(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114127689 B
(45)授权公告日 2025.06.06

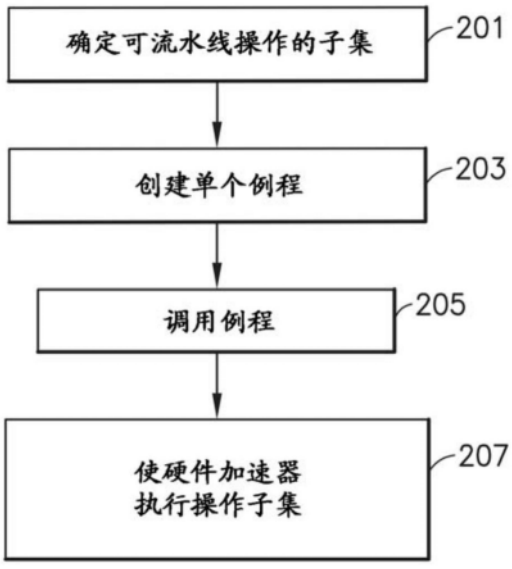
(21)申请号 202080051285.5
(22)申请日 2020.06.30
(65)同一申请的已公布的文献号
申请公布号 CN 114127689 A
(43)申请公布日 2022.03.01
(30)优先权数据
16/511,689 2019.07.15 US
(85)PCT国际申请进入国家阶段日
2022.01.14
(86)PCT国际申请的申请数据
PCT/EP2020/068377 2020.06.30
(87)PCT国际申请的公布数据
WO2021/008868 EN 2021.01.21
(73)专利权人 国际商业机器公司
地址 美国纽约阿芒克
(72)发明人 C·皮维特奥 N·约安诺
I·克拉夫祖克
M·勒加洛-布尔多
A·塞巴斯蒂安
E·S·埃勒夫塞里奥
(74)专利代理机构 北京市金杜律师事务所 11256
专利代理师 酆迅
(51)Int.Cl.
G06F 9/50(2006.01)
G06N 3/0464(2023.01)
G06N 3/08(2023.01)
(56)对比文件
S. R. Nandakumar.Mixed-precision_architecture_based_on_computational_memory_for_training_deep_neural_networks.《2018 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS(ISCAS)》.2018,正文摘要、第II-III部分.
审查员 彭超
权利要求书2页 说明书9页 附图7页

(54)发明名称

用于与硬件加速器接口的方法

(57)摘要

本公开涉及一种用于执行由至少一个操作集合组成的计算任务的方法,其中根据流水线方案确定操作集合的可流水线操作的子集。可以创建单个例程以使得能够由硬件加速器执行所确定的操作子集。例程具有指示计算任务的输入数据和配置参数值的值作为自变量,其中例程的调用使得根据配置参数值来调度硬件加速器上的操作子集。在接收到计算任务的输入数据时,例程可以被调用以使得硬件加速器根据调度由计算任务执行。



CN 114127689 B

1.一种用于执行由至少一个操作集组成的计算任务的计算机实现的方法,所述方法包括:

根据流水线方案确定所述操作集中的可流水线操作的子集;

创建单个例程,用于使得能够由硬件加速器执行所确定的操作子集,所述例程具有指示所述计算任务的输入数据和配置参数值的值作为自变量,其中对所述例程的调用使得能够根据所述配置参数值来调度所述硬件加速器上的所述操作子集;

在接收到所述计算任务的输入数据时,调用所述例程,从而使所述硬件加速器根据所述调度来执行所述计算任务。

2.根据权利要求1所述的方法,所述计算任务包括以下任一项:训练深度神经网络,使用训练后的神经网络来执行推断、矩阵向量乘法和矩阵-矩阵乘法。

3.根据权利要求2所述的方法,其中所述至少一个操作集包括用于所述训练的前向传播的第一操作集、用于所述训练的反向传播的第二操作集和用于所述训练的所述前向传播和所述反向传播两者的第三操作集;所述方法还包括:针对所述第一操作集、第二操作集和第三操作集中的每个操作集生成相应的合成操作,其中调用所述例程包括针对所生成的所述合成操作的至少部分中的每个合成操作执行单个应用编程接口(API)调用。

4.根据权利要求2所述的方法,所述配置参数包括描述所述深度神经网络的结构的参数以及用于配置所述深度神经网络的所述训练所需要的参数。

5.根据权利要求1所述的方法,还包括向所述硬件加速器提供应用编程接口API,并且使用所述API创建所述例程,其中对所述例程的所述调用是单个API调用。

6.根据权利要求1所述的方法,还包括:提供描述所述计算任务的计算图,所述计算任务涉及深度神经网络,通过解析所述计算图以使用所述计算图的节点来标识所述至少一个操作集来确定所述至少一个操作集,生成用户图使得所述至少一个操作集中的每个操作集由所述用户图的节点表示,其中调用所述例程包括标识所述用户图的表示相应操作集的每个节点,并且针对每个标识的节点执行针对由所述标识的所述节点表示的所述操作集的单个API调用。

7.根据权利要求1所述的方法,还包括从所述硬件加速器接收指示所述计算任务的结果的输出。

8.根据权利要求1所述的方法,其中提供所述流水线方案,使得所述操作子集中的每个操作子集包括能够并行执行的彼此独立的操作。

9.根据权利要求1所述的方法,其中所述硬件加速器根据使用忆阻器交叉杆阵列的所述流水线方案进行操作,其中可流水线操作的所述子集被确定,使得所述子集的每个操作子集可以在所述忆阻器交叉杆阵列的不同交叉杆阵列上并行地被执行。

10.根据权利要求1所述的方法,其中所述硬件加速器根据使用忆阻器交叉杆阵列的所述流水线方案来操作,所述计算任务包括训练深度神经网络,其中所述深度神经网络的每层与所述硬件加速器的两个交叉杆阵列相关联,所述两个交叉杆阵列包括相同的值,其中使所述硬件加速器执行所述计算任务包括:对于所述深度神经网络的每层,使用所述两个交叉杆阵列中的一个交叉杆阵列用于前向传播,并且另一个交叉杆阵列仅用于后向传播。

11.一种计算机程序产品,包括计算机可读存储介质,所述计算机可读存储介质具有随其体现的计算机可读程序代码,所述计算机可读程序代码被配置用于:

根据流水线方案确定计算任务的至少一个操作集的可流水线操作的子集;

创建单个例程,用于使得能够由硬件加速器执行所确定的操作子集,所述例程具有指示所述计算任务的输入数据和配置参数值的值作为自变量,其中对所述例程的调用使得根据所述配置参数值来调度所述硬件加速器上的所述操作子集;

在接收到调用所述例程的所述计算任务的输入数据时,由此使所述硬件加速器根据所述调度来执行所述计算任务。

12.根据权利要求11所述的计算机程序产品,所述计算任务包括以下一项:训练深度神经网络、矩阵向量乘法和矩阵-矩阵乘法。

13.根据权利要求12所述的计算机程序产品,至少一个操作集包括用于所述训练的前向传播的第一操作集、用于所述训练的后向传播的第二操作集以及用于所述训练的所述前向传播和所述后向传播两者的第三操作集,所述计算机可读程序代码还被配置用于:为所述第一操作集、第二操作集和第三组操作集中的每个操作集生成相应的合成操作,其中调用所述例程包括为所生成的所述合成操作的至少部分的每个合成操作执行单个应用编程接口(API)调用。

14.根据权利要求12所述的计算机程序产品,所述配置参数包括描述所述深度神经网络的结构的参数以及用于配置所述深度神经网络的所述训练所需的参数。

15.根据权利要求11所述的计算机程序产品,还被配置用于使用到所述硬件加速器的API来创建所述例程。

16.根据权利要求11所述的计算机程序产品,还被配置用于提供描述所述计算任务的计算图,所述计算任务涉及深度神经网络,通过解析用于使用所述计算图的节点来标识所述至少一个操作集的计算图来确定至少一个可流水线操作集,生成用户图使得所述至少一个操作集中的每个操作集由所述用户图的节点表示,其中调用所述例程包括标识所述用户图的表示相应操作集的每个节点,并且针对每个标识的节点执行针对由所述标识的所述节点表示的所述操作集的单个API调用。

17.根据权利要求11所述的计算机程序产品,还被配置用于从所述硬件加速器接收指示所述计算任务的结果的输出。

18.根据权利要求11所述的计算机程序产品,所述流水线方案被提供为使得所述子集中的每个子集包括能够并行执行的彼此独立的操作。

19.一种计算机系统,被配置用于:

根据流水线方案确定计算任务的至少一个操作集的可流水线操作的子集;

创建单个例程,用于使得能够由硬件加速器执行所确定的操作子集,所述例程具有指示所述计算任务的输入数据和配置参数值的值作为自变量,其中对所述例程的调用使得根据所述配置参数值来调度所述硬件加速器上的所述操作子集;

在接收到调用所述例程的所述计算任务的输入数据时,由此使所述硬件加速器根据所述调度来执行所述计算任务。

用于与硬件加速器接口的方法

背景技术

[0001] 本发明涉及数字计算机系统领域,并且更具体地,涉及用于执行由操作集组成的计算任务。

[0002] 硬件加速使得能够使用专门制造的计算机硬件来比在通用CPU上运行的软件中可能更有效地执行一些功能。例如,可以在被设计成比在通用计算机处理器上更快地计算操作的专用硬件中计算操作。然而,需要改进对这些运算中的多个运算的计算。

发明内容

[0003] 各种实施例提供了一种用于执行由操作集组成的计算任务的方法、计算机系统和计算机程序产品。

[0004] 在一个方面,本发明的实施例涉及一种用于执行由至少操作集组成的计算任务的计算机实现的方法。该方法包括:根据流水线方案确定所述操作集合中的可流水线操作的子集;创建单个例程,用于使得能够由硬件加速器执行所确定的操作子集,所述例程具有指示所述计算任务的输入数据和配置参数值作为自变量,其中对所述例程的调用使得根据所述配置参数值来调度所述硬件加速器上的所述操作子集;在接收到调用所述例程的所述计算任务的输入数据时,由此使所述硬件加速器根据所述调度来执行所述计算任务。

[0005] 在另一方面,本发明的实施例涉及一种计算机系统,其被配置用于:根据流水线方案确定计算任务的至少一个操作集合的可流水线操作的子集;创建单个例程,用于使得能够由硬件加速器执行所确定的操作子集,所述例程具有指示所述计算任务的输入数据和配置参数值作为自变量,其中对所述例程的调用使得根据所述配置参数值来调度所述硬件加速器上的所述操作子集;在接收到调用所述例程的所述计算任务的输入数据时,由此使所述硬件加速器根据所述调度来执行所述计算任务。

[0006] 在另一方面,本发明的实施例涉及一种计算机程序产品,其包括具有计算机可读程序代码的计算机可读存储介质。计算机可读程序代码被配置用于:根据流水线方案确定计算任务的至少一个操作集合的可流水线操作的子集;创建用于使得能够由硬件加速器执行所确定的操作集合的单个例程,所述例程具有指示所述计算任务的输入数据和配置参数值作为自变量,其中对所述例程的调用使得根据所述配置参数值来调度所述硬件加速器上的操作的子集;在接收到调用所述例程的所述计算任务的输入数据时,由此使所述硬件加速器根据所述调度来执行所述计算任务。

附图说明

[0007] 下面,仅通过示例,参考附图更详细地解释本发明的实施例,其中:

[0008] 图1描绘了硬件加速器的示例结构。

[0009] 图2A是根据本主题的示例的用于使用硬件加速器来执行由操作集组成的计算任务的方法的流程图。

[0010] 图2B说明用于矩阵-矩阵乘法的管线化方案。

[0011] 图3A示出了用于训练深度神经网络的示例硬件加速器。

[0012] 图3B描绘了示例代码。

[0013] 图3C描绘了用于训练深度神经网络的任务的流程的图。

[0014] 图4是示出训练深度神经网络的流程的图。

[0015] 图5示出了用于执行深度神经网络的训练的交叉杆阵列的示例结构。

具体实施方式

[0016] 本发明的各种实施例的描述将出于说明的目的而呈现,但不希望是详尽的或限于所揭示的实施例。在不背离所描述的实施例的范围和精神的情况下,许多修改和变化对于本领域的普通技术人员将是显而易见的。选择本文所使用的术语以最好地解释实施例的原理、实际应用或对市场上存在的技术改进,或使本领域的其他普通技术人员能够理解本文所公开的实施例。

[0017] 本主题可以通过并行地使用硬件加速器的尽可能多的单元来加速由硬件加速器执行的计算。与操作的串行执行相反,本主题可以利用流水线,因为它不仅向硬件加速器给出关于要执行的任务的一小部分的信息,而且给出关于整个任务的信息。

[0018] 在计算任务是深度神经网络(DNN)的训练的情况下,本主题不仅向硬件加速器给出关于网络的一小部分的信息,而且还给出流水线操作所需的整个网络的信息。本主题可以使得能够将它们分组为一个或多个复合操作,而不是将用于各个网络操作(例如,矩阵乘法、卷积、激活等...)的命令逐一发送到硬件加速器。硬件加速器然后可以根据预定义和优化的流水线来进行这些复合操作并执行它们。例如,由于计算存储器的非冯·诺依曼特性,位于不同交叉杆阵列上的计算资源可以以流水线的形式被重新使用。通过复合运算和流水线操作获得的加速对于线性代数应用可能特别有利。

[0019] 本主题可以提供用于与硬件加速器接口的软件接口。软件接口可以包括使得能够访问硬件加速器的硬件功能的功能。该单个例程可以是软件接口的这些功能的函数。当调用程序调用单个例程时,可以向硬件加速器发出命令以便执行计算任务。到硬件加速器的命令可以作为表示由硬件加速器支持的基本操作序列的复合操作被传递。复合操作可以是例如训练的前向传播和/或后向传播。一旦硬件加速器将数据发送回软件接口,软件接口就可以将数据提供给原始调用程序。可以为至少部分复合操作定义流水线方案(或执行流水线),例如,可以为每个复合操作定义流水线方案。这可以允许硬件加速器的计算能力的最佳使用。

[0020] 根据一个实施例,计算任务包括以下中的任一个:训练深度神经网络DNN、矩阵向量乘法和矩阵乘法。

[0021] 该实施例对于具有大密度矩阵的矩阵向量乘法可能特别有利。例如,由于物理限制,硬件加速器的交叉杆阵列可能仅达到要处理的矩阵的特定大小。为此,可以分割大矩阵的乘法。该实施例可以使用户能够将完整的矩阵-向量乘法作为复合操作来传递。矩阵可以被分解成适当的片并且跨硬件加速器的不同交叉杆阵列分布。然后可以并行地执行单独的矩阵向量乘法。例如,不适合单个交叉杆的矩阵可以用$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$来表示。矩阵M将乘以向

量$\begin{pmatrix} x \\ y \end{pmatrix}$。该实施例可以使得能够使用以下指令来执行该乘法：

[0022] 进行该例程的单个API调用，

[0023] 通过计算存储器软件栈将M分解成A、B、C和D，

[0024] 并行地计算A*X*B*Y*C*X*D*Y*和

[0025] 将计算存储器软件栈的计算结果相加。

[0026] 这与具有以下指令的另一乘法技术形成对比：

[0027] 例如由用户将M拆分成A、B、C和D，

[0028] 进行4个API调用，分别计算A*x、B*y、C*x和D*y，以及

[0029] 由用户相应地累加矩阵。

[0030] 根据一个实施例，所述至少操作集包括用于训练的前向传播的第操作集，和/或用于训练的后向传播的第二操作集，和/或用于训练的前向和后向传播两者的第三操作集。该方法包括：为所述第一操作、第二操作和第三操作集中的每操作集生成相应的合成操作，其中调用所述例程包括为所生成的合成操作的至少一部分的每个合成操作执行单个应用编程接口(API)调用。可以生成或定义复合操作，使得单个API调用足以触发和执行生成复合操作的全部操作。可以生成复合操作，使得其被配置为接收单个输入并且提供执行计算任务的结果(或操作集合的结果)作为输出。这可以使得单个例程能够将指示输入数据的值和计算任务的配置参数值作为自变量。通过对例程的单次调用，可以获得指示期望结果的输出。

[0031] 根据一个实施例，配置参数包括描述深度神经网络的结构的参数和配置深度神经网络的训练所需的参数。

[0032] 根据一个实施例，该方法还包括向硬件加速器提供应用编程接口API，并且使用API创建例程。硬件加速器可以例如是基于人工智能的硬件加速器。

[0033] 根据一个实施例，该方法还包括提供描述计算任务的计算图，计算任务涉及深度神经网络，通过解析计算图以使用计算图的节点来标识至少一个操作集来确定至少一个操作集，生成用户图使得至少一个操作集中的每个操作集由用户图的节点表示，其中调用例程包括标识用户图的表示相应操作集的每个节点，并且对于每个标识的节点执行针对由标识的节点表示的操作集的单个API调用。

[0034] 对于一些应用，操作的程序/序列被表示为计算图(数据流图)，其中节点表示计算的单位。该实施例可以使得能够将这样的计算图转换成完全使用计算存储器硬件的流(例如，通过生成使用复合运算的新表示)。为此，可以使用图解析器来将图中的可流水线化操作分组为复合操作。图解析器可以接收计算图作为输入，并且可以输出具有合并成复合操作的适当操作序列的变换图。使用这种图形解析器，在已经建立的深度学习框架中编写的程序可以直接与计算存储器深度学习加速器一起使用。

[0035] 根据一个实施例，该方法还包括从硬件加速器接收指示计算任务的结果的输出。

[0036] 根据一个实施例，提供流水线方案，使得操作子集中的每个包括可以并行执行的彼此独立的操作。

[0037] 根据一个实施例，硬件加速器根据使用忆阻器交叉杆阵列的流水线方案来操作。确定可流水操作的子集，使得可以在忆阻器交叉杆阵列的不同交叉杆阵列上并行地执行子

集的操作的每个子集。模拟存储器交叉杆阵列提供具有O(1)计算复杂性的廉价向量矩阵计算引擎,为神经网络和线性代数应用提供了有希望的显著加速。

[0038] 根据一个实施例,硬件加速器根据使用忆阻器交叉杆阵列的流水线方案进行操作,计算任务包括训练深度神经网络,其中深度神经网络的每层与硬件加速器的两个交叉杆阵列相关联,两个交叉杆阵列包括相同的值,其中使硬件加速器执行计算任务包括:对于深度神经网络的每层,使用两个交叉杆阵列中的一个交叉杆阵列用于前向传播,并且使用另一个交叉杆阵列仅用于后向传播。

[0039] 图1描绘了硬件加速器的示例结构。硬件加速器100例如可以是基于模拟和/或数字的加速器。

[0040] 硬件加速器100可以被配置为执行计算任务,诸如训练神经网络、利用训练的神经网络运行推理、图像处理、对整数求和等。

[0041] 如同大多数任务一样,计算任务可以被分解成操作集。例如,在求和数字的情况下,任务可以被分解为前缀和运算,其使得能够以最优方式获得整数的和。在机器学习的情况下,大多数计算任务是一个或多个向量-矩阵-乘法和激活函数的组合。例如,深度神经网络涉及向量-矩阵乘法,其中神经元激励的向量x_i将与权重矩阵w_{ij}相乘,从而生成用于下一层的神经元激励的新向量y_j。这将计算任务分解为乘法-累加运算($\Sigma w_{ij}\ x_i$),随后是非线性压制函数。

[0042] 因此,取决于计算任务,硬件加速器100的不同架构可以被设计成实现任务的操作。换言之,具有给定计算任务的本领域技术人员可以提供启用至少部分计算任务的硬件加速器的架构。在下文中,参考人工智能应用来描述硬件加速器100,但是其不限于此。

[0043] 硬件加速器100包括集成电路101。集成电路101被配置为对模拟和/或数字信号执行操作。集成电路101包括多个物理实现的功能单元103A-N,提供功能单元103A-N,使得执行计算任务不需要指令周期的常规指令获取和解码步骤。例如,功能单元103A-n可以形成芯片的层级,包括忆阻器阵列、在阵列外围的ADC、用于缓冲中间项的嵌入式DRAM(eDRAM)以及数字化的阵列输出,例如用于实现DNN的正向推理中涉及的乘法-累积操作。

[0044] 硬件加速器100的功能取决于被选择用于硬件加速器100的功能单元103A-N。例如,可以使用诸如忆阻器交叉杆阵列的大小、交叉杆的数目、ADC的数目等参数以便定义硬件加速器100可以根据其执行计算任务的算法。例如,该算法可以利用并行计算和流水线方案来减少计算任务的步骤数目,并且因此与执行计算的顺序执行的另一算法相比可以减少时间复杂度。

[0045] 因此,取决于用于操作硬件加速器100的算法,功能单元103A-N可以被配置为根据该算法在彼此之间接收和提供数据。为此,硬件加速器100可以包括在时间上控制事件并对事件排序的组件105。组件105可以包括一个或多个有限状态机。有限状态机可以通过将控制向量加载到硬件加速器100中来驱动,例如功能单元103A-N的映射,并且流水线方案可以被离线确定并被加载到驱动有限状态机的控制寄存器中。

[0046] 图2A是根据本主题的示例的用于使用硬件加速器(例如100)执行由操作集组成的计算任务的方法的流程图。

[0047] 为了简化的目的,参考作为矩阵-矩阵乘法的计算任务来描述图2A的方法,但是其不限于此。在矩阵-矩阵乘法的情况下,乘法可以被分解成矩阵-向量乘法的序列,其中,该

组运算是矩阵-向量乘法。

[0048] 为了最佳地或最大程度地使用硬件加速器100,可以使用流水线方案。流水线方案可以定义被划分为多个级的流水线,其中每一级并行地完成计算任务的一部分,并且这些级彼此相关以形成流水线。可以基于功能单元的结构和功能以及计算任务来确定流水线方案,例如,流水线方案的确定可以考虑关于硬件加速器的硬件能力的知识,诸如可以并行计算的忆阻交叉杆操作的数目。

[0049] 在矩阵-矩阵乘法示例之后,计算任务可以是要执行的矩阵乘法$M_1xM_2\cdots xM_5$的链。例如,矩阵中的每个矩阵可以是4×4矩阵。为了以最佳方式执行这个矩阵乘法链,可以使用以下方法或流程:矩阵$M_1\times M_2\cdots\times M_4$的每一矩阵可存储在相应交叉阵列中,最后矩阵$M_5$可分解为列向量且所述向量可馈送到交叉阵列中,如图2B中所说明。基于此流程,可定义管线化方案以最佳地执行如图2B的表220中所示的乘法$M_1\times M_2\cdots\times M_5$,其中定义5个级(或时间步长)222.1-5,并且在每一级中可执行一个或多于一个矩阵向量乘法。如图2B所示,在第一级222.1中,使用存储矩阵M_n的交叉阵列,例如,交叉阵列被馈送以向量x_1的4个元素,可以执行$x^2_1=M_nx_1$的仅一个初始第一矩阵向量乘法。该第一级222.1可以将X^2_1作为输出(乘法的结果)提供给第二级222.2。在第二级222.2中,存储矩阵M_n的交叉杆阵列可执行$x^2_2=M_nx_2$第二矩阵向量乘法,这是因为所述交叉杆阵列在完成第一级之后变为空闲。与第二乘法并行地,可以执行第三乘法,即$x^3_1=M_{n-1}x^2_1$。由于第三乘法需要第一乘法的结果,因此在执行第一乘法之后,仅在第二级222.2中执行第一乘法。在最后两个阶段222.4-5中,所有交叉杆阵列并行地运行相应的乘法,从而实现硬件加速器的完全使用。

[0050] 因此,基于流水线方案,可以在步骤201中根据流水线方案从操作集合确定可流水线操作的子集。可流水线化操作的子集可以例如包括可以例如在流水线的给定级中并行执行的操作。所确定的操作子集可以允许硬件加速器100的完全或最优利用。根据图2B的实施例,第一操作子集可以包括$x^2_1=M_nx_1$操作,第二操作子集可以包括$x^2_2=M_n2$和$x^3_1=M_{n-1}x^2_1$两个操作,第三操作子集可以包括$x^2_3=M_nx_3$、$x^3_2=M_{n-1}x^2_2$和$x^41=M_{n-2}x^3_1$三个操作,等等。

[0051] 已经定义了例如如图2B所示的要执行的操作的流水线,本方法可以是有利的,因为它可以仅需要单个例程来使得能够执行整个计算任务。在步骤203中可以创建单个例程,使得例程的自变量可以向硬件加速器指示使得能够执行流水线的数据,例如,不需要来自例程的进一步输入。例如,自变量可以包括指示输入数据的值和计算任务的配置参数值。在一个示例中,可以提供API以便与硬件加速器100对接,其中单个例程可以是API的函数。在这种情况下,单个例程的调用可以被称为API调用。在另一示例中,可以使用API的函数来定义单个例程。

[0052] 例程的调用使得根据配置参数值来调度硬件加速器100上的操作子集。例如,配置参数值可以作为控制向量被加载到硬件加速器100中以驱动在每个周期/阶段之后正确地操纵输入和输出的有限状态机。

[0053] 例如,例程的调用可以如下执行:1)进行引用所有5个矩阵的单个API调用;2)软件栈将$M_1M_2M_3$和M_4映射到交叉器阵列上,以及3)X的行向量以流水线方式通过交叉器。这与进行至少5个API调用以计算单独的矩阵-矩阵乘法的方法形成对比。

[0054] 步骤201和203可以例如在使用硬件加速器100进行计算之前离线执行。

[0055] 在接收到计算任务的输入数据时,在步骤205中可以调用例程,使得硬件加速器

100可以根据调度在步骤207中执行计算任务。可以从硬件加速器100接收计算任务的结果。根据上述示例,硬件加速器可以包括分别存储矩阵M_1至M_4的元素的4个交叉杆阵列。在此情况下,例程的自变量可以包括矩阵M_5的向量x_1至x4,作为输入数据以及作为指示矩阵M_1、M_2、M3和M_4的配置参数。例如,不是执行以下四个调用,mm_1=Matmul(M_4,M_5);mm_2=Matmul(M3,mm_1);mm3=Matmul(M_2,mm_2);以及OUTPUT=Matmul(M_1,mm3),单个调用(例如API调用)可以如下执行OUTPUT=composition(config,M_5),其中配置参数可以被定义为config=MatrixMatrixMultiplicationChain(M_1,M_2,M3,M_4)。

[0056] 图3A示出了用于训练DNN的示例硬件加速器300,该DNN具有输入层301、一个隐藏层303和输出层305。在这种情况下,该操作集可以包括用于训练的前向传播的操作和/或用于训练的后向传播的操作。

[0057] 三层分别具有784、250、10个神经形态神经元装置。输出层具有表示10个可能的数字0至9的10个神经形态神经元装置,并且输入层具有表示输入MNIST图像的像素数目的784个神经形态神经元装置。神经元装置中的每一个神经元装置可以被配置为使用激活函数以用于基于神经元装置的当前状态(例如,由x_i定义)来生成输出值。硬件加速器300还可以包括两个交叉杆阵列或忆阻交叉杆阵列(未示出),用于分别计算权重元素W_{JI}和W_{KJ}与激活向量x的乘法,例如,具有元素W_{JI}的矩阵W与输入层的激活向量x的矩阵-向量乘法可以通过第一忆阻交叉杆阵列,通过用第一忆阻交叉杆阵列的对应忆阻元件的电导来表示每个矩阵元素来实现,其中,矩阵W和向量x的乘法可以通过将表示向量值x的电压输入到第一忆阻交叉杆阵列来执行,并且所得到的电流指示W和x的乘积。交叉杆阵列的电阻存储元件(或器件)可以例如是相变存储器(PCM)、金属氧化物电阻RAM、导电桥RAM和磁RAM中的一项。在图3A的这个示例中,功能单元可以包括至少两个交叉杆阵列和神经形态神经元装置。

[0058] 知道了作为3层DNN的训练的计算任务并且可以访问硬件加速器300的功能单元的操作方式,可以利用给定数目的级来定义流水线方案(参见图3C),其中在每一级中,可以由硬件加速器300的功能单元并行地执行一个或多个操作。

[0059] 代替如图3B的代码310所示的对于每层操作(例如,矩阵乘法、卷积、激活、池化等...)具有一个API调用,可使用如图3B的代码312所示的单个API调用313。API调用313的输入可以是MNIST图像和描述DNN的配置参数314,如代码312所指示的。通过执行代码312,多个操作可被链接并且一起执行。

[0060] 图3C描绘了图示根据本主题的示例的用于图3A的DNN的训练的执行方案或算法的第一图330,以及图示根据本主题的示例的用于图3A的DNN的训练的执行方案的第二图350,以及图示根据本主题的另一示例的图3A的DNN的训练的执行方案的第三图360。

[0061] 例如,DNN的训练可能需要输入多个图像集,并且对于每个图像集,可以在不改变突触权重的情况下执行前向传播,使得可以通过组合针对该图像集(而不是仅一个图像)获得的误差来估计要后向传播的DNN的预测误差。

[0062] 第一图330是指示计算任务的流程的计算图。例如,可以响应于matmul函数333的第一API调用,将第一组输入的权重331和输入向量332相乘。第一API调用的结果用于执行S形函数334的第二API调用。第二API调用的结果被用来执行Matmul函数335的第三API调用,包括将权重336和第二AP调用的向量结果相乘。第三API调用的结果用于执行S形函数337的第四API调用。从第四API调用得到的向量和输入332的标签338可以用于计算损失函数339。

从第四API调用得到的向量和标签338之间的差可以用于计算由DNN执行的预测误差δ。可以反向传播计算出的预测误差δ。并且可以在反向传播之后使用所有权重的增量ΔW来更新权重331和336,如图340所示。可以针对每个附加输入332重复这些API调用,直到执行计算任务,例如,计算任务可能需要100个输入图像用于前向传播。在完成第一组输入的最后一个API调用之后,第二组输入进入第一图。因此,在处理第一集合(或第二集合的输入)时,遵循第一图300的流程执行的计算任务可能不会受益于权重336和331针对每个集合的输入不改变的事实,例如,存储权重336和331的交叉杆中的每一个不被用于并行计算。

[0063] 为了利用并行计算,可以使用由第二图350描述的流程。第二图350是根据本主题的示例的指示计算任务的流程的计算图。为了实现第二图350的流程,可以定义两个流水线方案,一个用于训练的前向传播,另一个用于训练的后向传播。在这种情况下,该组输入332与权重331和336结合被提供作为复合操作353的输入,该复合操作可以由单个例程调用以执行前向传播。复合操作353可以根据流水线方案处理输入,例如,如果输入集合包括两个图像,则在第一级期间,第一交叉杆阵列仅处理第一图像,而在流水线的第二级/周期期间,第二交叉杆存储权重336,并且并行地,使用存储权重331的第一交叉杆阵列处理第二图像。如上所述,使用损失函数339来估计预测误差。预测误差可以使用矩阵-向量乘法来反向传播。这由另一复合操作355指示。复合操作355可以以与前向传播类似的方式根据流水线方案处理预测误差的后向传播的输入。并且,如图示380所示,可以使用所有权重的增量ΔW来更新权重331和336。

[0064] 因此,在DNN的训练期间,第二图表350使得能够在不同的复合操作中执行前向和后向传播。前向和后向传播之间的这一分离可能是有利的,因为第二图表350设计可仅用于推断(而不需要执行后向传播)。另外,第二图350的流程可以直接与需要关于整个批次的信息(例如,批次归一化)的技术一起工作,并且该信息发生在前向和后向传播过程之间的阶段中。这在图4中示出,其中批归一化仍可保持与用于前向和后向传播的流水线方案分开或独立。这还可以使得能够具有用于选择损失函数的更大自由度,因为损失函数不被两个流水线方案覆盖。简要地,图4描述了两种方案,第一种方案具有卷积402、修正线性单元404、卷积406、修正线性单元408、批次归一化410、卷积412、修正线性单元414、卷积416和修正线性单元418的运算,第二种方案具有复合运算420、批次归一化422和复合运算424。

[0065] 回到图3C,为了进一步利用并行计算,可以使用由第三图描述的流程。第三图360是根据本主题的示例的指示计算任务的流程的计算图。为了实现第三图表360的流程,为前向和后向传播以及损耗函数计算定义了一个流水线方案。在这种情况下,输入集332与权重331和336两者结合被提供作为复合操作363的输入,该复合操作可以由单个例程调用,以根据试图尽可能多地并行化操作的流水线方案来执行前向传播和后向传播。要并行化的那些操作涉及使用交叉矩阵的矩阵向量乘法和使用神经元的激活函数以及损失函数计算。例如,当第二交叉杆阵列用于反向传播误差信号时,第一交叉杆阵列可用于计算前向传播的矩阵向量乘法。在该示例中,可能需要附加的存储器来保存前向和后向传播计算的激活和误差信号。

[0066] 因此,在DNN训练期间,第三图表360使得能够在相同的复合操作中执行前向和后向传播。这可能是有利的,因为它可能需要较少的存储器消耗。例如,一旦计算出ΔW,就可以丢弃预先存储的层激活,并且可以将存储器重新用于该批次中的另一样本。另一优点可

以是第三图的执行流可以要求更少的开销。例如,在复合操作的开始和结束处,可能总是存在并非所有阵列都被使用的开销时段。通过减少复合操作的数目,可以减少该开销。

[0067] 第三图360的流程的另一优点可以是该流程可以与图5所示的阵列复制技术组合,例如,DNN的两个交叉杆阵列可以被复制(即,包含相同权重的多个交叉杆阵列),使得一个交叉杆阵列仅用于前向传递,而另一个仅用于后向传递,如图5所示,图5的层1(项502)和层2(项504)分别涉及DNN的输入层301和隐藏层303。阵列Array1和Array 2是交叉杆阵列,它们执行分别在输入层和隐藏层之间以及在隐藏层和输出层之间发生的矩阵向量乘法。这可以允许在相同层上同时执行多个操作。具体地,图5示出了通过层1阵列2(项510)然后经由通过层2阵列2的前向传播518(项512)输入的数据514;然后从层2阵列2到层2阵列1(项508);然后从层2阵列1(项508)经由反向传播516通过层1阵列1(项506)。

[0068] 在此参考根据本发明实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述本发明实施例的各方面。将理解,流程图和/或框图的每个框以及流程图和/或框图中的框的组合可以由计算机可读程序指令来实现。

[0069] 本发明的实施例可以是系统、方法和/或计算机程序产品。计算机程序产品可以包括其上具有计算机可读程序指令的计算机可读存储介质(或多个介质),所述计算机可读程序指令用于使处理器执行本发明的实施例的各方面。

[0070] 计算机可读存储介质可以是能够保留和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质可以是例如但不限于电子存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或前述的任何合适的组合。计算机可读存储介质的更具体示例的非穷举列表包括以下:便携式计算机磁盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式光盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、诸如上面记录有指令的打孔卡或凹槽中的凸起结构的机械编码装置,以及上述的任何适当组合。如本文所使用的计算机可读存储介质不应被解释为暂时性信号本身,诸如无线电波或其他自由传播的电磁波、通过波导或其他传输介质传播的电磁波(例如,通过光纤线缆的光脉冲)、或通过导线传输的电信号。

[0071] 本文描述的计算机可读程序指令可以从计算机可读存储介质下载到相应的计算/处理设备,或者经由网络,例如因特网、局域网、广域网和/或无线网络,下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光传输光纤、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或网络接口从网络接收计算机可读程序指令,并转发计算机可读程序指令以存储在相应计算/处理设备内的计算机可读存储介质中。

[0072] 用于执行本发明的实施例的操作的计算机可读程序指令可以是汇编指令、指令集架构(ISA)指令、机器相关指令、微代码、固件指令、状态设置数据,或者以一种或多种编程语言的任意组合编写的源代码或目标代码,所述编程语言包括面向对象的编程语言,诸如Smalltalk、C++等,以及常规的过程式编程语言,诸如“C”编程语言或类似的编程语言。计算机可读程序指令可以完全在用户的计算机上执行,部分在用户的计算机上执行,作为独立的软件包执行,部分在用户的计算机上并且部分在远程计算机上执行,或者完全在远程计算机或服务器上执行。在后一种情况下,远程计算机可以通过任何类型的网络连接到用户

的计算机,包括局域网(LAN)或广域网(WAN),或者可以连接到外部计算机(例如,使用因特网服务提供商通过因特网)。在一些实施例中,为了执行本发明的实施例的各方面,包括例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA)的电子电路可以通过利用计算机可读程序指令的状态信息来执行计算机可读程序指令以使电子电路个性化。

[0073] 在此参考根据本发明实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述本发明实施例的各方面。将理解,流程图和/或框图的每个框以及流程图和/或框图中的框的组合可以由计算机可读程序指令来实现。

[0074] 这些计算机可读程序指令可以被提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器以产生机器,使得经由计算机或其他可编程数据处理装置的处理器执行的指令创建用于实现流程图和/或框图的一个或多个框中指定的功能/动作的装置。这些计算机可读程序指令还可以存储在计算机可读存储介质中,其可以引导计算机、可编程数据处理装置和/或其他设备以特定方式工作,使得其中存储有指令的计算机可读存储介质包括制品,该制品包括实现流程图和/或框图的一个或多个框中指定的功能/动作的各方面的指令。

[0075] 计算机可读程序指令还可以被加载到计算机、其他可编程数据处理装置或其他设备上,以使得在计算机、其他可编程装置或其他设备上执行一系列操作步骤,以产生计算机实现的过程,使得在计算机、其他可编程装置或其他设备上执行的指令实现流程图和/或框图的一个或多个框中指定的功能/动作。

[0076] 附图中的流程图和框图示出了根据本发明的各种实施例的系统、方法和计算机程序产品的可能实现的架构、功能和操作。在这点上,流程图或框图中的每个框可以表示指令的模块、段或部分,其包括用于实现指定的逻辑功能的一个或多个可执行指令。在一些替代实施方案中,框中所提及的功能可不按图中所提及的次序发生。例如,连续示出的两个框实际上可以基本上同时执行,或者这些框有时可以以相反的顺序执行,这取决于所涉及的功能。还将注意,框图和/或流程图图示的每个框以及框图和/或流程图图示中的框的组合可以由执行指定功能或动作或执行专用硬件和计算机指令的组合的专用的基于硬件的系统来实现。

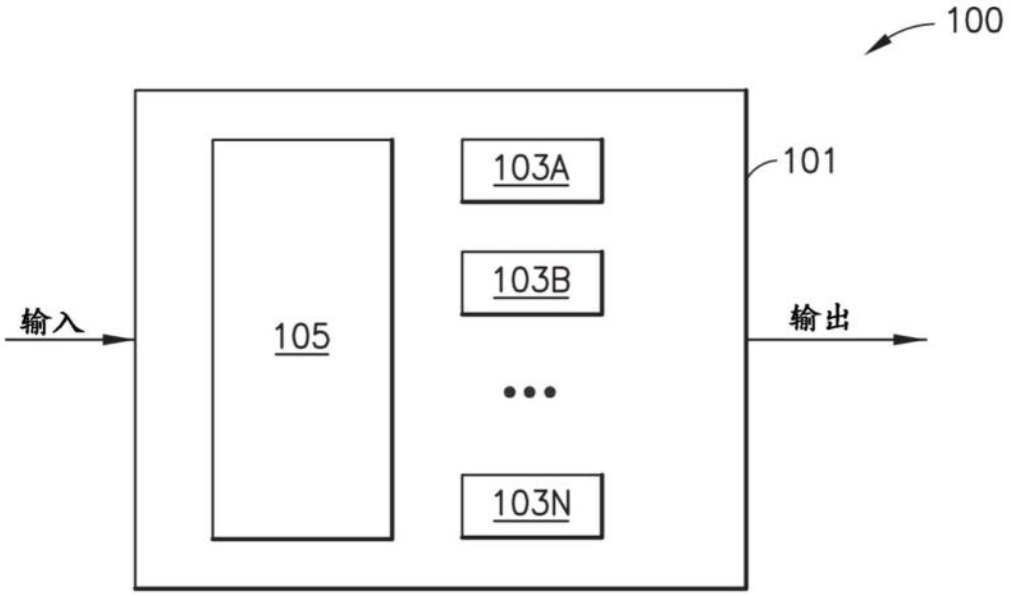

100
101
103A
103B
103N
105
输入
输出

图1

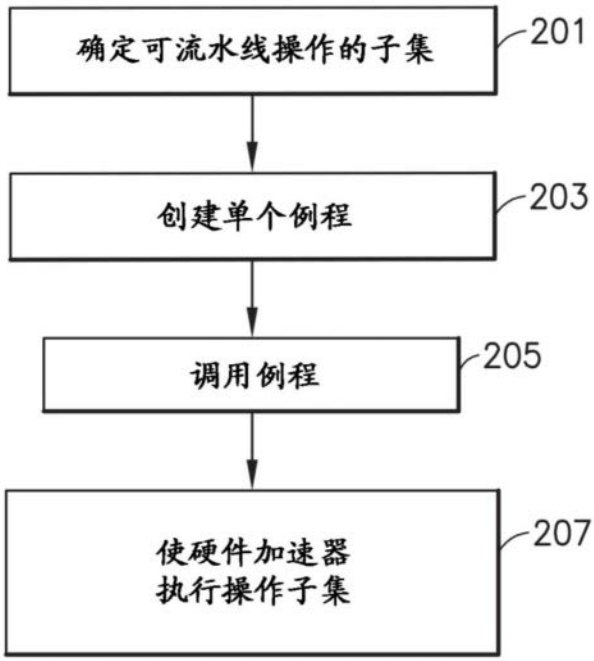

确定可流水线操作的子集
201
创建单个例程
203
调用例程
205
使硬件加速器
执行操作子集
207

图2A

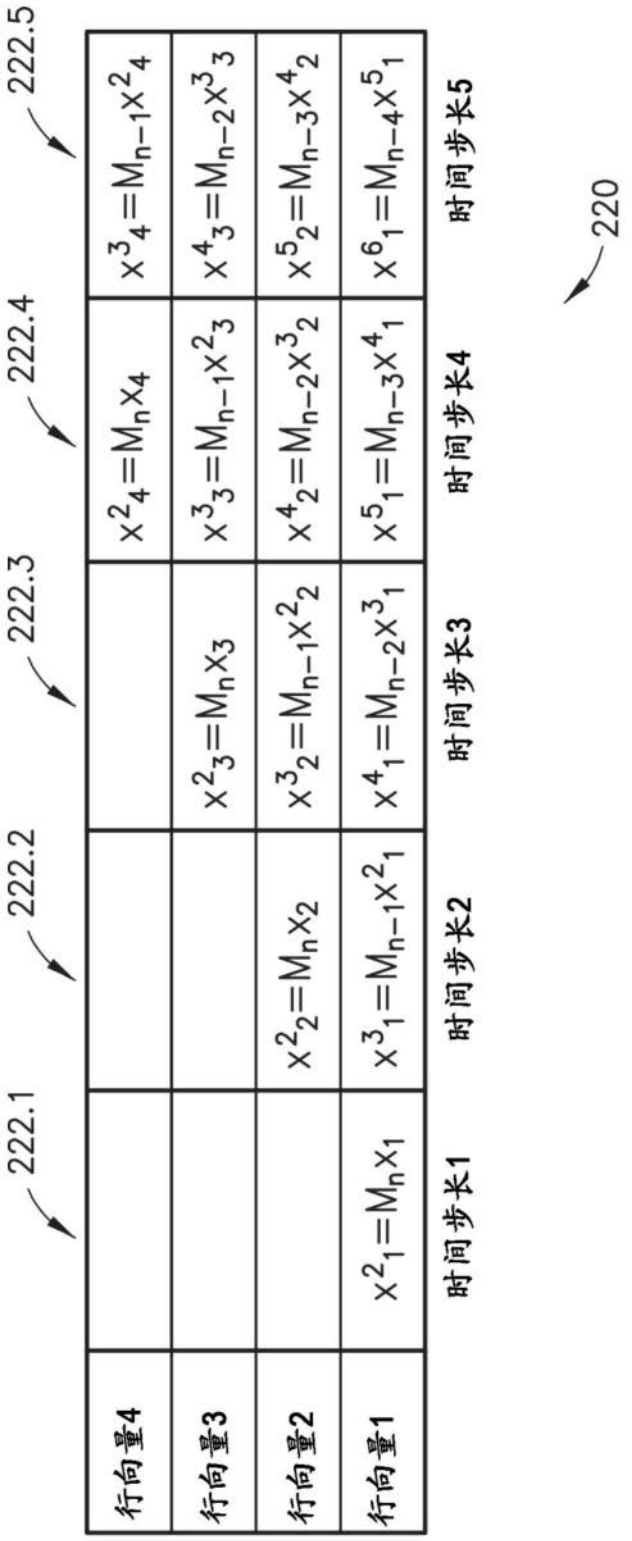

222.1
222.2
222.3
222.4
222.5
220
行向量4
行向量3
行向量2
行向量1
$x^2_4=M_nx_4$
$x^3_4=M_{n-1}x^2_4$
$x^2_3=M_nx_3$
$x^3_3=M_{n-1}x^2_3$
$x^4_3=M_{n-2}x^3_3$
$x^2_2=M_nx_2$
$x^3_2=M_{n-1}x^2_2$
$x^4_2=M_{n-2}x^3_2$
$x^5_2=M_{n-3}x^4_2$
$x^2_1=M_nx_1$
$x^3_1=M_{n-1}x^2_1$
$x^4_1=M_{n-2}x^3_1$
$x^5_1=M_{n-3}x^4_1$
$x^6_1=M_{n-4}x^5_1$
时间步长1
时间步长2
时间步长3
时间步长4
时间步长5

图2B

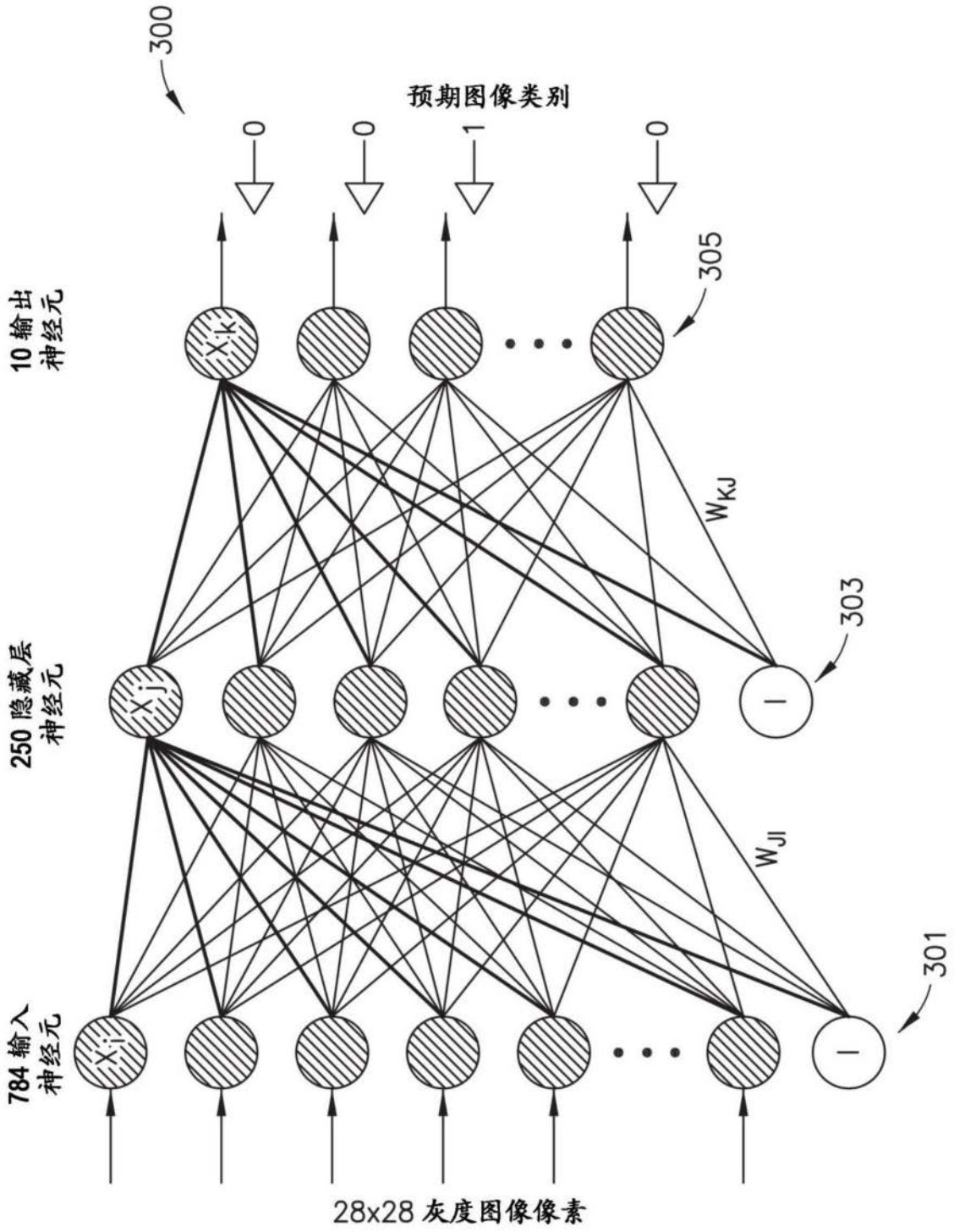

300
预期图像类别
0
0
1
0
305
10 输出
神经元
Xk
WKJ
303
250 隐藏层
神经元
Xj
WJI
301
784 输入
神经元
Xi
28x28 灰度图像像素

图3A

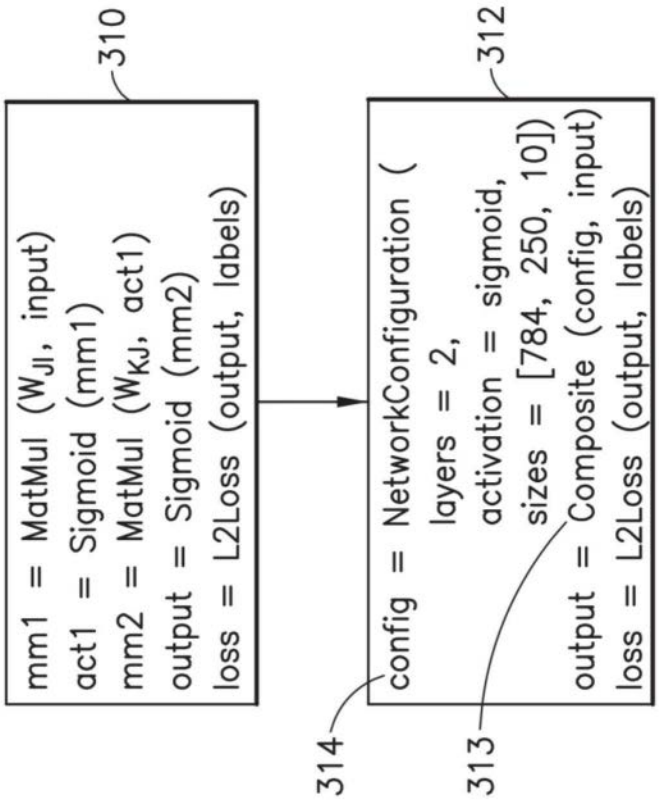

310
mm1 = MatMul (W_{JI}, input)
act1 = Sigmoid (mm1)
mm2 = MatMul (W_{KJ}, act1)
output = Sigmoid (mm2)
loss = L2Loss (output, labels)
312
config = NetworkConfiguration (
layers = 2,
activation = sigmoid,
sizes = [784, 250, 10])
output = Composite (config, input)
loss = L2Loss (output, labels)
314
313

图3B

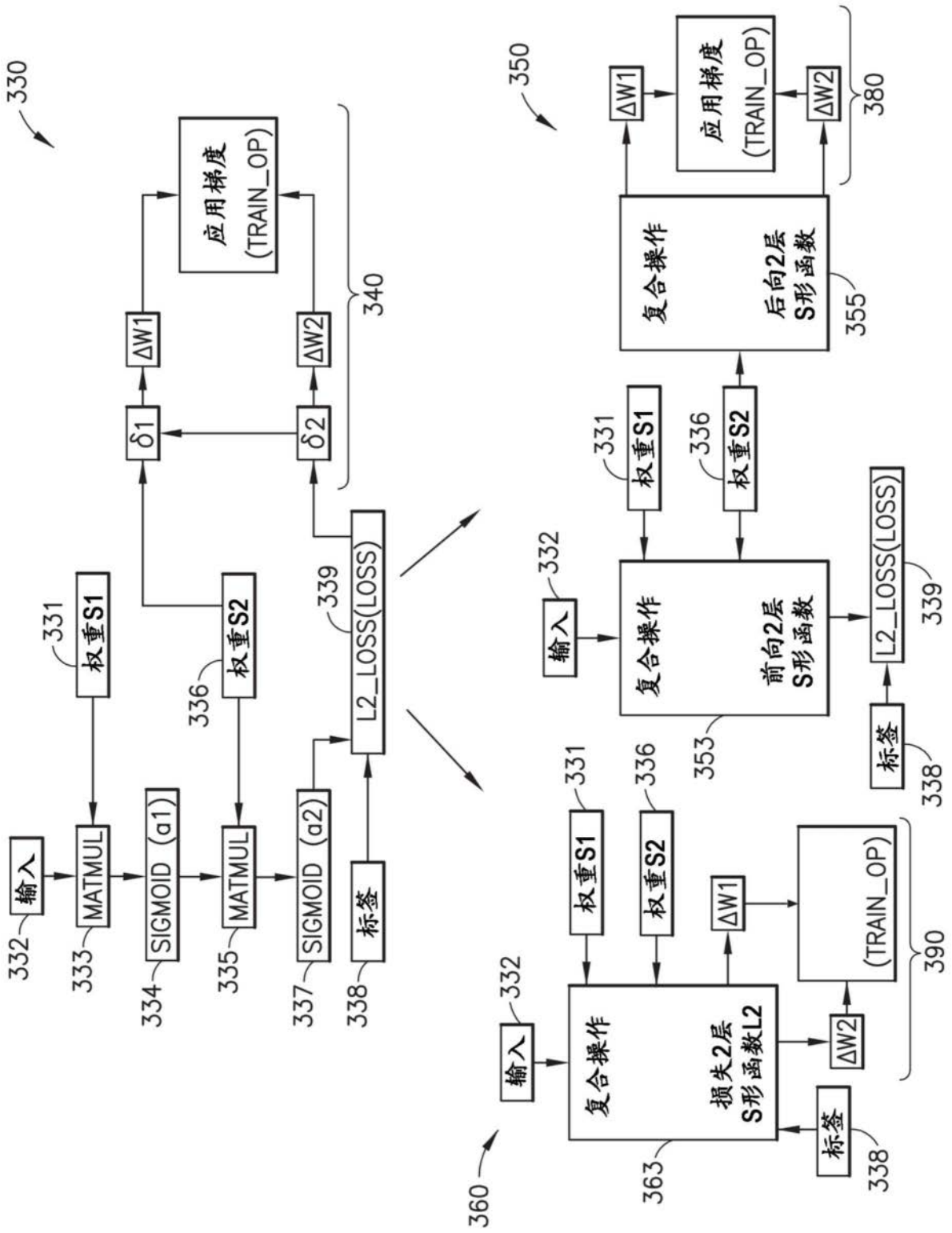

330
输入
332
MATMUL
333
SIGMOID (a1)
334
MATMUL
335
SIGMOID (a2)
337
标签
338
权重S1
331
权重S2
336
L2_LOSS(LOSS)
339
δ1
δ2
ΔW1
ΔW2
应用梯度
(TRAIN_OP)
340
350
复合操作
前向2层
S形函数
353
复合操作
后向2层
S形函数
355
L2_LOSS(LOSS)
339
标签
338
ΔW1
ΔW2
应用梯度
(TRAIN_OP)
380
360
复合操作
损失2层
S形函数L2
363
(TRAIN_OP)
390

图3C

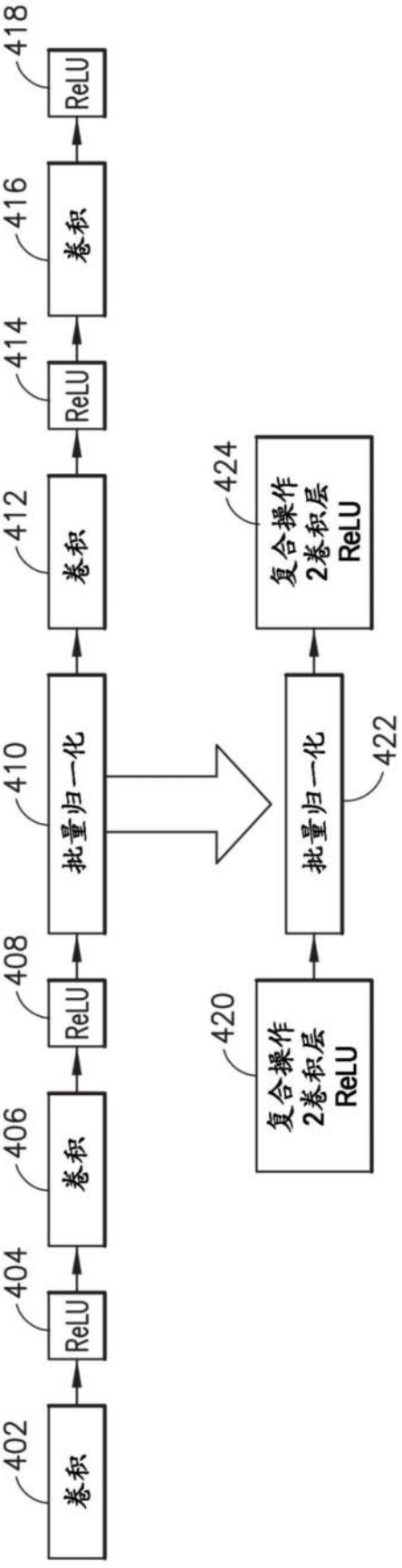

402
卷积
404
ReLU
406
卷积
408
ReLU
410
批量归一化
412
卷积
414
ReLU
416
卷积
418
ReLU
420
复合操作
2卷积层
ReLU
422
批量归一化
424
复合操作
2卷积层
ReLU

图4

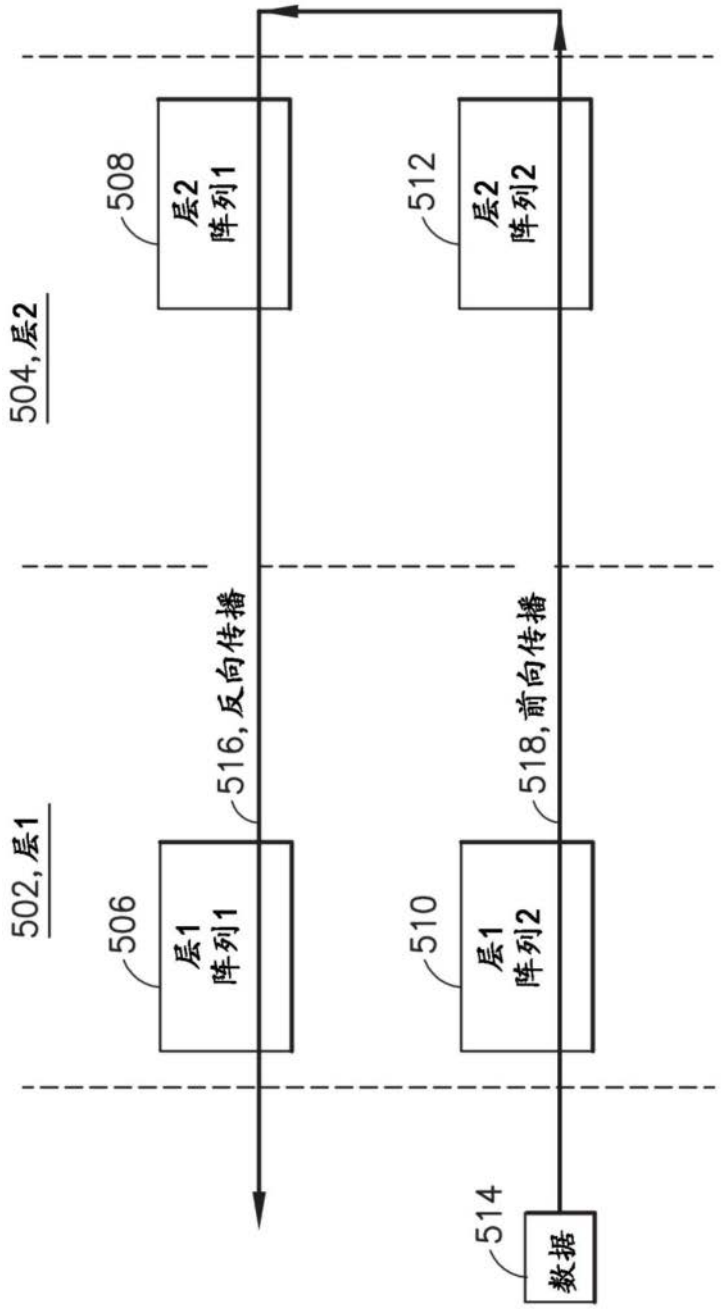

504,层2
502,层1
508
层2
阵列1
512
层2
阵列2
506
层1
阵列1
510
层1
阵列2
516,反向传播
518,前向传播
514
数据

图5