CMPG-765/CMPT- 465 Neural Networks and Learning Systems
Homework-3

1. (**undergrads - 60/100, graduates – 40/100**) Design a function utilizing a learning algorithm with the error-correction learning rule as it is described in the slide 36 of the Lecture-3 class notes. The function shall be robust, so it should not depend on the length of the input, the size of a learning set, and the inputs type (binary or real). Inputs and desired outputs should be transferred to the function as its arguments.
    - This function shall also have a flag-argument determining whether to generate starting weights randomly (in such a case starting weights should be generated inside the function as random numbers belonging to the interval [-0.5, 0.5]) or to start learning from the external weights (in the latter case a starting weighting vector shall be another calling argument, respectively).
    - The function shall also have an argument specifying a limit for the maximum number of the learning iterations to avoid falling in an infinite loop for input/output mappings, which are non-linearly separable.
    - The function shall return a weighting vector.

2. (**30/100**) Apply a function implementing the error-correction learning rule to the input/output mappings presented in slides 16 and 17 of the Lecture-3 class notes. You shall make two experiments:
    1) Start learning for both of the input/output mappings from random weights.
    2) For the input/output mapping presented in slide 17 start learning from the normalized Hebbian weights obtained in Homework 2. Compare the number of iterations for random starting weights and Hebbian starting weights.

3. (**Required for graduate students and extra credit for undergrads) (undergrads – 30 extra credit points, graduates – 30/100)**
    a) Run your program starting from the different random weights 5 times (you need to randomize the random numbers generator to ensure that different numbers are generated every time you call it) for both input/output mappings presented in slides 16 and 17 of the Lecture-3 class notes. Store your starting randomly generated weights.
    b) Considering Hebbian weights found in Project 2 for the input/output mapping from slide 16 as "ideal weights" compare a dot product of the Hebbian weighting vector and each of the starting random weighting vectors to a scalar (dot) product of the Hebbian weighting vector (which you found in Homework 2) and each of the weighting vectors resulted from the learning process with the error-correction learning rule. How they differ from each other for each experiment? Do you see any pattern followed from this comparison?

4. **(Both undergrads and graduates – 20 extra credit points)**. Compare the number of iterations required for the learning algorithm based on the error-correction learning rule applied to the input/output mapping from Homework 2 (see below) when the learning process starts from the random weights and from the Hebbian weights. Do you see any pattern followed from this comparison?

x_1	x_2	x_3	$f_1(x_1,x_2,x_3)$
0.5	1	0.5	**1**
0.5	1	-0.3	**1**
0.4	-0.5	0.4	**1**
0.4	-0.5	-0.5	**-1**
-0.3	0.7	0.5	**1**
-0.3	0.7	-0.4	**1**
-0.7	-1	0.3	**-1**
-0.7	-1	-0.5	**-1**

5. Write a brief report with your conclusions (**10/100**)
6. Turn in your source code, a screen shot of its test run and your report.