Short report on lab assignment 1 - Classification with a single-layer perceptron

Group 15 : Côme Lassarat, Victor Sanchez and Marie-Ange Stefanos

31st of January 2022

1 Main objectives and scope of the assignment

List here a concise list of your major intended goals, what you planned to do and what you wanted to learn/what problems you were set to address or investigate, e.g.

Our major goals in the assignment were:

- to design single-layer perceptrons so as to solve simple classification problems
- to implement and compare the resulting algorithms: perceptron learning and delta rule, online and batch mode
- to identify the characteristics of those algorithms implemented: what are their advantages and inconvenients?
- to monitor their learning process and behaviour when modifying the learning rate or the separability of the dataset

This study will only focus on simple binary classification problems randomly generated. In the following figures, "epoch: 0" actually corresponds to the first epoch of the given algorithm.

2 Methods

In order to reach the intended goals, the different methods and algorithms were coded in Python, a wide-spread programming language. Several libraries were also imported and used:

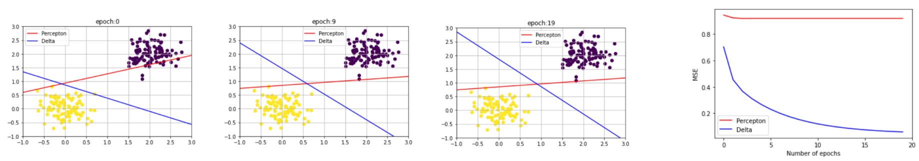
- **Numpy** to handle multi-dimensional arrays and to generate linearly-separable and not linearly-separable data
- **Matplotlib** to construct graphs (of errors for example)
- **copy** to deep-copy the arrays

3 Results and discussion

3.1 Classification with a single-layer perceptron

Before implementing the different methods, it is necessary to generate data that can be used for binary classification: two classes of different colours were created (see Figure 1a).

3.1.1 Online: Perceptron learning VS Delta learning

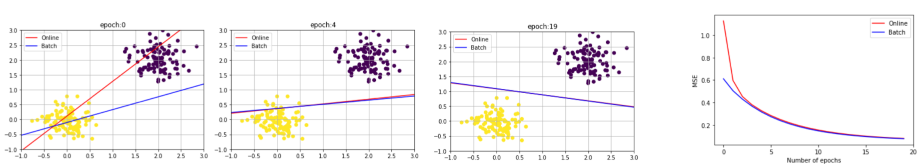


(a) (b)

Figure 1: Boundaries evolution through the epochs using percepton learning and delta learning for a learning rate $\eta = 10^{-3}$ (Online) (Figure 1a) and MSE error evolution (Figure 1b)

- Both algorithms converge since their MSE seem to converge (Figure 1b)
- Binary classification by the delta learning is better than the one with percepton learning: delta rule's MSE is lower (Figure 1b) and delta rule's boundaries seems more convenient (one can observe on Figure 1a that the percepton learning ended prematurely, which is a frequently encountered problem which depends on the initialization of the its weights)

3.1.2 Delta rule: Online vs Batch



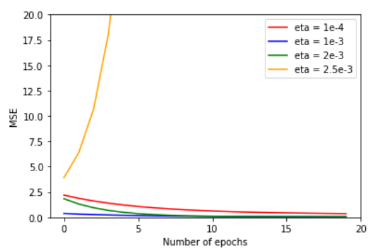
(a) (b)

Figure 2: Boundaries evolution through the epochs using delta rule for a learning rate $\eta = 10^{-3}$ (Online VS Batch)

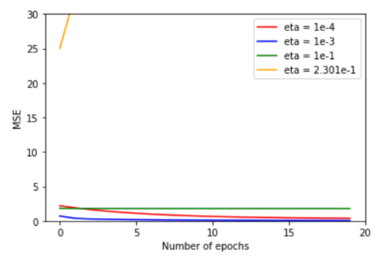
- When looking at Figure 2, the delta algorithm converge to the same boundary in batch and online mode, after around 10 epochs (depending on the convergence criteria used).

- Batch delta algorithm have a greater precision than the batch delta algorithm, which explained why the batch mode in mostly used for this method. Nevertheless, the online mode rapidly converges to the precision of the batch mode.

Now let's compare the influence of the learning rate η on the batch delta learning (Figure 3a) and on the online delta learning (Figure 3b)



(a) Batch delta learning



(b) Online delta learning

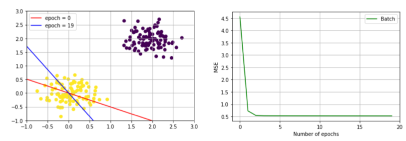
Figure 3: Learning curves for the Batch delta learning (a) and for the Online delta learning (b)

- One can notice on Figure 3 that the batch delta rule is more sensitive to the learning rate than the online delta rule: the algorithm diverges for a lower learning rate
- The Figure 3 highlights the following:
  - a low learning rate implies a slow convergence (red curves)
  - when the learning rate increases, the convergence get faster but doesn't change the precision (blue curves)
  - when the learning rate is too high, the convergence speed starts to decrease and the precision can also get smaller (green curves). Passed a certain threshold, the algorithm oscillates or diverges (yellow curves)
- the learning is very sensitive to initialization: the highest learning rate on Figure 3b and 3a sometimes lead to the convergence of the algorithm.

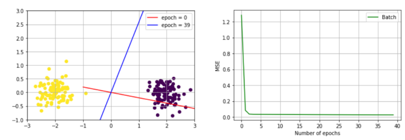
3.1.3 Delta Rule in batch mode: removing the bias

Finally, the delta rule is implemented with its bias removed to study its behaviour (Figure 4)

- On Figure 4a, the delta rule can't classify with sufficient robustness the dataset: it can only rotate along the origin of the graph.



(a) Classes not symmetrical along the Y-axis



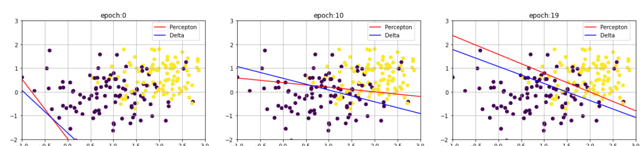
(b) Classes symmetrical along the Y-axis

Figure 4: Delta rule in batch mode implemented for a learning rate $\eta = 10^{-3}$

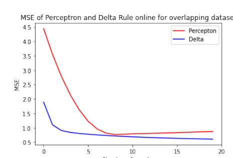
- Thus, the only case where the delta rule without bias can classify the data with the best robustness is when the clouds of points are symmetrical along a straight line passing through the origin (if it is admitted that the classes doesn't overlap with the each others and with the origin) (Figure 4b): it doesn't need bias.

3.2 Classification of data that are not linearly separable

3.2.1 Perceptron learning VS Delta learning with overlapping data



(a)



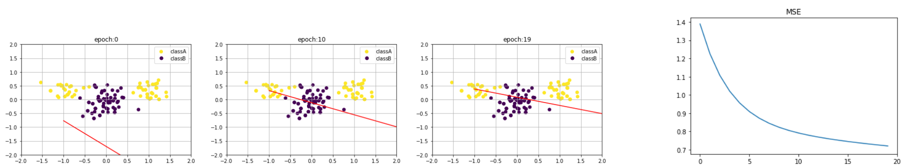
(b)

Figure 5: Boundaries evolution through the epochs using percepton learning and delta learning for a learning rate $\eta = 10^{-3}$ (Online) (Figure 5a) and MSE error evolution (Figure 5b)

- When two data clouds overlap, both perceptron learning and delta rule algorithms classify the data with error. It nevertheless seem that the delta rule converges towards a fixed boundary while the perceptron learning doesn't after 20 epochs (Figure 5)

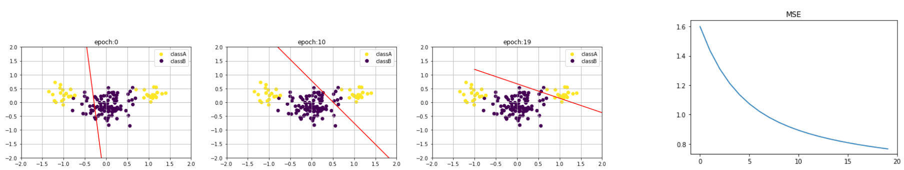
3.2.2 Delta rule in Batch mode with non linearly separable data

The following scenarios are analyzed:
- Scenario 1: removing random 25% from each class
- Scenario 2: removing random 50% from classA
- Scenario 3: removing random 50% from classB
- Scenario 4: removing 20% from a subset of classA for which classA(1,:)<0 and 80% from a subset of classA for which classA(1,:)>0



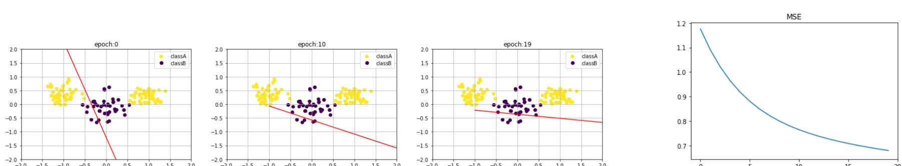
(a) (b)

Figure 6: Scenario 1 : Boundaries evolution through the epochs using percepton learning and delta learning for a learning rate $\eta = 10^{-3}$ (Online) (Figure 6a) and MSE error evolution (Figure 6b)



(a) (b)

Figure 7: Scenario 2 : Boundaries evolution through the epochs using percepton learning and delta learning for a learning rate $\eta = 10^{-3}$ (Online) (Figure 7a) and MSE error evolution (Figure 7b)



(a) (b)

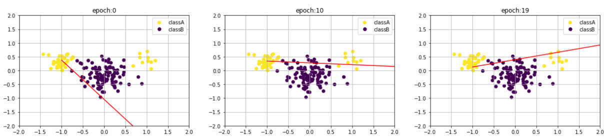
Figure 8: Scenario 3 : Boundaries evolution through the epochs using percepton learning and delta learning for a learning rate $\eta = 10^{-3}$ (Online) (Figure 8a) and MSE error evolution (Figure 8b)

For each scenario

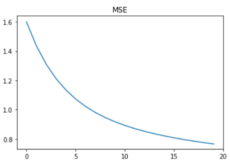
Accuracy (percentage of well-classified elements for each class comparison

- As the data is not linearly-separable, one can see on Figures 6b, 7b, 8b, 9b that the delta rule keeps trying to minimize the mean-square error, which corresponds to a classification that promotes the class which contains the most points (see Figures 6a, 7a, 8a, 9a).

- The fact that the delta rule tries to properly classify the data cloud which has the most element can also be seen on Figure 11: the majority cloud has an accuracy (percentage of well-classified elements) close to 100%. When



(a)



(b)

Figure 9: Scenario 4 : Boundaries evolution through the epochs using perceptron learning and delta learning for a learning rate $\eta = 10^{-3}$ (Online) (Figure 9a) and MSE error evolution (Figure 9b)

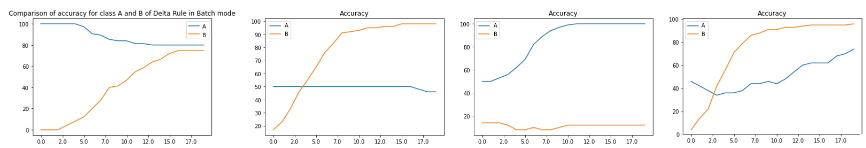


Figure 10

Figure 11: Comparison of Accuracy for each scenario

the data clouds have the same size, both of them have approximately the same accuracy (Scenario 1).

- All of this highlights that one must be careful that the data is well balanced if different classes don't overlap.

4 Final remarks

To conclude, this study first underlined the differences between the perceptron learning algorithm and the delta rule method when doing binary classification on linearly-separable dataset. It was shown that perceptron learning lacks of robustness for generalization whereas the delta rule (online and batch) algorithm is robust. Nevertheless, both perceptron learning and delta rule can be stuck in a local minima. Furthermore it was highlighted that the online method is faster to converge for larger data set than the batch mode. Then, the influence of the learning were tested to demonstrate the need to adapt it for every problems. In a second place, performance of the previous algorithms were studied on overlapping dataset (thus not linearly-separable) which resulted of a high bias and many errors. Finally, the batch delta rule algorithm were implemented on several pairs of unbalanced non-linearly separable data clouds to underline that the latter method tend to maximize the accuracy of the class composed of the most elements.