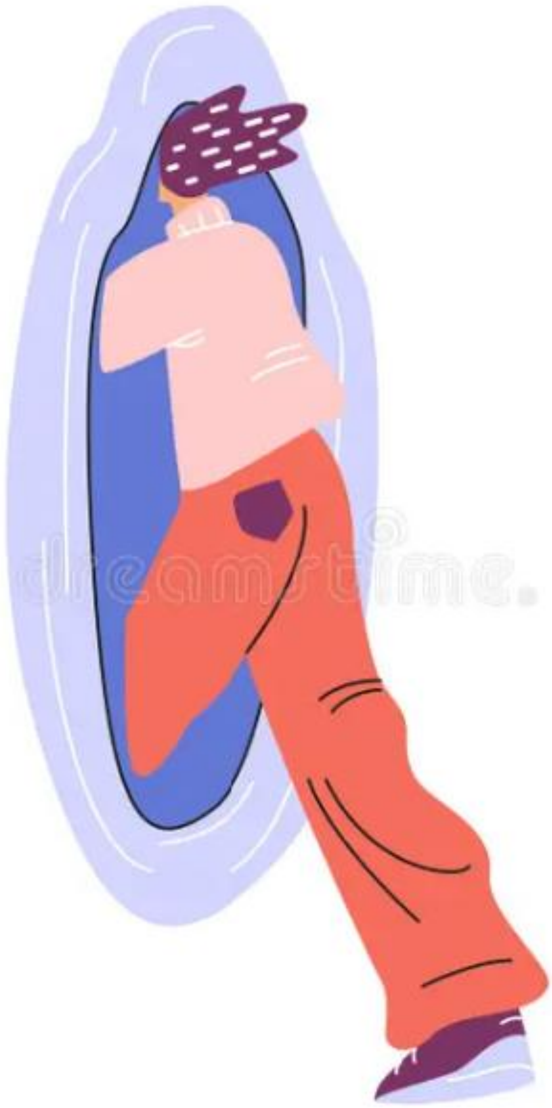
Why Walk Towards the Minimum When You Can Just Teleport?



JUNAID

Research Scholar – Data Science

Why Walk Towards the Minimum When You Can Just Teleport?

A Comparative Study of Optimization Techniques in Linear Regression

---

Introduction

In machine learning, optimization algorithms lie at the heart of training models. Among these, Gradient Descent is one of the most widely used due to its simplicity and low computational cost — but it's also relatively slow, often requiring hundreds or thousands of iterations to converge.

On the other extreme lies Newton's Method, known for its rapid convergence using second-order derivatives via the Hessian matrix. While fast, it comes with a tradeoff: computing and inverting the Hessian is expensive, especially for high-dimensional data.

Between these two extremes is BFGS (Broyden–Fletcher–Goldfarb–Shanno), a quasi-Newton method that strikes a balance. It approximates the Hessian rather than computing it directly, making it faster than Gradient Descent but less computationally intense than Newton's Method.

In this mini-study, I compare all three approaches in the context of linear regression and explore how second-order methods can feel like "teleporting" to the optimal solution — while first-order methods take the scenic route.

Dataset Overview

Area (sq. ft)	Price (Lakhs)
656	39.0
1260	83.2
1057	86.6
1259	59.0
1800	140.0
1325	80.1
1085	116.0
1110	45.0
1700	100.0
960	89.0
1800	90.0
700	49.0

Model Training and Results

We applied **Linear Regression** using different optimization strategies:

1. Scikit-learn's Linear Regression (Closed-form solution)

- Intercept: 18.0465
- Coefficient: 0.0517
- Accurate
- Instant solution (solves Normal Equation)

2. Gradient Descent (First-Order Optimization)

- Step Size: 0.1
- Convergence: 950 epochs
- Result: Same as scikit-learn
- Slower convergence
- Requires careful tuning of learning rate

3. Newton's Method (Second-Order Optimization)

- Uses: Hessian matrix (second derivatives)
- Convergence: 1 epoch
- Instantly converged to the same solution
- Feels like teleporting to the minimum

4. BFGS (Quasi-Newton Method)

- Method: Approximates Hessian
- Convergence: 4 iterations
- Function evaluations: 6
- Gradient evaluations: 6
- Intercept: 18.0465
- Coefficient: 0.0517
- High efficiency and low computational cost
- No need to compute exact Hessian

Analysis Table

Method	Epochs	Requires Hessian?	Speed	Accuracy
Gradient Descent	950	❌	🐢	✅
Newton’s Method	1	✅	🚀	✅
BFGS	4	⚠️ Approximate Hessian	🚀	✅

Limitations

- Newton's Method requires computation of the Hessian, which becomes expensive for high-dimensional problems.
- Gradient Descent requires careful tuning of the learning rate and many iterations.
- BFGS might still struggle with non-convex surfaces or noisy data.

Future Directions

- Apply these methods to larger, more complex datasets (non-linear, multivariate).
- Explore their effectiveness in deep learning contexts.
- Compare with adaptive optimizers like Adam or RMSprop.
- Implement regularization techniques (L1/L2) and observe changes in convergence behavior.

Conclusion

While Gradient Descent is a reliable workhorse, this study shows that smarter, second-order methods like Newton's and BFGS can drastically reduce convergence time. Instead of walking step by step down the loss curve, these methods leap directly to the bottom — making them efficient and powerful tools in the right context.