(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116011562 A
(43)申请公布日 2023.04.25

(21)申请号 202211111919.9

(22)申请日 2022.09.13

(71)申请人 上海壁仞智能科技有限公司
地址 201100 上海市闵行区陈行公路2388号16幢13层1302室

(72)发明人 请求不公布姓名

(74)专利代理机构 北京市柳沈律师事务所 11105
专利代理师 彭久云

(51)Int.Cl.
G06N 3/10(2006.01)

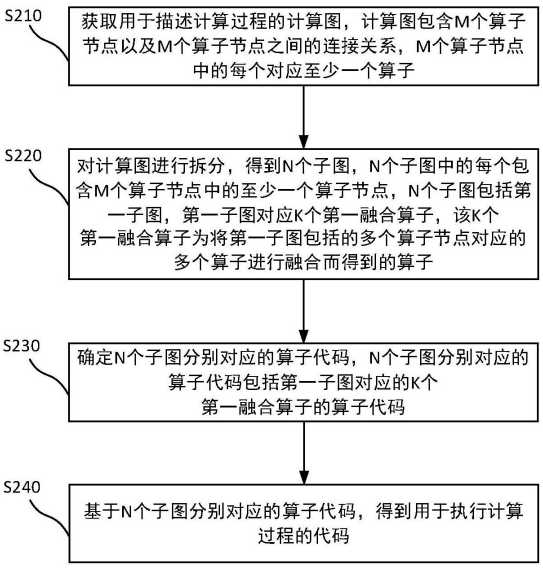
权利要求书3页 说明书17页 附图8页

(54)发明名称

算子处理方法及算子处理装置、电子设备和可读存储介质

(57)摘要

一种算子处理方法、算子处理装置、电子设备和计算机可读存储介质。该算子处理方法包括:获取计算图,该计算图包含M个算子节点以及M个算子节点之间的连接关系,每个算子节点对应至少一个算子;对计算图进行拆分,得到N个子图,每个子图包含至少一个算子节点,N个子图包括第一子图,第一子图对应K个第一融合算子,K个第一融合算子为将第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子;确定N个子图分别对应的算子代码,N个子图分别对应的算子代码包括第一子图对应的K个第一融合算子的算子代码;基于N个子图分别对应的算子代码,得到用于执行计算过程的代码。该方法可以适用于各种场景,适用范围广,泛化能力强。



CN 116011562 A

1.一种算子处理方法,包括:

获取用于描述计算过程的计算图,其中,所述计算图包含M个算子节点以及所述M个算子节点之间的连接关系,所述M个算子节点中的每个对应至少一个算子;

对所述计算图进行拆分,得到N个子图,其中,所述N个子图中的每个包含所述M个算子节点中的至少一个算子节点,所述N个子图包括第一子图,所述第一子图对应K个第一融合算子,所述K个第一融合算子为将所述第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子;

确定所述N个子图分别对应的算子代码,其中,所述N个子图分别对应的算子代码包括所述第一子图对应的所述K个第一融合算子的算子代码;

基于所述N个子图分别对应的算子代码,得到用于执行所述计算过程的代码,

其中,M、N和K均为不小于1的整数。

2.根据权利要求1所述的方法,其中,

所述第一子图包括的多个算子节点为P个算子节点,其中,P为大于1的整数;

确定所述N个子图分别对应的算子代码,包括:

针对所述第一子图,确定所述P个算子节点分别对应的算子的代码;

基于所述P个算子节点分别对应的算子的代码,得到所述第一子图对应的所述K个第一融合算子的算子代码。

3.根据权利要求2所述的方法,其中,针对所述第一子图,确定所述P个算子节点分别对应的算子的代码,包括:

针对所述P个算子节点中的每个算子节点,执行如下操作:

在所述算子节点对应的算子为预定义算子的情况下,针对所述预定义算子,获取与所述预定义算子对应的配置参数和代码模块,并基于所述配置参数和代码模块,得到所述预定义算子的代码;

在所述算子节点对应的算子为自定义算子的情况下,针对所述自定义算子,对所述自定义算子进行编译,以得到所述自定义算子的代码。

4.根据权利要求2所述的方法,其中,基于所述P个算子节点分别对应的算子的代码,得到所述第一子图对应的所述K个第一融合算子的代码,包括:

将所述P个算子节点分别对应的算子的代码进行组合,得到所述K个第一融合算子的算子代码。

5.根据权利要求1所述的方法,其中,对所述计算图进行拆分,得到N个子图,包括:

若Q个算子节点对应的算子的目标属性相同,则将所述Q个算子节点划分为一个子图,其中,所述目标属性包括类型属性、计算属性和数据传输属性中的至少一种,

其中,Q为大于1的整数。

6.根据权利要求5所述的方法,其中,对所述计算图进行拆分,得到N个子图,包括:

若所述Q个算子节点对应的算子均为相同类型的算子,则将所述Q个算子节点划分为一个子图。

7.根据权利要求5所述的方法,其中,对所述计算图进行拆分,得到N个子图,包括:

若所述Q个算子节点对应的算子配置为通过寄存器传输数据,则将所述Q个算子节点划分为一个子图。

8.根据权利要求5所述的方法,其中,对所述计算图进行拆分,得到N个子图,包括:

若所述Q个算子节点对应的算子配置为在同一个计算单元上运行,则将所述Q个算子节点划分为一个子图。

9.根据权利要求1所述的方法,其中,对所述计算图进行拆分,得到N个子图,包括:

若Q个算子节点对应的算子的类型和执行顺序与预定融合算子包含的算子的类型和执行顺序一致,则将所述Q个算子节点划分为一个子图。

10.根据权利要求5至9任一项所述的方法,其中,所述Q个算子节点在根据所述计算图确定的执行顺序上为彼此连续的或彼此并列的。

11.根据权利要求1至9任一项所述的方法,其中,

所述K个第一融合算子中的至少一个包括多个算子,所述多个算子依次连接并且顺次执行;

对于所述多个算子中的相邻两个算子,在执行顺序上的前一个算子的计算结果数据作为后一个算子的输入数据。

12.根据权利要求1至9任一项所述的方法,其中,

所述N个子图还包括第二子图,所述第二子图对应R个第二融合算子,所述R个第二融合算子为将所述第二子图包括的多个算子节点对应的多个算子进行融合而得到的算子,其中,R为不小于1的整数;

所述N个子图分别对应的算子代码还包括所述第二子图对应的所述R个第二融合算子的算子代码;

所述方法还包括:

在确定所述N个子图分别对应的算子代码之前,对所述K个第一融合算子和所述R个第二融合算子进行优化处理,以得到分别针对所述K个第一融合算子和所述R个第二融合算子的配置信息。

13.根据权利要求12所述的方法,其中,对所述K个第一融合算子和所述R个第二融合算子进行优化处理,以得到针对所述K个第一融合算子和所述R个第二融合算子的配置信息,包括:

对所述K个第一融合算子和所述R个第二融合算子进行资源配置优化,以得到针对每个所述第一融合算子和每个所述第二融合算子的优化后的资源配置信息,其中,所述资源包括计算资源和/或存储资源。

14.根据权利要求12所述的方法,其中,对所述K个第一融合算子和所述R个第二融合算子进行优化处理,以得到针对所述K个第一融合算子和所述R个第二融合算子的配置信息,包括:

基于所述计算图的计算过程,针对所述K个第一融合算子和所述R个第二融合算子制定同步策略,以得到针对所述K个第一融合算子和所述R个第二融合算子的同步配置信息。

15.根据权利要求12所述的方法,其中,基于所述计算图的计算过程,针对所述K个第一融合算子和所述R个第二融合算子制定同步策略,以得到针对所述K个第一融合算子和所述R个第二融合算子的同步配置信息,包括:

至少针对所述K个第一融合算子和所述R个第二融合算子中的两个融合算子执行如下操作:

若所述两个融合算子存在数据依赖关系，并且数据接收方的调度单元标识与数据产生方的调度单元标识以及所述数据产生方至所述数据接受方之间的调度单元标识均不相同，则将所述两个融合算子配置为进行数据同步处理，

其中，所述数据产生方为所述两个融合算子中的一者，所述数据产生方为另一者。

16.根据权利要求1至9任一项所述的方法，其中，所述第一融合算子包括所述第一子图包括的多个算子节点对应的多个算子；

所述方法还包括：

针对所述第一融合算子，在所述第一融合算子包含的所述多个算子之间进行优化处理，以得到针对所述多个算子的配置信息，

其中，所述优化处理包括资源配置优化和/或制定同步策略。

17.根据权利要求1至9任一项所述的方法，其中，基于所述N个子图分别对应的算子代码，得到用于执行所述计算过程的代码，包括：

将所述N个子图分别对应的算子代码进行组合，得到用于执行所述计算过程的代码。

18.一种信息处理装置，包括：

获取模块，配置为获取用于描述计算过程的计算图，其中，所述计算图包含M个算子节点以及所述M个算子节点之间的连接关系，所述M个算子节点中的每个对应至少一个算子；

拆分模块，配置为对所述计算图进行拆分，得到N个子图，其中，所述N个子图中的每个包含所述M个算子节点中的至少一个算子节点，所述N个子图包括第一子图，所述第一子图对应K个第一融合算子，所述K个第一融合算子为将所述第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子；

确定模块，配置为确定所述N个子图分别对应的算子代码，其中，所述N个子图分别对应的算子代码包括所述第一子图对应的所述K个第一融合算子的算子代码；

代码模块，配置为基于所述N个子图分别对应的算子代码，得到用于执行所述计算过程的代码，

其中，M、N和K均为不小于1的整数。

19.一种电子设备，包括：

处理器；

存储器，存储有一个或多个计算机程序模块；

其中，所述一个或多个计算机程序模块被配置为由所述处理器执行，用于实现权利要求1-17任一项所述的算子处理方法。

20.一种计算机可读存储介质，存储有非暂时性计算机可读指令，当所述非暂时性计算机可读指令由计算机执行时实现权利要求1-17任一项所述的算子处理方法。

算子处理方法及算子处理装置、电子设备和可读存储介质

技术领域

[0001] 本公开的实施例涉及一种算子处理方法、算子处理装置、电子设备和计算机可读存储介质。

背景技术

[0002] 在人工智能芯片上运行神经网络需要大量算子的支撑，神经网络可用于语音识别、图像识别、自然语言识别等领域，这些算子可以是预定义的底层算子，也可以是用户自定义的算子。算子之间的数据交互往往通过访存的方式实现，导致代码的执行效率较为低下。通常的做法是将若干个算子根据一定的模式组合在一起形成融合算子，融合算子内部可以通过寄存器交换数据，从而提高运行效率。

发明内容

[0003] 本公开至少一个实施例提供一种算子处理方法，包括：获取用于描述计算过程的计算图，其中，所述计算图包含M个算子节点以及所述M个算子节点之间的连接关系，所述M个算子节点中的每个对应至少一个算子；对所述计算图进行拆分，得到N个子图，其中，所述N个子图中的每个包含所述M个算子节点中的至少一个算子节点，所述N个子图包括第一子图，所述第一子图对应K个第一融合算子，所述K个第一融合算子为将所述第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子；确定所述N个子图分别对应的算子代码，其中，所述N个子图分别对应的算子代码包括所述第一子图对应的所述K个第一融合算子的算子代码；基于所述N个子图分别对应的算子代码，得到用于执行所述计算过程的代码，其中，M、N和K均为不小于1的整数。

[0004] 例如，在本公开一实施例提供的算子处理方法中，所述第一子图包括的多个算子节点为P个算子节点，其中，P为大于1的整数；确定所述N个子图分别对应的算子代码，包括：针对所述第一子图，确定所述P个算子节点分别对应的算子的代码；基于所述P个算子节点分别对应的算子的代码，得到所述第一子图对应的所述K个第一融合算子的算子代码。

[0005] 例如，在本公开一实施例提供的算子处理方法中，针对所述第一子图，确定所述P个算子节点分别对应的算子的代码，包括：针对所述P个算子节点中的每个算子节点，执行如下操作：在所述算子节点对应的算子为预定义算子的情况下，针对所述预定义算子，获取与所述预定义算子对应的配置参数和代码模块，并基于所述配置参数和代码模块，得到所述预定义算子的代码；在所述算子节点对应的算子为自定义算子的情况下，针对所述自定义算子，对所述自定义算子进行编译，以得到所述自定义算子的代码。

[0006] 例如，在本公开一实施例提供的算子处理方法中，基于所述P个算子节点分别对应的算子的代码，得到所述第一子图对应的所述K个第一融合算子的代码，包括：将所述P个算子节点分别对应的算子的代码进行组合，得到所述K个第一融合算子的算子代码。

[0007] 例如，在本公开一实施例提供的算子处理方法中，对所述计算图进行拆分，得到N个子图，包括：若Q个算子节点对应的算子的目标属性相同，则将所述Q个算子节点划分为一

个子图，其中，所述目标属性包括类型属性、计算属性和数据传输属性中的至少一种，其中，Q为大于1的整数。

[0008] 例如，在本公开一实施例提供的算子处理方法中，对所述计算图进行拆分，得到N个子图，包括：若所述Q个算子节点对应的算子均为相同类型的算子，则将所述Q个算子节点划分为一个子图。

[0009] 例如，在本公开一实施例提供的算子处理方法中，对所述计算图进行拆分，得到N个子图，包括：若所述Q个算子节点对应的算子配置为通过寄存器传输数据，则将所述Q个算子节点划分为一个子图。

[0010] 例如，在本公开一实施例提供的算子处理方法中，对所述计算图进行拆分，得到N个子图，包括：若所述Q个算子节点对应的算子配置为在同一个计算单元上运行，则将所述Q个算子节点划分为一个子图。

[0011] 例如，在本公开一实施例提供的算子处理方法中，对所述计算图进行拆分，得到N个子图，包括：若Q个算子节点对应的算子的类型和执行顺序与预定融合算子包含的算子的类型和执行顺序一致，则将所述Q个算子节点划分为一个子图。

[0012] 例如，在本公开一实施例提供的算子处理方法中，所述Q个算子节点在根据所述计算图确定的执行顺序上为彼此连续的或彼此并列的。

[0013] 例如，在本公开一实施例提供的算子处理方法中，所述K个第一融合算子中的至少一个包括多个算子，所述多个算子依次连接并且顺次执行；对于所述多个算子中的相邻两个算子，在执行顺序上的前一个算子的计算结果数据作为后一个算子的输入数据。

[0014] 例如，在本公开一实施例提供的算子处理方法中，所述N个子图还包括第二子图，所述第二子图对应R个第二融合算子，所述R个第二融合算子为将所述第二子图包括的多个算子节点对应的多个算子进行融合而得到的算子，其中，R为不小于1的整数；所述N个子图分别对应的算子代码还包括所述第二子图对应的所述R个第二融合算子的算子代码；所述方法还包括：在确定所述N个子图分别对应的算子代码之前，对所述K个第一融合算子和所述R个第二融合算子进行优化处理，以得到分别针对所述K个第一融合算子和所述R个第二融合算子的配置信息。

[0015] 例如，在本公开一实施例提供的算子处理方法中，对所述K个第一融合算子和所述R个第二融合算子进行优化处理，以得到针对所述K个第一融合算子和所述R个第二融合算子的配置信息，包括：对所述K个第一融合算子和所述R个第二融合算子进行资源配置优化，以得到针对每个所述第一融合算子和每个所述第二融合算子的优化后的资源配置信息，其中，所述资源包括计算资源和/或存储资源。

[0016] 例如，在本公开一实施例提供的算子处理方法中，对所述K个第一融合算子和所述R个第二融合算子进行优化处理，以得到针对所述K个第一融合算子和所述R个第二融合算子的配置信息，包括：基于所述计算图的计算过程，针对所述K个第一融合算子和所述R个第二融合算子制定同步策略，以得到针对所述K个第一融合算子和所述R个第二融合算子的同步配置信息。

[0017] 例如，在本公开一实施例提供的算子处理方法中，基于所述计算图的计算过程，针对所述K个第一融合算子和所述R个第二融合算子制定同步策略，以得到针对所述K个第一融合算子和所述R个第二融合算子的同步配置信息，包括：至少针对所述K个第一融合算子

和所述R个第二融合算子中的两个融合算子执行如下操作:若所述两个融合算子存在数据依赖关系,并且数据接收方的调度单元标识与数据产生方的调度单元标识以及所述数据产生方至所述数据接受方之间的调度单元标识均不相同,则将所述两个融合算子配置为进行数据同步处理,其中,所述数据产生方为所述两个融合算子中的一者,所述数据产生方为另一者。

[0018] 例如,在本公开一实施例提供的算子处理方法中,所述第一融合算子包括所述第一子图包括的多个算子节点对应的多个算子;所述方法还包括:针对所述第一融合算子,在所述第一融合算子包含的所述多个算子之间进行优化处理,以得到针对所述多个算子的配置信息,其中,所述优化处理包括资源配置优化和/或制定同步策略。

[0019] 例如,在本公开一实施例提供的算子处理方法中,基于所述N个子图分别对应的算子代码,得到所述计算图的代码,包括:将所述N个子图分别对应的算子代码进行组合,得到所述计算图的代码。

[0020] 本公开至少一个实施例提供一种算子处理装置,包括获取模块、拆分模块、确定模块和代码模块,获取模块配置为获取用于描述计算过程的计算图,其中,所述计算图包含M个算子节点以及所述M个算子节点之间的连接关系,所述M个算子节点中的每个对应至少一个算子;拆分模块配置为对所述计算图进行拆分,得到N个子图,其中,所述N个子图中的每个包含所述M个算子节点中的至少一个算子节点,所述N个子图包括第一子图,所述第一子图对应K个第一融合算子,所述K个第一融合算子为将所述第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子;确定模块配置为确定所述N个子图分别对应的算子代码,其中,所述N个子图分别对应的算子代码包括所述第一子图对应的所述K个第一融合算子的算子代码;代码模块配置为基于所述N个子图分别对应的算子代码,得到用于执行所述计算过程的代码;其中,M、N和K均为不小于1的整数。

[0021] 本公开至少一个实施例提供一种电子设备,包括处理器和存储器,存储器存储有一个或多个计算机程序模块,其中,所述一个或多个计算机程序模块被配置为由所述处理器执行,用于实现本公开任一实施例提供的算子处理方法。

[0022] 本公开至少一个实施例提供一种计算机可读存储介质,存储有非暂时性计算机可读指令,当所述非暂时性计算机可读指令由计算机执行时实现本公开任一实施例提供的算子处理方法。

附图说明

[0023] 为了更清楚地说明本公开实施例的技术方案,下面将对实施例的附图作简单地介绍,显而易见地,下面描述中的附图仅仅涉及本公开的一些实施例,而非对本公开的限制。

[0024] 图1示出了一种生成神经网络的代码的流程图;

[0025] 图2示出了本公开至少一实施例提供的一种算子处理方法的流程图;

[0026] 图3示出了本公开至少一实施例提供的一种计算图的部分区域的示意图;

[0027] 图4示出了本公开至少一实施例提供的处理流程的示意图;

[0028] 图5A示出了本公开至少一实施例提供的计算图的部分区域的示意图;

[0029] 图5B示出了将图5A所示的计算图部分区域拆分得到的子图的示意图;

[0030] 图6示出了本公开至少一实施例提供的融合算子的示意图;

[0031] 图7示出了本公开至少一个实施例提供的一种算子处理装置的示意框图；

[0032] 图8示出了本公开至少一个实施例提供的一种电子设备的示意框图；

[0033] 图9示出了本公开至少一个实施例提供的另一种电子设备的示意框图；以及

[0034] 图10示出了本公开至少一个实施例提供的一种计算机可读存储介质的示意图。

具体实施方式

[0035] 为使本公开实施例的目的、技术方案和优点更加清楚，下面将结合本公开实施例的附图，对本公开实施例的技术方案进行清楚、完整地描述。显然，所描述的实施例是本公开的一部分实施例，而不是全部的实施例。基于所描述的本公开的实施例，本领域普通技术人员在无需创造性劳动的前提下所获得的所有其他实施例，都属于本公开保护的范围。

[0036] 除非另外定义，本公开使用的技术术语或者科学术语应当为本公开所属领域内具有一般技能的人士所理解的通常意义。本公开中使用的“第一”、“第二”以及类似的词语并不表示任何顺序、数量或者重要性，而只是用来区分不同的组成部分。同样，“一个”、“一”或者“该”等类似词语也不表示数量限制，而是表示存在至少一个。“包括”或者“包含”等类似的词语意指出现该词前面的元件或者物件涵盖出现在该词后面列举的元件或者物件及其等同，而不排除其他元件或者物件。“连接”或者“相连”等类似的词语并非限定于物理的或者机械的连接，而是可以包括电性的连接，不管是直接的还是间接的。“上”、“下”、“左”、“右”等仅用于表示相对位置关系，当被描述对象的绝对位置改变后，则该相对位置关系也可能相应地改变。

[0037] 图1示出了一种生成神经网络的代码的流程图。如图1所示，预定义底层算子库101中可以包含多个常用的预定义算子(或称为底层预定义算子)；此外，该流程还使用一项或多项自定义算子，该自定义算子为不包含在预定义底层算子库101内的、用户自定义的算子。可以针对特定应用场景，预先基于预定义底层算子库101形成对应的预定义融合算子库102，预定义融合算子库102包括该特定应用场景所需的融合算子(例如融合算子A1～融合算子Ag，g为大于1的整数)。在需要构建神经网络105时，根据预定义融合算子库102中相关的融合算子的代码和/或自定义算子103的算子代码，得到神经网络的计算图104的代码。

[0038] 不同的神经网络、同一神经网络的不同层以及不同的数据结构需要的融合算子可能不同，所以针对特定的神经网络、特定的网络层以及特定的数据结构需要专门定制和优化所需要的融合算子，从而组合成计算图，以运行特定情况下的神经网络，该过程为如图1所示的自下而上的处理过程。在处理过程中，由于这些预定义融合算子库中的融合算子是针对特定情况开发的，所以适用场景较窄、泛化能力较弱。

[0039] 本公开至少一个实施例提供一种算子处理方法、算子处理装置、电子设备和计算机可读存储介质。该算子处理方法包括：获取用于描述计算过程的计算图，其中，计算图包含M个算子节点以及M个算子节点之间的连接关系，M个算子节点中的每个对应至少一个算子；对计算图进行拆分，得到N个子图，其中，N个子图中的每个包含M个算子节点中的至少一个算子节点，N个子图包括第一子图，第一子图对应K个第一融合算子，该K个第一融合算子为将第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子；确定N个子图分别对应的算子代码，其中，N个子图分别对应的算子代码包括第一子图对应的K个第一融合算子的算子代码；基于N个子图分别对应的算子代码，得到用于执行计算过程的代码，其

中,M、N和K均为不小于1的整数。

[0040] 该实施例中,算子处理方法采用自上而下的方式生成融合算子,即从全局的视角将计算图拆分成多个子图,根据子图确定待生成的融合算子的结构,然后生成相应的融合算子,并进一步生成计算图的代码。该算子处理方法可以在生成融合算子的过程中将网络结构中的共性抽象出来,没有包含特定情况的信息,可以适用于各种场景,适用范围广,泛化能力强。

[0041] 图2示出了本公开至少一实施例提供的一种算子处理方法的流程图。

[0042] 如图2所示,该方法可以包括步骤S210～S240。

[0043] 步骤S210:获取用于描述计算过程的计算图,计算图包含M个算子节点以及M个算子节点之间的连接关系,M个算子节点中的每个对应至少一个算子。

[0044] 步骤S220:对计算图进行拆分,得到N个子图,N个子图中的每个包含M个算子节点中的至少一个算子节点,N个子图包括第一子图,第一子图对应K个第一融合算子,该K个第一融合算子为将第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子。

[0045] 步骤S230:确定N个子图分别对应的算子代码,N个子图分别对应的算子代码包括第一子图对应的K个第一融合算子的算子代码。

[0046] 步骤S240:基于N个子图分别对应的算子代码,得到用于执行计算过程的代码。

[0047] 例如,M、N和K均为不小于1的整数。

[0048] 例如,本公开的上述实施例以计算图为神经网络模型的计算图为例进行说明,但是本公开并不以此为限,在实际应用中,计算图可以是任意计算模型的计算图。

[0049] 例如,在步骤S210中,计算图可以根据神经网络的描述信息自动生成,例如,可以根据神经网络的网络结构自动生成有向无环图(即计算图),该有向无环图包括多个节点以及节点之间的连线,这些连线表征节点之间的数据依赖关系和数据流向。

[0050] 例如,计算图利用图形来描述计算模型的计算过程。图3示出了本公开至少一实施例提供的一种计算图的部分区域的示意图,如图3所示,计算图包括多个算子节点,每个算子节点可以对应一个算子,此处的算子可以理解为预定义底层算子库中的预定义算子或者用户自定义的算子。例如,在图3中,计算图包括算子节点301、302和303,算子节点301对应Conv算子(卷积算子)、算子节点302对应BatchNormalization算子(批量归一化算子)、算子节点303对应Relu算子(线性整流激活函数)对应的节点等。在另一些实施例中,每个算子节点也可以对应多个算子。此外,计算图还可以包括其他节点,例如变量节点、计算结果节点等,变量节点是计算图中使用的参数,如图3所示的节点304为变量节点,节点304对应Weight(权重)参数;节点305、306和307为计算结果节点,分别对应计算结果数据(张量数据)Tensor1、Tensor2和Tensor3,计算结果节点可以表示与其相邻的前一个算子的计算结果数据。

[0051] 为了更清楚地描述本公开实施例,在以下的实施例中仅描述计算图中的算子节点,而忽略其他节点。

[0052] 例如,计算图还包括多个算子节点之间的连接关系,连接关系例如可以以带箭头的线条表示,相互连接的两个算子之间存在数据传输关系(或数据依赖关系)。例如,Conv算子的计算结果数据作为Batch Normalization算子的输入数据,则可以将Conv算子与Batch Normalization算子相连接并且连接线的箭头指向Batch Normalization算子。

[0053] 图4示出了本公开至少一实施例提供的处理流程的示意图，如图4所示，在步骤S220中，可以将计算图401拆分为多个子图，例如拆分为子图B1～BN。每个子图作为计算图的一部分，包含计算图的部分算子节点以及该部分算子节点之间的连接关系。例如，不同子图包括的算子节点的数量可以相同或不同，不同子图包括的算子节点之间没有交集。例如，N个子图中的至少一个子图包含多个(两个或两个以上)算子节点，例如，N个子图包括第一子图，该第一子图包含多个算子节点。在一些示例中，N个子图中的每个子图均可以包含多个算子节点；在另一些示例中，N个子图中可以有部分子图包含多个算子节点，部分子图包含一个算子节点。为了描述方便，以下将包含多个算子节点的子图称为多节点子图，将包含一个算子节点的子图称为单节点子图。

[0054] 例如，如图4所示，在得到多个子图B1～BN后，可以确定与该多个子图B1～BN对应的多个融合算子，例如融合算子C1～Cs(s为大于1的整数，s可以小于、等于或大于N)，这里所说的确定融合算子是指确定待生成的融合算子的结构信息，例如确定待生成的融合算子包含的算子的种类和连接关系等，以在步骤S230中进一步根据待生成的融合算子的这些信息，生成对应的融合算子。

[0055] 例如，在一些示例中，可以针对每个多节点子图，确定一个或多个待生成的融合算子，每个待生成的融合算子可以根据对应子图包含的全部或部分算子节点融合得到。例如，在某个多节点子图对应一个融合算子的情况下，该融合算子可以根据该多节点子图包含的全部算子节点对应的多个算子融合得到。在某个多节点子图对应两个或两个以上融合算子的情况下，该两个或两个以上融合算子中的每个融合算子可以根据该多节点子图包含的算子节点中的部分算子节点对应的算子融合得到。在以下的一些实施例中，以每个多节点子图对应一个融合算子为例进行说明。

[0056] 图5A示出了本公开至少一实施例提供的计算图的部分区域的示意图。图5B示出了将图5A所示的计算图部分区域拆分得到的子图的示意图。

[0057] 如图5A所示的计算图的部分区域包括算子节点501～509，如图5B所示，可以将算子节点502、503和504划分为一个多节点子图B1，进而确定对应的待生成的融合算子CBR(即Conv+Batch Normalization+Relu)；将算子节点505和506划分为一个多节点子图B2，进而确定对应的待生成的融合算子CB(即Conv+Batch Normalization)；将算子节点507、508和509划分为一个多节点子图B3，进而确定对应的待生成的融合算子CBA(即Conv+Batch Normalization+Add)。例如，在另一些示例中，也可以根据一个多节点子图确定多个待生成的融合算子。

[0058] 例如，第一子图对应K个第一融合算子，其中至少一个第一融合算子包括多个算子。以下以K为1为例进行说明，第一子图对应的第一融合算子可以包括第一子图包括的多个算子节点对应的多个算子，该多个算子依次连接并且顺次执行。对于该多个算子中的相邻两个算子，在执行顺序上的前一个算子的计算结果数据作为后一个算子的输入数据。例如，将图5B所示的子图B1作为第一子图，将CBR算子作为第一融合算子，Conv算子、Batch Normalization算子和Relu算子依次连接并且顺次执行，Conv算子的计算结果数据作为Batch Normalization算子的输入数据，Batch Normalization算子的计算结果数据作为Relu算子的输入数据。例如，在第一子图对应两个或以上的第一融合算子的情况下，每个第一融合算子可以包括第一子图对应的多个算子中的部分算子。

[0059] 例如，在步骤S230中，确定N个子图分别对应的算子代码包括确定N个子图对应的多个融合算子的算子代码。如图4所示，根据多个子图B1～BN得到待生成的多个融合算子C1～Cs后，可以根据预定义底层算子库403和/或自定义算子404，确定待生成的融合算子C1～Cs的代码，生成得到融合算子C1～Cs。

[0060] 图6示出了本公开至少一实施例提供的融合算子的示意图。如图3和图6所示，在将Conv算子、Batch Normalization算子和Relu算子进行融合之前，每个算子计算得到的结果数据(Tensor1～Tensor3)均需要写到内存，下一个算子再从内存中读取数据，算子之间的数据交互通过访存的方式实现，导致代码的执行效率较为低下。在将Conv算子、Batch Normalization算子和Relu算子融合得到融合算子CBR之后，根据融合算子CBR可以直接计算得到结果Tensor3，融合算子CBR的中间计算结果可以用执行该融合算子的处理器(例如图像处理器(GPU)、数据处理器(DPU)或特定领域架构(DSA)芯片(例如AI加速器)等)的寄存器来缓冲，而无需经过位于处理器之外的内存，因而提高了执行效率。

[0061] 例如，除第一子图之外，还可以包括其他子图。例如，N个子图在第一子图之外还可以包括第二子图，第二子图对应R个第二融合算子(R为不小于1的整数)，该R个第二融合算子为将第二子图包括的多个算子节点对应的多个算子进行融合而得到的算子。N个子图分别对应的算子代码包括第一子图对应的K个第一融合算子的算子代码以及第二子图对应的R个第二融合算子的算子代码。

[0062] 例如，在得到N个子图分别对应的算子代码之后，可以根据该N个子图分别对应的算子代码，得到用于执行计算图的计算过程的代码，进而可以得到神经网络等模型的运行代码。

[0063] 根据本公开实施例的算子处理方法，采用自上而下的方式生成融合算子，即从全局的视角将计算图拆分成多个子图，确定了待生成的融合算子的结构，然后生成相应的融合算子，并进一步生成计算图的代码。该算子处理方法可以在生成融合算子的过程中将网络结构中的共性抽象出来，没有包含特定情况的信息，可以适用于各种场景，适用范围广，泛化能力强。

[0064] 本公开至少一实施例的算子处理方法可以应用于人工智能芯片(例如图像处理器(GPU)、数据处理器(DPU)或特定领域架构(DSA)芯片(例如AI加速器)等)软件栈中融合算子代码生成的应用场景。融合算子是人工智能芯片高效计算的基础，该算子处理方法可以有益于实现人工智能芯片的高效计算。

[0065] 例如，在一些示例中，在步骤S220中，若存在Q个算子节点对应的算子的目标属性相同，则可以将该Q个算子节点划分为一个子图，其中，目标属性包括类型属性、计算属性和数据传输属性中的至少一种，其中，Q为大于1的整数。例如，该Q个算子节点在根据计算图确定的执行顺序上为彼此连续的或彼此并列的，也就是说，若执行顺序上连续或并列的Q个算子节点对应的算子的目标属性相同，则可以将该Q个算子节点划分为一个子图。

[0066] 例如，以类型属性为例，若该Q个算子节点对应的算子均为相同类型的算子，则将该Q个算子节点划分为一个子图。例如，若计算图中存在顺序连接的Q个算子节点并且该Q个算子节点均对应同一类型(例如Conv类型)的算子，则可以将该Q个Conv类型的算子节点划分为一个子图，例如，将Q个连续的Conv类型的算子融合为一个融合算子。基于这一方式，可以将重复的算子节点进行融合，提高执行效率。

[0067] 例如,以数据传输属性为例,若该Q个算子节点对应的算子配置为通过寄存器传输数据,则将该Q个算子节点划分为一个子图。例如,如图5B所示,若算子节点507、508和509对应的算子配置为通过寄存器传输数据,即算子节点507对应的Conv算子的计算结果存储于寄存器,算子节点508对应的Batch Normalization算子从该寄存器中获取Conv算子的计算结果,并且Batch Normalization算子的计算结果存储于寄存器,算子节点509对应的Add算子从该寄存器中获取Conv算子的计算结果,Add算子的计算结果例如可以存储于内存,这种情况下,可以将算子节点507、508和509划分为一个子图,以将Conv算子、Batch Normalization算子和Add算子融合为一个融合算子。例如,在另一些示例中,若Q个算子节点对应的算子配置为通过缓存传输数据,该Q个算子节点对应的算子之间传输的数据同样无需经过内存,因此也可以将该Q个算子节点划分为一个子图。基于这一方式,可以保证融合算子的中间计算结果通过寄存器等非访存的方式传输。

[0068] 例如,以计算属性为例,若该Q个算子节点对应的算子配置为在同一个计算单元上运行,则将Q个算子节点划分为一个子图。例如,可以采用多种计算单元来执行计算图的计算过程,例如包括张量计算核(tcore)和向量计算核(vcore)这两种计算单元。如图5B所示,若算子节点502、503和504对应的算子均利用tcore计算单元执行计算操作,算子节点505和506对应的算子均利用vcore计算单元执行计算操作,则可以将算子节点502、503和504划分为一个子图,将算子节点505和506划分为另一个子图。基于这一方式,可以保证融合算子的代码运行于同一计算单元上。

[0069] 例如,在一些示例中,在步骤S220中,若Q个算子节点对应的算子的类型和执行顺序与预定融合算子包含的算子的类型和执行顺序一致,则将该Q个算子节点划分为一个子图。例如,可以预先定义一些常用类型的融合算子,作为预定融合算子,当计算图中出现能够组合为预定融合算子的连续多个算子节点时,可以将该多个算子节点划分为一个子图,以融合得到预定融合算子。预定融合算子例如可以包括CBR类型的融合算子,CBR类型的融合算子的数据流通路径为Conv算子-Batch Normalization算子-Relu算子。若计算图中包括依次相连的Conv算子节点、Batch Normalization算子节点和Relu算子节点,则可以将该三个算子节点划分为一个子图,以融合得到CBR类型的融合算子。

[0070] 例如,可以根据实际情况,使用上述多种子图划分方式中的一种,或者也可以将上述多种子图划分方式中的两种或更多种进行组合。例如,将计算属性和数据传输属性组合,连续Q个算子节点对应的算子配置为在同一计算单元执行并且通过寄存器传输数据的情况下,将该Q个算子节点划分为一个子图。

[0071] 例如,如上所述,N个子图可以对应多个融合算子,若每个子图对应一个融合算子,那么每个融合算子可以是根据一个多节点子图包括的多个算子节点对应的多个算子进行融合而得到的算子。如图4所示,在执行步骤130中的确定N个子图分别对应的算子代码之前,可以利用优化管理器对多个待生成的融合算子进行优化处理,以得到分别针对多个待生成的融合算子的配置信息,其中,优化处理包括资源配置优化和/或制定同步策略。

[0072] 例如,在另一些示例中,也可以仅对多个融合算子中的部分融合算子进行优化处理。

[0073] 例如,在另一些示例中,N个子图除了包括多节点子图之外,还可以包括单节点子图,每个单节点子图可以包括一个算子节点,即对应至少一个算子。如图5B所示,例如,将算

子节点501划分为一个子图，该子图为单节点子图，该单节点子图对应Relu算子，该算子节点501对应的Relu算子无需进行融合处理，可以作为单独的算子使用。这种情况下，N个子图不仅对应多个融合算子还可以对应至少一个未经融合的单独的算子(或称为非融合算子)。在进行优化处理时，可以对N个子图对应的多个融合算子以及至少一个非融合算子共同进行优化处理。例如，在一些实施例中，可以将非融合算子按照融合算子的数据结构进行表达，因此，可以将具有融合算子数据结构的非融合算子看作是融合算子。在以下的关于优化处理的一些实施例中，将由若干个算子融合得到的融合算子以及具有融合算子数据结构的非融合算子统称为融合算子，也就是说，在以下的关于优化处理的一些实施例中，融合算子可以包括由若干个算子融合得到的融合算子，也可以包括以融合算子数据结构表达的非融合算子。

[0074] 例如，为了描述清楚，在以下的一些实施例中，以对第一融合算子和第二融合算子进行优化处理为例进行说明。例如，在确定N个子图分别对应的算子代码之前，可以对K个第一融合算子和R个第二融合算子进行优化处理，以得到分别针对K个第一融合算子和R个第二融合算子的配置信息。对其他融合算子的优化处理，可以参照该第一融合算子和第二融合算子。

[0075] 例如，优化处理可以包括对资源配置进行优化，例如，可以对K个第一融合算子和R个第二融合算子进行资源配置优化，以得到针对每个第一融合算子和每个第二融合算子的优化后的资源配置信息。

[0076] 例如，资源可以包括计算资源，例如处理器资源，例如以GPU为例，这些计算资源包括张量计算核、特殊函数单元(SFU)等。在优化处理过程中，可以调整分配至每个第一融合算子和每个第二融合算子的计算资源，以使计算资源的分配更为合理，提升计算资源的利用率。

[0077] 例如，资源可以包括存储资源，例如存储器(例如处理器之外的内存、处理器内的寄存器等)的存储空间。在优化处理过程中，可以调整分配至每个第一融合算子和每个第二融合算子的存储资源，以使存储空间的分配更为合理，提升存储空间的利用率。

[0078] 例如，优化处理还可以包括制定同步策略，例如，基于计算图的计算过程，针对K个第一融合算子和R个第二融合算子制定同步策略，以得到针对K个第一融合算子和R个第二融合算子的同步配置信息。

[0079] 例如，对于单分支网络(即整个神经网络只有一个分支)，数据依赖关系单一、模式固定，能够简单地达到数据同步。然而，对于多分支网络，没有固定的分支模式，不同的网络也会存在不同的多分支结构。为了保证数据同步，相关技术中根据数据依赖关系确定算子的强约束执行顺序，算子之间保证完全串行的计算过程，即：总是前一个算子的计算完全结束、且写出结果之后再启动下一个算子的计算。然而，这种执行顺序上的强约束使得计算过程完全串行化，这严重降低了神经网络的计算效率。基于以上，如何保证神经网络中融合算子间数据同步是网络计算过程高效执行的必要前提，本公开实施例的算子处理方法通过在融合算子间制定同步策略来应对各种类型神经网络结构，面对复杂的数据依赖关系仍能保证准确、高效的执行顺序。

[0080] 例如，可以根据计算过程，确定哪些融合算子之间需要执行数据同步处理，哪些融合算子之间不需要执行数据同步处理。对于需要进行数据同步处理的两个融合算子，需要

等待前一个算子计算结束后，再开始后一个算子的计算操作，进而可以保证具有数据依赖关系的融合算子之间的顺序执行，以实现神经网络的高效计算。

[0081] 例如，至少针对K个第一融合算子和R个第二融合算子中的两个融合算子执行如下操作：若该两个融合算子存在数据依赖关系，并且数据接收方的调度单元标识与数据产生方的调度单元标识以及数据产生方至数据接受方之间的调度单元标识均不相同，则将两个融合算子配置为进行数据同步处理，其中，数据产生方为两个融合算子中的一者，数据产生方为另一者。例如，可以针对每两个融合算子执行上述操作。

[0082] 例如，以两个融合算子为例(例如一个第一融合算子和一个第二融合算子)为例进行说明，可以根据执行顺序，将第一融合算子和第二融合算子中的一者作为数据产生方，另一者作为数据接收方(例如数据消费方)。例如，将两者中执行顺序靠前的作为数据产生方，执行顺序靠后的作为数据接收方。若第一融合算子和第二融合算子存在数据依赖关系，并且数据接收方的调度单元标识与数据产生方的调度单元标识以及数据产生方至数据接受方之间的调度单元标识均不相同，则将第一融合算子和第二融合算子配置为进行数据同步处理；否则，将第一融合算子和第二融合算子配置为不进行数据同步处理。

[0083] 例如，第一融合算子和第二融合算子存在数据依赖关系包括其中一个融合算子(例如第二融合算子)的计算过程需要直接或间接地使用另一个融合算子(例如第一融合算子)的计算结果，即其中一个融合算子(例如第一融合算子)的输出数据会流向另一个融合算子(例如第二融合算子)。

[0084] 如图5B所示，子图B2对应的融合算子CB的计算过程直接使用到子图B1对应的融合算子CBR的计算结果，因此该融合算子CB和融合算子CBR之间存在数据依赖关系，并且，融合算子CBR为数据产生方，融合算子CB为数据接收方；再例如，子图B3对应的融合算子CBA的计算过程间接使用到子图B1对应的融合算子CBR的计算结果，因此该融合算子CBA和融合算子CBR之间也存在数据依赖关系，并且，融合算子CBA为数据接收方，融合算子CBR为数据产生方；再例如，若某一融合算子(图中未示出)的计算过程所需的数据与子图B1对应的融合算子CBR的计算结果不相关，并且子图B1对应的融合算子CBR的计算过程也无需使用到该某一融合算子的计算结果，则可以认为该某一融合算子与子图B1对应的融合算子CBR不存在数据依赖关系。

[0085] 例如，若第一融合算子和第二融合算子不存在数据依赖关系，则两者不存在数据交互，因而无需进行数据同步处理。

[0086] 例如，若第一融合算子和第二融合算子存在数据依赖关系，还需要确定第一融合算子和第二融合算子的调度单元标识是否相同，若第一融合算子和第二融合算子的调度单元标识相同，则将第一融合算子和第二融合算子配置为不进行数据同步处理。

[0087] 例如，调度单元(warp)对应于包括多个线程的线程组，调度单元以轮询调度的方式执行并行计算。一般而言，程序包括多个工作组(workgroup)，每一个工作组包括多个线程组，每一个线程组包括多个线程(thread)。同一个工作组中的线程可以按照调度单位分组，然后一组一组地调度至硬件去执行。这个调度单位称作线程组。调度单元warp可以称为是最基本的执行单元，一个调度单元warp包含32个并行线程，这些线程以不同数据资源执行相同的指令。

[0088] 例如，一对融合算子之间是否进行数据同步既取决于这两者是否存储数据依赖关

系,还依赖于这两者所处的调度单元。作为示例,对于不同的调度单元,以调度单元标识进行区分。

[0089] 例如,神经网络中的融合算子的计算执行依赖于调度单元的调度,在计算执行之前,可以为各个融合算子分配调度单元标识(warp id),不同算子可能分配相同的调度单元标识(即,位于相同的调度单元),也可能分配不同的调度单元标识,不同调度单元标识下的融合算子属于不同的调度单元,能够以诸如轮询调度(round-robin)的方式并行执行。存在数据依赖关系的融合算子之间是否需要进行数据同步,还需要依据这两个算子之间调度单元标识的分配情况。

[0090] 例如,如果第一融合算子和第二融合算子存在数据依赖关系,并且第一融合算子和第二融合算子分配在同一个调度单元上,即,第一融合算子和第二融合算子具有相同的调度单元标识,则第一融合算子和第二融合算子之间无需进行数据同步处理,这是由于,同一个调度单元上的融合算子是顺序执行的,即在执行顺序上能够保证数据同步。

[0091] 例如,若第一融合算子和第二融合算子存在数据依赖关系并且第一融合算子的调度单元标识和第二融合算子的调度单元标识不相同,则需要对第一融合算子和第二融合算子进行数据同步处理。例如,在执行顺序上,第二融合算子位于第一融合算子之后,如果第二融合算子的调度单元标识和第一融合算子的调度单元标识不相同,则可以对第一融合算子和第二融合算子进行数据同步处理。

[0092] 例如,第一融合算子和第二融合算子存在数据依赖关系,以第一融合算子为数据产生方以及第二融合算子为数据接收方为例,若在执行顺序上第一融合算子和第二融合算子之间具有至少一个中间算子,判断第一融合算子的调度单元标识与第二融合算子的调度单元标识和该至少一个中间算子的调度单元标识是否均不相同,若均不相同则需要进行同步,若第一融合算子与第二融合算子和中间算子中的至少一者的调度单元标识相同,则无需进行数据同步处理。例如,在执行顺序上,第二融合算子位于第一融合算子之后,如果第二融合算子的调度单元标识与其前面从第一融合算子开始至该第二融合算子之间的若干个融合算子的调度单元标识均不相同,则第一融合算子和第二融合算子之间需要进行数据同步处理。否则,不需要进行同步。例如,第一融合算子和第二融合算子之间具有第三融合算子,若第一融合算子与第二融合算子和第三融合算子的调度单元标识均不相同,则需要进行数据同步处理,若第一融合算子与第二融合算子的调度单元标识相同和/或第一融合算子与第三融合算子的调度单元标识相同,则无需进行数据同步处理。

[0093] 例如,在一些实施例中,上述的调度单元标识可以替换为计算标识,该计算标识可以理解为是最小计算粒度(例如线程)的标识。例如,若第一融合算子和第二融合算子存在数据依赖关系,并且第一融合算子和第二融合算子具有相同的计算标识,例如第一融合算子和第二融合算子运行在同一个线程,由于运行在同一线程上的第一融合算子和第二融合算子会按照顺序执行,即在执行顺序上能够保证数据同步,因此不需要再对第一融合算子和第二融合算子进行数据同步。再例如,以第一融合算子为数据产生方以及第二融合算子为数据接收方为例,第一融合算子和第二融合算子之间具有第三融合算子,若第一融合算子与第二融合算子和第三融合算子的计算标识均不相同,则需要第一融合算子和第二融合算子进行数据同步处理,若第一融合算子与第二融合算子的计算标识相同和/或第一融合算子与第三融合算子的计算标识相同,则第一融合算子和第二融合算子无需进行数据同步

处理。

[0094] 例如，如果数据接收方的输入数据来自于多个数据产生方，即存在多对数据依赖关系，则需要对每一对数据依赖关系分别执行上述判断过程，从而有可能一个数据接收方需要与多个数据产生方进行同步，也有可能多个数据接收方需要与一个数据产生方进行同步。不论是一个数据接收方与一个数据产生方同步，还是一个数据接收方与多个数据产生方同步，或者多个数据接收方与一个数据产生方同步，同步关系总是成对的，并且，在同一个数据产生方或者同一个数据接收方上的多对同步关系可以使用不同的同步屏障。

[0095] 例如，本公开实施例的算子处理方法，在生成代码之前，先对待生成的融合算子进行全局优化处理，这里的全局优化可以理解为计算图整体视角的优化。硬件资源总是有限的，从计算图整体的角度考虑对融合算子进行高效配置，使得整体的性能达到最优。相比于相关技术中在生成预定义融合算子库时对各个融合算子分别优化的方式，本公开实施例的优化方法在生成代码之前结合多个待生成的算子进行全局优化，考虑了全局而非单个融合算子，进而可以实现全局最优。

[0096] 例如，在对待生成的融合算子进行整体优化之后，还可以对每个融合算子进行内部优化。以第一融合算子为例，第一融合算子包括第一子图包括的多个算子节点对应的多个算子，针对第一融合算子，在第一融合算子包含的多个算子之间进行优化处理，以得到针对多个算子的配置信息，其中，优化处理包括资源配置优化和/或制定同步策略。

[0097] 例如，第一融合算子包括Conv算子、Batch Normalization算子和Relu算子，可以针对该三个算子进行资源配置优化，和/或对该三个算子指定同步策略。资源配置优化和指定同步策略的过程可以参考以上描述，在此不再赘述。针对单个融合算子进行资源配置优化等处理，可以充分利用硬件资源(寄存器)生成高性能代码，减少数据低效的搬运，提高并行性。

[0098] 例如，优化处理除了包括上述的资源配置优化和制定同步策略之外，还可以包括其他优化处理，例如推导融合算子之间的参数等，参数例如包括表征融合算子的计算结果数据是否需要写到缓存的参数等。

[0099] 例如，通过预先制定全局优化策略，优化管理器可以对需要生成的所有融合算子进行优化，包括配置资源、制定同步机制等。由于从全局的视角出发，既考虑了提高单个融合算子的性能，又考虑了邻近融合算子的影响，从而能够达到全局的性能最优。优化管理器可以为任意场景下的融合算子制定优化的策略和目标，不需要再人为地优化每个融合算子，将无限的融合算子优化工作转化为有限的优化管理器完善工作。

[0100] 例如，在完成优化处理之后，可以执行步骤S230，以生成各个子图分别对应的算子代码。例如，以第一子图对应一个第一融合算子为例，第一子图包括的多个算子节点例如为P个算子节点，其中，P为大于1的整数。在步骤S230中，可以针对第一子图，确定该P个算子节点分别对应的算子的代码；基于该P个算子节点分别对应的算子的代码，得到第一子图对应的第一融合算子的算子代码。例如，可以根据相应的组合方式，将该P个算子节点分别对应的算子的代码进行组合，得到第一融合算子的算子代码。

[0101] 例如，在确定该P个算子节点分别对应的算子的代码的过程中，可以针对P个算子节点中的每个算子节点，执行如下操作：在算子节点对应的算子为预定义算子的情况下，针对预定义算子，获取与预定义算子对应的配置参数和代码模块，并基于配置参数和代码模

块,得到预定义算子的代码;在算子节点对应的算子为自定义算子的情况下,针对自定义算子,对自定义算子进行编译,以得到自定义算子的代码。

[0102] 例如,若某一算子节点对应的算子为预定义算子,则从预定义底层算子库中查找到相应的预定义算子,并提取出该预定义算子的代码模板,获取该预定义算子的配置信息,将配置信息填充入代码模板中,即可得到该预定义算子的端子代码,其中,配置信息可以包括上述优化过程所确定的配置信息,还可以包括其他配置信息。若某一算子节点对应的算子为自定义算子,则可以利用编译器、汇编器等得到自定义算子的算子代码。在得到每个融合算子包含的多个算子的算子代码之后,将算子代码按照预定方式进行组合可以得到融合算子的算子代码。

[0103] 例如,在步骤S240中,可以将N个子图分别对应的算子代码进行组合,以得到用于执行计算图的计算过程的代码。例如,若N个子图对应融合算子C1～Cs,在得到融合算子C1～Cs的代码后,可以将融合算子C1～Cs的代码根据相应的组合方式进行组合,以得到计算图对应的代码。计算过程的代码还可以包括除融合算子的算子代码之外的其他信息,因此,在组合时,还可以结合其他信息的代码,例如存储信息的代码、参数信息的代码等,组合得到用于执行计算图的计算过程的代码。

[0104] 例如,在N个子图即包括多节点子图又包括单节点子图的情况下,在步骤S230中,除了确定多节点子图对应的多个融合算子的算子代码之外,还可以根据预定义底层算子库和/或自定义算子获得单节点子图对应的非融合算子的算子代码。然后,可以结合融合算子的算子代码、非融合算子的算子代码以及其他信息,组合得到用于执行计算图的计算过程的代码。

[0105] 本公开至少一实施例的算子处理方法,减轻了融合算子开发和优化的工程量,提高了融合算子的泛化能力。

[0106] 本公开至少一实施例的算子处理方法,将网络结构中的共性抽象出来,没有包含特定情况的信息,适用范围广,泛化能力强。

[0107] 本公开至少一实施例的算子处理方法,针对特定情形只需要配置特定参数就可以生成对应的高性能融合算子代码,减轻了融合算子开发和维护的工程量。

[0108] 本公开至少一实施例的算子处理方法,从全局的视角出发在融合算子之间配置资源、设置同步机制,可以更容易达到全局性能最优。

[0109] 图7示出了本公开至少一个实施例提供的一种算子处理装置700的示意框图。

[0110] 例如,如图7所示,该算子处理装置700包括获取模块710、拆分模块720、确定模块730和代码模块740。

[0111] 获取模块710配置为获取用于描述计算过程的计算图,其中,该计算图包含M个算子节点以及该M个算子节点之间的连接关系,该M个算子节点中的每个对应至少一个算子。获取模块710例如可以执行图2描述的步骤S210。

[0112] 拆分模块720配置为对该计算图进行拆分,得到N个子图,其中,该N个子图中的每个包含该M个算子节点中的至少一个算子节点,该N个子图包括第一子图,该第一子图对应K个第一融合算子,该K个第一融合算子为将该第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子。拆分模块720例如可以执行图2描述的步骤S220。

[0113] 确定单元730配置为确定该N个子图分别对应的算子代码,其中,该N个子图分别对

应的算子代码包括该第一子图对应的该K个第一融合算子的算子代码。确定单元730例如可以执行图2描述的步骤S230。

[0114] 代码模块740配置为基于该N个子图分别对应的算子代码，得到用于执行所述计算过程的代码。代码模块740例如可以执行图2描述的步骤S240。

[0115] 例如，获取模块710、拆分模块720、确定模块730和代码模块740可以为硬件、软件、固件以及它们的任意可行的组合。例如，获取模块710、拆分模块720、确定模块730和代码模块740可以为专用或通用的电路、芯片或装置等，也可以为处理器和存储器的结合。关于上述各个单元的具体实现形式，本公开的实施例对此不作限制。

[0116] 需要说明的是，本公开的实施例中，算子处理装置700的各个单元与前述的算子处理方法的各个步骤对应，关于算子处理装置700的具体功能可以参考关于算子处理方法的相关描述，此处不再赘述。图7所示的算子处理装置700的组件和结构只是示例性的，而非限制性的，根据需要，该算子处理装置700还可以包括其他组件和结构。

[0117] 例如，在本公开一实施例提供的算子处理装置中，该第一子图包括的多个算子节点为P个算子节点，其中，P为大于1的整数。确定模块730进一步配置为：针对该第一子图，确定该P个算子节点分别对应的算子的代码；基于该P个算子节点分别对应的算子的代码，得到该第一子图对应的该K个第一融合算子的算子代码。

[0118] 例如，在本公开一实施例提供的算子处理装置中，确定模块730进一步配置为：针对该P个算子节点中的每个算子节点，执行如下操作：在该算子节点对应的算子为预定义算子的情况下，针对该预定义算子，获取与该预定义算子对应的配置参数和代码模块，并基于该配置参数和代码模块，得到该预定义算子的代码；在该算子节点对应的算子为自定义算子的情况下，针对该自定义算子，对该自定义算子进行编译，以得到该自定义算子的代码。

[0119] 例如，在本公开一实施例提供的算子处理装置中，确定模块730进一步配置为：将该P个算子节点分别对应的算子的代码进行组合，得到该K个第一融合算子的算子代码。

[0120] 例如，在本公开一实施例提供的算子处理装置中，拆分模块720进一步配置为：若Q个算子节点对应的算子的目标属性相同，则将该Q个算子节点划分为一个子图，其中，该目标属性包括类型属性、计算属性和数据传输属性中的至少一种，其中，Q为大于1的整数。

[0121] 例如，在本公开一实施例提供的算子处理装置中，拆分模块720进一步配置为：若该Q个算子节点对应的算子均为相同类型的算子，则将该Q个算子节点划分为一个子图。

[0122] 例如，在本公开一实施例提供的算子处理装置中，拆分模块720进一步配置为：若该Q个算子节点对应的算子配置为通过寄存器传输数据，则将该Q个算子节点划分为一个子图。

[0123] 例如，在本公开一实施例提供的算子处理装置中，拆分模块720进一步配置为：若该Q个算子节点对应的算子配置为在同一个计算单元上运行，则将该Q个算子节点划分为一个子图。

[0124] 例如，在本公开一实施例提供的算子处理装置中，拆分模块720进一步配置为：若Q个算子节点对应的算子的类型和执行顺序与预定融合算子包含的算子的类型和执行顺序一致，则将该Q个算子节点划分为一个子图。

[0125] 例如，在本公开一实施例提供的算子处理装置中，该Q个算子节点在根据该计算图确定的执行顺序上为彼此连续的或彼此并列的。

[0126] 例如,在本公开一实施例提供的算子处理装置中,该K个第一融合算子中的至少一个包括多个算子,该多个算子依次连接并且顺次执行;对于该多个算子中的相邻两个算子,在执行顺序上的前一个算子的计算结果数据作为后一个算子的输入数据。

[0127] 例如,在本公开一实施例提供的算子处理装置中,该N个子图还包括第二子图,该第二子图对应R个第二融合算子,该R个第二融合算子为将该第二子图包括的多个算子节点对应的多个算子进行融合而得到的算子,其中,R为不小于1的整数;该N个子图分别对应的算子代码还包括该第二子图对应的该R个第二融合算子的算子代码。该算子处理装置还包括优化模块,该优化模块配置为:在确定该N个子图分别对应的算子代码之前,对该K个第一融合算子和该R个第二融合算子进行优化处理,以得到分别针对该K个第一融合算子和该R个第二融合算子的配置信息。

[0128] 例如,在本公开一实施例提供的算子处理装置中,优化模块进一步配置为:对该K个第一融合算子和R个该第二融合算子进行资源配置优化,以得到针对每个第一融合算子和每个第二融合算子的优化后的资源配置信息,其中,该资源包括计算资源和/或存储资源。

[0129] 例如,在本公开一实施例提供的算子处理装置中,优化模块进一步配置为:基于该计算图的计算过程,针对该K个第一融合算子和该R个第二融合算子制定同步策略,以得到针对该K个第一融合算子和该R个第二融合算子的同步配置信息。

[0130] 例如,在本公开一实施例提供的算子处理装置中,优化模块进一步配置为:至少针对所述K个第一融合算子和所述R个第二融合算子中的两个融合算子执行如下操作:若该两个融合算子存在数据依赖关系,并且数据接收方的调度单元标识与数据产生方的调度单元标识以及该数据产生方至该数据接受方之间的调度单元标识均不相同,则将该两个融合算子配置为进行数据同步处理,其中,该数据产生方为该两个融合算子中的一者,该数据产生方为另一者。

[0131] 例如,在本公开一实施例提供的算子处理装置中,该N个子图对应多个融合算子,每个该融合算子为根据一个子图包括的多个算子节点对应的多个算子进行融合而得到的算子。优化模块进一步配置为:在确定该N个子图分别对应的算子代码之前,对该多个融合算子进行优化处理,以得到分别针对该多个融合算子的配置信息,其中,该优化处理包括资源配置优化和/或制定同步策略。

[0132] 例如,在本公开一实施例提供的算子处理装置中,该第一融合算子包括该第一子图包括的多个算子节点对应的多个算子。优化模块进一步配置为:针对该第一融合算子,在该第一融合算子包含的该多个算子之间进行优化处理,以得到针对该多个算子的配置信息,其中,该优化处理包括资源配置优化和/或制定同步策略。

[0133] 例如,在本公开一实施例提供的算子处理装置中,代码模块740进一步配置为:将该N个子图分别对应的算子代码进行组合,得到用于执行该计算过程的代码。

[0134] 本公开的至少一个实施例还提供了一种电子设备,该电子设备包括处理器和存储器,存储器存储有一个或多个计算机程序模块。该一个或多个计算机程序模块被存储在存储器中并被配置为由处理器执行,一个或多个计算机程序模块包括用于实现上述任一实施例的算子处理方法的指令,因此在由处理器执行时用于实现上述任一实施例的算子处理方法。该电子设备在生成融合算子的过程中可以将网络结构中的共性抽象出来,没有包含特

定情况的信息,可以适用于各种场景,适用范围广,泛化能力强。

[0135] 图8为本公开一些实施例提供的一种电子设备的示意框图。如图8所示,该电子设备800包括处理器810和存储器820。存储器820存储有非暂时性计算机可读指令(例如一个或多个计算机程序模块)。处理器810用于运行非暂时性计算机可读指令,非暂时性计算机可读指令被处理器810运行时执行上文所述的算子处理方法中的一个或多个步骤。存储器820和处理器810可以通过总线系统和/或其它形式的连接机构(未示出)互连。

[0136] 例如,处理器810和存储器820可以设置在服务器端(或云端)。

[0137] 例如,处理器801可以控制电子设备800中的其它组件以执行期望的功能。处理器810可以是中央处理单元(CPU)、图形处理单元(Graphics Processing Unit,GPU)或者具有数据处理能力和/或程序执行能力的其它形式的处理单元。例如,中央处理单元(CPU)可以为X86或ARM架构等。处理器810可以为通用处理器或专用处理器,可以控制电子设备800中的其它组件以执行期望的功能。

[0138] 例如,存储器820可以包括一个或多个计算机程序产品的任意组合,计算机程序产品可以包括各种形式的计算机可读存储介质,例如易失性存储器和/或非易失性存储器。易失性存储器例如可以包括随机存取存储器(RAM)和/或高速缓冲存储器(cache)等。非易失性存储器例如可以包括只读存储器(ROM)、硬盘、可擦除可编程只读存储器(EPROM)、便携式紧致盘只读存储器(CD-ROM)、USB存储器、闪存等。在计算机可读存储介质上可以存储一个或多个计算机程序模块,处理器810可以运行一个或多个计算机程序模块,以实现电子设备800的各种功能。在计算机可读存储介质中还可以存储各种应用程序和各种数据以及应用程序使用和/或产生的各种数据等。

[0139] 例如,在一些实施例中,电子设备800可以为手机、平板电脑、电子纸、电视机、显示器、笔记本电脑、数码相框、导航仪、可穿戴电子设备、智能家居设备等。

[0140] 例如,电子设备800可以包括显示面板,显示面板可以用于分割图像等。例如,显示面板可以为矩形面板、圆形面板、椭圆形面板或多边形面板等。另外,显示面板不仅可以为平面面板,也可以为曲面面板,甚至球面面板。

[0141] 例如,电子设备800可以具备触控功能,即电子设备800可以为触控装置。

[0142] 需要说明的是,本公开的实施例中,电子设备800的具体功能和技术效果可以参考上文中关于算子处理方法的描述,此处不再赘述。

[0143] 图9为本公开一些实施例提供的另一种电子设备的示意框图。该电子设备900例如适于用来实施本公开实施例提供的算子处理方法。电子设备900可以是终端设备等。需要注意的是,图9示出的电子设备900仅仅是一个示例,其不会对本公开实施例的功能和使用范围带来任何限制。

[0144] 如图9所示,电子设备900可以包括处理装置(例如中央处理器、图形处理器等)910,其可以根据存储在只读存储器(ROM)920中的程序或者从存储装置980加载到随机访问存储器(RAM)930中的程序而执行各种适当的动作和处理。在RAM 930中,还存储有电子设备900操作所需的各种程序和数据。处理装置910、ROM 920以及RAM930通过总线940彼此相连。输入/输出(I/O)接口950也连接至总线940。

[0145] 通常,以下装置可以连接至I/O接口950:包括例如触摸屏、触摸板、键盘、鼠标、摄像头、麦克风、加速度计、陀螺仪等的输入装置960;包括例如液晶显示器(LCD)、扬声器、振

动器等的输出装置970;包括例如磁带、硬盘等的存储装置980;以及通信装置990。通信装置990可以允许电子设备900与其他电子设备进行无线或有线通信以交换数据。虽然图9示出了具有各种装置的电子设备900,但应理解的是,并不要求实施或具备所有示出的装置,电子设备900可以替代地实施或具备更多或更少的装置。

[0146] 例如,根据本公开的实施例,上述算子处理方法可以被实现为计算机软件程序。例如,本公开的实施例包括一种计算机程序产品,其包括承载在非暂态计算机可读介质上的计算机程序,该计算机程序包括用于执行上述算子处理方法的程序代码。在这样的实施例中,该计算机程序可以通过通信装置990从网络上被下载和安装,或者从存储装置980安装,或者从ROM 920安装。在该计算机程序被处理装置910执行时,可以实现本公开实施例提供的算子处理方法中限定的功能。

[0147] 本公开的至少一个实施例还提供了一种计算机可读存储介质,该计算机可读存储介质存储有非暂时性计算机可读指令,当非暂时性计算机可读指令由计算机执行时可以实现上述的算子处理方法。利用该计算机可读存储介质,可以在生成融合算子的过程中将网络结构中的共性抽象出来,没有包含特定情况的信息,可以适用于各种场景,适用范围广,泛化能力强。

[0148] 图10为本公开一些实施例提供的一种存储介质的示意图。如图10所示,存储介质1000存储有非暂时性计算机可读指令1010。例如,当非暂时性计算机可读指令1010由计算机执行时执行根据上文所述的算子处理方法中的一个或多个步骤。

[0149] 例如,该存储介质1000可以应用于上述电子设备800中。例如,存储介质1000可以为图8所示的电子设备800中的存储器820。例如,关于存储介质1000的相关说明可以参考图8所示的电子设备800中的存储器820的相应描述,此处不再赘述。

[0150] 以上描述仅为本公开的较佳实施例以及对所运用技术原理的说明。本领域技术人员应当理解,本公开中所涉及的公开范围,并不限于上述技术特征的特定组合而成的技术方案,同时也应涵盖在不脱离上述公开构思的情况下,由上述技术特征或其等同特征进行任意组合而形成的其它技术方案。例如上述特征与本公开中公开的(但不限于)具有类似功能的技术特征进行互相替换而形成的技术方案。

[0151] 此外,虽然采用特定次序描绘了各操作,但是这不应当理解为要求这些操作以所示出的特定次序或以顺序次序执行来执行。在一定环境下,多任务和并行处理可能是有利的。同样地,虽然在上面论述中包含了若干具体实现细节,但是这些不应当被解释为对本公开的范围的限制。在单独的实施例的上下文中描述的某些特征还可以组合地实现在单个实施例中。相反地,在单个实施例的上下文中描述的各种特征也可以单独地或以任何合适的子组合的方式实现在多个实施例中。

[0152] 有以下几点需要说明:

[0153] (1)本公开实施例附图只涉及到本公开实施例涉及到的结构,其他结构可参考通常设计。

[0154] (2)在不冲突的情况下,本公开的实施例及实施例中的特征可以相互组合以得到新的实施例。

[0155] 以上所述,仅为本公开的具体实施方式,但本公开的保护范围并不局限于此,本公开的保护范围应以所述权利要求的保护范围为准。

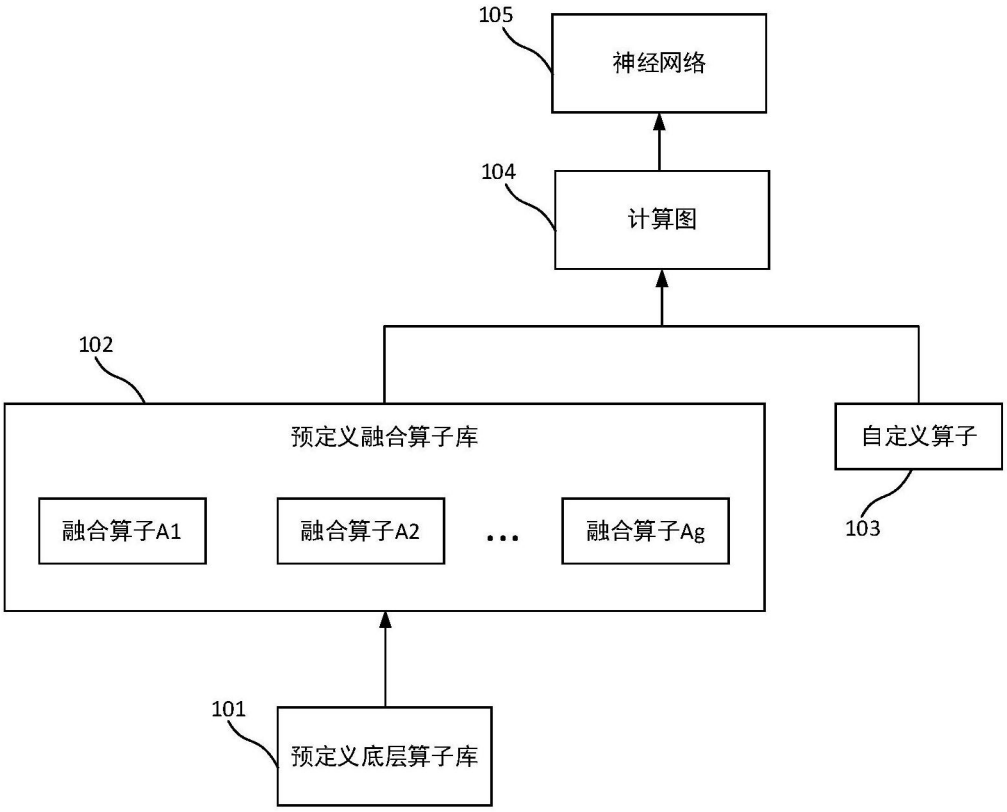

105
神经网络
104
计算图
102
预定义融合算子库
融合算子A1
融合算子A2
...
融合算子Ag
自定义算子
103
101
预定义底层算子库

图1

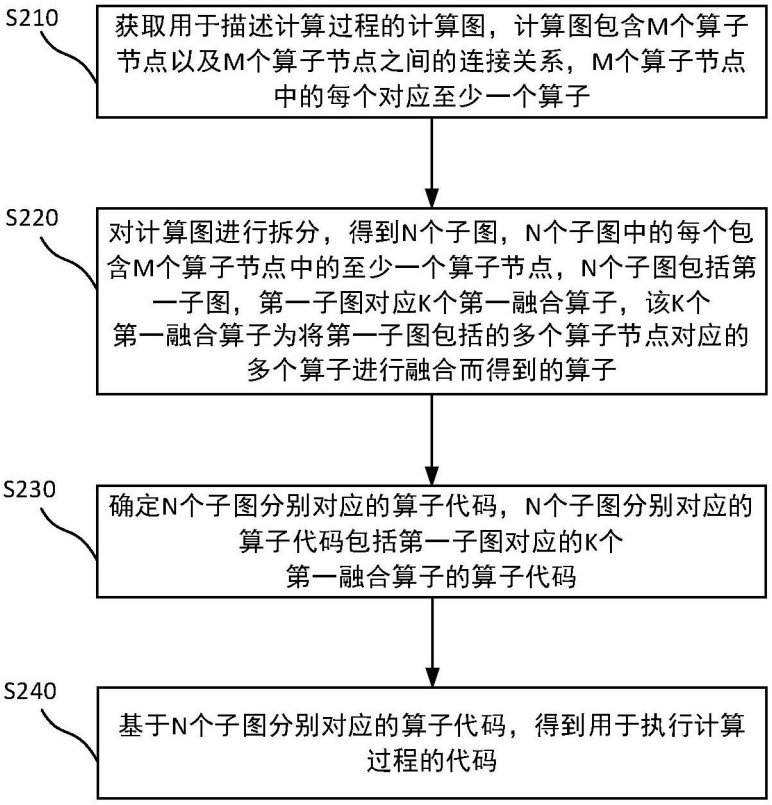

S210
获取用于描述计算过程的计算图，计算图包含M个算子节点以及M个算子节点之间的连接关系，M个算子节点中的每个对应至少一个算子
S220
对计算图进行拆分，得到N个子图，N个子图中的每个包含M个算子节点中的至少一个算子节点，N个子图包括第一子图，第一子图对应K个第一融合算子，该K个第一融合算子为将第一子图包括的多个算子节点对应的多个算子进行融合而得到的算子
S230
确定N个子图分别对应的算子代码，N个子图分别对应的算子代码包括第一子图对应的K个第一融合算子的算子代码
S240
基于N个子图分别对应的算子代码，得到用于执行计算过程的代码

图2

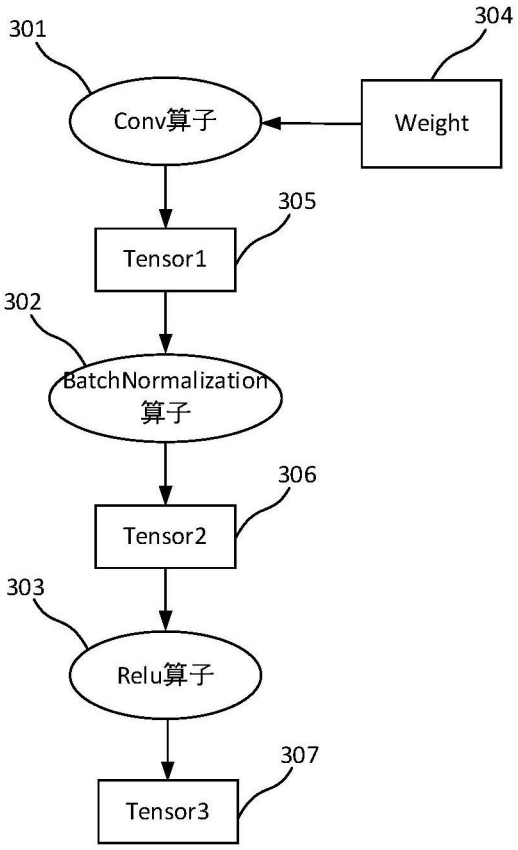

301
304
Conv算子
Weight
305
Tensor1
302
BatchNormalization
算子
306
Tensor2
303
Relu算子
307
Tensor3

图3

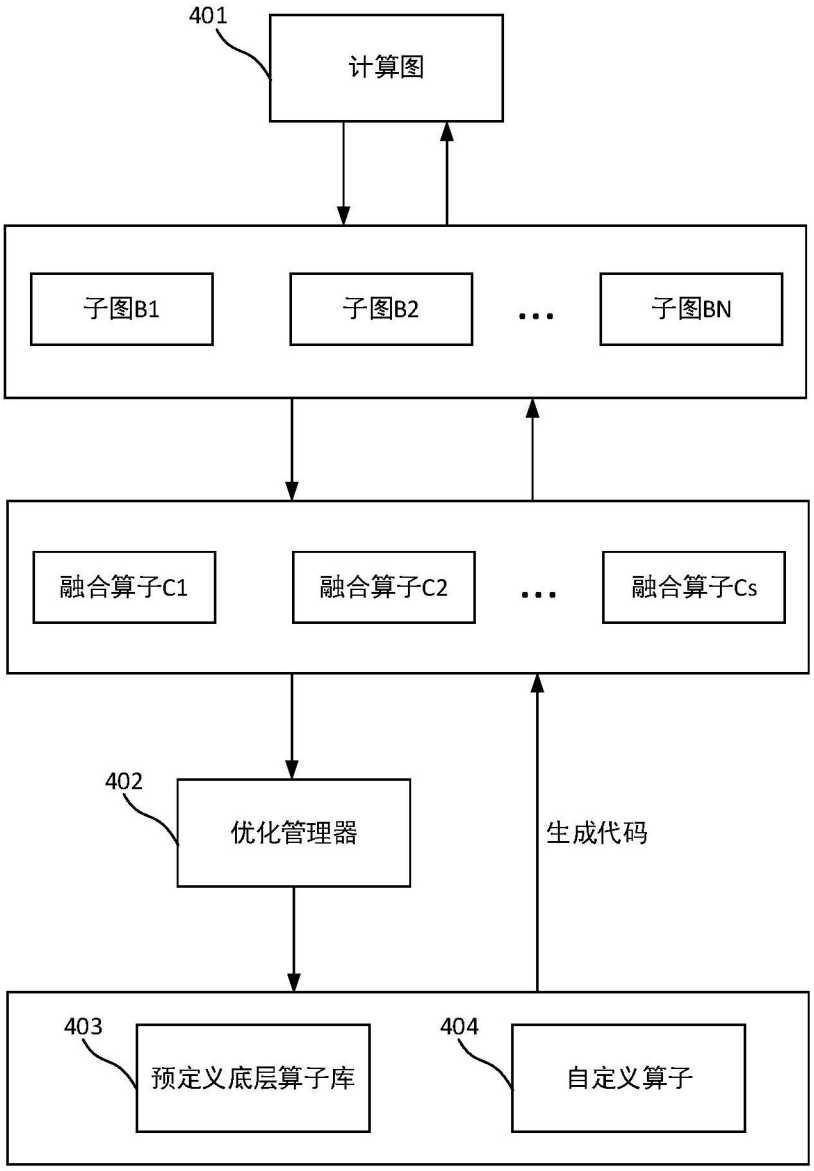

401
计算图
子图B1
子图B2
...
子图BN
融合算子C1
融合算子C2
...
融合算子Cs
402
优化管理器
生成代码
403
预定义底层算子库
404
自定义算子

图4

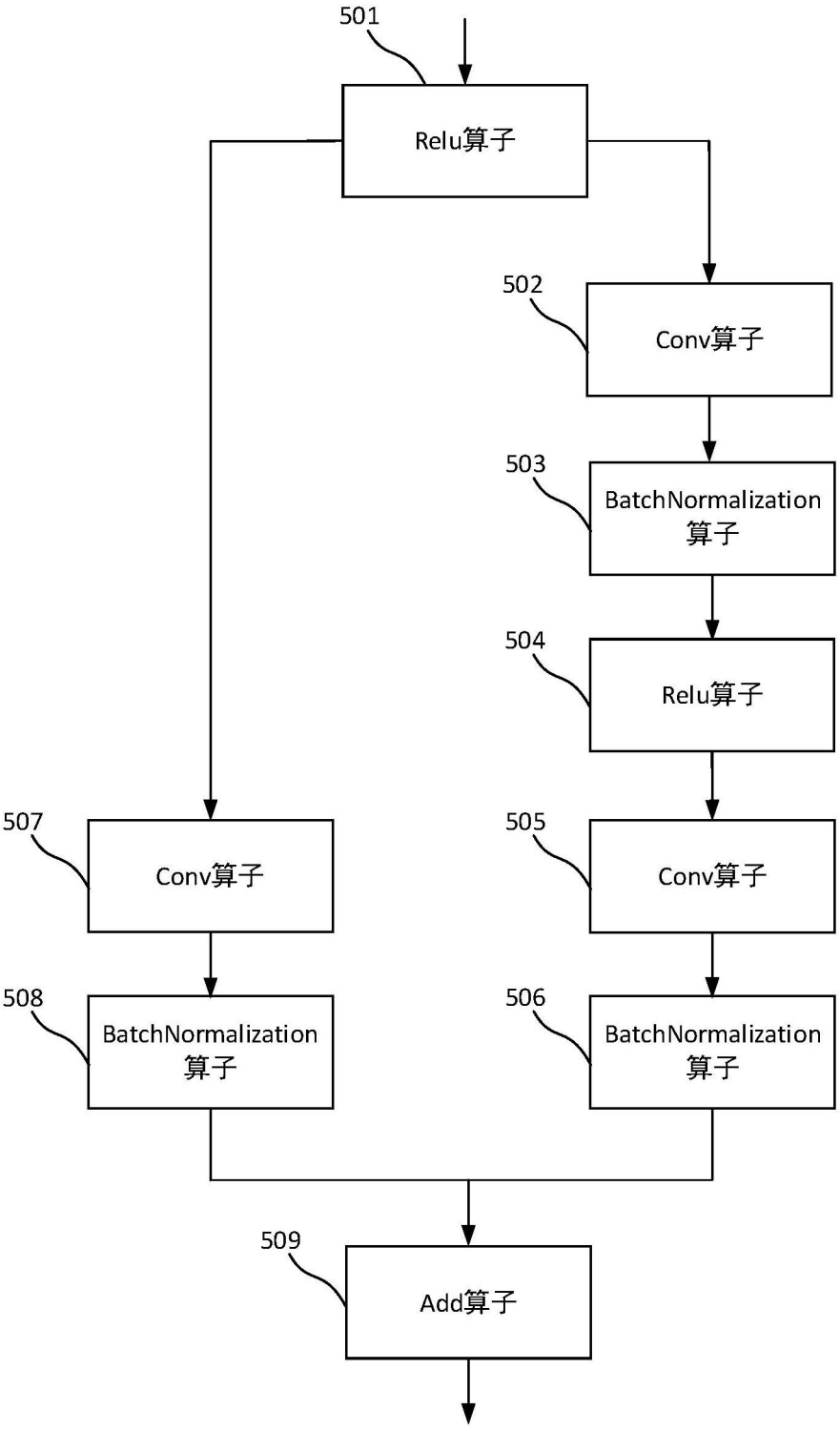

501
Relu算子
502
Conv算子
503
BatchNormalization
算子
504
Relu算子
505
Conv算子
506
BatchNormalization
算子
507
Conv算子
508
BatchNormalization
算子
509
Add算子

图5A

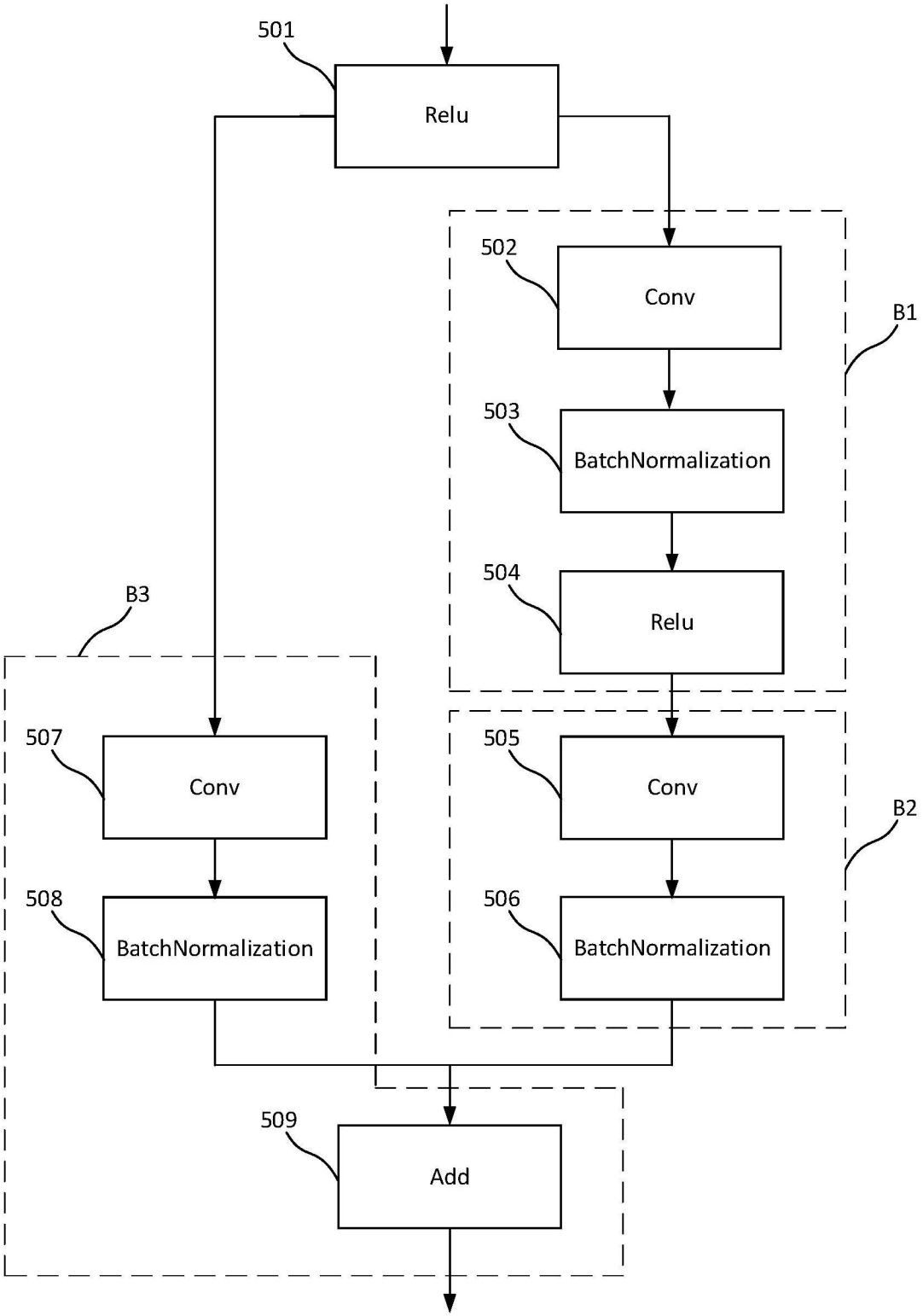

501
Relu
502
Conv
B1
503
BatchNormalization
504
Relu
B3
507
Conv
505
Conv
B2
508
BatchNormalization
506
BatchNormalization
509
Add

图5B


Weight
CBR
（Conv+BatchNormalization+Relu）
Tensor3

图6

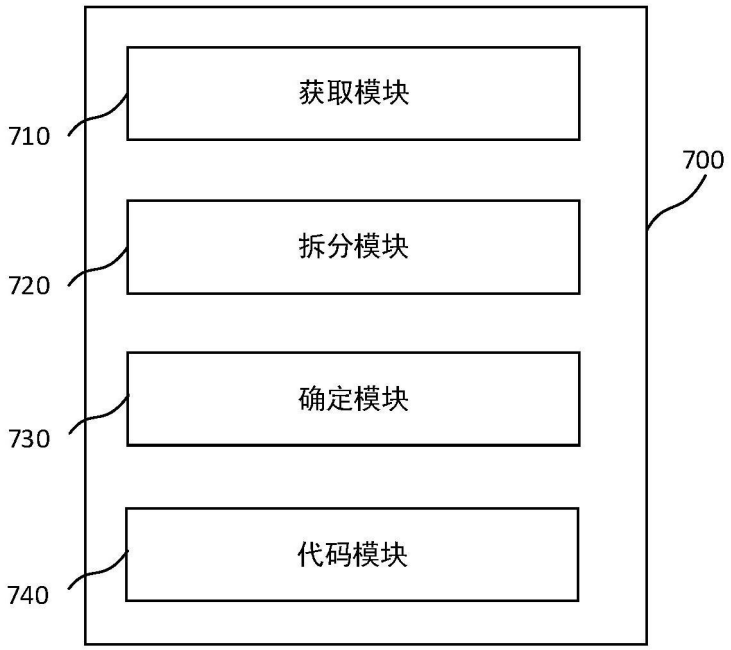

获取模块
710
700
拆分模块
720
确定模块
730
代码模块
740

图7

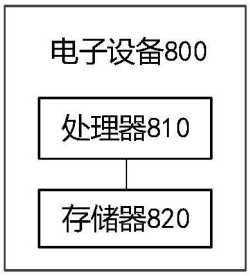

电子设备800
处理器810
存储器820

图8

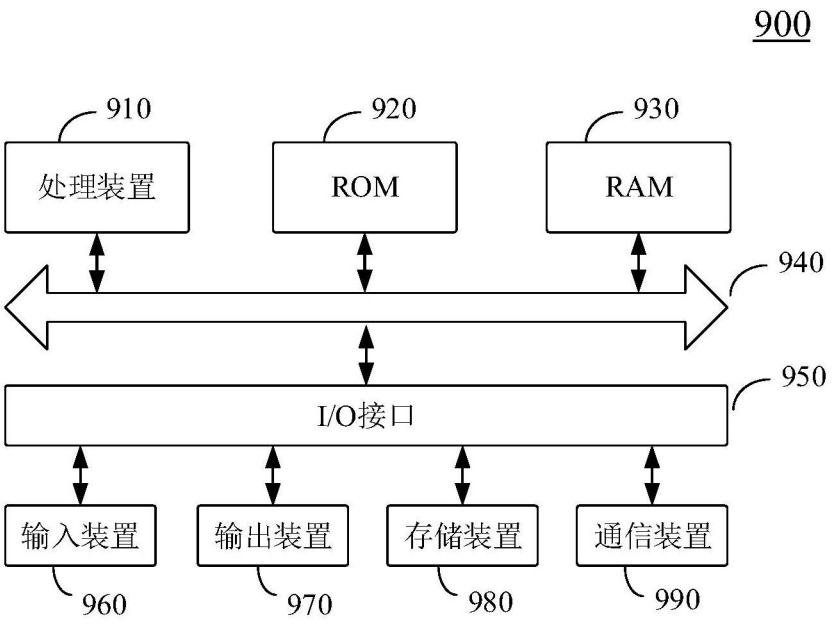

900
910
处理装置
920
ROM
930
RAM
940
950
I/O接口
输入装置
960
输出装置
970
存储装置
980
通信装置
990

图9

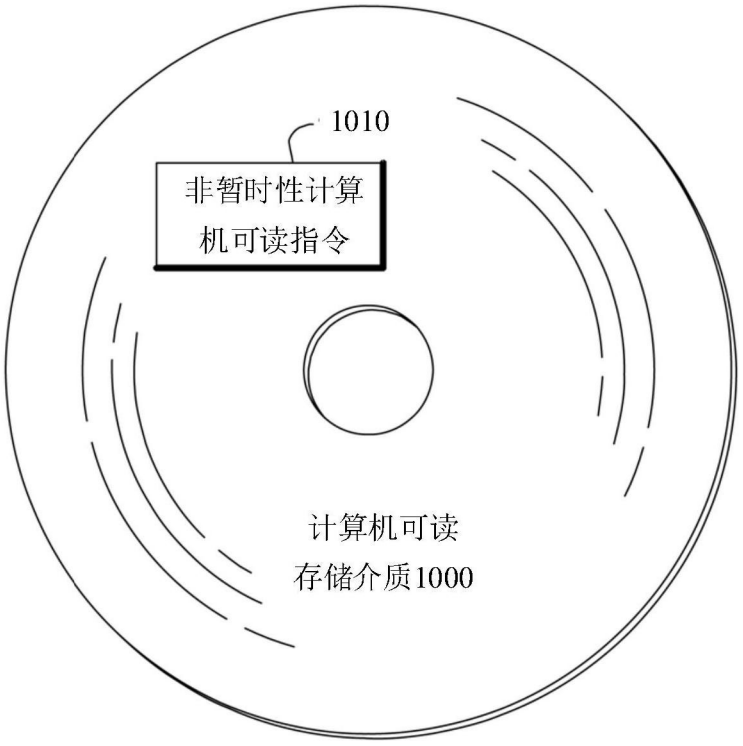

1010
非暂时性计算
机可读指令
计算机可读
存储介质1000

图10