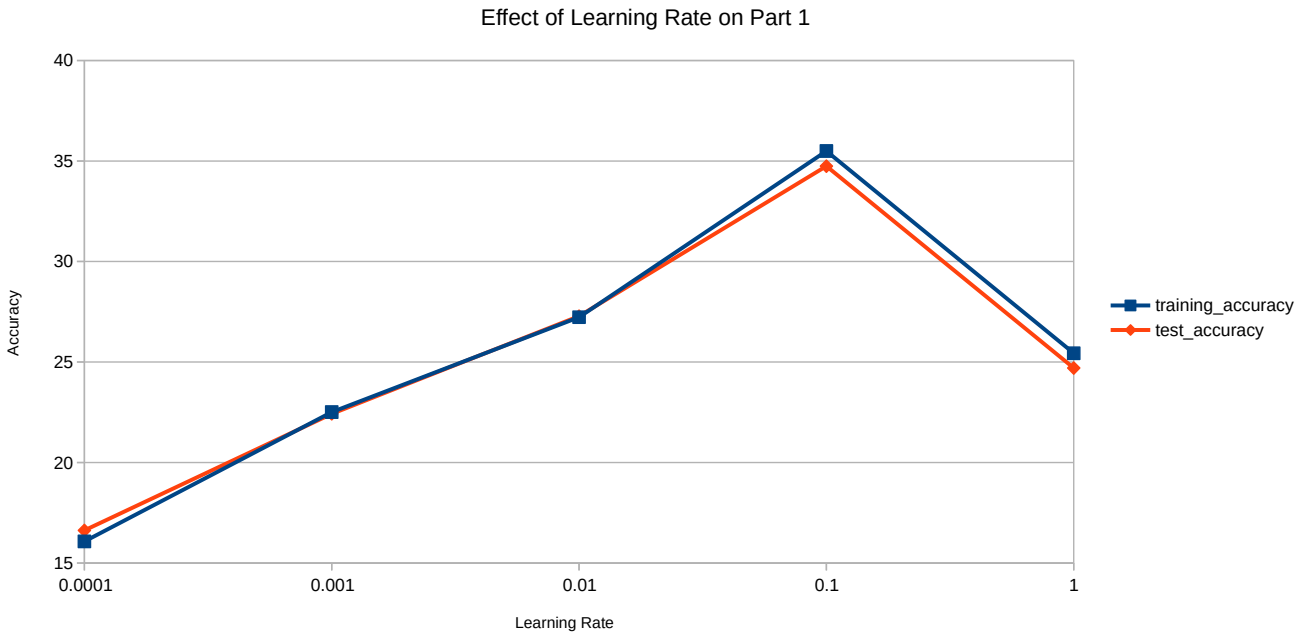
Implementation Assignment #3

Trevor Hammock, Charles Koll

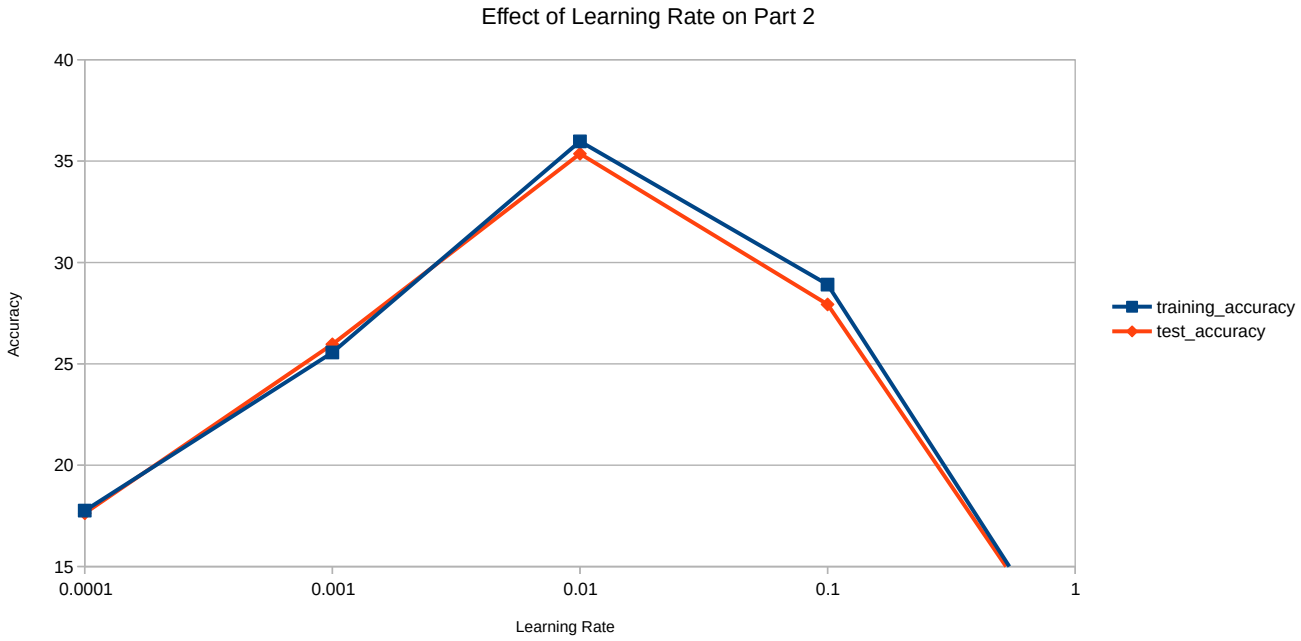
1. *What is a good learning rate that works for this data and this network structure? Present your plots for different choices of learning rates to help justify your final choice of the learning rate. How do you decide when to stop training? Evaluate your final trained network on the testing data (the test batch) and report its accuracy.*

A good learning rate is 0.1 for this data. We stopped training when we reached 5 epochs. If we kept training as long as the training accuracy increased, we would never stop.



2. *Repeat the same experiment as (1) but use Relu as the activation function for the hidden layer and answer the same questions as in (1).*

A good learning rate is 0.01 for this data. We stopped training when we reached 5 epochs. If we kept training as long as the training accuracy increased, we would never stop.



3. *Experiment with other parameters: drop out (d), momentum (m) and weight decay (wd). The goal is to improve the performance of the network by changing these hyper-parameters. Please describe what you have tried for each of these parameters. How do the choices influence the behavior of learning (as examined by monitoring the loss as a function of the training epochs)? Does it change how fast the training converges? How do they influence the testing performance? Please provide a summary of the results and discuss the impact of these parameters.*

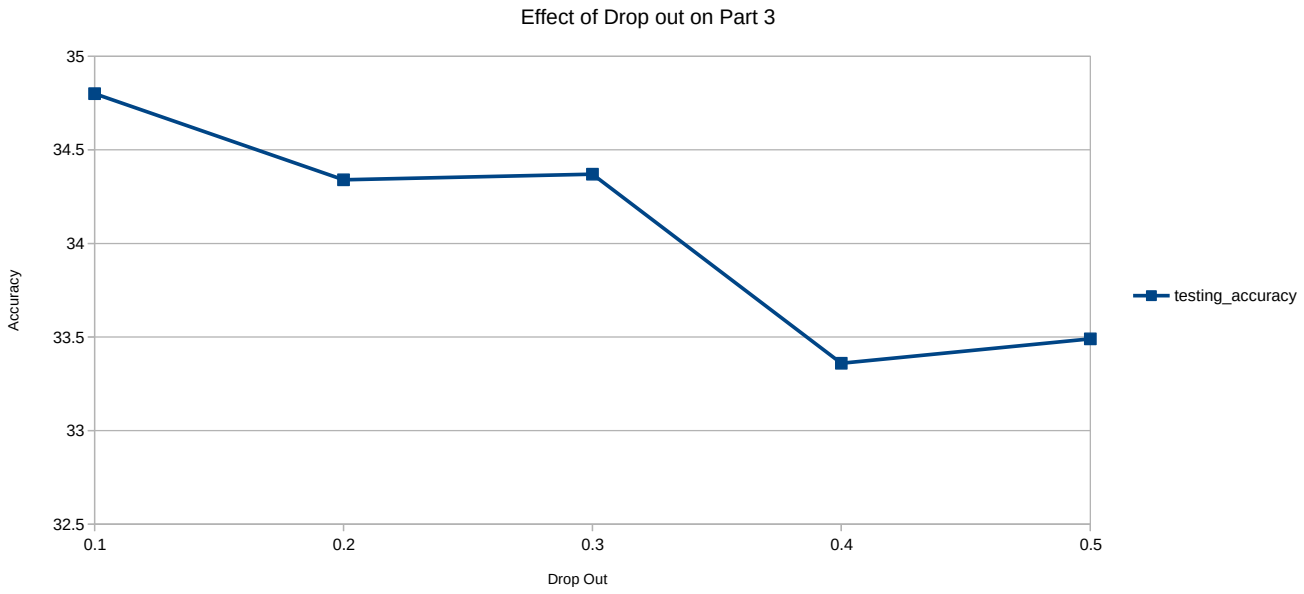
Drop out: (0.1, 0.2, 0.3, 0.4, 0.5)

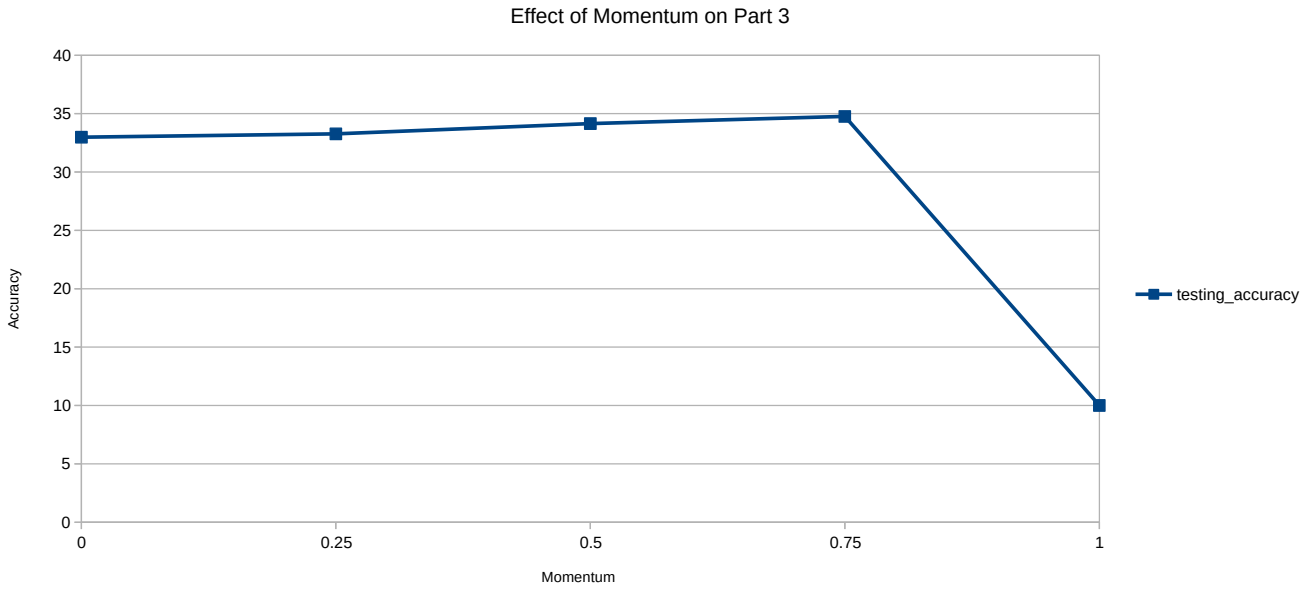
Momentum: (0, 0.25, 0.5, 0.75, 1)

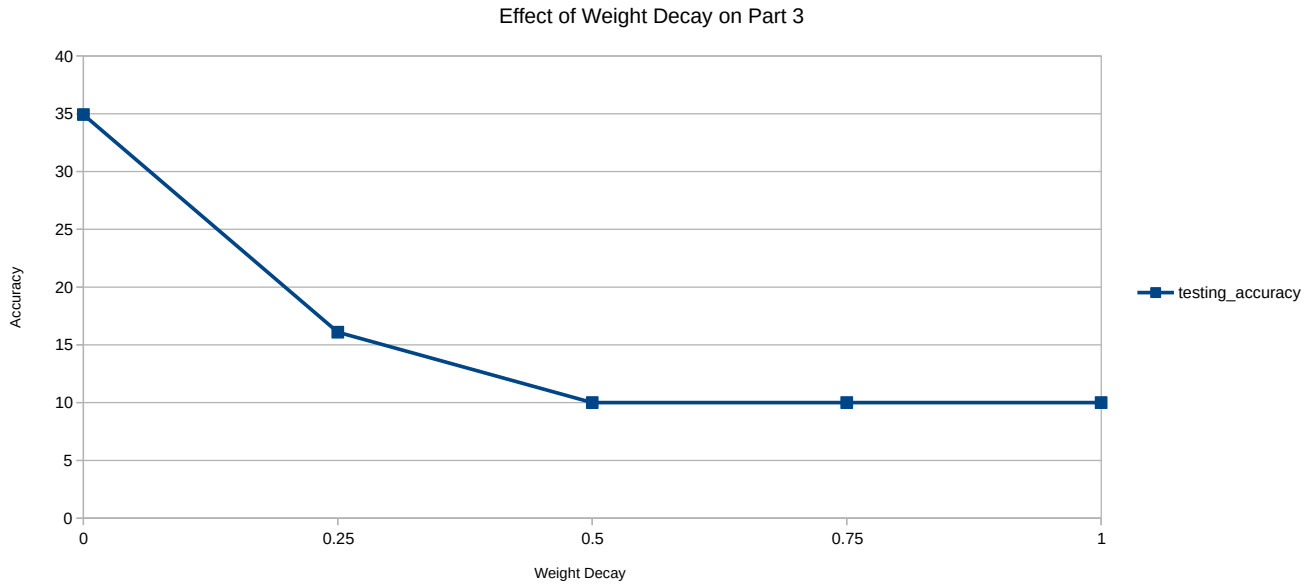
Weight decay: (0, 0.25, 0.5, 0.75, 1)

Fixed values: drop out – 0.2; momentum – 0.5; weight decay – 0

Changing drop out did not significantly impact the testing error. Changing momentum did not significantly impact the testing error except when momentum = 1, at which point it raised it by over 20%. Changing weight decay raised the testing error except for when weight decay = 0.


Effect of Drop out on Part 3
35
34.5
34
33.5
33
32.5
Accuracy
0.1
0.2
0.3
0.4
0.5
Drop Out
testing_accuracy


Effect of Momentum on Part 3
40
35
30
25
20
15
10
5
0
Accuracy
0
0.25
0.5
0.75
1
Momentum
testing_accuracy



4. *Train the new network with the same loss function, the SGD optimizer, and your choice of activation function for the hidden layers. What do you observe in terms of training convergence behavior? Do you find one structure to be easier to train than the other? How about the final performance of the network in terms of training error and testing error? Which one gives you better testing performance?*

The result of learning rates on accuracy is similar between the network we trained in Part 2 and this one in Part 4, except the one in Part 2 had slightly higher accuracy. Since Part 4 took longer to compute and had lower accuracy, it appears that Part 2 is a better model considering the parameters we used.