**Question 2**

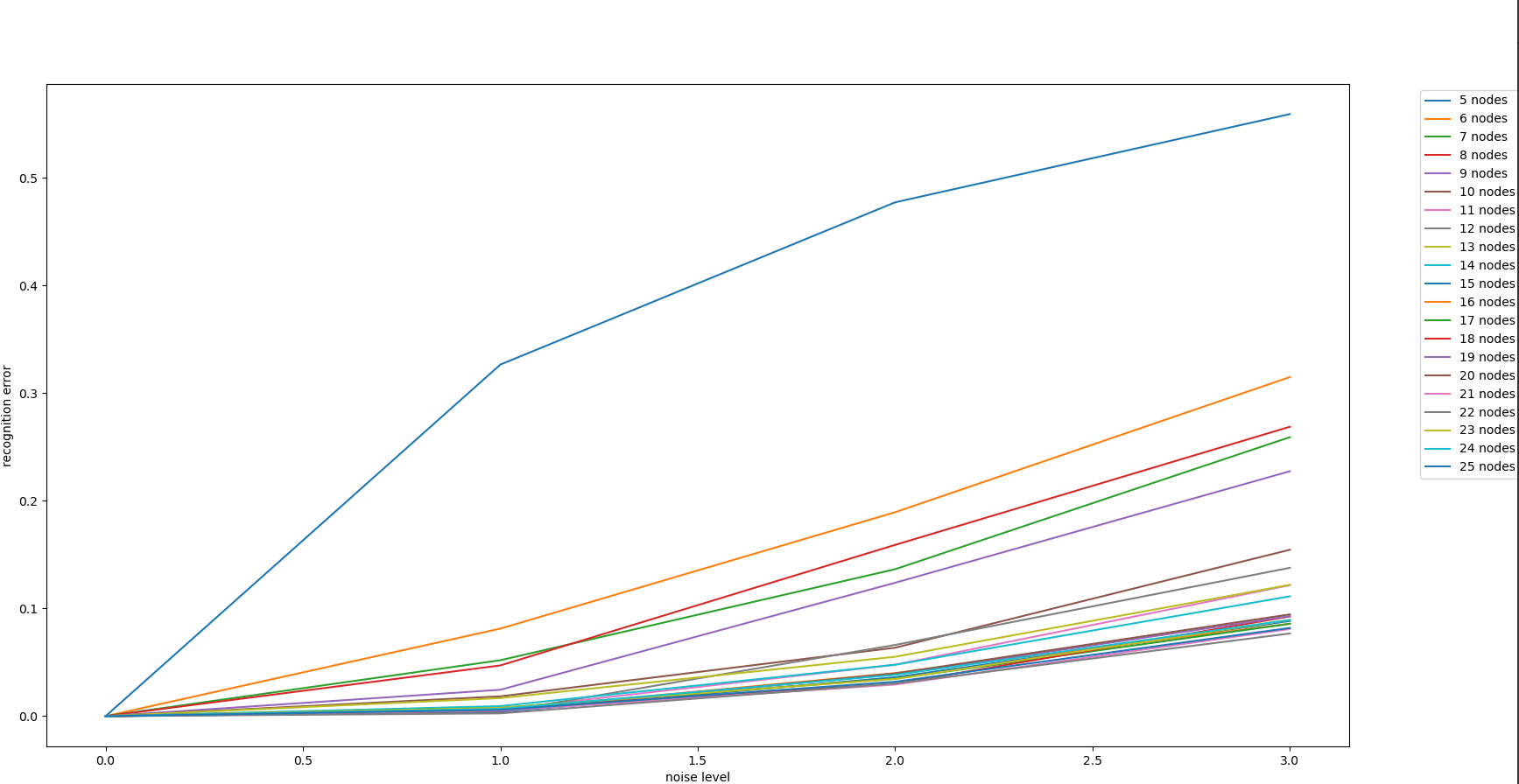
**Part A:**

1. **Run experiments with hidden neuron numbers in the range 5-25.**

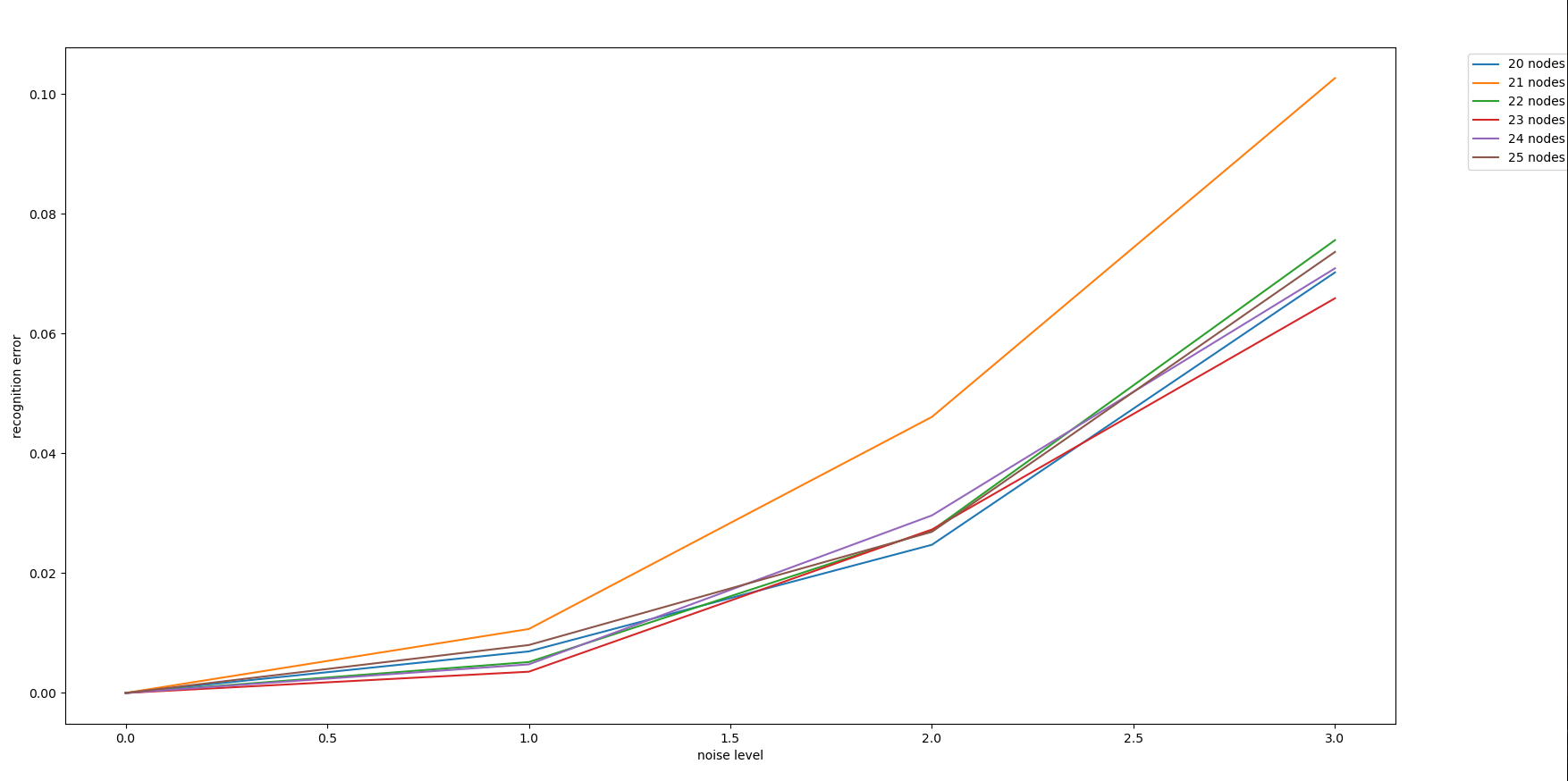
See Q2PartA.py and model\_Q2PartA.py to see the creation and training of networks with varying sizes of hidden layers.

1. **Plot a chart of recognition error against the number of hidden neurons.**

The following is the table for recognition error rate at differing levels of noise for networks with hidden layers of size 5-25:



In general the best results for recognition rate seem to occur when our hidden layer has between 20-25 neurons. If we plot just those networks we get a clearer picture of what our optimal size is:



In general all hidden layer sizes between 20 - 25 perform relatively similarly with 23 being the best by a slim margin. These values can also vary somewhat as the noisy data is randomly generated. For the sake of simplicity we will say that our optimal hidden layer size is 23, although it is not true 100% of the time.

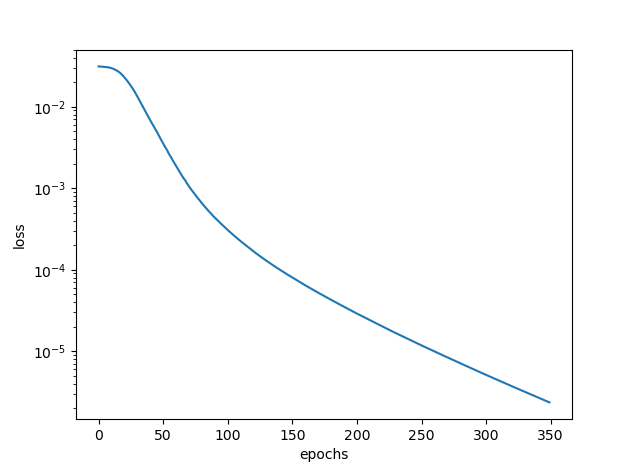
**Part B:**

1. **Confirm that Fig.13 is a reasonable representation of performance for the optimal number of hidden layer neurons chosen.**

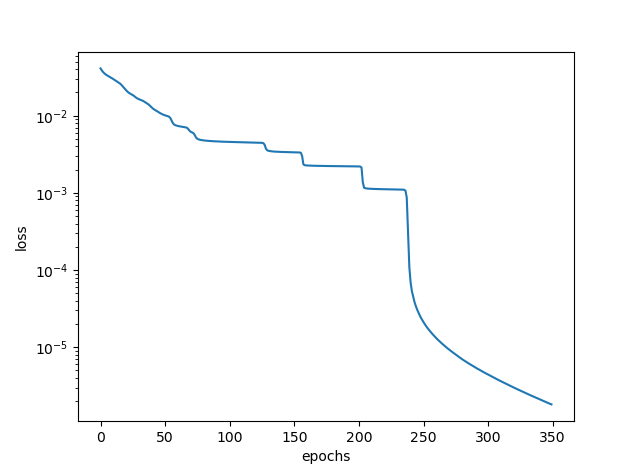
Answers to 1 and 2 are both written below

1. **Plot a chart in support of (1)**

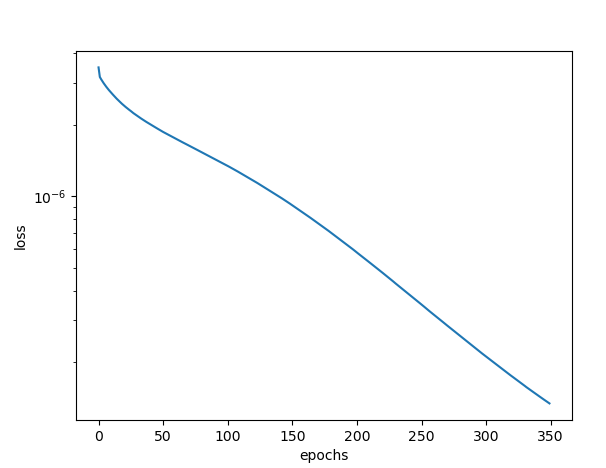
Figure 13.a is definitely a reasonable performance graph for our 23 neuron network. The following is an image of a 23 hidden neuron network trained on ideal data for a similar amount of time as the network in fig 13:



While the graphs are not exactly the same, the general rate of improvement is. It should be noted that this graph can vary depending on how our weights get initialized. When we get a more “unlucky” starting network we can see the sort of step pattern that occurs in fig 13:



As for fig 13.b, the performance seems more or less correct for a 23 hidden neuron network. The following is the performance graph of the network during training on ideal data for the second time:



Once again this graph takes on a slightly more linear shape than fig 13.b, but i believe this can also be accounted for by the randomness that occurs when adding noise to our data.

**Part C:**

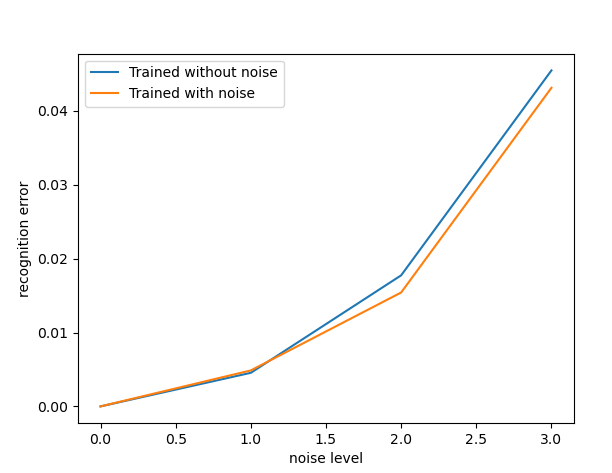
1. **Create testing data that has between 0 and 3 bits noise and include some examples in your submission.**

Within the submission are 4 text files called noNoise.txt, noisyData1.txt, noisyData2.txt, and noisyData3.txt.

These each contain all 31 character arrays with the relevant level of noise added, as well as the target outputs for this data. The data was generated using the same noise adding algorithm that was used for testing and training the network.

1. **Confirm that you can produce the recognition accuracy shown in Fig. 14.**

The following is the recognition error of a 23 hidden neuron network for noise levels 0 - 3:



The graph looks quite similar to fig 14, with the main difference being the actual recognition error values. I believe these are lower than fig 14 due to a difference in size of the training set, however the general shape of the graph is the same.