

LSTM 网络模型在 Web 服务器资源消耗预测中的应用研究^①



谭宇宁, 党伟超, 白尚旺, 潘理虎

(太原科技大学 计算机科学与技术学院, 太原 030024)
通讯作者: 党伟超, E-mail: dangweichao@tyust.edu.cn

摘要: 如何能够准确地对软件老化趋势进行预测, 并及时采取相应恢复策略是当前预防软件老化的一个关键问题. 为此, 针对老化数据的时序特性, 以循环神经网络 (Recurrent Neural Network, RNN) 及其变种长短时记忆单元 (Long Short-Term Memory, LSTM) 结构为基础, 设计了一种基于 LSTM 网络的软件老化资源预测方法, 并通过应用加速寿命测试实验搭建老化测试平台, 对 Web 服务器因内存泄漏引起的老化现象进行建模和预测. 实验结果表明, LSTM 老化预测模型在处理 Web 软件老化的时间序列建模问题上, 具有很强的适用性和更高的准确性, 能有效提高软件系统的可靠性和可用性.

关键词: 软件老化; 长短时记忆网络; 资源消耗; 循环神经网络; 软件可靠性

引用格式: 谭宇宁, 党伟超, 白尚旺, 潘理虎. LSTM 网络模型在 Web 服务器资源消耗预测中的应用研究. 计算机系统应用, 2019, 28(7): 214-220. <http://www.c-s-a.org.cn/1003-3254/6942.html>

Application of LSTM Network Model in Web Server Resource Consumption Prediction

TAN Yu-Ning, DANG Wei-Chao, BAI Shang-Wang, PAN Li-Hu

(School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China)

Abstract: How to predict the aging trend of the software accurately, and take the corresponding recovery strategy is a key problem of preventing software aging. To solve the problem, this study designs a resource prediction method based on Recurrent Neural Network (RNN) and its variant—Long Short-Term Memory (LSTM), and builds an accelerated aging test platform to model and forecast the aging phenomenon of the Web server due to memory leak. The experiments show that LSTM network prediction model proves to be superior to the other traditional models in dealing with the time sequence modeling of aging parameters, with the predicted results closer to the actual values and the higher prediction accuracy, which can effectively improve the reliability and availability of the software system.

Key words: software aging; LSTM; resource consumption; Recurrent Neural Network (RNN); software reliability

引言

软件老化是指软件 (特别是大访问量、大数据量的服务器软件) 在长期不间断地运行一段时间后系统

的性能持续下降、占用的资源不断增加、错误不断累积, 最终导致软件失效或系统宕机的现象^[1]. 为尽可能地减少甚至避免损失, 提高软件的可靠性和可用性,

① 基金项目: 山西省中科院科技合作项目 (20141101001); 山西省重点研发计划 (一般) 工业项目 (201703D121042-1); 山西省社会发展科技项目 (20140313020-1)

Foundation item: Science and Technology Collaborative Program between Shanxi Province and Chinese Academy of Sciences (20141101001); Key Research and Development Program for General Industry of Shanxi Province (201703D121042-1); Science and Technology Program for Social Development of Shanxi Province (20140313020-1)

收稿时间: 2018-12-10; 修改时间: 2019-01-11; 采用时间: 2019-01-15; csa 在线出版时间: 2019-07-01

Huang 等人提出一种主动性的容错策略—软件再生技术^[2](Software Rejuvenation, SR), 通过周期性地清除老化状态, 使得运行环境恢复正常, 从而避免因软件老化引起突发性失效. 尽管通过抗衰操作可以消除软件老化带来的影响, 然而对一个正常运行的系统执行抗衰操作势必会带来直接或间接的损失^[3], 因此如何能够准确地对软件老化趋势进行预测, 并及时采取相应恢复策略是当前预防软件老化的研究重点.

目前对于软件老化趋势的预测主要是对影响软件系统相关参数的资源损耗进行分析. 梁佩^[4]使用时间序列分析法以及马尔可夫模型对软件老化的资源消耗进行预测; 苏莉^[5]等人使用非线性有源回归网络模型来检测软件系统的老化现象; Jia^[6]等人则使用多元线性回归算法来分析和预测软件老化问题; 渊岚^[7]则建立了一个基于 AdaBoost 算法的 BP 神经网络模型来预测资源的消耗. 尽管很多学者使用回归分析法、时间序列法以及 BP 神经网络算法等方法来预测遭受软件老化影响的系统资源消耗情况, 然而已有的单一模型很难达到理想的预测效果^[8], 因此文献^[8]提出使用混合模型, 即将自回归累积移动平均模型 (Autoregressive Integrated Moving Average Model, ARIMA) 和人工神经网络结合来预测 Web 服务器中的资源消耗. 然而混合方法的构建过程复杂、人工依赖性强, 不利于在实际中推广和使用.

近年来, 随着深度学习技术的不断发展, 越来越多的深度学习模型逐渐被应用到各个领域. 深度学习模型是一种拥有多个非线性映射层级的深度神经网络模型, 能够对输入信号逐层抽象并提取特征, 挖掘出更深层次的潜在规律^[9]. 其中循环神经网络 (Recurrent Neural Network, RNN) 模型在结构设计中引入了时序概念, 在学习具有内在依赖性的时序数据时能够产生对过去数据的记忆状态, 能从原始数据中获取更多的数据波动以及规律性特征, 它的诞生解决了传统神经网络在处理序列信息方面的局限性. 作为近年来深度学习领域热点技术之一, 在机器翻译、语音识别及图像识别领域都取得了巨大成功^[10], 然而在软件可靠性领域对于资源消耗的预测目前还未发现展开过相关研究.

基于上述分析, 本文提出了一种基于 LSTM 的 Web 资源消耗预测模型, 该模型充分考虑了 Web 资源损耗的时间特性, 将当前的资源损耗情况动态的与历史数据相关联并将其与传统模型进行实验对比. 结果

表明该资源消耗预测模型在处理老化数据的时间序列建模问题上预测精度更高, 能够有效地应用于软件老化趋势的预测.

1 模型原理

1.1 循环神经网络

RNN 是一类由各神经元相互连接形成的有向循环神经网络, 其基于时序展开后的结构如下图 1 所示. 与传统的前馈神经网络 (Feedforward Neural Network, FNN) 不同, RNN 不仅通过层与层间的连接进行信息的传递, 而且通过在网络中引入环状结构, 建立了神经元到自身的连接. 每一步的输出不仅包括当前所见的输入样例, 还包括网络在上一个时刻所感知到的信息即当前时刻的 h_t 不仅仅取决于当前时刻的输入 x_t , 而且与上一时刻的 h_{t-1} 也相关.

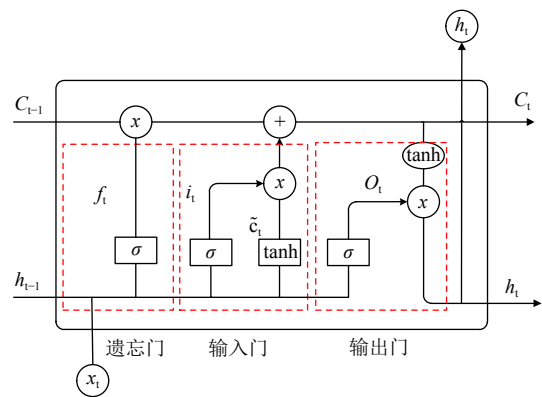


图 1 按照时序展开的 RNN 结构图

简单的循环神经网络由 1 个输入层、1 个隐含层以及 1 个输出层组成. 给定输入向量序列 $x=[x_1, x_2, \dots, x_T]$, 通过迭代下列公式 (1) 首先计算出 $t=1$ 至 $t=T$ 的隐含层状态序列 $h=[h_1, h_2, \dots, h_T]$, 然后根据公式 (2) 计算出输出序列 $o=[o_1, o_2, \dots, o_T]$.

$$h_t = f(U \cdot x_t + W \cdot h_{t-1} + b_h) \quad (1)$$

$$o_t = g(V \cdot h_t + b_o) \quad (2)$$

式中, U 为输出层到隐含层的权重矩阵; W 为隐含层到隐含层的权重矩阵; V 为隐含层到输出层的权重矩阵, f 和 b 分别表示输入层到隐含层的激活函数以及偏置, g 和 b 分别表示隐含层到输出层的激活函数以及偏置. 相比于 FNN 需要 n 个时刻来帮助学习一次权重, RNN 可以用 n 个时刻学习 n 次 W 和 U , 实现了在时间结构上的共享特性.

将式 (1) 带入式 (3) 可得:

$$o_t = V \cdot g(U \cdot x_t + W \cdot g(U \cdot x_{t-1} + W \cdot g(U \cdot x_{t-2} + W \cdot g(U \cdot x_{t-3} + \dots)))) \quad (3)$$

由式 (3) 可知循环神经网络在计算过程中虽然加入了上一时刻的输入, 但随着时间的推移, 后面节点对前面节点的感知能力下降, 即 RNN 存在梯度消失问题.

1.2 长短时记忆单元

与传统 RNN 结构相比, LSTM 在其基础上增加了一个细胞状态 (cell state). 在传递过程中, 通过当前输入、上一时刻隐藏层状态、上一时刻细胞状态以及三个基于 Sigmoid 函数的门结构来增加或删除细胞状态中的信息, 其具体单元结构如下图 2 所示. 其中门结构用来控制即时信息对历史信息的影响程度, 通过线性积累, 使得网络模型能够较长时间保存并传递信息^[11].

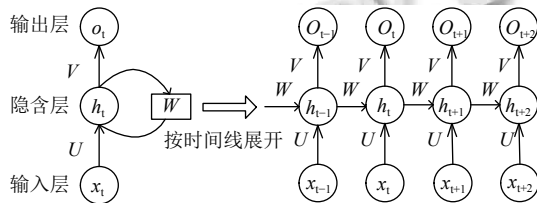


图2 LSTM 单元结构图

一个典型的 LSTM 单元共有三个门: 遗忘门、输入门以及输出门^[12]. 其中遗忘门和输入门主要用来控制上一时刻细胞状态 \$C_{t-1}\$ 以及当前输入新生成的 \$\tilde{c}_t\$ 中有多少信息可以加入到当前的细胞状态 \$C_t\$ 中来, 通过遗忘门和输入门的输出, 更新细胞状态, 输出门基于更新后的细胞状态输出隐藏状态 \$h_t\$, 各门计算公式如式 (4).

$$\begin{cases} f_t = \sigma(W_f \cdot h_{t-1} + W_f \cdot x_t + b_f) \\ i_t = \sigma(W_i \cdot h_{t-1} + W_i \cdot x_t + b_i) \\ o_t = \sigma(W_o \cdot h_{t-1} + W_o \cdot x_t + b_o) \end{cases} \quad (4)$$

其中: \$f_t, i_t, o_t\$ 分别表示遗忘门、输入门、输出门的结算结果; \$W_f, W_i, W_o\$ 分别为遗忘门、输入门、输出门的权重矩阵; \$b_f, b_i, b_o\$ 分别为遗忘门、输入门、输出门的偏置项. 最终的输出由输出门和单元状态共同确定, 具体计算公式如式 (5) 所示.

$$\begin{cases} h_t = o_t \odot \tanh(c_t) \\ \tilde{c}_t = \tanh(W_c \cdot h_{t-1} + W_c \cdot x_t + b_c) \\ c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \end{cases} \quad (5)$$

式中, \$x_t\$ 为 \$t\$ 时刻输入的单元状态; \$W_c\$ 为输入单元状态权重矩阵; \$b_c\$ 为输入单元状态偏置项; \$\tanh()\$ 为激活函数, \$\odot\$ 表示 hadamard 乘积.

1.3 LSTM 预测模型具体构建过程

Web 服务器资源消耗预测就是根据前 \$t\$ 时刻老化指标的资源使用特征来预测 \$t+1\$ 或者 \$t+x\$ 时间内的资源损耗, 以此判断 Web 服务器的老化状况. 因此通过使用老化数据对 LSTM 神经网络进行训练, 构建基于 LSTM 网络的软件老化资源预测模型, 其具体构建步骤如下所示:

(1) 首先将原始老化数据清洗后进行特征表示和特征提取: 定义老化资源损耗时间序列 \$F = \{f_1, f_2, \dots, f_n\}\$, 将其划分为测试集 \$F_{\text{train}} = \{f_1, f_2, \dots, f_m\}\$ 和训练集 \$F_{\text{test}} = \{f_{m+1}, f_{m+2}, \dots, f_n\}\$, 其中 \$m < n\$ 且 \$m, n \in N\$, 对 \$F_{\text{train}}\$ 集合中的元素 max-min 标准化, 处理后的训练集表示为式 (6).

$$F'_{\text{tr}} = \{f'_1, f'_2, \dots, f'_m\}$$

$$f'_i = \left(f_i - \min_{1 \leq j \leq m} \{f_j\} \right) / \left(\frac{\max_{1 \leq j \leq m} \{f_j\} - \min_{1 \leq j \leq m} \{f_j\}}{1} \right) \quad (6)$$

(2) 构建训练输入以及对应的训练输出: 对数据集进行时间融合, 按照滑动窗口的大小 \$s\$ 进行分割, 则模型输入、输出分别变为式 (7)、式 (8). 通过设置 \$s\$ 的值, 旨在训练 LSTM 网络学习样本数据前后的关联及规律.

$$\begin{cases} X = \{X_1, X_2, \dots, X_S\} \\ X_p = \{f'_p, f'_{p+1}, \dots, f'_{m-s+p-1}\} \end{cases} \quad (7)$$

$$\begin{cases} Y = \{Y_1, Y_2, \dots, Y_S\} \\ Y_p = \{f'_{p+1}, f'_{p+2}, \dots, f'_{m-t+p}\} \end{cases} \quad (8)$$

(3) 确定网络结构并初始化 LSTM 网络: 确定每层激活函数的选择、每层网络节点的舍弃率以及误差计算方式、权重参数迭代的更新方式. 给定初始权值矩阵, 设置最大迭代训练次数和最小误差值, 通过改变网络的各项参数来训练网络.

(4) 前向计算: 将 \$X\$ 输入网络, 根据前向计算公式 (4)–(5) 计算遗忘门、输入门以及输出门的值, 经过隐藏层后的输出结果可表示为: 其中 \$C_{p-1}\$ 和 \$H_{p-1}\$ 分别表示上一个 LSTM 细胞的状态以及隐含层的输出.

$$\begin{cases} P = \{P_1, P_2, \dots, P_L\} \\ P_p = \text{LSTM}_f(X_p, C_{p-1}, H_{p-1}) \end{cases} \quad (9)$$

(5) 误差反向传播: 采用批量梯度下降算法对训练数据进行批次 (batch) 划分, 通过对当前批次的损失函数进行优化, 实时调整 LSTM 网络的权值和偏置, 使网络误差不断减小, 既保证了参数的更新又减少了模型收敛所需要的迭代次数.

$$loss = \sum_{i=1}^{L(m-L)} (p_i - y_i)^2 / (L(m-L)) \quad (10)$$

(6) 将训练好的模型用于预测: 当迭代次数和最小误差值满足要求时停止训练模型, 并将未知的样本数据通过迭代输入模型得到预测序列 $P_{te} = \{p_{m+1}, p_{m+2}, \dots, p_n\}$, 并对其进行反标准化处理得到最终预测序如式 11.

$$\begin{cases} P'_{te} = Anti(P_{te}) = \{p'_{m+1}, p'_{m+2}, \dots, p'_n\} \\ p'_k = p_k \left(\max_{1 \leq j \leq m} \{f_j\} - \min_{1 \leq j \leq m} \{f_j\} \right) + \min_{1 \leq j \leq m} \{f_j\} \end{cases} \quad (11)$$

为评估基于 LSTM 的 Web 资源消耗预测模型的性能, 运用平均绝对误差 (Mean Absolute Error, MAE) 和均方根误差 (Root Mean Squared Error, RMSE) 作为评价指标来衡量模型的预测精度, 其计算公式分别如式 (12) 和式 (13) 所示.

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - y_i| \quad (12)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - y_i)^2} \quad (13)$$

其中, n 为 Web 老化资源参数的样本数目, p_i 为老化资源参数的预测值, y_i 为老化资源参数的真实值或观测值, MAE 和 RMSE 越小, 模型预测能力越好.

2 实验与结果分析

由于软件老化是一个错误不断累积的过程, 一个 Web 系统出现老化现象时并不会立刻失效, 需要耗费很长时间才能观察到系统故障. 当前绝大多数的老化实验只是简单识别老化效应, 很难准确把握软件老化的整体趋势^[13]. 因此针对软件老化的特性, 本文根据 R. Matias 等人提出的系统化方法, 将工业领域已成熟应用的加速测试理论^[14]引入到软件领域, 通过监测软件系统的运行状况, 根据采集到的老化数据, 建立一个基于 LSTM 的 Web 资源消耗预测模型.

2.1 Web 软件老化加速寿命实验

为研究因内存泄漏而导致应用程序故障的老化效应, 本文以一个典型的 Web 应用服务器为研究对象, 搭建了一个引入内存泄漏的软件老化测试实验平台. 该平台由一个 Web 服务器, 一个数据库服务器以及一组模拟的客户端组成, 具体配置如表 1 所示.

服务器端实现了一个符合 TPC-W 基准测试规范的多层电子商务网站系统. 该系统模拟了一个在线售

书网站, 包括主页、畅销页面、新书页面、搜索页面、购物车和订单状态页面等 14 种不同类型的网页, 并规定了一系列模拟真实环境下顾客的访问规则. 客户端则是一系列模拟浏览器 (Emulated Browser, EB), 以会话 (Session) 为单位与服务器端建立逻辑请求, 按照上述规则访问服务器. 模拟浏览器可以产生三种不同类型的工作负载, 分别是 Browsing 类型、Shopping 类型以及 Ordering 类型^[15]. 因 Shopping 类型的工作负载处于 Browsing 和 Ordering 之间, 因此本实验客户端主要模拟 Shopping 这种类型的工作负载, 以随机生成的概率对不同页面进行访问.

表 1 实验环境配置描述

配置	说明
-Xms 256 M	最小堆内存 256 MB
-Xmx 256 M	最大堆内存 256 MB
-Xmn 60 M	年轻代 60 MB

内存泄漏是造成软件老化的一个重要原因, 因此内存使用情况是衡量软件老化的一个重要指标, 通过使用采集到的 Java 虚拟机 (Java Virtual Machine, JVM) 可用内存对 Web 资源消耗进行预测来验证 LSTM 预测模型的准确性. 为产生软件老化现象, 修改了服务器端商品查询请求的 TPC-W_search_request_servlet 类, 为其注入内存泄漏代码. 由于 JVM 有垃圾回收 (Garbage Collection, GC) 机制, 任何不再被引用的对象都会被垃圾回收器回收, 其占用的内存也会被释放以便新对象使用. 为模拟内存泄漏现象, 增加了一个 HeapLeak 类, 使得 Tomcat 的整个生命周期保持对该类 HeapLeak 对象的引用, HeapLeak 对象在程序运行期间不会被垃圾回收器回收. 修改 JVM 堆内存的配置 (表 2), 使实验在受控环境下进行操作. 由于 Java 堆存的是对象实例, 所以当创建的对象实例数量达到最大堆容量限制后会造成堆溢出.

运行客户端, 每隔 1 秒采集一次 JVM 内存使用量, 实验持续 14 400 s, 共 4 个小时, 采集到样本 14 400 个. 每 30 s 取一次均值, 得到实验数据 (图 3).

2.2 LSTM 资源消耗预测建模

本文使用 Keras 框架搭建并训练 LSTM 预测模型, 所使用的网络主要由循环层 (Recurrent) 中的 LSTM 层和全连接层 (Dense) 组成. 取前 9650 个点 (真实时间近似 2.6 个小时) 对未来 4750 个点 (真实时间近似 1.4 小时) 进行建模预测, 即使用 67% 的数据作为训练集,

33% 的数据作为测试集. 根据 2.3 节提出的模型具体构建过程对标准化后的 JVM 内存序列建立一个含 30 个隐藏神经元的单层 LSTM 老化资源消耗预测模型, 根据网格搜索参数寻优法确定模型参数, 设置迭代次数 epoch=20, batch=10, time steps=10, lr=0.001, 损失函数为 MSE. 采用 Adam(Adaptive Moment Estimation) 算法对 lr 进行优化, 利用梯度的一阶矩和二阶矩估计

动态调整每个参数的学习率, 使得 lr 平稳迭代, 模型参数有效更新. 由于神经网络含有多个网络层以及大量参数, 为防止模型发生过拟合现象, 采用 Dropout 对数据进行正则化处理即在每轮权重更新时随机选择隐去一些节点, 从而限制模型单元之间的协同更新^[16]. 该模型使用的 Dropout 为 0.5, 即含有 Dropout 的网络层在训练过程中, 会有 50% 的节点被抛弃.

表 2 JVM 堆内存配置描述

	客户端	服务器	数据库
硬件	Intel® Core™ Duo CPU E8400@3.00 GHz 3.00 GHz	Intel® Xeon® CPU E5506@2.13 GHz	Intel® Xeon® CPU E5506@2.13 GHz
操作系统	Linux Centos 7kernel3.10.0-693.2.2.el7.	Linux Centos7kernel3.10.0-693.2.2.el7.	Linux Centos7kernel3.10.0-693.2.2.el7.
JVM		JDK 1.6.0	
软件	TPC-W 客户端	Apache Tomcat 6.0.43	MySQL5.6.35

2.3 实验分析

为验证 LSTM 模型在循环神经网络中的优势, 将 LSTM 中的隐含层单元替换为 RNN 结构, 按照上述相同参数进行实验, 结果如图 4 所示. 该图从整体上反映出了 RNN 以及 LSTM 资源消耗预测模型的预测能力, 其中实线表示真实值, 虚线表示测试值. 由图 4 可知两种预测模型测试值与真实值接近, 预测趋势与实际资源消耗趋势基本一致, 对于出现较大波动处的点也有较好的拟合, 说明 RNN 以及 LSTM 模型能有效地对软件老化趋势进行预测.

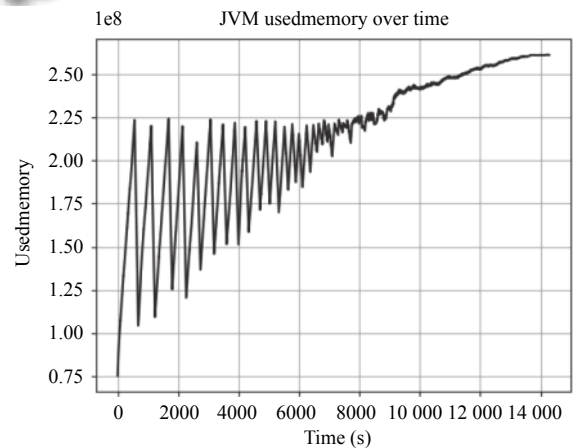


图 3 实验数据

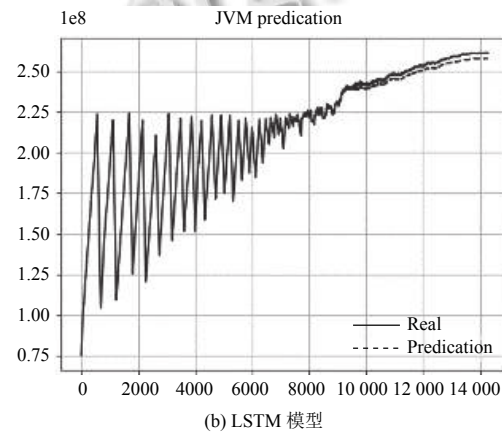
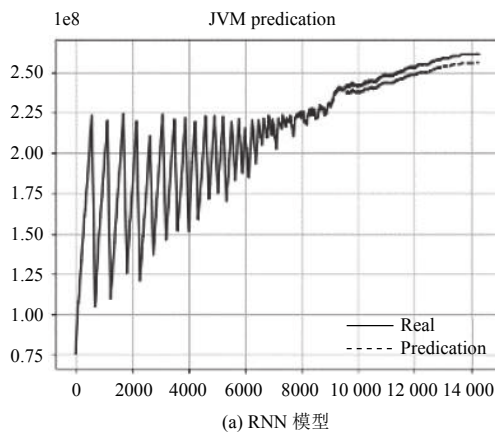


图 4 实验预测结果图

由 2 种模型对应的损失函数图 5(a)、5(b) 可知: 在老化资源时序预测问题上, 相比于 RNN, LSTM 算法训练过程相对稳定, 测试集上的误差波动较小.

为进一步验证 LSTM 的预测能力, 将其与传统预测方法: ARIMA 以及 BP 神经网络做对比, 实验结果如下图 6 所示, 由图可知 LSTM 模型的拟合

程度更好. 通过对 3 种不同预测模型的评价指标进行对比, 由表 3 可知, 采用 LSTM 网络预测算法在

预测 Web 老化资源时, 预测精度明显高于其他两种算法.

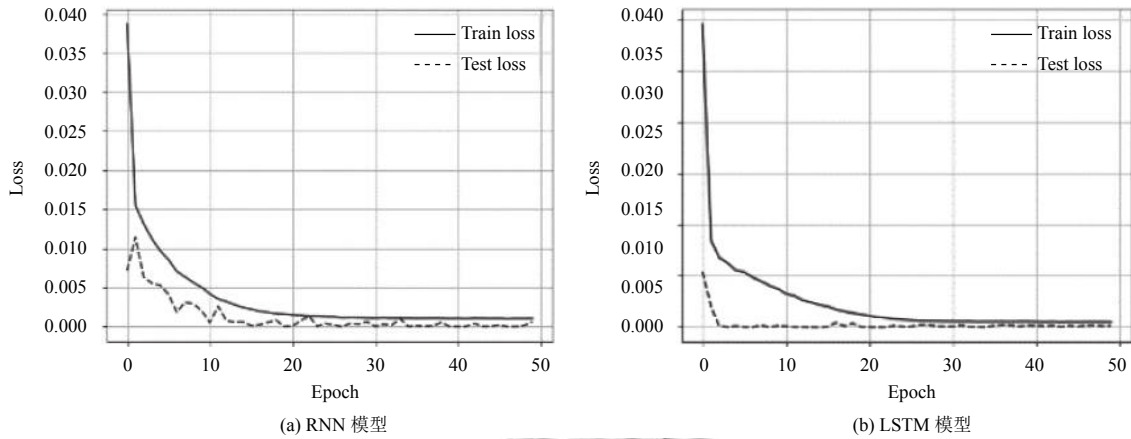


图 5 实验损失函数图

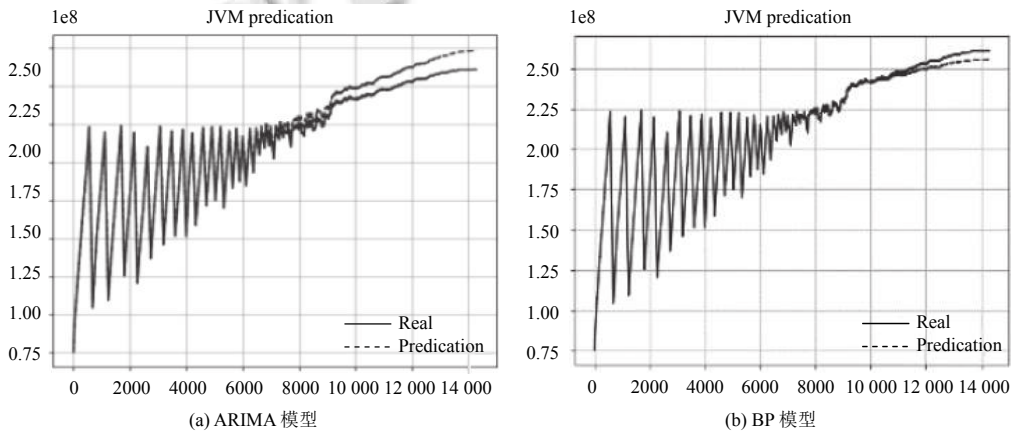


图 6 实验对比结果预测图

表 3 预测精度对比

算法	评价指标		
	ARIMA	BP	LSTM
RMSE	6.19×10^6	3.61×10^6	2.50×10^5
MAE	6.08×10^6	3.07×10^6	2.23×10^5

3 结束语

软件老化是影响软件系统可靠性的重要潜在因素, 本文以一个典型的 Web 应用服务器为实例, 通过随机注入内存泄漏的方式设计加速寿命测试实验来加速系统老化过程, 根据获取的老化数据构建了基于 LSTM 的 Web 服务器资源消耗预测模型. 结果证明该预测模型与 Web 服务器资源老化趋势一致, 拟合度很高, 能准确地描述软件老化现象. 与 ARIMA 以及 BP

神经网络相比预测度高、泛化能力好、误差较小, 说明 LSTM 网络模型能够很好地描述 Web 服务器资源的动态、非线性变化规律, 适用于老化参数的时间序列建模.

参考文献

- 1 Yan YQ. Variance analysis of software ageing problems. IET Software, 2018, 12(1): 41–48. [doi: 10.1049/iet-sen.2016.0290]
- 2 Kolettis N, Fulton N D. Software rejuvenation: Analysis, module and applications. Twenty-Fifth International Symposium on Fault-Tolerant Computing. Washington, DC, USA, 1995. 381–390.
- 3 Torquato M, Araujo J, Umesh I M, et al. SWARE: A methodology for software aging and rejuvenation

- experiments. *Journal of Information Systems Engineering & Management*, 2018, 3(2): 15.
- 4 梁佩. 基于 WEB 技术的软件老化趋势预测研究[硕士学位论文]. 哈尔滨: 哈尔滨工程大学, 2013.
 - 5 苏莉, 齐勇, 金玲玲, 等. 基于非线性多参数模型的软件老化检测. *计算机科学*, 2013, 40(1): 161–165, 170. [doi: [10.3969/j.issn.1002-137X.2013.01.037](https://doi.org/10.3969/j.issn.1002-137X.2013.01.037)]
 - 6 Jia SQ, Hou CY, Wang JS. Software aging analysis and prediction in a web server based on multiple linear regression algorithm. 2017 IEEE 9th International Conference on Communication Software and Networks. Guangzhou, China. 2017. 1452–1456.
 - 7 渊岚. 基于人工神经网络的软件系统老化预测研究[硕士学位论文]. 西安: 西安建筑科技大学, 2016.
 - 8 闫永权, 郭平. 使用混合模型预测 Web 服务器中的资源消耗. *计算机科学*, 2016, 43(10): 47–52. [doi: [10.11896/j.issn.1002-137X.2016.10.008](https://doi.org/10.11896/j.issn.1002-137X.2016.10.008)]
 - 9 LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, 521(7553): 436–444. [doi: [10.1038/nature14539](https://doi.org/10.1038/nature14539)]
 - 10 梁天新, 杨小平, 王良, 等. 记忆神经网络的研究与发展. *软件学报*, 2017, 28(11): 2905–2924. [doi: [10.13328/j.cnki.jos.005334](https://doi.org/10.13328/j.cnki.jos.005334)]
 - 11 Sak H, Senior A, Beaufays F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Fifteenth Annual Conference of the International Speech Communication Association. Singapore. 2014. 338–342.
 - 12 Cortez B, Carrera B, Kim Y J, *et al.* An architecture for emergency event prediction using LSTM recurrent neural networks. *Expert Systems with Applications*, 2018, 97: 315–324. [doi: [10.1016/j.eswa.2017.12.037](https://doi.org/10.1016/j.eswa.2017.12.037)]
 - 13 Zhao J, Jin YL, Trivedi KS, *et al.* Software rejuvenation scheduling using accelerated life testing. *ACM Journal on Emerging Technologies in Computing Systems*, 2014, 10(1): 1–23.
 - 14 Yin YC, Coolen FPA, Coolen-Maturi T. An imprecise statistical method for accelerated life testing using the power-Weibull model. *Reliability Engineering & System Safety*, 2017, 167: 158–167.
 - 15 Bezenek T, Cain T, Dickson R, *et al.* TPC-WBenchmark. <http://pharm.ece.wisc.edu/tpcw.shtml>.
 - 16 Srivastava N, Hinton G, Krizhevsky A, *et al.* Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014, 15(1): 1929–1958.