基于三次指数平滑法和时间卷积网络的云资源预测模型

谢晓兰[1,2]，张征征[2]，王建伟[2]，程晓春[3]

（1. 桂林理工大学广西嵌入式技术与智能系统重点实验室，广西 桂林 541004；
2. 桂林理工大学信息科学与工程学院，广西 桂林 541004；3. 密德萨斯大学计算机科学系，伦敦 NW4 4BT）

摘 要：以 Docker 和 Kubernetes 为代表的容器云具有额外的资源开销更小、启动销毁时间更短等优点，但它仍然存在过度供应和供应不足等资源管理问题。为了使 Kubernetes 集群对部署在其上的应用资源使用量能“提前”响应，并根据预测值为应用及时、准确、动态地调度和分配资源，提出了一种基于三次指数平滑法和时间卷积网络的云资源预测模型，根据历史数据预测未来的资源需求。为了找到参数的最优组合，使用 TPOT 调参思想对参数进行优化。对 Google 数据集 CPU 和内存的预测实验表明，所提模型与其他模型相比具有更好的预测性能。

关键词：资源预测；Kubernetes；指数平滑法；时间卷积网络

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2019172

Cloud resource prediction model based on triple exponential smoothing method and temporal convolutional network

XIE Xiaolan[1,2], ZHANG Zhengzheng[2], WANG Jianwei[2], GHENG Xiaochun[3]

1. Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology, Guilin 541004, China
2. College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China
3. Department of Computer Science, Middlesex University, London NW4 4BT, UK

Abstract: The container cloud represented by Docker and Kubernetes has the advantages of less additional resource overhead and shorter start-up and destruction time. However there are still resource management issues such as over-supply and under-supply. In order to allow the Kubernetes cluster to respond “in advance” to the resource usage of the applications deployed on it, and then to schedule and allocate resources in a timely, accurate and dynamic manner based on the predicted value, a cloud resource prediction model based on triple exponential smoothing method and temporal convolutional network was proposed, based on historical data to predict future demand for resources. To find the optimal combination of parameters, the parameters were optimized using TPOT thought. Experiments on the CPU and memory of the Google dataset show that the model has better prediction performance than other models.

Key words: resource prediction, Kubernetes, exponential smoothing method, temporal convolutional network

收稿日期：2019−06−04；**修回日期**：2019−07−13

通信作者：张征征，zhangzhengzhengaxp@163.com

基金项目：国家自然科学基金资助项目（No.61762031）；广西创新驱动重大专项基金资助项目（No.2018AA32003）；广西重点研发计划基金资助项目（No.AB17195029, No.AB18126006）；广西硕士研究生创新基金资助项目（No.YCSW2017156, No.YCSW2018157）；广西中青年教师基础能力提升基金资助项目（No.KY2016YB184）

Foundation Items: The National Natural Science Foundation of China (No.61762031), The Science and Technology Major Project of Guangxi(No.2018AA32003), The Key Research and Development Program of Guangxi (No.AB17195029, No.AB18126006), Innovation Project of Guangxi Graduate Education (No.YCSW2017156, No.YCSW2018157), Subsidies for the Project of Promoting the Ability of Young and Middle-Aged Scientific Research in Universities and Colleges of Guangxi (No.KY2016YB184)

1 引言

从 2013 年至今，以容器为中心的容器云平台发展迅速。AWS、Azure、阿里云作为三大公有云巨头，面向所有用户提供了类别丰富、服务完善的容器产品，阿里云、腾讯云、百度云、华为云等互联网公司为用户提供了优化的容器服务产品[1-3]。目前，越来越多的互联网公司和社区对容器云技术展开深入研究，根据 CNCF 欧洲峰会的调查数据，2017 年容器的使用规模从 20％增加到 45％。Docker、Kubernetes、cloud native、Kata 和 gVisor 等容器云平台的出现，表明容器云技术始终是云计算发展的趋势。基于容器技术的容器云平台可以更轻量级地为应用程序提供构建、发布和运行分布式应用程序的环境，为在其上运行的应用程序按需提供资源，然后根据每个应用程序的资源使用情况向用户收费。

与传统的云计算相比，虽然容器云具有额外资源开销较少、启动时间较短等优点[4]，但仍存在过度供应和供应不足的资源管理问题，而且相关文献较少。在过度供应问题中，用户不会受到影响，但过度供应会增加能源浪费，甚至增加网络、冷却、维护等成本。供应不足则会导致服务水平协议（SLA, service level agreement）违约、服务质量（QoS, quality of service）降低，甚至导致用户不满从而使用户流失和收入减少。因此，为用户分配的资源应尽可能地接近用户目前的申请需求。

Kubernetes 作为目前发展比较迅速的容器云代表，为了使 Kubernetes 集群能够“预知”部署在其上应用的未来资源使用量，需要为其建立预测模型，然后根据预测值为应用及时、准确、动态地调度和分配资源。本文提出了一种基于三次指数平滑法和时间卷积网络（ES-TCN, exponential smoothing-temporal convolutional network）的云资源预测模型，根据历史数据预测未来的资源需求，与其他预测模型相比，所提模型具有更好的预测性能，对提高资源利用率、降低能耗和成本及按需规划容器资源十分有利。本文的主要贡献如下。

1) 提出了一种解决 Kubernetes 资源使用量预测的 ES-TCN 模型。根据云资源时间序列的特点，设计了一种加权分配策略，为 ES 模型和 TCN 模型分配不同的权重，从而提高了模型的预测精度、稳定性和泛化能力。

2) 使用 TPOT（tree-based pipeline optimization tool）调参思想优化 TCN 模型的参数组合，一方面减少了模型的训练时间，另一方面为 TCN 模型找到了全局最优参数组合。

3) 使用 Google 数据集对所提模型进行了评价，并与 2 种单一预测模型以及现有的其他模型进行了对比实验。

2 相关工作

云资源预测由于潜在和显著的优点，长期以来一直受到研究者的关注。针对云资源预测的不同方面、目标和应用，国内外学者提出了多种预测模型。最初，一些学者提出了基于灰色理论、回归分析等算法的云资源预测模型。灰色理论模型只能描述具有强规律性和较差普遍性的云资源序列，回归分析法虽然可以追踪云资源变化的整体趋势，但不能描述变化的随机性，而且预测误差较大，因此 2 种算法都不能对云资源进行很好的预测。

随着云计算领域的迅速发展，云计算数据海量倍增和复杂多变的特性越加凸显，传统的预测模型已经不能满足有效预测云计算资源的需求。得益于机器学习和数据挖掘技术的迅速发展，一些学者提出了基于神经网络[5-6]、隐式马尔可夫[7]、贝叶斯[8]、支持向量机[9-11]等的云资源预测模型，它们可以挖掘云计算资源负荷随机、动态的变化趋势，得到相对理想的云资源预测结果。虽然神经网络具有良好的非线性映射能力，但是它需要的样本量大，而且不同类型的云计算资源的网络结构不同，因此，神经网络预测结果的稳定性较差。此外，神经网络阈值和权重的计算采用梯度下降的方式，具有收敛速度慢、容易陷入局部最优解等不足，降低了预测精度。文献[12]提出了一种贝叶斯模型，该模型用于预测 CPU/内存密集型应用程序的短期和长期虚拟资源需求。与其他模型相比，贝叶斯模型能够更好地预测云环境中的虚拟资源。贝叶斯模型的优点是能够全面识别虚拟化云场景中变量之间的依赖关系，缺点是没有考虑几种应用程序类型的组合，而且它只适合特定的问题。支持向量机预测结果更加稳定、可靠，但其参数对云计算资源预测结果影响较大。

有研究表明，云计算的资源序列与时间高度相关[13]，因此一些学者将云资源序列看作一个时间序列，并使用时间序列预测算法对云资源进行研究。时间序列预测算法是基于时间序列的历史信息来

完成预测，它的基本思想是从现有的序列数据中发现某种隐含的规则来预测序列的未来趋势。由于能够更好地挖掘事物发展规律与时间的关系，基于时间序列的预测模型已广泛用于金融、交通等领域。

针对时间序列预测问题，国内外相关学者已经开展了一些研究。最初，人们使用布朗提出的指数平滑、自回归移动平均（ARIMA, auto regressive integrated moving average）等算法进行建模[14-15]。ES 使用加权平滑系数在时间序列模型中通过上一期预测值与真实值的加权平均得到下一期的预测值，ES 不需要像 ARIMA 那样去进行复杂的数据预处理，它的参数选择可以通过交叉验证直接实现。同时，它只需要上期的预测值与观测值，不需要大量的数据计算，因此建模速度非常快，但主要针对小规模的资源预测问题。随着资源规模的增大，ES 的预测性能也随之下降，无法很好地拟合数据，而且云资源的变化具有动态性和时变性，因此 ES 和 ARIMA 无法满足云用户的各种需求。近年来，时间序列预测在越来越多的应用领域中成为最佳预测方法。文献[16]应用非线性时间序列方法提出了一种网络传播行为预测模型，并用粒子群优化算法对模型参数取优，结果表明，短期内该模型能够对网络传播行为做出准确预测。文献[17]针对流量数据具有高度的非线性和复杂性，现有的预测方法大多不能很好地捕获数据的时空相关性的问题，因此提出了一种新颖的基于深度学习的多组件时空图卷积网络（MCSTGCN, multi-component spatial-temporal graph convolution network），以解决交通流量预测问题，结果表明，MCSTGCN 模型的预测效果优于现有的预测方法。文献[18]针对目前民航运输业对机场延误预测高精度的要求，提出了一种基于区域残差和长短时记忆（RR-LSTM, regional residual and long term memory network）网络的机场延误预测模型。实验结果表明，RR-LSTM 网络模型预测准确率可达 95.52%，取得了比传统网络模型更好的预测效果。

循环神经网络（RNN, recurrent neural network）及其变体长短时记忆（LSTM, long short term memory）网络一直被认为是时序预测领域的最佳算法，然而，文献[19]提出了一种可用于解决时间序列预测的新算法——时间卷积网络（TCN, temporal convolutional network），并在文中对 TCN 和循环网络进行了系统的评估。结果表明，TCN 在不同任务和数据集上的性能优于典型的循环网络（如 LSTM）。而且 TCN 提出后在学术界引起了巨大的反响，许多原来使用 RNN 的领域都转而使用 TCN。因此，一些学者认为 TCN 将取代 RNN 的地位。

尽管 LSTM、TCN 等时序预测模型的预测效果要远远优于以往的预测模型，然而，面对复杂多变的云计算资源，只用单一模型很难达到期望的预测效果。大量实际研究结果表明，结合各种单一模型的优势构建合理的组合优化预测模型，可以更准确、全面地描述具有复杂性、非线性、动态性特点的云资源使用量情况，显著提升预测模型的预测精度和预测性能[20-25]。

目前，有许多组合模型可用于云资源预测。文献[26]提出了一种结合线性和非线性的动态组合预测模型来预测云资源的 CPU 负载。该模型的优点是可以动态调整预测器以适应时间模式序列。然而，它并不一定保证能准确地预测各种时间序列的未来值，而且无法集成同构或异构模型。文献[27]提出了一种组合预测模型，用于预测未来云计算环境中虚拟机（VM, virtual machine）资源、基础设施和服务水平的能源效率状况。该模型的优点是适合根据云计算趋势预测云系统的能耗，缺点是准确性取决于工作负载类型，性能与应用的组合模型中的每个单一预测模型高度相关。文献[28]提出了一种使用遗传算法来结合多个模型的自适应预测方法。更准确地说，该方法是结合不同时间序列预测模型的预测值进行预测，然后采用遗传算法对每个组成模型分配一个值，即每个组成模型的比例，这些值的总和必须为 1。该模型的优点是在使用前不需要训练，它独立于其组合预测模型，而且易于扩展其预测模型的数量，然而时间复杂度太高。文献[29]针对在部署过程中对重要的资源使用预测对于实现科学应用的最优调度至关重要，现有的云资源预测模型针对云度量的高方差而不能提供合理的精度的情况提出了一种集特征选择和资源利用预测技术于一体的智能回归集成预测方法，对未来的资源需求进行准确的预测以实现资源的自动配置。虽然预测结果在各衡量指标表现很好，但该方法仅对 CPU 的使用量进行了预测，对其他资源可能存在不通用的情况。

虽然目前有许多组合模型可用于云资源预测，更深层次的是每个预测模型具有不同的性能，并且在预测某些指标方面是有效的，而在其他指标中则可能非常不准确。虽然在组合模型中将这些模型视

为组成模型可能是合适的，但它在提高组合模型某一性能的同时也可能导致其他性能变差。基于此，本文提出了 ES-TCN 模型，该模型通过给单个预测模型赋予权重得到组合模型的预测值。

3 ES-TCN 模型

指数平滑法对不同时间数据的非等权处理较符合实际情况，在实际应用中仅需选择一个模型参数即可进行预测，简便易行，而且预测模型能自动识别数据模式的变化并对其进行调整，具有适应性，但对数据的转折点不能很好地预测，长期预测的效果较差，故多用于短期预测。在 TCN 中可以进行大规模并行处理，网络训练和验证的时间都会变短。TCN 的反向传播路径和序列的时间方向不同，这避免了 RNN 中经常出现的梯度爆炸或梯度消失问题。TCN 训练时需要的内存更少，尤其是对于长输入序列。但是 TCN 在预测时需要在内存中存储足够长的原始输入信息，以保证能获取到历史信息，且在不同领域的超参数（如 i 和 d）可能不同，因此在迁移模型时需要调整这些参数。

针对此，本文提出了 ES-TCN 云资源预测模型，并根据云资源时间序列的特点，设计了一种加权分配策略，为 ES 模型和 TCN 模型分配不同的权重，从而提高了模型的预测精度、稳定性和泛化能力。ES-TCN 模型的计算式为

$$Y_t = \omega_1 E_t + \omega_2 N_t \tag{1}$$

其中，Y_t 表示最终的预测值，E_t 表示 ES 模型得到的预测值，N_t 表示 TCN 模型得到的预测值，$\omega_1+\omega_2=1$。

一次指数平滑法常用于没有明显函数规律但确实存在某种前后关联的时间序列，线性时间序列通常使用二次指数平滑法，当时间序列的变动呈现出二次曲线趋势时，常使用三次指数平滑法，因此通过分析云资源时间序列的特点，本文使用三次指数平滑法。由于高次的指数平滑法是在低次的指数平滑法的基础上建立的，例如，二次指数平滑法是建立在一次指数平滑法基础之上的，所预测的效果也优于一次指数平滑法；三次指数平滑法是在二次指数平滑法基础之上再进行一次平滑，因此设 Kubernetes 的资源使用量的值为$\{k_t\}(t=1,2,3,\cdots)$，依次计算第 t 时刻不同次数的指数平滑法，并将得到的预测值分别记为 $S_t^{(1)}$、$S_t^{(2)}$和 $S_t^{(3)}$，各指数平滑法计算式为

$$S_t^{(1)} = \alpha k_{t-1} + (1-\alpha) S_{t-1}^{(1)} \tag{2}$$

$$S_t^{(2)} = a S_t^{(1)} + (1-\alpha) S_{t-2}^{(2)} \tag{3}$$

$$S_t^{(3)} = a S_t^{(2)} + (1-\alpha) S_{t-1}^{(3)} \tag{4}$$

其中，$S_t^{(3)}$表示三次指数平滑法得到的预测值，即式(1)所述的 E_t。在指数平滑算法中，α 的取值是决定最终预测结果的关键。一般情况下，α 取值范围为 0~1，根据经验，通常在 0.10~0.80 范围内取值[30]。α 的取值一旦固定就不能修改其加权系数。通常情况下，α 取较小值时，预测模型的平滑能力较强；α 取较大值时，模型对时间序列的变化反应速度较快。不同情况下的 α 取值如表 1 所示。

表 1 α 值的适用情况

α 值	适用情况
0.10~0.30	时间序列呈现水平趋势
0.30~0.50	序列有波动但长期趋势变化不大
0.60~0.80	序列波动较大且呈现明显上升或下降趋势

用 y_t 表示 TCN 模型得到的预测值，即式(1)所述的 N_t。y_t 的求解计算式如式(5)~式(7)所示。

作为一种特殊类型的一维卷积网络（1D CNN, one-dimensional convolutional neural network），TCN 是对序列信息进行编码的一种自然方法[31]。一个普通的一维卷积层可以写成

$$F(k_t) = (k * f)(t) = \sum_{j=0}^{i-1} f_j^{\mathrm{T}} k_{t-j}, t \geqslant i$$

$$y = \left(F(k_i), F(k_{i+1}), \cdots, F(k_n)\right) \tag{5}$$

其中，k 为输入序列，y 为输出序列，$f \in R^{id}$ 为大小为 i 的卷积滤波器。然后将几个普通的一维卷积层叠加起来构成 1D CNN。然而，1D CNN 在应用于模型序列时会受输出尺寸大小和接收域的限制，因此，TCN 使用因果卷积（causal convolution）和空洞卷积（dilated convolution）2 种技术来解决这些问题。

1) 因果卷积。如式(5)所示，普通一维卷积层将长度为 n 的序列作为输入，并输出长度为 $n-i+1$ 的序列。如果将更多这样的层堆叠在一起，则可以进一步缩小误差。因为本文希望所提模型在每个时间步长进行预测并实时更新它们，这个属性在本文的域中可能会有问题。因果卷积层通过在输入序列的开头填充长度为 $i-1$ 的零来解决这个问题。此外，它能确保没有未来的信息泄露到过去，这是预测未来交互的关键。它确保了输出序列 y 在每个时间步

长上被很好地定义，并且预测值 y_t 仅取决于输入 $k_{\leqslant t}$。计算式为

$$F(k_t)=(k*f)(t)=\sum_{j=0}^{i-1}f_j^{\mathrm{T}}k_{t-j}\qquad k_{\leqslant 0}:=0$$
$$y=\left(F(k_1),F(k_2),\cdots,F(k_n)\right)\tag{6}$$

2) 空洞卷积。CNN 通常都是可以被反向训练的，但是如果使用过多的历史信息就会导致 CNN 的深度增大，当 CNN 的深度增大到一定程度时就需要一种新的办法来解决反向训练的问题。TCN 使用空洞卷积来解决普通 1D CNN 的接收域与层数呈线性关系的问题。空洞卷积是一种技术，它允许接收域与层数呈指数关系。具体地，当与因果卷积结合时，第 r 层空洞卷积可以表示为

$$F(k_t)=(k*_{l_r}f)(t)=\sum_{j=0}^{i-1}f_j^{\mathrm{T}}k_{t-l_r j}\qquad k_{\leqslant 0}:=0$$
$$y=\left(F(k_1),F(k_2),\cdots,F(k_n)\right)\tag{7}$$

其中，l_r 是扩张因子，可以设定为$(i-1)^{r-1}$，以实现指数大的感受野。式(7)为时间卷积层，通过堆叠多个时间卷积层来构造 TCN。为了便于训练深度 TCN，通常的做法是将时间卷积层组织成块，并在块之间添加残差连接[32]。通过设置合适的滤波器大小和层数，y_{t+1} 可以依赖于整个历史交互作用。

模型的参数决定了最终预测的结果，因此参数的选取至关重要。TCN 作为神经网络的一种，调参占了很大一部分工作比例，而且训练一次 TCN 需要很长的时间，因此如何正确地、快速地调参十分重要。本文使用 TPOT 调参思想对 TCN 调参，既保证了模型的最终预测效果，又减少了调参的工作量，从而缩短了整个模型训练的时间。TPOT 是一种优化机器学习模型和自动选择模型参数的工具[33]，也是自动化机器学习（AML, automatic machine learning）的一种应用，能够自动执行机器学习问题中的重复步骤，从而节省时间。同时，TPOT 也是网格搜索的扩展，提供了如遗传算法这样的应用，可用来在某个配置中调节各个参数并达到最佳设置。

4　实验结果及分析

4.1　实验参数及数据集

本文使用 Google 数据集检验 ES-CTN 模型的有效性，该数据集记录了 Google 云计算中心大量任务的提交、调度、更新、结束事件及 CPU、内存、带宽等资源需求和使用记录，是云计算的重要研究资源。考虑到目前 Kubernetes 主要是对 CPU 和内存资源进行分配和调度，因此本文只对 CPU 和内存资源进行研究。但是本文提出的模型并不局限于对 CPU 和内存资源的预测，也可以用于预测其他资源。

由于 Google 数据集的数据量庞大，使数据处理工作量大、耗时长，本文先对数据集进行处理，提取和过滤数据后将其保存为 CSV 文件，再将 CSV 文件读入 Spyder 中进行实验。由于本文提出的 ES-TCN 模型的预测结果是基于 ES 和 TCN 的预测结果的，对于寻找 ES-TCN 模型的权重，可以通过遍历权重的方式实现。具体如下：设 ES 的权重 ω_1 初始值为 1，因为 $\omega_1+\omega_2=1$，则 TCN 的初始权重 ω_2 为 $1-\omega_1=0$，令 ω_1 每次以步长为 0.01 的递减方式进行迭代，计算 ES-TCN 模型的 MAE，最终从所有遍历的结果中取出最小的 MAE 对应的迭代次数，从而得到权重 ω_1 和 ω_2。本文的实验参数设置如表 2 所示。

表 2　　实验参数

参数	值
kernel_size	2
n_layers	6
n_filters	32
learning_rate	0.000 75
training_steps	10 000
ω_1(CPU)	0.4
ω_2(CPU)	0.6
α(CPU)	0.390 73
ω_1(memory)	0.27
ω_2(memory)	0.73
α(memory)	0.344 37

由于是对云资源时间序列进行预测，因此，要保证数据的准确性和完整性，数据必须按照一定的时间顺序排列，不能出现时间序列混乱的情况。本文使用的数据集缺失值较少，因此使用简单的均值插补法对缺失值进行处理。此外，本文还对数据进行了归一化处理。

4.2　预测性能对比

本文的研究工作是在 ES 和 TCN 模型的基础上进行的，并进行了适当的优化，因此，本文选取了 ES 和 TCN 这 2 种模型与本文提出的 ES-TCN 模型进行对比。不同模型的部分比较结果分别如图 1 和图 2 所示。

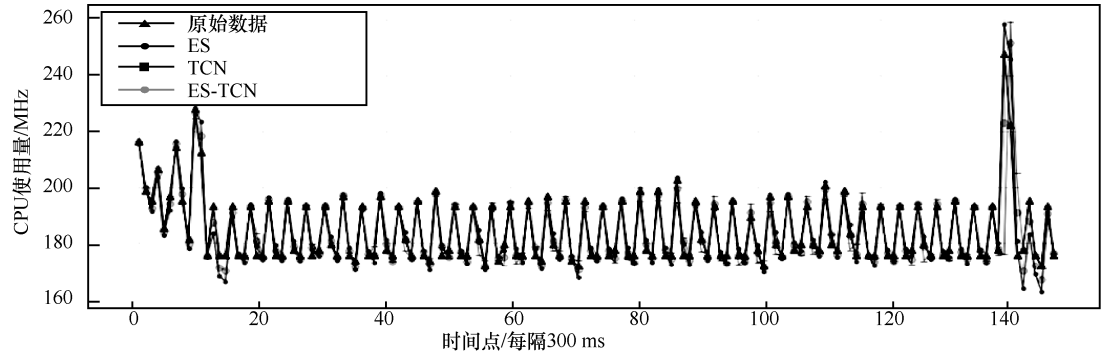

图 1 3 种模型的 CPU 预测性能对比结果

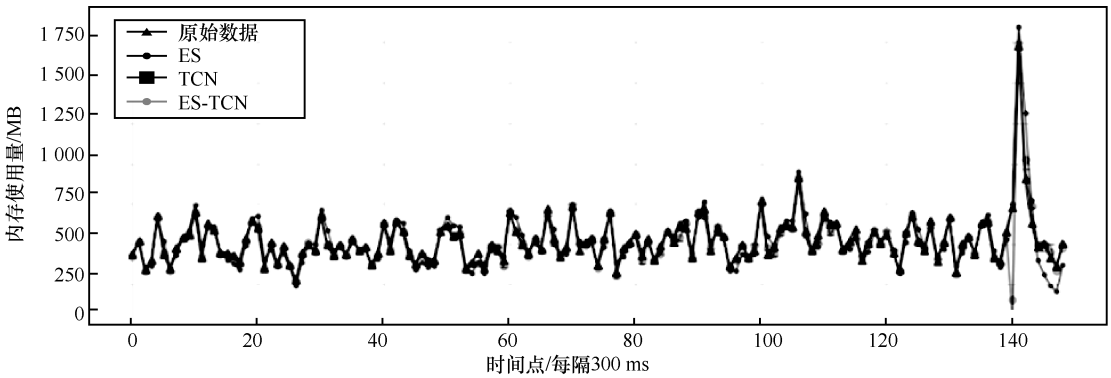

图 2 3 种模型的内存预测性能对比结果

虽然从图 1 和图 2 中可以看出 ES-TCN 模型比其他 2 种模型好，但图中展示的结果仍然不够明显。为了更清楚地看到结果，本文在实验时将 console 中的 Graphics 设置成 Qt5，通过 Qt5 自带的放大功能对预测结果进行了细微比较，得到 ES 模型在云资源规模较小时建模速度较快，且预测比较准确，但随着云资源规模的增大，预测精度迅速下降，不能很好地满足 Kubernetes 资源使用量预测需求。这是因为 ES 赋予远期较小、近期较大的比重，所以只能进行短期预测。TCN 模型虽然总体预测效果较好，但不稳定，这是因为虽然 TCN 具有良好的非线性映射能力，但是它在预测时需要在内存中存储足够长的原始输入信息以保证能获取到历史信息，因此，TCN 的稳定性较差。本文提出的 ES-TCN 模型虽然能比较好地追踪真实数据，预测效果也比较稳定，但是在波动变化较大时，预测误差比较明显，但与其他 2 种模型相比，这个预测误差相对较小。这是因为本文提出的模型结合了 ES 模型能自动识别数据模式的变化并对其进行调整，具有良好的适应性和 TCN 使用因果卷积、空洞卷积可以进一步缩小误差的优点，同时采用 TPOT 调参思想快速地为 TCN 模型找到模型参数的最优组合，提高了预测模型的准确度和稳定性及泛化能力。

4.3 预测误差对比

为了进一步说明实验结果，本文选取了常用的 4 种合适的度量标准：平均绝对误差（MAE, mean absolute error）、平均绝对百分比误差（MAPE, mean absolute percentage error）、均方根误差（RMSE, root mean square error）和平均绝对比例误差（MASE, mean absolute scaled error），用来对比模型性能。

由于 MAE 的离差被绝对值化，不会出现正负相抵消的情况，因此 MAE 能更好地反映预测误差的实际情况。MAE 的计算式为

$$\mathrm{MAE}=\frac{1}{n}\sum_{t=1}^{n}\left|\hat{y}_t-y_t\right| \tag{8}$$

如果一直让预测值远大于真实值会造成实际的资源利用率很低，使分配的资源大量闲置。因此，引入 MAPE 来避免发生预测值无限大于真实值的情况，从而避免资源浪费。MAPE 的计算式为

$$\mathrm{MAPE}=\frac{1}{n}\sum_{t=1}^{n}\left|\frac{\hat{y}_t-y_t}{y_t}\right| \tag{9}$$

RMSE 代表预测值和观察值的样本标准差，主要

用来聚集预测误差的大小，通常在不同的时间下，以一个量值来表现其预测的能力。RMSE 的计算式为

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\hat{y}_t-y_t\right)^2} \tag{10}$$

MASE 是针对所有值定义的归一化统计量，并且同等地对误差进行加权，因此它是用于比较不同预测模型质量的一个很好的指标。MASE 的计算式为

$$\mathrm{MASE}=\frac{1}{n}\sum_{t=1}^{n}\frac{\left|\hat{y}_t-y_t\right|}{\beta} \tag{11}$$

其中，$\beta=\dfrac{1}{n-1}\sum_{t=2}^{n}\left|y_t-y_{t-1}\right|$。

上述 4 种度量标准中，y_t 是真实值，$\hat{y}_t$ 是预测值，n 是时间序列长度，而且这 4 种度量标准的值越接近 0，表明真实值与预测值之间的误差越小，预测模型的性能越好。3 种预测模型的性能对比统计结果如图 3 所示。为了方便比较，首先对 CPU 和内存这 2 种资源进行了归一化处理。

此外，本文还将提出的 ES-TCN 模型与 ARIMA、RNN、LSTM 进行了对比实验，实验结果表明，与 ARIMA 模型相比，ES-TCN 模型不需要进行复杂的数据预处理；与 RNN 模型相比，ES-TCN 模型避免了 RNN 中经常出现的梯度爆炸或梯度消失问题；与 LSTM 模型相比，ES-TCN 模型可以进行大规模并行处理，网络训练和验证的时间都会变短，因此计算成本更低。

5　结束语

本文提出了一种新的云资源时间序列预测模型用来对部署在 Kubernetes 上的应用未来的资源（CPU 和内存）使用量进行预测，该模型结合了 ES 和 TCN，不仅提高了模型的泛化能力，还提高了模型的预测精度和稳定性。采用 TPOT 调参思想对参数进行优化，快速地为 TCN 模型找到模型参数的最优组合，在节省了模型训练时间、大大减少了工作量的同时，进一步提高了模型的预测性能。

为了验证本文提出的模型的预测效果，选取了 Google 云计算中心数据集的 CPU 和内存资源，并与 2 种单一预测模型以及其他模型进行了对比实验。同时，使用 4 种常用的标准指标用于比较模型的性能。实验结果表明，与其他模型相比，本文提出的模型具有更高的预测精度、稳定性和泛化能力，进一步完善了 Kubernetes，能够有效提高资源利用率，有利于容器资源的按需规划。

参考文献：

[1] BERNSTEIN D. Containers and cloud: from LXC to docker to Kubernetes[J]. IEEE Cloud Computing, 2014, 1(3): 81-84.

[2] FU S, LIU J, CHU X, et al. Toward a standard interface for cloud providers: the container as the narrow waist[J]. IEEE Internet Computing, 2016, 20(2): 66-71.

[3] VARGHESE B, SUBBA L T, THAI L, et al. Container-based cloud virtual machine benchmarking[C]//2016 IEEE International Conference on Cloud Engineering (IC2E). IEEE, 2016: 192-201.

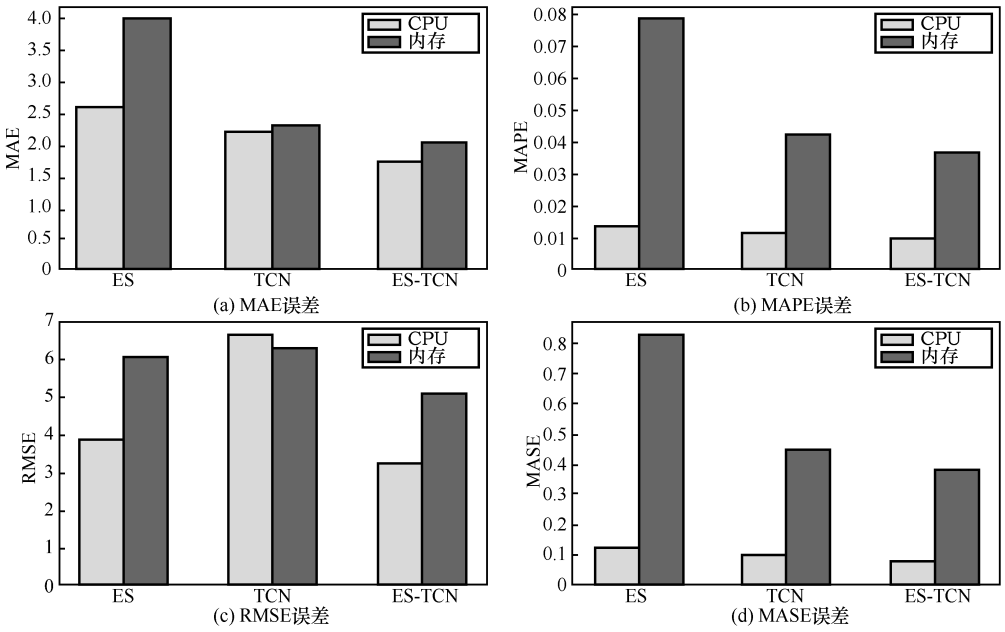

图 3　3 种预测模型的性能对比统计结果

[4] XIE X L, YUAN T W, ZHOU X, et al. Research on trust model in container-based cloud service[J]. Computers, Materials and Continua, 2018, 56(2): 273-283.
[5] ISLAM S, KEUNG J, LEE K, et al. Empirical prediction models for adaptive resource provisioning in the cloud[J]. Future Generation Computer Systems, 2012, 28(1): 155-162.
[6] DI S, KONDO D, CIRNE W. Characterization and comparison of cloud versus grid workloads[C]//2012 IEEE International Conference on Cluster Computing. IEEE, 2012: 230-238.
[7] XU D, YANG S, LIU R. A mixture of HMM, GA, and Elman network for load prediction in cloud-oriented data centers[J]. Journal of Zhejiang University: Science C, 2013, 14(11): 845-858.
[8] BASHAR A. Autonomic scaling of cloud computing resources using BN-based prediction models[C]//2013 IEEE 2nd International Conference on Cloud Networking. IEEE, 2013: 200-204.
[9] HU R, JIANG J, LIU G, et al. KSwSVR: a new load forecasting method for efficient resources provisioning in cloud[C]//2013 IEEE International Conference on Services Computing. IEEE, 2013: 120-127.
[10] RASHEDUZZAMAN M, ISLAM M A, RAHMAN R M. Workload prediction on Google cluster trace[J]. International Journal of Grid and High Performance Computing, 2014, 6(3): 34-52.
[11] ZHONG W, ZHUANG Y, SUN J, et al. A load prediction model for cloud computing using PSO-based weighted wavelet support vector machine[J]. Applied Intelligence, 2018, 48(11): 4072-4083.
[12] SHYAM G K, MANVI S S. Virtual resource prediction in cloud environment: a Bayesian approach[J]. Journal of Network and Computer Applications, 2016, 65: 144-154.
[13] JIANG Y, PERNG C, LI T, et al. Asap: a self-adaptive prediction system for instant cloud resource demand provisioning[C]//2011 IEEE 11th International Conference on Data Mining. IEEE, 2011: 1104-1109.
[14] ZHANG Q, ZHANI M F, BOUTABA R, et al. Harmony: dynamic heterogeneity-aware resource provisioning in the cloud[C]//2013 IEEE 33rd International Conference on Distributed Computing Systems. IEEE, 2013: 510-519.
[15] CALHEIROS R N, MASOUMI E, RANJAN R, et al. Workload prediction using ARIMA model and its impact on cloud applications' QoS[J]. IEEE Transactions on Cloud Computing, 2015, 3(4): 449-458.
[16] 田鹤, 赵海, 王进法, 等. 互联网传播行为的时序演化与预测[J]. 通信学报, 2018, 39(6): 116-126.
TIAN H, ZHAO H,WANG J F, et al. Timing evolution and prediction of Internet transmission behavior[J]. Journal on Communications, 2018, 39(6): 116-126.
[17] 冯宁, 郭晟楠, 宋超, 等. 面向交通流量预测的多组件时空图卷积网络[J]. 软件学报, 2019, 30(3): 759-769.
FENG N, GUO S N, SONG C, et al. Multi-component spatial-temporal graph convolution networks for traffic flow forecasting[J]. Journal of Software, 2019, 30(3): 759-769.
[18] 屈景怡, 叶萌, 渠星. 基于区域残差和 LSTM 网络的机场延误预测模型[J]. 通信学报, 2019, 40(4): 149-159.
QU J Y, YE M, QU X. Airport delay prediction model based on regional residual and LSTM network[J]. Journal on Communications, 2019, 40(4): 149-159.
[19] BAI S, KOLTER J Z, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv Preprint, arXiv:1803.01271, 2018.
[20] BUI D M, NGUYEN H Q, YOON Y I, et al. Gaussian process for predicting CPU utilization and its application to energy efficiency[J]. Applied Intelligence, 2015, 43(4): 874-891.
[21] BARATI M, SHARIFIAN S. A hybrid heuristic-based tuned support vector regression model for cloud load prediction[J]. The Journal of Supercomputing, 2015, 71(11): 4235-4259.
[22] HU W, YAN L, LIU K, et al. A short-term traffic flow forecasting method based on the hybrid PSO-SVR[J]. Neural Processing Letters, 2016, 43(1): 155-172.
[23] CHEN N, AGARWAL A, WIERMAN A, et al. Online convex optimization using predictions[C]//ACM SIGMETRICS Performance Evaluation Review. ACM, 2015: 191-204.
[24] HUANG L, CHEN M, LIU Y. Learning-aided stochastic network optimization with imperfect state prediction[C]//The 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, 2017: 12.
[25] CHEN K, HUANG L. Timely-throughput optimal scheduling with prediction[J]. IEEE/ACM Transactions on Networking, 2018, 26(6): 2457-2470.
[26] CAO J, FU J, LI M, et al. CPU load prediction for cloud environment based on a dynamic ensemble model[J]. Software: Practice and Experience, 2014, 44(7): 793-804.
[27] SUBIRATS J, GUITART J. Assessing and forecasting energy efficiency on cloud computing platforms[J]. Future Generation Computer Systems, 2015, 45: 70-94.
[28] MESSIAS V R, ESTRELLA J C, EHLERS R, et al. Combining time series prediction models using genetic algorithm to autoscaling web applications hosted in the cloud infrastructure[J]. Neural Computing and Applications, 2016, 27(8): 2383-2406.
[29] KAUR G, BALA A, CHANA I. An intelligent regressive ensemble approach for predicting resource usage in cloud computing[J]. Journal of Parallel and Distributed Computing, 2019, 123: 1-12.
[30] HUANG J, LI C, YU J. Resource prediction based on double exponential smoothing in cloud computing[C]//2012 2nd International Conference on Consumer Electronics, Communications and Networks . IEEE, 2012: 2056-2060.
[31] YOU J, WANG Y, PAL A, et al. Hierarchical temporal convolutional networks for dynamic recommender systems[C]//The World Wide Web Conference. ACM, 2019: 2236-2246.
[32] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2016: 770-778.
[33] OLSON R S, URBANOWICZ R J, ANDREWS P C, et al. Automating biomedical data science through tree-based pipeline optimization[C]//European Conference on the Applications of Evolutionary Computation. Springer, 2016: 123-137.

[作者简介]

谢晓兰（1974− ），女，广西桂林人，博士，桂林理工大学教授、博士生导师，主要研究方向为云计算、并行计算、大数据、地球物理勘查与信息技术。

张征征（1994− ），女，山东临沂人，桂林理工大学硕士生，主要研究方向为云计算、大数据。

王建伟（1993− ），男，河南周口人，桂林理工大学硕士生，主要研究方向为云计算、大数据。

程晓春（1973− ），男，吉林长春人，博士，密德萨斯大学计算机研究项目管理员，主要研究方向为智能计算、通信数据分析管理、通信安全。