Assignment 3 - option 1 - DD2424

Alexandra Hotti

May 2019

1 Relative Error between Analytical and Numerical Gradients

This section contains the relative computations between the numerical and learned analytical gradients computed for the weights and biases, as well as the scale γ and shift β used during batch normalization. The analytical gradients were computed using the function: *ComputeGradsNumSlow*. The relative errors was computed using Equation 1.

$$\frac{|g_a - g_n|}{max(\epsilon, |g_a| + |g_n|)} \tag{1}$$

The relative errors were computed both for a 2 layer network with 50 hidden nodes and a 3 layer network with 50 and 50 hidden nodes with $\lambda = 0.005$. The data set consisted of 10 000 points, with 20 dimensions and mini batches set to 100.

The relative errors for the 2 layer networks were less than $1e-7$ and the 3 layer networks were smaller than $1e-6$ which is satisfactory [1]. Thus, the gradient computations were deemed bug-free.

2 3-layer networks

Below are the results from training a 3 layer network with cyclical learning. After each epoch the data points were shuffled. In the second section batch normalization was used and in the first it was not. The weights and biases of the network with batch normalization was initialized using HE-initialization and the other network used Xavier-initialization. The following parameter setting was used in both of these cases:

$$epochs = 20, \lambda = 0.005, n_s = 2250, m_1 = 50, m_2 = 50, n_{cycles} = 2$$

Were m_1 and m_2 are the number of hidden nodes in the first and second layers.

2.1 A 3-layer network without batch normalization

Accuracy on held out test set: 52.81%

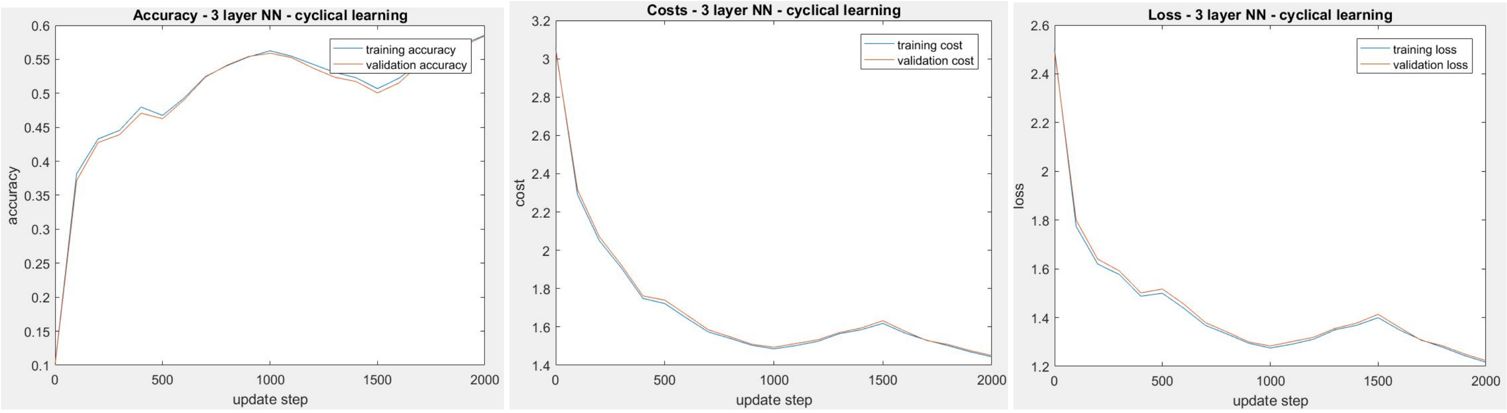

Figure 1: Accuracy, Cost and Loss plots for the training and validation data sets for the 3-layer Neural Network with Cyclical Learning Rate.

2.2 A 3-layer network with batch normalization

Accuracy on held out test set: 53.64%

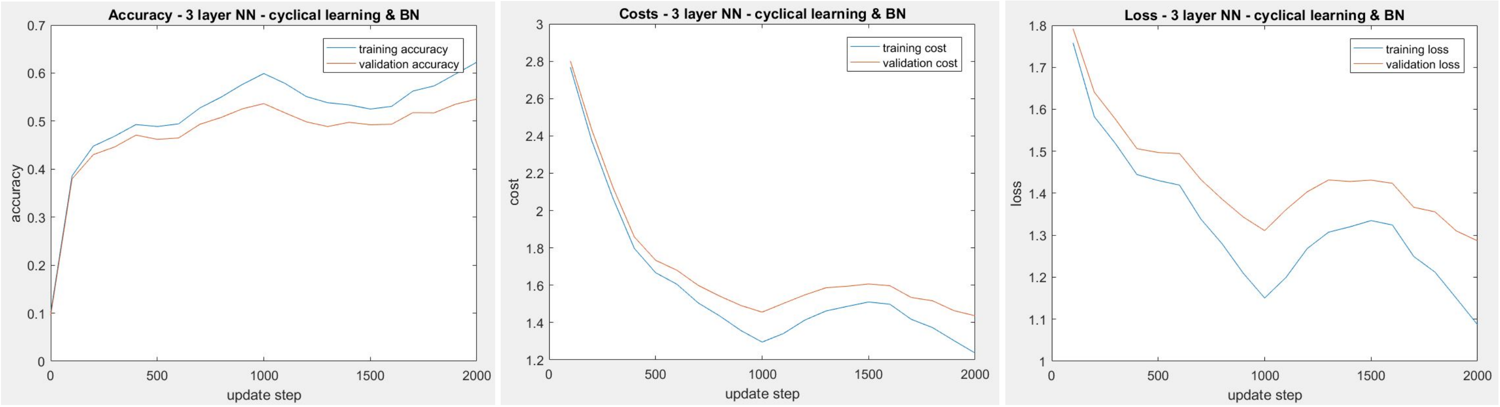

Figure 2: Accuracy, Cost and Loss plots for the training and validation data sets for the 3-layer Neural Network with Cyclical Learning Rate and Batch Normalization.

2.3 The effects of batch normalization for the 3-layer network

By comparing the graphs for the 3 layer network with and without batch normalization in Figure 1 and 2 a slight improvement has been made. The accuracy of the network with batch normalization on the held out test set in section 2.2 is slightly larger than the accuracy of the network without batch normalization in section 2.1. However, it seems as though the amount of overfitting has increased, as the gap between the training and validation plots have increased when batch normalization is used.

3 9-layer networks

Below are the results from a 9 layer network were a cyclical learning rate was used. After each epoch the data points were shuffled. In the second section batch normalization was used and in the first it was not. The weights and biases of the network with batch normalization was initialized using HE-initialization and the other network used Xavier-initialization. The following parameter setting was used during 2 cycles of training in both of these cases:

$$epochs = 20, \lambda = 0.005, n_s = 2250, m_1 = 50, m_2 = 50$$

Were m_1 and m_2 are the number of hidden nodes in the first and second layer.

3.1 A 9-layer network without batch normalization

Accuracy on held out test set: 45.54%

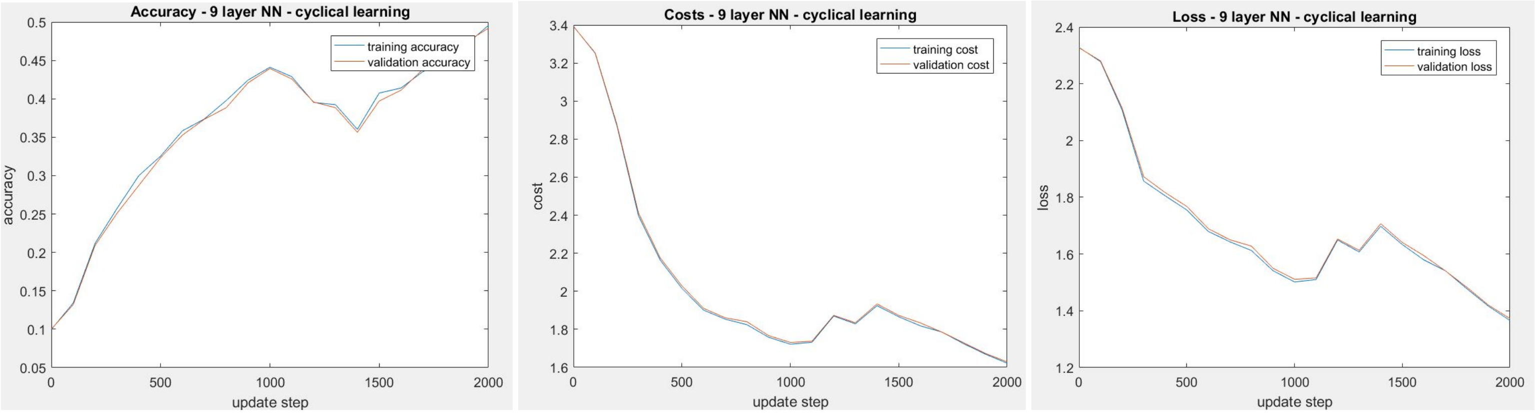

Figure 3: Accuracy, Cost and Loss plots for the training and validation data sets for the 9-layer Neural Network with Cyclical Learning Rate.

3.2 A 9-layer network with batch normalization

Accuracy on held out test set: 51.58%

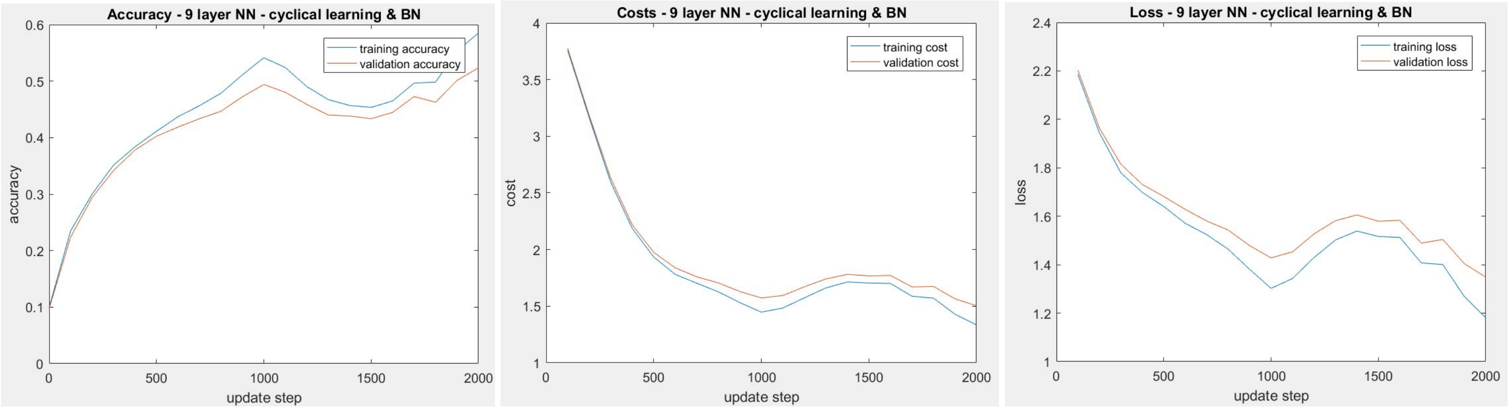

Figure 4: Accuracy, Cost and Loss plots for the training and validation data sets for the 9-layer Neural Network with Cyclical Learning Rate and Batch Normalization.

3.3 The effects of batch normalization for the 9-layer Network

By comparing the graphs below it seems as though batch normalization has increased the amount of overfitting a bit. As the gap between the plots in Figure 3 and 4 has increased. However the networks performance on the held out test set between section 3.1 and 3.2 has notably increased.

4 Coarse to Fine Grid Search to set λ for a 3-layer Network

In this section the network described in section 2.2 was optimized by tuning the amount of regularization used during training. First a coarse search for λ was performed. Then, based on the values found during the coarse search, a new search was performed in a finer range.

4.1 Coarse Search

The λ values where sampled during the coarse search using the following setting:

$$l = l_{min} + (l_{max} - l_{min}) \cdot rand(1,1)$$
$$\lambda = 10^{l}, l_{min} = -4, l_{max} = -1$$

The results for the coarse grid search can be found in table 1.

4.2 Fine Search

For the fine grid search the four λ:s corresponding to the four highest test set accuracies during the coarse search were used to pick a smaller range for the search. These λ values were calculated using the *logged lambda* values from run 6, 14, 20 and 9. Which provided the l_{min} and l_{max} values below.

The λ values where sampled during the fine search using the following setting:

$$l = l_{min} + (l_{max} - l_{min}) \cdot rand(1,1)$$
$$\lambda = 10^{l}, l_{min} = -2.884626952, l_{max} = -2.23064548$$

Table 1: Coarse λ grid search.

Run	λ	Test set Accuracy
1	0.0202	52.37 %
2	0.031345433063940	51.85 %
3	0.001624480510437	52.71 %
4	1.930745684623451e-05	51.56 %
5	0.031345433063940	52.03 %
6	0.005879691232066	53.21 %
7	0.030142780157156	51.57 %
8	4.487577644563437e-04	51.95 %
9	0.004550256502668	53.48 %
10	1.637924219043052e-04	52.32 %
11	3.400434539387020e-04	51.67 %
12	4.561086048204558e-04	52.22 %
13	0.001640935361605	52.47 %
14	0.005638891208495	53.47 %
15	0.078159022815250	49.75 %
16	1.177178082302112e-04	51.6 %
17	0.015080893454091	52.52 %
18	0.038819914557383	51.40 %
19	4.872715469491681e-04	51.76 %
20	0.001304286648069	52.90 %

Table 2: Fine λ grid search.

Run	λ	Test set Accuracy
1	0.004527095929130	53.37 %
2	0.001807976749149	52.85 %
3	0.004060833726532	53.82 %
4	0.001473056799293	52.74 %
5	0.001538055925974	53.19 %
6	0.002038931380506	52.78 %
7	0.003614662478461	53.75 %
8	0.002775069747319	53.39 %
9	0.005028863933547	53.55 %
10	0.001482615406090	52.33 %
11	0.003051432464234	53.24 %
12	0.002133645766569	53.22 %
13	0.001974270371861	53.09 %
14	0.002212808653263	52.75 %
15	0.003530100649325	53.37 %
16	0.005043458634648	53.76 %
17	0.002737025211624	52.80 %
18	0.003063746128017	53.31 %
19	0.001393864600768	52.84 %
20	0.003291402416366	53.37 %

Thus, the best performing network in the fine search had the λ value: 0.0040608337265324 and achieved a test set accuracy of 53.82%.

5 Sensitivity to Initialization

In this section the sensitivity of parameter initialization for the weights and biases are explored for a 3-layer Network.

5.1 A 3-layer Network with Batch Normalization

Table 3: Results for the Network with Batch Normalization for different standard deviations used to initialize the parameters of the network.

σ	Test set Accuracy
1e-1	53.26 %
1e-3	53.41 %
1e-4	53.26 %

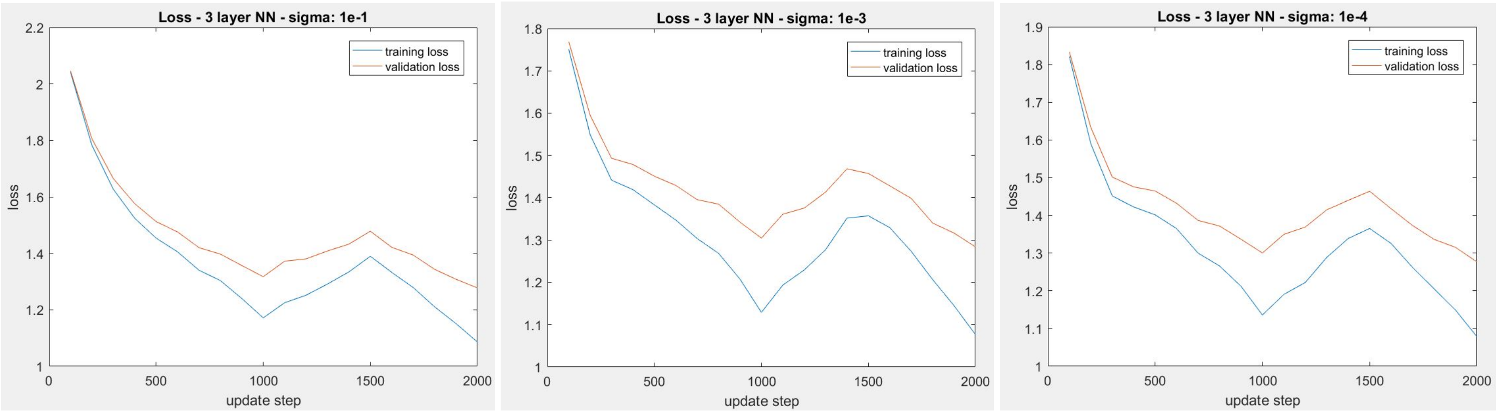

Figure 5: Loss plots for the 3-layer Neural Network with Batch Normalization, were 3 different distributions were used to initialize the weight and bias parameters.

5.2 A 3-layer Network without Batch Normalization

Table 4: Results for the Network without Batch Normalization for different standard deviations used to initialize the parameters of the network.

σ	Test set Accuracy
1e-1	53.11 %
1e-3	11.90 %
1e-4	10.00 %

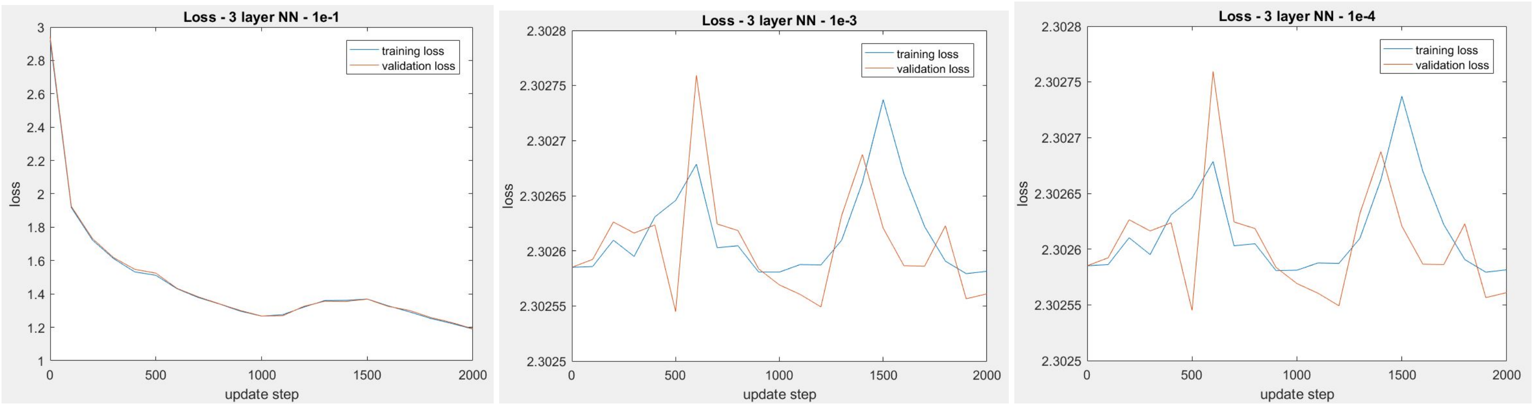

Figure 6: Loss plots for the 3-layer Neural Network with Batch Normalization, were 3 different distributions were used to initialize the weight and bias parameters.

5.3 Discussion

Comparing the losses achieved with batch normalization in Figure 5 and accuracies in table 3 it is apparent that very similar, results can be achieved despite using very different initialization. As both the accuracy remains high and the losses stable throughout training in all of the three cases.

Also, this is apparent if the results are compared to the losses in Figure 6 and accuracies in table 4 were batch normalization was not used. Here, lessening the standard deviation in the parameter initialization destabilises the loss plots and decreases the accuracy from around 50% to approximately 10%.

All in all, the performance of the network with batch normalization is stable and equivalent despite the initialization used, while the network without batch normalization greatly depends on the initialization.

References

[1] *Gradient Checks - Use relative error for the comparison*, (Accessed May 20, 2019).