1. Batch Gradient Descent

Definition: Updates model parameters using the average gradient computed over the **entire training dataset** in each iteration.

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

Advantages:

- **Stable convergence due to low-noise gradient updates .**
- **Computationally efficient for small datasets with vectorized operations .**

Disadvantages:

- **Computationally expensive for large datasets .**
- **Requires storing the entire dataset in memory.**

2. Stochastic Gradient Descent (SGD)

Definition: Updates parameters using the gradient of **one randomly selected training example** per iteration.

$$\theta = \theta - \eta \nabla J(\theta; x^{(i)}, y^{(i)})$$

Advantages:

- **Faster updates and convergence for large datasets .**
- **Memory-efficient and suitable for online/streaming data .**

Disadvantages:

- **Noisy updates lead to erratic convergence paths .**
- **Requires careful tuning of the learning rate to avoid overshooting minima .**

3. Mini-Batch Gradient Descent

Definition: Balances Batch and SGD by using **small subsets (mini-batches)** of data for gradient computation.

Advantages:

- **Efficient use of hardware (e.g., GPU parallelization) .**
- **Smoother convergence than SGD and faster than Batch GD .**

$$\theta = \theta - \eta \nabla J(\theta; \{x^{(i)}, y^{(i)}\}_{i=1}^{m})$$

Disadvantages:

- **Requires tuning of batch size and learning rate .**
- **Less stable than Batch GD for very small batch sizes .**

4. Momentum-Based Gradient Descent

Definition: Enhances standard gradient descent by introducing a **velocity term** to accelerate convergence in high-curvature regions.

$$v_t = \gamma v_{t-1} + \eta \nabla J(\theta_t)$$
$$\theta_{t+1} = \theta_t - v_t$$
where γ = momentum term.

Advantages:

- **Reduces oscillations and accelerates convergence .**
- **Effective for escaping shallow local minima .**

Disadvantages:

- **Introduces an additional hyperparameter (γγ) to tune .**
- **Risk of overshooting minima if momentum is too high .**

5. Adam (Adaptive Moment Estimation)

Definition: Combines **momentum** and **adaptive learning rates** by tracking exponentially decaying averages of gradients and squared gradients.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

Advantages:

- **Automatically adapts learning rates per parameter .**
- **Effective for non-convex optimization and deep learning .**

Disadvantages:

- **Complex implementation with multiple hyperparameters .**
- **Potential for convergence instability in certain tasks .**