

深度卷积记忆网络时空数据模型

秦 超¹ 高晓光¹ 万开方¹

摘 要 时空数据是包含时间和空间属性的数据类型。研究时空数据需要设计时空数据模型,用以处理数据与时间和空间的关系,得到信息对象由于时间和空间改变而产生的行为状态变化的趋势。交通信息数据是一类典型的时空数据。由于交通网络的复杂性和多变性,以及与时间和空间的强耦合性,使得传统的系统仿真和数据分析方法不能有效地得到数据之间的关系。本文通过对交通数据中临近空间属性信息的处理,解决了由于传统时空数据模型只关注时间属性导致模型对短时间间隔数据预测能力不足的问题,进而提高模型预测未来信息的能力。本文提出一个全新的时空数据模型—深度卷积记忆网络。深度卷积记忆网络是一个包含卷积神经网络和长短时间记忆网络的多元网络结构,可以提取数据的时间和空间属性信息,通过加入周期和镜像特征提取模块对网络进行修正。通过对两类典型时空数据集的验证,表明深度卷积记忆网络在预测短时间间隔的数据信息时,相较于传统的时空数据模型,不仅预测误差有了很大程度的降低,而且模型的训练速度也得到提升。

关键词 时空数据模型,深度卷积记忆网络,时间特征,空间特征

引用格式 秦超,高晓光,万开方.深度卷积记忆网络时空数据模型.自动化学报,2020,46(3):451–462

DOI 10.16383/j.aas.c180788

Deep Spatio-temporal Convolutional Long-short Memory Network

QIN Chao¹ GAO Xiao-Guang¹ WAN Kai-Fang¹

Abstract Spatio-temporal data is a data type that contains temporal and spatial attributes. Training spatio-temporal data needs spatio-temporal models to deal with the relationship between data and those two attributes, to get the trend as it changes with time and space. Traffic information data is a typical type of spatio-temporal data. Due to the complexity and variability of the traffic network, and the strong coupling with time and space, traditional system simulation and data analysis methods cannot effectively obtain the relationship between data. By dealing with the adjacent spatial attribute information in traffic data, this paper designs a new spatio-temporal model, DSTCL (deep spatio-temporal convolutional long-short memory network), to solve the problem that traditional spatio-temporal models only pay attention to the temporal attribute and these models have insufficient ability to predict short-term data, and then improve the ability to predict future information. DSTCL is a multi-network structure consisting of a convolutional neural network and a long short-term memory. It can extract the temporal and spatial attribute information of the data, and correct the network by adding the periodic feature extraction module and the mirror feature extraction module. By verifying the two types of typical spatio-temporal datasets, it is shown that when DSTCL predicts the short-term information, compared with the traditional spatio-temporal models, not only the prediction error is greatly reduced, but also the training speed has been improved.

Key words Spatio-temporal model, deep spatio-temporal convolutional long-short memory network (DSTCL), temporal attribute feature, spatial attribute feature

Citation Qin Chao, Gao Xiao-Guang, Wan Kai-Fang. Deep spatio-temporal convolutional long-short memory network. *Acta Automatica Sinica*, 2020, 46(3): 451–462

随着数据处理技术以及数据采集设备(如传感器或移动采集装置)的发展,具备时间和位置属性

的时空数据成为大数据时代典型的数据类型^[1]。设计时空数据模型,用以对时空数据进行时间和空间维度的处理,可以得到反映对象由于时间和空间改变而产生的行为状态变化的信息^[2–4],这类信息可以有效地预测对象未来的状态。

交通信息数据与时间和空间两类属性的密切联系,可以说是一类典型的时空数据。并且交通信息数据不同属性之间具有不同程度耦合性,处理起来比较复杂,所以目前研究时空模型的学者们主要工

收稿日期 2018-11-27 录用日期 2019-06-18

Manuscript received November 27, 2018; accepted June 18, 2019

国家自然科学基金(61573285)资助

Supported by National Natural Science Foundation of China (61573285)

本文责任编辑 陈德旺

Recommended by Associate Editor CHEN De-Wang

1. 西北工业大学电子信息学院 西安 710100

1. School of Electronics and Information Engineering, Northwestern Polytechnical University, Xi'an 710100

作在于设计时空数据模型以便有效地处理交通信息数据. 对交通信息进行研究可以提供准确的公路状况信息给决策者, 从而使决策者对公路进行良性管理.

处理交通数据的时空数据模型主要有两类, 系统仿真模型和数据分析模型.

系统仿真模型的原理是分析交通系统的内部结构, 建立一个白盒用来模拟系统的运行, 预测未来某一时刻的数据^[5]. Vlahogianni^[6]通过分析交通系统内各个部分之间的关系, 设计了一个计算智能模型 (Computational intelligence). 该模型可以解决由于交通数据输入维度增大导致解空间过大的问题, 并可以有效地处理大规模多目标数据. 由于系统仿真模型在处理诸如交通网络等复杂系统时不能有效地分析各个部分之间的隐藏关系, 所以学者们的关注点主要在对已经采集得到的交通数据设计数据分析模型.

数据分析模型将统计学和机器学习相结合, 建立对交通系统的黑盒模拟^[7]. Ahmed 等^[7]使用 AR-IMA (Autoregressive integrated moving average) 模型将交通信息中的时空数据转化成时序数据, 通过数据差分将非平滑时间序列转化成平滑时间序列, 使输出变量仅对其滞后量以及随机误差项的当前值和滞后量进行回归处理, 从而建立数据分析模型. Williams 等^[8]在 ARIMA 基础上提出了 S-AR-IMA (Seasonal ARIMA) 算法, 改变 ARIMA 模型滞后量的计算方法, 降低了对模式的识别能力和预测长时间间隔 (Long-term) 数据的预测误差. Lippi 等^[9]在 ARIMA 的基础结构上加入了一个卡尔曼滤波器, 提高了模型在解决复杂问题时的准确率. Kumar 等^[10]通过对动脉数据进行实验分析, 说明了 S-ARIMA 算法能够有效地处理对时间属性不敏感的数据信息. 由于 ARIMA 本质上是将交通信息当成一种时序数据使用差分方法处理, 使得算法对时间属性不敏感, 而且由于交通数据不同属性之间具有强耦合性, 模型容易产生一些非线性问题, 导致 ARIMA 及其衍生模型在处理高速改变的交通数据流时具有一些局限性, 在预测短时间间隔 (Short-term) 数据时表现较差^[10].

学者们通过非参数 (Non-parametric) 的机器学习技巧, 尝试解决 ARIMA 很难解决的这些非线性问题, 提高模型预测短时间间隔数据的能力. 神经网络是典型的非参数机器学习网络. 相对于传统的 ARIMA 模型, 神经网络具有很多优势. 首先体现在神经网络可以扩展到上千节点, 使得模型以比较大的精度逼近真实模型. 其次, 神经网络中含有非参数结构, 可以更灵活地处理输入数据. 更重要的是由于神经网络没有对数据进行差分处理, 可以

有效地处理高速改变的交通信息流. Dougherty 等^[11]使用神经网络处理交通信息, 包括司机的行为信息、交通参数估计、交通信息的预测等, 在理论和实验这两方面说明了神经网络处理交通数据的可行性. Hua 等^[12]通过实验证明使用人工神经网络预测短时间间隔之后的汽车运行时间, 预测误差比传统的 ARIMA 的方法要小. Smith 等^[13]使用 BP (Back propagation) 神经网络处理动态信息, 对比于传统的非参数回归模型和 ARIMA 模型, 在预测短时间间隔数据时预测误差得到了很大程度的降低. Chan 等^[14]说明神经网络在预测交通流密度方面具有获取数据非线性特征的能力, 并说明神经网络在交通数据处理方面已经得到了广泛应用.

Hinton 等^[15]改进了深度学习算法的训练机制, 使得深度学习算法受到学术界和工商界的广泛关注. 深度学习模型在图像处理、语音识别、文本翻译等领域取得了巨大成功^[16]. 在交通信息领域, 由于深度学习算法可以从大数据集中学习到有效的特征, 许多学者尝试使用该算法对海量的交通数据进行处理. Polson 等^[17]使用了深度结构的神经网络来预测交通流, 通过对 Interstate I-55 号道路上获取的数据进行训练, 预测短期的交通流密度. 这篇文献还对预测中出现的两处异常值做出了合理的解释. 对比于传统的 BP 神经网络, 在模型的收敛速度上得到提升. Jia 等^[18]通过使用多层的受限玻尔兹曼机模型, 建立以 MAPE (Mean absolute percentage error)、RMSE (Root mean square error) 以及 RMSN (Normalized root mean square error) 为预测误差的深度信念网络 (Deep belief network, DBN) 模型, 用来预测短时间间隔交通流速度, 实验结果表明使用 DBN 得到的预测误差比 BP 神经网络和 ARIMA 模型要小, 说明 DBN 在预测短时间间隔时空数据时效果更好. Lv 等^[19]建立堆叠自适应编码器 (Stacked autoencoders, SAEs) 网络预测短时交通流密度, 实验所得预测误差比 BP 神经网络和支持向量机 (Support vector machine, SVM) 等浅层网络模型以及 S-ARIMA 低. 类比图像数据和声音采样信息, 交通流数据在空间和时间领域具有很多有用的特征^[20]. 由于这些深度学习模型很难学习到数据有关时间和空间关系的良好特征表达, 在预测长时间间隔的交通数据的时候表现乏力, 限制了模型的泛化能力.

循环神经网络 (Recurrent neural network, RNN) 是一类用于处理序列数据的神经网络. RNN 中的循环单元可以很好地记忆前某个时刻网络学习到的数据信息, 可以学习到输入数据中时间属性的特征^[21]. 但传统的 RNN 网络在处理交通信

息时存在一些不足: 1) RNN 网络中的时间迭代步长需要在网络设计前给定. 2) RNN 在处理预测长时间间隔交通数据的时候表现乏力^[22]. Ma 等^[22]针对传统 RNN 存在的问题, 使用 RNN 中一种特殊结构的长短期记忆 (Long short-term memory, LSTM) 预测长时间间隔 (Long-term) 的交通流密度, 通过实验对比 LSTM 和传统 RNN 以及其他深度学习模型在预测长时间间隔的交通数据方面的能力, 结果说明了 LSTM 比传统 RNN 和其他深度学习模型在预测长时间间隔数据时效果更好.

ARIMA、神经网络、传统的深度学习网络以及 RNN 等模型, 重点关注数据时间属性的信息, 缺乏对临近空间信息的分析, 没有从本质上解决短时间间隔预测能力不足的问题. 临近空间数据对预测信息的变化趋势影响较大, 所以如果需要有效地预测高速改变的信息流, 应重点关注该预测点临近位置的数据. 在研究临近空间位置数据的时候, 可以将某一个时刻所有位置的数据信息做一个网格, 采用一些特定的结构进行处理. 网格中所有数据不是完全独立的, 位置距离越近, 数据的相关性越强, 而使用传统的全连接的网络结构很难学习到这个特征.

本文提出一个全新的时空数据模型——深度卷积记忆网络 (Deep spatial-temporal convolutional LSTM, DSTCL), 建立包含卷积神经网络和长短时间记忆网络的多元网络结构, 相对于传统的时空数据模型, 加入对模型的空间属性信息的处理, 重点解决模型预测中短时间间隔数据的误差较大的问题. 卷积神经网络 (Convolutional neural network, CNN) 在处理类似网格结构数据的时候, 可以高效地学习到临近空间信息的相关性以及输入变量不同区域之间的关系等信息^[23-24]. 短时间记忆网络可以学习到数据的记忆特征, 有效地处理时间属性信息. 多元网络 DSTCL 能够同时提取到数据的时间和空间属性信息, 更有效对长时间间隔、中时间间隔以及短时间间隔的数据进行预测. 考虑到交通数据的周期属性以及一天中镜像时间段的影响, 在 DSTCL 中加入了周期特征提取模块和镜像特征提取模块对网络进行修正.

本文结构如下: 第 1 节对需要处理的问题进行概述, 第 2 节给出模型设计的细节, 第 3 节给出实验结果和分析, 第 4 节是结论和展望.

1 问题描述

本文的目标是处理已经采集到的交通速度数据, 预测某一个位置未来某一个时刻的交通速度信息. 用公式表示为

$$v_{T,P} = \arg \max_v p(v_{T,P} | v_{t=T-n, p \neq P, N \geq n \geq 0}, v_{t=T-n, p=P, N \geq n \geq 0}, \theta) \quad (1)$$

其中, $v_{T,P}$ 表示 T 时刻 P 位置的速度, $v_{t=T-n, p \neq P, N \geq n \geq 0}$ 表示除 P 位置之外的地点 T 时刻之前 (包含 T 时刻) 的速度值, $v_{t=T-n, p=P, N \geq n \geq 0}$ 表示 P 位置 T 时刻之前 (不包括 T 时刻) 的速度值, N 表示输入时间步长, θ 表示模型的参数值.

如果预测 T 时刻 P 位置的速度, 按照式 (1), 需要其他地点 T 时刻之前的速度值, P 位置 T 时刻之后的速度值, 仍使用 N 表示输入时间步长, 如图 1 所示, 我们得到交通信息的数据矩阵.

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_P \end{bmatrix} = \begin{bmatrix} v_1(T-N) & \cdots & v_1(T-1) \\ \vdots & \ddots & \vdots \\ v_P(T-N) & \cdots & v_P(T-1) \end{bmatrix} \quad (2)$$

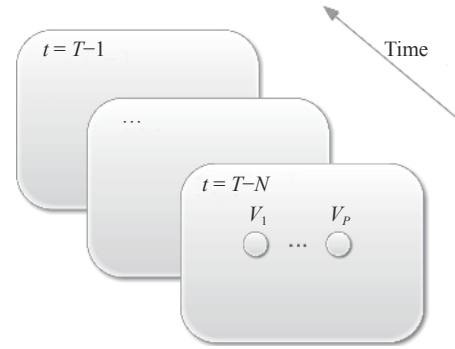


图 1 按照时间顺序对不同位置的交通数据进行处理
Fig. 1 Processing traffic data at different locations in chronological order

2 深度卷积记忆网络

深度卷积记忆网络主要任务是训练数据的空间特征和时间特征, 得到预测信息随时间和空间的变化趋势, 进而降低模型的预测误差, 提高训练速度. 此外, 还需要训练其他的特征, 如周期特征和镜像特征等, 提高模型的预测能力.

2.1 训练数据的空间特征

由于交通流的速度总是与临近位置的速度相关, 而且卷积网络在表达临近区域特征相关问题方面具有非常大的优势, 我们选择一维的卷积网络来抽取数据的空间特征信息, 如图 2 所示. 由于建模需要, 在使用卷积网络的时候不考虑时间这个属性, 仅使用上一个时间点的不同位置的速度, 舍弃掉其他时间点的速度. 将问题重新整理如下:

$$v_P = \arg \max_v p(v_P | v_{p, t=T-1}, \theta) \quad (3)$$

从数据矩阵中抽取

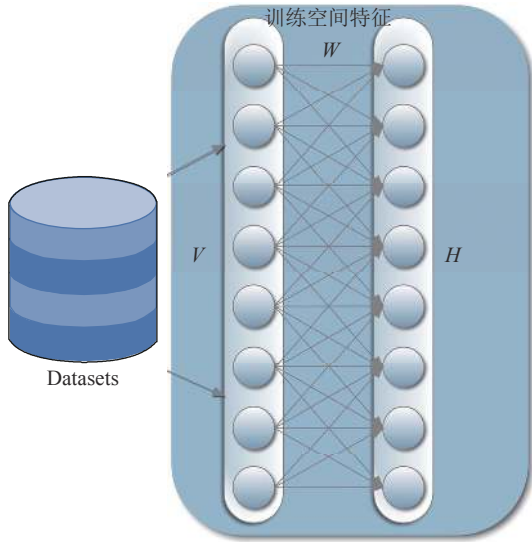


图2 使用 CNN 训练空间特征

Fig. 2 Training spatial features with CNN

$$v_p = [v_1(T-1), v_2(T-1), \dots, v_P(T-1)]^T \quad (4)$$

作为一维卷积网络的输入. 使用一维卷积网络处理数据集如图2所示, 第 i 个特征表达如下:

$$h_P^i = o(w_q^i * v_q^i + b_q^i) \quad (5)$$

其中, w_q^i 表示权值, o 表示非线性的激活矩阵, b_q^i 表示偏置, $*$ 表示卷积计算.

对一般的卷积神经网络来说, 卷积计算之后需要进行池化处理. 池化主要作用是保持图片的等变性. 对于小图像块识别这一类的任务来说, 不使用池化运算也可以有效处理^[25]. 由于交通数据集可以看成像素点比较少的图片 (小图像块) 也不需要考虑其等变性, 所以在本模型中, 卷积运算之后不使用池化处理. 具体设计上, 我们在卷积层对输入用零进行填充使其加宽, 保证输入和输出在维度上具有相同的大小. 这样可以使卷积运算不改变下一层的结构, 只要硬件支持, 该网络就可以包含任意数量的卷积层, 便于网络进行扩展. 由于空间特征提取模块包含卷积层的层数较少, 使用残差结构^[26] 效果提升不明显, 所以使用普通的 CNN 结构.

2.2 训练数据的时间特征

RNN 是专门用于处理类似序列 $x^1, x^2, \dots, x^T$ 的神经网络. 正如 CNN 可以很容易地扩展到具有很大宽度和高度的图像, 以及可以用来处理大小可变的图像, RNN 可以扩展到更长的序列 (比不基于序列的特化网络长得多). 一个 RNN 网络如图3所示, 其中 x 表示输入值, h 表示隐层值, o 表示模型输出值, L 表示代价损失函数, y 表示对应 x 的标签值, U 表示 x 的权值, W 表示记忆单元的权值, V 表

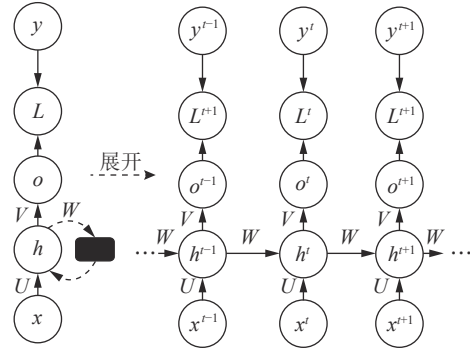


图3 循环神经网络的计算图模型

Fig. 3 Calculation graph model of RNN

示隐层的权值.

RNN 处理时序数据的一个优势就是可以学习到数据的记忆特征, 不再单独地进行每个时刻数据的训练, 而是通过隐层或者输出层将不同时刻的训练步骤联系起来, 即当前时刻的训练需要前几个时刻的数据提供记忆信息.

但由于传统的 RNN 的局限性, 预测长时间间隔的交通数据表现乏力^[22]. 而 LSTM 在预测长时数据时表现良好, 对预测间隔的大小限制较少, 所以我们选择 LSTM 作为训练数据时间特征的网络模型.

LSTM 对传统的 RNN 一个关键的扩展是使自循环的权重 (图3中的 W) 视上下文而定, 而不是固定的^[27]. 门控此自循环 (门控函数由另一个隐层单元控制) 的权重, 由于时间量是模型本身的输入, 使得模型累计的时间尺度可以动态改变.

LSTM 块如图4所示. LSTM 除了外部的 RNN 循环外, 还有内部的 LSTM “细胞” 循环, 因此 LSTM 不是简单地向输入和循环单元的仿射变换之后施加一个逐元素的非线性. LSTM “细胞” 彼此循环连接, 代替一般循环网络中普通的隐藏单元. 状态单元具有线性自循环, 其权重由遗忘门控制. “细胞” 的输出可以被输入门关闭. 所有的门控单元都具有

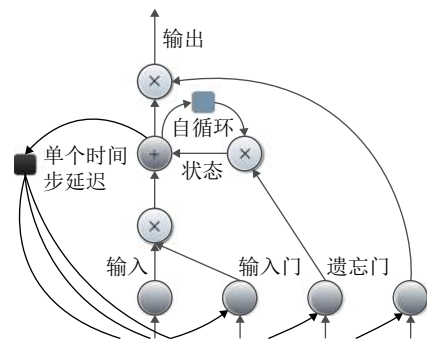


图4 LSTM “细胞” 结构框图

Fig. 4 Structure of LSTM cell

非线性激活函数, 而输入单元可具有任意的压缩非线性. 状态单元也可以用作门控单元的额外输入. 黑色方块表示单个时间步的延迟.

t 时刻第 i 个细胞的遗忘门 f_i^t , 输入门 g_i^t , 状态单元 s_i^t , 输出门 q_i^t 以及最后的输入 h_i^t 的表达如下:

$$f_i^t = \sigma \left(b_i^f + \sum_j U_{i,j}^f x_j^t + \sum_j W_{i,j}^f h_j^{t-1} \right) \quad (6)$$

$$s_i^t = g_i^t \sigma \left(b_i + \sum_j U_{i,j} x_j^t + \sum_j W_{i,j} h_j^{t-1} \right) + f_i^t s_i^{t-1} \quad (7)$$

$$g_i^t = \sigma \left(b_i^g + \sum_j U_{i,j}^g x_j^t + \sum_j W_{i,j}^g h_j^{t-1} \right) \quad (8)$$

$$q_i^t = \sigma \left(b_i^o + \sum_j U_{i,j}^o x_j^t + \sum_j W_{i,j}^o h_j^{t-1} \right) \quad (9)$$

$$h_i^t = \tanh(s_i^t) q_i^t \quad (10)$$

其中, x^t 表示当前的输入量, h^t 表示当前的隐层值, h^t 包含所有 LSTM 细胞的输出. b^f , U^f 和 W^f 分别是遗忘门的偏置、输入权重和循环权重. b^g , U^g 和 W^g 分别是输入门的偏置、输入权重和循环权重. b^o , U^o 和 W^o 分别是输出门的偏置、输入权重和循环权重. b , U 和 W 分别是 LSTM “细胞” 中的偏置、输入权重和循环权重.

在处理交通数据的时候, LSTM 可以有效地解决长期依赖问题, 对输入数据进行训练^[22]. 将问题按照时间属性整理为

$$v_T = \arg \max_v p(v_T | v_{t=t-N, N \geq n > 0}, \theta) \quad (11)$$

按照式 (11) 从数据矩阵抽取

$$V = (V_0, V_1, \dots, V_q, \dots, V_{N-1}) \quad (12)$$

其中, $V_q = [v_1(t - N + q), v_2(t - N + q), \dots, v_P(t - N + q)]^T$.

将 V_q 作为输入代入式 (6)~(10), 使用 LSTM 训练数据集如图 5 所示.

2.3 训练其他特征

在实验过程中, 发现不同日期同一时刻的速度数据可以很大程度上影响训练的效果. 原因可能是人们通常喜欢在某天的同一时刻或者该时刻的临近时间去做相同的事情, 所以另外使用堆叠自动编码器来训练同一时刻的数据. 堆叠自动编码器被证明可以有效地训练交通数据周期信息^[19]. 我们按照周

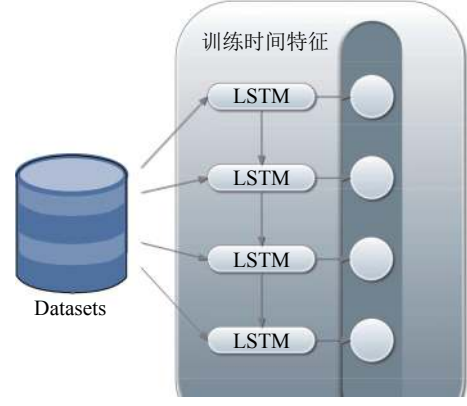


图 5 使用 LSTM 训练时间特征

Fig. 5 Training temporal features with LSTM

同一时刻和日同一时刻两个属性重新抽取数据矩阵.

$$v^d = \begin{bmatrix} v_1(t^d - N \times m) & v_1(t^d - (N-1) \times m) & \dots & v_1(t^d - m) \\ v_2(t^d - N \times m) & v_2(t^d - (N-1) \times m) & \dots & v_2(t^d - m) \\ \vdots & \vdots & \ddots & \vdots \\ v_P(t^d - N \times m) & v_P(t^d - (N-1) \times m) & \dots & v_P(t^d - m) \end{bmatrix} \quad (13)$$

$$v^w = \begin{bmatrix} v_1(t^w - N \times k) & v_1(t^w - (N-1) \times k) & \dots & v_1(t^w - k) \\ v_2(t^w - N \times k) & v_2(t^w - (N-1) \times k) & \dots & v_2(t^w - k) \\ \vdots & \vdots & \ddots & \vdots \\ v_P(t^w - N \times k) & v_P(t^w - (N-1) \times k) & \dots & v_P(t^w - k) \end{bmatrix} \quad (14)$$

其中, v^d 表示按照日同一时刻抽取数据, v^w 表示按照周同一时刻抽取数据. m 表示原始数据中每天抽取数据的个数. k 表示原始数据中每周抽取数据的个数. 按照图 6, 将这两个数据矩阵作为堆叠自动编

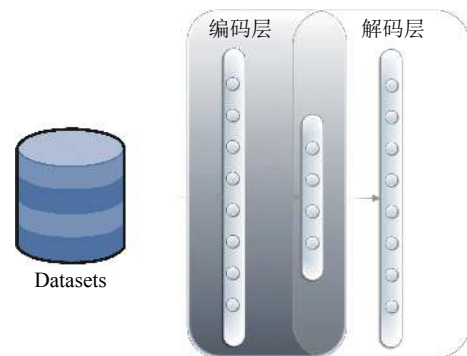


图 6 堆叠自动编码器训练周期特征

Fig. 6 Training periodic features with stacked auto-encoder

码器的输入。

由于同一天的数据中,上午时刻数据对下午时刻数据的影响很大,对下午时刻的数据进行处理的时候需要考虑上午数据的特征,这种特征称为镜像特征。重新抽取数据矩阵如下:

$$v^m = \begin{bmatrix} v_1(H_1) & v_1(H_2) & \cdots & v_1(H_\lambda) \\ v_2(H_1) & v_2(H_2) & \cdots & v_2(H_\lambda) \\ \vdots & \vdots & \ddots & \vdots \\ v_P(H_1) & v_P(H_2) & \cdots & v_P(H_\lambda) \end{bmatrix} \quad (15)$$

其中, λ 表示上午抽取的数据的个数, H_x 表示当天抽取的第 x 个数据。将该矩阵输入到堆叠自动编码器里,训练镜像特征。

2.4 建立模型

我们将第 2.1 节中的网络结构命名为空间特征提取模块;第 2.2 节中的网络结构命名为时间特征提取模块;第 2.3 节中的网络结构命名为周期和镜像特征提取模块。将上述 3 个模块输入到一个多层神经网络进行回归分析,如图 7 所示。整个网络采

用多元的结构,兼顾了时间和空间两个属性的信息,我们将其命名为深度卷积记忆网络。

通过加入输入门,控制不同时间段的输入进入镜像特征训练模块。门控函数为

$$s = \begin{cases} 0, & x \leq \lambda \\ 1, & x > \lambda \end{cases} \quad (16)$$

其中, x 表示输入数据在当天的批次, λ 表示上午抽取的批次。

3 实验

3.1 数据集

我们使用数据集 PeMSD7 (Caltrans performance measurement system district 7) 和 FMD (Fire material dataset) 来验证算法的有效性。这两个数据集中都包含时间信息和空间信息,并且这两类信息对数据之间的关系影响较大。

PeMSD7 数据集是由 Caltrans performance measurement system 中的 39 000 个独立的传感器实时搜集数据。这些传感器遍布在美国加利福尼亚州主要都市区高速公路附近^[28]。我们将数据集中的实时数据初步预处理成 128 个临近的传感器累计收集到时间间隔为 5 min 的数据,选择数据集的时间跨度为 2015 年 6 月到 2016 年 6 月。

初步预处理 PeMSD7 数据集之后对数据集中的缺失值进行插值填充。然后将数据集横轴按照时间顺序纵轴按照空间顺序进行排序。数据集 70% 的数据作为训练集,20% 的数据作为验证集,10% 的数据作为测试集。验证集的作用是防止模型过拟合。在训练集进行迭代的同时将每一次迭代得到的模型用于计算验证集的预测误差,绘出验证集的预测误差曲线,当验证集的预测误差曲线开始上升时,记录此时模型的迭代次数,并与训练集预测误差曲线收敛所需要的迭代次数进行对比,选择最小的值作为停止训练所需要的迭代次数,对应于最优的模型参数。

FMD 数据集共有 42 组实验,每次实验由 80 个热传感器和 100 个压力传感器实时收集材料着火后的温度和气压。我们将 FMD 数据集中的实时数据初步预处理成间隔 2 min 的数据。预处理之后,数据集有 24 000 条数据,把标签属性设置为消防喷头的开启状态。数据集 80% 的数据作为训练集,10% 的数据作为验证集,10% 的数据作为测试集。

3.2 实验设计

实验具体的硬件配置为: GPU 为 NVIDIA Ge-

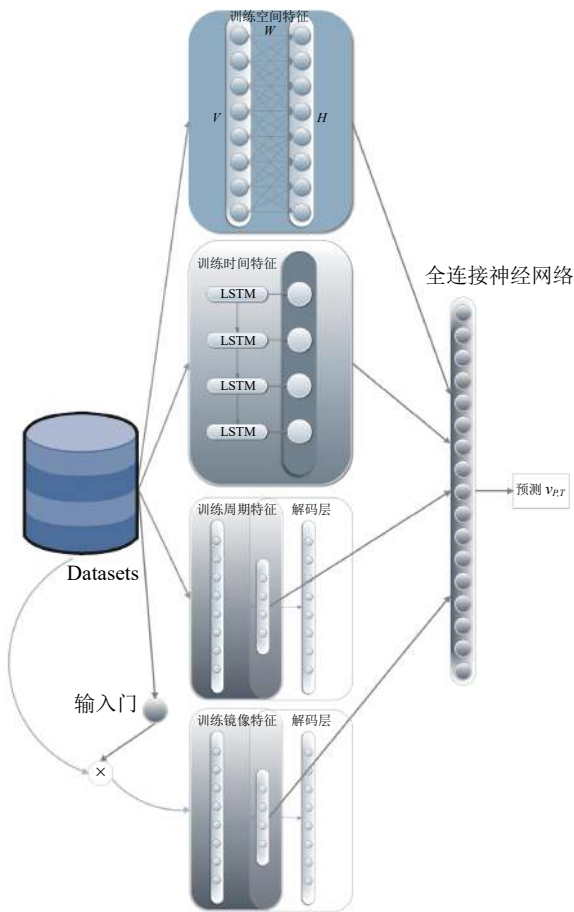


图 7 建立训练模型

Fig. 7 Building the training model

Force GTX 980Ti, CPU 为 Intel(R) Xeon(R) CPU E5-1607 v3. 实验使用的 Python 框架为 PyTorch. 实验选择的优化算法为 Adam 算法 (Adaptive moment estimation)^[24], 其超参数设置: $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$. 实验设置批数量 (Batch size) 为 16.

我们设计三组实验对比 5 类模型 (DSTCL, LSTM, S-ARIMA, DBN 和 ANN) 处理 PeMSD7 数据集的结果.

第 1 组实验研究模型的短时间间隔预测能力, 预测 10 min 之后的速度值. 我们设置输入时间步长为 18, 即模型使用 90 min 的历史数据去预测 10 min 之后的速度值. 我们将周期数据的时间间隔步长设置为 4, 即每次周期特征提取的时候使用当前时刻前后各 20 min 数据.

空间特征提取模块共有 3 个卷积层, 每个卷积层包含 128 个节点, 20 个滤波器. 初始输入一共 128 个节点 (高度值设置为 128, 宽度值设置为 1, 通道数设置为 1). 前两个卷积层滤波器的长度设置为 5, 输入填充的节点数设置为 2, 步长设置为 1; 最后一个卷积层滤波器长度设置为 4, 步长设置为 1, 对输入不进行填充. 使用 RELU 函数作为空间提取模块的激活函数. 由于模型的节点数较多, 为了避免发生过拟合现象, 在训练的时候, 使用 Dropout 方法^[29-30]. 设置 Dropout 的参数为 0.5, 即每次训练的时候激活 50% 的节点. 为了避免在训练过程中出现学习速度变慢, 训练陷入长时间的停滞状态或梯度爆炸等问题, 保证模型的非线性表达能力, 模型使用批归一化技术 (Batch normalization)^[31-32], 对每一批 (Batch) 数据进行归一化处理.

在训练前, 我们使用 K 均值聚类算法对空间特征提取模块进行预训练, 使每个学习到的中心作为卷积核, 这个方法被证明对小图像块是有效的^[33].

$$RELU(x) = \begin{cases} x, & x > \lambda \\ 0, & x \leq \lambda \end{cases} \quad (17)$$

时间特征提取模块包括 18 个 LSTM 块, 每一个 LSTM 块设置为 4 层, 每层的节点数分别为 128, 64, 32, 16. 周期和镜像特征提取模块中的周期特征提取部分编码层一共有 4 层, 每层的节点数分别为 60, 30, 15, 10; 镜像特征提取部分编码层一共有 6 层, 每层节点数分别为 179 712, 80 000, 4 000, 1 000, 500, 20. 设置 Dropout 参数为 0.5. 使用 RELU 函数作为这三个模块的激活函数. 周期和镜像特征提取模块的解码层只在模型训练的时候使用. 最后的全连接神经网络一共有 5 层, 节点数分别为 100, 40, 20, 10, 1.

第 2 组实验研究模型的中时间间隔预测能力,

预测 40 min 之后的速度值. 输入时间步长设置为 72, 即模型使用 360 min 的历史数据去预测 40 min 之后的速度值. 除了时间特征模块需要 72 个 LSTM 块之外, 其他模型超参数设置与第 1 组实验相同.

第 3 组实验研究模型的长时间间隔预测能力, 预测 100 min 之后的速度值. 输入步长设置为 180 min, 即模型用 900 min 的历史数据去预测 100 min 之后的速度值. 相对于第 1 组实验做如下超参数的改动, 时间特征模块设置为 180 个 LSTM 块, 最后全连接层设置为 6 层, 节点数分别为 300, 100, 40, 20, 10, 1.

我们设置一组实验处理 FMD 数据集. 对 FMD 数据集来说, 周期特征和镜像特征不那么明显, 在结构设计上不包含周期特征提取模块和镜像特征提取模块. 输入时间步长设置为 20, 即用 40 min 的历史数据去预测 10 min 之后的消防喷头开启状态. 空间特征提取模块有三个卷积层, 每层 180 个节点, 30 个滤波器, 前两层滤波器长度设置为 4, 输入填充的节点数设置为 2, 步长设置为 1; 最后一层滤波器的长度都设置为 3, 步长设置为 1, 对输入不进行填充. 训练时设置 Dropout 参数为 0.5, 并使用批归一化技术. 时间特征提取模块包含 20 个 LSTM 块, 每一个 LSTM 块设置为 4 层, 每层的节点数分别为 180, 80, 40, 10. 最后的全连接神经网络一共有 4 层, 节点数分别为 80, 30, 10, 1. 每层使用 RELU 作为激活函数.

针对 PeMSD7 数据集, 我们选择三种预测误差: MAE (Mean absolute error)、MAPE (Mean absolute percentage error) 和 RMSE (Root mean square error)^[15]. 这三个预测误差的计算式为

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \tilde{y}_t| \quad (18)$$

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \tilde{y}_t}{y_t} \right| \times 100\% \quad (19)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \tilde{y}_t)^2} \quad (20)$$

其中, y_t 表示真实值, $\tilde{y}_t$ 表示预测值.

针对 FMD 数据集, 我们使用 RMSE 作为预测误差.

为了验证 CNN、LSTM 以及堆叠自动编码器对整个模型预测能力的贡献, 设计了两组实验.

第 1 组: 研究 DSTCL 在分别去掉 3 个模块 (空间特征提取模块、时间特征提取模块以及周期和镜像提取模块) 后, 对长时间间隔 (100 min)、中时

间间隔 (40 min) 以及短时间间隔 (10 min) RMSE 结果的影响。

第 2 组: 分别用全连接的神经网络代替空间特征提取模块中的 CNN、时间特征提取模块中的 LSTM 以及周期和镜像特征提取模块中的堆叠自动编码器, 研究其对长时间间隔 (100 min)、中时间间隔 (40 min) 以及短时间间隔 (10 min) RMSE 结果的影响。

3.3 实验结果与分析

表 1~4 表示 DSTCL 模型与 LSTM、S-ARIMA、DBN、ANN 模型结果的对比。

表 1 表示针对 PeMSD7 数据集, 预测短时间间隔的 MAE、MAPE 和 RMSE 三种预测误差在测试集的值; 表 2 表示针对 PeMSD7 数据集, 预测中时间间隔的 MAE、MAPE 和 RMSE 三种预测误差在测试集的值; 表 3 表示针对 PeMSD7 数据集, 预测长时间间隔的 MAE、MAPE 和 RMSE 三种预测误差在测试集的值; 表 4 表示针对 FMD 数据集, 预测时间间隔为 10 min 的 RMSE 预测误差在测试集的值。

由表 1、表 2 和表 4 可以看出, 模型在预测中短时间间隔的数据时, DSTCL 的三种预测误差值都比其他模型低, 说明在预测中短时间间隔的数据时, DSTCL 算法效果最好。S-ARIMA 作为统计学方法, 由于对变化较快的数据处理能力较差, 在处理中短时间间隔的数据时预测误差最大, 效果最差。在处理 PeMSD7 数据集时, ANN 作为全连接的神经网络层, 预测误差较 LSTM 和 DBN 小, 说明其预测效果要优于 LSTM 和 DBN。

而从表 3 可以看出在预测长时间间隔的数据时, DSTCL 模型虽然效果不如 S-ARIMA, 但结果相差不大 (RMSE 差 0.23), 并且 DSTCL 模型的预测误差比 LSTM、DBN 和 ANN 模型的预测误差要小。表明 DSTCL 模型可以很好地抽取长时间间隔的数据特征。

我们选取高峰时间段传感器探测到的数据, 作为真实值对比 5 种模型的预测值 (预测时间间隔为 40 min), 结果如图 8 所示。

我们选择 15:00~19:00 的速度值是因为这个时间段是当地的人流高峰期, 速度变化比较大。由图 8 可以看出, 相对于其他模型, DSTCL 可以很好地获

表 1 预测 PeMSD7 时间间隔 10 min 各算法效果对比
Table 1 Prediction of the effect of each algorithm in the 10 min interval of PeMSD7

模型	MAE (10 min)	MAPE (10 min) (%)	RMSE (10 min)
DSTCL	2.61	6.0	4.32
LSTM	3.07	9.02	5.4
S-ARIMA	5.77	14.77	8.72
DBN	3.22	10.14	5.8
ANN	2.86	7.29	4.83

表 2 预测 PeMSD7 时间间隔 40 min 各算法效果对比
Table 2 Prediction of the effect of each algorithm in the 40 min interval of PeMSD7

模型	MAE (40 min)	MAPE (40 min) (%)	RMSE (40 min)
DSTCL	3.45	7.96	5.34
LSTM	3.81	9.46	5.92
S-ARIMA	4.8	14.47	8.6
DBN	4.11	10.66	6.5
ANN	3.63	9.98	5.77

表 3 预测 PeMSD7 时间间隔 100 min 各算法效果对比
Table 3 Prediction of the effect of each algorithm in the 100 min interval of PeMSD7

模型	MAE (100 min)	MAPE (100 min) (%)	RMSE (100 min)
DSTCL	4.15	9.94	7.05
LSTM	4.76	11.08	7.44
S-ARIMA	3.9	9.71	6.82
DBN	5.44	12.48	8.2
ANN	6.2	15.69	8.89

表 4 预测 FMD 时间间隔 10 min 各算法效果对比
Table 4 Prediction of the effect of each algorithm in the 10 min interval of FMD

模型	RMSE (10 min)
DSTCL	4.24
LSTM	4.62
S-ARIMA	8.44
DBN	5.21
ANN	5.37

取到速度随时间变化的趋势。

我们选择传感器探测到的 10 个位置数据, 作为真实值对比 5 种模型的预测值, 结果如图 9 所示。

由图 9 可以看出 DSTCL 在位置与速度的关系图像上, 曲线与真实值曲线最贴近。说明相对于其他模型, DSTCL 可以很好地获取速度与位置的变化关系。

由于 S-ARIMA 算法是一种统计学方法, 计算机制与 DSTCL、ANN、LSTM 和 DBN 这 4 种模型

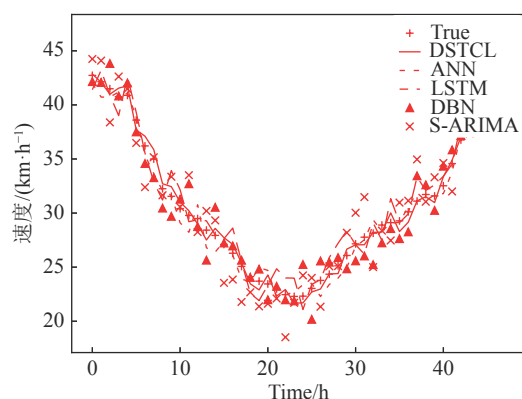


图 8 15:00~19:00 各模型预测与真实值的对比
Fig. 8 Comparison of model predictions from real values from 15:00 to 19:00

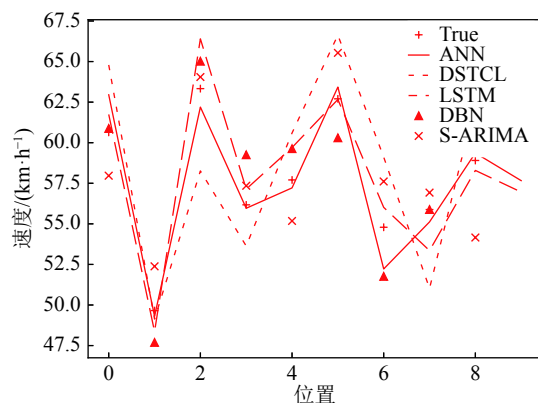


图 9 10 个不同位置各模型预测与真实值的对比
Fig. 9 Comparison of model predictions from real values of each model in 10 different locations

不同。我们针对 PEMS7 数据集, 只提取了 4 个模型 (DSTCL、ANN、LSTM 和 DBN) 在训练不同时间间隔的数据时训练集不同迭代次数的 RMSE, 结果如图 10~12 所示。

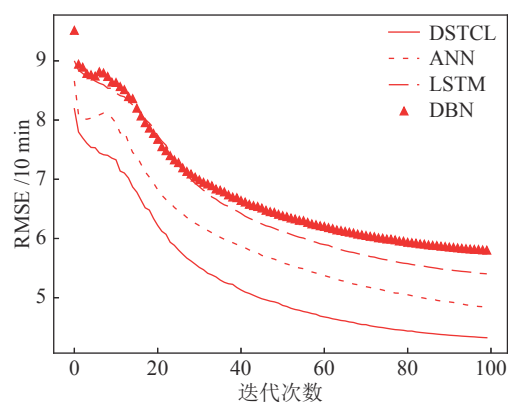


图 10 各个模型处理 10 min 间隔数据训练集 RMSE 变化
Fig. 10 Curve of RMSE of processing 10 min interval data in PeMSD7 training dataset

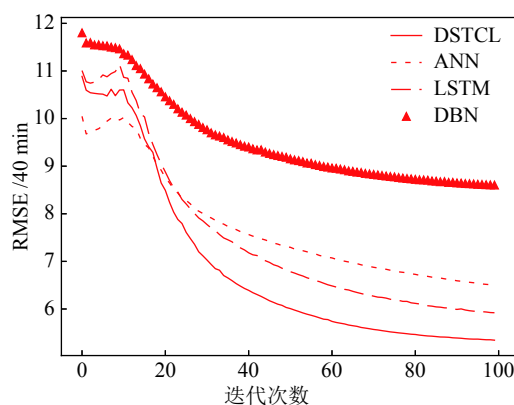


图 11 各个模型处理 40 min 间隔数据训练集 RMSE 变化
Fig. 11 Curves of RMSE of processing 40 min interval data in PeMSD7 training dataset

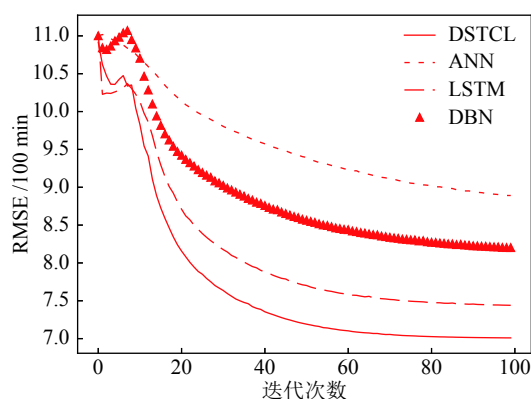


图 12 各个模型处理 100 min 间隔数据训练集 RMSE 变化
Fig. 12 Curves of RMSE of processing 100 min interval data in PeMSD7 training dataset

我们针对 FMD 数据集, 提取了这 4 个模型在训练数据时测试集不同迭代次数的 RMSE, 结果如图 13 所示.

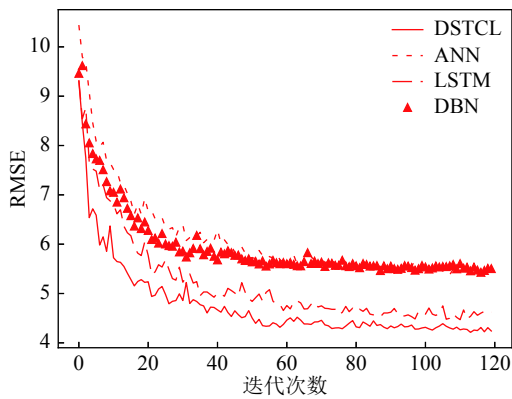


图 13 各个模型处理 FMD 测试集 RMSE 变化
Fig. 13 Curves of RMSE of processing 100-minute interval data in FMD testing dataset

由图 9~12 可以看出, DSTCL 模型在训练时梯度变化较大, 更新速度较其他模型快, 最终收敛值也较其他三个模型小, 说明模型在处理 PEMSD7 数据集时训练速度最快.

由图 13 中测试集的效果可以看出, 相对于其他三个模型, 使用 DSTCL 模型训练数据时, 可以使 RMSE 在测试集获得比较快的下降速度和收敛速度, 也可以说明模型在处理 FMD 数据集时速度最快.

表 5 表示针对 PeMSD7 数据集, DSTCL 在分别去掉三个模块 (空间特征提取模块、时间特征提取模块、周期和镜像特征提取模块) 后, 长时间间隔 (100 min)、中时间间隔 (40 min) 以及短时间间隔 (10 min) 的 RMSE 结果对比.

表 5 分别去掉三个模块 RMSE 结果对比
Table 5 Comparison of RMSE of removing three modules separately

	10 min	40 min	100 min
DSTCL1 (DSTCL去掉空间特征提取模块)	5.29	6.1	7.3
DSTCL2 (DSTCL去掉时间特征提取模块)	4.99	5.85	8.47
DSTCL3 (DSTCL去掉周期和镜像特征提取模块)	4.48	5.60	7.22

表 6 使用全连接神经网络替换三种结构的 RMSE 结果对比
Table 6 Comparison of RMSE of replacing three modules with fully ANN separately

	10 min	40 min	100 min
DSTCL4 (替换CNN)	5.16	5.98	7.21
DSTCL5 (替换LSTM)	4.85	5.52	8.1
DSTCL6 (替换堆叠自动编码器)	4.41	5.4	7.16

由表 5 可以看出, DSTCL1 在预测 10 min 和 40 min 的数据时, 相对于其他两种模型, 其 RMSE 最大; DSTCL2 在预测 100 min 的数据时, 相对于其他两种模型, 其 RMSE 最大. DSTCL3 整体相较于其他两种最优, 但不如 DSTCL. 说明空间特征提取模块对 DSTCL 预测中短时间间隔数据的能力贡献最大; 时间特征提取模块对 DSTCL 预测长时间间隔数据的能力贡献最大; 而在模型中加入周期和镜像特征提取模块相当于整体上进行优化.

表 6 表示针对 PeMSD7 数据集, 分别用全连接的神经网络代替空间特征提取模块中的 CNN、时间特征提取模块中的 LSTM 以及周期和镜像特征提取模块中的堆叠自动编码器后, 长时间间隔 (100 min)、中时间间隔 (40 min) 以及短时间间隔 (10 min) 的 RMSE 结果对比.

由表 6 可以看出, DSTCL4 在预测中短时间间隔的数据时, 相对于完全去掉空间特征提取模块的 DSTCL1 而言, RMSE 仅降低了 0.12 和 0.13, 对比 DSTCL5 和 DSTCL6, 其 RMSE 仍为最大, 说明 CNN 为空间特征提取模块中重要的结构, 对模型预测短时间间隔数据的能力贡献最大. DSTCL5 在预测长时间间隔的数据时, 对比完全去掉时间特征提取模块的 DSTCL2, RMSE 仅降低了 0.37, 相对于其他两种模型, 其 RMSE 仍为最大, 说明 LSTM 为时间特征提取模块中重要的结构, 对模型预测长时间间隔数据的能力贡献最大. DSTCL6 相对完全去掉周期和镜像特征提取模块的 DSTCL3, RMSE 降低了 (0.07, 0.16, 0.06), 整体相较于 DSTCL4 和 DSTCL5 最优, 但不如 DSTCL. 说明堆叠自动编码器为周期和镜像特征提取模块中重要的结构, 在模型中加入堆叠自动编码器相当于整体上进行优化.

4 结论

在本文中, 我们设计了一个时空数据模型——DSTCL 用于预测交通数据中的速度信息. DSTCL 模型是一个包含卷积神经网络和长短时间记忆网络的多元神经网络, 解决了其他时空数据模型只关注时间属性信息导致的对短时间间隔信息预测能力不足的问题, 通过对临近空间属性信息的特征和时间特征的提取, 得到对短时间间隔、中时间间隔和长时间间隔数据的有效预测. 通过实验对比了 DSTCL 与当前主流时空数据模型的训练效果. 实验结果显示 DSTCL 在预测中短时间间隔的信息的能力要优于其他时空数据模型. 而预测长时间间隔的数据时, DSTCL 的预测误差与 S-ARIMA 相差不大 (RMSE 差 0.23), 但低于其他三个模型 (ANN、LSTM 和 DBN), 说明 DSTCL 可以比较有效地预测长时间间隔的数据. 通过高峰时间段传感器探测得到的真实数据和不同时空数据模型的预测值对比, 显示出 DSTCL 相对于其他时空数据模型, 更好地捕捉到了短时间内敏感的速度变化信息, 说明 DSTCL 能更好地获取速度随时间变化的趋势. 比较临近位置的同一时刻的速度信息, 可以看到 DSTCL 相对于其他时空数据模型, 预测值与真实值的随空间信息改变的变化趋势相近, 说明 DSTCL 可以很好地获取速度信息随空间信息变化的趋势. 通过对比三种结构 (CNN、LSTM 以及堆叠自动编码器) 对模型预测性能的影响, 可以看出 CNN 对模型预测中短时间间隔数据的能力贡献较大; LSTM 对模型预测长时间间隔数据的能力贡献较大; 加入堆叠自动编码器相当于对模型整体上进行优化.

DSTCL 模型在处理数据时, 有时候出现训练前期预测误差波动的现象, 未来的主要工作是研究波动的原因, 并进一步优化模型的超参数, 使预测效果更优.

References

- 1 Zhao Fan, Jiang Tong-Hai, Zhou Xi, Ma Bo, Cheng Li. Visualization of multi-dimensional sparse spatial-temporal data. *Journal of University of Science and Technology of China*, 2017, **47**(7): 556–568
(赵凡, 蒋同海, 周喜, 马博, 程力. 面向多维稀疏时空数据的可视化研究. 中国科学技术大学学报, 2017, **47**(7): 556–568)
- 2 Cuturi M. Fast global alignment kernels. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). Bellevue, Washington, USA, 2011. 929–936
- 3 Goodchild M F. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 2007, **69**(4): 211–221
- 4 Ye M, Zhang Q, Wang L, Zhu J J, Yang R G, Gall J. A survey on human motion analysis from depth data. *Time-of-flight and Depth Imaging. Sensors, Algorithms, and Applications*. Springer, Berlin, Heidelberg, 2013: 149–187
- 5 Lagaros N, Papadrakakis M. Engineering and applied sciences optimization. *Computational Methods in Applied Sciences*, Springer, 2015, : 38
- 6 Vlahogianni E I. Computational intelligence and optimization for transportation big data: challenges and opportunities. *Computational Methods in Applied Sciences*, Springer, Cham, 2015, : 107–128
- 7 Ahmed M S, Cook A R. *Analysis of Freeway Traffic Timeseries Data by Using Box-jenkins Techniques*. 1979.
- 8 Williams B M, Hoel L A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. *Journal of Transportation Engineering*, 2003, **129**(6): 664–672
- 9 Lippi M, Bertini M, Frascioni P. Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 2013, **14**(2): 871–882
- 10 Kumar S V, Vanajakshi L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *European Transport Research Review*, 2015, **7**(3): 21
- 11 Dougherty M. A review of neural networks applied to transport. *Transportation Research Part C: Emerging Technologies*, 1995, **3**(4): 247–260
- 12 Hua J, Faghri A. Applications of artificial neural networks to intelligent vehicle-highway systems. *Transportation Research Record*, 1994, **1453**: 83
- 13 Smith B L, Demetsky M J. Short-term traffic flow prediction models — a comparison of neural network and nonparametric regression approaches. In: Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 1994. 2: 1706–1709
- 14 Chan K Y, Dillon T S, Singh J, Chang E. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg-Marquardt algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 2012, **13**(2): 644–654
- 15 Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006, **18**(7): 1527–1554
- 16 LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, **521**(7553): 436
- 17 Polson N G, Sokolov V O. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 2017, **79**: 1–17
- 18 Jia Y H, Wu J P, Du Y M. Traffic speed prediction using deep learning method. In: Proceedings of the 2016 IEEE International Conference on Intelligent Transportation Systems. IEEE, 2016. 1217–1222
- 19 Lv Y S, Duan Y J, Kang W W, Li Z X, Wang F Y. Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 2015, **16**(2): 865–873
- 20 Tan H C, Wu Y K, Shen B, Jin P J, Ran B. Short-term traffic prediction based on dynamic tensor completion. *IEEE Transactions on Intelligent Transportation Systems*, 2016, **17**(8): 2123–2133
- 21 Graves A. Generating sequences with recurrent neural networks. *ArXiv Preprint*, ArXiv: 1308.0850, 2013.
- 22 Ma X L, Tao Z M, Wang Y H, Yu H Y, Wang Y P. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 2015, **54**: 187–197
- 23 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 2012 Advances in Neural Information Processing Systems, 2012.

1097–1105

- 24 Szegedy C, Liu W, Jia Y Q, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1–9
- 25 Springenberg J T, Dosovitskiy A, Brox T, Riedmiller M. Striving for simplicity: the all convolutional net. ArXiv Preprint, ArXiv: 1412.6806, 2014.
- 26 Qiu Z F, Yao T, Mei T. Learning spatio-temporal representation with pseudo-3d residual networks. In: Proceedings of the 2017 IEEE International Conference on Computer Vision. 2017. 5533–5541
- 27 Gers F A, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. *Neural Computation*, 2000, **12**(10): 2451–2471
- 28 Chen C, Petty K F, Skabardonis A, Varaiya P, Jia Z F. Free-way performance measurement system: mining loop detector data. *Transportation Research Record: Journal of the Transportation Research Board*, 2001, **1748**(1): 96–102
- 29 Xiao T, Li H S, Ouyang W L, Wang X G. Learning deep feature representations with domain guided dropout for person re-identification. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016. 1249–1258
- 30 Ozan I, Ethem A. Dropout Regularization in Hierarchical Mixture of Experts. ArXiv Preprint, ArXiv: 1812.10158, 2018.
- 31 Cai Y Q, Li Q X, Shen Z W. On the convergence and robustness of batch normalization. ArXiv Preprint, ArXiv: 1810.00122, 2018.
- 32 Jung W, Jung D, Kim B, Lee S, Rhee W, Ahn J H. Restructuring batch normalization to accelerate CNN training. ArXiv Preprint, ArXiv: 1807.01702, 2018.
- 33 Coates A, Lee H, Ng A. An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, 2011. 215–223



秦 超 西北工业大学电子信息学院博士研究生. 主要研究方向为深度学习.

E-mail: woshiqchi@gmail.com

(**QIN Chao** Ph.D. candidate in the Department of System Engineering, Northwestern Polytechnical University. His main research interest is deep learning.)



高晓光 博士, 西北工业大学电子信息学院教授. 主要研究方向为深度学习, 贝叶斯网络.

E-mail: cxg2012@nwpu.edu.cn

(**GAO Xiao-Guang** Ph.D., professor in the Department of System Engineering, Northwestern Polytechnical University. Her research interest covers deep learning and Bayesian network theory.)



万开方 博士, 西北工业大学电子信息学院讲师. 主要研究方向为深度学习, 强化学习. 本文通信作者.

E-mail: wankaifang@nwpu.edu.cn

(**WAN Kai-Fang** Ph.D., lecturer in the Department of System Engineering, Northwestern Polytechnical University. His research interest covers deep learning and reinforcement learning Corresponding author of this paper.)