

深度学习与交通大数据实战

V1.0

By

张金雷

北京交通大学博士研究生

华盛顿大学联合培养博士生



公众号二维码



个人微信二维码

目录

1	说明	9
2	引言---第一次, 我们就这样不期而遇	9
3	你想要的交通大数据和学习资料, 分享给你!	12
3.1	Attention	12
3.2	滴滴出租车 GPS 数据及订单数据	13
3.3	Oracle 视频教程、笔记及软件	13
3.4	Python 视频教程及高清 PDF 书籍	13
3.5	机器学习视频及高清 PDF 书籍	13
3.6	ArcGIS 视频教程及软件	14
4	国内外其他交通数据集	14
5	LSTM:首篇 LSTM 网络用于交通速度预测	15
5.1	背景	15
5.2	创新点	16
5.3	主体内容	16
5.4	疑问	17
5.5	展望	17
5.6	Writing Tips from This Paper	17
6	AutoEncoder: 如何使用 AutoEncoder 预测客流量	18
6.1	文章信息	18
6.2	Introduction 和 Literature Review	18
6.3	本文的简介部分组成	18
6.4	创新点	19
6.5	主体内容	19
6.6	特别之处	23
6.7	展望	23
6.8	Writing Tips from This Paper	23
7	ConvLSTM:从短时降水量预测中学习 ConvLSTM	24
7.1	文章信息	24
7.2	摘要	25

7.3	简介和文献综述.....	25
7.4	创新点.....	26
7.5	主体内容.....	26
7.6	Experiment	28
7.7	结论和展望.....	28
8	Sequence to Sequence: 从 NLP 自然语言处理到工程领域中的时间序列 ...	29
8.1	文章信息.....	29
8.2	摘要.....	30
8.3	简介.....	30
8.4	创新点.....	32
8.5	主体内容.....	32
8.6	特别之处.....	33
8.7	展望.....	34
9	ConvLSTM& Bi-LSTM: 基于深度学习的短时道路交通流预测	34
9.1	文章信息.....	34
9.2	主体内容.....	35
9.3	创新点.....	35
9.4	大家比较关注的算法实现.....	35
9.5	ConvLSTM 的结构	36
9.6	双向 LSTM 的结构 (Bi-directional LSTM)	37
10	CNN: 使用卷积神经网络做交通速度预测.....	38
10.1	文章信息.....	38
10.2	摘要.....	38
10.3	简介和文献综述.....	39
10.4	理论部分.....	39
10.5	案例研究将在下篇文章中叙述.....	41
10.6	Writing Tips from This Paper	41
11	CNN: 使用卷积神经网络做交通速度预测-案例简介	41
11.1	文章信息.....	42
11.2	案例研究.....	42

11.3	展望.....	44
12	卷积神经网络如何处理一维时间序列数据?	45
12.1	概述.....	45
12.2	何时应用 1D CNN?	45
12.3	1D CNN 和 2D CNN 之间有什么区别?	45
12.4	问题描述.....	46
12.5	如何在 Python 中构造一个 1D CNN?	47
12.6	训练和测试该神经网络.....	49
13	考虑动态时空相似性的交通预测深度学习框架.....	50
13.1	文章信息.....	50
13.2	摘要.....	51
13.3	简介.....	51
13.4	符号说明.....	52
13.5	STDN 时空动态网络	52
13.6	案例研究.....	56
13.7	总结.....	56
14	考虑动态时空相似性的交通预测深度学习框架-案例研究.....	56
14.1	文章信息.....	56
14.2	摘要.....	57
14.3	案例研究.....	57
14.4	思考.....	59
14.5	数据获取.....	59
15	ConvLSTM 参数详解.....	59
15.1	最重要的输入尺寸和输出尺寸.....	59
15.2	参数.....	60
16	ResNet: 基于深度时空残差网络 ResNet 的城市交通流预测.....	64
16.1	文章信息.....	64
16.2	摘要.....	64
16.3	简介.....	65
16.4	模型介绍.....	66

16.5	模型超参数.....	69
16.6	实验部分.....	69
16.7	思考.....	70
17	图卷积神经网络 GCN ---如何在交通预测中考虑网络拓扑结构.....	70
17.1	文章信息.....	70
17.2	摘要.....	70
17.3	简介.....	70
17.4	基础介绍.....	71
17.5	模型框架.....	72
17.6	Graph CNNs 提取空间特征.....	73
17.7	Gated CNN 门控提取时间特征.....	74
17.8	时空卷积块.....	74
18	2019 运输科技领域最新 SCIE 期刊影响因子.....	75
19	GCN :基于图卷积神经网络的共享单车流量预测	79
19.1	文章信息.....	79
19.2	摘要.....	80
19.3	简介.....	80
19.4	概念定义和问题描述.....	80
19.5	主体内容.....	81
19.6	案例研究.....	85
20	GCN: 基于图卷积神经网络的轨道交通流量预测.....	85
20.1	文章信息.....	85
20.2	摘要.....	85
20.3	简介.....	86
20.4	理论介绍.....	86
20.5	特别之处.....	90
21	OD 需求预测: 基于时空网络的出租车 OD 需求预测.....	90
21.1	文章信息.....	90
21.2	摘要.....	90
21.3	简介.....	91

21.4	预备知识.....	93
21.5	模型简介.....	95
21.6	总结.....	99
22	Multi-GCN: 基于时空多图卷积网络的网约车需求预测	99
22.1	文章信息.....	99
22.2	摘要.....	99
22.3	简介.....	100
22.4	文献综述.....	101
22.5	主体内容.....	102
22.6	总结展望.....	108
23	Attention 注意力机制: 基于时空注意力机制的图卷积神经网络用于交通	
预测	108	
23.1	文章信息.....	109
23.2	摘要.....	109
23.3	简介.....	109
23.4	主体内容.....	109
23.5	实验部分.....	115
23.6	展望.....	115
24	理解多图卷积: 基于多图卷积神经网络的多模式交通预测应用对比 ...	115
24.1	文章信息.....	116
24.2	摘要和简介.....	116
24.3	模型对比.....	117
24.4	对比思考.....	120
25	OD 需求预测: 基于图卷积神经网络 GCN 的出租车 OD 需求预测.....	121
25.1	文章信息.....	121
25.2	摘要.....	121
25.3	简介.....	122
25.4	模型框架.....	124
25.5	总结.....	129
26	Dynamic GCN: 基于动态时空图卷积神经网络的交通流预测	129

26.1	文章信息.....	129
26.2	摘要.....	130
26.3	简介.....	130
27	OD 需求预测：基于深度学习框架的出租车 OD 需求预测应用对比.....	131
27.1	文章信息.....	131
27.2	摘要.....	131
28	T-GCN-时间图卷积网络用于交通预测	134
28.1	文章信息.....	134
28.2	摘要.....	134
28.3	简介.....	134
28.4	主体部分.....	135
28.5	实验部分.....	139
29	多步交通需求预测：基于注意力机制的神经网络模型	143
29.1	文章信息.....	143
29.2	摘要.....	143
29.3	简介.....	143
29.4	模型框架.....	144
29.5	评论.....	148
30	交通预见未来(26): TrajGRU 降雨量预测-一个基准和一个新的模型.....	148
30.1	文章信息.....	148
30.2	摘要.....	148
30.3	简介.....	149
30.4	模型.....	150
30.5	Experiments on MovingMNIST++.....	155
30.6	降雨量预测的一个新的 benchmark.....	155
30.7	HKO-7 Dataset.....	155
30.8	评估方法.....	156
30.9	总结.....	157
31	共享单车：基于深度学习和多源大数据的浮动共享单车流量预测	157
31.1	文章信息.....	157

31.2	摘要.....	158
31.3	简介.....	158
31.4	数据集.....	158
31.5	模型框架.....	160
31.6	实验部分.....	161
31.7	数据获取.....	162
32	基于手机信令数据的动态出行 OD 及出行需求估计	162
32.1	文章信息.....	162
32.2	摘要.....	162
32.3	简介.....	163
33	时空循环卷积神经网络用于交通速度预测	164
33.1	文章信息.....	164
33.2	摘要.....	165
33.3	简介.....	165
33.4	主体内容.....	166
33.5	总结.....	169
34	结束语	170

1 说明

此文档为公众号《当交通遇上机器学习》的历史 SCI 论文原创解读，主要包含 30 多篇 7 万多字的深度学习在交通大数据领域内的 SCI 英文论文的中文解读，其中包含：LSTM、Bi-LSTM、CNN、Seq2Seq、ResNet、ConvLSTM、GCN 等多种深度学习模型及其变体的实战应用，研究问题包含流量预测、速度预测、OD 预测等，研究对象包含公、共享单车、地铁、出租车网约车、高速公路等。

获取方式：扫描下方二维码关注公众号《当交通遇上机器学习》，回复“文档”即可获取下载链接。



2 引言---第一次，我们就这样不期而遇

很久以前，心里就一直有一粒被压抑的种子，想做一个公开性的产品，或直播和亲友唱歌聊天，或录视频记录生活，或做公众号分享自己的喜怒哀乐。但，一直没有付诸行动~很庆幸现在的自己，有足够的时间和精力，做一些自己想做的事情。

首先向大家介绍下自己：

我是北京交通大学博士研究生，不高，不帅，也没钱，但庆幸的是，这个年纪了，我还愿意抱着对生活的热情，去努力尝试，让心里那颗正在发芽的种子，再拼着生长~



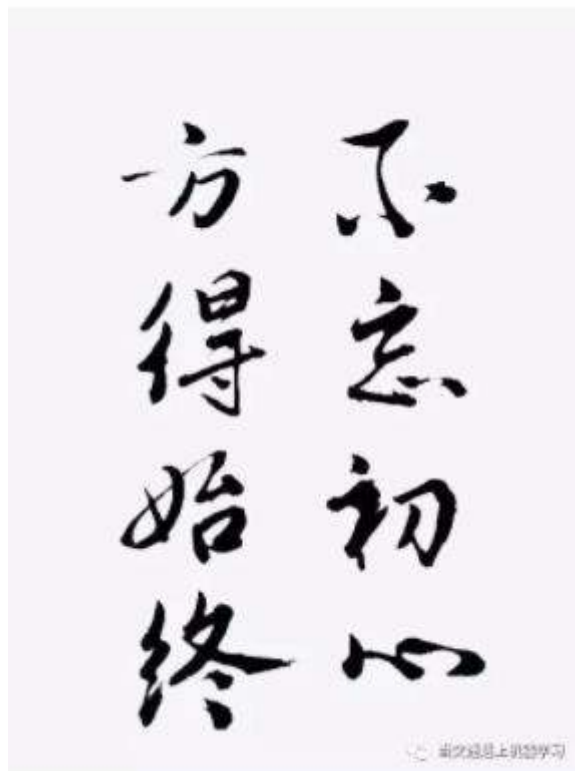
回望过去，博士生活已经过了一大半，但总感觉自己没有达到想要达到的水平，过去这两三年，我读了多少英文文章？我的英语写作能力提升了多少？我的科研水平提升了多少？我的为人处世能力又改变了多少？我不清楚，也没有底气。有时候突然就想，自己好 low 哦，为什么别人在大二的时候就能为公司开发小产品？为什么别人能拿全奖到国外读博？为什么别人在自己这个阶段能拿最高的知行奖学金？

因为，我不够努力，不够坚持。所以，我想做这个公众号，记录自己的科研成长之路。或许某天回过头来看，也会感慨，这小子，也曾努力坚持过！



想要做这个公众号，已经酝酿了有一段时间了，初衷不是为了博人眼球，只是想以科研为主线，分享自己博士生活中的点点滴滴，所以有没有人订阅，会不会有人喜欢，也不是最重要的。

我是基于机器学习做交通预测的，也算是交叉学科。（我这么说会不会被打，因为一直感觉自己在机器学习领域是很白很白的小白，有点拉低了机器学习这个名词的身份，哈哈哈）所以，公众号我给他起名叫做“当交通遇上机器学习”，有点北京遇上西雅图的意思，哈哈。公众号的内容也主要是分享机器学习在道路交通和轨道交通领域内的应用动态，偶尔可能也会分享分享自己的生活感悟。一周一到两篇文章的样子，一般会在周末发。刚开始，肯定会有很多很多很多我理解不对的地方，如果你愿意的话，指点指点我吧，因为我很想很想，成长再快一点。



当交通遇上机器学习，洞悉世界，预见未来，第一次，我们就这样不期而遇了，愿你，见证我的成长，愿你们，陪我走过剩下的不平坦的路~

这件事，我不知道自己能坚持多久，一周？两周？半年？两年？既然决定做了，那就尝试一下吧！未来的事，谁又和谁说得清呢~当遇见的生活不是你想要的那个样子时，想想自己来时的目标，或许，我们就可以走的更远一点。不忘初心，方得始终。与你，与我，共勉~

张金雷

2019年5月1日

于美国西雅图

3 你想要的交通大数据和学习资料，分享给你！

3.1 Attention

此次分享的资料主要有5类，其中视频教程都是我看过很适合初学者看的，

每类教程我只选了一个我觉得最好的，不好的不会推荐给大家的。分享的书籍有的我看过有的没看过，但也都是领域内很认可的书籍，总有一本适合大家的。关注后回复“数据”即可获得！

3.2 滴滴出租车 GPS 数据及订单数据

4 个月 174G 的滴滴出租车 GPS 数据和订单数据的获取方法及说明（对于做大数据的科研人员相当给力了）。

3.3 Oracle 视频教程、笔记及软件

Oracle 数据库资料：

非常适合初学者的 Oracle 数据库视频教程及 word 笔记；

Oracle11g 软件。

3.4 Python 视频教程及高清 PDF 书籍

python 资料：

非常适合初学者的 Python3 入门与进阶（慕课网）视频教程、Python 软件是开源的，推荐 Python3；

领域内非常认可的几本 python 相关的高清 PDF 书籍：机器学习周志华（西瓜书）、Python 基础教程（第 3 版，基于 Python3）、统计学习方法-李航、计量经济学基础 第五版 古扎拉蒂、波特。

3.5 机器学习视频及高清 PDF 书籍

机器学习：

吴恩达全套机器学习视频中英文字幕带 PPT；

针对于 python 和机器学习的高清 PDF 书籍：Python 机器学习（中文版）。

3.6 ArcGIS 视频教程及软件

ArcGIS:

ArcGIS 视频教程及 PPT（这是我参加培训的 video 及笔记总结，不太系统，但对初学者很实用）；

对应的 ArcGIS 及 ArcGIS Pro 软件；

以及《精通 ArcGIS 地理信息系统》的随书教学视频，比较系统，我没看过。

以上所有资料关注后回复“数据”即可获取！

如果你和我一样是轨道交通、交通运输、城市规划相关领域的，可以加下面的二维码扫码进群，人数满员后加微信：Dr_JinleiZhang，备注“进群”，我会拉你进交通大数据分享群！

4 国内外其他交通数据集

关注公众号：

后台回复“数据”即可获取高达 175G 的四个月的滴滴 GPS 数据和滴滴订单数据的获取方式，以及从数据处理（Oracle 数据库）、模型构建（机器学习）、编程实现（python）到可视化（ArcGIS）等一系列视频教程。

后台回复“纽约”获取美国纽约 10 年的出租车轨迹数据以及 7 年的共享单车轨迹数据下载地址。

后台回复“芝加哥”获取美国芝加哥 6 年的共享单车轨迹数据下载地址。

后台回复“加州”获取美国加州近 20 年的交通流量监测数据下载地址。

后台回复“GCN”获取 GCN 在交通领域内的应用总结 PPT。

后台回复“书籍”获取交通领域几本经典 pdf 书籍，包括城市规划原理，现代交通规划学，交通规划原理。

公众号以交通大数据为主线，专注于人工智能、机器学习、深度学习在道路交

通和轨道交通领域内的科研前沿与应用，在交通大数据与机器学习的道路上越走越远！

5 LSTM:首篇 LSTM 网络用于交通速度预测

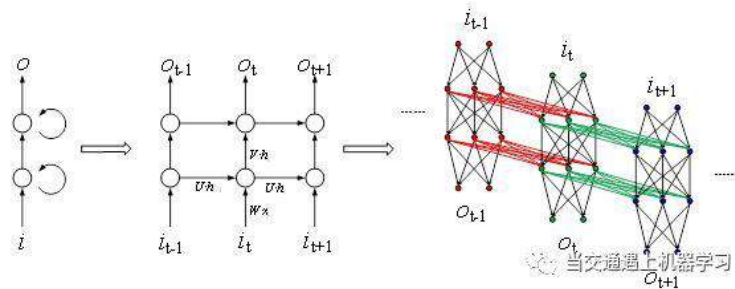
文章名称：《Long short-term memory neural network for traffic speed prediction using remote microwave sensor data》



这是北航马晓磊教授和华盛顿大学王印海教授于 15 年发在 Part C 上的一篇文章（运输科技 2 区，IF:3.968），短短 4 年时间，创造了近年来 PartC 单篇最高引用量，高达 300 多次。

5.1 背景

循环神经网络 RNN 之所以能够在自然语言处理 NLP 领域大火，很大程度上得益于长短期记忆网络 LSTM 的提出。LSTM 能有效解决传统 RNN 在处理时间序列长期依赖中的梯度消失和梯度爆炸的问题，这和很多交通领域的预测问题很相似，如何利用过去的交通状态数据预测未来的交通状态？在预测过程中哪些信息应该被保留下来，哪些信息应该被舍弃？过去很长时间的数据是否对预测当前状态有用？LSTM 能帮你很好的处理这些问题。



post 一张自己画的 RNN 图，哈哈哈

5.2 创新点

这篇文章之所以能得到很高的引用量，很大程度上在于这是第一篇将 LSTM 应用到交通领域的文章，这在科研领域大大体现了“人无我有，人有我精，人精我转”的魅力！这篇文章的主体内容其实不多，但文章将该领域的研究现状和文献综述描述的很是详细了，11 页的文章，文献综述占了 4 页，这也一定程度上体现了文章作者的用心。

5.3 主体内容

文章主要是借助道路交通的传感器数据（包括速度数据，客流量数据和道路占有率数据）来预测未来的道路交通速度。首先，讲述了 LSTM 的作用原理，其次，利用文章所使用的神经网络结构（一层输入，一层 LSTM，一层输出），以历史道路速度数据为输入，来预测未来的交通速度，并与其他多种预测方法诸如 Elman 神经网络，支持向量机 SVM，卡尔曼滤波器 Kalman filter 等进行对比。然后又以道路速度和交通流量作为输入，来预测未来的交通速度，以验证是否输入较多的信息，会有更准确的结果。

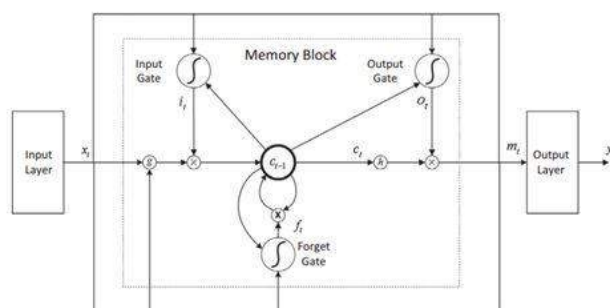


Fig. 1. LSTM neural network architecture.

当交通遇上机器学习

文章中的 LSTM 示意图

5.4 疑问

文章一直在强调其中的一个创新点是：LSTM can automatically calculate the optimal time lags, and thus no predetermined time window size is needed. 即 LSTM 能自动确定最优的 time lag, 因此在预测过程中不需要提前设置时间窗口，和几个小同行讨论过这个问题，大家在用这个算法过程中都会设置一个时间窗口，所以至今我也没搞明白作者在技术上是如何做到的，有做相关领域的大神还望指点一二~

5.5 展望

文章中给出的展望不像很多文章中列的是一些目前根本无法解决的问题，而是给出了很中肯的建议。例如探索将相邻传感器的数据作为输入、不同时间粒度下的预测效果对比、多层深度网络的使用。有此类数据的科研工作者不妨一试。

5.6 Writing Tips from This Paper

(1) Although traditional RNN exhibits a superior capability of modeling nonlinear time series problems in an effective fashion, there are still several issues to be addressed. 介绍既有研究的不足用。

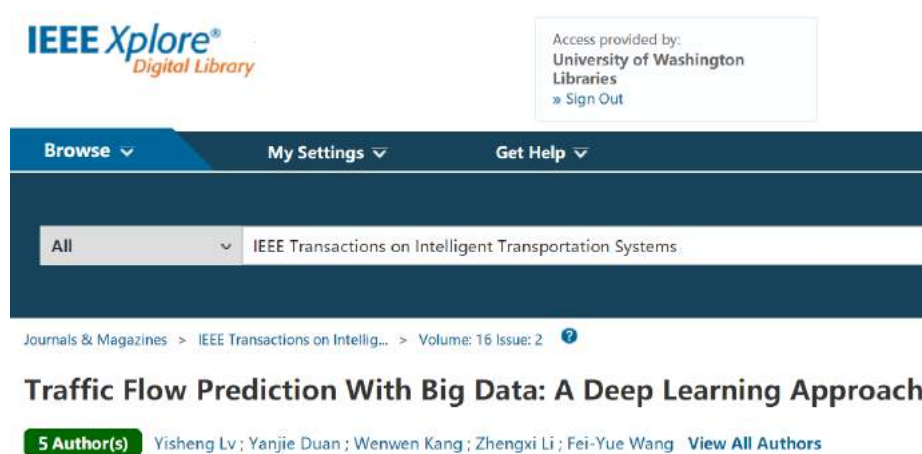
(2) To the best of our knowledge, there is no application of LSTM NN in the domain of transportation. 引出自己的创新点用。

6 AutoEncoder: 如何使用 AutoEncoder 预测客流量

6.1 文章信息

《Traffic Flow Prediction With Big Data: A Deep Learning Approach》。

中科院吕宜生老师 2015 年发在 IEEE Transactions on Intelligent Transportation Systems 上的一篇文章（运输科技 2 区，IF:4.051），被引 807 多次(可望不可及啊)。



6.2 Introduction 和 Literature Review

6.3 本文的简介部分组成

首先，介绍了道路交通短时客流预测的意义。其次，由于大数据时代下的数据驱动，基于机器学习的预测方法得到了很大的关注（引出下文）。紧接着，介绍了的深度学习优势：深度学习算法利用多层或深度神经网络体系结构，从最低层次到最高层次逐渐提取数据的固有特征，能够发现数据中大量的内在结构特征。由于交通流过程的复杂性，深度学习算法可以在不需要先验知识的情况下表征交通特征，对于交通流预测具有良好的性能。然后，介绍了文章的主要工作和创新点。最后简单写下文章结构。

本文的文献综述部分就是将既有的研究方法进行了分类，然后列了几个主流的

方法，每个方法后面罗列了一点文献。然后进行了如下两点总结：

(1) 很难说一种方法在任何情况下都明显优于其他方法。其中一个原因是，所提出的模型是用少量单独的特定交通数据建立的，而交通流预测方法的准确性依赖于所收集的时空交通数据中嵌入的交通流特征。

(2) 虽然神经网络的深层结构可以学习到比浅层网络更强大的模型，但是现有的基于神经网络的交通流预测方法通常只有一个隐含层。基于梯度的训练算法很难训练出层次较深的神经网络。深度学习的最新进展使得训练深度架构成为可能，这表明深度学习模型在某些领域具有优于或可与最先进的方法相媲美的性能。

6.4 创新点

第一次使用深度学习架构，嵌入 Stacked AutoEncoder (SAE) 堆叠自编码器作为主体网络结构块来预测交通流。

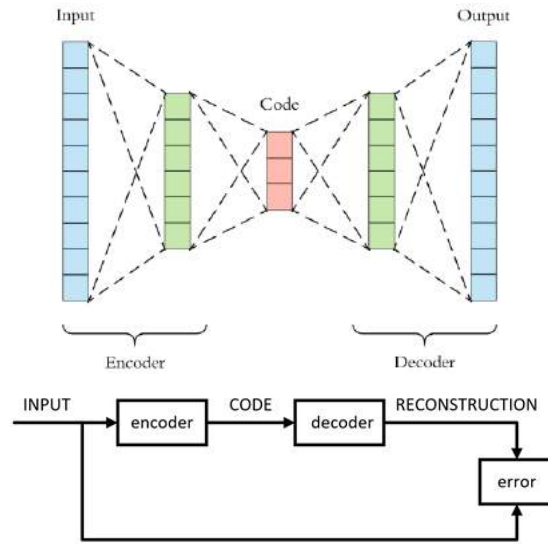
6.5 主体内容

何为 AutoEncoder ? 自编码也是一种网络结构, 即将数据输入到输入层, 通过编码和解码(其实也就是几层神经网络结构), 得到 output, 与其他网络不同的是, 其损失函数是使输入和输出之间的 error 尽可能小, 即使输出尽可能地还原输入, 如下所示, 输入是 X , 输出是 $Z(X)$, 损失函数是 θ 。

$$y(x) = f(W_1x + b) \quad (1)$$

$$z(x) = g(W_2y(x) + c) \quad (2)$$

$$\theta = \arg \min_{\theta} L(X, Z) = \arg \min_{\theta} \frac{1}{2} \sum_{i=1}^N \|x^{(i)} - z(x^{(i)})\|^2. \quad (3)$$



AE 示意图

SAE 模型是通过叠加自编码器, 将底层自编码器的输出作为当前层的输入, 形成一个深度网络。假设有 1 层的 SAEs, 第一层被训练为一个自动编码器, 训练集作为输入。在获得第一个隐层后, 将第 k 个隐层的输出作为第 $(k + 1)$ 个隐层的输入。这样, 多个自编码器就可以分层堆放。

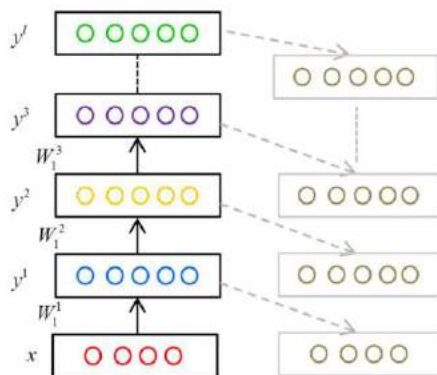
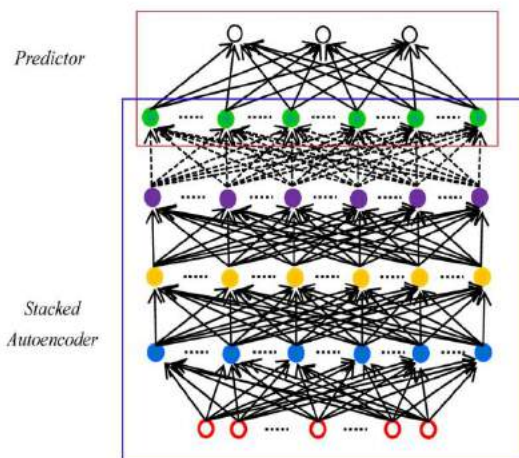


Fig. 2. Layerwise training of SAEs.

SAE 示意图

为了使用 SAE 网络进行流量预测，我们需要在顶层添加一个标准的预测器。在此基础上，提出了一种基于 logistic 回归的交通流预测方法。SAEs 加预测器构成了整个交通流预测的深层体系结构模型。



Model Structure

训练算法：

将 BP 方法与梯度下降算法相结合，可以直接训练深度网络。不幸的是，以这种方式训练的深层网络性能很差。最近，Hinton 等人开发了一种贪婪分层无监督学习算法，可以成功地训练深度网络。使用贪心分层无监督学习算法的关键是采用自底向上的方式逐层对深层网络进行预训练。在预训练阶段后，利用 BP 再自顶向下对模型参数进行调整。（看着高大上，原理懂，技术层面暂时还不会.....）

我们应该如何训练深度网络呢？贪婪训练方法是取得一定成功的一种方法。简单来说，逐层贪婪算法的主要思路是每次只训练网络中的一层，即我们首先训练一个只含一个隐藏层的网络，仅当这层网络训练结束之后才开始训练一个有两个隐藏层的网络，以此类推。在每一步中，我们把已经训练好的前 $k-1$ 层固定，然后增加第 k 层（也就是将我们已经训练好的前 $k-1$ 的输出作为输入）。每一层的训练可以是有监督的（例如，将每一步的分类误差作为目标函数），但更通常使用无监督方法（例如自动编码器）。这些各层单独训练所得到的权重被用来初始化最终（或者说全部）的深度网络的权重，然后对整个网络进行“微调”（即把所有层放在一起优化有标签训练集上的训练误差）。（原文：
<https://blog.csdn.net/dcxhun3/article/details/48131745>）

模型配置：

PeMS 系统采集的 30 秒时间间隔的交通流量数据，并以 5 分钟时间粒度进行积聚，3 个月工作日的数据，前两个月用来训练，第三个月用来测试。另外，需要确定输入层的大小、隐藏层的数量以及每个隐藏层中隐藏单元的数量。对于输入层，我们使用所有高速公路收集的数据 M 作为输入，即考虑到了交通流的空间相关性。考虑到交通流的时间关系，为了预测 t 时段的交通流，我们应该使用之前时段的交通流数据，即 X_{t-1} ， X_{t-2} X_{t-r} 。因此，该模型内在地考虑了交通流的时空相关性。输入空间的维数为 mr ，而输出的维数为 m ，其中 m 为高速公路的数量。

本文利用该模型分别对 15 分钟、30 分钟、45 分钟和 60 分钟的交通流进行了预测。我们选择 r 从 1 到 12，隐藏层的大小从 1 到 6，隐藏层单元的数量从 {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000}。

实验结果：

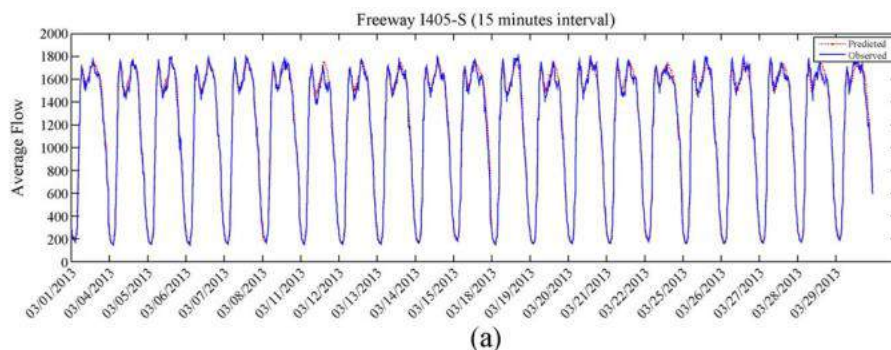


TABLE II
PERFORMANCE COMPARISON OF THE MAE, THE MRE, AND THE RMSE FOR SAEs, THE BP NN, THE RW, THE SVM, AND THE RBF NN

Task	Stacked Autoencoders			BP Neural Network			RW			SVM			RBF		
	MAE	MRE (%)	RMSE	MAE	MRE (%)	RMSE	MAE	MRE (%)	RMSE	MAE	MRE (%)	RMSE	MAE	MRE (%)	RMSE
15-min traffic flow prediction	34.1	6.75	50.0	60.8	10.9	94.1	38.3	7.8	56.7	38.7	8.0	62.3	38.3	7.4	55.9
30-min traffic flow prediction	64.1	6.48	95.2	114.3	11.3	173.3	125.0	12.1	182.6	115.5	10.3	188.3	120.0	13.0	177.3
45-min traffic flow prediction	92.0	6.17	138.1	151.2	10.2	237.0	260.0	17.1	374.7	220.0	15.8	350.4	228.6	16.4	335.6
60-min traffic flow prediction	122.8	6.21	183.9	202.8	9.8	321.5	445.0	22.3	633.4	372.9	22.1	607.5	443.4	26.4	652.6

6.6 特别之处

文章给出的结论中，不同时间粒度下的预测模型使用的网络结构是不同的，很值得借鉴！（15 分钟 3 层，每层 400 个神经元，30 分钟 3 层，每层 200 个神经元，45 分钟 2 层，每层 500 个神经元，60 分钟 4 层，每层 300 个神经元。）从结果中可以看出，隐藏层的最佳数量至少为 2 层，最多不超过 5 层。从经验中得到的教训表明，神经网络的隐层数既不应过小也不应过大。我们的结果证实了这些教训。

6.7 展望

本文的预测层只是一个逻辑回归。将其扩展到更强大的预测器可能会进一步提高性能。

6.8 Writing Tips from This Paper

(1) Herein (在本文中), a stacked autoencoder (SAE) model is used to learn generic traffic flow features, and it is trained in a layerwise greedy fashion. (in a ... fashion, 以一种什么样的方式)

(2) To the best of the authors' knowledge, it is the first time that the SAE approach is used to represent traffic flow features for prediction. (引出自己的创新点用, 深深的套路)

(3) The remainder/rest of this study/paper is organized as follows.

7 ConvLSTM:从短时降水量预测中学习 ConvLSTM

7.1 文章信息

《Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting》。

这篇文章是香港科技大学计算机学院施行健博四的时候（2015 年）发表在计算机领域的一篇会议论文（会议名称：Advances in Neural Information Processing Systems 28 (NIPS 2015)），至今被引 800 多次，这篇文章也是在国际上首次提出了卷积神经网络 CNN 和循环神经网络 LSTM 的结合体-ConvLSTM。经人肉该作者，其目前在美国亚马逊工作。

Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting

Part of: [Advances in Neural Information Processing Systems 28 \(NIPS 2015\)](#)

A note about reviews: "heavy" review comments were provided by reviewers in the program committee as part of the evaluation process for NIPS 2015, along with posted responses during the author feedback period. Numerical scores from both "heavy" and "light" reviewers are not provided in the review link below.

[\[PDF\]](#) [\[BibTeX\]](#) [\[Supplemental\]](#) [\[Reviews\]](#)

Authors

- [Xingjian SHI](#)
- [Zhouong Chen](#)
- [Hao Wang](#)
- [Dit-Yan Yeung](#)
- [Wai-kin Wong](#)

7.2 摘要

短时降水预报的目的是在较短的时间内预测局部地区未来的降水强度。以前很少有研究从机器学习的角度来研究这个至关重要且具有挑战性的天气预报问题。本文将降水临近预报问题描述为一个时空序列预测问题，其中输入和预测目标都是时空序列。通过对全连接 LSTM (fully connected LSTM, FC-LSTM) 进行扩展，使其在输入到状态和状态到状态转换中都具有卷积结构，我们提出了卷积 LSTM (ConvLSTM)，并利用它建立了短时降水预报模型。实验表明，ConvLSTM 网络能够更好地捕捉时空相关性，并始终优于 FC-LSTM。

7.3 简介和文献综述

文章主要目标是在相对较短的时间(例如，0-6 小时)内准确和及时地预测局部地区的降雨强度，即利用以前观测到的雷达回波序列预测局部区域未来固定长度的雷达图。

短时降水预报实质上是一个以过去雷达图序列为输入，以未来雷达图的一个固定数字序列(通常大于 1)为输出的时空序列预报问题。然而，由于时空序列的高维

性，特别是当需要进行多步预测时，这种学习问题注定很复杂，除非数据的时空结构被预测模型很好地捕获。此外，由于大气的混沌特性，建立有效的雷达回波数据预测模型更具挑战性。

本文提出了一种用于降水临近预报的卷积 LSTM (ConvLSTM) 网络。我们将降水临近预报问题定义为一个时空序列预测问题，为了更好地模拟时空关系，我们将 FC-LSTM 的思想推广到卷积结构的 ConvLSTM，它在输入到状态和状态到状态的转换中都具有卷积结构。通过叠加多个 ConvLSTM 层，形成一个编码预测结构，建立短时降水预报模型。为了进行评估，我们创建了一个新的现实生活中的雷达回波数据集。

7.4 创新点

- (1) 首次提出 ConvLSTM，能捕捉时空特征而非单纯时序特征。
- (2) 预测的是多步时空预测而非单步时空预测。

7.5 主体内容

在实际应用中，雷达图通常每 6-10 分钟从天气雷达上获取一次，在接下来的 1-6 小时内进行短时预报，即，预测接下来的 6-60 帧。从机器学习的角度来看，这个问题可以看作是一个时空序列预测问题。

假设我们有一个由 M 行 N 列组成的 $M \times N$ 网格表示的空间区域上观察一个动力系统。在网格中的每个单元格中，有 P 个测量值随时间而变化。因此，任何时刻的观测都可以用张量 $\mathbf{X} \in R^{P \times M \times N}$ 表示，其中 R 表示观测特征的域。时空序列预测问题是预测未来 k 个序列，该序列由之前的 J 次观测(包括当前的 J 次观测)得到：

$$\tilde{\mathcal{X}}_{t+1}, \dots, \tilde{\mathcal{X}}_{t+K} = \arg \max_{\mathcal{X}_{t+1}, \dots, \mathcal{X}_{t+K}} p(\mathcal{X}_{t+1}, \dots, \mathcal{X}_{t+K} \mid \hat{\mathcal{X}}_{t-J+1}, \hat{\mathcal{X}}_{t-J+2}, \dots, \hat{\mathcal{X}}_t) \quad (1)$$

对于短时降水预报，每个 timesteps 观测到的是二维 2D 雷达回波图。如果我们将地图分割成平铺的不重叠的 patch，并将 patch 内的像素作为其测量值(如图所

示), 那么短时预测问题自然就变成了一个时空序列预测问题。我们的时空序列预测问题不同于单步时间序列预测问题, 因为我们的预测目标是一个同时包含时空结构的序列。虽然预测长度为 K 个序列中的自由变量数可以达到 $O(M^K N^K P^K)$, 但在实践中, 我们可以利用可能预测空间的结构来降低维数, 从而使问题易于处理。

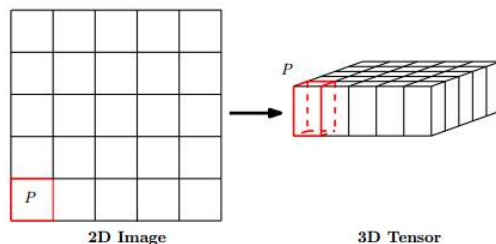


Figure 1: Transforming 2D image into 3D tensor

$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i)$	$i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i)$
$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f)$	$f_t = \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f)$
$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$	$\mathcal{C}_t = f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c)$
$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o)$	$o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o)$
$h_t = o_t \circ \tanh(c_t)$	$\mathcal{H}_t = o_t \circ \tanh(\mathcal{C}_t)$

LSTM 公式

ConvLSTM 公式

两者的不同点是传统 LSTM 是 1D 张量, ConvLSTM 的输入是 3D 张量。实际上, 传统 FC-LSTM 的输入、输出和隐藏状态也可以看作是后二维为 1 的三维张量。从这个意义上说, FC-LSTM 实际上是 ConvLSTM 的一个特例, 它的所有特性都位于一个单元上。

我们使用下图所示的结构, 它由两个网络组成, 一个编码网络和一个预测网络。预测网络的初始状态和输出是从编码网络的最后状态复制而来的。这两种网络都是通过叠加几个 ConvLSTM 层而形成的。由于我们的预测目标具有与输入相同的维数, 我们将预测网络中的所有状态连接起来, 并将它们输入到一个 1×1 卷积层中生成最终的预测。

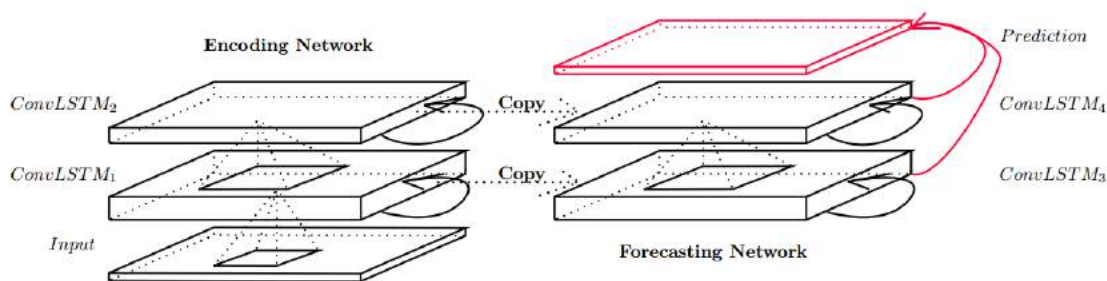


Figure 3: Encoding-forecasting ConvLSTM network for precipitation nowcasting

这种结构也类似于 LSTM 预测模型，只是我们的输入和输出都是保存所有空间信息的三维张量。由于该网络具有多层叠加的 ConvLSTM 层，因此具有较强的表示能力，适用于短时降水预报等复杂动力系统的预报。

7.6 Experiment

本文使用的雷达回波数据集是 2011 年至 2013 年在香港收集的三年天气雷达强度的子集。因为不是每天都下雨，我们的短时预报目标是降水，所以选择前 97 个雨天来形成数据集。天气雷达数据每 6 分钟记录一次，每天有 240 帧。为了得到用于训练、测试和验证的不相交子集，我们将每天的序列划分为 40 个不重叠的帧块，并随机分配 4 个训练块、1 个测试块和 1 个验证块。数据实例使用一个 20 帧宽的滑动窗口从这些块中分割出来。因此，我们的雷达回波数据集包含 8148 个训练序列，测试序列为 2037，验证序列为 2037，序列长度均为 20 帧(输入 5 帧，预测 15 帧)。(所以每个训练块可以得到 21 个训练窗口，一天有 4 个训练块，97 天，共 8148 个训练窗口，即 8148 个 sequence) 训练时的误差函数为 15 个预测的交叉熵误差。预测评价标准之一为 rainfall mean squared error (Rainfall-MSE)。

FC-LSTM 网络的性能不太好，这主要是由于雷达图具有很强的空间相关性，即，在局部地区，云的运动是高度一致的。全连接结构有太多冗余连接，使得优化不太可能捕捉到这些局部一致性。结果与 FC-LSTM 进行对比，结果毫无疑问 ConvLSTM 预测效果更好。

7.7 结论和展望

在本文中，我们成功地将机器学习方法，尤其是深度学习方法，应用到具有挑战性的短时降水预报问题中。将降水临近预报问题定义为一个时空序列预报问题，提出了一种新的 LSTM 扩展方法 ConvLSTM。ConvLSTM 层不仅保留了 FC-LSTM 的优点，而且由于其固有的卷积结构，也适用于时空数据。将 ConvLSTM 模型引入编码预测结构，建立了短时降水预报模型。

在未来的工作中，我们将研究如何将 ConvLSTM 应用到基于视频的动作识别中。其中一个想法是在卷积神经网络生成的空间特征图上添加 ConvLSTM，利用 ConvLSTM 的隐藏状态进行最终分类。

8 Sequence to Sequence: 从 NLP 自然语言处理到工程领域中的时间序列

NLP 自然语言处理中的单词，句子，经过各种处理编码，电脑识别到的还是一串数字，即一个有前后关系的时间序列。放到交通工程、土木工程领域，时间、速度、客流量、轨道形变量、桥梁变形位移等各类时间序列本来就是数值，相比于 NLP，还省去了编码的过程呢，哈哈哈！所以做个大自然的搬运工可不省事？！！！计算机小白的我，还是从 NLP 领域大火的 Seq2Seq 模型开始蜗牛吧.....

8.1 文章信息

《Sequence to Sequence Learning with Neural Networks》。

这篇文章是 2014 年 Google Brain 团队的三位老外发的一篇文章，首次真正提出了 Seq2Seq 模型,被引 6000 多次。其实还有另外一篇文章《Learning Phrase Representations using RNN Encoder - Decoder for Statistical Machine Translation》也是在 2014 年使用了 Seq2Seq 模型来解决机器翻译的问题，被引 5000 多次，这边文章比前者早了几个月，两篇文章的区别是前者是来自于谷歌的论文，工程应用性更强，后者来自蒙特利尔大学的一个博士生，学术性更强，后面这篇文章有空再看。

Sequence to Sequence Learning with Neural Networks

Part of: [Advances in Neural Information Processing Systems 27 \(NIPS 2014\)](#)

[\[PDF\]](#) [\[BibTeX\]](#) [\[Reviews\]](#)

Authors

- [Ilya Sutskever](#)
- [Oriol Vinyals](#)
- [Quoc V. Le](#)

Conference Event Type: Oral

8.2 摘要

本文提出了一种 end-to end 序列学习方法，使用多层 LSTM 将输入序列映射到一个固定维数的向量上（编码），然后用另一个深层 LSTM 从向量上解码目标序列。将该方法应用在 WMT-14 数据集的英法翻译任务中。此外，LSTM 擅长处理长句子（得益于 LSTM 处理 long-term dependency 的良好性能，交通领域即能记忆很久以前的交通状态，从而应用到当前状态的预测中）。LSTM 还学习了对词序敏感、对主动语态和被动语态相对不敏感的短语和句子表示。最后，我们发现对所有输入序列(不是目标序列)做逆序处理后显著地提高了 LSTM 的性能，因为这样做会在输入序列和目标序列之间引入许多短期依赖关系，从而使优化问题变得更加容易。

8.3 简介

深度神经网络(DNNs)是一种功能极其强大的机器学习模型，在语音识别和视觉对象识别等困难问题上表现优异。DNNs 之所以功能强大是因为它们可以用很少的步骤执行任意并行计算。虽然神经网络与传统的统计模型有关，但它们能够学习复杂的计算。此外，只要有足够的有标记的训练集来确定网络参数，就可以使用有监督的反向传播算法对 DNNs 进行训练。

尽管 DNNs 具有灵活性和强大的功能，但其输入和输出只能用固定维数的向量来编码。这是很大限制，因为许多重要的问题最好用长度未知的序列来表示。例如，语音识别和机器翻译（输入输出不可能长度相等）。因此，一个学习将序列映射到序列的不受输入输出维数限制的方法显然是有用的。

DNNs 要求输入和输出的维数是已知的和固定的。（对于 RNN，只要预先知道输入和输出之间的固定长度，RNN 就可以轻松地将序列映射到序列。然而，如何将 RNN 应用于输入和输出序列长度不同且关系复杂且非单调的问题，目前尚不清楚）。

本文展示了一个直观的长短期记忆(LSTM)结构解决一般序列对序列的问题（如下图）。其思想是使用一个 LSTM 来读取输入序列（每次一个时间步长）通过编码获得固定维度的向量，然后使用另一个 LSTM 进行解码，即从该向量中提取输出序列。第二个 LSTM 本质上是一个递归神经网络语言模型，但它的输入是编码得到的固定维度的向量。LSTM 能够成功地学习具有长期时间依赖关系的数据，所以本文中的模型也自然是基于 LSTM。

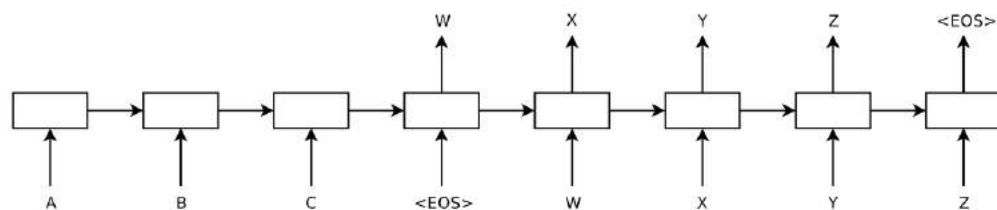


Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

简单理解一下就是输入序列为 ABC 以及输入序列结束符号<EOS>，从<EOS>开始解码出 WXYZ 以及结束符号<EOS>，停止解码。多说一句该模型架构在翻译任务上，均取得了不错的效果，且还有巨大的提升空间（比如引入 Attention 注意力机制）

该模型在长句子上表现很好,因为我们颠倒了训练集和测试集中输入序列中单词的顺序（只颠倒输入序列，不颠倒输出序列）。这样做,我们引入了许多短期依

赖关系从而使优化问题变得更简单。

LSTM 的一个特性是它能将一个可变长度的输入语句映射成一个固定维度的向量表示。考虑到翻译往往是原句段落的释义，翻译目标鼓励 LSTM 找到能够表示其含义的句子表示形式，因为具有含义相近的句子彼此接近，而含义不同的句子会距离很远。结果表明我们的模型能够识别词序，并且对主动语态和被动语态也不敏感。

8.4 创新点

(1) 首先我们使用了两种不同的 LSTM:一种用于输入序列，另一种用于输出序列，这样虽然增加了模型参数，但计算成本的增加可以忽略不计，并且可以提高模型的泛化能力。

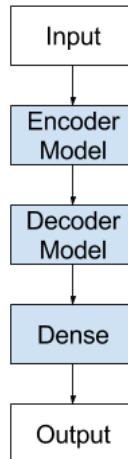
(2) 其次，我们发现深度 LSTM 明显优于浅层 LSTM，因此我们选择了一个 4 层的 LSTM。

(3) 第三，我们发现颠倒输入句子的单词顺序非常有价值。例如,不是映射将 a, b, c 映射到 α, β, γ , 而是将 c, b, a 映射到 α, β, γ 。其中 α, β, γ 是翻译后的 a, b, c 。这样, a 就离 α 很近, b 就离 β 很近, 依此类推。(正如上面所提到的, 一个段落中有含义相近的句子彼此接近, 而含义不同的句子会距离很远)

8.5 主体内容

只是学习建模思想，所以 NLP 领域的专业知识和建模过程就略过了。

模型的实现上，其实 Seq2Seq 模型就是一个使用了 LSTM 层 Encoder-Decoder 自编码器模型，Keras 中也比较适合新手实现。



类似于这样的.....

```

from keras.models import Sequential
from keras.layers.recurrent import LSTM
from keras.layers.wrappers import TimeDistributed
from keras.layers.core import Dense, RepeatVector

def build_model(input_size, max_out_seq_len, hidden_size):
    model = Sequential()
    model.add(LSTM(input_dim=input_size, output_dim=hidden_size, return_sequences=False))
    model.add(Dense(hidden_size, activation="relu"))
    model.add(RepeatVector(max_out_seq_len))
    model.add(LSTM(hidden_size, return_sequences=True))
    model.add(TimeDistributed(Dense(output_dim=input_size, activation="linear")))
    model.compile(loss="mse", optimizer='adam')

    return model
  
```

对上述代码作一点解释：

1. Encoder(即第一个 LSTM) 只在序列结束时输出一个语义向量, 所以其 "return_sequences" 参数设置为 "False"
2. Decoder(即第二个 LSTM) 需要在每一个 time step 都输出, 所以其 "return_sequences" 参数设置为 "True"
3. 使用 "RepeatVector" 将 Encoder 的输出(最后一个 time step)复制 N 份作为 Decoder 的 N 次输入
4. TimeDistributed 是为了保证 Dense 和 Decoder 之间的一致, 可以不用太关心

或者这样的.....

```

# define encoder
visible = Input(shape=(n_in,1))
encoder = LSTM(100, activation='relu')(visible)

# define reconstruct decoder
decoder1 = RepeatVector(n_in)(encoder)
decoder1 = LSTM(100, activation='relu', return_sequences=True)(decoder1)
decoder1 = TimeDistributed(Dense(1))(decoder1)

# define predict decoder
decoder2 = RepeatVector(n_out)(encoder)
decoder2 = LSTM(100, activation='relu', return_sequences=True)(decoder2)
decoder2 = TimeDistributed(Dense(1))(decoder2)

# tie it together
model = Model(inputs=visible, outputs=[decoder1, decoder2])
model.compile(optimizer='adam', loss='mse')
  
```

8.6 特别之处

将输入进行颠倒逆序：

文章说虽然我们对这一现象没有一个完整的解释，但我们认为这是由于对数据集引入了许多短期依赖关系造成的。

但是又强行解释到：通过颠倒输入序列中的单词，保持输入序列中对应单词与输出序列的平均距离不变。然而，输入序列中的前几个单词现在与输出序列中的前几个单词非常接近，因此问题的最小 time lag 大大减少，而句子末尾的词距离变长的代价似乎并不显著。因此，反向传播更容易在输入序列和输出序列之间“建立通信”，从而大大提高了整体性能。

8.7 展望

土木工程和交通工程领域的时间序列处理问题上，或许可以探索借鉴本文的建模思想，以得到更好的结果。

9 ConvLSTM& Bi-LSTM：基于深度学习的短时道路交通流预测

9.1 文章信息

《Short-Term Traffic Flow Prediction with Conv-LSTM》。

这篇文章是一篇会议论文，2017 第九届无线通信与信号处理国际会议 (WCSP)，福州大学物理与信息工程学院的几位老师，被引 10 次。



9.2 主体内容

本文提出了一种基于深度学习的短时交通流预测方法。交通流数据包含三个主要特征:时间特征、空间特征和周期性特征。我们把 CNN 和 LSTM 结合起来生成一个 ConvLSTM 模块,用于提取交通流的时空特征,然后使用 Bi-LSTM(双向 LSTM, Keras 中有相应模块)提取交通流的周期特征。

```
model = Sequential()
model.add(Bidirectional(LSTM(10, return_sequences=True),
                        input_shape=(5, 10)))
model.add(Bidirectional(LSTM(10)))
model.add(Dense(5))
model.add(Activation('softmax'))
model.compile(loss='categorical_crossentropy', optimizer='rmsprop')
```

利用 ConvLSTM 模块对相邻区域的短时交通流数据进行处理,提取时空特征;利用双向 LSTM 对预测点历史交通数据进行处理,提取交通流数据的周期特征。提出了一种无需数据预处理和数据特征提取的端到端深度学习短时交通流预测体系结构。最后,集中时空特征和周期特征对交通流进行预测。

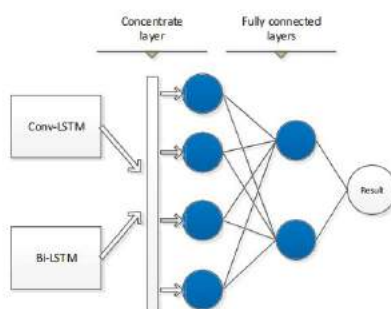


Fig. 1. The proposed deep architecture model for short-term traffic flow prediction

9.3 创新点

既使用了 ConvLSTM, 又使用了 Bi-LSTM, 结构新颖。

9.4 大家比较关注的算法实现

我们将交通流数据映射到一维向量。我们将预测点的交通流数据放入向量中

心，根据与预测点的距离，将其他点的交通流数据依距离放在该向量中心（预测点）两侧。

将不同时刻的一维空间信息向量组合成矩阵如下：

$$\mathbf{F} = \begin{bmatrix} F_1^1 & F_1^2 & \dots & F_1^s \\ F_2^1 & F_2^2 & \dots & F_2^s \\ \vdots & \vdots & \ddots & \vdots \\ F_t^1 & F_t^2 & \dots & F_t^s \end{bmatrix} \quad (3.1)$$

其中 s 表示预测点（一列代表一个预测点）， t 表示时间。

周期数据可以表示为以下矩阵：

$$\mathbf{D} = \begin{bmatrix} F_1^1 & F_1^2 & \dots & F_1^s \\ F_2^1 & F_2^2 & \dots & F_2^s \\ \vdots & \vdots & \ddots & \vdots \\ F_d^1 & F_d^2 & \dots & F_d^s \end{bmatrix} \quad (3.2)$$

and

$$\mathbf{W} = \begin{bmatrix} F_1^1 & F_1^2 & \dots & F_1^s \\ F_2^1 & F_2^2 & \dots & F_2^s \\ \vdots & \vdots & \ddots & \vdots \\ F_w^1 & F_w^2 & \dots & F_w^s \end{bmatrix} \quad (3.3)$$

其中 d 代表昨天的相同时刻， w 代表上周的相同时刻。

上述数据矩阵即为本文提出的深度神经网络结构的输入数据。其中，损失函数为 MSE，优化器为 RMSprop。

9.5 ConvLSTM 的结构

ConvLSTM 的输入数据既为公式 3.1。一行作为一个向量，代表着一个 time step 所有预测点之间的空间信息，我们使用一维 Conv（Conv1D）对公式 3.1 中的每一行进行处理，自然也是利用一维卷积核滤波器（kernal size=1），通过滑动滤波器获取每个 time step 局部感知域的卷积信息(空间信息)。然后，将局部特征聚合成全局特征。

$$G(i) = F(Aw + B) \quad (4)$$

where w is the filter weights of the node, B is bias, A is the value of the input node and F is the activation function.

紧接着加池化层，不同之处在于池化滤波器不进行复杂的卷积运算。本文中使用一维平均池化层(AveragePooling1D)。通过池化将生成的特征序列 C 缩减到原维度的一半大小。这两种特征提取使得深度神经网络在处理交通流数据时具有更高的失真容忍度。

分别对时间序列向量(式 3.1 中的每一行)的各个元素进行卷积和池化处理后，输出结果变为时间序列向量 $C_t = (C_1, C_2, C_3, \dots, C_t)$ 。向量中的每个元素都是区域内各点间交通流的空间相关性。（该表达式中每个元素代表着一个行向量，行向量的维度比式 3.1 中行向量的维度要小，因为经过了池化）

上面得到的时间序列向量 C_t 即为 LSTM 的输入数据。

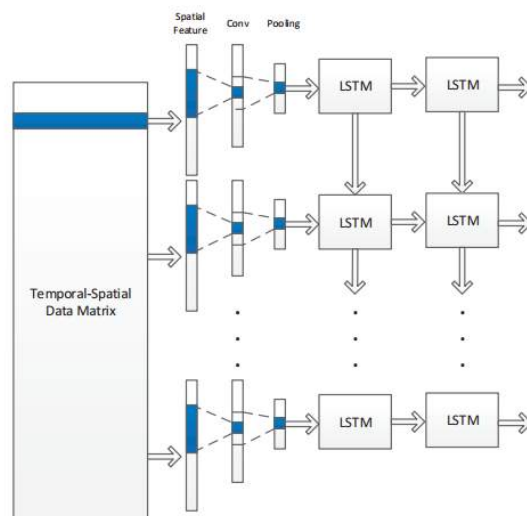


Fig. 3. The Conv-LSTM Structure

9.6 双向 LSTM 的结构 (Bi-directional LSTM)

交通流 also 具有很强的周期性特征。本文将增加交通流的周期性特征作为补充信息来预测短期交通流。在提取交通流数据的周期性特征时，我们会同时处理前一天的同一时间和上一周的同一时间的交通流信息。在处理了历史信息后，得到了全时间序列数据（我想应该是将上一周的数据和前一天的数据放到预测当天数据的前面，组成全时间序列数据以获取周期性信息。）。

双向 LSTM 的结构由上下堆叠的两个单向 LSTM 组成。因此，Bi-LSTM 输入

包含预测时间前后的时间序列，在每个 T 时刻，输入序列被输入给两个方向相反 LSTM，输出由两个 LSTM 决定。每个反向 LSTM 的误差传播与正向 LSTM 传播算法相同。Bi-LSTM 结构如下图所示，其中 x_i 为 LSTM 的输入， O_f 为正向 LSTM 的输出， O_b 表示反向 LSTM 的输出。

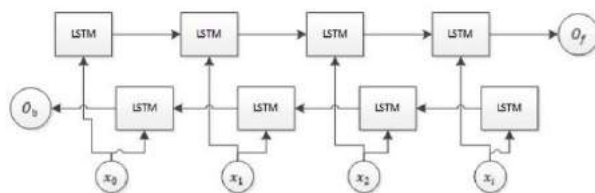


Fig. 4. The Bi-LSTM Structure

简单来说，就是在 Keras 中将 LSTM 换成 Bidirectional(LSTM)即可。一些细节信息例如使用了多少个 timestep, 历史数据如何处理输入到 Bi-LSTM 中，池化前后的维度变化等，文章交代的并不清楚，毕竟会议论文。

10 CNN：使用卷积神经网络做交通速度预测

10.1 文章信息

《Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction》。

北航马晓磊老师 2017 年发在开源期刊 Sensor 上的一篇文章（大类工程技术 3 区，IF:2.475），被引 168 次。

10.2 摘要

本文提出了一种基于卷积神经网络(CNN)的交通速度预测方法，该方法将交通网络作为图像进行学习，并对大规模、全网范围内的交通速度进行预测。利用二维时空矩阵将时空交通状态转化为描述交通流时空关系的图像。文章主体分为两步：利用 CNN 提取交通特征和网络范围的交通速度预测。以北京二环路和北京东北地区交通网络为例进行研究，并将该方法与常用的四种算法(普通最小二乘 OLS、k

近邻 KNN、人工神经网络 ANN 和随机森林 RW)和三种深度学习结构(堆叠式自动编码器 SAE、递归神经网络 RNN 和长短时记忆网络 LSTM)进行比较。结果表明,在可接受的执行时间内,该方法的平均精度比其他算法提高了 42.91%。CNN 可以在合理的时间内对模型进行训练,适用于大型交通网络。

10.3 简介和文献综述

意义+目前面临的挑战+目前使用方法分类+每一类别的综述+既有研究不足+自己做的内容+罗列本文贡献。

10.4 理论部分

本文使用的是出租车轨迹数据,利用轨迹数据获取某个路段某个时间段(本文 2 分钟)的平均交通速度,并整理成如下 M 矩阵形式。

Table 2. Example of taxi GPS data.

Taxi ID	Positioning Time	Latitude	Longitude	Speed (m/s)	Passenger State	Description
1379	2014/6/18 11:50:00	39.91004	116.30612	11.83	268435456	closed door, effective location, occupied.
1379	2014/6/18 11:51:01	39.9099	116.31544	9.26	268435456	closed door, effective location, occupied.
1379	2014/6/18 11:52:00	39.9097	116.32033	0	268435456	closed door, effective location, occupied.
1379	2014/6/18 11:53:00	39.90942	116.32373	9.77	268435456	closed door, effective location, occupied.
1379	2014/6/18 11:54:00	39.9096	116.3285	3.08	268435456	closed door, effective location, occupied.
1380	2014/6/18 5:38:20	39.91548	116.18801	4.11	0	closed door, effective location, vacant.
1380	2014/6/18 5:39:21	39.91453	116.18817	2.05	0	closed door, effective location, vacant.
1380	2014/6/18 5:40:20	39.9143	116.19248	4.63	0	closed door, effective location, vacant.
1380	2014/6/18 5:41:19	39.91246	116.19693	8.74	0	closed door, effective location, vacant.

In the passenger state column, "268435456" means there is at least one passenger in the taxi and "0" means the taxi is vacant.

$$M = \begin{bmatrix} m_{11}, m_{12}, \dots, m_{1N} \\ m_{21}, m_{22}, \dots, m_{2N} \\ \vdots \\ m_{Q1}, m_{Q2}, \dots, m_{QN} \end{bmatrix}$$

N 为时间间隔的长度, Q 为路段的长度, 像素 m_{ij} 是第 i 路段在 j 时刻的平均交通速度。设 x 轴和 y 轴分别表示矩阵的时间和空间。生成的矩阵可以看作是图像的一个通道 channel, 因此, 图像的宽度为 M 像素, 高度为 N 像素。

交通矩阵提取示意图如下:

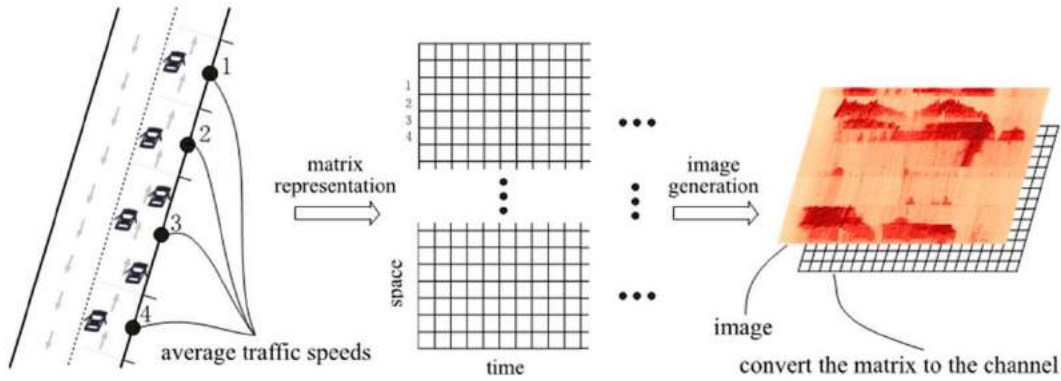


Figure 1. An illustration of the traffic-to-image conversion on a network.

CNN 用作图片识别和交通预测的区别:

首先，模型输入是不同的，即，输入图像只有一个通道值，该通道值由交通网络中所有道路的交通速度决定，图像像素值范围为 0 到网络的最大交通速度或速度上限。而在图像分类问题中，输入图像一般有三个通道，即 RGB 和像素值的范围从 0 到 255。虽然存在差异，为了防止模型权重增加模型训练难度，要对模型输入进行了规范化。

其次，模型输出是不同的。在交通环境下，模型输出是预测交通网络各路段的交通速度，而在图像分类问题中，模型输出是图像类标签。

第三，抽象特征有不同的含义。在交通背景下，卷积层和池化层提取的抽象特征是路段之间关于交通速度的关系。在图像分类问题中，根据训练目标，抽象特征可以是图像的浅层边缘和某些对象的深层形状。

第四，目标函数因模型输出的不同而不同。在交通方面，由于输出是连续的交通速度，因此应该采用连续的目标函数（MSE）。在图像分类问题中，通常使用交叉熵代价函数 cross entropy。

文章的网络架构如下：

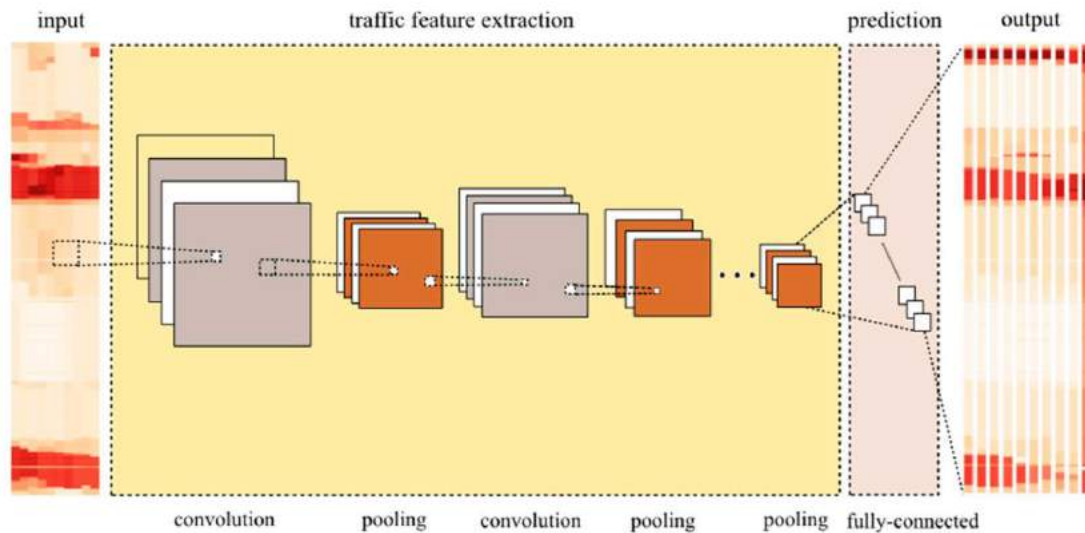


Figure 2. Deep learning architecture of CNN in the context of transportation.

10.5 案例研究将在下篇文章中叙述

10.6 Writing Tips from This Paper

(1) Recently, a wealth of literature leverage multiple hybrid models and spatiotemporal features to improve traffic prediction performance. (leverage: 利用)

(2) In summary, statistical methods have been widely used in traffic prediction, and promising results have been demonstrated. However, these models ignore the important spatiotemporal feature of transportation networks, and cannot be applied to predict overall traffic in a large-scale network. (In summary, ... However, ... 总结+转折 引出自己的创新点)

(3) Spatiotemporal features of network traffic can be extracted using a CNN in an automatic manner with a high prediction accuracy. (in an automatic manner 自动的)

11 CNN: 使用卷积神经网络做交通速度预测-案例简介

此篇文章主要介绍上一篇论文的案例研究部分，背景和理论部分请看上一篇文章。

11.1 文章信息

《Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction》。

北航马晓磊老师 2017 年发在开源期刊 Sensor 上的一篇文章（大类工程技术 3 区，IF:2.475），被引 168 次。

11.2 案例研究

从上篇文章得知，本文的输入为：

$$M = \begin{bmatrix} m_{11}, m_{12}, \dots, m_{1N} \\ m_{21}, m_{22}, \dots, m_{2N} \\ \vdots \quad \vdots \quad \dots \quad \vdots \\ m_{Q1}, m_{Q2}, \dots, m_{QN} \end{bmatrix}$$

N 为时间间隔的长度，Q 为路段的长度，像素 m_{ij} 是第 i 路段在 j 时刻的平均交通速度。设 x 轴和 y 轴分别表示矩阵的时间和空间。生成的矩阵可以看作是图像的一个通道 channel，因此，图像的宽度为 M 像素，高度为 N 像素。

实验数据：北京市 2015 年 5 月 1 号到 6 月 6 号出租车 GPS 数据，定位间隔约为 1 分钟，利用该轨迹数据获取某个路段某个时间段（本文 2 分钟）的平均交通速度得到上述矩阵。

本文选取了两个交通网络，一个是二环路（单向，236 个路段），一个是东北部的一个子网络（部分双向，352 个路段）。进行了 4 次实验，task1-task4.

Task 1: 使用过去 30 分钟的交通速度预测未来 10 分钟的；

Task 2: 使用过去 40 分钟的交通速度预测未来 10 分钟的；

Task 3: 使用过去 30 分钟的交通速度预测未来 20 分钟的；

Task 4: 使用过去 40 分钟的交通速度预测未来 10 分钟的

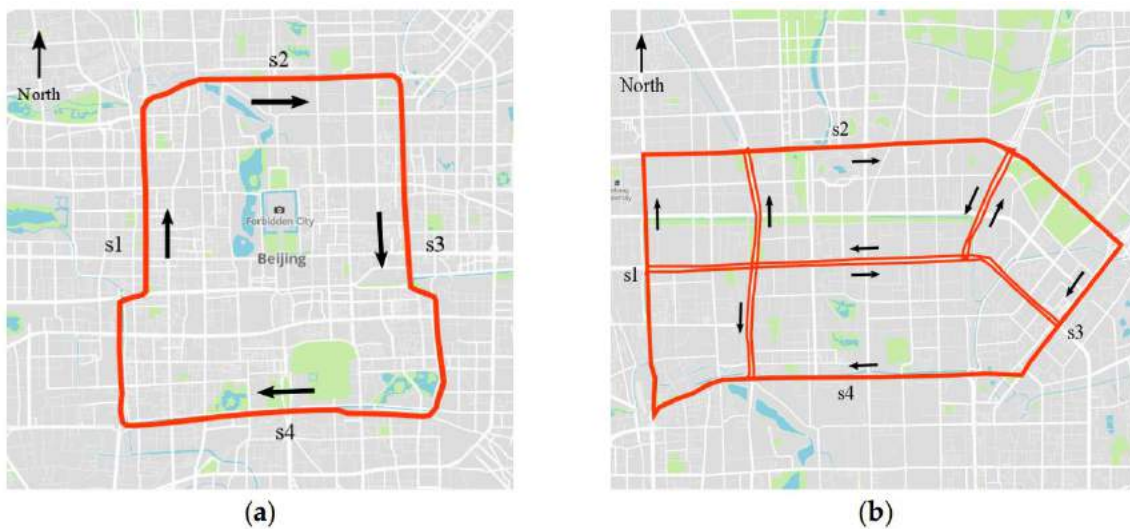
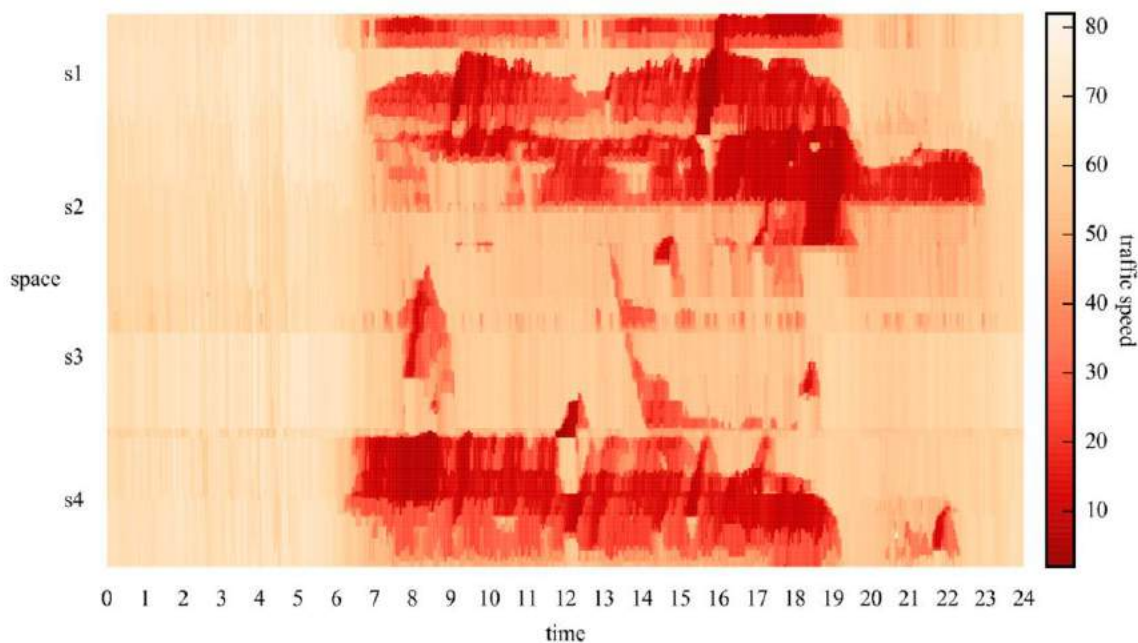


Figure 3. Two sub-transportation networks for testing: (a) Network 1, the second ring of Beijing; and (b) Network 2, a network in Northeast Beijing

为了获取交通速度矩阵，第一个网络直接拉直即可，第二个是把网络切割成一段一段的交通直线然后按顺序放置。

一天的数据处理成一张图片后共 37 张图片，如下图，横轴为时间，纵轴为各个路段。越红代表速度越低越拥挤。



CNN 参数：

Filter size=(3, 3), Pooling size=(2, 2), filter 分别为 256, 128, 64 个; 训练集为前 30 天的 21600 个样本 (2 分钟一个样本, 一个小时 30 个, 一天 24*30=720 个, 30 天 720*30=21600 个), 测试集为后 7 天 5040 个样本。

Table 2. Hyperparameters of the CNN.

Layer	Name	Parameters	Dimensions	Parameter Scale
Input	—	—	(1, 236, 20)	—
Layer 1	Convolution	Filter (256, 3, 3)	(256, 236, 20)	2304
Layer 1	Pooling	Pooling (2, 2)	(256, 118, 10)	0
Layer 2	Convolution	Filter (128, 3, 3)	(128, 118, 10)	1152
Layer 2	Pooling	Pooling (2, 2)	(128, 59, 5)	0
Layer 3	Convolution	Filter (64, 3, 3)	(64, 59, 5)	576
Layer 3	Pooling	Pooling (2, 2)	(64, 30, 3)	0
Layer 4	Data flatten	—	(5760,)	0
Layer 4	Fully-connected	—	(1180,)	6,796,800
Output	—	—	(1180,)	—

预测结果:

Table 3. Prediction performance (MSE) of the CNN and other algorithms.

Study Network	Model	MSE of Different Models (on Test Datasets)			
		Task 1	Task 2	Task 3	Task 4
Network 1	CNN	22.825 *	24.345 *	30.593 *	31.424 *
	OLS	27.047	31.273	41.334	48.107
	KNN	51.700	55.708	60.256	64.132
	RF	35.092	35.431	40.476	40.638
	ANN	67.764	52.339	58.797	57.225
	SAE	60.751	69.082	65.292	68.326
	RNN	33.408	36.833	40.551	39.038
Network 2	LSTM NN	37.759	33.218	42.909	42.865
	CNN	27.163 *	28.479 *	37.987 *	38.816 *
	OLS	33.741	41.657	50.123	62.282
	KNN	69.965	74.863	79.367	83.881
	RF	48.603	48.946	52.676	53.067
	ANN	124.937	147.489	133.299	168.136
	SAE	85.079	94.982	82.271	99.020
	RNN	48.877	47.470	52.577	52.114
	LSTM NN	43.304	45.657	50.928	48.345

Note: * indicates the best result.

11.3 展望

文章一直在强调两点:

1, CNN 不同于普通的全连接层, 由于滤波器 filter 的存在, CNN 能够提取局部特征。文章使用了数百个 filter, 一个 filter 可以能够提取 1 个交通特征, 因此,

对一个输入层，数百个 filter 就可以提取数百个交通特征。

2，池化层的设计目的是缩小采样和聚合数据，因为它们只从特定区域提取显著的数字。池化层保证 CNN 是局部不变的，这意味着不论是否进行特征移动、旋转或缩放 CNN 总是可以从输入中提取相同的特征。

借鉴意义：要反复强调自己的创新点。

12 卷积神经网络如何处理一维时间序列数据？

12.1 概述

许多文章都关注于二维卷积神经网络（2D CNN）的使用，特别是图像识别。而一维卷积神经网络（1D CNNs）只在一定程度上有所涉及，比如在自然语言处理（NLP）中的应用。目前很少有文章能够提供关于如何构造一维卷积神经网络来一些机器学习问题。

12.2 何时应用 1D CNN？

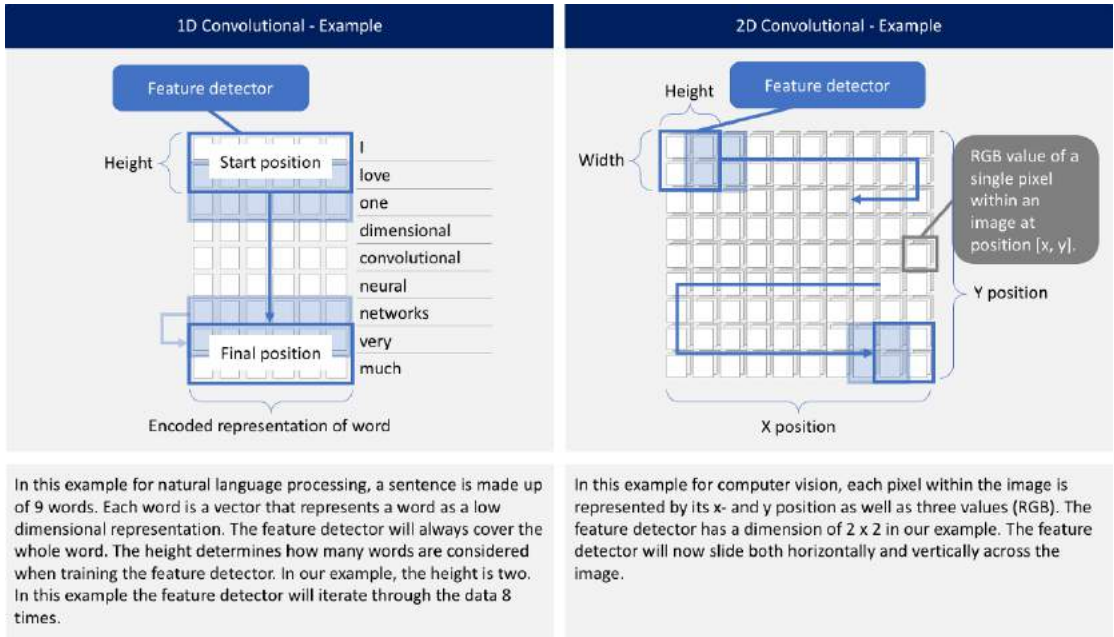
CNN 可以很好地识别出数据中的简单模式，然后使用这些简单模式在更高级的层中生成更复杂的模式。当你希望从整体数据集中较短的（固定长度，即 kernel size）片段中获得感兴趣特征，并且该特性在该数据片段中的位置不具有高度相关性时，1D CNN 是非常有效的。

1D CNN 可以很好地应用于传感器数据的时间序列分析（比如陀螺仪或加速度计数据）；同样也可以很好地用于分析具有固定长度周期的信号数据（比如音频信号）。此外，它还能应用于自然语言处理的任务（由于单词的接近性可能并不总是一个可训练模式的好指标，因此 LSTM 网络在 NLP 中的应用更有前途）。

12.3 1D CNN 和 2D CNN 之间有什么区别？

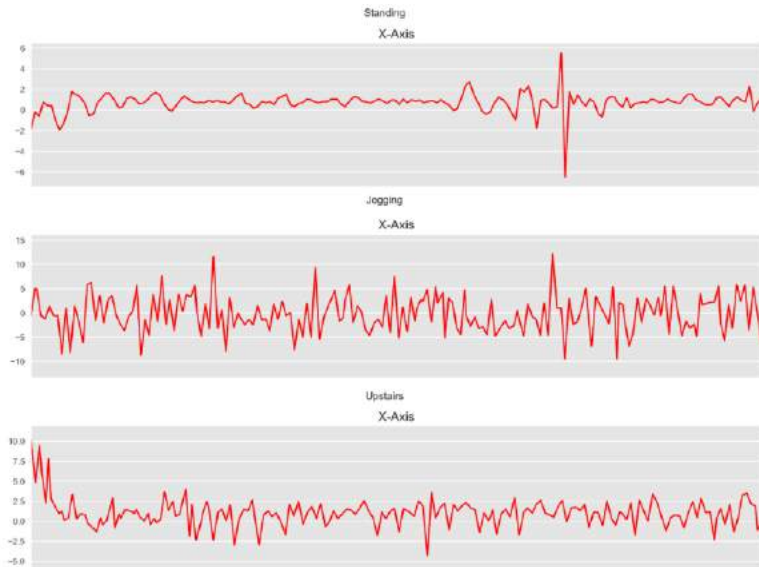
无论是一维、二维还是三维，卷积神经网络（CNNs）都具有相同的特点和相

同的处理方法。关键区别在于输入数据的维数以及特征检测器（或滤波器）如何在数据之间滑动：



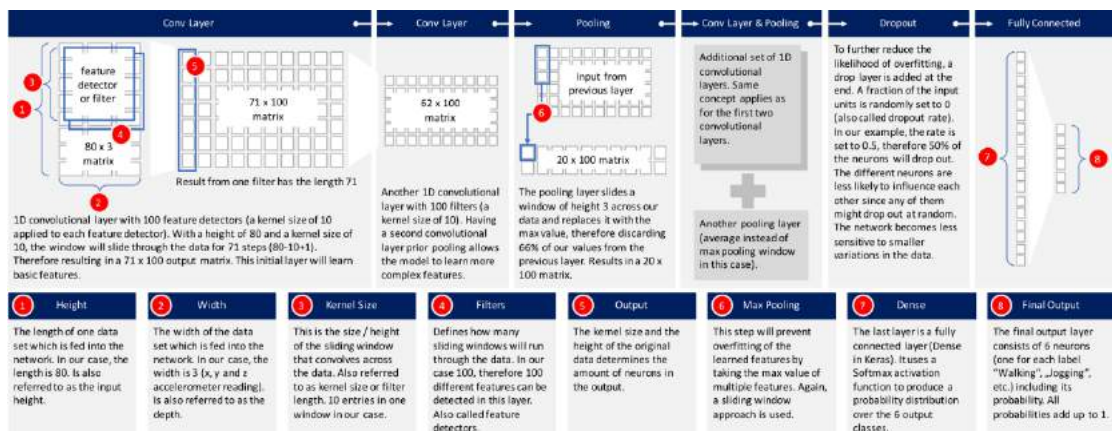
12.4 问题描述

在本文中，我们将专注于基于时间片的加速度传感器数据的处理，这些数据来自于用户的腰带式智能手机设备。基于 x、y 和 z 轴的加速度计数据，1D CNN 用来预测用户正在进行的活动类型（比如“步行”、“慢跑”或“站立”）。



12.5 如何在 Python 中构造一个 1D CNN?

目前已经有许多得标准 CNN 模型可用。我选择了 Keras 网站上描述的一个模型，并对它进行了微调，以适应前面描述的问题。下面的图片对构建的模型进行一个高级概述。其中每一层都将会进一步加以解释。



让我们先来看一下对应的 Python 代码，以便构建这个模型：

```
model_m = Sequential()
model_m.add(Reshape((TIME_PERIODS, num_sensors), input_shape=(input_shape,)))
model_m.add(Conv1D(100, 10, activation='relu', input_shape=(TIME_PERIODS, num_sensors)))
model_m.add(Conv1D(100, 10, activation='relu'))
model_m.add(MaxPooling1D(3))
model_m.add(Conv1D(160, 10, activation='relu'))
model_m.add(Conv1D(160, 10, activation='relu'))
model_m.add(GlobalAveragePooling1D())
model_m.add(Dropout(0.5))
model_m.add(Dense(num_classes, activation='softmax'))
print(model_m.summary())
```

复制代码

让我们深入到每一层中，看看到底发生了什么：

- **输入数据：** 数据经过预处理后，每条数据记录中包含有 80 个时间片（数据是以 20Hz 的采样频率进行记录的，因此每个时间

间隔中就包含有 4 秒的加速度计数据)。在每个时间间隔内, 存储加速度计的 x 轴、y 轴和 z 轴的三个数据。这样就得到了一个 80×3 的矩阵。由于我通常是在 iOS 系统中使用神经网络的, 所以数据必须平展成长度为 240 的向量后传入神经网络中。网络的第一层必须再将其变形为原始的 80×3 的形状。

- 第一个 1D CNN 层: 第一层定义了高度为 10 (也称为卷积核大小) 的滤波器 (也称为特征检测器)。只有定义了一个滤波器, 神经网络才能够在第一层中学习到一个单一的特征。这可能还不够, 因此我们会定义 100 个滤波器。这样我们就在网络的第一层中训练得到 100 个不同的特性。第一个神经网络层的输出是一个 71×100 的矩阵。输出矩阵的每一列都包含一个滤波器的权值。在定义内核大小并考虑输入矩阵长度的情况下, 每个过滤器将包含 71 个权重值。

- 第二个 1D CNN 层: 第一个 CNN 的输出结果将被输入到第二个 CNN 层中。我们将在这个网络层上再次定义 100 个不同的滤波器进行训练。按照与第一层相同的逻辑, 输出矩阵的大小为 62×100 。

- 最大值池化层: 为了减少输出的复杂度和防止数据的过拟合, 在 CNN 层之后经常会使用池化层。在我们的示例中, 我们选择了大小为 3 的池化层。这意味着这个层的输出矩阵的大小只有输入矩阵的三分之一。

- 第三和第四个 1D CNN 层: 为了学习更高层次的特征, 这

里又使用了另外两个 1D CNN 层。这两层之后的输出矩阵是一个 2×160 的矩阵。

- 平均值池化层：多添加一个池化层，以进一步避免过拟合的发生。这次的池化不是取最大值，而是取神经网络中两个权重的平均值。输出矩阵的大小为 1×160 。每个特征检测器在神经网络的这一层中只剩下一个权重。
- Dropout 层：Dropout 层会随机地为网络中的神经元赋值零权重。由于我们选择了 0.5 的比率，则 50% 的神经元将会是零权重的。通过这种操作，网络对数据的微小变化的响应就不那么敏感了。因此，它能够进一步提高对不可见数据处理的准确性。这个层的输出仍然是一个 1×160 的矩阵。
- 使用 Softmax 激活的全连接层：最后一层将会把长度为 160 的向量降为长度为 6 的向量，因为我们有 6 个类别要进行预测（即“慢跑”、“坐下”、“走路”、“站立”、“上楼”、“下楼”）。这里的维度下降是通过另一个矩阵乘法来完成的。Softmax 被用作激活函数。它强制神经网络的所有六个输出值的加和为一。因此，输出值将表示这六个类别中的每个类别出现的概率。

12.6 训练和测试该神经网络

下面是一段用以训练模型的 Python 代码，批大小为 400，其中训练集和验证集的分割比例是 80 比 20。

```

callbacks_list = [
    keras.callbacks.ModelCheckpoint(
        filepath='best_model.{epoch:02d}-{val_loss:.2f}.h5',
        monitor='val_loss', save_best_only=True),
    keras.callbacks.EarlyStopping(monitor='acc', patience=1)
]

model_m.compile(loss='categorical_crossentropy',
                optimizer='adam', metrics=['accuracy'])

BATCH_SIZE = 400
EPOCHS = 50

history = model_m.fit(x_train,
                      y_train,
                      batch_size=BATCH_SIZE,
                      epochs=EPOCHS,
                      callbacks=callbacks_list,
                      validation_split=0.2,
                      verbose=1)

```

该模型在训练数据上的准确率可达 97%。

根据测试集数据进行测试，其准确率为 92%。

原文地址：[Introduction to 1D Convolutional Neural Networks in Keras for Time Sequences](#)

译文出自：[掘金翻译计划](#) 并有简化。

13 考虑动态时空相似性的交通预测深度学习框架

13.1 文章信息

《Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction》

AAAI（美国人工智能协会）2019 大会上的论文。作者: 宾夕法尼亚州立大学的 Huaxiu Yao。

13.2 摘要

如何对复杂的时空依赖关系建模是当前交通预测的一个重点。既有研究大多将空间依赖看作是静态的, 时间依赖看作是严格周期性的。但本文基于两点进行研究: (1)位置之间的空间依赖关系是动态的; (2)时间依赖服从日、周两种模式, 但其动态时间变化不具有严格的周期性。为了解决这两个问题, 我们提出了一种新的时空动态网络(STDN), 其中引入了一种门控机制来学习位置之间的动态相似性, 并设计了一种周期性注意力转移机制来处理长期的周期性时间转移。

13.3 简介

在基于深度学习的交通预测中, 虽然考虑了时空依赖性, 但现有的方法存在两大局限性。

首先, 地点之间的空间依赖仅仅依赖于历史交通流的相似性, 模型学习的是静态空间依赖关系。然而, 位置之间的依赖关系可能会随着时间而改变, 例如, 早晨, 住宅区和商业中心之间的依赖性很强;而在深夜, 这两个地方的关系可能很弱。然而, 既有研究并没有考虑到这种动态依赖关系。

其次, 许多现有的研究忽略了长期周期性依赖的转变。交通数据表现出较强的日、周两种周期性模式。然而, 交通数据并不是严格的周期性的。例如, 工作日的高峰时间通常发生在下午早些时候, 但不同工作日可以从下午 4:30 到 6 点之间变化。既有研究虽然考虑了周期性, 但没有考虑周期性的时间转移。

本文提出的 STDN 是基于时空神经网络, 分别通过局部 CNN 和 LSTM 处理时空信息。提出了一种基于门控的局部 CNN, 利用交通流信息对不同位置之间的动

态相似性进行建模，从而处理空间依赖关系。提出了一种周期性注意力转移制来学习长期周期依赖关系，通过注意力机制捕捉交通序列的长期周期性信息和周期性时间转移信息。此外，使用 LSTM 来处理时间依赖关系。

模型使用的数据集是纽约的出租车和共享单车数据集。（BTW：国外的很多数据集都是公开的，大家从网上都可以下载到，当国内数据不太好获取时，不妨走出国门搜一搜）

13.4 符号说明

将整个城市划分为 $a*b$ 的网格， $n=a*b$ 代表网格数量，我们将一个区域的起始交通量定义为在固定的时间间隔内从该区域出发/到达该区域的次数。 $y_{i,t}^s$ 和 $y_{i,t}^e$ 代表区域 i 在时间间隔 t 内的客流产生量和吸引量。 $f_{i,t}^{j,\tau}$ 表示 OD 客流： t 时间从 i 区域出发， τ 时间到达 j 区域。

问题：给定 t 时刻及以前的客流，预测 $t+1$ 时刻出发和到达所有区域的客流。

13.5 STDN 时空动态网络

文中将该网络分成了三部分：(1) Local Spatial-Temporal Network, 局部时空网络；(2) Spatial Dynamic Similarity: Flow Gating Mechanism, 门控机制；(3) Temporal Dynamic Similarity: Periodically Shifted Attention Mechanism, 周期性注意力转移机制。

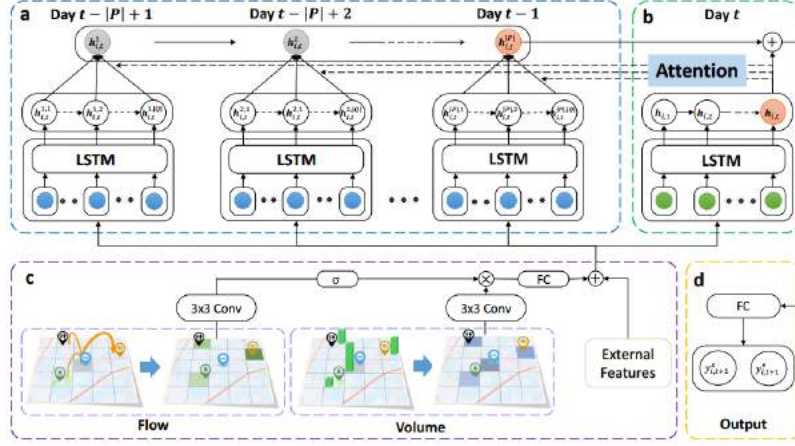


Figure 1: The architecture of STDN. (a) Periodically shifted attention mechanism captures the long-term periodic dependency and temporal shifting. For each day, we also use LSTM to capture the sequential information. (b) The short-term temporal dependency is captured by one LSTM. (c) The flow gating mechanism tracks the dynamic spatial similarity representation by controlling the spatial information propagation; FC means fully connected layers and Conv means several convolutional layers. (d) A unified multi-task prediction component predicts two types of traffic volumes simultaneously.

(1) 局部时空网络

使用 CNN 处理局部空间依赖，对于每个时间间隔 t ，我们将目标区域 i 及其周围的邻域作为一个具有两个通道的 $S \times S$ 图像， $\mathbf{Y}_{i,t} \in \mathbb{R}^{\tilde{S} \times S \times 2}$ 。一个通道包含客流产生量信息，另一个通道包含客流吸引量信息，目标区域在图像的中心。卷积公式：

$$\mathbf{Y}_{i,t}^{(k)} = \text{ReLU}(\mathbf{W}^{(k)} * \mathbf{Y}_{i,t}^{(k-1)} + \mathbf{b}^{(k)}), \quad (1)$$

其中 k 代表层数，卷积层后接 flatten 层和全连接层输出空间依赖关系。

使用原始的 LSTM 处理时间依赖关系。

$$\mathbf{h}_{i,t} = \text{LSTM}([\mathbf{y}_{i,t}; \mathbf{e}_{i,t}], \mathbf{h}_{i,t-1}), \quad (2)$$

其中 y 为 CNN 的输出， e 代表外部因素（天气等）， h 为上一时刻的输出。

(2) 空间动态相似性：门控机制

然而，既有文章提取的空间依赖性静态的，不能充分反映目标区域与其相邻区域之间的动态关系。表示区域间交互的更直接的方法是交通流。如果两个区域之间存在更多的客流，它们之间的关系就会更强(即，它们更相似，也就是文章中说的相似性)。因此，我们设计了一种门控机制(Flow Gating Mechanism)，以捕捉层次

结构中的动态空间依赖关系。

这块其实就是将局部区域 $S \times S$ 网格内的 OD 矩阵考虑到了空间关系的提取当中去了。提取两个矩阵，一个是出发的 OD 矩阵（ t 时刻 O 点出发到达各 D 点），一个是到达的 OD 矩阵（ t 时刻各个 D 点出发到达 O 点），其实这类也不能叫做 OD 矩阵，暂且这样叫吧。（这一块感觉提取方式就是提取全网的两个矩阵，然后用 3×3 的 filter 去卷积，就是所谓的局部区域 $S \times S$ 网格）

给定一个特定的区域 i ，我们从过去的 l 个时间间隔（即，时间区间 $t-l+1$ 到 t ）。将获得的 OD 矩阵进一步叠加，用 $\mathbf{F}_{i,t} \in \mathbb{R}^{S \times S \times 2l}$ 表示，其中 $S \times S$ 表示周围相邻区域的大小， $2l$ 为流动矩阵的个数（每个时间间隔两个上面说的 OD 矩阵），叠加的矩阵包含了所有与区域 i 相关的过去的流量间的相互作用。这一块的卷积结果用下式表示：

$$\mathbf{F}_{i,t}^{(k)} = \text{ReLU}(\mathbf{W}_f^{(k)} * \mathbf{F}_{i,t}^{(k-1)} + \mathbf{b}_f^{(k)}), \quad (3)$$

将式（1）和式（3）两个卷积的 output 相乘，即得到式（4）。这个乘法也就是文章中所说的门控机制。同样，卷积层后接 flatten 层和全连接层输出空间依赖关系，至此空间关系提取完毕。

$$\mathbf{Y}_{i,t}^{(k)} = \text{ReLU}(\mathbf{W}^{(k)} * \mathbf{Y}_{i,t}^{(k-1)} + \mathbf{b}^{(k)}) \otimes \sigma(\mathbf{F}_t^{i,k-1}), \quad (4)$$

（3）时间动态相似性：周期性注意力转移机制

（要读懂这一块肯定要理解透彻 RNN 注意力机制是怎么实现的，可参考这篇文章，讲的很清楚 https://blog.csdn.net/qq_40027052/article/details/78421155）

只使用前几个时间间隔（通常是几个小时），忽略了长期依赖性（如周期性）。所以应该考虑对预测目标的相对时间间隔（例如，昨天和前天的同一时间）进行建模。然而，单纯考虑相对时间间隔是不够的，忽略了周期性的时间转移，即，交通数据并非严格意义上的周期性。例如，工作日的高峰时间通常在下午，但可以从下午 4:30 到 6:00。由于交通事故或交通拥挤，周期性信息的时间转移在交通序列中普

遍存在。下图中显示了在不同的天和周之间的时间转移。因此，我们设计了一个周期性注意力转移机制(Periodically Shifted Attention Mechanism)来解决这些问题。

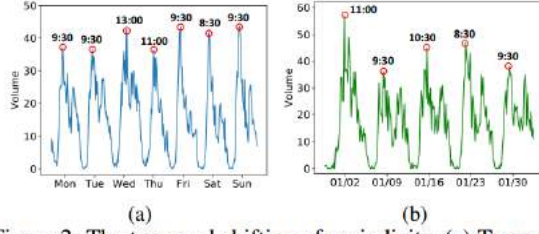


Figure 2: The temporal shifting of periodicity. (a) Temporal shifting between different days. (b) Temporal shifting between different weeks. Note that, each time in these figures represents a time interval (e.g., 9:30am means 9:00-9:30am).

我们的重点是处理不同天之间的周期性的变化。如图 1(a)所示，包含了前 P 天对应的的时间间隔来处理周期性依赖关系。对于每一天，为了解决时间移动的问题，我们在 Q 中进一步选择每天的 Q 时间间隔。例如，如果预测的时间是晚上 9:00-9:30，我们选择预测时间的前后 1 小时(即 8 点-10:30pm, $Q = 5$)。这些时间间隔 $q \in Q$ 用于处理潜在的时间周期性转移。此外，我们使用 LSTM 来表达每天 $p \in P$ 的序列信息，其表达式为：

$$\mathbf{h}_{i,t}^{p,q} = \text{LSTM}([\mathbf{y}_{i,t}^{p,q}; \mathbf{e}_{i,t}^{p,q}], \mathbf{h}_{i,t}^{p,q-1}), \quad (5)$$

$\mathbf{h}_{i,t}^{p,q}$ 为第 i 区域预测时间 t 在前一天 p 中的时间 q 的表示 (p 代表同一天，可以把 p 去掉来看这个公式)。我们采用注意机制来捕捉时间的变化，并得到每一天的加权表示。形式上，某天的 $\mathbf{h}_{i,t}^p$ 表示为每个选定时间间隔 q 中表示的加权和(即上面说的 $Q=5$ 那 5 个时段的加权和)，定义为：

$$\mathbf{h}_{i,t}^p = \sum_{q \in Q} \alpha_{i,t}^{p,q} \mathbf{h}_{i,t}^{p,q}, \quad (6)$$

α 代表同一天不同时段的重要性，其计算就用到了注意力机制：

$$\alpha_{i,t}^{p,q} = \frac{\exp(\text{score}(\mathbf{h}_{i,t}^{p,q}, \mathbf{h}_{i,t}))}{\sum_{q \in Q} \exp(\text{score}(\mathbf{h}_{i,t}^{p,q}, \mathbf{h}_{i,t}))}. \quad (7)$$

其中 h 就是 hidden state，上述中 5 个时间间隔的隐层状态（也即 5 个 timesteps 的隐层状态），score 函数就相当于注意力机制中的相似性函数，整个函数就是个 softmax。

上面求的是前一天的一个总的加权和隐藏层状态，前一天的状态传递到当天即为公式：

$$\hat{\mathbf{h}}_{i,t}^p = \text{LSTM}(\mathbf{h}_{i,t}^p, \hat{\mathbf{h}}_{i,t}^{p-1}). \quad (9)$$

Figure1 (a) 可以理解为有分支的 stacked LSTM, 分支求的是每天的隐层状态加权和，主 LSTM 利用该加权和进行预测。

13.6 案例研究

将在下篇文章中讲述。

13.7 总结

模型很复杂，实现起来比较麻烦，github 上有代码可以参考。

<https://github.com/tangxianfeng/STDN>

14 考虑动态时空相似性的交通预测深度学习框架-案例研究

本篇文章是上一篇文章的案例研究部分。公众号内回复“纽约”获取文章使用的相关数据下载地址，包括美国纽约 10 年的出租车轨迹数据以及 7 年的共享单车轨迹数据。

14.1 文章信息

《Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction》

AAAI（美国人工智能协会）2019 大会上的论文。作者: 宾夕法尼亚州立大学的 Huaxiu Yao。

14.2 摘要

如何对复杂的时空依赖关系建模是当前交通预测的一个重点。既有研究大多将空间依赖看作是静态的，时间依赖看作是严格周期性的。但本文基于两点进行研究：(1)位置之间的空间依赖关系是动态的；(2)时间依赖服从日、周两种模式，但其动态时间变化不具有严格的周期性。为了解决这两个问题，我们提出了一种新的时空动态网络(STDN)，其中引入了一种门控机制来学习位置之间的动态相似性，并设计了一种周期性注意力转移机制来处理长期的周期性时间转移。

14.3 案例研究

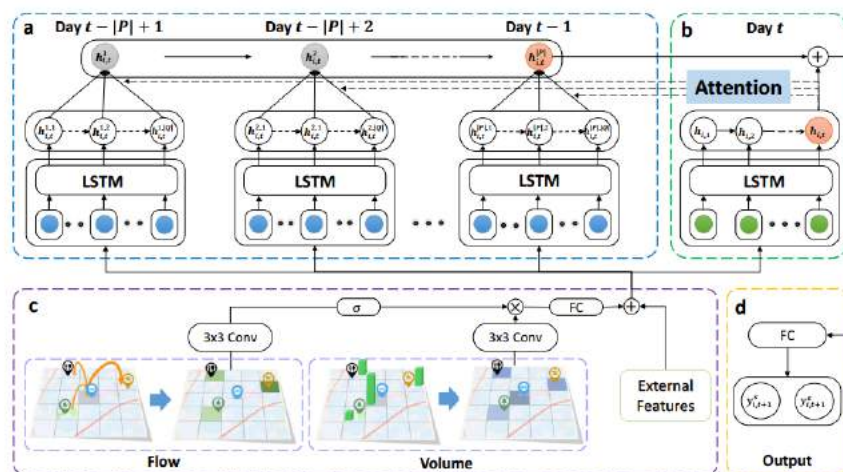


Figure 1: The architecture of STDN. (a) Periodically shifted attention mechanism captures the long-term periodic dependency and temporal shifting. For each day, we also use LSTM to capture the sequential information. (b) The short-term temporal dependency is captured by one LSTM. (c) The flow gating mechanism tracks the dynamic spatial similarity representation by controlling the spatial information propagation; FC means fully connected layers and Conv means several convolutional layers. (d) A unified multi-task prediction component predicts two types of traffic volumes simultaneously.

文章 joint training 部分说的将短期依赖和长期依赖连接起来指的就是上图中的 b 部分，即将 Day t 的短期依赖和 t 天以前的长期依赖通过那个加号连接起来，然后接全连接层输出最终结果。

模型的损失函数由于是预测的各个区域的吸引量和产生量，所以也就变为

$$\mathcal{L} = \sum_{i=1}^n \lambda (y_{i,t+1}^s - \hat{y}_{i,t+1}^s)^2 + (1 - \lambda) (y_{i,t+1}^e - \hat{y}_{i,t+1}^e)^2, \quad (11)$$

是 inflow 和 outflow 共同 MSE， λ 是衡量两者影响大小的参数。

文章所用数据是纽约的出租车和共享单车数据集。后台回复“纽约”获取相应数据集。

出租车：从 2015 年 1 月 1 日到 2015 年 3 月 1 号共 2200 万条出行记录，其中前 40 天训练，后 20 天测试。

共享单车：自行出轨迹数据是从 2016 年 7 月 1 号到 2016 年 8 月 29 号共 260 万条出行记录，其中前 40 天训练，后 20 天测试。

数据处理：我们将整个城市划分为 10×20 个区域。每个区域的大小约为 $1\text{km} \times 1\text{km}$ 。每个时间间隔的长度设置为 30 分钟。我们使用 Min-Max 归一化将流量转换为 $[0: 1]$ 规模。预测完成后，对预测值进行反归一化并进行评价。我们对训练和测试数据都使用滑动窗口来生成样本。在对模型进行测试时，我们对体积值小于 10 的样本进行过滤，这是工业和学术中常用的做法。因为在实际的应用中，关心低流量几乎没有什么意义。我们选择 80% 的训练数据用于学习模型，剩下 20% 用于验证。

评价指标：MAPE, RMSE

参数设置：我们根据验证集的性能来设置超参数。对于空间 CNN 卷积部分，我们将所有卷积核大小都设置为 3×3 ，64 过滤器。所考虑的每个邻域的大小设为 7×7 。设 $K = 3$ (层数)， $l = 2$ (考虑 OD 矩阵的时间跨度)。对于时间 LSTM 部分，我们将短期 LSTM 的长度设为 7 (即前 3.5 个小时)， $|P| = 3$ 表示长期周期信息 (即前 3 天)， $|Q| = 3$ 为周期性注意力转移机制 (即考虑相对于预测时间的前后半小时)，LSTM 隐藏层维数是 128。通过 Adam 进行优化，batch size 设置为 64。学习率设置为 0.001。LSTM 的 dropout 和 recurrent dropout 均为 0.5。 λ 设置为 0.5。

结果：不用说了，大家心知肚明，比所有模型都好。

14.4 思考

关于文章空间动态相似性：门控机制那部分，个人认为，使用 OD 矩阵需考虑一个问题就是，进行实时预测时，虽然实时的交通产生量和吸引量可以获取，但实时的 OD 矩阵是获取不了的，因为从 O 点到 D 点需要一定时间，存在一个 time lag 的问题，当然历史数据的 OD 矩阵是可以获取的，用历史 OD 数据来做模型是有很大的问题的。有想法的可以私聊就该问题进行探讨。

14.5 数据获取

公众号内回复“纽约”获取文章使用的相关数据下载地址，包括美国纽约 10 年的出租车轨迹数据以及 7 年的共享单车轨迹数据。

15 ConvLSTM 参数详解

15.1 最重要的输入尺寸和输出尺寸

以 `data_format='channels_last'` 为例：

输入尺寸：输入 5D 张量，尺寸为：(samples,time, rows, cols, channels)。即要提前将训练集和测试集 reshape 成如上形式的 tensor 张量，例如下图：

```
X_train = data1.reshape(1, data1.shape[0], data1.shape[1], data1.shape[2], 1)
Y_train = data2.reshape(1, data2.shape[0], data2.shape[1], data2.shape[2], 1)
```

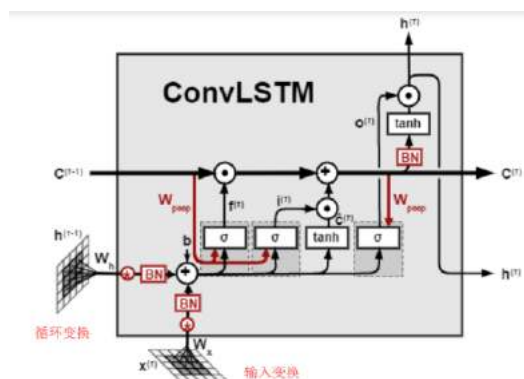
输出尺寸：如果 `return_sequences`，返回 5D 张量，尺寸为：(samples, timesteps, output_row, output_col, filters)。否则，返回 4D 张量，尺寸为：(samples, output_row, output_col, filters)。（o_row 和

`o_col` 取决于 `filter` 和 `padding` 的尺寸)

注意：假如上一层是 `ConvLSTM2D layer`，那么其输出为以上形式的 4D 张量或 5D 张量，当后面再接另外一个 `layer` 时，就要考虑该 `layer` 是否能接受 4D 张量或 5D 张量（即要考虑 `ConvLSTM2D` 的输出能否作为该 `layer` 的输入）

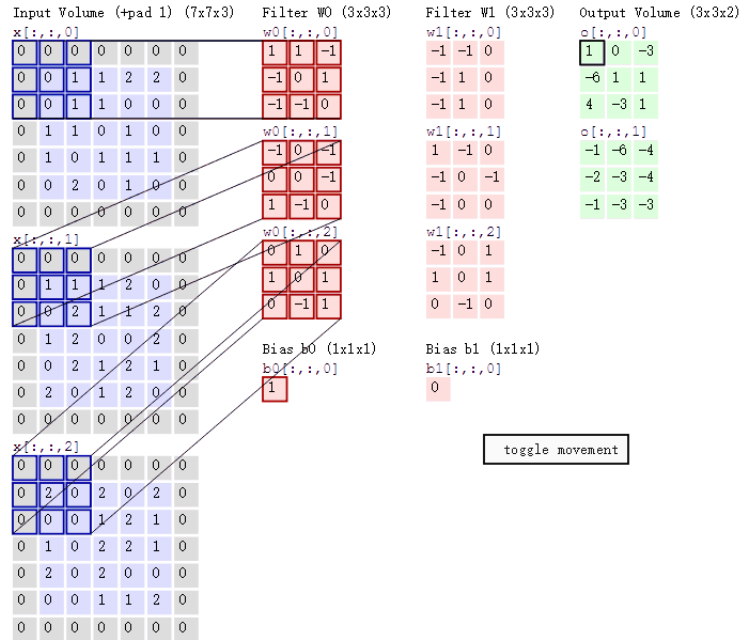
15.2 参数

`ConvLSTM2D` 是一个 LSTM 网络，但它的输入变换和循环变换是通过卷积实现的如下图。其很多参数的理解上也可参考 LSTM。



共 25 个参数，以下图为例对几个常用的参数进行解释，其他的请查阅 Keras 官网。

```
keras.layers.ConvLSTM2D(filters, kernel_size, strides=(1, 1),
padding='valid', data_format=None, activation='tanh',
recurrent_activation='hard_sigmoid', use_bias=True,
return_sequences=False, stateful=False, dropout=0.0,
recurrent_dropout=0.0)
```



➤ filters: 卷积核的数目（也就是下图中的两个 filter W0 和 W1，也可以理解为输出的维度即两个绿色的框，filter0 对应上面的绿色框，filter1 对应下面的绿色的框）

➤ kernel_size: 卷积核大小，上图中 filterW0 的宽和高（单个整数或由两个整数构成的 list/tuple，卷积核的宽度和长度。如为单个整数 1，则表示 kernel_size=（1×1）。）

➤ strides=(1, 1): 为卷积的步长，即卷积核向右和向下一次移动几格，对应图中最左列蓝色 3×3 网格向右和向下移动的步长。

（单个整数或由两个整数构成的 list/tuple，为卷积的步长。如为单个整数，则表示在各个空间维度的相同步长。）

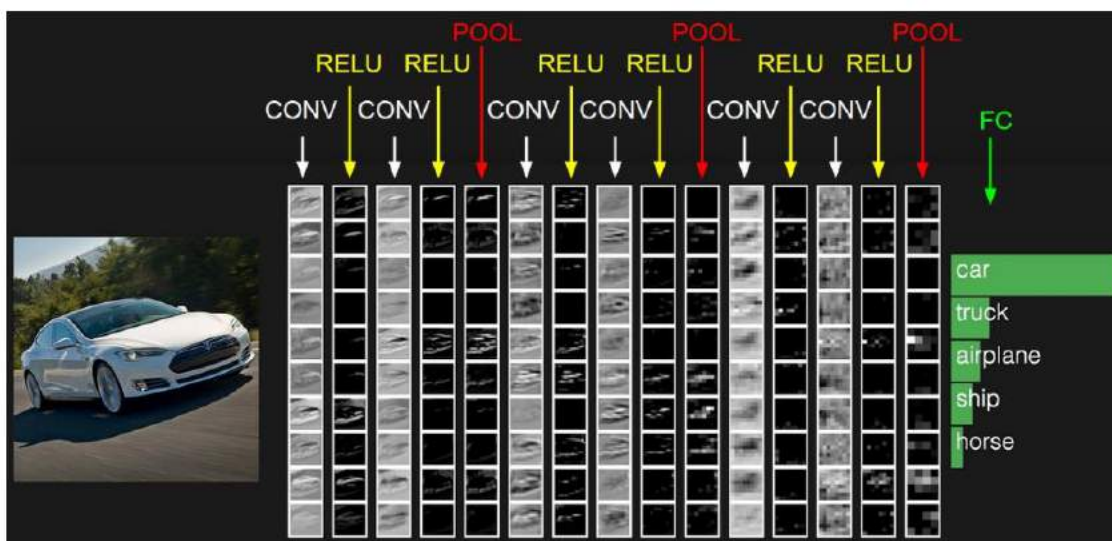
➤ padding: 补 0 策略，为“valid”或“same”。要对所有的像素

点进行处理就用 same, same 也是通用的, 很少有用 valid 的。

(“valid”代表只进行有效的卷积, 即对边界数据不处理。“same”代表保留边界处的卷积结果, 通常会导致输出 shape 与输入 shape 相同。)

➤ data_format: 即红绿蓝三个通道(channel)是在前面还是在后面, channels_last (默认) (width, height, channel) 或 channels_first (channel, width, height) 之一, 输入中维度的顺序。

➤ activation: 激活函数, 即下图中的 RELU 层, 为预定义的激活函数名, 如果不指定该参数, 将不会使用任何激活函数 (即使用线性激活函数: $a(x)=x$)



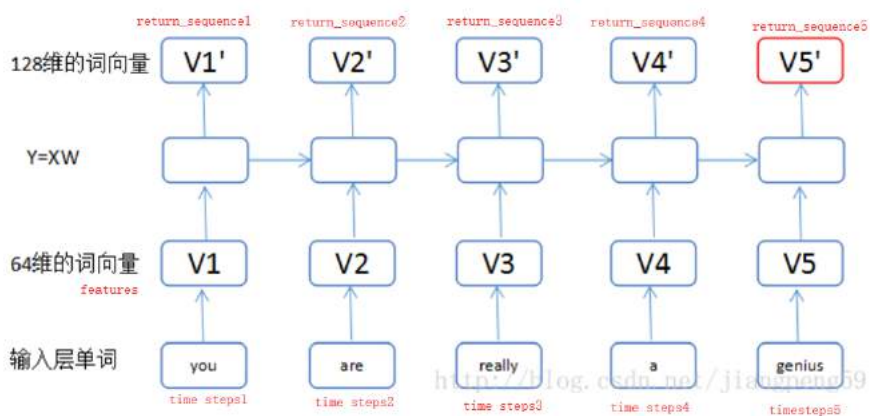
上图中CNN要做的事情是: 给定一张图片, 是车还是马未知, 是什么车也未知, 现在需要模型判断这张图片里具体是一个什么东西, 总之输出一个结果: 如果是车 那是什么车

➤ recurrent_activation: 用于循环时间步的激活函数

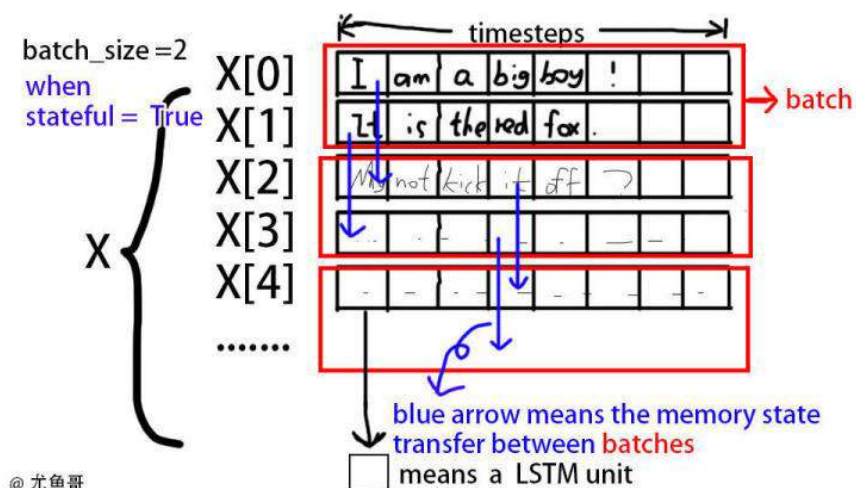
(recurrent_activation 是针对于 input/forget/output 三个 gate 的激活函数. activation 是针对于 cell state 和 hidden state 的激活函数. 下面所有参数带 recurrent

和不带 recurrent 的区别和此处相同)

- **use_bias**: 布尔值, 是否使用偏置项
- **return_sequences**: 布尔值。是返回输出序列中的最后一个输出, 还是全部序列。为 **true** 的话下图中 1 到 5 全部输出, 为 **false** 的话下图中只输出红框中的 5。



- **stateful**: 布尔值 (默认 **False**)。如果为 **True**, 则批次中索引 i 处的每个样品的最后状态 将用作下一批次中索引 i 样品的初始状态。对 **stateful** 的理解请看下图。Stateful 为 **true** 时, 训练和测试时的 **batch size** 必须得相同, 且样本数目必须得能被 **batch size** 整除。



- **dropout**: 在 0 和 1 之间的浮点数。单元的丢弃比例，用于输入的线性转换，防止过拟合。
- **recurrent_dropout**: 在 0 和 1 之间的浮点数。单元的丢弃比例，用于循环层状态的线性转换。

16 ResNet: 基于深度时空残差网络 ResNet 的城市交通流预测

16.1 文章信息

《Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction》。

2017 年 AAAI 大会的一篇会议论文，作者微软研究院的张俊波，会后文章发在爱斯维尔旗下期刊 Artificial Intelligence (IF:3.034) ,目前被引 257 次。

16.2 摘要

我们提出了一种基于深度学习的方法，称为时空残差网络 ST-ResNet，来同时预测城市中每个区域的流入和流出客流量。更具体地说，我们使用残差神经网络框架来对拥堵流的时间临近性、周期和趋势特性建模（closeness, period, trend，暂且这样翻译吧）。针对每个属性，我们设计了一个残差卷积单元的分支，每个残差卷积单元对拥堵流的空间特性进行建模，ST-ResNet 动态整合三个残差神经网络分支的输出，为不同的分支和区域分配不同的权重。将整合结果进一步结合外部因素（external），如天气和一周中的哪一天，来预测每个地区最终流量。北京和纽约市(NYC)的实验表明，提出的 ST-ResNet 优于六种著名的方法。

16.3 简介

简介部分用了北京和纽约的两个拥堵踩踏案例引出，第一次见。然后强调介绍了本文同时预测 inflow 和 outflow。同时预测流入流和流出流由于以下因素的影响极具挑战性：

空间依赖性 spatial dependency：邻近区域的流入流和流出流相互影响。

时间依赖性 temporal dependency：临近时刻的影响，前一天，上一周，季节性等周期特性。

外部因素的影响 external factors：天气和突发事件等。

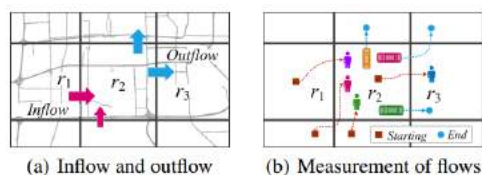


Figure 1: Crowd flows in a region

本文的四点主要内容：

1、ST-ResNet 采用基于卷积的残差网络对城市中任意两个区域之间的远近空间依赖关系进行建模，同时保证模型的预测精度不受神经网络深度结构的影响。

2、我们将人群流动的时间特性归纳为三类，即时间邻近性、周期性和趋势

性。ST-ResNet 使用三个残差网络分支分别对这些特性进行建模。

3、ST-ResNet 动态地聚合上述三个网络的输出，为不同的分支和区域分配不同的权重。这种聚集进一步与外部因素(如天气)相结合。

4、我们使用北京出租车的轨迹和气象数据，以及纽约自行车的轨迹数据来评估我们的方法。

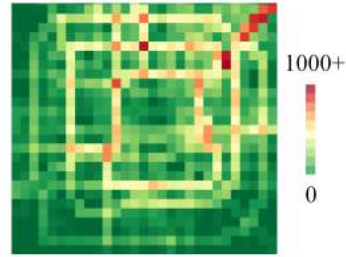
16.4 模型介绍

在本研究中，我们根据经纬度将一个城市划分为 $I \times J$ 网格地图，其中网格表示一个区域，如图 2(a)所示。设 P 为第 t 个时间区间的轨迹集合。网格 (i, j) 位于第 i 行和第 j 列。和既有研究类似，也是将流入流和流出流叠在一起看成一个 $2 \times I \times J$ 的张量（2 个 channel，每个 channel 为 I 行 J 列）

At the t^{th} time interval, inflow and outflow in all $I \times J$ regions can be denoted as a tensor $\mathbf{X}_t \in \mathbb{R}^{2 \times I \times J}$ where $(\mathbf{X}_t)_{0,i,j} = x_t^{in,i,j}$, $(\mathbf{X}_t)_{1,i,j} = x_t^{out,i,j}$. The inflow matrix is shown in Figure 2(b).



(a) Grid-based map segmentation



(b) Inflow matrix

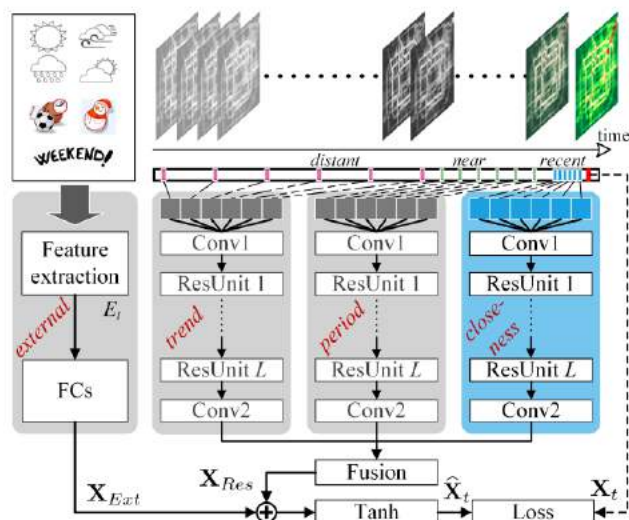


Figure 3: ST-ResNet architecture. Conv: Convolution; ResUnit: Residual Unit; FC: Fully-connected.

网络结构示意图

该网络结构主要由 4 部分组成，分别提取时间邻近性、周期性、趋势性以及外部因素的影响。

上图右上角，首先将一个城市在每个时间间隔内的流入流和流出流分别转化为一个 2 通道的类图矩阵，然后我们将时间轴划分为三个片段，表示最近的时间、稍远的时间和遥远的时间（其实文章中这块就是考虑了邻近时间段，前一天相同时间段，上一周相同时间段）。每一个片段分别被输入到三个 ResNet 分支中用来提取提取时间邻近性、周期性、趋势性。前三个部分与卷积神经网络共享相同的网络结构，然后是残差单元序列。这种结构捕捉了附近和远处区域之间的空间依赖关系。在外部因素的处理上，我们手动从外部数据集中提取一些特征，例如天气条件和事件，并将它们输入一个两层全连接的神经网络。前三部分的输出与外部因素的输出进行融合，然后利用 tanh 激活函数映射到 $(-1, 1)$ 区间内作为输出。

前三部分的网络结构：

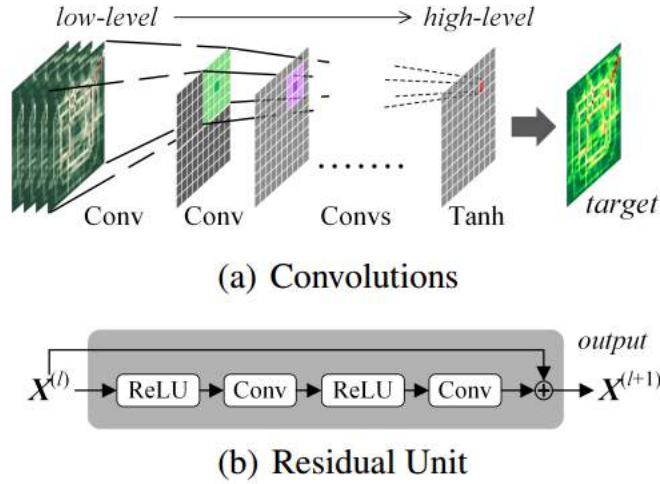


Figure 4: Convolution and residual unit

外部因素：

交通流量可以受到许多复杂的外部因素的影响，如天气和事件。图 5(a)显示节假日(中国春节)的人流与平时的人流有很大的不同。图 5(b)显示，与后一周的同一天相比，暴雨使得办公区域的人流急剧减少。

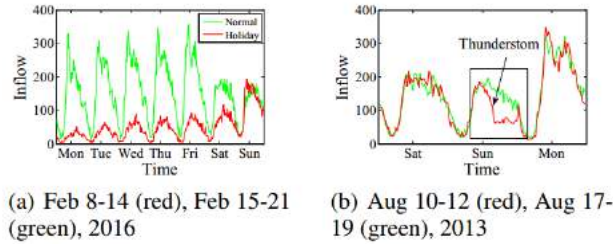


Figure 5: Effects of holidays and weather in Office Area of Beijing (the region is shown in Figure 2(a)).

设 E_t 为表示这些外部因素在预测时间区间 t 的特征向量。在本文中，我们主要考虑天气、假日事件和元数据(即工作日、工作日/周末)。为了预测 t 时段的流量，可以直接获取假日事件和元数据。然而，未来时间间隔 t 的天气是未知的，可以使用时间间隔 t 的天气预报或时间间隔 $t-1$ 的近似天气预报。在形式上，我们将两个全连接层叠加在 E_t 上，第一层可以看作是每个子因子的带激活函数的嵌入层。第二层用于将上一层的输出映射成与 X_t 相同形状便于融合。

融合：

对于前三部分的融合，由于不同区域都受时间邻近性、周期性、趋势性的影响，但影响程度可能不同。在此基础上，提出了一种基于参数矩阵的融合方法：

$$\mathbf{X}_{Res} = \mathbf{W}_c \circ \mathbf{X}_c^{(L+2)} + \mathbf{W}_p \circ \mathbf{X}_p^{(L+2)} + \mathbf{W}_q \circ \mathbf{X}_q^{(L+2)} \quad (4)$$

对于前三部分和外部因素的融合，直接相加再激活：

$$\hat{\mathbf{X}}_t = \tanh(\mathbf{X}_{Res} + \mathbf{X}_{Ext}) \quad (5)$$

损失函数：MSE。

16.5 模型超参数

在 ST-ResNet 的输出中，使用 tanh 作为最后的激活函数，其范围在-1 到 1 之间。使用 Min-Max 归一化方法将数据缩放到[- 1;1]。在评估中，我们将预测值重新缩放到正常值，并与 ground truth 进行比较。对于外部因素，我们使用 one hot 编码来转换元数据(即，Day Of Week, Weekend/Weekday)，节假日和天气条件为二元向量，利用 Min-Max 归一化将温度和风速等缩放到[0;1]。

使用 Python 中的 Theano 和 Keras 构建模型。Conv1 和所有剩余单元使用 64 个尺寸为 3×3 的滤波器，Conv2 使用 2 个尺寸为 3×3 的滤波器进行卷积。批次大小为 32。90% 的训练集，剩下的 10% 作为验证集。在固定数量的 epoch(例如 10,100 个 epoch)的完整训练数据上训练模型。考虑前一天和上一周的数据。评价指标 RMSE。

16.6 实验部分

本文采用了两个案例，北京的出租车轨迹数据和纽约的共享单车轨迹数据，数据量和数据范围都很大，实验部分比较复杂，没什么特别大的借鉴意义，不再赘述。实验部分比较复杂，没什么特别大的借鉴意义，不再赘述。

16.7 思考

该篇论文也用到了前面共享过的纽约公开的共享单车轨迹数据集。

此外，该类复杂结构模型的一个鸡肋点是如何选择基准进行比较，输入因素这么多，其他基准模型输入是什么，模型配置是什么样的，这在既有文献中大多是直接给出结果，过程基本是一个黑箱子，目前也没有统一的解决办法。

17 图卷积神经网络 GCN ---如何在交通预测中考虑网络拓扑结构

17.1 文章信息

《Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting》。

2017 年第二十七届 IJCAI 国际人工智能联合会议的一篇会议论文，作者是北京大学数学科学学院的一名博士生余冰,被引 53 次。（可能是由于非专业计算机，里面的很多计算机的东西介绍的比较模糊，且更偏重数学推导。）

17.2 摘要

在本文中，我们提出了一个新的深度学习框架，时空图卷积网络(STGCN)，解决交通领域的时间序列预测问题，我们没有使用常规的 CNN 和 RNN 单元，而是将问题用图表示出来，并建立具有完整卷积结构的模型，这使得模型训练速度更快，参数更少。实验表明，STGCN 模型通过对**多尺度**交通网络的建模，有效地捕获了时空相关性。

17.3 简介

对于深度学习方法而言，常规 CNN 只能处理网格结构(如图像、视频)，而不

处理一般网络结构，而 RNN 需要迭代训练，引入了逐级累积的误差，且基于 RNN 的网络(包括 LSTM)难以训练，计算量大。为了克服这些问题，我们引入了几种策略来有效地对交通流的时间动态和空间依赖性进行建模。为了充分利用空间信息，我们没有单独处理交通网络，而是用一般的图来对其进行建模（例如网格或路段）。为了解决递归网络的固有不足，我们在时间轴上采用了全卷积结构。总之，我们提出了一种新的深度学习结构，时空图卷积网络，用于交通预测任务。该体系结构由多个时空卷积块组成，时空卷积块由图卷积层和卷积序列学习层组合而成（参考论文 Convolutional neural networks on graphs with fast localized spectral filtering），对时空依赖关系进行建模。据我们所知，这是第一次使用图卷积神经网络结构进行交通预测。

17.4 基础介绍

在本文中，我们在一个图上定义了交通网络，重点研究了结构化交通时间序列。 V_t 在图中不是相互独立的，而是通过成对连接起来的。因此，数据点 V_t 可以看作是一个图信号（也就是文中所说的 signal），它定义在一个权值为 W_{ij} 的无向图(或有向图) G 上，如图 1 所示。

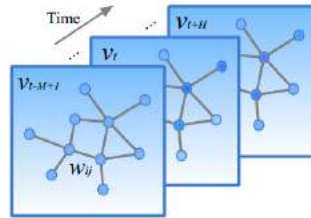


Figure 1: Graph-structured traffic data. Each v_t indicates a frame of current traffic status at time step t , which is recorded in a graph-structured data matrix.

在第 t 时刻，在图 $G_t = (V_t, E, W)$ 中， V_t 是一个有限的顶点集合，对应于交通网络中 n 个监测站的观测值； E 是一组边，表示站点之间的连通性；当 W 表示 G_t 的加权邻接矩阵。

常规网格化的标准卷积显然不适用于一般图。目前，有两种基本方法正在探索

如何将 CNNs 推广到结构化数据表。一种是扩展卷积的空间定义（参考论文 Learning convolutional neural networks for graphs），另一种是利用图的傅里叶变换在光谱域中进行操作（参考论文 Spectral networks and locally connected networks on graphs）。前一种方法将顶点重新排列成特定的网格形式，这些网格形式可以通过常规的卷积操作来处理。后者引入了频谱框架，将卷积应用于频谱域，通常称为频谱图卷积。后续的几项研究通过将计算复杂度从 $O(n^2)$ 降低到线性，使得图卷积更有前景（参考论文 1: Convolutional neural networks on graphs with fast localized spectral filtering, 参考论文 2: Semi-supervised classification with graph convolutional networks）。

本文基于谱图卷积介绍了图卷积算子的概念 $*_{\mathcal{G}}$ ，代表图信号 x 与内核 Θ 的乘积。

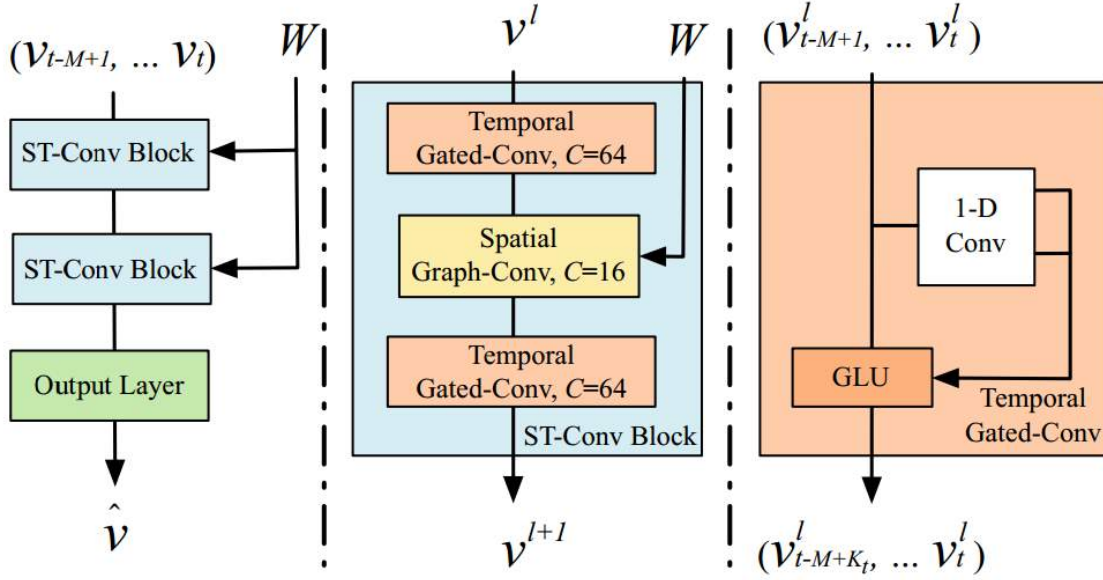
$$\Theta *_{\mathcal{G}} x = \Theta(L)x = \Theta(U\Lambda U^T)x = U\Theta(\Lambda)U^T x, \quad (2)$$

eigenvectors of the normalized graph Laplacian $L = I_n - D^{-\frac{1}{2}}WD^{-\frac{1}{2}} = U\Lambda U^T \in \mathbb{R}^{n \times n}$ (I_n is an identity matrix,

图的傅里叶基 U 是归一化的特征向量的矩阵图拉普拉斯算子 L ，（ L 中 I 是单位阵， D 是对角度矩阵， W 是邻接矩阵） Λ 是 L 的特征值对角矩阵，滤波器 $\Theta(\Lambda)$ 也是一个对角矩阵。根据这个定义，一个图信号 x 由内核过滤器 Θ 与图傅里叶变换 $U^T x$ 之间的多次乘法进行 filter。

17.5 模型框架

如图 2 所示，STGCN 由多个时空卷积块组成，每个卷积块组成类似于“三明治”结构，其中包含两个门控顺序卷积层和一个空间图卷积层。每个模块的细节如下。（最右边的组成中间的，中间的组成左边的，这种表示方法也是有点奇怪）



17.6 Graph CNNs 提取空间特征

交通网络一般以图形结构组织。用图来表示道路网络是自然和合理的。然而，以往的研究忽略了交通网络的空间属性，忽略了网络的连通性和全局性，因为它们被划分为多个段或网格。即使使用网格上的二维卷积，由于数据建模的折衷，它也只能粗略地捕获空间局部性。因此，在我们的模型中，将图卷积直接应用于图结构数据，在空间域中提取具有高度意义的模式和特征。式（2）中由于核函数要不断与图的傅里叶基相乘导致计算复杂度很高，本文采用两种近似策略来克服这一问题：切比雪夫多项式近似和1阶近似（推到过程其实可以不用看，直接用推倒结果即可）。近似之后图卷积操作即为：

$$\begin{aligned}\Theta *_{\mathcal{G}} x &= \theta(I_n + D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) x \\ &= \theta(\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}}) x.\end{aligned}\quad (5)$$

其中 \tilde{W} 为邻接矩阵和单位矩阵的和， \tilde{D} 为 \tilde{W} 的度对角矩阵， θ 为参数矩阵。

图卷积也可扩展到多维张量。

17.7 Gated CNN 门控提取时间特征

我们利用时间轴上的整个卷积结构来捕捉交通流的时间动态行为。这种特殊的设计允许通过多层卷积结构实现并行和可控的训练过程。

如图 2(右)所示，时域卷积层包含一个一维卷积，其次是门控线性单元(GLU，这个东西文章里面介绍的模模糊糊，没看懂说的是啥)。此外，在叠加的时间卷积层之间还添加了一个残差连接。

17.8 时空卷积块

为了融合时空域的特征，构造了时空卷积块(ST-Conv 块)联合处理图结构时间序列。块本身可以根据特定情况的规模和复杂性进行堆叠或扩展。

如图 2 (mid)所示，中间的空间层是连接两个时间层的桥梁，通过时间卷积可以实现图卷积到时间卷积的快速时空传播。“三明治”结构还帮助网络充分应用 bottleneck 策略（即两头大中间小，两边是 64 个 filter，中间是 16 个 filter），通过图卷积层对通道 C（也就是 filter）进行下缩放和上缩放，实现尺度压缩和特征压缩。此外，每个层都进行了归一化，防止过拟合。在叠加两个 ST-Conv 块后，我们在最后附加了一个额外的时域卷积层和一个全连通层作为输出层。

模型特点总结：

STGCN 是处理结构化时间序列的通用框架。它不仅能够解决交通网络建模和预测问题，而且可以应用于更一般的时空序列学习任务。

时空卷积块结合了图卷积和门控时间卷积，能够提取出最有用的空间特征，并连贯地捕捉到最基本的时间特征。

该模型完全由卷积结构组成，在输入端实现并行化，参数更少，训练速度更快。更重要的是，这种经济架构允许模型以更高的效率处理大规模网络。

6、实验

本文的实验也是用了两个数据集来验证（现在都流行两个数据集了么……），并与几个模型进行了对比，模型对比也并没有说清楚用了什么配置就直接给出了结果……

Model	BJER4 (15/ 30/ 45 min)		
	MAE	MAPE (%)	RMSE
HA	5.21	14.64	7.56
LSVR	4.24/ 5.23/ 6.12	10.11/ 12.70/ 14.95	5.91/ 7.27/ 8.81
ARIMA	5.99/ 6.27/ 6.70	15.42/ 16.36/ 17.67	8.19/ 8.38/ 8.72
FNN	4.30/ 5.33/ 6.14	10.68/ 13.48/ 15.82	5.86/ 7.31/ 8.58
FC-LSTM	4.24/ 4.74/ 5.22	10.78/ 12.17/ 13.60	5.71/ 6.62/ 7.44
GCGRU	3.84/ 4.62/ 5.32	9.31/ 11.41/ 13.30	5.22/ 6.35/ 7.58
STGCN(Cheb)	3.78/ 4.45/ 5.03	9.11/ 10.80/ 12.27	5.20/ 6.20/ 7.21
STGCN(1st)	3.83/ 4.51/ 5.10	9.28/ 11.19/ 12.79	5.29/ 6.39/ 7.39

Table 1: Performance comparison of different approaches on the dataset BJER4.

18 2019 运输科技领域最新 SCIE 期刊影响因子

2019 年 6 月 20 号，JCR 公布了最新一年的 SCIE 影响因子，2019 年相比 2018 年变动情况如下，今年运输科技领域新增了 3 个期刊，共 37 个，总体而言，大部分期刊的影响因子都较去年有所上升。

Rank	Full Journal Title	2019	2018
1	COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING	6.208	5.475
2	IEEE Vehicular Technology Magazine	6.145	6.038
3	TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES	5.775	3.968
4	IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS	5.744	4.051
5	IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY	5.339	4.432
6	IEEE Transactions on Transportation Electrification	5.27	新增

7	TRANSPORTATION RESEARCH PART B-METHODOLOGICAL	4.574	4.081
8	TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW	4.253	3.289
9	TRANSPORTATION RESEARCH PART D-TRANSPORT AND ENVIRONMENT	4.051	3.445
10	TRANSPORTATION RESEARCH PART A-POLICY AND PRACTICE	3.693	3.026
11	Vehicular Communications	3.53	3.289
12	TRANSPORTATION	3.457	3.151
13	TRANSPORTATION SCIENCE	3.31	3.338
14	IEEE Intelligent Transportation Systems Magazine	3.294	3.019
15	Journal of Intelligent Transportation Systems	2.568	2.164
16	International Journal of Engine Research	2.272	2.775
17	Transportmetrica B-Transport Dynamics	2.229	2.636
18	NETWORKS & SPATIAL ECONOMICS	2.084	2.695
19	IET Intelligent Transport Systems	2.05	1.387
20	Transportmetrica A-Transport Science	1.988	2.25
21	JOURNAL OF ADVANCED TRANSPORTATION	1.983	1.102
22	European Transport Research Review	1.727	1.758
23	PROCEEDINGS OF THE INSTITUTION OF MECHANICAL ENGINEERS PART F-JOURNAL OF RAIL AND RAPID TRANSIT	1.54	1.103
24	Transport	1.524	1.267

25	INTERNATIONAL JOURNAL OF AUTOMOTIVE TECHNOLOGY	1.523	1.264
26	JOURNAL OF TRANSPORTATION ENGINEERING	1.52	0.983
27	Transportation Letters-The International Journal of Transportation Research	1.5	0.727
28	PROCEEDINGS OF THE INSTITUTION OF MECHANICAL ENGINEERS PART D-JOURNAL OF AUTOMOBILE ENGINEERING	1.275	1.414
29	INTERNATIONAL JOURNAL OF VEHICLE DESIGN	1.19	0.652
30	TRANSPORTATION PLANNING AND TECHNOLOGY	0.893	0.663
31	PROCEEDINGS OF THE INSTITUTION OF CIVIL ENGINEERS-TRANSPORT	0.792	0.239
32	Promet-Traffic & Transportation	0.768	0.456
33	TRANSPORTATION RESEARCH RECORD	0.748	0.695
34	Journal of Transportation Engineering Part B-Pavements	0.722	新增
35	Journal of Transportation Engineering Part A-Systems	0.641	新增
36	International Journal of Heavy Vehicle Systems	0.45	0.105
37	ITE JOURNAL-INSTITUTE OF TRANSPORTATION ENGINEERS	0.155	0.2

2018 年运输科技领域 SCIE 排名情况如下。

Rank	Full Journal Title	2018
1	IEEE Vehicular Technology Magazine	6.038

2	COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING	5.475
3	IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY	4.432
4	TRANSPORTATION RESEARCH PART B-METHODOLOGICAL	4.081
5	IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS	4.051
6	TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES	3.968
7	TRANSPORTATION RESEARCH PART D-TRANSPORT AND ENVIRONMENT	3.445
8	TRANSPORTATION SCIENCE	3.338
9	TRANSPORTATION RESEARCH PART E-LOGISTICS AND TRANSPORTATION REVIEW	3.289
10	Vehicular Communications	3.289
11	TRANSPORTATION	3.151
12	TRANSPORTATION RESEARCH PART A-POLICY AND PRACTICE	3.026
13	IEEE Intelligent Transportation Systems Magazine	3.019
14	International Journal of Engine Research	2.775
15	NETWORKS & SPATIAL ECONOMICS	2.695
16	Transportmetrica B-Transport Dynamics	2.636
17	Transportmetrica A-Transport Science	2.25
18	Journal of Intelligent Transportation Systems	2.164
19	European Transport Research Review	1.758
20	PROCEEDINGS OF THE INSTITUTION OF MECHANICAL ENGINEERS PART D-JOURNAL OF AUTOMOBILE ENGINEERING	1.414

21	IET Intelligent Transport Systems	1.387
22	Transport	1.267
23	INTERNATIONAL JOURNAL OF AUTOMOTIVE TECHNOLOGY	1.264
24	PROCEEDINGS OF THE INSTITUTION OF MECHANICAL ENGINEERS PART F-JOURNAL OF RAIL AND RAPID TRANSIT	1.103
25	JOURNAL OF ADVANCED TRANSPORTATION	1.102
26	JOURNAL OF TRANSPORTATION ENGINEERING	0.983
27	Transportation Letters-The International Journal of Transportation Research	0.727
28	TRANSPORTATION RESEARCH RECORD	0.695
29	TRANSPORTATION PLANNING AND TECHNOLOGY	0.663
30	INTERNATIONAL JOURNAL OF VEHICLE DESIGN	0.652
31	Promet-Traffic & Transportation	0.456
32	PROCEEDINGS OF THE INSTITUTION OF CIVIL ENGINEERS-TRANSPORT	0.239
33	ITE JOURNAL-INSTITUTE OF TRANSPORTATION ENGINEERS	0.2
34	International Journal of Heavy Vehicle Systems	0.105

19 GCN:基于图卷积神经网络的共享单车流量预测

19.1 文章信息

《Bike Flow Prediction with Multi-Graph Convolutional Networks》。

文章来自 2018 年第 26 届 ACM 空间地理信息系统进展国际会议论文集，作者

来自香港科技大学，被引 7 次。

19.2 摘要

由于单站点流量预测的难度较大，近年来的研究多根据站点类别进行流量预测。但是，它们不能直接指导站点级的精细化系统管理。本文将应用深度学习技术进行站点级别的共享单车流量预测。本文提出了一种多图卷积神经网络模型来预测站点级的流量，其关键创新之处在于从图的角度来观察共享单车系统。更具体地说，我们为一个自行车共享系统构造了多个图来反映异构的站间关系。然后，我们融合了多个图形，并应用卷积层来预测未来的站点级自行车流量。

19.3 简介

本文所述的共享单车系统为有桩共享单车。单站点自行车流量往往具有复杂的动态规律，这使得其预测变得困难。因此，既有研究大多根据站点类别解决自行车流量预测问题。也就是说，他们首先将站点分类，然后预测每个类别的自行车流量。但基于类别的预测有两个问题：（1）：聚类结果很难评估是否合适；（2）：基于类别的预测结果不能进行站点级别的运营管理指导。

本文中，我们提出了一种新的多图卷积神经网络来捕捉不同站点之间的空间关系，例如距离和历史使用关系。在多图卷积层的基础上，提出了一种包含 LSTM 的 encoder-decoder 编码解码器结构，用来捕捉时间关系。因此，对于站点级自行车流量预测，可以有效地捕捉时空模式。

据我们所知，这是首次利用多图卷积神经网络来进行站点级别的共享自行车系统中流量预测。To the best of our knowledge, this is the first work of leveraging multi-graph convolutional neural networks in to predict station-level bike flow in a bike sharing system. 本文案例为纽约和芝加哥的共享单车系统。

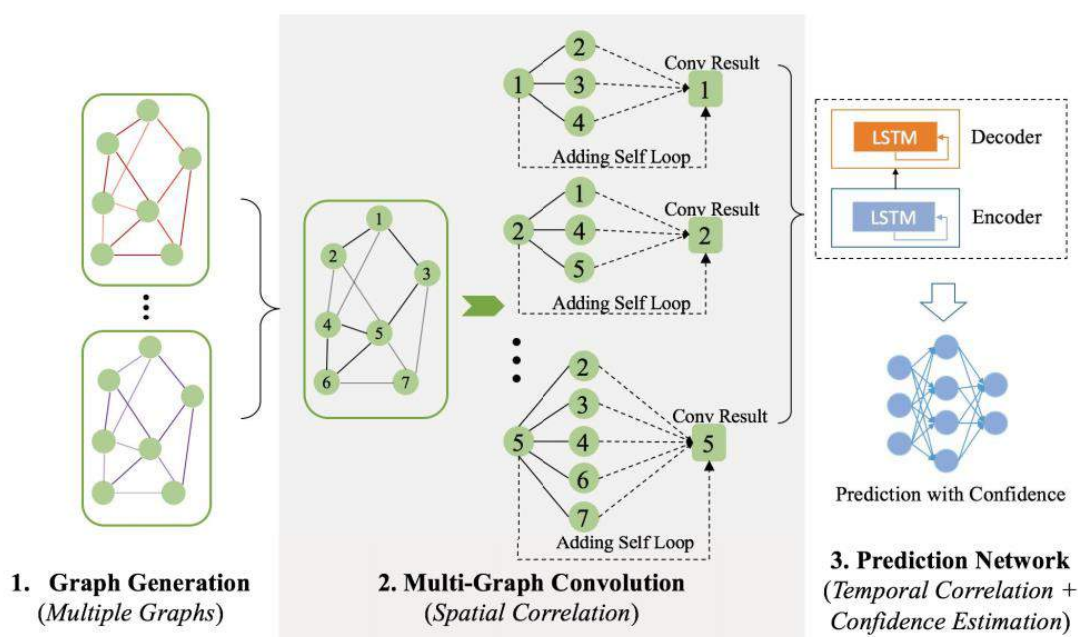
19.4 概念定义和问题描述

共享单车系统图:共享单车系统用加权图表示,节点为站点,边为站点间关系。边的权值表示站与站之间的关系强度。通常,权重越大,两个站点的相关性越高。如何构造图是我们方法的关键部分。

自行车流量:有两种类型的自行车流量:流入和流出 **inflow and outflow**。

问题描述: 利用 t 时刻以前的进站流和出站流预测 t 时刻的进站流和出站流。

19.5 主体内容



本文提出的多图卷积神经网络包含三部分,图生成-多图卷积-预测部分。

图生成:图的生成是图卷积模型成功的关键,如果所构建的图不能对站点间的有效关系进行编码,不仅不利于网络参数的学习,还会

降低预测性能。一般来说，我们希望为具有相似动态流模式的站点之间的边缘分配较大的权重。在此基础上，提出了建立站间图的三种方法:距离图、交互图和相关图。

权重 1: 距离图，即用距离的倒数表示站间权重。

$$G_d(V, E) \quad \text{weight} = \text{Distance}^{-1}$$

权重 2: 交互图，即用历史数据中两个车站之间的行车记录数表示站间权重。

$$G_i(V, E) \quad \text{weight} = \# \text{RidingRecordNumber}$$

权重 3: 相关图，计算每个站在每个时间间隔(如 1 小时)内的历史使用情况(流入或流出)，然后计算每两个站之间的相关性作为图中站间权重。本文使用皮尔逊系数来计算相关性。

$$G_c(V, E) \quad \text{weight} = \text{Correlation}$$

多图卷积：此部分可分为两块主要内容：图融合以及图卷积。

图融合即将上面的三个图权重矩阵融合为一个图权重。我们首先对每个图的邻接矩阵 A 进行标准化，然后通过元素级对不同图的邻接矩阵加权求和来组合不同的图。

$$A' = D^{-1}A + I$$

where D is :

$$D = \begin{pmatrix} \sum_{j=0}^{N-1} A_{0,j} & 0 & \dots & 0 \\ 0 & \sum_{j=0}^{N-1} A_{1,j} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sum_{j=0}^{N-1} A_{N-1,j} \end{pmatrix}$$

A 为邻接矩阵， D 为度矩阵， A' 为标准化后的临界矩阵。这一块其实更常用的方法如下图：

$$f(H^{(l)}, A) = \sigma \left(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right),$$

with $\hat{A} = A + I$, where I is the identity matrix and \hat{D} is the diagonal node degree matrix of \hat{A} .

为了使加权和运算后的融合结果保持归一化，我们在加权矩阵中进一步增加了一个 softmax 运算。假设我们有 N 个图要混合在一起（即 W_1 至 W_n 代表一条边的 N 个权重），我们可以把图的融合过程表示为：

$$W'_1, W'_2, \dots, W'_N = \text{Softmax}(W_1, W_2, \dots, W_N)$$

$$F = \sum_{i=1}^N W'_i \circ A'_i$$

where \circ is the element-wise product, F is the graph fusion result which will be used in the graph convolution part.

\circ 代表元素积， F 为融合结果，用于后面的图卷积。（上述过程总结起来就是先把邻接矩阵标准化，然后把 N 个权重矩阵进行 softmax 运算，再将两者进行加权相乘得到融合结果）

图卷积过程即为如下过程：

$$H_0^{t'} = (I^{t'}, O^{t'}), t' \in [0, t-1]$$

$$H_1^{t'} = F * W_c * H_0^{t'}$$

where W_c is the convolution weight matrix, $H_0^{t'}$ is the bike flow at time t' . We take $H_1^{t'}$ as the convolution result, and then use $H_1^{t'}$ as the input of the next prediction network.

预测网络图如下：

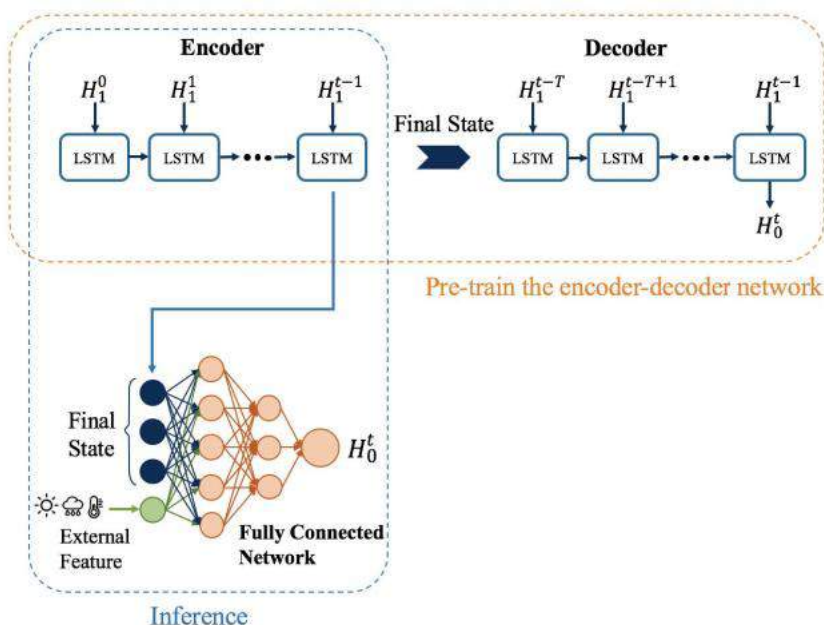


Figure 2: Structure of Prediction Network

采用 encoder-decoder 结构的预测网络详细如图 2 所示。编码器网络采用多图卷积结果序列 $[H_{10}, H_{11}, \dots, H_{1t-1}]$ 作为输入，将时间模式编码为 final 状态。解码器网络以编码器的 final 状态为初始状态，以多图卷积结果序列 $[H_{1t-T}, H_{1t-T}+1, \dots, H_{1t-1}]$ 作为输入（此处有点难理解，可理解为假设 encode 部分的输入为 10 个时间步，则 decoder 部分的输入为 5 个时间步，两者的输入都是多图卷积结果，只不过输入的序列长度不一样）。解码器的输出为 H_{0t} ，即为预测目标。我们可以将 T 设置为一个小的值(例如 t 的一半)，这意味着解码器可以根据短时间的历史数据和编码器的 final 状态预测未来的自行车流量。这也意味着编码器的 final 状

态为预测过程提供了重要的信息。在对编解码器结构进行预训练后，将编码器网络的 final 状态与外部因素(例如，温度，风速，工作日/周末)进行结合，输入到一个全连接层(图 2 的下部)预测自行车下次 HOt 流动。（先利用黄框对 encoder-decoder 结构进行预训练，提取 encoder 部分，利用蓝框作为最终的预测网络模型）

19.6 案例研究

本文的案例选用纽约和芝加哥的共享单车数据集，不再详述（目前公开的是从 2013 到 2019 年五年的数据）。公众号后台回复“纽约”获取纽约数据集下载地址，回复“芝加哥”获取芝加哥数据集下载地址。

20 GCN：基于图卷积神经网络的轨道交通流量预测

20.1 文章信息

《Predicting Station-Level Short-Term Passenger Flow in a Citywide Metro Network Using Spatiotemporal Graph Convolutional Neural Networks》。

2019 年中国海洋大学的几位老师发在 IISPRS International Journal of Geo-Information 上的一篇文章（地学 4 区，IF:1.84）。

20.2 摘要

本文提出了一种基于深度学习的方法 STGCNNmetro (spatial - temporal graph convolutional neural networks for metro)，用于同时预测城市全网每个地铁站的进站流量和出站流量。具体地说，STGCNNmetro 不再使用网格表示地铁站，而是将城市地铁网络转换成一张图，并使用图卷积神经网络(graph convolutional neural networks, GCNNs)进行预测。首先，我们应用立体图卷积运算来无缝地捕捉沿地铁网络的不规则时空依赖关系。其次，构造了一个由 GCNNs 组成的深层结构来捕捉城市层面上的时空依赖关系。最后，我们整合三个时间模式(最近的、每天的和每周的)，并融合从这些模式中捕获的时空依赖关系，形成最终的预测值。

STGCNNmetro 模型是一个端到端框架，可以接受原始客流量数据，自动捕获全市地铁网络的有效特征，并输出预测。通过对中国上海城市轨道交通短期客流量的预测，验证了该模型。实验表明，STGCNNmetro 模型优于 7 个已知的基线模型 (LSVR、PCA - kNN、NMF - kNN、Bayesian、MLR、M - CNN 和 LSTM)。此外，我们还探讨了模型对其参数的敏感性，并讨论了预测误差的分布。

20.3 简介

整个城市的地铁网络可以表示为一个图，其中地铁站为顶点，地铁线路为边。每个站点顶点都有一个由客流历史值组成的特征向量，可以定义一个邻接矩阵来编码站点之间的成对依赖关系。因此，地铁网络不需要用网格来表示地铁站，也不需要用 CNN 来捕捉特征，而是可以用一个通用的图形来描述，利用 GCNN 可以有效地捕捉地铁网络层面而不是网格层面上不规则的时空依赖关系。

20.4 理论介绍

(1) 用图表示地铁网络客流量的时间序列

根据前面 M 次观测，我们可以预测后续时间步长的客流量，在这项工作中，我们在一个图表上定义了一个城市范围的地铁网络，并关注结构化的时间序列客流量。如图 1 所示， V ， E ， W 分别代表顶点，边和加权邻接矩阵。其中顶点 V_t 代表 t 时刻所有地铁车站的客流量向量。

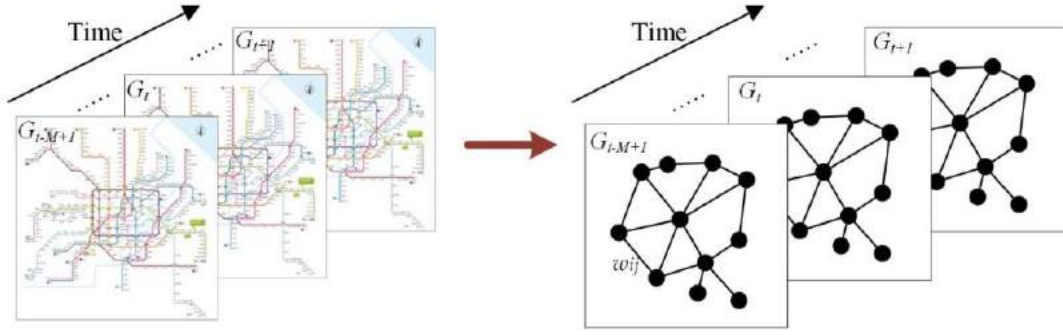


Figure 1. Graph-structured metro passenger flow-volume data. At the time step t , in the graph $G_t = (V_t, E, W)$, V_t is an observation vector of n metro stations at time step t , each element of which records the historical passenger flow volume for a single metro station; E is a set of edges, indicating the connectedness between stations; while $W \in R^{n \times n}$ denotes the weighted adjacency matrix of G_t .

(2) 图卷积操作

常规网格的标准卷积显然不适用于一般图。目前研究如何将 CNNs 推广到结构化数据形式的一种基本方法是利用图的傅里叶变换在频域中进行操作。其介绍了将卷积应用于光谱域的光谱框架，即通常所说的谱图卷积。一些后续研究通过将计算复杂度从 $O(n^2)$ 降低到线性成本，使得图卷积更加有效。本文使用的图卷积如下：

$$y = \theta * g x = \sigma(\sum_{j=0}^K \alpha_j L^j x),$$

图卷积算子 “ $*$ ” 是谱图卷积的概念，表示图信号 x 与核函数 Θ 的乘积。 K 表示从中心节点确定卷积的最大半径， α 卷积核参数, L 是归一化的图拉普拉斯算子， σ 是激活函数(例如 ReLU)。（搞明白这个公式推荐阅读 Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering）

(3) 利用立体图卷积来捕捉不规则的时空依赖关系

在构建了上一节所述的图结构地铁网络数据之后，我们使用 GCNN 提取了地铁网络的局部不规则时空依赖关系。以顶点 i 为例，图 2 给出了核大小为 1 的传统图卷积($K=1$)。实际上，这是用一阶邻接进行的卷积计算。显然，顶点 i 与其 5 个一阶相邻顶点之间的空间依赖关系捕获(在图 2 中标记为 1、2、3、4 和 5)。该操作要对所有顶点执行。针对地铁网络客流量时间前 M 个时间段的序列数据，我们构造了一个立体卷积核来无缝提取时空相关性。 M 个历史流量图被叠加在时间维度

上。在立体图卷积中，卷积在空间和时间维度上都是无缝执行的，如图 3 所示。定义了时空图卷积算子“ $*g'$ ”为图信号 x' 与立体核函数 Θ' 的乘积。对于顶点 i ，立体图卷积之后，能够捕获其一阶空间邻接顶点及其 M 阶时间邻接中包含的不规则时空依赖关系。

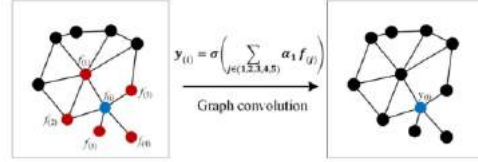


Figure 2. Procedure for capturing spatial dependency by graph convolution for $K=1$ (using only one vertex as an example). K is the kernel size of the graph convolution, which determines the maximum radius of the convolution from central nodes, α_1 is the convolution kernel parameter, σ is an activation function (i.e., rectified linear unit (ReLU)). $y_{(i)}$ indicates the passenger flow volume of vertex i to be predicted, and each $f_{(j)}$ indicates the historical passenger flow volume of vertex j .

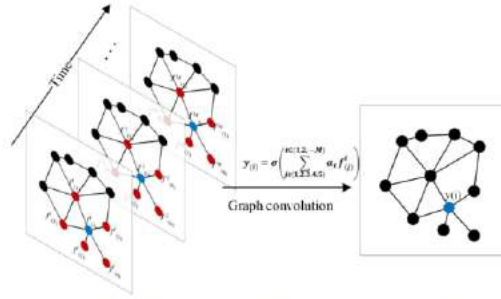


Figure 3. Procedure for capturing spatiotemporal dependencies by graph convolution for $K=1$ (using only one vertex as an example). α_t is the convolution kernel parameter at the time step t , and $f_{(j)}^t$ indicates the historical passenger flow volume of vertex j at the time step t .

(4) 使用深度 GCNNs 捕捉全网层面遥远的时空依赖关系

在地铁网络结构中，高阶邻接可以通过低阶邻接累积。例如，二阶邻接可以通过积累两个一阶邻接来获得。通常，一个城市地铁网络的空间规模很大，从市中心到郊区有很多车站。从直观上看，相邻车站之间的客流量可能会相互影响。这可以由一个 GCNN 层有效地处理，该层显示了强大的分层捕获空间结构信息的能力。此外，由于地铁系统连接两个相距很远的位置，这导致了遥远的车站之间的时空依赖关系。由于一个卷积层只考虑位置相近的站之间的依赖关系，为了捕获距离较远的站之间的时空依赖关系，需要叠加多个 GCNN 层来形成一个深度 GCNN 结构。

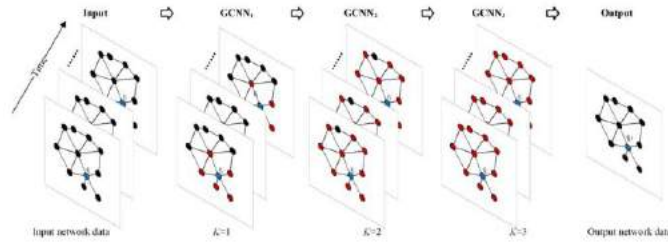


Figure 4. Procedure for capturing the spatiotemporal dependencies of metro network data using graph convolutional neural networks (GCNNs) (taking Station 6 as an example).

(5) 利用时空 GCNNs 预测全市地铁车站短期客流量

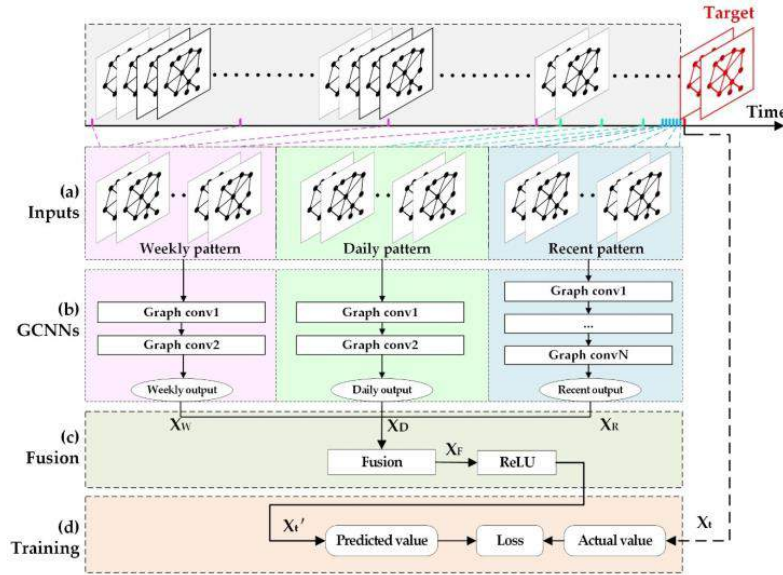


Figure 5. Architecture of the STGCNNmetro model. Conv: Convolution; X_t and X_t' represent actual and predicted values in the t th time interval, respectively; X_R , X_D , and X_W represent the outputs of the recent, daily, and weekly patterns, respectively; X_F is the fusion result of X_R , X_D , and X_W using a parametric-matrix-based method.

提出的基于深度学习新模型 STGCNNmetro 的总体框架如图 5 所示。显然，地铁系统的客流包括进出站。在本研究中，进站流量为给定时间间隔内进站的总客流量，出站流量为出站的总客流量。我们首先将历史上的全市地铁进站流量图和出站流量图在每个时间间隔上转换为结构化的时空时间序列。由于时空依赖具有不同的模式，历史流量时间序列分为近期、每日和每周模式。然后，构建了三个基于 GCNNs 的子模型来捕捉这三种模式的特征。在参数矩阵的基础上，对三种输出进行融合，对不同车站不同模式的分析结果赋予不同的权重。最后，选择 ReLU 函数激活融合结果，给出最终的预测值。

其他部分不做解释。

20.5 特别之处

文章画的图很漂亮，另外，文章的结果讨论特别详细。

21 OD 需求预测：基于时空网络的出租车 OD 需求预测

21.1 文章信息

《Contextualized Spatial–Temporal Network for Taxi origin–Destination Demand Prediction》。

2019 发在 IEEE Transactions on Intelligent Transportation Systems 上的一篇文章（运输科技 2 区，IF:5.744）。

21.2 摘要

目前大量的研究是基于出行起点的出行需求预测，很少有人关注这些人前往何处，即 D 点，本文认为仅根据出租车的原产地需求将出租车预先分配到每个区域并不是最优的。因此本文主要是进行所有区域之间即所有 OD 对之间的出租车 OD 需求预测。它的主要挑战来自于如何有效地捕获各种环境信息来学习需求模式。本文提出了一种新的符合实际环境的时空网络(contextualized spatial–temporal network, CSTN)，它由局部空间环境(local spatial context, LSC)、时间演化环境(temporal evolution context TEC)和全局相关环境(global correlation context, GCC)三部分组成。（context 这个词，网上大部分都是翻译成上下文，我感觉好像环境更好一点，本文所有 context 的翻译均为环境）。首先，局部空间环境模块利用两个卷积神经网络分别从起点和终点两个角度学习局部空间依赖关系。其次，将出租车需求的局部空间特征和气象信息集成到卷积长短时记忆网络(ConvLSTM)中，分析出租车需求演化。最后，为了捕获相隔较远的区域之间的相关性，全局相关环境模块计算任意两个区域之间的相似性，并将所有区域的特征与相似性权重相加，生成

每个区域的全局相关性特征。实验表明本文模型的表现最好。

21.3 简介

滴滴出行等出租车服务仍然存在着一些低效经营的问题，例如乘客等候时间较长及空载次数较多。主要是由于出租车需求预测不准确导致供需不匹配，导致大量出租车聚集在一些繁忙地区，造成供过于求，而在其他偏远地区，出租车的分布极为稀疏。解决这一问题的方法包括预测未来的出租车需求，同时提前将出租车分配到各个区域。

出租车需求预测作为智能交通系统中的一项重要任务，已经引起了广泛的研究兴趣，并取得了显著的成功。然而，现有的方法大多只对出租车在出发地点的需求进行建模，估计各个地区或某些特定地点的出租车需求，忽略了乘客目的地的影响。我们认为乘客目的地信息对于出租车预分配系统是至关重要的。出租车预分配系统在不考虑乘客目的地分布的情况下，仅根据预测的出租车发车需求提前部署出租车，可能存在以下问题：

(1) 受城市管理规定的限制(如北京的驾驶限制政策)，一些司机只允许在某些特定地区开车。如果一个出租车司机被分配到一个大部分乘客都要去的区域，而这个区域是司机被限制的地方，那么他就不能接受命令，这可能会导致资源的浪费。

(2) 一些司机更喜欢在他们熟悉的区域搭载乘客。与此同时，一些司机不愿意接受利润微薄的短途旅行订单。如果司机预分配区域内的大部分乘客的目的地不在他的操作区域，或者离接机地点太近，司机可能会拒绝这些请求。

(3) 如果一个司机被派往一个大部分乘客都会去，但是司机不熟悉的地区，即使有 GPS 导航，司机也可能会花更多的时间把乘客带到他们的目的地。这将降低出租车市场的运营效率和乘客满意度水平。

TABLE I
COMPARISON OF THE DEFINITION AND SCOPE
OF VARIOUS RELATED TASKS

Method	Task and Scope
Zhang et al. [4] Jin et al. [5]	Traffic Inflow and Outflow Prediction in all regions
Tong et al. [6] Yao et al. [7]	Taxi Demand Prediction in all regions
Toqu et al. [8] Azzouni et al. [9] Yang et al. [10]	Traffic Flow or Demand Prediction between some well-designed positions (e.g., highway toll booths, subway and bus stations)
Zhou et al. [11]	Passenger Pickup/Dropoff Demand Prediction in all regions
Ours	Taxi Demand Prediction between all regions

在本文中，我们提出了一个具有挑战性的出租车初始需求预测任务，该任务旨在预测任意两个区域之间未来的出租车需求。如果能够很好地预测出租车的发车需求和乘客的目的地，我们可以更有效地提前调配出租车，满足乘客的需求，同时避免上述问题。本文的关键在于如何捕捉不同时空的环境信息来学习需求模式。例如，一些在空间上相邻的区域通常具有相似的需求模式(比如出租车需求数量和需求趋势)，在我们的工作中称为局部空间环境(local spatial context, LSC)。此外，即使两个区域在空间上相距遥远，如果它们具有相似的功能(例如，它们都是住宅区)，需求模式可能仍然具有一定的相关性。我们将两个相距遥远的区域之间的这种关系称为全局关联环境(global correlation context, GCC)。最后，出租车需求是一个时变的过程，出租车需求的演变与出租车目前所处的状态以及不断变化的天气等多种因素有关，出租车需求的演变可以用时间演化环境(time evolution context, TEC)来表示。

本文提出了一种新的符合实际环境的时空网络，它由局部空间环境、时间演化环境和全局相关环境三部分组成。首先，局部空间环境模块利用两个卷积神经网络分别从起点和终点两个角度学习局部空间依赖关系。然后将两个网络的输出结合起来生成最终的局部空间特征，其中包含了不同视角下出租车需求模式的混合信息。其次，将出租车需求的局部空间特征和气象信息集成到卷积长短时记忆网络(ConvLSTM)中，分析出租车需求演化。最后，为了捕获相隔较远的区域之间的相关性，全局相关环境模块模块计算任意两个区域之间的相似性，并将所有区域的特征与相似性权重相加，生成每个区域的全局相关性特征。这样，每个区域包含了所有区域的信息，主要与与有高度相似性的区域相关。最后，将 TEC 模块生成的

局部时空特征与 GCC 模块生成的全局相关特征相结合，对未来出租车发车目的地需求进行预测。

这项工作的主要贡献有三方面：

(1) 将现有的出租车需求预测扩展到出租车原目的地需求预测的任务中，对智能交通系统具有更大的探索价值。据我们所知，我们是第一个研究跨区域出租车需求预测的。

(2) 提出了一种新的时空网络解决了这一问题，它很好地将局部空间环境、时间演化环境和全局相关环境集成到一个统一的框架中。

(3) 在出租车 OD 需求预测的很多基准上进行的大量实验表明，我们的方法比现有的最先进的方法有一定的优势。

21.4 预备知识

A、全网的区域划分

我们根据经纬度将一个城市划分为 $H \times W$ 不重叠网格地图。每个矩形网格代表城市中不同的地理区域。网格映射第 i 行和第 j 列上的区域在下面的部分中表示为 $R(i, j)$ 。图 1(a)显示了纽约市曼哈顿的分区区域。通过这种简单的划分方法，原始出租车请求记录可以直接转化为矩阵或张量，这是深度神经网络中最常见的输入数据格式。

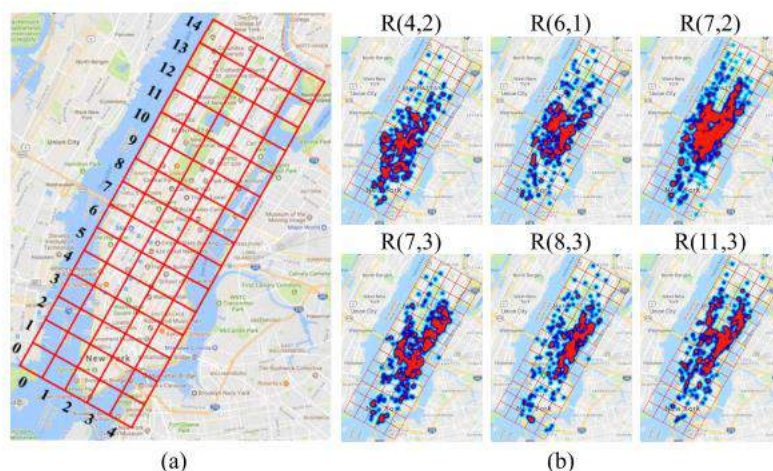


Fig. 1. (a) Illustration of the region partition on a city. We partition a city into a grid map based on the longitude and latitude. Here is the example of the Manhattan in New York City. (b) Visualization of the taxi demand from origin view by mapping the passengers' pick-up locations back to the geo-coordinates on Google map. The sub-figure with title "R(i,j)" is the taxi demand from all regions to the region R(i, j) during 8:00-8:30 am, May. 8, 2014.

B、出租车 OD 需求

OD 需求这块有点绕，需要搞明白，一个区域的需求产生量可能会前往其他所有区域（一对多的关系），而一个区域的需求吸引量可能会来自其他所有区域（多对一的关系）。

在打车行业，出租车公司或滴滴出行、优步等在线平台每秒钟都会收到大量乘客的打车请求。每个原始出租车请求包含乘客的起始位置、目的地位置、时间戳和其他信息(例如用户标识和电话号码)。在我们的工作中，出租车始发目的地需求定义为在每个时间间隔内从始发区域到目的地区域的出租车请求总数。

我们将时间区间 t 内出租车 OD 需求表示为三维矩阵 $X_t \in \mathbb{R} (N \times H \times W)$ ，其中 H 和 W 分别为城市网格地图的高度和宽度。 N 是城市区域总数，等于 $H \cdot W$ 。

（此处 $N=H \cdot W$ 是上面提及的多对一的关系，即到达一个区域的乘客可能来自其他所有区域，所以，一个区域得需要一个矩阵， N 个区域需要 $H \cdot W$ 个矩阵）

（高亮部分是一个总体表达式，是针对所有 O 点所有 D 的，公式好多，不写了直接上图……）

N is the total number of the regions in city and it is equal to $H \cdot W$. Specifically, $X_t(d, i_o, j_o)$, in which the destination index d is equal to $W \cdot i_d + j_d$, is the demand from origin region $R(i_o, j_o)$ to destination region $R(i_d, j_d)$. The value of $X_t(d, i_o, j_o)$ can be measured from the taxi request records in time interval t . In particular, the d^{th} channel of X_t , denoted as $X_t(d) \in R^{H \times W}$, is the taxi demand from all regions to region $R(i_d, j_d)$. Figure 1(b) shows some channels of X_t by mapping the passengers' pick-up locations back to the geo-coordinate on Google map. The taxi origin demand, denoted as $O_t \in R^{H \times W}$, can be easily calculated by $\sum_{d=0}^{N-1} X_t(d)$.

C、出租车 OD 需求预测

出租车 OD 需求预测问题是给定 t 时刻之前的数据，预测时间间隔 t 内的出租车 OD 需求。如图 2 所示，出租车需求受到气象条件的严重影响，因此我们也加入了历史气象数据来处理这个任务，我们将时间间隔 i 中的气象数据表示为 M_i 。

our final goal is to predict X_t with the historical demand data $\{X_i | i = t - n + 1, \dots, t\}$ and meteorological data $\{M_i | i = t - n + 1, \dots, t\}$, where n is the sequence length of time intervals.

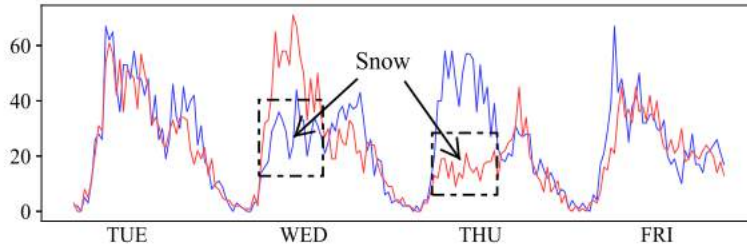
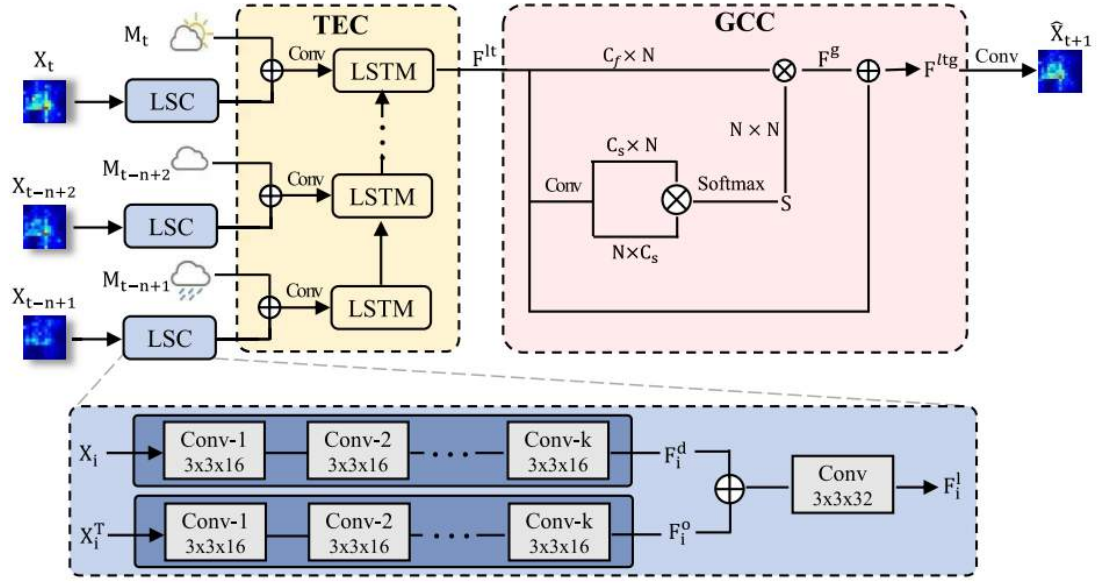


Fig. 2. Influence of meteorological conditions on taxi demand. We show the taxi demand from region(5, 2) to region (7, 2) of New York City in two time periods: Feb 4-7 (Red) and Feb 11-14 (Blue) in 2014. We can see that the heavy snow sharply reduces the taxi demand compared to the same day of the adjacent week.

21.5 模型简介

下面对模型的三个部分（蓝框、黄框、红框）依次进行介绍。



A、Local Spatial Context Modeling 局部空间环境

一般情况下，出租车的需求往往与当地的空間位置有关，空間相邻区域可能具有相似的需求模式。例如，人们倾向于在早高峰时段离开居住地前往就业地区。在这种情况下，城市郊区的大部分居住区域都有较高的O点需求，而城市中心的大部分工作区域都有较高的D点需求。反之亦然。近年来，研究者利用卷积层对出租车O点需求的局部空间环境进行了建模，但忽略了D点的需求。

在这项工作中，我们提出的LSC模块同时从O点视角和D点视角捕获出租车需求的局部空间信息。相当于利用了两个张量，一个是OD矩阵，一个是DO矩阵。下图说明了从OD矩阵转换为DO矩阵的过程。

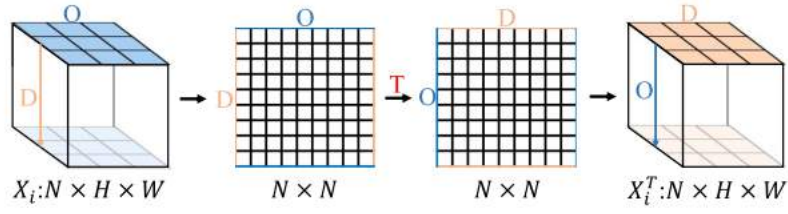
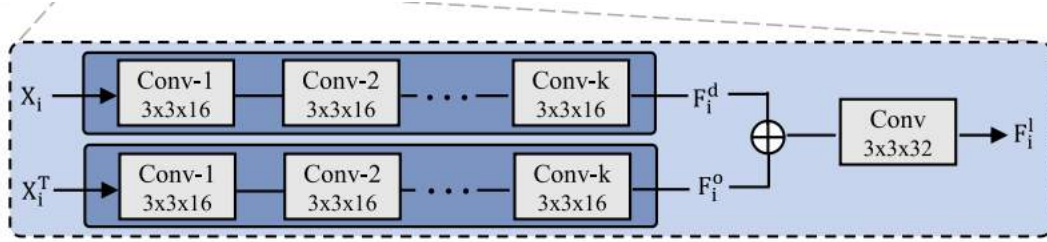


Fig. 4. Illustration of the generation process of **DO matrix** from **OD matrix**. N is the total number of regions and it is equal to $H \times W$. T denotes the matrix transposition. Each channel of the OD matrix X_i is the taxi demand from all origin regions to the corresponding region. X_i^T is called the DO matrix in our work and each channel denotes the taxi demand from the corresponding region to all destination regions.

对两个矩阵分别进行卷积在进行特征融合即可，不再详述。文章将下图称为 ConvNets。



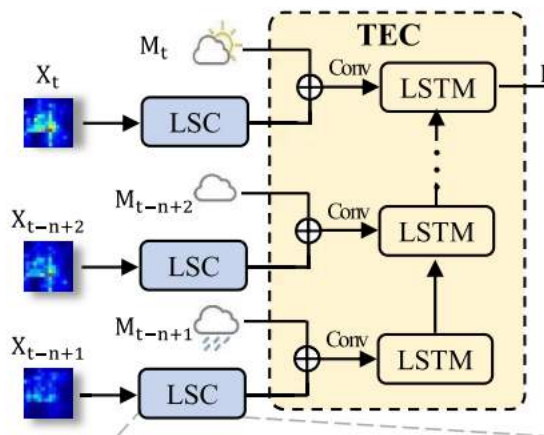
B、Temporal Evolution Context Modeling 时间演化环境

出租车需求是一个时变的过程，通常受多种复杂因素的影响。除了自身的内部状态，气象条件也影响着未来的需求。例如，持续的降雪可能会严重削弱居民的出行意愿，导致出行需求下降。因此，我们结合历史需求特征和不断变化的气象条件，把握出租车需求沿时间维度的演变趋势。

这一块也比较简单，即将第一部分的卷积结果和对天气数据的卷积结果进行融合，然后输入到 ConvLSTM 层即可。（这一块文章中写的比较复杂，把 ConvLSTM 的来龙去脉和公式都详细解释了一遍）

我们的目标是利用过去 n 个时间间隔的历史需求和气象条件来预测 X_t 的出租车需求。对于气象数据 M_i ，我们用多层感知器(MLP)对其进行编码，多层感知器由三个堆叠的全连接层实现，分别由 64 个、16 个和 8 个神经元组成。然后将 MLP 的输出特征复制 $H \cdot W$ 次，构建 3D 气象特征，以便与上一部分的输出结果进行融

合。



C、Global Correlation Context Modeling 全局相关环境

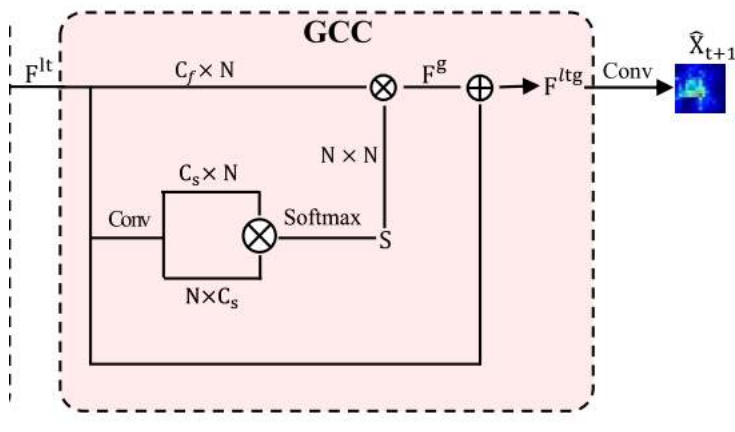
在上述两个模块中，ConvNets 和 ConvLSTM 只捕获了出租车需求的局部信息。然而，出租车需求分布也与区域属性有关，例如，城市不同区域的大部分居住区在早高峰时段可能会有较高的出租车需求。因此，即使两个区域相距遥远，只要两个区域的属性一致，它们的出租车需求模式也可能是相似的。我们将这种关联称为全局相关环境。

本文通过全局特征融合操作，获取各区域间的全局相关性。具体来说，我们将每个区域的全局相关特征表示为所有区域特征的加权和，权重计算为对应 OD 对之间的相似性。这样，每个区域包含了所有区域的信息，主要与与之高度相似的区域相关。

这一块比较复杂（可解释性也不太强），具体操作就是，先将 B 部分的输出经过一个有 C_s 个 filter 的 Conv 层，输出形状为 $C_s \times H \times W$ ，再将该输出 reshape 成 $C_s \times N$ ，然后转置为 $N \times C_s$ ，两者点成得到 $N \times N$ 的相似性矩阵 S 。

将 B 部分的输出 reshape 成 $C_f \times N$ ，并与相似性矩阵相乘，结果形状为 $C_f \times N$ ，再将该结果 reshape 成 $C_f \times H \times W$ 。该结果（即下图中的 F_g ）对全局相关环境进行编码，但这样缺乏结构性的局部特征，这会导致性能下降，因此再将该结果与 B 部分的输出连接，最后经过一个 Conv 层，得到最终的输出结果。该特征 F_{ltg} 融合了

局部空间信息、时间演化信息和全局相关信息的混合信息。



21.6 总结

文章后面的实验部分不再介绍。文章总体来说很新颖，尤其是全局空间相似性这块个人感觉可以借鉴学习。后台回复“纽约”获取文章的使用的纽约出租车数据集。

22 Multi-GCN：基于时空多图卷积网络的网约车需求预测

22.1 文章信息

《Spatiotemporal Multi-Graph Convolution Network for Ride-hailing Demand Forecasting》。

2019 AAAI 大会的一篇论文，被引 5 次。

22.2 摘要

区域级需求预测是网约车服务的一项重要任务。区域间具有复杂的时空依赖性，现有的方法主要侧重于对空间相邻区域之间的欧式相关关系进行建模，而我们

发现，距离可能较远的区域之间的非欧式相关关系对于准确预测也至关重要。为此本文提出了时空多图卷积网络(ST-MGCN)。首先将区域间的非欧式成对相关关系编码为多个图，然后利用多图卷积对这些相关关系进行建模。为了利用全局上下文信息对时间相关性进行建模，我们进一步提出了上下文门控循环神经网络，该网络利用上下文感知的门控机制对不同的历史观测值进行重新加权。我们在两个真实的大型网约车需求数据集上评估了所提出的模型，与最先进的基准模型相比，该模型提高了 10% 以上。

22.3 简介

区域级网约车需求预测的目标是根据历史观测，预测城市区域未来的需求。这一任务的挑战性主要在于复杂的时空相关性。一方面，不同区域之间存在复杂的依赖关系。例如，一个区域的需求通常受其空间相邻邻居的影响，同时与具有相似上下文环境的遥远区域相关。另一方面，不同的时间观测值之间也存在非线性依赖关系。对某一时刻的预测通常与各种历史观测结果相关联，如一小时前、一天前甚至一周前。

在时空相关性建模中，有两个重要的方面被忽略了。首先，这些方法主要针对不同区域间的欧氏相关关系进行建模，但我们发现非欧氏两两相关关系对准确预测也至关重要。图 1 显示了一个示例。对于区域 1，除了邻域 2，它还可能与一个遥远的区域 3 相关具有相似功能的，即，它们都在学校和医院附近。此外，区域 1 还可能受到区域 4 的影响，区域 4 通过公路与区域 1 直接相连。其次，在这些方法中，当使用 RNN 对时间相关性进行建模时，每个区域都是独立处理的，或者只基于局部信息。然而，我们认为全局和上下文信息也很重要。例如，全局网约车需求的增加/减少通常表明一些事件的发生，这些事件将影响未来的需求。

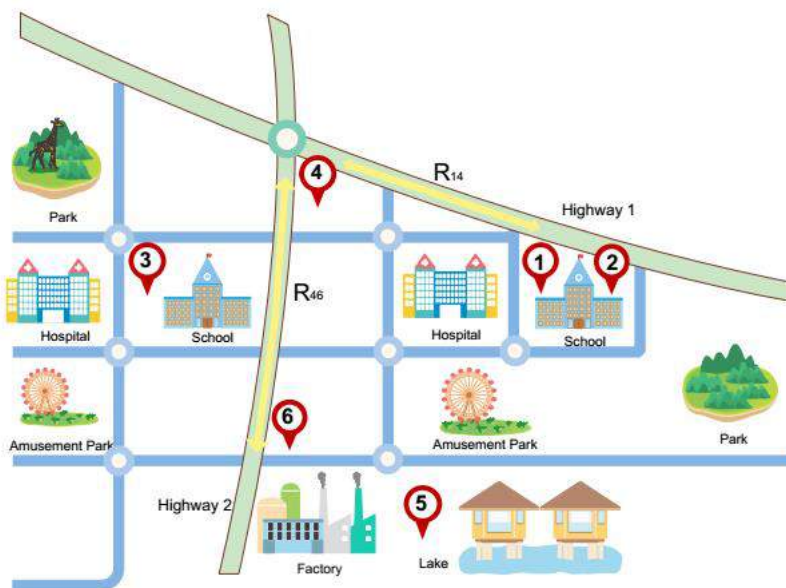


Figure 1: An example of different correlations among regions. To predict the demand in region 1, spatially adjacent region 2, functionality similar region 3 and transportation connected region 4 are considered more important, while distant and irrelevant regions 5 are less relevant.

为了解决这些问题，我们提出了一种新的深度学习模型，称为时空多图卷积网络(ST-MGCN)。主要贡献如下：

在网约车需求预测中，我们识别出区域间的非欧式相关关系，并提出使用多个图对其进行编码。然后，我们进一步利用提出的多图卷积来显式地建模这些相关性。

提出了基于上下文门控的 RNN (CGRNN)模型，在时序依赖关系建模时引入全局上下文信息。

22.4 文献综述

(1) 城市计算中的时空预测

非欧式结构化数据也存在于城市计算中。通常，基于站点的预测任务，比如基于站点的出租车需求预测，基于站点的自行车流量预测，是非欧式的，因为数据格

式不再是矩阵，卷积神经网络变得不那么有用了。人工特征工程或图卷积网络是处理非欧式结构数据的最新技术。与以往的研究不同，ST-MGCN 将区域间的成对关系编码为语义图。虽然 ST-MGCN 是为基于区域的预测而设计的，但区域间关系的不规则性使其成为非欧式数据的预测问题。然而，这些方法都没有明确地对区域间的非欧式成对关系建模。在这项工作中，ST-MGCN 利用提出的多图卷积融合相关区域的特征，能够从不同角度关联区域的需求值进行预测。

(2) 图卷积

$$L = I - D^{-1/2} A D^{-1/2}$$

$$X_{l+1} = \sigma\left(\sum_{k=0}^{K-1} \alpha_k L^k X_l\right),$$

(3) Channel-wise attention 对 channel 做注意力机制

Channel-wise attention 提出的初衷是学习每个 channel 的权重，以便找到最重要的 channel，并通过给出更高的权重来强调它们。

$$\begin{aligned} z_c &= F_{pool}(X_{:, :, c}) = \frac{1}{WH} \sum_{i=0}^W \sum_{j=0}^H X_{i,j,c} \text{ for } c = 1, 2, \dots, C \\ s &= \sigma(W_2 \delta(W_1 z)) \\ \tilde{X}_{:, :, c} &= X_{:, :, c} \circ s_c \text{ for } c = 1, 2, \dots, C \end{aligned} \quad (1)$$

Fpool 是一个全局平均池化操作，它将每个通道总结为一个标量 z_c ，其中 c 是通道索引。然后利用注意力运算对向量 z 进行非线性变换，生成自适应权值 s ， W_1 和 W_2 相应的权重， δ σ 是分别 ReLU 和 sigmoid 函数。然后，通过 channel 方向的点积将权值 s 应用于输入。最后，基于学习权值对输入通道进行缩放。在本文中，我们采用了 Channel-wise attention 的思想，并将其推广到一系列图之间的时间依赖关系建模中。

22.5 主体内容

（一）区域级别的网约车需求预测

将全网需求网格化，利用过去 T 个时段预测 $T+1$ 个时段。所提出的模型 ST-MGCN 的系统架构如图 2 所示。我们将区域间三种关系类型表示为多个图，图的顶点表示区域，边表示区域间的成对关系。首先，时间上，我们使用提出的上下文门控循环神经网络(CGRNN)考虑全局上下文信息，对不同时间的观测结果进行汇总，将多个时间步融合为一个时间步。然后，空间上，应用多图卷积来捕捉区域间不同类型的相关性。最后，利用全连接层进行特征融合转化为预测。

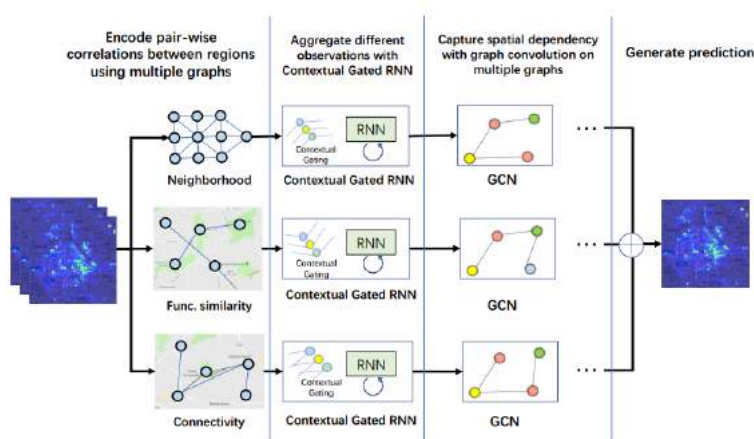


Figure 2: System architecture of the proposed *spatiotemporal multi-graph convolution network* (ST-MGCN). We en-

（二）空间依赖关系建模

所谓多图卷积，是指对空间上的多种相关关系分别进行图卷积，再进行特征融合。

在本节中，我们将展示如何使用多个图来编码区域之间不同类型的相关性，以及如何使用所提出的多图卷积来对这些关系建模。我们用图对区域间的三种相关性进行了建模，包括(1)邻域图，编码空间邻近性，(2)区域功能相似图，它对区域周围兴趣点的相似性进行编码，(3)交通连通性图，它编码了遥远地区之间的连通性。

（1）邻域图

区域的邻域是根据空间邻近性来定义的。我们在一个 3×3 的网格中，将一个

区域与其相邻的 8 个区域连接起来，构造出图形。

$$A_{N,ij} = \begin{cases} 1, & v_i \text{ and } v_j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

(2) 功能相似图

在对一个区域进行预测时，可以直观地参考与此区域功能类似的其他区域。区域功能可以用其周围每个类别的 POIs 来表征，两个顶点(区域)之间的边定义为 POI 相似性：

$$A_{S,i,j} = \text{sim}(P_{v_i}, P_{v_j}) \in [0, 1] \quad (3)$$

P_{v_i} 和 P_{v_j} 分别为区域 v_i 和 v_j 的 POI 向量，其维数等于 POI 类别的数量，每个条目表示区域内特定 POI 类别的数量。

(3) 交通连通性图

在进行时空预测时，交通系统也是一个重要因素。直观地说，那些地理上遥远但方便到达的区域可以相互关联。这些连接是由高速公路、高速公路或地铁等公共交通诱导的。这里，我们将这些道路直接相连的区域定义为“连通”，对应的边定义为：

$$A_{C,i,j} = \max(0, \text{conn}(v_i, v_j) - A_{N,i,j}) \in \{0, 1\} \quad (4)$$

为 v_i 与 v_j 之间连通性的指标函数。注意，为了避免冗余关联，邻域图被从连通图中删除，这也会导致稀疏图。

(4) 利用多图卷积对时空依赖关系进行建模

利用构建的上述三种类型的图，我们使用如下公式进行卷积， $f(A; \theta_i)$ 表示拉普拉斯矩阵。

$$\mathbf{X}_{l+1} = \sigma \left(\bigsqcup_{\mathbf{A} \in \mathbb{A}} f(\mathbf{A}; \theta_i) \mathbf{X}_l \mathbf{W}_l \right) \quad (5)$$

where $\mathbf{X}_l \in \mathbb{R}^{|V| \times P_l}$, $\mathbf{X}_{l+1} \in \mathbb{R}^{|V| \times P_{l+1}}$ are the feature vectors of $|V|$ regions in layer l and $l+1$ respectively. σ denotes the activation function, and \bigsqcup denotes the aggregation function, e.g., sum, max, average etc. \mathbb{A} denotes the set of graphs, and $f(\mathbf{A}; \theta_i) \in \mathbb{R}^{|V| \times |V|}$ represents the aggregation matrix of different samples based on graph $\mathbf{A} \in \mathbb{A}$ parameterized by θ_i , while $\mathbf{W}_l \in \mathbb{R}^{P_l \times P_{l+1}}$ denotes the feature transformation matrix, For example, if $f(\mathbf{A}; \theta_i)$ is the poly-

在实际过程中, $f(\mathbf{A}; \theta_i)$ 选为拉普拉斯算子 L 的 K 阶多项式函数, 图 3 显示了图卷积层示例。 K 表示 v_i 通过 K 步能够到达 v_j 。在卷积运算方面, k 定义了空间特征提取时感知域的大小。

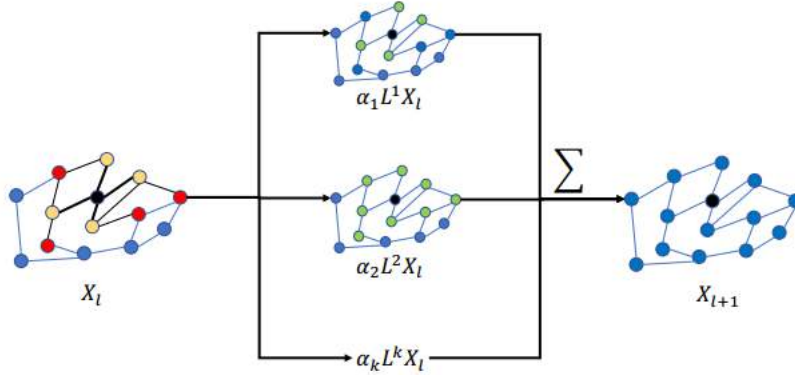


Figure 3: An example of the ChebNet graph convolution centralized at the black vertex. Left: The centralized region is marked black. The one-hop neighbors are marked yellow, while the two-hop neighbors are marked red. Middle: with the increase of degree of the graph Laplacian, the reception field grows (marks green). Right: The output of this layer is a sum among graph transformations with degree value from 1 to K .

基于多图卷积的空间依赖关系建模并不局限于上述三种类型的区域关系, 它可以很容易地扩展到其他区域关系以及其他时空预测问题的建模。它通过区域间的关

系提取特征，对空间依赖关系进行建模。在 K 较小的情况下，特征提取将重点放在较近的区域，增加 K 值或叠加多个卷积层的最大程度将增加感知域，从而鼓励模型捕获更多的全局依赖关系。

（三）时间相关的建模

即通过多步处理，将多个时间步的信息融合到一张图上。这一块比较难理解，首先解释一下维度关系。对于图 4 左上角的 $T \times |V| \times P$ ， T 代表 T 个时间步，其中每个时间步为一个方块，该方块的高即 V 代表所有的节点，宽即 P 代表每个节点 graph embedding 出来的高维表示。

图 4 左下角和图 4 左上角其实是一样的数据，只不过，左下角的数据是对每个时间步加了权重，所以他画的有的方块厚有的方块薄，即所谓的 channel-wise attention，而其中所使用的权重，便是根据 channel-wise attention 中的公式 6、7、8 得到的，比较复杂（公式 6 图卷积，公式 7 对图卷积结果进行池化，池化结果利用公式 8 得到权重）。

图 4 右边他说的共享权重的 RNN，其实就是将多张图经过 RNN 融合成一张图。按道理，对于每个节点，使用过去 T 个时间步预测未来 $T+1$ 个时间步，应该训练一个 RNN，50 个节点要训练 50 个 RNN，但是文章为了简化模型，对所有节点应用一个 RNN 模型，右边中间的黑色方块，是把 RNN 展开了，代表 T 个时间步，最后输出一个时间步，即融合为一张图。（其实共享权重的 RNN 效果肯定是不如对每个节点训练一个 RNN 效果好的，但是文章中说了这样做的原因：区域间共享 RNN 是为所有区域找到一个通用的聚合规则，从而促进模型的泛化，降低模型的复杂度）

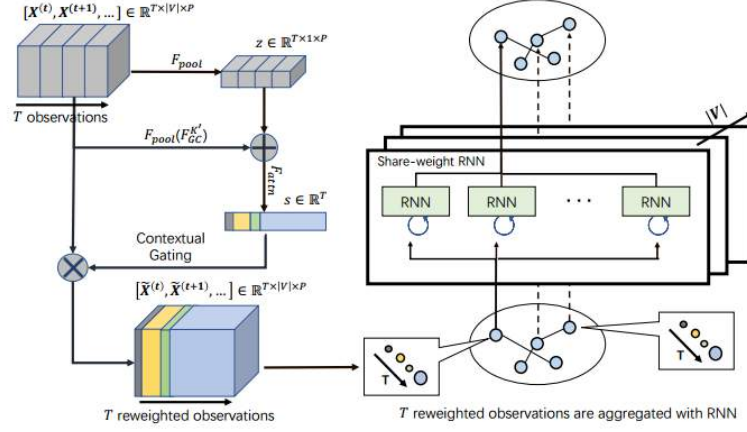


Figure 4: Temporal correlation modeling with contextual gated recurrent neural network (CGRNN). It first produces region descriptions using the global average pooling over the input and its graph convolution output for each observation. Then it transfer the summarized vector z into weights which are used to scale each observation. Finally, a shared RNN layer across all regions is applied to aggregate the gated input sequence of each region into a single vector.

下面对此部分详细解释。

此部分提出了一种基于上下文门控的循环神经网络(CGRNN)来模拟不同时间点下观测值之间的相关性。CGRNN 利用具有上下文感知门控机制的 RNN 将上下文信息集成到时态建模中，其体系结构如图 4 所示。

首先，将相关区域的信息视为上下文信息，利用对应的图拉普拉斯矩阵，通过 K 阶图卷积运算进行卷积(公式 6)。上下文门控机制的设计是先进进行图卷积再进行池化，以使池化操作包含拓扑信息。

$$\hat{\mathbf{X}}^{(t)} = [\mathbf{X}^{(t)}, F_G^{K'}(\mathbf{X}^{(t)})] \text{ for } t = 1, 2, \dots, T \quad (6)$$

其次，我们使用全局平均池化 F_{pool} 将所有节点的信息汇总为一个节点(公式 7)。

$$z^{(t)} = F_{pool}(\hat{\mathbf{X}}^{(t)}) = \frac{1}{|V|} \sum_{i=1}^{|V|} \hat{X}_{i,:}^{(t)} \text{ for } t = 1, 2, \dots, T \quad (7)$$

然后利用注意力运算对向量 z 进行非线性变换, 生成自适应权值 s (公式 8), W_1 和 W_2 相应的权重, δ σ 是分别 ReLU 和 sigmoid 函数。

$$s = \sigma(W_2 \delta(W_1 z)) \quad (8)$$

最后, 将 s 应用于对每个时间步进行加权(公式 9)。

$$\tilde{X}^{(t)} = X^{(t)} \circ s^{(t)} \quad \text{for } t = 1, 2, \dots, T \quad (9)$$

得到对时间步进行加权后的多张图后, 使用权值共享 RNN 将多张图融合为一张图 (公式 10); 区域间共享 RNN 是为所有区域找到一个通用的聚合规则, 从而促进模型的泛化, 降低模型的复杂度。

$$H_{i,:} = \text{RNN}(\tilde{X}_{i,:}^{(1)}, \dots, \tilde{X}_{i,:}^{(T)}; W_3) \quad \text{for } i = 1, \dots, |V| \quad (10)$$

22.6 总结展望

本文模型应用到了北京和上海的滴滴网约车上, 实验部分不再介绍, 其中的 POI 数据, 路网数据等感觉处理起来会比较麻烦。

总的来说, 本文就是先提取三张关系图, 即三个不同的拉普拉斯矩阵 L , 再将每张关系图的多个时间步融合为一个时间步, 即一张图, 融合后得到只含有一个时间步的三张图, 对该三张图分别进行图卷积, 再将结果进行特征融合即可得到预测值。

在未来的工作中, 我们计划研究以下几个方面:(1)评估所提出的模型对其他时空预测任务的影响;(2)扩展了多步序列预测方法。

23 Attention 注意力机制: 基于时空注意力机制的图卷积神经网络用于交通预测

23.1 文章信息

《Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting》。

北京交通大学博士生校友 2019 年初发在 AAAI 顶会上的一篇文章。

23.2 摘要

针对交通流预测问题，提出了一种基于注意力机制的时空图卷积网络 (ASTGCN) 模型。ASTGCN 主要由三个独立部分组成，分别对交通流的三个时间特性进行建模，即邻近、每日和每周的依赖关系。其中每个独立部分又包含两部分：1) 时空注意力机制，有效地捕捉交通数据中的动态时空关联；2) 时空卷积，即同时使用图卷积来捕捉空间模式和常用的标准卷积来描述时间特征。对三个分量的输出进行加权融合，生成最终预测结果。文章应用到加州 PeMS 的两个数据集上，实验表明所提出的 ASTGCN 模型性能优于较先进的基准模型。

23.3 简介

本文提出了一种基于注意力机制的时空图卷积网络 (ASTGCN) 用于预测交通网络中每个探测器的交通流量。该模型主要贡献总结如下：

(1) 提出了一种时空注意机制来研究动态时空相关性。利用空间注意力机制捕捉不同位置之间的动态空间相关性，利用时间注意力机制捕捉不同时间之间的动态时间相关性。

(2) 设计了一种时空卷积模块，用于时空相关性建模。它包括从原始的基于图的交通网络结构中获取空间特征的图卷积，以及描述邻近时间依赖关系的标准时间维卷积。

23.4 主体内容

4.1 问题描述和定义

将交通网络定义为无向图 $G = (V, E, A)$ ，如图 2(a)所示，其中 V 为 $|V| = N$ 个节点的集合； E 是一组边，表示节点之间的连通性； $A \in \mathbb{R} (N \times N)$ 表示图 G 的邻接矩阵。每个节点检测到 F 个观测值（速度、流量、时间占有率），代表着该节点的 F 个特征，如图 2(b)实线所示。本文即利用全网过去 T 个时间节点的数据（速度、流量和时间占有率）预测未来 P 个时间点的交通流量，即输入为 $X \in \mathbb{R} (N * F * T)$ ，输出为 $Y \in \mathbb{R} (N * P)$ ，其中 N 为观测站数据， $F=3$ 为每个节点的三个特征， T 为输入的 T 个时间步， P 为输出的时间步。

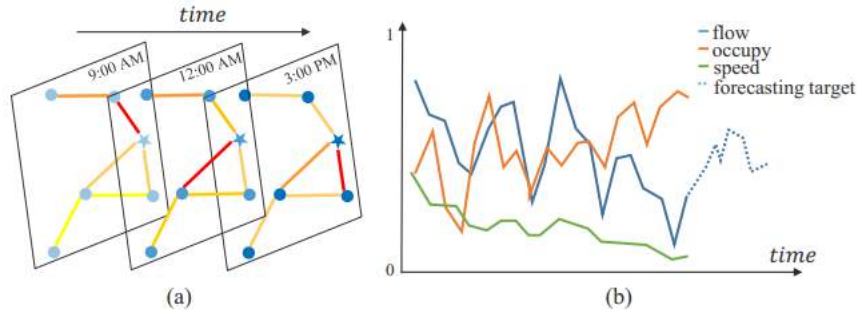


Figure 2: (a) The spatial-temporal structure of traffic data, where the data at each time slice forms a graph; (b) Three measurements are detected on a node and the future traffic flow is the forecasting target. Here, all measurements are normalized to $[0,1]$.

4.2 模型框架

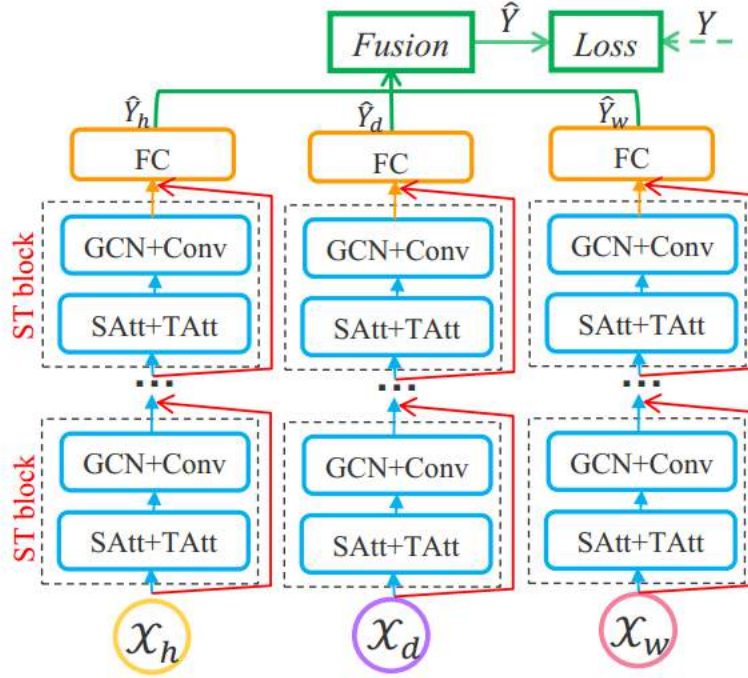


Figure 3: The framework of ASTGCN. SAtt: Spatial Attention; TAtt: Temporal Attention; GCN: Graph Convolution; Conv: Convolution; FC: Fully-connected; ST block: Spatial-Temporal block.

框架主要包含 3 个部分，分别提取邻近、日、周依赖特征。recent 部分包含邻近的 T 个时段，daily-period 部分包含前一天或多天与预测时段相同的多个时间序列，weekly-period 部分前一一周或多周与预测时段相同的多个时间序列。该三部分具有相同的网络结构，每部分由多个时空块和一个全连接层组成。在每个时空块中都有时空注意力模块和时空卷积模块。为了优化训练效率，文章采用了残差连接。最后，利用一个参数矩阵对三个分量加权合并，得到最终预测结果。下面对时空注意力模块和时空卷积模块进行详细介绍。

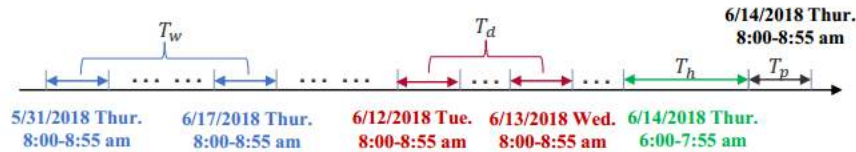


Figure 4: An example of constructing the input of time series segments (suppose the size of predicting window is 1 hour, and T_h , T_d and T_w are twice of T_p).

4.3 Spatial-Temporal Attention

(1) 空间注意力机制

在空间维度上，不同位置的交通状况相互影响，相互影响具有很强的动态性。因此我们使用注意力机制自适应地捕捉空间维度中节点之间的动态关联性。

$$\mathbf{S} = \mathbf{V}_s \cdot \sigma((\mathcal{X}_h^{(r-1)} \mathbf{W}_1) \mathbf{W}_2 (\mathbf{W}_3 \mathcal{X}_h^{(r-1)})^T + \mathbf{b}_s) \quad (1)$$

$$\mathbf{S}'_{i,j} = \frac{\exp(\mathbf{S}_{i,j})}{\sum_{j=1}^N \exp(\mathbf{S}_{i,j})} \quad (2)$$

为方便理解，公式 (1) 简化为： $\mathbf{S} = \mathbf{V} \cdot \sigma\{(\mathbf{X} \cdot \mathbf{W}_1) \cdot \mathbf{W}_2 \cdot (\mathbf{W}_3 \cdot \mathbf{X})^T + \mathbf{b}\}$

$$\mathbf{S} = \mathbf{V} \cdot \sigma\{(\mathbf{X} \cdot \mathbf{W}_1) \cdot \mathbf{W}_2 \cdot (\mathbf{W}_3 \cdot \mathbf{X})^T + \mathbf{b}\},$$

维度关系： $\mathbf{X} \in \mathbb{R} (N \times C \times T)$ ，其中 N 代表 N 个观测点， C 表示 channel 也即三个观测值（流量，速度和时间占有率）， T 代表输入的 T 个时间步。 $\mathbf{W}_1 \in \mathbb{R} (T)$ ， $\mathbf{W}_2 \in \mathbb{R} (T)$ ， $\mathbf{W}_3 \in \mathbb{R} (C \times T)$ ， $\mathbf{V} \in \mathbb{R} (N \times N)$ ， $\mathbf{b} \in \mathbb{R} (N \times N)$ ，最终结果 $\mathbf{S} \in \mathbb{R} (N \times N)$ 。公式 (2) 为 softmax 函数的表示形式，目的是将注意力矩阵 \mathbf{S} 进行归一化。归一化后的注意力矩阵 \mathbf{S}' 在后面的图卷积中将和邻接矩阵 \mathbf{A} 进行相乘。

(2) 时间注意力机制

在时间维度上，不同时间段的交通状况之间存在相关性，不同情况下的相关性也不同。同样，我们使用注意机制来自适应地赋予数据不同的权重。

$$\mathbf{E} = \mathbf{V}_e \cdot \sigma(((\mathcal{X}_h^{(r-1)})^T \mathbf{U}_1) \mathbf{U}_2 (\mathbf{U}_3 \mathcal{X}_h^{(r-1)}) + \mathbf{b}_e) \quad (3)$$

$$\mathbf{E}'_{i,j} = \frac{\exp(\mathbf{E}_{i,j})}{\sum_{j=1}^{T_{r-1}} \exp(\mathbf{E}_{i,j})} \quad (4)$$

为方便理解，公式 (3) 简化为： $\mathbf{E} = \mathbf{V} \cdot \sigma\{((\mathbf{X})^T \cdot \mathbf{U}_1) \cdot \mathbf{U}_2 \cdot (\mathbf{X} \cdot \mathbf{U}_3) + \mathbf{b}\}$

$$\mathbf{E} = \mathbf{V} \cdot \sigma\{((\mathbf{X})^T \cdot \mathbf{U}_1) \cdot \mathbf{U}_2 \cdot (\mathbf{X} \cdot \mathbf{U}_3) + \mathbf{b}\},$$

维度关系： $X \in \mathbb{R} (N \times C \times T)$ ，其中 N 代表 N 个观测点， C 表示 channel 也即三个观测值（流量，速度和时间占有率）， T 代表输入的 T 个时间步。 $U1 \in \mathbb{R} (N)$ ， $U2 \in \mathbb{R} (C \times N)$ ， $U3 \in \mathbb{R} (C)$ ， $V \in \mathbb{R} (T \times T)$ ， $b \in \mathbb{R} (T \times T)$ ，最终结果 $E \in \mathbb{R} (T \times T)$ 。公式（4）为 softmax 函数的表示形式，目的是将注意力矩阵 E 进行归一化。归一化后的注意力矩阵 E' 将和输入矩阵 X 进行相乘得到最终输入。

normalized by the softmax function. We directly apply the normalized temporal attention matrix to the input and get $\hat{\mathcal{X}}_h^{(r-1)} = (\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \dots, \hat{\mathbf{X}}_{T_{r-1}}) = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{T_{r-1}}) \mathbf{E}' \in \mathbb{R}^{N \times C_{r-1} \times T_{r-1}}$ to dynamically adjust the input by merging relevant information.

4.4 Spatial-Temporal Convolution

前面的时空注意力模块都是在为图卷积模块准备数据。本文提出的时空卷积模块由空间维上的图卷积和时间维上的标准卷积组成。

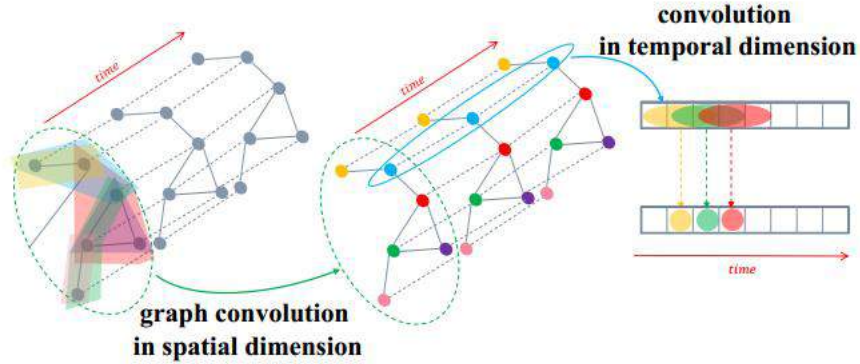


Figure 5: The architecture of spatial-temporal convolutions of ASTGCN.

（1）空间维度上的图卷积

关于图卷积，目前有两种处理方法，一种是切比雪夫多项式近似：

$$g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathbf{L}})x \quad (6)$$

其中, $\theta \in \mathbb{R}(K)$ 为多项式系数的向量,

$$\tilde{\mathbf{L}} = \frac{2}{\lambda_{\max}} \mathbf{L} - \mathbf{I}_N,$$

其中 λ_{\max} 为拉普拉斯矩阵的最大特征值, 切比雪夫多项式的迭代关系为:

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

$$\text{where } T_0(x) = 1, T_1(x) = x$$

公式相当于利用卷积核 g_θ 提取 0 到 $(K-1)$ 阶邻居的信息。将空间注意力机制应用到该图卷积过程中即为:

$$g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k(T_k(\tilde{\mathbf{L}}) \odot \mathbf{S}')x.$$

此外, 由于输入数据 $\mathbf{X} \in \mathbb{R}(N \times C \times T)$, 因此对每个时间步要有 C 个 filter。

ponent, the input is $\hat{\mathcal{X}}_h^{(r-1)} = (\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, \dots, \hat{\mathbf{X}}_{T_{r-1}}) \in \mathbb{R}^{N \times C_{r-1} \times T_{r-1}}$, where the feature of each node has C_{r-1} channels. For each time slice t , performing C_r filters on the graph $\hat{\mathbf{X}}_t$, we get $g_\theta *_G \hat{\mathbf{X}}_t$, where $\Theta = (\Theta_1, \Theta_2, \dots, \Theta_{C_r}) \in \mathbb{R}^{K \times C_{r-1} \times C_r}$ is the convolution kernel parameter (Kipf and Welling 2017). Therefore, each node is updated by the information of the $0 \sim K-1$ neighbors of the node.

(注: 还有另外一种常用的 GCN, 即 1 阶近似,

$$\begin{aligned} \Theta *_G x &= \theta(I_n + D^{-\frac{1}{2}} W D^{-\frac{1}{2}})x \\ &= \theta(\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}})x. \end{aligned} \quad)$$

(2) 时间维度上的标准卷积

图卷积获取了邻域信息, 在此基础上再叠加标准的卷积层, 在时间维度进一步合并相邻的时间步信息, 如图 5 所示。

$$\mathcal{X}_h^{(r)} = \text{ReLU}(\Phi * (\text{ReLU}(g_\theta *_G \hat{\mathcal{X}}_h^{(r-1)}))) \in \mathbb{R}^{C_r \times N \times T_r} \quad (7)$$

上面介绍的时空注意力模块和时空卷积模块构成了一个总的时空块。多个时空

块叠加，可进一步提取更大范围的动态时空关联性。最后，增加一个全连接层，保证各分量的输出具有与预测目标相同的尺寸和维度。最终的全连接层使用 ReLU 作为激活函数。

4.5 特征融合

$$\hat{\mathbf{Y}} = \mathbf{W}_h \odot \hat{\mathbf{Y}}_h + \mathbf{W}_d \odot \hat{\mathbf{Y}}_d + \mathbf{W}_w \odot \hat{\mathbf{Y}}_w \quad (8)$$

23.5 实验部分

实验部分采用了两个数据集，PeMSD4 和 PeMSD8。文章代码 GitHub 地址：

<https://github.com/guoshnBJTU/ASTGCN>。公众内回复“加州”获取文章数据集

下载地址和下载说明。

Model	PeMSD4		PeMSD8	
	RMSE	MAE	RMSE	MAE
HA	54.14	36.76	44.03	29.52
ARIMA	68.13	32.11	43.30	24.04
VAR	51.73	33.76	31.21	21.41
LSTM	45.82	29.45	36.96	23.18
GRU	45.11	28.65	35.95	22.20
STGCN	38.29	25.15	27.87	18.88
GLU-STGCN	38.41	27.28	30.78	20.99
GeoMAN	37.84	23.64	28.91	17.84
MSTGCN (ours)	35.64	22.73	26.47	17.47
ASTGCN (ours)	32.82	21.80	25.27	16.63

Table 1: Average performance comparison of different approaches on PeMSD4 and PeMSD8.

23.6 展望

未来可以考虑一些外部影响因素，例如天气因素和大型事件，进一步提高预测精度。

24 理解多图卷积：基于多图卷积神经网络的多模式交通预测应用对比

24.1 文章信息

随着图网络 GCN 的大火，各式各样的基于图网络的交通预测论文也层出不穷，多图卷积神经网络便是其中一个小方向。本文对比了公众号前面解读的两篇文章（基于多图卷积的共享单车流量预测和基于多图卷积的网约车流量预测）的相似之处，以进一步理解“多图卷积”的含义。

(1) : 《Bike Flow Prediction with Multi-Graph Convolutional Networks》。

2018 年第 26 届 ACM 空间地理信息系统进展国际会议论文集，被引 7 次。

(2) 《Spatiotemporal Multi-Graph Convolution Network for Ride-hailing Demand Forecasting》。

2019 AAAI 大会的一篇论文，被引 5 次。

24.2 摘要和简介

2.1 有桩共享单车摘要和简介：

本文将应用深度学习技术进行站点级别的共享单车流量预测。本文提出了一种多图卷积神经网络模型来预测站点级的流量，其关键创新之处在于从图的角度来观察共享单车系统。本文为一个自行车共享系统构造了多个图来反映异构的站间关系。然后，通过融合了多个图形，并应用卷积层来预测未来的站点级自行车流量。

本文提出了一种新的多图卷积神经网络来捕捉不同站点之间的空间关系，例如距离和历史使用关系。在多图卷积层的基础上，提出了一种包含 LSTM 的 encoder-decoder 编码解码器结构，用来捕捉时间关系。因此，对于站点级自行车流量预测，可以有效地捕捉时空模式。据文章所述，这是首次利用多图卷积神经网络来进行站点级别的共享自行车系统中流量预测。

问题描述：利用 t 时刻以前的进站流和出站流预测 t 时刻的进站流和出站流。

2.2 网约车摘要和简介

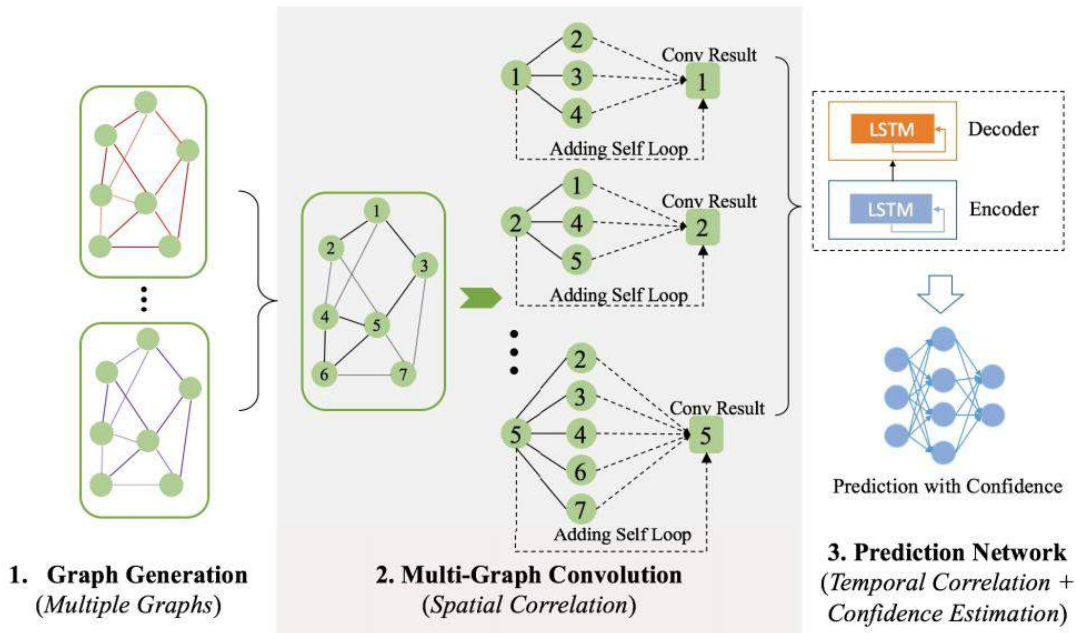
区域级网约车需求预测的目标是根据历史观测，预测城市某个区域未来的需求。这一任务的挑战性主要在于复杂的时空相关性。一方面，不同区域之间存在复杂的依赖关系，例如，一个区域的需求通常受其空间相邻区域的影响，同时与具有相似上下文环境的遥远区域相关。另一方面，不同的时间观测值之间也存在非线性依赖关系，对某一时刻的预测通常与各种历史观测结果相关联，如一小时前、一天前甚至一周前。

基于此本文提出了一种新的深度学习模型，称为时空多图卷积网络(ST-MGCN)。主要贡献如下：（1）在网约车需求预测中，识别出区域间的非欧式相关关系，并提出使用多个图对其进行编码。然后进一步利用提出的多图卷积对这些相关性进行建模。（2）提出了基于上下文门控机制的 RNN (CGRNN)模型，在时序依赖关系建模时引入全局上下文信息。

问题描述：利用 t 时刻以前的进站流和出站流预测 t 时刻的进站流和出站流。

24.3 模型对比

3.1 有桩共享单车模型框架：



文提出的多图卷积神经网络包含三部分，图生成-多图卷积-预测部分。

(1) 图生成

图的生成是图卷积模型成功的关键，如果所构建的图不能对站点间的有效关系进行编码，不仅不利于网络参数的学习，还会降低预测性能。一般来说，我们希望为具有相似动态流模式的站点之间的边缘分配较大的权重。在此基础上，提出了建立站间图的三种方法:距离图、交互图和相关图。

权重 1: 距离图，即用距离的倒数表示站间权重。

权重 2: 交互图，即用历史数据中两个车站之间的行车记录数表示站间权重。

权重 3: 相关图，计算每个站在每个时间间隔(如 1 小时)内的历史使用情况(流入或流出)，然后计算每两个站之间的相关性作为图中站间权重。本文使用皮尔逊系数来计算相关性。

(2) 多图卷积

此部分可分为两块主要内容：图融合以及图卷积。图融合即将上面的三个图权重矩阵融合为一个图权重。我们首先对每个图的邻接矩阵 A 进行标准化，然后在元素级对不同图的邻接矩阵加权求和来组合不同的图。为了使加权和运算后的融合结果保持归一化，我们在加权矩阵中进一步增加了一个 softmax 运算。得到加权矩阵后利用图卷积层进行训练。

$$f(H^{(l)}, A) = \sigma \left(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right),$$

with $\hat{A} = A + I$, where I is the identity matrix and \hat{D} is the diagonal node degree matrix of \hat{A} .

(3) 预测网络部分

采用 encoder-decoder LSTM 结构的预测网络。详细信息请查看顶部历史文章链接。

3.2 网约车模型框架

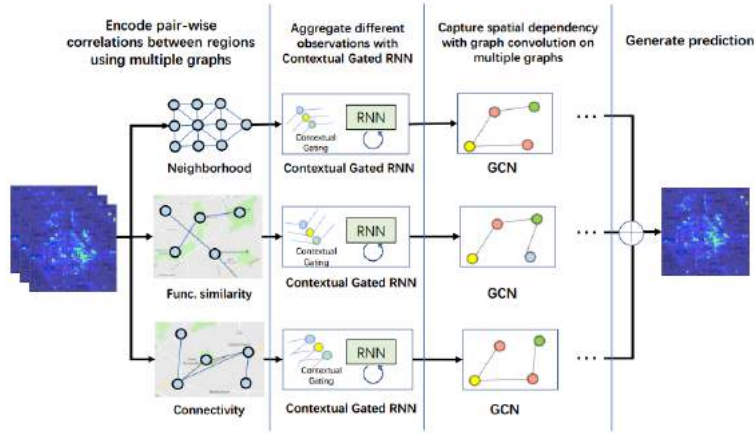


Figure 2: System architecture of the proposed *spatiotemporal multi-graph convolution network* (ST-MGCN). We en-

所提出的模型 ST-MGCN 的系统架构如图 2 所示。我们将区域间三种关系类型表示为多个图，图的顶点表示区域，边表示区域间的成对关系。首先，时间上，我们使用提出的上下文门控循环神经网络(CGRNN)考虑全局上下文信息，对不同时间的观测结果进行汇总，将多个时间步融合为一个时间步。然后，空间上，应用多图卷积来捕捉区域间不同类型的相关性。最后，利用全连接层进行特征融合转化为预测。

(一) 空间依赖关系建模

所谓多图卷积，是指对空间上的多种相关关系分别进行图卷积，再进行特征融合。我们用图对区域间的三种相关性进行了建模，包括(1)邻域图，编码空间邻近性，(2)区域功能相似图，它对区域周围兴趣点的相似性进行编码，(3)交通连通性图，它编码了遥远地区之间的连通性。

(1) 邻域图

区域的邻域是根据空间邻近性来定义的。我们在一个 3×3 的网格中，将一个区域与其相邻的 8 个区域连接起来，构造出图形。

(2) 功能相似图

在对一个区域进行预测时，可以直观地参考与此区域功能类似的其他区域。区域功能可以用其周围每个类别的 POIs 来表征，两个顶点(区域)之间的边定义为 POI

相似性:

(3) 交通连通性图

在进行时空预测时，交通系统也是一个重要因素。直观地说，那些地理上遥远但方便到达的区域可以相互关联。这些连接是由高速公路、高速公路或地铁等公共交通诱导的。这里，我们将这些道路直接相连的区域定义为“连通”，对应的边定义为：

(4) 利用多图卷积对时空依赖关系进行建模

利用构建的上述三种类型的图，我们使用如下公式进行卷积， $f(\mathbf{A}; \theta_i)$ 表示拉普拉斯矩阵。

$$\mathbf{X}_{l+1} = \sigma \left(\bigsqcup_{\mathbf{A} \in \mathbb{A}} f(\mathbf{A}; \theta_i) \mathbf{X}_l \mathbf{W}_l \right) \quad (5)$$

where $\mathbf{X}_l \in \mathbb{R}^{|V| \times P_l}$, $\mathbf{X}_{l+1} \in \mathbb{R}^{|V| \times P_{l+1}}$ are the feature vectors of $|V|$ regions in layer l and $l+1$ respectively. σ denotes the activation function, and \bigsqcup denotes the aggregation function, e.g., sum, max, average etc. \mathbb{A} denotes the set of graphs, and $f(\mathbf{A}; \theta_i) \in \mathbb{R}^{|V| \times |V|}$ represents the aggregation matrix of different samples based on graph $\mathbf{A} \in \mathbb{A}$ parameterized by θ_i , while $\mathbf{W}_l \in \mathbb{R}^{P_l \times P_{l+1}}$ denotes the feature transformation matrix, For example, if $f(\mathbf{A}; \theta_i)$ is the poly-

在实际过程中， $f(\mathbf{A}; \theta_i)$ 选为拉普拉斯算子 L 的 K 阶多项式函数，图 3 显示了图卷积层示例。 K 表示 v_i 通过 K 步能够到达 v_j 。在卷积运算方面， k 定义了空间特征提取时感知域的大小。

(二) 时间依赖关系建模

即通过多步处理，将多个时间步的信息融合到一张图上。这一块比较难理解，也不是本文的重点，详细信息请查看顶部历史文章链接。

24.4 对比思考

这两篇文章最大的相似之处是对同一种交通网络构建了多种空间关系，即多张图，也可以说多个邻接矩阵，然后通过对多张图的处理得到最终预测结果。

比较大的不同之处：

其一在于三种图考虑的空间关系不同，共享单车考虑了站间距离，站间流量交互，以及站间流入流出流量的相关性，而网约车论文考虑了邻域图，功能相似图和交通连通性。

其二在于如何融合构建的三张图，共享单车论文是对构建的三个邻接矩阵进行加权融合，而网约车论文是对三张图分别先进行图卷积再求和。至于那种方式在技术上和理论上更优一些还待验证。

其三在于时间依赖关系建模方面，共享单车论文最终采用了 LSTM encoder-decoder 的结构捕捉时间依赖关系，而网约车论文提出了一种基于上下文信息的门控循环神经网络来捕捉时间依赖关系。

其四在于共享单车论文为有桩共享单车，“桩”更类似与一张图中的节点，因此共享单车网络更类似于一张图。而网约车论文还是基于网格数据来做的，其是将一个区域看成了一个节点，和图中的节点还是有所区别的。

25 OD 需求预测：基于图卷积神经网络 GCN 的出租车 OD 需求预测

25.1 文章信息

《Origin-Destination Matrix Prediction via Graph Convolution: a New Perspective of Passenger Demand Modeling》。

北航计算机学院发在 2019KDD 上的一篇文章。

25.2 摘要

为了获得乘客的出行模式，打车平台需要提前预测一个地区到另一个地区的乘客需求数量，即 OD 矩阵预测(ODMP)问题。OD 矩阵预测比普通需求预测更具挑战性。除了要预测一个地区的需求产生量，还需要预测需求的目的地。此外，数据稀疏性是一个严重的问题。因此本文提出了一种基于网格嵌入的单馈多任务学习模型(GEML)。该模型主要包含两个部分，分别提取时间信息和空间信息。网格嵌入部分是为了对乘客的空间移动模式和不同区域的相邻关系进行建模，其预加权聚合器的目的是感知数据的稀疏性和范围；多任务学习部分则侧重于时间属性建模和捕获 ODMP 问题目标。两个数据集-UCAR 和 Didi-的结果表明 GEML 方法优于基准模型。

25.3 简介

除了预测某一区域内可能的乘客需求数量外，了解每次出行的来源地和目的地的乘客需求也很重要。因为不同时段两个区域之间的需求量不仅承载着乘客需求的强度，而且有利于挖掘有用的出行模式。本文从一个新的角度研究了乘客需求模型，即 OD 矩阵预测 (ODMP)。OD 矩阵包含两个方面的信息: (1)不同的 OD 组合; (2)每个 OD 对的旅客需求数量。ODMP 的目标是预测在给定时间段内从一个地理区域到另一个地理区域的叫车订单数量。为了同时兼顾出行产生量和目的地，时空特性以及数据稀疏性，本文提出了一种基于网格嵌入的单馈多任务学习模型 (GEML)，以基于图对出行模式进行建模。具体来说，我们用图表示与地理区域相关的乘客订单记录，其中节点表示地理区域(以网格形式定义)，节点之间的边表示乘客需求，边权重表示订单数量。利用改进后的网格，可以构造出给定时间间隔内的 OD 矩阵。如图 1 所示，将区域划分为 16 个网格，订单记录汇总在相应的 OD 矩阵中。

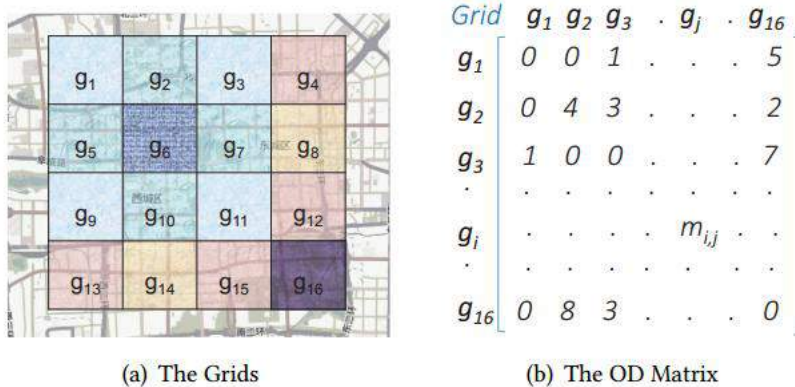


Figure 1: Example of Grids and OD Matrix

本文模型的灵感来自于最近大火的 GCNs，然而如果我们直接将已有的 GCNs 应用到 OD 矩阵所生成的图上，由于数据稀疏，学习到的具有很少订单的网格嵌入往往是不可靠和无效的，此外，如果没有任何历史订单记录的孤立节点(例如，新建社区)，学习到的网格嵌入也是不可行的(无论作为 O 点还是 D 点)。为了缓解数据的稀疏性问题，我们提出基于地理学第一定律探索网格的地理相关性，即所有的东西都是相关的，但附近的東西比遙遠的東西更相关。例如，在两个地理位置相近的网格中，乘客需求的数量往往接近彼此。特别地，我们考虑了网格嵌入部分的两种邻域，即地理邻域（地理上相邻的）和语义领域（通过 OD 流连接起来的）。前者用于度量一个网格与其邻域之间的内在紧密程度，后者用于对网络 OD 之间的交通流强度建模。

基于网格嵌入学习得到的网格的表示，结合乘客需求的重要时间信息，设计了一个面向 ODMP 的多任务神经网络。受既有工作的启发，我们对一个网格的流入和流出流分别建模，预测每个网格在不同时间段的流入和流出需求数量。引入这两个子任务的基本原理是，我们能够在每个网格上单独捕获更多的动态出行模式。通过补充两个单独的子任务，总体需求预测任务可以捕获更强的内在时间模式，因为每个网格中的总体需求具有更大的规模或粒度。例如，在早高峰时段，当网格划分的粒度很小时，网约车需求的目的地可能存在很大不同，导致数据稀疏性问题，这意味着乘客需求的目的地可能分布得非常广泛，但这些网格的总流入流和流出流是非常大的。

本文主要贡献如下：

(1) 提出 OD 矩阵预测问题预测给定时间段内的 OD 乘客需求，这对于网约车平台运营管理具有重要意义。

(2) 将研究区域划分为网格，设计了网格嵌入网络，通过在新定义的网格邻域(地理和语义邻域)之间的图卷积，对每个网格进行嵌入，该网络通过模仿 GCNs 中的信息传递模式来模拟不同网格之间的 OD 流关系。

(3) 借助 LSTM 设计了一个多任务学习网络用于捕捉乘客需求的时间趋势。两个子任务预测网格中的单个流入流和流出流需求，而主任务预测每对网格之间的需求。

(4) 在两个真实大规模叫车数据集上的大量实验表明提出的 GEML 模型性能优于基准模型。

25.4 模型框架

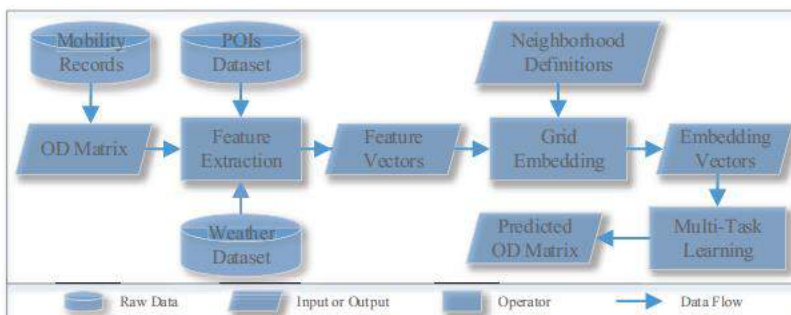


Figure 2: An overview of the GEML model.

GEML 模型能同时捕获空间和时间特性。从空间角度出发，提出了一种基于邻域的网格嵌入方法，通过聚集邻域信息来学习每个网格的向量表示。从时间的角度，我们设计了一个多任务学习框架来模拟乘客需求随时间的动态趋势。接下来，我们将介绍网格嵌入和多任务学习的技术细节。

4.1 Grid Embedding

在 ODMP 环境下，提出了网格嵌入部分的两种邻域，即地理邻域（地理上相

邻的)和语义领域(通过OD流连接起来的)。前者用于度量一个网格与其邻域之间的内在紧密程度,后者用于对网络OD之间的交通流强度建模,如图所示。

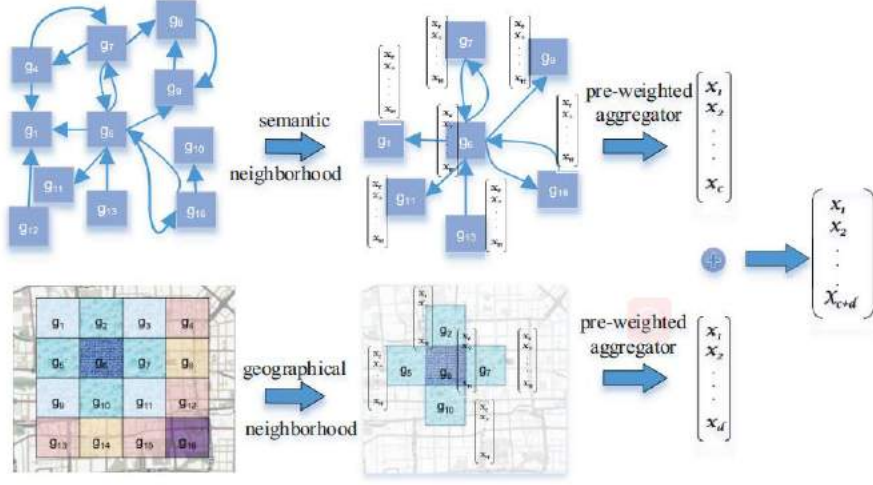


Figure 3: Illustration of grid embedding for both geographical and semantic neighborhoods.

(1) Geographical Neighborhood

两个区域中心点之间的距离小于一定的阈值可定义为地理邻域。

$$\Phi_i = \{g_j | dis(g_i, g_j) \leq L\}. \quad (1)$$

(2) Semantic Neighborhood

如果两个区域之间至少有一个 OD 流(可以是相反方向),即可定义为两者为语义邻域。在任意时间段 $t' = 1, 2, \dots$, 我们可以通过公式 2 获取网格 i 的语义邻域。由式(2)可知,不同网格在不同时间间隔的语义邻居个数是不确定的。由于 ODMP 问题对时间敏感,因此考虑不同网格在不同时间间隔的语义关系至关重要。

$$\Omega_{t'}^i = \{g_j | m_{i,j} > 0 \parallel m_{j,i} > 0, m_{i,j} \in M_{t'}, m_{j,i} \in M_{t'}\}. \quad (2)$$

(3) Pre-Weighted Aggregator for Grid Embedding

我们通过聚合地理邻域 Φ 和语义邻域 Ω 推断每个网格的向量表示,我们不是为每个网格训练一个不同的嵌入向量,而是训练一个聚合器函数,它学会从网格的邻域中积累和选择特征信息。在详细介绍用于网格嵌入的预加权聚合器之前,我们

首先简要介绍[10]采用的朴素聚合器形式：

$$\mathbf{v}_i = \sigma\left(\mathbf{W} \cdot \text{MEAN}(\{\mathbf{v}'_i\} \cap \{\mathbf{v}'_j, g_j \in N_i\})\right), \quad (3)$$

\mathbf{v}_i 是网格 g_i 的嵌入向量， $\text{MEAN}(\cdot)$ 表示对应元素均值， \mathbf{v}_j 是网格 g_j 的嵌入向量，朴素聚合方法即是计算 \mathbf{v}_i 和 \mathbf{v}_j 对应元素的均值，并将它们连接到之前的特性 \mathbf{v}_i 。然而，尽管有一些基本聚合器的变体(例如 pooling aggregator and LSTM aggregator)，现有的图卷积聚合方法在 ODMP 场景下缺乏充分捕捉不同网格之间关系的能力，无法进行需求建模，原因是这些聚合器在融合每个网格邻居的所有特性时无法区分它们的重要性，直观地说，两个网格之间的地理距离越近，它们的属性就越相似。此外，在语义邻居集中，邻居网格的受欢迎程度应该对聚合过程产生影响，因为它保留了具有代表性的出行模式。

在此基础上，提出了一种预加权聚合器，该聚合器可以选择性地将重点放在网格嵌入的重要邻域上。对于网格的地理邻域，我们利用相邻区域之间的距离作为聚合器的权重因子。因此，我们将地理邻域的预加权聚合器表示为：

$$\mathbf{r}_{t'}^i = \sigma\left(\mathbf{W}_g \cdot (\mathbf{f}_{t'}^i + \sum_{g_j \in \Phi_i} \frac{\text{dis}(g_i, g_j)}{\sum \text{dis}(g_i, g_j)} \mathbf{f}_{t'}^j)\right), \quad (4)$$

对于语义邻域，degree 代表 OD 流量，即两个区域之间的 OD 流（从 i 到 j 或从 j 到 i ）， ϵ 是一个非常小的值接近于零，以防 $\text{degree}(g_j) = 0$ 。

$$\mathbf{s}_{t'}^i = \sigma\left(\mathbf{W}_s \cdot (\mathbf{f}_{t'}^i + \sum_{g_j \in \Omega_{t'}^i} \frac{\text{degree}(g_j)}{\sum \text{degree}(g_j) + \epsilon} \mathbf{f}_{t'}^j)\right), \quad (5)$$

注意，两种表示都是随时间变化的，是一个动态指标。最后，将两种语义表示连接起来得到一个网格最终的语义表示。

$$\mathbf{v}_{t'}^i = [\mathbf{r}_{t'}^i, \mathbf{s}_{t'}^i], \quad (6)$$

4.2 Multi-Task Learning

本节出了一种具有 periodic-skip LSTM 的多任务学习方案，如图 4 所示。

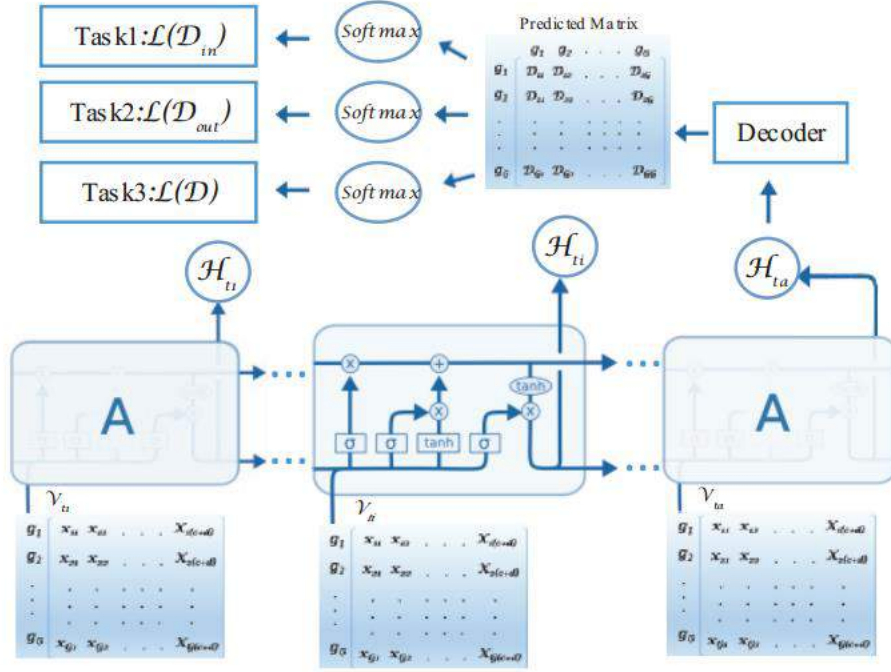


Figure 4: The architecture of the multi-task LSTM.

(1) Periodic-Skip LSTM

$$\mathbf{h}_t = LSTM(\mathbf{x}_t, \mathbf{h}_{t-1}). \quad (8)$$

为了全面了解乘客需求的动态模式，我们从 UCAR 数据集中随机抽取了 5 天，并在图 5 中绘制了每天的每小时乘客需求。

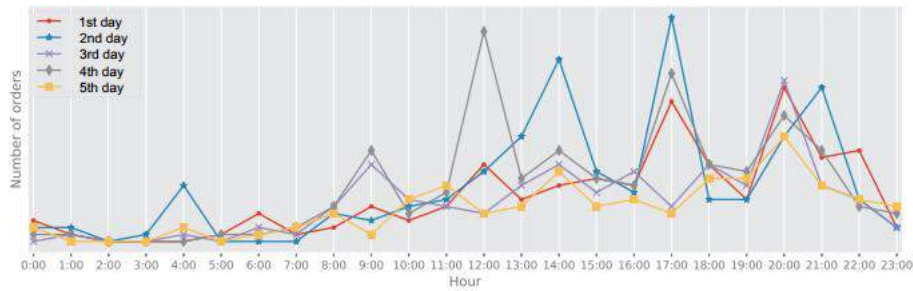


Figure 5: The number of passenger demands w.r.t. different hours and days.

显然，与此同时，乘客需求的数量也有类似的模式。然而，在预测下一小时的乘客需求时，LSTM 中的序列建模方案将迫使模型从之前的连续小时中收集信息。这可能对需求预测没有多大帮助，因为不相关的输入会产生很多噪音。为此，为了

更好地对周期性进行建模，我们取网格嵌入序列 $\{v_{i1}, v_{i2}, \dots, v_{it}\}$ 作为输入，进一步将 Eq.(8)转换为 Periodic-Skip LSTM，跳过不相关的顺序模式。

$$\mathbf{h}_t^i = LSTM^{ps}(\mathbf{v}_t^i, \mathbf{h}_{t-p}^i) \quad (9)$$

其中 p 是跳过的隐藏状态数。

(2) main task

由 periodic-skip LSTM 框架,我们可以得到在时间 t 网格 g_i 的向量表示。为了得到 OD 矩阵中每一项 m_{ij} 的值，我们构造了一个过渡矩阵 $\mathbf{W}_m \in \mathbb{R} (d \times d)$ 以对 OD 流进行建模。此时，预测值 m 可由下式计算。损失函数为均方误差。

$$\hat{m}_{i,j} = (\mathbf{W}_m \mathbf{h}_t^i)^\top \mathbf{h}_t^j, \quad (10)$$

and we use mean squared error to compute the loss function for the main task:

$$\mathcal{L}_{ODMP} = \frac{1}{|\mathbf{M}_{t+1}| \times N} \sum_{n=1}^N ||\mathbf{M}_{t+1} - \hat{\mathbf{M}}_{t+1}||, \quad (11)$$

where $m_{i,j} \in \mathbf{M}_{t+1}$ is the real value in the OD matrix at time $t + 1$, $n \leq N$ is the index of the training sample.

(3) Two Subtasks: Predicting the In- and Out-Degrees

在预测上述总体 OD 矩阵的主要任务的同时，我们还分别对流入流 (p) 和流出流 (q) 进行了建模。其中 \mathbf{W}_{in} 和 \mathbf{W}_{out} 是用于将网格嵌入投影到标量的两个投影权重。损失函数为均方误差， G 是网格的个数。

$$\hat{p}_i = \mathbf{w}_{in}^\top \mathbf{h}_t^i, \quad (12)$$

$$\hat{q}_i = \mathbf{w}_{out}^\top \mathbf{h}_t^i, \quad (13)$$

$$\mathcal{L}_{IN} = \frac{1}{|G| \times N} \sum_{n=1}^N \sum_{g_i \in G} (p_{i,n} - \hat{p}_{i,n})^2. \quad (14)$$

$$\mathcal{L}_{OUT} = \frac{1}{|G| \times N} \sum_{n=1}^N \sum_{g_i \in G} (q_{i,n} - \hat{q}_{i,n})^2. \quad (15)$$

where G is the number of grids in the OD matrix, $n \leq N$ is the index of the training sample.

(4) 损失函数

针对上述三个任务，我们将主要任务的损失和两个子任务的损失结合起来，制定出总体损失函数。

$$\mathcal{L}_{GEML} = \eta \mathcal{L}_{ODMP} + \eta_{in} \mathcal{L}_{IN} + \eta_{out} \mathcal{L}_{OUT} \quad (16)$$

Because the importances of different loss functions may be different, η_{in} , η_{out} , η are added as the weights of them.

实验部分不再详述。

25.5 总结

总体而言，文章的研究角度确实很新，值得借鉴。但这篇文章仍有一个致命弱点，即没有考虑行程时间，由 O 点到 D 点需要一定的行程时间，因此实时的 OD 矩阵是不可能得到的，因此利用真实的历史 OD 矩阵作为输入在实时预测中是不可行的。

26 Dynamic GCN: 基于动态时空图卷积神经网络的交通流预测

26.1 文章信息

《Dynamic Spatial-Temporal Graph Convolutional Neural Networks for Traffic Forecasting》。

湖南大学信息科学与工程学院 2019 年初发在 AAAI 顶会上的一篇文章。

26.2 摘要

图卷积神经网络能够利用基于节点距离的预先定义的拉普拉斯矩阵，对一张图中节点的空间依赖关系进行建模。然而，在许多应用场景中，空间依赖关系会随时间而变化，使用固定的拉普拉斯矩阵不能捕捉这种变化。为了跟踪交通数据之间的空间依赖关系，我们提出了一种动态的时空 GCNN 来进行交通预测，核心是对拉普拉斯矩阵进行动态分析。为了在参数学习过程中降低复杂度，我们将张量分解融入到深度学习框架中，实时交通数据被分解为一个稳定的、依赖于长期时空关系的全局分量和一个捕捉短期波动的局部分量。在理论推导的基础上，提出了一种新的估计具有上述两种分量的图的动态拉普拉斯矩阵的设计方法，并介绍了设计依据。利用两个数据集评价结果表明，该网络的表现比基准模型提高了 25%。

26.3 简介

GCNN 很大程度上依赖于图的拉普拉斯矩阵，被定义为节点度对角矩阵与邻接矩阵之差。以前的 GCNN 研究假设拉普拉斯矩阵严格不变，即输入图的邻接矩阵是常数。然而我们之前的研究表明，在不同的时间跨度内，交通模式之间存在着巨大的差异。此外，每天都可能发生交通事故，这也会影响到路网中路段之间的关系。这些因素会导致邻接矩阵的动态变化，从而影响拉普拉斯矩阵。因此，图的拉普拉斯矩阵可能是时变的。

为了解决上述问题，提出了一种新的时空结构-动态 GCNN (DGCNN)，来预测整个网络的交通速度，与现有的基于 GCNN 相比，本文的贡献如下：

(1) 将张量分解引入到深度学习框架中，从交通数据样本中提取全局和局部分量。从频率分析可以看出，一个时间跨度内的全网交通样本由两部分组成：一个是由路网结构决定的全局分量，另一个是由特定时段或交通事件决定的局部分量。文章使用一个特定的损失函数训练张量分解层。

(2) 为了根据全局和局部分量动态学习特定时刻的拉普拉斯矩阵，设计了一种基于深度学习的拉普拉斯矩阵估计器，并给出了详细的理论推导和设计依据。将实时估计的拉普拉斯矩阵输入到图卷积层进行预测。

27 OD 需求预测：基于深度学习框架的出租车 OD 需求预测应用对比

27.1 文章信息

前面看了 OD 需求预测的文章，有一些相似之处，拿出来放到一起对比下。

一、《Contextualized Spatial - Temporal Network for Taxi origin-Destination Demand Prediction》。

2019 发在 IEEE Transactions on Intelligent Transportation Systems 上的一篇文章（运输科技 2 区，IF:5.744）。

二、《Origin-Destination Matrix Prediction via Graph Convolution: a New Perspective of Passenger Demand Modeling》。

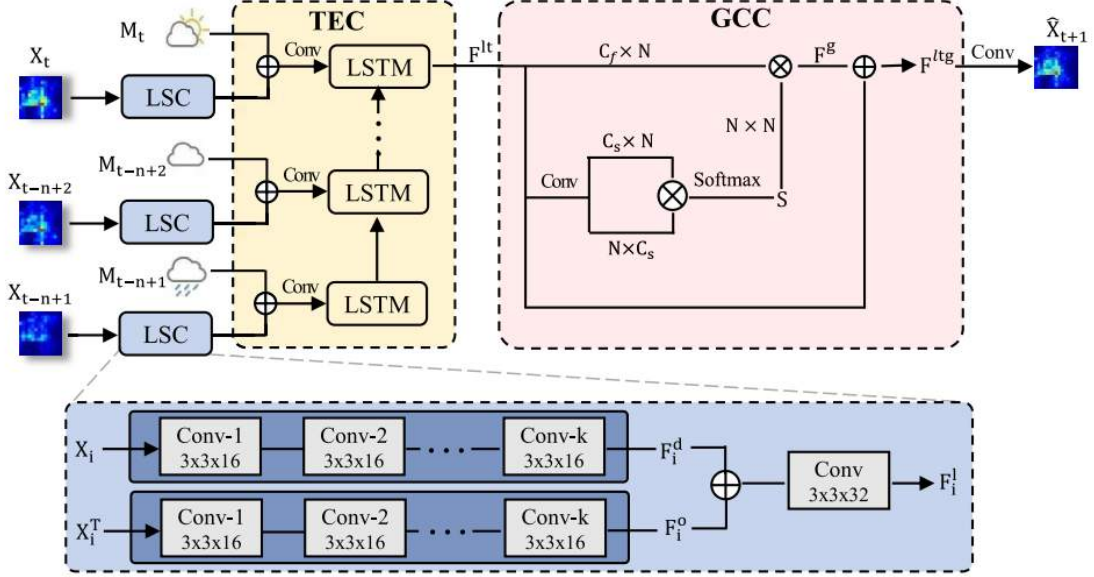
北航计算机学院发在 2019KDD 上的一篇论文。

27.2 摘要

一、第一篇

本文提出了一种新的符合实际环境的时空网络(contextualized spatial-temporal network, CSTN)，它由局部空间环境(local spatial context, LSC)、时间演化环境(temporal evolution context TEC)和全局相关环境(global correlation context, GCC)三部分组成。（context 这个词，网上大部分都是翻译成上下文，我感觉好像环境更好一点，本文所有 context 的翻译均为环境）。首先，局部空间环境模块利用两个卷积神经网络分别从起点和终点两个角度学习局部空间依赖关系。其次，将出租车需求的局部空间特征和气象信息集成到卷积长短时记忆网络(ConvLSTM)中，分析

出租车需求演化。最后，为了捕获相隔较远的区域之间的相关性，全局相关环境模块计算任意两个区域之间的相似性，并将所有区域的特征与相似性权重相加，生成每个区域的全局相关性特征。实验表明本文模型的表现最好。



这项工作的主要贡献有三方面：

(1) 将现有的出租车需求预测扩展到出租车原目的地需求预测的任务中，对智能交通系统具有更大的探索价值。据我们所知，我们是第一个研究跨区域出租车需求预测的。

(2) 提出了一种新的时空网络解决了这一问题，它很好地将局部空间环境、时间演化环境和全局相关环境集成到一个统一的框架中。

(3) 在出租车 OD 需求预测的很多基准上进行的大量实验表明，我们的方法比现有的最先进的方法有一定的优势。

二、第二篇

为了获得乘客的出行模式，打车平台需要提前预测一个地区到另一个地区的乘客需求数量，即 OD 矩阵预测(ODMP)问题。OD 矩阵预测比普通需求预测更具挑战性。除了要预测一个地区的需求产生量，还需要预测需求的目的地。此外，数据

稀疏性是一个严重的问题。因此本文提出了一种基于网格嵌入的单馈多任务学习模型(GEML)。该模型主要包含两个部分，分别提取时间信息和空间信息。网格嵌入部分是为了对乘客的空间移动模式和不同区域的相邻关系进行建模，其预加权聚合器的目的是感知数据的稀疏性和范围；多任务学习部分则侧重于时间属性建模和捕获 ODMF 问题目标。两个数据集-UCAR 和 Didi-的结果表明 GEML 方法优于基准。

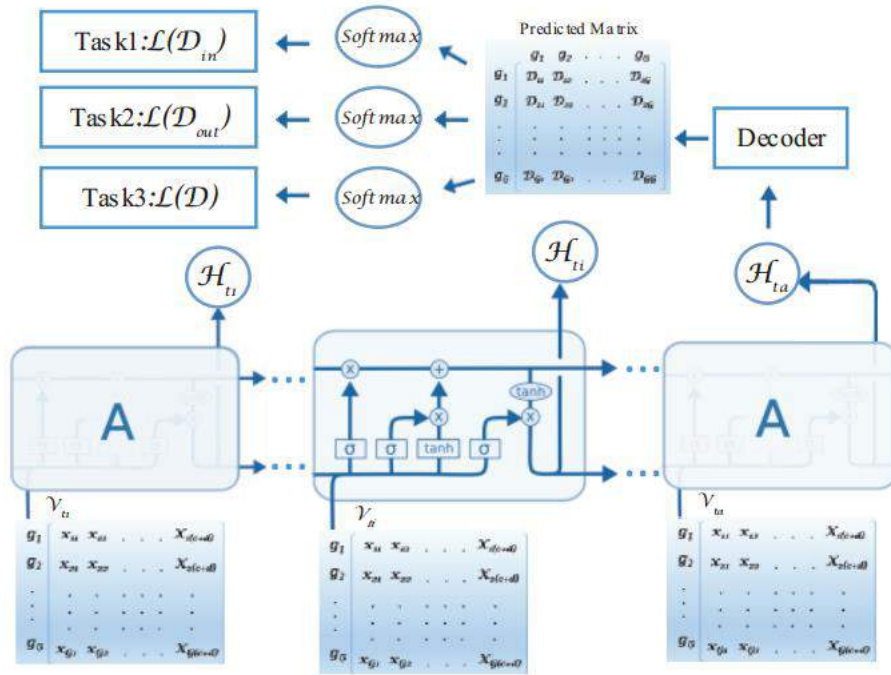


Figure 4: The architecture of the multi-task LSTM.

本文主要贡献如下：

- (1) 提出 OD 矩阵预测问题预测给定时间段内的 OD 乘客需求，这对于网约车平台运营管理具有重要意义。
- (2) 将研究区域划分为网格，设计了网格嵌入网络，通过在新定义的网格邻域(地理和语义邻域)之间的图卷积，对每个网格进行嵌入，该网络通过模仿 GCNs 中的信息传递模式来模拟不同网格之间的 OD 流关系。
- (3) 借助 LSTM 设计了一个多任务学习网络用于捕捉乘客需求的时间趋势。两个子任务预测网格中的单个流入流和流出流需求，而主任务预测每对网格之间的

需求。

(4) 在两个真实大规模叫车数据集上的大量实验表明提出的 GEML 模型性能优于基准模型。

28 T-GCN-时间图卷积网络用于交通预测

28.1 文章信息

《T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction》。

中南大学 地球科学与信息物理学院 2019 年发在 IEEE Transactions on Intelligent Transportation Systems 上的一篇文章。

28.2 摘要

为了同时捕获空间和时间依赖关系, 文章提出了一种基于神经网络的交通预测方法—时间图卷积网络(T-GCN)模型, 该模型结合了图卷积网络(GCN)和门控递归单元(GRU)。GCN 用于学习复杂的拓扑结构来捕获空间依赖关系, GRU 用于学习交通数据的动态变化来捕获时间依赖关系。T-GCN 模型应用于基于城市路网的交通预测。实验表明 T-GCN 模型预测结果优于基准模型。

28.3 简介

提出了一种新的基于城市道路网络的交通预测方法—时间图卷积网络(T-GCN)。主要贡献如下:

(1) T-GCN 模型集成了图卷积网络和门控递归单元。利用图卷积网络捕获路网拓扑结构, 对路网的空间依赖性进行建模。利用门控递归单元捕获道路交通数据的动态变化, 对时间依赖性进行建模。T-GCN 模型也可应用于其它时空预测任务。

(2) T-GCN 模型的预测结果表明,在不同的预测水平下, T-GCN 模型处于稳定状态,这说明 T-GCN 模型不仅可以实现短期预测,还可以用于长期的交通预测任务(15 分钟 30 分钟 45 分钟 60 分钟)。

(3) 我们使用两个真实的交通数据集来评估我们的方法。结果表明,与所有基线方法相比,该方法的预测误差降低了约 1.5%-57.8%,证明了 T-GCN 模型在交通预测方面的优越性

本文的其余部分组织如下。第二部分回顾了交通预测的相关研究。第三部分详细介绍了我们的方法。在第四部分,我们评估了 T-GCN 的预测性能,包括模型参数设计、预测结果分析、扰动分析和模型解释。我们在第五部分对本文进行了总结。

28.4 主体部分

在本研究中,交通预测的目标是基于道路上的历史交通信息,预测一定时期内的交通信息。在我们的方法中,交通信息是一个一般概念,可以是交通速度、交通流量或交通密度。为保证一般性,我们在实验部分以交通速度作为交通信息的一个例子。

(1) 问题定义

道路网络为 G , 我们使用未加权图 $G = (V, E)$ 来描述道路网的拓扑结构,每个道路作为一个节点, V 是道路节点集合, $V = \{v_1, v_2, \dots, v_N\}$, N 是节点的数量, E 代表边集合。邻接矩阵 A 表示道路之间的连接, $A \in \mathbb{R} (N \times N)$ 。邻接矩阵只包含 0 和 1 的元素。

特征矩阵 $X \in \mathbb{R} (N \times P)$ 。道路上的交通信息网络是网络中节点的属性特征, P 代表节点属性的数量特征(历史时间序列的长度), $X_t \in \mathbb{R} (N \times i)$ 我用于表示每条道路的速度。同样,节点属性特性可以是任何交通信息,比如交通速度、交通流和交通密度

因此，时空交通预测问题可以认为是在路网拓扑 G 和特征矩阵 X 的前提下学习映射函数 f ，然后计算未来 T 时刻的交通信息，如式 1 所示：

$$[X_{t+1}, \dots, X_{t+T}] = f(G; (X_{t-n}, \dots, X_{t-1}, X_t)) \quad (1)$$

(2) Overview

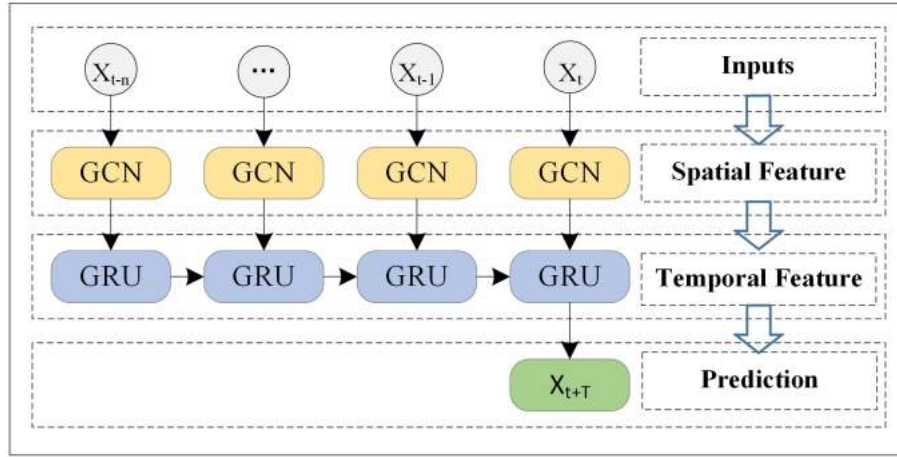


Fig. 3. Overview. We take the historical traffic information as input and obtain the finally prediction result through the Graph Convolution Network and the Gated Recurrent Unit model.

T-GCN 模型由图卷积网络和门控递归单元两部分组成。如图 3 所示，我们首先使用历史 n 时间序列数据作为输入，利用图卷积网络捕获城市路网拓扑结构，获得空间特征。其次，将得到的具有空间特征的时间序列输入门控递归单元模型，通过单元间的信息传递获得动态变化，捕捉时间特征。最后，通过全连通层得到结果。

(3) 模型详细介绍

1) Spatial Dependence Modeling

城市道路网络不是二维网格，而是图的形式，这意味着 CNN 模型不能反映城市道路网络复杂的拓扑结构因此不能准确地捕捉空间依赖关系。

给定一个邻接矩阵 A 和特征矩阵 X , GCN 模型在傅里叶域中构造一个过滤器。该滤波器作用于图的节点上, 通过它的一阶邻域捕获节点之间的空间特征, 然后通过叠加多个卷积层来构建 GCN 模型, 可以表示为:

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \hat{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} \theta^{(l)} \right) \quad (2)$$

where $\tilde{A} = A + I_N$ is the matrix with added self-connections, I_N is the identity matrix, \tilde{D} is the degree matrix, $\tilde{D} = \sum_j \tilde{A}_{ij}$, $H^{(l)}$ is the output of l layer, $\theta^{(l)}$ contains the parameters of that layer, and $\sigma(\cdot)$ represents the sigmoid function for a nonlinear model.

本研究选择 2 层 GCN 模型捕捉空间依赖关系, 可以表示为:

$$f(X, A) = \sigma \left(\hat{A} ReLU(\hat{A} X W_0) W_1 \right) \quad (3)$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ denotes pre-processing step, $W_0 \in R^{P \times H}$ represents the weight matrix from input to hidden layer, P is the length of feature matrix, and H is the number of hidden unit, $W_1 \in R^{H \times T}$ represents the weight matrix from hidden to output layer. $f(X, A) \in R^{N \times T}$ represents the output with the prediction length T , and $ReLU()$, standing for REctified Linear Unit, which is a frequently used activation layer in modern deep neural networks.

综上所述, 我们使用 GCN 模型从交通数据中学习空间特征。如图 4 所示, 假设节点 1 为中心道路, GCN 模型可以得到中心道路与其周围道路之间的拓扑关系, 对路网拓扑结构和道路属性进行编码, 得到空间依赖关系。

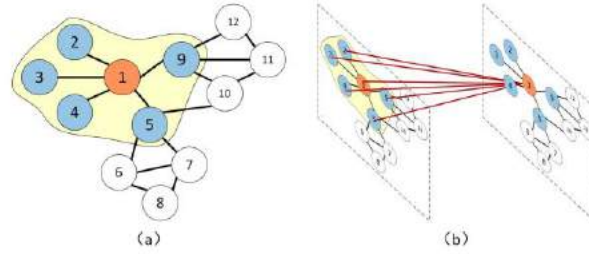


Fig. 4. Assuming that node 1 is a central road. (a) The blue nodes indicate the roads connected to the central road. (b) We obtain the spatial features by obtaining the topological relationship among the road 1 and its surrounding roads.

2) Temporal Dependence Modeling

然而，由于 LSTM 结构复杂，训练时间较长，而 GRU 模型结构相对简单，参数较少，训练能力较快。因此，我们选择了利用 GRU 模型从交通数据中获取时间相关性。其结构如下图。

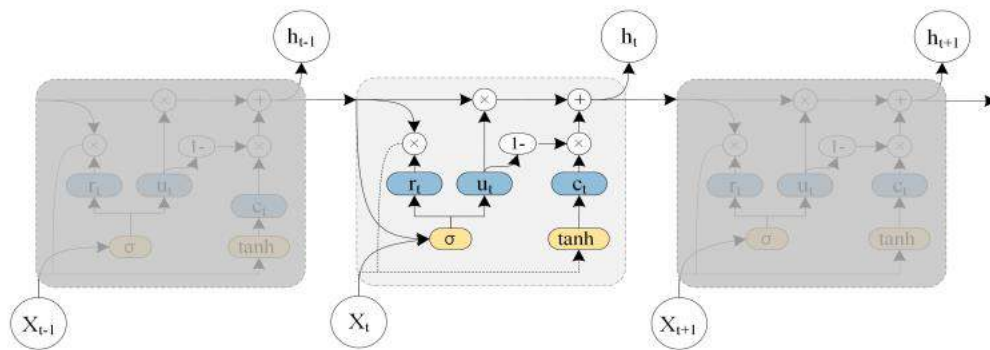


Fig. 5. The architecture of the Gated Recurrent Unit model.

3) Temporal Graph Convolutional Network

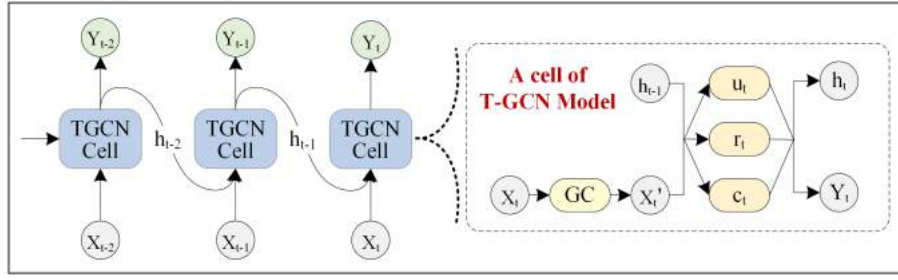


Fig. 6. The overall process of spatio-temporal prediction. The right part represents the specific architecture of a T-GCN unit, and GC represents graph convolution.

如图 6 所示,左边是时空预测的过程,右侧的是 T-GCN cell 的特定结构, h_{t-1} 表示 $t-1$ 时刻的输出, GC 是图卷积过程, u_t 和 r_t 是 t 时刻的更新门和重置门, h_t 表示 t 时刻的输出。(简单来说就是叠加了两个 GCN 层和一个 GRU 层)

4) loss function

加入了 L2 正则化防止过拟合。

$$loss = \| Y_t - \hat{Y}_t \| + \lambda L_{reg} \quad (8)$$

28.5 实验部分

在本节中,我们评估了 T-GCN 模型在两个实际数据集上的预测性能:SZ-taxi 数据集和 Los-loop 数据集。因为这两个数据集都与交通速度有关。不失一般性,在实验部分,我们使用交通速度作为交通信息。

(1) dataset

1) SZ-taxi。该数据集由深圳 2015 年 1 月 1 日至 1 月 31 日的出租车轨迹数据组成,本文选取罗湖区 156 条主要道路作为研究区域。实验数据主要包括两部分。一个是 156×156 的邻接矩阵,它描述了道路之间的空间关系。每一行表示一条道路,矩阵中的值表示道路之间的连接性。另一个是特征矩阵,它描述了每条道路上的速度随时间的变化。每一行代表一条路,每一列是不同时段道路上的交通速度。

我们每 15 分钟计算一次每条路上的车速。

2) Los-loop。该数据集是通过环形探测器在洛杉矶高速公路上实时采集的。我们选择了 207 个传感器和它从 2012 年 3 月 1 日到 3 月 7 日的交通速度。每 5 分钟计算一次交通速度。类似的，数据由一个邻接矩阵和一个特征矩阵组成。利用交通网络中传感器之间的距离计算邻接矩阵。由于数据集包含一些缺失的数据，我们使用线性插值方法来填充缺失的值。

实验中，输入数据归一化为区间[0,1]。另外，80%的数据作为训练集，剩下的20%作为测试集。我们预测未来 15 分钟，30 分钟，45 分钟和 60 分钟的交通速度。

(2) Evaluation Metrics

文章竟然选了 5 个评价指标，也是不怕麻烦~

(1) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N (y_i^j - \hat{y}_i^j)^2} \quad (9)$$

(2) Mean Absolute Error (MAE):

$$MAE = \frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N |y_i^j - \hat{y}_i^j| \quad (10)$$

(3) Accuracy:

$$Accuracy = 1 - \frac{\|Y - \hat{Y}\|_F}{\|Y\|_F} \quad (11)$$

(4) Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{j=1}^M \sum_{i=1}^N (y_i^j - \hat{y}_i^j)^2}{\sum_{j=1}^M \sum_{i=1}^N (y_i^j - \bar{Y})^2} \quad (12)$$

(5) Explained Variance Score (var):

$$var = 1 - \frac{Var\{Y - \hat{Y}\}}{Var\{Y\}} \quad (13)$$

其中，RMSE 和 MAE 用于测量预测误差，数值越小，预测效果越好。

Accuracy 用于检测预测精度，数值越大，预测效果越好。R2 和 var 计算相关系数，衡量预测结果代表实际数据的能力，值越大，预测效果越好。

(3) 选择模型参数

T-GCN 模型的超参数主要包括:学习速度、批量大小、训练时间和隐藏单元数。在实验中, 我们手动调整并设置学习率为 0.001, 批量大小为 32, 训练时间为 5000, 隐藏单元数是 T-GCN 模型的一个非常重要的参数, 不同的隐藏单元数对预测精度有很大的影响。为了选择最优值, 我们对不同的隐藏单元进行了实验, 并通过比较预测结果来选择最优值。在我们的实验中, 对于 SZ-taxi 数据集, 我们从 [8,16,32,64,100,128] 中选择隐藏单元的个数, 分析预测精度的变化。根据实验结果我们对 SZ-taxi 数据集设置了 100 个隐藏单元。采用同样的方法对 Los-loop 数据集设置了 64 隐藏单元。

另外, 使用 Adam 优化器训练 T-GCN 模型。

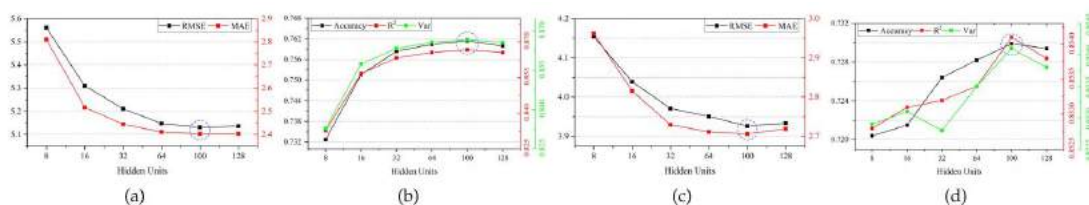


Fig. 7. Comparison of predicted performance under different hidden units in the training and test set based on SZ-taxi dataset. (a) Changes in RMSE and MAE in the training set. (b) Changes in Accuracy, R^2 and var in the training set. (c) Changes in RMSE and MAE in the test set. (d) Changes in Accuracy, R^2 and var based in the test set.

(4) Experimental Results

这篇文章的预测结果给的比较客观, 五个指标中, 并非所有情况下提出的模型都是最好的, 而是在大多数情况下是最好的比较符合实际。(*表示值小到可以忽略不计)。

TABLE I
THE PREDICTION RESULTS OF THE T-GCN MODEL AND OTHER BASELINE METHODS ON SZ-TAXI AND LOS-LOOP DATASETS

T	Metric	SZ-taxi						Los-loop					
		HA	ARIMA	SVR	GCN	GRU	T-GCN	HA	ARIMA	SVR	GCN	GRU	T-GCN
15min	RMSE	4.2951	7.2406	4.1455	5.6596	3.9994	3.9265	7.4427	10.0439	6.0084	7.7922	5.2182	5.1264
	MAE	2.7815	4.9824	2.6233	4.2367	2.5955	2.7117	4.0145	7.6832	3.7285	5.3525	3.0602	3.1802
	Accuracy	0.7008	0.4463	0.7112	0.6107	0.7249	0.7299	0.8733	0.8275	0.8977	0.8673	0.9109	0.9127
	R ²	0.8307	*	0.8423	0.6654	0.8329	0.8541	0.7121	*	0.8123	0.6843	0.8576	0.8634
	var	0.8307	0.0035	0.8424	0.6655	0.8329	0.8541	0.7121	*	0.8146	0.6844	0.8577	0.8634
30min	RMSE	4.2951	6.7899	4.1628	5.6918	4.0942	3.9663	7.4427	9.3450	6.9588	8.3353	6.2802	6.0598
	MAE	2.7815	4.6765	2.6875	4.2647	2.6906	2.7410	4.0145	7.6891	3.7248	5.6118	3.6505	3.7466
	Accuracy	0.7008	0.3845	0.7100	0.6085	0.7184	0.7272	0.8733	0.8275	0.8815	0.8581	0.8931	0.8968
	R ²	0.8307	*	0.8410	0.6616	0.8249	0.8456	0.7121	*	0.7492	0.6402	0.7957	0.8098
	var	0.8307	0.0081	0.8413	0.6617	0.8250	0.8457	0.7121	*	0.7523	0.6404	0.7958	0.8100
45min	RMSE	4.2951	6.7852	4.1885	5.7142	4.1534	3.9859	7.4427	10.0508	7.7504	8.8036	7.0343	6.7065
	MAE	2.7815	4.6734	2.7359	4.2844	2.7743	2.7612	4.0145	7.6924	4.1288	5.9534	4.0915	4.1158
	Accuracy	0.7008	0.3847	0.7082	0.6069	0.7143	0.7258	0.8733	0.8273	0.8680	0.8500	0.8801	0.8857
	R ²	0.8307	*	0.8391	0.6589	0.8198	0.8441	0.7121	*	0.6899	0.5999	0.7446	0.7679
	var	0.8307	0.0087	0.8397	0.6590	0.8199	0.8441	0.7121	*	0.6947	0.6001	0.7451	0.7684
60min	RMSE	4.2951	6.7708	4.2156	5.7361	4.0747	4.0048	7.4427	10.0538	8.4388	9.2657	7.6621	7.2677
	MAE	2.7815	4.6655	2.7751	4.3034	2.7712	2.7889	4.0145	7.6952	4.5036	6.2892	4.5186	4.6021
	Accuracy	0.7008	0.3851	0.7063	0.6054	0.7197	0.7243	0.8733	0.8273	0.8562	0.8421	0.8694	0.8762
	R ²	0.8307	*	0.8370	0.6564	0.8266	0.8422	0.7121	*	0.6336	0.5583	0.6980	0.7283
	var	0.8307	0.0111	0.8379	0.6564	0.8267	0.8423	0.7121	*	0.5593	0.5593	0.6984	0.7290

(5) 模型解释

为了更好的理解 T-GCN 模型，我们在 SZ-taxi 数据集中选择了一条道路，并将测试集的预测结果可视化。以 15 分钟为例。

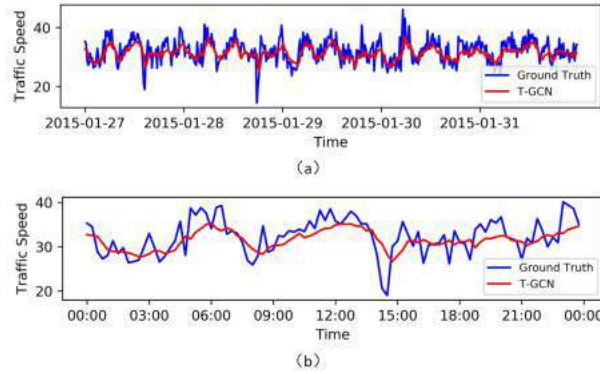


Fig. 12. The visualization results for prediction horizon of 15 minutes.

1) T-GCN 模型对局部极小值/极大值的预测较差。我们推测主要原因是 GCN 模型在傅里叶域中定义了一个平滑滤波器，并通过不断移动滤波器来捕捉空间特征。这一过程导致了整体预测结果的微小变化，使得峰值更加平滑。

2) 实际交通信息与“零出租车值”的预测结果存在一定的误差。“零出租车值”是指由于道路上没有出租车，使得真实值不为零的交通特征矩阵为零的现象。

3) 无论预测时长范围如何, T-GCN 模型总能取得较好的预测结果。T-GCN 模型可以捕获道路交通信息的时空特征, 得出了道路交通信息的变化趋势。此外, T-GCN 模型检测到交通高峰期的开始和结束时间, 并与实际交通速度做出相似的预测结果。这些性质有助于预测交通拥挤和其他交通现象。

29 多步交通需求预测: 基于注意力机制的神经网络模型

29.1 文章信息

《Predicting Multi-step Citywide Passenger Demands Using Attention-based Neural Networks》。

上海交通大学计算机学院发在 2018WSDM (第十一届网络搜索与数据挖掘国际会议。) 上的一篇会议论文。

29.2 摘要

既有研究重点是预测选定地点或热点地区的下一个时间步的乘客需求。然而, 我们认为城市全网的多步乘客需求包含了时变的需求趋势和全局的客流状态, 因此更有利于避免供需不匹配, 并制定有效的车辆分配/调度策略。本文提出了一种端到端深度神经网络模型, 采用基于卷积和 ConvLSTM 单元的 encoder-decoder 框架来识别复杂的特征, 以捕捉时空特性和上下车交互对全市乘客需求的影响。模型中嵌入了注意力模型来刻画潜在的全市出行规律的影响。我们使用出租车和自行车数据集评估模型, 实验结果表明模型效果较好。

29.3 简介

本文认为多步需求预测 (如下图所示) 更具有意义。首先, 多步乘客需求表明了需求的变化趋势, 这有助于避免存在临时需求波动时的冲动性车辆调度响应。相

比之下，短期乘客需求预测结果往往是短时间的，更容易造成不必要的车辆调度。其次，大量的车辆遍布全市。预计整个城市的乘客需求将概括全球状况，从而在实现更好的车辆分配方面提供更多信息。（这一块写的感觉没有说服力）

本文研究了多步城市客流需求预测问题（某个区域上下车客流需求）。这个问题的关键技术挑战是如何处理(1)复杂的时空对乘客需求的影响，以及(2)上下车交互作用。

本文提出了一种端到端的深度神经网络来解决多步城市乘客需求预测问题。我们将某一时段的全市范围内的上下车需求组织成一个 3D 需求张量，并将之前时间区间内的一系列需求张量作为输入。预测模型为 encoder-decoder 框架。在编码阶段，我们使用卷积单元从每个张量中提取空间特征，这些特征有效地捕捉到上下车之间的空间影响和相互作用。然后，我们利用 ConvLSTM 揭示复杂的时空影响，从而得到输入序列的高级表示。解码器的行为与编码器相反，其输出未来的需求张量。过程中嵌入了注意力机制。

文章主要贡献如下：

(1) 据我们所知，文章第一次定义了全网多步预测问题。针对这一预测任务，提出了一种端到端深度神经网络模型。我们的方法采用了一个基于卷积和 ConvLSTM 单元的编码器解码框架，能够有效地捕捉复杂的时空影响和上下车交互。

(2) 引入注意力模型并集成到解码器中，提高了预测性能。

(3) 使用纽约的出租车和自行车数据集验证模型，结果表明，该方法预测效果最好。

29.4 模型框架

本文仍是将某个区域根据经纬度划分为网格。将某个时间段全网所有区域的上下车需求组织为 3D 张量 ($M \in \mathbb{R}(n \times m \times 2)$)，多步预测就是输入过去输入过去多个时间段的需求，输出未来多个时间段的需求。

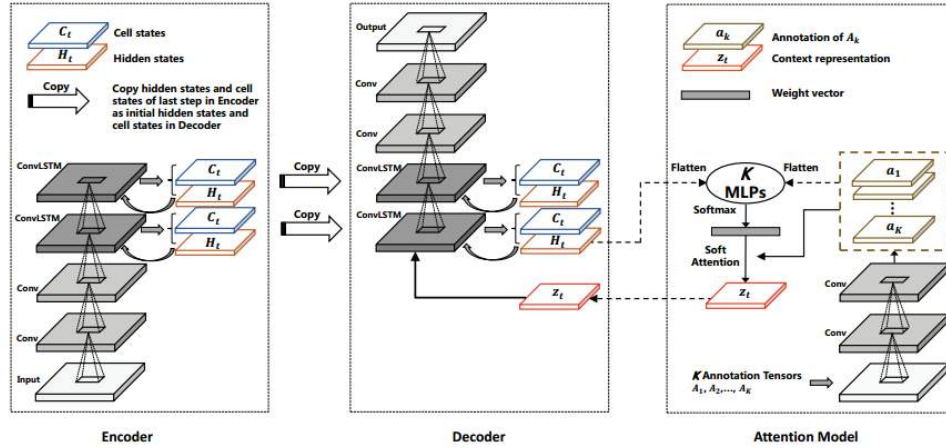


Figure 2: Overview of the Proposed Approach. The Encoder is composed of convolutional units and ConvLSTM units that encode the input sequence into fixed dimensional representations (hidden states and memory states); The Attention Model computes weights for salient parts (annotations) and identifies the attention information (context representation); The Decoder leverages the attention information (context representation) from the attention model and decodes the encoded representations from Encoder to generate future passenger demands.

4.1 overview

图 2 概述了端到端深度神经网络模型。采用 encoder-decoder 框架。首先将乘客需求张量的输入序列“编码”为固定维表示，然后对这些表示进行“解码”，以生成未来所需张量。

研究发现，客流需求分布具有一定的时空规律，这可能是由潜在的全市范围内的客流格局造成的。例如，地铁车站在工作日的高峰时段的需求总是很高，而在午夜的需求却很低。为了捕捉这种规律，我们对历史需求张量执行 k-means 聚类。由此产生的 K 个代表性需求张量称为标注张量。在本文中，我们利用注意力模型将这些标注张量纳入到下一步的需求预测中，这是一种全新的尝试。

4.2 基于 CNN+ConvLSTM 的编码器 encoder

Encoder 输入是前 N 个时间段的三维 tensor。从图 2 可知，encoder 部分其实就是叠加了两个 Conv 层两个 ConvLSTM 层，在 ConvLSTM 层的输出会有 hidden state 和 cell state，该状态会作为 decoder 的初始状态作为输入。

4.3 注意力模型

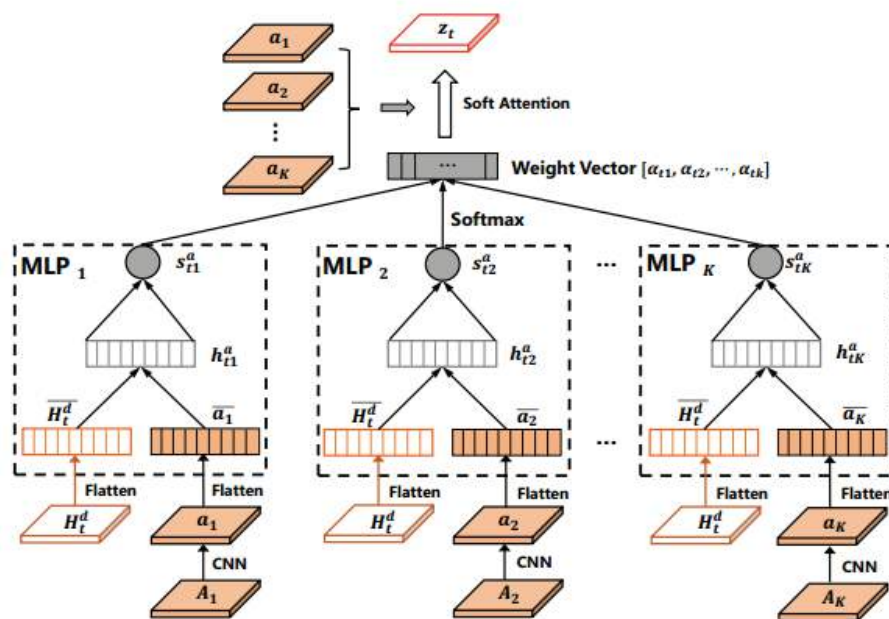


Figure 4: Structure of Attention Model

注意力模型主要分为两步，第一步，先利用 K 均值聚类确定 K 个具有代表性的需求张量 A (3 维张量)，即前文提到的标注张量（这 K 个标注张量应该就是 K 的聚类中心）。然后将该向量通过一个双层的 CNN 提取特征得到 a (也是三维张量)。第二步，获取权重向量。将 a 和上一步 encoder 输出的 hidden state 展平 flatten 后输入到多层神经网络中，输出一个单一值 α ，将 K 个 α 经过 softmax 函数得到归一化的权重矩阵。 K 个权重加权到 K 个 a 上，得到最终的加权后的 Z_t 。(加权这块因为 K 个 a 都是三维张量，权重向量是一维的，即由 K 个权重，所以就是 K 个三维张量的对应位置分别用权重向量进行加权求和，即可得到最终的 Z_t ， Z_t 仍然是三维张量)。

整个计算过程的公式如下。

$$h_{tk}^a = f(W_h^a \overline{H_{t-1}^d} + W_a \overline{a_k} + b_h^a), \quad \forall k \in [1, K] \quad (6)$$

$$s_{tk}^a = f(W_s^a h_{tk}^a), \quad \forall k \in [1, K] \quad (7)$$

$$\alpha_{tk} = \frac{\exp(s_{tk}^a)}{\sum_{k=1}^K \exp(s_{tk}^a)}, \quad \forall k \in [1, K] \quad (8)$$

$$z_t = \sum_{k=1}^K \alpha_{tk} a_k \quad (9)$$

where f is an activation function that could be *ReLU* or *tanh*. The context representation z_t is then fed into decoder.

4.4 decoder 部分

Decoder 的输入即为预测的 B 个时间步，每个时间步都是三维 tensor。Encoder 最终输出的 hidden state 和 cell state 作为 decoder 的初始状态 initial state。另外注意力部分的输出 Z_t 也是 decoder 部分的输出。ConvLSTM 部分的计算公式如下。

$$i_t^e = \sigma(W_{xi}^e * I_{t,L}^e + W_{hi}^e * H_{t-1}^e + b_i^e) \quad (1)$$

$$f_t^e = \sigma(W_{xf}^e * I_{t,L}^e + W_{hf}^e * H_{t-1}^e + b_f^e) \quad (2)$$

$$o_t^e = \sigma(W_{xo}^e * I_{t,L}^e + W_{ho}^e * H_{t-1}^e + b_o^e) \quad (3)$$

$$C_t^e = f_t^e \circ C_{t-1}^e + i_t^e \circ \tanh(W_{xc}^e * I_{t,L}^e + W_{hc}^e * H_{t-1}^e + b_c^e) \quad (4)$$

$$H_t^e = o_t^e \circ \tanh(C_t^e) \quad (5)$$

其输出 H 再经过两个 Conv 层得到最终的输出。

型在解决这一问题上有着巨大的潜力。然而，基于 ConvLSTM 的模型中的卷积递归结构是位置不变的 (location-invariant)，而自然运动和变换(如旋转)通常是位置变化的(location variant)。此外，由于基于深度学习的降水临近预报是一个新兴领域，尚未建立明确的评价方案。针对这些问题，提出了一种新的降水临近预报模型和一个基准。具体来说，本文超越了 ConvLSTM 并提出了 Trajectory GRU (TrajGRU)模型，该模型可以主动利用循环连接学习基于位置变化的结构。此外，文章还提供了一个基准，其中包括来自香港天文台的一个真实世界的大型数据集、一个新的 training loss 和一个促进未来研究和评估是否是 state of the art 的综合评估方案。

30.3 简介

降水临近预报是指利用雷达回波图 1、雨量计等观测数据，结合数值天气预报 (NWP)模型，对局部地区降水强度进行极短时间(如 0-6 小时)预报的问题。

文章作者提出的 ConvLSTM 模型使用卷积结构扩展了 LSTM，结果表明 ConvLSTM 具有更好的时空相关性捕获能力，预测精度高于既有模型。然而，尽管在这个方向上做出了开拓性的努力，这篇论文还是有一些不足之处。

首先，深度学习模型仅在包含 97 个雨天的相对较小的数据集上进行评估，仅比较 0.5mm/h 降雨率阈值下的临近预报技术得分。由于现实降水临近预报系统需要对暴雨等对社会造成更大威胁的强降水事件给予额外的关注，仅在 0.5mm/h 阈值(表示是否下雨)下的性能不足以体现算法的整体性能。事实上，降水临近预报领域的深度学习还处于起步阶段，如何评价模型以满足实际应用的需要还不明确。

其次，虽然 ConvLSTM 中使用的卷积递归结构在捕获时空相关性方面优于全连通递归结构 FC-LSTM，但它不是最优的，还有改进的空间。对于旋转和缩放等运动模式，不同的空间位置和时间戳会导致连续帧的局部相关结构不同。因此，使用位置不变滤波器来表示这种位置变量关系的卷积是低效的。之前的研究尝试通过将一个递归神经网络(RNN)的输出从原始预测修改为输入的某个位置变量变换来解决这个问题，比如光流或动态局部滤波器。然而，通过修正递归结构本身来解决这一问题的研究并不多。

在本文中，我们通过提出降水临近预报的基准和新模型来解决这两个问题。

对于新的基准，我们建立了 HKO-7 数据集，该数据集包含 2009 年至 2015 年香港附近的雷达回波数据。由于雷达回波图在真实场景中是连续的，能够瞬时获取，因此临近预报算法可以采用在线学习的方法来动态地适应新的模式。为此，我们在基准测试中使用了两个测试协议：离线时只能使用一个固定的窗口前面的雷达回波地图，在线时可免费使用所有历史数据和在线学习算法。降水临近预报任务的另一个问题是不同降雨率阈值下降水事件的比例高度不平衡。降雨量较大的情况较少发生，但对现实世界的影响较大。因此，我们提出了平衡均方误差 Balanced Mean Squared Error (B-MSE) 和平衡平均绝对误差 Balanced Mean Absolute Error (B-MAE) 是用于训练和评估的度量方法，它们在计算 MSE 和 MAE 时为较重的降雨分配了更多的权重。结果表明，在多降雨率阈值下，损失函数的平衡变量比原损失函数更符合短时预报的整体性能。此外，我们的实验表明，训练与平衡损失函数是至关重要的深度学习模型，以在较高的降雨率阈值情况下取得良好的性能。

对于新的模型，我们提出了轨迹门控递归单元(TrajGRU)模型，该模型利用子网络在状态转换之前输出状态到状态的连接结构。TrajGRU 允许沿着一些学习轨迹来聚合状态，因此比连接结构固定的 Convolutional GRU (ConvGRU) 更灵活。结果表明，TrajGRU 的性能在一个合成的 Moving MNIST++ 数据集和 HKO-7 数据集中优于 ConvGRU, Dynamic Filter Network (DFN)，以及 2D 和 3D 卷积神经网络。

利用新的数据集、测试方案、训练损失函数和模型，我们对七个模型进行了广泛的评估，同时提供一个大型的降水临近预报基准。实验表明：(1)所有的深度学习模型都优于基于光流的模型；(2)TrajGRU 在所有深度学习模型中整体性能最好；(3)应用在线微调后，在在线设置下测试的模型始终优于在离线设置下测试的模型。据我们所知，这是针对降水临近预报问题的第一个深度学习模型的综合基准。此外，由于降水临近预报可以看作是一个视频预报问题，我们的工作第一次提供了证据和理由，证明在线学习可能在总体上有助于视频预报。

30.4 模型

在本节中，我们提出了降水临近预报的新模型。首先介绍了本文所采用的通用 encoding-forecasting 结构。然后，我们回顾了 ConvGRU 模型，提出了新的 TrajGRU 模型。

4.1 encoding-forecasting 结构

我们采用了与 ConvLSTM 类似的公式，问题描述如下，利用前 j 个观测值预测后 K 步。

We adopt a similar formulation of the precipitation nowcasting problem as in [23]. Assume that the radar echo maps form a spatiotemporal sequence $\langle \mathcal{I}_1, \mathcal{I}_2, \dots \rangle$. At a given timestamp t , our model generates the most likely K -step predictions, $\hat{\mathcal{I}}_{t+1}, \hat{\mathcal{I}}_{t+2}, \dots, \hat{\mathcal{I}}_{t+K}$, based on the previous J observations including the current one: $\mathcal{I}_{t-J+1}, \mathcal{I}_{t-J+2}, \dots, \mathcal{I}_t$. Our encoding-forecasting network first encodes

本文的 encoding-forecasting 结构首先将观测值输入到 N 层 RNN 中，得到 N 个 RNN state:

$$\mathcal{H}_t^1, \mathcal{H}_t^2, \dots, \mathcal{H}_t^n = h(\mathcal{I}_{t-J+1}, \mathcal{I}_{t-J+2}, \dots, \mathcal{I}_t),$$

然后基于这些编码得到的状态，利用另外一个 N 层的 RNN 网络获取预测值:

$$\hat{\mathcal{I}}_{t+1}, \hat{\mathcal{I}}_{t+2}, \dots, \hat{\mathcal{I}}_{t+K} = g(\mathcal{H}_t^1, \mathcal{H}_t^2, \dots, \mathcal{H}_t^n).$$

图 1 展示了网络结构，其中， $n = 3$; $J = 2$; $K = 2$ 。我们在神经网络之间插入了下采样层和上采样层 downsample and upsample，这些层是通过带 stride 步长的 convolution 卷积和 deconvolution 反卷积来实现的。将预测网络的顺序颠倒，是因为高阶状态 high-level state 能够捕捉全局的时空表征，从而指导低阶状态 low-level state 的更新。此外，low-level state 进一步影响预测。这种结构比之前的 ConvLSTM 结构更合理 (ConvLSTM 并没有反向链接预测网络)，因为我们可以自由地在上层插入额外的 RNN 层，而不需要 skip-connection 来聚合底层信息。在这种一般的编码预测结构中，只要有状态张量，就可以选择任何类型的神经网络来代替 RNN，如 ConvGRU 或我们新提出的 TrajGRU。

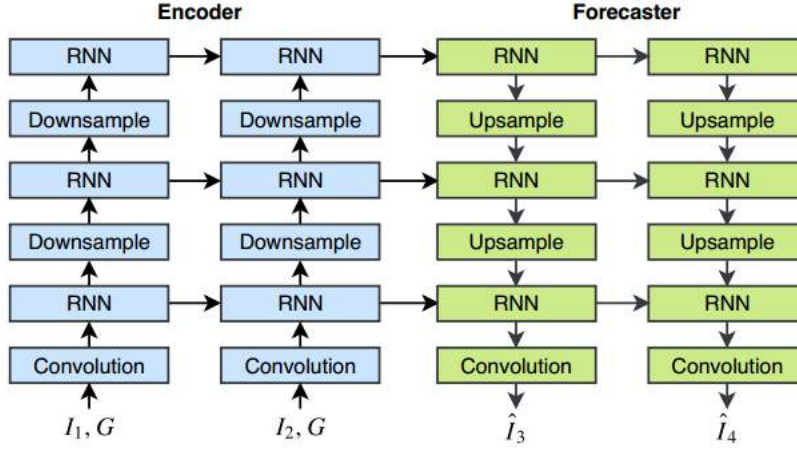


Figure 1: Example of the encoding-forecasting structure used in the paper. In the figure, we use three RNNs to predict two future frames \hat{I}_3, \hat{I}_4 given the two input frames I_1, I_2 . The spatial coordinates G are concatenated to the input frame to ensure the network knows the observations are from different locations. The RNNs can be either ConvGRU or TrajGRU. Zeros are fed as input to the RNN if the input link is missing.

4.2 Convolutional GRU

本文所采用的 ConvGRU 的主要计算公式如下：

The main formulas of the ConvGRU used in this paper are given as follows:

$$\begin{aligned}
 Z_t &= \sigma(W_{xz} * \mathcal{X}_t + W_{hz} * \mathcal{H}_{t-1}), \\
 \mathcal{R}_t &= \sigma(W_{xr} * \mathcal{X}_t + W_{hr} * \mathcal{H}_{t-1}), \\
 \mathcal{H}'_t &= f(W_{xh} * \mathcal{X}_t + \mathcal{R}_t \circ (W_{hh} * \mathcal{H}_{t-1})), \\
 \mathcal{H}_t &= (1 - Z_t) \circ \mathcal{H}'_t + Z_t \circ \mathcal{H}_{t-1}.
 \end{aligned} \tag{1}$$

The bias terms are omitted for notational simplicity. ‘ $*$ ’ is the convolution operation and ‘ \circ ’ is the Hadamard product. Here, $\mathcal{H}_t, \mathcal{R}_t, Z_t, \mathcal{H}'_t \in \mathbb{R}^{C_h \times H \times W}$ are the memory state, reset gate, update gate, and new information, respectively. $\mathcal{X}_t \in \mathbb{R}^{C_i \times H \times W}$ is the input and f is the activation, which is chosen to be leaky ReLU with negative slope equals to 0.2 [18] throughout the paper. H, W are the height and width of the state and input tensors, respectively. Every time a new input arrives, the reset gate will control whether to clear the previous state and the update gate will control how much the new information will be written to the state.

4.3 Trajectory GRU

当用于获取时空相关性时，ConvGRU 或者其他 ConvRNNs 等的不足在于所有位置的连接结构和权值都是固定的。卷积运算基本上是对输入应用一个位置不变的

滤波器。如果输入都是 0，重置门都是 1，我们可以在一个特定的位置 (i, j) 时刻，重写获取新状态的计算过程，具体如下：

reset gates are all one, we could rewrite the computation process of the new information at a specific location (i, j) at timestamp t , i.e., $\mathcal{H}'_{t, :, i, j}$, as follows:

$$\mathcal{H}'_{t, :, i, j} = f(\mathbf{W}_{hh} \text{concat}(\langle \mathcal{H}_{t-1, :, p, q} \mid (p, q) \in \mathcal{N}_{i, j}^h \rangle)) = f\left(\sum_{l=1}^{|\mathcal{N}_{i, j}^h|} \mathbf{W}_{hh}^l \mathcal{H}_{t-1, :, p_{l, i, j}, q_{l, i, j}}\right). \quad (2)$$

Here, $\mathcal{N}_{i, j}^h$ is the ordered neighborhood set at location (i, j) defined by the hyperparameters of the state-to-state convolution such as kernel size, dilation and padding [30]. $(p_{l, i, j}, q_{l, i, j})$ is the l th neighborhood location of position (i, j) . The $\text{concat}(\cdot)$ function concatenates the inner vectors in the set and \mathbf{W}_{hh} is the matrix representation of the state-to-state convolution weights.

当卷积的超参数固定时，邻域集合 \mathcal{N} 对所有位置保持不变。然而，大多数运动模式对于不同的位置有不同的邻域集。例如，旋转和缩放会产生指向不同方向的不同角度的流场。因此，更合理的做法是使用基于位置变化的连接结构 location-variant connection structure:

$$\mathcal{H}'_{t, :, i, j} = f\left(\sum_{l=1}^L \mathbf{W}_{hh}^l \mathcal{H}_{t-1, :, p_{l, i, j}(\theta), q_{l, i, j}(\theta)}\right), \quad (3)$$

where L is the total number of local links, $(p_{l, i, j}(\theta), q_{l, i, j}(\theta))$ is the l th neighborhood parameterized by θ .

基于此，我们提出了 TrajGRU，它使用当前输入和以前的状态在每个时间戳为每个位置生成本地邻域集。由于位置指数是离散的、不可微的，我们用一组连续的光流来表示它们“指标”。TrajGRU 的主要公式如下：

$$\begin{aligned}
\mathcal{U}_t, \mathcal{V}_t &= \gamma(\mathcal{X}_t, \mathcal{H}_{t-1}), \\
\mathcal{Z}_t &= \sigma(\mathcal{W}_{xz} * \mathcal{X}_t + \sum_{l=1}^L \mathcal{W}_{hz}^l * \text{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})), \\
\mathcal{R}_t &= \sigma(\mathcal{W}_{xr} * \mathcal{X}_t + \sum_{l=1}^L \mathcal{W}_{hr}^l * \text{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})), \\
\mathcal{H}'_t &= f(\mathcal{W}_{xh} * \mathcal{X}_t + \mathcal{R}_t \circ (\sum_{l=1}^L \mathcal{W}_{hh}^l * \text{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l}))), \\
\mathcal{H}_t &= (1 - \mathcal{Z}_t) \circ \mathcal{H}'_t + \mathcal{Z}_t \circ \mathcal{H}_{t-1}.
\end{aligned} \tag{4}$$

Here, L is the total number of allowed links. $\mathcal{U}_t, \mathcal{V}_t \in \mathbb{R}^{L \times H \times W}$ are the flow fields that store the local connection structure generated by the structure generating network γ . The $\mathcal{W}_{hz}^l, \mathcal{W}_{hr}^l, \mathcal{W}_{hh}^l$ are the weights for projecting the channels, which are implemented by 1×1 convolutions. The $\text{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})$ function selects the positions pointed out by $\mathcal{U}_{t,l}, \mathcal{V}_{t,l}$ from \mathcal{H}_{t-1} via the bilinear sampling kernel [10, 9]. If we denote $\mathcal{M} = \text{warp}(\mathcal{I}, \mathbf{U}, \mathbf{V})$ where $\mathcal{M}, \mathcal{I} \in \mathbb{R}^{C \times H \times W}$ and $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{H \times W}$, we have:

$$\mathcal{M}_{c,i,j} = \sum_{m=1}^H \sum_{n=1}^W \mathcal{I}_{c,m,n} \max(0, 1 - |i + \mathbf{V}_{i,j} - m|) \max(0, 1 - |j + \mathbf{U}_{i,j} - n|). \tag{5}$$

这种结构的优点是,我们可以学习子网的连接拓扑通过学习参数 γ 。在我们的实验中, γ 需要连接的 \mathcal{X}_t 和 \mathcal{H}_{t-1} 作为输入和固定是 one-hidden-layer 的卷积神经网络 (5×5 的 kernel size 和 32 个 filter)。因此, γ 只有少量的参数并增加几乎没有成本的整体计算量。与 $K \times K$ state-to-state 卷积的 ConvGRU 相比, 由于 $L < K^2$ (K 方), TrajGRU 能够学习到更有效的连接结构。对于 ConvGRU 和 TrajGRU, 模型参数的数量由 state-to-state 的权值大小决定, 对于 ConvGRU 和 TrajGRU 分别为:

$$O(L \times C_h^2) \text{ for TrajGRU and } O(K^2 \times C_h^2) \text{ for ConvGRU.}$$

如果 L 和 K^2 (K 方) 相同, TrajGRU 的参数数量也可以小于 ConvGRU, TrajGRU 模型能够更有效地使用这些参数。ConvGRU 和 TrajGRU 的循环连接结构说明如图 2 所示。

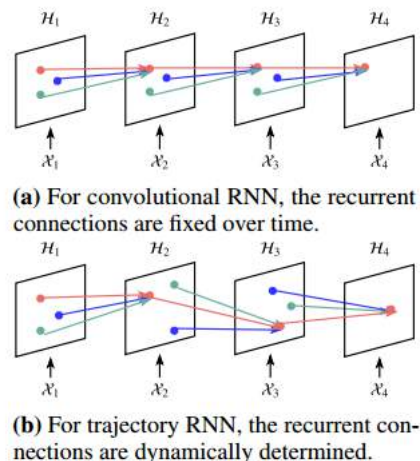


Figure 2: Comparison of the connection structures of convolutional RNN and trajectory RNN. Links with the same color share the same transition weights. (Best viewed in color)

最近，有研究在 CNN 中使用了类似的思想来扩展卷积运算。然而，他们提出的 Active Convolution Unit (ACU) 关注于图像领域，the need for location-variant filters is limited。我们的 TrajGRU 专注于视频，其中 location-variant filters 对于处理旋转等运动模式至关重要。此外，我们正在修改循环连接的结构，并测试了不同数量的链接，而改研究将链接数量固定为 9。

30.5 Experiments on MovingMNIST++

此部分不作介绍。

30.6 降雨量预测的一个新的 benchmark

30.7 HKO-7 Dataset

这个数据是香港天文台提供的 2009 年到 2015 年降雨雷达图数据，图片是 480*480 的灰度图，812 天用于训练、50 天用于验证、131 天用于测试。

首先数据集的转换关系是这样的: dBZ 是雷达的基本反射率 (雷达扫出来的结果), 然后通过 $\left[255 \times \frac{\text{dBZ}+10}{70} + 0.5\right]$ 并裁剪到 $[0, 255]$ 之间转换成普通的 png 图片。这种转换对于最后的训练和测试没有多大的影响, 只是个线性转换, 在深度学习中这个范围才是更常见也更容易解决问题的。

最后还有一种转换关系则是通过 Z-R 关系在 dBZ 和降水值 R (单位是 mm/h) 之间转换: $\text{dBZ} = 10 \log a + 10b \log R$, $a = 58.53, b = 1.56$, a, b 通过线性回归计算。

表 2 显示了数据集中不同降雨概率的分布。

Table 2: Rain rate statistics in the HKO-7 benchmark.

Rain Rate (mm/h)		Proportion (%)	Rainfall Level
$0 \leq x < 0.5$		90.25	No / Hardly noticeable
$0.5 \leq x < 2$		4.38	Light
$2 \leq x < 5$		2.46	Light to moderate
$5 \leq x < 10$		1.35	Moderate
$10 \leq x < 30$		1.14	Moderate to heavy
$30 \leq x$		0.42	Rainstorm warning

30.8 评估方法

随着雷达回波图实时获取, 临近预报算法可以应用在线学习来适应新出现的时空模式。我们在评估方案中提出两种设置:(1)离线设置, 算法始终接收 5 帧作为输入, 提前预测 20 帧;(2)在线设置, 算法依次接收长度为 5 的片段, 每接收一个新片段, 提前预测 20 帧。评估方案在附录中有更系统的描述。测试环境保证在脱机和联机设置中测试相同的序列集, 以便进行公平的比较。

由于降水值分布不均匀，并且大雨对现实生活的影响更大，所以 **使用了一个加权损失函数** 进行训练（这篇文章采用的是 MSE 和 MAE 之和）。

$$w(x) = \begin{cases} 1, & x < 2 \\ 2, & 2 \leq x < 5 \\ 5, & 5 \leq x < 10 \\ 10, & 10 \leq x < 30 \\ 30, & x \geq 30 \end{cases}$$

在数据预处理上，除了以上介绍的，由于原始雷达回波数据还存在大量噪声，作者采用马氏距离去除噪声点，得到一个 mask 之后，只有正常数据才会参与训练和评估。因此整个数据集分成了原始 png 和对应的 mask 两个部分。

具体过程是：计算每一张图片的均值 $\hat{\mu} = \frac{\sum_{i=1}^N x_i}{N}$ 和协方差矩阵 $\hat{\mathbf{S}} = \frac{\sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T}{N-1}$ ，然后计算出马氏距离：

$$D_M(x) = \sqrt{(x - \hat{\mu})^T \hat{\mathbf{S}}^{-1} (x - \hat{\mu})}$$

去除掉马氏距离大于 $\mu + 3\sigma$ 和像素值在 $[0, 71]$ 之间的数据，最后得到一个掩膜就是正常的数据。第一步是为了去除一些离散点，因为存在噪声；第二步则是去除雷达监测范围以外的区域，因为雷达扫描区域是一个圆形的。最后的效果则是：

30.9 总结

在本文中，我们提供了降水临近预报的第一个大规模基准，并提出了一种新的具有学习循环连接结构能力的 TrajGRU 模型。我们已经证明 TrajGRU 比 ConvGRU 更有效地捕捉时空相关性。在未来的工作中，我们计划测试 TrajGRU 是否有助于改善其他时空学习任务，如视觉对象跟踪和视频分割。我们也将尝试使用提出的算法建立一个临近预报系统。

31 共享单车：基于深度学习和多源大数据的浮动共享单车流量预测

31.1 文章信息

《Short-term FFBS demand prediction with multi-source data in a hybrid deep learning framework》。

2019年新发在交通传统期刊 IET Intelligent Transport Systems 上的一篇文章，运输科技四区，审稿周期三个月。

31.2 摘要

本文主要是利用深度学习方法来预测浮动共享单车的短时流量预测 free-floating bike sharing (FFBS)。上海共享单车的出行数据来源于摩拜公司。同时我们还收集了其他数据集，如天气数据和空气质量数据。共享单车需求的时空模式表明，工作日出行呈现出明显的通勤模式，而周末出行通常涉及不同的出行目的。在此基础上，提出了一种混合深度学习神经网络(HDL-net)来预测浮动共享单车的短时需求量（15，20，30分钟）。HDL-net 在早高峰时的性能优于在晚高峰和非高峰时的性能。此外，本文将 HDL-net 与 5 中 benchmark 进行了比较。结果表明，所提出的混合深度学习框架在三个时间区间的预测性能都优于基准模型。本研究的结果对于交通主管部门制定有效的供需平衡策略和自行车道规划方案具有重要的参考价值

31.3 简介

目前，共享单车主要有两种类型，即基于站点的自行车共享和自由浮动的自行车共享(FFBS)。最近，最大的共享单车服务提供商之一的摩拜公司，开始公布部分历史出行记录。摩拜发布的个人出行记录具有较高的地理分辨率，为交通研究者研究共享单车出行需求提供了很好的机会。本文使用了摩拜发布的出行记录数据提出了一种混合深度学习神经网络用于短期共享单车需求预测。提出的混合深度学习架构将从多源数据集成到一个端到端的学习框架中，在探索短期共享单车需求的空间和时间特性方面具有很大的潜力。

31.4 数据集

据报道，上海 2017 年已经有超过 150 万辆共享单车。本文将网格单元作为共享单车需求预测的基本空间单元。考虑到共享单车的出行观测大多位于上海老中心

区，我们将研究区域划分为 5×5 个网格。图 2 显示了研究期间聚集的共享单车需求的空间分布。

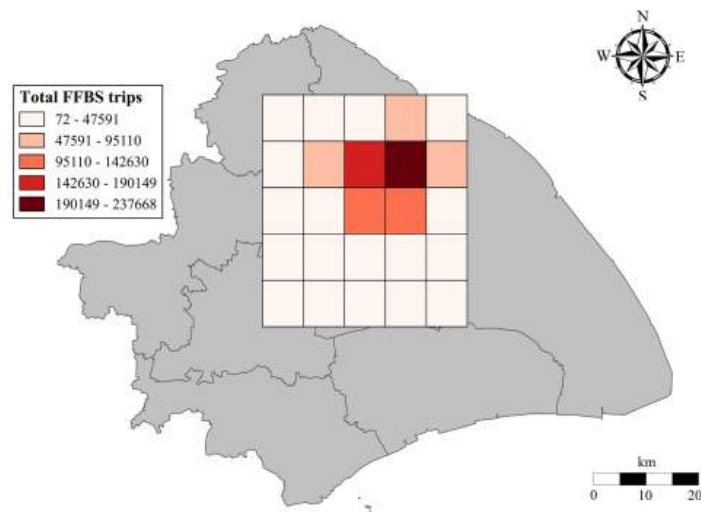


Fig. 2 Spatial distribution of aggregated FFBS demand in Shanghai city

本文收集了三种类型的数据。共享单车出行数据是从摩拜公司收集的，摩拜公司仅通过一些数据科学竞赛，如上海开放数据应用竞赛，以非公开的方式提供部分历史出行记录。如今，摩拜已经有超过 400 万辆自行车，分布在全球 80 个城市，每天产生约 2000 万订单。摩拜单车提供的共享数据包括 2016 年 8 月 1 日至 8 月 31 日期间的 1023,603 次个人出行。数据集中的 trips 由 306,936 个用户和 17688 辆自行车产生。主要使用的数据集字段为 the order ID, bike ID, user ID, pick-up timestamp, pick-up geo-location, drop-off timestamp and drop-off geo-location。而且，行程记录的行程持续时间大于 10 小时或超过 50 公里的行程距离被进一步删除，主要是因为一些用户可能忘记正确锁自行车。最后，本文一共选取了 1023306 个 trips。

气象资料来自上海气象局，由气象站提供的每小时综合天气资料。所得的天气资料包括每小时的总气温、雨量、风速、能见度、湿度及气压。空气质量数据来自国家环境监测中心，获得的每小时空气质量汇总信息包括 SO_2 、 NO_2 、 CO 、 O_3 、颗粒物(PM) 10 和 $\text{pm}_{2.5}$ 的最后，利用 ArcGIS 软件将采集到的共享单车出行数据、天气数据和空气质量数据聚合成相应的网格单元。表 1 总结了考虑的变量的描

述性统计。

Table 1 Descriptive statistics of variables

Variables	Description	Minimum	Maximum	Mean	Standard deviation
FFBS trips 15/20/30 min	number of trips in each grid cell during the predicted time interval	0	395	14.52	30.79
		0	515	19.06	40.31
		0	718	28.11	59.24
temperature	hourly average temperature during the time interval, °C	17.6	38.3	29.471	2.814
precipitation	hourly average precipitation during the time interval, mm	0	48.7	0.077	1.143
wind speed	hourly average wind speed during the time interval, m/s	0	13.9	3.124	1.929
visibility	hourly average visibility during the time interval, m	132	35,000	23,943	8549
humidity	hourly average humidity during the time interval, %	26	98	72.548	14.796
pressure	hourly average pressure during the time interval, hPa	1000	1010.2	1005.22	2.091
SO ₂	hourly AQI of SO ₂ during the time interval	3	10	4.07	1.165
NO ₂	hourly AQI of NO ₂ during the time interval	4	39	11.617	6.505
CO	hourly AQI of CO during the time interval	5	14	7.169	1.326
O ₃	hourly AQI of O ₃ during the time interval	3	109	23.27	14.701
PM 10	hourly AQI of PM 10 during the time interval	9	80	34.75	13.05
PM 2.5	hourly AQI of PM 2.5 during the time interval	9	74	27.341	12.093

31.5 模型框架

这一部分我们构建了一个混合深度学习框架来预测短期共享单车需求。该方法将 LSTM 和 ConvLSTM 结合端到端深度学习体系结构中的神经网络。

注意，本文中与天气和空气质量相关的变量是时间上变化但空间上静止的，而共享单车需求变量在研究期间是空间上和时间上变化的。在提出的混合深度学习框架中，堆叠的 LSTM 层来提取时间特征，并捕获天气和空气质量相关变量之间的时间相关性（可以看成是一个 timesteps 有多个 features，一个 feature 相当于一个指标）。堆叠的 ConvLSTM 层来捕获共享单车需求变量之间的空间和时间特征(见图 3)。然后将这三种类型的数据集的层次特征合并在一起，并输入到多个完全连接的层中，以生成最终的预测共享单车需求。

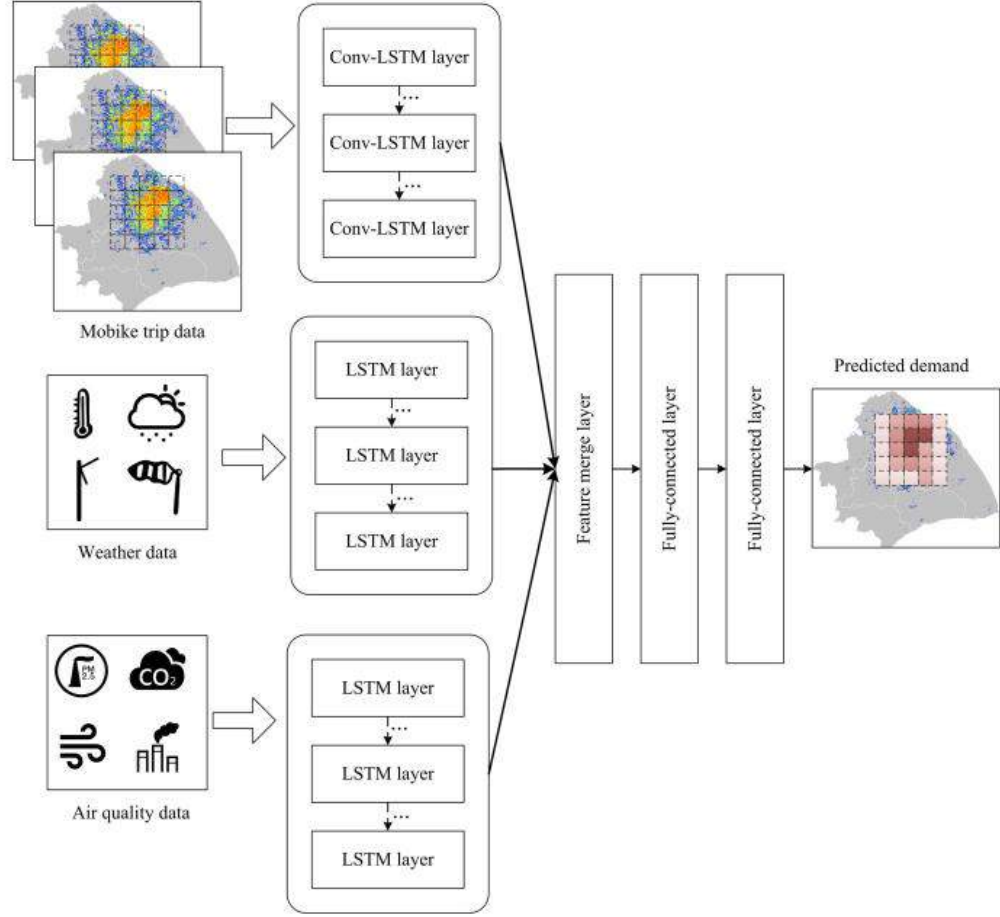


Fig. 3 Hybrid deep learning framework for short-term FFBS demand prediction

堆叠的 LSTM 层和 ConvLSTM 层比较简单，对于特征融合部分，将提取出的天气、空气质量特征和共享单车需求特征串联成一个稠密向量。然后将连接的向量连接到几个完全连接的层，以生成最终预测的共享单车需求。

$$\begin{aligned} \hat{y}_t = & W_{\text{weather}} \mathcal{X}_t^{\text{weather}} + W_{\text{air quality}} \mathcal{X}_t^{\text{air quality}} \\ & + W_{\text{FFBS demand}} \mathcal{X}_t^{\text{FFBS demand}} + b_t \end{aligned} \quad (13)$$

另外本文的目标函数使用了 L2 正则化，评估指标为 MSE，MAE 和 MAPE。

31.6 实验部分

实验部分比较了早晚高峰和平峰的预测结果，五个基准模型也相对比较简单，

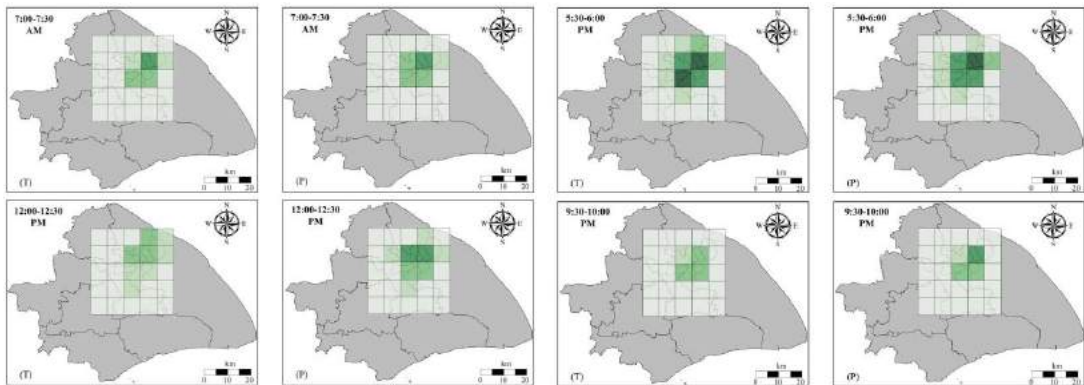


Fig. 9 Comparison of the ground truth (T) and predicted FFBS demand (P) for 30 min prediction task

Table 4 Comparison of different methods for short-term FFBS demand prediction

Models	Time interval								
	15 min			20 min			30 min		
	MSE	MAE	MAPE ^H , %	MSE	MAE	MAPE ^H , %	MSE	MAE	MAPE ^H , %
HDL-net	40.502	3.025	20.843	175.17	6.285	15.82	278.198	7.649	12.841
CNN	81.931	4.153	38.537	258.959	6.798	33.493	391.433	12.217	27.903
LSTM	72.778	4.13	27.133	206.566	6.366	24.586	299.713	9.798	19.121
ANN	84.762	3.867	28.367	221.007	6.789	27.294	356.572	10.484	22.786
GBRT	68.162	3.864	26.864	185.631	6.341	22.419	252.421	8.758	15.182
ARIMA	158.933	5.087	40.798	262.982	7.723	31.609	443.694	14.002	27.721

31.7 数据获取

公众号后台回复“摩拜”，获取上海摩拜共享单车 2016 年八月一个月的轨迹数据样例下载地址。

32 基于手机信令数据的动态出行 OD 及出行需求估计

32.1 文章信息

《Estimating Dynamic Origin-Destination Data and Travel Demand Using Cell Phone Network Data》。

海南大学 2013 年的一篇文章，放到现在似乎也不过时。

32.2 摘要

本研究以大型手机信令数据为基础，开发了手机定位追踪算法，以估计动态的

起点-终点(O-D)交通流量、出行需求量以及通勤客流量。以堪萨斯州地铁走廊为例,分析了使用手机数据来跟踪跨区域(城市)交通活动的可行性,并基于6周的观测数据得出了交通走廊沿线的O-D交通流量、出行需求量和通勤客流量。结果发现,与堪萨斯州地铁沿线的年平均日交通量数据相比,可用的手机信令数据检测到约17.6%的日常交通流量。大约58%的交通流量被确定为所研究走廊的OD流量。这表明大部分的交通来自三个主要地区(堪萨斯城大都市区、托皮卡市、和劳伦斯市),估计的动态出行需求可用于公共交通系统规划和时刻表优化。由于使用的手机信令数据分辨率较低,更利于收集长途或城际出行的交通数据。另外,需要收集更长时间的手机信令数据以增加样本量,这有利于获得稳定的交通流量,减少数据偏差。

32.3 简介

城市的快速发展导致了土地利用、出行方式和交通网络规划的巨大变化。住在郊区,但在市中心或其他邻近城市工作,是美国主要大都市区常见的通勤模式。随着时间的推移,这些相邻城市之间的出行需求预计会增加。因此,了解这些城市之间的交通流和出行需求是交通网络规划和管理的首要任务。传统的方法O-D数据收集通常依赖于家庭电话调查、人口普查数据或每隔5-10年进行一次的路边调查。但是,大多数这些方法在时间和费用方面需要额外的资源,并且可能不提供最新数据和每日数据。

无线网络和移动设备的发展使人们越来越有兴趣使用这些设备来提供有关车辆网络的信息。人们已经开始利用一些无线设备,如GPS,蓝牙和手机,进行交通数据收集和实时交通状况监测的测试。其中,手机的普及率最高,是一个能用于车辆和出行追踪的潜在设备。以往的研究[1-4]评价了手机在出行速度和出行时间估计以及拥挤路段识别方面的应用。

几位研究人员在推导O-D矩阵中检验了不同类型的手机网络数据。white和Wells[5]提出了一项试点研究,该研究调查了在英国肯特地区使用电话账单数据获取O-D信息的可行性。电话账单数据包括信号塔的位置,表明电话是在何时何地

打出的，该研究证明了从手机数据中获取 O-D 信息的可能性。Caceres 等人也评估了使用手机位置数据库来推断 O-D 矩阵的可行性。利用手机通信网络模拟器，模拟并提取手机网络的 O-D 矩阵。Zhang 等人[7]使用模拟的细胞探针轨迹信息估计了每日的 O-D 需求，并通过 VISSIM 仿真验证了该方法。这些结果表明，可以使用手机轨迹作为一种测量方法来估计 O-D 矩阵。然而，这些研究仅限于抽样和模拟测试。没有解决使用实地数据进行数据验证的问题。

本研究以大型手机手机信令为基础，开发了手机定位追踪算法，以估计动态的起点-终点(O-D)交通流量、出行需求量以及通勤客流量。以堪萨斯州地铁走廊为例，分析了使用手机数据来跟踪跨区域(城市)交通活动的可行性，并基于 6 周的观测数据得出了交通走廊沿线的 O-D 交通流量、出行需求量和通勤客流量。下一节介绍手机定位技术和现有的手机信令数据，接下来介绍开发的手机定位追踪算法，包括跟踪数据库生成、O-D 和出行需求估计算法。案例研究部分给出了跟踪过程和结果分析，包括与地面计数流量的比较。最后是结论和建议部分。

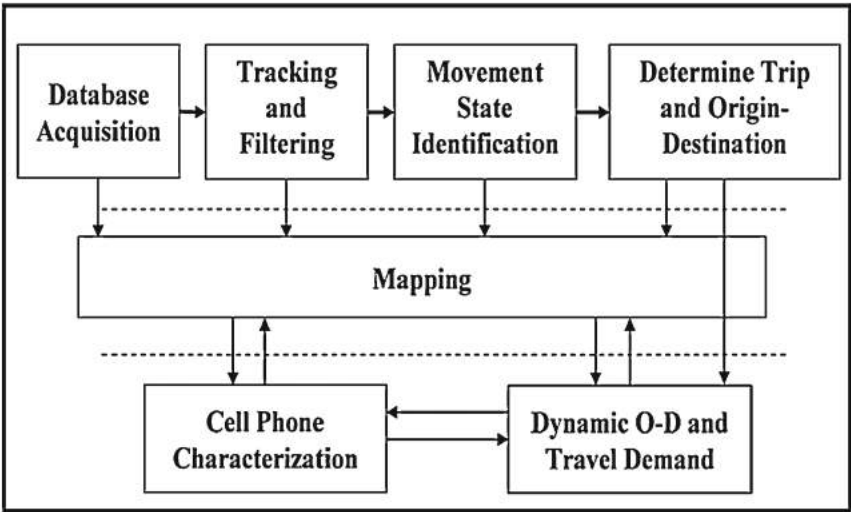


Fig. 1 Cell phone tracking algorithm modules and the operational process

33 时空循环卷积神经网络用于交通速度预测

33.1 文章信息

《Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks》。

北航 2017 年发在 sensors 上的一篇文章。

33.2 摘要

近几十年来，大规模交通网络流量预测已成为一个重要而具有挑战性的课题。受运动预测领域的启发，提出了一种保留交通网络精细结构的网格表示方法。将整个网络的流量速度转换为一系列静态图像，并将其输入到一个新的深度架构中，即时空循环卷积网络(SRCNs)，用于流量预测。所提出的 SRCNs 继承了深卷积神经网络(DCNNs)和长短时记忆(LSTM)神经网络的优点。DCNNs 可以捕获整个网络流量的空间依赖性，LSTMs 可以学习流量的时间依赖性。以北京某交通网络为例，通过 278 个节点的实验，证明了 SRCNs 在短期和长期交通预测方面都优于其他基于深度学习的算法。

33.3 简介

在计算机视觉领域，深度学习比传统的图像处理方法具有更好的性能。运动预测中的深度学习是基于同一物体的一系列历史场景来预测物体未来运动的研究领域。基于该方法的成功，通过基于网格的分割方法，将全网范围内的交通速度作为静态图像的集合，其中每个像素代表一个路段或多个路段的交通状况。随着时间的推移，整个网络的流量也在变化。给定一个包含动画的静态图像序列，我们能预测每个像素的未来运动吗？深度学习框架在提高运动预测精度方面具有优势。在学习视频序列时，应同时考虑空间和时间的长期依赖性。卷积神经网络(CNNs)采用带卷积滤波器的层，通过滑动窗口提取局部特征，可以对附近或全市范围的空间依赖关系进行建模。为了学习长时间跨度的时间序列，Hochreiter 和 Schmidhuber 于 1997 年提出的长短时记忆(long short-term memory, LSTM)神经网络(NNs)在短时交通预测中得到了有效的应用，并在捕捉交通流的长期时间依赖性方面取得了显著的效果。基于 CNNs 和 LSTMs 的成功，本文提出了一种基于时空图像的方法，利用

时空循环卷积网络(SRCNs)来预测整个网络的流量状态。利用深度卷积神经网络(Deep convolutional neural networks, DCNNs)挖掘整个交通网络中各链路之间的空间特征, 利用 LSTMs 学习交通拥堵演化的时间特征。将时空特征输入全连通层, 学习大规模交通网络中各环节的交通速度模式, 并对模型进行端到端的训练。

本文的贡献总结如下:

- 开发了一个名为 SRCN 的混合模型, 它结合了 DCNNs 和 LSTMs 来预测整个网络的流量速度。
- 提出了一种新的交通网络表示方法, 它可以在较好的尺度上保持交通网络的结构。
- 将网络流量的特殊时间特征建模为一个视频, 其中每个流量条件作为视频的一帧。在提出的 SRCN 体系结构中, DCNNs 从网络的角度捕获近端和远端空间依赖关系, 而 LSTMs 学习长期的时间依赖关系。通过 DCNNs 与 LSTMs 的集成, 分析了整个网络的时空交通数据。

33.4 主体内容

(1) 网络表示

假设要预测交通网络中每个路段的拥塞情况, 用 y 表示, 共 n 个路段。

Step1: 选择一个交通网络(请参阅图 2a), 根据道路情况把它分成 n 个路段, 并根据公式(1)计算这些路段在一个特定的时间段内的平均速度 (2 分钟), 其中 m 和 v_j 代表第 i 个路段上的车辆数和车辆的平均速度, 不同的颜色表示速度(如图 2b 所示)。图 3a 显示了一个小型网络的示例。

$$v_i = \frac{\sum_{j=1}^m \bar{v}_j}{m} \quad (1)$$

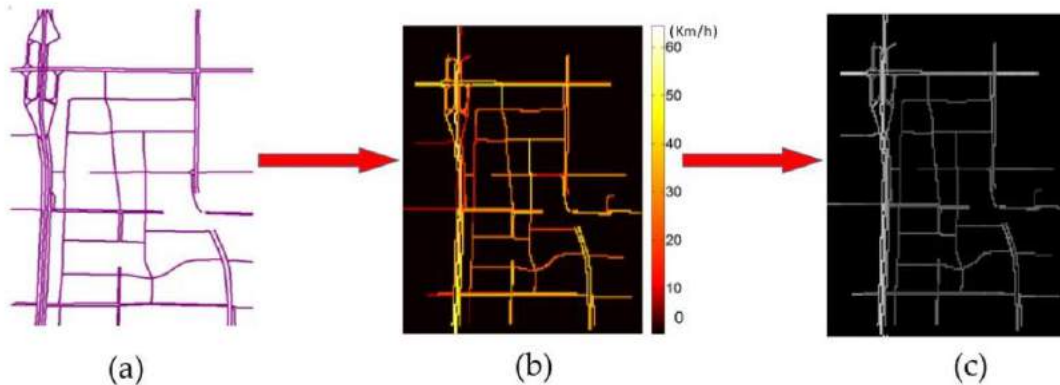


Figure 2. Grid-based transportation network segmentation process: (a) A transportation network without traffic information; (b) the speed of each link is mapped and colored in the network; (c) the speed of each link is normalized and greyed in the network.

Step2: 使用一个小网格划分交通网络，网格的大小约为 $0.0001^\circ \times 0.0001^\circ$ ，度表示经纬度，在北京， 0.0001° 约代表 10 米。划分出来的每个网格代表一个空间区域。

Step3: 将平均速度映射到网格。空白区域的值设置为 0;如果多个路段通过一个网格，便将它们的平均速度分配给相应的网格(如图 3c 所示)，并将速度归一化到 (0,1)区间内(如图 2c 所示)。

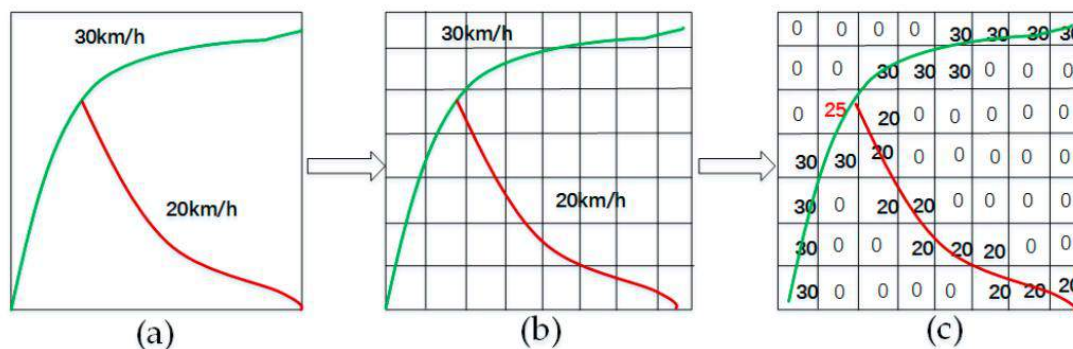


Figure 3. Traffic speed representation in a small-scale transportation network: (a) two links with different speeds in a transportation network; (b) the network is divided into several grids (the size of each grid is approximately $10 \text{ m} \times 10 \text{ m}$); (c) the speed of each link is mapped to each grid.

使用基于网格的网络分割方法，保持不同路段之间的相对拓扑不变。这种处理可以保留道路的几何信息，如细粒度上的 u 型急转弯和交叉口。

(2) 空间信息提取

空间信息可以通过深度卷积层来提取，因为路段之间的距离会因为卷积和池化过程而缩短图 4a 中每个网格框表示一个空间区域(类似于 step2 中的网络表示)，透明的绿色区域表示一个 3×3 卷积滤波器，相同颜色的线表示两个相邻的路段，不同颜色的线表示两个相距较远的路段。通过 CNN 的卷积和池化过程，图 4b 中的蓝线和红线之间的距离比图 4a 中的短。这些抽象特征对预测问题具有重要意义。

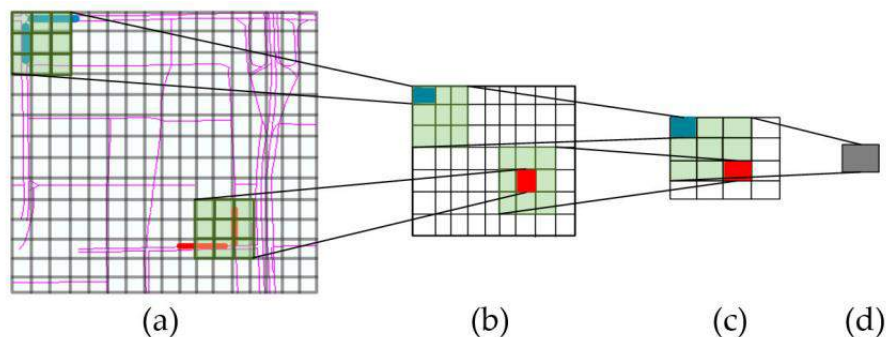


Figure 4. Convolutions for capturing near and far dependencies: (a) the original transportation network; (b) the transformed network with the first convolution; (c) the transformed network with the second convolution; (d) the transformed network with the third convolution.

本文所使用的网络结构即为 2 层 Conv2D 再加全连接层。

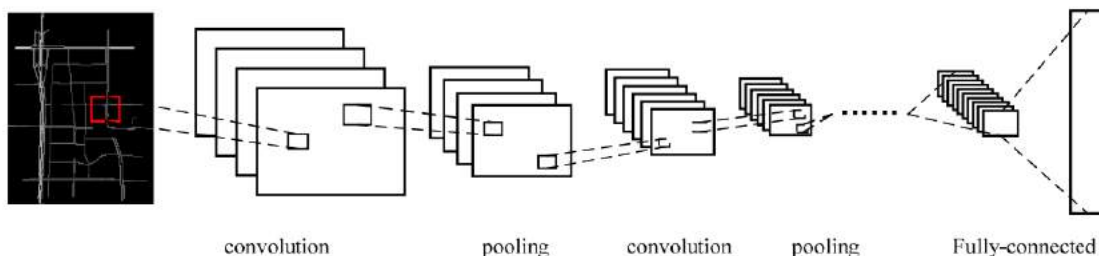


Figure 5. Structure of the DCNNs.

(3) 时间信息提取

时间信息即使用 LSTM。

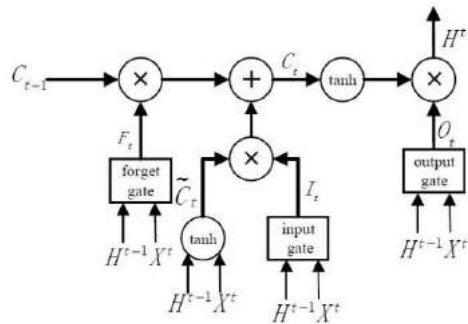


Figure 6. LSTM NN architecture.

(4) 时空循环卷积网络

本文所使用的网络结构为一个 DCNN（包括两层 Conv2D 和一层全连接层）、两个 LSTMs 和一个全连接的层组成。

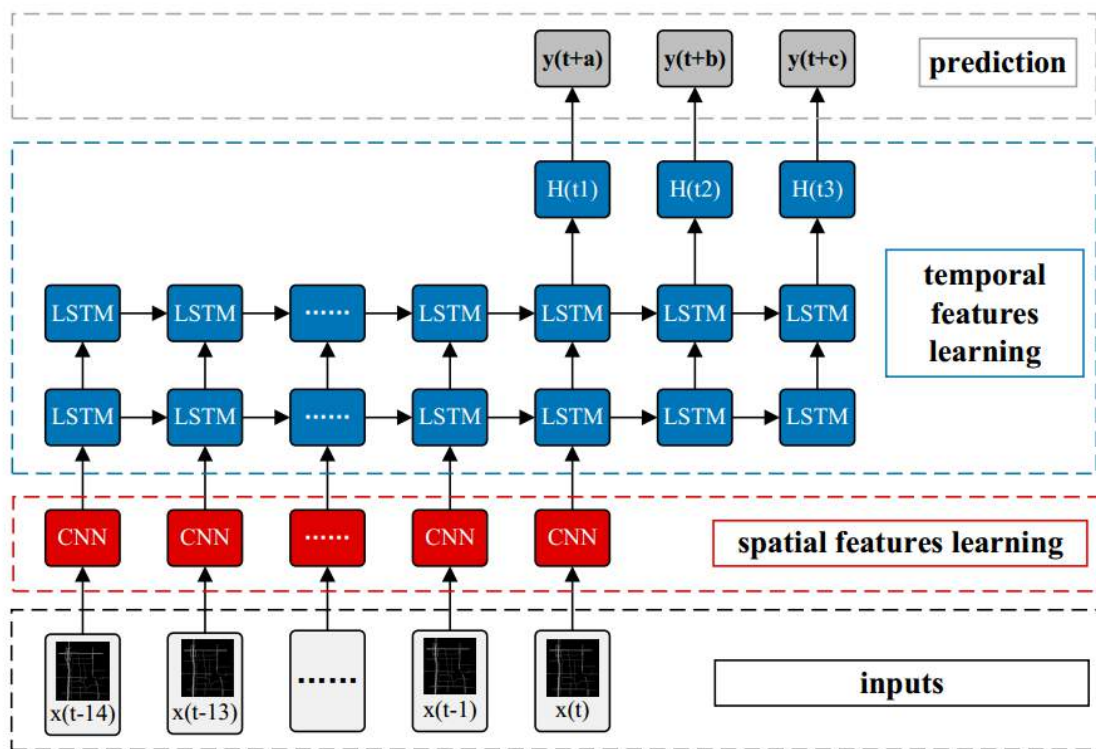


Figure 7. Framework of SRCNs.

33.5 总结

文章相对比较简单，难点在于数据预处理，不过数据预处理这块也可以一劳永逸。文章是 17 年的，在那会相对来说还比较新颖。

文章所使用的数据为 2015 年 6 月 1 日至 2015 年 8 月 31 日共 92 天的北京出租车 GPS 数据。所使用的交通网络并非北京市全网，也是二环到三环之间共 278 个路段。对于所有方法，将 time lag 设置为 15，这表示使用 $15 \times 2 = 30$ min 预测未来交通状态。

Table 2. Comparison of different methods in terms of short-term prediction.

Time Steps Algorithm	2 min		4 min		6 min		Average Error	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
SRCNs	0.1269	4.9258	0.1271	5.0124	0.1272	5.0612	0.1270	4.9998
LSTMs	0.1630	6.1521	0.1731	6.8721	0.1781	7.0016	0.1714	6.7527
SAEs	0.1591	6.2319	0.1718	6.8737	0.1742	7.2602	0.1684	6.7886
DCNNs	0.1622	6.6509	0.1724	6.8516	0.1775	7.2845	0.1707	6.9290
SVM	0.1803	7.6036	0.2016	8.0132	0.2123	8.2346	0.1984	7.9505

34 结束语

未完，待续.....

未来，可期.....