CSC246 Machine Learning
Homework 2: Perceptrons

Overview

The purpose of this project is to give you practical experience implementing a machine learning algorithm. You are asked to implement the perceptron algorithm and to validate your implementation with experiments. To receive credit, you will need to submit your source code and a writeup (with figures and explanations) by 1159PM, Sunday, February 16th. Submit your code via running the *sim*cs246/TURN_IN script on your assignment directory.

Datasets and Programming Languages

Your project must be completed using Python3 and Numpy. You are strongly encouraged to develop your solution using the Department of Computer Science instructional network, accessible via ssh: cycle{1,2,3}.csug.rochester.edu. If you do not have an account, you can get one here: `https://accounts.csug.rochester.edu`. If you have never used Linux (or unix, or a terminal) before, you may find the following tutorial helpful: `http://www.ee.surrey.ac.uk/Teaching/Unix/`.

The starter code is available on the CSUG network at: ∼cs246/pub/perceptrons/perceptron.py

You are also supplied with several datasets (in perceptrons/data/):

- linearSmoke.dat
- xorSmoke.dat
- challenge{0..4}.dat

The first two files are meant to help you during debugging. Perceptron training should rapidly converge on the linearSmoke dataset. This is an example of an easy dataset. In our testing, it converged in five iterations. Perceptron training should never converge on the xorSmoke dataset. This is an example of a hard dataset, even though it is small.

The remaining challenge datasets are meant to pose a challenge. At least two of them are in fact linearly separable. The remaining three are probably (but not certainly) not linearly separable. Once you are confident in your implementation (i.e., because it behaves as expected on the smoke test datasets), you should begin experimentation on the challenge datasets.

Experimentation

For each dataset you must identify whether it is definitely linearly separable or not.

For the linearly separable datasets, you should report the iteration at which your algorithm converges. You should also report the maximum vector norm over the dataset (i.e. the value R from the convergence proof discussed in class), and using the relation $k < R^2/\delta^2$ report an upper bound on the value for the separting margin delta. You should also produce a plot indicating accuracy per iteration (i.e., x-axis is number of iterations, y-axis is overall accuracy on the training data). This should include all iterations up to convergence.

For the non-separable datasets, you should report the maximum number of iterations you explored, the overall highest accuracy obtained (i.e., the maximum over all the iterations — since perceptron updates can termporarily make accuracy go down, this may not be the value of the last iteration), and a plot of accuracy by iteration. You must also include a short explanation of why you believe you have tested the data "enough" (i.e., by making reference to the graph or accuracy over time.)

Grading

Your submission will be graded according to the following approximate rubric:

- 50% – program correctness
- 5*10% – results per dataset
  - for linearly separable datasets:
    - correctly identified as linearly separable
    - plot of training until convergence
    - lower bound for δ
  - for non-separable datasets:

* training plot with max iterations and peak accuracy clearly labeled
* description/analysis of behavior (e.g., asymptote after N iters)

Note – please take care to make your writeup clear and concise. It should clearly identify your results. If the association between a plot/analysis and a dataset is not clearly indicated, you will receive zero credit for that portion.