

一种基于深度学习日志分析模型的压力测试算法

卢璐, 杨子江

东方证券股份有限公司, 上海

收稿日期: 2023年4月25日; 录用日期: 2023年5月24日; 发布日期: 2023年5月31日

摘要

软件新系统上线前, 常常需要对生产运行的系统开展压力测试, 以规避系统因突发巨量客户端访问或生产过程中多用户集中操作, 而出现报错及系统崩溃等严重问题。本文提出了一种基于深度学习日志分析模型的压力测试算法, 该算法基于现有生产日志数据创建双向长短期记忆循环神经网络(Bi-LSTM)的分析模型, 搭建并预测基于生产日志规律的闭环测试场景, 模拟预测生产环境的理论测试场景, 作为传统基于规则压力测试方法的补充。实验结果表明, 使用本文所提双向LSTM分析模型可以有效地模拟生产日志的规律, 使得压力测试更贴近真实场景, 能有效观察系统运行状况, 多维度了解系统性能的极限和缺陷。

关键词

深度学习, 日志分析, 压力测试, 双向长短期记忆循环神经网络

A Stress Testing Algorithm Based on Deep Learning Log Analysis Model

Lu Lu, Zijiang Yang

Orient Securities Company Limited, Shanghai

Received: Apr. 25th, 2023; accepted: May 24th, 2023; published: May 31st, 2023

Abstract

Before launching a new software system, stress testing on the production-environment system is necessary. It could avoid serious problems such as system errors and system crashes caused by sudden massive client access or concurrent operations by multiple users during the production

process. This paper proposes a deep learning log analysis model-based stress testing algorithm. The algorithm first creates a bi-directional long and short-term memory recurrent neural network (Bi-LSTM) analysis model based on existing production log data, then builds and predicts a closed-loop test scenario based on production log patterns, and finally simulates a theoretical test scenario of the production environment. This method can be served as a supplement to the traditional rule-based stress testing method. The experimental results show that using the proposed bi-directional LSTM analysis model in this paper can effectively simulate the pattern of production logs, which makes the stress testing closer to the real scenario. In this case, developers can effectively observe the system operation condition and understand the limits and defects of the system performance in multiple dimensions.

Keywords

Deep Learning, Log Analysis, Stress Testing, Bi-Directional Long Short-Term Memory Recurrent Neural Networks

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来,我国资本市场高速发展,证券业务快速扩容导致相关系统生产规模急剧增长,与之配套的软件系统稳定性要求越来越高。证券系统软件由于其支持多券商、多资金账户、高并发业务等特点,保持其不间断稳定运行的是一项最基本的功能保证,这需要在上线前进行大量的测试工作,压力测试是其中一个重要的环节。软件压力测试可以给软件不断加压,再计算机数量较少或系统资源匮乏的条件下运行,观察其可以运行到何种程度,从而发现性能缺陷。通过搭建与实际环境相似的测试环境,通过测试程序或某一段时间内,测试系统在不同压力情况下的效率状况,以及系统可以承受的压力情况。压力测试可以评估系统在实际使用环境下的效率情况,评价系统性能以及判断是否需要应用系统进行优化处理或结构调整。证券系统软件由于其特殊使用环境,压力测试的数据指标将直接影响用户的体验,其重要性更加突出。

目前,软件压力测试规则是基于存量数据的规模,业务内容,操作用户的数量、各类操作用户的比例,峰值业务量的大小,实时业务响应时间要求,以及批量处理过程的时间要求等作为测试依据。传统的测试方式需要人工调研分析后手动编写测试案例及测试脚本,使用脚本进行测试,最后分析资源消耗情况并进行优化[1]。

证券系统由于其庞大的系统交易数量以及复杂的业务逻辑,传统的人工设计测试案例难免存在设计不合理、考虑不完整等先天缺陷,常常导致误报、漏报等情况出现。为了追求更好的测试效果,压力测试场景力求尽可能真实模拟实际业务场景[2]。中国工商银行及中国农业银行都通过使用实际生产的业务日志数据作为测试案例,测试的覆盖面更广、更真实、也更符合实际业务需求,解决了人工设计案例不完整的问题[3][4]。但该方法存在2个典型缺陷:一是在多用户大量并发使用的情况下,业务日志的记录顺序具有随机性,为了保证测试的完整性,需要使用大量业务日志进行测试,会导致测试耗时特别长;二是其测试方式无法从繁杂的业务日志中提取出有序的业务数据顺序规律,这些规律是开发及运维人员参考并进行针对性的优化的依据。

为了保证系统的安全运行, 几乎所有系统都会使用日志来记录运行时的状态和详情消息, 包括接收的请求、处理后返回的响应等业务消息, 以及网络访问、系统错误与警告等系统信息[5][6][7]。近年来, 如何在统计分析以外提取更多日志的隐藏信息, 受到了越来越多的关注, 其主要日志信息提取方向是对异常的检测、定位和预警[8][9][10][11], 而对日志的顺序规律的研究和分析却尚不存在。

基于此, 本文提出了一种新的应用于压力测试的日志分析算法, 该算法引入深度学习机制, 能够更好地学习业务日志的内在规律, 并输出模拟的业务数据提供给压力测试。首先, 该算法从业务日志中提取出的各业务请求名称及对应时间数据, 调整相邻请求的时间间隔生成时间序列, 并采用独热编码把无序离散的业务请求名称转换为有序连续的向量矩阵, 方便机器学习理解。然后, 使用双向长短期记忆循环神经网络的深度学习算法, 多次调整参数以获取预测准确率最高的参数组。最后, 使用最优参数组学习大量业务日志, 生成业务请求预测时间序列, 可作为测试案例进行压力测试。

本文组织如下: 在第二节中, 本文将介绍软件开发与测试的一般流程, 并指出其流程中的问题与缺陷。第三节用于介绍本文用到的相关理论以及压力测试算法框架的设计, 并在第四节阐述整个实验的过程。最后, 在第五节给出结论。

2. 软件开发与软件测试

通常一个系统开发由分析、设计、编程(写代码)、测试、维护五个部分组成。其中分析指的是确定问题、结构化分析、建立逻辑模型、编写设计说明书。设计指的是概要设计(模块结构)、详细设计(程序流程、算法、数据结构)。编程指的是使用开发语言把软件设计转化为计算机课接受的程序。测试指的是设计测试用例(以较小的代价发现尽可能多的错误)、白盒法和黑盒法。维护指的是对软件产品进行的软件工程活动, 纠正运行中发现的错误, 对软件适当修改以适应新要求。

从图 1 可以发现, 软件开发会根据测试结果的正确与否来决定结果的走向。如果测试结果正确, 就可以将软件上线维护, 反之, 就会返回编程和设计。当上线维护发现错误或报警, 就会将结果反馈至测试, 待测试得出结果后再行决定流程的走向。目前传统测试方法主要采用人工手动执行一系列操作所对应的请求及响应, 所有测试以及设计的测试案例大多都基于单一用户无干扰的情况产生。但是, 在实际的上线运行中, 常常遇到在极短时间内有大量用户并发同时进行相同操作的情形, 这些请求的执行指令是一个天文数字, 是人力测试所无法覆盖的。因此只能按上线结果反馈来主导测试的被动操作过程。这种方法也是目前主流的测试方法, 很明显这样的测试方法是无法杜绝上线运行后不会出现错误或报警现象的发生。

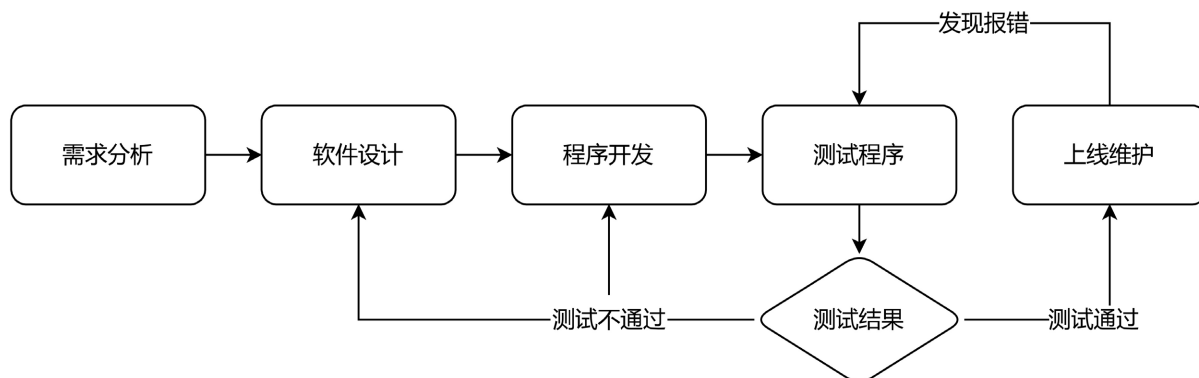


Figure 1. Diagram of the software development process

图 1. 软件开发流程示意图

3. 基于深度学习日志分析模型的压力测试建模

3.1. 相关理论

独热编码(One-Hot Encoding), 又名一位有效编码, 其方法是使用 N 位状态寄存器来对 N 个状态进行编码[3], 因此每个状态都有它独立的寄存器位, 并且在任意时候, 其中只有一位有效。独热编码的优点是解决了不好处理的属性数据的问题。在机器学习算法中, 无法直接学习非连续的离散数据, 而独热编码可以通过把特征转换为二进制的形式, 也能保证特征之间的距离计算更加合理, 方便进行后续的机器学习与训练。

双向长短期记忆循环神经网络(Bi-Directional LSTM RNN)是循环神经网络(RNN)的一种。Bi-LSTM 通过输入序列分别以正序和逆序输入至 2 个神经网络进行特征提取的方式, 使 t 时刻所获得特征数据同时拥有过去和将来之间的信息, 从而能够进行更精准的预测(图 2)。

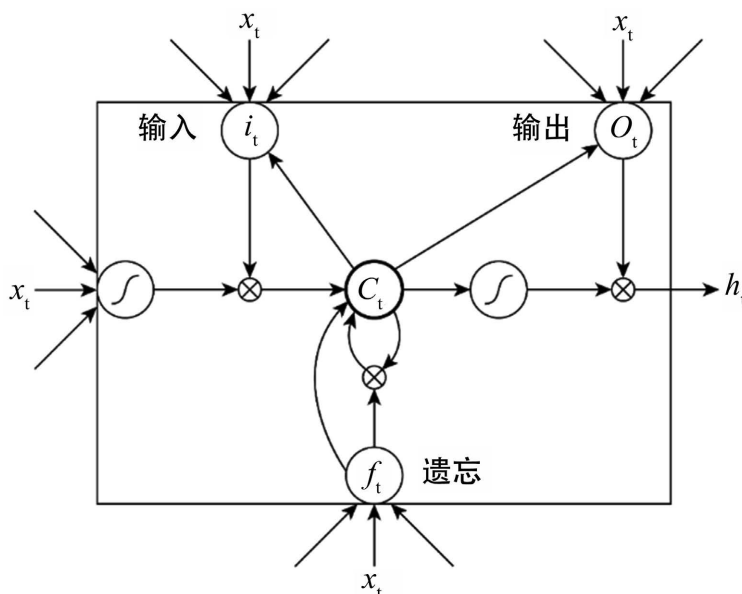


Figure 2. Diagram of a long-short term memory module
图 2. 长短时记忆模块示意图

循环神经网络(RNN)是时间上的展开, 在工作时一个重要的优点在于, 其能够在输入和输出序列之间的映射过程中利用上下文相关信息, 因此主要用于处理以时间序列数据作为输入的预测问题。然而, 标准的循环神经网络(RNN)能够存取的上下文信息范围很有限。这个问题就使得隐含层的输入对于网络输出的影响随着网络环路的不断递归而衰退。因此, 为了解决这个问题, 长短时记忆(LSTM)结构诞生了。与其说长短时记忆是一种循环神经网络, 倒不如说是一个加强版的组件被放在了循环神经网络中。具体地说, 就是把循环神经网络中隐含层的小圆圈换成长短时记忆的模块。这个模块的样子如图 3 所示。

如图 3 所示为相较于传统循环神经网络的双向循环神经网络(BRNN)在时间上展开, 图中展示的是一个沿着时间展开的双向循环神经网络。双向循环神经网络的基本思想是提出每一个训练序列向前和向后分别是两个循环神经网络, 而且这两个都连接着一个输出层。这个结构提供给输出层输入序列中每一个点的完整的过去和未来的上下文信息。图中, 六个独特的权值在每一个时步被重复的利用, 六个权值分别对应: 输入到向前和向后隐含层(w_1, w_3), 隐含层到隐含层自己(w_2, w_5), 向前和向后隐含层到输出层(w_4, w_6)。

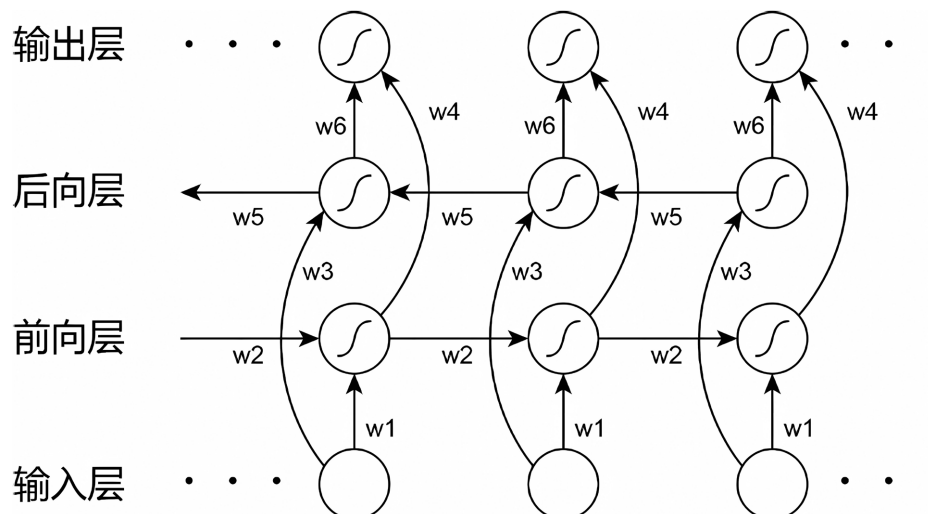


Figure 3. Diagram of the bidirectional recurrent neural network unfolded in time
图 3. 双向循环神经网络在时间上展开示意图

3.2. 基于生产日志模型的压力测试算法框架

与其等待错误的发生, 不如主动出击去寻找错误。目前, 开发测试时所使用的测试数据是在开发过程中设想的, 在各项测试达标后才会逐步进入生产并上线使用, 最终产生生产日志。也就是说, 测试数据与实际生产日志无关。生产日志分析模型的建立就是想要将生产日志中的信息, 应用在测试中, 使测试数据尽可能的接近现实情况。因此, 生产日志建模则会通过对生产日志进行归纳总结, 模拟出这一系列操作的顺序, 方便更好的覆盖这些运行环境, 以便更好的测试。这样就可以把被动的测试转化为主动的测试。从图 4 的流程示意图可以看出, 生产日志模块只需从上线维护获取一段时间内的生产日志数据, 便可将所建的生产日志模型提供给测试单元进行多维度的全方位的测试, 从而避免了单一状况的发生, 实现了主动测试的功能。

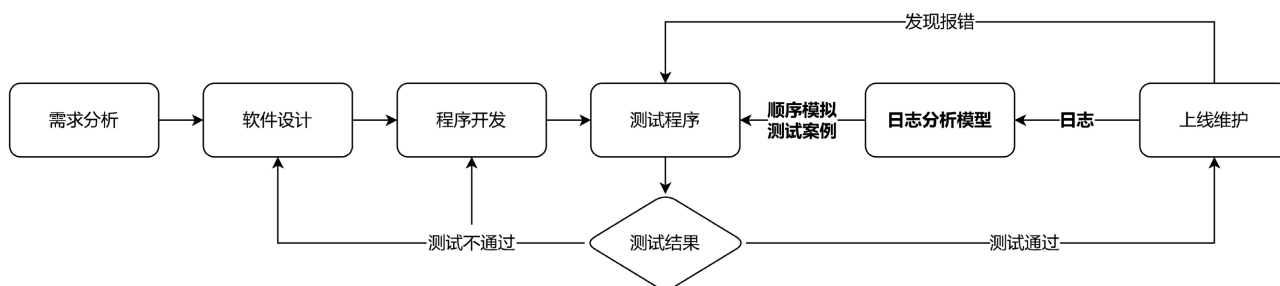


Figure 4. Diagram of the position of log analysis model in software development process
图 4. 日志分析模型在软件开发流程位置示意图

如图 5 所示, 基于双向长短期记忆循环神经网络模型算法系统首先构建一套初始模型并进行训练。系统可直接导入模型并产生第一批生产日志, 将其作为新的模型训练日志输入给基于双向长短期记忆循环神经网络模型算法。模型输出结果后可根据结果生成新的生产日志对后时序的压力测试状态进行预测, 并可与对应的真实系统生产日志进行对比获得准确度。如果准确度符合要求, 则可基于该预测结果产生的生产日志继续导入模型输出结果实现自闭环的压力测试状况预测。如果输出结果与实际生产日志结果差距较大, 则可进入关键参数修改循环, 修改后重新训练系统模型进入循环。

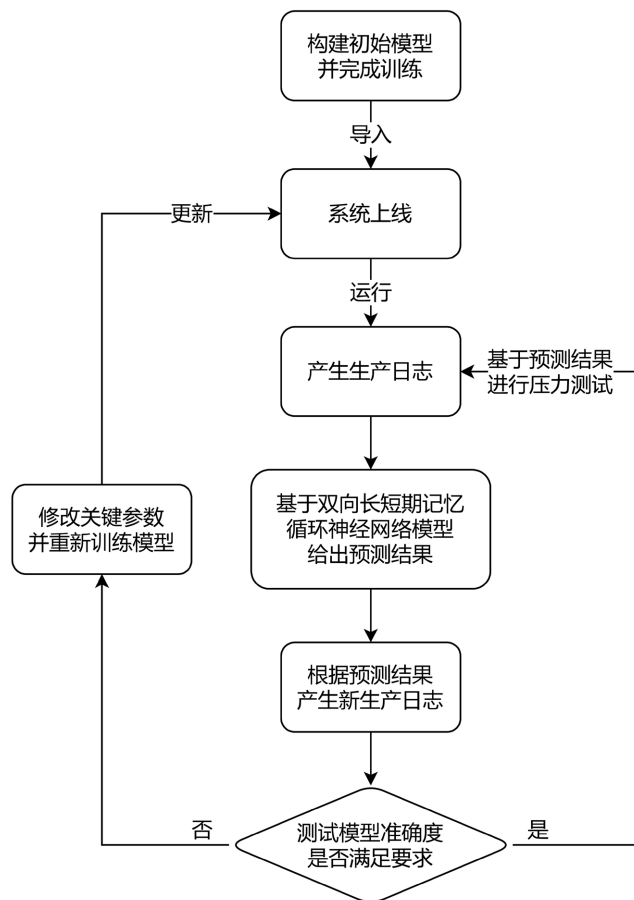


Figure 5. Diagram of the position of log analysis model in software development process

图 5. 基于双向长短期记忆循环神经网络模型算法系统流程

4. 实验

4.1. 原始日志预处理

交易系统网关日志的内容包括各请求及响应的业务名称、发生时间、内容的字节长度、处理请求的耗时等业务信息, 以及系统目前是否正常运行, 及故障的现象、原因及解决过程等系统信息。以某一天的日志为例, 共包含 26 万条业务记录和 23 万条系统记录。

日志预处理是从日志中提取出供 Bi-LSTM 算法使用的规范化时间序列数据, 主要分为特征提取和时间序列转换 2 步。特征提取是从原始的生产日志中提取出各业务请求名称及对应的时间, 例如委托、股票信息查询等请求。由于压力测试需要大量请求, 因而选取了业务请求最密集的交易时间端, 即 9:30~11:30 及 1:00~3:00 内的业务请求。时间序列转换是设立一个合理的等时间间隔的时间序列, 将各业务请求对应至序列中的各个时间点上, 无对应请求的时间点即标为空值。如多个业务请求对应同一个时间点, 则按业务请求顺序顺延对应至后续时间点。不合理的时间间隔, 会导致业务请求积压或过多的空值, 影响后续机器学习训练和预测的准确性。

4.2. 输入输出变量的选择

由于该时间序列的特征变量只有一项, 输入特征变量与输出变量都是业务请求名称, 是离散型特征,

所以使用 one-hot 编码进行处理。在机器学习算法预测时间序列上, 本文使用单变量时间顺序预测问题, 对应使用多对一的序列预测模型, 也就是通过之前 t 个时间点的业务请求预测当前时间点的业务请求。

4.3. 参数选择与模型训练

在使用 Bi-LSTM 算法进行预测的时候, 必须选择适当的参数, 即是否为有状态的 LSTM (Stateful)、神经元数量(Cells)、时间步长(Timestep)、每批数据大小(batch_size)。对于业务请求来说, 前后两个请求是有相关性的, 需要让 LSTM 算法学习到其时序特征, 因而设置 Stateful 为 True。

神经元的数量会影响网络的学习能力, 更多的神经元能够以更长的训练时间为代价从问题中学习更多的结构, 也会产生过度拟合训练数据的问题。在其他条件均相同的情况下, 不同神经元数量与训练集的损失值(train_loss)、测试集的损失值(val_loss)的关系如图 6、图 7 所示, 可以看出, 神经元数量为 8 的时候, train_loss 与 val_loss 最低。

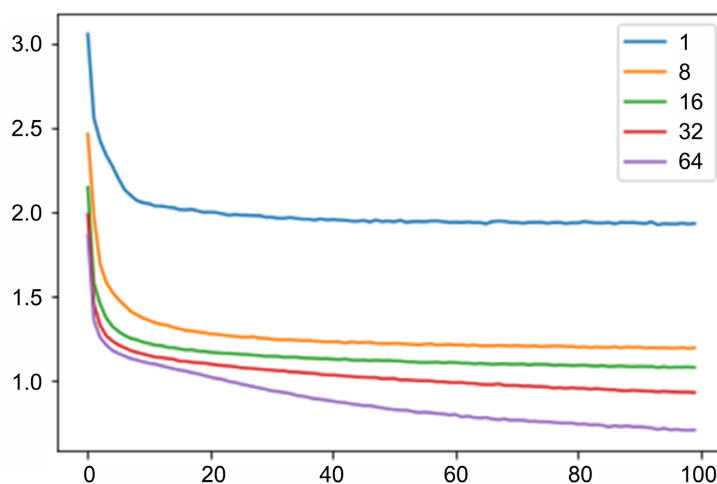


Figure 6. Diagram of the relationship between the number of different neurons and the loss value of the training set

图 6. 不同神经元数量与训练集的损失值的关系

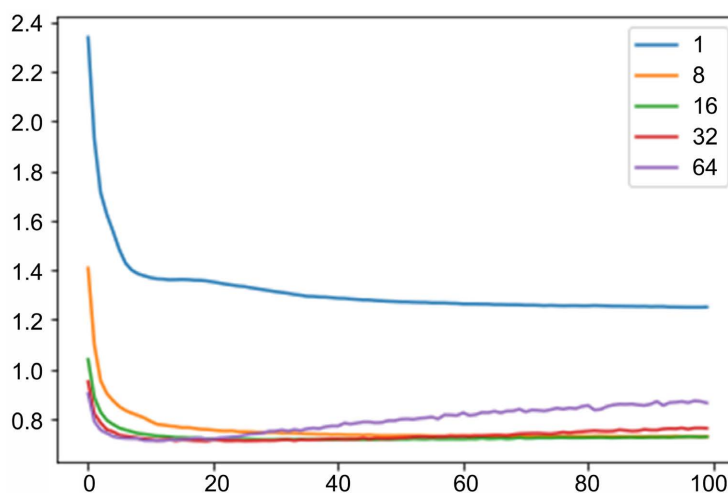


Figure 7. Diagram of the relationship between the number of different neurons and the loss value of the test set

图 7. 不同神经元数量与测试集的损失值的关系

时间步长就是用多少个时间点的参数叠加在一起去预测某个未来时间点的结果, 需要找到一个最优的时间步长, 其时间步长最短且 train_loss 与 val_loss 最低。在其他条件均相同的情况下, 不同时间步长与 train_loss 与 val_loss 的关系如图 8、图 9 所示。这里为了能够更好地用图表现出不同时间步长的差异, 截取了第 15 到 200 次的训练结果。可以看出, 时间步长为 10 的时候, train_loss 与 val_loss 最低。

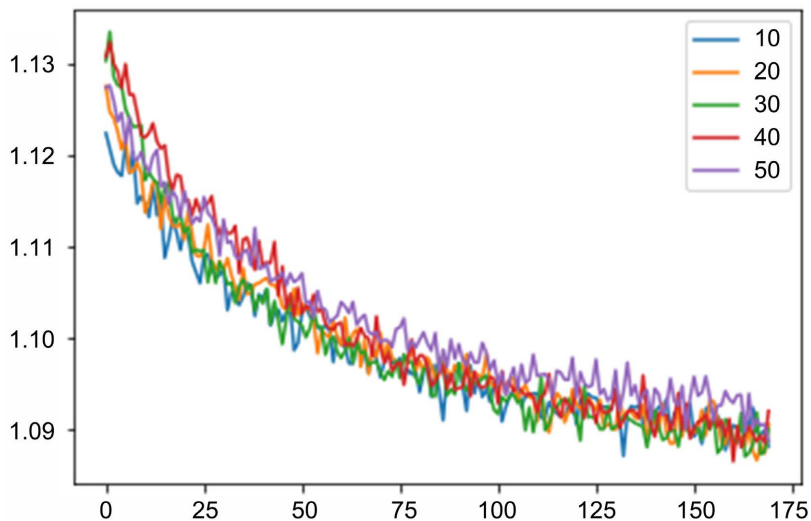


Figure 8. Diagram of the relationship between loss values of different time steps and training set

图 8. 不同时间步长与训练集的损失值的关系

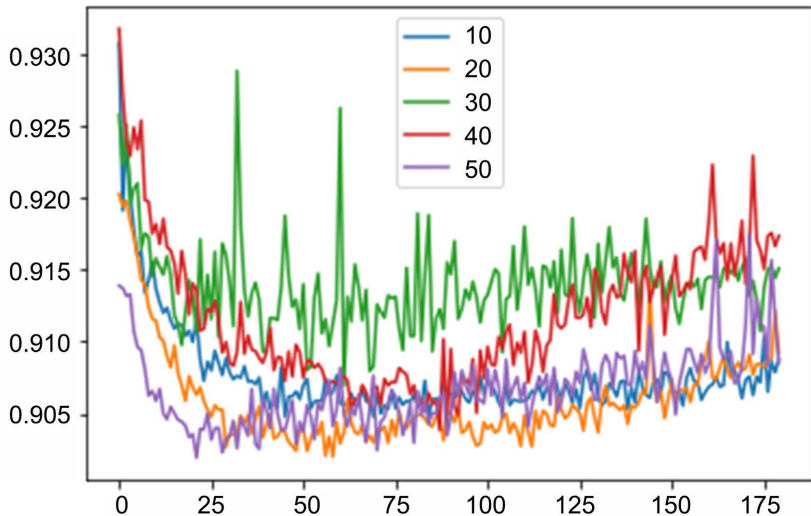


Figure 9. Diagram of the relationship between loss values of different time steps and test set

图 9. 不同时间步长与测试集的损失值的关系

4.4. 评价指标

本文采用准确率(P)作为评价指标, 即被算法预测正确的日志占总业务日志的百分比, 计算方法如下:

$$P = \frac{TP}{TP + FP} \tag{1}$$

其中 TP 为预测值与实际值相同数量, FP 为预测值与实际值不同的数量。

由于日志中存在多个业务数据并发, 即同一时间的业务请求在日志中有排序上的差异的情况。为了忽略并行请求带来的顺序影响, 统计每一段业务请求组中各业务请求实际出现值与预测值一致的数量之和占所有业务日志总数的百分比, 即命中率。

4.5. 结果分析

本文采用某个周三交易时间段内业务请求数共 100,411 条, 对应到 144,000 等间隔时间序列, 其中训练集、验证集、测试集的比例为 5:3:2。日志经处理后, 分别使用自回归移动平均模型 ARIMA、LSTM 算法、Bi-LSTM 算法进行训练, 其结果如表 1 所示。

Table 1. Analysis results of business request logs

表 1. 业务请求日志分析结果

算法	准确度	命中率
ARIMA	32%	33%
LSTM	68%	69%
Bi-LSTM	79%	82%

从表 1 可以看出, 双向 LSTM 的准确度和命中率明显高于 ARIMA 和 LSTM 方法。在业务日志预测中, 相较于只能发现正向规律的 ARIMA 和 LSTM, 能够同时发现正向与反向规律的双向 LSTM 可以缩小可能范围, 在预测的准确性上有更大的优势。

本文提出一种基于深度学习日志分析模型的压力测试算法, 通过双向 LSTM 方法能够很好地学习到业务请求的规律, 模拟出更接近真实的业务数据, 从而成为很好的压力测试测试案例。

5. 结论

本方法突破传统的基于规则的文本检测技术, 建立基于双向长短期记忆循环神经网络的分析模型技术与实际生产日志相结合, 模拟人脑的神经网络进行分析学习解析数据的算法, 由于具有试图模仿大脑的神经元之间传递、处理信息的模式, 深度学习可以组合多个低层特征以形成更抽象的高层特征, 有更好的学习样本数据的内在规律和表示层次能力。因此能尽可能真实准确地模拟推算出未来生产中实际运行的指令发送情况, 实现在压力测试环节中, 更加真实体现实际场景的要求, 也为后续的压力测试案例编排提供数据模型支撑, 达到提升系统安全性、稳定性、敏捷性的目的。

参考文献

- [1] 王宏. 基于最小业务单元的 Web 系统性能测试调优[J]. 信息技术与信息化, 2020(1): 25-27.
- [2] 朱靖君, 吴海燕, 高国柱, 程志锐. 一种基于日志分析的 Web 负载测试方法[J]. 计算机工程, 2010, 36(23): 25-27.
- [3] 张世瑛, 梁晔华. 商业银行大数据性能容量测试探索与实践[J]. 中国金融电脑, 2022(1): 91-94.
- [4] 冯瑶, 孙骏亚, 刘跃光, 王智忠. 基于交易日志的自动化测试技术在功能测试中的应用[J]. 中国金融电脑, 2018(11): 55-56.
- [5] 钟雅, 郭渊博. 基于机器学习的日志解析系统设计与实现[J]. 计算机应用, 2018, 38(2): 352-356.
- [6] 闫力, 夏伟. 基于机器学习的日志异常检测综述[J]. 计算机系统应用, 2022, 31(9): 57-69.
- [7] 尹慧, 花嵘, 郭宁, 尹韬. 基于深度学习的分布式安全日志分析方法[J]. 软件, 2020, 41(3): 272-277.
- [8] 张忠浩, 张磊, 肖莎, 赵鹏, 雷大洋, 刘赛. 基于深度学习的数据中心日志分析方法设计与实现[J]. 电力信息与通信技术, 2021, 19(11): 39-43. <https://doi.org/10.16543/j.2095-641x.electric.power.ict.2021.11.006>

- [9] 杨朝鹏, 林业贵, 罗飞鹏. 基于日志的机器学习方法实现故障快速定界的研究与应用[J]. 邮电设计技术, 2018(12): 23-26.
- [10] 熊逸文, 项菲, 苗杰. 基于机器学习算法匹配的互联网企业内容日志检测工具的设计与实现[J]. 江苏通信, 2022, 38(3): 97-100+124.
- [11] 黄文博, 王祎珺, 陈佩佩, 汪立欣. 一种基于机器学习的日志预警发现方法[J]. 中国科技信息, 2019(19): 78-79.