Stochastic Gradient Descent (SGD) – Simple Explanation

What is SGD?

Stochastic Gradient Descent (SGD) is a method used to **train machine learning models**. It helps the model learn from mistakes by slowly changing its weights (parameters) so that the error (loss) becomes smaller.

Meaning of Each Term

- **Gradient:** tells us which direction increases the loss (error).
- **Descent:** means we move in the opposite direction to reduce the loss.
- **Stochastic:** means random, because we use a small random group of data (called a mini-batch) instead of the whole dataset each time.

Therefore, **SGD means taking random small steps down the error hill.**

Why Do We Use SGD?

1. Fast for large datasets – it updates quickly without using the entire dataset every time.
2. Can learn in real-time as new data arrives.
3. The randomness helps prevent the model from getting stuck in local minima.

How Does SGD Work? (Step by Step)

1. Pick a small part of your data (mini-batch).
2. Make predictions using the current model weights.
3. Measure how wrong the predictions are (loss).
4. Find the direction that reduces the loss (gradient).
5. Move slightly in that direction using:

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \times \nabla L$$

where

- θ = model weight
- η = learning rate (step size)
- ∇L = gradient of the loss function

Repeat this process many times until the model performs well.

Real-Life Analogy: Walking Down a Foggy Hill

Imagine you are standing on a foggy mountain and want to reach the bottom (the point of lowest loss). You cannot see far, so you feel the ground around your feet to sense which direction goes downward. You take a small step that way, then repeat. After many small steps, you reach the bottom.

This is how SGD works – taking many small steps in the right direction.

Simple Numeric Example

We want to train a one-variable model:

$$\hat{y} = w \times x$$

Dataset

x	y
1	2
2	4
3	6
4	8

The real relationship is $y = 2x$. Our goal is to find $w \approx 2$.

Step 1: Start with Random Weight

Let the starting weight be $w_0 = 0$ and the learning rate $\eta = 0.1$.

Step 2: First Mini-Batch → (1,2) and (2,4)

$$\text{Predictions: } (0 \times 1, 0 \times 2) = (0, 0)$$

$$\text{Errors: } (0 - 2, 0 - 4) = (-2, -4)$$

$$\text{Gradient: } (-2 \times 1) + (-4 \times 2) = -2 - 8 = -10$$

$$\text{Update Weight: } w_1 = 0 - 0.1 \times (-10) = 1.0$$

Weight after Step 1: $w_1 = 1.0$

Step 3: Second Mini-Batch → (3,6) and (4,8)

$$\text{Predictions: } (1 \times 3, 1 \times 4) = (3, 4)$$

$$\text{Errors: } (3 - 6, 4 - 8) = (-3, -4)$$

$$\text{Gradient: } (-3 \times 3) + (-4 \times 4) = -9 - 16 = -25$$

$$\text{Update Weight: } w_2 = 1.0 - 0.1 \times (-25) = 3.5$$

Weight after Step 2: $w_2 = 3.5$

Result

The weight w moved as follows:

$$0 \rightarrow 1.0 \rightarrow 3.5$$

After more steps, w will settle near 2, which is the correct value.

When to Use SGD

- When the dataset is very large and using the full dataset at once is not practical.
- When quick, continuous updates are needed.
- When a simple and effective optimization method is preferred.
- Commonly used in deep learning, image classification, and natural language processing.

Pseudo-Code for SGD

```
initialize weight w randomly
repeat for several epochs:
    shuffle the data
    for each mini-batch:
        find gradient g
        update weight: w = w - learning_rate * g
```