Labwork 2: Linear Regression

Le Chi Thanh - M23.ICT.011

04/21/24

1 Introduction

Linear regression is a fundamental supervised learning method used for modeling the relationship between a dependent variable and one or more independent variables. In linear regression, the goal is to find the best-fitting linear equation that describes the relationship between the input variables and the target variable.
Gradient descent, on the other hand, is an optimization algorithm commonly used to minimize a given objective function. In the context of linear regression, gradient descent is utilized to iteratively adjust the model parameters in order to minimize the difference between the predicted and actual values of the target variable. By continuously updating the parameters in the direction of steepest descent of the loss function, gradient descent enables the model to converge towards an optimal solution.
In this report, we present an analysis of the implementation of linear regression using gradient descent. We discuss the algorithmic details of gradient descent, including the parameter initialization, iterative updates, and convergence criteria. Furthermore, we investigate the effect of different learning rates on the convergence behavior of the optimization process. By experimenting with various learning rates and observing their impact on convergence speed and stability, we aim to identify an optimal learning rate that facilitates efficient model optimization.

2 Implementation of the Algorithm:

The implementation of linear regression using gradient descent involves several key components. First, the dataset containing the input-output pairs is loaded and preprocessed. Then, an initial set of parameters is chosen, and the gradient descent algorithm is applied iteratively to update these parameters until a convergence criterion (threshold) is met. During each iteration, the loss function, representing the discrepancy between the predicted and actual values, is computed along with its partial derivatives with respect to the model parameters. These derivatives guide the parameter updates in the direction that minimizes the loss, ultimately leading to the identification of the optimal parameter values.

We break down the process into several key components, including dataset loading, parameter initialization, gradient descent iterations, and convergence criteria.

2.1. Dataset Loading

The first step in implementing linear regression is to load the dataset containing the input-output pairs. In our implementaion, we open the csv dataset file and iterate over the data line in the file to save each input-ouput pair as a tuple to the 'DATASET' list.

2.2. Gradient Descent Iterations

The core of the algorithm lies in the iterative application of gradient descent to update the model parameters and minimize the loss function. We define the gradientDescend2d function, which takes initial parameter values (param0 and param1), learning rate (L), convergence threshold (t), the objective function (f), and its partial derivatives (d_w0 and d_w1) as input arguments. Within a while loop,we compute the current value of the objective function (oldF) using the provided parameters. We then update the parameters w0 and w1 using the gradient descent update rules with a specific learning rate. After the parameter updates, we compute the new value of the objective function (newF) and calculate the absolute difference between the new and old function values (diff).

If the difference falls below the convergence threshold (t), indicating that the optimization has converged, we stop the process and return the optimized parameter values.

2.3. Loss Functions and Derivatives

We define the loss function (loss) as a lambda function that takes the dataset and model parameters (w0 and w1) as input and computes the total loss over all input-output pairs in the dataset. Similarly, we define lambda functions for the partial derivatives of the loss function with respect to w0 (loss_w0) and w1 (loss_w1). These functions are used within the gradientDescend2d function to compute the gradient of the loss function and guide the parameter updates.

3 Analysis of Learning Rate Effects on Convergence

The learning rate (L) is a critical hyperparameter in the gradient descent optimization algorithm. It determines the size of the steps taken during each iteration of parameter updates. In this section, we investigate the impact of different learning rates on the convergence behavior of the linear regression model trained using gradient descent.

3.1. Experiment Setup

We conduct experiments with a range of learning rates to observe their effects on convergence. Learning rates are chosen from a predefined set (e.g., 0.01, 0.1, 0.5, 1.0). For each learning rate, we run the linear regression algorithm using gradient descent and record convergence-related metrics.

3.2. Result Observation

We observe the behavior of the loss function over iterations for each learning rate, smaller learning rate may result in slower convergence, with a gradual decrease in the loss function over a larger number of iterations. Conversely, a larger learning rate may lead to faster initial convergence but may oscillate or diverge later in the optimization process. The optimal learning rate strikes a balance between convergence speed and stability, resulting in a smooth decrease in the loss function.

We also analyze the number of iterations required for convergence at different learning rates. A very small learning rate may necessitate a large number of iterations for convergence, indicating slow optimization. On the other hand, a very large learning rate may cause premature convergence or divergence, requiring fewer iterations but compromising accuracy. The optimal learning rate achieves convergence within a reasonable number of iterations while maintaining accuracy and stability.

3.3. Discussion

The choice of learning rate directly impacts the performance (speed and number of iterations) of the trained linear regression model. A well-tuned learning rate leads to faster convergence and better model accuracy.

4 Conclusion

In conclusion, this report has provided a comprehensive overview of the implementation of linear regression using gradient descent optimization. Key points discussed include dataset loading, gradient descent iterations, the analysis of learning rate effects on convergence, and potential improvements to the algorithm.

Key Points Summary:

- Linear regression, coupled with gradient descent optimization, offers a powerful approach for modeling relationships between variables and making predictions.
- The learning rate (L) plays a crucial role in the convergence behavior of gradient descent optimization. Selecting an appropriate learning rate is essential for achieving efficient convergence and model accuracy.
- Through experiments, we observed that different learning rates can significantly impact convergence speed, stability, and the overall performance of the optimization algorithm.
- Careful consideration of convergence metrics, such as the loss function trend and the number of iterations, helps in identifying the optimal learning rate that balances convergence speed and stability.
- Inadequate choices of initial parameter values and learning rates can lead to slow convergence, instability, or suboptimal model performance.

Importance of Selecting an Appropriate Learning Rate:
Selecting an appropriate learning rate is critical for achieving efficient convergence in gradient descent optimization. A well-tuned learning rate facilitates faster convergence, improves model accuracy, and ensures stability during the optimization process.

Potential Extensions or Improvements to the Algorithm:

- Advanced optimization techniques: Explore alternative optimization algorithms, such as stochastic gradient descent, Adam, or RMSprop, which adaptively adjust the learning rate during training.
- Hyperparameter tuning: Implement automated hyperparameter tuning techniques, such as grid search or random search, to systematically search for optimal parameter values, including the learning rate.

- Regularization methods: Incorporate regularization techniques, such as L1 or L2 regularization, to prevent overfitting and improve the generalization performance of the model.

Challenges in Linear Regression:

- Difficulty in choosing initial parameter values: The choice of initial parameter values can significantly affect convergence behavior and model performance. Optimal strategies or techniques, such as random initialization or heuristics, are needed to address this challenge effectively.

- Optimal learning rate selection: Determining the optimal learning rate requires careful experimentation and analysis. Inadequate learning rates can lead to slow convergence or instability, consuming valuable computational resources.

In conclusion, the successful implementation of linear regression using gradient descent optimization requires careful parameter tuning and experimentation. By addressing challenges such as selecting appropriate initial parameter values and learning rates, we can improve the efficiency and effectiveness of the optimization process, ultimately leading to better-performing models in terms of speed and accuracy.