DD2424 - Assignment 2
Bonus Point

Thi Thuy Nga Nguyen

1 Improvements

In this part, I used all data (5 batches) for training except 5000 images for validation. The hyper-parameter was set as in the best model found in the last part of assignment 2. Specifically, $\lambda = 0.00268647$, n_batch = 100, eta_min = 1e-5, eta_max = 1e-1, n_s = 900.

1.1 Ensemble

At the end of each cycle the network parameters (W, b) were recorded except the first one, then averaged. This average was used to classify a unseen sample.
Practically, I trained network for 6 cycles and 50 nodes in hidden layer and got 52.25% in test accuracy. Using the ensemble the test accuracy was slightly improved to 52.47%.

1.2 More nodes in hidden layer

I trained the network for 3 cycles with more hidden nodes and got an significant improvement in performance. In particular, with 100 hidden nodes and more regularization $\lambda = 0.003$ the test accuracy was improved to 53.36% (comparing to 52.25% with 50 nodes).

1.3 Random jitter

In the mini-batch before doing the forward and backward pass a small random jitter which has Gaussian distribution with mean 0 and standard deviation 0.01 was added to each image. The network was trained for 3 cycles with 50 nodes in hidden layer. It was gotten 51.64% in the test accuracy which was not better than training without random jitter.

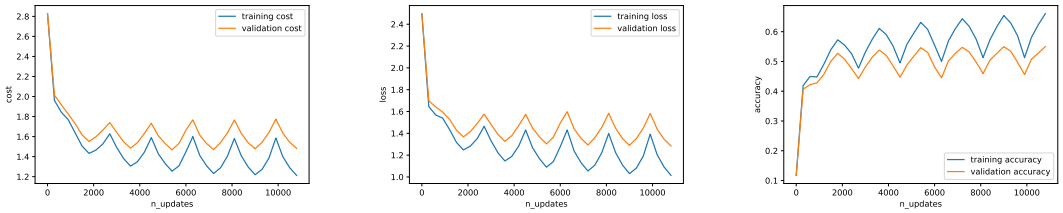
1.4 Random jitter and more nodes in hidden layer

I combined adding random jitter and using more hidden nodes and got better performance. In particular, a random Gaussian noise with mean 0 and standard deviation

0.01 was added in each image in mini-batch and the network with 100 hidden nodes $\lambda = 0.003$ was then trained for 3 cycles. I got 53.43% in the test accuracy which was slightly improved comparing to training without jitter (53.36%).

1.5 Conclusion

We can see that in using more hidden nodes brought the significant gain while the improvement coming from ensemble and random jitter was not stable. The best performance I got was 54.36% using 100 hidden nodes with $\lambda = 0.003$ and training for 6 cycles. Its loss, cost and accuracy plots for training and validation set after every 300 updates are shown in figure 1



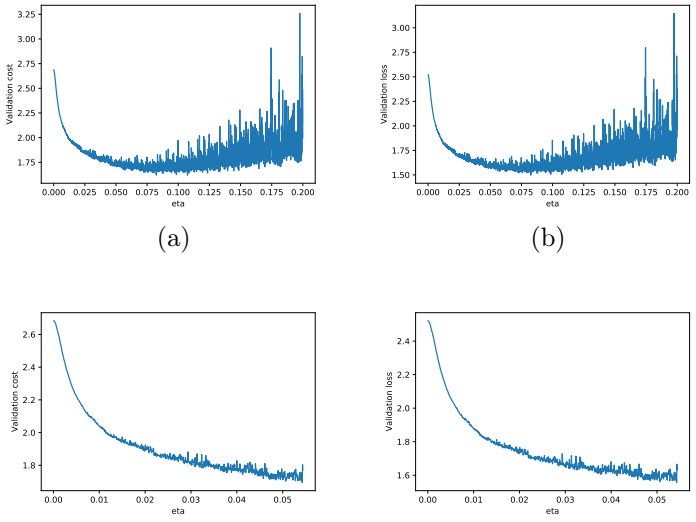
(a) (b) (c)

Figure 1: The cost (a), loss (b) and accuracy (c) after each updating after 6 cycles of training with 100 hidden nodes and $\lambda = 0.003, n\_batch = 100, eta\_min = 1e-5, eta\_max = 1e-1, n\_s = 900$.

2 Find range for cyclical learning rate

To find the the range for cyclical learning rate, I trained the network for a half of cycle in which η is linearly increasing from 0 to 0.2 with 8 epochs, n_s = 3600 and the same hyper-parameter as above. The value of eta_min was determined at the point that the loss curve started decreasing and eta_max was the point that the loss became more fluctuation (I checked an increment of 0.1 in the loss). The range of $(5.55555556 \times 10^{-5}, 5.43888889 \times 10^{-2})$ was found out. The loss and cost plots in this pre-training were shown in figure 2

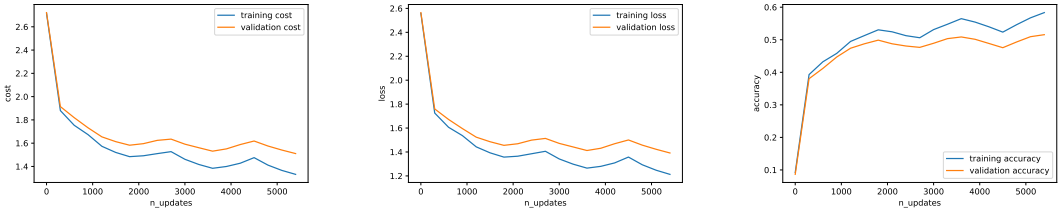
Using this range I trained the network of 50 hidden nodes for 3 cycles and hyper-parameter $\lambda = 0.00268647$, n_batch = 100, n_s = 900. In the result, the test accuracy was 51.5% and the cost, loss, accuracy of training and validation set after every 300 updates could be seen in figure 3



(a) (b)

(c) (d)

Figure 2: The cost (a), loss (b) in pre-training to search the range of learning rate for cyclical method; the cost (c), loss (d) in the found range. The hyper-parameter was $\lambda = 0.00268647, n\_batch = 100, n\_s = 3600, n\_epochs = 8$.



(a) (b) (c)

Figure 3: The cost (a), loss (b) and accuracy (c) after each updating after 3 cycles of training with $\lambda = 0.00268647, n\_batch = 100, eta\_min = 5.55555556e-5, eta\_max = 5.43888889e-2, n\_s = 900, n\_epochs = 12$.