doi: 10.3969/j.issn.0490-6756.2019.01.014

基于 LSTM 的 WEB 服务响应时间大数据预测方法

刘承启[1], 林振荣[2], 黄文海[1]
(1.南昌大学网络中心, 南昌 330031; 2.南昌大学信息工程学院, 南昌 330031)

摘 要: 有效地预测 Web 服务器响应时间,对 Web 服务提供方保障服务质量有着重要的指导意义.利用大数据方法对处理大量历史指标数据的处理能提高预测的效率.本文提出一种使用相关性分析除去与响应时间相关性不高的指标项,使用特征降维的方法减小计算的数据量,使用动态调节参数的多层 LSTM 优化算法对数据做训练并预测响应时间的方法来提高预测的效率和准确率.通过实验证明,本文提出的方法能高效和准确地预测 Web 服务响应时间.

关键词: Web 服务; 响应时间; LSTM; 大数据
中图分类号: TP18 **文献标识码**: A **文章编号**: 0490-6756(2019)01-0071-07

WEB service response time prediction method based on big data and LSTM

LIU Cheng-Qi[1], *LIN Zhen-Rong*[2], *HUANG Wen-Hai*[1]
(1. Center of Network, Nanchang University, Nanchang 330031, China;
2. School of Information Engineering, Nanchang University, Nanchang 330031, China)

Abstract: Effective prediction of web services response time has important guiding significance for service providers to guarantee the quality of service, using big data approach to process a large number of indictors' historical data can improve the efficiency of prediction. Correlation analysis is proposed to remove indicators' items that are not highly correlated with response time, the computed data volume is reduced by the feature dimension reduction, and the multi-layer LSTM optimization algorithm with dynamic adjustment parameters is designed to predict the response time of web services. Experiments show that the proposed method can predict the response time of Web services efficiently and accurately.

Keywords: Web service; Response time; LSTM; Big data

1 引 言

Web 服务器响应时间是衡量 Web 服务端性能的重要指标,合理的响应时间是用户接受网站的必要条件.目前已有很多研究人员对 Web 服务器响应时间做了深入研究[1-3].文献[1]通过构建服务器端软件网状性能度量模型来计算各性能指标与服务器响应时间的影响程度.文献[2]提出一种基于时间序列分析的 Web 服务响应时间的动态预测方法.文献[3]用 GA 遗传算法优化 BP 神经网络的权值阈值,用改进的 BP 算法预测云服务响应时间.上述文献中的预测方法中单独考虑服务器响应时间与时间的相关性,或响应时间与其它指标的相关性,没有把响应时间同时与时间和其它指标相关.本文的出发点是把响应时间与时间和其它相关指标同时做计算,综合两种思路的优点来预测响应时间.

RNN(Recurrent Neural Network)是一种用

收稿日期: 2018-07-26
基金项目: 江西省科技支撑计划项目(20151BBE50057); 江西省教育厅科技项目(GJJ161675, GJJ161675)
作者简介: 刘承启(1977-), 男, 江西鄱阳人, 硕士, 工程师, 研究方向为计算机网络与数据挖掘. E-mail: splcq@ncu.edu.cn
通讯作者: 林振荣. E-mail: zrlin@ncu.edu.cn

于处理序列数据的神经网络[4]. LSTM(Long Short Term Memory network)是 RNN 的一种变体[5,6],它解决了简单的 RNN 结构求解过程中易发生的梯度消失或梯度爆炸问题. 当前运维大数据已成为运维发展的热门方向,本文主要阐述在大数据框架下如何结合 LSTM 算法对 Web 服务相关多项指标做计算,高效预测 Web 服务响应时间.

2 相关理论和技术

2.1 Pearson 相关系数

Pearson 相关系数[7]用于判断两组数的线性关系程度. 对于随机变量 X 和 Y,皮尔森相关系数的表达为(x 和 y 的协方差)/(x 的标准差 * y 的标准差),如式(1)所示.

$$r=\frac{\sum XY-\frac{\sum X\sum Y}{N}}{\sqrt{\left(\sum X^2-\frac{(\sum X)^2}{N}\right)\left(\sum Y^2-\frac{(\sum Y)^2}{N}\right)}} \quad (1)$$

其中,相关系数 r 取值在 -1 到 1 之间,$r=0$ 时,称 X,Y 不相关;$|r|=1$ 时,称 X,Y 完全相关,此时,X,Y 之间具有线性函数关系;$|r|<1$ 时,X 的变动引起 Y 的部分变动,r 的绝对值越大,X 的变动引起 Y 的变动就越大.

2.2 PCA 特征降维

PCA(Principal Component Analysis)[8]一种线性降维方法,它的目标是通过某种线性投影,将高维的数据映射到低维的空间中表示,并期望在所投影的维度上数据的方差最大,以此使用较少的数据维度,同时保留住较多的原数据点的特性.

设 n 维向量 w 为目标子空间的一个坐标轴方向(称为映射向量),最大化数据映射后的方差,有

$$\max_W \frac{1}{m-1}\sum_{i=1}^{m}(\boldsymbol{W}^T(x_i-\bar{x}))^2 \quad (2)$$

其中,m 是数据实例的个数;x_i 是数据实例 i 的向量表达;x 是所有数据实例的平均向量;定义 $\boldsymbol{W}$ 为包含所有映射向量为列向量的矩阵,经过线性代数变换,可以得到如下优化目标函数.

$$\min_W tr(\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{W}),\ \text{s.t.}\ \boldsymbol{W}^T\boldsymbol{W}=I \quad (3)$$

其中,tr 表示矩阵的迹.

$$\boldsymbol{A}=\frac{1}{m-1}\sum_{i-1}^{m}(x_i-\bar{x})(x_i-\bar{x})^T \quad (4)$$

其中,$\boldsymbol{A}$ 是数据协方差矩阵. 得到最优的 $\boldsymbol{W}$ 是由数据协方差矩阵前 k 个最大的特征值对应的特征向量作为列向量构成的. 这些特征向量形成一组正交基并且最好地保留了数据中的信息.

PCA 的输出就是 $Y=W'X$,由 X 的原始维度降低到了 k 维.

2.3 LSTM(Long Short-Term Memory)

LSTM 是长短期记忆网络,是一种时间递归神经网络,其结构如图 1 所示.

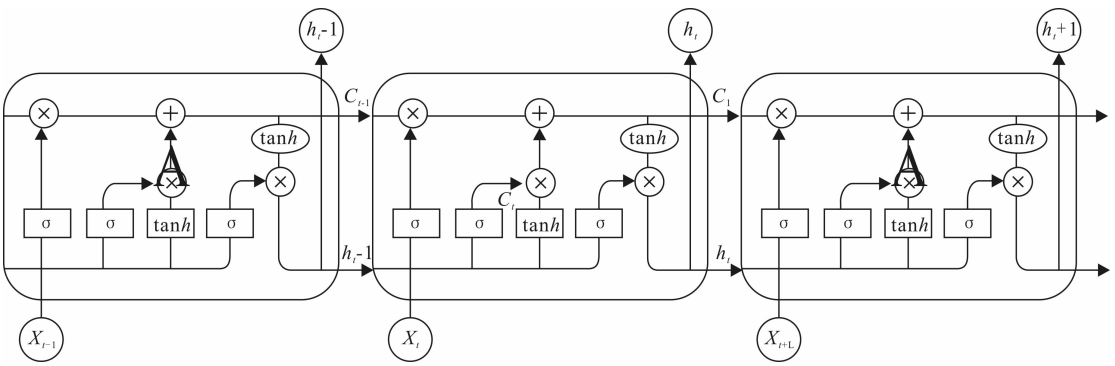


图 1 LSTM 示意图
Fig. 1 LSTM schematic diagram

图 1 中,LSTM 的细胞单元用输入门、遗忘门和输出门等 3 个门结构,来控制细胞状态和隐藏状态. 门结构用 sigmoid 函数和向量点乘来实现来控制增加或删除信息的程度. 遗忘门(forget gate)决定上一时刻细胞状态 C_{t-1} 中的多少信息(由 f_t 控制,值域为(0,1)) 可以传递到当前时刻 C_t 中;输入门 (input gate)用来控制当前输入新生成的信息 $\tilde{C}_t$ 中有多少信息(由 i_t 控制,值域为 (0,1))可以加入到细胞状态 C_t 中. tanh 层用来产生当前时刻新的信息,sigmoid 层用来控制有多少新信息可以传递给细胞状态;基于遗忘门和输入门的输出,来更新细胞状态. 更新后的细胞状态由两部分构成,分别是来自上一时刻旧的细胞状态信息 C_{t-1} 和当前输入新生成的信息 $\tilde{C}_t$;输出门(output gate) o_t

控制有多少细胞状态信息($\tanh(C_t)$，将细胞状态缩放至 $(-1,1)$) 可以作为隐藏状态的输出 h_t.

LSTM 的前向计算方法可表示为[9]

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f) \tag{5}$$

$$C_t=f_t*C_{t-1}+i_t*\widetilde{C}_t \tag{6}$$

$$i_t=\sigma(W_i\cdot[h_{t-1},x_t]+b_i) \tag{7}$$

$$\widetilde{C}_t=\tan h(W_c\cdot[h_{t-1},x_t]+b_c) \tag{8}$$

$$o_t=\sigma(W_o\cdot[h_{t-1},x_t]+b_o) \tag{9}$$

$$h_t=o_t*\tan h(C_t) \tag{10}$$

其中，x_t 为输入；h_t 为输出；i_t 为输入门的输出；f_t 为遗忘门的输出；c_t 为当前时刻 t 的细胞单元状态；o_t 为输出门的输出；W 和 b 为参数矩阵；σ 为 sigmoid 激活函数；tan$h$ 为双曲正切激活函数.

3　研究方法

Web 服务器响应时间取决于 Web 服务器软硬件环境、Web 应用中间件和数据库中间件等诸多方面，每个方面有多项指标. 需要从众多指标项中分析出与响应时间相关度大的指标用于预测. 结合第 2 节介绍的相关理论和技术，本节给出在大数据环境下基于 LSTM 的 Web 服务器响应时间预测方法.

3.1　大数据服务器响应预测框架

本文设计的大数据服务器响应预测框架如图 2 所示.

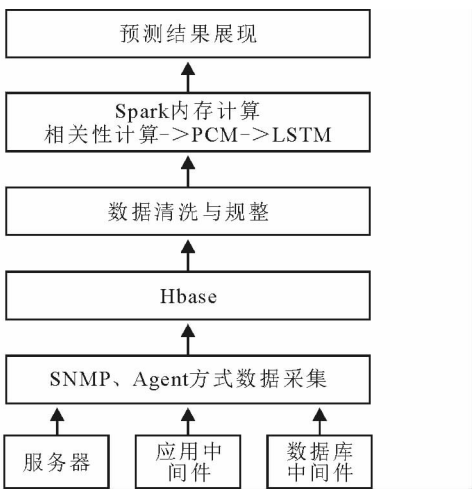


图 2　大数据服务器响应预测框架

Fig. 2　Big data server response prediction framework

如图 2 所示，首先采用 SNMP 和 Agent 的方式采集服务器、应用中间件和数据库中间件等运行指标，这些指标采集后存入 HBase 库中. 这些指标包括服务器监控类：CPU 利用率、内存利用率、网卡流量及进程个数等指标；应用中间件监控类：活动 socket 连接数、当前可用堆栈、堆栈大小、缓存个数、当前活动连接计数、堆可用百分比、JVM 当前堆栈利用率、server 的请求队列大小及应用当前 session 个数等；数据库中间件监控类：表空间使用率、阻塞会话数、当前打开线程数、用户连接数、并发最大连接数及当前锁总数等指标. 由于性能指标众多，本文不再一一列举.

采集到各类型的数据存在采集时间差，采集数据不全和重复等问题，必须经过数据清洗和规整才能用于算法分析. 对于上述情况，本文采用固定时间窗口内数据对齐，重复项数据取均值、去极值、漏项数据以上条同类数据补齐的方式来达到清洗和规整的要求.

由于采集到的指标项很多，不是每种指标项都对预测 Web 服务响应时间都有作用. 本文通过使用 Pearson 相关系数的方法把各指标项与 Web 服务响应时间做相关性分析，去除相关性很小的指标项. 为提高训练效率，本文把保留下来的指标项用 PCA 算法做降维操作，然后使用 LSTM 算法进行训练和预测.

3.2　基于多层 LSTM 的 web 服务响应时间预测模型

本文构建的 LSTM 预测模型如图 3 所示. 预测模型由输入层、隐藏层、输出层、网络训练和网络预测等 5 个模块组成.

输入层负责把经 PCA 降维的指标矩阵和对应的 Web 服务响应时间序列一起划分为训练集和测试集，并做标准化处理，然后把数据分割送入下层处理. 隐藏层采用多层 LSTM 细胞搭建循环神经网络. 输出层输出模型和预测结果. 网络训练是数据分为多个批次，每次读取一批数据进行反向传播算法训练. 重复进行传播和权重更新(weight update)，直到误差(loss)收敛.

本文采用均方根误差(RMSE)[10]，均方根误差是均方误差的算术平方根，可表示为式(11).

$$RMSE=\sqrt{\frac{1}{N}\sum_{t=1}^{N}(o_t-p_t)^2} \tag{11}$$

其中，N 为预测结果个数；o 为真实值；p 为预测值. 采用 adam[11] 的随机梯度下降作优化.

3.3　多层 LSTM 参数评估算法

参数的选择对 LSTM 的预测效果有很大的影响，本文设计了一套评估算法，以 RMSE 值和算法

运行时间为主要衡量指标,比较各参数对预测的影响.

算法描述为:将数据集(data)按指标项划分成输入项(T)和预测项(P). 然后执行以下步骤.

1) 第一层循环将输入项(T)进行降维. 将 T 的维数(T. length)每次循环降一维直到降维后数据信息(T. information)小于原始信息的 Information.

2) 第二层循环隐藏层从一层开始每次循环将 LSTM 隐藏层(LSTM hide)加一层,直到隐藏层数(LSTM hide. length)为输入的循环结束层数(hide 值).

3) 第三层循环将 LSTM 隐藏层神经元个数从 n 个开始每次循环增加 s 个直到 max 个神经元.

4) 第四层循环将 LSTM 窗口大小从 1 增加到 batch_size.

循环完成输出每种参数组合下 LSTM 模型在各个参数的运行的时间和 RMSE 值. 算法伪代码描述如算法 1.

算法 1 多层 LSTM 参数评估算法

输入: hide, information, n, max, s, batch_size

输出: norm, result

Begin

1) Get T, P from data

2) For$T$. information>information

3) T_d =PCA(T)

4) For LSTM $_{\text{hide}}$ in 1 : hide

5) For LSTM $_{\text{neuron}}$. number in n: max by step s

6) For LSTM $_{\text{batch}}$ in 1:batch_size

7) Append norm with LSTM $_{\text{hide}}$, LSTM$_{\text{neuron}}$, T_d. length

8) StartTime=Time. clock()

9) RMSE = Execute LSTM $_{\text{predict}}$(LSTM $_{\text{hide}}$, LSTM$_{\text{neuron}}$, batch_size, T_d, P)

10) TimeEnd = Time. clock()

11) Append result with RMSE, TimeEnd-StartTime

12) end(for);end(for);end(for);end(for)

13) Return norm, result

End.

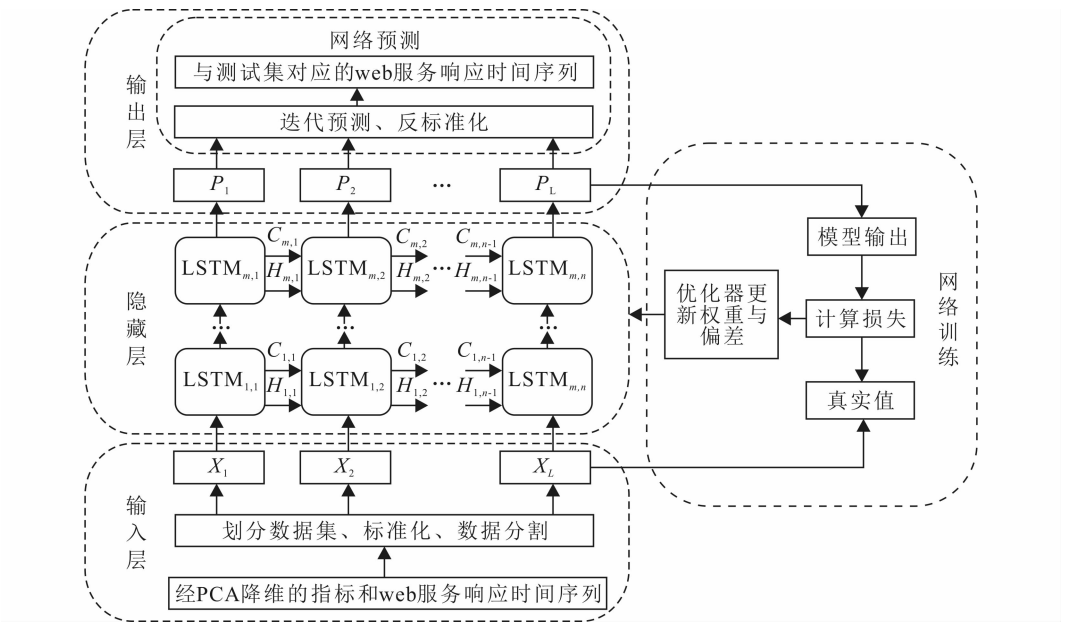


图 3 基于多层 LSTM 的 Web 服务响应时间预测模型

Fig. 3 Prediction model of Web service response time based on multitier LSTM

4 实验验证

4.1 实验系统配置

实验配置情况见表 1,使用三台 Linux 服务器做 Spark 和 HBase 集群,配置 Spark 版本为 2.1.1. Spark 中还包含一个提供常见的机器学习(ML)功能的程序库 MLlib[12, 13]. MLlib 提供了很多种机器学习算法,包括分类、回归、聚类、协同过滤等.

表 1　系统配置表

Tab. 1　System configuration table

节点名称	CPU	内存	硬盘
Spark master	8 核	64 G	2 T
Spark worker1	8 核	32 G	2 T
Spark worker2	8 核	32 G	2 T

4.2　数据来源与预处理

本文实验数据来自某单位 OA 办公系统运行数据. OA 系统采用 Linux 操作系统，应用中间件为 Tomcat，数据库中间件为 Oracle. 通过 Agent 方式采集到 48 项各类运行指标项. 数据采集时间间隔为 20 s，采集时间范围为 6 个月，总数据量约 3800 万条. 利用式(1)计算每个指标项与服务响应时间的相似度. 根据计算结果去除 JVM 内存最大值、磁盘使用率、Oracle 自由空间碎片索引比值等与 Web 服务响应时间相关度较低的 33 个指标项数据. 由于保留的 Java 堆内存使用量、Tomcat 当前活动会话数、Tomcat 总请求数等 15 个指标项中处在某些项之间相似度比较高的情况，所以本文采用 PCA 降维的方法来取得特征矩阵，减少 LSTM 算法的计算数据量，提高计算效率.

4.3　LSTM 预测与参数调优

本文对 PCA 输出维数、LSTM 神经元个数、LSTM 预测步长、LSTM 隐藏层层数做了调整，并做了对比试验，其中使用 70%样本做训练，30%样本做测试，以下是实验结果：

图 4 展示了是否经过相似度计算对预测的影响. 图中 raw data 指使用所有指标项来预测. Post processing data 指除掉相似度低的指标项后的预测效果.

图 5～图 9 展示了降维后使用维数、神经元个数、预测步长、窗口大小、隐藏层数等单个参数变化时损失变化的情况. 图中横坐标为循环次数，纵坐标为损失值. 图 5 中 dimension 值为保留维度；图 6 中 neuron 为神经元个数；图 7 中 predict1、predict2、predict3 分别指预测步长 1、2、3 步；图 8 中 batch_size 指窗口大小；图 9 中 hidden1、hidden2、hidden3 分别指隐藏层数为 1、2、3.

图 10 展示了窗口大小和神经元个数变化时 RMSE 值的变化. 图 11 展示了隐藏层和窗口大小变化时 RMSE 值的变化.

4.4　LSTM 与其它预测方法对比

本文将 LSTM 与 BP 神经网络、ARIMA 时间序列预测使用同样样本进行预测效果对比. 本文实验中使用的 BP、ARIMA 算法均未做改进或优化，只是为了比较在本文场景下基础算法对对 Web 服务器响应时间的适用性，以下为结果.

BP 模型采用两个隐藏层，第一层 50 个神经元，第二层 10 个神经元. LSTM 模型用一个隐藏层 500 个神经元. 两个模型的窗口大小为 6 循环次数 50 次. 图 12 展示了 BP 与 LSTM 损失变化对比. 图 13 展示了三种算法的预测效果对比.

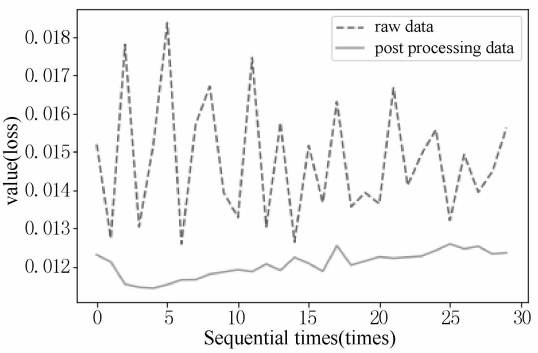


图 4　是否去除相似度低的指标项的损失变化

Fig. 4　Loss changes before and after similarity filtering

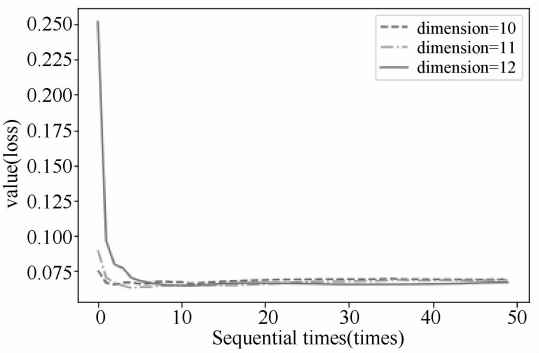


图 5　不同神经元个数的损失变化

Fig. 5　Loss changes of different dimensionality preserving numbers

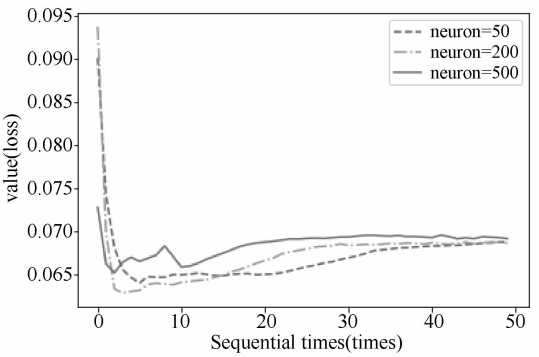


图 6　不同神经元个数的损失变化

Fig. 6　Loss of number of different neurons

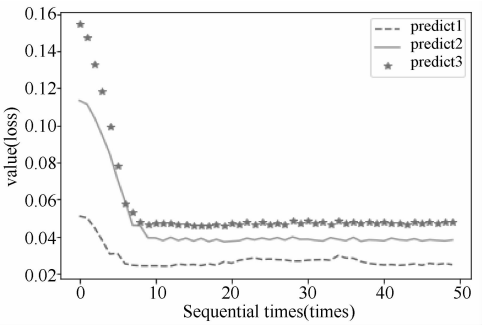


图 7 不同预测步长的损失变化

Fig. 7 Loss variation of different prediction step sizes

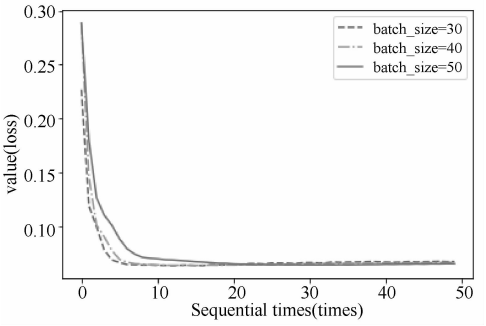


图 8 不同窗口大小的损失变化

Fig. 8 Loss changes of different window sizes

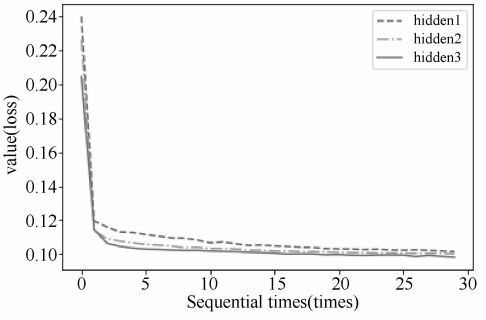


图 9 不同隐藏层个数的损失变化

Fig. 9 Loss variation of number of hidden layers

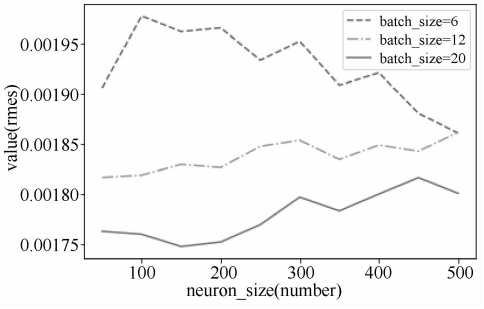


图 10 窗口大小和神经元个数变化时 RMSE 值的变化

Fig. 10 The change of RMSE value when the window size and number of neurons change

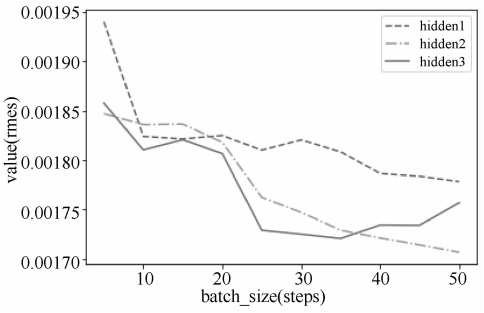


图 11 隐藏层和窗口大小变化时 RMSE 值的变化

Fig. 11 Change of RMSE value when hidden layer and window size change

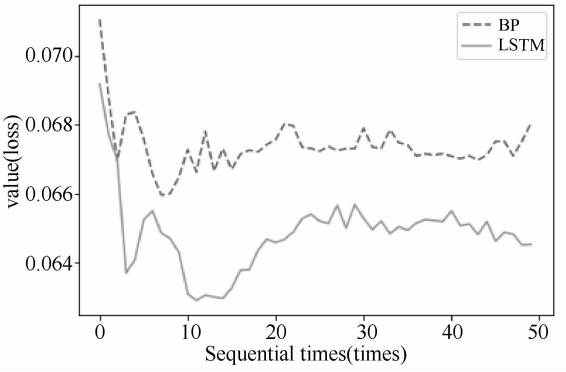
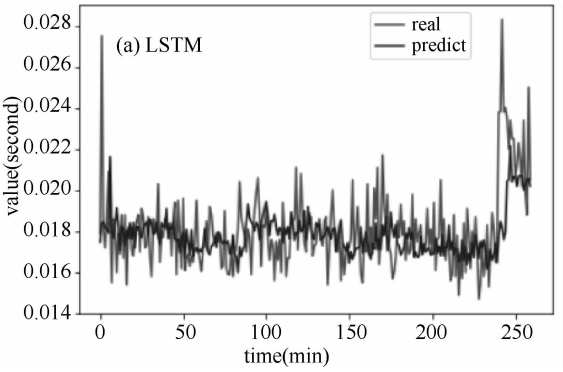
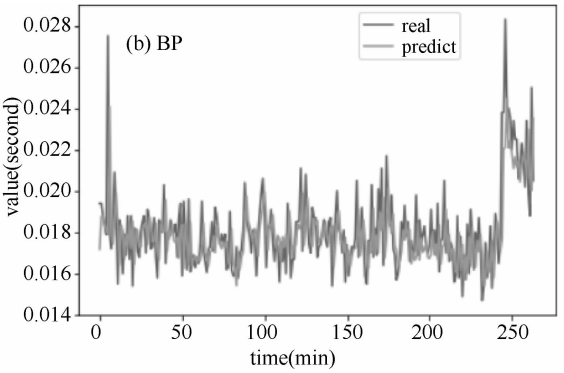


图 12 BP 与 LSTM 损失变化

Fig. 12 Loss of BP and LSTM





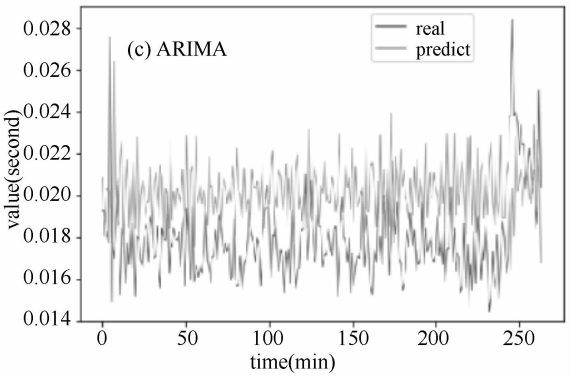


图 13　几种算法的 48 小时预测效果对比

Fig. 13　Comparison of 48 hours prediction effect of several algorithms

5　结　论

本文提出一种顺序使用相关性、特征降维、多层 LSTM 等算法对 Web 服务响应时间做预测的方法，并通过多层 LSTM 参数评估算法来做参数优选. 实验结果表明，1）通过相关度分析和 PCA 降维能在预测精度降低很小的情况下提高预测速度；2）隐藏层的数目对预测结果影响不大，用两个隐藏层能对本文实验样本能取得较好的预测效果；3）窗口的大小对预测结果影响较大，窗口大小为 30 至 40 能取得比较好的预测效果；4）神经元个数越多，损失函数收敛地越快，但对最终的预测效果影响不大；5）在本文应用场景下，使用 LSTM 算法较 BP 和 ARIMA 能更好的做出预测.

总体来说，本文提出的预测方法能有效用于对 Web 服务响应时间的预测. 基于本文的预测方法，可做进一步的优化. 可以考虑对 Web 服务响应时间抖动剧烈的时间段做局部参数调优，来减少局部预测误差.

参考文献：

[1]　刘雪梅. 服务器端软件性能分析和诊断方法研究[D]. 哈尔滨：哈尔滨工程大学，2010.

[2]　郑晓霞，赵俊峰，程志文，等. 一种 WebService 响应时间的动态预测方法 [J]. 小型微型计算机系统，2011，32：1570.

[3]　陈文文. 基于 BP 神经网络的云服务响应时间预测方法研究 [D]. 沈阳：东北大学，2013.

[4]　Zaremba W，Sutskever I，Vinyals O. Recurrent neural network regularization [J]. Eprint Arxiv，2014，5：1.

[5]　Hocheriter S，Schmidhuber J. Long Short-term memory [J]. Neural Computation，1997，9：1735.

[6]　Gers F A，Schmidhuber J，Cummins F. Learning to forget：continual prediction with LSTM [J]. Neural Comput，2000，12：2451.

[7]　Benesty J，Chen J，Huang Y，*et al*. Pearson correlation coefficient [M]// Noise Reduction in Speech Processing. Berlin，Germany：Springer Berlin Heidelberg，2009.

[8]　吴晓婷，闫德勤. 数据降维方法分析与研究 [J]. 计算机应用研究，2009，26：2832.

[9]　高云龙，左万利，王英，等. 基于集成神经网络的短文本分类模型 [J]. 吉林大学学报：理学版，2018，56：933.

[10]　Willmott C J，Matsuura K. Advantages of the mean absolute error (MAE) over the Root mean square error (RMSE) in assessing average model performance [J]. Climate Res，2005，30：79.

[11]　Kingma D P，Ba J. Adam：a method for stochastic optimization [J]. Comput Sci，2014，9：1.

[12]　Zaharia M，Chowdhury M，Franklin M J，*et al*. Spark：cluster computing with working sets [C]// Usenix Conference on Hot Topics in Cloud Computing. [s. l.]：USENIX Association，2010.

[13]　Meng X，Bradley J，Yavuz B，*et al*. MLlib：machine learning in apache spark [J]. J Mach Learn Res，2015，17：1235.

引用本文格式：

中　文：刘承启，林振荣，黄文海. 基于 LSTM 的 WEB 服务响应时间大数据预测方法[J]. 四川大学学报：自然科学版，2019，56：71.

英　文：Liu C Q，Lin Z，Huang W H. Big data prediction method of WEB service response time based on LSTM [J]. J Sichuan Univ：Nat Sci Ed，2019，56：71.