(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 112154462 A
(43)申请公布日 2020.12.29

(21)申请号 201980033991.4
(22)申请日 2019.05.07
(30)优先权数据
62/675,497 2018.05.23 US
16/024,369 2018.06.29 US
(85)PCT国际申请进入国家阶段日
2020.11.19
(86)PCT国际申请的申请数据
PCT/US2019/030988 2019.05.07
(87)PCT国际申请的公布数据
WO2019/226324 EN 2019.11.28
(71)申请人 微软技术许可有限责任公司
地址 美国华盛顿州
(72)发明人 V·塞沙德利 A·费尼沙耶 D·纳拉亚南 A·哈拉普 N·D·兰格拉詹
(74)专利代理机构 北京市金杜律师事务所 11256
代理人 黄倩
(51)Int.Cl.
G06N 3/063(2006.01)
G06N 3/08(2006.01)

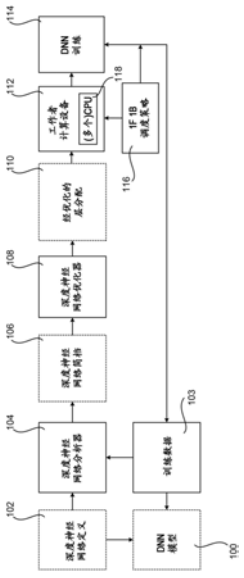
权利要求书2页 说明书11页 附图9页

(54)发明名称

高性能流水线并行深度神经网络训练

(57)摘要

使用DNN的简档将深度神经网络(DNN)的层划分为多个阶段。阶段中的每个阶段包括DNN的层中的一个或多个层。将DNN的层划分为多个阶段以各种方式被优化,包括优化划分以最小化训练时间,最小化用于训练DNN的工作者计算设备之间的数据通信,或确保工作者计算设备执行大致相等量的处理来训练DNN。阶段被分配给工作者计算设备。工作者计算设备使用调度策略来处理训练数据的批次,该调度策略使得工作者在DNN训练数据的批次的前向处理与DNN训练数据的批次的后向处理之间交替。这些阶段可以被配置成用于模型并行处理或数据并行处理。



CN 112154462 A

1.一种用于并行训练DNN模型的计算机实现的方法,包括:

生成深度神经网络(DNN)模型的简档,所述DNN模型包括多个层;

基于所述简档将所述DNN模型的所述层划分为多个阶段,其中所述多个阶段中的每个阶段包括所述DNN模型的所述层中的一个或多个层,并且其中所述划分被优化,以最小化训练所述DNN模型的时间;以及

使多个计算设备训练所述DNN模型。

2.根据权利要求1所述的计算机实现的方法,其中所述划分还被优化,以最小化所述计算设备之间的数据通信。

3.根据权利要求1所述的计算机实现的方法,其中所述划分还被优化,使得所述多个计算设备中的每个计算设备在所述DNN模型的训练期间执行大致相同量的处理。

4.根据权利要求1所述的计算机实现的方法,其中划分所述DNN模型的所述层还包括:计算要被提供给所述多个计算设备的DNN训练数据的批次的最佳数目,以使所述多个计算设备的处理效率最大化。

5.根据权利要求1所述的计算机实现的方法,还包括将所述多个阶段中的至少一个阶段分配给多个计算设备中的每个计算设备,所述计算设备被配置成:通过在DNN训练数据的所述批次的前向处理和所述DNN训练数据的批次的后向处理之间交替,来处理所述DNN训练数据的批次以训练所述DNN模型。

6.一种计算设备,包括:

一个或多个处理器;以及

至少一个计算机存储介质,其上存储有计算机可执行指令,所述计算机可执行指令在由所述一个或多个处理器执行时,将使所述计算设备:

将深度神经网络(DNN)模型的所述层划分为多个阶段,其中所述多个阶段中的每个阶段包括所述DNN模型的所述层中的一个或多个层,并且其中所述划分被优化,以最小化训练所述DNN模型的时间;以及

将所述多个阶段中的至少一个阶段分配给多个工作者计算设备中的每个工作者计算设备,所述计算设备被配置成:通过在DNN训练数据的所述批次的前向处理和所述DNN训练数据的批次的后向处理之间交替,来处理所述DNN训练数据的批次以训练所述DNN模型。

7.根据权利要求6所述的计算设备,其中所述划分还被优化,以最小化所述工作者计算设备之间的数据通信。

8.根据权利要求6所述的计算设备,其中所述划分还被优化,使得所述多个计算设备中的每个计算设备在所述DNN模型的训练期间执行大致相同量的处理。

9.根据权利要求6所述的计算设备,其中所述多个阶段中的至少一个阶段被配置用于模型并行处理,并且其中所述多个阶段中的至少一个阶段被配置用于数据并行处理。

10.根据权利要求6所述的计算设备,其中所述至少一个计算机存储介质在其上存储有另外的计算机可执行指令,以用于:

生成所述DNN模型的简档;

基于所述简档,将所述DNN模块的所述层划分为所述多个阶段。

11.一种计算机存储介质,其上存储有计算机可执行指令,所述计算机可执行指令在由计算设备的一个或多个处理器执行时,将使所述计算设备:

将深度神经网络(DNN)模型的所述层划分为多个阶段,其中所述多个阶段中的每个阶段包括所述DNN模型的所述层中的一个或多个层,并且其中所述划分被优化,以最小化训练所述DNN模型的时间;以及

将所述多个阶段中的至少一个阶段分配给多个工作者计算设备中的每个工作者计算设备,所述计算设备被配置成:通过在DNN训练数据的批次的前向处理和所述DNN训练数据的所述批次的后向处理之间交替,来处理所述DNN训练数据的所述批次以训练所述DNN模型。

12.根据权利要求11所述的计算机存储介质,其中所述划分还被优化,以最小化所述工作者计算设备之间的数据通信。

13.根据权利要求11所述的计算机存储介质,其中所述划分还被优化,使得所述多个计算设备中的每个计算设备在所述DNN模型的训练期间执行大致相同量的处理。

14.根据权利要求11所述的计算机存储介质,其中划分所述DNN模型的所述层还包括:计算要被提供给所述多个计算设备的所述DNN训练数据的所述批次的最佳数目,以使所述多个计算设备的处理效率最大化。

15.根据权利要求11所述的计算机存储介质,其中所述计算机存储介质在其上存储有另外的计算机可执行指令,以用于:

生成所述DNN模型的简档;以及

基于所述简档,将所述DNN模块的所述层划分为所述多个阶段。

高性能流水线并行深度神经网络训练

背景技术

[0001] 在诸如人脑的生物神经系统中，深度神经网络(“DNN”)在信息处理和通信模式之后被松散建模。DNN可以被用来解决复杂的分类问题，诸如但不限于对象检测、语义标记和特征提取。结果，DNN构成很多人工智能(“AI”)应用的基础，诸如计算机视觉、语音识别和机器翻译。在很多领域，DNN可以达到或甚至超过人类的准确性。

[0002] DNN的高级性能源于它们在对大数据集使用统计学习以获取输入空间的有效表示之后，从输入数据中提取高级特征的能力。但是，DNN的优越性能以高计算复杂度为代价。诸如图形处理单元(“GPU”)的高性能通用处理器通常被用来提供很多DNN应用所需要的高水平的计算性能。

[0003] 然而，随着DNN变得越来越广泛被开发和使用，模型大小已经增加以提高效能。如今的模型具有数十到数百层，通常总共有一千万到两千万个参数。这种增长不仅给已经是时间和资源密集的DNN训练过程带来压力，而且还使得用于训练DNN的常用并行化方法崩溃。

[0004] 关于这些和其他技术挑战，提出了本文进行的公开。

发明内容

[0005] 本文公开了用于高性能流水线并行DNN模型训练的技术。所公开的DNN模型(在本文中可以被简称为“DNN”)训练系统通过将跨计算设备的训练过程的各方面流水线化，来使DNN模型的训练并行化，其中计算设备被配置成处理各范围的DNN层。除了其他技术益处之外，当训练大型DNN模型时或当有限的网络带宽引起较高的通信计算比时，所公开的流水线并行计算技术还可以消除由之前的并行化方法引起的性能影响。

[0006] 对于大型DNN模型，相对于通过使能重叠的通信和计算所进行的数据并行训练，所公开的流水线并行DNN训练技术还可以将通信开销减少多达百分之九十五(95％)。附加地，所公开的技术可以通过在流水线阶段之间划分DNN层以平衡工作和最小化通信，对模型参数进行版本控制以实现后向传递正确性，以及调度双向训练流水线的前向和后向传递，来保持GPU的生产力。

[0007] 使用以上简要描述并且在下文中更充分描述的机制，对于DNN训练，所公开的技术的实施方式在“达到目标精确度所需时间”方面已经显示出比数据并行训练快五倍。这种效率提高可以减少对各种类型的计算资源的利用，包括但不限于存储器、处理器周期、网络带宽和功率。还可以通过所公开的技术的实施方式来实现本文未具体标识的其他技术益处。

[0008] 为了实现上面简要提及的技术益处以及潜在的其他益处，所公开的技术利用流水线机制、模型并行化和数据并行化的组合。该组合在本文中被称为“流水线并行”DNN训练。为了实现流水线并行DNN训练，生成DNN模型的简档。可以通过利用DNN训练数据的子集(例如，几千个微型批次)，在少量计算设备(例如一个)上执行DNN，来生成DNN简档。

[0009] 一旦已经生成了DNN简档，就基于该简档将DNN模型的层划分到阶段。阶段中的每个阶段包括DNN模型的一个或多个层。在一些实施例中，DNN的划分被优化，以最小化将DNN

模型训练到所需准确性水平的时间。

[0010] DNN模型的划分也可以或备选地被优化，以最小化计算设备之间的数据通信，或将用于训练DNN模型的计算设备配置成在训练期间均执行大致相同量的处理。对DNN模型的层的划分还可以包括计算提供给用于训练的计算设备的DNN训练数据的批次的最佳数目，以最大化其处理效率。

[0011] 一旦DNN模型被划分为阶段，阶段就被个体地分配给将训练DNN模型的计算设备。一些阶段或所有阶段可以被配置成用于模型并行处理，并且一些阶段或所有阶段可以被配置成用于数据并行处理。

[0012] 在一些配置中，利用一次前向、一次后向(“1F1B”)调度策略来配置计算设备。1F1B调度策略将计算设备配置成在DNN训练数据的批次的前向处理和DNN训练数据的批次的后向处理之间交替。一旦以该方式配置了计算设备，它们就可以开始处理DNN训练数据以训练DNN模型。

[0013] 应当理解，上述主题可以被实现为计算机控制的装置、计算机实现的方法、计算设备或诸如计算机可读介质的制品。通过阅读以下具体实施方式并且查看相关附图，这些和各种其他特征将变得明显。

[0014] 提供本发明内容以便以简化的形式介绍下面在具体实施方式中进一步描述的所公开的技术的一些方面。本发明内容既不旨在标识所要求保护的主题的关键特征或必要特征，也不旨在用于限制所要求保护的主题的范围。此外，所要求保护的主题不限于能够解决在本公开的任何部分中指出的任何或所有缺点的实施方式。

附图说明

[0015] 图1是示出了本文公开的用于配置计算设备以实现流水线并行DNN训练的一种机制的方面的计算架构图；

[0016] 图2是示出例程的流程图，其图示了用于生成DNN的简档的说明性计算机实现的过程的方面；

[0017] 图3是示出例程的流程图，其图示了用于优化DNN的层到计算设备的分配以用于DNN的管线并行训练的说明性计算机实现的过程的方面；

[0018] 图4A和图4B是计算系统图，其示出了DNN的层到计算设备的几种示例分配，以用于DNN的管线并行训练；

[0019] 图5是工作流图，其示出了本文公开的用于将工作分配给计算设备以进行管线并行DNN训练的一次前向一次后向调度策略的方面；

[0020] 图6是示出例程的流程图，其图示了用于使用一次前向一次后向调度策略执行流水线并行DNN训练的说明性计算机实现的过程的方面；

[0021] 图7是计算机架构图，其示出了用于可以实现本文提出的技术的方面的计算设备的说明性计算机硬件和软件架构；以及

[0022] 图8是图示其中可以实现所公开技术的方面的分布式计算环境的网络图。

具体实施方式

[0023] 以下详细描述涉及用于高性能流水线并行DNN模型训练的技术。除了其他技术益

处外，当训练大型DNN模型时或当网络带宽引起较高的通信计算比时，所公开的技术还可以消除由之前的并行化技术引起的性能影响。所公开的技术还可以在流水线阶段之间划分DNN模型的层，以平衡工作和最小化网络通信，并且有效地调度双向DNN训练流水线的前向传递和后向传递。所公开技术的这些方面和其他方面可以减少对各种类型的计算资源的利用，包括但不限于存储器、处理器循环、网络带宽和功率。还可以通过所公开的技术的实施方式来实现本文未具体标识的其他技术益处。

[0024] 在描述所公开的用于流水线并行DNN训练的技术之前，将提供DNN模型、DNN模型训练和DNN模型的并行训练的几种方法的简要概述。DNN模型通常由不同类型的层的序列组成(例如，卷积层、全连接层和池化层)。通常使用标记的数据集(例如，已经被标记有描述图像中内容的图像集)来训练DNN模型。跨多个代次(epoch)训练DNN模型。在每个代次，DNN模型以多个步骤通过据集中的所有训练数据来训练。在每个步骤中，当前模型首先对训练数据的子集(其在本文中可以被称为“微型批次”或“批次”)进行预测。该步骤通常被称为“前向传递”。

[0025] 为了进行预测，来自微型批次的输入数据被馈送到DNN模型的第一层，该第一层通常被称为“输入层”。然后，通常使用经学习的参数或权重，DNN模型的每个层对其输入计算函数，以产生用于下一层的输入。最后一层(通常被称为“输出层”)的输出是类别预测。基于DNN模型预测的标签和训练数据的每个实例的实际标签，输出层计算损失或误差函数。在DNN模型的“后向传递”中，DNN模型的每个层将计算针对前一层的误差和梯度，或者向将DNN模型的预测移向期望输出的该层的权重来更新。

[0026] DNN训练的一个目标是在尽可能少的时间内获得具有所需准确性水平的DNN模型。可以用两个指标来量化该目标：统计学效率(即，达到所需准确性水平所需的代次的数目)和硬件效率(即，完成单个代次所需的时间)。达到所需准确性水平的总训练时间是这两个指标的乘积。训练DNN模型的结果是被称为“权重”或“核”的参数集。这些参数表示可以被应用于输入的转变函数，结果是分类或经语义标记的输出。

[0027] 为了在合理的时间量内训练大型模型，可以使用以下两种方法中的一种方法来跨多个GPU并行地执行训练：模型并行性或数据并行性。在使用模型并行性或模型并行处理的情况下，整个DNN模型被复制到多个配备GPU的计算设备(在本文中，可以被称为“工作者”或“工作者设备”)上，其中每个工作者处理训练数据的不同子集。在个体工作者设备上计算的权重更新被聚合，以获得反映跨所有训练数据的更新的最终权重更新。在每次聚合期间和之后，在工作者设备之间通信的数据量与DNN模型的大小成比例。

[0028] 尽管模型并行性使得能够训练非常大的模型，但是常规的模型并行性对于训练DNN模型效率低下，因为DNN训练要求：在后向传递可以确定参数更新之前，前向传递要遍历所有层。结果，常规的模型并行性可以导致计算资源的严重欠利用，因为它要么一次仅主动使用一个工作者设备(如果在各层之间进行划分)，要么不能将计算和通信重叠(如果每个层被划分)。

[0029] 在数据并行性或数据并行处理中，训练数据集被跨多个GPU划分。每个GPU维持DNN模型的完整副本，并且在其自己的训练数据划分上进行训练，同时周期性地与其他GPU同步权重。权重同步的频率会影响统计效率和硬件效率。权重在每个微型批次处理结束时的同步(可以被称为批量同步并行或“BSP”)减少了训练中的陈旧程度，从而确保了统计效率。但

是,BSP要求每个GPU等待来自其他GPU的梯度,从而大大降低了硬件效率。

[0030] 尽管数据并行DNN训练可以很好地与具有高计算通信比的一些DNN模型一起工作,但是两个趋势威胁了其有效性。首先,不断增长的DNN模型大小会增加每个聚合网络的通信。实际上,一些当前的DNN模型足够大,以至于数据通信开销超过了GPU的计算时间,这限制了扩展,并且几乎占据了整个DNN训练时间。其次,GPU计算能力的快速提高进一步将DNN训练的瓶颈转移到所有类型DNN模型的数据通信。本文公开的技术解决了这些以及潜在的其他考虑。

[0031] 现在参考附图(其中贯穿几个附图,相同的附图标记表示相同的元件),将描述用于高性能流水线并行DNN训练的各种技术的方面。在下面的详细描述中,对形成其一部分的附图进行参考,并且通过说明的方式示出了特定的配置或示例。

[0032] 图1是计算架构图,其示出了本文公开的用于流水线并行DNN训练的系统的方面。如将在下面更详细地讨论的,图1中图示的系统划分DNN,并且将DNN的层的子集分配给不同的工作者设备以进行训练。该系统还将对训练数据的微型批次的处理流水线化,在之前的DNN层的处理完成前,将多个微型批次注入处理第一个DNN层的工作者设备。这可以保持处理流水线满载,并且确保在工作者设备上的并发处理(即每个工作者在任何特定时间点处理不同的微型批次)。所公开的系统还将数据并行性用于所选择的层的子集,以平衡工作者设备中间的计算负载。流水线化、模型并行性和数据并行性的这种组合在本文中被称为“流水线并行”DNN训练。

[0033] 如图1中所示,在描述DNN模型100的架构的一些配置中利用DNN定义102。例如但不限于,DNN定义102可以包括描述DNN模型100的架构的数据,包括但不限于:DNN模型100的层、DNN模型100的层的配置,以及将要用于训练DNN模型100的训练数据103的类型和数据量。DNN定义102被提供给处于一种配置的DNN分析器。

[0034] DNN分析器104是软件或硬件组件,其确定DNN模型100的层在用于训练DNN模型100的工作者计算设备112之间的最佳划分。DNN分析器104的输入是DNN定义102、将要由DNN模型100使用的训练数据103,以及标识将用于训练DNN模型100的工作者计算设备112的数目的数据。通过在工作者计算设备112的子集(例如,单个工作者计算设备112)上简短地训练DNN模型100并且在一些配置中观察DNN模型100的性能特性,DNN分析器104确定DNN模型100的层的最佳划分。DNN分析器104使用来自训练数据103的微型批次的子集(例如,1000个微型批次)来训练DNN模型100。DNN分析器104输出DNN简档106,其包括描述DNN模型100的性能特性的数据。

[0035] 暂时参考图2,将描述示出例程200的流程图,其示出了用于生成DNN简档106的说明性计算机实现的过程的方面。应当理解,关于图2和其他附图描述的逻辑操作可以被实现为(1)在计算设备上运行的计算机实现的动作或程序模块的序列和/或(2)计算设备内的互连机器逻辑电路或电路模块。

[0036] 本文公开的技术的特定实施方式是取决于计算设备的性能和其他要求的选择问题。因此,本文描述的逻辑操作被不同地称为状态、操作、结构设备、动作或模块。这些状态、操作、结构设备、动作和模块可以以硬件、软件、固件、专用数字逻辑及其任何组合来实现。应当理解,可以执行比附图中所示和本文描述更多或更少的操作。也可以以与本文描述的顺序不同的顺序执行这些操作。

[0037] 例程200在操作202处开始，在此DNN分析器104针对DNN模型100中的每个层计算总计算时间。针对每个层的总计算时间被计算为该层的前向传递和后向传递所需的时间量之和。例程200然后从操作202进行到操作204，在此DNN分析器104计算DNN模型100中从每个层到下一层的输出激活的大小。这也与后向传递中的输入梯度的大小匹配。

[0038] 从操作204，例程200进行到操作206，在此DNN分析器104针对DNN模型的每个层确定权重的大小。从操作206，例程200进行到操作208，在此DNN分析器104存储DNN简档106。DNN简档106包括描述针对每个层的总计算时间、从每个层到下一层的输出激活的大小以及针对每个层的权重的大小的数据。DNN简档106可以包括描述处于其他配置的DNN模型100的性能特性的其他数据。从操作208，例程200进行到操作210，其在此结束。

[0039] 如图1中所示，在一些配置中，DNN分析器104将DNN简档106提供给DNN优化器108。DNN优化器108是利用DNN简档106来生成经优化的层分配110的软件或硬件组件。为了生成经优化的层分配110，DNN优化器108将DNN模型100划分为多个阶段。多个阶段中的每个阶段包括DNN模型100的一个或多个层。

[0040] 在一些实施例中，DNN模型100的划分被优化，以最小化将DNN模型100训练到期望准确性水平所需的时间。DNN模型100的划分也可以，或备选地被优化，以最小化计算设备112之间的数据通信，或将用于训练DNN模型100的计算设备112配置成在训练期间执行大致相同量的处理。对DNN模型100的层的划分还可以包括计算提供给用于训练的计算设备112的DNN训练数据103的批次的最佳数目，以最大化其处理效率。

[0041] 一些阶段或所有阶段可以被配置成用于模型并行处理。一些阶段或所有阶段可以被配置成用于数据并行处理。当使用数据并行处理时，可以向给定阶段分配多个工作者计算设备112，每个每个工作者计算设备在执行期间处理不同的微型批次。

[0042] 一旦DNN模型被划分为多个阶段，这些阶段就被个体地分配给将训练DNN模型100的工作者计算设备112中的GPU 118。每个阶段被映射到单独的GPU 118，GPU 118针对该阶段的所有层执行前向传递和后向传递。包含输入层的阶段在本文中可以被称为输入阶段，并且包含输出层的阶段在本文中可以被称为输出阶段。

[0043] 暂时参考图3，将描述示出例程300的流程图，其图示了用于优化DNN模型100的层到计算设备112的分配以用于DNN的管线并行训练的说明性计算机实现的过程的方面。例程300在操作302处开始，在此DNN优化器108计算DNN模型100的层到多个阶段的最佳划分。如上所述，DNN模型100的划分可以被优化，以最小化训练DNN模型100的时间，最小化计算设备112之间的数据通信，或将用于训练DNN模型100的计算设备112配置成在训练期间执行大致相同量的处理。在其他配置中，可以针对其他度量优化DNN模型100的划分。

[0044] 从操作302，例程300进行到操作304，在此DNN优化器108计算针对每个阶段的复制因子。然后，例程300从操作304进行到操作306，在此DNN优化器108计算DNN训练数据103的微型批次的最佳数目，以提供给用于训练的计算设备112来最大化其处理效率。然后，例程300从操作306进行到操作308，在此DNN优化器108存储经优化的层分配110，包括但不限于：定义DNN模型100的层的最佳划分的数据、复制因子以及DNN训练数据103的微型批次的最佳数目，以提供给用于训练的计算设备112来最大化其处理效率。然后，例程300从操作308进行到操作310，其在此结束。

[0045] 如图1中所示，在一些配置中，工作者计算设备112被配置有1F1B调度策略116。

1F1B调度策略将计算设备112配置成在DNN训练数据103的批次的前向处理和DNN训练数据103的批次的后向处理之间交替。下面将参照图5和图6描述关于1F1B调度策略的附加细节。一旦使经用优化的层分配110和1F1B调度策略116配置了计算设备，它们就可以开始处理DNN训练数据来执行DNN训练114。

[0046] 图4A和图4B是计算系统图，其示出了DNN模型100的层到计算设备112的几个示例分配，以用于DNN 100的流水线并行训练。在图4A中所示的示例配置中，DNN模型100包括七个层402A-402G，包括输入层402A和输出层402G。在该示例中，DNN优化器108已经生成了包括三个阶段404A-404C的优化的层分配110。输入阶段404A包括层402A-402C，阶段404B包括层402D，并且输出阶段404C包括层402E-402G。阶段404A被分配给工作者计算设备112A，阶段404B被分配给工作者计算设备112B，并且阶段404C被分配给工作者计算设备112C。

[0047] 在图4A中所示的示例中，使用并行模型处理来实现阶段404A-404C中的每个阶段。然而，在图4B中所示的示例中，使用数据并行处理来实现阶段404B。在该示例中，两个工作者计算设备112B和112D通过对DNN训练数据103的不同微型批次的操作来实现阶段404B。

[0048] 图5是工作流图，其示出了本文公开的用于将训练数据103的微型批次分配给计算设备112以用于流水线并行DNN训练的1F1B调度策略116的方面。如上面简要讨论的那样，1F1B调度策略116将计算设备112配置成：在DNN训练数据103的批次的前向处理和DNN训练数据103的批次的后向处理之间交替。

[0049] 与常规的单向流水线不同，DNN训练是双向的(即，前向传递后跟随有后向传递，通过相同的层以相反的顺序进行)。1F1B调度策略116在每个工作者计算设备112上交织前向和后向的微型批次处理，并且在后向传递上，将训练数据103的微型批次路由通过相同的工作者112。这有助于使所有工作者计算设备112保持繁忙而不会造成流水线停顿，同时防止过多的进行中的微型批次。

[0050] 在图5中所示的示例中，四个工作者计算设备112实现了DNN模型100的四个阶段并且处理训练数据103的微型批次。工作流图中被标记为‘A’的行对应于第一工作者112，被标记为‘B’的行对应于第二工作者112，行‘C’对应于第三工作者112，并且行‘D’对应于第四工作者112。每个行中的框中的数字指示当前正在由对应的工作者112处理的微型批次的编号。具有对角线的框指示正在以前向方向处理的微型批次，具有阴影线的框指示正在以后向方向处理的微型批次，没有线的框指示工作者112在对应的时间段期间空闲。

[0051] 在启动状态下，输入阶段允许足够数目的训练数据103的微型批次(在该示例中为四个)，以在流水线进入稳定状态时使其保持满载。这些微型批次以其方式传播到输出阶段。一旦在输出阶段完成针对第一个微型批次的前向传递，对该相同的微型批次执行后向传递，然后针对后续的微型批次，开始在执行前向传递和执行后向传递之间交替。这可以在图5中所示的工作流图的行D中看到。随着后向传递开始传播到流水线的前面的阶段，针对不同的微型批次，每个阶段开始在前向传递和后向传递之间交替。因此，在稳定状态中，每个工作者112处于忙碌，进行针对微型批次的前向传递或后向传递之一。

[0052] 在完成针对微型批次的前向传递时，每个阶段将其输出激活异步地发送到下一阶段，同时开始针对另一个微型批次执行工作。同样，在完成针对微型批次的后向工作后，每个阶段将梯度异步地发送到前一阶段，同时开始针对另一个微型批次的计算。

[0053] 当阶段在跨多个GPU复制的数据并行配置下运行时(如图4B中所示的示例中)，确

定性轮询负载平衡可以被用来跨GPU地分配来自之前阶段的工作。该确定性负载平衡确保了针对微型批次的后向工作经过微型批次在其前向工作阶段中经过的相同阶段。针对流水线中的阶段的1F1B调度策略和针对跨复制阶段的负载平衡的轮询调度策略二者都是静态策略。因此，它们可以由每个工作者112独立地执行，而无需昂贵的分布式协调。

[0054] 图6是示出例程600的流程图，其图示了用于使用1F1B调度策略116来执行流水线并行DNN训练的说明性计算机实现的过程的方面。例程600在操作602处开始，在此工作者计算设备112开始处理训练数据103的微型批次。

[0055] 然后，例程600从操作602进行到操作604，在此工作者计算设备112确定其是否已达到稳定状态。如果是，则例程600从操作604进行到操作606。如果否，则例程600进行回到操作602。在操作606处，工作者计算设备112针对训练数据103的微型批次执行前向处理。然后，例程600从操作606进行到操作608，在此工作者计算设备112针对微型批次执行后向处理。例程600然后进行到操作610，在此工作者计算设备112确定是否所有训练数据103都已经被处理。如果否，则例程返回操作606。一旦所有的训练数据103都已经被处理，则例程600从操作610进行到操作612，其在此结束。

[0056] 本文公开的技术还可以实现DNN模型100训练流水线的其他优化。特别地，在一些配置中利用权重贮藏来维持权重的多个版本，每个活跃的微型批次有一个版本。

[0057] 当执行前向工作时，每个阶段使用可用权重的最新版本来处理微型批次。在完成前向工作之后，权重的副本被存储为针对该微型批次的中间状态的一部分。当针对微型批次执行后向传递时，将使用相同版本的权重来计算权重梯度。以该方式，权重贮藏确保了：在阶段内，针对给定微型批次的前向和后向工作两者都使用相同版本的模型参数。也可以利用其他优化，包括但不限于在训练开始时预先分配所有GPU存储器以最小化动态存储器分配。

[0058] 图7是计算机架构图，其示出了用于可以实现本文提出的各种技术的计算设备的说明性计算机硬件和软件架构。特别地，可以利用图7中图示的架构来实现服务器计算机、移动电话、电子阅读器、智能电话、台式计算机、交替现实或虚拟现实(“AR/VR”)设备、平板计算机、膝上型计算机、或另一种类型的计算设备。

[0059] 虽然本文描述的主题是在执行DNN模型的并行化训练的服务器计算机的一般上下文中被提出，但本领域技术人员将认识到，可以结合其他类型的计算系统和模块来执行其他实施方式。本领域技术人员还将认识到，本文描述的主题可以与其他计算机系统配置一起被实践，其他计算机系统配置包括手持式设备、多处理器系统、基于微处理器或可编程的消费电子产品、嵌入在设备(诸如，可穿戴计算设备、汽车、家庭自动化等)中的计算或处理系统、微型计算机、大型计算机等。

[0060] 图7所示的计算机700包括一个或多个中央处理单元702(“CPU”)、系统存储器704(包括随机存取存储器706(“RAM”)和只读存储器(“ROM”)708)，以及将存储器704耦合到CPU 702的系统总线710。基本输入/输出系统(“BIOS”或“固件”)可以被存储在ROM 708中，基本输入/输出系统包含用于帮助诸如在启动期间在计算机700内的元件之间传输信息的基本例程。计算机700还包括用于存储操作系统722、应用程序和其他类型的程序的大容量存储设备712。大容量存储设备712还可以被配置成存储其他类型的程序和数据，诸如DNN定义102、DNN分析器104、DNN简档106、DNN优化器108和经优化的层分配110(图7中未示出)。

[0061] 大容量存储设备712通过连接到总线710的大容量存储控制器(图7中未示出)连接到CPU 702。大容量存储设备712及其关联的计算机可读介质为计算机700提供非易失性存储。尽管本文中包含的计算机可读介质的描述是指大容量存储设备，诸如硬盘、CD-ROM驱动器、DVD-ROM驱动器或USB存储密钥，但是本领域技术人员应当理解，计算机可读介质可以是计算机700可以访问的任何可用的计算机存储介质或通信介质。

[0062] 通信介质包括诸如载波或其他传输机制等调制数据信号中的计算机可读指令、数据结构、程序模块或其他数据，并且包括任何传递介质。术语“调制数据信号”是指具有以能够将信息编码在信号中的方式来改变或设置其一个或多个特性的信号。作为示例而非限制，通信介质包括诸如有线网络或直接有线连接等有线介质，以及诸如声学、射频、红外和其他无线介质等无线介质。以上任何内容的组合也应当被包括在计算机可读介质的范围内。

[0063] 作为示例而非限制，计算机存储介质可以包括以用于存储诸如计算机可读指令、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。例如，计算机存储介质包括但不限于RAM、ROM、EPROM、EEPROM、闪存或其他固态存储技术、CD-ROM、数字多功能磁盘(“DVD”)、HD-DVD、BLU-RAY或其他光学存储器、磁带盒、磁带、磁盘存储器或其他磁性存储设备、或者可以用于存储期望信息并且可以由计算机700访问的任何其他介质。出于权利要求的目的，短语“计算机存储介质”及其变体不包括波或信号本身或通信介质。

[0064] 根据各种配置，计算机700可以使用通过诸如网络720等网络到远程计算机的逻辑连接来在联网环境中操作。计算机700可以通过连接到总线710的网络接口单元716连接到网络720。应当理解，网络接口单元716也可以用于连接到其他类型的网络和远程计算机系统。计算机700还可以包括输入/输出控制器718，用于接收和处理来自多个其他设备(包括键盘、鼠标、触摸输入、电子笔(图7中未示出)或物理传感器，诸如视频相机)的输入。类似地，输入/输出控制器718可以向显示屏或其他类型的输出设备(在图7中也未示出)提供输出。

[0065] 应当理解，本文中描述的软件组件在被加载到CPU 702中并且被执行时，可以将CPU 702和整个计算机700从通用计算设备转换为被定制为促进本文中介绍的功能的专用计算设备。CPU 702可以由可以个体或共同地呈现任何数目的状态的任何数目的晶体管或其他分立电路元件构成。更具体地，响应于本文中公开的软件模块中包含的可执行指令，CPU 702可以作为有限状态机操作。这些计算机可执行指令可以通过指定CPU 702如何在状态之间转换来对CPU 702进行转换，从而对构成CPU 702的晶体管或其他分立硬件元件进行转换。

[0066] 对本文中提出的软件模块进行编码还可以变换本文中提出的计算机可读介质的物理结构。在本说明书的不同实现中，物理结构的特定变换取决于各种因素。这种因素的示例包括但不限于用于实现计算机可读介质的技术、计算机可读介质的特征是主要存储还是辅助存储等。例如，如果计算机可读介质被实现为基于半导体的存储器，则可以通过变换半导体存储器的物理状态来将本文中公开的软件编码在计算机可读介质上。例如，该软件可以变换构成半导体存储器的晶体管、电容器或其他分立电路元件的状态。该软件还可以转换这些组件的物理状态，以便在其上存储数据。

[0067] 作为另一示例，本文中公开的计算机存储介质可以使用磁性或光学技术来实现。在这种实施方式中，当软件被编码在其中时，本文中提出的软件可以变换磁性或光学介质的物理状态。这些变换可以包括改变给定磁性介质内的特定位置的磁性特性。这些变换还可以包括改变给定光学介质内的特定位置的物理特征或特性，以改变这些位置的光学特性。在不背离本说明书的范围和精神的情况下，物理介质的其他变换是可能的，其中提供前述示例仅是为了促进该讨论。

[0068] 鉴于以上所述，应当理解，在计算机700中发生了很多类型的物理变换以便存储和执行本文中提出的软件组件。还应当理解，图7中针对计算机700示出的架构或类似架构可以用于实现其他类型的计算设备，包括手持计算机、视频游戏设备、嵌入式计算机系统、移动设备(诸如智能手机、平板电脑和AR/VR设备)，以及本领域技术人员已知的其他类型的计算设备。还可以想到，计算机700可以并非包括图7所示的所有组件，可以包括图7中未明确示出的其他组件，或者可以使用与图7所示的架构完全不同的架构。

[0069] 图8是图示根据本文中呈现的各种配置的可以在其中实现所公开的技术的各方面的分布式网络计算环境800的网络图。如图8中所示，一个或多个服务器计算机800A可以经由通信网络720(其可以是固定有线或无线LAN、WAN、内联网、外联网、对等网络、虚拟专用网络、因特网、蓝牙通信网络、专有低压通信网络或其他通信网络)与多个客户端计算设备(诸如但不限于平板电脑800B、游戏控制台800C、智能手表800D、电话800E(诸如智能电话)、个人计算机800F和AR/VR设备800G)互连。

[0070] 例如，在通信网络720是因特网的网络环境中，服务器计算机800A可以是专用服务器计算机，该专用服务器计算机可操作以经由多种已知协议中的任何一种来处理与客户端计算设备800B-800G的数据以及与客户端计算设备800B-800G传送数据，诸如超文本传输协议(“HTTP”)、文件传输协议(“FTP”)或简单对象访问协议(“SOAP”)。另外，网络计算环境800可以利用各种数据安全协议，诸如安全套接字层(“SSL”)或相当好的隐私(“PGP”)。每个客户端计算设备800B-800G可以配备有操作系统，该操作系统可操作以支持一个或多个计算应用或终端会话，诸如网络浏览器(图8中未示出)或其他图形用户界面(图8中未示出)或移动桌面环境(图8中未示出)，以获取对服务器计算机800A的访问。

[0071] 服务器计算机800A可以通信地耦合到其他计算环境(图8中未示出)，并且接收有关参与用户的交互/资源网络的数据。在说明性操作中，用户(图8中未示出)可以与在客户端计算设备800B-800G上运行的计算应用交互以获取期望数据和/或执行其他计算应用。

[0072] 数据和/或计算应用可以存储在一个或多个服务器800A上，并且通过示例性通信网络720通过客户端计算设备800B-800G传送到合作用户。参与用户(图8中未示出)可以请求访问全部或部分容纳在服务器计算机800A上的特定数据和应用。这些数据可以在客户端计算设备800B-800G与服务器计算机800A之间传送以进行处理和存储。

[0073] 服务器计算机800A可以托管用于数据、应用的生成、认证、加密和通信的计算应用、过程和小程序，并且可以与其他服务器计算环境(图8中未示出)、第三方服务供应商(图8中未示出)、网络附加存储(“NAS”)和存储区域网络(“SAN”)协作以实现应用/数据交易。

[0074] 应当理解，图7中所示的计算架构和图8中所示的分布式网络计算环境为了便于讨论而被简化。还应当理解，计算架构和分布式计算网络可以包括和利用本文中未具体描述的更多的计算组件、设备、软件程序、网络设备和其他组件。

[0075] 本文提出的公开内容还涵盖以下实例中阐述的主题：

[0076] 示例A：一种用于并行训练DNN模型的计算机实现的方法，包括：生成深度神经网络(DNN)模型的简档，所述DNN模型包括多个层；基于所述简档将所述DNN模型的所述层划分为多个阶段，其中所述多个阶段中的每个阶段包括所述DNN模型的所述层中的一个或多个层，并且其中所述划分被优化，以最小化训练所述DNN模型的时间；以及使多个计算设备训练所述DNN模型。

[0077] 示例B：根据示例A所述的计算机实现的方法，其中所述划分还被优化，以最小化所述计算设备之间的数据通信。

[0078] 示例C：根据示例A-B中任一项所述的计算机实现的方法，其中所述划分被进一步优化，以使所述多个计算设备中的每个计算设备在所述DNN模型的训练期间执行大致相同量的处理。

[0079] 示例D：根据示例A-C中任一项所述的计算机实现的方法，其中对所述DNN模型的所述层的划分还包括：计算要被提供给所述多个计算设备的DNN训练数据的批次的最佳数目，以使所述多个计算设备的处理效率最大化。

[0080] 示例E：根据示例A-D中任一项所述的计算机实现的方法，还包括将所述多个阶段中的至少一个阶段分配给多个计算设备中的每个计算设备，所述计算设备被配置成：通过在DNN训练数据的所述批次的前向处理和所述DNN训练数据的批次的后向处理之间交替，来处理DNN所述训练数据的批次以训练所述DNN模型。

[0081] 示例F：根据示例A-E中任一项所述的计算机实现的方法，其中所述多个阶段中的至少一个阶段被配置成用于模型并行处理。

[0082] 示例G：根据示例A-F中任一项所述的计算机实现的方法，其中所述多个阶段中的至少一个阶段被配置成用于数据并行处理。

[0083] 示例H：根据示例A-G中任一项所述的计算机实现的方法，其中所述多个阶段中的至少一个阶段被配置成用于模型并行处理，并且其中所述多个阶段中的至少一个阶段被配置成用于数据并行处理。

[0084] 示例I：一种计算设备，包括：一个或多个处理器；以及至少一个计算机存储介质，其上存储有计算机可执行指令，所述计算机可执行指令在由所述一个或多个处理器执行时，将使得所述计算设备：将DNN模型的所述层划分为多个阶段，其中所述多个阶段中的每个阶段包括所述DNN模型的所述层中的一个或多个层，并且其中所述划分被优化，以最小化训练所述DNN模型的时间；以及将所述多个阶段中的至少一个阶段分配给多个工作者计算设备中的每个工作者计算设备，所述计算设备被配置成：通过在DNN训练数据的所述批次的前向处理和所述DNN训练数据的批次的后向处理之间交替，来处理所述DNN训练数据的批次以训练所述DNN模型。

[0085] 示例J：根据示例I所述的计算设备，其中所述划分还被优化，以最小化所述工作者计算设备之间的数据通信。

[0086] 示例K：根据示例I-J中任一项所述的计算设备，其中所述划分还被优化，以使所述多个计算设备中的每个计算设备在所述DNN模型的训练期间执行大致相同量的处理。

[0087] 示例L：根据示例I-K中任一项所述的计算设备，其中所述多个阶段中的至少一个阶段被配置成用于模型并行处理，并且其中所述多个阶段中的至少一个阶段被配置成用于

数据并行处理。

[0088] 示例M:根据示例I-L中任一项所述的计算设备,其中所述至少一个计算机存储介质在其上存储有另外的计算机可执行指令,以用于:生成所述深度神经网络(DNN)模型的简档;基于所述简档,将所述DNN模块的所述层划分为所述多个阶段。

[0089] 示例N:根据示例I-M中任一项所述的计算设备,其中通过利用所述DNN训练数据的子集,在所述多个工作者计算设备的子集上训练所述DNN模型预先确定的时间段,来生成所述DNN模块的所述简档。

[0090] 示例O:一种计算机存储介质,其上存储有计算机可执行指令,所述计算机可执行指令在由计算设备的一个或多个处理器执行时,将使得所述计算设备:将深度神经网络(DNN)模型的所述层划分为多个阶段,其中所述多个阶段中的每个阶段包括所述DNN模型的所述层中的一个或多个层,并且其中所述划分被优化,以最小化训练所述DNN模型的时间;以及将所述多个阶段中的至少一个阶段分配给多个工作者计算设备中的每个工作者计算设备,所述计算设备被配置成:通过在DNN训练数据的所述批次的前向处理和所述DNN训练数据的所述批次的后向处理之间交替,来处理所述DNN训练数据的批次以训练所述DNN模型。

[0091] 示例P:根据示例O所述的计算机存储介质,其中所述划分还被优化,以最小化所述工作者计算设备之间的数据通信。

[0092] 示例Q:根据示例O-P中任一项所述的计算机存储介质,其中所述划分还被优化,以使所述多个计算设备中的每个计算设备在训练所述DNN模型期间执行大致相同量的处理。

[0093] 示例R:根据示例O-Q中任一项所述的计算机存储介质,其中对所述DNN模型的所述层的划分还包括:计算要被提供给所述多个计算设备的所述DNN训练数据的所述批次的最佳数目,以使所述多个计算设备的处理效率最大化。

[0094] 示例S:根据示例O-R中任一项所述的计算机存储介质,其中所述计算机存储介质在其上存储有另外的计算机可执行指令,以用于:生成所述DNN模型的简档;以及基于所述简档,将所述DNN模块的所述层划分为所述多个阶段。

[0095] 示例T:根据示例O-S中任一项所述的计算机存储介质,其中通过利用所述DNN训练数据的子集,在所述多个工作者计算设备的子集上训练所述DNN模型预先确定的时间段,来生成所述DNN模块的所述简档。

[0096] 基于前述内容,应当理解,本文中已经公开了用于高性能流水线并行DNN训练的技术。尽管已经以计算机结构特征、方法和转换动作、特定的计算机器和计算机可读介质专用的语言描述了本文中提出的主题,但是应当理解,所附权利要求中阐述的主题不必限于本文中描述的特定特征、动作或介质。相反,特定特征、动作和介质被公开作为实现所要求保护的主题的示例形式。

[0097] 上述主题仅以说明的方式被提供,并且不应当被解释为是限制性的。可以在不遵循示出和描述的示例配置和应用的情况下,并且在不脱离在所附权利要求中阐述的本公开的范围的情况下,对本文中描述的主题进行各种修改和改变。

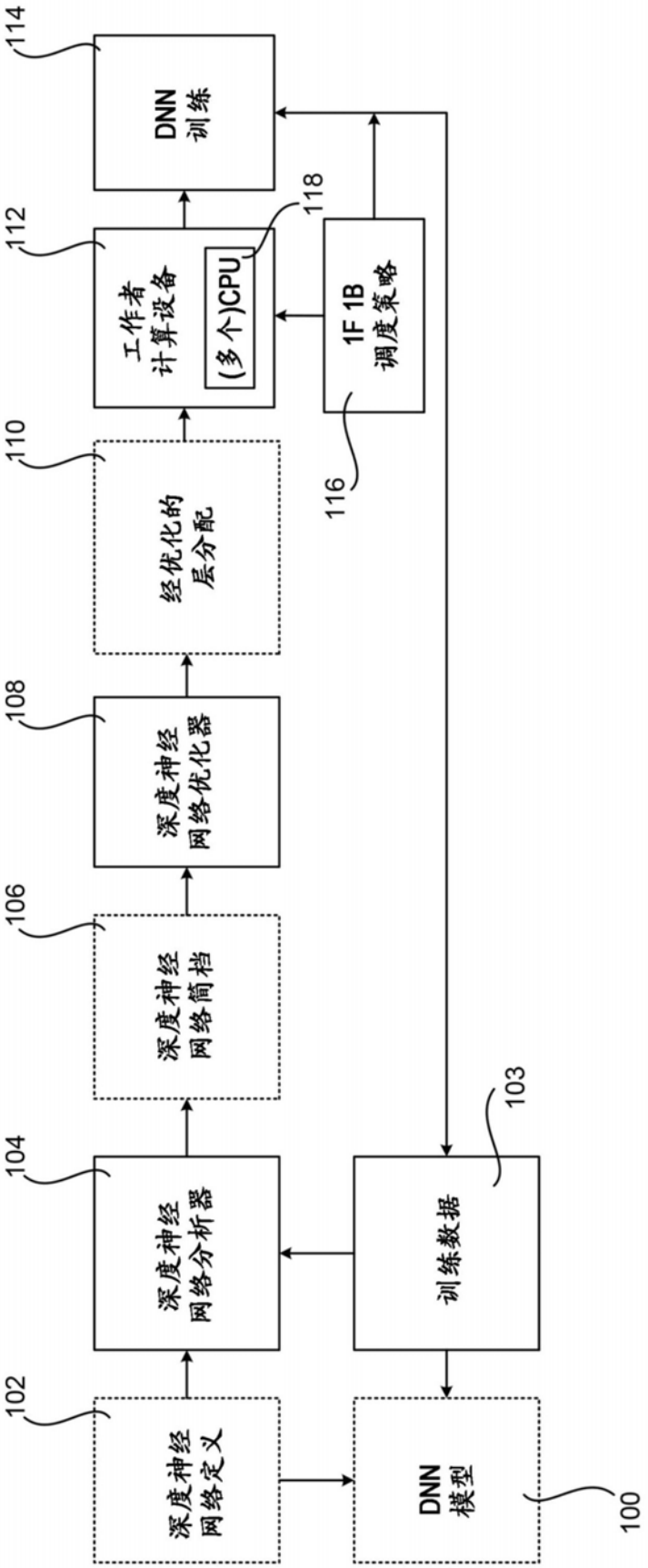

DNN
训练
114
工作者
计算设备
(多个)CPU
112
118
1F 1B
调度策略
116
经优化的
层分配
110
深度神经
网络优化器
108
深度神经
网络简档
106
深度神经
网络分析器
104
训练数据
103
深度神经
网络定义
102
DNN
模型
100

图1

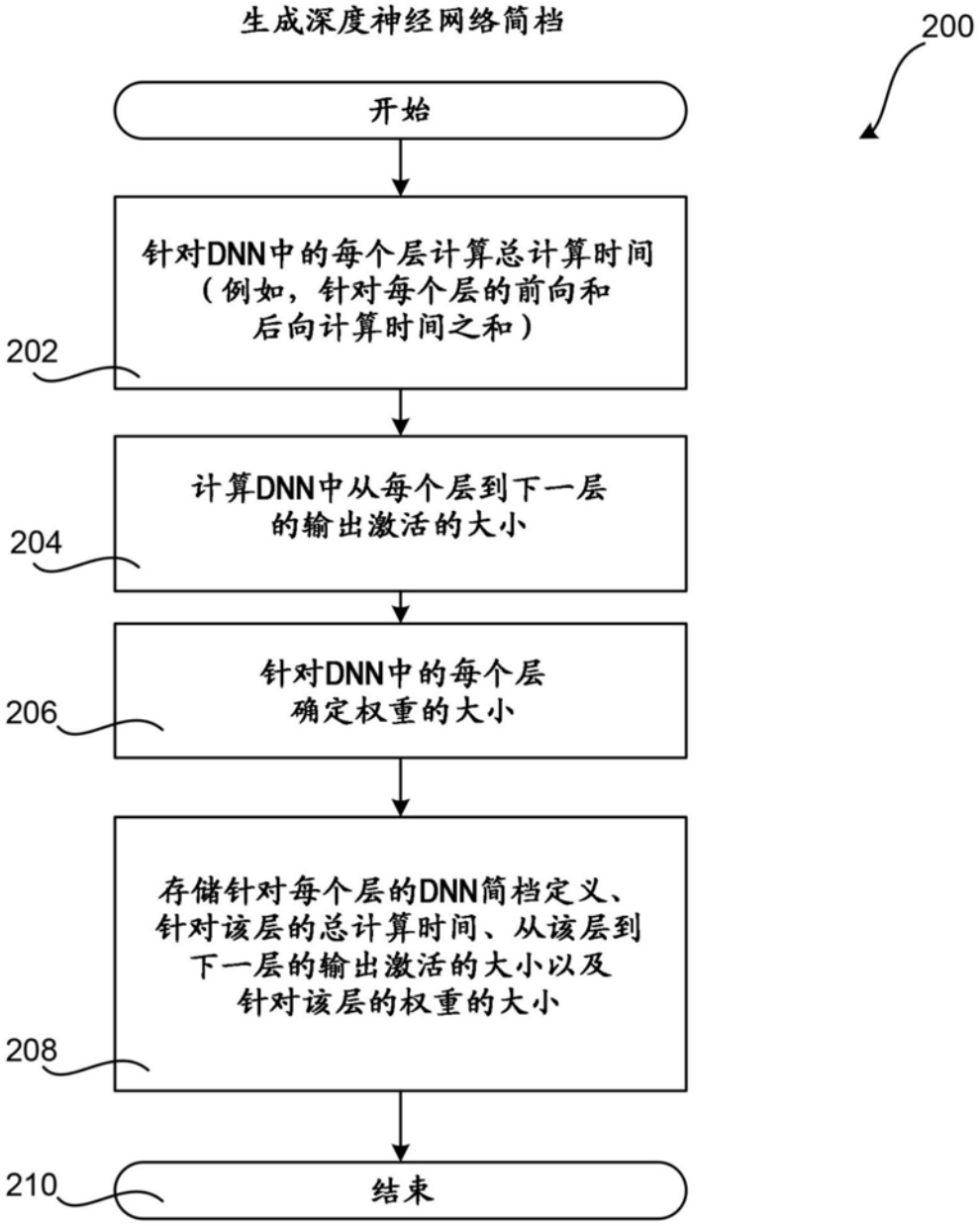

生成深度神经网络简档
200
开始
针对DNN中的每个层计算总计算时间
（例如，针对每个层的前向和
后向计算时间之和）
202
计算DNN中从每个层到下一层
的输出激活的大小
204
针对DNN中的每个层
确定权重的大小
206
存储针对每个层的DNN简档定义、
针对该层的总计算时间、从该层到
下一层的输出激活的大小以及
针对该层的权重的大小
208
结束
210

图2

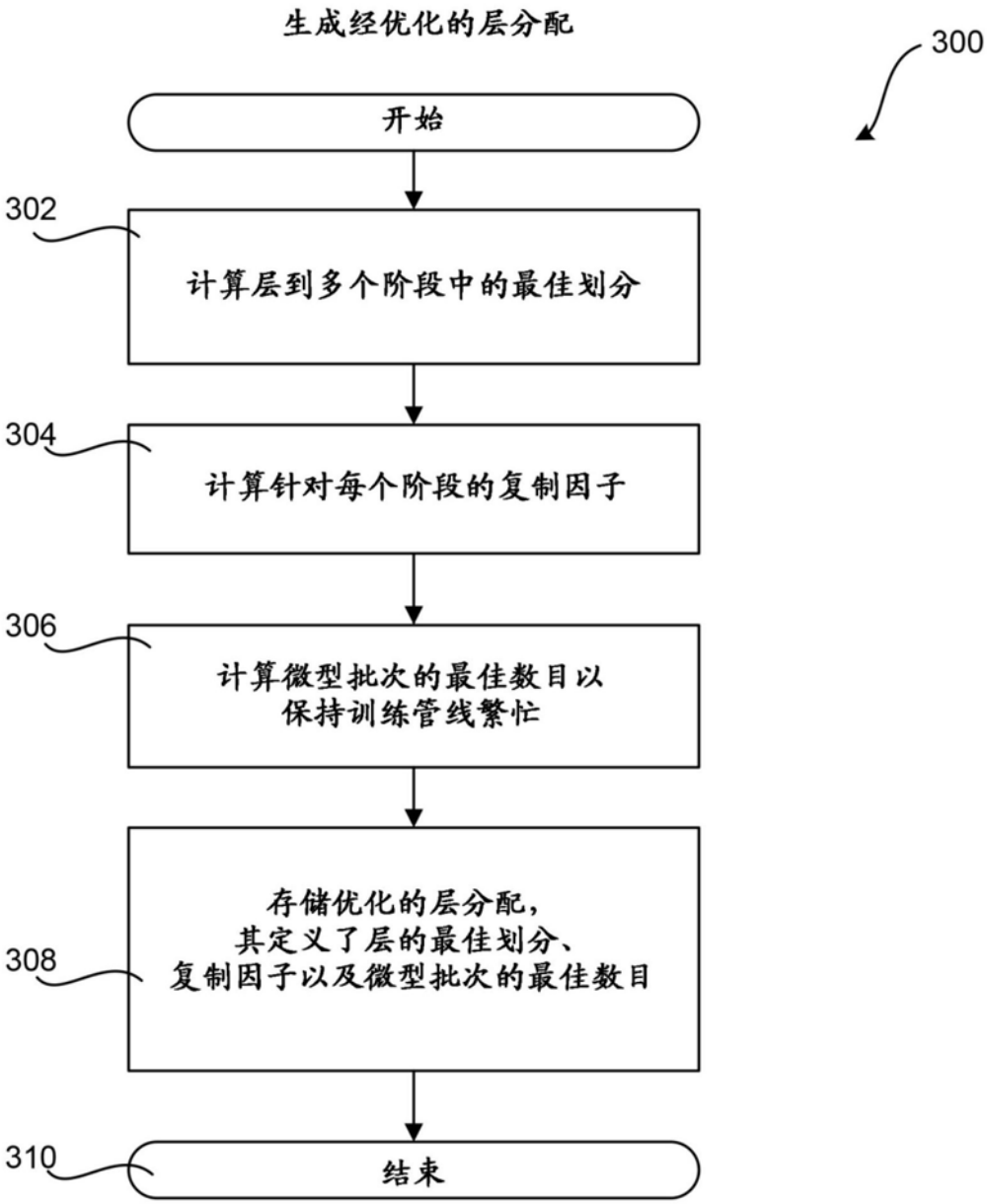

生成经优化的层分配
300
开始
302
计算层到多个阶段中的最佳划分
304
计算针对每个阶段的复制因子
306
计算微型批次的最佳数目以
保持训练管线繁忙
存储优化的层分配，
其定义了层的最佳划分、
复制因子以及微型批次的最佳数目
308
310
结束

图3

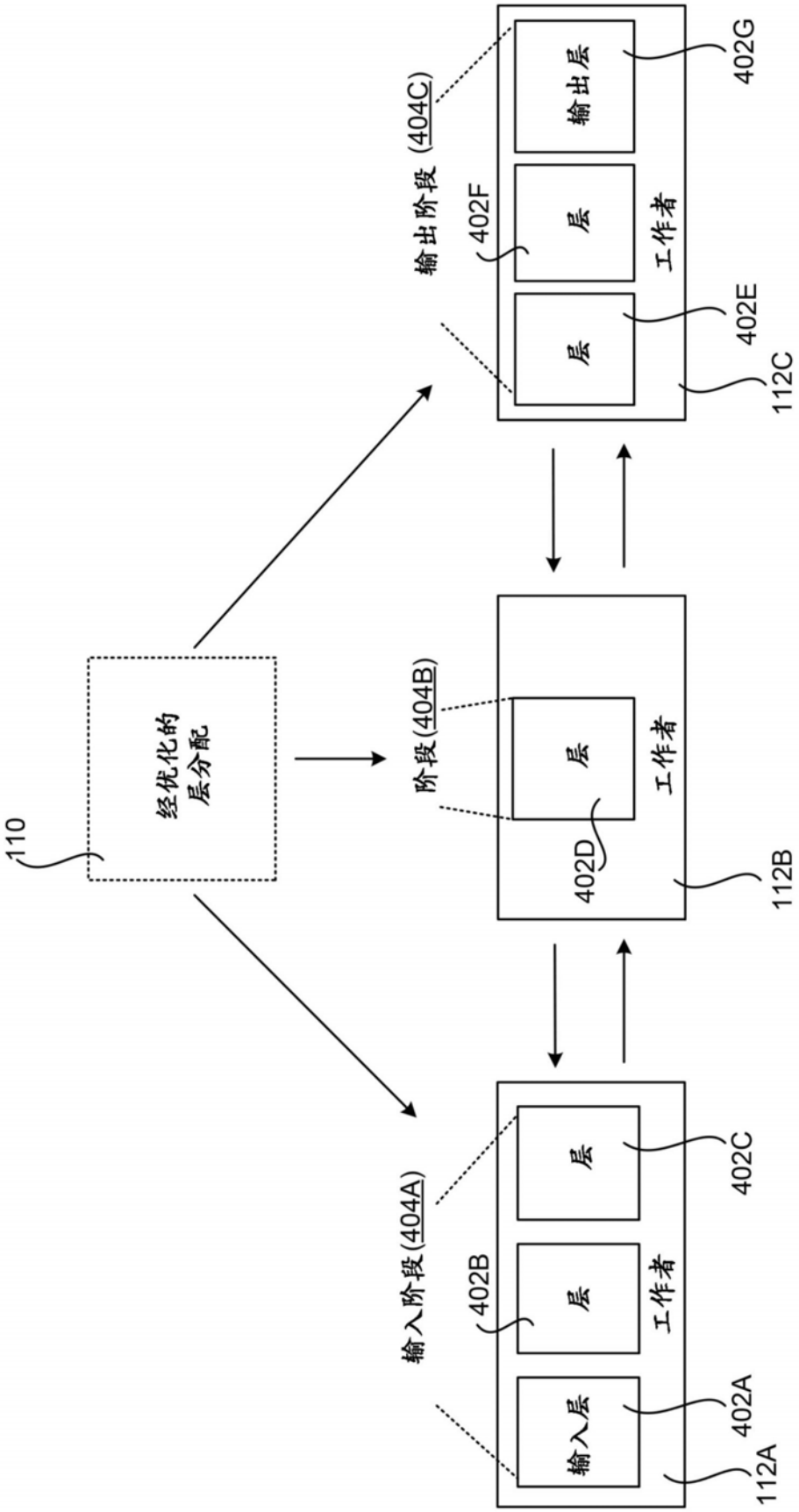

经优化的
层分配
110
输入阶段(404A)
输入层
层
层
工作者
402A
402B
402C
112A
阶段(404B)
层
工作者
402D
112B
输出阶段(404C)
层
层
工作者
402E
402F
402G
112C

图4A

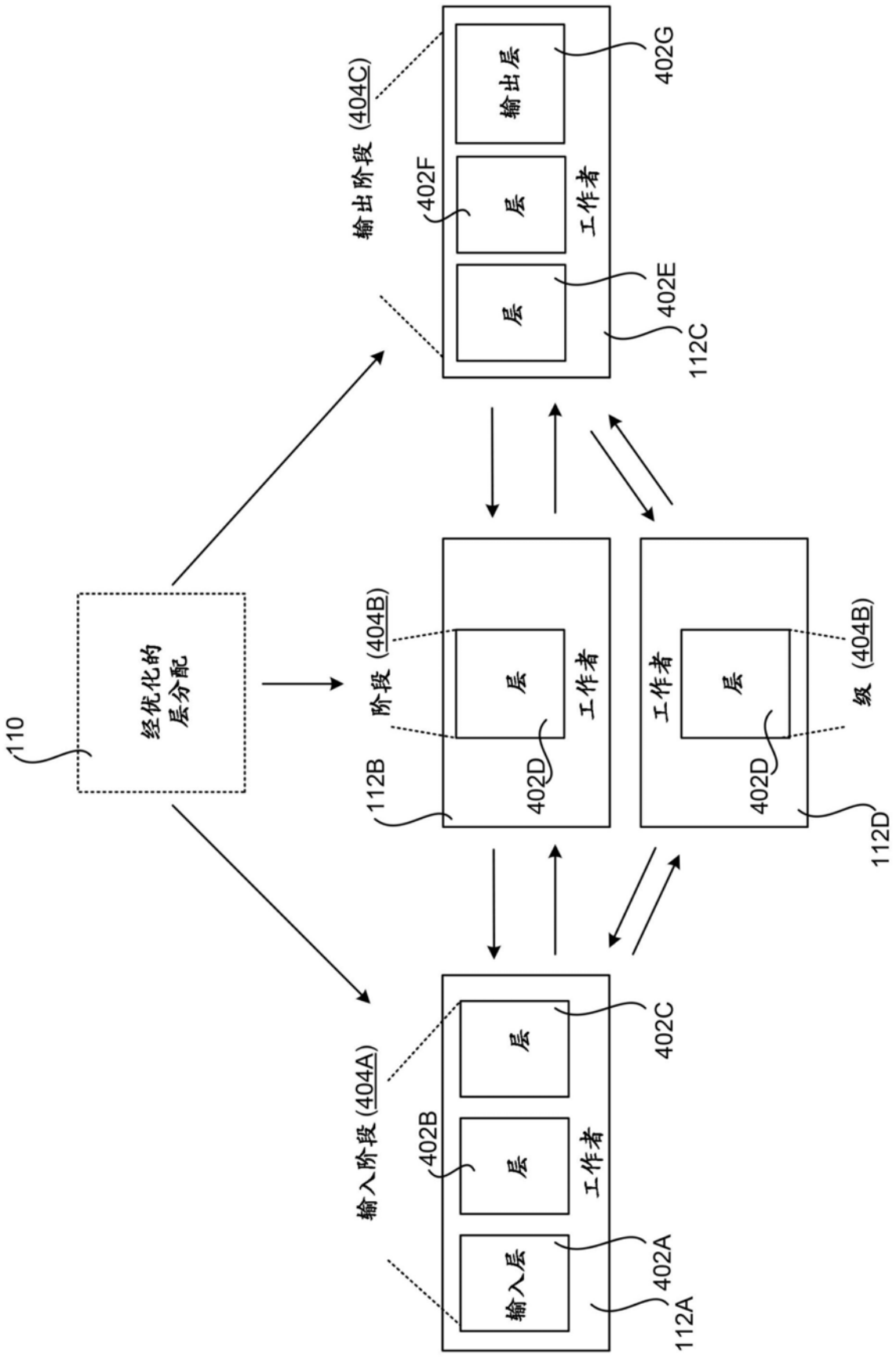

110
经优化的
层分配
输入阶段 (404A)
输入层
层
层
工作者
402A
402B
402C
112A
阶段 (404B)
层
工作者
402D
112B
级 (404B)
工作者
层
402D
112D
输出阶段 (404C)
输出层
层
工作者
402E
402F
402G
112C

图4B

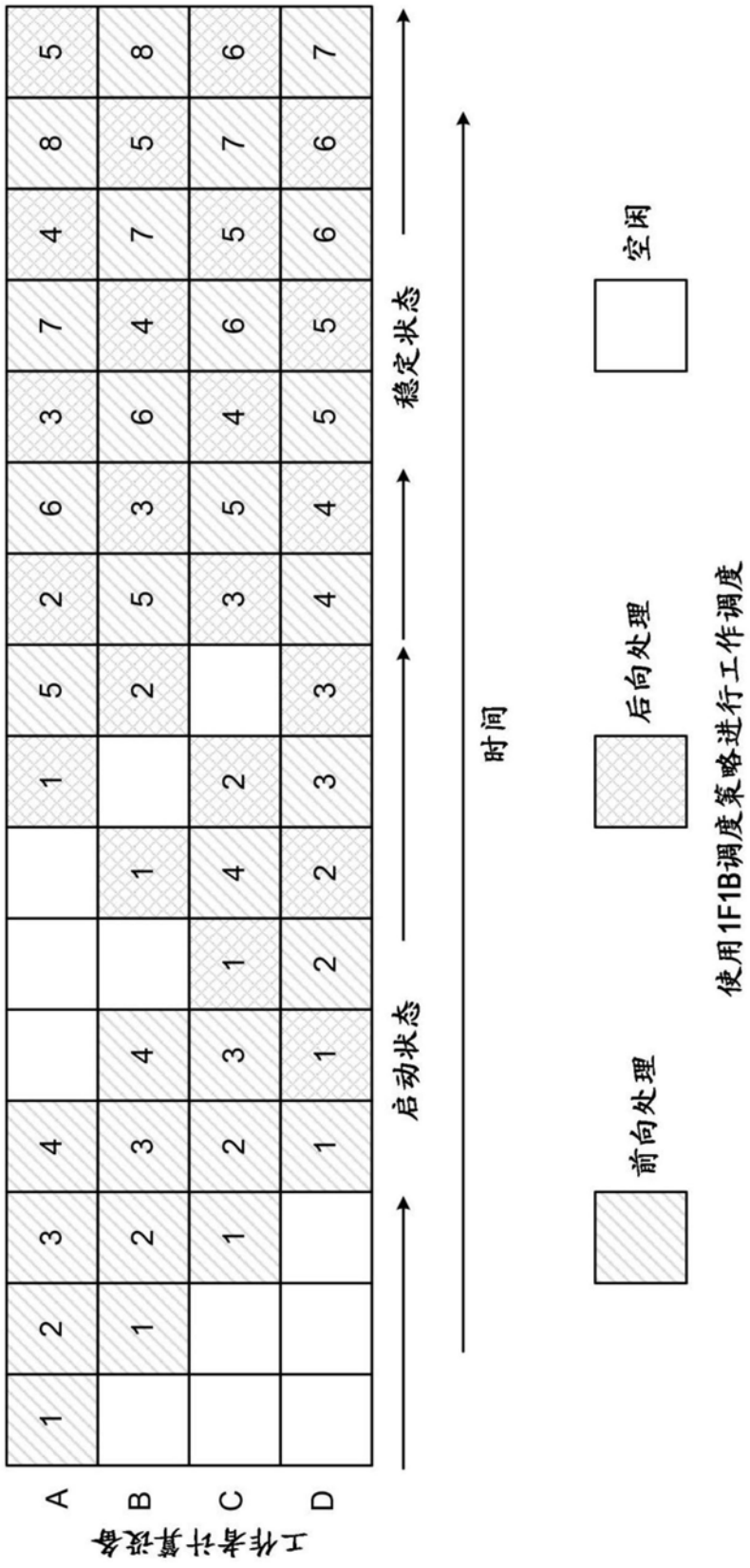


图5

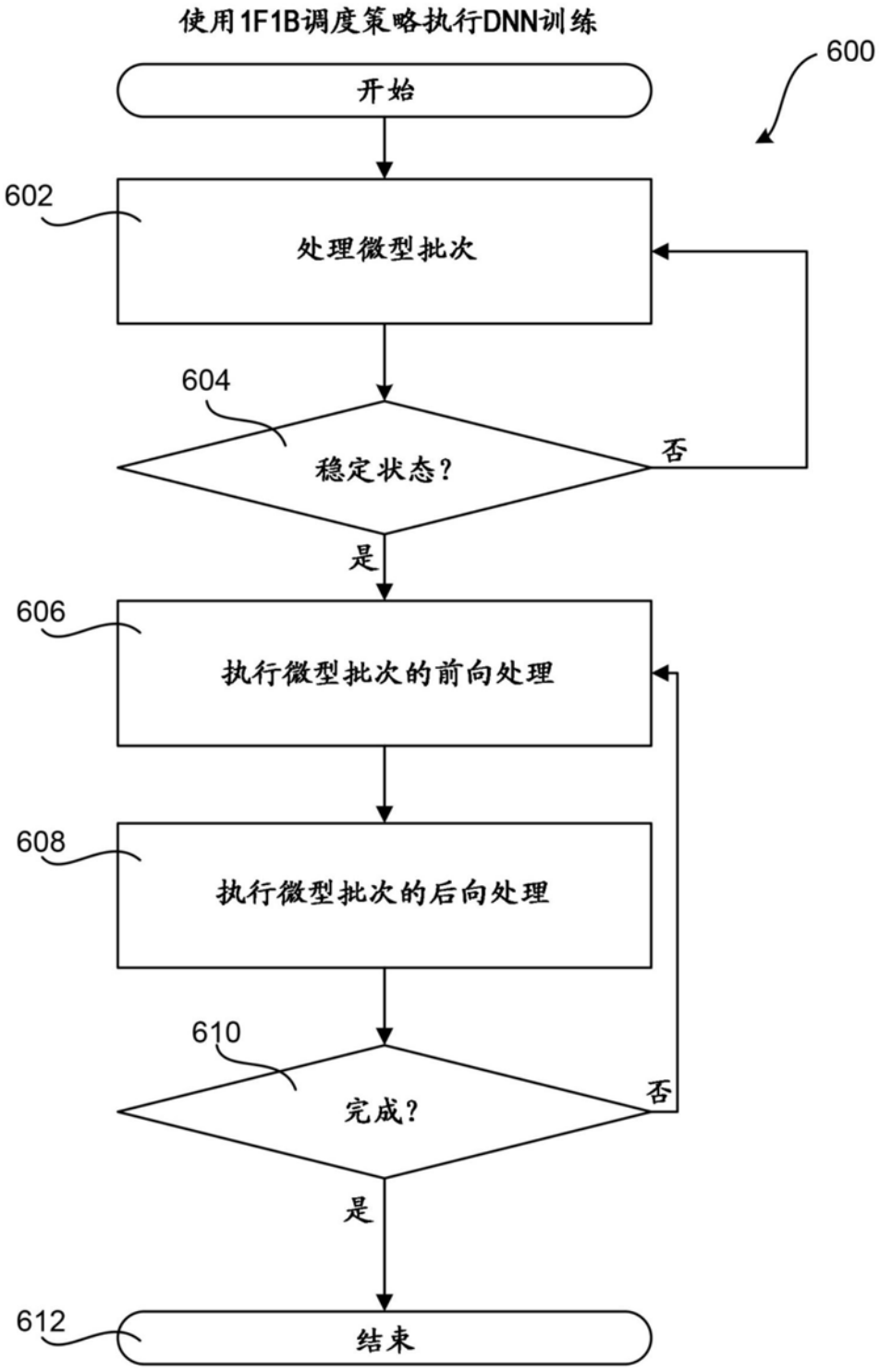

使用1F1B调度策略执行DNN训练
600
开始
602
处理微型批次
604
稳定状态?
否
是
606
执行微型批次的前向处理
608
执行微型批次的后向处理
610
完成?
否
是
612
结束

图6

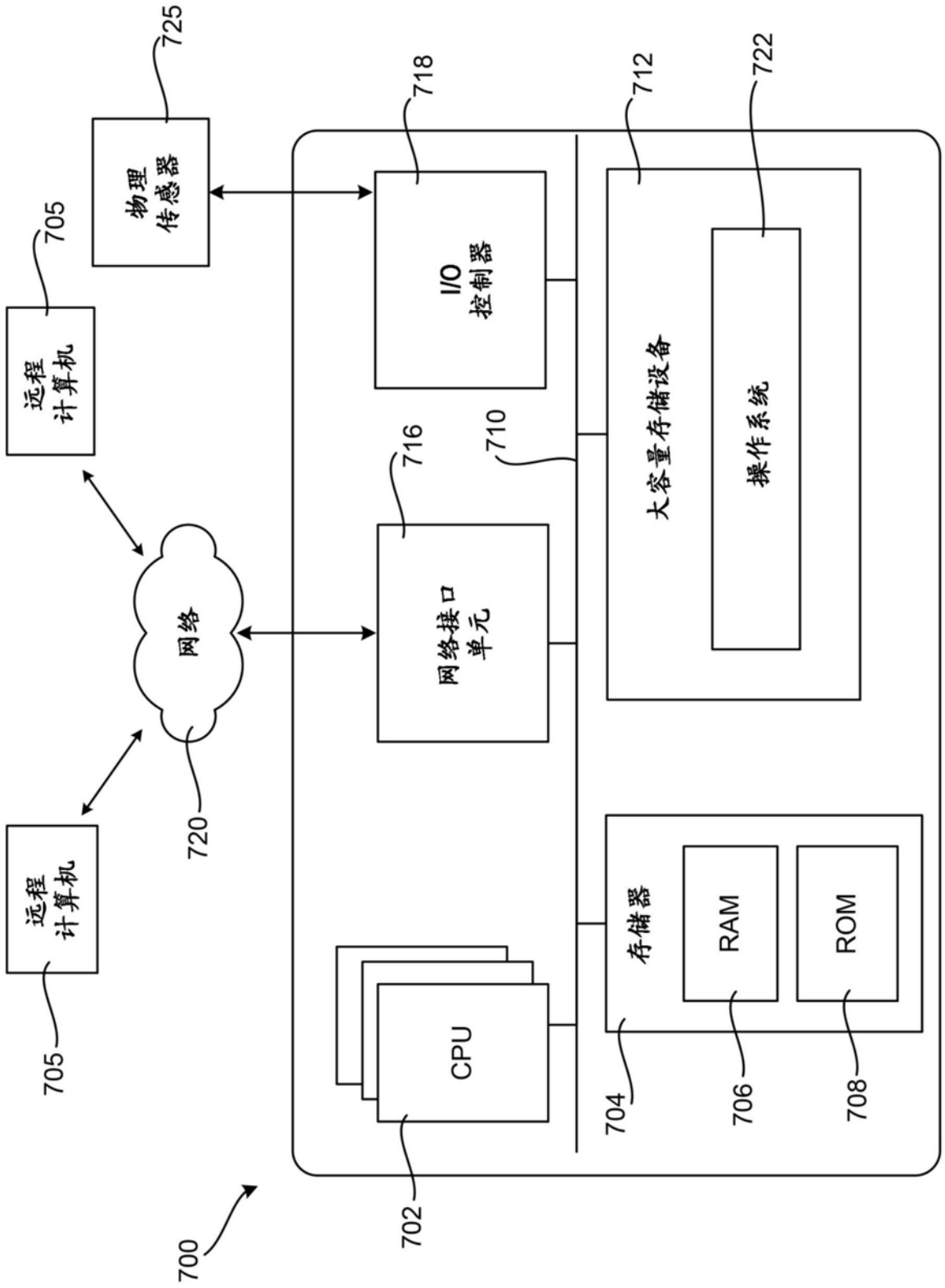

700
702
CPU
704
存储器
706
RAM
708
ROM
710
712
大容量存储设备
716
网络接口单元
718
I/O控制器
720
网络
722
操作系统
725
物理传感器
705
远程计算机
705
远程计算机

图7

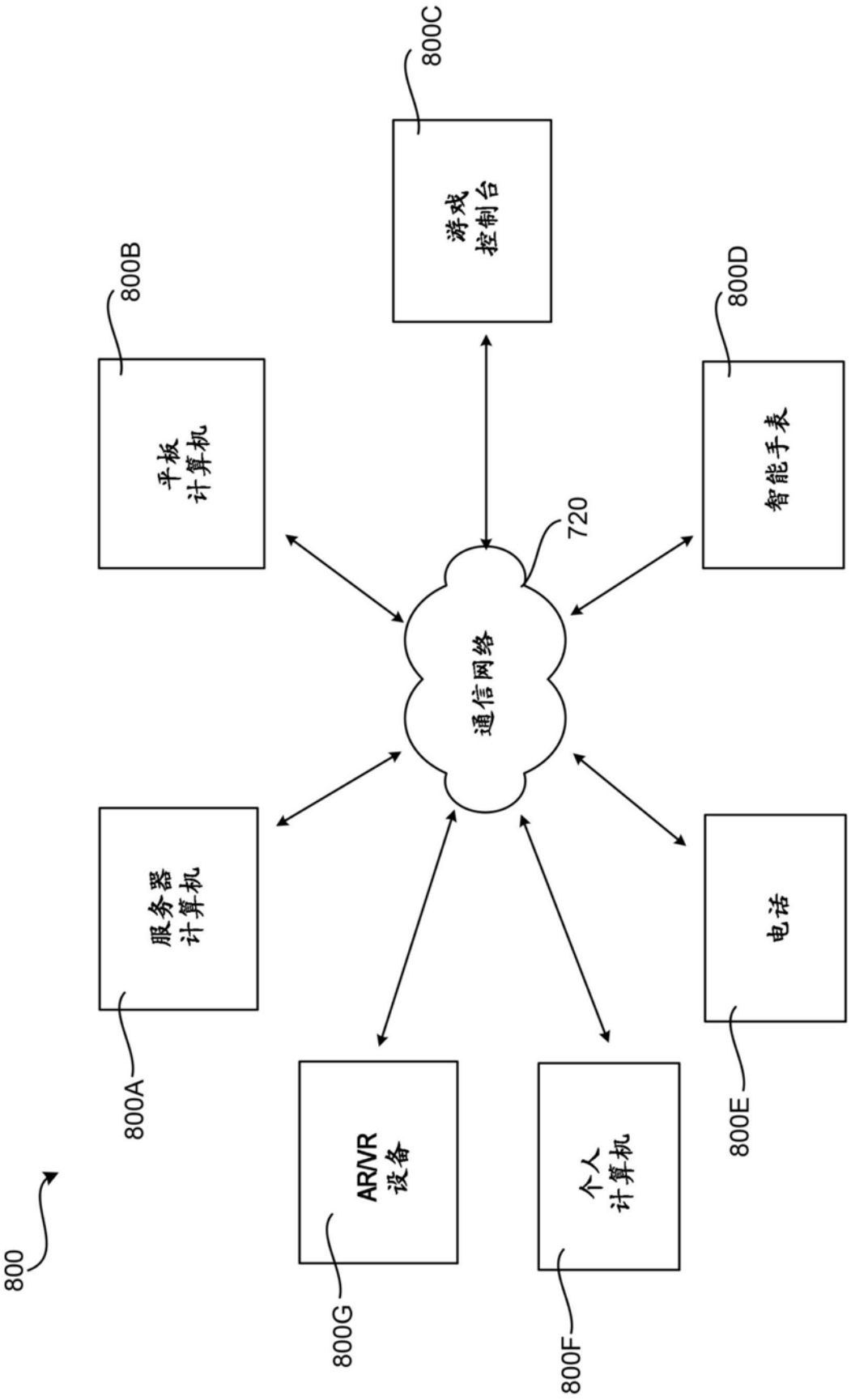

800
800A
服务器计算机
800B
平板计算机
800C
游戏控制台
800D
智能手表
800E
电话
800F
个人计算机
800G
AR/VR设备
720
通信网络

图8