[DD2424] Assignment 2

Victor Hansjons Vegeborn

June 1, 2018

Gradient Check

The gradient check was based on `ComputeGradsNumSlow.m` provided in the course. While performing the gradient check, some of the elements within the larger weights and bias terms showed very large errors. By reducing the dimensions of the weights and the bias terms I could obtain very small errors. Evidence to support that the calculations of the gradients are correct is the actual performance of the network and the actual convergence.

Momentum

For this task of the assignment I let the hyper-parameters be set to $\eta = 0.01$ and $\lambda = 1e-5$. The momentum term was set to $\rho = 0.9$. I trained the network over 10 epochs and with a batch size of 100. Without momentum I obtained a training loss of 2.19200781 on the 10:th epoch, while adding momentum gave a training loss of 1.55919794. Training over time can be seen in Figure 1.

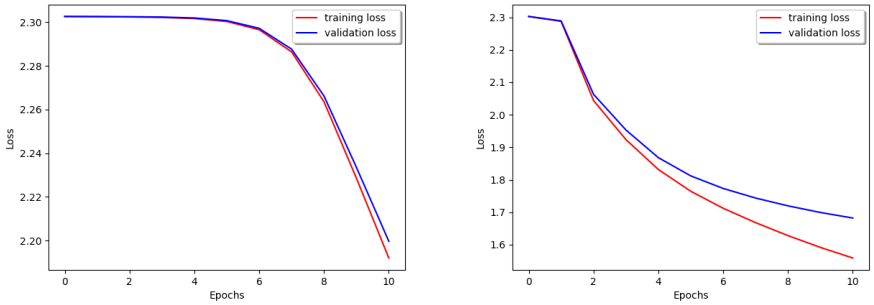

Figure 1: No momentum (left), with momentum (right).

Conclusion

Adding momentum clearly helps the gradient to converge faster. However, the increase in training speed is dependent on the hyper-parameters. During training with $\eta = 0.001$, the effect was not as prominent.

Coarse Search

The generated uniform random values in the coarse search was given from the following ranges: $\eta \in [10^{-4}, 10^{-1}]$, $\lambda \in [10^{-6}, 10^{-1}]$. From these ranges 100 value pairs was set during training over 5 epochs. The top 3 results are shown in Figure 1

Validation loss	η	λ
1.6436	0.032558	1.3197e-06
1.6543	0.028111	4.7226e-05
1.6575	0.045343	4.8037e-06

Table 1: Coarse search

Fine Search

The ranges set for the fine search was the following: $\eta \in [10^{-1.9}, 10^{-1.1}]$, $\lambda \in [10^{-5,9}, 10^{-3.5}]$. For the fine search I generated 80 uniformly randomized pairs and trained the network for 7 epochs. The final results are shown in Figure 2

Validation loss	η	λ
1.6132	0.026326	3.3945e-06
1.6240	0.034267	2.1996e-05
1.6249	0.026988	5.4151e-05

Table 2: Fine search

Training

Training over 10 epochs on the first training data, I was able to obtain an accuracy of 0.4438 with the hyper-parameters set to the best resulting parameters in the fine search (see Figure 2). Momentum was used and $\rho = 0.9$, with a decay rate set to 0.95. The training and validation loss can be seen in Figure 2

The final training was done on all datasets with the same hyper-parameters set as stated above. As instructed, I stored 1000 samples for validation. I trained the network over 30 epochs and was able to obtain a final accuracy of

0.4893. Momentum and decay was used as in the previous test. The training and validation loss can be seen in Figure 2.

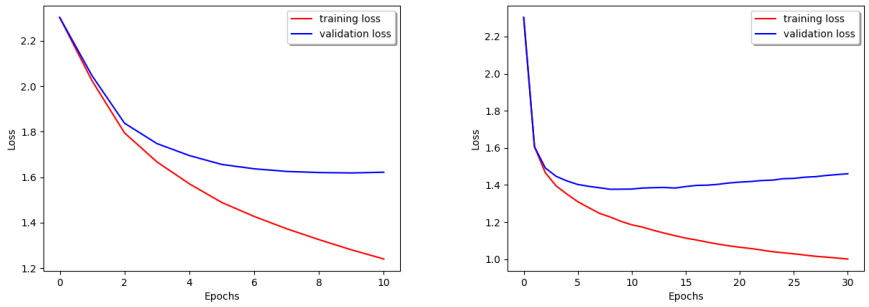

Figure 2: Training on small dataset over 10 epochs (left), Training on large dataset over 30 epochs

Training revised

It is evident that the final training suffered from overfitting (as can be seen from the increasing validation loss in Figure 2). The slope of the training curve also suggests that the learning rate is quite high. Therefore, I decided to manually tweak some of the parameters to counter this behaviour. I decreased the learning rate and the momentum term while increasing the regularization term. Finally, the values opted for was the following: $\eta = 0.018$, $\lambda = 8.0e - 04$, $\rho = 0.8$. This gave a final result of 0.5158 on the test dataset, which is quite the improvement. The training and validation loss can be seen in figure 3

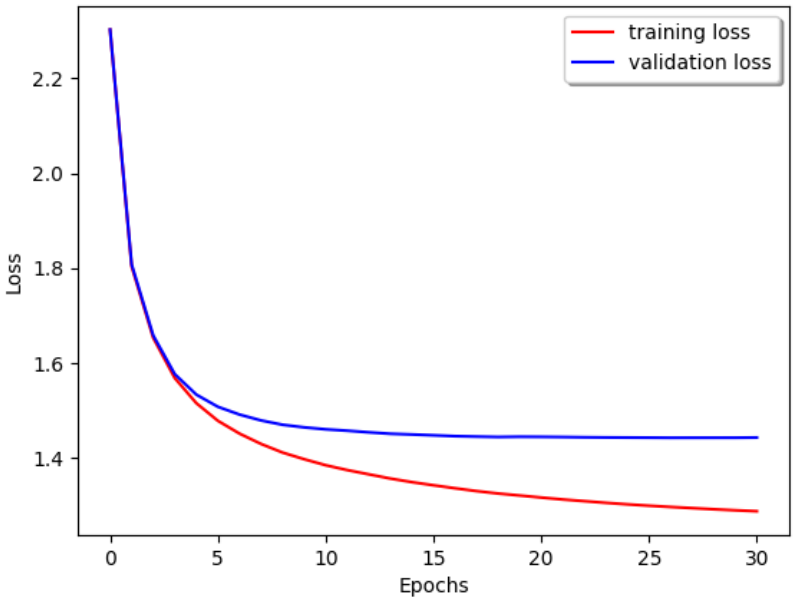

Figure 3: Final training of network: $\eta = 0.018$, $\lambda = 8.0e-04$, $\rho = 0.8$. Test accuracy = 0.5158