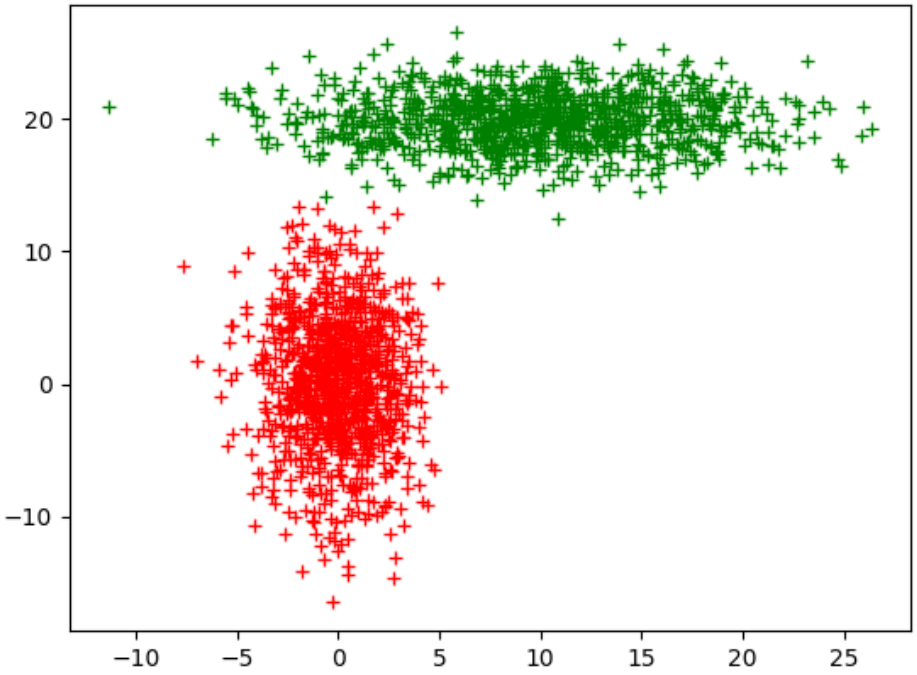
Deep Learning and Computer Vision

Training a perceptron

In this exercise, the aim is to code and train a single neuron network on a binary classification problem.

We have 1000 samples for each category, distributed in a 2D space as illustrated in the figure below.



Distribution of the training samples. Green points are from one category and red points are from the second category.

You have to start from the provided code that draws the data from two Gaussian distribution.

1/ Create a data numpy array containing the (shuffled) 2000 points with their labels (1 or 0).

2/ Create one function that makes the prediction from one point (2D) and one weight vector (3D). The activation function is a step function with threshold at 0. For testing, you can just put random values in the weight vector.

3/ Create a second function that updates the weights from the current weight vector and one point (2D vector + label). The learning rate is also one input of the function. Use the update rule for a perceptron, studied in class.

4/ Train the network on the training data for several epochs.

5/ Check the train accuracy after each epoch.

6/ Draw the learned linear separator in the 2D space.

7/ Test the trained perceptron on test data. This test data can be drawn with the same Gaussian distributions used to draw the training set.