(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111164620 B
(45)授权公告日 2024.08.20

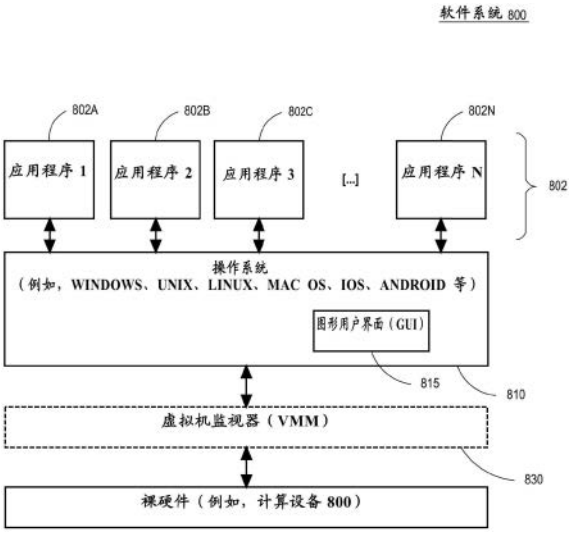
(21)申请号 201880062950.3
(22)申请日 2018.09.28
(65)同一申请的已公布的文献号
申请公布号 CN 111164620 A
(43)申请公布日 2020.05.15
(30)优先权数据
62/565,004 2017.09.28 US
15/884,163 2018.01.30 US
(85)PCT国际申请进入国家阶段日
2020.03.27
(86)PCT国际申请的申请数据
PCT/US2018/053515 2018.09.28
(87)PCT国际申请的公布数据
WO2019/067957 EN 2019.04.04
(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚
(72)发明人 S·阿格尔沃 S·伊蒂库拉
V·瓦拉达拉珍 N·阿格尔沃
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
专利代理师 周衡威
(51)Int.Cl.
G06N 20/20(2006.01)
(56)对比文件
李建平等.仿真元建模中的拟合方法及其应用研究.《中国博士学位论文全文数据库》信息科技辑.2009,第2.1.3节、第2.3.1节、第3.4.3节、第4.3.1节、第5.2.1节、第5.2.2节、表4.1、表4.2、表5.1、表5.2、表5.3、图3.2.
审查员 何江琴
权利要求书2页 说明书16页 附图8页

(54)发明名称

用于自动机器学习模型选择的特定于算法的神经网络体系架构

(57)摘要

提供了基于经训练的特定于算法的回归器的性能预测来选择机器学习算法的技术。在实施例中,计算机通过对于每个元特征从推理数据集中导出相应的元特征值来从推理数据集中导出各元特征值。对于每个可训练的算法和分别与该算法相关联的每个回归元模型,通过基于以下至少一项来调用该元模型来计算相应的分数:元特征值的相应子集,和/或该算法的超参数的相应子集的超参数值。基于相应的分数来选择(一个或多个)算法。基于推理数据集,可以调用所选择的(一个或多个)算法来获得结果。在实施例中,经训练的回归器是不同地配置的人工神经网络。在实施例中,经训练的回归器被包含在特定于算法的集合体中。还提供了对回归器和/或集合体进行最佳训练的技术。



CN 111164620 B

1.一种计算机实现的方法,包括:

通过对于多个元特征中的每个元特征从包含多个数据单元的推理数据集的所述多个数据单元中的多个数据单元中导出所述推理数据集的相应的元特征值,来从推理数据集中导出多个元特征值,其中所述推理数据集包括照片集合,每个元特征值表征作为整体的所述推理数据集;

对于多个可训练的算法中的每个算法:

对于能够预测该算法的性能的相应多个回归元模型中的每个元模型,通过利用所述多个元特征值的相应的元特征值的子集作为输入来调用所述元模型,计算多个元模型分数中所述元模型的相应的分数,其中所述相应的分数指示以下中的至少一者:用所述推理数据集训练的所述算法的准确性,或者用所述推理数据集训练所述算法所需的持续时间;

基于所述多个元模型分数选择所述多个可训练的算法的子集,并且

基于所述推理数据集,训练所述多个可训练的算法的所选择的所述子集,

其中,所述方法还包括:

存储多个测试数据集;

对于所述多个可训练的算法中的每个算法:

对于基于所述算法的多个模型中的每个模型:

基于所述算法的所述多个超参数的相应特定值来配置所述模型;以及

对于所述多个测试数据集中的每个测试数据集:

基于所述测试数据集测试所述模型以计算相应的测试分数;以及

记录不同元组,所述不同元组引用:所述多个超参数的所述相应特定值、所述测试数据集、所述相应的测试分数和所述算法;以及

对于能够预测所述算法的性能的所述相应多个回归元模型中的每个元模型,基于为所述算法记录的所述不同元组中的至少一个元组来训练所述元模型。

2.如权利要求1所述的方法,其中:

所述多个可训练的算法的所述子集包括多个算法;

选择所述多个可训练的算法的所述子集包括基于所述相应的分数对所述多个可训练的算法进行排名。

3.如权利要求1所述的方法,其中,通过调用所述元模型计算所述相应的分数包括:基于所述算法的多个超参数的相应子集中的超参数值来调用所述元模型。

4.如权利要求3所述的方法,其中,所述多个超参数的所述相应子集中的所述超参数值是默认值。

5.如权利要求1所述的方法,其中:

所述多个可训练的算法中的每个算法与相应的集合体相关联,该相应的集合体包含能够预测该算法的性能的所述相应多个回归元模型;

所述方法还包括:对于所述多个可训练的算法中的每个算法,导出相应的集合体分数,该相应的集合体分数基于能够预测该算法的性能的所述相应多个回归元模型的分数;

基于所述相应的分数选择所述多个可训练的算法的所述子集包括:基于所述相应的集合体分数选择所述多个可训练的算法的所述子集。

6.如权利要求1所述的方法,其中,至少以下之一:

所述多个元特征排除缺失所述不同元组的超过阈值的百分比的值的元特征,或者

所述不同元组中的所述至少一个元组排除缺失值的元组。

7.如权利要求1所述的方法,还包括利用原始训练数据集交叉验证所述多个模型,所述原始训练数据集被划分为:多个训练数据集和所述多个测试数据集。

8.如权利要求1所述的方法,还包括并行地生成所述不同元组中的多个元组。

9.如权利要求1所述的方法,其中,所述多个可训练的算法中的每个算法包括以下之一:支持向量机(SVM)、随机森林、决策树或人工神经网络。

10.如权利要求1所述的方法,其中,所述多个可训练的算法中的每个算法包括以下之一:分类、回归或异常检测。

11.如权利要求1所述的方法,其中,能够预测所述算法的性能的所述相应多个回归元模型中的每个元模型包括不同的人工神经网络。

12.如权利要求1所述的方法,其中,计算所述相应的分数包括softmax函数。

13.如权利要求1所述的方法,还包括通过独热或独冷中的至少一个编码方案来转换所述多个元特征中的非数值型元特征的值。

14.如权利要求1所述的方法,还包括通过零均值或单位方差中的至少一个编码方案来转换所述多个元特征中的数值型元特征的值。

15.如权利要求1所述的方法,还包括:对于所述多个可训练的算法中的每个算法,优化以下各项中的至少一项:

作为输入被注入到能够预测所述算法的性能的所述相应多个回归元模型中的每个元模型中的超参数和/或元特征的百分比,或者

能够预测所述算法的性能的所述相应多个回归元模型中的元模型的计数。

16.如权利要求15所述的方法,其中,所述优化包括以下至少之一:梯度下降、贝叶斯优化、SVM或决策树。

17.如权利要求1所述的方法,还包括:通过样本装袋、特征装袋或提升中的至少一项来分配超参数和/或元特征作为能够预测所述算法的性能的所述相应多个回归元模型中的每个元模型的输入。

18.一个或多个存储指令的非暂态计算机可读介质,所述指令在由一个或多个处理器执行时,使得所述一个或多个处理器执行如权利要求1-17中任一项所述的方法。

19.一种计算机系统,包括:

一个或多个处理器;以及

一个或多个存储指令的非暂态计算机可读介质,所述指令在由所述一个或多个处理器执行时,使得所述一个或多个处理器执行如权利要求1-17中任一项所述的方法。

20.一种包括用于执行如权利要求1-17中任一项所述的方法的构件的装置。

用于自动机器学习模型选择的特定于算法的神经网络体系架构

技术领域

[0001] 本公开涉及机器元学习(meta-learning)。本文提出的是基于经训练的特定于算法的回归器(regressor)的性能预测来最佳地选择机器学习算法的技术,以及用于训练回归器和/或回归器的集合体(ensemble)的技术。

背景技术

[0002] 机器学习用于各种应用和领域,诸如医疗保健、物联网(IOT)、金融和安全性。数十年的研究创造了可应用于这些应用的各种算法和技术。为应用选择最佳算法可能是困难的而且是资源密集型的。例如,可以通过几种算法来完成分类任务,诸如支持向量机(SVM)、随机森林、决策树、人工神经网络等。这些算法中的每一个都有许多变体和配置,并且对不同的数据集不同地执行。选择最佳算法通常是由具有多年经验的数据科学家或机器学习专家执行的手动任务。

[0003] 某些自动算法选择技术会导致大量的计算开销,诸如在产品研发(R&D)期间的开销,这可能会延长上市时间。存在数百种机器学习算法。训练和测试每种机器学习算法以找到最佳表现的可能是不可行的。选择性训练的自动方法通常最终使用单个回归器/分类器来预测算法性能,这会导致不同算法在选择模型中相互干扰,从而降低准确度。这些方法也没有考虑算法超参数(hyperparameter),其可以显著影响算法性能和行为。

附图说明

[0004] 在附图中:

[0005] 图1是描绘实施例中的示例计算机的框图,该示例计算机基于经训练的回归器的性能预测来最佳地选择可训练的算法;

[0006] 图2是描绘实施例中的用于基于经训练的回归器的性能预测来最佳地选择可训练的算法的示例处理的流程图;

[0007] 图3是描绘实施例中的示例计算机的框图,该示例计算机针对上下文准确性能预测来训练特定于算法的机器学习集合体;

[0008] 图4是描绘实施例中用于针对上下文准确性能预测来训练特定于算法的机器学习集合体的示例处理的流程图;

[0009] 图5是描绘实施例中的针对一致性、可移植性和通用性来调整特征(超参数和元特征)值的示例计算机的框图;

[0010] 图6是描绘实施例中的示例计算机的框图,该示例计算机通过诸如提升(boosting)和装袋(bagging)之类的优化来改善集合体元学习;

[0011] 图7是图示可以在其上实现本发明的实施例的计算机系统的框图;

[0012] 图8是图示可以用于控制计算系统的操作的基本软件系统的框图。

具体实施方式

[0013] 在下面的描述中,出于解释的目的,阐述了许多具体细节以便提供对本发明的透彻理解。但是,将清楚的是,可以在没有这些具体细节的情况下实践本发明。在其它情况下,以框图形式示出了众所周知的结构和设备,以避免不必要地使本发明模糊。

[0014] 本文根据以下概要描述各实施例:

[0015] 1.0 总体概述

[0016] 2.0 示例计算机

[0017] 2.1 机器学习算法

[0018] 2.2 元模型(meta-model)

[0019] 2.3 超参数(hyperparameter)

[0020] 2.4 元特征(meta-feature)

[0021] 2.5 推理数据集(inference dataset)

[0022] 2.6 推理

[0023] 2.7 分数

[0024] 2.8 结果

[0025] 3.0 示例选择处理

[0026] 3.1 准备

[0027] 3.2 评分

[0028] 3.3 选择

[0029] 4.0 元学习

[0030] 4.1 集合体

[0031] 4.2 集合体分数

[0032] 4.3 Softmax

[0033] 4.4 性能元组(tuple)

[0034] 4.5 模型与元模型

[0035] 4.6 模型测试

[0036] 4.7 元模型训练

[0037] 4.8 特定于竞争算法的集合体

[0038] 4.9 训练数据集分区

[0039] 5.0 示例元学习处理

[0040] 5.1 准备

[0041] 5.2 建模

[0042] 5.3 集合体训练

[0043] 6.0 值转换

[0044] 6.1 原始数据

[0045] 6.2 类别

[0046] 6.3 独热(One-Hot)编码

[0047] 6.4 数字

[0048] 6.5 零均值

[0049] 6.6 单位方差

[0050] 6.7 归一化

[0051] 7.0 元优化

[0052] 7.1 方差

[0053] 7.2 偏差

[0054] 7.3 自举聚合

[0055] 7.4 特征装袋

[0056] 7.5 样本装袋

[0057] 7.6 数据集分区

[0058] 7.7 重叠

[0059] 7.8 提升

[0060] 7.9 集合体优化

[0061] 8.0 硬件概述

[0062] 9.0 软件概述

[0063] 10.0 云计算

[0064] 1.0总体概述

[0065] 本文提供了用于基于经训练的特定于算法的回归器的性能预测来最优地选择机器学习算法的技术。在实施例中,计算机通过对于每个元特征从推理数据集中导出相应的元特征值来从推理数据集中导出元特征值。对于每个可训练的算法和分别与该算法相关联的每个回归元模型,通过基于以下至少一项调用元模型来计算相应分数:a)元特征值的相应子集,和/或b)该算法的超参数的相应子集的超参数值。基于相应的分数选择算法中的一个或多个算法。基于推理数据集,可以调用该一个或多个算法以获得结果。

[0066] 在实施例中,经训练的回归器是经不同地配置的人工神经网络。在实施例中,经训练的回归器被包含在特定于算法的集合体中。本文还提供了用于回归器和/或集合体的最佳训练的技术。

[0067] 2.0示例计算机

[0068] 图1是描绘实施例中的示例计算机100的框图。计算机100基于经训练的回归器的性能预测来最佳地选择可训练的算法。计算机100可以是一个或多个计算机,诸如嵌入式计算机、个人计算机、诸如刀片之类的机架服务器、大型机、虚拟机,或在数值和符号处理期间使用暂时存储器的任何计算设备。

[0069] 2.1机器学习算法

[0070] 计算机100包含或访问多个不同的可训练的机器学习算法的规范,诸如121-123,其中的每个算法可以执行分析,诸如分类、回归、聚类或异常检测。例如,算法121可以是支持向量机(SVM)或人工神经网络(ANN),并且算法122可以是决策树或随机森林。

[0071] 算法121-123中的每个算法都是可训练的,并且可能是应该得到调整(再训练)或尚未训练的。算法121-123中的每个算法可能或可能没有准备就绪(经训练),以在推理数据集110上立即使用。推理数据集110可以是穷举的或代表性的经验数据,算法121-123中的任何算法可能最终用于训练或推理,诸如数据挖掘。

[0072] 训练121-123中的任何一种算法在计算上都非常昂贵,这可能会因推理数据集110

中原始数据量的增加而恶化。计算可行性可能要求计算机100(或另一个计算机)仅训练算法121-123中的一个或一小子集。

[0073] 理想地,计算机100将选择算法121-123中的可以利用推理数据集110产生最优(最准确、误差最小)的结果的一些算法(以用于训练和/或推理)。但是,由于算法121-123中的一些或全部可能仍需要训练或再训练,因此对于算法121-123的准确度预测可能是困难的或不可能的。

[0074] 可用算法(诸如121-123)的数量可能会进一步使准确度预测恶化。机器学习有数百种算法,并且仍在迅速增长。这些算法中的许多算法在可重用的库(诸如TensorFlow和scikit-learn)中是容易获得的。

[0075] 2.2元模型

[0076] 计算机100为算法121-123中的每个算法创建或获得元模型,以快速且准确地预测每个算法的性能。例如,计算机100可以创建元模型151-153作为算法121的性能预测器。

[0077] 元模型151-153中的每个元模型本身都是可训练的回归算法的实例,但是为其训练元模型的算法是不同的。例如,元模型151-153各自可以是已经被训练以预测算法121的性能的不同的神经网络,算法121可以是支持向量机而不是神经网络。元模型的训练将在本文后面讨论。

[0078] 在操作中,计算机100获得推理数据集110,并且应该使用元模型(诸如151-153)来选择算法121-123的或多或少的最佳子集,以最终针对推理数据集110进行调整。在预测算法性能时,元模型应考虑算法的特征和推理数据集110的特征。

[0079] 2.3超参数

[0080] 算法的特征被称为超参数。例如,算法121具有超参数181-184。

[0081] 如果算法121是支持向量机,那么超参数通常包括C和γ(gamma)。如果算法121是神经网络,那么超参数可以包括诸如层的计数和/或每层神经元的计数之类的特征。

[0082] 算法121-123中的每个算法可以具有基于超参数值的许多配置替代方案。例如,算法121的每个不同的配置是基于超参数181-184的一组不同的值的。

[0083] 超参数181-184中的每个超参数在逻辑上可以是多维超空间中的单独的轴。算法121的每个不同的配置对应于该超空间中的不同的点。

[0084] 超参数181-184中的一些可以是连续变量,这意味着,即使此类超参数的微小子范围也可能包含无限数量的点。由于这种棘手的组合,计算机100不应考虑超空间中的许多或大多数点。

[0085] 替代地,计算机100可以智能地或随机地对超空间进行采样,以限制计算机100实际对其进行预测性能的算法121的配置替代方案。对于每个实际配置替代方案或一组相关的配置替代方案,计算机100具有单独的元模型,诸如151-153。

[0086] 元模型151-153中的每一个被训练以预测算法121的特定配置(或一组相关配置)将如何针对与推理数据集110相似或不相似的各种数据集执行。相关配置是针对超参数181-184的子集具有相同或相似值的配置。

[0087] 例如,元模型151是通过观察算法121的实例的性能来训练的,算法121的这些实例具有针对超参数182-183有相同或相似值的配置。因此,虚线箭头从值142-143通向元模型151。

[0088] 值143和145是超参数183的不相似的值。例如,超参数183可以是神经网络中的层的计数,并且推理数据集110可以是照片的集合,使得与全色照片相比,对单色照片的分析需要更少的层。

[0089] 从值143和145通向相应的元模型151和153的发散虚线箭头可以显示,元模型151是通过观察为单色摄影配置有较少层的算法121的性能而训练的,并且元模型153是通过观察为全色摄影配置有较多层的算法121的性能而训练的。

[0090] 2.4元特征

[0091] 数据集本身的特征整体上被称为元特征。例如,推理数据集110具有元特征131-134。

[0092] 例如,如果推理数据集110是照片的集合,那么元特征131可以是照片的计数或每张照片的像素的算术平均值,并且元特征132可以是所有照片的所有像素亮度的统计方差或所有照片的边缘的中值计数,这在计算上可能多少有些严格。

[0093] 与可能具有许多值(诸如超参数184的值144和146)的超参数不同,每个元特征最多具有一个值。例如,元特征131具有值171。

[0094] 一些元特征可能适用于部分但并非全部数据集。例如,一些元特征可能自然缺少用于推理数据集110的值。例如,如果推理数据集110的照片均不包含人物,那么用于统计发型风格的元特征可能缺少值。

[0095] 2.5推理数据集

[0096] 元特征值171-174可以表征推理数据集110,使得有些相似的数据集(诸如单色照片)应该具有多少有些相似的元特征值(诸如颜色计数)。同样,算法121的不同配置替代方案可能更适合或更不适合分析不同类别的数据集。

[0097] 例如,元模型151可以对应于对单色照片表现良好的一组超参数值,并且元模型153可以对应于对全色照片表现良好的另一组超参数值。如果推理数据集110主要包含单色照片,那么元模型151应指示其超参数值的更好的适用性。

[0098] 2.6推理

[0099] 而对于大多数全色照片,元模型153应该指示其超参数值的更好的适用性。因此,通过用新的(不熟悉的)推理数据集(诸如110)的元特征值以及超参数值和各个子集刺激已经训练的元模型,计算机100可以检测各个算法121-123的各个超参数配置有多合适。

[0100] 因此,计算机100可以自发地将可训练的算法及其替代配置与特定的不熟悉的数据集进行匹配。因此,计算机100(或下游计算机)可以基于该数据集高效地将训练限制到上下文有前途(contextually promising)的算法(和配置)的最优子集。

[0101] 例如,元模型151-153各自可以是已经训练的神经网络,其将超参数值的子集和元特征值的子集作为刺激输入,如进入元模型151-153的虚线箭头所示。元模型的训练将在本文后面讨论。

[0102] 在实施例中,从超参数值(不是元特征值)进入元模型151-153的虚线箭头指示被用于训练元模型151-153的实际值。在一个实施例中,这些特定的虚线箭头指示在训练之后在算法选择的推理期间用于刺激元模型151-153的实际值。在一个实施例中,这些特定的虚线箭头指示用于训练和推理二者的实际值。在一个实施例中,在推理期间替代地使用默认超参数值(未示出)。因此,在许多实施例中,推理不需要重用用于训练同一元模型的相同的

超参数值。

[0103] 2.7分数

[0104] 元模型151-153已经是经训练的回归器,其处理输入以发出比较性的适用性分数。例如,元模型151发出分数161。

[0105] 分数161-163共享性能测量标度(scale)。例如,分数可以预测性地测量特定算法的特定配置在用特定训练数据集训练固定持续时间之后的熟练程度(准确度,诸如错误率),推理数据集110代表该训练数据集(例如是其小样本)。

[0106] 同样,分数可以替代地预测性地测量特定算法的特定配置针对特定训练数据集达到固定熟练程度需要多少时间。替代地,分数可能只是抽象适用性的比较性度量。

[0107] 无论分数语义如何,每个算法的每个元模型都发出分数。计算机100可以选择基于经排序的分数而排名的最佳的一个或几个算法(也许也是最佳超参数值),诸如所示的122。

[0108] 2.8结果

[0109] 然后,计算机100(或下游计算机)可以使用所选择的算法122来获得结果,诸如190。例如,计算机100可以使用推理数据集110(或包括110的较大数据集)来实际训练算法122的一个或几个替代配置。例如,结果190可以是算法122的准备好用于生产的配置良好且经良好训练的实例。

[0110] 本文的技术以各种方式改善了计算机100本身的性能。通过修剪超参数超空间,避免了训练过多数量的超参数配置。通过选择非常合适的算法和/或它们的配置,避免了训练过多数量的不同算法。通过基于对实际数据集元特征值的适合度进行评分,提高了选择的上下文适用性。

[0111] 因此,随后的(例如,通过计算机100的)训练进行得更快。同样,经训练的所选择的(一个或多个)算法在生产使用中(例如,通过计算机100)实现更高的准确度。因此,计算机100作为算法训练计算机被加速,并且作为生产推理计算机更加可靠(准确)。通过减少这些活动的计算负担,可以加快本文的技术(节省时间)并节省能源。

[0112] 3.0示例选择处理

[0113] 图2是描绘实施例中的计算机100基于经训练的回归器的性能预测来最佳地选择可训练的算法的流程图。图2是参考图1讨论的。

[0114] 3.1准备

[0115] 步骤202是准备。在步骤202中,对于每个元特征,通过从推理数据集中导出相应的元特征值来从推理数据集中导出各元特征值。例如,元特征131-134可以由人类专家预定义为通常可从某个应用的许多或所有数据集中获得的方面。

[0116] 例如,大多数应用数据集由数据单元(诸如像素、照片或元组(例如数据库表行))组成。在某种程度上,机器学习算法可以具有使用很少的训练数据就可以充分学习的一些配置,以及使用许多训练数据更准确地学习的其它配置,一个有用的元特征可以是数据集的大小,诸如行、像素或照片的计数。

[0117] 可以从推理数据集110提取或合成元特征值171-174。因此,推理数据集110的特征可以是已知的。

[0118] 3.2评分

[0119] 对计算机100可用的每个可训练的算法重复步骤204。对于与同一算法相关联的每

个元模型,步骤204通过基于以下至少一项来调用该元模型来计算相应的分数:a)元特征值的相应子集,或b)算法的超参数的相应子集的超参数值。

[0120] 例如,可以使用元特征值171-174的相应子集和超参数值141-146的相应子集作为推理输入来单独地刺激已经训练的不同元模型151-153。例如,元模型151基于元特征值171-172和超参数值142-143来计算分数161。

[0121] 3.3选择

[0122] 在充分重复步骤204之后,所有算法121-123的所有元模型都具有分数。基于这些分数,选择至少一个有前途的算法进行训练。例如,计算机100选择所有算法中具有最高评分元模型或所有算法中具有最高均值、中位数或众数分数的算法122。

[0123] 基于推理数据集,步骤208调用所选择的(一个或多个)算法以获得结果。这可能需要或可能不需要训练所选择的算法的至少一个模型(不同配置的实例)。

[0124] 步骤208可以通过调用所选择的算法的(一个或多个)经训练的模型来获得结果而结束。例如,该结果可以是推理数据集110或更大数据集内的对象的分类/识别。

[0125] 稍后将针对图3讨论将模型与算法的元模型进行区分的解释。稍后还将针对图3-4和图6讨论训练模型和元模型。

[0126] 4.0元学习

[0127] 图3是描绘实施例中的示例计算机300的框图。计算机300训练用于上下文准确的性能预测的特定于算法的机器学习集合体。计算机300可以是计算机100的实现方案。

[0128] 4.1集合体

[0129] 计算机300包含可配置的可训练的算法321-323。每个算法都与其自己的机器学习集合体(诸如390)相关联。

[0130] 集合体390是可训练的,因为它包含可训练的元模型351-352。元模型351-352中的每一个对应于算法321的超参数和/或超参数值的至少一个子集。

[0131] 4.2集合体分数

[0132] 当使用相同的输入值刺激元模型351-352时,生成相应的性能预测分数361-362。集合体390对分数361-362进行积分以合成集合体分数395,这是集合体390的性能预测分数,从统计学上讲,该集合体分数395通常比单独分数361-362中的任一个更准确。

[0133] 4.3SOFTMAX

[0134] 在实施例中,分数361-362被平均以合成集合体分数395。在实施例中,(归一化指数)softmax函数对分数361-362进行积分以合成集合体分数395。与统计平均值不同,softmax函数可以使用S形(sigmoidal)归一化来减少由异常值(偏差)分数引起的失真,异常值在它们产生噪声的情况下会抑制(一个或多个)元模型。

[0135] 虽然未示出,但是集合体390本身可以是包含元模型351-352的神经网络的复合神经网络。集合体390可以具有附加的最终层,其将softmax应用于分数361-362以计算集合体分数395。

[0136] 4.4性能元组

[0137] 按顺序用性能元组331-333作为输入来训练集合体390。元组331-333中的每个元组是算法321在使用特定的超参数值进行配置、经训练然后使用特定的测试数据集进行测试时的性能的历史记录。

[0138] 例如,元组332具有测试分数342,该测试分数342是在算法321使用超参数382进行配置、使用训练数据集(未示出)进行训练,然后使用测试数据集311进行测试以生成测试分数342时发生的。每个机器学习模型(诸如371-372)是使用相应的超参数值(例如分别是381-382)进行配置的算法321的经训练的实例。

[0139] 4.5模型与元模型

[0140] 模型371-372和元模型351-352的全部均是经训练和/或可训练的。但是,模型371-372是算法321的实例,并使用实际应用训练数据进行训练。而元模型351-352是不同算法的实例。

[0141] 例如,算法321和模型371-372可以是支持向量机(SVM),而元模型351-352可以是预测这些SVM的性能的神经网络。在配置和训练模型371-372之后,使用测试数据集311-312对它们进行测试,以生成分数341-343。

[0142] 4.6模型测试

[0143] 测试数据集是跨多个测试可重用的,以生成多个测试分数。例如,模型371-372二者都可以用测试数据集311进行测试以分别产生测试分数341-342。

[0144] 模型是跨多个测试可重用的,以生成多个测试分数。例如,可以用测试数据集311-312来测试模型372,以分别产生测试分数343-343。

[0145] 对于每个测试,记录引用该测试的变量的元组,这些变量可以包括测试数据集的身份、测试分数、超参数值以及算法或模型的身份。例如,元组332记录算法321是用超参数值382配置的以创建和训练机器学习模型372,该机器学习模型372在用测试数据集311进行测试时产生了测试分数342。

[0146] 元组331-333可以存储在易失性存储器中或持久地存储在诸如关系数据库或盘文件中。在生成元组331-333的模型训练和测试之后,该元组被用作训练集合体390的输入。

[0147] 4.7元模型训练

[0148] 对于集合体训练期间的每个元组,可以从该元组获得超参数值,并且可以从该元组的训练集直接或间接地获得元特征值。在实施例中,针对该元组获得的这些值可以被用于基于这些值或包括这些值的值的子范围来选择元模型351-352中的一个或一些以用于训练。在实施例中,这些值被用于训练所有的元模型351-352。

[0149] 这些值作为刺激输入被注入到一些或所有元模型以进行训练。因此,元模型351-352和集合体390可以学习预测算法321和/或模型371-372的性能。

[0150] 4.8特定于竞争算法的集合体

[0151] 虽然未示出,但是算法322-323各自也具有其自己的可训练集合体。由于每个集合体(及其元模型)是针对单独的算法进行训练的,因此每个集合体都学习很好地预测其自己的算法的性能而没有交叉训练干扰,否则交叉训练干扰会由于不得不学习多个不同算法的矛盾性能怪异性而引起。

[0152] 因此,在集合体训练之后的推理期间,计算机300可以调用多个集合体以检测每个算法对于同一不熟悉的推理数据集将执行得如何。例如,计算机300可以选择具有(一个或多个)最佳集合体分数的一个或几个算法以随后利用推理数据集所代表的不熟悉的训练数据集进行昂贵的训练。因此,可以避免对针对该推理数据集具有无前途的集合体分数的算法进行昂贵的训练,而不会失去最佳性。

[0153] 4.9训练数据集分区

[0154] 在实施例中,诸如k-fold交叉验证之类的交叉验证被用于从一个原始训练数据集创建许多对训练数据集和测试数据集,使得每个对包含原始训练数据集但在训练数据集和测试数据集之间以不同方式进行分区。

[0155] 同样,一些元模型可以保留(专用)用于仅使用超参数、元特征、值和/或值子范围的特定子集进行训练。在训练特定元模型时,其值与该元模型的期望值不相关的元组可以被跳过。如果元组缺少元模型期望的值,那么在训练该元模型时可以跳过该元组。

[0156] 在实施例中,缺失不同元组(所有元组或一个元模型的装袋元组,如稍后针对图6所述)的超过阈值的的百分比的值的元特征被排除。在实施例中,缺失值的各个元组(所有元组或一个元模型的装袋元组)被排除。

[0157] 例如,对于元组331-333,只有332适合于训练两个元模型351-352,如流入到元模型351-352中的虚线箭头所指示的。接受同一共享元组的多个元模型(诸如332)可以采用与该共享元组相关联的(超参数和元特征)值的不同子集。

[0158] 由于元模型351-352是用元组的多少有些不同的子集和共享元组中的值的多少有些不同的子集训练的,因此元模型351-352实际上是用不同的数据进行训练的。因此,即使最初配置相同(例如,相同的层计数),在训练之后,元模型351-352实际上是不同的。因此,集合体390对预测的实际多样性进行积分。下面针对图6进一步讨论训练数据集分区。

[0159] 训练多个算法的元模型和多个模型的是计算密集型的,并且应服从水平缩放以进行加速。例如,集群中的每个计算机可以串联训练一个或几个模型或元模型。因为每个模型的训练(尤其是测试)可以是并发的,因此可能并发地生成元组。

[0160] 5.0示例元学习处理

[0161] 图4是描绘实施例中的计算机300训练特定于算法的机器学习集合体以进行上下文准确的性能预测的流程图。图4是参考图3讨论的。

[0162] 5.1准备

[0163] 步骤401是准备。在步骤401中,接收、存储、生成或以其它方式获得测试数据集。例如,人类专家或数据仓库提供测试数据集311-312作为文件、查询结果集或流。

[0164] 对于每个可用的机器学习算法,重复步骤402-405。具体地,针对每个算法的每个模型重复步骤402-404。具体地,对于每个测试数据集以及每个算法的每个模型,重复步骤403-404。

[0165] 步骤402基于算法的超参数的相应特定值来配置算法的模型。例如,模型371-372中的每个模型单独用对于算法321而言是公共的一组共享超参数的一组不同的值来配置,诸如分别为381-382。

[0166] 在充分重复步骤402之后,所有算法321-323的所有模型被不同地配置并且准备好用于训练。虽然未示出,但是模型训练可以基于训练数据集(未示出)在步骤402-403之间发生。

[0167] 5.2建模

[0168] 步骤403的每次发生执行不同的测试。步骤403使用测试数据集之一来测试算法的模型,以计算相应的测试分数。例如,在一个测试期间,当由测试数据集311刺激时,模型372可以生成测试分数342。

[0169] 步骤404为在步骤403中执行的每个测试记录不同的元组。每个元组引用、识别或包含以下各项:算法的超参数的相应特定值、测试数据集、相应的测试分数以及该算法。

[0170] 例如,当用测试数据集311对模型372进行测试时,记录元组332。元组332可以指示测试数据集311、测试分数342、超参数382、算法321和/或模型372。

[0171] 在充分重复步骤404之后,已经用每个测试数据集311-312单独测试所有算法321-323的所有模型,并且记录了所有元组331-333。因此,所有集合体都准备好用于训练。

[0172] 5.3集合体训练

[0173] 步骤405独立地训练每个算法321-323的集合体,诸如390。对于与算法相关联的每个元模型,步骤405基于为该算法记录的至少一个不同的元组训练该算法的所有元模型。

[0174] 例如,元组331-332被用于训练元模型352。在充分重复步骤405之后,已经训练了所有算法321-323的所有元模型,这意味着已经训练了所有集合体,诸如390。

[0175] 在实施例中,用Keras Python库、TensorFlow和/或Apache MXNet的某种组合来实现集合体训练,其诸如利用多个图形处理单元(GPU)来水平地进行缩放。

[0176] 在步骤405之后,算法321-323的每个集合体都准备好进行推理,诸如在暴露于特定推理数据集(未显示且可能大于任何测试数据集311-312)期间。

[0177] 6.0值转换

[0178] 图5是描绘实施例中的示例计算机500的框图。计算机500调整特征(超参数和元特征)值以实现均匀性、可移植性和通用性。计算机500可以是计算机100的实现方案。

[0179] 6.1原始数据

[0180] 计算机500包含从原始(例如,野生)数据转换而来的调整后的数据510和520。原始数据自然可以是数值型的或类别型的。

[0181] 6.2类别

[0182] 类别型数据是非数值型数据,类别型数据的域由离散(符号)值组成。为了一致性、可移植性和通用性,可以将非数值型数据转换成数值型的。

[0183] 例如,从1月到12月的日历月份可以转换成从1到12的整数。因此,保留了月份的自然相对排序,并且月份之间(例如,3月＝3和7月＝7之间)的自然距离是有意义的(例如,7-3＝4)。

[0184] 但是,类别型数据可能具有自然无序的值(类别),在这种情况下,密集数值型编码可能会导致相邻数字被错误地视为语义相似(相邻)。为了防止这种错误的关联使训练失真,可以将数字的几何标度分配给类别,使得每个数字与所有其它值至少相差一个数量级。

[0185] 6.3独热编码

[0186] 例如,表独热510将陆地大陆显示为类别,其自然是无序的。通过独热编码,可以将七到八位的串中的每一位保留用于大陆,有点像位图。

[0187] 例如,最低有效位被保留用于非洲,它被独热编码为00000001。替代地,类别可以被编码为独冷(one-cold),它是独热的按位逆转,使得位图中只有一位是清除的(零)。

[0188] 6.4数字

[0189] 可以对数值型数据进行归一化以实现一致性、可移植性和通用性。归一化可能需要施加零均值和/或单位方差。

[0190] 6.5零均值

[0191] 零均值使数字的范围移位,直到算术均值变为零为止,该算术均值也可以是或者可以不是算术中位数和/或众数。例如,原始数据的高斯分布520的原始均值为1010,然后从所有值中减去该原始均值以实现归一化的零均值,其中较小的值变为负数。

[0192] 6.6单位方差

[0193] 单位方差缩放(例如,压缩)数字的范围,直到统计方差为1。例如,所示的相邻原始值970和990可以是990-970=相隔20个单位。这两个数字被压缩为相隔一个单位,在该示例中分别为-2和-1。

[0194] 6.7归一化

[0195] 当发生零均值和单位方差调整时(如针对高斯分布520所示的那样),那么值、子范围和距离变得均匀,而不管原来的原始标度有多少差异。例如,指南针方向的范围可以从零至360,并且华氏温度的季节范围可以从-20至120。

[0196] 在如所描述的归一化之后,两个经归一化的指南针方向之间的给定量值的增量可以表示与两个经归一化的温度之间的相同给定量值变化相同的变化量。因此,可以使用类似的增量沿着任一(方向或温度)轴进行多维解空间的探索。

[0197] 因此,所有轴都可以或多或少均匀地被探索和评分,使得一个轴的优化不会偶然地主宰对解空间的其它轴的考虑。这种归一化还使分数易于比较,使得可以可靠地对兄弟元模型的分数(以及从不同算法发出的分数)进行排名(按值相对排序)。

[0198] 7.0元优化

[0199] 图6是描绘实施例中的示例计算机600的框图。计算机600通过诸如提升和装袋之类的优化来改善集合体元学习。计算机500可以是计算机系统100的实现方案。

[0200] 计算机600训练包括通过使用元组630作为训练数据集而学习的元模型651-656的集合体(未示出)。元组630记录了训练之后来自特定机器学习算法X(未示出)的各种配置针对各种测试数据集的性能,如针对图3所解释的。元模型651-656可以是另一个机器学习算法Y(未显示)的实例,该算法可以与算法X相同或可以不相同。

[0201] 元组630包含配置了算法X的不同实例的超参数680。元组630还包含为算法X从各种测试数据集(未示出)中提取的元特征640。

[0202] 虽然未示出,但是元组630也可以包含其它数据,诸如算法X的测试分数。元组630的每个元组(行)表示涉及算法X的对一个测试数据集进行推理的一个模型(实例)的一次测试。

[0203] 7.1方差

[0204] 偏差和方差之间的权衡是机器学习(诸如在训练元模型651-656的情况下)特有的。方差(又称过度拟合)是训练期间的过度敏感,其导致噪声被误认为有意义的模式,这产生持久的错误印象。

[0205] 7.2偏差

[0206] 偏差(又称欠拟合)是过低敏感,其导致有意义的模式作为噪声而被忽略,这使训练延长或阻止完整训练。

[0207] 7.3自举聚合

[0208] 可以通过经由使用称为自举聚合(又称装袋)的技术将集合体训练数据集分解成部分重叠的数据子集而重构该数据集(诸如元组630)来减少失真(诸如方差)。

[0209] 7.4特征装袋

[0210] 特征装袋需要使用训练数据集特征(诸如元特征640和超参数680)的不同的部分重叠子集来训练每个元模型651-656。例如,根据图例620,元组630的仅元特征F-G和超参数I-J被用于训练元模型655。

[0211] 7.5样本装袋

[0212] 样本装袋需要使用元组630的训练数据集行的不同的部分重叠子集来训练每个元模型651-656。例如,根据图例620,元组630的仅行Q-T被用于训练元模型651。

[0213] 7.6数据集分区

[0214] 虽然未显示,但是样本装袋和特征装袋二者都可以用于训练同一元模型。例如,元模型651-656可以各自用元组630的行的相应子集的列的相应子集来训练。

[0215] 7.7重叠

[0216] 元组630的子集可以部分重叠。例如,虽然未示出,但是样本装袋可以利用元组630的底部四行来训练元模型652,该底部四行630包括行T,该行T也如图所示用于训练元模型651。理想情况下,元模型的训练子集的最多三分之一应与(一个或多个)其它子集重叠,无论是样本还是特征装袋。

[0217] 7.8提升

[0218] 另一个集合体元学习优化是假设提升(hypothesis boosting),该假设提升可以减少方差(尤其是偏差)。提升将权重分配给元组630的行。

[0219] 元组630的每一行取决于训练元模型651-656从该行学习的容易程度(容易性)。最初,所有行被相等地加权。

[0220] 在训练期间,被准确处理(低错误)的行的权重减小。相反,具有高错误的行的权重增加。

[0221] 随着元组630的一些或全部行的权重演变,可以重复进行训练,以实现比每行仅进行一次训练更高的准确度。在实施例中,仅对具有至少阈值权重的行重复训练。

[0222] 在实施例中,阈值逐渐增加。在实施例中,当很少或没有行权重超过阈值时,训练停止。

[0223] 提升还可以基于集合体的元模型651-656的准确度将权重分配给集合体的元模型651-656。因此,更可靠的元模型可能比较不可靠的元模型对集合体分数具有更大的影响。

[0224] 可以基于在训练期间的不同时间观察到的准确度来调整元模型权重。在实施例中,当大多数或所有元模型权重停止改变至少阈值量时,训练停止。

[0225] 7.9集合体优化

[0226] 集合体元学习适合在训练之前进行附加的元优化。集合体本身具有超参数(未示出),诸如每个元模型用于特征装袋的特征(超参数、元特征)的百分比、用于样本装袋的行的百分比,以及元模型的计数。

[0227] 同样,元模型651-656本身具有超参数(未示出),诸如在集合体由神经网络组成的情况下的层的计数,或者在由另一机器学习算法组成的情况下的其他超参数。元优化可以使用梯度下降、贝叶斯优化、SVM或决策树。计算机600可以使用超参数优化工具(诸如hyperopt Python库)来优化集合体和/或其组成元模型651-656的配置。根据设计,hyperopt是水平可伸缩的。

[0228] 8.0硬件概述

[0229] 根据一个实施例,本文描述的技术由一个或多个专用计算设备实现。专用计算设备可以是硬连线的以执行这些技术,或者可以包括数字电子设备(诸如被持久地编程为执行这些技术的一个或多个专用集成电路(ASIC)或现场可编程门阵列(FPGA)),或者可以包括被编程为根据固件、存储器、其它存储装置或组合中的程序指令执行这些技术的一个或多个通用硬件处理器。这种专用计算设备还可以将定制的硬连线逻辑、ASIC或FPGA与定制编程相结合,以实现这些技术。专用计算设备可以是台式计算机系统、便携式计算机系统、手持设备、联网设备或者结合硬连线和/或程序逻辑以实现这些技术的任何其它设备。

[0230] 例如,图7是示出可以在其上实现本发明的实施例的计算机系统700的框图。计算机系统700包括总线702或用于传送信息的其它通信机制,以及与总线702耦合用于处理信息的硬件处理器704。硬件处理器704可以是例如通用微处理器。

[0231] 计算机系统700还包括耦合到总线702用于存储信息和要由处理器704执行的指令的主存储器706,诸如随机存取存储器(RAM)或其它动态存储设备。主存储器706也可以用于在要由处理器704执行的指令执行期间存储临时变量或其它中间信息。当这些指令被存储在处理器704可访问的非暂态存储介质中时,它们使计算机系统700成为被定制以执行指令中指定的操作的专用机器。

[0232] 计算机系统700还包括耦合到总线702用于存储静态信息和用于处理器704的指令的只读存储器(ROM)708或其它静态存储设备。提供了诸如磁盘或光盘的存储设备710,并且存储设备710被耦合到总线702,用于存储信息和指令。

[0233] 计算机系统700可以经由总线702耦合到显示器712,诸如阴极射线管(CRT),用于向计算机用户显示信息。输入设备714(其包括字母数字和其它键)被耦合到总线702,用于将信息和命令选择传送到处理器704。另一种类型的用户输入设备是光标控件716,诸如鼠标、轨迹球、或光标方向键,用于向处理器704传送方向信息和命令选择并且用于控制光标在显示器712上的移动。这种输入设备通常具有在两个轴(第一轴(例如,x)和第二轴(例如,y))中的两个自由度,这允许设备在平面中指定位置。

[0234] 计算机系统700可以使用定制的硬连线逻辑、一个或多个ASIC或FPGA、固件和/或程序逻辑来实现本文描述的技术,这些定制的硬连线逻辑、一个或多个ASIC或FPGA、固件和/或程序逻辑与计算机系统结合使计算机系统700成为或将计算机系统700编程为专用机器。根据一个实施例,本文的技术由计算机系统700响应于处理器704执行主存储器706中包含的一条或多条指令的一个或多个序列而执行。这些指令可以从另一个存储介质(诸如存储设备710)读取到主存储器706中。包含在主存储器706中的指令序列的执行使处理器704执行本文描述的处理步骤。在替代实施例中,可以使用硬连线电路系统代替软件指令或与软件指令组合使用。

[0235] 如本文所使用的术语“存储介质”是指存储使机器以特定方式操作的数据和/或指令的任何非瞬态介质。这种存储介质可以包括非易失性介质和/或易失性介质。非易失性介质包括例如光盘或磁盘,诸如存储设备76。易失性介质包括动态存储器,诸如主存储器706。存储介质的常见形式包括例如软盘、柔性盘、硬盘、固态驱动器、磁带或任何其它磁性数据存储介质、CD-ROM、任何其它光学数据存储介质、具有孔模式的任何物理介质、RAM、PROM和EPROM、FLASH-EPROM、NVRAM、任何其它存储器芯片或盒带。

[0236] 存储介质与传输介质不同但可以与传输介质结合使用。传输介质参与在存储介质之间传输信息。例如,传输介质包括同轴电缆、铜线和光纤,包括包含总线702的电线。传输介质还可以采取声波或光波的形式,诸如在无线电波和红外线数据通信期间生成的那些波。

[0237] 各种形式的介质可以涉及将一条或多条指令的一个或多个序列携带到处理器704以供执行。例如,指令最初可以被携带在远程计算机的磁盘或固态驱动器上。远程计算机可以将指令加载到其动态存储器中,并且使用调制解调器经电话线发送指令。计算机系统700本地的调制解调器可以接收电话线上的数据,并且使用红外线发射器将数据转换为红外线信号。红外线探测器可以接收在红外线信号中携带的数据,并且适当的电路系统可以将数据放在总线702上。总线702将数据携带到主存储器706,处理器704从该主存储器706检索并执行指令。由主存储器706接收到的指令可以可选地在被处理器704执行之前或执行之后存储在存储设备76上。

[0238] 计算机系统700还包括耦合到总线702的通信接口718。通信接口718提供耦合到网络链路720的双向数据通信,其中网络链路720连接到本地网络722。例如,通信接口718可以是综合业务数字网(ISDN)卡、电缆调制解调器、卫星调制解调器、或向对应类型的电话线提供数据通信连接的调制解调器。作为另一个示例,通信接口718可以是提供到兼容的局域网(LAN)的数据通信连接的LAN卡。也可以实现无线链路。在任何这种实现中,通信接口718都发送和接收携带表示各种类型信息的数字数据流的电信号、电磁信号或光信号。

[0239] 网络链路720通常通过一个或多个网络向其它数据设备提供数据通信。例如,网络链路720可以通过本地网络722提供到主计算机724或到由互联网服务提供商(ISP)726操作的数据设备的连接。ISP 726又通过现在通常称为“互联网”728的世界范围的分组数据通信网络提供数据通信服务。本地网络722和互联网728二者都使用携带数字数据流的电信号、电磁信号或光信号。通过各种网络的信号以及在网络链路720上并且通过通信接口718的信号是传输介质的示例形式,其中信息将数字数据携带到计算机系统700或者携带来自计算机系统700的数字数据。

[0240] 计算机系统700可以通过(一个或多个)网络、网络链路720和通信接口718发送消息和接收数据,包括程序代码。在互联网示例中,服务器730可以通过互联网728、ISP 726、本地网络722和通信接口718传送对应用程序的请求代码。

[0241] 接收到的代码可以在其被接收到时由处理器704执行,和/或存储在存储设备710或其它非易失性存储器中以供以后执行。

[0242] 10.0软件概述

[0243] 图8是可以用于控制计算机系统700的操作的基本软件系统800的框图。软件系统800及其部件,包括它们的连接、关系和功能,仅仅是示例性的,并且不意味着限制(一个或多个)示例实施例的实现方案。适于实现(一个或多个)示例实施例的其它软件系统可以具有不同的部件,包括具有不同的连接、关系和功能的部件。

[0244] 提供软件系统800用于指导计算机系统700的操作。可以存储在系统存储器(RAM)706和固定存储装置(例如,硬盘或闪存)76上的软件系统800包括内核或操作系统(OS)810。

[0245] OS 810管理计算机操作的低级方面,包括管理进程的执行、存储器分配、文件输入和输出(I/O)以及设备I/O。表示为802A、802B、802C...802N的一个或多个应用可以被“加

载”(例如,从固定存储装置76传送到存储器706中)以供系统800执行。意图在计算机系统700上使用的应用或其它软件也可以被存储为可下载的计算机可执行指令集,例如,用于从互联网位置(例如,Web服务器、app商店或其它在线服务)下载和安装。

[0246] 软件系统800包括图形用户界面(GUI)815,用于以图形(例如,“点击”或“触摸手势”)方式接收用户命令和数据。进而,这些输入可以由系统800根据来自操作系统810和/或(一个或多个)应用802的指令来操作。GUI 815还用于显示来自OS 810和(一个或多个)应用802的操作结果,用户可以提供附加的输入或终止会话(例如,注销)。

[0247] OS 810可以直接在计算机系统700的裸硬件820(例如,(一个或多个)处理器704)上执行。可替代地,管理程序或虚拟机监视器(VMM)830可以被插入在裸硬件820和OS 810之间。在这个配置中,VMM 830充当OS 810与计算机系统700的裸硬件820之间的软件“缓冲”或虚拟化层。

[0248] VMM 830实例化并运行一个或多个虚拟机实例(“游客机”)。每个游客机包括“游客”操作系统(诸如OS 810),以及被设计为在游客操作系统上执行的一个或多个应用(诸如(一个或多个)应用802)。VMM 830向游客操作系统呈现虚拟操作平台并管理游客操作系统的执行。

[0249] 在一些实例中,VMM 830可以允许游客操作系统如同其直接在计算机系统800的裸硬件820上运行一样运行。在这些实例中,被配置为直接在裸硬件820上执行的游客操作系统的相同版本也可以在VMM 830上执行而无需修改或重新配置。换句话说,VMM 830可以在一些情况下向游客操作系统提供完整的硬件和CPU虚拟化。

[0250] 在其它实例中,游客操作系统可以被专门设计或配置为在VMM 830上执行以提高效率。在这些实例中,游客操作系统“意识到”它在虚拟机监视器上执行。换句话说,VMM 830可以在某些情况下向客户操作系统提供半虚拟化。

[0251] 计算机系统进程包括硬件处理器时间的分配,以及存储器的分配(物理和/或虚拟)、用于存储由硬件处理器执行的指令的存储器的分配,用于存储由硬件处理器执行指令所生成的数据,和/或用于当计算机系统进程未运行时在硬件处理器时间的分配之间存储硬件处理器状态(例如,寄存器的内容)。计算机系统进程在操作系统的控制下运行,并且可以在计算机系统上执行的其它程序的控制下运行。

[0252] 8.0云计算

[0253] 本文一般地使用术语“云计算”来描述计算模型,该计算模型使得能够按需访问计算资源的共享池,诸如计算机网络、服务器、软件应用和服务,并且允许以最少的管理工作或服务提供商交互来快速提供和释放资源。

[0254] 云计算环境(有时称为云环境或云)可以以各种不同方式实现,以最好地适应不同要求。例如,在公共云环境中,底层计算基础设施由组织拥有,该组织使其云服务可供其它组织或公众使用。相反,私有云环境一般仅供单个组织使用或在单个组织内使用。社区云旨在由社区内的若干组织共享;而混合云包括通过数据和应用可移植性绑定在一起的两种或更多种类型的云(例如,私有、社区或公共)。

[0255] 一般而言,云计算模型使得先前可能由组织自己的信息技术部门提供的那些职责中的一些代替地作为云环境内的服务层来输送,以供消费者使用(根据云的公共/私人性质,在组织内部或外部)。取决于特定实现方案,由每个云服务层提供或在每个云服务层内

提供的部件或特征的精确定义可以有所不同,但常见示例包括:软件即服务(SaaS),其中消费者使用在云基础设施上运行的软件应用,同时SaaS提供者管理或控制底层云基础设施和应用。平台即服务(PaaS),其中消费者可以使用由PaaS的供应者支持的软件编程语言和开发工具,以开发、部署和以其它方式控制它们自己的应用,同时PaaS提供者管理或控制云环境的其它方面(即,运行时执行环境下的一切)。基础设施即服务(IaaS),其中消费者可以部署和运行任意软件应用,和/或提供进程、存储装置、网络和其它基础计算资源,同时IaaS提供者管理或控制底层物理云基础设施(即,操作系统层下面的一切)。数据库即服务(DBaaS),其中消费者使用在云基础设施上运行的数据库服务器或数据库管理系统,同时DbaaS提供者管理或控制底层云基础设施和应用。

[0256] 呈现上述基本计算机硬件和软件以及云计算环境是为了说明可以用于实现(一个或多个)示例实施例的基本底层计算机部件。然而,(一个或多个)示例实施例不必限于任何特定的计算环境或计算设备配置。相反,根据本公开,(一个或多个)示例实施例可以在本领域技术人员依据本公开将理解为能够支持本文呈现的(一个或多个)示例实施例的特征和功能的任何类型的系统体系架构或处理环境中实现。

[0257] 在前面的说明书中,已经参考众多具体细节描述了本发明的实施例,这些细节可以从实现到实现有所变化。因而,说明书和附图应被视为说明性而非限制性的。本发明范围的唯一和排他性指示,以及申请人意图作为本发明范围的内容,是以发布这种权利要求书的具体形式从本申请发布的权利要求书集合的字面和等同范围,包括任何后续更正。

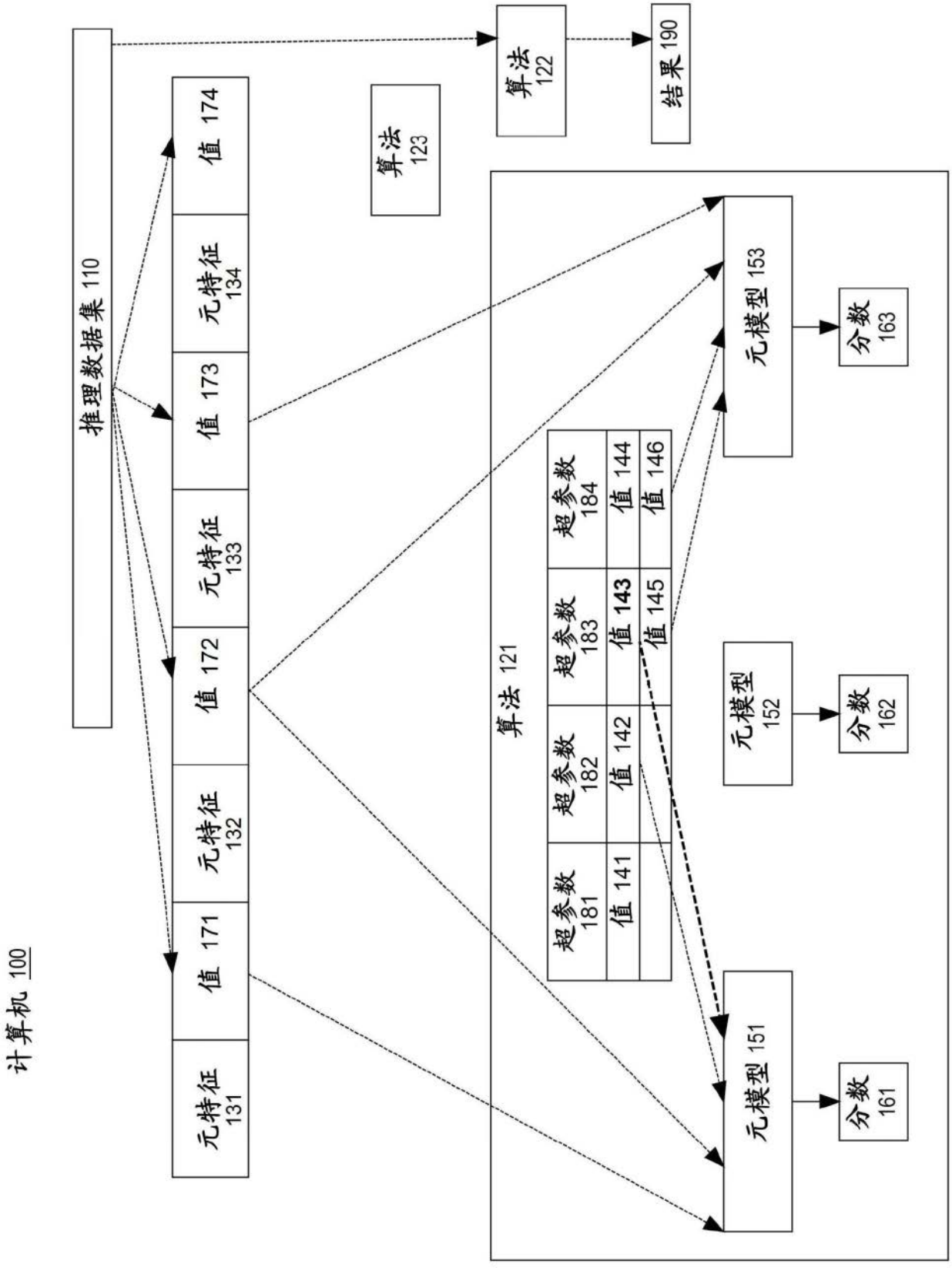

计算机 100
推理数据集 110
元特征 131
值 171
元特征 132
值 172
元特征 133
值 173
元特征 134
值 174
算法 123
算法 122
结果 190
算法 121
超参数 181
值 141
超参数 182
值 142
超参数 183
值 143
值 145
超参数 184
值 144
值 146
元模型 151
元模型 152
元模型 153
分数 161
分数 162
分数 163

图1

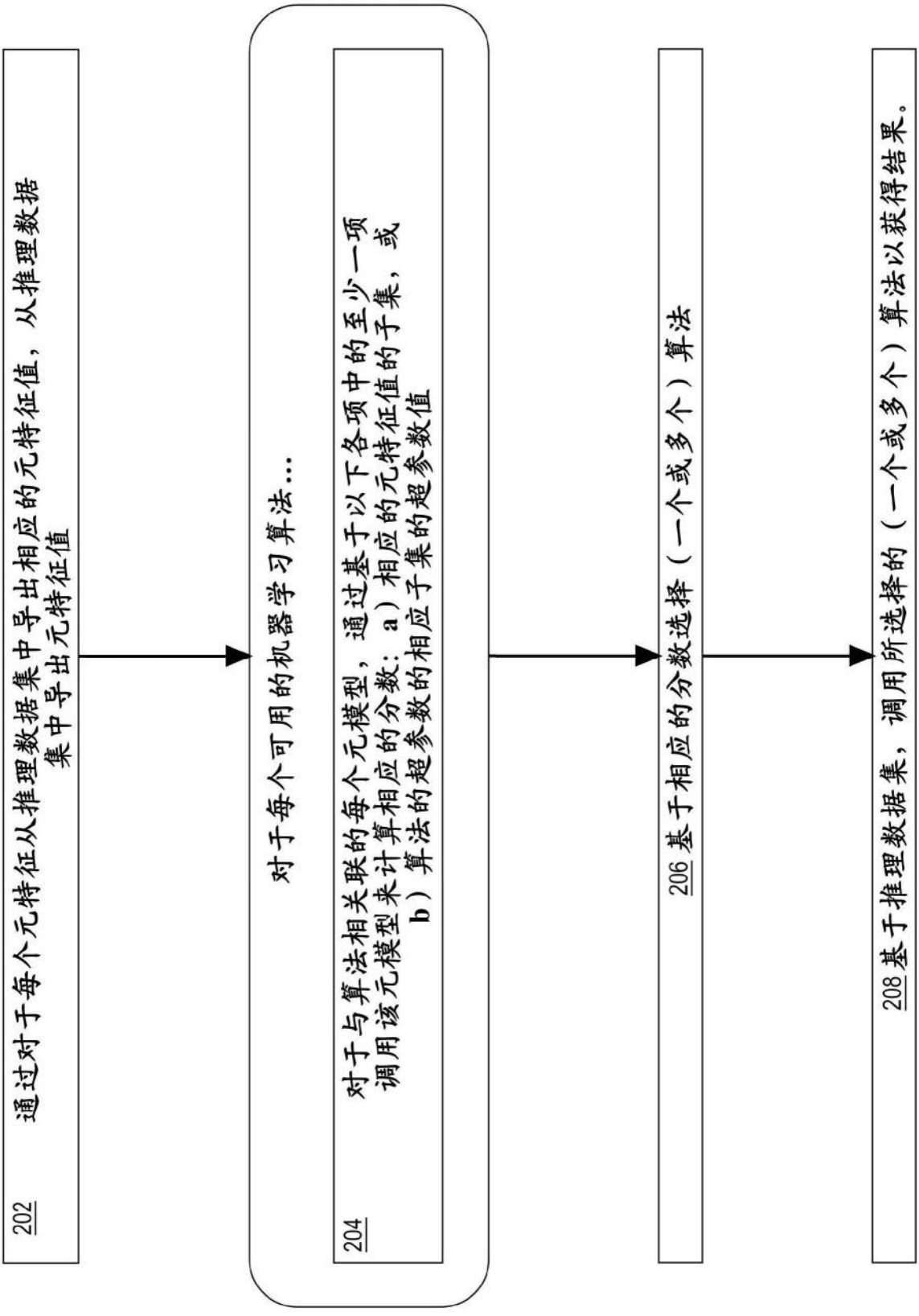

202 通过对于每个元特征从推理数据集中导出相应的元特征值，从推理数据集中导出元特征值
对于每个可用的机器学习算法...
204 对于与算法相关联的每个元模型，通过基于以下各项中的至少一项调用该元模型来计算相应的分数：a）相应的元特征值的子集，或b）算法的超参数的相应子集的超参数值
206 基于相应的分数选择（一个或多个）算法
208 基于推理数据集，调用所选择的（一个或多个）算法以获得结果。

图2

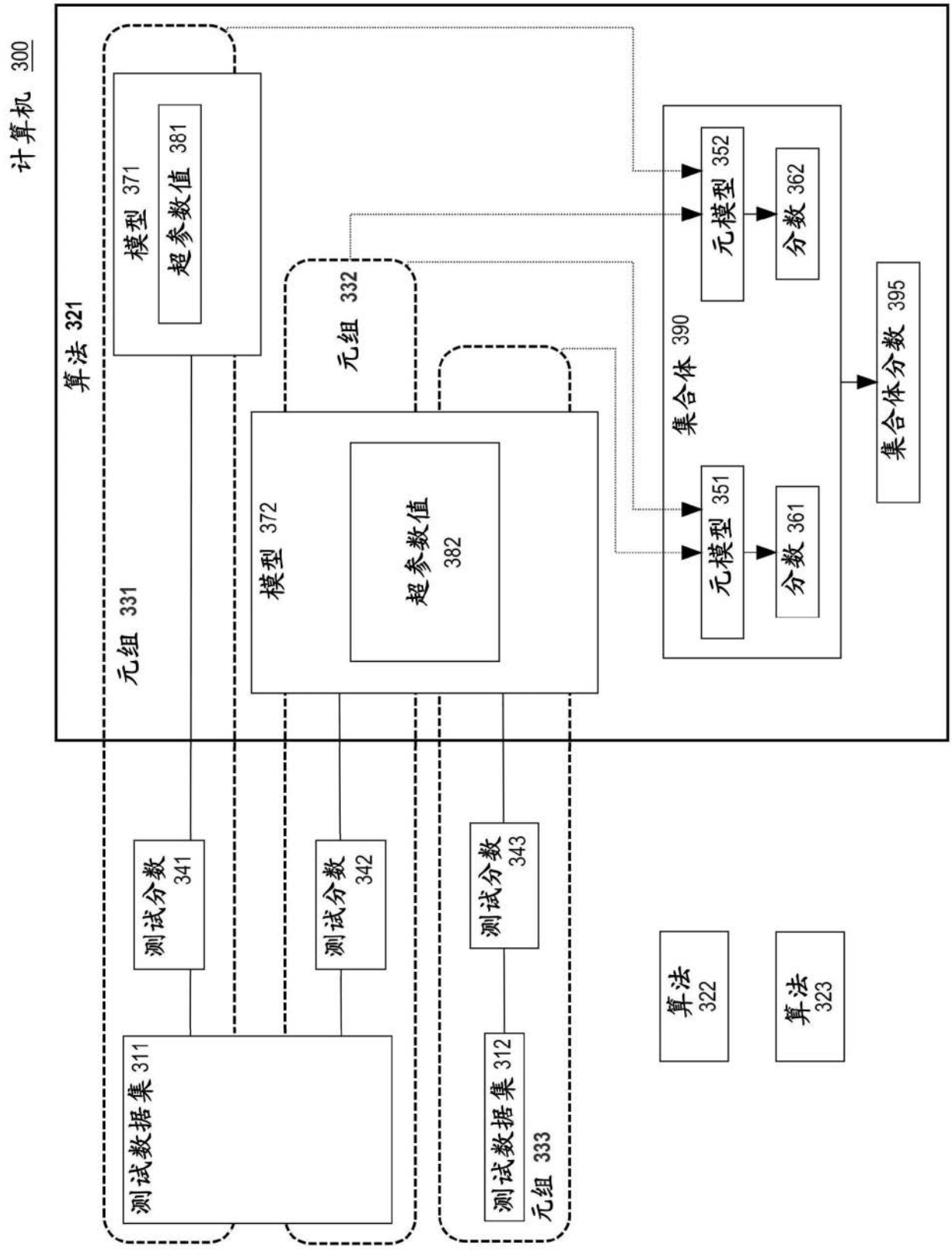

计算机 300
算法 321
模型 371
超参数值 381
元组 331
测试数据集 311
测试分数 341
元组 332
模型 372
超参数值 382
测试分数 342
元组 333
测试数据集 312
测试分数 343
集合体 390
元模型 352
分数 362
元模型 351
分数 361
集合体分数 395
算法 322
算法 323

图3

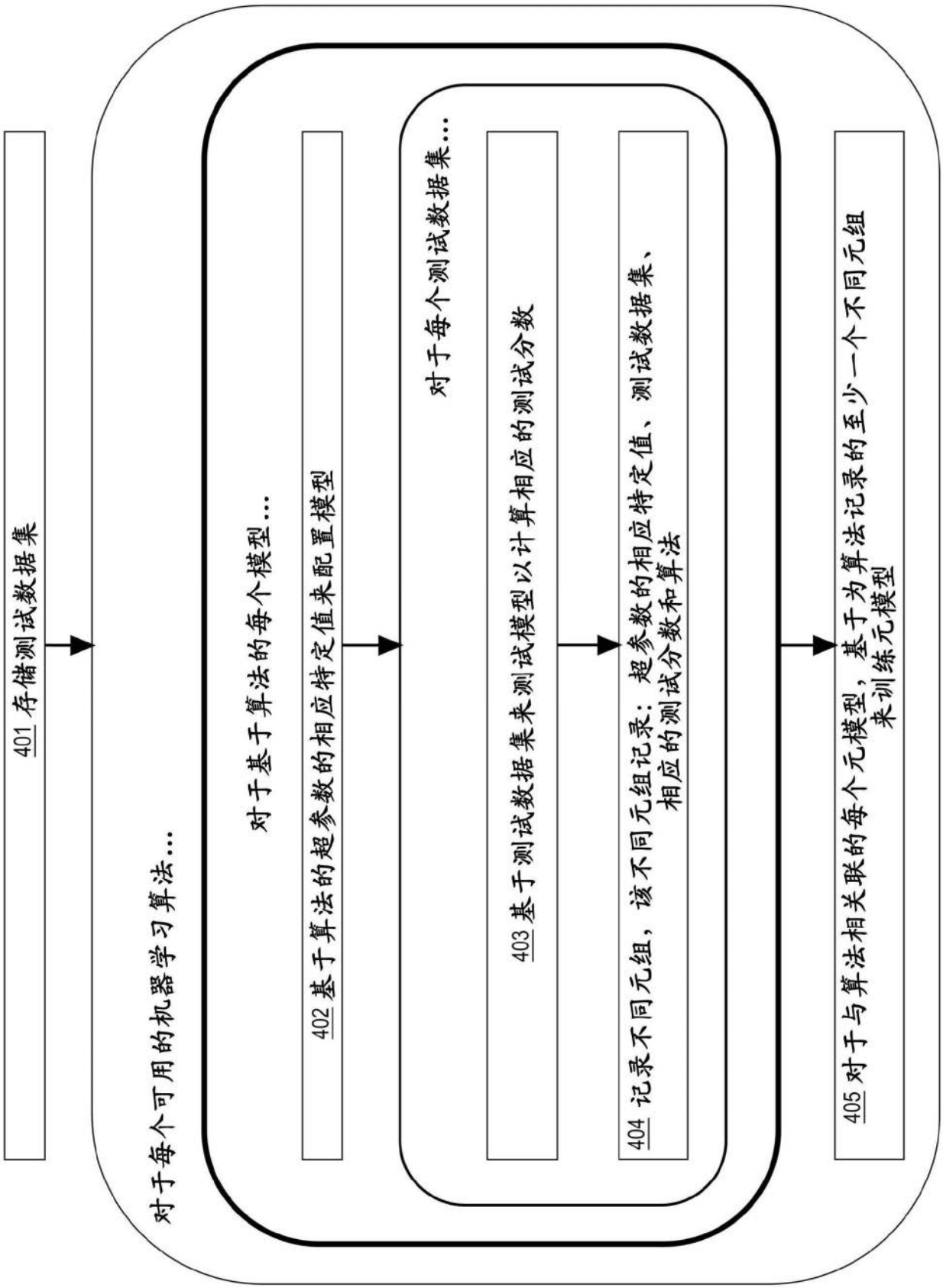

401 存储测试数据集
对于每个可用的机器学习算法...
对于基于算法的每个模型...
402 基于算法的超参数的相应特定值来配置模型
对于每个测试数据集...
403 基于测试数据集来测试模型以计算相应的测试分数
404 记录不同元组，该不同元组记录：超参数的相应特定值、测试数据集、相应的测试分数和算法
405 对于与算法相关联的每个元模型，基于为算法记录的至少一个不同元组来训练元模型

图4

计算机 500

独热 510							
类别	非洲	南极洲	亚洲	澳洲	欧洲	北美洲	南美洲
位	00000001	00000010	00000100	00001000	00010000	00100000	01000000

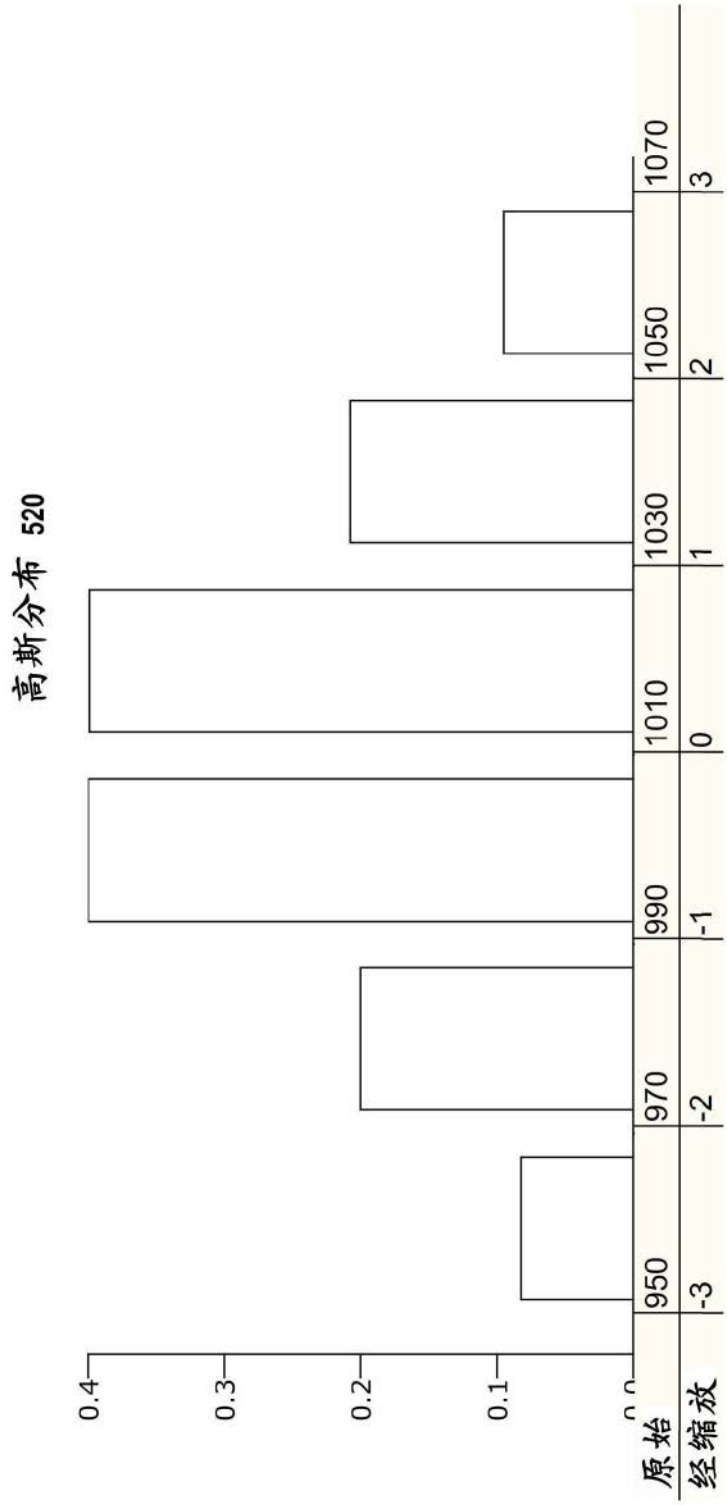


图5

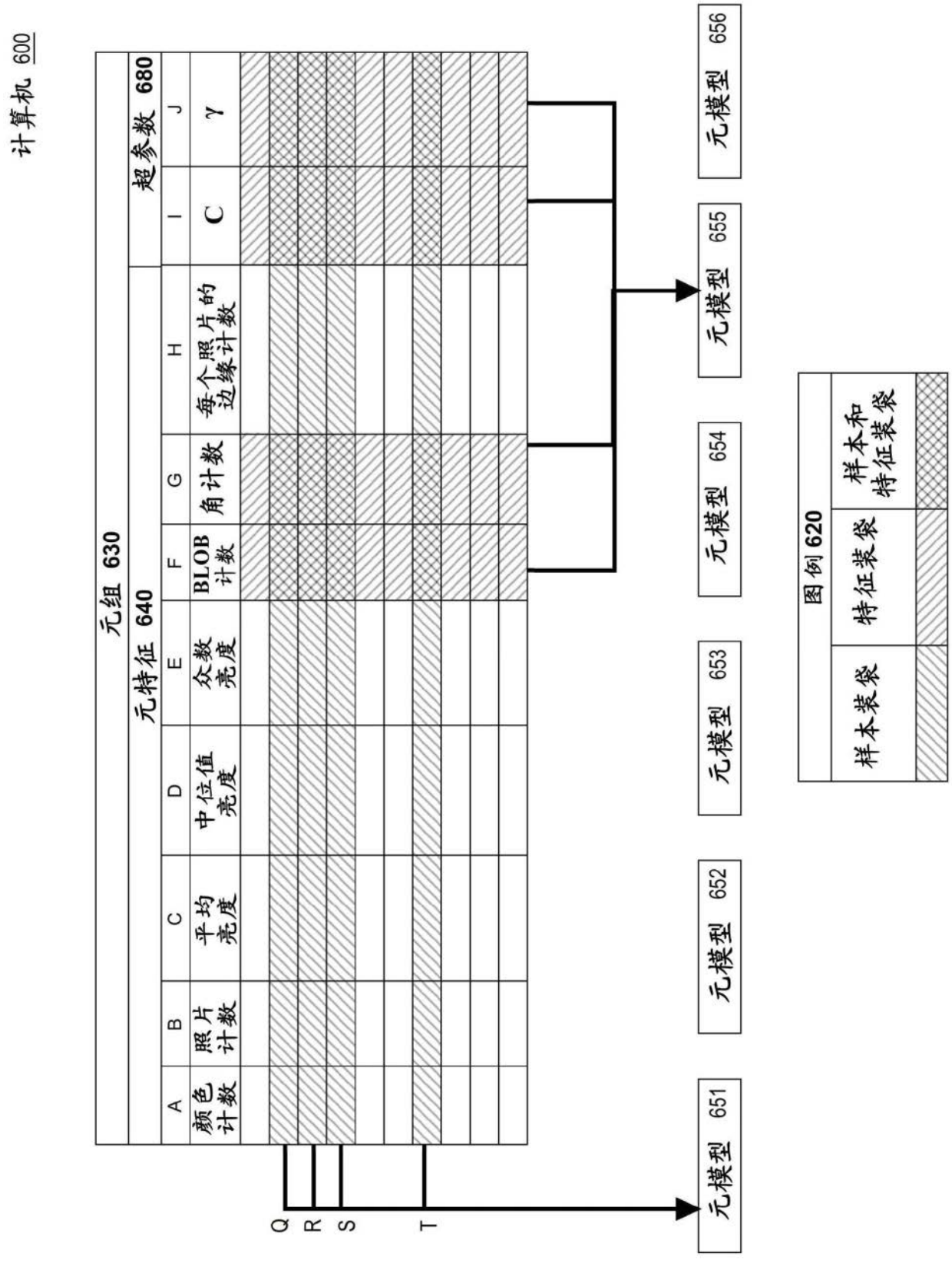

计算机 600
元组 630
元特征 640
超参数 680
A 颜色计数
B 照片计数
C 平均亮度
D 中位值亮度
E 众数亮度
F BLOB计数
G 角计数
H 每个照片的边缘计数
I C
J γ
Q
R
S
T
元模型 651
元模型 652
元模型 653
元模型 654
元模型 655
元模型 656
图例 620
样本装袋
特征装袋
样本和特征装袋

图6

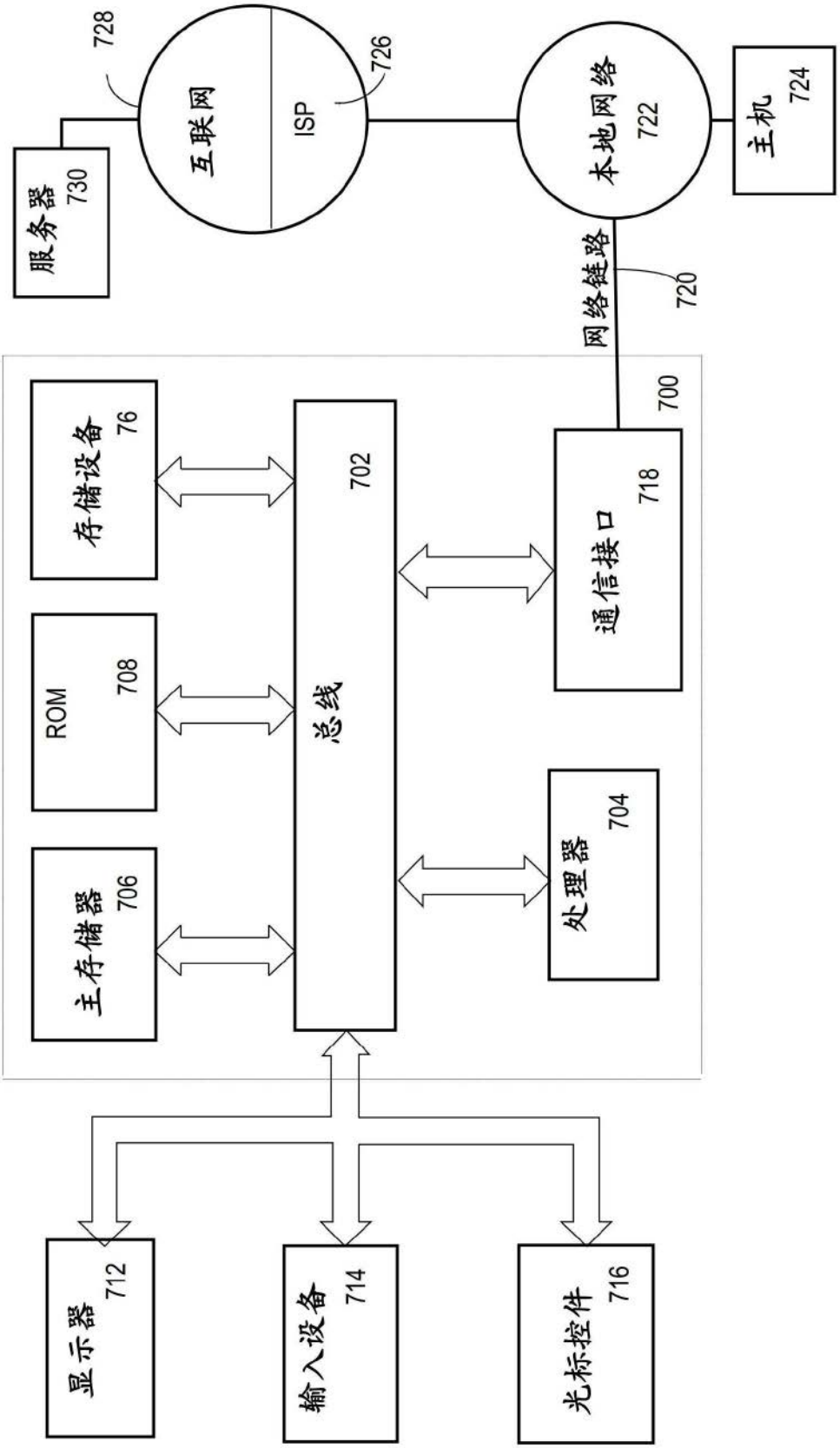

728
互联网
ISP
726
本地网络
722
主机
724
服务器
730
网络链路
720
700
存储设备
76
702
通信接口
718
ROM
708
总线
主存储器
706
处理器
704
显示器
712
输入设备
714
光标控件
716

图7

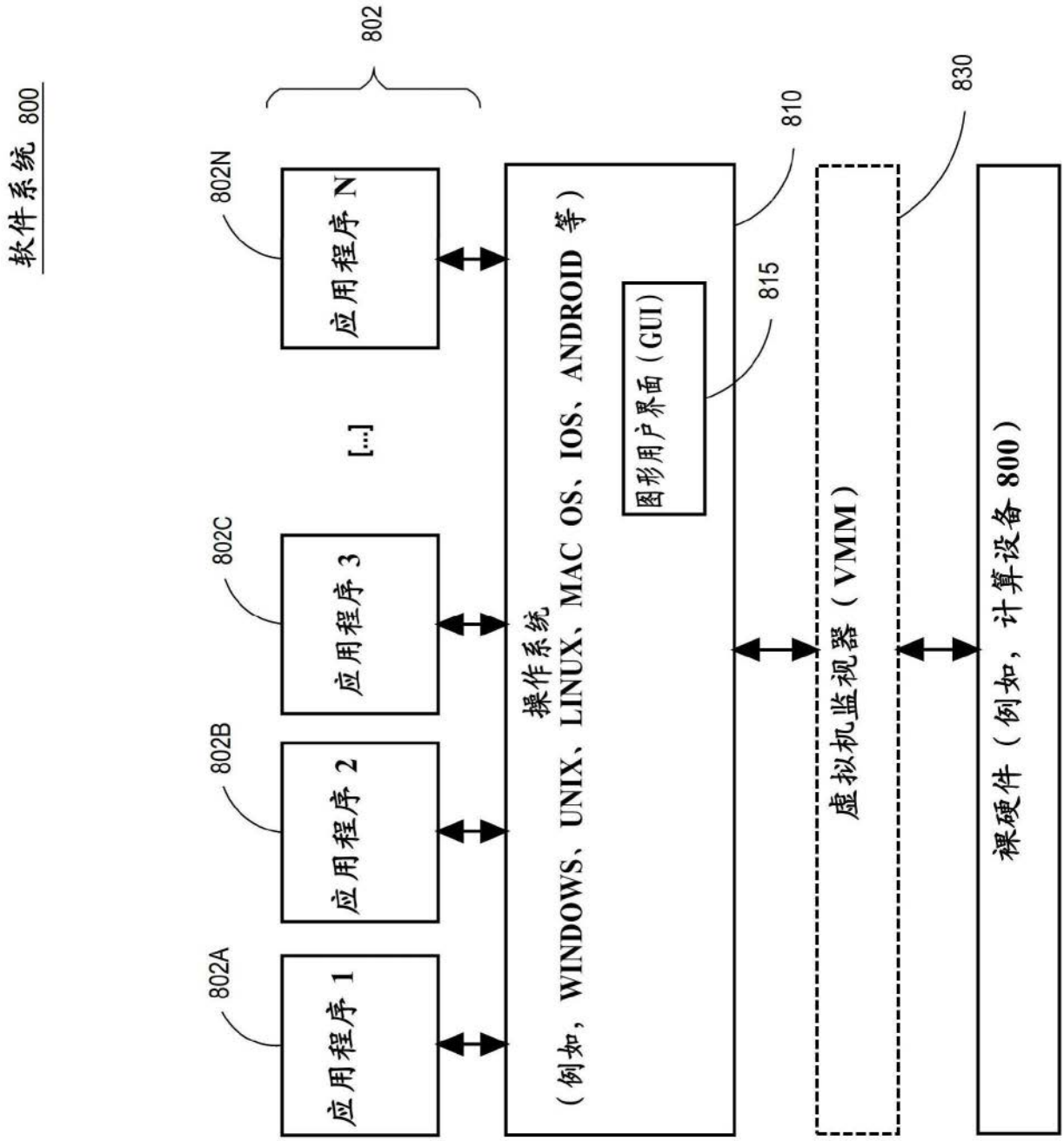

软件系统 800
802
802A
应用程序 1
802B
应用程序 2
802C
应用程序 3
[...]
802N
应用程序 N
操作系统
（例如，WINDOWS、UNIX、LINUX、MAC OS、IOS、ANDROID 等）
图形用户界面（GUI）
815
810
虚拟机监视器（VMM）
830
裸硬件（例如，计算设备 800）

图8