CSCE 420 Project Specification (PRELIMINARY)

Dr. Daugherity

Due: 11:59 P.M. Monday, April 23, 2018

This project is to design and write a graphical character recognition program using neural networks and to explore their properties. The input will be a 5 by 7 dot pattern and the neural network will try to identify the letter. For example, if the input is

○●●●○

●○○○●

●○○○○

●○○○○

●○○○○

●○○○●

○●●●○

the neural net should identify the letter as C (although there may very well be less intense outputs for other similar letters such as G and O).

Use the 5 by 7 dot matrix font linked on Piazza and implement a neural net with 35 inputs (the dots) and 26 outputs A to Z (upper-case only). A minimum of two layers is required.

Implement the corrected back propagation algorithm posted on Piazza and train the neural net until it is fairly accurate (meaning the largest output is the correct letter 90% or so of the time). Capture the accuracy after each back-propagation pass and plot versus pass number to show the learning.

Hints: For the initial weights try random numbers between, say, -0.1 and +0.1 (but not zero). Experiment to find a useful value for α, the learning rate. You might try 0.1 or smaller, or an decreasing schedule like 0.1 divided by the pass number.

Now see how it performs when random input dots are flipped (black to white or white to black); this represents noise in scanned input. Test each of the 26 letters by progressively flipping bits at random (but without repetition) until the largest output is no longer correct; this gives you an estimate of the noise immunity of your neural net. Tabulate the number of flips each letter requires to make the output incorrect.

For the output layer use the logistic sigmoid activation function, but for all other layers use the hyperbolic tangent sigmoid activation function. Hint: Training and execution will be faster if you use table lookup or a decent approximation instead of calling the *exp* and *tanh* library functions at every step.

Note that back propagation requires both the activation function and its derivative, but the derivative can easily be calculated by

for the logistic sigmoid, and

for the hyperbolic tangent sigmoid.

Your program should run on compute.cse.tamu.edu and must be submitted both to CSNET and also on a CD or DVD or USB jump drive (which cannot be returned).

The project report (described below) should be submitted on paper in class, along with your CD or DVD or USB jump drive. Write a report according to the outline below.

REPORT OUTLINE

The project report must be printed on a laser printer. The report should include the following sections:

1. Statement of the problem, significance, etc.
2. Restrictions and limitations
3. Explanation of your approach (*e.g.*, analysis and experiments to choose how many layers, how many neurons in each layer, learning rate and schedule, number of back-propagation iterations, etc.)
4. Sample run (screen shots)
5. Results and analysis
6. Conclusions - What did you show? What did you learn?
7. Future research (how your program could be improved or extended)
8. Instructions on how to run your program
9. Listing of the COMMENTED program
10. Bibliography - references used, if any