From the Toy Text or Classic Control collections of problems available, choose one. You may not use the MountainCar problem.

Adapt the neural network q-learning code example to learn a neural network for a Q-function of the problem. Train several different networks for the problem, and measure the performance using the problem’s rewards. (You can make your own immediate rewards to train the network, but must use the official rewards to measure per-epoch performance.)

Try with 1, 2 and 3 hidden layers, all of the same size. Try with 3 sizes of layers, each different by a factor of 2. For example, you may try the following network structures:

1- 1 hidden layer with 10 nodes. 2- 1 hidden layer with 20 nodes. 3- 1 hidden layer with 40 nodes. 4- 2 hidden layers with 10 nodes each. 5- 2 hidden layers with 20 nodes each. 6- 2 hidden layers with 40 nodes each. 7- 3 hidden layers with 10 nodes each. 8- 3 hidden layers with 20 nodes each. 9- 3 hidden layers with 40 nodes each.

Do not feel constrained to use the sizes suggested here. You choose, but follow the pattern. Train each network under identical situation, for the same number of epochs before measuring the performance.

In your report, include a table of the performance of each network, and an analysis of the smallest network you would use for this problem.

Notes

- For measuring the performance after training, you’ll want to use a similar structure to that used in the Q-table learning assignments.