Gradient Descent Explained

Gradient Descent is an iterative optimization algorithm used to minimize an objective function, such as a cost function, by adjusting its parameters. The goal is to find the parameter values that result in the lowest possible value of the objective function.

Here's a breakdown of how it works:

1. **Purpose**: To find the optimal parameters (θ) that minimize an objective function $J(\theta)$.

2. **Iterative Process**: It repeatedly updates the parameters over several steps.

3. **Update Rule**: In each step, parameters are updated by moving in the direction opposite to the gradient of the objective function. The formula is: $\theta(t+1) = \theta(t) - \alpha * r\_\theta J(\theta(t))$.

* θ**(t)** are the current parameter values.

* α **(learning rate)** is a positive scalar that controls the step size.

* **r**$\_\theta$ **J(**θ**(t))** is the gradient of the objective function, which indicates the direction of the steepest ascent. Moving in the negative gradient direction ensures the steepest descent.

4. **Convergence**: While it can get stuck in local minima for some functions, for convex functions (like the cost function in linear regression), it converges to the global minimum if the learning rate is appropriate.

5. **Variants**:

* **Batch Gradient Descent**: Uses the entire training set to calculate the gradient at each step. This can be slow for large datasets.

* **Stochastic Gradient Descent (SGD)**: Updates parameters based on the gradient of a *single training example* at a time. This is faster for large datasets but can cause oscillations around the minimum.

Is the Loss Function Always Convex?

No, the loss function is not always convex. While some common loss functions like the Negative Log-Likelihood (NLL) loss for Generalized Linear Models (GLM), logistic loss, hinge loss, and exponential loss are convex, there are many cases where they are not. For example, some loss functions can be discontinuous and non-convex. In practice, it's often preferred to choose loss functions that are convex and continuous when possible, as they are easier to optimize.