Practice 01 – basic neural networks

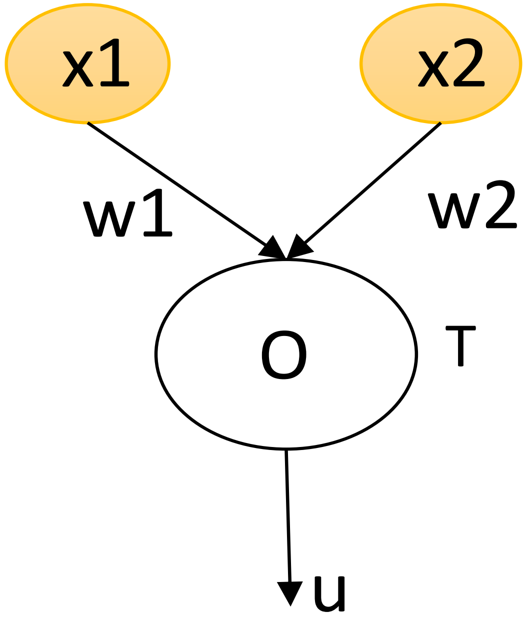
Matteo Rossi (matteo2.rossi@polimi.it)

Neuroengineering – Prof. Pietro Cerveri

A.Y. 2021/2022

Exercise 1

Let us consider the simple perceptron depicted in the figure with 2 binary inputs (x1 and x2), one output neuron with activation threshold S and signum activation function.



Questions:

a) What is the most likely goal for such kind of network?

b) Let assume that w1 = -1, w2 = 1, T = 0.5. In the input space, plot the separation line represented by the neuron activation highlighting the two different classifications.

c) Would be the network successfully exploited as a OR logic port?

Exercise 2

Train a perceptron with two inputs and a signum activation function (cfr. Exercise 1) using the batch updating strategy and the error correction rule (Rosenblatt). Inputs (X1 and X2) as well as targets (t) are summarized in the following table:

X1	X2	t
3	4	1
6	1	1
4	1	-1
1	2	-1

Initial weights and threshold: T $= 4.9,\ w_1 =\ -0.3;\ \ w_2 = 0.6,\ \eta = 0.05$

The threshold can be considered as an additional weight (w0) whose corresponding input (x0) is always = -1.

Exercise 3

Let us consider a perceptron (signum activation function) featuring two inputs (x_1 and x_2).

Questions:

a) You report the input-output relation of the perceptron for this specific setup

b) Let T=-0.8, w_1=-1.5 and w_2=2 be the threshold and the two weights, you define and draw the discrimination line laying onto (x_1, x_2) plane

c) Given that PP = [(1, 1); (1.5, -0.5); (0.5, 2); (-1.8, -1.5)] is a set a 4 input pairs of the network, you verify the classification results for each pair

d) Aiming at discriminating two classes (one for the first two input pairs and the other one for the second two pairs), you compute the opportune weight update to get the goal (assume a learning rate η of 0.5. Draw the new discrimination line

Exercise 4

Let us consider a perceptron with two inputs x1 and x2. The perceptron learning rule:

a) may not be applied with the paradigm of batch update

b) allows training the net to reconstruct a continuous function y = f(x1,x2)

c) can be applied with any step activation function

d) operates to orient the decision boundary along the direction parallel to weight vector

e) requires the computation of the derivative of the activation function