TDT4265 - Computer Vision & Deep Learning

Assignment 1 Report - Group 113

> Dionysios Rigatos
> dionysir@stud.ntnu.no
> Joel Constantinos
> joelc@stud.ntnu.no

Task 1

Task 1a)

For this proof, we want to show that:

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

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

Using the chain rule, we can write the derivative of the cost function as:

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

We already know that:

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

What is left is to find $\frac{\partial C^n(w)}{\partial \hat{y}^n}$:

$$\frac{\partial C^n(w)}{\partial \hat{y}^n} =$$

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

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

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

So we now have:

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

Finally, we can write the derivative of the cost function as:

$\frac{\partial C^n(w)}{\partial w_i} =$

$= \frac{\partial C^n(w)}{\partial \hat{y}^n} \frac{\partial \hat{y}^n}{\partial w_i} =$

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

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

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

$= -x_i^n [y^n(1-\hat{y}^n) - \hat{y}^n(1-y^n)] =$

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

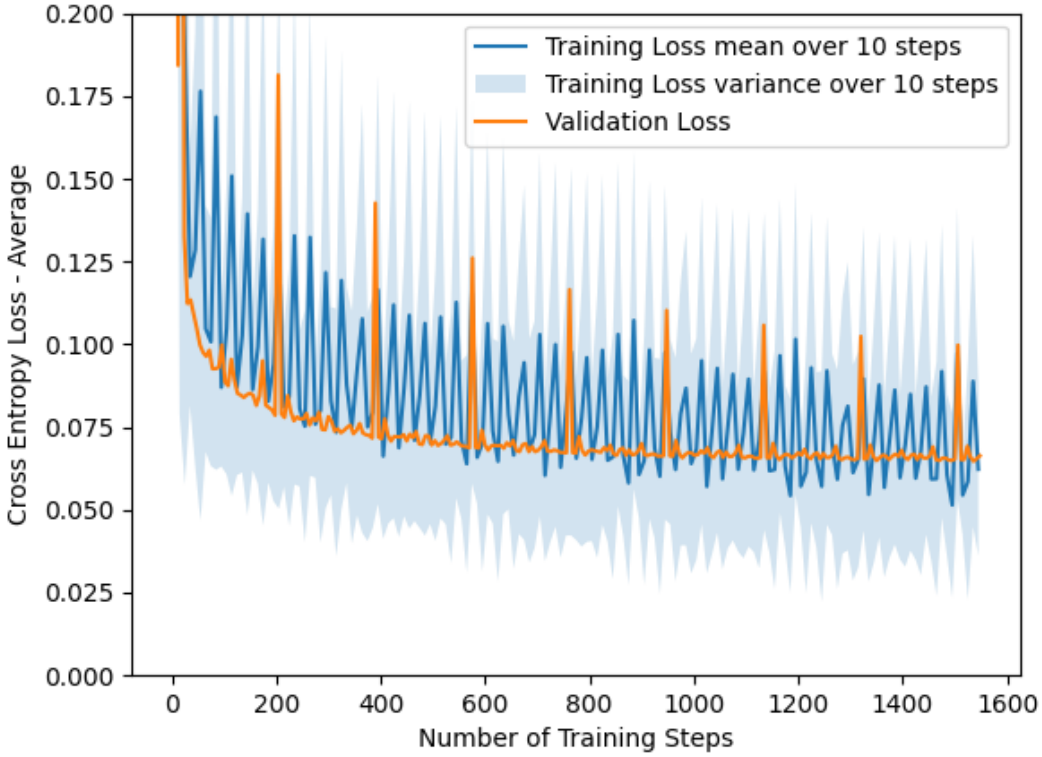
$= -x_i^n (y^n - \hat{y}^n) =$

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

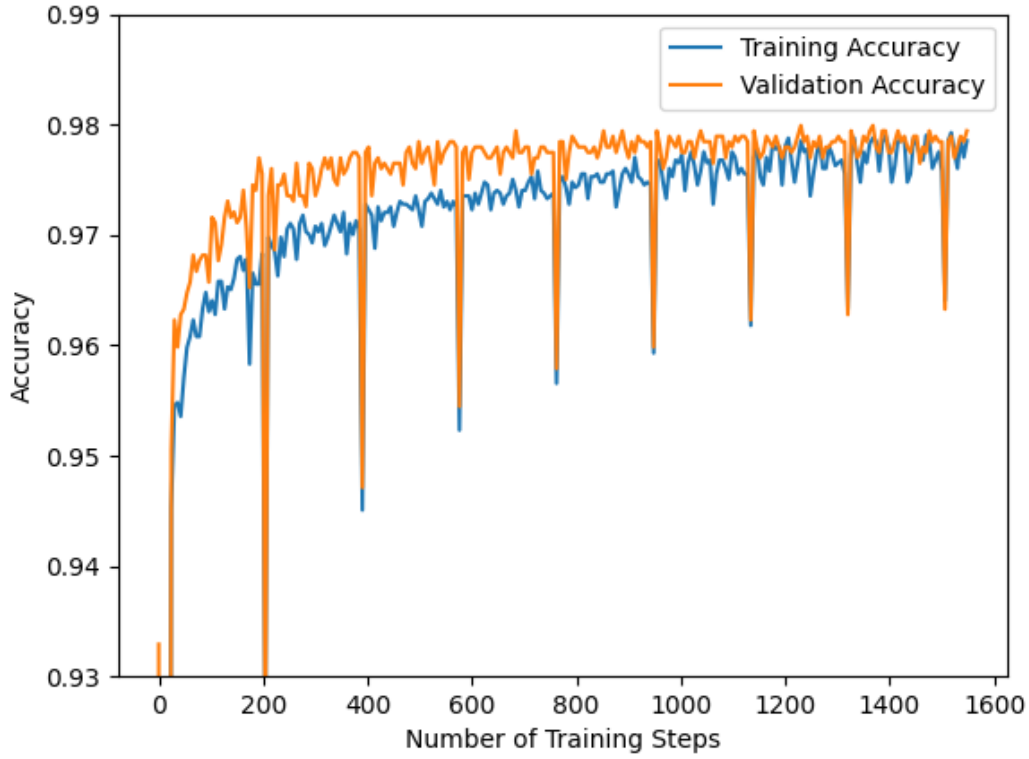
□

Task 2

Task 2b)


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

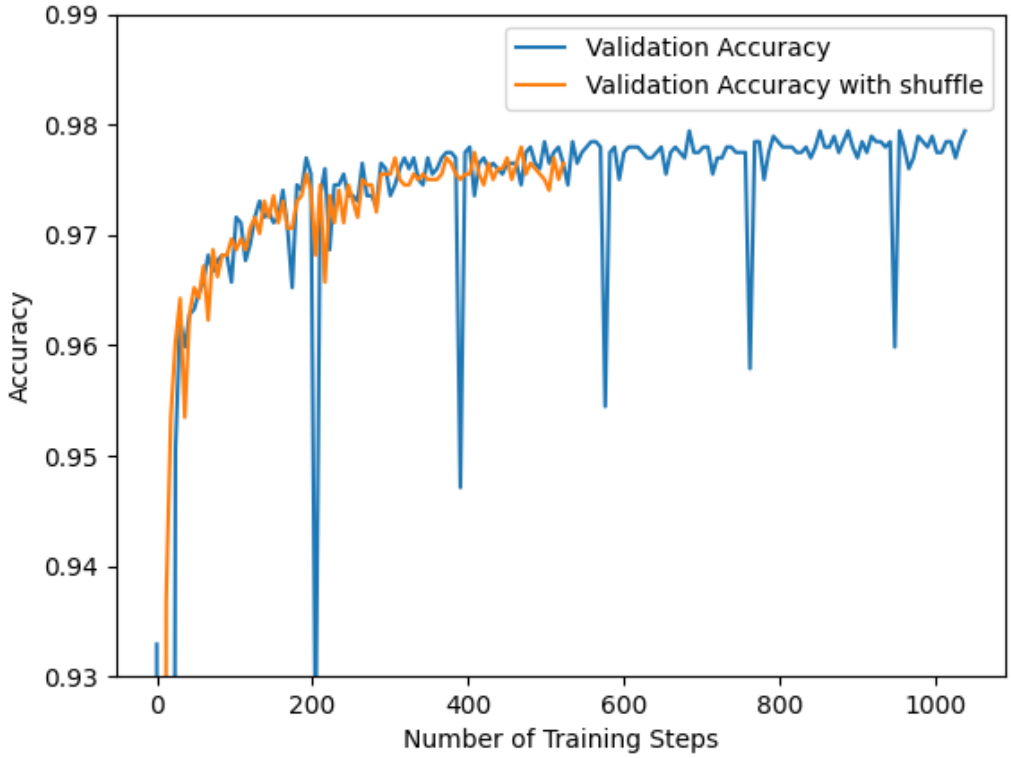
Task 2c)


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

Task 2d)

The early stopping kicked in at epoch number 33.

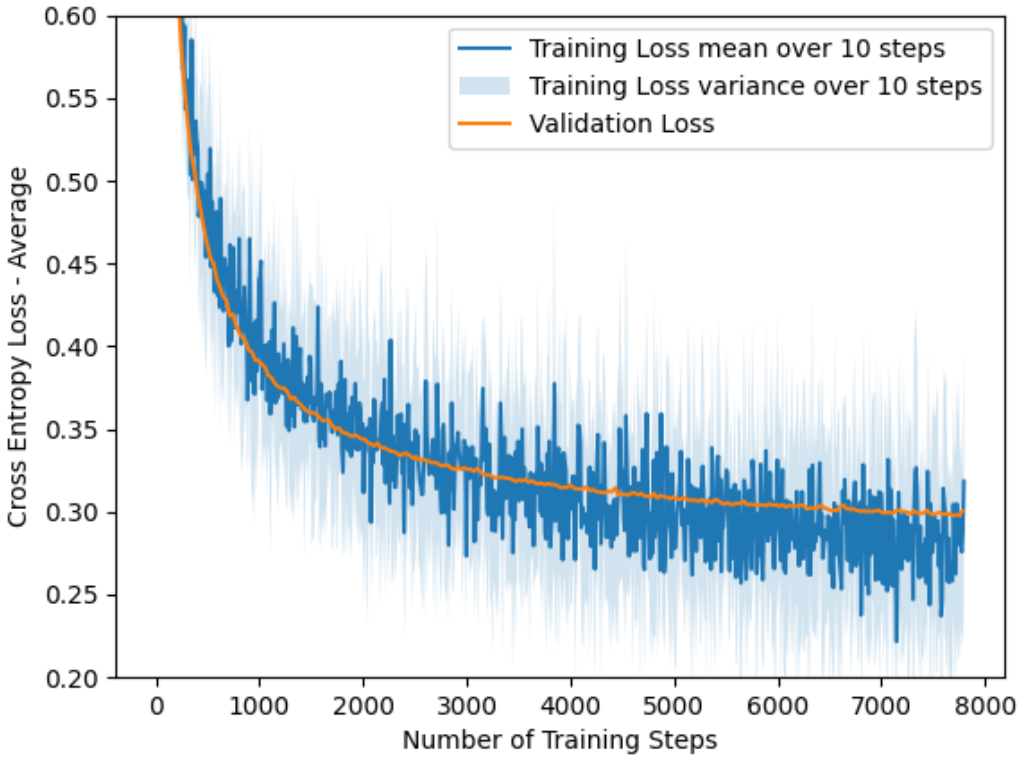
Task 2e)



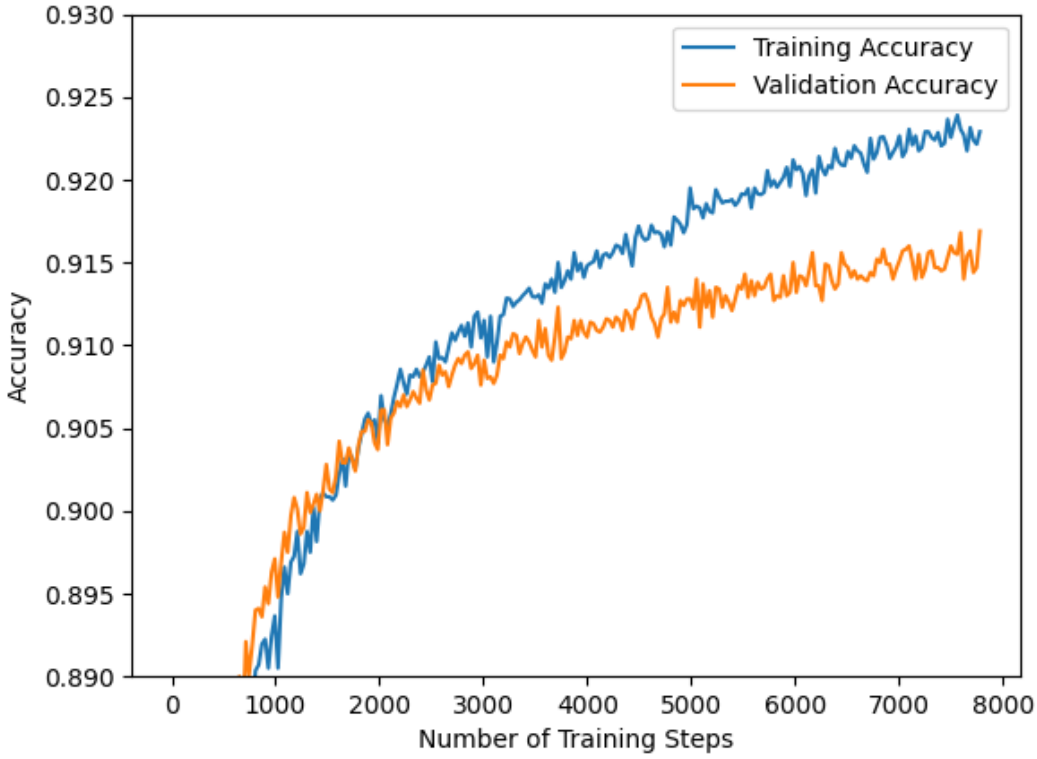
It seems that dataset shuffling has significantly improved convergence speed of the validation set opposed to the unshuffled runs (with early stopping on in both cases). The spikes we observed during unshuffled validation are indication of a problematic batch that consistently dunks the validation accuracy - a pattern that is not present in the shuffled runs as the data is not presented in the same order every time. Consequently, the model is trained on a more diverse set of data and the gradient updates are more consistent and eliminate the spikes early on.

Task 3

Task 3b)

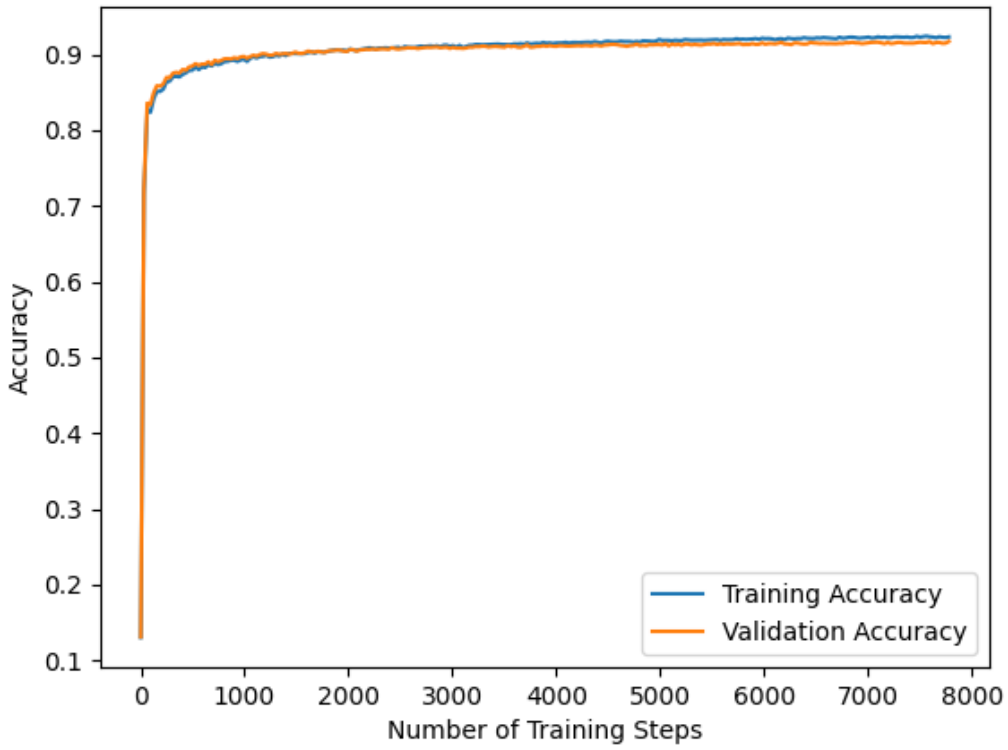

Training Loss mean over 10 steps
Training Loss variance over 10 steps
Validation Loss
Cross Entropy Loss - Average
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0
1000
2000
3000
4000
5000
6000
7000
8000
Number of Training Steps

Task 3c)


Training Accuracy
Validation Accuracy
Accuracy
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
7000
8000
Number of Training Steps

Task 3d)

Looking at the plot, we can see that there are signs of divergence between the training and validation accuracy - with the training accuracy increasing and the validation accuracy reaching a plateau. While **this may be an early sign of overfitting**, looking at the scale of the initial plot, the difference is not significant. This becomes more evident when we look at the zoomed out accuracy plot, where the trend seems reasonable - a slightly higher training accuracy than validation accuracy with a very small gap.



Task 4

Task 4a)

For this proof, we want to derive the update term for L2 regularization:

- $\frac{\partial J(w)}{\partial w}$

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

So, we have:

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

We already know from Task (1a) that:

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

So the derivative of $C(w_{kj})$ with respect to w_{kj}, for the entire dataset, is:

- $\frac{\partial C(w)}{\partial w_{kj}} = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial C^n(w)}{\partial w_{kj}}$

So now we need to find the derivative of the L2 regularization term:

$\frac{\partial R(w)}{\partial w_{kj}} =$

$\frac{\partial \sum_{i=1,j=1}^{N} w_{ij}^2}{\partial w_{kj}} =$

$= 2w_{kj}$

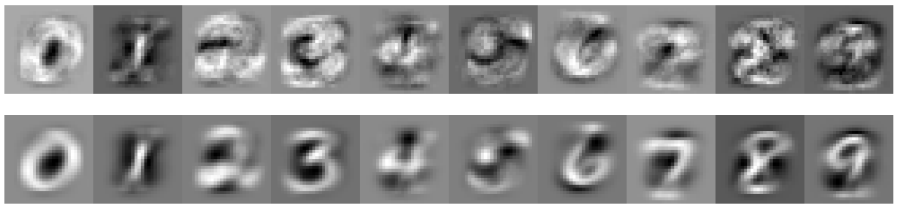
So, finally, we have:

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

$= \frac{1}{N}\sum_{n=1}^{N} -(y_k^n - \hat{y_k}^n)x_j^n + 2\lambda w_{kj}$

□

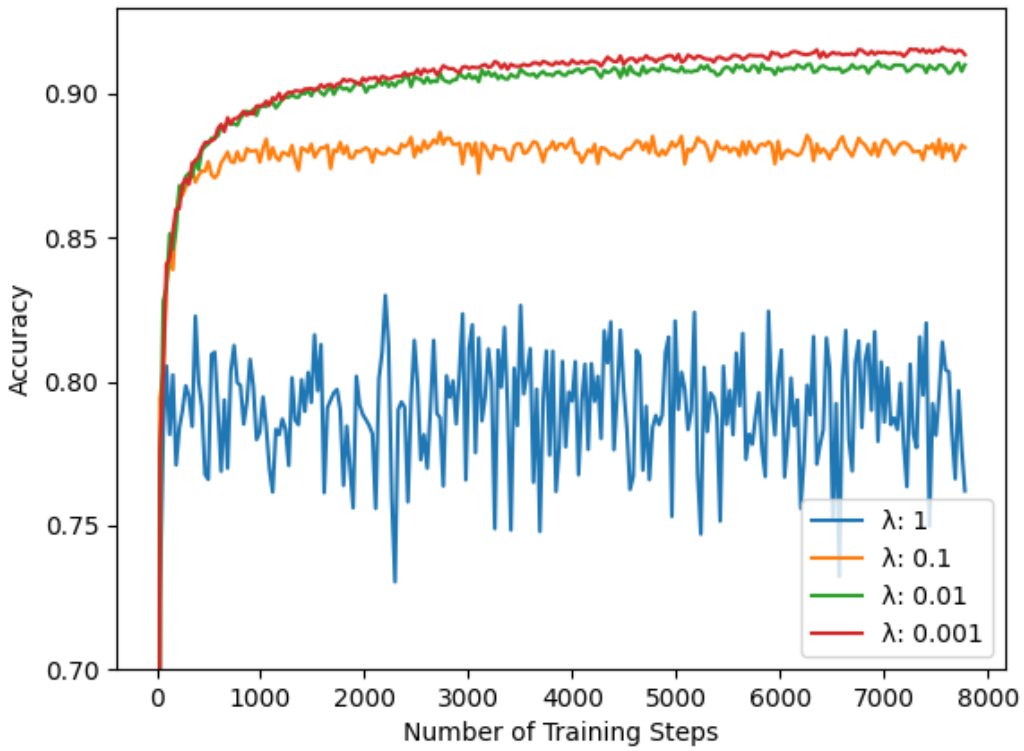
Task 4b)



It is evident that the model with the regularized weights has significantly less noise.

Due to the L2 regularization penalty, the weights are kept small. This leads to a reduction in the variance of the model, which in turn reduces the noise in the training and validation loss.

Task 4c)



NOTE: Early stopping was not used in this task.

Task 4d)

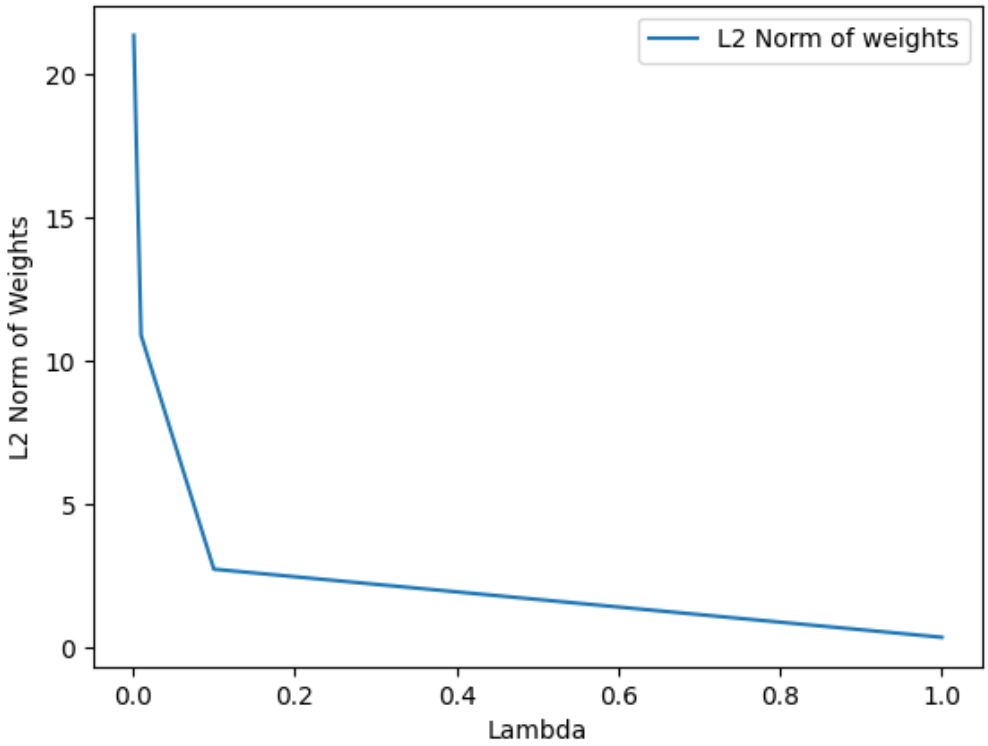
Regularization is supposed to improve the generalization of the model, and the validation accuracy is a good indicator of this.

However, this is not the case here. This might be because the dataset is too small, and the model is not complex enough so as to benefit from regularization.

Another reason for decreasing accuracy as lambda increases is that the regularization strength increases. With a higher lambda, the penalty induced on the weight matrix is much stronger and the model ends up being more focused on minimizing the weights rather than the loss function.

This results in a model that is less capable of generalizing to the validation set, and the validation accuracy decreases.

Task 4e)



The more we increase the lambda regularization value, the more the model is penalized for having large weights.

This is evident in the L2 norm of the weights, which decreases significantly as the lambda value increases.