Analysis of Logistic Regression on MNIST Dataset

Lucas Teixeira Rocha - m11813 .
Computer Science and Engineering - 2nd Cycle Degree
Universidade da Beira Interior
Covilhã, Portugal
lucas.rocha@ubi.pt

***Abstract*—This study delineates a meticulous exploration into the performance of logistic regression, a foundational machine learning algorithm, on the MNIST dataset for multi-class classification. Through a comparative analysis between a custom logistic regression implementation, MyLogisticRegression, and scikit-learn's LogisticRegression, the study unveils the critical role of normalization strategies, hyperparameter tuning, and stopping criteria in enhancing model performance. The findings reveal that with appropriate hyperparameter settings and normalization, MyLogisticRegression can achieve competitive accuracy, demonstrating the potential of logistic regression in handling multi-class classification tasks. The study further underscores the advantages of leveraging well-optimized implementations provided by established machine learning libraries. The insights garnered align with the broader understanding of feature scaling in machine learning as highlighted by Andrew Ng [1]. This investigation not only contributes to a deeper understanding of logistic regression behavior on multi-class classification tasks but also lays a solid groundwork for future explorations into more complex models and optimization techniques.**

I. Introduction

The analysis and understanding of machine learning algorithms are crucial for advancing the field and enabling the development of more efficient and accurate models. Among the myriad of machine learning algorithms, logistic regression serves as a fundamental building block, often used for binary and multi-class classification tasks [2]. Despite its simplicity, logistic regression's performance can be significantly affected by various factors such as hyperparameters, data preprocessing techniques, and the choice of optimization algorithms.

The MNIST dataset, a collection of handwritten digits, is a widely recognized benchmark dataset for evaluating the performance of machine learning algorithms in multi-class classification tasks [3]. Analyzing logistic regression performance on this dataset not only provides insights into the behavior of the algorithm but also serves as a benchmark for comparing other more complex models like Convolutional Neural Networks (CNNs) [4].

A detailed investigation into the performance of logistic regression with a focus on hyperparameters, normalization strategies, and stopping criteria remains an area warranting thorough exploration. The comparison between custom implementations and widely tested library implementations such as scikit-learn's LogisticRegression provides a valuable perspective on the efficacy and efficiency of logistic regression in handling multi-class classification tasks.

This study aims to bridge this gap by conducting a comprehensive analysis of a custom logistic regression implementation dubbed MyLogisticRegression against scikit-learn's LogisticRegression on the MNIST dataset. The exploration of different hyperparameters, normalization strategies, and stopping criteria provides a rich understanding of their impact on model performance. The findings from this analysis are expected to contribute to the broader understanding of logistic regression behavior in multi-class classification tasks.

II. Methodology

A. Implementation

Despite the name, logistic regression is a supervised classification algorithm. It's used to estimate the probability that an occurrence belongs to a classification, e.g., spam folder of email, where a threshold probability of ¿50% predicts that the occurrence belongs to the positive class (denoted as 0, i.e. normal email) or predicts that the classification belongs to the negative class (denoted as -1, i.e. spam email, denoted as 1, i.e. spam). The logistic function is a sigmoid function whose output is restricted to a range of 0 to 1 [5].

Two logistic regression implementations are explored in this study:

- **MyLogisticRegression**: A custom implementation of logistic regression with multinomial classification capability. This implementation utilizes the softmax function for probability estimation, gradient descent for optimization, and L2 regularization to prevent overfitting.
- **ScikitLogisticRegression**: Logistic Regression implementation provided by the scikit-learn library [6], which is well-optimized and widely used in the machine learning community.

Both implementations are evaluated on their ability to accurately classify handwritten digits from the MNIST dataset.

B. Dataset

The MNIST dataset, a collection of 60,000 training and 10,000 testing examples, serves as the platform for this analysis. Each example consists of a grayscale image of a handwritten digit, with a dimensionality of 28×28 pixels, flattened into a 784-dimensional vector. The labels are provided as integer values ranging from 0 to 9, representing the respective digits [3]. In order to avoid memory and time constraints, a subset corresponding to 10% of the training data was selected, after the results shown in Tables I and II.

TABLE I
ACCURACY X TIME CONSUMPTION OF SCIKIT-LEARN LOGISTICREGRESSION MODEL WITH DIFFERENT SAMPLE SIZESS

Sample size	Accuracy	Duration	Time-Normalized Acc.
60k samples	89.92%	3min 32s	0.0042
12k samples	89.29%	55s	0.0160
6k samples	89.24%	28s	0.0314

TABLE II
ACCURACY X TIME CONSUMPTION OF MYLOGISTICREGRESSION MODEL WITH DIFFERENT SAMPLE SIZES

Sample size	Accuracy	Duration	Time-Normalized Acc.
60k samples	92.56%	5min 42s	0.0016
12k samples	90.83%	34s	0.0265
6k samples	89.68%	17s	0.0517

C. Data Preprocessing

Two normalization strategies are employed to evaluate their impact on model performance:

- **Min-max Normalization**: Scales the pixel values to a range of [0, 1].
- **Z-score Normalization**: Standardizes the pixel values to have a mean of 0 and a standard deviation of 1.

D. Hyperparameter Tuning

A range of hyperparameters are tuned to understand their influence on model performance:

- **Learning Rate**: Different learning rates (0.01, 0.1, and 1.0) are tested to assess their impact on the convergence and accuracy of MyLogisticRegression.
- **Regularization Strength**: Various levels of L2 regularization strength are explored to gauge their effect on model generalization.
- **Stopping Criteria**: Different stopping criteria are evaluated to understand their impact on training efficiency and model accuracy.

E. Evaluation Metrics

The models' performance is evaluated based on Accuracy:

- **Accuracy**: The proportion of correctly classified examples out of the total number of examples.

To identify the 10% subset of the training data as the most efficient segment for processing, the following metrics were applied:

- **Execution Time**: The total time taken to train the model and make predictions.
- **Time-Normalized Accuracy**: A derived metric calculated as the accuracy per unit of execution time.

F. Experimental Setup

The experiments were conducted on a standard Google Colab computing environment. All implementations and evaluations are carried out using Python programming language.

III. RESULTS

The performance of the logistic regression models, MyLogisticRegression and ScikitLogisticRegression, was extensively evaluated under various configurations. The results provide a comprehensive insight into the behavior of logistic regression on the MNIST dataset.

A. Normalization Strategies

Normalization plays a vital role in improving the overall performance of regression models in general. The results, as shown in Table III, reveal that z-score normalization enhances the performance of MyLogisticRegression, achieving an accuracy of 90.29%. In contrast, ScikitLogisticRegression performs optimally with Min-max normalization, reaching an accuracy of 89.68%.

TABLE III
ACCURACY OF LOGISTIC REGRESSION MODELS WITH DIFFERENT NORMALIZATION STRATEGIES

Normalization Strategy	MyLogisticRegression	ScikitLogisticRegression
Min-max Normalization	89.24%	89.68%
Z-score Normalization	90.29%	88.97%

B. Hyperparameter Tuning

The choice of hyperparameters significantly influences the performance and convergence of logistic regression models [1]. Tables IV and V showcases the accuracy of the models under different regularization strengths and learning rates. The results indicate that a learning rate of 0.1 within the default regularization weight yields optimal performance for MyLogisticRegression, whereas ScikitLogisticRegression benefits from strong regularization.

TABLE IV
ACCURACY OF LOGISTIC REGRESSION MODELS UNDER DIFFERENT REGULARIZATION WEIGHTS

Hyperparameter	MyLogisticRegression	ScikitLogisticRegression
Stronger Regularization	86.16%	89.45%
Strong Regularization	89.06%	90.67%
Default Regularization	90.29%	89.68%
Weak Regularization	90.22%	88.58%
Weaker Regularization	90.23%	87.99%

TABLE V
ACCURACY OF MYLOGISTICREGRESSION MODEL UNDER DIFFERENT LEARNING RATES

Learning Rate	MyLogisticRegression Accuracy
0.01	89.57%
0.1	90.30%
1	90.30%

C. Stopping Criteria

The stopping criteria control the training duration and indirectly influence the model's accuracy. The results, depicted in Table VI, demonstrate a slight sensitivity in MyLogisticRegression's performance to the stopping criteria, whereas

ScikitLogisticRegression maintains consistent accuracy across different stopping configurations.

TABLE VI
ACCURACY OF LOGISTIC REGRESSION MODELS UNDER DIFFERENT STOPPING CRITERIA

Stopping Criteria	MyLogisticRegression	ScikitLogisticRegression
Default	90.29%	90.67%
Stopping Sooner	90.31%	90.67%
Training Longer	90.29%	90.67%

These results underscore the importance of appropriate hyperparameter selection and data preprocessing in achieving optimal logistic regression performance on the MNIST dataset.

IV. Discussion

The experimental results present an insightful exploration into the behavior of logistic regression on the MNIST dataset. The comparative analysis between MyLogisticRegression and ScikitLogisticRegression sheds light on the intricacies involved in implementing and optimizing logistic regression for real-world tasks.

A. Normalization Strategies

Normalization strategies have shown to be critical in enhancing the performance of logistic regression models. The z-score normalization emerged as a favorable strategy for MyLogisticRegression. Although the scikit-learn implementation exhibited a preference for Min-max normalization, the underlying optimizations within scikit-learn's LogisticRegression could attribute to this behavior.

B. Hyperparameter Tuning

The significant impact of hyperparameters on the model performance is consistent with the literature. The choice of learning rate and regularization strength can greatly influence the convergence and generalization of logistic regression models. A learning rate of 0.1 appeared to strike a balance between convergence speed and accuracy for MyLogisticRegression, while ScikitLogisticRegression seemed to benefit from stronger regularization, possibly due to its well-optimized implementation.

C. Stopping Criteria

The sensitivity of MyLogisticRegression to stopping criteria, albeit slight, underscores the importance of adequate training duration for achieving optimal performance. On the other hand, the consistent performance of ScikitLogisticRegression across different stopping criteria could be attributed to its robust implementation and the underlying optimization algorithms employed in scikit-learn [6].

D. Implementation Differences

The discrepancies in performance between MyLogisticRegression and ScikitLogisticRegression could be ascribed to the well-optimized nature of scikit-learn's implementation. The library is designed to provide efficient tools for data mining and analysis, which likely contributes to the higher efficiency and slightly better accuracy observed in ScikitLogisticRegression.

E. Implications and Future Work

This study elucidates the importance of data preprocessing, hyperparameter tuning, and a well-optimized implementation in achieving competitive performance in logistic regression models. The insights garnered provide a foundation for further investigations into more complex models and optimization techniques. Additionally, the exploration of other advanced optimization algorithms and regularization techniques could further enhance the performance of logistic regression on multi-class classification tasks.

V. Conclusion

A deeper understanding of the impact of normalization strategies, hyperparameter tuning, and stopping criteria on model accuracy and efficiency was achieved. The critical role of normalization in enhancing model performance aligns with the broader understanding in the machine learning community regarding the importance of feature scaling [1]. The significant impact of hyperparameters on model performance reiterates the necessity for a well-considered hyperparameter selection process. Additionally, the sensitivity to stopping criteria underscores the importance of adequate training duration to achieve optimal model performance.

The observed performance discrepancy between MyLogisticRegression and ScikitLogisticRegression highlights the advantages of leveraging well-optimized implementations provided by established machine learning libraries. However, the competitive accuracy achieved by the custom implementation demonstrates the potential of logistic regression in handling multi-class classification tasks when appropriately configured.

This investigation not only enriches the understanding of logistic regression on multi-class classification tasks but also lays a solid foundation for future explorations into more complex models and optimization techniques. The insights garnered are instrumental for researchers and practitioners aiming to leverage logistic regression in various domains.

References

[1] A. Ng, "Machine learning," 2021. Coursera course.
[2] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics, 2001.
[3] Y. LeCun and C. Cortes, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
[5] www.turing.com, "Effects of normalization techniques on logistic regression in data science," visited on October 31, 2023.
[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.