Deep Learning DD2424 - Assignment 2

Anton Stagge

April 2019

1 Introduction

In this assignment I have created a 2 layer dense network, and successfully trained it to classify CIFAR-10.

2 Test against numerical gradients

To test my new function that calculated gradients, I had to modify the Matlab code provided to calculate the gradient with respect to both $W1$ and $W2$. This was quite simply done by just adding another for loop over the number of layers. Then to test my analytically calculated gradients, I calculated the relative error between the two. At first I started with just 1 image in the batch, and with a step-size value h of 10^{-5} I got the results:
W1 relative error: 1.5247773302268788e-06
b1 relative error: 3.8128984617123615e-07
W2 relative error: 1.4792241413492886e-06
b2 relative error: 9.502276856263709e-07

I then raised the *n_batch* to 10 and got the results:
W1 relative error: 2.988665856777013e-06
b1 relative error: 9.477457767024689e-07
W2 relative error: 3.3422544155187e-06
b2 relative error: 2.6931750696474207e-06

This was calculated using forward difference and took up to 3 minutes to calculate on my computer. That's why a decided that this was enough evidence that my analytically calculated gradients were correct and bug free.

3 Cyclic eta

The graphs represented in Figure 1 and Figure 2 show my validation and training cost as well as accuracy respectively having the same parameters as Figure 3 in the assignment. That being $eta\_min = 10^{-5}$, $eta\_max = 10^{-1}$ $lambda = .01$, $n\_s = 500$ and training for one full cycle. The final accuracies with this model were:
Test accuracy: 0.454400
Train accuracy: 0.582600
Valid accuracy: 0.446000

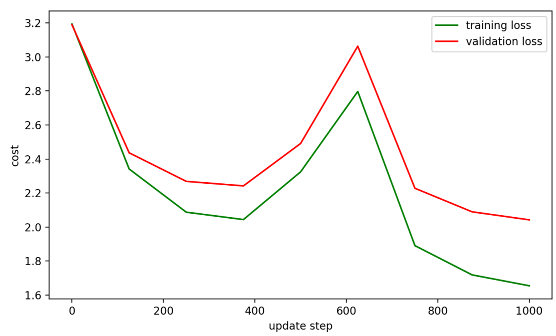

Figure 1: The graph of the training and validation cost computed 9 times per cycle. With cyclic eta with $n\_s = 500$ and $lambda = 0.01$ and 1 cycle of training.

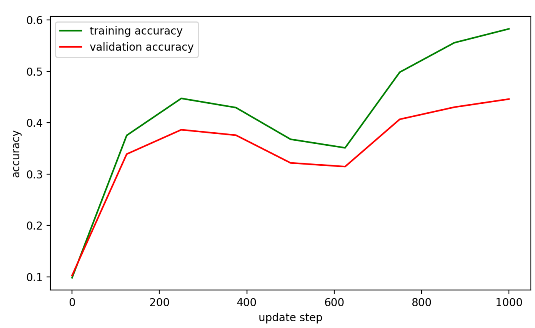

Figure 2: The graph of the training and validation accuracy computed 9 times per cycle. With cyclic eta with $n\_s = 500$ and $lambda = 0.01$ and 1 cycle of training.

The graphs represented in Figure 3 and Figure 4 show my validation and training cost as well as accuracy respectively having the same parameters as Figure 4 in the assignment. That being $eta\_min = 10^{-5}$, $eta\_max = 10^{-1}$ $lambda = .01$, $n\_s = 800$ and training for **three** full cycle. The final accuracies with this model were:
Test accuracy: 0.446400
Train accuracy: 0.689100
Valid accuracy: 0.446100

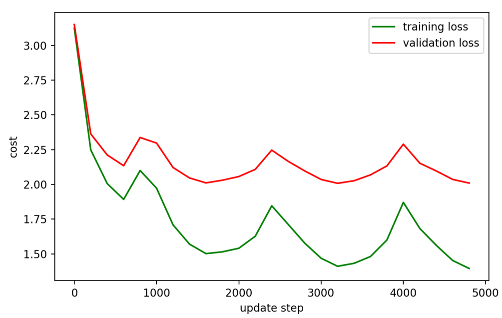

Figure 3: The graph of the training and validation cost computed 9 times per cycle. With cyclic eta with $n\_s = 800$ and $lambda = 0.01$ and 3 cycle of training.

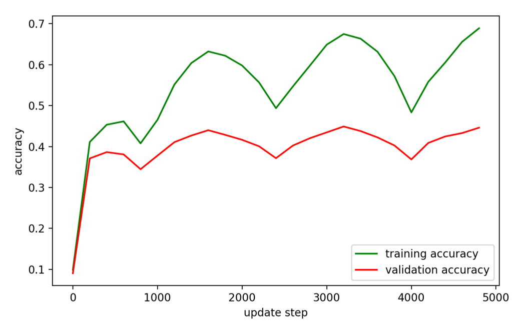

Figure 4: The graph of the training and validation accuracy computed 9 times per cycle. With cyclic eta with $n\_s = 800$ and $lambda = 0.01$ and 3 cycle of training.

3.1 A bug in the Assignment instruction code?

I was wondering why my graphs differed so much from the graphs in the assignment instruction sheet. I managed to recreate the graphs almost exactly by introducing a bug in the code. That bug was to let eta be 0.01 for the first cycle of training and then let it cycles between *eta_min* and *eta_max*. Figure 5 shows the correct eta used in my graphs above, while Figure 6 shows the eta with the bug introduced. Figure 9 and Figure 8 shows the graphs of validation and accuracy with the bug introduced. As one can see, they are almost identical to the ones in the assignment instructions.

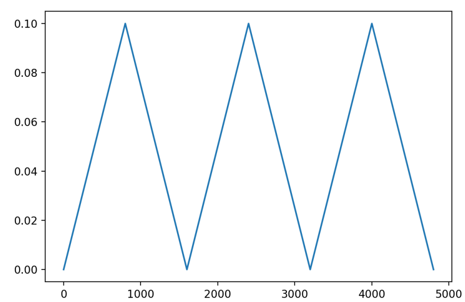

Figure 5: The graph of the values of eta, computed 9 times per cycle. With cyclic eta with $n\_s = 800$ and $lambda = 0.01$ and 3 cycle of training.

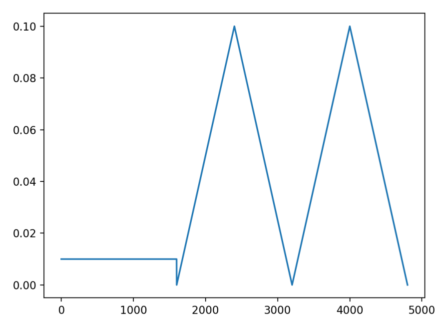

Figure 6: The graph of the values of eta, computed 9 times per cycle. With cyclic eta with $n\_s = 800$ and $lambda = 0.01$ and 3 cycle of training. This time with the bug mentioned above introduced.

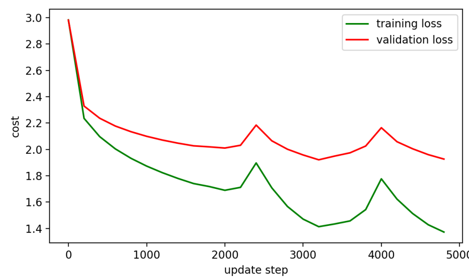

Figure 7: The graph of the training and validation cost computed 9 times per cycle. With cyclic eta with $n\_s = 800$ and $lambda = 0.01$ and 3 cycle of training. This time with the bug mentioned above introduced.

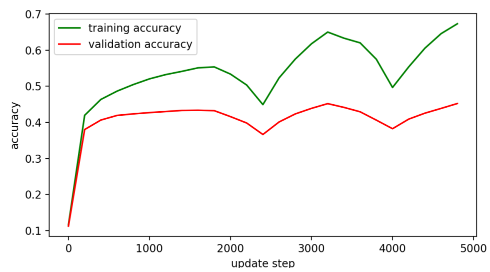

Figure 8: The graph of the training and validation accuracy computed 9 times per cycle. With cyclic eta with $n\_s = 800$ and $lambda = 0.01$ and 3 cycle of training. This time with the bug mentioned above introduced.

4 Grid search for lambda

4.1 Coarse

First I did a search with 15 different values of lambdas uniformly random between 0 and 10^{-5}. I Trained the models with all training data but 5000 that was used as validation data. I trained for 2 full cycles and recorded the best validation accuracy, measures every time eta was eta_min. n_s was set to $2 * floor(n\_tot/n\_batch)$ which was 900, since n_batch was set to 100. From this search I found the 3 best values of lambda were:
Lambda: 4.2×10^{-5}, with best accuracy: 51.72 percent.
Lambda: 1.2×10^{-5}, with best accuracy: 52.04 percent.
Lambda: 2.622×10^{-3}, with best accuracy: 52.16 percent.

4.2 Fine

Secondly I did a search with 15 other different values of lambdas uniformly random between 10^{-3} and 10^{-5}. With the rest of the hyper-parameters the same as above. From this search I found the 3 best values of lambda were:
Lambda: 10^{-5}, with best accuracy: 52.22 percent.
Lambda: 3×10^{-5}, with best accuracy: 52.60 percent.
Lambda: 3.08×10^{-4}, with best accuracy: 52.32 percent.

4.3 Finest

Thirdly I did a search with 6 other different values of lambdas uniformly random between 10^{-4} and 10^{-6}. With the rest of the hyper-parameters the same as above, except this time training for 3 cycles. From this search I found the 3 best values of lambda were:
Lambda: 4.1×10^{-5}, with best accuracy: 52.88 percent.
Lambda: 3×10^{-6}, with best accuracy: 52.86 percent.
Lambda: 2.6×10^{-4}, with best accuracy: 52.26 percent.

4.4 Final model

For the final model trained on all the training data, except for 1000 used for validation, I used a lambda of 4.1×10^{-5} as this was the lambda that achieved the highest accuracy in the grid search and I trained for 5 cycles. The cost/loss function for the training and validation data is shown in Figure **??** and the accuracy in Figure **??**. The test accuracy achieved by the final model was 51.15 percent, and the highest validation accuracy was 53 percent.

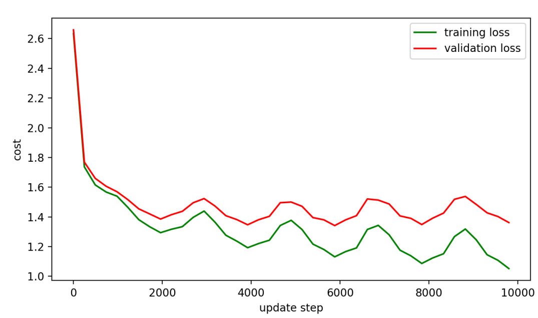

Figure 9: The graph of the training and validation cost computed 9 times per cycle for the final model.

Figure 10: The graph of the training and validation accuracy computed 9 times per cycle for the final model.