作业三

最优化方法

截止时间：11 月 27 日 23:59（周三晚）

请在规定时间前提交到大夏学堂，超时得分将有折扣。
计算、证明题提交 pdf 电子版，编程题提交 Python 代码、结果及必要的解释。

1 阅读任务

阅读以下阅读材料，并针对每部份简要探讨对该部份的认识。

- Chapter 2. Numerical Optimization by Jorge Nocedal and Stephen J. Wright, Springer, 2006.
- Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2(1), 183-202. (论文 pdf 见大夏学堂)

2 梯度下降收敛性分析 - 凸函数情形

在本问题中，我们将分析梯度下降在合适假设下的收敛性。考虑最小化可微凸函数 f, $\operatorname{dom} f = \mathbb{R}^n$, 其梯度为 L-Lipschitz 连续函数，即

$$\|\nabla f(x) - \nabla f(y)\|_2 \le L\|x - y\|_2, \quad \text{for all } x, y.$$

使用梯度下降，从起始点 $x^{(0)}$，更新步骤为

$$x^{(k)} = x^{(k-1)} - t_k \cdot \nabla f(x^{(k-1)}), \quad k = 1, 2, 3, \ldots,$$

其中每个 $t_k \le 1/L$。记更新点为 $x^+ = x - t\nabla f(x)$，其中 $t \le 1/L$。

假设 f 为凸函数，给定以下条件成立

$$f(x^+) \le f(x) - \frac{t}{2}\|\nabla f(x)\|_2^2. \tag{1}$$

注意到，根据此性质，我们知道梯度下降对于 $t \le 1/L$ 为下降方法（即每一步都会降低目标值）。

(a) 根据 (1)，利用凸函数 f 的一阶条件，证明

$$f(x^+) \le f^\star + \nabla f(x)^T(x - x^\star) - \frac{t}{2}\|\nabla f(x)\|_2^2.$$

(b) 根据以上结果，证明

$$f(x^+) \le f^\star + \frac{1}{2t}\big(\|x - x^\star\|_2^2 - \|x^+ - x^\star\|_2^2\big).$$

(c) 将以上结果在 $1,\ldots,k$ 取加和，得到

$$\sum_{i=1}^{k}(f(x^{(i)})-f^\star)\le\frac{1}{2t}\|x^{(0)}-x^\star\|_2^2.$$

(d) 利用梯度下降为一种下降方法的结论，证明以下

$$f(x^{(k)})-f^\star\le\frac{\|x^{(0)}-x^\star\|_2^2}{2tk},$$

即我们建立了所需的达到 ε-次最优 $O(1/\varepsilon)$ 的收敛速率。

3 近端算子的性质与实例

我们将考察近端算子的几个性质与实例。除非特殊说明，h 为凸函数，定义域 $\mathrm{dom}(h)=\mathbb{R}^n$ 并且 $t>0$ 任意，考虑其相关的近端算子

$$\mathrm{prox}_{h,t}(x)=\operatorname*{argmin}_z\ \frac{1}{2t}\|x-z\|_2^2+h(z).$$

(a) 证明 $\mathrm{prox}_{h,t}$ 是 $\mathbb{R}^n$ 上良定义的函数，即任意点 $x\in\mathbb{R}^n$ 有一个唯一的映射值 $\mathrm{prox}_{h,t}(x)$。

(b) 近端最小化算法（近端梯度下降的一个特例）的更新定义如下：

$$x^{(k+1)}=\mathrm{prox}_{h,t}(x^{(k)}),\quad k=1,2,3,\ldots.$$

写出该更新应用于 $h(x)=\frac{1}{2}x^TAx-b^Tx$ 的形式，其中 $A\in\mathbb{S}^n$。证明这与解决线性系统 $Ax=b$ 的 *迭代改进* 算法等价，即

$$x^{(k+1)}=x^{(k)}+(A+\epsilon I)^{-1}(b-Ax^{(k)}),\quad k=1,2,3,\ldots,$$

其中 $\epsilon>0$ 为常数。

4 实践：使用 Pytorch 编写牛顿法实现

参考牛顿法课件第 2 页、第 10 页（见大夏学堂），写出一般的牛顿法程序，用于求解无约束问题（提示：可根据作业二第 3 题的程序进行修改，其中 Hessian 矩阵可利用 torch.autograd.functional.hessian() 函数计算）。用你的程序求解以下问题，并与一阶算法比较：

(a) 课程讲义中的例子：$f(x)=(10x_1^2+x_2^2)/2+5\log(1+\exp(-x_1-x_2))$，初始点 $x_0=(20,20)^T$（调整目标函数 x_1,x_2 前的系数比较对一阶与二阶算法的影响）。

(b) 作业二第 3 题中的 Rosenbrock, Beale 函数优化问题。