

说明书

联邦训练的方法、装置及计算机系统

5 技术领域

本说明书一个或多个实施例涉及计算机技术领域，尤其涉及一种联邦训练的方法、装置及计算机系统。

背景技术

10 随着人工智能和大数据时代的到来，机器学习（Machine Learning）技术各个领域得到了广泛的应用。然而，由于数据的安全性和隐私保护等问题，传统的集中式机器学习方法面临着诸多限制。于是，联邦学习（Federated Learning）应运而生。

联邦学习（或称联邦训练）是一种分布式机器学习技术，它允许多个参与方在保护数据隐私的前提下进行模型训练。每个参与方只在本地维护自己的数据集，并通过联邦学习
15 算法交换部分的模型更新参数，从而实现模型的联合训练。联邦学习因其保护数据隐私和保密性的特性，被广泛应用于医疗、金融、物联网等领域。

然而，由于联邦学习涉及多个参与方的协作，导致其模型优化问题更加复杂。由此，本说明书实施例提供一种在联邦学习框架下优化模型的新算法，可以更好地满足实际应用要求，例如提高联邦学习的训练效率，实现模型参数的快速收敛，提高训练出的模型的可用性等。
20

发明内容

本说明书一个或多个实施例描述了联邦训练方法和对应装置，以满足实际应用中对联邦学习性能和效果的更高要求。

25 根据第一方面，提供了一种联邦训练方法，涉及中间方和多个参与方，所述方法应用于其中任意一个参与方，所述方法涉及的任一轮迭代训练包括：利用本轮对应的一批次本地训练数据，计算本地模型当前的第一参数对应的第一梯度；根据所述第一梯度和当前的本地近似矩阵，计算指示参数更新方向的目标向量，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；利用所述目标向量和获取的步长将所述第一参数更新为第二参数，并计算所述第二参数对应的第二梯度；将所述第二参数和第二梯度发送给中间方；
30 从所述中间方接收目标参数和目标矩阵；所述目标参数和目标矩阵由中间方基于所述多个

参与方各自提供的第二参数和第二梯度，更新全局模型参数和全局近似矩阵后而确定；将本地模型参数和本地近似矩阵对应更新为所述目标参数和目标矩阵，用于下一轮迭代训练。

根据第二方面，提供了一种联邦训练方法，涉及中间方和多个参与方，所述方法应用于所述中间方，所述方法涉及的任一轮迭代训练包括：从任意一个参与方接收其本地模型当前的第二参数和对应的第二梯度，所述第二参数是利用目标向量和步长对第一参数进行更新而得到，所述目标向量是根据第一梯度和当前的本地近似矩阵而计算得到，所述第一梯度是所述第一参数在本轮对应的一批次本地训练数据上的梯度，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵。基于所述多个参与方各自提供的第二参数和第二梯度，确定本轮更新后的全局模型参数和全局近似矩阵。基于本轮更新后的全局模型参数和全局近似矩阵，将任意一个参与方对应的目标参数和目标矩阵提供给该参与方，使得该参与方利用所述目标参数和目标矩阵对应更新本地模型参数和本地近似矩阵，用于下一轮迭代训练。

根据第三方面，提供了一种联邦训练装置，涉及中间方和多个参与方，所述装置集成于其中任意一个参与方，所述装置通过以下模块执行任一轮迭代训练：第一梯度计算模块，配置为利用本轮对应的一批次本地训练数据，计算本地模型当前的第一参数对应的第一梯度；目标向量计算模块，配置为根据所述第一梯度和当前的本地近似矩阵，计算指示参数更新方向的目标向量，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；第一参数更新模块，配置为利用所述目标向量和获取的步长将所述第一参数更新为第二参数，并计算所述第二参数对应的第二梯度；中间数据传输模块，配置为将所述第二参数和第二梯度发送给中间方；目标数据接收模块，配置为从所述中间方接收目标参数和目标矩阵；所述目标参数和目标矩阵由中间方基于所述多个参与方各自提供的第二参数和第二梯度，更新全局模型参数和全局近似矩阵后而确定；本地更新模块，配置为将本地模型参数和本地近似矩阵对应更新为所述目标参数和目标矩阵，用于下一轮迭代训练。

根据第四方面，提供一种联邦训练装置，涉及中间方和多个参与方，所述装置集成于所述中间方，所述装置通过以下模块执行任一轮迭代训练：中间数据接收模块，配置为从任意一个参与方接收其本地模型当前的第二参数和对应的第二梯度，所述第二参数是利用目标向量和步长对第一参数进行更新而得到，所述目标向量是根据第一梯度和当前的本地近似矩阵而计算得到，所述第一梯度是所述第一参数在本轮对应的一批次本地训练数据上的梯度，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；全局数据更新模块，配置为基于所述多个参与方各自提供的第二参数和第二梯度，确定本轮更新后的

全局模型参数和全局近似矩阵；目标数据发送模块，配置为基于本轮更新后的全局模型参数和全局近似矩阵，将任意一个参与方对应的目标参数和目标矩阵提供给该参与方，使得该参与方利用所述目标参数和目标矩阵对应更新本地模型参数和本地近似矩阵，用于下一轮迭代训练。

5 根据第五方面，提供了一种计算机系统，包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器集成于多个参与方中的任意一个参与方，所述多个参与方与中间方协同进行联邦训练。所述处理器执行所述程序时实现第一方面提供的方法中的步骤。

10 根据第六方面，提供了一种计算机系统，包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器集成于中间方，所述中间方与多个参与方协同进行联邦训练。所述处理器执行所述程序时实现第一方面提供的方法中的步骤。

根据第七方面，提供了一种计算设备，包括存储器和处理器，其中所述存储器中存储有可执行代码，所述处理器执行所述可执行代码时，实现第一方面或第二方面提供的方法。

15 采用本说明书实施例提供的方法和装置，在联邦学习中，能够更好地保护多个参与方各自的数据隐私，减少数据传输量，并且提高模型的优化效率。

附图说明

20 为了更清楚地说明本发明实施例的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其它的附图。

图 1 为本说明书实施例披露的联邦训练方案的实施架构示意图；

25 图 2 为本说明书实施例披露的联邦训练方法涉及的中间方和多个参与方之间的通信交互示意图；

图 3 为本说明书实施例披露的联邦训练装置集成于任一参与方时的功能结构示意图；

图 4 为本说明书实施例披露的联邦训练装置集成于中间方时的功能结构示意图；

图 5 为本说明书实施例提供的计算机系统的结构示意图。

30 具体实施方式

下面结合附图，对本说明书提供的方案进行描述。

承前所述，由于联邦学习涉及多个参与方的协作，导致其模型优化问题更加复杂。传统的优化算法在联邦学习中难以满足实际应用中的更高需求，例如，传统的优化算法对计算和存储资源的要求较高，收敛速度慢等。

5 申请人还观察到，BFGS (Broyden-Fletcher-Goldfarb-Shanno) 优化算法，以及其改进算法，例如，L-BFGS (Limited-memory BFGS) 优化算法、或 L-BFGS-B (L-BFGS with Bound Constraints) 优化算法，具有较快的收敛速度和较好的数值稳定性，能够处理大规模问题。为简洁描述，以下将 BFGS 算法及其改进算法统称为 BFGS 系算法。

然而，传统的 BFGS 系算法仅适用于集中式机器学习，并不适用于联邦学习的场景。基于以上观察和分析，申请人提出对 BFGS 系算法进行改造，使其适用于联邦学习场景，或者说，对 BFGS 系算法与联邦学习框架进行有机结合，由此，提出了一种新的联邦学习方案。

图 1 为本说明书实施例披露的联邦训练方案的实施架构示意图。如图 1 所示，方案的实施涉及中间方和多个参与方，图 1 中以 P_i 方指代其中的每一个参与方。在方案涉及的任意的第 $k+1$ 轮次迭代训练中， P_i 方会利用本地的一批次训练数据 d 和针对本地模型参数 x_k 的海森逆矩阵的近似矩阵 B_k ，更新本地模型参数 x_k 和其数据 d 上的梯度 ∇_k ，并将更新得到的参数 x_k' 和梯度 ∇_k' 提供给中间方。于是，中间方可以基于多个参与方各自提供的参数 x_k' 和梯度 ∇_k' ，确定本轮更新后的全局模型参数 $x_{k+1}^{(G)}$ 和全局近似矩阵 $B_{k+1}^{(G)}$ ，进而根据 $x_{k+1}^{(G)}$ 和 $B_{k+1}^{(G)}$ 确定向 P_i 方发送的目标参数 $x^{(t)}$ 和目标矩阵 $B^{(t)}$ 。基于此， P_i 方利用目标参数 $x^{(t)}$ 将本地模型的参数 x_k' 更新为 x_{k+1} ，以及，利用目标矩阵 $B^{(t)}$ 将近似矩阵 B_k 更新为近似矩阵 B_{k+1} ，从而， P_i 方可以利用本轮更新后的参数 x_{k+1} 和近似矩阵 B_{k+1} 参与下一轮次（第 $k+2$ 轮次）的迭代训练。如此，可以在保证训练效果的同时减少迭代轮次和数据传输量，高效实现联邦学习框架下的模型优化。

在进一步介绍上述联邦训练方案的具体实施步骤之前，先对中间方、多个参与方的关联关系和实体形式，以及训练数据和全局模型参数在多个参与方中的分布进行介绍。

25 1) 关于中间方和多个参与方之间的关联关系。一般地，多个参与方之间互相独立。中间方可以是多个参与方之一，也可以是多个参与方以外的其他方。

2) 关于中间方和多个参与方的实体形式。可以理解，方案涉及的各方均可以实现为任何具有计算、处理能力的装置、平台、服务器或设备集群等。

3) 关于训练数据在多个参与方中的分布。

30 在一种典型情况中，训练数据垂直分布于多个参与方，也即，多个参与方拥有相同的

样本空间和不同的特征空间，换言之，多个参与方持有相同样本的不同特征部分。

在一个实施例中，多个参与方希望共同训练的共享模型（或称全局模型）是风险评估模型，用于预测用户未来的信用风险。示例性的，多个参与方中的数据分布为：

参与方甲（如某银行）：拥有用户的贷款记录、还款历史等信贷信息。

5 参与方乙（如某电商平台）：拥有相同用户的在线购物行为、消费习惯等电商数据。

参与方丙（如某信用评分机构）：拥有相同用户的信用评分和其他信用相关的信息。

在另一个实施例中，多个参与方希望共同训练推荐模型，用于为用户推荐其可能感兴趣的对象，如商品，文章或音视频等数字内容，公众号等。示例性的，多个参与方中的数据分布为：

10 参与方甲（如某在线视频平台）：拥有用户的观看历史、偏好的视频类型、观看时间等数据。

参与方乙（如某电商平台）：拥有相同用户的购物历史、偏好的商品类型、搜索历史等数据。

15 以上介绍多个参与方中针对相同样本的不同特征集。需说明的是，文中提及的所有用户数据，均是在用户确认授权的情况下进行采集和使用。

另外，多个参与方均拥有样本标签。一方面，对于样本标签的内容，例如，在上述风险评估场景下，样本标签可以是信用风险标签，又例如，在上述推荐场景下，样本标签可以是行为类别标签，指示某个用户是否对某个对象（如某个商品）做出预定行为（如购买）。另一方面，对于各参与方中样本标签的来源，比如，可以是各参与方可以自行打标而获得。

20 又例如，多个参与方互相之间可以针对相同样本进行标签的共享和统一。

再者，对于以上提及的相同样本，一般情况下，多个参与方之间的数据处于安全隔离的状态，任一参与方都不知道其他参与方拥有哪些样本。于是，多个参与方可以各自基于本地的样本标识集，共同执行隐私集合求交（Private Set Intersection，简称 PSI）协议，执行的结果是多个参与方可以获知多个样本标识集之间的交集，但不能获知除此以外的其他任何信息。

25 以上介绍训练数据垂直分布于多个参与方中的典型情况。另外，用于应对训练数据垂直分布的联邦学习，也称纵向联邦学习。

在另一种典型情况中，训练数据水平分布于多个参与方，也即，多个参与方拥有不同的样本空间和相同的特征空间，换言之，多个参与方持有不同样本的全部特征。

30 在一个实施例中，多个参与方希望共同训练针对用户的信用评分模型，用于预测用户

未来的信用风险。示例性的，多个参与方中的数据分布为：

参与方甲（如银行 1）：位于地区 1，拥有用户的贷款记录、还款历史和信用卡使用书等信贷信息。

参与方乙（如银行 2）：位于地区 2，也拥有其客户的贷款记录、还款历史和信用卡使用书等信贷信息。

参与方丙（如某在线借贷平台）：收集了用户的借贷记录、还款状况和用户在平台的交易行为数据。

可以理解，在水平分布场景下，多个参与方各自独立持有本地样本的样本标签，因多个参与方各自维护不同样本，无需就样本的标签值进行交互或共享。

以上介绍训练数据垂直分布于多个参与方中的典型情况。另外，用于应对训练数据垂直分布的联邦学习，也称横向联邦学习。

需理解，训练数据在多个参与方中的分布不限于水平分布和垂直分布，还存在更加复杂的情况，在此不作穷举。本说明书实施例方案适用于所有的数据分布场景，后文中为帮助理解方案，主要会结合纵向联邦学习和横向联邦学习进行示例性说明。

4) 关于全局模型参数在多个参与方中的分布。

需理解，全局模型参数在多个参与方中的分布，一般需要基于训练数据在多个参与方中的分布进行适应性设计。比如说，在横向联邦学习场景下，比较简单的情況是，多个参与方可以各自持有全局模型的完整副本。

又比如说，在纵向联邦学习场景下，多个参与方持有相同样本的不同特征部分，相应地，各个参与方可以只持有全局模型参数中与处理本地特征部分相关的那部分模型参数，此时，各个参与方的本地模型为全局模型的局部模型。

以上，为帮助理解，介绍了中间方、多个参与方的关联关系和实体形式，以及训练数据和全局模型参数在多个参与方中的分布。接下来，结合图 2 和更多的实施例，进一步介绍图 1 示意的联邦训练方案的具体实施步骤。

图 2 为本说明书实施例披露的联邦训练方法涉及的中间方和多个参与方之间的通信交互示意图，其中联邦训练方法包括对各个参与方中的本地模型和中间方维护的全局模型的多轮次迭代训练，其中任一的第 $k + 1$ 轮次的迭代训练包括以下交互步骤：

首先，在步骤 S201，每个参与方（图 2 中示意为 P_i 方）利用本轮对应的一批次本地训练数据 d ，计算本地模型当前的第一参数 x_k 对应的第一梯度 ∇_k 。

需理解，在本轮迭代训练中，多个参与方各自利用的本地训练数据对应相同的一批次

样本标识（例如一批次用户标识）。迭代轮次数和样本标识之间的映射关系，可以是多个参与方提前协商好的，也可以在每轮次迭代训练中，由指定参与方随机采样一批次样本标识后，广播给其他参与方，本说明书实施例对此不作限定。

5 在一个实施例中，假定联邦学习为纵向联邦学习，多个参与方各自持有相同样本的不同特征部分，此时，每个参与方（ P_i 方）可以根据多个参与方提前协商好的迭代轮次数和训练样本标识之间的映射关系，确定本轮对应的一批次样本标识，然后利用该一批次样本标识对应的本地特征部分作为一批次本地训练数据；或者， P_i 方可以根据中间方提供的一批次样本标识，获取对应的本地特征和标签，作为一批次本地训练数据。总之，在纵向联邦学习场景下，多个参与方在每轮次各自利用的本地训练数据对应同一批次的样本标识；
10 另外， P_i 方的本地模型可以为全局模型的第 i 个局部模型。

在另一个实施例中，假定联邦学习是横向联邦学习，多个参与方各自持有不同样本的完整特征，此时， P_i 方可以随机采样一批次样本数据，作为本轮使用的一批次训练数据。总之，多个参与方在每轮次确定使用的本地训练数据时完全互相独立；另外， P_i 方的本地模型可以为全局模型的副本。

15 需理解，文中将 P_i 方部署的本地模型在本轮（第 $k+1$ 轮）迭代中的起始参数称为第一参数（记作 x_k ），本地模型的参数在本轮迭代中会发生多次更新。在本轮迭代为首轮的情况下，第一参数 x_0 可以由 P_i 方本地初始化得到，或者，也可以从中间方获取；在本轮迭代不是首轮的情况下，第一参数 x_k 是经过上一轮（第 k 轮）更新后的参数。另外，第一参数中的“第一”，以及文中他处的“第一”、“第二”等类似用语，均是为了区分同类事物，
20 不具有排序等其他限定作用。

一般地，计算模型参数的梯度需要明确：模型优化目标中的目标函数（如损失函数）、模型参数的当前参数值、训练数据（含样本特征和标签）。对于后两者，上述已对应介绍第一参数 x_k 和一批次本地训练数据 d 。对于 P_i 方使用的目标函数 $f(x)$ ，可以由 P_i 方自行选定或设计。或者，也可以采用多个参与方预先协商好的函数类型，如交叉熵损失函数，等等。
25 另外，对于梯度的具体计算，可以采用已有方式实现，不作赘述。

由上， P_i 方可以根据本轮（记作第 $k+1$ 轮）使用的本地训练数据和目标函数 $f(x)$ ，计算出本地模型当前的第一参数 x_k 对应的第一梯度 ∇_k （或记作 ∇f_k ）。

接着，在步骤 S202， P_i 方根据第一梯度 ∇f_k 和当前的本地近似矩阵 B_k ，计算指示参数更新方向的目标向量 p_k ，本地近似矩阵 B_k 用于近似当前的第一参数 x_k 所对应海森矩阵的逆
30 矩阵。

需说明，相对于直接将第一梯度 ∇f_k （梯度为模型参数的一阶导数）直接作为指示参数更新方向的目标向量 p_k ，在第一梯度 ∇f_k 的基础上额外引入海森矩阵（为模型参数的二阶导数）的逆矩阵，具有以下有益效果：1）可以更准确地估计目标函数 $f(x)$ 的局部曲率，从而在更少的迭代次数内实现更快的收敛速度；2）通过考虑二阶导数，能够在某种程度上抵抗训练过程中的数值不稳定性和过拟合风险，增强模型的稳健性；3）有助于更好地理解 and 适应目标函数 $f(x)$ 的非线性变化，从而提高模型的训练效果和预测精度。

然而，在模型参数量较大时，无论是海森矩阵还是海森矩阵的逆，矩阵的求解都需要耗费大量的计算资源。由此，借鉴 BFGS 算法的思想，提出放弃显式地计算海森矩阵和其逆，而是通过维护一个对海森矩阵的逆的估计来更新优化模型参数，文中将海森矩阵的逆的估计称为近似矩阵。

Pi 方中的本地近似矩阵 B_k 用于近似本地模型当前的第一参数 x_k 所对应海森矩阵的逆矩阵。需理解，跟本地模型参数类似地，本地近似矩阵 B_k 在每轮次迭代中也会被更新。在本轮迭代为首轮的情况下，本地近似矩阵 B_k 可以由 Pi 方本地初始化得到，例如，初始化为单位矩阵，或者，也可以从中间方获取；在本轮迭代不是首轮的情况下，本地近似矩阵 B_k 是经过上一轮更新而得到。另外，在纵向联邦学习的场景下，Pi 方的本地模型参数为全局模型参数的一部分，相应地，本地近似矩阵 B_k 的量级较小，需要占用的存储资源较少。

在本步骤中，可以计算第一梯度 ∇f_k 和当前的本地近似矩阵 B_k 之间的乘积的相反数，作为指示参数更新方向的目标向量 p_k ，对此可记作下式：

$$p_k = -B_k * \nabla f_k \quad (1)$$

需理解，本步骤中目标向量 p_k 的计算还可以采用公式（1）的变体，例如，在公式右侧的表达式上乘以一个缩放系数。

由上，每个参与方（Pi 方）可以计算出指示参数更新方向的目标向量 p_k 。

然后，在步骤 S203，Pi 方利用目标向量 p_k 和获取的步长 α_k 将本地模型当前的第一参数 x_k 更新为第二参数 x_k' ，并计算该第二参数 x_k' 对应的第二梯度 $\nabla f_k'$ 。需说明，本轮训练中后续还会对第二参数 x_k' 进行进一步更新。

对于上述步长 α_k 的获取，在一个实施例中，Pi 方基于目标向量 p_k ，可以通过线搜索确定步长，使得目标函数在本轮使用的一批次本地训练数据上的下降幅度较大。需理解，其中线搜索可以采用精确线搜索：通过解析方法或高精度数值方法精确地确定最优步长。这种方法在理论上可以得到最佳步长，但在实践中往往计算成本较高；或者，也可以采用非精确线搜索：通过某些准则和迭代过程近似地确定一个好的步长，非精确线搜索方法如

Backtracking 线搜索、Goldstein 条件和 Wolfe 条件等，是实际应用中更常用的方法，因为它们提供满意收敛速度的同时减少了计算成本。

进一步，借鉴 L-BFGS-B 算法的思想，提出支持 Pi 方设置本地的业务约束条件，从而在通过线搜索确定步长 α_k 时，需要以满足本地约束条件为前提。对于本地约束条件的来源，在一种可能的情况中，可以由 Pi 方根据业务需求等自行设定；示例性的，本地的业务约束条件可以是预算约束、曝光多样性约束、参数的非负性约束。在另一种可能的情况中，可以由多个参数方共同协商全局约束条件后，再各自根据本地的模型参数和训练数据等，适应性设定；示例性的，全局约束条件同样可以是预算约束、曝光多样性约束、参数的非负性约束等。

另一方面，对于本地约束条件的数学形式，基本可以分为等式约束和不等式约束，在文中分别表示为 $h(x) = 0$ 和 $g(x) \leq 0$ ，需理解，其中 $h(x) = 0$ 表示若干等式约束式的集合， $g(x) \leq 0$ 表示若干不等式约束式的集合，另外， $h(x) = 0$ 和 $g(x) \leq 0$ 并非必须同时存在，可以只有其中的某一类约束。为帮助理解，举个简单的例子，假定在广告推荐场景下，要将营销成本控制在预算以内，此时，对于目标用户群体，可以利用本地模型的参数 x 计算出各个目标用户点击广告的概率，再结合每条广告的成本计算出总成本的期望值，约束条件可以表示为期望值减去预算小于或等于零，这符合不等式 $g(x) \leq 0$ 的形式。

基于此，本步骤可以实施为：Pi 方基于目标向量 p_k ，通过线搜索确定在满足本地约束条件的前提下，使得目标函数在一批次本地训练数据上的下降幅度达到预定标准的步长 α_k ，例如达到最大幅度或者大于预定阈值。其中本地约束条件 $g(x) \leq 0$ 和 $h(x) = 0$ 在本轮迭代中被转化为以下形式：

$$g(x_k + \alpha_k * p_k) \leq 0 \quad (2)$$

$$h(x_k + \alpha_k * p_k) = 0 \quad (3)$$

在式 (2) 和 (3) 中， x_k 表示本地模型当前的第一参数， α_k 表示要求解的步长， p_k 表示指示参数更新方向的目标向量，可以理解，计算 α_k 和 p_k 的目的是更新第一参数 x_k ，相应地，在已知 p_k 和 x_k 的情况下，可以以更新后的模型参数（即 $x_k + \alpha_k * p_k$ ）仍满足本地约束条件为目标，求解 α_k 。

示例性的，通过线搜索确定满足本地约束条件的步长 α_k 可以采用投影法、罚函数法和二分搜索等。

以上，介绍通过线搜索确定步长 α_k ，使得（可选的，在满足本地约束条件的前提下）目标函数在本轮使用的一批次本地训练数据上的下降幅度达到预定标准。

在另一个实施例中，步长 α_k 可以由工作人根据实际经验预先设定，例如，随着迭代轮次的增加，步长 α_k 阶梯式减小。

以上介绍本轮中使用的步长 α_k 的获取。基于获取的步长 α_k ，利用目标向量 p_k 和获取的步长 α_k 将本地模型当前的第一参数 x_k 更新为第二参数 x_k' 。示例性的，可以采用下式：

$$x_k' = x_k + \alpha_k * p_k \quad (4)$$

在一个实施例中，若上述在确定步长 α_k 时没有引入本地约束条件，则在利用公式（4）计算出第二参数 x_k' 后，还可以将第二参数 x_k' 更新为满足本地约束条件。需理解，对于本地约束条件的介绍可以参见前述实施例中的相关描述。又或者，即使上述在确定步长 α_k 时已经引入本地约束条件，也可以仍执行针对第二参数 x_k' 的更新操作，以确保第二参数 x_k' 满足本地约束条件。

进一步，在一个具体的实施例中，本地约束条件包括不等式约束条件 $q(x) \leq 0$ ；相应地，上述将第二参数 x_k' 更新为满足本地约束条件可以包括：通过投影法，将第二参数 x_k' 中超出不等式约束条件所对应约束范围的分量映射回约束边界。示例性的，对于每个不满足不等式约束的分量 $q_i(x_k') > 0$ ，更新点 x_k' 的对应分量可以通过以下投影运算得到：

$$x_i(k)' = \max(x_i(k)', 0) \quad (5)$$

在另一个具体的实施例中，本地约束条件包括等式约束条件 $h(x) = 0$ ；相应地，上述将第二参数 x_k' 更新为满足本地约束条件可以包括：通过求解基于目标函数 $f(x)$ 和等式约束条件 $h(x) = 0$ 而构建的拉格朗日函数，更新第二参数 x_k' 。

示例性的，构建的拉格朗日函数（或称增广目标函数）形如下式：

$$L(x_k', \lambda) = f(x_k') + \lambda^T * h(x_k') \quad (6)$$

其中， T 表示转置运算符，需理解，当 $h(x_k')$ 为一组等式时，拉格朗日乘子 λ 为向量， $\lambda^T * h(x_k')$ 可以实现利用向量 λ 对 $h(x_k')$ 对应的一组等式进行线性组合。另外，在更新 x_k' 时，需要同时更新拉格朗日乘子 λ ，并且，更新点 x_k' 和拉格朗日乘子 λ 具体可以通过求解拉格朗日函数的梯度为零的条件得到。

由上，可以得到利用公式（4）计算出的第二参数 x_k' ，或者，可以得到进一步更新为满足本地约束条件的第二参数 x_k' 。

进一步，可以计算第二参数 x_k' 对应的第二梯度 $\nabla f_k'$ 。如前所述，计算模型参数的梯度需要明确：模型优化目标中的目标函数、模型参数的当前参数值、训练数据，相应地，计算第二梯度 $\nabla f_k'$ 用到的是本地的目标函数 $f(x)$ 、第二参数 x_k' 、本轮对应的上述一批本地训练数据 d 。

以上，Pi 方可以得到第二参数 x_k' 对应的第二梯度 $\nabla f_k'$ 。

之后，在步骤 S204，Pi 方将第二参数 x_k' 和第二梯度 $\nabla f_k'$ 发送给中间方。

5 在一个实施例中，考虑到直接传递参数明文或梯度明文，仍存在泄露 Pi 方本地原始训练数据的风险，由此提出，Pi 方对第二参数 x_k' 和第二梯度 $\nabla f_k'$ 分别进行加噪处理后，再发送给中间方。示例性的，加噪处理中添加的噪声可以是可量化隐私保护的差分隐私（Differential Privacy，简称 DP）噪声。

在另一个实施例中，Pi 方可以直接将第二参数 x_k' 和第二梯度 $\nabla f_k'$ 的明文发送给中间方。

由上，中间方可以接收到每个参与方提供的第二参数 x_k' 和第二梯度 $\nabla f_k'$ 。

10 于是，在步骤 S205，中间方基于多个参与方对应提供的多个第二参数 $\{x_k'\}$ 和多个第二梯度 $\{\nabla f_k'\}$ ，确定本轮更新后的全局模型参数 $x_{k+1}^{(G)}$ 和全局近似矩阵 $B_{k+1}^{(G)}$ 。

具体地，将对应多个本地模型的多个第二参数 $\{x_k'\}$ 聚合为对应全局模型的参数，作为本轮更新后的全局模型参数 $x_{k+1}^{(G)}$ 。以及，将对应多个本地模型的多个第二梯度 $\{\nabla f_k'\}$ 聚合为对应全局模型的参数梯度，作为本轮更新后的全局梯度 $\nabla f_{k+1}^{(G)}$ 。需理解，对于参数和梯度的聚合方式，与本地模型和全局模型之间的关系相适应，另外，聚合时还可以引入权重，比如说，中间方倾向于对权威性更强或训练数据的量级更大的参与方所提供的参数和梯度，15 分配更高的权重。

在一个实施例中，假定联邦学习为纵向联邦学习，多个参与方对应部署的多个本地模型均为全局模型的局部模型，此时，可以根据局部模型和全局模型之间的结构关系，对多个第二参数 $\{x_k'\}$ 进行拼接等处理，可以得到全局模型参数 $x_{k+1}^{(G)}$ 。类似地，对多个第二梯度20 $\{\nabla f_k'\}$ 进行拼接等处理，可以得到全局梯度 $\nabla f_{k+1}^{(G)}$ 。

在另一个实施例中，假定联邦学习为横向联邦学习，多个参与方对应部署的多个本地模型均为全局模型的副本，此时，可以将对多个第二参数 $\{x_k'\}$ 进行加权求和的结果，作为全局模型参数 $x_{k+1}^{(G)}$ 。示例性的，其中对参数的加权求和可以采用下式：

$$x_{k+1}^{(G)} = \sum_{i=1}^n \beta_i * i_{-}x_k' \quad (7)$$

25 其中， $i_{-}x_k'$ 中的 i 显式地表示第 i 个参与方， β_i 表示中间方为 $i_{-}x_k'$ 分配的权重， n 表示多个参与方的个数。

类似的，可以将对多个第二梯度 $\{\nabla f_k'\}$ 进行加权求和的结果，作为本轮更新后的全局梯度 $\nabla f_{k+1}^{(G)}$ 。示例性的，其中对梯度的加权求和可以采用下式：

$$\nabla f_{k+1}^{(G)} = \sum_{i=1}^n \gamma_i * i_{-}\nabla f_k' \quad (8)$$

30 其中， $i_{-}\nabla f_k'$ 中的 i 显式地表示第 i 个参与方， γ_i 表示中间方为 $i_{-}\nabla f_k'$ 分配的权重， n 表示

多个参与方的个数。需理解， γ_i 与 β_i 可以相同，也可以不同。

另一方面，在一个实施例中，还可以将聚合得到的全局模型参数 $x_{k+1}^{(G)}$ 进一步更新为满足全局约束条件。需说明，对全局约束条件的介绍可以参见前述实施例中的相关描述；对基于全局约束条件更新全局模型参数 $x_{k+1}^{(G)}$ 的介绍，可以参见前述实施例中对Pi方基于本地约束条件更新本地模型参数的相关描述，不另作赘述。如此，可以使得训练出的模型满足业务的全局约束。

基于以上得到的全局模型参数 $x_{k+1}^{(G)}$ 和全局梯度 $g_{k+1}^{(G)}$ ，可以进一步确定本轮更新后的全局近似矩阵 $B_{k+1}^{(G)}$ 。对于全局近似矩阵 $B_{k+1}^{(G)}$ 的更新，可以借鉴BFGS算法中更新近似矩阵的方式，区别主要在于，传统的BFGS算法用于集中式机器学习而非分布式机器学习，不存在局部和全局的概念性区分。示例性的，可以采用下式(9)将全局近似矩阵 $B_k^{(G)}$ 更新为全局近似矩阵 $B_{k+1}^{(G)}$ 。

$$B_{k+1}^{(G)} = B_k^{(G)} + \frac{S \cdot S^T}{S^T \cdot Y} - \left(\frac{B_k^{(G)} \cdot Y \cdot Y^T \cdot B_k^{(G)}}{Y^T \cdot B_k^{(G)} \cdot Y} \right) \quad (9)$$

其中， $S = x_{k+1}^{(G)} - x_k^{(G)}$ 表示步长差， $Y = g_{k+1}^{(G)} - g_k^{(G)}$ 表示全局梯度差，上标 T 表示转置运算符。需理解，对于其中的 $x_k^{(G)}$ 、 $g_k^{(G)}$ 和 $B_k^{(G)}$ ，在本轮迭代不是首轮，即 $k > 0$ 的情况下，这三个数据经过上一轮迭代而得到；在本轮迭代是首轮，即 $k = 0$ 的情况下，这三个数据可以是经过初始化而得到，示例性的，中间方可以从多个参与方分别接收本地初始化产生的 x_0 、 B_0 ，从而将 $\{x_0\}$ 和 $\{B_0\}$ 分别对应聚合为 $x_0^{(G)}$ 和 $B_0^{(G)}$ ，以及，直接将 $g_0^{(G)}$ 初始化为零向量。

观察公式(9)可知，中间方计算 $B_{k+1}^{(G)}$ 需要存储 $B_k^{(G)}$ ，假设全局模型的参数维数为1千万，那么，矩阵 $B_k^{(G)}$ 的维数为1千万*1千万，这对内存的消耗是巨大的。由此提出，可以进一步借鉴L-BFGS算法的思想，观察到 $B_{k+1}^{(G)}$ 实际上是基于 $B_0^{(G)}$ 和 k 组 S 和 Y 迭代得到的，其中 $B_0^{(G)}$ 一般可以直接初始化为单位矩阵，因此其占用的内存空间非常小，于是，可以只存储 $B_1^{(G)}$ 和 k 组 S 和 Y ，比如说，当需要计算 $B_{10}^{(G)}$ ，就用单位矩阵以及存储下的 S_1 和 Y_1 到 S_{10} 和 Y_{10} 计算就可以了，这样一个时间换空间的办法，可以让全局模型的参数为1千万个维度的情况下，存储1千万*1千万的矩阵变为了存储20个1千万维的向量，有效节省了内存空间。

另外，当内存迭代次数非常大的时候，可以丢掉一些存不下的数据。假定设置的存储向量数为100组 S 和 Y ，当迭代次数超过100时，就会丢掉第1组 S 和 Y ，每多一次迭代就扔掉最前的 S 和 Y ，这样虽然会损失一些精度，但是可用性也足够高。

由上，中间方可以确定根据多个参与方在本轮提供的参数和梯度，确定本轮更新后的

全局模型参数 $x_{k+1}^{(G)}$ 和全局近似矩阵 $B_{k+1}^{(G)}$ 。

之后,在步骤 S206,中间方基于本轮更新后的全局模型参数 $x_{k+1}^{(G)}$ 和全局近似矩阵 $B_{k+1}^{(G)}$,分别确定每个参与方(Pi方)对应的目标参数 $x^{(t)}$ 和目标矩阵 $B^{(t)}$ 。

需理解,针对每个参与方,中间方可以根据全局模型与该参与方本地模型之间参数变量的对应关系,进行目标参数 $x^{(t)}$ 和目标矩阵 $B^{(t)}$ 的确定。

在一个实施例中,假定联邦学习是纵向联邦学习,此时,针对每个参与方,中间方可以根据全局模型与该参与方部署在本地的局部模型之间参数变量的对应关系,从本轮更新后的全局模型参数 $x_{k+1}^{(G)}$ 中提取与本地模型对应的参数部分,作为目标参数 $x^{(t)}$ 。

类似地,针对每个参与方,中间方可以从全局近似矩阵 $B_{k+1}^{(G)}$ 中提取与该参与方的本地模型对应的子矩阵,作为目标矩阵 $B^{(t)}$ 。

在另一个实施例中,假定联邦学习是横向联邦学习,且每个参与方本地部署的是全局模型的副本,此时,中间方可以直接将全局模型参数 $x_{k+1}^{(G)}$ 确定为目标参数 $x^{(t)}$,以及,将全局近似矩阵 $B_{k+1}^{(G)}$ 确定为目标矩阵 $B^{(t)}$ 。

由上,中间方可以分别为多个参与方中的每个参与方确定对应的目标参数 $x^{(t)}$ 和目标矩阵 $B^{(t)}$ 。从而在步骤 S207,中间方可以将各组目标参数和目标矩阵提供给对应的参与方。

之后,在步骤 S208,每个参与方(Pi方)将本地模型参数和本地近似矩阵对应更新为中间方提供的目标参数 $x^{(t)}$ 和目标矩阵 $B^{(t)}$,用于下一轮(第 $k+2$ 轮)迭代训练。示例性的,可以将本步骤的执行记作:本地模型参数 $x_{k+1} = x^{(t)}$,本地近似矩阵 $B_{k+1} = B^{(t)}$ 。

通过执行图2中示出的上述交互步骤,可以实现本说明书实施例披露的联邦学习方案中的任一轮次迭代更新。如本领域技术人员所知,对模型的训练一般持续到模型收敛,相应地,在上述每轮迭代更新中,中间方在聚合得到本轮更新后的全局模型参数后,可以判断该全局模型参数是否满足收敛条件(如达到最大迭代次数或目标函数收敛),满足收敛条件就停止迭代,否则,将更新后的模型参数发送回参与方,继续新一轮迭代更新。

综上,采用本说明书实施例提供的联邦学习方案,通过对BFGS系算法进行改造,使其适用于联邦学习框架,从而在联邦学习中,能够更好地保护多个参与方各自的数据隐私,减少数据传输量,并且提高模型的优化效率。

与上述联邦学习方法相对应的,本说明书实施例还披露联邦学习装置。图3为本说明书实施例披露的联邦训练装置集成于任一参与方时的功能结构示意图,所述装置300通过以下模块执行任一轮迭代训练:

第一梯度计算模块 310，配置为利用本轮对应的一批次本地训练数据，计算本地模型当前的第一参数对应的第一梯度；目标向量计算模块 320，配置为根据所述第一梯度和当前的本地近似矩阵，计算指示参数更新方向的目标向量，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；第一参数更新模块 330，配置为利用所述目标向量和获取的步长将所述第一参数更新为第二参数，并计算所述第二参数对应的第二梯度；中间数据传输模块 340，配置为将所述第二参数和第二梯度发送给中间方；目标数据接收模块 350，配置为从所述中间方接收目标参数和目标矩阵；所述目标参数和目标矩阵由中间方基于所述多个参与方各自提供的第二参数和第二梯度，更新全局模型参数和全局近似矩阵后而确定；本地更新模块 360，配置为将本地模型参数和本地近似矩阵对应更新为所述目标参数和目标矩阵，用于下一轮迭代训练。

在一个实施例中，所述本轮为首轮，所述第一参数和本地近似矩阵通过本地初始化而得到。

在一个实施例中，所述第一梯度的计算还基于用于优化本地模型的目标函数；所述装置 300 还包括步长获取模块 370，配置为：基于所述目标向量，通过线搜索确定满足本地约束条件的步长，使得所述目标函数在所述一批次本地训练数据上的下降幅度大于预定阈值，所述本地约束条件根据业务需求而预先设定。

在一个实施例中，第一参数更新模块 330 具体配置为：计算所述第一参数与所述目标向量和所述步长之间乘积的和，作为第三参数；将所述第三参数更新为满足本地约束条件的所述第二参数，所述本地约束条件根据业务需求而预先设定。

进一步，在一个具体的实施例中，本地约束条件包括不等式约束条件；第一参数更新模块 330 进一步配置为：通过投影法，将所述第三参数中超出所述不等式约束条件所对应约束范围的分量映射回约束边界。

在另一个具体的实施例中，所述第一梯度的计算还基于用于优化本地模型的目标函数，所述本地约束条件包括等式约束条件；第一参数更新模块 330 进一步配置为：通过求解基于所述目标函数和等式约束条件而构建的拉格朗日函数，将所述第三参数更新为所述第二参数。

在一个实施例中，所述中间数据传输模块 340 具体配置为：在对所述第二参数和/或第二梯度添加差分隐私噪声后发送至中间方。

在一个实施例中，所述多个参与方各自在本轮使用的一批次本地训练数据对应相同样本的不同特征部分，所述多个参与方各自部署的本地模型为全局模型的局部模型。

图 4 为本说明书实施例披露的联邦训练装置集成于中间方时的功能结构示意图，中间方与多个参与方协同进行联邦训练。所述装置 400 通过以下模块执行任一轮迭代训练：

中间数据接收模块 410，配置为从任意一个参与方接收其本地模型当前的第二参数和对应的第二梯度，所述第二参数是利用目标向量和步长对第一参数进行更新而得到，所述目标向量是根据第一梯度和当前的本地近似矩阵而计算得到，所述第一梯度是所述第一参数在本轮对应的一批次本地训练数据上的梯度，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；全局数据更新模块 420，配置为基于所述多个参与方各自提供的第二参数和第二梯度，确定本轮更新后的全局模型参数和全局近似矩阵；目标数据发送模块 430，配置为基于本轮更新后的全局模型参数和全局近似矩阵，将任意一个参与方对应的目标参数和目标矩阵提供给该参与方，使得该参与方利用所述目标参数和目标矩阵对应更新本地模型参数和本地近似矩阵，用于下一轮迭代训练。

在一个实施例中，全局数据更新模块 420 包括：全局参数聚合单元 421，配置为将所述多个参与方各自提供的第二参数聚合为对应全局模型的参数，作为本轮更新后的全局模型参数；全局梯度聚合单元 422，配置为将所述多个参与方各自提供的第二梯度聚合为对应全局模型的参数梯度，作为本轮更新后的全局梯度；全局矩阵更新单元 423，配置为基于本轮更新后的全局模型参数和上一轮更新后的全局模型参数，以及本轮更新后的全局梯度和上一轮更新后的全局梯度，确定本轮更新后的全局近似矩阵。

在一个具体的实施例中，所述全局参数聚合单元 421 具体配置为：将聚合得到的参数更新为满足全局约束条件后，作为所述本轮更新后的全局模型参数，所述全局约束条件基于业务需求而预先设定。

在一个具体的实施例中，所述全局模型用于执行用户推荐任务，所述全局约束条件包括以下中的至少一项：预算约束、曝光多样性约束、参数的非负性约束。

在一个具体的实施例中，其中，所述第二参数满足其对应参与方中的本地约束条件，该本地约束条件基于所述全局约束条件、本地模型和本地训练数据而适应性设定。

需理解，为了描述的方便，描述以上装置时以功能分为各种模块分别描述。当然，在实施本说明书时可以把各模块的功能在同一个或多个软件和/或硬件中实现。

The specification also provides a computer-readable non-transitory storage medium, wherein the storage medium stores a computer program, which, when executed by a processor, can be used to perform one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Herein, the computer-readable non-transitory storage medium or media may include one or more semiconductor-based or other integrated circuits (ICs)

(such, as for example, field-programmable gate arrays (FPGAs) or application-specific ICs (ASICs)), hard disk drives (HDDs), hybrid hard drives (HHDs), optical discs, optical disc drives (ODDs), magneto-optical discs, magneto-optical drives, floppy diskettes, floppy disk drives (FDDs), magnetic tapes, solid-state drives (SSDs), RAM-drives, SECURE DIGITAL cards or drives, any other suitable computer-readable non-transitory storage media, or any suitable combination of two or more of these, where appropriate. A computer-readable non-transitory storage medium may be volatile, non-volatile, or a combination of volatile and non-volatile, where appropriate.

The specification also provides a computer system. FIG. 5 illustrates an example computer system 500. In particular embodiments, one or more computer systems 500 perform one or more steps of one or more methods described or illustrated herein. In particular embodiments, one or more computer systems 500 provide functionality described or illustrated herein. In particular embodiments, software running on one or more computer systems 500 performs one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Particular embodiments include one or more portions of one or more computer systems 500. Herein, reference to a computer system may encompass a computing device, and vice versa, where appropriate. Moreover, reference to a computer system may encompass one or more computer systems, where appropriate.

This disclosure contemplates any suitable number of computer systems 500. This disclosure contemplates computer system 500 taking any suitable physical form. As example and not by way of limitation, computer system 500 may be an embedded computer system, a system-on-chip (SOC), a single-board computer system (SBC) (such as, for example, a computer-on-module (COM) or system-on-module (SOM)), a desktop computer system, a laptop or notebook computer system, an interactive kiosk, a mainframe, a mesh of computer systems, a mobile telephone, a personal digital assistant (PDA), a server, a tablet computer system, or a combination of two or more of these. Where appropriate, computer system 500 may include one or more computer systems 500; be unitary or distributed; span multiple locations; span multiple machines; span multiple data centers; or reside in a cloud, which may include one or more cloud components in one or more networks. Where appropriate, one or more computer systems 500 may perform without substantial spatial or temporal limitation one or more steps of one or more methods described or illustrated herein. As an example and not by way of limitation, one or more computer systems 500 may perform in real time or in batch mode one or more steps of one or more methods described or illustrated herein. One or more computer systems 500 may perform at different times or at different locations one or more steps of one or more methods described or illustrated herein,

where appropriate.

In particular embodiments, computer system 500 includes a processor 502, memory 504, storage 506, an input/output (I/O) interface 508, a communication interface 510, and a bus 512. Although this disclosure describes and illustrates a particular computer system having a particular
5 number of particular components in a particular arrangement, this disclosure contemplates any suitable computer system having any suitable number of any suitable components in any suitable arrangement.

In particular embodiments, processor 502 includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute
10 instructions, processor 502 may retrieve (or fetch) the instructions from an internal register, an internal cache, memory 504, or storage 506; decode and execute them; and then write one or more results to an internal register, an internal cache, memory 504, or storage 506. In particular embodiments, processor 502 may include one or more internal caches for data, instructions, or addresses. This disclosure contemplates processor 502 including any suitable number of any
15 suitable internal caches, where appropriate. As an example and not by way of limitation, processor 502 may include one or more instruction caches, one or more data caches, and one or more translation lookaside buffers (TLBs). Instructions in the instruction caches may be copies of instructions in memory 504 or storage 506, and the instruction caches may speed up retrieval of those instructions by processor 502. Data in the data caches may be copies of data of 50 in memory
20 504 or storage 506 for instructions executing at processor 502 to operate on; the results of previous instructions executed at processor 502 for access by subsequent instructions executing at processor 502 or for writing to memory 504 or storage 506; or other suitable data. The data caches may speed up read or write operations by processor 502. The TLBs may speed up virtual-address translation for processor 502. In particular embodiments, processor 502 may include one or more internal
25 registers for data, instructions, or addresses. This disclosure contemplates processor 502 including any suitable number of any suitable internal registers, where appropriate. Where appropriate, processor 502 may include one or more arithmetic logic units (ALUs); be a multi-core processor; or include one or more processors 502. Although this disclosure describes and illustrates a particular processor, this disclosure contemplates any suitable processor.

In particular embodiments, memory 504 includes main memory for storing instructions for
30 processor 502 to execute or data for processor 502 to operate on. As an example and not by way of limitation, computer system 500 may load instructions from storage 506 or another source (such as, for example, another computer system 500) to memory 504. Processor 502 may then load the instructions from memory 504 to an internal register or internal cache. To execute the instructions,

processor 502 may retrieve the instructions from the internal register or internal cache and decode them. During or after execution of the instructions, processor 502 may write one or more results (which may be intermediate or final results) to the internal register or internal cache. Processor 502 may then write one or more of those results to memory 504. In particular embodiments, processor 502 executes only instructions in one or more internal registers or internal caches or in memory 504 (as opposed to storage 506 or elsewhere) and operates only on data in one or more internal registers or internal caches or in memory 504 (as opposed to storage 506 or elsewhere). One or more memory buses (which may each include an address bus and a data bus) may couple processor 502 to memory 504. Bus 512 may include one or more memory buses, as described below. In particular embodiments, one or more memory management units (MMUs) reside between processor 502 and memory 504 and facilitate accesses to memory 504 requested by processor 502. In particular embodiments, memory 504 includes random access memory (RAM). This RAM may be volatile memory, where appropriate. Where appropriate, this RAM may be dynamic RAM (DRAM) or static RAM (SRAM). Moreover, where appropriate, this RAM may be single-ported or multi-ported RAM. This disclosure contemplates any suitable RAM. Memory 504 may include one or more memories 504, where appropriate. Although this disclosure describes and illustrates particular memory, this disclosure contemplates any suitable memory.

In particular embodiments, storage 506 includes mass storage for data or instructions. As an example and not by way of limitation, storage 506 may include a hard disk drive (HDD), a floppy disk drive, flash memory, an optical disc, a magneto-optical disc, magnetic tape, or a Universal Serial Bus (USB) drive or a combination of two or more of these. Storage 506 may include removable or non-removable (or fixed) media, where appropriate. Storage 506 may be internal or external to computer system 500, where appropriate. In particular embodiments, storage 506 is non-volatile, solid-state memory. In particular embodiments, storage 506 includes read-only memory (ROM). Where appropriate, this ROM may be mask-programmed ROM, programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), electrically alterable ROM (EAROM), or flash memory or a combination of two or more of these. This disclosure contemplates mass storage 506 taking any suitable physical form. Storage 506 may include one or more storage control units facilitating communication between processor 502 and storage 506, where appropriate. Where appropriate, storage 506 may include one or more storages 506. Although this disclosure describes and illustrates particular storage, this disclosure contemplates any suitable storage.

In particular embodiments, I/O interface 508 includes hardware, software, or both, providing one or more interfaces for communication between computer system 500 and one or more I/O

devices. Computer system 500 may include one or more of these I/O devices, where appropriate. One or more of these I/O devices may enable communication between a person and computer system 500. As an example and not by way of limitation, an I/O device may include a keyboard, keypad, microphone, monitor, mouse, printer, scanner, speaker, still camera, stylus, tablet, touch
5 screen, trackball, video camera, another suitable I/O device or a combination of two or more of these. An I/O device may include one or more sensors. This disclosure contemplates any suitable I/O devices and any suitable I/O interfaces 508 for them. Where appropriate, I/O interface 508 may include one or more device or software drivers enabling processor 502 to drive one or more of these I/O devices. I/O interface 508 may include one or more I/O interfaces 508, where
10 appropriate. Although this disclosure describes and illustrates a particular I/O interface, this disclosure contemplates any suitable I/O interface.

In particular embodiments, communication interface 510 includes hardware, software, or both providing one or more interfaces for communication (such as, for example, packet-based communication) between computer system 500 and one or more other computer systems 500 or
15 one or more networks. As an example and not by way of limitation, communication interface 510 may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for communicating with a wireless network, such as a WI-FI network. This disclosure contemplates any suitable network and any suitable communication interface 510 for it. As an example and not
20 by way of limitation, computer system 500 may communicate with an ad hoc network, a personal area network (PAN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), or one or more portions of the Internet or a combination of two or more of these. One or more portions of one or more of these networks may be wired or wireless. As an example, computer system 500 may communicate with a wireless PAN (WPAN) (such as, for
25 example, a BLUETOOTH WPAN), a WI-FI network, a WI-MAX network, a cellular telephone network (such as, for example, a Global System for Mobile Communications (GSM) network), or other suitable wireless network or a combination of two or more of these. Computer system 500 may include any suitable communication interface 510 for any of these networks, where appropriate. Communication interface 510 may include one or more communication interfaces
30 510, where appropriate. Although this disclosure describes and illustrates a particular communication interface, this disclosure contemplates any suitable communication interface.

In particular embodiments, bus 512 includes hardware, software, or both coupling components of computer system 500 to each other. As an example and not by way of limitation, bus 512 may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced

Industry Standard Architecture (EISA) bus, a front-side bus (FSB), a HYPERTRANSPORT (HT) interconnect, an Industry Standard Architecture (ISA) bus, an INFINIBAND interconnect, a low-pin-count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCIe) bus, a serial advanced technology attachment (SATA) bus, a Video Electronics Standards Association local (VLB) bus, or another suitable bus or a combination of two or more of these. Bus 512 may include one or more buses 512, where appropriate. Although this disclosure describes and illustrates a particular bus, this disclosure contemplates any suitable bus or interconnect.

本领域技术人员应该可以意识到，在上述一个或多个示例中，本发明所描述的功能可以用硬件、软件、固件或它们的任意组合来实现。当使用软件实现时，可以将这些功能存储在计算机可读介质中或者作为计算机可读介质上的一个或多个指令或代码进行传输。

以上所述的具体实施方式，对本发明的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本发明的具体实施方式而已，并不用于限定本发明的保护范围，凡在本发明的技术方案的基础之上，所做的任何修改、等同替换、改进等，均应包括在本发明的保护范围之内。

权 利 要 求 书

1.一种联邦训练方法，涉及中间方和多个参与方，所述方法应用于其中任意一个参与方，所述方法涉及的任一轮迭代训练包括：

利用本轮对应的一批次本地训练数据，计算本地模型当前的第一参数对应的第一梯度；

根据所述第一梯度和当前的本地近似矩阵，计算指示参数更新方向的目标向量，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；

利用所述目标向量和获取的步长将所述第一参数更新为第二参数，并计算所述第二参数对应的第二梯度；

将所述第二参数和第二梯度发送给中间方；

从所述中间方接收目标参数和目标矩阵；所述目标参数和目标矩阵由中间方基于所述多个参与方各自提供的第二参数和第二梯度，更新全局模型参数和全局近似矩阵后而确定；

将本地模型参数和本地近似矩阵对应更新为所述目标参数和目标矩阵，用于下一轮迭代训练。

2.根据权利要求1所述的方法，其中，所述本轮为首轮，所述第一参数和本地近似矩阵通过本地初始化而得到。

3.根据权利要求1所述的方法，其中，所述第一梯度的计算还基于用于优化本地模型的目标函数；其中，所述步长的获取包括：

基于所述目标向量，通过线搜索确定满足本地约束条件的步长，使得所述目标函数在所述一批次本地训练数据上的下降幅度大于预定阈值，所述本地约束条件根据业务需求而预先设定。

4.根据权利要求1所述的方法，其中，利用所述目标向量和获取的步长将所述第一参数更新为第二参数，包括：

计算所述第一参数与所述目标向量和所述步长之间乘积的和，作为第三参数；

将所述第三参数更新为满足本地约束条件的所述第二参数，所述本地约束条件根据业务需求而预先设定。

5.根据权利要求4所述的方法，其中，本地约束条件包括不等式约束条件；其中，将所述第三参数更新为满足本地约束条件的所述第二参数，包括：

通过投影法，将所述第三参数中超出所述不等式约束条件所对应约束范围的分量映射回约束边界。

6.根据权利要求4所述的方法，其中，所述第一梯度的计算还基于用于优化本地模型

的目标函数，所述本地约束条件包括等式约束条件；其中，将所述第三参数更新为满足本地约束条件的所述第二参数，包括：

通过求解基于所述目标函数和等式约束条件而构建的拉格朗日函数，将所述第三参数更新为所述第二参数。

7.根据权利要求 1 所述的方法，其中，在将所述第二参数和第二梯度发送给中间方之前，所述方法还包括：

对所述第二参数和/或第二梯度添加差分隐私噪声。

8.根据权利要求 1 所述的方法，其中，所述多个参与方各自在本轮使用的一批次本地训练数据对应相同样本的不同特征部分，所述多个参与方各自部署的本地模型为全局模型的局部模型。

9.一种联邦训练方法，涉及中间方和多个参与方，所述方法应用于所述中间方，所述方法涉及的任一轮迭代训练包括：

从任意一个参与方接收其本地模型当前的第二参数和对应的第二梯度，所述第二参数是利用目标向量和步长对第一参数进行更新而得到，所述目标向量是根据第一梯度和当前的本地近似矩阵而计算得到，所述第一梯度是所述第一参数在本轮对应的一批次本地训练数据上的梯度，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；

基于所述多个参与方各自提供的第二参数和第二梯度，确定本轮更新后的全局模型参数和全局近似矩阵；

基于本轮更新后的全局模型参数和全局近似矩阵，将任意一个参与方对应的目标参数和目标矩阵提供给该参与方，使得该参与方利用所述目标参数和目标矩阵对应更新本地模型参数和本地近似矩阵，用于下一轮迭代训练。

10.根据权利要求 9 所述的方法，其中，基于所述多个参与方各自提供的第二参数和第二梯度，确定本轮更新后的全局模型参数和全局近似矩阵，包括：

将所述多个参与方各自提供的第二参数聚合为对应全局模型的参数，作为本轮更新后的全局模型参数；

将所述多个参与方各自提供的第二梯度聚合为对应全局模型的参数梯度，作为本轮更新后的全局梯度；

基于本轮更新后的全局模型参数和上一轮更新后的全局模型参数，以及本轮更新后的全局梯度和上一轮更新后的全局梯度，确定本轮更新后的全局近似矩阵。

11.根据权利要求 10 所述的方法，其中，将所述多个参与方各自提供的第二参数聚合

为对应全局模型的参数，作为本轮更新后的全局模型参数，包括：

将聚合得到的参数更新为满足全局约束条件后，作为所述本轮更新后的全局模型参数，所述全局约束条件基于业务需求而预先设定。

12.根据权利要求 11 所述的方法，其中，所述全局模型用于执行用户推荐任务，所述全局约束条件包括以下中的至少一项：预算约束、曝光多样性约束、参数的非负性约束。

13.根据权利要求 11 所述的方法，其中，所述第二参数满足其对应参与方中的本地约束条件，该本地约束条件基于所述全局约束条件、本地模型和本地训练数据而适应性设定。

14.一种联邦训练装置，涉及中间方和多个参与方，所述装置集成于其中任意一个参与方，所述装置通过以下模块执行任一轮迭代训练：

第一梯度计算模块，配置为利用本轮对应的一批次本地训练数据，计算本地模型当前的第一参数对应的第一梯度；

目标向量计算模块，配置为根据所述第一梯度和当前的本地近似矩阵，计算指示参数更新方向的目标向量，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；

第一参数更新模块，配置为利用所述目标向量和获取的步长将所述第一参数更新为第二参数，并计算所述第二参数对应的第二梯度；

中间数据传输模块，配置为将所述第二参数和第二梯度发送给中间方；

目标数据接收模块，配置为从所述中间方接收目标参数和目标矩阵；所述目标参数和目标矩阵由中间方基于所述多个参与方各自提供的第二参数和第二梯度，更新全局模型参数和全局近似矩阵后而确定；

本地更新模块，配置为将本地模型参数和本地近似矩阵对应更新为所述目标参数和目标矩阵，用于下一轮迭代训练。

15.一种联邦训练装置，涉及中间方和多个参与方，所述装置集成于所述中间方，所述装置通过以下模块执行任一轮迭代训练：

中间数据接收模块，配置为从任意一个参与方接收其本地模型当前的第二参数和对应的第二梯度，所述第二参数是利用目标向量和步长对第一参数进行更新而得到，所述目标向量是根据第一梯度和当前的本地近似矩阵而计算得到，所述第一梯度是所述第一参数在本轮对应的一批次本地训练数据上的梯度，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；

全局数据更新模块，配置为基于所述多个参与方各自提供的第二参数和第二梯度，确

定本轮更新后的全局模型参数和全局近似矩阵；

目标数据发送模块，配置为基于本轮更新后的全局模型参数和全局近似矩阵，将任意一个参与方对应的目标参数和目标矩阵提供给该参与方，使得该参与方利用所述目标参数和目标矩阵对应更新本地模型参数和本地近似矩阵，用于下一轮迭代训练。

16.一种计算机系统，包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器集成于多个参与方中的任意一个参与方，所述多个参与方与中间方协同进行联邦训练；所述处理器执行所述程序时实现如下步骤：

利用本轮对应的一批次本地训练数据，计算本地模型当前的第一参数对应的第一梯度；

根据所述第一梯度和当前的本地近似矩阵，计算指示参数更新方向的目标向量，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；

利用所述目标向量和获取的步长将所述第一参数更新为第二参数，并计算所述第二参数对应的第二梯度；

将所述第二参数和第二梯度发送给中间方；

从所述中间方接收目标参数和目标矩阵；所述目标参数和目标矩阵由中间方基于所述多个参与方各自提供的第二参数和第二梯度，更新全局模型参数和全局近似矩阵后而确定；

将本地模型参数和本地近似矩阵对应更新为所述目标参数和目标矩阵，用于下一轮迭代训练。

17.一种计算机系统，包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器集成于中间方，所述中间方与多个参与方协同进行联邦训练；所述处理器执行所述程序时实现如下步骤：

从任意一个参与方接收其本地模型当前的第二参数和对应的第二梯度，所述第二参数是利用目标向量和步长对第一参数进行更新而得到，所述目标向量是根据第一梯度和当前的本地近似矩阵而计算得到，所述第一梯度是所述第一参数在本轮对应的一批次本地训练数据上的梯度，所述本地近似矩阵用于近似所述第一参数所对应海森矩阵的逆矩阵；

基于所述多个参与方各自提供的第二参数和第二梯度，确定本轮更新后的全局模型参数和全局近似矩阵；

基于本轮更新后的全局模型参数和全局近似矩阵，将任意一个参与方对应的目标参数和目标矩阵提供给该参与方，使得该参与方利用所述目标参数和目标矩阵对应更新本地模型参数和本地近似矩阵，用于下一轮迭代训练。

18.一种计算机可读非易失性存储介质，其中，所述存储介质存储有计算机程序，所述计算机程序被处理器执行时实现上述权利要求 1-13 任一项所述的方法。