CSC3022F: Machine Learning

Assignment 7:

Artificial Neural Networks

Department of Computer Science
University of Cape Town

Due: Friday, 19 June, 2020, 5.00 PM

PART I : **Problem Description**

Implement (in C++) an algorithm that uses *only* perceptrons with *threshold* activation functions and the *perceptron learning rule* (chapter 4 [Mitchell, 1997]) to solve the binary A XOR B problem. Table 1 shows the correct inputs and outputs for the binary XOR problem.

Question 1: What is the minimum number of perceptrons required to solve the XOR problem, and how should they be connected?

Question 2: Devise a list of training examples to teach the perceptrons to solve the binary XOR problem. How many training examples did it take for your algorithm to correctly learn to solve the binary XOR problem?

NOTE: Your perceptrons ***must not*** use *linear* or *continuous* activation functions, *gradient descent* or the *back-propagation* algorithm.

In a ZIP file, place the source code,makefile and a text file containing your list of training examples, as well as answers to question 1 and 2. Upload the ZIP file to Vula

Table 1: Inputs and outputs for the binary XOR problem.

Input 1	Input 2	Output
1	1	0
1	0	1
0	1	1
0	0	0

PART II : **Problem Description**

Implement (in C++) a fully connected feed-forward neural network that consists of 3 input neurons, 2 hidden layer neurons and 1 output neuron. The hidden layer neurons and output neuron use the *Sigmoid* (chapter 4 [Mitchell, 1997]) activation function.

The ANN input nodes 1, 2 and 3, respectively, have the following inputs:

$$x = [\ 1.30,\ 2.70,\ 0.80\ ]$$

The ANNs expected output is:

$y = 0.36$

Table 2 specifies the weights connecting the inputs to the hidden layer neurons. Note that each column denotes a given node connected to a given hidden layer. For example, the top value in column 1 is the value of the weight connecting input node 1 to hidden layer node 1, and the bottom value in column 1 is the value of weight connecting input node 1 to hidden layer node 2.

The bias values for the hidden layer neurons 1 and 2, respectively, are:

$$\mathrm{b} = [\ 0.1,\ -0.3\ ]$$

Table 2: Values of weights connecting input to hidden layer nodes.

Input 1	Input 2	Input 3
0.1	0.2	0.5
-0.4	1.0	-0.6

Question 3:

Given the ANN input x, what are the output values of hidden layer neurons 1 and 2 ?

Question 4:

Given that the weights from hidden layer nodes 1 and 2, respectively, are:

$$\mathrm{w} = [\ 0.8,\ 1.0\ ]$$

And the bias value for the output node is: $\mathrm{b} = -0.3$, and the ANN input x, what is the output value of the output neuron ?

Question 5:

Given this output, what is the *Mean Squared Error* for ANN input x.

In the ZIP file, place the source code,makefile and a text file answers to questions 3, 4 and 5. Upload the ZIP file *to Vula*. This is the SAME ZIP file as Part 1. Do **NOT** submit two ZIP files.

Alongside your archive, please submit a file containing the hash, of your archive, produced by a MD5 checksum. See the link below for a tutorial on how to produce the hash:
https://www.tecmint.com/generate-verify-check-files-md5-checksum-linux/

Failure to include a '.md5' file waives your right to appeal your mark in relation to issues that arise from corrupted, missing or incorrect submissions.

References

[Mitchell, 1997] Mitchell, T. (1997). *Machine Learning.* McGraw Hill, New York, USA.

You are required to use Git for version control. Git related penalties for Assignment 7 are the following:

[-10%] - usage of git is absent. This refers to both the absence of a git repo and undeniable evidence that the student used git as a last minute attempt to avoid being penalized.
[-5%] - Commit messages are meaningless or lack descriptive clarity. eg: "First", "Second", "Histogram" and "fixed bug" are examples of bad commit messages. A student who is found to have violated this requirement for numerous commits will receive this penalty.
[-5%] - frequency of commits. Good Git practices advocate for frequent commits that are small in scope. Students should ideally be committing their work after a single feature has been added, removed or modified. Tutors will look at the contents of each commit to determine whether this penalty is applicable. A student who commits seemingly unrelated work in large batches on two or more occasions will receive this penalty.

Please note that all of the git related penalties are cumulative and are capped at [-10%] (ie: You may not receive more than [-10%] for git related penalties).