**Gradient Descent**

Imagine you're hiking up a mountain. You're not just randomly wandering; you're constantly checking the slope and adjusting your path to find the easiest, most efficient way to reach the summit. Gradient descent works similarly. It's an optimization algorithm that iteratively adjusts the parameters of a model to minimize a cost function.

Think of the cost function as the "height" of the mountain. Gradient descent calculates the slope (gradient) at the current position and takes a step in the direction that leads to the steepest descent. By repeatedly taking small steps, the algorithm gradually moves towards the lowest point on the "mountain," which represents the optimal solution.

|  |  |  |
| --- | --- | --- |
|  | [linkedin.com](https://substack.com/redirect/ee934b9a-646f-423f-92d2-cfa8569c842a?j=eyJ1IjoiMzl0ejhqIn0.pKoeDBGzANykuTjTSYrkQwO0iaF_Zw6MrAb-sVXzQ00) |  |

**How Gradient Descent Works ?**

The process begins with an initial guess of the parameters (or weights) of our model. From this point, gradient descent calculates the gradient (or slope) of the cost function at that position. The algorithm then updates the parameters by taking a step in the opposite direction of the gradient, scaled by a factor known as the **learning rate**. This can be summarized in the formula: *θ[new]​ = θ[old] ​− η ∇J(θ)*

Where:

* θ represents the parameters,
* η is the learning rate,
* ∇J(θ) is the gradient of the cost function.

**Importance of Learning Rate**

The learning rate is crucial; if it's too small, convergence to the minimum can be painfully slow, while a rate that's too high may cause divergence, leading to erratic behavior. Thus, finding an optimal learning rate is often a matter of trial and error and requires careful tuning.

**Real-World Applications**

* **Linear Regression**: To minimize the difference between predicted and actual outcomes.
* **Neural Networks**: For adjusting weights during training to improve accuracy.
* **Logistic Regression**: To optimize classification tasks.