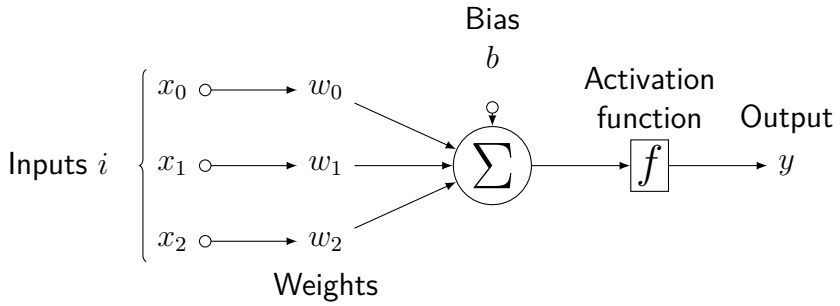
Computer Science 3B
Practical Assignment 03
2017-08-17

Deadline - 2017-08-17 17h00 Marks: 90

This practical assignment must be uploaded to eve.uj.ac.za **before** 2017-08-17 17h00. Late or incorrect submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student.

Good coding practices include a proper coding convention and a good use of commenting. Marks will be deducted if these are not present. See the reminder page for more details.

Previously we created a dynamic size ***perceptron***. Realistically the number of inputs of the network will not change over the lifetime. Instead we should to test a fixed-size perceptron.



$$s = b + \sum_{n=0}^{i} (x_n * w_n)$$

$$y = \begin{cases} 0, & \text{if } s < a \\ s, & \text{otherwise} \end{cases}$$

Write an 80x86 assembly program that will calculate the output of a single perceptron that has 5 inputs.

1. **Initialisation** - The program will first ask the user to initialise the perceptron's weights(w_n), bias(b) and activation value(a). The weights will be stored in an array.
2. **Testing** - The program will ask the user for the required inputs(x_n). The inputs will be stored in an array. All inputs for the perceptron must be captured **before** performing the required calculation.
3. **Calculation** - Calculate and display the weighted sum(s) and output of the perceptron(y).
4. **Repeat** - The program will ask the user if they wish to test another sequence. If the user responds positively then the program must repeat from the testing phase.

Note: Check the reminder page for submission details.

Test set1:

bias b - 1 **activation value** a - 100 **expected output** y - 175

x_0	1	w_0	25
x_1	2	w_1	20
x_2	3	w_2	15
x_3	4	w_3	10
x_4	5	w_4	5

Test set2:

bias b - 45 **activation value** a - 200 **expected output** y - 485

x_0	1	w_0	8
x_1	1	w_1	13
x_2	2	w_2	21
x_3	3	w_3	34
x_4	5	w_4	55

Mark sheet

1. Design [10]
2. Capture bias and activation value [05]
3. Capture weights into array [10]
4. Capture inputs into array [10]
5. Calculation using arrays [15]
6. Display result [05]
7. Repeat loop with exit condition [05]
8. Structure and layout (no temporary variables, correct data types) [05]
9. Commenting [05]
10. Correct execution. [20]

NB

Submissions which **do not assemble** will be capped at 40%!

Execution marks are awarded for a correctly functioning application and not for having some related code.

Reminder

Your submission must follow the naming convention as set out in the general learning guide.
Example

Surname	Berners-Lee	**Initials**	TJ
Student number	209912345	**Module Code**	CSC3B10
Current Year	2017	**Practical number**	P03

`Berners-Lee_TJ_209912345_ CSC3B10_2017_P03`

Your submission must include the following in a single zip (compressed) file:

- *Source file* (`asm` file) - File containing your solution. Your details must be included at the top of the source code.
- *Program design* (`pdf` file) - File containing your design. Your details must be included at the top of the design.