

# DeepSTN+: Context-aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis

## 文章简介

作者: Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, Depeng Jin

作者机构: 清华大学

发表期刊/会议: AAAI2019

采用网格流量预测方法，对比17年郑宇的ST-ResNet，抛出了三个问题: (1)卷积捕获的空间范围小; (2)人口流动和区域的功能相关; (3)之前的融合机制不好。主要在方法上的改进是: 改了一下残差卷积，给POI信息增加了时间维度，多组件的信息提前融合，减少了参数，稳定了模型训练。

## 摘要概括

人口流量预测在城市规划、交通管控中的很多应用中都很重要。目的是预测流入和流出流量。我们提出了DeepSTN+，一个基于深度学习的卷积模型，预测超大城市的人口流量。首先，DeepSTN+ 使用 *ConvPlus* 结构对大范围的空间依赖建模。此外，POI 分布和时间因素相融合来表达区域属性的影响，以此引入人口流动的先验知识。最后，我们提出了一个有效的融合机制来稳定训练过程，提升了结果。基于两个真实数据集的大量实验结果表明我们模型的先进性，和SOTA相比比高了 8% ~ 13% 左右。

## 背景介绍概括

现有的基于grid region进行城市人流预测的方法存在的问题主要有：

- (1) 不能捕获区域间的空间依赖。由于现代城市中高级的运输系统的存在，人们可以通过地铁或出租车在短时间内移动到很远的地方。因此，区域间的大范围空间依赖在人口移动中逐渐扮演重要的角色。现存的工作使用多层卷积网络来建模。然而，它们只能一步一步地捕获近邻的空间依赖，不能直接地捕获大范围的空间依赖。
  - (2) 忽略了人口流动的区域功能的影响。人口移动是发生在物理世界中的，会直接受到区域属性的影响。举个例子，人们通常早上从家出发到公司，晚上回来。显然，区域的功能（属性）包含了关于人类移动的先验知识。然而，现存的解决方案没有考虑过区域的属性。
  - (3) 冗余以及不稳定的神经网络结构。ST-ResNet 利用了三个独立分支，每个分支都是残差卷积单元，用来处理不同的输入，在模型的结尾用一个线性操作融合三个输出。但是，最后的融合机制导致不同组件间的交互产生了缺陷，这个缺陷导致了网络内产生了无效的参数和不稳定的性质。
- 总结一下，模型应该考虑大范围的空间依赖，区域的影响，更有效的融合机制这三点因素。我们提出的DeepSTN+ 解决了上述挑战。我们设计了一个 *ConvPlus* 结构直接地捕获大范围空间依赖。*ConvPlus* 放在残差单元前面作为一个全局特征提取器提取出区域间的全局特征。其次，我们设计了一个 *SemanticPlus* 结构来学习人口在区域间移动的先验知识。用静态的 POI 分布作为输入，*SemanticPlus* 利用时间因素给不同时间上不同的 POI 分配权重。最后，我们引入早融合和多尺度融合机制来减少训练参数，捕获不同级别特征间的复杂关系。这样，我们的系统可以对更复杂的空间关联性建模，获得更好的效果。

### 研究问题相关定义

Definition 1. (Region (Zhang et al. 2016)). 为了表示城市的区域，我们基于经纬度将城市划分成  $H \times W$  个区域，所有的网格大小相同。

Definition 2. (InFlow/OutFlow (Zhang et al. 2016)). 为了表示城市内的人口流动，我们定义了region  $(h, w)$  在时间段  $i$  的流入和流出流量:

$$\begin{aligned} x_i^{h,w,in} &= \sum_{Tr_k \in P} |j > 1 | g_{j-1} \notin (h, w) \& g_j \in (h, w) | \\ x_i^{h,w,out} &= \sum_{Tr_k \in P} |j \geq 1 | g_{j-1} \in (h, w) \& g_j \notin (h, w) | \end{aligned} \quad (1)$$

这里  $P$  表示时段  $i$  的轨迹集合。  $T_r : g_1 \rightarrow g_2 \rightarrow \dots g_{|T_r|}$  是  $P$  中的一条轨迹，  $g_i$  代表一个经纬度坐标。  $g_j \in (h, w)$  表示点  $g_j$  在网格  $(h, w)$  内，反之则表示不在网格  $(h, w)$  之内。

Definition 3. (Crowd Flow Prediction). 给定历史观测值，  $\{X_i | i = 1, 2, \dots, n - 1\}$ ，预测  $X_n$ 。

### 本文提出的模型

图 1 展示了我们模型的框架。主要有三个部分：流量输入、SemanticPlus 和 ResPlus 单元。流量输入包含 *closeness*, *period*, *terend*，由于数据的时间范围限制可以减少为 *closeness*, *period*。SemanticPlus 包含 POI 分布和时间信息。ResPlus 单元可以捕获远距离空间依赖。每个区域的流入和流出流量通过每小时或者每半小时统计得到流量地图的时间序列。这些流量地图通过 Min-Max 归一化处理到  $[-1, 1]$ 。如图 2 所示，人口分布地图通过近期时间、近邻历史、远期历史选择后作为输入放入模型。不同类型的 POI 分布通过 Min-Max 归一化到  $[0, 1]$ 。如图 2 做部分所示，POI 分布地图通过时间信息赋予了不同的权重。之后，POI 信息和人流信息通过早融合后放入堆叠的 ResPlus 单元中。最后，ResPlus 单元不同级别的特征融合后进入卷积部分，然后通过 Tanh 映射到  $[-1, 1]$ 。下面会介绍细节。

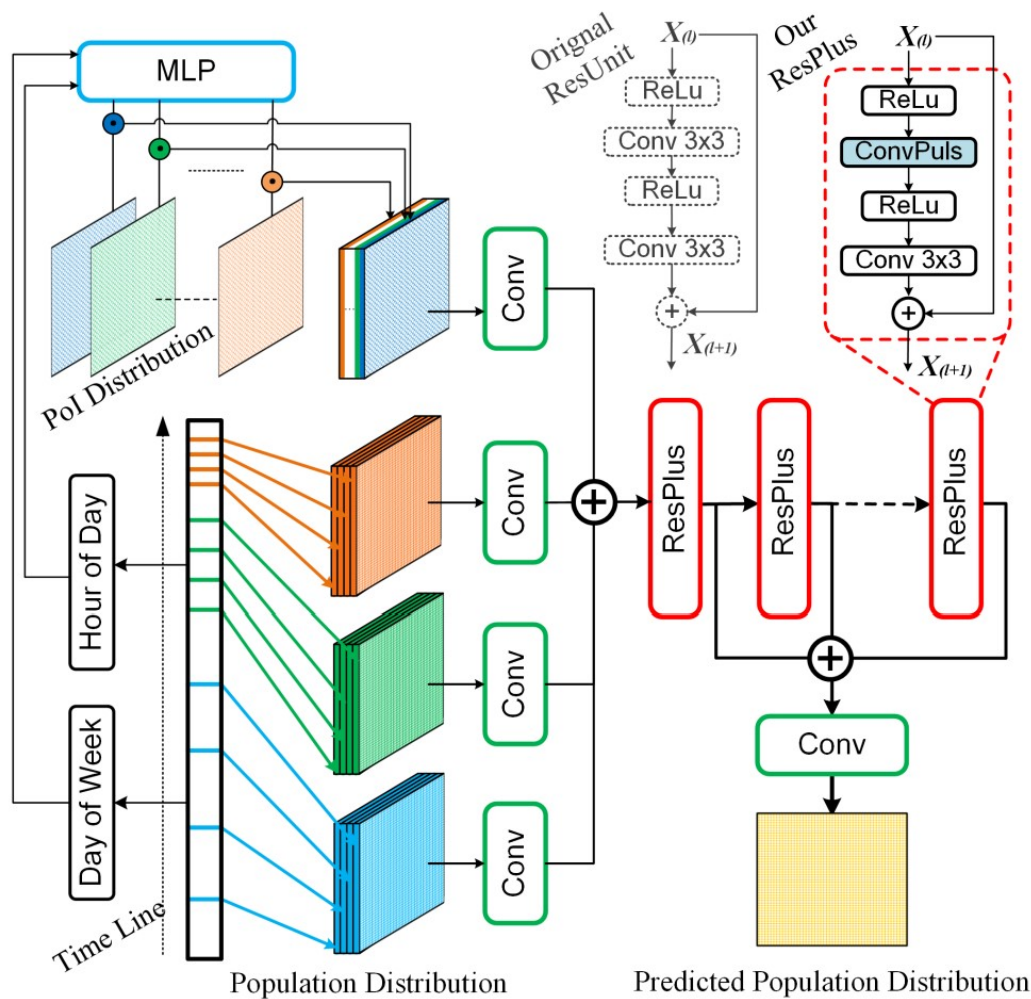


Figure 2: DeepSTN+ architecture, where Conv is Convolution; ResPlus is ResPlus Unit; MLP is Multi-Layer Perceptron.

#### ResPlus模块

很多处理人口流量预测的深度学习模型主要包含两个部分：基于 RNN 的结构，像 ConvLSTM 和 Periodic-CRN，以及基于 CNN 的结构，如 Deep-ST 和 ST-ResNet。但是，训练基于 RNN 结构的模型费时。因此我们选用基于 CNN 的结构 ST-ResNet 作为我们的基础模型。

在这篇论文中，我们设计 ConvPlus 来捕获城市内远距离的空间依赖。如图 2，ResPlus 单元使用一个 ConvPlus 和一个典型卷积。我们尝试了 Batch Normalization 和 Dropout，为了简介没有在图里面画出来。

典型卷积的每个通道对应一个卷积核。卷积核使用这些核来计算地图上的互相关系数，比如捕获梯度上的特征。卷积核的大小一般很小。在 ST-ResNet 和 DeepSTN+ 里面，卷积核的大小是  $3 \times 3$ 。但是城市中存在着远距离的依赖。人们可能坐地铁去上班。我们称这类关系叫远距离空间依赖关系。这种关系使得堆叠卷积难以有效地捕获这个关系。

如图 2 左部分所示，在 ConvPlus 结构中，我们将典型卷积的一些通道分离来捕获每个区域的远距离空间依赖。然后用一个全连接层直接捕获每两个区域之间的远距离空间依赖，在这层前面用一个池化层来减少参数。因此，在 ConvPlus 的输出有两类通道。ConvPlus 的输出有着和普通卷积一样的输出，可以用于下一个卷积的输入。

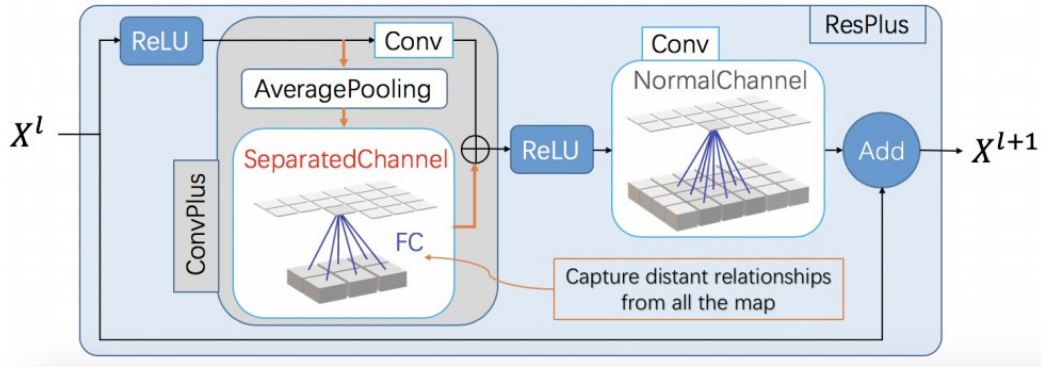


Figure 3: The architecture of ResPlus unit, where FC denotes a fully-connected layer. An ordinary Conv (right) and a ConvPlus (left) are shown in this figure. Normal channels capture close relationships while separated channels capture long-range spatial dependence. An average pooling layer is used to reduce the number of parameters.

#### SemanticPlus模块

POI 在人口流动上有很强烈的影响，这些影响随时间变化而变化。因此，我们继承这个先验知识到模型内。我们手机了包括类型、数量、位置的 POI 信息。然后统计每个网格内 POI 的数量，使用一个一维向量表示每种 POI 的分布。图 5 展示了北京的流量分布地图和餐饮分布地图。它们的分布很相似，并且互相关系数有 0.87，暗示了它们之间的潜在关系。我们使用一个时间向量来表示每个人口流量地图的时间。时间向量包含两个部分：一个 one-hot 向量表示一天中的各个时间，如果时段按小时走，那长度就是 24；另一个 one-hot 向量表示是一周中的哪天，长度是 7。一个时间向量拼接了这两个向量。为了建模对流量地图有变化的时间影响的 POI 信息，我们将时间向量转换为 POI 的影响强度。我们使用大小为  $PN \times H \times W$  的  $X^s$  来表示 POI 地图 ( $PN$  表示 POI 类数， $H$  和  $W$  代表 grid region 的行和列)；使用一个大小为  $PN$  的向量  $R$  表示 POI 的影响强度。公式表达式为:  $S = X^s \times R$ 。

#### Fusion

与郑宇的 ST-ResNet 一样，建模时考虑三种不同的时间尺度的特征: closeness, period, trend，分别用  $X^c$ 、 $X^p$  和  $X^t$  表示。本文作者认为， $X^c$ 、 $X^p$  和  $X^t$  三者应该用更复杂的融合方式，而不是线性组合的方式 (郑宇等人)。这些带有 POI 信息的流量信息也有复杂的交互。为了建模这种相互影响，我们使用早融合而不是末端融合使得不同的信息能更早的融合起来。早融合减少了大约三分之二的参数。此外，ST-ResNet 有些时候不能收敛。我们发现这个问题可以通过早融合减少参数来简化模型解决。考虑到不同层的特征有不同的函数，我们在模型末端设定了一个多尺度的融合机制。这里我们通过公式来描述整个 DeepSTN+ 网络模型:

$$\hat{X} = f_{con}(f_{Res}(f_{EF}(X^c + X^p + X^t + S)))$$



函数  $f_{EF}$  表示融合使用的卷积操作；函数  $f_{con}$  表示最后一步的多尺度融合，表示卷积层后的一个concat层。 $S$ 表示SemanticPlus的输出，即带有时间权重的POI分布。

### Model Training

算法 1 描述了训练过程。前 7 行是构建训练集和 POI 信息，模型通过 Adam Optimization训练。

Procedure: DeepSTN+ Training Procedure
<b>Input:</b> historical observations: $\{\mathbf{X}_0, \cdots, \mathbf{X}_{n-1}\}$ ; PoI distributions: $\mathbf{X}^s$ ; time-vector: $\{\mathbf{I}_0, \cdots, \mathbf{I}_{n-1}\}$ ; length of closeness, period, trend sequences: $l_c, l_p, l_t$ ; period span: $p$ ; trend span: $t$ .
<b>Output:</b> Learned DeepSTN+ model
// construct the training data $\mathbb{D}$
1 $\mathbb{D} \leftarrow \emptyset$
2 <b>for</b> all available time interval:
3 $\mathbf{X}_i^c = [\mathbf{X}_{i-l_c}, \mathbf{X}_{i-(l_c-1)}, \cdots, \mathbf{X}_{i-1}]$
4 $\mathbf{X}_i^p = [\mathbf{X}_{i-l_p \cdot p}, \mathbf{X}_{i-(l_p-1) \cdot p}, \cdots, \mathbf{X}_{i-p}]$
5 $\mathbf{X}_i^t = [\mathbf{X}_{i-l_t \cdot t}, \mathbf{X}_{i-(l_t-1) \cdot t}, \cdots, \mathbf{X}_{i-t}]$
6   put an training instance $(\{\mathbf{X}_i^c, \mathbf{X}_i^p, \mathbf{X}_i^t, \mathbf{X}^s, \mathbf{I}_i\}, \mathbf{X}_i)$ into $\mathbb{D}$
7 <b>end</b> // $\mathbf{X}_i$ is the target at time $i$
// train the model
8 initialize all learnable parameters $\theta$ in DeepSTN+
9 <b>repeat</b>
10   randomly select a batch of instances $\mathbf{D}$ from $\mathbb{D}$
11   optimize $\theta$ using Adam and $\mathbf{D}$
12 <b>until</b> model overfitting

### 实验说明及结果对比

使用了两个数据集，一个是公开数据集BikeNYC，另一个是自己独家的数据集MobileBJ。

MobileBJ: 该数据是中国一个很大的移动网络供应商提供的，时间范围是2018年4月1日~4月30日。记录了用户请求区域服务时的位置。我们用定义 2 转换成了网格流量。我们选择最后一周的数据作为测试集，前面的作为训练集。收集了17类的POI信息。

BikeNYC: 纽约的共享单车数据集。时间范围是2014年4月1日~9月30日。数据包含了旅途时长，出发和到达站的 ID，起始和结束时间。最后 14 天的数据用来测试，其他的训练。我们选了 9 类 POI 信息。

Dataset	MobileBJ	BikeNYC
Data type	Mobile application	Bike rent
Location	Beijing	New York
Time span	4/1/2018-4/30/2018	4/1/2014-9/30/2014
Time interval	30 minutes	1 hour
Grid map size	(19,21)	(21,12)
PoI Num	264581	26202

Table 1: Datasets

实验对比的Baseline Methods:

- HA
- VAR
- ARIMA
- ConvLSTM
- ST-ResNet

评价指标:

(1)RMSE (2)MAE

$$\begin{aligned}
 RMSE &= \sqrt{\frac{1}{T} \sum_{i=1}^T ||X_i - \hat{X}_i||_2^2} \\
 MAE &= \frac{1}{T} \sum_{i=1}^T |X_i - \hat{X}_i|
 \end{aligned}
 \tag{2}$$

实验结果展示

Model	RMSE	$\Delta$	MAE
HA	136.32	223.14%	51.60
VAR	62.75	48.76%	44.27
ARIMA	58.63	28.43%	30.05
ConvLSTM	44.31	5.04%	27.75
ST-ResNet	42.19	0	26.95
DeepSTN	39.85	-5.54%	26.53
DeepSTN+plus	37.69	-10.67%	23.85
DeepSTN+PoI	39.12	-7.27%	25.87
DeepSTN+PoI*time	37.62	-10.83%	24.89
DeepSTN+plus+PoI*time	<b>36.29</b>	<b>-13.97%</b>	<b>22.94</b>
DeepSTN+plus+PoI*time+con	<b>36.89</b>	<b>-12.56%</b>	<b>23.43</b>

Table 4: Comparison among different baselines and variants of DeepSTN+ on MobileBJ.

Model	RMSE	$\Delta$	MAE
HA	7.885	21.79%	2.823
VAR	10.097	55.94%	5.49
ARIMA	10.894	68.25%	3.246
ConvLSTM	6.412	-0.97%	2.543
ST-ResNet	6.475	0	2.395
DeepSTN	6.213	-4.05%	2.388
DeepSTN+plus	6.128	-5.36%	2.362
DeepSTN+PoI	6.191	-4.39%	2.381
DeepSTN+PoI*time	6.021	-7.01%	2.340
DeepSTN+plus+PoI*time	<b>5.984</b>	<b>-7.58%</b>	<b>2.292</b>
DeepSTN+plus+PoI*time+con	<b>5.955</b>	<b>-8.03%</b>	<b>2.285</b>

Table 5: Comparison among different baselines and variants of DeepSTN+ on BikeNYC.

