Newton's Method

Newton's method is a powerful iterative approach within second-order optimization algorithms. It aims to find the minimum (or maximum) of a function by utilizing both the gradient (first derivative) and the Hessian (second derivative) of the objective function.

Objective:

The goal is to find a point where the function's gradient is zero, indicating a local minimum or maximum.

Steps:

1. **Start with an initial guess:** This could be any point on the landscape (function).

2. **Calculate the gradient and Hessian:** The gradient tells you the direction of steepest descent (or ascent), while the Hessian tells you about the curvature (cavity or bump) at that point.

3. **Update Rule**: Update the current estimate (x_k) using the formula:

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

4. **Find the minimum of the quadratic model:** This minimum acts as an approximation for the minimum of the actual function.

Hint:

- Hessian is positive definite (eigenvalues positive) → to identify local minima

- Hessian is negative definite (eigenvalues negative) → to identify local maxima

5. **Update your position:** Move to the point corresponding to the minimum of the quadratic approximation.

6. **Repeat:** Go back to step 2 and iterate with the new position as your starting point.

Advantages:

- **Fast Convergence:** By considering curvature, Newton's method can take larger steps towards the minimum compared to first-order methods, leading to faster convergence, especially for well-behaved functions.
- Newton's method is a strong recommendation for solving optimization problems involving convex functions. Its guaranteed convergence and fast convergence properties make it a powerful tool.

Disadvantages:

- **Computational Cost:** Calculating the Hessian and potentially inverting it can be expensive, particularly for high-dimensional problems.
- **Sensitivity to Initial Guess:** Newton's method can converge slowly or even diverge if the initial guess is far from the minimum or if the function has a complex landscape.
- Computing inverse Hessian explicitly is too expensive
  • O(k^3) if there are k model parameters: inverting a k x k matrix