DD2424 Assignment 2 basic

Jacob Malmberg, 198508060558

April 2019

1 Analytical gradient check

The analytical gradients were checked against numerical gradients computed using ComputeGradsNumSlow.m. This was done for a batch containing the first 100 datapoints of data_batch_1.mat. All 3072 dimensions were included. The relative error was computed following the guidelines laid out on page 7 of Assignment 1:

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

Eps was set to 1e-9 and h was set to 1e-6. The results are summarized in table 1. According to `http://cs231n.github.io/neural-networks-3/#gradcheck` these are good values which make me conclude that my gradient computations are bug free.

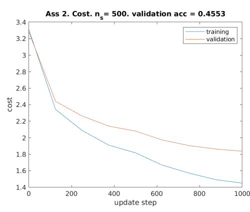
	$\lambda = 0$
Max b1 relative error	5.17e-07
Max b2 relative error	9.13e-08
Max W1 relative error	9.12e-08
Max W2 relative error	2.82e-07

Table 1: Table showing summary of relative errors

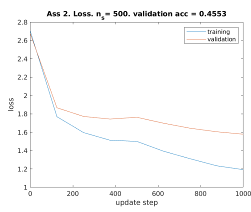
2 Cyclical learning rate, figures

Cyclical learning rate was implemented as described in the assignment. eta_min was set to 1e-5, eta_max was set to 1e-1, and lambda was set to 0.01. The results for one cycle of training using n_s=500 is displayed in figure 1 with the accuracy on the validation set being 45.53%. The results for three cycles of training using n_s=800 is displayed in figure 2 with the accuracy on the validation set being 46.7%.

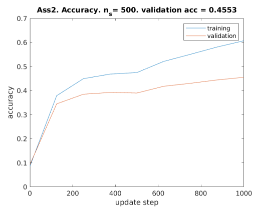
In figure 2 it can clearly be seen that the loss and accuracy goes up and down as η varies. In figure 1, loss and cost goes down almost linearly. It is also worth noting that the loss/accuracy oscillates far more in figure 2 than in figure



(a) Cost plot

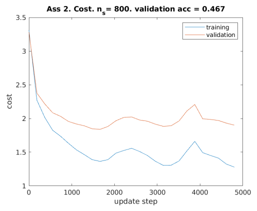


(b) Loss plot

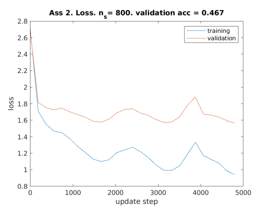


(c) Accuracy plot

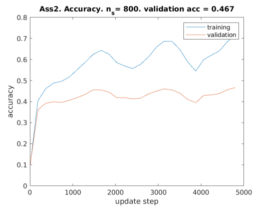
Figure 1: Training curves for n_s=500. Corresponds to figure 3 in the assignment.



(a) Cost plot



(b) Loss plot



(c) Accuracy plot

Figure 2: Training curves for n_s=800. Corresponds to figure 4 in the assignment.

1. This may be because n_s is larger in figure 2. Further, the accuracy is only slightly higher despite training for more cycles and thus more epochs in figure 2. We can also see that the training loss & cost is lower than the validation loss & cost for both networks. There is a noticeable gap, which indicates some overfitting. In the accuracy plot of fig 2, the accuracy seem to reach three peaks which coincides with the end of the three learning rate cycles (update step ∼1600, ∼3200, ∼4800).

Lastly, since the networks in fig 1 & 2 were trained for a different number of cycles and update steps, comparisons should be taken with a grain of salt.

3 Coarse search for lambda

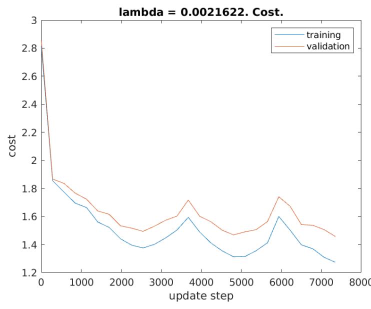
15 lambda values were generated from [1e-5, 1e-1]. The range for the generated lambda values was [1.1174e-5, 4.9723e-2]. The batch size was set to 100, epochs was set to 8 and the training dataset had N = 45000 and the validation dataset had N = 5000. Single precision was used instead of double precision was used to facilitate this since my computer does not have enough RAM for doubles. Each network was trained during 2 cycles with $n_s = 2floor(n/n\_batch) = 900$. The three best-performing networks had validation accuracies of 52.7%, 52.45% and 52.02%. The corresponding lambdas were 4.731e-3, 4.158e-4 and 4.87e-05.

4 Fine search for lambda

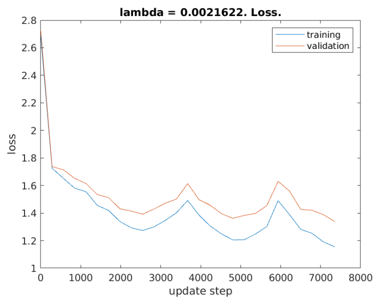
15 lambda values were generated from [4.87e-05, 4.731e-3], the range which covered the three best networks found in the coarse search. The range for the generated lambda values was [1.052e-4, 4.3754e-3]. The networks were trained in the same fashion as they were for the coarse search. The three best-performing networks had validation accuracies of 52.9%, 52.7% and 52.56%. The corresponding lambdas were 2.1622e-3, 2.7406e-3 and 1.4084e-03. The best performing lambda value was thus 2.1622e-3 with an accuracy of 52.9%.

5 Performance of the best lambda

A network was trained using lambda = 2.1622e-3 for three cycles. n_s was set to 1225 and batch size was 100, resulting in 15 epochs of training and and 7350 update steps. The accuracy on the test set was 52.18%. The cost and loss plots are displayed in figure 3.



(a) Cost plot



(b) Loss plot

Figure 3: Loss and cost of the best lambda