



(12) 发明专利申请

(10) 申请公布号 CN 111788583 A

(43) 申请公布日 2020.10.16

(21) 申请号 201980007826.1

(22) 申请日 2019.02.11

(30) 优先权数据

62/628,882 2018.02.09 US

(85) PCT国际申请进入国家阶段日

2020.07.09

(86) PCT国际申请的申请数据

PCT/US2019/017495 2019.02.11

(87) PCT国际申请的公布数据

W02019/157442 EN 2019.08.15

(71) 申请人 渊慧科技有限公司

地址 英国伦敦

(72) 发明人 K.西蒙扬 N.E.卡尔赫布雷纳

E.K.埃尔森

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 金玉洁

(51) Int.Cl.

G06N 3/063 (2006.01)

G06N 3/08 (2006.01)

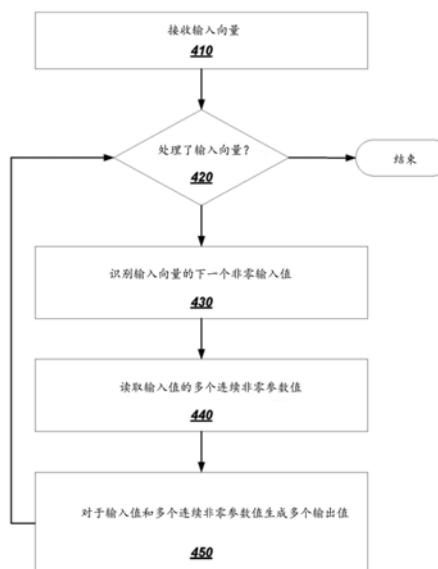
权利要求书2页 说明书12页 附图4页

(54) 发明名称

连续稀疏性模式神经网络

(57) 摘要

用于使用具有连续稀疏性模式的神经网络的方法、系统和装置,包括编码在计算机存储介质上的计算机程序。方法之一包括将具有连续稀疏性模式的神经网络的第一参数矩阵存储在与计算设备相关联的存储中。计算设备执行神经网络的推断过程以生成输出向量,包括从与计算设备相关联的存储中读取来自输入向量的一个或多个激活值,从与计算设备相关联的存储中读取非零参数值块,以及将一个或多个激活值中的每一个乘以非零参数值块中的一个或多个。



1. 一种方法,包括:

接收神经网络的一个或多个参数矩阵,其中至少第一参数矩阵具有以连续稀疏性模式存储的参数值,连续稀疏性模式是由零值和非零值的多个非方形的矩形块组成的二维模式,零值的数量大于非零值的数量;

将第一参数矩阵存储在与计算设备相关联的存储装置中;

接收激活值的输入向量;以及

由计算设备执行神经网络的推断传递以生成输出向量,包括执行包括以下的操作:

从与计算设备相关联的存储装置读取来自输入向量的一个或多个激活值,

从与计算设备相关联的存储装置读取非零参数值的块,以及

将一个或多个激活值中的每一个乘以非零参数值的块中的一个或多个。

2. 根据权利要求1所述的方法,其中,推断传递包括以下操作:

从与计算设备相关联的存储装置读取来自输入向量的第一激活值,

从与计算设备相关联的存储装置读取多个连续非零参数值,以及

通过包括在从输入向量读取另一激活值之前将第一激活值乘以多个连续非零参数值的过程,生成输出向量的多个输出值。

3. 根据权利要求1所述的方法,其中,推断传递包括以下操作:

从与计算设备相关联的存储装置读取来自输入向量的多个激活值,

从与计算设备相关联的存储装置读取多个非零参数值的块,以及

通过包括在读取另一参数值之前将多个非零激活值中的每一个乘以多个非零参数值的块中的相应的一个或多个的过程,生成输出向量的输出值。

4. 根据任一前述权利要求所述的方法,还包括:

接收表示计算设备的处理器的数量的值;

计算第一参数矩阵的非零值的块的数量;以及

将基本相等数量的块或块参数值分配给计算设备的每一个处理器。

5. 根据权利要求4所述的方法,其中,计算设备的处理器是中央处理单元CPU的处理核心。

6. 根据权利要求4所述的方法,其中,计算设备的处理器是专用处理器的处理元件。

7. 根据权利要求1-6中任一项所述的方法,还包括:

生成对第一参数矩阵中的接下来的非零值的位置进行编码的多个源代码语句,

其中,由计算设备执行推断传递包括运行生成的多个源代码语句。

8. 根据权利要求7所述的方法,其中,源代码语句省略读取第一参数矩阵中的接下来的非零值的位置的一个或多个存储器访问指令。

9. 根据权利要求1-8中任一项所述的方法,还包括:

训练神经网络,包括执行包括以下的操作:

在一个或多个块的参数值的量值满足阈值标准的确定之后,将零分配给参数值的一个或多个块。

10. 根据权利要求9所述的方法,包括在分配步骤之前,根据每一个块的参数值的函数对参数值的块进行排序的步骤。

11. 根据权利要求1-10中任一项所述的方法,其中,块中的每一个都具有大小 $n \times m$,其中

n和m是不同的整数,读取激活值的步骤包括读取m个激活值,并且对激活值执行乘法的步骤包括将m个激活值中的每一个乘以参数值的块中的相应的n个。

12.根据权利要求11所述的方法,其中,n至少是m的四倍大,或者反之亦然。

13.根据权利要求11或权利要求12所述的方法,其中,n是1或m是1。

14.根据权利要求1-13中任一项所述的方法,还包括:

用具有比对非零参数值中的至少一些进行编码的第二数据类型更低的精度的第一数据类型,对非零参数值之间的增量进行编码。

15.根据权利要求1-14中任一项所述的方法,其中,神经网络实时生成原始音频波形。

16.根据权利要求1-15中任一项所述的方法,其中,计算设备是手持计算设备。

17.根据权利要求1-16中任一项所述的方法,其中,计算设备安装在自动或半自动车辆上。

18.一种系统,包括:一个或多个计算机和存储指令的一个或多个存储设备,当所述指令由一个或多个计算机运行时,所述指令可操作以使所述一个或多个计算机执行权利要求1至17中任一项所述的方法。

19.一种编码有计算机程序的计算机存储介质,所述程序包括指令,当所述指令由数据处理装置运行时,所述指令可操作以使数据处理装置执行权利要求1至17中任一项所述的方法。

连续稀疏性模式神经网络

[0001] 背景

[0002] 本说明书涉及用于在处理硬件上实现神经网络的技术。

[0003] 神经网络是采用一层或多层非线性计算单元的来预测接收到的输入的输出的机器学习模型。一些神经网络除了输出层之外,还包括一个或多个隐藏层。在推断传递(inference pass)(即,作为整体由神经网络执行的、在某时间步长基于对神经网络的相应输入集生成输出集的处理)期间,每个隐藏层的输出用作网络中的下一层(即,下一个隐藏层或输出层)的输入。

[0004] 网络的每个层根据该层的相应参数集的当前值以及一个或多个变换操作生成下一层的输出。每个层的变换操作可以由运行安装的软件内核的处理设备来执行,该软件内核实现该层的变换操作。因此,描述为执行操作的层意味着实现该层的变换操作的处理设备执行变换操作。

[0005] 自然,随着神经网络的大小和复杂性增加,执行通过网络的推断传递所需的时间也会增加。一般,神经网络的推断速度与模型中的参数的数量成比例。这可以造成显著的处理瓶颈,该处理瓶颈可以限制可以在某些类型的处理硬件上实现的神经网络的类型。

[0006] 这对于具有重要的吞吐量要求、重要的延迟要求或两者兼有的神经网络来说尤其是挑战。例如,由于生成原始音频波形的的基本高吞吐量特性,实时音频生成神经网络带来了巨大的计算挑战。真实音频波形生成通常需要每秒生成数千个样本,例如每秒24,000个样本,例如,这可能需要每秒执行24,000次推断传递。这些性能要求使得难以直接在消费者硬件(例如手机和平板计算机)上实现高吞吐量神经网络,消费者硬件缺乏在数据中心中可用的高性能或并行处理硬件的种类。作为另一示例,为感测或控制自动和半自动车辆而运行的神经网络具有重要的延迟要求。换句话说,用于在真实世界驾驶条件下执行感测和控制的神经网络必须足够快,使得车辆能够在某些明确定义的时间段内(例如,在微秒或毫秒内)可靠地做出反应。

[0007] 提高推断速度的一种方法是生成具有稀疏连接的层。在这种情况下,具有层之间的稀疏连接的模型意味着相当大量可能的连接通过具有为零的值而被停用。例如,在每个具有1000个神经元的两个层之间,有1000x1000个可能的连接,或者100万个可能的连接。代替具有100万个连接,可以将连接缩减到可能的连接的小部分,例如,仅10,000或50,000个连接。

[0008] 通过将模型训练为在连接中具有方块稀疏性,可以进一步提高推断速度。通过对于参数矩阵的方形区域计算单个代表值(例如平均值或最大值),方块稀疏性对连接的稀疏性模式施加(impose)某种结构。方块稀疏性的确在模型参数中施加了某种结构,但是矩阵中方块的模式具有基本一致的随机性。

发明内容

[0009] 本说明书描述了用于通过将参数值排列为连续稀疏性模式来提高神经网络的推断速度并降低神经网络的数据存储要求的技术。训练模型为具有连续稀疏性涉及施加结构

化稀疏性模式,在该结构化稀疏性模式中,沿着参数矩阵中的相同维度,相当大量的非零值可能是连续的。这也意味着沿着参数矩阵中的相同维度,相当大量的零值可能是连续的。

[0010] 实验发现,具有连续稀疏性的神经网络与具有非结构化稀疏性(即,相同的稀疏性,但是非零值随机分布)的神经网络的性能大致相同,但是关键地,稀疏性模式可以用降低的数据存储容量来存储。例如,如果参数值在 $1 \times m$ 个块中(其中 m 是大于1的整数),与非结构化稀疏性相比,存储非零值的位置所需的数据减少了 m 倍。

[0011] 此外,如下所说明的,连续稀疏性允许神经网络比非结构化稀疏性以更有效的编码来实现,从而减少实现其要求的计算操作的数量。

[0012] 这两个优点在如下情况下都是关键的:尽管神经网络可能是在服务器上创建的,但是被诸如移动电话的具有低存储器容量和有限计算能力的客户端设备所利用。注意,当前移动CPU(中央处理单元)上的计算能力和存储器带宽分别比GPU(图形处理单元)上的小三个数量级和两个数量级。

[0013] 被认为足够大量的连续零或非零参数值取决于模型的基本(underlying)训练集和设计目标。例如,系统可以根据指定这些约束中的一个或多个的超参数(例如,在参数矩阵内以至少频率 f 具有至少长度 l 的连续非零参数的块的可能性)来训练模型。

[0014] 为了表示的简化,本说明书将使用术语“稀疏”来指代值的(二维)模式(即矩阵),其中零值的数量大于非零值的数量。具体地,零值的比例可以是至少80%、至少90%、或至少95%(诸如至少96.4%或至少97.8%),或者甚至至少99%。术语“稀疏性模式”是指指示参数矩阵的非零值在哪里的数据集。

[0015] 术语“块”用于指模式中非零值的(非正方形)矩形。通常,在不包括稀疏性模式的至少一个零值的情况下,块不能在任一维上扩展;因此,块可能是可以形成的非零值的最大连续块。然而,两个或更多这样的块相邻也是可能的。

[0016] 我们将使用术语“连续稀疏性”来指代实质上由多个零值组成的任何稀疏性模式,并且非零值的块具有其中一个维度远大于另一维度(例如,是另一维度的至少4倍大(例如,刚好4倍大),或者甚至至少16倍大(例如,刚好16倍大))的块大小,并且零值的数量大于非零值的数量。例如,块中的每一个可以具有相同的大小;也就是说,这些块是非零值的 $n \times m$ 块(其中 n 和 m 是整数), n 不同于 m 。

[0017] 具有连续稀疏性模式的参数矩阵可以具有其中非零参数值沿着参数矩阵的列是连续的(例如,在大小相等的块的情况下, n 大于 m)的列稀疏性,或者可以具有其中非零参数值沿着参数矩阵的行是连续的(例如,在大小相等的块的情况下, m 大于 n)的行稀疏性。注意,在行稀疏性的情况下,在参数矩阵的每行中可能不止一个块(例如,多个块被零值隔开);类似地,在列稀疏性的情况下,在参数矩阵的每列中可能不止一个块(例如,多个块被零值隔开)。在一些参数矩阵中,两个或更多个块可以是相邻的。此外,在行稀疏性情况下,多个列可以每个包含多个非零值;并且在列稀疏性的情况下,多个行可以每个包含多个非零值。

[0018] 这里,要理解,参数矩阵中的水平(行)方向(即其中块具有范围 m 的方向)表示层的不同输入,而参数矩阵中的垂直(列)方向(即其中块具有范围 n 的方向)对应于层的不同输出(或层的不同神经元)。例如,在一种情况下,层执行参数矩阵和表示对该层的输入的向量 x 之间的乘法(换句话说,在诸如 $R \cdot x$ 的操作中,网络参数的参数矩阵 R 乘以表示输入值的向

量 x),然后对乘法的结果中的每一个执行函数。在行稀疏性的情况下,生成给定输出值的过程可以包括相应行的每个块的乘法过程(其中各个输入值乘以块的参数值,然后对结果求和)、行的多个块上的结果的求和以及对求和结果的执行非线性函数。非零参数值的每个块的乘法过程可以在从存储读装置取该非零参数值的块之后执行,并且在从存储器读取任何其他参数值之前完成。

[0019] 例如,连续稀疏性模式可以具有其中维度之一为1的块,例如具有维度 $1 \times m$ 或 $n \times 1$ 。然而,可以使用任何 $n \times m$ 块大小,使得 n 远大于(例如四倍或十六倍) m ,反之亦然。

[0020] 可以实现本说明书中描述的主题的特定实施例,以便实现以下优点中的一个或多个。使用连续稀疏性模式允许在仅具有用于控制机动车辆功能的CPU和车载处理设备的能力较弱的处理硬件(例如消费者硬件,诸如手机(或其他移动设备)和平板计算机)上实现具有非常重要的吞吐量要求、非常重要的延迟要求或两者的神经网络。在这些计算环境中,神经网络可以在硬件上本地实现,这意味着网络可以在没有到数据中心的网络连接的情况下运行,而不需要使用专用并行处理硬件(如GPU),且也不需要专用硬件。例如,这允许在手持移动设备上执行实时音频波形生成,例如,用于文本到语音应用。并且还允许在有重要的延迟要求的环境中实现更精密和更复杂的神经网络。神经网络的另一应用是图像的自动生成。神经网络的另一应用是生成文本或任何其他合适的一维数据流。神经网络的进一步应用将是基于对由传感器接收到的信息进行编码并表征环境的网络输入,来控制诸如真实世界环境的环境中的代理(例如,机械设备)。在一个应用中,代理可以是运输设备,诸如自动或半自动车辆。

[0021] 如果对神经网络的输入是文本,那么在连续时间对神经网络的输入可以包括文本段的连续字符。

[0022] 可替代地,在其他应用中,在连续时间对神经网络的输入可以包括由传感器在相应连续时间获得的表征环境的数据(例如,音频数据)。

[0023] 在一个示例中,神经网络可以是循环神经网络(recurrent neural network, RNN),其中多个神经元每个在每个时间步长(例如,在每次推断传递中)生成取决于神经元在先前时间步长的状态(其又反过来取决于在先前推断传递中对神经元的(多个)输入)的输出。例如,神经网络的层可以接收在其给定时间步长的输入当中的、在至少一个先前时间步长(例如,包括紧接的先前时间步长)的神经层的输出。

附图说明

[0024] 图1包括参数矩阵的可视化,其示出了与随机稀疏性模式相比,连续稀疏性模式的结构。

[0025] 图2是使用连续稀疏性模式的示例系统200的图。

[0026] 图3是用于训练神经网络以具有参数矩阵的示例过程的流程图,该参数矩阵具有连续稀疏性模式。

[0027] 图4是用于在具有参数矩阵的神经网络上执行推断传递的示例过程的流程图,该参数矩阵具有连续稀疏性模式。

[0028] 各个附图中相同的附图标记和指定指示相同的元件。

具体实施方式

[0029] 图1包括参数矩阵的可视化,其示出了与随机稀疏性模式相比,连续稀疏性模式的结构。在图1中,黑色区域表示零参数值,并且白色区域表示非零参数值。

[0030] 随机稀疏性参数矩阵120具有随机稀疏性模式。因此,非零参数值实质上均匀地分布在矩阵中。

[0031] 连续稀疏性参数矩阵110具有连续稀疏性模式。因此,相当大量非零值沿着参数矩阵的行是连续的。

[0032] 可以在训练期间通过根据量值(magnitude)周期性地对参数的块进行排序(sort),并且将参数满足指示参数小的标准的块的所有参数归零来获得连续稀疏性。例如,块的参数的量值的和可以低于特定阈值。结果是比随机稀疏性模式具有更多结构的稀疏性模式,并且该稀疏性模式相对于方块稀疏性模式提高了推断速度。

[0033] 连续稀疏性可以用于以多种方式提高神经网络的推断速度。

[0034] 图2是使用连续稀疏性模式的示例系统200的图。系统200是可以实现本说明书中所描述的技术的系统的示例。在一个应用中,系统200可以在并行处理硬件上实现信号生成神经网络,例如生成音频、图像、文本或任何其他合适的一维数据流的神经网络。在这样做时,一些矩阵操作可以由所有可用的处理单元并行执行,而其他矩阵操作可以由可用的处理单元的子集并行执行。

[0035] 系统200包括由多个层定义的神经网络系统210。一般,每个层都具有各自的参数矩阵。因此,层N具有与层N+1分离的参数矩阵。每个层都由各自的计算系统实现。为简单起见,只有两个这样的层被示为由两个各自的计算系统实现,层N计算系统220和层N+1计算系统250。这些计算系统中的每一个都可以使用通用或专用计算硬件的任何适当组合来实现。例如,每个计算系统可以使用CPU、GPU或专用集成电路(例如,机器学习加速器)来实现。计算系统220和250可以由单独的计算硬件或相同的计算硬件来实现。

[0036] 层N计算系统220包括被配置为接收输入205(例如输入向量)的激活引擎222。输入向量的分量在本文中称为“激活值”。在层N是网络的第一层的情况下,激活值是对神经网络的输入。在一个应用中,输入向量可以由具有预定义顺序的元素组成(例如,这些元素可以指示文本段中的各个文本字符;或者指示描述连续时间的环境的数据)。在层N是神经网络的隐藏层的情况下,激活值是来自先前(即,N-1)层的输出。激活引擎222被配置为使用层N的输入205和参数矩阵来使用激活函数228生成多个输出值225。

[0037] 层N计算系统220包括连续稀疏性存储子系统224。连续稀疏性存储子系统224是被配置为获得层N的参数矩阵的子系统。参数矩阵由存储在连续稀疏性模式中的参数值组成,如下所述。在一种情况下,稀疏性模式是使得其完全由(i)零值和(ii)垂直扩展的非零值的块组成。换句话说,在这种情况下,连续稀疏性存储子系统224可以获得单个输入值的多个连续非零值。在一些实现方式中,连续稀疏性存储子系统224实现应用编程接口(application programming interface,API),其将接下来从层N的参数矩阵中读取的多个连续参数值作为变量(argument)。

[0038] 连续稀疏性存储子系统224可以是一个或多个层的单独存储子系统,或者连续稀疏性存储子系统224可以是整个神经网络系统210的全局存储子系统。在一些实现方式中,连续稀疏性存储子系统224由快速存储器(例如,SRAM)实现,该快速存储器对于执行激活函

数228的操作的处理组件而言是本地的。换句话说,层N的参数矩阵可以全部或部分地存储在激活引擎222本地的存储器中。

[0039] 层N计算系统220还包括后激活引擎226,该后激活引擎226被配置为接收由激活引擎222生成的输出值225,并生成被提供给神经网络系统210的下一层250的层N输出235。后激活引擎226可以对由激活函数228生成的输出225执行聚合 (aggregation) 操作,包括求和、池化 (pooling) 和soft-max操作。

[0040] 为了使用连续存储的参数值,激活引擎222可以从输入205读取每个非零值,然后从连续稀疏性存储子系统224读取多个连续存储的参数值。然后,激活引擎222可以生成单个输入值的多个输出值225,例如,通过将输入值乘以连续存储的参数值中的每一个,并将各个乘积或乘积的部分和提供给后激活处理引擎226。

[0041] 图3是用于训练神经网络以具有参数矩阵的示例过程的流程图,该参数矩阵具有连续稀疏性模式。该示例过程可以由可以实现神经网络系统的操作和训练更新的任何适当的计算系统来执行。可选地,图3的过程可以由第一计算机系统 (例如服务器) 执行以产生神经网络模型,并且该神经网络模型可以随后被传输到具有相对较低的计算能力和存储器容量的第二计算机系统 (例如移动电话或其他客户端设备)。为了方便起见,示例过程将被描述为由根据本说明书适当编程的一个或多个计算机的系统执行。

[0042] 系统执行训练迭代 (310)。一般,每次训练迭代在执行通过神经网络系统的一个或多个推断传递后,更新神经网络的参数矩阵的权重。在一些实现方式中,每次训练迭代执行在与推断传递的方向相反的方向上更新通过层的参数矩阵的权重的反向传播过程。

[0043] 系统确定是否更新层的参数稀疏性模式 (320);换句话说,重置参数值,使得参数矩阵满足连续稀疏性的约束。例如,系统可以在每次训练迭代之后更新稀疏性模式,或在每次特定数量的迭代发生是定期更新,或者在预定数量的迭代之后仅更新一次。

[0044] 如果稀疏性模式要被更新 (320),则系统可以采用将参数矩阵划分成多个 $n \times m$ 块,其中 n 和 m 是不同的整数。划分可以是预定义的 (即,块的位置可以是预定义的),并且在每次更新稀疏性模式时可以是相同的。该系统根据每个块中的参数的绝对量值,例如根据每个块中参数的绝对量值的和,对划分的参数的块排序 (分支到330);然后,系统清除那些参数满足指示参数小的阈值标准的块的所有值。例如,阈值标准可以是块的参数的绝对量值的和小于阈值 (340)。清除值可以包括分配在推断期间将被忽略的特殊值,例如,零或另一保留值 (为了简单起见,这里使用术语“零值”来涵盖两种可能性)。

[0045] 可以选择阈值,使得其值被清除的块的比例等于某一频率值 (这确定有多少权重保持在非零值;换句话说,频率值决定了结果参数矩阵的稀疏性)。在步骤330中获得的块的顺序使得容易确定这样的阈值。注意,在图3的方法的变型中,阈值可以是预定的,并且在这种情况下,可以省略排序步骤 (330)。

[0046] 步骤340具有蒸馏 (distill) 神经网络的效果,即显著地减少存储神经网络所需的数据量和操作神经网络所需的计算能力。这使得神经网络适合于由第二计算机系统使用。

[0047] 系统可以被配置为使用超参数 K 、 k 和 Z ,其中 K 指定迭代间隔 (即,在步骤320中,系统确定更新参数稀疏性模式的时间之间发生的迭代的数量), k 指定在每次更新时保留多少权重,并且 Z 表示 k 的最大值。在一些实现方式中, k 的值根据下式随着训练进度而增加:

$$[0048] \quad k = Z \left(1 - \left(1 - \frac{t-t_0}{S} \right)^3 \right),$$

[0049] 其中t是当前迭代数量，t₀是第一次更新发生时的迭代，而S是期望的更新步骤的总数。

[0050] 系统确定训练是否完成 (350)。如果没有，系统执行另一训练迭代 (分支到310)。

[0051] 如果训练完成 (350)，系统可选地执行一个或多个预推断优化 (360)。预推断优化每个都被设计为利用新训练的参数矩阵中的连续稀疏性模式。预推断优化可以包括低精度增量编码、稀疏性模式展开 (unroll) 和精确负载均衡。

[0052] 首先，系统可以使用低精度增量编码来对参数矩阵进行编码。连续稀疏性允许使用增量编码对参数矩阵进行额外的数据压缩，这提供了优于使用常规压缩稀疏行 (compressed sparse row, CSR) 格式的优势。特别地，系统可以使用具有比对参数进行编码的数据类型低的精度的数据类型对块之间的增量进行增量编码。例如，系统可以替代地使用8位整数，而不是使用16位整数来对增量进行编码。这减少了存储大量关于参数矩阵的元数据所需的存储空间，这使得整个模型的存储需求更小。

[0053] 此外，参数值的连续模式允许系统将稀疏性模式展开到神经网络内核本身的源代码中。换句话说，系统可以自动生成对代码中的稀疏性模式进行编码的神经网络内核的源代码。这允许系统跳过大量的存储器访问，否则需要从存储器中提取指定下一个非零参数值的位置的数据。相反，系统可以简单地生成直接对下一个非零值的位置进行编码的源代码语句 (statement)。通过减少分支和存储器停顿，这也加快了推断时间。

[0054] 参数值的连续模式也意味着可以用比随机稀疏性模式更少的源代码语句来对整个稀疏性模式进行编码。这意味着整个神经网络内核通常可以在存储器层次的更快部分这完全适配，例如L1高速缓存而不是主存储器。在一些实现方式中，整个内核足够小以适配在L1高速缓存中，同时编码参数矩阵足够小以适配在数据高速缓存中。

[0055] 考虑以下具有展现行稀疏性的连续稀疏性模式的示例8x8矩阵：

$$[0056] \quad P = \begin{bmatrix} 0 & 0 & 0 & a & b & c & d & e \\ 0 & d & e & f & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & h \\ 0 & i & j & k & l & m & n & o \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ q & r & s & t & 0 & 0 & 0 & 0 \\ 0 & u & v & w & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x & y & z \end{bmatrix}$$

[0057] 编译器可以展开循环读取这些参数值的程序代码，如表1的第二列中的伪代码所示。换句话说，源代码不具有for循环结构，而是源代码具有一系列从矩阵读取不同参数值的顺序读取指令。为了清楚起见，已经省略了神经网络内核的其他操作，包括乘法累加和其他操作。

行	未使用编码的展开	使用编码的展开
0	Read (A1, 1)	Read (A4, 5)
1	Read (A2, 1)	Read (B2, 3)
2	Read (A3, 1)	Read (C7, 2)
3	Read (A4, 1)	Read (D2, 7)
4	Read (A5, 1)	Read (F1, 4)
5	Read (A6, 1)	Read (G2, 3)
6	Read (A7, 1)	Read (H6, 3)
7	Read (A8, 1)	
[0058] 8	Read (B1, 1)	
9	Read (B2, 1)	
10	Read (B3, 1)	
11	Read (B4, 1)	
.	.	
.	.	
.	.	
61	Read (H6, 1)	
62	Read (H7, 1)	
63	Read (H8, 1)	

[0059] 表1

[0060] 从这个示例可以看出,展开仅生成64个读取指令,以读取参数矩阵的所有值。

[0061] 表1中的最后一列示出了使用编码的展开。特别地,因为参数矩阵的连续非零值的位置在编译时是已知的,所以系统可以生成直接对连续非零值的位置进行编码的读取指令。

[0062] 在本示例中,伪代码指令读取采用指定非零参数值的块的起始位置的第一变量,并且采用指定要读取多少连续值的第二变量。从上述的排序和清除过程中,这些是预先知道的。因此,系统可以仅用7条读取指令而不是64条读取指令,从示例参数矩阵中读取所有非零值。

[0063] 这种优化也使得代码的大小更小,这进一步使得模型的存储需求更小。

[0064] 最后,连续稀疏性模式允许系统在多计算元件设备上执行非常精确的负载均衡。对多个处理单元之中具有未知稀疏性模式的稀疏矩阵向量乘法的操作进行负载均衡是有挑战性的。

[0065] 但是在推断期间,模型在长时间段是固定的,并且这允许系统提前预计算出N个处理器之中的最佳负载均衡。例如,系统可以将基本相等数量的连续块或块参数值(即,在块之一内的参数值)分配给CPU、GPU或另一并行处理设备的每个处理核心,这使得比随机稀疏模式的负载均衡来更快地计算推断。作为另一示例,系统可以将基本相等数量的块分配给具有多个处理元件的专用处理器的每个处理元件。

[0066] 图4是用于在具有参数矩阵的神经网络上执行推断传递的示例过程的流程图,该参数矩阵具有连续稀疏性模式。该示例过程可以由被配置为实现神经网络系统的一个层的

操作的计算机系统执行。例如,图2的层N计算系统220可以执行示例过程。为了方便起见,示例过程将被描述为由根据本说明书适当编程的一个或多个计算机的系统执行。

[0067] 系统接收输入向量(410)。如上所述,输入向量可以是由神经网络系统的前一层生成的输出,或者输入向量可以是神经网络系统的初始输入的初始表示。

[0068] 系统确定是否已经处理了所有的输入向量(420)。

[0069] 如果没有,系统识别输入向量的下一个非零输入值(分支到430)。换句话说,系统识别还未被处理的输入向量的下一个非零输入值。对于具有多个处理器的设备,这可能会跳过由其他处理组件处理的输入值。

[0070] 系统读取输入值的多个连续非零参数值(440)。系统可以识别与输入值相对应的参数矩阵的行,并从该行读取连续参数值。如果系统使用上述使用位置编码的展开,系统可以在恒定时间内立即从参数矩阵中读取非零值,而无需搜索。

[0071] 系统为输入值和多个连续非零参数值生成多个输出值(450)。例如,系统可以将输入值乘以多个连续非零参数值中的每一个。然后,系统可以可选地生成要与其他输入值的结果相加的部分和。

[0072] 如果已经处理了所有的输入值(420),则过程结束(分支到结束)。

[0073] 因此,使用连续稀疏性模式执行推断传递加快推断传递中涉及的矩阵向量乘法。

[0074] 注意,上述技术可以在具有行稀疏性或列稀疏性的参数矩阵中赋予优势。当参数具有连续列稀疏性时,相同的输入向量值可以被从存储器中读取一次,并用于与多个参数值相乘。这可以在从存储器读取任何进一步的输入向量值之前完成。当参数具有连续行稀疏性时,可以用较少的存储器操作读取输入向量值。例如,1xn块大小意味着一个存储器操作可以用于从输入向量加载块所需的所有一个或多个值,而nx1块大小意味着虽然从输入向量加载值需要多个存储器操作,但是这些值可以被重复使用n次。大小为1xn的块具有输入激活的所有存储器加载都将是连续的优势性质,而大小为nx1的块意味着多个行可以共享相同的输入激活。

[0075] 本说明书中所描述的主题和功能操作的实施例可以实现在数字电子电路中、在有形体现的计算机软件或固件中、在包括本说明书中公开的结构及其结构等同物的计算机硬件中、或者在它们中的一个或多个的组合中。本说明书中所描述的主题的实施例可以实现为一个或多个计算机程序,即编码在有形非暂时性存储介质上的计算机程序指令的一个或多个模块,用于由数据处理设备执行或控制数据处理装置的操作。计算机存储介质可以是机器可读存储设备、机器可读存储基底、随机或串行存取存储器设备或者它们中的一个或多个的组合。可替代地或另外地,程序指令可以编码在人工生成的传播信号(例如,机器生成的电、光或电磁信号)上,生成该传播信号以对信息进行编码,用于传输到合适的接收器装置来供数据装置执行。

[0076] 术语“数据处理装置”是指数据处理硬件,并且包含用于处理数据的所有种类的装置、设备和机器,例如包括可编程处理器、计算机或多个处理器或计算机。该装置还可以是或还包括专用逻辑电路,例如FPGA(现场可编程门阵列)或ASIC(专用集成电路)。除了硬件之外,该装置可以可选地包括为计算机程序创建执行环境的代码,例如,构成处理器固件的代码、协议栈、数据库管理系统、操作系统或它们中的一个或多个的组合。

[0077] 还可以被称为或描述为程序、软件、软件应用、APP、模块、软件模块、脚本或代码的

计算机程序可以以任何形式的编程语言编写,包括编译或解释语言、或者声明性或过程性语言,并且计算机程序可以以任何形式部署,包括作为独立程序或作为模块、组件、子例程或适合在计算环境中使用的其他单元。程序可以,但不一定,对应于文件系统中的文件。程序可以存储在保存其他程序或数据的文件(例如,存储在标记语言文档中的一个或多个脚本)的部分中,存储在专用于所讨论的程序的单个文件中,或者存储在多个协调文件(例如存储一个或多个模块、子程序或代码的部分的文件)中。计算机程序可以部署为在位于一个站点或跨多个站点分布并通过数据通信网络互连的一个或多个计算机上运行。

[0078] 对于一个或多个计算机的系统被配置为执行特定的操作或动作意味着该系统已经在其上安装了软件、固件、硬件或它们的组合,这些软件、固件、硬件或它们的组合在操作中使得该系统执行这些操作或动作。对于一个或多个计算机程序被配置为执行特定的操作或动作意味着该一个或多个程序包括当由数据处理装置运行时使该装置执行这些操作或动作的指令。

[0079] 本说明书中所描述的过程和逻辑流程可以由运行一个或多个计算机程序的一个或多个可编程计算机来执行,以通过对输入数据进行操作并生成输出来执行功能。过程和逻辑流程也可以由专用逻辑电路,例如,FPGA或ASIC,或者由专用逻辑电路和一个或多个编程的计算机的组合来执行。

[0080] 适于计算机程序的运行的计算机可以基于通用或专用微处理器或两者,或者任何其他类型的中央处理单元。一般,中央处理单元将从只读存储器或随机存取存储器或两者接收指令和数据。计算机的基本元件是用于执行(perform)或运行(execute)指令的中央处理单元和用于存储指令和数据的一个或多个存储器设备。中央处理单元和存储器可以由专用逻辑电路补充或并入专用逻辑电路。一般,计算机还将包括一个或多个用于存储数据的大容量存储设备(例如磁盘、磁光盘或光盘),或者可操作地耦合为一个或多个大容量存储设备,以从用于存储数据的一个或多个大容量存储设备接收数据或向其传送数据,或者两者。然而,计算机不需要具有这样的设备。此外,计算机可以嵌入到另一设备中,例如移动电话、个人数字助理(PDA)、移动音频或视频播放器、游戏控制台、全球定位系统(GPS)接收器或便携式存储设备,例如通用串行总线(USB)闪存驱动器,仅举几个示例。

[0081] 适于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储器设备,例如,包括半导体存储器设备,例如,EPROM、EEPROM和闪存设备;磁盘,例如内部硬盘或可移动磁盘;磁光盘;以及CD-ROM和DVD-ROM盘。

[0082] 为了提供与用户的交互,本说明书中所描述的主题的实施例可以在具有用于向用户显示信息的显示器设备(例如,CRT(阴极射线管)或LCD(液晶显示器))以及键盘和定点设备(例如鼠标、轨迹球或存在敏感(presence sensitive)显示器或者用户可以通过其向计算机提供输入的其他表面)的计算机上实现。也可以使用其他类型的设备来提供与用户的交互;例如,提供给用户的反馈可以是任何形式的感觉反馈,例如视觉反馈、听觉反馈或触觉反馈;并且可以以任何形式接收来自用户的输入,包括声音、语音或触觉输入。此外,计算机可以通过向用户使用的设备发送文档和从用户使用的设备接收文档来与用户交互;例如,通过响应于从用户设备上的网络浏览器接收到的请求,将网页发送到网络浏览器。此外,计算机可以通过向个人设备(例如智能手机)发送文本消息或其他形式的消息、运行消息传递应用、以及接收来自用户的响应消息来与用户交互。

- [0083] 除了上述实施例,以下实施例也是创新的:
- [0084] 实施例是一种方法,包括:
- [0085] 接收神经网络的一个或多个参数矩阵,其中至少第一参数矩阵具有以连续稀疏性模式存储的参数值,连续稀疏性模式是由零值和非零值的多个非方形的矩形块组成的二维模式,零值的数量大于非零值的数量;
- [0086] 将第一参数矩阵存储在与计算设备相关联的存储装置中;
- [0087] 接收激活值的输入向量;以及
- [0088] 由计算设备执行神经网络的推断传递以生成输出向量,包括执行包括以下的操作:
- [0089] 从与计算设备相关联的存储装置读取来自输入向量的一个或多个激活值,
- [0090] 从与计算设备相关联的存储装置读取非零参数值的块,以及
- [0091] 将一个或多个激活值中的每一个乘以非零参数值的块中的一个或多个。
- [0092] 实施例2是根据实施例1,其中推断传递包括以下操作:
- [0093] 从与计算设备相关联的存储装置读取来自输入向量的第一激活值,
- [0094] 从与计算设备相关联的存储装置读取多个连续非零参数值,以及
- [0095] 通过包括在从输入向量读取另一激活值之前将第一激活值乘以多个连续非零参数值的过程,生成输出向量的多个输出值。
- [0096] 实施例3是根据实施例1的方法,其中推断传递包括以下操作:
- [0097] 从与计算设备相关联的存储装置读取来自输入向量的多个激活值,
- [0098] 从与计算设备相关联的存储装置读取多个非零参数值的块,以及
- [0099] 通过包括在读取另一参数值之前将多个非零激活值中的每一个乘以多个非零参数值的块中的相应的一个或多个的过程,生成输出向量的输出值。
- [0100] 实施例4是任一前述实施例的方法,还包括:
- [0101] 接收表示计算设备的处理器的数量的值;
- [0102] 计算第一参数矩阵的非零值的块的数量;以及
- [0103] 将基本相等数量的块或块参数值分配给计算设备的每个处理器。
- [0104] 实施例5是实施例4的方法,其中计算设备的处理器是中央处理单元(CPU)的处理核心。
- [0105] 实施例6是实施例4的方法,其中计算设备的处理器是专用处理器的处理元件。
- [0106] 实施例7是实施例1-6中的任一的方法,还包括:
- [0107] 生成对第一参数矩阵中的接下来的非零值的位置进行编码的多个源代码语句(statement),
- [0108] 其中由计算设备执行推断传递包括运行生成的多个源代码语句。
- [0109] 实施例8是实施例7的方法,其中源代码语句省略读取第一参数矩阵中的接下来的非零值的位置的一个或多个存储器访问指令。
- [0110] 实施例9是实施例1-8的任一的方法,还包括:
- [0111] 训练神经网络包括执行包括以下的操作:
- [0112] 在一个或多个块的参数值的量值满足阈值标准的确定之后,将零分配给参数值的一个或多个块。

[0113] 实施例10是根据实施例9的方法,包括在分配步骤之前,根据每个块的参数值的函数对参数值的块进行排序的步骤。

[0114] 实施例11是实施例1-10的任一的方法,其中块的每一个都具有大小 $n \times m$,其中 n 和 m 是不同的整数,读取激活值的步骤包括读取 m 个激活值,并且对激活值执行乘法的步骤包括将 m 个激活值中的每一个乘以参数值的块中的相应的 n 个。

[0115] 实施例12是实施例11的方法,其中 n 至少是 m 的四倍大,或者反之亦然。

[0116] 实施例13是实施例11或实施例12的方法,其中 n 是1或 m 是1。

[0117] 实施例14是实施例1-13的任一的方法,还包括:

[0118] 用具有比对非零参数值中的至少一些进行编码的第二数据类型更低的精度的第一数据类型,对非零参数值之间的增量进行编码。

[0119] 实施例15是实施例1-14的任一的方法,其中神经网络实时生成原始音频波形。

[0120] 实施例16是实施例1-15的任一的方法,其中计算设备是手持计算设备。

[0121] 实施例17是实施例1-16的任一的方法,其中计算设备安装在自动或半自动车辆上。

[0122] 实施例18是一种系统,包括:一个或多个计算机和存储指令的一个或多个存储设备,当指令由一个或多个计算机运行时,指令可操作以使一个或多个计算机执行实施例1至15中的任一的方法。

[0123] 实施例19是一种编码有计算机程序的计算机存储介质,程序包括指令,当指令由数据处理装置运行时,指令可操作以使数据处理装置执行实施例1至15中的任一的方法。

[0124] 尽管本说明书包含多个具体的实现方式细节,但这些不应被解释为对任何发明的范围或对可能要求保护的范围的限制,而是对特定于特定发明的特定实施例的特征的描述。本说明书中在单独的实施例的上下文中描述的某些特征也可以在单个实施例中组合实现。相反,在单个实施例的上下文中描述的各种特征也可以在多个实施例中单独地或以任何合适的子组合来实现。此外,尽管特征可以在上面被描述为以某些组合起作用,并且甚至最初被如此要求保护,但是在一些情况下来自所要求保护的组合的一个或多个特征可以从该组合删除,并且所要求保护的组合可以针对于子组合或子组合的变型。

[0125] 类似地,尽管在附图中以特定的顺序描述了操作,但是这不应该被理解为要求应当以所示的特定顺序或顺序地执行这些操作,或者应当执行所有示出的操作,以获得期望的结果。在某些情况下,多任务和并行处理可能是有利的。此外,上述实施例中的各种系统模块和组件的分离不应被理解为在所有实施例中都需要这样的分离,并且应当理解,所描述的程序组件和系统一般可以一起集成在单个软件产品中或者封装到多个软件产品中。

[0126] 已经描述了主题的特定实施例。其他实施例在以下权利要求的范围内。例如,权利要求中列举的动作可以以不同的顺序执行,并且仍然获得期望的结果。作为一个示例,附图中描述的过程不一定必须需要所示的特定顺序或依次顺序来获得期望的结果。在某些情况下,多任务和并行处理可能是有利的。

[0127] 在另一示例中,尽管图2描述了前馈神经网络,其中,在每个推断传递中,每个神经元生成仅是其输入和相应的网络参数的函数的输出,但是在变型中神经层中的一个或多个可能是循环神经层,其中,神经层的输出取决于在上一推断传递中神经层的状态。在一个示例中,层 N 输入205可以包括在多个先前时间中的一个的层 N 输出235,例如在先前推断传递

期间的层N输出235。在构成第一隐藏层的层N计算系统220的情况下,这可以添加到对输入到神经网络的数据。在作为神经网络的之后的隐藏层的层N计算机系统220的情况下,这可以添加到对前一隐藏层的输出。

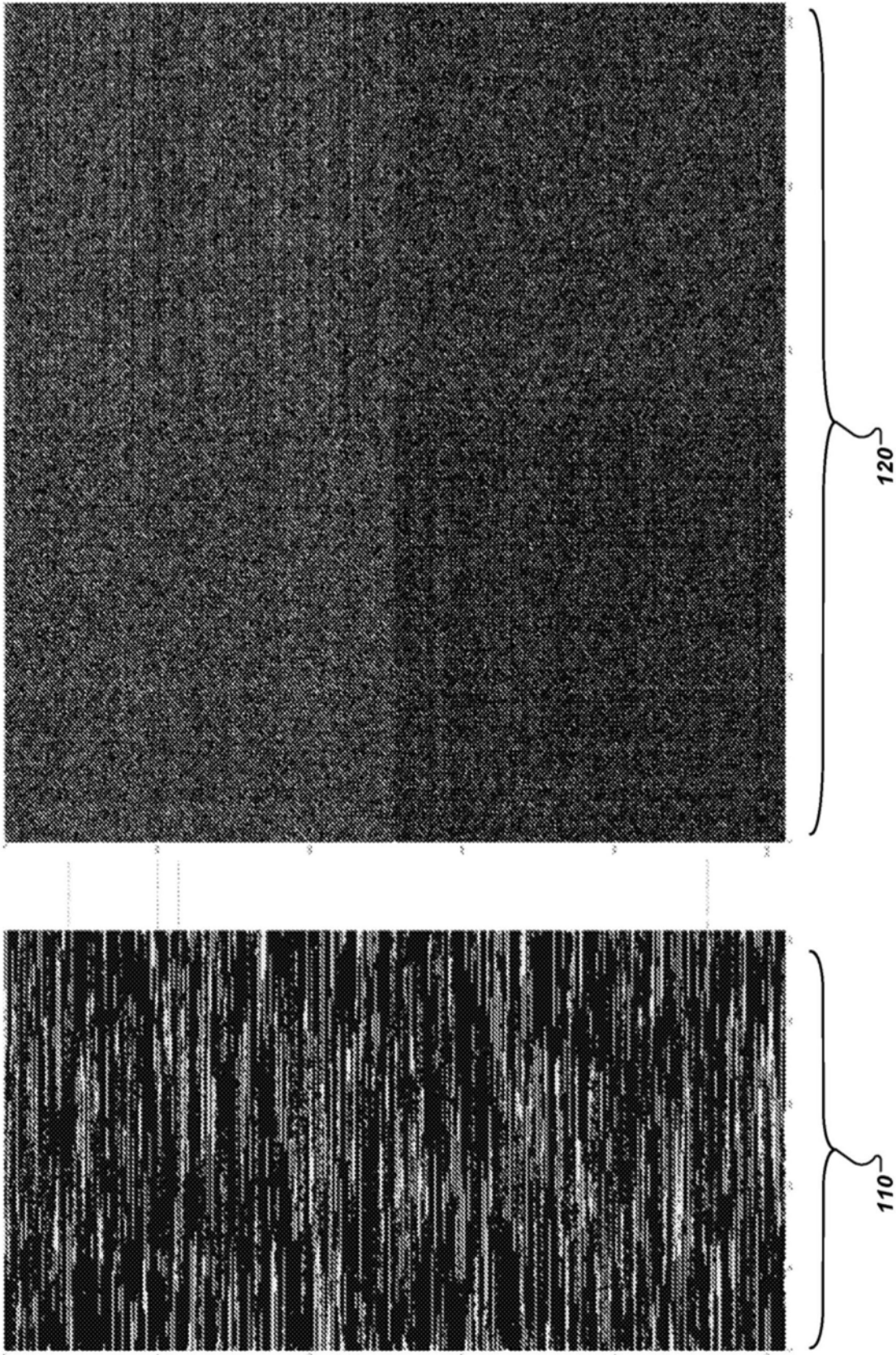


图1

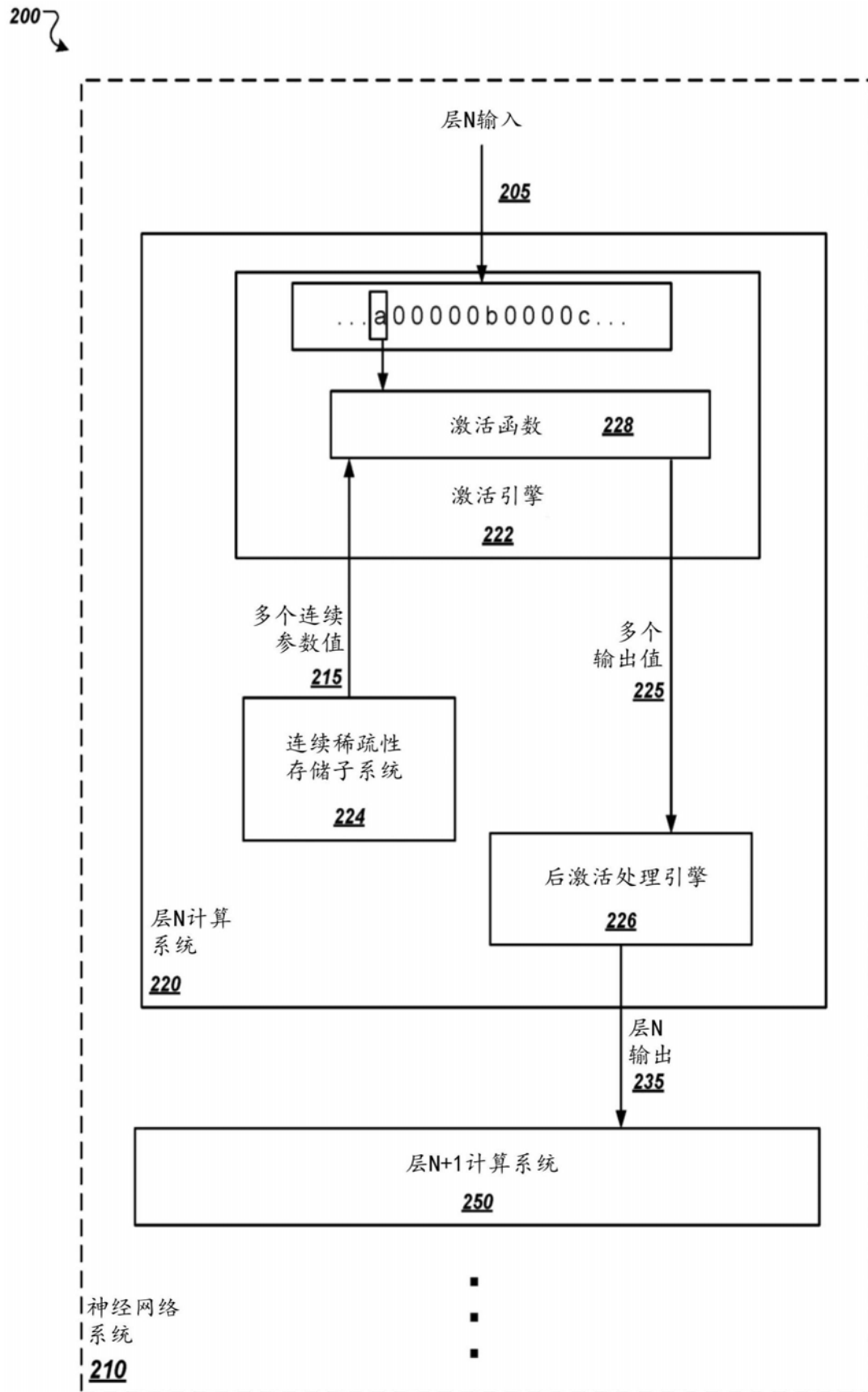


图2

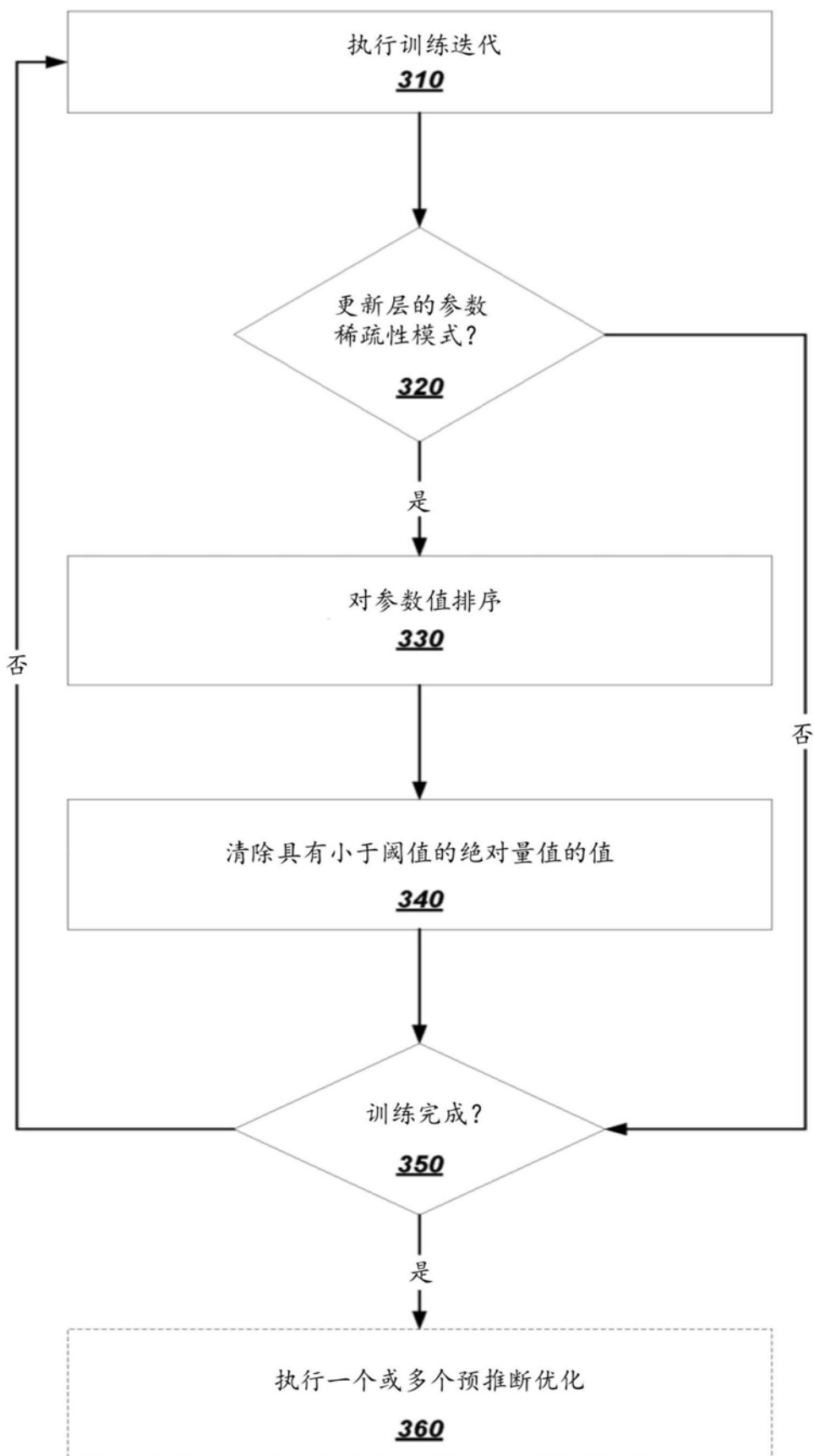


图3

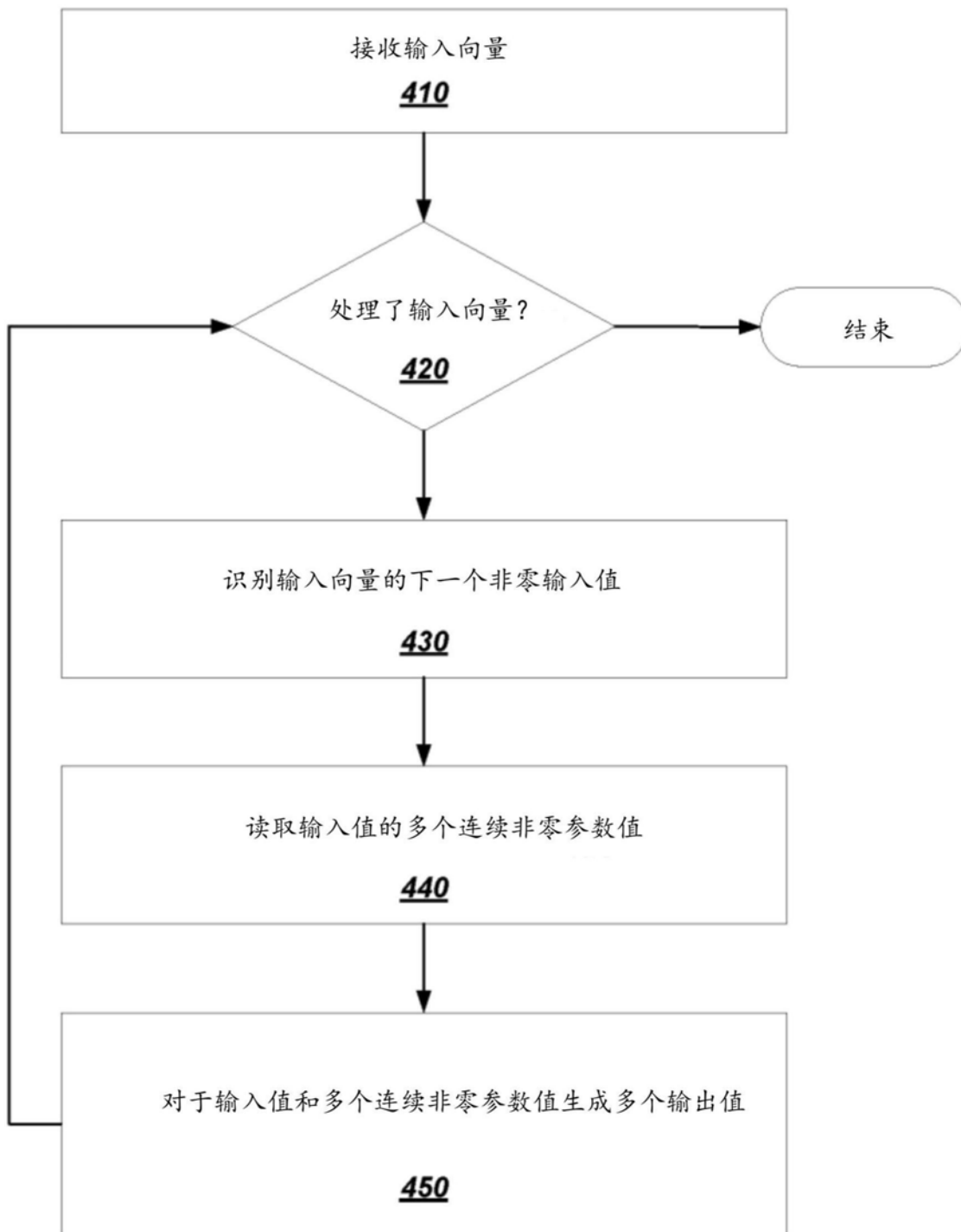


图4