

# Comp 642: Assignment #1 (80 points)

Rice University — Due Date: Thursday, 02/07/2023

**Submission Instructions:** For coding questions, please submit python notebook along with all the plots and 2-3 paragraphs explaining what you observe and what are your conclusions. Please do familiarize yourself with python notebooks (like Jupyter or Google Collab), they are very convenient, you can run them in your browser without any installations.

## 1 Logistic Regression by Hand

20 Points

You are given a data set of three samples with 2-dimensional feature vectors.  $D = \{5, 10, TargetValue = 1\}, \{40, -9, TargetValue = 0\}, \{10, 2, TargetValue = 2\}$ , here  $TargetValue$  is the label, which takes only three possible values (three classes)  $\{0, 1, 2\}$ .

We will train a logistic regressions model with parameter vectors  $x$ , with cross-entropy loss.

**Questions: Full Pseudo Codes whenever an algorithm is expected**

- How many parameters are there in logistic regressions
- Write down the loss as a function of parameters.
- Calculate the partial derivative of the loss function with respect to each parameter
- Write the iterative gradient descent update algorithm, assuming a step size  $\eta$
- Assuming that we calculate stochastic gradients, i.e., pick a random point and compute the gradient with respect to that point only. Write down the algorithm for the ADAM update rule.

## 2 (Coding) Python Notebook and Variants of Gradient Descent 40 Points

We provide a python notebook (*GradientAndGradientFreeOpt.ipynb*) with a sample generated dataset. We fit a regression model with mean square error (MSE) loss. The notebook demonstrates and compares two optimization methods. One is full gradient descent, and the other is *zeroth order descent*, which randomly chooses eight directions and picks the one that results in the best descent (or decrease in the loss functions). We have given a template for plotting different things and comparing the convergence behavior and the fit quality.

**Questions: We expect full working codes and plots. Please submit the notebook and all the plots. Write a short conclusion about what you observe**

- (SGD) Write a new method that gives stochastic gradient descent (SGD) (randomly pick one data sample and return the gradient only on that sample). Compare the convergence and accuracy of SGD with others.
- (Averaged SGD) Implement averaged SGD (See <http://dustintran.com/blog/on-asymptotic-convergence-of-averaged-sgd/>). That is, take the running average of the parameters during each iteration
- Assuming SGD, Implement ADAM and any two of your favorite gradient descent idea from <https://runder.io/optimizing-gradient-descent/>. Compare and contrast convergence accuracy, etc.

### 3 (Coding) Double Descent

20 Points

We provide a python notebook (DoubleDescent.ipynb) with 10k samples from the MNIST dataset. Here we are demonstrating double descent phenomena with ridge regression. We used an example of a feature generator that mixes random numbers.

**Questions: We expect full working codes and plots. Please submit the notebook and all the plots. Write a short conclusion about what you observe.**

- Instead of Ridge Regression, use any of your favorite classifiers for MNIST and see if you still get the double descent phenomena.
- Using <https://github.com/gwgundersen/random-fourier-features/blob/master/rffridge.py> as example, write your own feature generator (Random Fourier Features) to replace the *generate\_synthetic\_data* function. See if you can again demonstrate the double descent phenomena.