Assignment 2

CSG220, Spring 2007
Due: Thursday, Feb. 8

1. Do Exercise 4.8 (p. 125). The hyperbolic tangent function $\tanh$ is defined as follows:

$$\tanh(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}.$$

As part of this exercise, prove the result given in the hint.

Also, sketch a graph of the $\tanh$ function, indicating how it behaves as its input approaches $\pm\infty$. There is a simple mathematical relationship between this function and the logistic squashing function $f(s) = 1/(1 + e^{-s})$. Describe this mathematical relationship precisely.

2. Assume given a supervised 2-class classification problem in which all input attributes are binary-valued. Name the classes 0 and 1, so the target attribute is also binary-valued. Prove that a naive Bayes classifier applied to any training data having this form is a linear separator. That is, for this resulting naive Bayes classifier there is a separating hyperplane such that every instance on one side of this hyperplane is classified as 0 and every instance on the other side is classified as 1. Thus you are to prove that any classifier resulting from naive Bayes in this situation is no more expressive than a simple perceptron.

Hint: Let x_i represent the i^{th} input attribute and let y represent the output. In this case the output y of the naive Bayes classifier will be 1 when $\hat{P}(y = 1|x_1, x_2, \ldots, x_n) > \hat{P}(y = 0|x_1, x_2, \ldots, x_n)$ and 0 when the reverse inequality holds. (For simplicity, you may ignore what happens when these two conditional probabilities are equal.) First express each of these probability estimates as an appropriate product (making the naive Bayes conditional independence assumptions). Here it is helpful to make the general observation that $p^x(1-p)^{1-x}$ is a convenient way to write the function giving p when $x = 1$ and $1 - p$ when $x = 0$, for any x taking on only the values 0 or 1. The other key step involves transforming the resulting products into sums by taking logarithms and rearranging terms appropriately.

Your proof should yield explicit expressions for all the weights (including bias weight) in terms of the quantities used in the naive Bayes calculation.

For the remaining problems, you may either write programs from scratch yourself or use the suite of programs available at the “Homegrown Machine Learning Programs” link on the course web page.

3. Run the perceptron algorithm on all of the PlayTennis data. Use a 1-out-of-3 encoding for Outlook and for Temperature, and a single node for each of Humidity and Wind. Is the training data linearly separable? Briefly comment on which attribute values seem to play the biggest role in this trained perceptron’s decision.

4. Run the perceptron algorithm on all the PlayTennis data again, but this time using a single node for each attribute. Encode the values of Outlook as Sunny = 0, Overcast = 1/2, and Rain = 1, and encode the values of Temperature as Cool = 0, Mild = 1/2, and Hot = 1. Briefly comment on the results. Is the training data in this form linearly separable?

5. (a) Run the backpropagation algorithm on all of the PlayTennis data, using the same encoding as in the previous problem (using 2 hidden units). Briefly comment on the results.

(b) Reinitialize the same 2-hidden-unit multilayer network, and this time randomly divide the 14 examples into two halves, one for training and one for testing. What are the training set and test set accuracies you get in this experiment? (Assume output > 0.5 means PlayTennis = yes and output < 0.5 means PlayTennis = no for every test example.)

What to turn in for problems 3, 4, and 5: (a) output data obtained during (or upon completion of) the running of your programs; (b) any input data or script files used in the running of your programs (Lisp dribble files are very helpful here to capture both input and output); and (c) appropriate written commentary recording your observations and answers to the questions above. For the programs themselves, if you simply ran existing programs, do not turn in any source code but just indicate where you obtained the programs. Do turn in source code for any programs that you wrote yourself or modified, and include appropriate comments indicating which parts you wrote or modified.