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**
2	91.06	94.20	94.89	83.79	95.82
1.5	92.34	93.86	94.93	95.34	95.72
1.0	91.42	93.91	85.38	95.11	75.31
0.5	89.47	93.37	94.22	94.61	95.23
0.25	90.80	93.13	93.93	94.48	94.20

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

Eta/2nd HLS	**10**	**20**	**30**	**40**	**50**
2	91.98	92.15	93.11	92.62	92.61
1.5	92.75	92.57	92.75	92.36	91.62
1.0	92.29	92.51	92.46	93.15	93.29
0.5	91.86	92.26	91.90	92.96	91.70
0.25	91.38	91.49	91.40	91.62	91.93

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**
2	94.33	94.25	94.03	94.61	94.71
1.5	93.56	93.74	94.72	83.35	94.33
1.0	93.97	93.90	94.09	94.30	94.83
0.5	93.20	93.80	94.06	93.74	93.63
0.25	92.68	93.26	93.57	93.70	93.78

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**
2	94.45	95.33	94.64	95.17	95.24
1.5	94.53	95.07	95.19	95.12	95.11
1.0	94.21	94.62	94.78	94.76	94.90
0.5	94.15	94.14	94.12	94.69	94.51
0.25	93.51	93.90	93.97	93.67	94.13

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**
2	94.88	95.14	95.61	95.62	95.65
1.5	94.47	95.11	95.76	95.21	95.49
1.0	94.53	94.58	94.88	95.30	95.54
0.5	94.03	94.85	94.67	94.71	94.73
0.25	92.99	94.18	94.06	94.18	94.68

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**
2	94.85	95.45	95.43	86.26	95.90
1.5	94.96	95.33	95.72	95.36	95.64
1.0	94.38	94.89	95.21	95.52	95.55
0.5	93.88	94.69	94.59	94.87	95.24
0.25	93.99	94.37	94.35	94.67	94.20

An optimum learning rate is important as it decides whether your network converges to the global minima or not. Selecting a small learning rate can help a neural network converge to the global minima but it takes a huge amount of time. Therefore, you must train the network for a longer period. From the tables above, 1.0 – 2.0 yield higher accuracy. The number of layers also play an important role and increasing the number of layers increases the accuracy.

Problem 4:

Network Name/various parameters	**Architecture**	**Learning rate**	**mini-batch size**	**number of epochs**	**classification performance**
net_784_20_30_50_40_20_30_10_10_10.pck	784X20X30X50X40X20_30X10	0.1	10	300	92.30%
net_784_20_40_10_10_15.pck	784X20X40X10	0.1	15	300	90.92%
net_784_20_50_40_10_10_15.pck	784X20X50X40X10	0.1	15	300	91.43%
net_784_20_50_40_30_20_10_50_20.pck	784X20X50X40X30X20X10	0.5	20	300	93.43%
net_784_30_10_30_5.pck	784X30X10	0.3	5	300	94.70%

The architecture of each individual network in the ensemble, its learning rate, its mini-batch size, and the number of epochs is compared above in the table.

Theoretically, we should train a bunch of different models (i.e., an ensemble) and use all of them to classify test data. Each model in the ensemble may be better than the rest on some data features. Therefore, when put together, an ensemble of models is more robust to noise and more accurate.

The accuracy of ensembled network of all above networks is **94.83%.**
Or Ensemble of networks defined by nets classified 9483 images correctly out of 10000 validation images.
This justifies the thought of ANN researchers that the ensembles perform better than individual networks.