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 (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet). 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`.

Task 1

task 1a)

Assignment 1

$\hat{y} = a = f(z)$, $z = wx + b$

1a) Regression: $C(w) = \frac{1}{N}\sum_{n=1}^{N} C^n$ $\quad C^n(w) = -(y^n \ln(\hat{y}^n) + (1-y^n)\ln(1-\hat{y}^n))$

$$\frac{\partial C^n}{\partial w_i} = \frac{\partial C^n}{\partial f(x^n)} \cdot \frac{\partial f(x^n)}{\partial w_i} \qquad \hat{y}^n = f(x^n)$$

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

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

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

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

Softmax:

task 1b)

Softmax:

1b) $\hat{Y}_k = \frac{e^{z_k}}{\sum_{k'}^{K} e^{z_{k'}}}$, $z_k = w_k^T x = \sum_i^I w_{k,i} x_i \Rightarrow \frac{\partial z_k^n}{w_{kj}} = x_j^n$

$C(w) = \frac{1}{N} \sum_{n=1}^{N} C^n(w)$, $C^n(w) = -\sum_{k=1}^{K} Y_k^n \ln(\hat{Y}_k^n)$

$$\frac{\partial C^n}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}} \left(-\sum_{k''=1}^{K} Y_{k''}^n \ln(\hat{Y}_{k''}^n) \right)$$

$$= -\frac{\partial}{\partial w_{kj}} \sum_{k''=1}^{K} Y_{k''}^n \ln\left(\frac{e^{z_{k''}^n}}{\sum_{k'}^{K} e^{z_{k'}^n}} \right)$$

$$\Rightarrow -\frac{\partial C^n}{\partial w_{kj}} = + \frac{\partial}{\partial w_{kj}} \sum_{k''=1}^{K} Y_{k''}^n \left(z_{k''}^n - \ln\left(\sum_{k'}^{K} e^{z_{k'}^n} \right) \right)$$

For $k''=k$ we have:

$$\frac{\partial}{\partial w_{kj}} y_k^n \left(z_k^n - \ln \left(\sum_{k'}^{K} e^{z_{k'}^n} \right) \right)$$

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

For $k'' \neq k$ we have:

$$\frac{\partial}{\partial w_{kj}} \sum_{k'' \neq k}^{K} y_{k''}^n \left(z_{k''}^n - \ln \left(\sum_{k'}^{K} e^{z_{k'}^n} \right) \right)$$

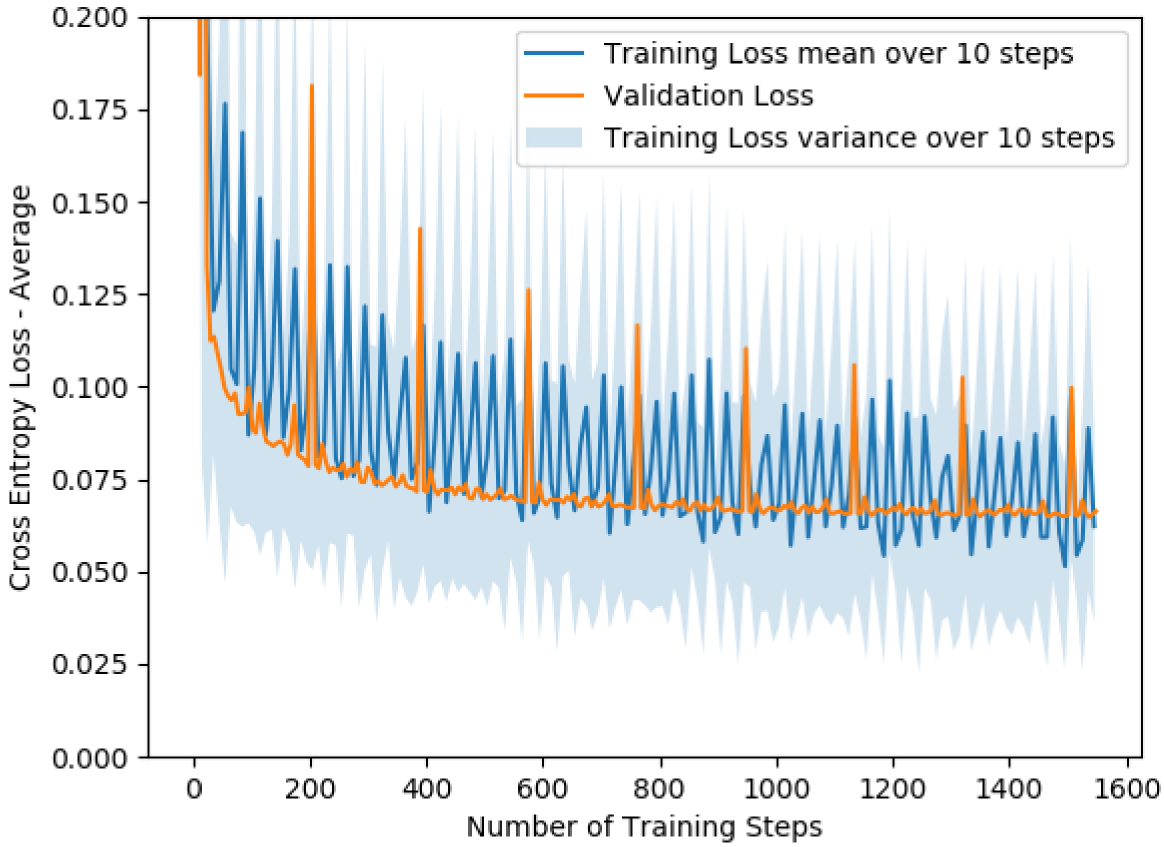
$$= \sum_{k'' \neq k}^{K} 0 - \frac{y_{k''}^n}{\sum_{k'}^{K} e^{z_{k'}^n}} e^{z_k^n} x_j^n = -x_j^n \sum_{k'' \neq k}^{K} y_{k''}^n \hat{y}_{k''}^n$$

$$\Rightarrow + \frac{\partial C^n}{\partial w_{kj}} = -x_j^n \left(y_k^n - \sum_{k''}^{K} y_{k''}^n \hat{y}_{k''}^n \right)$$

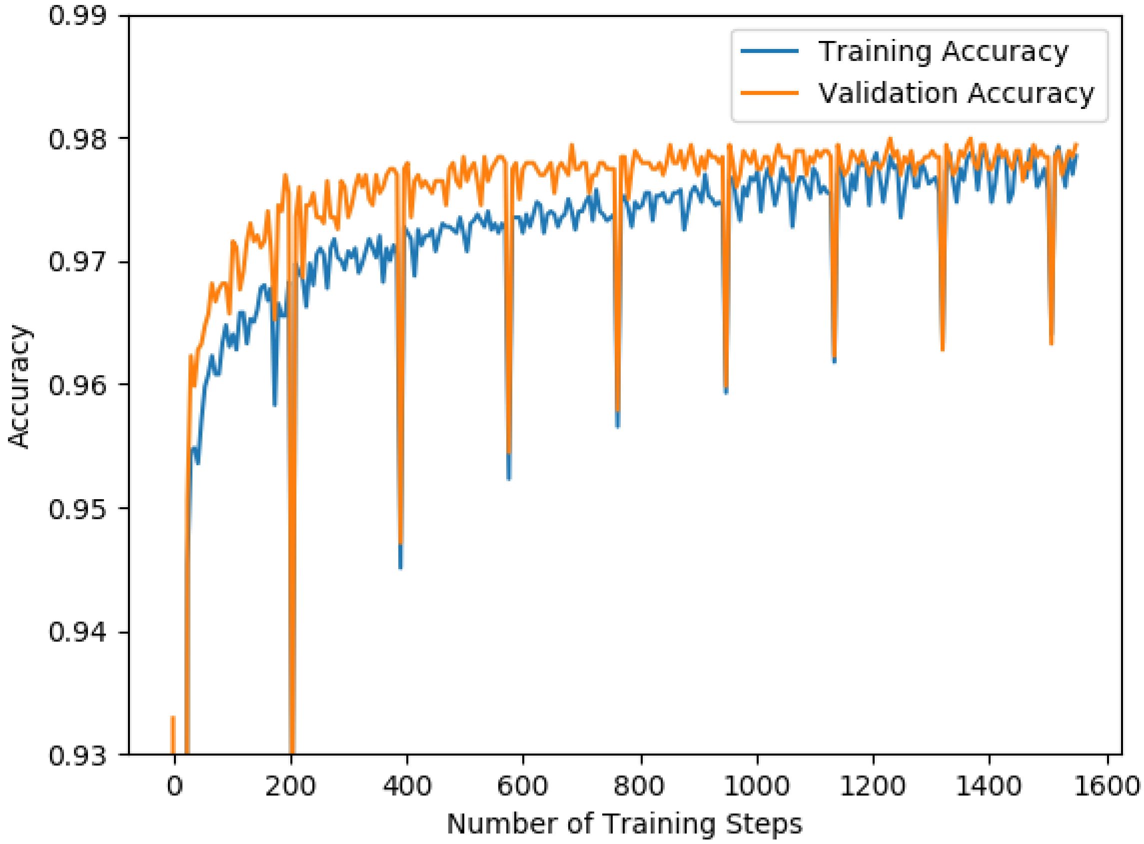
$$= -x_j^n \left(y_k^n - \hat{y}_k^n \right)$$

Task 2

Task 2b)



Task 2c)

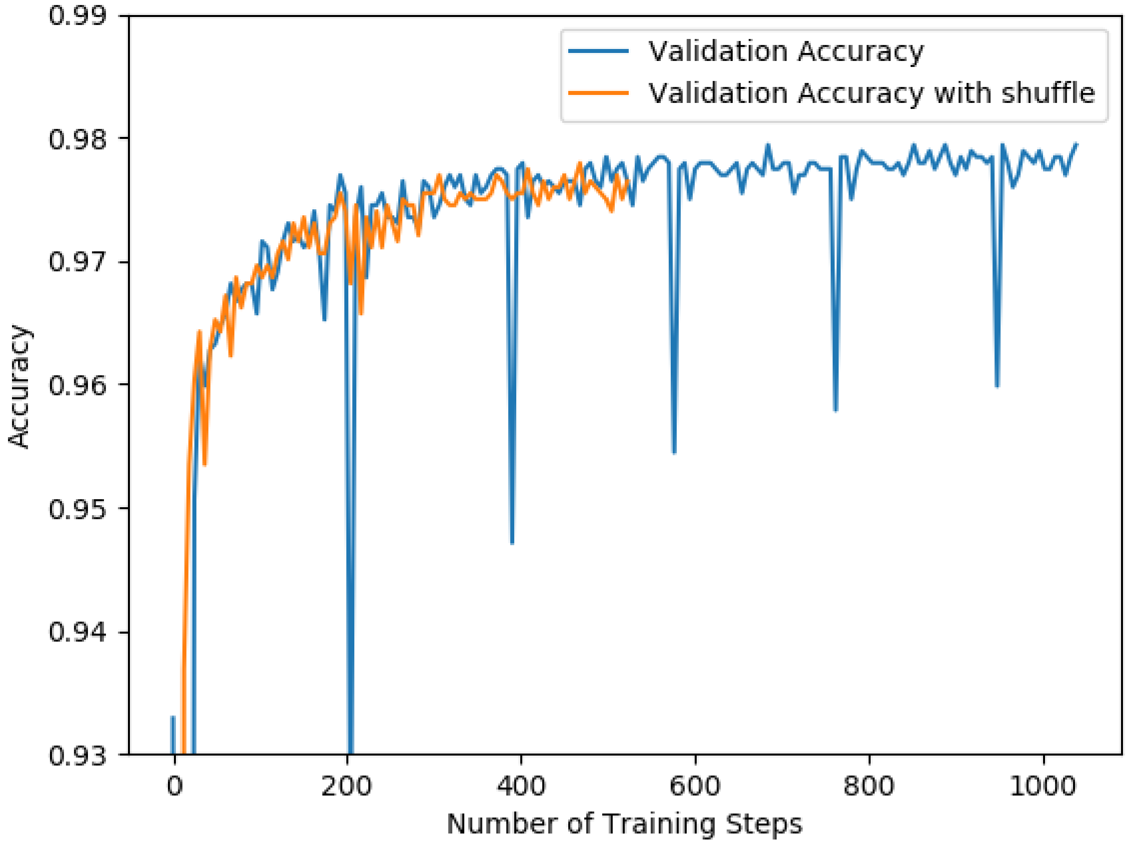


Task 2d)

The early stop kicked in after 33 epochs

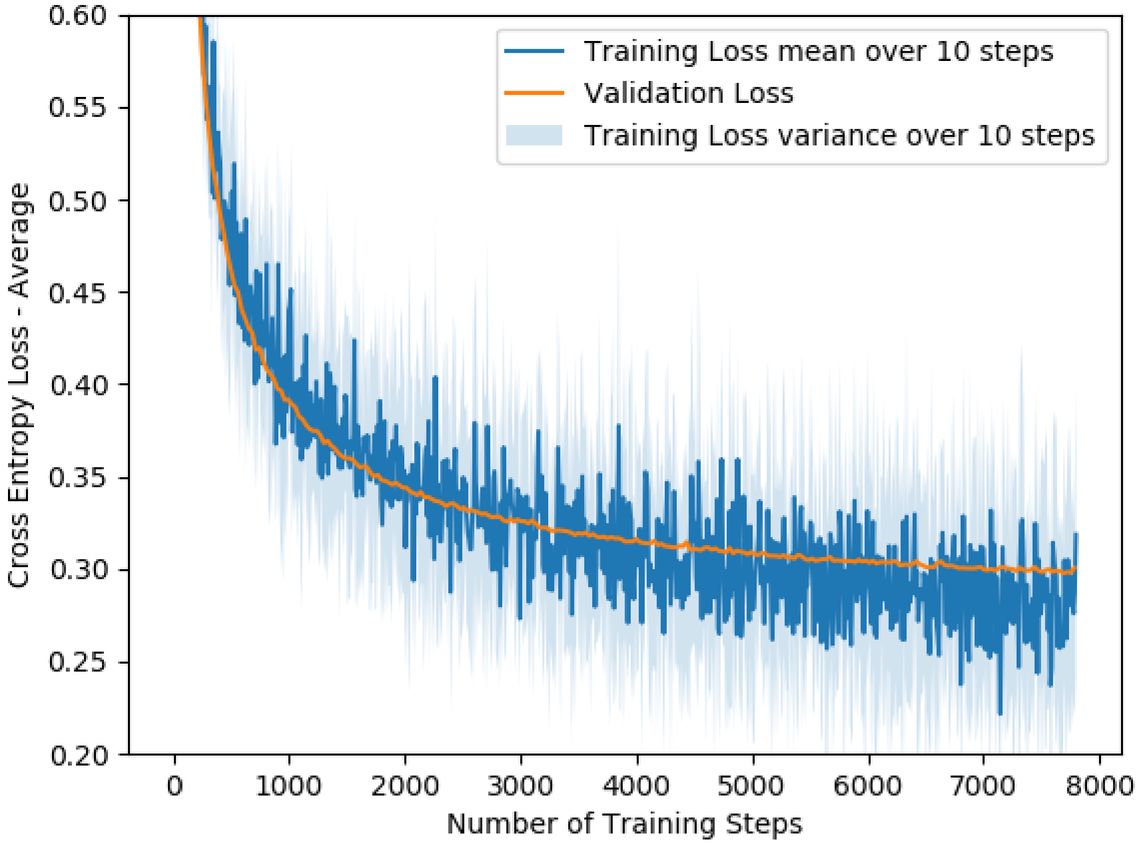
Task 2e)

The validation accuracy gets less spikes when shuffling because the shuffling makes sure that we dont get the periodically minibatch combo, where one minibatch yields a poor validation accuracy for the other minibatch.

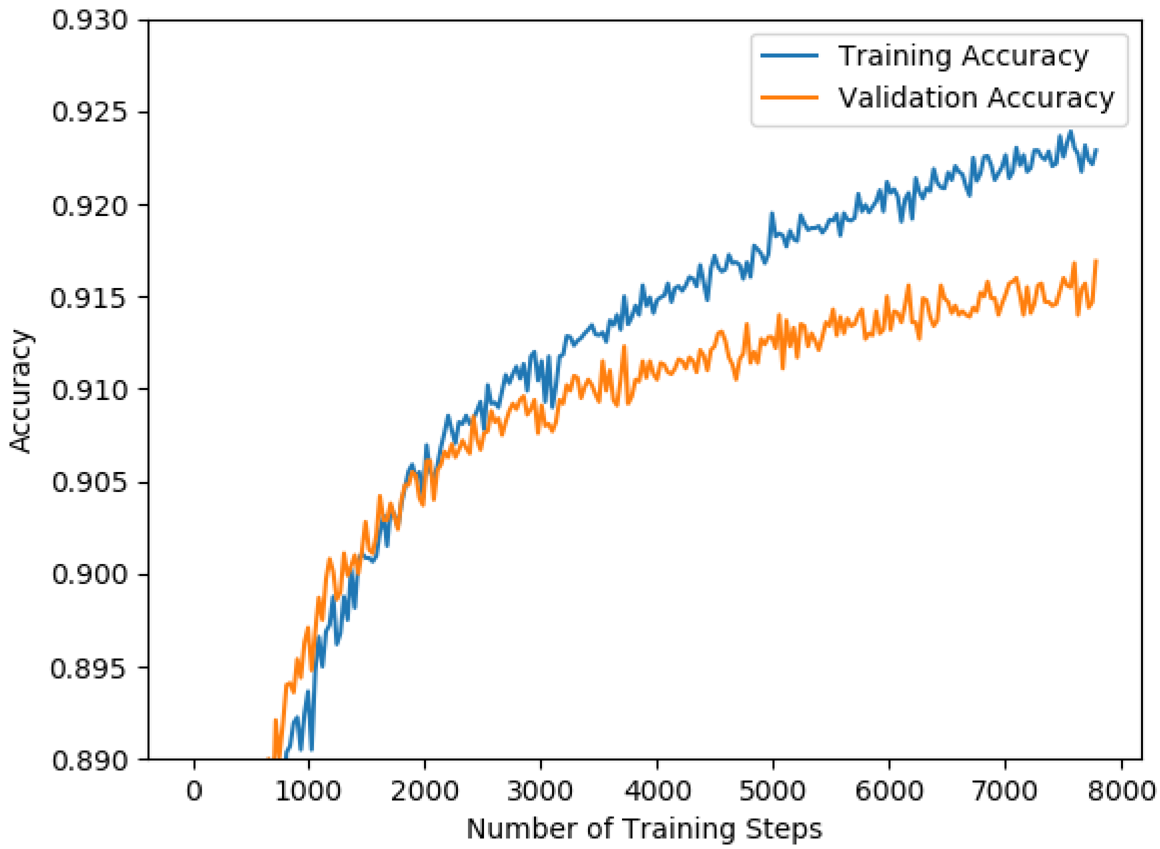


Task 3

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
7000
8000

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
7000
8000

Task 3d)
We can see signs of overfitting in the image above, where the validation accuracy starts flattening before the training accuracy.

Task 4

Task 4a)

4a) $J(w) = C(w) + \lambda R(w)$, $R(w) = \|w\|^2 = \sum_{i,j} w_{i,j}^2$

$(= w^T w)$

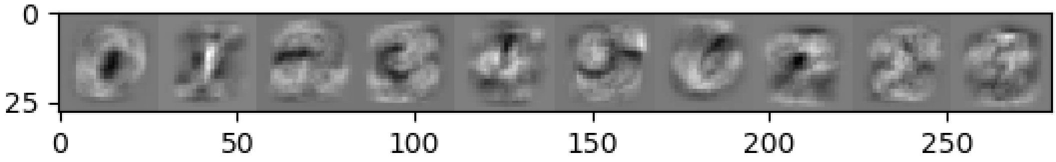
$$\frac{\partial R(w)}{\partial w} = 2w$$

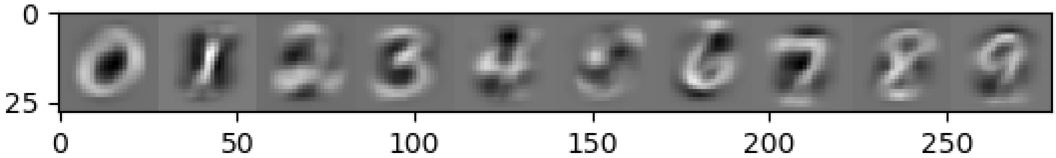
$$\Rightarrow \frac{\partial J(w)}{\partial w} = \frac{\partial C}{\partial w} + \lambda \frac{\partial R}{\partial w} = \frac{\partial C}{\partial w} + 2\lambda w$$

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

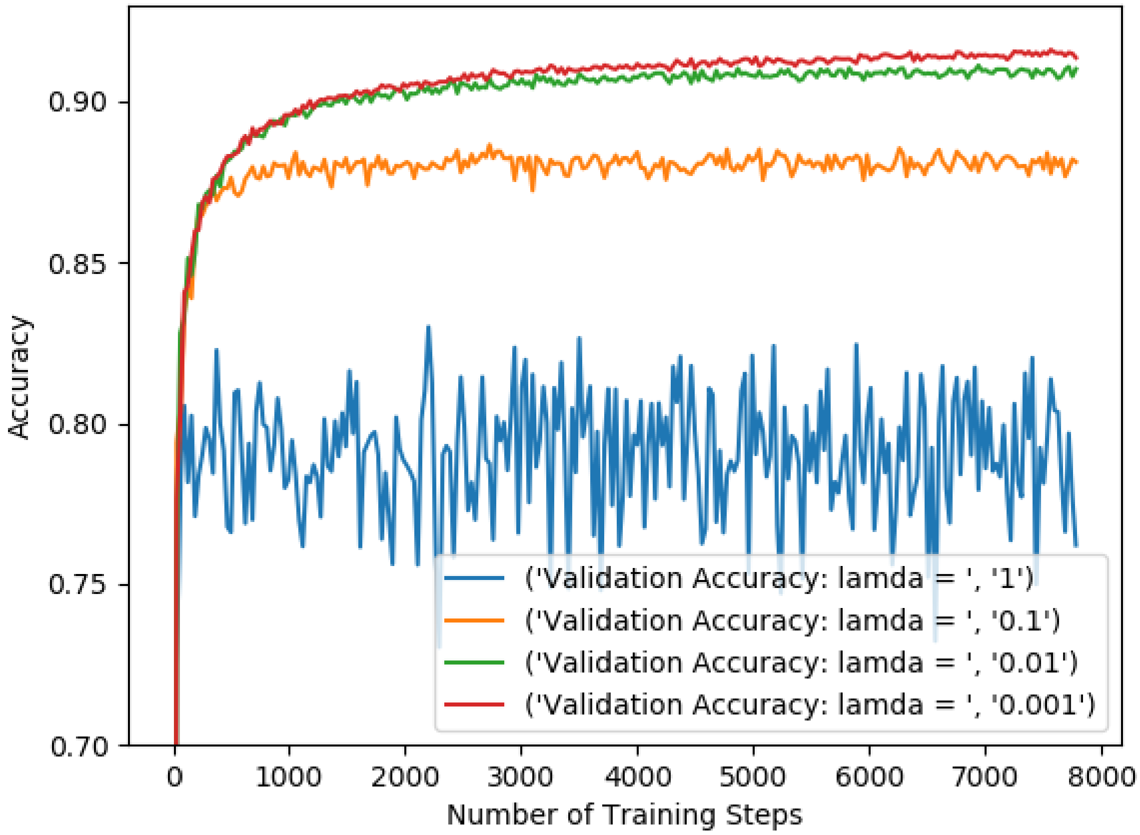
Task 4b)

The weights with lambda=1 is less noisy since the L2-regularisation reduces the complexity of the model, making it less smooth, hence giving a less noisy image.





Task 4c)

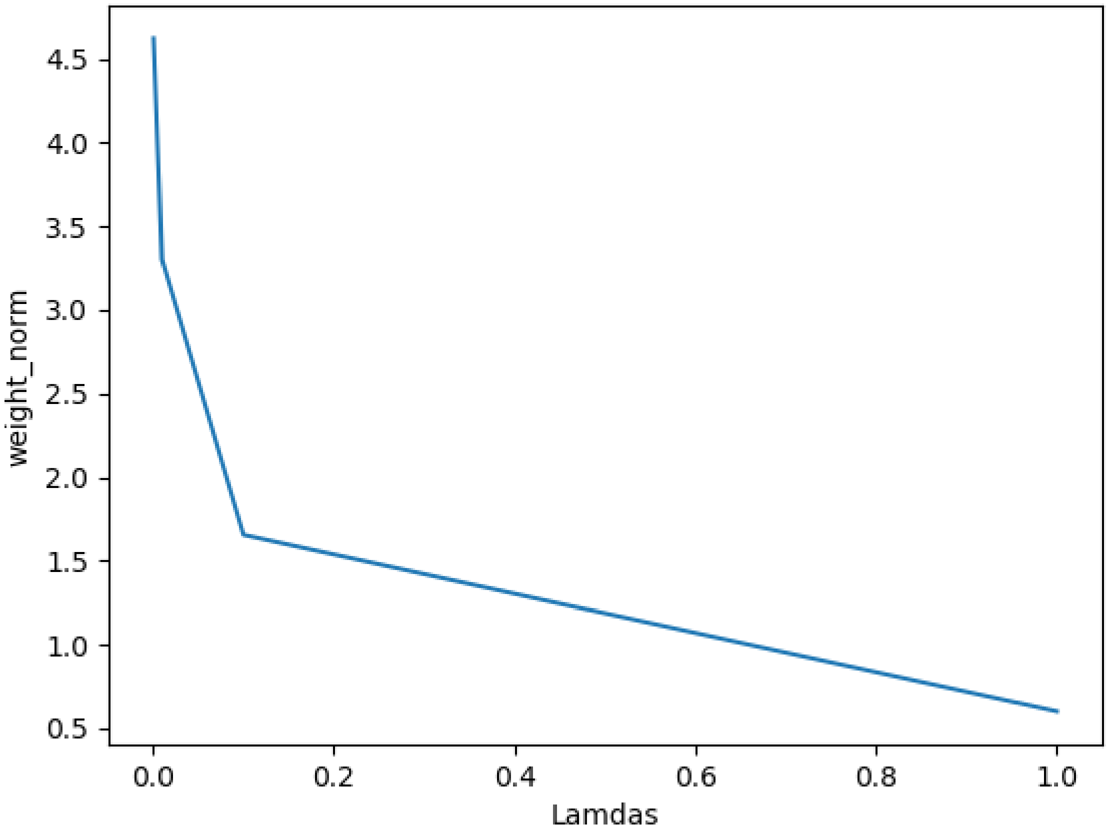


Task 4d)

The validation accuracy drops with lambda since it generalizes a low complexity problem which gives an underfitted result.

Task 4e)

We see that higher lambda yields a lower L2-norm, which is what one could expect since the penalization is higher.



4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
weight_norm
0.0
0.2
0.4
0.6
0.8
1.0
Lamdas