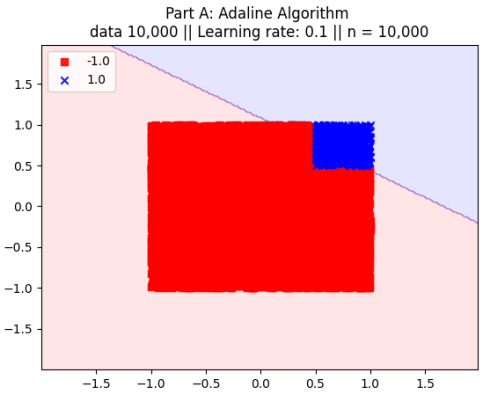
Back- propagation Algorithm – NN

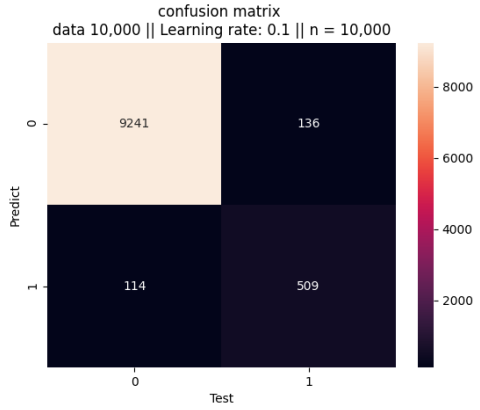
We create two-dimension random data <x, y> in size 1,000, with range (-1, 1).
Then classified each sample to "1" or "-1" such that if x and y both bigger than 0.5 they get the value "1", if-else, "-1".

The MLPClassifier from SKlearn use Feed-Forward Networks and Backpropagation,
with given data and desired output, the MLPClassifier train the model and change the weight and bias accordingly.
you can learn all about MLPClassifier Here.

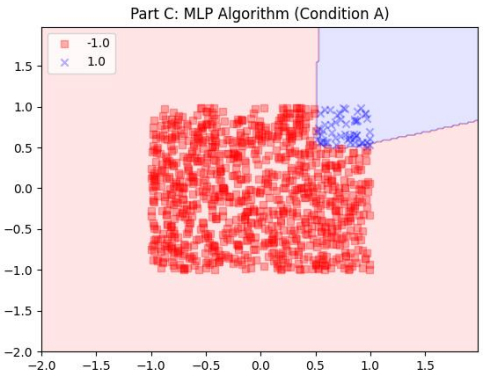
Part C:
In this part we will use data from parts A and B and try to improve our result by using Backpropagation (the back is our model prediction):

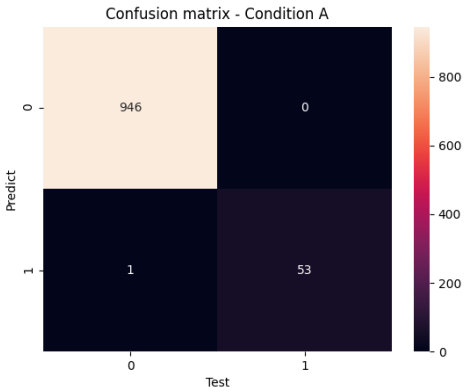
Adaline (our result: **97.5%**):





MLPClassifier (our result: **99.9%**):



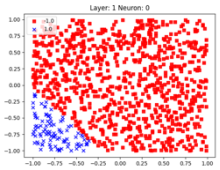


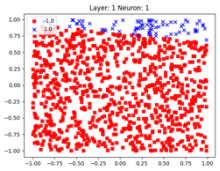
We can see that the result matches more accurately with MLP because the data is not linearly separate.

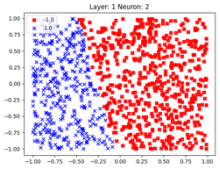
Now let's look at the second problem that points have value 1 only if
1/2 <= x**2 + y**2 <= 3/4, to answer this problem we use MLP with 2 hidden layers such

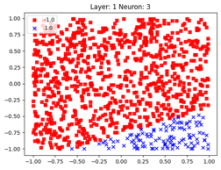
that the first layer has 8 neurons and the second has 2.
our final result was **96.4%.**
The output of etch neuron is as follow (etch point <x, y>, blue =1, red = -1):

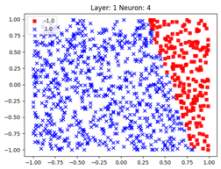
layer 1

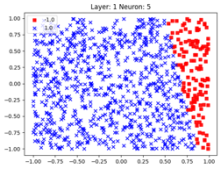


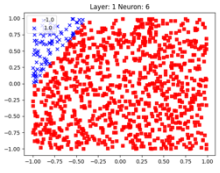


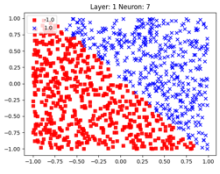




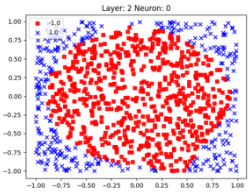


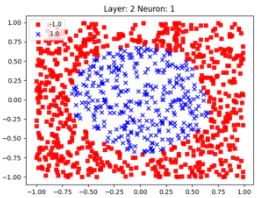




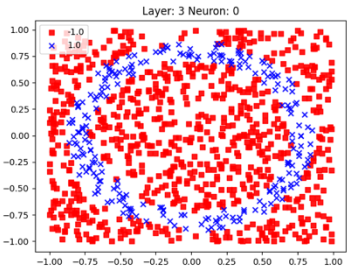


layer 2



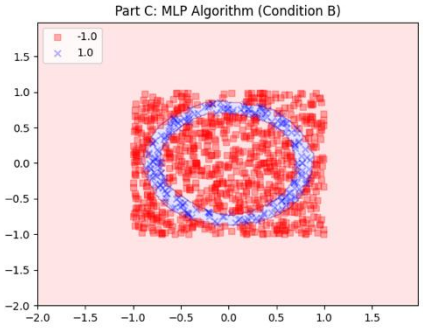


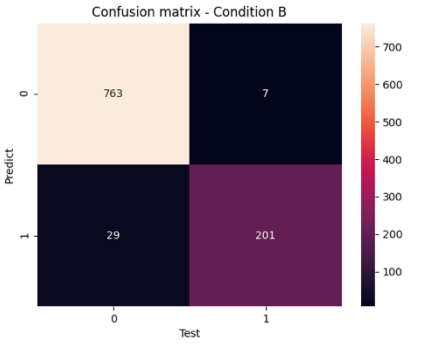
output layer



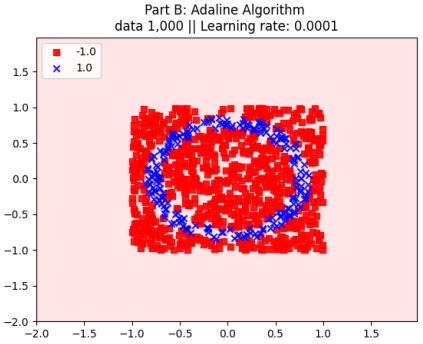
Lat's look at the final result of MLP vs Adaline (train and test data), our model prediction is at the back and the 'dots' are the train data:

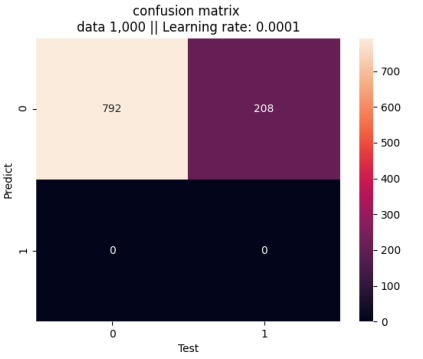
MLP





Adaline





It can be seen that Adaline always predict -1 and he's right at **79.2%,**
On the other hand, the MLP model succeed to predict the circle almost exactly(!) and he's right at **96.4%.**

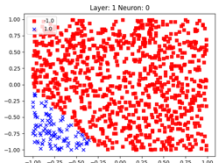
(this is the same samples, but we had to make the MLP plot brighter so that the circle could be seen)

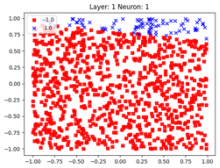
Part D:

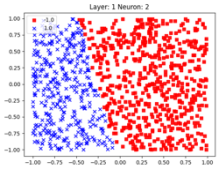
In this part, we will use the trained neurons from the next to last level of Part C as input and only an Adaline for the output. That is, we will give Adaline the output of the neurons from Part C in the level below the output, and train only the Adaline.

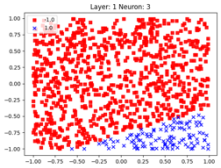
Now let's look at the second problem that points have value 1 only if:
1/2 <= x**2 + y**2 <= 3/4.
the output of etch neuron is as follow (etch point <x,y>, blue =1, red = -1):

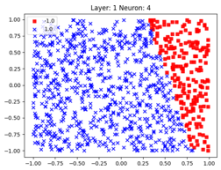
MLP layer 1 **MLP layer 2** **ADALINE output layer**

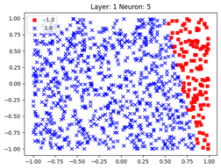


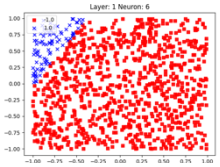


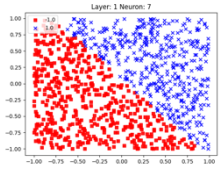


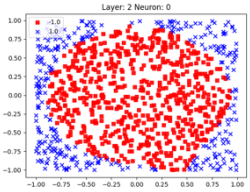


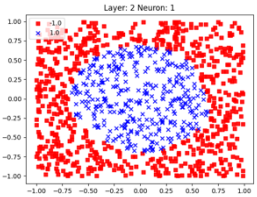


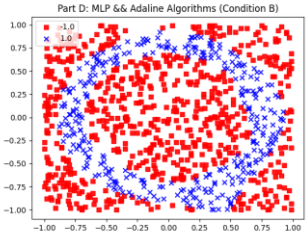




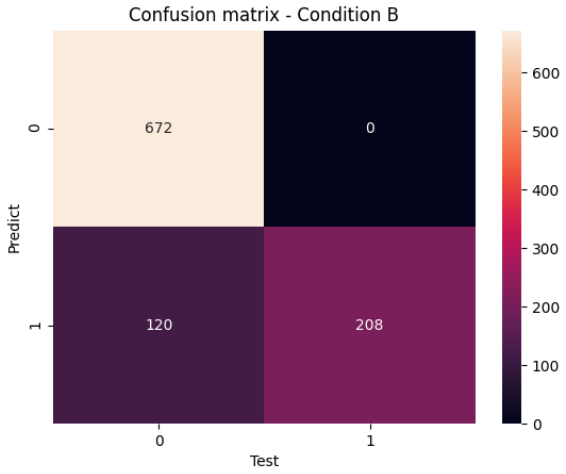








Lat's look at the final result of the combination between MLP and Adaline Algorithms (train and test data):



It can be seen that Adaline with MLP input predict a better result, and instead of correction of **79.2%** in part C, now we get to correction of **88%.**

Moreover, we can tell that we got a better prediction while using only MLP Algorithm.