Homework 2

John Randis

9/27/2016

2.1

$$\in (M, N, \delta) = \sqrt{\frac{1}{2N} ln \frac{2M}{\delta}}$$

a)

$$.05 \leq \sqrt{\frac{1}{2N} ln \frac{2(1)}{.03}}$$

$$.0025 \leq \frac{1}{2N} ln \frac{2}{.03}$$

$$.0025 \leq \frac{1}{2N}(4.1997)$$

$$.005N \leq 4.1997$$

$$840 \leq N$$

b)

$$.05 \leq \sqrt{\frac{1}{2N} ln \frac{2(100)}{.03}}$$

$$.0025 \leq \frac{1}{2N} ln \frac{200}{.03}$$

$$.0025 \leq \frac{1}{2N} \ln(8.80487)$$

$$.0025 \leq \frac{1}{2N}(2.1753)$$

$$.005N \leq 2.1753$$

$$435 \leq N$$

c)

$$.05 \leq \sqrt{\frac{1}{2N} ln \frac{2(1000)}{.03}}$$

$$.0025 \ \leq \frac{1}{2N} ln \frac{2000}{.03}$$

$$.0025 \ \leq \frac{1}{2N} ln \frac{2000}{.03}$$

$$.0025 \ \leq 11.10746)$$

$$.005N \ \leq 11.10746$$

$$2221.492 \leq N$$

2.11

Theorem: $E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4m_H(2N)}{\delta}}$

a) N = 100:

$$\sqrt{\frac{8}{100} ln \frac{(4)(101)(2)(100)}{.1}}$$

$$\sqrt{.08\ln(808{,}000)}$$

$$\sqrt{1.088185387}$$

$$1.043161247$$

The out-of-sample error is less than or equal to 1.043161247 + E_{in}(g).

b) N = 1000:

$$\sqrt{\frac{8}{1000} ln \frac{(4)(1001)(2)(1000)}{.1}}$$

$$\sqrt{.008\ln(8{,}008{,}000)}$$

$$\sqrt{.127167612}$$

$$.356605682$$

The out-of-sample error is less than or equal to.356605682+ E_{in}(g).

2.12

Theorem: $N \ \geq \frac{8}{\epsilon^2} ln \, (\frac{4((2N)^{d_{vc}}+1)}{\delta})$

$$N \geq \frac{8}{.05^2} ln \left(\frac{4((2N)^{10}+1)}{.05}\right)$$

$$N \geq 320ln \left(\frac{4(1024N^{10}+1)}{.05}\right)$$

$$N \geq 320ln \left(\frac{4096N^{10}+4}{.05}\right)$$

$$N \geq 320ln \left(\frac{4096N^{10}+4}{.05}\right)$$

$$N \geq 320ln \frac{4096N^{10}}{.05} + 80$$

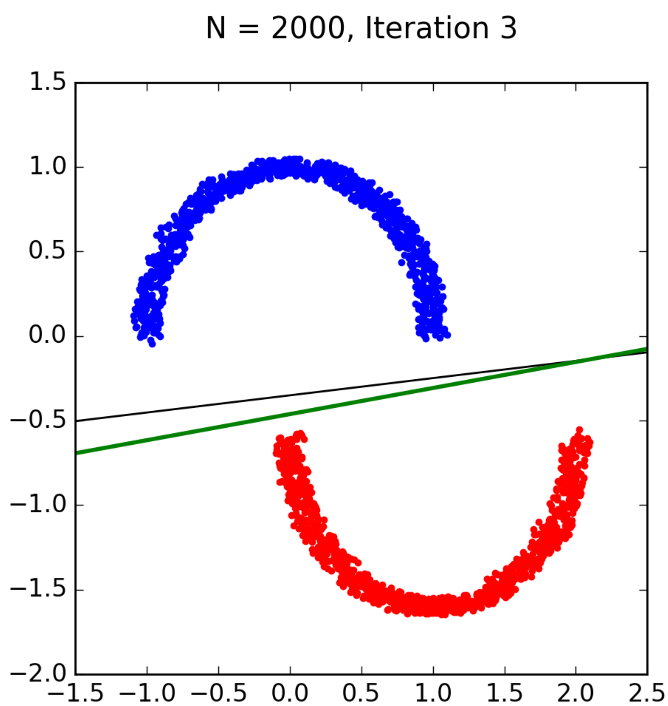
$$N - 80 \geq 320ln \frac{4096N^{10}}{.05}$$

$$.05N - 4 \geq 16ln(4096N^{10})$$

$$N \geq 36{,}344.2$$

3.1

a)



The perceptron learning algorithm took 3 iterations to converge and find the best fit line in between the two semi-circles. w is equal to [-1.   0.33415214 -2.16534827].

b)

Using linear regression, w ends up being [-0.32193933 0.09342563 -0.91802757]. It seemed to run faster than with the perceptron learning algorithm. I find it very interesting that w_0 is -1 when using PLA and -0.32193933 when using linear regression. Both w's fit the data, but they are very different results.