ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.15, No.19,July. 2019
E-mail：jslt@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

基于切片GRU的日志异常检测研究

王易东[1],王彬[2]

(1.中国海洋大学信息科学与工程学院,山东 青岛 266100;2.中国海洋大学继续教育学院,山东 青岛 266100)

摘要:深度学习应用在日志异常检测领域中,取得了较好的准确率。日志异常检测对实时性的要求很高,这对异常检测模型的运行速度提出了巨大挑战,目前基于循环神经网络的检测算因其固有的循环结构导致并行计算难以实现,这极大限制了模型的计算速度。为提高日志异常检测效率,设计了基于切片GRU日志异常检测模型,提出了一种基于最小质因数切片的日志异常检测算法。构建的切片GRU模型主要对隐藏层做了改进,输入序列在进入隐藏层之前进行了切片,每一层的GRU块按照切片数量进行分组,组与组之间并行计算,层与层之间串行计算,多个隐藏层共同处理输入序列,切片GRU模型的这种层次结构加快了模型的运行速度,在不影响检测精度的前提下,算法在不同长度的日志序列上均有不效率提升,且输入序列越长,提升的幅度越大。

关键词:日志异常检测;深度学习;循环神经网络;GRU网络;切片算法

中图分类号:TP319 文献标识码:A

文章编号:1009-3044(2019)19-0059-02

DOI:10.14004/j.cnki.ckt.2019.2446

开放科学(资源服务)标识码(OSID):

1 引言

日志异常检测可以看作一种自然语言序列预测任务,通过对比预测值与实际值来判定异常。循环神经网络(recurrent neural network ,RNN)通过一个循环结构,将上一种状态的输出作为当前的输入,从而追踪历史,做出预测,可以用于自然语言序列预测任务。由于梯度消失/梯度爆炸的问题,传统RNN在实际中很难处理长期依赖,而长短期记忆网络(Long Short-Term Memory,LSTM)和门控循环单元(Gated Recurrent Unit,GRU)则解决了这个问题,可以从语料中学习到长期依赖关系。Du M[1]等将系统日志建模为自然语言序列,提出了一种基于LSTM的深度神经网络模型——DeepLog,在多个大型数据集上取得了较高的检测精度。

然而日志异常检测对实时性的要求很高,这对异常检测模型的运行速度提出了巨大挑战。LSTM和GRU作为RNN的变种,其固有的循环结构导致并行计算难以实现,切片循环神经网络[2](Sliced Recurrent Neural Networks,SRNN)从结构上对RNN进行改进,通过将输入序列切分为多个子序列实现并行计算。本文参照Du M等[1]提出的Deeplog模型构造日志异常检测模型,用GRU网络替换Deeplog中的LSTM,将SRNN的思路应用到日志异常检测模型中,提出了一种基于最小质因数切片的日志异常检测算法,基于该算法改进异常检测模型,并结合日志序列的特点,调整模型参数,本文的最后将改进后的切片模型与原始GRU模型做对比,从性能和速度两方面来评价模型的提升。

2 基于GRU的日志异常检测模型介绍

在系统日志中,所有相似日志条目中的公共常量消息叫作log key,可以用来表示日志消息类型。N元语法(N-gram)是NLP领域中一种常用的概率语言模型,可以通过极大似然估计从长度为N的已知log key序列中计算出下一个log key出现的条件概率。本文基于N-gram语言模型,参考了文献[1]中的设计思路,构建了一种基于GRU的日志异常检测模型。

模型总体上可分为输入层(input layer)、嵌入层(embedding layer)、隐藏层(hidden layer)和输出层(output layer),信息沿各层依次传递。图1给出了基于GRU的日志异常检测模型的结构示意图,该模型的运算速度相比于DeepLog约提升了16.7%,同时该模型也是本文切片GRU日志异常检测模型的基础,结合下文中的切片算法,可以进一步大幅提高日志异常检测效率。

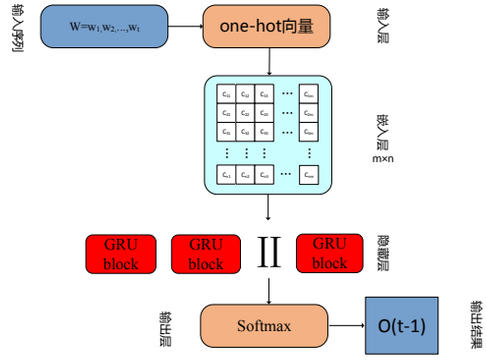

图1 基于GRU的日志异常检测模型

3 基于最小切片因子的切片算法

相比于标准GRU模型,本文构建的切片GRU模型主要对隐藏层做了改进。改进模型与原模型的不同之处在于:输入序列在进入隐藏层之前进行了切片[3],每一层的GRU块按照切片数量进行分组,组与组之间并行计算,层与层之间串行计算,多个隐藏层共同处理输入序列,切片GRU模型的这种层次结构[4]加快了模型的运行速度。

本文对文献[2]中的切片方法做了改进,提出一种基于最小

收稿日期:2019-03-15

切片因子切片算法，和文献[2]中的方法相比，本文算法细化了切片过程，改进了切片方式，具有更强的合理性。算法的核心思想是对输入序列进行多次切片，每次切片数都等于切片序列（子序列）的最小切片因子，直到最小子序列的长度也等于最小切片因子为止。最小切片因子的取值为2、3、5，对于长度等于最小切片因子的序列，无法通过切片加快其计算速度。

假设模型的输入是一个长度为n的log key序列X=$\{x_1,x_2,\cdots x_h\}$，若n是最小切片因子的倍数，直接对序列进行切片，得到$m_1\left(m_1\in\{2,3,5\}\right)$个子序列；若不是，则在序列最后加入一个元素后再切片，该元素仅用来辅助切片，不参与序列中记忆传递。

经过第一轮切片，输入序列X被切分为m_1个子序列，每个子序列长度$l=\frac{n}{m_1}$，第i个子序列可以表示为：

$$Y_i=\left\{x_{(i-1)\times l+1},x_{(i-1)\times l+2},\cdots,x_{i\times l}\right\},\ i\in[0,m_1] \quad (3\text{-}1)$$

若子序列长度l不等于最小切片因子，则继续对其切片。经过k次切片后，得到k+1个隐藏层，其中第0层为开始层，输入最终的切片结果；第k层为结束层，输出隐藏层的最终计算结果。不考虑第k层，每一层的子序列个数为：

$$S_L=\prod_{j=1}^{k-L}m_j,L\in[0,\mathrm{k}-1] \quad (3\text{-}2)$$

其中S_L代表第L层的子序列个数，m_j表示第j轮的切片数。

除第0层外，每一层的子序列长度记作T_L，其计算公式为：

$$T_L=m_{k+1-L},L\in[1,\mathrm{k}] \quad (3\text{-}3)$$

第0层的子序列长度则等于最小切片因子，将其称为最小子序列，每个最小子序列的长度w_0的计算公式如下：

$$T_0=\frac{h}{S_0} \quad (3\text{-}4)$$

其中S_0代表第0层的最小子序列个数，可由公式(3-2)计算得出。

每一层中的单个子序列均由GRU块传递序列信息，子序列之间进行并行计算，随后将计算结果传递到上一层，由上层子序列中的GRU块将序列信息拼接起来，最终获得整个序列的信息。假设每个GRU单元的计算时间为t，则模型的总体计算时间为：

$$\mathrm{T}=\left[\frac{h}{S_0}+m_{k+1-L}\right]t,L\in[1,\mathrm{k}] \quad (3\text{-}5)$$

相较于原模型,计算速度理论上的提升为：

$$\frac{T_{SGRU}}{T_{GRU}}=\frac{1}{s_0}+\frac{m_{k+1-L}}{\mathrm{h}} \quad (3\text{-}6)$$

4 实验及结果分析

为评估切片算法的速度与性能，本节在大型日志数据集上进行了对比实验，实验环境实验环境在个人笔记本上配置，处理器为Intel Core i7-6700HQ(2.60GHz)，NVIDIA GeForce GTX 965M GPU(2GB)，16GB RAM(2133MHz)，操作系统为Ubuntu 16.04(64位)。GRU网络的搭建和训练基于深度学习框架keras，tensorflow作为后端，编程环境为Python3.6.5。对比实验分别从速度和精度两方面对切片模型的优越性进行验证。

首先验证切片模型的速度提升，选取包含11175629条数据的亚马逊大型HDFS日志[5]作为实验数据集，其中前1%日志数据中的正常会话作为训练集训练模型，整个数据集用来测试模型性能。对比实验共分为3组，分别选取长度(h)为6、9、12的log key序列作为输入，对比两个模型的运行速度、检测精度。

表1给出了不同输入长度下两种模型的运行速度对比，从表中可以看出，不同序列长度下的切片GRU模型相较于原模型的速度提升均达到了切片模型速度提升的预期值。观察长度为6和12的序列在不同切片方式下的运行速度，发现不同结构下检测每条日志花费的时间相当，从而证明了序列的不同切片方式并不会影响模型的运算速度。对比不同序列长度下的性能提升幅度，可以看到序列长度越长，切片GRU模型相较于原模型速度提升越大。

表1 不同序列长度下切片模型和原模型运行速度对比

	总时间(ms)	平均时间(ms)	速度提升(%)
原模型(h=6)	5200135	0.465	
切片模型(2,3)	4315359	0.386	17.0
切片模型(3,2)	4309866	0.385	17.0
原模型(h=9)	7860837	0.703	
切片模型(h=9)	5194781	0.465	33.9
原模型(h=12)	10357913	0.926	
切片模型(2,2,3)	6042185	0.541	41.6
切片模型(2,3,2)	6053848	0.542	41.5
切片模型(3,2,2)	6048011	0.541	41.6

5 结语

当前基于深度学习的日志异常检测领域中，涉及提升运算速度的研究相对较少，且大都通过改进循环单元来提升速度，提升幅度较低。本文研究基于并行计算的思想，提出了切片GRU模型，使用分解质因数的思想对GRU模型的输入序列进行切片，多次切片形成了多个隐藏层，每个隐藏层中的切片序列之间并行计算，各隐藏层之间串行计算。在大型日志数据集上进行的切片GRU模型与普通GRU模型的对比实验表明，切片算法对GRU模型的异常检测效率提升显著，在处理较长输入序列时提升尤为明显。本文研究为今后相关工作提供了算法参考和模型构建基准，具有一定理论指导意义。

参考文献:

[1] Du M, Li F, Zheng G, et al. Deeplog: Anomaly detection and diagnosis from system logs through deep learning[C]. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security.ACM,2017:1285-1298.

[2] Yu Z, Liu G.Sliced Recurrent Neural Networks[J].2018.

[3] Oord A, Dieleman S, Zen H, et al. Wavenet: A generative model for raw audio[J].arXiv preprint arXiv,2016.

[4] Yang Z,Yang D,Dyer C,et al.Hierarchical attention networks for document classification[C].Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.2016:1480-1489.

[5] Xu W, Huang L, Fox A, et al. Detecting large-scale system problems by mining console logs[C]. Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles. ACM,2009: 117-132.

【通联编辑：朱宝贵】