

CSC3022H: Machine Learning

Lab 3: Artificial Neural Networks

Department of Computer Science
University of Cape Town, South Africa

August 23, 2015

Due: Friday, September 11, 2015, 10.00 am

Problem Description

Implement (in C++) the *Artificial Neural Network* (ANN) shown in figure 1, using the perceptron learning rule (chapter 4 [Mitchell, 1997]) with a linear activation function in the output neuron. Table 1 shows the training data that must be used to train the ANN.

Question 1: Initialise all weights = 0. How many iterations of the perceptron learning rule does it take before an error = 0 is achieved?

Question 2: Change the activation function to a threshold function and re-run the perceptron learning rule. Is an error = 0 achieved in the same number of iterations? Briefly justify why or why not.

In a ZIP file, place the source code, executable, and two text files. The first text file is for the ANN using the linear activation function and the second for the ANN using the threshold activation function. Both text files should contain the ANN weight values, output and error (for all training examples) at each iteration that the perceptron learning rule was run (i.e.: until error = 0).

Upload the ZIP file to *Vula* before 10.00 am, Friday September 11.

Table 1: Training data.

Input 1	Input 2	Input 3	Input 4	Target Output
1	-1	1	-1	1
1	1	1	1	1
1	1	1	-1	-1
1	-1	-1	1	-1

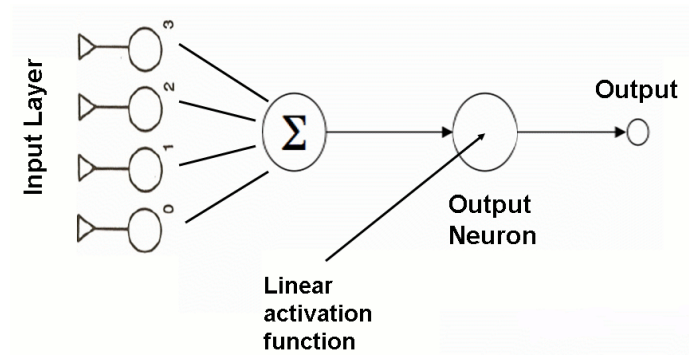


Figure 1: ANN with 4 input neurons connected to 1 output neuron (with a linear activation function). The 4 weights connecting the input neurons to the output are labeled $[0, 1, 2, 3]$.

References

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