Student name: Sahil Sajan
Student ID: 202405476

Comparison of Batch and Stochastic Gradient Descent for Logistic Regression on College Placement Data

Introduction

This study looked at implementing and evaluating the performance of both Batch Gradient Descent (BGD) and Stochastic Gradient Descent (SGD), in the process of training a logistic regression model using the College Placement dataset. The study aims to look at the two methods from the point of view of how efficiently they learn and generalize to test data.

Dataset and pre-processing

This dataset contains student placement information, where placement of a student (1) or non-placement (0) is predicted on basis of many features.

We have removed the column "Student ID," which would not be useful in making the predictions.
We standardized the dataset features across the board using StandardScaler(), thereby ensuring a uniform scale.
Then we applied an 80-20 split, with 80% of the dataset being used for training and the remaining for testing to evaluate model performance.

Gradient Descent Approaches

Two gradient descent variations were used:

Batch Gradient Descent (BGD)

Updates model weights after the entire dataset is processed during an iteration.
Stable but slow on account of gradient computation over the entire dataset.

Stochastic Gradient Descent (SGD)

Updates model weights after each individual data point.
Fast but has some level of noise, where it also can't produce smooth convergence as BGD.

Results and Observations

We monitored the loss function (binary cross-entropy) over 10,000 iterations for both methods whereby performance was compared:

Batch Gradient Descent (BGD)

Convergence: Smooth and stable
Final Loss (Training Set): Lower than SGD, indicating better stability
Time Efficiency: Slowest, as it updates after seeing all data

Stochastic Gradient Descent (SGD)
Convergence: Fast but more like a saw-tooth pattern due to constant changes in update
Final Loss (Training Set): Larger than BGD but still within reasonable limits
Time Efficient: Fastest, update happens after every data instance.

Comparison on Test Data

Generalization of the model to unseen data was tested.

BGD performed better on the test data since its smooth convergence produced more stable predictions.
SGD's slight variations indicate that more learning rate tuning for reasonably stable updates is needed.

Conclusion

BGD is stable but slow with its suitability for cases where computational power is available.
SGD is fast but susceptible to high variance unless careful tuning is done.
In training on large datasets, SGD is to be chosen for its efficiency, while BGD performs better with smaller datasets, where smooth convergence is much needed.

Final Recommendation

Batch Gradient Descent is the technique of choice for this placement dataset, as it performs better in terms of accuracy and stability despite being slower. However, SGD could be enhanced using techniques such as mini-batch gradient descent and adaptive learning rates.