

Intro to Deep AI - Fall 2023
Homework 4
Perceptron and Logistic Regression
Submission Due date: Saturday, Dec 8th, 11:59 pm

Description

You are asked to write a program with two functions that apply perceptron or logistic regression techniques to the input and print their outcome.

Perceptron

The first function receives n (<100) triplets of x_1 , x_2 , y from the user, and returns the value for w (weight) that the perceptron algorithm computes to classify the inputs. Here x_1 and x_2 show the input features for each sample, and y shows the class. Consider a binary classification with two possible values for y : -1 and $+1$. Use the same format for input and output as the example below. Note that before the actual input, another input character P will determine the call for the perceptron function.

Example Input:

P (0, 2,+1) (2, 0, -1) (0, 4,+1) (4, 0, -1) *#P indicates perceptron*

Example output:

-2.0, 0.0 *# referring to $w=[-2.0, 0.0]$*

Use the same procedure for updating w , as discussed in slides ($w = w + y \cdot f$). Start from $w=[0,0]$, and update w by a maximum of $n*100$ times, where n is the number of input samples (100 times iterating over all of the input samples).

Logistic regression

The second function receives a similar input as described above with the only difference in the first input character (L instead of P). The desired task will still be binary classification. The output, in this case, will be printing the probability values that logistic regression computes for each input belonging to the positive class. Set alpha (learning rate) equal to 0.1.

Example Input:

L (0, 2,+1) (2, 0, -1) (0, 4, -1) (4, 0, +1) (0, 6, -1) (6, 0, +1) *#L indicates*

Logistic Reg.

Example output:

0.29 0.71 0.14 0.86 0.06 0.94

For the logistic regression use the basic procedure introduced in class ($w = w + \alpha \cdot \nabla g(w)$). Start from $w=[0,0]$, and update w by a maximum of $n*100$ times, where n is the number of input samples (100 times iterating over all of the input samples). Note that when $g(z)=\text{sigmoid}(z)$, we will have $g'(z)=g(z)(1-g(z))$. This is also shown on the Common Activation Functions slide. Not necessarily needed here, but for this case, you can further simplify this calculation ([see here](#), note the end of the doc).

Submitting Instructions and Grading

Follow the same procedure for accessing the homework on Moodle server as described in [Homework 1](#). Look for the “Alpha Beta Search” homework. The grading policy will be also similar (i.e., 10 test cases, and 10 points for each correct answer). The grade you see on Moodle will be a proposed grade. Your submission can be manually checked. **Partial credit (up to 50 points)** will be given to programs that implement the algorithms logically, run, and generate output, but do not pass the test cases.

Integrity policy

Note that we will check for similarities between the submissions (including the previous submissions and manually submitted ones by the instructors). To ensure our course remains fair to everyone, we will proactively and carefully check for any suspicious pattern that violates our course policy. As we have done this in the past, we will certainly escalate and report any violation to the appropriate contacts at UD. Sharing or using a shared code is prohibited. Our course policy is simple: **your code, in its entirety, MUST be yours**. You can talk to others about the algorithm(s) to be used to solve a homework problem; as long as you then mention their name(s) on the work you submit. Borrowing a few lines of code is fine (say from StackOverflow), but again, you must acknowledge the source.

Questions?

Post them on [Ed](#).

Some Other Examples

Input:

L (0, 2,+1) (2, 0, -1) (0, 4,+1) (4, 0, -1)

Output:

0.98 0.02 1.0 0.0

Input:

L (0, 2,+1) (2, 0, -1) (0, 4, -1) (4, 0, +1) (0, 6, -1) (6, 0, +1)

Output:

0.29 0.71 0.14 0.86 0.06 0.94

Input:

P (0, 2,+1) (2, 0, -1) (0, 4, -1) (4, 0, +1) (0, 6, -1) (6, 0, +1)

Output:

0.0, -2.0

Input:

P (0, 2,+1) (2, 0, -1) (0, 4,+1) (4, 0, -1)

Output:

-2.0, 0.0

Input:

L (8, 12,+1) (2, 4, -1) (0, 5, -1) (2, 9, +1) (4, 7, -1) (5, 3, +1)

Output:

0.92 0.64 0.46 0.6 0.76 0.84

Input:

L (8, 12,+1) (-3, 6, -1) (0, 18, -1) (11, 15, +1) (9, 7, -1) (10, 8, +1)

Output:

1.0 0.0 0.01 1.0 1.0 1.0

Input:

L (8, 2,+1) (-3, 6, -1) (0, 18, -1) (11, 1, +1) (9, 7, -1) (10, 2, +1)

Output:

0.99 0.0 0.0 1.0 0.01 1.0

Input:

L (6, 2,+1) (7, 6, -1) (10, 18, -1) (11, 1, +1) (9, 7, -1) (10, 2, +1)

Output:

0.97 0.01 0.0 1.0 0.01 1.0

Input:

L (16, 2,+1) (7, 6, -1) (8, 5, -1) (11, 1, +1) (9, 7, -1) (10, 2, +1)

Output:

1.0 0.0 0.01 1.0 0.0 0.99

Input:

P (1, -2,+1) (-2, 3, -1) (2, 9,+1) (4, 8, -1)

Output:

-8.0, -6.0

Input:

P (3, -12,+1) (3, 11, -1) (5, 19,+1) (4, 12, -1) (5, 20, +1) (3, 2, -1)

Output:

6.0, 1.0

Input:

P (2, 10,+1) (-3, 2, -1) (3, 9,+1) (14, 1, -1) (15, 0, +1) (7, 6, -1) (5, 12, +1) (4, 22, -1)

Output:

-2.0, -10.0