Assignment 2 - DD2424 - One Layer Network

Tristan Perrot

April 2024

1 Exercise 1

For this assignment. I decided to create a python object that would represent our classifier. I made this choice because it would simplify the creation of each different classifier and would allow me to easily test different configurations.

How I checked my analytic gradient computations?

After doing the math to compute the gradient and checked it with the given formula on the slides. I decided to implement the gradient computation in the python object. I then compared the results of the analytic gradient with the numerical gradient and the **PyTorch** gradient. I use the **relative error** to compare the gradients for every weights and biases. I get a relative error of 10^{-14} for torch vs analytical. However, I get a relative error of 10^{-4} for analytical vs numerical and torch vs numerical. The last relative error can be explained because of the numerical approximation of the gradient.

2 Exercise 2

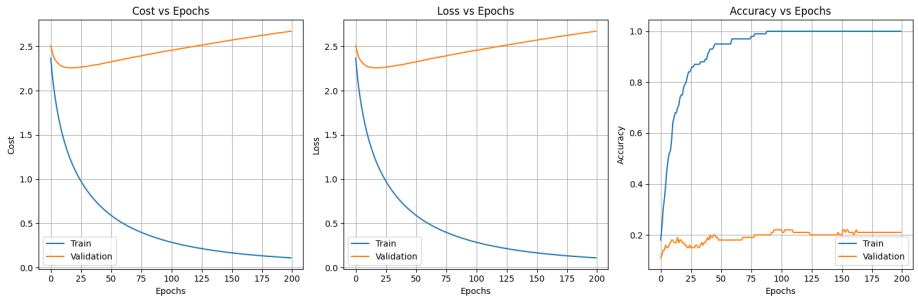

Figure 1: Small training with no regularization. It definitely overfits the data.

Here, just to see if the classifier works, I trained it on a small dataset with no regularization. As we can see in figure 1, the classifier overfits the data. The training loss goes to zero, but the validation loss increases. This is a clear sign of overfitting. I get these accuracies:

- Training accuracy: 100%
- Validation accuracy: 21%
- Test accuracy: 14%

3 Exercise 3

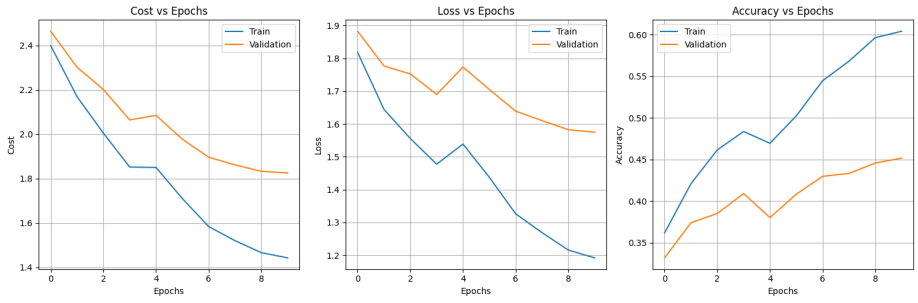

Figure 2: **Training curves (cost, loss, accuracy) for one cycle of training.** The hyper-parameter settings of the training algorithm are `eta min = 1e-5`, `eta max = 1e-1`, `lambda = .01`, `n s = 500` and `batch size = 100`.

Now that the classifier works. I trained the classifier on the CIFAR-10 dataset with cyclical learning rates. Now, the learning rate is not constant but varies between a minimum and a maximum value.

4 Exercise 4

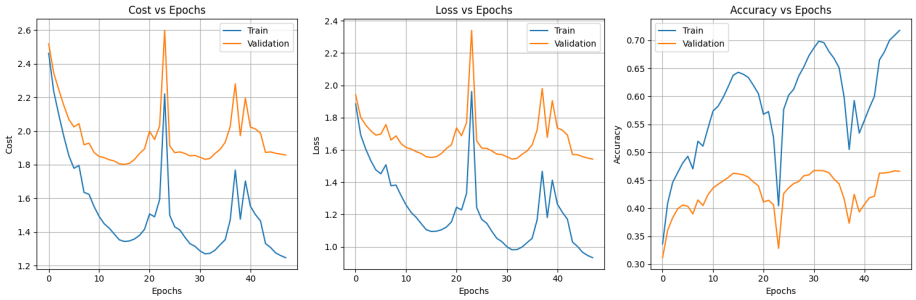

Figure 3: **Training curves (cost, loss, accuracy) for three cycles of training.** The hyper-parameter settings of the training algorithm are `eta min = 1e-5`, `eta max = 1e-1`, `lambda = .01`, `n s = 800` and `batch size = 100`.

Now that the cyclical learning rate works, I trained the classifier on the CIFAR-10 dataset with three cycles of training. As we can see in figure 3, the training loss decreases and the validation loss decreases as well and we can see the cycles. I obtain these accuracies:

- Training accuracy: 71.78%
- Validation accuracy: 46.61%
- Test accuracy: 48.09%

4.1 Coarse to fine lambda search

4.1.1 Coarse search

For the coarse search, I used the `l min = -5` and `l max = -1` (so with `lambda` between 10^{-lmin} and 10^{-lmax}) and I used 20 values between these two values randomly chosen. I trained the classifier for 2 cycles. I obtain this top 3 results:

- Lambda: 8.887e-05, Accuracy: 47.3%
- Lambda: 0.0007362, Accuracy: 46.8%
- Lambda: 0.0008308, Accuracy: 46.4%

4.1.2 Fine search

For the fine search, I used the worst lambda and the best lambda from the top 5 results of the coarse search. I used 20 values between these two values randomly chosen. I trained the classifier for 2 cycles. I obtain this top 3 results:

- Lambda: 0.0001349, Accuracy: 47.1%

- Lambda: 4.93e-05, Accuracy: 45.7%
- Lambda: 0.0001497, Accuracy: 45.7%

4.2 Final test accuracy

After that I have redone a fine search with the same method as before but with the best lambdas from the fine search. I obtain the best lambda: 0.0003 with an accuracy of 48.5% on the validation set. Then, with this parameter, a `n s = 600`, 5 cycles of training and a batch size of 200.

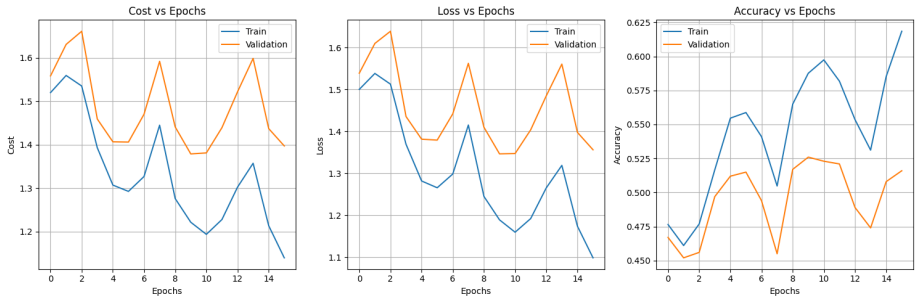

Figure 4: **Training curves (cost, loss, accuracy) for eight cycles of training.** After the coarse and fine search.

I obtain these accuracies:

- Train Accuracy: 63.12%
- Validation Accuracy: 52.2%
- Test Accuracy: 51.49%

5 Conclusion

In conclusion, I have implemented a two layer neural network with a cyclical learning rate and a regularization term. I have trained this classifier on the CIFAR-10 dataset and I have done a coarse and fine search for the regularization term. I have obtained a test accuracy of 51.49% which is not bad for a two layer neural network.