**最优化HW2编程报告**

1. **梯度下降法**

**1.1简要介绍**

梯度下降法是一种常用的优化算法，主要用于最小化或最大化目标函数。在机器学习和深度学习领域，梯度下降法被广泛应用于调整模型参数以使损失函数最小化。以下是梯度下降法的简要介绍和算法流程：

1. 目标函数（损失函数）： 首先，定义一个需要优化的目标函数（损失函数），通常表示为J(θ)，其中θ是模型的参数。这个目标函数衡量模型在给定数据上的性能，我们的目标是找到能够使目标函数最小化的参数θ。

2. 初始化参数： 随机或根据某种规则初始化模型参数θ。

3. 计算梯度：对目标函数关于参数θ的梯度进行计算。梯度表示目标函数在当前参数点的变化方向，是一个向量。使用链式法则计算梯度，通常表示为∇J(θ)。

4. 更新参数：根据梯度的反方向调整参数，以减小目标函数的值。更新规则通常是通过学习率（learning rate）来控制，即新的参数θ' = θ - α∇J(θ)，其中α是学习率。

1. 重复步骤3和步骤4： 不断迭代更新参数，直到满足停止条件，如达到最大迭代次数、目标函数的变化足够小等。

**1.2 算例介绍**

目标函数：f=100\*(x(1)^2-x(2))^2+(x(1)-1)^2;

初始点：x0=[-2,2]

步长计算方法：Armijo搜索

**1.3 实验结果**

|  |  |
| --- | --- |
| 最优步长搜索方法： | Armijo搜索 |
| 迭代次数： | 1608 |
| 最优解： | [9.999896663403633e-01;  9.999792988598010e-01] |
| 最小值： | 1.068996304380932e-10 |
| 消耗时间： | 2.765625000000000e+00 |

1. **BFGS算法实验结果**

**2.1简要介绍**

BFGS（Broyden-Fletcher-Goldfarb-Shanno）算法是一种常用的无约束非线性优化算法，用于最小化目标函数。它是拟牛顿法的一种，旨在寻找目标函数的极小值点。BFGS算法的名称来自于其发明者的姓氏。以下是BFGS算法的简要介绍和算法流程：

1. 初始化：选择初始点x₀，并初始化正定对称矩阵H₀，该矩阵近似于目标函数的二阶导数（Hessian矩阵）的逆。通常，初始的Hessian逆矩阵可以选择为单位矩阵。

2. 计算梯度： 计算目标函数在当前点x的梯度，表示为g(x)。如果梯度接近于零，则当前点可能是极小值。

3. 线搜索： 在当前点x₀沿着搜索方向s找到一个合适的步长α，以使目标函数减小。这一步通常使用线搜索方法，如Armijo规则或Wolfe条件。

4. 更新参数： 根据线搜索得到的步长α，更新当前点x₀，并计算新的梯度g(x)。

5. 更新拟牛顿矩阵：使用BFGS公式更新拟牛顿矩阵H。BFGS公式的更新规则确保在每一步中，Hessian矩阵的逆的近似保持正定性，从而提高了算法的稳定性和收敛性。

6. 收敛判断： 检查是否满足停止条件，例如梯度足够小或目标函数的变化足够小。如果满足条件，则算法收敛，否则返回步骤2。

7. 重复迭代： 不断迭代步骤2到步骤6，直到满足停止条件。

BFGS算法是一种迭代算法，通过不断更新当前点和拟牛顿矩阵，寻找目标函数的最小值。相较于批量梯度下降等算法，BFGS算法通常在每一步考虑更多的信息，因此收敛速度可能更快。然而，BFGS算法的计算代价较高，特别是在高维问题中。

**2.2算例介绍**

目标函数：f=100\*(x(1)^2-x(2))^2+(x(1)-1)^2;

初始点：x0=[-2,2]

步长计算方法：牛顿法迭代

**2.3 实验结果**

|  |  |
| --- | --- |
| 最优步长搜索方法： | 牛顿法迭代 |
| 迭代次数： | 19 |
| 最优解： | [1.000000199047205e+00;  1.000000196490727e+00] |
| 最小值： | 4.104025886610649e-12 |
| 消耗时间： | 3.593750000000000e-01 |

1. **信赖域（Dogleg）法实验结果**

**3.1简要介绍**

DogLeg法是信赖域算法中一种用于解决无约束非线性优化问题的方法，它主要用于在信赖域内找到一个步长，使得在该步长上的模型（通常是二次模型）与实际目标函数的改善程度较大。DogLeg法的名称源于步长的形状在图上类似狗腿的轮廓。以下是DogLeg法的简要介绍和算法流程：

1. 初始化： 选择初始点x₀，初始信赖域半径Δ₀，以及初始步长。通常，步长可以选择为0，即从初始点开始。
2. 计算梯度： 计算目标函数在当前点x的梯度，表示为g(x)。
3. 构建二次模型： 在当前点x附近构建一个二次模型，通常使用目标函数在当前点的梯度和Hessian矩阵（或其近似）。
4. 解决子问题： 在信赖域内，通过解决一个子问题来选择步长。DogLeg法中的子问题是一个二次规划问题，目标是在信赖域内找到一个步长，最大化在该步长上的模型改善程度。
5. 如果步长为0，则直接选择一个足够大的步长，以使模型改善程度增加。
6. 如果在信赖域边界上的步长可以显著改善模型，则选择边界上的步长。
7. 否则，通过解决二次规划问题来选择一个介于0和信赖域边界之间的步长，使得在该步长上的模型改善程度最大。
8. 更新参数： 根据所选择的步长更新当前点，并计算新的梯度g(x)。
9. 信赖域半径调整： 根据实际目标函数的改善程度和模型预测的改善程度，动态调整信赖域半径Δ。
10. 收敛判断： 检查是否满足停止条件，例如梯度足够小或目标函数的变化足够小。如果满足条件，则算法收敛，否则返回步骤2。
11. 重复迭代： 不断迭代步骤2到步骤7，直到满足停止条件。

DogLeg法在信赖域算法中是一种有效的步长选择方法，它通过在信赖域内灵活地选择不同类型的步长，结合了牛顿法的快速收敛和梯度下降法的全局搜索特性。

**3.2算例介绍**

目标函数：f=100\*(x(1)^2-x(2))^2+(x(1)-1)^2;

初始点：x0=[-2,2]

**3.3 实验结果**

|  |  |
| --- | --- |
| 迭代次数： | 33 |
| 最优解： | [9.999947241678511e-01;  9.999894329794273e-01] |
| 最小值： | 2.785807194532047e-11 |
| 消耗时间： | 2.187500000000000e-01 |

1. **共轭梯度法实验结果**

**4.1简要介绍**

共轭梯度法是一种用于求解对称正定线性方程组或最小化二次型目标函数的迭代优化算法。它通常用于大规模的线性系统或二次型优化问题，特别是在数值计算和机器学习中的一些应用。以下是共轭梯度法的简要介绍和算法流程：

1. 初始化： 选择一个初始点x₀、目标函数（二次型）的梯度g₀（初始残差），以及一个初始搜索方向p₀。
2. 循环迭代： 在每一次迭代中执行以下步骤：
3. 步长确定： 计算步长α，以最小化目标函数沿着搜索方向的值。对于二次型目标函数，这可以通过求解一维线性搜索问题来实现。
4. 更新参数： 使用步长α，更新当前点x₀：x₁ = x₀ + αp₀。
5. 计算梯度： 计算新点x₁处的梯度g₁。
6. 共轭方向： 计算新的搜索方向p₁，使得p₁与前一次的搜索方向p₀共轭。共轭梯度法的关键之一就是利用前一次的搜索方向，避免在新的搜索方向上浪费搜索。
7. 更新残差： 计算新的残差r₁，即负梯度的方向：r₁ = g₁ - Ap₁，其中A是线性方程组的系数矩阵。
8. 收敛判断： 检查是否满足停止条件，例如梯度足够小或残差足够小。如果满足条件，则算法收敛，否则继续迭代。
9. 更新参数和方向： 将当前点、残差和搜索方向更新为新的点、残差和搜索方向：x₀ = x₁，g₀ = g₁，p₀ = p₁。
10. 收敛或达到最大迭代次数： 当满足停止条件或达到最大迭代次数时，结束算法。

共轭梯度法的优势在于它能够在较少的迭代次数内收敛，特别适用于解决大规模的线性方程组或二次型优化问题。这是因为共轭梯度法充分利用了问题的特性，通过选择共轭的搜索方向，避免了不必要的搜索，从而提高了收敛速度**。**

**4.2算例介绍**

目标函数：f=100\*(x(1)^2-x(2))^2+(x(1)-1)^2;

初始点：x0=[-2,2]

步长计算方法：牛顿法迭代

Beta计算方法：FR;PR;HS

**4.3实验结果**

|  |  |
| --- | --- |
| beta采用方法 | FR |
| 迭代次数： | 4 |
| 最优解： | [9.999511276557218e-01;  9.999020620740683e-01] |
| 最小值： | 2.392332983793068e-09 |
| 消耗时间： | 6.093750000000000e-01 |
| beta采用方法 | PR |
| 迭代次数： | 6 |
| 最优解： | [1.000018129151053e+00;  1.000036265170940e+00] |
| 最小值： | 3.286703952975925e-10 |
| 消耗时间： | 4.531250000000000e-01 |
| beta采用方法 | HS |
| 迭代次数： | NAN |
| 最优解： | NAN |
| 最小值： | NAN |
| 消耗时间： | NAN |

1. **不同方法比较结果**
2. 对于Rosenbrock 函数这类二次型非正定的矩阵而言，传统的梯度下降法需要的搜索次数和花费时间都是最多的，其在部分区域迭代极慢。
3. BFGS由于使用迭代产生正定矩阵，有较好的数值稳定性，对于不同起始点的收敛效果都不错。
4. 信赖域方法可以处理非线性优化问题，但是信赖域算法的迭代次数与信赖域的大小选择有较强的相关性，且迭代过程中计算过程较为复杂。
5. 共轭梯度法中FR计算方式和PR计算方式的收敛速度相比于其他算法快很多，且计算过程较为简洁，但是HS法受到数值不稳定性的影响，在本次实验中迭代无法收敛。