Properties of the Sign Gradient Descent Algorithm

Syed Farhan Syed Sathik Basha - CS23B2039
Sivaniranjan - CS23B2040

Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram

December 14, 2024

Table of Contents

Problem Definition

Challenges with Traditional GD:

- Slow convergence due to lack of curvature consideration.
- Sensitivity to initialization.
- Local minima entrapment.

Why Sign Gradient Descent?

- Reduces sensitivity to noise by considering only gradient direction.
- Provides robustness in non-smooth optimization landscapes.

Objective

Primary Goals:

- Analyze convergence properties of SGD, ASGD, and HGD.
- Evaluate SGD's ability to escape local minima and find global minima.
- Validate results using test functions:
  - Trimodal Function:

$$f(x) = \sin(x) + \sin(3x) + \sin(5x) + 3 \tag{1}$$

  - Quadratic Function:

$$f(x) = x^2 + 4x + 4 \tag{2}$$

Basic Optimization Methods

- **Gradient Descent (GD):**

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k) \tag{3}$$

- **Sign Gradient Descent (SGD):**

$$x_{k+1} = x_k - \gamma_k \operatorname{sgn}(\nabla f(x_k)) \tag{4}$$

- **Adaptive SGD (ASGD):**

$$\gamma_{k+1} = \gamma_k \cdot \text{decay\_rate} \tag{5}$$

- **Hybrid GD (HGD):**

$$x_{k+1} = x_k - \gamma_{k,1} \nabla f(x_k) - \gamma_{k,2} \operatorname{sgn}(\nabla f(x_k)) \tag{6}$$

Sign Function and Euclidean Norm

Gradient Direction:

$$\operatorname{sgn}(\nabla f(x)) = \left(\operatorname{sgn}\left(\frac{\partial f(x)}{\partial x_1}\right), \ldots, \operatorname{sgn}\left(\frac{\partial f(x)}{\partial x_n}\right)\right)^T \tag{7}$$

Euclidean Distance:

$$\|x_{k+1} - x_k\|^2 \tag{8}$$

Measures step size stability and convergence robustness.

Sign Gradient Descent Algorithms: Notation and Definitions

- Let $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$.
- The gradient of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)^T \tag{9}$$

- A critical point $x^* \in \mathbb{R}^n$ satisfies $\nabla f(x^*) = 0$.
- The sign of the gradient is defined as:

$$\mathrm{sgn}(\nabla f(x)) = \left(\mathrm{sgn}\left(\frac{\partial f(x)}{\partial x_1} \right), \ldots, \mathrm{sgn}\left(\frac{\partial f(x)}{\partial x_n} \right) \right)^T \tag{10}$$

Steps for Sign Gradient Descent Algorithm

1. **Initialization:**
   - Choose an initial point x_0 in the parameter space. This could be random or based on prior knowledge.
   - Set the initial learning rate γ_0. This can be a fixed value or subject to adjustment over iterations.
2. **Define the Objective Function:**
   - Specify the objective function $f(x)$ that you want to minimize. This function should ideally be differentiable to compute the gradient.
3. **Compute the Gradient:**
   - At each iteration k, compute the gradient of the objective function at the current point:

$$\nabla f(x_k) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

Steps for Sign Gradient Descent Algorithm (Continued)

4. **Compute the Sign of the Gradient:**
   - Determine the sign of the gradient:

$$\operatorname{sgn}(\nabla f(x_k)) = \left(\operatorname{sgn}\left(\frac{\partial f(x_k)}{\partial x_1}\right), \ldots, \operatorname{sgn}\left(\frac{\partial f(x_k)}{\partial x_n}\right)\right)^T$$

   - The sign function is defined as:

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

5. **Update the Parameters:**
   - Update the current point using the sign of the gradient:

$$x_{k+1} = x_k - \gamma_k \cdot \operatorname{sgn}(\nabla f(x_k))$$

Steps for Sign Gradient Descent Algorithm (Continued)

6. **Stopping Criteria:**
   - Define a criterion to stop the algorithm, such as:
     - A maximum number of iterations $k_{\max}$.
     - The change in the objective function being below a certain threshold:

$$|f(x_{k+1}) - f(x_k)| < \epsilon$$

     - The norm of the gradient being sufficiently small:

$$\|\nabla f(x_{k+1})\| < \delta$$

7. **Iteration:**
   - Repeat steps 3 to 7 until one of the stopping criteria is met.
8. **Output:**
   - Once the stopping criteria are met, output the final parameters x_k and the corresponding function value $f(x_k)$.

Action Plan

1. Define test functions (Trimodal, Quadratic, Cubic, Quartic).
2. Compare GD, SGD, and ASGD for:
   - Convergence time
   - Steps to local/global minima
3. Visualize optimization paths.
4. Analyze SGD limitations and propose improvements.

Expected Outcomes

- Faster convergence with SGD and ASGD compared to GD.
- Robustness of SGD to noisy gradients.
- Trade-offs:
  - High step size: Risk of overshooting.
  - Small step size: Difficulty in escaping local minima.

Visual Results

Performance Table:

Method	**Iterations to Convergence**	**Convergence Time (s)**
GD	X iterations	Y seconds
SGD	P iterations	Q seconds
ASGD	R iterations	S seconds

Observations:

- GD converges reliably but requires more iterations and time.
- SGD converges faster but may exhibit instability for high step sizes.
- ASGD combines speed and stability with adaptive step sizes.

Global Minima Analysis

Key Observations:

- **High Step Size (γ_k):**
  - Frequent overshooting prevents convergence.
  - May oscillate around local minima.
- **Small Step Size (γ_k):**
  - Unable to escape shallow local minima.
  - Prolonged convergence time.

Limitation:

- Lack of curvature information limits precision near global minima.

Conclusion

Summary:

- SGD improves robustness and speed over GD.
- ASGD balances speed and stability with dynamic step sizes.
- Limitations include:
  - Overshooting for high step size.
  - Stagnation for small step size.
- Combining SGD with curvature consideration may improve global minima search.