(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113033082 A
(43)申请公布日 2021.06.25

(21)申请号 202110258488.8

(22)申请日 2021.03.10

(71)申请人 中国科学技术大学苏州高等研究院
地址 215000 江苏省苏州市工业园区独墅湖高教区仁爱路166号

(72)发明人 朱宗卫 周学海 李曦 王超

(74)专利代理机构 苏州创元专利商标事务所有限公司 32103
代理人 范晴 丁浩秋

(51)Int.Cl.
G06F 30/27(2020.01)
G06N 20/20(2019.01)
H04L 29/08(2006.01)
G06F 111/08(2020.01)

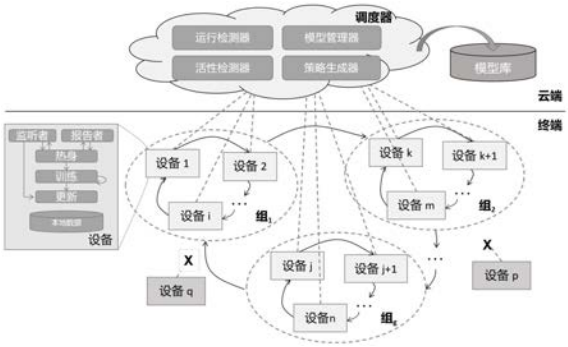
权利要求书2页 说明书8页 附图5页

(54)发明名称

基于异构算力感知的去中心化联邦学习框架及建模方法

(57)摘要

本发明公开了一种基于异构算力感知的去中心化联邦学习框架,包括云端协调器和若干设备端;所述云端协调器用于运行时管理、训练和参数更新方案生成、定期模型备份;所述设备端用于向云端协调器传输设备信息,在本地运行模型,更新设备端参数;所述云端协调器获取设备端一次训练时间的最小公倍数为超周期,设备端在超周期内计算不同的步长,在超周期的整数倍时聚合模型。根据设备计算能力不同而运行不同的本地步骤,在模型聚合过程中,为了减少慢节点的负面影响;采用了分布式的点对点通信方式,可以在不增加整体通信量的情况下,消除在分布式训练过程中中央服务器的通信压力。



CN 113033082 A

1.一种基于异构算力感知的去中心化联邦学习框架,其特征在于,包括云端协调器和若干设备端;

所述云端协调器用于运行时管理、训练和参数更新方案生成、定期模型备份;

所述设备端用于向云端协调器传输设备信息,在本地运行模型,更新设备端参数;

所述云端协调器获取设备端一次训练时间的最小公倍数为超周期,设备端在超周期内计算不同的步长,在超周期的整数倍时聚合模型。

2.根据权利要求1所述的基于异构算力感知的去中心化联邦学习框架,其特征在于,所述云端协调器包括活性检测器、策略生成器、运行监测器和模型管理器;

所述活性检测器通过监控设备状态添加可用设备;

所述策略生成器用于生成训练配置,向设备端发送训练配置;

所述运行监测器在每一轮通信中收集设备端的参数版本,预测下一轮的参数版本分布,并将其发送给策略生成器;

所述模型管理器定期获取最新的模型,并将其放入数据库中备份。

3.根据权利要求2所述的基于异构算力感知的去中心化联邦学习框架,其特征在于,所述策略生成器还用于根据设备端情况,确定全局同步拓扑;根据设备端异构程度分布,确定设备端同步步调不等待的最小时间,以此获得分组同步时刻,各设备端的参数版本号概率分布;根据参数版本号概率分布,确定分组同步的设备集合及其拓扑。

4.根据权利要求2所述的基于异构算力感知的去中心化联邦学习框架,其特征在于,所述策略生成器利用概率分布、期望参数版本和基于概率选择函数得到训练配置,所述概率选择函数$P(i,j)$为:

$$\begin{cases} P(i,j) = f(v_{(i,j)}) / \sum_{n=1}^{N_{avl}} f(v_{(n,j)}) \\ f(x) = \int \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right) \end{cases}$$

其中,N_{avl}是可用设备的总数,$v_{(i,j)}$是设备i在第j轮迭代中的实际参数版本,μ是所有$v_{(i,j)}$的四分位数。

5.根据权利要求1所述的基于异构算力感知的去中心化联邦学习框架,其特征在于,对设备端进行分组,确定组间同步周期为组内同步周期的整数倍。

6.根据权利要求1所述的基于异构算力感知的去中心化联邦学习框架,其特征在于,所述设备端在本地异步计算梯度和更新模型参数,在到达超周期时,设备端传递参数并执行部分模型聚合和同步;在设备端传递参数并执行部分模型聚合和同步时采用容错机制,所述容错机制为:第二设备端在工作过程中断开连接,所述第二设备端的下游设备端包括第三设备端,所述第三设备端在模型同步时无法接收到参数,在一定等待时间后,所述第三设备端向所述第二设备端发送握手消息,确认设备状态之后向第二设备端的上游设备第一设备端发出警告,所述第一设备端直接与第三设备端通信。

7.一种基于异构算力感知的去中心化联邦学习建模方法,其特征在于,包括以下步骤:

S01:在每轮训练开始之前,云端协调器的活性检测器通过监控设备状态添加可用设备;

S02:通过策略生成器生成训练配置,向设备端发送训练配置;

S03:每个设备端进入协商阶段,并将其本轮计算时间发送给云端协调器,得到该设备算力;

S04:策略生成器利用概率分布、期望参数版本和基于概率选择函数得到训练配置;各设备端根据训练配置信息异步的进行本地训练;到达更新周期时,各设备端根据云端协调器所给的拓扑进行部分模型同步,将同步后的模型广播给其他设备端;

S05:运行监测器在每一轮通信中收集设备端的参数版本,预测下一轮的参数版本分布,并将其发送给策略生成器;

S05:重复步骤S04-S05,直到模型收敛;

S06:模型管理器定期获取最新的模型,并将其放入数据库中备份。

8.根据权利要求7所述的基于异构算力感知的去中心化联邦学习建模方法,其特征在于,所述步骤S04之前还包括,策略生成器还用于根据设备端情况,确定全局同步拓扑;根据设备端异构程度分布,确定设备端同步步调不等待的最小时间,以此获得分组同步时刻,各设备端的参数版本号概率分布;根据参数版本号概率分布,确定分组同步的设备集合及其拓扑。

9.根据权利要求7所述的基于异构算力感知的去中心化联邦学习建模方法,其特征在于,所述步骤S04中,设备端在本地异步计算梯度和更新模型参数,在到达超周期时,设备端传递参数并执行部分模型聚合和同步;在设备端传递参数并执行部分模型聚合和同步时采用容错机制,所述容错机制为:第二设备端在工作过程中断开连接,所述第二设备端的下游设备端包括第三设备端,所述第三设备端在模型同步时无法接收到参数,在一定等待时间后,所述第三设备端向所述第二设备端发送握手消息,确认设备状态之后向第二设备端的上游设备第一设备端发出警告,所述第一设备端直接与第三设备端通信。

10.根据权利要求7所述的基于异构算力感知的去中心化联邦学习建模方法,其特征在于,所述步骤S04中还包括,对设备端进行分组,确定组间同步周期为组内同步周期的整数倍。

基于异构算力感知的去中心化联邦学习框架及建模方法

技术领域

[0001] 本发明属于大数据的数据汇聚技术领域，具体地涉及一种基于异构算力感知的去中心化联邦学习框架及建模方法。

背景技术

[0002] 人工智能越来越多的应用在人类生活的各个方面，然而，传统的人工智能学习面临了两个突出问题。

[0003] 1)数据孤岛问题

一个AI项目可能涉及多个领域，需要融合各个公司、各个部门的数据。(比如研究居民线上消费问题，需要各个消费平台的数据，可能还需要银行数据等等)但在现实中想要将分散在各地、各个机构的数据进行整合几乎是不可能的。

[0004] 2)数据隐私问题

GDPR的出台，使得各方对数据所有权和隐私性的关注越来越多，对用户隐私和安全管理日趋严格，导致获取数据集越来越困难。

[0005] 经典的机器学习方法基于样本数据(库)训练得到适用于不同任务和场景的机器学习模型。这些样本数据(库)一般通过从不同用户、终端、系统中收集并集中存储而得到。在实际应用场景中，这种收集样本数据的方式面临很多问题。一方面，这种方法损害了数据的隐私性和安全性。在一些应用场景中，例如金融行业、政府行业等，受限于数据隐私和安全的要求，根本无法实现对数据的集中存储；另一方面，这种方法会增加通信开销。在物联网等一些大量依赖于移动终端的应用中，这种数据汇聚的通信开销成本是非常巨大的。

[0006] 要解决大数据的困境，仅仅靠传统的方法已经出现瓶颈。两个公司简单的交换数据，法规GDPR都是不允许的。用户是原始数据的拥有者，在用户没有批准的情况下，公司间不能交换数据。如何在满足隐私监管要求的前提下，设计一个机器学习框架，让数据拥有方不暴露自己的数据，但能共同使用数据，让人工智能系统能够更加高效、准确地共同使用各自的数据，解决数据孤岛的问题。因此，一个满足隐私保护和数据安全的一个可行的解决方案就诞生了，即联邦学习。

[0007] 联邦学习允许多个用户(称为客户机)协作训练共享的全局模型，而无需分享本地设备中的数据。由中央服务器协调完成多轮联邦学习以得到最终的全局模型。其中，在每一轮开始时，中央服务器将当前的全局模型发送给参与联邦学习的客户机。每个客户机根据其本地数据训练所接收到的全局模型，训练完毕后将更新后的模型返回中央服务器。中央服务器收集到所有客户机返回的更新后，对全局模型进行一次更新，进而结束本轮更新。通过上述多轮学习和通信的方法，联邦学习消除了在单个设备上聚合所有数据的需要，克服了机器学习任务中的隐私和通信挑战，允许机器学习模型学习分散在各个用户(客户机)上存储的数据。

[0008] 联邦学习自提出以来获得了广泛的关注，并在一些场景中得以应用。联邦学习解决了数据汇聚的问题，使得一些跨机构、跨部门的机器学习模型、算法的设计和训练成为了

可能。特别地，对于移动设备中的机器学习模型应用，联邦学习表现出了良好的性能和鲁棒性。此外，对于一些没有足够的私人数据来开发精确的本地模型的用户(客户机)来说，通过联邦学习能够大大改进机器学习模型和算法的性能。但是，由于联邦学习侧重于通过分布式学习所有参与客户机(设备)的本地数据来获得高质量的全局模型，因此它无法捕获每个设备的个人信息，从而导致推理或分类的性能下降。此外，传统的联邦学习需要所有参与设备就协作训练的共同模型达成一致，这在实际复杂的物联网应用中是不现实的。研究人员将联邦学习在实际应用中面临的问题总结如下：

1)由于CPU、GPU、内存等的可变性，不同节点的系统配置可能会有所不同。节点计算能力的不平衡会加剧掉队问题，并导致一些慢节点严重落后。

[0009] 2)联邦学习框架通信量庞大。FedAvg的集中式模型聚合策略会给中央服务器带来很大的通信和计算压力，导致可扩展性差和通信瓶颈。

[0010] 3)设备分布广泛，易造成通信不可靠性，从而降低性能。

[0011] 为了解决这些异构性挑战，许多研究人员曾进行以下优化：

1)采用异步方式进行模型聚合，然而，陈旧的落后节点参数会导致不正确的收敛或迭代次数增加。

[0012] 2)采用集中模型同步和聚合方法，然而，在海量设备的情况下，通信压力剧增。

[0013] 3)采用分布式设计联邦学习框架，然而，该框架假设设备是同构的，同步聚合模型，并不适合在异构设备上训练模型。

[0014] 本发明因此而来。

发明内容

[0015] 针对传统联邦学习假设设备端计算能力是平均的。然而，当应用于异构设备时，快速设备需要等待慢速设备，这样就会浪费快速设备的计算能力的技术问题，本发明提出了一种基于异构算力感知的去中心化联邦学习框架及建模方法，根据设备计算能力不同而运行不同的本地步骤，在模型聚合过程中，为了减少慢节点的负面影响；采用了分布式的点对点通信方式，可以在不增加整体通信量的情况下，消除在分布式训练过程中中央服务器的通信压力。

[0016] 本发明的技术方案是：

一种基于异构算力感知的去中心化联邦学习框架，包括云端协调器和若干设备端；

所述云端协调器用于运行时管理、训练和参数更新方案生成、定期模型备份；

所述设备端用于向云端协调器传输设备信息，在本地运行模型，更新设备端参数；

所述云端协调器获取设备端一次训练时间的最小公倍数为超周期，设备端在超周期内计算不同的步长，在超周期的整数倍时聚合模型。

[0017] 优选的技术方案中，所述云端协调器包括活性检测器、策略生成器、运行监测器和模型管理器；

所述活性检测器通过监控设备状态添加可用设备；

所述策略生成器用于生成训练配置，向设备端发送训练配置；

所述运行监测器在每一轮通信中收集设备端的参数版本，预测下一轮的参数版本

分布，并将其发送给策略生成器；

所述模型管理器定期获取最新的模型，并将其放入数据库中备份。

[0018] 优选的技术方案中，所述策略生成器还用于根据设备端情况，确定全局同步拓扑；根据设备端异构程度分布，确定设备端同步步调不等待的最小时间，以此获得分组同步时刻，各设备端的参数版本号概率分布；根据参数版本号概率分布，确定分组同步的设备集合及其拓扑。

[0019] 优选的技术方案中，所述策略生成器利用概率分布、期望参数版本和基于概率选择函数得到训练配置，所述概率选择函数$P(i,j)$为：

$$\begin{cases} P(i,j) = f(v_{(i,j)}) / \sum_{n=1}^{N_{avl}} f(v_{(n,j)}) \\ f(x) = \int \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right) \end{cases}$$

其中，N_{avl}是可用设备的总数，$v_{(i,j)}$是设备i在第j轮迭代中的实际参数版本，μ是所有$v_{(i,j)}$的四分位数。

[0020] 优选的技术方案中，对设备端进行分组，确定组间同步周期为组内同步周期的整数倍。

[0021] 优选的技术方案中，所述设备端在本地异步计算梯度和更新模型参数，在到达超周期时，设备端传递参数并执行部分模型聚合和同步；在设备端传递参数并执行部分模型聚合和同步时采用容错机制，所述容错机制为：第二设备端在工作过程中断开连接，所述第二设备端的下游设备端包括第三设备端，所述第三设备端在模型同步时无法接收到参数，在一定等待时间后，所述第三设备端向所述第二设备端发送握手消息，确认设备状态之后向第二设备端的上游设备第一设备端发出警告，所述第一设备端直接与第三设备端通信。

[0022] 本发明还公开了一种基于异构算力感知的去中心化联邦学习建模方法，包括以下步骤：

S01：在每轮训练开始之前，云端协调器的活性检测器通过监控设备状态添加可用设备；

S02：通过策略生成器生成训练配置，向设备端发送训练配置；

S03：每个设备端进入协商阶段，并将其本轮计算时间发送给云端协调器，得到该设备算力；

S04：策略生成器利用概率分布、期望参数版本和基于概率选择函数得到训练配置；各设备端根据训练配置信息异步的进行本地训练；到达更新周期时，各设备端根据云端协调器所给的拓扑进行部分模型同步，将同步后的模型广播给其他设备端；

S05：运行监测器在每一轮通信中收集设备端的参数版本，预测下一轮的参数版本分布，并将其发送给策略生成器；

S05：重复步骤S04-S05，直到模型收敛；

S06：模型管理器定期获取最新的模型，并将其放入数据库中备份。

[0023] 优选的技术方案中，所述步骤S04之前还包括，策略生成器还用于根据设备端情况，确定全局同步拓扑；根据设备端异构程度分布，确定设备端同步步调不等待的最小时间，以此获得分组同步时刻，各设备端的参数版本号概率分布；根据参数版本号概率分布，

确定分组同步的设备集合及其拓扑。

[0024] 优选的技术方案中，所述步骤S04中，设备端在本地异步计算梯度和更新模型参数，在到达超周期时，设备端传递参数并执行部分模型聚合和同步；在设备端传递参数并执行部分模型聚合和同步时采用容错机制，所述容错机制为：第二设备端在工作过程中断开连接，所述第二设备端的下游设备端包括第三设备端，所述第三设备端在模型同步时无法接收到参数，在一定等待时间后，所述第三设备端向所述第二设备端发送握手消息，确认设备状态之后向第二设备端的上游设备第一设备端发出警告，所述第一设备端直接与第三设备端通信。

[0025] 优选的技术方案中，所述步骤S04中还包括，对设备端进行分组，确定组间同步周期为组内同步周期的整数倍。

[0026] 与现有技术相比，本发明的优点是：

本发明对传统的联邦学习进行改进，旨在解决以下问题：

1、慢节点落后严重问题：传统的联邦学习由于各个节点的系统配置不同，节点计算能力的不平衡会加剧掉队问题，并导致一些慢节点严重落后，而HADFL根据设备计算能力不同而运行不同的本地步骤，在模型聚合过程中，为了减少慢节点的负面影响，采用一种版本敏感的概率部分模型聚合方案，既不浪费慢节点的计算能力，又可以利用它们带来的噪声进行更快的训练。

[0027] 2、可扩展性差和通信瓶颈问题：传统联邦学习采用的集中式模型聚合策略会给中央服务器带来很大的通信和计算压力，导致可扩展性差和通信瓶颈，而HADFL采用了分布式的点对点通信方式，可以在不增加整体通信量的情况下，消除在分布式训练过程中中央服务器的通信压力。

附图说明

[0028] 下面结合附图及实施例对本发明作进一步描述：

图1为本发明基于异构算力感知的去中心化联邦学习框架的组成图；

图2为分布式训练、FedAvg和HADFL训练周期比较，三种设备计算能力比是4:2:1；

图3为本发明模型融合和容错机制；

图4a-4f为两组数据重复进行三次实验的实验结果。

具体实施方式

[0029] 为使本发明的目的、技术方案和优点更加清楚明了，下面结合具体实施方式并参照附图，对本发明进一步详细说明。应该理解，这些描述只是示例性的，而并非要限制本发明的范围。此外，在以下说明中，省略了对公知结构和技术的描述，以避免不必要地混淆本发明的概念。

[0030] 实施例：

下面结合附图，对本发明的较佳实施例作进一步说明。

[0031] 如图1所示，一种基于异构算力感知的去中心化联邦学习框架，包括云端协调器和若干设备端；

所述云端协调器用于运行时管理、训练和参数更新方案生成、定期模型备份；

所述设备端用于向云端协调器传输设备信息,在本地运行模型,更新设备端参数;

所述云端协调器获取设备端一次训练时间的最小公倍数为超周期,设备端在超周期内计算不同的步长,在超周期的整数倍时聚合模型。

[0032] 发明了一个基于异构算力感知的去中心化联邦学习框架(HADFL),对传统的联邦学习进行改进,旨在解决以下问题:

慢节点落后严重问题:传统的联邦学习由于各个节点的系统配置不同,节点计算能力的不平衡会加剧掉队问题,并导致一些慢节点严重落后,而HADFL根据设备计算能力不同而运行不同的本地步骤,在模型聚合过程中,为了减少慢节点的负面影响,采用一种版本敏感的概率部分模型聚合方案,既不浪费慢节点的计算能力,又可以利用它们带来的噪声进行更快的训练。

[0033] 可扩展性差和通信瓶颈问题:传统联邦学习采用的集中式模型聚合策略会给中央服务器带来很大的通信和计算压力,导致可扩展性差和通信瓶颈,而HADFL采用了分布式的点对点通信方式,可以在不增加整体通信量的情况下,消除在分布式训练过程中中央服务器的通信压力。

[0034] 该框架允许异构设备在模型聚合之前运行不同的本地步骤。利用动态预测函数,根据历史运行信息预测参数版本,对长期运行具有良好的指导作用;该框架采用分散式模型聚合策略,设备之间通过传递模型参数来进行通讯;该框架充分考虑了网络连接在运行过程中的不可靠性,采用了容错参数同步方案。

[0035] 一较佳的实施例中,云端协调器包括活性检测器、策略生成器、运行监测器和模型管理器;

所述活性检测器通过监控设备状态添加可用设备;

所述策略生成器用于生成训练配置,向设备端发送训练配置;

所述运行监测器在每一轮通信中收集设备端的参数版本,预测下一轮的参数版本分布,并将其发送给策略生成器;

所述模型管理器定期获取最新的模型,并将其放入数据库中备份。

[0036] 如图2所示,设备端在超周期Tsync(不同设备端训练一次数据的训练时间的最小公倍数)内本地计算不同的步长,只在超周期的整数倍时进行聚合模型,由此可见,HADFL可充分利用不同设备的计算能力。

[0037] 一较佳的实施例中,策略生成器的功能如下:

1)根据设备端情况,确定全局同步拓扑;

2)根据设备端异构程度分布,确定设备端同步步调不等待的最小时间,以此获得分组同步时刻,各设备端的参数版本号概率分布;

3)根据参数版本号概率分布,确定分组同步的设备集合及其拓扑。

[0038] 一较佳的实施例中,策略生成器利用概率分布、期望参数版本和基于概率选择函数得到训练配置。

[0039] 一较佳的实施例中,策略生成器的技术如下:

1)定义设备端一次训练时间的最小公倍数为超周期;

2)基于概率选择函数确定设备选中概率,所述概率选择函数$P(i,j)$为:

$$\begin{cases} P(i,j) = f(v_{(i,j)}) / \sum_{n=1}^{N_{avl}} f(v_{(n,j)}) \\ f(x) = \int \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right) \end{cases}$$

其中，N_{avl}是可用设备的总数，$v_{(i,j)}$是设备i在第j轮迭代中的实际参数版本，μ是所有$v_{(i,j)}$的四分位数。

[0040] 概率选择功能减少流浪者设备的参数对模型收敛的影响。与其他联邦学习框架相比，本框架的概率选择函数不完全丢弃落后设备，而是利用其参数带来的噪声帮助模型跳出局部最小值，更快地收敛。

[0041] 3)对设备端进行分组，确定组间同步周期为组内同步周期的整数倍。在设备端数量较多的情况下，为了便于管理和避免可能出现的系统错误，将设备分成多个组，如图1所示，组间同步周期可以为组内同步周期的整数倍。

[0042] 设备端核心技术

在设备接收到云端策略生成器发送的参数包(初始版本的模型参数、训练的超参数等)之后，进入warm-up阶段。warm-up是一种学习率优化方法，在模型训练之初选用较小的学习率，训练一段时间之后使用预设的学习率进行训练。

[0043] 一较佳的实施例中，设备端在本地异步计算梯度和更新模型参数，如图3所示，在到达超周期时，设备端传递参数并执行部分模型聚合和同步。具体的实现中，设备端采用scatter-gather的方式进行传递参数并执行部分模型聚合和同步。

[0044] 为了避免由于网络连接不稳定而导致的系统错误，在设备端传递参数并执行部分模型聚合和同步时采用容错机制，容错机制为：第二设备端在工作过程中断开连接，所述第二设备端的下游设备端包括第三设备端，所述第三设备端在模型同步时无法接收到参数，在一定等待时间后，所述第三设备端向所述第二设备端发送握手消息，确认设备状态之后向第二设备端的上游设备第一设备端发出警告，所述第一设备端直接与第三设备端通信。具体的，如图3所示，设备2在工作过程中断开连接，导致其下游设备3在模型同步时无法接收到参数。HADFL规定在一定等待时间后，设备3会向设备2发送握手消息，确认设备状态之后向设备2的上游设备1发出警告。通信时，设备1将绕过设备2，直接与设备3通信。这样可以提高整个系统的可靠性。

[0045] 系统的工作流程如下：

S01：在每轮训练开始之前，云端协调器的活性检测器通过监控设备状态添加可用设备；

S02：通过策略生成器生成训练配置，向设备端发送训练配置(即初始模型参数和训练超参数)；

S03：每个设备端进入协商阶段，并将其本轮计算时间发送给云端协调器，得到该设备算力；

S04：策略生成器利用概率分布、期望参数版本和基于概率选择函数得到训练配置；各设备端根据训练配置信息异步的进行本地训练；到达更新周期时，各设备端根据云端协调器所给的拓扑进行部分模型同步，将同步后的模型广播给其他设备端；具体的实现中，以非阻塞的方式将同步后的模型广播给其他设备端。所述概率选择函数$P(i,j)$为：

$$\begin{cases} P(i,j) = f(v_{(i,j)}) / \sum_{n=1}^{N_{avl}} f(v_{(n,j)}) \\ f(x) = \int \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right) \end{cases}$$

其中，N_{avl}是可用设备的总数，$v_{(i,j)}$是设备i在第j轮迭代中的实际参数版本，μ是所有$v_{(i,j)}$的四分位数。

[0046] S05：运行监测器在每一轮通信中收集设备端的参数版本，预测下一轮的参数版本分布，并将其发送给策略生成器；

S05：重复步骤S04-S05，直到模型收敛；

S06：模型管理器定期获取最新的模型，并将其放入数据库中备份。

[0047] 步骤S04之前还包括，策略生成器还用于根据设备端情况，确定全局同步拓扑；根据设备端异构程度分布，确定设备端同步步调不等待的最小时间，以此获得分组同步时刻，各设备端的参数版本号概率分布；根据参数版本号概率分布，确定分组同步的设备集合及其拓扑。

[0048] 步骤S04中，设备端在本地异步计算梯度和更新模型参数，在到达超周期时，设备端传递参数并执行部分模型聚合和同步；在设备端传递参数并执行部分模型聚合和同步时采用容错机制，所述容错机制为：第二设备端在工作过程中断开连接，所述第二设备端的下游设备端包括第三设备端，所述第三设备端在模型同步时无法接收到参数，在一定等待时间后，所述第三设备端向所述第二设备端发送握手消息，确认设备状态之后向第二设备端的上游设备第一设备端发出警告，所述第一设备端直接与第三设备端通信。

[0049] 步骤S04中还包括，对设备端进行分组，确定组间同步周期为组内同步周期的整数倍。

[0050] 实验设置

测试平台设置：在4个Nvidia Tesla V100 gpu上部署HADFL框架，使用PCIE Express 3.0 x8进行通信。CUDA版本是10.0.130。我们使用sleep()函数来模拟不同程度的异构，并使用一个数组来表示计算能力的比例。如[2,1]表示GPU0的计算能力是GPU1的两倍。

[0051] 模型和数据集：使用ResNet-18和vgg-16这两个CNN模型，数据集使用CIFAR-10。

[0052] 比较基准：采用两种训练方案来进行对比。(1)基于Pytorch的分布式训练方案(2) Decentralized Federated Average (Decentralized-FedAvg)

实验结果：

我们对[3,3,1,1]和[4,2,2,1]两种非均匀分布的系统进行了对比实验。训练数据分割到四个gpu上，每次选择两个gpu来执行部分同步。重复进行三次实验，实验结果如图4a-4f所示。

[0053] 同时记录达到最大测试精度所需的平均时间，如下表所示。

	ResNet-18 [3,3,1,1]		ResNet-18 [4,2,2,1]		vgg-16 [3,3,1,1]		vgg-16 [4,2,2,1]	
	accuracy	time	accuracy	time	accuracy	time	accuracy	time
Distributed training	91%	2431.38 s	91%	4076.28 s	87%	1349.73 s	87%	1791.36 s
Decentralized-FedAvg	91%	1699.05 s	91%	2747.12 s	86%	1952.01 s	86%	2424.12 s
HADFL	90%	805.00 s	91%	871.50 s	86%	794.02 s	86%	1324.04 s

[0054] 实验结果分析

由图4a-4f和表1可看出,HADFL比其他两种方案收敛速度更快。训练ResNet-18时,在[3,3,1,1]的异构分布下,HADFL实现对分布式训练加速3.02倍,对decentralized-FedAvg加速2.11倍,在[4,2,2,1]的异构分布下,对分布式训练加速4.68倍,对decentralized-FedAvg加速3.15倍;在训练vgg-16时,在[3,3,1,1]的异构分布下,HADFL实现对分布式训练加速1.70倍,对decentralized-FedAvg加速2.46倍,在[4,2,2,1]的异构分布下,对分布式训练加速1.35倍,对decentralized-FedAvg加速1.83倍。

[0055] 实施例1

金融行业数据的生产习惯和数据存储的过程当中,纬度更多是偏向于资金流,所以需要去做更多的资源整合,需要有一个非常好的办法来量化金融风险、防范系统性风险、量化用户价值,从而达到业务的指标。但无奈的是,当金融机构去整合更多的数据孤岛资源的时候,由于行业的要求,会受到一定限制。此时利用本专利基于异构算力感知的去中心化联邦学习可以实现在隐私保护和数据合规的情况下进行内外部的大数据合作。

[0056] 在金融行业,HADFL应用服务主要运用在零售信贷风控、贷记卡风控、风险定价、反洗钱、精准营销等领域。从实际应用流程上看,HADFL应用服务结合实际场景,通过HADFL系统、准备数据、训练模型、上线测试、模型优化等流程,完成联合建模,并在生产环境中投产使用。该过程最显著的特点是各方数据不出本地,确保数据隐私。

[0057] 实施例2

在医疗AI领域,获取高质量的医学影像数据难度较大。一方面来自于医学影像数据前处理和标注所需的投入,占据了开发成本的绝大部分,工作量巨大;其次由于医学影像数据绝对的私密性,数据的拥有方采取高度保护措施,也加大了AI研发机构获取数据的难度,然而,只有获取更多的数据进行训练,AI模型才能更强健。

[0058] HADFL能在无需共享患者数据的情况下,即可实现协作与分散化的神经网络训练。各节点负责训练其自身的本地模型,并定期提交给参数服务器。该服务器不断累积并聚合各自的贡献,进而创建一个全局模型,分享给所有节点。全局模型可以分散到各个医院或研究中心,利用它们本地的数据进行训练,之后再将训练后的模型回传,而数据始终保存在本地。通过各个医院、研究中心等机构不断地训练,“全局AI模型”不断壮大,再分享给各个节点,实现了数据与模型训练的“双赢”。

[0059] 应当理解的是,本发明的上述具体实施方式仅仅用于示例性说明或解释本发明的原理,而不构成对本发明的限制。因此,在不偏离本发明的精神和范围的情况下所做的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。此外,本发明所附权利要求旨在涵盖落入所附权利要求范围和边界、或者这种范围和边界的等同形式内的全部变化和修改例。

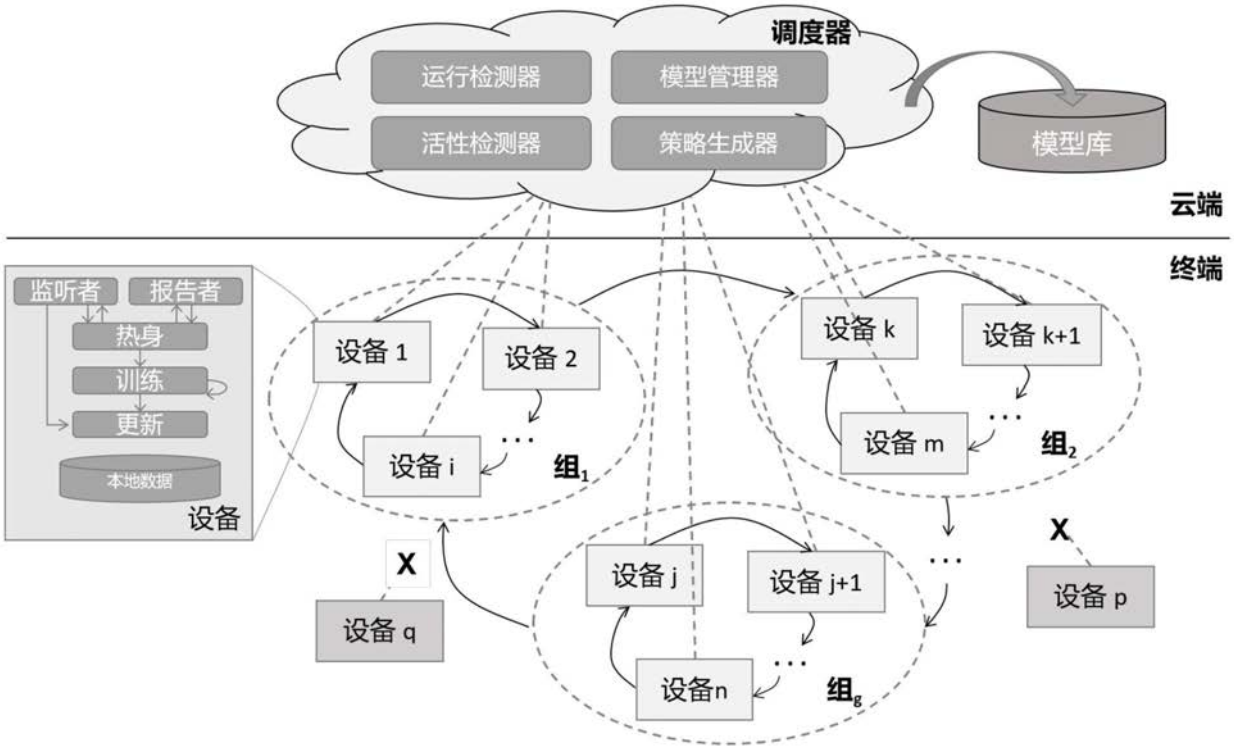

调度器
运行检测器
模型管理器
活性检测器
策略生成器
模型库
云端
终端
监听者
报告者
热身
训练
更新
本地数据
设备
设备 1
设备 2
设备 i
组1
设备 k
设备 k+1
设备 m
组2
X
设备 q
设备 j
设备 j+1
设备n
组g
X
设备 p

图1

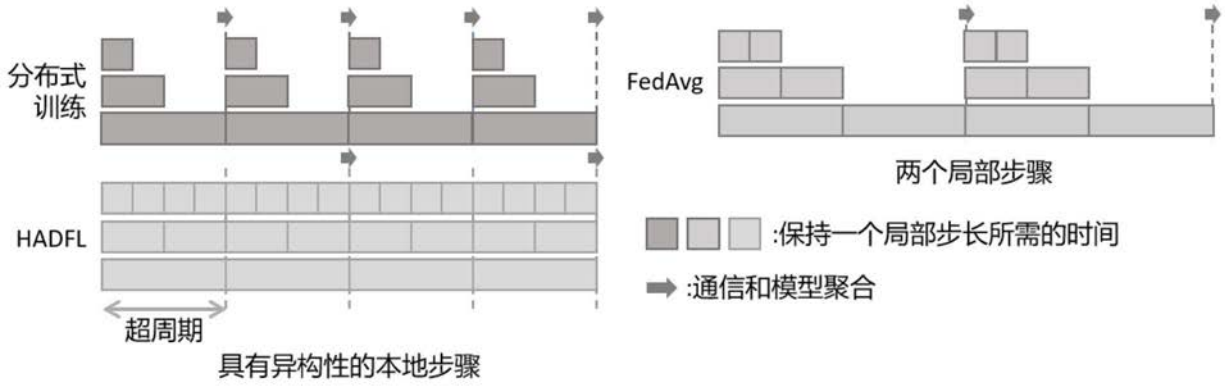

分布式
训练
HADFL
超周期
具有异构性的本地步骤
FedAvg
两个局部步骤
:保持一个局部步长所需的时间
:通信和模型聚合

图2

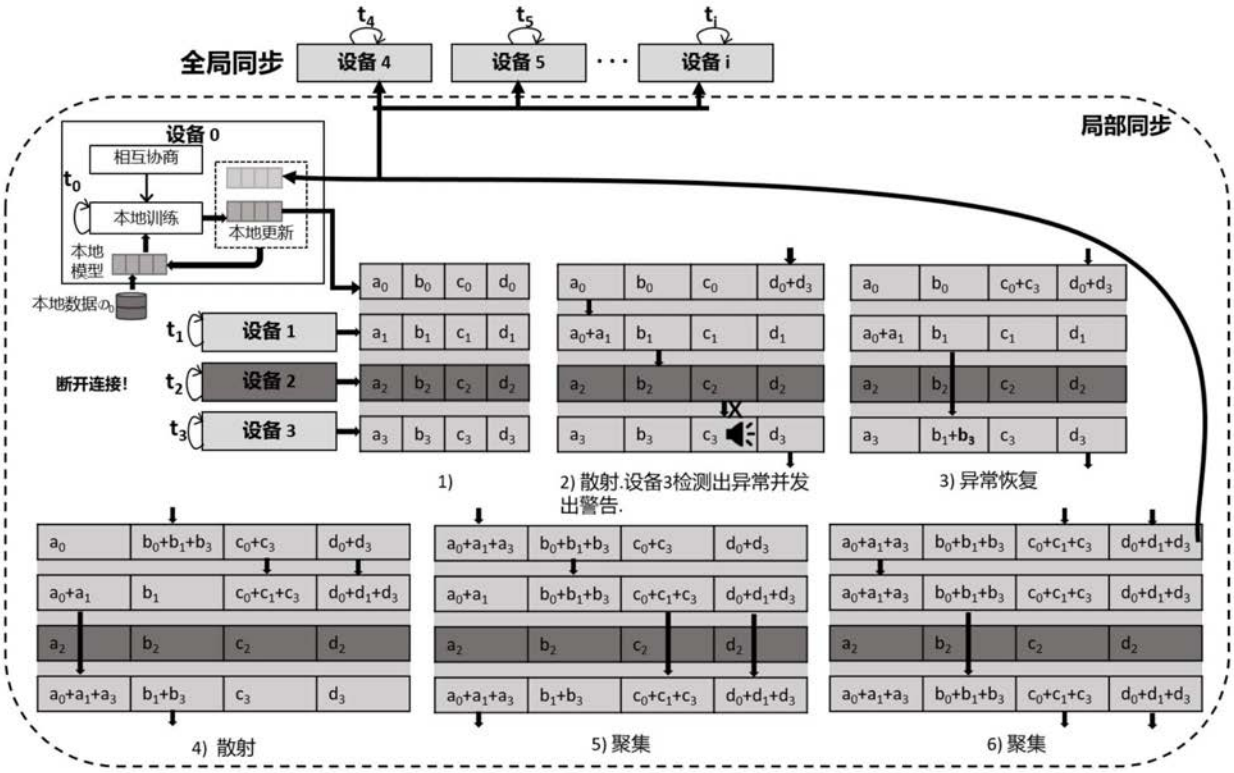

全局同步
设备 4
设备 5
设备 i
局部同步
设备 0
相互协商
本地训练
本地更新
本地模型
本地数据
设备 1
设备 2
设备 3
断开连接!
1)
2) 散射.设备3检测出异常并发出警告.
3) 异常恢复
4) 散射
5) 聚集
6) 聚集

图3

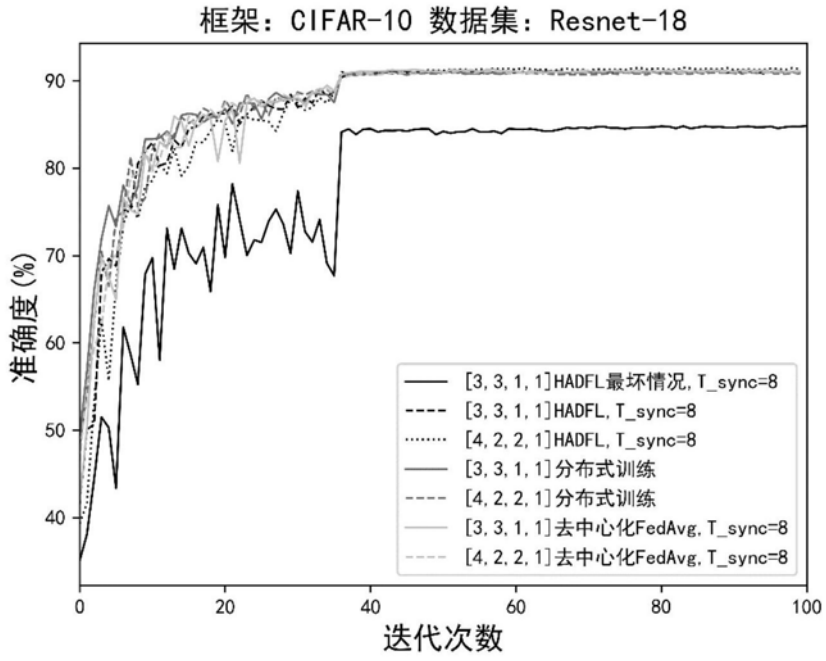

框架：CIFAR-10 数据集：Resnet-18
准确度(%)
迭代次数
[3, 3, 1, 1]HADFL最坏情况, T_sync=8
[3, 3, 1, 1]HADFL, T_sync=8
[4, 2, 2, 1]HADFL, T_sync=8
[3, 3, 1, 1]分布式训练
[4, 2, 2, 1]分布式训练
[3, 3, 1, 1]去中心化FedAvg, T_sync=8
[4, 2, 2, 1]去中心化FedAvg, T_sync=8

图4a

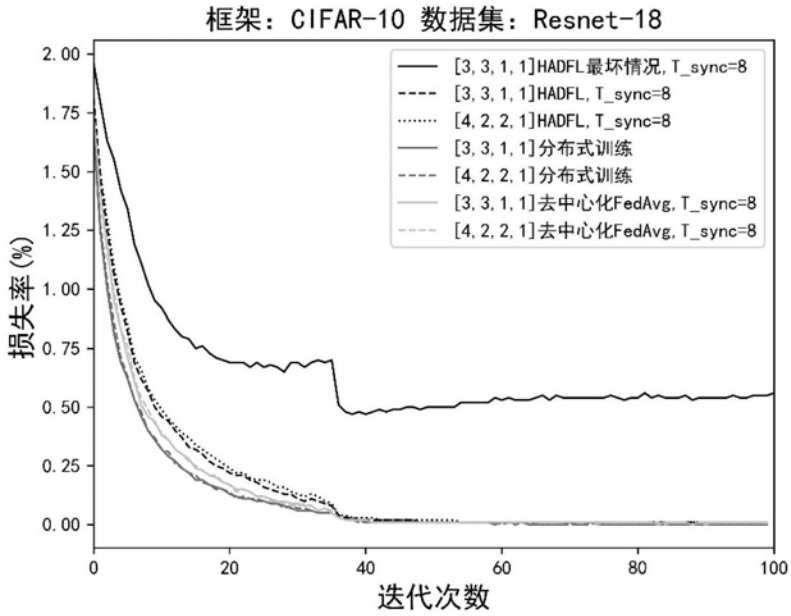

框架：CIFAR-10 数据集：Resnet-18
损失率(%)
迭代次数
[3, 3, 1, 1]HADFL最坏情况, T_sync=8
[3, 3, 1, 1]HADFL, T_sync=8
[4, 2, 2, 1]HADFL, T_sync=8
[3, 3, 1, 1]分布式训练
[4, 2, 2, 1]分布式训练
[3, 3, 1, 1]去中心化FedAvg, T_sync=8
[4, 2, 2, 1]去中心化FedAvg, T_sync=8

图4b

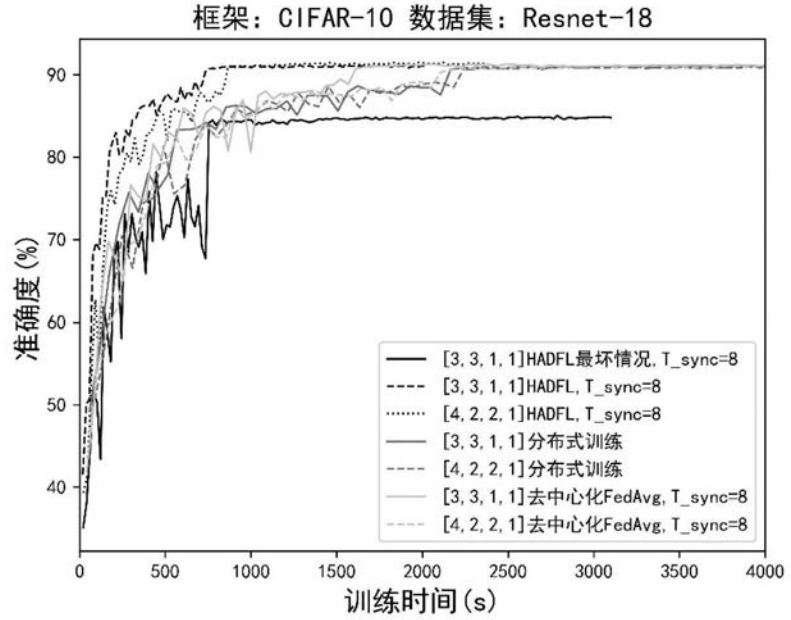

框架：CIFAR-10 数据集：Resnet-18
准确度(%)
训练时间(s)
[3, 3, 1, 1]HADFL最坏情况, T_sync=8
[3, 3, 1, 1]HADFL, T_sync=8
[4, 2, 2, 1]HADFL, T_sync=8
[3, 3, 1, 1]分布式训练
[4, 2, 2, 1]分布式训练
[3, 3, 1, 1]去中心化FedAvg, T_sync=8
[4, 2, 2, 1]去中心化FedAvg, T_sync=8

图4c

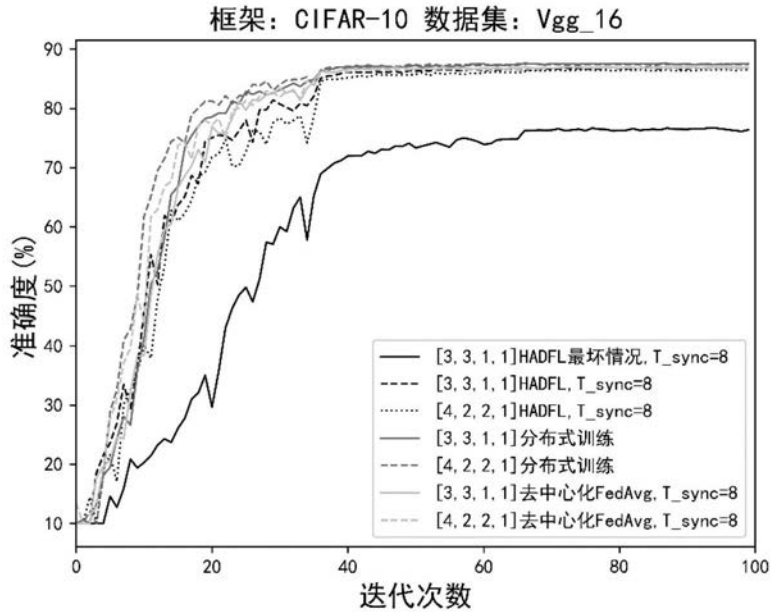

框架：CIFAR-10 数据集：Vgg_16
准确度(%)
迭代次数
[3, 3, 1, 1]HADFL最坏情况, T_sync=8
[3, 3, 1, 1]HADFL, T_sync=8
[4, 2, 2, 1]HADFL, T_sync=8
[3, 3, 1, 1]分布式训练
[4, 2, 2, 1]分布式训练
[3, 3, 1, 1]去中心化FedAvg, T_sync=8
[4, 2, 2, 1]去中心化FedAvg, T_sync=8

图4d

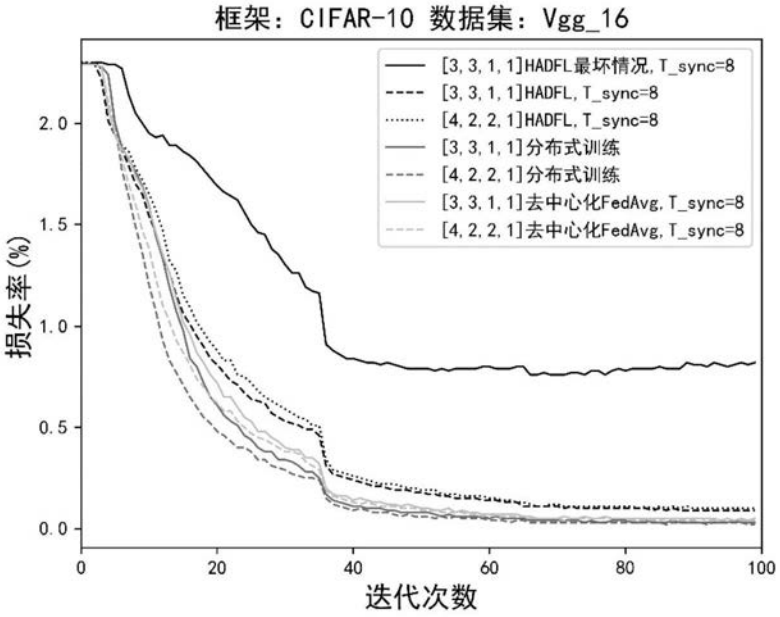

框架：CIFAR-10 数据集：Vgg_16
损失率(%)
迭代次数
[3, 3, 1, 1]HADFL最坏情况, T_sync=8
[3, 3, 1, 1]HADFL, T_sync=8
[4, 2, 2, 1]HADFL, T_sync=8
[3, 3, 1, 1]分布式训练
[4, 2, 2, 1]分布式训练
[3, 3, 1, 1]去中心化FedAvg, T_sync=8
[4, 2, 2, 1]去中心化FedAvg, T_sync=8

图4e

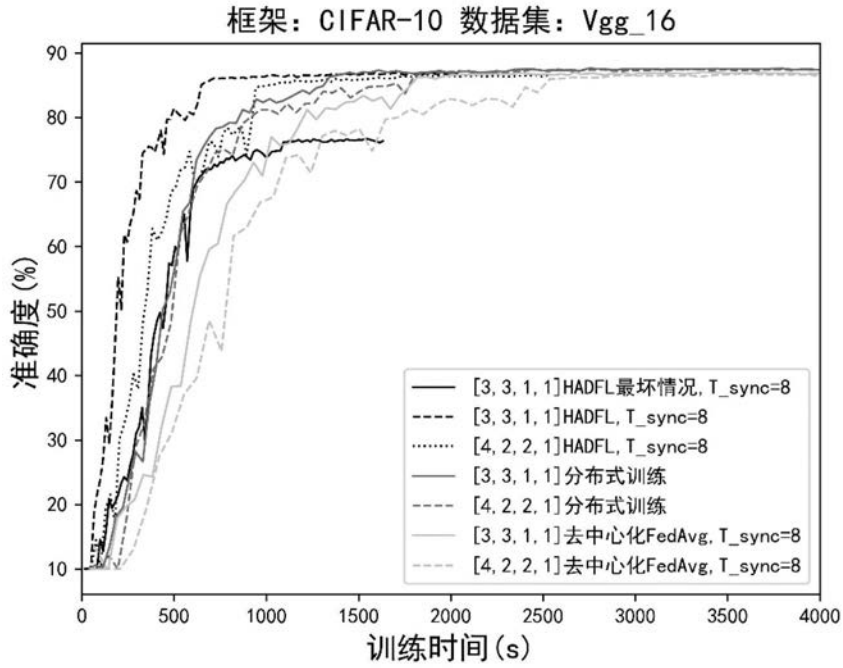

框架：CIFAR-10 数据集：Vgg_16
准确度(%)
训练时间(s)
[3, 3, 1, 1]HADFL最坏情况, T_sync=8
[3, 3, 1, 1]HADFL, T_sync=8
[4, 2, 2, 1]HADFL, T_sync=8
[3, 3, 1, 1]分布式训练
[4, 2, 2, 1]分布式训练
[3, 3, 1, 1]去中心化FedAvg, T_sync=8
[4, 2, 2, 1]去中心化FedAvg, T_sync=8

图4f