

# Self-Att-BiLSTM: 一种面向业务流程活动与时间的多任务预测方法

贺琪<sup>1</sup>, 杨巧青<sup>1</sup>, 黄冬梅<sup>2</sup>, 宋巍<sup>1\*</sup>, 杜艳玲<sup>1</sup>

<sup>1</sup>上海海洋大学信息学院, 上海 201306;

<sup>2</sup>上海电力大学, 上海 200090

**摘要** 业务流程中事件日志的分析与预测可以为流程监控和管理提供决策信息, 现有研究方法多针对特定单个任务预测, 不同任务间预测方法的可迁移性不高。多任务预测可以共享多个任务间的信息, 提升单个任务预测的精度, 但现有研究对重复活动的多任务预测效果有待提高。针对以上问题, 提出一种注意力机制与双向长短时记忆结合的深度神经网络模型, 实现对业务流程中重复活动和时间的多任务预测。预测模型可以共享不同任务已经学到的特征表示, 实现多任务并行训练。在多个数据集中对不同方法进行对比, 结果表明, 所提方法提高了预测效率和预测精度, 尤其对重复活动的预测精度有较好提升。

**关键词** 图像处理; 业务流程监控预测; 多任务学习; 注意力机制; 双向长短时记忆网络

中图分类号 TP391

文献标志码 A

doi: 10.3788/LOP202158.0410003

## Self-Att-BiLSTM: A Multitask Prediction Method for Business Process Activities and Time

He Qi<sup>1</sup>, Yang Qiaoqing<sup>1</sup>, Huang Dongmei<sup>2</sup>, Song Wei<sup>1\*</sup>, Du Yanling<sup>1</sup>

<sup>1</sup>College of Information Technology, Shanghai Ocean University, Shanghai 201306, China;

<sup>2</sup>Shanghai University of Electric Power, Shanghai 200090, China

**Abstract** Decision information for process monitoring and management can be obtained by analyzing and predicting the event log of the business process. The existing research methods are mostly targeted at specific single-task prediction, and the portability between different task prediction methods is not high. Through multitask prediction, information can be shared among multiple tasks, improving the single-task prediction accuracy. However, the multitask prediction effect of existing research on repetitive activities must be improved. Based on the aforementioned problems, we propose a depth neural network model combining the attention mechanism and bidirectional long-short term memory, achieving multitask prediction for repetitive activities and time associated with the business process. The proposed prediction model can share the learned feature representation of different tasks and achieve multitask parallel training. Comparison is performed by applying different methods on datasets. The obtained results demonstrate that the proposed method improves the prediction efficiency and accuracy, especially in case of repetitive activities.

**Key words** image processing; business process monitoring and prediction; multitask learning; attention mechanism; bidirectional long-short term memory network

**OCIS codes** 100.4996; 150.5495; 100.5010

收稿日期: 2020-06-19; 修回日期: 2020-07-16; 录用日期: 2020-08-03

基金项目: 国家自然科学基金(41671431, 61702323)、国家重点研发计划(2016YFC1400304)

\* E-mail: wsong@shou.edu.cn

## 1 引言

预测性业务流程监控是流程挖掘<sup>[1-2]</sup>的一个研究热点。预测性业务流程监控可以识别出流程运行中不符合某些标准的异常事件,及时地做出相应措施。针对活动和时间的多任务预测,业务流程不仅可以准确、高效地提高企业服务效率、降低运行成本、预防违规活动发生及提前发现异常案例等,而且可以人为地规划和分配资源,达到更高的效率和实现更好的服务目标。目前预测的范畴主要包括预测流程未来活动<sup>[3]</sup>、预测流程结果<sup>[4]</sup>、预测时间相关属性<sup>[5-6]</sup>、预测业务流程执行结束后的性能<sup>[7]</sup>等。关于业务流程预测,已有的成果所使用的方法包括使用转换系统(TS)来预测剩余时间<sup>[8]</sup>、使用概率有限自动机(PFA)预测未来活动<sup>[9]</sup>、使用马尔可夫链方法预测未来活动<sup>[10]</sup>、基于传统的机器学习方法(包括决策树<sup>[11]</sup>、随机森林预测流程结果<sup>[12]</sup>)等。

21 世纪以来,由于关于深度学习方面的研究不断深入,传统手工提取特征的过程逐渐被关于深度神经网络的方法所取代。对于事件日志这样的大量样本数据,通过学习事件日志中的内在特征和表示层次,深度神经网络可以容易地预测业务流程中的相关任务。Evermann 等<sup>[13]</sup>使用递归神经网络(RNN)预测下一个活动,该网络由两个隐藏的 RNN 层组成,其中使用基本的长短期记忆(LSTM)网络<sup>[14-15]</sup>单元。Tax 等<sup>[16]</sup>使用 LSTM 预测下一个活动及其持续时间。Navarin 等<sup>[17]</sup>提出一种用于业务流程实例的数据感知剩余时间预测的 LSTM 网络。Tello-Leal 等<sup>[18]</sup>使用 LSTM 神经网络预测业务流程中的活动。然而一方面,现有的深度学习的方法多数使用单一的信息进行预测,这些预测方法由于都是对某个任务定制的,可迁移性不高,并且没有考虑事件日志中属性之间的相互关联关系,因此不同的预测任务需要构建不同的模型。另一方面,虽然 Tax 等<sup>[16]</sup>使用 LSTM 网络来对业务流程活动和时间进行多任务预测,但在活动和时间方面的精度均不高,并且它仅仅解决了非重复活动的预测。

综上,为了提高活动和时间多任务预测的精度以及解决对于重复活动的预测,本文提出一种 Self-Att-BiLSTM 的深度神经网络模型,用于进行业务流程活动和时间多任务预测。首先将事件轨迹转换为特征编码,并将其作为输入,通过嵌入层将高维稀疏特征转换为低维稠密并且有相互关系的向量;利用双向 LSTM(BiLSTM)对事件日志进行时序信

息建模,使用注意力机制(Self-Attention)捕获事件日志中远距离的相互依赖的特征并且动态地调整事件日志中重复活动的隐藏状态的重要性,在网络共享层之后,时间通过全连接层预测,活动通过全连接层和 softmax 分类器预测。

## 2 业务流程相关定义

在业务流程中,每个事件的相关属性都会被顺序记录下来形成事件日志。事件日志是一个具有层次结构的文件,其中包含执行业务流程的数据,这些数据是由事件轨迹组成的。事件是一个事件轨迹的原子部分,它包含着其他属性。事件典型的属性是执行活动的名称、活动的时间戳、活动生命周期(例如“开始”或“完成”)、组织资源及角色。事件按照其发生的时间戳排序,其他属性可能包含特定情况的信息。每个事件轨迹表示一个业务流程实例(也称案例)的执行。业务流程中,对于活动和时间的预测通常理解为在事件日志的事件轨迹运行过程中预测下一个执行活动和该活动发生的时间。如下为流程相关定义。

**定义 1 事件(Event):**也可称为活动,是业务流程中最小的单元,它的基本形式定义为  $e = (c, a, t_{\text{timestamp}}, d_1, \dots, d_m)$ 。其中,  $e$  中的属性  $c$  是案例的序号,每个案例的序号都是唯一的。而  $a$  则是该事件任务的名称,值是唯一的,包括任务名称或者代称。 $t_{\text{timestamp}}$  表示事件的时间戳。 $d_1, \dots, d_m$  则是该事件对应的其他属性。

**定义 2 事件轨迹(Event Trace):**业务流程中一个事件轨迹是一个非空的有序事件序列,称为流程序列。 $\sigma = \langle e_1, e_2, e_3, \dots, e_n \rangle$  为长度为  $n$  的事件轨迹, $e_i \in \sigma$  则是事件轨迹  $\sigma$  中的第  $i$  个发生的事件。事件轨迹中每个事件的  $c$  属性都是一致的,每一个事件轨迹也称为一个案例或者流程序列。

**定义 3 事件日志(Event Log):**记录的事件日志是一系列事件轨迹的集合。 $L = \{\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_k\}$  是包含了  $k$  个事件轨迹的事件日志,其中  $\sigma_j \in L$  是事件日志  $L$  中第  $j$  个事件轨迹。所有被记录的事件轨迹均为同一业务流程运行过程中的事件日志。

## 3 Self-Att-BiLSTM 多任务预测模型

为了进行重复活动和时间的多任务预测,采用多任务学习框架<sup>[19-21]</sup>来进行业务流程活动和时间预测,通过网络共享层表示方法,把活动和时间这两个任务放在一起学习。使用来自活动和时间这些任务

的样本来优化所有网络参数,同时提升多任务的性能。通过共享相关任务之间的表示,模型能更好地泛化学习任务,使得每一个任务的预测效果都有提

升。图 1 为活动和时间的多任务预测网络模型框架,主要包含输入层、Embedding 层、BiLSTM 层、Self-Attention 层、多任务输出层。

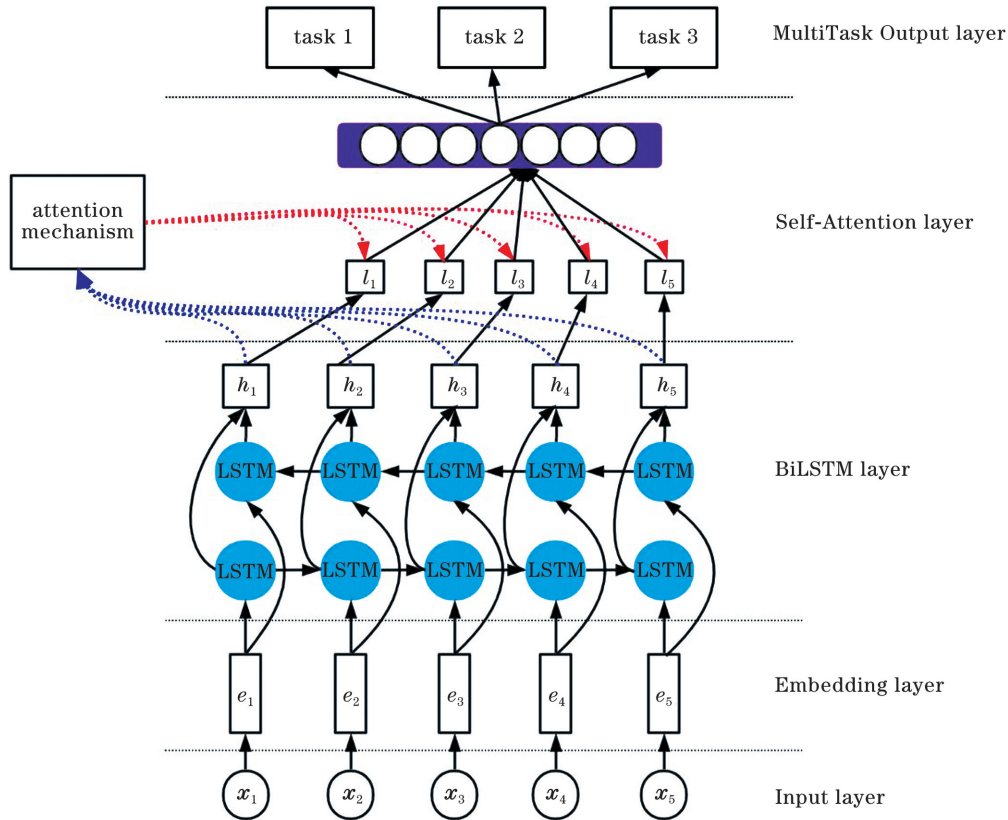


图 1 Self-Att-BiLSTM 多任务预测网络模型

Fig. 1 Self-Att-BiLSTM multitask prediction network model

### 3.1 输入层与 Embedding 层

事件日志需要将包含的属性处理为特征向量才能用于预测模型中,事件日志为  $L = \{\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_k\}$ ,  $\sigma_j$  表示第  $j$  个事件轨迹,其中  $\sigma_j = \langle e_{t1}, e_{t2}, e_{t3}, \dots, e_{tn} \rangle$ ,  $n = |\sigma_j|$ ,将  $L$  中每个事件轨迹构建为特征向量  $x = [x_1, x_2, x_3, \dots, x_p]$ ,  $p$  个特征向量代表  $p$  个样本,其中每一个样本的特征向量  $x_i$  是一个二维向量,第一个维度表示事件轨迹的时刻信息,第二个维度表示事件属性的个数。事件属性包括两部分,一部分为数值型属性,一部分为非数值型属性。数值型的特征可以根据事件日志  $L$  中该属性的所有可能值的范围对属性值进行规范化,非数值型的特征采用 one-hot 编码形式,1 代表该位置上活动编为 true,其他位置均为 0。对于时间预测,采用活动时间间隔,为特征向量  $x$  添加三个基于时间的特征,即活动时间间隔、活动距离午夜的时间间隔、活动开始执行的星期信息。向模型中添加后面两个辅助特征后,模型在训练过程中可以获得更多业务流

程中与活动、时间相关的特殊信息。

将特征向量输入到 Embedding 层,如图 2 所示,通过线性映射将高维稀疏的特征变为低维稠密向量  $e = [e_1, e_2, e_3, \dots, e_t]$ ,  $e_t \in \mathbf{R}^d$ ,其中  $d$  表示事件中所有属性编码的维数,  $e_t$  表示  $t$  时刻的输入。使用 Embedding 层,是因为 one-hot 编码的向量是高维且稀疏的,并且 one-hot 编码不依赖任何内在的联系。在训练神经网络的过程中使用 Embedding 层时,每个 embeddings 向量都会得到更新,训练过程中会探索多维空间中不同向量之间的相似性。

### 3.2 BiLSTM 层

BiLSTM 由一个前向传播和一个后向传播的 LSTM 组成,LSTM 对长期依赖关系具有强大的建模功能。LSTM 的细胞单元如图 3 所示,其中,  $C_t$  为当前状态值,  $H_t$  为输出值。在事件日志中,为了得到更好的预测模型,需要同时考虑上下文。使用 BiLSTM 隐藏状态捕捉过去和未来的信息状

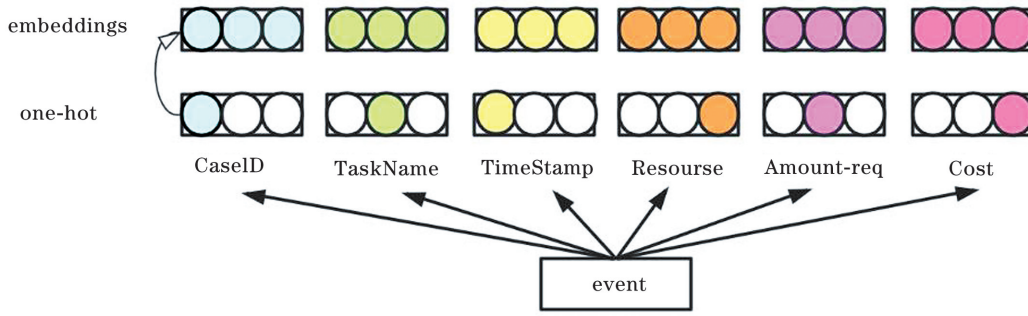


图 2 将 one-hot 编码转换为 embeddings 向量

Fig. 2 Transformation from one-hot code to embeddings vector

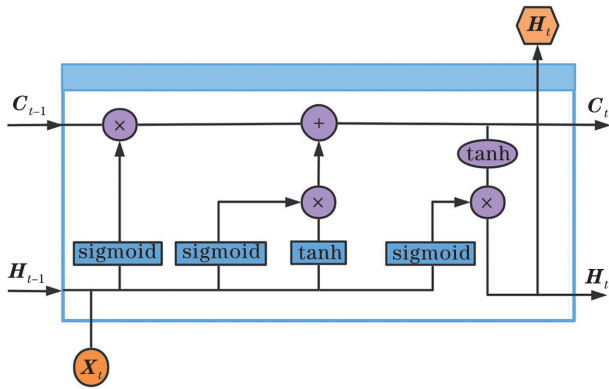


图 3 LSTM 单元结构

Fig. 3 LSTM cell structure

态,并将这些信息连接起来构成最后的状态输出。

LSTM 通过当前输入的  $e_t$  和之前的隐藏状态  $h_{t-1}$  来更新一个隐藏状态  $h_t$ 。 $i_t$  为输入门,有选择性地保留输入的信息,并对细胞状态进行更新; $g_t$  为遗忘门,有选择性地遗忘冗余信息; $o_t$  为输出门,决定输出细胞状态的某些部分。具体表达式分别为

$$i_t = \text{sigmoid}(w_i * [h_{t-1}, e_t] + b_i), \quad (1)$$

$$g_t = \text{sigmoid}(w_g * [h_{t-1}, e_t] + b_g), \quad (2)$$

$$c'_t = \tanh(w_c * [h_{t-1}, e_t] + b_c), \quad (3)$$

$$c_t = g_t * c_{t-1} + i_t * c'_t, \quad (4)$$

$$o_t = \text{sigmoid}(w_o * [h_{t-1}, e_t] + b_o), \quad (5)$$

$$h_t = o_t * \tanh c_t, \quad (6)$$

式中: $w$  为上一个向量的输出与当前特征向量的输入经过控制门的权重; $b$  为通过控制门的偏置;sigmoid、tanh 为激活函数。通过引入隐藏的连接, BiLSTM 网络以相反的时间顺序流动扩展了单向 LSTM 网络。通过同时前向传播来获得输出状态  $\vec{h}_t$  和后向传播获得输出状态  $\overleftarrow{h}_t$  来更新当前的状态  $h_t$ 。与传统的单项 LSTM 相比, BiLSTM 可以同时学习过去和将来的信息,从而获得更加稳健的特征信息。

$$\vec{h} = \overrightarrow{L}_{\text{LSTM}}(h_{t-1}, e_t), \quad (7)$$

$$\overleftarrow{h} = \overleftarrow{L}_{\text{LSTM}}(h_{t+1}, e_t), \quad (8)$$

$$h_t = [h_t, \overleftarrow{h}_t]. \quad (9)$$

### 3.3 Self-Attention 层

Self-Attention 层可以增强预测重复活动的特征,通过使用注意力机制<sup>[22]</sup>来构造预测模型。注意力机制考虑了每个输入元素不同的权重系数,更加注重与其输入元素相关的内容,忽略了一些其他无关的信息。其最大的优点就是能一步到位地捕捉全局依赖和局部依赖,引入 Self-Attention 后,会更容易捕获事件日志中长距离的相互依赖的特征并且动态地调整事件日志中重复活动隐藏状态的重要性,它很少依赖外部的信息,更加善于捕获事件中内部特征的关联性。循环神经网络 LSTM 按照时间序列进行计算,当有依赖关系的特征距离较远时,要进行多个时间步才能将两者的信息联系起来。在计算过程当中, Self-Attention 会直接对事件轨迹中任意两个活动的依赖进行关联,这样既可以对出现的重复活动进行连接,并且距离较远的特征关系被很大程度缩短,还高效地使用了相关特征。该方法更适用于处理顺序数据的注意力机制,可以充分考虑每个时间戳的上下文信息。将 BiLSTM 层输出的向量矩阵  $[h_1, h_2, h_3, \dots, h_t]$  输入到 Self-Attention 层中。引入自注意矩阵  $A$  来捕获任何特征相对于所有邻居的相似性。Self-Attention 的实现类似 LSTM 的细胞结构,具体表达式为

$$g_{t,t'} = \tanh(w_g h_t + w'_g h_{t'} + b_g), \quad (10)$$

$$a_{t,t'} = \text{sigmoid}(w_a g_{t,t'} + b_a), \quad (11)$$

$$l_t = \sum_{t'=1}^T a_{t,t'} \cdot h_{t'}, \quad (12)$$

式中: $a_{t,t'} \in A$  表示在时间戳为  $t$  和  $t'$  时通过隐藏状态  $h_t$  和  $h_{t'}$  获得的相似性矩阵; $w_g$  和  $w'_g$  分别对应隐

藏状态  $\mathbf{h}_t$  和  $\mathbf{h}_{t'}$  的权重矩阵,  $\mathbf{w}_a$  表示它们非线性组合对应的权重矩阵;  $\mathbf{b}_a$  和  $\mathbf{b}_g$  为偏置向量;  $l_t$  表示在  $t$  时刻的注意力隐藏状态是由在时间为  $t'$  内所有其他隐藏状态  $\mathbf{h}_{t'}$  和相似性矩阵  $\mathbf{a}_{t,t'}$  加权求和得到的。

### 3.4 多任务输出层

多任务学习共享表示既可以防止过拟合, 又能够学习更好的特征表达来提高泛化能力。在多任务输出层主要有两个任务, 一是预测活动, 二是预测时间, 将预测活动任务转换为一个多分类任务, 将预测时间转换为一个回归任务。对于预测活动, 使用 softmax 分类器输出活动预测值, 使用的交叉熵损失函数表达式为

$$L_{\text{loss}}(\mathbf{x}) = - \sum_{i_L=1}^{p_L} y_{i_L} \log f_{i_L}(\mathbf{x}), \quad (13)$$

式中:  $\mathbf{x}$  为输入的事件轨迹;  $p_L$  为待分类的活动总数;  $y_{i_L}$  为第  $i_L$  个类别对应的真实标签;  $f_{i_L}(\mathbf{x})$  为对应的模型输出值。

对于时间预测, 使用的损失函数为平均绝对误差(MAE), 是预测时间结果与预期产出之间的平均绝对差。在神经网络进行前向传播阶段, 依次调用每个层的前向函数, 得到逐层的输出, 对最后一层与目标函数进行比较后得到损失函数, 通过计算误差来更新权值。反向传播到达网络第一层, 在反向传播结束时, 前向传播与后向传播的网络的权值将一起更新, 得到的损失值越小, 模型的预测结果就越接近真实情况, 神经网络模型的准确性也就越好。采用 ReLU 激活函数来进行时间预测, 表达式为

$$R_{\text{ReLU}}(x) = \max(0, x). \quad (14)$$

ReLU 激活函数可以使网络训练更快, 因为时间预测是非线性的, ReLU 激活函数增加网络的非线性, 还可以防止梯度消失, 并且减少过拟合风险。

## 4 实验及结果分析

### 4.1 实验配置与数据

实验均使用 Python3.6 软件实现, 采用 Keras 框架搭建神经网络预测模型。Keras 是一个深度学习框架, 有许多模块包, 方便模型构建来验证模型的准确性和精度。实验运行环境为 Windows10, Intel Core i5 CPU, 2.6GHz, 8G RAM。表 1 为实验中的参数设置。

表 1 实验参数

Table 1 Experimental parameters

Parameter	Content
Learning rate	0.001
Loss function 1	Categorical_crossentropy
Loss function 2	MAE
Optimizer	Nadam
Dropout	0.2
Activation function	softmax, ReLU
Number of LSTM units	50
Batch_size	128
Epoch	50

实验采用 2 个数据集, 一个为 Helpdesk 数据集, 另一个为 BPI12 数据集, 这两个数据集均为真实流程数据。

Helpdesk 数据集: 此事件日志包含来自意大利软件公司 Helpdesk 帮助台的票务管理流程。这个流程由 9 个活动组成, 所有情况都是在票务管理系统中插入一张新票开始的。每个案例在问题解决和票据关闭时结束。这个事件日志包含了大约 3804 个案例和 13710 个事件。

BPI12 数据集: 此事件日志源于业务流程智能挑战(BPI12), 为某大型金融机构业务流程的真实流程数据。事件日志描述的是全球金融组织中申请个人贷款或透支的流程。这个流程包含 3 个子流程: 一是记录应用程序的状态, 二是记录与应用程序相关的工作项状态, 三是记录反馈用户的状态。该日志总共包括 262200 事件和 13087 个案例。

### 4.2 实验评价方法

同时进行活动和时间的多任务预测, 对于活动的预测任务是一个分类问题, 通常使用分类算法度量指标精度。精度指标既适用于二分类任务, 同时适用于多分类问题。简单来说, 精度是分类正确的样本数占样本总数的比例。精度的表达式为

$$A_{\text{cc}} = \frac{N_{\text{tp}} + N_{\text{tn}}}{N_{\text{tp}} + N_{\text{tn}} + N_{\text{fp}} + N_{\text{fn}}}, \quad (15)$$

式中:  $N_{\text{tp}}$ 、 $N_{\text{tn}}$ 、 $N_{\text{fp}}$  和  $N_{\text{fn}}$  分别为真正例、真负例、假正例和假负例。时间预测任务是一个回归问题, 采用 MAE 进行评估, MAE 可以用于误差相互抵消的情况, 因而可以准确反映实际预测误差的大小。其基本的数学表达式为

$$M_{\text{AE}} = \frac{1}{D} \sum_{d=1}^D |y_d - \hat{y}_d|, \quad (16)$$

式中:  $D$  为事件日志的总个数,  $d$  为事件日志序号。MAE 作为误差度量, 越低越好。因为方均根误差 (RMSE) 对数据点上异常值的错误非常敏感, 而如果两个活动之间的时间间隔非常大, RMSE 会认为是异常值, 所以采用 MAE 进行时间预测评估。

### 4.3 实验结果与分析

在两个数据集上评估预测活动和时间的多任务学习性能, 将数据集中 75% 的事件日志作为训练集, 在训练集中分出 10% 的数据作为验证集, 将 25% 的事件日志作为测试集。表 2 显示 Self-Att-BiLSTM 多任务预测深度网络框架在 Helpdesk 数据集、BPI12 事件日志及三个子流程上关于预测活动的准确性 (Accuracy)、关于预测时间方面的

MAE。分别采用不同长度的事件轨迹, 通过取平均值进行多任务预测, 对实验结果、Tax 等<sup>[16]</sup> 多任务研究结果、Breuker 等<sup>[9]</sup> 单个预测活动结果进行对比。并且在每个数据集中筛选出含有重复活动的事件轨迹, 单独对 Tax 等<sup>[16]</sup> 的方法与所提方法进行对比。图 4 为两种方法对重复活动预测精度的对比, 实验在不同数据集上对于重复活动的预测精度均比 Tax 等<sup>[16]</sup> 的预测精度有明显提高。在 BPI12\_O 数据集上, 所提方法的准确率提高最为明显, 为 25.5 个百分点, 只有在 BPI12\_W 数据集上略显不足, 原因在于 BPI12\_W 为人为手动执行事件, 存在多变因素, 使得模型没有学习到明显特征。实验结果表明, 在重复活动预测方面, 所提方法是可行的。

表 2 不同方法的结果对比

Table 2 Result comparison of different methods

Dataset	Method	Accuracy / %	MAE / day
BPI12	Self-Att-BiLSTM	80.57	1.39
	Method in Ref. [16]		
	Method in Ref. [9]		
BPI12_W	Self-Att-BiLSTM	78.21	1.45
	Method in Ref. [16]	76.00	1.56
	Method in Ref. [9]	71.90	
BPI12_O	Self-Att-BiLSTM	81.64	2.36
	Method in Ref. [16]		
	Method in Ref. [9]	81.10	
BPI12_A	Self-Att-BiLSTM	81.33	0.99
	Method in Ref. [16]		
	Method in Ref. [9]	80.10	
Helpdesk	Self-Att-BiLSTM	81.13	2.05
	Method in Ref. [16]	71.23	3.75
	Method in Ref. [9]		

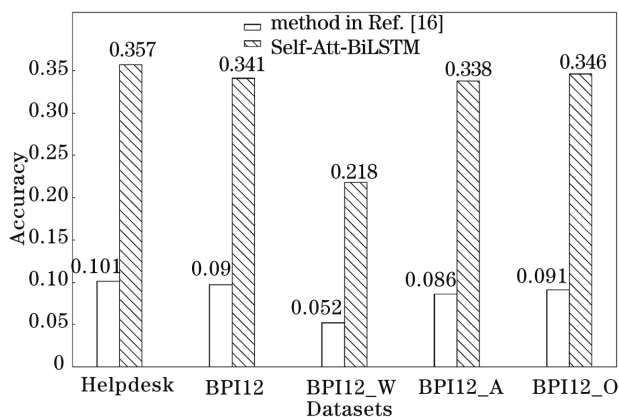


图 4 不同数据集中重复活动预测精度对比

Fig. 4 Comparison of prediction accuracy of repeated activities on different datasets

从表 2 可知: 在不同数据集上, 所提方法的活动精度均比 Tax 方法的高, 尤其在 Helpdesk 数据集上更为明显, 所提方法的活动预测精度比 Tax 方法提高了 9.9 个百分点, 在 BPI12\_W 上提高了 2.21 个百分点。因为 BPI12\_W 的流程是人工手动执行的且流程执行顺序比较复杂, 所以所提方法在活动精度方面提高不明显。所提方法在 BPI12 数据集上的活动精度略比三个子流程的活动精度平均值高约 0.18 个百分点。在时间 MAE 方面, 所提方法的 MAE 均比 Tax 方法小, 在 Helpdesk 数据集上, MAE 降低了 1.7, 在 BPI12\_W 上, MAE 仅仅降低了 0.11。所提方法在数据集 BPI12 上的时间 MAE 比三个子流程的时间 MAE 平均值低 0.21。从表 2 可以看出, 数据集的完整度在很大程度上决定预测的准确度, 在

BPI12\_O 中,所提方法的时间 MAE 表现较差的原因是时间数据间隔多变,导致预测时间 MAE 高。

图 5 为不同数据集上的多任务模型训练损失图,对于图 5(d),有些过拟合,导致预测活动精度不高。

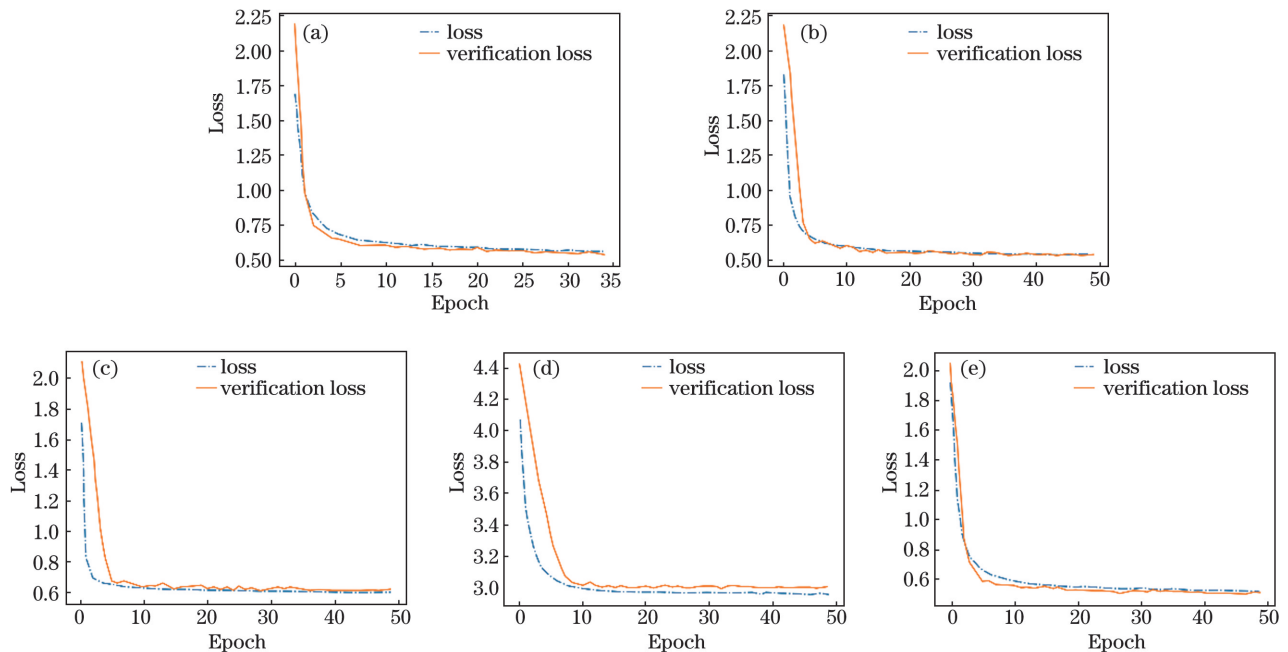


图 5 不同数据集上的模型训练损失图。(a) BPI12; (b) BPI12\_O; (c) BPI12\_A; (d) BPI12\_W; (e) Helpdesk  
Fig. 5 Model training loss under different datasets. (a) BPI12; (b) BPI12\_O; (c) BPI12\_A; (d) BPI12\_W; (e) Helpdesk

对多任务与 Breuker 等<sup>[9]</sup>单个预测活动任务进行对比,所提方法在活动预测方面均比 Breuker 方法预测结果高。从表 2 可以看出,所提方法在 BPI12\_W 的活动精度比 Breuker 方法高 6.31 个百分点。通过与单任务对比发现,多任务预测可以同时提升活动与时间的预测精度,使模型具有更好的泛化能力。结果证明,所提 Self-Att-BiLSTM 模型进行活动和时间多任务预测时具有一定的可行性。

## 5 结 论

提出了一种 Self-Att-BiLSTM 深度神经网络模型,用来进行业务流程活动和时间多任务预测。该方法将业务流程中的事件日志作为输入,将事件日志中的属性编码为 one-hot,并转换为 embeddings 向量,应用 Self-Att-BiLSTM 模型进行多任务预测。BiLSTM 有助于对事件日志序列进行建模,而 Self-Attention 则有助于通过不同的权重从序列中提取出信息。通过共享相关任务之间的表示,模型能更好地泛化学习任务,使得每一个任务的预测效果都有提升。最后对预测活动和时间有效性进行实验验证,结果表明,所提方法在活动和时间多任务方面都有所提升,尤其在重复活动方面更加显著。未来将进一步扩展其他事件属性的预测,进一步提高活动和时间预测的精度。

## 参 考 文 献

- [1] van der Aalst W M P, Weijters A J M M. Process mining: a research agenda [J]. Computers in Industry, 2004, 53(3): 231-244.
- [2] van der Aalst W M P. Process mining: discovery, conformance and enhancement of business processes [M]. Wang J M, Wen L J, Transl. Beijing: Tsinghua University Press, 2014.  
van der Aalst W M P. 过程挖掘: 业务过程的发现、合规和改进[M]. 王建民, 闻立杰, 译. 北京: 清华大学出版社, 2014.
- [3] Pravilovic S, Appice A, Malerba D. Process mining to forecast the future of running cases [M] // Appice A, Ceci M, Loglisci C, et al. New frontiers in mining complex patterns. Lecture notes in computer science. Cham: Springer, 2014, 8399: 67-81.
- [4] Kang B, Kim D, Kang S H. Periodic performance prediction for real-time business process monitoring [J]. Industrial Management & Data Systems, 2012, 112(1): 4-23.
- [5] Rogge-Solti A, Weske M. Prediction of remaining service execution time using stochastic petri nets with arbitrary firing delays [M] // Basu S, Pautasso C, Zhang L, et al. Service-oriented computing. Lecture notes in computer science. Heidelberg: Springer, 2013, 8274: 389-403.

- [6] Zheng T T, Chen J X, Xu Y, et al. Approach for individual task completion time prediction in business processes [J]. *Computer Integrated Manufacturing Systems*, 2019, 25(4): 993-1000.  
郑婷婷, 陈洁璇, 许洋, 等. 业务流程中一种个性化的任务完成时间预测方法 [J]. *计算机集成制造系统*, 2019, 25(4): 993-1000.
- [7] Maggi F M, di Francescomarino C, Dumas M, et al. Predictive monitoring of business processes [M] // Jarke M, Mylopoulos J, Quix C, et al. *Advanced information systems engineering. Lecture notes in computer science*. Cham: Springer, 2014, 8484: 457-472.
- [8] van der Aalst W M P, Schonenberg M H, Song M. Time prediction based on process mining [J]. *Information Systems*, 2011, 36(2): 450-475.
- [9] Breuker D, Group H, Matzner M, et al. Comprehensible predictive models for business processes [J]. *MIS Quarterly*, 2016, 40(4): 1009-1034.
- [10] Lakshmanan G T, Shamsi D, Doganata Y N, et al. AMarkov prediction model for data-driven semi-structured business processes [J]. *Knowledge and Information Systems*, 2015, 42(1): 97-126.
- [11] Conforti R, de Leoni M, La Rosa M, et al. Supporting risk-informed decisions during business process execution [M] // Salinesi C, Norrie M C, Pastor O. *Advanced information systems engineering. Lecture notes in computer science*. Cham: Springer, 2013, 7908: 116-132.
- [12] Leontjeva A, Conforti R, di Francescomarino C, et al. Complex symbolic sequence encodings for predictive monitoring of business processes [M] // Motahari-Nezhad H R, Recker J, Weidlich M. *Business process management. Lecture notes in computer science*. Cham: Springer, 2015, 9253: 297-313.
- [13] Evermann J, Rehse J R, Fettke P. A deep learning approach for predicting process behaviour at runtime [M] // Dumas M, Fantinato M. *Business process management workshops. Lecture notes in business information processing*. Cham: Springer, 2017, 281: 327-338.
- [14] Greff K, Srivastava R K, Koutnik J, et al. LSTM: a search space odyssey [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, 28(10): 2222-2232.
- [15] Zhu M K, Lu X L. Human action recognition algorithm based on Bi-LSTM-Attention model [J]. *Laser & Optoelectronics Progress*, 2019, 56(15): 151503.  
朱铭康, 卢先领. 基于 Bi-LSTM-Attention 模型的人体行为识别算法 [J]. *激光与光电子学进展*, 2019, 56(15): 151503.
- [16] Tax N, Verenich I, La Rosa M, et al. Predictive business process monitoring with LSTM neural networks [M] // Dubois E, Pohl K. *Advanced information systems engineering. Lecture notes in computer science*. Cham: Springer, 2017, 10253: 477-492.
- [17] Navarin N, Vincenzi B, Polato M, et al. LSTM networks for data-aware remaining time prediction of business process instances [C] // 2017 IEEE Symposium Series on Computational Intelligence (SSCI), November 27-December 1, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 477-492.
- [18] Tello-Leal E, Roa J, Rubiolo M, et al. Predicting activities in business processes with LSTM recurrent neural networks [C] // 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K), November 26-28, 2018, Santa Fe. New York: IEEE Press, 2018.
- [19] Ruder S. An overview of multi-task learning in deep neural networks [EB/OL]. (2017-06-15) [2020-06-04]. <http://ArXiv abs/1706.05098>.
- [20] Yang H X, Chen F, Gan W F. Deep face recognition algorithm based on multitask learning [J]. *Laser & Optoelectronics Progress*, 2019, 56(18): 181005.  
杨恢先, 陈凡, 甘伟发. 基于多任务学习的深层人脸识别算法 [J]. *激光与光电子学进展*, 2019, 56(18): 181005.
- [21] Hou C P, Jiang T L, Lang Y, et al. Human activity and identity multi-task recognition based on convolutional neural network using Doppler radar [J]. *Laser & Optoelectronics Progress*, 2020, 57(2): 021009.  
侯春萍, 蒋天丽, 郎玥, 等. 基于卷积神经网络的雷达人体动作与身份多任务识别 [J]. *激光与光电子学进展*, 2020, 57(2): 021009.
- [22] Zheng G, Mukherjee S, Dong X L, et al. OpenTag: open attribute value extraction from product profiles [C] // *Proceeding of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 19-23, 2018, London, England. New York: ACM Press, 2018: 1049-1058.