



(12) 发明专利申请

(10) 申请公布号 CN 112565331 A

(43) 申请公布日 2021.03.26

(21) 申请号 202011205945.9

(22) 申请日 2020.11.02

(71) 申请人 中山大学

地址 510260 广东省广州市海珠区新港西路135号

(72) 发明人 刘芳 黎燊 肖依 金航

(74) 专利代理机构 广州粤高专利商标代理有限公司 44102

代理人 张金福

(51) Int. Cl.

H04L 29/08 (2006.01)

G06N 3/04 (2006.01)

G06N 3/08 (2006.01)

G06N 20/00 (2019.01)

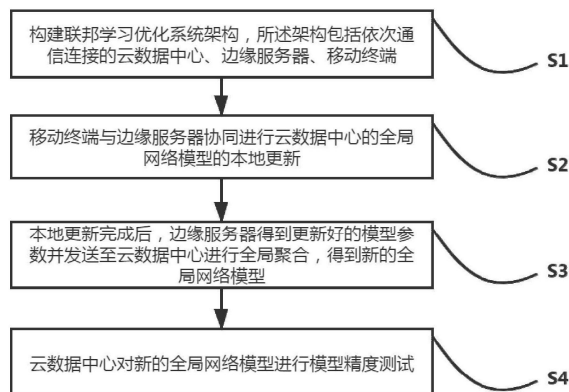
权利要求书2页 说明书8页 附图1页

(54) 发明名称

一种基于边缘计算的端-边协同联邦学习优化方法

(57) 摘要

本发明公开了一种基于边缘计算的端-边协同联邦学习优化方法,包括以下步骤:构建联邦学习优化系统架构,包括云数据中心、边缘服务器、移动终端;移动终端与边缘服务器协同进行模型的本地更新;本地更新完成后,边缘服务器将更新好的模型参数发送至云数据中心进行全局聚合,得到新的模型;云数据中心对新的模型进行模型精度测试。本发明通过建立联邦学习优化系统架构,加入边缘服务器与移动终端进行协同的本地更新,极大减少了移动终端的计算开销;并在协同更新完成后在边缘服务器上对多个移动终端的运算结果进行先聚合,以减少所需的全局聚合次数,极大降低通信延时,加快联邦学习的模型训练速度。



1. 一种基于边缘计算的端-边协同联邦学习优化方法,其特征在於,包括以下步骤:

S1: 构建联邦学习优化系统架构,所述架构包括依次通信连接的云数据中心、边缘服务器、移动终端;

S2: 移动终端与边缘服务器协同进行云数据中心的全局网络模型的本地更新;

S3: 本地更新完成后,边缘服务器得到更新好的模型参数并发送至云数据中心进行全局聚合,得到新的全局网络模型;

S4: 云数据中心对新的全局网络模型进行模型精度测试。

2. 根据权利要求1所述的一种基于边缘计算的端-边协同联邦学习优化方法,其特征在於,步骤S1具体过程为:

S101: 所述云数据中心将原始模型参数发送至边缘服务器;

S102: 所述边缘服务器将原始模型参数保存至本地并发送至区域内的移动终端。

3. 根据权利要求1所述的一种基于边缘计算的端-边协同联邦学习优化方法,其特征在於,步骤S2的具体过程为:

S201: 将云数据中心的全局网络模型分割为浅层和剩余层;

S202: 移动终端执行浅层函数计算,得到浅层计算结果,并将浅层计算结果以及对应的数据标签发送至边缘服务器进行聚合,得到浅层输出值矩阵;

S203: 边缘服务器将浅层输出值矩阵作为剩余层函数输入并执行计算,得到剩余层计算结果,然后进行模型参数更新。

4. 根据权利要求3所述的一种基于边缘计算的端-边协同联邦学习优化方法,其特征在於,步骤S201中所述全局网络模型为CNN网络模型。

5. 根据权利要求3所述的一种基于边缘计算的端-边协同联邦学习优化方法,其特征在於,步骤S202中移动终端执行浅层函数计算,其计算公式具体为: $C(x) = x_{conv}$ ①;其中, x 为移动终端本地样本数据, x_{conv} 为浅层计算结果。

6. 根据权利要求3所述的一种基于边缘计算的端-边协同联邦学习优化方法,其特征在於,步骤S202中聚合得到浅层输出值矩阵,其计算公式具体为:

$$x_{conv} \leftarrow [x_{conv}^1, x_{conv}^2, x_{conv}^3, \dots, x_{conv}^k] \text{ ②};$$

$$y_{label} \leftarrow [y_{label}^1, y_{label}^2, y_{label}^3, \dots, y_{label}^k] \text{ ③};$$

其中, x_{conv} 为浅层计算结果, x_{conv}^k 为第k个移动终端的浅层计算结果, y_{label} 为样本数据标签, y_{label}^k 为第k个移动终端的浅层计算结果对应的样本数据标签。

7. 根据权利要求3所述的一种基于边缘计算的端-边协同联邦学习优化方法,其特征在於,步骤S203中边缘服务器执行剩余层函数计算,其公式具体为: $E(x_{conv}) = x_{couput}$ ④;其中, x_{couput} 为剩余层计算结果。

8. 根据权利要求3所述的一种基于边缘计算的端-边协同联邦学习优化方法,其特征在於,步骤S203中进行模型参数更新,其具体公式为:

$$\omega = \omega - \mu \nabla \text{loss}(x_{output}, y_{label}) \text{ ⑤};$$

其中, ω 为模型参数值, μ 为梯度下降的步长, x_{output} 为剩余层计算结果, y_{label} 为样本数据标签。

9. 根据权利要求1所述的一种基于边缘计算的端-边协同联邦学习优化方法, 其特征在于, 步骤S3中更新好的模型参数在云数据中心进行全局聚合, 其具体公式为:

$$\omega_{t+1}^i \leftarrow \sum_{m=1}^M \frac{n_m}{n} \omega_t^i \text{ ⑥};$$

其中, ω_{t+1}^i 为更新后的模型参数进行全局聚合得到的值; n 为所有参与协同本地更新的移动终端所使用的本地样本数据量总和, n_m 为第 m 个边缘服务器区域内参与协同本地更新的移动终端所使用的样本数据量。

10. 根据权利要求1所述的一种基于边缘计算的端-边协同联邦学习优化方法, 其特征在于, 步骤S4中模型精度测试的具体过程为:

S401: 采用新的模型对云数据中心保存的测试样本数据进行预测;

S402: 若预测结果达到预设模型精度指标, 则模型训练结束; 若预测结果未达到预设模型精度指标, 则云数据中心将模型参数下发, 继续进行模型训练。

一种基于边缘计算的端-边协同联邦学习优化方法

技术领域

[0001] 本发明边缘计算和联邦学习领域,更具体地,涉及一种基于边缘计算的端-边协同联邦学习优化方法。

背景技术

[0002] 随着万物互联时代的到来,网络边缘环境下的数据量的急剧增长,核心网络难以承受大量的数据传输,并且现有的一些延迟敏感性应用也无法承受大量数据传输至云数据中心处理带来的巨大延迟。因此,边缘计算应运而生,通过网络边缘环境下的边缘节点进行高效的数据处理。Gartner估计到2025年,将有75%的数据在传统云数据中心外进行处理,移动终端与边缘节点的协同数据处理将能提高网络边缘环境下的数据消化能力。

[0003] 移动互联网技术的发展推动了人工智能技术的快速发展,并且被逐渐应用在了社会生活中随处可见的生活服务中,例如语音分析、语义理解、智慧医疗以及智能驾驶等。但是由于近年来不断有网络安全事件发生,互联网用户对个人数据隐私安全的日渐重视,并且在例如医疗、金融、保险等敏感数据行业中用户数据的隐私安全是十分重要的,这些问题导致了数据的共享和融合变得十分困难,小数据以及数据孤岛问题阻碍了人工智能技术的持续健康发展以及进一步的落地。而联邦学习作为近两年学术界和产业界都十分关注的研究热点,为解决数据孤岛问题提供了全新的思路。

[0004] 但是随着移动终端设备朝着轻量化方向发展,其便携性导致了电池容量有限,本地更新时移动终端需要多次训练更新本地模型的情况,引发了高额的计算开销,并导致全局聚合过程中通信耗时严重,极大影响了联邦学习的整体效率。

[0005] 传统联邦学习为端-云两层架构,在模型的本地更新过程中,移动终端通常需要结合本地样本数据对模型进行多次训练更新,此过程会占用大量的计算存储资源,加速电量消耗。随着移动终端设备朝着轻量化发展,电池容量有限,这将严重影响到用户正常使用移动终端设备。而全局聚合过程也需要进行多次,且每次需要上传的模型参数数据量大,严重占用带宽资源,这带来了较大的通信延时,极大影响了联邦学习的整体效率。

[0006] 在现有技术中,公开号为CN111447083A的中国发明专利,于2020年07月24日公开了一种动态带宽和不可靠网络下的联邦学习架构及其压缩算法,框架包括云端、客户端和边缘服务器,边缘服务器有若干个并能够相互之间进行数据交换,其中一个作为协调器。虽然该方案通过引入边缘服务器,缩短了传统的客户端到云之间的传输距离,在一定程度上减少了客户端与服务器之间的通信延迟,但是并未能完全解决上述问题,因此,用户急需一种基于边缘计算的端-边协同联邦学习优化方法。

发明内容

[0007] 本发明为解决联邦学习过程中移动终端本地更新训练次数过多导致计算开销大以及全局聚合通信延时高的问题,提供了一种基于边缘计算的端-边协同联邦学习优化方

法。

[0008] 本发明的首要目的是为解决上述技术问题,本发明的技术方案如下:

[0009] 一种基于边缘计算的端-边协同联邦学习优化方法,包括以下步骤:S1:构建联邦学习优化系统架构,所述架构包括依次通信连接的云数据中心、边缘服务器、移动终端;S2:移动终端与边缘服务器协同进行云数据中心的全局网络模型的本地更新;S3:本地更新完成后,边缘服务器得到更新好的模型参数并发送至云数据中心进行全局聚合,得到新的全局网络模型;S4:云数据中心对新的全局网络模型进行模型精度测试。

[0010] 优选地,步骤S1具体过程为:S101:所述云数据中心将原始模型参数发送至边缘服务器;S102:所述边缘服务器将原始模型参数保存至本地并发送至区域内的移动终端。

[0011] 优选地,步骤S2的具体过程为:S201:将云数据中心的全局网络模型分割为浅层和剩余层;S202:移动终端执行浅层函数计算,得到浅层计算结果,并将浅层计算结果以及对应的数据标签发送至边缘服务器进行聚合,得到浅层输出值矩阵;S203:边缘服务器将浅层输出值矩阵作为剩余层函数输入并执行计算,得到剩余层计算结果,然后进行模型参数更新。

[0012] 优选地,步骤S201中所述模型为CNN网络模型。

[0013] 优选地,步骤S202中移动终端执行浅层函数计算,其计算公式具体为: $C(x) = x_{conv} \textcircled{1}$;其中, x 为移动终端本地样本数据, x_{conv} 为浅层计算结果。

[0014] 优选地,步骤S202中聚合得到浅层输出值矩阵,其计算公式具体为:

$$[0015] \quad x_{conv} \leftarrow [x_{conv}^1, x_{conv}^2, x_{conv}^3, \dots, x_{conv}^k] \textcircled{2};$$

$$[0016] \quad y_{label} \leftarrow [y_{label}^1, y_{label}^2, y_{label}^3, \dots, y_{label}^k] \textcircled{3};$$

[0017] 其中, x_{conv} 为浅层计算结果, x_{conv}^k 为第k个移动终端的浅层计算结果, y_{label} 为样本数据标签, y_{label}^k 为第k个移动终端的浅层计算结果对应的样本数据标签。

[0018] 优选地,步骤S203中边缘服务器执行剩余层函数计算,其公式具体为: $E(x_{conv}) = x_{couput} \textcircled{4}$;其中, x_{couput} 为剩余层计算结果。

[0019] 优选地,步骤S203中进行模型参数更新,其具体公式为:

$$[0020] \quad \omega = \omega - \mu \nabla \text{loss}(x_{output}, y_{label}) \textcircled{5};$$

[0021] 其中, ω 为模型参数值, μ 为梯度下降的步长, x_{output} 为剩余层计算结果, y_{label} 为样本数据标签。

[0022] 优选地,步骤S3中更新好的模型参数在云数据中心进行全局聚合,其具体公式为:

$$[0023] \quad \omega_{t+1}^i \leftarrow \sum_{m=1}^M \frac{n_m}{n} \omega_t^i \textcircled{6};$$

[0024] 其中, ω_{t+1}^i 为更新后的模型参数进行全局聚合得到的值; n 为所有参与协同本地更新的移动终端所使用的本地样本数据量总和, n_m 为第m个边缘服务器区域内参与协同本地更新的移动终端所使用的样本数据量。

[0025] 优选地,步骤S4中模型精度测试的具体过程为:S401:采用新的模型对云数据中心保存的测试样本数据进行预测;S402:若预测结果达到预设模型精度指标,则模型训练结束;若预测结果未达到预设模型精度指标,则云数据中心将模型参数下发,继续进行模型训练。

[0026] 与现有技术相比,本发明技术方案的有益效果是:

[0027] 本发明通过建立联邦学习优化系统架构,加入边缘服务器与移动终端进行协同的本地更新,极大减少了移动终端的计算开销;并在协同更新完成后在边缘服务器上对多个移动终端的运算结果进行先聚合,以减少所需的全局聚合次数,极大降低通信延时,加快联邦学习的模型训练速度。

附图说明

[0028] 图1为本方法步骤图;

[0029] 图2为本方法联邦学习优化系统架构示意图。

具体实施方式

[0030] 为了能够更清楚地理解本发明的上述目的、特征和优点,下面结合附图和具体实施方式对本发明进行进一步的详细描述。需要说明的是,在不冲突的情况下,本申请的实施例及实施例中的特征可以相互组合。

[0031] 在下面的描述中阐述了很多具体细节以便于充分理解本发明,但是,本发明还可以采用其他不同于在此描述的方式来实施,因此,本发明的保护范围并不受下面公开的具体实施例的限制。

[0032] 实施例1

[0033] 如图1至图2所示,一种基于边缘计算的端-边协同联邦学习优化方法,包括以下步骤:S1:构建联邦学习优化系统架构,所述架构包括依次通信连接的云数据中心、边缘服务器、移动终端;S2:移动终端与边缘服务器协同进行云数据中心的全球网络模型的本地更新;S3:本地更新完成后,边缘服务器得到更新好的模型参数并发送至云数据中心进行全局聚合,得到新的全球网络模型;S4:云数据中心对新的全球网络模型进行模型精度测试。

[0034] 上述方案中,在所述联邦学习优化系统架构中,将传统的端-云两层架构更改为端-边-云三层架构,主要针对模型的本地更新过程进行优化,通过加入距离移动终端较近的边缘服务器进行端-边协同的本地更新过程;将模型训练过程中的部分计算以及模型参数的更新过程卸载给边缘服务器进行处理,可极大减少移动终端的计算开销,避免了移动终端进行长时间计算占用存储资源;另外,在协调的本地更新时,便在边缘服务器上对多个移动终端的运算结果进行聚合,获得较好的模型精度,减少了边缘服务器上传模型参数至云数据中心进行聚合的次数,极大降低了全局通信过程中的通信延时。对于本申请中的联邦学习优化系统架构,以1个云数据中心、M个边缘服务器、第m个边缘服务器区域内有 K_m 个移动终端为例进行说明。

[0035] 具体地,步骤S1具体过程为:S101:所述云数据中心将原始模型参数发送至边缘服务器;S102:所述边缘服务器将原始模型参数保存至本地并发送至区域内的移动终端。

[0036] 上述方案中,云数据中心与M个边缘服务器通信连接,第m个边缘服务器与其区域内的 K_m 个移动终端通信连接;当移动终端获取到原始模型参数后,结合自身的本地样本数据进行更新。

[0037] 具体地,步骤S2的具体过程为:S201:将云数据中心的全局网络模型分割为浅层和剩余层;S202:移动终端执行浅层函数计算,得到浅层计算结果,并将浅层计算结果以及对应的数据标签发送至边缘服务器进行聚合,得到浅层输出值矩阵;S203:边缘服务器将浅层输出值矩阵作为剩余层函数输入并执行计算,得到剩余层计算结果,然后进行模型参数更新。

[0038] 上述方案中,移动终端与边缘服务器进行协同的本地更新,其过程具体如下列算法1:

算法 1: 端-边协同本地更新算法

M 为边缘服务器个数; m 为边缘服务器下标; K 为移动终端个数; k 为移动终端下标; B 为本地 batch 大小; E 为本地更新 epoch 次数; μ 为学习率。

1 输入: k 个移动终端的本地样本数据

2 输出: 更新后的模型参数

3 // 在边缘服务器上运行

4 **EdgeUpdate()**:

5 **foreach** epoch e from 1 to E **do**:

6 **foreach** batch $b \in \beta$ **do**:

7 **for** 每个客户端 k **in parallel do**:

8 $x_{conv}^k, y_{label}^k \leftarrow$ Received from client k

9 **end**

10 $x_{conv} \leftarrow [x_{conv}^1, x_{conv}^2, \dots, x_{conv}^k]$

11 $y_{label} \leftarrow [y_{label}^1, y_{label}^2, \dots, y_{label}^k]$

12 $x_{output} = E(x_{conv})$

13 $\omega \leftarrow \omega - \mu \nabla \text{loss}(\omega; x_{output}, y_{label})$

14 **end**

15 **end**

16 // 将 ω 发送至云服务器端

[0039]

```

17     return  $\omega$ 
18 End
19
20 // 客户端执行浅层网络计算
21 ClientCompute( $k, \omega$ ):
[0040] | 22     foreach epoch  $i$  from 1 to  $E$  do:
| 23         foreach batch  $b \in \beta$  do:
| 23              $x_{conv} \leftarrow \mathbf{Conv1AndPoolingLayer}(\omega; b)$ 
| 24             Send $x_{conv}, y_{label}$  to EdgeServer
|                                     end
25     end
26 End

```

[0041] 在算法1中第21至26行,移动终端获取原始模型参数后,将自身的本地样本数据集分为多个大小相同的batch,执行批量梯度更新和随机梯度更新;在处理每个batch时,当移动终端执行完浅层计算,都会将运算结果以及对应的数据标签发送至边缘服务器进行聚合,即在协同本地更新的过程中,每个batch的处理都需要与边缘服务器进行一次通信,显然batch的大小设置会影响移动终端与边缘服务器之间对所有本地数据完成一次训练

所需的通信次数 t ,如公式所示: $t = \frac{D_k}{S_{batch}}$ 其中 D_k 为第 k 个移动终端的本地样本数据量,

S_{batch} 为样本batch的大小;当 t 太大,移动终端与边缘服务器之间通信次数过多,尽管端-边之间网络带宽较大,但由于网络的不稳定性,过多的通信次数依然会影响端-边之间协同的本地更新过程;当 S_{batch} 设置得太小,会增加端-边之间通信次数,但 S_{batch} 太大,又会导致本地更新过程中网络模型的收敛速度变慢,因此需要合理设置每个移动终端训练过程中 S_{batch} 的大小;而且,边缘服务器需要等待所有参与本次本地更新的移动终端完成浅层网络计算,边缘服务器区域内所有移动终端与边缘服务器的通信次数 t 的大小必须相同。另外,协同的本地更新过程中,对于每个移动终端与边缘服务器所需的总通信次数 T ,如公式所

示: $T = E * \frac{D_k}{S_{batch}}$ 其中 E 为本地的epoch次数;当 E 太大,会导致端-边之间总通信次数增多,

因而需要综合考量 E 和 S_{batch} 的值,判断当前端-边之间的网络状况,在可接受的通信次数的前提下,合理减小 S_{batch} 并增大 E ,以得到较好的模型精度。

[0042] 具体地,步骤S201中所述模型为CNN网络模型。

[0043] 上述方案中,在进行协同的本地更新过程之前,先对CNN网络模型进行分割,选择合适的分割点,使发送至边缘服务器的数据量较小,保证移动终端与边缘服务器通信延时最小。

[0044] 具体地,步骤S202中移动终端执行浅层函数计算,其计算公式具体为: $C(x) = x_{conv}$ ①;其中, x 为移动终端本地样本数据, x_{conv} 为浅层计算结果。

[0045] 上述方案中,模型分割后,移动终端负责执行浅层函数计算。

[0046] 具体地,步骤S202中聚合得到浅层输出值矩阵,其计算公式具体为:

$$[0047] \quad x_{conv} \leftarrow [x_{conv}^1, x_{conv}^2, x_{conv}^3, \dots, x_{conv}^k] \text{ ②};$$

$$[0048] \quad y_{label} \leftarrow [y_{label}^1, y_{label}^2, y_{label}^3, \dots, y_{label}^k] \text{ ③};$$

[0049] 其中, x_{conv} 为浅层计算结果, x_{conv}^k 为第k个移动终端的浅层计算结果, y_{label} 为样本数据标签, y_{label}^k 为第k个移动终端的浅层计算结果对应的样本数据标签。

[0050] 上述方案中,实际应用中,由于用户使用习惯的不同,移动终端的本地样本数据量有限,且每个移动终端的本地样本数据集是非独立同分布的;所以,将边缘服务器上获取到的多个移动终端的浅层计算结果以及对应的样本数据标签进行聚合,将数据集之间的样本特征也聚合到了一起,使得边缘服务器计算出的梯度下降值的梯度下降方向更加准确,加快了训练过程的模型收敛速度,解决了单个移动终端上本地样本数据集较小、样本数据集之间非独立同分布的问题。

[0051] 具体地,步骤S203中边缘服务器执行剩余层函数计算,其公式具体为: $E(x_{conv}) = x_{couput}$ ④;其中, x_{couput} 为剩余层计算结果。

[0052] 上述方案中,模型分割后,边缘服务器负责执行剩余层函数计算。

[0053] 具体地,步骤S203中进行模型参数更新,其具体公式为:

$$[0054] \quad \omega = \omega - \mu \nabla loss(x_{output}, y_{label}) \text{ ⑤};$$

[0055] 其中, ω 为模型参数值, μ 为梯度下降的步长, x_{output} 为剩余层计算结果, y_{label} 为样本数据标签。

[0056] 上述方案中,当边缘服务器完成模型参数更新后,会将更新后的模型参数返回移动终端,继续进行下一次小批量样本数据的模型训练更新。

[0057] 具体地,步骤S3中更新好的模型参数在云数据中心进行全局聚合,其具体公式为:

$$[0058] \quad \omega_{t+1}^i \leftarrow \sum_{m=1}^M \frac{n_m}{n} \omega_t^i \text{ ⑥};$$

[0059] 其中, ω_{t+1}^i 为更新后的模型参数进行全局聚合得到的值; n 为所有参与协同本地更新的移动终端所使用的本地样本数据量总和, n_m 为第m个边缘服务器区域内参与协同本地更新的移动终端所使用的样本数据量。

[0060] 上述方案中,在边缘服务器与移动终端协同进行多次本地更新过程后,M个边缘服务器会各自将模型参数发送至云数据中心进行聚合;云数据中心根据每个边缘服务器区域下的所有移动终端的样本数据量对模型参数进行加权平均求和,得到新的全局模型,其具体过程如算法2所示:

算法 2：全局聚合算法

M 为边缘服务器个数； m 为边缘服务器下标； K 为移动终端个数； k 为移动终端下标； B 为本地 batch 大小； E 为本地更新 epoch 次数； μ 为学习率。

```

1 输入：每个边缘服务器上传的模型参数
2 输出：全局聚合后的全局模型参数
3 // 在云服务器运行
4 GlobalAggregation():
[0061] 5   foreach round  $t = 1, 2, \dots$  do:
6       for 每个边缘服务器  $m$  in parallel do:
7            $\omega_{t+1}^k \leftarrow \text{EdgeUpdate}(k, \omega_t)$ 
8       end
9           // 每台边缘服务器上传本地更新后的模型参数  $\omega_{t+1}^k$  进行全局
           聚合
10           $\omega_{t+1} \leftarrow \sum_{k=1}^K \frac{n}{n_k} \omega_{t+1}^k$ 
11      end
[0062] 12 End

```

[0063] 具体地，步骤S4中模型精度测试的具体过程为：S401：采用新的模型对云 数据中心保存的测试样本数据进行预测；S402：若预测结果达到预设模型精度指 标，则模型训练 结束；若预测结果未达到预设模型精度指标，则云数据中心将模 型参数下发，继续进行模 型训练。

[0064] 上述方案中，对模型精度进行测试，若已经达标，则不需再进行训练，以避 免导致 过拟合问题；若未达标，则继续进行训练。

[0065] 本发明通过建立联邦学习优化系统架构，分割模型训练过程，将部分复杂的 计算 和模型参数的更新卸载至距离移动终端较近的边缘服务器执行，极大减少了 移动终端的 计算开销，避免移动终端长时间计算占用存储资源；并且在边缘服务 器上对多个移动终端 的运算结果进行先聚合，减少所需的全局聚合次数，极大降 低通信延时，加快联邦学习的 模型训练速度。

[0066] 相同或相似的标号对应相同或相似的部件；

[0067] 附图中描述位置关系的用语仅用于示例性说明，不能理解为对本专利的限制；

[0068] 显然，本发明的上述实施例仅仅是为清楚地说明本发明所作的举例，而并非 是对 本发明的实施方式的限定。对于所属领域的普通技术人员来说，在上述说明 的基础上还可

以做出其它不同形式的变化或变动。这里无需也无法对所有的实施方式予以穷举。凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明权利要求的保护范围之内。

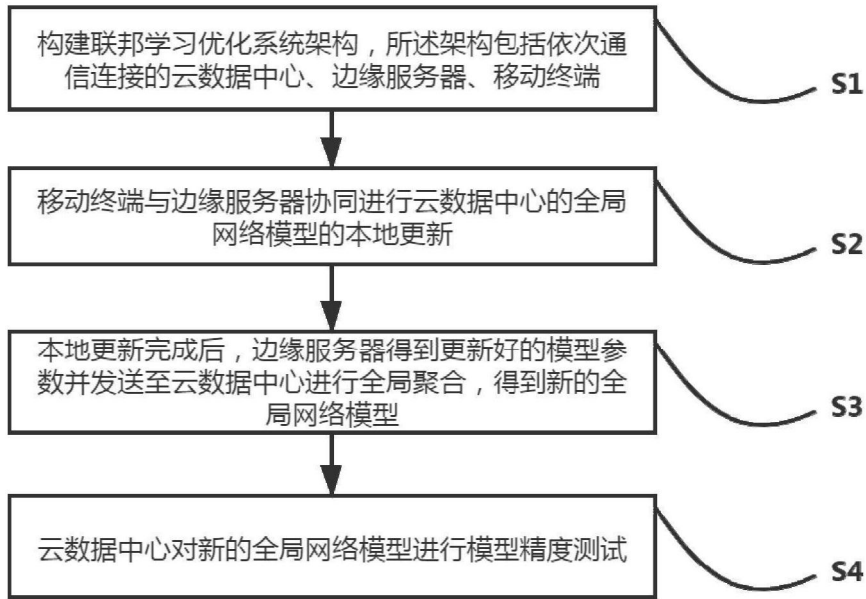


图1

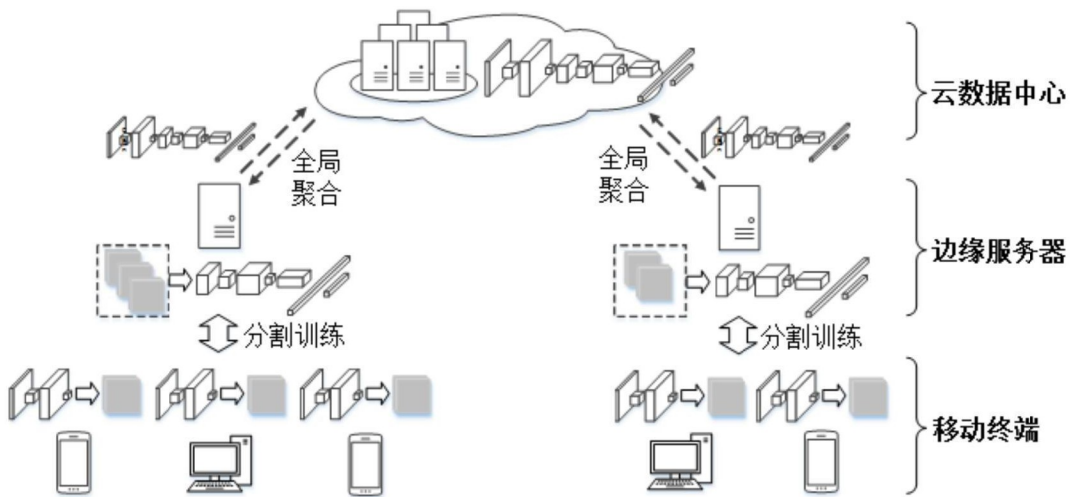


图2