



(21) 申请号 202410776929.7

(22) 申请日 2024.06.17

(65) 同一申请的已公布的文献号

申请公布号 CN 118627552 A

(43) 申请公布日 2024.09.10

(73) 专利权人 北京海云捷迅科技股份有限公司

地址 100000 北京市海淀区知春路106号太
平洋国际大厦11层1101室

(72) 发明人 万毅 陈俊帆 高培 田亮

(74) 专利代理机构 北京同进知识产权代理事务

所(普通合伙) 16144

专利代理师 刘冀 郭新禹

(51) Int. Cl.

G06N 3/0464 (2023.01)

G06N 3/063 (2023.01)

(56) 对比文件

CN 106875012 A, 2017.06.20

CN 116360852 A, 2023.06.30

审查员 洪汇隆

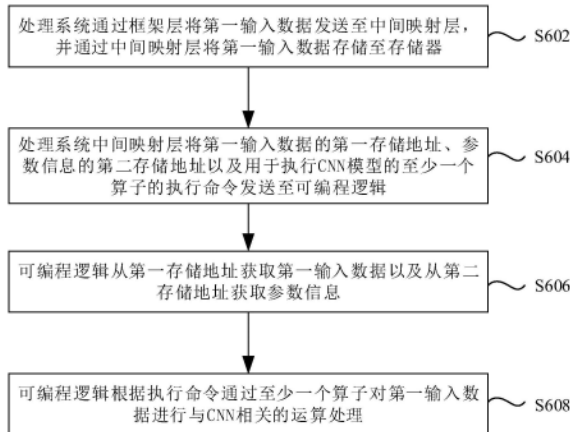
权利要求书3页 说明书11页 附图4页

(54) 发明名称

CNN加速框架以及CNN加速方法

(57) 摘要

本申请公开了一种CNN加速框架以及CNN加速方法。其中,CNN加速方法,应用于CNN加速框架,包括:处理系统通过框架层将第一输入数据发送至中间映射层,并通过中间映射层将第一输入数据存储至存储器;处理系统中间映射层将第一输入数据的第一存储地址、参数信息的第二存储地址以及用于执行CNN模型的至少一个算子的执行命令发送至可编程逻辑,其中参数信息与至少一个算子对应;可编程逻辑从第一存储地址获取第一输入数据以及从第二存储地址获取参数信息;以及可编程逻辑根据执行命令通过至少一个算子对第一输入数据进行与CNN相关的运算处理。



1. 一种CNN加速框架,其特征在于,包括:处理系统以及可编程逻辑,其中所述处理系统配置用于执行以下操作:

通过框架层将第一输入数据发送至中间映射层,并通过所述中间映射层将所述第一输入数据存储至存储器;以及

通过所述中间映射层将所述第一输入数据的第一存储地址、参数信息的第二存储地址以及用于执行CNN模型的至少一个算子的执行命令发送至所述可编程逻辑,其中所述参数信息与所述至少一个算子对应,并且

所述可编程逻辑配置用于执行以下操作:

从所述第一存储地址获取所述第一输入数据以及从所述第二存储地址获取所述参数信息;以及

根据所述执行命令通过所述至少一个算子对所述第一输入数据进行与CNN相关的运算处理,并且

所述可编程逻辑还配置用于执行以下操作:

将所述至少一个算子执行后生成的输出数据存储至所述存储器,并且

所述处理系统还配置用于执行以下操作:

通过中间映射层从所述存储器中获取所述输出数据,并将所述输出数据发送至所述框架层;

在所述输出数据为算子链表中的最后一个算子生成的数据的情况下,通过所述框架层将所述输出数据发送至应用层;以及

在所述输出数据为所述算子链表中的非最后一个算子生成的数据的情况下,将所述输出数据作为所述算子链表的下一个算子的输入数据,其中

将所述输出数据发送至所述框架层的操作,包括:通过所述中间映射层调用所述可编程逻辑对所述输出数据重排为与由所述框架层执行的算子匹配的数据格式,并将重排后的所述输出数据发送至所述框架层。

2. 根据权利要求1所述的CNN加速框架,其特征在于,根据所述执行命令通过所述至少一个算子对所述第一输入数据进行与CNN相关的运算处理的操作,包括:

根据所述执行命令将所述至少一个算子的参数信息和所述第一输入数据输入至预设的CNN加速器;以及

利用所述CNN加速器通过所述至少一个算子中的各个算子依次对所述第一输入数据进行与CNN相关的运算处理,其中由所述至少一个算子作为子图算子组成子图。

3. 根据权利要求2所述的CNN加速框架,其特征在于,通过所述中间映射层将所述第一输入数据存储至存储器的操作,包括:

通过所述中间映射层调用所述可编程逻辑对所述第一输入数据重排为与所述子图的第一个算子匹配的数据格式,并将重排后的所述第一输入数据存储至所述存储器。

4. 根据权利要求1所述的CNN加速框架,其特征在于,还包括:

通过所述中间映射层接收应用层发送的第二输入数据;

调用所述可编程逻辑对所述第二输入数据进行预处理,生成预处理数据;以及

通过所述中间映射层将所述预处理数据返回所述应用层作为所述第一输入数据。

5. 根据权利要求1所述的CNN加速框架,其特征在于,所述处理系统还配置用于执行以

下操作：

通过应用层载入CNN模型,并将所述CNN模型发送至所述框架层;
通过所述框架层分析所述CNN模型,确定与所述CNN模型对应的全部算子;
通过所述框架层根据所述全部算子的执行顺序建立算子链表;以及
通过所述中间映射层将由所述可编程逻辑执行的至少一个算子的参数信息存储至存储器。

6.根据权利要求5所述的CNN加速框架,其特征在于,通过所述中间映射层将所述可编程逻辑执行的至少一个算子的参数信息存储至所述存储器的操作,包括:

通过所述框架层从所述全部算子中确定由所述可编程逻辑执行的所述至少一个算子;
通过所述框架层对所述至少一个算子的参数信息重排为与所述至少一个算子匹配的数据格式,并将重排后的所述参数信息发送至所述中间映射层;以及
通过所述中间映射层将重排后的所述参数信息存储至所述存储器。

7.根据权利要求1所述的CNN加速框架,其特征在于,通过框架层将第一输入数据发送至中间映射层的操作,包括:利用API接口通过所述框架层将所述第一输入数据发送至所述中间映射层。

8.根据权利要求1所述的CNN加速框架,其特征在于,通过所述中间映射层将所述第一输入数据的第一存储地址、参数信息的第二存储地址以及所述执行命令发送至所述可编程逻辑的操作,包括:利用跨硬件交互总线通过所述中间映射层将所述第一输入数据的第一存储地址、所述参数信息的第二存储地址以及所述执行命令发送至所述可编程逻辑。

9.一种CNN加速方法,应用于CNN加速框架,其特征在于,包括:

处理系统通过框架层将第一输入数据发送至中间映射层,并通过所述中间映射层将所述第一输入数据存储至存储器;

所述处理系统所述中间映射层将所述第一输入数据的第一存储地址、参数信息的第二存储地址以及用于执行CNN模型的至少一个算子的执行命令发送至可编程逻辑,其中所述参数信息与所述至少一个算子对应;

可编程逻辑从所述第一存储地址获取所述第一输入数据以及从所述第二存储地址获取所述参数信息;以及

所述可编程逻辑根据所述执行命令通过所述至少一个算子对所述第一输入数据进行与CNN相关的运算处理,并且

所述方法还包括:

所述可编程逻辑将所述至少一个算子执行后生成的输出数据存储至所述存储器,并且

所述处理系统通过中间映射层从所述存储器中获取所述输出数据,并将所述输出数据发送至所述框架层;

所述处理系统在所述输出数据为算子链表中的最后一个算子生成的数据的情况下,通过所述框架层将所述输出数据发送至应用层;以及

所述处理系统在所述输出数据为所述算子链表中的非最后一个算子生成的数据的情况下,将所述输出数据作为所述算子链表的下一个算子的输入数据,其中

将所述输出数据发送至所述框架层的操作,包括:通过所述中间映射层调用所述可编程逻辑对所述输出数据重排为与由所述框架层执行的算子匹配的数据格式,并将重排后的

所述输出数据发送至所述框架层。

CNN加速框架以及CNN加速方法

技术领域

[0001] 本申请涉及计算机技术领域,特别是涉及一种CNN加速框架以及CNN加速方法。

背景技术

[0002] 近年来,随着物联网(IoT)和移动计算的快速发展,边缘计算作为一种新兴技术受到广泛关注。特别是在视觉识别等领域,卷积神经网络(CNN)因其卓越的表现成为主流选择之一。然而,高性能计算需求与有限的硬件资源和严格的能量消耗约束形成了鲜明对比。在此背景下,可编程逻辑以其独特的可重配置特性和出色的能源效率,逐渐被视为加速边缘端CNN推理的理想平台。

[0003] 其中可编程逻辑加速CNN具有以下优势:

[0004] (1) 定制化性能:根据特定CNN模型的特点,可编程逻辑可通过重新配置逻辑块来针对性地优化计算流水线,大幅提升了处理效率。

[0005] (2) 低延时和高吞吐率:相较于传统处理器,可编程逻辑能在维持高速数据处理的同时减小等待时间。

[0006] (3) 节能效益:利用可编程逻辑的可关断特性,仅激活必要的计算单元,大大降低了整体能耗。

[0007] 基于可编程逻辑的优点,现有技术将处理系统结合可编程逻辑,实现CNN加速。其中可编程逻辑可快速推理CNN模型,与CNN模型对应的机器学习框架部署于处理系统。

[0008] 但是不同可编程逻辑的资源差异较大,因此处理系统不能轻易切换不同的可编程逻辑,处理系统和可编程逻辑具有高耦合性。

[0009] 针对上述的现有技术中存在的处理系统和可编程逻辑具有高耦合性的技术问题,目前尚未提出有效的解决方案。

发明内容

[0010] 本申请的实施例提供了一种CNN加速框架以及CNN加速方法,以至少解决现有技术中存在的处理系统和可编程逻辑具有高耦合性的技术问题。

[0011] 根据本申请实施例的一个方面,提供了一种CNN加速框架,包括:处理系统以及可编程逻辑,其中处理系统配置用于执行以下操作:通过框架层将第一输入数据发送至中间映射层,并通过中间映射层将第一输入数据存储至存储器;以及通过中间映射层将第一输入数据的第一存储地址、参数信息的第二存储地址以及用于执行CNN模型的至少一个算子的执行命令发送至可编程逻辑,其中参数信息与至少一个算子对应,并且可编程逻辑配置用于执行以下操作:从第一存储地址获取第一输入数据以及从第二存储地址获取参数信息;以及根据执行命令通过至少一个算子对第一输入数据进行与CNN相关的运算处理。

[0012] 根据本申请实施例的另一方面,还提供了一种CNN加速方法,应用于CNN加速框架,包括:处理系统通过框架层将第一输入数据发送至中间映射层,并通过中间映射层将第一输入数据存储至存储器;处理系统中间映射层将第一输入数据的第一存储地址、参数信息

的第二存储地址以及用于执行CNN模型的至少一个算子的执行命令发送至可编程逻辑,其中参数信息与至少一个算子对应;可编程逻辑从第一存储地址获取第一输入数据以及从第二存储地址获取参数信息;以及可编程逻辑根据执行命令通过至少一个算子对第一输入数据进行与CNN相关的运算处理。

[0013] 在本申请实施例中,CNN加速框架利用处理系统中设置的中间映射层,与可编程逻辑进行交互,利用存储器向可编程逻辑发送CNN模型的算子的参数信息和输入数据,之后可编程逻辑根据参数信息和输入数据实现CNN加速。从而使得本技术方案中的处理器系统与可编程逻辑在解耦状态实现CNN加速,与现有技术相比,降低了处理系统与可编程逻辑之间的依赖,从而使得处理系统可利用中间映射层可以连接不同的可编程逻辑。进而解决了现有技术中存在的处理系统和可编程逻辑具有高耦合性的技术问题。

附图说明

[0014] 此处所说明的附图用来提供对本申请的进一步理解,构成本申请的一部分,本申请的示意性实施例及其说明用于解释本申请,并不构成对本申请的不当限定。在附图中:

[0015] 图1是根据本申请实施例所述的CNN加速框架的示意图;

[0016] 图2是根据本申请实施例所述的可编程逻辑的示意图;

[0017] 图3是根据本申请实施例所述的算子链表的示意图;

[0018] 图4是根据本申请实施例所述的执行算子链表的一个示意图;

[0019] 图5是根据本申请实施例所述的执行算子链表的另一个示意图;

[0020] 图6是根据本申请实施例的第二个方面所述的CNN加速方法的流程示意图;以及

[0021] 图7是根据本申请实施例所述的CNN加速框架进行CNN加速的顺序流程示意图。

具体实施方式

[0022] 为了使本技术领域的人员更好地理解本申请的技术方案,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述。显然,所描述的实施例仅仅是本申请一部分的实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都应当属于本申请保护的范围。

[0023] 需要说明的是,本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0024] 图1示例性的示出了本申请实施例所述的CNN加速框架的结构示意图。

[0025] 其中CNN加速框架包括处理系统以及可编程逻辑。

[0026] 其中处理系统包括应用层、框架层以及中间映射层。

[0027] 应用层用于与用户交互,并且将需要CNN模型处理的输入数据发送给框架层,并且

接收框架层最终返回的输出数据。其中输入数据可以为图像数据。

[0028] 框架层用于依据算子链表中的算子执行顺序,对输入数据进行处理。框架层通过算子链表中由框架层执行的算子对输入数据进行处理,对于算子链表中的由可编程逻辑执行的算子,则向中间映射层发送由可编程逻辑的算子处理的输入数据。并且框架层还用于接收中间映射层返回的输出数据。

[0029] 中间映射层用于将输入数据存储至存储器中。之后中间映射层将输入数据的存储地址、参数信息的存储地址以及执行命令发送给可编程逻辑。其中参数信息为构建算子链表时由中间映射层存储至存储器的。

[0030] 参考图2所示,可编程逻辑包括数据重排模块、预处理模块以及CNN加速器。

[0031] 数据重排模块用于对输入数据以及输出数据进行数据重排。

[0032] 预处理模块用于对待处理的图像数据(即,输入数据)进行预处理,其中预处理包括图像解码、图像缩放和归一化等操作。

[0033] 可编程逻辑根据中间映射层发送的输入数据的存储地址和参数信息的存储地址,从存储器中获取输入数据和参数信息,通过CNN加速器根据存储地址和参数信息执行CNN加速器中的算子。

[0034] 图3示出了算子链表的示意图,参考图3所示,其中算子链表中的算子包括子图算子和单算子,框架层执行算子链表中的单算子。可编程逻辑执行算子链表中的子图算子。例如,算子链表包括算子1~算子5。其中算子1和算子5为单算子,其算子类型与框架层匹配。算子2~算子4为子图算子,组成子图1。其中算子2~算子4的算子类型与可编程逻辑匹配,并且相邻,因此可作为子图算子组成一个子图,由可编程逻辑执行。从而依据算子链表中各个算子的执行顺序,框架层和可编程逻辑依次执行算子链表中的算子。

[0035] 需要说明的是,算子链表中可以有多个子图,子图中可包括多个子图算子且数量不限,本实施例仅是示例性示出了算子链表的结构,以说明算子执行的过程,此处不再赘述。

[0036] 参考图1所示,提出了一种CNN加速框架,包括:处理系统以及可编程逻辑,其中处理系统配置用于执行以下操作:通过框架层将第一输入数据发送至中间映射层,并通过中间映射层将第一输入数据存储至存储器;以及通过中间映射层将第一输入数据的第一存储地址、参数信息的第二存储地址以及用于执行CNN模型的至少一个算子的执行命令发送至可编程逻辑,其中参数信息与至少一个算子对应,并且可编程逻辑配置用于执行以下操作:从第一存储地址获取第一输入数据以及从第二存储地址获取参数信息;以及根据执行命令通过至少一个算子对第一输入数据进行与CNN相关的运算处理。

[0037] 具体地,用户通过终端设备上设置的应用程序向处理系统的应用层发送图像数据,应用层接收到图像数据后将图像数据发送至框架层。

[0038] 进一步地,框架层将接收到的图像数据作为输入数据11。之后,框架层获取预先建立的算子链表,并根据预先为算子添加的标记,判断算子链表中的第一个算子的算子类型。其中算子类型包括单算子和子图算子。并且其中算子链表中的算子为CNN模型的计算模块。其中算子的标记为解析CNN模型时,为各个算子添加的,用于标识算子的算子类型。

[0039] 参考图3以及图4所示,算子1为算子链表中的第一个算子,框架层判断算子1的算子类型为单算子,因此将输入数据11输入至算子1,通过算子1对输入数据11进行与CNN相关

的运算处理,生成输出数据12,并将输出数据12存储于本地。其中与CNN相关的运算处理包括卷积操作、池化操作或者逐元素操作。

[0040] 进一步地,框架层对算子1执行完毕后,判断算子链表中的算子2的算子类型是单算子还是子图算子。则框架层判定算子2为子图算子,则将输出数据12作为输入数据21(即,第一输入数据),之后将输入数据21(即,第一输入数据)发送至中间映射层。

[0041] 进一步地,中间映射层接收到输入数据21(即,第一输入数据)之后,将输入数据21(即,第一输入数据)发送至存储器中进行存储。

[0042] 并且中间映射层访问算子链表,确定算子2所属的子图1,并且确定子图1中所有的子图算子。其中子图1中所有的子图算子为算子2~算子4(即,至少一个算子)。

[0043] 之后中间映射层将输入数据21(即,第一输入数据)的存储地址(即,第一存储地址),与算子2~算子4对应的参数信息的存储地址(即,第二存储地址)以及用于执行算子2~算子4的执行命令发送给可编程逻辑。其中子图算子的参数信息已预先存储在存储器中。

[0044] 进一步地,可编程逻辑接收到第一存储地址、第二存储地址以及执行命令,则根据第一存储地址从存储器中获取输入数据21(即,第一输入数据),根据第二存储地址从存储器中获取与算子2~算子4对应的参数信息。可编程逻辑通过算子2~算子4对输入数据21(即,第一输入数据)进行与CNN相关的运算处理,生成输出数据22。

[0045] 正如背景技术中所述的,基于可编程逻辑的优点,现有技术将处理系统结合可编程逻辑,实现CNN加速。其中可编程逻辑可快速推理CNN模型,与CNN模型对应的机器学习框架部署于处理系统。但是不同可编程逻辑的资源差异较大,因此处理系统不能轻易切换不同的可编程逻辑,处理系统和可编程逻辑具有高耦合性。

[0046] 针对以上所述的技术问题,通过本申请实施例的技术方案,CNN加速框架利用处理系统中设置的中间映射层,与可编程逻辑进行交互,利用存储器向可编程逻辑发送CNN模型的算子的参数信息和输入数据,之后可编程逻辑根据参数信息和输入数据实现CNN加速。从而使得本技术方案中的处理器系统与可编程逻辑在解耦状态实现CNN加速,与现有技术相比,降低了处理系统与可编程逻辑之间的依赖,从而使得处理系统可利用中间映射层连接不同的可编程逻辑。进而解决了现有技术中存在的处理系统和可编程逻辑具有高耦合性的技术问题。

[0047] 可选地,根据执行命令通过至少一个算子对第一输入数据进行与CNN相关的运算处理的操作,包括:根据执行命令将至少一个算子的参数信息和第一输入数据输入至预设的CNN加速器;以及利用CNN加速器通过至少一个算子中的各个算子依次对第一输入数据进行与CNN相关的运算处理,其中由至少一个算子作为子图算子组成子图。

[0048] 具体地,可编程逻辑将从存储器中获取到的算子2~算子4(即至少一个算子)的参数信息和输入数据21(即,第一输入数据)输入至预设的CNN加速器。其中CNN加速器中设置有子图1,其中子图1包括算子2~算子4。并且在本实施例中,子图中的子图算子可以连续对数据进行处理,如果可编程逻辑中设置有单算子,则需要将每个单算子生成的输出数据返回至处理器系统。

[0049] 进一步地,参考图5所示,可编程逻辑通过CNN加速器的算子2利用参数信息对输入数据21进行处理,生成输出数据211。其中当算子为卷积算子,其参数信息至少包括该卷积层中的权重参数。从而当算子2为卷积算子时,则根据参数信息对输入数据21进行卷积操

作,生成输出数据211。

[0050] 进一步地,可编程逻辑将输出数据211发送至存储器中进行存储,并将输出数据211作为输入数据212输入CNN加速器的算子3。可编程逻辑通过CNN加速器的算子3根据参数信息对输入数据212进行处理,生成输出数据213。

[0051] 进一步地,可编程逻辑将输出数据213发送至存储器中进行存储,并将输出数据213作为输入数据214。之后可编程逻辑通过CNN加速器的算子4根据参数信息对输入数据214进行处理,生成输出数据22。之后可编程逻辑将输出数据22存储至存储器中。

[0052] 从而本技术方案通过将算子类型相同且连续的算子组成子图,从而可编程逻辑的子图中的子图算子可以依次对输入数据进行处理,无需将各个算子生成的输出数据返回至处理系统,减少了数据传输的时间。并且本技术方案依据可编程逻辑数据处理速度快的特点,实现CNN加速处理。

[0053] 可选地,通过中间映射层将第一输入数据存储至存储器的操作,包括:通过中间映射层调用可编程逻辑对第一输入数据重排为与子图的第一个算子匹配的数据格式,并将重排后的第一输入数据存储至存储器。

[0054] 具体地,中间映射层接收到框架层发送的输入数据21,之后将输入数据21发送存储器,之后将输入数据21的存储地址发送至可编程逻辑。之后中间映射层生成用于对输入数据21进行重排的数据重排命令,并将该数据重排命令发送至可编程逻辑。

[0055] 进一步地,可编程逻辑接收到存储地址和数据重排命令后,根据存储地址从存储器中获取输入数据21,并通过数据重排模块对输入数据21进行重排,从而将输入数据21重排为与子图1中的第一个算子,即算子2,匹配的数据格式。之后,可编程逻辑将重排后的输入数据21发送至存储器进行存储。

[0056] 进一步地,中间映射层自动从存储器中获取重排后的输入数据21,并根据该输入数据21生成用于执行子图1的执行命令。之后中间映射层将重排后的输入数据21存储至存储器,并将重排后的输入数据21的存储地址(即,第一存储地址)以及执行命令发送给可编程逻辑。

[0057] 进一步地,可编程逻辑接收到第一存储地址以及执行命令之后,从存储器中获取重排后的输入数据21(即,第一输入数据),通过CNN加速器对重排后的输入数据21进行处理,生成输出数据22。

[0058] 从而,在本技术方案中可编程逻辑在对输入数据进行CNN加速处理之前,会对输入数据进行数据重排,从而得到与子图中的第一个算子的数据格式匹配的输入数据,从而使得每个输入数据都适用于可编程逻辑,提高了可编程逻辑的对输入数据进行处理适应性。并且本技术方案对输入数据进行数据重排,将输入数据生成可适用于不同可编程逻辑的输入数据,进而提高了用于部署CNN模型的机器学习框架在可编程逻辑的可移植性。

[0059] 可选地,CNN加速框架,还包括:通过中间映射层接收应用层发送的第二输入数据;调用可编程逻辑对第二输入数据进行预处理,生成预处理数据;以及通过中间映射层将预处理数据返回应用层作为第一输入数据。

[0060] 具体地,应用层接收用户通过应用程序发送的图像数据(即,第二输入数据),之后应用层将图像数据(即,第二输入数据)发送至中间映射层。

[0061] 进一步地,中间映射层接收到图像数据(即,第二输入数据)后,将图像数据(即,第

二输入数据)存储至存储器中,并根据图像数据(即,第二输入数据)生成预处理命令。其中预处理命令包括CNN模型需要的输入尺寸和数据格式等信息。之后中间映射层将图像数据(即,第二输入数据)存储地址以及预处理命令发送至可编程逻辑。

[0062] 进一步地,可编程逻辑接收到预处理命令后,对预处理命令进行解析,从而确定CNN模型需要的尺寸和数据格式等信息。之后可编程逻辑将接收到的图像数据(即,第二输入数据)输入至预处理模块。之后可编程逻辑通过预处理模块根据预处理命令将图像数据(即,第二输入数据)进行预处理,生成预处理数据。其中预处理操作至少包括:(1)将图像数据(即,第二输入数据)缩放至符合CNN模型的输入尺寸;(2)对图像数据(即,第二输入数据)进行归一化处理;(3)根据CNN模型的输入数据的数据格式,对图像数据(即,第二输入数据)的数据格式进行转换。

[0063] 进一步地,可编程逻辑将预处理数据发送至存储器进行存储。之后中间映射层从存储器自动读取到预处理数据,并将预处理数据发送至应用层。

[0064] 进一步地,应用层接收到预处理数据后,将该预处理数据作为输入数据11(即,第一输入数据)发送至框架层。

[0065] 从而本技术方案在对第一输入数据进行CNN处理之前,还会通过可编程逻辑生成符合CNN模型的输入尺寸和数据格式的预处理数据(即,第一输入数据),从而使得第一输入数据更加适应CNN模型的处理,并且提高了对第一输入数据的处理速度。

[0066] 可选地,处理系统还配置用于执行以下操作:通过应用层载入CNN模型,并将CNN模型发送至框架层;通过框架层分析CNN模型,确定与CNN模型对应的全部算子;通过框架层根据全部算子的执行顺序建立算子链表;以及通过中间映射层将由可编程逻辑执行的至少一个算子的参数信息存储至存储器。

[0067] 具体地,应用层载入CNN模型,并将CNN模型发送至框架层。

[0068] 进一步地,框架层接收到CNN模型后,对CNN模型进行分析,确定CNN模型头文件。其中CNN模型头文件包括:CNN模型需要的输入尺寸、通道数和数据格式等。之后框架层将CNN模型头文件发送至应用层。

[0069] 并且框架层对CNN模型进行分析,确定CNN模型的全部算子,并且确定各个算子的权重等参数信息。

[0070] 例如CNN模型的全部算子依照执行顺序包括:算子1、算子2、...、算子5。

[0071] 进一步地,框架层判断该CNN模型是否首次执行,当判定CNN模型为首次执行,则根据CNN模型的全部算子的执行顺序建立算子链表。

[0072] 图3示出了算子链表的示意图。其中算子链表的首节点为算子1,末节点为算子5。并且其中首节点的输入数据指针指向待推理的图像数据,其他节点的输入数据指针指向上一个节点的输出数据处。

[0073] 进一步地,框架层从全部算子中确定子图算子的参数信息,并且将子图算子的参数信息发送至中间映射层,中间映射层将参数信息发送至存储器进行存储。

[0074] 从而,本技术方案在执行CNN模型之前,对CNN模型进行分析,从而预先为应用层确定对输入数据进行预处理所需要输入尺寸以及数据格式,避免在CNN模型执行时临时确定,并且提高了对输入数据的处理速度。

[0075] 可选地,通过中间映射层将可编程逻辑执行的至少一个算子的参数信息存储至存

储器的操作,包括:通过框架层从全部算子中确定由可编程逻辑执行的至少一个算子;通过框架层对至少一个算子的参数信息重排为与至少一个算子匹配的数据格式,并将重排后的参数信息发送至中间映射层;以及通过中间映射层将重排后的参数信息存储至存储器。

[0076] 具体地,CNN模型的全部算子依照执行顺序包括:算子1、算子2、...、算子5。

[0077] 进一步地,框架层确定算子1~算子5的各个算子的算子类型,并根据框架层和可编程逻辑各自支持的算子类型,对算子1~算子5进行分配,从而确定由可编程逻辑执行的算子以及由框架层执行的算子。

[0078] 例如框架层支持的算子类型为池化算子,可编程逻辑支持的算子类型为卷积算子和激活算子。其中算子1和算子5为池化算子,算子2和算子4为卷积算子,算子3为激活算子。

[0079] 从而框架层将算子1和算子5分配给框架层,作为由框架层执行的单算子。并且根据生成子图的原则,将算子2~算子4作为子图算子组成子图1,并将子图1分配给可编程逻辑,作为由可编程逻辑执行的子图。其中生成子图的原则为:符合可编程逻辑支持的算子类型,并且多个子图算子是相邻的。

[0080] 进一步地,框架层在确定算子2~算子4为由可编程逻辑执行的子图算子后,确定算子2~算子4的参数信息,并将算子2~算子4的参数信息进行重排,生成符合可编程逻辑的数据格式的参数信息。

[0081] 进一步地,框架层将重排后的参数信息发送至中间映射层,之后中间映射层将重排后的参数信息发送至存储器进行存储。其中在之后的CNN模型执行过程中,中间映射层向可编程逻辑发送的参数信息的第二存储地址中存储的数据为重排后的参数信息。

[0082] 从而,本技术方案利用CNN模型中的算子建立算子链表,从而预先确定子图以及单算子,从而使得在CNN模型执行时,能够快速确定执行的装置是处理系统还是可编程逻辑,提高了CNN模型的执行速度。并且本技术方案对参数信息进行数据重排,能够使得参数信息适用于可编程逻辑,提高参数信息和可编程逻辑的匹配精度。

[0083] 可选地,CNN加速框架,还包括:可编程逻辑还配置用于执行以下操作:将至少一个算子执行后生成的输出数据存储至存储器,并且处理系统还配置用于执行以下操作:通过中间映射层从存储器中获取输出数据,并将输出数据发送至框架层;在输出数据为算子链表中的最后一个算子生成的数据的情况下,通过框架层将输出数据发送至应用层;以及在输出数据为算子链表中的非最后一个算子生成的数据的情况下,将输出数据作为算子链表的下一个算子的输入数据。

[0084] 具体地,参考图4所示,可编程逻辑根据执行命令对算子2~算子4执行完毕,生成输出数据22之后,将输出数据22发送至存储器进行存储。

[0085] 进一步地,在输出数据22存储至存储器后,中间映射层自动从存储器中获取到输出数据22。之后中间映射层将输出数据22返回至框架层。

[0086] 进一步地,框架层接收到中间映射层发送的输出数据22之后,判断算子4是否为算子链表中的最后一个算子。

[0087] 更具体的,例如框架层判定算子4为算子链表中的非最后一个算子,则将输出数据22作为输入数据31。

[0088] 进一步地,框架层通过算子5对输入数据31进行与CNN相关的运算处理,生成输出数据32,并将输出数据32存储至本地。

[0089] 进一步地,框架层判断算子5是否为算子链表中的最后一个算子。从而框架层判断算子5为算子链表中的最后一个算子,则将输出数据33发送至应用层。

[0090] 进一步地,应用层将输出数据32发送至用户的终端设备上安装的应用程序进行显示。

[0091] 从而本技术方案通过算子链表中算子的执行顺序,利用算子处理图像数据,从而将上一个算子的输出数据作为下一个算子的输入数据,从而使得单算子和子图算子可以连续执行,提高了CNN模型的执行速度。

[0092] 可选地,将输出数据发送至框架层的操作,包括:通过中间映射层调用可编程逻辑对输出数据重排为与由框架层执行的算子匹配的数据格式,并将重排后的输出数据发送至框架层。

[0093] 具体地,可编程逻辑将输出数据22存储至存储器之后,中间映射层自动从存储器中获取输出数据22。

[0094] 之后中间映射层根据输出数据22生成将输出数据22重排为符合框架层的数据格式的数据重排命令,之后将输出数据22发送至存储器进行存储。

[0095] 进一步地,中间映射层将输出数据22的存储地址以及数据重排命令发送至可编程逻辑。

[0096] 可编程逻辑根据数据重排命令,将输出数据22输入至数据重排模块,通过数据重排模块对输出数据22进行重排,生成重排后的输出数据22。其中重排后的输出数据22与框架层的数据格式相同。

[0097] 进一步地,可编程逻辑将重排后的输出数据22发送至存储器进行存储。框架层自动从存储器读取到重排后的输出数据22,并将重排后的输出数据22发送至框架层。

[0098] 进一步地,框架层接收到输出数据22之后,将重排后的输出数据22作为输入数据31。

[0099] 从而本技术方案将可编程逻辑输出的输出数据进行数据重排,从而获得与框架层的数据格式相同的输出数据,从而当下一个算子为框架层执行的单算子时,可以直接将重排后的输出数据作为输入数据,提高了CNN的执行速度。

[0100] 可选地,通过框架层将第一输入数据发送至中间映射层的操作,包括:利用API接口通过框架层将第一输入数据发送至中间映射层。从而本技术方案利用API接口交互效率高的特点,提高了框架层和中间映射层的交互效率。

[0101] 可选地,通过中间映射层将第一输入数据的第一存储地址、参数信息的第二存储地址以及执行命令发送至可编程逻辑的操作,包括:利用跨硬件交互总线通过中间映射层将第一输入数据的第一存储地址、参数信息的第二存储地址以及执行命令发送至可编程逻辑。

[0102] 具体地,当处理系统为ARM架构,跨硬件交互总线可以是AXI总线或者Avalon总线;当处理系统为x86架构,跨硬件交互总线可以是PCI Express总线。

[0103] 从而,本技术方案通过安装不同的总线去匹配处理系统的系统架构,从而使得处理系统和可编程逻辑更加适配。

[0104] 从而根据本实施例的第一个方面,CNN加速框架利用处理系统中设置的中间映射层,与可编程逻辑进行交互,利用存储器向可编程逻辑发送CNN模型的算子的参数信息和输

入数据,之后可编程逻辑根据参数信息和输入数据实现CNN加速。从而使得本技术方案中的处理器系统与可编程逻辑在解耦状态实现CNN加速,与现有技术相比,降低了处理系统与可编程逻辑之间的依赖,从而使得处理系统可利用中间映射层可以连接不同的可编程逻辑。进而解决了现有技术中存在的处理系统和可编程逻辑具有高耦合性的技术问题。

[0105] 在上述运行环境下,根据本实施例的第二个方面,提供了一种CNN加速方法,该方法由图1中所示的CNN加速框架实现。图6示出了该方法的流程示意图,参考图6所示,该方法包括:

[0106] S602:处理系统通过框架层将第一输入数据发送至中间映射层,并通过中间映射层将第一输入数据存储至存储器;

[0107] S604:处理系统中间映射层将第一输入数据的第一存储地址、参数信息的第二存储地址以及用于执行CNN模型的至少一个算子的执行命令发送至可编程逻辑,其中参数信息与至少一个算子对应;

[0108] S606:可编程逻辑从第一存储地址获取第一输入数据以及从第二存储地址获取参数信息;以及

[0109] S608:可编程逻辑根据执行命令通过至少一个算子对第一输入数据进行与CNN相关的运算处理。

[0110] 从而根据本实施例的第二个方面,CNN加速框架利用处理系统中设置的中间映射层,与可编程逻辑进行交互,利用存储器向可编程逻辑发送CNN模型的算子的参数信息和输入数据,之后可编程逻辑根据参数信息和输入数据实现CNN加速。从而使得本技术方案中的处理器系统与可编程逻辑在解耦状态实现CNN加速,与现有技术相比,降低了处理系统与可编程逻辑之间的依赖,从而使得处理系统可利用中间映射层可以连接不同的可编程逻辑。进而解决了现有技术中存在的处理系统和可编程逻辑具有高耦合性的技术问题。

[0111] 综上,参考图7所述,CNN加速框架进行CNN加速的顺序步骤如下:

[0112] 701:处理系统的应用层从磁盘或者网络读取CNN模型,并将CNN模型发送给框架层;

[0113] 702:框架层解析CNN模型,确定CNN模型头文件,并将CNN模型头文件返回给应用层;

[0114] 703:框架层根据CNN模型的算子,建立算子链表,并且将子图算子的参数信息进行重排,将重排后的参数信息发送至中间映射层;

[0115] 704:中间映射层将重排后的参数信息存储至存储器中;

[0116] 705:应用层获取图像数据(即,第二输入数据),将图像数据(即,第二输入数据)发送至中间映射层;

[0117] 706:中间映射层调用可编程逻辑的预处理模块对图像数据进行预处理,生成预处理数据,并将预处理数据返回至应用层;

[0118] 707:应用层将预处理数据作为第一输入数据,并向第一输入数据发送至框架层;

[0119] 708:框架层将第一输入数据发送至中间映射层;

[0120] 709:中间映射层调用可编程逻辑的数据重排模块,对第一输入数据进行重排,并将重排后的第一输入数据存储至存储器中;

[0121] 710:中间映射层将第一输入数据的第一存储地址,参数信息的第二存储地址以及

执行命令发送至可编程逻辑；

[0122] 711:可编程逻辑根据第一存储地址从存储器获取第一输入数据以及根据第二存储地址从存储器中获取参数信息；

[0123] 712:可编程逻辑根据执行命令,将第一输入数据和参数信息输入CNN加速器中,通过CNN加速器的子图算子对第一输入数据进行与CNN相关的运算处理,生成输出数据,之后可编程逻辑将输出数据存储至存储器中；

[0124] 713:中间映射层从存储器中获取到输出数据,调用可编程逻辑的数据重排模块,对输出数据进行数据重排,并将重排后的输出数据发送至框架层；

[0125] 714:在输出数据为算子链表中的最后一个算子生成的输出数据的情况下,通过所述框架层将所述输出数据发送至应用层；

[0126] 715:在所述输出数据为所述算子链表中的非最后一个算子生成的数据的情况下,将所述输出数据作为所述算子链表的下一个算子的输入数据,直到算子链表的算子执行完毕。

[0127] 需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本发明并不受所描述的动作顺序的限制,因为依据本发明,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作和模块并不一定是本发明所必须的。

[0128] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算机,服务器,或者网络设备等)执行本发明各个实施例所述的方法。

[0129] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0130] 在本发明的上述实施例中,对各个实施例的描述都各有侧重,某个实施例中没有详述的部分,可以参见其他实施例的相关描述。

[0131] 在本申请所提供的几个实施例中,应该理解到,所揭露的技术内容,可通过其它的方式实现。其中,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,单元或模块的间接耦合或通信连接,可以是电性或其它的形式。

[0132] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0133] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单

元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0134] 所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,RandomAccess Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0135] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

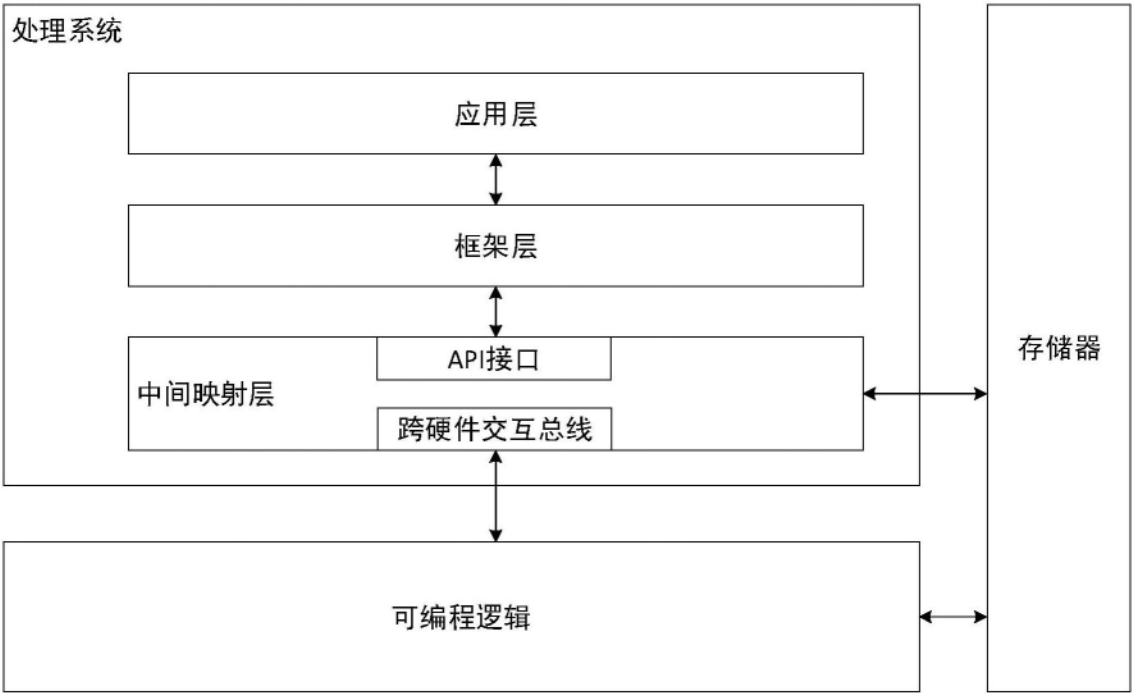


图1

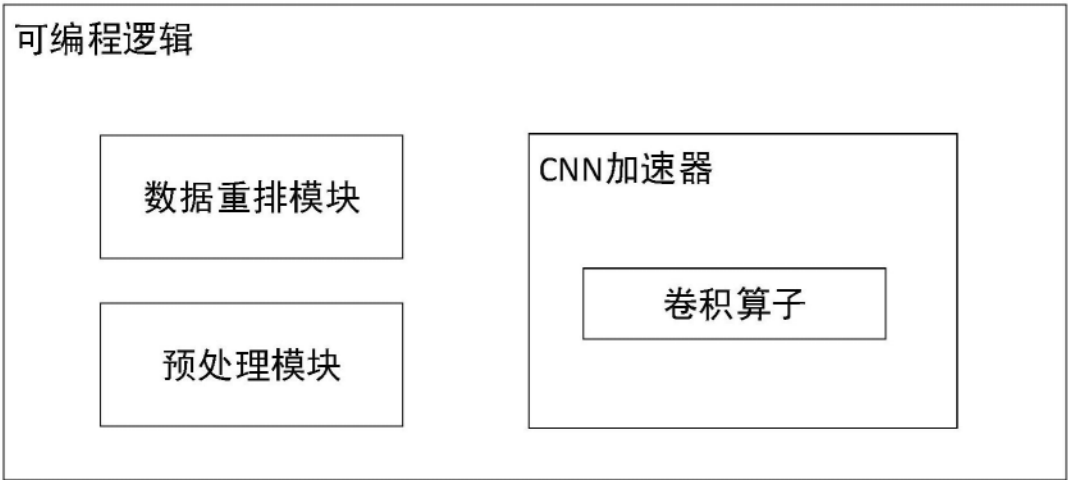


图2

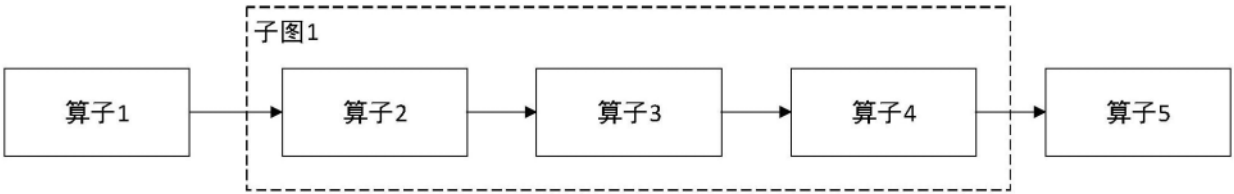


图3

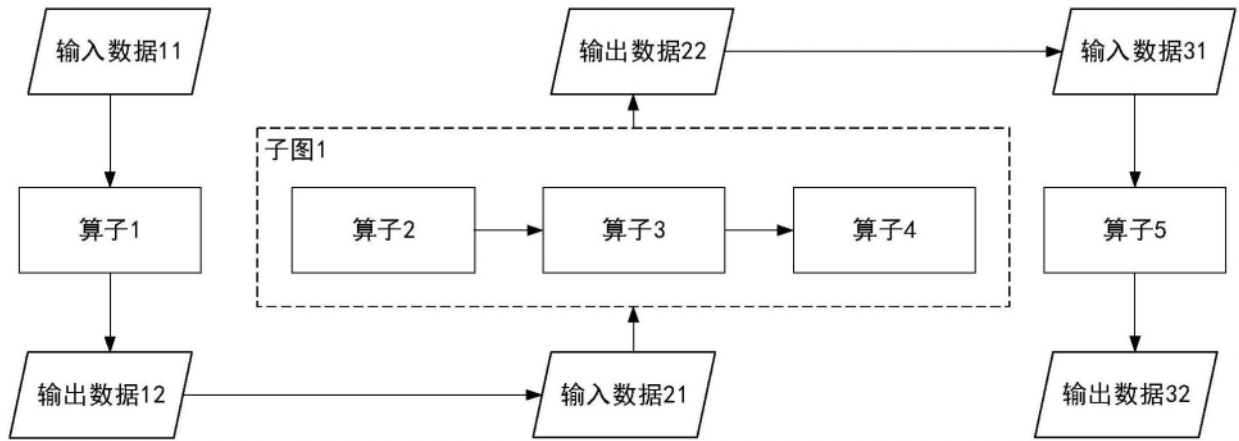


图4

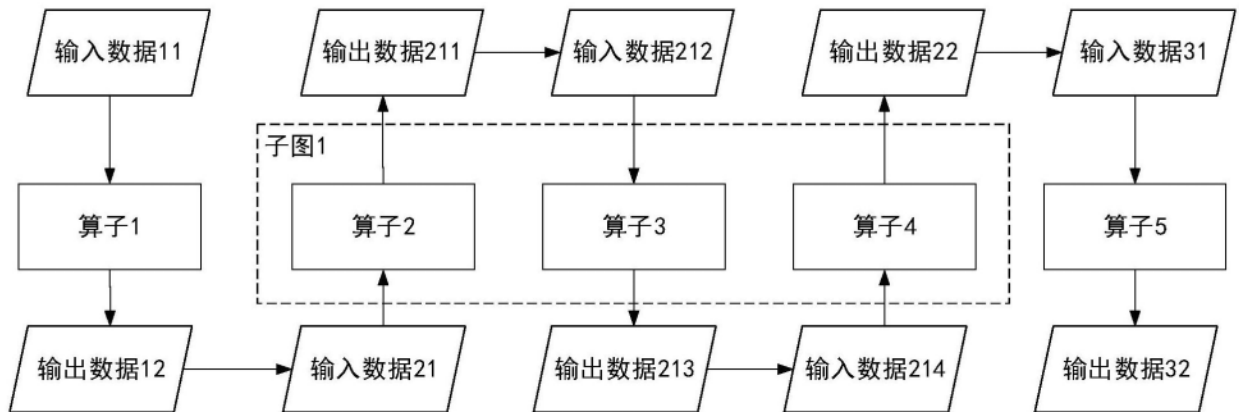


图5

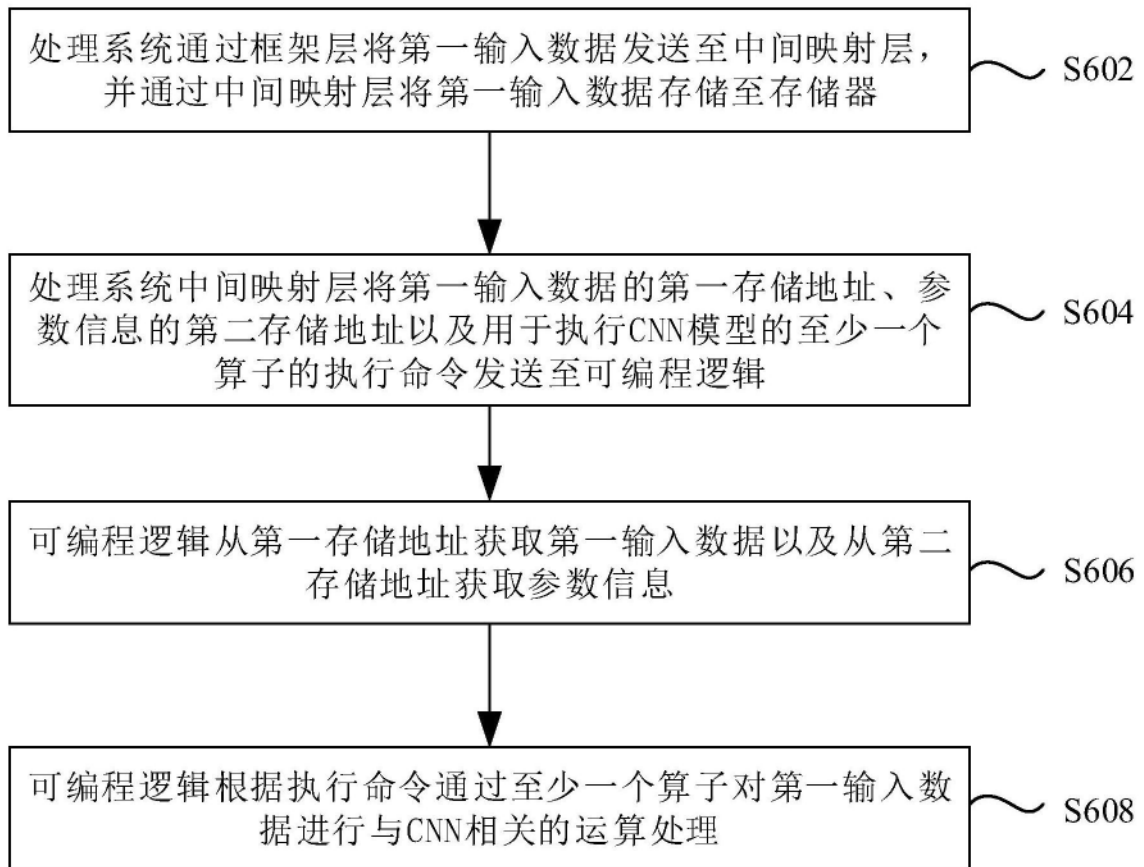


图6



图7