



(12) 发明专利申请

(10) 申请公布号 CN 112685275 A

(43) 申请公布日 2021.04.20

(21) 申请号 202011611065.1

(22) 申请日 2020.12.30

(71) 申请人 北京迈格威科技有限公司

地址 100090 北京市海淀区科学院南路2号
融科资讯中心A座316-318

(72) 发明人 柳志贤 史晓鹏

(74) 专利代理机构 北京超凡宏宇专利代理事务
所(特殊普通合伙) 11463

代理人 何少岩

(51) Int. Cl.

G06F 11/34 (2006.01)

G06F 11/30 (2006.01)

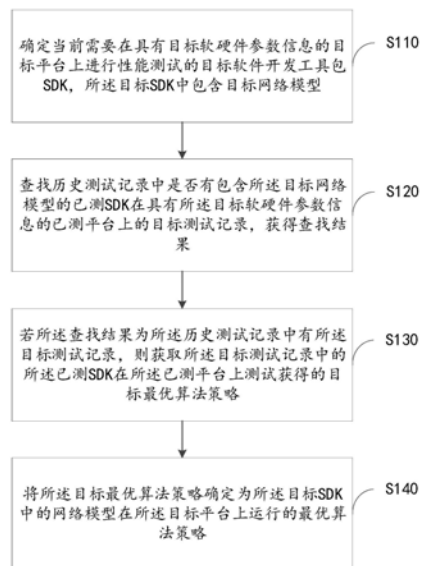
权利要求书2页 说明书15页 附图2页

(54) 发明名称

算法策略搜索方法、装置、电子设备及存储
介质

(57) 摘要

本申请提供一种算法策略搜索方法、装置、
电子设备及存储介质,涉及软件技术领域。该方
法通过查找历史测试记录中是否有包含目标网
络模型的SDK在具有目标软硬件参数信息的已测
平台上的目标测试记录,获得查找结果,这样可
在进行算法策略搜索之前查找历史测试记录,从
而在历史测试记录中有包括相应的目标测试记
录时,可直接复用目标测试记录中的目标最优算
法策略作为目标SDK在目标平台运行的最优算法
策略,无需再进行算法策略的搜索,可节省算法
策略搜索的时间,以减少整个性能测试的时间,
提高测试效率。



1. 一种算法策略搜索方法,其特征在于,所述方法包括:

确定当前需要在具有目标软硬件参数信息的目标平台上进行性能测试的目标软件开发工具包SDK,所述目标SDK中包含目标网络模型;

查找历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果;

若所述查找结果为所述历史测试记录中有所述目标测试记录,则获取所述目标测试记录中的所述已测SDK在所述已测平台上测试获得的目标最优算法策略;

将所述目标最优算法策略确定为所述目标SDK中的网络模型在所述目标平台上运行的最优算法策略。

2. 根据权利要求1所述的方法,其特征在于,还包括:

若所述查找结果为所述历史测试记录中未有所述目标测试记录,则进行算法策略搜索。

3. 根据权利要求2所述的方法,其特征在于,所述历史测试记录包括各个已测SDK中包含的网络模型的类型以及各个已测平台对应的软硬件参数信息;确定所述历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录的方式包括:

将所述历史测试记录中各个已测SDK包含的网络模型的类型与所述目标网络模型的类型进行匹配,以及将各个已测平台对应的软硬件参数信息与所述目标软硬件参数信息进行匹配;

若所述目标网络模型的类型与所述历史测试记录中的其中一个已测SDK包含的网络模型的类型一致,以及所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息一致,则确定所述历史测试记录中有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录;否则,则确定所述历史测试记录中未有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录。

4. 根据权利要求3所述的方法,其特征在于,确定所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息是否一致的方式包括:

判断所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息之间的偏差是否在预设范围内;

若是,则确定所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息一致。

5. 根据权利要求2所述的方法,其特征在于,所述进行算法策略搜索之后,还包括:

获取进行算法策略搜索后的搜索结果,所述搜索结果包括所述目标SDK中的网络模型在所述目标平台上使用的最优算法策略;

将所述搜索结果、所述目标平台的目标软硬件参数信息以及所述目标SDK中的目标网络模型存入所述历史测试记录中。

6. 根据权利要求5所述的方法,其特征在于,所述将所述搜索结果存入所述历史测试记录中,包括:

从预先创建的第一文件夹中读取所述搜索结果;

将所述搜索结果从所述第一文件夹中迁移到存储所述历史测试记录的第二文件夹中。

7. 一种算法策略搜索装置,其特征在于,所述装置包括:

信息确定模块,用于确定当前需要在具有目标软硬件参数信息的目标平台上进行性能测试的目标软件开发工具包SDK,所述目标SDK中包含目标网络模型;

信息查找模块,用于查找历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果;

算法策略获取模块,用于若所述查找结果为所述历史测试记录中有所述目标测试记录,则获取所述目标测试记录中的所述已测SDK在所述已测平台上测试获得的目标最优算法策略;

算法策略确定模块,用于将所述目标最优算法策略确定为所述目标SDK中的网络模型在所述目标平台上运行的最优算法策略。

8. 根据权利要求7所述的装置,其特征在于,所述算法策略获取模块,还用于若所述查找结果为所述历史测试记录中未有所述目标测试记录,则进行算法策略搜索。

9. 一种电子设备,其特征在于,包括处理器以及存储器,所述存储器存储有计算机可读指令,当所述计算机可读指令由所述处理器执行时,运行如权利要求1-6任一所述的方法。

10. 一种可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时运行如权利要求1-6任一所述的方法。

算法策略搜索方法、装置、电子设备及存储介质

技术领域

[0001] 本申请涉及软件技术领域,具体而言,涉及一种算法策略搜索方法、装置、电子设备及存储介质。

背景技术

[0002] 目前,网络模型使用不同的算法策略在平台上运行的性能通常不同,所以,为了使得网络模型在某个平台上运行的性能达到最优,需要进行算法策略的搜索,即搜索出最优的算法策略,以供网络模型使用。

[0003] 而现有技术中,将网络模型在不同的平台上进行性能测试时,均需要进行算法策略的搜索过程,该搜索过程耗时较长,需耗费大量的测试时间,导致测试效率较低。并且,在对大量的网络模型在大量的平台上进行性能测试时,若每次均需进行算法策略搜索,则需要耗费较长的时间。

发明内容

[0004] 本申请实施例的目的在于提供一种算法策略搜索方法、装置、电子设备及存储介质,用以改善现有技术中对网络模型在平台上进行性能测试过程中需要每次进行算法策略搜索而耗费较长测试时间,导致测试效率低的问题。

[0005] 第一方面,本申请实施例提供了一种算法策略搜索方法,所述方法包括:确定当前需要在具有目标软硬件参数信息的目标平台上进行性能测试的目标软件开发工具包SDK,所述目标SDK中包含目标网络模型;查找历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果;若所述查找结果为所述历史测试记录中有所述目标测试记录,则获取所述目标测试记录中的所述已测SDK在所述已测平台上测试获得的目标最优算法策略;将所述目标最优算法策略确定为所述目标SDK中的网络模型在所述目标平台上运行的最优算法策略。

[0006] 在上述实现过程中,通过查找历史测试记录中是否有包含目标网络模型的SDK在具有目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果,这样可在进行算法策略搜索之前查找历史测试记录,从而在历史测试记录中有包括相应的目标测试记录时,可直接复用目标测试记录中的目标最优算法策略作为目标SDK在目标平台运行的最优算法策略,无需再进行算法策略的搜索,可节省算法策略搜索的时间,以减少整个性能测试的时间,提高测试效率。

[0007] 可选地,还包括:

[0008] 若所述查找结果为所述历史测试记录中未有所述目标测试记录,则进行算法策略搜索。

[0009] 在上述实现过程中,在历史测试记录中未有相应的目标测试记录时,才进行算法策略搜索,从而在进行算法策略搜索之前来查找历史测试记录中是否有记录可复用的算法策略,对于不需要进行算法策略搜索时,可以节省算法策略搜索的时间,以减少目标网络模

型的测试时间。

[0010] 可选地,所述历史测试记录包括各个已测SDK中包含的网络模型的类型以及各个已测平台对应的软硬件参数信息;确定所述历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录的方式包括:

[0011] 将所述历史测试记录中各个已测SDK包含的网络模型的类型与所述目标网络模型的类型进行匹配,以及将各个已测平台对应的软硬件参数信息与所述目标软硬件参数信息进行匹配;

[0012] 若所述目标网络模型的类型与所述历史测试记录中的其中一个已测SDK包含的网络模型的类型一致,以及所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息一致,则确定所述历史测试记录中有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录;否则,则确定所述历史测试记录中未有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录。

[0013] 在上述实现过程中,通过将目标网络模型的类型与历史测试记录中记录的各个网络模型的类型进行比对,以及将目标软硬件参数信息与历史测试记录中记录的各个已测平台的软硬件参数信息进行比对,从而可更加准确确定出历史测试记录中是否包含有对应的目标测试记录。

[0014] 可选地,确定所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息是否一致的方式包括:

[0015] 判断所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息之间的偏差是否在预设范围内;

[0016] 若是,则确定所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息一致。

[0017] 在上述实现过程中,通过判断目标软硬件参数信息与其中一个已测平台的软硬件参数信息之间的偏差是否在预设范围,这样针对偏差在预设范围内的目标平台,也可无需进行算法策略搜索,可直接复用其中一个已测平台对应的算法策略,在针对大量的SDK在不同平台进行测试时,可大大节省测试时间。

[0018] 可选地,所述进行算法策略搜索之后,还包括:

[0019] 获取进行算法策略搜索后的搜索结果,所述搜索结果包括所述目标SDK中的网络模型在所述目标平台上使用的最优算法策略;

[0020] 将所述搜索结果、所述目标平台的目标软硬件参数信息以及所述目标SDK中的目标网络模型存入所述历史测试记录中。

[0021] 在上述实现过程中,将获得的搜索结果以及其他信息存储在历史测试记录中,以便于对后续的网络模型的性能测试提供更多的数据参考。

[0022] 可选地,所述将所述搜索结果存入所述历史测试记录中,包括:

[0023] 从预先创建的第一文件夹中读取所述搜索结果;

[0024] 将所述搜索结果从所述第一文件夹中迁移到存储所述历史测试记录的第二文件夹中。

[0025] 在上述实现过程中,将当前的搜索结果存储到另一文件夹中,可便于知晓每次的

搜索结果,并在历史测试记录中呈现针对各个网络模型的测试记录。

[0026] 第二方面,本申请实施例提供了一种算法策略搜索装置,所述装置包括:

[0027] 信息确定模块,用于确定当前需要在具有目标软硬件参数信息的目标平台上进行性能测试的目标软件开发工具包SDK,所述目标SDK中包含目标网络模型;

[0028] 信息查找模块,用于查找历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果;

[0029] 算法策略获取模块,用于若所述查找结果为所述历史测试记录中有所述目标测试记录,则获取所述目标测试记录中的所述已测SDK在所述已测平台上测试获得的目标最优算法策略;

[0030] 算法策略确定模块,用于将所述目标最优算法策略确定为所述目标SDK中的网络模型在所述目标平台上运行的最优算法策略。

[0031] 可选地,所述算法策略获取模块,还用于若所述查找结果为所述历史测试记录中未有所述目标测试记录,则进行算法策略搜索。

[0032] 可选地,所述历史测试记录包括各个已测SDK中包含的网络模型的类型以及各个已测平台对应的软硬件参数信息;所述信息查找模块,用于将所述历史测试记录中各个已测SDK包含的网络模型的类型与所述目标网络模型的类型进行匹配,以及将各个已测平台对应的软硬件参数信息与所述目标软硬件参数信息进行匹配;若所述目标网络模型的类型与所述历史测试记录中的其中一个已测SDK包含的网络模型的类型一致,以及所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息一致,则确定所述历史测试记录中有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录;否则,则确定所述历史测试记录中未有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录。

[0033] 可选地,所述信息查找模块,用于判断所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息之间的偏差是否在预设范围内;若是,则确定所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息一致。

[0034] 可选地,所述装置还包括:

[0035] 记录模块,用于获取进行算法策略搜索后的搜索结果,所述搜索结果包括所述目标SDK中的网络模型在所述目标平台上使用的最优算法策略;将所述搜索结果、所述目标平台的目标软硬件参数信息以及所述目标SDK中的目标网络模型存入所述历史测试记录中。

[0036] 可选地,所述记录模块,用于从预先创建的第一文件夹中读取所述搜索结果;将所述搜索结果从所述第一文件夹中迁移到存储所述历史测试记录的第二文件夹中。

[0037] 第三方面,本申请实施例提供一种电子设备,包括处理器以及存储器,所述存储器存储有计算机可读取指令,当所述计算机可读取指令由所述处理器执行时,运行如上述第一方面提供的所述方法中的步骤。

[0038] 第四方面,本申请实施例提供一种可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时运行如上述第一方面提供的所述方法中的步骤。

[0039] 本申请的其他特征和优点将在随后的说明书阐述,并且,部分地从说明书中变得显而易见,或者通过实施本申请实施例了解。本申请的目的和其他优点可通过在所写的说

说明书、权利要求书、以及附图中所特别指出的结构来实现和获得。

附图说明

[0040] 为了更清楚地说明本申请实施例的技术方案,下面将对本申请实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本申请的某些实施例,因此不应被看作是对范围的限定,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他相关的附图。

[0041] 图1为本申请实施例提供了一种算法策略搜索方法的流程图;

[0042] 图2为本申请实施例提供了一种算法策略搜索装置的结构框图;

[0043] 图3为本申请实施例提供了一种用于执行算法策略搜索方法的电子设备的结构示意图。

具体实施方式

[0044] 下面将结合本申请实施例中附图,对本申请实施例中的技术方案进行清楚、完整地描述。

[0045] 在介绍本申请的具体实施例之前,为了便于对本申请的理解,下面先介绍本申请涉及的相关背景知识。

[0046] 软件开发工具包(Software Development Kit,SDK)一般都是一些软件工程师为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件时的开发工具的集合。本申请实施例中为了便于对各个网络模型使用不同算法策略在各个平台上进行性能测试,可以将网络模型封装在SDK中,以通过SDK在平台上的运行来测试网络模型的性能。

[0047] 平台上包含网络模型所使用的多种算法策略,算法策略可以理解为是用于描述网络模型中的各个节点对应的算子的组合计算策略,比如网络模型中的某个节点可以表示为 $y(x) = a*x^2 + b*x + c$,其中, x 是输入, a, b, c 是该节点的权重参数,所谓的算法策略是指 x^2 、 $b*x$ 、 $a*x^2$ 这些算子在平台上是如何运算的,如可以在平台的cpu上计算 $b*x$ 、 $a*x^2$,可以在平台的gpu上计算 x^2 ,或者选择其他参与计算的参数等,这些参数是指可以影响计算速度的平台的一些软硬件参数,如何选择这些软硬件参数的组合来运行这些算子即可以理解为算法策略的搜索。

[0048] 各个软硬件参数的一种组合可称为一种算法策略,SDK在平台上运行时,可按照这多种算法策略进行运行,然后确定出运行速度最快的算法策略作为最优算法策略。

[0049] 例如,如平台中针对卷积神经网络模型设置有8种算法策略,卷积神经网络模型选择不同的算法策略在平台上运行时其性能可能不同,所以,需要从中选择一个最优算法策略作为卷积神经网络模型最后在该平台上运行时所使用的算法策略。在本申请实施例中,可以通过本申请实施例的算法策略搜索机制,从而可搜索得到最优的算法策略,从而实现性能最优。

[0050] 需要说的是,不同的平台上针对同一网络模型设置的算法策略可能不同,这样同一网络模型在不同平台实现性能最优时所确定的最优算法策略也可能不同。

[0051] 在本申请实施例中,为了测试网络模型采用不同算法策略在不同平台上的运行性能,可以获取网络模型采用不同算法策略在平台上进行运行时的性能,如对网络模型的计

算速度、计算时间等指标进行测试,从而可获得网络模型所使用的最优算法策略。

[0052] 持续集成(Continuous Integration,CI),即在代码构建过程中持续地进行代码的集成、构建、以及自动化测试等;有了CI工具,可以在代码提交的过程中通过单元测试等尽早地发现引入的错误;持续交付(Continuous Deployment,CD),在代码构建完毕后,可以方便地将新版本部署上线,这样有利于快速迭代并交付产品。

[0053] GitLab CI/CD(简称GitLab CI)是一套基于GitLab的CI/CD系统,可以让开发人员通过.gitlab-ci.yml在项目中配置CI/CD流程,在提交后,系统可以自动/手动地执行任务,完成CI/CD操作。

[0054] 目前针对某些SDK,如画质类SDK,配备了上述的GitLab CI/CD,目的是为了保证代码质量,提早发现code代码错误,提早解决。所以,如果希望CI/CD中测试的SDK性能最优,则需要对每次提交后的代码,重新在平台上使用现有的算法策略搜索机制,确定SDK中的网络模型使用的最优算法策略。

[0055] 现有的算法策略搜索机制分为两步,首先进行算法策略搜索,获得多个算法策略,然后找出最优的算法策略存入algo_policy文件夹中,再利用该文件生成一个缓存,在下次使用时,可直接快速从缓存中读取最优算法策略即可。而目前采用的算法策略搜索机制获得最优的算法策略的过程中,平均耗时约15分钟,假设一天有20次提交了新的SDK(其中存在更新模型的操作),则为了这20次CI/CD测试的性能都是最优,需要执行20次算法策略的搜索过程,即 $20 \times 15 = 300$ 分钟,但是假设有3个平台,有4个SDK产品需要进行测试,则需要 $3 \times 4 \times 300 = 3600$ 分钟,显然需要耗时较长的时间。

[0056] 以上现有技术中的方案所存在的缺陷,均是发明人在经过实践并仔细研究后得出的结果,因此,上述问题的发现过程以及下文中本发明实施例针对上述问题所提出的解决方案,都应该是发明人在本发明过程中对本发明做出的贡献。

[0057] 为了改善现有技术中测试过程耗时较长的问题,本申请实施例提供一种算法策略搜索方法,通过查找历史测试记录中是否有包含目标网络模型的已测SDK在具有目标软硬件参数信息的已测平台上的测试记录,获得查找结果,这样可在进行算法策略搜索之前查找历史测试记录,从而在历史测试记录中有包括相应的目标测试记录时,可直接复用目标测试记录中的目标最优算法策略作为目标SDK在目标平台运行的最优算法策略,无需再进行算法策略的搜索,可节省算法策略搜索的时间,以减少整个性能测试的时间,提高测试效率。

[0058] 请参照图1,图1为本申请实施例提供的一种算法策略搜索方法的流程图,该方法包括如下步骤:

[0059] 步骤S110:确定当前需要在具有目标软硬件参数信息的目标平台上进行性能测试的目标SDK,所述目标SDK中包含目标网络模型。

[0060] 其中,目标网络模型可以是指任何一种类型的网络模型,如卷积神经网络模型、循环神经网络模型等,目标网络模型使用不同的算法策略在目标平台上运行的性能通常不同。所以,需要对目标网络模型在目标平台上运行的最优性能进行测试,找到最优算法策略。

[0061] 例如,对于卷积神经网络模型和循环神经网络模型来说,其模型中均包含相应的卷积操作、池化操作等不同算法操作,这些操作在不同平台上,可以使用不同的算法策略来

运行,如不同的操作采用不同处理器实现,如直接使用CPU实现,或者利用GPU的cuda来实现,也可以利用Opencl来实现,在这些实现中,分块的大小、Opencl工作组之类的设置与选择,都会影响模型在平台上的运行性能,这些选择可以称为是算法策略搜索的一部分。

[0062] 例如,在某个平台上,对应的算法策略有5种,进行算法策略搜索时,就是把这5种算法策略都遍历一遍,然后查找执行速度最快的算法策略作为最优算法策略。

[0063] 也就是说,网络模型使用不同的算法策略在平台上运行,其性能一般有所差别,所以,为了使得网络模型在平台上运行的性能最优,需通过算法策略搜索过程找到最优的算法策略。

[0064] 其中,目标平台可以理解为是目标SDK需要进行性能测试所运行的平台,在实际应用中,平台可以是指不同型号的芯片,不同类型的手机等,或者不同的服务器等硬件平台,当然,其可以是指软件平台,如不同的操作系统。

[0065] 软硬件参数信息可以是指平台的相关参数信息,如对于平台为手机来说,其软硬件参数信息可以包括CPU和GPU、操作系统等硬件的相关信息,如CPU和GPU的型号、内存、运行速度等信息、操作系统的型号等信息,也即软硬件参数信息可以是指平台本身的软硬件信息等。当然,在平台为软件平台时,其对应软硬件参数信息可以包括该软件平台对应的参数信息和/或该软件平台所运行的硬件设备的硬件信息等。

[0066] 可以理解地,在实际情况中,可以根据需求设置各个平台对应的软硬件参数信息。对于不同类型的平台,其对应的软硬件参数信息的类型也可以不同。

[0067] 另外,获得目标软硬件参数信息的方式可以通过用户输入获得的,也可以是从预先存储的各个平台对应的软硬件参数信息中查找获得的。目标SDK可以是对当前需要进行测试的SDK进行标记而确定的,其标记的方式可以通过用户根据需求进行标记的。

[0068] 需要说明的是,若涉及到多个SDK需要在多个平台进行测试,对于在同一个平台上进行测试多个SDK时,可以根据需求设定相应的测试顺序,以依次将各个SDK在各个平台进行测试,从而可避免同时进行测试造成测试混乱的问题。而每个SDK在对应的平台上进行测试的测试过程与目标SDK在目标平台上进行测试的测试过程类似,本申请实施例中以目标SDK在目标平台上进行测试为例进行说明。

[0069] 步骤S120:查找历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果。

[0070] 历史测试记录中包括有包含各个网络模型的已测SDK在具有不同软硬件参数信息的已测平台上进行性能测试的测试记录,即历史测试记录包括多条测试记录,每条测试记录可以包括有包含网络模型的已测SDK在具有对应的软硬件参数信息的已测平台上进行性能测试的测试信息。

[0071] 所以,可以从历史测试记录中的多个测试记录中查找是否有包含目标网络模型的已测SDK在具有目标软硬件参数信息的已测平台上的目标测试记录,获得对应的查找结果。

[0072] 其中,获得的查找结果可以为历史测试记录中未包含有目标测试记录,或者查找结果为历史测试记录中包含有目标测试记录。

[0073] 若上述的目标SDK在其他平台上已经进行测试过了,当前需要将其在目标平台上进行测试,则历史测试记录中包括有该目标SDK的相关信息;或者,若别的SDK在目标平台上进行过测试,则历史测试记录中可包括有该目标平台的相关信息,如目标平台的软硬件参

数信息；若目标SDK为新的未测试过的SDK，而目标平台也为新加入的平台，则可以从历史测试记录中查找是否有与目标SDK匹配的已测SDK，以及与目标平台匹配的已测平台。

[0074] 步骤S130：若所述查找结果为所述历史测试记录中有所述目标测试记录，则获取所述目标测试记录中的所述已测SDK在所述已测平台上测试获得的目标最优算法策略。

[0075] 在查找结果为历史测试记录中有包含目标网络模型的已测SDK在具有目标硬件参数信息的已测平台的目标测试记录，则确定不需要进行算法策略搜索，表示目标SDK在目标平台上进行测试可无需再进行算法策略搜索，可直接复用历史测试记录中的目标测试记录，如可获取目标测试记录中记录的包含目标网络模型的已测SDK在具有目标硬件参数信息的已测平台上运行的目标最优算法策略，然后将该目标最优算法策略作为目标SDK在目标平台上运行的最优算法策略，从而可节约测试时间。

[0076] 步骤S140：将所述目标最优算法策略确定为所述目标SDK中的网络模型在所述目标平台上运行的最优算法策略。

[0077] 例如，若目标网络模型为卷积神经网络模型，从历史测试记录中的测试记录1中获取包含有卷积神经网络模型的SDK1在具有相同的目标硬件参数的平台1上运行时的目标最优算法策略，如算法策略1，此时，测试记录1即为上述的目标测试记录，SDK1即为已测SDK，平台1即为已测平台，则将算法策略1确定为目标SDK在目标平台上运行的最优算法策略。这种情况下，即表示测试目标SDK在目标平台上的运行性能时，不需要在进行算法策略的搜索了，而可以直接复用测试记录1中记载的算法策略1，这样可省去15分钟的算法策略的搜索时间，从而可有效减少测试时间。

[0078] 在上述实现过程中，通过查找历史测试记录中是否有包含目标网络模型的SDK在具有目标硬件参数信息的已测平台上的目标测试记录，获得查找结果，这样可在进行算法策略搜索之前查找历史测试记录，从而在历史测试记录中有包括相应的目标测试记录时，可直接复用目标测试记录中的目标最优算法策略作为目标SDK在目标平台运行的最优算法策略，无需再进行算法策略的搜索，可节省算法策略搜索的时间，以减少整个性能测试的时间，提高测试效率。

[0079] 在一些实施方式中，在查找结果为历史测试记录中未有包含目标网络模型的已测SDK在具有目标硬件参数信息的已测平台上的目标测试记录，则进行算法策略搜索，即运行目标SDK，然后找到该目标SDK运行性能最优时对应的最优算法策略。

[0080] 在上述实现过程中，在历史测试记录中未有相应的目标测试记录时，才进行算法策略搜索时，从而在进行算法策略搜索之前来查找历史测试记录中是否有记录可复用的算法策略，对于不需要进行算法策略搜索时，可以节省算法策略搜索的时间，以减少目标网络模型的测试时间。

[0081] 在一些实施方式中，历史测试记录中包括各个已测SDK中包含的网络模型的类型以及各个已测平台对应的硬件参数信息，确定历史测试记录中是否有包含目标网络模型的已测SDK在具有目标硬件参数信息的已测平台上的目标测试记录的方式包括：

[0082] 将历史测试记录中各个已测SDK包含的网络模型的类型与目标网络模型的类型进行匹配，以及将各个已测平台对应的硬件参数信息与目标硬件参数信息进行匹配；

[0083] 若目标网络模型的类型与历史测试记录中的其中一个已测SDK包含的网络模型的类型一致，以及目标硬件参数信息与历史测试记录中的其中一个已测平台的硬件参数

信息一致,则确定历史测试记录中有包含目标网络模型的已测SDK在具有目标软硬件参数信息的已测平台上的目标测试记录;

[0084] 若目标网络模型的类型与历史测试记录中的任意一个已测SDK包含的网络模型的类型不一致,和/或目标软硬件参数信息与历史测试记录中的任意一个已测平台的软硬件参数信息不一致,则确定历史测试记录中未有包含目标网络模型的已测SDK在具有目标软硬件参数信息的已测平台上的目标测试记录。

[0085] 其中,历史测试记录中记录的各个测试记录可如下表1所示:

[0086] 表1

测试记录	SDK 标识	网络模型的类型	平台标识	软硬件参数信息
[0087] 测试记录 1	SDK1	卷积神经网络模型	平台 1	CPU 型号: xx; CPU 运行速度: xx;
测试记录 2	SDK2	循环神经网络模型	平台 1	CPU 型号: xx; CPU 运行速度: xx;
[0088] 测试记录 3	SDK1	卷积神经网络模型	平台 2	CPU 型号: xx; CPU 运行速度: xx; GPU 型号: xx;
.....

[0089] 可以理解地,在历史测试记录中的每条测试记录中包括有SDK中包含的网络模型的类型以及平台的软硬件参数信息,所以,可以将目标SDK包含的目标网络模型的类型与每条测试记录中记载的网络模型的类型进行比对,判断其类型是否相同,以及将目标软硬件参数信息与每条测试记录中记载的软硬件参数信息进行比对,判断这些软硬件参数信息是否比对一致。

[0090] 其中,为了便于进行网络模型的类型比对,在进行测试记录存储时,可以将同一种类型的网络模型标记相同的标识,这样,在进行网络模型的类型比对时,可以将网络模

型的标识进行比对。例如,预先存储有各个不同类型的网络模型对应的标识,如卷积神经网络模型对应的标识为1,循环神经网络模型对应的标识为2,长短期记忆网络模型对应的标识为3,……,这样在进行历史测试记录存储时,可以根据每次测试时各个SDK中包含的网络模型的类型查找对应的标识,然后将标识记录在历史测试记录中。在获得目标SDK后,可以先获取目标SDK中包含的目标网络模型的类型,然后查找该类型的目标网络模型对应的标识,然后将标识与历史测试记录中的每条测试记录记载的网络模型的标识进行比对,在两个标识相同时,则认为目标网络模型的类型与该条测试记录所记载的网络模型的类型一致,若两个标识不相同,则认为目标网络模型的类型与该条测试记录所记载的网络模型的类型不一致。

[0091] 对于软硬件参数信息的比对时,可以将每个参数信息进行一一比对,其是将每个相同类型的参数信息之间进行比对,如目标软硬件参数信息包括CPU与GPU的型号、运行速度和内存,则可从每条测试记录中记录的软硬件参数信息中提取CPU的型号,将其与目标软硬件参数信息中的CPU型号进行比较,以及将测试记录中的CPU的运行速度与目标软硬件参数信息中的CPU的运行速度进行比较,将测试记录中的CPU的内存与目标软硬件参数信息中的CPU的内存进行比较;对于GPU的比对也如此,按照该方式,可将目标软硬件参数信息中的每个参数信息进行一一比对,在所有的参数信息均比对一致时,则确定目标软硬件参数信息与该条测试记录记载的软硬件参数信息一致,若在任一参数信息比对不一致时,则确定目标软硬件参数信息的与该条测试记录所记载的软硬件参数信息不一致。

[0092] 在一些其他实施方式中,在进行软硬件参数信息比对时,若目标软硬件参数信息包括多个参数信息时,可以在这多个参数信息中有预设数量以上的参数信息与某条测试记录中记载的软硬件参数信息比对一致,则认为目标软硬件参数信息与该条测试记录中记载的软硬件参数信息比对一致。

[0093] 其中,预设数量可以根据实际的参数信息的个数来灵活设定,如对于参数信息包括3个时,其预设数量可以设定为2或3,在实际情况中,可以灵活设定预设数量的值。

[0094] 例如,在上将目标网络模型的类型与测试记录1中的网络模型的类型进行比对,确定比对一致,将目标软硬件参数信息与测试记录1中的软硬件参数信息进行比对,确定比对一致时,则认为历史测试记录中有包含目标网络模型的已测SDK在具有目标软硬件参数信息的已测平台上的目标测试记录1,这种情况则不需要进行算法策略搜索。

[0095] 需要说明的是,上述进行网络模型的比对时,也可以不仅仅比对网络模型的类型,还可以比对网络模型的结构信息等其他相关信息;在进行软硬件参数信息时,可以只选择其中一些参数信息进行比对,而不需要将全部的参数信息进行比对。也就是说,在具体实现过程中,可以根据实际需求设置需要进行比对的信息,如还可以对网络模型的输入/输出数据的大小进行比对。

[0096] 在上述实现过程中,通过将目标网络模型的类型与历史测试记录中记录的各个网络模型的类型进行比对,以及将目标软硬件参数信息与历史测试记录中记录的各个已测平台的软硬件参数信息进行比对,从而可更加准确确定出历史测试记录中是否包含有对应的目标测试记录。

[0097] 在一些实施方式中,在进行软硬件参数信息比对时,还可以判断目标软硬件参数信息与历史测试记录中的其中一个已测平台的软硬件参数信息之间的偏差是否在预设范

围内,若是,则确定目标软硬件参数信息与历史测试记录中的其中一个已测平台的软硬件参数信息一致。

[0098] 其中,对于每个参数信息可以预先设置对应的预设范围,其预设范围可根据实际情况灵活设定,其对于不同的参数信息设置的预设范围可以不同。例如,若目标软硬件参数信息中的每个参数信息与上述表1中测试记录1中的软硬件参数信息中对应的参数信息之间的偏差均处于对应的预设范围时,则认为两者的软硬件参数信息比对一致。

[0099] 在上述实现过程中,通过判断目标软硬件参数信息与其中一个已测平台的软硬件参数信息之间的偏差是否在预设范围,这样针对偏差在预设范围内的目标平台,也可无需进行算法策略搜索,可直接复用其中一个已测平台对应的算法策略,在针对大量的SDK在不同平台进行测试时,可大大节省测试时间。

[0100] 在一些实施方式中,历史测试记录中还可以包括各个SDK中的网络模型在各个平台上运行的最优算法策略,如下表2所示:

[0101] 表2

[0102]

测试记录	已测 SDK 标识	网络模型 的类型	已测平台 标识	软硬件参 数信息	最优算法 策略
测试记录 1	SDK1	卷积神经 网络模型	平台 1	CPU 型 号：xx； CPU 运行 速度： xx；…… ；	算法策略 1
测试记录 2	SDK2	循环神经 网络模型	平台 1	CPU 型 号：xx； CPU 运行 速度： xx；…… ；	算法策略 2
测试记录 3	SDK1	卷积神经 网络模型	平台 2	CPU 型 号：xx； CPU 运行 速度： xx；GPU 型号： xx；…… ；	算法策略 3

[0103]

……	……	……	……	……	……
----	----	----	----	----	----

[0104] 所以,在上述确定不需要进行算法策略搜索后,从而可以目标测试记录中获得目标最优算法策略,然后将目标最优算法策略确定为目标SDK中的网络模型在目标平台上运行的最优算法策略。

[0105] 例如,若目标网络模型的类型为卷积神经网络模型,其目标软硬件参数信息与平台1中的软硬件参数信息匹配,则可从测试记录1(即目标测试记录)中获取SDK1在平台1上运行时的最优算法策略,即算法策略1,然后将算法策略1确定为目标SDK在目标平台上运行的最优算法策略。这种情况下,即表示测试目标SDK在目标平台上的运行性能时,不需要在

进行算法策略的搜索了,而可以直接复用测试记录1中记载的算法策略1,这样可省去15分钟的算法策略的搜索时间,从而可有效减少测试时间。

[0106] 在上述实现过程中,将目标最优算法策略进行复用,无需进行算法策略的搜索过程,减少了测试时间。

[0107] 在一些实施方式中,在确定需要进行算法策略搜索时,则可以执行算法策略搜索操作,通过搜索获得目标SDK在目标平台上运行时的最优算法策略,也就是说,可以获取进行算法策略搜索后的搜索结果,该搜索结果包括目标SDK中的网络模型在目标平台上使用的最优算法策略,然后可将搜索结果、目标平台的目标软硬件参数信息以及目标SDK中的目标网络模型存入历史测试记录中。

[0108] 其中,目标网络模型存入历史测试记录中可以是指存储的是目标网络模型的标识,或者目标网络模型的类型等信息,当然,也可以存储真正的目标网络模型的网络结构等参数信息。

[0109] 也即可以按照上述表2所示的格式将搜索结果、目标软硬件参数信息以及目标网络模型存储在历史测试记录中,作为历史测试记录积累下来,从而可以为后续的测试过程的算法策略的复用提供较多的数据。

[0110] 在一些实施方式中,可以将搜索结果存入预先创建的第一文件夹中,历史测试记录存储在第二文件夹中,如在系统中维护一个全局唯一的名称叫big_algo_policy的文件夹,该文件夹用于存储历史测试记录。然后可预设创建一个名称为empty_algo_policy的空文件夹,用于存储每次进行算法策略搜索后获得的新的搜索结果,即可以从预先创建的第一文件夹中读取搜索结果,然后将搜索结果从第一文件夹中迁移到存储历史测试记录的第二文件夹中。当然,也可以将搜索结果复制到第二文件夹中。

[0111] 例如,在上述判断是否需要进行搜索时,可以从big_algo_policy文件夹中读取历史测试记录,然后从历史测试记录中查找相应的测试记录,若确定需要进行算法策略搜索,则执行算法策略搜索操作。然后将搜索获得的搜索结果存入empty_algo_policy文件夹中,然后可通过ls-l命令,查看到empty_algo_policy的文件大小,若其文件大小不为0,则确定搜索结果在该文件夹中,此时将该文件夹中的内容拼接到big_algo_policy文件夹末尾,即将搜索结果移入big_algo_policy文件夹的末尾,如将上述表2的最后新增一行,然后将搜索结果、目标软硬件参数信息以及目标网络模型的类型添加到上述表2的最后新增一行中。

[0112] 当然,在一些其他实施方式中,还可以直接在存储历史测试记录的表格中的最后新建一行,然后将获得的搜索结果、目标软硬件参数信息以及目标网络模型的类型直接存入最后新增一行中。

[0113] 在上述实现过程中,将当前的搜索结果存储在另一文件夹中,可便于知晓每次的搜索结果,并在历史测试记录中呈现针对各个网络模型的测试记录。

[0114] 在一些实施方式中,历史测试记录还可存储于git仓库中,通过使用额外的git仓库存储历史测试记录,这样在每次进行性能测试时,可将历史测试记录从该git仓库中读取,然后进行后续的信息匹配过程。这样在人工发现问题时,可通过查找git仓库中存储的历史测试记录,即可对问题进行定位。

[0115] 请参照图2,图2为本申请实施例提供的一种算法策略搜索装置100的结构框图,该

装置100可以是电子设备上的模块、程序段或代码。应理解,该装置100与上述图1方法实施例对应,能够执行图1方法实施例涉及的各个步骤,该装置100具体的功能可以参见上文中的描述,为避免重复,此处适当省略详细描述。

[0116] 可选地,所述装置100包括:

[0117] 信息确定模块110,用于确定当前需要在具有目标软硬件参数信息的目标平台上进行性能测试的目标软件开发工具包SDK,所述目标SDK中包含目标网络模型;

[0118] 信息查找模块120,用于查找历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果;

[0119] 算法策略获取模块130,用于若所述查找结果为所述历史测试记录中有所述目标测试记录,则获取所述目标测试记录中的所述已测SDK在所述已测平台上测试获得的目标最优算法策略;

[0120] 算法策略确定模块140,用于将所述目标最优算法策略确定为所述目标SDK中的网络模型在所述目标平台上运行的最优算法策略。

[0121] 可选地,所述算法策略获取模块130,还用于若所述查找结果为所述历史测试记录中未有所述目标测试记录,则进行算法策略搜索。

[0122] 可选地,所述历史测试记录包括各个已测SDK中包含的网络模型的类型以及各个已测平台对应的软硬件参数信息;所述信息查找模块120,用于将所述历史测试记录中各个已测SDK包含的网络模型的类型与所述目标网络模型的类型进行匹配,以及将各个已测平台对应的软硬件参数信息与所述目标软硬件参数信息进行匹配;若所述目标网络模型的类型与所述历史测试记录中的其中一个已测SDK包含的网络模型的类型一致,以及所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息一致,则确定所述历史测试记录中有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录;否则,则确定所述历史测试记录中未有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录。

[0123] 可选地,所述信息查找模块120,用于判断所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息之间的偏差是否在预设范围内;若是,则确定所述目标软硬件参数信息与所述历史测试记录中的其中一个已测平台的软硬件参数信息一致。

[0124] 可选地,所述装置100还包括:

[0125] 记录模块,用于获取进行算法策略搜索后的搜索结果,所述搜索结果包括所述目标SDK中的网络模型在所述目标平台上使用的最优算法策略;将所述搜索结果、所述目标平台的目标软硬件参数信息以及所述目标SDK中的目标网络模型存入所述历史测试记录中。

[0126] 可选地,所述记录模块,用于从预先创建的第一文件夹中读取所述搜索结果;将所述搜索结果从所述第一文件夹中迁移到存储所述历史测试记录的第二文件夹中。

[0127] 需要说明的是,本领域技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的装置的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再重复描述。

[0128] 请参照图3,图3为本申请实施例提供的一种用于执行算法策略搜索方法的电子设备的结构示意图,所述电子设备可以包括:至少一个处理器210,例如CPU,至少一个通信接口220,至少一个存储器230和至少一个通信总线240。其中,通信总线240用于实现这些组件

直接的连接通信。其中,本申请实施例中设备的通信接口220用于与其他节点设备进行信令或数据的通信。存储器230可以是高速RAM存储器,也可以是非易失性的存储器(non-volatile memory),例如至少一个磁盘存储器。存储器230可选的还可以是至少一个位于远离前述处理器的存储装置。存储器230中存储有计算机可读取指令,当所述计算机可读取指令由所述处理器210执行时,电子设备执行上述图1所示方法过程。

[0129] 本申请实施例提供一种可读存储介质,所述计算机程序被处理器执行时,执行如图1所示方法实施例中电子设备所执行的方法过程。

[0130] 本实施例公开一种计算机程序产品,所述计算机程序产品包括存储在非暂态计算机可读存储介质上的计算机程序,所述计算机程序包括程序指令,当所述程序指令被计算机执行时,计算机能够执行上述各方法实施例所提供的方法,例如,包括:确定当前需要在具有目标软硬件参数信息的目标平台上进行性能测试的目标软件开发工具包SDK,所述目标SDK中包含目标网络模型;查找历史测试记录中是否有包含所述目标网络模型的已测SDK在具有所述目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果;若所述查找结果为所述历史测试记录中有所述目标测试记录,则获取所述目标测试记录中的所述已测SDK在所述已测平台上测试获得的目标最优算法策略;将所述目标最优算法策略确定为所述目标SDK中的网络模型在所述目标平台上运行的最优算法策略。

[0131] 综上所述,本申请实施例提供的一种算法策略搜索方法、装置、电子设备及存储介质,通过查找历史测试记录中是否有包含目标网络模型的SDK在具有目标软硬件参数信息的已测平台上的目标测试记录,获得查找结果,这样可在进行算法策略搜索之前查找历史测试记录,从而在历史测试记录中有包括相应的目标测试记录时,可直接复用目标测试记录中的目标最优算法策略作为目标SDK在目标平台运行的最优算法策略,无需再进行算法策略的搜索,可节省算法策略搜索的时间,以减少整个性能测试的时间,提高测试效率。

[0132] 在本申请所提供的实施例中,应该理解到,所揭露装置和方法,可以通过其它的方式实现。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,又例如,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通信接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0133] 另外,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。

[0134] 再者,在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部分,也可以是各个模块单独存在,也可以两个或两个以上模块集成形成一个独立的部分。

[0135] 在本文中,诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。

[0136] 以上所述仅为本申请的实施例而已,并不用于限制本申请的保护范围,对于本领域的技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原则之内,所作的

任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。

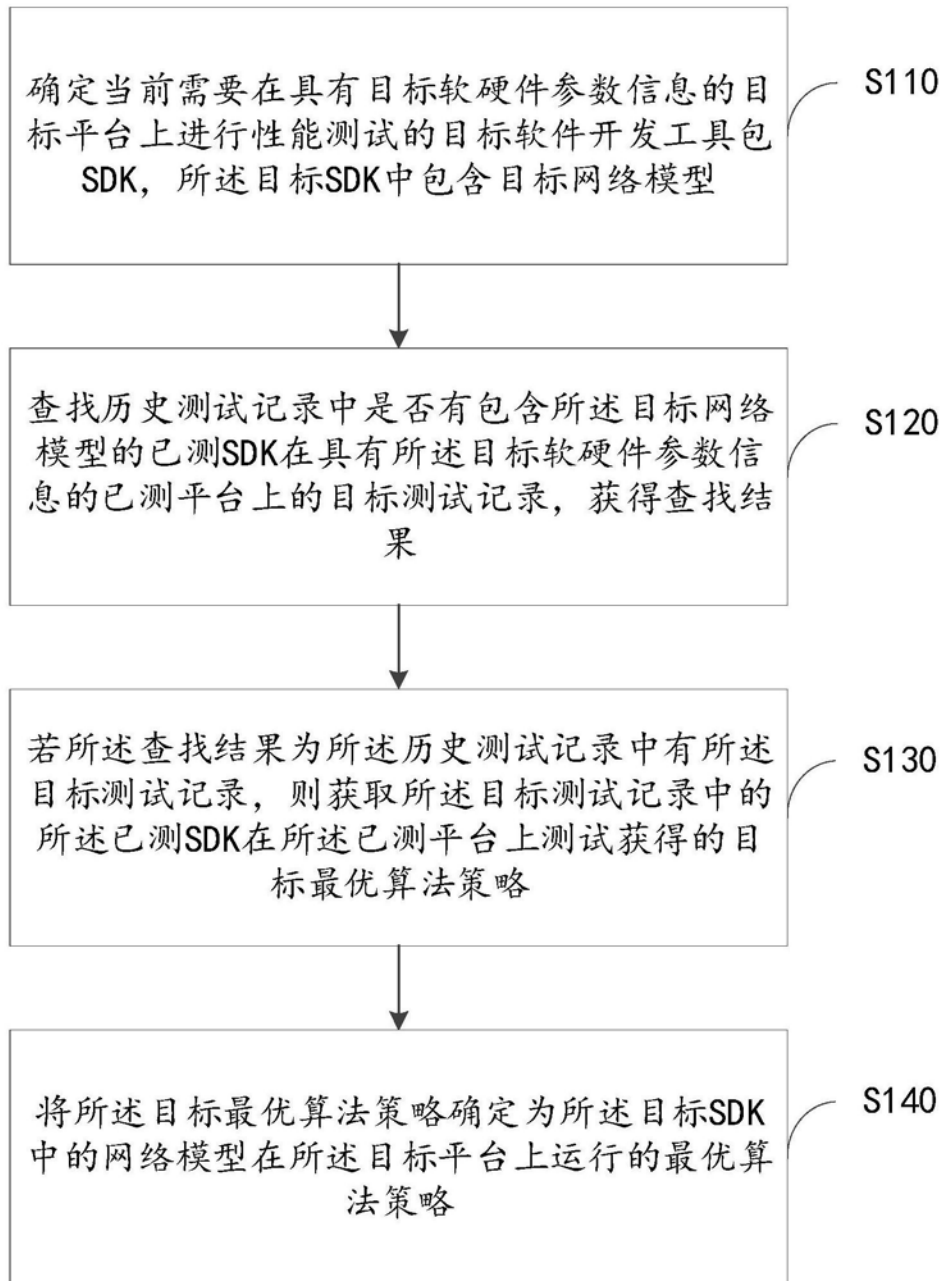


图1

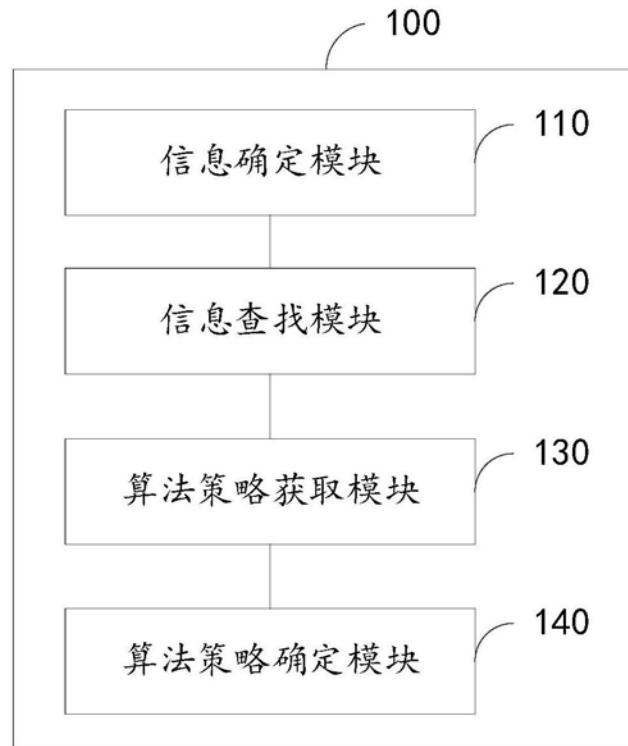


图2

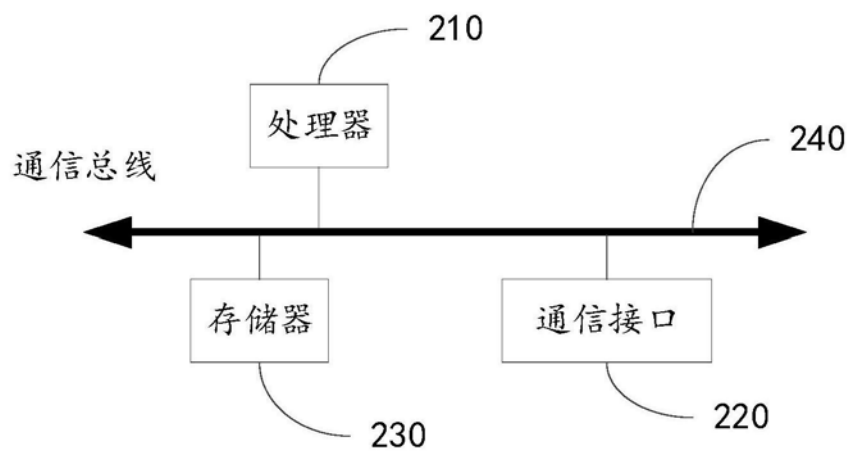


图3