ECE 457B, Fall 2025, Assignment 2
Due: Fri, Oct 10, 11:59pm ET
Absolutely *no* extensions!

Submission:

> You do not submit any of your code for this assignment. Rather, you should submit a single pdf file with your responses to the problems in this assignment... which are based on observations from running code that you write/modify. So your submission instruction: submit your single pdf file to the appropriate dropbox on Learn.
>
> We may carry out plagiarism-checking on your submissions. You should credit all software/people as appropriate. It does not have to be elaborate. E.g., "Credit: Alice Wong, and ChatGPT."

The intent of this assignment is to get you to download and play with the code from `https://github.com/rasbt/machine-learning-book`, and exercise your understanding of (i) the Perceptron learning model, and, (ii) the ADAptive LInear NEuron (Adaline).

0. (0 points) Download the code from `https://github.com/rasbt/machine-learning-book`. You can either `git clone`, or simply download a `.zip`. Get `ch02/ch02.py` working. By "get it working", I mean that you should be able to run it, and have it generate, in turn, the 8 pictures from your textbook, Figure (2.6)–(2.8), (2.11), (2.14)–(2.15).

For Problems (1)–(2) below, you should isolate the code for the Perceptron in `ch02.py`. This should be between Lines 1 and approximately 300 in `ch02.py`. For Problems (3)–(4) below, you should isolate the code for Adaline in `ch02.py`. This is a bit trickier because you presumably want to retain the parts of the code that read the data files in.

Each problem (1)–(4) below is worth 25 points. Parts (a) and (b) of Problem (2) are weighted equally.

1. Learning boolean functions that are linearly separable as Perceptrons.

   We want to check whether 2-input boolean functions that are linearly-separable can be learned using the Perceptron learning rule. For each of the following, adopt as initial values $w = [0.25 \quad -0.125], b = 0$ and the learning rate η to 0.1. Also generate input files — there would be four lines each only, for each of the following three boolean functions. Then report: (i) whether you achieve convergence over 10 epochs, (ii) the number of epochs it takes to converge for each of the following boolean functions, (iii) the final values for w, b, and, (iv) whether those final values for w, b provide a correct boundary between the 0's and 1's.

   The three 2-input boolean functions are:

   - $f(x_1, x_2) = x_1$ AND x_2
   - $f(x_1, x_2) = x_1$ OR x_2
   - $f(x_1, x_2) = x_1$ IMPLIES x_2

2. Learning an axis-aligned rectangle as a Perceptron:

   There is a file rectangle.data in Learn which has 100 lines, each line a point in $\mathbb{R}^3$ and a bit indicating whether the point is within an axis-aligned, three-dimensional rectangle.

   (a) Change the code for your Perceptron to adopt this as your input samples. Adopt $w = [0.25 \;\; -0.125 \;\; 0.0625], b = 0$ as your initial values, and use the same learning rate $\eta = 0.1$ as above. Now, run the fit method for (i) 10, (ii) 20, (iii) 30, and (iv) 100 epochs. Write a brief paragraph as to what you observe. Specifically comment on whether we are appearing to converge as we increase the number of epochs.

   (b) Suppose, you adopt the same initial values for w, b, and learning rate η as above. First train on the first 60 samples from rectangle.data only, over 10 epochs. Then, adopt $\epsilon = 0.2, \delta = 0.25$. Now, splitting the remaining 40 samples into 8 groups of 5 each, check whether you achieve an error proportion of at most ϵ on at least $1 - \delta$ of the 8 groups. Report your numbers, and state whether you do or do not.

The intent with Problems (3)–(4) is to repeat the exercises (1)–(2) above, but this time using the Adaline model for learning, without any feature scaling.

3. With the same initial values for w, b as in Problem (1), learn the three boolean functions AND, OR and IMPLIES using the Adaline model. Try four different values for the learning rate, η: $0.01, 0.05, 0.1, 0.5$ and 0.75. Use the same threshold of 0.5 in predict() that comes with the code to separate 0 from 1, and adopt 10 for the number of epochs.

   Write what you observe. Specifically, what learning rate(s) seem to work well? Does the mean squared error appear to converge towards 0 for any of the learning rates? For which ones does it increase as epochs elapse? For what learning rates do the resultant w, b values correctly predict the output bit for all the four values of the input bits?

4. Repeat the rectangle learning from Problem (2)(a) over all 100 samples in rectangle.data, but this time using Adaline. Use the same initial values for w, b, adopt a learning rate of $\eta = 0.01$, and learn over 10 epochs.

   What do you observe? What is the evolution of the mean squared error over the 10 epochs? How good are the resultant w, b from the standpoint of correctly predicting the class of the training samples?