近端梯度下降法（Proximal Gradient Descent）

背景介绍

在许多优化问题中，目标函数包含光滑项和非光滑项。例如，稀疏回归中的 ℓ_1 正则化项。传统的梯度下降法难以处理非光滑项，而近端梯度下降法通过结合梯度下降和近端算子的方法，有效解决了这一问题。这使得该方法在机器学习、信号处理等领域得到了广泛应用。

算法思想

近端梯度下降法旨在最小化如下形式的优化问题：

$$\min_{x \in \mathbb{R}^n} f(x) + g(x),$$

其中，$f(x)$ 是可微的凸函数，$g(x)$ 是凸但可能不可微的函数。该方法通过迭代地应用梯度下降步骤和近端算子，逐步逼近最优解。

定义 1 (近端梯度下降法)**.** 近端梯度下降法的迭代步骤为：

$$x_{k+1} = prox_{\alpha g}\left(x_k - \alpha \nabla f(x_k)\right),$$

其中，$\alpha > 0$ 是步长，$prox_{\alpha g}$ 是 g 的近端算子，定义为：

$$prox_{\alpha g}(v) = \arg\min_x \left(\frac{1}{2\alpha} \|x - v\|_2^2 + g(x) \right).$$

算法步骤

定义 2 (近端梯度下降算法). *1. 初始化 $x_0 \in \mathbb{R}^n$，选择步长 $\alpha > 0$。*

2. 对于每次迭代 $k = 0, 1, 2, \ldots$，执行：

(a) 计算梯度步：$v_k = x_k - \alpha \nabla f(x_k)$。

(b) 应用近端算子：$x_{k+1} = prox_{\alpha g}(v_k)$。

3. 重复步骤 2，直到满足停止条件。

理解与几何意义

理解

近端梯度下降法结合了梯度下降和近端算子的优势。每次迭代中，首先沿梯度方向进行一步下降，以减少光滑部分的目标函数值；然后通过近端算子调整步进结果，以处理非光滑项。这种方法不仅保持了优化方向的有效性，还能够引导解具备特定的结构性，如稀疏性。

例子

例子 1 (稀疏回归（LASSO）). 考虑目标函数：

$$\min_{x} \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1,$$

其中，$f(x) = \frac{1}{2}\|Ax-b\|_2^2$，$g(x) = \lambda\|x\|_1$。

近端梯度下降法的迭代步骤为：

$$x_{k+1} = soft\left(x_k - \alpha A^T(Ax_k - b), \alpha\lambda\right),$$

其中，$soft$ 为软阈值算子，定义为：

$$soft(v,\tau) = sign(v) \cdot \max\{|v| - \tau, 0\}.$$

收敛性

定理 1 (近端梯度下降法的收敛性). 假设 f 是 L-*Lipschitz* 可微的凸函数，且 g 是凸函数。当步长 α 满足 $0 < \alpha < \frac{1}{L}$ 时，序列 $\{x_k\}$ 收敛到最优解。

理解

由于 f 具有 L-Lipschitz 连续梯度，结合 g 的凸性，每一步迭代都确保目标函数值不增加，且序列 $\{x_k\}$ 逐步逼近最优解。这保证了算法的收敛性。

几何解释

理解

近端算子的几何意义:

近端算子 $\text{prox}_{\alpha g}(v)$ 可以看作是在点 v 处，对函数 g 加上一个二次罚项的最小化过程。这相当于在 v 附近寻找一个平衡点，使得 x 既接近 v，又在 $g(x)$ 的约束下尽可能优化。