(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2023 年 5 月 19 日 (19.05.2023)

WIPO | PCT

(10) 国际公布号
WO 2023/082575 A1

(51) 国际专利分类号:
G06N 3/04 (2006.01) *G06F 12/02* (2006.01)

(21) 国际申请号: PCT/CN2022/092481

(22) 国际申请日: 2022 年 5 月 12 日 (12.05.2022)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202210447287.7 2022年4月27日 (27.04.2022) CN

(71) 申请人: 之江实验室 (**ZHEJIANG LAB**) [CN/CN]; 中国浙江省杭州市余杭区中泰街道科创大道之江实验室南湖总部, Zhejiang 311121 (CN)。

(72) 发明人: 王宏升 (**WANG, Hongsheng**); 中国浙江省杭州市余杭区中泰街道科创大道之江实验室南湖总部, Zhejiang 311121 (CN)。 谭博文 (**TAN, Bowen**); 中国浙江省杭州市余杭区中泰街道科创大道之江实验室南湖总部, Zhejiang 311121 (CN)。 鲍虎军 (**BAO, Hujun**); 中国浙江省杭州市余杭区中泰街道科创大道之江实验室南湖总部, Zhejiang 311121 (CN)。 陈光 (**CHEN, Guang**); 中国浙江省杭州市余杭区中泰街道科创大道之江实验室南湖总部, Zhejiang 311121 (CN)。

(74) 代理人: 北京志霖恒远知识产权代理有限公司 (**BEIJING ZHILIN HENGYUAN INTELLECTUAL PROPERTY AGENCY CO., LTD.**); 中国北京市东城区北三环东路36号环球贸易中心C座2001-2007室, Beijing 100013 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE,

(54) **Title:** GRAPH EXECUTION PIPELINE PARALLELISM METHOD AND APPARATUS FOR NEURAL NETWORK MODEL COMPUTATION

(54) 发明名称: 一种面向神经网络模型计算的图执行流水并行方法和装置

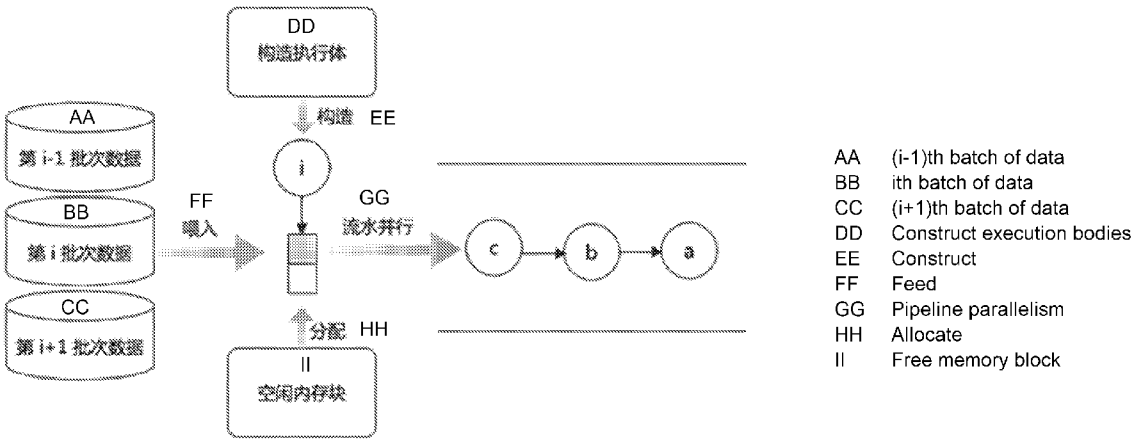

图 1

(57) **Abstract:** Provided in the present invention are a graph execution pipeline parallelism method and apparatus for neural network model computation, and provided are a graph execution pipeline parallelism method and apparatus for neural network model computation in a deep learning training system. The method comprises a graph execution flow during a process for neural network model computation, and a process during which functional modules work cooperatively. According to the graph execution pipeline parallelism method for neural network model computation, graph execution bodies on a local machine are created according to a physical computational graph generated by means of compiling a deep learning framework, and a scheme for allocating a plurality of free memory blocks to each graph execution body is designed, such that the entire computational graph simultaneously participates in deep learning training tasks of different batches of data by means of pipeline parallelism, thereby fully increasing the utilization rate of a memory and the parallelism rate of data.

WO 2023/082575 A1

[见续页]

SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) **指定国**(除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

— 在修改权利要求的期限届满之前进行，在收到该修改后将重新公布(细则48.2(h))。

— 根据申请人的请求，在条约第21条(2)(a)所规定的期限届满之前进行。

(57) **摘要**：本发明提出了一种面向神经网络模型计算的图执行流水并行方法和装置，提供了一种深度学习训练系统中面向神经网络模型计算的图执行流水并行方法和装置。包括面向神经网络模型计算过程中的图执行流程和各功能模块协同工作的过程。所述面向神经网络模型计算的图执行流水并行方法是根据深度学习框架编译生成的物理计算图创建本机上的图执行体，通过设计为每个图执行体分配多个空闲内存块的方案，实现了整张计算图以流水并行的方式同时参与到不同批次数据的深度学习训练任务中，充分提高了内存的使用率和数据的并行速率。

一种面向神经网络模型计算的图执行流水并行方法和装置

本发明要求于2022年4月27日向中国国家知识产权局提交的申请号为202210447287.7、发明名称为“一种面向神经网络模型计算的图执行流水并行方法和装置”中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

[0001] 本发明涉及深度学习技术领域，特别涉及一种面向神经网络模型计算的图执行流水并行方法和装置。

背景技术

[0002] 随着人工智能产业化应用的快速发展，实际应用场景对大模型的需求变得越来越紧迫，机器学习工作负载的结构越来越趋于复杂的大模型，导致用于大模型计算的图的执行成本非常高。已有的用于神经网络模型计算的图执行方法多数基于同步的方法，导致整个图执行系统的资源利用率不高，限制了分布式系统的加速比和吞吐率。

[0003] 为解决以上问题，本发明提供的用于神经网络模型计算的图执行流水并行方法隔离了各批次的训练数据与不同子图，每批次的训练数据按照 1F1B 前向后向的方式依次流过前向计算图和反向计算图。本发明可使得每个设备进程上都会有一个批次的数据正在被处理，使所有设备进程保持忙碌，而不会出现管道暂停，整个流水线是比较均衡的。同时能确保以固定周期执行每个子图上的参数更新，也有助于防止同时处理过多小批量并确保模型收敛。

发明内容

[0004] 本发明的目的在于提供一种面向神经网络模型计算的图执行流水并行方法和装置，以克服现有技术中的不足。

[0005] 为实现上述目的，本发明提供如下技术方案：

本申请公开了一种面向神经网络模型计算的图执行流水并行方法，所述神经网络模型中设有若干个执行体，所述执行体共有 2*N 个，所述 N 为正整数，所述执行体设有若干个内存块；所述方法具体包括如下步骤：

S1、将训练数据分成若干个批次子数据；

S2、若干个批次子数据依次输入神经网络模型中，当第 i 批次的子数据输入后，第 n 执行体对第 i 批次的子数据执行自身核函数计算，并将执行结果写入第 n 执行体空闲的内存块中；接着输入第 i+1 批次的子数据；所述 i，n 均为正整数；

S3、当第 i+1 批次的子数据输入后，第 n 执行体对第 i+1 批次的子数据执行 S2 操作的同时，将第 i 批次所在的内存块的地址发送至第 n+1 执行体；第 n+1 执行体解析第 i 批次所在的内存块，得到第 n 执行体对第 i 批次的子数据的执行结果，并将第 n 执行体的执行结果作为第 n+1 执行体的输入数据，执行自身核函数计算，并将执行结果写入第 n+1 执行体空闲的内存块中；接着输入第 i+2 批次的子数据；

S4、当第 i+2 批次的子数据输入后，第 n 执行体对第 i+2 批次的子数据执行 S2 操作，第 n 执行体和第 n+1 执行体对第 i+1 批次的子数据执行 S3 的操作；同时第 n+1 执行体将第 i 批次所在的内存块的地址发送至第 n+2 执行体，第 n+2 执行体解析第 i 批次所在的内存块，得到第 n+1 执行体对第 i 批次的子数据的执行结果，并将第 n+1 执行体的执行结果作为第 n+2 执行体的输入数据，执行自身核函数计算，并将执行结果写入第 n+2 执行体空闲的内存块中；

S5、第 n 执行体回收发送给第 n+1 执行体的内存块；

S6、最后一个执行体执行自身核函数计算，并将执行结果写入最后一个执行体的内存块，执行完毕即刻自行回收内存块。

[0006] 作为优选，执行体在执行自身核函数计算前，执行体会检查自身是否存在空闲的内存块；若存在，则对第 i 批次的子数据执行自身核函数计算；若不存在，则令第 i 批次等待存在空闲的内存块。

[0007] 作为优选，对于第 N*n+1 批次的子数据，执行体在执行自身核函数计算前，会检查第 N*（n-1）+1 批次的子数据所在的执行体是否执行完毕，所述 n 为正整数。

[0008] 作为优选，步骤 S5 具体包括如下操作：

S51、第 n+1 执行体通知第 n 执行体已消费完发送给第 n+1 执行体的内存块；

S52、第 n 执行体回收发送给第 n+1 执行体的内存块，并将其标记为空闲。

[0009] 作为优选，还包括执行体的构造，所述执行体的构造具体包括如下子步骤：

S01、创建算子核函数的任务队列：将当前算子核函数的计算任务依次加入当前核函数任务队列；

S02、创建执行体的线程：所述执行体的线程负责从所述核函数任务队列中依次获取当前待处理任务，并提交给线程池；

S03、创建核函数的执行体：根据当前核函数任务和当前线程的上下文信息创建用于算子核函数计算的执行体；并使用执行体运行任务队列中的核函数任务；

S04、创建事件召回队列：将任务执行体处理完的任务添加到事件召回队列中；

S05、创建事件召回队列的线程：所述事件召回队列的线程负责将事件召回队列中已处理的任

务依次取出并返回。

[0010] 本发明还公开了一种面向神经网络模型计算的图执行装置，包括执行体构造模块和执行体流水并行工作模块，所述执行体构造模块用于执行体的构造，所述执行体流水并行工作模块用于执行上述一种面向神经网络模型计算的图执行流水并行方法。

[0011] 本发明还公开了一种面向神经网络模型计算的图执行装置，包括存储器和一个或多个处理器，所述存储器中存储有可执行代码，所述一个或多个处理器执行所述可执行代码时，用于上述一种面向神经网络模型计算的图执行流水并行方法。

[0012] 本发明还公开了一种计算机可读存储介质，其上存储有程序，该程序被处理器执行时，实现上述的一种面向神经网络模型计算的图执行流水并行方法。

[0013] 本发明的有益效果：

提供了一种面向神经网络模型计算的图执行流水并行方法和装置，根据深度学习框架编译生成的物理计算图创建本机上的图执行体，通过设计为每个图执行体分配多个空闲内存块的方案，实现了整张计算图以流水并行的方式同时参与到不同批次数据的深度学习训练任务中。本发明公开的基于多个空闲张量存储块的图执行体并行执行方法比已有方法更加容易实现大模型的分布式训练。在大规模深度神经网络的分布式应用场景下，本发明对用户的使用门槛较低，并且能够使模型学习到大量分批次流入神经网络的数据的内在关联，从而获得对应场景中的“智能”感知与判断能力。本发明为深度学习相关的算法工程师提供了一套简洁易用的神经网络模型的运行装置，使之能方便地训练深度学习模型。

[0014] 本发明的特征及优点将通过实施例结合附图进行详细说明。

附图说明

[0015] 图 1 面向神经网络模型计算的图执行流水并行方法的架构图；

图 2 创建管理任务执行体线程模块的流程图；

图 3 任务执行体流水并行工作模块基本动作；

图 4 执行体流水并行的执行过程；

图 5 是本发明一种面向神经网络模型计算的图执行装置的结构示意图。

具体实施方式

[0016] 为使本发明的目的、技术方案和优点更加清楚明了，下面通过附图及实施例，对本发明进行进一步详细说明。但是应该理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限制本发明的范围。此外，在以下说明中，省略了对公知结构和技术的描述，以避免不必要地混淆本发明的概念。

[0017] 如图 1 所示，所述一种面向神经网络模型计算的图执行流水并行方法的架构图。如图将训练数据分批次喂入神经网络模型，根据深度学习框架编译生成的物理计算图创建本机上的图执行体，为每个图执行体分配多个空闲内存块，使得整张计算图以流水并行的方式同时参与深度学习训练任务中，具体操作如下：

S1、将训练数据分成若干个批次子数据；

S2、若干个批次子数据依次输入神经网络模型中，当第 i 批次的子数据输入后，第 n 执行体对第 i 批次的子数据执行自身核函数计算，并将执行结果写入第 n 执行体空闲的内存块中；接着输入第 i+1 批次的子数据；所述 i，n 均为正整数；

S3、当第 i+1 批次的子数据输入后，第 n 执行体对第 i+1 批次的子数据执行 S2 操作的同时，将第 i 批次所在的内存块的地址发送至第 n+1 执行体；第 n+1 执行体解析第 i 批次所在的内存块，得到第 n 执行体对第 i 批次的子数据的执行结果，并将第 n 执行体的执行结果作为第 n+1 执行体的输入数据，执行自身核函数计算，并将执行结果写入第 n+1 执行体空闲的内存块中；接着输入第 i+2 批次的子数据；

S4、当第 i+2 批次的子数据输入后，第 n 执行体对第 i+2 批次的子数据执行 S2 操作，第 n 执行体和第 n+1 执行体对第 i+1 批次的子数据执行 S3 的操作；同时第 n+1 执行体将第 i 批次所在的内存块的地址发送至第 n+2 执行体，第 n+2 执行体解析第 i 批次所在的内存块，得到第 n+1 执行体对第 i 批次的子数据的执行结果，并将第 n+1 执行体的执行结果作为第 n+2 执行体的输入数据，执行自身核函数计算，并将执行结果写入第 n+2 执行体空闲的内存块中；

S5、第 n 执行体回收发送给第 n+1 执行体的内存块。

[0018] S6、最后一个执行体执行自身核函数计算，并将执行结果写入最后一个执行体的内存块，执行完毕即刻自行回收内存块。

[0019] 在一种可行的实施例中，执行体在执行自身核函数计算前，执行体会检查自身是否存在空闲的内存块；若存在，则对第 i 批次的子数据执行自身核函数计算；若不存在，则令第 i 批次等待存在空闲的内存块。

[0020] 在一种可行的实施例中，对于第 N*n+1 批次的子数据，执行体在执行自身核函数计算前，会检查第 N*（n-1）+1 批次的子数据所在的执行体是否执行完毕，所述 n 为正整数。

[0021] 在一种可行的实施例中，步骤 S5 具体包括如下操作：

S51、第 n+1 执行体通知第 n 执行体已消费完发送给第 n+1 执行体的内存块；

S52、第 n 执行体回收发送给第 n+1 执行体的内存块，并将其标记为空闲。

[0022] 在一种可行的实施例中，还包括执行体的构造，所述执行体的构造具体包括如下子步

骤：

S01、创建算子核函数的任务队列：将当前算子核函数的计算任务依次加入当前核函数任务队列；

S02、创建执行体的线程：所述执行体的线程负责从所述核函数任务队列中依次获取当前待处理任务，并提交给线程池；

S03、创建核函数的执行体：根据当前核函数任务和当前线程的上下文信息创建用于算子核函数计算的执行体；并使用执行体运行任务队列中的核函数任务；

S04、创建事件召回队列：将任务执行体处理完的任务添加到事件召回队列中；

S05、创建事件召回队列的线程：所述事件召回队列的线程负责将事件召回队列中已处理的任务依次取出并返回。

[0023] 本发明一种面向神经网络模型计算的图执行装置，包括执行体构造模块和执行体流水并行工作模块，

参阅图 2，所述执行体构造模块包括如下基本动作：

创建算子核函数的任务队列：将当前算子核函数的计算任务依次加入当前核函数任务队列；

创建任务执行体的线程：创建任务执行体的线程。所述任务执行体的线程负责从所述任务队列中依次获取当前待处理任务；当服务器收到一个请求时，将请求提交给线程池，并继续等待其他请求。如果池中有一个可用的线程，它就会被唤醒，请求就会立即得到服务。如果池子里没有可用的线程，任务就会被排队，直到有一个空闲的线程。一旦一个线程完成了它的服务，它就会返回到池子里，等待更多的工作。当提交给线程池的任务可以异步执行时，线程池就能很好地工作。

创建核函数的任务执行体：根据当前核函数任务和当前线程的上下文信息创建用于算子核函数计算的任务执行体。并使用所述任务执行体运行任务队列中的核函数任务。

[0024] 创建事件召回队列：当处理完上述任务队列中的全部任务执行体时，创建事件召回队列，依次将上述任务执行体处理完的任务添加到事件召回队列中；

创建事件召回队列的线程： 创建事件召回队列的线程。所述事件召回队列的线程负责将事件召回队列中已处理的任务依次取出并返回。

[0025] 参阅图 3，所述执行体流水并行工作模块包括如下基本动作：执行体输入数据、当前执行体向下游执行体发送消息、下游执行体准备待消费的张量数据、当前执行体向上游执行体发送消息、上游执行体回收已被消费完的张量数据、尾执行体自行回收计算数据。

[0026] 执行体输入数据：在 t 时刻，对于第 i 批次数据，执行体输入第 i 批次数据，加载其内

部的算子核函数计算任务，执行核函数计算，生成核函数计算任务的输出张量数据，将执行结果写入内存空闲块；

当前执行体向下游执行体发送消息：在 t 时刻，对于第 i 批次数据，将当前执行体生产所得的张量数据存储到空的存储单元中，再将存储单元的地址和当前执行体对应的下游执行体的身份标识号打包成消息，之后发送消息至目标执行体，所述目标执行体就是当前执行体对应的下游执行体；

下游执行体准备待消费的张量数据：在 t 时刻，对于第 i 批次数据，下游执行体收到消息，从消息中解析出上述当前执行体生产的张量数据，所述张量数据将作为下游执行体运行其算子核函数时的输入张量，并检查自己生产的内存块是否有空闲内存块可用，如发现有可用的空闲内存块，则下游执行体执行对应算子的核函数计算任务，读取空闲内存块，下游执行体将执行生成的输出张量结果写入内存块；

当前执行体向上游执行体发送消息：在 t 时刻，对于第 i 批次数据，执行体给上游的生产者执行体发消息通知上游的生产者执行体执行体消费完了上游的生产者执行体的内存块，上游执行体可以回收其输出张量数据的存储单元；

上游执行体回收已被消费完的数据：在 t 时刻，对于第 i 批次数据，上游执行体收到下游执行体发送的回收消息，就开始检查内存块是否都已被所有的消费者执行体消费完毕，如是，则将内存块回收，标记为空闲块；

尾执行体自行回收计算数据：在 t 时刻，对于第 i 批次数据，尾执行体执行对应算子的核函数计算任务，写入自己内存块空闲，执行体 A 执行完毕即刻自行回收内存块。

[0027] 本发明一种面向神经网络模型计算的图执行装置的实施例可以应用在任意具备数据处理能力的设备上，该任意具备数据处理能力的设备可以为诸如计算机等设备或装置。装置实施例可以通过软件实现，也可以通过硬件或者软硬件结合的方式实现。以软件实现为例，作为一个逻辑意义上的装置，是通过其所在任意具备数据处理能力的设备的处理器将非易失性存储器中对应的计算机程序指令读取到内存中运行形成的。从硬件层面而言，如图 5 所示，为本发明一种面向神经网络模型计算的图执行装置所在任意具备数据处理能力的设备的一种硬件结构图，除了图 5 所示的处理器、内存、网络接口、以及非易失性存储器之外，实施例中装置所在的任意具备数据处理能力的设备通常根据该任意具备数据处理能力的设备的实际功能，还可以包括其他硬件，对此不再赘述。上述装置中各个单元的功能和作用的实现过程具体详见上述方法中对应步骤的实现过程，在此不再赘述。

[0028] 对于装置实施例而言，由于其基本对应于方法实施例，所以相关之处参见方法实施例

的部分说明即可。以上所描述的装置实施例仅仅是示意性的，其中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模块来实现本发明方案的目的。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。

[0029] 本发明实施例还提供一种计算机可读存储介质，其上存储有程序，该程序被处理器执行时，实现上述实施例中的一种面向神经网络模型计算的图执行装置。

[0030] 所述计算机可读存储介质可以是前述任一实施例所述的任意具备数据处理能力的设备的内部存储单元，例如硬盘或内存。所述计算机可读存储介质也可以是任意具备数据处理能力的设备的外部存储设备，例如所述设备上配备的插接式硬盘、智能存储卡(Smart Media Card，SMC)、SD 卡、闪存卡(Flash Card）等。进一步的，所述计算机可读存储介质还可以既包括任意具备数据处理能力的设备的内部存储单元也包括外部存储设备。所述计算机可读存储介质用于存储所述计算机程序以及所述任意具备数据处理能力的设备所需的其他程序和数据，还可以用于暂时地存储已经输出或者将要输出的数据。

[0031] 实施例：

参阅图 4，构建物理计算图由正向算子 x->正向算子 y->正向算子 z 及反向算子 Z->反向算子 Y->反向算子 X 组成，分别根据各个算子创建运行自身核函数的执行体，对应构成执行体 a->执行体 b->执行体 c->执行体 C->执行体 B->执行体 A 的执行计算图；启动运行时的执行体，并行运行整张计算图。

[0032] T1 时刻：

输入第 1 批次数据，执行体 a 输入数据：执行体 a 运行正向算子 x 的核函数，将运行结果的输出张量写入内存空闲块 r11。

[0033] 执行体 b，执行体 c，执行体 C，执行体 B，执行体 A 由于没有可读的输入张量数据，所以执行体 b，c，C，B，A 处在等待状态。

[0034] 步骤 3: 并行运行整张计算图。

[0035] T2 时刻：

对于第 2 批次数据，执行体 a 输入数据：执行体 a 还会检查自己是否有空闲块可写，发现有，T2 时刻执行体 a 也在执行第 2 批次输入数据，将执行结果写入内存空闲块 r12。

[0036] 同时对于第 1 批次数据，当前执行体 a 向下游执行体 b 发送消息、下游执行体 b 准备待消费的张量数据：执行体 a 给执行体 b 发消息通知执行体 b 读取执行体 a 产出的内存块 r11，

执行体 b 收到消息，并检查自己生产的内存块 b 是否有空闲内存块可用，发现有可用的空闲内存块 r21，于是 T2 时刻执行体 b 执行前向算子 b 的核函数计算任务，读取内存块 r11，执行体 b 将执行生成的输出张量结果写入内存块 r21。

[0037] 于是执行体 a 和执行体 b 就开始并行工作。执行体 c，C，B，A 由于没有数据可读，仍在等待。

[0038] T3 时刻：

对于第 3 批次数据，执行体 a 输入数据：执行体 a 还会检查自己是否有空闲块可写，发现有，则执行体 a 也在执行第 3 批次输入数据，将执行结果写入内存空闲块 r13。

[0039] 同时对于第 1 批次数据，当前执行体 b 向下游执行体 c 发送消息、下游执行体 c 准备待消费的张量数据、当前执行体 b 向上游执行体 a 发送消息、上游执行体 a 回收已被消费完的张量数据：执行体 b 生产出了内存块 r21，于是给下游的消费者执行体 c 发消息通知执行体 c 读取执行体 b 产出的内存块 r21，执行体 c 收到内存块 r21，发现自己有内存块 r31 空闲，于是执行体 c 开始执行，读内存块 r21, 写入内存块 r31。同时执行体 b 给上游的生产者执行体 a 发消息通知执行体 a 执行体 b 用完了执行体 a 的内存块 r11，执行体 a 收到了执行体 b 用完还回来的内存块 r11，检查内存块 r11 所有的消费者都用完了，于是将内存块 r11 回收，标记为空闲块。

[0040] 同时对于第 2 批次数据，当前执行体 a 向下游执行体 b 发送消息、下游执行体 b 准备待消费的张量数据：执行体 a 给执行体 b 发消息通知执行体 b 读取执行体 a 产出的内存块 r12，执行体 b 收到消息，并检查自己生产的内存块 b 是否有空闲内存块可用，发现有可用的空闲内存块 r22，于是执行体 b 执行前向算子 b 的核函数计算任务，读取内存块 r12，执行体 b 将执行生成的输出张量结果写入内存块 r22。

[0041] 于是执行体 a，执行体 b，执行体 c 就开始并行工作。执行体 C，B，A 由于没有数据可读，仍在等待。

[0042] T4 时刻：

对于第 4 批次数据，执行体 a 输入数据：执行体 a 还会同时检查自己是否有空闲块可写和执行体 A 执行完毕，发现没有，则等待不进入流水线。

[0043] 同时对于第 1 批次数据，当前执行体 c 向下游执行体 C 发送消息、下游执行体 C 准备待消费的张量数据、当前执行体 c 向上游执行体 b 发送消息、上游执行体 b 回收已被消费完的张量数据：执行体 c 生产出了内存块 r31，于是给下游的消费者执行体 C 发消息通知执行体 C 读取执行体 c 产出的内存块 r31，执行体 C 收到内存块 r31，发现自己有内存块 r11 空闲，

于是执行体 C 开始执行，读内存块 r31, 写入内存块 r11。同时执行体 c 给上游的生产者执行体 b 发消息通知执行体 b 执行体 c 用完了执行体 b 的内存块 r21，执行体 b 收到了执行体 c 用完还回来的内存块 r21，检查内存块 r21 所有的消费者都用完了，于是将内存块 r21 回收，标记为空闲块。

[0044] 同时对于第 2 批次数据，当前执行体 b 向下游执行体 c 发送消息、下游执行体 c 准备待消费的张量数据、当前执行体 b 向上游执行体 a 发送消息、上游执行体 a 回收已被消费完的张量数据：执行体 b 生产出了内存块 r22，于是给下游的消费者执行体 c 发消息通知执行体 c 读取执行体 b 产出的内存块 r22，执行体 c 收到内存块 r22，发现自己有内存块 r32 空闲，于是执行体 c 开始执行，读内存块 r22, 写入内存块 r32。同时执行体 b 给上游的生产者执行体 a 发消息通知执行体 a 执行体 b 用完了执行体 a 的内存块 r12，执行体 a 收到了执行体 b 用完还回来的内存块 r12，检查内存块 r12 所有的消费者都用完了，于是将内存块 r12 回收，标记为空闲块。

[0045] 同时对于第 3 批次数据，当前执行体 a 向下游执行体 b 发送消息、下游执行体 b 准备待消费的张量数据：执行体 a 给执行体 b 发消息通知执行体 b 读取执行体 a 产出的内存块 r13，执行体 b 收到消息，并检查自己生产的内存块 b 是否有空闲内存块可用，发现有可用的空闲内存块 r23，于是执行体 b 执行前向算子 b 的核函数计算任务，读取内存块 r13，执行体 b 将执行生成的输出张量结果写入内存块 r23。

[0046] 于是执行体 a，执行体 b，执行体 c，执行体 C 就开始并行工作。执行体 B，A 由于没有数据可读，仍在等待。

[0047] T5 时刻：

对于第 4 批次数据，执行体 a 输入数据：执行体 a 还会同时检查自己是否有空闲块可写和执行体 A 执行完毕，发现没有，则等待不进入流水线。

[0048] 同时对于第 1 批次数据，当前执行体 c 向下游执行体 C 发送消息、下游执行体 C 准备待消费的张量数据、当前执行体 c 向上游执行体 b 发送消息、上游执行体 b 回收已被消费完的张量数据：执行体 c 生产出了内存块 r11，于是给下游的消费者执行体 B 发消息通知执行体 B 读取执行体 C 产出的内存块 r11，执行体 B 收到内存块 r11，发现自己有内存块 r21 空闲，于是执行体 B 开始执行，读内存块 r11, 写入内存块 r21。同时执行体 C 给上游的生产者执行体 c 发消息通知执行体 c 执行体 C 用完了执行体 c 的内存块 r31，执行体 c 收到了执行体 C 用完还回来的内存块 r31，检查内存块 r31 所有的消费者都用完了，于是将内存块 r31 回收，标记为空闲块。

[0049] 同时对于第 2 批次数据，当前执行体 c 向下游执行体 C 发送消息、下游执行体 C 准备待消费的张量数据、当前执行体 c 向上游执行体 b 发送消息、上游执行体 b 回收已被消费完的张量数据：执行体 c 生产出了内存块 r32，于是给下游的消费者执行体 C 发消息通知执行体 C 读取执行体 c 产出的内存块 r32，执行体 C 收到内存块 r32，发现自己有内存块 r12 空闲，于是执行体 C 开始执行，读内存块 r32, 写入内存块 r12。同时执行体 c 给上游的生产者执行体 b 发消息通知执行体 b 执行体 c 用完了执行体 b 的内存块 r22，执行体 b 收到了执行体 c 用完还回来的内存块 r22，检查内存块 r22 所有的消费者都用完了，于是将内存块 r22 回收，标记为空闲块。

[0050] 同时对于第 3 批次数据，当前执行体 b 向下游执行体 c 发送消息、下游执行体 c 准备待消费的张量数据、当前执行体 b 向上游执行体 a 发送消息、上游执行体 a 回收已被消费完的张量数据：执行体 b 生产出了内存块 r23，于是给下游的消费者执行体 c 发消息通知执行体 c 读取执行体 b 产出的内存块 r23，执行体 c 收到内存块 r23，发现自己有内存块 r33 空闲，于是执行体 c 开始执行，读内存块 r23, 写入内存块 r33。同时执行体 b 给上游的生产者执行体 a 发消息通知执行体 a 执行体 b 用完了执行体 a 的内存块 r13，执行体 a 收到了执行体 b 用完还回来的内存块 r13，检查内存块 r13 所有的消费者都用完了，于是将内存块 r13 回收，标记为空闲块。

[0051] 于是执行体 a，执行体 b，执行体 c，执行体 C, 执行体 B 就开始并行工作。执行体 A 由于没有数据可读，仍在等待。

[0052] T6 时刻：

对于第 4 批次数据，执行体 a 输入数据：执行体 a 还会同时检查自己是否有空闲块可写和执行体 A 执行完毕，发现没有，则等待不进入流水线。

[0053] 同时对于第 1 批次数据，当前执行体 B 向下游执行体 A 发送消息、下游执行体 A 准备待消费的张量数据、尾执行体 A 自行回收计算数据当前执行体 B 向上游执行体 C 发送消息、上游执行体 C 回收已被消费完的张量数据：执行体 B 生产出了内存块 r21，于是给下游的消费者执行体 A 发消息通知执行体 A 读取执行体 B 产出的内存块 r21，执行体 A 收到内存块 r21，发现自己有内存块 r31 空闲，于是执行体 A 开始执行，读内存块 r21, 写入内存块 r31, 执行体 A 执行完毕即刻自行回收内存块 r31。同时执行体 B 给上游的生产者执行体 C 发消息通知执行体 C 执行体 B 用完了执行体 C 的内存块 r11，执行体 C 收到了执行体 B 用完还回来的内存块 r11，检查内存块 r11 所有的消费者都用完了，于是将内存块 r11 回收，标记为空闲块。

[0054] 同时对于第 2 批次数据，当前执行体 C 向下游执行体 B 发送消息、下游执行体 B 准

备待消费的张量数据、当前执行体 C 向上游执行体 c 发送消息、上游执行体 c 回收已被消费完的张量数据：执行体 C 生产出了内存块 r12，于是给下游的消费者执行体 B 发消息通知执行体 B 读取执行体 C 产出的内存块 r12，执行体 B 收到内存块 r12，发现自己有内存块 r22 空闲，于是执行体 B 开始执行，读内存块 r12, 写入内存块 r22。同时执行体 C 给上游的生产者执行体 c 发消息通知执行体 c 执行体 C 用完了执行体 c 的内存块 r32，执行体 c 收到了执行体 C 用完还回来的内存块 r32，检查内存块 r32 所有的消费者都用完了，于是将内存块 r32 回收，标记为空闲块。

[0055] 同时对于第 3 批次数据，当前执行体 c 向下游执行体 C 发送消息、下游执行体 C 准备待消费的张量数据、当前执行体 c 向上游执行体 b 发送消息、上游执行体 b 回收已被消费完的张量数据：执行体 c 生产出了内存块 r33，于是给下游的消费者执行体 C 发消息通知执行体 C 读取执行体 c 产出的内存块 r33，执行体 C 收到内存块 r33，发现自己有内存块 r13 空闲，于是执行体 C 开始执行，读内存块 r33, 写入内存块 r13。同时执行体 c 给上游的生产者执行体 b 发消息通知执行体 b 执行体 c 用完了执行体 b 的内存块 r23，执行体 b 收到了执行体 c 用完还回来的内存块 r23，检查内存块 r23 所有的消费者都用完了，于是将内存块 r23 回收，标记为空闲块。

[0056] 于是至此执行体 a，执行体 b，执行体 c，执行体 C，执行体 B，执行体 A 全部开始并行工作。

[0057] T7 时刻：

对于第 4 批次数据，执行体 a 输入数据：执行体 a 还会同时检查自己是否有空闲块可写和执行体 A 执行完毕，发现有，则执行体 a 也在执行第 4 批次输入数据，将执行结果写入内存空闲块 r11。

[0058] 同时对于第 1 批次数据，全部执行体执行完毕。

[0059] 同时对于第 2 批次数据，当前执行体 B 向下游执行体 A 发送消息、下游执行体 A 准备待消费的张量数据、当前执行体 B 向上游执行体 C 发送消息、上游执行体 C 回收已被消费完的张量数据：执行体 B 生产出了内存块 r22，于是给下游的消费者执行体 A 发消息通知执行体 A 读取执行体 B 产出的内存块 r22，执行体 A 收到内存块 r22，发现自己有内存块 r32 空闲，于是执行体 A 开始执行，读内存块 r22, 写入内存块 r32, 执行体 A 执行完毕即刻自行回收内存块 r32。同时执行体 B 给上游的生产者执行体 C 发消息通知执行体 C 执行体 B 用完了执行体 C 的内存块 r12，执行体 C 收到了执行体 B 用完还回来的内存块 r12，检查内存块 r11 所有的消费者都用完了，于是将内存块 r12 回收，标记为空闲块。

[0060] 同时对于第 3 批次数据，当前执行体 C 向下游执行体 B 发送消息、下游执行体 B 准备待消费的张量数据、当前执行体 C 向上游执行体 c 发送消息、上游执行体 c 回收已被消费完的张量数据：当前执行体 c 向下游执行体 C 发送消息、下游执行体 C 准备待消费的张量数据、当前执行体 c 向上游执行体 b 发送消息、上游执行体 b 回收已被消费完的张量数据：执行体 c 生产出了内存块 r13，于是给下游的消费者执行体 B 发消息通知执行体 B 读取执行体 C 产出的内存块 r13，执行体 B 收到内存块 r13，发现自己有内存块 r23 空闲，于是执行体 B 开始执行，读内存块 r13, 写入内存块 r23。同时执行体 C 给上游的生产者执行体 c 发消息通知执行体 c 执行体 C 用完了执行体 c 的内存块 r33，执行体 c 收到了执行体 C 用完还回来的内存块 r33，检查内存块 r33 所有的消费者都用完了，于是将内存块 r32 回收，标记为空闲块。于是执行体 a，执行体 b，执行体 c，执行体 C 就开始并行工作。执行体 B，A 由于没有数据可读，仍在等待。

[0061] T8 时刻：

执行体 a，b，c，C，B，A 都在工作，此时一个批次数据执行体全部执行完毕并完成输入下一组批次数据。通过多个空闲内存块的设计，执行体就实现了流水并行。

[0062] 以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内所作的任何修改、等同替换或改进等，均应包含在本发明的保护范围之内。

权 利 要 求 书

1. 一种面向神经网络模型计算的图执行流水并行方法，其特征在于，所述神经网络模型中设有若干个执行体，所述执行体共有 2*N 个，所述 N 为正整数，所述执行体设有若干个内存块；所述方法具体包括如下步骤：

S1、将训练数据分成若干个批次子数据；

S2、若干个批次子数据依次输入神经网络模型中，当第 i 批次的子数据输入后，第 n 执行体对第 i 批次的子数据执行自身核函数计算，并将执行结果写入第 n 执行体空闲的内存块中；接着输入第 i+1 批次的子数据；所述 i，n 均为正整数；

S3、当第 i+1 批次的子数据输入后，第 n 执行体对第 i+1 批次的子数据执行 S2 操作的同时，将第 i 批次所在的内存块的地址发送至第 n+1 执行体；第 n+1 执行体解析第 i 批次所在的内存块，得到第 n 执行体对第 i 批次的子数据的执行结果，并将第 n 执行体的执行结果作为第 n+1 执行体的输入数据，执行自身核函数计算，并将执行结果写入第 n+1 执行体空闲的内存块中；接着输入第 i+2 批次的子数据；

S4、当第 i+2 批次的子数据输入后，第 n 执行体对第 i+2 批次的子数据执行 S2 操作，第 n 执行体和第 n+1 执行体对第 i+1 批次的子数据执行 S3 的操作；同时第 n+1 执行体将第 i 批次所在的内存块的地址发送至第 n+2 执行体，第 n+2 执行体解析第 i 批次所在的内存块，得到第 n+1 执行体对第 i 批次的子数据的执行结果，并将第 n+1 执行体的执行结果作为第 n+2 执行体的输入数据，执行自身核函数计算，并将执行结果写入第 n+2 执行体空闲的内存块中；

S5、第 n 执行体回收发送给第 n+1 执行体的内存块；

S6、最后一个执行体执行自身核函数计算，并将执行结果写入最后一个执行体的内存块，执行完毕即刻自行回收内存块。

2. 如权利要求 1 所述的一种面向神经网络模型计算的图执行流水并行方法，其特征在于：执行体在执行自身核函数计算前，执行体会检查自身是否存在空闲的内存块；若存在，则对第 i 批次的子数据执行自身核函数计算；若不存在，则令第 i 批次等待存在空闲的内存块。

3. 如权利要求 2 所述的一种面向神经网络模型计算的图执行流水并行方法，其特征在于，对于第 N*n+1 批次的子数据，执行体在执行自身核函数计算前，会检查第 N*（n-1）+1 批次的子数据所在的执行体是否执行完毕，所述 n 为正整数。

4. 如权利要求 1 所述的一种面向神经网络模型计算的图执行流水并行方法，其特征在于：步骤 S5 具体包括如下操作：

S51、第 n+1 执行体通知第 n 执行体已消费完发送给第 n+1 执行体的内存块；

S52、第 n 执行体回收发送给第 n+1 执行体的内存块，并将其标记为空闲。

5. 如权利要求 1 所述的一种面向神经网络模型计算的图执行流水并行方法，其特征在于：还包括执行体的构造，所述执行体的构造具体包括如下子步骤：

S01、创建算子核函数的任务队列：将当前算子核函数的计算任务依次加入当前核函数任务队列；

S02、创建执行体的线程：所述执行体的线程负责从所述核函数任务队列中依次获取当前待处理任务，并提交给线程池；

S03、创建核函数的执行体：根据当前核函数任务和当前线程的上下文信息创建用于算子核函数计算的执行体；并使用执行体运行任务队列中的核函数任务；

S04、创建事件召回队列：将任务执行体处理完的任务添加到事件召回队列中；

S05、创建事件召回队列的线程：所述事件召回队列的线程负责将事件召回队列中已处理的任务依次取出并返回。

6. 一种面向神经网络模型计算的图执行装置，其特征在于：包括执行体构造模块和执行体流水并行工作模块，所述执行体构造模块用于执行体的构造，所述执行体流水并行工作模块用于执行如权利要求 1-4 任一项所述的一种面向神经网络模型计算的图执行流水并行方法。

7. 一种面向神经网络模型计算的图执行装置，其特征在于：包括存储器和一个或多个处理器，所述存储器中存储有可执行代码，所述一个或多个处理器执行所述可执行代码时，用于实现权利要求 1-5 任一项所述的一种面向神经网络模型计算的图执行流水并行方法。

8. 一种计算机可读存储介质，其特征在于：其上存储有程序，该程序被处理器执行时,实现权利要求 1-5 任一项所述的一种面向神经网络模型计算的图执行流水并行方法。

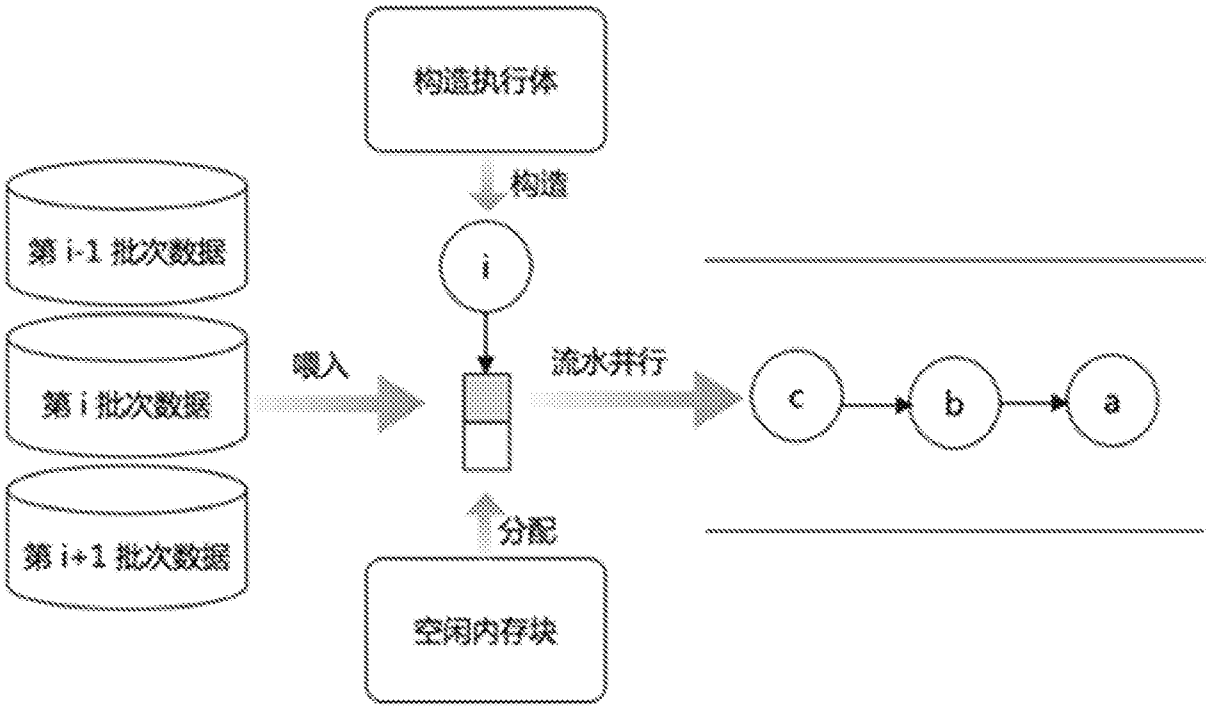
构造执行体
构造
第 i-1 批次数据
第 i 批次数据
第 i+1 批次数据
i
输入
流水并行
c
b
a
分配
空闲内存块

图 1

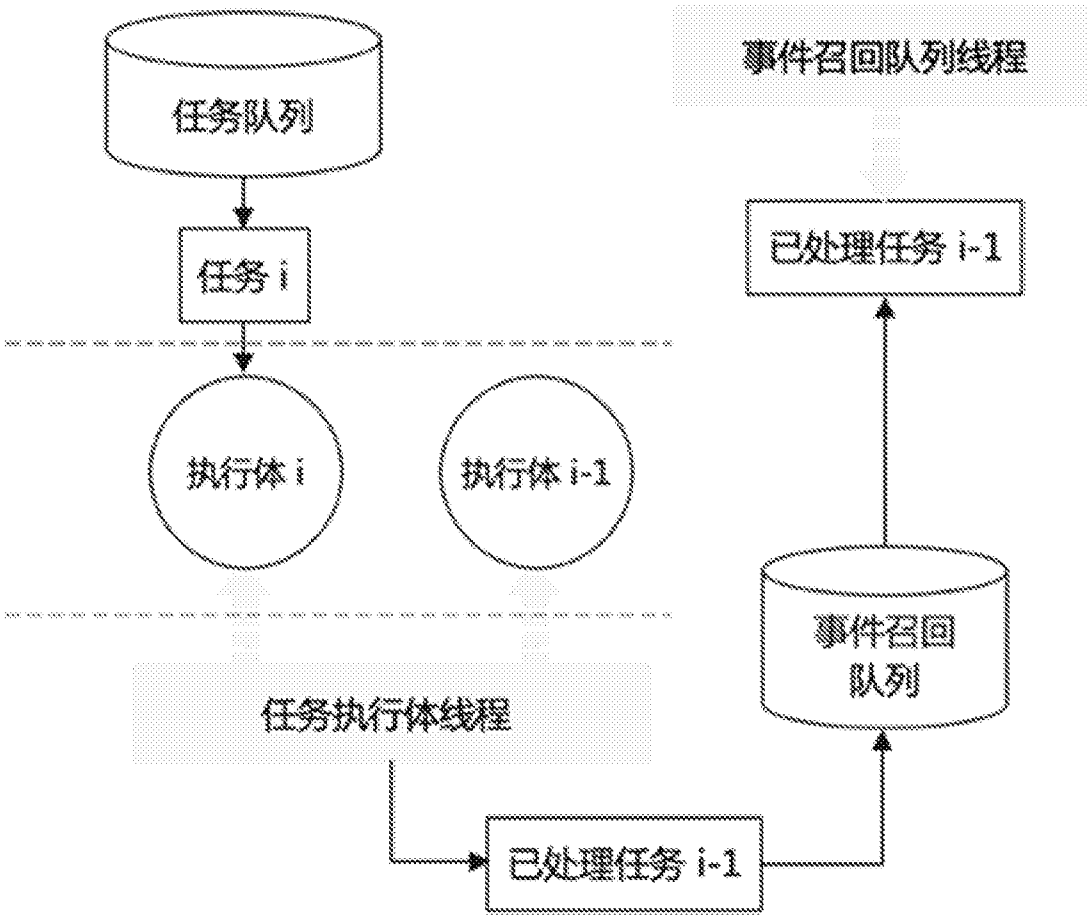
事件召回队列线程
任务队列
已处理任务 i-1
任务 i
执行体 i
执行体 i-1
事件召回
队列
任务执行体线程
已处理任务 i-1

图 2

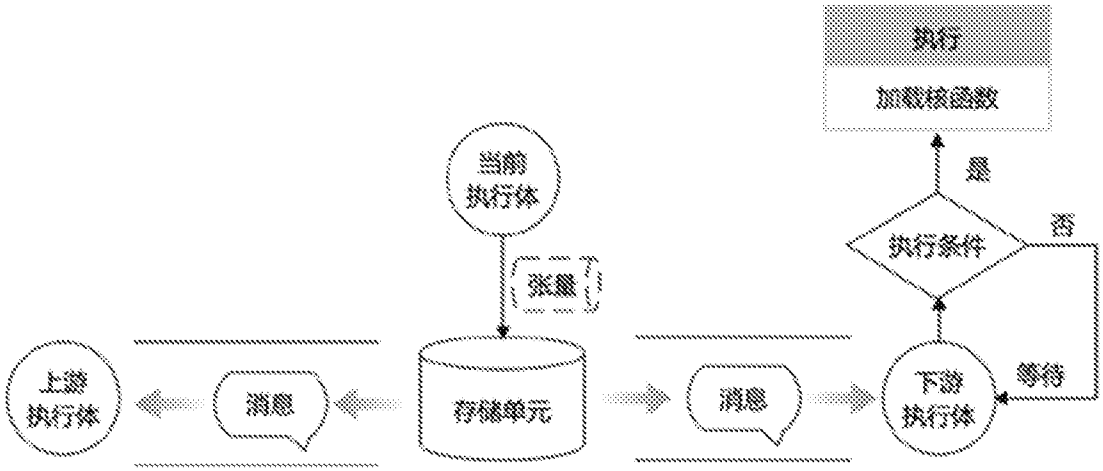

图 3

	计算图流 ->								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
r11	a1	a1		C1	C1		a4	a4	
r12		a2	a2		C2	C2		a5	a5
r13			a3	a3		C3	C3		a6
r21		b1	b1		B1	B1		b4	b4
r22			b2	b2		B2	B2		b5
r23				b3	b3		B3	B3	
r31			c1	c1		A1			c4
r32				c2	c2		A2		
r33					c3	c3		A3	

图 4

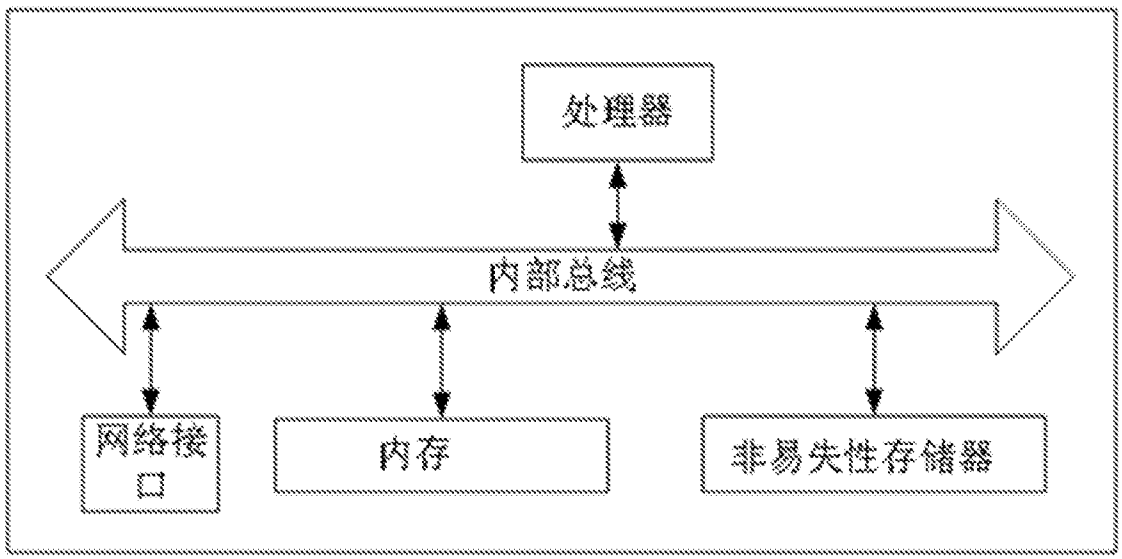

图 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2022/092481

A. CLASSIFICATION OF SUBJECT MATTER

G06N 3/04(2006.01)i; G06F 12/02(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N; G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNTXT, ENTXT, ENTXTC, CNKI, DWPI: 神经网络, 模型, 图, 执行, 并行, 内存块, 分批, 批次, 空闲, 计算, neural network, NN, model, graph, execut+, parallel, memory block, batch??, free, computation

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 114548383 A (ZHEJIANG LAB) 27 May 2022 (2022-05-27) claims 1-8	1-8
X	CN 114237918 A (ZHEJIANG LAB) 25 March 2022 (2022-03-25) description, paragraphs 24-60	1-8
A	CN 114139702 A (GUANGDONG INSPUR BIG DATA RESEARCH CO., LTD.) 04 March 2022 (2022-03-04) entire document	1-8
A	CN 114186687 A (ZHEJIANG LAB) 15 March 2022 (2022-03-15) entire document	1-8
A	CN 112884086 A (BEIJING BAIDU NETCOM SCIENCE AND TECHNOLOGY CO., LTD.) 01 June 2021 (2021-06-01) entire document	1-8
A	US 2019362227 A1 (MICROSOFT TECHNOLOGY LICENSING LLC.) 28 November 2019 (2019-11-28) entire document	1-8

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 November 2022	**05 January 2023**
Name and mailing address of the ISA/CN **China National Intellectual Property Administration (ISA/CN)** **No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088, China** Facsimile No. **(86-10)62019451**	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2022/092481

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 114548383 A	27 May 2022	None	
CN 114237918 A	25 March 2022	CN 114237918 B	27 May 2022
CN 114139702 A	04 March 2022	None	
CN 114186687 A	15 March 2022	CN 114186687 B	17 May 2022
CN 112884086 A	01 June 2021	None	
US 2019362227 A1	28 November 2019	EP 3797385 A1	31 March 2021
		CN 112154462 A	29 December 2020
		WO 2019226324 A1	28 November 2019
		IN 202017050201 A	12 February 2021

国际检索报告

国际申请号
PCT/CN2022/092481

A. 主题的分类

G06N 3/04(2006.01)i; G06F 12/02(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06N; G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

CNTXT,ENTXT,ENTXTC,CNKI,DWPI,神经网络，模型，图，执行，并行，内存块，分批，批次，空闲，计算，neural network，NN，model，graph，execut+，parallel，memory block，batch??，free，computation

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
PX	CN 114548383 A (之江实验室) 2022年5月27日 (2022-05-27) 权利要求1-8	1-8
X	CN 114237918 A (之江实验室) 2022年3月25日 (2022-03-25) 说明书第24-60段	1-8
A	CN 114139702 A (广东浪潮智慧计算技术有限公司) 2022年3月4日 (2022-03-04) 全文	1-8
A	CN 114186687 A (之江实验室) 2022年3月15日 (2022-03-15) 全文	1-8
A	CN 112884086 A (北京百度网讯科技有限公司) 2021年6月1日 (2021-06-01) 全文	1-8
A	US 2019362227 A1 (MICROSOFT TECHNOLOGY LICENSING LLC) 2019年11月28日 (2019-11-28) 全文	1-8

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2022年11月29日	2023年1月5日
ISA/CN的名称和邮寄地址 中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 史江峰 电话号码 62089926

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2022/092481

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	114548383	A	2022年5月27日		无		
CN	114237918	A	2022年3月25日	CN	114237918	B	2022年5月27日
CN	114139702	A	2022年3月4日		无		
CN	114186687	A	2022年3月15日	CN	114186687	B	2022年5月17日
CN	112884086	A	2021年6月1日		无		
US	2019362227	A1	2019年11月28日	EP	3797385	A1	2021年3月31日
				CN	112154462	A	2020年12月29日
				WO	2019226324	A1	2019年11月28日
				IN	202017050201	A	2021年2月12日