计算机科学
COMPUTER SCIENCE

基于v-Informer的云平台资源负载预测方法

尤文龙, 邓莉, 李锐龙, 谢雨欣, 任正伟

引用本文

尤文龙, 邓莉, 李锐龙, 谢雨欣, 任正伟. 基于v-Informer的云平台资源负载预测方法[J]. 计算机科学, 2024, 51(12): 147-156.

YOU Wenlong, DENG Li, LI Ruilong, XIE Yuxin, REN Zhengwei. Load Prediction Method of Cloud Resource Based on v-Informer [J]. Computer Science, 2024, 51(12): 147-156.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231000098

基于 v-Informer 的云平台资源负载预测方法

尤文龙　邓　莉　李锐龙　谢雨欣　任正伟
武汉科技大学计算机科学与技术学院　武汉 430065
智能信息处理与实时工业系统湖北省重点实验室　武汉 430065
(1526347207@qq.com)

摘　要　目前,云计算技术的使用非常广泛。随着用户量的增加,云计算资源的分配管理也越来越重要,而准确的负载预测是分配管理的重要依据。但由于云平台任务有多个负载特征,且特征的相关性变化趋势各不相同,因此难以从长期的历史数据中提取出有效的依赖信息。在 Informer 模型的基础上,提出了一种针对高动态云平台任务 CPU 长期负载预测方法 v-Informer,该方法通过变分模态分解来分解负载序列中的变化趋势,引入多头自注意力机制捕获其中的长期依赖性和局部非线性关系,同时应用梯度集中技术改进优化器,减少计算开销。分别在微软云平台和谷歌云平台数据上进行实验,结果表明,与目前已有的 CPU 负载预测模型 LSTM,Transformer,TCN 和 CEEMDAN-Informer 相比,v-Informer 在 Google 数据集上的预测误差分别减少了 34%,19%,15%和 6.5%;在微软数据集上的预测误差分别减少了 32%,16%,12%和 7%,具有较好的预测精度。

关键词:云平台;CPU 负载;多步预测;模态分解;Informer;梯度收敛
中图分类号　TP391

Load Prediction Method of Cloud Resource Based on v-Informer

YOU Wenlong,DENG Li,LI Ruilong,XIE Yuxin and REN Zhengwei
College of Computer Science and Technology,Wuhan University of Science and Technology,Wuhan 430065,China
Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System,Wuhan 430065,China

Abstract　Cloud computing technology is widely used at present. With the increase in the number of users,the allocation and management of cloud computing resources is becoming more and more important,and accurate load prediction is an important basis for allocation and management. Based on the Informer model,this paper proposes a long-term CPU load prediction method for high dynamic cloud platform tasks,called v-Informer. v-Informer decomposes the variation trend in the load sequence through variational mode decomposition,and introduces a multi-head self-attention mechanism to capture the long-term dependence and local nonlinear relationship. At the same time,the gradient concentration technique is used to improve the optimizer and reduce the computational cost. Experiments are carried out on the data of Microsoft and Google cloud platforms. The results show that,compared with the existing CPU load prediction models LSTM,Transformer,TCN and CEEMDAN-Informer,the prediction error of v-Informer is reduced by 34%,19%,15% and 6.5% respectively on the Google dataset. The prediction error on the Microsoft dataset is reduced by 32%,16%,12% and 7% respectively,with better prediction accuracy.

Keywords　Cloud platform,CPU load,Multi-step forecasting,Modal decomposition,Informer,Gradient convergence

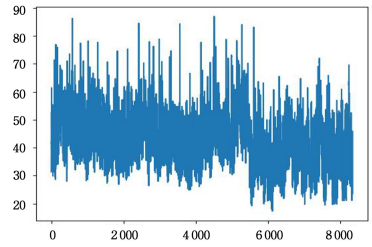
1　引言

云计算以一种经济便利的方式提供计算资源,越来越多的企业将其应用部署在云端[1]。多租户多类型应用的特性给云平台的资源管理带来了诸多挑战,如云数据中心的功耗波动更剧烈、资源利用率不平衡等[2]。谷歌于 2011 年公开的云平台资源使用 trace 数据集表明[3],CPU、内存等计算资源的使用受到多种因素的影响,难以捕捉其变化规律。图 1 描述了谷歌云平台任务(ID:2902878580)和微软 Azure 云虚拟机(ID:BgFwBxavXoeyIWqsmjgt8Q + 0A + n7z9Z3S/ShovqbTs4qCyuMqzUcPkuuk1rDq1Wj)在连续 30 天的 CPU 资源使用情况,其中谷歌 CPU 负载最小值 17.3,最大值 86.9,中位数 41.6,方差 100.5,微软 CPU 负载最小值 6.36,最大值 86.8,中位数 7.7,方差 5.5。可以看出,两个 CPU 负载序列均具有非平稳性且变化幅度大。

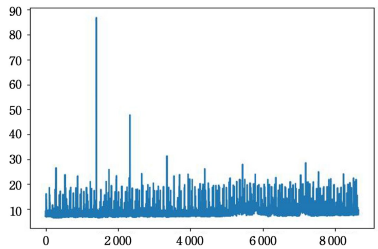
负载预测为云平台的资源管理提供有力的信息支撑,准确的多步预测可以实现更有效的资源部署和调整,从而提升用户的服务质量,提高资源利用率,减少能源消耗[4-5]。CPU

到稿日期:2023-10-16　返修日期:2024-03-11
基金项目:国家自然科学基金(61902285)
This work was supported by the National Natural Science Foundation of China(61902285).
通信作者:邓莉(dengli@wust.edu.cn)

是数据中心的核心计算资源，研究人员主要关注 CPU 资源负载的预测[6-7]。Ouhame 等[8]提出一种基于 CNN-LSTM 的预测模型，利用卷积神经网络层和长短期记忆神经网络提取虚拟机的特征，提高了预测准确率。但由于云平台任务负载变化幅度和频率无规律，这类方法对预测精度的提升仍然有限。Zhou 等[9]应用变分模态分解和 R-Transformer 集成预测方法，有效应对云平台负载序列的非平稳性和高随机性，在谷歌和阿里巴巴集群跟踪数据集上提高了预测精度。但该方法没有考虑预测较远未来的工作负载，对云平台资源管理的帮助有限。



(a)Google 平台任务负载(ID:2902878580)



(b)微软云平台虚拟机负载(ID:BgFwBxavXoeyIWqsmjgt8Q+0A+n7z9Z3S/ShovqbTs4qCyuMqzUcPkuuk1rDq1Wj)

图 1 Google 数据集某任务 CPU 平均负载和微软数据集某虚拟机 CPU 平均负载

Fig. 1 Average CPU loads of certain task in Google trace and certain VM in Azure trace

现有的研究工作很少关注资源负载的长期预测。一方面，云资源负载预测本身受多种因素的影响；另一方面，现有长期预测方法大多是基于多个单步预测模型的叠加，难以达到较好的可接受的长期预测性能。云资源负载的长期预测主要面临以下挑战：

1)相互关联的多维数据。云平台任务类型多而杂，任务类型及其用户的实时请求变化、不同资源之间的相互依赖等，都会引起任务资源使用的波动和变化。

2)不可忽视的累计误差。当前多步预测方法主要是多个单步预测模型的叠加，n 步预测需要叠加 n 个单步预测模型，导致误差的累积，从而极大地影响长期负载的预测精度。

3)庞大的计算开销。预测步数越多，需要接收的输入序列越长，需要占用的内存容量就越大，计算量也会随之增加。

针对上述问题，本文提出了一种基于变分模态分解的编码器-解码器预测模型 v-Informer(Variational Mode Decomposition-based Informer)，首先通过变分模态分解将原始资源负载序列分解成若干个子序列，使用自注意力机制学习多特征之间的依赖关系，然后拼接各个子序列的结果得到最终预测结果。该方法的主要创新性工作表现在：

1)采用变分模态分解将资源负载数据序列分解为多个具有稳定特性的本征模态函数(Intrinsic Mode Function，IMF)，有效提取出原始序列中的非线性特征。

2)采用 encoder-decoder 结构，使用稀疏自注意力减少内部计算开销，同时将 Adam 优化器中的梯度收敛方式替换成梯度集中(Gradient Centralization，GC)，以加快收敛速度并提高计算精度。

本文第 2 章介绍相关研究工作；第 3 章介绍数据分析内容；第 4 章介绍提出模型的过程和方法论；第 5 章介绍模型在 trace 上的实验效果；最后总结全文并展望未来。

2 相关工作

目前云计算环境资源负载预测的研究工作可分为基于传统回归模型、基于经典机器学习方法和基于深度学习模型等。这些方法都是从历史工作负载数据中提取依赖信息来预测未来的变化。

自回归、移动平均和自回归综合移动平均都属于传统回归模型。Hu 等[10]提出了一种基于自回归的预测方法，但该模型本质上是严格线性的，缺乏对复杂环境下工作负载的适应性。

基于传统机器学习的负载预测方法训练参数较少，所花费的计算开销较小，在数据集较小、负载模式简单时更具优势。Zhong 等[11]提出一种加权小波支持向量机预测模型，通过粒子群算法来优化参数，并在谷歌数据集上进行了验证。Peng 等[12]提出一种鲸鱼优化算法结合极限学习机的预测模型，该模型具有很强的非线性映射能力。Kumar 等[13]提出一种自适应差分进化算法，其准确程度优于粒子群算法和遗传算法。

基于深度学习的方法在应对复杂庞大的数据时更有优势，其通过增加训练参数和神经网络层数来提高预测精度。Duggan 等[14]使用经典的 RNN 架构来预测云数据中心的未来工作负载，在处理短期依赖时表现出良好的效果，但针对长期依赖效果并不明显。文献[15]提出一种基于 LSTM 的 N-LSTM 模型，实现在不规则时间间隔内的预测，可适应复杂的负载变化。文献[16]提出 CNN-LSTM 模型，利用向量自回归方法过滤多元数据之间的线性相互依赖关系，然后利用 CNN 层提取特征信息，再输入 LSTM 层生成预测。该方法可以有效预测具有复杂趋势的云工作负载。文献[17]提出基于编码器-解码器的 Bi-LSTM 网络，结合自注意力机制，自适应学习多元数据的长期依赖性和潜在的相关性特征。尽管该模型同时使用 LSTM 和 Bi-LSTM 增强了序列上下文的学习能力，但能预测的步数仍然较少，提前预测的时间有限。

现有云计算资源负载预测的研究工作大多关注未来短时期内的预测。本文提出一种基于变分模态分解的编码器-解码器预测模型 v-Informer，先通过变分模态分解提取原始负载序列中的非线性特征，然后使用自注意力机制学习多个特征之间的依赖信息，并借助稀疏自注意力、梯度集中等加快收敛速度和提高计算精度，在真实世界云 trace 数据集上得到了较好的长期负载预测效果。

3 数据分析

2011 年 Google 发布了其生产集群约 12 500 台计算机为

期 29 天的资源使用状态监测数据集，本文从任务资源使用情况表（task_resource_usage. csv）中随机选取 50 个任务在 29 天内的资源使用情况。任务资源表只记录了每个任务中包含的各个作业的资源负载，因此需要把每个任务在同一个时间戳下所有作业的资源负载转换为当前时间戳下任务的负载信息。2019 年微软云平台 Azure 发布了约 200 万台虚拟机持续 30 天的跟踪数据集，本文从虚拟机信息表中随机选择 50 个虚拟机，提取这些虚拟机在 30 天内的 CPU 使用情况。

首先对这些提取的数据做自相关性分析，然后进行预处理。

3.1 自相关性

在时间序列数据中，自相关反映某一个时刻的值和另一个时刻的值之间的关联程度。自相关性的值可以衡量序列数据在时间上的依赖程度，自相关性 r 的计算式如式(1)所示：

$$r=\frac{\sum_{i=1}^{n-k}(v_i-\bar{v})(v_{i+k}-\bar{v})}{\sum_{i=1}^{n}(v_i-\bar{v})^2} \tag{1}$$

其中，v_i 为时间戳 i 的任务负载值，$\bar{v}$ 为任务的平均负载值，k 为滞后阶数。图 2 展示了 Google 任务（ID:4665712499）平均 CPU 使用率的自相关图。可以看出，平均 CPU 使用率在相邻的时间间隔内具有较强的相关性，时间间隔越大，自相关性越弱。在滞后阶数为 1 时，自相关系数为 0.988，滞后阶数为 40 时，自相关系数为 0.473。自相关性分析为任务资源负载预测的可行性提供了理论依据。表 1 列出了 Google 任务（ID:4665712499）平均 CPU 负载的相关统计信息，这些统计信息是基于负载的 0-1 归一化值而进行的。可以看到，该任务平均 CPU 负载的标准差为 0.42，变异系数为 1.32。这些数值表明，任务资源负载在时间序列上具有一定的离散性，波动性较强，变化幅度大，增加了任务资源负载预测的难度。

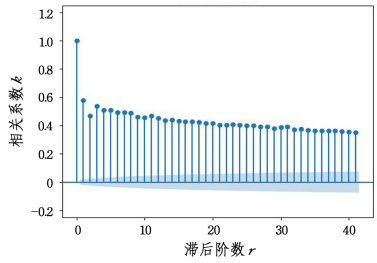


图 2 Google 任务（ID:4665712499）CPU 自相关图

Fig 2 CPU autocorrelation plot of Google task(ID:4665712499)

表 1 Google 任务（ID:4665712499）统计信息

Table 1 Google task statistics(ID:4665712499)

统计量	统计值
平均值	0.320 688 549 877 015 9
标准差	0.423 798 550 002 705 44
方差	0.179 605 210 984 395 6
变异系数	1.321 526 915 024 662 5

3.2 数据预处理

首先对提取的数据进行填充，以保证数据的完整性。然后分析任务负载预测关联的影响因素，确定负载预测所需的特征数据。最后对得到的数据进行标准化处理。

真实云环境监控的跟踪数据，由于机器错误或者人为错误等，常常会有遗失信息。在真实云平台监控数据中，缺失数据量并不少。例如：在 Google 跟踪数据中，约有 10%的缺失数据，连续缺失数据的量多达 126 个。这种情形下，单纯的前向/后向填充方法更容易引入失真数据，对预测模型的训练效果影响更大。K-最近邻（K-Nearest Neighbor，KNN）方法通过欧几里得距离标准查找最近的邻居，综合考虑最近邻居的加权特征值来估算缺失值，使得缺失数据的填充值更接近于真实值。本文采用 KNN 进行缺失数据的填充。

填充完缺失值后，还需要对这些数据进行特征选择，即去除冗余特征或不相关特征。适合的特征选择方法可以有效提高模型预测速度和精度[18]。本文使用的是基于相关性的特征选择（Correlation-based Feature Selection，CFS），CFS 可以快速筛选出和预测目标相关的特征集。从空的特征集开始，CFS 使用信息增益计算各个备选特征和已选特征集之间的相关性，采用正向最佳优先策略确定本次选择的特征，并将该特征加入已选特征集中，直到无新的特征加入。经过特征选择后，谷歌数据集的特征数量由 15 缩减为 10，特征具体内容如表 2 所列。由于微软数据集的特征均与 CPU 使用率直接相关，因此特征数量未缩减，其特征具体信息如表 3 所列。

表 2 谷歌数据集特征

Table 2 Characteristics of Google dataset

特征名	特征含义
mean CPU usage rate	5 min 内平均 CPU 使用率
canonical memory usage	用户规范内存使用量
assigned memory usage	分配的内存使用量
unmapped page memory usage	未映射的页内存使用量
total page memory usage	总页内存使用量
maximum memory usage	最大内存使用量
memory access cpi	每条指令的内存访问周期
mean local disk space used	平均使用的本地磁盘空间
maximum CPU usage	最大 CPU 使用量
cycles per instruction(CPI)	每条指令的周期

表 3 微软数据集特征

Table 3 Microsoft dataset characteristics

特征名	特征含义
Min CPU	5 min 内 CPU 最小使用率
Max CPU	5 min 内 CPU 最大使用率
Avg CPU	5 min 内 CPU 平均使用率

特征选择后的谷歌数据特征与 CPU 平均使用率的相关系数绝对值如图 3 所示，按大小来划分比例，其中除 memory access cpi 是负相关，其余特征均为正相关。由图 3 可以看出，与内存相关的特征占比普遍比较大，说明内存对 CPU 使用率有较大的正向影响。

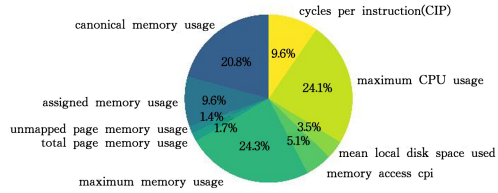


图 3 谷歌数据集特征与平均 CPU 使用率相关性图

Fig. 3 Correlation between Google dataset characteristics and average CPU utilization

数据预处理的最后步骤是数据标准化。本文使用最大-

最小线性归一化方法来标准化数据，以便统一各特征的变化幅度[19]，处理方式如式(2)所示：

$$v'=\frac{v-v_{\min}}{v_{\max}-v_{\min}} \tag{2}$$

其中，$v_{\min}$是该特征在所有样本中的最小值，$v_{\max}$是该特征在所有样本中的最大值；v是该特征实际值；v'是归一化后的标准值，取值范围为[0,1]。

4 v-Informer 预测方法

v-Informer 预测方法基于编码器-解码器结构，首先通过变分模态分解将原始资源负载序列分解成若干个子序列，有效提取出原始序列中的非线性特征，并借助自注意力机制学习多特征之间的依赖关系，最后拼接各个子序列的结果得到最终预测结果。v-Informer 模型结构如图 4 所示。

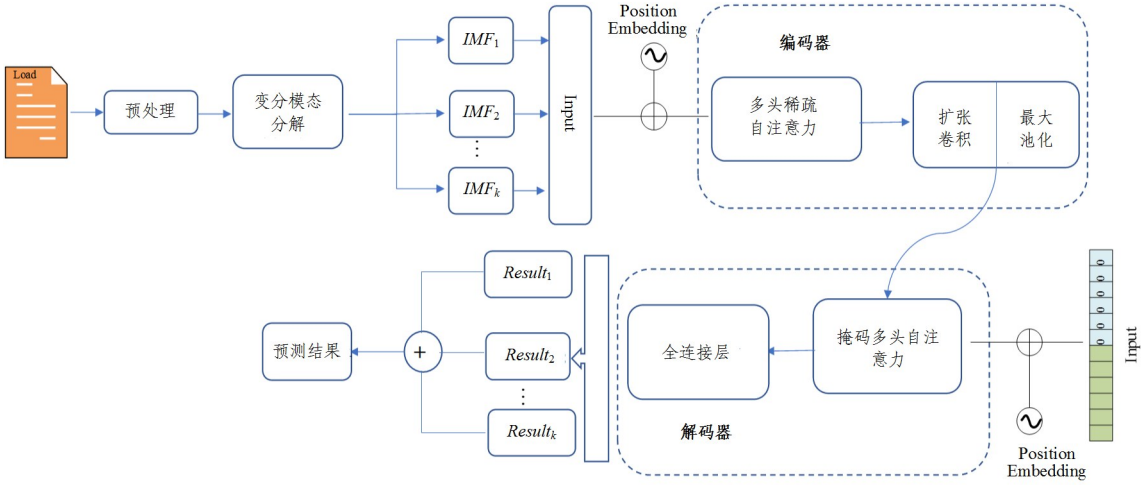


图 4 v-Informer 模型结构图

Fig. 4 Architecture of v-Informer

另外，在 v-Informer 预测方法中，Adam 优化器使用梯度集中作为收敛规则，以提高模型训练速度和预测精度。

4.1 原始资源负载序列分解

云平台数据中心的负载数据往往不是平稳变化的，而是存在一定的趋势性、季节性和周期性，如节假日用户访问量的增加、用户在白天和黑夜使用量的差异。同时，在采集云平台负载数据时难免会引入噪声数据，直接在原始资源负载序列的基础上进行预测很难得到精确的结果。因此本文引入变分模态分解(Variational Mode Decomposition，VMD)处理原始序列。

VMD 是一种自适应、非递归的信号处理方法，可以将输入信号分解为 K 个离散的子模式，每个子模式在频域谱中具有有限的带宽并且都可以在中心频率ω_k进行压缩，从而降低时间序列的复杂度和非平稳性，获得包含多个不同频率尺度且相对平稳的子序列。首先构建变分问题，使子模式的带宽之和最小，相应的约束变分如式(3)所示：

$$\min_{\{\boldsymbol{u}_k\},\{\omega_k\}}\left\{\sum_K \left\|\partial_t\left[\left(\delta(t)+\frac{j}{\pi t}\right)*\boldsymbol{u}_k(t)\right]e^{-j\omega_k t}\right\|_2^2\right\}$$
$$\text{s.t.}\ \sum_k \boldsymbol{u}_k=\boldsymbol{f} \tag{3}$$

其中，K为需要分解的模态个数，$\{\boldsymbol{u}_k\}$和$\{\omega_k\}$是所有子模式及其中心频率的简写符号，$\delta(t)$为狄拉克函数，$*$为卷积运算符，$\boldsymbol{f}$是原始信号。为了将约束变分问题转换为非约束变分问题，引入拉格朗日乘数λ，如式(4)所示：

$$L(\{\boldsymbol{u}_k\},\{\omega_k\},\lambda)=\alpha\sum_k\left\|\partial_t\left[\left(\delta(t)+\frac{j}{\pi t}\right)*\boldsymbol{u}_k(t)\right]e^{-j\omega_k t}\right\|_2^2+\left\|f(t)-\sum_k\boldsymbol{u}_k(t)\right\|^2+\left\langle\lambda(t),f(t)-\sum_k\boldsymbol{u}_k(t)\right\rangle \tag{4}$$

其中，α是二次惩罚因子。现在，原始最小化问题(3)的解决方案转化为寻找迭代子优化中增广拉格朗日量的鞍点。利用交替方向乘子迭代算法结合 Parseval/Plancherel 以及傅里叶等距变换，得到各模态分量和中心频率，寻优迭代得到最终的$\boldsymbol{u}_k$，ω_k和λ。

通过 VMD 获取的子模态被称为本征模态函数信号，在本文中，可以将其看作在相同时序内不同的变化趋势。与原始序列相比，IMF_i的非平稳性降低，每个 IMF 包含原始序列不同频率的信息。如图 5 所示，将 ID 为“BgFwBxavXoeyIWqsmjgt8Q + 0A + n7z9Z3SSHovqbTs4qCyuMqzUcPkuuk1rDq1Wj”的微软虚拟机平均 CPU 负载进行变分模态分解，IMF_1包含原始序列的低频部分，可以看作是原始序列的平滑趋势变化；剩下的 IMF 包含高频部分，反映了原始序列的细节趋势。表 4 列出了分解后的IMF_i与原始序列的皮尔逊相关系数，可以看到 3 个 IMF 与原始序列的相关系数均大于0.3，说明每个 IMF 都包含部分原始特征。将分解得到的 IMF 信号独立输入预测模型，学习到更细粒度的原始特征，输出每个 IMF 的预测结果，将这些结果相加得到最终的 CPU 工作负载预测。

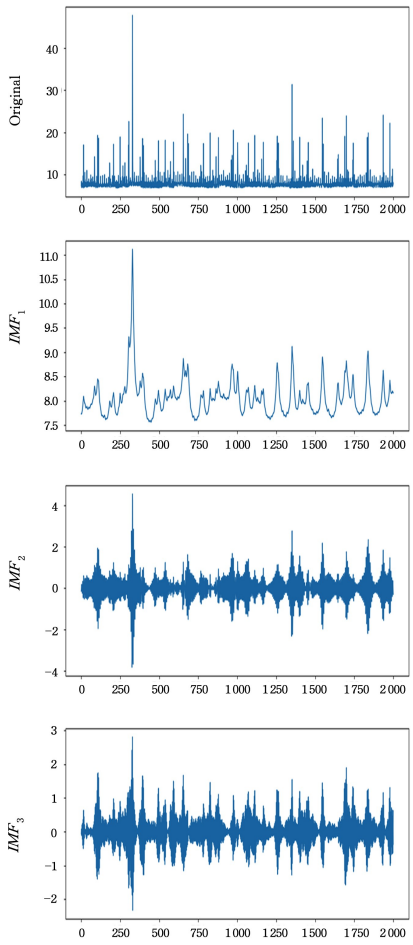


图 5 使用 VMD 分解微软数据集的样本

Fig. 5 Example of decomposing Microsoft dataset using VMD

表 4 分解后的 IMF_i 与原始序列的相关系数

Table 4 Correlation coefficient between IMF_i and the original series

子模态	相关系数
IMF_1	0.337
IMF_2	0.455
IMF_3	0.383

4.2 特征依赖关系学习

多头自注意力机制提升了模型专注于不同位置的能力，可以捕捉更多的信息。它利用多个查询，并行计算输入信息，每个自注意力头关注输入信息的不同特征，然后进行拼接，从而更好地学习负载特征之间的联系，增强了模型的非线性拟合能力，对于学习负载序列的长距离依赖关系非常有效[20]。多头自注意力机制的计算式如式(5)、式(6)所示：

$$Multi(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=Conc(head_1,\cdots,head_n)\boldsymbol{W}^O \tag{5}$$

$$head_i=Attention(\boldsymbol{Q}\boldsymbol{W}_i^Q,\boldsymbol{K}\boldsymbol{W}_i^K,\boldsymbol{V}\boldsymbol{W}_i^V) \tag{6}$$

其中，$\boldsymbol{W}_i^Q\in\mathbb{R}^{d_{\text{model}}\times d_Q}$，$\boldsymbol{W}_i^K\in\mathbb{R}^{d_{\text{model}}\times d_k}$，$\boldsymbol{W}_i^V\in\mathbb{R}^{d_{\text{model}}\times d_V}$，$\boldsymbol{W}^O\in\mathbb{R}^{hd_V\times d_{\text{model}}}$ 均是可以学习的参数。为了减少计算时间、降低空间复杂度，使用稀疏自注意力代替原始自注意力。稀疏自注意力的计算式如式(7)所示[20]：

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\frac{\bar{\boldsymbol{Q}}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d}}\right)\boldsymbol{V} \tag{7}$$

其中，$\bar{\boldsymbol{Q}}$ 是 $\boldsymbol{Q}$ 的稀疏矩阵，它只包含稀疏度量 $M(q_i,\boldsymbol{K})$ 下的前 u 个查询。第 i 个查询的稀疏度量如式(8)所示[20]：

$$M(q_i,\boldsymbol{K})=\ln\sum_{j=1}^{L_K}\mathrm{e}^{\frac{q_ik_j^{\mathrm{T}}}{\sqrt{d}}}-\frac{1}{L_k}\sum_{j=1}^{L_K}\frac{q_ik_j^{\mathrm{T}}}{\sqrt{d}} \tag{8}$$

其中，第一项是 q_i 在所有键上的对数和，第二项是它们的算数平均值。为了提取具有更细粒度信息的特征图，堆叠了多个自注意力块，同时在每个自注意力块之间引入注意力蒸馏机制以抑制内存消耗的增加。自注意力块包含一维卷积、激活函数和最大池化操作，通过将前一层的序列长度减半，解决了内存占用过多的问题，如式(9)所示[20]：

$$\boldsymbol{X}_{j+1}^t=MaxPool(ELU(Conv1d([\boldsymbol{X}_j^t]_{AB}))) \tag{9}$$

其中，$\boldsymbol{X}_{j+1}^t$ 是当前稀疏自注意力层的输出，$[\boldsymbol{X}_j^t]_{AB}$ 是上一稀疏自注意力层的输出，ELU 是激活函数。

4.3 以梯度集中为收敛规则的优化器

虽然 Informer 模型已经减少了内存和计算量的使用，但是由于云计算平台任务数量庞大，因此需要在保证预测精度的同时尽可能提升预测速度。梯度集中是一种高效、稳定的优化器技术，它通过将梯度向量集中到零均值来直接对梯度进行操作，可以被看作是一种具有约束损失函数的投影梯度下降方法[21]。如图 6 所示，$\boldsymbol{W}$ 是权重矩阵，ℓ 是损失函数，$\nabla_{\boldsymbol{W}}\mathcal{L}$ 是权重的梯度，$\phi_{\text{GC}}(\nabla\boldsymbol{w}_i\mathcal{L})$ 是集中后的梯度，用 $\phi_{\text{GC}}(\nabla\boldsymbol{w}_i\mathcal{L})$ 替换 $\nabla_{\boldsymbol{W}}\mathcal{L}$，即可将 GC 嵌入到当前的网络优化器中。

对于一个全连接层或卷积层，假设已经通过反向传播得到了梯度 $\nabla_{\boldsymbol{W}}\mathcal{L}$，对于权值矩阵 $\boldsymbol{W}$ 的第 i 个向量 $\boldsymbol{W}_i$ 对应的梯度 $\nabla\boldsymbol{w}_i\mathcal{L}$ 做出的梯度集中处理如式(10)所示[21]：

$$\phi_{\text{GC}}(\nabla\boldsymbol{w}_i\mathcal{L})=\nabla\boldsymbol{w}_i\mathcal{L}-\mu_{\nabla\boldsymbol{w}_i}\mathcal{L} \tag{10}$$

其中，$\mu_{\nabla\boldsymbol{w}_i}\mathcal{L}=\frac{1}{M}\sum_{i=1}^{M}\nabla_{\boldsymbol{W}_{i,j}}\mathcal{L}$，即对每个列向量减去其均值，整体操作可以用一个矩阵 $\boldsymbol{P}$ 来表示，如式(11)所示：

$$\phi_{\text{GC}}(\nabla_{\boldsymbol{W}}\mathcal{L})=\boldsymbol{P}\nabla_{\boldsymbol{W}}\mathcal{L},\boldsymbol{P}=\boldsymbol{I}-\mathbf{e}\mathbf{e}^{\mathrm{T}} \tag{11}$$

其中，$\mathbf{e}=\frac{1}{\sqrt{M}}\mathbf{1}$ 表示一个 n 维单位向量，$\mathbf{1}$ 表示一个 n 维单位矩阵。

图 7 以矩阵的形式详细展示了梯度平均值的操作，定义全连接层的权重矩阵为 $\boldsymbol{W}_{\text{fc}}\in\mathbb{R}^{C_{\text{in}}\times C_{\text{out}}}$，卷积层的权重张量为 $\boldsymbol{W}_{\text{Conv}}\in\mathbb{R}^{C_{\text{in}}\times C_{\text{out}}\times K}$，$C_{\text{in}}$ 和 C_{out} 分别是输入和输出通道数，K 是卷积核大小。通过对权重矩阵(张量)的每个列向量(切片)均减去该列向量(切片)的均值，使得每个列向量均值为零，完成梯度集中操作。

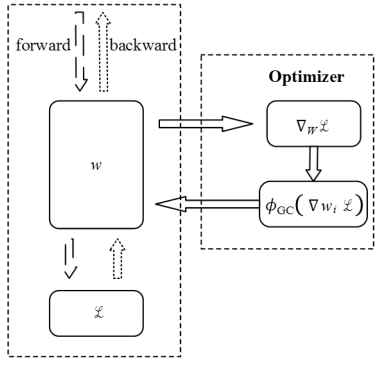


图 6 梯度集中示意图

Fig. 6 Illustration of gradient concentration

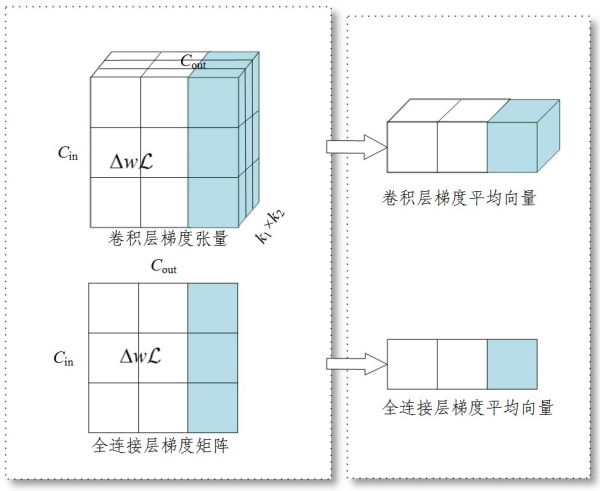


图 7 卷积层/全连接层计算梯度矩阵/张量的切片/列平均值图解

Fig. 7 Illustration of slice/column averages of computed gradient matrices/tensors by convolutional/fully connected layers

GC 通过对权重向量引入新的约束来约束损失函数，使得权重空间和输出特征空间正则化，提高模型的泛化能力，并且比原损失函数具有更好的利普西茨性，使得训练过程更加稳定和高效[21]。通过梯度集中的收敛方法，可以提高训练速度和稳定性，减少模型训练所需要的时间，提高预测速度。

5 实验

本文基于 Google 和微软公开发布的 2 个数据集进行测试，对比了 v-Informer 和目前主流方法的性能。实验结果表明，v-Informer 具有较好的预测性能。同时，也进行了消融实验，验证 v-Informer 方法各组成要素的重要性。

5.1 实验环境及数据

实验在 6 核 CPU(型号 intel i5-12400F)、16 GB 的 DDR4 内存、NVIDIA GeForce RTX 2060 12 GB GPU、win10 操作系统的环境下进行。

实验使用的数据集为 2011 年发布的 Google 云平台跟踪数据集[1)] 和 2019 年发布的微软 Azure 云平台跟踪数据集[2)]，按 6:2:2 的比例将数据集划分训练集、验证集和测试集。

从 google trace 中随机选择 50 个任务在连续 29 天内的资源使用状态，从 Azure trace 中随机选择 50 个虚拟机在连续 30 天内的资源使用状况。

5.2 评价指标

本文使用均方误差(Mean Square Error，MSE)以及平均绝对误差(Mean Absolute Error，MAE)来评价不同方法在任务 CPU 负载预测的性能。MSE 是比较常用的损失函数，其值越小，说明模型的预测精度更高；MAE 可以准确反映出实际误差的大小，同时避免误差相互抵消的情况。两个指标的定义如下所示：

$$MSE=\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i)^2 \tag{12}$$

$$MAE=\frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i-y_i| \tag{13}$$

其中，n 表示参与模型测试的样本数，$\hat{y}_i$ 表示 CPU 平均使用率的预测值，y_i 表示 CPU 平均使用率的真实值。

本文将 v-Informer 和多步 LSTM(n-LSTM)[22]、Transformer[23]、TCN[24]、Informer[20] 以及 CEEMDAN-Informer[25] 进行了 CPU 负载信息的预测性能的对比，其中 Informer 模型源自文献[20]，本文将其迁移来预测 CPU 负载信息。通过网格搜索找到 v-Informer 最优超参数，超参数列表如表 5 所列。

表 5 超参数

Table 5 Hyperparameters

参数名	数值
factor	5
d_model	512
Patience	5
k	3
n_heads	(8,3)
e_layers	2
dropout	0.05
batch_size	32
learning_rate	0.001
seq_len	(30,60,90,100,140)
label_len	(20,40,60,50,70)
pred_len	(10,20,30,50,70)

5.3 实验结果分析

n-LSTM，Transformer，TCN，Informer，CEEMDAN-Informer 和 v-Informer 分别在 Google 数据集中 50 个任务上的预测性能的平均值如表 6 所列，在微软数据集中 50 个虚拟机上的预测性能的平均值如表 7 所列。

表 6 基于 Google 数据集预测性能

Table 6 Prediction performance based on Google trace

步长	n-LSTM		Transformer		TCN		Informer		CEEMDAN-Informer		v-Informer	
	MSE	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*
10	0.0181	0.1080	0.0163	0.1020	0.0160	0.1000	0.0150	0.0949	0.0146	0.0916	**0.0139**	**0.0867**
20	0.0219	0.1230	0.0175	0.1090	0.0170	0.1070	0.0158	0.1000	0.0155	0.0982	**0.0146**	**0.0926**
30	0.0226	0.1330	0.0186	0.1180	0.0180	0.1140	0.0169	0.1090	0.0161	0.1040	**0.0152**	**0.0947**
50	0.0258	0.1470	0.0201	0.1260	0.0193	0.1210	0.0182	0.1150	0.0172	0.1110	**0.0162**	**0.1030**
70	0.0292	0.1640	0.0215	0.1320	0.0205	0.1270	0.0192	0.1220	0.0183	0.1170	**0.0171**	**0.1090**

1) https://github.com/google/Clusterdata/blob/master/ClusterData2011_2.md

2) https://github.com/Azure/AzurePublicDataset/blob/master/AzurePublicDatasetV2.md

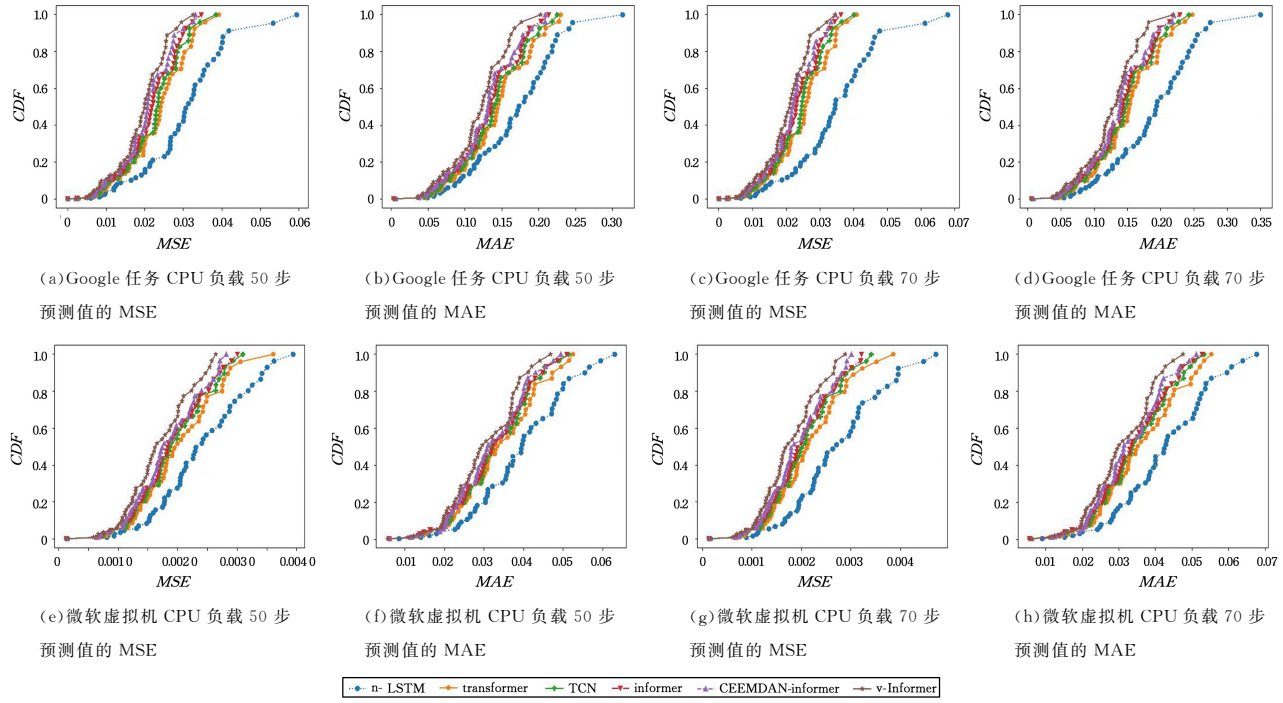
表 7 基于微软数据集预测性能

Table 7 Prediction performance based on Microsoft trace

步长	n-LSTM		Transformer		TCN		Informer		CEEMDAN-Informer		v-Informer	
	MSE	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*
10	0.00167	0.0288	0.00153	0.0280	0.00147	0.0274	0.00142	0.0267	0.00140	0.0265	**0.00129**	**0.0256**
20	0.00186	0.0307	0.00161	0.0291	0.00156	0.0284	0.00151	0.0276	0.00147	0.0274	**0.00135**	**0.0261**
30	0.00195	0.0326	0.00173	0.0301	0.00167	0.0292	0.00163	0.0286	0.00158	0.0282	**0.00147**	**0.0267**
50	0.00215	0.0360	0.00180	0.0308	0.00173	0.0299	0.00169	0.0295	0.00163	0.0291	**0.00151**	**0.0276**
70	0.00242	0.0389	0.00192	0.0325	0.00182	0.0312	0.00177	0.0305	0.00171	0.0298	**0.00157**	**0.0280**

可以看出，步长为 50 时，v-Informer 在 google 任务数据下的 MSE 和 MAE 相比 n-LSTM 分别降低了 37.2% 和 29.9%；在微软虚拟机数据下的 MSE 和 MAE 相比 n-LSTM 分别降低了 29.8% 和 23.3%，明显优于 n-LSTM。这表明基于 Informer 的模型比基于 LSTM 的模型拥有更好的预测能力，体现了编码器-解码器结构和多头自注意力减少误差的累积、增强对特征依赖关系学习的能力。同时，v-Informer 在微软虚拟机和 Google 任务数据上的 MSE 和 MAE 均优于 Transformer，TCN，Informer 和 CEEMDAN-Informer，MSE 平均下降 17.8%，14.4%，10.9% 和 6.6%；MAE 平均下降 14.4%，11.3%，8.4% 和 6.5%。此外，在其余步长时，v-Informer 的评价指标也优于其他模型，可以看出 v-Informer 拥有最好的预测精度。

基于 50 个任务或虚拟机的预测性能评价指标，绘制了相应的累积分布图，图 8 给出了上述 6 种方法的评价指标在不同步长下的累积分布结果。



(a)Google 任务 CPU 负载 50 步预测值的 MSE　(b)Google 任务 CPU 负载 50 步预测值的 MAE　(c)Google 任务 CPU 负载 70 步预测值的 MSE　(d)Google 任务 CPU 负载 70 步预测值的 MAE

(e)微软虚拟机 CPU 负载 50 步预测值的 MSE　(f)微软虚拟机 CPU 负载 50 步预测值的 MAE　(g)微软虚拟机 CPU 负载 70 步预测值的 MSE　(h)微软虚拟机 CPU 负载 70 步预测值的 MAE

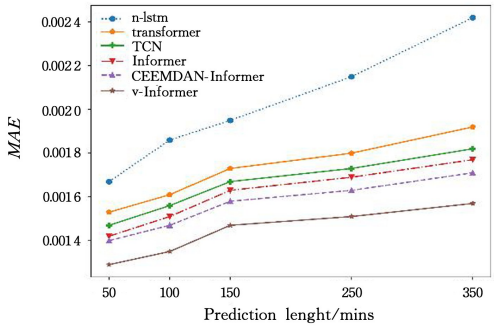
图 8 Google 和微软评价指标累积分布图

Fig. 8 Cumulative distribution of evaluation metrics for Google and Microsoft datasets

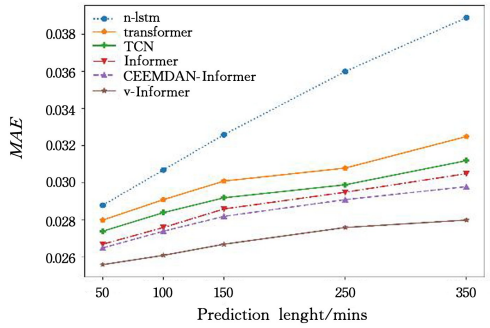
从图 8(a)、图 8(b)中可以看到，对于 google 任务的 CPU 负载，预测 50 步时，LSTM 只有 40% 的 MSE 低于 0.03，Transformer 有 80% 的 MSE 低于 0.03，TCN 有 90% 的 MSE 低于 0.03，Informer 有 95% 的 MSE 低于 0.03，CEEMDAN-Informer 和 v-Informer 的 MSE 均低于 0.03，但代表 v-Informer 模型的曲线始终位于 CEEMDAN-Informer 模型的上方。虽然 MAE 也是 v-Informer 最优，但领先幅度没有 MSE 大，这可能是由于模型在训练时以 MSE 为评价指标。从图 8(e)、图 8(f)中可以看到，对于微软虚拟机的负载，预测 50 步时，当 CDF 指数为 0.5，v-Informer，CEEMDAN-Informer，Informer，TCN，Transformer，n-LSTM 对应的横坐标分别为 0.0016，0.0017，0.00175，0.00177，0.0018，0.0023，这表明 v-Informer 所得的 MSE 集合中有 50% 的值小于 0.0016，CEEMDAN-Informer 有 50% 的值小于 0.0017，Informer 有 50% 的值小于 0.00175，TCN 有 50% 的值小于 0.00177，Transformer 有 50% 的值小于 0.0018，n-LSTM 有 50% 的值小于 0.0023。而且，v-Informer 的曲线始终位于其他 5 条曲线的上方。

在图 8(c)、图 8(d)、图 8(g)、图 8(h)中，v-Informer 的曲线同样位于其他 5 条曲线的上方，并且这种差距相较于图 8(a)、图 8(b)、图 8(e)、图 8(f)有一定的增大。图 9 给出了微软虚拟机数据预测准确率与预测时长的关系，发现 v-Informer模型预测误差的增长率保持在一个较低的水平，受累计误差的影响较小，并且随着预测时长的增加，v-Informer 的预测

误差的增长率是最小的。综合来看，v-Informer 在各个步长上均优于其他模型，这主要得益于 VMD 分解原始序列，获得更平滑稳定的子序列，分别学习子序列中的原始序列特征，降低了预测难度，同时 GC 加速了梯度收敛，提高了预测精度。



(a)预测时长与 MSE 的关系



(b)预测时长与 MAE 的关系

图 9 微软预测时长与预测准确率关系图

Fig. 9 Microsoft prediction time versus prediction accuracy

模型预测花费的时间开销也是评估方法的一个重要指标，在保证预测精度的前提下缩短预测时间，能够为云平台提供充足的反应和调度时间。表 8 列出了不同模型预测 30 步后结果所花费的时间，再结合表 6、表 7、表 9、表 10 可以看出，LSTM，Transformer 和 CEEMDAN-Informer 在预测时间和精度上的性能均弱于 g-Informer(g-Informer 是在 v-Informer 中去掉变分模态分解部分而得到的方法)。虽然 TCN 的预测时间要短于 g-Informer，但其预测精度更低。综合来看，g-Informer 优于 TCN。这表明以梯度集中为收敛规则的 Adam 优化器在预测时间和预测精度上均优于原始 Adam 优化器。

表 8 不同方法的执行时间对比

Table 8 Comparison of execution times of different methods

(ms)

数据集	n-LSTM	Transformer	TCN	Informer	CEEMDAN-Informer	g-Informer
谷歌	265102	213045	136012	185413	193225	156021
微软	221564	186453	115421	152130	158135	120312

表 9 基于 Google 数据集的消融实验结果

Table 9 Ablation experimental results on Google dataset

步长	Informer		g-Informer		v-Informer	
	MSE	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*
10	0.0151	0.0949	0.0145	0.0903	**0.0140**	**0.0868**
20	0.0158	0.1008	0.0154	0.0970	**0.0146**	**0.0926**
30	0.0169	0.1091	0.0159	0.1027	**0.0152**	**0.0947**
50	0.0182	0.1151	0.0170	0.1101	**0.0162**	**0.1032**
70	0.0193	0.1223	0.0181	0.1160	**0.0171**	**0.1094**

表 10 基于微软数据集的消融实验结果

Table 10 Ablation experimental results on Microsoft dataset

步长	Informer		g-Informer		v-Informer	
	MSE	*MAE*	*MSE*	*MAE*	*MSE*	*MAE*
10	0.00142	0.02671	0.00137	0.02637	**0.00129**	**0.02563**
20	0.00151	0.02769	0.00143	0.02712	**0.00135**	**0.02613**
30	0.00163	0.02866	0.00156	0.02789	**0.00147**	**0.02679**
50	0.00169	0.02953	0.00163	0.02908	**0.00152**	**0.02762**
70	0.00177	0.03056	0.00171	0.02955	**0.00158**	**0.02809**

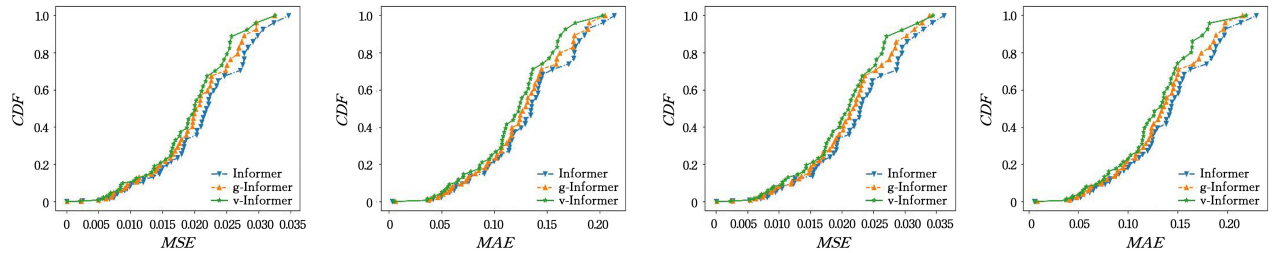
5.4 消融实验结果分析

这一部分比较了 Informer，g-Informer 和 v-Informer 在 Google 和微软数据集 50 个任务或虚拟机上的预测结果的平均值，结果如表 9 和表 10 所列，可以看出 VMD 和 GC 对负载预测的精度提升均有帮助。

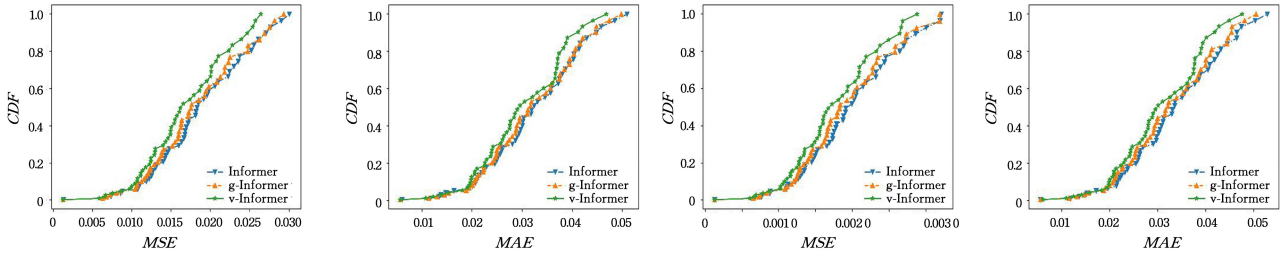
对于 Google 任务负载，步长为 50 时，g-Informer 相较于 Informer 的 MSE 和 MAE 分别减少了 6.6%，4.3%；v-Informer 相较于 g-Informer 的 MSE 和 MAE 分别减少了 4.7%；6.4%。对于微软虚拟机负载，步长为 50 时，g-Informer 相较于 Informer 的 MSE 和 MAE 分别减少了 4.1%，1.7%；v-Informer 相较于 g-Informer 的 MSE 和 MAE 分别减少了 6.8%，6.4%，拥有最优的预测精度。

根据 50 个任务或虚拟机的预测性能评价指标，绘制出相应的在不同步长下的累积分布图。图 10(a)、图 10(b)、图 10(e)、图 10(f)分别描述了步长为 50 时的误差累积分布，从图中可以看出，v-Informer 的曲线一直位于其他两条曲线的上方，表明 v-Informer 的 MSE 和 MAE 都优于其余两种方法。同时，g-Informer 的曲线除了 CDF 值在[0.8，1.0]的区间内时与代表 Informer 的曲线有所重叠外，在其余区间时均位于 Informer 之上。

图 10(c)、图 10(d)、图 10(g)、图 10(h)描述了步长为 70 时的误差累积分布，类似地，v-Informer 均位于其他曲线的上方，g-Informer 除了与 Informer 有小部分重叠外，其余均位于 Informer 上方。这说明变分模态分解和梯度集中对模型的预测精度均有显著的提升效果。获得这样的提升的原因在于，经过 VMD 分解后的子序列具有平稳性，分解了趋势变化，从而降低了预测难度；同时，通过梯度集中正则化权重空间和特征空间，提高了约束损失函数的利普西茨性，从而提高模型的泛化性和精确度。



(a)Google 任务 CPU 负载 50 步预测值消融实验的 MSE　(b)Google 任务 CPU 负载 50 步预测值消融实验的 MAE　(c)Google 任务 CPU 负载 70 步预测值消融实验的 MSE　(d)Google 任务 CPU 负载 70 步预测值消融实验的 MAE



(e)微软虚拟机 CPU 负载 50 步预测值消融实验的 MSE　(f)微软虚拟机 CPU 负载 50 步预测值消融实验的 MAE　(g)微软虚拟机 CPU 负载 70 步预测值消融实验的 MSE　(h)微软虚拟机 CPU 负载 70 步预测值消融实验的 MAE

图 10　Google 和微软数据集消融实验评价指标累积分布图

Fig. 10　Cumulative distribution of evaluation metrics for ablation experiments on Google and Microsoft datasets

结束语　本文研究了云平台任务资源负载信息的长期预测，提出了基于变分模态分解的编码器-解码器预测模型 v-Informer，通过变分模态分解得到不同中心频率的子序列，应用多头稀疏自注意力、注意力蒸馏以及生成式解码来根据 CPU 长期负载信息进行多步预测，最后将子序列的预测结果相加得到最终预测值。本文方案充分学习了历史数据在时间上的依赖关系，缓解了误差累积的现象。在 Google 云平台和微软云平台真实数据集上的实验结果表明，本文提出的 v-Informer 可以有效增强云平台 CPU 负载多步预测方面的能力。但是，某些云平台的负载信息没有固定的时间间隔，需要考虑时间间隔的变化。后续的工作将研究不固定的时间间隔下 CPU 负载序列的预测。

参 考 文 献

[1] MASDARI M，VALIKARDAN S，SHAHI Z，et al. Towards workflow scheduling in cloud computing：a comprehensive analysis[J]. Journal of Network and Computer Applications，2016，66：64-82.

[2] MASDARI M，NABAVI S S，AHMADI V. An overview of virtual machine placement schemes in cloud computing[J]. Journal of Network and Computer Applications，2016，66：106-127.

[3] REISS C，WILKES J，HELLERSTEIN J L. Google cluster-usage traces：format＋ schema[J]. Google Inc.，White Paper，2011，1：1-14.

[4] MASDARI M，SALEHI F，JALALI M，et al.：A survey of PSO-based scheduling algorithms in cloud computing[J]. Journal of Network and Systems Management，2017，25(1)：122-158.

[5] SINGH S，CHANA I. A survey on resource scheduling in cloud computing：issues and challenges[J]. Journal of Grid Computing，2016，14(2)：217-264.

[6] MASDARI M，KHOSHNEVIS A. A survey and classification of the workload forecasting methods in cloud computing[J]. Cluster Computing，2020，23(4)：2399-2424.

[7] KUMAR J，SINGH A K，BUYYA R. Self directed learning based workload forecasting model for cloud resource management[J]. Information Sciences，2021，543：345-366.

[8] OUHAME S，HADI Y，ULLAH A. An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model[J]. Neural Computing and Applications，2021，33：10043-10055.

[9] ZHOU S，LI J，ZHANG K，et al. An accurate ensemble forecasting approach for highly dynamic cloud workload with VMD and R-transformer[J]. IEEE Access，2020，8：115992-116003.

[10] HU Y，DENG B，PENG F，et al. Workload prediction for cloud computing elasticity mechanism[C]// 2016 IEEE International Conference on Cloud Computing and Big Data Analysis(ICCCBDA). IEEE，2016：244-249.

[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，2018，48：4072-4083.

[12] PENG H，WEN W S，TSENG M L，et al. A cloud load forecasting model with nonlinear changes using whale optimization algorithm hybrid strategy[J]. Soft Computing，2021，25(15)：10205-10220.

[13] KUMAR J，SINGH A K. Workload prediction in cloud using artificial neural network and adaptive differential evolution[J]. Future Generation Computer Systems，2018，81：41-52.

[14] DUGGAN M，MASON K，DUGGAN J，et al. Predicting host CPU utilization in cloud computing using recurrent neural net-

works[C]// 2017 12th International Conference for Internet Technology and Secured Transactions(ICITST). IEEE, 2017: 67-72.

[15] GUO W, GE W, LU X, et al. Short-term load forecasting of virtual machines based on improved neural network[J]. IEEE Access, 2019, 7: 121037-121045.

[16] OUHAME S, HADI Y, ULLAH A. An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model[J]. Neural Computing and Applications, 2021, 33(16): 10043-10055.

[17] DU S, LI T, YANG Y, et al. Multivariate time series forecasting via attention-based encoder-decoder framework[J]. Neurocomputing, 2020, 388: 269-279.

[18] YU Z H, LIU Z Q, LIU Y, et al. Software defect prediction feature selection method based on adaptive mixed particle swarm optimization[J]. Computer Applications, 2023, 43(4): 1206-1213.

[19] LI Y H, GUO H G, LIU P P, et al. Load prediction method of cloud platform based on temporal convolutional network [J]. Computer Science, 2023, 50(7): 254-260.

[20] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2021: 11106-11115.

[21] YONG H, HUANG J, HUA X, et al. Gradient centralization: A new optimization technique for deep neural networks[C]// European Conference on Computer Vision. Cham: Springer, 2020: 635-652.

[22] SONG B, YU Y, ZHOU Y, et al. Host load prediction with long short-term memory in cloud computing[J]. The Journal of Supercomputing, 2018, 74: 6554-6568.

[23] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.

[24] BAI S, KOLTER J Z, KOLTUN V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv: 1803. 01271, 2018.

[25] WU T, PAN M, YU Y. A Long-term Cloud Workload Prediction Framework for Reserved Resource Allocation[C]// 2022 IEEE International Conference on Services Computing(SCC). IEEE, 2022: 134-139.

YOU Wenlong, born in 1999, postgraduate. His main research interests include cloud computing and artificial intelligence.

DENG Li, born in 1972, Ph.D, associate professor, is a member of CCF (No. 57882M). Her main research interests include cloud computing and distributed computing.

(责任编辑:何杨)