Problem 3

Table 1: Training ANNs with 1 Hidden Layer on MNIST; mini batch = 10, num epochs=50

Eta/ HLS	10	20	30	40	50
$\eta = 2$	9031	9346	9443	9480	8566
$\eta = 1.5$	9176	9290	9385	9440	9447
$\eta = 1.0$	9019	9300	9374	9376	9351
$\eta = 0.5$	9106	9185	9251	9299	9298
$\eta = 0.25$	8869	8999	8280	9116	9284

Table 2: Training ANNs with 2 Hidden Layers on MNIST; 1st HLS=10; mini batch = 10, num epochs=50t

Eta/ HLS	10	20	30	40	50
$\eta = 2$	9157	9203	9212	9195	9058
$\eta = 1.5$	9165	9070	9044	9155	9152
$\eta = 1.0$	9039	9124	9173	9176	9121
$\eta = 0.5$	9047	9094	9114	9078	9070
$\eta = 0.25$	8895	8903	8937	8934	8065

Table 3: Training ANNs with 2 Hidden Layers on MNIST; 1st HLS=20; mini batch = 10, num epochs=50

Eta/ HLS	10	20	30	40	50
$\eta = 2$	9324	9410	9357	9403	9415
$\eta = 1.5$	9347	9329	9338	9372	9360
$\eta = 1.0$	9300	9374	9268	9315	9332
$\eta = 0.5$	9145	9191	9234	9286	9258
$\eta = 0.25$	9048	9084	9093	9082	9116

Table 4: Training ANNs with 2 Hidden Layers on MNIST; 1st HLS=30; mini batch = 10, num epochs=50

Eta/ HLS	10	20	30	40	50
$\eta = 2$	9305	9455	9415	9470	9463
$\eta = 1.5$	9343	8519	9401	9402	9434
$\eta = 1.0$	9309	9338	9398	9416	9405
$\eta = 0.5$	9187	9251	9331	9295	9223
$\eta = 0.25$	9011	9150	9169	9159	9151

Table 5: Training ANNs with 2 Hidden Layers on MNIST; 1st HLS=40; mini batch = 10, num epochs=50

Eta/ HLS	10	20	30	40	50
$\eta = 2$	9404	9423	9458	9488	9468
$\eta = 1.5$	9382	9449	9434	9462	9456

$\eta = 1.0$	9350	9370	9464	9400	9431
$\eta = 0.5$	9280	9341	9358	9335	8462
$\eta = 0.25$	9038	9121	9165	9194	9182

Table 6: Training ANNs with 2 Hidden Layers on MNIST; 1st HLS=50; mini batch = 10, num epochs=50

Eta/ HLS	10	20	30	40	50
$\eta = 2$	9378	9456	9481	9509	9481
$\eta = 1.5$	9382	9453	9459	9459	9474
$\eta = 1.0$	9394	9376	9399	9430	9449
$\eta = 0.5$	9281	9292	9311	9335	9370
$\eta = 0.25$	9051	9207	9191	9225	9214

Observation from Problem 3

From all the different variation of setups, none of them were able to pass 95% accuracy (9500). I think this is interesting because there seems to be a limit to what the artificial networks can do. There also seems to be a weak trend that when the learning rate is decreased, the fitness decreases also. This surprises me because I would expect a smaller step to cause the system to approach better to a solution and not overshoot it. But the data also shows the any of the learning rates are shown to be possible of degeneration. However, overall, the fitness increases with the number of epochs

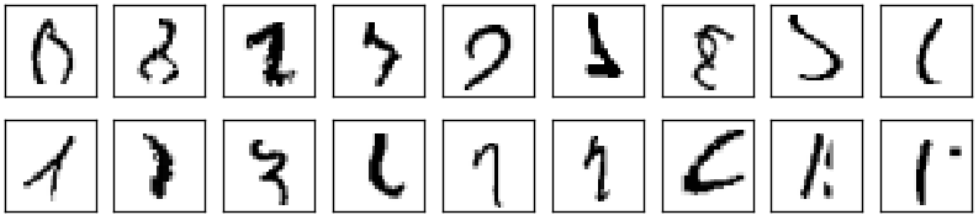
Problem 4

Table Name	Individual Fitness (out of 10000)
net_784x10x30x10_mbs-10_eta-50.pkl	9249
net_784x10x50x60x30x10_mbs-10_eta-50.pkl	9311
net_784x50x60x10_mbs-20_eta-30.pkl	9340
net_784x50x60x30x10_mbs-10_eta-50.pkl	9503
net_784x50x60x30x10_mbs-20_eta-30.pkl	9358
net_784x60x30x20x40x20x10_mbs-10_eta-50.pkl	9461

Total Fitness (Net Ensemble)	9537
Average Individual Score	9370.3

Observations

First off, I expect that the total fitness score is larger than any of the individual scores. This makes me believe that all the nets are trained in random, different ways. This causes that one image that fools one net won't confuse the other nets. This doesn't always hold true because out of all the time ran the ensemble function, it was never able to pass 9600. This might be that there are 400 images that are extremely difficult in nature. I even suspect that they would be confused by a human that looked at the image.



Seen here are some of the more notorious images in the MNIST data set. As shown, these don't even resemble numbers to humans and are probably better considered noise in the system. From what I read about in papers and articles about other attempts to create a better system, they all seem to cap out in fitness at 9975. This is because there are some images that are mislabeled, badly drawn (like shown above), and there shouldn't ever be a network that scores 1000.

There were 6 artificial networks used in the ensemble.

!) A network with layers 784x10x30x10, with a mini batch size of 10 and a training coefficient of 0.5

2) A network with layers 784x10x50x60x30x10, with a mini batch size of 10 and a training coefficient of 0.5

3) A network with layers 784x50x60x10, with a mini batch size of 20 and a training coefficient of 0.3

4) A network with layers 784x50x60x30x10, with a mini batch size of 10 and a training coefficient of 0.5

5) A network with layers 784x50x60x30x10, with a mini batch size of 20 and a training coefficient of 0.3

6) And finally, a network with layers 784x60x30x20x40x20x10, with a mini batch size of 10 and a training coefficient of 0.5

The batch size and training coefficient is not very differentiated within the networks. This was unintentional and was chosen at random while creating the network. All networks were trained over 30 epochs.