Introduction to Machine Learning - COS 324

Programming Assignment 2

Due date: 11:59:59pm on 13th March. Submit your completed .ipynb file on Codepost, under Programming Assignment 2.

IMPORTANT:

- Consulting with other students from this course is allowed. If you do so, clearly state (in a comment in your code) whom you consulted with.

- Searching the internet or literature for solutions is <u>prohibited</u>.

In this assignment you will need to implement portions of python code. The end result should be a fully linear regression in python that runs fast and stops after at most 100 epochs.
The given notebook contains functions for generating some linear data, then adding noise to it. You are also provided with functions for logging and some skeleton code (in the linear_regression_cd function) for using linear regression. Your goal will be to fit a linear regression to the noisy data, filling in the necessary functions.
Note: some of the ACT's might seem ambiguous at first: to get an idea of what the intended behavior is, you may look later in the code to see how that particular ACT is used. If it's not used later in the code, then you can assume it's meant as an internal function for your own use (a.k.a. it will be used in another ACT).

1. Replace ACT1 - ACT8 with your own code.
   Each prompt requires only a few lines of code, possibly a single line.

2. Implement and **test** one ACT at a time. Unlike PA1, in this assignment you must write your own asserts to verify your code's correctness. The asserts don't need to be elaborate, so long as they reasonably show that your ACT is working.
   Make sure your implementation is correct prior to moving to the next ACT.

3. Use the code under Main (the last cell) to generate data & debug your implementations of ACTs.

4. Once all ACTs run correctly, execute the notebook in full.

5. Document your code, adding comments to your completed ACTs.

6. Submit a notebook that includes printouts and graphs generated by executing your entire notebook. Remove all code used for debugging and excess printing.