Table of Contents

[Problem 1 1](#_Toc178850192)

[1a) Derivation of the Mean Squared Error Loss and Its Gradient for Linear Regression 1](#_Toc178850193)

[1b) Why Smooth Loss Functions Are Preferred in Gradient Descent: Issues with MAE 1](#_Toc178850194)

[1c) Optimizing Weights Using Gradient Descent: A Step-by-Step Guide 2](#_Toc178850195)

[1d) Visualizing Gradient Descent: Trajectory of Weight Updates on the Loss Surface 3](#_Toc178850196)

[1 e) Explaining the Gradient Descent Trajectory 5](#_Toc178850197)

[1 f) Visualizing Gradient Descent Divergence: Impact of a Large Learning Rate 6](#_Toc178850198)

[1g) Strategies to Overcome Local Minima in Gradient Descent Optimization 7](#_Toc178850199)

# Problem 1

## 1a) Derivation of the Mean Squared Error Loss and Its Gradient for Linear Regression

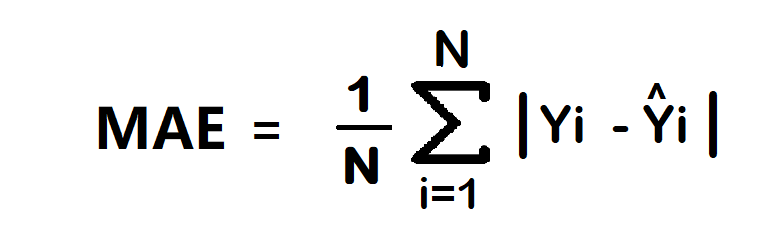
The most common loss function in linear regression is the Mean Squared Error (MSE). This loss function measures the average squared difference between the actual values and the predicted values.

reMarkable formula of the MSE loss function…

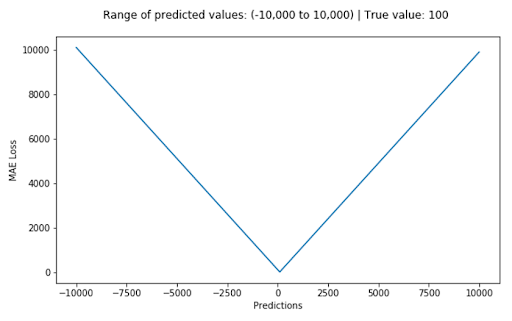
## 1b) Why Smooth Loss Functions Are Preferred in Gradient Descent: Issues with MAE

Smooth loss functions, such as Mean Squared Error (MSE), are preferred in gradient descent due to their continuity and differentiability everywhere. It means that the gradient can be computed without encountering undefined values, facilitating the gradient descent algorithm's ability to efficiently identify the weights which produce the minimal loss.

The Mean Absolute Error (MAE) loss function, on the other hand, is not a smooth function and exhibits significant drawbacks when employed in gradient descent.



The MAE loss function has constant gradients, with -1 for negative values and 1 for positive values. The MAE loss function is also non-differentiable at zero. Figure 1 illustrates this. This results in the derivative of the absolute value at 0 being undefined. This indicates that the MAE loss function will shift between 1 and -1, complicating the convergence of the gradient descent algorithm.



## 1c) Optimizing Weights Using Gradient Descent: A Step-by-Step Guide

This is an analysis of how weights can be optimized through gradient descent thought a step-by-step walkthrough:

* Step 1: Initialize the weights
* Step 2: Compute the gradients for Mean Squared Error loss function.
* Step 3: Update the weights
* Step 4: Repeat this possess for a fixed number of iterations.

In a real life scenario, instead of looping for a fixed number of iterations, this should continue iterating until the variation in the loss function is less than a specified threshold, or when the gradient magnitudes are minimal.

Pseudo-code



## 1d) Visualizing Gradient Descent: Trajectory of Weight Updates on the Loss Surface

Figure X demonstrates the approximately iterative learning process as the weights gets updated over time. I have tried to visualize that the arrows are larger in the beginning and becomes smaller as it approaches the optimal point. This reflects how the gradient decent take longer steps in the beginning and smaller steps as it approaches the optimal point (minimal loss).

A graph of a loss surface

Description automatically generated

## 1 e) Explaining the Gradient Descent Trajectory

Loss surface

The loss surface illustrates different loss levels for different weight combinations of w0 and w1. The contour lines indicate areas of constant loss. The loss surface can be compared topographical map that illustrates fluctuations in the terrain.

* The red star signifies the point of the minimal loss, referred to as the global minimum.
* The blue star signifies the algorithm`s starting point, with the initial weights set at 0.01.
* The yellow star may signify a local minimum.

Gradient descent steps

Gradient descent is an iterative process in which the weights are progressively updated in the direction of the steepest descent. The orientation of the gradients is consistently perpendicular to the contour lines. The gradient vector indicates the direction of the greatest increase of the function, which is consistently perpendicular [1].

The arrows I drew on the loss surface in the previous task signify the trajectory of the weights as they are iteratively updated through gradient descent. The initial starting point is at a peak where the loss is high (the blue star). Initially, the algorithm makes more significant adjustments to its weights due to the elevated loss. As it approaches the global minimum, it makes minor adjustments to the weights. I have attempted to illustrate this by reducing the size of the arrows as it approaches the global minimum.

Learning rate

The learning rate regulates the magnitude of each step in the gradient descent process. Given that the task description indicates the learning rate is sufficiently small, we may conclude that the progression towards the global minimum is controlled. A larger learning rate may impede the function's convergence.

## 1 f) Visualizing Gradient Descent Divergence: Impact of a Large Learning Rate

Figure X illustrates the potential appearance of the weight update process when using an excessively high learning rate.

A diagram of a graph

Description automatically generated

The figure illustrates that the arrows are significantly longer, indicating substantial updates to the weights. When the learning rate is excessively high, the algorithm frequently exhibits a zigzag pattern as it attempts to identify the optimal point, resulting in overly large weight updates that cause overshooting. An excessively high learning rate will result in an endless iterative process that fails to identify the optimal point.

## 1g) Strategies to Overcome Local Minima in Gradient Descent Optimization

Stochastic Gradient Descent (SGD) and Momentum are both effective strategies that can be utilizes in order to avoid ending up in a local minimum during the weight optimalization process.

Stochastic Gradient Descent significantly diverges from Gradient Descent in various aspects, particularly by incorporating some randomness into the optimization process [2]. There are two different approaches of employing SGD:

* Select a sample at random and adjust the weights according to this sample.
* Employing mini-batch processing. This method randomly selects x samples and adjusts its weights according to the samples.

Stochastic Gradient Descent introduces randomness into the optimization process, which can prevent the weights from becoming stuck in a local minimum [2].

Integrating SDG with Momentum is an improved strategy for avoiding a local minimum. The SDG alone does not guarantee an escape from the local minimum; however, when combined with momentum, it enhances the likelihood of overcoming the local minimum and finding the global minimum.   
Momentum operates by employing a weighted combination of past gradients with the current gradient. Unlike 'normal' Gradient Descent, where weight updates are solely based on the gradient of the current iteration, momentum maintains an overall mean of previous gradients. These past gradients are utilized to influence future weight updates [3].

# Problem 2

## 1a) Load the data set eport general information of the data. Plot data as a histogram.

The dataset consists of 2 rows and 3600 columns. The first row contains the features and the second row contains the binary values (1 or 0), representing the class lables.

The figure representates the distribution of two classes (1 and 0). Class 0 (blue) and Class 1 (red) show distinct distributions of feature values. Class 0 features are concentrated in the lower range (mostly between 0 and 10), indicating that features for this class generally have smaller values. Class 1 features dominate the higher range (from 10 to 25), suggesting that the feature values for this class are generally larger.

**Seperability**

Based on figure 5, it looks like the data has good separability between class 0 and class 1. Other than some clear overlap in the feature values between 3 and 10, the classes looks well-seperated. Good seperability between two classes makes it easier for the model to destinguish between the two classes.

References

[1] “Reddit - Dive into anything,” Reddit.com, 2024. <https://www.reddit.com/r/AskPhysics/comments/1av2qvs/why_is_the_gradient_of_a_vector_field_always/> (accessed Oct. 02, 2024).

[2] J. Starmer, “Stochastic Gradient Descent, Clearly Explained!!!,” YouTube. May 13, 2019. Accessed: Oct. 03, 2024. [YouTube Video]. Available: <https://www.youtube.com/watch?v=vMh0zPT0tLI&ab_channel=StatQuestwithJoshStarmer>

[3] J. Brownlee, “Gradient Descent With Momentum from Scratch - MachineLearningMastery.com,” MachineLearningMastery.com, Feb. 04, 2021. <https://machinelearningmastery.com/gradient-descent-with-momentum-from-scratch/> (accessed Oct. 03, 2024).

‌