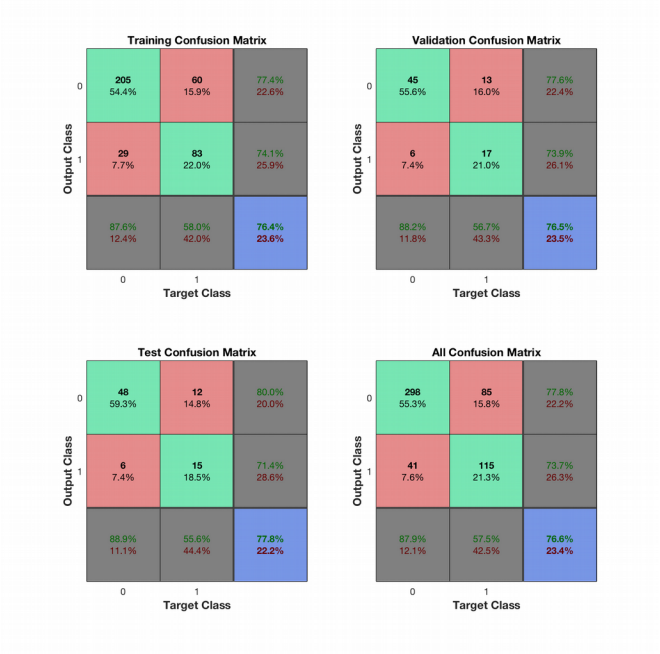
Dominic DiPasquale
CS1567 Introduction to Machine Learning
Professor Milos Hauskrecht

Problem Assignment 7

Problem 1:

a) It seemed that changing the variable for the number of epochs had a minimal effect on the error in our neural network, most of which hovered in the .4-.5 range. It seemed that the difference was also minimal for this dataset between the different training functions, which traincgb preforming .02-.04 better than something like trainrp. All in all for this particular implementation epochs and training function did not seem to change the outcome much.
The best weights as reported are 1.1444 3.0604 -0.8236 -0.0490 -0.4154 2.3983 1.2798 0.5400



I thought that this visual aid was really cool because I know that confusion matrices are really important and I've implemented many before.

b) For main1.m I saw a misclassification error of 0.2375 for the training set, a mse of 0.5235 for the training set. For the testing set misclassification was 0.2227, and mse was 0.4796.
For logistic_NN.m misclassification for training was 0.2412, mse was 0.4849 for training. For the testing set misclassification was 0.2052, and mse was 0.4550.
Based on these numbers it seems that no hidden layers outpreformed the 2 hidden layers. That is not to say this is always the case as the numbers are very close, and running the algorithms multiple times changed the values reported.

c) By changing the number of hidden layers I saw that 10 more consistantly gave a mse test of around (0.45) while 2 layers would sometimes be comparable to that there were also many times when it jumped up to (0.55). Again I observed that changing the training function to things like trainscg, traincgf, trainoss, etc the mse test really wasn't changed. Since I tried so many different configurations I will report trainoss, with 1500 epochs and 5

hidden layers class_error_train = 0.2319, mse_error_train = 0.4710, class_error_test = 0.2358, mse_error_test = 0.4628

Problem 2:

a) Strictly based on the numbers, the restricted tree has a lower error with (.2576), while the unrestricted tree has an error of (.2751). This is probably do to a form of overfitting as we have seen int previous models. Pruning will decrease the overfitting and thus increase the models general power so we should probably always prune when we have the opportunity.

b) By changing the minparentsize to 100 I was able to push the error all the way down to (.2314). After 100 the numbers seemed to jump back up to what they were before. Another thing that I observed was anything above the number of examples had very high errors which makes sense since we aren't providing enough parameters. Changing the split criterion was a little more interesting, twoing reported similar results to gdi for the same numbers in minparentsize and leaf size. Deviance was considerably worse in the case with 100 min parent size, and slightly worse with a min parent size of 10. Overall the smallest error I could achieve was (.2314).

Problem 3:

a) Trying 1,3 and 5 it is clear that 5 is better, it has the highest accuracy (.7773) and the lowest error (.2227) out of all 3. Because of this I tried using 100 nearest neighbors, but 100 was comparable to having 1 nearest neighbor, clearly this is susceptible to overfitting as well.

b) By normalizing the data the algorithm preformed better for numbers 1 and 3. For 5 the non normalized data seemed to preform better The difference was minimal but definitely present.

	Error	Accuracy	Error (norm)	Accuracy (norm)
1NN	0.2969	0.7031	0.2882	0.7118
3NN	0.2664	0.7336	0.2445	0.7555
5NN	0.2227	0.7773	0.2314	0.7686