# Gradient Descent Models Comparison and Techniques to Avoid Local Minima

Gradient descent is an optimization algorithm commonly used in machine learning to minimize functions by iteratively moving in the direction of steepest descent. There are three primary variants of gradient descent, each with unique characteristics: Batch Gradient Descent, Mini-Batch Gradient Descent, and Stochastic Gradient Descent (SGD). This report discusses the distinctions between these variants and presents methods for avoiding local minima in optimization problems.

## 1. Batch Gradient Descent

Batch Gradient Descent calculates the gradient of the cost function with respect to the entire dataset and updates the model parameters in one large step. This variant is stable, but it requires significant memory and computational resources, making it slow for large datasets. Batch Gradient Descent is known for its precise direction toward the global minimum, but it may overfit and be computationally heavy.

## 2. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent updates the model parameters using only one sample at a time. This approach introduces noise, resulting in a less stable convergence but enabling the algorithm to escape local minima more effectively. SGD is much faster than Batch Gradient Descent, especially for large datasets, but it can be highly noisy, sometimes oscillating around the optimal solution.

## 3. Mini-Batch Gradient Descent

Mini-Batch Gradient Descent combines aspects of both Batch Gradient Descent and SGD. It divides the dataset into small batches, updating model parameters after computing the gradient for each batch. This approach balances the trade-off between the stability of Batch Gradient Descent and the speed of SGD, making it widely used for deep learning tasks. Mini-batch helps improve computational efficiency and reduces memory usage while smoothing the learning process compared to SGD.

## Comparison of Gradient Descent Variants

The table below summarizes the key differences between Batch Gradient Descent, SGD, and Mini-Batch Gradient Descent:

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Memory Requirement | Convergence Stability | Time Efficiency |
| Batch Gradient Descent | High | High | Slow |
| Stochastic Gradient Descent (SGD) | Low | Low (Noisy) | Fast |
| Mini-Batch Gradient Descent | Moderate | Moderate | Moderate to Fast |

## Techniques to Avoid Local Minima

### 1. Momentum

Momentum is a technique that adds a fraction of the previous update to the current update. It helps smooth the optimization path by dampening oscillations and speeding up convergence, particularly useful when navigating regions with sharp gradients.

### 2. Nesterov Accelerated Gradient (NAG)

Nesterov Accelerated Gradient improves upon Momentum by calculating the gradient at an approximate future position, allowing the optimization algorithm to adjust its direction more accurately. NAG provides faster convergence rates and enhances stability when the function is non-convex.

### 3. Adaptive Gradient Methods (AdaGrad, RMSProp, Adam)

Adaptive gradient methods, including AdaGrad, RMSProp, and Adam, adjust the learning rate based on historical gradients. AdaGrad adapts the learning rate for each parameter, RMSProp handles decaying averages, and Adam combines RMSProp with momentum, making it effective for complex problems. These methods help to avoid local minima by dynamically adjusting learning rates.

### 4. Learning Rate Schedules

Learning rate schedules involve gradually reducing the learning rate as the training progresses. This strategy allows the algorithm to take larger steps initially to escape local minima and smaller steps as it nears convergence for better precision.