Alexander Palomba

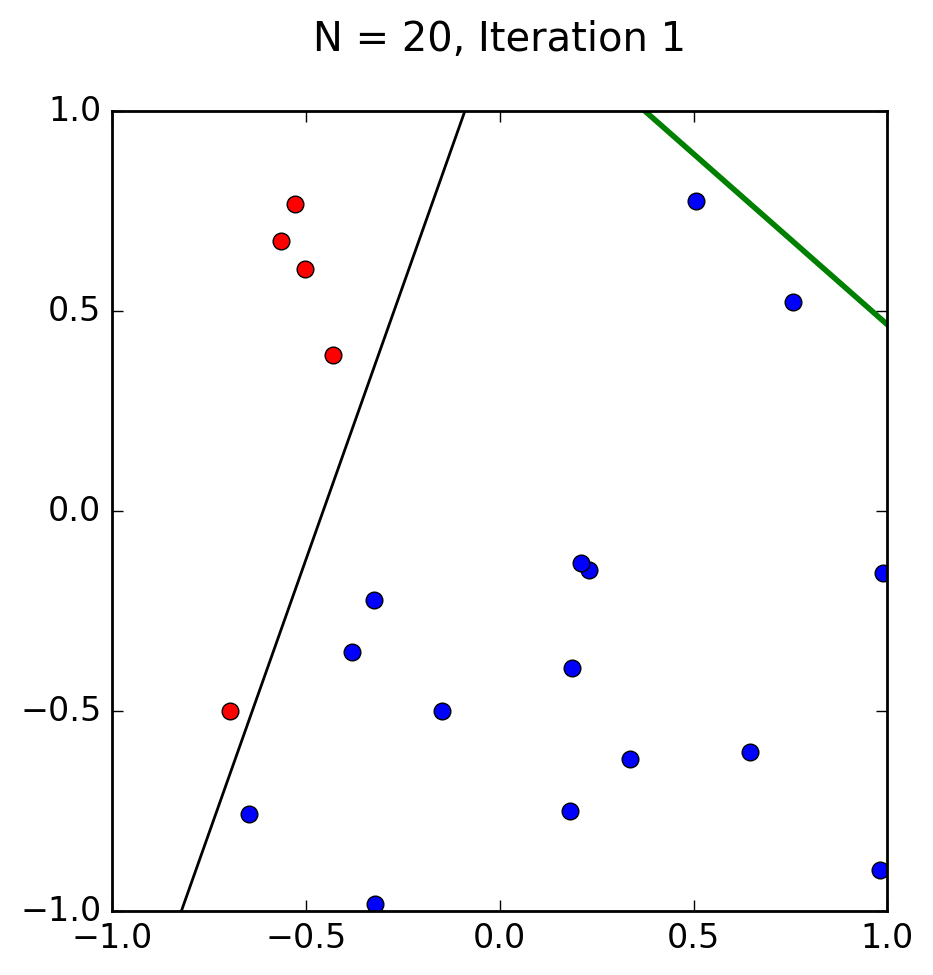
Artificial Intelligence

20 September 2016

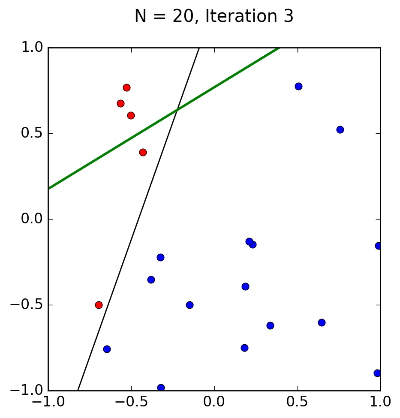
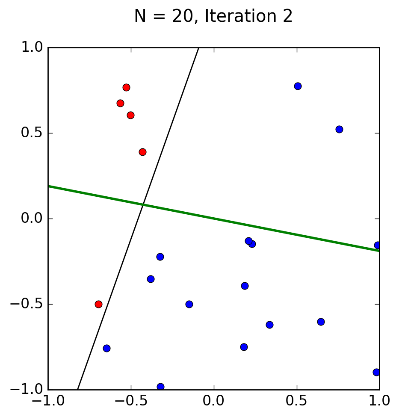
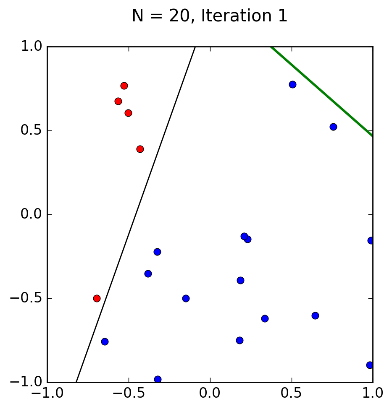
Homework 01

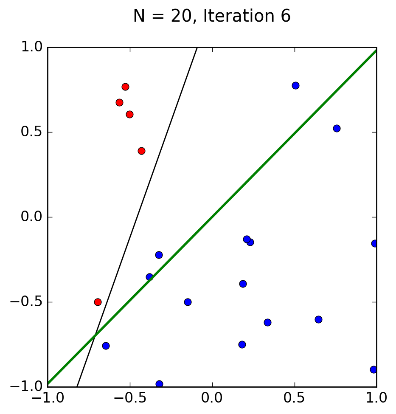
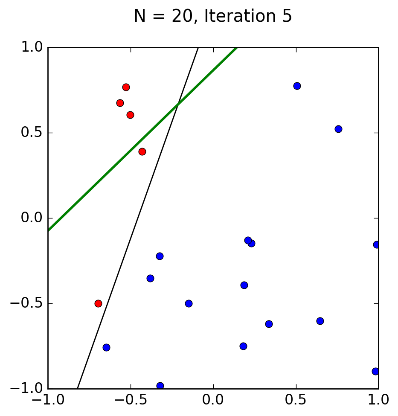
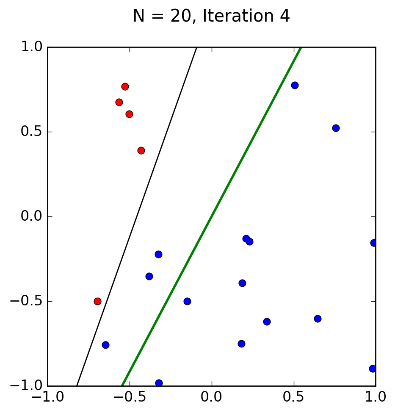
Problem 1.4

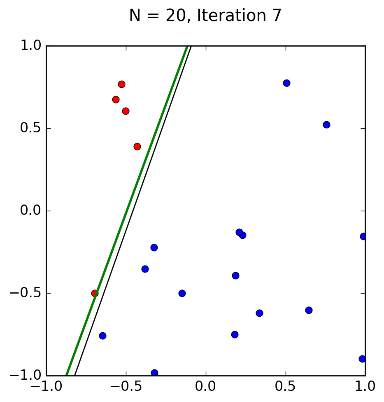
***a.***



***b.***

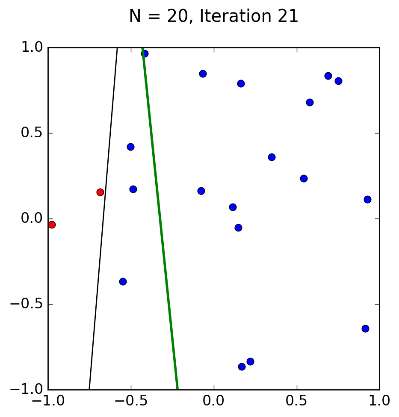
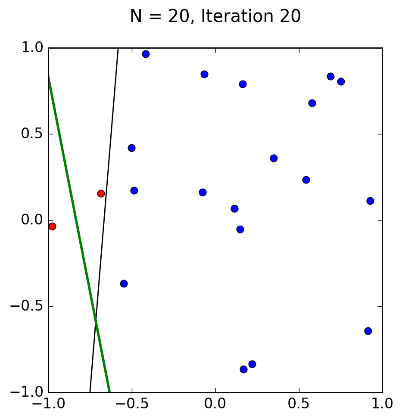
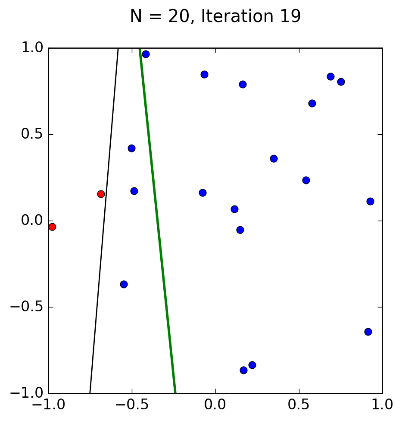
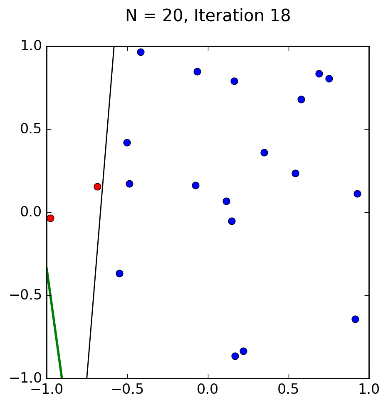
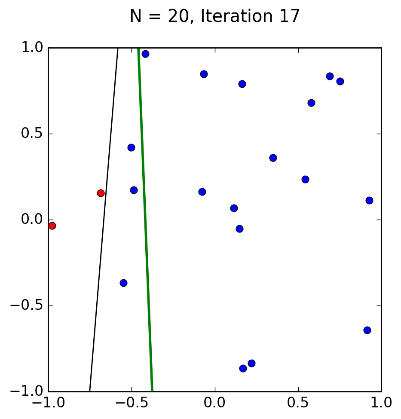
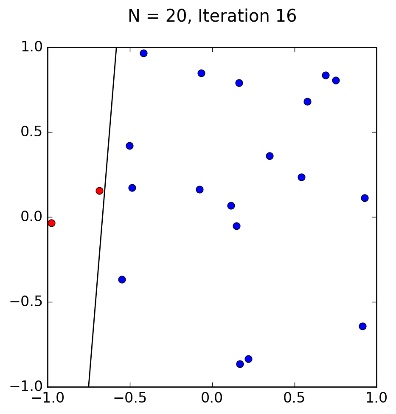
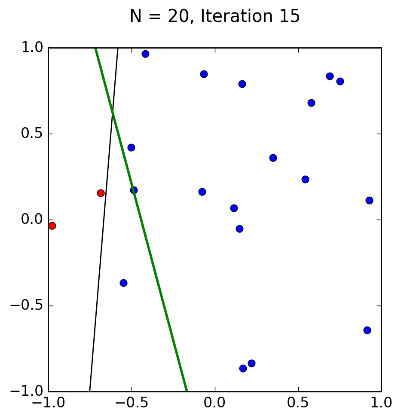
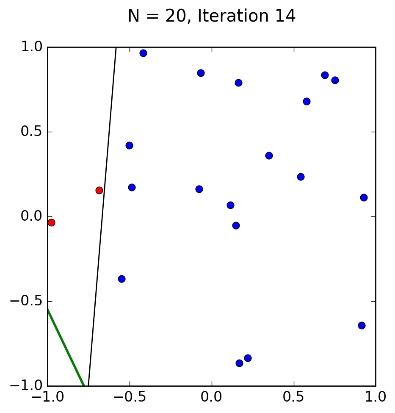
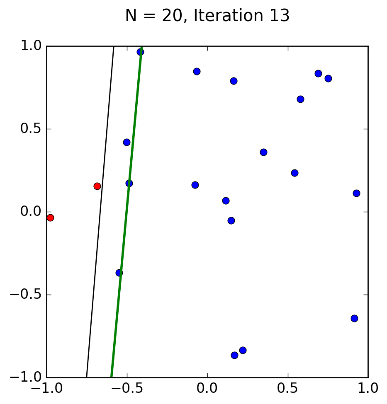
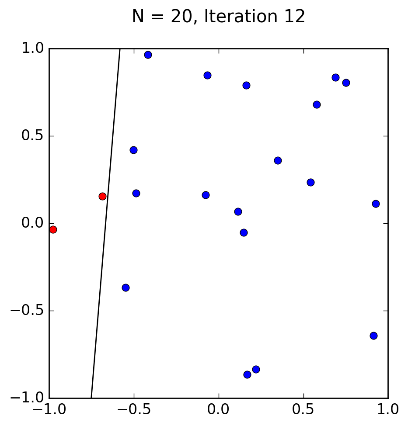
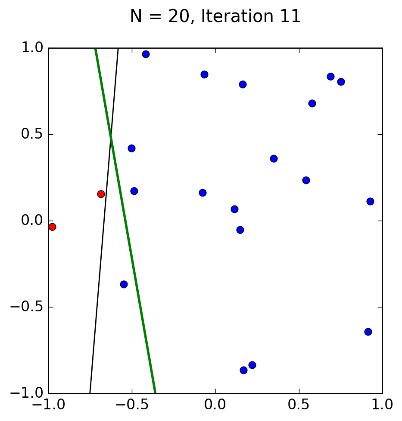
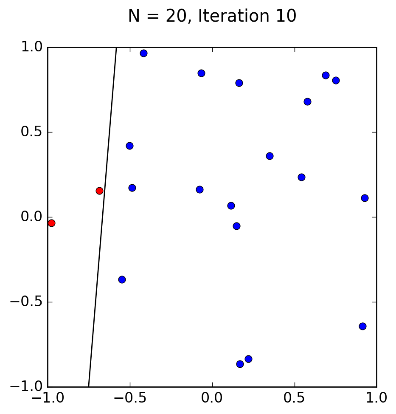
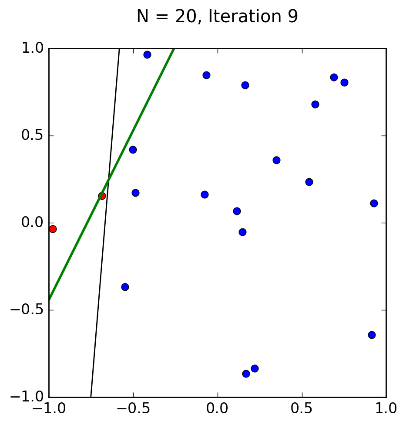
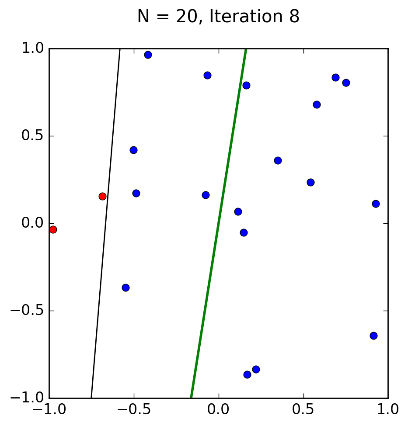
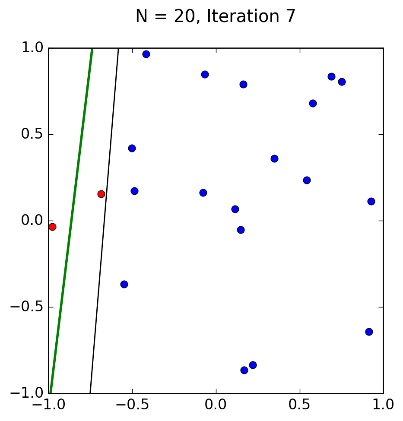
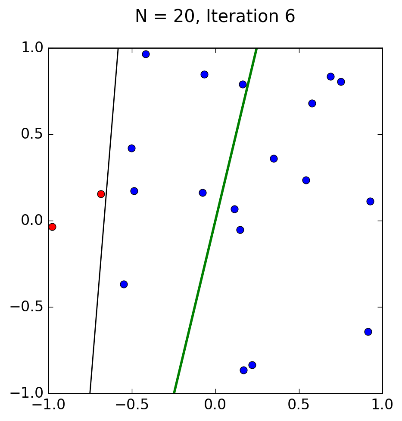
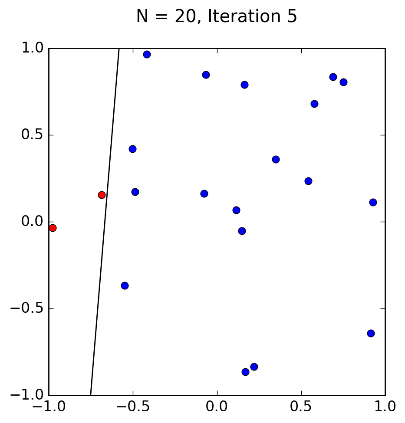
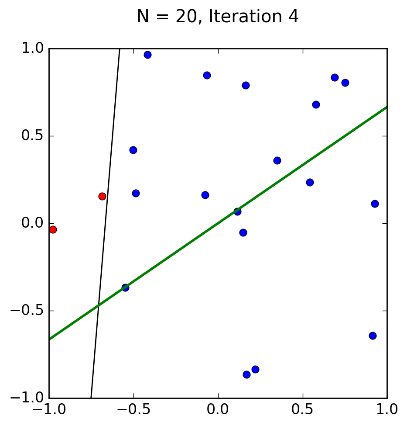
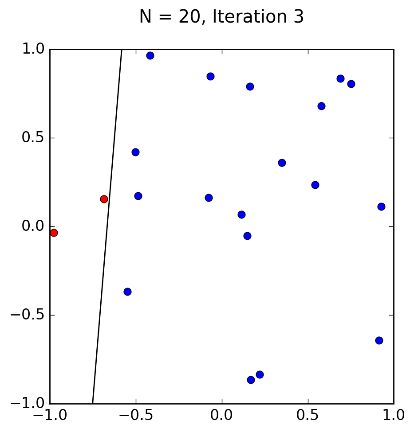
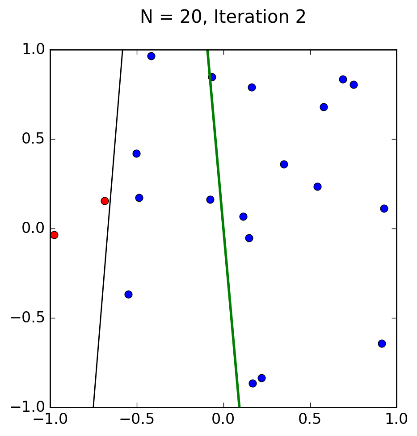
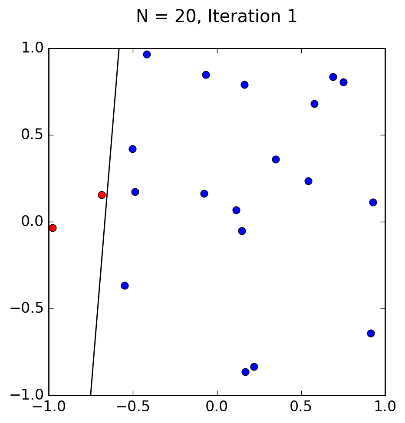


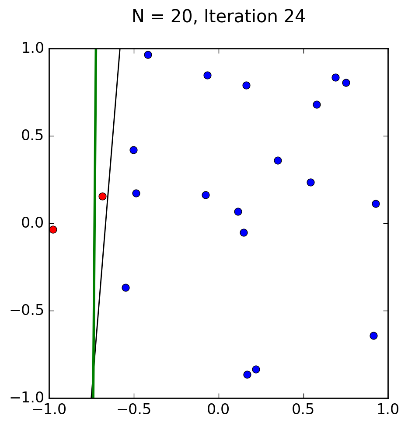
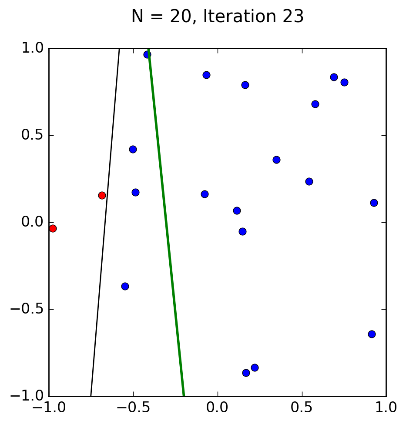
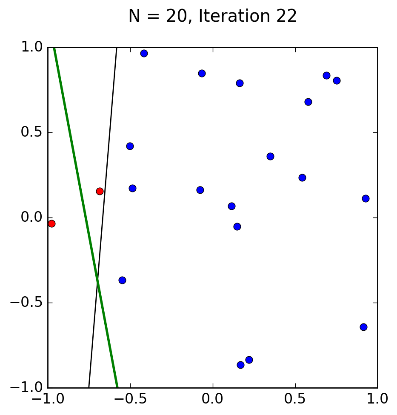


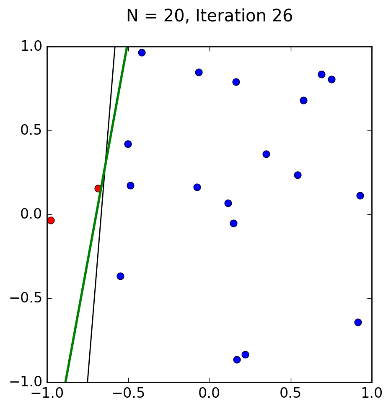


In this final iteration, *f* is reasonably close to *g.*

***c.***

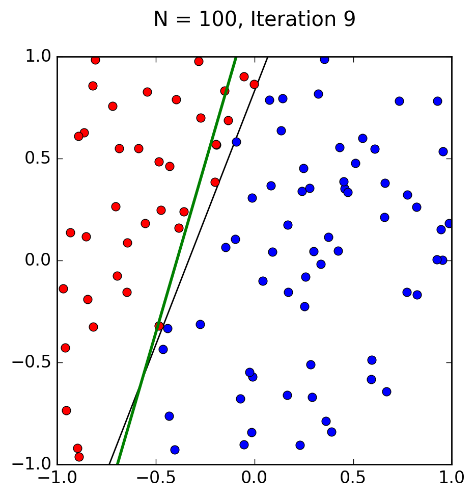
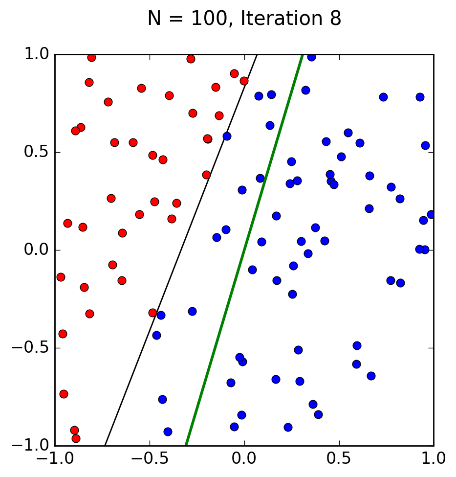
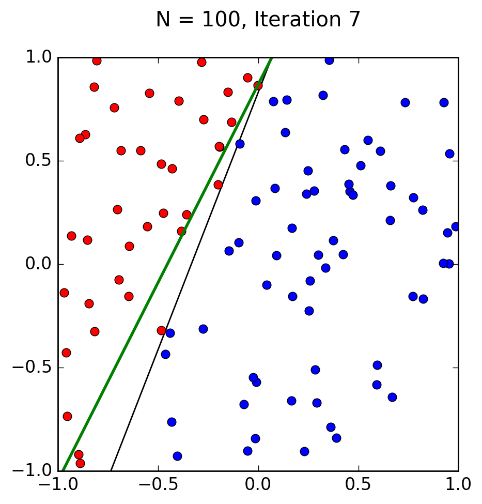
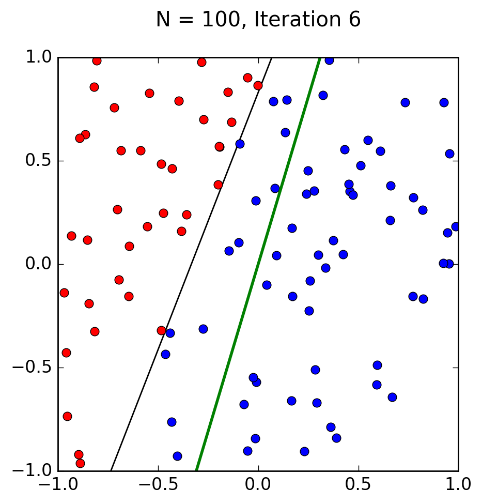
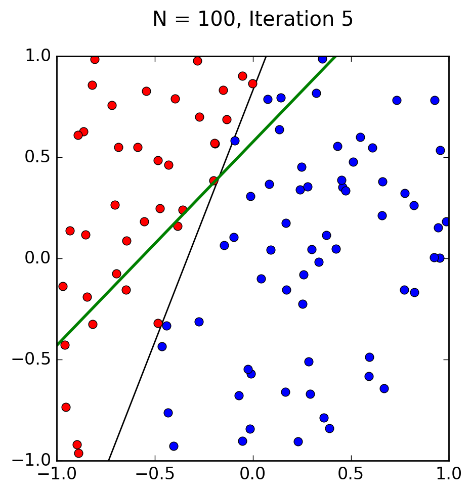
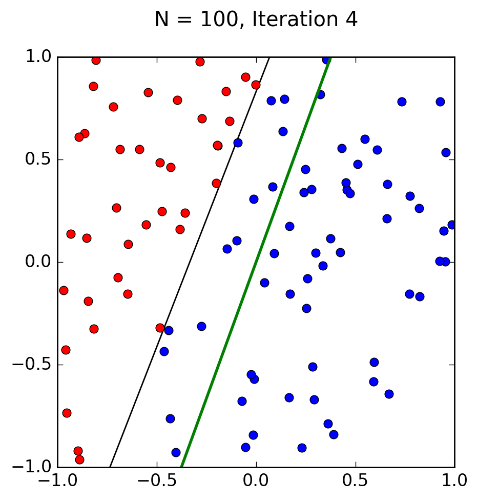
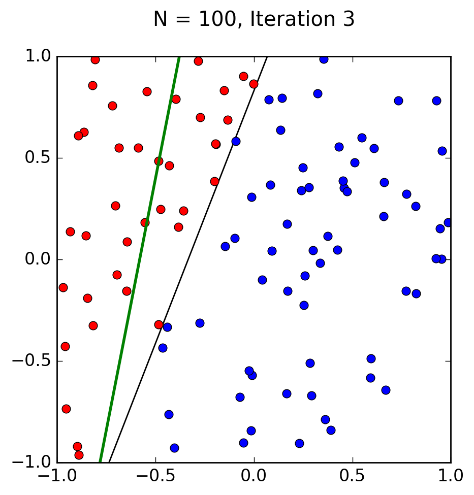
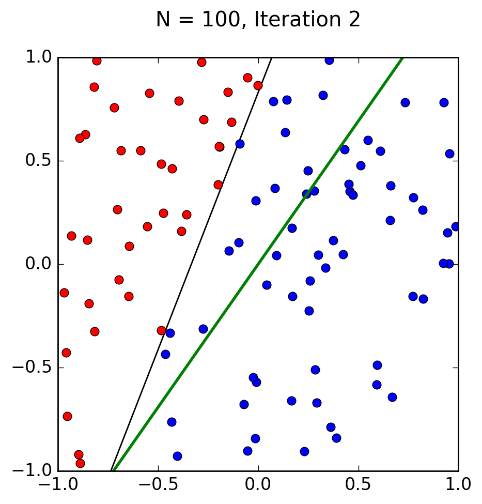
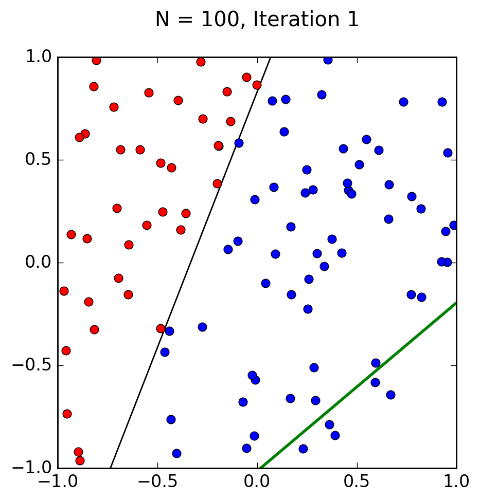


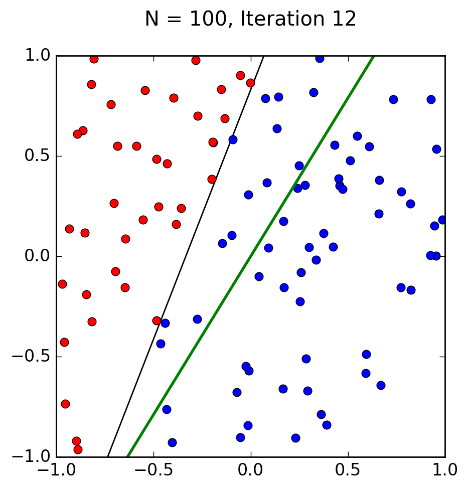
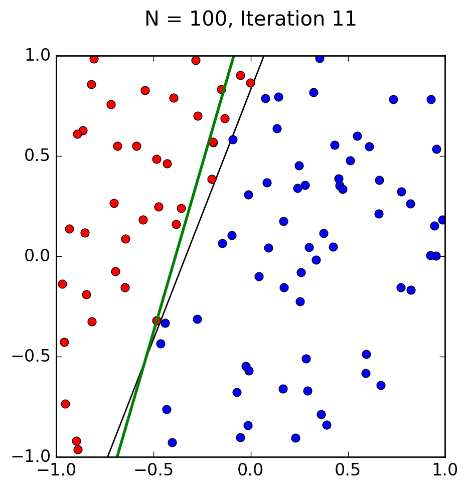
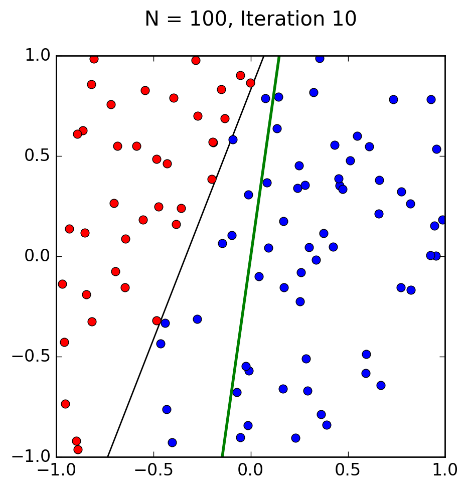


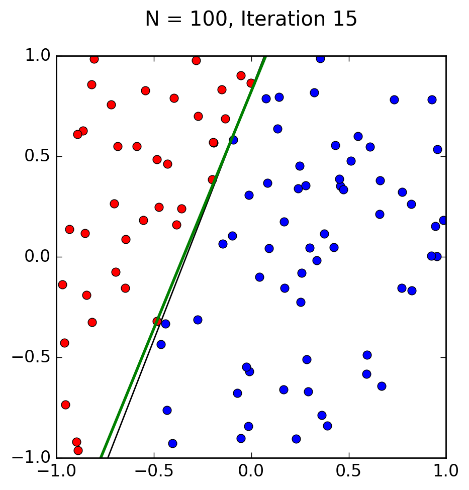
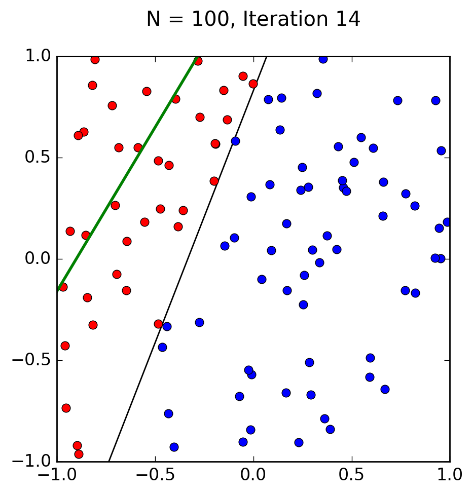
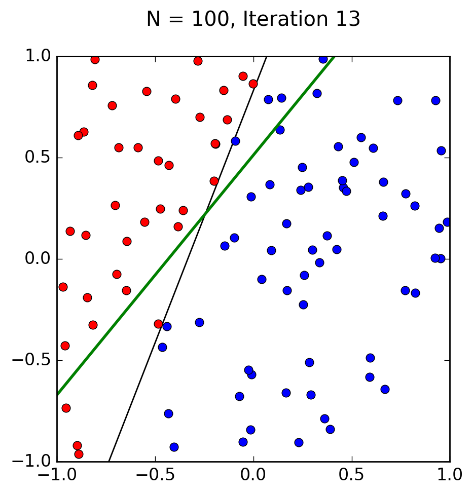


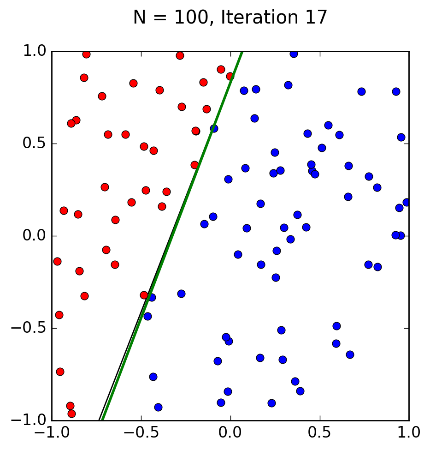
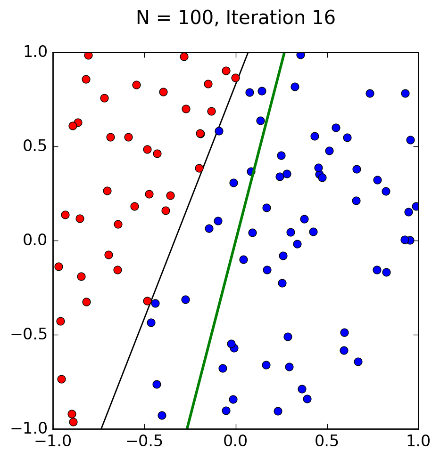
It took the perceptron algorithm significantly more iterations to reach a valid line separating the two classes, and *g* is not as close to *f* as it was in the first set.

***d.***



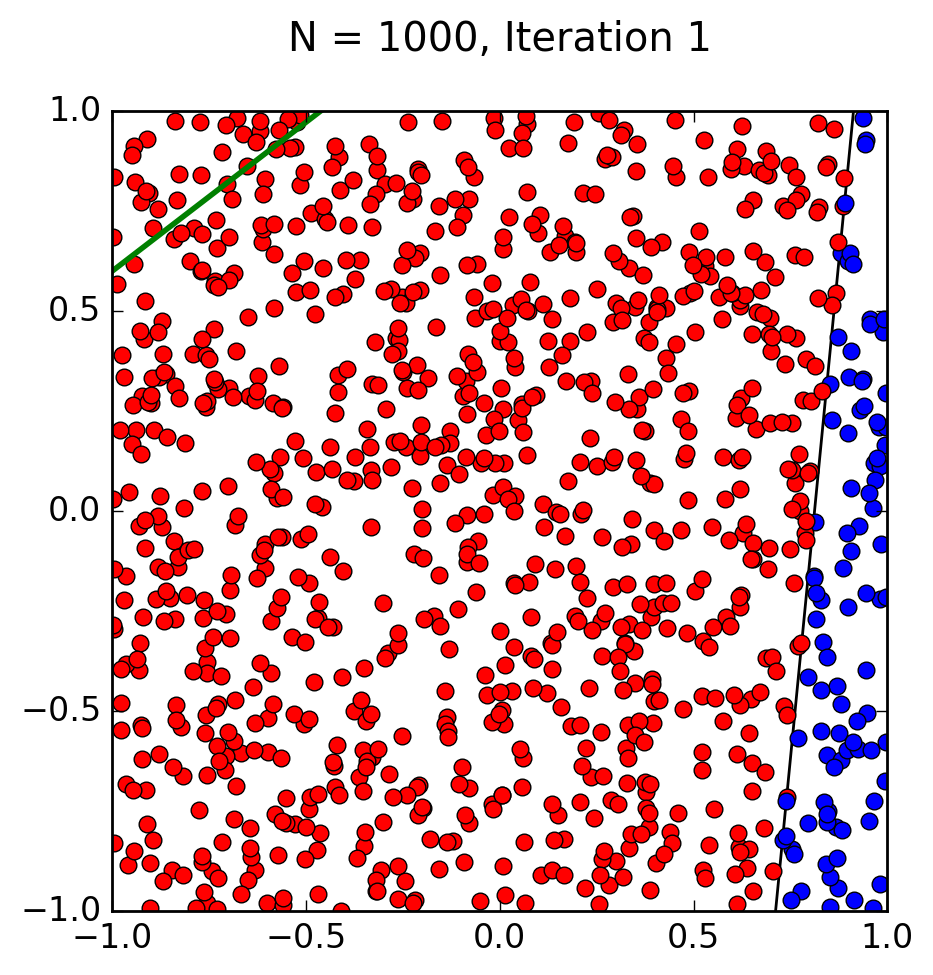


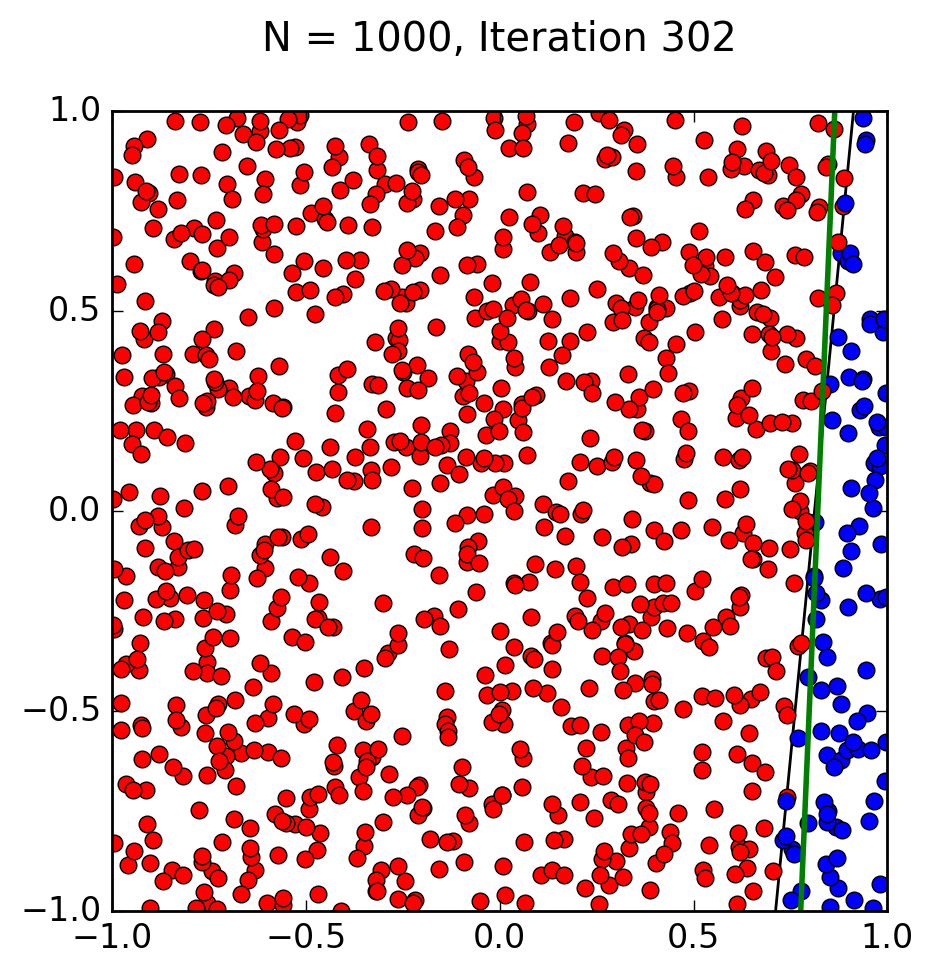
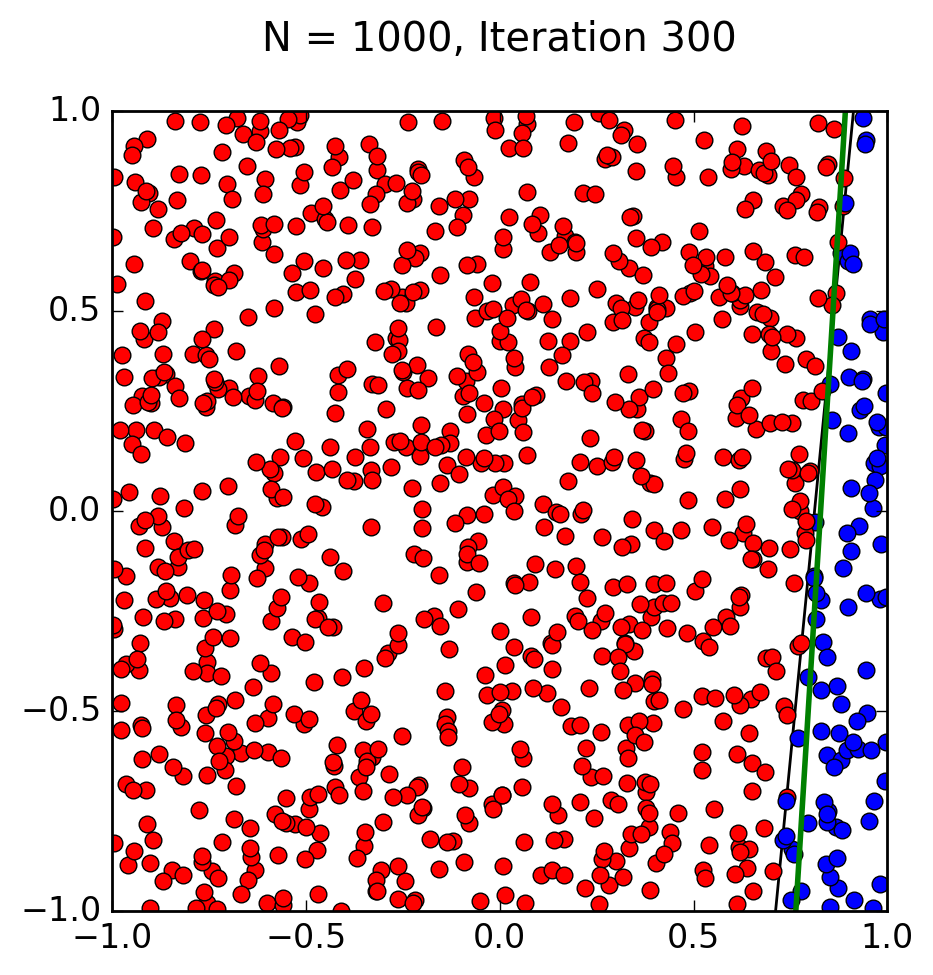
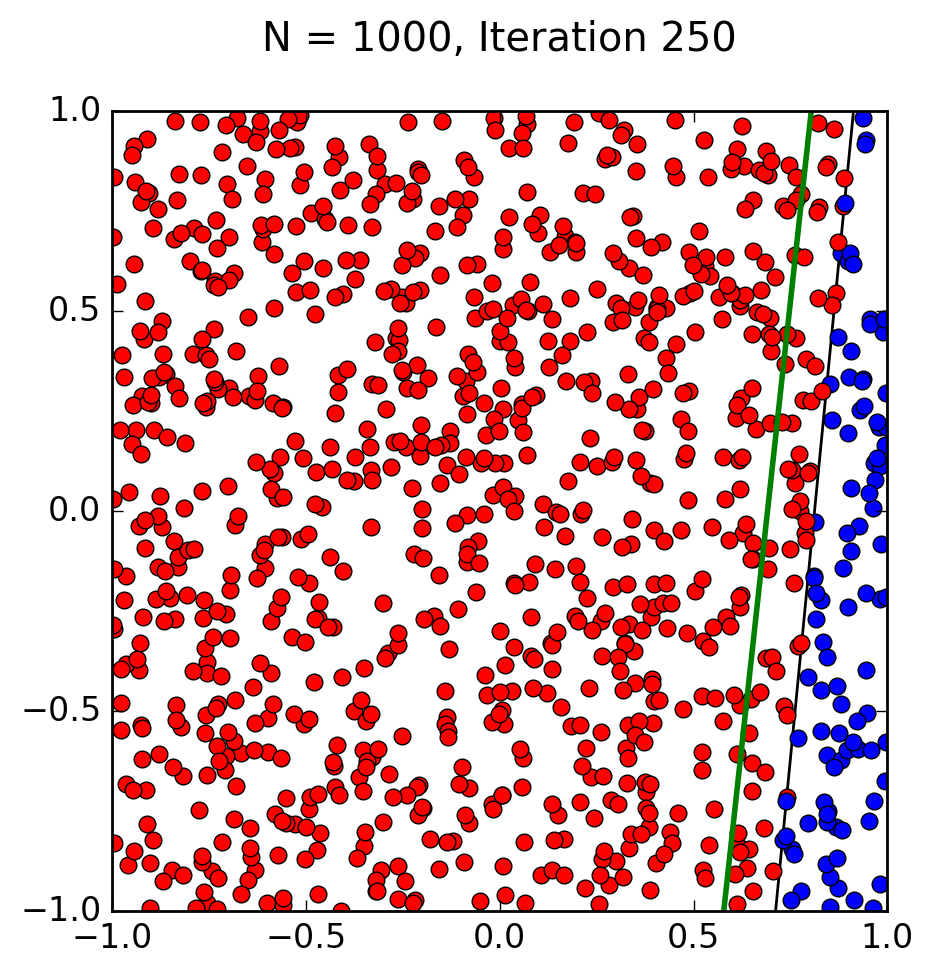
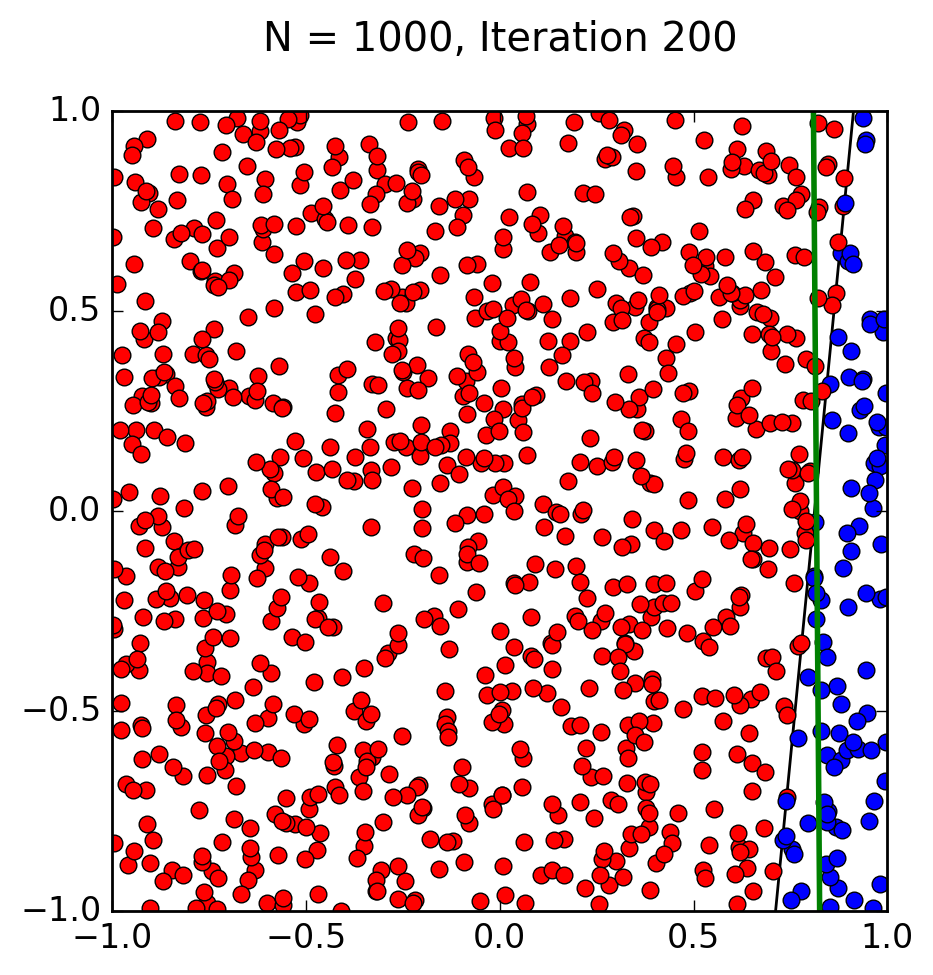
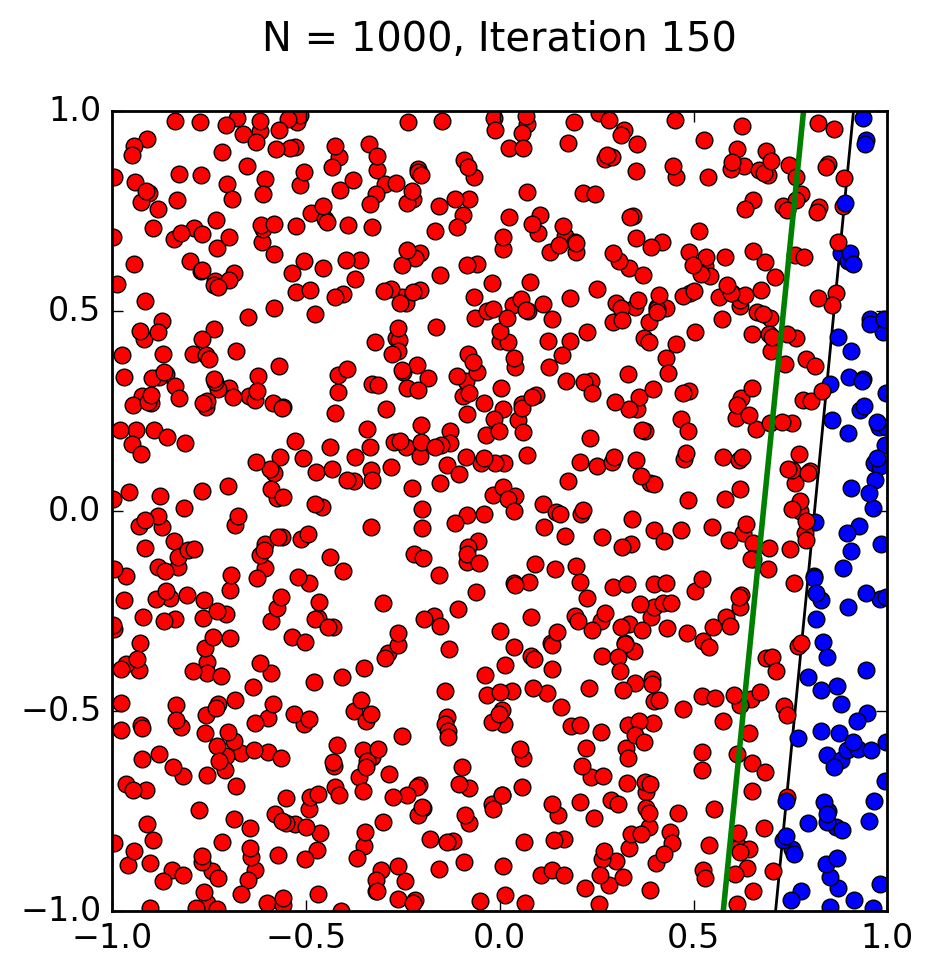
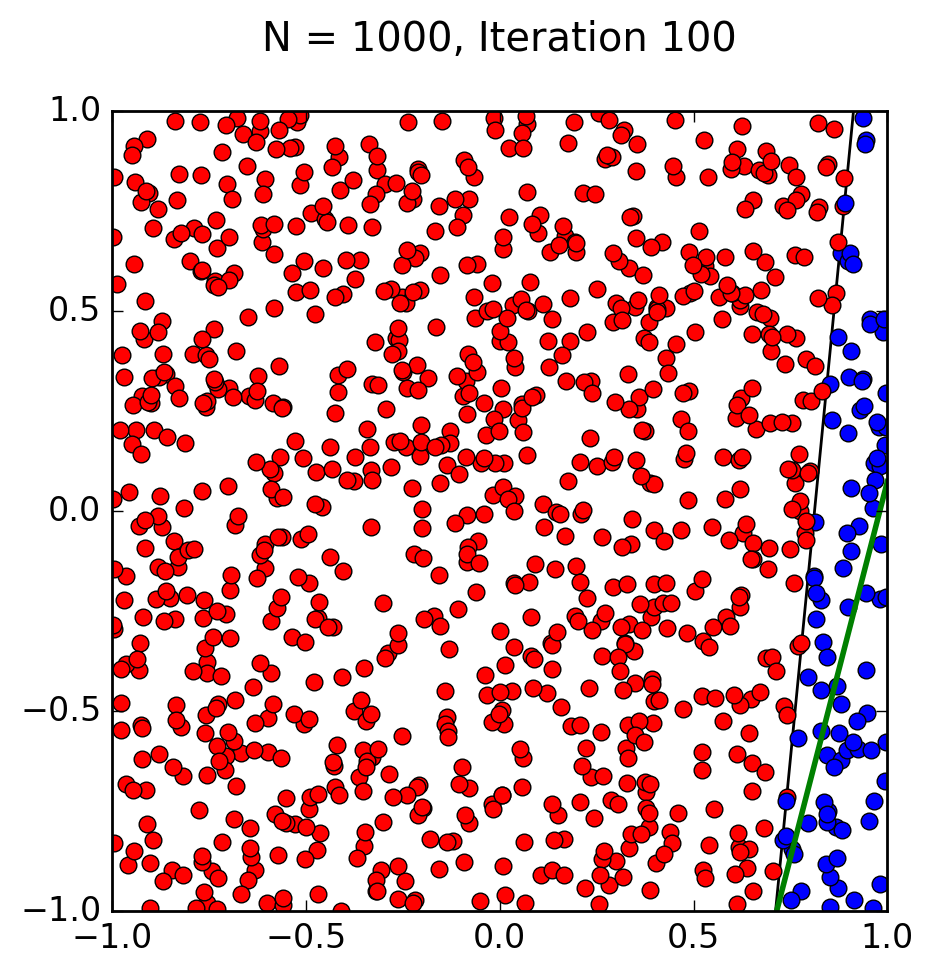
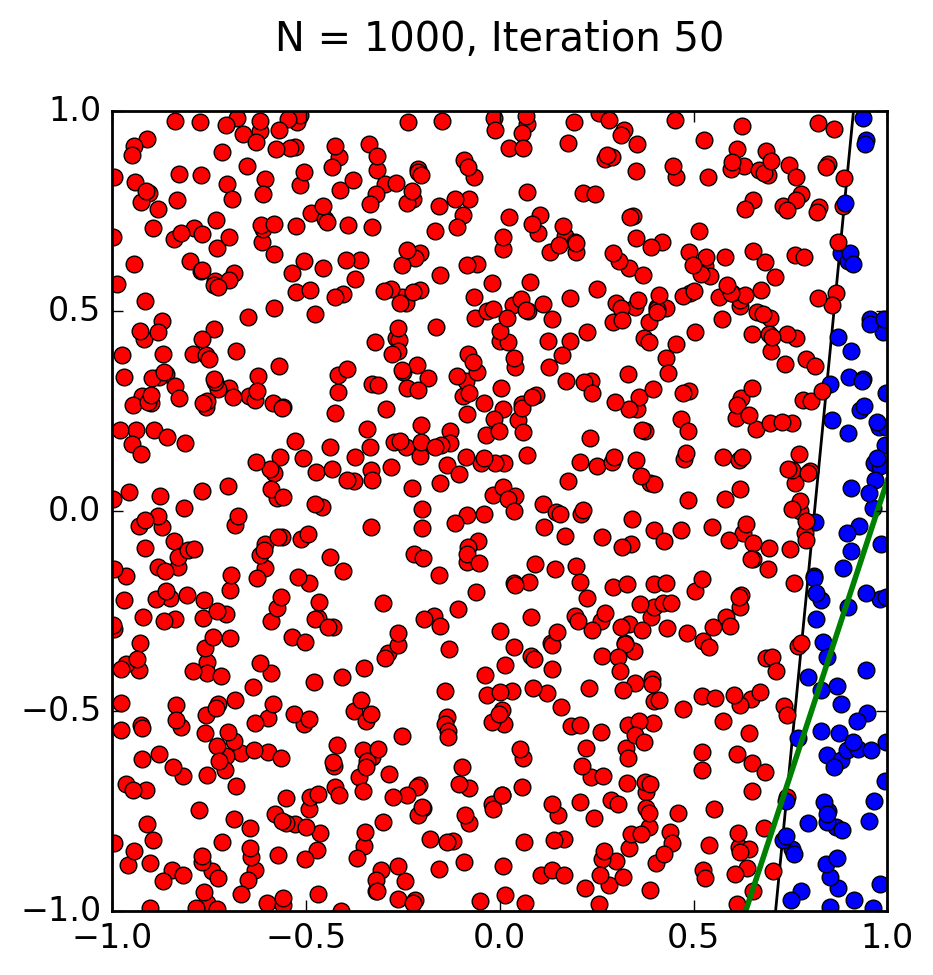




Surprisingly, the perceptron algorithm took only 17 iterations to compute a valid function for *g*, and it is very close to *f*.

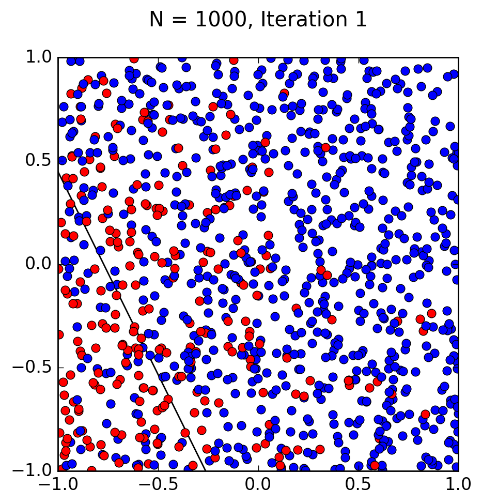
***e.***

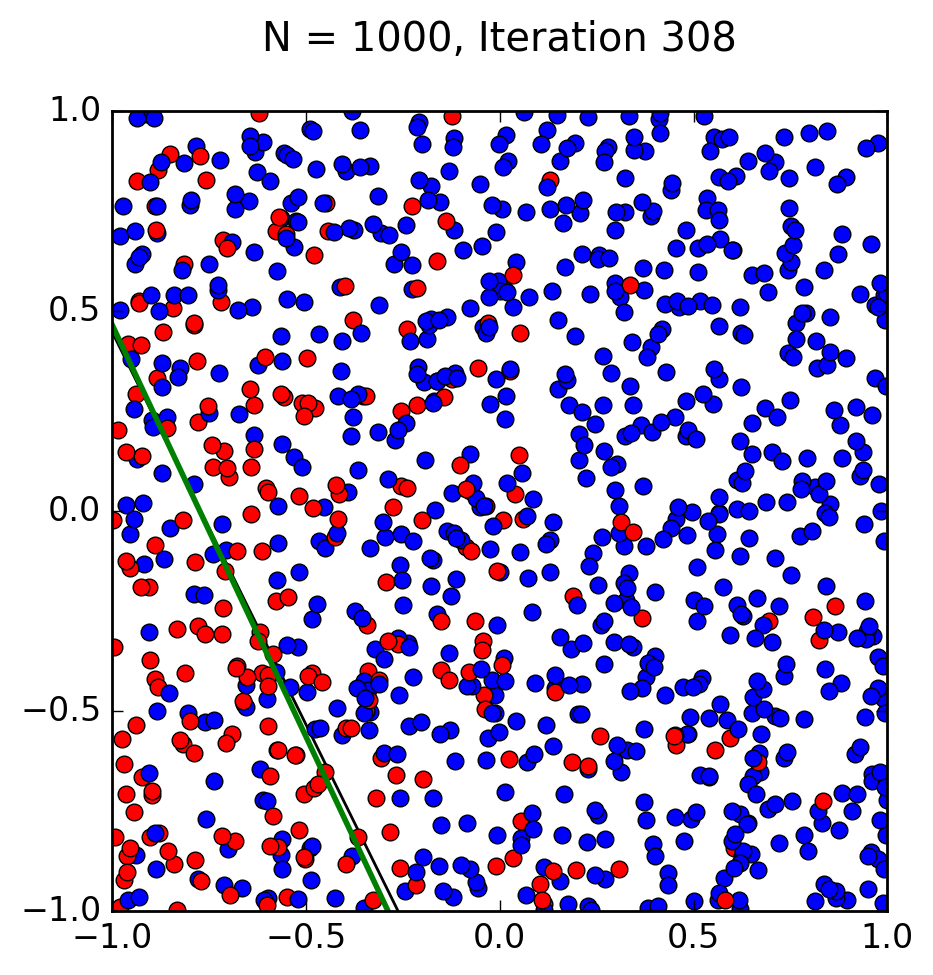
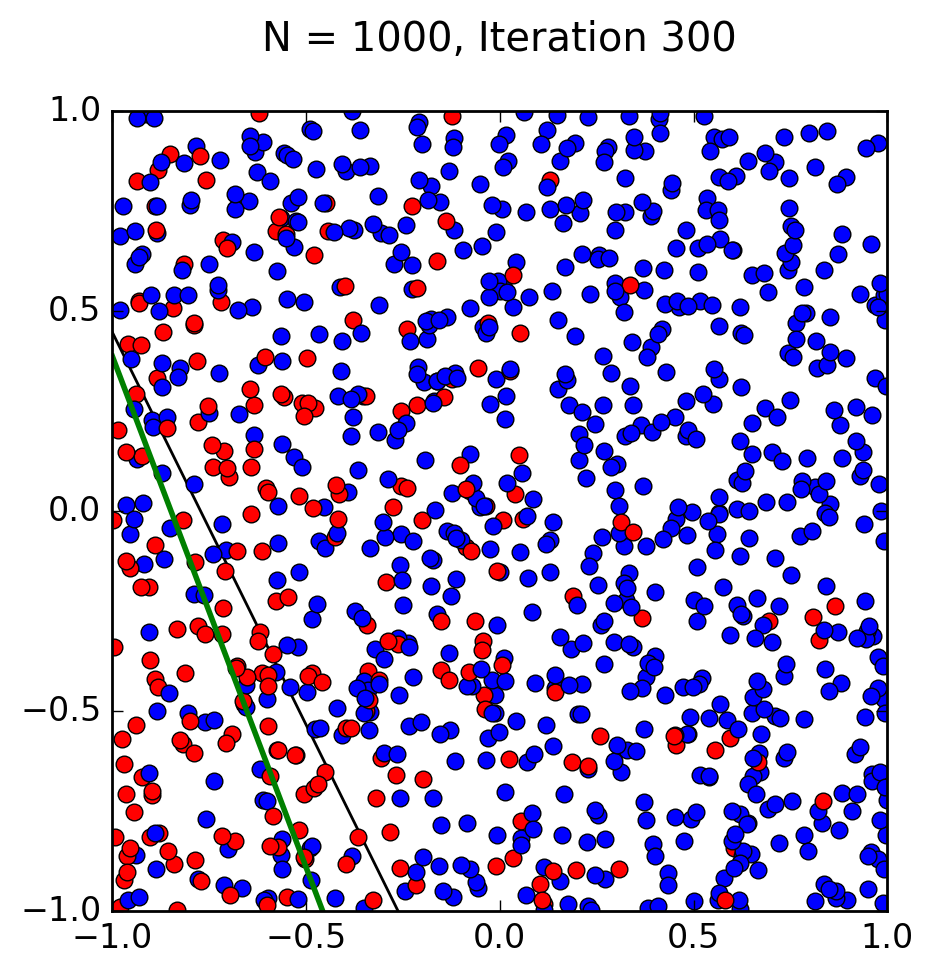
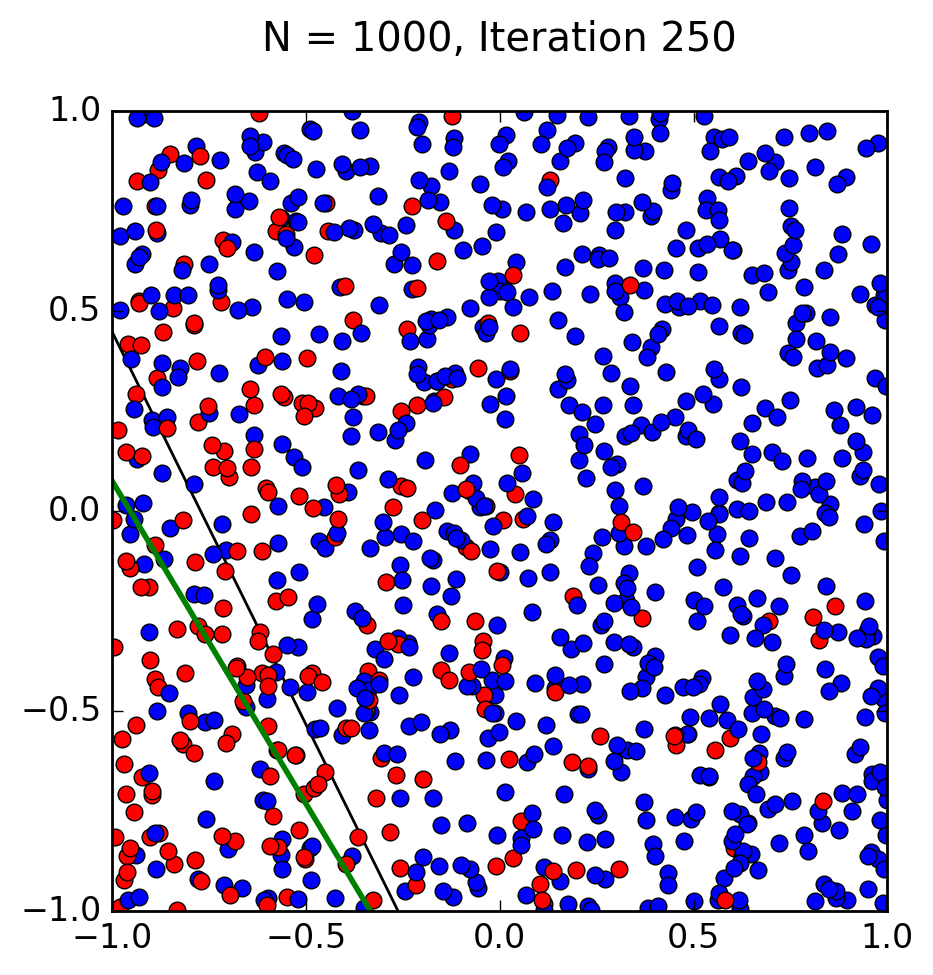
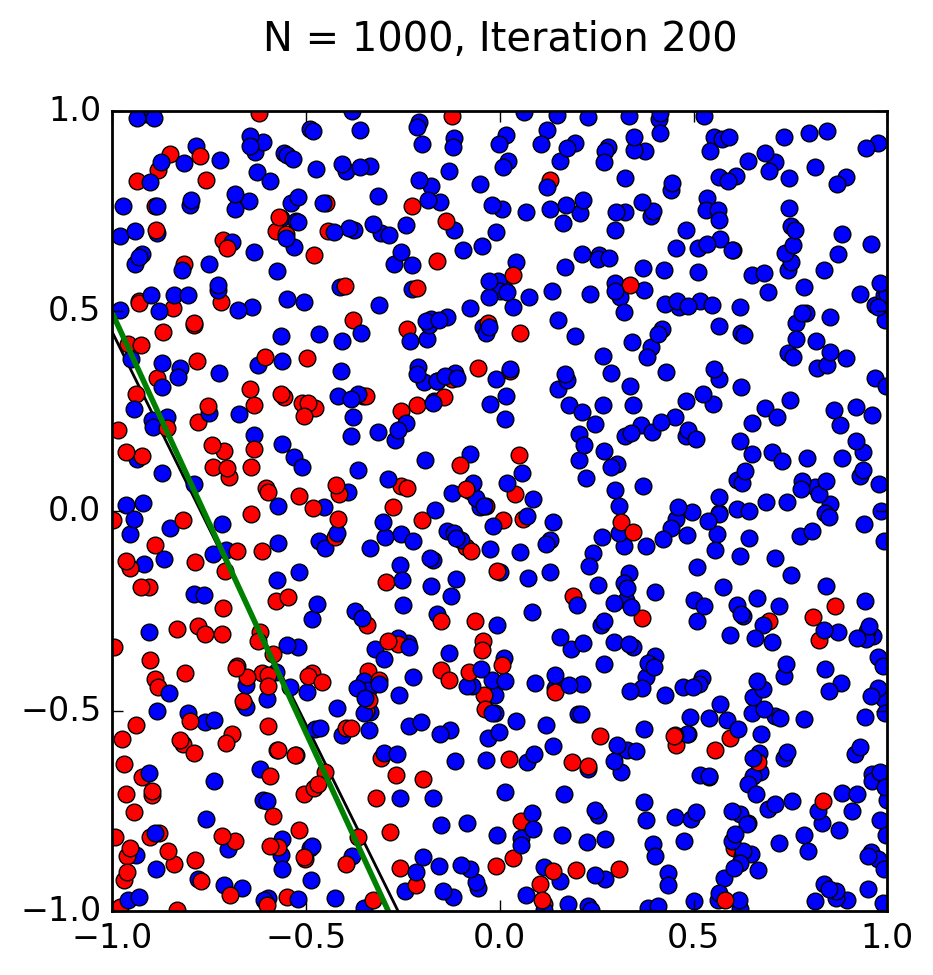
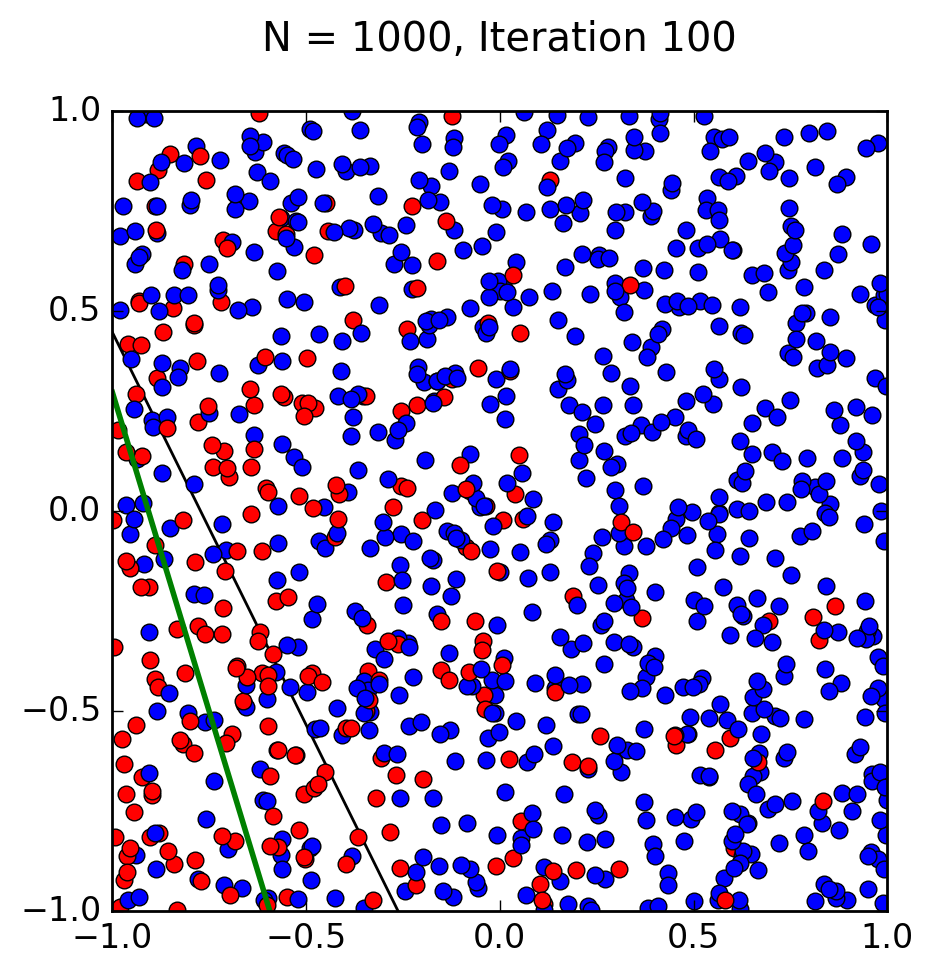
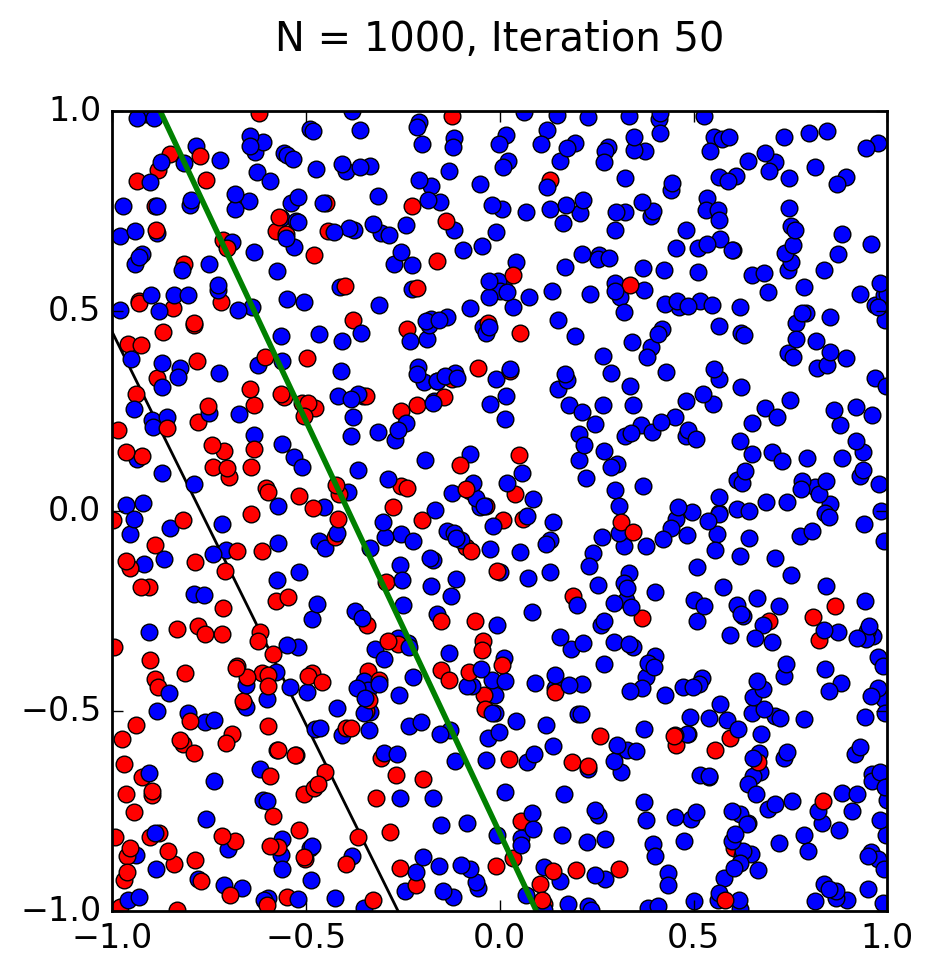




My computer just gave up after 302 iterations, but still got reasonably close to the target function.

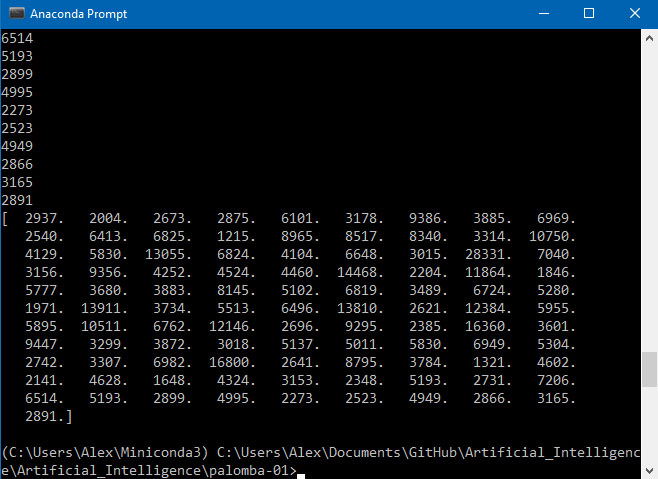
***f.***

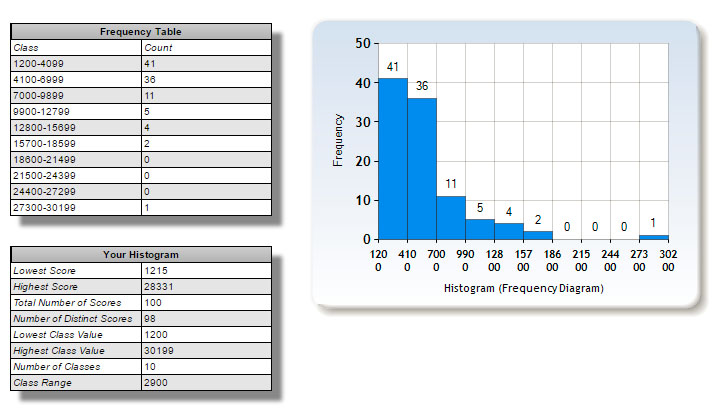




My computer stopped at 308, but considering the data that is to follow, I’m assuming this is only because my computer ran out of memory.

***g.***





***h.***

Based on all the data I was able to observe, given that *d* equals the number of dimensions and *N* the number of unique data points, the size of *d* affects the running time far more than the size of *N*; also, the accuracy of the function *g* with regards to the target function *f*, seemed to be more accurate when both *N* and *d* were small.

**Sources**

Code:

*Machine Learning Classics: The Perceptron* (<https://datasciencelab.wordpress.com/2014/01/10/machine-learning-classics-the-perceptron/>)

Pablo Rivera-Rivas (Office Hours)

Histogram:

Easy Histogram Maker (<http://www.socscistatistics.com/descriptive/histograms/>)