AssignmentReport-Group13

February 3, 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`.

Task 1

Task 1a)

We want to proof that the gradient of the cost function

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

is equal to its gradient.

$$\frac{\partial C^n(w)}{\partial w_i} = -(y^n - \hat{y}^n) x_i^n \tag{2}$$

Hint: To solve this, you have to use the chain rule. Also, you can use the fact that:

$$\frac{\partial \hat{y}^n}{\partial w_i} = x_i^n \hat{y}^n (1 - \hat{y}^n) \tag{3}$$

Proof

Here is presented the proof the this problem.

$$\frac{\partial C^n(w)}{\partial w_i} = -(y^n \frac{1}{\hat{y}^n} \frac{\partial \hat{y}^n}{\partial w_i} + (1 - y^n)(\frac{1}{1 - \hat{y}^n}) \frac{\partial (1 - \hat{y}^n)}{\partial w_i}) \tag{4}$$

applying the **"Hint"** to the previous equation, we obtain the following:

$$\frac{\partial C^n(w)}{\partial w_i} = -(y^n \frac{1}{\hat{y}^n} x_i^n \hat{y}^n (1-\hat{y}^n) + (\frac{1-y^n}{1-\hat{y}^n})(-x_i^n \hat{y}^n (1-\hat{y}^n))) \tag{5}$$

then with a simplification and group by the common factors, we will obtain:

$$\frac{\partial C^n(w)}{\partial w_i} = -(x_i^n (y^n - y^n \hat{y}^n - \hat{y}^n + y^n \hat{y}^n)) \tag{6}$$

which hereby proof the equation:

$$\frac{\partial C^n(w)}{\partial w_i} = -(y^n - \hat{y}^n) x_i^n \tag{7}$$

Task 1b)

We want to prove that:

$$\frac{\partial C^n(\omega)}{\partial \omega_{kj}} = -x_j^n (y_k^n - \hat{y}_k^n) \tag{8}$$

I will omit the n notation, as during this proof, we always talk about the n. sample.

Proof

We know that:

$$C(\omega) = -\sum_{k'=1}^{K} y_{k'} ln(\hat{y}_{k'}) \tag{9}$$

We break up the summation into two cases, if $k' = k$ the respective derivative equals:

$$\frac{\partial y_k ln(\hat{y}_k)}{\partial \omega_{kj}} = x_j y_k (1 - \hat{y}_k) \tag{10}$$

if $k' \neq k$, for every k' the derivative equals:

$$\frac{\partial y_{k'} ln(\hat{y}_{k'})}{\partial \omega_{kj}} = -y_{k'} (\hat{y}_k x_j) \tag{11}$$

the summation of the elements in eq. (11):

$$\sum_{k'=1, k' \neq k}^{K} -(y_{k'} \hat{y}_k x_j) = \hat{y}_k x_j (y_k - 1) \tag{12}$$

since:

$$\sum_{k'=1}^{K} y_{k'} = 1 \tag{13}$$

substitute eq. (10) and eq. (12) to eq. (8), and multiply out x_j:

$$\frac{\partial C(\omega)}{\partial \omega_{kj}} = \frac{\partial(-\sum_{k'=1}^{K} y_{k'} ln(\hat{y}_{k'}))}{\partial \omega_{kj}} = -(x_j(y_k - \hat{y}_k y_k + \hat{y}_k y_k - \hat{y}_k) = -x_j(y_k - \hat{y}_k) \tag{14}$$

which is our original statement, eq. (8) .

Proof of eq. (10)

Using the basic division rule of logarithm function, we can split the left side to two parts:

$$\frac{\partial y_k ln(\hat{y}_k)}{\partial \omega_{k,j}} = y_k(\frac{\partial ln(e^{z_k})}{\partial \omega_{k,j}} - \frac{\partial ln(\sum_{k'}^{K} e^{z_{k'}})}{\partial \omega_{k,j}}) \tag{15}$$

since

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{k'}^{K} e^{z_{k'}}} \tag{16}$$

After using the chain-rule, we get:

$$\frac{\partial y_k ln(\hat{y}_k)}{\partial \omega_{k,j}} = y_k(\frac{\partial z_k}{\partial \omega_{k,j}} - \frac{1}{\sum_{k'}^{K} e^{z_{k'}}} e^{z_k} \frac{\partial z_k}{\partial \omega_{k,j}}) \tag{17}$$

We know that:

$$z_k = \sum_{i}^{I} w_{k,i} x_i \tag{18}$$

so:

$$\frac{\partial z_k}{\partial \omega_{k,j}} = x_j \tag{19}$$

as all the other elements of the summation are independent from the variable.

Substitute eq. (16) and eq. (18) into eq. (17):

$$\frac{\partial y_k ln(\hat{y}_k)}{\partial \omega_{k,j}} = y_k(x_j - \hat{y}_k x_j) \tag{20}$$

which is eq. (10).

Proof of eq. (11):

Pretty similarly as eq. (15):

$$\frac{\partial y_{k'} ln(\hat{y}_{k'})}{\partial \omega_{k,j}} = y_{k'}(\frac{\partial z_{k'}}{\partial \omega_{k,j}} - \frac{\partial ln(\sum_{k''}^{K} e^{z_{k''}})}{\partial \omega_{k,j}}) \tag{21}$$

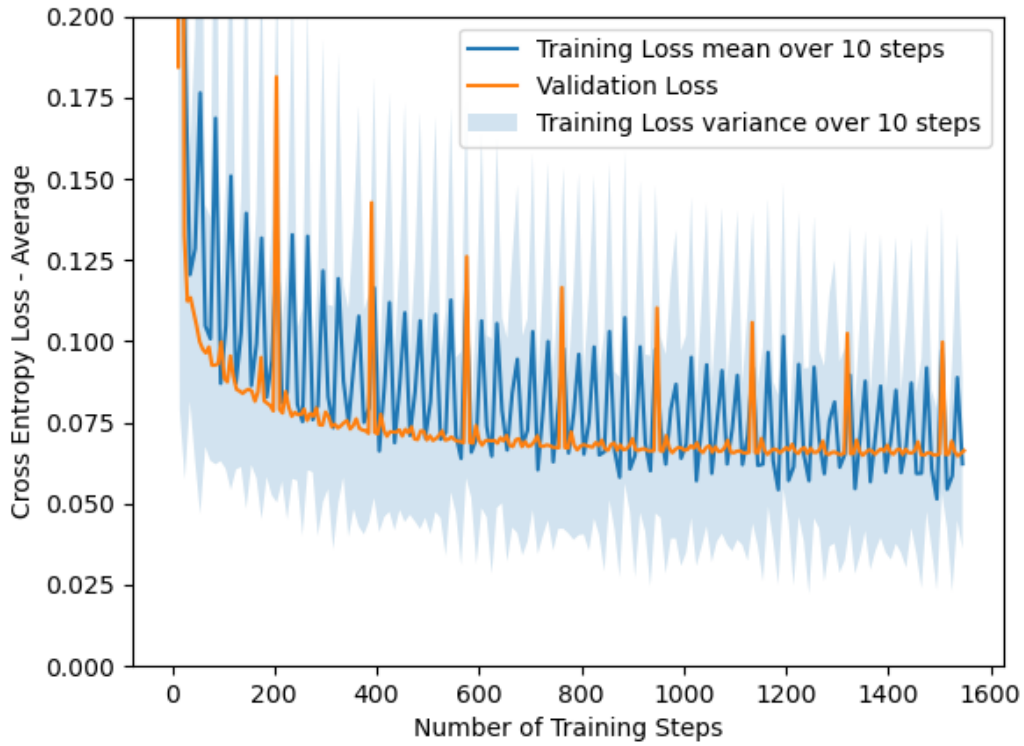
The second element of the left side is the same as in the previous proof, the first element is zero as $k' \neq k$, therefore:

$$\frac{\partial y_{k'} ln(\hat{y}_{k'})}{\partial \omega_{k,j}} = y_{k'}(0 - \hat{y}_k x_j) \tag{22}$$

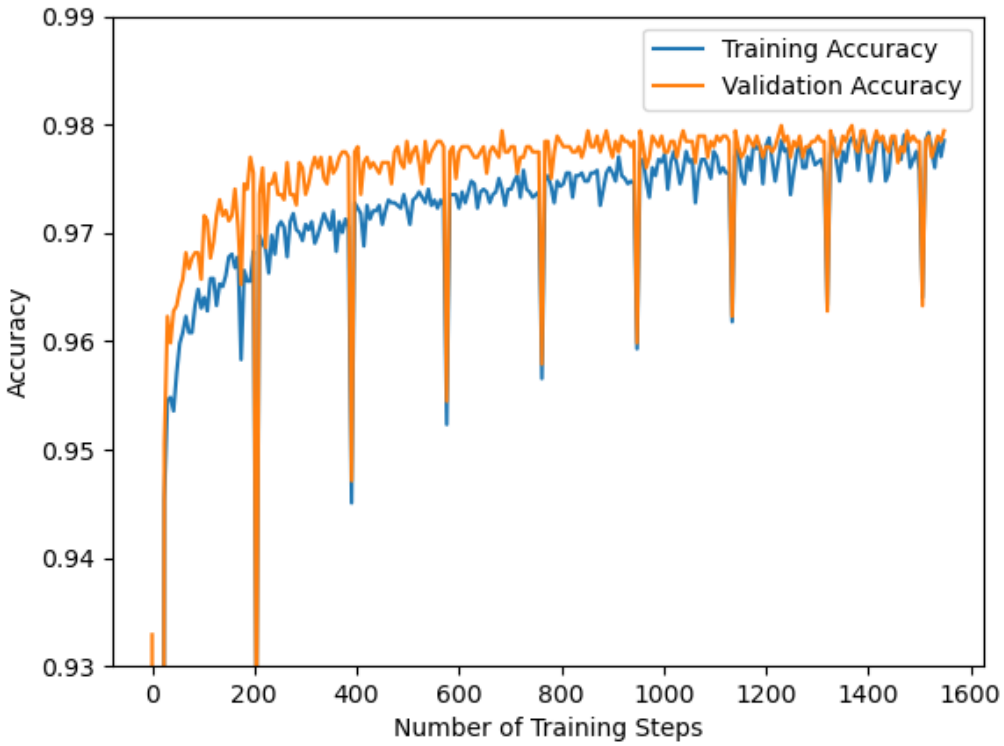
which is eq. (11).

Task 2

Task 2b)



Task 2c)

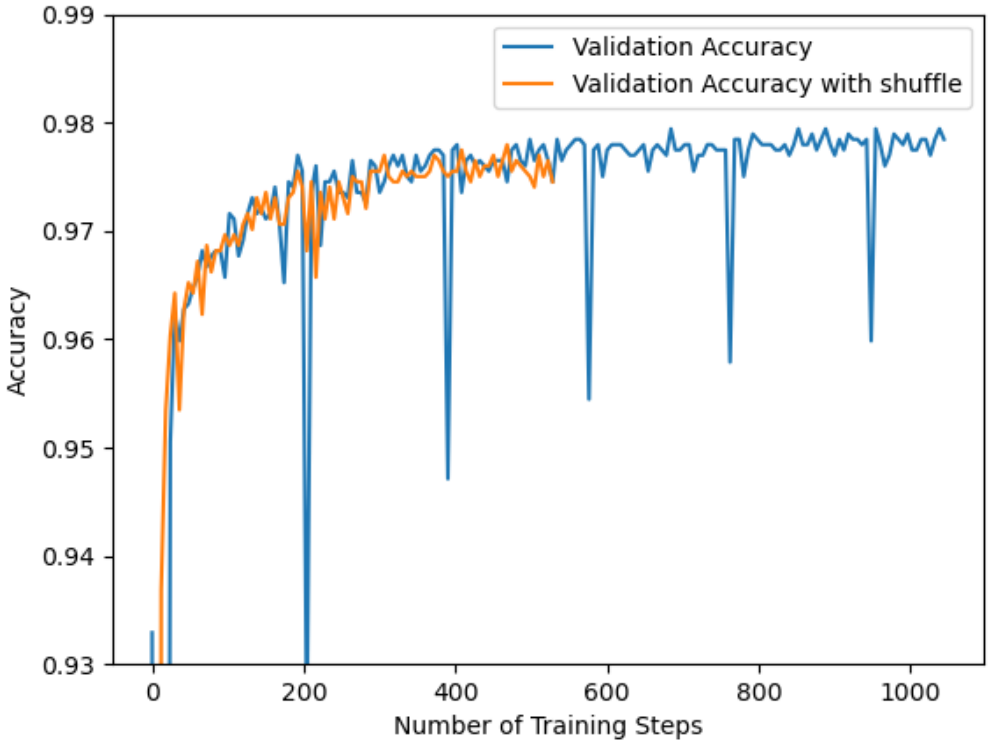


Task 2d)

With the following implementation, the training stops at **33** epochs

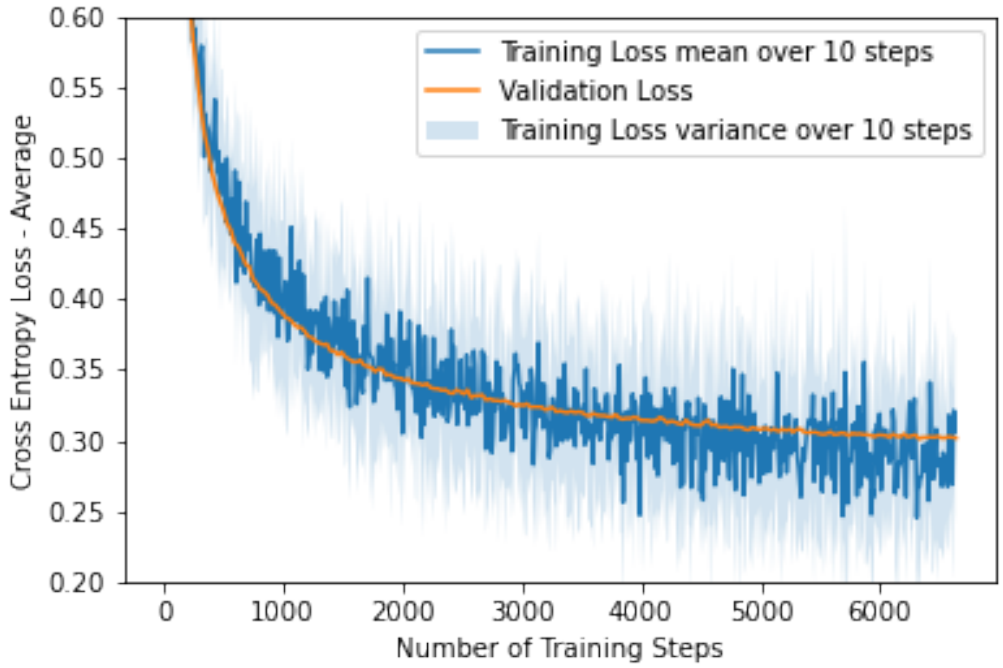
Task 2e)

We can notice that due to shufflin the spikes are now gone. This could suggest us that the spikes were caused by a bad recurrent batch that will worse the accuracy of the model. Now with the shuffling the training data of the bad batch is shuffled across the entire dataset.

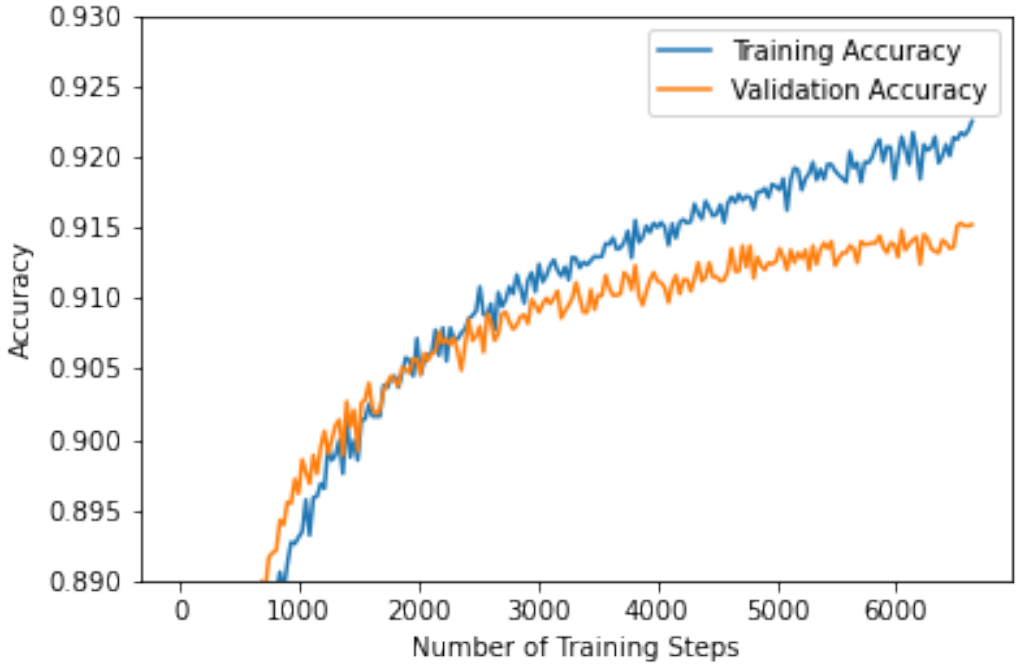

Validation Accuracy
Validation Accuracy with shuffle
Accuracy
Number of Training Steps
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0
200
400
600
800
1000

Task 3

Task 3b)



Task 3c)



Task 3d)

In the above image, we can notice that the accuracy for the validation set has reach a plateu while the training accuracy is still increasing. The accuracies are splitting apart across training steps, this will suggest us that the model started to overfit, which led an increase of training accuracy over the validation accuracy. In this way, the model has lost a little of generalization ("*The model is memorizing not learning*")

Task 4

Task 4a)

Using the results of Task 1b):

$$\frac{\partial J(\omega)}{\partial \omega_{k,j}} = \frac{\partial C(\omega)}{\partial \omega_{k,j}} + \frac{\partial \lambda R(\omega)}{\partial \omega_{k,j}} = -x_j(y_k - \hat{y}_k) + 2\lambda\omega_{k,j} \tag{23}$$

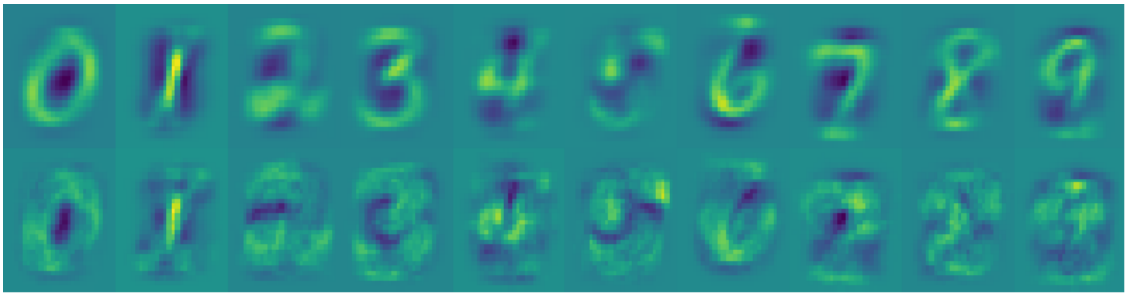
since:

$$R(\omega) = \sum_{k',j'} \omega^2_{k',j'} \tag{24}$$

and only one element of the summation, when $k' = k$ and $j' = j$, is dependent from $\omega_{k,j}$.

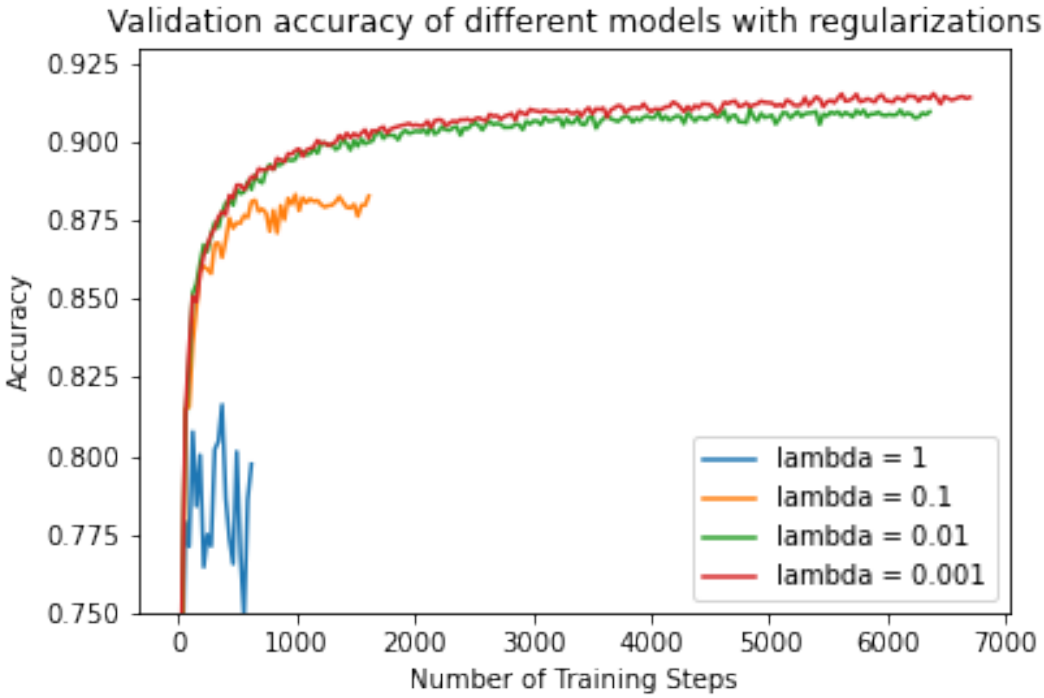
Task 4b)

Weights with, and without regularization (respectively):



The images with the strong regularization, are less noisy as L2 regularization tries to set every weight to zero, and only the most determinative weights will have a value other than zero. Moreover, we would like to stress once more that the L2 regularization its a method to reduce the complexity of the model by penalyzing the loss during the training. This could be grasped just by looking at the plotted weights (image above), where thank to the regularization factor (lambda = 1) the weights appear less variable then the weights of the model with no L2 regularization (hence less complex).

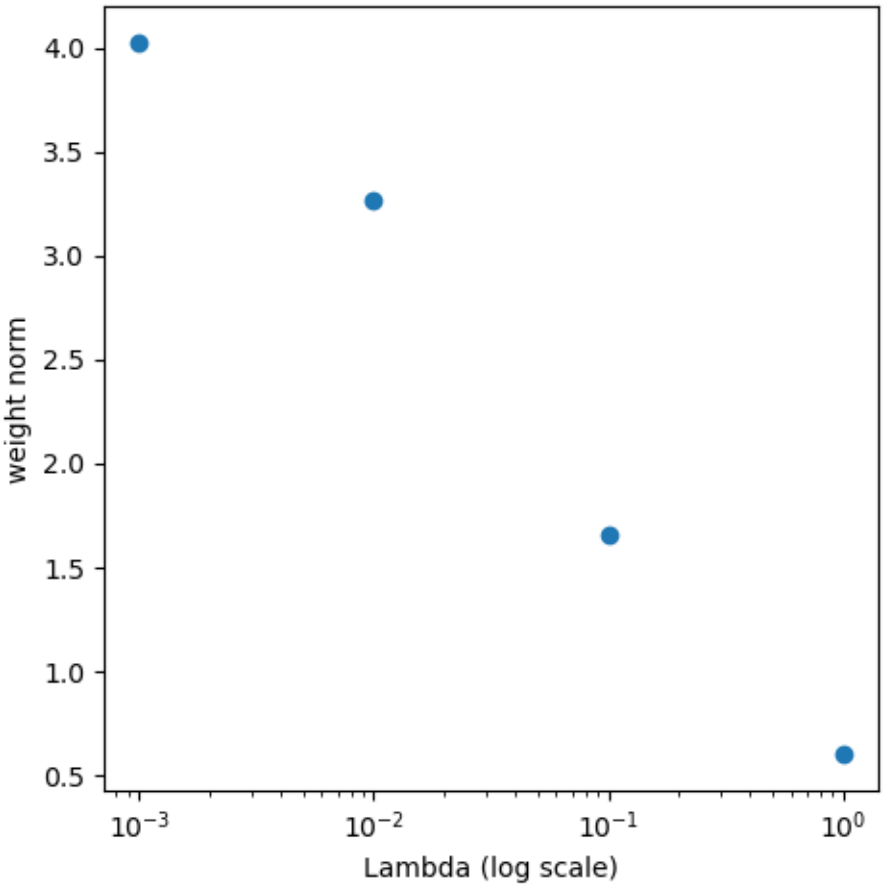
Task 4c)



Task 4d)

Because the model is now optimized according two constraints (accuracy, and weight size), so naturaly it will be worse if we examine just the accuracy. From another side: the regularization usually reduce the complexity of the model, as lot of weights will be zeros, and the new 'smaller' model won't be as accurate as the larger one.

Task 4e)



The weights of the models are decreasing with increasing values of lambda.