COL774 Assignment 3

Navneel Singhal

December 18, 2020

Contents

1 Neural Networks

1. The fundamental equations are as mentioned in class. The neural network is fully characterized by the following variables:

   (a) θ matrices

   (b) Features and target classes

   (c) Architecture and activation function at each node

   (d) Stochastic gradient descent parameters

   Note that the formulae derived in class were for a single training example, and for each example, there is a δ and a unit output for each sample in mini-batch; as it turns out, this can be vectorized as well.
   One very non-trivial task was to initialize θ, and this was done using He initialization, where the θ for a layer is sampled from a distribution (chosen to be uniform in my implementation) whose variance is $\frac{2}{n}$ where n is the number of units in this layer. This works for both sigmoid as well as ReLU.

2. In this part, my chosen stopping criterion is the difference of the average error over the current epoch and the average error of the previous epoch is less than a given ε. The chosen value of ε is 10^{-4} for the sake of good convergence.

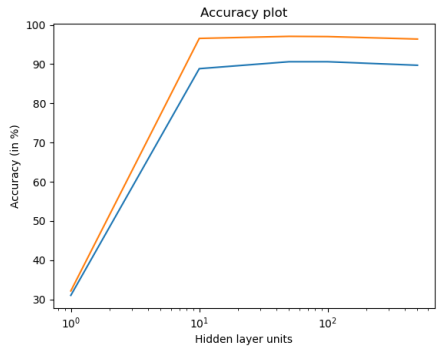
   The first table is for learning rate 0.1.

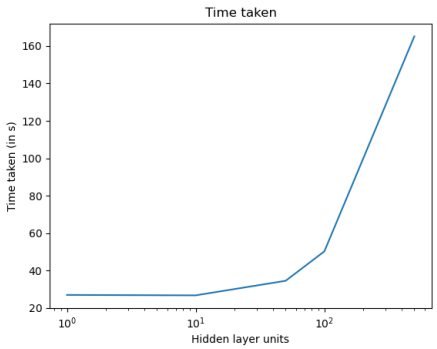
Hidden layer units	Test accuracy	Training accuracy	Time taken	Epochs
1	31.04%	32.165%	27.04s	127
10	88.84%	96.56%	26.80s	108
50	90.61%	97.08%	34.59s	103
100	90.61%	97.03%	50.31s	104
500	89.71%	96.39%	165.14s	79

   The second table is for learning rate 0.001.

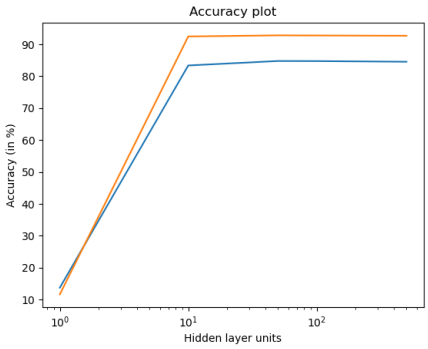
Hidden layer units	Test accuracy	Training accuracy	Time taken	Epochs
1	13.75%	11.66%	17.44s	102
10	83.38%	92.48%	205.93s	966
50	84.78%	92.82%	212.65s	618
100	84.75%	92.78%	311.73s	580
500	84.54%	92.69%	1216.10s	513

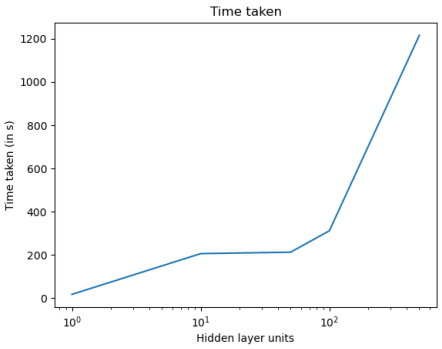
   The plots are as follows for learning rate 0.1:





The plots are as follows for learning rate 0.01:



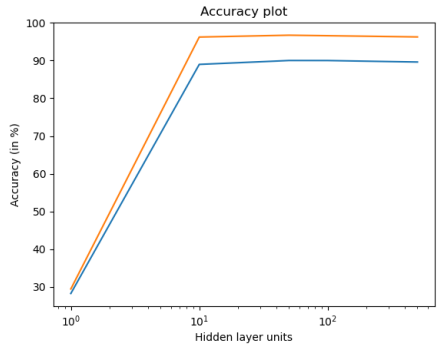


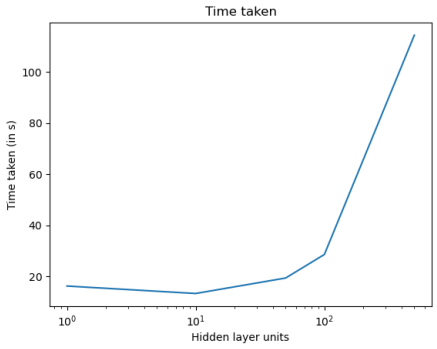
As can be seen from the plots, the training time increases quite quickly with the number of hidden layer units, however both the test and the training accuracies saturate after one point (in this case it is very slightly decreasing).

3. The stopping criterion didn't need to be changed and worked for both of these problems, and it is the same as the previous part.

Hidden layer units	Test accuracy	Training accuracy	Time taken	Epochs
1	28.26%	29.46%	16.25s	100
10	88.97%	96.22%	13.33s	69
50	90.00%	96.71%	19.39s	64
100	89.99%	96.57%	28.61s	63
500	89.60%	96.25%	114.46s	54

The plots are as follows:





The results are quite similar to the previous part, albeit with a very slight decrease in accuracy. The training time has indeed seen quite a good bit of reduction.

4. In this part, I experimented with a 2-hidden-layer neural network with 100 units each, with the hidden layers having sigmoid and ReLU activation.

Activation function	Test accuracy	Training accuracy	Time taken	Epochs
ReLU	93.21%	99.25%	25.85s	46
Sigmoid	90.4%	96.79%	41.17s	71

The test set accuracy when using ReLU activation is substantially better than the accuracy when using sigmoid activation, as can be seen in the table. As can be seen, ReLU performs better than sigmoid in both accuracy as well as time aspects. The lower time can be attributed to the fact that ReLU activation is faster to compute than sigmoid, as well as the lesser number of epochs needed for convergence.

For comparison with results using a single hidden layer with sigmoid, it can be seen that ReLU outperforms that, and sigmoid has almost the same accuracy.

5. In this part, I used the `sklearn` implementation of the neural network with a similar structure. The statistics are as follows.

Activation function	Test accuracy	Training accuracy	Time taken
ReLU	90.70%	97.95%	145.49s
Sigmoid	90.87%	96.79%	54.32s

As can be seen, my implementation performs better than the `sklearn` implementation when it comes to ReLU activation, and almost the same accuracy in the case of sigmoid. The time taken is also larger in the case of `sklearn` implementation, and I suspect that this is because of the initialization scheme (which I experimented with in part b as well).