CMSC 409: Artificial Intelligence
Fall 2020, Instructor: Dr. Milos Manic, http://www.people.vcu.edu/~mmanic
Project 2

CMSC 409: Artificial Intelligence
Project No. 2
Due Thursday, Oct. 1, 2020, noon

Student certification:
Team member 1:
Print Name: ______________________________ *Date:* ________________
I have contributed by doing the following: ____
Signed: ____ *(you can sign/scan or use e-signature)*

Team member 2:
Print Name: ______________________________ *Date:* ________________
I have contributed by doing the following: ____
Signed: ____ *(you can sign/scan or use e-signature)*

Team member 3:
Print Name: ______________________________ *Date:* ________________
I have contributed by doing the following: ____
Signed: ____ *(you can sign/scan or use e-signature)*

Pr.2.1 Perceptron-based classifier (10 pts)
In this assignment you will use the datasets from Project 1.1. In language of your preference (Python, Java, Matlab, C++), implement a perceptron-based classifier that will iterate until the **total error** is:

	Data	Data_ALT
A	Epsilon $\varepsilon < 10^{-5}$	Epsilon $\varepsilon < 10^{-5}$
B	Epsilon $\varepsilon < 100$	Epsilon $\varepsilon < 200$
C	Epsilon $\varepsilon < 400$	Epsilon $\varepsilon < 700$

Note: epsilon stopping criterion is different for original and alternate data set.

To do this, you need to introduce a stopping criterion. You should also introduce a limit on maximum number of iterations (let that be *ni*=5,000). Normalize the datasets first. Initialize your neuron using random values between (-0.5, 0.5).

Please use unipolar version of:
- **a)** Hard activation function

b) Soft activation function

Note: For either activation function, you may need to experiment with different learning rates (alpha). Smaller alpha (especially with hard activation function) may yield better error. In addition to alpha, relative to soft activation function, the smaller gain may result in more iterations, but more "stable" convergence.

For the scenario **a)** do the following for each of the datasets.

1. Choose 75% of the data for training, and the rest for testing. Train and test your neuron. Plot the data and decision line for training and testing data (separately). Calculate errors for training and testing dataset. Show the Total Error (TE); Show the confusion matrix and compare to the one from Project 1.
2. Choose 25% of the data for training, and the rest for testing. Train and test your neuron. Plot the data and decision line for training and testing data (separately). Calculate errors for training and testing dataset. Show the Total Error (TE); Show the confusion matrix and compare to the one from Project 1.
3. Compare 1. and 2. Are errors different and if so, why? What is the effect of different data sets and effect of different training/testing distributions? When would you use option 1 and when option 2 above? Comment and discuss.

Repeat steps 1. through 3. for scenario **b)**.

Important: The data sets list the data points for both types of patterns ("small" car and "big" car). Think about which data points you should use for training and which for testing (i.e. algorithm will fail if trained on one type of patterns and tested on totally different one).

Pr. 2.2 Soft vs. hard activation function (5 pts)
Compare and discuss results when hard activation was used vs. when soft activation was used. Comment for each training/testing distribution and each data set.

This assignment may be updated!

Note:

1. The code must be user friendly. The TA must be able to test it simply by executing the code.
2. Project deliverable should be a zip file containing:
    a. Written report with answers to the questions above in word or pdf format.
    b. Training/testing data sets as decided in Pr. 2.1 steps 1 & 2
    c. Requested plots.
    d. Source code. Python code can be in Python notebook file (.ipynb) or Python file (.py).
3. Submit your zip file to Blackboard. Please name the zip file as GroupName_Project2.zip.