人工智能程设

题目 Title : Matlab第一次大作业

院 系

School (Department) : 智能工程学院

专业

Major : 智能科学与技术

学生姓名

Student Name : 周德峰

学 号

Student No. : 21312210

指导教师(职称)

Supervisor （Title） : 王帅

时间： 2022 年 11 月 29 日

作业题目

$minf(x) = 2x_1^2 + 4x_2^2 - 6x_1 - 2x_1x_2$ 试采用MATLAB实现最速下降法求解该问题，给出具体的迭代过程、最终优化结果、涉及的代码，以及自己的心得体会。

实验原理

最速下降法是一种常见的无约束优化方法，其问题定义为：

Min $f(x), x\in R^n$，f 在 R^n 上连续可微。其步骤如下：

(1). 给出起始点 x_0，设定一个 $\varepsilon > 0$（如 0.01）作为终止条件，循环标号 $k = 0$；

(2). $p_k = -\nabla f(x_k)$，即找出目前搜索点的负梯度方向；

(3). $||p_k|| \le \varepsilon$，停止，取 $x_* \approx x_k$；否则转下一步；

(4). x_{k+1} 的更新规则为 $x_{k+1} = x_k + \alpha_k p_k$；这里，$\alpha_k$ 的定义为：

$$\alpha_k = -\frac{\nabla f(x_k)^T p_k}{(p_k)^T A p_k}$$，此处的 A 代表的是 $\nabla^2 f(x)$

实验思路

1，minf(X)，即最小化函数

2，使用syms确定自变量，方面后续求解相关雅可比矩阵和海塞矩阵

3，求出相关表达式，然后根据传入的初始值求出对应的值

4，设置循环和初始值，求出最优化结果

核心中间量求取

一阶雅可比矩阵

1，传统求解

即使用`diff`函数，`Y = diff(X,n,dim)`输入函数，阶数，求导变量，即可求得表达式

```
% dif1_x1=diff(fx,1,x1);        dif1_x1表示对x1变量求一阶导
% dif1_x2=diff(fx,1, x2);
% dif2_x11=diff(fx,2,x1);
% dif2_x22=diff(fx,2,x2);
% dif2_x12=diff(dif1_x1, 1,x2); dif2_x12表示在x1变量一阶导的基础上对x2求导
% dif2_x21=diff(dif1_x2, 1, x1);
% dif1_fx=[dif1_x1; dif1_x2];
% dif2_fx=[dif2_x11, dif2_x12; dif2_x21, dif2_x22]
```

2，使用内置函数jacobian求解

`J=(jacobian(f, x))'`可以直接求得雅可比矩阵

海塞矩阵

1，传统求解

同样，对一阶表达式再差分一次便可求得

2，内置函数求解

同样再次使用jacobian

或者使用matlab内置的hessian函数，可以直接求得海塞矩阵

α_k的求解

根据公式$\alpha_k=-\frac{\nabla f(x_k)^T p_k}{(p_k)^T A p_k}$，输入对应的表达式即可

`alpha=((fpk(x1,x2))')*fpk(x1,x2)/((fpk(x1,x2)')*H*fpk(x1,x2));`

式中H为海塞矩阵

求解

设置循环，当模值大于设置的阈值，则不断循环，小于时在自动退出

```
while norm(fpk(x1,x2))>kexi      %循环条件，如果模值大于设置的值，就一直循环
    %alpha=((fpk(x1,x2))')*fpk(x1,x2)/((fpk(x1,x2)')*dif2_fx*fpk(x1,x2));
    c=sprintf("第%d次优化结果如下", i);      %设置输出提示
    disp(c);
    alpha=((fpk(x1,x2))')*fpk(x1,x2)/((fpk(x1,x2)')*H*fpk(x1,x2));       %H为海塞矩阵
    x_real=x_real-alpha*fpk(x1,x2);      %对x优化
    [x1,x2]=deal(x_real(1,1), x_real(2,1))  %将优化结果赋值给x1，x2
    normx=norm(fpk(x1,x2))        %输出每一次的模值
    i=i+1;
end
```

结果

ε为0.01，x1=1，x2=0时，优化次数为6

最后优化结果为:

$x1=\frac{877}{512}$，约为1.71289

$x2=\frac{219}{512}$，约为0.4277

模值norm=$\frac{\sqrt{32768}}{32768}$

x1=2，x2=3时，优化次数为5

优化结果为：

$x_1=\frac{1487877}{867328}$ 约为1.71547

$x_2=\frac{372055}{867328}$ 约为0.4289

norm=$\frac{\sqrt{31213}\sqrt{1919025152}}{1919025152}$

实验结果截图

> 第5次优化结果如下
>
> x1 =
>
> $$\frac{219}{128}$$
>
> x2 =
>
> $$\frac{55}{128}$$
>
> normx =
>
> $$\frac{\sqrt{2048}}{2048}$$
>
> 第6次优化结果如下
>
> x1 =
>
> $$\frac{877}{512}$$
>
> x2 =
>
> $$\frac{219}{512}$$
>
> normx =
>
> $$\frac{\sqrt{32768}}{32768}$$
>
> v =
>
> $$\begin{pmatrix} \frac{877}{512} \\ \frac{219}{512} \end{pmatrix}$$

心得体会

1，在将具体值代入到抽象表达式时，使用matlab function

2，矩阵计算时，务必重视维度匹配

3，多个变量同时赋值，使用`[v1, v2]=deal(x1,x2)`， 中间间隔逗号

4，初始值和阈值不同，优化结果会有细小偏差，阈值越小，结果越精确