(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115392480 A
(43)申请公布日 2022.11.25

(21)申请号 202210940246.1

(22)申请日 2022.08.05

(71)申请人 北京富算科技有限公司
地址 100020 北京市朝阳区东三环中路9号19层2201

(72)发明人 尤志强 卞阳 陈立峰

(74)专利代理机构 上海弼兴律师事务所 31283
专利代理师 罗朗 林嵩

(51)Int.Cl.
G06N 20/00(2019.01)
G06K 9/62(2022.01)

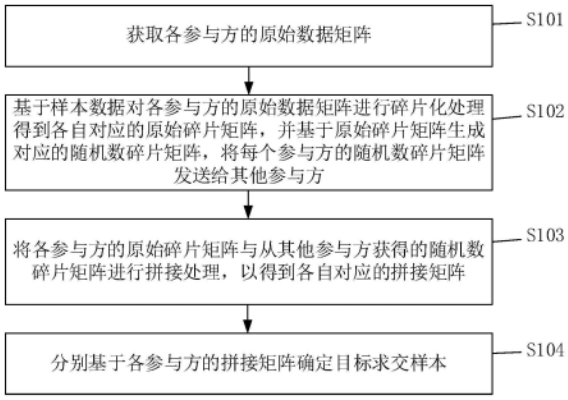
权利要求书4页 说明书25页 附图9页

(54)发明名称

安全求交、联邦学习模型的训练方法及系统、设备及介质

(57)摘要

本发明公开了一种数据安全求交、联邦学习模型的训练方法及系统、设备及介质，其中，安全求交方法包括获取各参与方的原始数据矩阵，对各个参与方的原始数据矩阵进行碎片化处理后进行拼接，然后通过密态排序以及密态对齐，生成碎片态的求交样本，由于求交样本为碎片态，可以保证交集结果不泄露，从而安全求交的全流程中不暴露任何敏感信息，既保护交集以外的信息，同时输出的结果又可以保护交集信息，进而能够执行高标准的安全要求和实现保护敏感数据的目标。



CN 115392480 A

1.一种数据共享中的安全求交方法,其特征在于,应用于至少两个参与方之间数据共享场景中,所述安全求交方法包括:

获取各所述参与方的原始数据矩阵;

其中,所述原始数据矩阵包括至少一组样本数据,所述样本数据包括用于标识所述参与方中每个对象的唯一标识和与所述唯一标识相对应的属性数据;

基于所述样本数据对各所述参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵,并基于所述原始碎片矩阵生成对应的随机数碎片矩阵,将每个参与方的所述随机数碎片矩阵发送给其他参与方;

将各所述参与方的原始碎片矩阵与从其他参与方获得的所述随机数碎片矩阵进行拼接处理,以得到各自对应的拼接矩阵;

分别基于各所述参与方的拼接矩阵确定目标求交样本。

2.如权利要求1所述的数据共享中的安全求交方法,其特征在于,所述基于各所述参与方的拼接矩阵确定目标求交样本包括:

分别基于各所述参与方的所述拼接矩阵进行排序,以得到与所述拼接矩阵对应的排序矩阵;

分别基于各所述参与方的所述排序矩阵进行样本特征对齐计算,以确定所述目标求交样本。

3.如权利要求2所述的数据共享中的安全求交方法,其特征在于,在得到各自对应的原始碎片矩阵之后,所述安全求交方法还包括:

基于预设密态打乱算法对所述原始碎片矩阵中若干列进行密态打乱,以得到样本顺序变换后的新的原始碎片矩阵;

其中,不同所述参与者对应的所述原始碎片矩阵采用相同的所述预设密态打乱算法进行密态打乱处理。

4.如权利要求2所述的数据共享中的安全求交方法,其特征在于,所述原始碎片矩阵包括样本数据碎片,所述样本数据碎片包括唯一标识碎片和与所述唯一标识碎片对应的属性数据碎片;

所述分别基于各所述参与方的所述拼接矩阵进行排序,以得到各自对应的排序矩阵包括:

分别基于预设排序算子提取各所述参与方的所述拼接矩阵中的所述唯一标识碎片相同的样本数据碎片并进行排序,以得到各自对应的所述排序矩阵。

5.如权利要求2所述的数据共享中的安全求交方法,其特征在于,所述分别基于各所述参与方的所述排序矩阵进行样本特征对齐计算,以确定所述目标求交样本包括:

分别基于各所述参与方的所述排序矩阵依次比较相邻的所述样本数据碎片对应的所述唯一标识碎片是否相同,以根据比较结果进行样本特征对齐计算,得到所述目标求交样本。

6.如权利要求5所述的数据共享中的安全求交方法,其特征在于,所述以根据比较结果进行样本特征对齐计算包括:

根据预设转换算子将碎片化的所述比较结果转化为对应的第一比较值或第二比较值;

将相邻的所述样本数据碎片中的对应属性数据碎片进行密态求和,并将各个求和值依

次与所述第一比较值或所述第二比较值相乘，得到所述目标求交样本。

7.如权利要求6所述的数据共享中的安全求交方法，其特征在于，所述根据预设转换算子将碎片化的所述比较结果转化为对应的第一比较值或第二比较值包括：

在所述比较结果相同时，基于B2A算子将碎片化的所述比较结果转化为算术类型的第一比较值；

在所述比较结果不相同时，基于所述B2A算子将碎片化的所述比较结果转化为算术类型的第二比较值。

8.如权利要求6所述的数据共享中的安全求交方法，其特征在于，所述根据比较结果进行样本特征对齐计算还包括：

在所述比较结果相同时，并将相邻的所述样本数据碎片中对应的所述属性数据碎片进行密态求和；在所述比较结果不相同时，丢弃排序位置靠前的样本数据，以得到新的目标求交样本。

9.如权利要求6所述的数据共享中的安全求交方法，其特征在于，在所述根据比较结果进行样本特征对齐计算之后，所述安全求交方法还包括：

将碎片化的所述比较结果进行恢复处理。

10.如权利要求1所述的数据共享中的安全求交方法，其特征在于，在所述获取各所述参与方的原始数据矩阵之后，所述安全求交方法还包括：

判断各个所述参与方的原始数据矩阵中属性数据对应的列数是否相同，若不相同，则根据预设补齐规则生成虚拟属性数据列进行补齐，以得到补齐后的所述原始数据矩阵；

其中，各所述参与方所对应补齐后的所述原始数据矩阵的列数相等。

11.如权利要求1所述的数据共享中的安全求交方法，其特征在于，所述基于所述样本数据对各所述参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵，并基于所述原始碎片矩阵生成对应的随机数碎片矩阵包括：

基于所述样本数据对各所述参与方的原始数据矩阵中的每一个原始数据都减去一个随机数，以得到差值碎片和随机数碎片，将所有的差值碎片作为原始碎片矩阵，将所有的随机数碎片作为随机数碎片矩阵。

12.如权利要求4所述的安全求交方法，其特征在于，所述预设排序算子基于快速排序算法或排序网络算法中实现；

和/或，

所述排序网络算法基于双调排序算法实现。

13.如权利要求1-11中任一项所述的数据共享中的安全求交方法，其特征在于，所述安全求交方法还包括：

判断所述原始数据矩阵中的唯一标识的类型；

若所述唯一标识为字符串型，则将所述字符串的唯一标识进行数值化处理以得到数值化的唯一标识；

若所述唯一标识为数值型，则不进行操作。

14.一种联邦学习模型的训练方法，其特征在于，所述训练方法包括：

获取各参与方利用如权利要求1-13中任一项所述安全求交方法得到的碎片化的目标求交样本；

基于预设划分策略获取各所述参与方对所述目标求交样本执行划分后得到的训练集碎片和测试集碎片；

获取各所述参与方利用各自的所述训练集碎片、所述测试集碎片通过安全多方计算算子进行特征与权重参数计算得到的预测碎片；

获取各所述参与方利用各自的所述预测碎片通过所述安全多方计算算子进行梯度计算得到的梯度碎片；

获取各所述参与方利用各自的所述梯度碎片通过所述安全多方计算算子进行更新权重系数计算，以更新初始权重碎片得到新的权重碎片，并利用新的权重碎片进行迭代；

在所述权重碎片满足预设条件时则获取目标权重碎片，并利用所述目标权重碎片建立所述联邦学习模型。

15.如权利要求14所述的联邦学习模型的训练方法，其特征在于，在得到所述预测碎片之后，所述训练方法还包括：

获取各所述参与方基于各自的所述预测碎片通过所述安全多方计算算子进行损失值计算得到损失值碎片；

任一所述参与方接收其他所述参与方发送的所述损失值碎片，并将所有的损失值碎片恢复至对应的明文后上报训练日志。

16.如权利要求14或15所述的联邦学习模型的训练方法，其特征在于，在得到梯度碎片之后，所述训练方法还包括：

分别通过所述安全多方计算算子判断各所述参与方的特征对应的所述梯度碎片的梯度值是否小于预设阈值，若是则任一所述参与方接收其他所述参与方发送的比较结果碎片，并将所述比较结果碎片恢复至对应的明文。

17.如权利要求14所述的联邦学习模型的训练方法，其特征在于，所述训练方法还包括：

判断训练状态是否为终止训练；

若是，则输出并根据使用需求保存模型参数为对应的模型参数明文或模型参数碎片；

若否，则执行对所述梯度碎片通过所述安全多方计算算子进行梯度更新权重系数计算得到新的所述目标权重碎片。

18.一种数据共享中的安全求交系统，其特征在于，应用于至少两个参与方之间数据共享场景中，所述安全求交系统包括：

获取模块，用于获取各所述参与方的原始数据矩阵；

其中，所述原始数据矩阵包括至少一组样本数据，所述样本数据包括用于标识所述参与方中每个对象的唯一标识和与所述唯一标识相对应的属性数据；

碎片化模块，用于基于所述样本数据对各所述参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵，并基于所述原始碎片矩阵生成对应的随机数碎片矩阵，将每个参与方的所述随机数碎片矩阵发送给其他参与方；

拼接模块，用于将各所述参与方的原始碎片矩阵与从其他参与方获得的所述随机数碎片矩阵进行拼接处理，以得到各自对应的拼接矩阵；

求交样本确定模块，用于分别基于各所述参与方的拼接矩阵确定目标求交样本。

19.一种联邦学习模型的训练系统，其特征在于，所述联邦学习模型的训练系统包括：

求交样本获取模块，用于获取各所述参与方利用如权利要求18所述的安全求交系统得到的碎片化的目标求交样本；

划分模块，用于基于预设划分策略获取各所述参与方对所述目标求交样本执行划分后得到的训练集碎片和测试集碎片；

预测碎片计算模块，用于获取各所述参与方利用各自的所述训练集碎片、所述测试集碎片通过安全多方计算算子进行特征与权重参数计算得到的预测碎片；

梯度碎片计算模块，用于获取各所述参与方利用各自的所述预测碎片通过所述安全多方计算算子进行梯度计算得到的梯度碎片；

权重碎片更新模块，用于获取各所述参与方利用各自的所述梯度碎片通过所述安全多方计算算子进行更新权重系数计算，以更新初始权重碎片得到新的权重碎片，并利用新的权重碎片进行迭代；

模型建立模块，用于在所述权重碎片满足预设条件时则获取目标权重碎片，并利用所述目标权重碎片建立所述联邦学习模型。

20.一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，其特征在于，所述处理器执行计算机程序时实现如权利要求1-13中任一项所述的数据共享中的安全求交方法；或，实现如权利要求14-17中任一项所述的联邦学习模型的训练方法。

21.一种计算机可读存储介质，其上存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现如权利要求1-13中任一项所述的数据共享中的安全求交方法；或，实现如权利要求14-17中任一项所述的联邦学习模型的训练方法。

安全求交、联邦学习模型的训练方法及系统、设备及介质

技术领域

[0001] 本发明涉及安全多方计算的技术领域，特别涉及一种数据安全求交、联邦学习模型的训练方法及系统、设备及介质。

背景技术

[0002] 随着人工智能技术的发展，人们为解决数据孤岛的问题，提出了“联邦学习”的概念，联邦学习本质上是一种分布式机器学习框架，其做到了在保障数据隐私安全及合法合规的基础上，实现数据共享，共同建模。它的核心思想是在多个数据源共同参与模型训练时，不需要进行原始数据流转的前提下，仅通过交互模型中间参数进行模型联合训练，原始数据可以不出本地。这种方式实现数据隐私保护和数据共享分析的平衡，即“数据可用不可见”的数据应用模式。联邦学习中的发起方和参与方作为成员方，在不用给出己方数据的情况下，也可进行模型训练得到模型参数，并且可以避免数据隐私泄露的问题。由于联邦学习过程需要大量的数据来支持，而数据又大都分布于不同的数据持有方，所以需要联合各个数据持有方来进行模型构建。

[0003] 其中，纵向联邦学习是在参与者的数据特征重叠较小，而用户重叠较多的情况下，取出参与者用户相同而用户数据特征不同的那部分用户及数据进行联合训练机器学习模型。然而，在纵向联邦学习过程中，现有的算法实现，普遍仅关注了模型本身的安全性保护，比如通过使用同态加密等手段保护模型训练过程中通信交互的数据，然而却忽视了上游安全求交(PSI)阶段的安全性保护，造成敏感信息的泄漏，并且在目前业内的实现方案中，例如，对齐阶段采用的PSI协议，主流的有基于Diffie-Hellmann(密钥协商)的PSI方案、基于RSA(一种加密算法)盲签名的PSI方案、基于OPRF(不经意伪随机函数)的PSI、基于布隆过滤器&OT的方案等。在纵向联邦场景，这些协议都会存在这种互相暴露交集用户的敏感信息的现象。另外，在纵向联邦学习中，如果不进行交集共享，传统的算法设计，无法有效进行联合建模。

发明内容

[0004] 本发明要解决的技术问题是为了克服现有技术中无法避免求交阶段的容易暴露用户的敏感信息的情况发生，进而无法保证信息的安全性和隐私性的缺陷，提供一种数据安全求交、联邦学习模型的训练方法及系统、设备及介质。

[0005] 本发明是通过下述技术方案来解决上述技术问题：

[0006] 本发明提供一种数据共享中的安全求交方法，应用于至少两个参与方之间数据共享场景中，所述安全求交方法包括：

[0007] 获取各所述参与方的原始数据矩阵；

[0008] 其中，所述原始数据矩阵包括至少一组样本数据，所述样本数据包括用于标识所述参与方中每个对象的唯一标识和与所述唯一标识相对应的属性数据；

[0009] 基于所述样本数据对各所述参与方的原始数据矩阵进行碎片化处理得到各自对

应的原始碎片矩阵，并基于所述原始碎片矩阵生成对应的随机数碎片矩阵，将每个参与方的所述随机数碎片矩阵发送给其他参与方；

[0010] 将各所述参与方的原始碎片矩阵与从其他参与方获得的所述随机数碎片矩阵进行拼接处理，以得到各自对应的拼接矩阵；

[0011] 分别基于各所述参与方的拼接矩阵确定目标求交样本。

[0012] 较佳地，所述基于各所述参与方的拼接矩阵确定目标求交样本包括：

[0013] 分别基于各所述参与方的所述拼接矩阵进行排序，以得到与所述拼接矩阵对应的排序矩阵；

[0014] 分别基于各所述参与方的所述排序矩阵进行样本特征对齐计算，以确定所述目标求交样本。

[0015] 较佳地，在得到各自对应的原始碎片矩阵之后，所述安全求交方法还包括：

[0016] 基于预设密态打乱算法对所述原始碎片矩阵中若干列进行密态打乱，以得到样本顺序变换后的新的原始碎片矩阵；

[0017] 其中，不同所述参与者对应的所述原始碎片矩阵采用相同的所述预设密态打乱算法进行密态打乱处理。

[0018] 较佳地，所述原始碎片矩阵包括样本数据碎片，所述样本数据碎片包括唯一标识碎片和与所述唯一标识碎片对应的属性数据碎片；

[0019] 所述分别基于各所述参与方的所述拼接矩阵进行排序，以得到各自对应的排序矩阵包括：

[0020] 分别基于预设排序算子提取各所述参与方的所述拼接矩阵中的所述唯一标识碎片相同的样本数据碎片并进行排序，以得到各自对应的所述排序矩阵。

[0021] 较佳地，所述分别基于各所述参与方的所述排序矩阵进行样本特征对齐计算，以确定所述目标求交样本包括：

[0022] 分别基于各所述参与方的所述排序矩阵依次比较相邻的所述样本数据碎片对应的所述唯一标识碎片是否相同，以根据比较结果进行样本特征对齐计算，得到所述目标求交样本。

[0023] 较佳地，所述以根据比较结果进行样本特征对齐计算包括：

[0024] 根据预设转换算子将碎片化的所述比较结果转化为对应的第一比较值或第二比较值；

[0025] 将相邻的所述样本数据碎片中的对应属性数据碎片进行密态求和，并将各个求和值依次与所述第一比较值或所述第二比较值相乘，得到所述目标求交样本。

[0026] 较佳地，所述根据预设转换算子将碎片化的所述比较结果转化为对应的第一比较值或第二比较值包括：

[0027] 在所述比较结果相同时，基于B2A算子将碎片化的所述比较结果转化为算术类型的第一比较值；

[0028] 在所述比较结果不相同时，基于所述B2A算子将碎片化的所述比较结果转化为算术类型的第二比较值。

[0029] 较佳地，所述根据比较结果进行样本特征对齐计算的步骤包括：

[0030] 在所述比较结果相同时，并将相邻的所述样本数据碎片中对应的所述属性数据碎

片进行密态求和;在所述比较结果不相同时,丢弃排序位置靠前的样本数据,以得到新的目标求交样本。

[0031] 较佳地,在所述根据比较结果进行样本特征对齐计算之后,所述安全求交方法还包括:

[0032] 将碎片化的所述比较结果进行恢复处理。

[0033] 较佳地,在所述获取各所述参与方的原始数据矩阵之后,所述安全求交方法还包括:

[0034] 判断各个所述参与方的原始数据矩阵中属性数据对应的列数是否相同,若不相同,则根据预设补齐规则生成虚拟属性数据列进行补齐,以得到补齐后的所述原始数据矩阵;

[0035] 其中,各所述参与方所对应补齐后的所述原始数据矩阵的列数相等。

[0036] 较佳地,所述基于所述样本数据对各所述参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵,并基于所述原始碎片矩阵生成对应的随机数碎片矩阵包括:

[0037] 基于所述样本数据对各所述参与方的原始数据矩阵中的每一个原始数据都减去一个随机数,以得到差值碎片和随机数碎片,将所有的差值碎片作为原始碎片矩阵,将所有的随机数碎片作为随机数碎片矩阵。

[0038] 较佳地,所述预设排序算子基于快速排序算法或排序网络算法中实现;

[0039] 和/或,

[0040] 所述排序网络算法基于双调排序算法实现。

[0041] 较佳地,所述安全求交方法还包括:

[0042] 判断所述原始数据矩阵中的唯一标识的类型;

[0043] 若所述唯一标识为字符串型,则将所述字符串的唯一标识进行数值化处理以得到数值化的唯一标识;

[0044] 若所述唯一标识为数值型,则不进行操作。

[0045] 本发明还提供一种联邦学习模型的训练方法,所述联邦学习模型的训练方法包括:

[0046] 获取各参与方利用如上所述安全求交方法得到的碎片化的目标求交样本;

[0047] 基于预设划分策略获取各所述参与方对所述目标求交样本执行划分后得到的训练集碎片和测试集碎片;

[0048] 获取各所述参与方利用各自的所述训练集碎片、所述测试集碎片通过安全多方计算算子进行特征与权重参数计算得到的预测碎片;

[0049] 获取各所述参与方利用各自的所述预测碎片通过所述安全多方计算算子进行梯度计算得到的梯度碎片;

[0050] 获取各所述参与方利用各自的所述梯度碎片通过所述安全多方计算算子进行更新权重系数计算,以更新初始权重碎片得到新的权重碎片,并利用新的权重碎片进行迭代;

[0051] 在所述权重碎片满足预设条件时则获取目标权重碎片,并利用所述目标权重碎片建立所述联邦学习模型。

[0052] 较佳地,在得到所述预测碎片之后,所述联邦学习模型的训练方法还包括:

[0053] 获取各所述参与方基于各自的所述预测碎片通过所述安全多方计算算子进行损失值计算得到损失值碎片；

[0054] 任一所述参与方接收其他所述参与方发送的所述损失值碎片，并将所有的损失值碎片恢复至对应的明文后上报训练日志。

[0055] 较佳地，在得到梯度碎片之后，所述联邦学习模型的训练方法还包括：

[0056] 分别通过所述安全多方计算算子判断各所述参与方的特征对应的所述梯度碎片的梯度值是否小于预设阈值，若是则任一所述参与方接收其他所述参与方发送的比较结果碎片，并将所述比较结果碎片恢复至对应的明文。

[0057] 较佳地，所述联邦学习模型的训练方法还包括：

[0058] 判断训练状态是否为终止训练；

[0059] 若是，则输出并根据使用需求保存模型参数为对应的模型参数明文或模型参数碎片；

[0060] 若否，则执行对所述梯度碎片通过所述安全多方计算算子进行梯度更新权重系数计算得到新的所述目标权重碎片。

[0061] 本发明还提供一种数据共享中的安全求交系统，应用于至少两个参与方之间数据共享场景中，所述安全求交系统包括：

[0062] 获取模块，用于获取各所述参与方的原始数据矩阵；

[0063] 其中，所述原始数据矩阵包括至少一组样本数据，所述样本数据包括用于标识所述参与方中每个对象的唯一标识和与所述唯一标识相对应的属性数据；

[0064] 碎片化模块，用于基于所述样本数据对各所述参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵，并基于所述原始碎片矩阵生成对应的随机数碎片矩阵，将每个参与方的所述随机数碎片矩阵发送给其他参与方；

[0065] 拼接模块，用于将各所述参与方的原始碎片矩阵与从其他参与方获得的所述随机数碎片矩阵进行拼接处理，以得到各自对应的拼接矩阵；

[0066] 求交样本确定模块，用于分别基于各所述参与方的拼接矩阵确定目标求交样本。

[0067] 较佳地，所述求交样本确定模块包括：

[0068] 排序单元，用于分别基于各所述参与方的所述拼接矩阵进行排序，以得到与所述拼接矩阵对应的排序矩阵；

[0069] 对齐单元，用于分别基于各所述参与方的所述排序矩阵进行样本特征对齐计算，以确定所述目标求交样本。

[0070] 较佳地，所述安全求交系统还包括：

[0071] 打乱模块，用于基于预设密态打乱算法对所述原始碎片矩阵中若干列进行密态打乱，以得到样本顺序变换后的新的原始碎片矩阵；

[0072] 其中，不同所述参与者对应的所述原始碎片矩阵采用相同的所述预设密态打乱算法进行密态打乱处理。

[0073] 较佳地，所述原始碎片矩阵包括样本数据碎片，所述样本数据碎片包括唯一标识碎片和与所述唯一标识碎片对应的属性数据碎片；

[0074] 所述排序单元，还用于分别基于预设排序算子提取各所述参与方的所述拼接矩阵中的所述唯一标识碎片相同的样本数据碎片并进行排序，以得到各自对应的所述排序矩

阵。

[0075] 较佳地，所述对齐单元，还用于分别基于各所述参与方的所述排序矩阵依次比较相邻的所述样本数据碎片对应的所述唯一标识碎片是否相同，以根据比较结果进行样本特征对齐计算，得到所述目标求交样本。

[0076] 较佳地，所述对齐单元，还用于根据预设转换算子将碎片化的所述比较结果转化为对应的第一比较值或第二比较值；

[0077] 将相邻的所述样本数据碎片中的对应属性数据碎片进行密态求和，并将各个求和值依次与所述第一比较值或所述第二比较值相乘，得到所述目标求交样本。

[0078] 较佳地，所述对齐单元，还用于在所述比较结果相同时，基于B2A算子将碎片化的所述比较结果转化为算术类型的第一比较值；

[0079] 在所述比较结果不相同时，基于所述B2A算子将碎片化的所述比较结果转化为算术类型的第二比较值。

[0080] 较佳地，所述对齐单元，还用于在所述比较结果相同时，并将相邻的所述样本数据碎片中对应的所述属性数据碎片进行密态求和；在所述比较结果不相同时，丢弃排序位置靠前的样本数据，以得到新的目标求交样本。

[0081] 较佳地，所述安全求交系统还包括：

[0082] 恢复模块，用于将碎片化的所述比较结果进行恢复处理。

[0083] 较佳地，所述安全求交系统还包括：

[0084] 补齐模块，用于判断各个所述参与方的原始数据矩阵中属性数据对应的列数是否相同，若不相同，则根据预设补齐规则生成虚拟属性数据列进行补齐，以得到补齐后的所述原始数据矩阵；

[0085] 其中，各所述参与方所对应补齐后的所述原始数据矩阵的列数相等。

[0086] 较佳地，所述碎片化模块，还用于基于所述样本数据对各所述参与方的原始数据矩阵中的每一个原始数据都减去一个随机数，以得到差值碎片和随机数碎片，将所有的差值碎片作为原始碎片矩阵，将所有的随机数碎片作为随机数碎片矩阵。

[0087] 较佳地，所述预设排序算子基于快速排序算法或排序网络算法中实现；

[0088] 和/或，

[0089] 所述排序网络算法基于双调排序算法实现。

[0090] 较佳地，所述安全求交系统还包括：

[0091] 类型转换模块，用于判断所述原始数据矩阵中的唯一标识的类型；

[0092] 若所述唯一标识为字符串型，则将所述字符串的唯一标识进行数值化处理以得到数值化的唯一标识；

[0093] 若所述唯一标识为数值型，则不进行操作。

[0094] 本发明还提供一种联邦学习模型的训练系统，所述训练系统包括：

[0095] 求交样本获取模块，用于获取各所述参与方利用如所述安全求交方法得到的碎片化的目标求交样本；

[0096] 划分模块，用于基于预设划分策略获取各所述参与方对所述目标求交样本执行划分后得到的训练集碎片和测试集碎片；

[0097] 预测碎片计算模块，用于获取各所述参与方利用各自的所述训练集碎片、所述测

试集碎片通过安全多方计算算子进行特征与权重参数计算得到的预测碎片；

[0098] 梯度碎片计算模块，用于获取各所述参与方利用各自的所述预测碎片通过所述安全多方计算算子进行梯度计算得到的梯度碎片；

[0099] 权重碎片更新模块，用于获取各所述参与方利用各自的所述梯度碎片通过所述安全多方计算算子进行更新权重系数计算，以更新初始权重碎片得到新的权重碎片，并利用新的权重碎片进行迭代；

[0100] 模型建立模块，用于在所述权重碎片满足预设条件时则获取目标权重碎片，并利用所述目标权重碎片建立所述联邦学习模型。

[0101] 较佳地，所述训练系统还包括：

[0102] 损失值碎片计算模块，用于获取各所述参与方基于各自的所述预测碎片通过所述安全多方计算算子进行损失值计算得到损失值碎片；

[0103] 任一所述参与方接收其他所述参与方发送的所述损失值碎片，并将所有的损失值碎片恢复至对应的明文后上报训练日志。

[0104] 较佳地，所述训练系统还包括：

[0105] 梯度碎片比较模块，用于分别通过所述安全多方计算算子判断各所述参与方的特征对应的所述梯度碎片的梯度值是否小于预设阈值，若是则任一所述参与方接收其他所述参与方发送的比较结果碎片，并将所述比较结果碎片恢复至对应的明文。

[0106] 较佳地，所述训练系统还包括：

[0107] 训练状态判断模块，用于判断训练状态是否为终止训练；

[0108] 若是，则输出并根据使用需求保存模型参数为对应的模型参数明文或模型参数碎片；

[0109] 若否，则执行对所述梯度碎片通过所述安全多方计算算子进行梯度更新权重系数计算得到新的所述目标权重碎片。

[0110] 本发明还提供一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行计算机程序时实现如上所述的数据共享中的安全求交方法和/或实现如上所述的联邦学习模型的训练方法。

[0111] 本发明还提供一种计算机可读存储介质，其上存储有计算机程序，所述计算机程序被处理器执行时实现如上所述的数据共享中的安全求交方法和/或实现如上所述的联邦学习模型的训练方法。

[0112] 本发明的积极进步效果在于：通过获取各参与方的原始数据矩阵，对各个参与方的原始数据矩阵进行本地碎片化处理后进行拼接，然后通过密态排序以及密态对齐，生成碎片态的求交样本，由于求交样本为碎片态，可以保证交集结果不泄露，从而安全求交的全流程中不暴露任何敏感信息，既保护交集以外的信息，同时输出的结果又可以保护交集信息，进而能够执行高标准的安全要求和实现保护敏感数据的目标。

附图说明

[0113] 图1为本发明实施例1提供的第一流程示意图。

[0114] 图2为本发明实施例1提供的第二流程示意图。

[0115] 图3为本发明实施例1提供的第三流程示意图。

[0116] 图4为本发明实施例1提供的一种安全求交流程图。

[0117] 图5为本发明实施例1提供的另一种安全求交流程图。

[0118] 图6为本发明实施例2提供的系统结构示意图。

[0119] 图7为本发明实施例3提供的第一流程示意图。

[0120] 图8为本发明实施例3提供的一种联邦学习模型的训练方法的流程图。

[0121] 图9为本发明实施例4提供的系统结构示意图。

[0122] 图10为本发明实施例5提供的电子设备的结构示意图。

具体实施方式

[0123] 下面通过实施例的方式进一步说明本发明，但并不因此将本发明限制在所述的实施例范围之中。

[0124] 为了更清楚地说明本说明书实施例的技术方案，下面将对实施例描述中所需要使用的附图作简单的介绍。显而易见地，下面描述中的附图仅仅是本说明书的一些示例或实施例，对于本领域的普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图将本说明书应用于其它类似情景。除非从语言环境中显而易见或另做说明，图中相同标号代表相同结构或操作。

[0125] 应当理解，本文使用的“系统”、“装置”、“单元”和/或“模组”是用于区分不同级别的不同组件、元件、部件、部分或装配的一种方法。然而，如果其他词语可实现相同的目的，则可通过其他表达来替换所述词语。

[0126] 如本说明书中所示，除非上下文明确提示例外情形，“一”、“一个”、“一种”和/或“该”等词并非特指单数，也可包括复数。一般说来，术语“包括”与“包含”仅提示包括已明确标识的步骤和元素，而这些步骤和元素不构成一个排它性的罗列，方法或者设备也可能包含其它的步骤或元素。

[0127] 本说明书中使用了流程图用来说明根据本说明书的实施例的系统所执行的操作。应当理解的是，前面或后面操作不一定按照顺序来精确地执行。相反，可以按照倒序或同时处理各个步骤。同时，也可以将其他操作添加到这些过程中，或从这些过程移除某一步或数步操作。

[0128] 安全求交(PSI)，也称为隐私保护集合交集协议，是纵向联邦学习不可或缺的一部分，PSI协议允许持有各自集合的两方来共同计算两个集合的交集运算。在协议交互的最后，两方应该得到正确的交集，而且不会得到交集以外另一方集合中的任何信息。从PSI定义看，虽然该协议对于参与方，仅提供了交集部分的用户样本交集信息，但该信息对于很多敏感场景或者安全性要求高的机构，是不能接受的，比如机构A有数据集Da，机构B有数据集Db，通过安全求交之后，A和B机构同时得到了Da和Db的交集Dc，那么A机构就知道Dc中的用户是属于B机构的，且由于是交集用户，A机构知道这交集部分的用户id、手机、身份证等等敏感信息。同理B机构也可以知道Dc是属于A机构的用户，同样可以获得非常多的信息。在目前业内的实现方案中，安全求交普遍存在这种互相暴露交集用户的敏感信息的现象。因此，各大银行、运行商等机构对全匿踪的纵向联邦学习需求非常强烈，希望全匿踪的纵向联邦学习框架，在安全求交部分，可以在保护交集以外信息不被泄漏的同时，对交集也起到保护作用，参与方无法知晓真正的交集信息。

[0129] 基于上述原因，如图1所示，本实施例提供一种数据共享中的安全求交方法，应用于至少两个参与方之间数据共享场景中，在本实施例中，以参与方为两个为例进行说明，如一方为P_0方(即GUEST)，另一方则为P_1方(即HOST)，本实施例的安全求交方法包括：

[0130] S101、获取各参与方的原始数据矩阵。

[0131] 其中，原始数据矩阵包括至少一组样本数据，样本数据包括用于标识参与方中每个对象的唯一标识和与唯一标识相对应的属性数据。

[0132] 作为可选地实施方式，本实施例的安全求交方法还包括：

[0133] S101a、判断原始数据矩阵中的唯一标识的类型。

[0134] 若唯一标识为字符串型，则执行步骤S101a1，若唯一标识为数值型，则执行步骤S101a2。

[0135] 步骤S101a1、将字符串的唯一标识进行数值化处理以得到数值化的唯一标识，步骤S101a2、不进行操作。

[0136] 需要说明的是，在求交阶段，需要指定某列作为求交的对象列，比如采用身份证号、手机号等作为判断是否为同一用户的依据。因此求交的对象列必须具备唯一性要求。比如下表采用的是id列，该列值为字符串，因此首先需要做数值化，本实施方式可以采用hash数值化，将字符串id值数值化，方便后续步骤的执行，数值化需要满足映射后的唯一性要求，即同一字符串id映射后也是唯一的数值，不能存在交叉重复。

[0137] 在一个实施方式中各个参与方可以通过商讨确定原始数据矩阵的组成规范，从而可以在获取原始数据矩阵之后可以直接进行下一步，例如，各方都具有相同数量的特征数量，如同具有一个唯一标识、两个特征数据和一个标签数据，本实施例对此不作限定。

[0138] 而在另一个实施方式中，当各个参与方没有规定原始数据矩阵的组成时，如下两表分别示出了P_0方和P_1方的原始特征数据。

[0139] P_0方(GUEST)原始特征数据D_0：

[0140]

id	Y	X_{a1}	X_{a2}	X_{a3}
224361	0	2.3	5.5	4
428493	1	3.6	3.5	-3
672684	0	-1.2	0.5	1.2
1030916	1	0.8	0.13	3

[0141] P_1方(HOST)原始特征数据D_1：

[0142]

id	X_{b1}	X_{b2}
858329	0.77	1.45
428493	2.6	3.9
1030916	-2.2	-1.1

[0143] 从而在步骤S101之后，本实施例的安全求交方法还包括：

[0144] S101b、判断各个参与方的原始数据矩阵中属性数据对应的列数是否相同，若不相同，则根据预设补齐规则生成虚拟属性数据列进行补齐，以得到补齐后的原始数据矩阵。

[0145] 其中，各参与方所对应补齐后的原始数据矩阵的列数相等。

[0146] 也即对原始数据矩阵非id域补齐，各参与方需要对各自的样本数据进行补齐操作，该阶段，各方会进行同步特征数量，然后按照预设补齐规则，生成虚拟标签列和虚拟标

签列。预设补齐规则可以包括在虚拟标签列和虚拟标签列的填充值0，也可以是某种规则生成的值，例如在P_0方同时加1，在P_1方同时减1，只需要保证该信息可以被后续操作进行去除即可，本实施方式对此不作限定，本实施方式采用的是0填充，通过对P_0方和P_1方的原始数据进行增广补齐之后，就得到同样列大小的矩阵，如下表的D'_0和D'_1所示。

[0147] P_0方特征矩阵根据P_1特征数及标签进行补齐，得到D'_0：

[0148]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
224361	0	2.3	5.5	4	0	0
428493	1	3.6	3.5	-3	0	0
672684	0	-1.2	0.5	1.2	0	0
1030916	1	0.8	0.13	3	0	0

[0149] 同理，P_1方特征矩阵根据P_0特征数进行补齐，得到D'_1：

[0150]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
858329	0	0	0	0	0.77	1.45
428493	0	0	0	0	2.6	3.9
1030916	0	0	0	0	-2.2	-1.1

[0151] S102、基于样本数据对各参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵，并基于原始碎片矩阵生成对应的随机数碎片矩阵，将每个参与方的随机数碎片矩阵发送给其他参与方。

[0152] 需要说明的是，参与方还可以将的随机数碎片矩阵通过秘密共享给其他参与方。

[0153] 可选地，本实施例步骤S102包括：

[0154] S1021、基于样本数据对各参与方的原始数据矩阵中的每一个原始数据都减去一个随机数，以得到差值碎片和随机数碎片，将所有的差值碎片作为原始碎片矩阵，将所有的随机数碎片作为随机数碎片矩阵。

[0155] 碎片化公式如下：

[0156] 对x进行碎片化$Shr_A^i(x)$：

[0157] P_i选择$r \in_R Z_{2^l}$，使$\langle x\rangle_A^i = x - r$，并且发送r给$P_{1-i}$，使$\langle x\rangle_A^{1-i} = r$，

[0158] 其中，r为随机数。

[0159] 在本实施例中分别对P_0和P_1进行碎片化处理，如下表的所示：

[0160] P_0方对D'_0进行本地碎片化并秘密共享(id也需要碎片化)

[0161]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<224361>	<0>	<2.3>	<5.5>	<4>	<0>	<0>
<428493>	<1>	<3.6>	<3.5>	<-3>	<0>	<0>
<672684>	<0>	<-1.2>	<0.5>	<1.2>	<0>	<0>
<1030916>	<1>	<0.8>	<0.13>	<3>	<0>	<0>

[0162] 同理，P_1方对D'_1进行本地碎片化并秘密共享

[0163]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<858329>	<0>	<0>	<0>	<0>	<0.77>	<1.45>

<428493>	<0>	<0>	<0>	<0>	<2.6>	<3.9>
<1030916>	<0>	<0>	<0>	<0>	<-2.2>	<-1.1>

[0164] 注:<x>表示x的碎片态。

[0165] S103、将各参与方的原始碎片矩阵与从其他参与方获得的随机数碎片矩阵进行拼接处理,以得到各自对应的拼接矩阵。如下表所示D_f:

[0166] MPC Concat执行得到拼接后的拼接矩阵D_f

[0167]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<224361>	<0>	<2.3>	<5.5>	<4>	<0>	<0>
<428493>	<1>	<3.6>	<3.5>	<-3>	<0>	<0>
<672684>	<0>	<-1.2>	<0.5>	<1.2>	<0>	<0>
<1030916>	<1>	<0.8>	<0.13>	<3>	<0>	<0>
<858329>	<0>	<0>	<0>	<0>	<0.77>	<1.45>
<428493>	<0>	<0>	<0>	<0>	<2.6>	<3.9>
<1030916>	<0>	<0>	<0>	<0>	<-2.2>	<-1.1>

[0168] S104、分别基于各参与方的拼接矩阵确定目标求交样本。

[0169] 需要说明的是,可以通过数学方式从拼接矩阵选择出唯一标识相同的样本数据,进而可以确定目标求交样本。

[0170] 在一种可选的实施方式中,如图2所示,本实施例的步骤S104包括:

[0171] S1041、分别基于各参与方的拼接矩阵进行排序,以得到与拼接矩阵对应的排序矩阵。

[0172] S1042、分别基于各参与方的排序矩阵进行样本特征对齐计算,以确定目标求交样本。

[0173] 需要说明的是,对于该目标求交样本,由于不进行恢复明文态,保持全碎片态,因此双方都不知道交集具体是哪些,因此具备强安全性,也不存在差分攻击的问题。

[0174] 下面概括一下本实施例的安全求交流程,P_0方和P_1方,首先各自在本地执行id数值化、样本矩阵补齐、补齐矩阵碎片化,然后通过密态id排序以及密态对齐,生成纯碎片态的对齐样本结果,P_0方持有S_0碎片,P_1方持有S_1碎片,只有当S_0+S_1才能恢复明文态的交集结果。因此处于碎片态,是可以保证交集结果不泄露,双方都不知道交集是哪一部分用户。

[0175] 本实施方式提供了一种全匿的安全求交方法,不仅可以在保护交集以外信息不被泄漏的同时,也能对交集也起到保护作用,使得各个参与方无法知晓真正的交集信息,并且,全匿的求交结果依然可以支持纵向联邦学习。

[0176] 作为本实施例的一种可选地实施方式,原始碎片矩阵包括样本数据碎片,样本数据碎片包括唯一标识碎片和与唯一标识碎片对应的属性数据碎片。

[0177] 本实施例的步骤S1041包括:

[0178] S1041a、分别基于预设排序算子提取各参与方的拼接矩阵中的唯一标识碎片相同的样本数据碎片并进行排序,以得到各自对应的排序矩阵。

[0179] 密态排序之后得到新的排序后的排序矩阵D'_f:

[0180]

$Sorted-id$	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<224361>	<0>	<2.3>	<5.5>	<4>	<0>	<0>
<428493>	<1>	<3.6>	<3.5>	<-3>	<0>	<0>
<428493>	<0>	<0>	<0>	<0>	<2.6>	<3.9>
<672684>	<0>	<-1.2>	<0.5>	<1.2>	<0>	<0>

[0181]

<858329>	<0>	<0>	<0>	<0>	<0.77>	<1.45>
<1030916>	<1>	<0.8>	<0.13>	<3>	<0>	<0>
<1030916>	<0>	<0>	<0>	<0>	<-2.2>	<-1.1>

[0182] 需要说明的是，本实施例的预设排序算子基于快速排序算法或排序网络算法中实现。

[0183] 本实施例的预设排序算子实现逻辑可以基于快速排序算法实现，快速排序是一种成熟的比较方案，计算复杂度为O(n)，这里采用基于多方安全计算(MPC)排序算子实现密态的排序，也就是对碎片化的id列进行排序，得到排序后的碎片态样本矩阵，其中涉及的比较算子也是采用MPC中常用的比较算子，在此不作介绍。

[0184] 在一种可选地实施方式中，排序网络算法基于双调排序算法实现，具体地，双调排序(bitonic sort)属于排序网络(Sorting Network)的一种。相较于传统的排序算法，排序网络算子的价值在于，可以同时处理多个比较器，排序的速度将大幅度提高。简单来说，它是一种可以并行计算的排序算法。

[0185] 双调排序网络算子，是基于Batcher定理而构建的。Batcher定理是说将任意一个长为2^n的双调序列A分为等长的两半X和Y，将X中的元素与Y中的元素一一按原序比较，即a[i]与a[i+n](i<n)比较，将较大者放入MAX序列，较小者放入MIN序列。则得到的MAX和MIN序列仍然是双调序列，并且MAX序列中的任意一个元素不小于MIN序列中的任意一个元素。

[0186] 所谓双调序列(Bitonic Sequence)是指由一个非严格增序列X和非严格减序列Y构成的序列，比如序列(23,10,8,3,5,7,11,78)

[0187] 在本方案中，对其做了进一步改进，可以输入任意长度的双调序列，不需要遵循双调序列长度总和必须为2^n的限制。另外，将双调排序算子利用多方安全计算算子进行了改进，实现了可并行计算的隐私计算排序算法，大幅提升了在全密态数据计算下的排序性能，相较于传统的排序方法，速度提高了500倍。

[0188] 作为本实施例的另一种可选地实施方式，本实施例的步骤S1042包括：

[0189] S1042a、分别基于各参与方的排序矩阵依次比较相邻的样本数据碎片对应的唯一标识碎片是否相同，以根据比较结果进行样本特征对齐计算，得到目标求交样本。

[0190] 可选地，如图3所示，本实施例的步骤S1042a包括：

[0191] S1042a1、根据预设转换算子将碎片化的比较结果转化为对应的第一比较值或第二比较值；

[0192] S1042a2、将相邻的样本数据碎片中的对应属性数据碎片进行密态求和，并将各个求和值依次与第一比较值或第二比较值相乘，得到目标求交样本。

[0193] 具体地，基于排序后的碎片态ID列，执行逐项比较操作。

[0194] 由于同一ID仅会出现最多一次相同，且比较过程中，并不清楚参与比较的id原始值，因此对Sorted-id列进行逐项两两比较，并将比较结果B2A，且对非id部分进行密态求和，将B2A结果与密态和相乘即可。执行复杂度为O(n)。

[0195] 基于该假设，以上述D'_f为例，执行密态对齐操作：

[0196] 标签部分：$\langle Y\rangle_j^{psi} = \left(B2A(< id_j >==< id_{j+1} >)\right) * (< Y_j > +< Y_{j+1} >)$

[0197] 特征部分：$\langle f\rangle_{ij}^{psi} = B2A(< id_j >==< id_{j+1} >) * (< f_{ij} > +< f_{ij+1} >)$

[0198] 其中，i表示特征索引，j表示样本索引。

[0199] 执行结束，就得到了最终的全密态的对齐样本，如下表：

[0200]

$Sorted-id$	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<224361>	<0>	<0>	<0>	<0>	<0>	<0>
<428493>	<1>	<3.6>	<3.5>	<-3>	<2.6>	<3.9>

[0201]

</>	<0>	<0>	<0>	<0>	<0>	<0>
<672684>	<0>	<0>	<0>	<0>	<0>	<0>
<858329>	<0>	<0>	<0>	<0>	<0>	<0>
<1030916>	<1>	<0.8>	<0.13>	<3>	<-2.2>	<-1.1>

[0202] 需要说明的是，由于第二条<428493>与第三条<428493>样本的id相同，因此第一条<428493>进行了密态对齐。而第三条<428493>与第四条<672684>样本的id不同，因此第三条的<428493>值将变成全0，为了不引起误解，这里将第三条样本id由<428493>改成占位符，这样做，不会改变真实交集的结果，因为交集<428493>样本的真实数据仅出现一次。另外，<1030916>同样存在id重复，因此也是交集。由于样本的最后一条已经参与了与倒数第一条的计算，因此最后一条直接丢弃即可。

[0203] 在本实施例中，本实施例的步骤S1042a1包括：

[0204] 在比较结果相同时，基于B2A算子将碎片化的比较结果转化为算术类型的第一比较值；在比较结果不相同时，基于B2A算子将碎片化的比较结果转化为算术类型的第二比较值。

[0205] 还需要说明的是，这里的B2A算子，是一种MPC多方安全计算算子，可以实现将碎片化的布尔类型结果转换为算术类型的结果，如碎片化布尔结果为<False>，通过B2A算子转换之后，变成碎片态结果<0>。同理对碎片化布尔结果为<True>，通过B2A算子转换之后，变成碎片态结果<1>。因此当<1>去乘后面的加法结果，表示保留加法结果本身，如果用<0>去乘，则整体变成<0>。其中，第一比较值可以为碎片态结果<1>，第二比较值可以为碎片态结

果<0>。

[0206] 在一种可选的实施方式中，在步骤S101之后，本实施例的安全求交方法还包括：

[0207] S101c、基于预设密态打乱算法对原始碎片矩阵中若干列进行密态打乱，以得到样本顺序变换后的新的原始碎片矩阵；

[0208] 其中，不同参与者对应的原始碎片矩阵采用相同的预设密态打乱算法进行密态打乱处理。

[0209] 还需要说明的是，本实施例不仅可以原始碎片矩阵进行密态打乱，还可以对补齐后的碎片矩阵进行密态打乱。

[0210] 通过对样本数据的密态打乱，使得各参与方都无法推知真实的样本顺序，也就无法推测在后续的比较过程中参与比较的对象，进一步保证用户信息的安全性。

[0211] 其中，预设密态打乱算法的逻辑如下：假如原始的补齐后的碎片态样本矩阵为M，P_0方生成变换矩阵1并碎片化，P_1方生成变换矩阵2并碎片化，其中变换矩阵是指对单位阵的调整，由于P_0方无法知晓P_1方生成的变换矩阵2，P_1方也无法推知P_0方生成的变换矩阵1，因此通过碎片态矩阵连乘，可以得到碎片态的变换样本矩阵，也就是对样本顺序进行了密态打乱，从而各参与方无法感知样本的真实排列顺序。

[0212] 可选地，打乱公式如下：

[0213] ShuffleMatrix＝矩阵M@变换矩阵1@变换矩阵2

[0214] 因此只需要P_0，P_1各自生成变换矩阵并碎片化，然后与碎片增广矩阵进行矩阵乘，即可得到密态打乱的结果变换矩阵的生成，基于对单位阵的调整即可，比如需要交换2，3列，仅需交换单位阵中2，3列的顺序即可。示例如下：

[0215] $$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \text{--> } \begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$$

[0216] 本实施例通过采取排序近邻比较思想，将隐私求交算法的计算复杂度控制在O(n)，在保证安全性同时，提升了计算性能。另外我们提出了两种不同的全匿踪的安全求交思路，一种是基于密态对齐，一种是通过密态洗牌机制，丰富了技术实现，也可以满足更多应用场景。

[0217] 可选地，在碎片矩阵进行密态打乱后，且在步骤S1042之后，本实施例的安全求交方法还包括：

[0218] 将碎片化的比较结果进行恢复处理。

[0219] 可选地，作为步骤S1042a的另一种实施方式，S1042a还包括：

[0220] 在比较结果相同时，并将相邻的样本数据碎片中对应的属性数据碎片进行密态求和；在比较结果不相同时，丢弃排序位置靠前的样本数据，以得到新的目标求交样本。

[0221] 具体地，基于排序后的碎片态id列，执行逐项比较操作。

[0222] 由于同一id仅会出现最多一次相同，且比较过程中，并不清楚参与比较的id原始值，因此对Sorted-id列进行逐项比较，并将比较结果恢复用于判断是否相同，不会暴露信息。执行复杂度为O(n)。

[0223] 基于该假设，以上述D'_f为例，执行密态对齐操作：

[0224] <124360>与<328492>密态比较，恢复结果为False，丢弃<124360>该条样本。

[0225] <328492>与<328492>密态比较，恢复结果为True，对这两条样本非id碎片数值进行密态加法。

[0226] 直接跳到索引3的样本，执行索引3和4的碎片id比较：

[0227] <572683>与<748329>密态比较，恢复结果为False，丢弃<572683>该条样本。

[0228] <748329>与<930913>密态比较，恢复结果为False，丢弃<124360>该条样本。

[0229] <930913>与<930913>密态比较，恢复结果为True，对这两条样本非id碎片数值进行密态加法。

[0230] 执行结束，就得到了新的目标求交样本，如下表：

[0231]

Sorted-id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<428493>	<1>	<3.6>	<3.5>	<-3>	<2.6>	<3.9>
<1030916>	<1>	<0.8>	<0.13>	<3>	<-2.2>	<-1.1>

[0232] 在一个具体的例子中，本实施例利用上述步骤相互组合进而可以提供两种不同的安全求交流程，如图4和图5所示。

[0233] 本实施例通过获取各参与方的原始数据矩阵，对各个参与方的原始数据矩阵进行本地碎片化处理后进行拼接，然后通过密态排序以及密态对齐，生成碎片态的求交样本，由于求交样本为碎片态，可以保证交集结果不泄露，从而安全求交的全流程中不暴露任何敏感信息，既保护交集以外的信息，同时输出的结果又可以保护交集信息，进而能够执行高标准的安全要求和实现保护敏感数据的目标。同时本实施例的安全求交方案计算复杂度低，在实现安全性的同时，还能够保障计算性能，以满足真实场景所需。

[0234] 实施例2

[0235] 本发明还提供一种数据共享中的安全求交系统，应用于至少两个参与方之间数据共享场景中，如图6所示，本实施例的安全求交系统包括：

[0236] 获取模块1，用于获取各参与方的原始数据矩阵。

[0237] 其中，原始数据矩阵包括至少一组样本数据，样本数据包括用于标识参与方中每个对象的唯一标识和与唯一标识相对应的属性数据。

[0238] 作为可选地实施方式，本实施例的安全求交系统还包括：

[0239] 类型转换模块5，用于判断原始数据矩阵中的唯一标识的类型.

[0240] 若唯一标识为字符串型，则将字符串的唯一标识进行数值化处理以得到数值化的唯一标识，若唯一标识为数值型，则不进行操作。

[0241] 需要说明的是，在求交阶段，需要指定某列作为求交的对象列，比如采用身份证号、手机号等作为判断是否为同一用户的依据。因此求交的对象列必须具备唯一性要求。比如下表采用的是id列，该列值为字符串，因此首先需要做数值化，本实施方式可以采用hash数值化，将字符串id值数值化，方便后续步骤的执行，数值化需要满足映射后的唯一性要求，即同一字符串id映射后也是唯一的数值，不能存在交叉重复。

[0242] 在一个实施方式中各个参与方可以通过商讨确定原始数据矩阵的组成规范，从而可以在获取原始数据矩阵之后可以直接进行下一步，例如，各方都具有相同数量的特征数量，如同具有一个唯一标识、两个特征数据和一个标签数据，本实施例对此不作限定。

[0243] 而在另一个实施方式中，当各个参与方没有规定原始数据矩阵的组成时，如下两表分别示出了P_0方和P_1方的原始特征数据。

[0244] P_0方(GUEST)原始特征数据D_0：

[0245]

id	Y	X_{a1}	X_{a2}	X_{a3}
224361	0	2.3	5.5	4
428493	1	3.6	3.5	-3
672684	0	-1.2	0.5	1.2
1030916	1	0.8	0.13	3

[0246] P_1方(HOST)原始特征数据D_1：

[0247]

id	X_{b1}	X_{b2}
858329	0.77	1.45
428493	2.6	3.9
1030916	-2.2	-1.1

[0248] 可选地，本实施例的安全求交系统还包括：

[0249] 补齐模块6，用于判断各个参与方的原始数据矩阵中属性数据对应的列数是否相同，若不相同，则根据预设补齐规则生成虚拟属性数据列进行补齐，以得到补齐后的原始数据矩阵.

[0250] 其中，各参与方所对应补齐后的原始数据矩阵的列数相等。

[0251] 也即对原始数据矩阵非id域补齐，各参与方需要对各自的样本数据进行补齐操作，该阶段，各方会进行同步特征数量，然后按照预设补齐规则，生成虚拟标签列和虚拟标签列。预设补齐规则可以包括在虚拟标签列和虚拟标签列的填充值0，也可以是某种规则生成的值，例如在P_0方同时加1，在P_1方同时减1，只需要保证该信息可以被后续操作进行去除即可，本实施方式对此不作限定，本实施方式采用的是0填充，通过对P_0方和P_1方的原始数据进行增广补齐之后，就得到同样列大小的矩阵，如下表的D'_0和D'_1所示。

[0252] P_0方特征矩阵根据P_1特征数及标签进行补齐，得到D'_0：

[0253]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
224361	0	2.3	5.5	4	0	0
428493	1	3.6	3.5	-3	0	0
672684	0	-1.2	0.5	1.2	0	0
1030916	1	0.8	0.13	3	0	0

[0254] 同理，P_1方特征矩阵根据P_0特征数进行补齐，得到D'_1：

[0255]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
858329	0	0	0	0	0.77	1.45
428493	0	0	0	0	2.6	3.9
1030916	0	0	0	0	-2.2	-1.1

[0256] 碎片化模块2，用于基于样本数据对各参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵，并基于原始碎片矩阵生成对应的随机数碎片矩阵，将每个参与方的随机数碎片矩阵发送给其他参与方。

[0257] 需要说明的是，参与方还可以将的随机数碎片矩阵通过秘密共享给其他参与方。

[0258] 可选地，碎片化模块2，还用于基于样本数据对各参与方的原始数据矩阵中的每一

个原始数据都减去一个随机数，以得到差值碎片和随机数碎片，将所有的差值碎片作为原始碎片矩阵，将所有的随机数碎片作为随机数碎片矩阵。

[0259] 碎片化公式如下：

[0260] 对x进行碎片化$Shr_A^i(x)$：

[0261] P_i选择$r \in_R Z_{2^l}$，使$\langle x\rangle_A^i = x - r$，并且发送r给$P_{1-i}$，使$\langle x\rangle_A^{1-i} = r$，

[0262] 其中，r为随机数。

[0263] 在本实施例中分别对P_0和P_1进行碎片化处理，如下表的所示：

[0264] P_0方对D'_0进行本地碎片化并秘密共享(id也需要碎片化)

[0265]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<224361>	<0>	<2.3>	<5.5>	<4>	<0>	<0>
<428493>	<1>	<3.6>	<3.5>	<-3>	<0>	<0>
<672684>	<0>	<-1.2>	<0.5>	<1.2>	<0>	<0>
<1030916>	<1>	<0.8>	<0.13>	<3>	<0>	<0>

[0266] 同理，P_1方对D'_1进行本地碎片化并秘密共享

[0267]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<858329>	<0>	<0>	<0>	<0>	<0.77>	<1.45>
<428493>	<0>	<0>	<0>	<0>	<2.6>	<3.9>
<1030916>	<0>	<0>	<0>	<0>	<-2.2>	<-1.1>

[0268] 注：<x>表示x的碎片态。

[0269] 拼接模块3，用于将各参与方的原始碎片矩阵与从其他参与方获得的随机数碎片矩阵进行拼接处理，以得到各自对应的拼接矩阵。如下表所示D_f：

[0270] MPC Concat执行得到拼接后的拼接矩阵D_f

[0271]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<224361>	<0>	<2.3>	<5.5>	<4>	<0>	<0>
<428493>	<1>	<3.6>	<3.5>	<-3>	<0>	<0>
<672684>	<0>	<-1.2>	<0.5>	<1.2>	<0>	<0>
<1030916>	<1>	<0.8>	<0.13>	<3>	<0>	<0>
<858329>	<0>	<0>	<0>	<0>	<0.77>	<1.45>
<428493>	<0>	<0>	<0>	<0>	<2.6>	<3.9>
<1030916>	<0>	<0>	<0>	<0>	<-2.2>	<-1.1>

[0272] 求交样本确定模块4，用于分别基于各参与方的拼接矩阵确定目标求交样本。

[0273] 需要说明的是，可以通过数学方式从拼接矩阵选择出唯一标识相同的样本数据，进而可以确定目标求交样本。

[0274] 在一种可选的实施方式中，本实施例的求交样本确定模块4包括：

[0275] 排序单元41，用于分别基于各参与方的拼接矩阵进行排序，以得到与拼接矩阵对应的排序矩阵；

[0276] 对齐单元42，用于分别基于各参与方的排序矩阵进行样本特征对齐计算，以确定

目标求交样本。

[0277] 需要说明的是，对于该目标求交样本，由于不进行恢复明文态，保持全碎片态，因此双方都不知道交集具体是哪些，因此具备强安全性，也不存在差分攻击的问题。

[0278] 下面概括一下本实施例的安全求交流程，P_0方和P_1方，首先各自在本地执行id数值化、样本矩阵补齐、补齐矩阵碎片化，然后通过密态id排序以及密态对齐，生成纯碎片态的对齐样本结果，P_0方持有S_0碎片，P_1方持有S_1碎片，只有当S_0+S_1才能恢复明文态的交集结果。因此处于碎片态，是可以保证交集结果不泄露，双方都不知道交集是哪一部分用户。

[0279] 本实施方式提供了一种全匿的安全求交方法，不仅可以在保护交集以外信息不被泄漏的同时，也能对交集也起到保护作用，使得各个参与方无法知晓真正的交集信息，并且，全匿的求交结果依然可以支持纵向联邦学习。

[0280] 作为本实施例的一种可选地实施方式，原始碎片矩阵包括样本数据碎片，样本数据碎片包括唯一标识碎片和与唯一标识碎片对应的属性数据碎片。

[0281] 本实施例的排序单元41，还用于分别基于预设排序算子提取各参与方的拼接矩阵中的唯一标识碎片相同的样本数据碎片并进行排序，以得到各自对应的排序矩阵。

[0282] 密态排序之后得到新的排序后的排序矩阵D'_f：

[0283]

Sorted-id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<224361>	<0>	<2.3>	<5.5>	<4>	<0>	<0>
<428493>	<1>	<3.6>	<3.5>	<-3>	<0>	<0>
<428493>	<0>	<0>	<0>	<0>	<2.6>	<3.9>
<672684>	<0>	<-1.2>	<0.5>	<1.2>	<0>	<0>
<858329>	<0>	<0>	<0>	<0>	<0.77>	<1.45>
<1030916>	<1>	<0.8>	<0.13>	<3>	<0>	<0>
<1030916>	<0>	<0>	<0>	<0>	<-2.2>	<-1.1>

[0284] 需要说明的是，本实施例的预设排序算子基于快速排序算法或排序网络算法中实现。

[0285] 本实施例的预设排序算子实现逻辑可以基于快速排序算法实现，快速排序是一种成熟的比较方案，计算复杂度为O(n)，这里采用基于多方安全计算(MPC)排序算子实现密态的排序，也就是对碎片化的id列进行排序，得到排序后的碎片态样本矩阵，其中涉及的比较算子也是采用MPC中常用的比较算子，在此不作介绍。

[0286] 在一种可选地实施方式中，排序网络算法基于双调排序算法实现，具体地，双调排序(bitonic sort)属于排序网络(Sorting Network)的一种。相较于传统的排序算法，排序网络算子的价值在于，可以同时处理多个比较器，排序的速度将大幅度提高。简单来说，它是一种可以并行计算的排序算法。

[0287] 双调排序网络算子，是基于Batcher定理而构建的。Batcher定理是说将任意一个长为2^n的双调序列A分为等长的两半X和Y，将X中的元素与Y中的元素一一按原序比较，即a[i]与a[i+n](i<n)比较，将较大者放入MAX序列，较小者放入MIN序列。则得到的MAX和MIN序列仍然是双调序列，并且MAX序列中的任意一个元素不小于MIN序列中的任意一个元素。

[0288] 所谓双调序列(Bitonic Sequence)是指由一个非严格增序列X和非严格减序列Y构成的序列，比如序列(23,10,8,3,5,7,11,78)

[0289] 在本方案中，对其做了进一步改进，可以输入任意长度的双调序列，不需要遵循双调序列长度总和必须为2^n的限制。另外，将双调排序算子利用多方安全计算算子进行了改进，实现了可并行计算的隐私计算排序算法，大幅提升了在全密态数据计算下的排序性能，相较于传统的排序方法，速度提高了500倍。

[0290] 作为本实施例的另一种可选地实施方式，本实施例的对齐单元42，还用于分别基于各参与方的排序矩阵依次比较相邻的样本数据碎片对应的唯一标识碎片是否相同，以根据比较结果进行样本特征对齐计算，得到目标求交样本。

[0291] 可选地，对齐单元42，还用于根据预设转换算子将碎片化的比较结果转化为对应的第一比较值或第二比较值。

[0292] 将相邻的样本数据碎片中的对应属性数据碎片进行密态求和，并将各个求和值依次与第一比较值或第二比较值相乘，得到目标求交样本。

[0293] 具体地，基于排序后的碎片态ID列，执行逐项比较操作。

[0294] 由于同一ID仅会出现最多一次相同，且比较过程中，并不清楚参与比较的id原始值，因此对Sorted-id列进行逐项两两比较，并将比较结果B2A，且对非id部分进行密态求和，将B2A结果与密态和相乘即可。执行复杂度为O(n)。

[0295] 基于该假设，以上述D'_f为例，执行密态对齐操作：

[0296] 标签部分：$\langle Y\rangle_j^{psi} = \left(B2A(< id_j >==< id_{j+1} >)\right) * (< Y_j > + < Y_{j+1} >)$

[0297] 特征部分：$\langle f\rangle_{ij}^{psi} = B2A(< id_j >==< id_{j+1} >) * (< f_{ij} > + < f_{ij+1} >)$

[0298] 其中，i表示特征索引，j表示样本索引。

[0299] 执行结束，就得到了最终的全密态的对齐样本，如下表：

[0300]

Sorted-id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<224361>	<0>	<0>	<0>	<0>	<0>	<0>
<428493>	<1>	<3.6>	<3.5>	<-3>	<2.6>	<3.9>
</>	<0>	<0>	<0>	<0>	<0>	<0>
<672684>	<0>	<0>	<0>	<0>	<0>	<0>
<858329>	<0>	<0>	<0>	<0>	<0>	<0>
<1030916>	<1>	<0.8>	<0.13>	<3>	<-2.2>	<-1.1>

[0301] 需要说明的是，由于第二条<428493>与第三条<428493>样本的id相同，因此第一条<428493>进行了密态对齐。而第三条<428493>与第四条<672684>样本的id不同，因此第三条的<428493>值将变成全0，为了不引起误解，这里将第三条样本id由<428493>改成占位符，这样做，不会改变真实交集的结果，因为交集<428493>样本的真实数据仅出现一次。另外，<1030916>同样存在id重复，因此也是交集。由于样本的最后一条已经参与了与倒数第一条的计算，因此最后一条直接丢弃即可。

[0302] 在本实施例中，对齐单元42，还用于在比较结果相同时，基于B2A算子将碎片化的比较结果转化为算术类型的第一比较值。

[0303] 在比较结果不相同时，基于B2A算子将碎片化的比较结果转化为算术类型的第二比较值。

[0304] 还需要说明的是，这里的B2A算子，是一种MPC多方安全计算算子，可以实现将碎片化的布尔类型结果转换为算术类型的结果，如碎片化布尔结果为<False>，通过B2A算子转换之后，变成碎片态结果<0>。同理对碎片化布尔结果为<True>，通过B2A算子转换之后，变成碎片态结果<1>。因此当<1>去乘后面的加法结果，表示保留加法结果本身，如果用<0>去乘，则整体变成<0>。其中，第一比较值可以为碎片态结果<1>，第二比较值可以为碎片态结果<0>。

[0305] 在一种可选的实施方式中，本实施例的安全求交系统还包括：

[0306] 打乱模块7，用于基预设密态打乱算法对原始碎片矩阵中若干列进行密态打乱，以得到样本顺序变换后的新的原始碎片矩阵。

[0307] 其中，不同参与者对应的原始碎片矩阵采用相同的预设密态打乱算法进行密态打乱处理。

[0308] 还需要说明的是，本实施例不仅可以原始碎片矩阵进行密态打乱，还可以对补齐后的碎片矩阵进行密态打乱。

[0309] 通过对样本数据的密态打乱，使得各参与方都无法推知真实的样本顺序，也就无法推测在后续的比较过程中参与比较的对象，进一步保证用户信息的安全性。

[0310] 其中，预设密态打乱算法的逻辑如下：假如原始的补齐后的碎片态样本矩阵为M，P_0方生成变换矩阵1并碎片化，P_1方生成变换矩阵2并碎片化，其中变换矩阵是指对单位阵的调整，由于P_0方无法知晓P_1方生成的变换矩阵2，P_1方也无法推知P_0方生成的变换矩阵1，因此通过碎片态矩阵连乘，可以得到碎片态的变换样本矩阵，也就是对样本顺序进行了密态打乱，从而各参与方无法感知样本的真实排列顺序。

[0311] 可选地，打乱公式如下：

[0312] ShuffleMatrix＝矩阵M@变换矩阵1@变换矩阵2

[0313] 因此只需要P_0，P_1各自生成变换矩阵并碎片化，然后与碎片增广矩阵进行矩阵乘，即可得到密态打乱的结果变换矩阵的生成，基于对单位阵的调整即可，比如需要交换2，3列，仅需交换单位阵中2，3列的顺序即可。示例如下：

[0314] $$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \text{--> } \begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}$$

[0315] 本实施例通过采取排序近邻比较思想，将隐私求交算法的计算复杂度控制在O(n)，在保证安全性同时，提升了计算性能。另外我们提出了两种不同的全匿踪的安全求交思路，一种是基于密态对齐，一种是通过密态洗牌机制，丰富了技术实现，也可以满足更多应用场景。

[0316] 可选地，在碎片矩阵进行密态打乱后，本实施例的安全求交系统还包括：

[0317] 恢复模块8，用于将碎片化的比较结果进行恢复处理。

[0318] 可选地，对齐单元42，还用于在比较结果相同时，并将相邻的样本数据碎片中对应的属性数据碎片进行密态求和；在比较结果不相同时，丢弃排序位置靠前的样本数据，以得到新的目标求交样本。

[0319] 较佳地，安全求交系统还包括：

[0320] 具体地，基于排序后的碎片态id列，执行逐项比较操作。

[0321] 由于同一id仅会出现最多一次相同，且比较过程中，并不清楚参与比较的id原始值，因此对Sorted-id列进行逐项比较，并将比较结果恢复用于判断是否相同，不会暴露信息。执行复杂度为O(n)。

[0322] 基于该假设，以上述D'_f为例，执行密态对齐操作：

[0323] <124360>与<328492>密态比较，恢复结果为False，丢弃<124360>该条样本。

[0324] <328492>与<328492>密态比较，恢复结果为True，对这两条样本非id碎片数值进行密态加法。

[0325] 直接跳到索引3的样本，执行索引3和4的碎片id比较：

[0326] <572683>与<748329>密态比较，恢复结果为False，丢弃<572683>该条样本。

[0327] <748329>与<930913>密态比较，恢复结果为False，丢弃<124360>该条样本。

[0328] <930913>与<930913>密态比较，恢复结果为True，对这两条样本非id碎片数值进行密态加法。

[0329] 执行结束，就得到了新的目标求交样本，如下表：

[0330]

Sorted-id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<428493>	<1>	<3.6>	<3.5>	<-3>	<2.6>	<3.9>
<1030916>	<1>	<0.8>	<0.13>	<3>	<-2.2>	<-1.1>

[0331] 本实施例通过获取各参与方的原始数据矩阵，对各个参与方的原始数据矩阵进行本地碎片化处理后进行拼接，然后通过密态排序以及密态对齐，生成碎片态的求交样本，由于求交样本为碎片态，可以保证交集结果不泄露，从而安全求交的全流程中不暴露任何敏感信息，既保护交集以外的信息，同时输出的结果又可以保护交集信息，进而能够执行高标准的安全要求和实现保护敏感数据的目标。同时本实施例的安全求交方案计算复杂度低，在实现安全性的同时，还能够保障计算性能，以满足真实场景所需。

[0332] 实施例3

[0333] 需要先说明的是联邦学习模型可以为各种模型，如支持向量机(SupportVectorMachine，SVM)、逻辑回归(logisticsregression，LR)等线性模型，也可以支持深度神经网络(DeepNeuralNetwork，DNN)模型等，在此不作限制。

[0334] 本实施例的联邦学习算法的实现是基于多方安全计算秘密共享技术。基于多方安全计算协议，可以实现碎片态数据计算的各类算子，比如加法、减法、乘法、除法、比较、最大最小值、中位数、各类非线性函数的多项式表达等，通过密态算子，可以实现建模算法的联邦化。

[0335] 本实施例以联邦学习逻辑回归算法为例，阐述联邦学习模型的训练方法的实现逻辑。

[0336] 本实施例提供一种联邦学习模型的训练方法，如图7所示，本实施例的训练方法包括：

[0337] S1、获取各参与方利用实施例1的安全求交方法得到的碎片化的目标求交样本。

[0338] 本步骤用于模型训练数据的获取，加载完成后的数据类型是一种碎片状态，各方持有其中的一份碎片，无法推知完整的数据明文结果，从而可以保证信息的安全性。

[0339] S2、基于预设划分策略获取各参与方对目标求交样本执行划分后得到的训练集碎片和测试集碎片。

[0340] 本步骤中，各方按照约定的训练集测试集划分策略，各方执行数据集的划分。预设划分策略可以是比例，也可以是随机数生成策略。

[0341] S3、获取各参与方利用各自的训练集碎片、测试集碎片通过安全多方计算算子进行特征与权重参数计算得到的预测碎片。

[0342] 在本步骤中，基于MPC算子执行特征与权重参数计算，得到预测碎片Y'_0，此处的用多项式分段函数形式模拟sigmoid函数，$Y'_0=h_w(x^i)$，其中h为sigmoid函数。

[0343] S4、获取各参与方利用各自的预测碎片通过安全多方计算算子进行梯度计算得到的梯度碎片。

[0344] 具体的，基于MPC算子计算得到碎片态梯度g_0。

[0345] $$g_i=\frac{\Delta L(w)}{\Delta w_j}=\frac{1}{m}\sum_{i=1}^{m}(h_w(x_i)-Y_i)x_{ij}+\frac{\lambda}{m}w_j$$

[0346] 在本步骤中，在联邦逻辑回归模型中，对于梯度的计算，需要进行求平均，batchsize也就是对应该公式中的m。此时的m需要对全匿PSI中比较结果的B2A进行密态求和，得到聚合之后的真实交集数量的碎片态结果，再进行密态均值计算。举例来说，全匿PSI输出结果的一个批次中，由于同时存在交集和非交集部分，批次大小是大于真实的交集量级的，假设一个批次大小为500条样本，而真实交集为300，梯度计算是需要基于300来操作，而不是500，这个是相对于非全匿方案的差别。全匿方案中首先需要计算非0部分的量级，也就是对比较结果进行B2A，做密态sum，得到密态的交集大小碎片结果m，也就是<300>，然后再进行密态除法得到密态均值。自始自终，没有暴露交集量级和交集结果，在不影响模型训练的基础上，实现全匿的目标。

[0347] S5、获取各参与方利用各自的梯度碎片通过安全多方计算算子进行更新权重系数计算，以更新初始权重碎片得到新的权重碎片，并利用新的权重碎片进行迭代。

[0348] 可选地，基于MPC算子计算得到碎片态梯度g更新权重系数w，得到新的权重碎片w_0。

[0349] $$w_i=w_i-lr\cdot\frac{\Delta L(w_i)}{\Delta w_i}$$

[0350] 在本步骤中，生成最终的学习权重碎片，各方可以保存模型参数碎片或者对各自持有特征对应的权重进行恢复保存私有权重，完成模型的参数保存。

[0351] 还需要说明的是，对于不同的模型，训练主流程中的具体计算函数会存在一定的差异，通过通用多方安全计算算子，可以实现任意的计算函数，对于非线性函数，一般可以通过泰勒展开或者其他近似多项式表达来解决，也就可以证明本实施例的联邦学习模型可以为各种模型，提高了适用性。

[0352] S6、在权重碎片满足预设条件时则获取目标权重碎片，并利用目标权重碎片建立联邦学习模型。

[0353] 在一种可选地实施方式中，在步骤S3之后，本实施例的联邦学习模型的训练方法还包括：

[0354] S301a、获取各参与方基于各自的预测碎片通过安全多方计算算子进行损失值计算得到损失值碎片。

[0355] 可选地，基于MPC算子执行损失值计算，得到LogLoss碎片值J'_0

[0356] $$J \approx \frac{1}{m}\sum_{i=1}^{m}\left(\log 2 - \frac{1}{2}Y_i Y_i' + \frac{1}{8}(Y_i')^2\right)$$

[0357] S301b、任一参与方接收其他参与方发送的损失值碎片，并将所有的损失值碎片恢复至对应的明文后上报训练日志。

[0358] 可选地，接收损失值碎片J'_1恢复LogLoss值明文上报到训练日志。

[0359] 在一种可选地实施方式中，在步骤S1之前，本实施例的联邦学习模型的训练方法还包括：

[0360] S0、加载超参数配置信息，其中，超参数配置信息包括预设迭代次数；

[0361] 在步骤S3之后，本实施例的联邦学习模型的训练方法还包括：

[0362] S302a、判断实际迭代次数是否达到预设迭代次数；

[0363] 若是，执行步骤S302a1；

[0364] 若否，则执行步骤S4。

[0365] S302a1、则终止训练。

[0366] 在一种可选地实施方式中，在步骤S5之后，本实施例的训练方法还包括：

[0367] S501、分别通过安全多方计算算子判断各参与方的特征对应的梯度碎片的梯度值是否小于预设阈值，若是则任一参与方接收其他参与方发送的比较结果碎片，并将比较结果碎片恢复至对应的明文。

[0368] 在一种可选地实施方式中，本实施例的训练方法还包括：

[0369] S6、判断训练状态是否为终止训练。

[0370] 若是，则执行步骤S61；

[0371] 若否，则执行步骤S62。

[0372] S61、则输出并根据使用需求保存模型参数为对应的模型参数明文或模型参数碎片。

[0373] S62、则执行对梯度碎片通过安全多方计算算子进行梯度更新权重系数计算得到新的目标权重碎片。

[0374] 还需要说明的是，对于只有两个参与方可以利用乘法三元组生成。

[0375] 如图8所示，示出了一种联邦学习模型的训练方法的流程图。

[0376] 本实施例的所提联邦学习模型的训练方法，可以扩展到各类机器学习算法，通过各参与方加载各自本地的碎片化样本求交数据，采用通用的多方安全计算算子，可以实现端到端模型训练，与全匿踪的安全求交完全适配，通过全匿踪的安全求交与全匿踪的联邦学习算法的有机结合，形成了完整的端到端全匿踪纵向联邦学习框架，可以支撑高安全性高性能要求的联合建模场景，不暴露敏感信息，满足高标准客户的要求。

[0377] 实施例4

[0378] 本发明还提供一种联邦学习模型的训练系统，如图9所示，本实施例的训练系统包括：

[0379] 求交样本获取模块101，用于获取各参与方利用实施例2的安全求交系统得到的碎片化的目标求交样本。

[0380] 本步骤用于模型训练数据的获取，加载完成后的数据类型是一种碎片状态，各方

持有其中的一份碎片，无法推知完整的数据明文结果，从而可以保证信息的安全性。

[0381] 划分模块102，用于基于预设划分策略获取各参与方对目标求交样本执行划分后得到的训练集碎片和测试集碎片。

[0382] 本步骤中，各方按照约定的训练集测试集划分策略，各方执行数据集的划分。预设划分策略可以是比例，也可以是随机数生成策略。

[0383] 预测碎片计算模块103，用于获取各参与方利用各自的训练集碎片、测试集碎片通过安全多方计算算子进行特征与权重参数计算得到的预测碎片。

[0384] 在本步骤中，基于MPC算子执行特征与权重参数计算，得到预测碎片Y'_0，此处的用多项式分段函数形式模拟sigmoid函数，$Y'_0=h_w(x^i)$，其中h为sigmoid函数。

[0385] 梯度碎片计算模块104，用于获取各参与方利用各自的预测碎片通过安全多方计算算子进行梯度计算得到的梯度碎片。

[0386] 具体的，基于MPC算子计算得到碎片态梯度g_0。

[0387] $$g_i=\frac{\Delta L(w)}{\Delta w_j}=\frac{1}{m}\sum_{i=1}^{m}(h_w(x_i)-Y_i)x_{ij}+\frac{\lambda}{m}w_j$$

[0388] 在本步骤中，在联邦逻辑回归模型中，对于梯度的计算，需要进行求平均，batchsize也就是对应该公式中的m。此时的m需要对全匿PSI中比较结果的B2A进行密态求和，得到聚合之后的真实交集数量的碎片态结果，再进行密态均值计算。举例来说，全匿PSI输出结果的一个批次中，由于同时存在交集和非交集部分，批次大小是大于真实的交集量级的，假设一个批次大小为500条样本，而真实交集为300，梯度计算是需要基于300来操作，而不是500，这个是相对于非全匿方案的差别。全匿方案中首先需要计算非0部分的量级，也就是对比较结果进行B2A，做密态sum，得到密态的交集大小碎片结果m，也就是<300>，然后再进行密态除法得到密态均值。自始自终，没有暴露交集量级和交集结果，在不影响模型训练的基础上，实现全匿的目标。

[0389] 权重碎片更新模块105，用于获取各参与方利用各自的梯度碎片通过安全多方计算算子进行更新权重系数计算，以更新初始权重碎片得到新的权重碎片，并利用新的权重碎片进行迭代。

[0390] 可选地，基于MPC算子计算得到碎片态梯度g更新权重系数w，得到新的权重碎片w_0。

[0391] $$w_i=w_i-lr\cdot\frac{\Delta L(w_i)}{\Delta w_i}$$

[0392] 在本步骤中，生成最终的学习权重碎片，各方可以保存模型参数碎片或者对各自持有特征对应的权重进行恢复保存私有权重，完成模型的参数保存。

[0393] 还需要说明的是，对于不同的模型，训练主流程中的具体计算函数会存在一定的差异，通过通用多方安全计算算子，可以实现任意的计算函数，对于非线性函数，一般可以通过泰勒展开或者其他近似多项式表达来解决，也就可以证明本实施例的联邦学习模型可以为各种模型，提高了适用性。

[0394] 模型建立模块106，用于在权重碎片满足预设条件时则获取目标权重碎片，并利用目标权重碎片建立联邦学习模型。

[0395] 在一种可选地实施方式中，本实施例的训练系统还包括：

[0396] 损失值碎片计算模块107，用于获取各参与方基于各自的预测碎片通过安全多方计算算子进行损失值计算得到损失值碎片。

[0397] 可选地，基于MPC算子执行损失值计算，得到LogLoss碎片值J'_0

[0398] $$J \approx \frac{1}{m}\sum_{i=1}^{m}\left(\log 2 - \frac{1}{2}Y_i Y_i' + \frac{1}{8}(Y_i')^2\right)$$

[0399] 任一参与方接收其他参与方发送的损失值碎片，并将所有的损失值碎片恢复至对应的明文后上报训练日志。

[0400] 可选地，接收损失值碎片J'_1恢复LogLoss值明文上报到训练日志。

[0401] 在一种可选地实施方式中，本实施例的训练系统还包括：

[0402] 梯度碎片比较模块108，用于分别通过安全多方计算算子判断各参与方的特征对应的梯度碎片的梯度值是否小于预设阈值，若是则任一参与方接收其他参与方发送的比较结果碎片，并将比较结果碎片恢复至对应的明文。

[0403] 在一种可选地实施方式中，本实施例的训练系统还包括：

[0404] 训练状态判断模块109，用于判断训练状态是否为终止训练；

[0405] 若是，则输出并根据使用需求保存模型参数为对应的模型参数明文或模型参数碎片；

[0406] 若否，则执行对梯度碎片通过安全多方计算算子进行梯度更新权重系数计算得到新的目标权重碎片。

[0407] 还需要说明的是，对于只有两个参与方可以利用乘法三元组生成。

[0408] 本实施例的所提联邦学习模型的训练方法，可以扩展到各类机器学习算法，通过各参与方加载各自本地的碎片化样本求交数据，采用通用的多方安全计算算子，可以实现端到端模型训练，与全匿踪的安全求交完全适配，通过全匿踪的安全求交与全匿踪的联邦学习算法的有机结合，形成了完整的端到端全匿踪纵向联邦学习框架，可以支撑高安全性高性能要求的联合建模场景，不暴露敏感信息，满足高标准客户的要求。

[0409] 实施例5

[0410] 图10为本发明实施例3提供的一种电子设备的结构示意图。电子设备包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，处理器执行程序时实现上述实施例的方法。图10显示的电子设备30仅仅是一个示例，不应对本发明实施例的功能和使用范围带来任何限制。

[0411] 如图10所示，电子设备30可以以通用计算设备的形式表现，例如其可以为服务器设备。电子设备30的组件可以包括但不限于：上述至少一个处理器31、上述至少一个存储器32、连接不同系统组件(包括存储器32和处理器31)的总线33。

[0412] 总线33包括数据总线、地址总线和控制总线。

[0413] 存储器32可以包括易失性存储器，例如随机存取存储器(RAM)321和/或高速缓存存储器322，还可以进一步包括只读存储器(ROM)323。

[0414] 存储器32还可以包括具有一组(至少一个)程序模块324的程序/实用工具325，这样的程序模块324包括但不限于：操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。

[0415] 处理器31通过运行存储在存储器32中的计算机程序，从而执行各种功能应用以及数据处理，例如本发明实施例1的播放控制方法。

[0416] 电子设备30也可以与一个或多个外部设备34(例如键盘、指向设备等)通信。这种通信可以通过输入/输出(I/O)接口35进行。并且，模型生成的设备30还可以通过网络适配器36与一个或者多个网络(例如局域网(LAN)，广域网(WAN)和/或公共网络，例如因特网)通信。如图10所示，网络适配器36通过总线33与模型生成的设备30的其它模块通信。应当明白，尽管图中未示出，可以结合模型生成的设备30使用其它硬件和/或软件模块，包括但不限于：微代码、设备驱动器、冗余处理器、外部磁盘驱动阵列、RAID(磁盘阵列)系统、磁带驱动器以及数据备份存储系统等。

[0417] 应当注意，尽管在上文详细描述中提及了电子设备的若干单元/模块或子单元/模块，但是这种划分仅仅是示例性的并非强制性的。实际上，根据本发明的实施方式，上文描述的两个或更多单元/模块的特征和功能可以在一个单元/模块中具体化。反之，上文描述的一个单元/模块的特征和功能可以进一步划分为由多个单元/模块来具体化。

[0418] 实施例6

[0419] 本实施例提供了一种计算机可读存储介质，其上存储有计算机程序，程序被处理器执行时实现上述实施例的方法中的步骤。

[0420] 其中，可读存储介质可以采用的更具体可以包括但不限于：便携式盘、硬盘、随机存取存储器、只读存储器、可擦拭可编程只读存储器、光存储器件、磁存储器件或上述的任意合适的组合。

[0421] 在可能的实施方式中，本发明还可以实现为一种程序产品的形式，其包括程序代码，当程序产品在终端设备上运行时，程序代码用于使终端设备执行实现上述实施例的步骤。

[0422] 其中，可以以一种或多种程序设计语言的任意组合来编写用于执行本发明的程序代码，程序代码可以完全地在用户设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户设备上部分在远程设备上执行或完全在远程设备上执行。

[0423] 虽然以上描述了本发明的具体实施方式，但是本领域的技术人员应当理解，这仅是举例说明，本发明的保护范围是由所附权利要求书限定的。本领域的技术人员在不背离本发明的原理和实质的前提下，可以对这些实施方式做出多种变更或修改，但这些变更和修改均落入本发明的保护范围。

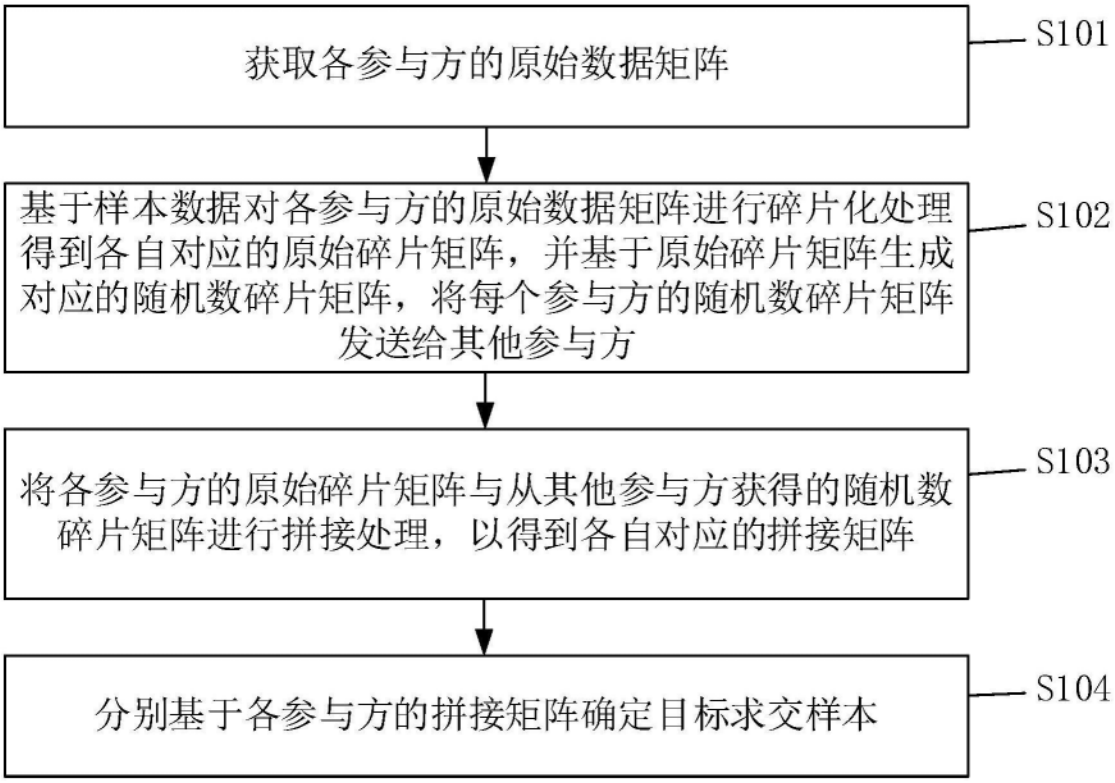

获取各参与方的原始数据矩阵
S101
基于样本数据对各参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵，并基于原始碎片矩阵生成对应的随机数碎片矩阵，将每个参与方的随机数碎片矩阵发送给其他参与方
S102
将各参与方的原始碎片矩阵与从其他参与方获得的随机数碎片矩阵进行拼接处理，以得到各自对应的拼接矩阵
S103
分别基于各参与方的拼接矩阵确定目标求交样本
S104

图1

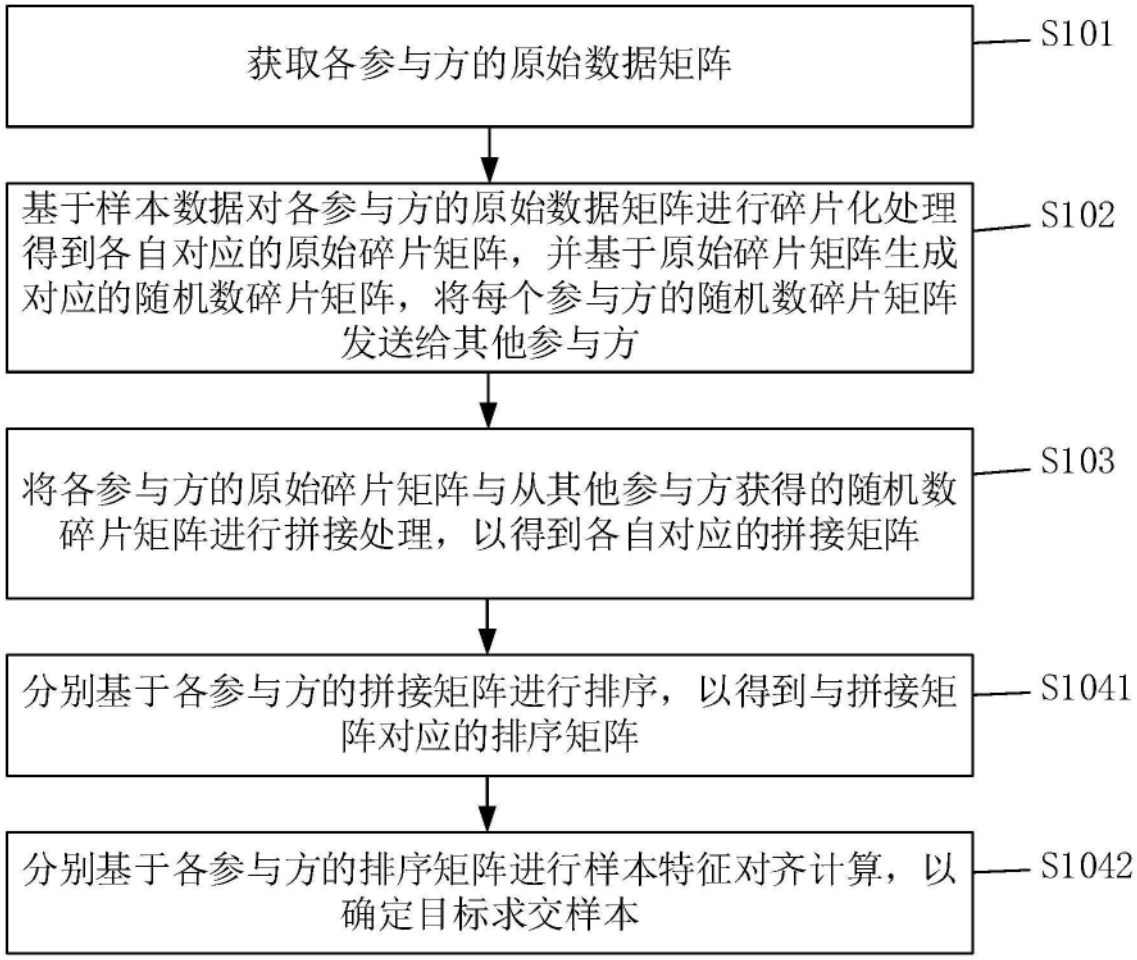

获取各参与方的原始数据矩阵
S101
基于样本数据对各参与方的原始数据矩阵进行碎片化处理得到各自对应的原始碎片矩阵，并基于原始碎片矩阵生成对应的随机数碎片矩阵，将每个参与方的随机数碎片矩阵发送给其他参与方
S102
将各参与方的原始碎片矩阵与从其他参与方获得的随机数碎片矩阵进行拼接处理，以得到各自对应的拼接矩阵
S103
分别基于各参与方的拼接矩阵进行排序，以得到与拼接矩阵对应的排序矩阵
S1041
分别基于各参与方的排序矩阵进行样本特征对齐计算，以确定目标求交样本
S1042

图2

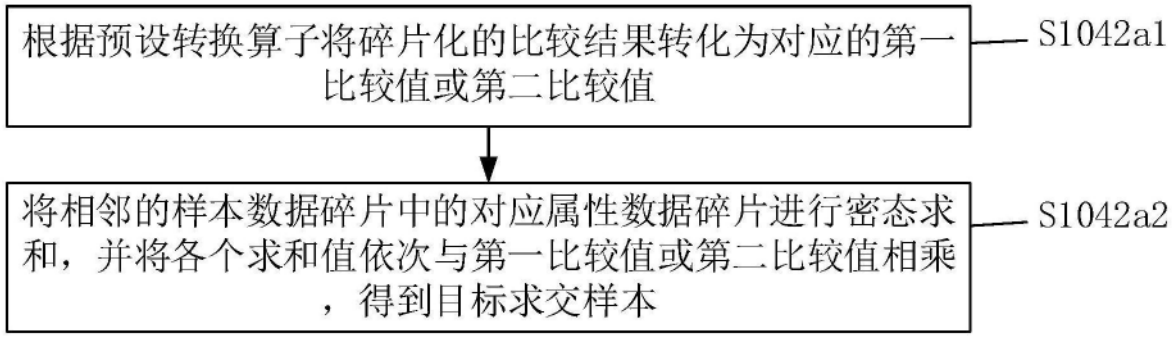

根据预设转换算子将碎片化的比较结果转化为对应的第一比较值或第二比较值
S1042a1
将相邻的样本数据碎片中的对应属性数据碎片进行密态求和，并将各个求和值依次与第一比较值或第二比较值相乘，得到目标求交样本
S1042a2

图3

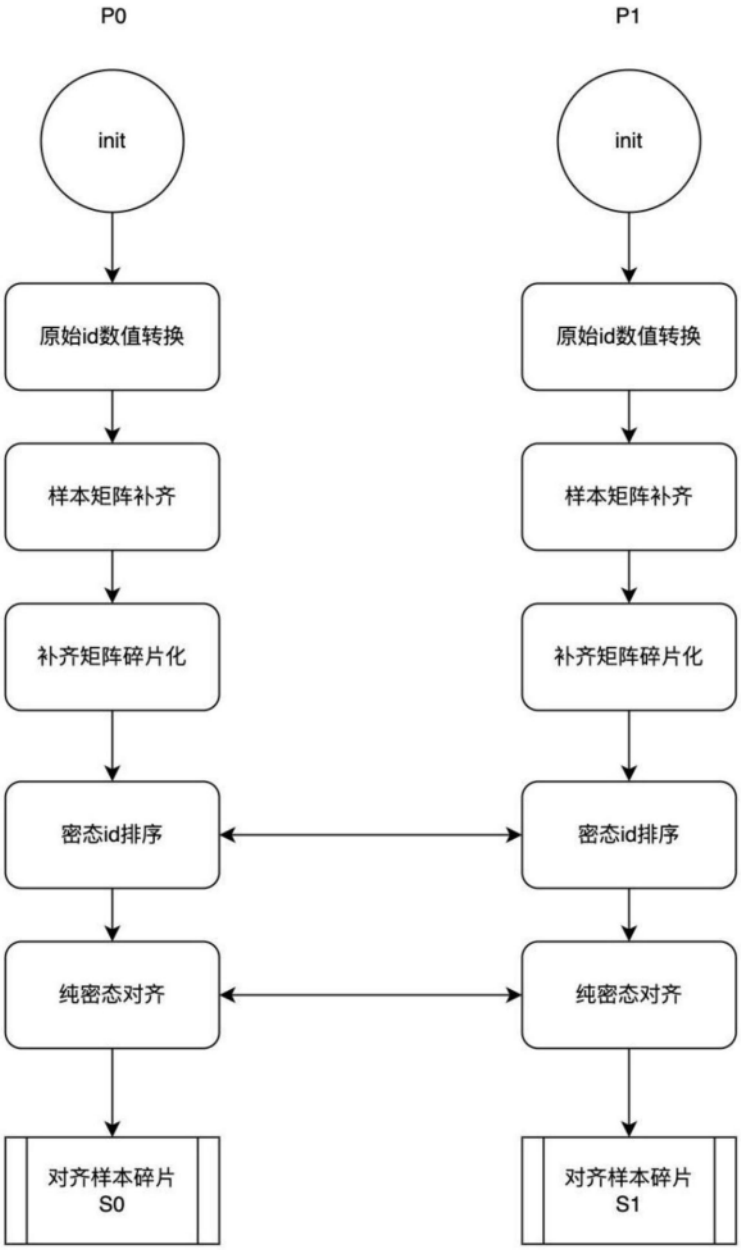

P0
P1
init
init
原始id数值转换
原始id数值转换
样本矩阵补齐
样本矩阵补齐
补齐矩阵碎片化
补齐矩阵碎片化
密态id排序
密态id排序
纯密态对齐
纯密态对齐
对齐样本碎片
S0
对齐样本碎片
S1

图4

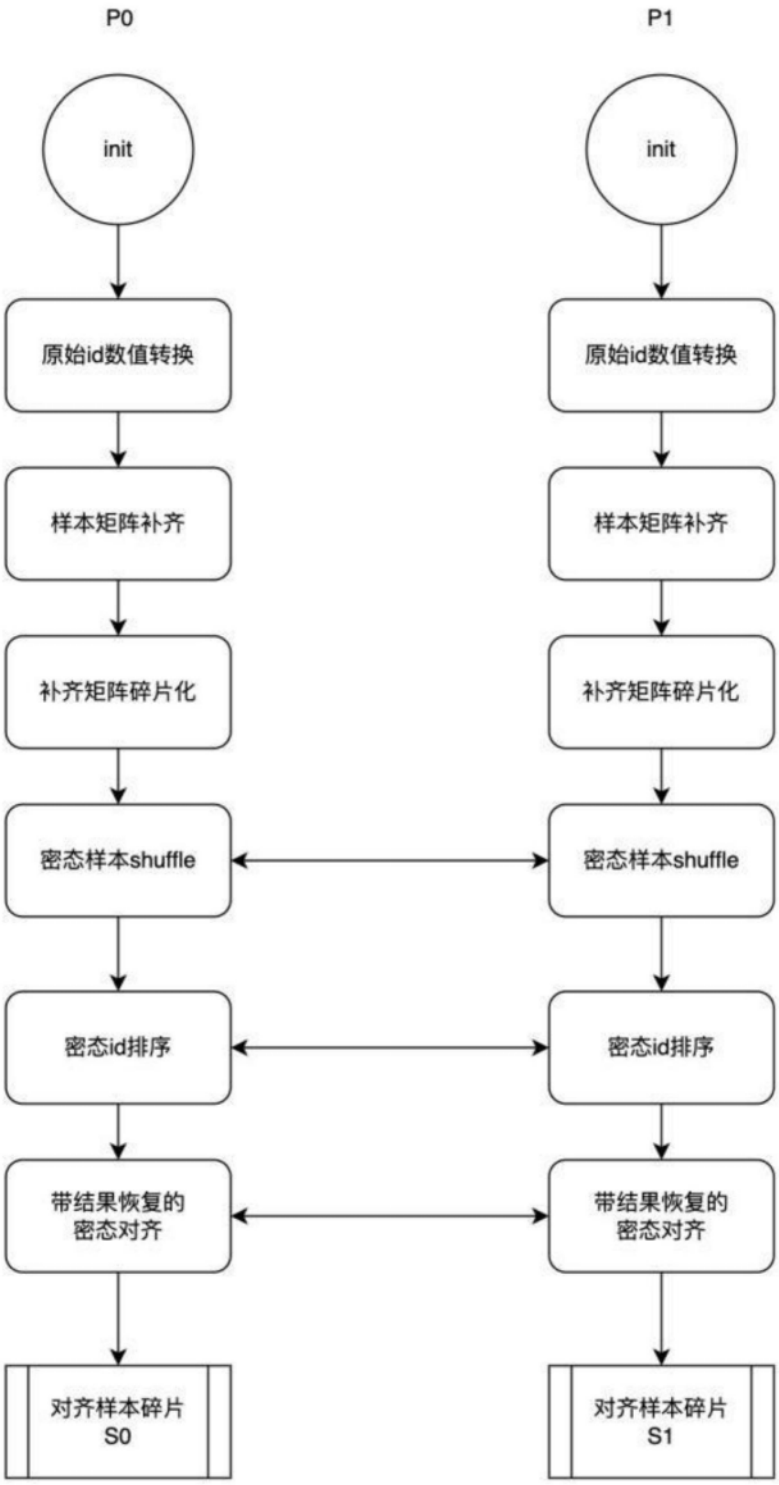

P0
P1
init
init
原始id数值转换
原始id数值转换
样本矩阵补齐
样本矩阵补齐
补齐矩阵碎片化
补齐矩阵碎片化
密态样本shuffle
密态样本shuffle
密态id排序
密态id排序
带结果恢复的
密态对齐
带结果恢复的
密态对齐
对齐样本碎片
S0
对齐样本碎片
S1

图5

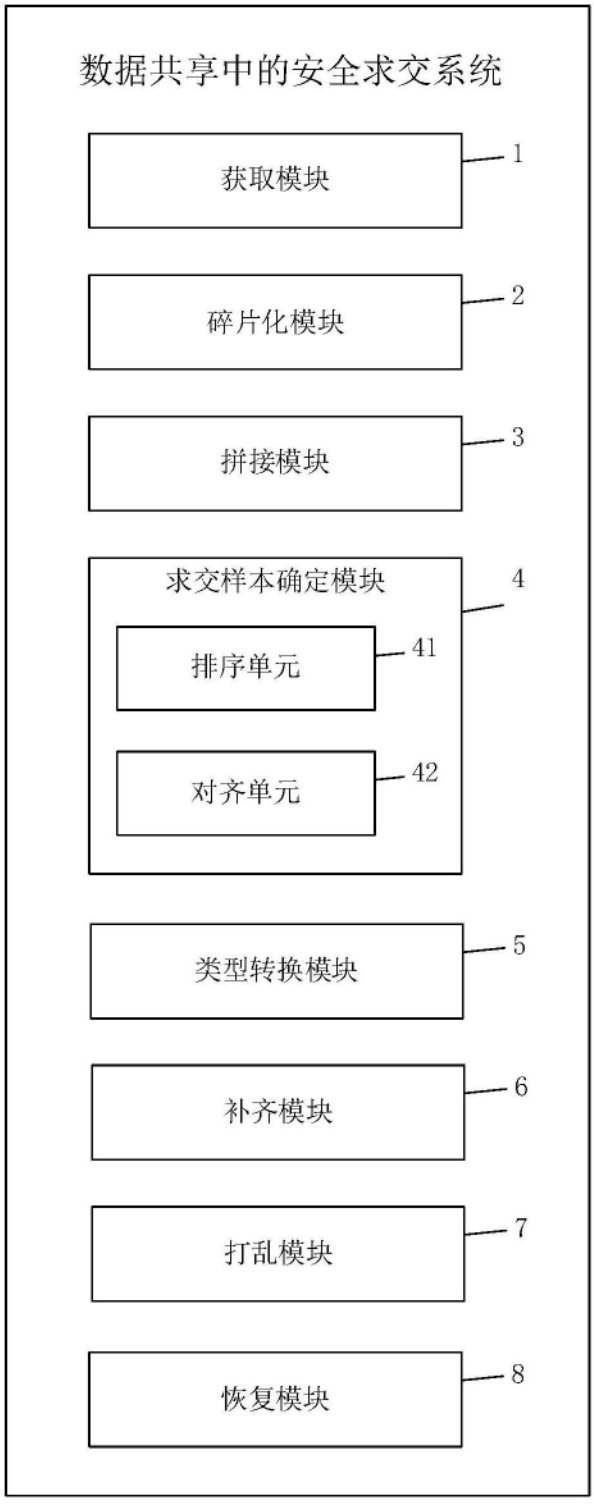

数据共享中的安全求交系统
获取模块
1
碎片化模块
2
拼接模块
3
求交样本确定模块
4
排序单元
41
对齐单元
42
类型转换模块
5
补齐模块
6
打乱模块
7
恢复模块
8

图6

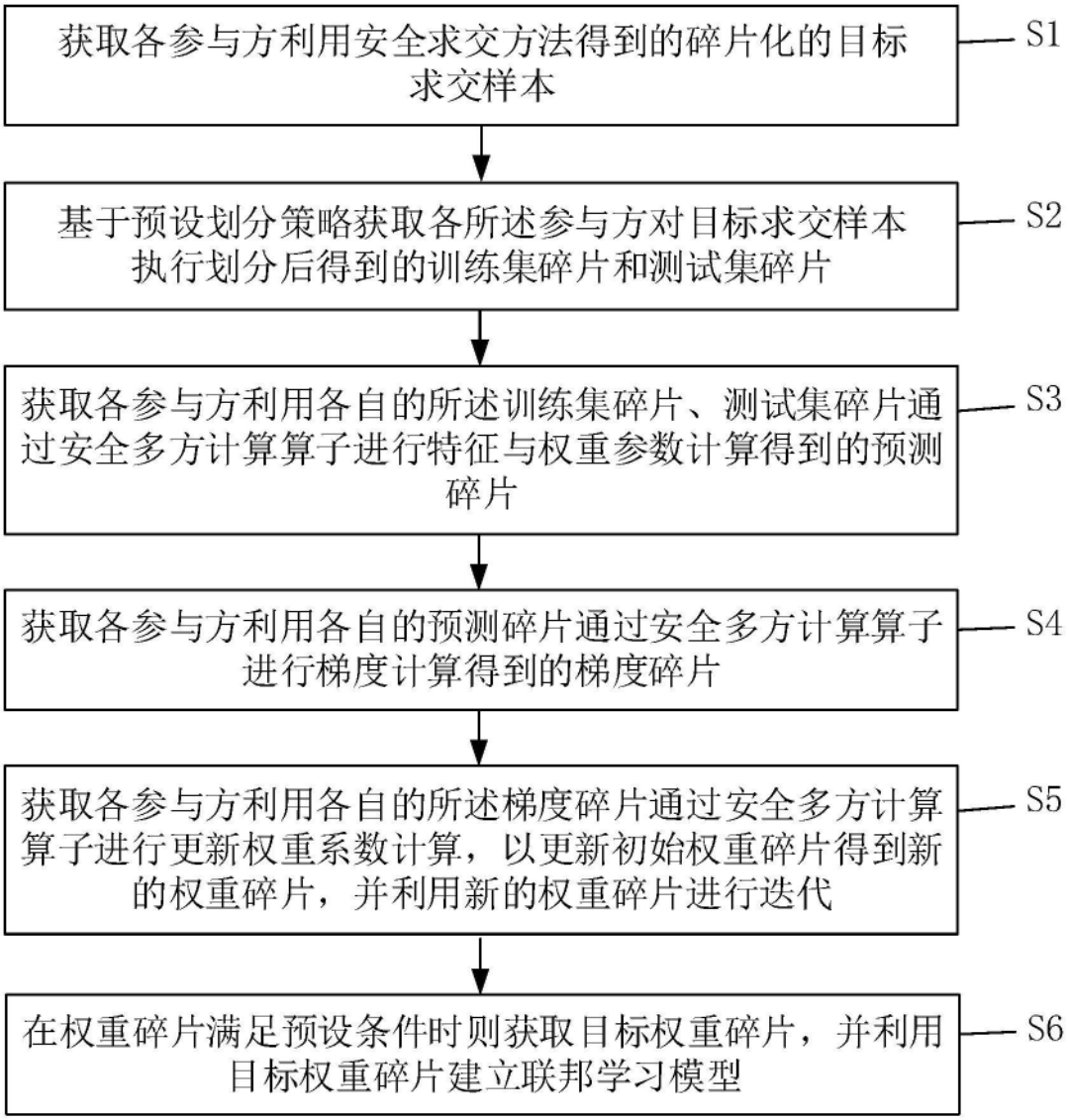

获取各参与方利用安全求交方法得到的碎片化的目标求交样本
S1
基于预设划分策略获取各所述参与方对目标求交样本执行划分后得到的训练集碎片和测试集碎片
S2
获取各参与方利用各自的所述训练集碎片、测试集碎片通过安全多方计算算子进行特征与权重参数计算得到的预测碎片
S3
获取各参与方利用各自的预测碎片通过安全多方计算算子进行梯度计算得到的梯度碎片
S4
获取各参与方利用各自的所述梯度碎片通过安全多方计算算子进行更新权重系数计算，以更新初始权重碎片得到新的权重碎片，并利用新的权重碎片进行迭代
S5
在权重碎片满足预设条件时则获取目标权重碎片，并利用目标权重碎片建立联邦学习模型
S6

图7

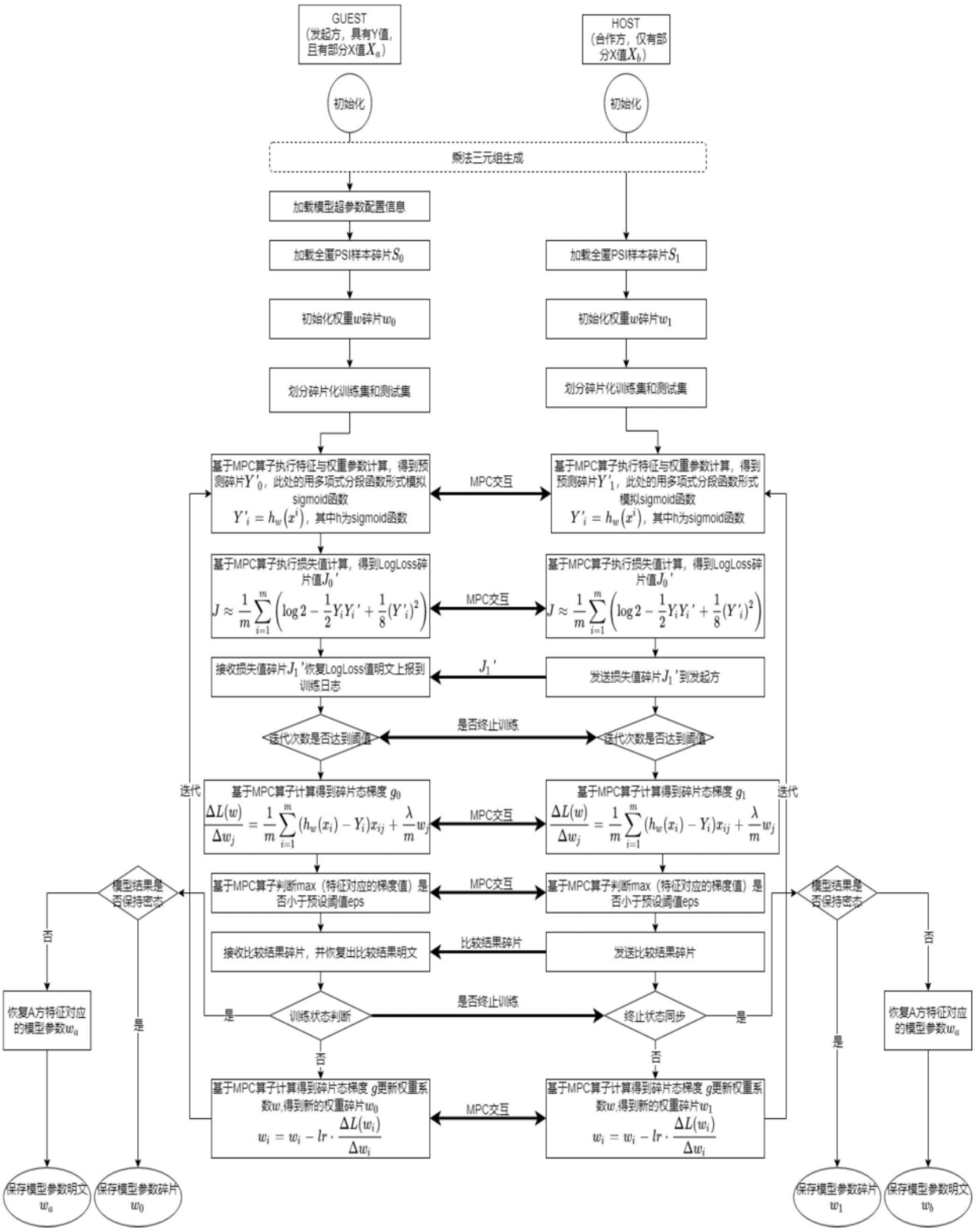

GUEST
（发起方，具有Y值，且有部分X值X_a）
HOST
（合作方，仅有部分X值X_b）
初始化
初始化
乘法三元组生成
加载模型超参数配置信息
加载全置PSI样本碎片S_0
加载全置PSI样本碎片S_1
初始化权重w碎片w_0
初始化权重w碎片w_1
划分碎片化训练集和测试集
划分碎片化训练集和测试集
基于MPC算子执行特征与权重参数计算，得到预测碎片Y'_0，此处的用多项式分段函数形式模拟sigmoid函数
$Y'_i = h_w(x^i)$，其中h为sigmoid函数
MPC交互
基于MPC算子执行特征与权重参数计算，得到预测碎片Y'_1，此处的用多项式分段函数形式模拟sigmoid函数
$Y'_i = h_w(x^i)$，其中h为sigmoid函数
基于MPC算子执行损失值计算，得到LogLoss碎片值J_0'
$J \approx \frac{1}{m}\sum_{i=1}^{m}\left(\log 2 - \frac{1}{2}Y_iY_i' + \frac{1}{8}(Y_i')^2\right)$
MPC交互
基于MPC算子执行损失值计算，得到LogLoss碎片值J_0'
$J \approx \frac{1}{m}\sum_{i=1}^{m}\left(\log 2 - \frac{1}{2}Y_iY_i' + \frac{1}{8}(Y_i')^2\right)$
接收损失值碎片J_1'恢复LogLoss值明文上报到训练日志
J_1'
发送损失值碎片J_1'到发起方
迭代次数是否达到阈值
是否终止训练
迭代次数是否达到阈值
迭代
基于MPC算子计算得到碎片态梯度 g_0
$\frac{\Delta L(w)}{\Delta w_j} = \frac{1}{m}\sum_{i=1}^{m}(h_w(x_i) - Y_i)x_{ij} + \frac{\lambda}{m}w_j$
MPC交互
基于MPC算子计算得到碎片态梯度 g_1
$\frac{\Delta L(w)}{\Delta w_j} = \frac{1}{m}\sum_{i=1}^{m}(h_w(x_i) - Y_i)x_{ij} + \frac{\lambda}{m}w_j$
迭代
模型结果是否保持密态
基于MPC算子判断max（特征对应的梯度值）是否小于预设阈值eps
MPC交互
基于MPC算子判断max（特征对应的梯度值）是否小于预设阈值eps
模型结果是否保持密态
否
接收比较结果碎片，并恢复出比较结果明文
比较结果碎片
发送比较结果碎片
否
恢复A方特征对应的模型参数w_a
是
是
训练状态判断
是否终止训练
终止状态同步
是
恢复A方特征对应的模型参数w_a
否
否
基于MPC算子计算得到碎片态梯度 g更新权重系数w,得到新的权重碎片w_0
$w_i = w_i - lr \cdot \frac{\Delta L(w_i)}{\Delta w_i}$
MPC交互
基于MPC算子计算得到碎片态梯度 g更新权重系数w,得到新的权重碎片w_1
$w_i = w_i - lr \cdot \frac{\Delta L(w_i)}{\Delta w_i}$
保存模型参数明文w_a
保存模型参数碎片w_0
保存模型参数碎片w_1
保存模型参数明文w_b

图8

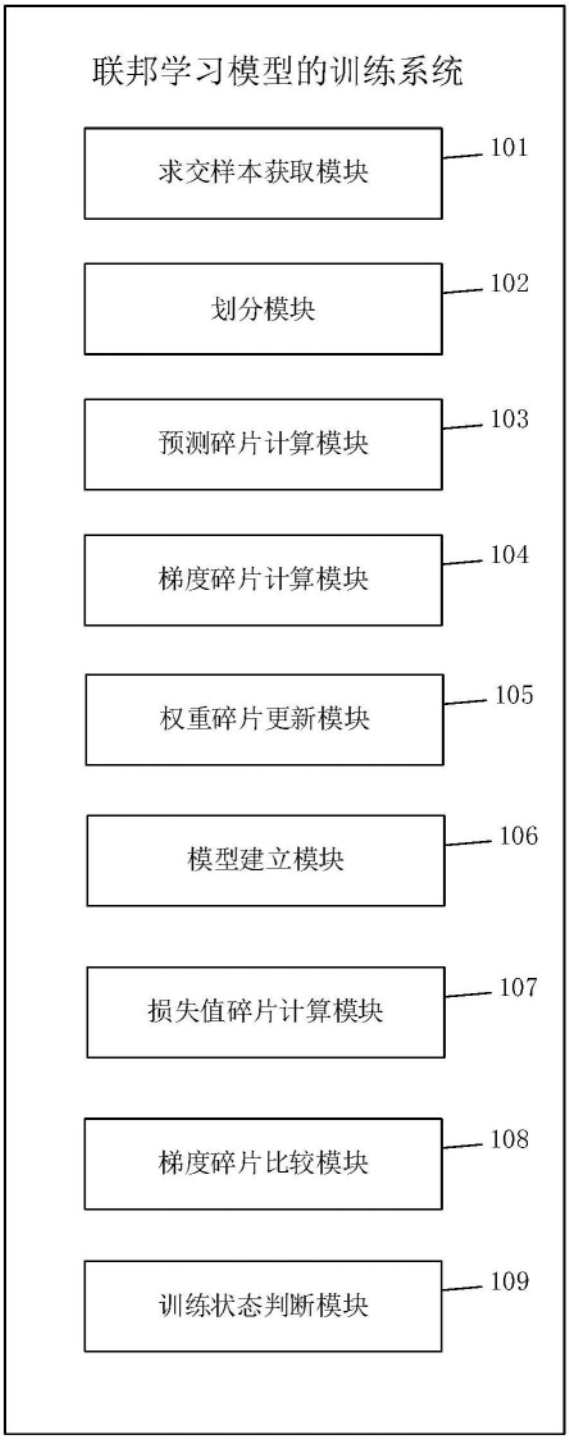

联邦学习模型的训练系统
求交样本获取模块
101
划分模块
102
预测碎片计算模块
103
梯度碎片计算模块
104
权重碎片更新模块
105
模型建立模块
106
损失值碎片计算模块
107
梯度碎片比较模块
108
训练状态判断模块
109

图9

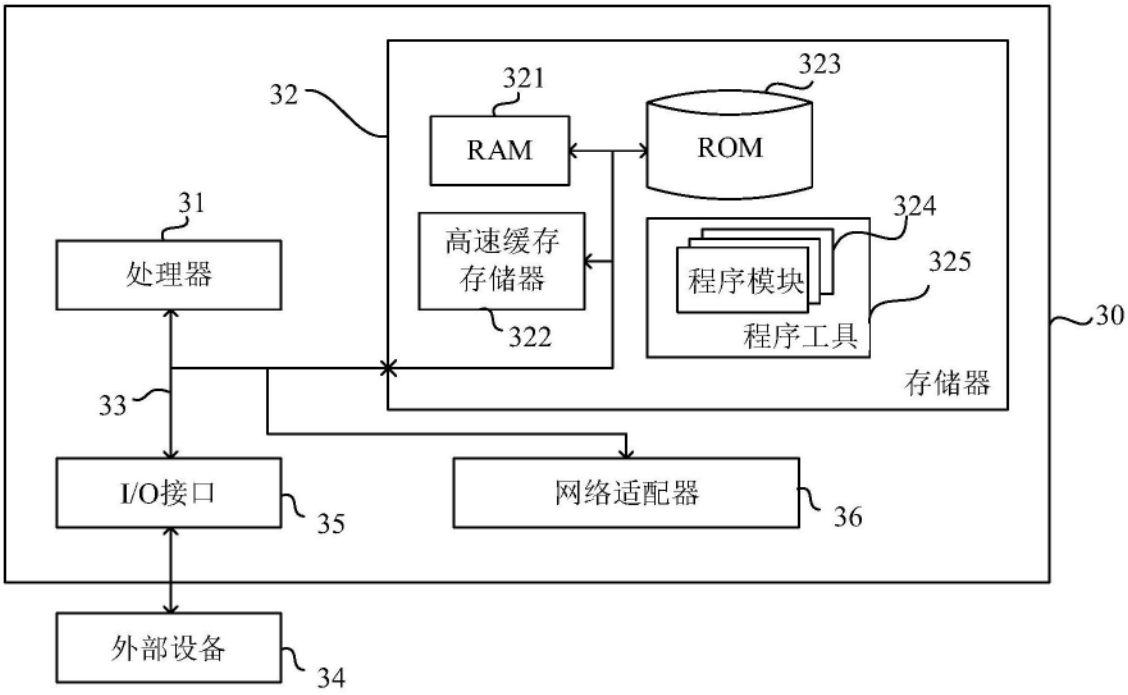

32
321
323
RAM
ROM
31
处理器
高速缓存
存储器
322
324
325
程序模块
程序工具
30
存储器
33
I/O接口
35
网络适配器
36
外部设备
34

图10