# **Gradient Descent**

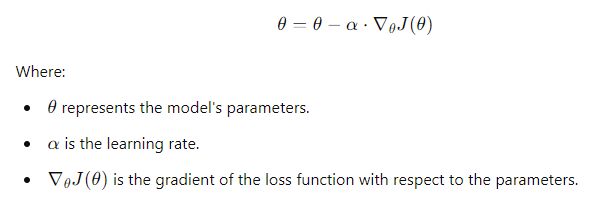
Gradient descent is an optimization algorithm commonly used in machine learning and deep learning to minimize a loss function, which measures the difference between the predicted outputs of a model and the actual data.

**Key Concepts:**

1. **Loss Function**:
   * A mathematical function that measures how well a model's predictions match the actual data. Common loss functions include Mean Squared Error (MSE) for regression and Cross-Entropy Loss for classification.
2. **Gradient**:
   * The gradient is a vector of partial derivatives of the loss function with respect to each parameter in the model. It indicates the direction and rate of fastest increase in the loss function.
3. **Learning Rate**:
   * A small scalar value that determines the step size during each iteration while moving towards a minimum. If the learning rate is too high, the algorithm may overshoot the minimum; if too low, it may take a long time to converge.

**How Gradient Descent Works:**

1. **Initialization**:
   * Start with random values for the parameters (weights and biases in the context of neural networks).
2. **Compute the Gradient**:
   * For each iteration, compute the gradient of the loss function with respect to each parameter. This gradient tells us how much the loss function will change if we adjust the parameters slightly.
3. **Update the Parameters**:
   * Adjust the parameters in the opposite direction of the gradient by a factor of the learning rate. This step moves the parameters closer to the point where the loss function is minimized.



1. **Repeat**:
   * Continue the process until the gradient is close to zero or the improvement in the loss function is negligible, indicating convergence.

**Types of Gradient Descent:**

1. **Batch Gradient Descent**:
   * Uses the entire dataset to compute the gradient. It is very stable but can be slow and computationally expensive.
2. **Stochastic Gradient Descent (SGD)**:
   * Uses only one data point at each iteration to compute the gradient. It is faster and can escape local minima, but it is noisier and less stable.
3. **Mini-Batch Gradient Descent**:
   * Uses a small subset of the data (mini-batch) at each iteration to compute the gradient. It strikes a balance between batch and stochastic gradient descent.

**Visualizing Gradient Descent:**

Imagine standing at the top of a mountain (the loss function) and trying to get to the lowest point (the minimum loss). The gradient tells you which direction to walk to go downhill the fastest, and the learning rate controls how big your steps are. The goal is to reach the lowest point without overshooting or getting stuck in a local dip.

**Applications:**

* **Training Neural Networks**: Gradient descent is used to minimize the loss function and find the optimal weights and biases that allow the network to make accurate predictions.
* **Linear Regression**: In linear regression, gradient descent helps in finding the best-fit line by minimizing the sum of squared errors.