1. Privacy Preserving Logistics Regression

1.1. Model: Logistic Regression

$$\hat{\boldsymbol{y}}(\boldsymbol{x}) = \operatorname{sigmoid}(\boldsymbol{x}\boldsymbol{w}^{\mathrm{T}} + \boldsymbol{b}) \Rightarrow \operatorname{predict}(\boldsymbol{x}) = \begin{cases} \hat{\boldsymbol{y}}(\boldsymbol{x}) \geq 0.5 : 1 \\ \hat{\boldsymbol{y}}(\boldsymbol{x}) < 0.5 : 0 \end{cases}$$

1.2. General DP-SGD

- Traditional SGD:
  1. Compute $\nabla L(\theta)$ on random sample
  2. Update $\theta_{\text{new}} = \theta - \eta \nabla L(\theta)$
- Differential Privacy SGD:
  1. Compute $\nabla L(\theta)$ on random sample
  2. Clip and add noise to $\nabla L(\theta)$
  3. Update $\theta_{\text{new}} = \theta - \eta(\nabla L(\theta))$

1.3. PPLR using DP-SGD

Algorithm 1: Algorithm 1: Diffential Privacy SGD (Outline)

Input:
- Examples $\{x_1, ..., x_N\}$
- Loss function $\mathcal{L} = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$
- Parameters:
  - Learning rate η_t
  - Noise scale σ
  - Group size L
  - Gradient Norm Bound C

Initialize θ_0 randomly
for $t \in [T]$ **do**
  Take a random sample L_t with sampling probability $\frac{L}{N}$
  Compute Gradient: For each $i \in L_t$, compute $\boldsymbol{g}_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$
  Clip Gradient: $\bar{\boldsymbol{g}}_t(x_i) \leftarrow \frac{\boldsymbol{g}_t(x_i)}{\max\left(1, \|\boldsymbol{g}_t(x_i)\|_2 / C\right)}$
  Add Noise: $\bar{\boldsymbol{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\boldsymbol{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \boldsymbol{I})\right)$
  Descent: $\theta_{t+1} \leftarrow \theta_t - \eta_t \bar{\boldsymbol{g}}_t$
Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

1.4. Requirements

- (20′) 填充实验代码中的缺失部分, 正确实现 DP-SGD 加噪机制
- (20′) 验证不同差分隐私预算下对于模型效果的影响
  - 需要根据 ε 和 δ 计算出对应的隐私预算;
  - 探究相同的总隐私消耗量下, 不同的迭代轮数对于模型效果的影响
- 实验报告: 说明代码实现方法, 给出不同参数设置下的实验评估结果, 格式为 PDF.

2. ElGamal Encryption

2.1. The Algorithm

2.1.1. Basics

- ElGamal 加密算法是一种公钥加密算法, 由 Taher ElGamal 于 1985 年提出. 它提供了一种保护通信隐私的方法, 允许数据在发送方使用接收方的公钥进行加密, 并由接收方使用其私钥进行解密
- ElGamal 算法的安全性基于计算离散对数的困难性
- Reference:
  - ElGamal, Taher. "A public key cryptosystem and a signature scheme based on discrete logarithms." IEEE transactions on information theory 31.4 (1985): 469-472.
  - https://en.wikipedia.org/wiki/ElGamal_encryption

2.1.2. Procedures

- Key Generation
  - 随机选择一个大素数 p 和原根 g, 其中 p 是 g 的生成元
  - 随机选择私钥 x, 满足 $0 < x < p - 1$
  - 计算公钥 $y = g^x \bmod p$

  公钥为 (p, g, y), 私钥为 x
- Encryption
  - 将明文消息 m 表示为一个位于 0 和 $p - 1$ 之间的整数
  - 随机选择一个临时私钥 k, 使得 $0 < k < p - 1$
  - 计算临时公钥 $c_1 = g^k \bmod p$
  - 计算临时密文 $c_2 = \left(y^k m\right) \bmod p$

  密文为 (c_1, c_2)
- Decryption
  - 利用私钥 x 计算 c_1 的模反演 $s = c_1^x \bmod p$
  - 计算明文消息 $m = \left(c_2 \cdot s^{-1}\right) \bmod p$, 其中 s^{-1} 是 s 的模逆元

2.1.3. Properties

随机性 ElGamal 加密中使用了随机值, 相同的明文在多次加密时产生不同的密文
乘法同态性 ElGamal 满足乘法同态性. 即两个密文的乘积解密后等于对应明文的乘积

2.2. Requirements

- (30′) 实现 `elgamal.py` 代码中缺失的部分函数, 保证加解密功能正确
  - 要求添加代码注释 (参考已有的注释部分)
  - 测试不同 `key_size` 设置下三个阶段的时间开销
  - 加解密的数据量可以酌情设置
- (15′) 验证 ElGamal 算法的随机性以及乘法同态性质, 对比乘法同态性质运算的时间开销, 即 `time(decrypt([a]*[b]))` 和 `time(decrypt([a])*decrypt([b]))`, 并给出原因说明.
- (Optional, 15′) 在大数据量的场景下, 优化 ElGamal 算法加解密的时间开销.
  - 可考虑的方案: 预计算、批量加密和解密、python 并行计算
  - 请给出方案说明以及方案有效性的证明
- 实验报告: 证明代码有效性以及完成题目要求即可, 格式为 PDF.