DD2424 Deep Learning in Data Science

Assignment 2 - 25th June, 2019

Musab Tayyib Gelisgen - gelisgen@kth.se

In this assignment, a 2-layer neural network is trained with CIFAR-10 dataset using mini-batch gradient descent by updating previously written code in assignment-1 and also adding new functions. There are 50000 training images and 10000 test images used in this assignment. The tests done are provided below.

1. Gradients

In the assignment I have used two functions to calculate the gradients, one with analytically and one with numerically. Then I have calculated the relative error by the analytical and numerical derivatives separately. When tried with many parameter settings, the resulting errors are very low with a value smaller than e^{-6}. Also, the gradient values obtained with both methods are very close to each other. Then I concluded that I managed to write the code without bugs and correctly.

2. Learning Rate (Cyclic)

The learning rate in the assignment increases and decreases according to the $\eta_{\min}$ (1e-5) and $\eta_{\max}$ (1e-1) during the batch updates in an epoch. The increase in learning rate from $\eta_{\min}$ to $\eta_{\max}$ and decrease from $\eta_{\min}$ to $\eta_{\max}$ is a single cycle of learning rate.

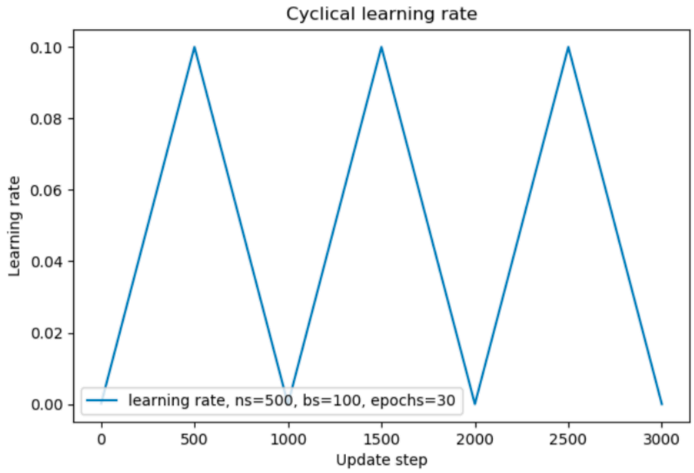

Figure 1: Cyclic learning rate

3. Loss and Accuracy

According to the previously given cyclical learning rate cycles, the accuracy and the loss are shown below.

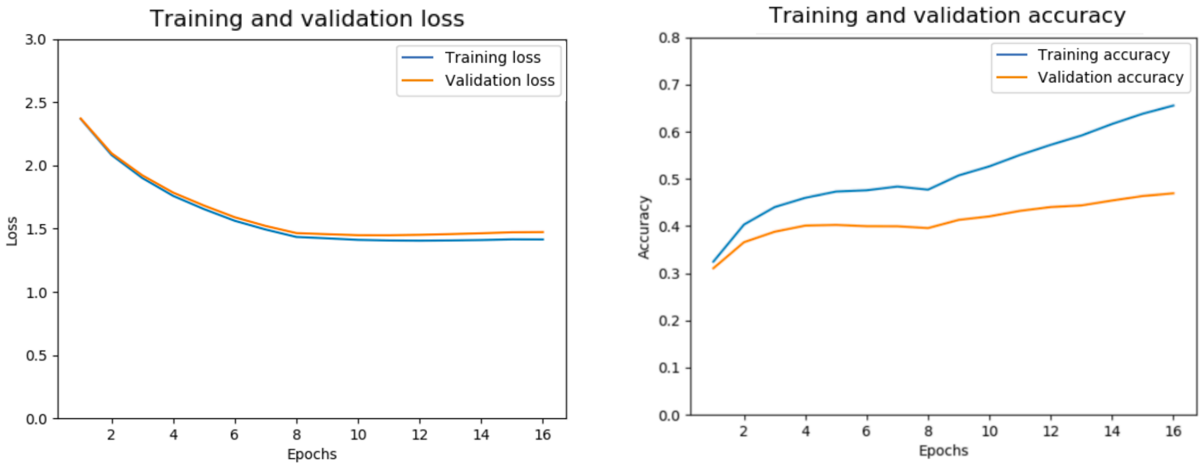

Figure 2: Loss and accuracy of 3 cycles when n_s:500

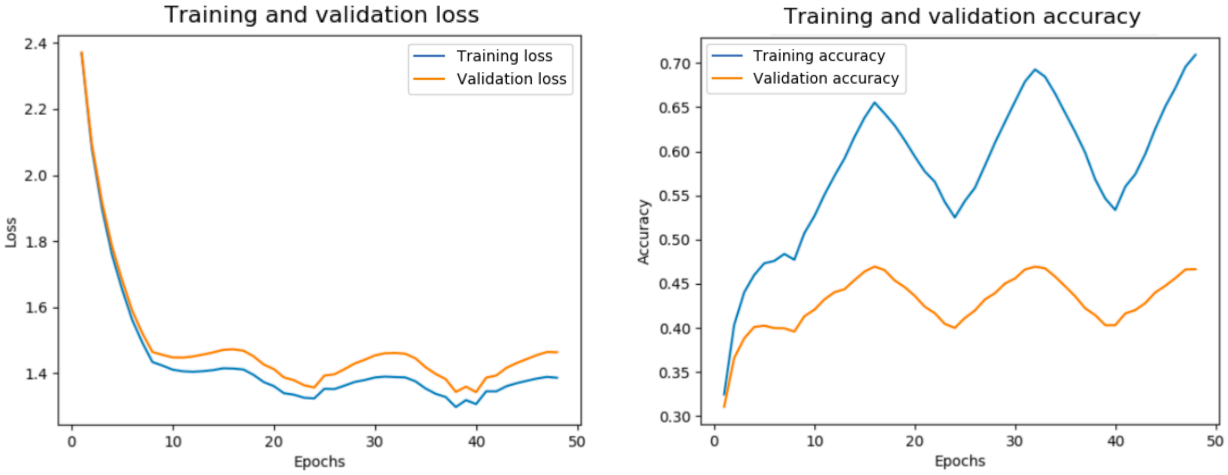

Figure 3: Loss and accuracy of 3 cycles when n_s:800

One explanation to the figures above might be the learning happens the most at the beginning of a cycle. This can be seen from the decrease in the loss and the increase in the accuracy at the beginning of the cycles. The learning can be badly affected by the learning rate sometimes such as when the learning rate starts to decrease.

4. Setting lambda

Coarse to fine random search method is used in this assignment when trying to optimize the lambda. First, I began with a rather large range of random lambda values and then I tested the accuracy in the validation set with these random lambda values. The first lambda values used are 1e0, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9 and I used 2 cycles in each network. The best accuracy is obtained when lambda is set to 1e-4 followed by 1e-3 and 1e-5. This helped me look at a narrower range much closer so I continued with looking around the 1e-4 range closer. I have chosen my new interval as 1e-3 and 1e-5 and divided that interval into 20 pieces. Then I have continued my experiments by using these new lambda values with 3 cycles in each network.

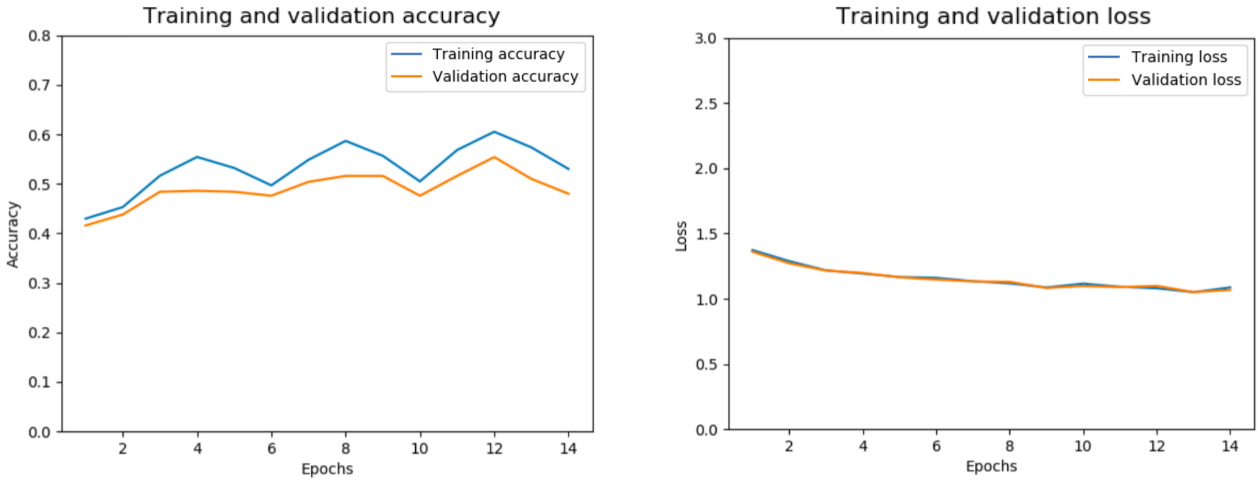

Figure 4: Accuracy and loss when lamba is set to 6e-4.

The best accuracy I have obtained is **0.5419** when lambda is set to 6e-4 followed by 7e-4 and 2e-4 The plots regarding the best experiment is provided below when lambda is set to 6e-4 and the network is trained on all the training data except for 1000 examples in a validation set for 3 cycles.