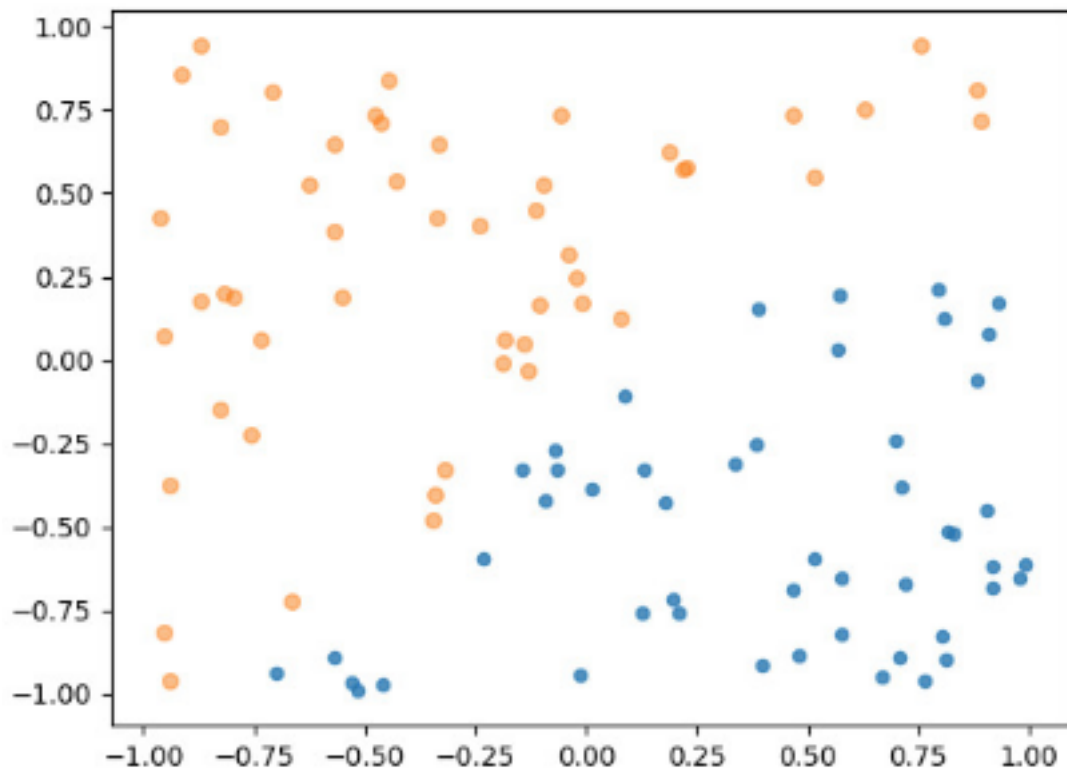


## Perceptron / Neural Networks

### Part 1



### Part 2

We are going to make an analysis about the training with different parameters. This analysis will be based on the OR function.

There are three variables that are going to be modified to realize the analysis. Inside the code, those variables are: `initial_w` (the initial weight), `learning_rate` (how fast the learning rate advance), and `n_iterations` (how many times it will learn).

The initial setup is the following:

`Initial_w = 0.1`

`Learning_rate = 0.03`

`n_iterations = 5`

Given that input, the result is the following:

```
[0.1, 0.1, 0.1]
[0.1, 0.1, 0.1]
[0.1, 0.1, 0.1]
[0.1, 0.1, 0.1]

[0.1, 0.1, 0.07]
[0.1, 0.1, 0.07]
[0.1, 0.1, 0.07]
[0.1, 0.1, 0.07]

[0.1, 0.1, 0.04000000000000001]
[0.1, 0.1, 0.04000000000000001]
[0.1, 0.1, 0.04000000000000001]
[0.1, 0.1, 0.04000000000000001]

[0.1, 0.1, 0.010000000000000009]
[0.1, 0.1, 0.010000000000000009]
[0.1, 0.1, 0.010000000000000009]
[0.1, 0.1, 0.010000000000000009]

[0.1, 0.1, -0.01999999999999999]
[0.1, 0.1, -0.01999999999999999]
[0.1, 0.1, -0.01999999999999999]
[0.1, 0.1, -0.01999999999999999]

0.0
1.0
0.0

[0.3, 0.3, 0.3]
[0.3, 0.3, 0.3]
[0.3, 0.3, 0.3]
[0.3, 0.3, 0.3]

[0.3, 0.3, 0.0]
[0.3, 0.3, 0.0]
[0.3, 0.3, 0.0]
[0.3, 0.3, 0.0]

[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]

[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]

[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]
[0.3, 0.3, -0.3]

0.0
1.0
0.0
```

The first alteration is the following

Initial\_w = 0.3

Learning\_rate = 0.3

n\_iterations = 5

Until this point, even changing the initial\_w and learning\_rate to higher numbers (up to .8), we realize that since it is simple, the answers will keep being correct. In order to make it fail, we have to adjust the n\_iterations.

In the following setup, the program fails

Initial\_w = 0.8

Learning\_rate = 0.8

n\_iterations = 1

```
[0.8, 0.8, 0.8]
[0.8, 0.8, 0.8]
[0.8, 0.8, 0.8]
[0.8, 0.8, 0.8]

0.0
1.0
1.0
```

When the problem is not hard, the `n_iterations` must be lower in order to fail, since the `initial_w` and `learning_rate` will be enough for a good result.

*Explanations as to what are ANNs good for.*

ANN are good for solving problems in the same way the human brain could do it. They can be used to infer a function from observation.

*Where would you use them?*

We would use them to solve problems of classification, data processing, regression analysis

*Are they worth the effort implementing or not?*

Yes, in problems like computer vision, solving them using regular programming is harder than using ANNs since they learn how to solve the problem.

*What kinds of problems do they not solve?*

When you want to understand the underlying process of a given problem, the ANNs won't work since there is not many information about the actual process inside the neural network