University of Washington Bothell
Computing & Software Systems

Course title:	Introduction to Machine Learning
Course number:	CSS 490 / 590
Term:	Winter 2012
Instructor:	Jeff Howbert

Exercises 7

Date assigned:	Mar. 6, 2012
Date due:	Mar. 11, 2012, 10:00 PM

Questions 1) through 4) explore the effects of changing various parameters on classification accuracy in a small ensemble of neural networks. You will apply the script in `matlab_demo_15.m` to the vowel-recognition dataset `vowel-context.data` (see links in Schedule under Lecture 15).

Some information about the vowel-recognition dataset can be found in two documents, `vowel-context.names` and "An empirical comparison bewteen SVMs and ANNs for speech recognition", by Jixin Li (also linked from the Schedule). Read the Li paper before you start working with the script. Besides giving you important background on the dataset, it contains performance data you will compare your results in Question 4).

In brief, there are 990 samples in dataset `vowel-context.data`. They contain ten features, which represent patterns present in audio recordings of 11 spoken words. [I have not been able to access the original thesis which presumably describes how these features were extracted from the recordings.] The words are very similar, having the same initial and terminal consonants, and differ only in the vowels between the consonants:

heed hid head had hard hud hod hoard hood who'd heard

The words have been given 11 class labels (0 through 10) in the dataset. The dataset already contains a partition into training and test sets, with 528 and 462 samples, respectively. The script is set up to use this particular training/test partition.

Some of the structure of the neural networks is defined by the dimensions of the input data and the number of class labels. Specifically, there are 10 input nodes, corresponding to the 10 features in the samples, and 11 output nodes, corresponding to the 11 class labels. Certain parameters in the script `matlab_demo_15.m` control other aspects of structure in the neural networks or the ensemble:

`nRand`	The number of neural networks in the ensemble. Each network has a different initial randomization of its connection weights.
`nHid`	The number of hidden nodes in the neural networks.

`outputTransferFcn` The transfer function to apply in the output nodes.

The ensemble prediction is formed very simply, by summing up the real-valued predictions output by the individual neural networks.

The only thing you need to do to answer Questions 1) through 4) is change the values of `nRand`, `nHid`, and `outputTransferFcn` in the script. I recommend you save the modified script and run it from the command line, rather than executing by highlighting + F9.

1) [10 points] Run the script with values of `nRand` = 1, 3, 10, and 30, while keeping `nHid` = 5 and `outputTransferFcn` = `'purelin'` (linear).
   - Cut-and-paste the number of correct predictions for each value of `nRand` into your answer document (do not include the confusion matrix outputs), and indicate which result is which.
   - What trend do you see in the accuracy as `nRand` is increased?
   - This trend can be explained by the combination of a certain property of neural networks with a certain property of ensembles. What are those two properties? (Review the slides in Lecture 12 on important properties of base classifiers, and on bias-variance decomposition.)

2) [10 points] Run the script with `outputTransferFcn` = `'purelin'` (linear), `'tansig'` (tanh sigmoid), and `'logsig'` (logistic sigmoid), while keeping `nHid` = 5 and `nRand` = 10.
   - Cut-and-paste the number of correct predictions for each setting of `outputTransferFcn` into your answer document (do not include the confusion matrix outputs), and indicate which result is which.
   - Which choice of transfer function is best for this dataset (given current settings for the other parameters)? Which is worst?
   - Based on these results, do you think it's always true that a nonlinear transfer function gives a better fit than a linear one?

3) [10 points] Run the script with values of `nHid` = 1, 2, 3, 4, 5, 7, 9, 11, 13, 17, and 21, while keeping `nRand` = 10 and `outputTransferFcn` = `'tansig'`.
   - Cut-and-paste the number of correct predictions for each value of `nHid` into your answer document (do not include the confusion matrix outputs), and indicate which result is which.
   - Include a plot of the number of correct predictions vs. `nHid` in your answers document.
   - Which trend do you see in the accuracy as `nHid` is increased?
   - Given there are costs to having more hidden nodes (greater complexity and risk of overfitting; longer times to train), what value of `nHid` would you choose for this particular ensemble model?

4) [10 points] Compare your best accuracy obtained in Question 3) with the overall model accuracies reported by Li for various configurations of SVMs and neural networks (bottom line of Tables 3 and 4).

- How does your best neural network ensemble compare with the neural network models reported by Li?
- How does your best neural network ensemble compare with the SVM models reported by Li?
- Do you think Li's study presents a fair picture of the potential of neural networks for this difficult multiclass classification problem, relative to SVMs?

5) [10 points] Now that you've been through an introductory course on machine learning, let's revisit the last question from Exercises 1.

   Describe a situation in your life where machine learning might be beneficial. This could be something you deal with at work, at school, at home, or on the internet (e.g. a social networking site). Say as much as you can about the problem to be solved, the data or information you might collect, and the types of machine learning you think are applicable. (There isn't necessarily a correct answer to this question; I just want you see how your ability to recognize opportunities for applying machine learning has improved.)