(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116305195 A
(43)申请公布日 2023.06.23

(21)申请号 202310114185.8

(22)申请日 2023.02.15

(71)申请人 富算科技(上海)有限公司
地址 200135 上海市浦东新区自由贸易试验区浦东大道1200号2层A区

(72)发明人 尤志强 卞阳 王兆凯 赵东 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.
G06F 21/60(2013.01)
G06F 21/62(2013.01)
G06F 17/16(2006.01)
G06N 20/10(2019.01)

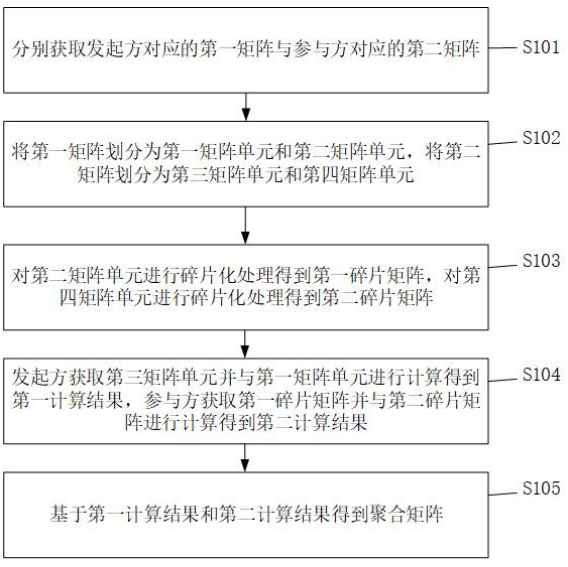
权利要求书2页 说明书13页 附图13页

(54)发明名称

多方安全计算、学习模型的训练方法、系统、设备及介质

(57)摘要

本发明公开了一种多方安全计算、学习模型的训练方法、系统、设备及介质，其中多方安全计算包括分别获取发起方对应的第一矩阵与参与方对应的第二矩阵；将第一矩阵划分为第一矩阵单元和第二矩阵单元，将第二矩阵划分为第三矩阵单元和第四矩阵单元；对第二矩阵单元进行碎片化处理得到第一碎片矩阵，对第四矩阵单元进行碎片化处理得到第二碎片矩阵；发起方获取第三矩阵单元并与第一矩阵单元进行计算得到第一计算结果，参与方获取第一碎片矩阵并与第二碎片矩阵进行计算得到第二计算结果；基于第一计算结果和第二计算结果得到聚合矩阵，能够将直方图部分进行计算效率和通讯量优化，在保证数据安全的前提下能大大提升模型训练性能。



CN 116305195 A

1.一种多方安全计算方法,其特征在于，应用于至少一个发起方与至少一个参与方之间数据共享场景中,所述方法包括：

分别获取所述发起方对应的第一矩阵与所述参与方对应的第二矩阵；

将所述第一矩阵划分为第一矩阵单元和第二矩阵单元,将所述第二矩阵划分为第三矩阵单元和第四矩阵单元；

对所述第二矩阵单元进行碎片化处理得到第一碎片矩阵,对所述第四矩阵单元进行碎片化处理得到第二碎片矩阵；

所述发起方获取所述第三矩阵单元并与所述第一矩阵单元进行计算得到第一计算结果,所述参与方获取所述第一碎片矩阵并与所述第二碎片矩阵进行计算得到第二计算结果；

基于所述第一计算结果和所述第二计算结果得到聚合矩阵。

2.如权利要求1所述的多方安全计算方法,其特征在于,所述将所述第一矩阵划分为第一矩阵单元和第二矩阵单元包括：

基于预设划分比例将所述第一矩阵进行划分得到所述第一矩阵单元和所述第二矩阵单元。

3.如权利要求1所述的多方安全计算方法,其特征在于,所述将所述第二矩阵划分为第三矩阵单元和第四矩阵单元包括：

将所述第二矩阵按列进行打乱处理,得到打乱后的第二矩阵,并记录打乱处理对应的乱序索引；

基于预设划分比例将打乱后的所述第二矩阵进行划分,得到所述第三矩阵单元和所述第四矩阵单元。

4.如权利要求1-3任一项所述的多方安全计算方法,其特征在于,所述发起方获取所述第三矩阵单元并与所述第一矩阵单元进行计算得到第一计算结果包括：

将所述第三矩阵单元进行压缩处理后发送至所述发起方；

所述发起方将压缩后的所述第三矩阵单元与所述第一矩阵单元进行矩阵乘计算,得到所述第一计算结果。

5.如权利要求1-3任一项所述的多方安全计算方法,其特征在于,所述参与方获取所述第一碎片矩阵并与所述第二碎片矩阵进行计算得到第二计算结果包括：

所述参与方将所述第一碎片矩阵与所述第二碎片矩阵进行矩阵乘计算,得到第三碎片矩阵；

将所述第三碎片矩阵进行恢复处理,得到所述第二计算结果。

6.如权利要求3所述的多方安全计算方法,其特征在于,所述基于所述第一计算结果和所述第二计算结果得到聚合矩阵包括：

对所述第一计算结果和所述第二计算结果进行聚合处理得到聚合结果；

利用所述乱序索引对所述聚合结果进行重排序处理,得到所述聚合矩阵。

7.如权利要求6所述的多方安全计算方法,其特征在于,所述对所述第一计算结果和所述第二计算结果进行聚合得到聚合结果包括：

对所述第一计算结果、所述第二计算结果和预设噪声矩阵按位进行聚合处理得到所述聚合结果。

8.一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:

分别获取发起方和参与方的求交特征数据集;

利用所述求交特征数据集构建XGBoost树模型;

通过如权利要求1-7中任一项所述的多方安全计算方法得到的聚合矩阵以计算所述XGBoost树模型的最优分割点;

利用所述最优分割点更新所述XGBoost树模型;

将待预测的数据输入至更新后的所述XGBoost树模型进行预测,得到预测结果。

9.一种多方安全计算系统,其特征在于,所述系统包括:

获取模块,用于分别获取发起方对应的第一矩阵与参与方对应的第二矩阵;

划分模块,用于将所述第一矩阵划分为第一矩阵单元和第二矩阵单元,将所述第二矩阵划分为第三矩阵单元和第四矩阵单元;

碎片化模块,用于对所述第二矩阵单元进行碎片化处理得到第一碎片矩阵,对所述第四矩阵单元进行碎片化处理得到第二碎片矩阵;

计算模块,用于通过所述发起方获取所述第三矩阵单元并与所述第一矩阵单元进行计算得到第一计算结果,通过所述参与方获取所述第一碎片矩阵并与所述第二碎片矩阵进行计算得到第二计算结果;

聚合模块,用于基于所述第一计算结果和所述第二计算结果得到聚合矩阵。

10.一种联邦学习模型的训练系统,其特征在于,所述训练系统包括:

数据集获取模块,用于分别获取发起方和参与方的求交特征数据集;

模型构建模块,用于利用所述求交特征数据集构建XGBoost树模型;

分割点计算模块,用于通过如权利要求9中所述的多方安全计算系统得到的聚合矩阵以计算所述XGBoost树模型的最优分割点;

模型更新模块,用于利用所述最优分割点更新所述XGBoost树模型;

模型预测模块,用于将待预测的数据输入至更新后的所述XGBoost树模型进行预测,得到预测结果。

11.一种电子设备,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1-7中任一项所述的多方安全计算方法;或,实现如权利要求8所述的联邦学习模型的训练方法。

12.一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1-7中任一项所述的多方安全计算方法;或,实现如权利要求8所述的联邦学习模型的训练方法。

多方安全计算、学习模型的训练方法、系统、设备及介质

技术领域

[0001] 本发明涉及多方安全计算的技术领域，特别涉及一种多方安全计算、学习模型的训练方法、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。

[0003] 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。

[0004] XGBoost(Exterme Gradient Boosting)全称为极限梯度提升树模型，是一种基于决策树的集成机器学习算法，因其模型预测能力强，在工业界被广泛使用，比如应用在广告推荐、金融风控等业务场景。然而将该算法应用于联邦学习场景，由于对该算法的联邦化设计目前还处于不成熟阶段，普遍来说当前的联邦学习XGBoost算法训练的通信量比较大，导致训练耗时过长，不能满足业界需求。

发明内容

[0005] 本发明要解决的技术问题是为了克服现有技术中的联邦学习模型训练耗时长、效率低的缺陷，提供一种多方安全计算、学习模型的训练方法、系统、设备及介质。

[0006] 本发明是通过下述技术方案来解决上述技术问题：

本发明提供一种多方安全计算方法，应用于至少一个发起方与至少一个参与方之间数据共享场景中，所述方法包括：

分别获取所述发起方对应的第一矩阵与所述参与方对应的第二矩阵；

将所述第一矩阵划分为第一矩阵单元和第二矩阵单元，将所述第二矩阵划分为第三矩阵单元和第四矩阵单元；

对所述第二矩阵单元进行碎片化处理得到第一碎片矩阵，对所述第四矩阵单元进行碎片化处理得到第二碎片矩阵；

所述发起方获取所述第三矩阵单元并与所述第一矩阵单元进行计算得到第一计算结果，所述参与方获取所述第一碎片矩阵并与所述第二碎片矩阵进行计算得到第二计算结果；

基于所述第一计算结果和所述第二计算结果得到聚合矩阵。

[0007] 较佳地,所述将所述第一矩阵划分为第一矩阵单元和第二矩阵单元包括:

基于预设划分比例将所述第一矩阵进行划分得到所述第一矩阵单元和所述第二矩阵单元。

[0008] 较佳地,所述将所述第二矩阵划分为第三矩阵单元和第四矩阵单元包括:

将所述第二矩阵按列进行打乱处理,得到打乱后的第二矩阵,并记录打乱处理对应的乱序索引;

基于预设划分比例将打乱后的所述第二矩阵进行划分,得到所述第三矩阵单元和所述第四矩阵单元。

[0009] 较佳地,所述发起方获取所述第三矩阵单元并与所述第一矩阵单元进行计算得到第一计算结果包括:

将所述第三矩阵单元进行压缩处理后发送至所述发起方;

所述发起方将压缩后的所述第三矩阵单元与所述第一矩阵单元进行矩阵乘计算,得到所述第一计算结果。

[0010] 较佳地,所述参与方获取所述第一碎片矩阵并与所述第二碎片矩阵进行计算得到第二计算结果包括:

所述参与方将所述第一碎片矩阵与所述第二碎片矩阵进行矩阵乘计算,得到第三碎片矩阵;

将所述第三碎片矩阵进行恢复处理,得到所述第二计算结果。

[0011] 较佳地,所述基于所述第一计算结果和所述第二计算结果得到聚合矩阵包括:

对所述第一计算结果和所述第二计算结果进行聚合处理得到聚合结果;

利用所述乱序索引对所述聚合结果进行重排序处理,得到所述聚合矩阵。

[0012] 较佳地,所述对所述第一计算结果和所述第二计算结果进行聚合得到聚合结果包括:

对所述第一计算结果、所述第二计算结果和预设噪声矩阵按位进行聚合处理得到所述聚合结果。

[0013] 本发明还提供一种联邦学习模型的训练方法,所述训练方法包括:

分别获取发起方和参与方的求交特征数据集;

利用所述求交特征数据集构建XGBoost树模型;

通过如上所述的多方安全计算方法得到的聚合矩阵以计算所述XGBoost树模型的最优分割点;

利用所述最优分割点更新所述XGBoost树模型;

将待预测的数据输入至更新后的所述XGBoost树模型进行预测,得到预测结果。

[0014] 本发明还提供一种多方安全计算系统,所述系统包括:

获取模块,用于分别获取所述发起方对应的第一矩阵与所述参与方对应的第二矩阵;

划分模块,用于将所述第一矩阵划分为第一矩阵单元和第二矩阵单元,将所述第二矩阵划分为第三矩阵单元和第四矩阵单元;

碎片化模块,用于对所述第二矩阵单元进行碎片化处理得到第一碎片矩阵,对所述第四矩阵单元进行碎片化处理得到第二碎片矩阵;

计算模块，用于通过所述发起方获取所述第三矩阵单元并与所述第一矩阵单元进行计算得到第一计算结果，通过所述参与方获取所述第一碎片矩阵并与所述第二碎片矩阵进行计算得到第二计算结果；

聚合模块，用于基于所述第一计算结果和所述第二计算结果得到聚合矩阵。

[0015] 较佳地，所述划分模块，还用于基于预设划分比例将所述第一矩阵进行划分得到所述第一矩阵单元和所述第二矩阵单元。

[0016] 较佳地，所述划分模块，还用于将所述第二矩阵按列进行打乱处理，得到打乱后的第二矩阵，并记录打乱处理对应的乱序索引；

基于预设划分比例将打乱后的所述第二矩阵进行划分，得到所述第三矩阵单元和所述第四矩阵单元。

[0017] 较佳地，所述计算模块，还用于将所述第三矩阵单元进行压缩处理后发送至所述发起方；

所述发起方将压缩后的所述第三矩阵单元与所述第一矩阵单元进行矩阵乘计算，得到所述第一计算结果。

[0018] 较佳地，所述计算模块，还用于所述参与方将所述第一碎片矩阵与所述第二碎片矩阵进行矩阵乘计算，得到第三碎片矩阵；

将所述第三碎片矩阵进行恢复处理，得到所述第二计算结果。

[0019] 较佳地，所述聚合模块，还用于对所述第一计算结果和所述第二计算结果进行聚合处理得到聚合结果；

利用所述乱序索引对所述聚合结果进行重排序处理，得到所述聚合矩阵。

[0020] 较佳地，所述聚合模块，还用于对所述第一计算结果、所述第二计算结果和预设噪声矩阵按位进行聚合处理得到所述聚合结果。

[0021] 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：

数据集获取模块，用于分别获取发起方和参与方的求交特征数据集；

模型构建模块，用于利用所述求交特征数据集构建XGBoost树模型；

分割点计算模块，用于通过如上所述的多方安全计算系统得到的聚合矩阵以计算所述XGBoost树模型的最优分割点；

模型更新模块，用于利用所述最优分割点更新所述XGBoost树模型；

模型预测模块，用于将待预测的数据输入至更新后的所述XGBoost树模型进行预测，得到预测结果。

[0022] 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多方安全计算方法；或，实现如上所述的联邦学习模型的训练方法。

[0023] 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多方安全计算方法；或，实现如上所述的联邦学习模型的训练方法。

[0024] 本发明的积极进步效果在于：分别获取发起方对应的第一矩阵与参与方对应的第二矩阵；将第一矩阵划分为第一矩阵单元和第二矩阵单元，将第二矩阵划分为第三矩阵单元和第四矩阵单元；对第二矩阵单元进行碎片化处理得到第一碎片矩阵，对第四矩阵单元

进行碎片化处理得到第二碎片矩阵;发起方获取第三矩阵单元并与第一矩阵单元进行计算得到第一计算结果,参与方获取第一碎片矩阵并与第二碎片矩阵进行计算得到第二计算结果;基于第一计算结果和第二计算结果得到聚合矩阵,能够将直方图部分进行计算效率和通讯量优化,从而通过采用混合态多方安全计算机制,在保护样本数据分布的前提下,大幅度降低计算耗时。

附图说明

[0025] 图1为现有XGBoost 在Host方的直方图碎片态矩阵计算方法。

[0026] 图2为本发明实施例提供的多方安全计算方法的第一流程示意图。

[0027] 图3为本发明实施例提供的多方安全计算方法的第二流程示意图。

[0028] 图4为本发明实施例提供的多方安全计算方法的打乱过程示意图。

[0029] 图5为本发明实施例提供的多方安全计算方法的第三流程示意图。

[0030] 图6为本发明实施例提供的多方安全计算方法的第四流程示意图。

[0031] 图7为本发明实施例提供的多方安全计算方法的第五流程示意图。

[0032] 图8a为本发明实施例提供的多方安全计算方法的第六流程示意图的第一部分。

[0033] 图8b为本发明实施例提供的多方安全计算方法的第六流程示意图的第二部分。

[0034] 图9为本发明实施例提供的多方安全计算系统的模块示意图。

[0035] 图10为本发明实施例提供的联邦学习模型的训练方法的第一流程示意图。

[0036] 图11a为本发明实施例提供的联邦学习模型的训练方法的第二流程示意图的第一部分。

[0037] 图11b为本发明实施例提供的联邦学习模型的训练方法的第二流程示意图的第二部分。

[0038] 图11c为本发明实施例提供的联邦学习模型的训练方法的第二流程示意图的第三部分。

[0039] 图11d为本发明实施例提供的联邦学习模型的训练方法的第二流程示意图的第四部分。

[0040] 图12为本发明实施例提供的联邦学习模型的训练系统的模块示意图。

[0041] 图13为本发明实施例提供的实现多方安全计算方法或联邦学习模型的训练方法的电子设备的结构示意图。

具体实施方式

[0042] 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。

[0043] 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。

[0044] 在本申请中提及“实施例”意味着,结合实施例描述的特定特征、结构或特性可以

包含在本申请的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例，也不是与其它实施例互斥的独立的或备选的实施例。本领域技术人员显式地和隐式地理解的是，本申请所描述的实施例可以与其它实施例相结合。

[0045] 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。

[0046] 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。

[0047] 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。

[0048] XGBoost是一种基于决策树的集成机器学习算法，因其模型预测能力强，在工业界被广泛使用，比如应用在广告推荐、金融风控等业务场景。然而将该算法应用于联邦学习场景中时，由于对该算法的联邦化设计目前还处于不成熟阶段，普遍来说当前的联邦学习XGBoost算法训练存在以下几个缺点：

(1)交互数据过大

在MPC XGBoost中假设一个数据集有40万样本数据，每条数据有600个特征，那么在直方图计算过程中会有255G的数据传输量，如图1示出了一个原始MPC XGBoost 在Host方直方图碎片态矩阵计算过程。这还仅仅是构建一张直方图的，这种数据级别的数据交互使得难以在大样本上进行高效的模型训练。

[0049] (2)计算性能不高

原始MPC XGBoost 中核心交互部分采用全碎片态进行计算，使得计算性能过低。主要是为了保证数据安全性，在矩阵乘法时没有利用稀疏化矩阵可加速计算的特性。还是以40万训练样本集数据为例，其中每条数据有600个特征，每个特征分50个桶，整个过程需要执行240亿次乘法运算和239亿9994万次加法运算。

[0050] 由此可见，当前的联邦学习XGBoost算法训练的交互数据过大且计算性能不高，导致训练耗时过长，从而满足不了业界的需求。

[0051] 基于此，如图2所示，本实施例提供一种多方安全计算方法，应用于至少一个发起方与至少一个参与方之间数据共享场景中，在本实施例中，以一个发起方(GUEST)和一个参与方(HOST)为例进行说明，本实施例的多方安全计算方法包括：

S101、分别获取发起方对应的第一矩阵与参与方对应的第二矩阵。

[0052] 在一种可选地实施方式，第一矩阵可以为梯度矩阵，其可以包括一阶梯度和二阶梯度，第二矩阵可以为稀疏矩阵，本实施例对此不作限制。

[0053] 还需要说明的是，第一矩阵的列数与第二矩阵的行数可以对应。

[0054] S102、将第一矩阵划分为第一矩阵单元和第二矩阵单元，将第二矩阵划分为第三矩阵单元和第四矩阵单元。

[0055] 可选地，步骤S102可以作为第一阶段，其主要过程为将发起方(GUEST)的第一矩阵(梯度矩阵)分割成两部分，即划分为第一矩阵单元gl和第二矩阵单元gr，将参与方(HOST)的第二矩阵(稀疏矩阵)也同样分割成两部分，即划分为第三矩阵单元hl和第四矩阵单元hr。

[0056] S103、对第二矩阵单元进行碎片化处理得到第一碎片矩阵，对第四矩阵单元进行碎片化处理得到第二碎片矩阵。

[0057] S104、发起方获取第三矩阵单元并与第一矩阵单元进行计算得到第一计算结果，参与方获取第一碎片矩阵并与第二碎片矩阵进行计算得到第二计算结果。

[0058] 可选地，步骤S103和步骤S104可以作为第二阶段，将发起方的第二矩阵单元gr进行碎片化后得到第一碎片矩阵<gr>后秘密共享，将参与方的第四矩阵单元hr进行碎片化得到第二碎片矩阵<hr>后秘密共享。

[0059] 将参与方的第三矩阵单元hl压缩后传递给发起方，发起方在本地进行第一矩阵单元gl和第三矩阵单元hl计算得到第一计算结果，即直方图矩阵histo1。

[0060] 参与方获取第一碎片矩阵<gr>并与第二碎片矩阵<hr>进行计算得到第二计算结果，即直方图矩阵<histo2>，目前<histo2>是碎片状态。

[0061] S105、基于第一计算结果和第二计算结果得到聚合矩阵。

[0062] 可选地，步骤S105可以作为第三阶段，发起方将第一计算结果histo1发送给参与方，参与方将第二计算结果<histo2>恢复为histo2后得到聚合矩阵，即最终的Host方直方图，并同步至发起方。

[0063] 从而，本实施例将原始的发起方的第一矩阵和参与方秘密共享第二矩阵乘法得到HOST直方图拆分为三个阶段，通过这三个阶段的优化调整可以降低发起方和参与方的数据传输量，提高了计算效率。

[0064] 需要说明的是，发起方的梯度矩阵的shape(形状)为(2，m)，参与方的稀疏矩阵shape为(m，f*k)，其中m为样本量，f为特征数，k为每个特征分桶数，本实施例以m为6为例。

[0065] 作为本实施例的一种可选地实施方式，本实施例的步骤S102包括：

S102a、基于预设划分比例将第一矩阵进行划分得到第一矩阵单元和第二矩阵单元。

[0066] 可选地，如图3所示，基于预设划分比例将发起方的第一矩阵(梯度矩阵)分割成两部分，然后发起方对梯度矩阵在样本维度进行切分，即将Guest梯度矩阵的m列切分成两部分，得到第一矩阵单元gl和第二矩阵单元gr。

[0067] 需要说明的是，对比原始处理方式，本实施例在第二阶段设分割比例为95%和5%，这样在大数据样本训练的情况下传输数据量下降20多倍，使得数据传输瓶颈得到缓解，整体运行效率会有数倍提升。本实施例使用多态计算方式可将95%稀疏矩阵进行高效的矩阵乘法，还是以40万训练样本集数据为例，其中每条数据有600个特征，每个特征分50个桶，现有方案乘法过程需要执行240亿次乘法运算和239.9994亿次加法运算，而本实施例只要执行4.8亿次乘法和4.799988亿次加法计算，极大提升了计算性能。

[0068] 作为本实施例的另一种可选地实施方式，如图3所示，本实施例的步骤S102还包括：

S102b1、将第二矩阵按列进行打乱处理，得到打乱后的第二矩阵，并记录打乱处理

对应的乱序索引。

[0069] S102b2、基于预设划分比例将打乱后的第二矩阵进行划分，得到第三矩阵单元和第四矩阵单元。

[0070] 可选地，如图4所示，参与方首先会将第二矩阵(稀疏矩阵)按列打乱，例如，由原来列序号[0,1,2,3,4,5,6,7,8]打乱为[8,1,4,3,2,6,5,7,0]，并记录打乱处理对应的乱序索引，然后参与方分对特征分桶稀疏矩阵在样本维度进行切分，即将Host特征分桶稀疏矩阵m行切分为两部分，得到第三矩阵单元hl和第四矩阵单元hr。

[0071] 作为本实施例的一种可选地实施方式，如图5所示，本实施例的S104包括：

S104a1、将第三矩阵单元进行压缩处理后发送至发起方。

[0072] S104a2、发起方将压缩后的第三矩阵单元与第一矩阵单元进行矩阵乘计算，得到第一计算结果。

[0073] 将Host方的乱序的第三矩阵单元hl压缩后传递给Guest方，Guest方在本地进行第一矩阵单元gl和第三矩阵单元hl乘法，其中，第一矩阵单元的梯度矩阵shape为(2,4)，乱序的第三矩阵单元的稀疏矩阵shape为(4,9)，相乘得到shape为(2,9)的直方图矩阵histo1，即第一计算结果。

[0074] 作为本实施例的另一种可选地实施方式，如图6所示，本实施例的S104包括：

S104b1、参与方将第一碎片矩阵与第二碎片矩阵进行矩阵乘计算，得到第三碎片矩阵。

[0075] S104b2、将第三碎片矩阵进行恢复处理，得到第二计算结果。

[0076] 发起方将第一碎片矩阵<gr>秘密共享给参与方，参与方将第二碎片矩阵<hr>秘密共享给发起方，然后第一碎片矩阵<gr>和第二碎片矩阵<hr>进行mpc乘法，梯度矩阵shape为(2,2)，乱序的稀疏矩阵shape为(2,9)，相乘得到shape为(2,9)的直方图矩阵<histo2>，即第二计算结果，目前<histo2>是碎片状态。

[0077] 通过上述过程，可以降低训练过程中通讯量，联邦学习xgboost主要瓶颈在于Host方直方图的计算，之所以为瓶颈是因为保证Guest方梯度矩阵和Host方特征数据矩阵不被泄露的情况下进行矩阵乘法，原始做法是将Host方特征数据矩阵分桶稀疏化然后直接与Guest方进行mpc乘法，但这种方式会有巨大的数据传输开销，不可在大数据样本进行执行。而本方案使用两种矩阵方式进行组合执行矩阵乘法，在保证数据不泄露的前提下大大下降数据传输量。

[0078] 作为本实施例的一种可选地实施方式，如图7所示，本实施例的S105包括：

S1051、对第一计算结果和第二计算结果进行聚合处理得到聚合结果。

[0079] 作为本实施例的另一种可选地实施方式，本实施例的S1051包括：

S10511、对第一计算结果、第二计算结果和预设噪声矩阵按位进行聚合处理得到聚合结果。

[0080] S1052、利用乱序索引对聚合结果进行重排序处理，得到聚合矩阵。

[0081] 发起方将第一计算结果histo1发送给参与方，参与方将第二计算结果<histo2>为恢复histo2，然后构造一个和第一计算结果histo1同shape大小的噪声矩阵，最终将histo1，histo2和噪声矩阵按位相加得到参与方的直方图r_histo，即聚合结果，由于r_histo还是乱序的，用第一阶段的乱序索引恢复r_histo得到最终的Host方直方图，即聚合

矩阵。

[0082] 由此,本实施例通过多态混合计算技术在稀疏矩阵上的运算加速,在进行稀疏矩阵计算时,HOST数据拥有方可以通过按列打乱然后压缩再发给GUEST计算方, GUEST计算方在压缩数据上进行加速计算后回传给数据拥有方,HOST数据拥有方恢复顺序后通过噪声辅助来掩盖原始数据,大大提高了多方计算的安全性,使得能够利用稀疏矩阵的特性,将按列打乱后的稀疏矩阵进行压缩,乘法运算时免去了不必要的乘0计算,整个过程计算量成倍下降,避免了数据安全性导致计算开销较大的问题。

[0083] 在一个例子中,如图8a和图8b所示(图8a作为第一部分,图8b作为第二部分,共同构成一个完整流程图),本实施例的多方安全计算方法的第一阶段主要过程为Host秘密共享稀疏矩阵按列打乱顺序,然后和Guest梯度矩阵都按比例分割成两部分,如图中上部分方框内内容所示,Guest梯度矩阵的shape为(2,m),Host稀疏矩阵shape为(m,f*k),其中m为样本量,f为特征数,k为每个特征分桶数。这里Host首先会将稀疏矩阵按列打乱,图中由原来列序号[0,1,2,3,4,5,6,7,8]打乱为[8,1,4,3,2,6,5,7,0]。然后Guest方和Host方分别对梯度矩阵和特征分桶稀疏矩阵在样本维度进行切分,即将Guest梯度矩阵的m列切分成两部分,将Host特征分桶稀疏矩阵m行切分为两部分,图中m为6,切分成了4和2,为了后续表述将切分后的梯度矩阵表示为gl和gr,将切分后的特征分桶稀疏矩阵表示为hl和hr。

[0084] 第二阶段主要过程为分别将第一阶段得到的切分数据进行不同方式的矩阵乘法处理。第一部分将Host方的乱序的稀疏矩阵hl压缩后传递给Guest方,Guest方在本地进行梯度矩阵gl和稀疏矩阵hl乘法,如图6中间部分左侧方框里的内容,梯度矩阵shape为(2,4),乱序的稀疏矩阵shape为(4,9),相乘得到shape为(2,9)的直方图矩阵histo1。

[0085] 第二部分Guest将梯度矩阵gr碎片化秘密共享,Host将稀疏矩阵hr碎片化秘密共享,然后碎片gr和碎片hr进行mpc乘法,如图6中间部分右侧方框里的内容,梯度矩阵shape为(2,2),乱序的稀疏矩阵shape为(2,9),相乘得到shape为(2,9)的直方图矩阵<histo2>,目前<histo2>是碎片状态。

[0086] 第三阶段主要过程为Guest方将histo1发送给Host方,Host恢复<histo2>为histo2。然后构造一个和histo1同shape大小的噪声矩阵,最终将histo1,histo2和噪声矩阵按位相加得到Host方的直方图r_histo,r_histo还是乱序的,用第一阶段的乱序索引恢复r_histo得到最终的Host方直方图。

[0087] 对比原始处理方式,本文方案在第二阶段设分割比例为95%和5%,这样在大数据样本训练的情况下传输数据量下降20多倍,使得数据传输瓶颈得到缓解,整体运行效率会有数倍提升。本文使用多态计算方式可将95%稀疏矩阵进行高效的矩阵乘法,拿40万样本数据,每条数据有600个特征,每个特征分50个桶为例,原始方案乘法过程需要执行240亿次乘法运算和239.9994亿次加法运算,而本方案只要执行4.8亿次乘法和4.799988亿次加法计算,极大提升了计算性能。

[0088] 与上面介绍的多方安全计算对应地,本实施例还提供了一种多方安全计算系统。下面将分别进行介绍。具体地,如图9所示,本实施例还提供一种多方安全计算系统,该系统包括:

获取模块1,用于分别获取发起方对应的第一矩阵与参与方对应的第二矩阵。

[0089] 在一种可选地实施方式,第一矩阵可以为梯度矩阵,其可以包括一阶梯度和二阶

梯度,第二矩阵可以为稀疏矩阵,本实施例对此不作限制。

[0090] 还需要说明的是,第一矩阵的列数与第二矩阵的行数可以对应。

[0091] 划分模块2,用于将第一矩阵划分为第一矩阵单元和第二矩阵单元,将第二矩阵划分为第三矩阵单元和第四矩阵单元。

[0092] 可选地,划分模块2的处理过程可以作为第一阶段,其主要过程为将发起方(GUEST)的第一矩阵(梯度矩阵)分割成两部分,即划分为第一矩阵单元gl和第二矩阵单元gr,将参与方(HOST)的第二矩阵(稀疏矩阵)也同样分割成两部分,即划分为第三矩阵单元hl和第四矩阵单元hr。

[0093] 碎片化模块3,用于对第二矩阵单元进行碎片化处理得到第一碎片矩阵,对第四矩阵单元进行碎片化处理得到第二碎片矩阵。

[0094] 计算模块4,用于通过发起方获取第三矩阵单元并与第一矩阵单元进行计算得到第一计算结果,通过参与方获取第一碎片矩阵并与第二碎片矩阵进行计算得到第二计算结果。

[0095] 可选地,碎片化模块3和计算模块4的处理过程作为第二阶段,将发起方的第二矩阵单元gr进行碎片化后得到第一碎片矩阵<gr>后秘密共享,将参与方的第四矩阵单元hr进行碎片化得到第二碎片矩阵<hr>后秘密共享。

[0096] 将参与方的第三矩阵单元hl压缩后传递给发起方,发起方在本地进行第一矩阵单元gl和第三矩阵单元hl计算得到第一计算结果,即直方图矩阵histo1。

[0097] 参与方获取第一碎片矩阵<gr>并与第二碎片矩阵<hr>进行计算得到第二计算结果,即直方图矩阵<histo2>,目前<histo2>是碎片状态。

[0098] 聚合模块5,用于基于第一计算结果和第二计算结果得到聚合矩阵。

[0099] 可选地,聚合模块5的处理过程可以作为第三阶段,发起方将第一计算结果histo1发送给参与方,参与方将第二计算结果<histo2>恢复为histo2后得到聚合矩阵,即最终的Host方直方图,并同步至发起方。

[0100] 从而,本实施例将原始的发起方的第一矩阵和参与方秘密共享第二矩阵乘法得到HOST直方图拆分为三个阶段,通过这三个阶段的优化调整可以降低发起方和参与方的数据传输量,提高了计算效率。

[0101] 需要说明的是,发起方的梯度矩阵的shape为(2,m),参与方的稀疏矩阵shape为(m,f*k),其中m为样本量,f为特征数,k为每个特征分桶数,本实施例以m为6为例。

[0102] 作为本实施例的一种可选地实施方式,划分模块2,还用于基于预设划分比例将第一矩阵进行划分得到第一矩阵单元和第二矩阵单元。

[0103] 可选地,如图3所示,基于预设划分比例将发起方的第一矩阵(梯度矩阵)分割成两部分,然后发起方对梯度矩阵在样本维度进行切分,即将Guest梯度矩阵的m列切分成两部分,得到第一矩阵单元gl和第二矩阵单元gr。

[0104] 需要说明的是,对比原始处理方式,本实施例在第二阶段设分割比例为95%和5%,这样在大数据样本训练的情况下传输数据量下降20多倍,使得数据传输瓶颈得到缓解,整体运行效率会有数倍提升。本实施例使用多态计算方式可将95%稀疏矩阵进行高效的矩阵乘法,还是以40万训练样本集数据为例,其中每条数据有600个特征,每个特征分50个桶,现有方案乘法过程需要执行240亿次乘法运算和239.9994亿次加法运算,而本实施例只要执

行4.8亿次乘法和4.799988亿次加法计算，极大提升了计算性能。

[0105] 作为本实施例的另一种可选地实施方式，划分模块2，还用于将第二矩阵按列进行打乱处理，得到打乱后的第二矩阵，并记录打乱处理对应的乱序索引；

基于预设划分比例将打乱后的第二矩阵进行划分，得到第三矩阵单元和第四矩阵单元。可选地，如图4所示，参与方首先会将第二矩阵(稀疏矩阵)按列打乱，例如，由原来列序号[0,1,2,3,4,5,6,7,8]打乱为[8,1,4,3,2,6,5,7,0]，并记录打乱处理对应的乱序索引，然后参与方分对特征分桶稀疏矩阵在样本维度进行切分，即将Host特征分桶稀疏矩阵m行切分为两部分，得到第三矩阵单元hl和第四矩阵单元hr。

[0106] 作为本实施例的一种可选地实施方式，计算模块4，还用于将第三矩阵单元进行压缩处理后发送至发起方；

发起方将压缩后的第三矩阵单元与第一矩阵单元进行矩阵乘计算，得到第一计算结果。

[0107] 将Host方的乱序的第三矩阵单元hl压缩后传递给Guest方，Guest方在本地进行第一矩阵单元gl和第三矩阵单元hl乘法，其中，第一矩阵单元的梯度矩阵shape为(2,4)，乱序的第三矩阵单元的稀疏矩阵shape为(4,9)，相乘得到shape为(2,9)的直方图矩阵histo1，即第一计算结果。

[0108] 作为本实施例的另一种可选地实施方式，计算模块4，还用于参与方将第一碎片矩阵与第二碎片矩阵进行矩阵乘计算，得到第三碎片矩阵；

将第三碎片矩阵进行恢复处理，得到第二计算结果。

[0109] 发起方将第一碎片矩阵<gr>秘密共享给参与方，参与方将第二碎片矩阵<hr>秘密共享给发起方，然后第一碎片矩阵<gr>和第二碎片矩阵<hr>进行mpc乘法，梯度矩阵shape为(2,2)，乱序的稀疏矩阵shape为(2,9)，相乘得到shape为(2,9)的直方图矩阵<histo2>，即第二计算结果，目前<histo2>是碎片状态。

[0110] 通过上述过程，可以降低训练过程中通讯量，联邦学习xgboost主要瓶颈在于Host方直方图的计算，之所以为瓶颈是因为保证Guest方梯度矩阵和Host方特征数据矩阵不被泄露的情况下进行矩阵乘法，原始做法是将Host方特征数据矩阵分桶稀疏化然后直接与Guest方进行mpc乘法，但这种方式会有巨大的数据传输开销，不可在大数据样本进行执行。而本方案使用两种矩阵方式进行组合执行矩阵乘法，在保证数据不泄露的前提下大大下降数据传输量。

[0111] 作为本实施例的一种可选地实施方式，聚合模块5，还用于对第一计算结果和第二计算结果进行聚合处理得到聚合结果；

聚合模块5，还用于对第一计算结果、第二计算结果和预设噪声矩阵按位进行聚合处理得到聚合结果

利用乱序索引对聚合结果进行重排序处理，得到聚合矩阵。

[0112] 发起方将第一计算结果histo1发送给参与方，参与方将第二计算结果<histo2>为恢复histo2，然后构造一个和第一计算结果histo1同shape大小的噪声矩阵，最终将histo1，histo2和噪声矩阵按位相加得到参与方的直方图r_histo，即聚合结果，由于r_histo还是乱序的，用第一阶段的乱序索引恢复r_histo得到最终的Host方直方图，即聚合矩阵。

[0113] 由此,通过多态混合计算技术在稀疏矩阵上的运算加速,在进行稀疏矩阵计算时,HOST数据拥有方可以通过按列打乱然后压缩再发给GUEST计算方, GUEST计算方在压缩数据上进行加速计算后回传给数据拥有方,HOST数据拥有方恢复顺序后通过噪声辅助来掩盖原始数据,大大提高了多方计算的安全性,使得能够利用稀疏矩阵的特性,将按列打乱后的稀疏矩阵进行压缩,乘法运算时免去了不必要的乘0计算,整个过程计算量成倍下降,避免了数据安全性导致计算开销较大的问题。

[0114] 正如前述所说,当前的联邦学习XGBoost算法训练的通信量比较大,导致训练耗时过长,不能满足业界需求。因此,本实施例还提供一种联邦学习模型的训练方法,如图10所示,该训练方法包括:

S1、分别获取发起方和参与方的求交特征数据集。

[0115] S2、利用求交特征数据集构建XGBoost树模型。

[0116] S3、通过如上的多方安全计算方法得到的聚合矩阵以计算XGBoost树模型的最优分割点。

[0117] S4、利用最优分割点更新XGBoost树模型。

[0118] S5、将待预测的数据输入至更新后的XGBoost树模型进行预测,得到预测结果。

[0119] 在一个例子中,如图11a、图11b、图11c和图11d所示(图11a作为第一部分,图11b作为第二部分,图11c作为第三部分,图11d作为第四部分共同构成一个完整流程图),Guest发起方根据uid获取Guest方求交特征数据集,接收Host参与方的特征基本信息,生成随机种子并同步给Host方,开始初始化预测值p为0,并判断构建的树是否达到指定数量,若是则随机采样训练样本和训练特征,以计算计算一阶导数和计算二阶导数,随后判断是否达到树构建停止条件,若是则初始化Guest的特征直方图histo,并接收Host方发送过来的R_Bin_a',使用R_Bin_a'、g和h在本地计算出乱序直方图h_histo_a',将h_histo_a'发送给Host方,接收Host发送过来的<r_bin_b1>碎片,根据随机种子获得R_Bin_b对应的样本,将对应样本的g和h拼接的矩阵gh,分片后得到(<gh1><gh2>),发送<gh2>给Host,将碎片态<gh>与碎片态<r_bin_b>相乘生成host直方图<h_histo_b1'>,发送<h_histo_b1'>分片给host方,计算Guest方本地直方图g_histo,同时还同步计算Guest方特征数据的分桶边界数值和计算Guest方本地直方图g_histo,进而获得Guest和Host所有histo内容,根据计算待分裂节点的最优分割点,给达到停止分裂条件的节点赋值,发送给Host节点分裂信息,更新树结构,发送给Host下一level的节点信息,最后利用用新树预测原始数据,更新p值。

[0120] 与上述Guest发起方进行步骤相对应的,Host参与方的步骤如下,Host参与方根据uid获取host方求交特征数据集,发送给Guest特征基本信息,接收Guest的随机种子,并判断构建的树是否达到指定数量,若是则随机采样训练样本和训练特征,随后判断是否达到树构建停止条件,若是则初始化Host的特征直方图histo,构建特征分桶稀疏化矩阵Bin,按列打乱Bin得到R_Bin和恢复索引Bin_index,根据比例将R_Bin按行随机切分成R_Bin_a和R_Bin_b,将R_Bin_a压缩成索引形式 R_Bin_a' [(V1,IDX1),(V2,IDX2)...],发送R_Bin_a'给guest方,接收Guest方发送过来的h_histo_a',将R_Bin_b进行分片(<r_bin_b1><r_bin_b2>),发送分片<r_bin_b1>给Guest方发送分片<r_bin_b1>给Guest方,发送分片<r_bin_b1>给Guest方,将碎片态<gh>与碎片态<r_bin_b>相乘生成host直方图<h_histo_b2'>,本地恢复h_histo_b',然后将h_histo_a',h_histo_b'和混淆矩阵按位相加得到h_

histo',利用恢复索引Bin_index将h_histo'恢复成为h_histo,发送本地h_histo给Guest,接收Guest节点分裂信息,以更新树结构,接收Guest下一level的节点信息,最后用新树预测原始数据。

[0121] 在现有的XGBoost算法训练过程中最大的瓶颈在计算Host直方图,Guest方的梯度矩阵和Host的特征矩阵都不能相互泄露,而通过本实施例的多方安全计算方法所得到的聚合矩阵(Host直方图),不仅可以保证数据不相互泄露,又可以通过混合态多方安全计算机制在精度没有损失的前提下,大大幅度降低计算耗时。

[0122] 与上面介绍的联邦学习模型的训练方法对应地,本实施例还提供了一种联邦学习模型的训练系统。下面将分别进行介绍。具体地,如图12所示,本实施例还提供一种联邦学习模型的训练系统,该系统包括:

数据集获取模块101,用于分别获取发起方和参与方的求交特征数据集。

[0123] 模型构建模块102,用于利用求交特征数据集构建XGBoost树模型。

[0124] 分割点计算模块103,用于通过如上的多方安全计算系统得到的聚合矩阵以计算XGBoost树模型的最优分割点。

[0125] 模型更新模块104,用于利用最优分割点更新XGBoost树模型。

[0126] 模型预测模块105,用于将待预测的数据输入至更新后的XGBoost树模型进行预测,得到预测结果。

[0127] 在现有的XGBoost算法训练过程中最大的瓶颈在计算Host直方图,Guest方的梯度矩阵和Host的特征矩阵都不能相互泄露,而通过本实施例的多方安全计算方法所得到的聚合矩阵(Host直方图),不仅可以保证数据不相互泄露,又可以通过混合态多方安全计算机制在精度没有损失的前提下,大大幅度降低计算耗时。

[0128] 还需要说明的是,本实施例的多方安全计系统或联邦学习模型的训练系统,例如可以是:单独的芯片、芯片模组或电子设备,也可以是集成于电子设备内的芯片或者芯片模组。关于上述实施例中描述的各个装置、产品包含的各个模块/单元,其可以是软件模块/单元,也可以是硬件模块/单元,或者也可以部分是软件模块/单元,部分是硬件模块/单元。例如,对于应用于或集成于芯片的各个装置、产品,其包含的各个模块/单元可以都采用电路等硬件的方式实现,或者,至少部分模块/单元可以采用软件程序的方式实现,该软件程序运行于芯片内部集成的处理器,剩余的(如果有)部分模块/单元可以采用电路等硬件方式实现;对于应用于或集成于芯片模组的各个装置、产品,其包含的各个模块/单元可以都采用电路等硬件的方式实现,不同的模块/单元可以位于芯片模组的同一组件(例如芯片、电路模块等)或者不同组件中,或者,至少部分模块/单元可以采用软件程序的方式现,该软件程序运行于芯片模组内部集成的处理器,剩余的(如果有)部分模块/单元可以采用电路等硬件方式实现;对于应用于或集成于终端的各个装置、产品,其包含的各个模块/单元可以都采用电路等硬件的方式实现,不同的模块/单元可以位于终端内同一组件(例如,芯片、电路模块等)或者不同组件中,或者,至少部分模块/单元可以采用软件程序的方式实现,该软件程序运行于终端内部集成的处理器,剩余的(如果有)部分模块/单元可以采用电路等硬件方式实现。

[0129] 图13为本实施例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

例中的方法。图13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。

[0130] 如图13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。

[0131] 总线33包括数据总线、地址总线和控制总线。

[0132] 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。

[0133] 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。

[0134] 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，例如本发明如上所述的方法。

[0135] 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13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。

[0136] 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。

[0137] 本实施例还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如上述实施例的方法中的步骤。

[0138] 其中，可读存储介质可以采用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。

[0139] 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如上所述的方法中的步骤。

[0140] 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。

[0141] 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。

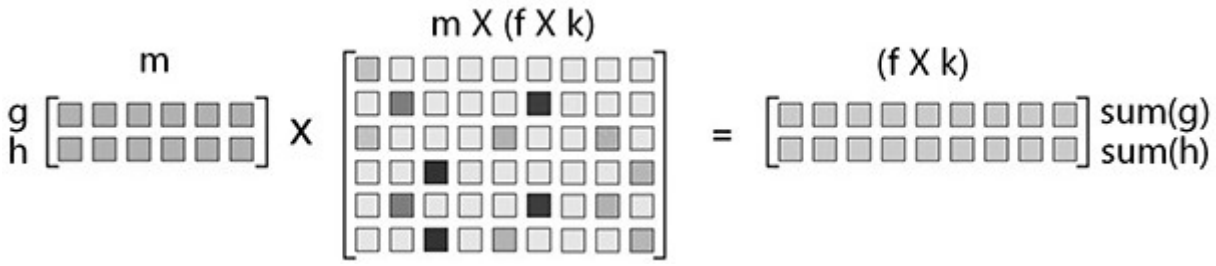

m X (f X k)
m
g
h
X
=
(f X k)
sum(g)
sum(h)

图1

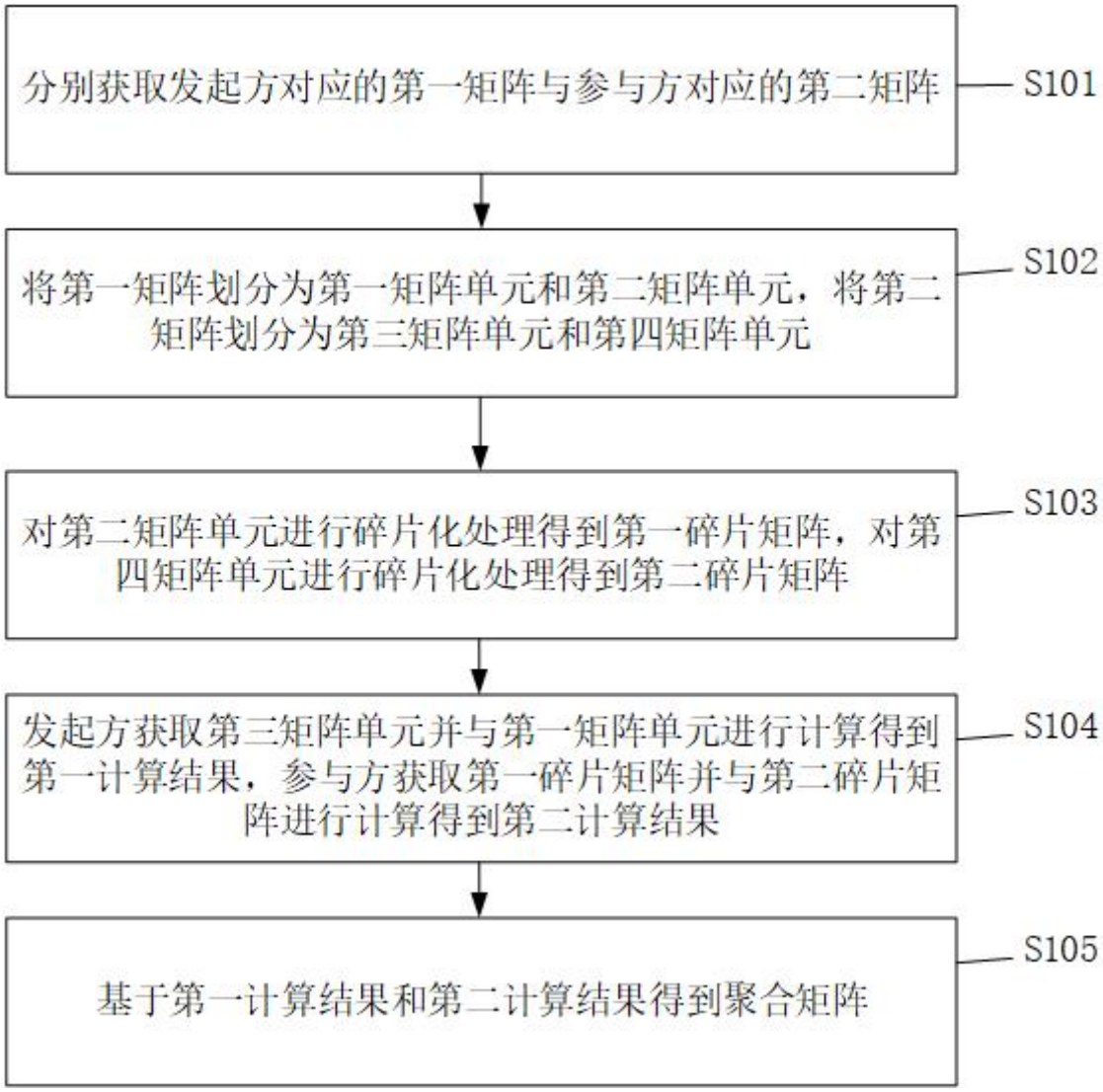

分别获取发起方对应的第一矩阵与参与方对应的第二矩阵
S101
将第一矩阵划分为第一矩阵单元和第二矩阵单元，将第二矩阵划分为第三矩阵单元和第四矩阵单元
S102
对第二矩阵单元进行碎片化处理得到第一碎片矩阵，对第四矩阵单元进行碎片化处理得到第二碎片矩阵
S103
发起方获取第三矩阵单元并与第一矩阵单元进行计算得到第一计算结果，参与方获取第一碎片矩阵并与第二碎片矩阵进行计算得到第二计算结果
S104
基于第一计算结果和第二计算结果得到聚合矩阵
S105

图2

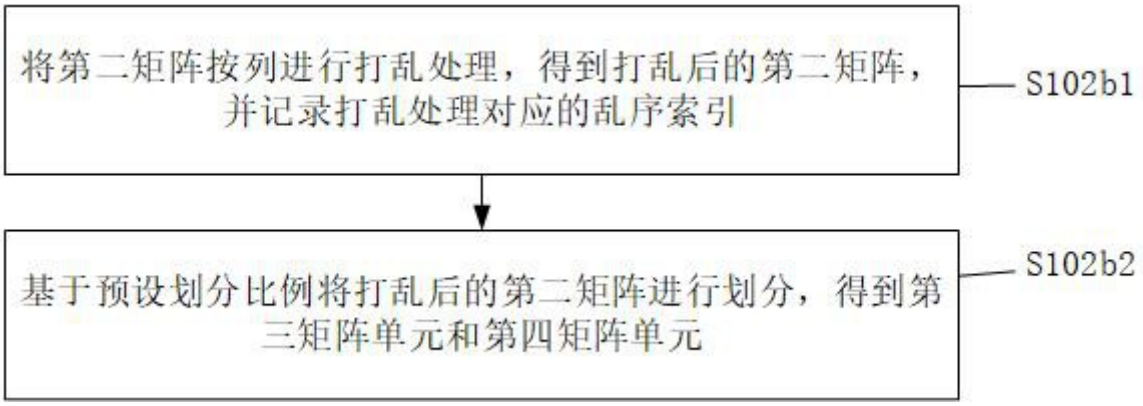

将第二矩阵按列进行打乱处理，得到打乱后的第二矩阵，并记录打乱处理对应的乱序索引
S102b1
基于预设划分比例将打乱后的第二矩阵进行划分，得到第三矩阵单元和第四矩阵单元
S102b2

图3

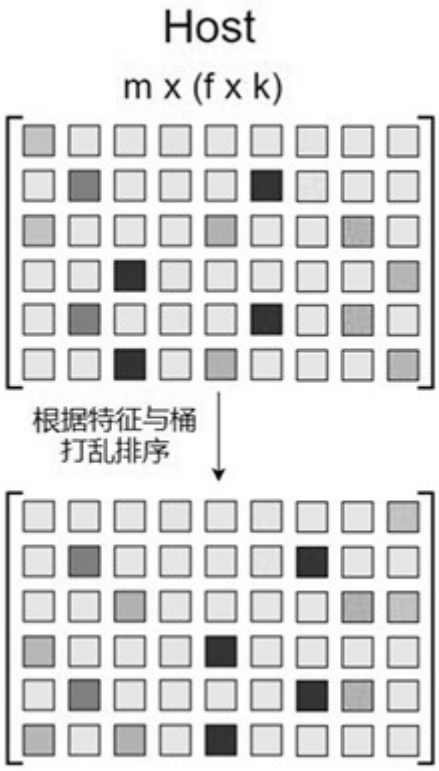

Host
m x (f x k)
根据特征与桶
打乱排序

图4

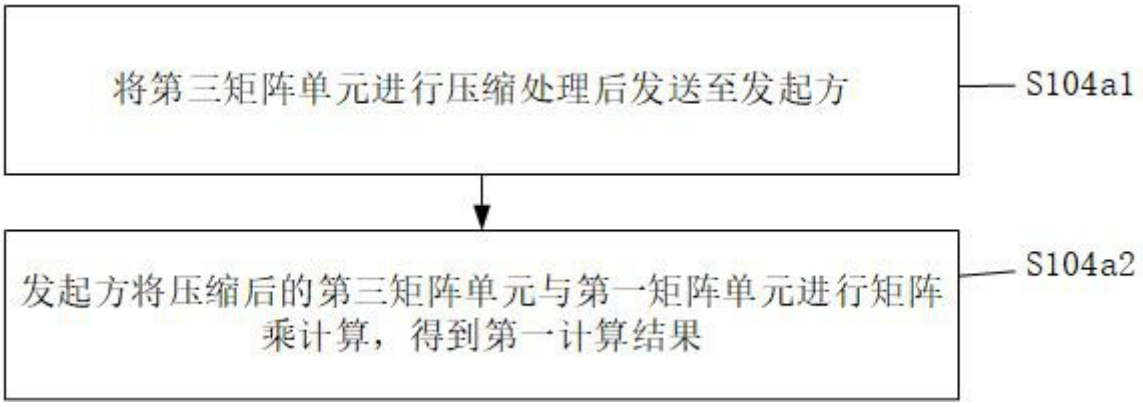

将第三矩阵单元进行压缩处理后发送至发起方
S104a1
发起方将压缩后的第三矩阵单元与第一矩阵单元进行矩阵乘计算，得到第一计算结果
S104a2

图5

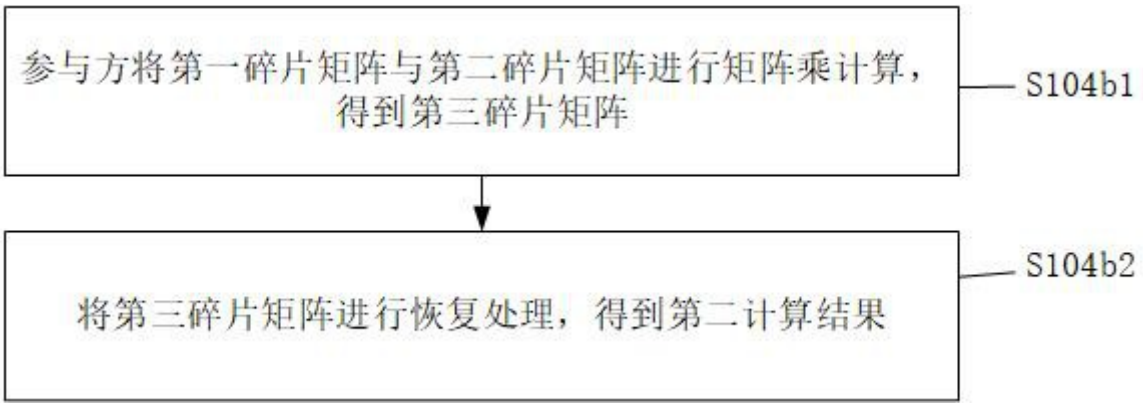

参与方将第一碎片矩阵与第二碎片矩阵进行矩阵乘计算，得到第三碎片矩阵
S104b1
将第三碎片矩阵进行恢复处理，得到第二计算结果
S104b2

图6

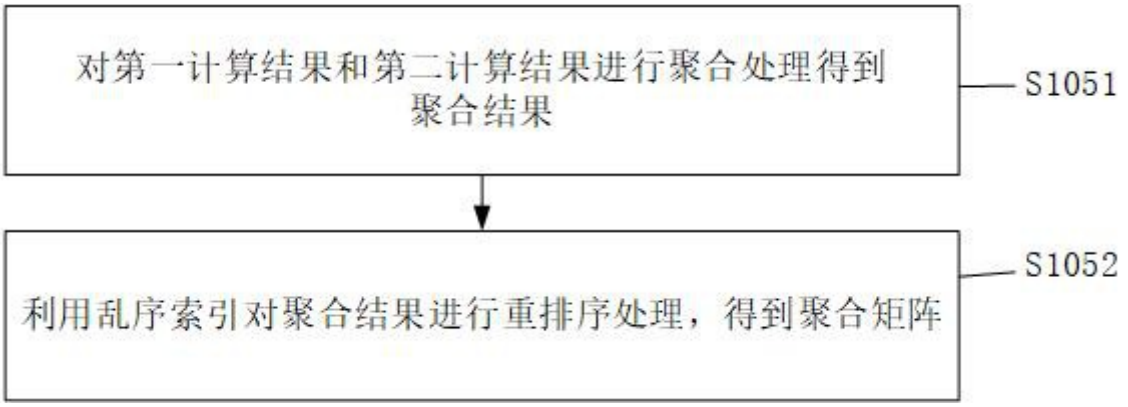

对第一计算结果和第二计算结果进行聚合处理得到聚合结果
S1051
利用乱序索引对聚合结果进行重排序处理，得到聚合矩阵
S1052

图7

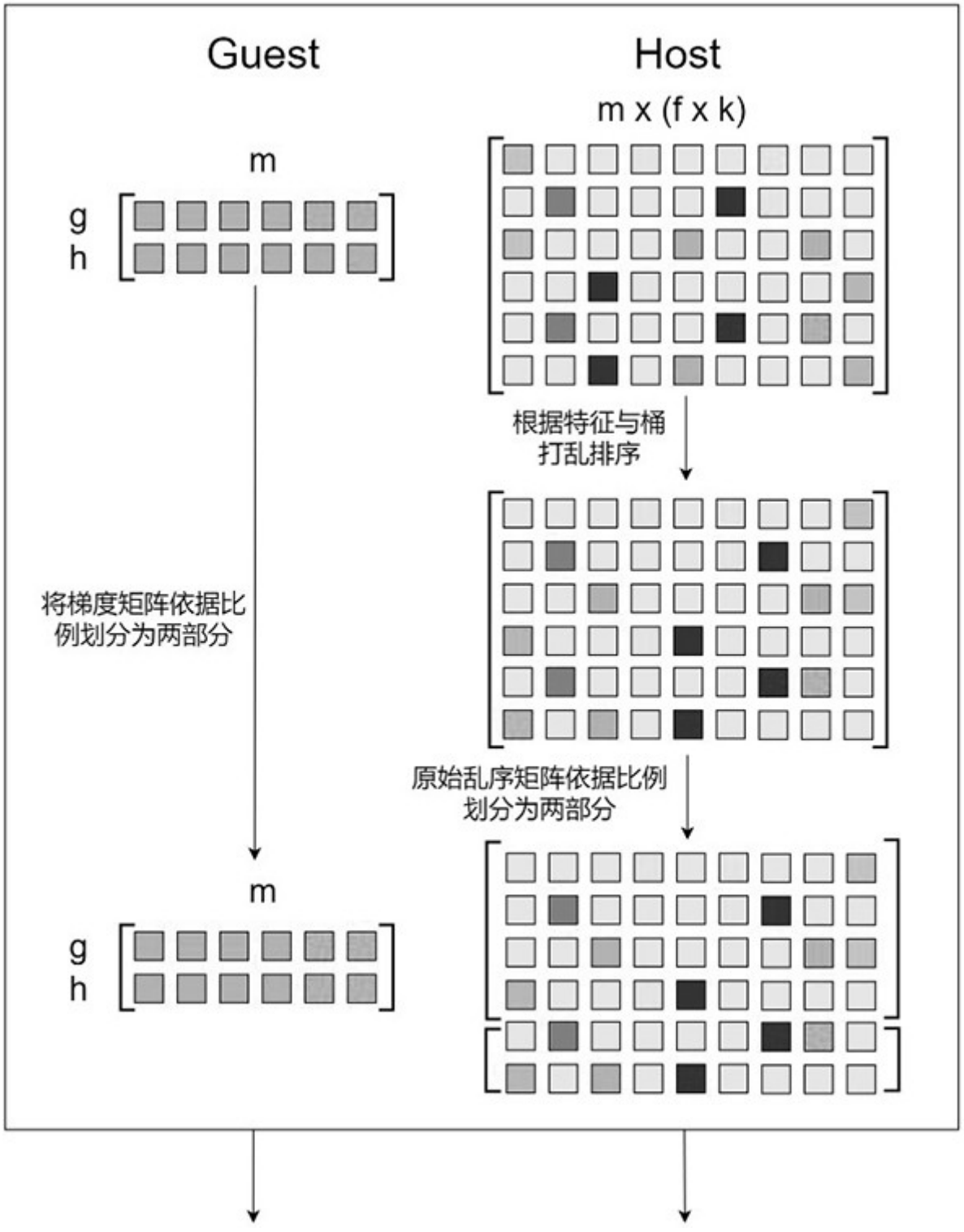

Guest
Host
m x (f x k)
m
g
h
根据特征与桶
打乱排序
将梯度矩阵依据比
例划分为两部分
原始乱序矩阵依据比例
划分为两部分
m
g
h

图8a

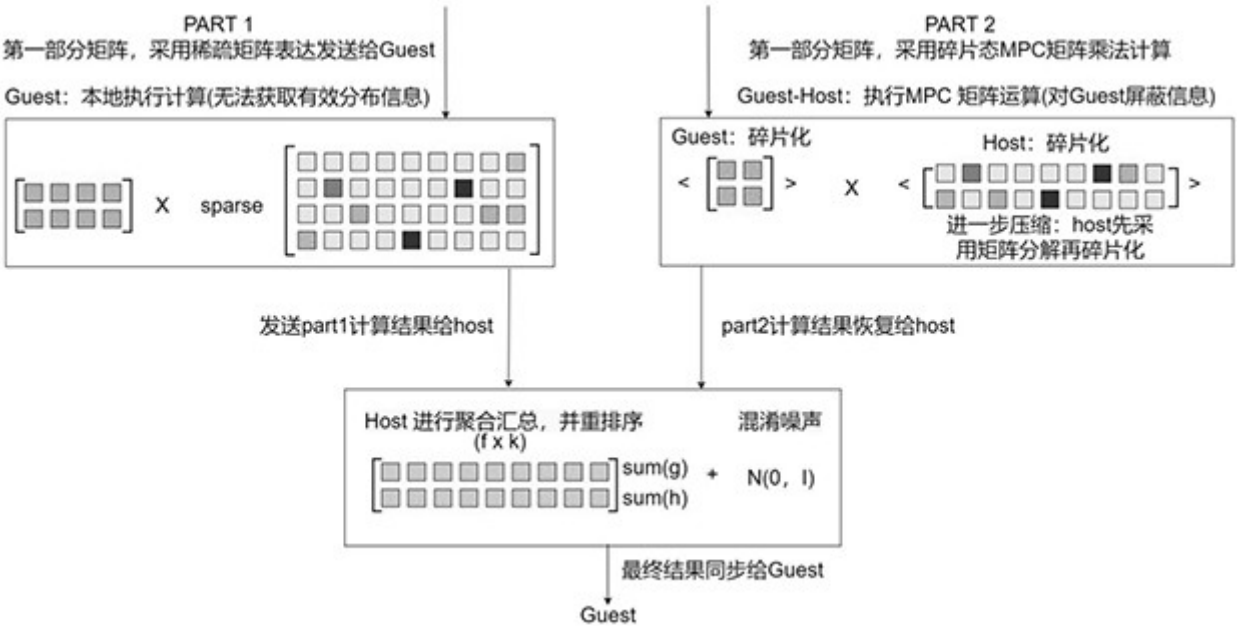

PART 1
第一部分矩阵，采用稀疏矩阵表达发送给Guest
Guest：本地执行计算(无法获取有效分布信息)
X
sparse
PART 2
第一部分矩阵，采用碎片态MPC矩阵乘法计算
Guest-Host：执行MPC 矩阵运算(对Guest屏蔽信息)
Guest：碎片化
Host：碎片化
X
进一步压缩：host先采
用矩阵分解再碎片化
发送part1计算结果给host
part2计算结果恢复给host
Host 进行聚合汇总，并重排序
(f x k)
混淆噪声
sum(g)
sum(h)
N(0，I)
最终结果同步给Guest
Guest

图8b

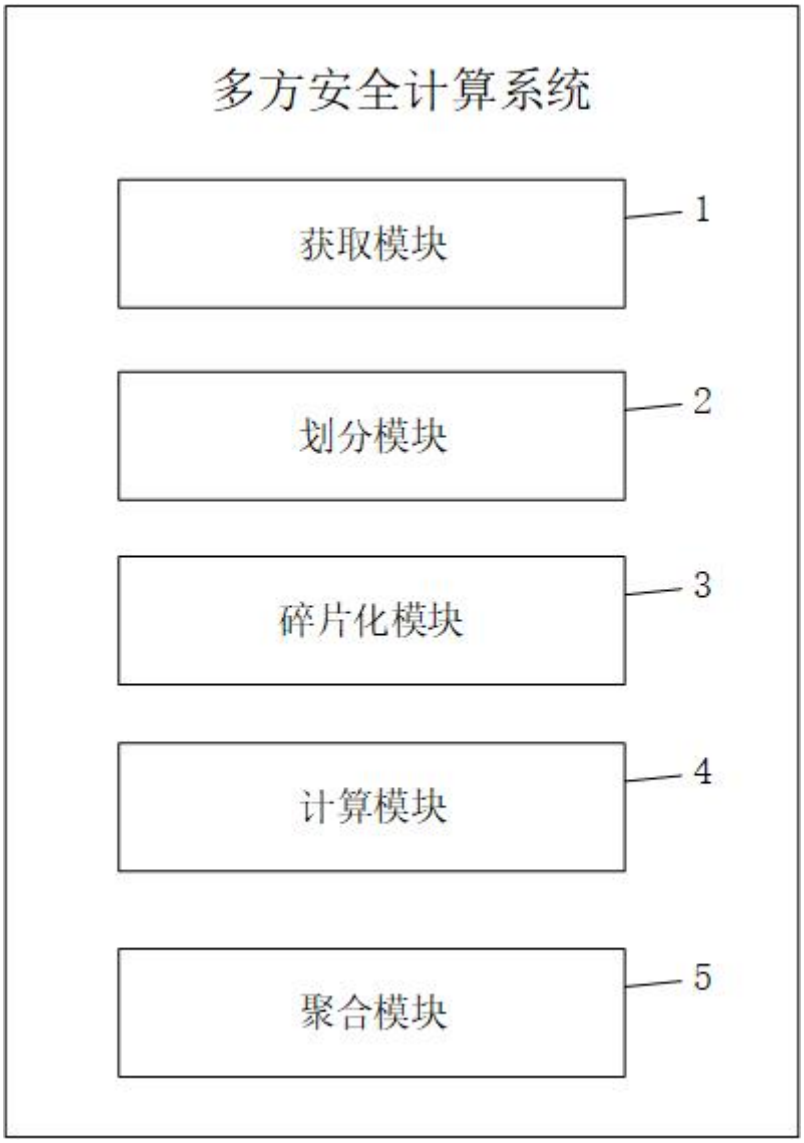


图9

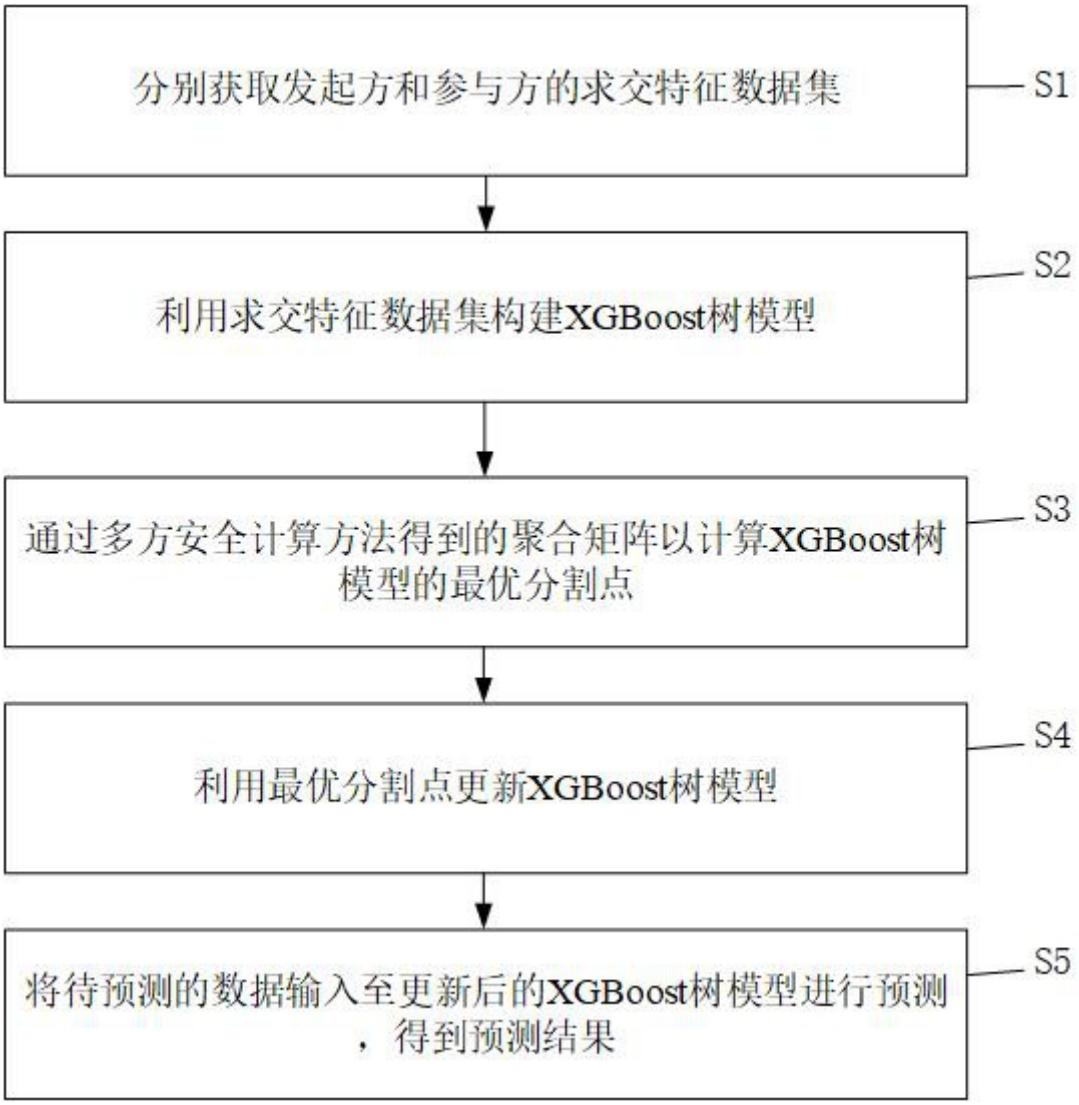

分别获取发起方和参与方的求交特征数据集
S1
利用求交特征数据集构建XGBoost树模型
S2
通过多方安全计算方法得到的聚合矩阵以计算XGBoost树模型的最优分割点
S3
利用最优分割点更新XGBoost树模型
S4
将待预测的数据输入至更新后的XGBoost树模型进行预测，得到预测结果
S5

图10

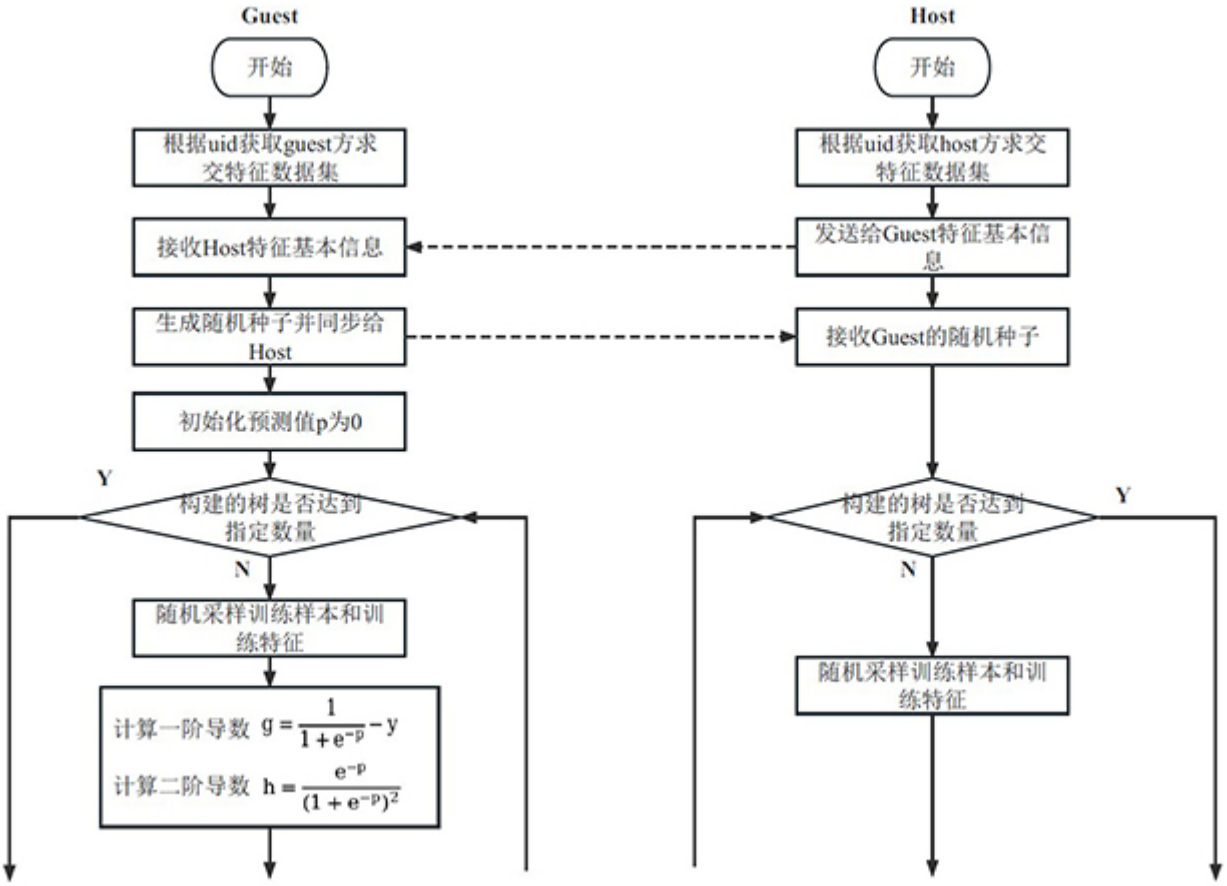

Guest
开始
根据uid获取guest方求交特征数据集
接收Host特征基本信息
生成随机种子并同步给Host
初始化预测值p为0
Y
构建的树是否达到指定数量
N
随机采样训练样本和训练特征
计算一阶导数 $g=\frac{1}{1+e^{-p}}-y$
计算二阶导数 $h=\frac{e^{-p}}{(1+e^{-p})^2}$
Host
开始
根据uid获取host方求交特征数据集
发送给Guest特征基本信息
接收Guest的随机种子
构建的树是否达到指定数量
Y
N
随机采样训练样本和训练特征

图11a

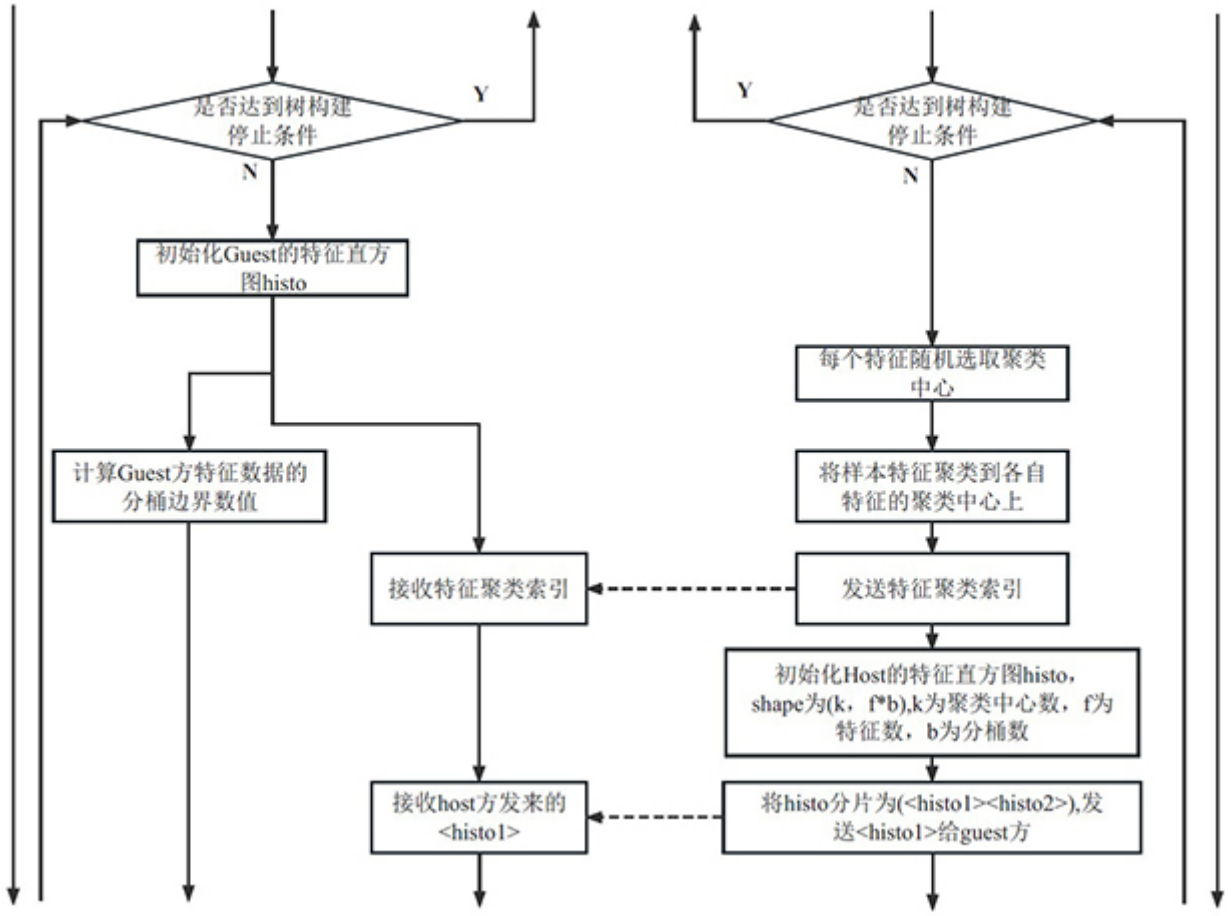

是否达到树构建停止条件
Y
N
初始化Guest的特征直方图histo
计算Guest方特征数据的分桶边界数值
接收特征聚类索引
接收host方发来的<histo1>
是否达到树构建停止条件
Y
N
每个特征随机选取聚类中心
将样本特征聚类到各自特征的聚类中心上
发送特征聚类索引
初始化Host的特征直方图histo，shape为(k，f*b),k为聚类中心数，f为特征数，b为分桶数
将histo分片为(<histo1><histo2>),发送<histo1>给guest方

图11b

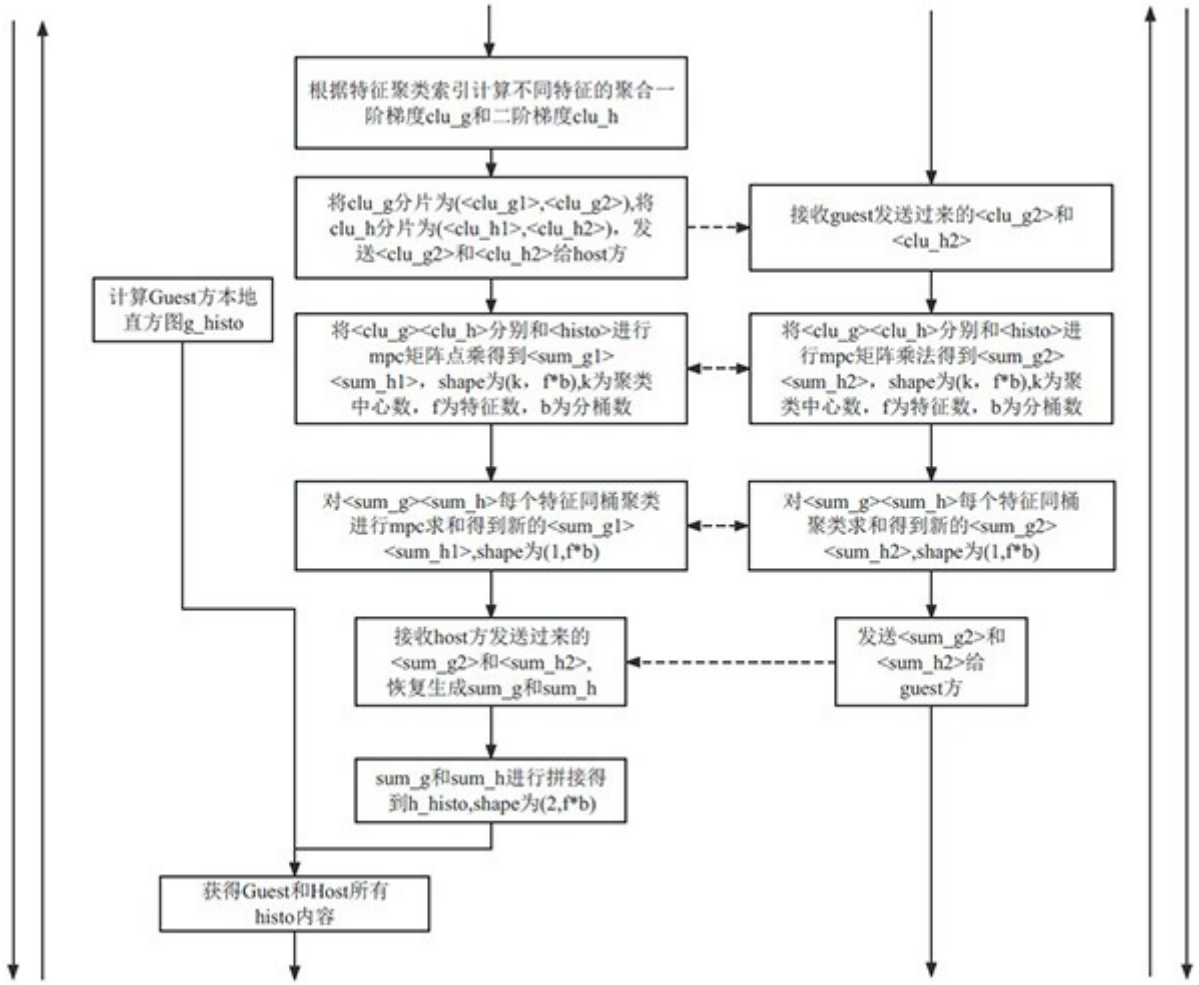

根据特征聚类索引计算不同特征的聚合一阶梯度clu_g和二阶梯度clu_h
将clu_g分片为(<clu_g1>,<clu_g2>),将clu_h分片为(<clu_h1>,<clu_h2>)，发送<clu_g2>和<clu_h2>给host方
接收guest发送过来的<clu_g2>和<clu_h2>
计算Guest方本地直方图g_histo
将<clu_g><clu_h>分别和<histo>进行mpc矩阵点乘得到<sum_g1><sum_h1>，shape为(k，f*b),k为聚类中心数，f为特征数，b为分桶数
将<clu_g><clu_h>分别和<histo>进行mpc矩阵乘法得到<sum_g2><sum_h2>，shape为(k，f*b),k为聚类中心数，f为特征数，b为分桶数
对<sum_g><sum_h>每个特征同桶聚类进行mpc求和得到新的<sum_g1><sum_h1>,shape为(1,f*b)
对<sum_g><sum_h>每个特征同桶聚类求和得到新的<sum_g2><sum_h2>,shape为(1,f*b)
接收host方发送过来的<sum_g2>和<sum_h2>,恢复生成sum_g和sum_h
发送<sum_g2>和<sum_h2>给guest方
sum_g和sum_h进行拼接得到h_histo,shape为(2,f*b)
获得Guest和Host所有histo内容

图11c

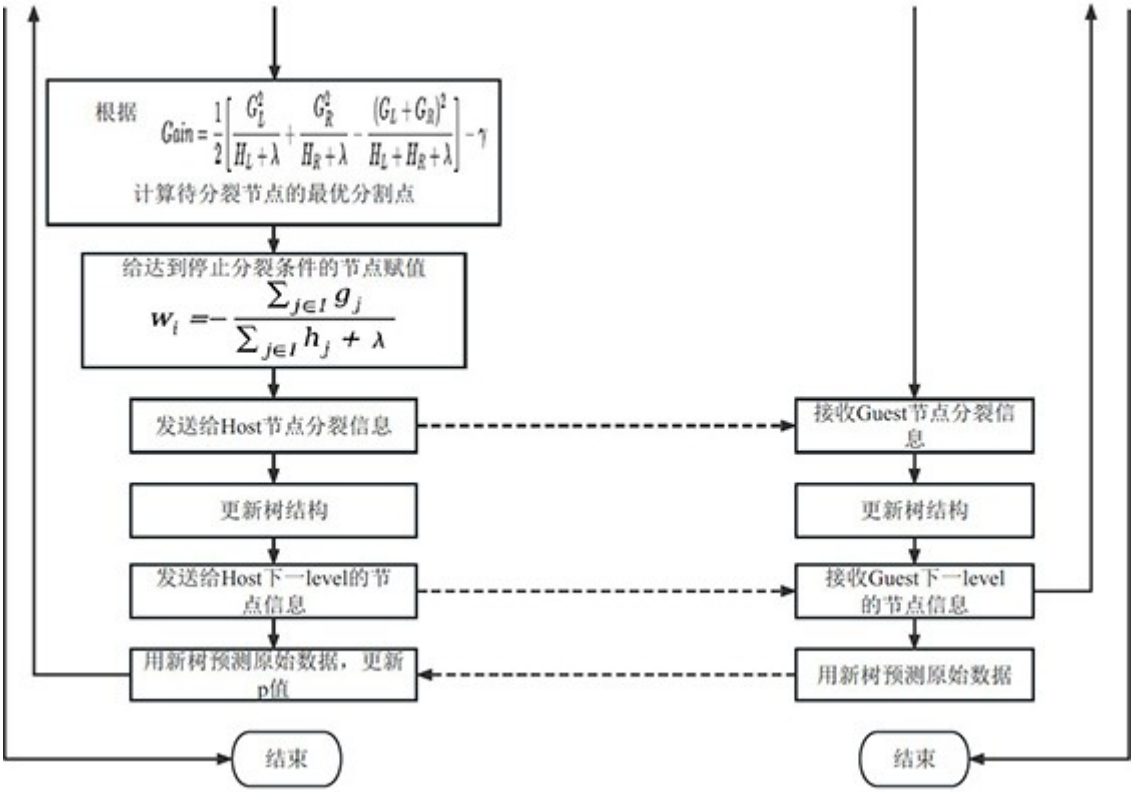

根据 $Gain=\frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right]-\gamma$
计算待分裂节点的最优分割点
给达到停止分裂条件的节点赋值
$w_i=-\frac{\sum_{j\in I} g_j}{\sum_{j\in I} h_j+\lambda}$
发送给Host节点分裂信息
更新树结构
发送给Host下一level的节点信息
用新树预测原始数据，更新p值
结束
接收Guest节点分裂信息
更新树结构
接收Guest下一level的节点信息
用新树预测原始数据
结束

图11d

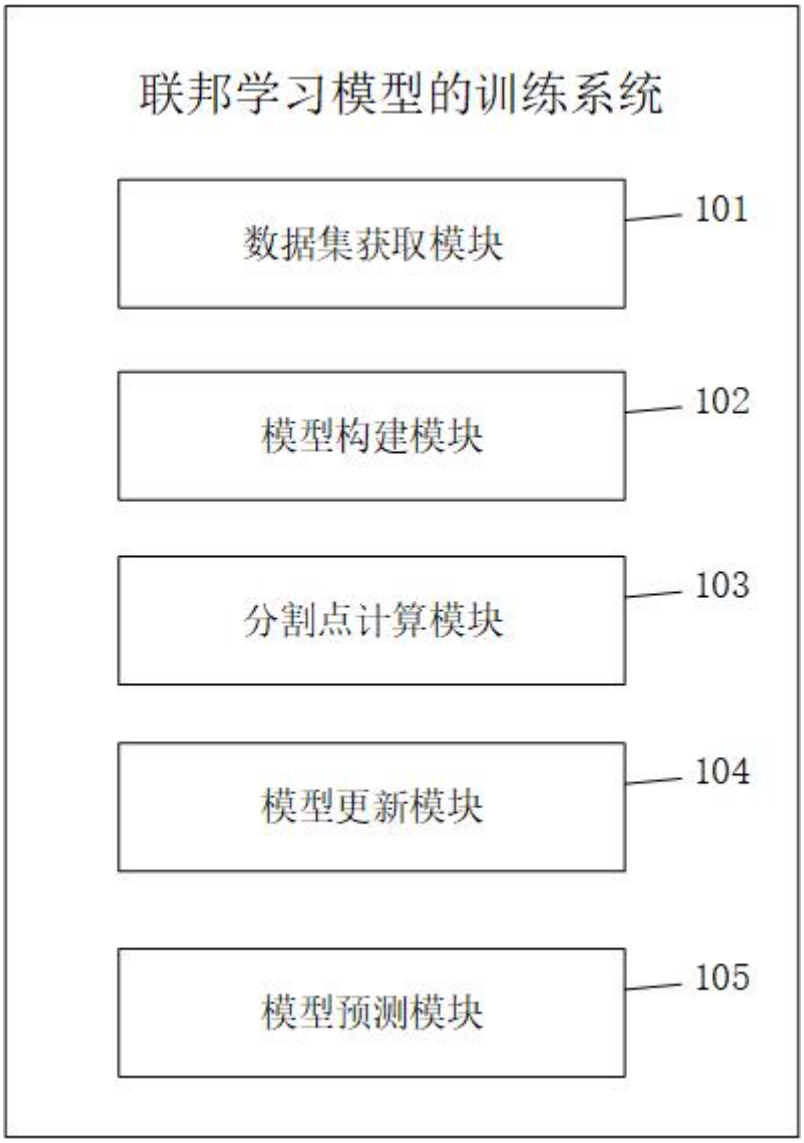

联邦学习模型的训练系统
数据集获取模块
101
模型构建模块
102
分割点计算模块
103
模型更新模块
104
模型预测模块
105

图12

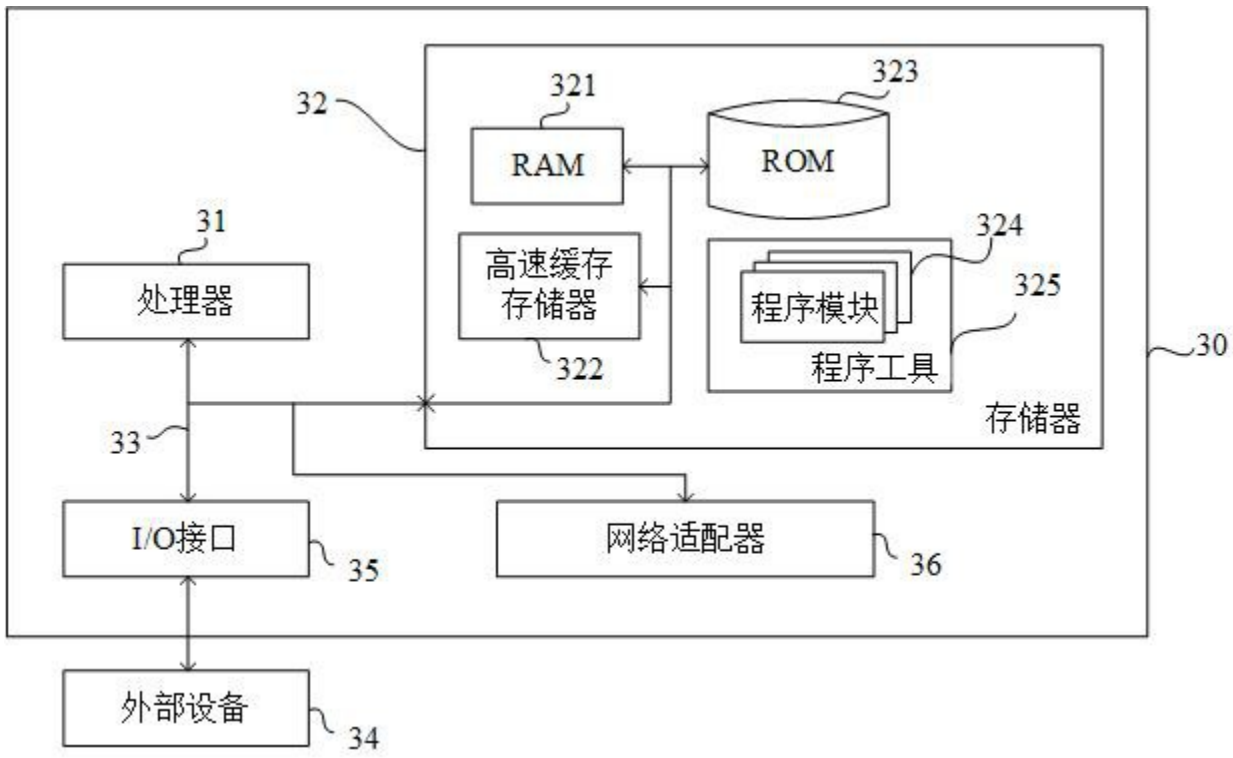

32
321
323
RAM
ROM
31
处理器
高速缓存
存储器
322
324
325
程序模块
程序工具
30
存储器
33
I/O接口
35
网络适配器
36
外部设备
34

图13