Perceptrons

Exercise 1
Construct by hand a Perceptron which correctly classifies the following data; use your knowledge of plane geometry to choose appropriate values for the weights w_0, w_1 and w_2.

<u>**Training Example**</u>	x_1	x_2	<u>**Class**</u>
a.	0	1	-1
b.	2	0	-1
c.	1	1	+1

Identify the equation of the line and the weights of the Perceptron (including bias).

Answer
- the gradient must be $\frac{1}{2}$ because that's the slope between the two negative points. Any other gradient would give poorer linear separability for the negative data points.
- The y intercept (or x_1 intercept) is $\frac{5}{4}$ because that is the point that maximises linear separability between positives and negatives for this data set.
- In theory, any y intercept > 1 and $< \frac{3}{2}$ would satisfy this requirement, however they would all be sub optimal compared to the two points above as they would reduce linear separability for one or the other class.

Given the points above we can make the following calculation:

$$\begin{aligned} y &= mx + b \\ x_2 &= \frac{5}{4} - \frac{x_1}{2} \\ 1x_2 &= \frac{5}{4} - \frac{1}{2} \\ 4x_2 &= 5 - 2x_1 \text{ multiply both sides by 4} \\ 2x_2 + 4x_2 - 5 &= 0 \text{ rearrange terms to one size} \end{aligned}$$

Therefore we can conclude that the weights are:
$w_0 = -5$

$w_1 = 2$

$w_2 = 4$

Exercise 2:
Demonstrate the Perceptron Learning Algorithm on the above data, using a learning rate of 1.0 and initial weight values of:

$$w_0 = -1.5$$
$$w_1 = 0$$
$$w_2 = 2$$

The first three steps are shown below. You should continue until all items are correctly classified.

Iteration	w_0	w_1	w_2	Item	x_1	x_2	Class	$s = w_0 + w_1x_1 + w_2x_2$	Action
1	-1.5	0	2	a	0	1	-1	+0.5	Subtract
2	-2.5	0	1	b	2	0	-1	-2.5	None
3	-2.5	0	1	c	1	1	+1	-1.5	Add

Answer

Iteration	w_0	w_1	w_2	Item	x_1	x_2	Class	$s = w_0 + w_1x_1 + w_2x_2$	Action
1	-1.5	0	2	**a**	**0**	**1**	**-1**	+0.5	Subtract
2	-2.5	0	1	**b**	**2**	**0**	**-1**	-2.5	None
3	-2.5	0	1	**c**	**1**	**1**	**+1**	-1.5	Add
4	-1.5	1	2	**a**	**0**	**1**	**-1**	+0.5	Subtract
5	-2.5	1	1	**b**	**2**	**0**	**-1**	-0.5	None
6	-2.5	1	1	**c**	**1**	**1**	**+1**	-0.5	Add
7	-1.5	2	2	**a**	**0**	**1**	**-1**	+0.5	Subtract
8	-2.5	2	1	**b**	**2**	**0**	**-1**	+1.5	Subtract
9	-3.5	0	1	**c**	**1**	**1**	**+1**	-2.5	Add
10	-2.5	1	2	**a**	**0**	**1**	**-1**	-0.5	None
11	-2.5	1	2	**b**	**2**	**0**	**-1**	-0.5	None
12	-2.5	1	2	**c**	**1**	**1**	**+1**	+0.5	None

Explanation: Changes to weights marked in yellow.
at epoch 1:

- it1 is a false positive and needs subtraction. w_1 is updated from 0 to 1
- it2 is a true negative. But w_1 is updated from 0 to 1 anyway
- it3 is a false negative and needs addition. w_1 is updated from 0 to 1

At epoch 2:

- it4 is a false positive and needs subtraction. w_1 is updated from 1 to 2
- it5 is a true negative. But w_1 is updated from 1 to 2 anyway.
- it6 is a false negative and needs addition. w_1 is reduced from 1 to 0. w_0 increased from 2.5 to 3.5

At epoch 3:

- it7 is a false positive and needs subtraction. w_1 is reduced from 2 to 1. w_0 increased from 1.5 to 2.5

- it8 is a false positive and needs subtraction. w_1 is reduced from 2 to 1 and w_2 is increased from 1 to 2
- it9 is a false negative and needs addition. w_1 is increased from 0 to 1 and w_2 is increased from 1 to 2. w_0 reduced from 3.5 to 2.5

At epoch 4:

- it10 is a true negative. No change required
- it11 is a true negative. No change required
- it12 is a true positive. No change required

I don't understand the logic behind the decision on whether to adjust, and if the decision is make an adjustment, which weight to adjust. In particular iteration 2 & 5. I thought that if no change was required we didn't change. And yet in these two iterations we make a change to the weights.

- In the case of iteration 2, the only explanation I can think of is that the result (-2.5) is too negative and we want the negative number to be closer to zero. Lets accept that hypothesis for now.
- In the case of iteration 5, I really don't understand why we change it, because -0.5 is the final answer in iteration 11. There seems to be no need to make any change and yet we do.

Notes from the Video

- The decision about how to change w_0 depends on the initial state of the Perceptron. We change it by +/- η.
- The decision about how to change the weight depends on the x value and the target output.
- If $g(s)$ is 0 and we want it to be 1, then we want to reduce s. If x is negative, we use a negative weight. Conversely, if x is positive we use a positive weight. We want the product to increase. The easiest way to achieve that is to multiply x_1 by η and add that to w_1.
- The reverse is true for If $g(s)$ is 1 and we want it to be 0, we do the opposite.

How might this change the answer for question 2:

I think there is a more direct route to achieve the same outcome, which I try to demonstrate below and in the attached ipynb

Iteration	w_0	w_1	w_2	Item	x_1	x_2	Class	$s = w_0 + w_1x_1 + w_2x_2$	Action
1	-1.5	0	2	**a**	**0**	**1**	**-1**	+0.5	Subtract
2	-2.5	0	1	**b**	**2**	**0**	**-1**	-2.5	None
3	-2.5	0	1	**c**	**1**	**1**	**+1**	-1.5	Add
4	-2.5	0	2	**a**	**0**	**1**	**-1**	-0.5	None
5	-1.5	0	1	**b**	**2**	**0**	**-1**	-1.5	None
6	-1.5	0	1	**c**	**1**	**1**	**+1**	-0.5	Add
7	-2.5	0	2	**a**	**0**	**1**	**-1**	-0.5	None
8	-0.5	0	1	**b**	**2**	**0**	**-1**	-0.5	None
9	-0.5	0	1	**c**	**1**	**1**	**+1**	+0.5	None