联邦学习

合肥工業大學

苏伊阳

2021.12.9

目录

概念

联邦学习

- 本质：联邦学习本质上是一种**分布式**机器学习技术，或机器学习框架。
- 目标：联邦学习的目标是在保证**数据隐私安全及合法合规**的基础上，实现共同建模，**提升AI模型的效果。**

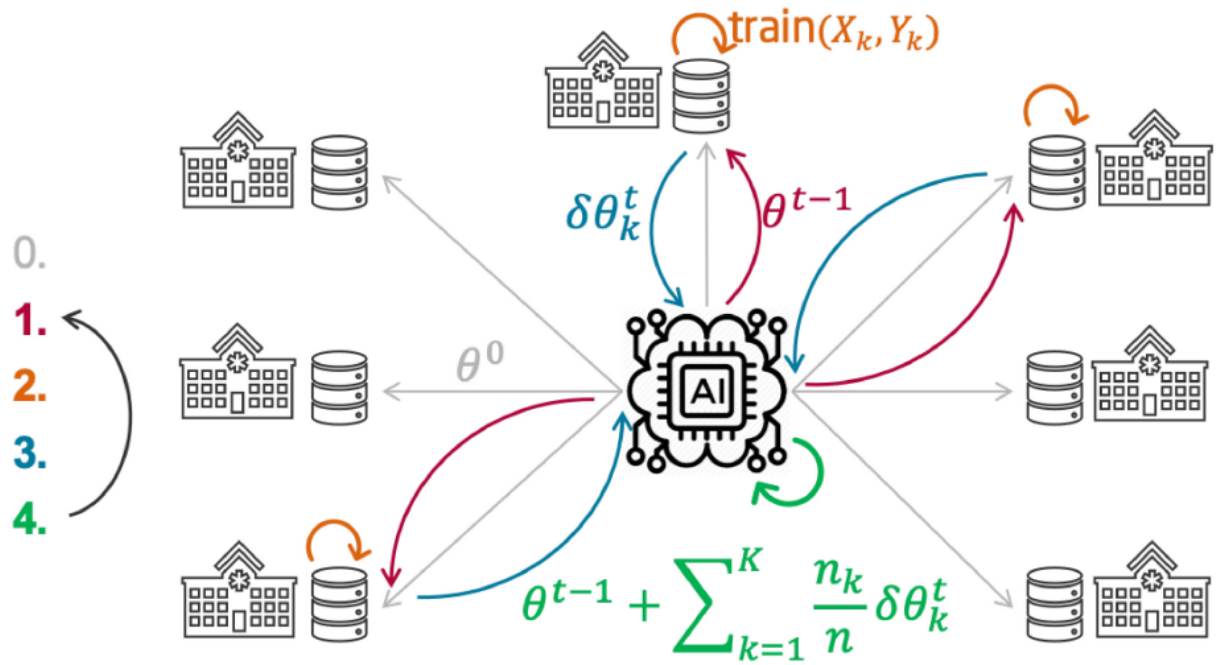


Fig. 1. Overall training process for federated learning. The initial model is distributed (0). Per global epoch, some clients are selected and receive the current parameter values (1). The selected clients update locally (2). The local updates are sent back to the server (3). The server aggregates all received local updates (4). Steps 1 through 4 are repeated until convergence.

概念

数据问题

- **大规模分布**:数据分散在世界各地，被大量客户端（医院等）持有

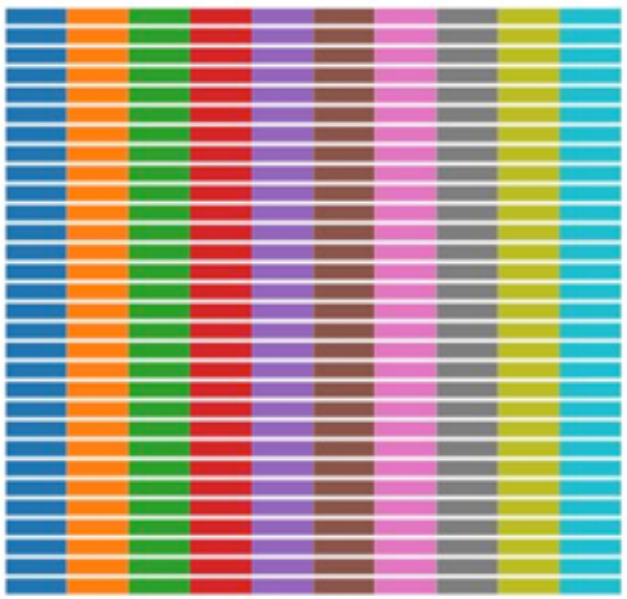
  例如：智能手机上收集的传感器数据用于医疗

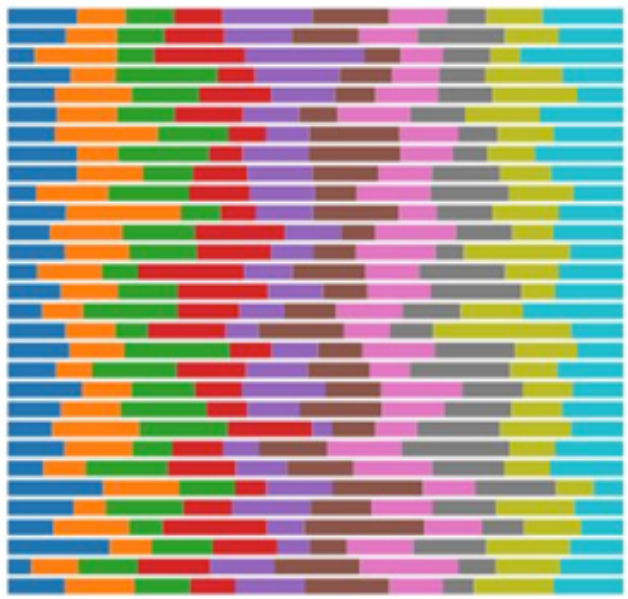
- **非独立同分布**:不同参与用户的数据不是独立的、相同分布

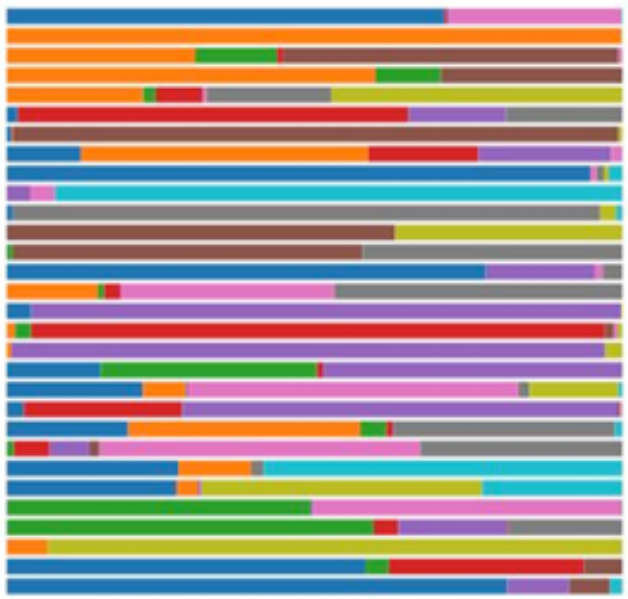
  例如：医疗数据大都不是独立同分布的

- **不平衡**:有些用户可能有很多数据样本，而有些用户可能只有一点点数据样本

  例如：医院有大型医院、小医院，数据规模不平衡



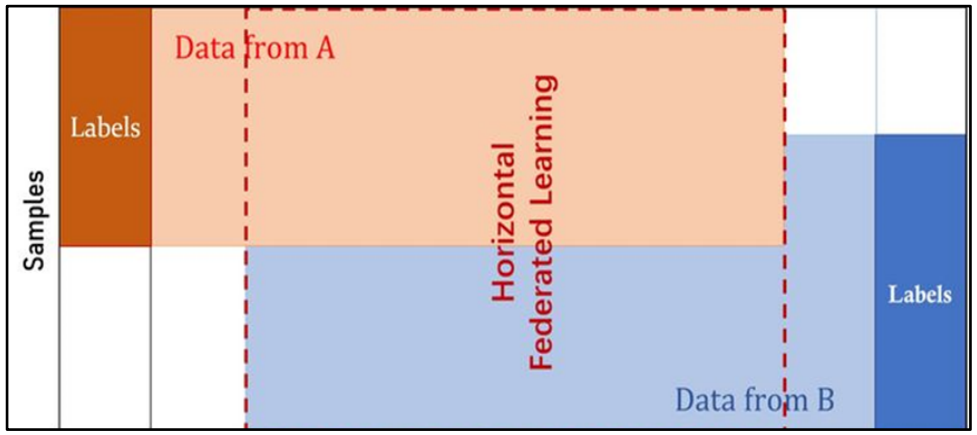


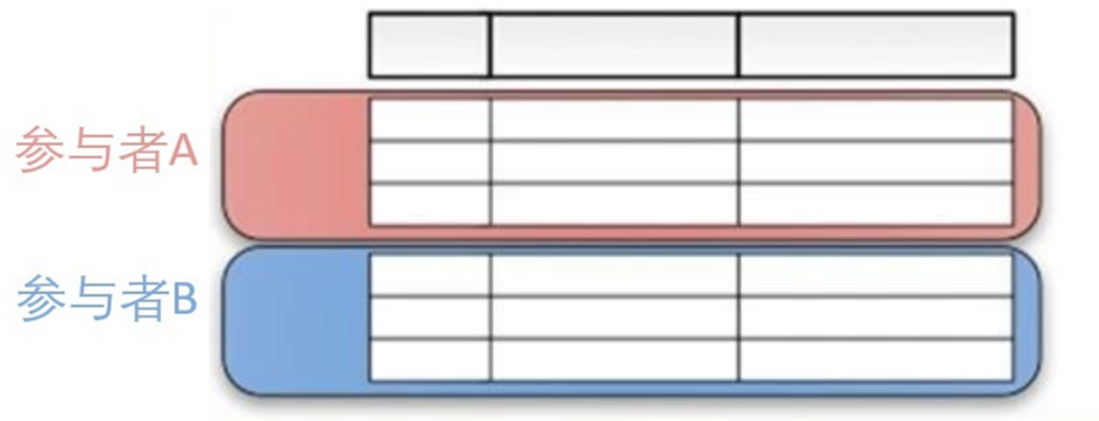


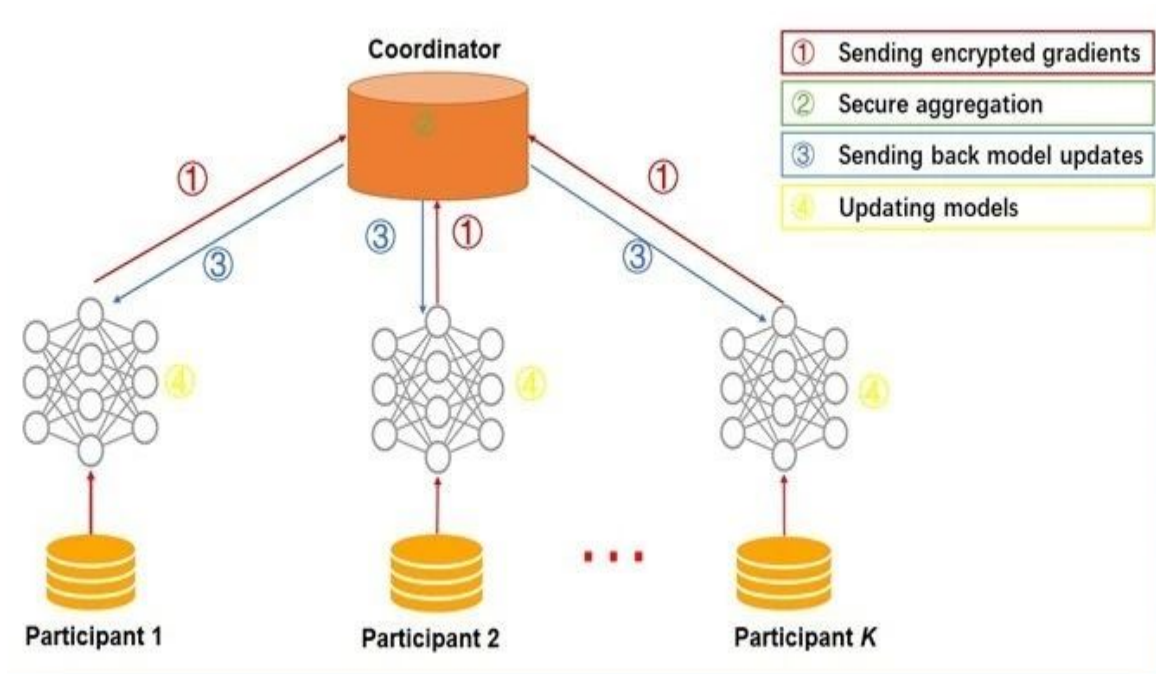
联邦学习分类

横向联邦学习

两个数据集的用户特征 (X1, X2, …) 重叠部分较大，而用户 (U1, U2, …) 重叠部分较小







- 参与方各自从服务器下载最新模型；
- 加密梯度上传给服务器，服务器聚合各用户的梯度更新模型参数；
- 服务器返回更新后的模型给各参与方；
- 各参与方更新各自模型。

联邦学习分类

横向联邦学习-FederatedAveraging

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
$m \leftarrow \max(C \cdot K, 1)$
$S_t \leftarrow$ (random set of m clients)
for each client $k \in S_t$ **in parallel do**
$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
for batch $b \in \mathcal{B}$ **do**
$w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

C是随机分数，用来随机挑选客户端的数量

K是总共的客户端数量

$$\min_{w \in R^d} f(w) \qquad f(w) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

$$f_i(w) = l(x_i, y_i; w)$$

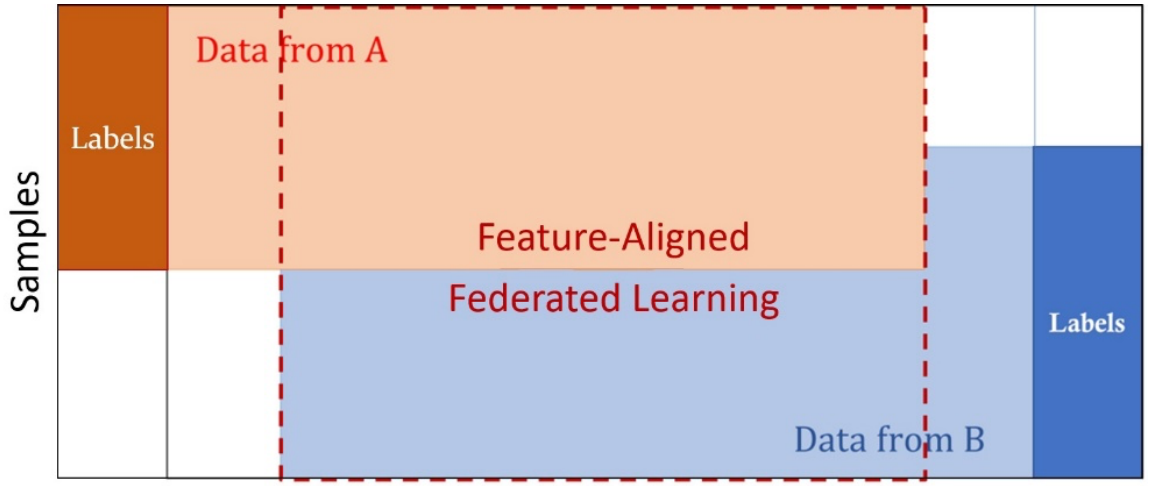
$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \quad F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w)$$

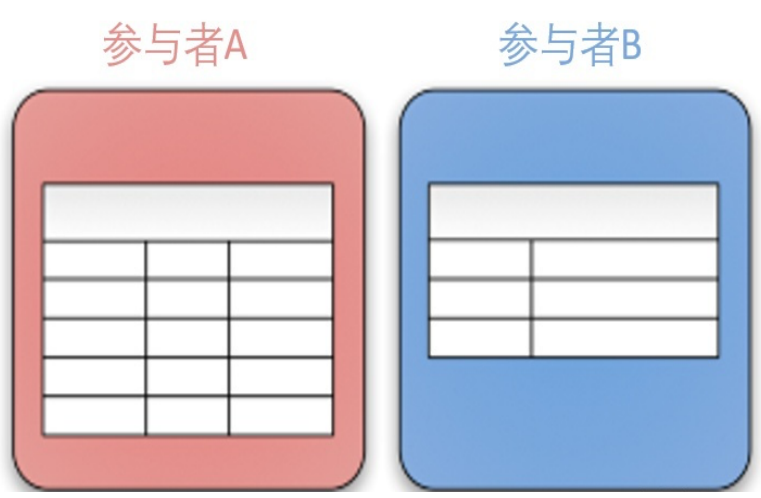
联邦学习分类

纵向联邦学习

两个数据集的用户(U1，U2，...)重叠部分较大，而用户特征(X1，X2，...)重叠部分较小；

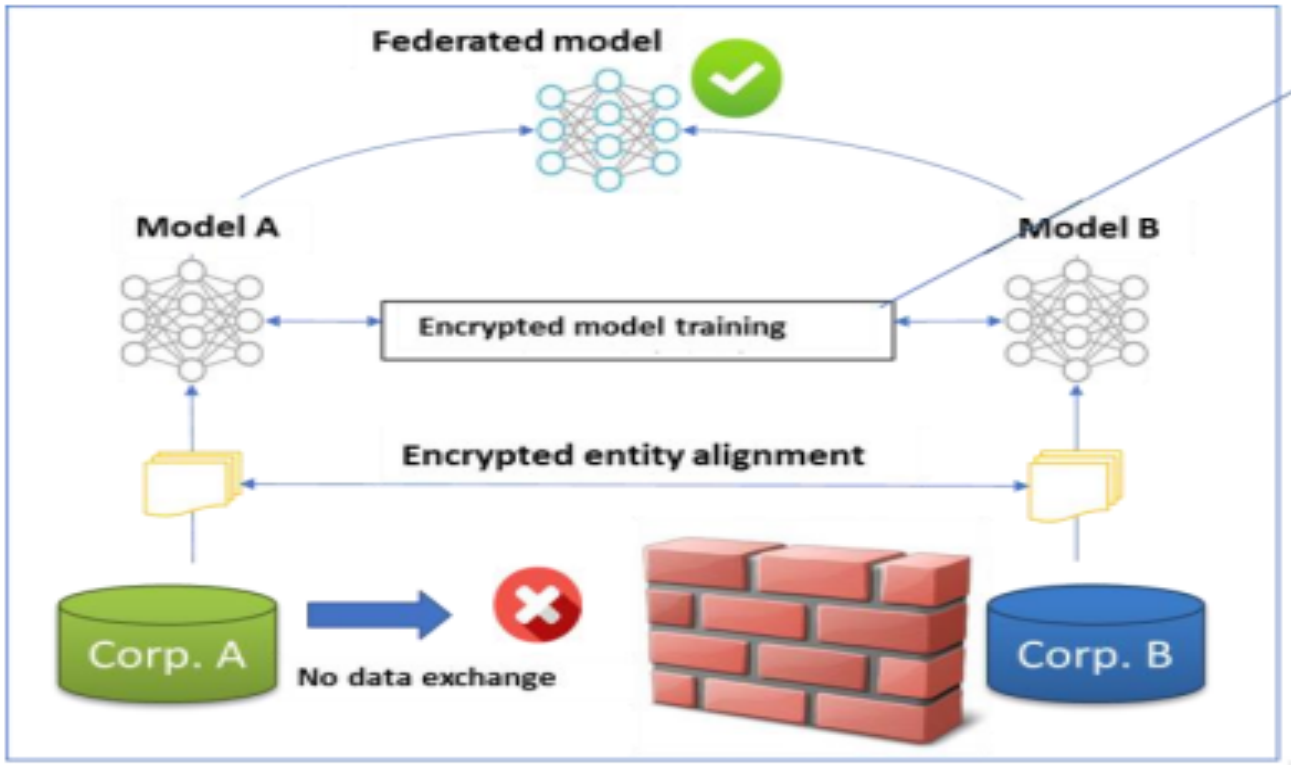
只有一方有标签数据Y

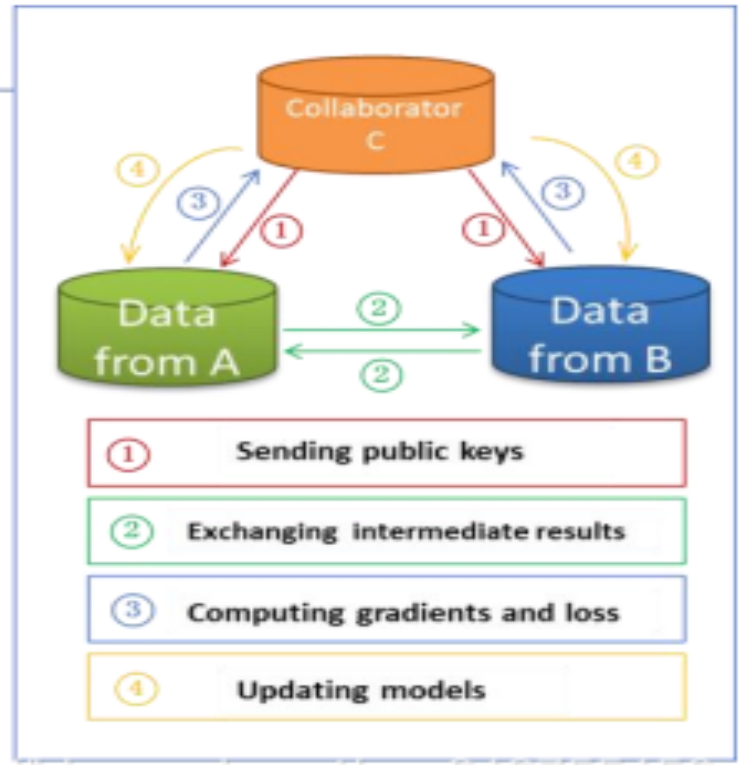




联邦学习分类

纵向联邦学习





第一步：加密样本对齐，不会暴露非交叉用户

第二步：对齐样本进行模型加密训练：

- 由第三方C向A和B发送公钥，用来加密需要传输的数据；
- A和B分别计算和自己相关的特征中间结果，并加密交互，用来求得各自梯度和损失；
- A和B分别计算各自加密后的梯度并添加掩码发送给C，同时B计算加密后的损失发送给C；
- C解密梯度和损失后回传给A和B，A、B去除掩码并更新模型。

联邦学习分类

纵向联邦学习

基于RSA和哈希算法的解决方案

- Common input: $n, e, H(), H'()$
- $H()$ is a Full-Domain Hash $H : \{0,1\}^* \rightarrow \mathbb{Z}_n^*$
- Client's input: $\mathcal{C} = \{hc_1, \cdots, hc_v\}$, where: $hc_i = H(c_i)$
- Server's input: $d, \mathcal{S} = \{hs_1, \cdots, hs_w\}$, where: $hs_j = H(s_j)$

OFF-LINE:

1. Server:
   $\forall j$, compute: $K_{s:j} = (hs_j)^d \bmod n$ and $t_j = H'(K_{s:j})$
2. Client:
   $\forall\, i$, compute: $R_{c:i} \leftarrow \mathbb{Z}_n^*$ and $y_i = hc_i \cdot (R_{c:i})^e \bmod n$

ON-LINE:

3. Client $\longrightarrow$ Server: $\{y_1, .., y_v\}$
4. Server:
   $\forall\, i$, compute: $y'_i = (y_i)^d \bmod n$
5. Server $\longrightarrow$ Client: $\{y'_1, ..., y'_v\},\ \{t_1, .., t_w\}$
6. Client:
   $\forall i$, compute: $K_{c:i} = y'_i / R_{c:i}$ and $t'_i = H'(K_{c:i})$
   OUTPUT: $\{t'_1, .., t'_v\} \cap \{t_1, .., t_w\}$

Blind RSA-based PSI Protocol with linear complexity

$\{c_1, c_2, c_3, \cdots, c_v\}$客户端A的ID集合

$\{s_1, s_2, s_3, \cdots, s_w\}$客户端B的ID集合

(n, e)公钥(n, d)私钥

$R_{c:i}$客户端A产生的随机数

[1]De Cristofaro E, Tsudik G. Practical Private Set Intersection Protocols with Linear Computational and Bandwidth Complexity[J]. IACR Cryptol. ePrint Arch., 2009, 2009: 491.
[2]De Cristofaro E, Tsudik G. On the performance of certain private set intersection protocols[J]. IACR, 2012: 54.

联邦学习分类

纵向联邦学习-线性回归为例

	party A	party B	party C
step 1	initialize Θ_A	initialize Θ_B	create an encryption key pair, send public key to A and B;
step 2	compute $[[u_i^A]],[[\mathcal{L}_A]]$ and send to B;	compute $[[u_i^B]],[[d_i^B]],[[\mathcal{L}]]$, send $[[d_i^B]]$ to A, send $[[\mathcal{L}]]$ to C;	
step 3	initialize R_A, compute $[[\frac{\partial \mathcal{L}}{\partial \Theta_A}]]+[[R_A]]$ and send to C;	initialize R_B, compute $[[\frac{\partial \mathcal{L}}{\partial \Theta_B}]]+[[R_B]]$ and send to C;	C decrypt $\mathcal{L}$, send $\frac{\partial \mathcal{L}}{\partial \Theta_A}+R_A$ to A, $\frac{\partial \mathcal{L}}{\partial \Theta_B}+R_B$ to B;
step 4	update Θ_A	update Θ_B	
what is obtained	Θ_A	Θ_B	

目标函数$\min_{\Theta_A,\Theta_B}\Sigma_i \| \Theta_A x_i^A + \Theta_B x_i^B - y_i \|^2 + \frac{\lambda}{2}(\| \Theta_A \|^2 + \| \Theta_B \|^2)$

$$u_i^A = \Theta_A x_i^A, u_i^B = \Theta_B x_i^B \qquad [[L]] = \left[\left[\Sigma_i \left(\left(u_i^A + u_i^B - y_i\right)\right)^2 + \frac{\lambda}{2}(\| \Theta_A \|^2 + \| \Theta_B \|^2)\right]\right]$$

$$[[L_A]] = \left[\left[\Sigma_i \left(u_i^A\right)^2 + \frac{\lambda}{2}\| \Theta_A \|^2\right]\right], [[L_B]] = \left[\left[\Sigma_i \left(\left(u_i^B - y_i\right)^2\right) + \frac{\lambda}{2}\| \Theta_B \|^2\right]\right] \qquad [[L_{AB}]] = 2\Sigma_i \left(\left[\left[u_i^A\right]\right]\left(u_i^B - y_i\right)\right)$$

$$[[L]] = [[L_A]] + [[L_B]] + [[L_{AB}]] \qquad [[d_i]] = \left[\left[u_i^A\right]\right] + \left[\left[u_i^B - y_i\right]\right]$$

$$\left[\left[\frac{\partial \mathcal{L}}{\partial \Theta_A}\right]\right] = \Sigma_i [[d_i]] x_i^A + [[\lambda \Theta_A]] \qquad \left[\left[\frac{\partial \mathcal{L}}{\partial \Theta_B}\right]\right] = \Sigma_i [[d_i]] x_i^B + [[\lambda \Theta_B]]$$

个人想法

联邦学习

联邦学习 = 分布式计算 + 数据加密技术，提升模型效果的同时保护隐私安全

医学图像分类、疾病诊断

开源框架	FATE	TensorFlow Federated	PaddleFL	Pysyft
受众定位	工业产品/学术研究	学术研究	学术研究	学术研究
牵头公司/机构	微众银行	Google	百度	OpenMined
联邦学习类型	横向联邦学习 纵向联邦学习 联邦迁移学习	横向联邦学习	横向联邦学习 纵向联邦学习	横向联邦学习
联邦特征工程算法	特征分箱 特征选择 特征相关性分析 支持	不支持	不支持	不支持
机器学习算法	LR，GBDT，DNN等	LR，DNN等	LR，DNN等	LR，DNN等
安全协议	同态加密，SecretShare，RSA，DiffieHellman	DP	DP	同态加密，SecretShare
联邦在线推理	支持	不支持	不支持	不支持
Kubernetes	支持	不支持	不支持	不支持
代码托管平台	Github(https://github.com/FederatedAI/FATE)	Github(https://github.com/tensorflow/federated)	Github(https://github.com/PaddlePaddle/PaddleFL)	Github(https://github.com/OpenMined/PySyft)

使用Docker Compose 部署 FATE

准备工作

1. 两个主机（物理机或者虚拟机，都是Centos7系统）；
2. 所有主机安装Docker 版本 : 18+；
3. 所有主机安装Docker-Compose 版本: 1.24+；
4. 部署机可以联网，所以主机相互之间可以网络互通；
5. 运行机已经下载FATE 的各组件镜像（离线构建镜像参考文档构建镜像）。

```
[root@pretend docker-deploy]# docker images
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
federatedai/python-nn         1.6.1-release       ba1f043eaf41        3 weeks ago         4.39GB
federatedai/eggroll           1.6.1-release       ac61d8403e25        3 weeks ago         2.19GB
federatedai/fateboard         1.6.1-release       43fc3a239cd6        3 weeks ago         205MB
federatedai/python            1.6.1-release       afcae32c1f28        3 weeks ago         2.08GB
federatedai/base-image        1.6.1-release       7842b57140ca        4 weeks ago         1.81GB
mysql                         8                   ecac195d15af        7 weeks ago         516MB
redis                         5                   02fee89f17ad        8 weeks ago         110MB
federatedai/client            1.6.1-release       1adc06d207fb        3 months ago        5.4GB
federatedai/serving-server    2.0.4-release       4bf5f2ad9fc5        7 months ago        234MB
federatedai/serving-proxy     2.0.4-release       1b63abead29d        7 months ago        266MB
maven                         3.6-jdk-8           d1b3f61d61f2        8 months ago        525MB
centos/python-36-centos7      latest              602660fa9b4e        14 months ago       650MB
mcr.microsoft.com/java/jre    8u192-zulu-alpine   73f726f40401        3 years ago         143MB
[root@pretend docker-deploy]#
```

镜像

```
root@192.168.75.132's password:
Authentication failed.
[root@pretend docker-deploy]# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED         S
TATUS         PORTS
              NAMES
764c1a653cad   redis:5                              "docker-entrypoint.s…"   6 minutes ago   U
p 6 minutes   6379/tcp
              serving-10000_redis_1
9c3ff1d57020   federatedai/serving-proxy:2.0.4-release   "/bin/sh -c 'java -D…"   6 minutes ago   U
p 6 minutes   0.0.0.0:8059->8059/tcp, :::8059->8059/tcp, 0.0.0.0:8869->8869/tcp, :::8869->8869/tcp,
8879/tcp   serving-10000_serving-proxy_1
03041fb29ad1   federatedai/serving-server:2.0.4-release   "/bin/sh -c 'java -c…"   6 minutes ago   U
p 6 minutes   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp
              serving-10000_serving-server_1
90c4b3fc0393   federatedai/fateboard:1.6.1-release       "/bin/sh -c 'java -D…"   6 minutes ago   U
p 6 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
              confs-10000_fateboard_1
04acdaf5f941   federatedai/client:1.6.1-release          "/bin/sh -c 'flow in…"   6 minutes ago   U
p 6 minutes   0.0.0.0:20000->20000/tcp, :::20000->20000/tcp
              confs-10000_client_1
0a7af25eded1   federatedai/python-nn:1.6.1-release       "container-entrypoin…"   6 minutes ago   U
p 6 minutes   0.0.0.0:9360->9360/tcp, :::9360->9360/tcp, 8080/tcp, 0.0.0.0:9380->9380/tcp, :::9380->
9380/tcp   confs-10000_python_1
91cd8979be03   federatedai/eggroll:1.6.1-release         "/tini -- bash -c 'j…"   6 minutes ago   U
p 6 minutes   4671/tcp, 8080/tcp
              confs-10000_nodemanager_1
508fc8b85ff0   federatedai/eggroll:1.6.1-release         "/tini -- bash -c 'j…"   6 minutes ago   U
p 6 minutes   8080/tcp, 0.0.0.0:9370->9370/tcp, :::9370->9370/tcp
              confs-10000_rollsite_1
bc3c54e94995   mysql:8                                  "docker-entrypoint.s…"   6 minutes ago   U
p 6 minutes   3306/tcp, 33060/tcp
              confs-10000_mysql_1
5037c92e594d   federatedai/eggroll:1.6.1-release         "/tini -- bash -c 'j…"   6 minutes ago   U
p 6 minutes   4670/tcp, 8080/tcp
              confs-10000_clustermanager_1
[root@pretend docker-deploy]#
```

部署成功

感谢聆听！

合肥工業大學