COMP 3310 Artificial Intelligence
Makeup Project
Due Date: 06/30/16

Overview:

An important application of neural networks has been in the area of representing Boolean functions. You will need to write a C++ program that is capable of classifying an *n*-variable Boolean function. The program should be able to build a 2-layer Backpropagation network from a data file. The data file will include the value of *n* followed by all possible 2^n instances of thee function. The last value in each line represents the class of that instance, i.e., the value of the function (0 or 1). You will be given two sample data files that will contain all possible instances of an *n*-variable majority Boolean function (true when more than *n*/2 inputs are true) – one for 5 variables and one for 10 variables.

You will use the Backpropagation algorithm to implement the neural network. The neural network will contain one input layer (*n* nodes), one hidden layer (user will specify the number of hidden nodes), and one output layer (1 node). The user will need to provide the name of the input file, the number of hidden nodes, the number of epochs, and the learning rate – all from the command line. The first step will be to train the network using the training data. Then you will test the network using the same data.

At the beginning, you will need to initialize the network weights. Initialize the weights using random values (with a precision of two decimal digits) between –1 and +1. Learning is accomplished by successively adjusting the weights based on the set of training examples. During this iterative process, a training example is presented to the network and propagated forward to determine the resulting value at the output node. The difference between the calculated output and the desired output represents an error that is back propagated through the network in order to adjust the weights. The learning process should continue for some predetermined epochs. After the training process is complete, you will need to present each instance to the network and then calculate the resulting output. If the resulting output is in the range 0-0.49, consider that as "false". If the resulting output is in the range 0.5-1, consider that as "true".

A sample execution of the program should look like the following. For testing each instance, the first classification represents the calculated class and the classification inside the parenthesis represents the actual class of that instance. Note that the network weights are randomly initialized, so your output may be different.

```
> g++ program2.cpp –o program2
> program2 –f input1.txt –h 3 –e 10 –l .5
00000: False (False)
00001: False (False)
00010: False (False)
.
10011: True (True)
10100: True (True)
.
11111: True (True)

Accuracy: 0.88
```

Instructions:

- You MUST develop your program in the Linux environment using the g++ compiler. **YOU WILL LOSE 50% OF THE POINTS IF YOUR PROGRAM DOES NOT COMPILE IN LINUX.**
- Use meaningful variable names, helpful comments, and a consistent style.
- This will be an individual assignment. However, feel free to contact the instructor (and ONLY the instructor) if you need help. FAILURE TO FOLLOW THESE REQUIREMENTS WILL RESULT IN A GRADE OF F IN THE ASSIGNMENT.

Deliverables:

- You will need to email the source file(s) by 06/30/16.