Comparison of Gradient Descent (GD), Stochastic Gradient Descent (SGD), and Mini-Batch Gradient Descent (MBGD)

Type	Description	When to Use	Suitable Datasets
Gradient Descent (GD)	Updates weights using the entire dataset at once.	- When the dataset is **small** and can fit in memory.	- Small datasets where computational resources are not a concern.
Stochastic Gradient Descent (SGD)	Updates weights for **each data point** individually.	- When the dataset is **large** and cannot be processed entirely at once.	- Large datasets, particularly online learning or streaming data.
Mini-Batch Gradient Descent (MBGD)	Updates weights using a small, random subset of the dataset (mini-batch).	- When the dataset is **large**, but you want a balance between computational efficiency and stability.	- Large datasets with a need for faster computation without sacrificing stability.

In-Depth Use Cases for Each Gradient Descent

1. Gradient Descent (GD):

- **How It Works**:
  - Computes the gradients for the entire dataset at each step.
  - Updates weights after calculating the Mean Squared Error (MSE) across all samples.
- **Pros**: Stable convergence, ensures a global look at the loss function.
- **Cons**: Computationally expensive for large datasets.
- **Examples**:
  - Small datasets (e.g., **house price prediction** with $< 10,000$ samples).
  - Solving mathematical optimization problems in **scientific computation**.

---

2. Stochastic Gradient Descent (SGD):

- **How It Works**:
  - Computes gradients and updates weights for each individual sample.
  - Introduces randomness, which can help escape local minima.
- **Pros**:
  - Faster updates, suitable for streaming or online data.
  - Useful for non-convex optimization problems.
- **Cons**:
  - High variance in updates, leading to less stable convergence.
- **Examples**:
  - **Dynamic datasets** (e.g., predicting website traffic using real-time data).
  - **Recommender systems** (e.g., Netflix or Amazon recommendations).

---

3. Mini-Batch Gradient Descent (MBGD):

- **How It Works**:
  - Combines the stability of GD and the speed of SGD.
  - Splits the dataset into mini-batches (e.g., batch size = 16, 32, or 64).
- **Pros**:
  - Computationally efficient due to vectorized operations.
  - Reduces memory usage compared to full-batch GD.
  - More stable updates compared to SGD.
- **Cons**:
  - Requires tuning the batch size for optimal performance.
- **Examples**:
  - **Deep learning** (training neural networks where datasets are large, like ImageNet).
  - **Financial data**: Predicting stock prices from historical trends.

Choosing the Right Gradient Descent for Your Dataset

Dataset Characteristics	Recommended GD
Small datasets (<10,000 rows)	**Gradient Descent (GD)**: Stable and accurate, no need for batch splitting.
Large datasets (>10,000 rows)	**Mini-Batch Gradient Descent (MBGD)**: Strikes a balance between stability and computational efficiency.
Streaming data or online tasks	**Stochastic Gradient Descent (SGD)**: Efficient and ideal for constantly changing or incoming data.
Highly noisy data	**Mini-Batch Gradient Descent (MBGD)**: Mitigates the instability of SGD by averaging out some of the noise per batch.
Deep learning applications	**Mini-Batch Gradient Descent (MBGD)**: Faster convergence due to GPU optimization and manageable memory requirements.