Homework 2

Max Schemitsch

3/6/2019

1 Problem 2.1

First we can solve the equation for N (and also sub in 0.03 for δ and 0.05 for σ.)

$$\begin{aligned}
\sqrt{\frac{1}{2N}\ln\frac{2M}{0.03}} &\leq 0.05 \\
\frac{1}{2N}\ln\frac{2M}{0.03} &\leq 0.0025 \\
\frac{1}{2}\ln\frac{2M}{0.03} &\leq 0.0025N \\
\frac{\frac{1}{2}\ln\frac{2M}{0.03}}{.0025} &\leq N \\
200\ln\frac{2M}{0.03} &\leq N
\end{aligned} \tag{1}$$

Plugging in $M = 1$ gives us $N = 840$ samples.
Plugging in $M = 100$ gives us $N = 1761$ samples.
Plugging in $M = 10000$ gives us $N = 2682$ samples.

2 Problem 2.11

Using this Python code, we can calculate the E_{out} bound with $\delta = 0.1$ and for $N = 100$ and $N = 10000$.

```
import numpy as np
import math
dvc=1
N=100
M=10000
d=0.1

e = math.sqrt(8 / N * np.log((4 * ((2 * N) ** dvc + 1))) / d)
f = math.sqrt(8 / M * np.log((4 * ((2 * M) ** dvc + 1))) / d)
print(e)
print(f)
```

```
2.3133697100427275
0.3005306228976483
```

Thus our E_{out} bounds for $N = 100$, $N = 10000$ are roughly 2.3134 and 0.3005 respectively.

3 Problem 2.12

Using this Python code I borrowed from
https://nbviewer.jupyter.org/github/tournami/Learning-From-Data-MOOC/blob/master/Homework%204.html,
we can calculate the sample size with $\sigma = 0.05$, $\delta = 0.05$ and $d_{vc} = 10$

```
import numpy as np

def get_N(dvc=10, delta=0.05, epsilon=0.05, initial_N=1000, tolerance = 1):

    new_N = 8 / epsilon**2 * np.log((4 * ((2 * initial_N)**dvc + 1)) / delta)

    if abs(new_N - initial_N) < tolerance: # Did it converge?
        return new_N

    else: # If so return N
        return get_N(dvc, delta, epsilon, new_N, tolerance) # Iterate

print("Our sample size must be at least " + format(int(get_N())) + ".")
```

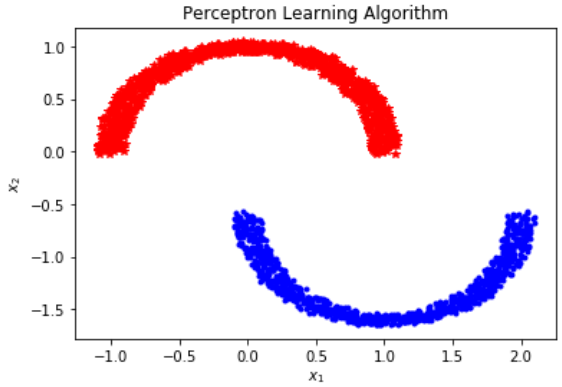
Our sample size must be at least 452956.

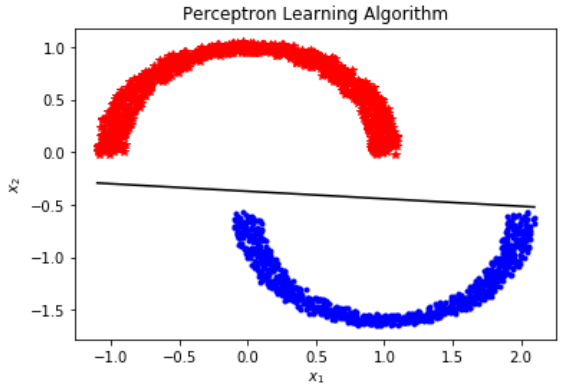
4 Problem 3.1

4.1 A

We can use the Python code in SemiCircle_PLA.py to solve part A:

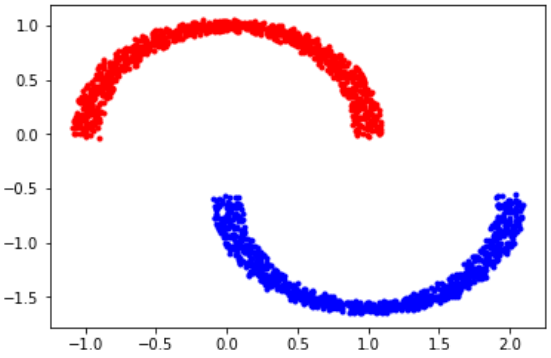
```
2000
993
96
0
Total iterations: 3
[3]
```





4.2 B

We can use the Python code in SemiCircle_Linear.py to solve part B:




```
Our W is:
[-0.32326725  0.093165   -0.9181648 ]
```