CS4811: Homework 1 --- Artificial Neural Networks

Dates:

- **Assigned:** Wednesday, January 11, 2017
- **Interim program and report (10%) due:** Friday, January 20, 2017
- **Neural network 1 (30%) and report (15%) due:** Friday, January 27, 2017
- **Neural network 2 (30%) and report (15%) due:** Friday, February 3, 2017

General guidelines: This is an individual assignment. All the work should be the author's and in accordance with the university's academic integrity policies. You are allowed to use any previously written source for your research on the topic. However, any source other than the textbook and the class notes should be cited and the bibliographic information should be given.

You should implement the neural network from scratch without using any other existing software. You are **not allowed** to use code that is not your own. You should implement the algorithm given in the handouts in a language of your choice. We should be able to run your programs on the Department's Linux platforms.

Problem: In this assignment, you will implement a backpropagation learner for a feed forward neural network. Each node in the network should have the real inputs and the bias input. It should emit one output.

Tasks: Your task is to create two neural networks with possibly different architectures. Implement the algorithm and fully comment it so that the relevant sections of the pseudo code are seen in your code. Start setting up your program and experiments right away and submit an interim program and report within one week. Create the following two learning systems:

> **Neural network 1.** Create a backpropagation based neural network that learns the X-OR function: Train the network using the 4 examples that correspond to the correct outputs. After the learning process is over, generate several points (use at least 0.1 increments) in the 1x1 box and show the shape of the function learned by plotting the positive points. You may use your own plotting/displaying methods or use a spreadsheet after creating the data points with your program.
>
> **Neural network 2.** Create a backpropagation based neural network that learns the $y = x^2$ function. The program should learn from data that is classified as negative or positive, not from x,y pairs given as input. To do this, generate several examples such that $y \geq x^2$ are positive and $y < x^2$ are negative. Use these examples to train your network. After the learning process is over, generate several more points and show the shape of the function learned by plotting the positive points.

The interim report: Submit a single pdf file that contains the program listing and the report. You may use pdf software to split or join pdf files. The interim report should include the following:

- Provide a printout of your neural network program with a clear explanation of what works so far. Present the working parts with output examples. The program should be fully commented. Include your name, class, date, and assignment title at the top.
- Provide answers to the following six questions:
  1. Did you implement the basic framework of a feed-forward multi-layer neural network?
  2. Did you test whether feed-forward and back-propagation work? Which data sets did you use?
  3. Did you implement or think about how to change the number of layers or nodes?

    4. Did you prepare the data points for the training examples for the XOR function and the x^2 function? Did you plan how to input those points?
    5. Did you implement or think about how to graph the final neural network?
    6. What problems or obstacles did you encounter so far? How do you plan to deal with them?
- Provide a daily schedule of what you plan to do to finish your assignment. The dates should cover from January 20 to February 3.

Program submission: The program and related files should be submitted as a single compressed tar, gzipped tar, or zip file. The submission should contain the following:

- Program files
  The program should be fully commented. Include your name, date, class, and assignment title at the top.
- A README file with instructions on the following
    - How to run your program with the training sets you provided
    - How to change the training sets
    - How to graph the results
    - How to change the network structure
- Input data files, if any
- Output data files, if any
- Files related to graphs, for example, .csv or Excel files

Report submission: The report for each neural network should include the following:

1. **The software**
    1. A pdf printout of your program
    2. Describe the design of your program in essay format. Also include
        1. Did you fully implement the pseudocode provided?
        2. Does every component work?
        3. How can the number of layers or nodes be changed?
2. **The training process**
    1. How did you decide on the network configuration to use?
    2. How were the weights initialized?
    3. How many iterations were needed to converge (or stop)?
    4. What was the training data used (provide in text)? How did you create the training data?
3. **The end results**
    1. Tell how far the network converged. What percentage of the training data is classified correctly?
    2. Show the following
        - The trained networks (nodes and weights)
        - A graph of the training data
        - A graph of the function learned by using more inputs than the training data