
# Linear Regression

Regression is a supervised machine learning problem. Unlike classification, where new datapoints are classified as categorical features, regression problems predict the value of a continuous numeric quantity. Regression could be of multiple types:

- Simple Linear Regression
- Multiple Regression
- Logistic Regression
- Ridge and Lasso Regression

This digital note is part of my machine learning docs; find more at my GitHub profile.

## Linear Regression

Simple linear regression models the relationship between two variables. The function is linear because the output can be fit into a straight line. The aim is to find a best-fit line that minimizes the residual error (or squared error) between the actual and predicted datapoints. Statistically, the hypothesis of such a best-fit line could be derived through this linear function:

$$y = mx + b$$

- y is a linear function of x (hypothesis)
- m is the slope of the line (change in the slope concerning change in 1 unit of y)
- b is bias (intercept of graph)

## Mathematic Intuition

Mathematically, the values of slope and intercept can be derived with the following formulas:

$$w = \frac{n\sum(xy) - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$$

$$b = \bar{y} - w\bar{x}$$

---

The intercept for simple linear regression with one variable can also be calculated with the least squares method:

$$b = \frac{\sum y - w\sum x}{n}$$

## Geometric Intuition

Geometrically, we aim to derive the best-fit line that minimizes residual or squared error.
The
**vertical distance** from the point to the line is the **error for each point**. The error (loss) can be calculated with the loss function.

## Loss Function

**Mean Squared Error (MSE)** is the loss function of simple linear regression.

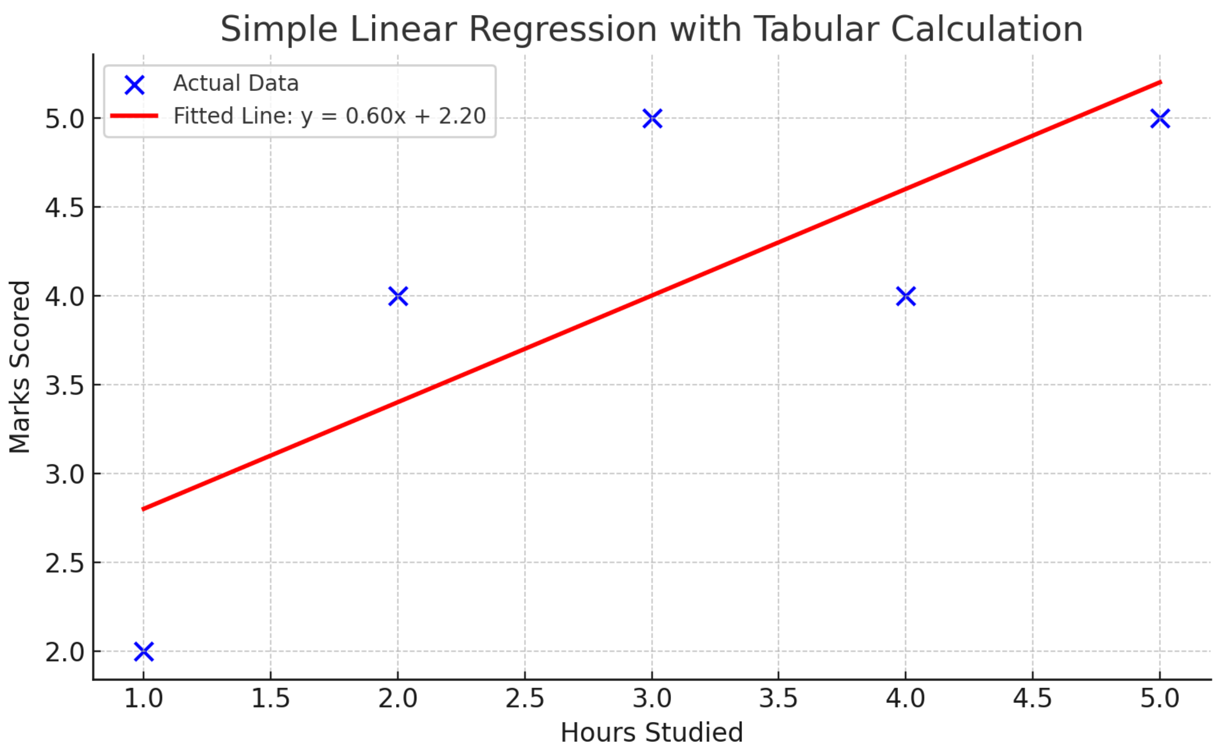
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - (wx_i + b))^2$$

## Simple Linear Regression Problem

| x (hours studied) | y (marks scored) |
| --- | --- |
| 1 | 2 |
| 2 | 4 |
| 3 | 5 |
| 4 | 4 |
| 5 | 5 |

| x | y | x.y | $x^2$ |
| --- | --- | --- | --- |
| 1 | 2 | 2 | 1 |
| 2 | 4 | 8 | 2 |
| 3 | 5 | 15 | 9 |
| 4 | 4 | 16 | 16 |
| 5 | 5 | 25 | 25 |
| **SUMMITION** | 20 | 66 | 55 |

$$y = 0.6x + 2.2$$




## Gradient Descent

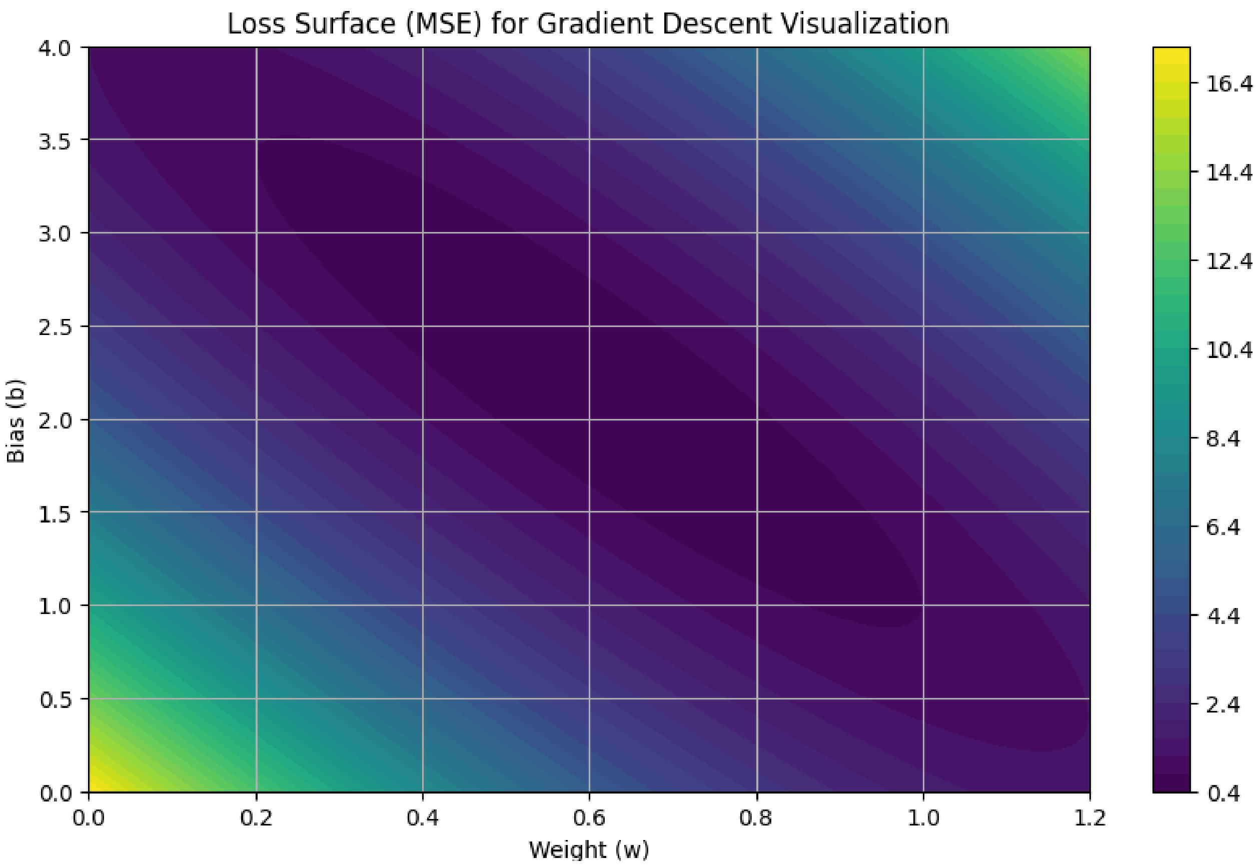
Gradient descent is an optimizer that derives the optimal value of slope and intercept that minimizes the loss function.

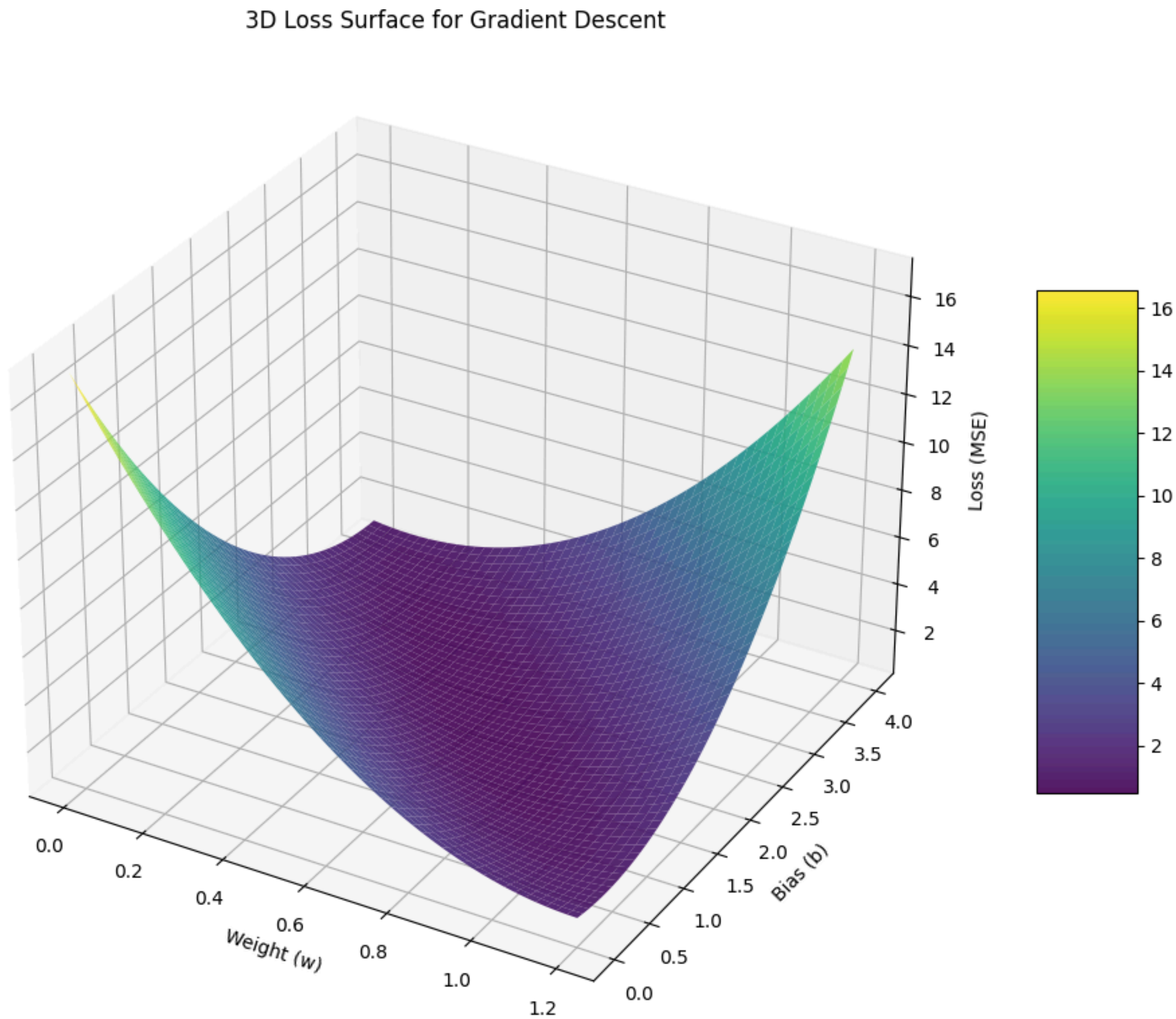
- Start with random values of slope and intercept
- Calculate the gradient of the loss function.
- Update the parameters in the opposite direction of the gradient with the following formula:

$$w := w - \alpha \cdot \frac{\partial \text{MSE}}{\partial w}$$

$$b := b - \alpha \cdot \frac{\partial \text{MSE}}{\partial b}$$

- Alpha is the learning rate and must be chosen appropriately.
- Gradient Descent guarantees convergence to **global minima**.

- 


3D Loss Surface for Gradient Descent
Weight (w)
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Bias (b)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Loss (MSE)
2
4
6
8
10
12
14
16