《基于深度学习的交通流量预测与优化》
文献报告

2020 年 11 月 2 日

目录

1 A Comprehensive Survey on Traffic Prediction

论文引用：Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., & Yin, B. (2020). A Comprehensive Survey on Traffic Prediction. arXiv preprint arXiv:2004.08555.

1.1 摘要

1. 总结了现有的流量预测方法，并给出它们的分类。
2. 列出了交通预测的常见任务和这些任务的最新技术。
3. 收集和组织现有文献中广泛使用的公共数据集
4. 讨论了未来可能的发展方向。

1.2 导论

1. 智能交通系统(ITS)是智慧城市不可缺少的一部分，而交通预测是其发展的基石。例如：交通量预测可以帮助城市缓解拥堵；网约车需求预测可以促使汽车共享公司将车辆预先分配到高需求地区。
2. 交通预测问题的挑战：
   - 复杂空间依赖性：不同位置对预测位置的影响不同，相同位置对预测位置的影响也随着时间的变化而变化。不同位置之间的空间相关性是高度动态的。
   - 动态时间依赖性：同一位置不同时间的观测值呈现非线性变化，远时步长的交通状态与预测时间步长的相关性比近时要大。
   - 外部因素：交通时空序列数据还受到天气、事件或道路属性等外部因素的影响。
3. 交通预测的主要预测任务：
   - 交通流：在一定时间内通过道路上某一点的车辆数量。
   - 速度：车辆单位时间内行驶的距离
   - 需求：使用历史请求数据来预测未来时间戳中某个区域的请求数量
   - 占用率：车辆占用道路空间的程度。
   - 出行时间：从一点到另一点的耗费时间

1.3 预测方法

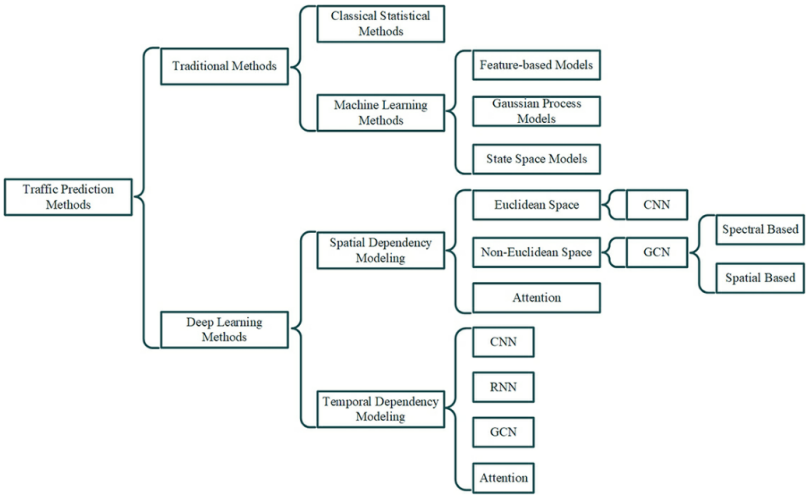


Fig. 2. Categories of traffic prediction methods.

1.3.1 传统方法

1. 经典统计模型

   以ARIMA一类经典统计模型为代表，问题在于不适合处理复杂、动态的时间序列数据。此外，由于通常只考虑时间信息，交通数据的空间依赖性被忽略或很少考虑。

2. 机器学习模型机器学习方法可以建模更复杂数据,大致可分为三类:

   (a) 基于特征的模型：训练基于人工工程交通特征的回归模型来解决交通预测问题这些方法易于实现，并能在一些实际情况下提供预测。尽管存在这种可行性，但基于特性的模型有一个重要的局限性:模型的性能严重依赖于人工设计的特征。

   (b) 高斯过程模型：高斯过程通过不同的核函数来模拟交通数据的内在特征，这些核函数需要同时包含空间和时间的相关性。虽然这类方法在流量预测([25]-[27])中被证明是有效可行的，但是它们具有较高的计算负荷和存储压力，在有大量训练样本的情况下不合适。

   (c) 状态空间模型：该模型的优点是能够自然地对系统的不确定性进行建模，并能更好地捕捉到时空数据的潜在结构。然而，这些模型([28]-[38])的整体非线性是有限的，大多数情况下，它们对复杂动态交通数据的建模不是最优的。

1.3.2 深度学习模型

1. 建模空间依赖性

   - CNN：将不同时刻的交通网络结构转换成图像，并将这些图像划分为标准网格，每个网格代表一个区域。这样CNNs就可以用来学习不同区域之间的空间特征。
   - GCN：传统的CNN仅限于欧氏数据的建模，因此使用GCN来建模非欧氏空间结构数据，更符合交通道路网络的结构。
   - Attention

2. 建模时间依赖性

   - CNN：一维卷积。因为交通预测问题和长期相关性的关联大，短期相关性的关联性小，卷积也能很好的学习到特征，速度很快。

- RNN：传统的RNN中LSTM,GRU面对长时期遗忘的问题很严重,产生梯度消失等问题。
- GCN
- Attention

1.4 公共数据集

用于预测任务的公共数据信息有两类：1) 预测中常用的公共时空序列数据，2) 用于提高预测精度的外部数据。

外部数据包含的信息一般有：

- 天气
- 驾驶员ID：由于驾驶员个人情况的不同，预测会产生一定的影响，因此需要对驾驶员进行标签，该信息主要用于个人预测
- 活动：包括各种节日、交通管制、交通事故、体育赛事、音乐会等活动。
- 时间信息：工作日和周末；每天的不同时段

1.5 实验

实验结果总结：

1. 图神经是未来。图是表达交通拓扑结构的自然表达，应运而生的图神经网络在各种交通预测问题上有着良好的发挥。
2. 注意力机制很有用。注意力机制能注意到重要的时间片，如相同的周等，对于以往问题有显著作用。

1.6 未来研究方向

- 少样本问题：很多地方传感器不够多，样本较少，如何少样本中提取足够信息。
- 知识图融合问题：和其他信息相关联，更好的获得地区知识，以便智慧交通的建成。
- 长期预测：大量的研究集中于短时预测，长时间预测受到外部因素影响更为严重，更好的进行长期预测也是课题之一。

2 DNN-based prediction model for spatio-temporal data

论文引用：APAZhang, J. , Zheng, Y. , Qi, D. , Li, R. , & Yi, X. . (2016). DNN-based prediction model for spatio-temporal data. the 24th ACM SIGSPATIAL International Conference. ACM.

2.1 引言

目前很多领域的数据都属于时空数据（Spatial-Temporal Data），即ST数据。这类数据具有具有独特的空间属性(即地理层次和距离)和时间属性(即紧密性closeness、周期period和趋势trend)。

本文提出了基于DNN的DeepST模型，包含两个关键组件：时空组件和全局组件，来较好地捕捉交通流量的时空属性，从而实现短期的交通流量预测。主要贡献：

1. 为时空数据设计了一种新的深度学习架构，并提出使用1)时间属性来选择合适的时间戳，用于建模时间的紧密性、周期和趋势;2)卷积捕获空间远近依赖关系;3)早期和晚期融合，融合相似ST数据以及全球信息。

2. 应用DeepST来预测整个城市的人流，并开发了一个实时的人流预测系统(称为UrbanFlow)，该系统可以有效地监控细粒度的人流，并提供城市未来的人流。

2.2 模型

模型主要由两个组件构成：时空组件（spatio-temporal）和全局组件（global）。时空组件用于对捕捉数据的时间依赖性和空间依赖性，空间组件用于将前一组件得到的时空特征与全局特征（星期几，是否为周末等）结合，最终输出对接下来一个时间步的流量预测。

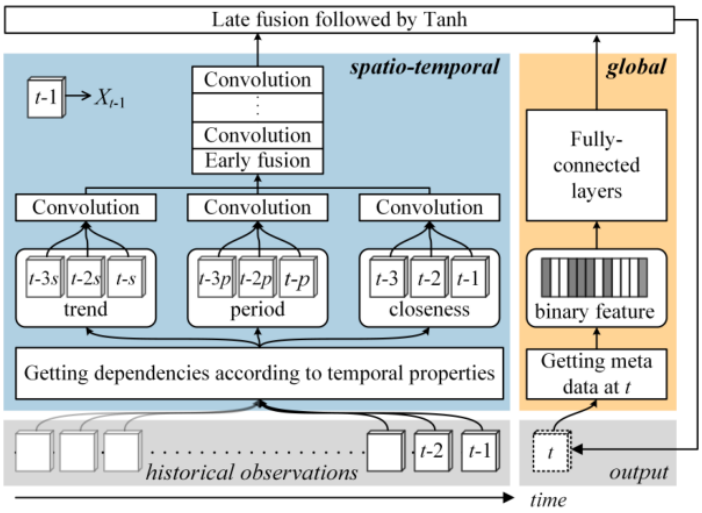


Figure 3: DeepST Architecture.

2.2.1 时空组件

时间依赖性建模 如模型图中时空组件的下半部分所示，根据数据的时间属性，分别建立了三个模块：邻近性（closeness）、周期性（period）和趋势性（trend），每个模块由特定的时间戳构成。邻近性模块表示为

$$[X_{t-l_c},\cdots,X_{t-1}]$$

其中l_c为选取的时间戳个数。同理，周期性和趋势性表示为

$$[X_{t-l_p p},X_{t-(l_p-1)p},\cdots,X_{t-p}],$$

$$[X_{t-l_s s},X_{t-(l_s-1)s},\cdots,X_{t-s}],$$

其中l_p和l_s为两个模块中选取的时间戳个数，p为要观察的单个周期长度，s为要观察的趋势时间跨度。

空间依赖性建模 根据卷积神经网络（convolutional neural network）模型的原理，卷积层可以很好地捕捉数据在空间上的局部特征，且多个卷积层堆叠可以捕捉到更远的空间依赖性。因此，将前述三个模块的向量输入一个卷积模块：

$$H_c^{(1)}=f(\sum_{j=1}^{l_c}W_{cj}^{(1)}*X_{t-j}+b_c^{(1)})$$

$$H_p^{(1)} = f(\sum_{j=1}^{l_p} W_{pj}^{(1)} * X_{t-j\cdot p} + b_p^{(1)})$$

$$H_s^{(1)} = f(\sum_{j=1}^{l_s} W_{sj}^{(1)} * X_{t-j\cdot s} + b_s^{(1)})$$

输出即为包含了空间依赖性和特定属性的时间依赖性的特征向量。

2.2.2 全局组件

在这一组件中，为了将全局性的信息和数据特征结合，首先输入在t时刻的元数据向量，包括两个信息：星期几，是工作日还是周末。将元数据向量（利用one-hot编码方式）转换为二进制向量G_t。与时空组件输出的特征向量H_{st}融合，采用tanh激活函数得到最终的预测输出：

$$\hat{X}_t = \tanh(W_{st} \cdot H_{st} + W_G \cdot G_t)$$

模型采用平方误差作为损失函数：$\|\hat{X}_t - X_t\|_2^2$

2.3 实验

2.3.1 实验设置

1. 数据集：

Table 2: Datasets

Dataset	TaxiBJ15	TaxiGY16	LoopBJ15	BikeNYC14
Data type	Taxi GPS	Taxi GPS	Loop detector	Bike rent
Location	Beijing	Guiyang	Guiyang	New York
Start time	3/1/2015	3/18/2016	10/1/2015	4/1/2014
End time	6/30/2015	5/4/2016	4/1/2016	9/30/2014
Test set	Last week	Last week	Last week	Last 10 days
Training set	others that are NOT in the test set			
Time interval	0.5 hour	0.5 hour	0.5 hour	1 hour
Gird map size	(32, 32)	(32, 32)	(32, 32)	(16, 8)
Trajectory data				
#taxis/bikes	30,000+	6,000+	\	6,800+
#time intervals	5,760	2,304	8,832	4,392

2. 对比模型：

Table 1: Description on Models

Models	Description
Baselines	
ARIMA	autoregressive integrated moving average
SARIMA	seasonal ARIMA
VAR	vector autoregressive model
CNN	convolutional neural networks, the input is X_{t-1}
DeepST	
C	temporal closeness sequence
CP	C + periodic sequence
CPT	CP + seasonal trend sequence
CPTM	CPT + meta data

Note: Convolutions in DeepST and CNN have the similar setting. There are a total of 4 convolutional layers, each of which has 64 feature maps with 3×3 kernels.

3. 评价指标：RMSE

2.3.2 实验结果

1. 单步预测实验结果：

Table 3: RMSE. The smaller, the better.

Models	TaxiBJ15	TaxiGY16	LoopGY16	BikeNYC14
ARIMA	25.58	23.31	137.83	10.56
SARIMA	29.11	26.51	135.25	10.07
VAR	25.59	22.70	146.16	9.92
CNN	26.08	22.92	183.51	8.55
DeepST (ours)				
C	23.63	22.09	132.26	8.39
CP	23.84	21.51	**129.13**	7.64
CPT	23.33	20.98	130.53	7.56
CPTM	**22.59**	**19.97**	130.25	**7.43**

包含三个时间模块和元数据的DeepST模型表现最好。

2. 多步预测实验结果：

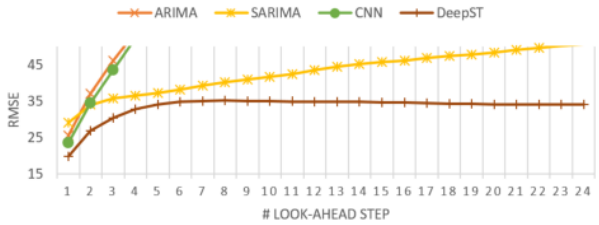


Figure 5: Multi-step Ahead Prediction

DeepST模型的误差最小，随预测步数增加表现保持稳定。

3. 数据量对结果的影响：数据越多，模型表现越好。

3 Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework

论文引用：Wu, Y., & Tan, H. (2016). Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv preprint arXiv:1612.01022.

3.1 引言

1. 问题背景：准确可靠的短期交通流预测是主动动态交通控制、智能路线引导和智能定位服务等多种智能交通系统的关键前提

2. 介绍了目前主要使用的两类模型：统计模型（ARIMA, 马尔科夫链，贝叶斯网络等）和神经网络。统计方法的主要缺点：无法很好地处理非线性数据，且存在维数诅咒。神经网络的主要优点：非参数方法，输入变量更灵活；非线性激活函数，便于处理非线性问题。

3. 介绍目前对这类问题的相关研究使用的模型：

   (a) 深度多层全连接网络结合无监督算法预训练策略（RBM，SAE等），缺点：在内存和计算方面代价昂贵；没有对特征进行假设，很难获取具有代表性的特征。

   (b) 卷积神经网络CNN，对拓扑局部性（即空间依赖性）建模。

(c) LSTM等递归神经网络，对长时记忆（即时间依赖性）建模。

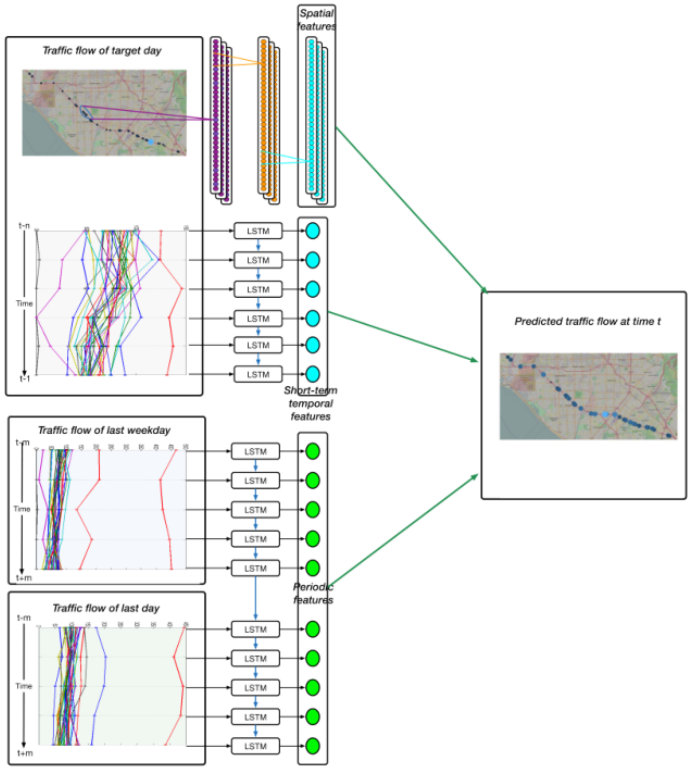
4. 文章贡献：

(a) 提出了一种新的基于CNN和LSTM (CLTFP)组合的短期交通流预测方法。

(b) 根据递增的可预测性分析CLTFP捕获的特性，该分析图形化地演示了黑盒类型的CLTFP如何理解未来和过去交通流之间的因果关系。

3.2 模型

CLTFP模型由一个1-D CNN，两个LSTM RNNs和一个全连接层组成。一维CNN模块输出数据的空间特征，LSTM模块输出数据的时间特征，最终将所有模块输出的特征向量依次连接，输入一个全连接层，进行下一个时间步的流量预测。



3.2.1 空间特征建模

对于空间特征，采用1-D的CNN来捕捉。取p个地点为观察对象，第t时刻往前n个历史时间步的数据作为输入，来预测接下来h个时间步的交通流量。整合以上历史数据可以得到矩阵

$$\mathbf{S} = \begin{bmatrix} S_1 \\ S_2 \\ \cdot \\ \cdot \\ \cdot \\ S_p \end{bmatrix} = \begin{bmatrix} s_1(t-n) & s_1(t-n+1) & \cdots & s_1(t-1) \\ s_2(t-n) & s_2(t-n+1) & \cdots & s_2(t-1) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ s_p(t-n) & s_p(t-n+1) & \cdots & s_p(t-1) \end{bmatrix}.$$

矩阵的列向量即为某个时刻所有地点的交通流量$T_q = [s_1(t-n+q), s_2(t-n+q), \cdots, s_p(t-n+q)]^T (0 < q \leq n-1)$。基于这个矩阵，该CNN模块的输入为n个序列$T_q$（每个时间步作为一个通道），

每层卷积层包含k_l个一维卷积核，每个卷积核对序列T_q进行卷积操作，最终得到包含了数据空间特征的输出。

3.2.2 短期时间特征建模

采用LSTM模块来捕捉交通流数据中的短期空间特征。
输入为$T=(T_0,T_1,\cdots,T_{n-1})$，其中$T_q$含义同前。通过LSTM模块后，输出为时间特征$H=(H_0,H_1,\cdots,H_{n-1})$。

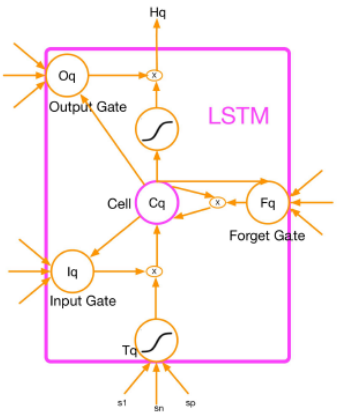


Figure 2: LSTM structure

3.2.3 周期性特征建模

对于时序中存在的周期性特征，同样采用LSTM模块。分别考虑日周期性和周周期性，利用第t_d天往前和往后最近的n_d个天的相同时刻的交通流量，及第t_w周往前和往后最近的n_w个周的相同时刻的交通流量，得到矩阵

$$\mathbf{S^d}=\begin{bmatrix} s_1(t^d-n^d) & s_1(t^d-n^d+1) & \cdots & s_1(t^d+n^d) \\ s_2(t^d-n^d) & s_2(t^d-n^d+1) & \cdots & s_2(t^d+n^d) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ s_p(t^d-n^d) & s_p(t^d-n^d+1) & \cdots & s_p(t^d+n^d), \end{bmatrix}.$$

$$\mathbf{S^w}=\begin{bmatrix} s_1(t^w-n^w) & s_1(t^w-n^w+1) & \cdots & s_1(t^w+n^w) \\ s_2(t^w-n^w) & s_2(t^w-n^w+1) & \cdots & s_2(t^w+n^w) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ s_p(t^w-n^w) & s_p(t^w-n^w+1) & \cdots & s_p(t^w+n^w), \end{bmatrix}$$

即

$$T^d=(T_0^d,T_1^d,\cdots,T_{2n^d}^d),T_q^d=[s_1(t^d-n^d+q),s_2(t^d-n^d+q),\cdots,s_p(t^d-n^d+q)]$$

$$T^w=(T_0^w,T_1^w,\cdots,T_{2n^w}^w),T_q^w=[s_1(t^w-n^w+q),s_2(t^w-n^w+q),\cdots,s_p(t^w-n^w+q)]$$

输出得到日周期性特征$H^d=(H_0^d,H_1^d,\cdots,H_{2n^d}^d)$和周周期性特征$H^w=(H_0^w,H_1^w,\cdots,H_{2n^w}^w)$

3.2.4 特征组合

把上述模块得到的空间特征、短期时间特征、日周期性特征和周周期性特征依次连接成一个特征向量作为最后一个全连接层的输入，输出接下来h个时间步的交通流量预测值。

3.3 实验

3.3.1 实验设置

1. 数据集

项目	信息
地点	North-bound I-405 trip的33个地点
时间	2014/04/01 - 2015/06/30
取样时间间隔	5分钟
训练集	前110000个数据
测试集	训练集以外的数据

2. 对比模型：LSTM，SAE，一个浅层NN，GBRT
3. 超参数设置：
   - 1-D CNN：三层卷积层，每层30个卷积核，前两层卷积核长度为3，第三层的卷积核长度为2，激活函数采用SReLu。
   - 采用L1规范化，规范化参数为0.002
4. 评价指标：MAE，MAPE，ACE

3.4 实验结果

Table 1: The quantitative results of different methods

features	MAE	MAPE	ACE
CLTFP	19.37	7.36%	0.9263
LSTM	21.53	8.55%	0.9137
SAE	20.36	8.07%	0.9198
NN	20.61	8.31%	0.9174
GBRT	22.52	8.52%	0.9109

CLTFP在预测精度和空间分布方面优于实验的其他模型。

4 Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction

论文引用：Zhang, J., Zheng, Y., & Qi, D. (2016). Deep spatio-temporal residual networks for citywide crowd flows prediction. arXiv preprint arXiv:1610.00081.

4.1 引言

1. 预测交通流量的重要性
2. 预测目标：
   - 流入流（inflow）：给定时间区间内进入一个区域的交通总流量（包含行人、汽车和公共汽车等）
   - 流出流（outflow）：给定时间区间内流出一个区域的交通总流量
3. 介绍了目前两种主流深度神经网络架构，考虑了部分时间或空间属性：1）卷积神经网络（CNN）：空间结构 2）递归神经网络（RNN）：时间依赖关系
4. 深度学习在时空预测问题上遇到的挑战（需要考虑的因素）
   - 空间依赖性

     1) 近处（nearby）：一个区域的流出流影响邻近区域的流入流；一个区域的流入流也会影响该区域的流出流

     2) 远处（distant）：住在其他地方的上班族需要进入该区域工作，因此一个区域的流入流还受到距离较远区域流出流的影响
   - 时间依赖性

     1) 邻近性 closeness：同一天邻近时间点的影响

     2) 周期性 period：每天同一时间段时空流间高度相关

     3) 趋势性 trend：rush hour的出现与季节温度有关（冬天温度下降，白天变短，人们起的越来越迟）
   - 外部因素影响：天气因素、突发事件、特殊事件等
5. 文章的主要贡献
   - 提出了ST-ResNet模型，基于卷积残差网络（Convolutional-based Residual Network），使得网络能够反映近处/远处两种空间依赖性，同时模型的预测精度不受神经网络的深层结构的影响。
   - 总结了交通流量的时间特性：邻近（closeness）、时段（period）、趋势（trend），并在ST-ResNet中分别使用不同的残差网络描述了这三种特性
   - ST-ResNet动态聚合前述三个残差网络的输出，并为不同区域赋予了不同的权重，并且纳入了外部因素的影响
   - 将构建的网络在北京出租车轨迹及气象数据集，以及纽约自行车轨迹数据集上进行了验证。结果证明文章的方法优于其他的九种方法。
   - 基于该网络搭建了实时监控和预测区域交通流的系统。并且，该系统基于云和GPU，提供了高效灵活的计算环境支持。
6. 相比前人研究的创新点：
   - 实际应用：部署了一个基于云平台的系统（3），能够利用实时数据来预测贵阳市任一区域交通流量
   - 预测跨度更大：能够进行多步预测，能对较长期流量进行预测（4.4）
   - 更全面的试验：证实系统鲁棒性和有效性
   - 探索了相关领域研究进展，阐述与别人工作的区别与联系（6）

4.2 预备知识

4.2.1 交通流量预测问题

假设$\mathbb{P}$为发生在t^{th}时间区间的轨迹（trajectories）集合，对区域(i,j)而言，流入流出流被定义为：

$$x_t^{in,i,j} = \sum_{Tr\in\mathbb{P}} |\{k>1 | g_{k-1} \notin (i,j) \wedge g_k \in (i,j)\}|$$
$$x_t^{out,i,j} = \sum_{Tr\in\mathbb{P}} |\{k>1 | g_k \in (i,j) \wedge g_{k+1} \notin (i,j)\}|$$

其中，$Tr: g_1 \to g_2 \to \cdots \to g_{|Tr|}$是$\mathbb{P}$中的一条轨迹，$g_k \in (i,j)$代表该坐标在区域$(i,j)$中。$|\Delta|$代表集合的势（集合包含元素数）。因此，在$t^{th}$时间区间，$I \times J$区域的流入流和流出流可以用张量$X_t \in \mathbb{R}^{2\times I\times J}$表示，其中$(X_t)_{0,i,j} = x_t^{in,i,j}$，$(X_t)_{1,i,j} = x_t^{out,i,j}$。

因此，交通流量预测问题就可以定义为：给定历史观测数据$\{X_t | t = 0,\cdots,n-1\}$，预测$X_n$。

4.2.2 深度残差学习

深度残差学习下，CNN的深度可以达到100甚至多于1000层。这种学习方法在多种复杂识别任务上都展现了很高水准的性能。一个残差块可以表示为：

$$X^{(l+1)} = X^l + \mathscr{F}(X^{(l)})$$

$X^{(l)}$和$X^{(l+1)}$分别为l^{th}残差块的输入和输出，$\mathscr{F}$是需要学习的残差函数（比如：通过两层3×3卷积层来实现残差映射）。残差学习，核心在于通过学习残差函数间接得到输出来避免网络退化导致的效果下降（同时使用了浅层和深层的输出）。

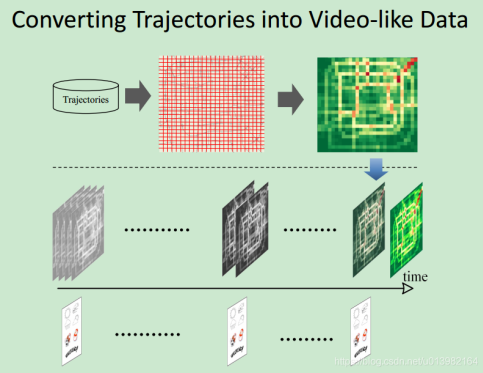
4.3 模型

LSTM这类RNN网络能够学习较长时间内的时间依赖性，然而为了描述时间的period和trend特征，需要输入很长的时空数据序列，使得训练过程变得复杂。基于时空领域知识，可以通过提取关键帧的方式来进行预测。因此，文章使用了closeness、period和trend来选取关键帧。

4.3.1 数据转化

- 轨迹数据：每一时段轨迹数据→ inflow和outflow→ 2-channel tensor → 按时间堆积得到类似视频流数据
- 气象/外部因素数据：提取特征 → 按时间堆积得到时序数据

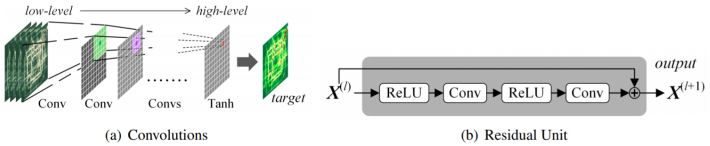
4.3.2 模型架构



由closeness、period、trend、external4个模块组成。前三者各使用一个DRN来训练，将结果乘以参数矩阵得到混合后的X_{Res}。再与输入external提取的特征进行训练的全连接网络输出的X_{Ext}相加，通过tanh（收敛快于S型神经元）得到最终的模型输出。通过缩小该输出和实际流量的差异（代价）来训练整个网络。

4.3.3 closeness, period, trend

结构一致，由卷积单元和残差单元组成。



(a) Convolutions (b) Residual Unit

Figure 6: Convolution and residual unit

卷积单元 CNN擅长处理层次化的空间结构信息，但传统深层CNN会出现分辨率损失的问题（降采样，池化层的使用使得像素产生损失，subsampling=pooling），导致空间依赖关系的丢失。因此，文章的卷积单元中没有使用池化层，而只用到了卷积层，使得网络能够同时保留nearby dependencies和distant dependencies。对于邻近区域和较远区域之间的空间依赖性，单层卷积的卷积核只能覆盖小区域（如3*3），卷积核反映出小区域内部的空间依赖性（也就是所说的nearby dependencies）。随着卷积层加深，新的卷积核在之前的特征图上进行特征映射，即此时的卷积核覆盖了多个小区域，反映了各个小区域之间的空间依赖性，这样不断叠加，就能渐渐描绘出较远区域之间的空间依赖性（distant dependencies）。

以closeness为例，选取recent time部分的一些关键帧（关键时间区间）的交通流三维张量组成序列：

$$[X_{t-l_c}, X_{t-(l_c-1)}, \ldots, X_{t-1}]$$

在时间维度上将这些2channel的张量连接起来，得到一个新的张量$X_c^{(0)} \in \mathbb{R}^{2l_c \times I \times J}$，作为Conv1的输入，进行如下卷积，类型为等卷积：

$$\mathrm{X}_c^{(1)} = f(W_c^{(1)} * \mathrm{X}_c^{(0)} + b_c^{(1)})$$

后续Conv2同上，均使用ReLU神经元。

残差单元 主要是为了解决网络层数增多导致的训练退化问题。

在Conv1和Conv2之间，共使用了L个残差单元，表达式如下：

$$\mathrm{X}_c^{(l+1)} = \mathrm{X}_c^{(l)} + \mathscr{F}(\mathrm{X}_c^{(l)}; \theta_c^{(l)}), l = 1, \cdots, L$$

$\mathscr{F}$是由残差单元中的两层卷积+两个ReLU函数表示的映射，ReLU前都使用了BN技术（这个好像是在残差网络提出的论文随后一篇优化论文里提出的，对比了多种结构，实验证明了BN加在ReLU前更好）。

另外两个特征period, trend类似：

$$[X_{t-l_p \cdot p}, X_{t-(l_p-1)\cdot p}, \ldots, X_{t-p}]$$

$$[X_{t-l_q \cdot q}, X_{t-(l_q-1)\cdot q}, \ldots, X_{t-q}]$$

l_c, l_p, l_q代表序列长度，p, q代表跨度（实际取了1天和1周，closeness取1，即一个区间间隔），最后三个网络的Conv2分别输出$X_c^{(L+2)}, X_p^{(L+2)}, X_q^{(L+2)}$。

将时间上的三个特性以参数矩阵方式融合：

$$\mathrm{X}_{Res} = \mathrm{W}_c \odot \mathrm{X}_c^{(L+2)} + \mathrm{W}_p \odot \mathrm{X}_p^{(L+2)} + \mathrm{W}_q \odot \mathrm{X}_q^{(L+2)}$$

W_c, W_p, W_q反映了三个特性影响程度的参数。

4.3.4 external component

主要提取的特征：

- holiday：是否为节假日
- weather：是否为雨天、温度、风速等
- weekday/weekend：工作日/休息日
- DayOfWeek：星期几

用E_t表示这些外部因素在时间间隔t的特征向量。相对于其他外部因素，将要预测时刻t的天气数据是无法预测的，文章使用了前一刻$t-1$的天气数据来近似替代。网络包括两层全连接层，前一层用作embedding，后一层用作低维到高维的映射（映射为与其他三个时间特征输出的X_t相同维度的数据），最终的输出用X_{Ext}表示。

与前面三个时间特征模块结合：

$$\hat{\mathrm{X}}_t = \tanh(\mathrm{X}_{Res} + \mathrm{X}_{Ext})$$

代价函数采用MSE：

$$\mathscr{L}(\theta) = ||\mathrm{X}_t - \hat{\mathrm{X}}_t||_2^2$$

其中θ代表网络中的所有参数。

4.3.5 训练步骤

Algorithm 1: Training of ST-ResNet

Input: Historical observations: $\{\mathbf{X}_0, \cdots, \mathbf{X}_{n-1}\}$;
external features: $\{E_0, \cdots, E_{n-1}\}$;
lengths of *closeness*, *period*, *trend* sequences: l_c, l_p, l_q;
peroid: p; trend span: q.
Output: ST-ResNet model $\mathcal{M}$.
// construct training instances
$\mathcal{D} \longleftarrow \emptyset$
for *all available time interval* $t (1 \le t \le n-1)$ **do**
  $\mathcal{S}_c = [\mathbf{X}_{t-l_c}, \mathbf{X}_{t-(l_c-1)}, \cdots, \mathbf{X}_{t-1}]$
  $\mathcal{S}_p = [\mathbf{X}_{t-l_p \cdot p}, \mathbf{X}_{t-(l_p-1)\cdot p}, \cdots, \mathbf{X}_{t-p}]$
  $\mathcal{S}_q = [\mathbf{X}_{t-l_q \cdot q}, \mathbf{X}_{t-(l_q-1)\cdot q}, \cdots, \mathbf{X}_{t-q}]$
  // $\mathbf{X}_t$ is the target at time t
  put an training instance $(\{\mathcal{S}_c, \mathcal{S}_p, \mathcal{S}_q, E_t\}, \mathbf{X}_t)$ into $\mathcal{D}$
// train the model
initialize the parameters θ
repeat
  randomly select a batch of instances $\mathcal{D}_b$ from $\mathcal{D}$
  find θ by minimizing the objective (6) with $\mathcal{D}_b$
until *stopping criteria is met*
output the learned ST-ResNet model $\mathcal{M}$

4.4 实验

4.4.1 实验设置

1. 数据集：包含轨迹数据和气象数据

Table 2: Datasets (holidays include adjacent weekends).

Dataset	TaxiBJ	BikeNYC
Data type	Taxi GPS	Bike rent
Location	Beijing	New York
Time Span	7/1/2013 - 10/30/2013 3/1/2014 - 6/30/2014 3/1/2015 - 6/30/2015 11/1/2015 - 4/10/2016	4/1/2014 - 9/30/2014
Time interval	30 minutes	1 hour
Gird map size	(32, 32)	(16, 8)
Trajectory data		
Average sampling rate (s)	~ 60	\
# taxis/bikes	34,000+	6,800+
# available time interval	22,459	4,392
External factors (holidays and meteorology)		
# holidays	41	20
Weather conditions	16 types (*e.g.*, Sunny, Rainy)	\
Temperature / °C	[−24.6, 41.0]	\
Wind speed / mph	[0, 48.6]	\

90%用作训练，10%用作验证。

2. 对比模型：HA, ARIMA, SARIMA, VAR, ST-ANN, DeepST, RNN, LSTM, GRU
3. 预处理：对inflow, outflow，使用Min-Max归一化将数据映射到[-1,1]上，以进行cost计算由于输出层激活函数：tanh → [-1,1]，评估时将预测值rescale回原来量纲值
4. 超参数：
   - 使用Keras内置的均匀分布来初始化超参数。
   - Conv1和所有的残差单元使用64个大小为3×3的卷积核，Conv2使用2个大小为3×3的卷积核
   - 使用Adam优化算法，$batch_size$=32
   - TaxiBJ使用12个残差单元，BikeNYC使用4个残差单元
   - 时间区间间隔p,q固定，p取1天，q取1周
   - $l_c \in \{1,2,3,4,5\}, l_p \in \{1,2,3,4\}, l_q \in \{1,2,3,4\}$，得到80个不同的模型
   - 采用early-stop，在best validation score时停止训练，然后再使用所有训练集数据训练一段时间

Table 3: Details of convolutions and residual units

layer name	output size	closeness	period	trend
Conv1	32 × 32	$3\times3, 64$	$3\times3, 64$	$3\times3, 64$
ResUnit 1	32 × 32	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$
ResUnit 2	32 × 32	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$
ResUnit 3	32 × 32	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$
ResUnit 4	32 × 32	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$
Conv2	32 × 32	$3\times3, 2$	$3\times3, 2$	$3\times3, 2$

5. 评价指标：RMSE

4.4.2 实验结果

单步预测结果：

Table 5: Comparisons with baselines on TaxiBJ and BikeNYC. The results of ARIMA, SARIMA, VAR and DeepST on BikeNYC are taken from [3].

	RMSE	
Model	TaxiBJ	BikeNYC
HA	57.69	21.58
ARIMA	22.78	10.07
SARIMA	26.88	10.56
VAR	22.88	9.92
ST-ANN	19.57	7.57
DeepST	18.18	7.43
RNN-3	23.42	7.73
RNN-6	23.80	7.93
RNN-12	32.21	11.36
RNN-24	38.66	12.95
RNN-48	46.41	12.15
RNN-336	39.10	12.01
LSTM-3	22.90	8.04
LSTM-6	20.62	7.97
LSTM-12	23.93	8.99
LSTM-24	21.97	10.29
LSTM-48	23.02	11.15
LSTM-336	31.13	10.71
GRU-3	22.63	7.40
GRU-6	20.85	7.47
GRU-12	20.46	6.94
GRU-24	20.24	11.96
GRU-48	21.37	9.65
GRU-336	31.34	12.85
ST-ResNet	**16.89** (12 residual units)	**6.33** (4 residual units)
ST-ResNet-noExt	17.00 (12 residual units)	\

多步预测结果：

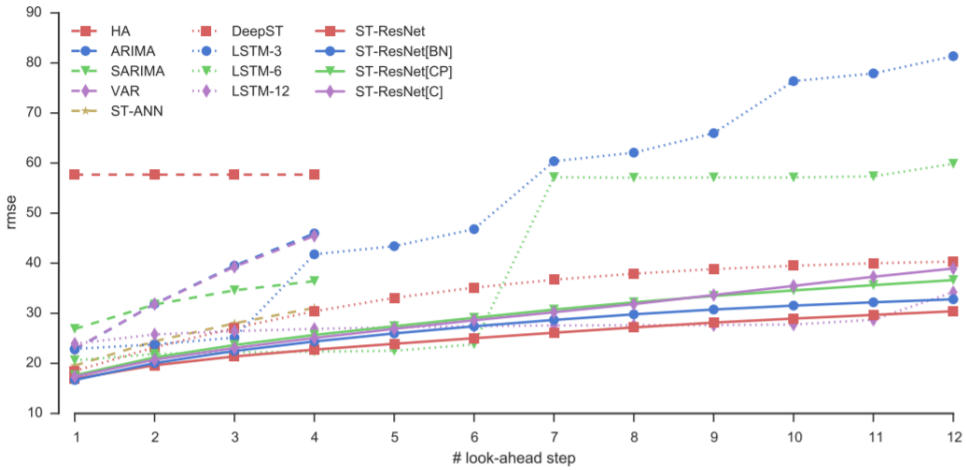


Figure 16: Multi-step ahead prediction

LSTM-12在step更大时表现更好：可能是由于LSTM同时输入了过去12个时段数据，而ST-ResNet只是用了前三个时段的数据。

5 Traffic speed prediction using deep learning method

论文引用：Jia, Y., Wu, J., & Du, Y. (2016, November). Traffic speed prediction using deep learning

method. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) (pp. 1217-1222). IEEE.

5.1 引言

交通流速预测是交通信息预测的子领域。作者将当前交通信息的预测方法分为两个主要流派：参数模型（parametric model）和非参数模型（non-parametric），并分别介绍了不同流派的代表模型以及已有研究取得的成果。

作者着重指出，随着近年来深层神经网络的兴起，深度学习方法逐渐被用于交通信息的预测中，并被证明在大规模数据集上能够取得很好的效果。特别地，由于深度信念网络（DBN）能够在无监督情况下很好地提取数据中的非线性关系，已经开始被应用于交通流预测中。

参数模型主要包含：自回归滑动平均模型（ARIMA）及其衍生模型（ARIMAX/SARIMA），卡尔曼滤波（Kalman Filter）为代表的状态空间模型（state-space models）。

非参数模型主要包含一些机器学习模型，特别是浅层神经网络：反向传播网络（BPNN），长短时记忆神经网络（LSTM NN）。这些模型相比于传统的参数模型，能够预测的时间范围更长，效果也相对更好。

文中提到的深度学习模型包括：栈式自编码机（Stacked Autoencoder，SAE），深度信念网络（Deep Belief Network，DBN）。已有研究在DBN上使用多任务回归层，在交通流预测上取得不错效果。

文章动机：现有研究没有对比过DBN与传统统计参数模型以及浅层机器学习方法在交通流速预测上的具体效果。

5.2 模型

直接采用DBN网络。输入X为预测时间节点前的交通流速序列向量，输出Y为预测范围内的交通流速序列向量。

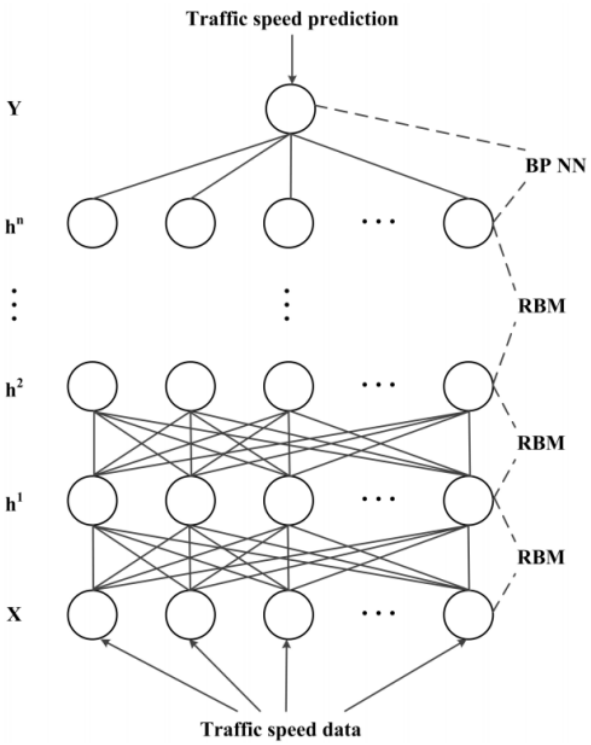


Figure 1. Structure of the DBN model.

训练过程：

1. 预训练：使用无监督学习方法训练每一层RBN（受限玻尔兹曼机）
2. 微调：再通过部分带标签数据，使用反向传播算法整体优化模型

5.3 实验

5.3.1 实验设置

1. 数据集：选用了2013年6月至2013年8月期间的北京二环三环间主干道（德胜门-马甸桥）交通流数据，通过探测器分别采集各车道数据，时间间隔为2min一次，包含速度、流量、占有率三大特征。划分数据集如下：训练集：6月1日-8月24日；测试集：8月25日-8月31日

   数据预处理：对各车道用加权方式进行聚合，得到输入输出的标准交通流速：

$$\text{Speed}_{\text{segment}} = \frac{\sum_i (\text{Flow}_i \times \text{Speed}_i)}{\sum_i \text{Flow}_i}$$

   其中，$\text{Flow}_i, \text{Speed}_i$分别代表第$i$个车道上的交通流量与速度。

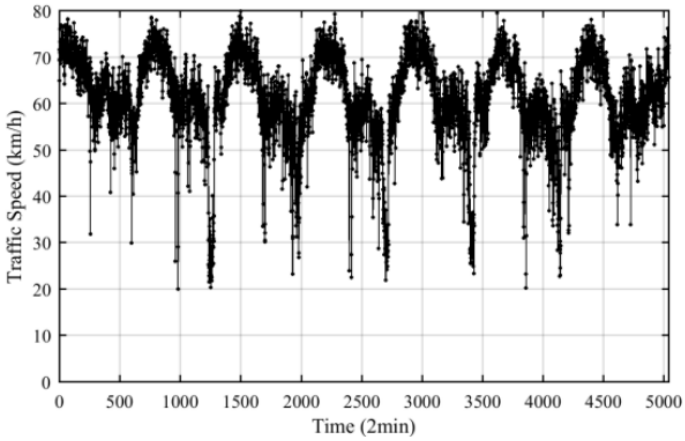


Figure 2. Traffic speed data on study segment for some week.

2. 超参数

   模型的超参数主要包含：

   - 输入序列长度（previous intervals）
   - 隐藏层层数（layer number）
   - 隐藏层各层神经元数（layer units）
   - 训练迭代次数（epochs）

   采用网格搜索方法，以MAPE（平均绝对百分比误差）为评价指标在训练集上进行测试，得到不同预测时间范围下的最优超参数组合：

TABLE I. OPTIMAL PARAMETERS FOR THE DBN MODEL.

Prediction horizon	*Previous intervals*	*Layer number*	*Layer units*	*Epochs*
2-minute	8	1	500	60
10-minute	12	1	400	85
30-minute	25	1	200	45

参数组合结果表明，对于当前较小的北京干线数据集，多个RBN层并不能取得更好的效果，一层RBN已经足够对其时空特征进行建模。但对于更大的数据集，或许需要使用更复杂的网络架构。

5.3.2 实验结果

分别测试了2min、10min和30min下DBN、ARIMA与BPNN的预测效果。实验结果证明：在不同预测时间跨度下，DBN的预测效果均优于ARIMA和BPNN（分别代表传统参数统计方法以及浅层机器学习方法）。

TABLE II. EXPERIMENT RESULTS

Prediction horizon	*Model*	*MAPE*	*RMSE*	*RMSN*
2-minute	DBN	5.8099%	4.3585	0.0710
	BPNN	5.9681%	4.4880	0.0731
	ARIMA	5.9752%	4.5116	0.0735
10-minute	DBN	7.3359%	5.5365	0.0902
	BPNN	7.5277%	5.7688	0.0940
	ARIMA	7.8387%	6.0834	0.0991
30-minute	DBN	8.4782%	6.3109	0.1029
	ARIMA	8.8433%	6.5561	0.1069
	BPNN	8.9312%	6.9797	0.1137

其他实验结论：随着预测时间长度的增加，三种预测方法预测结果的准确度都在下降，即：更多交通流的随机特征被丢失。这种准确度的下降尤其体现在早晚高峰时段。

6 Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction

论文引用：Yao, H. , Wu, F. , Ke, J. , Tang, X. , Jia, Y. , Lu, S. , et al. (2018). Deep multi-view spatial-temporal network for taxi demand prediction.

6.1 引言

1. 问题背景：高效的交通系统需要精准的出行需求预测系统，使得交通系统能够提前进行资源分配，避免不必要的能源消耗。当前网约车服务的兴起，使得人们能够采集到大量交通数据，如何借助海量的数据预测出行需求进行在AI界引发了越来越多的关注。

2. 研究问题：如何利用历史数据对某一区域下一时刻的打车服务需求进行预测。

3. 文章提出将CNN与LSTM放在一个框架下以同时捕获两种关系，其关键思想在于：

   - Local CNN的提出：作者认为在输入中包含弱相关区域会损害模型的性能，只考虑输入空间上邻近的区域（空间上邻近区域常常认为具有更强的需求模式相关性）
   - 图与图嵌入方法的使用：Local CNN的使用会忽略一些空间距离远但需求模式相关度较高的区域，因此可以使用图来描述这种需求模式语义相关性（边权为需求模式相关度），并通过图嵌入方法将其作为环境特征输入到模型中。

4. 文章贡献：

   - 提出能够同时包含空间、时间以及语义关系的统一多视图模型
   - 提出能够捕捉本地特征及与邻近区域关系的local CNN模型
   - 根据需求模式的相似性建立了一个区域图，用于对具有相关性的远距离区域进行建模。这种隐含的语义关系通过图嵌入方式进行学习。
   - 在滴滴的大规模数据集上进行了大量的测试，结果一致表明其优于其他前沿预测方法

6.2 预备知识

基于2017 Zhang文章进行定义。

- 空间（locations，L）：将城市划分为不重叠的栅格区域：$L = l_1, l_2, \ldots, l_i, \ldots, l_N$。
- 时间（time intervals，T）：按时间顺序，以30min为区间划分时间片：$\mathscr{I} = I_0, I_1, \ldots, I_t, \ldots, I_T$。
- 出租车请求（taxi request，o）：$\Phi o.t, o.l, o.u\Psi$，三个分量分别代表时间戳、位置和用户身份编号
- 需求（demand，y）：$y_t^i = |\{o : o.t \in I_t \vee o.l \in I_i\}|$，$|\cdot|$代表集合的势，表示区域$i$在区间$t$的出租车总请求量
- 外部环境特征（context features）：所有外部环境特征（如时间特征、空间特征、气象因素）构成的外部环境特征向量$\mathrm{e}_t^i \in \mathbb{R}^r$，维度$r$为特征数量。
- 需求预测问题（demand prediction problem）定义如下：

$$y_{t+1}^i = \mathscr{F}(\mathscr{Y}_{t-h,\ldots,t}^L, \mathscr{E}_{t-h,\ldots,t}^L)$$

其中，$\mathscr{Y}_{t-h,\ldots,t}^L$代表历史需求（$t-h$到$t$时间段内），$\mathscr{E}_{t-h,\ldots,t}^L$代表所有区域的历史外部环境特征。

6.3 模型框架

分为空间、时间、语义三个视图。整体架构如下：

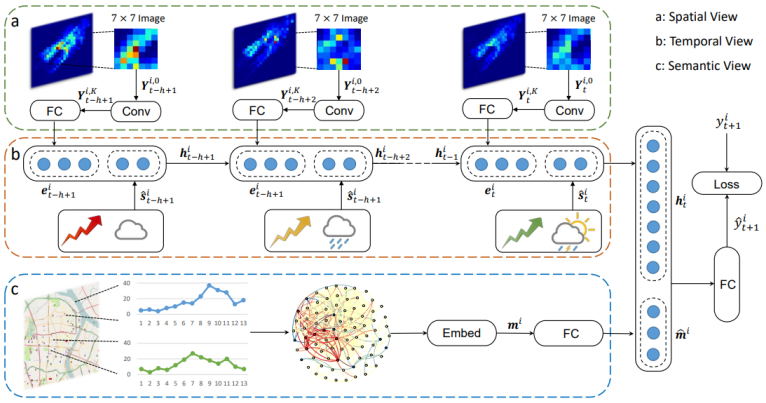


Figure 1: The Architecture of DMVST-Net. (a). The spatial component uses a local CNN to capture spatial dependency among nearby regions. The local CNN includes several convolutional layers. A fully connected layer is used at the end to get a low dimensional representation. (b). The temporal view employs a LSTM model, which takes the representations from the spatial view and concatenates them with context features at corresponding times. (c). The semantic view first constructs a weighted graph of regions (with weights representing functional similarity). Nodes are encoded into vectors. A fully connected layer is used at the end for jointly training. Finally, a fully connected neural network is used for prediction.

6.3.1 空间视图：Local CNN

对邻近区域的空间关系进行建模。

在每个时间段t，以区域i为中心，提取一个通道数为1的$S \times S$图像（值为对应需求数，S代表空间粒度，城市padding为0），可以得到对应的张量$\mathrm{Y}_t^i \in \mathbb{R}^{S \times S \times 1}$。将其作为输入$\mathrm{Y}_t^{i,0}$，通过$K$层卷积处理，输出$\mathrm{Y}_t^{i,K}$。

$$\mathrm{Y}_t^{i,k} = f(\mathrm{Y}_t^{i,k-1} * \mathrm{W}_t^k + \mathrm{b}_t^k)$$

激活函数为ReLU，$\mathrm{W}_t^k, \mathrm{b}_t^k$为所有区域所共享。输出$\mathrm{Y}_t^{i,k} \in \mathbb{R}^{S \times S \times \lambda}$再通过展开层（flatten layer）映射为特征向量$\mathrm{s}_t^i \in \mathbb{R}^{S^2\lambda}$，该向量再经过一层全连接层来降维：

$$\hat{\mathrm{s}}_t^i = f(\mathrm{W}_t^{fc}\mathrm{s}_t^{i,k-1} + \mathrm{b}_t^{fc}), \text{其中} \hat{\mathrm{s}}_t^i \in \mathbb{R}^d$$

6.3.2 时间视图：LSTM

对需求时间序列的序列关系进行建模。

使用LSTM，其公式如下：

$$\begin{aligned}
\mathrm{i}_t^i &= \sigma(\mathrm{W}_i\mathrm{g}_t^i + \mathrm{U}_i\mathrm{h}_{t-1}^i + \mathrm{b}_i) \\
\mathrm{f}_t^i &= \sigma(\mathrm{W}_f\mathrm{g}_t^i + \mathrm{U}_f\mathrm{h}_{t-1}^i + \mathrm{b}_f) \\
\mathrm{o}_t^i &= \sigma(\mathrm{W}_o\mathrm{g}_t^i + \mathrm{U}_o\mathrm{h}_{t-1}^i + \mathrm{b}_o) \\
\theta_t^i &= \tanh(\mathrm{W}_g\mathrm{g}_t^i + \mathrm{U}_g\mathrm{h}_{t-1}^i + \mathrm{b}_g) \\
\mathrm{c}_t^i &= \mathrm{f}_t^i \circ \mathrm{c}_{t-1}^i + \mathrm{i}_t^i \circ \theta_t^i \\
\mathrm{h}_t^i &= \mathrm{o}_t^i \circ \tanh(\mathrm{c}_t^i)
\end{aligned}$$

$\mathrm{i}_t^i, \mathrm{f}_t^i, \mathrm{o}_t^i$分别为输入、遗忘、输出门，$\mathrm{g}_t^i \in \mathbb{R}^{r+d}$为空间视图输出与环境因素向量的拼接向量，最终输出$\mathrm{h}_t^i$：

$$\mathrm{g}_t^i = \hat{\mathrm{s}}_t^i \oplus \mathrm{e}_t^i$$

6.3.3 语义视图：结构化嵌入

具有相同功能属性的区域会有相近的需求模式，但它们在空间上可能相隔很远。因此，文章通过建立全连接图$G(V,E,D)$来描绘这种语义上的相关性。节点集合V即区域集合L，$E \in V \times V$，边权D为相似度ω。

$$w_{ij} = \exp(-\alpha \mathrm{DTW}(i,j))$$

α为距离衰减因子（$\alpha = 1$），DTW(i, j)为两个区域需求模式间的DTW（动态时间规整）值（描述两个序列的相似程度指标，这里比较的序列为平均周需求序列）

使用图嵌入方法LINE将该图嵌入到低维向量空间得到m^i，再通过全连接层转化为$\hat{\mathrm{m}}^i$以实现整个网络的共同训练：

$$\hat{\mathrm{m}}^i = f(W_{fe}\mathrm{m}^i + b_{fe})$$

6.3.4 预测元件

将时间视图和语义视图输出拼接为q_t^i：

$$\mathrm{q}_t^i = \mathrm{h}_t^i \oplus \hat{\mathrm{m}}^i$$

整合通过全连接层输出$[0, 1]$的最终值（输入已标准化）：

$$\hat{y}_{t+1}^i = \sigma(W_{ff}\mathrm{q}_t^i + b_{ff})$$

6.3.5 训练过程

考虑到MSE易受极端值支配，损失函数由MSE和MAE组成：

$$\mathscr{L}(\theta) = \sum_{i=1}^{N}((y_{t+1}^i - \hat{y}_{t+1}^i)^2 + \gamma(\frac{y_{t+1}^i - \hat{y}_{t+1}^i}{y_{t+1}^i})^2)$$

训练过程算法如下，采用Adam优化求解器，使用Tensorflow和Keras搭建网络架构：

Algorithm 1: Training Pipeline of DMVST-Net

Input: Historical observations: $\mathcal{Y}_{1,\ldots,t}^L$; Context features: $\mathcal{E}_{t-h,\ldots,t}^L$; Region structure graph $G = (V, E, D)$; Length of the time period h;
Output: Learned DMVST-Net model

```
Initialization;
for ∀i ∈ L do
    Use LINE on G and get the embedding result m^i;
    for ∀t ∈ [h, T] do
        S_spa = [Y^i_{t-h+1}, Y^i_{t-h+2}, ..., Y^i_t];
        S_cox = [e^i_{t-h+1}, e^i_{t-h+2}, ..., e^i_t];
        Append < {S_spa, S_cox, m^i}, y^i_{t+1} > to Ω_bt ;
    end
end
Initialize all learnable parameters θ in DMVST-Net;
repeat
    Randomly select a batch of instance Ω_bt from Ω;
    Optimize θ by minimizing the loss function Eq. (9)
      with Ω_bt
until stopping criteria is met;
```

6.4 实验

6.4.1 实验设置

1. 数据集筛去了需求数¡10的样本。

项目	信息
数据集	广州市滴滴出行数据
时间	2017.2.1-2017.3.26
区域划分	20×20，0.7km×0.7km
环境特征	时间特征（前4时间片平均）、空间特征（区域中心经纬度）、天气特征、活动特征（节假日）
取样时间间隔	30分钟
输入时间序列	8×30min=4h
训练集	前47天
测试集	后7天

2. 比较基准：

   - Historical average (HA)
   - Autoregressive integrated moving average (ARIMA)
   - Linear regression (LR)：包含最小二乘回归、岭回归、lasso回归
   - Multiple layer perceptron (MLP)：(128,128,64,64)
   - XGBoost
   - ST-ResNet

   同时进行了使用不同组件的效果对比：

   - 时间视图
   - 时间视图+语义视图
   - 时间视图+空间视图（直接使用邻居节点需求）：为了与后者比较，体现出LCNN的优点
   - 时间试图+空间视图（Local CNN，LCNN）
   - DMVST-Net

3. 预处理

   - 历史需求：进行Max-Min标准化
   - 环境因素：对离散变量进行独热编码，连续变量Max-Min标准化

4. 超参数

超参数	值
粒度S	9
卷积层数K	3
卷积核大小τ	3×3
卷积核数λ	64
输出维度d	64
序列长度h	8
图嵌入输出维数	32
语义层输出维数	6
批大小	64
early-stop round	10
max epoch	100

训练集中，前90

6.4.2 实验结果

Table 1: Comparison with Different Baselines

Method	MAPE	RMSE
Historical average	0.2513	12.167
ARIMA	0.2215	11.932
Ordinary least square regression	0.2063	10.234
Ridge regression	0.2061	10.224
Lasso	0.2091	10.327
Multiple layer perceptron	0.1840	10.609
XGBoost	0.1953	10.012
ST-ResNet	0.1971	10.298
DMVST-Net	**0.1616**	**9.642**

Table 2: Comparison with Variants of DMVST-Net

Method	MAPE	RMSE
Temporal view	0.1721	9.812
Temporal + Semantic view	0.1708	9.789
Temporal + Spatial (Neighbor) view	0.1710	9.796
Temporal + Spatial (LCNN) view	0.1640	9.695
DMVST-Net	**0.1616**	**9.642**

其他结果：

1. 一周不同时间不同方法效果对比周末预测效果差于工作日，即周末更难进行预测。（工作日的白天都需要通勤，形成相似需求模式）

   因此，文章使用工作日相比周末预测误差的相对增加作为指标，测试了模型的健壮性，结果证明SMVST-Net的健壮性要普遍优于其他方法。

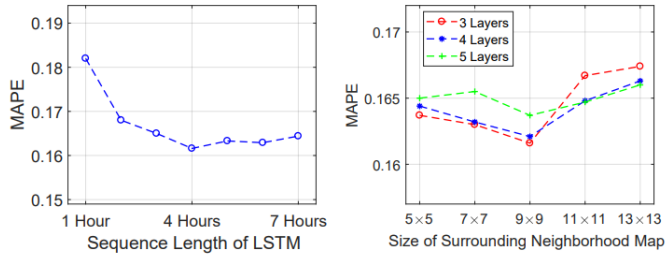
$$\frac{\bar{wk} - \bar{wd}}{\bar{wd}}$$

$\bar{wk}, \bar{wd}$分别代表工作日和周末的平均预测误差。

2. LSTM输入序列长度和LCNN输入粒度对结果的影响

   输入序列长度：随着输入序列长度增加，误差下降，但超过4h后，参数不断增多，使得训练变得困难，误差不再继续下降。

   输入粒度大小：当粒度超过9后，随着选择邻近范围大小的增加，本地的显著关联会逐渐被平均掉，证明了LCNN的有效性。



(a) (b)

Figure 3: (a) MAPE with respect to sequence length for LSTM. (b) MAPE with respect to the input size for local CNN.

6.5 结论与讨论

测试结果证明SMVST-Net在时空关系和语义关系的捕捉上取得了较好的效果，优于其他模型。此外，文章将在后续继续探究如何进一步改进性能以获得更好的解释性，以及将更多的隐藏信息（如POI）纳入模型考虑范围内。

7 Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting

论文引用：Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Transactions on Intelligent Transportation Systems.

7.1 引言

1. 对比了两类模型（传统统计方法和机器学习模型）的优缺点，引出了深度神经网络模型的研究。

2. 指出了目前应用的RNN模型和CNN模型的特点和缺陷。其中，CNN模型只适用于规则的欧式数据（二维的矩阵、图像等），无法应用于更一般的非欧式数据，因此从CNN中学习到的空间特征不能完全表现交通网络结构。

3. 介绍目前有关图卷积的研究：

   针对CNN的主要缺陷，大量研究集中于如何将卷积算子拓展到更一般的图结构数据上。目前有两种主要的示性图卷积的方法：一种利用了图谱理论，基于拉普拉斯矩阵，缺点在于不能充分捕捉图的特殊属性，且由于采用了多层图卷积层，学习到的空间依赖性缺乏可解释性。另一

种形式的图卷积是动态地对图数据进行的，例如图卷积中的动态边条件滤波器，高阶自适应图卷积网络。缺点在于不能完全考虑交通网络的物理特性。

以上图卷积网络的共同缺陷是卷积算子的接受域没有根据交通网络的真实结构局限在图中。例如在频谱图卷积网络中可以从图中某个顶点的K-局域邻居中获取特征，但是如何选择K的值，以及局域邻居是否真正影响到顶点，仍然是有待回答的问题。

4. 文章贡献

- 为了适应交通网络的物理特性，提取综合特征，提出了一种交通图卷积算子，以学习交通网络中真正有影响的邻域的特征。
- 提出了一种交通图卷积LSTM神经网络，用于学习交通数据中复杂的空间和动态时间依赖关系。3.为了使学习到的局部化图卷积特征更具一致性和可解释性，我们提出了两个正则化项，包括一个关于图卷积权重的L1范数和一个关于交通图卷积特征的L2范数，可以选择性地添加到模型的损失函数中。

7.2 模型

7.2.1 交通网络图

交通网络图的特点：

- 图中没有孤立的节点/边，图结构很少发生变化
- 各道路的交通状况，即节点和边的属性随时间变化
- 图中表示道路的边具有有意义的物理特征，如道路长度、类型、速度限制和车道数。

表示方法：考虑交通网络为一个无向图$G=(V,\varepsilon)$。虽然现实中有一些道路是定向的，但由于这些道路上发生的交通拥堵的影响会双向传播到的上下行道路，因此根据这种双向影响考虑G为一个无向图。

Neighborhood matrix 邻居矩阵 一个节点的交通状态不仅会受到相邻节点的影响，其自身在当前时刻的交通状态也会影响它未来的状态，因此基于交通流量图的邻接矩阵，定义一个同时包含邻居节点和自身的neighborhood matrix 邻居矩阵，来描述图的一阶邻居关系。记A为邻接矩阵，一阶邻居矩阵就是

$$\tilde{A}=A+I$$

其中I为单位矩阵。k阶邻居矩阵可以用$(A+I)^k$来描述。由于$(A+I)^k$可能存在元素大于1，而k阶邻居矩阵本身的意义在于用0-1元素标记节点之间的邻居关系，因此定义k阶邻居矩阵为

$$\tilde{A}^k_{i,j}=\min((A+I)^k_{i,j},1)$$

free-flow reachable matrix 自由流可达矩阵 考虑到道路网络中车辆交通的基本物理特性，我们需要明白，一个道路段对相邻路段的影响主要通过两种方式传递:1)减速和/或阻塞向上传播;和2)下游行驶的特定车辆组的驾驶员行为和车辆特征。因此，对于基于交通网络的图或其他类似图，非相邻节点之间的流量影响传输不能绕过中间节点/节点，因此需要考虑相邻节点与邻近节点之间影响的可达性。为了保证k-hop相邻节点间的交通影响传输遵循已建立的交通流理论，我们定义了一个自由流可达矩阵

$$FFR_{i,j}=\begin{cases}1 & S^{FF}_{i,j}m\Delta t-Dist_{i,j}\geq 0\\ 0 & otherwise\end{cases}$$

$S_{i,j}^{FF}$为节点i与j之间的自由流速度（free-flow speed），即在没有拥堵或其他不利条件(如恶劣天气)的情况下，一个驾车者所能行驶的平均速度。 m用来度量在自由流速度下行驶的距离的时间间隔Δt的数量。

$FFR_{i,j}$衡量了车能否在一定数量的时间间隔内以自由流速度从节点i移动到节点j。FFR矩阵的所有对角元素均为1。

流量预测问题 学习一个函数$F(\cdot)$，将历史T个时间步的交通流量图数据$X_T = [x_1, \cdots, x_t, \cdots, x_T]$，映射到接下来一个或多个时间步的图。

$$F([x_1, \cdots, x_t, \cdots, x_T]; G(V, \varepsilon, \tilde{A}^k, FFR)) = x_{T+1}$$

7.2.2 交通图卷积算子TGC

神经网络卷积层的核心思想是从二维或三维矩阵结构的输入数据中提取局部特征。基于邻居矩阵，以一阶邻居为感受野，定义一阶图卷积算子

$$\tilde{A}x_t W$$

类似地，定义k阶交通图卷积算子

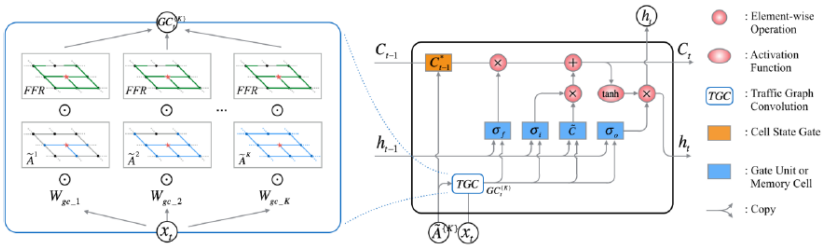
$$GC_t^k = (W_{gc_k} \odot \tilde{A}^k \odot FFR)x_t$$

$W_{gc_k} \in \mathbb{R}^{N \times N}$为要训练的参数，衡量了节点之间的交互影响，提高了模型的可解释性。（频谱图卷积中，Laplacian矩阵，是度矩阵和邻接矩阵结合，理论上来说，相比之下，FFR矩阵更多地考虑了节点之间的交通可达性，以及在流量上的相互影响）当k越大，感受野越大，提取到的空间特征就越丰富，但当k增大到一定程度时，$\tilde{A}^k \odot FFR$收敛于FFR，$\tilde{A}$作为提取局部特征的媒介就失去了意义，而且k增大会增加计算成本。用历史时刻t的交通流量图，用以上定义的TGC卷积算子操作，可以得到

$$GC_t^K = [GC_t^1, GC_t^2, \cdots, GC_t^K]$$

与谱图卷积方法得到的算子SGC和LSGC的比较，优点为结合了道路的物理属性；缺点为参数个数过多，如果图节点很多，训练难度就很大，且计算复杂度高。

7.2.3 TGC-LSTM网络



网络架构 基于以上定义的图卷积算子，设计了交通图卷积LSTM递归神经网络，可以很好地捕捉交通数据中复杂的空间依赖性和动态时间依赖性。

如网络图所示，用交通图卷积算子对输入进行处理得到$GC_t^K \in \mathbb{R}^{KN}$，作为LSTM网络的输入。与一般的LSTM不同之处在于，为了让交通图中每个节点的LSTM单元状态被邻居节点影响，在LSTM当前单元的前一个单元状态C_{t-1}传递过来之后，进行如下操作：

$$C_{t-1}^* = W_N \odot (\tilde{A}^K \odot FFR) \cdot C_{t-1}$$

其中权重矩阵W_N用来衡量邻居节点的单元状态对该节点状态变化的贡献。该节点的单元状态和隐藏状态则为：

$$C_t = f_t \odot C^*_{t-1} + i_t \odot \tilde{C}_t$$

$$h_t = o_t \odot \tanh(C_t)$$

在最后一个时间步T，输出T时刻的预测值$\hat{y}_T = h_T$。

规范化 为了让图卷积得到的特征在一个合理的规模下，并且让权重参数更加稳定和可解释，提出两个规范化项：

1. 图卷积算子权重W_{gc}的L1规范化

$$R^{\{1\}} = \|W_{gc}\|_1 = \sum_{i=1}^{K} |W_{gc_i}|$$

使W_{gc}稀疏且稳定，从而更直观地区分出哪一个相邻节点或哪一组节点贡献最大。

2. 图卷积特征GC_T^K的L2规范化

考虑到相邻节点对某一特定节点的影响必须通过相关节点与影响节点之间的所有节点进行传输，因此在图卷积中从不同跳数中提取的特征不应发生显著变化。因此，在每个时间步的损失函数上添加一个基于l2范数的TGC特征正则化项：

$$R^{\{2\}} = \|GC_T^{\{K\}}\|_2 = \sqrt{\sum_{i=1}^{K-1} (GC_T^i - GC_T^{i+1})^2}$$

损失函数采用MSE，

$$Loss = L(y_T, \hat{y}_T) = L(x_{T+1}, h_T) + \lambda_1 R^{\{1\}} + \lambda_2 R^{\{2\}}$$

7.3 实验

7.3.1 实验设置

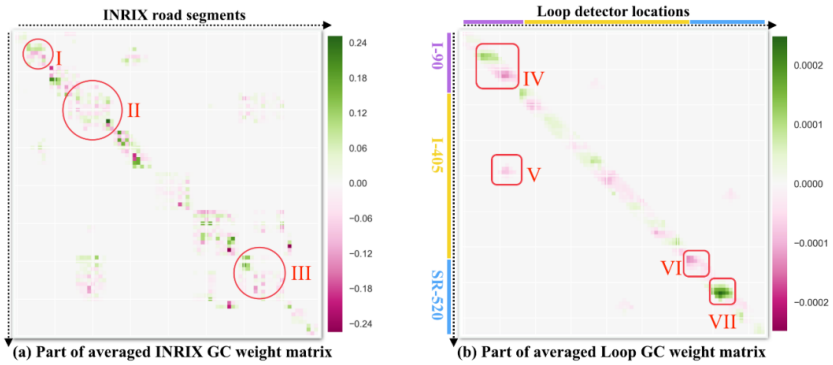
1. 数据集:
   - LOOP data：2015全年，the Greater Seattle Area，每五分钟，323个节点
   - INRIX data：2012全年，the Seattle downtown area，GPS探测数据，1014个路段，每五分钟
2. 对比模型：ARIMA, SVR, FNN, LSTM, DiffGRU, Conv+LSTM, LSGC+LSTM, SGC+LSTM, TGC-LSTM
3. 超参数：
   - 邻居矩阵的hops数K=3
   - 规范化参数$\lambda_1 = \lambda_2 = 0.01$
   - 学习率10^{-5}
   - mini-batch大小为10
   - 梯度下降算法：RMSProp，$\alpha = 0.99, \varepsilon = 10^{-8}$
   - 预测长度：10步
4. 评价指标：MAE, MAPE, RMSE

7.4 实验结果

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES. (THE NUMBER OF HOPS K IS SET AS 3 IN THE GRAPH CONVOLUTION RELATED MODEL)

Model	LOOPf Data			INRIX Data		
	MAE (mph)±STD	MAPE	RMSE	MAE (mph)±STD	MAPE	RMSE
ARIMA	6.10± 1.09	13.85%	10.65	4.80 ± 0.32	13.51%	10.85
SVR	6.85± 1.17	14.39%	11.12	4.78 ± 0.37	13.37%	10.44
FNN	4.45± 0.81	10.19%	7.83	2.31 ± 0.17	8.35%	5.92
LSTM	2.70± 0.18	6.83%	4.97	1.14 ± 0.09	3.88%	2.43
DiffGRU	4.64±0.38	11.18%	8.22	2.44 ± 0.09	8.91%	6.34
Conv+LSTM	2.71±0.12	6.79%	5.02	1.13 ± 0.08	3.80%	2.37
LSGC+LSTM	3.16± 0.23	7.51%	6.18	1.38 ± 0.12	4.54%	2.82
SGC+LSTM	2.64± 0.12	6.52%	4.80	1.07 ± 0.08	3.74%	2.28
TGC-LSTM	**2.57**± 0.10	**6.01%**	**4.63**	**1.02** ± 0.07	**3.28%**	**2.18**

1. TGC-LSTM在所有指标上的表现都最好。
2. 随着K的增加，模型表现会变好，但是提升的效果量边际递减。
3. 训练效率：收敛所需要的epoch数量相比其他模型较少，一个epoch训练时长比较大。且K越大，训练损失收敛率越大。
4. 规范化效果



该图为相邻节点间的权重贡献图。这些点形成了多个集群，显示了几个附近或连接的路段的权重。考虑到路段受邻近路段或邻近连通路段的影响，集群中绝对权重较大的节点极有可能成为局部交通网络中的关键路段。这样，我们可以从交通图的卷积权值矩阵中推断出交通网络的瓶颈。