(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113296918 A
(43)申请公布日 2021.08.24

(21)申请号 202110847670.7

(22)申请日 2021.07.27

(71)申请人 北京大学
地址 100871 北京市海淀区颐和园路5号

(72)发明人 崔斌 黎洋 沈彧 江淮钧 刘子瑞

(74)专利代理机构 北京路浩知识产权代理有限公司 11002
代理人 王庆龙

(51)Int.Cl.
G06F 9/48(2006.01)
G06F 9/50(2006.01)
G06N 20/00(2019.01)

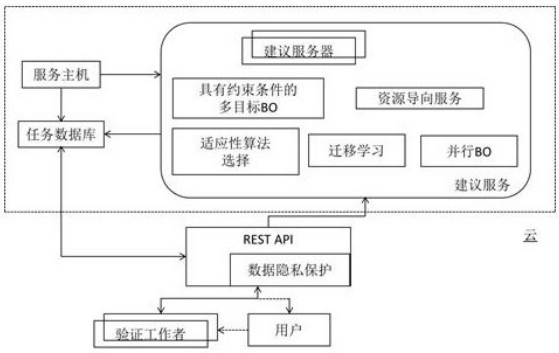
权利要求书2页 说明书9页 附图6页

(54)发明名称

求解通用黑盒优化问题的计算机系统、方法及电子设备

(57)摘要

本发明涉及一种求解通用黑盒优化问题的计算机系统、方法及电子设备。该方法包括：由用户向服务主机提交优化任务；由所述服务主机根据负载均衡分配建议服务器，将任务信息与所述建议服务器绑定；以及由所述用户提供的验证工作者与所述建议服务器交互，持续从所述建议服务器拉取新配置，进行验证，并将结果更新至所述建议服务器，直至所述优化任务结束。本发明通过实现分布式黑盒优化的服务架构、基于局部惩罚机制的分布式并行框架以及基于验证历史的迁移学习方法，提供了一种分布式的、高容错的、可扩展的、高效的系统，获得的性能和效率优于现有系统。



CN 113296918 A

1.一种求解通用黑盒优化问题的方法,其特征在于,所述方法包括:

由用户向服务主机提交优化任务;

由所述服务主机根据负载均衡分配建议服务器,将任务信息与所述建议服务器绑定;以及

由所述用户提供的验证工作者与所述建议服务器交互,持续从所述建议服务器拉取新配置,进行验证,并将结果更新至所述建议服务器,直至所述优化任务结束,

其中,利用所述服务主机进行节点管理、负载均衡和错误恢复,

利用任务数据库保存所有任务的状态,

利用所述建议服务器为每个任务生成新配置,

利用由所述用户提供和拥有的验证工作者,执行配置的验证,

利用REST API在所述用户/验证工作者和所述建议服务器之间建立连接。

2.根据权利要求1所述的求解通用黑盒优化问题的方法,其特征在于,所述REST API包括:

注册接口,用于接受一个工作者调用创建任务时被创建的全局任务标识符,所述标识符绑定现在的验证工作者和相应的任务;

建议接口,用于基于对当前任务的历史观察,给出下一个待验证的超参数配置;

更新接口,用于根据对现有工作者的观察,更新优化的历史,其中对所述现有工作者的观察包括目标值、约束条件的结果和验证信息;

早停接口,用于返回一个布尔变量,所述布尔变量表示现在的验证过程是否需要早停;

推断接口,用于通过使用性能资源外推,交互式地给用户提供资源配置方面的建议。

3.根据权利要求1所述的求解通用黑盒优化问题的方法,其特征在于,所述方法还包括:所述建议服务器根据自动化算法来生成新配置,并使用基于局部惩罚的并行化机制和迁移学习机制来提高采样效率。

4.根据权利要求3所述的求解通用黑盒优化问题的方法,其特征在于,所述自动化算法包括:根据传入任务的特征自适应选择的算法和设置。

5.根据权利要求3所述的求解通用黑盒优化问题的方法,其特征在于,所述基于局部惩罚的并行化机制包括同步并行模式和异步并行模式。

6.根据权利要求5所述的求解通用黑盒优化问题的方法,其特征在于,所述基于局部惩罚的并行化机制包括:

使用中位数插补方法,将串行算法扩展到并行运行,

其中,对于正在验证中的配置,将正在验证中的配置的性能结果设置为已有验证结果的中位数,加入验证历史;以及

所述串行推荐算法根据所述验证历史推荐新的待验证配置。

7.根据权利要求3所述的求解通用黑盒优化问题的方法,其特征在于,所述迁移学习机制包括:

对于每个优化目标,基于历史,对每个历史任务都训练一个代理模型;

对当前任务,利用当前验证历史再次训练另一个代理模型;

给予每个代理模型对应的权重,按照权重将各个模型集成为一个集成代理模型;

利用集成代理模型指导对配置空间的搜索,从而给出新的待验证配置推荐。

8.根据权利要求7所述的求解通用黑盒优化问题的方法,其特征在于,所述给予每个代理模型对应的权重,按照权重将各个模型集成为一个集成代理模型,包括:

对当前模型与所有历史模型使用gPoE方法按权重进行集成,得到集成代理模型,

其中,模型的权重使用RGPE方法,并根据历史任务与当前任务的相似性计算而得出。

9.一种求解通用黑盒优化问题的计算机系统,其特征在于,包括:

服务主机,用于节点管理、负载均衡和错误恢复;

任务数据库,用于保存所有任务的状态;

建议服务器,用于为每个任务生成新配置;

验证工作者,由用户提供和拥有,用于执行配置的验证;以及

REST API,用于在所述用户/验证工作者和所述建议服务器之间建立连接,

其中,在求解通用黑盒优化问题时,所述计算机系统执行以下操作:

由所述用户向所述服务主机提交优化任务;

由所述服务主机根据负载均衡分配所述建议服务器,将任务信息与所述建议服务器绑定;以及

由所述用户提供的验证工作者与所述建议服务器交互,持续从所述建议服务器拉取新配置,进行验证,并将结果更新至所述建议服务器,直至所述优化任务结束。

10.一种电子设备,其特征在于,包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现如权利要求1-8中任一项所述求解通用黑盒优化问题的方法的步骤。

求解通用黑盒优化问题的计算机系统、方法及电子设备

技术领域

[0001] 本发明涉及机器学习领域，更具体地，涉及一种求解通用黑盒优化问题的计算机系统、方法及电子设备。

背景技术

[0002] 黑盒优化的目标是在给定有限的验证资源预算的情况下优化目标函数。这里的“黑盒”是指目标函数不可解析，即目标函数的导数等信息无法获得。由于对目标函数进行验证通常是代价高昂的，黑盒优化的目标是尽可能快地找到一个配置，使得这个配置下的目标函数值接近全局最优。

[0003] 传统的单目标黑盒优化有广泛的应用场景，包括自动化A/B测试，自动化实验设计，数据库参数调优，机器学习超参数调优等。近年来，一些领域对黑盒优化提出了更高的要求，通常表现为要求同时支持更多的目标函数以及满足特定的约束条件。例如自动化建筑领域要求寻找最好的建筑设计以同时保证最低的能源消耗以及建筑成本(多个目标)，自动机器学习云服务要求在满足用户的性能要求的前提下满足公平性保证(约束条件)。以上场景往往需要黑盒优化服务具备那些传统解决方案所不支持的更泛化通用的功能，即多目标和约束条件。

[0004] 现有的黑盒优化库或平台在被应用到现实场景中时往往有以下缺点：(1)有限的应用范围。受底层算法的限制，绝大多数现有的黑盒优化平台无法以统一的方式处理各种优化问题。例如，Hyperopt、SMAC3和HpBandSter只能处理没有约束条件的优化问题。BoTorch和GPflowOpt虽然能够解决具有多目标的或约束条件的广义优化问题，但它们只支持具有连续值参数的优化问题，这很大程度限制了它们的应用范围。(2)有限的扩展性和效率。大多数现有的软件包以串行的方式执行优化过程，在大规模任务上效率低下，对任务规模扩展性差。此外，大多数现有系统也无法利用过去任务的先验知识来加速在相似的任务上的优化。

发明内容

[0005] 针对现有技术中的问题，本发明提供一种求解通用黑盒优化问题的计算机系统、方法及电子设备。

[0006] 第一方面，本发明提供一种求解通用黑盒优化问题的方法，包括：

由用户向服务主机提交优化任务；

由所述服务主机根据负载均衡分配建议服务器，将任务信息与所述建议服务器绑定；以及

由所述用户提供的验证工作者与所述建议服务器交互，持续从所述建议服务器拉取新配置，进行验证，并将结果更新至所述建议服务器，直至所述优化任务结束，

其中，利用所述服务主机进行节点管理、负载均衡和错误恢复，

利用任务数据库保存所有任务的状态，

利用所述建议服务器为每个任务生成新配置，

利用由所述用户提供和拥有的验证工作者，执行配置的验证，

利用REST API在所述用户/验证工作者和所述建议服务器之间建立连接。

[0007] 进一步地，所述REST API包括：

注册接口，用于接受一个工作者调用创建任务时被创建的全局任务标识符，所述标识符绑定现在的验证工作者和相应的任务；

建议接口，用于基于对当前任务的历史观察，给出下一个待验证的超参数配置；

更新接口，用于根据对现有工作者的观察，更新优化的历史，其中对所述现有工作者的观察包括目标值、约束条件的结果和验证信息；

早停接口，用于返回一个布尔变量，所述布尔变量表示现在的验证过程是否需要早停；

推断接口，用于通过使用性能资源外推，交互式地给用户提供资源配置方面的建议。

[0008] 进一步地，所述方法还包括：所述建议服务器根据自动化算法来生成新配置，并使用基于局部惩罚的并行化机制和迁移学习机制来提高采样效率。

[0009] 进一步地，所述自动化算法包括：根据传入任务的特征自适应选择的算法和设置。

[0010] 进一步地，所述基于局部惩罚的并行化机制包括同步并行模式和异步并行模式。

[0011] 进一步地，所述基于局部惩罚的并行化机制包括：

使用中位数插补方法，将串行算法扩展到并行运行，

其中，对于正在验证中的配置，将正在验证中的配置的性能结果设置为已有验证结果的中位数，加入验证历史；以及

所述串行推荐算法根据所述验证历史推荐新的待验证配置。

[0012] 进一步地，所述迁移学习机制包括：

对于每个优化目标，基于历史，对每个历史任务都训练一个代理模型；

对当前任务，利用当前验证历史再次训练另一个代理模型；

给予每个代理模型对应的权重，按照权重将各个模型集成为一个集成代理模型；

利用集成代理模型指导对配置空间的搜索，从而给出新的待验证配置推荐。

[0013] 进一步地，所述给予每个代理模型对应的权重，按照权重将各个模型集成为一个集成代理模型，包括：

对当前模型与所有历史模型使用gPoE方法按权重进行集成，得到集成代理模型，

其中，模型的权重使用RGPE方法，并根据历史任务与当前任务的相似性计算而得出。

[0014] 第二方面，本发明提供一种求解通用黑盒优化问题的计算机系统，包括：

服务主机，用于节点管理、负载均衡和错误恢复；

任务数据库，用于保存所有任务的状态；

建议服务器，用于为每个任务生成新配置；

验证工作者，由用户提供和拥有，用于执行配置的验证；以及

REST API，用于在所述用户/验证工作者和所述建议服务器之间建立连接，

其中，在求解通用黑盒优化问题时，所述计算机系统执行以下操作：

由所述用户向所述服务主机提交优化任务；

由所述服务主机根据负载均衡分配所述建议服务器，将任务信息与所述建议服务器绑定；以及

由所述用户提供的验证工作者与所述建议服务器交互，持续从所述建议服务器拉取新配置，进行验证，并将结果更新至所述建议服务器，直至所述优化任务结束。

[0015] 第三方面，本发明还提供一种电子设备，包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器执行所述计算机程序时实现如第一方面中任一项所述求解通用黑盒优化问题的方法的步骤。

[0016] 本发明通过实现分布式黑盒优化的服务架构、基于局部惩罚机制的分布式并行框架以及基于验证历史的迁移学习方法，提供了一种分布式的、高容错的、可扩展的、高效的系统，获得的性能和效率优于现有系统。

附图说明

[0017] 图1为本发明实施例提供的黑盒优化系统架构图；

图2(a)和图2(b)分别为本发明实施例提供的同步并行机制和异步并行机制的示意图；

图3为单目标优化问题中优化32d-Ackley函数的实验结果；

图4为带约束条件的单目标优化问题中优化10d-Keane函数的实验结果；

图5为多目标优化问题中优化有两个目标的3d-ZDT2函数的实验结果；

图6为带约束条件的多目标优化问题中优化有两个目标的2d-SRN函数的实验结果；

图7为机器学习超参数优化问题中优化LightGBM模型的实验结果；

图8为机器学习超参数优化问题中优化LibSVM模型的实验结果；

图9为迁移学习的实验结果；以及

图10为电子设备的结构示意图。

具体实施方式

[0018] 为了更清楚地说明本发明或现有技术中的技术方案，以下将对实施例或现有技术描述中所需要使用的附图作一简单地介绍，显而易见地，以下描述中的附图是本发明的一些实施例，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的附图。

[0019] 为了使本发明的描述更清楚，对本发明涉及的术语进行了解释，如下：

配置：模型的一组超参数值。是一个从给定的搜索空间X中采样的向量x。x中的每一个维度都是模型一个超参数在它可行范围内的一个取值。

[0020] 测试：在一个配置x下，对模型进行的一次验证。其有三种状态：已准备好、正在运行、已完成。测试完成后，可以获得其对应的验证结果f(x)。

[0021] 任务：定义在一个搜索空间X上的黑盒优化问题。任务的类型是由任务目标和约束条件所定义的。

[0022] 工作者：执行任务的一个节点。

[0023] 代理模型:由于无法使用通过实际验证以外的方式得到一个配置的真实结果,为了降低对实际验证的依赖,贝叶斯优化使用代理模型拟合观察结果,即使用代理模型预测一组配置的真实结果。

[0024] 采集函数:贝叶斯优化针对代理模型定义一个采集函数,每一轮推荐能够使得采集函数最大的配置。

[0025] 贝叶斯优化:一种被广泛使用的优化算法。通过使用代理模型拟合现有的观察结果进行拟合,并通过优化采集函数的方式推荐需要实际验证的配置。

[0026] 本发明的目的在于提供一种求解通用黑盒优化问题的服务框架,具有以下性质:1)简单易用。本发明提供了用户友好的可视化界面来追踪和管理黑盒优化任务,旨在最小化用户的交互操作;2)性能优异。本发明包含了现有主流优化算法,并能够自动选择合适的优化算法;3)合适地管理资源。本发明能够为用户预估模型带来的代价,帮助用户最小化模型验证的资源开销;4)可灵活扩展。本发明可以针对输入变量的维度、优化目标数、任务数、测试数和并行度进行灵活扩展;5)高效率。本发明可高效地利用并行资源,同时使用迁移学习和多精度进行了系统优化;6)支持数据隐私保护。

[0027] 图1示出了根据本发明的分布式黑盒优化的服务架构。参照图1,架构中包含五个主要组件:

服务主机,负责节点管理、负载均衡和错误恢复;

任务数据库,负责保存所有任务的状态;

建议服务器,负责为每个任务生成新配置;

REST API,负责在用户/验证工作者(下文中称工作者)和建议服务器之间建立了连接;

验证工作者,由用户提供和拥有,负责执行配置的验证。

[0028] 本发明提供的服务架构主要运行流程如下:

(1)用户向服务主机提交优化任务;

(2)服务主机根据负载均衡分配建议服务器,将任务信息与建议服务器绑定;以及

(3)用户提供的工作者与建议服务器交互,持续从建议服务器拉取新配置,进行验证,并将结果更新至建议服务器,直至优化任务结束。

[0029] 本发明为支持大量优化任务同时进行,需要同时为多个任务生成推荐配置的建议。这种强度的工作量单个机器无法承受的。因此,本发明将建议服务器在多台机器上部署,这些建议服务器组成了一个可大规模扩展的基础结构。本发明的另一个主要组件是服务主机。服务主机负责管理建议服务器并平衡工作负载,其作为统一的端点,接受工作者的请求。因此,每个工作者就无需了解具体的调度细节。服务主机可以根据任务绑定关系将请求转交给对应的建议服务器,建议服务器根据自动算法选择模块确定的算法(将在稍后进行描述)来生成新配置,并在这个过程中使用基于局部惩罚的并行化机制和迁移学习框架来提高采样效率(将在稍后进行描述)。用户/工作者和系统交互均通过REST API进行,任务状态与任务运行过程中的优化历史数据记录在任务数据库中。

[0030] 由于服务运行中机器崩溃错误难以避免,本发明的服务架构包含错误恢复机制。服务主机通过活动服务器列表记录并监控每个建议服务器的状态。如果一台建议服务器崩溃或被人为关闭,在该建议服务器上的任务将被服务主机调度到一个新的建议服务器,新

的建议服务器利用存储在任务数据库中的相关优化历史记录继续执行优化任务。另外，服务主机的快照也存储在远程数据库中，如果服务主机崩溃，服务架构将重启节点并从数据库中恢复快照。

[0031] 接下来，将对建议服务器生成配置时所采取的算法进行说明。

[0032] 在本发明中，集成了一系列优化算法，在各种黑盒优化问题中都实现了很高的性能。现有的黑盒优化框架对各个任务都使用相同的算法，对每个算法也都使用相同的设置。与现有技术不同，本发明根据传入任务的特征自适应地选择合适的算法和设置。例如，对于单目标优化问题，本发明使用作为一种优选方案的EI作为采集函数。对于多目标优化问题，当目标数小于5时，本发明使用作为一种优选方案的EHVI作为采集函数；对于目标数较多的优化问题，本发明使用作为一种优选方案的MESMO作为采集函数。此外，本发明还会根据配置空间和试验次数自动选择贝叶斯优化中的代理模型：如果输入空间是有条件限制的，比如要求一个参数必须小于另一个参数，或要求试验次数超过500，本发明提出使用概率随机森林作为代理模型，以避免传统方法中，用高斯过程作为代理模型所产生的不相容性和高计算复杂度。在其它情况下，本发明使用高斯过程作为代理模型。此外，当搜索空间只包含浮点参数和整数参数时，本发明使用L-BFGS-B算法优化采集函数；当某些参数不是数值类型的参数时，本发明使用局部与全局相结合的算法优化采集函数。

[0033] 接下来，将对基于局部惩罚机制的分布式并行框架进行说明。

[0034] 为充分利用并行资源，本发明提供了一种分布式的并行执行机制，支持多个工作者同时验证多组配置。本发明支持以下两种并行验证方式(如图2(a)和图2(b)所示)：

同步并行：其中，各个工作者从建议服务器请求待验证的配置。各自对配置进行验证。当所有工作者都验证完成后，再开始下一轮验证。

[0035] 异步并行：其中，每个工作者只要完成了当前的验证，就立刻从建议服务器请求新的待验证的配置，进行下一轮验证。

[0036] 并行验证时，建议服务器根据工作者需求，使用并行推荐算法推荐一个或多个新的待验证的配置。在并行推荐算法的设计中，需要考虑正在验证但还未得到验证结果的配置，避免推荐的配置一直与验证中的配置相同或相似，使得多个工作者验证相同或相似的配置而浪费资源。本发明提出了一个无关算法本身的机制，可以将任意串行推荐算法扩展到并行运行，而无需为每个算法单独设计并行版本。作为一种优选方案，本发明使用基于局部惩罚机制的中位数插补方法，将串行算法扩展到并行运行。对于正在验证中的配置，中位数插补方法将正在验证中的配置的性能结果设置为已有验证结果的中位数，加入验证历史，串行推荐算法根据该验证历史推荐新的待验证配置。使用该方法避免工作者验证相似的配置，增大了推荐算法对配置空间的探索性。

[0037] 以下示出了针对两种并行方式使用中位数插补方法进行配置推荐的具体流程。

[0038] 对于同步并行方式，流程如下：

(1)给定验证历史D以及这一轮需要推荐的配置数n；

(2)将验证历史D拷贝为D'；

(3)计算验证历史D'中性能结果的中位数y*；

(4)i在1到n中循环：

(5)依据验证历史D'，使用串行推荐算法，得到配置推荐x_i；

(6)将(x_i,y*)加入验证历史D';

(7)返回所有在步骤(5)中得到的配置推荐x_1到x_n。

[0039] 对于异步并行方式,流程如下:

(1)给定验证历史D和正在验证中的配置x_1到x_k(共k个);

(2)将验证历史D拷贝为D';

(3)计算验证历史D'中性能结果的中位数y*;

(4)i在1到k中循环:

(5)将(x_i,y*)加入验证历史D';

(6)依据验证历史D',使用串行推荐算法,得到配置推荐x,返回x。

[0040] 接下来,将对基于验证历史的迁移学习方法进行说明。

[0041] 在执行黑盒优化任务时,用户经常会运行与之前类似的任务。基于此观察,本发明引入一种通用的迁移学习框架,用来加速当前任务的执行。本发明的迁移学习框架具有以下优点:支持通用的黑盒优化问题,且与大多数贝叶斯优化方法兼容。

[0042] 对于有p个优化目标的多目标优化问题,本发明分别对关于这p个目标的知识进行迁移,从而将这种多目标的迁移学习任务转换成p个单目标的迁移学习任务。

[0043] 对于每个目标,首先基于历史,对每个历史任务T_1到T_n都训练一个代理模型M_1到M_n。然后,对当前任务,利用当前验证历史再训练一个代理模型M_{n+1}。给予每个代理模型对应的权重w_1到w_{n+1},按照权重使用预定方法将各个模型集成为一个代理模型M*。利用M*指导对配置空间的搜索,给出新的待验证配置推荐。作为一种优选方案,本发明使用RGPE方法,计算历史任务模型预测当前任务验证历史产生的逆序对比例,进而计算权重,从而反映不同历史任务和当前任务之间的相似性;使用gPoE方法,根据模型权重对多个模型进行集成,从而自动排除不可信模型对集成模型预测的影响。

[0044] 本发明还可以合适地管理资源,例如指导用户如何配置验证资源,以最小化工作者数量和时间开销。本发明使用加权成本模型来推断性能曲线与验证曲线,以多个函数族作为基础模型,采用马尔可夫-蒙特卡洛方法对模型参数进行估计。本发明根据现有的观察结果建立一个成本模型,并用其预测接近最优的试验次数,根据此预测指导用户配置验证资源。

[0045] 在下文中,为了更详细地描述黑盒优化服务框架,提供了黑盒优化任务描述语言以及工作者工作流的示例。

[0046] 任务描述语言:为方便用户,本发明设计了一个用来定义优化任务的任务描述语言(TDL)。此任务描述语言的核心是定义搜索空间。这包括每个参数的类型、取值范围以及各个参数之间的关系。本发明中支持的类型有浮点型(float)、整数型(integer)、序数型(ordinal)和类别型(categorical)。此外,用户还可以对各个超参数添加限制条件,限制其搜索空间。用户还可以在任务描述语言中指定时间预算、任务类型、工作者个数、并行策略和历史记录。下面给出一个任务描述语言的示例:

```
task_config = {
“parameter”: {
“x1”: {“type”: “float”, “default”: 0, “bound”: [-5, 10]},
“x2”: {“type”: “integer”, “bound”: [0, 15]},
```

```
        "x3": {"type": "categorical", "default": "a1", "choice": ["a1", "a2",
"a3"]},
        "x4": {"type": "ordinal", "default": 1, "choice": [1, 2, 3]}
        },
        "condition": {
        "cdn1": {"type": "equal", "parent": "x3", "child": "x1", "value":
"a3"}
        },
        "number_of_trials": 200,
        "time_budget": 10800,
        "task_type": "soc",
        "parallel_strategy": "async",
        "worker_num": 10,
        "use_history": True
        }
```

以上示例定义了四个不同类型的参数x1-4;一个限制条件cdn1,其含义是,x1仅在x3="a3"时是活跃的;验证次数为200;时间预算为10800秒;任务类型为单目标带约束条件优化("soc");并行策略为异步;工作者数量为10;启用迁移学习。

[0047] 基本工作流:给定了任务描述语言后,本发明的工作者基本工作流程如下:

```
global_task_id ← worker.CreateTask(task_TDL) // 根据任务描述语言创建
任务,注册工作者并获得全局任务标识符
worker.BindTask(global_task_id) // 将工作者与全局任务标识符绑定
while not worker.TaskFinished(): // 在任务完成前循环
config ← worker.GetSuggestions() // 调用推荐接口,获得待验证的配置
result ← Evaluate(config) // 在目标函数上验证配置,得到结果
Worker.UpdateObservations(config, result) // 将验证结果更新到服务器上
```

其中,Evaluate指用户对所提供的目标函数的验证过程。通过调用创建任务(CreateTask)接口,辅助进程可以获得一个唯一的全局任务标识符(global_task_id)。使用相同全局任务标识符的所有辅助进程都会链接到相同的任务,进行并行计算。任务尚未完成时,工作者将持续调用获得配置推荐(GetSuggestions)接口和更新结果(UpdateObservations)接口,从建议服务器中提取建议并更新其相应的观察结果。

[0048] 系统交互接口:在本发明的设计中,用户可以通过REST API和系统进行交互。其中,最重要的服务调用接口有:

1)注册(Register)接口。本接口接受一个工作者调用创建任务时被创建的全局任务标识符,这个标识符绑定现在的工作者和相应的任务。这样,多个工作者之间可以共享优化后的历史记录。

[0049] 2)建议(Suggest)接口。本接口基于对当前任务的历史观察,给出下一个要验证的超参数配置。

[0050] 3)更新(Update)接口。本接口根据对现有工作者的观察,更新优化的历史。对现有

工作者的观察包括三部分:目标值、约束条件的结果和验证信息。

[0051] 4)早停(StopEarly)接口。本接口返回一个布尔变量,表示现在的验证过程是否需要早停。

[0052] 5)推断(Extrapolate)接口。本接口通过使用性能资源外推,交互式地给用户提供资源配置方面的建议。

[0053] 为了验证本发明在解决黑盒优化问题方面的效果,根据实验配置和测评指标对实验结果进行了分析,具体如下:

(1)实验配置和测评指标

以下示出本发明在解决各种黑盒优化问题上的实验结果。对于单目标优化问题,示出优化32d-Ackley函数的实验结果。对于带约束条件的单目标优化问题,示出10d-Keane函数的实验结果。对于多目标的优化问题,示出优化有两个目标的3d-ZDT2函数的结果。对于带约束条件的多目标优化问题,示出有两个目标的2d-SRN函数的结果。在这两个数学问题中,参数类型均为浮点型。最大试验次数根据问题的难度在80到500之间变化。

[0054] 此外,实验部分也示出了本发明在机器学习超参数优化任务上的性能。实验使用25个分类数据集,样本量范围为1千到10万。数据集被切分成三份,实验使用训练集进行训练,并在验证集上进行优化,最终汇报在测试集上的排名。实验使用对LightGBM模型以及使用线性核的LibSVM模型进行超参数优化,其中LightGBM模型的超参数类型都为浮点型,LibSVM模型中还含有类别型超参数和带条件限制的超参数。

[0055] 最后,实验部分示出了本发明迁移学习方法的性能。对比了提供迁移学习支持的Vizier框架,并以不使用迁移学习的SMAC3框架作为基准方法。在25个OpenML数据集上,使用“留一法”进行实验。即每次选择一个数据集,使用在其它所有数据集上的优化历史作为先验知识进行迁移学习,指导优化算法在当前数据集上对LightGBM模型的超参数优化过程。

[0056] 针对数学函数使用的测评指标如下:

最优间隔(optimality gap):在单目标优化问题中,使用最优间隔衡量不同方法在单目标优化问题上的表现。即如果f在x*取最优值,x’是当前找到的最优配置,则当前最优间隔的值是|f(x*)-f(x’)|。

[0057] 超体积差(hypervolume difference):在多目标优化问题中,使用超体积差衡量不同方法在多目标优化问题上的表现。给定参考点r,超体积HV(P,r)计算帕累托前沿P与参考点r在超空间中围成的凹多面体体积。如果理想的帕累托前沿为P*,当前优化方法找到的帕累托前沿为P’,则超体积差为HV(P*,r)-HV(P’,r)。超体积差越小,代表优化算法越好。

[0058] (2)具体实验结果

在以下的结果中,OpenBox代表本发明。

[0059] 图3示出了单目标优化问题中,优化32d-Ackley函数的实验结果。结果表明,在8种参与比较的优化方法中,只有本发明能够持续稳定地优化这个目标函数。这印证了本发明可以适应于各种各样的输入维度。注意到,本发明相对实现了比其它方法能提供10倍以上的加速比。

[0060] 图4示出了带约束条件的单目标优化问题中,优化10d-Keane函数的实验结果。对比了目前三种最主流的支持约束条件的软件包。结果表明,本发明的收敛结果远远好于所

有对比方法。在10维的Keane问题中，真实的最优配置很难获得，本发明是五种方法中唯一能成功优化这个函数的方法。

[0061] 图5示出了多目标优化问题中，优化有两个目标的3d-ZDT2函数的实验结果。对比了目前三种最主流的支持约束的多目标优化的软件包。结果表明，随着试验次数增加，GPflowOpt和Hypermapper的超体积差减少缓慢；而BoTorch和OpenBox在50次试验内就获得了令人满意的实验结果。

[0062] 图6示出了带约束条件的多目标优化问题中，优化有两个目标的2d-SRN函数的实验结果。结果表明，在试验次数达到30次以上时，BoTorch和Hypermapper便不再能提供更好的推荐配置点。本发明在解决复杂的带约束条件的多目标优化问题上，效果远好于BoTorch和Hypermapper。

[0063] 图7示出了机器学习超参数优化任务中，对LightGBM模型优化的实验结果，以箱型图示出了各方法在25个数据集上优化最终结果的排名。结果表明，本发明对LightGBM模型执行超参数优化的效果平均排名位列第一，且效果稳定。

[0064] 图8示出了机器学习超参数优化任务中，对LibSVM模型优化的实验结果，以箱型图示出了各方法在25个数据集上优化最终结果的排名。实验对比了目前两种支持类别型超参数的软件包。结果表明，本发明对LightGBM模型执行超参数优化的效果平均排名位列第一，且效果稳定。

[0065] 图9示出了迁移学习相关的实验结果。实验结果表明，相比于不使用迁移学习的SMAC3，本发明和Vizier实现了更好的采样效率；同Vizier的相比，本发明的迁移学习功能表现更佳。此外，本发明的迁移学习方法支持通用的黑盒优化问题，而Vizier则不能支持。

[0066] 本发明通过实现分布式黑盒优化的服务架构、基于局部惩罚机制的分布式并行框架以及基于验证历史的迁移学习方法，提供了一种分布式的、高容错的、可扩展的、高效的系统，获得的性能和效率优于现有系统。

[0067] 另一方面，本发明提供了一种电子设备。如图10所示，电子设备1000包括处理器1001、存储器1002、通信接口1003和通信总线1004。

[0068] 其中，处理器1001、存储器1002、通信接口1003通过通信总线1004完成相互间的通信。

[0069] 处理器1001用于调用存储器1002中的计算机程序，处理器1001执行计算机程序时实现如上所述的本发明实施例所提供的求解通用黑盒优化问题的步骤。

[0070] 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实施例方案的目的。本领域普通技术人员在不付出创造性的劳动的情况下，即可以理解并实施。

[0071] 最后应说明的是：以上实施例仅用于说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

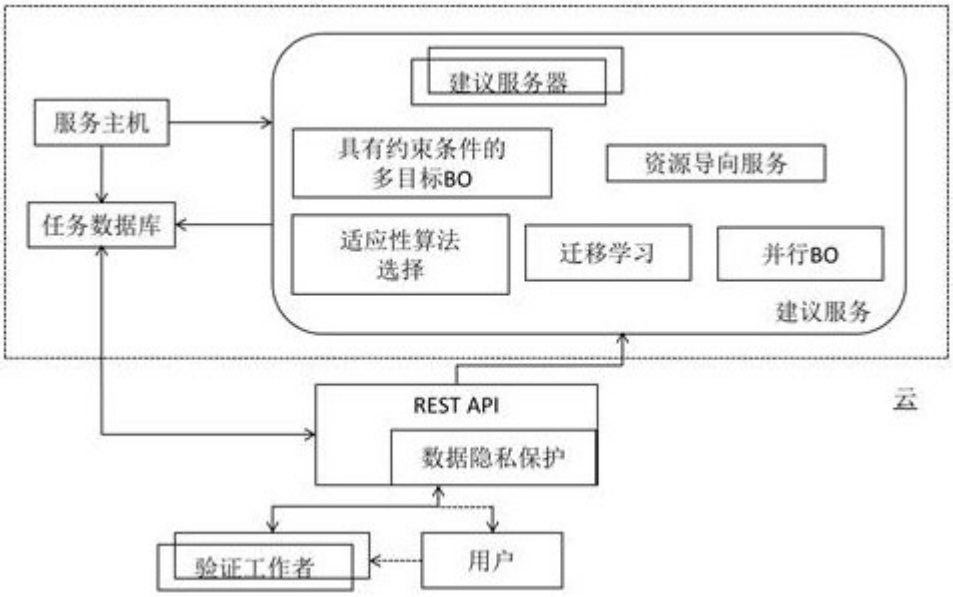

建议服务器
服务主机
具有约束条件的
多目标BO
资源导向服务
任务数据库
适应性算法
选择
迁移学习
并行BO
建议服务
云
REST API
数据隐私保护
验证工作者
用户

图1

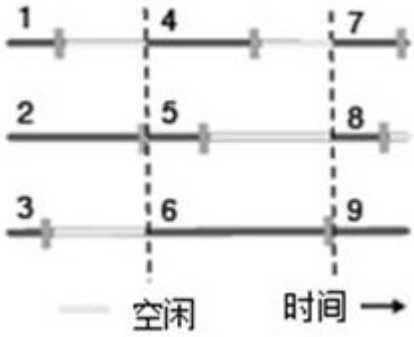

1
4
7
2
5
8
3
6
9
空闲
时间

图2(a)

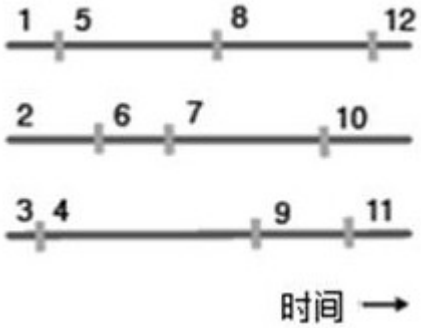

1
5
8
12
2
6
7
10
3
4
9
11
时间

图2(b)

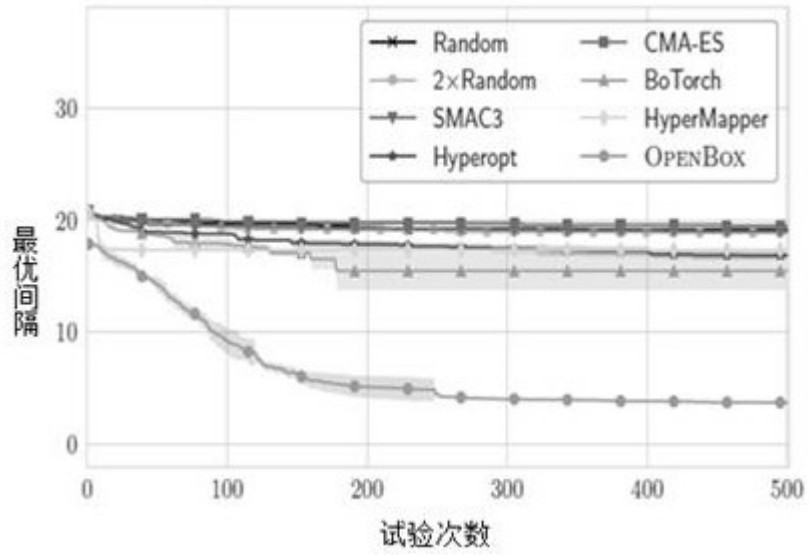

Random
2×Random
SMAC3
Hyperopt
CMA-ES
BoTorch
HyperMapper
OpenBox
最优间隔
30
20
10
0
0
100
200
300
400
500
试验次数

图3

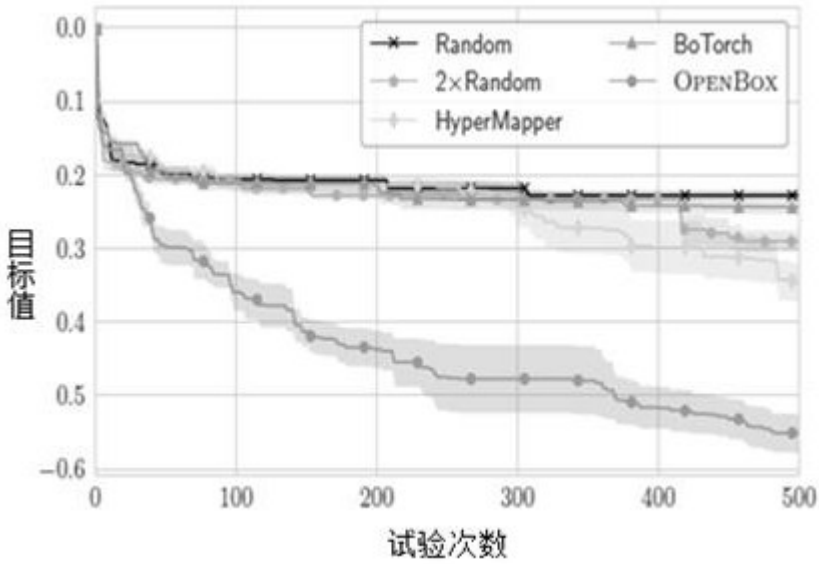

Random
2×Random
HyperMapper
BoTorch
OpenBox
目标值
0.0
0.1
0.2
0.3
0.4
0.5
−0.6
0
100
200
300
400
500
试验次数

图4

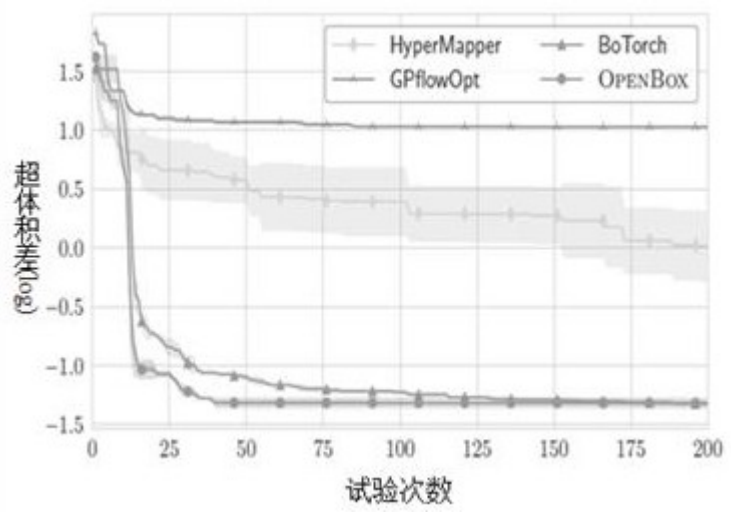

HyperMapper
BoTorch
GPflowOpt
OpenBox
超体积差(log)
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
0
25
50
75
100
125
150
175
200
试验次数

图5

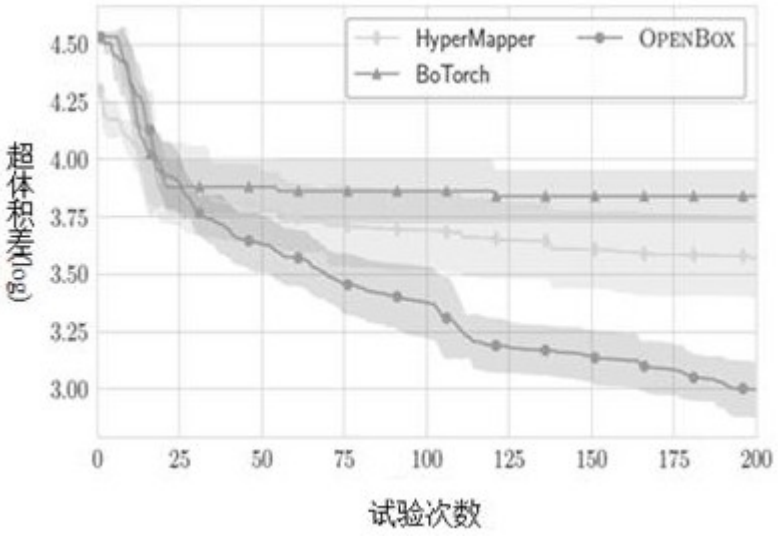

HyperMapper
OpenBox
BoTorch
超体积差(log)
4.50
4.25
4.00
3.75
3.50
3.25
3.00
0
25
50
75
100
125
150
175
200
试验次数

图6

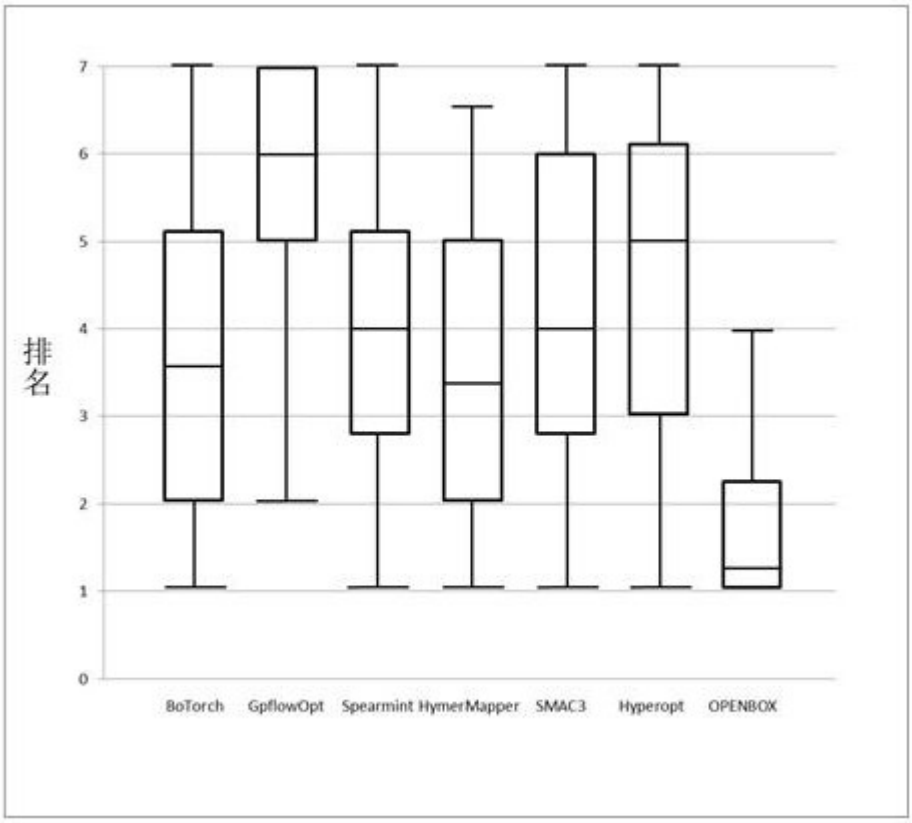

排名
7
6
5
4
3
2
1
0
BoTorch
GpflowOpt
Spearmint
HymerMapper
SMAC3
Hyperopt
OPENBOX

图7

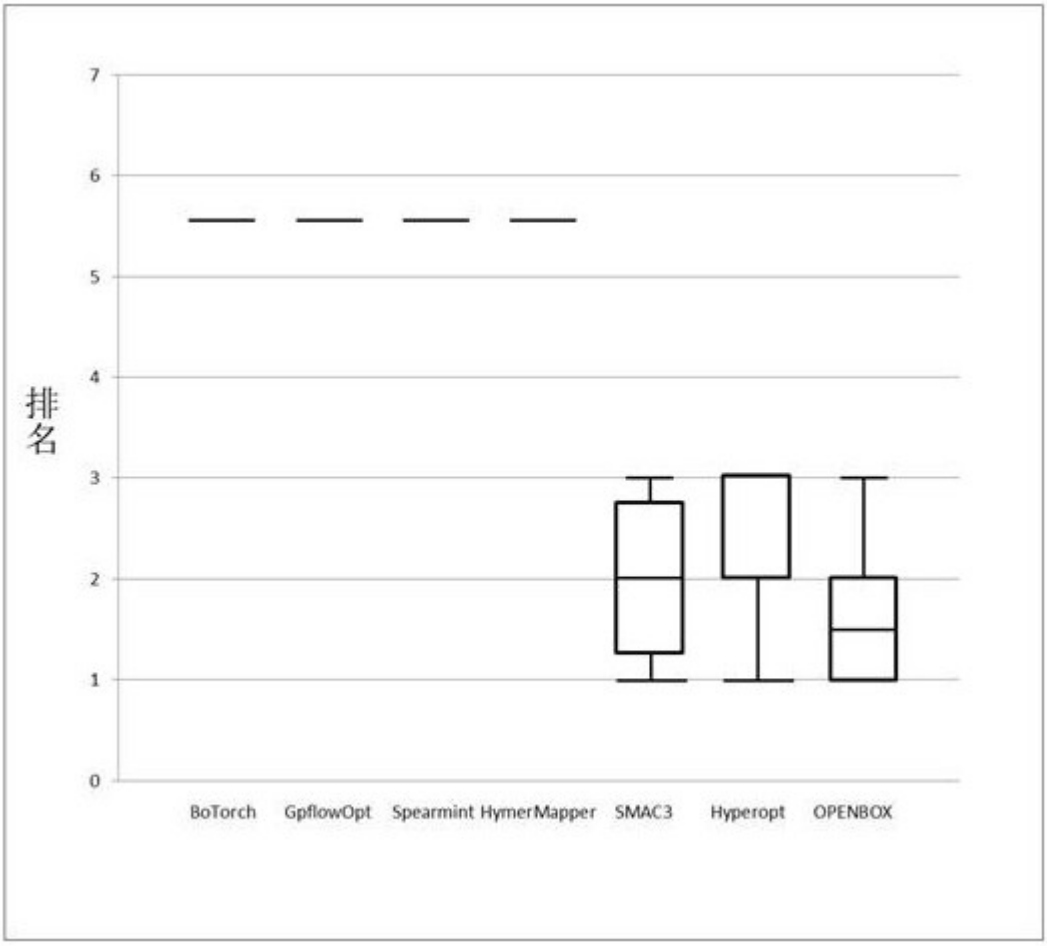

排名
7
6
5
4
3
2
1
0
BoTorch
GpflowOpt
Spearmint
HymerMapper
SMAC3
Hyperopt
OPENBOX

图8

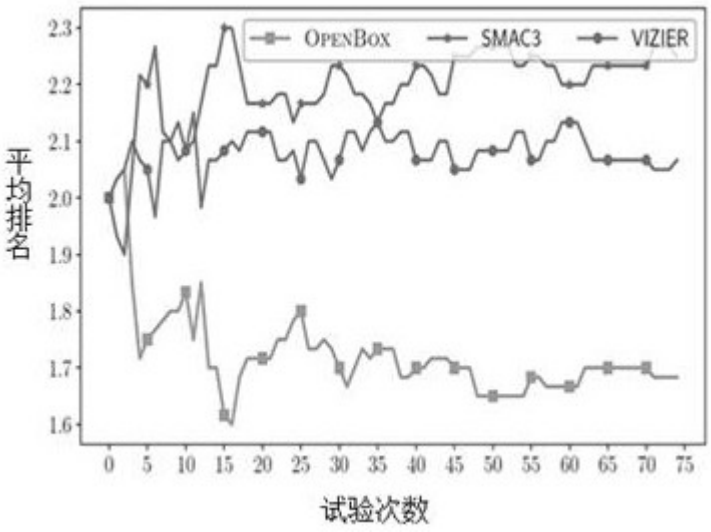

OpenBox
SMAC3
VIZIER
平均排名
2.3
2.2
2.1
2.0
1.9
1.8
1.7
1.6
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75
试验次数

图9

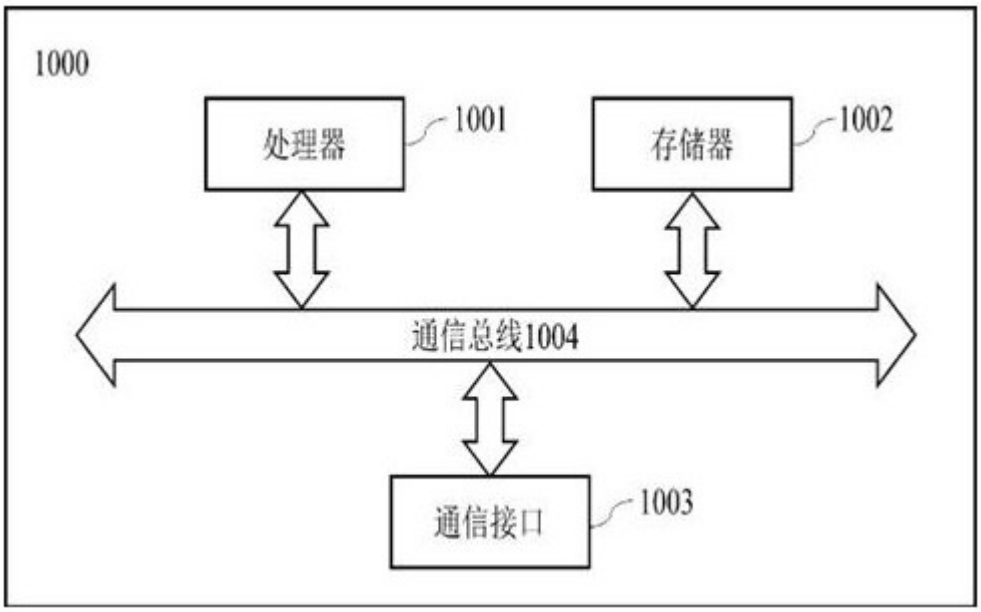


图10