

Homework 1

Generate a dataset of two-dimensional points, and choose a random line in the plane as your target function f , where one side of the line maps to $+1$ and the other side to -1 . Let the inputs $x_n \in \mathbb{R}^2$ be random points in the plane, and evaluate the target function f on each x_n to get the corresponding output $y_n = f(x_n)$. Experiment with the perceptron algorithm in the following settings:

1. Generate a dataset of size 20. Plot the examples (x_n, y_n) as well as the target function f on a plane.

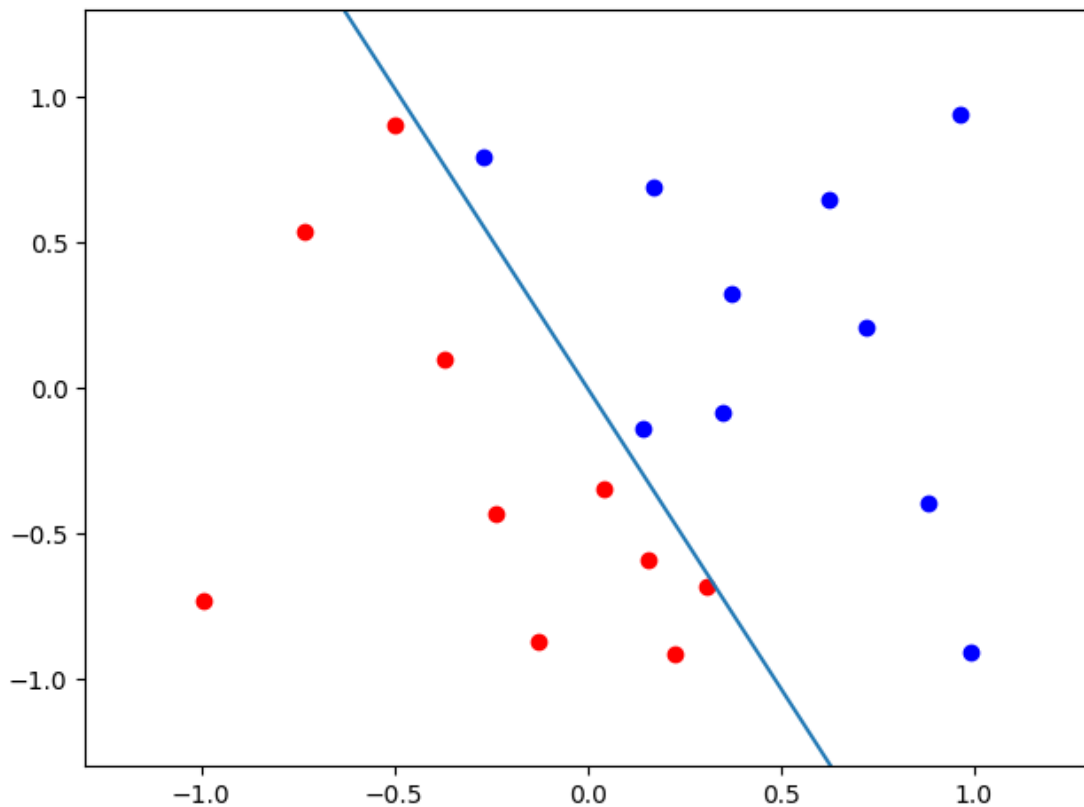


Figure 1

2. Run the perceptron algorithm on the dataset. Report the number of updates that the algorithm takes before converging. Plot the examples (x_n, y_n) , the target function f , and the final hypothesis g in the same figure.

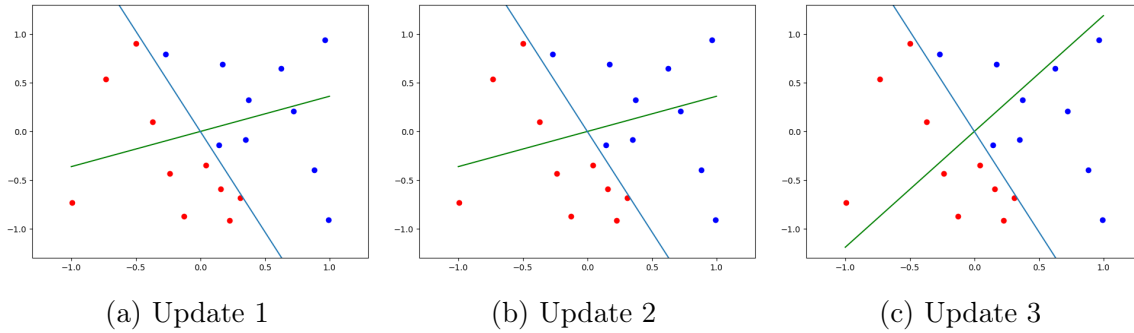


Figure 2: Updates

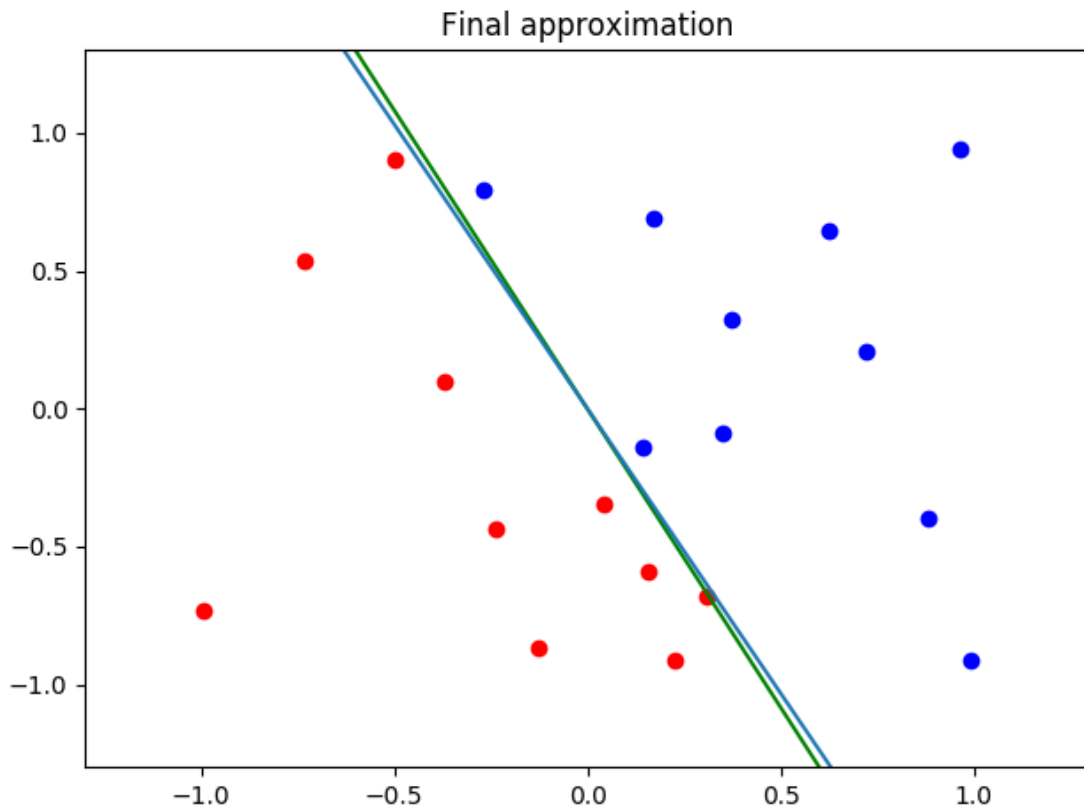


Figure 3

3. Repeat everything in 2. with another randomly generated dataset of size 20, and compare the result to 2.

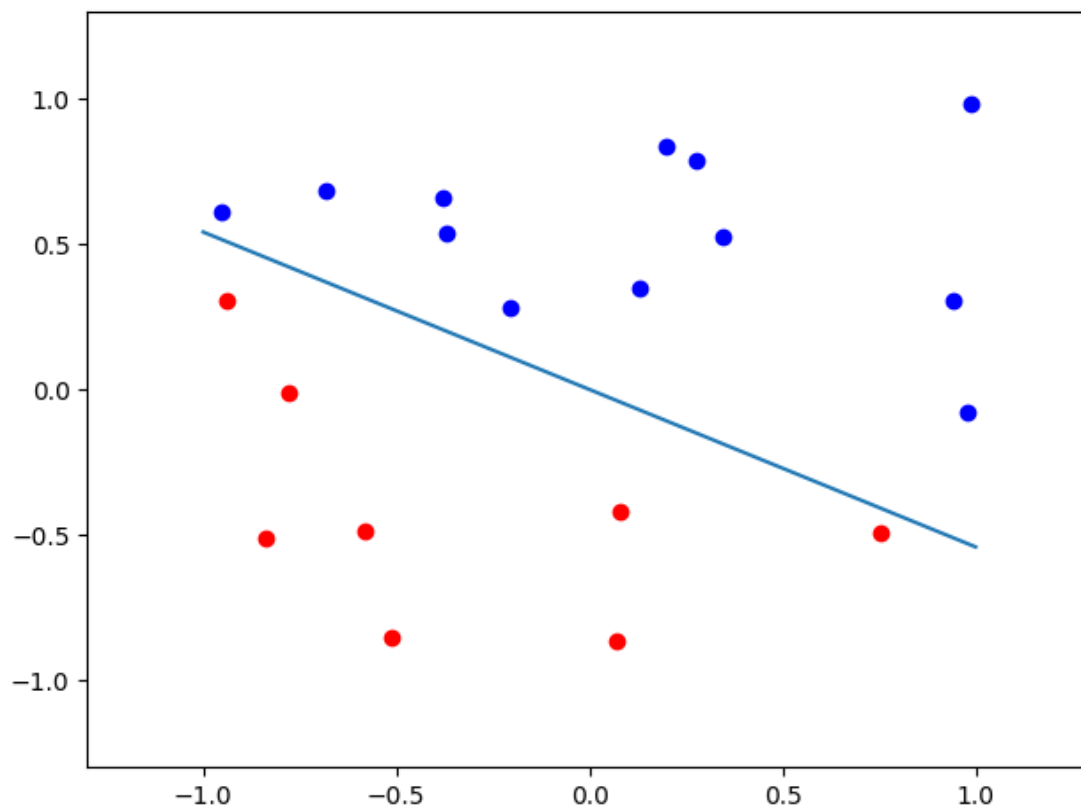
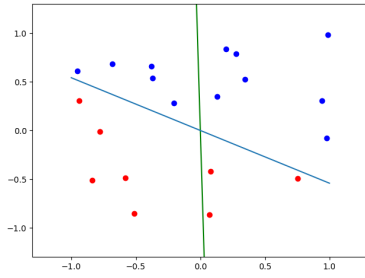
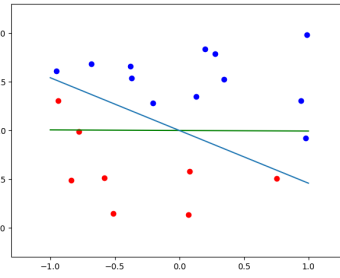


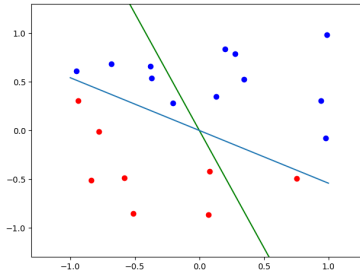
Figure 4: Another randomly generated dataset and the target function



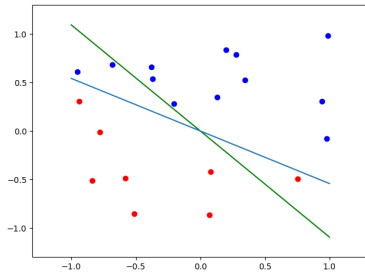
(a) Update 1



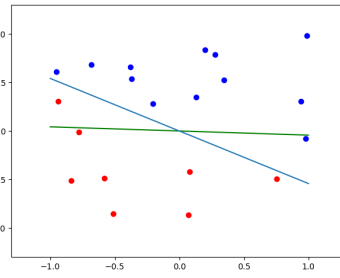
(b) Update 2



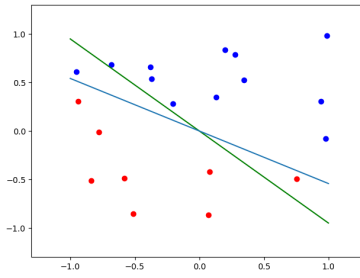
(c) Update 3



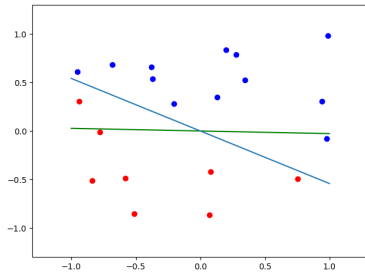
(d) Update 4



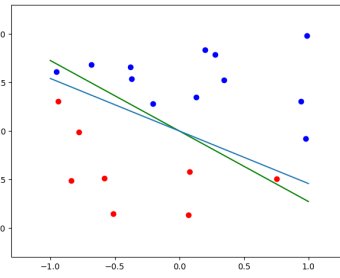
(e) Update 5



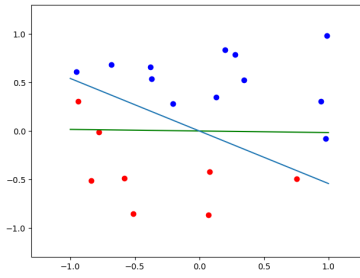
(f) Update 6



(g) Update 7



(h) Update 8



(i) Update 9

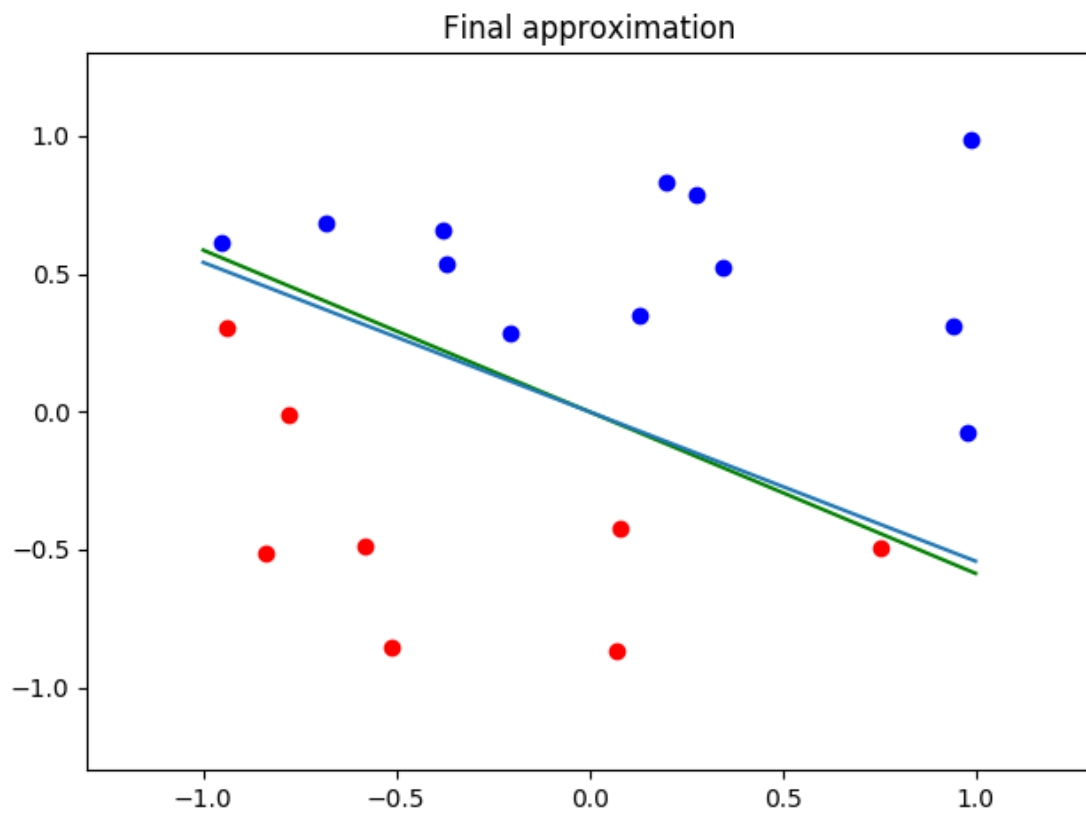


Figure 6

4. Repeat everything in 2. with another randomly generated dataset of size 100, and compare the result to 2.

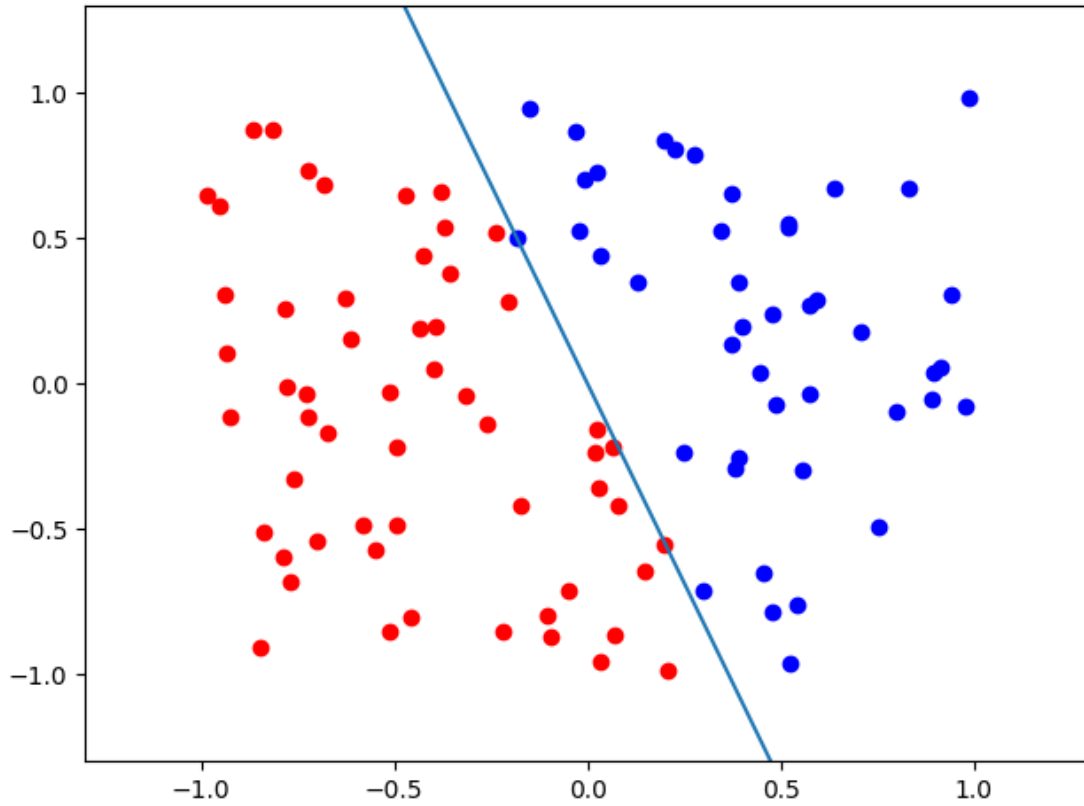
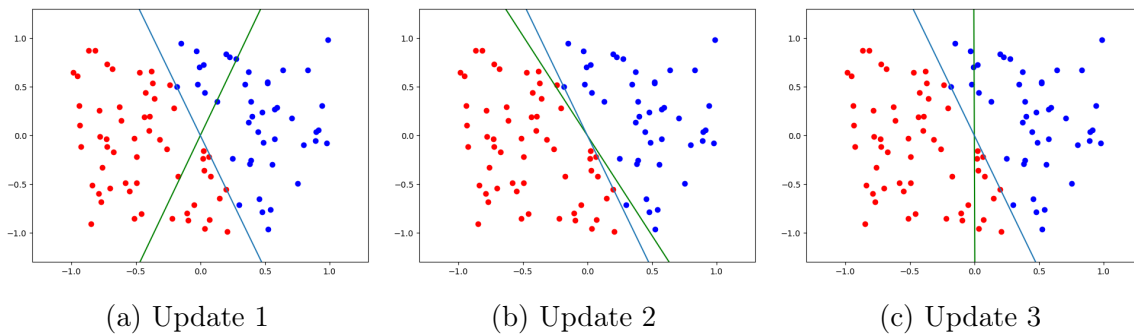


Figure 7: Randomly generated dataset of size 100 and the target function



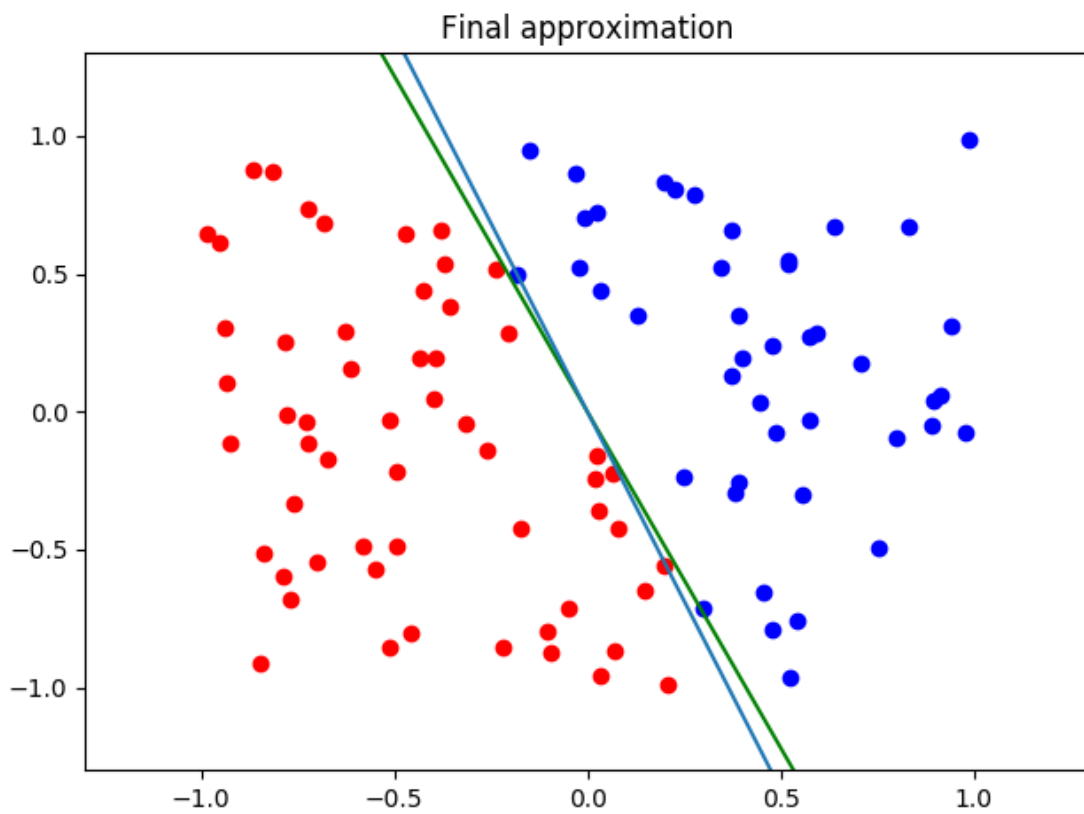


Figure 9

5. Repeat everything in 2. with another randomly generated dataset of size 1000, and compare the result to 2.

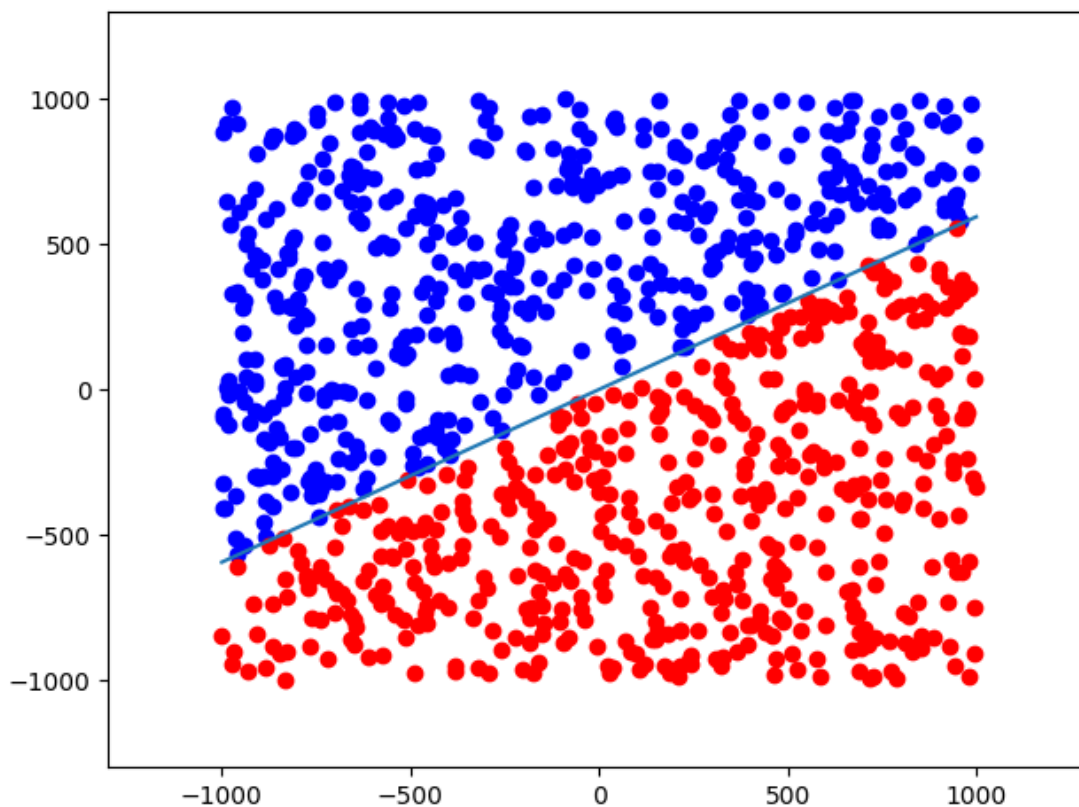


Figure 10: Randomly generated dataset of size 1000 and the target function

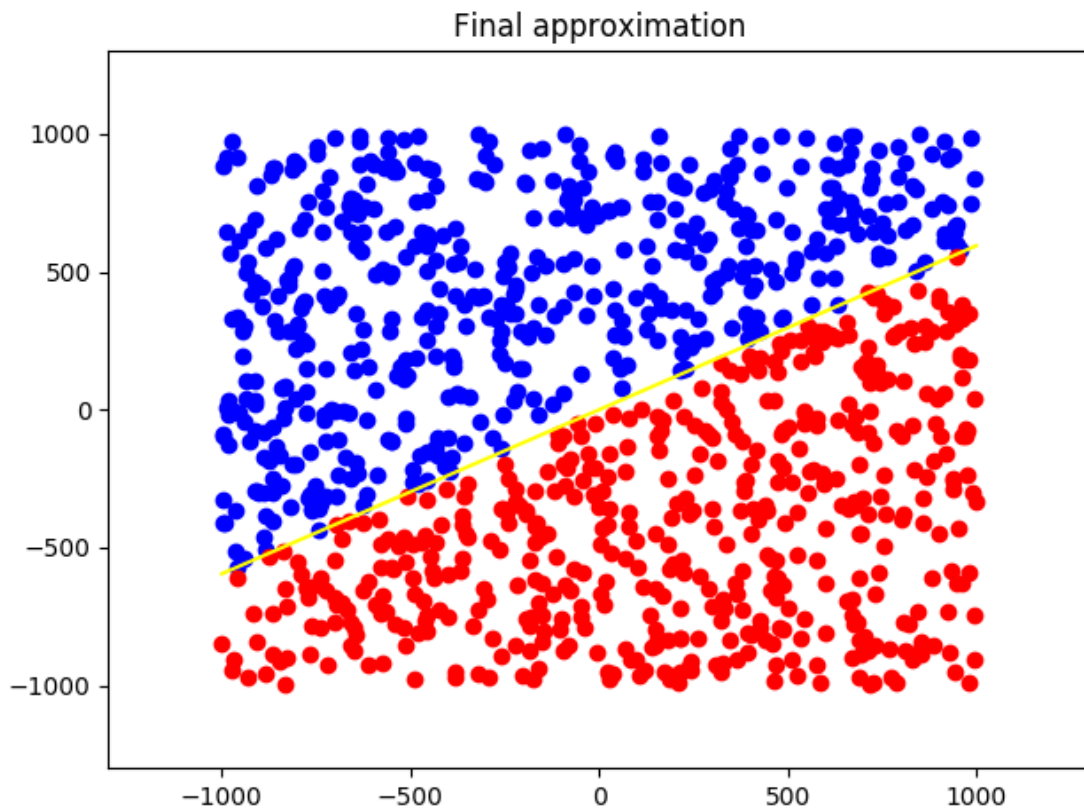


Figure 11: Final hypothesis (in yellow) g after **97 iterations**

6. Modify the experiment such that $x_n \in \mathbb{R}^{10}$ instead of \mathbb{R}^2 . Run the algorithm on a randomly generated dataset of size 1000. How many updates does the algorithm take to converge?

The algorithm takes 3512 updates to converge.

7. Summarize your conclusions regarding the accuracy and running time of the algorithm as a function of N (the number of data points) and d (the number of dimensions).

\mathbb{R}^2 , dataset of size 20 (10 updates): **0.0005986690521240234 seconds**

\mathbb{R}^2 , dataset of size 100 (4 updates): **0.0011091232299804688 seconds**

\mathbb{R}^2 , dataset of size 1000 (97 updates): **0.03201484680175781 seconds**

\mathbb{R}^{10} , dataset of size 1000 (3512 updates): **1.349951982498169 seconds**

Conclusions

A larger quantity of samples does not imply a greater number of updates (see cases first and two) but (obviously) it does affect the execution time.

This measures depend on the dispersion of the samples, which is not proportional to the number of samples in each example. This disadvantages the cases with a high number of samples.