

一种芯片及基于其的数据处理方法

技术领域

[01]本说明书实施例涉及数据处理技术领域，更具体地，涉及一种芯片及基于其的数据处理方法。

5 背景技术

[02]随着 AI 技术的发展，其对计算量的需求越来越大，特别是作为线上服务的神经网络预测对处理时延有着相当苛刻的要求，在该情况下，服务器 CPU 上跑的软件通常满足不了这种要求。现有技术中，通过硬件加速的方法来缩短这种处理时延，在该情况中，需要硬件具备一定的灵活性，以满足神经网络结构的升级以及待处理数据长短的变化。

- 10 当程序遇到一大段计算量时，它会把待处理的数据通过 PCIE 总线发给芯片，芯片算完之后再通过 PCIE 总线将结果传回程序内存。芯片上的运算单元通常被设计成较通用的，通过服务器 CPU 下发的控制参数来指定具体怎么运算，比如通过一个通用的卷积运算单元来处理各种卷积操作。但如果一段程序包含有多次芯片运算单元的调用，服务器 CPU 和芯片之间数据交互的开销就会非常大。目前通用的做法是在芯片上把各个运算单
- 15 元如卷积的控制设计成格式化的二进制指令，然后把神经网络的各层调用映射到这些指令上、并集成成一个序列，再下放到芯片上，芯片上的逻辑解析每一条指令并启动相应的运算单元。

[03]因此，需要一种更有效的基于芯片的数据处理方案。

发明内容

- 20 [04]本说明书实施例旨在提供一种更有效的基于芯片的数据处理方案，以解决现有技术中的不足。

[05]为实现上述目的，本说明书一个方面提供一种基于芯片的数据处理方法，所述芯片包括嵌入式 CPU 和多个算子单元，用于执行计算任务，所述方法在芯片端执行，包括：

[06]获取待处理数据；以及

- 25 [07]通过由所述嵌入式 CPU 执行以下步骤而进行所述计算任务的计算：配置所述多个算子单元，以使得所述多个算子单元与该计算任务中包括的不同类型的计算分别对应，以

及，基于所述待处理数据，通过调用所述多个算子单元，进行所述计算任务的计算，以获取处理结果。

[08]本说明书另一方面提供一种基于芯片的数据处理方法，所述芯片包括嵌入式 CPU 和多个算子单元，用于执行神经网络的预测计算，所述方法在芯片端执行，包括：

5 [09]获取待处理数据；以及

[10]基于所述待处理数据，执行所述神经网络中的每一层的计算，以获取处理结果，其中，通过由所述嵌入式 CPU 执行以下步骤而进行所述神经网络的每一层的计算：配置所述多个算子单元，以使得所述多个算子单元与该层中包括的不同类型的计算分别对应，以及，基于神经网络上一层中各个神经元的值，通过调用所述多个算子单元，执行该层
10 中每个神经元处的计算，以获取该层中每个神经元的值。

[11]在一个实施例中，在所述数据处理方法中，所述芯片与服务器连接。

[12]在一个实施例中，在所述数据处理方法中，所述待处理数据通过所述服务器传送至所述芯片。

[13]在一个实施例中，在所述数据处理方法中，所述嵌入式 CPU 通过运行基于程序的汇编指令而执行所述步骤，其中，所述程序由所述服务器传送至所述芯片。
15

[14]在一个实施例中，在所述数据处理方法中，所述程序为 C 语言程序。

[15]在一个实施例中，在所述数据处理方法中，调用所述多个算子单元包括，通过 C 语言程序中的函数调用，调用所述多个算子单元。

[16]在一个实施例中，在所述数据处理方法中，所述芯片还包括 DMA 模块，所述方法还包括，在获取所述处理结果之后，通过所述嵌入式 CPU 控制所述 DMA 模块，以将所述处理结果写入所述服务器的内存中。
20

[17]在一个实施例中，在所述数据处理方法中，调用所述多个算子单元包括，异步地调用所述多个算子单元。

[18]在一个实施例中，在所述数据处理方法中，配置所述多个算子单元包括，对所述待处理数据的长度进行判断，并基于所述长度获取发送给各个算子单元的参数。
25

[19]在一个实施例中，在所述数据处理方法中，所述芯片用于执行多种神经网络的预测计算，其中，获取待处理数据包括，获取待处理数据及其对应的神经网络的类型，其中，基于所述待处理数据，执行所述神经网络中的每一层的计算，以获取处理结果包括，基

于所述待处理数据及其对应的神经网络的类型，执行该类型的神经网络中的每一层的计算，以获取处理结果。

[20]本说明书另一方面提供一种芯片，用于执行计算任务，所述芯片包括：

[21]多个算子单元，以及

- 5 [22]嵌入式 CPU，配置为，通过执行以下步骤而进行所述计算任务的计算：配置所述多个算子单元，以使得所述多个算子单元与该计算任务中包括的不同类型的计算分别对应，以及，基于从所述芯片的外部输入的待处理数据，通过调用所述多个算子单元，进行所述计算任务的计算，以获取处理的结果。

[23]本说明书另一方面提供一种芯片，用于执行神经网络的预测计算，所述芯片包括：

- 10 [24]多个算子单元,以及

[25]嵌入式 CPU，配置为，基于从所述芯片的外部输入的待处理数据，执行所述神经网络中的每一层的计算，以获取处理结果，其中，通过由所述嵌入式 CPU 执行以下步骤而进行所述神经网络的每一层的计算：配置所述多个算子单元，以使得所述多个算子单元与该层中包括的不同类型的计算分别对应，以及，基于神经网络上一层中各个神经元的值，通过调用所述多个算子单元，执行该层中每个神经元处的计算，以获取该层中每个神经元的值。

15

[26]在一个实施例中，所述芯片与服务器连接。

[27]在一个实施例中，在所述芯片中，所述待处理数据通过所述服务器传送至所述芯片。

20

[28]在一个实施例中，在所述芯片中，所述嵌入式 CPU 通过运行基于程序的汇编指令而执行所述步骤，其中，所述程序由所述服务器传送至所述芯片。

[29]在一个实施例中，在所述芯片中，所述程序为 C 语言程序。

[30]在一个实施例中，在所述芯片中，所述嵌入式 CPU 还配置为，通过 C 语言中的函数调用，调用所述多个算子单元。

25

[31]在一个实施例中，在所述芯片中，所述芯片还包括 DMA 模块，所述嵌入式 CPU 还配置为，在获取所述处理结果之后，通过控制所述 DMA 模块，以将所述处理结果写入所述服务器的内存中。

[32]在一个实施例中，在所述芯片中，所述嵌入式 CPU 还配置为，异步地调用所述多个算子单元。

[33]在一个实施例中,在所述芯片中,所述嵌入式 CPU 还配置为,对所述待处理数据的长度进行判断,并基于所述长度获取发送给各个算子单元的参数。

[34]在一个实施例中,在所述芯片中,所述芯片为 ASIC 芯片或 FPGA 芯片。

[35]在一个实施例中,所述芯片用于执行多种神经网络的预测计算,其中,所述嵌入式 CPU 还配置为,基于从所述芯片的外部输入的待处理数据及其对应的神经网络的类型,执行所述类型的神经网络中的每一层的计算,以获取处理结果。

[36]本说明书另一方面提供一种计算设备,包括存储器和处理器,其特征在于,所述存储器中存储有可执行代码,所述处理器执行所述可执行代码时,实现上述任一项数据处理方法。

10 [37]通过根据本说明书实施例的基于芯片的数据处理方案,因为引入了嵌入式 CPU,天然在芯片内部就可以完成较细算子的控制;C 语言函数调用的方式又大大节省了程序指令的空间,并且,嵌入式 CPU 上 C 程序天然就可以完美地进行控制类型的操作,比如各种 if/for/while 等嵌套;嵌入式 CPU 上的 C 程序可以对输入数据的长度做判断,并灵活产生计算过程中下发给各个算子单元的参数;另外,当所述神经网络结构升级时,可
15 通过服务器向芯片传入新的程序,以进行更新的神经网络的预测计算。

附图说明

[38]通过结合附图描述本说明书实施例,可以使得本说明书实施例更加清楚:

[39]图 1 示出根据本说明书实施例的数据处理系统的 100 的示意图;

[40]图 2 示出根据本说明书实施例的一种基于芯片的数据处理方法;

20 [41]图 3 示出了神经网络的示意图;以及

[42]图 4 示出根据本说明书实施例的一种芯片 400。

具体实施方式

[43]下面将结合附图描述本说明书实施例。

[44]图 1 示出根据本说明书实施例的数据处理系统的 100 的示意图。如图 1 所示,系统
25 100 包括服务器 11 和芯片 12,其例如通过 PCIE 总线连接。服务器 11 中包括服务器 CPU111 和内存 112。芯片 12 中包括嵌入式 CPU121、多个算子单元 122、以及 DMA 模

块 123。该芯片 12 用于执行计算任务，例如神经网络的预测计算。服务器 11 在需要通过调用芯片 12 以进行例如神经网络的预测计算时，其例如通过服务器 CPU111 将该神经网络预测的待处理数据经 PCIE 总线从内存传到芯片中。芯片 12 中的嵌入式 CPU121 通过运行程序调度各个算子单元 122 基于所述待处理数据完成神经网络每一层的计算，

5 从而获取最终的处理结果，即神经网络的预测结果。在获取处理结果之后，由嵌入式 CPU121 控制 DMA 模块经 PCIE 总线将该处理结果传回服务器内存 112。

[45]图 2 示出根据本说明书实施例的一种基于芯片的数据处理方法，所述芯片包括嵌入式 CPU 和多个算子单元，用于执行计算任务，所述方法在芯片端执行，包括：

[46]在步骤 S202，获取待处理数据；以及

10 [47]在步骤 S204，通过由所述嵌入式 CPU 执行以下步骤而进行所述计算任务的计算：配置所述多个算子单元，以使得所述多个算子单元与该计算任务中包括的不同类型的计算分别对应，以及，基于所述待处理数据，通过调用所述多个算子单元，进行所述计算任务的计算，以获取处理结果。

[48]首先，在步骤 S202，获取待处理数据。在一个实施例中，所述芯片与服务器连接。

15 所述服务器在调用该芯片进行计算时，将该待处理数据传送至该芯片。可以理解，该芯片不限于与服务器连接，例如，其可以为独立的用于进行任务计算的硬件，并通过自身的输入模块获取待处理数据。

[49]在一个实施例中，该芯片位于 PCIE 板卡上，为 ASIC 芯片或 FPGA 芯片，所述服务器通过服务器 CPU 经 PCIE 总线将所述待处理数据传送到芯片。在一个实施例中，所述

20 服务器通过其中运行的应用程序将所述待处理数据传送到芯片。在一个实施例中，所述芯片可同时接收多个应用程序传送的待处理数据，所述芯片通过嵌入式 CPU 进行对所述多个待处理数据的收集。

[50]在步骤 S204，通过由所述嵌入式 CPU 执行以下步骤而进行所述计算任务的计算：配置所述多个算子单元，以使得所述多个算子单元与该计算任务中包括的不同类型的计

25 算分别对应，以及，基于所述待处理数据，通过调用所述多个算子单元，进行所述计算任务的计算，以获取处理结果。

[51]嵌入式 CPU 通过运行基于程序的汇编指令而进行所述计算。所述程序例如为 C 语言程序。所述芯片例如通过其中的 SOC（片上系统编译器），将所述程序翻译成汇编语言程序。其中，所述程序由所述服务器传送至所述芯片。

[52]在进行计算时,所述嵌入式 CPU 首先通过运行程序配置所述多个算子单元,以使得所述多个算子单元与该计算任务中包括的不同类型的计算分别对应,所述不同类型的计算例如包括:矩阵相乘的计算、向量相加的计算、非线性运算等等。然后,嵌入式 CPU 通过继续运行程序以执行具体的计算过程,其中的计算基于所述待处理数据进行的,并且,在计算中遇到各种类型的计算时,通过调用各个相应的算子单元完成该计算。在复杂的计算任务中,可能需要经过多轮的计算,才能获取最终的处理结果,其中,在每轮计算中,都通过预设的程序重新配置算子单元,以适用于该轮的计算需要。

[53]在一个实施例中,所述计算任务例如为神经网络的预测计算。在该情况中,所述嵌入式 CPU 基于所述待处理数据,执行所述神经网络中的每一层的计算,以获取处理结果,即神经网络的预测结果。其中,通过由所述嵌入式 CPU 执行以下步骤而进行所述神经网络的每一层的计算:配置所述多个算子单元,以使得所述多个算子单元与该层中包括的不同类型的计算分别对应,以及,基于神经网络上一层中各个神经元的值,通过调用所述多个算子单元,执行该层中每个神经元处的计算,以获取该层中每个神经元的值。

[54]也就是说,对于神经网络的每一层,嵌入式 CPU 都会重新配置算子单元,并依次进行该层中各个神经元处的计算。从 C 语言程序的角度看,程序会进行两层的循环。在外层循环中,程序循环 n 次,其中 n 是神经网络除输入层之外的层数,以对神经网络除输入层的每一层进行计算。而在内层循环中,即在对每层的计算中,程序循环 m 次,其中 m 是神经元的个数,以对每层中的每个神经元进行计算。

[55]图 3 示出了神经网络的示意图,如图 3 中所示,左侧的层为输入层,其包括 (1) 和 (2) 两个输入神经元,中间的层为隐藏层,其包括 (3)、(4) 和 (5) 三个神经元,右侧的层为输出层,其包括输出神经元 (6)。其中,在中间层中,在神经元 (3) 处,其针对上一层神经元的参数分别为 W_{13} 、 W_{23} ,在神经元 (4) 处,参数包括 W_{14} 、 W_{24} ,在神经元 (5) 处,参数包括 W_{15} 、 W_{25} ,以及在神经元 (6) 处,参数包括 W_{36} 、 W_{46} 和 W_{56} 。

[56]在进行对图 3 所示的神经网络的计算中,首先,对中间层进行计算,在该层的计算中,例如包括矩阵相乘的计算、向量相加的计算等,从而,嵌入式 CPU 可配置与矩阵相乘和向量相加分别对应的两个算子单元,并通过程序循环对神经元 (3)、(4) 和 (5) 分别进行计算。例如,在对神经元 (3) 的计算中,可将从输入层获取的待处理数据和神经元 (3) 处的参数 (W_{13} 、 W_{23}) 输入上述两个算子单元中,以获得该神经元 (3) 处

的中间结果，其可视为神经元（3）处的值。

[57]在完成对中间层的计算（即，获取神经元（3）、（4）和（5）各自的值）之后，嵌入式 CPU 在外层循环中进入下一次循环，即，对输出层进行计算，在对输出层的计算中例如包括 sigmoid 函数计算，因此，嵌入式 CPU 基于运行的程序配置执行 sigmoid 函数计算的算子单元，并对神经元（6）进行计算。在对神经元（6）的计算中，嵌入式 CPU 基于运行的程序调用算子单元，并将神经元（3）、（4）和（5）各自的值和神经元（6）的参数输入该算子单元，从而获得神经元（6）的值，该值即为该神经网络的预测值。

[58]从上述对图 3 所示的神经网络的计算实例中可以得出，在对每个神经元的计算中，通过调用相应的算子单元进行具体的计算，从而得到该神经元处的值。其中，当该神经元位于神经网络的隐藏层时，通过计算获取的该神经元处的值为中间结果，将该中间结果存储在芯片中或芯片所在的 PCIE 卡中的 ddr 内存中，以备后续的计算使用。当该神经元位于神经网络的输出层时，通过计算获取的该神经元处的值即为该神经网络的预测值，则通过嵌入式 CPU 配置 DMA 描述符，以控制 DMA 模块将该预测值写入服务器的内存中。

[59]根据本说明书实施例的该数据处理方法尤其适用于神经网络的预测计算。对于芯片中神经网络的计算，目前通用的做法是在芯片上把各个运算单元如卷积的控制设计成格式化的二进制指令，然后把神经网络的各层调用映射到这些指令上、并集成为一个序列，再下放到芯片上，芯片上的逻辑解析每一条指令并启动相应的运算单元。然而，该方式适用于较大算子的运算单元，对于较细算子的控制，用这种指令序列的方式效率不高、且指令序列的量会非常多；对于控制类型的操作，指令序列的方式几乎难以完成（比如用指令序列做多个循环的嵌套就非常困难）；对于输入数据长度适配的问题，比如每次 CNN 网络的输入图像大小不一样或者每次 LSTM 网络输入的语音长度不一致，这会让指令序列无所适从；如果每次输入数据长度不一样都重新下载指令序列，这个过程引起的延时对业务来说太长了。

[60]而在本说明书实施例的基于芯片的数据处理方案中，因为引入了嵌入式 CPU，天然在芯片内部就可以完成较细算子的控制；C 语言函数调用的方式又大大节省了程序指令的空间，并且，嵌入式 CPU 上 C 程序天然就可以完美地进行控制类型的操作，比如各种 if/for/while 等嵌套；嵌入式 CPU 上的 C 程序可以对输入数据的长度做判断，并灵活产生计算过程中下发给各个算子单元的参数；另外，当所述神经网络结构升级时，可通

过服务器向芯片传入新的程序，以进行更新的神经网络的预测计算。

[61]在一个实施例中，所述嵌入式 CPU 还配置为，通过 C 语言中的函数调用，调用所述多个算子单元。

[62]在一个实施例中，所述芯片还包括 DMA 模块，所述嵌入式 CPU 还配置为，在获取
5 所述处理结果之后，通过控制所述 DMA 模块，以将所述处理结果写入所述服务器的内存中。

[63]在一个实施例中，所述嵌入式 CPU 还配置为，异步地调用所述多个算子单元。通过异步地调用多个算子单元，从而可以尽可能地让多个算子单元并行地工作，从而提高硬件的利用率。

10 [64]在一个实施例中，所述嵌入式 CPU 还配置为，对待处理数据或神经网络各层的输入数据的长度进行判断，并基于所述长度获取发送给各个算子单元的参数。

[65]在一个实施例中，所述芯片用于执行多种神经网络的预测计算，其中，获取待处理数据包括，获取待处理数据及其对应的神经网络的类型，其中，基于所述待处理数据，执行所述神经网络中的每一层的计算，以获取处理结果包括，基于所述待处理数据及其
15 对应的神经网络的类型，执行该类型的神经网络中的每一层的计算，以获取处理结果。所述多种神经网络例如包括 cnn、dnn、rnn 等，所述芯片例如通过获取指示神经网络类型的控制信息（如神经网络标识等），从而获取神经网络的类型。芯片在获取神经网络类型之后，会在嵌入式 CPU 中基于神经网络的类型选定与该类型对应的程序并运行该程序，从而执行该类型的神经网络的计算。

20 [66]图 4 示出根据本说明书实施例的一种芯片 400，用于执行计算任务，所述芯片包括：

[67]多个算子单元 41，以及

[68]嵌入式 CPU42，配置为，通过执行以下步骤而进行所述计算任务的计算：配置所述多个算子单元，以使得所述多个算子单元与该计算任务中包括的不同类型的计算分别对应，以及，基于从所述芯片的外部输入的待处理数据，通过调用所述多个算子单元，进
25 行所述计算任务的计算，以获取处理的结果。

[69]在一个实施例中，所述计算任务为神经网络的预测计算，所述嵌入式 CPU42 配置为，基于从所述芯片的外部输入的待处理数据，执行所述神经网络中的每一层的计算，以获取处理结果，其中，通过由所述嵌入式 CPU 执行以下步骤而进行所述神经网络的每一层的计算：配置所述多个算子单元，以使得所述多个算子单元与该层中包括的不同类型

的计算分别对应, 以及, 基于神经网络上一层中各个神经元的值, 通过调用所述多个算子单元, 执行该层中每个神经元处的计算, 以获取该层中每个神经元的值。

[70]在一个实施例中, 所述芯片与服务器连接。

[71]在一个实施例中, 在所述芯片中, 所述待处理数据通过所述服务器传送至所述芯片。

- 5 [72]在一个实施例中, 在所述芯片中, 所述嵌入式 CPU 通过运行基于程序的汇编指令而执行所述步骤, 其中, 所述程序由所述服务器传送至所述芯片。

[73]在一个实施例中, 在所述芯片中, 所述程序为 C 语言程序。

[74]在一个实施例中, 在所述芯片中, 所述嵌入式 CPU 还配置为, 通过 C 语言中的函数调用, 调用所述多个算子单元。

- 10 [75]在一个实施例中, 在所述芯片中, 所述芯片还包括 DMA 模块 43, 所述嵌入式 CPU 还配置为, 在获取所述处理结果之后, 通过控制所述 DMA 模块, 以将所述处理结果写入所述服务器的内存中。

[76]在一个实施例中, 在所述芯片中, 所述嵌入式 CPU 还配置为, 异步地调用所述多个算子单元。

- 15 [77]在一个实施例中, 在所述芯片中, 所述嵌入式 CPU 还配置为, 对所述待处理数据的长度进行判断, 并基于所述长度获取发送给各个算子单元的参数。

[78]在一个实施例中, 在所述芯片中, 所述芯片为 ASIC 芯片或 FPGA 芯片。

[79]在一个实施例中, 所述芯片用于执行多种神经网络的预测计算, 其中, 所述嵌入式 CPU 还配置为, 基于从所述芯片的外部输入的待处理数据及其对应的神经网络类型,

- 20 执行所述类型的神经网络中的每一层的计算, 以获取处理结果。

[80]本说明书另一方面提供一种计算设备, 包括存储器和处理器, 其特征在于, 所述存储器中存储有可执行代码, 所述处理器执行所述可执行代码时, 实现图 2 所示的数据处理方法。

- [81]通过根据本说明书实施例的基于芯片的数据处理方案, 因为引入了嵌入式 CPU, 天然在芯片内部就可以完成较细算子的控制; C 语言函数调用的方式又大大节省了程序指令的空间, 并且, 嵌入式 CPU 上 C 程序天然就可以完美地进行控制类型的操作, 比如各种 if/for/while 等嵌套; 嵌入式 CPU 上的 C 程序可以对输入数据的长度做判断, 并灵活产生计算过程中下发给各个算子单元的参数; 另外, 当所述神经网络结构升级时, 可
- 25

通过服务器向芯片传入新的程序，以进行更新的神经网络的预测计算。

[82]本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于系统实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[83]上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者可能是有利的。

[84]本领域普通技术人员应该还可以进一步意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、计算机软件或者二者的结合来实现，为了清楚地说明硬件和软件的可互换性，在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执轨道，取决于技术方案的特定应用和设计约束条件。本领域普通技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

[85]结合本文中所公开的实施例描述的方法或算法的步骤可以用硬件、处理器执轨道的软件模块，或者二者的结合来实施。软件模块可以置于随机存储器（RAM）、内存、只读存储器（ROM）、电可编程 ROM、电可擦除可编程 ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[86]以上所述的具体实施方式，对本发明的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本发明的具体实施方式而已，并不用于限定本发明的保护范围，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。