Question:

Assume f(x) = x2 ∇f(x) = 2x ∇f(0) = 0 0 is a minimizer But, assume g(x) = -x2 ∇g(x) = -2x ∇g(0) = 0 0 is not a minimizer why

Answer:

The function f(x) = x^2 is a convex function, which means that it has a single global minimum at x=0. The gradient of f(x) is ∇f(x) = 2x, which is 0 at x=0. Therefore, 0 is a minimizer of f(x).

On the other hand, the function g(x) = -x^2 is a concave function, which means that it has a single global maximum at x=0. The gradient of g(x) is ∇g(x) = -2x, which is also 0 at x=0. However, since g(x) is a concave function, the maximum at x=0 is a "maximizer" rather than a "minimizer". Therefore, 0 is not a minimizer of g(x), but rather a maximizer.

In summary, the reason why 0 is a minimizer of f(x) but not of g(x) is due to the difference in the convexity/concavity of the two functions.

Question:

smooth vs non smooth function with example

Answer:

In optimization, a smooth function is a function that has a continuous first derivative, while a non-smooth function is a function that does not have a continuous first derivative or has no derivative at certain points.

Here are some examples:

Smooth function: f(x) = x^2 - 3x + 2 This function is smooth because its first derivative, f'(x) = 2x - 3, is continuous for all values of x.

Non-smooth function: g(x) = |x| This function is non-smooth because it has no derivative at x=0.

Another example of a non-smooth function is the absolute value function: h(x) = |x - 1| This function is non-smooth because its first derivative, h'(x), is discontinuous at x=1.

Non-smooth functions can pose challenges for optimization algorithms, as they may not have well-defined gradients or Hessians at certain points. Therefore, optimization techniques designed for smooth functions may not be suitable for non-smooth functions. Specialized methods, such as subgradient methods or proximal gradient methods, may be needed to handle non-smooth functions.

Question:

how to find a function is non smooth

Answer:

To determine if a function is non-smooth, you need to check if it has a continuous first derivative or if the derivative is undefined at certain points.

Here are some steps you can follow to check if a function is non-smooth:

1. Compute the first derivative of the function. If the function is non-smooth, its first derivative may not be continuous or may not exist at certain points. For example, the absolute value function f(x) = |x| is non-smooth at x = 0 because its derivative is undefined at that point.
2. Check for points where the first derivative is undefined. Non-smooth functions often have points where the derivative is undefined, such as vertical tangents or corners. For example, the function g(x) = max(x,0) is non-smooth at x = 0 because its derivative is undefined at that point.
3. Look for discontinuities in the first derivative. Non-smooth functions may have discontinuities in their first derivative, which means that the derivative is not continuous. For example, the function h(x) = |x-1| is non-smooth at x=1 because its derivative is discontinuous at that point.
4. Check for other properties that may indicate non-smoothness, such as sharp bends or kinks in the graph of the function.

In summary, to determine if a function is non-smooth, you need to look for points where the derivative is undefined or discontinuous, as well as other properties that may indicate non-smoothness.

Question:

Quasi-Newton Method 2 advantages and 2 disadvantages

Answer:

Quasi-Newton methods are a class of optimization algorithms that approximate the Hessian matrix of the objective function using only the first-order derivative information. Here are two advantages and two disadvantages of Quasi-Newton methods:

Advantages:

1. Faster convergence: Quasi-Newton methods can converge faster than first-order methods because they use second-order information to update the search direction. This can lead to fewer iterations and faster convergence to a solution.
2. No need for exact Hessian: Quasi-Newton methods do not require an exact calculation of the Hessian matrix, which can be computationally expensive or even impossible for some functions. Instead, they use an approximation of the Hessian matrix based on the first-order derivative information, which can be easier and less expensive to compute.

Disadvantages:

1. Memory requirements: Quasi-Newton methods require storage of a matrix that approximates the Hessian, which can be memory-intensive for large problems. This can limit the size of the problem that can be solved with Quasi-Newton methods.
2. Accuracy of Hessian approximation: The accuracy of the Hessian approximation can affect the performance of Quasi-Newton methods. If the approximation is poor, it may take more iterations for the algorithm to converge, or it may not converge at all. The quality of the approximation depends on the choice of update formula and the quality of the initial approximation.

In summary, Quasi-Newton methods have the advantage of faster convergence and do not require an exact Hessian calculation, but can be limited by memory requirements and the accuracy of the Hessian approximation.

.------------------------------------------------------------------------------------------------------

Question:

The combination of Armijo and curvature conditions is known as

Answer: The combination of Armijo and curvature conditions is known as the Armijo-Goldstein condition. It is commonly used as a line search method in optimization algorithms, particularly in the context of nonlinear programming. The Armijo-Goldstein condition helps determine the step size or the amount of progress to be made in each iteration of an optimization algorithm. It ensures that the chosen step size satisfies both the sufficient decrease condition (Armijo condition) and the curvature condition (Goldstein condition) for the objective function being optimized.

Question:

0.8, 0.64, 0.512, 0.4096 ,.................... The fifth value in the sequence (the blank space) is

Answer: The pattern in the given sequence is that each term is obtained by multiplying the previous term by 0.8.

Using this pattern, we can calculate the fifth value:

0.8 \* 0.4096 = 0.32768

Therefore, the fifth value in the sequence is 0.32768.

Question: 0.8, 0.64, 0.512, 0.4096. ..... The sequence has ............ convergence

Answer:

The given sequence converges to 0.

We can observe that each term in the sequence is obtained by multiplying the previous term by 0.8. As the terms progress, they become smaller and smaller, approaching zero.

In mathematical terms, the sequence converges to zero as the number of terms increases. This means that as we continue the sequence indefinitely, the terms will approach and get arbitrarily close to zero.

Question: The need of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ condition is to ensure that the reduction of cost functions on every step is not too small.

Answer: The need of a suitable learning rate condition is to ensure that the reduction of cost functions on every step is not too small.

Question: Unlike mechanistic models, \_\_\_\_\_\_\_\_ can be built without strong theoretical understanding about the system we are studying.

Answer: Unlike mechanistic models, empirical models can be built without strong theoretical understanding about the system we are studying.

Question: \_\_\_\_\_\_ method is based on Newton Method but it modifies the Hessians in order to make them positive definite.

Answer: The trust region method is based on the Newton method but it modifies the Hessians in order to make them positive definite.

Question: is trust region method is based on Newton Method but it modifies the Hessians in order to make them positive definite.

Answer: Apologies for the confusion. The trust region method is indeed based on the Newton method, but it does not modify the Hessians to make them positive definite. Instead, it uses a trust region approach to restrict the search space within a specific region where a simplified model (usually a quadratic model) can be trusted to approximate the objective function. The trust region method iteratively updates the solution by solving a subproblem within the trust region, taking into account both the local quadratic model and the actual function evaluations. The modification of Hessians to ensure positive definiteness is often associated with the quasi-Newton methods, such as the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm.

Question: is quasi-Newton method is based on Newton Method but it modifies the Hessians in order to make them positive definite.

Answer: Yes, you are correct. The quasi-Newton method is indeed based on the Newton method, but it modifies the Hessians (approximations of the second derivatives) in order to make them positive definite. The Newton method typically requires the calculation and inversion of the Hessian matrix, which can be computationally expensive. The quasi-Newton methods approximate the Hessian matrix using updates based on the gradients of the objective function, allowing for a more efficient computation of the search direction.

One popular quasi-Newton method is the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm. It iteratively updates the approximation of the Hessian matrix while maintaining positive definiteness. The BFGS method avoids the need for explicitly calculating and inverting the Hessian matrix, making it more practical for large-scale optimization problems.