Assignment 2 - Deep Learning DD2424

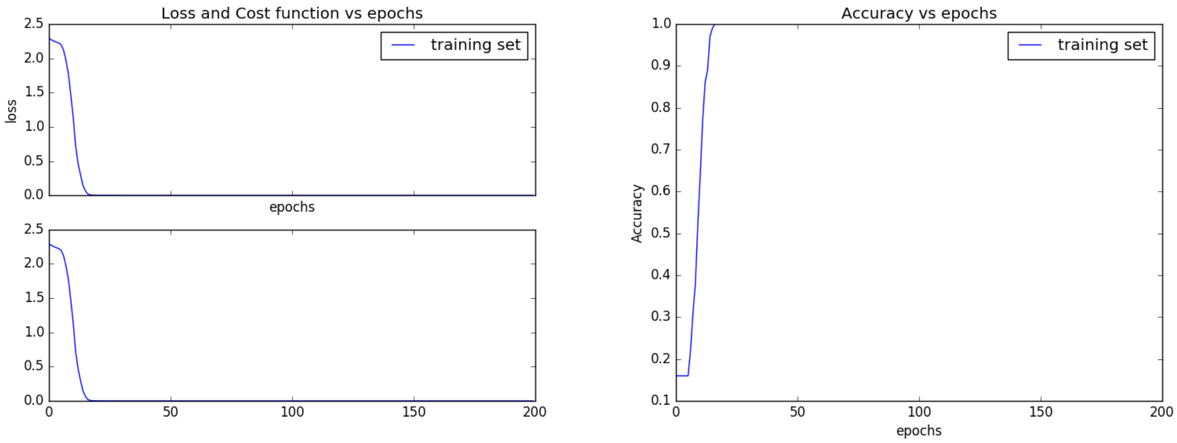
Erifili Ichtiaroglou

April 2017

1 Exercise 1-4

For the second assignment we had to implement a 2-layer neural network and train it. The first step was to calculate the gradients for softmax and Relu function for each layer. The calculation of the gradients was correct and checked with the help of the function ComputeGradsNumSlow(). We also managed to overfit the model when training it with 100 images for 200 epochs as Figure 1 shows.

Figure 1: Mini-batch learning with $\lambda = 0$, $\eta = 1.0$, $epochs = 200$, $batch = 100$.



After that we implemented the mini batch and we added the momentum term to speed up the training. The training was quite faster when using the momentum term, as the algorithm converged faster that before. For the next part we had to find the optimal values for the hyper-parameters. We run a grid search (8x8) and we sampled the values of η and λ uniformly from the intervals $\{10^{-5}, 1\}$ and $\{10^{-8}, 1\}$ respectively. We trained the network for 5 epochs, using the first batch as training set and the second as the validation set. After that we run again the grid search but in a narrower interval as we observed that the biggest accuracy on the validation set occurred for η between $\{10^{-3}, 10^{-1}\}$ and λ between $\{10^{-8}, 10^{-3}\}$. For these values of the parameters we found the three pairs that gave us the best accuracy at the validation set and we trained the network for each pair for 10 epochs. Table 1 shows the results after the training.

From Table 1 and the results of our 64 previous trials with different pairs of hyper-parameters we concluded that the best η exists in the interval $\{0.01, 0.05\}$ and the best λ in the interval $\{6 * 10^{-6}, 10^{-3}\}$. We run one last test (64 different pairs) for the values of hyper-parameters around the means of these intervals to examine more thoroughly this region. Table 2 shows the accuracy on validation set that we managed to achieve with these settings.

η	λ	batch$_s ize$	epochs	accuracy
0.0510	0.00022	100	10	43.67%
0.0247	$6 * 10^{-7}$	100	15	44.17%
0.0282	0.00022	100	10	43.8%

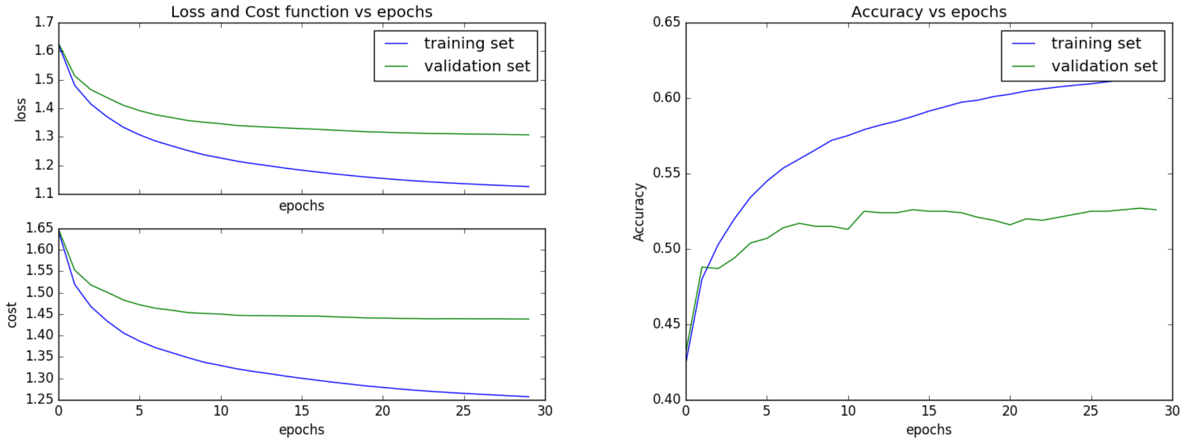
Table 1: The accuracy of the validation set after training the network with the best found hyper-parameters.

η	λ	batch$_s ize$	epochs	accuracy
0.0228	0.00078	100	15	44.99%
0.025	0.00025	100	15	44.11%
0.037	0.0004	100	10	43.78%

Table 2: The accuracy of the validation set after training the network with the best found hyper-parameters.

From all the trials we completed we found that the best results were given for $\eta = 0.0228$ and $\lambda = 0.00078$. For these chosen hyper-parameters we trained the model using all the 5 batches as the training set, except the last 1000 images that were used as the validation set. After training the model we examined the network's performance for the test batch. Figure 2 presents the graphs that occurred after this training. In order to train the model we used the momentum term and the eta-decaying method. Without using this methods the accuracy we got was smaller, as expected because these two methods are used to increase the performance of the network.

Figure 2: Mini-batch learning with $\lambda = 0.00078$, $\eta = 0.0228$, $epochs = 30$, $batch = 100$
accuracy at the validation set 52.7%
accuracy at test set 52.05%



2 Exercise 5

For this part of the exercise, in order to optimize the performance of the network we tried some more tricks. We trained the model for 60 epochs and we kept the weights that gave us the best accuracy for the validation set. In addition we increased the number of hidden nodes from 50 to 100 and we incorporated Dropout regularization technique for preventing overfitting.We tried 2 different tests with dropping out 5% and 10% of the network's units. Figure 3 and 4 show the results after incorporating these techniques.

Figure 3: Mini-batch learning with $\lambda = 0.00078$, $\eta = 0.0228$, $epochs = 60$, $batch = 100$
percentage of drop out units 5%
accuracy at the validation set 55.8%
accuracy at test set 54.55%.
Best model occurred at epoch 14.

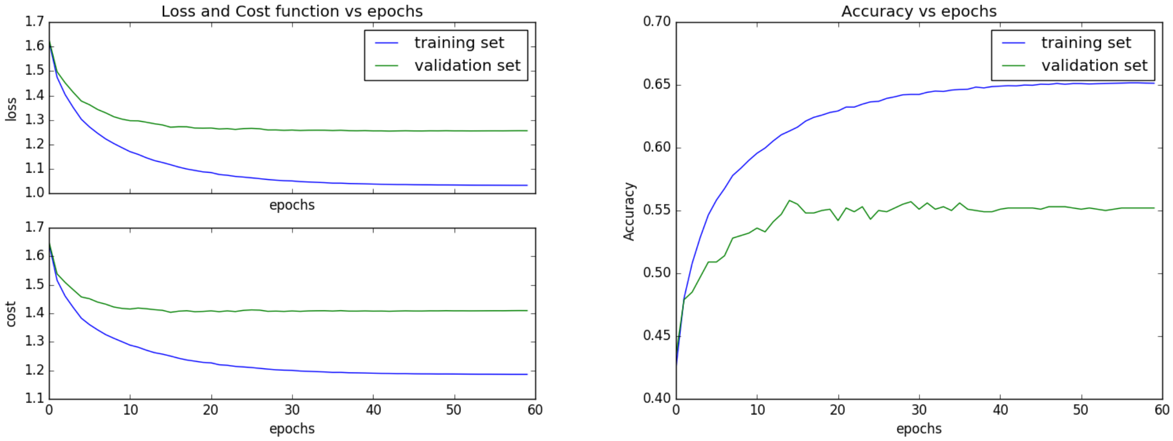
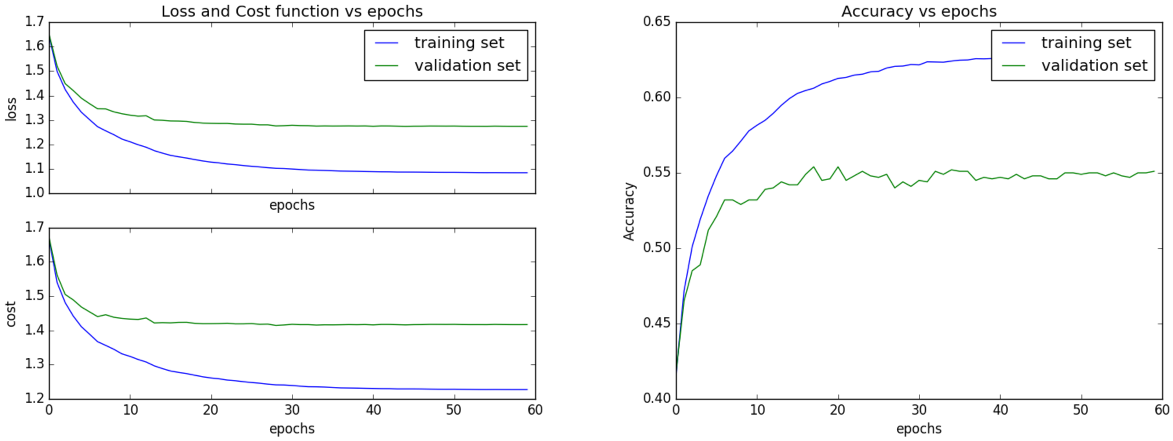


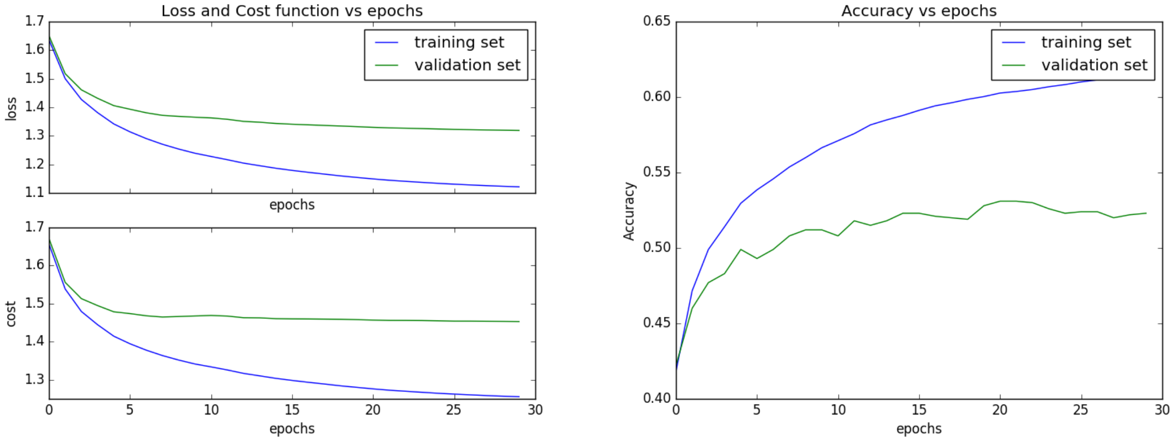
Figure 4: Mini-batch learning with $\lambda = 0.00078$, $\eta = 0.0228$, $epochs = 60$, $batch = 100$
percentage of drop out units 10%
accuracy at the validation set 55.4%
accuracy at test set 53.8%.
Best model occurred at epoch 17.



3 Exercise 6

For the last part of the Assignment we used another activation function different from ReLu. We implemented LeakyReLu and we tried a lot of different parameters when training the model. Figure 5 shows the results of the network when using LeakyReLu and the same parameters as we did for ReLu activation function, in order to observe any differences between these 2 implementations.

Figure 5: Mini-batch learning with $\lambda = 0.00078$, $\eta = 0.0228$, $epochs = 30$, $batch = 100$
accuracy at the validation set 53.1%
accuracy at test set 51.57%.



We observe that there is not a very big difference between the results we got from the 2 activation function when it comes to the accuracy of the model. That is probably because for a 2 layer network the difference in implementation of the 2 activation functions is small.