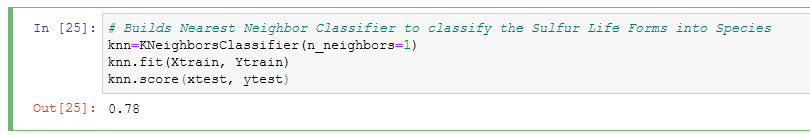
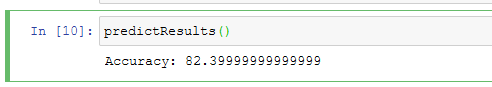
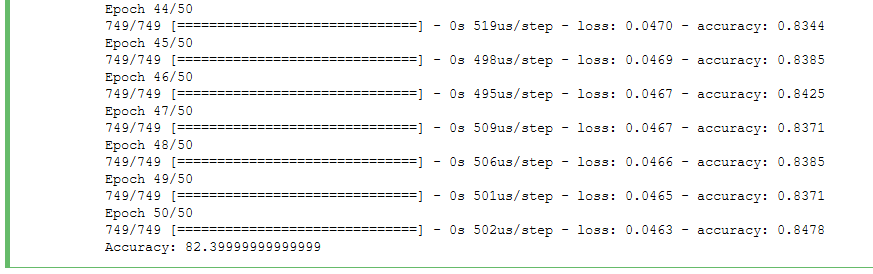
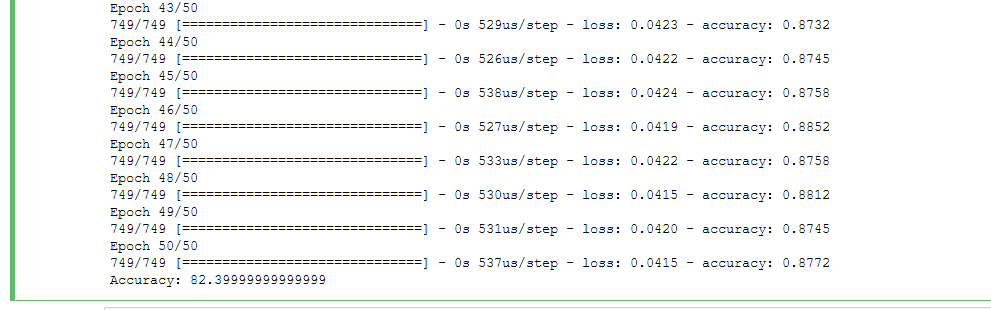
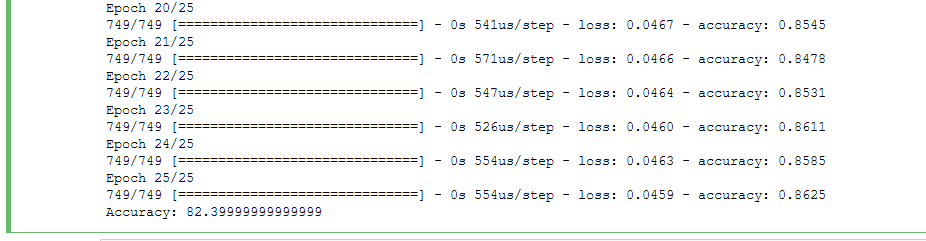
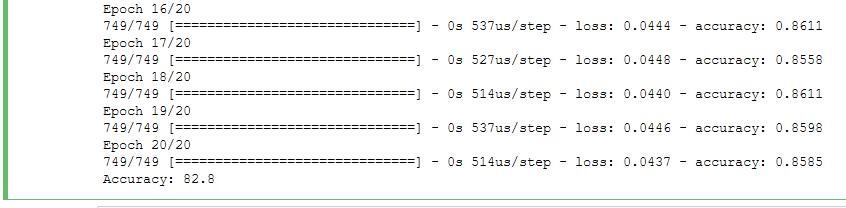
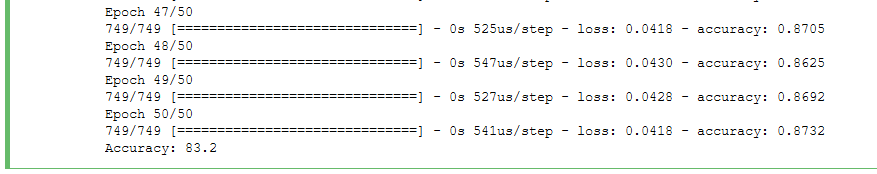
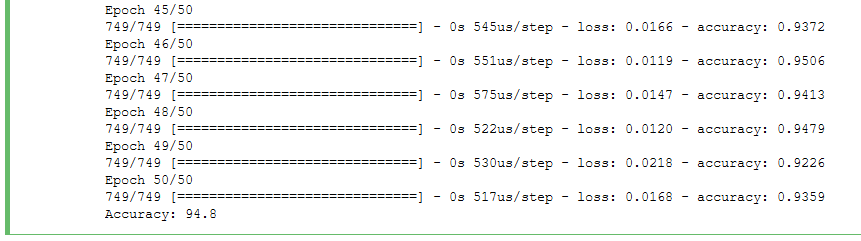
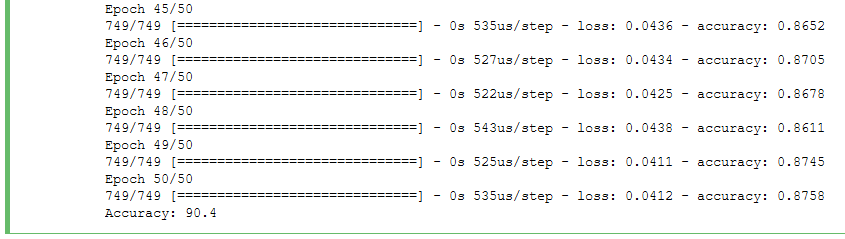
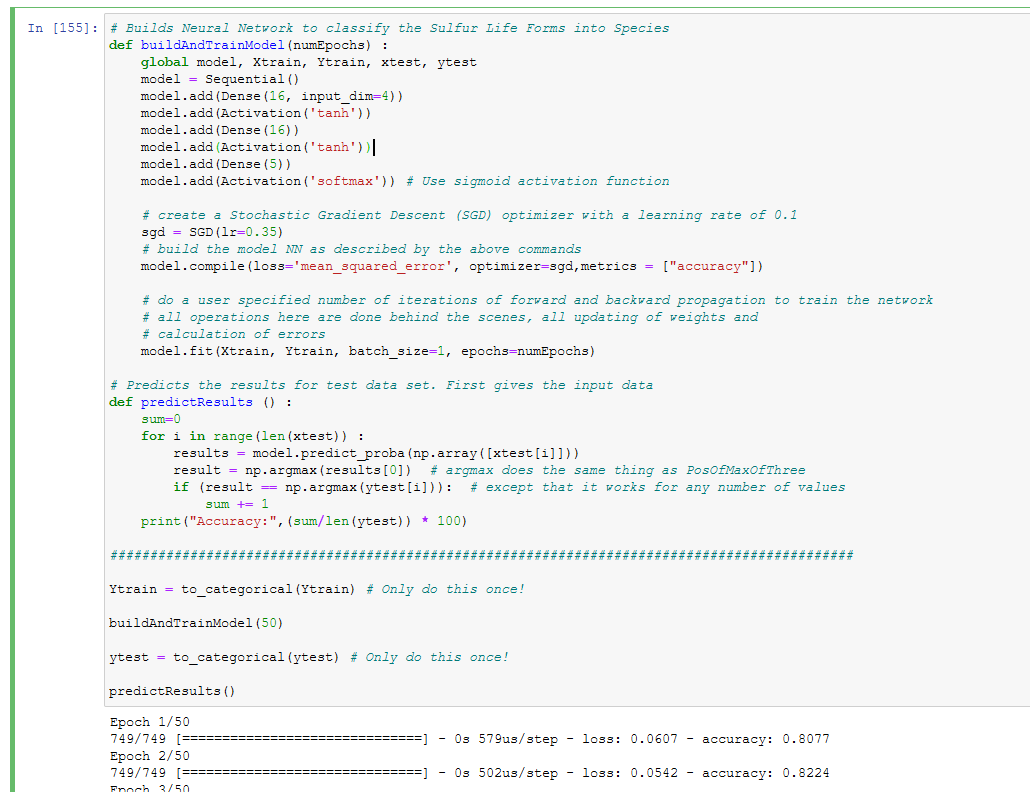
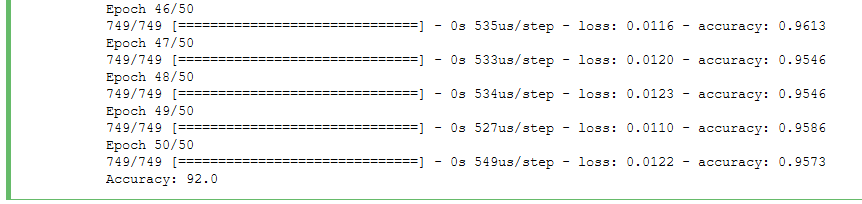
Kyle T. Wylie – 3 May 2020 – Final Exam Take-home Portion

1. KNN - 1 Neighbor: 0.78 or ~78%. 
2. ANN - 3 Inputs -> 3 hidden (tanh) -> 5 hidden (softmax): 0.8239… or ~ 82%. 
3. Such a classifier would be as accurate as “(number of occurrences of “Alliorated”) / (total number of creatures)”, which in this case since Alliorated occurred 821 times, and the total is 999 creatures, 821/999 = 0.821… or ~82%. 
4. The previous employee seems to have produced a KNN classifier and an ANN classifier that are about as effective as simply claiming that all of the creatures are Alliorated (see above percentages), and so ideally the classifiers should be able to do better, assuming there is some correlation in the data to be found. I would therefore deem their work unacceptable.
5. b: My initial thought was that the classifier simply needed more training cycles (it was initially set to just 10), so leaving the network untouched and running 50 cycles yielded the following result: Here it can be clearly seen that the network as given doesn’t improve much with additional training. My next thought was that perhaps the network is too small, since it only has 8 hidden neurons total. So I added a hidden layer from with 16 (tanh) after the first to try and give it some more “processing” capabilities, these were the results: This appeared to result in a network which performed better on the training data, but for some reason did not do any different on the testing data. I then thought that I might be overfitting this new network to the training data, so I reduced the training cycles from 50 to 25, here were the results: Once again causing no change to the output (I played with a few different numbers of cycles and found that the final accuracy would fluctuate by a few percent up or down with no real pattern). I then decided to try and increase the learning rate from 0.1 to 0.25, to see if maybe it was getting stuck in a local minimum, these were the results: The result was an increase in accuracy by a fraction of a percent. Making slight alterations to this value seemed to do little to alter the resulting accuracy. The very last thing I did was I decided to bring the number of cycles back up to 50 and increase the first hidden layer’s neurons from 3 to 16, here was the result: At an accuracy of 83.2% for the testing data, this was the best result I was able to get from the network up until this point. I made a few alterations to the activation function (from tanh to softmax) but found that they tended to decrease the output accuracy, so anytime I did that I ended up switching back.
6. To make this determination, I began by producing two augmented versions of the original dataset: one with the CDD column, and the other with the MSP column. I then attempted to fit the network with each new dataset (the input neurons had to be increased from 3 to 4) to see how it performed. Here are the results…

…with CDD: 

…with MSP: Since the network seemed to perform better with the CDD column added, I would have to pick the “Cellular Divisions since Dispersion” as my additional dataset.

1. This was the final network, using the original dataset augmented with the “Cellular Divisions since Dispersion” data: …and this was the final accuracy: Which overall constitutes an increase in accuracy over the original network (or, as the original employee pointed out, simply regarding all creatures as “Alliorated”) of ~15.6%.