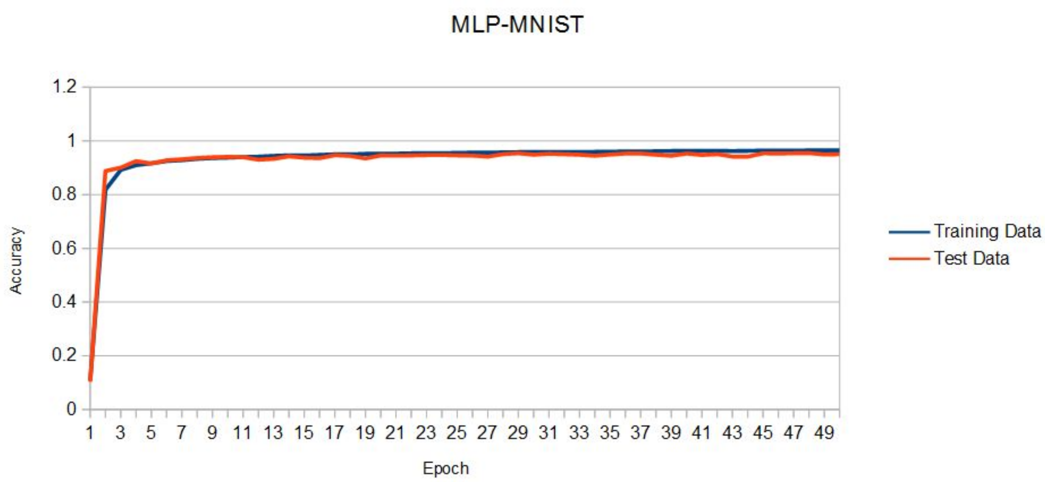
Merlin Carson
CS445, Winter 2018

Programming Assignment #1: MLP-MNIST

Experiment #1:

1) 20 Hidden Layer nodes

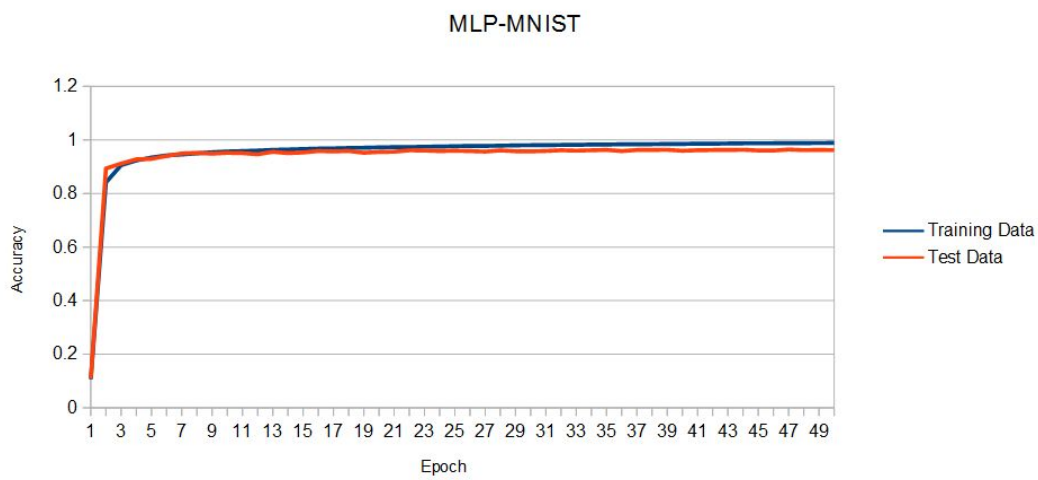


Test -- Confusion Matrix

965	0	3	0	0	2	0	4	4	2
0	1120	5	1	1	2	1	2	3	0
8	6	977	6	5	2	4	8	15	1
1	0	10	946	2	16	0	10	19	6
3	1	6	0	916	0	9	1	1	45
7	1	1	16	0	843	6	2	8	8
21	3	8	0	15	7	896	1	7	0
2	9	10	1	5	1	0	966	3	31
11	1	6	9	6	7	4	4	903	23
6	5	2	7	6	0	0	7	1	975

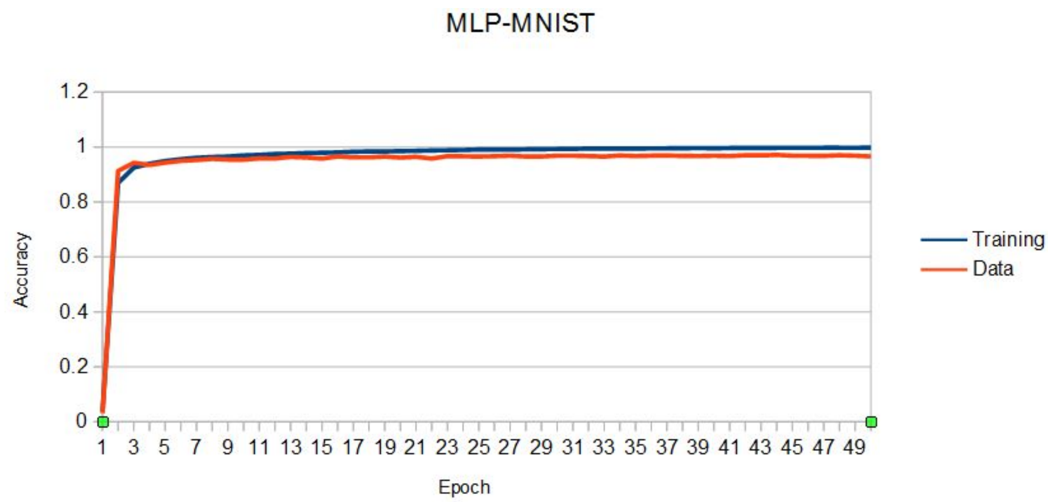
train accuracy: 0.964817 test accuracy: 0.9507

2) 50 Hidden Layer nodes




```
Test -- Confusion Matrix
   958     1     6     1     1     1     0     2     3     7

     0  1113     3     3     2     2     0     3     8     1

     5     1   990     2     4     3     2     9    14     2

     1     1     9   962     0    12     0     5     9    11

     1     2     5     0   960     0     3     3     1     7

     2     2     1    17     1   857     2     0     6     4

     7     3     4     0    14    12   908     0    10     0

     1     3    13     3     5     1     0   986     1    15

     6     2     5     5     5     5     4     4   932     6

     1     4     2     6    30     2     0     6     5   953


train accuracy: 0.9892 test accuracy: 0.9619
```

3) 100 Hidden Layer nodes



Test -- Confusion Matrix

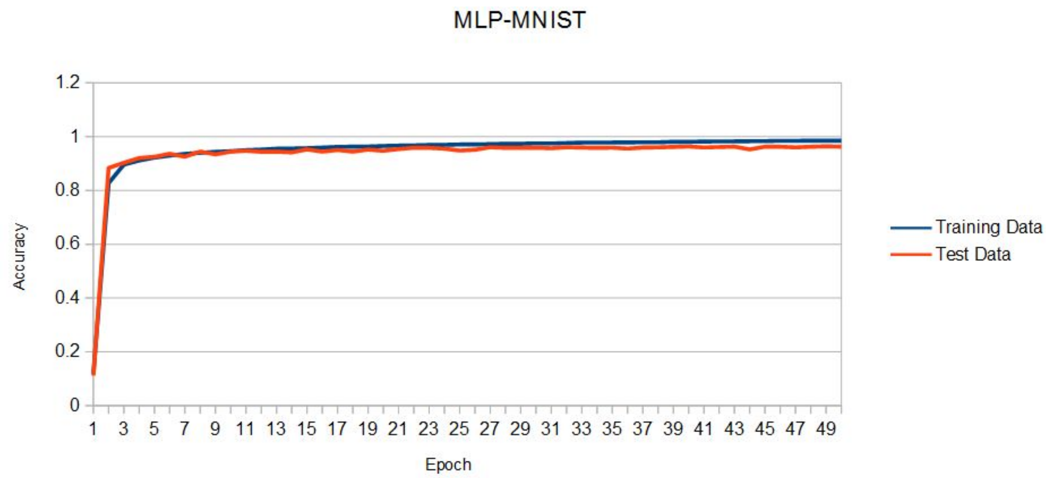
969	1	2	0	0	1	1	2	1	3
1	1120	4	1	1	0	1	2	4	1
3	4	997	4	3	0	0	16	5	0
0	1	8	968	1	17	0	9	5	1
0	0	5	0	943	1	3	10	1	19
2	0	1	9	1	867	5	4	1	2
13	2	4	0	5	13	916	1	2	2
0	3	9	0	2	2	1	1006	1	4
8	4	6	13	4	10	5	8	910	6
3	4	1	4	16	3	0	13	4	961

train accuracy: 0.99795 test accuracy: 0.9657

I programmed an MLP using C++. My optimization utilized stochastic gradient descent with momentum and weight decay. I trained the network with the MNIST data set(60k training set, 10k validation set) and randomized the order of the examples at each epoch. The first experiment consisted of three different tests, a hidden layer of 20 nodes, one with 50 nodes and one with 100 nodes, all trained for 50 epochs each. While all three converged with a less than 2% delta from each other, there were noticeable differences. It's clear from the graphs and confusion matrices that the more neurons resulted in a slightly higher accuracy(>97% for the 100 neuron model). However the less neurons the model had the quicker it converged to its maximum accuracy. Unlike the perceptron network all three showed obvious signs of overfitting quickly after achieving their maximum on the validation set by continuing to increase accuracy on the training set(nearly 100% with the 100 neuron model) while the validation set started to decrease and oscillate. All three obtained significantly higher metrics(~10%) than the single layer perceptron network. Even after the first epoch the MLP showed noticeably higher accuracies.

Experiment #2:

1) $\alpha = 0.0$

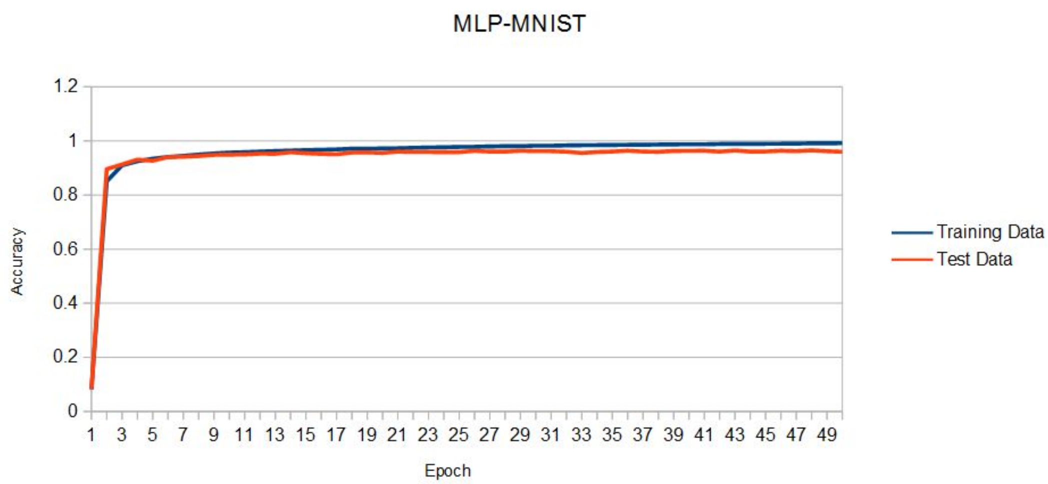


Test -- Confusion Matrix

968	1	2	1	1	2	2	1	1	1
0	1121	5	0	0	1	2	1	4	1
7	1	991	10	5	1	1	6	8	2
0	0	6	971	1	13	0	4	7	8
2	0	5	1	942	0	3	2	3	24
5	0	1	15	0	853	5	1	4	8
10	2	4	0	8	11	919	1	2	1
2	8	18	5	1	4	0	965	4	21
8	0	6	10	3	4	4	3	928	8
6	2	1	13	8	1	0	7	1	970

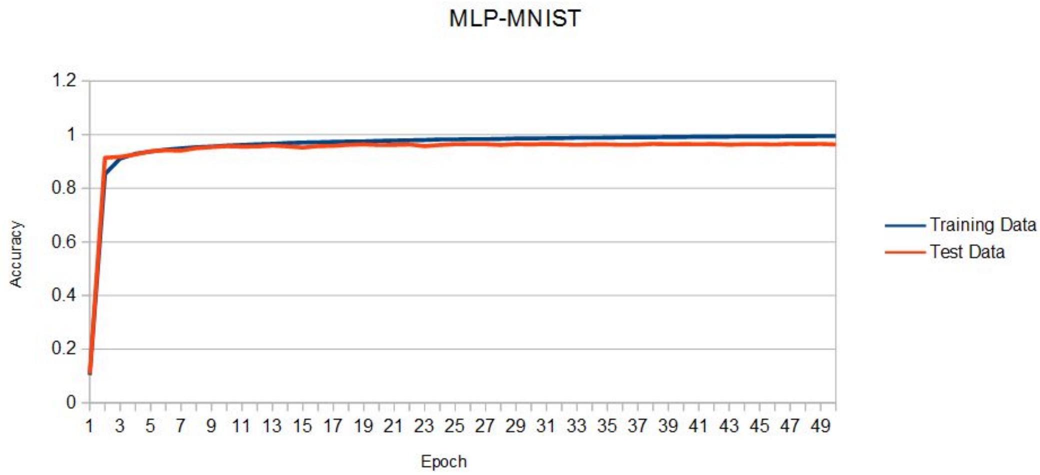
train accuracy: 0.985083 test accuracy: 0.9628

2) $\alpha = 0.25$



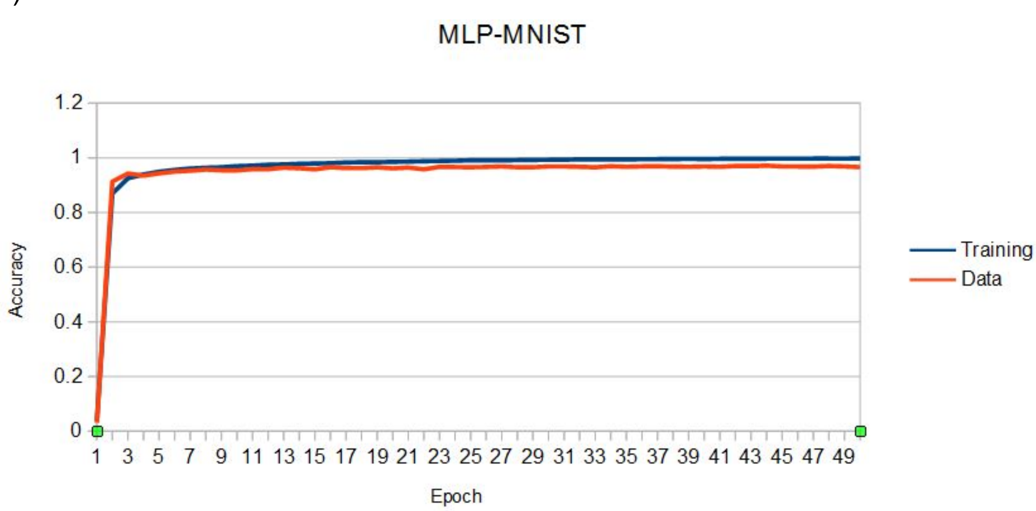

```
Test -- Confusion Matrix
   964     0     1     4     1     2     3     2     1     2

     0  1120     2     5     0     0     2     2     4     0

     5     1   967    29     4     1     2    10    12     1

     0     0     0   995     0     2     0     4     6     3

     1     0     3     1   952     0     9     4     0    12

     3     2     0    30     2   837     7     0     7     4

     6     3     1     1     6     5   929     1     6     0

     0     1     8    10     4     1     0   988     3    13

     4     0     3    21     8     5     3     3   924     3

     3     3     0    18    38     4     3    12     3   925


train accuracy: 0.99185 test accuracy: 0.9601
```

3) α = 0.5




```
Test -- Confusion Matrix
  960     0     1     1     1     7     6     0     3     1

    1  1121     1     1     0     2     1     1     7     0

    4     9   980     7     5     1     5    14     7     0

    1     0     3   963     0    24     1     7     4     7

    0     0     4     1   944     0     7     7     5    14

    4     0     0    10     0   862     8     1     5     2

    4     2     0     1    10     7   931     1     2     0

    1     3     5     2     3     1     1  1000     4     8

    4     1     6    12     4    11     3     4   922     7

    2     2     1     7    19    11     3    13     2   949


train accuracy: 0.99475 test accuracy: 0.9632
```

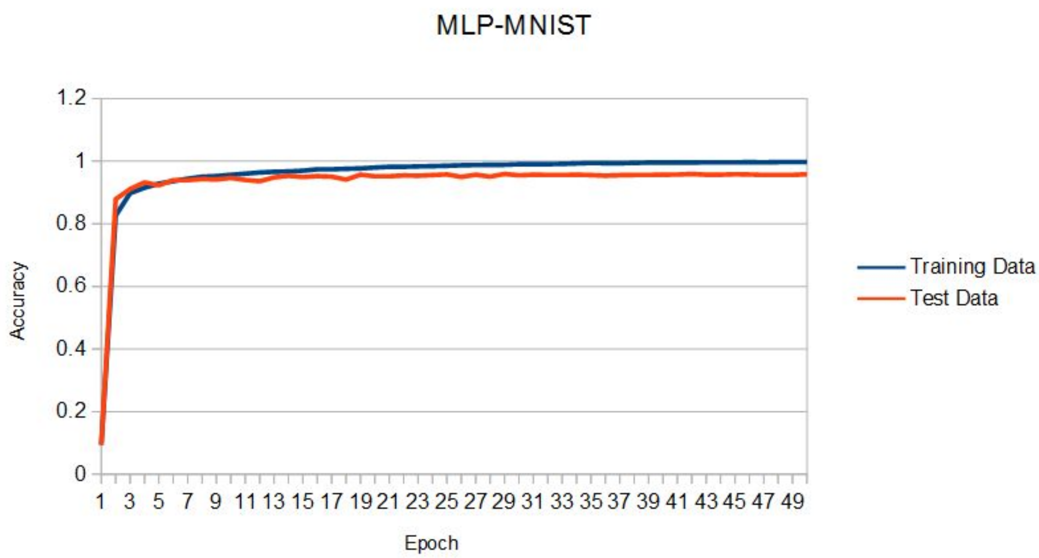
4) α = 0.9



The second experiment consisted of fixing the number of neurons in the hidden layer to 100 and varying the momentum rate ($\alpha = 0.0,\ \alpha = 0.25,\ \alpha = 0.5\ and\ \alpha = 0.9$). With a larger momentum the model was able to achieve a higher accuracy. While it took longer to converge, the models with larger momentum approached their maximums quicker than those with a lower rate. Overfitting was less noticeable on those with lower momentum rates, however all three models showed clear signs of it as their training data accuracies continued to increase while the validation accuracies decreased and began to oscillate.

Experiment #3:

1)Network 1(30k training set)

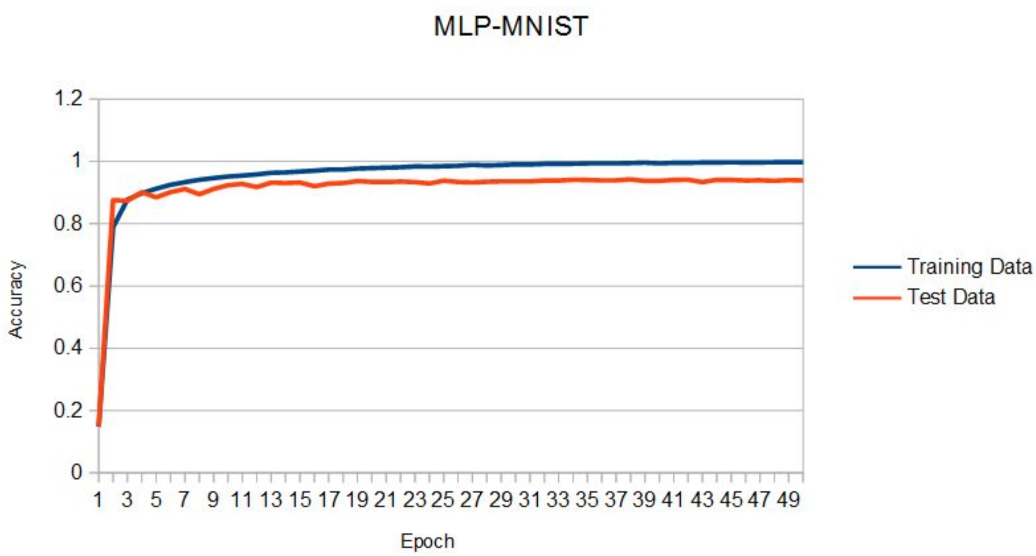


Test -- Confusion Matrix

955	0	4	0	1	6	2	4	7	1
0	1114	11	1	0	0	2	2	5	0
4	1	994	5	3	1	1	6	17	0
0	0	9	968	0	10	1	3	17	2
2	0	1	2	951	2	2	3	3	16
4	2	1	15	3	839	8	1	17	2
5	5	8	1	14	7	904	2	12	0
0	6	17	3	4	0	0	989	3	6
4	1	5	3	3	9	3	2	938	6
2	5	4	10	19	9	0	9	16	935

train accuracy: 0.9973 test accuracy: 0.9587

2)Network 2(15k training set)



Test -- Confusion Matrix

938	0	3	0	1	10	17	2	5	4
0	1115	4	0	0	1	4	1	10	0
4	6	962	7	2	2	12	10	25	2
1	2	19	929	0	15	2	5	28	9
1	0	4	0	919	0	16	3	6	33
6	1	3	21	4	811	20	2	19	5
3	5	3	1	2	4	929	5	5	1
1	11	16	7	5	2	1	956	2	27
3	0	7	13	4	8	13	5	908	13
3	7	1	14	27	5	2	13	13	924

train accuracy: 0.997533 test accuracy: 0.9391

For the final experiment the dataset was reduced to two smaller training subsets(30k and 15k). While the model seemed to converge to its maximum quicker, there was much more oscillation around the validation set accuracies. There was also significantly more obvious signs of overfitting than when using the full training dataset(60k). The accuracy of the training set climbed to nearly 100% while the validation set oscillated on a significantly lower value than when training with a larger amount of data. It's clear that the more variety of data, the better the model generalizes, while with a smaller set it's quicker to start learning to memorize the training data.