CS/CNS/EE 156 a — Caltech

Learning Systems — Fall 2000

Homework # 2

DUE TUESDAY, OCTOBER 24, 2000, AT 2:30 PM

Collaboration in the sense of discussions is allowed, but you should write the final solutions alone and understand them fully. Do not read class notes or homework solutions from previous years at any time. Other books and notes can be consulted, but not copied from. You should justify your answers, at least briefly. Definitions and notation follow the lectures.

The handouts and data for the homeworks can be found at:

http://work.caltech.edu/cs156/00/homeworks.htm

1. Growth Function and VC Dimension

(i) Find the VC dimension D and the growth function $m(N)$ for the 'two intervals' learning model, where $g_\alpha : R \to \{-1, +1\}$ and $g_\alpha(x) = +1$ if x lies within either of two (arbitrary) intervals. Verify that $m(N)$ obeys the VC bound.

(ii) Let D be the VC dimension of the 'triangle' learning model, where $g_\alpha : R^2 \to \{-1, +1\}$ and $g_\alpha(x) = +1$ if x lies within an (arbitrary) triangle. Show that $D \geq 8$.

(iii) Construct a learning model that has VC dimension $D = 7$, for which you can compute the growth function $m(N)$ exactly. Verify that $m(N) \leq N^6 + 1$.

2. VC Dimension of Perceptrons

Show that the VC dimension of a perceptron (a single neuron with a hard threshold) with d inputs is exactly $D = d + 2$ by showing that

(i) There are $d+1$ points that can be fully dichotomized. *Hint:* Construct an invertible matrix using d+1 points, and conclude that a perceptron can implement any dichotomy on these points.

(ii) There are no $d + 2$ points that can be fully dichotomized. *Hint:* Use the fact that any d+2 vectors of dimension d+1 have to be linearly dependent to construct a ± 1 pattern on any given $d+2$ points that cannot be implemented by a perceptron.

3. Weight Decay

Modify your backpropagation code from HW1 by changing the nonlinearity $\theta(s)$ of the output unit *only* into a linear function ($\theta(s) = s, d\theta(s)/ds = 1$ at the final layer). Use a network with 8 hidden units and a learning rate of 0.05. Implement the weight decay regularization method on this network using the training data set:

http://www.work.caltech.edu/cs156/00/hw/hw2/insample.dat

and the test data set:

http://www.work.caltech.edu/cs156/00/hw/hw2/test.dat

Plot training and test errors for runs with no weight decay, and with weight decay parameter values $\lambda = 0.001$, $\lambda = 0.01$ and $\lambda = 0.1$. Compare the results.

4. Validation for Early Stopping

Using the network from problem 3 but with no weight decay, modify your code to implement the use of a validation set.

(i) Divide the in-sample data from insample.dat randomly into a training set of 25 examples and a validation set of 10 examples. Train until the change in training error between epochs is smaller than 1e-7 or for 5000 epochs, whichever comes first. Plot training, test and validation set errors.

(ii) Execute 10 runs with different splits of the in-sample data into 25 training examples and 10 validation examples. For each run find the test error at the final epoch (E_{NV}, the no-validation out-of-sample error) and the test error at the minimum of the validation error (E_{ES}, the early-stopping out-of-sample error). Compare the average E_{NV} to the average E_{ES}.

(iii) Repeat (ii), but with 10 training examples and 25 validation examples.