CS434 Assignment #3 Report

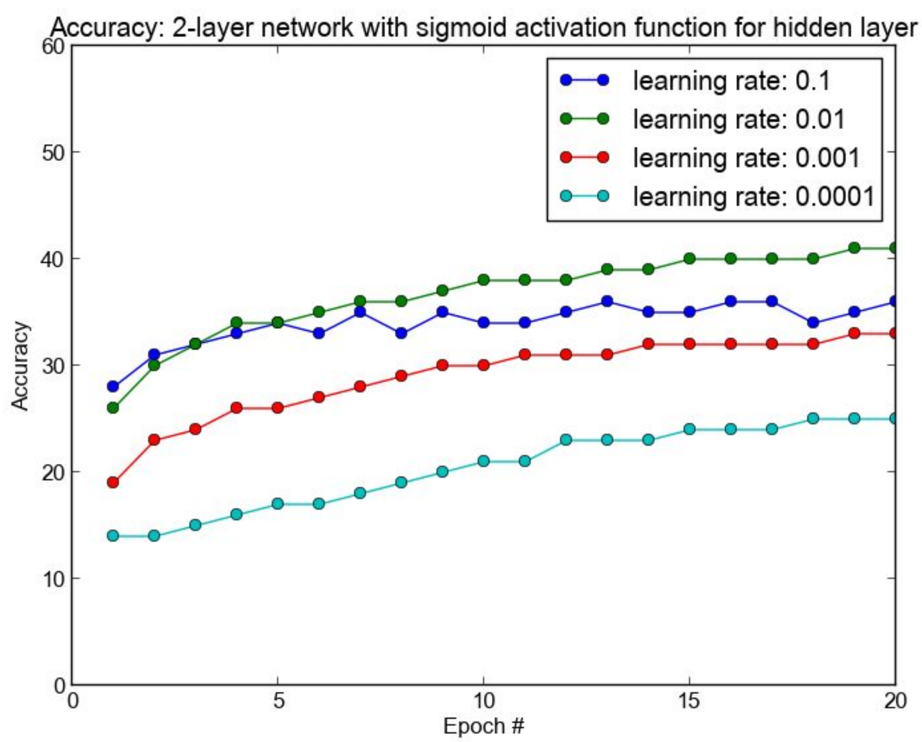
Name: Tsewei Peng

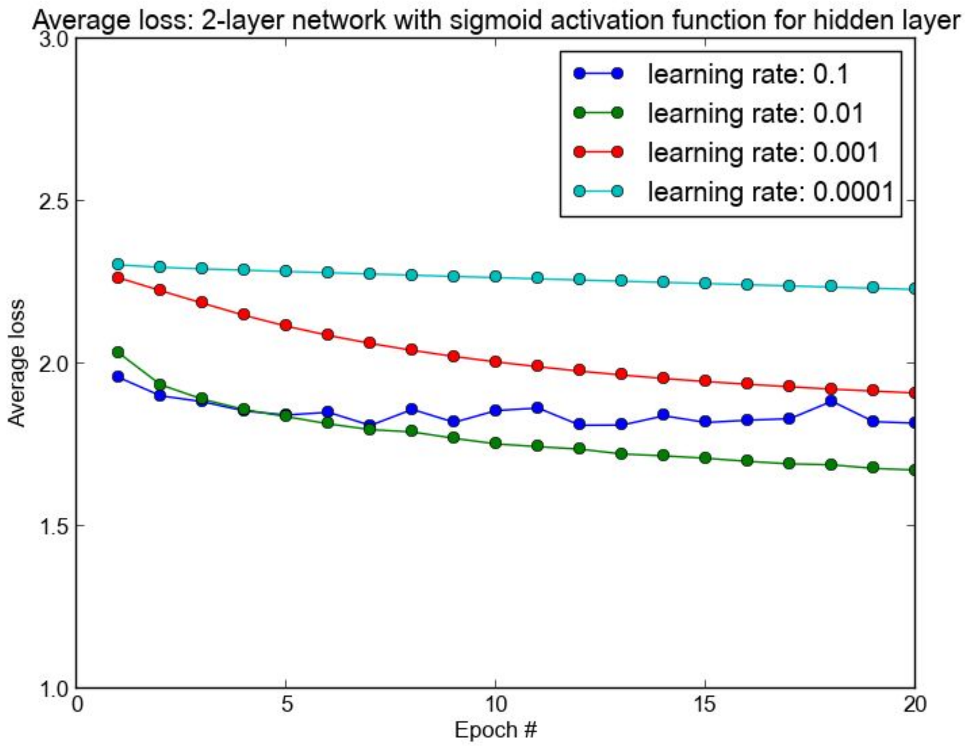
Note: the program contains all the training functions for all problems. Please make sure to comment some functions out to run one specific function, otherwise the program takes a long time to run.

Problem 1. Sigmoid Activation Function

A good learning rate for this data and network structure is **0.01**.
Below are the plots average loss (negative log-likelihood) and validation accuracy for the 2-layer network with different learning rate. As we can see from the plot, with 20 epochs of training, learning rate of 0.01 produces the least average loss and the highest accuracy. Learning rates of 0.001 and 0.0001 converge slower than 0.01, while learning rate of 0.1 converges too fast, causing the accuracy to fluctuate too much.

After much testing, I found that 20 epochs is a good stopping point because the network accuracy on validation data stops growing significantly at around 15 epochs. Also, with 20 epochs we are able to see the trend of the training convergence with different learning rate. I decided not to use threshold of accuracy change as stopping points because it is dependent on the learning rate; smaller learning rate means smaller change of accuracy making it hard to decide what is a good threshold.





The final trained network with learning rate of 0.01 on testing data: (outputted from program)

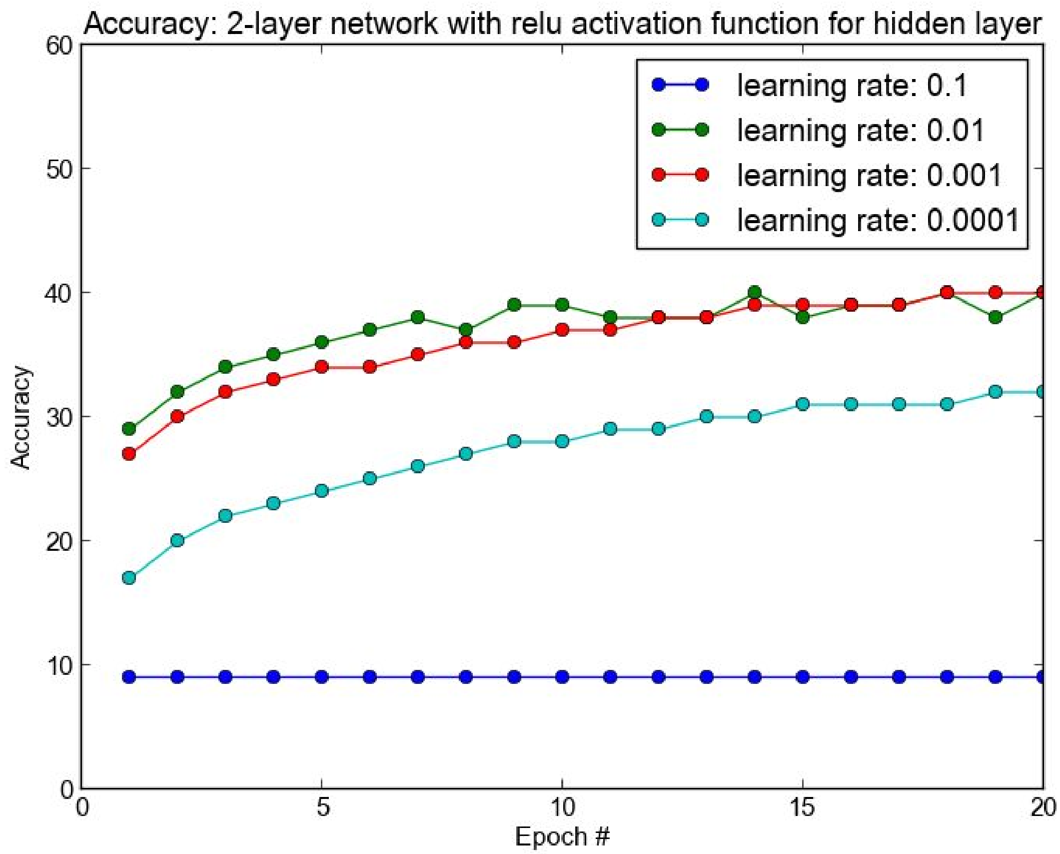
Learning Rate	Testing Accuracy
0.1	37%
0.01	41%
0.001	33%
0.0001	25%

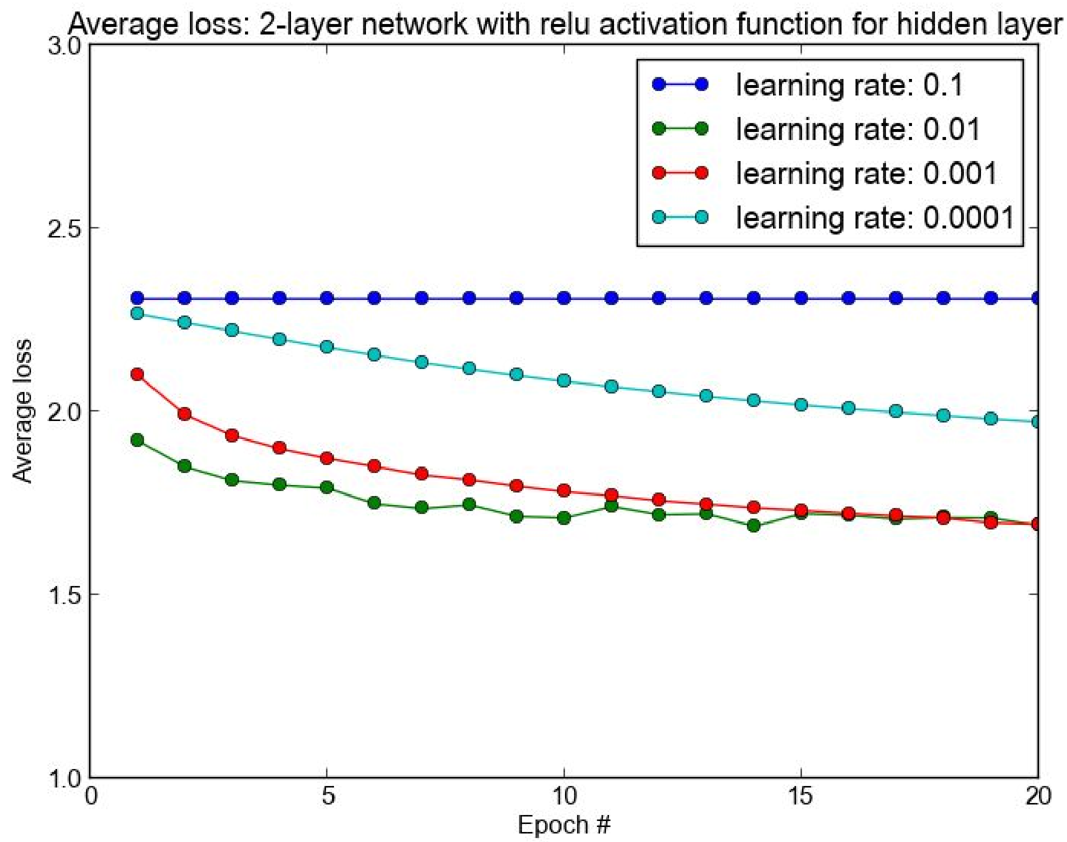
Problem 2. Relu activation function

A good learning rate for this data and network structure is 0.01.

Below are the plots average loss (negative log-likelihood) and validation accuracy for the 2-layer network with different learning rate. As we can see from the plot, with 20 epochs of training, learning rate of 0.01 and 0.001 produces similar result in terms of accuracy. However, learning rate of 0.01 converges much faster than learning rate of 0.001, therefore learning rate of 0.01 is the better learning rate. The other two learning rate, 0.1 and 0.0001, are not a good learning rate; 0.0001 converges too slow, while 0.1 has too much fluctuation, causing the accuracy to not converge at all.

After much testing, I found that 20 epochs is a good stopping point because the network accuracy on validation data stops growing significantly at around 15 epochs. Also, with 20 epochs we are able to see the trend of the training convergence with different learning rate. I decided not to use threshold of accuracy change as stopping points because it is dependent on the learning rate; smaller learning rate means smaller change of accuracy making it hard to decide what is a good threshold.





The final trained network with learning rate of 0.01 on testing data: (Outputted from program)

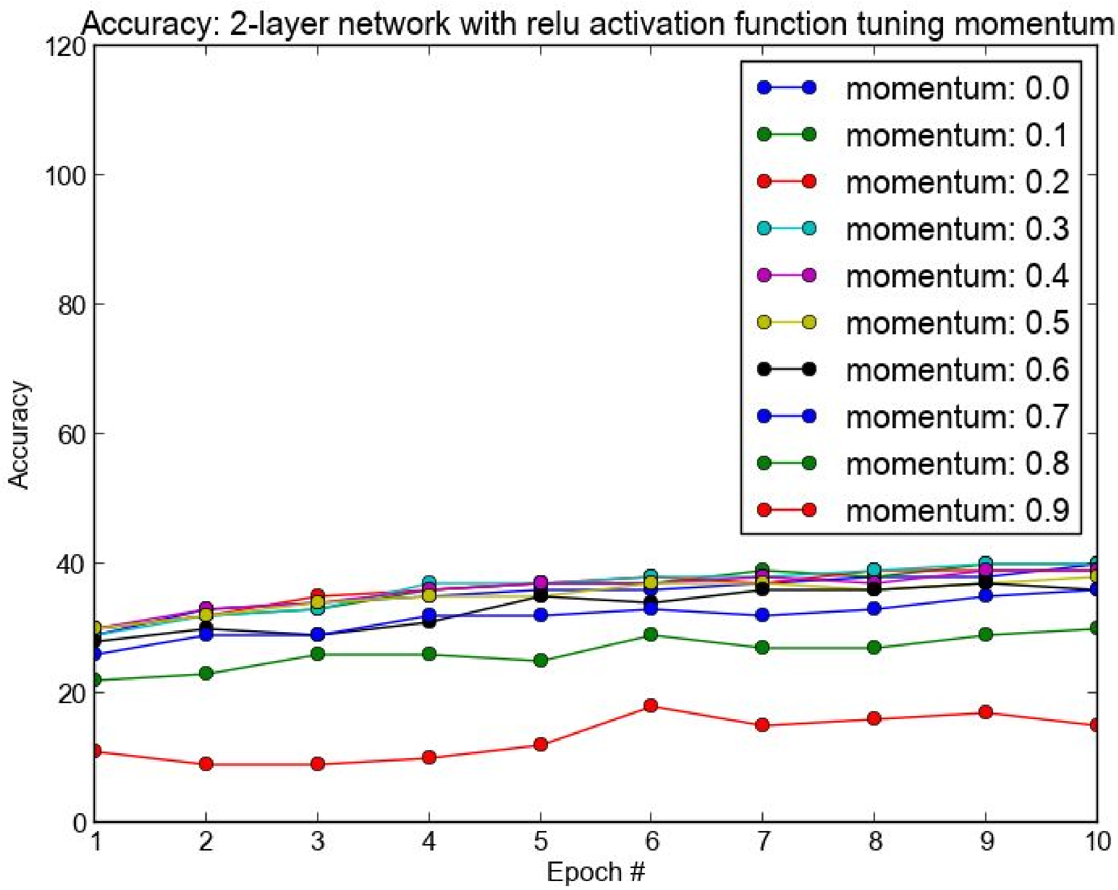
Learning Rate	Testing Accuracy
0.1	10
0.01	40
0.001	40
0.0001	32

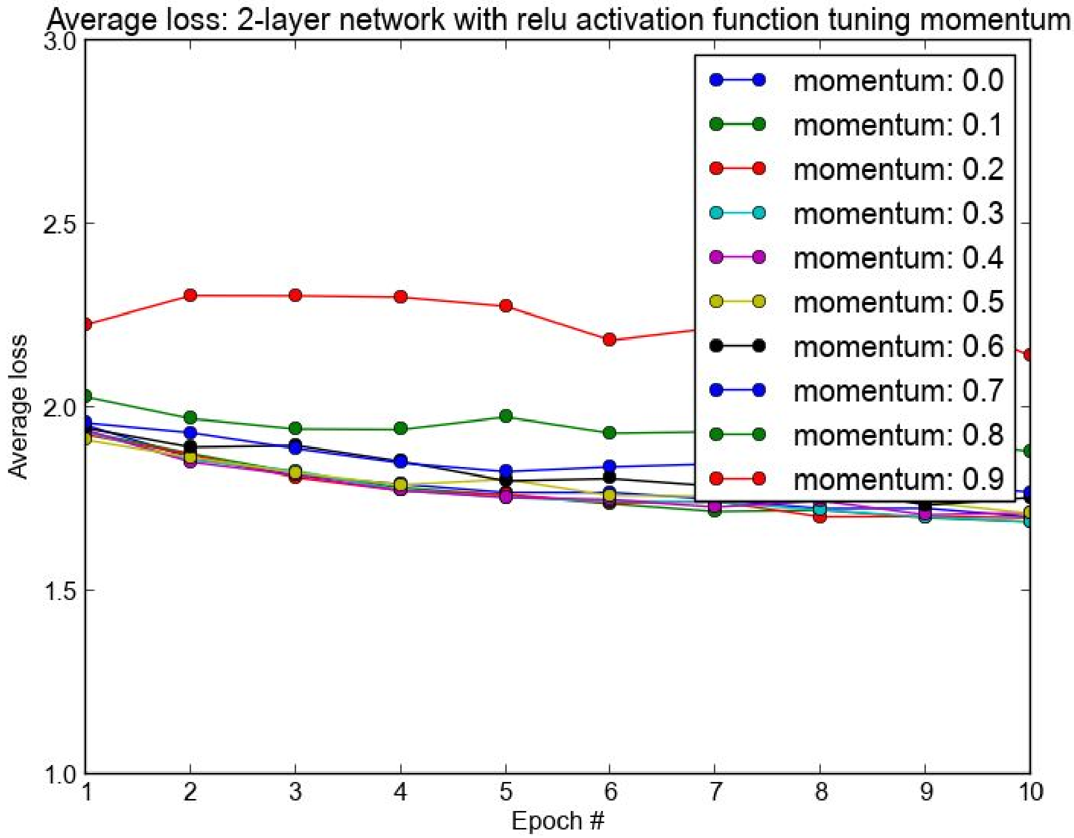
Problem 3. Tuning 2-layer network with relu activation function

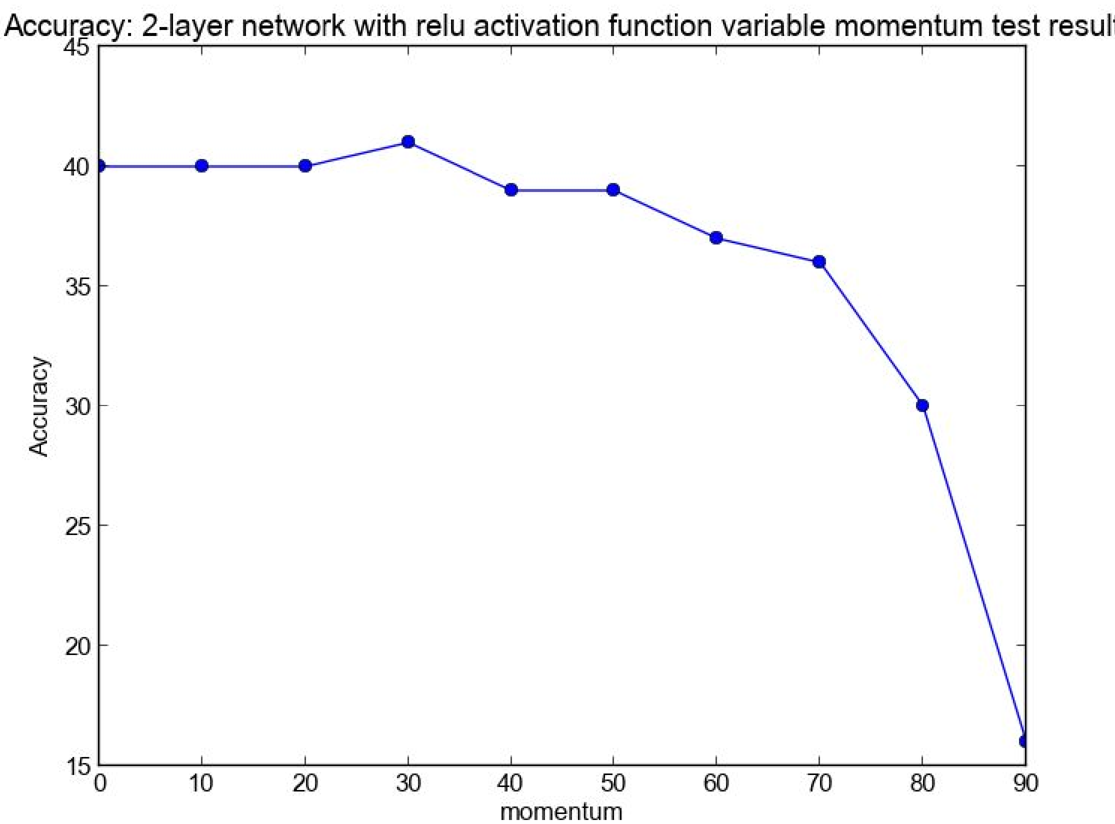
After some testing I decided to change the epochs time to 10 to increase my efficiency in completing this assignment. Since problem 2 suggest that learning rate of 0.01 is optimal, the program will go along with that.

Momentem

The first variable tested is the momentum. The range of momentum tested ranges from 0 to 0.9, with a 0.1 step increment, resulting in 10 results. Below are the graphs of validation accuracy and average loss for each result for each epoch. The last graph is the testing accuracy for each resulting network model.

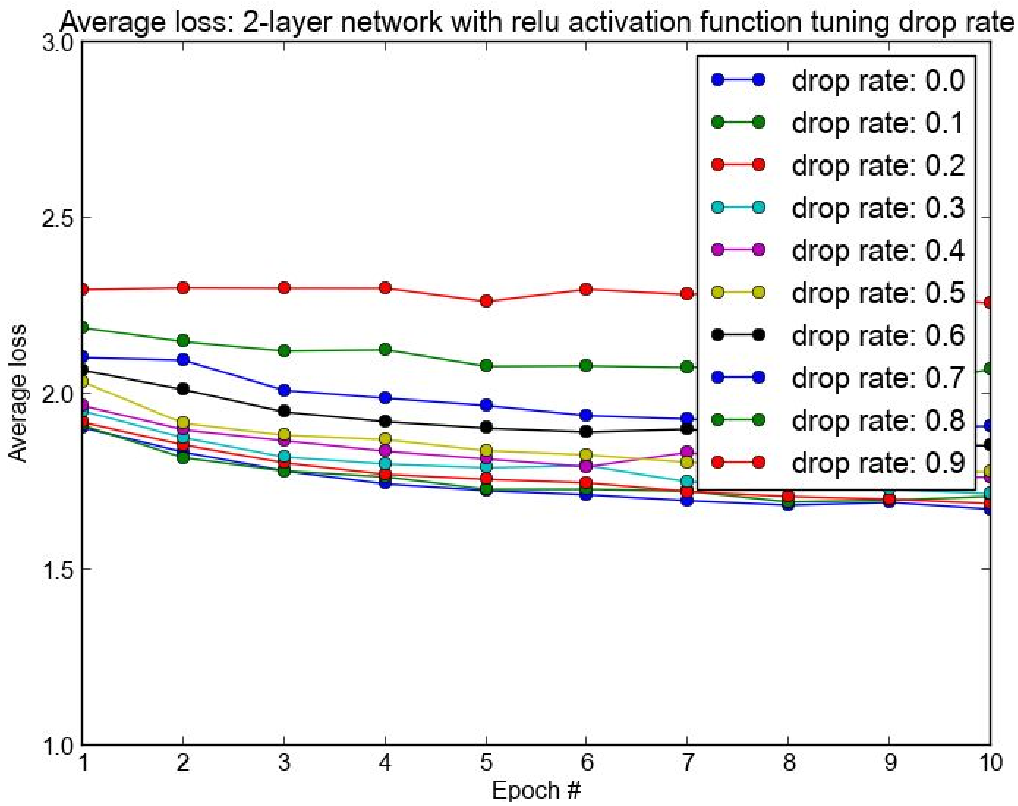


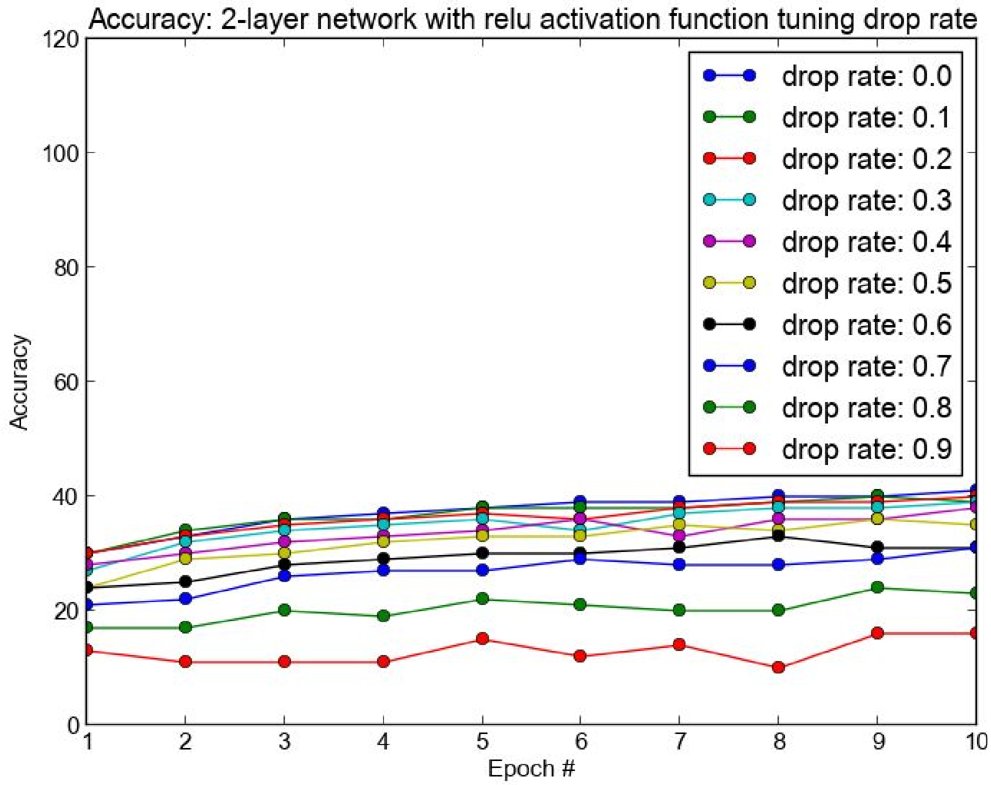

Average loss: 2-layer network with relu activation function tuning momentum
Average loss
Epoch #
momentum: 0.0
momentum: 0.1
momentum: 0.2
momentum: 0.3
momentum: 0.4
momentum: 0.5
momentum: 0.6
momentum: 0.7
momentum: 0.8
momentum: 0.9


Accuracy: 2-layer network with relu activation function variable momentum test result
Accuracy
momentum

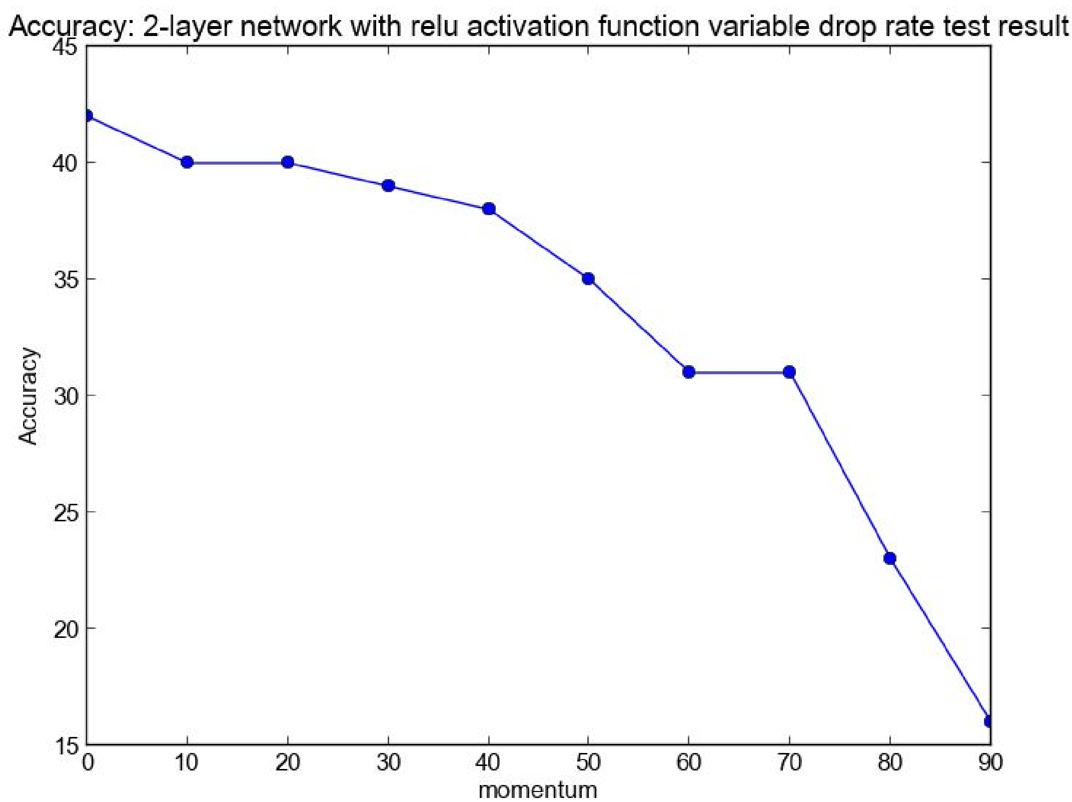
As we can see, the validation accuracy is generally highest with 0.3 momentum. As momentum increases, initially the speed of convergence is around the same, and when it passes 0,4, the speed of convergence decreases rapidly. In our testing accuracy, 0.3 momentum peaks the accuracy. 0.3 is chosen as the optimal momentum.
Optimal momentum: 0.3

Drop Rate
After setting the optimal momentum, the next variable tested is drop rate. The range of drop rate tested ranges from 0 to 0.9, with a 0.1 step increment, resulting in 10 results. Below are the graphs of validation accuracy and average loss for each result for each epoch. The last graph is the testing accuracy for each resulting network





Accuracy: 2-layer network with relu activation function tuning drop rate
drop rate: 0.0
drop rate: 0.1
drop rate: 0.2
drop rate: 0.3
drop rate: 0.4
drop rate: 0.5
drop rate: 0.6
drop rate: 0.7
drop rate: 0.8
drop rate: 0.9
Accuracy
Epoch #
0
20
40
60
80
100
120
1
2
3
4
5
6
7
8
9
10



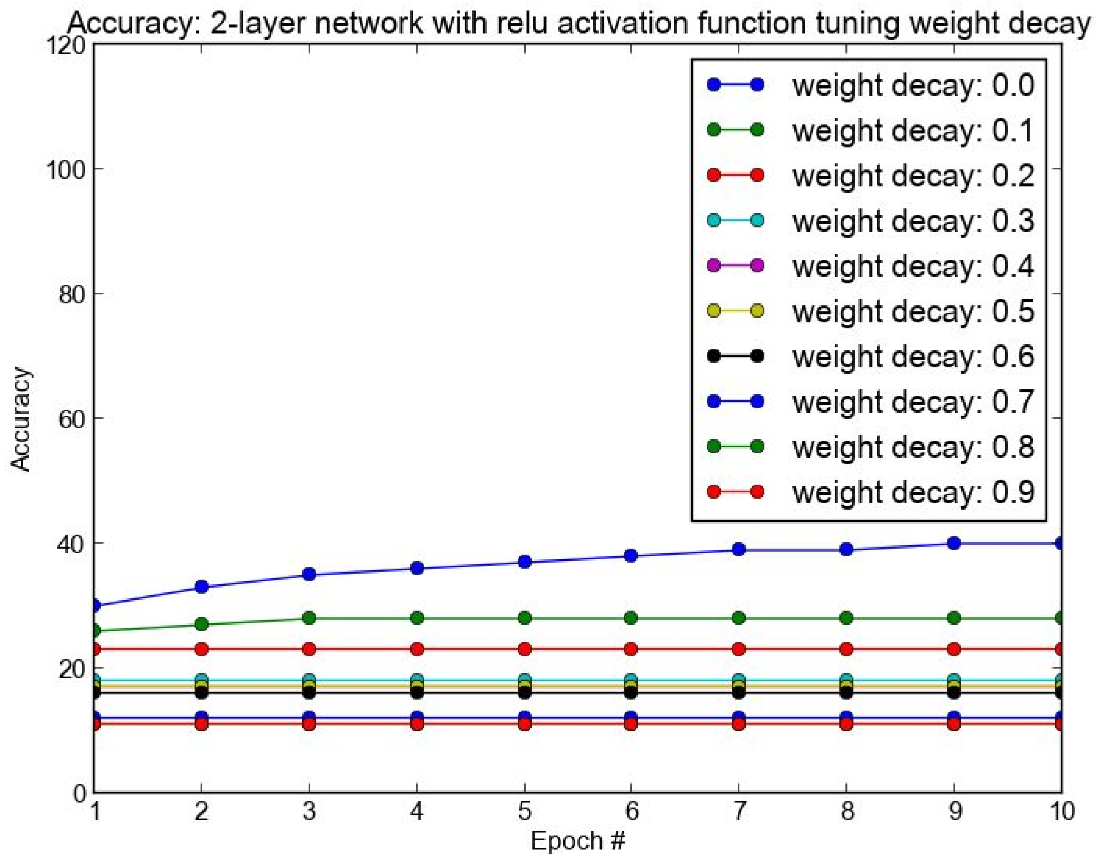
Accuracy: 2-layer network with relu activation function variable drop rate test result
Accuracy
momentum
15
20
25
30
35
40
45
0
10
20
30
40
50
60
70
80
90

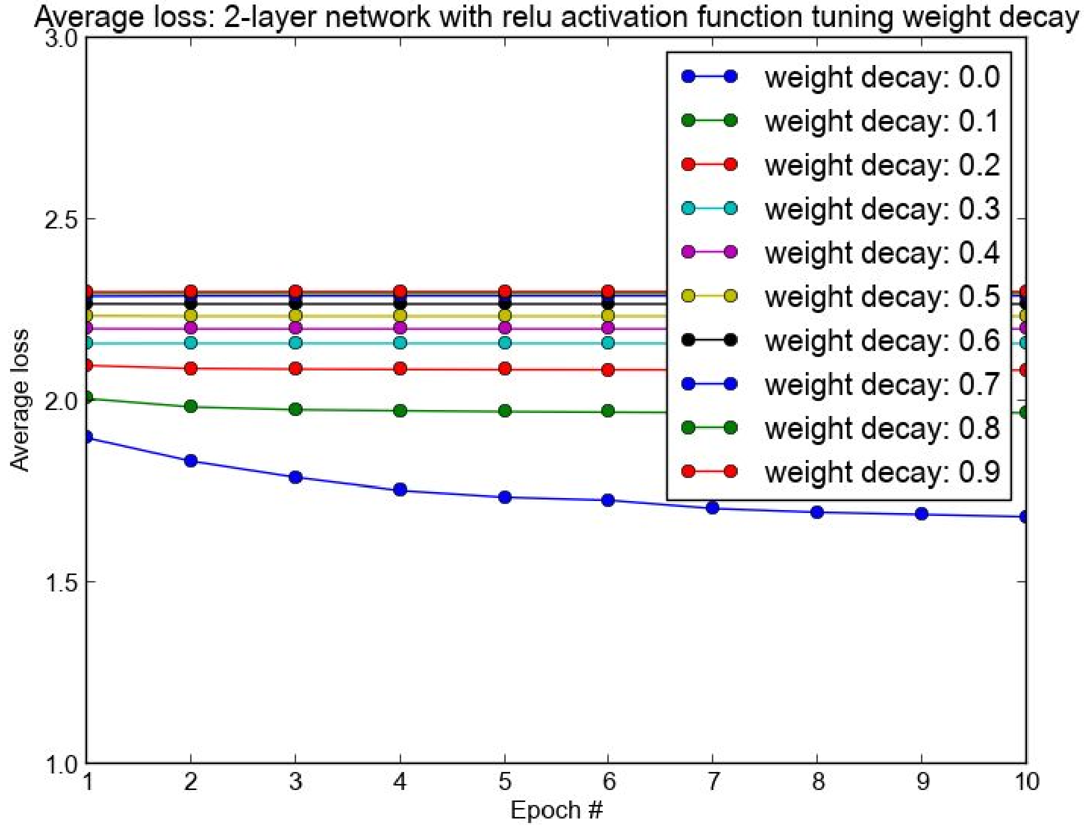
As we can see, when drop rate is 0, 0.1, 0.2, the validation accuracy converges really fast comparing to other drop rates, and it is hard to distinguish them on the graph because the lines are so close. But on the testing accuracy graph, we can see that drop rate of 0 leads to highest testing accuracy, so the optimal drop rate is 0.

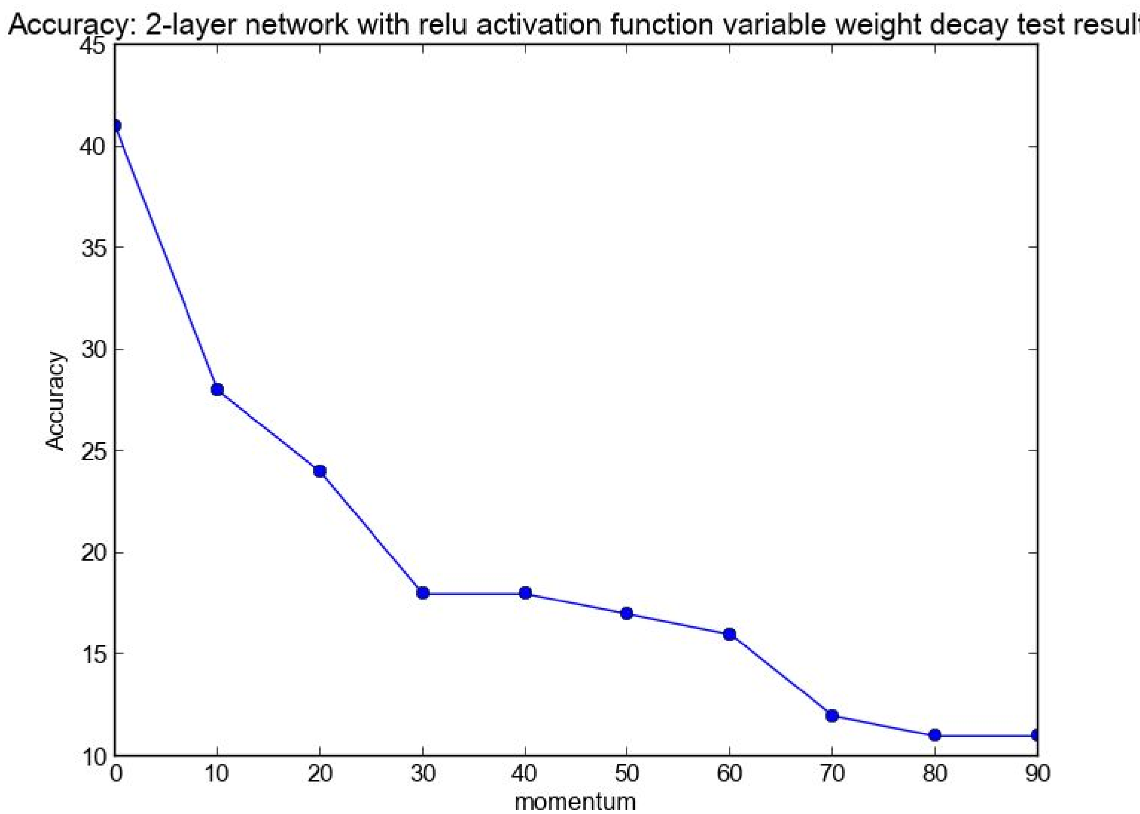
Optimal drop rate: 0

Weight Decay

After setting the optimal drop rate, the last variable tested is weight decay. The range of weight decay tested ranges from 0 to 0.9, with a 0,1 step increment, resulting in 10 results. Below are the graphs of validation accuracy and average loss for each result for each epoch. The last graph is the testing accuracy for each resulting network model.




Average loss: 2-layer network with relu activation function tuning weight decay
Average loss
Epoch #
weight decay: 0.0
weight decay: 0.1
weight decay: 0.2
weight decay: 0.3
weight decay: 0.4
weight decay: 0.5
weight decay: 0.6
weight decay: 0.7
weight decay: 0.8
weight decay: 0.9
3.0
2.5
2.0
1.5
1.0
1
2
3
4
5
6
7
8
9
10


Accuracy: 2-layer network with relu activation function variable weight decay test result
Accuracy
momentum
45
40
35
30
25
20
15
10
0
10
20
30
40
50
60
70
80
90

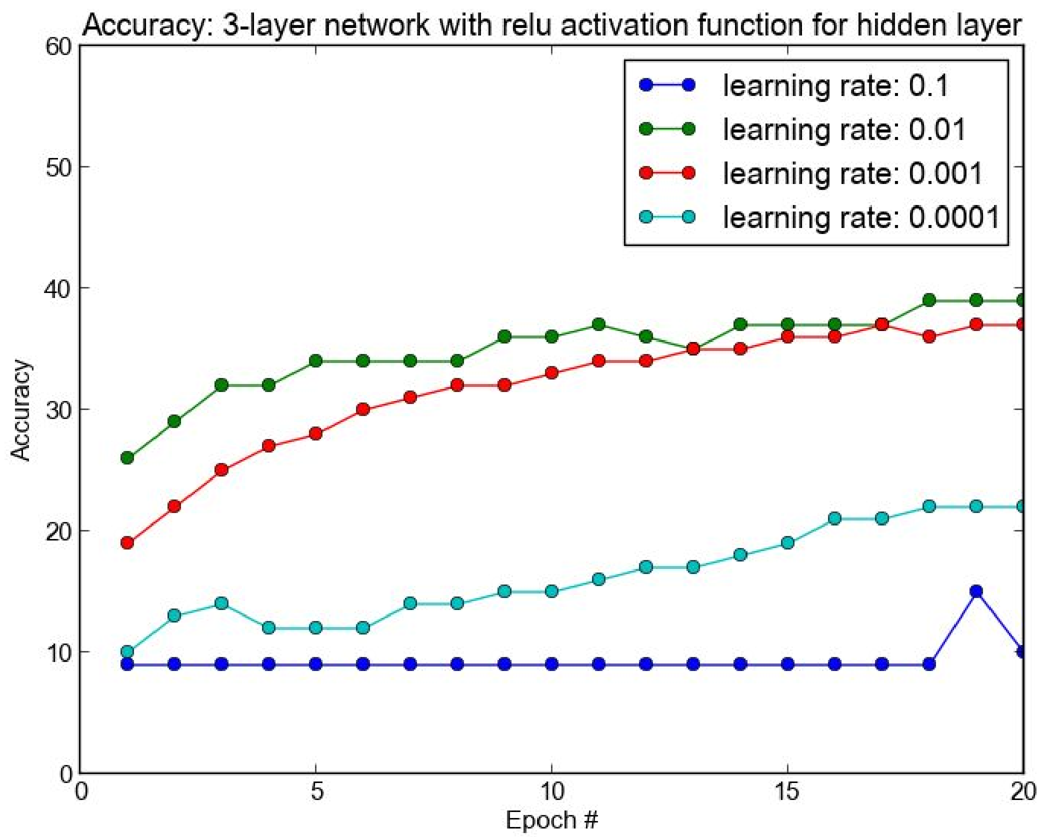
As we can see, the validation accuracy converges the fastest when weight decay is 0. The lower the weight, the faster the convergence. The testing accuracy is the highest when weight decay is 0, and as weight decay increases, the testing accuracy decreases rapidly. The optimal weight decay is 0.
Optimal weight decay: 0

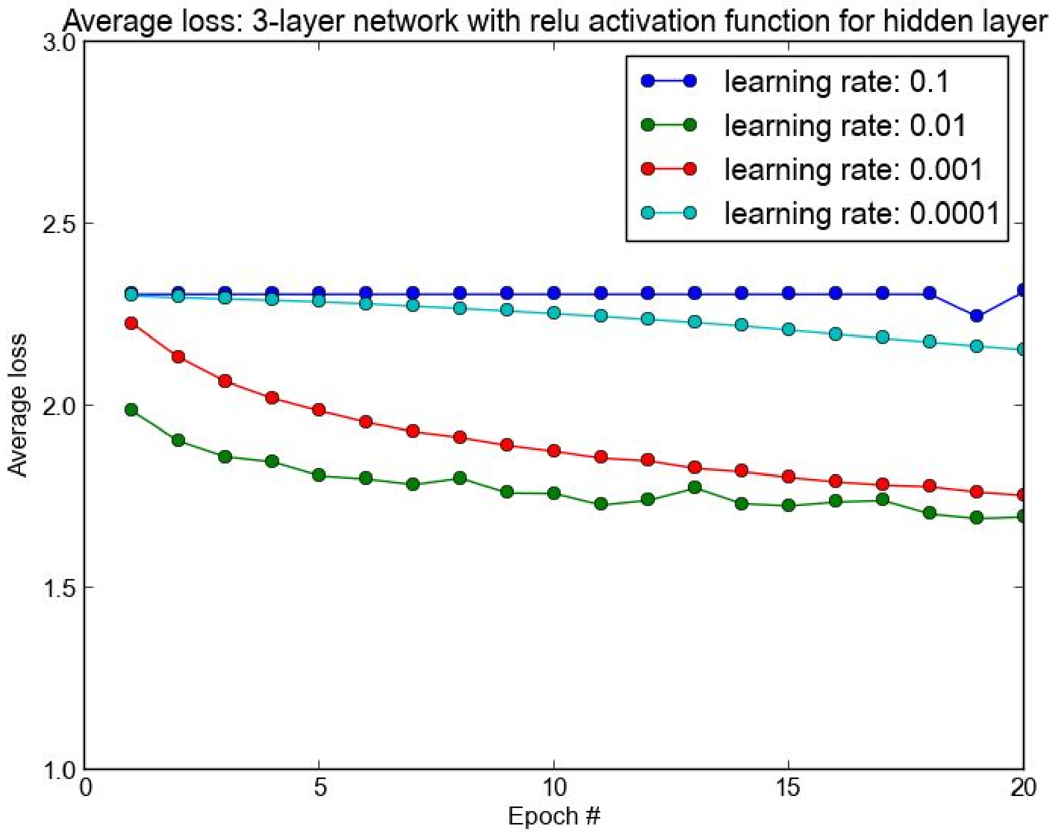
With all the variables settings set to optimal, the final testing accuracy with 20 epochs:
Final Testing Accuracy: 40%

Problem 4. Tuning 3-layer network with relu activation function

Selected activation function for hidden layers: **relu** activation function

First I tested different learning rate to decide which one is optimal. Graphs below are the validation accuracy, average loss for each learning rate throughout the training.





In terms of final performance, the result are as follows: (Outputted from program)

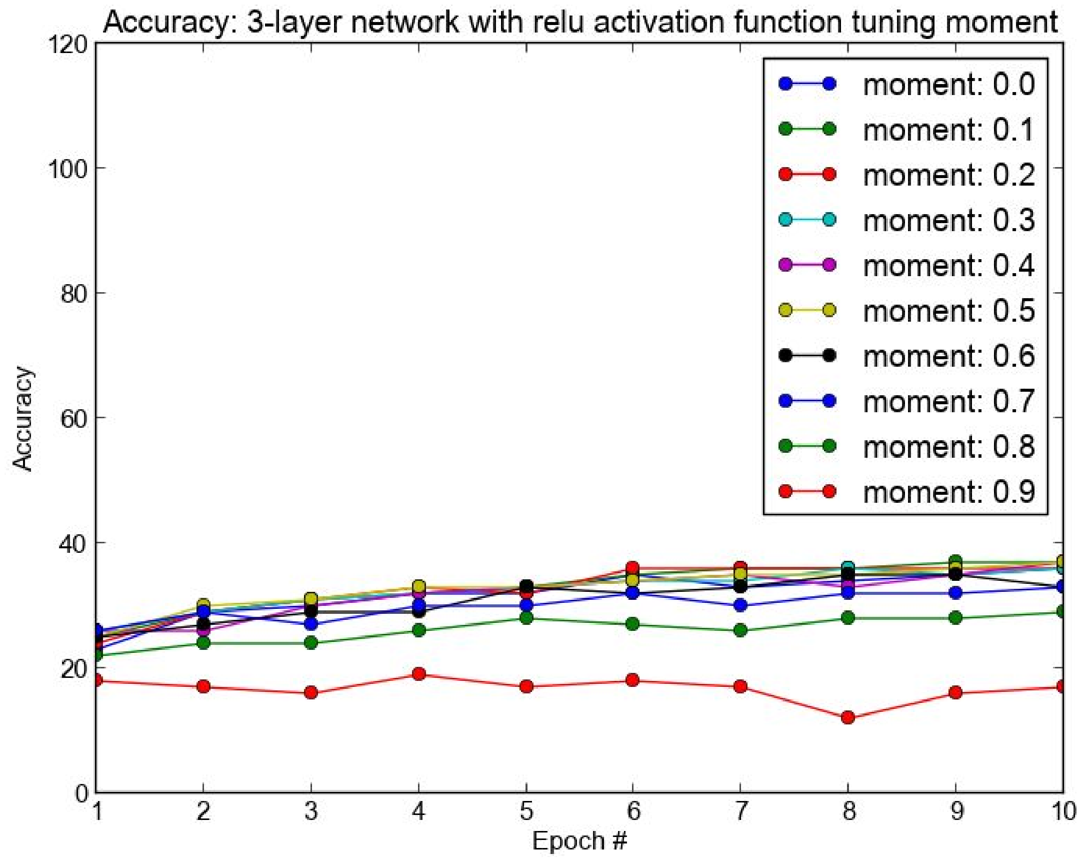
Learning rate	Testing Accuracy
0.1	15
0.01	36
0.001	36
0.0001	25

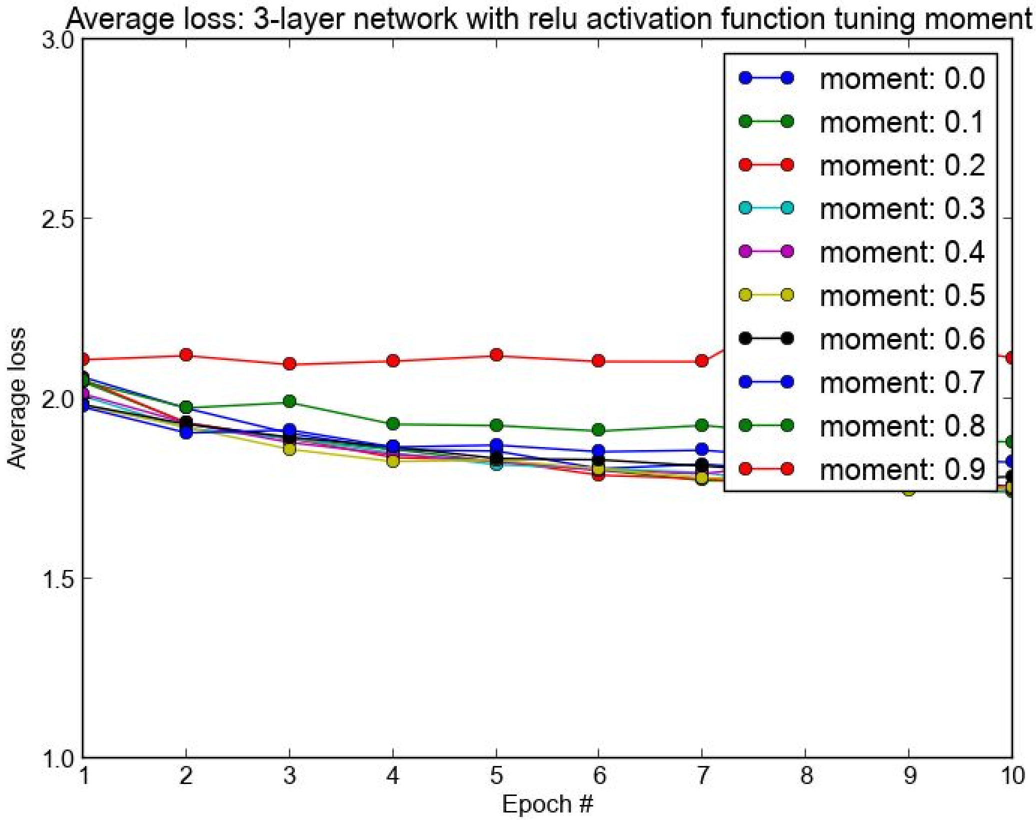
As we can see, after 20 epochs, the testing accuracy of learning rates 0.01 and 0.001 are very close. However the validation accuracy converges faster with learning rate of 0.01, so learning rate of 0.01 is the optimal learning rate for this model of network.

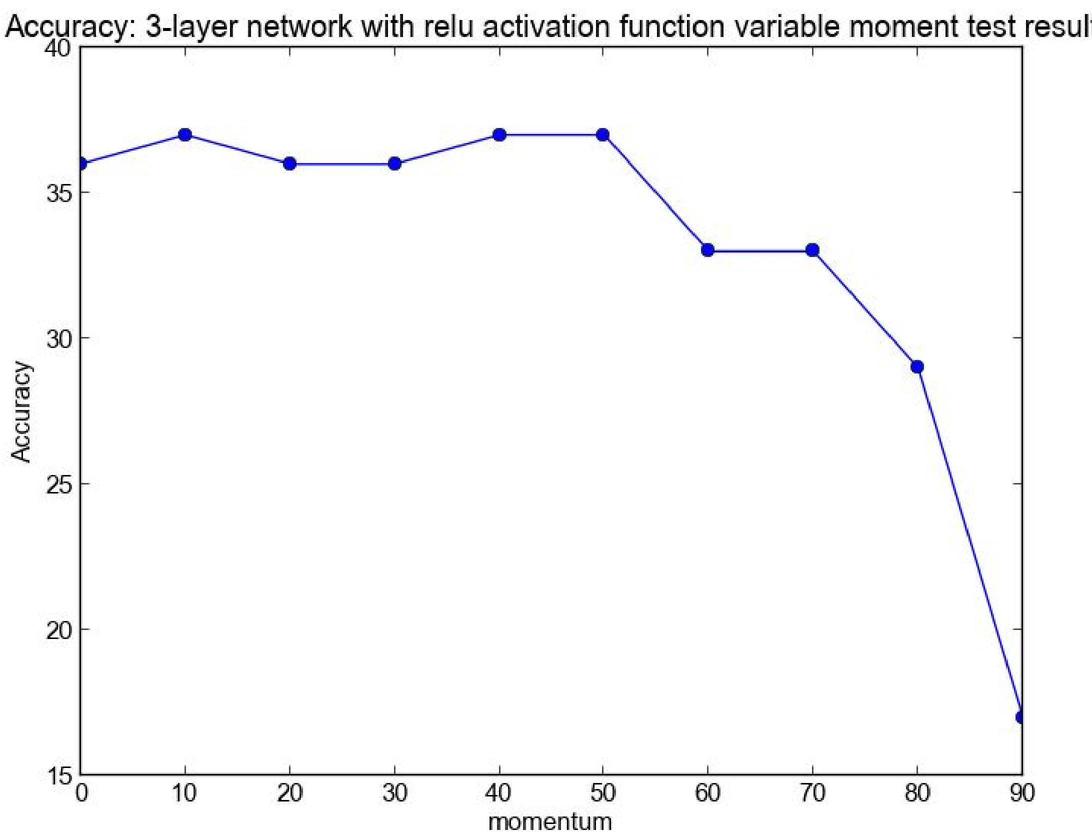
Optimal learning rate: 0.01

After getting the optimal learning rate, the variable to be tested is momentum. The range of momentum tested ranges from 0 to 0.9, with a 0.1 step increment, resulting in 10 results.

Below are the graphs of validation accuracy and average loss for each result for each epoch. The last graph is the testing accuracy for each resulting network model.

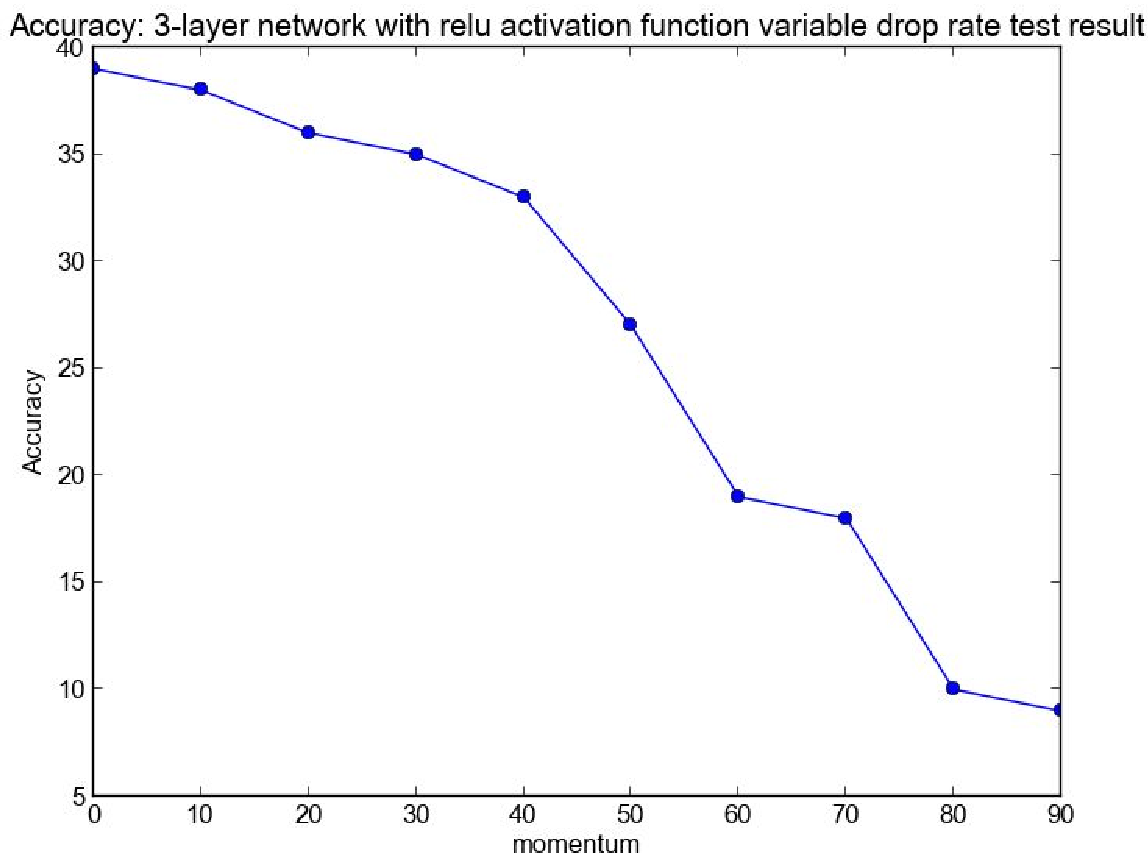



Average loss: 3-layer network with relu activation function tuning moment
Average loss
Epoch #
3.0
2.5
2.0
1.5
1.0
1
2
3
4
5
6
7
8
9
10
moment: 0.0
moment: 0.1
moment: 0.2
moment: 0.3
moment: 0.4
moment: 0.5
moment: 0.6
moment: 0.7
moment: 0.8
moment: 0.9


Accuracy: 3-layer network with relu activation function variable moment test result
Accuracy
momentum
40
35
30
25
20
15
0
10
20
30
40
50
60
70
80
90

As we can see, when momentum is below 0.6, the validation accuracy converges relatively fast. However, testing accuracy is the highest at momentum 0,1, 0,4, and 0,5. Momentum of 0,1 is chosen because the testing accuracy takes a dip right after 0.5. This can mean that testing accuracy for momentum 0.4 and 0.5 are outliers.
Optimal momentum: 0.1

After getting the optimal momentum, the variable to be tested is drop rate. The range of drop rate tested ranges from 0 to 0.9, with a 0.1 step increment, resulting in 10 results. Below are the graphs of testing accuracy for each resulting network model.



As we can see from the above graph, the optimal drop rate is 0.
Optimal drop rate: 0

With all the variables settings set to optimal, the final testing accuracy with 20 epochs:
Final Testing Accuracy: 41

In terms of training convergence behavior, this model is not as fast as the previous models. The resulted testing accuracy is also worse than previous models. This network model is not the best model for this dataset. All the models are not easy to train, but it is interesting to find that sometimes deeper neural network does not produce the best network model.