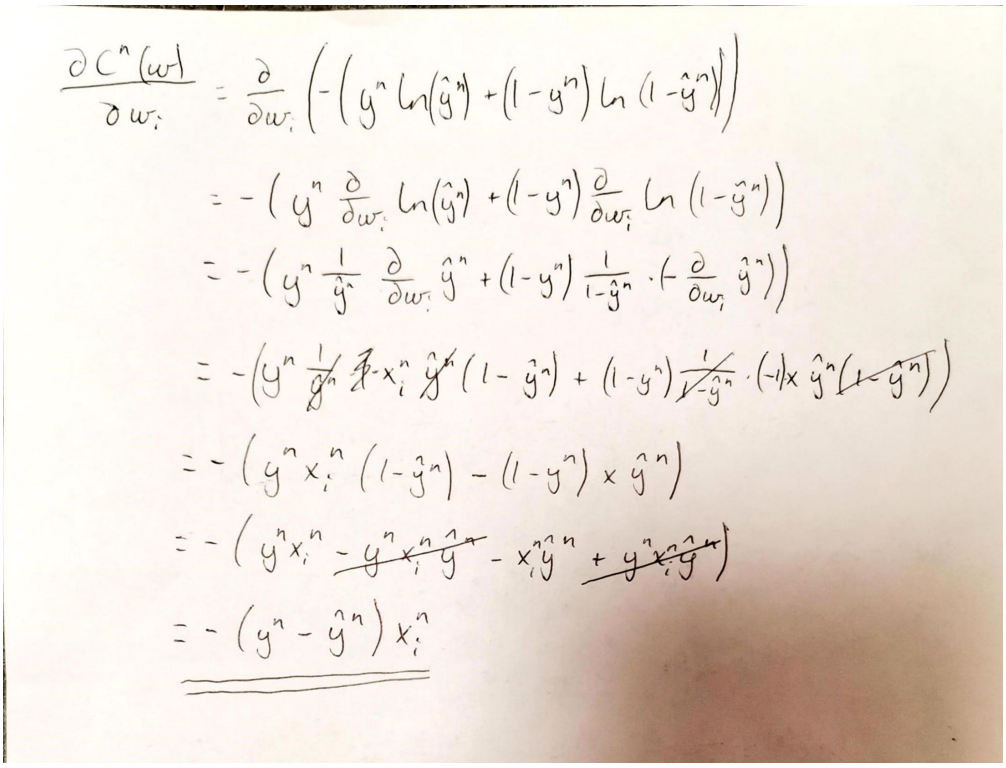
Assignment 1 - Sigmund S. Mestad

February 4, 2022

1 Task 1

1.1 task 1a)



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

$$= -\left(y^n \frac{\partial}{\partial w_i}\ln(\hat{y}^n) + (1-y^n)\frac{\partial}{\partial w_i}\ln(1-\hat{y}^n)\right)$$

$$= -\left(y^n \frac{1}{\hat{y}^n}\frac{\partial}{\partial w_i}\hat{y}^n + (1-y^n)\frac{1}{1-\hat{y}^n}\cdot\left(-\frac{\partial}{\partial w_i}\hat{y}^n\right)\right)$$

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

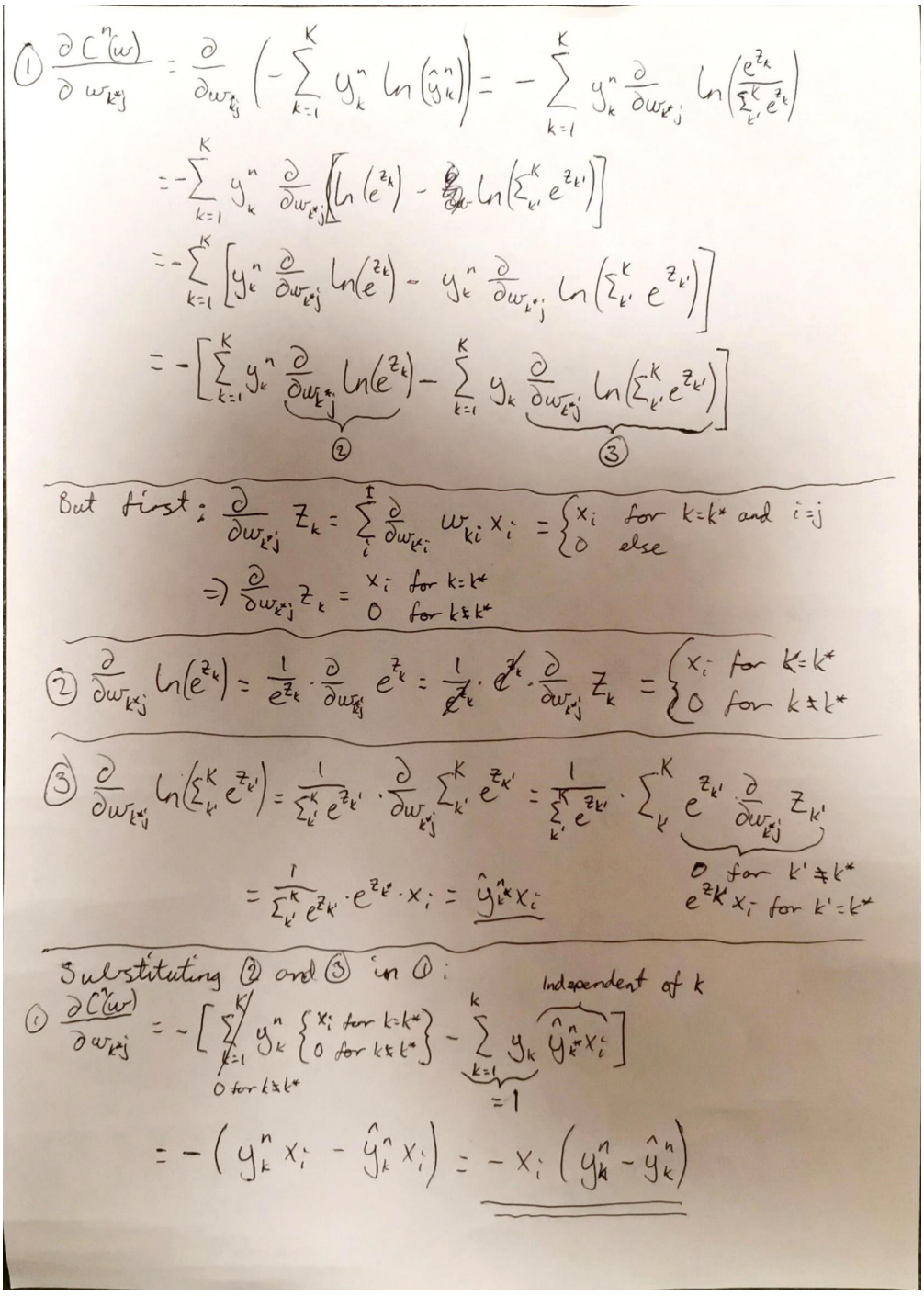
$$= -\left(y^n x_i^n(1-\hat{y}^n) - (1-y^n)\times\hat{y}^n\right)$$

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

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

Task 1a - The gradient for Logistic Regression

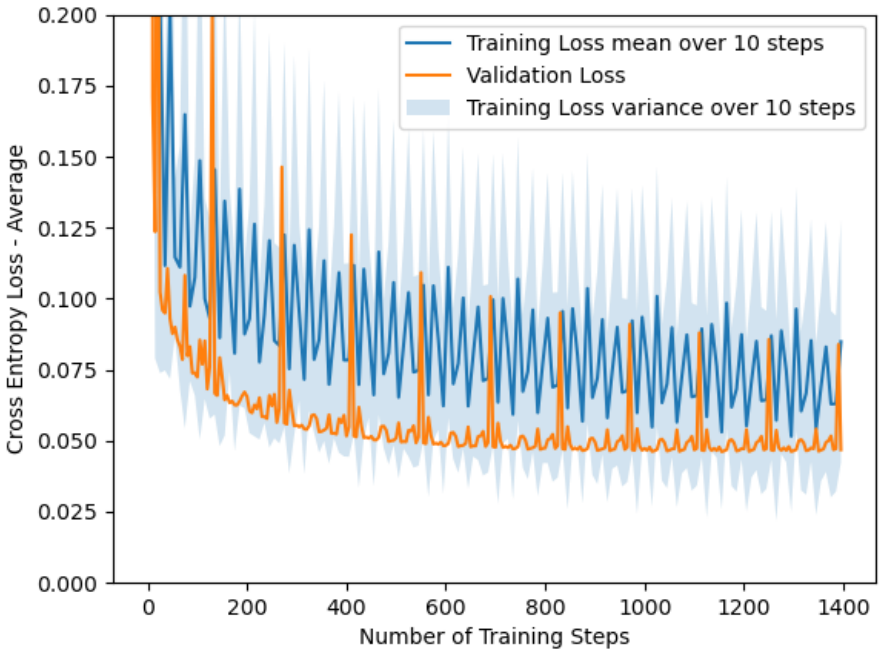
1.2 task 1b)



Task 1b - The gradient for Softmax Regression

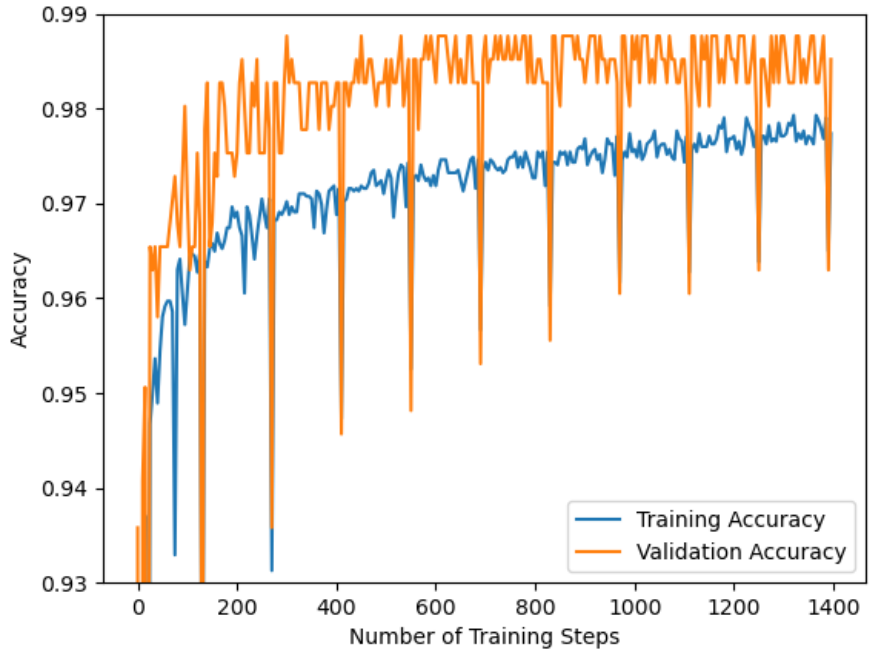
2 Task 2

2.1 Task 2b)



Task 2b - Loss

2.2 Task 2c)

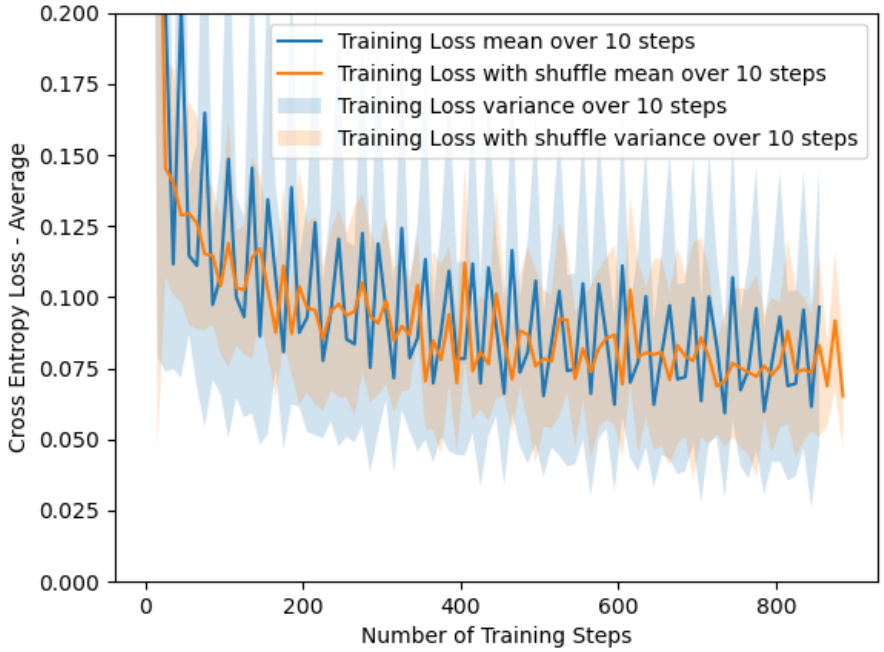


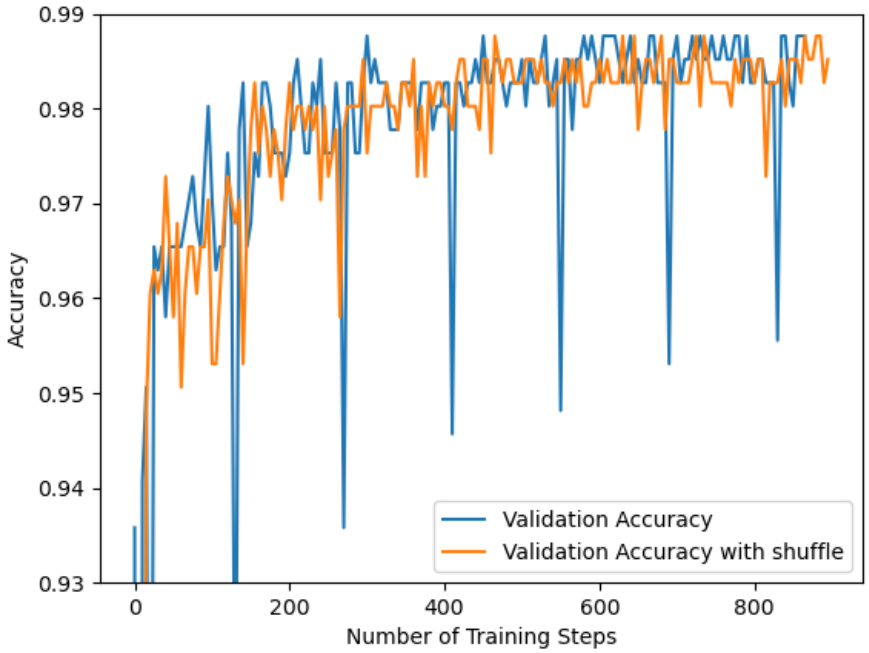
Task 2c - Accuracy

2.3 Task 2d)

Early stop kicks in after 32 epochs

2.4 Task 2e)

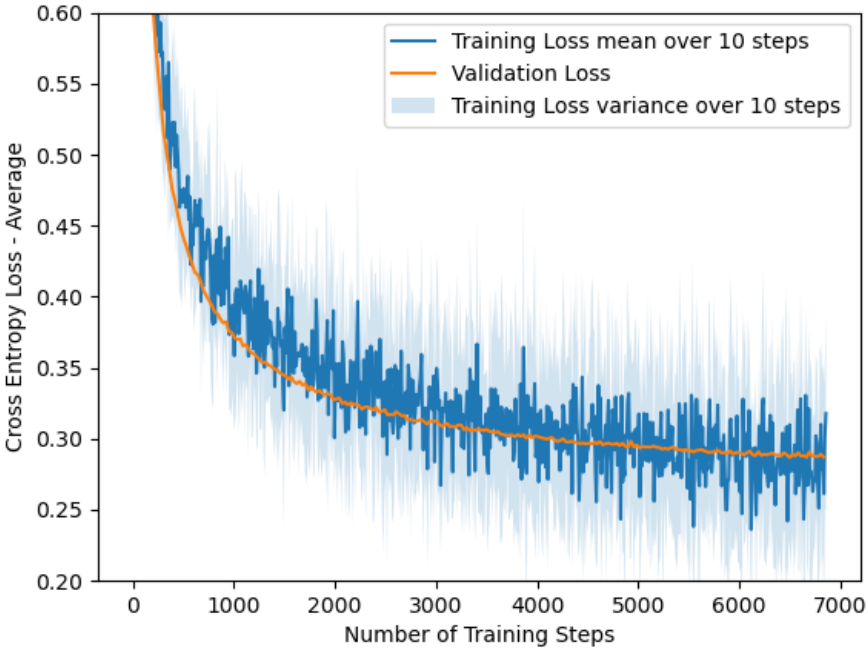




The accuracy have less spikes when shuffeling because the network is robust and generilized when presented with a new batch of images every time. This makes it less vulnarable for some "bad" batch confusing the network.

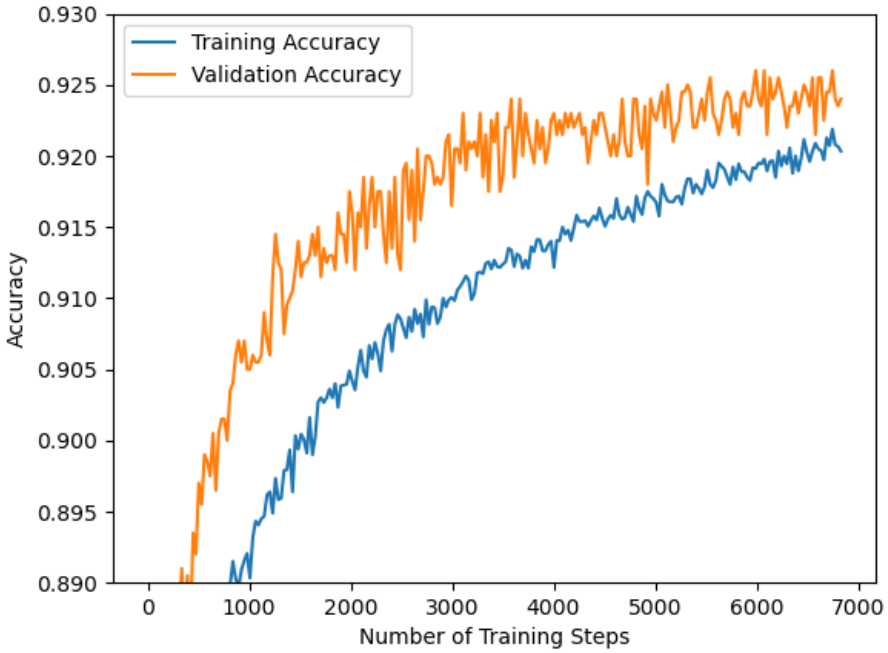
3 Task 3

3.1 Task 3b)



Task 3b - Training and validation loss

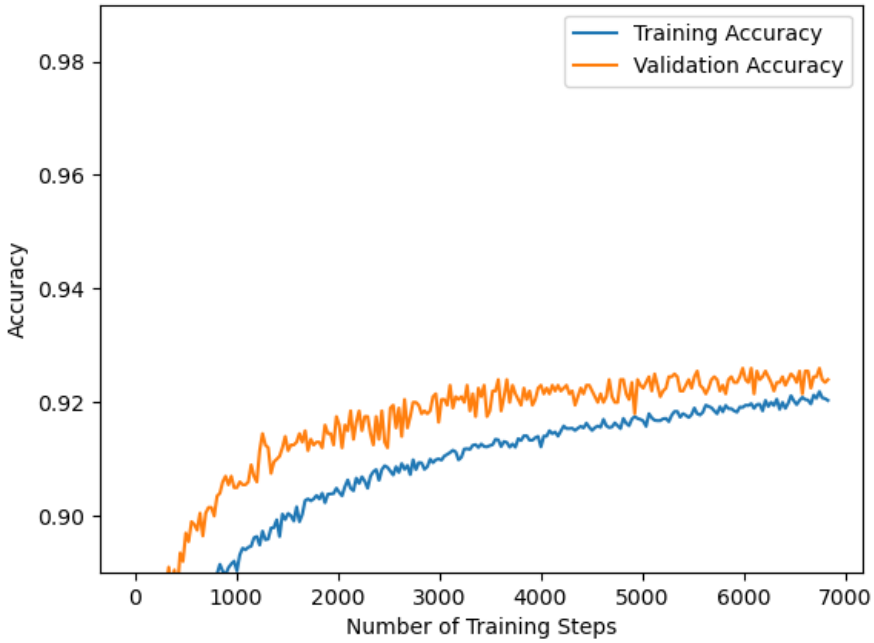
3.2 Task 3c)



Task 3c - Training and validation accuracy

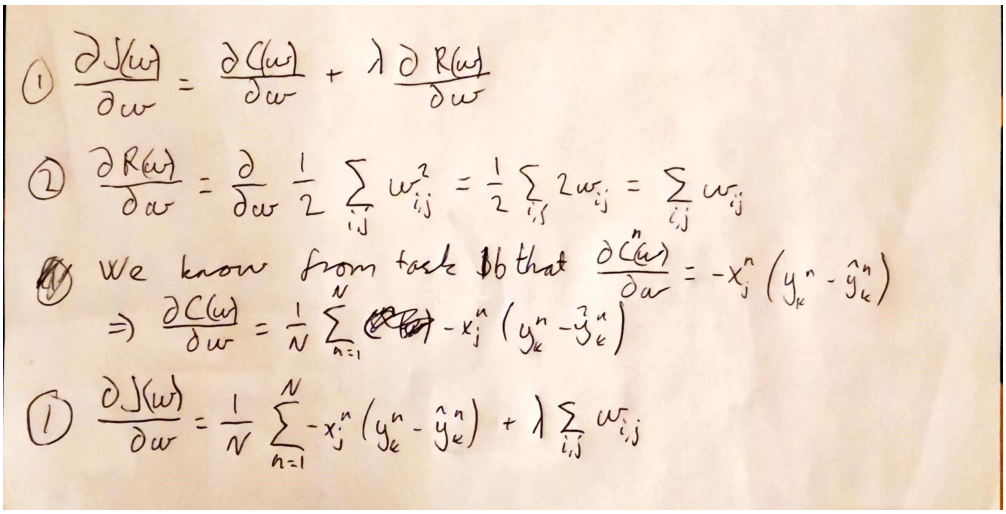
3.3 Task 3d)

The model doesn't show serious signs of overfitting. The validation accuracy seems to reach a platou at around 40-50 epochs (the early stop kicks in at 49). Doubling the number of epochs, we see that the training accuracy continues to grow, while the validation accuracy is completely flat (see plot below). A possible explaination of why the network doesn't seem to overfit could be because the model is too simple to learn all the details and noise of the training data which normally would lead to worse performance.



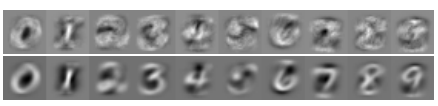
4 Task 4

4.1 Task 4a)



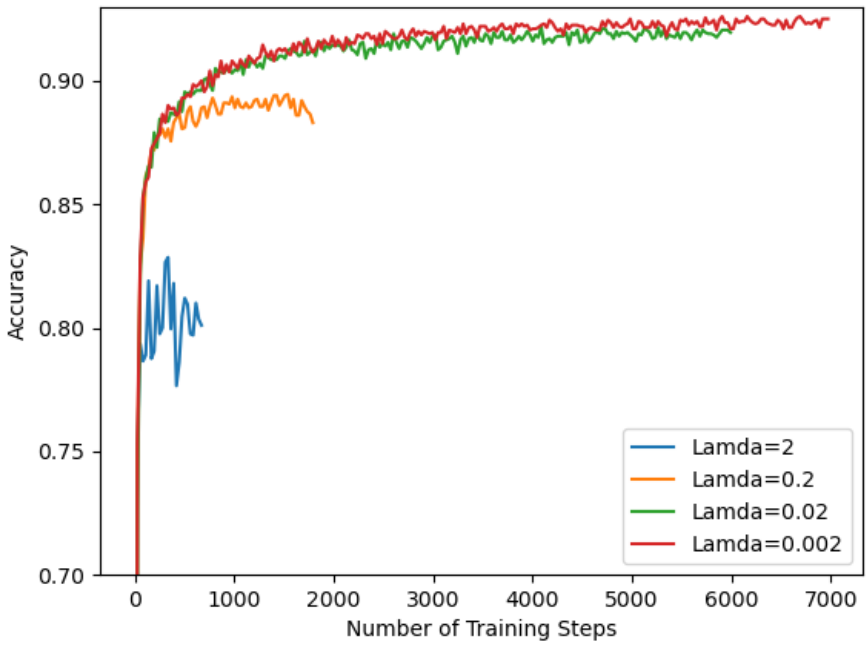
Task 4a - Update term for softmax regression with L2 regularization

4.2 Task 4b)



The regularization term reduces all weights, reducing the effect of less common patterns. Weights with low consensus will diminish.

4.3 Task 4c)

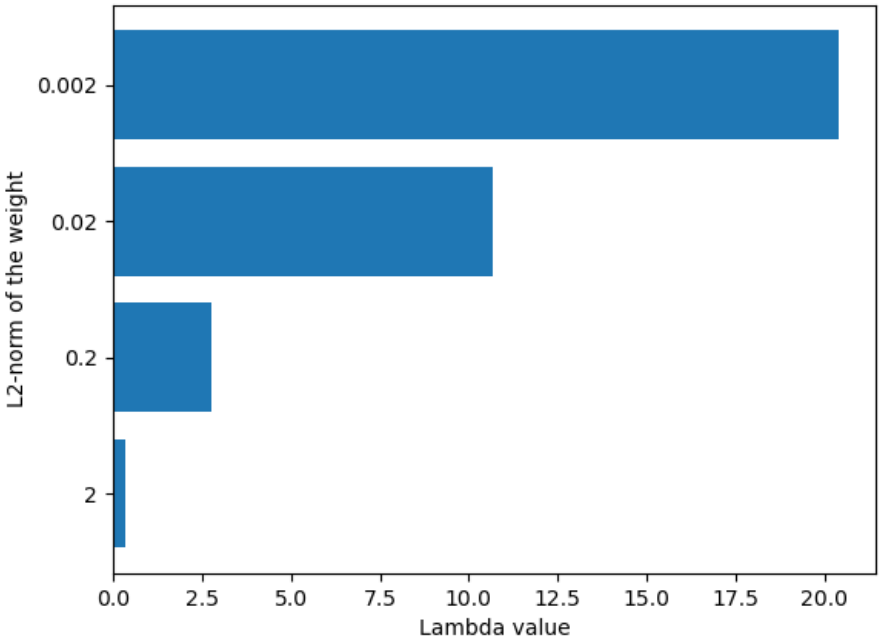


Task 4c - Accuracy with different lambda values

4.4 Task 4d)

The validation accuracy degrades when applying regularization because the model architecture is so simple. Making the model even more general, reduces its capacity to differnsiate the numbers.

4.5 Task 4e)



Task 4e - L2-normalization of weights

It's clear that more regularization leads to smaller weights.