Security algorithms under Vertical Federated learning in the FATE framework

FATE Framework

FATE (Federated AI Technology Enabler) 是微众银行(*WeBank*) AI部门发起的全球首个联邦学习工业级开源框架，可以让企业和机构在保护数据安全和数据隐私的前提下进行数据协作。FATE项目使用安全多方计算 (MPC) 以及同态加密 (HE) 技术构建底层安全计算协议，以此支持不同种类的机器学习的安全计算，包括逻辑回归、基于树的算法、深度学习和迁移学习等。

设计原则

- 支持多种主流算法：为机器学习、深度学习、迁移学习提供高性能联邦学习机制。
- 支持多种多方安全计算协议：同态加密(HE)、秘密共享(SS)、哈希散列(Hash)等。
- 支持友好的跨域信息管理方案，解决了联邦学习的信息安全审计难的问题。

首次发布

2019年1月份，FATE开源，目前1.11.0版本。

https://github.com/WeBankFinTech/FATE

Horizontal Federated Learning

横向联邦学习也称为特征对齐的联邦学习(Feature-Aligned Federated Learning)，即横向联邦学习参与者的数据特征是对齐的。

特点：样本空间重叠部分小，特征空间重叠部分大。

举例：微众和合作行共建反洗钱模型，期望优化反洗钱模型

◆ 设定：

✓ Y 表示“是否存在洗钱行为”

✓ 合作行和微众都有（X,Y）

✓ 双方不暴露自己的（X,Y）

◆ 传统建模方法问题：

✓ 微众和合作行各自样本不够多

◆ 期望结果：

✓ 保护隐私条件下，建立联合模型

✓ 联合模型效果超过单边数据建模

微众银行

ID 证件号 电话号	X1 资金来源和经营范围不符笔数	X2 大额交易笔数	Y 表现数据
U1	5	15	有
U2	8	20	有
U3	0	5	无
U4	0	0	无
U5	2	1	无
U6	50	50	有
U7	60	6	有

业务系统A 数据

合作行

ID 证件号 电话号	X1 资金来源和经营范围不符笔数	X2 大额交易笔数	Y 表现数据
U8	5	10	有
U9	10	2	有
U10	2	30	有
U11	0	10	有
U12	8	7	有

业务系统B 数据

Vertical Federated Learning

纵向联邦学习也称为样本对齐的联邦学习(Sample-Aligned Federated Learning)，即纵向联邦学习的样本是对齐的。

特点：特征空间重叠部分小，样本空间重叠部分大。

举例：微众与合作企业联合建模，微众有Y（业务表现），期望优化本方的Y预测模型

◆ 设定：

✓ 只有微众拥有 Y= “逾期表现”

✓ 合作企业无法暴露含有隐私的 X

◆ 传统建模方法问题：

✓ 合作企业缺乏Y无法独立建模

✓ X数据全量传输到微众不可行

◆ 期望结果：

✓ 保护隐私条件下，建立联合模型

✓ 联合模型效果超过单边数据建模

合作企业

ID 证件号 电话号	X1 帐龄	X2 月薪	X3 等级
U1	9	8000	A
U2	4	5000	C
U3	2	3500	C
U4	10	10000	A
U5	5	7500	B
U6	5	7500	A
U7	8	8000	B

业务系统A 数据

微众银行

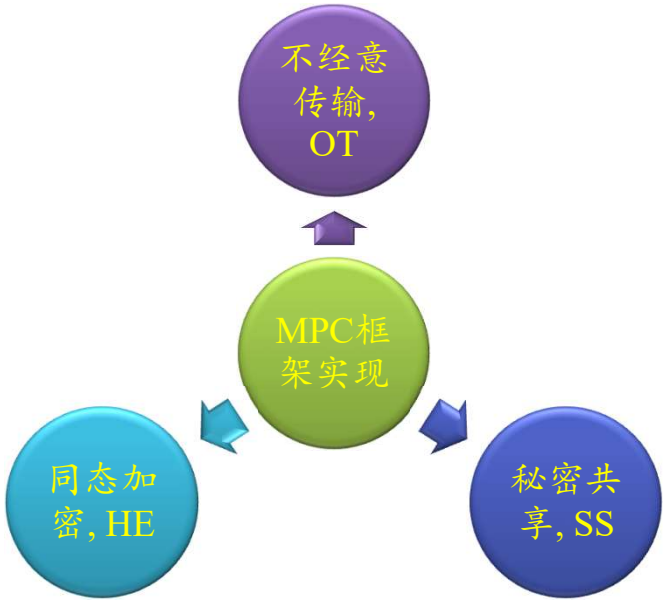
ID 证件号 电话号	X4 央行征信分	X5 微众 内部分	Y 表现数据
U1	600	600	无
U2	550	500	有
U3	520	500	有
U4	600	600	无
U8	600	600	无
U9	520	500	有
U10	600	600	无

业务系统B 数据

MPC

安全多方计算(Secure Muti-party Computation, 简称MPC, SMC或SMPC)由图领奖获得者姚期智教授于1982年提出，也就是经典的百万富翁问题：两个争强好胜的富翁Alice和Bob在街头相遇，如何在不暴露各自财富的前提下比较出谁更富有?

数学描述为“有n个参与者$p_i = \{p_1, p_2, \ldots\ldots, p_n\}$，要以一种安全的方式共同计算一个结果。具体地讲，每个参与者p_i拥有自己的隐私信息x_i，n个参与者要通过一个函数$f_{unc}(x_1, x_2, \ldots\ldots, x_n) = (y_1, y_2, \ldots\ldots, y_n)$进行共同计算。最终，每个参与者只能通过自身信息输入x_i获得y_i，而不能了解其他方的输入与输出信息。”



Secret Sharing(SS)

算数秘密共享(Arithmetic Secret Sharing)

1.加法秘密共享

Alice拥有信息 x 和 Bob拥有信息 y，目标共同计算$z = x + y$

Alice

$\langle x \rangle = \langle x_1 \rangle + \langle x_2 \rangle$

Bob

$\langle y \rangle = \langle y_1 \rangle + \langle y_2 \rangle$

可信计算节点(s_1 , s_2)辅助

Alice将$\langle x_1 \rangle$给到 s_1，同时将$\langle x_2 \rangle$给到 s_2

Bob 将 $\langle y_1 \rangle$给到 s_1，同时将$\langle y_2 \rangle$给到 s_2

s_1

$\langle x_1 \rangle$ $\langle y_1 \rangle$

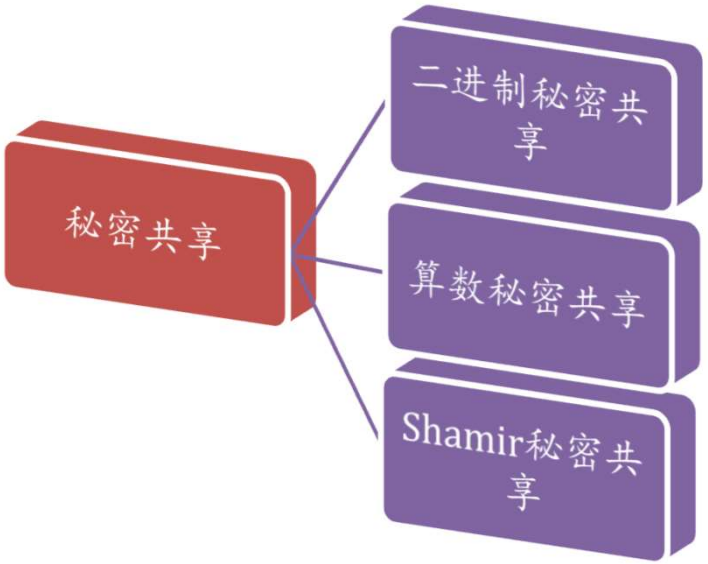
$\langle z_1 \rangle = \langle x_1 \rangle + \langle y_1 \rangle$

s_2

$\langle x_2 \rangle$ $\langle y_2 \rangle$

$\langle z_2 \rangle = \langle x_2 \rangle + \langle y_2 \rangle$

$z = \langle z_1 \rangle + \langle z_2 \rangle = x + y$



Secret Sharing(SS)

2. 乘法秘密共享

Alice拥有信息 x 和 Bob拥有信息 y，目标共同计算 $z = x \times y$

可信方M提供三元组满足$\{a, b, c | ab = c\}$

Alice 将$\langle x_1 \rangle$给到s_1，同时将$\langle x_2 \rangle$给到s_2

Bob 将 $\langle y_1 \rangle$给到s_1，同时将$\langle y_2 \rangle$给到s_2

M将$\langle a_1 \rangle, \langle b_1 \rangle$给到$s_1$，同时将$\langle a_2 \rangle, \langle b_2 \rangle$给到$s_2$

s_1	s_2	
$\langle d_1 \rangle = \langle x_1 \rangle - \langle a_1 \rangle$	$\langle d_2 \rangle = \langle x_2 \rangle - \langle a_2 \rangle$	//利用a, b 切片对x, y盲化
$\langle e_1 \rangle = \langle y_1 \rangle - \langle b_1 \rangle$	$\langle e_2 \rangle = \langle y_2 \rangle - \langle b_2 \rangle$	

s_1和s_2共同计算$\langle d \rangle = \langle d_1 \rangle + \langle d_2 \rangle, \langle e \rangle = \langle e_1 \rangle + \langle e_2 \rangle$并公布$\langle d \rangle, \langle e \rangle$。

s_1 计算 $\langle z_1 \rangle = \langle d \rangle \langle e \rangle + \langle a_1 \rangle \langle e \rangle + \langle b_1 \rangle \langle d \rangle + \langle c_1 \rangle$

s_2 计算 $\langle z_2 \rangle =$ $\langle a_2 \rangle \langle e \rangle + \langle b_2 \rangle \langle d \rangle + \langle c_2 \rangle$

$z = \langle z_1 \rangle + \langle z_2 \rangle = x \times y$

Secret Sharing(SS)

南京航空航天大学 Nanjing University of Aeronautics and Astronautics

Shamir秘密共享

Alice拥有信息 $a_0 = x$ 和 Bob拥有信息 $b_0 = y$，目标共同计算$z = x + y$

① Alice 随机选择t-1个数，$a_1, a_2, \ldots\ldots, a_{t-1}$

② 构建多项式 $f(x_i) = a_0 + a_1 x_i^1 + a_2 x_i^2 + \cdots + a_{t-1} x_i^{t-1}$

③ Bob 随机选择t-1个数，$b_1, b_2, \ldots\ldots, b_{t-1}$

④ 构建多项式 $\varphi(x_i) = b_0 + b_1 x_i^1 + b_2 x_i^2 + \cdots + b_{t-1} x_i^{t-1}$

⑤ 随机选择n组x_i，Alice计算$f(x_1)\ldots f(x_n)$ 和 Bob计算$\varphi(x_1)\ldots\varphi(x_n)$分发给$n$个节点，$n$个节点分别获得

$(x_1, f(x_1) + \varphi(x_1)), (x_2, f(x_2) + \varphi(x_2)), \ldots, (x_n, f(x_n) + \varphi(x_n))$

⑥ 可信第三方任意选择t个节点进行计算：// $a_0 + b_0$，$a_1 + b_1$，……，$a_{t-1} + b_{t-1}$　t 个未知数

$$\begin{bmatrix} 1 & x_1^1 & x_1^2 & x_1^3 \cdots & x_1^{t-1} \\ 1 & x_2^1 & x_2^2 & x_2^3 \cdots & x_2^{t-1} \\ \vdots & & & \ddots & \vdots \\ 1 & x_t^1 & x_t^2 & x_t^3 \cdots & x_t^{t-1} \end{bmatrix} * \begin{bmatrix} a_0 + b_0 \\ a_1 + b_1 \\ \vdots \\ a_{t-1} + b_{t-1} \end{bmatrix} = \begin{bmatrix} f(x_1) + \varphi(x_1) \\ f(x_2) + \varphi(x_2) \\ \vdots \\ f(x_t) + \varphi(x_t) \end{bmatrix}, \quad \begin{bmatrix} a_0 + b_0 \\ a_1 + b_1 \\ \vdots \\ a_{t-1} + b_{t-1} \end{bmatrix}$$

恢复多项式，$x_i = 0$ 得到$a_0 + b_0$

Homomorphic Encryption(HE)

南京航空航天大学 Nanjing University of Aeronautics and Astronautics

乘法同态加密(RSA)

简介：RSA是1977年由MIT的罗纳德·李维斯特(Ron Rivest)、阿迪·萨莫尔(Adi Shamir)和伦纳德·阿德曼(Leonard Adleman)一起提出的，广泛应用在互联网传输、银行以及信用卡产业中。

1.密钥生成 (安全密钥长度 n　RSA_768(破解) < RSA_1024 < RSA_2048)

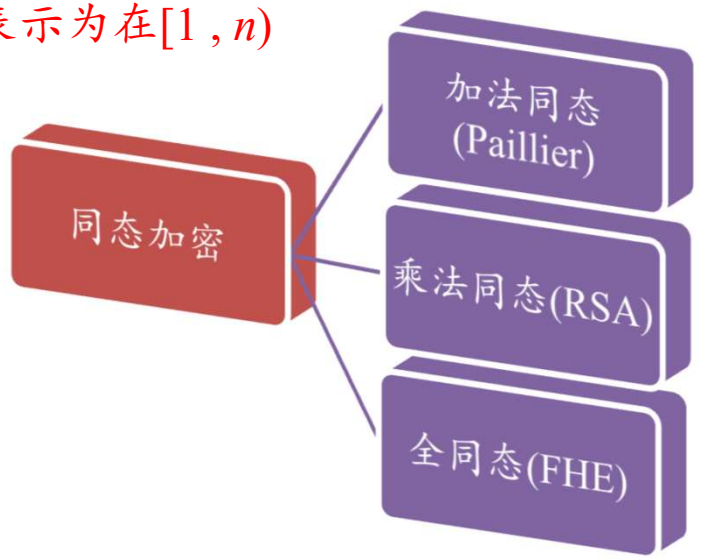
① 选取两个大素数 p和q (1024位或2048位)，$gcd(p,q)=1$ (最大公约数，互素)

②计算$n=p*q$ ，$\varphi(n)=(p-1)*(q-1)$，其中$\varphi(n)$为欧拉函数，表示为在$[1,n)$区间与n互素元素的个数，比如 $\varphi(5)$=4 $(1,2,3,4)$

③选取大整数e，$1<e<\varphi(n)$且$gcd(e,\varphi(n))=1$

④计算e的逆元d，即 $ed\equiv 1 \bmod \varphi(n)$ (ed与1模$\varphi(n)$同余)

⑤密钥对：公钥(n,e) (公开)，私钥(n,d) (秘密保存)



2.加密/解密

Alice(发送者)利用Bob公钥加密信息m　密文：$C=m^e \bmod n$

3.解密

Bob(接收者)利用私钥解密密文C　明文：$C^d \bmod n=(m^e \bmod n)^d \bmod n=m^{ed} \bmod n=m \bmod n=m$

Homomorphic Encryption(HE)

加法同态加密(Paillier)

简介：Paillier加密算法是Pascal Paillier 在1999年发明的概率公钥加密算法，该算法基于复合剩余类的困难问题，是一种满足加法的同态加密算法，已经广泛应用在加密信号处理或第三方数据处理领域。

1.密钥生成

① 选取两个大素数 p和q (1024位或2048位)， p与q互为素数

②计算$n = p * q$, (n为Carmichael数) ， $\lambda = lcm\{(p-1), (q-1)\}$, 其中$lcm$为最小公倍数

③选取随机数$g \in z_{n^2}^*$，其中 $z_{n^2}^*$表示为n^2阶整数群{ 0, 1, 2, ..., n^2-1 }, * 表示不包含 0 元素

④设置函数 $L(x) = \frac{x-1}{n}$ ， 计算$u = \left(L\left(g^{\lambda} \bmod n^2\right)\right)^{-1} \bmod n$

⑤密钥对：公钥(n, g)，私钥(λ, u)

2.加密

Alice(发送者)利用Bob公钥加密信息m 密文：$C = g^m r^n \bmod n^2$ ， 其中随机数$r \in z_n^*$

3.解密

Bob(接收者)利用私钥解密密文C 明文：$m = L\left(C^{\lambda} \bmod n^2\right) * u \bmod n$

Homomorphic Encryption(HE)

- 乘法同态运算法则(RSA)

对于明文m_1和与 m_2，满足：$[[m_1 \times m_2]] = [[m_1]] \otimes [[m_2]]$ ×与⊗意义一致，区分明密文域计算

安全性：数学困难问题(n 的大整数分解)，公钥(n, e)是公开，私钥(n, d)

$n = p * q$ $\varphi(n) = (p-1) * (q-1)$ $ed \equiv 1 \bmod \varphi(n)$

- 加法同态运算法则(Paillier)

对于明文m_1和 m_2，满足：$[[m_1 + m_2]] = [[m_1]] \otimes [[m_2]]$ 数乘：$[[m \times k]] = [[m]]^k$

安全性：数学困难问题(复合剩余类问题)

$C = g^m r^n \bmod n^2$ ，攻击者已知密文C、公开的公钥(n, g)，但明文m和随机数r未知

剩余类问题：5 mod 3=2 ，所有x mod 3 = 2的x构成的集合称为模3，2的一个剩余类(5 , 8 , 11 ,14,……)

密文C的剩余类集合表示为 $C + kn^2$，通过剩余类集合元素 $C + kn^2$ mod n^2 都会得到密文C，攻击者会利用$k = 0, \pm 1, \pm 2\ldots\ldots$ 计算 $C + kn^2$并枚举明文m和随机数r，判断$C + kn^2 = g^m r^n$是否成立，由于 mod n^2，n^2太大，剩余类元素太多暴力破解计算量太大，同时在枚举计算的时候，由于离散对数数学困难问题计算 g^m和 r^n 困难(大指数级别)。

Private Set Intersection (PSI)

南京航空航天大学 Nanjing University of Aeronautics and Astronautics

- 隐私集合求交(PSI) (RSA盲签名实现)

纵向联邦学习中的关键前置步骤，微众在与多家银行联合建模前，需要找到银行之间共有的数据样本，并且不暴露每家银行独有的样本。

Alice拥有样本集合$\{ID_1, ID_2, ID_3, ID_4, ID_5\}$， Bob拥有样本集合$\{ID_1, ID_2, ID_3, ID_6, ID_7\}$

算法要求获得公共数据集样本$\{ID_1, ID_2, ID_3\}$，而不能暴露 $\{ID_4, ID_5, ID_6, ID_7\}$

1.签名

签名和现实生活中的签名本质一样，用于对信息发送方身份验证(防止冒充)和检验信息完整性(消息是否被篡改)。

映射关系：“多对一”，“一对一”，密码学中的映射函数都是选取抗碰撞的哈希函数，保证了输入与输出映射关系“一对一”。 (哈希碰撞：不同的输入，获得了相同的输出)

Alice (pk, sk) (发送者)

① 利用Bob公钥pk_1加密信息: 密文$C = (m)_{pk_1}$

② 利用自身私钥sk对密文C进行签名$[C]_{sk}$

③ 计算$Hash(m)$

④ 将$[C]_{sk}$和$Hash(m)$发送给Bob

Bob(pk_1, sk_1) (接收者)

⑤利用Alice公钥pk验证签名，并解密密文获得m (身份认证)

⑥对解密的m哈希处理，验证结果?=$Hash(m)$ (完整性检验

检验发送的哈希值与解密后的获得哈希值是否一致)

Private Set Intersection (PSI)

2. RSA盲签名实现PSI

$ID_i = \{ID_1, ID_2, ID_3, \ldots\ldots, ID_{n-1}, ID_n\}$　　$ID_i = \{ID_1, ID_2, ID_3, \ldots\ldots, ID_{v-1}, ID_v\}$

$H: \{0,1\}^* \to z_n^*$ 和 H'(实际需要定义)　　公钥(n, e) 私钥(n, d)

Alice　　Bob

① Alice计算$h_{a_i} = H(ID_i)$

Bob计算$h_{b_i} = H(ID_i)$　//样本ID哈希

② Bob对h_{b_i}签名 $h_{b_i}^d \bmod n$

③ Bob对签名进行哈希$h'_{b_i} = H'\left(h_{b_i}^d \bmod n\right)$

④ Alice选择 $r_i \in z_n^*$，计算$y_i = (ha_i * r_i^e) \bmod n$ (盲化h_{a_i}，对h_{a_i}保护)

⑤ Alice将y_i发送给Bob解密

⑥Bob解密获得$y'_i = \left[\left(ha_i^d \bmod n\right) * r_i\right] \bmod n$给Alice

⑦ Alice计算$ha_i^d \bmod n = (y'_i / r_i) \bmod n$　//通过⑥转换，如：5=(1/3) mod 7　5乘3 mod 7余1

⑧ Alice获得 $ha_i^d \bmod n$ 与Bob②对应，并计算$h'_{a_i} = H'\left(h_{a_i}^d \bmod n\right)$与Bob③对应

⑨ Alice计算$h'_{a_i} \cap h'_{b_i}$，获得相同样本ID (h'_{a_i}与h'_{b_i}中携带各自ID信息，相同ID哈希值唯一)

Secure Linear Regression

- 纵向联邦安全线性回归算法(Paillier实现)

Fate中使用Host作为联邦学习中的参与者，Guest作为建模发起者(拥有label一方)，可信三方为Arbiter。

Host 方拥有样本特征x_i^A，Guest方拥有样本特征x_i^B 与标签y_i，密钥由Arbiter掌控。

$$L=\frac{1}{2}\sum_{i=1}\left(\theta_A x_i^A+\theta_B x_i^B-y_i\right)^2+\frac{\lambda}{2}\left(\theta_A^2+\theta_B^2\right) \qquad [[L]]=\left[\left[\frac{1}{2}\sum_{i=1}\left(\theta_A x_i^A+\theta_B x_i^B-y_i\right)^2+\frac{\lambda}{2}\left(\theta_A^2+\theta_B^2\right)\right]\right]$$

Host方加密梯度: $$\left[\left[\frac{\partial L}{\partial \theta_A}\right]\right]=\left[\left[\sum_{i=1}\theta_A\left(x_i^A\right)^2+x_i^A\left(\theta_B x_i^B-y_i\right)+\lambda\theta_A\right]\right]$$

$$=\left[\left[\sum_{i=1}x_i^A\left(\theta_A x_i^A+\theta_B x_i^B-y_i\right)\right]\right]\otimes[[\lambda\theta_A]]$$

Host 与Guest 计算本地加密梯度，需要获得红色公共部分

Guest方加密梯度: $$\left[\left[\frac{\partial L}{\partial \theta_B}\right]\right]=\left[\left[\sum_{i=1}\theta_A x_i^A x_i^B+x_i^B\left(\theta_B x_i^B-y_i\right)+\lambda\theta_B\right]\right]$$

$$=\left[\left[\sum_{i=1}x_i^B\left(\theta_A x_i^A+\theta_B x_i^B-y_i\right)\right]\right]\otimes[[\lambda\theta_B]]$$

Secure Linear Regression

以Host为例，根据paillier加法同态运算性质展开：

$$\left[\left[\sum_{i=1} x_i^A\left(\theta_A x_i^A + \theta_B x_i^B - y_i\right)\right]\right]$$

① $$= \left[\left[x_1^A\left(\theta_A x_1^A + \theta_B x_1^B - y_1\right) + x_2^A\left(\theta_A x_2^A + \theta_B x_2^B - y_2\right) + \cdots + x_i^A\left(\theta_A x_i^A + \theta_B x_i^B - y_i\right)\right]\right]$$

② $$= \left[\left[\theta_A x_1^A + \theta_B x_1^B - y_1\right]\right]^{x_1^A} \otimes \left[\left[\theta_A x_2^A + \theta_B x_2^B - y_2\right]\right]^{x_2^A} \otimes \cdots \otimes \left[\left[\theta_A x_i^A + \theta_B x_i^B - y_i\right]\right]^{x_i^A}$$

$$= \prod_{i=1} \left[\left[\theta_A x_i^A + \theta_B x_i^B - y_i\right]\right]^{x_i^A} = \prod_{i=1} \left(\left[\left[\theta_A x_i^A\right]\right] \otimes \left[\left[\theta_B x_i^B - y_i\right]\right]\right)^{x_i^A}$$

Host向Guest传递加密后的 $\left[\left[\theta_A x_i^A\right]\right]$，Guest 计算中间结果 $\left[\left[d_i\right]\right] = \left[\left[\theta_A x_i^A\right]\right] \otimes \left[\left[\theta_B x_i^B - y_i\right]\right]$ 共享

Host与Guest在本地分别计算 $\left[\left[\frac{\partial L}{\partial \theta_A}\right]\right] = \prod_{i=1} \left[\left[d_i\right]\right]^{x_i^A} \otimes \left[\left[\lambda\theta_A\right]\right]$ ， $\left[\left[\frac{\partial L}{\partial \theta_B}\right]\right] = \left[\left[d_i\right]\right]^{x_i^B} \otimes \left[\left[\lambda\theta_B\right]\right]$，各自将加密梯度传递给Arbiter进行解密后返回更新模型。

Secure Logistic Regression

南京航空航天大学 Nanjing University of Aeronautics and Astronautics

- 纵向联邦安全逻辑回归算法(Paillier实现)

Host 方拥有样本特征x_i^A，Guest方拥有样本特征x_i^B与标签y_i ，令 $z = \theta_A x_i^A + \theta_B x_i^B$

$$L = -\sum_{i=1} y_i \log(g(z)) + (1-y_i)\log(1-g(z)) = \sum_{i=1}(1-y_i)z + \log(1+e^{-z})$$

Host方加密梯度:

$$\left[\left[\frac{\partial L}{\partial \theta_A}\right]\right] = \left[\left[\frac{\partial L}{\partial z}\frac{\partial z}{\partial \theta_A}\right]\right] = \left[\left[\sum_{i=1}\left(\frac{1}{1+e^{-z}} - y_i\right)x_i^A\right]\right] \approx \left[\left[\sum_{i=1}\left(\frac{1}{2} + \frac{z}{4} - y_i\right)x_i^A\right]\right]$$

$$= \prod_{i=1}\left\{[[2]] \otimes \left[\left[\theta_A x_i^A\right]\right] \otimes \left[\left[\theta_B x_i^B - 4y_i\right]\right]\right\}^{\frac{1}{4}x_i^A} = \prod_{i=1}\left\{[[2]] \otimes [[d_i]]\right\}^{\frac{1}{4}x_i^A}$$

由于z位于sigmoid指数上，Guest的$\theta_B x_i^B$与label y_i分离且无法对自身信息进行完整加密，利用Taylor二阶展开近似

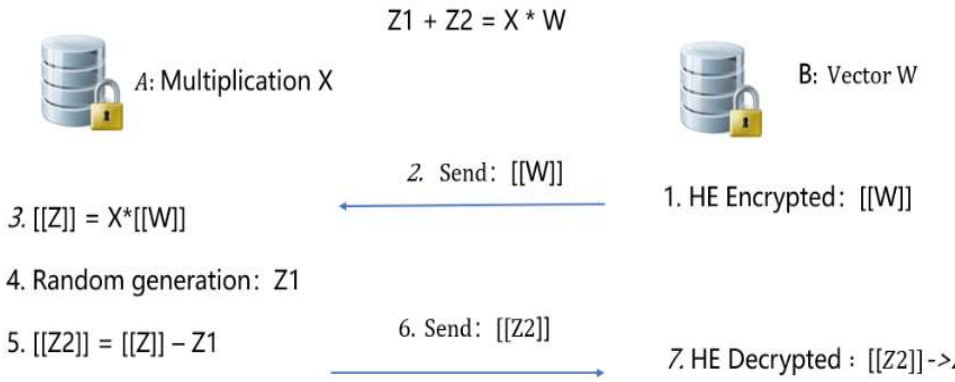
Guest方加密梯度:

$$\left[\left[\frac{\partial L}{\partial \theta_B}\right]\right] = \left[\left[\frac{\partial L}{\partial z}\frac{\partial z}{\partial \theta_B}\right]\right] = \left[\left[\sum_{i=1}\left(\frac{1}{1+e^{-z}} - y_i\right)x_i^B\right]\right] \approx \left[\left[\sum_{i=1}\left(\frac{1}{2} + \frac{z}{4} - y_i\right)x_i^B\right]\right]$$

$$= \prod_{i=1}\left\{[[2]] \otimes \left[\left[\theta_A x_i^A\right]\right] \otimes \left[\left[\theta_B x_i^B - 4y_i\right]\right]\right\}^{\frac{1}{4}x_i^B} = \prod_{i=1}\left\{[[2]] \otimes [[d_i]]\right\}^{\frac{1}{4}x_i^B}$$

Heterogeneous SSHE Logistic Regression

- 安全矩阵乘法(Secure Matrix Multiplication，SMM)(Paillier实现)



(pk, sk)

Alice　　Bob

①Bob利用pk加密参数w，获得密文$[[w]]$

②Bob $[[w]]$发送给Alice

③ Alice计算$[[z]] = x[[w]]$

④ Alice选取随机数z_1　　//Alice保护x，将对$[[z]]$拆分

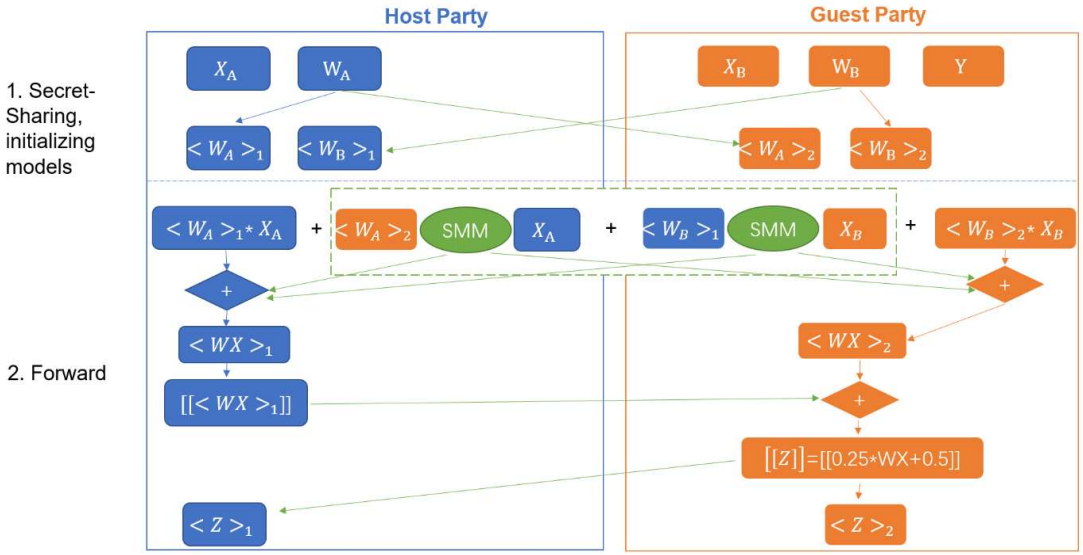
⑤ Alice得到$[[z]] = z_1 + [[z_2]]$

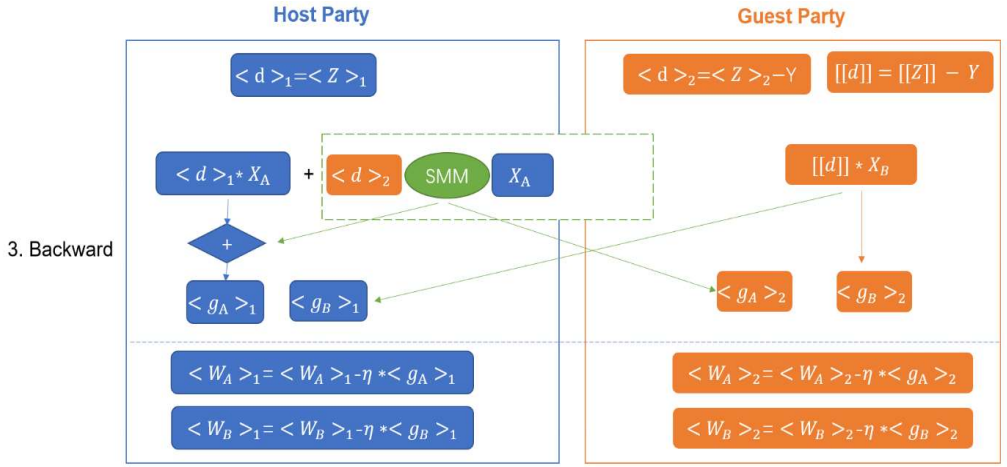
⑥ Alice发送$[[z_2]]$给Bob

⑦ Bob利用sk解密$[[z_2]]$，获得明文z_2

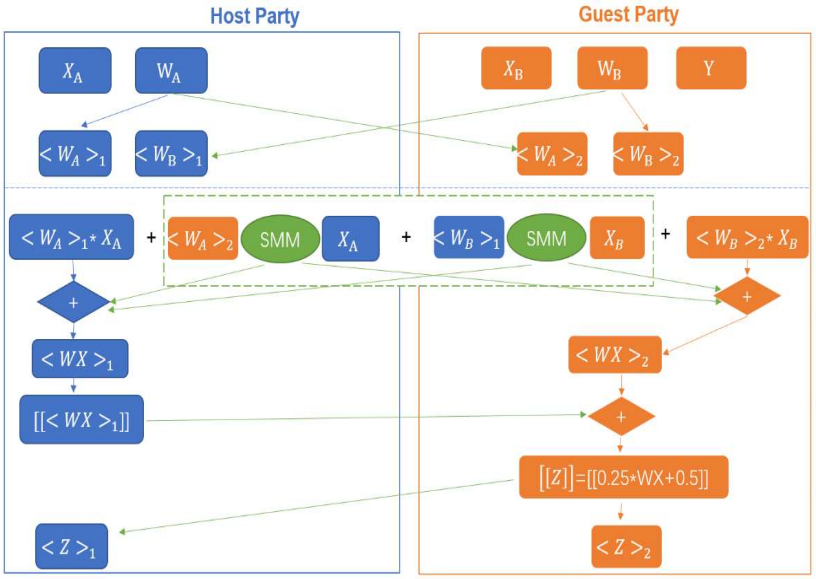
Heterogeneous SSHE Logistic Regression

- 无可信三方纵向联邦安全逻辑回归算法(SSHE)





Heterogeneous SSHE Logistic Regression



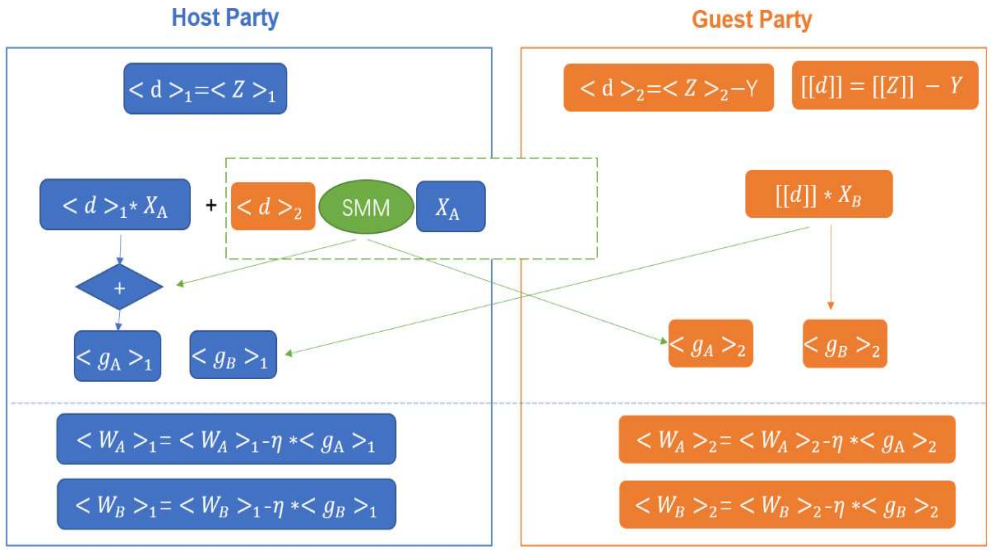
$$\left[\left[\sum_{i=1}\left(\frac{1}{1+e^{-wx}}-y_i\right)x_i^A\right]\right]\approx\left[\left[\sum_{i=1}\left(\frac{1}{2}+\frac{wx}{4}-y_i\right)x_i^A\right]\right]$$

这部分计算的是 $z=\frac{1}{2}+\frac{wx}{4}$

	Host	Guest
	x_A w_A	x_B w_B
①	$\langle w_A\rangle_1$ $\langle w_A\rangle_2$	$\langle w_B\rangle_1$ $\langle w_B\rangle_2$ //参数切片
②	$\langle w_A\rangle_1$ $\langle w_B\rangle_1$	$\langle w_A\rangle_2$ $\langle w_B\rangle_2$ //切片交换
③	$\langle z_A\rangle_1 = x_A * \langle w_A\rangle_1$	$\langle z_B\rangle_2 = x_B * \langle w_B\rangle_2$ //未交换信息计算
④	$[[\langle w_A\rangle_2]]$	$[[\langle w_B\rangle_1]]$//交换切片被加密返回
	$[[\langle z_A\rangle_2]] = x_A * [[\langle w_A\rangle_2]]$	$[[\langle z_B\rangle_1]] = x_B * [[\langle w_B\rangle_1]]$
	$= \langle\langle z_A\rangle_2\rangle_1 + [[\langle\langle z_A\rangle_2\rangle_2]]$ SMM	$= \langle\langle z_B\rangle_1\rangle_1 + [[\langle\langle z_B\rangle_1\rangle_2]]$
	$[[\langle\langle z_B\rangle_1\rangle_2]]$	$[[\langle\langle z_A\rangle_2\rangle_2]]$ //红框交换进行解密
⑤	$\langle wx\rangle_1 = \langle z_A\rangle_1 + \langle\langle z_A\rangle_2\rangle_1 + \langle\langle z_B\rangle_1\rangle_2$	$\langle wx\rangle_2 = \langle z_B\rangle_2 + \langle\langle z_B\rangle_1\rangle_1 + \langle\langle z_A\rangle_2\rangle_2$
⑥	$[[\langle wx\rangle_1]]$	$[[\langle wx\rangle_2]]$

⑦ Guest: $[[z]] = \{[[\langle wx\rangle_1]] \otimes [[\langle wx\rangle_2]]\}^{\frac{1}{4}} \otimes [[0.5]] = [[\langle z\rangle_1]] + [[\langle z\rangle_2]]$

Heterogeneous SSHE Logistic Regression



⑧ $[[\langle z\rangle_1]]$ $[[\langle z\rangle_2]]$ // 对$[[z]]$拆分

⑨ $\langle z\rangle_1$ $\langle z\rangle_2$ //各自解密

⑩ $\langle d\rangle_1 = \langle z\rangle_1$ $\langle d\rangle_2 = \langle z\rangle_2 - Y$

// 完整中间部分$d = \langle d\rangle_1 + \langle d\rangle_2$

⑪ $[[d]] = [[z]] - Y$ // ⑦ Guest已经计算$[[z]]$

⑫ $[[\langle g_B\rangle_1]] + [[\langle g_B\rangle_2]] = [[d]] * x_B$

⑬ $[[\langle g_B\rangle_1]] \longrightarrow \langle g_B\rangle_1$ // 解密，辅助计算Guest部分梯度

⑭ $\langle g_A\rangle_1 = \langle d\rangle_1 * x_A$ // Host计算部分梯度

⑮ // Host 缺少$\langle d\rangle_2$ $[[\langle d\rangle_2]]$ //SMM，Guest辅助

$$[[\langle g_A\rangle_2]] = [[\langle d\rangle_2]] * x_A$$

$$= \langle\langle g_A\rangle_2\rangle_1 + [[\langle\langle g_A\rangle_2\rangle_2]]$$

$[[\langle\langle g_A\rangle_2\rangle_2]] \longrightarrow \langle\langle g_A\rangle_2\rangle_2$

⑭ $\langle g_A\rangle_1$ ⑮ $\langle\langle g_A\rangle_2\rangle_1$

⑬ $\langle g_B\rangle_1$

⑮ $\langle\langle g_A\rangle_2\rangle_2$

⑫ $\langle g_B\rangle_2$

利用最初共享的参数切片辅助更新，交换参数信息

Heterogeneous Neural Networks

纵向联邦安全神经网络算法(Hetero NN)

① **Bottom Model** (参与者底层模型，各参与者控制)

横向联邦学习中，各参与者拥有的样本特征一致，一般本地模型由第三方或者建模发起者统一初始化。

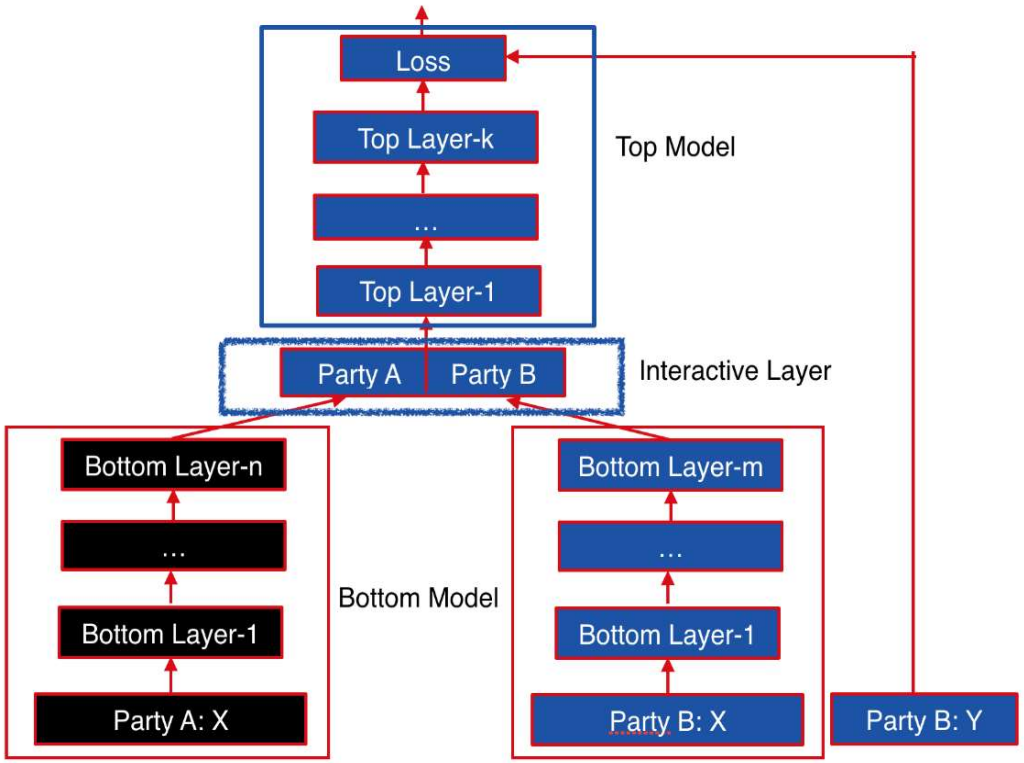
纵向联邦学习各参与者拥有的特征不一致，本地模型结构不同。

② **Interactive Layer**

各参与者的底层模型输出结果，Guest分配权重聚合，并使用激活函数非线性化处理。

③ **Top Model** (顶层模型，由建模发起者Guest控制)

将Interactive Layer的输出作为输入，训练Top Model。



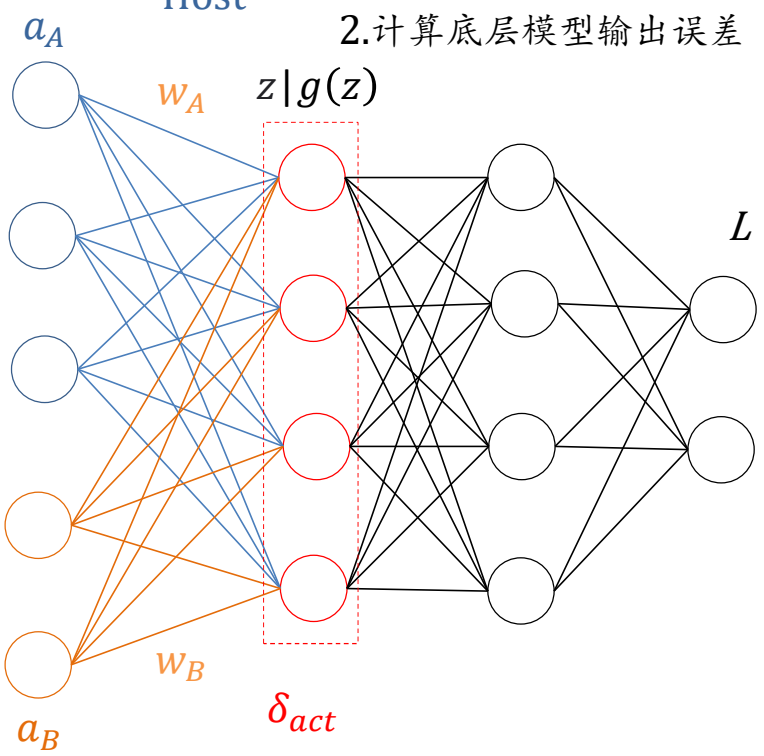
Heterogeneous Neural Networks

1. Forward propagation

	Host	Guest
Bottom Model		
①	a_A= Bottom. forward propagation(x_i^A)	a_B= Bottom. forward propagation(x_i^B)
Interactive Layer		
②	Host 加密底层模型输出a_A，$[[a_A]]$	Guest 为a_A分配权重w_A，计算$[[z_A]] = [[a_A]] * w_A$，选择噪声ε_B，计算$[[z_A + \varepsilon_B]] = [[z_A]] \otimes [[\varepsilon_B]]$ //噪声ε_B，Guest保护自身为Host分配的w_A
③	解密$[[z_A + \varepsilon_B]]$获得 $z_A + \varepsilon_B$，计算噪声 $a_A * \varepsilon_{acc}$，将$z_A + \varepsilon_B + a_A * \varepsilon_{acc}$给Guest。 // ε_{acc}为反向传播中累计噪声 // $a_A * \varepsilon_{acc}$保护z_A，若直接返回z_A，Guest知道ε_B与w_A，能反推Host 底层输出a_A	
④		$z = z_B + (z_A + \varepsilon_B + a_A * \varepsilon_{acc}) - \varepsilon_B$，其中$z_B = a_B w_B$ // 消除噪声ε_B干扰后的z
⑤		计算$g(z)$ //交互层激活值计算

Heterogeneous Neural Networks

2. Back propagation



1.反向传播，更新w_B与w_A ⑥

2.计算底层模型输出误差 ⑦

⑧

⑨

⑩

⑪

⑫

⑬

Host

Guest

交互层激活误差 $\delta_{act}=\frac{\partial L}{\partial z}$ // 交互层误差项计算

计算$\delta_{bottom_B}=\delta_{act}*\frac{\partial z}{\partial a_B}=\delta_{act}*w_B$ //Guest输出误差

w_B梯度$g_B=\delta_{act}*\frac{\partial z}{\partial w_B}=\delta_{act}*a_B$ //Guest w_B梯度计算

$w_B:=w_B-\eta*g_B$ //更新权重w_B

生成噪声ε_B，计算$\left[\left[\delta_{act}*\frac{\partial z}{\partial w_A}+\varepsilon_B\right]\right]=[[\delta_{act}]]^{\frac{\partial z}{\partial w_A}}\otimes[[\varepsilon_B]]$

// Guest计算w_A梯度，噪声保护δ_{act}(因为$\frac{\partial z}{\partial w_A}$Host知道)

$\delta_{act}*\frac{\partial z}{\partial w_A}+\varepsilon_B=\delta_{act}*a_A+\varepsilon_B$ // Host解密获得存在噪声的w_A梯度

生成噪声ε_A，计算$\delta_{act}*a_A+\varepsilon_B+\frac{\varepsilon_A}{\eta}$ // Host保护a_A，利用噪声ε_A

$\varepsilon_{acc}:=\varepsilon_{acc}+\varepsilon_A$并加密$[[\varepsilon_{acc}]]$ // 每次反向传播累计噪声ε_A，记为ε_{acc}

$w_A:=w_A-\eta*(\delta_{act}*a_A+\varepsilon_B+\frac{\varepsilon_A}{\eta}-\varepsilon_B)$ //更新权重w_A

$=w_A-\eta*(\delta_{act}*a_A)-\varepsilon_A$

$=W_A^{True}-\varepsilon_{acc}$ //每次iter都会减去ε_A，累计噪声为ε_{acc}

计算$\left[\left[\delta_{bottom_A}\right]\right]=[[\delta_{act}*(w_A+\varepsilon_{acc})]]=\{[[w_A]]\otimes[[\varepsilon_{acc}]]\}^{\delta_{act}}$ // Host解密

// w_A在上次iter多$-\varepsilon_{acc}$，需修正$w_A+\varepsilon_{acc}$

THANKS