## horizontal line

How to train a Perceptron ? (Class4)

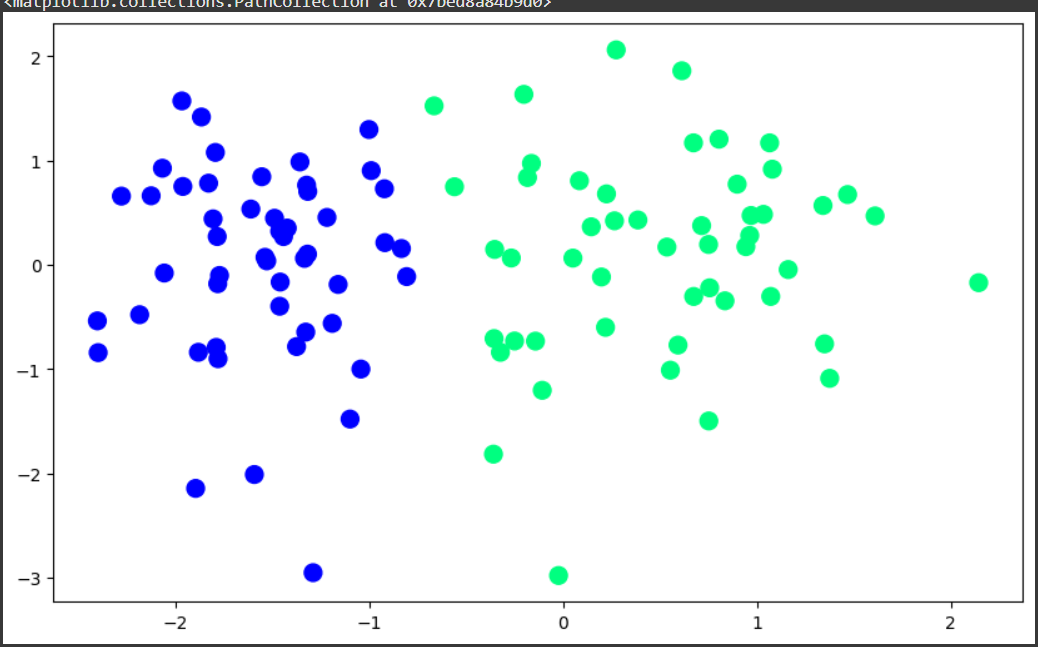
06.01.2025

[Perceptron trick collaboratory file](https://colab.research.google.com/drive/19JfVPk6LDYL6_z0AZEY_W2OMmcUZzzIM)

# Perceptron Trick

The equation of the line is of the form Ax+By+c = 0

Now see, we have various data points of the dataset having features X1 , X2 and a target variable ‘Y’.



Now the equation is W0X0 + W1X1 + W2X2 = 0 .

First we decide the values of W0 ,W1, W2 randomly as [1,1,1].

We randomly select any point and put its coordinates in X0, X1 , X2.

As we have only X1 and X2 we have inserted one new column as X0 all have values 1 for all rows. This is the ‘bias’.

NOW , to classify the points there requires a line and we need to find out the best fitted line that classifies all the points .

We will start with a random line , W0, W1, W2 = [1,1,1] and any random data point coordinates [ X0,X1,X2 ].

If model gives WiXi >=0 means student is placed

or WiXi <0 means not placed .

Now we will compare the model’s predictions with the Actual ones . If they are the same it means that the data point is truly classified by the line , no need to transform it .

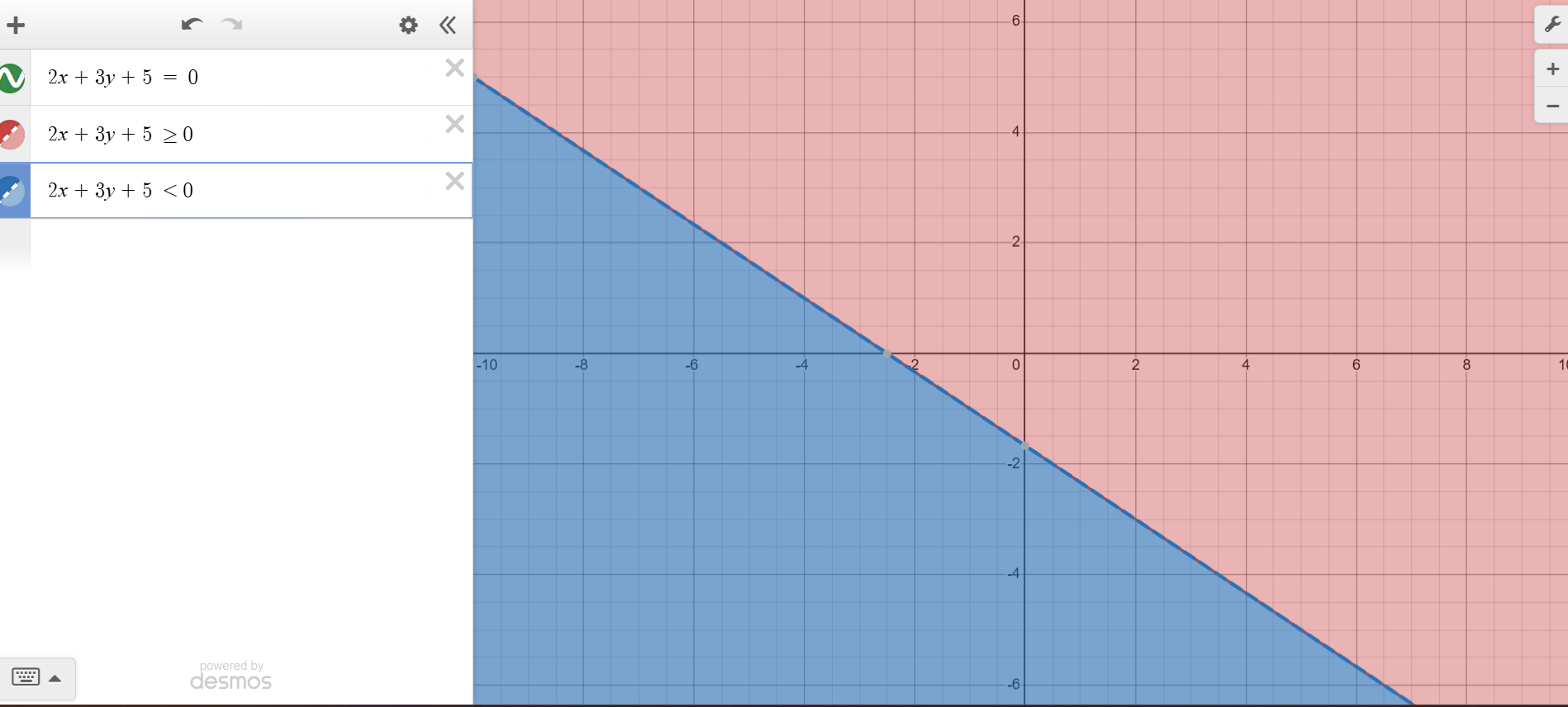
But , if it doesn't match it means the line misclassified it and the line transformed in such a way that the point is classified now.

This goes for all points one by one (i.e. epochs) thus in this way the line transforms and we get the one best fitted.

# How to Label Regions ?

Ax + By + c >=0 is the positive region.

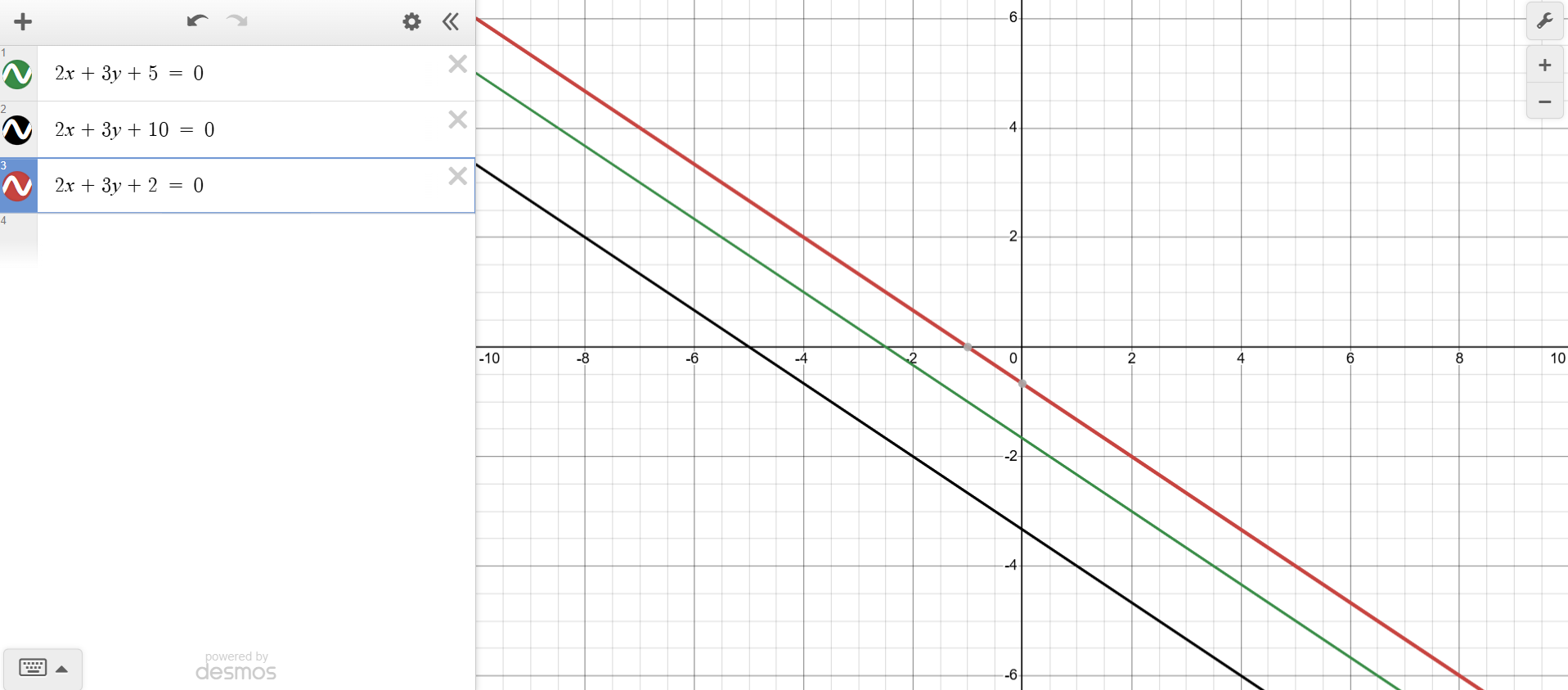
Ax + By + c < 0 is the negative region.



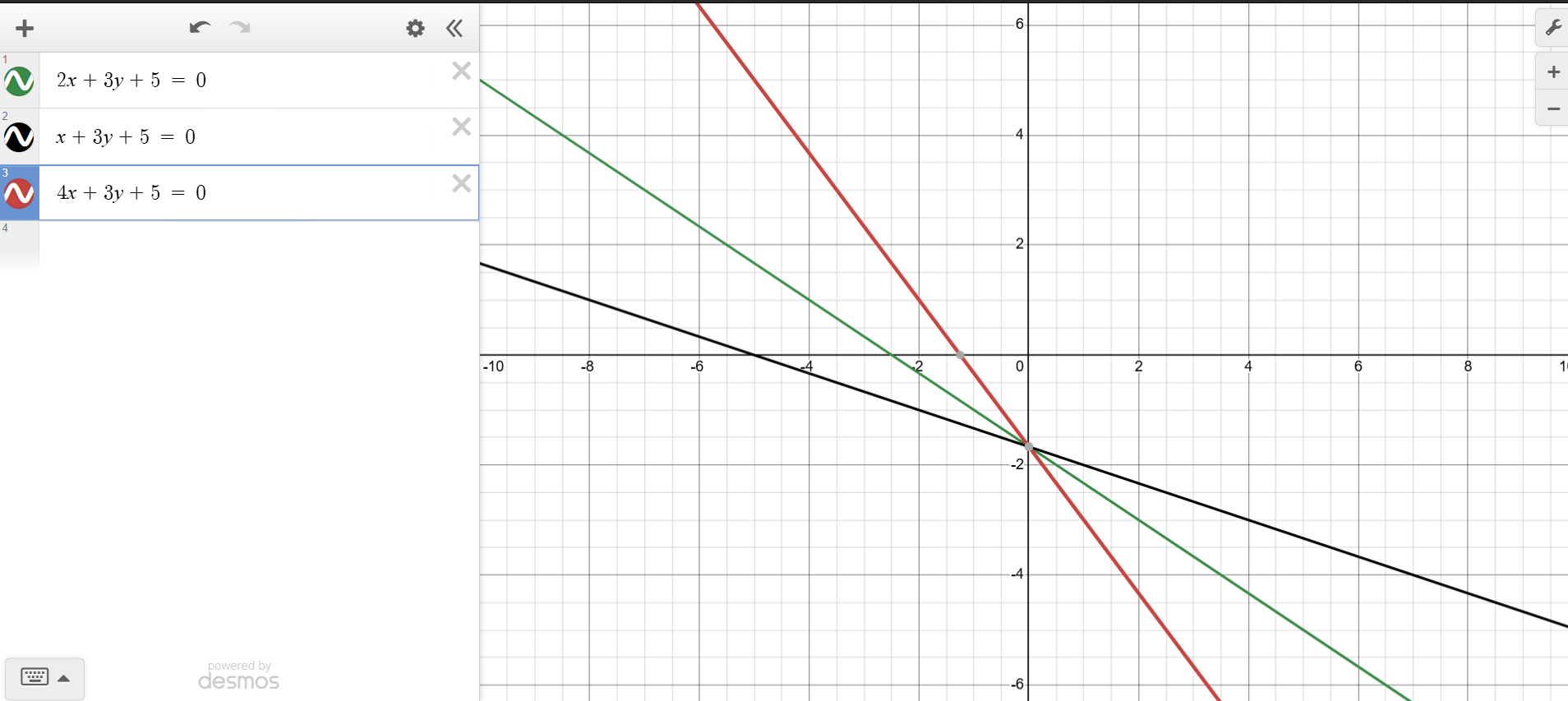
# Transformations

Ax+ By +c = 0

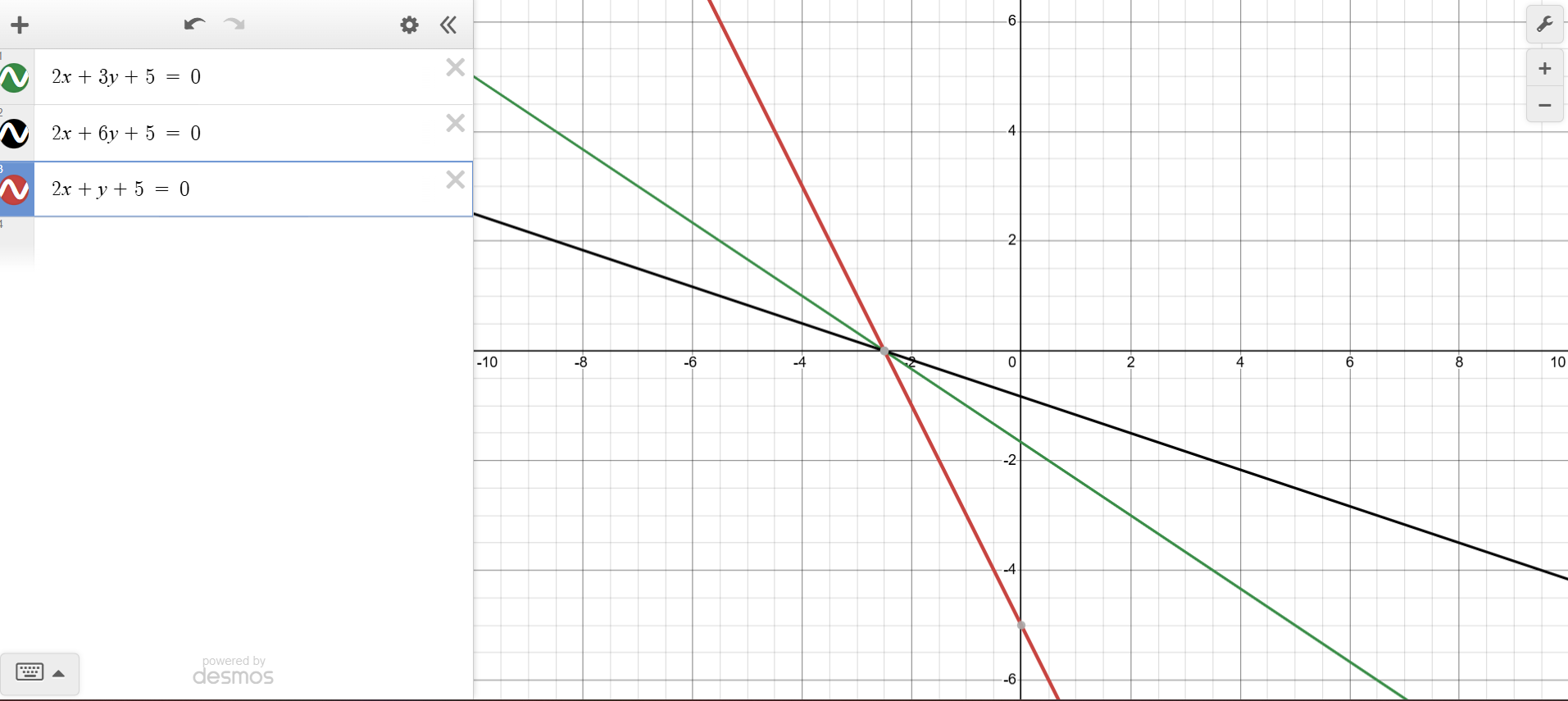
On changing C , the line transforms parallely.



On changing A , the line transforms along the y-axis .



On changing B , the line transforms along x-axis.



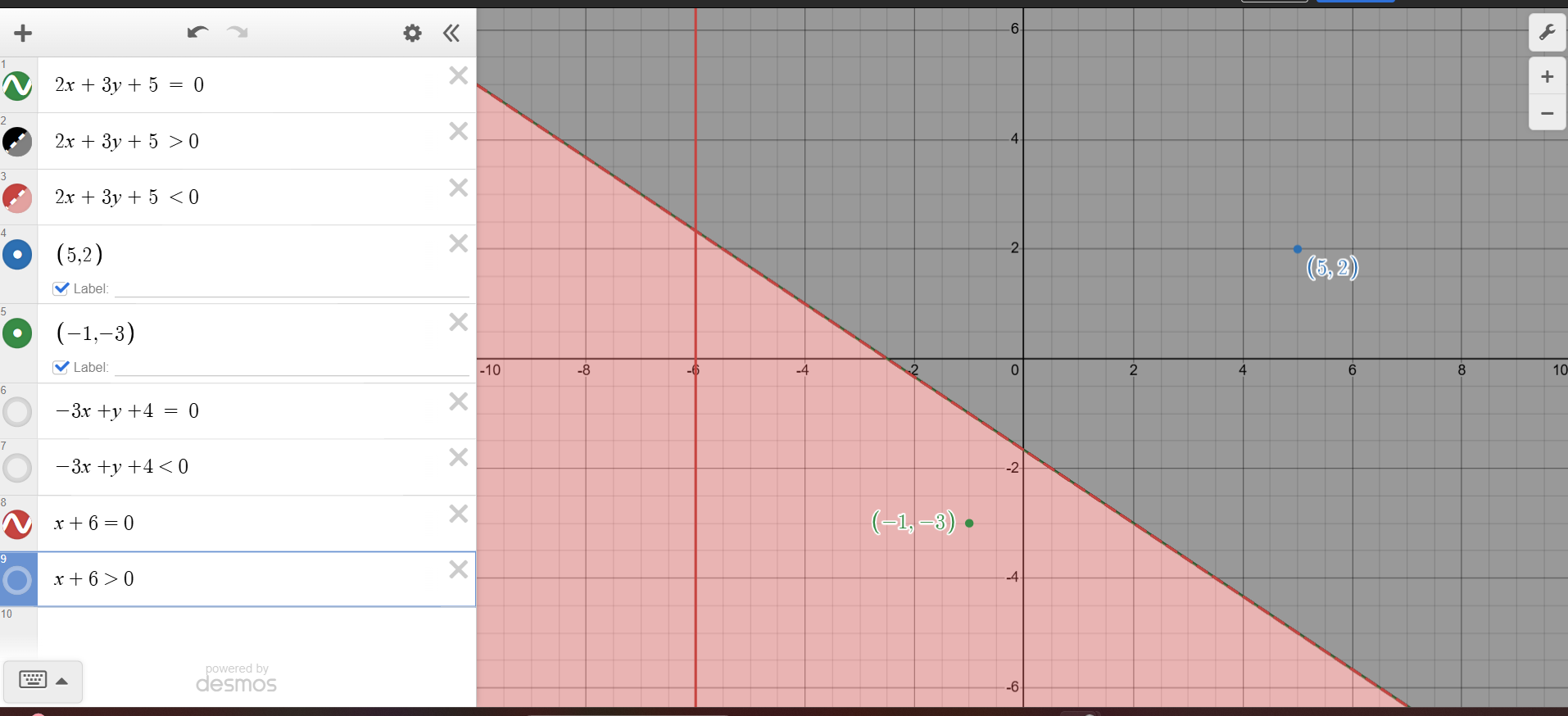
As you can see there are 2 misclassified points 👍

Case 1 : The red point (-ve point) on the green side (+ve side).

Case 2 : The green point (+ve point) on the red side (-ve side).

In 2 cases we do nothing when model predicting +ve , actual is +ve

And model predicting -ve , actual is negative.



But these misclassifications ,

For Case 1 : Take line coefficients and from it subtract the coordinates of point . That will be the new coefficients of the line .

E.g. line is 2x + 3y + 5 = 0 and the coordinate of the point is (5,2).

You will simply add the 3rd coordinate as 1 (bias).

2 3 5

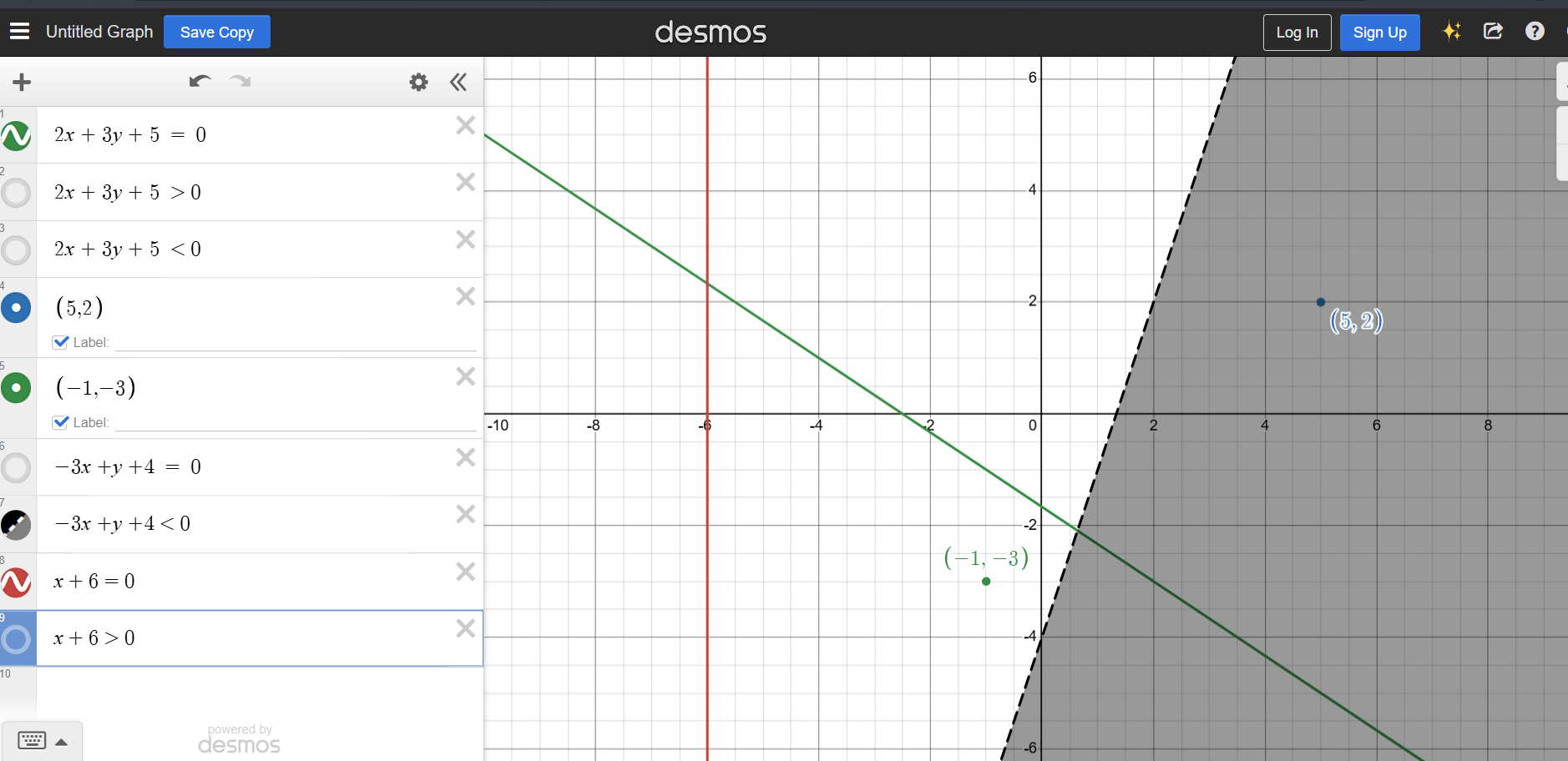
5 2 1

—\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-3x +y +4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ So the new line will be , -3x +y +4 = 0

This implies ,  **Wnew = Wold -**



For Case 2 : simply add them .

E.g. line is 2x + 3y + 5 = 0 and the coordinate of the other point is (-1,-3).

2 3 5

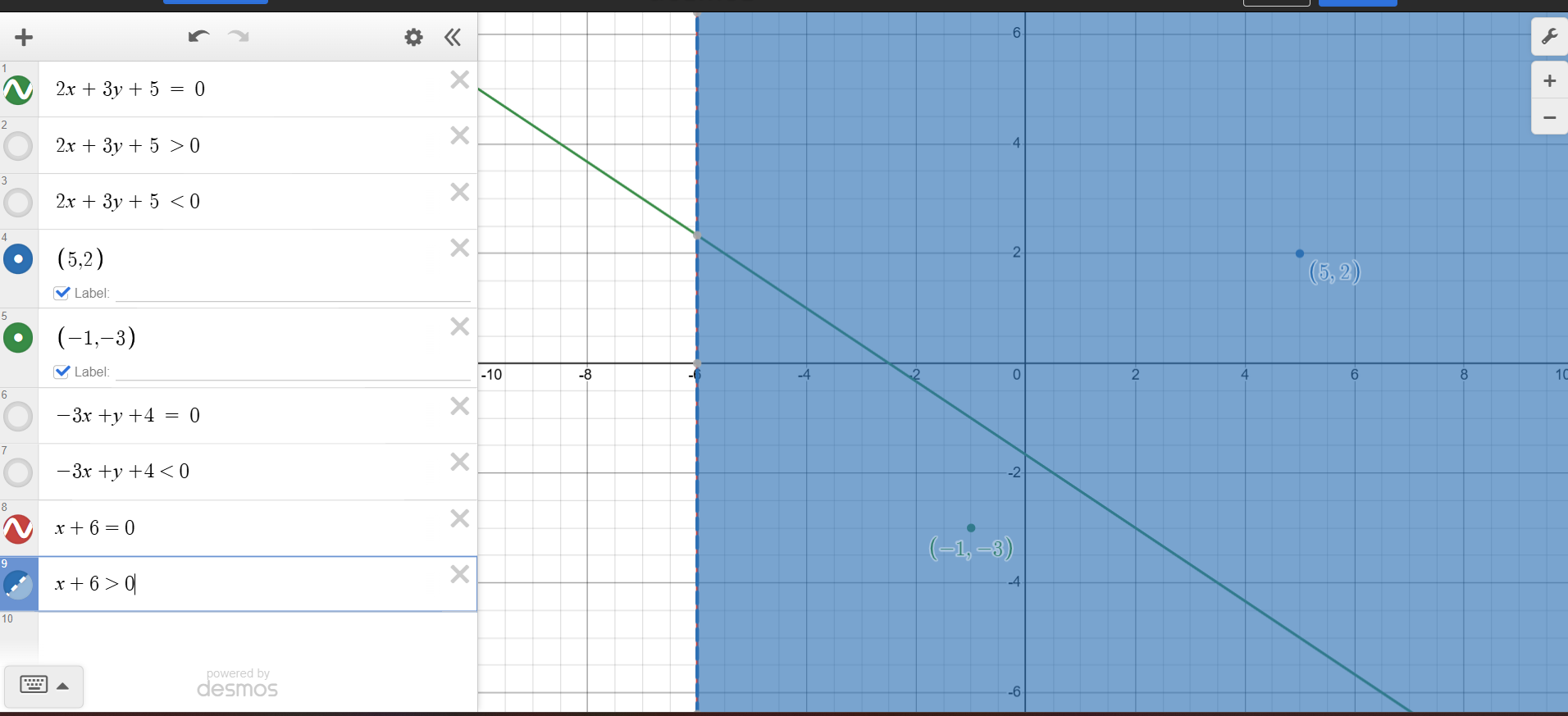
-1 -3 1

+\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1x +0y +6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ The line for this point is , x+ 0y +6 =0.

This implies , **Wnew = Wold +**



) — This is the learning rate . used to reduce the high transformation at large angles .

# Algorithm

Epoch = 1000 , = 0.01

For i in range (epoch) :

Randomly select a student

Initialize weights with [1,1,1]

If (Yi N) and WiXi >=0 :

Wnew = Wold -

If (Yi P) and WiXi <0 :

Wnew = Wold +

# Simplified Algo

For i in range (epoch) :

Randomly select a student

Initialize weights with [1,1,1]

Wnew = Wold -

| Yi | Yi(hat) | Yi - Yi(hat) | Impact |
| --- | --- | --- | --- |
| 1 | 1 | 0 | No changes (no line transformation) |
| 0 | 0 | 0 | No changes (no line transformation) |
| 1 | 0 | 1 | Case 2 |
| 0 | 1 | -1 | Case 1 |

# Practical Code Demo

