313326985 Shahar Shcheranski
206172686 Sarit Hollander

Assignment Report

Configuration:

Layers_dims = [784, 20, 7, 5, 10]

Learning_rate = 0.009

Num_ iterations = 100

Batch_size = 256

Stopping criterion = when the improvement < 0.001 →

[np.abs(costs[-2] - costs[-1]) > stopping_rate] and the costs are calculated every 100 iterations (num_iterations)

Results (batchnorm is off):

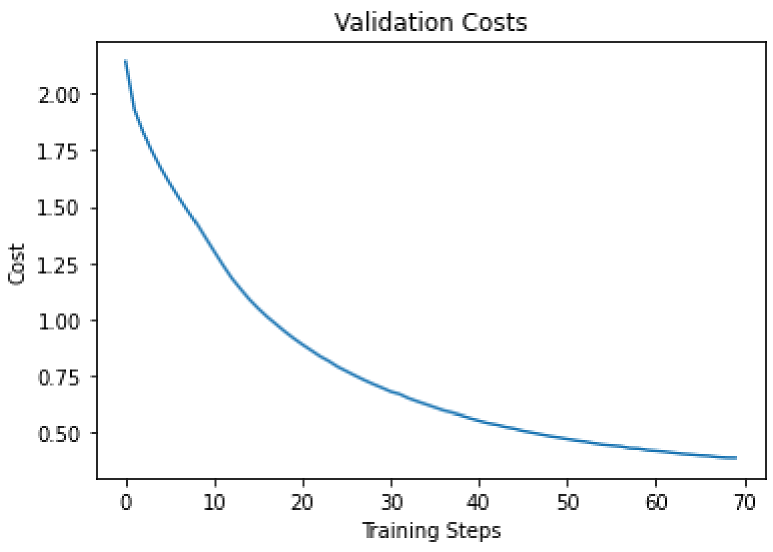
Number of epochs: 37 and 44 training steps

Train accuracy: 90.95625000000001%

Validation accuracy: 90.46666666666667%

Test accuracy: 90.49000000000001%

Running Time: 228.07242965698242



313326985 Shahar Shcheranski
206172686 Sarit Hollander

<u>Results (batchnorm is on)</u>:

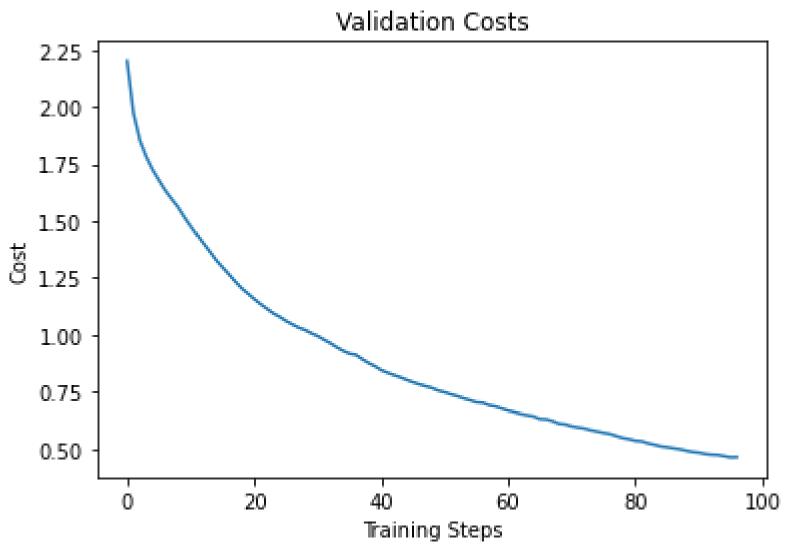
Number of epochs: 51 epochs and 112 training steps

Train accuracy: 89.57083333333333%

Validation accuracy: 88.84166666666667%

Test accuracy: 89.13%

Running Time: 326.4042909145355



313326985 Shahar Shcheranski
206172686 Sarit Hollander

Bonus – dropout functionality:

The changes in the code: we added to the linear activation function the option to use the dropout functionality whenever the activation function is relu. In the dropout function (both in forward propagation and backward) the activation is changing according to the dropping rate. We specify the probability at which outputs of the layer are dropped out, or inversely, the probability at which outputs of the layer are retained.

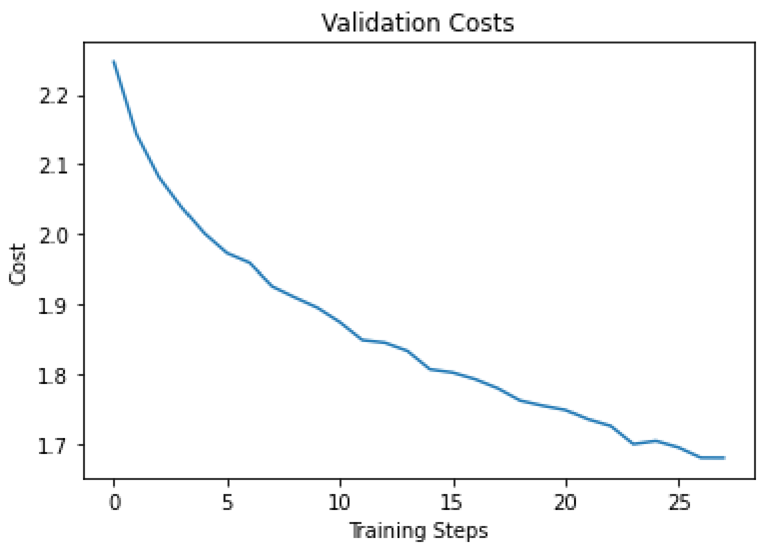
Number of epochs: 14 epochs and 168 training steps

Train accuracy: 40.5625%

Validation accuracy: 40.733333333333334%

Test accuracy: 40.68%

Running Time: 95.6027467250824



The dropout functionality is most affective on larger networks or when we train the network for a very long time. Here, our network is not large enough to have an impact and improve the results, as we see in these poor results.