CSCI 3813/780 Machine Learning
Assignment #2
Training Multi-layer Neural Networks
Due before class, Monday, April 7, 2014

Introduction

This assignment asks you to implement the back propagation algorithm described in class for training multi-layer neural networks from data.

The algorithm

A typical multi-layer network consists of an input, hidden and output layer, each fully connected to the next, with activation feeding forward. Each arc has a weight associated with it. The weights determine the function computed by the neural network. When the weights are unknown, we have to estimate them from data.

The back propagation algorithm is a gradient descent method for training multi-layer neural networks. Each step of the algorithm calculates the prediction error based on the current weights, propagates the errors backward to previous layers, and adjusts the weights based on the errors. The step is repeated until convergence when the total output error is small enough.

The dataset

The *Computer Hardware Data Set* for this assignment is from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/Computer+Hardware). The dataset has 9 attributes and totally 209 data points. The first two attributes are names and should be excluded in your analysis. Therefore, the number of inputs is only 7. The task is to predict the relative CPU performance based on its cycle time, memory size, etc. For this assignment, use 3 hidden units.

Evaluation and analysis

Perform the following analysis with your implemented algorithm:

1. Normalize each variable of this dataset before the analysis, that is, divide the values of each variable by its maximum value so that all values are between 0.0 and 1.0.
2. Implement the back propagation algorithm, and use 5-fold cross validation to evaluate it.
3. For each experiment, you should stop training the neural network when the total output error is below some threshold value $\in$. Try the following values for $\in$: 0.1, 0.01, and 0.001.
4. For each experiment, plot the total output error against the number of epochs (iterations) during the training
5. Analyze your results and report any observations you may have based on your experiments. For example, explain how the stopping threshold value affects the training and testing error, etc.

Deliverables

Submit a single .zip file named LastName.FirstInitial.HW# containing the following to the instructor (changhe.yuan@qc.cuny.edu). In addition, submit a hardcopy of your reports in class on the due date.

1) Well commented code; C++, Java, Matlab, and Python are all allowed.
2) A readme file explaining how to compile and run your program;
3) A report explaining the neural network method, the back propagation algorithm, and your experimental results.