Most of my research was with the Iris data set, as this was the only data set that we could split into training and test sample sets and test our generalisation ability. My initial explorations were testing the behaviour of various activation functions across all the COSC420 Sample data sets. The functions I choose to explore were Sigmoid, Relu (three variants), and Sine. After this initial investigation, I turned my attention to decreasing my population test error on the Iris sample set by extensive parameter tuning and recording how these changes resulted in different training performance. I will begin with a brief explanation of the program structure and then explore each of the research avenues and adventures.

My program initially asks the folder that you want to run the in, out, and param input files from, this allows you to place the input files in separate folders and call them, rather than moving the python file around. There is logic related to generalisation that only happens if you run the iris set from within a folder 6\_iris (or 7\_iris\_extra) relative to the program. After this you can choose the activation functions for the hidden and output layer. This structure allowed me to iterate quick, but it could do with some work. I have tried to be explicit with variable and function names to make things readable, I hope it suffices for your review.

When you run my program, you will be asked what folder you want to run the data out of. This allows you to store the separate models of in.txt, out.txt and param.txt in folders relative to the program, rather than having to move the python script around, or rename these config files. You can leave this blank which will run the program based on the current folder contents. After you choose the folder (or not) to run from, you choose the activation function for the hidden layer, and the output layer. After this, you also tell the program whether this model needs to learn generalisation (Applicable only for Iris). When generalisation is enabled, the logic is slightly different for testing, because it will be tested against the test data set only. Once the program is logged up, you are presented with a rudimentary user interface where we can choose several actions. The bulk action is the only action that requires further user input. It will ask you the number of iterations you want to perform, and the number of epochs per each iteration.