Gradient Descent

Gradient descent is an optimization algorithm used to minimize the value of a function. It's commonly used in machine learning for tasks such as training neural networks and optimizing model parameters. The basic idea behind gradient descent is to iteratively update the parameters of a function in the direction of the negative gradient of the function with respect to those parameters, thereby gradually reducing the function's value until a minimum is reached.

- Here's the formula for the parameter update step in gradient descent:

$$\theta = \theta - \alpha \nabla J(\theta)$$

where

θ → represents the parameters of the function (e.g., model weights).

α → learning rate

, which controls the size of the steps taken during optimization. It's a hyperparameter that needs to be tuned.

$\nabla J(\theta)$ → Gradient of the cost function J w.r.t θ.

The gradient points in the direction of the steepest increase of the function.

The steps of the gradient descent algorithm are as follows:

1. Initialize the parameters θ randomly or with some predefined values.

2. Compute the gradient of the cost function J with respect to the parameters θ

3.Update the parameters θ by taking a small step in the direction opposite to the gradient.

4.Repeat steps 2 and 3 until convergence, or for a fixed number of iterations.

Gradient descent aims to find the minimum of the cost function by iteratively adjusting the parameters in the direction that reduces the function's value. The learning rate determines the step size of each update, while the gradient provides the direction of the steepest descent. By following the negative gradient, gradient descent can converge to a local or global minimum of the function, depending on its properties and the chosen learning rate.