Assignment 2 - Deep Learning

Alexandra Hotti

March 2019

1 Analytic gradient compared to numerical gradient

The analytical gradient computations were compared against the numerical gradient which was computed via the function *ComputeGradsNumSlow*. The gradients were compared by computing the relative error between each element of the analytical and numerical bias and weight vectors. The relative error was computed via Equation 1.

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

Provided are boxplots of the relative errors for each element in b_1, b_2, W_1 and W_2 for two different parameter settings. Since all the relative errors where below the value 1e-6 for both of the settings the values where deemed small enough according to [1]. Thus, it was concluded that the analytical gradient was implemented correctly.

1.1 Parameter setting 1

The parameter setting for these four plots where:

$$n = 20, d = 20, h = 1e-5, \lambda = 0, \epsilon = 0.1, m = 50$$

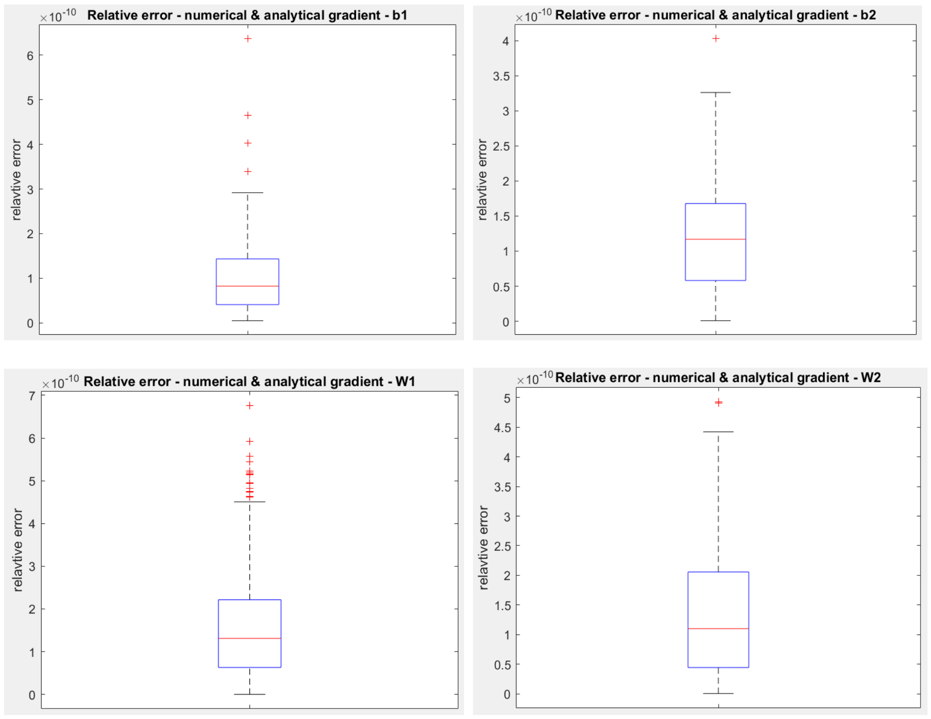


Figure 1: Setting 1 - Relative Errors for gradients of the weights and biases in the first and second layer.

1.2 Parameter setting 2

The parameter setting for these four plots were:

$$n = 20, d = 3072, h = 1e - 5, \lambda = 0.1, \epsilon = 0.1, m = 50.$$

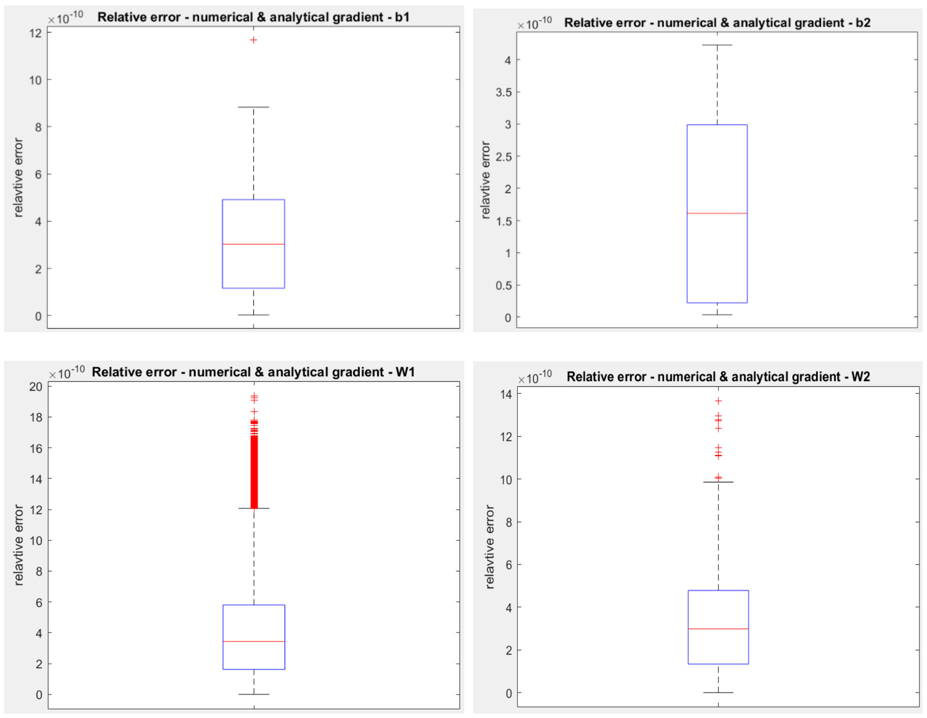


Figure 2: Setting 2 - Relative Errors for gradients of the weights and biases in the first and second layer.

2 Cyclical Learning Rate

The following section contains results were a cyclical learning rate was used during training. The figures below contains computed losses, costs and accuracies using the parameter settings from Figure 3 and Figure 4 in the Assignment description. Note that at update step 0 the values were computed using the randomized initial parameters.

2.1 Setting 1

Final test accuracy: 45.88 %
The following parameters were used during training:

$$n_s = 500, no_{cycles} = 1, batch_{size} = 100, eta_{min} = 1e-5, eta_{max} = 1e-1, \lambda = .01$$

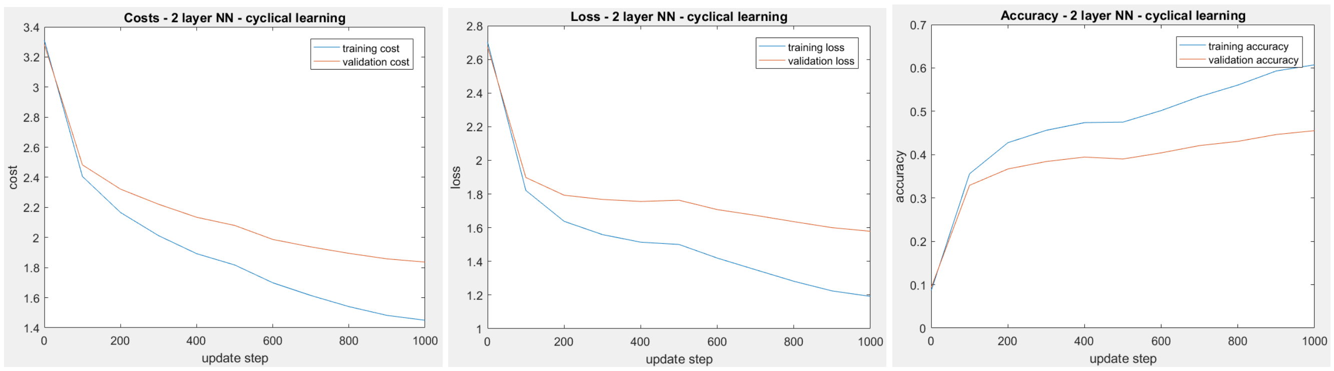


Figure 3: Results from section 2.1

2.2 Setting 2

Final test accuracy: 46.77 %
The following parameters were used during training:

$$n_s = 800, no_{cycles} = 3, batch_{size} = 100, eta_{min} = 1e-5, eta_{max} = 1e-1, \lambda = 0.01$$

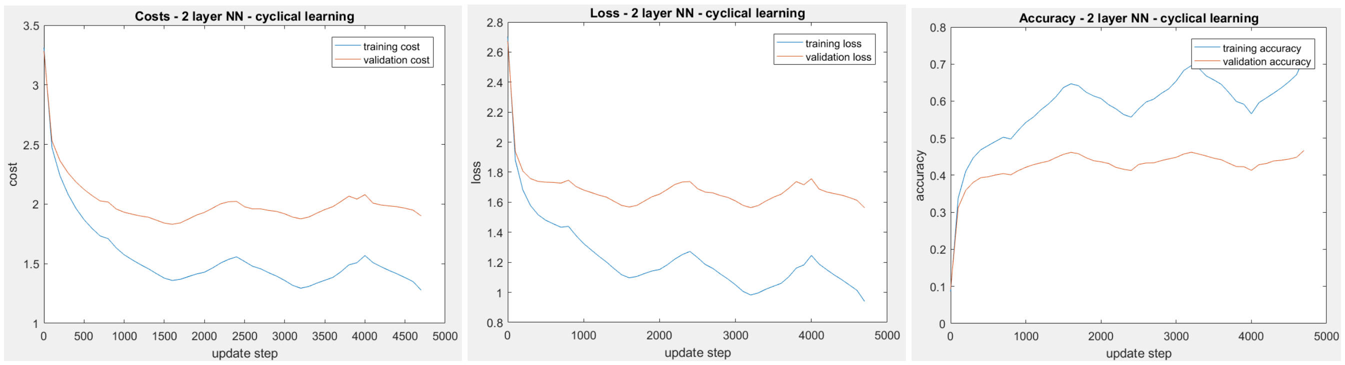


Figure 4: Results from section 2.2

2.3 Discussion

In the second parameter setting the graphs move up and down as n_t varies, while the accuracies in the first graphs stably increases and its losses stably decreases. Perhaps this is caused by using a to larger n_s value in the second setting. However, it is hard to compare the two settings as we are training for only 1 cycle in the first figures and 3 in the second figures. In fact, after a third of the update steps in the second figures the graphs have not yet exhibited the oscillating pattern and in fact look similar to the graphs in the first setting.

Moreover, we only gain a slight increase in the final test accuracy in the second setting when we use 2 more cycles, more epochs and a larger n_s.

In both figures the training accuracies are higher than the validation accuracies. Also, the training costs and losses are lower compared to the results achieved on the validation set. Which is to be expected. Also, all the validation and corresponding training plots seem to move in the same direction. However, the gaps between the graphs are somewhat larger in the second setting, which could indicate that we are overfitting to the data.

3 Coarse-to-fine random search to set λ

In this section a network was optimized by 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.

3.1 Coarse Search

During the coarse search the network was initially trained using 20 different values for λ. The λ values were sampled using the following setting:

$$l = l_{min} + (l_{max} - l_{min}) \cdot rand(1,1)$$
$$\lambda = 10^l, l_{min} = -5, l_{max} = -1$$

The following hyperparameters were used in the network:

$$m = 50, training - dataset - size = 45000, batch_{size} = 100, no_{cycles} = 2, n_s = 900, epoch = 8$$

Were n_s was set using: $n_s = 2 \cdot floor(\frac{45000}{100}) = 900$.

The validation accuracies achieved during the final epoch for the 20 different λ values ranged between: $43.76\% - 52.58\%$ The lambdas resulting in the 5 final testset accuracies were recorded and their corresponding l values were calculated as follows:

$$l = \frac{log(\lambda)}{log(10)}$$

Table 1: Coarse λ grid search.

Run	λ	Test set Accuracy	l
1	0.002179	52.58 %	-2.661644379
2	0.00202	52.40 %	-2.694655098
3	4.89e-05	52.24 %	-4.31086003
4	0.004074	52.06 %	-2.389996346
5	0.00029	52.04 %	-3.537818377

Thus, the λ from run 3 was used as the new l_{min} and run 4 as the new l_{max} for the fine grid search.

3.2 Fine Search

Now the fine search was performed using the distribution described in section 3.1 to generate λ. Were l_{min} and l_{max} were set to:

$$l_{min} = -2.389996346, l_{max} = -4.31086003$$

Again the network was trained using 20 sampled λ values and the hyperparameter presented in section 3.1.
The validation accuracies achieved during the final epoch for the 20 different lambdas ranged between: 51.4% – 52.54%. Thus, the highest accuracy was slightly lower than the highest achieved during the coarse search. However, the variance was notably lower and the mean higher.

Table 2: Fine λ grid search.

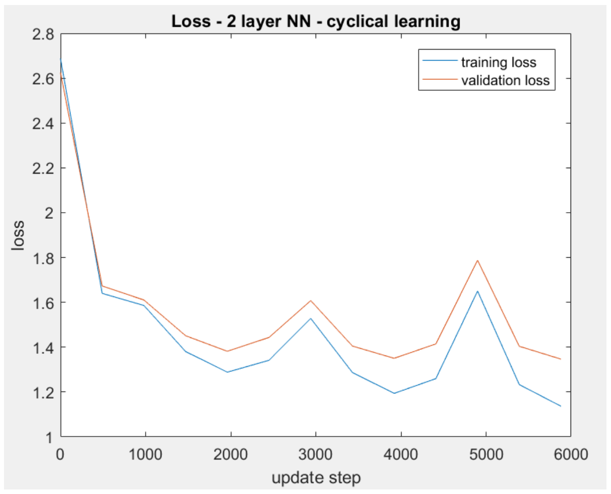
λ	Test set Accuracy
0.000649	52.54 %
0.001479	52.42 %
0.000701	52.28 %

3.3 Final hyperparameters

The best performing λ can be found in the table above: 0.000649. Finally, a new network was trained using the new hyperparameters below.

$$m = 50, \lambda = 0.000648717 dataset = 49000, batch_{size} = 100, no_{cycles} = 3, n_s = 980, epoch = 12$$

Which gave the final test set accuracy of: 51.33%. Were n_s was set using: $n_s = 2 \cdot floor(\frac{49000}{100}) = 980$. Which yielded the following training losses and accuracies:



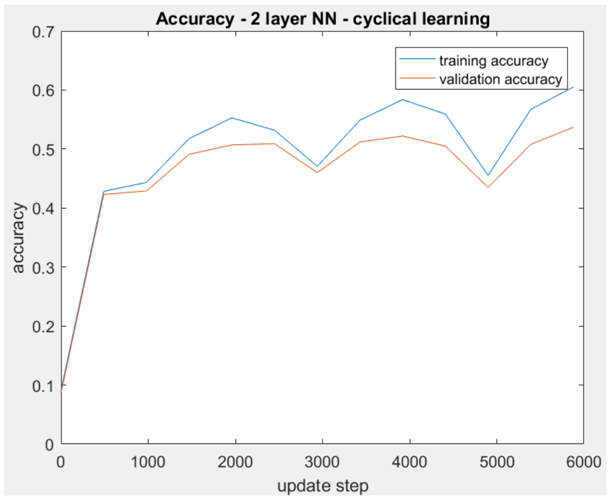


Figure 5: Results from section 3.3

References

[1] *Gradient Checks - Use relative error for the comparison*, (Accessed March 12, 2019).