Homework 2, Logistic Regression (22 points)

We will implement a logistic regression for multi-class classification. The major components include.

* The project will be due on **11/12/2017 midnight.**
* Late submission will not get credits.
* Submission via emails will not get credits.
* If you cannot finish all, submit solutions for some problems to get partial credits

## 1, Prediction

For a given weight w and a new feature vector x, predict the class. Implement in python. Use matrix operation so you can do prediction on a set of test data without loop.

## 2, Training

Learning the weights by minimizing the cross-entropy loss function.

1. Derive the gradient formula.
2. Implement functions to evaluate the gradient and the loss. Remember to use matrix operation.
3. Implement the steepest descent (using gradient with a stepsize)

* Hint: start with an initial stepsize (parameter to tune), increase the stepsize by a factor of 1.01 each iteration where the loss goes down, and decrease it by a factor 0.5 if the loss went up. If you are smart you may also undo the last update in that case to make sure the loss decreases every iteration.

1. To make sure the gradient is calculated correctly, we often validate by comparing with finite-difference approximation. See the following link for a detailed explanation.

<http://ufldl.stanford.edu/tutorial/supervised/DebuggingGradientChecking/>

Use scipy.optimize.check\_grad. Assuming a correct evaluation of the loss function, if your gradient is implemented correctly, the output should be very small. Of course this function is very expensive, thus you should try it on a small scale data (say try it on synthetic dataset, which only has 2 features).

## 3, Evaluation Metrics

Evaluate by comparing different baselines, different parameter settings. The point is to see both the accuracy and the efficiency.

1. Accuracy: number of correctly classified test data over the whole dataset. (Function is already implemented for you.)
2. Convergence rate: plot the loss function as a function of the number of iterations. You should see a converging curve. Try this for different initial learning rates, draw different curves.
3. Measure the average time for each iteration of training, and for prediction (on different datasets).

## 4, Data

Different datasets have been provided.

## 5, Grade-breakdown

Partition the report into two sections: binary classification and multi-label

1. binary-label implementation (12 pts)
   1. Successful implementation on binary datasets
   2. In report, write down the math formula. Explain implementation details in the report: the usage of each function, especially the ones you added.
   3. Report accuracy and average running time (for an optimally selected initial stepsize) for all given binary-labeled datasets.
   4. convergence rate for different initial stepsizes on binary synthetic-easy/medium/hard datasets.
2. Multi-label implementation (10 pts)
   1. Successful implementation on multi-label datasets
   2. In report (second section), write down the math formula. Explain implementation details in the report: the usage of each function, especially the ones you added.
   3. Report accuracy and average running time (for an optimally selected initial stepsize) for all given multi-labeled datasets (use nclasses=3 for synthetic datasets).
   4. convergence rate for different initial stepsizes on multi-label synthetic-easy/medium/hard datasets.

## 6, Submission Requirement (Please read carefully, otherwise you may lose points).

You are supposed to submit a report and code for running the experiments. In the report, the following sections are required:

* All files should be saved in a folder and then packed into a single .zip file and be submitted via blackboard (NOT .rar or .tar.gz).
* The folder name (before compression) as well as the final zip file name should be “FirstName-LastName-HW2”.
* Do not change the key function names and input/output: mytrain\_binary, mypredict\_binary, mytrain\_multi, mypredict\_multi
* In the zip file, include a text file: README.txt. Write down which problems have you finished. Also write down on which platform (mac, linux, window) is the code compiled and executed.
* This is NOT something you can finish in three days. To understand the problem itself takes quite some time. You have to start as early as possible.