Homework 1

Deadline: Wednesday, Sept 15
Points Available: 60

Instructions

- Preferably, create an 1 notebook file. Or, create a PDF and submit code files separately.
- You document should be similar to my notes for Week 2 in GitHub, containing three parts:
  1. Typed work or pictures of your written work for mathematical questions.
  2. Text explanations and well-commended code for the other questions.
- Submit your notebook or PDF + code files (+ any other files, if needed) in Canvas.
  - Any language I can easily run is acceptable, but I highly recommend Python due to the built-in functions and compatibility with code from my notes.

Problems

1. Find a formula for the exact gradient of the sum of squares loss function based on a linear regression model with $d + 1$ weights. [5 points]
2. Write an implementation of linear regression by gradient descent using the exact gradient formula **instead of** a function approximating the gradient (e.g. `computeGradient`).

   Follow the scikit-learn structure where the classifier is an object from a class (in the computer science context) with `fit` and `predict` functions. [10 points]
3. Add elastic-net regularization to your implementation. Hyperparameters λ_1 and λ_2 should be inputs to the class or the `fit` function.

   [*Hint.* Don't forget to adjust the gradient calculation.] [5 points]
4. Load the diabetes dataset from scikit-learn.[1] Tune the hyperparameters of the model to predict disease progression as well as possible. Run the model at least ten different options for the hyperparameters, and document your performance. [10 points]
5. Prove $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ for the sigmoid function σ. [5 points]
6. Repeat problems 1-2 with the logistic binary classifier. [15 points]
7. Download and read in a credit default dataset from the UCI Machine Learning Repository.[2] Tune the hyperparameters of the logistic binary classifier to predict credit defaults as well as possible. Run the model at least ten different options for the hyperparameters, and document your performance. [10 points]

Note: For problems 4 and 7, use random dataset splits of 60%/20%/20% train/dev/test sets. When tuning hyperparameters, use the dev set. At the end, use the test set.

[1] See https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html for documentation and a link to the dataset outside the scikit-learn library, if needed.

[2] See https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients.