Variants of Gradient Descent Algorithms

Batch Gradient Descent (BGD) and Stochastic Gradient Descent (SGD) are two optimization techniques used to minimize the loss function in machine learning models. The key difference between them lies in how they update the model parameters.

1. Batch Gradient Descent (BGD) [Also known as the Vanilla GD]

- Uses the entire dataset to compute the gradient of the cost function before updating the parameters.
- Ensures a smooth and stable convergence since it uses the average gradient over all samples.
- Computationally expensive for large datasets because it requires processing the entire dataset in each iteration.
- Slower updates since the model parameters are updated only after processing all samples.
- More likely to converge to the global minimum but can get stuck in local minima for non-convex functions.

2. Stochastic Gradient Descent (SGD)

- Uses only **one** randomly selected data point per iteration to compute the gradient and update the parameters.
- Much faster than BGD because it updates parameters more frequently.
- Since it updates frequently with noisy gradients, it does not necessarily converge smoothly and can oscillate around the minimum.
- Less memory-intensive as it processes only one data point at a time.
- Can escape local minima due to its noisy nature, which can be an advantage in non-convex optimization.

Comparison Summary

Feature	Batch Gradient Descent	Stochastic Gradient Descent
Gradient Calculation	Uses the entire dataset	Uses one random sample
Update Frequency	Once per epoch	After every sample
Convergence	More stable but slower	Noisy but faster
Computational Cost	High for large datasets	Lower and scalable
Suitability	Small to medium datasets	Large datasets

Mini-Batch Gradient Descent (MBGD)

A compromise between the two is **Mini-Batch Gradient Descent**, where a small batch of samples (e.g., 32 or 64) is used to compute the gradient at each step. This balances the efficiency of SGD with the stability of BGD.