# 跨域推荐场景

案例：两个手机应用，媒体、电商，想要通过多方合作优化推荐效果，但是不希望泄露信息

拆分学习思想：

拆分网络结构，每个机构保留一部分网络

机构只在本地网络结构上进行前向、反向计算，然后把结果传给下一个设备

跨域推荐场景挑战：

1.模型设计？

模型有特征交叉、attention模块，如何拆分

2.通信瓶颈？

推荐模型更新频率高，数据量大

3.安全性保证？

梯度可能泄露信息

# 跨域推荐全链路解决方案

## 模型设计

模型拆分比DNN麻烦一点

以DeepFM为案例，要求在FM layer做特征的两两交叉

FM层有两部分组成：

一阶和（来源是下面一层的Dense Embeddings）

二阶和（来源是再下一层的features）

把FM的计算流程拆分出两块。接收方收到两块的结果后，就可以计算最终的loss

## 通信瓶颈

一般通信流程：

1特征提供方计算前向

2隐层传入label方，继续前向

3计算loss，梯度和反向传播

4梯度传给没label方

5最后的反向传播

以上流程的问题：每个batch需要通信两次，导致通信量、次数都很大

优化：

1. 带宽：量化、稀疏化减少单次通信量
2. 延迟：减少通信次数，SLStateAsync, SLAsync。训练过程中每一轮都多次迭代
3. 并行：计算和通信并行，pipeline

流水线并行案例：

Server - - 3 3 3 3 - -

Upload - 2 2 2 2 - - -

Download - - - 4 4 4 4 -

Client 1 1 1 1 5 5 5 5

问题：模型更新在第二个batch开始时（上面最下面一层的第二个1）还没更新完

为了让模型进行完整的更新，需要缓存每个batch的参数

用缓存的参数计算梯度，更新现在的参数

流水线并行度是2-4时比较合适

## 安全性保证

跨域推荐的攻击场景主要关心这两种攻击方式：

1. RIA重构攻击，获取某一方的原始特征

根据本地模型、隐层数据、对方模型结构、数据集进行推测

1. LIA推测label

具体攻击手段主要是隐层梯度、模型数据集。

需要评估攻击方法的有效性、性能指标

从而针对性对模型进行加固

# SF中试用跨域推荐

## 官方API、文档

模型：secretflow.ml.nn.applications

有DeepFM, BST, MMOE等

通信：

secretflow.ml.nn.sl.backend.tensorflow.strategy (减少通信次数、流水线并行)

secretflow.utils.compressor (稀疏化和量化)

安全：

secretflow.ml.nn.sl.attack (攻防算法)

secretflow.security.privacy (dp)

SplitRec文档：

[SplitRec：当拆分学习遇上推荐系统 | SecretFlow v1.8.0b0 | 隐语 SecretFlow](https://www.secretflow.org.cn/zh-CN/docs/secretflow/v1.8.0b0/user_guide/federated_learning/vertical_federated_learning/SplitRec)

文字版：

<https://www.secretflow.org.cn/zh-CN/docs/secretflow/v1.8.0b0/user_guide/federated_learning/vertical_federated_learning/SplitRec>

## 代码中的应用

SLModel类的参数：

1. dp\_strategy\_dict:

加各类dp，embedding和label都可以加dp

1. base\_model\_dict, model\_fuse:

使用自定义、预定义的模型

1. compressor:

压缩，比如TopKSparse稀疏化

1. strategy:

策略，如pipeline

然后通过model.fit进行训练