I designed my neural network to have two inputs, ten hidden nodes, and one output node. The inputs correspond to the ‘x’ and ‘y’ grid inputs. I decided using a single binary output made the most sense as we are answering one yes or no question. To determine the number of hidden nodes I tested even number of nodes from two to twelve. The total cost figure was very close between all the tests, but it seemed as though ten nodes was a little more consistently lower in cost.

In order to solve the problem, I focused on the circle and the .4 radius. I used the formula for the distance between two points to provide how far from the center the test point was located. If it was greater than .4, it was given the label ‘0’. If it was .4 or less, the test point was given the label ‘1’. The training set was built by building a tuple for randomly set ‘x’ and ‘y’. The values of those random ‘x’ and ‘y’ are put through the distance between two points formula. The tuple of ‘x’ and ‘y’, along with the label of ‘0’ or ‘1’ is given to the ClassificationDataSet object via the addSample method. This is repeated within the for loop for 1000 iterations. The data is split so that 75% of it goes to training with the remaining left for test data. The network is then built using the buildNetwork function. Finally backpropogation is accomplished with the BackpropTrainer function.

There is a problem with the percent error piece of my code. It keeps throwing an error complaining about a KeyError with ‘class’. I researched the problem and could not find a resolution. I also could not work through the plotting of the points.