The BFGS Optimization Algorithm

Pratham Lalwani

UC Merced

May 13, 2025

Outline

Problem Setup

- Given $f : \mathbb{R}^n \to \mathbb{R}$, say we are interested is minimizing the function, which is,

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x}).$$

- From calculus, $\nabla f(\boldsymbol{x}) = \mathbf{0}$ and solve analytically if it can be done.
- This problem arises everywhere especially nowadays with Machine Learning where $f(\boldsymbol{x})$ is usually a cost function we are trying to minimize.

Gradient Descent and Its Limitations

With no way to compute a analytic solution one might turn a simple algorithm like Gradient Descent.

- The next iterate is given by : $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma \nabla f(\boldsymbol{x}_k)$, where γ is a fixed constant called step size or learning rate.
- Pros: simple, easy to implement and not computationally expensive (per step)
- Cons: slow convergence, sensitivity to step size.

Gradient Descent

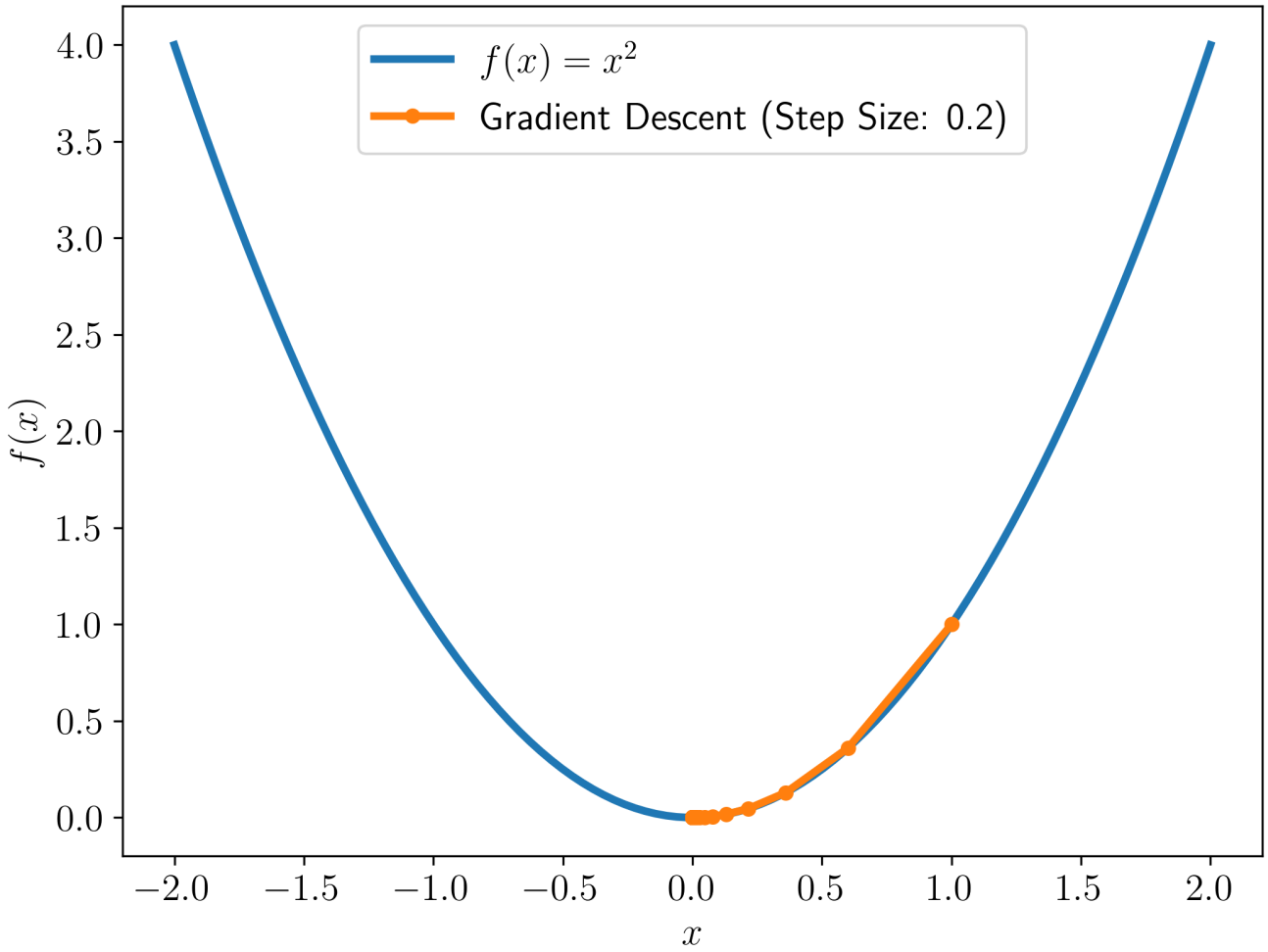

Figure: Gradient Descent on x^2

Gradient Descent

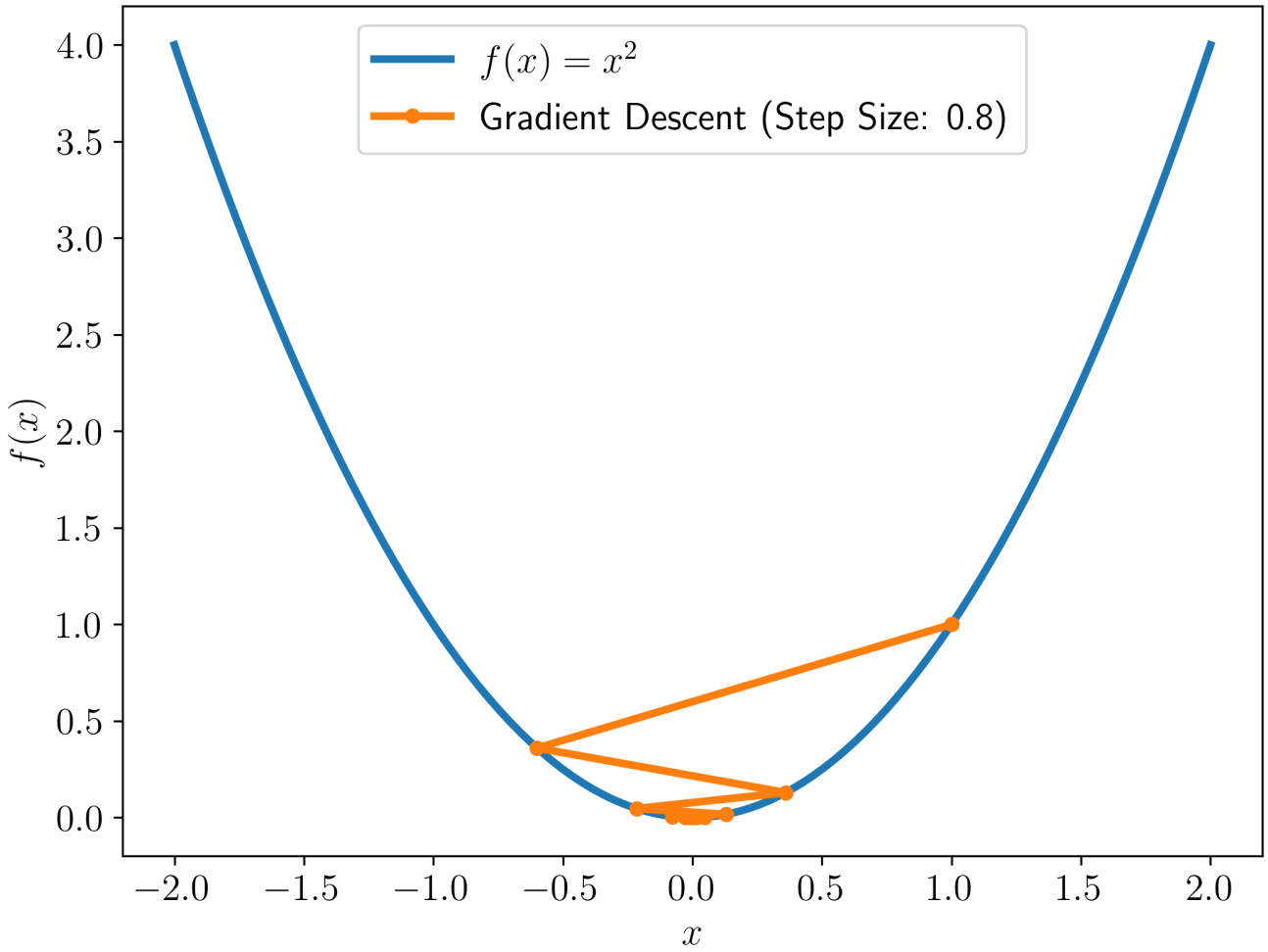

Figure: Gradient Descent on x^2

But wait a minute

Instead of having a fixed Learning Rate what if we have a adaptive learning rate, which is "greedy". If we have,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}).$$

But wait a minute

Instead of having a fixed Learning Rate what if we have a adaptive learning rate, which is "greedy". If we have,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}).$$

We define,

$$\phi(\alpha) := f(\boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x})).$$

History of Quasi-Newton Algorithms

- Aim: approximate Hessian without second derivatives
- Early methods: Davidon–Fletcher–Powell (DFP)
- Development of symmetric rank-one (SR1), BFGS

The BFGS Algorithm

- Builds positive-definite Hessian approximation B_k
- Update: $B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}$
- Search direction: $p_k = -B_k^{-1} \nabla f(x_k)$
- Widely used: stable and fast convergence

Rosenbrock Example

- Test function: $f(x, y) = (a - x)^2 + b(y - x^2)^2$
- Typical parameters: $a = 1, b = 100$
- Illustrates curved valley and optimization challenge

Rosenbrock Example Results

Figure: BFGS optimization path on Rosenbrock function

Results

- Converges in fewer iterations compared to gradient descent
- Good performance on moderate-scale problems
- Memory cost: $O(n^2)$ storage for Hessian approximation

Convergence of BFGS

- Locally superlinear convergence under standard assumptions
- Requires line search satisfying Wolfe conditions
- Practical performance often close to Newton's method

Line Search

- Determines step length α_k along direction p_k
- Common strategies: backtracking, Wolfe conditions
- Ensures sufficient decrease and curvature conditions

Demonstration of Line Search

Figure: Illustration of backtracking line search

Conclusion

- BFGS balances efficiency and robustness
- Widely used in unconstrained optimization
- Extensions: limited-memory L-BFGS for large-scale problems