联邦学习 Federated Learning

zhuanlan.zhihu.com/p/93761403

Kylin 机器学习考古学家

【参加完CNCC2019会议的一个小汇报，在此记录分享】

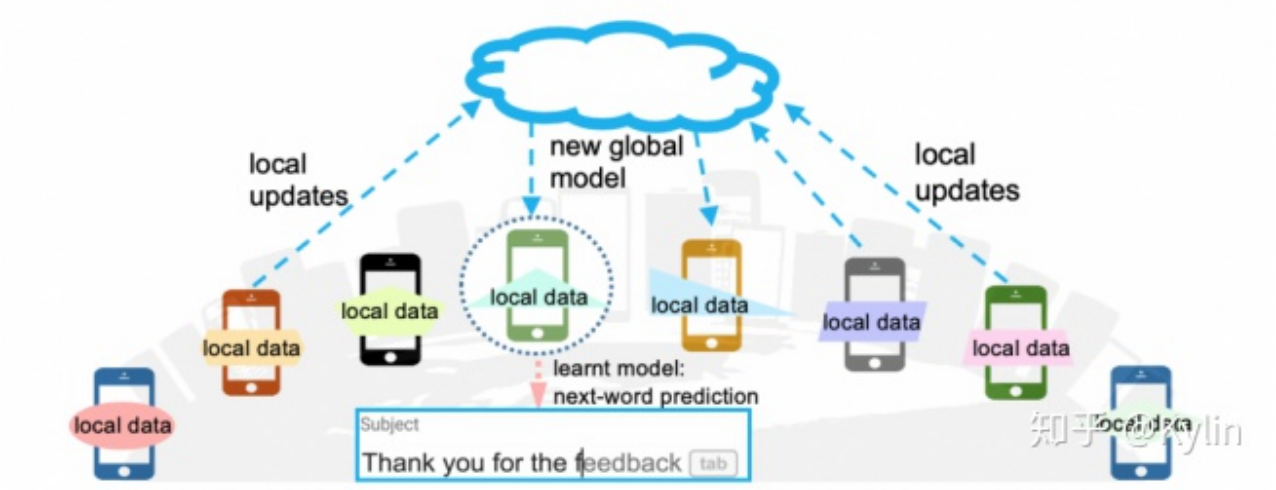
目录

---

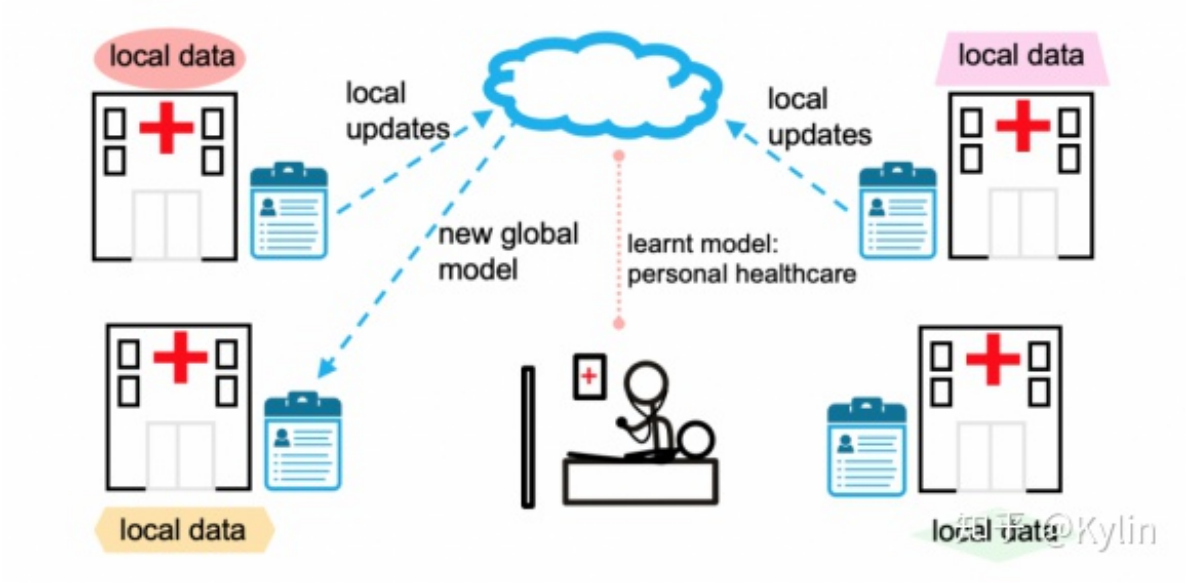
1. 联邦学习的发展背景

1.1 目前面临的一些问题:

系统部署：移动手机和可穿戴设备是现代十分常见的数据产生设备。这些设备每天都会产生巨量的各种形式的数据。考虑到算力需求，数据传输以及个人隐私的限制，系统部署越来越倾向于在本地存储数据，模型计算由边缘设备完成。[1]



数据孤岛：在大多数行业中，数据是以孤岛的形式存在的，由于行业竞争、隐私安全、行政手续复杂等问题，即使是在同一个公司的 不同部门之间实现数据整合也面临着重重阻力，在现实中想要将分散在各地、各个机构的数据进行整合几乎是不可能的，或者说所需的成本是巨大的。[2]



1.2 联邦学习成为可行方案

如何在满足数据隐私、安全和监管要求的前提下，设计一个机器学习框架，让人工智能系统能够更加高效、准确的共同使用各自的数据，是当前人工智能发展的一个重要课题。我们倡议把研究的重点转移 到如何解决数据孤岛的问题。我们提出一个满足隐私保护和数据安全的一个可行的解决方案，叫做联邦学习[3][4]

2. 联邦学习的概念

2.1 联邦学习公式化定义[1]

经典的联邦学习问题需要从上百万的远程设备中存储的海量数据里面学习到一个全局的统计模型。这个任务可以用以下目标函数来表述：

$$\min \ F(w), \ \ where \ F(w) := \sum_{k=1}^{m} p_k F_k(w)$$

其中，m代表设备总量，F_k为第个k 设备的本地目标函数，p_k被定义为对应设备的影响权重。p_k具有如下性质：$p_k \geq 0$ 并且 $\sum_{k=1} p_k = 1$。F_k通常被定义为基于本地数据的经验风险。

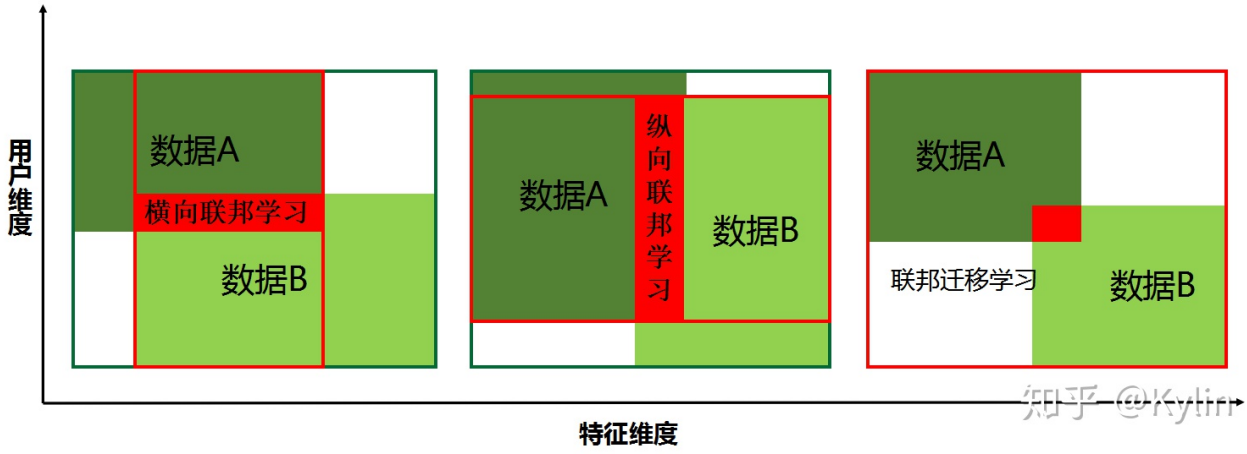
2.2 联邦学习与现有研究的区别

	联邦学习	差分隐私理论	分布式机器学习
是否传输数据	X	O	O
是否加密/加噪声	O	O	O
模型性能是否影响	X	O	X
是否获取数据所有权	X	O	O

联邦学习而言，首先在于联邦学习中的工作节点代表的是模型训练的数据拥有方，其对本地的数据具有完全的自治权限，可以自主决定何时加入联邦学习进行建模，相对地在参数服务器中，中心节点始终占据着主导地位。[2]

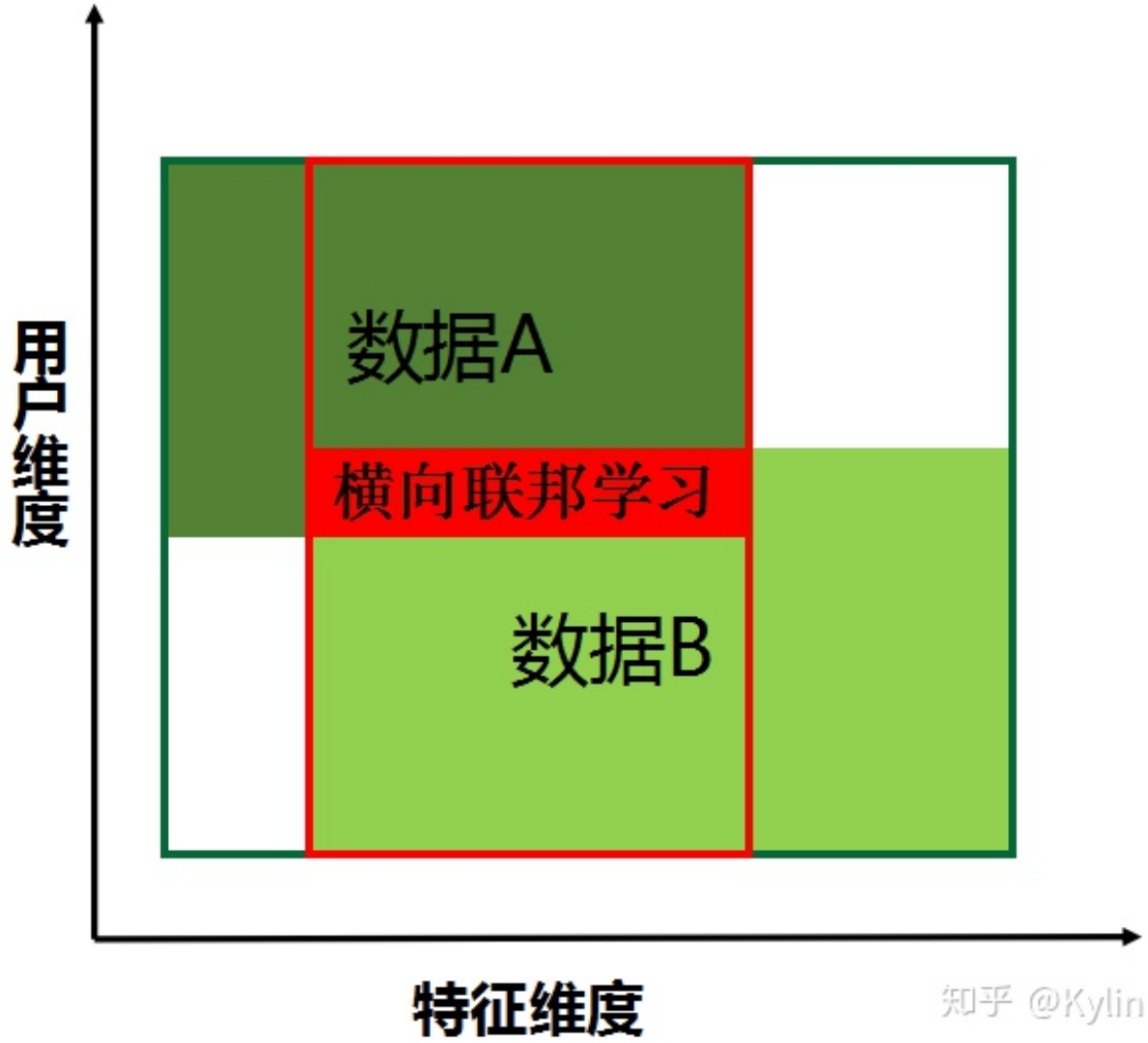
3. 联邦学习的分类

根据数据集分布情况，可以将联邦学习分为以下三个类别：横向联邦学习、纵向联邦学习与联邦迁移学习[2]。



3.1 横向联邦学习：

在两个数据集的用户特征重叠较多而用户重叠较少的情况下，我们把数据集按照横向即特征维度切分，并取出双方特征相同而用户不完全相同的那部分数据进行训练。这种方法叫做横向联邦学习。



应用场景：

- 比如有两家不同地区银行，它们的用户群体分别来自各自所在的地区，相互的交集很小。但是，它们的业务很相似，因此，记录的用户特征是相同的。此时，就可以使用横向联邦学习来构建联合模型。
- Google 在 2017 年提出了一个针对安卓手机模型更新的数据联合建模方案[5-6] ：在单个用户使用安卓手机时不断在本地更新模型参数并将参数上传到安卓云上，从而使特征维度相同的各数据拥有方建立联合模型的一种联邦学习方案 。

横向联邦学习的学习过程[7-8]：

step1：在本地计算模型梯度，然后将梯度结果加密上传到服务器；

step2：服务器A聚合各用户的梯度更新模型参数；

step3：服务器A返回更新后的模型给各参与方；

step4：各参与方基于加密梯度更新各自模型。

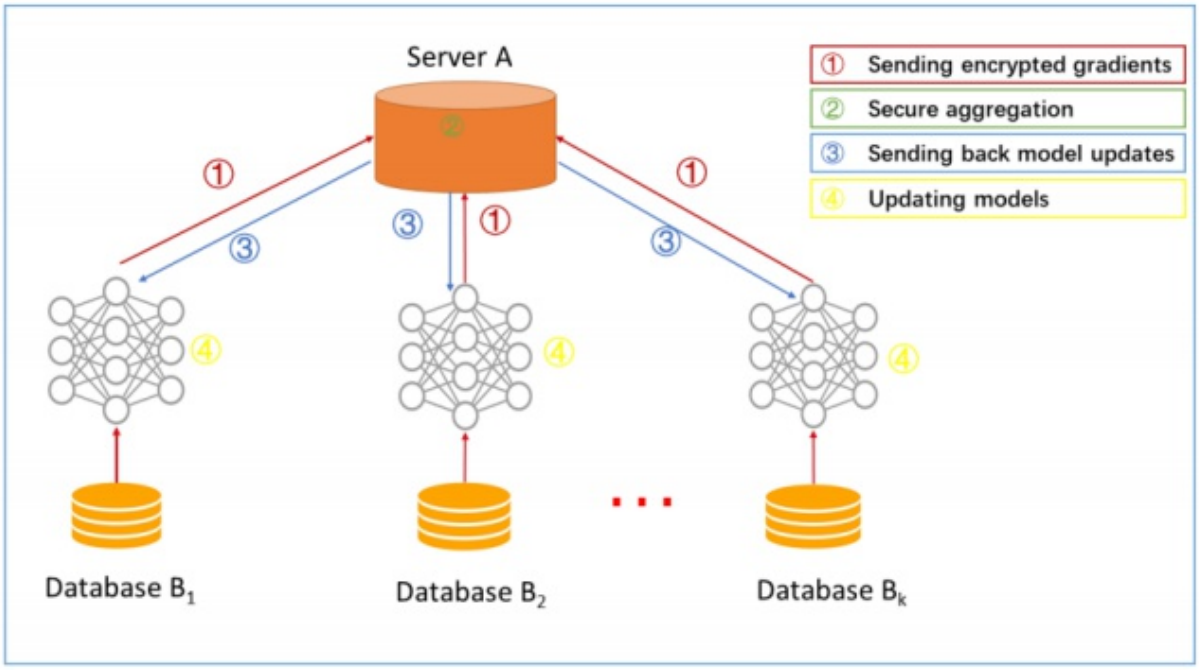
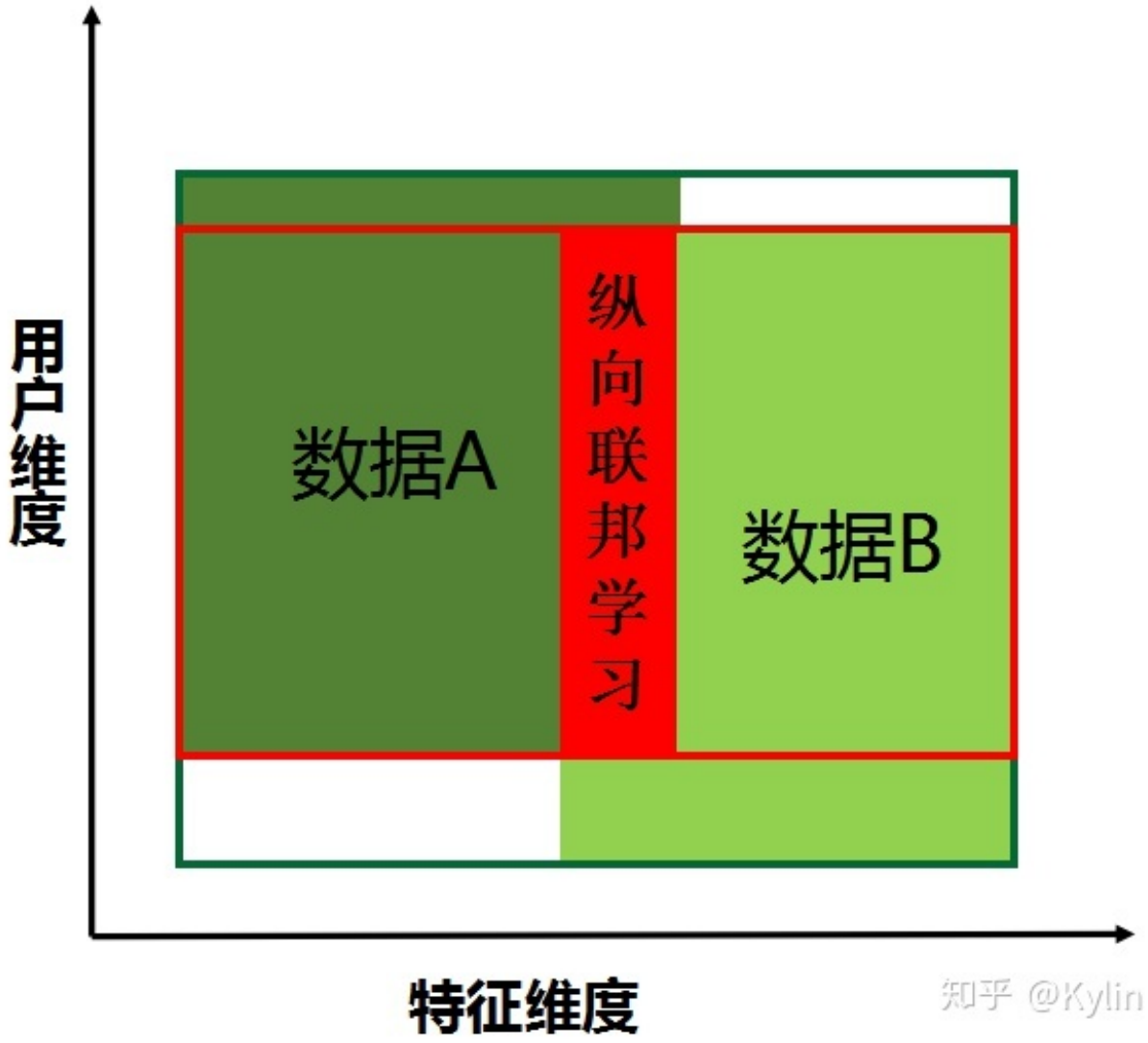


Fig. 3. Architecture for a horizontal federated learning system

3.2 纵向联邦学习：

在两个数据集的用户重叠较多而用户特征重叠较少的情况下，我们把数据集按照用户维度切分，并取出双方用户相同而用户特征不完全相同的那部分数据进行训练。



应用场景：

> 比如有两个不同机构， 一家是某地的银行，另一家是同一个地方的电商 。 它们的用户群体很有可能包含该地的大部分居民，因此用户的交集较大。但是，由于银行记录的都是用户的收支行为与信用评级，而电商则保有用户的浏览与购买历史，因此它们的用户特征交集较小。纵向联邦学习就是将这些不同特征在加密的状态下加以聚合，以增强模型能力的联邦学习。

纵向联邦学习的学习过程[7-8]：

第一步：第三方C加密样本对齐[9-10]。是在系统级做这件事，因此在企业感知层面不会暴露非交叉用户。

第二步：对齐样本进行模型加密训练：

step1：由第三方C创建加密数据对，并且向A和B发送公钥；

step2：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；

step3：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；

step4：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。

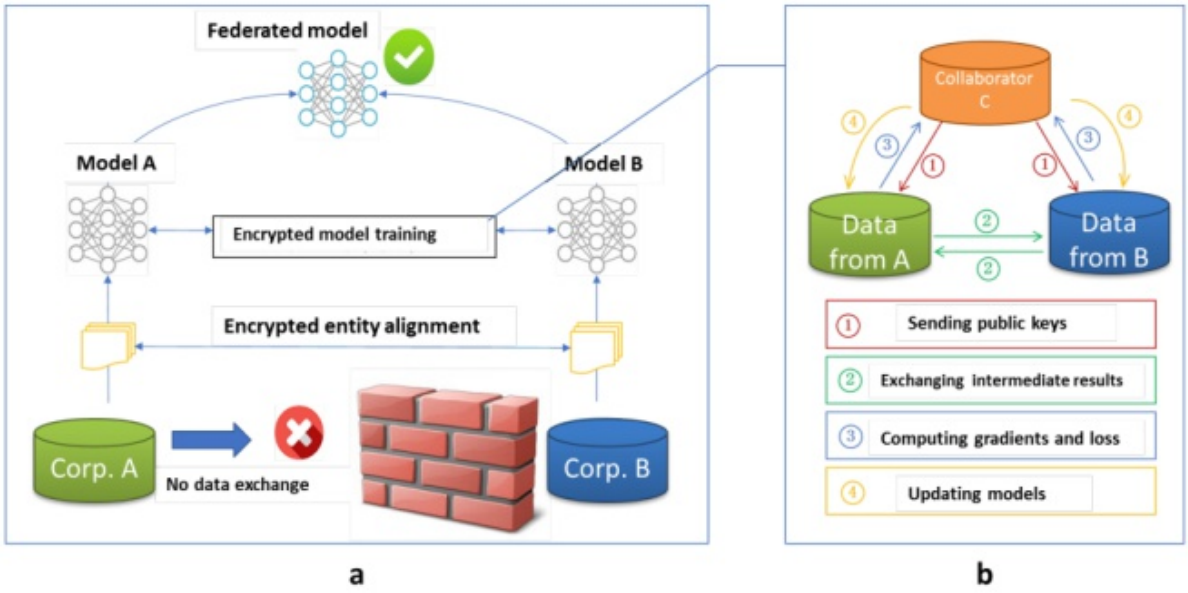


Fig. 4. Architecture for a vertical federated learning system

下面的推导我进行了部分修改，与相关原著有些不同，如有错误忘多指正。

步骤解读：以线性回归和可加性同态加密为例具体说明纵向联邦学习的训练过程[7][12]：

给定学习率 η，正则化参数 λ，两组数据集：$\{x_i^A\}_{i\in D_A}, \{x_i^B\}_{i\in D_B}$ 以及对应的模型参数：Θ_A，Θ_B。

训练的目标函数为：

$$\min_{\Theta_A,\Theta_B} \frac{1}{2}\sum_i (\Theta_A x_i^A + \Theta_B x_i^B - y_i)^2 + \frac{\lambda}{2}(\| \Theta_A \|^2 + \| \Theta_B \|^2)$$

令 $u_i^A = \Theta_A x_i^A$，$u_i^B = \Theta_B x_i^B$，定义 $[[\bullet]]$ 为可加性同态加密符号。那么加密之后的损失函数为：

$$[[L]] = [[\frac{1}{2}\sum_i (u_i^A + u_i^B - y_i)^2 + \frac{\lambda}{2}(\| \Theta_A \|^2 + \| \Theta_B \|^2)]]$$

令 $[[L_A]] = [[\frac{1}{2}\sum_i (u_i^A)^2 + \frac{\lambda}{2}\| \Theta_A \|^2]]$，$[[L_B]] = [[\frac{1}{2}\sum_i (u_i^B - y_i)^2 + \frac{\lambda}{2}\| \Theta_B \|^2]]$ 以及 $[[L_{AB}]] = [[\sum_i u_i^A (u_i^B - y_i)^2]]$

那么原始加密损失函数可以分解为：$[[L]] = [[L_A]] + [[L_B]] + [[L_{AB}]]$。

令 $[[L_A]] = [[\frac{1}{2}\sum_i (u_i^A)^2 + \frac{\lambda}{2}\| \Theta_A \|^2]]$，$[[L_B]] = [[\frac{1}{2}\sum_i (u_i^B - y_i)^2 + \frac{\lambda}{2}\| \Theta_B \|^2]]$ 以及 $[[L_{AB}]] = [[\sum_i u_i^A (u_i^B - y_i)^2]]$

那么原始加密损失函数可以分解为：$[[L]] = [[L_A]] + [[L_B]] + [[L_{AB}]]$。

基于上述定义，分布求出损失函数关于 Θ_A，Θ_B 梯度：$[[\frac{\partial L}{\partial \Theta_A}]] = [[\frac{\partial L_A}{\partial \Theta_A} + \frac{\partial L_{AB}}{\partial \Theta_A}]] = [[\frac{\partial L_A}{\partial \Theta_A}]] + [[\frac{\partial L_{AB}}{\partial \Theta_A}]]$。

1. 第一部分梯度：

$$\begin{aligned}
[[\frac{\partial L_A}{\partial \Theta_A}]] &= [[\frac{\partial}{\partial \Theta_A}(\frac{1}{2}\sum_i (u_i^A)^2 + \frac{\lambda}{2}\| \Theta_A \|^2)]] \\
&= [[\frac{1}{2}\sum_i \frac{\partial (u_i^A)^2}{\partial \Theta_A} + \lambda \Theta_A{}^2]] \\
&= [[\frac{1}{2}\sum_i \frac{\partial (\Theta_A x_i^A)^2}{\partial \Theta_A} + \lambda \Theta_A{}^2]] \\
&= [[\sum_i \Theta_A x_i^A \frac{\partial (\Theta_A x_i^A)}{\partial \Theta_A} + \lambda \Theta_A]] \\
&= [[\sum_i (\Theta_A x_i^A)(x_i^A)^T + \lambda \Theta_A]] \\
&= [[\sum_i u_i^A (x_i^A)^T + \lambda \Theta_A]] \quad (u_i^A = \Theta_A x_i^A)
\end{aligned}$$

2. 第二部分梯度：

$$\begin{aligned}
[[\frac{\partial L_{AB}}{\partial \Theta_A}]] &= [[\frac{\partial}{\partial \Theta_A}(\sum_i u_i^A (u_i^B - y_i))]] \\
&= [[\sum_i ((u_i^B - y_i)\frac{\partial u_i^A}{\partial \Theta_A})]] \\
&= [[\sum_i ((u_i^B - y_i)\frac{\partial}{\partial \Theta_A}(\Theta_A x_i^A))]] \\
&= [[\sum_i ((u_i^B - y_i)(x_i^A)^T]]
\end{aligned}$$

3. 最后可得：

$$\begin{aligned}
[[\frac{\partial L}{\partial \Theta_A}]] &= [[\frac{\partial L_A}{\partial \Theta_A}]] + [[\frac{\partial L_{AB}}{\partial \Theta_A}]] \\
&= [[\sum_i u_i^A (x_i^A)^T + \lambda \Theta_A]] + [[\sum_i ((u_i^B - y_i)(x_i^A)^T]] \\
&= [[(\sum_i u_i^A (x_i^A)^T + \lambda \Theta_A + \sum_i ((u_i^B - y_i)(x_i^A)^T]] \\
&= [[\sum_i (u_i^A + (u_i^B - y_i))(x_i^A)^T + \lambda \Theta_A]] \\
&= [[\sum_i d_i (x_i^A)^T + \lambda \Theta_A]] \quad ([[d_i]] = [[u_i^A + u_i^B - y_i]]) \\
&= [[\sum_i d_i (x_i^A)^T + \lambda \Theta_A]] \\
&= [[\sum_i d_i (x_i^A)^T]] + [[\lambda \Theta_A]] \quad \square
\end{aligned}$$

类似于上述过程，最终可以得到：$[[\frac{\partial L}{\partial \Theta_A}]] = [[\sum_i d_i (x_i^A)^T]] + [[\lambda \Theta_A]]$ 和 $[[\frac{\partial L}{\partial \Theta_B}]] = [[\sum_i d_i (x_i^B)^T]] + [[\lambda \Theta_B]]$

纵向联邦学习的训练过程表和推理过程表[7]：

Table 1. Training Steps for Vertical Federated Learning : Linear Regression

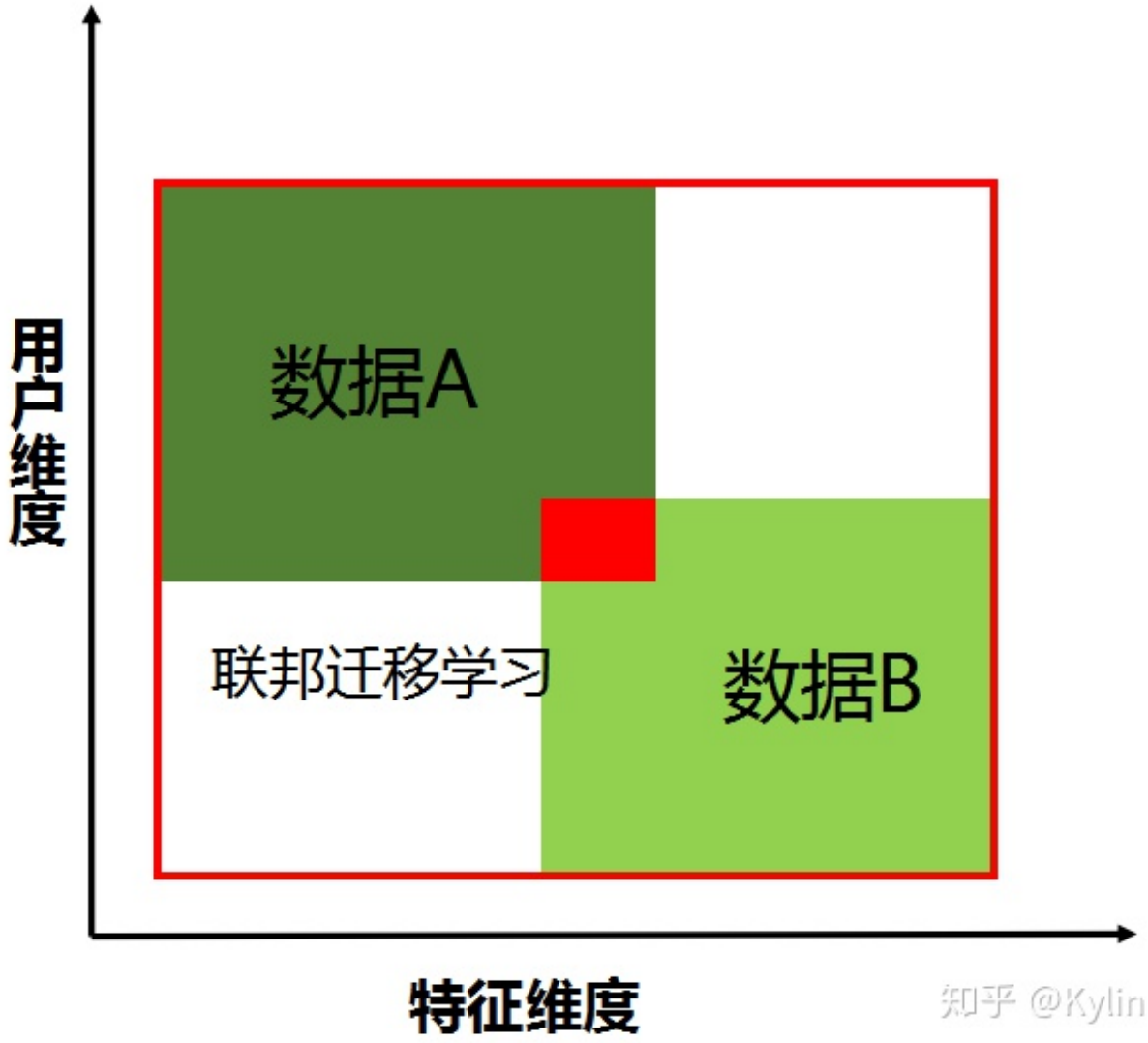
	party A	party B	party C
step 1	initialize Θ_A	initialize Θ_B	create an encryption key pair, send public key to A and B;
step 2	compute $[[u_i^A]],[[\mathcal{L}_A]]$ and send to B;	compute $[[u_i^B]],[[d_i^B]],[[\mathcal{L}]]$, send $[[d_i^B]]$ to A, send $[[\mathcal{L}]]$ to C;	
step 3	initialize R_A, 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	

Table 2. Evaluation Steps for Vertical Federated Learning : Linear Regression

	party A	party B	inquisitor C
step 0			send user ID i to A and B;
step 1	compute u_i^A and send to C	compute u_i^B and send to C;	get result $u_i^A+u_i^B$;

3.3 联邦迁移学习：

在两个数据集的用户与用户特征重叠都较少的情况下，我们不对数据进行切分，而可以利用 迁移学习来克服数据或标签不足的情况。这种方法叫做联邦迁移学习。



应用场景：

> **比如有 两个不同机构，一家是位于中国的银行，另一家是位于美国的电商。由于受到地域限制，这两家机构的用户群体交集很小。同时，由于机构类型的不同，二者的数据特征也只有小部分重合。 在这种情况下 ，要想进行有效的联邦学习，就必须 引入迁移学 习，来解决 单边数据规模小和标签 样本少的问题 ，从而提升模型的效果 。**

问题定义：

给定源数据集 $D_A := \{(x_i^A, y_i^A)\}_{i=1}^{N_A}, x_i^A \in R^a, y_i^A \in \{+1, -1\}$, 目标数据集 $D_B := \{x_j^B\}_{j=1}^{N_B}, x_j^B \in R^a$ 。

源数据集和目标数据集分隔在不同的领域，无法实现相互暴露。两个数据集之间存在着共现子集：$D_{AB} := \{(x_i^A, x_i^B)\}_{i=1}^{N_{AB}}$

共现子集里面存在小部分来自目标数据集的样本可以被源数据集的标签标定：$D_C := \{(x_i^B, y_i^A)\}_{i=1}^{N_C}$

根据上述的定义，联邦迁移学习就是基于两个互相不可暴露的数据，搭建一个迁移学习模型，以实现利用源数据的标注来预测目标数据集的结果。

基于深度学习模型的联邦迁移学习：

模型框架：给定两个深度学习模型A/B分别用于对源数据集和目标数据提取特征。$u_i^A = Net^A(x_i^A),\ u_i^B = Net^B(x_i^B)$

对目标数据集进行标定，其实是学习了一个预测函数。$\varphi(u_j^B) = \varphi(u^A, y^A, u_j^B)$

不失一般性的可以假设标签预测函数具有线性可分性，进而可以用如下方式表示：

$\varphi(u_j^B) = \Phi_A G(u_j^B)$ ，其中 $\Phi_A = \frac{1}{N_A}\sum_i^{N_A} y_i^A u_i^A,\ G(u_j^B) = (u_j^B)^T$

基于经验风险的目标函数：

$$\arg\min_{\Theta_A,\Theta_B} L_1 = \sum_i^{N_c} l_1(y_i^A, \varphi(u_i^B))$$

数据对齐目标函数：

$$\arg\min_{\Theta_A,\Theta_B} L_2 = -\sum_i^{N_{AB}} l_2(u_i^A, u_i^B)$$

正则化项：

$$L_3^A = -\sum_l^{L_A} \| \theta_l^A \|_F^2, L_4^B = -\sum_l^{L_B} \| \theta_l^B \|_F^2$$

最终的目标函数：

$$\arg\min_{\Theta_A,\Theta_B} L = L_1 + \gamma L_2 + \frac{\lambda}{2}(L_3^A + L_4^B)$$

联邦迁移学习的训练过程表和推理过程表[7]：

Algorithm 2: Federated Transfer Learning: Prediction

Input: Model parameters Θ^A, Θ^B, $\{x_j^B\}_{j \in N_B}$

Output: $\{y_j^B\}_{j \in N_B}$

B do:

$u_j^B \leftarrow Net^B(\Theta^B, x_j^B)$;

encrypts $\{[[\mathcal{G}(u_j^B)]]\}_{j \in N_B}$ and sends to A;

A do:

creates random mask m^A;

computes $[[\varphi(u_j^B)]]_B = \Phi^A [[\mathcal{G}(u_j^B)]]_B$ and sends $[[\varphi(u_j^B) + m^A]]_B$ to B;

B do:

decrypts $\varphi(u_j^B) + m^A$ and sends to A;

A do:

gets $\varphi(u_j^B)$ and y_j^B, sends y_j^B to B.

4.联邦学习的学习资料：

联邦学习生态网站：主页

Algorithm 1: Federated Transfer Learning: Training

Input: learning rate η, weight parameter γ, λ, max iterations m, tolerance t
Output: Model parameters Θ^A, Θ^B
A, B initializes Θ^A, Θ^B and creates an encryption key pair, respectively, and sends public key to each other.
$iter = 0$; **while** $iter \leq m$ **do**
  A do:
  $u_i^A \leftarrow Net^A(\Theta^A, x_i^A)$ for $i \in \mathcal{D}_A$;
  computes and encrypts $\{h_k^A(u_i^A, y_i^A)\}_{k=1,2...K_A}$ and sends to B;
  B do:
  $u_i^B \leftarrow Net^B(\Theta^B, x_i^B)$ for $i \in \mathcal{D}_B$;
  computes and encrypts $\{h_k^B(u_i^B)\}_{k=1,2...K_B}$, and send to A;
  A do:
  creates random mask m^A;
  computes $[[\frac{\partial \mathcal{L}}{\partial \theta_i^A} + m^A]]_B$ and $[[\mathcal{L}]]_B$ by equation (7) and (9) and sends to B;
  B do:
  create random mask m^B;
  computes $[[\frac{\partial \mathcal{L}}{\partial \theta_i^B} + m^B]]_A$ by equation (8) and sends to A;
  B do:
  decrypts $\frac{\partial \mathcal{L}}{\partial \theta_i^A} + m^A$,$\mathcal{L}$ and sends to A;
  A do:
  decrypts $\frac{\partial \mathcal{L}}{\partial \theta_i^B} + m^B$ and sends to B;
  B do:
  update $\theta_i^B = \theta_i^B - \eta \frac{\partial \mathcal{L}}{\partial \theta_i^B}$;
  A do:
  update $\theta_i^A = \theta_i^A - \eta \frac{\partial \mathcal{L}}{\partial \theta_i^A}$;
  if $\mathcal{L}_{prev} - \mathcal{L} \leq t$ **then**
    send stop signal to B;
    break;
  else
    $\mathcal{L}_{prev} = \mathcal{L}$;
    $iter = iter + 1$;
    continue;
  end
end

知乎 @Kylin

开源项目：

FATE FederatedAI/FATE

Eggroll WeBankFinTech/eggroll

参考资料：

[1] Federated Learning: Challenges, Methods, and Future Directions

[2] **联邦学习白皮书V1.0 深圳前海微众银行股份有限公司**

[3] **马上科普：杨强：GDPR对AI的挑战和基于联邦迁移学习的对策 杨强： GDPR 对 AI 的挑战和基于联邦迁移学习的对策**

[4] zhu anlan.zhihu.com/p/41052548 **机器之心专访杨强教授：联邦迁移学习与金融领域的 AI 落地**

[5] Konečný J, McMahan H B, Yu F X, et al. Federated learning: Strategies forimproving communication efficiency[J]. arXiv preprint arXiv:1610.05492, 2016.

[6] McMahan H B, Moore E, Ramage D, et al. Communication efficient learning of deep networks from decentralized data[J]. arXiv preprint arXiv:1602.05629, 2016.

[7] Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2019, 10(2): 12.

[8] **沐清予：详解联邦学习**Federated Learning **详解联邦学习**Federated Learning

[9] Gang Liang and Sudarshan S Chawathe. 2004. Privacy-preserving inter-database operations. In International Conference on Intelligence and Security Informatics. Springer, 66–82

[10] Monica Scannapieco, Ilya Figotin, Elisa Bertino, and Ahmed K. Elmagarmid. 2007. Privacy Preserving Schema and Data Matching. (SIGMOD'07). ACM, New York, NY, USA, 653–664.

[11] Liu Y, Chen T, Yang Q. Secure Federated Transfer Learning[J]. arXiv preprint arXiv:1812.03337, 2018.

[12] Acar, A.; Aksu, H.; Uluagac, A. S.; and Conti, M. 2018. A survey on homomorphic encryption schemes: Theory and implementation. ACM Comput. Surv.51(4):79:1–79:35.