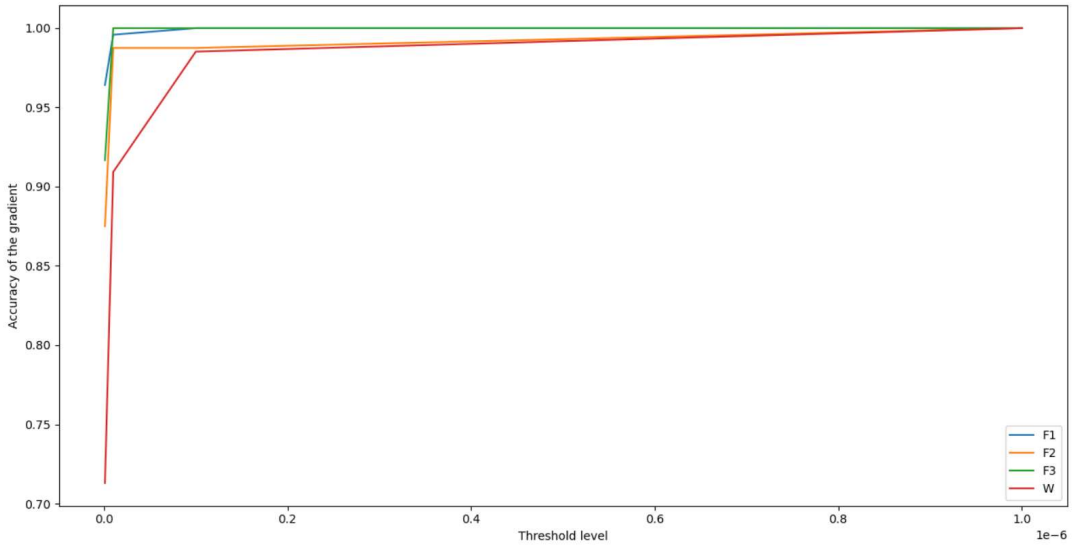
Assignment 3 bonus points

ssbl@kth.se

The first bonus point implementation was to develop a neural network which could increase the amount of convolutional layers implemented.

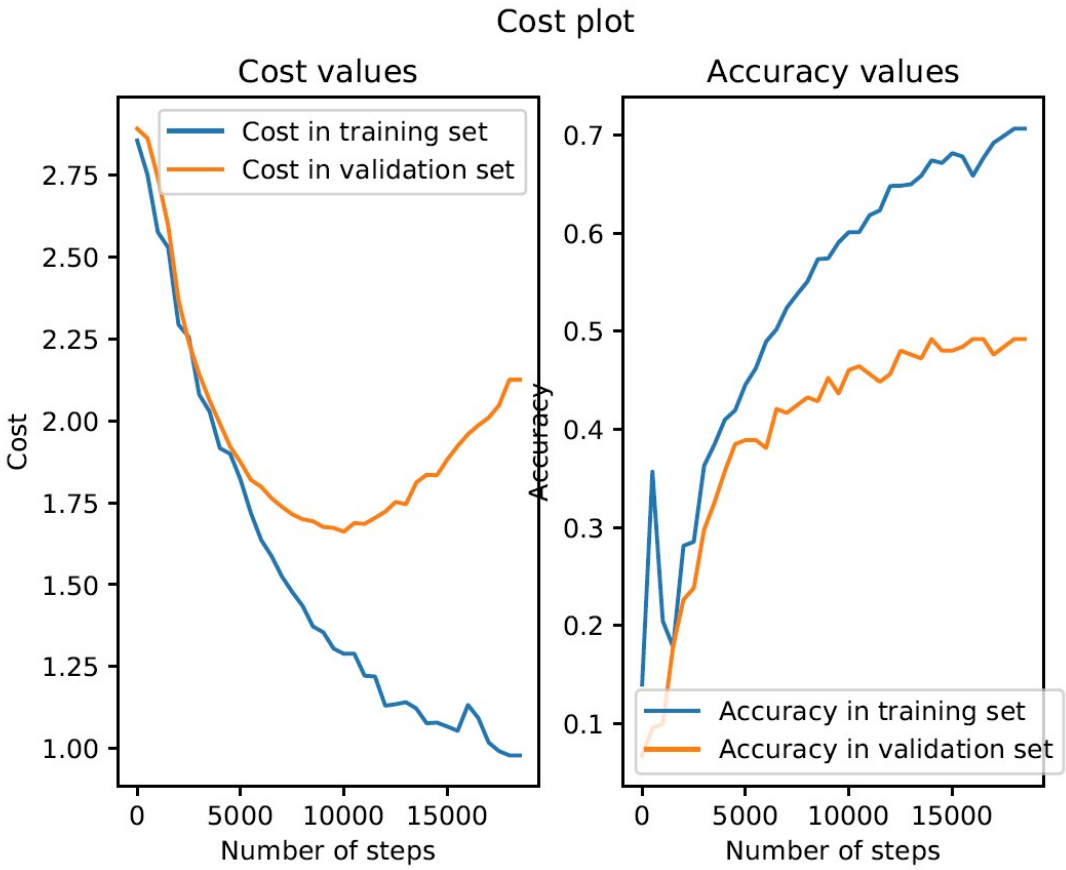
The first part was to check if this generalisation had been correctly implemented. In order to do this, the back-propagation was compared with numerical approximations. The test was done with three convolutional layers and a fully connected layer. The three layers had 5, 4 and 3 filters per layer respectively. The accuracies obtained were the following:



This graph has three convolutional layers plus the fully connected layer.

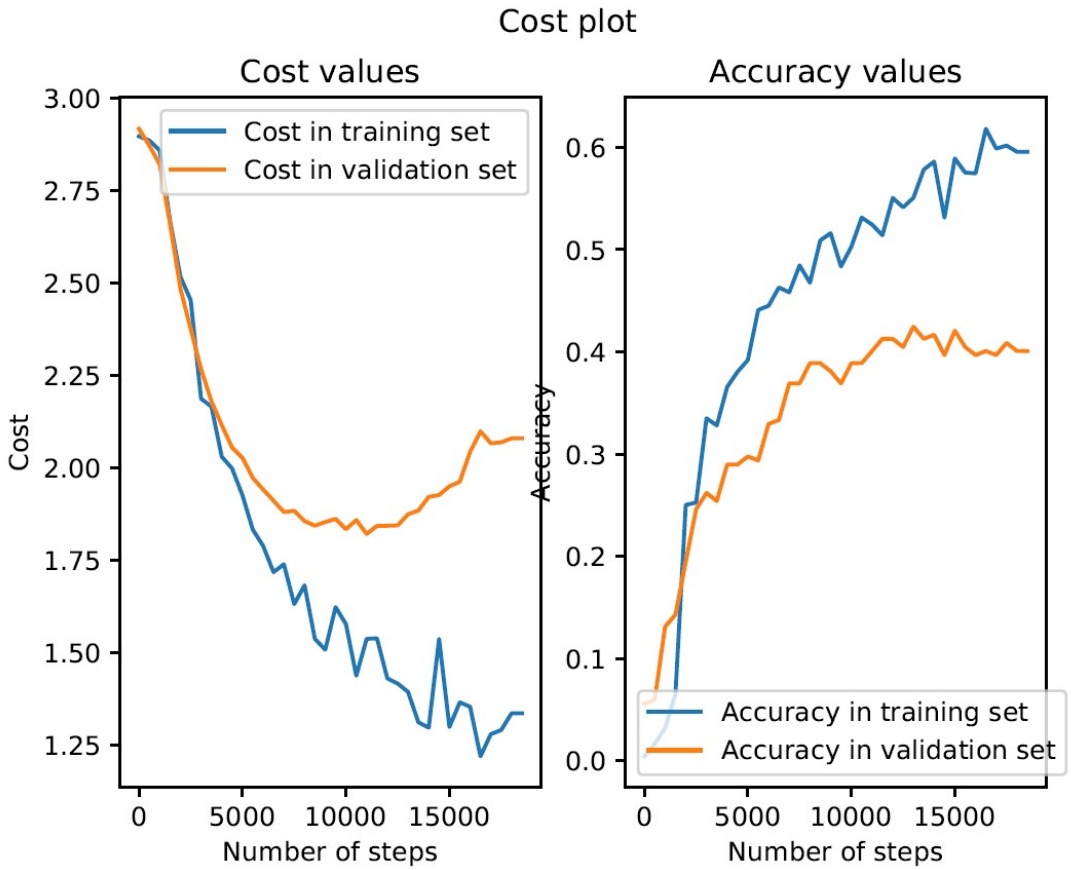
After this was done, a graph search was done in order to be able to know which parameters gave the best accuracies.

The number of layers tested was: 2, 3, 4 and 5, with 10, 20 or 30 filters per layer and a k of 4 or 5 for each filter. The best parameters were: 4 layers, with 20 filters per layer and a k of 4 per filter. With this set-up a final accuracy of 49% was achieved in the validation dataset. The cost and accuracies obtained were:



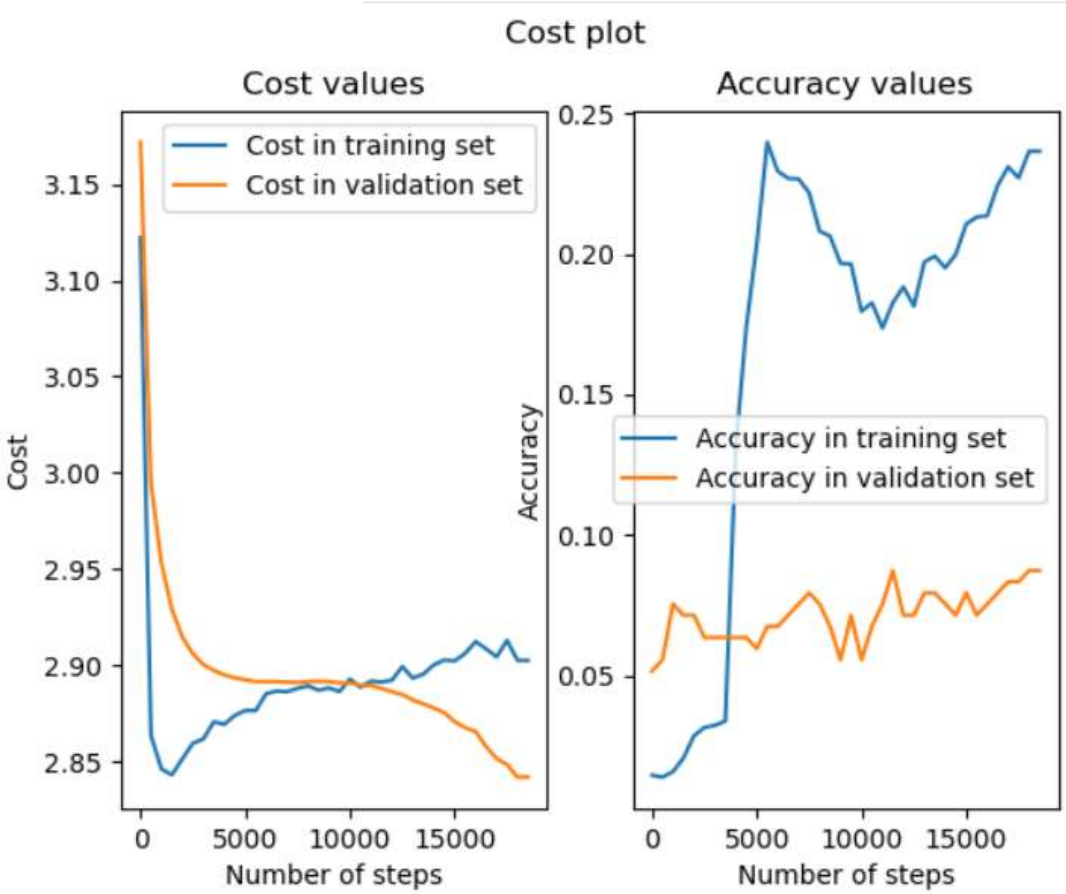
As we can see, there is an improvement in accuracies of around 10%. Also, this demonstrates that the deeper does not always means it is better: 5 convolutional layers did not outperform 4 layers.

The second bonus point implemented was to introduce the independent b in each layer of both the convolutional and the fully connected layers. The new network was tested with the same hyperparameters as the ones previously obtained: 4 layers, 20 filters per layer with a depth of 4. The results were the following:



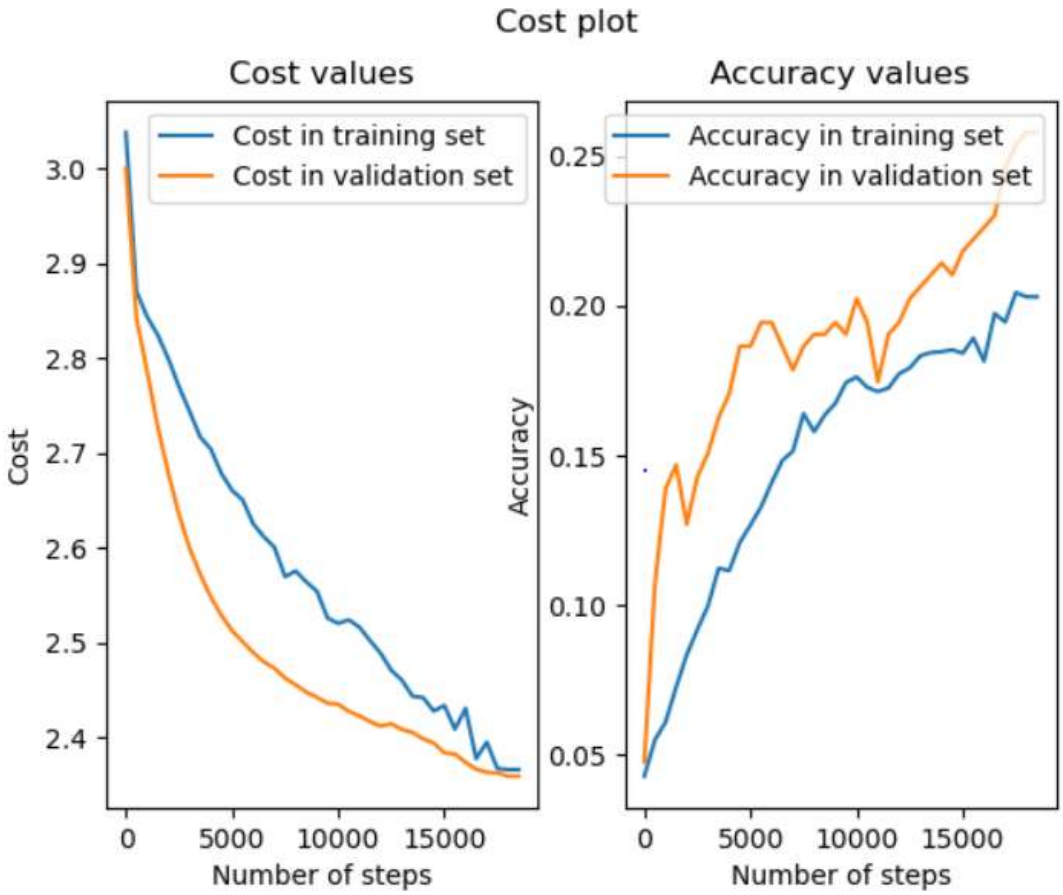
The accuracy led to a similar value as for two layers.

The last bonus point was to introduce strides and paddings. Once correctly implemented, the following architecture was tested: 2 layers, with 2 filters per layer, depths of 5 for the first layer and 3 for the second layer and strides of 2 in the first layer and 7 in the second and only padding in the second convolutional layer with a padding of 1 (chosen so the network could work). The following accuracy and loss functions were obtained:



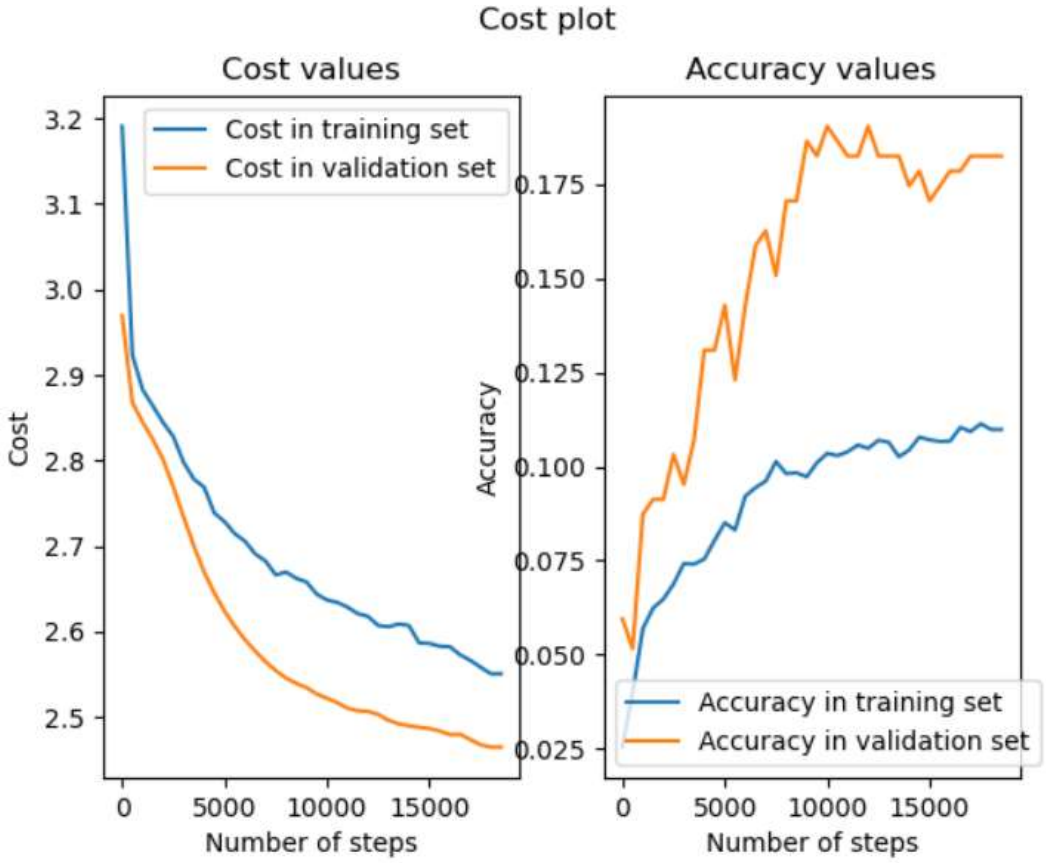
Not too much learning was done in the validation dataset as the final output of each image ended up with two inputs before the fully connected layer, although it can be seen that some learning can be done in the training set.

Another test done was with: 2 layers, 10 filters in each layers, depths of 5 and 4 respectively, strides of 2 in each layers and a padding of P = 2 in the second layer. The result was the following:



As we can see, now the learning is more accurate as there are more input nodes for the fully connected layer

Another experiment was repeated with 3 layers, 7 filters per layer, depths of 5, 2 and 2 and strides of 2, 2, 1 with a 0, 1 and 0 padding. The results are the following:


Cost plot
Cost values
Accuracy values
Cost in training set
Cost in validation set
Accuracy in training set
Accuracy in validation set
Cost
Accuracy
Number of steps
Number of steps
3.2
3.1
3.0
2.9
2.8
2.7
2.6
2.5
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0
5000
10000
15000
0
5000
10000
15000