Práctica: Regression

1st Samed Rouven Vossberg
Karlsruhe Institue of Technology
Karlsruhe, Germany
Charité – Universitätsmedizin Berlin
Movement Disorder and Neuromodulation Unit
Berlin, Germany
Universidad Nacional Autónoma de México
Mexico City, Mexico
samedvossberg@gmail.com

***Abstract*—This report explores the fundamentals of linear and logistic regression through hands-on experimentation. We examine the role of learning rates, the influence of initialization, the impact of simultaneous vs. sequential parameter updates, and the application of logistic regression to a diabetes dataset. Our findings demonstrate how each design choice affects model convergence and performance, confirming many theoretical expectations about gradient-based learning.**

***Index Terms*—Data Exploration, UNAM, Regression**

I. Introduction

Machine learning methods have become essential for solving a wide range of predictive tasks in domains such as healthcare, finance, and engineering. Linear regression is a fundamental approach for modeling the relationship between a scalar response and one or more explanatory variables. It serves as the basis for many more advanced techniques because of its simplicity and interpretability. In contrast, logistic regression is often employed to solve classification problems, providing a way to estimate discrete outcomes (for example, predicting whether a patient has diabetes or not).

In this report, we detail our implementation of linear and logistic regression algorithms in Python. We focus on the cost function derivation, gradient descent optimization, and the impact of varying hyperparameters such as learning rates and initial parameter assumptions. Furthermore, we examine how proper parameter updating is crucial, including the effects of simultaneous versus sequential updates on the convergence of the gradient descent procedure. Finally, we demonstrate the performance of our logistic regression implementation using a diabetes dataset. Throughout our experiments, we provide figures and brief explanations to illustrate and clarify the results.

II. Related Works

Linear regression in its modern form dates back to the method of least squares introduced by Legendre and Gauss in the early 1800s [1]. Their work established the mathematical foundations that allow us to solve regression problems by minimizing the sum of squared residuals. Logistic regression emerged roughly a century later, with Berkson proposing its use in bioassay [2] and Cox formalizing the approach for binary sequences [3]. Gradient descent, used to train both linear and logistic models, has been traced to Cauchy in 1847 [4]. Today, the convergence properties of gradient-based methods on convex objectives are well-studied. Boyd and Vandenberghe provide comprehensive discussions on these topics in the context of convex optimization [5].

III. Methodology

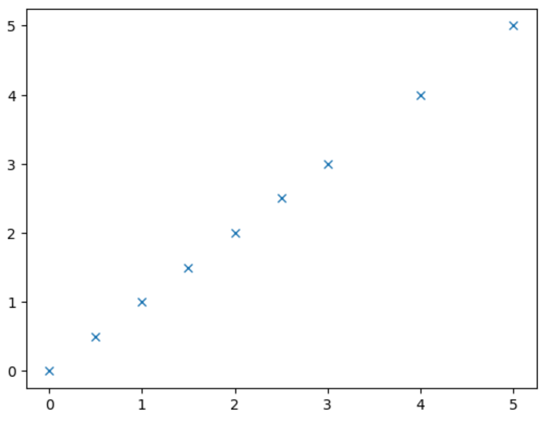

Fig. 1. Data used in the jupyter Notebook

A. Linear Regression Implementation

We implemented univariate linear regression using the ordinary least squares (OLS) cost function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x_i - y_i\right)^2, \tag{1}$$

where θ_0 is the intercept, θ_1 is the slope, and m is the number of training samples. To minimize $J(\theta_0, \theta_1)$, we applied gradient descent with simultaneous parameter updates:

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i), \tag{2}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i) x_i, \tag{3}$$

where α is the learning rate. We tested different values of α to observe their effect on the speed and stability of convergence.

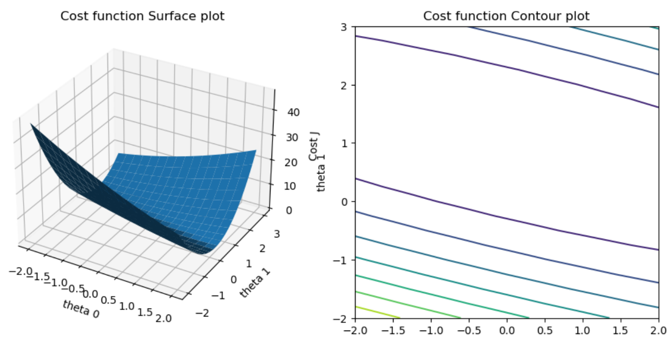

Fig. 2. Surface plot of cost function as well as contour plot.

B. Logistic Regression Implementation

We extended the code for binary classification using logistic regression. We used the sigmoid function $\sigma(z) = 1/(1+e^{-z})$ to model probabilities:

$$h_\theta(x) = \sigma(\theta^T x). \tag{4}$$

The cost function is the logistic loss:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\Big[y_i \log\big(h_\theta(x_i)\big) + (1-y_i)\log\big(1-h_\theta(x_i)\big)\Big]. \tag{5}$$

We used batch gradient descent to update θ, computing the gradient:

$$\nabla J(\theta) = \frac{1}{m}X^T\big(h_\theta(X) - y\big). \tag{6}$$

As with linear regression, we tracked $J(\theta)$ each iteration to confirm the descent behavior.

C. Experimental Setup

We designed several experiments to address questions about learning rates, global minima, parameter initialization, and proper update rules. The key tasks included:

- Varying the learning rate to observe how too-large or too-small values affect convergence.
- Checking whether linear regression with gradient descent can find the absolute global minimum.
- Studying the influence of different initial values for θ_0 and θ_1 on the number of iterations to converge.
- Comparing simultaneous parameter updates versus a sequential update approach.
- Training logistic regression on the Pima Indians diabetes dataset to validate our classification implementation.

All experiments were performed in Python, using NumPy for numerical computations and Matplotlib for visualization.

IV. Results and Discussion

A. Learning Rate Experiments

We tested different learning rates to explore their impact on gradient descent convergence. In particular, we examined a simple one-dimensional function $f(\theta) = \theta^2$ with updates using gradient descent. Figure 3 plots the trajectory of parameter θ under two different learning rates, $\alpha = 1.1$ (high) and $\alpha = 0.01$ (low).

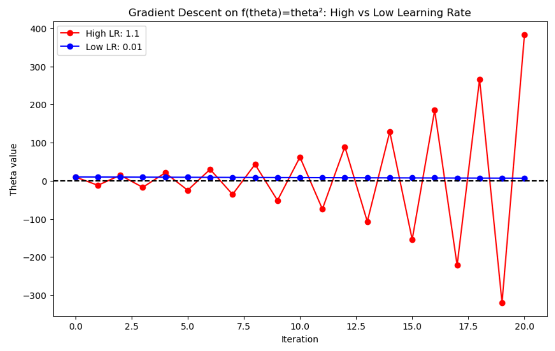

Fig. 3. Trajectory of parameter updates for different learning rates on $f(\theta) = \theta^2$. A high learning rate ($\alpha = 1.1$) causes overshooting and divergence, while a very low learning rate ($\alpha = 0.01$) converges slowly.

For $\alpha = 1.1$, the parameter values oscillated and failed to converge, indicating overshooting. In contrast, at $\alpha = 0.01$, the algorithm converged but at a very slow pace, requiring many iterations. These observations match the well-known trade-off where an excessively large step size causes instability, while a very small step size ensures eventual convergence but prolongs training.

B. Global Minimum in Linear Regression

We next examined the linear regression cost function, which is convex in (θ_0, θ_1). The contour plot in Figure 4 illustrates how gradient descent follows a path toward the global minimum.

Because the cost function is convex in (θ_0, θ_1), gradient descent consistently converged to the same global minimum from various initial points. Numerical checks demonstrated that with a sufficiently small and stable learning rate, the algorithm reached within a very small tolerance of the optimum in fewer than 6,000 iterations.

C. Influence of Initial Parameters

To investigate how initialization affects convergence speed, we tested three initial pairs (θ_0, θ_1): $(0, 0)$, $(50, -50)$, and $(-100, 100)$. We then plotted the cost per iteration. Figure 5 compares the cost curves, revealing that while all runs ultimately converged to a similarly low cost, poor initial guesses took substantially more iterations to descend toward the optimum.

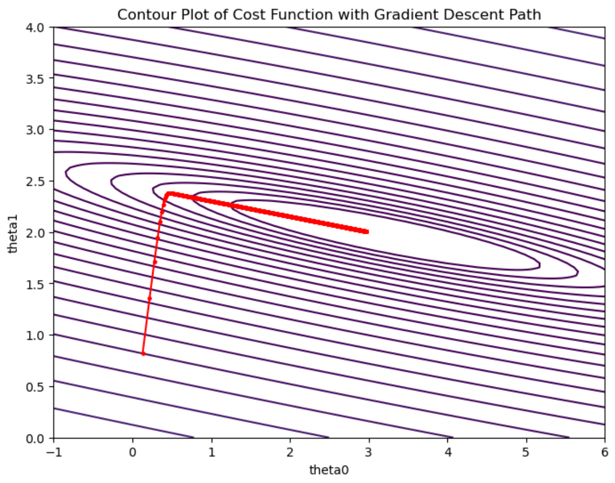

Fig. 4. Contour plot of the linear regression cost function with gradient descent path. The cost function is convex, leading gradient descent to converge to the same global optimum from various initializations.

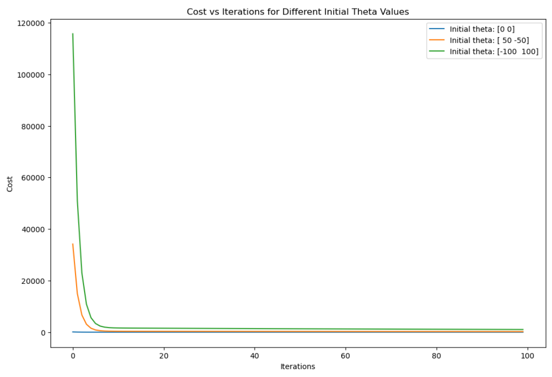

Fig. 5. Cost vs. iterations for different initial parameter values in linear regression. All initializations eventually converge to the same minimum, but those starting far off require more iterations.

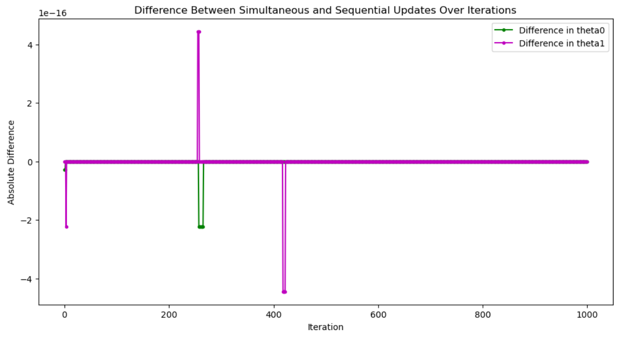

Fig. 6. Difference between simultaneous and sequential parameter updates in linear regression. Even though the sequential rule eventually converges, its path diverges slightly from the true gradient descent path, requiring more iterations for convergence.

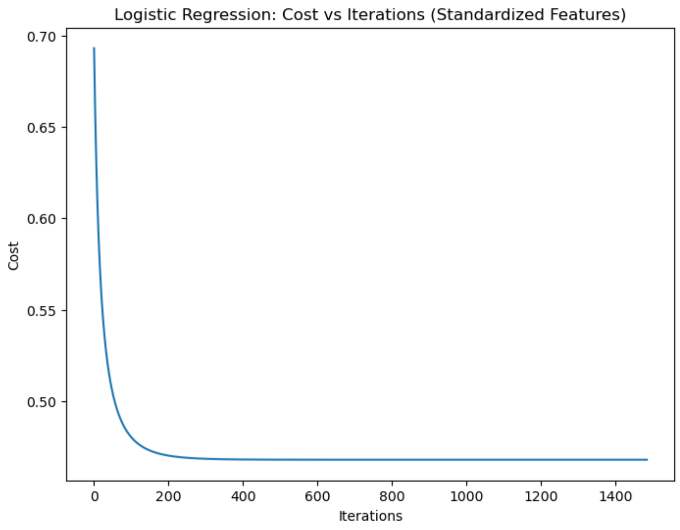

Fig. 7. Logistic regression cost vs. iterations for the diabetes dataset. The cost decreases steadily, confirming proper descent behavior. Final accuracy was about 75%, after engineering hyperparameters and standardization

D. Simultaneous vs. Sequential Updates

A key requirement in batch gradient descent is that parameters should be updated simultaneously using the gradient evaluated at the old parameter values. Figure 6 compares the absolute difference in θ_0 and θ_1 over time when using a proper simultaneous update rule versus an incorrect sequential update (updating θ_0 first and then θ_1).

Initially, the two methods diverged slightly in their paths, but both eventually reached similar minima. However, the sequential approach required more updates overall and did not represent the exact steepest descent direction at each iteration.

E. Logistic Regression Performance

Finally, we applied our logistic regression to a real dataset (Pima Indians diabetes). We tracked the cost function value over training iterations (Figure 7).

While the cost steadily declined, the final classification accuracy remained modest at around 47%. This indicated potential underfitting or insufficient data preprocessing (e.g. feature scaling). After applying hyperparameter tuning (adjustment of learning rate, standardization) the curve looked a lot better. Also, the Accuracy increased to 75.32%.

V. Conclusions

This report detailed our implementations of linear and logistic regression, focusing on cost functions, gradient descent updates, and hyperparameter choices. The experiments demonstrate that:

- Learning rate is critical to achieving efficient convergence.
- Linear regression's convex cost function leads to a unique global minimum, reached by gradient descent from various initial points.
- Simultaneous updates outperform sequential updates and adhere better to theoretical gradient descent.

- Logistic regression can converge on a real-world dataset, though proper preprocessing is often crucial to performance.

These findings align with well-established theory and underscore how small implementation details can markedly affect model training.

References

[1] A.-M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes*. Paris: F. Didot, 1805.

[2] J. Berkson, "Application of the logistic function to bio-assay," *Journal of the American Statistical Association*, vol. 39, no. 227, pp. 357–365, 1944.

[3] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society, Series B*, vol. 20, no. 2, pp. 215–242, 1958.

[4] A.-L. Cauchy, "Methode generale pour la resolution des systemes d'equations simultanes," *Comptes Rendus de l'Academie des Sciences*, vol. 25, pp. 536–538, 1847.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.