# 基础知识

联邦大模型的背景

## 大模型微调技术

高效参数微调PEFL(parameter-efficient finetuning techniques)

意思：训练大模型时只训练一小部分参数

主要方法：Lora(Low-Rank Adaptation), P-Tuning v2(Prompt Tuning V2)

优点：

1. 显存占用少
2. 训练速度快
3. 存储成本低
4. 精度和全量微调出于同一个水平相

### Lora

在大模型W边上新建两个梯形全连接矩阵A、B(旁路)

A的输入维度和W一样

B的输出维度和W一样

A的输出、B的输入比较小

通过更新AB的参数，取代更新W的参数

### P-Tuning V2

在prompt前面加上一些token，只更新token，保持原来的prompt不动

让token+prompt适配训练数据

## 联邦大模型现状

### 联邦大模型FLLM微调架构

架构：1个server + n个client

Server：梯度聚合和分发

流程：（单轮流程）

1. 先找好backbone，并且添加一些可以微调的部分，训练并调整可微调部分
2. 上传可微调部分的梯度到服务器
3. 服务端聚合（通过平均值等方式），下发新的梯度

重复上面的内容，让模型收敛。

问题：backbone不一定公开

### 安全联邦大模型SFLLM微调架构

**微调方式一：**

如果模型是黑盒，可以采用进化算法方式，在本地更新时不用梯

（零阶优化，两次前向代替反向）

给模型一个prompt，模型直接通过前向推理的logits更新参数

上面这一个方式的问题：

返回logits不是原始文本，不符合当前的一些接口

需要上传原始prompt到模型方，泄露prompt

**优化一下，方式二：**

场景：本地prompt不上传，backbone是黑盒

优化方式是，

把backbone压缩出一个可以替代的模型emulator，发给数据方

数据方在emulator上训练出一个adapter（参数微调），返回给模型方

模型方可以直接用优化的模型

优化后的问题：模型压缩不一定安全

## 可信执行环境TEE

一种通用的硬件环境，有运算、存储功能（黑盒环境）

TEE能提供安全性、完整性保护的独立环境

基本思想：分配一块特殊内存，计算敏感数据，外部硬件无法访问

流行TEE：Intel SGX/TDX, AMD SEV

# 存在的问题

训练阶段的安全问题：（数据泄露）

1. 恶意服务端偷取微调参数，推断客户端的数据
2. 恶意客户端从微调参数、前向embedding（一些特征）推测其他客户端的数据
3. 微调参数被各方持有，导致滥用

新场景：参数可控的分布式训练

安全要求：任何参与方无法拿到微调参数，保证每个参与方数据的安全性

# 算法模型

sf设计了两个算法达到参数可控的目的

## 小内存(消费级)TEE保护的LLM拆分

### 方法介绍

思想：借助tee硬件能力，把可微调的参数（通信参数）放入TEE内

1. 每个client和server都部署TEE，服务端TEE只用于参数聚合
2. client把微调参数放入TEE，大模型底座放在明文GPU，减少训练时间、内存消耗
3. TEE和明文环境通信时，用OTP(one time pad)加密。出TEE加密，进TEE解密

OTP思想：随机构建一个偏移量。在某个域上满足f(x+r)=f(x)+f(r), gpu计算f(x+r)，回到tee后减去f(r)

大模型的线性层放入GPU计算，非线性结构在TEE（softmax，layer norm）

### 局限性

不一定每个客户端都有tee，需要成本

Tee和明文交互导致IO开销大

OTP需要生成密钥，并且需要真随机数才能满足绝对安全，工程上落地有点困难

## 大内存TEE保护的LLM拆分

方法借鉴：和FL比较像的拆分学习（把模型拆分为多块）

（仅限公开backbone）

思想：只需要server有tee，只更新一部分模型

1. 大模型按层切成2部分，A为前N层，B为后M层，M<<N，A在客户端，B在服务器
2. Client只做前向（前N层为开源模型），最后一层embedding发送给服务器
3. 服务器在TEE内微调后M层

Client之间无法互相知道信息

模型微调：

1. 只微调后M层（每一层有4个大线性层：QKV/Dense/2个MLP Linear, MLP用lora代替进行更新）
2. 为了减少可微调参数，引入线性层稀疏化，不重要的head冻结（更新绝对值前25%大的节点）

联邦学习的效果比集中式差一点，这个架构把embedding放一起更新，和集中式差不多，好于FL；参数在server端控制下，保证了安全

# 实验结果

实验对比：

FL-LLM

SWMT（全部模型都在tee）

上面两个算法

实验结论：

1. 方法二在给定数据集的准确率高于其他方法，但是微调参数数量是方法一的5倍
2. 方法一准确率低于FL-LLM
3. SWMT耗时最长
4. 方法二训练时间短
5. FL-LLM延迟低，但是不能保证模型参数、数据的安全性