

基于压缩动量项的增量型 ELM 虚拟机能耗预测

邹伟东¹ 夏元清¹

摘要 在基于基础设施即服务 (Infrastructure as a service, IaaS) 的云服务模式下, 精准的虚拟机能耗预测, 对于在众多物理服务器之间进行虚拟机调度策略的制定具有十分重要的意义. 针对基于传统的增量型极限学习机 (Incremental extreme learning machine, I-ELM) 的预测模型存在许多降低虚拟机能耗预测准确性和效率的冗余节点, 在现有 I-ELM 模型中加入压缩动量项将网络训练误差反馈到隐含层的输出中使预测结果更逼近输出样本, 能够减少 I-ELM 的冗余隐含层节点, 从而加快 I-ELM 的网络收敛速度, 提高 I-ELM 的泛化性能.

关键词 虚拟机能耗预测, 增量型极限学习机, 压缩动量项, 网络训练误差

引用格式 邹伟东, 夏元清. 基于压缩动量项的增量型 ELM 虚拟机能耗预测. 自动化学报, 2019, 45(7): 1290–1297

DOI 10.16383/j.aas.c180703

Virtual Machine Power Prediction Using Incremental Extreme Learning Machine Based on Compression Driving Amount

ZOU Wei-Dong¹ XIA Yuan-Qing¹

Abstract In cloud service models which is based on infrastructure as a service (IaaS), how to accurately predict power of virtual machine is very important for making scheduling strategy of virtual machines among many physical servers. However, the traditional incremental extreme learning machine (I-ELM) includes too many redundant hidden nodes, resulting in decreased efficiency and accuracy of virtual machine power prediction. Connecting compression driving amount to I-ELM, the paper builds the intelligent prediction model of I-ELM based on the compression driving amount (CDAI-ELM), and uses the model for predicting virtual machine power.

Key words Power prediction of virtual machine, incremental extreme learning machine (I-ELM), compression driving amount, network training error

Citation Zou Wei-Dong, Xia Yuan-Qing. Virtual machine power prediction using incremental extreme learning machine based on compression driving amount. *Acta Automatica Sinica*, 2019, 45(7): 1290–1297

随着计算机网络、物联网和智能技术的发展, 科技社会获得的数据呈爆炸式增长. 数据量巨大性给数据的处理和转移应用分析带来了诸多挑战. 云计算能按需分配计算资源, 是一种很好的大数据解决方案, 同时云计算是推动信息技术能力实现按需供给、促进信息技术和数据资源充分利用的全新业态, 是信息化发展的重大变革和必然趋势^[1-2]. 虚拟化技术是云计算的关键使能技术, 利用虚拟化技术将大量的物理服务器资源转换为可灵活按需分配的虚拟资源. 精准的估计与预测虚拟机能耗不但可以优

化云计算计费方式, 而且可改善虚拟机迁移调度策略, 提高云计算能效效率, 进一步降低云服务提供商的能耗花费^[3]. 云服务提供商可根据准确的虚拟机能耗预测设计高效的定价策略与计费方法, 为消费者提供更多的优惠措施, 避免了因定价高而致使消费者流失, 促使云计算的服务模式获得长足的发展.

虚拟机能耗模型具有非线性、强耦合和时变性的特征, 物理服务器的负载、CPU 利用率和内存利用率都会对虚拟机能耗预测产生重要影响, 因此, 它的预测模型难以通过机理法用简单的数学公式或传递函数进行描述^[4-5]. 当前伴随人工智能技术的快速发展, 研究人员利用人工智能技术解决虚拟机能耗预测的相关问题, 其中基于神经网络的机器学习方法在虚拟机能耗预测中的应用尤其突出. 神经网络能加快计算速度, 可利用有限的参数描述较为复杂的系统. 神经网络建模相对于机理建模方法的主要优点是无需具体的数学公式^[6], 更适用于长期预测, 它能更灵活地获取虚拟机能耗参数和非线性特性, 使得神经网络对虚拟机能耗建模更具有吸引力.

收稿日期 2018-11-05 录用日期 2019-03-08
Manuscript received November 5, 2018; accepted March 8, 2019
国家重点研发计划 (2018YFB1003700), 国家自然科学基金 (61836001), 中国博士后科学基金 (2018M641217) 资助
Supported by National Key Research and Development Program of China (2018YFB1003700), National Natural Science Foundation of China (61836001), and China Postdoctoral Science Foundation (2018M641217)

本文责任编辑 程龙
Recommended by Associate Editor CHENG Long

1. 北京理工大学自动化学院 北京 100081
1. School of Automation, Beijing Institute of Technology, Beijing 100081

鉴于神经网络在建模中的优势, 其在虚拟机能耗建模中的应用越来越广泛.

唐轶轩等^[7] 首先利用支持向量机 (Support vector machine, SVM) 建立虚拟机能耗与内存利用率、CPU 利用率之间的模型, 在结合待部署虚拟机的资源需求的基础上, 利用所建立的模型进行虚拟机能耗预测, 降低了数据中心能耗. Xu 等^[8] 在深入研究多个虚拟机环境中物理服务器负载对虚拟机负载的影响分析基础上, 提出了基于 RBF (Radial basis function) 神经网络的虚拟机能耗预测模型, 根据实验结果可知, 基于 RBF 神经网络的虚拟机能耗预测模型的性能优于分段线性回归等预测模型. 赵雅倩^[9] 利用模糊神经网络构建虚拟机能耗预测模型, 可有效避免硬阈值分区间模拟所产生的误差, 同时可以更好模拟虚拟机能耗与各相关参数之间的非线性关系. 为了有效地降低云资源浪费和提高云计算系统利用率, 贾灵昊等^[10] 提出利用马尔科夫模型对虚拟机能耗进行建模. 但支持向量机 (SVM)、BP 神经网络、RBF 神经网络等传统神经网络需人为设定许多的神经网络训练参数, 以迭代的方式更新参数, 容易陷入局部优化, 导致网络训练误差较大.

近年来, 增量型极限学习机 (Incremental Extreme Learning Machine, I-ELM)^[11] 在机器学习中备受青睐, 其原因在于增量型极限学习机的算法步骤少, 不需要迭代更新一些神经网络参数, 学习速度非常快, 并且不会产生局部最优解, 具有良好的泛化性能, 有效地克服了传统神经网络所存在的诸多缺陷, 充分发挥了其在机器学习中的巨大优势, 也促进了人工智能的进一步发展^[12-13]. I-ELM 采用增量式算法自适应地选取隐含层节点, 在算法迭代过程中随机产生隐含层节点参数, 利用最小二乘法计算其所对应的输出权值. 尽管算法实现简单, 但是增量型极限学习机中存在输出权值较小的隐含层节点, 导致许多隐含层节点对最终网络输出起到的作用很小^[14]. 大量冗余的隐含层节点只能增加网络结构的复杂性, 降低学习效率; 同时, 由于收敛速率较低, 增量型极限学习机通常需要更多的隐含层节点, 有时甚至超过学习样本的数量^[15-16], 因此研究高效的预测方法对虚拟机能耗预测具有十分重要的意义.

针对上述问题, 本文对增量型极限学习机进行改进, 通过计算每一个隐含层节点的网络输出误差, 构造合适的压缩动量项, 构建基于压缩动量项的增量型极限学习机模型, 实现对虚拟机能耗的精确预测, 能够有效提高预测精度和效率.

1 基于核函数的极限学习机

2012 年, 在深入研究支持向量机的基础上, Huang 等将核函数引入到极限学习机中 (Kernel

ELM, KELM)^[17], 获取最小平方优化解, 使得极限学习机具有更稳定的泛化性能. 该算法适用于当研究人员无法了解隐含层与输入层之间映射关系的时候, 研究人员仅需了解一个核函数便可构造一个极限学习机. KELM 的计算步骤可归纳为: 给定一个含有 N 个样本的训练集合 $\{(\mathbf{x}_i, \mathbf{t}_k) \mid \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_k \in \mathbf{R}^m, i, k = 1, 2, \dots, N\}$, 确定核函数 $K(\mathbf{x}_i, \mathbf{x}_j)$, 计算输出方程:

$$f(\mathbf{x}) = \mathbf{K} \left(\frac{\mathbf{I}}{C} + \Omega_{\text{ELM}} \right)^{-1} \mathbf{T} \quad (1)$$

$$\Omega_{\text{ELM}} = \mathbf{H}\mathbf{H}^T : \Omega_{\text{ELM},i,j} = h(\mathbf{x}_i) \cdot h(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \quad (2)$$

$$\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_1), K(\mathbf{x}_i, \mathbf{x}_2), \dots, K(\mathbf{x}_i, \mathbf{x}_N)]^T \quad (3)$$

式中, C 为正则化系数.

2 宽度学习系统

针对深度学习算法存在的训练时间长、难以收敛等问题, Chen 等提出了宽度学习系统 (Broad learning system, BLS)^[18]. BLS 方法利用稀疏自动编码器产生最优的输入权值矩阵, 对输入样本进行线性变换后形成特征节点, 再对特征节点经过激励函数非线性变换后获得增强节点. 通过合并增强节点输出与特征节点输出形成 BLS 的输出矩阵, 利用岭回归广义逆直接计算输出权值矩阵. BLS 的计算步骤可归纳为: 给定一个含有 N 个样本的训练集合 $\{(\mathbf{X}, \mathbf{Y}) \mid \mathbf{X} \in \mathbf{R}^{N \times M}, \mathbf{Y} \in \mathbf{R}^{N \times Q}\}$, 其中 \mathbf{X} 为输入样本, \mathbf{Y} 为输出样本, N 为样本个数, M 为每一个输入样本的特征维数, Q 为每一个输出样本的特征维数. 假设 BLS 网络结构中包含 s 个特征节点和 z 个增强节点, 则特征节点输出与增强节点输出分别表示为式 (4) 和式 (5):

$$\mathbf{G}^{N \times s} = \mathbf{X}^{N \times M} \cdot \mathbf{U}_e^{M \times s} \quad (4)$$

$$\mathbf{J}^{N \times z} = \psi(\mathbf{G}^{N \times s} \cdot \mathbf{U}_h^{s \times z} + v_h^{N \times z}) \quad (5)$$

$$\mathbf{A}^{N \times (s+z)} = [\mathbf{G}^{N \times s} \mid \mathbf{J}^{N \times z}] \quad (6)$$

式中, $\mathbf{U}_e^{M \times s}$ 为利用稀疏自动编码器产生最优的输入权值矩阵, $\mathbf{U}_h^{s \times z}$ 为特征节点到增强节点之间的随机输入权值矩阵, $v_h^{N \times z}$ 为增强节点的随机偏置, $\psi(\cdot)$ 为非线性激励函数. $\mathbf{A}^{N \times (s+z)}$ 表示特征节点和增强节点相连得到的合并矩阵, 即为宽度学习模型

的节点输出矩阵. 通过岭回归广义逆直接计算输出权值矩阵 $\mathbf{W}^{(s+z) \times N}$, 具体求解如下:

$$\mathbf{W}^{(s+z) \times N} = (\lambda \mathbf{I} + \mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}^T \mathbf{Y} \quad (7)$$

式中, λ 为正则化系数.

3 增量型极限学习机

增量型极限学习机通常是从一个规模比较小的神经网络开始, 根据误差大小向网络中逐个增加隐含层节点, 利用最小二乘法计算所增加的隐含层节点的输出权值, 进而计算网络训练误差, 直至达到最大隐含层节点数或者网络的期望误差时停止增加隐含层节点.

I-ELM 算法的迭代公式可表示为:

$$\mathbf{f}_n(\mathbf{x}) = \mathbf{f}_{n-1}(\mathbf{x}) + \beta_n \mathbf{g}_n(\mathbf{x}), \quad (8)$$

式中, $\mathbf{g}_n(\mathbf{x})$ 表示在第 n 步神经网络新增加的隐含层节点输出, 连接第 n 个新增隐含层节点与输出层的权值可表示为:

$$\beta_n = \frac{\langle \mathbf{e}_{n-1}, \mathbf{g}_n(\mathbf{x}) \rangle}{\|\mathbf{g}_n(\mathbf{x})\|^2}. \quad (9)$$

I-ELM 算法流程:

给定神经网络的训练样本 $\phi = (\mathbf{x}_i, \mathbf{t}_i)_{i=1}^N \in \mathbf{R}^m \times \mathbf{R}^n$, 隐含层节点的激励函数 $\mathbf{g} : \mathbf{R} \rightarrow \mathbf{R}$, 最大隐含层节点数 L_{\max} , 期望精度为 ϵ .

第 1 步, 神经网络初始化阶段: 令 $L = 0$, 初始网络误差为 $\mathbf{E} = \mathbf{T}$, 其中 $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_N]^T$;

第 2 步, 神经网络训练阶段:

当 $L < L_{\max}$ 且 $\|\mathbf{E}\| > \epsilon$ 时,

1) 增加一个隐含层节点: $L \leftarrow L + 1$;

2) 随机产生新增隐含层节点的输入权值和阈值 (\mathbf{a}_L, b_L) ;

3) 根据最小二乘法计算新增隐含层节点的输出权值: $\beta_L = \frac{\mathbf{E} \cdot \mathbf{g}_L^T}{\mathbf{g}_L \cdot \mathbf{g}_L^T}$;

4) 计算神经网络在增加了第 L 个隐含层节点后的误差: $\mathbf{E} \leftarrow \mathbf{E} - \beta_L \cdot \mathbf{g}_L$;

结束.

4 基于压缩动量项的增量型极限学习机

4.1 基于压缩动量项的增量型极限学习机的拓扑结构

如图 1 所示, 为了改进增量型极限学习机的收敛速度和泛化性能, 在原拓扑结构的的基础上引入压缩动量项, 优化网络结构, 提高网络学习效率.

图 1 中, \mathbf{x} 为输入变量, \mathbf{y} 为目标输出变量, \mathbf{a}_n 和 b_n 分别表示第 n 个隐含层节点的输入权值

和阈值, β_n 表示第 n 个隐含层节点的输出权值, $\mathbf{g}_n(\mathbf{a}_n, \mathbf{x}, b_n)$ 表示第 n 个加法型或 RBF 型隐含层节点的输出, \mathbf{f}_{n-1} 表示当前包含 $n-1$ 个隐含层节点的网络输出, $\mathbf{e}_{n-1} = \mathbf{y} - \mathbf{f}_{n-1}$ 表示当前包含 $n-1$ 个隐含层节点的网络训练误差, α_{n-1} 为第 n 个隐含层节点的压缩因子.

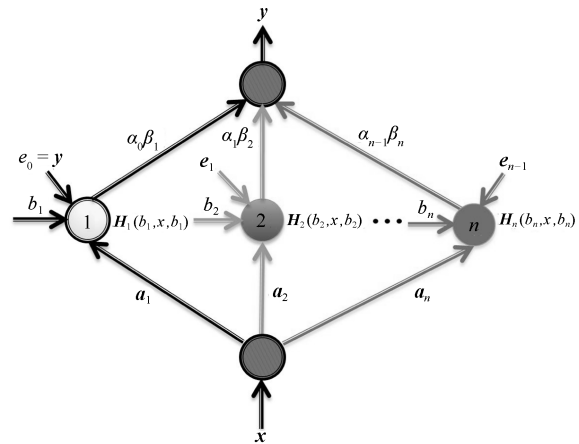


图 1 基于压缩动量项的增量型极限学习机拓扑结构图
Fig. 1 Topological structure of CDAI-ELM

4.2 基于压缩动量项的增量型极限学习机的收敛性分析和证明

定理 1. 若一个递增型单隐含层前馈神经网络, 以任意分段连续的函数作为隐含层激励函数, 随机产生隐含层节点输出矩阵 $\mathbf{g}_n(\mathbf{a}_n, \mathbf{x}, b_n)$, $n \in \mathbf{Z}^+$. λ_n 为常数且满足 $\lambda_1 = 0, 0 < \lambda_n < 1/2, n > 1$, $0 < B < 1$ 且满足 $\|\mathbf{g}_n(\mathbf{a}_n, \mathbf{x}, b_n) + \mathbf{e}_{n-1}\| < B$. 对于任意连续非常值的目标函数 \mathbf{f} , 有 $\|\mathbf{e}_n\| = \|\mathbf{f} - \mathbf{f}_n\| < B(1 + \sum_{i=1}^n \lambda_i^2)^{-\frac{1}{2}}$, 若压缩因子^[19] α_{n-1} 和输出权值 β_n 由如下公式计算获取:

$$\alpha_{n-1} = \frac{\|\mathbf{e}_{n-1}\|}{1 + \|\mathbf{e}_{n-1}\|^2} \quad (10)$$

$$\beta_n = \frac{\langle \mathbf{e}_{n-1}, \mathbf{g}_n + \mathbf{e}_{n-1} \rangle}{\alpha_{n-1} \|\mathbf{g}_n + \mathbf{e}_{n-1}\|^2} \quad (11)$$

式中, \mathbf{e}_n 为在训练中由于增加了第 n 个隐含层节点而引入的网络训练误差, 其表达了当网络中仅有 n 个隐含层节点时, 网络产生的输出 \mathbf{f}_n 与样本给定的理想输出之间的差值.

证明. 根据定理 1 的假设条件可知, 当 $n = 1$ 时, 可得:

$$\|\mathbf{e}_1\| = \|\mathbf{f} - \mathbf{f}_1\| < \|\mathbf{f}\| < \|\mathbf{f} + \mathbf{g}_1\| = \|\mathbf{e}_0 + \mathbf{g}_1\| < B \quad (12)$$

根据数学归纳法^[20] 可知, 假设 $\|\mathbf{e}_{n-1}\| = \|\mathbf{f} -$

$f_{n-1} \| < B(1 + \sum_{i=1}^n \lambda_i^2)^{-\frac{1}{2}}$ 成立, 则

$$\begin{aligned}
 \|e_n\|^2 &= \|e_{n-1} - \alpha_{n-1}\beta_n(g_n + e_{n-1})\|^2 = \\
 &\|e_{n-1}\|^2 + \alpha_{n-1}^2\beta_n^2\|g_n + e_{n-1}\|^2 - \\
 &2\alpha_{n-1}\beta_n\langle e_{n-1}, g_n + e_{n-1} \rangle = \\
 &\|e_{n-1}\|^2 + \frac{\langle e_{n-1}, g_n + e_{n-1} \rangle^2}{\|g_n + e_{n-1}\|^2} - \\
 &2\frac{\langle e_{n-1}, g_n + e_{n-1} \rangle^2}{\|g_n + e_{n-1}\|^2} = \\
 &\|e_{n-1}\|^2 - \frac{\langle e_{n-1}, g_n + e_{n-1} \rangle^2}{\|g_n + e_{n-1}\|^2} < \\
 &\|e_{n-1}\|^2 - \frac{\lambda_n^2\|e_{n-1}\|^4}{\|g_n + e_{n-1}\|^2} < \\
 &\|e_{n-1}\|^2 \left(1 - \frac{\lambda_n^2\|e_{n-1}\|^2}{B^2}\right) < \\
 &\|e_{n-1}\|^2 \left(1 + \frac{\lambda_n^2\|e_{n-1}\|^2}{B^2}\right)^{-1} = \\
 &\frac{1}{\frac{1}{\|e_{n-1}\|^2} + \frac{\lambda_n^2}{B^2}} < \\
 &\frac{1}{\frac{1}{B^2} \left(1 + \sum_{i=1}^{n-1} \lambda_i^2\right) + \frac{\lambda_n^2}{B^2}} = \\
 &B^2 \left(1 + \sum_{i=1}^n \lambda_i^2\right)^{-\frac{1}{2}} \quad (13)
 \end{aligned}$$

故可得 $\|e_n\| = \|f - f_n\| < B(1 + \sum_{i=1}^n \lambda_i^2)^{-\frac{1}{2}}$. □

4.3 基于压缩动量项的增量型极限学习机的学习步骤

如图 2 所示, 基于压缩动量项的增量型极限学习机与其他增量型极限学习机算法的区别在于在其模型和训练过程中引入压缩动量项, 进一步加快网络的收敛速度, 提高网络的泛化性能, 训练步骤如图 2 所示.

5 实验分析

5.1 基于压缩动量项的增量型极限学习机的收敛速度和泛化性能分析

通过 10 个回归问题对 CDAI-ELM、I-ELM、CI-ELM 和 EM-ELM 等 4 种算法从神经网络训练时间、隐含层节点数和测试精度等方面进行比较分析. 4 种算法均选择 Sigmoid 函数作为其隐含层节点的激励函数, 即 $g(a, b, x) = \frac{1}{1 + \exp(-(\mathbf{a} \cdot \mathbf{x} + b))}$.

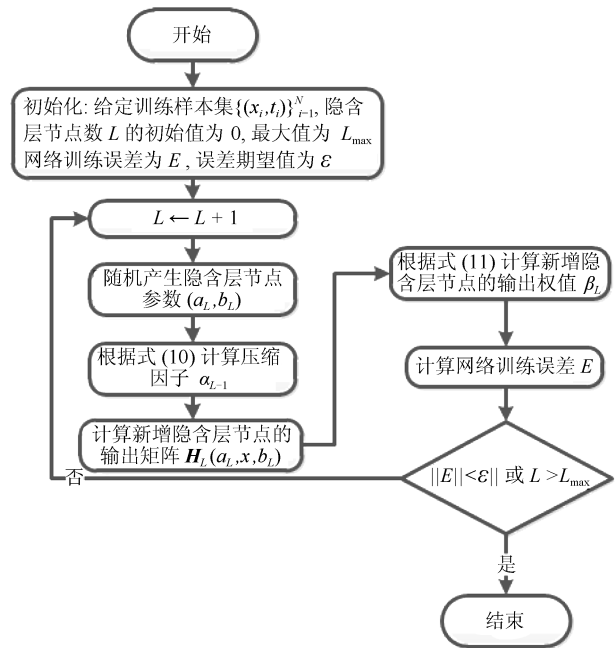


图 2 基于压缩动量项的增量型极限学习机算法流程图
Fig. 2 Flow diagrams of algorithm for CDAI-ELM

针对每一个回归问题, 每一种测试算法均取相同的期望精度, 所有测试算法中最大隐含层节点数为 300, EM-ELM 算法中初始隐含层节点数是 1, 且在迭代运算过程中每次仅增加 1 个隐含层节点. 所有回归问题的测试数据均来自于 KDD 数据库^[21], 相关测试数据的信息如表 1 所示. 所有测试实验均在 Matlab 2014a 环境中运行, 最后的测试结果取 200 次实验结果的平均值. 最好的实验结果均用粗体标注, 相近的实验结果均用下划线标明.

表 1 回归数据集
Table 1 Datasets of regression

回归数据集	属性	训练数据	测试数据
Auto MPG	4	853	850
Automobile	16	8 795	8 774
BlogFeedback	281	530	500
Housing	77	153	150
NoisyOffice	128	468	300
Facebook metrics	19	300	200
SML2010	68	336	200
wiki4HE	26	2 898	2 000
UJIIndoorLoc	529	2 100	2 077
YearPredictionMSD	90	2 800	3 075

在表 2 中, 在相同期望精度的情况下, 当训练精度提高时, 所需要的隐含层节点数也随之增多, 神经网络的泛化性能也随之提升, 但是当隐含层节点数到达一定程度后, 神经网络结构变得复杂, 其泛

化性能并未提升, 将当前的精度作为期望精度. 从表 2 可知, 对所有的回归问题, CDAI-ELM 算法所需要的隐含层节点数方差均为最小的, 但 I-ELM 算法所需要的隐含层节点数方差均为最多的. 在不同的回归问题中, I-ELM 算法所需要的隐含层节点数方差是 CDAI-ELM 算法的数十倍, CDAI-ELM 算法所需要的隐含层节点数方差均小于 CI-ELM 算法和 EM-ELM 算法, 在表 2 中由粗体标注, 这也说明了 CDAI-ELM 算法可获得更稳定且更紧凑的神经网络结构.

如表 3 所示, 在 10 个回归问题中, 除了 BlogFeedback 和 wiki4HE 两个例子中 EM-ELM 算法的测试误差比 CDAI-ELM 算法小, Housing、NoisyOffice、SML2010 和 Facebook metrics 4 个例子中 CDAI-ELM 算法与 EM-ELM 算法的测试误差比较接近外, 其余例子中 CDAI-ELM 算法的测试误差均为最小, 这也说明了 CDAI-ELM 算

法具有较好的泛化性能.

如表 4 所示, 在 10 个回归问题中, 除 NoisyOffice, wiki4HE 和 UJIIndoorLoc 三个例子中 CI-ELM 算法的训练时间比 CDAI-ELM 算法短, Housing 在 CDAI-ELM 算法与 CI-ELM 算法所用训练时间接近, 其余例子中 CDAI-ELM 算法的训练时间均为最小, 这也说明了 CDAI-ELM 算法可更快地确定最优的神经网络结构.

5.2 虚拟机能耗预测实验分析

本文使用一台戴尔 PowerEdge T640 塔式服务器进行实验仿真. 该服务器包含 2 个 Xeon Silver 4114 的标配 CPU, 64 GB 内存和 120 TB 硬盘空间, 服务器的操作系统为 CentOS 7.5. 利用 KVM 虚拟化技术在服务器上创建了具有相同配置的 5 台虚拟机, 具体配置为单核 Xeon Silver 4114 CPU, 8 GB 内存和 100 GB 硬盘空间.

表 2 相同期望误差下 4 种算法隐含层节点数方差的比较

Table 2 Variance of number of hidden layer node for four algorithms under same expected error

回归数据集	期望误差	I-ELM	CI-ELM	EM-ELM	CDAI-ELM
Auto MPG	0.11	49.46	5.21	2.76	1.82
Automobile	0.15	8.82	33.08	3.55	1.95
BlogFeedback	0.2	28.56	24.87	2.50	1.58
Housing	0.12	35.95	9.98	2.51	2.26
NoisyOffice	0.08	46.81	6.28	2.18	1.72
Facebook metrics	0.06	28.87	10.02	2.28	1.46
SML2010	0.21	36.89	17.79	2.82	2.31
wiki4HE	0.13	32.81	8.19	2.88	2.14
UJIIndoorLoc	0.09	50.71	9.67	3.07	2.51
YearPredictionMSD	0.08	51.21	12.13	3.51	2.87

表 3 4 种算法的测试误差和方差比较

Table 3 Comparison result of testing error and variance for four algorithms

回归数据集	I-ELM		CI-ELM		EM-ELM		CDAI-ELM	
	误差	方差	误差	方差	误差	方差	误差	方差
Auto MPG	0.1021	0.0051	0.0952	0.0041	0.0953	0.0052	0.0811	0.0040
Automobile	0.1323	0.0149	0.1302	0.0129	0.1301	0.0118	0.1257	0.0107
BlogFeedback	0.1896	0.0121	0.1882	0.0123	0.1712	0.0108	0.1822	0.0109
Housing	0.1017	0.0064	0.1008	0.0061	<u>0.0985</u>	0.0051	<u>0.0973</u>	0.0061
NoisyOffice	0.0511	0.0039	0.0481	0.0034	<u>0.0401</u>	0.0029	<u>0.0392</u>	0.0023
Facebook metrics	0.0642	0.0058	0.0581	0.0041	<u>0.0581</u>	0.0018	<u>0.0585</u>	0.0023
SML2010	0.1555	0.0158	0.1502	0.0129	<u>0.1461</u>	0.0078	<u>0.1452</u>	0.0074
wiki4HE	0.1592	0.0311	0.1522	0.0302	0.1468	0.0031	0.1511	0.0051
UJIIndoorLoc	0.1315	0.0102	0.1291	0.0091	0.1278	0.0072	0.1116	0.0059
YearPredictionMSD	0.0912	0.0041	0.0902	0.0039	0.0903	0.0040	0.0868	0.0031

表 4 4 种算法训练时间的比较 (s)

Table 4 Comparison result of training time for four algorithms (s)

回归数据集	I-ELM	CI-ELM	EM-ELM	CDAI-ELM
Auto MPG	0.0124	0.0073	0.0061	0.0052
Automobile	0.0272	0.0171	0.0182	0.0088
BlogFeedback	0.0469	0.0391	0.0236	0.0151
Housing	0.0391	<u>0.0119</u>	0.0182	<u>0.0120</u>
NoisyOffice	0.0411	0.0051	0.0083	0.0071
Facebook metrics	0.481	0.0179	0.0171	0.0159
SML2010	0.0218	0.0107	0.0081	0.0059
wiki4HE	0.0089	0.0081	0.0298	0.0297
UJIIndoorLoc	0.0471	0.0091	0.0288	0.0272
YearPredictionMSD	0.0301	0.0297	0.0271	0.0197

实验采集了 30 天的虚拟机的运行参数和能耗数据, 记录每天从 0 时到 24 时虚拟机的运行参数和能耗值, 并计算出当天的虚拟机的运行参数和能耗值的平均值作为当天的数据, 一共记录 30 天的数据, 形成数据集 $\phi = (\mathbf{x}_j, y_j)_{j=1}^M$, 其中, y_j 为第 j 天的虚拟机能耗值; 对应的第 j 天的输入样本表达为 $\mathbf{x}_j = [x_{1j}, x_{2j}, x_{3j}, x_{4j}]$, 其中 x_{1j} 表示为第 j 天的虚拟机 CPU 利用率, x_{2j} 表示为第 j 天的内存利用率, x_{3j} 表示为第 j 天的执行指令数, x_{4j} 表示为第 j 天的丢失的 Cache 数. 因为是短期预测, 故以前 20 组数据作为训练样本, 后 10 组数据作为测试样本.

5.3 评价指标

本文选取模型有效性 MV、均方根误差 (Root mean square error, RMSE)、平均绝对误差 (Mean absolute error, MAE)、相对百分误差 (Mean absolute percent error, MAPE) 和皮尔逊相关系数 (Pearson correlation coefficients, PCCs) 作为衡量预测模型的泛化性能与精度的评价指标^[22-24].

均方根误差 RMSE 可表示为:

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^K (\hat{\alpha}_j - y_j)^2}{K}} \quad (14)$$

模型有效性 MV 可表示为:

$$\text{MV} = 1 - \frac{\sum_{j=1}^K (\hat{\alpha}_j - y_j)^2}{\sum_{j=1}^K (\bar{\alpha} - y_j)^2} \quad (15)$$

平均绝对误差 MAE 可表示为:

$$\text{MAE} = \frac{1}{K} \sum_{j=1}^K |\hat{\alpha}_j - y_j| \quad (16)$$

相对百分误差 MAPE 可表示为:

$$\text{MAPE} = \frac{1}{K} \sum_{j=1}^K \frac{|\hat{\alpha}_j - y_j|}{\hat{\alpha}_j} \times 100\% \quad (17)$$

皮尔逊相关系数 PCCs 可表示为:

$$\text{PCCs} = \frac{\sum_{j=1}^K (y_j - \bar{y})(\hat{\alpha}_j - \bar{\alpha})}{\sqrt{\sum_{j=1}^K (y_j - \bar{y})^2} \sqrt{\sum_{j=1}^K (\hat{\alpha}_j - \bar{\alpha})^2}} \quad (18)$$

式中, y_j 为预测模型输出值, \bar{y} 为预测模型输出值的平均值, $\hat{\alpha}_j$ 为真实值, $\bar{\alpha}$ 为真实值的平均值, K 为样本数. 均方根误差 RMSE、平均绝对误差 MAE 和相对百分误差 MAPE 的大小反应了预测模型输出曲线在真实曲线上的波动情况, 模型有效性 MV 和皮尔逊相关系数 PCCs 反应了预测模型输出与真实值的误差相对于真实值的离散程度.

5.4 预测结果分析

为了验证本文所提算法的有效性, 采用支持向量机 (SVM)、基于核函数的极限学习机 (KELM)、宽度学习系统 (BLS) 和基于压缩动量项的增量型极限学习机 (CDAI-ELM) 分别对虚拟机能耗数据进行建模预测. CDAI-ELM 算法选用 Sigmoid 函数作为隐含层激励函数, 隐含层节点参数 (\mathbf{a}, b) 服从一致分布, 从 $[-1, 1]^d \times [-1, 1]$ 中随机选取, 最大隐含层节点数为 300. 选取 RBF 函数作为 SVM 和 KELM 的核函数, 即 $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$, 核函数参数和误差惩罚参数分别取值为 $\gamma_{\text{SVM}} = 2^{-0.3}$, $C_{\text{SVM}} = 2^{1.1}$, $\gamma_{\text{KELM}} = 2^3$, $C_{\text{KELM}} = 2^5$. 宽度学习算法选用双曲正切函数作为增强节点的激励函数, 特征节点数和增强节点数分别为 50 和 10, 收敛系数和正则化系数分别为 0.1 和 0.02.

从图 3 和 4 可知, 基于 CDAI-ELM 的能耗预测模型精度与其他模型相比有较大的改进. 与 BLS、SVM 和 KELM 相比之下, 基于 CDAI-ELM 的能耗预测模型的均方根误差分别减小了 14.987、10.8682、14.9334, 模型有效性分别提高了 0.161049、0.192734、0.191137, 平均绝对误差分别减少了 11.99114、9.187067、13.20551, 相对百分误差分别减少了 5.816683%、4.478663%、6.377395%, 皮尔逊相关系数分别提高了 0.275662、0.104929、0.310567. 根据表 5 可知, SVM 模型的训练时间均大于其他模型, KELM 模型训练时间与 BLS 模型接近, 而 CDAI-ELM 模型训练时间仅需 0.5772 s, 均优于其他模型, 其原因在于输入数据经过隐含层的特征映射后, 根据每一个隐含层节点的网络输出误差构造合适的压缩动量项, 隐含层输出矩阵的规

模均小于其他算法, 致使算法的计算复杂度降低, 网络训练时间变短. 根据图 3 和 4 及表 5 可知, BLS 模型、SVM 模型和 KELM 模型训练时间均大于 CDAI-ELM 模型, 且对虚拟机能耗数据进行预测时效果较差, 预测误差较大, 而 CDAI-ELM 模型对虚拟机能耗数据进行预测准确度较高, 预测误差小, 收敛速度快, 说明 CDAI-ELM 模型对虚拟机能耗进行模拟预测是行之有效的.

限学习机的研究进展, 针对增量型极限学习机因存在大量冗余节点, 致使网络结构复杂, 降低网络收敛速度, 通过引入压缩动量项优化增量型极限学习机网络结构, 提出了基于压缩动量项的增量型 ELM 虚拟机能耗预测方法, 实现了对虚拟机能耗的高效精准预测. 然而本文仅针对 5 个虚拟机进行预测建模, 如何对大规模的虚拟机集群进行预测建模, 仍需进一步进行实验验证.

References

- 1 Wu Zhi-Xue. Advances on virtualization technology of cloud computing. *Journal of Computer Applications*, 2017, **37**(4): 915–923
(武志学. 云计算虚拟化技术的发展与趋势. 计算机应用, 2017, **37**(4): 915–923)
- 2 Cui Yong, Song Jian, Miao Cong-Cong, Tang Jun. Mobile cloud computing research progress and trends. *Chinese Journal of Computers*, 2017, **40**(2): 273–295
(崔勇, 宋健, 缪葱葱, 唐俊. 移动云计算研究进展与趋势. 计算机学报, 2017, **37**(4): 915–923)
- 3 Xia Yuan-Qing, Yan Ce, Wang Xiao-Jing, Song Xiang-Hui. Intelligent transportation cyber-physical cloud control systems. *Acta Automatica Sinica*, 2019, **45**(1): 132–142
(夏元清, 闫策, 王笑京, 宋向辉. 智能交通信息物理融合云控制系统. 自动化学报, 2019, **45**(1): 132–142)
- 4 Jin Shun-Fu, Hao Shan-Shan, Wang Bao-Shuai. Virtual machine scheduling strategy based on dual-speed and work vacation mode and its parameter optimization. *Journal on Communications*, 2017, **38**(12): 10–20
(金顺福, 郝闪闪, 王宝帅. 融合双速率和工作休眠的虚拟机调度策略及参数优化. 通信学报, 2017, **38**(12): 10–20)
- 5 Zhao Chun, Yan Lian-Shan, Cui Yun-He, Xing Huan-Lai, Feng Bin. Dynamic adjusting threshold algorithm for virtual machine migration. *Journal of Computer Applications*, 2017, **37**(9): 2547–2550
(赵春, 闫连山, 崔允贺, 邢焕来, 冯斌. 基于动态调整阈值的虚拟机迁移算法. 计算机应用, 2017, **37**(9): 2547–2550)
- 6 Zhou Ping, Zhang Li, Li Wen-Peng, Dai Peng, Chai Tian-You. Autoencoder and PCA based RVFLNs modeling for multivariate molten iron quality in blast furnace ironmaking. *Acta Automatica Sinica*, 2018, **44**(10): 1799–1811
(周平, 张丽, 李温鹏, 戴鹏, 柴天佑. 集成自编码与 PCA 的高炉多元铁水质量随机神经网络建模. 自动化学报, 2018, **44**(10): 1799–1811)
- 7 Tang Yi-Xuan, Xue Xiao-Ru, Yao Zhen, Xu Min, Zhang Yong-Mei, Zhang He-Liang, Xu Dao-Lei. An energy-aware VM deployment strategy for the resource pool of power information system. *Electric Power Information and Communication Technology*, 2017, **15**(6): 95–100
(唐轶轩, 薛晓茹, 姚振, 徐敏, 张永梅, 张禾良, 徐道磊. 电力信息系统资源池的能耗感知虚拟机部署策略. 电力信息与通信技术, 2017, **15**(6): 95–100)
- 8 Xu H. Research on neural network based on virtual machine power prediction model [Master thesis], Beijing University of Posts and Telecommunications, 2015.
- 9 赵雅倩. 一种基于模糊神经网络的虚拟机能耗预测方法及系统, 中国专利 CN105975385A, 2016.09.28.

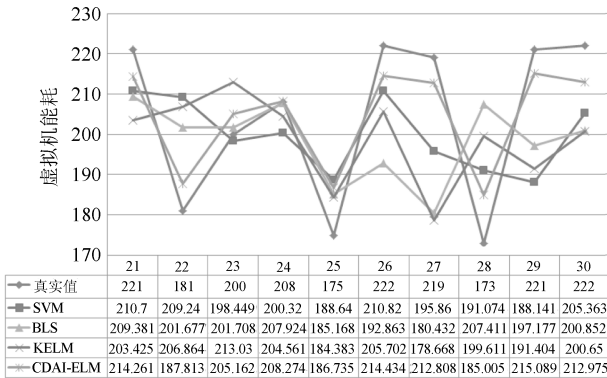


图 3 基于 SVM、KELM、BLS 和 CDAI-ELM 的虚拟机能耗预测曲线

Fig. 3 Predicted curve for power of virtual machine based on SVM, KELM, BLS 和 CDAI-ELM

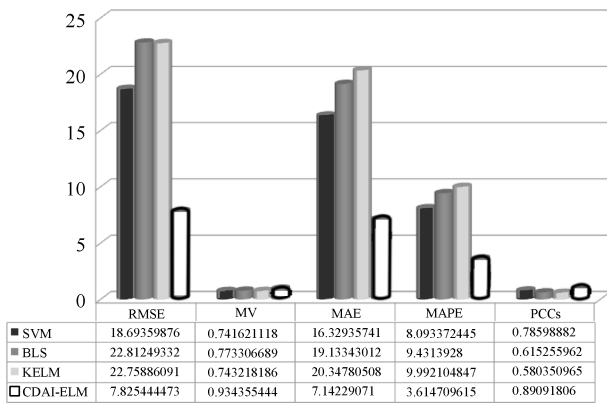


图 4 4 种模型预测结果

Fig. 4 Predicted results of four models

表 5 4 种模型训练时间 (s)

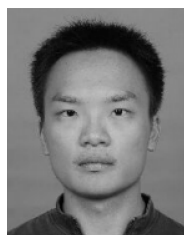
Table 5 Training time of four models (s)

预测模型	训练时间 (s)
SVM	3.4788
BLS	0.9828
KELM	1.06
CDAI-ELM	0.5772

6 结论

本文分析了国内外虚拟机能耗预测和增量型极

- 10 Jia Jiong-Hao, Chen Ning-Jiang, Li Xiang, Huang Ru-Wei. A dynamic adjustment strategy for virtual machines in cloud based on availability capability. *Journal of Guangxi University (Nat Sci Ed)*, 2016, **41**(3): 796–803
(贾灵昊, 陈宁江, 李湘, 黄汝维. 基于可用能力建模的云虚拟机动态调整策略. 广西大学学报(自然科学版), **41**(3): 796–803)
- 11 Huang G B, Chen L, Siew C K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 2006, **17**(4): 879–892
- 12 Tian Z D, Li S J, Wang Y H, Sha Y. Short-term wind power prediction based on empirical mode decomposition and improved extreme learning machine. *Journal of Electrical Engineering and Technology*, 2018, **13**(5): 1841–1851
- 13 Tian Z D, Li S J, Wang Y H, Sha Y. An adaptive online sequential extreme learning machine for short-term wind speed prediction based on improved artificial bee colony algorithm. *Neural Network World*, 2018, **28**(3): 191–212
- 14 Tian Z D, Li S J, Wang Y H, Sha Y. Network traffic prediction method based on improved ABC algorithm optimized EM-ELM. *Journal of China Universities of Posts and Telecommunications*, 2018, **25**(3): 33–44
- 15 Tang X L, Han M. Partial Lanczos extreme learning machine for single output regression problems. *Neurocomputing*, 2009, **13**(72): 3066–3076
- 16 Han Min, Li De-Cai. An norm 1 regularization term ELM algorithm based on surrogate function and Bayesian framework. *Acta Automatica Sinica*, 2011, **37**(11): 1344–1350
(韩敏, 李德才. 基于替代函数及贝叶斯框架的 1 范数 ELM 算法. 自动化学报, 2011, **37**(11): 1344–1350)
- 17 Huang G B, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 2012, **2**(42): 513–529
- 18 Chen C L P, Liu Z L. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **29**(1): 10–24
- 19 Sara S, Nicholas F, Geoffrey E H. Dynamic routing between capsules. In: Proceedings of the 31st Conference on Neural Information Processing Systems. California, USA, 2017. arXiv: 1710.09829
- 20 Wang B. The approximation order of convex incremental extreme learning machine [Master thesis], Northwest University, 2015
- 21 Knowledge discovery in databases, <http://www.kdd.org.htm>
- 22 Tian Zhong-Da, Li Shu-Jiang, Wang Yan-Hong, Wang Xiang-Dong. Network traffic prediction based on ARIMA with Gaussian process regression compensation. *Journal of Beijing University of Posts and Telecommunications*, 2017, **40**(6): 65–73
(田中大, 李树江, 王艳红, 王向东. 高斯过程回归补偿 ARIMA 的网络流量预测. 北京邮电大学学报, 2017, **40**(6): 65–73)
- 23 Tian Z D, Li S J, Wang Y H, Sha Y. A prediction method based on wavelet transform and multiple models fusion for chaotic time series. *Chaos, Solitons and Fractals*, 2017, **98**: 158–172
- 24 Tian Z D, Li S J, Wang Y H, Sha Y. Short-term wind speed prediction based on improved PSO algorithm optimized EM-ELM. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 2019, **41**(1): 26–46



邹伟东 北京理工大学自动化学院博士后. 主要研究方向为极限学习机, 云数据中心优化调度管理.

E-mail: zouweidong1985@163.com

(**ZOU Wei-Dong** Postdoctor at the School of Automation, Beijing Institute of Technology. His research interest covers extreme learning machine, cloud data center optimization scheduling and management.)



夏元清 北京理工大学自动化学院教授. 主要研究方向为云控制, 云数据中心优化调度管理, 智能交通, 模型预测控制, 自抗扰控制, 飞行器控制和空天地一体化网络协同控制. 本文通信作者.

E-mail: xia_yuanqing@bit.edu.cn

(**XIA Yuan-Qing** Professor at the School of Automation, Beijing Institute of Technology. His research interest covers cloud control, cloud data center optimization scheduling and management, intelligent transportation, model predictive control, active disturbance rejection control, flight control and networked cooperative control for integration of space, air and earth. Corresponding author of this paper.)