Backpropagation

Due date: Monday, 3/3/03

Overview

For this assignment you will implement a backpropagation network and train it to perform a simple character recognition task. Describe all relevant equations, including the feedforward activation calculation, signal function, backpropagated errors (deltas), and weight update rules employed. Compare how the network performs under different choices of parameters and architectures.

Assignment

You are to train the network to recognize bitmapped images of Arabic numerals and to convert them to a binary representation. The input patterns consist of 15 bits arranged in a 3x5 grid, as shown in Figure 1. The target output patterns are 4-bit binary representations of the digits (0000, 0001, 0010, 0011, 0100 etc. for the digits 0, 1, 2, 3, 4, etc.).

Your program should train the network using a large number of training epochs, each of which consists of all 10 input-output pairs. During training, you should track the current mean-squared error, which you can use to determine if the learning has converged sufficiently.

Write a concise but complete report of your implementation. Include all relevant equations (activation equations, signal function, delta calculation, weight update rule, etc.). Specify the architecture you chose. Specify the parameter values you chose. Discuss all of the above choices and why you made them. Because, in fact, it is very likely that some of these choices were made by trial and error, report on how you came to make the choices you made. What happened with the first delta parameter choice you tried? How did you decide on the number of hidden units to use?

Your report should include:
1) Plots of the mean-square error on the test set as a function of training epoch for different choices of network architecture. Discuss differences in rate of convergence.
2) Plots of the mean-square error on the test set as a function of training epoch for different choices of learning rate, for at least two network architectures. Discuss any interesting trends.
3) A plot comparing the asymptotic error size for different network architectures you tried. Comment on differences in the final error size.
4) A comparison of the number of epochs necessary and the computations necessary to train the same basic architecture using batch versus incremental updates. Discuss any differences.
5) An analysis of what happens when the input patterns are “distorted” by random input noise during testing. Add small amounts of Gaussian noise to each input bit. Show how performance varies as a function of the input noise standard deviation. What happens if you train the network using the same statistical distribution of noise used during testing? Discuss the results.
6) A plot of how the convergence changes when momentum is included (for at least one example network architecture and learning rate). Discuss this result.
7) Any other plots or analyses that help explain what choices affect network performance on the task.

Hints

(1) You may want to test your code on a smaller task (such as the XOR problem) first.
(2) Be sure to train the network with a sufficient number of training epochs. For the character recognition problem, roughly 500 epochs should be sufficient. For the XOR problem, 900 epochs may be necessary.
Figure 1: The input and output patterns.