Deep Learning Assignment1, Bonus Point

Carlo Saccardi, saccardi@kth.se

April 2022

The best performances obtained before this bonus point were achieved with the following parameters: $lambda = 0.1, nepochs = 40, nbatch = 100, eta = .001$ The resulting validation accurancy was equal to 0.38.

1 Bonus point 1, Optimize the performance of the network

### 1.1 Use all the available training data for training

The first improvement was obtained by training the network on the entire dataset for each epoch. Moreover, the size of the validation set was decreased to 1000 samples. As a result, the accurancy on the validation set increased a bit, but this improvement is not significant (0.396).

Use all the available training data
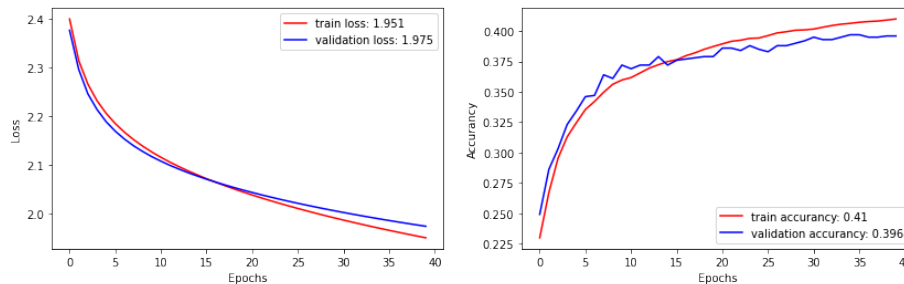
1.2 Augment the data This method didn't lead to any improvements. My guess is that the way I was asked to implement this consisted in randomly substituting some original samples with their flipped version (horizontal flip), instead of adding this augmented samples to the data-set and make it larger. This basically consisted in adding noise to some sample, and indeed, the performances did not improve. Maybe I misunderstood the task, but from the Matlab function that was provided on the Assignment this was my intuition.

Augmented data
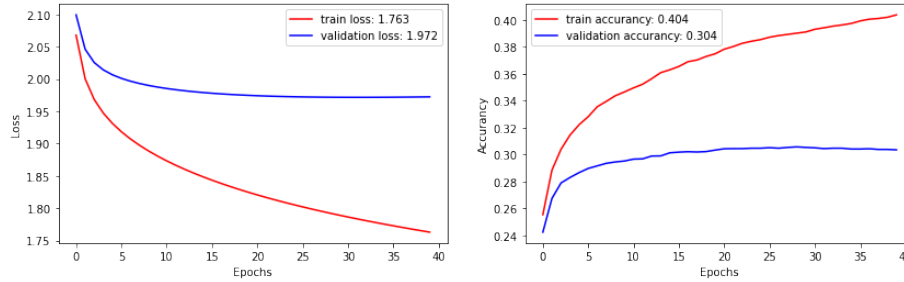
1.3 Grid Search Obviously grid search led to some improvement, since I was able to find the best combination of parameters among the following values: $lambdas = [0, 0.1, 1], etas = [0.00001, 0.0001, 0.001], nbatches = [10, 100, 200]$ The best set of parameters was: $lambda = 0.1, etas = 0.001, nbatches = 200$ which led to a validation accuracy equal to 0.385. Again, not a significant improvement.

1.4 decaying the learning rate Finally, I tried to by decaying the learning rate by a factor of 10 after every 10 epoch. The resulting validation accuracy was equal to 0.384. Notice that the plot curves are more unstable in the early epochs when the learning rate is still high, and the flatten down after the learning rate is decreased.

decaying the learning rate