



(12) 发明专利



(10) 授权公告号 CN 113785271 B

(45) 授权公告日 2025. 02. 25

(21) 申请号 202080033150.6
(22) 申请日 2020.03.02
(65) 同一申请的已公布的文献号
申请公布号 CN 113785271 A
(43) 申请公布日 2021.12.10
(30) 优先权数据
16/403,083 2019.05.03 US
16/421,225 2019.05.23 US
(85) PCT国际申请进入国家阶段日
2021.11.02
(86) PCT国际申请的申请数据
PCT/US2020/020622 2020.03.02
(87) PCT国际申请的公布数据
W02020/226732 EN 2020.11.12

(73) 专利权人 特斯拉公司
地址 美国得克萨斯州
(72) 发明人 D·达斯·萨玛 W·麦吉
E·塔尔佩斯
(74) 专利代理机构 北京市金杜律师事务所
11256
专利代理师 马明月
(51) Int.Cl.
G06F 9/30 (2018.01)
G06N 3/045 (2023.01)
G06N 20/00 (2019.01)
(56) 对比文件
US 2017221176 A1, 2017.08.03
US 2018322382 A1, 2018.11.08
审查员 任丽霞

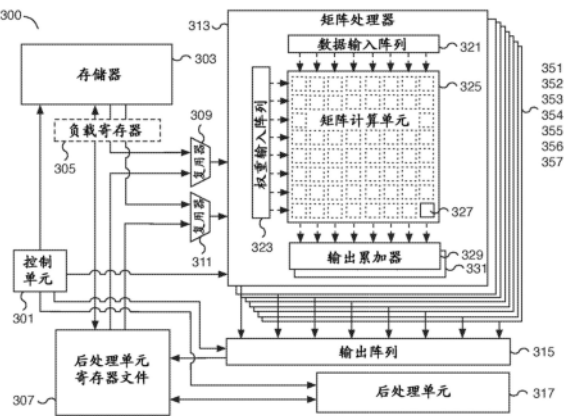
权利要求书2页 说明书21页 附图5页

(54) 发明名称

具有指定偏差浮点操作数的矩阵乘法指令
的系统和方法

(57) 摘要

一种微处理器系统包括矩阵计算单元和控制单元。矩阵计算单元包括多个处理元件。控制单元被配置为向矩阵计算单元提供矩阵处理器指令。矩阵处理器指令指定使用第一浮点表示格式格式化的浮点操作数。矩阵计算单元累加使用浮点操作数计算的中间结果值。中间结果值采用第二浮点表示格式。



1. 一种微处理器系统,包括:
矩阵计算单元,包括一个或多个处理元件;以及
控制单元,被配置为向所述矩阵计算单元提供矩阵处理器指令,其中所述矩阵处理器指令指定使用指数格式化的浮点操作数,所述指数使用指定偏置进行偏置,其中所述浮点操作数是矩阵,
其中所述指定偏置能够通过所述控制单元使用所述矩阵处理器指令进行配置,并且其中所述矩阵处理器指令从浮点格式组中为所述浮点操作数指定浮点数格式。
2. 根据权利要求1所述的系统,其中所述矩阵的每个元素使用8位浮点格式。
3. 根据权利要求2所述的系统,其中所述8位浮点格式为符号位分配1位,为指数字段分配4位,并且为尾数字段分配3位。
4. 根据权利要求2所述的系统,其中所述8位浮点格式为符号位分配1位,为指数字段分配5位,并且为尾数字段分配2位。
5. 根据权利要求1所述的系统,其中所述浮点格式组中的每个浮点格式利用相同的总位数来表示浮点数,并且利用不同位数用于所述浮点数的尾数字段。
6. 根据权利要求1至5中任一项所述的系统,其中所述指定偏置使用寄存器参量来配置。
7. 根据权利要求1至5中任一项所述的系统,其中所述指定偏置选自非连续的预先确定的浮点指数偏置的集合。
8. 根据权利要求1至5中任一项所述的系统,其中所述指定偏置能够经由所述矩阵处理器指令重新配置。
9. 根据权利要求1至5中任一项所述的系统,其中所述一个或多个处理元件中的每个处理元件包括浮点乘法器和累加器。
10. 根据权利要求1至5中任一项所述的系统,其中所述一个或多个处理元件中的每个处理元件被配置为与其他处理元件并行执行浮点乘法运算。
11. 根据权利要求1至5中任一项所述的系统,其中所述矩阵处理器指令指定用于存储所述矩阵计算单元的中间结果的指定累加器。
12. 根据权利要求1至5中任一项所述的系统,还包括:
后处理单元,所述控制单元被配置为向所述后处理单元提供后处理指令;以及
后处理寄存器文件,其中所述后处理指令指定存储在所述后处理寄存器文件中的后处理单元操作数,
其中所述矩阵计算单元被配置为接收所述矩阵处理器指令。
13. 根据权利要求12所述的系统,其中所述后处理单元是向量计算单元。
14. 根据权利要求12所述的系统,其中所述后处理单元操作数是向量操作数,并且所述后处理指令为所述后处理单元操作数的每个向量元素指定数据大小。
15. 一种用于矩阵计算的方法,包括:
从控制单元接收矩阵处理器指令,其中所述矩阵处理器指令指定使用指数格式化的浮点操作数,所述指数使用指定偏置进行偏置,其中所述指定偏置能够通过所述控制单元使用所述矩阵处理器指令进行配置;
发出所述浮点操作数的数据值的一个或多个读取指令;

接收所述浮点操作数的所述数据值;以及

将一个或多个接收的数据值加载到矩阵计算单元中,其中所述浮点操作数是矩阵,并且其中所述矩阵处理器指令从浮点格式组中为所述浮点操作数指定浮点数格式。

具有指定偏差浮点操作数的矩阵乘法指令的系统和方法

[0001] 相关申请的交叉引用

[0002] 本申请要求于2019年5月3日提交的题为“具有混合数据格式的可扩展矩阵节点引擎的数据路径”的美国专利申请号16/403083的优先权,并且还要求于2019年5月23日提交的题为“具有可配置数据格式的可扩展矩阵节点”的美国专利申请号16/421225的优先权,其公开内容通过引用整体并入本文。

背景技术

[0003] 机器学习训练是数据和计算密集型操作。这个过程既乏味又耗时,需要大量相关训练数据和计算资源来处理它。而且,数据和计算资源只会随着正在被解决的问题的复杂性而增加。为了训练机器学习模型,高功率CPU使用训练数据执行复杂矩阵运算以确定适当权重。为了提高训练速度,图形处理单元(GPU)用作传统CPU的备选或补充。GPU允许并行化训练中的一些训练并且帮助优化某些数学运算。然而,传统上讲,GPU针对处理图形问题而设计,诸如将三维世界渲染到二维显示器上。当应用于机器学习时,GPU可能需要大量计算能力来满足它们所提供的计算能力。而且,GPU所使用的数据格式和数据管道针对图形处理而设计,而非用于训练机器学习模型。因此,需要一种功能强大、计算能力强且能效高的机器学习训练系统。这种系统应当支持高数据带宽,以显着增加可以处理的训练数据量。而且,应当针对训练数据和所得机器学习模型优化数据格式和数据管道。

附图说明

[0004] 在以下具体实施方式和附图中公开了本发明的各个实施例。

[0005] 图1是图示了用于训练机器学习模型的过程的实施例的流程图。

[0006] 图2是图示了用于训练机器学习模型的系统的实施例的框图。

[0007] 图3是图示了用于执行矩阵计算的节点引擎的实施例的框图。

[0008] 图4是图示了8位浮点格式的实施例的框图。

[0009] 图5是图示了21位浮点格式的实施例的框图。

[0010] 图6是图示了用于执行矩阵计算的过程的实施例的流程图。

[0011] 图7是图示了用于执行矩阵计算的过程的实施例的流程图。

[0012] 图8是图示了用于执行多个交错矩阵计算的过程的实施例的流程图。

具体实施方式

[0013] 本发明可以以多种方式实现,包括作为一种过程、装置、系统、物质组成、计算机可读存储介质上体现的计算机程序产品、和/或一种处理器,诸如被配置为执行耦合到处理器的存储器上存储和/或由其提供的指令的处理器。在本说明书中,这些实现方式或本发明可以采用的任何其他形式可以称为技术。一般而言,在本发明的范围内可以更改所公开的过程的步骤的次序。除非另有说明,被描述为被配置为执行任务的诸如处理器或存储器之类的部件可以被实现为被临时配置为在给定时间执行任务的通用部件或被制造为执行任务

的特定部件。如本文中所使用的,术语‘处理器’是指被配置为处理诸如计算机程序指令之类的数据的一个或多个设备、电路和/或处理核。

[0014] 下文提供了本发明的一个或多个实施例的详细描述以及说明本发明的原理的附图。结合这些实施例对本发明进行描述,但本发明不限于任何实施例。本发明的范围仅由权利要求限定,并且本发明涵盖许多替代、修改和等同物。在以下描述中对许多具体细节进行阐述以提供对本发明的透彻理解。这些细节是为了举例的目的而提供的,并且本发明可以根据权利要求来实践,而无需这些特定细节中的一些或全部特定细节。为了清楚起见,与本发明相关的技术领域中的已知技术材料尚未详细描述,以免不必要地混淆本发明。

[0015] 公开了一种具有多个矩阵处理器和可配置数据格式的可扩展(scalable)节点引擎。作为机器学习模型的训练平台的核心部件,节点引擎可以在网络中被布置为对机器学习模型执行训练。随着计算和数据需求的增加,可以增加网络中节点引擎的数目来处置附加要求。与针对类似工作负载委派任务的传统CPU和GPU相比较,所公开的节点引擎在每平方毫米每瓦的性能方面非常高效。节点引擎架构部分地通过优化用于机器学习工作负载的数据格式和数据路径来实现这种性能改进。例如,节点引擎包括多个矩阵处理器,每个矩阵处理器可以交错多个矩阵运算。具有八个矩阵处理器的组的节点引擎可以在每个周期计算矩阵乘法的结果。当停止等待针对第一相关矩阵运算集合的数据时,每个矩阵处理器可以交错第二相关矩阵运算集合,以利用否则将空闲的计算资源。在一些实施例中,使用低位浮点格式存储矩阵操作数并且使用高位浮点格式计算中间结果和最终结果。低位格式提高了矩阵处理器的读取数据带宽,而高位格式保留了矩阵结果的准确性和精确性,例如,通过防止损失量化结果的准确性。可以选择不同的可配置数据格式来指定不同的数据格式配置,例如,以改变分配给尾数(mantissa)字段和指数字段的位数。这允许基于用于特定机器学习任务的特定矩阵运算而优化数据格式。附加地,数据格式可以包括用于偏置指数的可配置偏置。这改进了指数的范围并且允许利用较大范围。

[0016] 在一些实施例中,节点引擎布置在网状网络中。每个节点引擎包括控制单元、存储器、寄存器、多个矩阵处理器、以及诸如向量计算单元之类的后处理单元。控制单元可以处理定制指令,这些定制指令包括涉及多个矩阵处理器中的一个矩阵处理器的矩阵计算指令,并且用于同步不同矩阵处理器与节点引擎之间的结果。矩阵结果可以存储在寄存器文件中,并且通过后处理单元使用向量运算进行处理。运行节点引擎的软件能够进行大型矩阵运算并且细分问题。问题的不同子部件可能分布到不同节点引擎和每个节点引擎的不同矩阵处理器。例如,可以对两个大矩阵进行切片,使得每个切片都优化到矩阵处理器的矩阵大小。然后,切片可以分配到不同节点引擎的不同矩阵处理器,其中对切片执行矩阵乘法。可以组合每个矩阵乘法的结果来计算原始较大矩阵的乘法结果。

[0017] 在一些实施例中,微处理器系统包括矩阵计算单元和控制单元。矩阵计算单元包括一个或多个处理元件。例如,矩阵计算单元包括用于确定来自两个操作数的两个元素的计算结果的计算单元矩阵。 8×8 矩阵计算单元包括64个计算单元。同样, $M \times N$ 矩阵计算单元包括 $M \times N$ 个计算单元。矩阵计算单元是经由控制单元控制的矩阵处理器的一部分。在一些实施例中,控制单元被配置为向矩阵计算单元提供矩阵处理器指令。例如,控制单元向矩阵处理器提供矩阵乘法指令,以供矩阵计算单元执行。矩阵处理器指令指定浮点操作数,该浮点操作数使用指数格式化,该指数已经使用指定可配置偏置进行了偏置。例如,矩阵乘法指

令指定两个浮点矩阵操作数。矩阵操作数的每个元素都使用特定浮点格式和可配置指数偏置进行格式化。与矩阵操作数一起,矩阵处理器指令指定矩阵元素所使用的浮点格式(诸如为符号位分配1位、为指数分配4位、为尾数分配3位的格式)以及特定指数偏置。在各个实施例中,偏置能够通过指定对应于指数偏置的值来配置。在一些实施例中,偏置可重新配置。例如,矩阵指令可以指定用于重新配置可配置偏置的新偏置。在一些实施例中,浮点格式支持非规则数(denormal number)以增加可以表示的值的数目。

[0018] 在一些实施例中,矩阵处理器指令指定使用第一浮点表示格式格式化的浮点操作数。例如,该指令指定8位浮点格式,该8位浮点格式为指数分配4位,为尾数分配3位,以及单个符号位。所指定的格式用于矩阵操作数的元素。可以选择格式以增加进入矩阵处理器的矩阵计算单元的数据带宽。矩阵计算单元累加使用浮点操作数计算的中间结果值,并且该中间结果值采用第二浮点表示格式。例如,中间结果使用不同浮点格式,诸如21位浮点格式。作为另一示例,中间结果可以使用不同浮点格式,诸如27位或另一适当浮点格式。可以选择专用于中间结果的位数,以防止量化结果时的准确性损失。可以选择使用大量位来表示中间结果的格式,以防止使用第一浮点格式可能导致的上溢(overflow)错误和/或下溢(underflow)错误。矩阵计算单元输出累加中间结果作为以第三浮点表示格式格式化的输出。例如,多个累加中间结果可以作为矩阵结果从矩阵处理器中移出。可以使用与矩阵处理器所连接到的总线兼容的第三格式来输出结果。例如,节点引擎可以利用宽度为64个字节的内部总线。中间累加结果可以作为16位浮点值从矩阵计算单元输出,允许针对每个移动指令从矩阵处理器移动32个元素。具有64个元素的累加结果可以使用两个移动指令从矩阵处理器移动到节点引擎的寄存器文件中,其中每个指令移动32个元素。上移指令可以用于移动高位32个元素(例如,元素32至63),而下移指令可以用于移动低位32个元素(例如,元素0至31)。在一些实施例中,移动指令为非破坏性指令,并且当将值从矩阵处理器的源累加器移动到矩阵处理器外部的存储器位置(诸如输出阵列或寄存器)时,不会清除源累加器的内容。

[0019] 图1是图示了用于训练机器学习模型的过程的实施例的流程图。例如,图1的过程可以用于训练自动驾驶或驾驶员辅助驾驶的模型。在车辆诸如由人类驾驶员自动驾驶或由人工驾驶和辅助驾驶混合驾驶时,可以捕获驾驶数据。所捕获的数据准备作为训练数据,并且用于训练新机器学习模型以改善驾驶体验。新驾驶体验可以在诸如安全性、效率(功率、时间等)、舒适性、性能、便利性等之类的领域进行改进。一旦新模型经过训练和验证,新训练的模型都会部署到其被一个或多个机器学习网络用于实现改进驾驶特征和功能的车辆。新特征其中可以包括自动驾驶特征或辅助驾驶特征,诸如自主变道、自主车道并入高速公路、自主驶出高速公路、改进对障碍物和道路场景的检测、以及基于自主导航的驾驶等。在各个实施例中,机器学习模型可以在利用多个节点引擎并且其中每个节点引擎包括多个矩阵处理器和可配置数据格式的训练平台上进行训练。

[0020] 在101处,捕获数据用于机器学习训练。在一些实施例中,在车辆由人类、自动驾驶系统或两者驾驶时,捕获了与车辆驾驶相对应的数据。所捕获的车辆驾驶条件的数据可以包括图像传感器数据、车辆操作参数(例如,速度、转向等)、车辆类型信息(例如,左侧驾驶、右侧驾驶、车辆型号等)、自动驾驶是否启用、自动驾驶最后一次脱离的时间、所检测的障碍物、驾驶条件等。可以在没有干扰车辆驾驶并且无需驾驶员辅助的情况下被动地捕获数据。

[0021] 在各个实施例中,车辆可以配备有不同布置的传感器以捕获不同形式的数据。在一些实施例中,传感器数据可以是视觉数据、超声波数据、LiDAR数据、或其他适当传感器数据。例如,从高动态范围前向相机捕获图像。作为另一示例,从侧向超声波传感器捕获超声波数据。在一些实施例中,车辆附有用于捕获数据的多个传感器。例如,在一些实施例中,八个环绕相机附着到车辆并且提供车辆周围的360度可见度,其中范围可达250米。不同布置的相机传感器可以包括宽前相机、窄前相机、后视相机、前视侧相机和/或后视侧相机。在一些实施例中,附加超声波传感器和/或雷达传感器用于捕获周围细节。例如,十二个超声波传感器可以附着到车辆以检测硬物体和软物体两者。还可以使用附加前向雷达来捕获周围环境的数据。在各个实施例中,尽管有大雨、雾、灰尘和其他车辆,雷达传感器仍然能够捕获周围细节。各个传感器用于捕获车辆周围的环境,并且存储所捕获的数据以供考虑作为深度学习网络的训练数据。

[0022] 一旦被捕获,从一个或多个车辆捕获的数据被传输到机器学习训练平台。例如,具有无线连接性(诸如蜂窝或WiFi连接)的车辆可以将数据无线地传送到机器学习训练平台。作为另一选项,当技术人员正在维修车辆时,可以从车辆下载所捕获的数据。在各个实施例中,从多个车辆(诸如车辆车队)捕获的数据在机器学习训练平台处聚集并且用作训练数据源的至少一个训练数据源。

[0023] 在103处,准备所捕获的用于训练机器学习模型的数据。在101处,从车辆捕获的数据准备作为训练数据。在一些场景中,数据被分为训练数据和验证数据。准备数据可以包括选择(或剔除)所捕获的数据以标识特别好的训练数据。在一些实施例中,数据被注释以标识用于训练的特征。例如,可以对车道标记、交通灯、交通标志、车辆、行人等进行注释,以增强训练数据作为数据准备的一部分的有用性。作为另一示例,数据可以被转换为不同的格式或作为准备过程的一部分进行预先处理。在一些实施例中,数据可以从源数据格式转换为与矩阵处理器兼容的格式。例如,作为固定点数据捕获的数据可以转换为浮点数据以提高精确性。

[0024] 在105处,训练机器学习模型。使用在103处准备的训练数据,训练一个或多个机器学习模型。训练可以利用训练数据集和验证数据集两者。在一些实施例中,训练利用由多个节点引擎组成并且其中每个节点引擎包括多个矩阵处理器的机器学习平台。通过利用例如组织成网格或另一适当架构的多个节点引擎,可以并行化复杂机器学习训练问题,并且更快且更有效地执行该问题。同样,由于每个节点引擎包括多个矩阵处理器,所以每个节点可以并行执行多个矩阵运算。在一些实施例中,通过并行操作多个矩阵处理器,节点引擎可以在每个时钟周期输出矩阵乘法的结果。显着降低了用于等待数据读取的延迟,显着降低了矩阵乘法结果之间的延迟,并且显着增加了性能带宽。

[0025] 训练的结果是一个或多个经过训练的机器学习模型。在一些实施例中,训练多个模型,每个模型用于潜在不同的神经网络。例如,一个机器学习模型可以被训练为利用来自前向相机的传感器数据作为输入,而另一模型可以被训练为利用来自侧向超声波传感器的传感器数据作为输入。

[0026] 在107处,分发经过训练的机器学习模型。例如,经过训练的模型被分发到车辆并安装到该车辆上。该模型可以经由无线更新、由技术人员在维修车辆时、或其他方式安装。在某些情形下,模型以数据格式打包,以便于安装在车辆上。例如,模型可以被压缩以使将

模型传输到车辆所需的时间和带宽最小。在一些实施例中,多个模型(例如,每个模型用于在车辆上运行的不同神经网络引擎)可以打包在一起并且作为单个封装传输到车辆。

[0027] 在109处,应用经过训练的机器学习模型。例如,车辆上的卷积神经网络利用新模型来处理传感器数据并且实现自动驾驶或驾驶员辅助特征。在一些实施例中,应用多于一个模型和/或利用多于一个神经网络。例如,在一些车辆上,利用多个神经网络来处理来自不同传感器的不同数据。一旦利用了新模型,就可以捕获反映新模型性能的数据,并且该数据用于未来训练。图1的过程可以用于不断提高机器学习网络的性能。这样,处理循环回到捕获数据的101。可以分析数据以标识当前部署的模型的困难用例,并且可以利用对应捕获的数据进行未来训练。

[0028] 图2是图示了用于训练机器学习模型的系统的实施例的框图。使用图2的训练系统,可以训练机器学习模型以实现自动驾驶和/或驾驶员辅助驾驶功能。在一些实施例中,图2的训练系统用于执行图1的过程。在所示的示例中,训练系统利用位于车辆上的车辆子系统201的某些训练相关子系统。训练相关子系统与位于一个或多个训练数据中心221中的训练系统的服务器侧通信。车辆子系统201包括传感器203、深度学习网络205、AI处理器207、车辆控制模块209、网络接口211、车辆数据捕获系统213和捕获数据存储装置215。例如,可以存在附加车辆子系统以执行其他功能,但未示出。一个或多个训练数据中心221包括训练平台223、训练数据存储装置227、以及模型数据存储装置229。训练平台223包括至少一个或多个节点引擎225。节点引擎被连接(例如,在网状网络中)以对机器学习训练执行并行处理。在一些实施例中,训练平台223、训练数据存储装置227和模型数据存储装置229位于单个数据中心,但还可以跨越多个数据中心分布或复制。

[0029] 在一些实施例中,车辆(未示出)包括车辆子系统201以实现自主和驾驶员辅助功能并且捕获可以用于训练一个或多个机器学习模型以实现和/或改进功能和/或新特征的数目。在各个实施例中,不同的车辆子系统可以通信连接。例如,来自传感器203的传感器数据被馈送到车辆数据捕获系统213以存储在捕获数据存储装置215中。所捕获的数据经由网络接口211被发送到训练平台223。作为另一示例,来自传感器203的传感器数据被馈送到在AI处理器207上运行的深度学习网络205。在AI处理器207上运行的深度学习网络205的输出被馈送到车辆控制模块209。在各个实施例中,网络接口211是无线网络接口,诸如包括WiFi和/或蜂窝网络连接性的无线网络接口。网络接口211用于与远程服务器通信、拨打电话、发送和/或接收文本消息、将传感器数据传输到训练平台223等。在一些实施例中,视情况,车辆子系统201可以包括附加的或更少的子系统。例如,在一些实施例中,图像预处理器(未示出)用于预处理所捕获的传感器数据。作为另一示例,在一些实施例中,在将输出提供给车辆控制模块209之前,后处理部件(未示出)用于对深度学习网络205的输出执行后处理。在一些实施例中,触发分类器部件(未示出)用于将驾驶数据识别为潜在训练数据。

[0030] 在一些实施例中,传感器203包括一个或多个传感器。传感器203可以在车辆的不同位置处附着到车辆和/或沿一个或多个不同方向取向。例如,传感器203可以沿前向、后向、侧向等方向附着到车辆的前部、侧面、后部和/或车顶等。在一些实施例中,传感器203可以是诸如高动态范围相机之类的图像传感器。在一些实施例中,传感器203包括非视觉传感器。传感器203可以包括雷达、LiDAR和/或超声波传感器等。在某些实施例中,传感器203没有安装到带有车辆控制模块209的车辆。例如,传感器203可以安装在相邻车辆上和/或附着

到道路或环境,并且被包括作为用于捕获传感器数据的系统的一部分。

[0031] 在一些实施例中,深度学习网络205是用于实现自主车辆控制的深度学习网络。例如,深度学习网络205可以是诸如卷积神经网络(CNN)之类的人工神经网络,该人工神经网络使用传感器数据进行训练并且其输出被提供给车辆控制模块209。深度学习网络205所使用的机器学习模型可以使用图2的系统进行训练。

[0032] 在一些实施例中,人工智能(AI)处理器207是用于运行深度学习网络205的硬件处理器。在一些实施例中,AI处理器207是用于使用卷积神经网络(CNN)对传感器数据执行推断的专用AI处理器。AI处理器207可以针对传感器数据的位深度进行优化和/或针对诸如包括卷积、点积、向量和/或矩阵运算等之类的神经网络操作之类的深度学习操作进行优化。在一些实施例中,AI处理器207使用图形处理单元(GPU)来实现。在各个实施例中,AI处理器207耦合到存储器,该存储器被配置为向AI处理器提供指令,该指令当执行时,使得AI处理器对所接收的输入传感器数据执行深度学习分析并且确定用于至少部分地自主操作车辆的机器学习结果。

[0033] 在一些实施例中,车辆控制模块209被用于处理人工智能(AI)处理器207的输出并且将输出变换为车辆控制操作。在一些实施例中,车辆控制模块209用于控制车辆以进行自动驾驶并且可以调整车辆的速度和/或转向。例如,车辆控制模块209可以用于通过制动、转向、变道、加速和并入另一车道等来控制车辆。在一些实施例中,车辆控制模块209用于控制车辆照明,诸如刹车灯、转向信号、车头灯等。在一些实施例中,车辆控制模块209用于控制车辆音频条件,诸如车辆的音响系统、播放音频警报、启用麦克风、启用喇叭等。在一些实施例中,车辆控制模块209用于控制包括警告系统在内的通知系统,以通知驾驶员和/或乘客驾驶事件,诸如潜在碰撞或接近预期目的地。在一些实施例中,车辆控制模块209用于调整诸如车辆的传感器203之类的传感器。例如,车辆控制模块209可以用于改变一个或多个传感器的参数,诸如修改定向、改变输出分辨率和/或格式类型,增加或减少捕获率,调整所捕获的动态范围,调整相机的焦点,启用和/或禁用传感器等。在各个实施例中,车辆控制模块209用于实现车辆的自动驾驶和/或驾驶员辅助控制。

[0034] 在一些实施例中,网络接口211是用于发送和/或接收包括所捕获的传感器数据在内的数据的通信接口。在各个实施例中,网络接口211包括蜂窝或无线接口,用于与诸如训练平台223之类的远程服务器接口,以连接和进行语音呼叫,发送和/或接收文本消息,传输传感器数据,接收对包括新训练的机器学习模型在内的自动驾驶系统的更新等。例如,网络接口211可以用于接收针对传感器203、深度学习网络205、AI处理器207、车辆控制模块209和/或车辆数据捕获系统213的指令和/或操作参数的更新。例如,深度学习网络205的机器学习模型可以使用网络接口211来更新。作为另一示例,网络接口211可以用于更新传感器203的固件和/或车辆数据捕获系统213的操作参数,诸如用于确定要捕获数据的类型和数量的过滤器和/或参数。

[0035] 在一些实施例中,车辆数据捕获系统213和捕获数据存储装置215用于捕获和存储与车辆驾驶条件相关联的数据。车辆数据捕获系统213所捕获的数据存储在捕获数据存储装置215中。捕获数据存储装置215可以使用诸如硬盘驱动器、非易失性存储器等之类的任何适当数据存储装置来实现。在一些实施例中,捕获数据存储装置215使用数据库、文件系统或另一数据组织手段来实现。所捕获的车辆驾驶条件的数据可以包括图像传感器数据、

车辆操作参数(例如,速度、转向等)、车辆类型信息(例如,左侧驾驶、右侧驾驶、车辆型号等)、自动驾驶是否启用、自动驾驶最后一次脱离的时间、所检测的障碍物、驾驶条件等。可以在不干扰车辆驾驶和不需要驾驶员辅助的情况下被动捕获数据。车辆数据捕获系统213所捕获的数据包括从传感器203捕获的数据。

[0036] 在一些实施例中,车辆数据捕获系统213经由网络接口211与训练平台223通信。网络接口211可以是无线网络,诸如WiFi和/或蜂窝网络。车辆数据捕获系统213利用网络接口211将捕获数据存储装置215中存储的所捕获的数据传输到训练平台223。在一些实施例中,网络接口211用于下载经过训练的机器学习模型以安装在于该车辆上运行的深度学习网络205中。

[0037] 在图2的示例中,训练系统的服务器侧部件位于一个或多个训练数据中心221的一个或多个数据中心中并且包括训练平台223、训练数据存储装置227和模型数据存储装置229。训练平台223包括一个或多个计算机服务器,用于从车辆数据采集系统213接收所捕获的数据。训练平台223通过一个或多个训练数据中心221的诸如有限网络或光学网络之类的计算机网络经由无线网络接口211通信地连接到车辆数据采集系统213。训练平台223还包括一个或多个节点引擎225。例如,多个节点引擎225可以连接在网状网络中。训练平台223从车辆数据捕获系统213接收所捕获的数据,将数据处理成可用的训练(和验证)数据,并且利用节点引擎225来训练一个或多个新机器学习模型。训练数据存储装置227用于存储从一个或多个车辆接收的所捕获的数据。在一些实施例中,包括带注释的数据的用作训练数据的经过处理的所捕获的数据存储在训练数据存储装置227中。一旦完成训练,模型数据存储装置229就用于存储经过训练的机器学习模型。例如,不同版本的经过训练的机器学习模型可以存储在模型数据存储装置229中,并且用于确定不同模型的相对功能并且标识改进领域。在一些实施例中,一个或多个数据存储装置用于实现训练数据存储装置227和模型数据存储装置229。

[0038] 在一些实施例中,节点引擎225包括多个连接节点,该多个连接节点可以用于并行化计算任务。每个连接节点包括至少一个矩阵处理器,并且可能多于一个矩阵处理器。例如,单个节点可以包括八个矩阵处理器,每个矩阵处理器能够确定至少一个矩阵乘法结果。在一些实施例中,矩阵乘法结果需要单个矩阵处理器至少最少数目的时钟周期来计算。通过扩展每个节点以包括多个矩阵处理器,在与计算矩阵乘法的最小数目的时钟周期相对应的初始延迟之后,节点可以针对每个时钟周期输出一个矩阵乘法的结果。例如,如果矩阵乘法需要八个时钟周期才能完成,在七个时钟周期的初始延迟之后,具有八个矩阵处理器的节点可以在每个时钟周期确定矩阵乘法的结果。在各个实施例中,吞吐量还由存储器访问确定,包括访问矩阵操作数的延迟在内。在各个实施例中,节点引擎能够使用多种数字格式来执行矩阵计算。例如,节点可以利用固定点数格式和浮点数格式。对于浮点格式,节点可配置为以诸如8位格式、16位格式和32位格式之类的多种格式运行。对于每位深度,可以选择一个或多个不同格式。依据计算目标,不同格式可以用于表示数值。可以选择格式来为浮点数的尾数分配更高的精确性,并且可以选择另一格式来为浮点数的指数分配更高的精确性。在一些实施例中,浮点格式利用可配置偏置来进一步定制计算运算。数字格式的可配置性允许训练系统例如基于预期输入值、中间值和输出值来靶向不同的机器学习运算。在各个实施例中,包括对多种浮点格式和使用可配置偏置的浮点格式的支持的节点的可配置性

大大提高了矩阵计算运算的带宽和性能,而不会牺牲精确性和准确性。同样,功耗和效率也得到显著提高。

[0039] 图3是图示了用于执行矩阵计算的节点引擎的实施例的框图。在所示的示例中,节点引擎300包括控制单元301、存储器303、负载寄存器305、后处理单元寄存器文件307、多路复用器309和311、矩阵处理器313和351至357、输出阵列315和后处理单元317。在各个实施例中,节点引擎可以包括多个矩阵处理器以并行计算多个矩阵运算。在所示的示例中,节点引擎300包括八个矩阵处理器313和351至357。每个矩阵处理器包括数据输入阵列、权重输入阵列、多个输出累加器、以及矩阵计算单元。在所示的示例中,矩阵处理器313包括数据输入阵列321、权重输入阵列323、以及两个输出累加器329和331。数据输入阵列和权重输入阵列将输入馈送到矩阵计算单元325。例如,输入阵列(例如,数据输入阵列321和/或权重输入阵列323)中的数据在多个周期(例如,八个连续周期)上移位一定数目的字节(例如,八个字节)以馈送矩阵计算单元325。在一些实施例中,每个矩阵处理器包括单个数据输入阵列和单个权重输入阵列。矩阵计算单元325包括诸如计算单元327之类的计算单元矩阵。 $M \times N$ 维矩阵计算单元包括 $M \times N$ 个计算单元。每个输入阵列的大小被设计为适合整个输入矩阵,并且每个输出累加器的大小被设计为适合整个矩阵结果。在一些实施例中,节点引擎支持多种浮点格式,这些浮点格式包括图4的8位浮点格式400和410以及图5的21位浮点格式500。在一些实施例中,节点引擎300用于执行图1、图6、图7和/或图8的过程。

[0040] 在一些实施例中,节点引擎300可以包括未示出的附加部件和附加控制线。例如,节点引擎300可以包括附加寄存器,诸如标量寄存器、一个或多个存储器高速缓存、用于格式化针对矩阵处理器的值的数据格式化器、以及从控制单元301到诸如多路复用器309和311和矩阵处理器351至357之类的子部件的附加控制线,作为几个示例。在一些实施例中,某些寄存器(未示出)专用于存储诸如数字格式和浮点数的可配置偏置之类的可配置参数。在一些实施例中,连接节点引擎300的不同部件的总线为宽数据总线。可以选择总线的大小以优化传送矩阵值。例如,总线的宽度可以都是64个字节。这允许将64个1字节元素的 8×8 矩阵作为包含单元从存储器传送到寄存器、矩阵处理器等。

[0041] 在所示的示例中,控制单元301通信地连接到节点引擎300的一个或多个部件,该一个或多个部件包括存储器303、矩阵处理器313、输出阵列315和后处理单元317。尽管未示出,但是控制单元301还通信地连接到剩余矩阵处理器351至357中的每个矩阵处理器。在各个实施例中,控制单元301用于同步计算运算的处理,这些计算运算包括矩阵运算和后处理运算(诸如向量运算)和/或存储器和寄存器的访问。例如,控制单元301向矩阵处理器313发送信号以调度矩阵计算指令并且可以监测来自矩阵处理器313的就绪信号以指示何时可以接收新指令和/或何时完成矩阵运算以及何时矩阵结果准备就绪。

[0042] 在一些实施例中,存储器303是用于存储矩阵计算和后处理计算的输入操作数和输出结果的存储器模块。存储器303可以包括一个或多个高速缓存(未示出)。在所示的示例中,存储器303连接到负载寄存器305、多路复用器309和311、以及后处理单元寄存器文件307。附加的连接或更少的连接是可能的,这取决于在存储器中存储数据和从存储器检索数据所需的灵活性。如所示出的,数据可以从存储器读取到负载寄存器305和后处理单元寄存器文件307并且从负载寄存器305和后处理单元寄存器文件307读取到存储器。与寄存器的连接允许将数据值快速存储在寄存器中,例如,作为矩阵或向量计算的参量(argument)。存

存储器303还连接到多路复用器309和311,以使可以从存储器中检索输入矩阵。在一些实施例中,对存储器303的存储器访问由存储器仲裁器(未示出)控制,以例如通过将存储器请求排队并且将某些存储器读取优先于其他存储器读取来优化存储器请求。在一些实施例中,存储器303是静态随机存取存储器(SRAM)。

[0043] 在一些实施例中,节点引擎300包括诸如负载寄存器305和后处理单元寄存器文件307之类的寄存器。这些寄存器可以用于优化存储器访问。作为几个示例,寄存器可以用于存储从存储器303检索的值,以在将值写入存储器303之前存储这些值,存储矩阵处理器的输入值和输出值,以及存储后处理单元的输入值和输出值。在一些实施例中,后处理单元寄存器文件307是用于后处理单元317的寄存器文件并且与后处理单元317的不同车道配置(例如,64个车道配置、32个车道配置和/或16个车道配置)兼容。例如,后处理单元寄存器文件307的寄存器可以使用诸如1字节值、2字节值和4字节值之类的各种字节格式来寻址。在一些实施例中,每个寄存器的大小为64个字节并且该寄存器可以存储64个1字节元素、32个2字节元素或16个4字节元素。在各个实施例中,数据格式可以被配置并且包括各种8位浮点格式、16位浮点格式和32位浮点格式。

[0044] 在一些实施例中,多路复用器用于选择到矩阵处理器的输入操作数源。在所示的示例中,多路复用器309和311用于选择针对数据输入矩阵的源和针对矩阵处理器313的权重输入矩阵。依据在每个多路复用器处接收的控制信号,数据可以源自存储器303或后处理单元寄存器文件307。在一些实施例中,源自存储器303的数据经由负载寄存器305的寄存器检索。在一些实施例中,多路复用器309和311还用于选择针对矩阵处理器351至357的数据输入矩阵和权重输入矩阵。通过偏移(offset)节点引擎的多个矩阵处理器的处理,一对多路复用器用于选择节点引擎的所有矩阵处理器的输入。在各个实施例中,多路复用器309和311用于控制哪个矩阵处理器接收哪个矩阵操作数。依据配置,单个矩阵处理器、所有矩阵处理器的子集、或所有矩阵处理器接收所选择的矩阵操作数。在各个备选实施例中,节点引擎300包括专用于矩阵处理器351至357中的每个矩阵处理器的附加多路复用器(未示出)。

[0045] 在一些实施例中,矩阵处理器313接收矩阵运算指令并且执行诸如矩阵乘法之类的矩阵计算。对于每个矩阵指令,矩阵处理器313在一个或多个输入阵列中存储一个或多个矩阵操作数。例如,数据矩阵存储在诸如数据输入阵列321之类的数据输入阵列中,而权重矩阵存储在诸如权重输入阵列323之类的权重输入阵列中。在各个实施例中,矩阵操作数是一对数据和权重矩阵、一对数据和梯度矩阵、一对权重和梯度矩阵、或另一对适当矩阵操作数。在各个实施例中,矩阵处理器313用于计算多个相关矩阵计算,作为用于计算矩阵的矩阵乘法的过程的一部分,这些矩阵太大而无法适应在矩阵处理器313的输入阵列321和323中。相关矩阵计算的结果组合为计算较大矩阵的矩阵乘法的过程的一部分。在各个实施例中,矩阵处理器313交错多个矩阵运算(相关或不相关)。例如,矩阵处理器313可以交错对第一对矩阵执行一个或多个相关矩阵运算与对第二对矩阵执行一个或多个相关矩阵运算。例如,矩阵处理器313可以分别对作为较大矩阵 W_A 和 D_A 的一部分(例如,其切片)的矩阵 W_1 和 D_1 执行矩阵乘法,并且随后分别对作为较大矩阵 W_B 和 G_B 的一部分(例如,其切片)的矩阵 W_2 和 G_2 执行矩阵乘法。矩阵 W_1 和 D_1 的矩阵乘法结果是用于计算较大矩阵 W_A 和 D_A 的矩阵乘法的部分结果,矩阵 W_2 和 G_2 的矩阵乘法结果是用于计算较大矩阵 W_B 和 G_B 的矩阵乘法的部分结果。输入矩阵 W_1 和 D_1 以及输入矩阵 W_2 和 G_2 存储在一对权重和数据输入阵列中,诸如阵列321和323。在

一些实施例中,单独输出累加器329和331分别用于累加中间和/或最终结果 $W_1 * D_1$ 以及中间和/或最终结果 $W_2 * G_2$ 。例如,输出累加器329用于累加与矩阵 W_1 和 D_1 相关联的矩阵乘法的中间和/或最终结果,并且输出累加器331用于累加与矩阵 W_2 和 G_2 相关联的矩阵乘法的中间和/或最终结果。

[0046] 在一些实施例中,数据输入阵列和权重输入阵列的大小被设计为以线性化形式适合整个矩阵。例如,能够对大小为 $M \times N$ 和 $N \times O$ 的两个矩阵执行矩阵乘法的矩阵处理器具有用于接收对应 $M \times N$ 和 $N \times O$ 个输入矩阵的大小为 $M \times N$ 元素的输入阵列和大小为 $N \times O$ 元素的另一输入阵列。在一些实施例中,矩阵处理器对两个 8×8 矩阵执行计算,并且权重输入阵列和数据输入阵列各自的大小被设计为接收64个元素。同样,输出累加器的大小被设计为存储整个结果矩阵。用于存储大小为 $M \times N$ 和 $N \times O$ 的两个矩阵之间的矩阵乘法结果的输出累加器的大小被设计为接收 $M \times O$ 个元素。在一些实施例中,矩阵处理器对两个 8×8 矩阵执行计算并且将中间和最终矩阵结果存储在累加器中,该累加器的大小被设计为适合与 8×8 结果矩阵相对应的64个元素。

[0047] 在所示的示例中,输入阵列馈送矩阵计算单元325。矩阵计算单元325由计算单元(诸如计算单元327)矩阵组成。每个计算单元是可以接收两个操作数的处理元件,每个元素来自每个输入矩阵,并且对两个输入操作数执行诸如乘法之类的计算。在一些实施例中,计算是乘法和加法。例如,两个输入元素相乘,并且结果与累加器中的当前结果相加并且存储回到累加器。在一些实施例中,诸如计算单元327之类的每个计算单元包括用于执行诸如乘法、除法、加法或减法运算之类的算术逻辑运算的算术逻辑单元。在一些实施例中,可以在同一时钟周期中执行多个运算,诸如执行部分点积所需的乘法和加法运算。每个计算单元可以包括加法器、乘法器、和/或与一对或多对数据和权重输入阵列相对应的一个或多个累加器。在一些实施例中,诸如计算单元327之类的每个计算单元包括浮点乘法器和一个或多个累加器。尽管在图3中输出累加器329和331被描绘为与计算单元327分开,但是在一些实施例中,输出累加器329和331的对应部分集成到它们各自的计算单元中。例如,每个计算单元的累加器一起构成输出累加器329和331。

[0048] 在各个实施例中,矩阵计算单元325的计算单元支持浮点运算,诸如浮点乘法和加法。在各个实施例中,每个计算单元包括乘法器和一个或多个累加器以在单个周期中执行乘法和加法运算。在每个矩阵计算开始之前,可以清除指定累加器。在执行矩阵计算的过程期间,指定累加器用于累加和存储中间结果。在一些实施例中,矩阵处理器313是 8×8 矩阵处理器,并且矩阵计算单元325包括64个计算单元。每个周期,128个元素可以加载到矩阵计算单元325中,两个输入元素作为64个计算单元中的每个计算单元的操作数。每个计算单元还可以访问存储在指定累加器中的累加器值。

[0049] 在一些实施例中,矩阵乘法需要多个时钟周期来完成。对于每个时钟周期,从输入操作数中检索单个行和单个列。例如,从数据输入阵列中存储的矩阵中检索行,而从权重输入阵列中存储的矩阵中检索列。在一些实施例中,通过将输入阵列中的数据移动整行或整列来检索数据。每个行和每个列都是向量,并且每个向量都跨整个计算单元复制。每个行“向下”(down)矩阵计算单元325的行复制,并且每个列“跨越”矩阵计算单元325的列复制。对于 8×8 矩阵处理器,权重输入矩阵的每个列是8个元素并且数据输入矩阵的每个行是8个元素。对于每次传递,针对矩阵计算单元325的八个列中的每个列,复制单个权重列,并且针

对矩阵计算单元325的八个行中的每个行,复制单个数据行。通过一次跨越并向下一个行和一个列复制数据, 8×8 的矩阵处理器可以在8个周期内完成矩阵乘法。在每个周期期间,乘法和累加的中间结果存储在指定累加器中。到第八个也是最后一个周期,最终矩阵结果存储在指定累加器中。使用不同维度(例如, 4×4 或 16×16 矩阵)的矩阵处理器可以与对应大小的输入阵列、累加器和计算单元一起使用。

[0050] 在一些实施例中,输入数据元素是8位浮点值。通过利用8位值,矩阵处理器的带宽性能得到显著提高。通过利用可配置浮点值和可配置偏置,机器学习训练所需的精确性和准确性得以保留,并且带宽得以增加。利用8位格式,64字节 $\times$ 64字节矩阵处理器可以计算两个 8×8 矩阵(总共128个元素)的矩阵乘法。相比之下,使用32位格式,64字节 $\times$ 64字节矩阵处理器可以计算两个 4×4 矩阵(总共只有32个元素)的矩阵乘法。通过使用可配置8位浮点格式优化矩阵元素,用于将矩阵元素加载到矩阵处理器的带宽显著提高。单位面积的功耗也显著改善。为防止上溢错误和下溢错误,存储在指定累加器中的中间和最终结果利用较大位格式,诸如21位、27位或其他适当浮点格式。使用8位元素作为输入元素并且使用21位格式存储中间结果保留训练所需的精确性和准确性,同时还维持矩阵处理器的高输入带宽。在各个实施例中,每个输出累加器使用21位浮点数存储结果矩阵的每个元素,诸如图5的格式500。在一些实施例中,矩阵处理器313是使用8位浮点输入值执行矩阵运算并且使用21位浮点值计算中间和最终矩阵结果的 8×8 矩阵处理器。输入阵列为64字节(64个8位元素),输出累加器为168字节(64个21位元素)。在各个实施例中,输出累加器由矩阵计算指令指定。同样,8位浮点格式和指数偏置可以通过矩阵计算指令和/或一个或多个寄存器参量来配置。

[0051] 在一些实施例中,矩阵处理器313支持多个不同的8位浮点格式。例如,不同的格式400和410被支持并且可以基于计算任务来进行选择。每个格式分配不同数目的位来表示浮点数的指数和尾数。依据用例,选择一个格式或另一格式。在需要高精度数字的情况下,可以为尾数分配更多位,并且选择格式,诸如具有比格式410更多尾数位的格式400。可以选择具有更多尾数位的格式来执行梯度下降,其中需要非常小的增量(δ)来保留精确性。作为另一示例,可以选择具有更多尾数位的格式来执行前向传播以计算成本函数。作为另一优化,每个浮点格式都利用可配置偏置。可配置偏置用于移动指数范围。例如,在没有指数偏置的情况下,由3位表示的指数可以指定 2^0 与 2^7 (含)之间的指数值。偏置5将指数的范围移动到具有介于 2^{-5} 与 2^{+2} (含)之间的指数值。作为另一示例,使用4位来表示指数,并且偏置15将指数的范围从 2^0 和 2^{31} (含) 移动到介于 2^{-15} 与 2^{+16} (含) 之间。在各个实施例中,通过优化指数字段的位数和偏置的位数,可以优化使用指数表达的范围和浮点数的数字覆盖范围以保留预期输入和结果的准确性和精确性。

[0052] 在一些实施例中,浮点格式支持非规则数。例如,具有零值的指数字段无需没有前导零的归一化尾数。通过支持非规则数,可以增加指数范围和可以表示的值的数目。在各个实施例中,诸如计算单元327之类的每个计算单元包括对使用一个或多个非规则操作数执行浮点运算的支持。

[0053] 在一些实施例中,可配置偏置的值受用于表示可配置偏置的位数的限制。例如,3位可配置偏置可以具有八个不同值(0至7(含))。在一些实施例中,作为优化,由可配置偏置表示的值并非连续。例如,由3位可配置偏置表示的八个值不限于值0至7。相反,偏置可以选

自8个不同值。例如,可配置偏置可以选自八个预先确定的值:1、3、5、7、9、11、15和17。在一些实施例中,预先确定的值基于最有用偏置来确定。可以至少部分选择预先确定的值以使范围最大并且使不同偏置的范围之间的重叠最小。在一些实施例中,可配置偏置由矩阵处理器指令指定和/或存储在寄存器(未示出)中。在一些实施例中,可配置偏置可重新配置。例如,在执行算术运算后,可以重新配置可配置偏置以调整到结果的新范围。在一些实施例中,重新配置被指定为计算指令的一部分。例如,指令可以指定用于重新配置可配置偏置的新偏置。

[0054] 在一些实施例中,矩阵计算单元的计算单元可以被分组为还支持用于较大输入数字格式的矩阵运算。例如,每个以8位浮点矩阵元素作为输入进行运算的 8×8 矩阵计算单元的计算单元可以分组为使用16位浮点矩阵元素作为输入来执行 4×4 矩阵运算。在一些实施例中,输出累加器的大小被设计为防止量化结果的准确性的损失。例如,使用单位用于符号位、8位用于指数、7位用于尾数和不可配置指数偏置的16位浮点格式利用浮点结果的27位中间浮点格式。27位浮点格式可以为符号位分配单位,为指数分配9位,为尾数分配17位。对经分组操作模式的支持部分通过支持更多个操作数格式使得矩阵计算单元更加通用。

[0055] 在各个实施例中,经分组操作模式通过将输入操作数拆分成多个分量并且将每个拆分分量提供给该组的不同计算单元来执行矩阵运算。每个拆分分量都表示为浮点数,并且当加在一起时,不同拆分分量合计原始操作数。例如,输入操作数被拆分成最重要位(即,高分量)和最低重要位(即,低分量)。在各个实施例中,高分量的指数使用输入操作数的相同指数值,而低分量的指数被调整为考虑从输入操作数中减去最重要位。在一些实施例中,最低重要位的分量被归一化。在一些实施例中,计算单元支持非规则数并且该分量可以表示为非规则数。

[0056] 在各个实施例中,当使用两倍于计算单元格式大小的操作数编号格式(例如,16位浮点操作数而非8位浮点操作数)对两个输入操作数执行乘法时,四个计算单元分组在一起,每个输入操作数都有对应高分量和低分量。通过将高-高分量、高-低分量、低-高分量和低-低分量配对并且将不同对提供给组的不同计算单元,每个输入操作数的高和低分量提供给处理元件。在该组的每个计算单元处,执行矩阵乘法并且结果存储在与计算单元相关联的输出累加器中。在一些实施例中,输出累加器利用具有比原始输入操作数更高的位数的浮点格式。例如,对于没有可配置指数偏置的16位输入操作数,输出累加器可以利用27位。当经分组单元的输出结果加在一起时,结果是原始输入操作数的矩阵乘法。在一些实施例中,结果被移出矩阵计算单元并且使用诸如向量计算单元之类的后处理单元加在一起。例如,浮点加法指令用于使分量结果相加以确定乘法结果。浮点向量相加指令可以用于使结果得向量的分量相加。在各个实施例中,矩阵计算单元是图3的矩阵计算单元325并且后处理单元是图3的后处理单元317。

[0057] 在一些实施例中,节点引擎300包括多个矩阵处理器313和351至357。矩阵处理器351至357的功能和部件关于矩阵处理器313进行描述。在一些实施例中,每个矩阵处理器需要至少最小数目的周期来完成矩阵乘法,例如, 8×8 矩阵处理器需要八个周期。通过在单个节点引擎中并入多个矩阵处理器,矩阵乘法可以分布到不同矩阵处理器。结果输出可以交错以在每个周期从不同矩阵处理器中读取矩阵结果。对于8个 8×8 矩阵处理器的集合,每个矩阵处理器可以每8个周期输出矩阵结果。交错处理器允许来自不同处理器的每个时钟周

期的矩阵结果。在一些实施例中,可以使用不同大小的矩阵处理器,例如,4×4处理器或16×16处理器。同样,基于矩阵处理器计算管道的深度,节点引擎中可以包括不同数目的矩阵处理器。

[0058] 在一些实施例中,矩阵指令指定特定矩阵运算、特定矩阵处理器,指定用于存储矩阵结果的累加器,并且指定矩阵操作数的位置。矩阵操作数的位置可以使用寄存器值或存储器地址来指定。例如,矩阵指令可以指定矩阵乘法、矩阵乘法处理器313、输出累加器329、后处理单元寄存器文件307的寄存器、以及存储器303的存储器地址。在一些实施例中,控制单元301发出矩阵指令。在一些实施例中,运算包括矩阵乘法、矩阵加法、点积、矩阵求逆等。在一些配置中,每个矩阵处理器的输出累加器唯一标识矩阵处理器。通过将特定输出累加器指定为矩阵指令的一部分,以固有方式选择矩阵处理器。例如,对累加器使用A0-A11命名方案,第一输出累加器和第二输出累加器(例如,A0和A1)被映射到矩阵处理器313,第三输出累加器和第四输出累加器(例如,A2和A3)被映射到矩阵处理器351,第五输出累加器和第六输出累加器(例如,A4和A5)被映射到矩阵处理器352等。在该示例中,累加器329和331分别被称为A0和A1。由于只有矩阵处理器313可以将结果存储到累加器A1,所以指定累加器A1的矩阵乘法指令被发布到矩阵处理器313。

[0059] 在一些实施例中,输出阵列315用于检索一个或多个矩阵处理器的结果。在一些实施例中,输出阵列315包括多路复用器以确定将结果从哪个矩阵处理器加载到输出阵列中。在一些实施例中,输出阵列是64字节阵列并且需要两个移动指令来将矩阵结果从矩阵处理器移动到输出阵列中。例如,使用21位浮点值的矩阵结果需要168个字节。每个21位浮点值在移动命令期间被转换为16位浮点值。仅使用两个移动指令,64个元素的结果矩阵就从64个21位浮点值转换为64个16位浮点值。例如,上移指令将最高的32个元素移入输出阵列,而下移指令将剩余的最低32个元素移入输出阵列。在各个实施例中,输出阵列是64字节的,以使在执行第二移动之前,第一移动的结果首先存储在寄存器(诸如后处理单元寄存器文件307的寄存器)中。在各个实施例中,输出阵列是临时输出阵列,直至值被移动到存储器或寄存器为止。在一些实施例中,移动指令是非破坏性的并且不会从矩阵处理器清除矩阵结果,例如,通过清除源累加器。

[0060] 在一些实施例中,后处理单元317用于执行诸如归一化、扩展、激活函数、池化等之类的后处理。在一些实施例中,后处理单元317是对向量的每个元素进行操作的向量计算引擎。后处理单元可以利用包括浮点数格式在内的不同的数字格式,诸如1字节数字格式、2字节数字格式和4字节数字格式。在一些实施例中,可以配置后处理单元317的通道数。例如,采用64字节向量的后处理单元317可以对与64个通道配置、32个通道配置和16个通道配置相对应的64个1字节元素、32个2字节元素或16个4字节元素进行操作。在所示的示例中,后处理单元317利用后处理单元寄存器文件307来检索用于输入的数据并且来存储后处理结果。在一些实施例中,附加后处理单元(未示出)可以根据需要包括在节点引擎中以执行附加机器学习功能。

[0061] 图4是图示了8位浮点格式的实施例的框图。在所示的示例中,8位浮点格式400和410是用于使用符号、尾数和指数表示浮点数的不同8位浮点格式。在一些实施例中,诸如节点引擎300之类的节点引擎和诸如图3的矩阵处理器313之类的矩阵处理器利用8位浮点格式400和410进行矩阵运算。通过使用诸如格式400和410之类的8位浮点格式而非16位浮点

格式、32位浮点格式或其他浮点格式执行矩阵运算,矩阵处理器的带宽显着增加。在一些实施例中,格式400和410支持可配置偏置。可配置偏置在表示指数时实现更大的范围以提高精确性,同时仍维持8位数据大小。在一些实施例中,浮点格式400和410支持非规则数以增加可以表示的值的数目。

[0062] 在所示的示例中,8位浮点格式400包括用于符号位401的单位、用于指数403的4位、以及用于尾数405的3位。符号位401、指数403和尾数405一共占8位,并且可以用来表示浮点数。同样,8位浮点格式410包括用于符号位411的单位、用于指数413的5位、以及用于尾数415的2位。符号位411、指数413和尾数415总共占8位,并且可以用于表示浮点数。在一些实施例中,可配置偏置用于偏置指数。例如,格式400的4位指数403允许指数403具有16个不同值(即,值0至15(含))。使用没有偏置的4位(或等同于可配置偏置设为零),指数403可以表示具有分别与具有值0和15的指数字段相对应的值 2^0 至 2^{15} 的指数。通过使用可配置偏置,可以移动指数的范围。例如,使用设置为值5的可配置偏置,指数403可以表示具有值 2^{-5} 至 2^{10} 的指数。在各个实施例中,可配置偏置的值受用于表示可配置偏置的位数限制。例如,3位可配置偏置可以具有八个不同的值。在一些实施例中,由可配置偏置表示的值并非连续。例如,由3位可配置偏置表示的八个值不限于值0至7。相反,偏置可以选自8个不同值。例如,可配置偏置可以选自八个预先确定的值:1、3、5、7、9、11、15和17。在一些实施例中,预先确定的值基于最有用偏置来确定。在一些实施例中,至少部分选择预先确定的值以使指数的范围最大并且使不同偏置的范围之间的重叠最小。在一些实施例中,可配置偏置由矩阵处理器指令指定和/或存储在寄存器(未示出)中。

[0063] 在各个实施例中,矩阵处理器支持多种不同的8位浮点格式,诸如格式400和410。通过支持多种格式,可以在指数或尾数中利用精确性。例如,诸如梯度下降之类的某些操作可能需要附加精确性,并且由此尾数需要更大数目的位。作为另一示例,更多位可以用于尾数以用于将值聚类在一起并且无需附加指数范围的运算。相比之下,对于某些操作,值的范围可能更大,并且需要较大的指数范围。使用格式410,尾数专用的位更少,指数专用的位更多。在一些实施例中,格式由矩阵处理器指令指定并且可以存储在寄存器(未示出)中。在各个实施例中,可以支持未描绘的附加浮点格式。例如,可以支持4位尾数格式和3位指数格式(未示出)。

[0064] 图5是图示了21位浮点格式的实施例的框图。在所示的示例中,浮点格式500是用于使用符号、尾数和指数表示浮点数的21位浮点格式。在一些实施例中,诸如节点引擎300之类的节点引擎和诸如图3的矩阵处理器313之类的矩阵处理器利用诸如格式500之类的21位浮点格式用于某些矩阵运算,诸如用于存储矩阵乘法和/或矩阵加法的结果(和中间结果)。在一些实施例中,格式500由诸如图3的输出累加器329和331之类的矩阵处理器的累加器使用。例如,如果结果局限于同一8位格式,则两个8位乘法操作数的乘法结果可能导致上溢错误或下溢错误。对结果使用大于8位的格式防止上溢错误和下溢错误。同样,当使用8位矩阵元素计算矩阵乘法时,使用21位浮点格式存储中间和最终结果防止上溢错误或下溢错误。使用位深小于32位的结果提高存储器使用效率。在各个实施例中,位深为21位的格式500用于优化存储器使用和准确性两者。在一些实施例中,格式500支持可配置偏置。可配置偏置允许更大的范围以提高精确性,同时仍维持21位数据大小。在一些实施例中,可配置偏置由矩阵处理器指令指定和/或存储在寄存器(未示出)中。

[0065] 在所示的示例中,21位浮点格式500包括用于符号位501的单位、用于指数503的7位、以及用于尾数505的13位。符号位501、指数503和尾数505一共占21位,并且可以用来表示浮点数。在一些实施例中,可配置偏置用于偏置指数。例如,格式500的7位指数503允许指数503具有128个不同值(即,值0至127(含))。使用没有偏置的7位(或等同于可配置偏置设置为零),指数503可以表示具有值 2^0 至 2^{127} 的指数,其分别与具有值0和127的指数字段相对应。

[0066] 在各个实施例中,格式500由用于诸如图3的节点引擎300和矩阵处理器313之类的节点引擎的矩阵处理器的诸如图3的输出累加器329和331之类的一个或多个累加器使用。在一些实施例中,寄存器(未示出)用于存储用于在特定累加器中存储浮点数的可配置偏置的设置。在一些实施例中,可以使用多个21位格式(例如,对于指数字段和尾数字段具有不同的位分配)并且特定格式由矩阵处理器指令指定。可配置偏置的值可以使用矩阵处理器指令指定和/或存储在寄存器中。

[0067] 尽管图5描绘了可以由诸如图3的输出累加器329和331之类的矩阵处理器的累加器使用的21位浮点格式,但可以使用具有备选位深的格式。例如,依据操作要求,诸如用于防止精确性损失的要求,当支持对某些16位浮点运算的操作时,27位浮点格式可以用于防止量化结果的精确性的损失。作为一个示例,27位浮点格式可以包括用于单位的符号位、用于指数的9位、以及用于尾数的17位。27位浮点格式可以用于累加对16位浮点操作数的乘法运算。在一些实施例中,16位浮点操作数使用用于符号位的单位、用于指数的8位和用于指数的7位表示。

[0068] 图6是图示了用于执行矩阵计算的过程的实施例的流程图。图6的过程由诸如图2的训练平台223的训练平台使用以通过诸如图2的节点引擎225或图3的节点引擎300之类的一个或多个节点引擎执行矩阵计算。在一些实施例中,训练平台接收一个或多个矩阵计算运算,并且跨越不同节点引擎并行化这些运算。然后,每个节点引擎还可以将其跨越不同矩阵处理器的运算并行化。视情况而定,可以在一个或多个节点引擎处组合结果以确定结果,诸如机器学习模型的权重矩阵。在一些实施例中,图6的过程作为图1的步骤105的一部分来执行。

[0069] 在601处,接收计算指令。在一些实施例中,计算指令由诸如图2的训练平台223之类的训练平台接收。训练平台处理计算指令并且对不同节点引擎执行必要工作划分和分配。例如,在启动机器学习训练过程的训练平台的服务器处接收请求将图像与滤波器卷积的计算指令。在一些实施例中,指令可以包括执行计算指令所需的参数,包括所涉及的运算和操作数。例如,指令可以包括输入操作数的大小(例如,每个输入矩阵的大小)、每个输入矩阵的起始地址、步幅参数、填充参数、和/或矩阵、向量和/或后处理命令。例如,计算指令可以描述图像数据大小(例如, 96×96 、 1920×1080 等)和位深(例如,8位、16位等)以及滤波器大小和位深等。在许多场景中,矩阵计算的矩阵可能大于矩阵处理器内部所能适配的矩阵,使得可以执行附加处理来细分计算,以使它可以由不同的节点引擎或矩阵处理器执行。

[0070] 在603处,确定矩阵运算和操作数。在601处接收的计算指令的一个或多个矩阵大于矩阵处理器的输入矩阵的情况下,601的计算指令被拆分成小分量运算。在603处,确定矩阵运算和与较小分量运算相对应的操作数,并且可以包括将原始矩阵操作数切片、分割或划分成较小矩阵并且对较小矩阵执行矩阵运算。可以组合对较小矩阵的矩阵运算的结果以

完成在601处接收的计算指令。可以指派不同的节点引擎和矩阵处理器来执行计算指令的不同分量。在一些实施例中,矩阵操作数的元素可以被转换为8位浮点格式或以转换为8位浮点格式为目标。节点引擎使用诸如图4的格式400或格式410之类的8位浮点格式来增加处理和性能带宽以及矩阵处理器的功率效率。在一些实施例中,选择或将选择针对对应浮点格式的可配置偏置。例如,选择具有高精度尾数的格式以进行梯度下降运算。

[0071] 在各个实施例中,较大矩阵被切片成较小二维矩阵,其中大小局限于矩阵处理器的适当维度。在一些实施例中,经切片矩阵是具有指向引用原始矩阵的元素的地址的较小矩阵。经切片矩阵可以序列化为向量进行处理。在一些实施例中,矩阵的不同切片可以与先前切片重叠。在各个实施例中,矩阵可以仅在与读取缓冲器大小的倍数相对应的边界处被切片。例如,在每个读取缓冲器的大小为8个字节的情况下,经切片矩阵的每个行都必须以8的倍数的地址开始。在矩阵适配在计算阵列的情况下,则无需切片(即,所使用的矩阵切片只是原始矩阵)。

[0072] 在605处,分配并执行矩阵运算。例如,与在603处确定的矩阵运算和操作数相对应的矩阵运算被分配给一个或多个节点引擎以及节点引擎的一个或多个矩阵处理器。在各个实施例中,矩阵运算使用8位元素矩阵由一个或多个矩阵处理器来执行。矩阵结果的元素的值被累加为21位浮点格式、27位浮点格式或其他适当浮点格式。在各个实施例中,矩阵结果可以以包括8位浮点格式、16位浮点格式和32位浮点格式的几种格式中的一个格式移出矩阵处理器。在各个实施例中,每个节点引擎可以通过利用多个矩阵处理器并行执行多个矩阵运算。

[0073] 在一些实施例中,对矩阵操作数的引用与对节点引擎的操作一起分配。这样,节点引擎可以执行数据读取以加载经切片矩阵的对应元素。在一些实施例中,节点引擎将线性化经切片矩阵用于加载到存储器和/或寄存器中,其中输入矩阵然后可以发送到矩阵处理器。在一些实施例中,节点引擎的控制单元协调运算的调度、发布和同步,这些运算包括:加载经切片矩阵操作数(包括寻址矩阵操作数的指定步幅、填充和其他参数);以及操作矩阵处理器。一旦向矩阵处理器发出矩阵运算,矩阵处理器就花费一定数目的时钟周期来完成矩阵运算。在一些实施例中,矩阵处理器使用图7和/或图8的过程来执行矩阵运算。

[0074] 在607处,执行后处理。在一些实施例中,后处理可以由节点引擎执行并且可以包括在完成矩阵运算之后执行的附加向量运算。后处理操作可以由节点引擎的诸如向量处理器或向量计算单元之类的后处理单元执行。在一些实施例中,向量后处理包括:执行复杂操作,诸如算术运算、扩展、归一化和/或对向量的每个元素应用诸如线性整流单元(ReLU)函数之类的激活函数。在一些实施例中,依据所需精确性,向量的元素可以被转换/格式化为8位元素、16位元素或32位元素。在各个实施例中,每个节点引擎的分布式矩阵运算的结果可以发送回到训练平台服务器或由训练平台服务器重定向并且用于其他处理。例如,在605处分配并执行的矩阵运算的结果可以被组合并用作附加向量或矩阵运算的操作数。在607处启动后处理之后,处理循环回到601以接收附加计算指令。在一些实施例中,后处理无需在处理循环回到601以获得附加计算指令之前完成。

[0075] 图7是图示了用于执行矩阵计算的过程的实施例的流程图。图7的过程被诸如图3的节点引擎300的矩阵处理器313和351至357之类的矩阵处理器用来执行矩阵计算。在一些实施例中,节点引擎的每个矩阵处理器可以并行执行图7的过程。例如,矩阵处理器313和

351至357各自都对不同的矩阵参量并行执行图7的过程,尽管每个矩阵处理器都可能处于不同的处理步骤以错开它们相应运算的完成。在一些实施例中,该过程用于使用数据矩阵和权重矩阵执行卷积。在一些场景中,输入矩阵是较大矩阵的切片。在各个实施例中,图7的过程可以经由控制单元由矩阵计算指令启动。该指令可以指定两个矩阵操作数(例如,数据和权重矩阵的存储器或寄存器位置)、可配置偏置、浮点格式、以及指定累加器,以存储矩阵计算结果。在一些实施例中,指定累加器在矩阵计算开始之前归零。在一些实施例中,指定累加器是图3的输出累加器329或331。在一些实施例中,在图6的605处执行图7的过程。

[0076] 在701处,接收数据输入矩阵。例如,与训练传感器数据相对应的数据输入矩阵的元素被线性化并存储在矩阵处理器的数据输入阵列中。在一些实施例中,数据输入矩阵存储在诸如图3的矩阵处理器313的数据输入阵列321之类的数据输入阵列中。每个数据输入阵列能够存储整个线性化矩阵,以供通过矩阵计算单元对对应矩阵处理器进行处理。因此,能够将两个 8×8 矩阵相乘的矩阵处理器使用能够存储输入 8×8 数据矩阵的所有64个元素的数据输入阵列。例如,在一些实施例中,每个数据输入阵列是64字节的,并且存储每个元素作为8位浮点数。浮点数的格式可以使用图4的格式400或410,并且包括可配置偏置。可配置偏置可以由矩阵指令和/或寄存器指定。所接收的数据输入矩阵可以从寄存器或从诸如SRAM之类的存储器接收。在一些实施例中,发出一个或多个读取以将整个数据输入矩阵加载到矩阵处理器,但是整个矩阵并非立即可用。例如,对于经切片矩阵,一些行(或列)的数据在数据可用之前可能需要附加延迟。因此,数据输入阵列的数据可能会逐个到达。在一些实施例中,单次读取足以加载整个数据输入矩阵。在一些实施例中,数据输入矩阵是梯度输入矩阵。

[0077] 在703处,接收权重输入矩阵。例如,与滤波器的机器学习权重相对应的权重输入矩阵的元素被线性化并存储在矩阵处理器的权重输入阵列中。在一些实施例中,权重输入矩阵存储在诸如图3的矩阵处理器313的权重输入阵列323之类的权重输入阵列中。每个权重输入阵列能够存储整个线性化矩阵,以便通过矩阵计算单元对对应矩阵处理器进行处理。因此,能够将两个 8×8 矩阵相乘的矩阵处理器使用能够存储输入 8×8 权重矩阵的所有64个元素的权重输入阵列。例如,在一些实施例中,每个权重输入阵列是64字节的,并且将每个元素存储为8位浮点数。浮点数的格式可以使用图4的格式400或410,并且包括可配置偏置。可配置偏置可以由矩阵指令和/或寄存器指定。所接收的权重输入矩阵可以从寄存器或从诸如SRAM之类的存储器接收。在一些实施例中,发出一个或多个读数以将整个权重输入矩阵加载到矩阵处理器,但是整个矩阵并非立即可用。例如,对于经切片矩阵,在权重数据可用之前,一些行(或列)的权重数据可能需要附加延迟。因此,权重输入阵列的权重数据可能会逐个到达。在一些实施例中,单次读取足以加载整个权重输入矩阵。在一些实施例中,权重输入矩阵是梯度输入矩阵。

[0078] 在705处,一对向量参量被加载到矩阵计算单元中。从每个输入矩阵,与行相对应的向量和与列相对应的向量作为输入参量加载到诸如图3的矩阵计算单元325之类的矩阵计算单元。作为加载过程的一部分,跨越整个矩阵计算单元复制列向量,并且向下整个矩阵计算单元复制行向量。例如,与权重输入矩阵的一列相对应的整个向量加载到计算单元中。跨越整个行复制列向量的每个元素。因此, 8×8 矩阵计算单元的每一列接收相同8元素列向量,并且加载到矩阵计算单元的一行的每个单元的值均相同。同样,与数据输入矩阵的一行

相对应的整个向量加载到计算单元中,并且向下整个列复制行向量的每个元素。因此, 8×8 矩阵计算单元的每一行接收相同8元素列向量,并且加载到矩阵计算单元的一列的每个单元的值均相同。对于 8×8 矩阵计算单元,加载八分之一的输入矩阵元素。在705处,来自每个输入矩阵的卸载向量对被加载到矩阵计算单元中。通过步骤705的每个后续循环从输入权重和数据矩阵加载下一可用列和行。因此, 8×8 矩阵需要至少8个周期才能完成加载,而 4×4 矩阵需要至少4个周期才能完成加载。

[0079] 在707处,经加载的向量的值相乘。对于诸如图3的计算单元327之类的矩阵计算单元的每个计算单元,使用加载在对应计算单元处的元素执行矩阵乘法。在各个实施例中,对两个8位浮点值执行乘法,并且存储作为高位浮点值,以防止上溢并且维持精确性。在一些实施例中,高位浮点格式是图5的21位浮点格式。在一些实施例中,高位浮点格式是27位浮点格式以进一步减少量化结果的准确性的损失。对于 8×8 矩阵计算单元,64个计算单元中的每个计算单元都执行矩阵乘法。

[0080] 在709处,乘法结果累加到指定累加器中。例如,每个计算单元在707处的乘法结果各自累加到矩阵处理器的累加器中的一个累加器中。在一些实施例中,矩阵处理器包括诸如图3的两个输出累加器329和331之类的不止一个累加器。这是有益的,以使矩阵处理器可以交错不同矩阵运算的操作。在一些实施例中,每个计算单元包括累加器,该累加器将累加器中与该计算单元相对应的元素的当前值与该单元的矩阵乘法的结果相加。在各个实施例中,累加器的大小被设计为存储矩阵的每个元素的累加结果。因此, 8×8 矩阵计算单元的每个累加器具有至少64个元素。在一些实施例中,与707处的乘法结果相似,累加器的元素使用比矩阵处理器的输入更高位的浮点值以防止上溢并且维持精确性。在一些实施例中,高位浮点格式是图5的21位浮点格式或另一高位浮点格式。在一些实施例中,用于 8×8 矩阵计算单元的累加器是168字节以允许64个元素,每个元素存储21位浮点数。

[0081] 在711处,确定针对矩阵运算是否剩余附加向量。例如,为了将两个矩阵相乘,每个时钟周期最多加载来自权重输入矩阵的一个列和来自数据输入矩阵的一个行。为了完成整个矩阵乘法,必须加载每一列和每一行。 8×8 矩阵需要至少8个周期才能将两个输入矩阵完全加载到矩阵计算单元中。同样, 4×4 矩阵需要至少4个周期才能将两个输入矩阵完全加载到矩阵计算单元中。在剩余有要加载的附加向量的情况下,处理继续返回到705。在剩余没有要加载的附加向量(两个整个输入矩阵都已经被加载)的情况下,矩阵乘法完成并且处理继续到713。

[0082] 在713处,矩阵结果从指定累加器加载到输出阵列中。由于矩阵计算完成,所以矩阵结果存储在指定累加器中。在一些实施例中,矩阵的元素作为21位浮点值存储在指定累加器中。因此,对于 8×8 矩阵,累加器存储64个值,大小为168个字节。在一些实施例中,需要多个移动操作来将结果从累加器移动到输出阵列,诸如图3的输出阵列315。在一些实施例中,输出阵列和到输出阵列的总线的宽度为64个字节。累加器结果从21位浮点值转换为16位浮点值,该16位浮点值可以存储在两个64字节部件中。使用 8×8 结果矩阵为例,从矩阵处理器的累加器中移动结果需要两个移动操作。例如,上移操作用于将累加器的(与矩阵的32个元素相对应的)高位作为16位浮点值移入64位输出阵列。一旦在输出阵列中移动,32个元素就可以存储在诸如图3的后处理单元寄存器文件307的寄存器中的一个寄存器之类的寄存器中或移动到存储器。随后,下移操作用于将累加器的(与矩阵的其余32个元素相对应

的)低位作为16位浮点值移入64位输出阵列。一旦在输出阵列中,剩余32个元素就可以存储在寄存器中。在各个实施例中,需要两个或更多个运算来将矩阵结果移出矩阵处理器。通过将21位浮点值转换为16位浮点值,仅需两个移动操作。在一些实施例中,这些值可以作为8位浮点值、16位浮点值或32位浮点值移出。在所描述的示例中,值作为16位值移出以供诸如图3的后处理单元317之类的后处理单元进行后续处理。在一些实施例中,后处理单元是向量计算引擎。在各个实施例中,输出阵列连接到节点引擎的每个矩阵处理器的累加器,并且充当多路复用器以从不同矩阵处理器接收移动的结果(例如,上移指令和下移指令)。

[0083] 图8是图示了用于执行多个交错矩阵计算的过程的实施例的流程图。图8的过程被诸如图3的节点引擎300的矩阵处理器313和351至357之类的矩阵处理器用来交错多个矩阵计算,诸如两个矩阵乘法运算。交错矩阵计算中的每个交错矩阵计算可以使用多个中间矩阵乘法来实现,其中中间乘法的结果用于计算较大矩阵计算。为了提高处理带宽和效率,每个中间矩阵乘法的结果都存储在矩阵处理器中,并且当交错备选矩阵运算时并未清除。不同的矩阵运算可以是不同的,并且每个矩阵运算都具有非重叠矩阵运算数。

[0084] 在一些实施例中,节点引擎的每个矩阵处理器一次可以处理一个以上的矩阵运算,一个矩阵运算与矩阵处理器的每个输出累加器相对应。在一些实施例中,交错多个矩阵运算的能力允许对非常大的矩阵执行矩阵乘法运算。较大的矩阵被切片成适配矩阵处理器的输入阵列的较小矩阵,并且组合较小矩阵的矩阵乘法的结果。在各个实施例中,例如,在等待存储器读取完成的同时,能够交错多个矩阵运算通过利用矩阵计算单元来增加矩阵处理器的带宽和性能。因此,当第一相关矩阵运算集合的未决矩阵运算的输入操作数不可用时(例如,由于存储器读取的延迟)但第二相关矩阵运算集合的未决矩阵运算的输入操作数可用时,第二相关矩阵运算集合可以利用矩阵计算单元。通过利用多个累加器,矩阵计算单元可以通过将中间结果存储在专用于特定相关矩阵运算集合的累加器中来在多个矩阵计算之间切换。在一些实施例中,数据输入阵列是图3的数据输入阵列321,权重输入阵列是图3的权重输入阵列323,并且多个累加器是图3的输出累加器329和331。尽管相对于图3的矩阵处理器313示出了两个累加器,但是可以包括附加累加器以允许交错附加矩阵运算。

[0085] 图8的过程是图7的过程的特殊变型,该特殊变型利用多个权重输入阵列操作数、多个数据输入阵列操作数、以及多个输出累加器,以支持交错两个矩阵乘法运算。如关于图7所描述的,图8的过程以类似方式实现了图7的步骤,这些步骤包括:跨越矩阵计算单元加载列向量,向下矩阵计算单元加载行向量,将操作数乘以计算单元,以及将乘法结果在指定累加器中累加;但注意不要混合或擦除两个交错矩阵运算的中间结果。在一些实施例中,在图6的605处执行图8的过程。

[0086] 在801处,确定矩阵处理器是否可以接收附加矩阵运算指令。在图8的示例中,矩阵处理器能够交错两个矩阵运算。确定被执行的过程中当前是否存在两个矩阵运算。在矩阵处理器可以接收附加矩阵运算指令的情况下,则处理继续到803。例如,矩阵处理器可以接收附加矩阵运算指令,这是因为它只处于处理单个矩阵运算的中间或处于空闲而非处理任何矩阵运算。在矩阵处理器不能接收附加矩阵运算指令的情况下,处理循环回到801,直至矩阵处理器可用于接收新矩阵运算指令。例如,矩阵处理器当前处于处理两个矩阵运算的中间,并且不能接收另一运算,直至当前运算中的至少一个当前运算完成。在一些实施例中,就绪信号被发布到控制单元以用信号通知矩阵处理器准备好接收附加指令。

[0087] 在803处,矩阵处理器接收矩阵指令并且发出对相关矩阵运算的读取请求。例如,矩阵处理器接收具有与两个输入矩阵相对应的两个操作数的矩阵乘法指令。发出对矩阵操作数的值的读取。这些值可以从寄存器和/或存储器中读取。例如,矩阵参量可以指定寄存器和/或存储器中的地址。在一些实施例中,因为存储器读取可能花费多个时钟周期才能使数据可用,所以存储器读取可能停止矩阵计算。在一些实施例中,由于矩阵没有顺序地存储在存储器中,所以可以发出多个存储器读取。这可能是将较大矩阵切片成较小矩阵操作数的结果。

[0088] 在一些实施例中,所接收的指令指定特定累加器来存储矩阵结果。为了交错多个矩阵运算,每个运算利用其自身的累加器。指定累加器用于存储中间和最终矩阵结果。在一些实施例中,指定累加器使用比用于输入操作数的格式更高位的浮点格式来存储中间结果。当结果被量化时,高位格式使准确性的损失最小。

[0089] 在各个实施例中,当与矩阵操作数相对应的数据可用时,接收值,并且为矩阵处理器准备这些值。在一些实施例中,矩阵操作数对于矩阵处理器而言太大,并且执行多个中间矩阵运算以完成矩阵指令。在数据不可用的情况下,矩阵计算单元可能会停止并空闲。代替保持空闲,只要用于第二运算的数据可用,就可以执行第二矩阵运算。

[0090] 在803处,处理继续到801和805两者。处理循环回到801以获取新指令,同时还继续到805以执行在803处接收的指令。在各个实施例中,获取新指令与处理当前矩阵运算并行发生。在一些实施例中,到801和805的两个处理分支使用基于流水线的途径来实现。

[0091] 在805处,确定数据是否准备好用于当前矩阵运算。例如,要从当前矩阵运算的矩阵操作数加载的元素必须可用于加载到矩阵计算单元的计算单元。在一些实施例中,加载到矩阵计算单元中的数据是针对矩阵计算单元的输入阵列确定大小的矩阵操作数的切片。对于权重输入阵列,元素的未决列必须准备就绪。对于数据输入阵列,未决元素行必须准备就绪。在当前矩阵运算的权重列和数据行的元素可用的情况下,处理继续到807。在当前矩阵运算的未决元素不可用的情况下,处理继续到813。例如,由于存储器读取和/或缓存错过的延迟,所以未决元素可能不可用。代替在等待数据变得可用时停止,矩阵计算单元可以潜在地用于备选矩阵运算。

[0092] 在807处,来自当前矩阵运算的权重列和数据行的值加载到相应计算单元,对值执行计算操作,并且计算结果累加到指定累加器中。在一些实施例中,计算操作是与将来自两个不同矩阵的元素相乘的乘法操作。在一些实施例中,关于图7的步骤701、703、705、707、709和/或711,对807处的过程进行描述。例如,值被加载为具有可配置偏置的8位浮点值。例如乘法和累加之类的计算的结果作为21位浮点格式存储在第一累加器中。在一些场景中,在807处执行与矩阵运算有关的附加配置,诸如清除累加器,确定浮点格式,和/或确定浮点格式的可配置偏置等。

[0093] 在809处,确定当前矩阵运算的矩阵指令是否完成。在矩阵指令完成的情况下,处理继续到811。在矩阵指令未完成的情况下,处理继续到805,其中确定当前矩阵运算的附加数据是否准备好通过矩阵计算单元加载和处理。在一些实施例中,关于图7的步骤711,对809处的过程进行描述。

[0094] 在一些备选实施例(未示出)中,在矩阵指令未完成的情况下,处理继续到813,其中确定备选矩阵运算是否未决以及未决备选矩阵运算的数据是否准备好通过矩阵计算单

元加载和处理。在该备选实施例下,代替完成当前矩阵运算,只要数据可用,矩阵计算单元就继续在两个不同的矩阵运算之间来回交替,只要存在两个并发矩阵运算即可。

[0095] 在811处,存储在指定累加器中的矩阵结果被加载到输出阵列中。由于一些实施例使用诸如21位浮点格式或27位浮点格式之类的更高位深的浮点格式存储所得矩阵,所以结果移出矩阵处理器可能需要多个移动指令。在一些实施例中,通过首先将矩阵元素转换为16位浮点值,矩阵结果经由输出阵列移到两个64字节寄存器中。在一些实施例中,关于图7的步骤713,对811处的过程进行描述。处理循环回到步骤805,其中如果未决,则矩阵处理器准备好开始矩阵运算或在备选矩阵运算上取得进展。

[0096] 在一些备选实施例(被示为虚线)中,处理继续到813,其中确定备选矩阵运算是否未决以及用于未决备选矩阵运算的数据是否准备好通过矩阵计算单元加载和处理。在该备选实施例下,一旦当前矩阵指令完成,在存在未决完成的备选矩阵运算的情况下,矩阵计算单元切换到备选矩阵运算。

[0097] 在813处,确定备选矩阵运算是否未决以及用于未决备选矩阵运算的数据是否准备好通过矩阵计算单元加载和处理。例如,在处理第一矩阵运算的同时在803处接收到第二矩阵运算的情况下,未决完成的第二矩阵运算发出对其对应矩阵参量的读取。确定是否存在未决的第二备选矩阵运算以及其数据是否准备好加载到矩阵计算单元中。在用于备选矩阵运算的操作数数据可用的情况下,处理继续到815。在一些实施例中,操作数数据是大小被设计为用于矩阵计算单元的输入阵列的较大操作数矩阵的切片。对于权重输入阵列,未决元素列必须准备就绪。对于数据输入阵列,未决元素行必须准备就绪。在不存在未决备选矩阵运算或备选矩阵运算的未决元素不可用的情况下,则处理继续到805。例如,由于来自存储器读取的延迟和/或高速缓存错失,所以未决元素可能不可用。代替在等待数据变得可用的同时停止,再次检查与当前矩阵运算相对应的数据的可用性。具有可用数据的第一矩阵运算将其数据加载到矩阵计算单元中以供处理。

[0098] 在815处,包括矩阵计算单元的矩阵处理器被切换为对未决完成的备选矩阵运算执行处理。备选矩阵运算现在被指定为当前矩阵运算,而先前的当前矩阵运算被指定为备选矩阵运算。由于第一矩阵运算可能已停止(或在一些实施例中,已完成),所以矩阵计算单元现在致力于未决完成的第二矩阵运算。在各个实施例中,视情况而定,对应输出累加器被指定为先前中间结果的源以及用于累加中间和最终结果的目的地。处理继续到807,其中对新指定的当前矩阵运算进行计算进展。

[0099] 尽管为了清楚理解的目的,已经对前述实施例进行了一些详细描述,但是本发明不限于所提供的细节。存在实现本发明的许多备选方式。所公开的实施例是说明性的而非限制性的。

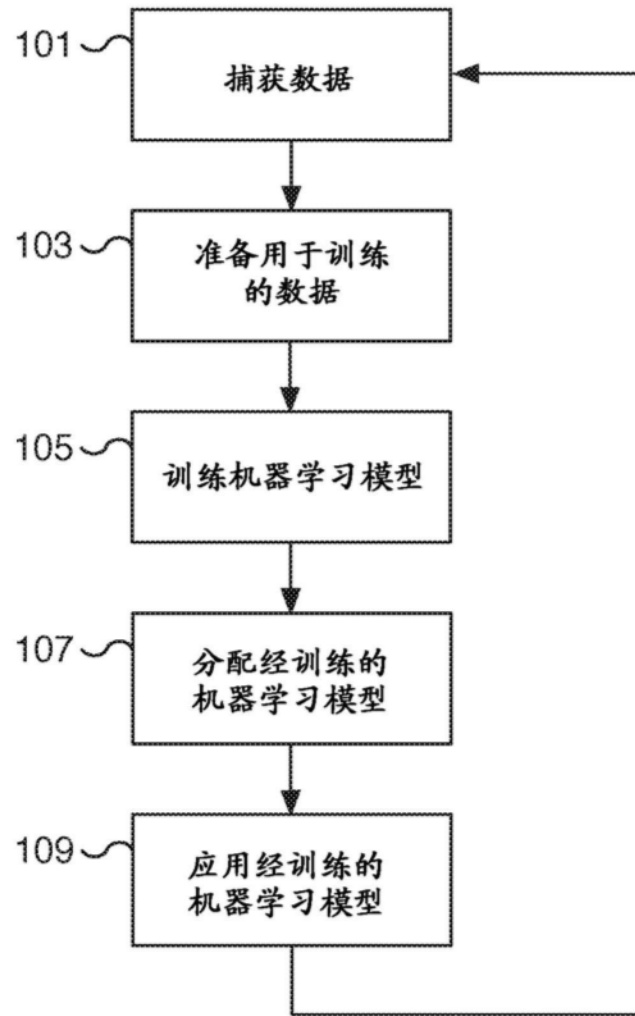


图1

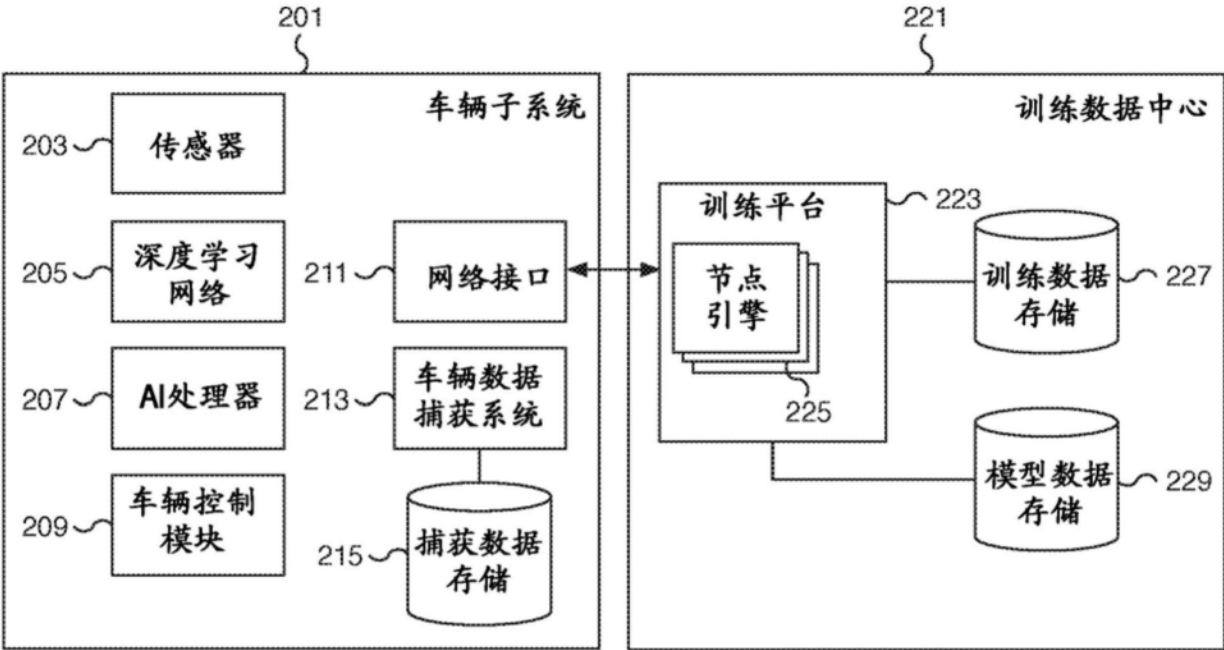


图2

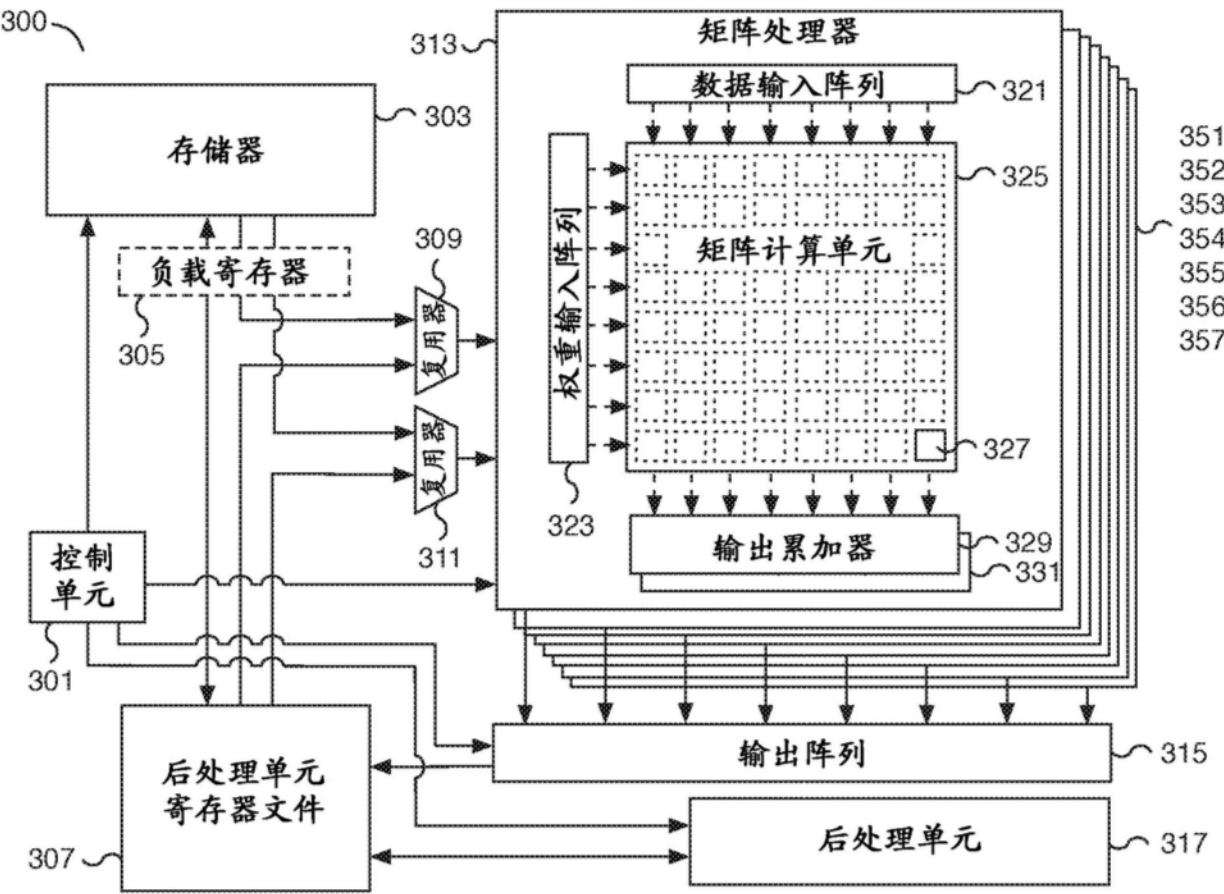


图3

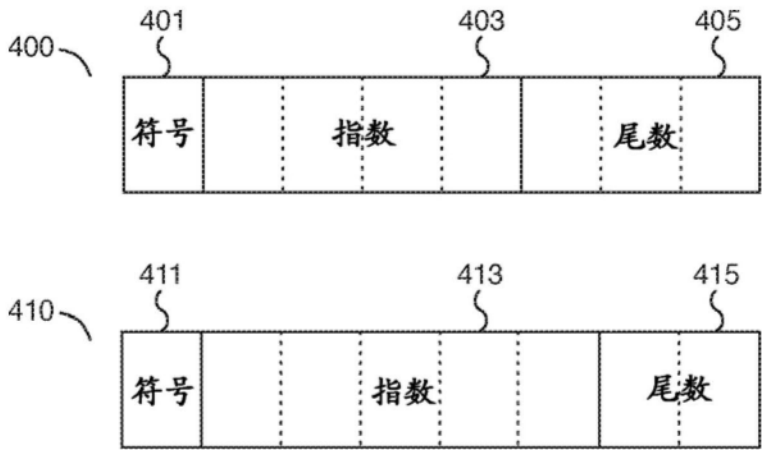


图4

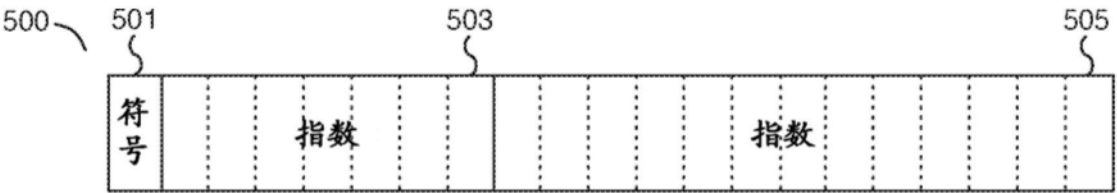


图5

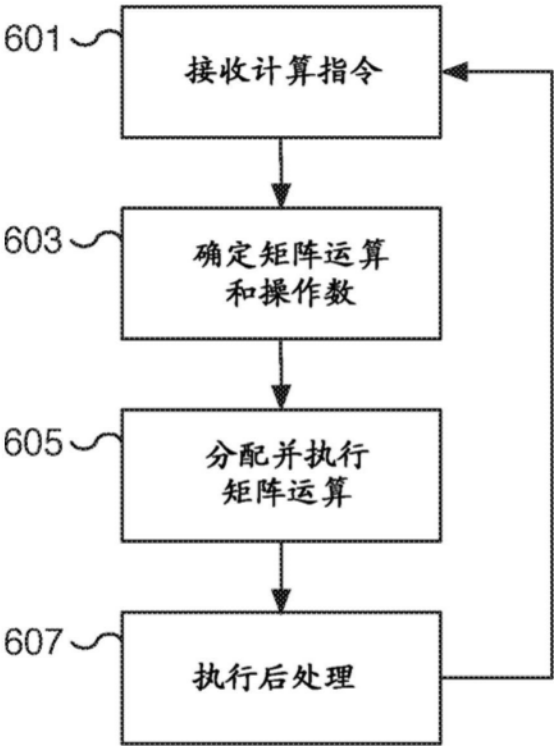


图6

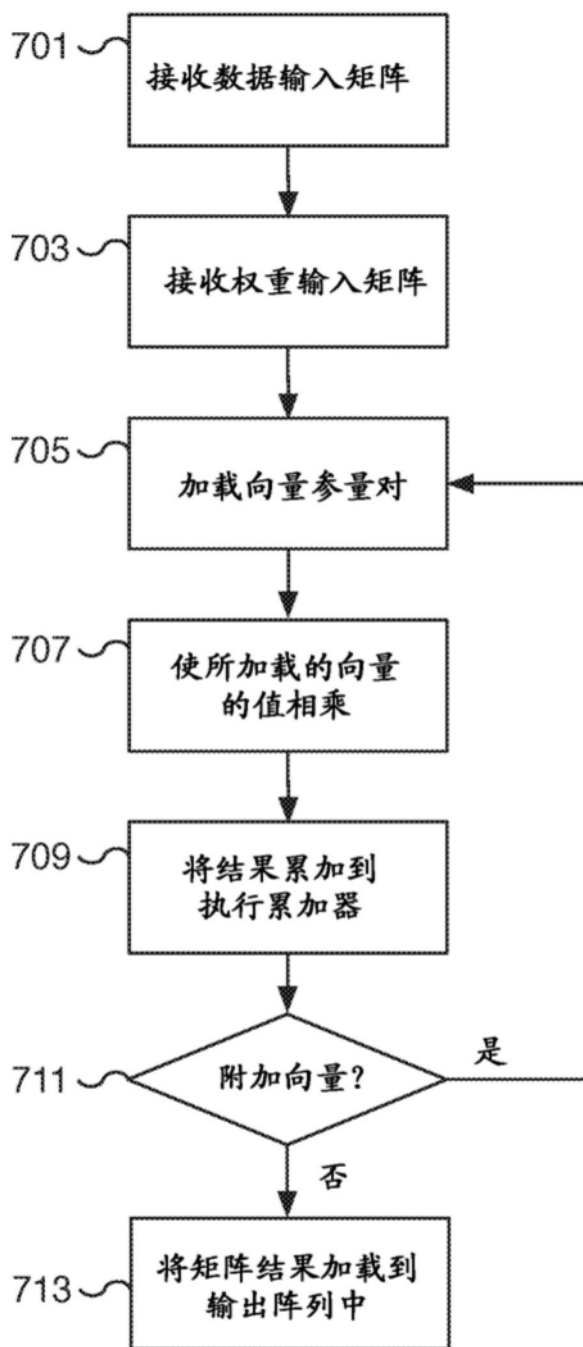


图7

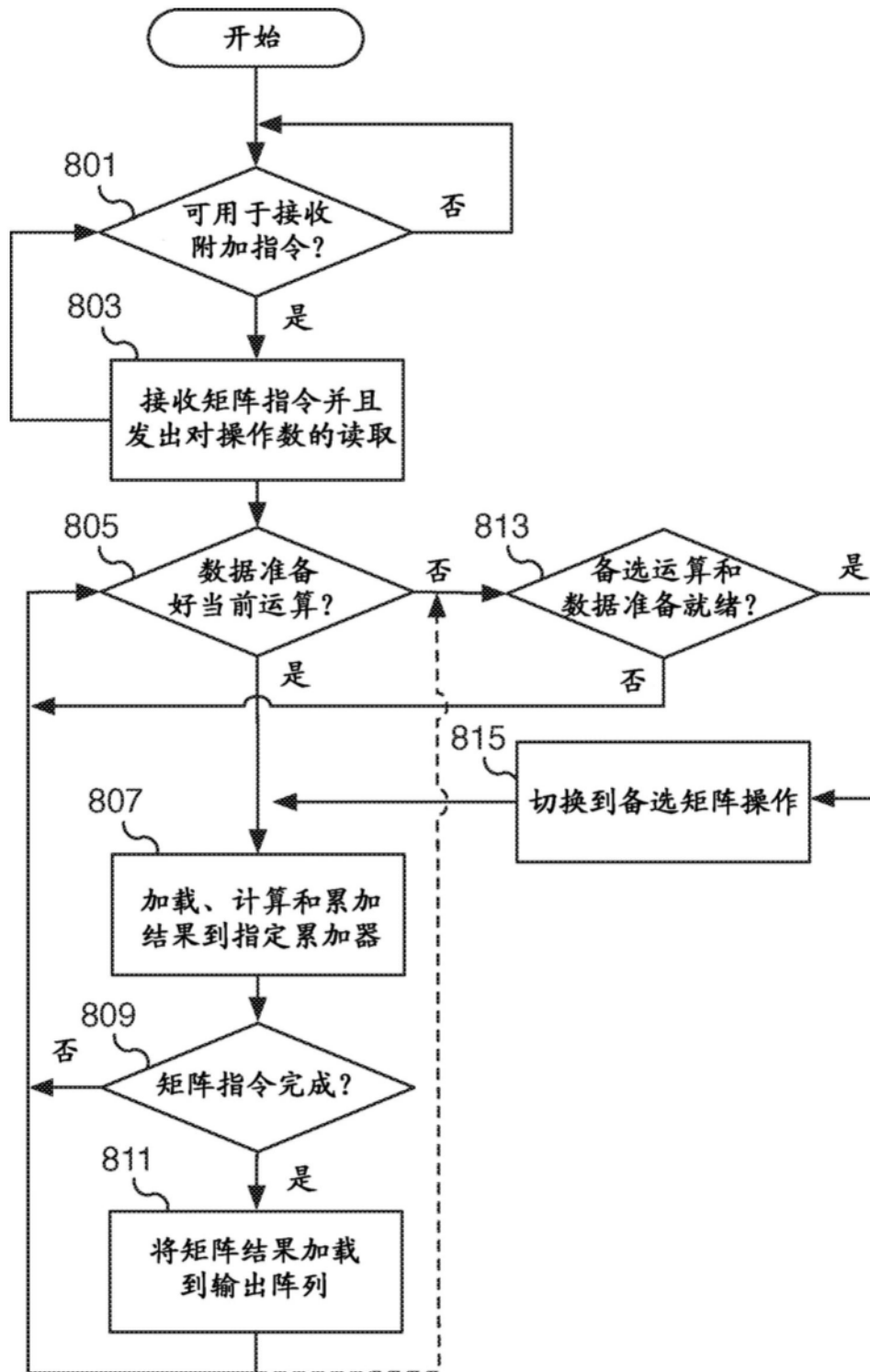


图8