CSE 416: Introduction to Machine Learning
Assignment #3
Instructor: Pemi Nguyen
due: Friday, April 22, 2022, 11:59 pm.

Instructions:

- **Answers:** Please provide detailed yet concise explanations for your work.
- **Turn-in:** Do not write your name on your pages (your Gradescope account will identify you to us) and do not include a copy of the exercise's question in what you turn in. You must use Gradescope to upload your written homework solutions. You will submit a single PDF file containing your solutions to all the exercises in the homework. You must follow the Gradescope prompts that have you link exercise numbers to your pages. **We do not accept handwritten solutions**, so please typeset them, such as Microsoft Word or LaTeX editor.

1. Please determine whether the following practices are good / correct or not. Please explain why in either case.

   (a) Pranav, a software engineer at Google, is training a logistic regression classifier to detect spam emails, and he uses gradient descent to learn the weights for the model. On his first attempt, he sees the training loss after each iteration keep increasing and never seem to converge. He determines that the learning rate is too small, and that he should increase it.

   (b) Jerry, an animal lover, has a dataset of 10 million data points that contain images of 4 types of animals: dogs, cats, horses, rabbits, which has 4, 3, 1, and 2 million data points respectively. He plans to train a model to detect these different animals from images. His current trained model achieves a validation accuracy of 26 %, which he claims to be a good enough baseline model because it's better than a random draw (1/4 = 25%).

   (c) Pemi is writing down the types of errors that can occur from his model to detect whether a patient has cancer. "False positives would result in patients with cancer thinking they don't have it, and false negatives would result in cancer-free patients thinking they have it and expensive, unnecessary medical treatment," he writes. He adds that we should consider both false negative and false positive errors to improve the performance of this model.

   (d) Amal works for a small company with limited computing resources. He's supposed to use gradient descent to train a classifier. However, he has 1 million training examples. Rather than using the whole dataset to calculate the gradient, in each iteration, he decides to just randomly select 1 % of the training examples to significantly decrease the number of computations.

   (e) Rahul wants to use L2-regularization for a restaurant review sentiment analysis task. He wants to use these values of lambdas: [0.1, 1, 10, 100], these values of learning rates [0.4, 0.04, 0.004, 0.0004] and these number of iterations: [100, 1000]. He wants

to use Grid Search to narrow down the best set of hyperparameters, and he concludes that when using Grid Search, he will train have to train 10 models to find the best one.

2. From lecture, we have derived the loss function for logistic regression (also called cross-entropy loss) as:

$$\widehat{w} = \underset{w}{\text{argmin}}\, L(w) = \underset{w}{\text{axrgmin}}\; -\frac{1}{n}\sum_{i=1}^{n} \log(\sigma(f(x^{(i)})) + (1 - y^{(i)}) \log(1 - \sigma(f(x^{(i)})))$$

where :

- $(x^{(i)}, y^{(i)})$ represents a pair of feature and label of example i in a dataset of size n
- $y^{(i)} \in \{0, +1\}$.
- $f(x^{(i)}) = w^T x^{(i)}$ is a score function
- $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function

We can easily derive the gradient of this loss function with respect to the each weight w_j:

$$\frac{\partial L}{\partial w_j} = \frac{1}{n}\sum_{i=1}^{n} (\sigma(f(x^{(i)})) - y^{(i)}) x_j^{(i)}$$

For this problem, **you don't have to understand any of the above mathematical formulas or read them at all** (but they're worth exploring if you're curious). We will use them to implement logistic regression from scratch with the following Python code.

There are two errors in this code in the functions `gradient_descent()` and `predict()`. Identify both, briefly explain why they're not correct and propose an approach to fix them.

Note: None of the errors are related to coding. Assume each line of code doesn't have syntax errors and demonstrate general implementations well. However, the programmer has a slight conceptual misunderstanding of gradient descent and decision boundary, which involves some maths.

3. Based on our implementation of the two models from the programming assignment (majority class classifier and sentiment classifier with logistic regression), which model would you predict will perform better in the future? Briefly explain.

4. **Extra credit**

Let $\widehat{w}$ be the solution to an unregularized logistic regression problem, and let $\widehat{w}^*$ be the solution to the same problem with L2 regularization. Prove mathematically:

$$\left\|\widehat{w}^*\right\|_2^2 \leq \left\|\widehat{w}\right\|_2^2$$