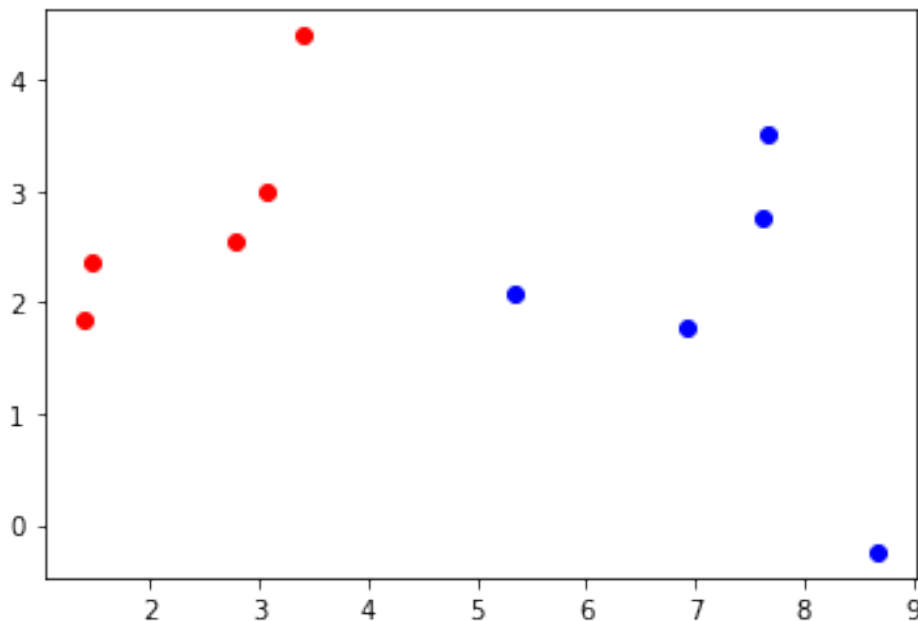


Notebook Chap8 N1_Perceptron

1. Description of the toy data set



1. Is this data set linearly separable?

YES; $X[1]=4.5$ ($X[1]$ is the horizontal axis) for instance is the equation of a straight line separating the two clusters.

2. Define a single neuron, with a threshold step activation function

1. Show that the predict function defines an separation hyperplan in $\mathbb{R}^{\dim(X)}$

$w[0]+w[1]X[1]+w[2]X[2]. \dots x[\dim(X)]X[\dim(X)] = 0$ defines an hyperplane of dimension $\dim(X)-1$. (Note that we set $X[0]=1$ by default).

2. Express the equation of the separation line as a function of the $\{w[k]\}$ in the case $\dim(X)=2$

$w[0]+w[1]X[1]+w[2]X[2] = 0$ is the equation of a line.

3. Show in that latter case that setting $w[2]=0$ amounts to define a threshold on the first coordinate

Setting activation=0 gives $X[1] = -w[0]/w[1] = cste$: this is the equation of a vertical line. For any new observation, the classification rule is based on activation= $w[0]+w[1]X[1]=w[1](X[1]-cste)$; comparing activation to zero amounts to compare $X[1]$ to a threshold= $cste$).

4. Propose a set of values for $\{w[0], w[1], w[2]\}$ which defines a good classifier for the data above.

Set $w[2]=0$; $w[1]=1$ and $w[0]=-4.5$ for instance.

3. Application of « predict » function

1. What is the equation of the boundary?

see question 2.4 above. Boundary equation is $X[1]=4.5$

2. What is the value of the bias (or intercept)?

The intercept is the value of $w[0]$, i.e. -4.5

3. Is this solution unique?

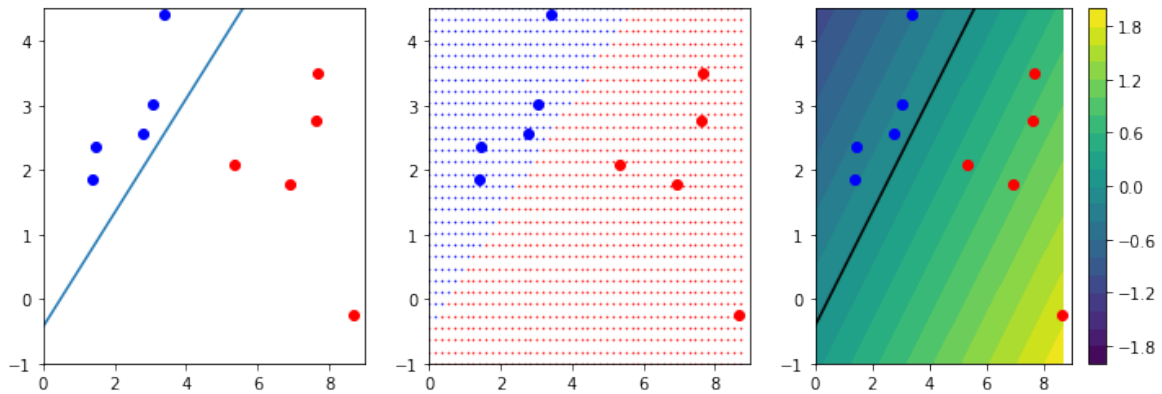
NO, any value of $w[0]$ between 3 and 5 would lead to a acceptable solution separating the two clusters on this example.

4. Learning weights using SGD

1. How many operation (take only multiplication into account) are necessary to complete an epoch in the function train_weights defined above?

*3 coefficient must be updated ($w[0], w[1]$ and $w[2]$) for each data. An epoch is defined by a « cycle » during which the coefficient are updated once for each new data. 3 multiplications are required for the update phase, for each data. Furthermore, for each new value of vector W , the classification results must be evaluated (to compute the error), requiring 2 multiplications for each sample, that is 20 multiplications. Finally, a total of 23 multiplications are required for each sample, leading to a total of $23*10=230$ multiplications for an epoch. Note that we assume that computing the gradient does not require any multiplication.*

5. Apply the weights estimated by SGD and visualize classification results



1. Considering the given code lines in the notebook, which is the most informative representation?

The left side plot represent the data and the decision border. The central plot adds the information about the decision rule (red or blue decision associated to the partition of the plane). The right side plot adds the information about the value taken by the activation function.

6.Using Scikit-learn perceptron function

- 1.Using the sklearn reference documentation, identify the role of the parameters "Shuffle" and "Validation_fraction"

« Shuffle » option allows to consider the sample in a different order, set at random, for each epoch. Validation fraction allows to evaluate the error on a subset that does not match with the training set (the sample used for the update operation are not those used for evaluating the error)

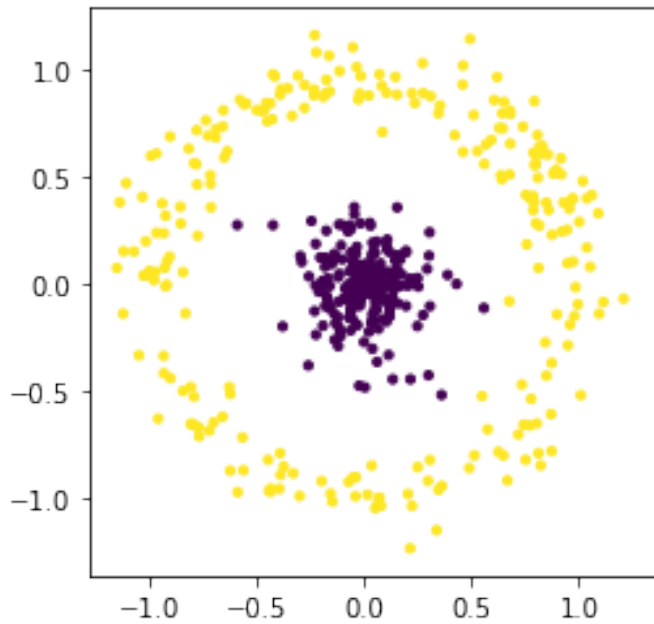
2. Discuss the interest of introducing such parameters

This intends to avoid over fitting.

3. What was the value of parameter "eta0" in our "train_weights" code?

eta_0 in scikit-learn code matches the parameter l_rate in our « train_weights » code.

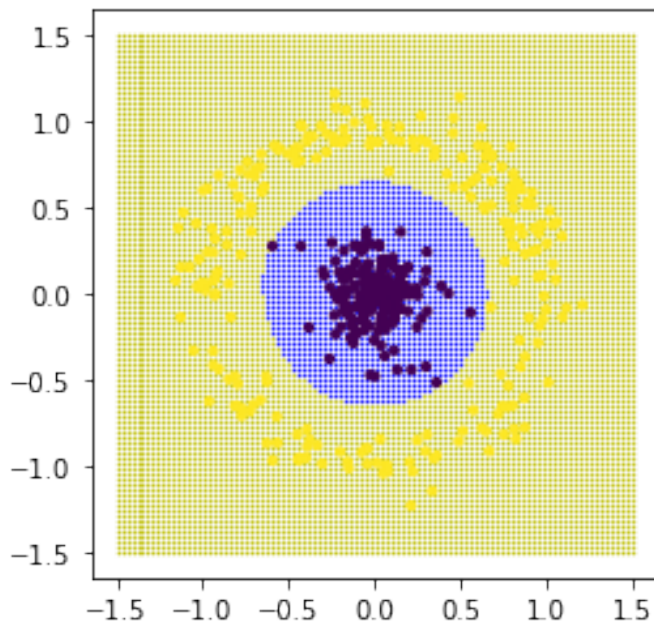
7. Problem



1. Do you think that a perceptron (also called single layer perceptron) is a performant classifier for this problem?

If one considers the cartesian representation of these two cluster data, it is clear that no straight line can separate the clusters. As a perceptron for 2-dim data defines a linear separator, it is expected that no solution will give satisfactory results.

Considering the given code, the obtained result is



1. Run the code above a few times : describe the shape of the decision regions that you obtain

Obtained regions exhibit cylindrical symmetry.

2. Is it in contradiction with the property of (single layer) perceptron stating that their are linear separators?

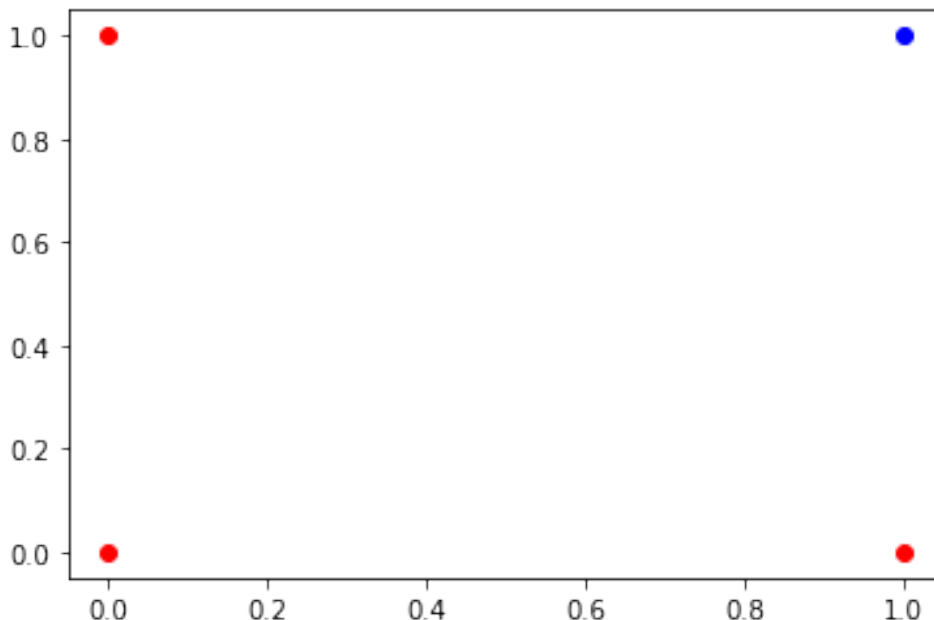
No : analysing the code shows the representation space is not cartesian but polar. The transform relating these two representations is NON LINEAR. Transforming the cartesian plane into the polar representation makes it possible -in this case- to separate the cluster with a linear (in the polar representation) classifier.

3. What do you conclude about the choice of the representation space of the analyzed data?

The representation space of the data plays a key role in these approaches. applying non linear transforms to sample sets may allow linear approaches to perform well. This is at the core of non linear embedding methods.

Notebook Chap8 N2_MultiLayer Perceptron Classifier

1.Example of boolean functions



1. Is this 2 clusters problem linearly separable?

YES

2. how many layers are necessary to separate the two clusters?

A single layer perceptron is enough

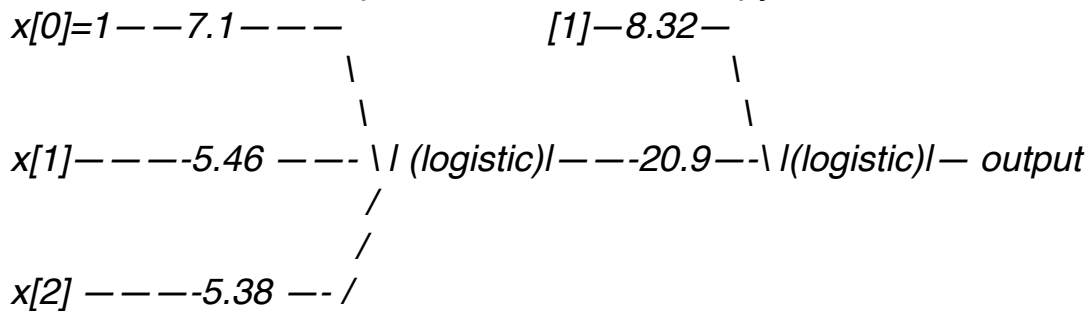
3. Would a Perceptron (similar to the the one studied in N1_Perceptron.ipynb) be a acceptable solution?

YES

4. Scikit MLPClassifier : Draw the learner structure and its edges with corresponding weights.

MLPC specifies a single hidden layer: the MLPC includes an input layer, an hidden layer and an output layer.

Note that the MLPC optimizes the cross entropy loss function.

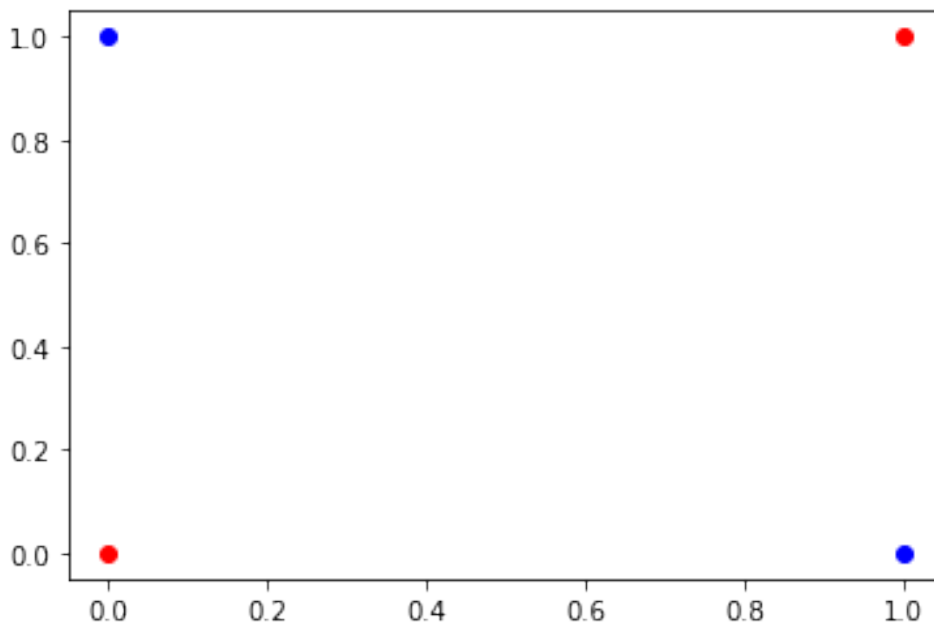


5. What is the activation function used in this MLPC?

Logistic function on this example

5. In the MLPC computation above, 'lbfgs' is used as a solver, not 'sgd'. Explain why

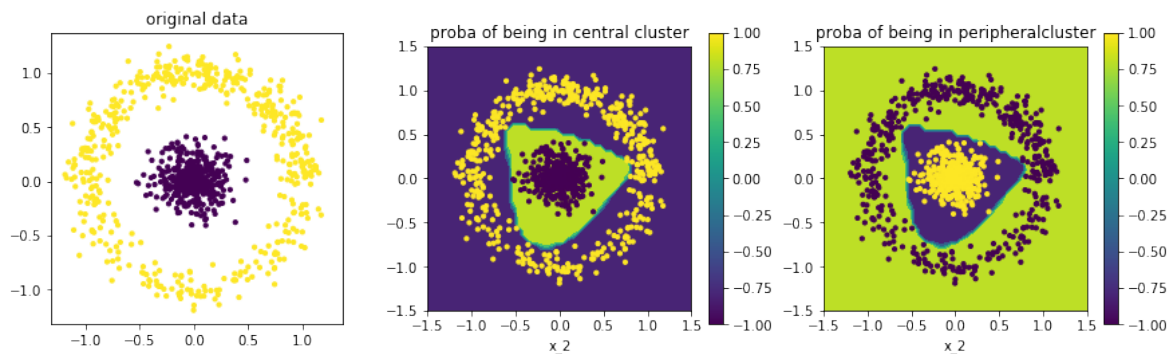
Using SGD on such a small data sets does not make sense.



6. Would any solution involving a single layer lead to a correct result?

No : no single line allows to seprate the two clusters.

2. Concentric clusters



1. Run the learning process many times with input parameters `hidden_layer_sizes=(3,2)`. Comment your findings. Propose an interpretation.
Some results appear to be of bad quality. This is due to the fact that calling the optimization algorithm with random sequences and random initial conditions lead sometimes to solutions that correspond to local minima of the cost function : the algo remains stuck on this local minimum although the solution is not acceptable.

2. Increase the number of neurons in the first and second layer and comment your observation. Can you explain your findings?
By increasing the complexity of the networks, many more local minima are also created, that correspond to correct classifiers. An acceptable solution is most of the time obtained.

3. What is different between this method and the method used in `N1_Perceptron.ipynb` notebook?
In the present case, the data are expressed in the cartesian representation space, where no linear separation is possible... on the contrary to the case in the previous notebook where data were expressed in polar coordinates

4. How would you evaluate the performances of the obtained MLPC?
Cross validation... as usual.