计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201100140

基于 BP 神经网络的智能云效能模型

夏 静 马 中 戴新发 胡哲琨
武汉数字工程研究所 武汉 430205

摘 要 面对日趋庞大和复杂的智能应用,建立有效的云服务质量模型是评价云服务质量的重要手段。然而,由于智能云各层资源的多样性、动态性等特点,智能云服务效能的评估具有很大的难度。针对目前智能云计算领域缺乏标准和统一的云服务质量评价指标和云服务建模手段的问题,文中将智能云抽象的服务质量具体化为云服务效能,云服务效能被定义为反映云服务能力水平的服务可用性、可靠性,以及体现服务效率的性能,即通过云服务效能输出定量的评价智能云的整体服务能力水平。并且提出了一种基于 BP 神经网络的智能云效能模型,通过 BP 神经网络模拟智能云服务的输入特征与服务效能之间复杂的非线性关系,一旦确定输入特征,即可预测输出的服务效能评价指标,这就要求效能模型能够实时并准确地根据系统配置输入特征,预测当前系统的服务能力。实验结果表明,BP 神经网络模型作为智能云服务效能模型的建模工具,具有较好的计算效率和准确率。

关键词:BP 神经网络;智能云;效能模型;服务效能;输入特征
中图法分类号 TP183

Efficiency Model of Intelligent Cloud Based on BP Neural Network

XIA Jing,MA Zhong,DAI Xin-fa and HU Zhe-kun
Wuhan Digital Engineering Institute,Wuhan 430205,China

Abstract Recently,we are facing the increasingly large and complex intelligent applications in cloud computing. Establishing an effective quality model of cloud service is an important methodology to evaluate cloud service quality. However,due to the diversity and dynamic characteristics of intelligent cloud resources,it is very difficult to evaluate the service efficiency of intelligent cloud. At present,there is a lack of standard and unified cloud service quality evaluation and cloud service model in the field of intelligent cloud computing. In this paper,the abstract service quality of intelligent cloud is embodied as cloud service efficiency,and cloud service efficiency is defined as the service availability,reliability and performance reflecting service efficiency. That is to quantitatively evaluate the overall service capability of intelligent cloud through the output of cloud service efficiency. Moreover,this paper proposes an efficiency model of intelligent cloud based on BP neural network. The complex nonlinear relationship between input characteristics and output service efficiency of intelligent cloud is simulated by BP neural network. Once the input characteristics are determined,the output service efficiency can be computed. The efficiency model is responsible for predicting the service level of the current system in real time according to the input characteristics of the system accurately. The experimental results show that the BP neural network model,as a modeling tool of service efficiency model,has good computing efficiency and accuracy.

Keywords BP neural network,Intelligent cloud,Efficiency model,Service efficiency,Input characteristics

高质量的云服务直接影响着用户的体验,也影响着云服务提供商的口碑和信誉。因此,云平台如何提供高质量的智能应用服务成为了近年来云计算领域和人工智能领域的研究热点。而建立有效的云服务质量模型不仅实现了云服务质量的预测,而且对早期云的设计和配置起到了重要的作用。

由于云服务系统资源的多样性、动态性等特点,云服务质量的评估和计算具有很大的难度。因此,目前智能云服务系统缺乏标准和统一的服务建模手段。在云服务系统建模过程中,主要存在以下需要解决的科学问题。

(1)输入。模型的输入是决定模型输出质量的关键要素。智能云服务系统是一个复杂的系统,输入的软硬件资源众多,资源之间存在频繁的高并发调用,并且决定资源组织方式的系统架构对服务质量也有重要影响。因此,如何基于确定的系统构架建立模型的输入特征,是建立服务质量模型的首要问题。

(2)输出。智能云服务系统是一个分布式的复杂动态

到稿日期:2020-11-20 返修日期:2021-01-19 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:热能动力技术重点实验室开放基金资助项目(TPL2019C01)
This work was supported by the Open Fund of Key Laboratory of Thermal Power Technology(TPL2019C01).
通信作者:夏静(673718032@qq.com)

系统，很难用一个特定的系统输出指标来完整地描述云服务质量。因此，如何构建服务质量的输出评价指标体系也是建模的重要内容，即如何建立服务质量的表示模型也是需要从输出层面考虑的重要问题。

(3)模型。对于复杂的分布式动态系统，很难用一个统一的模型完整地模拟输入与输出之间复杂的非线性关系。近年来，各类模型层出不穷，都有各自的优缺点。例如，由于云服务系统的排队特性，很适合利用基于概率的模型进行建模。但是，由于概率模型不确定的特性，对于服务质量要求高的领域，该类模型具有一定的局限性。因此，如何设计高质量的服务质量模型是保障服务质量的重中之重。

因此，本文首先将基于智能服务框架，综合分析各层资源配置对服务效能的影响，建立效能模型的输入特征。其次，本文将抽象的云服务质量输出具体化为云服务效能，云服务效能被定义为反映服务能力水平的服务可用性、可靠性，以及体现服务效率的性能。最后，在输入特征和服务效能评价指标之间，利用 BP 神经网络模型，模拟输入特征与服务效能之间复杂的非线性关系，一旦确定输入特征，即可预测输出的云服务效能评价指标。

本文的主要贡献如下：

(1)为了综合分析智能云各输入要素对云服务质量的影响，本文首先设计了面向高可靠智能应用的智能服务框架系统架构。该架构的基本思想是：智能应用驱动的积木式优化策略搭建，方便为不同类型的智能应用搭建最优化的配置方案；配置管理与功能独立的分层次设计，着力构建结构清晰的服务框架；服务效能驱动的两级控制策略。该智能服务框架为建立高质量的效能模型提供了框架层面的保障。

(2)基于舰载智能服务框架的系统架构，从计算资源特征、存储资源特征、网络资源特征、虚拟化配置特征、智能化配置特征、运行时环境特征 6 个方面提取影响云效能模型的输入特征；并且构建了评估指标“整体服务效能”的表示模型，将抽象的服务质量具体化为云服务可用性、可靠性及服务效率。

(3)提出了一种基于智能服务框架的系统架构，利用神经网络模型可逼近任意复杂非线性关系的特性，基于 BP 神经网络模型，模拟系统输入特征与服务效能的复杂非线性关系，建立基于 BP 神经网络的效能模型。

1 相关工作

2020 年，因为新冠疫情的爆发，各行业线上业务对云服务的需求呈指数级增长，而用户选择云厂商的标准是其云服务质量水平的高低。然而，由于云计算本身处于发展上升期，其相关的技术标准、协议体系、支撑理论尚未成熟，现有理论和技术方案仍存在诸多局限性，业界缺乏对云服务系统服务质量进行精确分析的方法和预测模型。

目前国内外学者对云服务质量建模方面的研究，主要包括对云服务系统进行建模，以及对云服务质量评价指标进行建模两大类。

1.1 云服务系统建模的相关研究

云服务系统建模方面的研究主要分为以下几类。

(1)基于随机 Petri 网和排队理论建模。由于随机 Petri 网和排队模型方便模拟云服务系统中的高并发调用、互操作以及非确定性等特性，因此常基于随机 Petri 网和排队理论对云服务系统建模。Schunselaar 等[1]基于 Petri 网的工作流程建模符号，对云基础设施层建模，探索可以提升服务效率的建模方法。文献[2-3]基于随机 Petri 网对 IaaS 云服务进行了可靠性指标的建模，并预测了系统的整体生存率。文献[4-6]分别利用排队理论、贝叶斯理论、概率统计等方法，从时间性能角度评价了云服务质量。但是仅从时间性能角度评价云服务质量是远远不够的。并且基于随机 Petri 网建立的模型过于粗糙，未考虑系统架构对服务质量的重要影响。

(2)基于用户评价进行建模。文献[7]基于加权层次方法和用户评价对云服务质量建模。文献[8]基于用户行为信任度提出了服务选择模型，帮助用户选择符合其需求的云服务提供商。文献[9]从功能适应度、人性化、可信度、价值等角度构建云服务评测体系，即考虑主观因素对模型的影响建立了云服务模型。该类建模方法主要通过用户的评价和打分进行建模，进而实现对云服务质量的评价。但是，该类方法过于主观，忽略了云服务系统中客观的软硬件要素对服务质量的重要影响。

(3)基于异常检测机制建模。由于异常检测机制是保障服务水平的重要手段，因此文献[10]基于 Amazon EC2 云平台，利用控制理论开展了云服务质量测评研究，建立了初步的模型。文献[11]通过设计自适应的异常检测机制，保障了云平台的服务质量。文献[12]针对虚拟机迁移过程，提出了一套保证云服务质量的管理机制。文献[13]基于服务异常和失效服务率，利用马尔可夫模型对云服务质量进行评价。但是，以上研究并未给出精确的服务质量模型，且未对影响服务性能的效率进行定量分析，仅仅探讨了影响云服务质量的一些关键要素。

(4)基于基础资源特征建模。云平台的基础资源是保障云服务的基础。因此，通过研究影响服务质量的基础软硬件资源特征也是建立模型的突破口。文献[14-16]全面分析了影响服务水平的软硬件资源等特征，建立了树状结构模型以评价服务质量。文献[17-19]通过对物理资源、网络资源、系统架构以及虚拟机资源等进行分析，构建了多目标优化模型，以实现对云服务的评价。但是，以上文献对云服务质量的输出指标缺乏分析且评价输出指标过于单一。

(5)基于智能算法对复杂系统建模。动车组运行控制系统基于智能算法对系统建模，以保证高速动车组有序安全地运行。近年来，该行业较多地利用智能模型对动车组系统建模，其主要思路是利用智能模型可模拟任何复杂非线性关系的能力，实现对动车组系统输入要素与输出目标对象之间关系的模拟。经过实际的检验，该类基于智能算法的建模手段在该领域已经趋于成熟。文献[20-22]均是利用智能算法实现对动车组控制系统的建模，从而保证动车运行控制系统的高效、可靠运行。

综上，目前对云服务系统的建模缺乏统一标准的规范和建模手段，尤其是对于更复杂的智能云服务系统建模，更是鲜有研究。智能云多是基于普通的云服务系统扩展建设而成，但是在应用类型和软硬件资源方面都有不同的扩展。因此，

挖掘并模拟智能云服务质量输入和输出的复杂非线性关系更为复杂。

以上文献对本文设计的智能云模型的启发在于：1）如何从模型输入层面提取智能云各层重要软硬件资源的输入表示特征；2）智能云系统架构决定了各资源要素的组织形式，因此建模时不能忽略系统架构对模型的重要作用；3）智能算法在模拟复杂非线性关系时具有强大的拟合能力。

1.2　云服务质量相关研究

由于对云服务系统建模具有一定的难度，且不同的云服务系统从架构到服务对象都具有一定的差异性。因此，该行业一直缺乏统一的建模手段和遵循标准。但是，对云服务质量的研究却相对成熟，即从云服务系统输出层建立评价指标模型。近年来，结合云服务可用性和可靠性对保障云服务能力水平的重要性，以及服务性能对用户体验的重要性，将对云服务质量评价指标的建模主要分为对云可用性、可靠性、服务性能的建模。

1.2.1　云服务可用性相关研究

国内外学者在可用性模型的研究方面做出了很大的贡献。其研究主要分为以下几类：

（1）基于单一指标建立可用性评测模型。文献[23]基于可用性的定义，用虚拟机热迁移时间的均值替代系统平均恢复时间（Mean Time To Repair，MTTR），以计算得到云服务的可用性。文献[24]用虚拟机的启动时间替代 MTTR，以量化云平台的可用性，建立云可用性评测模型。文献[25]利用串联方式对云计算三层服务可用性进行建模，但是实际上其模型结构过于简单。以上方法对可用性进行建模，基本都是按照可用性定义用某些单一的指标近似替代可用性定义里的相关参数，并未建立云计算平台重要输入参数与可用性的复杂内在联系。因此，以上文献中给出的可用性模型实际上都只是可用性的一种表示，即一种可用性计算方式，并未给出云计算平台重要输入组件与可用性的内在联系。

（2）基于优化策略提升系统可用性。文献[26]利用通信优化和负载均衡技术优化了负载调度算法，提升了云平台的高可用性，但是并未量化云平台的可用性。文献[27]采用可靠性框图法，基于 IaaS 硬件层，构建了云平台的 SLA 制定，但是并未量化云平台的可用性。文献[28]利用贪心算法将安全性和可用性融入任务调度，以提升平台的安全性和可用性，但并未给出云平台可用性的量化模型。文献[29]以提升云平台的高可用性为目标，利用系统架构设计和调度策略优化，探讨了影响系统高可用性的关键要素，但是也未给出量化云平台可用性的模型。

（3）基于高可用设计提升系统可用性。文献[30]在 OpenStack 平台上设计了高可用性集群，以提升平台的高可用性，但是也未给出平台可用性的量化模型。文献[31]基于马尔可夫链概率模型，对大型 IaaS 云平台的可用性建立了随机模型，设计了高可用的 IaaS 层云服务。文献[32]基于 OpenStack 和分布式存储 Ceph，通过优化策略实现了节点的高可用，但是也未给出平台可用性的量化模型。以上文献虽然没有对云平台可用性进行建模，但是探讨了影响云计算平台的关键要素，对本文输入特征选择有一定的启发意义。

综上所述，对云服务可用性进行综合、有效的测评面临许多挑战，但是也为智能云服务可用性建模提供了重要的参考标准。

1.2.2　云服务可靠性相关研究

国内外学者在可靠性模型的研究方面做出了很多有益的探索。其研究主要分为以下几类。

（1）基于单一指标或资源类型研究系统的可靠性。文献[33]利用排队模型，基于 Hadoop 作业调度算法，建立了集群系统的可靠性模型。文献[34]基于博弈论方法，基于任务调度算法提升了云计算系统的可靠性，并给出了最优化的系统配置求解算法。文献[35]通过分析云计算平台网络资源的配置策略，基于树状网络结构建立了系统可靠性的数学模型。以上文献在可靠性建模方案中考虑的输入特征过于单一，且适用于各自的云服务系统配置环境不具备良好的模型迁移能力。但是，其考虑的输入资源特征具有很好的参考价值。

（2）基于资源故障研究云服务的可靠性。文献[36]通过研究云平台的主要故障类型和模式，基于概率模型建立了云服务的可靠性模型。文献[37]基于电源故障问题，通过整合调度算法，保障了云服务的可靠性。文献[38]分析了硬件冗余方案，可降低硬件故障对云服务可靠性的影响。文献[39]分析了硬件冗余管理方案与云服务可靠性的关系。文献[40]验证了硬件故障对云服务可靠性的重要程度。文献[41]分析了内存、硬盘、处理器等关键组件的故障特征和类型，其目的在于提升系统的服务可靠性。但是，其并未形成完善的云服务可靠性模型，仅仅从软硬件资源故障的角度试图建立与系统性能之间的关系是远远不够的。

（3）基于层次分析法或阶段分析法研究云服务的可靠性。文献[42]基于排队模型和蒙特卡罗方法，对云服务的请求阶段和执行阶段分别建立可靠性模型，以保证数据中心服务的可靠性。文献[43]基于排队模型和图论，对服务的请求阶段和执行阶段建立服务可靠性模型，并通过实例进行验证。以上文献未对子模型和子阶段模型的关联进行详细分析。但是，其分层和分阶段的输入特征采集对本文的可靠性分析具有指导意义。

本节探索了云服务质量建模方面的研究现状，这些建模方法中缺乏对云服务质量输出的明确定义，从而导致对云平台本身的质量监控不到位；接着，介绍了云服务系统可用性、可靠性的建模研究现状，并分析总结了现有建模方法各自的优缺点。

2　面向高可靠智能应用的智能服务框架

为了综合分析智能云各输入要素对云服务质量的影响，本文首先设计了面向高可靠智能应用的智能服务框架。

本文提出的智能服务框架系统架构的基本思想是：智能应用驱动的积木式优化策略搭建，方便为不同类型的智能应用搭建最优化的配置方案；配置管理与功能独立的分层次设计，着力构建结构清晰的服务框架。该智能服务框架为建立服务效能模型提供了框架层面的保障。本文提出的面向高可靠智能应用的智能服务框架的系统架构如图 1 所示。

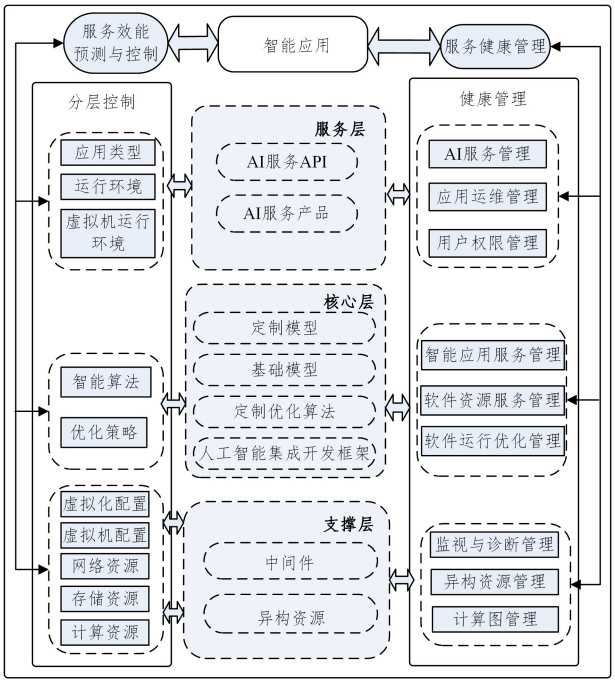


图 1 智能服务框架的系统架构

Fig. 1 System architecture of intelligent service framework

(1)针对不同类型的典型智能应用,基于人工智能集成开发框架中的资源分配方式、算法实现方式、节点协同方式等进行重新定义,通过定制与智能应用适配的优化方案,从资源层面保障高质量的智能应用服务。

(2)分层次地对各层资源的运行状况进行实时监控与管理,以期对面向智能应用的各软硬件要素实现健康管理。并且相比复杂的集中管理与监控,本文构架通过分层定义各层管理接口,保证了智能应用的高可靠性。

(3)融入了服务效能预测与优化控制的新控制架构,依托云平台提供的丰富算力,有力支撑复杂控制策略实施以及智能控制任务实施等原有系统难以承载的任务类型。

2.1 支撑层

支撑层在整个智能服务框架的系统架构中起着重要的基础支撑作用,主要包括异构资源子层和中间件层。

(1)异构资源子层

异构资源子层需高效兼容各类异构软硬件资源,有效实现资源的动态流转,利用 CPU,GPU,FPGA 等高性能计算部件,集成海量存储资源和多种高速可扩展互连总线接口,通过云操作系统等,最终构成异构资源加速平台。其中的异构硬件层架构如图 2 所示,主要由若干异构计算、存储和管理等基本单元构成。管理单元通过千兆网络等管理总线与每个单元内部的管理部件连接,计算、存储等单元通过高速互连网络进行连接。而硬件资源抽象和封装主要利用云操作系统面向云计算设备,通过计算虚拟化、存储虚拟化、网络虚拟化等技术为用户和多种应用提供强大的计算、存储、处理、实时控制等服务,具有强大的异构资源管理能力和资源动态扩展能力。最终实现以全自动弹性模式、按需、敏捷地获取各类软硬件资源服务,最终目标是推动智能云服务的投入产出比最优化。因此,本文的所有设计均是基于异构资源子层的特性,以最大限度地发挥资源使用效率。

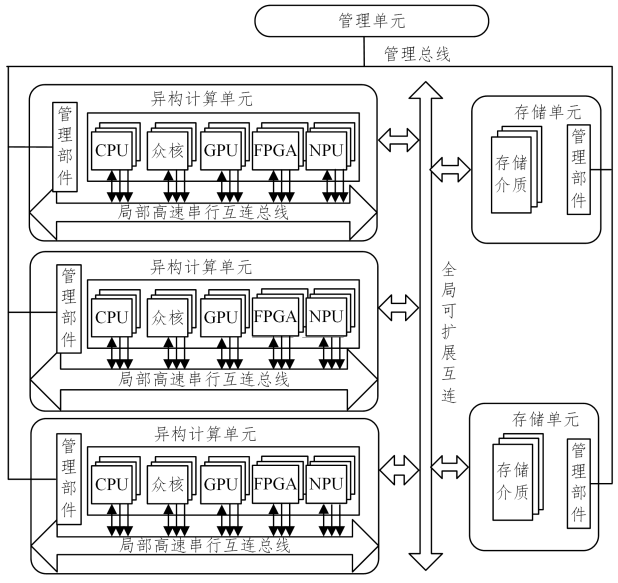


图 2 异构硬件层架构

Fig. 2 Heterogeneous hardware layer architecture

(2)中间件层设计

中间件层为人工智能集成开发框架的高效运行提供了关键支持,主要包括 MPI 并行计算库、神经网络加速库、基础数学库 BLAS、异构计算库 OpenCL 等。结合人工智能集成开发框架对异构计算中间件和底层支持库的要求,需要针对智能计算平台移植并调优与分布式智能计算相关的支持库,一方面为智能计算的加速提供支撑,另一方面提升硬件的使用效率。

2.2 核心层

核心层是面向智能应用的核心支撑平台,其驻留在支撑层之上,为智能应用提供公共运行环境;该层基于标准、开放的人工智能集成开发框架,根据智能业务类型和特定需求合理分配系统算力,采取面向智能应用的定制优化算法,通过提升资源的利用效率,最终提供高可靠的智能应用服务。图 3 给出了核心层细化后的架构。

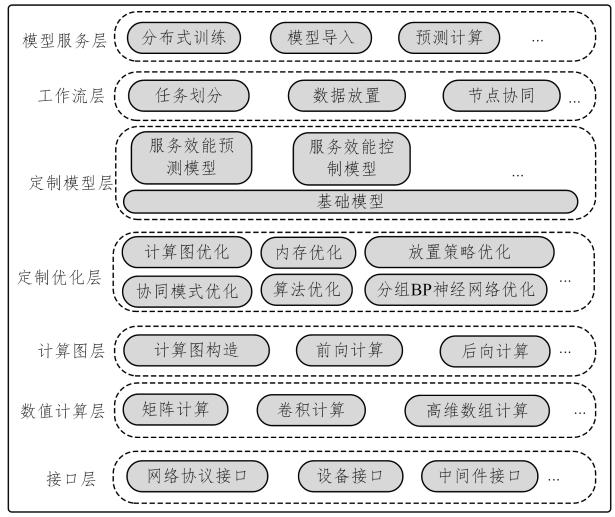


图 3 核心层架构

Fig. 3 Core layer architecture

核心层需结合基础硬件资源平台的异构资源特性,尤其

是高效计算资源的访存特性等特点，提供高效的智能软件运行框架，最终面向智能应用提供最优的服务效能。下面对核心层架构涉及的各子层进行介绍。

(1)接口层。该子层需结合硬件资源平台异构资源的特性，完成与硬件资源接口和中间件接口的衔接。

(2)数值计算层。该子层支撑线性计算、卷积计算等神经网络最基础的计算，以及负责处理高维复杂数组形式的计算。

(3)计算图层。该子层是实现分布式计算的关键子层，其中的计算图划分算法决定了计算任务在分布式系统节点之间的分配方式，目的是让分布式系统中的各类计算节点尽可能均衡地共同完成任务。

(4)定制优化层。该子层针对不同的典型智能应用类型，对人工智能集成开发框架中的资源分配方式、算法实现方式、节点协同方式等进行重新定义，通过定制与智能应用适配的优化方案，最大限度地让各类资源“物尽其用”，最终提供高可靠的智能应用服务。

(5)定制模型层。针对典型智能应用快速定制与便捷开发的需求，该层需根据典型应用领域的特点，设计各类智能模型，通过定制模型层的设计，构造面向典型智能应用的快速定制开发套件和模型，提升人工智能集成开发框架的快捷开发能力。

(6)工作流层。该子层主要为分布式计算过程中涉及工作流的任务划分策略、数据放置策略以及节点协同等提供调优。

(7)模型服务层。该子层向上层提供服务的接口，主要包括分布式训练、模型导入以及推理计算等服务。

因此，核心层在整个服务框架系统架构中起着关键作用，是人工智能任务的编程接口与智能应用承载平台。该层是基于现有的商用主流框架，通过面向典型智能应用构建定制开发工具箱，并以与开发环境无缝集成的形式，最终构建“计算高效、安全可靠、便捷易用”的人工智能集成开发框架。

2.3　服务层

服务层以多层级用户快速进行应用开发部署为导向，着力构建面向典型智能应用的快速服务产品，提高系统面向集成开发用户、资源用户的智能服务能力。该层不仅可提供 AI 容器服务、AI 服务虚拟机、AI 在线服务等产品式的智能服务，还可提供 AI 训练 API 服务和 AI 预测 API 服务。

2.4　控制与健康管理

本文设计的智能服务框架系统架构通过对服务效能的预测与控制，以及服务健康管理两大手段，为智能应用提供高可靠的服务保障。

(1)分层监控与控制

对服务提供商而言，需要设计可靠且有效的控制策略，以保证智能云服务效能维持一个较高的水平。因此，本文设计的智能服务框架系统架构融入了对云本身服务能力水平的预测和控制，以提升智能应用的服务效能。图 1 中的左边部分展示了智能应用控制接口架构。该接口架构主要体现了分层控制的设计思路。并通过分层监控为服务效能的预测与分析提供输入特征。而控制策略的有效实施则依赖于对服务效能的预测与分析。

具体来讲，从支撑层的硬件资源配置到服务层同时运行的虚拟机数量等都是影响智能应用服务效能的重要输入，因此控制接口与每层的关键要素直接关联，即通过分层控制方法，构建逻辑清晰的控制接口规范，设计功能完备的系统级控制管理接口簇，自下而上地为服务提供商提供便捷的控制接口。并且，便于根据服务效能模型自上而下地控制各层资源，保障智能云的服务效能。通过这种双向管理与映射方式，形成分层次、多粒度、结构化、多形式的控制管理接口体系，为智能云服务提供高效支撑。

(2)服务健康管理

通过服务健康管理接口可为应用开发和系统管理提供支撑。并且通过进一步标准化健康管理接口的迭代优化与设计实现，有助于形成高质量的完备接口规范，推动云系统服务向高可靠、通用化方向发展。

架构通过分层次地设计每层要素的健康管理接口，相比复杂的面向整个云的集中式管理，更易操作和实施。因此，服务框架将各软硬件要素的健康管理纳入到系统架构中，融入了健康管理的服务框架更便于智能运维和故障诊断的设计与实现。本方案构建的服务健康管理接口簇如图 4 所示。分层次、分粒度的健康管理是本文设计的智能服务框架的新特性，也是保障服务效能的有效手段。

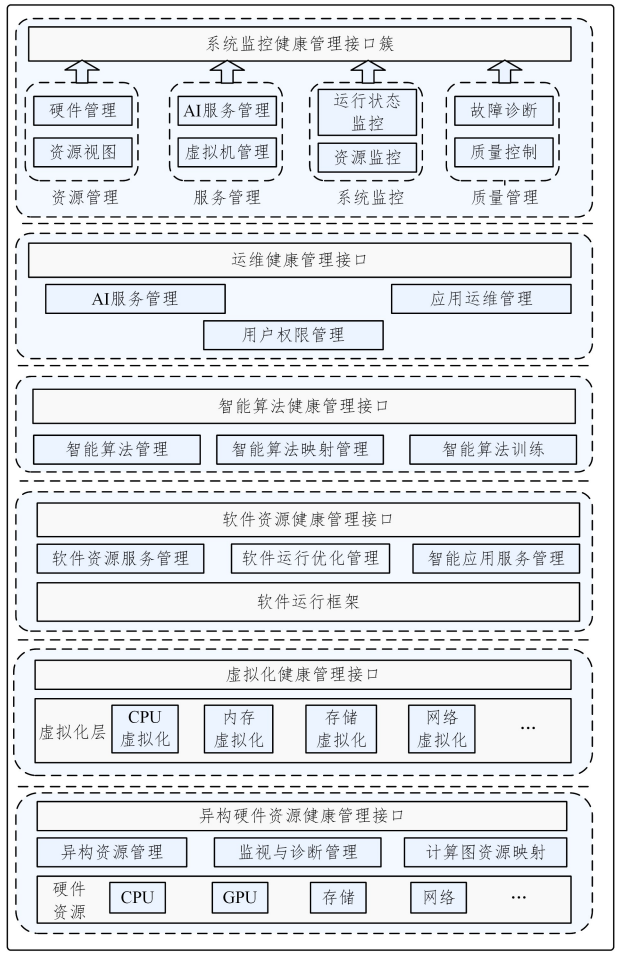


图 4　智能应用健康管理接口簇

Fig. 4　Interface cluster of intelligent application health management

2.5　本节小结

本节设计智能服务框架的目的是为建立高质量服务质量

模型提供框架层面的保障。该架构的基本思想是:智能应用驱动的积木式优化策略搭建,方便为不同类型的智能应用搭建最优化的配置方案;配置管理与功能独立的分层次设计,着力构建结构清晰的服务框架;服务效能驱动的控制策略,保证智能云服务的质量。

3 基于 BP 神经网络的云服务效能模型

本节基于智能服务框架的系统架构,从计算资源特征、存储资源特征、网络资源特征、虚拟化配置特征、智能化配置特征、运行时环境特征 6 个方面提取影响效能模型的输入特征。并构建了评估指标"整体服务效能"的表示模型,让抽象的服务能力水平以定量的方式得以评估。为了实时而准确地根据输入特征预测整体服务效能,本文利用神经网络模型可逼近任意复杂非线性关系的特性,基于 BP 神经网络模型构建智能云效能模型。

3.1 基于智能服务框架的输入特征选择

智能服务框架每一层的资源配置都涉及到诸多特征参数,且层内和层间各个特征之间主要是复杂非线性关系。并且上层运行不同类型的智能应用,如分别运行内存密集型应用或 I/O 密集型应用时,可能表现出不同的服务效能。因此,需要分析和测试影响效能模型的输入特征参数。本节将基于智能服务框架的系统架构,综合分析各层资源配置对服务效能的影响,建立评价基于智能服务框架效能模型的输入特征。

支撑层主要提供基础设施保障,该层影响整体效能模型的因素主要有各类硬件参数及虚拟机配置相关参数,如开机状态的服务器数量、每个服务器的 CPU 内核数、CPU 微架构和主频、I/O 带宽、待机状态的服务器数量、硬盘容量、内存容量、同时运行的虚拟机数量、虚拟机迁移策略等配置都会影响整体效能模型的输出服务效能;核心层会根据上层应用选择不同的优化策略,因此优化策略也是影响整体服务性能的重要特征,并且智能模型的规模也是决定计算效率的关键要素,因此在不同的应用目标驱动下,选取的智能模型的规模对服务效能也有极大影响;服务层由于不同的服务提供形式,所分配的资源会有所不同,然而其他层已详细地分析了各层资源对性能的影响。因此,本文不考虑服务呈现层对性能的影响。但是不同智能应用的资源访问偏好因素,如属于内存密集型的应用和 I/O 密集型的应用也会影响服务的整体效能水平。综合考虑待评估的系统架构中各层的配置参数,以上影响整体服务效能的输入特征参数可被具体分类为计算资源特征、存储资源特征、网络资源特征、虚拟化配置特征、智能化配置特征、运行时环境特征 6 个方面。各类资源的输入特征具体如下。

(1)计算资源特征:主要包括 CPU 类型、CPU 主频、开机状态的服务器数量、待机状态的服务器数量。

(2)存储资源特征:主要包括硬盘容量和内存容量。

(3)网络资源特征:主要包括 I/O 带宽。

(4)虚拟化配置特征:主要包括虚拟机迁移策略、vCPU-CPU 绑定方式。

(5)智能化配置特征:主要包括框架优化算法、智能模型规模。

(6)运行时环境特征:主要包括平台上同时运行的虚拟机数量、是否有正在训练的模型以及虚拟机上运行的负载类型。

输入特征选择是建立智能云效能预测模型的前提,根据特征的可能取值,配置得到一组效能模型的输入特征 D,D 中的特征需包含所选取的输入特征的所有取值,并且每个特征在 D 中的取值在该特征的值域范围内均匀分布,最理想的情况是每个可能取值均覆盖,使得建立的模型更精确,准确率更高。当值域范围连续或者较大时,可以均匀选取一组离散值来替代。

3.2 服务效能定义

本文定义智能云服务效能来表示云服务质量。然而,服务效能是抽象的,如何将效能具体化,即采用何种评价指标作为云服务效能的表示。国家标准与技术研究所(NIST)对云计算的定义中明确说明:云计算是一种通过组织各类资源来提高其系统可用性的一种计算模式。目前国内外主流云计算厂家也将云服务可用性作为评价云服务质量水平的主要标准。云服务可用性指云服务在满足用户的服务需求时,所具有的有效性、满意度和效率。因此,可用性是评价云服务效能的关键指标,即云服务的主要目标是提高系统的可用性。并且,目前云厂商在提供给用户的服务水平协议中也明确说明了各自的可用性。因此,本文首先将云服务可用性作为整体服务效能的第一个重要评价指标。

对用户而言,无法直接测评云服务的可用性是否达到承诺的标准,他们更关注系统的可靠性和服务的实时响应时间。具体来讲,传统的可靠性标准定义为在系统运行过程中,即执行任务期间,满足用户需求的能力。云服务的可靠性直接表现为服务提供商承诺为用户提供计算、存储、网络等资源服务的保障能力;而实时响应时间是服务效率最直接的表现。因此,本文将可靠性作为整体服务效能的第二个重要评价指标,服务效率作为整体服务效能的第三个重要评价指标。

本文将可用性、可靠性和服务效率作为整体服务效能的 3 个评价指标,即服务效能用一个三元组(可用性、可靠性、服务效率)表示,其完整地诠释了智能云服务的质量水平。

3.2.1 服务效能的可用性指标定义

对于企业来说,首先关注云服务的可用性指标。目前,各主流云服务商都对云服务可用性进行了定义,阿里云服务可用性计算式如式(1)所示,华为云服务可用性计算式如式(2)所示。

$$\text{阿里云服务可用性}=\frac{T_1}{T_1+T_2}*100\% \tag{1}$$

$$\text{华为云服务可用性}=\frac{(T_1-T_2)}{T_1}*100\% \tag{2}$$

其中,式(1)的 T_1 代表每个服务周期单台云服务器所有的可用时间,T_2 代表每个服务周期单台云服务器所有的不可用时间。式(2)中的 T_1 代表该服务周期的总时间,T_2 代表该服务周期的不可用时间。

阿里云服务可用性和华为云服务可用性都侧重于从硬件层面评价底层服务器的可用性。然而,影响服务可用性的不仅包括硬件层面的服务器,还包括各类软件配置策略和智能优化策略等。智能云服务系统中服务层提供服务的形式

多样，包括 AI 服务虚拟机、AI 服务软件、AI 在线服务、AI 容器服务、AI 服务 API。因此，针对服务层提供的云服务，本文将云服务可用性 A 定义为式(3)。一个服务周期内，不可用时间指无法为用户服务的时间，而不可用性是指不可用时间与服务周期的比率。因智能云向用户提供的各类虚拟化服务互不影响，因此本文将基于不可用性均值来评价整体云服务的可用性。

$$A=1-(\alpha\bar{F}_1+\beta\bar{F}_2+\gamma\bar{F}_3+\theta\bar{F}_4+\mu\bar{F}_5) \tag{3}$$

其中，A 是智能服务框架的云服务可用性，$\bar{F}_1$ 是服务周期内所有 AI 服务虚拟机的不可用性均值，$\bar{F}_2$ 是服务周期内所有 AI 服务软件的不可用性均值，$\bar{F}_3$ 是服务周期内所有 AI 在线服务的不可用性均值，$\bar{F}_4$ 是服务周期内所有 AI 容器服务的不可用性均值，$\bar{F}_5$ 是服务周期内所有 AI 服务 API 的不可用性均值。$\alpha,\beta,\gamma,\theta,\mu$ 分别代表 $\bar{F}_1,\bar{F}_2,\bar{F}_3,\bar{F}_4,\bar{F}_5$ 的权值，我们采用层次分析法确定权值。

首先，建立指标的层次结构模型，其结构模型如图 5 所示。其次，设定 n 阶比较矩阵 $\boldsymbol{D}$，比较矩阵根据参数的重要程度人为赋值。然后，根据比较矩阵 $\boldsymbol{D}$ 的特征值求得特征向量 $\boldsymbol{\omega_D}$，再根据式(4)求得比较矩阵的不一致程度值 CI，通过查验资料获得比较矩阵 $\boldsymbol{D}$ 的不一致性指标并获得不一致性比率。若矩阵获得满意的一致性，则 $\boldsymbol{\omega_D}=(\alpha,\beta,\gamma,\theta,\mu)$；否则重新调整比较矩阵 $\boldsymbol{D}$，直到获得满意的一致性。

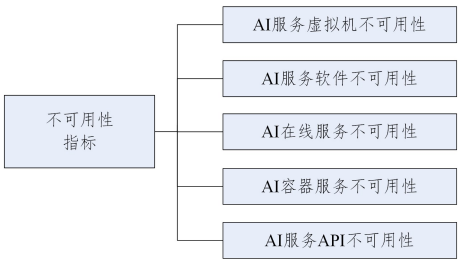


图 5　不可用性指标的层次模型

Fig. 5　Hierarchy model of unavailability

$$CI=\frac{\lambda_{\max}(\boldsymbol{D})-n}{n-1} \tag{4}$$

$(\alpha\bar{F}_1+\beta\bar{F}_2+\gamma\bar{F}_3+\theta\bar{F}_4+\mu\bar{F}_5)$ 是服务层的各服务类型对应不可用性的加权平均。而单个服务的不可用性如式(5)所示：

$$F=\frac{MTTR}{MTTF+MTTR},\text{if } MTTR \tag{5}$$

其中，MTTF(Mean Time To Failure)是单个服务的平均无故障运行时间，即服务每两次故障出现的平均间隔时间；MTTR 是单个服务的平均修复时间，即服务恢复故障所需时间。其中，不可用性的阈值 ε 决定了不可用性的计算粒度，只有当连续不可用时间(无法提供服务并且无法正常恢复)超过 ε，才纳入不可用性的计算，低于 ε 的不可用时间不计算在内。例如，$\varepsilon=1$ min，通过系统日志确定超过 1 min 后无法提供服务的数据，对于超过 1 min 的不可用时间进行不可用性计算；而低于 1 min 的不可用时间则不纳入不可用性计算。

而各服务类型对应的不可用性均值则是该服务类型下所有服务实例不可用性的均值。以提供 AI 服务虚拟机的服务类型为例，假设该服务类型下的服务实例个数为 n，则该服务类型对应的不可用性均值的计算式如式(6)和式(7)所示。式(6)未考虑权重信息，AI 在线服务和 AI 服务软件的不可用性均值计算均可采用该计算式；而式(7)考虑了权重信息，AI 服务虚拟机、AI 容器服务、AI 服务 API 的不可用性计算均可采用该计算式。具体使用时，根据各子系统的级别和重要程度对权重赋值。

$$\bar{F}_{vm}=\frac{1}{n}\sum_{i=1}^{n}F_{vm_i},\text{if } T_{F\text{-}vm_i}>\delta \tag{6}$$

$$\bar{F}_{vm}=\sum_{i=1}^{n}\alpha_i F_{vm_i},\sum_{i=1}^{n}\alpha_i=1\&\&,\text{if } F_{vm_i}>\delta \tag{7}$$

其中，α_i 为应用 i 提供服务的 AI 服务虚拟机 vm_i 对应的权值，该值可以根据应用 i 的重要性程度而被赋予不同的权值，也可采用层次分析法获得。

本文基于可用性定义，对智能云的整体可用性进行分析和定义，并将其作为整体服务效能输出的重要表示特征。

3.2.2　服务效能的可靠性指标定义

云计算服务可靠性直接表现为服务提供商承诺为用户提供计算、存储、网络等资源服务的保障能力。因此，用户对可靠性的关注超过了服务提供商对其承诺的服务可用性。作为一个负责任的云服务提供商，将可靠性作为服务效能的另一个指标无疑是至关重要的。

云计算服务系统将云服务可靠性定义为式(8)，其在概率的数学基础上给出了详细的定义。其中，$R(t)$ 代表云服务可靠性；m 代表用户提交的请求被划分为子任务的个数；$P_B^{(m)}$ 代表用户服务被阻塞的概率；$T_{SRT}^{(m)}$ 代表用户请求从发起到完成的时间，即服务响应时间；t 代表指定的时间；P_r 代表规定时间内云平台满足用户服务需求的概率。云服务可靠性定义的内在涵义指用户提交的请求在规定时间 t 内能够完成的概率。

$$R(t)=(1-P_B^{(m)})P_r\{T_{SRT}^{(m)}<t\} \tag{8}$$

对于智能云而言，基于概率的定义具有一定的随机性，这不符合对服务的高可靠要求。可靠性一般包括硬件可靠性、软件可靠性、系统可靠性。而智能云为用户提供的服务形式主要为虚拟机，因此，本文将结合传统意义上对可靠性的定义和分类，重新对服务效能的可靠性进行定义。从虚拟机服务能力、虚拟资源隔离性、平台资源管理能力、虚拟机性能折损 4 个方面对服务效能的可靠性进行重新定义。通过对各级测评指标对应的属性进行分类整理，得到表 1 所列的服务效能的可靠性评价体系。

表 1　服务效能的可靠性评价体系

Table 1　Reliability evaluation system of service efficiency

测评项目	对应属性	测评工具
虚拟机服务能力	CPU 性能	SPEC
	内存性能	Stream
	网络性能	Netperf
	磁盘性能	Iozone
虚拟资源隔离性	测评不同虚拟机之间的相互影响	Unixbench
平台资源管理能力	资源扩展能力	平台功能
	虚拟资源扩展能力	平台功能
	虚拟机迁移能力	平台功能
虚拟机性能折损	与同配置物理机的性能差距	Unixbench

(1)虚拟机服务能力。虚拟机是云平台服务的重要代表,其服务能力直接体现了虚拟化性能。处理器性能、内存性能、网络性能、磁盘性能4个要素全面反映了虚拟机的服务能力水平。因此,云服务能力水平首先关注了云平台的重要代表——虚拟机的服务能力。

(2)虚拟资源隔离性。虚拟化技术是云平台的关键技术,它可让多个虚拟机运行在同一台物理机上,提高资源使用率,但是由于对物理资源的争用,彼此性能难免受到影响。虚拟资源隔离性是评价虚拟机之间相互影响的指标,即我们通过计算性能隔离性、网络性能隔离性和磁盘性能隔离性,评价共享物理资源对虚拟机的性能影响。因此,平台云服务能力水平的另一个重点关注的指标就是虚拟资源隔离性。

(3)平台资源管理能力。合理的资源管理策略在一定程度上决定了云平台的服务水平。由于扩展性是云平台的另一个重要特征,因此,本文重点对资源扩展能力、虚拟资源扩展能力、虚拟机迁移3个主要涉及资源扩展能力的要素进行测评。

(4)虚拟机性能折损。由于各虚拟机对物理资源的争用,以及虚拟化软件的存在,虚拟机相比同配置的物理机,有一定程度的性能损失。合理范围内的性能折损是可接受的,而超出合理范围内的性能折损则会严重影响云平台的服务能力水平。因此,合理地评价虚拟机性能折损对云平台服务能力水平具有重要意义。

下面给出可靠性定义的详细过程:

(1)数据归一化处理。可靠性评价体系中各测评项目的值域区间和单位均不同。因此,在对可靠性进行定义前,首先需要进行归一化处理。本文采用式(9)对测评数据进行归一化处理。其中,S_x表示各项指标下属性测评的得分x的归一化值,x_{avg},x_{min},x_{max}分别表示多次测量x获得的平均值、最小值和最大值。

$$S_x=\frac{x_{avg}-x_{min}}{x_{max}-x_{min}} \tag{9}$$

(2)构建可靠性层次结构模型。利用层次分析法,建立基于智能服务框架可靠性指标的层次结构模型。其结构模型如图6所示。

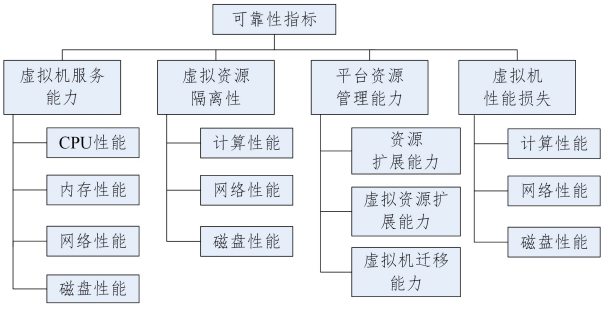


图6 可靠性指标层次模型

Fig. 6 Hierarchy model of reliability

(3)建立比较矩阵$\boldsymbol{D}$。构建4阶比较矩阵$\boldsymbol{D}$,比较矩阵的值由我们根据各测评项目的重要程度决定。例如,我们认为虚拟机服务能力与虚拟资源隔离性的重要性之比为6,平台资源管理能力与虚拟机资源隔离性的重要性之比为3,虚拟机性能损失与虚拟机资源隔离性的重要性之比为2。

(4)权值求解。$\lambda_{max}(\boldsymbol{D})$代表该矩阵$\boldsymbol{D}$的最大特征向量。根据式(10)求得比较矩阵的不一致程度值CI,通过查验资料判定该矩阵是否具有满意的一致性。若矩阵获得满意的一致性,则$\boldsymbol{\omega_D}=(\alpha,\beta,\gamma,\theta)$;否则重新调整比较矩阵$\boldsymbol{D}$,直到获得满意的一致性。

$$CI=\frac{\lambda_{max}(\boldsymbol{D})-n}{n-1} \tag{10}$$

(5)可靠性指标求解。最后根据归一化的各项属性测评分数与求得的权值,得到云服务可靠性指标,如式(11)所示。其中,$\bar{S}_1$,$\bar{S}_2$,$\bar{S}_4$分别代表一个服务周期内,对所有虚拟机服务能力、虚拟资源隔离性、虚拟机性能损失求得的均值。例如,$\bar{S}_1$代表一个服务周期内所有虚拟机的CPU性能得分均值、内存性能得分均值、网络性能得分均值、磁盘性能得分均值的加权得分;$\bar{S}_3$代表一个服务周期内平台的资源管理能力得分均值;$\bar{S}_4$代表一个服务周期内所有虚拟机的性能折损得分,由虚拟机的计算性能、网络性能以及磁盘性能减去同配置物理机的对应性能得分而获得的折损均值。R则代表最终的云服务可靠性指标值。

$$\begin{cases} R=\alpha\bar{S}_1+\beta\bar{S}_2+\gamma\bar{S}_3+\theta\bar{S}_4 \\ \bar{S}_1=\alpha(\bar{S}_{cpu}+\bar{S}_m+\bar{S}_n+\bar{S}_d) \\ \bar{S}_2=\beta(\bar{S}_{cpu}+\bar{S}_n+\bar{S}_d) \\ \bar{S}_3=\gamma(\bar{T}_1+\bar{T}_n+\bar{T}_d) \\ \bar{S}_4=\theta(\bar{S}_{cpu}+\bar{S}_n+\bar{S}_d) \end{cases} \tag{11}$$

本文采用两级层次分析法来确定影响服务可靠性的评测项目的权值。通过综合考虑虚拟机服务能力、虚拟资源隔离性、资源管理能力、虚拟机性能折损4个方面,对智能服务框架的可靠性进行重新定义。

3.2.3 服务效能的服务效率指标定义

云服务可用性描述了云服务在时间维度的可用范围,而云服务可靠性提供了服务层面的资源保障,让用户能够安心使用云服务。云服务效率则直接决定了服务的时效性,是用户体验最直接的重要指标。

本文在定义云服务效率时,将虚拟机上运行不同类型的基准测试应用程序的平均运行时间作为服务效率的评价。基准测试程序主要包括内存密集型应用和I/O密集型应用。云服务效率的定义如式(12)所示。

$$T=\frac{1}{m\times n}\sum_{j=1}^{n}\sum_{i=1}^{m}t_{ij} \tag{12}$$

其中,n代表一个服务周期内的虚拟机数量,m代表测试应用的数量,t_{ij}代表第j台虚拟机上运行第i个测试应用程序的运行时间。即云服务效率本质上是应用的运行时间。

3.3 基于BP神经网络的效能模型构建

神经网络领域证明了BP神经网络可以任意逼近任何一个非线性函数,而且证明了隐含层数越多,通过传递的误差越大,泛化能力也就越低。因此,本文采用3层BP神经网络模型,基于智能服务框架的总体架构,从框架各层提取影响整体服务效能的关键输入特征,利用BP神经网络模型,对整体

服务效能进行预测。本文构建的服务效能预测模型的网络结构如图 7 所示，智能云各要素的配置作为输入层节点，服务效能输出指标作为输出层节点。

表 2 列出了效能模型构建过程中相关符号的形式化定义，本文将基于表 2 中定义的符号给出效能模型详细的推导过程。

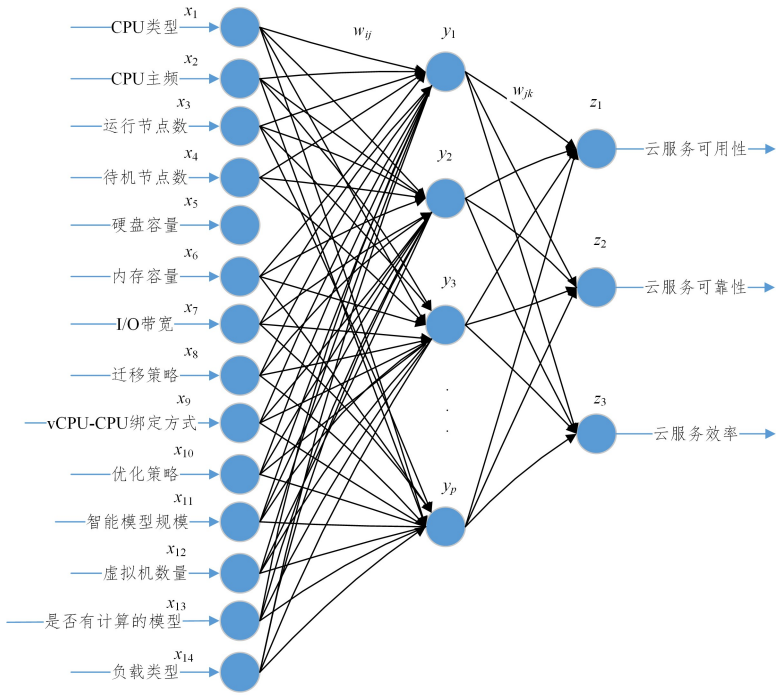


图 7　整体服务效能的 BP 神经网络预测模型示意图

Fig. 7　Schematic diagram of BP neural network prediction model for service efficiency

表 2　效能模型的相关符号定义

Table 2　Symbolic definition of effectiveness model

符号及定义	描述
$d=[d_1,d_2,d_3]\in \mathrm{R}^2$	期望服务效能输出
$p\in Z$	隐含层节点数量
$K\in Z$	分组数量
$x_i,1\leqslant x\leqslant 14$	输入层节点
$y_j,1\leqslant j\leqslant p$	隐含层节点
$z_k,k=1,2,3$	输出层节点
$w_{i,j}\in \mathrm{R}$	输入层节点 i 到隐含层节点 j 的权值
$w_{j,k}\in \mathrm{R}$	隐含层节点 j 到输出层节点 k 的权值
$f(x)=1/(1+\mathrm{e}^{-x})\in \mathrm{R}^d$	隐含层和输出层的传递函数
$e(\cdot)\in \mathrm{R}^d$	目标效能输出与实际效能输出的误差函数
$b_y\in \mathrm{R}$	隐含层神经元阈值
$b_o\in \mathrm{R}$	输出层神经元阈值
$L\in Z$	样本数据个数
$\boldsymbol{G}_k(k=1,2,\cdots,K)\in \mathrm{R}$	第 K 组节点
$\boldsymbol{d}(k)=(\boldsymbol{d}_1(k),\boldsymbol{d}_2(k),\boldsymbol{d}_3(k))\in \mathrm{R}^{1\times 3}$	对应期望输出
$\boldsymbol{w}(k)_j=(\boldsymbol{w}_{1,j}(k),\boldsymbol{w}_{2,j}(k),\cdots,\boldsymbol{w}_{14,j}(k))\in \mathrm{R}^{1\times 14}$	隐含层第 j 个节点对应的权重向量
$\boldsymbol{B}_y(k)=[\boldsymbol{b}_y(k)]\in \mathrm{R}^{1\times p}$	隐含层对应阈值向量
$\boldsymbol{W}_y(k)=(\boldsymbol{w}(k)_1^{\mathrm{T}},\boldsymbol{w}(k)_2^{\mathrm{T}},\cdots,\boldsymbol{w}(k)_p^{\mathrm{T}})^{\mathrm{T}}\in \mathrm{R}^{p\times 14}$	隐含层对应权重矩阵
$\boldsymbol{y}^{\mathrm{in}}(k)\in \mathrm{R}^{p\times 1}$	隐含层输入向量
$\boldsymbol{y}^{\mathrm{out}}(k)\in \mathrm{R}^{p\times 1}$	隐含层输出向量
$\boldsymbol{w}(k)_m=(\boldsymbol{w}_{1,m}(k),\boldsymbol{w}_{2,m}(k),\cdots,\boldsymbol{w}_{p,m}(k))\in \mathrm{R}^{1\times p}$	输出层第 m 个节点对应权重向量
$\boldsymbol{B}_o(k)=[\boldsymbol{b}_o(k)]\in \mathrm{R}^{1\times 3}$	输出层对应阈值向量
$\boldsymbol{W}_o(k)=(\boldsymbol{w}(k)_1^{\mathrm{T}},\boldsymbol{w}(k)_2^{T},\boldsymbol{w}(k)_3^{\mathrm{T}})^{\mathrm{T}}\in \mathrm{R}^{3\times p}$	输出层对应权重矩阵
$\boldsymbol{z}^{\mathrm{in}}(k)\in \mathrm{R}^{3\times 1}$	输出层的输入向量
$\boldsymbol{z}^{\mathrm{out}}(k)\mathrm{R}^{3\times 1}$	输出层的输出向量
$\boldsymbol{X}=(\boldsymbol{X}(1)^{\mathrm{T}},\boldsymbol{X}(2)^{\mathrm{T}},\cdots,\boldsymbol{X}(K)^{\mathrm{T}})^{\mathrm{T}}\in \mathrm{R}^{K\times 17}$	所有输入向量构成的输入数据矩阵
$\boldsymbol{W}_y=(\boldsymbol{W}_y(1)^{\mathrm{T}},\boldsymbol{W}_y(2)^{\mathrm{T}},\cdots,\boldsymbol{W}_y(K)^{\mathrm{T}})$	隐含层对应权重矩阵
$\boldsymbol{B}_y=(\boldsymbol{B}_y(1)^{\mathrm{T}},\boldsymbol{B}_y(2)^{\mathrm{T}},\cdots,\boldsymbol{B}_y(K)^{\mathrm{T}})\in \mathrm{R}^{p\times K}$	隐含层对应阈值矩阵
$\boldsymbol{W}_o=(\boldsymbol{W}_o(1)^{\mathrm{T}},\boldsymbol{W}_o(2)^{\mathrm{T}},\cdots,\boldsymbol{W}_o(K)^{\mathrm{T}})^{\mathrm{T}}$	输出层对应权重矩阵
$\boldsymbol{B}_o=(\boldsymbol{B}_o(1),\boldsymbol{B}_o(2),\cdots,\boldsymbol{B}_o(K))\in \mathrm{R}^{3\times K}$	输出层对应阈值矩阵

(续表)

符号及定义	描述
$\boldsymbol{Y}^{\text{in}}=(\boldsymbol{y}^{\text{in}}(1),\boldsymbol{y}^{\text{in}}(2),\cdots,\boldsymbol{y}^{\text{in}}(K))\in\mathrm{R}^{p\times K}$	隐含层所有节点的输入矩阵
$\boldsymbol{Y}^{\text{out}}=(\boldsymbol{y}^{\text{out}}(1),\boldsymbol{y}^{\text{out}}(2),\cdots,\boldsymbol{y}^{\text{out}}(K))\in\mathrm{R}^{p\times K}$	隐含层所有节点的输出矩阵
$\boldsymbol{Z}^{\text{in}}=(\boldsymbol{z}^{\text{in}}(1),\boldsymbol{z}^{\text{in}}(2),\cdots,\boldsymbol{z}^{\text{in}}(K))\in\mathrm{R}^{3\times K}$	输出层所有节点的输入矩阵
$\boldsymbol{Z}^{\text{out}}=(\boldsymbol{z}^{\text{out}}(1),\boldsymbol{z}^{\text{out}}(2),\cdots,\boldsymbol{z}^{\text{out}}(K))\in\mathrm{R}^{3\times K}$	输出层所有节点的输出矩阵
$N(k)\in\mathrm{R}$	第 k 组包含的节点数量
$\delta_i(k)\in\mathrm{R}$	第 k 组中节点 i 的梯度值
$\boldsymbol{W}_o^t=(\boldsymbol{w}_{jk}^t(1),\boldsymbol{w}_{jk}^t(2),\cdots,\boldsymbol{w}_{jk}^t(K))^{\mathrm{T}}$	输出层向量
$\boldsymbol{G}_o'=(\boldsymbol{g}_o'(1),\boldsymbol{g}_o'(2),\cdots,\boldsymbol{g}_o'(k))^{\mathrm{T}}$	输出层所有 K 组节点构成合并梯度的修正向量
$w_{jk}^t(k)\in\mathrm{R}$	第 k 组节点在第 t 次迭代时输出层模型参数
$l_k^t\in\mathrm{R}$	第 k 组第 t 次迭代时的学习速率
$\delta_i^y(k)\in\mathrm{R}$	隐含层第 k 组节点 i 的梯度值
$\boldsymbol{W}_h^t=(\boldsymbol{w}_{ij}^t(1),\boldsymbol{w}_{ij}^t(2),\cdots,\boldsymbol{w}_{ij}^t(K))^{\mathrm{T}}$	第 t 次迭代时,隐含层参数向量
$\boldsymbol{G}_h'=(\boldsymbol{g}_h'(1),\boldsymbol{g}_h'(2),\cdots,\boldsymbol{g}_h'(k))^{\mathrm{T}}$	隐含层所有 K 组节点构成合并梯度修正向量
$w_{ij}^t(k)\in\mathrm{R}$	第 k 组节点在第 t 次迭代时隐含层模型参数

本文将基于分布式 BP 神经网络优化模型建立整体服务效能预测模型,其构建过程如下。

3.3.1 构造输入数据

预测模型的数据集包含训练和测试数据,按照 7:3 的比例分配。

3.3.2 模型参数初始化

决定模型收敛效率和准确度的重要参数有隐含层神经元个数、权重参数。本文对这两重要参数进行初始化时采用的方法如下。

(1)隐含层神经元个数

目前深度学习领域没有对神经网络模型隐含层的神经元个数给出一个统一的公式,这也是 BP 神经网络模型的缺点之一。隐含层神经元个数太多,会导致训练时间过长,甚至造成过拟合现象;个数太少,会造成模型精度太低。因此,隐含层神经元个数决定了模型的性能和效率。常用的方法是根据经验值确定隐含层神经元个数。本文根据式(13)确定隐含层神经元个数,其中规定:如果 $i>p$,则 $C_p^i=0$。

$$\sum_{i=0}^{14}C_p^i>L \tag{13}$$

(2)权重参数初始化

BP 神经网络权重初始值的选择决定了算法搜索的局部极小值和模型的收敛范围。本文对两级网络的初始权值采用了不同的选择方式:输入层到隐含层的权重初始赋值为(−0.1,0.1)之间很小的随机数,隐含层到输出层的权重初始值赋值为(−1,1)之间的随机数。

3.3.3 模型迭代更新

模型正向计算即对隐含层和输出层的输入和输出进行前向计算。

隐含层输入向量 $\boldsymbol{y}^{\text{in}}(k)\in\mathrm{R}^{p\times 1}$ 和输出向量 $\boldsymbol{y}^{\text{out}}(k)\in\mathrm{R}^{p\times 1}$ 的计算式如式(14)和式(15)所示。

$$\begin{aligned}\boldsymbol{y}^{\text{in}}(k)&=(X(k)\times\boldsymbol{W}_y(k)^{\mathrm{T}}-\boldsymbol{B}_y(k))^{\mathrm{T}}\\&=\sum_{i=1}^{14}w_{ij}x_i(k)-b_y,j=1,2,\cdots,p\end{aligned} \tag{14}$$

$$\boldsymbol{y}^{\text{out}}(k)=f(\boldsymbol{y}^{\text{in}}(k)) \tag{15}$$

输出层输入向量 $\boldsymbol{z}^{\text{in}}(k)\in\mathrm{R}^{3\times 1}$ 和输出向量 $\boldsymbol{z}^{\text{out}}(k)\in\mathrm{R}^{3\times 1}$ 的计算式如式(16)和式(17)所示。

$$\begin{aligned}\boldsymbol{z}^{\text{in}}(k)&=(\boldsymbol{W}_o(k)\times\boldsymbol{y}^{\text{out}}(k)-\boldsymbol{B}_o(k))^{\mathrm{T}}\\&=\sum_{j=1}^{p}w_{jk}\boldsymbol{y}^{\text{out}}(k)-b_o,k=1,2,3\end{aligned} \tag{16}$$

$$\boldsymbol{z}^{\text{out}}(k)=f(\boldsymbol{z}^{\text{in}}(k)) \tag{17}$$

所有隐含层节点的输入矩阵 $\boldsymbol{Y}^{\text{in}}$ 和输出矩阵 $\boldsymbol{Y}^{\text{out}}$ 分别如式(18)和式(19)所示。

$$\begin{aligned}\boldsymbol{Y}^{\text{in}}&=(\boldsymbol{X}\times\boldsymbol{W}_y)^{\mathrm{T}}-\boldsymbol{B}_y\\&=\left[\begin{pmatrix}\boldsymbol{X}(1)\\\boldsymbol{X}(2)\\\cdots\\\boldsymbol{X}(K)\end{pmatrix}(\boldsymbol{W}_y(1)^{\mathrm{T}},\boldsymbol{W}_y(2)^{\mathrm{T}},\cdots,\boldsymbol{W}_y(K)^{\mathrm{T}})\right]^{\mathrm{T}}-\boldsymbol{B}_y\end{aligned} \tag{18}$$

$$\boldsymbol{Y}^{\text{out}}=f(\boldsymbol{Y}^{\text{in}}) \tag{19}$$

所有输出层节点的输入矩阵 $\boldsymbol{Z}^{\text{in}}$ 和输出矩阵 $\boldsymbol{Z}^{\text{out}}$ 的计算式分别如式(20)和式(21)所示。

$$\begin{aligned}\boldsymbol{Z}^{\text{in}}&=(\boldsymbol{W}_o\times\boldsymbol{Y}^{\text{out}}-\boldsymbol{B}_o{}^{\mathrm{T}})^{\mathrm{T}}\\&=\left[\begin{pmatrix}W_o(1)\\W_o(2)\\\cdots\\W_o(K)\end{pmatrix}(\boldsymbol{y}^{\text{out}}(1),\boldsymbol{y}^{\text{out}}(2),\cdots,\boldsymbol{y}^{\text{out}}(K))-\boldsymbol{B}_o\right]^{\mathrm{T}}\end{aligned} \tag{20}$$

$$\boldsymbol{Z}^{\text{out}}=f(\boldsymbol{Z}^{\text{in}}) \tag{21}$$

计算输出层梯度值的过程如下:

$$e=\frac{1}{2}\sum_{i=1}^{3}(\boldsymbol{d}_i(k)-\boldsymbol{z}_i^{\text{out}}(k))^2$$

$$\frac{\partial e}{\partial w_{jk}}=\frac{\partial e}{\partial\boldsymbol{z}^{\text{in}}(k)}\frac{\partial\boldsymbol{z}^{\text{in}}(k)}{\partial w_{jk}}$$

$$\frac{\partial\boldsymbol{z}^{\text{in}}(k)}{\partial w_{jk}}=\frac{\partial(\sum_{j=1}^{p}w_{jk}\boldsymbol{y}^{\text{out}}(k)-b_o)}{\partial w_{jk}}=\boldsymbol{y}^{\text{out}}(k)$$

$$\begin{aligned}\frac{\partial e}{\partial\boldsymbol{z}^{\text{in}}(k)}&=\frac{\partial\left[\frac{1}{2}\sum_{i=1}^{3}(d_i(k)-\boldsymbol{z}_i^{\text{out}}(k))^2\right]}{\partial\boldsymbol{z}^{\text{in}}(k)}\\&=-(d_i(k)-\boldsymbol{z}_i^{\text{out}}(k))\boldsymbol{z}_i^{\text{out}'}(k)\\&=-(d_i(k)-\boldsymbol{z}_i^{\text{out}}(k))f'(\boldsymbol{z}^{\text{in}}(k))=-\delta^z(k)\end{aligned}$$

为了提升模型的计算效率,输出层第一级参数服务器分小组并行合并梯度向量 $\boldsymbol{g}_o(k)$,如式(22)所示。

$$\begin{aligned}\boldsymbol{G}_o&=(\boldsymbol{g}_o(1),\boldsymbol{g}_o(2),\cdots,\boldsymbol{g}_o(K))\\&=(\sum_{i=1}^{N(1)}\delta_i^z(1),\sum_{i=1}^{N(2)}\delta_i^z(2),\cdots,\sum_{i=1}^{N(K)}\delta_i^z(K))\end{aligned} \tag{22}$$

设二级参数服务器只要收到了超过一半分组的合并梯度值,就进行模型参数更新,不需要等待剩余分组的

合并梯度值。假设第 t 次迭代时，式(23)描述了如何利用第二级参数服务器进行输出层第 k 组优化梯度值的参数更新。

$$w_{jk}^{t+1}(k)=w_{jk}^{t}(k)-\sum_{j=1}^{m}\left(\frac{l_k^t}{B_i(k)\times N(k)}\sum_{i=1}^{B_i(k)\times N(k)}\boldsymbol{g}_o(k)\right),\quad m\geqslant\frac{M}{2}\tag{23}$$

假设 $\boldsymbol{g}_o'(k)$ 由式(24)得出。

$$\boldsymbol{g}_o'(k)=\sum_{j=1}^{m}\left(\frac{l_k^t}{B_i(k)\times N(k)}\sum_{i=1}^{B_i(k)\times N(k)}\boldsymbol{g}_o(k)\right),m\geqslant\frac{M}{2}\tag{24}$$

所有节点构成合并梯度的修正向量 $\boldsymbol{G}_o'$，则 K 组节点的输出层参数更新如式(25)所示。

$$\boldsymbol{W}_o^{t+1}=\boldsymbol{W}_o^{t}-\boldsymbol{G}_o'\tag{25}$$

其中，$B_i(k)\times N(k)$ 是第 k 组数据块的大小，第二级参数服务器将更新后的模型参数 $w_{jk}^{t+1}(k)$ 发送回对应的第 K 分组。然后，K 组工作节点开始下一轮的迭代。

并行计算隐含层的梯度值及权重更新，求解隐含层的梯度值的过程如下：

$$\frac{\partial e}{\partial w_{jk}}=-\delta_z(m)\boldsymbol{y}^{\text{out}}(m)$$

$$\frac{\partial e}{\partial w_{ij}}=\frac{\partial e}{\partial \boldsymbol{y}^{\text{in}}(m)}\frac{\partial \boldsymbol{y}^{\text{in}}(m)}{\partial w_{ij}}$$

$$\frac{\partial \boldsymbol{y}^{\text{in}}(m)}{\partial w_{ij}}=\frac{\partial\left[\sum_{i=1}^{17}w_{ij}x_i(m)-b_y\right]}{\partial w_{ij}}=x_i(m)$$

$$\frac{\partial e}{\partial \boldsymbol{y}^{\text{in}}(m)}=\frac{\partial\left[\frac{1}{2}\sum_{i=1}^{3}(d_i(m)-\boldsymbol{z}_i^{\text{out}}(m))^2\right]}{\partial \boldsymbol{y}^{\text{out}}(m)}\frac{\partial \boldsymbol{y}^{\text{out}}(m)}{\boldsymbol{y}^{\text{in}}(m)}$$

$$=\frac{\partial\left[\frac{1}{2}\sum_{i=1}^{3}(d_i(m)-f(\sum_{j=1}^{p}w_{jk}\boldsymbol{y}^{\text{out}}(m)-b_o))^2\right]}{\partial \boldsymbol{y}^{\text{out}}(m)}$$

$$\frac{\partial \boldsymbol{y}^{\text{out}}(m)}{\partial \boldsymbol{y}^{\text{in}}(m)}$$

$$=-\sum_{i=1}^{3}(d_i(m)-\boldsymbol{z}_i^{\text{out}}(m))f'(\boldsymbol{z}_i^{\text{in}}(m))w_{jk}$$

$$\frac{\partial \boldsymbol{y}^{\text{out}}(m)}{\partial \boldsymbol{y}^{\text{in}}(m)}$$

$$=-(\sum_{i=1}^{3}\delta_z(m)w_{jk})f'(\boldsymbol{y}^{\text{in}}(m))=-\delta_y(m)$$

为了提升计算效率，本文采用了两阶段梯度下降优化算法，即第一级参数服务器分小组并行合并隐含层梯度向量 $\boldsymbol{g}_h(k)$，如式(26)所示。

$$\begin{aligned}\boldsymbol{G}_h&=(\boldsymbol{g}_h(1),\boldsymbol{g}_h(2),\cdots,\boldsymbol{g}_h(K))\\&=(\sum_{i=1}^{N(1)}\delta_i^y(1),\sum_{i=1}^{N(2)}\delta_i^y(2),\cdots,\sum_{i=1}^{N(K)}\delta_i^y(K))\end{aligned}\tag{26}$$

设二级参数服务器只要收到了超过一半分组的合并梯度值，就进行模型参数更新，不需要等待剩余分组的合并梯度值。式(27)描述了如何利用第二级参数服务器进行隐含层第 k 组优化梯度值的参数更新，其中 $m\geqslant M/2$。

$$w_{ij}^{t+1}(k)=w_{ij}^{t}(k)-\sum_{j=1}^{m}\left(\frac{l_k^t}{B_i(k)\times N(k)}\sum_{i=1}^{B_i(k)\times N(k)}\boldsymbol{g}_h(k)\right)\tag{27}$$

合并梯度修正值 $\boldsymbol{g}_h'(k)$ 由式(28)得出，其中 $m\geqslant M/2$。

$$\boldsymbol{g}_h'(k)=\sum_{j=1}^{m}\left(\frac{l_k^t}{B_i(k)\times N(k)}\sum_{i=1}^{B_i(k)\times N(k)}\boldsymbol{g}_h(k)\right)\tag{28}$$

则 K 组节点的隐含层参数更新如式(29)所示。

$$\boldsymbol{W}_h^{t+1}=\boldsymbol{W}_h^{t}-\boldsymbol{G}_h'\tag{29}$$

其中，$B_i(k)\times N(k)$ 是第 k 组数据块的大小，第二级参数服务器将更新后的隐含层模型参数 $w_{ij}^{t+1}(k)$ 发送回对应的第 k 分组。然后，K 组工作节点开始下一轮迭代。

3.3.4　整体效能模型构建

利用分布式 BP 神经网络优化模型，构建面向智能服务框架的整体效能预测模型的主要流程如图 8 所示。首先利用“输入特征向量-实际服务效能三元组”构造数据集样本，并按照一定比例进行训练集和测试集的划分；然后基于训练集，利用分布式 BP 神经网络优化模型构建效能预测模型；最后通过测试集计算来评价模型的准确率和效率，最终确定整体效能预测模型。

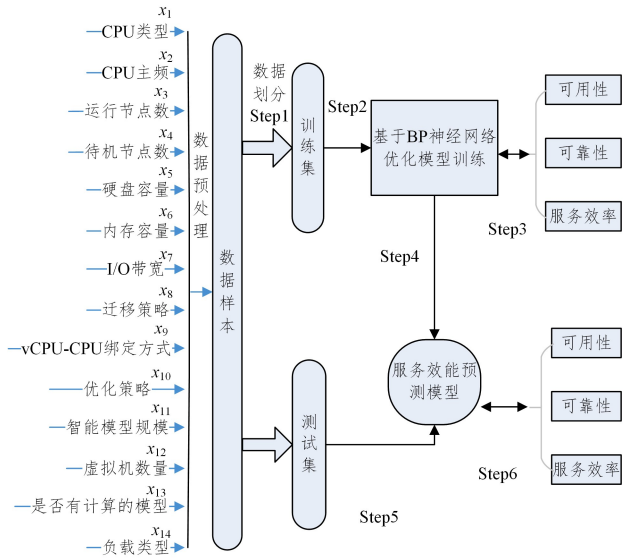


图 8　整体效能模型构建示意图

Fig. 8　Schematic diagram of the overall effectiveness model construction

整体效能模型的构建流程如下：

步骤 1(输入)：按照 3.3.1 节构造样本数据。

步骤 2 和步骤 3(迭代计算构建预测模型)：按照 3.3.2 节初始化隐藏层个数和权重参数，然后进行迭代计算以构建整体效能模型；分组计算模型各层神经元的输入和输出，合并参数并更新；分组更新模型权重参数以及对应的学习率参数；重复步骤 1 和步骤 2，直到模型收敛，构建整体效能的预测模型。

步骤 4 和步骤 5：利用测试集，通过预测输出服务效能与实际输出服务效能来评价模型的准确率和效率，如果满足要求则结束；否则重新进行效能模型的构建。

4　实验与结果

4.1　实验环境

本文搭建了效能模型专门的分布式训练平台，其配置如表 3 所列，为了验证提出的面向智能服务框架的效能模型的有效性，硬件方面采用智能应用支撑服务器，软件方面依托

深度学习框架 MXNet。该智能应用支撑服务器具体的配置如表 4 所列。

表 3 效能模型训练平台的实验环境汇总

Table 3 Experimental environment summary of effectiveness model training platform

节点	CPU	GPU	内存/GB
1	2(Intel Xeon)	4 * Tesla	16
2	2(Intel Xeon)	2 * Tesla	32
3	1(Intel Core i7)	1(GeForce GTX Titan Z)	12
4	1(Intel Core i7)	1(GeForce GTX Titan Z)	12

表 4 智能应用支撑服务器配置

Table 4 Configuration of intelligent application support server

配置项	规格
CPUs	1 * (Xeon E5)
System Memory	256 GB RAM
Operating System	Ubuntu16.0.4
GPUs	4 * Tesla
# CUDA Cores	3584
GPU Core Frequency/MHz	1480
GPU Memory/GB	16
CUDA	7.5

4.2 数据样本集构造

本节构建了效能模型的数据样本集，如表 5 所列。为了建模，表 6 中的特征表示做了简化处理，并给出了特征的取值或取值范围。

表 5 效能模型的样本集结构

Table 5 Sample set structure of efficiency model

输入特征向量	整体服务效能三元组
$(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{16})$	(Z_1,Z_2,Z_3)

为了便于理解，下面对表 6 中几个重要的特征进行说明。

开机状态的服务器数量。开机状态服务器数量指除了智能应用支撑服务器之外的服务器，目前最多有 4 台服务器。因此，开机状态的服务器数量在 1～4 之间。

待机状态的服务器数量。云端可根据实际运行的虚拟机数量灵活调整服务器的开机数量，尽可能地节约资源和能耗。

vCPU-CPU 绑定方式。其指 vCPU 是否绑定在固定的物理服务器。对于一些非数值类型的配置方式，输入特征通过指定数值的方式进行设定。例如，1 为不绑定，即 vCPU 线程在物理核心上动态切换；2 为绑定，即 vCPU 线程按序依次绑定在物理核心上。

优化算法。优化算法是系统总体架构设计的定制优化算法，即为了保证智能云的高性能智能服务，不同的智能应用类型可定制优化策略。输入特征通过指定数值的方式对优化算法进行设定。

智能模型。由于不同规模的智能模型训练时，对资源的使用效率有直接影响。因此，输入特征将根据模型的规模对模型进行分类，具体按照模型的层数对模型规模进行确定。

表 6 效能模型的输入特征

Table 6 Input characteristics of effectiveness model

特征分类	特征	特征取值
计算资源特征	CPU 类型	Intel(1), AMD(2)
	CPU 主频/GHz	1.6～3.3
	开机状态服务器数量	1～4
	待机状态服务器数量	0～2
存储资源特征	硬盘容量	256G～1T
	内存容量/GB	16～128
网络资源特征	I/O 带宽	100M(1), 1000M(2), 10000M(3)
虚拟化配置特征	虚拟机迁移策略	离线迁移(1), 在线迁移(2)
	vCPU-CPU 绑定方式	不绑定(1), 绑定(2)
智能化配置特征	优化算法	无任何优化(1), BP 神经网络优化模型(2), 内存优化(3), 软硬件协同模式优化(4), 数据放置策略优化(5), 安全优化(6), 通信优化(7), 计算图优化(8)
	智能模型规模	小规模神经网络模型(1), 中等规模神经网络模型(2), 大规模神经网络模型(3), 超大规模神经网络模型(4)
运行时环境特征	平台上同时运行的虚拟机数量	$[0,n]$
	是否有正在计算的模型	无模型(0), 有模型(1)
	虚拟机上运行的负载类型	无负载(1), I/O 密集型应用(2), 计算密集型应用(3), 典型智能应用(4)

虚拟机上运行的负载类型。各类应用包括典型智能应用和基准测试应用程序两大类，其中的典型智能应用主要包括人脸识别和目标检测；基准测试程序主要包括内存密集型应用和 I/O 密集型应用。利用 Parsec 中的基准测试程序模拟不同的负载类型。

虚拟机服务的系统级别。其为虚拟机服务的应用系统所属级别，如优先级最高的系统属于一级系统。

4.3 数据样本集构造

数据样本集构造。根据表 6 中影响服务效能的输入特征，在每个输入特征变量的值域范围内随机取值，可获得数据集的一个输入特征配置。根据表 6 多次取值可得到一组配置。

本文获取了 600 个不同的效能模型的输入特征配置，并根据建立的“整体服务效能”评估指标，获得对应的服务效能三元组的实测值。最后获得服务效能的数据样本集。

4.4 实验结果与分析

本文采用 BP 神经网络优化模型，实现对云服务效能的预测。实验基于数据样本集，主要根据模型预测值和实际值的对比来实现对服务效能模型的评测。实验中，本节将提出的效能模型(表示为 DP_BP)与常用的线性回归预测模型(表示为 LR)、未采用任何优化算法的 BP 神经网络预测模型(表示为 BP)进行了对比分析。

线性回归模型通过最小二乘法找到多个输入特征参数的最佳组合，以实现对服务效能的预测，如式(30)所示。

$$y=\alpha_0+\alpha_1 x_1+\alpha_2 x_2+\cdots+\alpha_k x_k+\beta \tag{30}$$

其中，α_0 是常数，$\alpha_1,\alpha_2,\cdots,\alpha_k$ 是回归系数，β 是误差项。

本节按照以下步骤完成对比实验分析。

(1)服务效能模型准确率比较

实验中，3 种模型都使用相同的训练样本集。服务效能评测基于效能模型的数据样本集，并按照 7∶3 的比例进行训练和测试。应用于服务效能预测的 3 种模型的预测准确率对比图如图 9 所示。

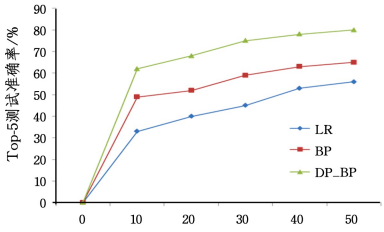


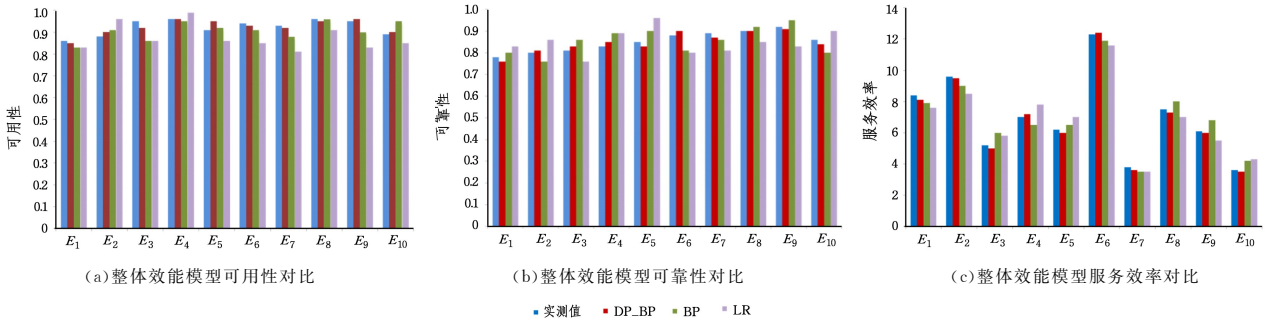
图 9　服务效能的 Top-5 准确率分析

Fig. 9　Top-5 accuracy analysis of service efficiency

(2)服务效能模型预测准确性对比

随机配置 10 组不同的智能云整体输入特征(表示为E_i，$i\in[1,10]$)，通过“整体服务效能”评估指标的定义获得可用性、可靠性、服务效率的实测值，用于表示整体效能模型实际的服务质量水平，然后将实测值与预测值进行比较。对比分析中，不仅将实测值与本文提出模型的预测值进行比较，还将与未采用优化算法的 BP 神经网络模型的服务效能预测值进行比较，同时还将与基于线性回归模型的预测值进行比较。比较结果如图 10 所示。从图 10 可以看出，基于 BP 神经网络优化模型，相比线性回归模型和未采用任何优化策略的 BP 神经网络模型，其服务效能的预测值更接近实测值。

为了准确计算以上 3 种模型预测值与实测值之间的误差，本节采取平均绝对误差、均方误差和均方根误差来评估误差的大小。



(a)整体效能模型可用性对比　(b)整体效能模型可靠性对比　(c)整体效能模型服务效率对比

图 10　整体效能模型预测准确性对比

Fig. 10　Comparison of prediction accuracy of overall effectiveness model

整体服务效能的平均绝对误差(MAE)、均方误差(SE)和均方根误差(RSE)分别如式(31)－式(33)所示。

$$\begin{cases} A_{MAE}=\dfrac{1}{n}\sum\limits_{i=1}^{n}|A_{\text{test}}(i)-A_{\text{predict}}(i)| \\ R_{MAE}=\dfrac{1}{n}\sum\limits_{i=1}^{n}|R_{\text{test}}(i)-R_{\text{predict}}(i)| \\ T_{MAE}=\dfrac{1}{n}\sum\limits_{i=1}^{n}|T_{\text{test}}(i)-T_{\text{predict}}(i)| \end{cases} \tag{31}$$

$$\begin{cases} A_{SE}=\dfrac{1}{n}\sum\limits_{i=1}^{n}\left|\dfrac{A_{\text{test}}(i)-A_{\text{predict}}(i)}{A_{\text{test}}(i)}\right| \\ R_{SE}=\dfrac{1}{n}\sum\limits_{i=1}^{n}\left|\dfrac{R_{\text{test}}(i)-R_{\text{predict}}(i)}{R_{\text{test}}(i)}\right| \\ T_{SE}=\dfrac{1}{n}\sum\limits_{i=1}^{n}\left|\dfrac{T_{\text{test}}(i)-T_{\text{predict}}(i)}{T_{\text{test}}(i)}\right| \end{cases} \tag{32}$$

$$\begin{cases} A_{RSE}=\sqrt{\dfrac{1}{n}\sum\limits_{i=1}^{n}(A_{\text{test}}(i)-A_{\text{predict}}(i))^2} \\ R_{RSE}=\sqrt{\dfrac{1}{n}\sum\limits_{i=1}^{n}(R_{\text{test}}(i)-R_{\text{predict}}(i))^2} \\ T_{RSE}=\sqrt{\dfrac{1}{n}\sum\limits_{i=1}^{n}(T_{\text{test}}(i)-T_{\text{predict}}(i))^2} \end{cases} \tag{33}$$

其中，$A_{\text{test}}(i)$，$R_{\text{test}}(i)$，$T_{\text{test}}(i)$分别为可用性、可靠性和服务效率的实测值，$A_{\text{predict}}(i)$，$R_{\text{predict}}(i)$，$T_{\text{predict}}(i)$分别为舰载公共计算环境可用性、可靠性和服务效率的预测值。如果 3 个评价指标的误差越小，则该模型预测的服务效能越接近真实值，说明模型的预测效果好。表 7 列出了 3 种模型在整体服务效能方面的 3 个误差评价指标的计算结果。

表 7　服务效能的 3 个误差评价指标结果

Table 7　Three error evaluation results of service efficiency

误差模型	平均绝对误差(*MAE*)			均方误差(*SE*)			均方根误差(*RSE*)		
	可用性 *A*	可靠性 *R*	服务效率 *T*	可用性 *A*	可靠性 *R*	服务效率 *T*	可用性 *A*	可靠性 *R*	服务效率 *T*
DP_BP	0.22	0.37	0.36	0.01	0.02	0.04	0.26	0.36	0.56
BP	0.36	0.51	0.89	0.03	0.05	0.06	0.35	0.51	0.98
LR	0.56	0.85	1.56	0.06	0.08	1.12	0.65	0.89	1.23

从表 7 中可以看出，对于本文基于 BP 神经网络优化模型的效能模型，所预测的两类服务效能结果的 3 个误差评价指标最小，证明了本文提出的效能模型相比其他两种方法，在服务效能的预测方面具有较小的误差和应用普适性。

结束语　本文基于智能服务框架的系统总体架构，设计了一种预测云服务质量的整体效能模型。现存的云服务质量建模主要是针对某些特定领域或服务类型的建模方法，且整个行业仍然缺乏统一的标准和规范。本文分析了影响整体服务效能的输入特征，并对整体服务效能评价指标进行了定义，最后基于 BP 神经网络优化模型，模拟输入与输出服务效能之间复杂的非线性关系，建立了效能模型。实验结果表明，本文提出的基于 BP 神经网络优化模型作为预测模型具有好的计算效率和准确率。并且，本文提出的效能模型在服务效能的预测方面，3 个误差指标都较小。因此，该方法具有良好的普适性。

参 考 文 献

[1]　SCHUNSELAAR D M M, VERBEEK H M W, REIJERS H A,

et al. YAWL in the Cloud: Supporting Process Sharing and Variability[J]. Lecture Notes in Business Information Processing, 2015, 202(3): 367-379.

[2] BRUNEO D, LONGO F, GHOSH R, et al. Analytical Modeling of Reactive Autonomic Management Techniques in IaaS Clouds [C]// IEEE International Conference on Cloud Computing. IEEE, 2015: 797-804.

[3] BRUNEO D, LONGO F, SCARPA M, et al. An SRN-Based Resiliency Quantification Approach[M]// Application and Theory of Petri Nets and Concurrency. Springer International Publishing, 2015: 98-116.

[4] ZHOU A, WANG S, CHENG B, et al. Cloud Service Reliability Enhancement via Virtual Machine Placement Optimization[J]. IEEE Transactions on Services Computing, 2017(6): 902-913.

[5] BAI Y, ZHANG H, FU Y. Reliability modeling and analysis of cloud service based on complex network[C]// 2016 Prognostics and System Health Management Conference (PHM-Chengdu). IEEE, 2016.

[6] ALANNSARY M O, TIAN J. Measurement and Prediction of SaaS Reliability in the Cloud[C]// IEEE International Conference on Software Quality. IEEE, 2016.

[7] QI L, NI J, XIA X, et al. A Multi-dimensional Weighting Method for Historical Records in Cloud Service Evaluation[C]// IEEE Fourth International Conference on Big Data & Cloud Computing. IEEE Computer Society, 2014.

[8] HOSSEINI S B, SHOJAEE A, AGHELI N. A new method for evaluating cloud computing user behavior trust[C]// Information & Knowledge Technology. IEEE, 2015.

[9] KHANNA G, CHATUREDI S K, SOH S. Time Varying Communication Networks: Modelling, Reliability Evaluation and Optimization[M]// Advances in Reliability Analysis and its Applications. Cham: Springer, 2020.

[10] MARIOS F, YAR R, CORNEL B, et al. Evaluating Adaptation Methods for Cloud Applications: An Empirical Study[C]// IEEE International Conference on Cloud Computing. IEEE, 2017.

[11] PENDLEBURY J, EMEAKAROHA V C, OSHEA D, et al. SOMBA-Automated Anomaly Detection for Cloud Quality of Service[C]// The 2nd International Conference on Cloud Computing Technologies and Applications—CloudTech 2016. IEEE, 2016.

[12] KIKUCHI S, MATSUMOTO Y. Using Model Checking to Evaluate Live Migrations [J]. It Professional, 2013, 15(2): 36-41.

[13] QIU X, DAI Y, XIANG Y, et al. A Hierarchical Correlation Model for Evaluating Reliability, Performance, and Power Consumption of a Cloud Service[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2016, 46(3): 401-412.

[14] QIANG L, SHEN P, DI W U, et al. Research on key technologies of highly reliable flexible distribution facing tidal load[J]. IOP Conference Series: Earth and Environmental Ence, 2020, 431(1): 12-17.

[15] MA Z, JIANG R, YANG M, et al. Research on the measurement and evaluation of trusted cloud service [J]. Soft Computing, 2016, 19(10): 86-98.

[16] LIU Z C, ZE X. Using Queue Model to Evaluation the Reliability in Cloud Platforms[J]. International Journal of Grid and Distributed Computing, 2016, 9(10): 89-98.

[17] LUO J, SONG W, YIN L. Reliable Virtual Machine Placement Based on Multi-Objective Optimization with Traffic-Aware Algorithm in Industrial Cloud [J]. IEEE Access, 2018: 23043-23052.

[18] ZHANG W, WANG Y. Adaptive Management and Multi Objective Optimization of Virtual Machine Placement in Cloud Computing[J]. Journal of Computational & Theoretical Nanoence, 2016, 13(12): 9683-9687.

[19] CHEN S, MO B, HAN X, et al. A Classification Method of Oracle Materials Based on Local Convolutional Neural Network Framework[J]. IEEE Computer Graphics and Applications, 2020(99): 1-10.

[20] ZHANG F, LIU B, WANG G, et al. Research on the Control Method of Coal Sample Blanking Based on the BP Neural Network and the PID Algorithm[M]// Recent Trends in Intelligent Computing, Communication and Devices. Springer Nature Singapore Pte Ltd, 2020.

[21] CHEN J, ZHOU B, LIAO X D. Study on Fault Identification and Warning Based on BP Neural Network for Emu Passenger Compartment Air Conditioning[J]. Railway Standard Design, 2018, 34(11): 2302-2312.

[22] BRUSKILLl A W, LUBITZ W D. A neural network shelter model for small wind turbine siting near single obstacles[J]. Wind & Structures an International Journal, 2012, 15(1): 43-64.

[23] CHEN G, WU D, CHEN G, et al. Research on Digital Forensics Framework for Malicious Behavior in Cloud[C]// 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2020.

[24] SUNGHO K, SOOJUNG L, KIWON L, et al. Usability Evaluation of Graphic User Interfaces for a Military Computer-Based Training System[J]. Journal of the Ergonomics Society of Korea, 2015, 34(5): 401-410.

[25] KUMAR R, BAZ A, ALHAKAMI H, et al. A Hybrid Model of Hesitant Fuzzy Decision-Making Analysis for Estimating Usable-Security of Software[J]. IEEE Access, 2020, 8(4): 72694-72712.

[26] HUANG W T, ZHOU M A. Research and optimization of high availability for MapReduce[J]. Computer Engineering and Design, 2014, 11(2): 102-109.

[27] MA Z C, YANG P, ZHANG X F, et al. Study on dynamic risk assessment method of electric power system in cloud computing environment[J]. Modern Electronics Technique, 2016, 1(3): 105-112.

[28] LIU Y Q, SHAO H R, JING W P. DAG Task Scheduling Integrating with Security and Availability in Cloud Environment [J]. Computer Engineering, 2014, 40(12): 12-18.

[29] YANG H, KIM Y. Design and Implementation of High-Availability Architecture for IoT-Cloud Services[J]. Sensors (Basel, Switzerland), 2019, 19(15): 3276-3286.

[30] GHOSH R，LONGO F，FRATTINI F，et al. Scalable Analytics for IaaS Cloud Availability[J]. IEEE Transactions on Cloud Computing，2014，2(1)：57-70.

[31] LUO B，QIAO Y，XIAO F U. Design and Implementation of High-availability Based on OpenStack Cloud Platform[J]. Computer Science，2017，S1(3)：89-96.

[32] NZANY F，YANG Y. Task scheduling and virtual resource optimising in Hadoop YARN-based cloud computing environment [J]. International Journal of Cloud Computing，2018，7(2)：83-102.

[33] IBRAHIM E，El-BAHNASAWY N A，OMARA F A. Task Scheduling Algorithm in Cloud Computing Environment Based on Cloud Pricing Models[C]// Computer Applications & Research. IEEE，2016：65-71.

[34] DAI Y S，LEVITIN G. Reliability and performance of tree-structured grid services[J]. IEEE Transactions on Reliability，2006，55(2)：337-349.

[35] SONG Y J，YANG X Z，LI D Y，et al. Reliability Count Evaluation of Computers Based on Cloud Models for Environmental Factors[J/OL]. Journal of Computer Research and Development，2001. https://www. researchgate. net/publication/291132967_Reliability_count_evaluation_of_computers_based_on_cloud_models_for_environmental_factors.

[36] GHOMI E J，RAHMANI A M，QADER N. Applying queue theory for modeling of cloud computing：A systematic review [J]. Concurrency and Computation：Practice and Experience，2019，38(2)：631-636.

[37] SHARMA Y，JAVADI B，SI W，et al. On the reliability and energy efficiency in cloud computing[J]. Journal of Network & Computer Applications，2016，74(1)：66-85.

[38] BITTENCOURT L F，VIEIRA C，MADEIRA E. Reducing Costs in Cloud Application Execution Using Redundancy-Based Scheduling[C]// IEEE/ACM International Conference on Utility & Cloud Computing. ACM，2014.

[39] QI Z，ZHANI M F，JABRI M，et al. Venice：Reliable virtual data center embedding in clouds[C]// IEEE INFOCOM 2014—IEEE Conference on Computer Communications. IEEE，2014.

[40] EBENEZER A S，RAJSINGH E B，KALIA B. Support vector machine-based proactive fault-tolerant scheduling for grid computing environment[J]. International Journal of Advanced Intelligence Paradigms，2020，16(3)：381-385.

[41] VISHWANATH K V，NAGAPPAN N. Characterizing Cloud Computing Hardware Reliability[C]// Proceedings of the 1st ACM Symposium on Cloud Computing. ACM，2010.

[42] YANG B，HU H，GUO S. Cost-oriented Task Allocation and Hardware Redundancy Policies in Heterogeneous Distributed Computing Systems Considering Software Reliability[J]. Computers & Industrial Engineering，2009，56(4)：1687-1696.

[43] YANG B，TAN F，DAI Y S. Performance Evaluation of Cloud Service Considering Fault Recovery[J]. The Journal of Supercomputing，2013，65(1)：426-444.

XIA Jing，born in 1982，Ph.D. Her main research interests include virtualization technology，intelligent computing and intelligent cloud.

（责任编辑：喻藜）