Assignment 1 Report

Gradient Descent for Linear & Logistic Regression

# Objective

The objective of the assignment was to implement and evaluate the gradient descent algorithm for linear and logistic regression. The experiments in this assignment involved analyzing the effects weight initialization, learning rate, as well as batch size on the convergence of the gradient descent method. This report presents the implementation, results, and observations for experimental objectives including:

1. The impact of weight initialization
2. The effect of learning rates on convergence
3. The influence of batch size on training dynamics

# Experimental Setup

## Linear Regression

* Data Generation:
  + Feature vectors (X) were sampled in uniform with 4 features
  + Target values (y) were generated using the following equation:
    - y = 5x1 – 2x2 + 3.5x3 – 1.5x4 + N(0,2)
    - Where N(0,2) represents Gaussian noise with a variance of 2
  + Generated 8,192 training samples
  + Generated 8,192 testing samples
* Loss Function: Mean Squared Error (MSE)
* Evaluation Metric: Test set MSE

## Logistic Regression

* Data Generation:
  + Samples for two classes were generated from Gaussian distributions:
    - Class 0: Mean = [1,1,1,1]
    - Class 1: Mean = [0.5,0.5,0.5,0.5]
    - Where covariance = identity matrix
  + Generated 4,096 training samples
  + Generated 4,096 testing samples
* Loss Function: Cross-Entropy Loss
* Evaluation Metric: Test set accuracy

## Gradient Descent Implementation

A combined gradient descent function was implemented for both regression tasks:

* Optimization Type: Mini-batch Stochastic Gradient Descent (SGD)
* Batch Size: Varied across [2,32,256,512,4096]
* Learning Rates: [0.05,0.01,0.1,0.2]
* Iterations: 100 epochs

# Experiments and Results

## 1st Experiment: Weight Initialization

* Setup:
  + Initialized weights to zeros
  + Initialized weights to uniform random values between [0,1]
  + Learning rate = 0.1
  + Batch size = 32
* Results:
  + Linear Regression
    - Zeros Initialization: Test MSE = 2.0137
    - A graph of a graph

      AI-generated content may be incorrect.Uniform Initialization: Test MSE = 2.0201
  + Logistic Regression:
    - Zeros Initialization: Test Accuracy = 56.53%
    - A graph of a logistic regression

      AI-generated content may be incorrect.Uniform Initialization: Test Accuracy = 56.80%
* Observations:
  + Linear Regression: Both zeros and uniform initialization resulted in similar performance, indicating the impact of initialization for this problem is minimal
  + Logistic Regression: The uniform initialization barely outperformed zeros initialization. This is likely due to better symmetrical breaking in the process of optimization

## 2nd Experiment: Effect of Learning Rate

* Setup:
  + Learning rates: [0.05,0.01,0.1,0.2]
  + Initialized weights to zero
  + Batch size: 32
* Results:
  + Linear Regression:
    - The graph below displays the impact of learning rate on loss:
      * Lower learning rates, such as 0.01, showed slower convergence
      * Higher learning rates, such as 0.2, caused oscillations
      * The optimal learning rate was 0.1 for this scenariA graph with red and green lines

        AI-generated content may be incorrect.o
  + Logistic Regression:
    - Similar trends from linear regression were observed:
      * High learning rates oscillated; smaller rates slowly converged
* A graph of a graph

  AI-generated content may be incorrect.Observations:
  + The learning rates impact convergency significantly. Higher rates accelerate training, but they risk instability. While lower rates are more stable, they displayed slow convergence

## 3rd Experiment: Effect of Batch Size

* Setup:
  + Batch sizes: [2,32,256,512,4096]
  + Learning rate = 0.1
  + Initialized weights to zeros
* Results:
  + Linear Regression:
    - Smaller batch sizes, such as 2, resulted in high variance, but faster updates
    - Larger batch sizes, such as 4096, seemed to smooth the loss curve, but also slowed convergence
    - A graph of a graph

      AI-generated content may be incorrect.Medium batch sizes, such as 32 or 256, achieved ideal balances
  + Logistic Regression:
    - A graph of a logistic regression

      AI-generated content may be incorrect.Similar trends as linear regression was shown, with smaller batch sizes showing higher variability, and larger batch sizes displaying smoother convergence
* Observations:
  + Smaller batch sizes bring noise in updates, which helps escape local minima, but also causes more instability
  + Larger batch sizes lead to more stable convergences, but also causes slower updates

# Graphs

* The attached plots throughout this report illustrate the loss vs. iterations for each experiment. Key takeaways of these graphs include:
  + Initialization of weights has a minimal impact for linear regression, while displaying slight improvement for logistic regression
  + Learning rates and batch size significantly impact convergence dynamics

# Conclusion

* The experiments conducted in this report highlight the importance of hyperparameter tuning within gradient descent optimization. Key findings of this report include:
  + Weight Initialization: shows a minimal impact on linear regression and a slight impact on logistic regression
  + Learning Rate: these experiments concludes that a learning rate of 0.1 provided the greatest trade-off between stability and speed
  + Batch Size: The best balance of stability and speed was observed within medium batch sizes.
* The implementation and results of this study align with the theoretical expectations presented in class lectures, and the experimental objectives were achieved successfully

# Resources

* The following resources were referenced in preparing this report:
  + ***Deep Learning Foundations and Concepts****, Christopher Bishop and Hugh Bishop. Available at:* [*https://www.bishopbook.com/*](https://www.bishopbook.com/)*.*
  + ***Deep Learning with Python****, François Chollet, Manning; 1st Edition, December 22, 2017. ISBN: 978-1617294433.*
  + ***Linear Neural Networks Lecture Notes****, M. R. Hassan.*
  + ***Probability Refresher****, course lecture materials.*