

计算图的推理方法、系统及存储介质

本申请要求于 2023 年 3 月 16 日提交中国专利局、申请号为 2023102699664、发明名称为“计算图的推理方法、系统及存储介质”的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。

5

技术领域

本申请涉及云计算领域，具体而言，涉及一种计算图的推理方法、系统及存储介质。

背景技术

10

目前，在进行远程推理时，通常需要将动态计算图转换为静态计算图，进而导出静态计算图对应的模型文件，最后对该模型文件进行远程推理。然而，在较多场景下，动态计算图无法成功转换为静态计算图，因此通常需要用户手动修改静态计算图，才能确保转换成功。

15

因而，在进行远程推理时，在较多场景下无法直接将动态计算图转换为静态计算图，从而容易导致计算图的远程推理效率低的技术问题。

针对上述的问题，目前尚未提出有效的解决方案。

发明内容

20

本申请实施例提供了一种计算图的推理方法、系统及存储介质，以至少解决计算图的推理效率低的技术问题。

25

根据本申请实施例的一个方面，提供了一种计算图的推理方法。该方法可以包括：监测待推理的原始类型计算图，其中，原始类型计算图为在同一时间段内构建和计算的计算图；响应于监测到的原始类型计算图转换为对应的目标类型计算图失败，将原始类型计算图划分为一级子原始类型计算图，其中，目标类型计算图为分别在不同时间段内构建和计算的计算图；将一级子原始类型计算图转化为对应的至少一目标类型计算图；将与一级子原始类型计算图对应的目标类型计算图，生成待推理文件，且将待推理文件发送至远程设备上，其中，待推理文件为提供给远程设备进行推理的文件。

30

根据本申请实施例的另一个方面，还提供了一种计算图的推理装置。该装置可以包括：监测模块，用于监测待推理的原始类型计算图，其中，原始类型计算图为在同一时间段内构建和计算的计算图；划分模块，用于响应于监测到的原始类型计算图转换为对应的目标类型计算图失败，将原始类型计算图划分为一级子原始类型计算图，其中，目标类型计算图为分别在不同时间段内构建和计算的计算图；转化模块，用于将一级子原始类型计算图转化为对应的至少一目标类型计算图；发送模块，用于将与一级子原始类型计算图对应的目标类型计算图，生成待推理文件，且将待推理文件发送至远程设备上，其中，待推理文件为提供给远程设备进行推理的文件。

35

根据本申请实施例的另一方面，还提供了一种计算图的推理系统，包括：本地设备，用于监测待推理的原始类型计算图，其中，原始类型计算图为在同一时间段内构建和计算

的计算图；响应于监测到的原始类型计算图转换为对应的目标类型计算图失败，将原始类型计算图划分为一级子原始类型计算图，其中，目标类型计算图为分别在不同时间段内构建和计算的计算图；将一级子原始类型计算图转化为对应的至少一目标类型计算图；将与一级子原始类型计算图对应的目标类型计算图生成待推理文件；远程设备，用于获取待推理文件，且对待推理文件进行推理，得到推理结果。

根据本申请实施例的另一方面，还提供了一种计算机可读存储介质，计算机可读存储介质包括存储的程序，其中，在程序被处理器运行时控制计算机可读存储介质所在设备执行计算图的推理方法。

容易注意到的是，上面的通用描述和后面的详细描述仅仅是为了对本申请进行举例和解释，并不构成对本申请的限定。

附图说明

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：

图 1 是根据本申请实施例的一种用于实现计算图的推理方法的计算机终端（或移动设备）的硬件结构框图。

图 2 是根据本申请实施例的一种计算图的推理方法的计算环境的结构框图。

图 3 是根据本申请实施例的一种服务网格的结构框图。

图 4 是根据本申请实施例的一种计算图的推理方法的流程图。

图 5 是根据本申请实施例的一种计算图的推理方法的示意图。

图 6 是根据本申请实施例的另一种计算图的推理方法的流程图。

图 7 是根据本申请实施例的一种计算图的推理系统的示意图。

图 8 是根据本申请实施例的一种计算图的推理装置的结构框图。

图 9 是根据本申请实施例的一种计算图的计算机终端的结构框图。

具体实施方式

为了使本技术领域的人员更好地理解本申请方案，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。

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，意图在于覆盖不排除的包含，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。

首先，在对本申请实施例进行描述的过程中出现的部分名词或术语适用于如下解释：

计算图，用于描述运算的有向无环图。

原始类型计算图，计算图的构建和计算同时发生，例如，动态计算图。

目标类型计算图，计算图的构建和计算分开，预先定义整个运算流，例如，静态计算图。

5 一级子原始类型计算图，原始类型计算图的子图。

二级子原始类型计算图，一级子原始类型计算图的子图。

目标算子，用于建立原始类型计算图的基础单元。

本地推理，在一台机器上调用深度学习框架提供的应用程序接口（Application Program Interface，简称为 API），来对构建好的计算图进行计算，推理也是在该机器上执行的。

10 远程推理，用户在本地机器上调用远程推理框架提供的 API，来实现推理功能，真正执行推理的是远程机器。远程推理的执行过程：本地机器基于远程推理框架将计算图通过网络发送至远程机器，并将用户的推理请求通过网络发送至远程机器，远程机器基于计算图和用户的推理请求完成推理操作，并将推理结果通过网络将返回至本地机器。

15 实施例 1

根据本申请实施例，提供了一种计算图的推理方法，需要说明的是，在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行，并且，虽然在流程图中示出了逻辑顺序，但是在某些情况下，可以以不同于此处的顺序执行所示出或描述的步骤。

20 本申请实施例一所提供的方法实施例可以在移动终端、计算机终端或者类似的运算装置中执行。图 1 示出了一种用于实现计算图的推理方法的计算机终端（或移动设备）的硬件结构框图。如图 1 所示，计算机终端 10（或移动设备）可以包括一个或多个（图中采用 102a、102b，……，102n 来示出）处理器 102（处理器 102 可以包括但不限于微处理器（Micro Controller Unit，MCU）或可编程逻辑器件（Field Programmable Gate Array，FPGA）等的处理装置）、用于存储数据的存储器 104、以及用于通信功能的传输模块 106。除此以外，还可以包括：显示器、输入/输出（Input/Output，I/O）接口、通用串行总线（Universal Serial Bus，USB）端口（可以作为 BUS 总线的端口中的一个端口被包括）、网络接口、电源和/或相机。本领域普通技术人员可以理解，图 1 所示的结构仅为示意，其并不对上述电子装置的结构造成限定。例如，计算机终端 10 还可包括比图 1 中所示更多或者更少的组件，或者具有与图 1 所示不同的配置。

30 应当注意到的是上述一个或多个处理器 102 和/或其他数据处理电路在本文中通常可以被称为“数据处理电路”。该数据处理电路可以全部或部分的体现为软件、硬件、固件或其他任意组合。此外，数据处理电路可为单个独立的处理模块，或全部或部分的结合到计算机终端 10（或移动设备）中的其他元件中的任意一个内。如本申请实施例中所涉及到的，该数据处理电路作为一种处理器控制（例如与接口连接的可变电阻终端路径的选择）。

35 存储器 104 可用于存储应用软件、软件程序以及模块，如本申请实施例中的计算图的推理方法对应的程序指令/数据存储装置，处理器 102 通过运行存储在存储器 104 内的软件程序以及模块，从而执行各种功能应用以及数据处理，即实现上述的计算图的推理方法。

存储器 104 可包括高速随机存储器, 还可包括非易失性存储器, 如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中, 存储器 104 可包括相对于处理器 102 远程设置的存储器, 这些远程存储器可以通过网络连接至计算机终端 10。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

5 传输装置 106 用于经由一个网络接收或者发送数据。上述的网络具体实例可包括计算机终端 10 的通信供应商提供的无线网络。在一个实例中, 传输装置 106 包括一个网络适配器 (Network Interface Controller, NIC), 其可通过基站与其他网络设备相连从而可与互联网进行通讯。在一个实例中, 传输装置 106 可以为射频 (Radio Frequency, RF) 模块, 其用于通过无线方式与互联网进行通讯。

10 显示器可以例如触摸屏式的液晶显示器 (Liquid Crystal Display, LCD), 该液晶显示器可使得用户能够与计算机终端 10 (或移动设备) 的用户界面进行交互。

图 1 示出的硬件结构框图, 不仅可以作为上述计算机终端 10 (或移动设备) 的示例性框图, 还可以作为上述服务器的示例性框图, 一种实施例中, 图 2 以框图示出了使用上述图 1 所示的计算机终端 10 (或移动设备) 作为计算环境 201 中计算节点的一种实施例。图 15 2 示出了一种计算图的推理方法的计算环境的结构框图, 如图 2 所示, 计算环境 201 包括运行在分布式网络上的多个 (图中采用 210-1, 210-2, ..., 来示出) 计算节点 (如服务器)。计算节点都包含本地处理和内存资源, 终端用户 202 可以在计算环境 201 中远程运行应用程序或存储数据。应用程序可以作为计算环境 201 中的多个服务 220-1, 220-2, 220-3 和 220-4 进行提供, 分别代表服务 “A”, “D”, “E” 和 “H”。

20 终端用户 202 可以通过客户端上的网页 (web) 浏览器或其他软件应用程序提供和访问服务, 在一些实施例中, 可以将终端用户 202 的供应和/或请求提供给入口网关 230。入口网关 230 可以包括一个相应的代理来处理针对服务 (计算环境 201 中提供的一个或多个服务) 的供应和/或请求。

服务是根据计算环境 201 支持的各种虚拟化技术来提供或部署的。在一些实施例中, 25 可以根据基于虚拟机 (Virtual Machine, VM) 的虚拟化、基于容器的虚拟化和/或类似的方式提供服务。基于虚拟机的虚拟化可以通过初始化虚拟机来模拟真实的计算机, 在不直接接触任何实际硬件资源的情况下执行程序 and 应用程序。在虚拟机虚拟化机器的同时, 根据基于容器的虚拟化, 可以启动容器来虚拟化整个操作系统 (Operating System, OS), 以便多个工作负载可以在单个操作系统实例上运行。

30 在基于容器虚拟化的一个实施例中, 服务的若干容器可以被组装成一个工作负载容器组 (Pod) (例如, Kubernetes Pod)。举例来说, 如图 2 所示, 服务 220-2 可以配备一个或多个 Pod 240-1, 240-2, ..., 240-N (统称为 Pod)。Pod 可以包括代理 245 和一个或多个容器 242-1, 242-2, ..., 242-M (统称为容器)。Pod 中一个或多个容器处理与服务的一个或多个相应功能相关的请求, 代理 245 通常控制与服务相关的网络功能, 如路由、负载均衡等。35 其他服务也可以配备类似的 Pod。

在操作过程中, 执行来自终端用户 202 的用户请求可能需要调用计算环境 201 中的一个或多个服务, 执行一个服务的一个或多个功能可能需要调用另一个服务的一个或多个功能。如图 2 所示, 服务 “A” 220-1 从入口网关 230 接收终端用户 202 的用户请求, 服务 “A”

220-1 可以调用服务“D” 220-2, 服务“D” 220-2 可以请求服务“E” 220-3 执行一个或多个功能。

5 上述的计算环境可以是云计算环境, 资源的分配由云服务提供上管理, 允许功能的开发无需考虑实现、调整或扩展服务器。该计算环境允许开发人员在不构建或维护复杂基础设施的情况下执行响应事件的代码。服务可以被分割完成一组可以自动独立伸缩的功能, 而不是扩展单个硬件设备来处理潜在的负载。

10 另一种实施例中, 图 3 以框图示出了使用上述图 1 所示的计算机终端 10(或移动设备)作为服务网格的一种实施例。图 3 示出了一种服务网格的结构框图, 如图 3 所示, 该服务网格 300 主要用于方便多个微服务之间进行安全和可靠的通信, 微服务是指将应用程序分解为多个较小的服务或者实例, 并分布在不同的集群/机器上运行。

如图 3 所示, 微服务可以包括应用服务实例 A 和应用服务实例 B, 应用服务实例 A 和应用服务实例 B 形成服务网格 300 的功能应用层。在一种实施方式中, 应用服务实例 A 以容器/进程 308 的形式运行在机器/工作负载容器组 314 (Pod), 应用服务实例 B 以容器/进程 310 的形式运行在机器/工作负载容器组 316 (Pod)。

15 在一种实施方式中, 应用服务实例 A 可以是对静态计算图对应的模型文件进行远程推理而实现的商品查询服务, 应用服务实例 B 可以是对另一静态计算图对应的模型文件进行远程推理而实现的商品下单服务, 其中, 上述静态计算图对应的模型文件可以由本申请实施例图 4 所示的计算图的推理方法得到。

20 如图 3 所示, 应用服务实例 A 和网格代理 (sidecar) 303 共存于机器工作负载容器组 614, 应用服务实例 B 和网格代理 305 共存于机器工作负载容器 314。网格代理 303 和网格代理 305 形成服务网格 300 的数据平面层 (data plane)。其中, 网格代理 303 和网格代理 305 分别以容器/进程 304, 容器/进程 306 的形式运行, 可以接收请求 312, 以用于进行商品查询服务, 并且网格代理 303 和应用服务实例 A 之间可以双向通信, 网格代理 305 和应用服务实例 B 之间可以双向通信。此外, 网格代理 303 和网格代理 305 之间还可以双向通信。

25 在一种实施方式中, 应用服务实例 A 的流量都通过网格代理 303 被路由到合适的目的地, 应用服务实例 B 的网络流量都通过网格代理 305 被路由到合适的目的地。需要说明的是, 在此提及的网络流量包括但不限于超文本传输协议 (Hyper Text Transfer Protocol, HTTP), 表述性状态传递 (Representational State Transfer, REST), 高性能、通用的开源框架 (google Remote Procedure Call, gRPC), 开源的内存中的数据结构存储系统 (如远程字典服务 (Remote Dictionary Server, Redis)) 等形式。

30 在一种实施方式中, 可以通过为服务网格 300 中的代理 (Envoy) 编写自定义的过滤器 (Filter) 来实现扩展数据平面层的功能, 服务网格代理配置可以是为了使服务网格正确地代理服务流量, 实现服务互通和服务治理。网格代理 303 和网格代理 305 可以被配置成执行至少如下功能中的一种: 服务发现 (service discovery), 健康检查 (health checking), 35 路由 (Routing), 负载均衡 (Load Balancing), 认证和授权 (authentication and authorization), 以及可观测性 (observability)。

如图 3 所示, 该服务网格 300 还包括控制平面层。其中, 控制平面层可以由一组在一个专用的命名空间中运行的服务, 在机器/工作负载容器组 (machine/Pod) 302 中由托管

控制面组件 301 来托管这些服务。如图 3 所示, 托管控制面组件 301 与网络代理 303 和网格代理 305 进行双向通信。托管控制面组件 301 被配置成执行一些控制管理的功能。例如, 托管控制面组件 301 接收网络代理 303 和网格代理 305 传送的遥测数据, 可以对这些遥测数据做聚合。这些服务, 托管控制面组件 301 还可以提供面向用户的应用程序接口 (Application Programming Interface, API), 以便较容易地操纵网络行为, 以及向网络代理 303 和网格代理 305 提供配置数据等。

在上述运行环境下, 本申请提供了如图 4 所示的计算图的推理方法。需要说明的是, 该实施例的计算图的推理方法可以由图 1 所示实施例的计算机终端执行, 可以应用于本地设备。图 4 是根据本申请实施例 1 的一种计算图的推理方法的流程图。如图 4 所示, 该方法可以包括如下步骤:

步骤 S401, 监测待推理的原始类型计算图, 其中, 原始类型计算图为在同一时间段内构建和计算的计算图。

在本申请上述步骤 S401 提供的技术方案中, 可以监测待推理的原始类型计算图, 其中, 原始类型计算图为在同一时间段内构建和计算的计算图, 例如, 该原始类型计算图可以为动态计算图, 其可以称为动态图。

在一实施方式中, 在监测到待推理的原始类型的计算图之后, 可以直接使用深度学习框架提供的 API 将原始类型计算图转换为目标类型计算图, 该目标类型计算图可以为静态计算图。

步骤 S402, 响应于监测到的原始类型计算图转换为对应的目标类型计算图失败, 将原始类型计算图划分为一级子原始类型计算图, 其中, 目标类型计算图为分别在不同时间段内构建和计算的计算图。

在本申请上述步骤 S402 提供的技术方案中, 在监测待推理的原始类型计算图之后, 可以对监测到的原始类型计算图进行类型转换, 比如, 对监测到的原始类型计算图进行从动态计算图到静态计算图的类型转换。响应于监测到的原始类型计算图转换为对应的目标类型计算图失败, 可以将原始类型计算图划分为一级子原始类型计算图。

在该实施例中, 由于在许多场景下, 可能存在无法直接将原始类型计算图转换为目标类型计算图的情况, 例如, 原始类型计算图中包含了可以简单有效地面向对象编程的计算机编程语言 python 特有的语法, 或者原始类型计算图的输入输出不是深度学习框架所能支持的类型, 在这种情况下, 深度学习框架提供的 API 在将原始类型计算图转换为目标类型计算图的过程中会自动抛出异常。当监测到深度学习框架提供的 API 抛出异常, 也即, 响应到原始类型计算图转换为对应的目标类型计算图失败, 之后, 可以将原始类型计算图划分为一级子原始类型计算图进行转换。

在一实施方式中, 上述目标类型计算图为在不同时间段内构建和计算的计算图, 可以为静态计算图, 其也可以称为静态图。由于原始类型计算图可以为动态计算图, 因此该一级子原始类型计算图可以为对动态计算图进行划分后得到的一级子动态计算图。

在一实施方式中, 如果原始类型计算图满足深度学习框架的标准格式, 则该原始类型计算图划分之后得到的一级子原始类型计算图的类型和数量是确定的。例如, 原始类型计算图满足深度学习框架的标准格式, 则该原始类型计算图划分后得到的一级子原始类型计

算图可以为如图 5 所示的子动态计算图 1、子动态计算图 2 以及子动态计算图 3，此处仅为示例性举例，并不对本申请实施例构成限定。

步骤 S403，将一级子原始类型计算图转化为对应的至少一目标类型计算图。

在本申请上述步骤 S403 提供的技术方案中，可以将一级子原始类型计算图转化为对应的至少一目标类型计算图，比如，可以将子动态计算图转化为对应的至少一子静态计算图。

在该实施例中，在将原始类型计算图划分为一级子原始类型计算图之后，可以将一级子原始类型计算图转化为对应的至少一目标类型计算图。

举例而言，由前述介绍可知，一级子原始类型计算图可以为一级子动态计算图，目标类型计算图可以为静态计算图，基于此，将一级子原始类型计算图转化为对应的至少一目标类型计算图，也即将一级子动态计算图转化为对应的至少一子静态计算图，其中，如果一级子原始类型计算图包括子动态计算图 1、子动态计算图 2 和子动态计算图 3，如图 5 所示，可以将子动态计算图 3 转化为对应的子静态计算图 3'。

步骤 S404，将与一级子原始类型计算图对应的目标类型计算图，生成待推理文件，且将待推理文件发送至远程设备上，其中，待推理文件为提供给远程设备进行推理的文件。

在该实施例中，在将一级子原始类型计算图转化为对应的至少一目标类型计算图之后，可以将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，且将待推理文件发送至远程设备上，以便由远程设备基于待推理文件执行推理操作，得到推理结果。

在该实施例中，待推理文件可以为模型文件，其中，模型文件可以为二进制数据，远程设备可以为远程计算机终端。

举例而言，在将一级子动态计算图转化为对应的至少一子静态计算图之后，可以将至少一子静态计算图生成模型文件，且将模型文件发送至远程设备，远程设备可以部署该模型文件，并加载该模型文件执行推理操作，得到推理结果。

在一实施方式中，在执行远程推理时，还可以向远程设备发送推理请求，其中，推理请求中包括推理需要的输入数据，远程设备在接收到推理请求后，可以响应该推理请求，加载模型文件来执行推理操作，得到推理结果。

在一实施方式中，在远程设备完成推理操作，得到推理结果之后，远程设备还可以向本地设备回传得到的推理结果，以实现推理结果的反馈。

基于上述实施例步骤 S401 至步骤 S404 所公开的方案，通过将原始类型计算图转换为对应的目标类型计算图，当转换失败时，可以将原始类型计算图划分为一级子原始类型计算图，由于一级子原始类型计算图可以进一步划分至根据深度学习框架定义的算子，而根据深度学习框架定义的算子必然能够转化为目标类型计算图，因此将原始类型计算图划分之后进行转换，可以提升转换的成功率，在将一级子原始类型计算图转化为对应的至少一目标类型计算图后，可以将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，并将待推理文件通过网络发送至远程设备，以使远程设备基于待推理文件执行推理操作，得到推理结果，能够实现将任意动态计算图转换为静态计算图，从而生成待推理文件进行推理，而无需人为对静态计算图进行手动修改，操作方式较为简单，达到了将任意动态计算图转换为静态计算图进行推理的目的，从而实现了提高计算图的推理效率的技术效

果，进而解决了计算图的推理效率低的技术问题。

下面对该实施例的上述方法进行介绍。

作为一种实施方式，步骤 S403 中，将一级子原始类型计算图转化为对应的至少一目标类型计算图，包括：对一级子原始类型计算图进行至少一级类型转换，得到对应的至少一目标类型计算图。

在该实施例中，通过递归方式对一级子原始类型计算图进行至少一级类型转换，直至可以成功得到对应的至少一目标类型计算图为止。

在该实施例中，通过递归方式对一级子原始类型计算图进行至少一级类型转换的过程如下：对一级子原始类型计算图进行一级类型转换，如果可以成功得到对应的至少一目标类型计算图，则停止转换，直接生成与目标类型计算图对应的模型文件，如果将一级子原始类型计算图进行一级类型转换后，无法得到对应的目标类型计算图，则需要将一级子原始类型计算图划分为二级子原始类型计算图，并将二级子原始类型计算图确定为新的一级子原始类型计算图，进而对新的一级子原始类型计算图进行一级类型转换，直至可以成功得到对应的目标类型计算图为止。

举例而言，如图 5 所示，对子动态计算图 3 进行类型转换，可以得到对应的子静态计算图 3'，对子动态计算图 2 进行类型转换，无法得到对应的子静态计算图，在这种情况下，可以将子动态计算图 2 进一步进行划分，得到子动态计算图 2 对应的子动态计算图 21 和子动态计算图 22，进而对子动态计算图 21 和子动态计算图 22 进行类型转换，得到子静态计算图 21' 和子静态计算图 22'。

在该实施例中，通过对一级子原始类型计算图进行至少一级类型转换，以得到对应的至少一目标类型计算图，能够通过递归的方式多次尝试转换，直至成功将子动态计算图转换为对应的子静态计算图，以便将子静态计算图生成待推理文件，并使远程终端设备基于待推理文件执行推理操作，得到推理结果，而无需人为修改静态计算图，提高了对计算图进行推理的效率。

作为一种实施方式，对一级子原始类型计算图进行至少一级类型转换，得到对应的至少一目标类型计算图，包括：转换步骤，对一级子原始类型计算图进行类型转换；确定步骤，确定一级子原始类型计算图转换为对应的目标类型计算图是否成功；划分步骤，响应于一级子原始类型计算图转换为对应的目标类型计算图失败，将一级子原始类型计算图划分为二级子原始类型计算图，且将二级子原始类型计算图确定为一级子原始类型计算图，返回转换步骤；获取步骤，响应于一级子原始类型计算图转换为对应的目标类型计算图成功，获取一级子原始类型计算图对应的目标类型计算图。

在该实施例中，二级子原始类型计算图为一级子原始类型计算图的子图。对一级子原始类型计算图进行转换，并判断转换是否成功，当转换失败时，将一级子原始类型计算图划分为二级子原始类型计算图，且将二级子原始类型计算图确定为一级子原始类型计算图，并返回转换步骤，对重新确定的一级子原始类型计算图进行转换，当转换失败时，将重新确定的一级子原始类型计算图再次划分为二级子原始类型计算图，且将二级子原始类型计算图确定为一级子原始类型计算图，继续返回转换步骤，基于此，采用递归的方式对一级子原始类型计算图进行至少一级类型转换，直至一级子原始类型计算图转换为对应的

目标类型计算图成功，从而可以获取一级子原始类型计算图对应的目标类型计算图。

举例而言，如图5所示，对子动态计算图2进行转换，并确定子动态计算图2是否成功转换为对应的子静态计算图，当转换失败时，可以将子动态计算图2划分为子动态计算图21和子动态计算图22，并将子动态计算图21和子动态计算图22确定为一级子动态计算图，并返回转换步骤，对子动态计算图21和子动态计算图22进行转换，子动态计算图21成功转换为对应的子静态计算图21'，而子动态计算图22转换为对应的子静态计算图失败，则将子动态计算图22再次划分为子动态计算图221和子动态计算图222，并将子动态计算图221和子动态计算图222确定为一级子动态计算图，继续返回转换步骤，对子动态计算图221和子动态计算图222进行转换，直至子动态计算图221成功转换为对应的子静态计算图221'，子动态计算图222成功转换为对应的子静态计算图222'。

在该实施例中，采用递归的方式对一级子原始类型计算图进行至少一级类型转换，直至一级子原始类型计算图转换为对应的目标类型计算图成功，从而可以获取一级子原始类型计算图对应的目标类型计算图，能够通过递归的方式多次尝试转换，直至将子动态计算图成功转换为子静态计算图，从而能够避免动态计算图无法转换为静态计算图的问题，并且无需人为手动修改静态计算图，进而能够提高对计算图进行推理的效率。

作为一种实施方式，原始类型计算图为由目标算子建立得到，目标算子用于成功转换为对应的目标类型计算图，其中，在将一级子原始类型计算图划分为二级子原始类型计算图之后，计算图的推理方法还包括：响应于二级子原始类型计算图为目标算子，获取由目标算子成功转换为目标类型计算图，且停止执行以下步骤：将二级子原始类型计算图确定为一级子原始类型计算图。

在该实施例中，目标算子为用于建立原始类型计算图的基础单元，可以为建立动态计算图的基础单元，可以包括深度学习框架定义的算子以及基础运算，其中，基础运算可以包括加减乘除，基于此，当子原始类型计算图划分至目标算子时，目标算子可以成功转换为静态计算图。因此，当二级子原始类型计算图为目标算子时，由于目标算子能够转换为目标类型计算图，因此可以直接获取由目标算子成功转换为目标类型计算图，并且无需进一步将二级子原始类型计算图确定为一级子原始类型计算图。

举例而言，如图5所示，子动态计算图1无法转换为对应的子静态计算图时，可以将子动态计算图1划分为子动态计算图11和子动态计算图12，当子动态计算图11和子动态计算图12为深度学习框架定义的算子时，可以直接获取由子动态计算图11和子动态计算图12成功转换的子静态计算图11'和子静态计算图12'，并且无需进一步将子动态计算图11和子动态计算图12确定为一级子动态计算图。

在该实施例中，通过对子动态计算图进行多次的划分，可以得到多级子动态计算图，当子动态计算图被划分至最内层时，也就是当子动态计算图被划分为多个深度学习框架定义的算子时，可以将多个深度学习框架定义的算子直接转换为对应的子静态计算图，从而能够将任意动态计算图转换为静态计算图，进而能够避免动态计算图无法转换为静态计算图的问题。

作为一种实施方式，计算图的推理方法还包括：基于深度学习框架构建目标算子，其中，深度学习框架用于使目标算子转换为对应的目标类型计算图。

在该实施例中，可以基于深度学习框架构建目标算子，其中，深度学习框架可以用于使目标算子转换为对应的目标类型计算图。

举例而言，在将目标算子转换为对应的子静态计算图时，可以使用深度学习框架进行转换操作。

5 作为一种实施方式，步骤 S403 中，将一级子原始类型计算图转化为对应的至少一目标类型计算图，包括：基于深度学习框架提供的接口，将一级子原始类型计算图转化为对应的至少一目标类型计算图。

在该实施例中，可以基于深度学习框架提供的接口，将一级子原始类型计算图转化为对应的至少一目标类型计算图。

10 举例而言，在将子动态计算图转换为子静态计算图时，可以使用深度学习框架提供的接口实现转换。

作为一种实施方式，基于深度学习框架提供的接口，将原始类型计算图转换为对应的目标类型计算图。

15 在该实施例中，还可以基于深度学习框架提供的接口，将原始类型计算图转换为对应的目标类型计算图。

举例而言，可以使用深度学习框架提供的接口，将动态计算图转换为对应的静态计算图。

作为一种实施方式，基于深度学习框架输出的异常提示信息，确定监测到的原始类型计算图转换为对应的目标类型计算图失败。

20 在该实施例中，还可以基于深度学习框架输出的异常提示信息，确定监测到的原始类型计算图转换为对应的目标类型计算图失败。

举例而言，在使用深度学习框架提供的接口将动态计算图转换为静态计算图失败时，深度学习框架的接口内部会自动输出异常提示信息，从而能够根据异常提示信息确定动态计算图转换为静态计算图失败。

25 作为一种实施方式，步骤 S402 中，将原始类型计算图划分为一级子原始类型计算图，包括：基于深度学习框架对应的格式，将原始类型计算图划分为一级子原始类型计算图，其中，原始类型计算图和一级子原始类型计算图满足深度学习框架对应的格式。

30 在该实施例中，可以基于深度学习框架对应的格式，将原始类型计算图划分为一级子原始类型计算图，其中，原始类型计算图和一级子原始类型计算图满足深度学习框架对应的格式。

举例而言，当动态计算图满足深度学习框架对应的格式，可以根据该深度学习框架对应的格式，将动态计算图划分为一级子动态计算图，其中，一级子动态计算图也满足深度学习框架对应的格式。

35 作为一种实施方式，响应于监测到的原始类型计算图转换为对应的目标类型计算图成功，将转化为的目标类型计算图生成待推理文件。

在该实施例中，当监测到的原始类型计算图成功转换为对应的目标类型计算图时，可以将转化为的目标类型计算图生成待推理文件。

举例而言，当动态计算图成功转换为对应的静态计算图时，可以将静态计算图生成模

型文件，进而可以将模型文件通过网络发送至远程计算机，以便远程计算机基于模型文件执行推理操作，得到推理结果。

在上述步骤中，通过将原始类型计算图转换为对应的目标类型计算图，当转换失败时，可以将原始类型计算图划分为一级子原始计算图，由于一级子原始类型计算图可以进一步划分至根据深度学习框架定义的算子，而根据深度学习框架定义的算子必然能够转化为目标类型计算图，因此将原始类型计算图划分之后进行转换，可以提升转换的成功率，在将一级子原始类型计算图转化为对应的至少一目标类型计算图后，可以将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，并将待推理文件通过网络发送至远程设备，以使远程设备基于待推理文件执行推理操作，得到推理结果，能够实现将任意动态计算图转换为静态计算图，从而生成待推理文件进行推理，而无需人为对静态计算图进行修改，操作方式较为简单，达到了将任意动态计算图转换为静态计算图进行推理的目的，从而实现了提高计算图的推理效率的技术效果，进而解决了计算图的推理效率低的技术问题。

需要说明的是，本申请所涉及的用户信息（包括但不限于用户设备信息、用户个人信息等）和数据（包括但不限于用于分析的数据、存储的数据、展示的数据等），均为经用户授权或者经过各方充分授权的信息和数据，并且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准，并提供有相应的操作入口，供用户选择授权或者拒绝。

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，本申请并不受所描述的动作顺序的限制，因为依据本申请，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一种实施例，所涉及的动作和模块并不一定是本申请所必须的。

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。基于这样的理解，本申请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（如只读存储器（Read-Only Memory, ROM）/随机存取存储器（Random Access Memory, 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）执行本申请各个实施例所述的方法。

实施例 2

下面对该实施例的上述方法的实施方式进行介绍。

目前，在利用深度学习框架进行远程推理时，通常需要将动态计算图转换为静态计算图，进而导出模型文件，最后进行远程推理。然而，在较多场景下，动态计算图无法成功转换为静态计算图。举例而言，在动态计算图中包含了 python 特有的语法，或者动态计算图的输入输出数据不属于深度学习框架所能支持的类型时，导致该动态计算图无法成功转换为静态计算图。

相关技术中，在动态计算图无法转换为静态计算图时，通常采取的解决方案是用户对

静态计算图进行手动修改，从而确保转换成功。由此可知，利用深度学习框架进行远程推理时，在较多场景下无法直接将动态计算图转换为静态计算图，从而容易导致计算图的远程推理效率低下的技术问题。

为解决上述问题，本申请实施例提出了一种计算图的推理方法，通过将原始类型计算图转换为对应的目标类型计算图，当转换失败时，可以将原始类型计算图划分为一级子原始计算图，由于一级子原始类型计算图可以进一步划分至根据深度学习框架定义的算子，而根据深度学习框架定义的算子必然能够转化为目标类型计算图，因此将原始类型计算图划分之后进行转换，可以提升转换的成功率，在将一级子原始类型计算图转化为对应的至少一目标类型计算图之后，可以将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，并将待推理文件通过网络发送至远程设备，以使远程设备基于待推理文件执行推理操作，得到推理结果，能够实现将任意动态计算图转换为静态计算图，从而生成待推理文件进行推理，而无需人为对静态计算图进行手动修改，操作方式较为简单，达到了将任意动态计算图转换为静态计算图进行推理的目的，从而实现了提高计算图的推理效率的技术效果，进而解决了计算图的推理效率低的技术问题。

下面对计算图的推理进行介绍。

图 6 是根据本申请实施例的另一种计算图的推理方法的流程图。如图 6 所示，首先监测待推理的原始类型计算图，并且基于深度学习框架提供的接口将原始类型计算图转换为对应的目标类型计算图，进而判断原始类型计算图是否成功转换为对应的目标类型计算图，当原始类型计算图成功转换为对应的目标类型计算图时，则直接将目标类型计算图生成待推理文件，当原始类型计算图转换为目标类型计算图失败时，则可以基于深度学习框架对应的格式，将原始类型计算图划分为一级子原始类型计算图，其中，一级子原始类型计算图也满足深度学习框架对应的格式，随后基于深度学习框架提供的接口，将所有的一级子原始类型计算图转换为对应的目标类型计算图，并判断一级子原始类型计算图是否成功转换为对应的目标类型计算图，当一级子原始类型计算图成功转换为对应的目标类型计算图时，则直接将目标类型计算图生成待推理文件，当一级子原始类型计算图转换为对应的目标类型计算图失败时，则可以基于深度学习框架对应的格式，将所有的一级子原始类型计算图进一步划分为二级子原始类型计算图，并将二级子原始类型计算图确定为一级子原始类型计算图，并返回转换步骤，即将一级子原始类型计算图转换为对应的目标类型计算图，能够通过递归的方式将一级子原始类型计算图划分为二级子原始类型计算图，直至二级原始类型计算图能够转换为对应的目标类型计算图，最后可以生成待推理文件。

图 5 是根据本申请实施例的一种计算图的推理方法的示意图。如图 5 所示，当监测到的动态计算图无法直接转换为静态计算图时，可以基于深度学习框架对应的格式，将动态计算图划分为子动态计算图 1、子动态计算图 2 以及子动态计算图 3，其中，子动态计算图 3 可以直接转换为对应的子静态计算图 3'，而子动态计算图 1 和子动态计算图 2 无法直接转换为对应的子静态计算图，因此将子动态计算图 1 划分为子动态计算图 11 和子动态计算图 12，其中，子动态计算图 11 和子动态计算图 12 能够直接转换为对应的子静态计算图 11' 和子静态计算图 12'，随后将子动态计算图 2 划分为子动态计算图 21 和子动态计算图 22，其中，子动态计算图 21 能够直接转换为对应的子静态计算图 21'，而子动态计算图

22 无法直接转换为对应的子静态计算图,因此将子动态计算图 22 划分为子动态计算图 221 和子动态计算图 222, 其中, 子动态计算图 221 和子动态计算图 222 能够直接转换为对应的子静态计算图 221' 和子静态计算图 222', 最后通过递归的方式可以将动态计算图逐层划分为 6 个子动态计算图,从而可以将 6 个子动态计算图转换为对应的 6 个子静态计算图, 5 进而能够将转换得到的 6 个子静态计算图生成 6 个模型文件并发送至远程计算机, 以便远程计算机按照顺序遍历模型文件, 并执行推理, 从而得到推理结果。

需要说明的是, 本申请上述实施例中涉及到的实施方案与实施例 1 提供的方案以及应用场景、实施过程相同, 但不仅限于实施例 1 所提供的方案。

实施例 3

10 根据本申请实施例, 还提供一种计算图的推理系统, 如图 7 所示, 计算图的推理系统包括本地设备 701, 用于监测待推理的原始类型计算图, 其中, 原始类型计算图为在同一时间段内构建和计算的计算图; 响应于监测到的原始类型计算图转换为对应的目标类型计算图失败, 将原始类型计算图划分为一级子原始类型计算图, 其中, 目标类型计算图为分别在不同时间段内构建和计算的计算图; 将一级子原始类型计算图转化为对应的至少一目标类型计算图; 将与一级子原始类型计算图对应的目标类型计算图生成待推理文件; 远程 15 设备 702, 用于获取待推理文件, 且对待推理文件进行推理, 得到推理结果。

在该实施例中, 可以利用本地设备监测待推理的原始类型计算图, 其中, 原始类型计算图为在同一时间段内构建和计算的计算图, 例如, 该原始类型计算图可以为动态计算图, 其可以称为动态图。

20 在一实施方式中, 在本地设备监测到待推理的原始类型的计算图之后, 可以直接使用深度学习框架提供的 API 将原始类型计算图转换为目标类型计算图, 该目标类型计算图可以为静态计算图。

在该实施例中, 在本地设备监测到待推理的原始类型计算图之后, 可以对本地设备监测到的原始类型计算图进行类型转换, 比如, 对本地设备监测到的原始类型计算图进行从 25 动态计算图到静态计算图的类型转换。响应于本地设备监测到的原始类型计算图转换为对应的目标类型计算图失败, 可以将原始类型计算图划分为一级子原始类型计算图。

在该实施例中, 由于在许多场景下, 可能存在无法直接将原始类型计算图转换为目标类型计算图的情况, 例如, 原始类型计算图中包含了可以简单有效地面向对象编程的计算机编程语言 python 特有的语法, 或者原始类型计算图的输入输出不是深度学习框架所能支持的类型, 在这种情况下, 深度学习框架提供的 API 在将原始类型计算图转换为目标类型 30 计算图的过程中会自动抛出异常。当监测到深度学习框架提供的 API 抛出异常, 也即, 响应到原始类型计算图转换为对应的目标类型计算图失败, 之后, 可以将原始类型计算图划分为一级子原始类型计算图进行转换。

在一实施方式中, 上述目标类型计算图为在不同时间段内构建和计算的计算图, 可以 35 为静态计算图, 其也可以称为静态图。由于原始类型计算图可以为动态计算图, 因此该一级子原始类型计算图可以为对动态计算图进行划分后得到的一级子动态计算图。

在一实施方式中, 如果原始类型计算图满足深度学习框架的标准格式, 则该原始类型计算图划分之后得到的一级子原始类型计算图的类型和数量是确定的。例如, 原始类型计

算图满足深度学习框架的标准格式，则该原始类型计算图划分后得到的一级子原始类型计算图，可以为如图 5 所示的子动态计算图 1、子动态计算图 2 以及子动态计算图 3，此处仅为示例性举例，并不对本申请实施例构成限定。

在该实施例中，可以在本地设备中将一级子原始类型计算图转化为对应的至少一目标类型计算图，比如，可以在本地设备中将子动态计算图转化为对应的至少一子静态计算图。

举例而言，由前述介绍可知，一级子原始类型计算图可以为一级子动态计算图，目标类型计算图可以为静态计算图，基于此，将一级子原始类型计算图转化为对应的至少一目标类型计算图，也即将一级子动态计算图转化为对应的至少一子静态计算图，其中，如果一级子原始类型计算图包括子动态计算图 1、子动态计算图 2 和子动态计算图 3，如图 5 所示，可以将子动态计算图 3 转化为对应的子静态计算图 3'。

在该实施例中，在本地设备将一级子原始类型计算图转化为对应的至少一目标类型计算图之后，可以将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，其中，待推理文件可以为模型文件，比如，模型文件可以为二进制数据。

在该实施例中，在本地设备将与一级子原始类型计算图对应的目标类型计算图生成待推理文件之后，远程设备可以基于网络获取待推理文件，且对待推理文件进行推理，得到推理结果。

在一实施方式中，本地设备 701 还用于基于深度学习框架将待推理文件发送至远程设备上。

在该实施例中，本地设备 701 还可以基于深度学习框架将待推理文件发送至远程设备上，以便远程设备加载待推理文件来执行推理操作，得到推理结果。

在一实施方式中，本地设备 701 还用于基于深度学习框架向远程设备发送推理请求，远程设备用于响应推理请求，对推理文件进行推理，得到推理结果。

在该实施例中，本地设备 701 还可以基于深度学习框架向远程设备发送推理请求，远程设备在接收到推理请求后，可以响应该推理请求，对推理文件进行推理操作，得到推理结果。

基于上述实施例的方案，通过在本地设备中将原始类型计算图转换为对应的目标类型计算图，当转换失败时，可以将原始类型计算图划分为一级子原始计算图，由于一级子原始类型计算图可以进一步划分至根据深度学习框架定义的算子，而根据深度学习框架定义的算子必然能够转化为目标类型计算图，因此在将原始类型计算图划分为一级子原始计算图后，可以提升转换的成功率，在将一级子原始类型计算图转化为对应的至少一目标类型计算图后，可以将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，并将待推理文件通过网络发送至远程设备，以使远程设备基于待推理文件执行推理操作，得到推理结果，能够实现将任意动态计算图转换为静态计算图，从而生成待推理文件进行推理，而无需人为对静态计算图进行手动修改，操作方式较为简单，达到了将任意动态计算图转换为静态计算图进行推理的目的，从而实现了提高计算图的推理效率的技术效果，进而解决了计算图的推理效率低的技术问题。

实施例 4

根据本申请实施例，还提供了一种用于实施上述图 4 所示的计算图的推理方法的计算

图的推理装置。

图 8 是根据本申请实施例的一种计算图的推理装置的结构框图。如图 8 所示，该计算图的推理装置可以包括：监测模块 801，划分模块 802，转化模块 803 和发送模块 804。

监测模块 801，用于监测待推理的原始类型计算图，其中，原始类型计算图为在同一时间段内构建和计算的计算图。

划分模块 802，用于响应于监测到的原始类型计算图转换为对应的目标类型计算图失败，将原始类型计算图划分为一级子原始类型计算图，其中，目标类型计算图为分别在不同时间段内构建和计算的计算图。

转化模块 803，用于将一级子原始类型计算图转化为对应的至少一目标类型计算图。

发送模块 804，用于将与一级子原始类型计算图对应的目标类型计算图，生成待推理文件，且将待推理文件发送至远程设备上，其中，待推理文件为提供给远程设备进行推理的文件。

在一实施方式中，转化模块 803 还用于将一级子原始类型计算图转化为对应的至少一目标类型计算图，包括：对一级子原始类型计算图进行至少一级类型转换，得到对应的至少一目标类型计算图。

在一实施方式中，转化模块 803 还用于对一级子原始类型计算图进行至少一级类型转换，得到对应的至少一目标类型计算图，包括：转换步骤，对一级子原始类型计算图进行类型转换；确定步骤，确定一级子原始类型计算图转换为对应的目标类型计算图是否成功；划分步骤，响应于一级子原始类型计算图转换为对应的目标类型计算图失败，将一级子原始类型计算图划分为二级子原始类型计算图，且将二级子原始类型计算图确定为一级子原始类型计算图，返回转换步骤；获取步骤，响应于一级子原始类型计算图转换为对应的目标类型计算图成功，获取与一级子原始类型计算图对应的目标类型计算图。

在一实施方式中，转化模块 803 还用于响应于二级子原始类型计算图为目标算子，获取由目标算子成功转换为目标类型计算图，且停止执行以下步骤：将二级子原始类型计算图确定为一级子原始类型计算图。

在一实施方式中，转化模块 803 还用于基于深度学习框架构建目标算子，其中，深度学习框架用于使目标算子转换为对应的目标类型计算图。

在一实施方式中，转化模块 803 还用于将一级子原始类型计算图转化为对应的至少一目标类型计算图，包括：基于深度学习框架提供的接口，将一级子原始类型计算图转化为对应的至少一目标类型计算图。

在一实施方式中，转化模块 803 还用于基于深度学习框架提供的接口，将原始类型计算图转换为对应的目标类型计算图。

在一实施方式中，上述计算图的推理装置还包括确定模块 805，用于基于深度学习框架输出的异常提示信息，确定监测到的原始类型计算图转换为对应的目标类型计算图失败。

在一实施方式中，划分模块 802 还用于将原始类型计算图划分为一级子原始类型计算图，包括：基于深度学习框架对应的格式，将原始类型计算图划分为一级子原始类型计算图，其中，原始类型计算图和一级子原始类型计算图满足深度学习框架对应的格式。

在一实施方式中，上述计算图的推理装置还包括生成模块 806，用于响应于监测到的原始类型计算图转换为对应的目标类型计算图成功，将转化为的目标类型计算图生成待推理文件。

5 此处需要说明的是，上述监测模块 801，划分模块 802，转化模块 803 和发送模块 804 对应于实施例 1 中的步骤 S401 至步骤 S404，需要说明的是，上述模块或单元可以是存储在存储器中并由一个或多个处理器处理的硬件组件或软件组件，上述模块也可以作为装置的一部分可以运行在实施例 1 提供的设备中。

实施例 5

10 本申请的实施例可以提供一种计算机终端，该计算机终端可以是计算机终端群中的任意一个计算机终端设备。在本实施例中，上述计算机终端也可以替换为移动终端等终端设备。

在本实施例中，上述计算机终端可以位于计算机网络的多个网络设备中的至少一个网络设备。

15 在本实施例中，上述计算机终端可以执行计算图的推理方法中以下步骤的程序代码：监测待推理的原始类型计算图，其中，原始类型计算图为在同一时间段内构建和计算的计算图；响应于监测到的原始类型计算图转换为对应的目标类型计算图失败，将原始类型计算图划分为一级子原始类型计算图，其中，目标类型计算图为分别在不同时间段内构建和计算的计算图；将一级子原始类型计算图转化为对应的至少一目标类型计算图；将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，且将待推理文件发送至远程设
20 备上，其中，待推理文件为提供给远程设备进行推理的文件。

图 9 是根据本申请实施例的一种计算机终端的结构框图。如图 9 所示，该计算机终端 A 可以包括：一个或多个（图中仅示出一个）处理器 902、存储器 904、存储控制器、以及外设接口，其中，外设接口与射频模块、音频模块和显示器连接。

25 其中，存储器可用于存储软件程序以及模块，如本申请实施例中的计算图的推理方法对应的程序指令/模块，处理器通过运行存储在存储器内的软件程序以及模块，从而执行各种功能应用以及数据处理，即实现上述的计算图的推理方法。存储器可包括高速随机存储器，还可以包括非易失性存储器，如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中，存储器可包括相对于处理器远程设置的存储器，这些远程存储器可以通过网络连接至终端 A。上述网络的实例包括但不限于互联网、企业内部网、
30 局域网、移动通信网及其组合。

处理器可以通过传输装置调用存储器存储的信息及应用程序，以执行下述步骤：监测待推理的原始类型计算图，其中，原始类型计算图为在同一时间段内构建和计算的计算图；响应于监测到的原始类型计算图转换为对应的目标类型计算图失败，将原始类型计算图划分为一级子原始类型计算图，其中，目标类型计算图为分别在不同时间段内构建和计算的
35 计算图；将一级子原始类型计算图转化为对应的至少一目标类型计算图；将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，且将待推理文件通过网络发送至远程设备上，其中，待推理文件为提供给远程设备进行推理的文件。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：对一级子原始类型计

算图进行至少一级类型转换，得到对应的至少一目标类型计算图。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：转换步骤，对一级子原始类型计算图进行类型转换；确定步骤，确定一级子原始类型计算图转换为对应的目标类型计算图是否成功；划分步骤，响应于一级子原始类型计算图转换为对应的目标类型计算图失败，将一级子原始类型计算图划分为二级子原始类型计算图，且将二级子原始类型计算图确定为一级子原始类型计算图，返回转换步骤；获取步骤，响应于一级子原始类型计算图转换为对应的目标类型计算图成功，获取一级子原始类型计算图对应的目标类型计算图。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：响应于二级子原始类型计算图为目标算子，获取由目标算子成功转换为目标类型计算图，且停止执行以下步骤：将二级子原始类型计算图确定为一级子原始类型计算图。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：基于深度学习框架构建目标算子，其中，深度学习框架用于使目标算子转换为对应的目标类型计算图。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：基于深度学习框架提供的接口，将一级子原始类型计算图转化为对应的至少一目标类型计算图。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：基于深度学习框架提供的接口，将原始类型计算图转换为对应的目标类型计算图。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：基于深度学习框架输出的异常提示信息，确定监测到的原始类型计算图转换为对应的目标类型计算图失败。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：基于深度学习框架对应的格式，将原始类型计算图划分为一级子原始类型计算图，其中，原始类型计算图和一级子原始类型计算图满足深度学习框架对应的格式。

在一实施方式中，上述处理器还可以执行如下步骤的程序代码：响应于监测到的原始类型计算图转换为对应的目标类型计算图成功，将转化为的目标类型计算图生成待推理文件。

采用本申请实施例，提供了一种计算图的推理方案。通过将原始类型计算图转换为对应的目标类型计算图，当转换失败时，可以将原始类型计算图划分为一级子原始计算图，由于一级子原始类型计算图可以进一步划分至根据深度学习框架定义的算子，而根据深度学习框架定义的算子必然能够转化为目标类型计算图，因此将原始类型计算图划分之后进行转换，可以提升转换的成功率，在将一级子原始类型计算图转化为对应的至少一目标类型计算图后，可以将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，并将待推理文件通过网络发送至远程设备，以使远程设备对待推理文件进行推理，得到推理结果，能够实现将任意动态计算图转换为静态图，从而生成待推理文件进行推理，而无需人为对静态计算图进行手动修改，操作方式较为简单，达到了将任意动态计算图转换为静态计算图进行推理的目的，从而实现了提高计算图的推理效率的技术效果，进而解决了计算图的推理效率低的技术问题。

本领域普通技术人员可以理解，图 9 所示的结构仅为示意，计算机终端也可以是智能手机（如安卓（Android）手机、苹果操作系统（iphone Operate System,iOS）手机等）、平

板电脑、掌上电脑以及移动互联网设备 (Mobile Internet Devices, MID)、笔记本电脑 (PAD) 等终端设备。图 9 其并不对上述电子装置的结构造成限定。例如, 计算机终端 A 还可包括比图 9 中所示更多或者更少的组件 (如网络接口、显示装置等), 或者具有与图 9 所示不同的配置。

- 5 本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通过程序来指令终端设备相关的硬件来完成, 该程序可以存储于一计算机可读存储介质中, 存储介质可以包括: 闪存盘、只读存储器 (Read-Only Memory, ROM)、随机存取器 (Random Access Memory, RAM)、磁盘或光盘等。

实施例 6

- 10 本申请的实施例还提供了一种计算机可读存储介质。在本实施例中, 上述计算机可读存储介质可以用于保存上述实施例一所提供的计算图的推理方法所执行的程序代码。

在本实施例中, 上述计算机可读存储介质可以位于计算机网络中计算机终端群中的任意一个计算机终端中, 或者位于移动终端群中的任意一个移动终端中。

- 15 在本实施例中, 计算机可读存储介质被设置为存储用于执行以下步骤的程序代码: 监测待推理的原始类型计算图, 其中, 原始类型计算图为在同一时间段内构建和计算的计算图; 响应于监测到的原始类型计算图转换为对应的目标类型计算图失败, 将原始类型计算图划分为一级子原始类型计算图, 其中, 目标类型计算图为分别在不同时间段内构建和计算的计算图; 将一级子原始类型计算图转化为对应的至少一目标类型计算图; 将与一级子原始类型计算图对应的目标类型计算图生成待推理文件, 且将待推理文件发送至远程设备上, 其中, 待推理文件为提供给远程设备进行推理的文件。

在本实施例中, 计算机可读存储介质被设置为存储用于执行以下步骤的程序代码: 对一级子原始类型计算图进行至少一级类型转换, 得到对应的至少一目标类型计算图。

- 25 在本实施例中, 计算机可读存储介质被设置为存储用于执行以下步骤的程序代码: 转换步骤, 对一级子原始类型计算图进行类型转换; 确定步骤, 确定一级子原始类型计算图转换为对应的目标类型计算图是否成功; 划分步骤, 响应于一级子原始类型计算图转换为对应的目标类型计算图失败, 将一级子原始类型计算图划分为二级子原始类型计算图, 且将二级子原始类型计算图确定为一级子原始类型计算图, 返回转换步骤; 获取步骤, 响应于一级子原始类型计算图转换为对应的目标类型计算图成功, 获取一级子原始类型计算图对应的目标类型计算图。

- 30 在本实施例中, 计算机可读存储介质被设置为存储用于执行以下步骤的程序代码: 响应于二级子原始类型计算图为目标算子, 获取由目标算子成功转换为对应的目标类型计算图, 且停止执行以下步骤: 将二级子原始类型计算图确定为一级子原始类型计算图。

- 35 在本实施例中, 计算机可读存储介质被设置为存储用于执行以下步骤的程序代码: 基于深度学习框架构建目标算子, 其中, 深度学习框架用于使目标算子转换为对应的目标类型计算图。

在本实施例中, 计算机可读存储介质被设置为存储用于执行以下步骤的程序代码: 基于深度学习框架提供的接口, 将一级子原始类型计算图转化为对应的至少一目标类型计算图。

在本实施例中，计算机可读存储介质被设置为存储用于执行以下步骤的程序代码：基于深度学习框架提供的接口，将原始类型计算图转换为对应的目标类型计算图。

在本实施例中，计算机可读存储介质被设置为存储用于执行以下步骤的程序代码：基于深度学习框架输出的异常提示信息，确定监测到的原始类型计算图转换为对应的目标类型计算图失败。

在本实施例中，计算机可读存储介质被设置为存储用于执行以下步骤的程序代码：基于深度学习框架对应的格式，将原始类型计算图划分为一级子原始类型计算图，其中，原始类型计算图和一级子原始类型计算图满足深度学习框架对应的格式。

在本实施例中，计算机可读存储介质被设置为存储用于执行以下步骤的程序代码：响应于监测到的原始类型计算图转换为对应的目标类型计算图成功，将转化后的目标类型计算图生成待推理文件。

在本申请实施例中，对原始类型计算图进行转换，当将原始类型计算图转换为对应的目标类型计算图失败时，可以将原始类型计算图划分为一级子原始类型计算图，进而将一级子原始类型计算图转化为对应的至少一目标类型计算图，最后将与一级子原始类型计算图对应的目标类型计算图生成待推理文件，并将待推理文件发送至远程设备，以使远程设备基于待推理文件执行推理操作。也就是说，在本申请实施例中，当原始类型计算图转换为对应的目标类型计算图失败时，可以将原始类型计算图划分为子原始类型计算图，对该子原始类型计算图进一步转换，如此递归执行，直至成功转换为目标类型计算图为止，这样能够实现将任意原始类型计算图转换为目标类型计算图，而无需人为对原始类型计算图进行手动修改，操作方式较为简单，达到了将任意原始类型计算图转换为目标类型计算图进行推理的目的，从而实现了提高计算图的推理效率的技术效果，进而解决了计算图的推理效率低的技术问题。

上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

在本申请的上述实施例中，对各个实施例的描述都各有侧重，某个实施例中未详述的部分，可以参见其他实施例的相关描述。

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。

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。

另外，在本申请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。

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，本申请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）执行本申请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，Random Access 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。

以上所述仅是本申请的实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本申请原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本申请的保护范围。

权 利 要 求

1. 一种计算图的推理方法，包括：
- 监测待推理的原始类型计算图，其中，所述原始类型计算图为在同一时间段内构建和计算的计算图；
- 5 响应于监测到的所述原始类型计算图转换为对应的目标类型计算图失败，将所述原始类型计算图划分为一级子原始类型计算图，其中，所述目标类型计算图为分别在不同时间段内构建和计算的计算图；
- 将所述一级子原始类型计算图转化为对应的至少一所述目标类型计算图；
- 将与所述一级子原始类型计算图对应的所述目标类型计算图，生成待推理文件，且将
- 10 所述待推理文件发送至远程设备上，其中，所述待推理文件为提供给所述远程设备进行推理的文件。
2. 根据权利要求 1 所述的方法，将所述一级子原始类型计算图转化为对应的至少一所述目标类型计算图，包括：
- 对所述一级子原始类型计算图进行至少一级类型转换，得到对应的至少一所述目标类
- 15 型计算图。
3. 根据权利要求 2 所述的方法，对所述一级子原始类型计算图进行至少一级类型转换，得到对应的至少一所述目标类型计算图，包括：
- 转换步骤，对所述一级子原始类型计算图进行类型转换；
- 确定步骤，确定所述一级子原始类型计算图转换为对应的所述目标类型计算图是否成
- 20 功；
- 划分步骤，响应于所述一级子原始类型计算图转换为对应的所述目标类型计算图失败，将所述一级子原始类型计算图划分为二级子原始类型计算图，且将所述二级子原始类型计算图确定为所述一级子原始类型计算图，返回所述转换步骤；
- 获取步骤，响应于所述一级子原始类型计算图转换为对应的所述目标类型计算图成
- 25 功，获取所述一级子原始类型计算图对应的所述目标类型计算图。
4. 根据权利要求 3 所述的方法，所述原始类型计算图为由目标算子建立得到，所述目标算子用于成功转换为对应的所述目标类型计算图，其中，在所述将所述一级子原始类型计算图划分为二级子原始类型计算图之后，所述方法还包括：
- 响应于所述二级子原始类型计算图为所述目标算子，获取由所述目标算子成功转换为
- 30 的所述目标类型计算图，且停止执行以下步骤：将所述二级子原始类型计算图确定为所述一级子原始类型计算图。
5. 根据权利要求 4 所述的方法，所述方法还包括：
- 基于深度学习框架构建所述目标算子，其中，所述深度学习框架用于使所述目标算子转换为对应的所述目标类型计算图。
- 35 6. 根据权利要求 1 所述的方法，将所述一级子原始类型计算图转化为对应的至少一所述目标类型计算图，包括：
- 基于深度学习框架提供的接口，将所述一级子原始类型计算图转化为对应的至少一所述目标类型计算图。

7. 根据权利要求 1 所述的方法，所述方法还包括：
基于深度学习框架提供的接口，将所述原始类型计算图转换为对应的所述目标类型计算图。
8. 根据权利要求 1 所述的方法，所述方法还包括：
- 5 基于深度学习框架输出的异常提示信息，确定监测到的所述原始类型计算图转换为对应的所述目标类型计算图失败。
9. 根据权利要求 1 所述的方法，将所述原始类型计算图划分为一级子原始类型计算图，包括：
- 10 基于深度学习框架对应的格式，将所述原始类型计算图划分为所述一级子原始类型计算图，其中，所述原始类型计算图和所述一级子原始类型计算图满足所述深度学习框架对应的格式。
10. 根据权利要求 1 至 9 中任意一项所述的方法，所述方法还包括：
响应于监测到的所述原始类型计算图转换为对应的所述目标类型计算图成功，将转化为的所述目标类型计算图生成所述待推理文件。
- 15 11. 一种计算图的推理系统，包括：
本地设备，用于监测待推理的原始类型计算图，其中，所述原始类型计算图为在同一时间段内构建和计算的计算图；响应于监测到的所述原始类型计算图转换为对应的目标类型计算图失败，将所述原始类型计算图划分为一级子原始类型计算图，其中，所述目标类型计算图分别为在不同时间段内构建和计算的计算图；将所述一级子原始类型计算图转化为对应的至少一所述目标类型计算图；将与所述一级子原始类型计算图对应的所述目标类型计算图生成的待推理文件；
- 20 远程设备，用于获取所述待推理文件，且对所述待推理文件进行推理，得到推理结果。
12. 根据权利要求 11 所述的系统，所述本地设备用于基于深度学习框架将所述待推理文件发送至所述远程设备上。
- 25 13. 根据权利要求 11 所述的系统，所述本地设备用于基于深度学习框架向所述远程设备发送推理请求，所述远程设备用于响应所述推理请求，对所述推理文件进行推理，得到所述推理结果。
14. 一种计算机可读存储介质，所述计算机可读存储介质包括存储的程序，其中，在所述程序被处理器运行时控制所述计算机可读存储介质所在设备执行权利要求 1 至 10 中任意一项所述的方法。
- 30