|  |  |
| --- | --- |
| **Ex No: 1**  **Date: 08/08/2024** | **Python and NumPy basics** |

**Objective:**

To implement foundational functions for logistic regression, including the sigmoid function, vectorized operations, and performance optimization using Python and Numpy, as well as to measure computation time for improved efficiency in model training.

**Descriptions:**

In this lab, we explore key components and operations critical to logistic regression, a fundamental algorithm used for binary classification tasks. Logistic regression assigns probabilities to class labels based on input features and is widely used in applications such as image classification, fraud detection, and medical diagnosis.

Key concepts covered in this lab include:

* **Sigmoid Function**: The sigmoid function, often represented as  is used to map real-valued inputs to a range between 0 and 1, making it ideal for probability estimation in binary classification problems.
* **Vectorized Operations**: Efficient computation with numpy vectorized operations enables faster processing of large datasets by applying operations simultaneously across array elements, improving the model’s computational performance.
* **Computation Time Measurement**: Tracking execution time for operations helps optimize the logistic regression model by minimizing unnecessary computation overhead, a crucial aspect for scaling machine learning algorithms.

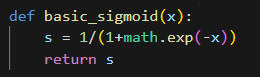
**Model:**

The logistic regression model in this lab is based on the following key steps:

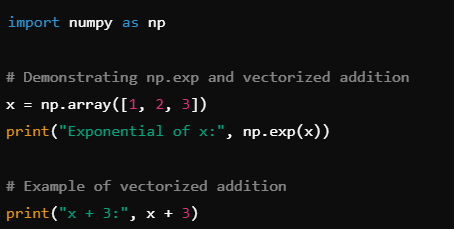
1. **Sigmoid Function**: The sigmoid function  is applied to map predictions to probabilities. This helps in determining the class likelihood for binary classification (1 for the positive class, 0 for the negative class).
2. **Vectorization for Efficiency**: Operations such as addition, subtraction, and dot products are vectorized using numpy. This allows the model to handle data in bulk, reducing computational load and increasing speed.
3. **Performance Optimization with Timing**: Using Python's time module, we measure the execution time for matrix and vector operations. Efficient use of vectorization and optimization techniques can significantly reduce the model's training time, which is essential for practical applications of logistic regression.

**Code Implementation**

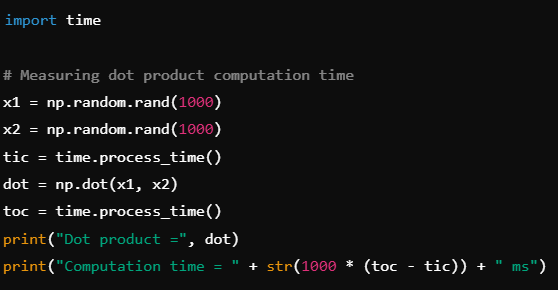
* Sigmoid Function Implementation:



* Vectorized Operations:



* Dot Product and Computation Time



**Results and Observations:**

Through this lab, we observe:

1. The **sigmoid function** successfully converts values to a probability range between 0 and 1, making it suitable for binary classification outputs.
2. **Vectorized operations** such as addition and dot products greatly improve the performance of basic operations, which is crucial in handling large datasets efficiently.
3. **Computation time measurements** show that vectorized numpy operations can process large arrays in milliseconds, underlining the importance of vectorization for scaling machine learning algorithms.

**Code:**