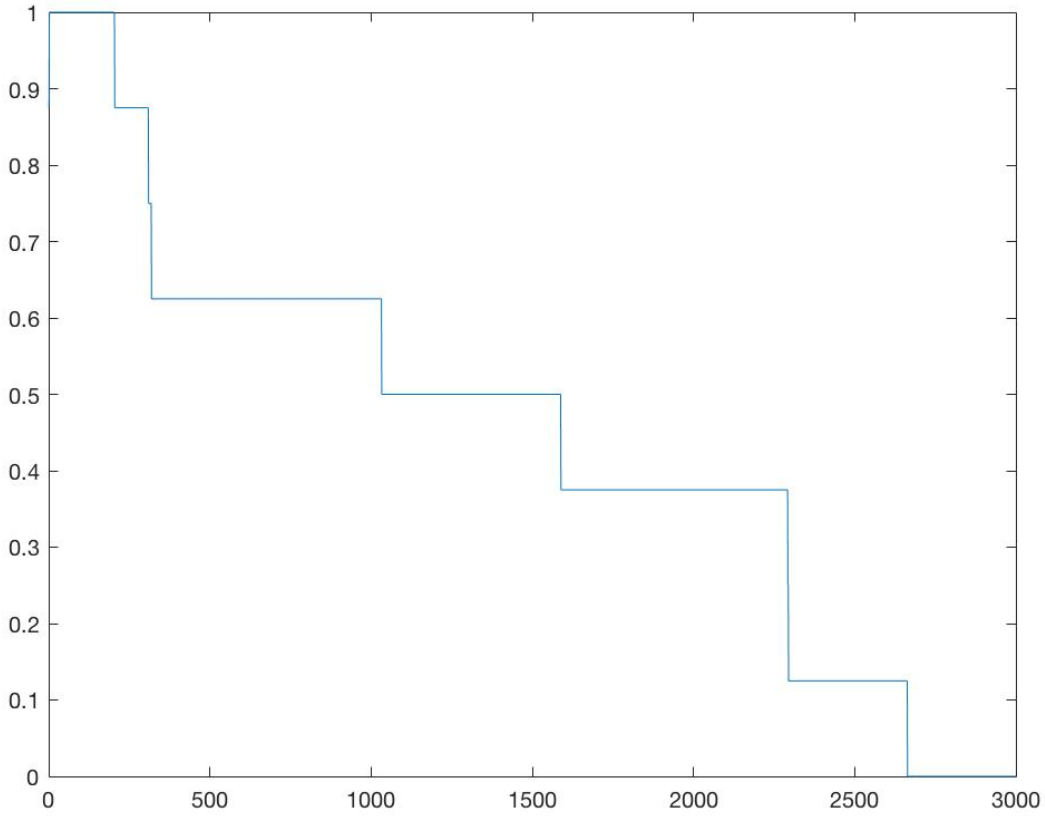
Student: Kien Le
Tufts ID: 1234053

Comp135: Machine Learning
Programming Project 4

For the 838 dataset
Training set error vs iterations



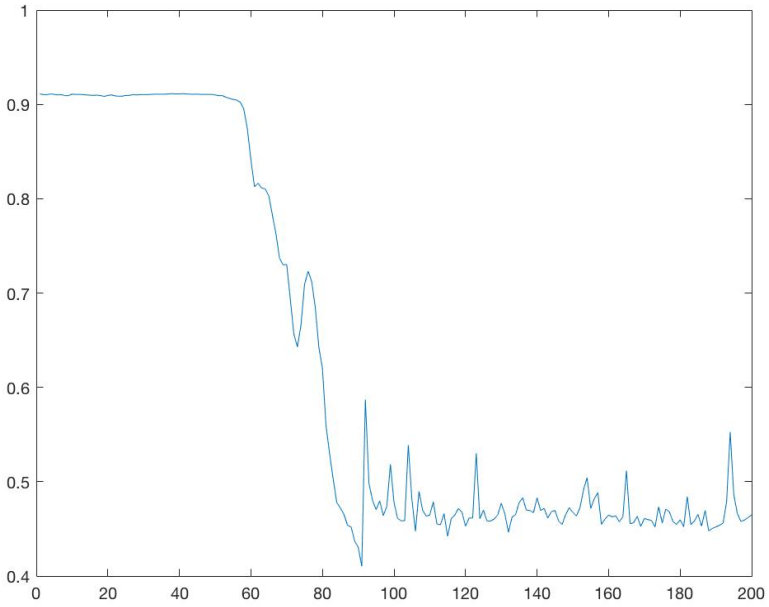
From the plot, we see that we achieve zero error on the dataset when the iteration is large enough. The learning curve is like a “step” function since we only have 8 examples on the dataset, then the training error drop after decreasing each one error. However, to get zeros error, we need more than 2600 iterations. That means the network is getting better very slowly.

The representation of the hidden unit representation after the last iteration

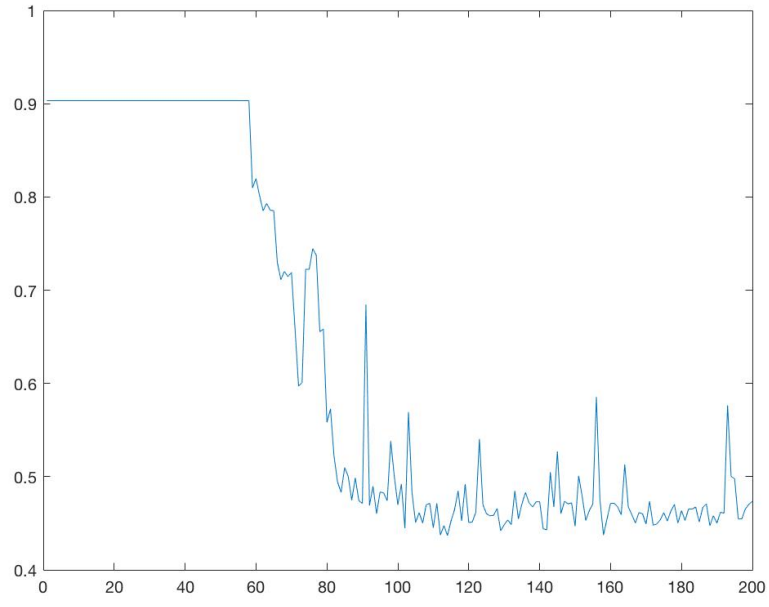
```
0.6274    0.9617    0.0398
0.9289    0.0311    0.9452
0.9795    0.3501    0.1057
0.4702    0.2985    0.4060
0.0897    0.3862    0.9764
0.3444    0.4046    0.3467
0.3688    0.3491    0.3761
0.0411    0.9756    0.6320
```

The table above shows the hidden representation of the number units from 1 to 8 after all the iterations. From that, we see that the neural network has done really well in finding a binary representation for almost of 8 (there are only 2 missing). The problem here might be due to the initializing since it might be far away from the true weights to get the binary representation. And it probably takes more than 3000 iterations to achieve the correct representation.

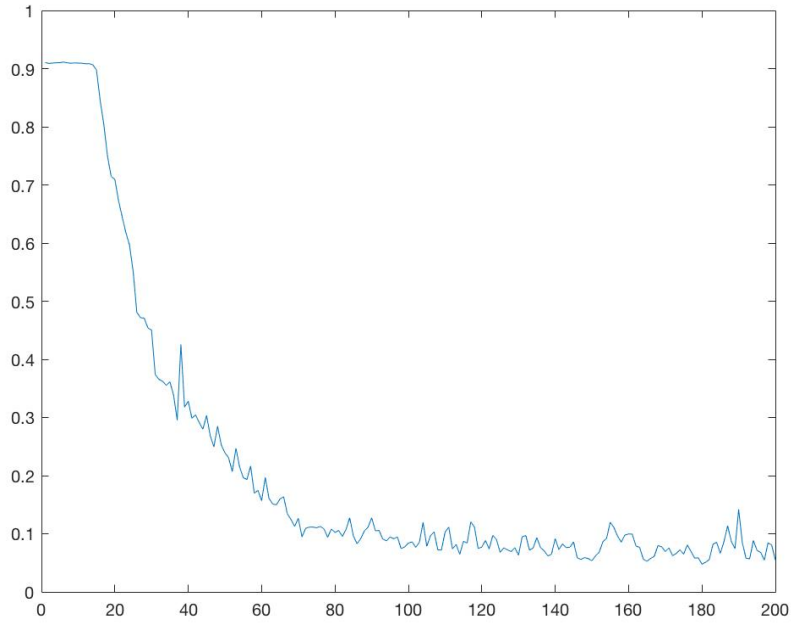
For the opt digit dataset
(1) d=3; w=5
Training set error vs iterations



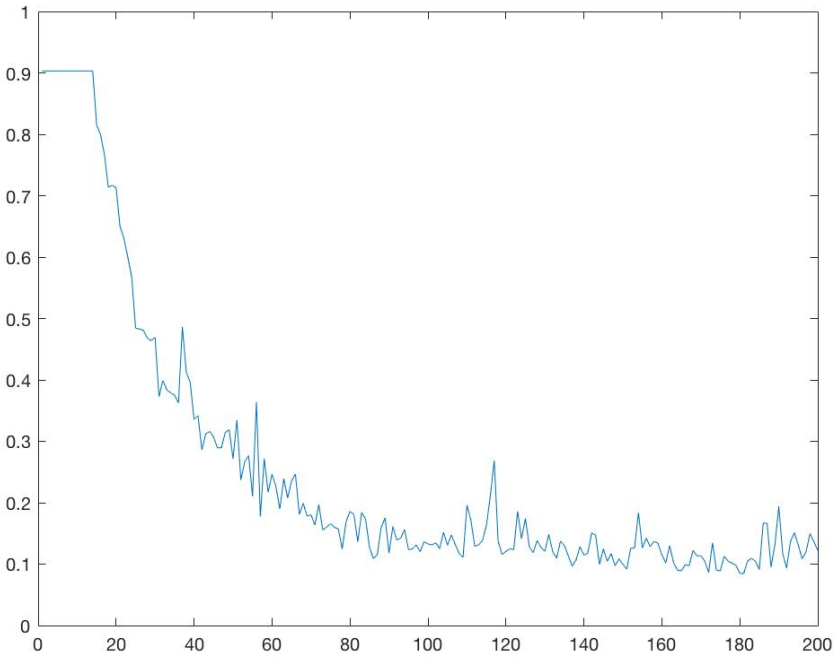
Testing set error vs iterations



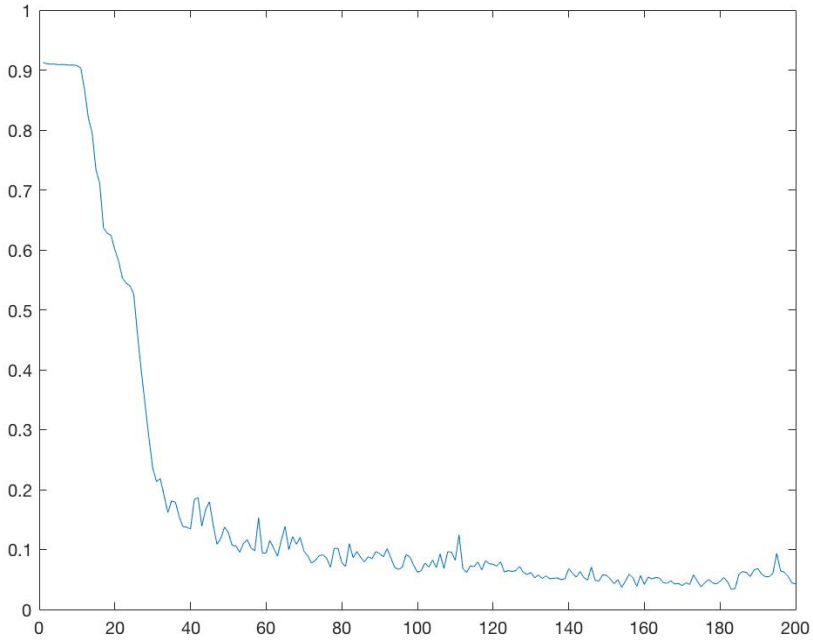
d=3, w=10
Training set error vs iterations



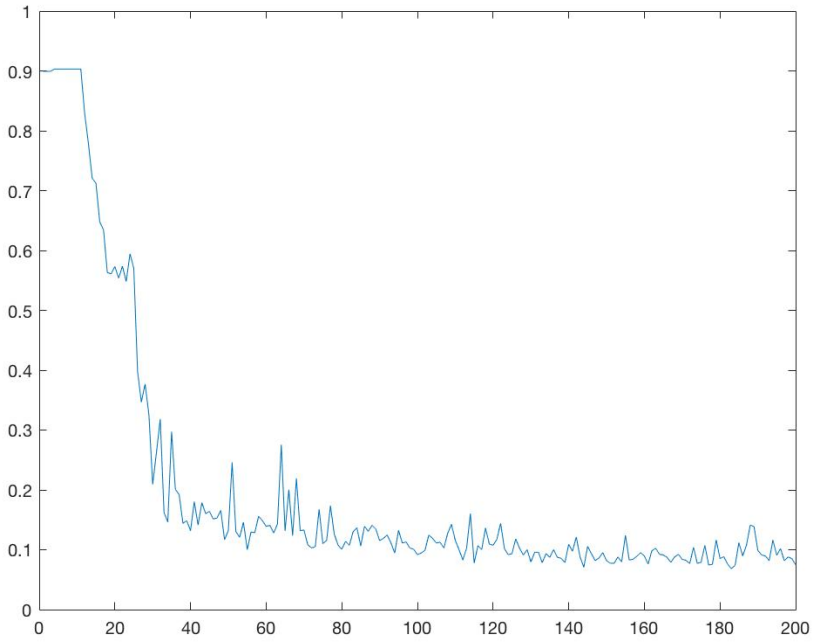
Testing set error vs iterations



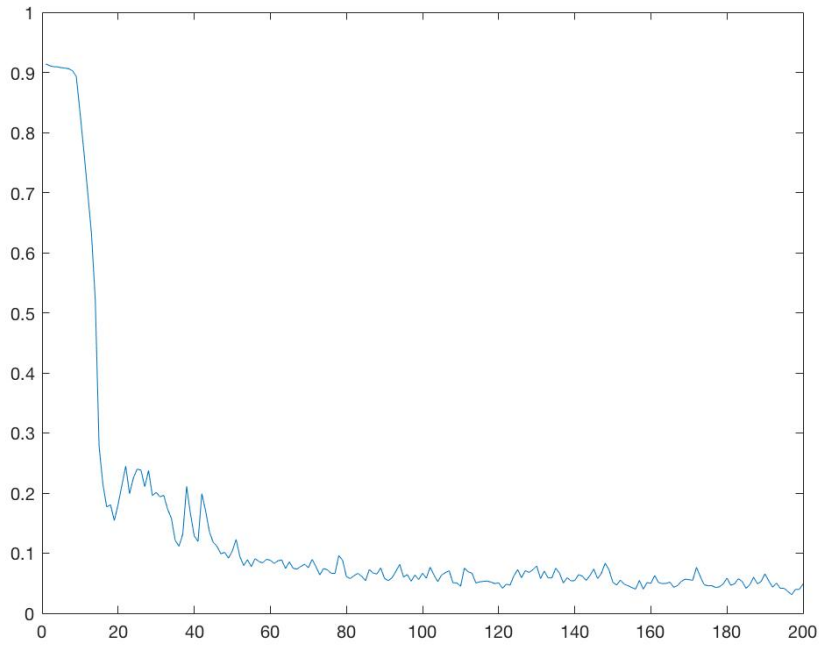
d=3, w=15
Training set error vs iterations



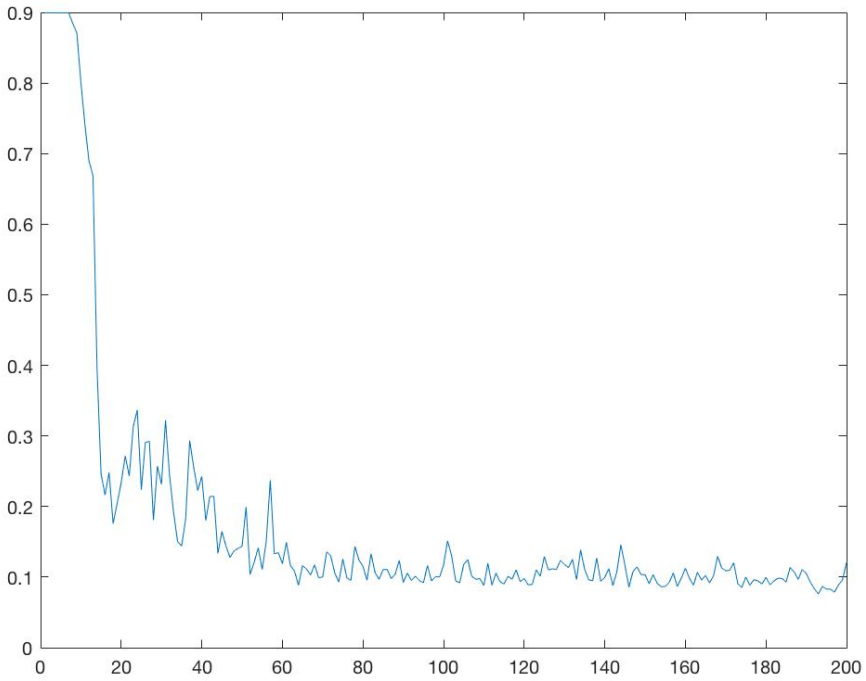
Testing set error vs iterations



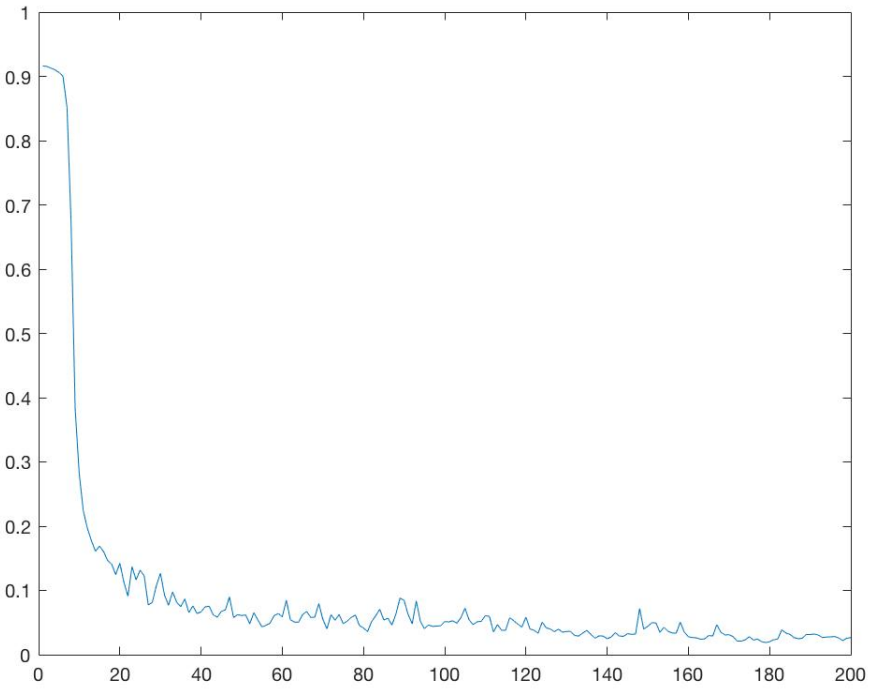
d=3, w=20
Training set error vs iterations



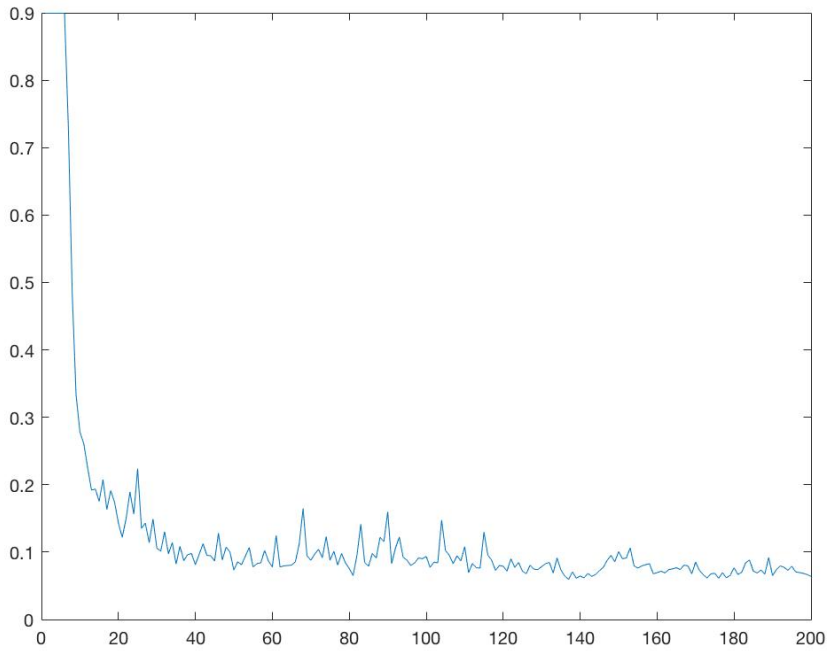
Testing set error vs iterations



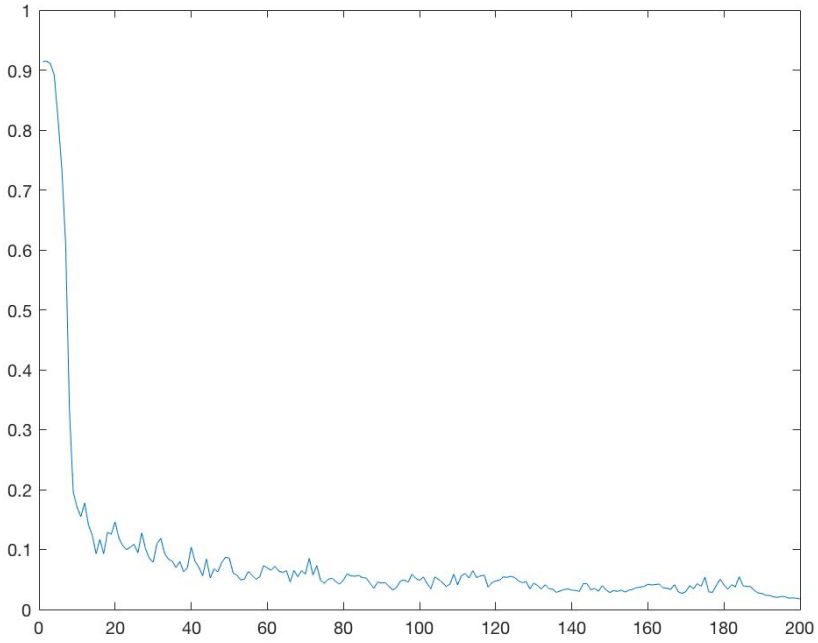
d=3, w=30
Training set error vs iterations



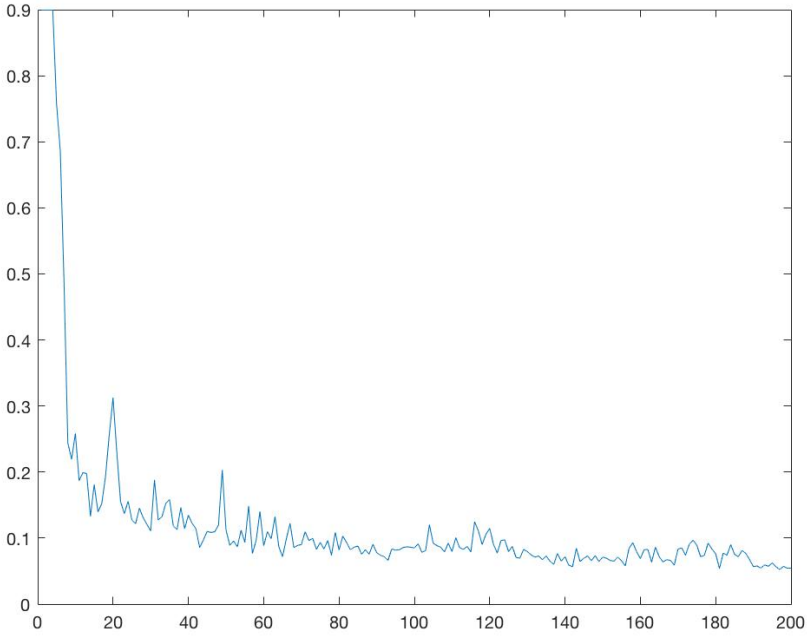
Testing set error vs iterations



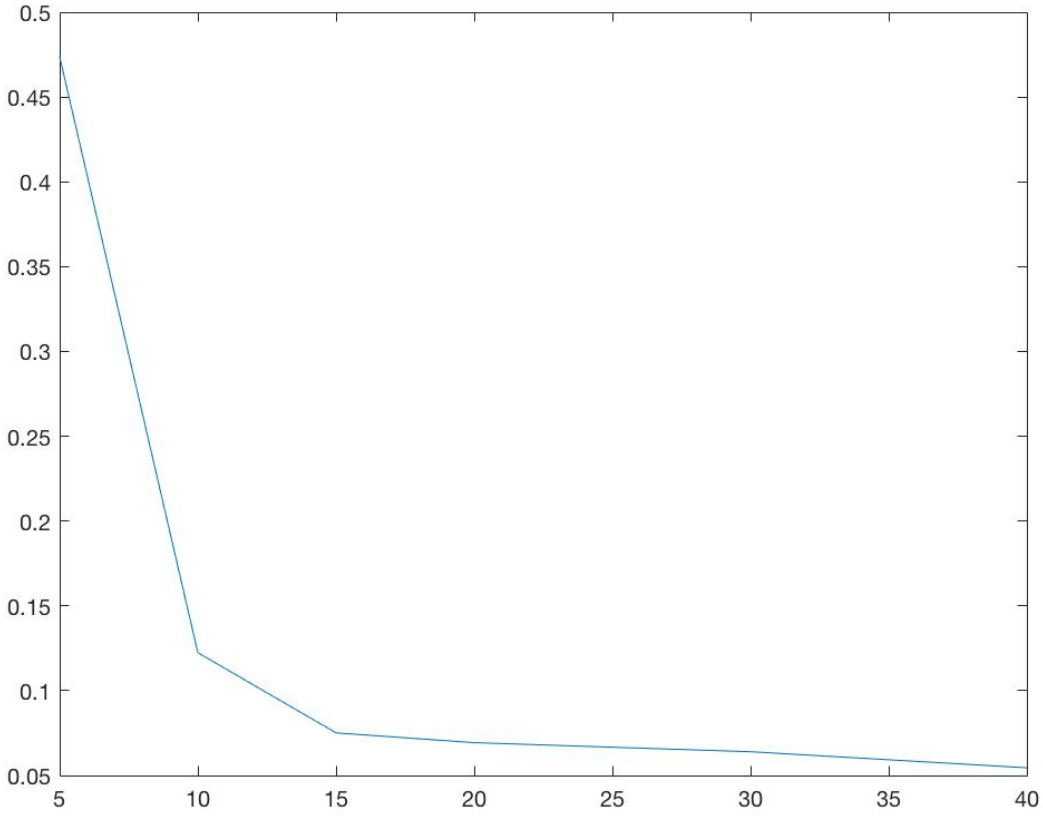
d=3, w=40
Training set error vs iterations



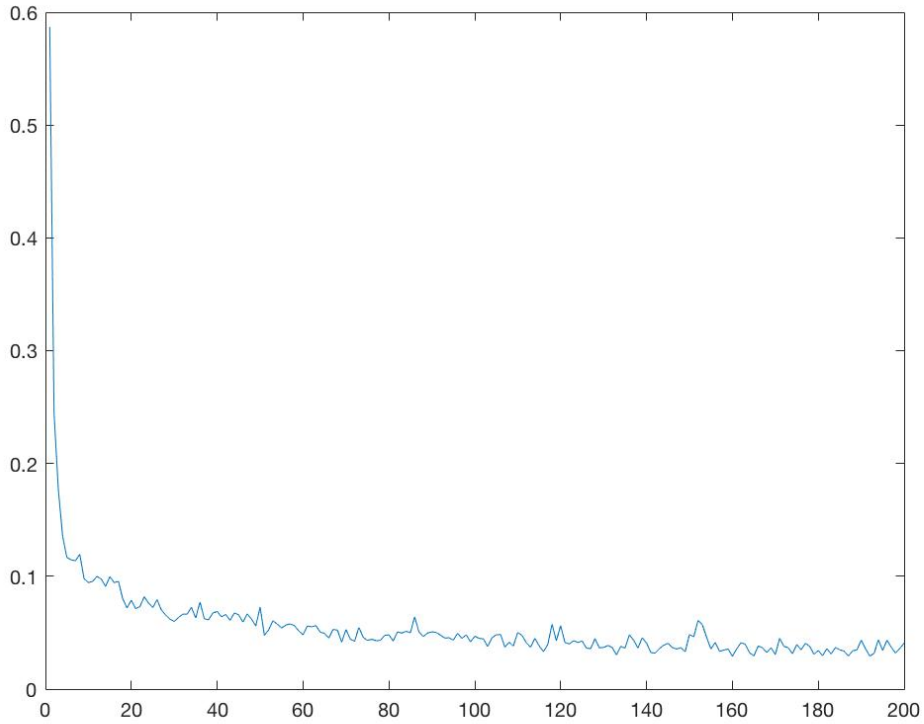
Testing set error vs iterations



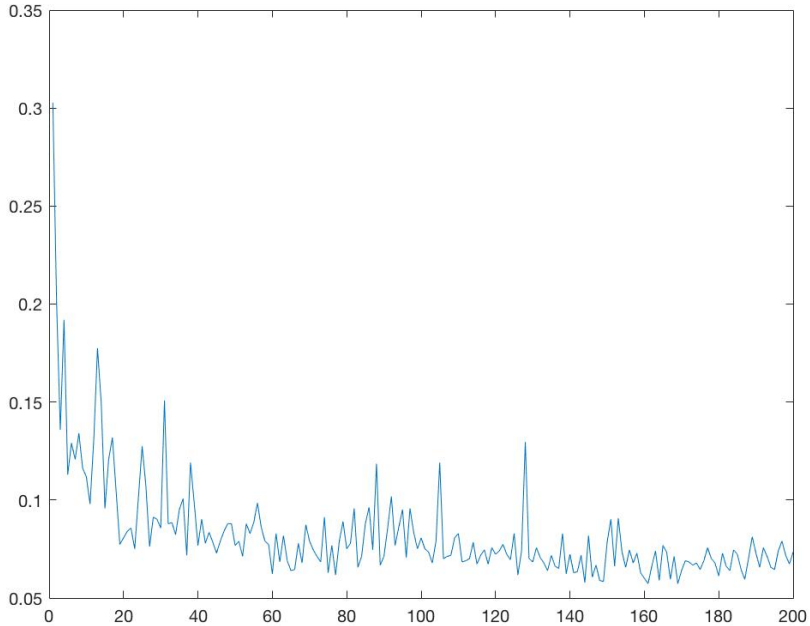
Test error after the last iteration vs w



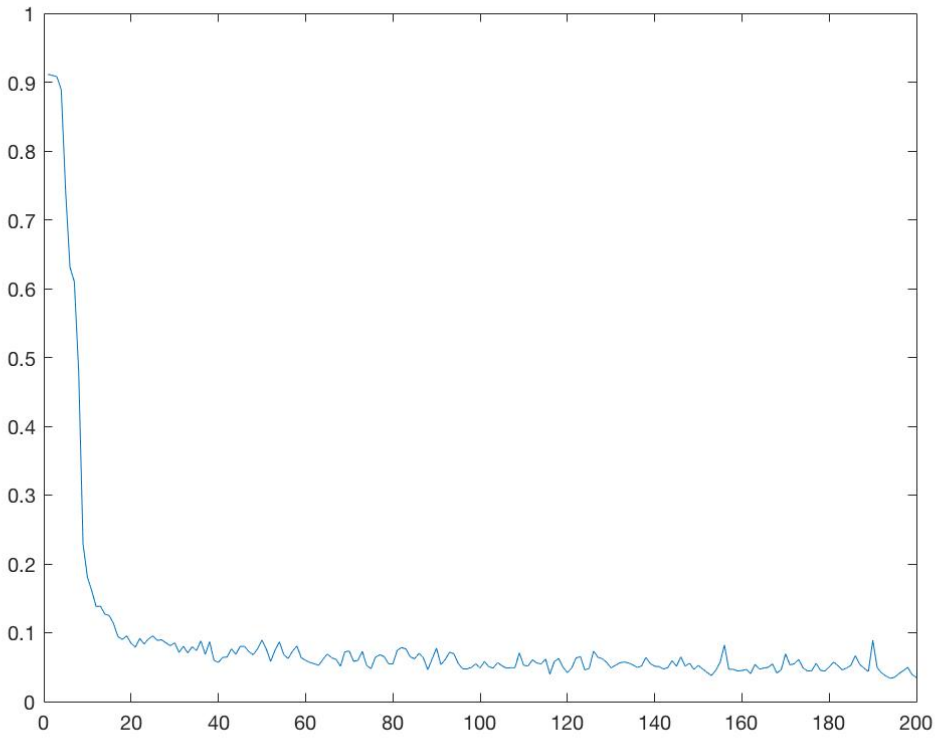
(2) w=10, d=1
Training set error vs iterations



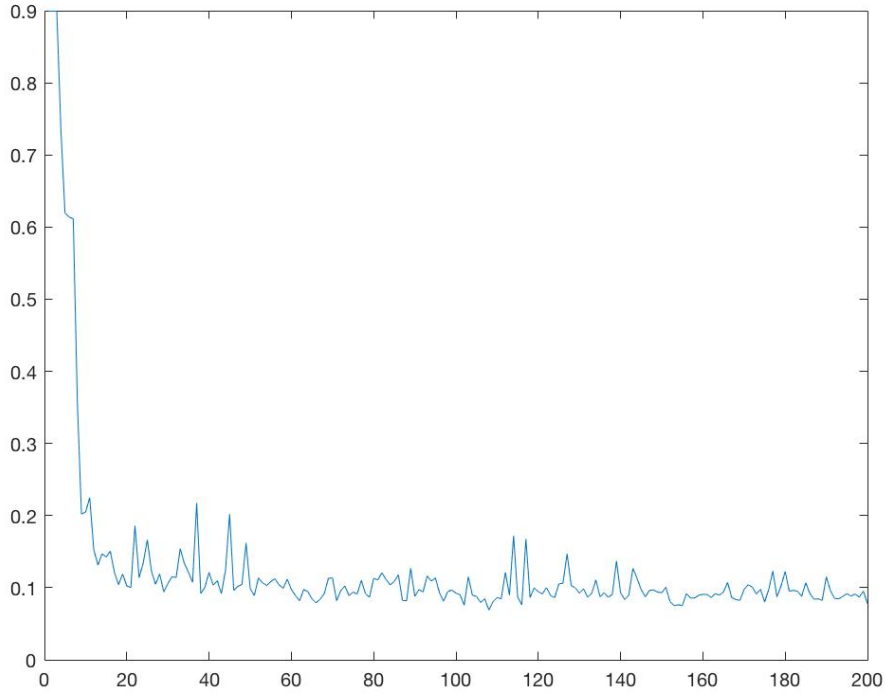
Testing set error vs iterations



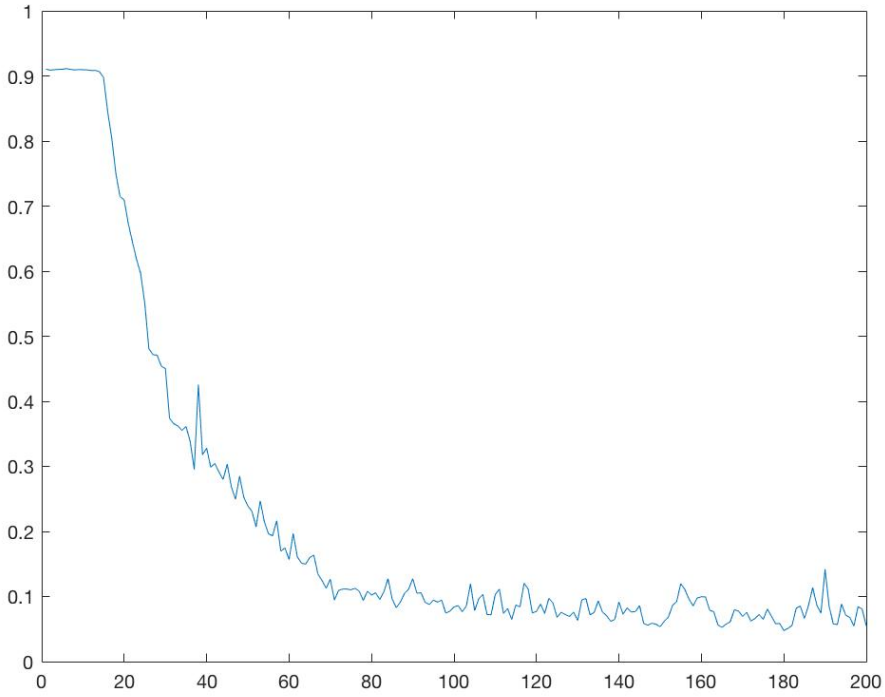
w=10, d=2
Training set error vs iterations



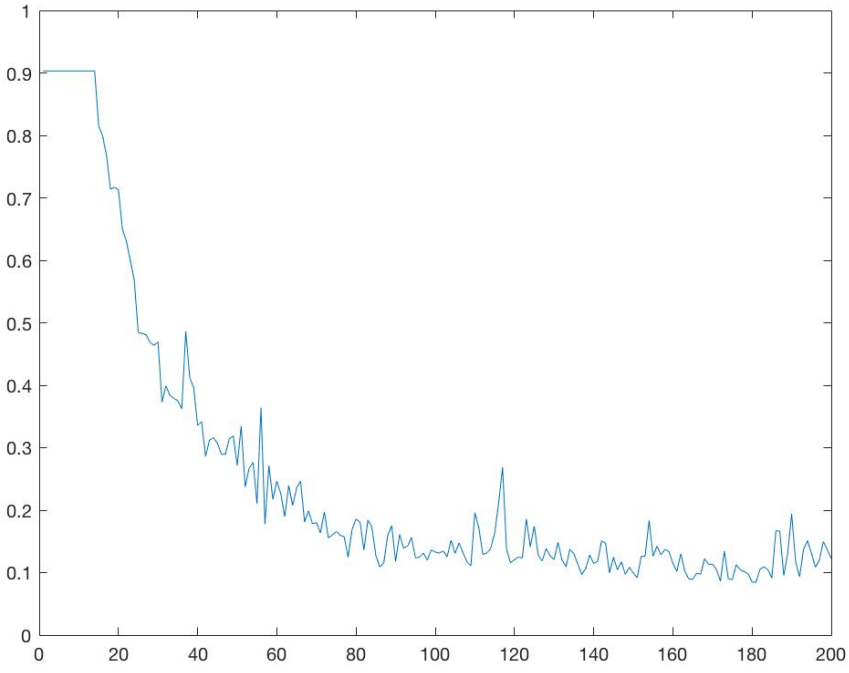
Testing set error vs iterations



w=10; d=3
Training set error vs iterations

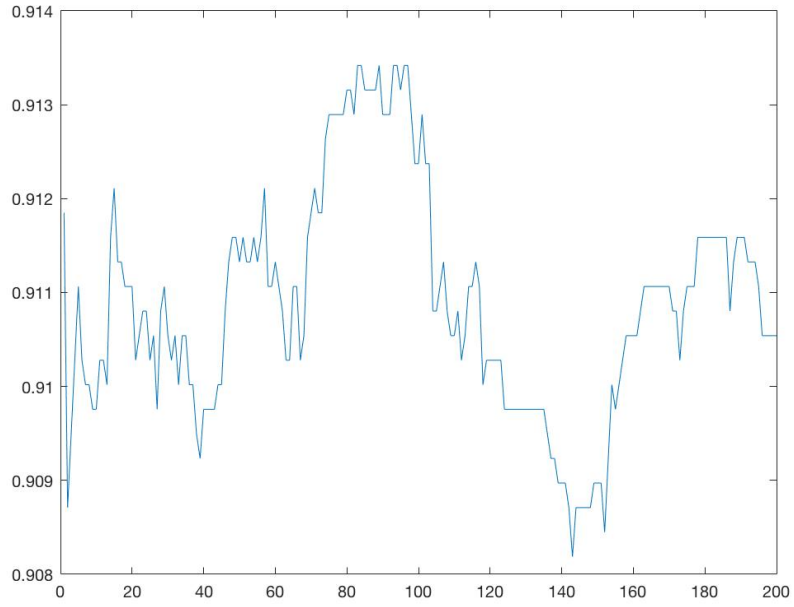


Testing set error vs iterations

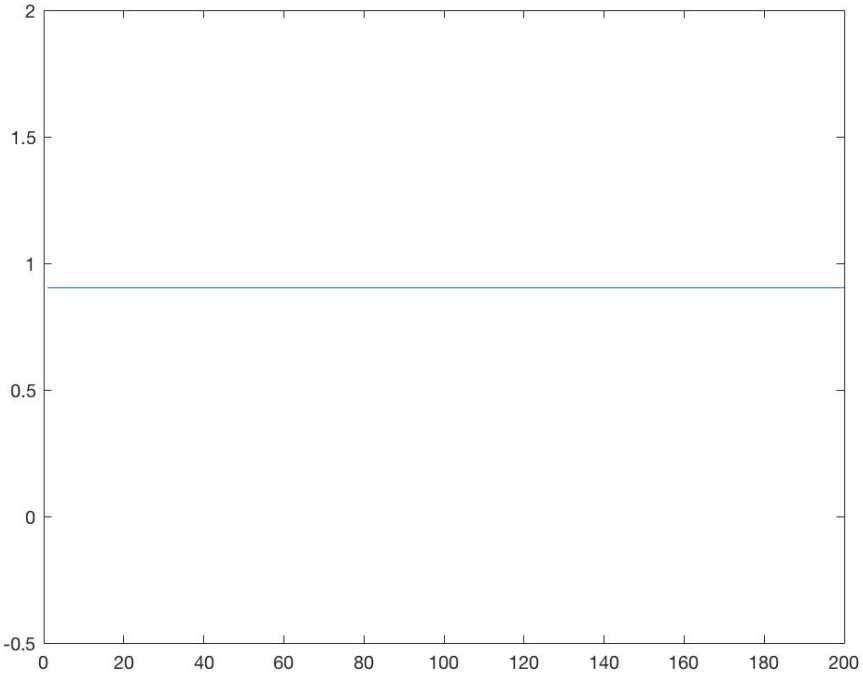


w=10,d=4

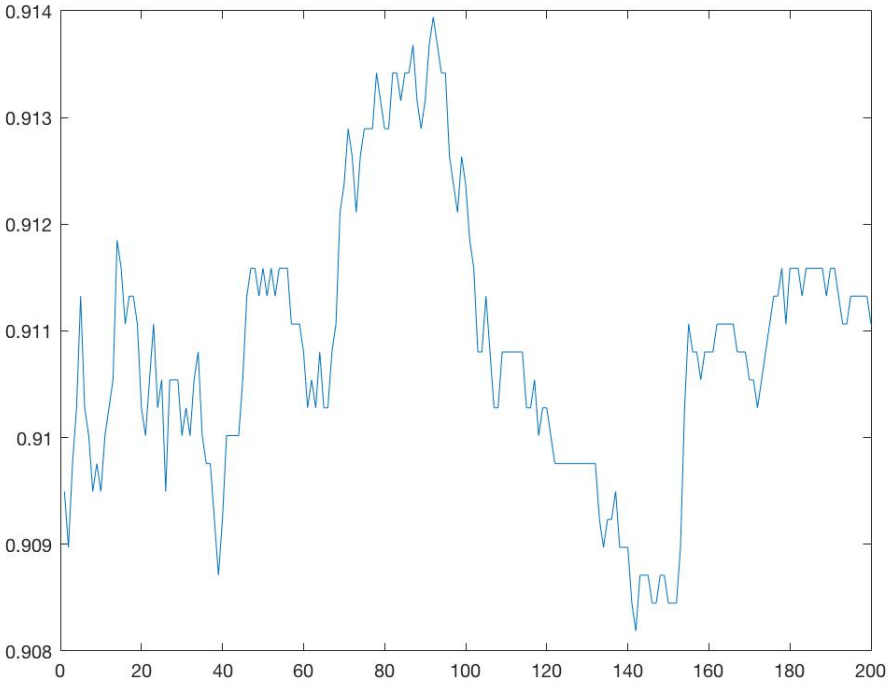
Training set error vs iterations



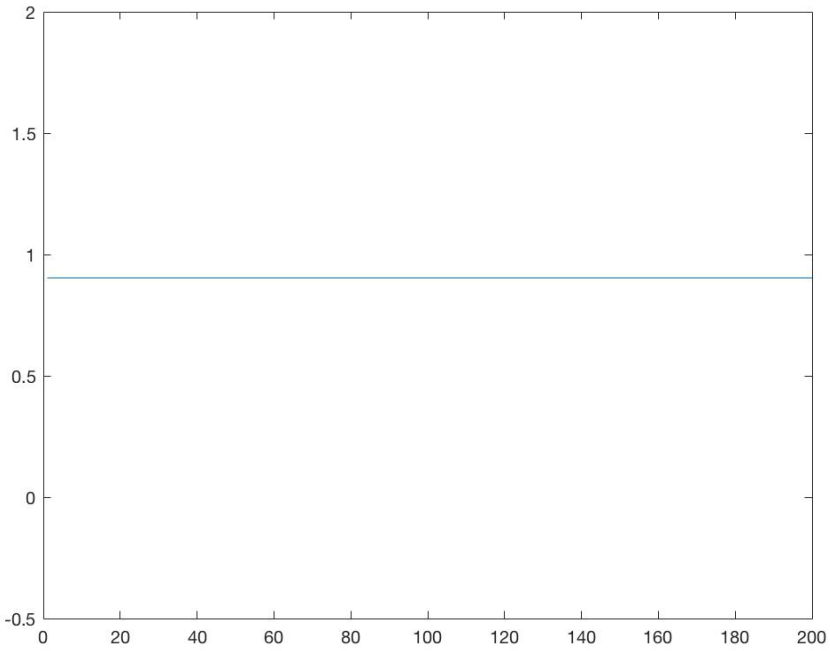
Testing set error vs iterations



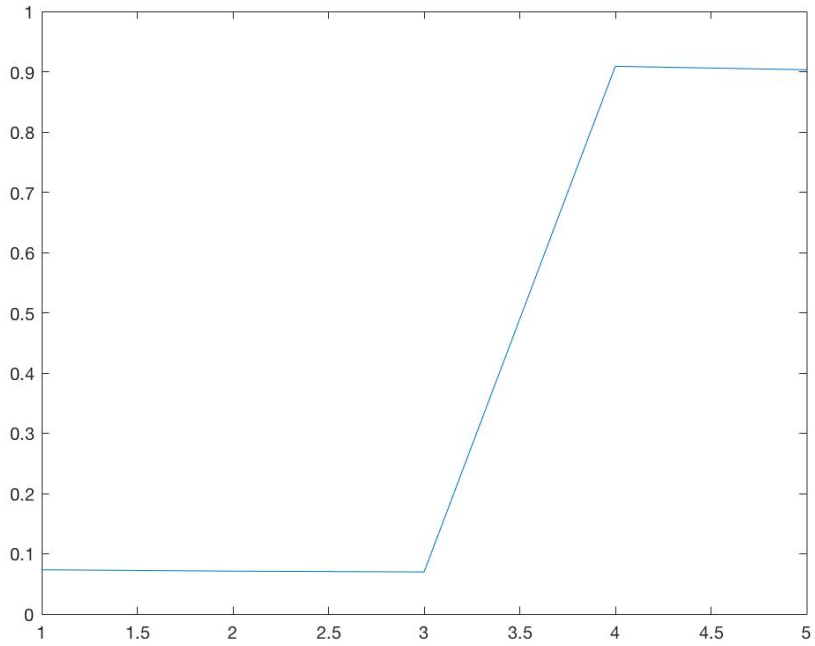
w=10; d=5
Training set error vs iterations



Testing set error vs iterations



Test error after the last iteration vs d



From all these plot above, for most neural networks of different widths and depths, as the number of iterations increases, the error rate decreases. Actually, in the first few iterations, the error rate drops so quickly. This is because at the starting points, we are quite far away from the true label, so every iteration make a great move toward the true weight. When we get closer to the true weight, the change in delta becomes smaller, then we only make a small improvement each iteration.
In addition, we see that the training error is usually lower than the test error since we are training on the train dataset and trying to generalize the network to apply to the test set. So that the accuracy on the training set is better makes sense. Also, the learning curve fluctuates a lot but the overall trend is still going down. This is because we are using stochastic gradient decent and this algorithm updates on every training examples. So at a particular iteration, it might go up or down depending on the current weight.
When the values of depth are 4 or 5, the accuracies are really bad. As depth goes from 1 to 3, the error rate decrease so much, even below 0.1. These are “deep” enough networks. With values of depth 4 or 5, the error rates are around 0.9 because of the problem of vanishing gradient. That means when we back propagate to the lower layers, the amount of changes becomes smaller, then it affects the weights less. So the outputs do not change much, then the error rates also do not decrease much.
Among all the experiment I tested, the values (w,d) of (40,3) is the best one with the error rate of only 0.05454.