University of Waterloo

Department of Systems Design Engineering

SYDE 522: Machine Intelligence

Quiz #3 - Goal seeking behaviour of a soccer-playing robot

Deadline: **Friday March 27, 2015**

A mobile robot has an infrared scanner to measure the bearing to a goal. It is controlled by changing the speeds of the independent left and right wheel motors. We want the robot to embody **Goal Seeking Behaviour**, specifically by having the robot turn until its heading is aligned with the target, then move forward at full speed. To accomplish this, we had a human control the robot in this manner, and recorded the resulting control signals to use as training data.

In this assignment, we will generate a controller for the left wheel motor using decision trees and neural networks. The training data is noisy, so we will have to be careful not to **overfit**. You can load the training data as follows:

```
data = readtable('GS.csv')

x = data.Inp1;

y = data.Out1;
```

Training and Testing data

The first step is to divide the data into training and testing components. Divide your data so that 30% is testing data and 70% is training data. How did you divide it, and why did you choose this method? How else might you have done it, and why didn't you do it that way?

Decision Trees

Use the MATLAB function **`fitrtree`** to train a decision tree.

1. View the learned tree with **`view(tree, 'mode', 'graph')`**. What type of tree is it? What are the branch nodes doing? What are the leaf nodes doing? What else do you notice about it?
2. Run the decision tree on the training input using the **`predict`** function, and compare it to the training output (make a plot). Now run it on the test data. Which looks like a better fit? Compute the RMSE on both the training and test data. Which one is better, and why?
3. Try pruning the tree using the **`prune`** function. Limit the tree to 10 levels. (Note that the prune function takes the number of levels to *remove*. Read the documentation to determine how to prune so you only have 10 levels left.)

Neural Networks

Use the **`feedforwardnet`** function to generate a neural network object, and the **`train`** function to train it.

1. Train a neural network with only one hidden unit. Plot the network output on the testing data. Describe the shape of the output. Why does it have this shape? Use the view command to view the neural network structure. What does this tell you?

2. Train a neural network with 10 hidden units. When the "Neural Network Training" window has popped up and finished training, click the "Performance" button. What is the resulting plot showing you? Plot the network output on the testing data. Does this fit look better than the network with one hidden unit? Measure the RMSE of the network on the testing data, and compare it to the RMSE of the network with only one hidden unit. Which is better?
3. Train networks with 1, 2, 5, 10, 20, 50, 100, and 200 hidden units, and measure the RMSE on both the training and testing data of each network. Make a plot comparing the number of hidden units in the network and its RMSE on each data set (you might want to use a log scale on the x-axis, see `semilogx`), and describe the results. Why did you get these results? What is the optimal number of neurons (of the ones tested) on the test set, and why is the RMSE lower there than for other numbers of neurons? Why should you really use a validation set when determining something like this?

Comparison

1. What are the similarities and differences between pruning decision trees and changing the number of hidden units on a neural network? What are the advantages to using a deeper tree or larger network? What are the advantages to using a shallower tree or smaller network? How can you choose the right size of tree or network?
2. What similarities and differences did you notice between the types of functions learned by decision trees and neural networks? Which one results in better performance? Which one trains faster? What are some advantages and disadvantages to each method? Which method do you prefer for this problem?

Submissions

You should submit the following:

1. Matlab/Octave code and instructions about how to run it.
2. A report containing a summary about what you have done, including answers to all the above questions and the requested plots. Feel free to use additional plots to help explain your results. As much as possible, write like you would in a scientific paper: in paragraphs, not as separate answers to each question.
3. Zip the quiz report and the source code (including a README file) and name it "**Quiz#3-Your Team Number#.zip**".
4. Upload this file to **Quiz#3** drop box available on UW LEARN.

Definitions

RMSE (root-mean-square error): The scalar error found by taking the difference between two vectors, squaring all elements, finding the mean across all elements, and finally taking the square root.

Training data: A subset of the data used for training the system.

Testing data: A subset of the data used on the final system to evaluate performance. The idea is that the error on the test set should approximate the error that the system will make in the real world.

Validation data: A subset of the data that is used for testing during training. For example, we may use it to pick the optimal number of hidden units by training networks of different sizes and comparing their performance on the validation set. The validation set is different from the test set; the test set is used only at the very end, to evaluate the final performance.