

Homework 1

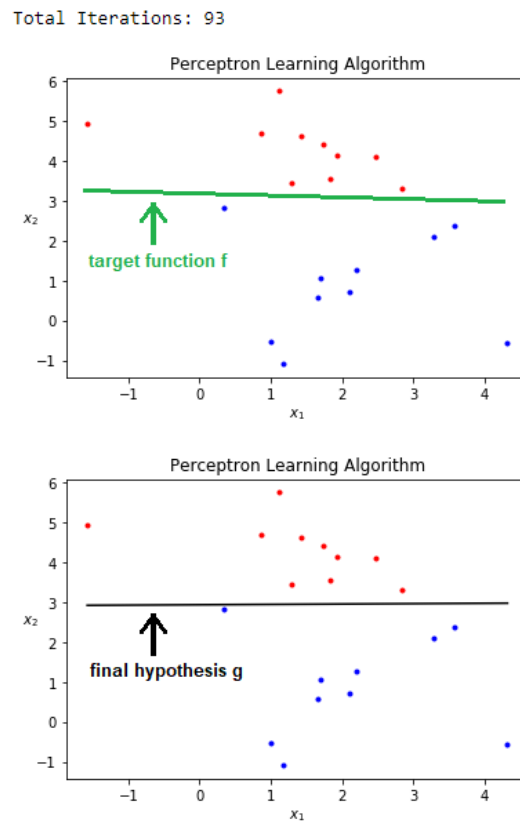
Max Schemitsch

2/13/2019

1 Problem 1.4

1.1 A) and B)

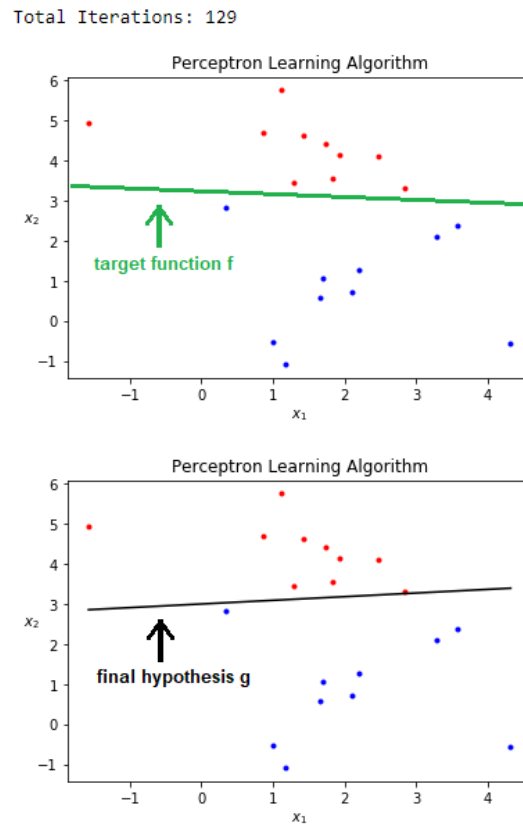
Using the code found in 'Master PLA.py', we can generate a linearly separable data set of size 20 with target function f and final hypothesis g .



After 93 iterations, we can see that f is very close to g . They both satisfy the division of classes.

1.2 C)

We can do the same thing as before with another random data set.

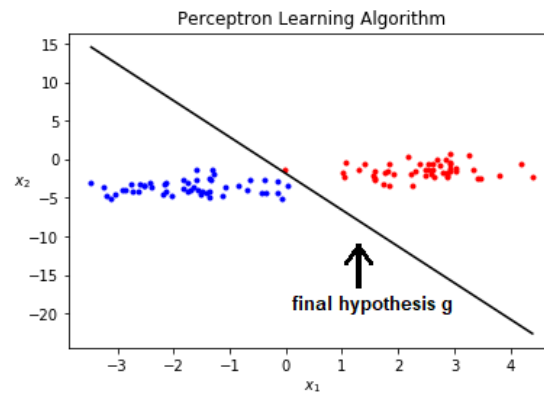
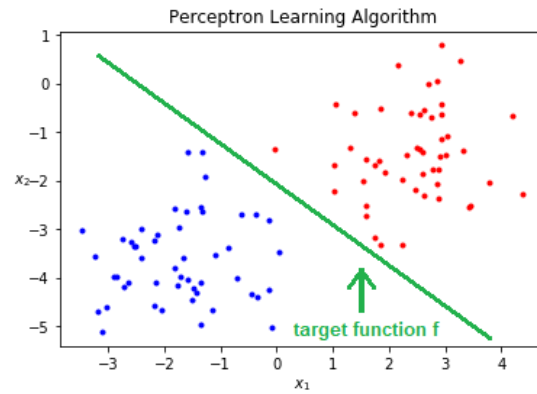


After 129 iterations, this particular f function is close to this particular g function. My initial target function, however, had a negative slope as opposed to the positive slope of g .

1.3 D)

We obtain similar results with a size of 100. We have 8 iterations and a similar f and g .

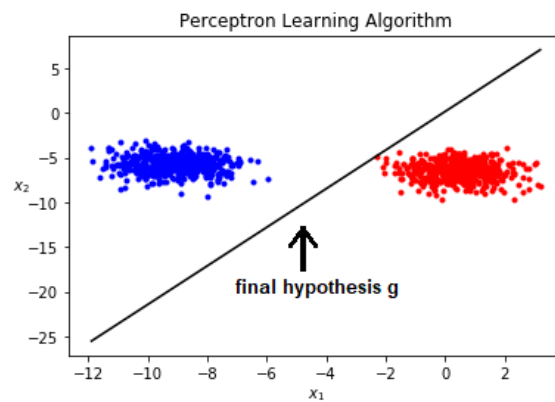
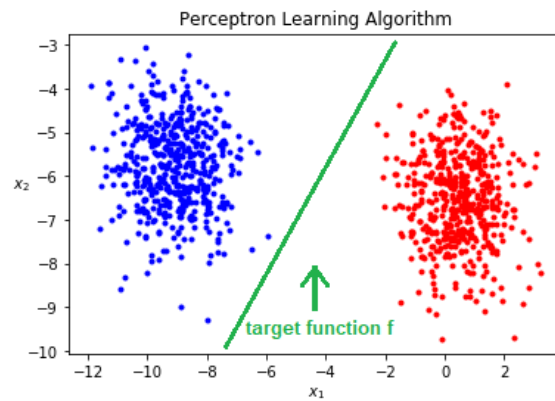
Total Iterations: 8



1.4 E)

We obtain another set of similar results with a size of 1000.

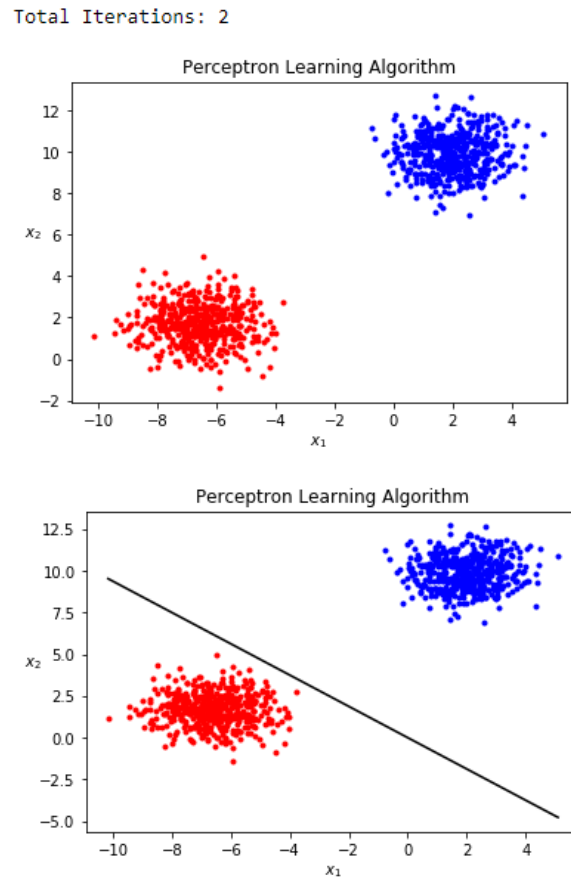
Total Iterations: 5



We can see that while my f function is roughly in the middle of the data sets, g sits right on the border of the red classified data.

1.5 F)

In order to modify the number of dimension from 2 to 10, we must alter part of the code found in 'Master PLA.py'. We must set $nfeatures = 10$ and make sure we have $np.zeros(11)$. We can then run the code to obtain this:



This particular random data set only took two iterations to converge.

1.6 G)

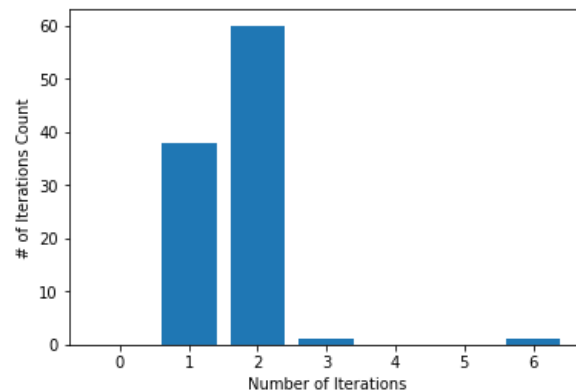
In order to run the experiment 100 times and collect 100 different iteration counts, we apply this bit of code before and after our *main()*:

```
itlst = []
    for x in range(100):
        .....
itlst.append(it)
    print(itlst)
```

With our list of iteration counts, we can then plot a histogram using this bit of code:

```
his = plt.hist(itlst , bins = [0, 1, 2, 3, 4, 5, 6, 7]);
fig, ax = plt.subplots()
offset = 0
plt.bar(his[1][1:], his[0])
ax.set_xticks(his[1][1:] + offset)
ax.set_xticklabels( ('0', '1', '2', '3', '4', '5', '6', '7') )
plt.xlabel("Number_of_Iterations")
plt.ylabel("#_of_Iterations_Count")
```

This code produces our histogram:



We can see that the majority (around 60) of experiments took 2 iterations and roughly 38 of them took only 1 iteration. Only one experiment needed 3 iterations and another needed 6.

1.7 H)

When working through this homework, it seemed that the most important aspect of this PLA was the data set size. I could not get a 1000 size data set to run the algorithm when I had the random state set equal to a value. It would never get to the point of converging, and run forever. When I removed the random state variable, the algorithm worked almost every time. It also seems that the running time for this PLA was linked directly to the data set size (it felt exponentially longer from 20 then to 100 and finally to 1000). Dimension size also directly increased the running time as well.