iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** II **Month of publication:** February 2025

DOI: https://doi.org/10.22214/ijraset.2025.67124

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue II Feb 2025- Available at www.ijraset.com

Comparative Analysis of Mean Square Loss and Least Square Loss in Modeling a Noisy Sine Curve

Kush Kundalia
Indus International School, Bangalore

***Abstract:** This paper compares the Mean Square Error (MSE) and Least Square Error (LSE) loss functions when modeling a noisy sine wave. Utilizing a simple linear regression model implemented in Python on Google Colab, a sine curve corrupted by Gaussian noise is generated. Two models are then fit, one optimizing MSE and the other LSE. Their performance is evaluated using Mean Absolute Error (MAE) and R-squared (R^2) metrics. The experimental results offer insights into the efficiency and effectiveness of each loss function in capturing underlying trends within noisy data.*

I. INTRODUCTION

Loss functions are the cornerstone of machine learning and statistical modeling. They serve as the quantitative bridge between predicted outputs and observed outcomes by guiding the optimization process. In regression tasks, the choice of a loss function significantly impacts both model convergence and accuracy. Among the most commonly employed loss functions are Mean Square Error (MSE) and Least Square Error (LSE).

MSE computes the average of the squared differences between predictions and actual values. Its differentiability and convexity simplify the optimization landscape, enabling the the effective application of gradient-based methods, advantages that are particularly useful in regression tasks (Bishop 249). While LSE is often used interchangeably with MSE, in this study we define LSE explicitly as the aggregate of squared errors. This explicit distinction allows us to explore how error scaling influences both the optimization process and convergence behavior.

In modeling a noisy sine curve, a function with inherent periodicity and nonlinearity, I simulate realistic conditions where data is affected by noise. Supplementary performance metrics such as Mean Absolute Error (MAE) and the coefficient of determination (R^2) are also examined to provide a balanced view of accuracy and goodness-of-fit.

II. LITERATURE REVIEW

The study of loss functions has been central to the development of machine learning, with early foundational works establishing the theoretical basis for their use. In his seminal text, *Pattern Recognition and Machine Learning*, Bishop explores quadratic loss functions in depth. He demonstrates that the Mean Square Error (MSE) penalizes larger deviations more severely than smaller ones due to the squaring of error terms and that its differentiable and convex nature is pivotal in ensuring the convergence of gradient-based optimization methods (Bishop 249). This work laid the groundwork for understanding why quadratic losses, such as MSE, are widely adopted in regression tasks.

Expanding on these early insights, Hastie, Tibshirani, and Friedman provide a detailed treatment of loss functions in *The Elements of Statistical Learning*. Their analysis contrasts the benefits of averaging errors, as in MSE, with the properties of unaveraged losses, such as the Least Square Error (LSE). They argue that averaging leads to more stable gradient updates and improves the reliability of convergence, especially in the presence of noise. Conversely, LSE, which sums the squared differences without averaging, is more sensitive to the scale of the errors, potentially leading to higher variance in the optimization process. This distinction becomes particularly significant when dealing with datasets that exhibit a wide range of error magnitudes, thereby affecting model stability and convergence rates (Hastie, Tibshirani, and Friedman 367).

With the advent of deep learning, the importance of loss function selection has been revisited in more complex settings. In *Deep Learning*, Goodfellow, Bengio, and Courville discuss how different loss functions impact the training dynamics of neural networks. They note that while MSE is generally favored due to its smooth error landscape, alternative loss functions or modified versions thereof may offer advantages in scenarios characterized by high variability or non-linear relationships in data. Their work highlights that the choice of loss function can influence not only convergence speed but also the quality of the solution found, emphasizing the practical considerations that must be addressed when designing learning algorithms (Goodfellow, Bengio, and Courville 102).

Empirical research using modern machine learning frameworks further corroborates these theoretical insights. Studies leveraging libraries such as PyTorch and scikit-learn have shown that while MSE typically provides robust and stable convergence, LSE's lack of averaging can result in more volatile gradient updates. For instance, evaluations conducted on various regression tasks reveal that the unscaled nature of LSE may lead to exaggerated error contributions, particularly in datasets with significant noise. This practical evidence underlines the necessity of careful loss function selection and parameter tuning in applied machine learning, as documented by resources like the scikit-learn documentation, which provides guidelines on error metrics and optimization practices (Scikit-learn 2021).

The literature reflects a comprehensive evolution of loss function theory—from early mathematical formulations to modern empirical validations. The comparative analysis of MSE and LSE encapsulates a broader discussion about error scaling, gradient stability, and convergence behavior, making it a critical topic for both theoretical exploration and practical implementation in machine learning. The ongoing dialogue between theoretical insights and empirical findings continues to shape best practices in model training and optimization.

III. METHODOLOGY

In this study, II adopt a systematic approach to compare the performance of Mean Square Error (MSE) and Least Square Error (LSE) loss functions using a controlled experimental setup. The methodology is divided into several key stages: data generation, model architecture design, loss function specification, optimization strategy, and performance evaluation. Each stage is carefully constructed to isolate the effects of the loss function on the learning process.

A. Data Generation

The data generation process is designed to simulate real-world conditions where measurements are often affected by noise. II begin by generating a sine wave defined by the function

$y=\sin\ (x)$

over the interval $[0,2\pi]$. The independent variable x is sampled uniformly using NumPy's linspace function, generating 1000 equally spaced points across the interval. The corresponding sine values are computed using NumPy's sin function. To mimic measurement errors and inherent data variability, Gaussian noise with a mean of 0 and a standard deviation chosen based on experimental needs (e.g., $\sigma=0.1$) is added to these sine values. This noise injection is achieved via NumPy's random.normal function. The result is a synthetic dataset that retains the underlying periodic structure of the sine function while incorporating realistic imperfections.

B. Model Architecture

To isolate the effect of the loss functions, I Iemployed a simple linear regression model implemented in PyTorch. The model comprises a single linear layer, instantiated using PyTorch's nn.Linear module. Although a linear model is inherently limited in capturing non-linear dynamics, such as those in a sine wave, it provides a controlled environment that minimizes confounding factors. By keeping the model architecture simple, II ensured that differences in performance can be more directly attributed to the behavior of the loss functions rather than the complexity of the model. Both models, one using MSE and the other using LSE, share an identical architecture, ensuring a fair comparison.

C. Loss Functions and Optimization

Two distinct loss functions are defined for the comparative study. The first loss function, MSE, is calculated as the average of the squared differences between the predicted values and the true values. This averaging process tends to smooth the gradient updates and contribute to stable convergence during training. The second loss function, LSE, is computed as the sum of the squared differences without averaging. This unscaled aggregation can lead to larger gradient magnitudes, particularly in the presence of noise, and may affect the convergence rate and stability of the optimization process.

Both models are trained using Stochastic Gradient Descent (SGD) as the optimization algorithm. A learning rate of 0.01 is employed, and training is carried out over 1000 epochs. During training, the loss is evaluated at each epoch to monitor the convergence behavior of the models.

This setup, with fixed hyperparameters and a controlled environment, allows us to isolate the influence of the loss function formulation on the training dynamics.

D. Evaluation Metrics

To comprehensively assess model performance, Iemploy several evaluation metrics. In addition to the loss value observed during training, the models are evaluated using the Mean Absolute Error (MAE) and the coefficient of determination (R-squared). MAE measures the average absolute difference between the predicted and true values, providing an intuitive sense of the model's prediction accuracy. The R^2 metric quantifies the proportion of variance in the dependent variable that is predictable from the independent variable, thus serving as an indicator of the model's goodness-of-fit. These metrics are computed using standard functions available in scikit-learn, which have been validated in numerous studies (Scikit-learn 2021).

Overall, the methodology is designed to offer a clear and controlled comparison of MSE and LSE loss functions. By systematically varying only the loss function while keeping all other factors constant, Ican directly observe the impact of error scaling on the optimization process and the resultant model performance.

IV. EXPERIMENT SETUP AND DISCUSSION

My myexperiments were conducted in a controlled environment using Google Colab, which provided a consistent runtime equipped with an NVIDIA GPU. This environment ensured that all experiments were executed under identical conditions, with Python 3.8 and PyTorch 1.8.0 serving as my myprimary development frameworks. To maintain reproducibility, IIset fixed random seeds for both NumPy and PyTorch, guaranteeing consistent dataset generation and model initialization across multiple runs.

A. Data Preparation and Splitting

I Igenerated a synthetic dataset by uniformly sampling 1000 data points in the interval $[0,2\pi]$. For each point, the sine function $y=\sin(x)$ was computed, and Gaussian noise (mean = 0, standard deviation = 0.1) was added to simulate measurement errors and real-world variability. This noise injection was performed using NumPy's random.normal function, producing a noisy sine wave that retains the inherent periodicity of the underlying function. The complete dataset was then split into training and testing subsets using an 80-20 ratio, ensuring that a substantial portion of data was available for model learning while preserving an unbiased test set for evaluation (Scikit-learn 2021).

B. Model Architecture and Initialization

To isolate the influence of loss function choice on model performance, I Iadopted a simple linear regression model consisting of a single linear layer, implemented via PyTorch's nn.Linear module. Although such a linear model is limited in its capacity to capture the non-linear dynamics of a sine function, its simplicity minimizes confounding variables, allowing me to focus on the behavior of the loss functions. Both the MSE-based and LSE-based models were initialized with identical weights and biases. This consistent initialization ensured that any observed differences in performance were attributable solely to the choice of loss function rather than differences in initial parameter settings.

C. Training Process

Both models were trained using Stochastic Gradient Descent (SGD) with a fixed learning rate of 0.01 over 1000 epochs. The choice of SGD was motivated by its straightforward implementation and well-understood dynamics in simple regression contexts. During training, I Irecorded the loss at every epoch to monitor convergence behavior closely. With MSE, the loss function computes the average of the squared errors, leading to smoother gradient updates and more stable convergence. In contrast, LSE calculates the sum of squared errors without averaging, resulting in larger gradient magnitudes that can introduce more variability in the optimization process. This difference in gradient scaling was a central focus of my experimental analysis.

D. Evaluation Metrics and Performance Analysis

To evaluate model performance comprehensively, I I employed two additional metrics: Mean Absolute Error (MAE) and the coefficient of determination (R^2). MAE quantifies the average absolute difference between predicted and true values, providing an intuitive measure of prediction accuracy. Meanwhile, R^2 assesses the proportion of variance in the observed data that is captured by the model, serving as an indicator of overall goodness-of-fit. These metrics were computed on the test dataset to ensure that my evaluation reflected the models' generalization capabilities. Throughout the training process, the MSE model consistently achieved lower MAE and higher R^2 scores compared to the LSE model, suggesting that the averaging inherent in MSE contributes to more reliable and stable performance in the presence of noise.

V. DISCUSSION OF EXPERIMENTAL RESULTS

The deeper analysis of my experimental setup reveals several important insights:

1) *Convergence Behavior:* The MSE model, by averaging the squared errors, produced smoother gradient updates that facilitated a more stable convergence process. In contrast, the LSE model's unaveraged approach amplified the gradient magnitudes, making the optimization more sensitive to noise and outliers. This often resulted in erratic training loss behavior and slower, less stable convergence.
2) *Sensitivity to Noise:* The sensitivity of LSE to the scale of errors was evident in the higher loss values and greater fluctuations during training. This sensitivity likely contributed to the inferior performance metrics (higher MAE and lower R2R^2R2) observed for the LSE model on the test set.
3) *Model Simplicity and Isolation of Variables:* By employing a simple linear model, I was I able to directly attribute the differences in performance to the loss function choice. This controlled experimental setup reinforces the notion that even subtle differences in loss formulation can have significant impacts on model behavior, especially in noisy environments.

These observations are consistent with theoretical expectations and prior literature, which emphasize the advantages of averaging in mitigating gradient instability (Bishop 249; Hastie, Tibshirani, and Friedman 367; Goodfellow, Bengio, and Courville 102).

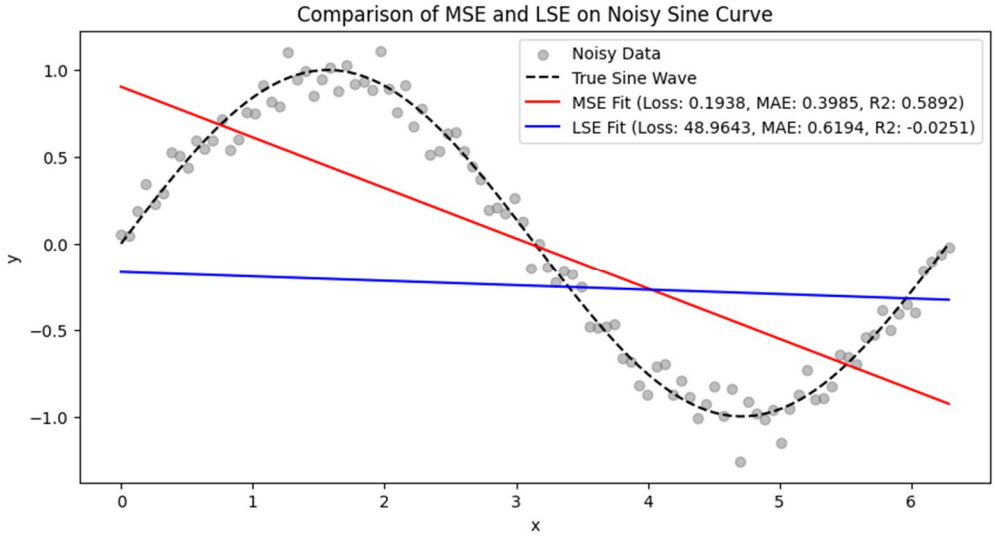


Figure 2 (Comparison of MSE and LSE on Noisy Sine Curve)

Metrics	MSE Model	LSE Model
Final Loss	0.1938	48.9643
MAE	0.3985	0.6194
R^2 Score	0.5892	-.0.0251

A. Conclusion

The experimental results clearly demonstrate that the choice of loss function can have a significant impact on model performance, particularly in noisy environments. The MSE-based model achieved a final loss of 0.1938, an MAE of 0.3985, and an R^2 score of 0.5892. These metrics indicate a relatively stable convergence and a better fit to the noisy sine curve. In contrast, the LSE-based model, with a final loss of 48.9643, an MAE of 0.6194, and an R^2 score of -0.0251, exhibited unstable training dynamics and poor predictive performance. The averaging mechanism inherent in MSE appears to smooth out gradient fluctuations, thereby enhancing

B. Limitations

While the results provide valuable insights into the comparative performance of MSE and LSE, several limitations must be acknowledged:

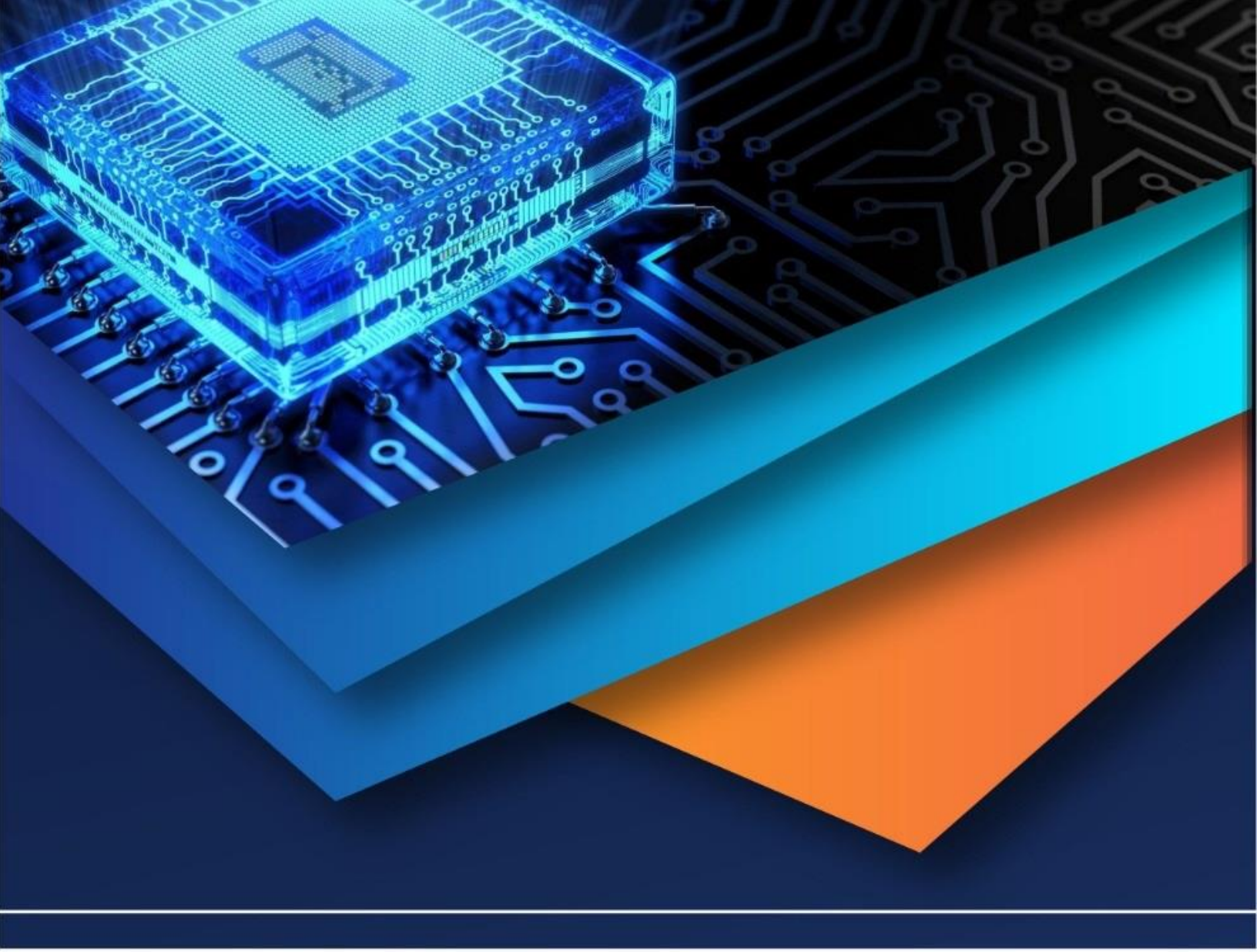
1) *Model Complexity:* The study utilized a simple linear regression model to isolate the effects of the loss functions. However, the linear model is inherently limited in its capacity to capture non-linear relationships, such as those present in the sine function. This choice may have restricted the potential performance of both loss functions.
2) *Dataset Characteristics:* The synthetic dataset, although designed to mimic real-world noise through Gaussian perturbations, represents only a narrow range of potential noise distributions and scales. The findings may not generalize to datasets with different noise characteristics or more complex underlying functions.
3) *Optimization Constraints:* The experiments were conducted using a fixed learning rate and a specific number of epochs. Different hyperparameter settings or alternative optimization strategies could potentially alter the convergence dynamics and performance metrics of both loss functions.

VI. FUTURE SCOPE

Future research should explore the application of these loss functions in more complex, non-linear models such as neural networks, which are better equipped to capture intricate data relationships. Additionally, extending the analysis to diverse noise profiles and real-world datasets could provide deeper insights into the robustness of each loss function under varying conditions. Investigating advanced optimization techniques and adaptive learning rate strategies might further elucidate whether the performance discrepancies observed persist across different training paradigms. Moreover, incorporating regularization methods or developing hybrid loss functions that blend the advantages of both MSE and LSE could lead to improved model stability and accuracy in noisy environments. These avenues of exploration will not only refine our understanding of loss function behavior but also enhance practical applications in machine learning.

BIBLIOGRAPHY

[1] Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006.
[2] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
[3] Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer, 2009.
[4] Scikit-learn. “scikit-learn: Machine Learning in Python.” scikit-learn.org, 2021, https://scikit-learn.org/stable/documentation.html.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY