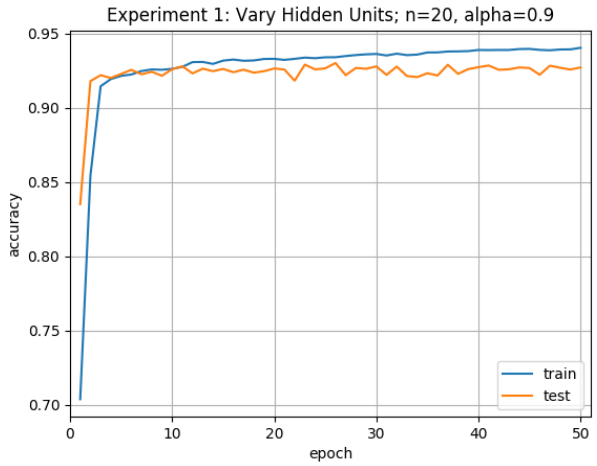
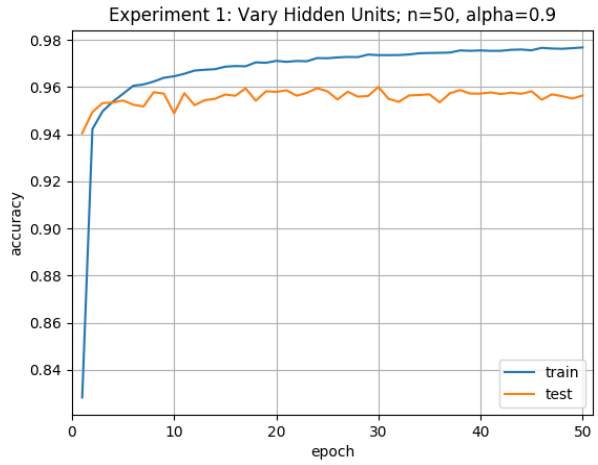
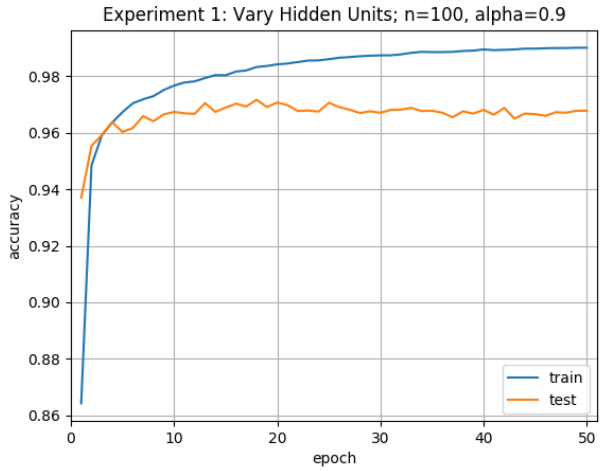
Carlos Miguel Sayao
CS445

Programming Assignment 1



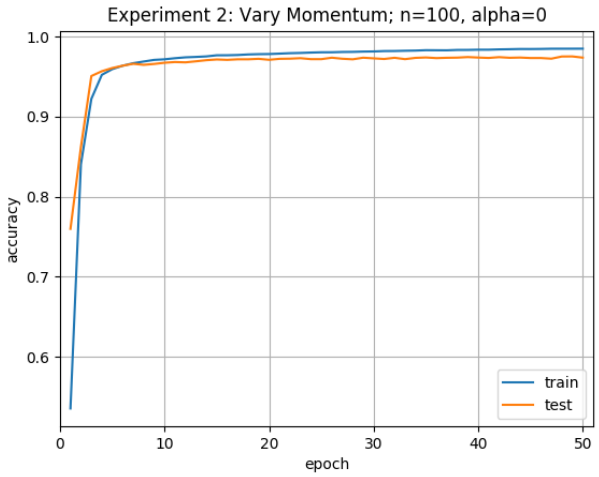




Experiment 1: Vary number of hidden units. n = 20, 50, 100

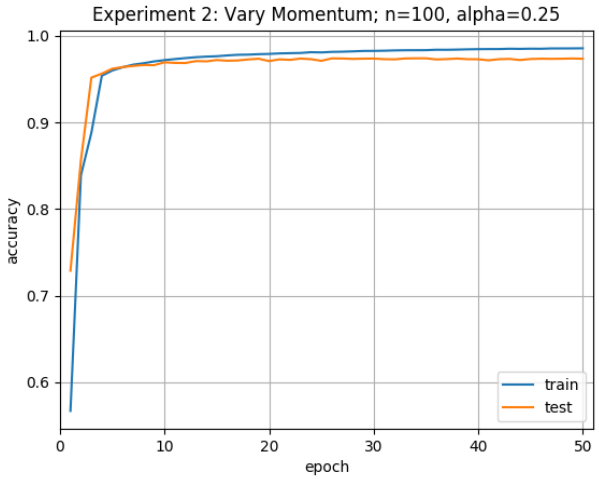
1. As n increases, our training and testing accuracy gets better. This makees sense. Given more hidden units, the output units have a more diverse set of hidden neurons to choose from, minimizing nonlinear separability.
2. After about 25 epochs, our training accuracy doesn't get much better. (Only by a little.) However, the training accuracy continues to get better. Convergence appears at about 5 epochs. The more hidden units, the sooner they diverge.
3. The way testing accuracy doesn't get any better while training accuracy does could be a sign of overfit. Another could be the deviance of test accuracy.
4. Training accuracy in the perceptron never got above 90%. I wasn't able to finish the testing for the perceptron, but I want to see the testing accuracy. The MLP already proves to be more accurate.

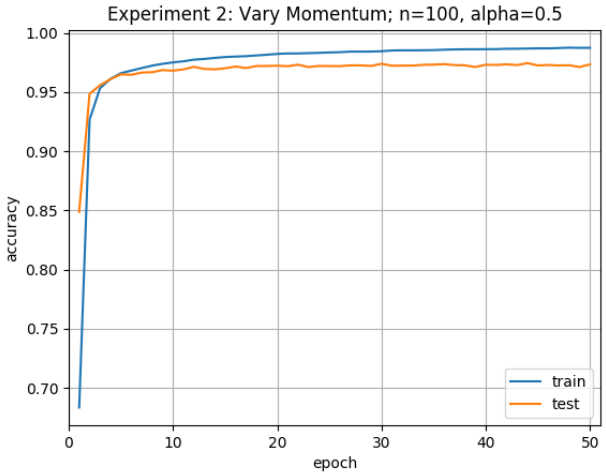
Experiment 2: Vary the momentum value
Alpha = 0, 0.25, 0.5



1. It doesn't seem like a varied momentum changed the experiment much. But looking closer, we can see that learning rates were quicker at n=0.5. I accidentally removed the part of my program that saved the y's at each epoch, but I looked through the output and found the following for the last 5 epochs (momentum, training accuracy, test accuracy):

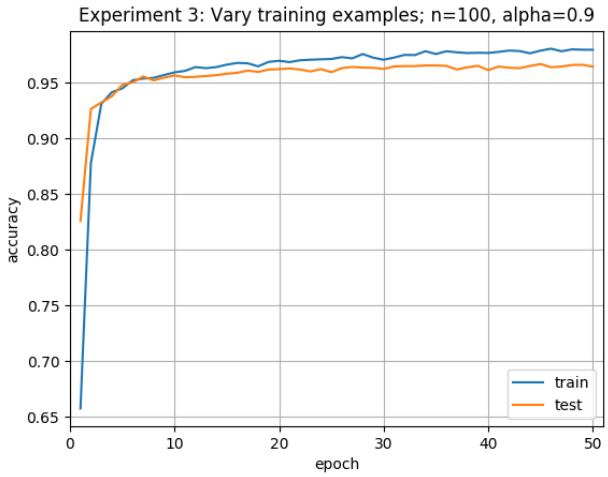
```
0
98.4667      97.33
98.4917      97.25
98.4933      97.52
97.495       97.53
98.5         97.38

.25
98.8233      97.37
98.5617      97.36
98.5633      97.37
97.5667      97.4
98.5783      97.37

.5
98.68597.28
98.71        97.24
98.7433      97.26
98.7267      97.11
98.7         97.33
```

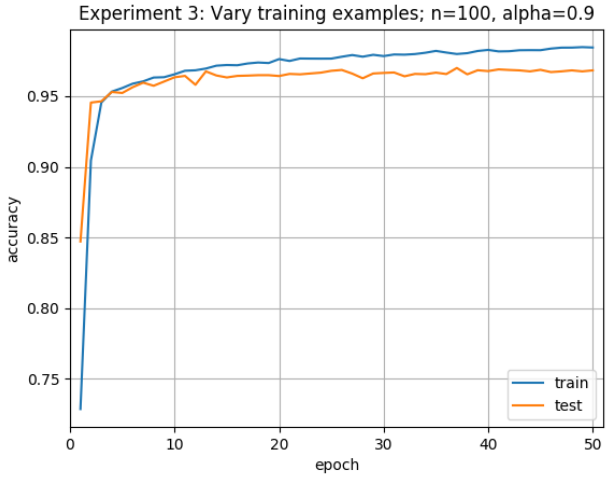


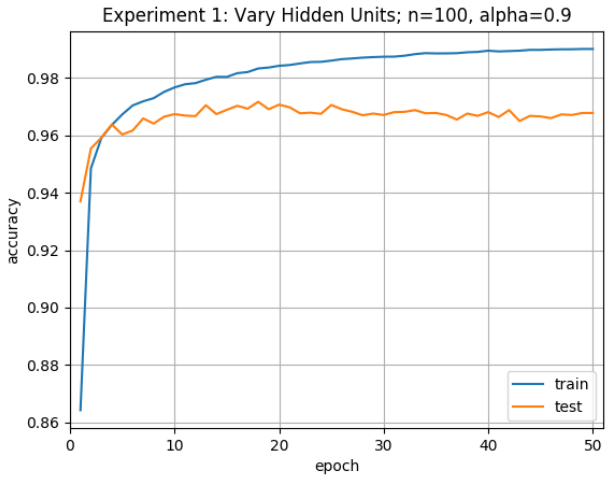


2. After around 25 epochs, our test accuracy stays steady. Again, without a thourough set of data I can't say for certain, but it looks like the testing begins to converge sooner and more accurately than the third experiment of experiment 1.

3. It looks like alpha=0.5 might have the most overfit by looking at the deviations of test accuracy at the last 5 epochs.







Experiment 3: Vary the number of training examples. N=15000, 30000

```
15000
98.0667   96.4
97.8333   96.46
98.0      96.6
97.9733   96.62
97.9667   96.46
~97.968   ~96.508

30000
98.3733   96.71
98.44     96.76
97.33     96.83
98.4967   96.76
98.45     96.83
~98.218   ~96.778
```

1. In the last 5 epochs, training accuracy was about 0.25% better in the 30,000 sample experiment. Looking back at experiment 1, we can see that---all other values equal---60,000 samples gave the best training and testing accuracy.
2. Testing converges at about 3 or 4 epochs. The more samples to train on, the sooner the diverge.
3. From experiment 1, it looks like 60,000 samples gave the most overfit. Average test accuracy seems to go down after about 20 epochs there, but it is still the best test and train accuracy. For 30,000 samples, the last 10 epochs look like overfit because test accuracy seems unincreasing. The last 20 epochs of 15,000 samples look like overfit as well, where training accuracy rises and test accuracy stays unincreasing. Overall, where the two measures diverge can be called overfit.