计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(11):40−46 [doi: 10.15888/j.cnki.csa.007666]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 Pytorch 和神经网络的云数据中心故障检测①

来风刚[1], 刘　军[2,3], 李济伟[1], 王怀宇[1], 牟霄寒[1], 刘　赛[2,3]

[1](国家电网有限公司信息通信分公司, 北京 100761)
[2](南瑞集团 (国网电力科学研究院) 有限公司, 南京 211106)
[3](南京南瑞信息通信科技有限公司, 南京 210003)
通讯作者: 刘　赛, E-mail: 948457390@qq.com

摘　要: 随着现代科技的迅猛发展, 数据中心已经成为信息化社会的 IT 基础设施, 存储管理大量关键数据. 当前, 数据中心的管理大多是依靠经验丰富的专业运维人员使用计算机自动监测机房设备各项指标, 并对设备做出多次检查, 耗时且繁琐. 深度学习和人工智能技术当前吸引了越来越多的注意力, 并在互联网和工业领域取得了许多成功应用. 本文设计了基于门控循环单元的深度学习框架对云数据中心机房设备故障进行自动化的诊断, 并联合时序信息基于过去设备的运行状态信息对未来状态进行预测. 其中, 序列数据以固定时间窗分割后输入双向 GRU 单元层, 使网络学习到数据点的前后时间依赖关系. 在 GRU 层输出基础上, 我们添加了自注意力层和 embedding 层, 让神经网络能够学习到对故障预测更有效的特征并进一步对特征进行降维. 最后, 多层感知机被用于对降维后的数据进行分类. 基于真实数据集的实验结果显示, 本文提出的基于 GRU 的深度学习框架相比 LSTM, SVM 和 KNN 等常用模型能够更准确地检测出云数据中心故障.

关键词: 循环神经网络; 云数据中心; 硬件故障诊断

引用格式: 来风刚,刘军,李济伟,王怀宇,牟霄寒,刘赛.基于 Pytorch 和神经网络的云数据中心故障检测.计算机系统应用,2020,29(11):40–46. http://www.c-s-a.org.cn/1003-3254/7666.html

Cloud Data Center Fault Detection Based on Pytorch and Neural Network

LAI Feng-Gang[1], LIU Jun[2,3], LI Ji-Wei[1], WANG Huai-Yu[1], MOU Xiao-Han[1], LIU Sai[2,3]

[1](State Grid Information and Telecommunication Branch, Beijing 100761, China)
[2](Nari Group Corporation (State Grid Electric Power Research Institute), Nanjing 211106, China)
[3](Nanjing Nari Information Communication Technology Co. Ltd., Nanjing 210003, China)

Abstract: With the rapid development of modern technology, the data center has become the IT infrastructure of the information society, storing and managing a large amount of key data. At present, the management of data centers mostly relies on experienced professional operation and maintenance personnel to use computers to automatically monitor equipment room equipment indicators and make multiple inspections of equipment, which is time-consuming and tedious. Deep learning and artificial intelligence technologies are currently attracting more and more attention and have achieved many successful applications in the Internet and industrial fields. This study designs a Gated Recurrent Unit (GRU) based deep learning framework to automatically diagnose equipment failures in cloud data center equipment rooms and combines timing information to predict future states based on past equipment operating status information. Series data are split into fixed time windows as input to the bidirectional GRU layer which makes the network learn the time dependency relationship in data points. Besides, we add an attention layer and embedding layer after the output of GRU unit, to help the neural network learning more efficient features for prediction task and further dimension reduction. At last, multi-layer

① 基金项目: 国家电网有限公司 2019 年总部科技项目 (5700-201917224A-0-0-00)
Foundation item: Year 2019, Headquarter Science and Technology Project of State Grid (5700-201917224A-0-0-00)
收稿时间: 2020-04-02; 修改时间: 2020-04-28; 采用时间: 2020-05-10; csa 在线出版时间: 2020-10-29

perception is used to classify the data. Experimental results based on real data sets show that proposed neural network framework based on GRU can accurately detect cloud data center faults compared with LSTM, SVM and KNN.

Key words: recurrent neural network; cloud data center; hardware troubleshooting

信息时代的到来使得数据量急剧增长, 云数据中心逐渐成为研究的热门之一[1,2]. 近年来, 数据中心同样在智能电网中得到广泛应用, 通过将数据在云端进行存储, 处理, 能够实现高速率、低成本储存和管理, 提高运行效率[3]. 设备管理是云数据中心正常运行的必要环节之一, 通过对设备运行参数进行采集, 对设备运行状况进行诊断. 设备运行故障能否被及时检测, 直接关系到数据中心的安全运行.

在智能电网领域, 设备故障检测方法主要针对于电网中各种元件. 当故障元件的电气量发生突变时, 利用一些智能诊断方法对电气量的异常信息进行分析, 从而找出产生故障的元件, 发现故障原因. 这些诊断方法主要包括专家系统[4], 贝叶斯网络[5], 多信息融合技术[6] 和神经网络[7] 等. 然而, 这些故障诊断方法主要针对于电网中的电力传输部件, 例如变压器, 电缆, 开关等. 云数据中心作为新兴技术之一, 其运行设备主要由网络设备构成, 而电网现有的设备故障检测方案显然无法满足云数据中心的要求.

目前云数据中心故障检测方法主要从网络层次进行故障诊断, 这些方法将数据中心设备划分为“基本功能单元组”, 利用不同算法对发生故障单元进行检测[8,9]. 考虑到数据中心数据的复杂性, 这些算法多数基于仿真实验进行. 且在单元层次对故障进行检测后, 后续仍需要故障发生位置进行进一步的确认. 因此这类算法在实际应用中的可靠性有待商榷.

为解决上述问题, 对云数据中心设备故障进行检测, 同时对未来运行状况进行预测, 保证设备正常运行. 本文研究了一种基于 Pytorch 以及神经网络的设备故障检测方法, 在对数据进行预处理后, 利用自然语言处理类方法使神经网络能够学习到对故障检测有效的特征, 并使用长短期记忆网络 (Long Short-Term Memory, LSTM) 对故障进行检测. 基于阿里集群数据的实验结果显示, 本方案能够有效检测设备运行故障, 并对运行状况进行预测. 本文的创新点总结如下:

(1) 采用 GRU (Gate Recurrent Unit) 模型作为架构基础, 相比常用的 LSTM 模型, 运算量显著减少, 训练速度得到提升.

(2) 双向 GRU 叠加的设计使得每个 GRU 单元在当前输入的基础上, 除了能够得到过去时间点的信息, 还能得到该时间点之后的数据, 提高了检测的准确度.

(3) 在双向 GRU 输出的基础上采用了自注意力机制, 使得重要信息能够通过训练获得更高的权重, 解决了信息超载问题并进一步提高检测准确率.

(4) Embedding 层和多层感知机对自注意力层的输出进行进一步处理, 在降低数据维度基础上提高了分类效果.

本文后续内容组织如下, 首先介绍循环神经网络的基本架构, 以及优缺点, 并引出在检测任务中常用的 LSTM 模型. 接下来对模型所采用的数据集进行概括, 并提出基于 GRU 改进后的模型架构. 最后展示了实验结果并对提出的框架作了简要总结.

1 循环神经网络

神经网络被大多数人视为能够拟合任意函数的黑盒子, 给定足够的训练数据, 以及数据标签, 在设定合适的损失函数后, 神经网络就能够被充分训练并在输出层得到特定的 y. 这一拟合能力近似于常见的连续非线性函数, 称为通用近似定理[10]:

令 $\phi(x)$ 代表非常数、有界、单调递增的连续函数, J_D 代表一个 D 维的单位超立方体, 定义为 $[0,1]^D$. 令 $C(J_D)$ 表示 J_D 上的连续函数集合, 则对于任意函数 $f \in C(J_D)$, 必然存在一个整数 M, 以及一组实数 $v_m, b_m \in \mathbb{R}$, 向量 $w_m \in \mathbb{R}^D$, $m = 1, \cdots, M$, 使得:

$$|F(x) - f(x)| < \varepsilon, x \in J_D \tag{1}$$

其中, $F(x)$ 作为函数 f 的近似实现, 且 $F(x)$ 定义如下:

$$F(x) = \sum_{m=1}^{M} v_m \phi(w_m^{\mathrm{T}} + b_m) \tag{2}$$

根据通用近似定理, 如果神经网络具有线性输出层, 以及至少一个使用“挤压”性质的激活函数的隐藏层, 则该神经网络就能够以任意的精度近似任何一个定义在 $\mathbb{R}^D$ 中的有界闭集函数. 这里, “挤压”性质的激活

函数可以是类似 Sigmoid 函数的有界函数. 神经网络强大的拟合能力使其能够拟合任意复杂的非线性关系, 在各种复杂的应用场景中提供了坚实的理论基础.

循环神经网络 (Recurrent Neural Network, RNN) 是深度学习的经典算法之一, 其具有记忆性, 能够捕捉输入信息中的时间顺序关联, 因此常被用于时间序列数据分析[11]. RNN 已经在自然语言处理领域, 例如语言识别, 语言建模和机器翻译等场景取得成功应用. 然而, 普通的神经网络存在一个缺陷, 即只能近似输入与输出之间的关系, 而前一个与后一个输入之间的关系无法处理. 对于大多数序列信息, 例如文本数据, 金融时间数据, 前后数据之间存在复杂的依赖关系, 普通的神经网络难以捕捉. RNN 正是针对处理序列信息而设计的, 图 1 展示了 RNN 的基本架构.

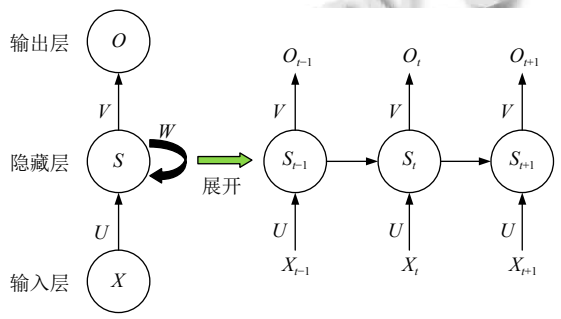

图 1 循环神经网络的基本架构

如图 1 所示, RNN 由输入层、隐藏层和输出层组成, 隐藏层展开后可以看到由多个神经元串联组成, 这里则是循环神经网络和普通神经网络的根本差别. U, V, W 分别表示输入层到隐藏层, 隐藏层到输出层和隐藏层的参数, X_{t-1}, X_t, X_{t+1} 表示不同时刻的输入. 从图 1 中可以发现, 隐藏层 S_t 在 t 时刻除了接受来自 X_t 的输入外, 还从左边接受 S_{t-1} 时刻的输入, 这就使得未来的时刻能够保留过去时刻传来的信息, 因此循环神经网络能够捕捉输入数据中的时序关系.

上述过程用公式表述如下:

$$O_t = g(VS_t) \tag{3}$$

$$S_t = f(UX_t + WS_{t-1}) \tag{4}$$

可以看到 S_t 的值由 X_t 和 S_{t-1} 共同决定, O_t 则是最终输出.

在文献 [12] 中, 作者使用 RNN 对 5432 例阿兹海默症病人的进展进行预测, 时间跨度从 2005 至 2017 年. 实验结果显示循环神经网络能够很好地捕捉单个病人在多次检查中的时序变化信息, 即便时间间隔并不均匀. 更多地, 实验结果显示循环神经网络在测试集预测病人下一次检查结果的 AD 进展上达到了 99% 的准确率, 这表明 RNN 在时序信息分析上的强大能力. 类似的, 在云数据中心中, 设备的信息、日志数据等同样以时间序列顺序被记录. 这些数据在时间顺序上的变化能够精确地被循环神经网络所捕捉, 并且学习到数据随时间变化的复杂非线性关系, 从而实现对设备未来状态的预测.

在实际应用中, 研究者发现循环神经网络难以处理信息的长时间依赖关系, 一个简单的例子是在英文句子生成时, 如果句子很长, 在生成谓语动词时, 循环神经网络无法记住主语的单复数形式并选择合适的谓语动词. 为了解决这一问题, 门控机制被提出, 即大多数错误检测模型常用的 LSTM. 在 LSTM 中, 门控机制包括遗忘门 (forget gate)、输入门 (input gate) 和输出门 (output gate). 其中, 遗忘门用于控制前一时刻输入信息通过的比例, 仅仅保留部分重要信息向后传递而忽略无关信息. 包括细胞状态和隐藏状态两种, LSTM 原理如图 2 所示.

图 2 中, X_t 表示输入 h_t 表示输出. σ和tanh激活函数的组合控制输入信息能否通过门控向后传递, 即选择性遗忘. 基于 LSTM 模型强大的处理长时程信息依赖的能力, 该模型已经成为错误检测研究中的常用模型[13,14].

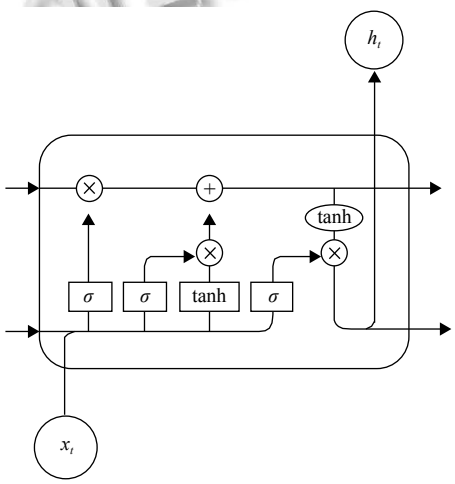

图 2 长短期记忆网络的基本架构

2 Pytorch 机器学习框架

Pytorch 是一个著名的基于 Torch 的 Python 机器

学习库, 由 Facebook 的人工智能研究组在 Torch 的基础上开发. Pytorch 在学术领域广泛应用, 作为常用深度学习研究平台之一[15], 其 API 接口统一, 采用了动态计算图机制和自动求导机制, 能够方便地搭建定制化的神经网络并进行训练. 相对于谷歌开发的 TensorFlow 框架, Pytorch 拥有以下几点优势:

(1) Pytorch 中的模型定义更为简单, 并且提供了容易调用的包, 而 TensorFlow 接口定义种类繁杂.

(2) Pytorch 所采用的动态图计算机制相比 TensorFlow 的静态图更加灵活, 对于研究者更加友好,

(3) 对 Caffe 具有良好的支持, 可以联合英伟达 GPU 进行高效的神经网络训练.

3 实验及结果分析

3.1 实验数据

云数据中心已经成为互联网的基础设施, 在网络流量日益增长的今天起到核心作用, 与此同时, 云数据中心的网络设备故障的发生率也在上升, 这使得服务器的性能下降, 对用户终端的使用体验造成影响. 本文采用循环神经网络训练公开的数据中心数据集, 并对未来故障进行预测, 这样运维人员可以在潜在故障发生前进行干预并解决.

实验使用数据集为阿里巴巴集团在 Github 平台公布的阿里云集群数据[16], 该数据集包括以下两部分数据:

(1) cluster-trace-v2017: 1300 台机器 12 小时的运行数据. 该批数据包含在线服务的数据集合以及批次化的工作负荷记录.

(2) cluster-trace-v2018: 4000 台机器长达 8 天的运行数据. 该批数据除了包含 v2017 的数据种类外, 额外包含 DAG 产品的运行负荷信息[17].

本文采用了 cluster-trace-v2018 数据集进行训练和测试, 预测所包含的故障分为设备响应时间过长, CPU 利用率过高, 内存利用率过高和传输信息速率低 5 种. 实验所采取的流程如图 3 所示.

如图 3 所示, 首先对数据进行预处理, 包括去冗余与清洗, 标准化两步. 冗余数据指与故障预测没有关联的数据或者数据不随时间变化的数据, 例如设备型号信息, 不同客户账号信息等. 冗余信息对预测效果没有影响并且会造成额外计算负担, 因此预先对冗余进行清洗是重要一步. 数据集标准化是提高预测效果的重要步骤, 当不同机器间指标水平相差较大时, 直接用原始指标进行分析会突出数值较高指标在预测中的作用, 导致高数值指标在预测中所占权重过大, 影响结果的可靠性. 这里我们对机器的不同指标采用最为常用的 Z-score 标准化, 以时间序列 $x_1, x_2, \cdots, x_n$ 为例, Z-score 计算方式为:

$$y_i = \frac{x_i - \overline{x}}{s} \tag{5}$$

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{6}$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{7}$$

则新序列 $y_1, y_2, \cdots, y_n$ 的均值为 0, 标准差为 1. Z-score 方法对数值区间较大, 离群值较多的情况比较合适, 在对不同指标进行 Z-score 处理后, 能够使数据落入一个较小的区间, 减轻不同指标间数值差异的影响.

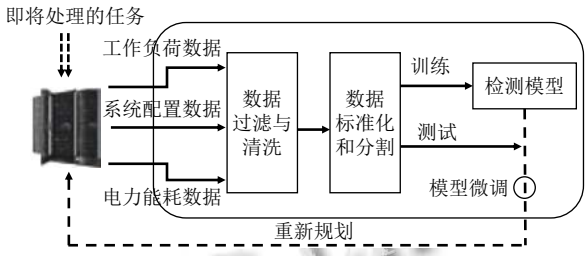

图 3 实验流程图

本文以每 1 分钟的数据为一个窗口, 将 1 分钟的阿里云硬件集群的各项指标进行平均, 作为单个数据进行统计. 本文使用统计检验将 CPU 使用率和内存分配率等指标与平均指标严重偏离的时间点作为故障时间点 (P-value<0.05). CPU 使用率在一天内使用率波动较大, 在部分时间点存在高峰, 经过统计后发现在高峰使用期故障发生率较高, 这与真实情况相符合. 针对内存使用故障, 使用请求内存和真实使用内存之差作为判断故障指标 (P-value<0.05). 统计结果表明 CPU 使用率高峰和内存使用故障高峰存在一定的关联, 这正常情况相一致. 在实验中仅考虑单个时间点只存在一种故障, 将故障检测视为单标签分类问题. 图 4 显示了不同故障数目的柱形图.

3.2 网络结构与实验结果

本文使用的检测模型设计架构如图 5 所示.

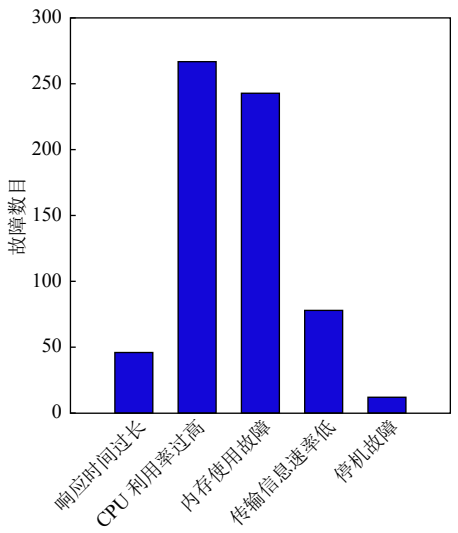

图 4 故障数目统计

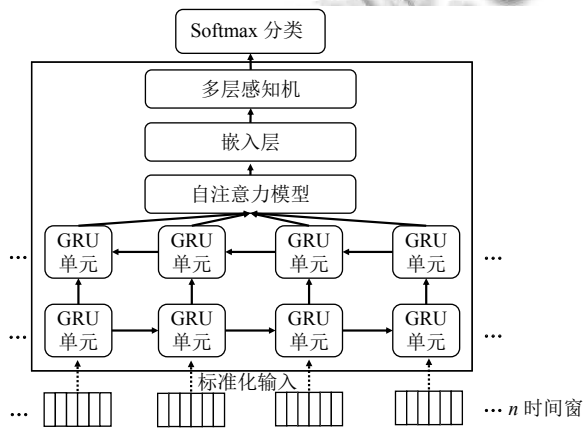

图 5 神经网络架构

图 5 中, 输入层包含 200 个神经元, 因此数据被分割为 200 个时间窗输入. 本文使用双向门控循环神经网络处理数据输入. 双向门控循环神经网络 (BiGRU) 可以看做的 LSTM 一种拓展, 将 LSTM 中的 LSTM 模块替换为 GRU 并使用双层 GRU 模块反向叠加. GRU 将 LSTM 中隐藏状态和细胞状态合并成一种状态, 因此显著缩短了训练时间. 更明确地说, GRU 读取词嵌入向量t_i以及隐藏层状态向量h_{i-1}后, 经过门控计算产生输出向量C_i和隐藏层状态向量h_i, 具体计算方法参考下列公式:

$$\begin{cases} z_i = \sigma(W_z t_i + V_z h_{i-1} + b_z) \\ r_i = \sigma(W_z t_i + V_z h_{i-1} + b_r) \\ c_i = \tanh(W t_i + v(r_i \odot h_{i-1}) + b) \\ h_i = z_i \odot h_{i-1} + (1 - z_i) \odot c_i \end{cases} \tag{8}$$

其中, $z \in \Re^d$, $r \in \Re^d$分别表示接受 d 维向量的输入门与重置门, $\{W_z, W_r, W, V_z, v_r, V\}$表示权重矩阵, $\{b_z, b_r, b\}$为偏置向量, $\odot$表示矩阵点乘. 在双向 GRU 后是 self-attention 层、embedding 层和多层感知机 (Multi-Layer Perception, MLP). 输出层为 5 个神经元组成的 Softmax 层, 计算输入时间段在 5 类故障上的概率.

为了让神经网络能够学习到对预测故障有效的特征, 本文采用了自然语言处理中的 embedding 技术. 类似于常用的预训练词向量, 在输入层后加入 embedding 层, 其中包含 100 个神经元, 对输入特征起到降维作用[17]. 此外, 在 embedding 层前加入了注意力层, 并使用了经典的自注意力模型. 自注意力机制引入了查询向量 q (query vector), 通过打分函数计查询向量和输入向量直接的相关性, 同时引入了一个注意力变量 $t \in [1, N]$代表选择的索引位置. 具体计算方式如下:

$$\begin{aligned} a_i &= p(t = i|X, q) \\ &= Softmax(s(x_x, q)) \\ &= \frac{\exp(s(x_i, q))}{\sum_{j=1}^{N} \exp(s(x_i, q))} \end{aligned} \tag{9}$$

其中, a_i 是注意力分布, $s(x_i,q)$ 是注意力打分函数. 自注意力打分函数采用基于缩放点积函数, 缩放点积的定义如下:

$$s(x_i, q) = \frac{x_i^{\mathrm{T}} q}{\sqrt{d}} \tag{10}$$

其中, d 表示输入向量的维度. 缩放点积模型是基于点积模型的一种改进, 区别在于缩放点积模型除以向量维度 d 的平方根. 当 d 很大时, 点积模型的值会出现较大的方差, 因此导致 Softmax 的梯度变小, 缩放点积模型的提出解决了这一问题. 可以看出, 自注意力层通过查询向量实现对输入数据的权重分配, 这一查询向量可以通过反向传播进行学习和优化, 从而能够对重要的特征分配更大的权重. 在阿里云数据中心数据集中, 不同类型的数据对故障分类的重要性存在差别. 例如, 线程分配数目和消息队列排队数目对 CPU 响应时间过长和 CPU 利用率过高这两类故障类型起到直接的影响, 缓存区数据量和堆栈区使用率则对内存使用故障和传输速率低起到决定性作用, 而在输入数据中无法体现此类差别. 因此, 在双向 GRU 层充分学习输入数据的时序信息后, 引入自注意力层能够对映射后特征进行权重分配, 使得不同位置的特征对每类故障学

习不同的权重, 从而提升最终的检测效果. 在模型实现中, 我们采用了以自注意力模型为基础的 Multi-head Attention. Multi-head Attention 在自注意力基础上进行了进一步拓展, 能够同时学习到 GRU 输出序列的位置编码信息和特征的权重信息. 该注意力机制在著名的自然语言处理模型 Transformer 和 Bert 中被广泛采用.

实验平台的具体配置如下:

操作系统: Windows 10.

GPU: NVIDIA RTX2080Ti, 11 GB 显存.

RAM: 64 GB.

深度学习框架: Pytorch 1.3 稳定版.

开发工具: Visual Studio Code.

编程语言: Python 3.6.

Adam 是一种基于随机梯度下降 (Stochastic Gradient Descent, SGD) 的一阶优化算法, 与 SGD 不同在于 SGD 在训练过程中学习率不会改变, 而 Adam 通过计算梯度的一阶和二阶矩估计动态改变学习率, 是一种自适应学习率优化算法, 同时结合了 AdaGrad 和 RMSProp 两种算法的优点[18].

本文将数据集分割为训练集与测试集, 其中验证集数目占 20%, 训练集数目占 80%. 使用 Adam 优化算法训练神经网络, 实验结果显示 Adam 算法效果卓越, 如图 6 所示, 在使用 Adam 算法后, 训练集和测试集上的 loss 均能够降低到 0.05 左右.

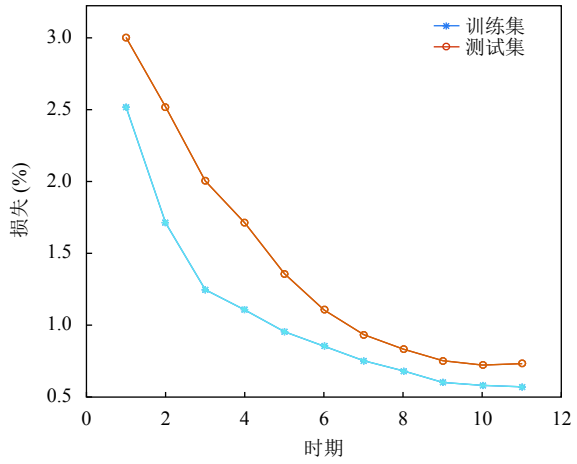

图 6 模型损失函数变化

图 7 展示了神经网络对于故障检测的准确率, 在实验中, 设置 batch 大小为 200, 通过 11 个 epoch 后算法已经接近收敛并在测试集上获得了超过 98% 的准确率.

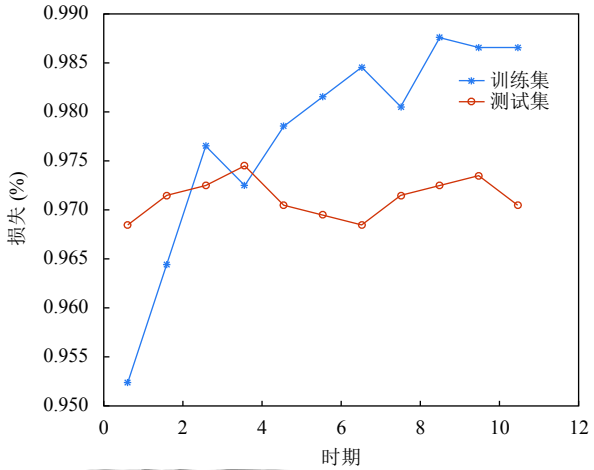

图 7 模型准确率

作为对比, 本文使用 SVM, KNN 和普通 LSTM 模型对故障进行检测, 检测结果如表 1 所示.

表 1 不同模型准确率对比

模型	训练集准确率	测试集准确率
BiGRU+embedding+attention	0.987	0.971
LSTM	0.964	0.933
SVM	0.977	0.903
KNN	0.918	0.882

综合来说, 深度学习模型相比传统机器学习模型拥有更高的诊断准确率. SVM 作为经典的分类模型, 其在训练集上表现出良好的效果, 但是测试集上准确率显著下降, 存在明显的过拟合现象. KNN 分类器因为没有显式的训练过程, 在训练集和测试集上的诊断效果均较为一般, LSTM 在训练集和测试集上表现较为稳定, 表明深度学习模型能够学习到数据中的时序变化信息. 相比普通 LSTM, 加入了 embedding 层和自注意力机制后 BiGRU 模型准确度相比基线模型 LSTM 有 2% 的准确度的提升. 实验结果显示深度学习模型能够在云数据中心进行部署, 相比传统机器学习模型拥有更高的故障诊断准确率. 除此之外, 随着数据的积累, 深度学习模型的准确率能够进一步提高.

4 结论与展望

为了解决当前云数据中心缺乏故障检测方法, 且当前方法均基于仿真数据实验的问题, 本文提出了一种基于 Pytorch 和双向 GRU 网络的云数据中心故障检测方法. GRU 模型的使用相比传统的 LSTM 提高了训练速度, 并且双向机制的结合进一步提高了模型的检

测准确度. 在对数据进行预处理后, 利用 embedding 技术使神经网络能够提取关于故障检测的相关特征, 并使用特征进行进一步加工和处理, 最后利用 Adam 优化算法训练神经网络. 基于阿里云集群数据的实验结果显示, 相比于其他模型, 本文提出的模型准确率有着明显改善, 有助于智能电网云数据中心故障检测的准确率, 可靠性的全面提升.

参考文献

1 Uchechukwu A, Li KQ, Shen YM. Retraction note: Sustainable cost and energy consumption analysis for cloud data centers. Frontiers of Computer Science, 2020, 14(1): 239. [doi: 10.1007/s11704-015-4020-6]

2 Mokadem R, Hameurlain A. A data replication strategy with tenant performance and provider economic profit guarantees in cloud data centers. Journal of Systems and Software, 2020, 159: 110447. [doi: 10.1016/j.jss.2019.110447]

3 宋杰, 孙宗哲, 刘慧, 等. 混合供电数据中心能耗优化研究进展. 计算机学报, 2018, 41(12): 2670–2688. [doi: 10.11897/SP.J.1016.2018.02670]

4 赵伟, 白晓民, 丁剑, 等. 基于协同式专家系统及多智能体技术的电网故障诊断方法. 中国电机工程学报, 2006, 26(20): 1–8. [doi: 10.3321/j.issn:0258-8013.2006.20.001]

5 罗孝辉, 童晓阳. 计及可信度的变结构贝叶斯网络电网故障诊断. 电网技术, 2015, 39(9): 2658–2664.

6 徐彪, 尹项根, 张哲, 等. 基于拓扑图元信息融合的电网故障诊断模型. 电工技术学报, 2018, 33(3): 512–522.

7 Novelo AF, Cucarella EQ, Moreno EG, *et al*. Fault diagnosis of electric transmission lines using modular neural networks. IEEE Latin America Transactions, 2016, 14(8): 3663–3668. [doi: 10.1109/TLA.2016.7786348]

8 宣恒农, 张润驰, 左苗, 等. 面向数据中心网络的分层式故障诊断算法. 电子学报, 2014, 42(12): 2536–2542. [doi: 10.3969/j.issn.0372-2112.2014.12.029]

9 Qi XG, Wang BC, Liu LF. Fault diagnosis based on dial-test data in datacenter networks. Journal of Systems Engineering and Electronics, 2019, 30(5): 1035–1043. [doi: 10.21629/JSEE.2019.05.19]

10 Cybenko G. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 1989, 2(4): 303–314. [doi: 10.1007/BF02551274]

11 Greff K, Srivastava RK, Koutník J, *et al*. LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(10): 2222–2232. [doi: 10.1109/TNNLS.2016.2582924]

12 Wang TY, Qiu RG, Yu M. Predictive modeling of the progression of Alzheimer's disease with recurrent neural networks. Scientific Reports, 2018, 8(1): 9161. [doi: 10.1038/s41598-018-27337-w]

13 Filonov P, Lavrentyev A, Vorontsov A. Multivariate industrial time series with cyber-attack simulation: Fault detection using an LSTM-based predictive data model. arXiv preprint arXiv: 1612.06676, 2016.

14 Park D, Kim S, An YL, *et al*. LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks. Sensors, 2018, 18(7): 2110. [doi: 10.3390/s18072110]

15 Ketkar N. Introduction to Pytorch. In: Ketkar N, ed. Deep Learning with Python. Berkeley: Apress, 2017. 195–208.

16 Lu CZ, Ye KJ, Xu GY, *et al*. Imbalance in the cloud: An analysis on Alibaba cluster trace. Proceedings of 2017 IEEE International Conference on Big Data (Big Data). Boston, MA, USA. 2017. 2884–2892.

17 Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar. 2014. 1532–1543.

18 Kingma DP, Ba LJ. Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations. San Diego, CA, USA. 2015.