# Stochastic Gradient Descent (SGD) – Simplified Theory

## 1. Introduction

- **Gradient Descent** is an optimization algorithm that minimizes a **cost function** to find the best model parameters.
- **SGD** is a **variant of Gradient Descent** that updates parameters **one sample at a time** instead of using the whole dataset (like Batch GD).

## 2. Problem with Batch Gradient Descent

- Batch GD computes gradients using **all training samples** before updating weights:

$$\theta_j = \theta_j - \alpha \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) x_{ij}$$

- **Issues:**
  - **Slow** for very large datasets because only **one update per epoch**.
  - Requires **entire dataset in memory** for every calculation.
  - Can get stuck in **local minima** for non-convex functions.

## 3. What is Stochastic Gradient Descent

- **SGD** updates parameters using **one training example** at a time:

$$\theta_j = \theta_j - \alpha (y_i - \hat{y}_i) x_{ij}$$

- This means:
  1. **Faster updates** → weights change more frequently.
  2. Introduces **noise** (not exact gradient direction), which helps escape local minima.
  3. The cost function oscillates but generally approaches the optimum over time.

## 4. Mathematical Difference between BGD and SGD

| Aspect | Batch GD | Stochastic GD (SGD) |
|---|---|---|
| **Gradient Computation** | Uses **all samples** | Uses **one sample** |
| **Update Frequency** | **Once per epoch** | **Every sample** |
| **Convergence** | Stable, slow | Faster, but noisy |
| **Memory** | High (entire dataset) | Low (one sample) |

## 5. Advantages of SGD

- **Faster training** for large datasets.
- Works well for **online learning** (data arriving in streams).
- Helps escape **local minima** due to noise in updates.

---

## 6. Disadvantages of SGD

- Convergence is **less stable**, cost function **fluctuates**.
- Requires careful **tuning of learning rate**.
- If learning rate is too high → algorithm may **diverge**.

---

## 7. When to Use SGD

- When the dataset is:
    1. **Very large** (millions of samples).
    2. **Streaming or online data** where full dataset is unavailable at once.
    3. Non-convex cost function where local minima are a problem.

---

## 8. Learning Schedules

- To improve convergence, the **learning rate (α\alpha)** can be **decreased over time**:

$$\alpha_t = \frac{\alpha_0}{1 + decay \cdot t}$$

Where:

- $\alpha_0$ = initial learning rate.
- $t$ = iteration number.

This ensures **large steps in the beginning**, then **smaller steps** as we approach the minimum.

---

## Key Takeaways

- **SGD updates weights one sample at a time**, making it faster but noisier than Batch GD.
- Works well for **large-scale and online learning problems**.
- Requires **learning rate scheduling** for better convergence.
- Update rule:

$$\theta_j = \theta_j - \alpha(y_i - \hat{y}_i)x_{ij}$$

---