doi:10.3772/j.issn.1002-0470.2020.09.002

基于卷积神经网络的虚拟机多类型负载联合预测方法①

余 显②[*,**] 李振宇[*] 张广兴[*] 谢高岗[**,***]

([*]中国科学院计算技术研究所 北京 100190)

([**]中国科学院大学 北京 100190)

([***]中国科学院计算机网络信息中心 北京 100190)

摘 要 虚拟机(VM)负载预测对提高云数据中心的资源利用率及用户服务质量起着至关重要的作用。然而现有的预测方法通常只考虑单一负载类型,在真实的云环境中,要么难以保障预测精度,要么因为需要同时建立多个预测模型而产生庞大的训练和预测时间开销。针对现有预测方法无法有效兼顾多种类型负载场景下预测精度和时间开销的问题,提出了一种基于卷积神经网络(CNN)的多类型负载联合预测方法(TSF),能自动化构建并提取关键训练样本,并充分挖掘其中潜在的时序特征和空间特征,从而在考虑多种虚拟机负载情况下,能有效降低训练和预测时间成本,同时提高预测精度。

关键词 云数据中心;虚拟机(VM);多类型负载联合预测(TSF);卷积神经网络(CNN);局部特征增强

0 引 言

近年来,云数据中心已经被广泛部署并应用于各个领域。通过“现收现付制”的服务模式,云数据中心能够给每个租户提供非常弹性的计算、存储、网络等资源。每一个应用或者服务都以虚拟机(virtual machine, VM)的存在形式来共享这些资源。但是随着云计算业务的繁荣和用户需求的井喷式增长,数据中心和虚拟机的规模也与日俱增。与此同时,如何积极有效地提高主机的资源利用率、保障每一个租户的服务质量成为了运营商不可避免的重大挑战[1]。

为了达到这些目标,目前的解决办法是通过实时预测虚拟机未来负载变化情况,然后据此动态完成虚拟机资源的动态迁移和调度[2,3]。其中虚拟机的负载是指虚拟机各种类型资源的占用情况,包括中央处理单元(CPU)、内存、存储、网络资源等。典型负载预测的做法要求为每一种类型的负载建立单独的预测模型。负载种类越多,需要训练的模型数量也就越多,这就意味着成倍的存储开销和训练时间开销,极大地限制了预测模型在实时系统中的可用性。此外,现有模型通常只学习单一维度负载时序上的潜在模式,而忽略了不同类型负载之间的空间相关关系,进而一定程度上降低了模型的预测精度。

针对上述问题,本文提出一种结合负载时间和空间特征的多负载联合预测方法(multi-type load joint forecasting,TSF),来同时训练并预测多种类型负载。为了保障模型的预测精度,TSF 除了按照传统的预测方法学习负载数据的时序特征(如周期性、自相似性)以外,还将不同负载之间线性和非线性依赖等空间特征纳入学习范畴,并引入卷积神经网络(convolution neural network,CNN)[4]来挖掘这

① 国家重点研发计划(2018YFB1800201)和国家自然科学基金(61802366)资助项目。

② 男,1992 年生,博士生;研究方向:云网络,CDN 缓存;联系人,E-mail:yuxian@ict.ac.cn

(收稿日期:2019-10-10)

些潜在的空间特征。其主要挑战在于不同时间段下这种空间特征会表现出比较大的强弱差异,而这种差异性会阻碍模型的有效训练,导致预测精度的降低。为了解决这一问题,首先引入最大化信息系数(maximal information coefficient,MIC)来量化空间特征,然后提出一种局部空间特征增强算法来提取具有强空间特征的训练样本。这样不仅能够大幅度降低负载数据在空间特征上表现的差异性,还能最大程度保留其潜在的关键知识,从而能够有效保障模型的预测精度。最后,提出了一种双模型预测机制,通过集成原始模型和特征增强后模型来共同训练和挖掘负载模式,以此得到更为准确的预测模型。基于2份真实数据集和1份人造数据集,对TSF的性能展开了充分的实验验证。实验结果表明,TSF不仅能够在考虑多种类型负载时保持较低的训练和预测时间开销,同时在预测精度方面相比其他方法也能有明显提升。

本文共分为4节,其中第1节为相关工作,主要介绍常用的预测方法;第2节首先介绍虚拟机负载时间和空间特征的基本概念,然后提出多负载联合预测方法(TSF);第3节介绍实验数据和环境配置,并给出实验结果;第4节对本文主要成果和未来工作进行总结和说明。

1　相关工作

现有的虚拟机负载预测方法通常可分为线性预测方法和非线性预测方法。

线性预测方法通常假设数据具有一定的线性特征。最典型的线性预测方法包括差分整合移动平均自回归模型(autoregressive integrated moving average model,ARIMA)[5]、霍尔特-温特(Holt-Winters,HW)模型[6]和线性回归模型(linear regression,LR)以及其相关的衍生模型[7]。这些模型因为其简单性常被作为基础预测模型进行广泛应用。然而,在面对实际的云数据中心环境时,虚拟机负载的这种线性假设难以成立,从而很难保证预测模型的精度。

为了提高模型的预测精度,很多方法进一步考虑学习数据的非线性特征。文献[8]提出了多个维度安全测量指标,然后通过支持向量回归模型(support vector regression,SVR)来对动车组设备进行安全评估。文献[9]则是利用人工神经网络模型(artificial neural network,ANN)来捕获网络流量的时序特征,进而完成相关预测。文献[10]分析了视频流码率的行为变化,并通过隐马尔可夫模型(hidden markov model,HMM)来对码率选择问题进行建模。文献[11]提出利用残差长短记忆网络(long short-term memory,LSTM)来实现短期交通流量数据的预测和自适应建模分析。文献[12]进一步提出了一种双向的长短时神经网络模型(bi-directional LSTM,BiLSTM),来更好地挖掘语音时序数据的前后关系,其后也被广泛应用于主题建模[13]等其他领域。然而,这些方法通常只考虑对单一维度的时序数据进行建模,在实际云系统中,必须考虑多种不同类型的虚拟机负载。此时,这些方法需要针对每一种负载训练单独的预测模型,从而极大地增加了系统的训练和预测负担,无法满足在线系统实时性预测和动态更新的需求。不同于这些方法,TSF是为多种类型负载构建一个统一的预测模型,能够有效避免因负载类型增加而导致时间开销成倍增长的问题。

此外,针对全网环境,文献[14]提出了基于动态图卷积神经网络的模型来完成交通环境下未来车流量的预测。文献[15]则考虑部分测量数据丢失的场景,并利用矩阵填充技术来预测得到这些丢失数据。然而,在真实云计算环境中,由于受到调度策略的影响,每个虚拟机所在主机位置并不固定,这就导致这些方法[14-15]难以通过流量矩阵的形式来学习虚拟机彼此的空间结构关系。相反,TSF仅考虑单个虚拟机的负载预测,并不依赖其他虚拟机的状态和位置信息,从而能够轻易在不同虚拟机上进行扩展。

2　多类型负载联合预测方法

本节首先介绍并分析虚拟机负载的时序特征和空间特征。有效地学习这些特征是TSF能够精准预测的基础。然后详细介绍TSF具体的工作原理,其主要包含如下几部分:特征单元-标签对构造,这

些特征单元-标签对表示 TSF 预测模型训练和预测的输入-输出对实例(即训练样本),本文期望通过从训练样本中挖掘出负载数据潜在的时序和空间特征;局部空间特征增强算法,用来自动化选择高信息量的训练样本,以提高预测模型的学习效率;基于 CNN 的预测模型框架以及双模型训练过程。

2.1　时序特征和空间特征分析

虚拟机负载时序特征是指在时间维度上数据采样点之间存在的相关性依赖,以及不同时间段之间潜在的周期性、自相似性等。这也是现有预测方法的实现基础。然而,当考虑多种类型负载时,单一维度的时序特征不足以保证预测模型的精确度。为此,本文提出虚拟机负载空间特征的概念。定义空间特征为不同类型负载之间的线性或非线性相关关系。期望预测模型不仅能够学习单一维度负载的时序特征,还能通过挖掘不同类型负载间的空间特征来提高预测精度(具体学习方法详见第 2.3 节)。

考虑到最大化信息系数(MIC)不仅能够反映 2 个变量之间的线性关系,还能表示其间的非线性关系,并且在表示依赖关系强度方面具有较高的鲁棒性和较低的计算复杂度,因此使用该指标来量化以及分析不同负载之间的线性和非线性依赖关系。MIC 值的具体计算公式如下:

$$MIC(X, Y) = \max_{|X||Y|<B} \left(\max_{\forall Grids} \left(\sum_{x\in X, y\in Y} P(x, y) \log \frac{P(x, y)}{\sum_{x\in X} P(x, y) \sum_{y\in Y} P(x, y)} \right) \right) \Big/ \log \min\{|X|, |Y|\} \tag{1}$$

其中,X、Y 分别表示需要研究的 2 个不同时序序列对应的随机变量,B 表示以 X 和 Y 为坐标系分割的方格总数。

以 228 个负载样本作为窗口大小,计算 2 个真实虚拟机 CPU 利用率和内存利用率采样的时序数据在该窗口大小下所有连续时间片段对之间的 MIC 值(该虚拟机负载信息如第 3 节实验评估部分 WS-1 和 WS-2 所述)。图 1 给出了对应所有得到的 MIC 值的概率密度函数(PDF)和累计概率密度函数(CDF)关系。从图中可以看出,CPU 和内存之间存在明显的相关关系,通过学习这种依赖关系有助于提高预测精度(在第 2.3 节介绍如何学习这种关系)。此外,不同时间片段的 CPU 和内存利用率之间相关关系有所不同。这也说明不同虚拟机负载之间的相关性并非固定不变,而是会受到业务负载大小变化以及业务类型变化的影响而发生改变。这种依赖关系的变化或者差异对预测模型是否能有效挖

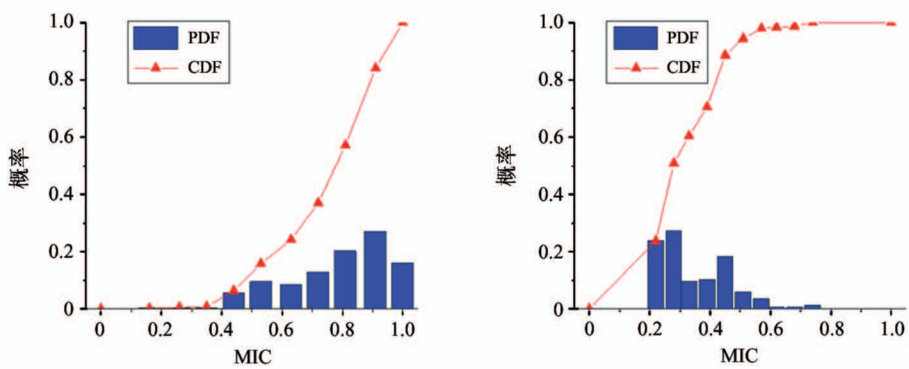


图 1　2 个真实虚拟机 CPU 和内存负载所有不同时间片段对的 MIC 值对应的 PDF 和 CDF

掘虚拟机负载的空间特征、提高精确度是一个巨大的挑战。在第2.4节中详细介绍如何处理这一问题。

2.2　基于滑动窗口的特征单元-标签对构造

为了挖掘负载的潜在模式,采用监督学习方式来训练预测模型。在此之前,需要构造训练所需要的特征单元和标签信息(二者共同组成训练样本)。在此问题中,特征单元即代表虚拟机历史负载数据,标签代表需要预测的未来时刻虚拟机负载。给定采样得到的虚拟机历史负载数据,现有的基于监督学习的时序预测方法在构造特征-标签对时通常只考

虑一种类型负载(如 CPU 负载),无法直接有效地应用到多类型负载场景,故本文提出了一种新的基于滑动窗口的特征单元-标签对构造方法。

具体的构造流程如图 2 所示。首先对虚拟机的负载进行周期性采样得到所有类型负载的时序数据。假定 D 表示负载种类数目,每种类型负载的总采样数为 T,并且第 i 种类型负载的所有数据集合用 $\boldsymbol{R}^i$ 表示如下:

$$\boldsymbol{R}^i = [r_1^i, r_2^i, \cdots, r_t^i, r_{t+1}^i, \cdots, r_T^i] \tag{2}$$

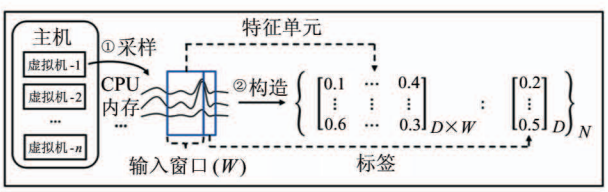
其中 r_t^i 表示第 i 种负载在 t 时刻的值大小(如可表示 CPU 在 t 时刻的利用率为 20%)。以 t 时刻为起始时刻来说明每一回合如何得到训练样本,其中用窗口大小 W 来表示预测时所使用的每种负载历史数据量。那么可以得到输入窗口或特征单元 $\boldsymbol{F}$ 形式如下:

$$\boldsymbol{F} = \begin{bmatrix} r_t^1 & r_{t+1}^1 & \cdots & r_{t+W}^1 \\ r_t^2 & r_{t+1}^2 & \cdots & r_{t+W}^2 \\ \vdots & \vdots & \vdots & \vdots \\ r_t^D & r_{t+1}^D & \cdots & r_{t+W}^D \end{bmatrix} \tag{3}$$

本文只考虑单步长预测问题,故 $\boldsymbol{F}$ 对应的标签 $\boldsymbol{L}$ 可表示为

$$\boldsymbol{L} = [r_{t+W+1}^1, r_{t+W+1}^2, \cdots, r_{t+W+1}^D]^{\mathrm{T}} \tag{4}$$

每个训练样本 s 即可表示为($\boldsymbol{F}$,$\boldsymbol{L}$),其中 t 的取值范围可写成[1, $T-W-1$]。通过改变 t 的取值来改变输入窗口在原始数据中的位置,可以进一步得到其他的训练样本。依此类推,共可以得到 $T-W-1$ 组训练样本,并用集合 S 表示。S 将被直接用于预测模型的训练。



其中,W 表示输入窗口的大小,即每次输入的 CPU 或者内存样本的数量,D 表示采样的负载种类,N 表示根据采样数据构造得到的“特征单元-标签”对的数量。

图 2　特征单元-标签对构造过程

2.3　基于 CNN 的预测模型框架

卷积神经网络(CNN)[4] 得益于非常优秀的对空间结构的学习能力,目前已被广泛地应用于图像分类[4]、网络知识学习[13]、图像生成[16] 等多个领域。这种特性正好能被有效用于学习虚拟机负载的空间特征。此外,CNN 模型的训练过程以卷积计算为主。相对于 LSTM 等其他深度学习模型而言,CNN 更加完美地匹配图形处理单元(GPU)等硬件设备的加速特性,从而能更好地适用于对时延性能要求较高的线上系统。因此,采用 CNN 作为基础模型从训练样本中挖掘负载数据潜在的时序和空间特征。

本文所采用的 CNN 模型结构如图 3 所示。该结构共包含 1 个输入层、7 个卷积层、2 个全连接层和 1 个输出层。输入层负责接收特征单元,特征单元的窗口大小设置为 8;卷积层用于从特征单元中挖掘其所包含的时序和空间特征,其每一层的结构大小如图中所示。举例说明,卷积层 -1 的结构为[2, 2, 1, 16],表示第 1 个卷积层共拥有 16 个卷积核,每个核的大小为[2, 2, 1],其他卷积层可依次类比。全连接层负责将卷积层的输出转化为规定输出大小,其结构分别为[448, 128]和[128, 2]。输出层则对应最终的标准化输出,并与预测标签相对应。值得注意的是,此处确定的模型结构实际上是结合实验和经验所得。限于文章篇幅,本文仅在实验评估章节中讨论输入层窗口大小对预测模型性能的影响(详见第 3.3.2 小节)。

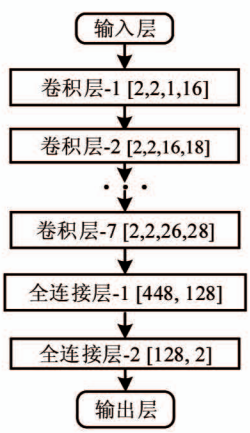


图 3　基于 CNN 的预测模型结构

2.4　局部空间特征增强算法

从第 2.1 节的分析中,可以知道不同时间段下

虚拟机负载的空间特征可能会有所差异，而这种差异会使得预测模型难以准确地学习出负载的潜在模式，从而降低预测精度。

进一步分析发现，这种空间特征差异具体体现在 MIC 值的大小上。而根据最大化信息系数的定义，可以知道 2 个变量之间的 MIC 值越接近 1，这些变量的相关关系越强。若给定特征单元 $\boldsymbol{F} = [r^1, r^2, \cdots, r^D]$，其中 r^i 表示 $\boldsymbol{F}$ 中第 i 种类型的负载所对应的采样数据。定义 $\boldsymbol{F}$ 对应的训练样本 s 所包含的空间特征 Φ 为任意 2 个不同维度负载间 MIC 的平均结果，其具体的计算公式如下：

$$\Phi(s) = \frac{\sum_{r^i, r^j \in F, i \neq j} MIC(r^i, r^j)}{D^2 - D} \tag{5}$$

于是，可以认为训练样本的 Φ 值越大，则其对应训练样本所包含的空间特征越强；反之，则越弱。基于这种想法，本文提出了一种局部空间特征增强算法（SFE）来提取关键的训练样本。即在 SFE 算法的作用下，具有弱空间特征的训练样本参与训练的数量将会减少，从而能进一步加强模型对负载数据关键知识的捕获能力。

SFE 算法详细过程如算法 1 所示。该算法给定一定数量训练样本集合 S 作为输入，以及采样阈值 Δ（表示需要过滤的训练样本占总体的比），算法输出处理后保留的训练样本集合 S_new。具体而言，SFE 算法首先对每一个训练样本计算其空间特征（第 1 ~ 3 行）；然后按照空间特征大小，从 S 中提取前百分比 $1-\Delta$ 数量的训练样本，赋值给 S_new（第 4 行）；最后，对 S_new 进行重采样，采样数为 $\| S \| \cdot \Delta$，采样得到的训练样本会添加到 S_new 中，使 S_new 和 S 样本数保持一致（第 5 行）。本文将在第 3 节讨论 Δ 对预测精度的影响。

算法 1　局部空间特征增强算法（SFE）

输入： 训练样本集合（$\boldsymbol{S}$），样本数量 N（$N \in \mathbb{N}^+$）；采样阈值（Δ）；

输出： 新的训练样本集合（$\boldsymbol{S\_new}$）

1：**for** s in $\boldsymbol{S}$ **do**

2：　$\Phi(\boldsymbol{S}) = \frac{\sum_{1 \leqslant i, j \leqslant D, i \neq j} MIC(\boldsymbol{r}^i, \boldsymbol{r}^j)}{D^2 - D}$

3：**end for**

4：$\boldsymbol{S\_new} \leftarrow top\_mic\ (\boldsymbol{S}, 1-\Delta)$

5：$\boldsymbol{S\_new} \leftarrow over\_sampling\ (\boldsymbol{S\_new}, \mathrm{N} \cdot \Delta)$

6：Finished

进一步分析 SFE 算法的时间复杂度如下。计算所有训练样本空间特征的时间复杂度为 $O(D^2)$；借助堆的数据结构，从 S 中提取空间特征最大的前 $N \cdot (1-\Delta)$ 个训练样本最坏情况下的时间复杂度为 $O(N\lg(N \cdot (1-\Delta)))$；算法中从 S_new 重复采样的时间复杂度为 $O(N \cdot \Delta)$。考虑到实际情况中 D 通常远小于 N，并且 N 是一个正整数，因此 TSF 算法时间复杂度为 $O(N \cdot \lg N)$。值得注意的是，TSF 算法的主要时间消耗来自从 S 中提取空间特征较大的训练样本，因此可以进一步通过优化 Top-k 算法来降低其时间复杂度。

2.5　双模型训练结构

TSF 按照定义好的 CNN 模型结构，采用双模型训练机制来得到最终的预测模型。即 TSF 通过初始化构造的训练样本以及经 SFE 算法处理后的训练样本同时训练出 2 个不同的预测模型，再根据测试集上结果选择出精度更高的模型作为最终使用的预测模型，过程如图 4 所示。同时训练 2 个模型的原因在于 SFE 算法实际是过滤了部分弱空间特征的训练样本，因此可能会造成一定的信息损失。故 TSF 使用未经 SFE 算法处理的训练样本进行训练能有效防止因信息损失而造成的模型精度下降问题。CNN 模型详细计算过程可参考文献[4]。

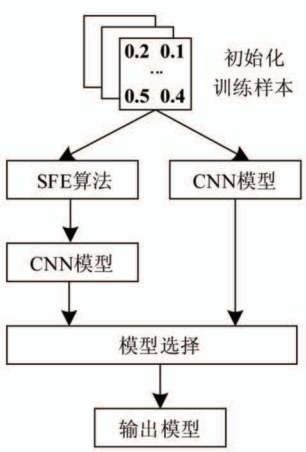


图 4　TSF 双模型训练结构

考虑到均方根误差(root mean square error, RMSE)已经被广泛应用于评估预测值和真实值之间的拟合程度,因此 TSF 采用 RMSE 作为损失函数,其计算方式如下:

$$RMSE = \sqrt{\frac{(P-R)^2}{M}} \tag{6}$$

其中,P 和 R 分别表示预测值和真实值,M 表示待预测值的数量。

3　实验评估

3.1　实验配置及说明

为了评估 TSF 性能,从真实数据中心 Bitbrains[17] 数据集中随机抽取了 2 个不同虚拟机 CPU 和内存利用率 3 周的历史负载数据(采样频率为 5 min)。用前 2 周数据来训练预测模型,用第 3 周数据进行测试。提取的虚拟机负载数据信息具体如表 1 中 WS-1 和 WS-2 所示。此外,人为构造了一份无相关关系的负载数据(WS-3)来进一步验证 TSF 方法的有效性。WS-3 中 CPU 负载表示的是周期为 2π、振幅为 0.5 的正弦曲线。

表 1　虚拟机负载数据集

名称	负载类型	平均	标准差	最大值
WS-1	CPU	0.3793	0.4316	1
	内存	0.1029	0.1255	0.9513
WS-2	CPU	0.4643	0.1030	0.8346
	内存	0.6306	0.2265	0.9435
WS-3	CPU	0.5	0.3534	1
	内存	0.5	0	0.5

本文共对比了如下几种不同的典型预测方法:ARIMA[5]、LR[7]、SVR[8]、BiLSTM[13]。需要注意的是,这些方法均单独对 CPU 和内存负载数据进行建模,并采用网格搜索的办法来寻找最佳参数,其中 BiLSTM 和 TSF 使用自适应矩估计(adaptive moment estimation, Adam)优化算法[18]来训练参数。采用所有类型负载上的平均 RMSE 作为预测误差来评估预测方法精度,其计算方法如下:

$$\overline{RMSE} = \frac{1}{D} \cdot \sum_{i=1}^{D} RMSE(load_i) \tag{7}$$

其中,$RMSE(load_i)$ 表示第 i 种负载的预测误差。所有实验都运行在曙光服务器 W580-G20 上。该服务器拥有 14 个 CPU 核,每个核的配置为 Intel(R) Xeon(R) CPU E5-2660 v4@2.00 GHz;其物理内存为 64 GB;运行过程中 BiLSTM 以及 TSF 均采用同型号 GPU-NVIDIA Corporation GK210GL [Tesla K80] 进行加速。为了保证结果的准确性,每种预测方法均运行 10 次。

3.2　性能分析

(1)预测精度。图 5 展示了所有方法在每个数据集下多次实验的所有类型负载平均预测误差的平均值。从图 5(a)中可以得到各方法误差为 TSF < BiLSTM < LR < SVR < ARIMA。其中 TSF 要明显优于其他预测方法,并且相对 BiLSTM 的误差下降了约 11.06%,这是因为 TSF 方法更好地捕获和学习到负载的时间和空间特征。考虑到其他对比方法都是单独对 CPU 或者内存负载进行预测,这也进一步证明多负载联合预测方案的优越性。

此外,在数据集 WS-2 上(如图 5(b)),可以看到 TSF、BiLSTM 以及 LR 之间表现出几乎一样的预测误差,这和 WS-1 上的性能水平明显不同。本文认为产生这一差异的主要原因在于 WS-2 中负载比 WS-1 的更加稳定,更符合线性特征,这也能从表 1 中 WS-1 中 CPU 负载方差远大于 WS- 2 中 CPU 负载方差可以看出。这样一来,BiLSTM 和 TSF 会因为模型本身的复杂性而导致在训练过程中容易发生过拟合问题。反之,LR 方法因为其简单性而不会出现类似问题,从而能够在 WS-2 上保持和 BiLSTM 和 TSF 相仿的预测性能。与此同时,可以看到在 WS-3 上 TSF 和 BiLSTM 的平均预测误差都近似等于 0.12%,但是 TSF 略优于 BiLSTM,表明 TSF 方法同样能够有效地应用于低相关性的负载联合预测场景。

(2)训练和预测时间。考虑到 ARIMA、LR 以及 SVR 方法较高的误差,在此只对比 BiLSTM 和 TSF 方法所需要的训练和预测时间。图 6 反映了 2 种方法训练时间和预测时间随负载类型数量的变化而变化的折线图。每个方法在保证模型正常收敛的前提下均迭代 500 次。每组实验重复运行 10 次,取每组

平均值进行记录。

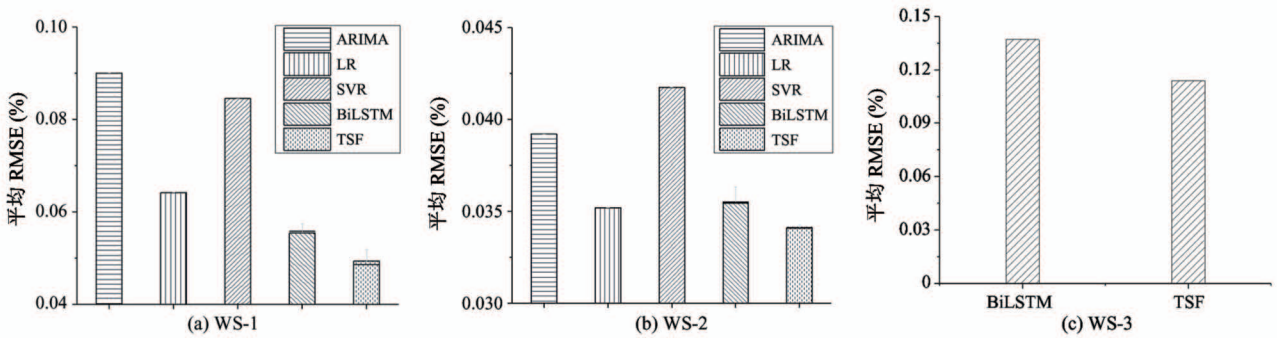


图 5　不同虚拟机负载数据集下的平均预测误差

从图 6(a)中可以看出,BiLSTM 的训练时间随负载类型数量的增加而线性增加,这是因为 BiLSTM 每次训练都只考虑一种类型负载。尽管 TSF 训练时间和负载类型数量也呈正比例关系,但是 TSF 联合了不同负载的空间特征,每次都是对所有类型负载进行训练,使得其增长速率远低于 BiLSTM 的方法。同理,图 6(b)反映了 TSF 在预测速度上要远快于 BiLSTM 方法。这也说明 TSF 方法能够更好地应用于线上系统。

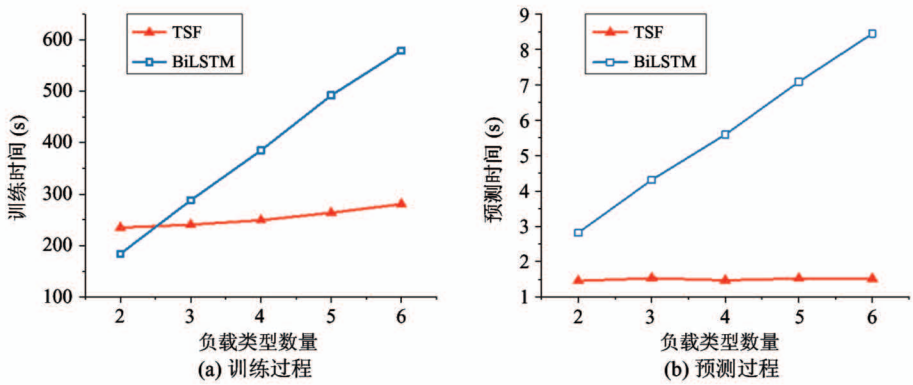


图 6　SF 训练和预测时间随负载类型数量的变化

3.3　参数评估

3.3.1　空间特征增强阈值

图 7 展示了在不同特征增强阈值 Δ 下 TSF 预测精度的平均-标准差图。如图 7 所示,WS-1 和 WS-2 上表现出 2 种完全不同的变化趋势,并且当 Δ 从 0 变到非 0 时,WS-1 上的预测误差降低,而 WS-2 上的预测误差则急剧增加。这表明 SFE 算法在 WS-2 上会起到负作用。这是因为 WS-2 负载波动小,线性特征较强(可从表 1 中得出),SFE 算法间接等同于去掉了部分训练样本,导致信息丢失。这也是 TSF 同时训练包含 SFE 算法和不包含 SFE 算法 2 种版本的预测模型的主要原因。此外,当 Δ 处于 0.05 ~ 0.2 之间时,TSF 平均预测误差变化非常小,这说明 Δ 参数的引入并不会增加额外的调参代价,比如可以直接在 TSF 训练之前令 Δ 为 0.1。

3.3.2　输入窗口大小

图 8 展示了 TSF 在图 3 所示固定隐藏层结构下,其预测精度随输入窗口大小变化而变化的平均-标准差图,每个盒中的横线表示每组实验的平均误差水平。如图所示,随着输入窗口的增大,TSF 的平均预测误差以及其波动也随之增大。造成这种现象的主要原因在于一定的模型隐藏层结构(包括卷积层和全连接层)限制了模型的学习能力,同时输入窗口越大表示其所包含的信息量也就越多,从而固定的隐藏层结构使得模型会随着输入窗口变大学习能力变弱。限于文章篇幅,本文不详细讨论隐藏层

结构对 TSF 预测性能的影响。本文所使用的窗口大小为 8。

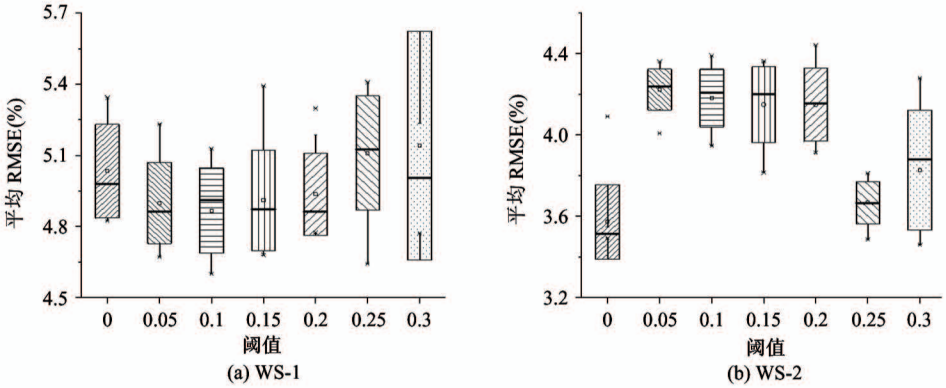


图 7　TSF 预测误差随 Δ 变化的平均值-标准差图,每个盒中的横线表示每组实验的平均误差水平

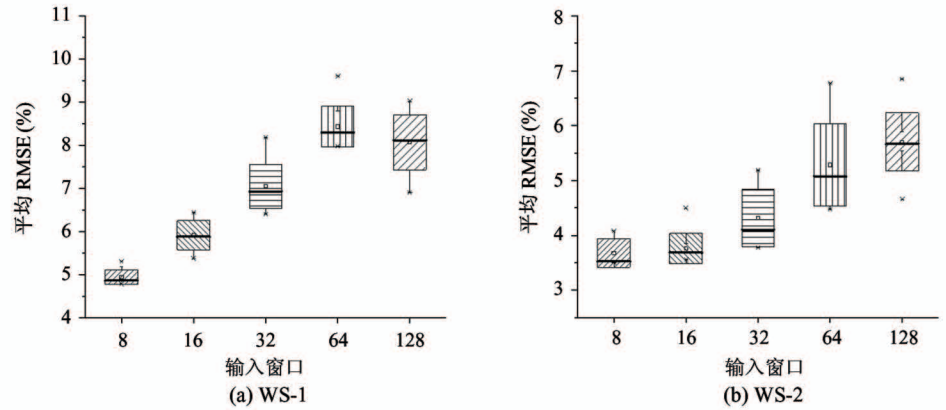


图 8　TSF 预测误差随输入窗口大小(W)变化的平均值-标准差图

4　结　论

本文旨在面向真实云数据中心环境,研究能够同时对多种虚拟机负载展开预测的方法,在控制训练和预测时间开销的同时,尽可能提高预测精度。为了实现这一目标,本文首先在现有方法对时序特征学习的基础上,引入不同负载间空间特征的概念,然后引入 CNN 模型来挖掘和学习虚拟机负载中这 2 种潜在特征;然后提出如何自动化构建和提取关键性训练样本的方法,并提出一种双模型的训练机制来共同提升预测模型的预测精度;最后基于公开的 Bitbrains 虚拟机负载数据集和人造数据集,对本文提出的预测方法进行了评估。实验结果充分证明了本文方法一方面能降低训练和预测的时间,同时又能达到更高的预测精度,从而能更好地适用于云资源调度系统。

未来工作中,将采用更多真实环境下虚拟机负载数据来评估 TSF 性能,同时进一步优化 TSF 方法在不同负载间空间特征的表示及学习能力。

参考文献

[1] Dempsey D, Kelliher F. Cloud Computing[M]. London: Palgrave Macmillan, 2018: 9-28

[2] Hieu N T, Di Francesco M, Ylä-Jääski A. Virtual machine consolidation with usage prediction for energy-efficient cloud data centers[C]//2015 IEEE 8th International Conference on Cloud Computing, Warsaw, Poland, 2015: 750-757

[3] Dabbagh M, Hamdaoui B, Guizani M, et al. Toward energy-efficient cloud computing: prediction, consolidation, and overcommitment[J]. *IEEE Network*, 2015, 29(2): 56-61

[4] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]// Proceedings of the 25th International Conference on Neural Information Processing Systems, New York, USA,

2012: 1097-1105

[5] Contreras J, Espinola R, Nogales F J, et al. ARIMA models to predict next-day electricity prices[J]. *IEEE Transactions on Power Systems*, 2003, 18(3): 1014-1020

[6] Chatfield C. The Holt-winters forecasting procedure[J]. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 1978, 27(3): 264-279

[7] Rao C R, Toutenburg H. Linear Models[M]. New York: Springer, 1995: 3-18

[8] 孙思齐, 马小宁, 薛蕊. 基于 SVR 的动车组设备安全评估方法研究[J]. 计算机仿真, 2019(5):179-183

[9] Cortez P, Rio M, Rocha M, et al. Multi-scale Internet traffic forecasting using neural networks and time series methods[J]. *Expert Systems*, 2012, 29(2): 143-155

[10] Sun Y, Yin X, Jiang J, et al. CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction[C]//Proceedings of the 2016 ACM SIGCOMM Conference, New York, USA, 2016: 272-285

[11] 李月龙,唐德华,姜桂圆,等. 基于维度加权的残差 LSTM 短期交通流量预测[J]. 计算机工程, 2019, 45(6): 1-5

[12] Graves A, Mohamed A R, Hinton G. Speech recognition with deep recurrent neural networks[C]//The 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, 2013: 6645-6649

[13] 彭敏, 杨绍雄, 朱佳晖. 基于双向 LSTM 语义强化的主题建模[J]. 中文信息学报, 2018, 32(4):40-49

[14] Diao Z L, Wang X, Zhang D F, et al. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting[C]//In the Association for the Advance of Artificial Intelligence (AAAI), Honolulu, USA, 2019: 890-897

[15] Xie K, Wang L, Wang X, et al. Accurate recovery of internet traffic data: a sequential tensor completion approach[J]. *IEEE/ACM Transactions on Networking*, 2018, 26(2): 793-806

[16] 朱俊鹏, 赵洪利, 杨海涛. 基于卷积神经网络的视差图生成技术[J]. 计算机应用, 2018, 38(1):255-259

[17] Shen S, van Beek V, Iosup A. Statistical characterization of business-critical workloads hosted in cloud datacenters [C]//2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Shenzhen, China, 2015: 465-474

[18] Kingma D P, Ba J. Adam: a method for stochastic optimization[J]. *arXiv*:1412.6980, 2014

Multi-type load joint forecasting method of virtual machine based on convolution neural network

Yu Xian[*,**], Li Zhenyu[*], Zhang Guangxing[*], Xie Gaogang[***]

([*] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**] University of Chinese Academy of Sciences, Beijing 100190)

([***] Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190)

Abstract

Virtual machine (VM) load prediction has always played a vital role in improving cloud data center resource utilization and user service quality. However, existing prediction approaches usually only consider a single type of VM load. In real cloud environment, these approaches are either hard to guarantee the prediction accuracy, or because of the need to establish multiple prediction models simultaneously, they will generate much training and prediction time overhead. Therefore, in view of the fact that the existing prediction approaches cannot effectively balance the prediction accuracy and time overhead in scenarios involving multiple types of loads, a multi-type load joint forecasting method (TSF) is proposed based on convolutional neural network (CNN). It automatically constructs and extracts the key training sample, and fully learns the potential temporal and spatial characteristics among them, such that it can effectively reduce training and prediction time consumption, and improves the prediction precision while taking into account the multiple types of VM loads.

Key words: cloud data center, virtual machine (VM), multi-type load joint forecasting (TSF), convolutional neural network (CNN), local spatial feature enhancement