Deep Learning: Bonus Assignment 2

Anton Stråhle

April 7, 2020

Bonus 1

The improvements that I have chosen are the following

(i) Increasing the number of nodes in the hidden layer

(ii) Implementing dropout

(iii) Exhaustive search to find good values for the amount of regularization, the length of cycles, the number of cycles etc.

In the cases below we set $\eta_{\min}$ to 0.00001 and $\eta_{\max}$ to 0.1. The batch size is also set to 100.

The improvements that I have chosen to implement and the accuracies that they produce cannot really be compared on their own as they all depend on each other. By increasing the number of nodes in the hidden layer we get a more complex network architecture which allows for higher accuracies. On the other hand by doing this we need to regulate the network further which can be done by either tuning the regularization parameter λ or by for example implementing dropout. I have chosen to do both. As such the implementation does not increase the accuracy on it's own by rather allows for the increase brought by the usage of more nodes in the hidden layer.

The tuning of n_s as well as the number of cycles does of course have positive effects on the accuracy without the implementation of of any of the other improvements. With enough regularization these increase, at least in the scenarios where I attempted them, seemed to steadily increase the accuracy.

In order to achieve the highest accuracy possible I wanted to include as many nodes as possible in the hidden layer and then include dropout as well as tune the regularization parameter in order to further regularize, as would be necessary due to the more complex network architecture.

Initially I chose a large number of nodes in the hidden layer that I still felt was computationally feasible. This turned out to be around 300 (quite a significant

increase from the original 50). In order to deal with this large number of nodes in implemented dropout whilst coarsely searching for combinations of λ and the dropout probability p through a grid search that gave good accuracies on the validation set. For 300 nodes in the hidden layer a good pair turned out to be $\lambda = 0.00196$ and $p = 0.8$.

With these values I turned to the tuning of n_s as well as the number of cycles. For n_s I examined several multiples of $\frac{\#\text{data points}}{\#\text{batch size}}$. In the base part of the assignment the multiple 2 was suggested and as such I chose to examine the multiples above that. The reason for me examining these multiples as doing so would fit into the epoch structure of my mini-batch gradient descent. After a somewhat rough search through these multiples I found that 5 seemed to work well and generated good accuracies with the aforementioned parameters.

Lastly, I did a non-random grid search for the number of cycles to run where I achieved the highest validation accuracy of 58.7% at 8 cycles. With the parameters mentioned I achieved a test accuracy of 58.1%. I am sure that I would have been able to push the accuracy above 60% by using even more nodes in the hidden layer and further regularizing by tuning λ and p in order to accommodate this increase in the network architecture.