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
η = 2.0	9168	9390	9499	9556	9575
η = 1.5	9165	9357	9481	9548	9577
η = 1.0	9100	9384	9435	9515	9567
η = 0.5	9090	9358	9481	9478	9499
η = 0.25	9039	9334	9348	8560	9424

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

Eta/HLS	10	20	30	40	50
η = 2.0	9264	9120	9298	9220	9272
η = 1.5	9195	9251	9232	9233	9257
η = 1.0	9194	9251	9280	9239	9273
η = 0.5	9244	9186	9270	9274	9204
η = 0.25	9085	9199	9201	9225	9115

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
η = 2.0	9451	9398	9427	9481	9479
η = 1.5	9421	9437	9463	9406	9417
η = 1.0	9425	9394	9402	9442	9423
η = 0.5	9310	9352	9393	9413	9397
η = 0.25	9344	9295	9375	9341	9385

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
η = 2.0	9505	9475	9562	9536	9562
η = 1.5	9500	9499	9516	9516	9519
η = 1.0	9357	9431	9466	9489	9505
η = 0.5	9364	9358	9450	9469	9475
η = 0.25	9346	9379	9395	9419	9396

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
η = 2.0	9493	9518	9564	9535	9538
η = 1.5	9489	9508	9546	9542	9540
η = 1.0	9473	9531	9481	9531	9559
η = 0.5	9394	9447	9482	9500	9510
η = 0.25	9336	9384	9416	9446	9426

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
η = 2.0	9481	9512	9541	9596	9545
η = 1.5	9506	9536	9541	9557	9552
η = 1.0	9495	9510	9496	9520	9503
η = 0.5	9420	9462	9480	8602	9518
η = 0.25	9354	9420	9446	9417	9434

Observations on the impact of the architecture and η on recognition accuracy.

Generally speaking, and the qualitative term is “generally”, improvements were seen in accuracy as the learning rate scaled up and as the number of layers, and nodes in the number of layers were increased. It seemed as if the more room we gave the models to “squish” and “squirm” the more accurate they became. However these improvements were small, and “nothing to write home about”.

PROBLEM 4

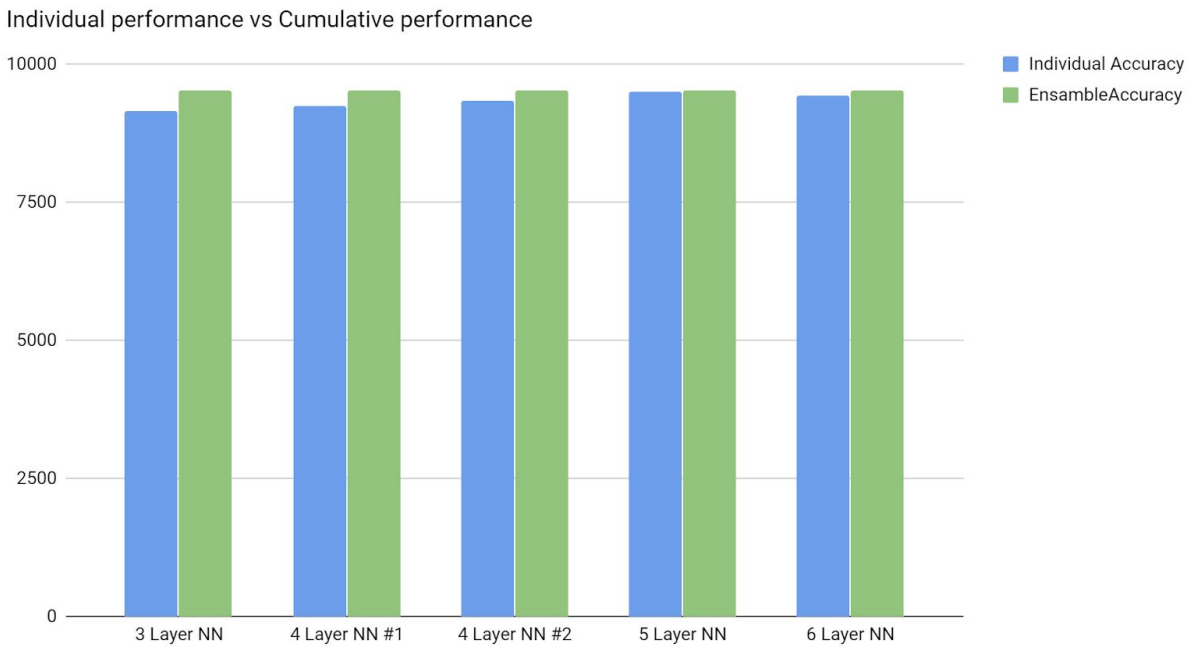

Chart 4.1 Here is shown the individual performance of each NN measured against the performance of the ensemble.

Net Information	Validation Accuracy
Net Name: 3 Layer NN **Net Architecture: 784x10x10** **Net Mini Batch Size: 15** **Net Learning Rate: .3** **Net Num_Epochs: 50**	**9151 / 10000**
Net Name: 4 Layer NN #1 **Net Architecture: 784x10x10x10** **Net Mini Batch Size: 5** **Net Learning Rate: .25** **Net Num_Epochs: 50**	**9244 / 10000**
Net Name: 4 Layer NN #2 **Net Architecture: 784x20x20x10** **Net Mini Batch Size: 5** **Net Learning Rate: .1** **Net Num_Epochs: 50**	**9345 / 10000**
Net Name: 5 Layer NN **Net Architecture: 784x40x40x40x10** **Net Mini Batch Size: 10** **Net Learning Rate: .25** **Net Num_Epochs: 50**	**9492 / 10000**
Net Name: 6 Layer NN **Net Architecture: 784x80x80x80x80x10** **Net Mini Batch Size: 20** **Net Learning Rate: .25** **Net Num_Epochs: 50**	**9421 / 10000**
Net Name: Ensemble Net	**9529 / 10000**

Table 4.2 This table shows all the stats for the various nets for Problem 4

Overall, gains were achieved by using the ensemble approach. Here is the part where I have to start some speculation. Of the five networks that I trained, they had validation accuracies from 9151 with a 3 NN to 9492 with a 5 layer NN. My approach consisted increasing the size of the hidden layers by 225% with every successive net. While I can readily agree with other ANN researchers that such ensembles are more robust, in addition to not being as sensitive to noise as singular NN's, my work just didn't show that.

I really liked how this homework started off by making simple NN's for the MNIST dataset, and progressed to doing an ensemble type of training and voting. Perhaps with more epochs higher accuracies could be achieved than what I had experienced. I only wish training could have gone faster, it kind of felt like being in a time crunch to train all my datasets for approximately 20ish hours, however that is probably par for the course.