Stochastic Gradient Descent (SGD) updates weights based on a single sample at a time, rather than the entire dataset, making it computationally efficient for large datasets. Let's go through an example of **SGD** with detailed calculations for a house price prediction task.

Step 1: Dataset and Model

Dataset

We are predicting house prices using two features:

- x_1 = Area (sq. ft.)
- x_2 = Bedrooms

x_1 (Area)	x_2 (Bedrooms)	y (Price)
2600	3	550000
3000	4	565000
3200	3	610000
3600	5	680000

Model

The model equation is:

$$\text{Predicted Price} = w_1 \cdot x_1 + w_2 \cdot x_2 + \text{bias}$$

Initial Parameters:

- $w_1 = 0.5$, $w_2 = 0.5$, $\text{bias} = 0$
- Learning rate (α) = 0.000000001

Step 2: SGD Process

Iteration 1

For each sample, we calculate:

1. Predicted price
2. Error
3. Gradients
4. Update the weights and bias

Sample 1:

- $x_1 = 2600, x_2 = 3, y = 550000$

1. **Predicted Price**:

$$\text{Predicted} = w_1 \cdot x_1 + w_2 \cdot x_2 + \text{bias} = 0.5 \cdot 2600 + 0.5 \cdot 3 + 0 = 1301.5$$

2. **Error**:

$$\text{Error} = \text{Predicted} - y = 1301.5 - 550000 = -548698.5$$

3. **Gradients**:
   - $\frac{\partial MSE}{\partial w_1} = 2 \cdot \text{Error} \cdot x_1 = 2 \cdot (-548698.5) \cdot 2600 = -2853230200$
   - $\frac{\partial MSE}{\partial w_2} = 2 \cdot \text{Error} \cdot x_2 = 2 \cdot (-548698.5) \cdot 3 = -3292191$
   - $\frac{\partial MSE}{\partial \text{bias}} = 2 \cdot \text{Error} = 2 \cdot (-548698.5) = -1097397$
4. **Weight Updates**:
   - $w_1 = w_1 - \alpha \cdot \frac{\partial MSE}{\partial w_1} = 0.5 - 0.000000001 \cdot (-2853230200) = 0.50000285323$
   - $w_2 = w_2 - \alpha \cdot \frac{\partial MSE}{\partial w_2} = 0.5 - 0.000000001 \cdot (-3292191) = 0.50000032922$
   - $\text{bias} = \text{bias} - \alpha \cdot \frac{\partial MSE}{\partial \text{bias}} = 0 - 0.000000001 \cdot (-1097397) = 0.0010974$

Sample 2:

- $x_1 = 3000, x_2 = 4, y = 565000$

1. **Predicted Price**:

$$\text{Predicted} = w_1 \cdot x_1 + w_2 \cdot x_2 + \text{bias} = 0.50000285323 \cdot 3000 + 0.50000032922 \cdot 4 + 0.0010974 = 1500.0198$$

2. **Error**:

$$\text{Error} = \text{Predicted} - y = 1500.0198 - 565000 = -563499.98$$

3. **Gradients**:
   - $\frac{\partial MSE}{\partial w_1} = 2 \cdot \text{Error} \cdot x_1 = 2 \cdot (-563499.98) \cdot 3000 = -3380999880$
   - $\frac{\partial MSE}{\partial w_2} = 2 \cdot \text{Error} \cdot x_2 = 2 \cdot (-563499.98) \cdot 4 = -4507999.84$
   - $\frac{\partial MSE}{\partial \text{bias}} = 2 \cdot \text{Error} = 2 \cdot (-563499.98) = -1126999.96$
4. **Weight Updates**:
   - $w_1 = 0.50000285323 - 0.000000001 \cdot (-3380999880) = 0.50000623432$
   - $w_2 = 0.50000032922 - 0.000000001 \cdot (-4507999.84) = 0.50000078002$
   - $\text{bias} = 0.0010974 - 0.000000001 \cdot (-1126999.96) = 0.0022244$

Repeat for All Samples

After 4 samples:

The weights (w_1, w_2) and bias are updated after **every single sample**, not after the entire dataset. This is the essence of SGD.

Key Points:

1. **Stochastic Gradient Descent** updates weights after each sample, making it faster and more responsive to the latest data points.
2. In large datasets, it converges faster than batch gradient descent.
3. **Implementation in Python** (optional) can demonstrate this concept practically.