$$

Task 2

Task 2b)



Task 2c)



Task 2d)

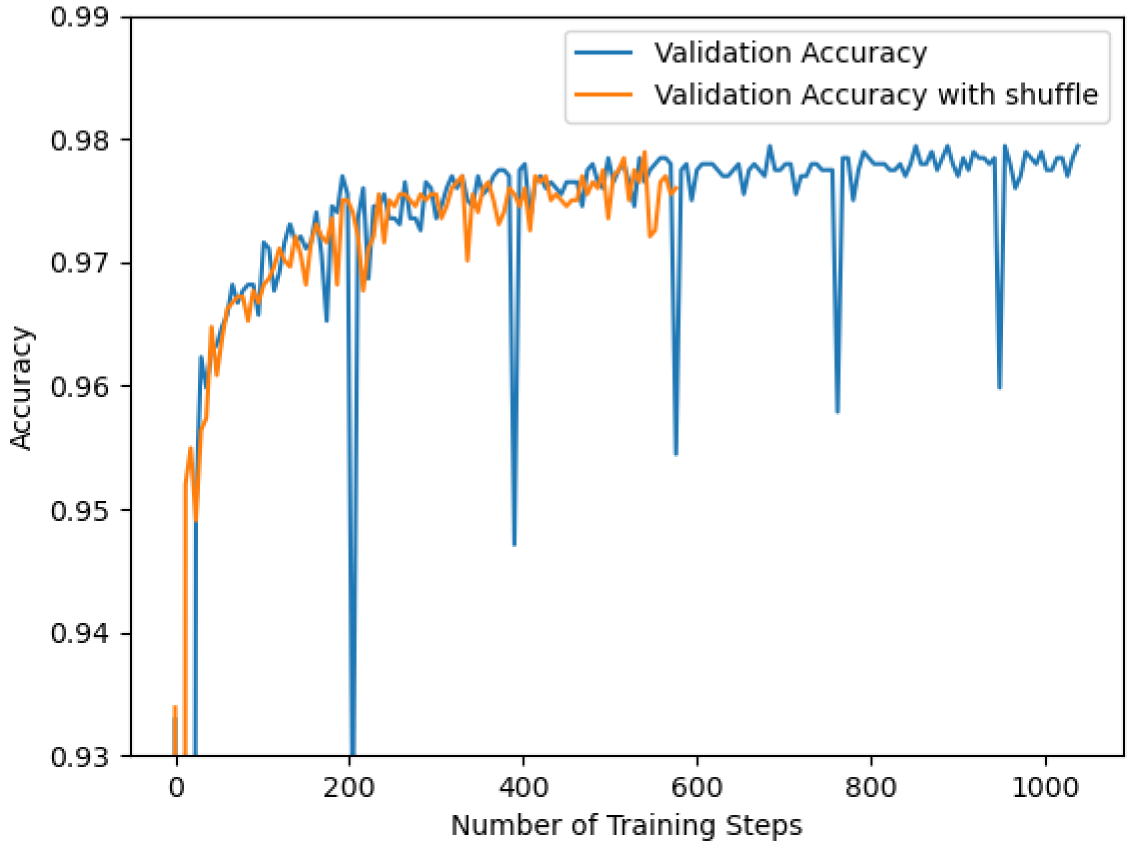
Early Stopping kicks in at Epoch 33 when using `shuffle = False`.

Early Stopping kicks in at Epoch 18 when using `shuffle = True`.

Task 2e)

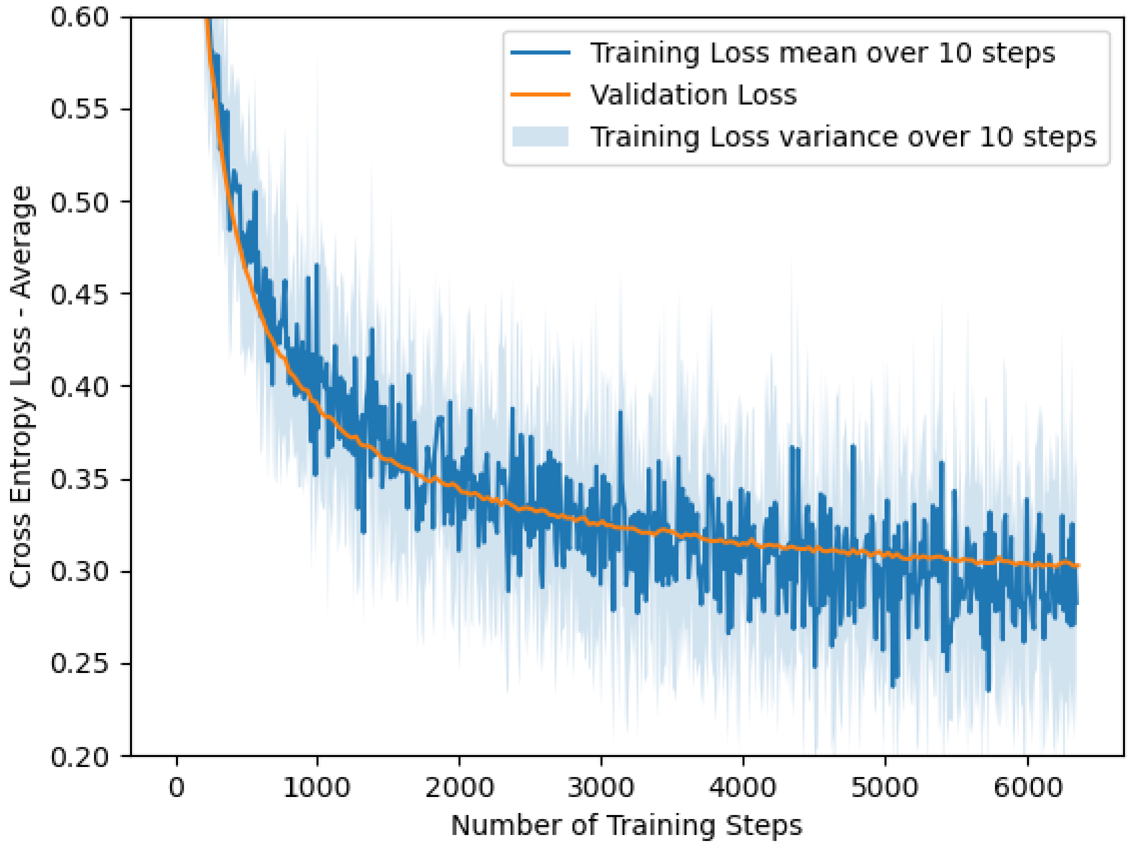
The accuracy without shuffle has spikes as it contains a batch that the model performes significantly worse on than the rest. This batch appears at around the same time for each epoch - giving us evenly spaced out spikes whenever it appears.

Shuffeling solves this problem as the model is presented new, random, batches each epoch.

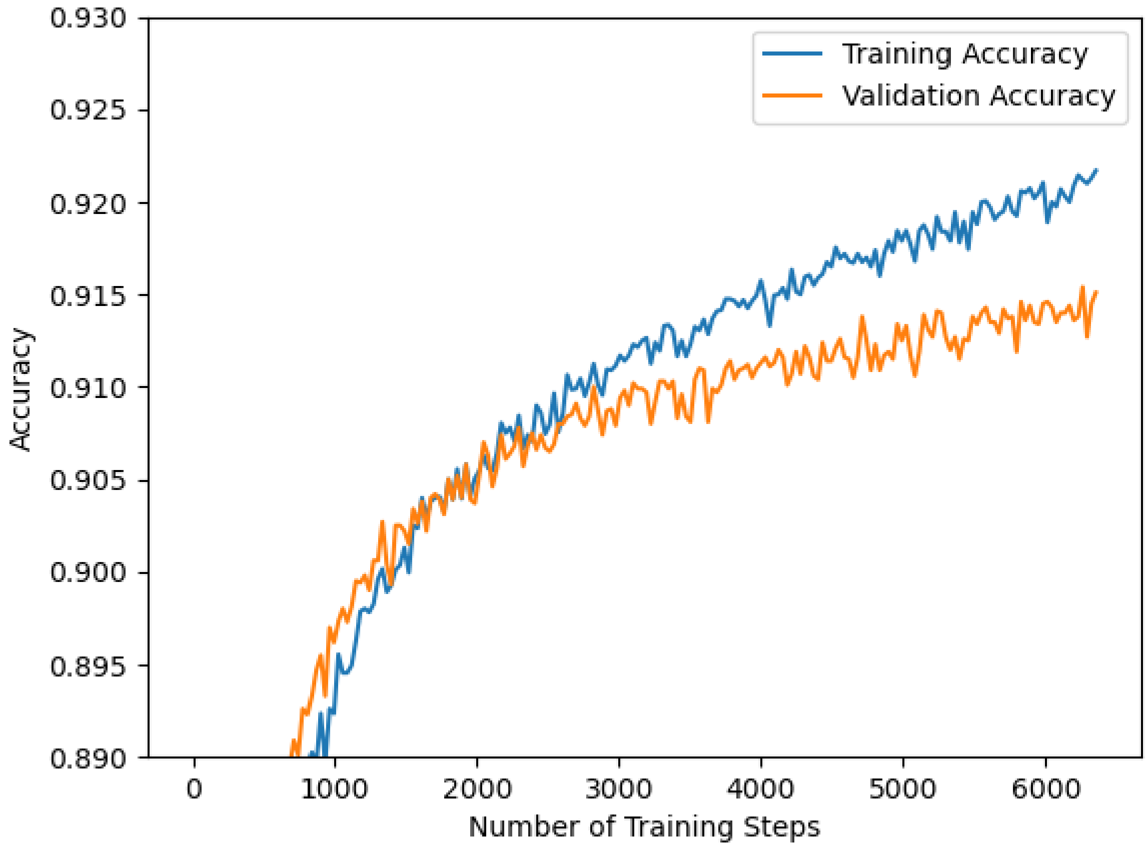


Task 3

Task 3b)


Training Loss mean over 10 steps
Validation Loss
Training Loss variance over 10 steps
Cross Entropy Loss - Average
Number of Training Steps
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0
1000
2000
3000
4000
5000
6000

Task 3c)


Training Accuracy
Validation Accuracy
Accuracy
Number of Training Steps
0.930
0.925
0.920
0.915
0.910
0.905
0.900
0.895
0.890
0
1000
2000
3000
4000
5000
6000

Task 3d)

As the plot shows, there is a growing gap between training accuarcy and validation accuracy. Such a trend could suggest overfitting. However, continuously increasing accuracy of the validation set argues otherwise. We believe that as long as the model is not trained on the validation set, an increase in validation accuracy is a positive event.

If the Validation Accuracy trend was decreasing or flat, steps to reduce overfitting (ie. Early Stopping) would have to be implemented. Early stopping could be implemented in a more strict way - for example requiring a minimum increase in val. accuracy over time.

Task 4

Task 4a)

We have

$$J(w) = C(w) + \lambda R(w)$$

where, $R = \|w\|^2 = \sum_{i,j} w_{i,j}^2$.

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

We have $\frac{\partial C}{\partial w}$ from assignment one

$$\frac{\partial C}{\partial w} = -(y_k^n - \hat{y}_k^n)x^n$$

derivation of $\frac{\partial R}{\partial w}$ gives

$$\frac{\partial R}{\partial w} = 2w_{i,j}$$

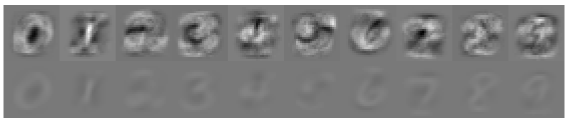
which gives

$$\frac{\partial J}{\partial w} = 2\lambda w_{i,j} - (y_k^n - \hat{y}_k^n)x^n$$

Task 4b)

With regularization, a penalization term is added to the objective that is to be minimized. With l2-norm penalization, the objective function is penalized for large weight sizing.

As seen in the picture below, when using `lambda = 0` , the objective function is NOT penalized and will set weights freely, "without worries". The model is not punished for sizing weights as it wishes, which results in a noisy weighting. When `lambda = 1` , much less weight is given to each input pixel, as the objective funciton is much more greedy in where it is willing to "allocate resources", resulting in significantly less noise.



Task 4c)

Regularization is implemented to reduce overfitting. It achieves this by penalizing the size of the weights. By adding a penalty term to the equation (which is to be minimized) the resulting model will avoid oversizing weights - and hopefully be able to give generalized and better predictions.

That said, the value of lambda (size of penalty term) has to be tested to establish the best value. A large lambda can penalize weights too much - causing underfitting. Choosing `lambda` is a trade-off between overfitting and accuracy.

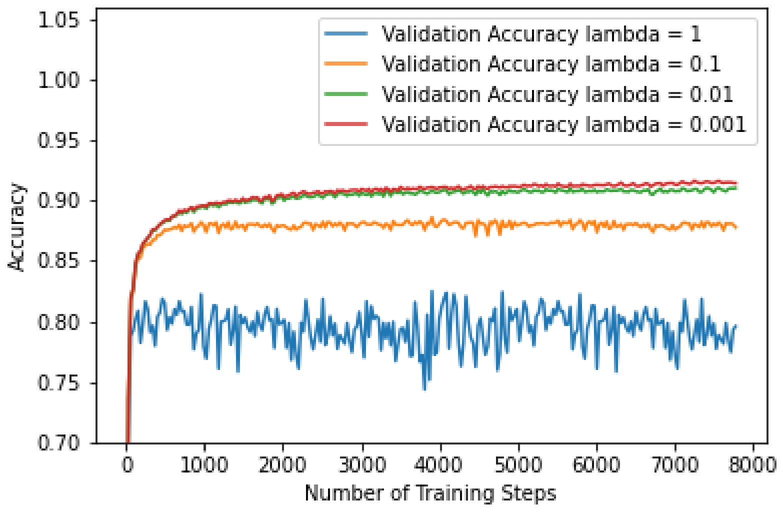
As seen in our plots:

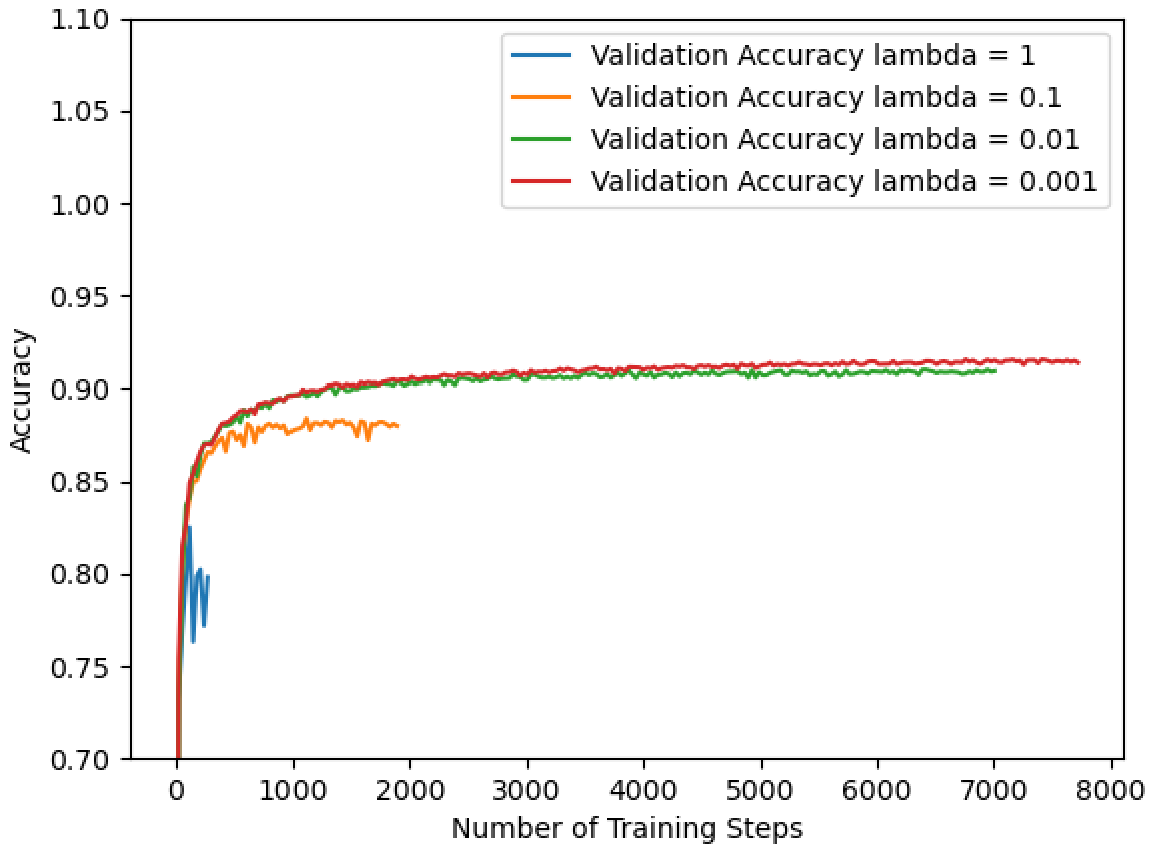
`lambda = 1` underfits the model.

`lambda = 0.1` underfits less than `lambda = 1` , but still suboptimal.

`lambda = 0.01` and `lambda = 0.001` acheive very similar results.

As the model trained with `lambda = 0.01` acheives similar results to `lambda = 0.001` . Our evaluation of the plots suggest choosing the model with `lambda = 0.01` as it is less likely to be overfit, even though `lambda = 0.001` has a performs slightly better.





ES kicks in, Step, Epoch: 279, 1

ES kicks in, Step, Epoch: 1891, 12

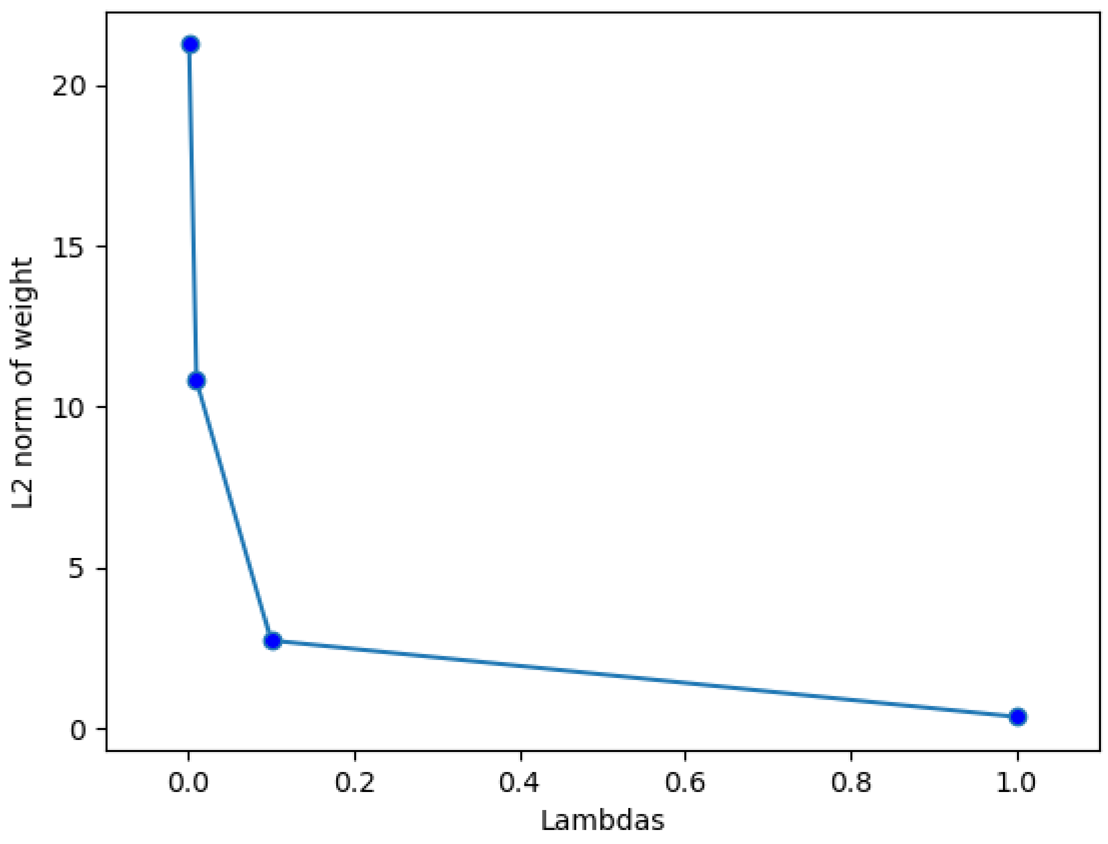
ES kicks in, Step, Epoch: 7006, 44

ES kicks in, Step, Epoch: 7719, 49

Task 4d)

By punishing weight-sizing, we have to expect lower accuracies. The gains of using regularization are not percieved through accuracy plots, but hopefully by achieving a more generalized model.

Task 4e)



When lambda increase the lengths of the L2 norm decreases. (See discussion in 4a, b, c, d)

```
In [ ]:
```