Homework Assignment N°2

BML36
Thibault Douzon
Rajavarman Mathivanan

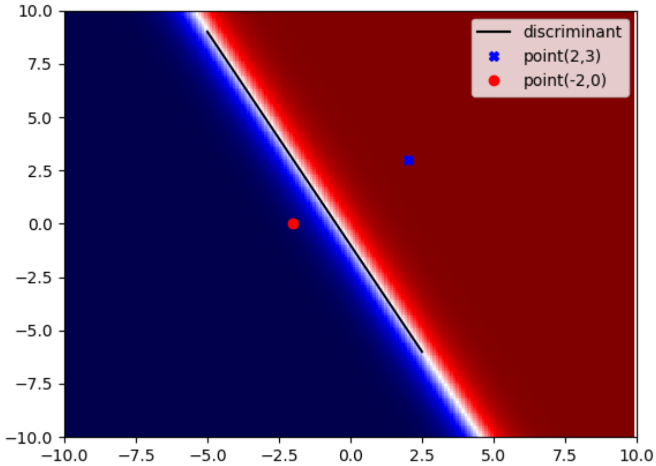
September 12th, 2018

Contents

1 Exercise 1: Perceptron Learning Rule

1.1 Part a

To have an idea of what our perceptron looks like, let's plot its discriminant line and the two data points.



We can also compute the prediction of the model for the two datapoints:

$$\text{prediction}_1 = sign(w^\top x_1) = sign(8) = 1 \neq -1$$

$$\text{prediction}_2 = sign(w^\top x_2) = sign(-3) = -1 \neq 1$$

So our two datapoints are missclassified, now we can say that on the plot below area in red (upper right of the plot) represents the area classified as class 1 by our model and blue (lower left part of the plot) is for class -1

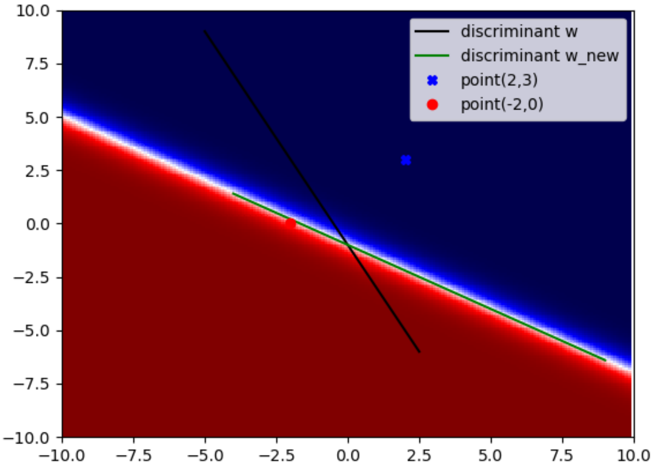
1.2 Part b

After one iteration of a batch learning algorithm we get the following new discriminant:

$$w_{new} = w_{old} + \eta \sum_{i=1}^{N} x_i c_i$$

$$w_{new} = \begin{bmatrix} 1 & 0 & -0.5 \end{bmatrix}$$

And now, both data points are acorrectly classified by the new model:



1.3 Part c

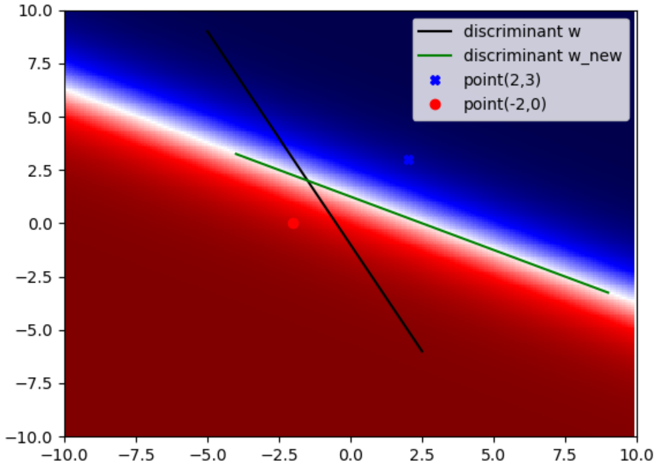
Learning is an iterative process. In a batch version learning, at each iteration we compute a new model based on the whole dataset whereas in a stochastic version, at each iteration we pick up a single data in the dataset and we base our computation on this data point only.
When the dataset becomes too big, the computation time of a batch iteration is going to need ressources proportional to the size of the dataset and a stochastic iteration will always use the same amount of ressources independantly of the size of the dataset.

1.4 Part d

When applying the stochastic algorithm, we get the following (different) result:

$$w_{new} = \begin{bmatrix} 1 & -0.4 & -0.8 \end{bmatrix}$$

And again, both points are correctly classified after the complete sweep:



2 Exercise 2: Logistic classification & discrimination

2.1 Part a

Initialize w_0 ?

1. Some fixed w_0 like $\begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}$
2. Some computation around the dataset like the mean: $w_0 = \frac{1}{N}\sum_{i=1}^{N} x_i$, concatenated with a constant.
3. Some random vector

How to learn: for batch learning use this equation at each step

$$w_{n+1} = w_n - \eta \nabla E(w_n) = w_n - \eta \sum_{n=1}^{N} (y(n) - t_n)\, x_n$$

How to stop the iterative process ?

1. Stop when the norm of the difference vector is low: $\Delta_n = \frac{\|w_{n+1} - w_n\|}{\|w_n\|} < \epsilon$
   This is a commonly used criterion that stops the process when the steps we take are getting small compared to our current result.
2. Stop after fixed number of iteration
   This ensures we won't enter in a infinite non-convergent process.
3. Stop when a threshold error is reached: $E(w_n) < \epsilon$
   This is actually a bad idea because most of the time we can't be certain it is possible to reach such threshold on the error. It would result in an infinite process.

Our algorithm goes as follows:

1. Chose ϵ, N and η respectivelly for precision, maximum number of iterations and speed convergency.

2. Set current error Δ to $+\infty$ and n to 0

3. Chose the initial discriminant: $w_{current}$. WE NEED TO CHOSE THE METHOD !

4. While $\Delta > \epsilon \wedge n < N$ do

   (a) Compute and store next discriminant w_{next}:

   $$w_{next} = w_{current} - \eta \sum_{n=1}^{N} \left(\sigma({w_{current}}^\top x_n) - t_n \right) x_n$$

   (b) Compute and store the new error Δ:

   $$\Delta = \frac{\|w_{next} - w_{current}\|}{\|w_{current}\|}$$

   (c) Prepare for next iteration: store w_{next} in place of $w_{current}$ and increment n

5. If $\Delta > \epsilon$, it means we have not converged enough towards the limit. We should consider increasing N OR using another algorithm for convergence (eg. Newton-Raphson)

6. Result is stored in $w_{current}$, number of steps in n.