



# Deep Learning Based Wireless Traffic Prediction for Green Cellular Networks

张传亭  
2018-01-26



# 目录

## 1 背景介绍

## 2 基于深度学习的流量预测

### 2.1 深度学习基本概念

### 2.2 数据情况

### 2.3 流量预测模型STDenseNet

### 2.4 结果及分析

## 3 下一步工作

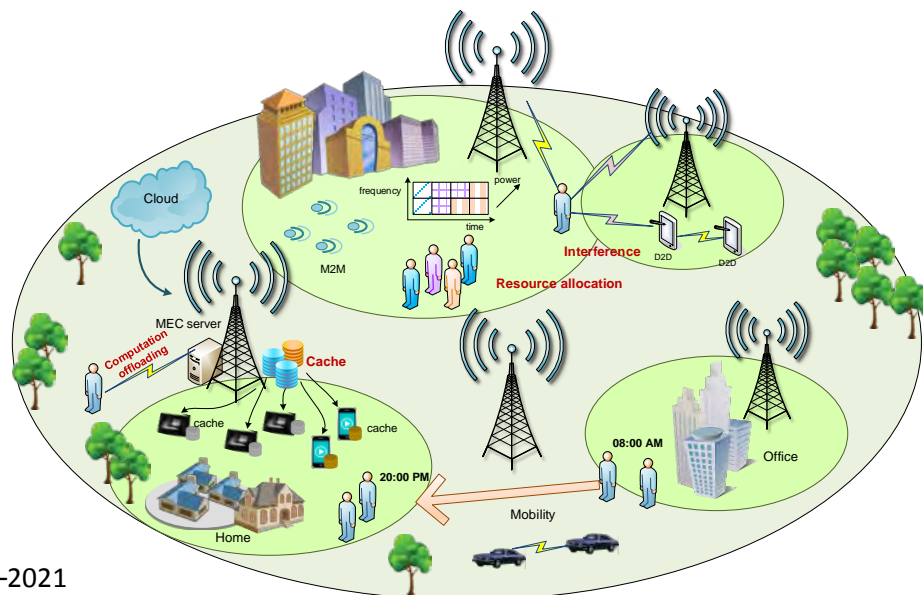


## 移动数据流量大幅提升

IP Traffic, 2016-2021							
	2016	2017	2018	2019	2020	2021	CAGR 2016-2021
<b>By Type (Petabytes [PB] per Month)</b>							
Fixed Internet	65,942	83,371	102,960	127,008	155,121	187,386	23%
Managed IP	22,911	27,140	31,304	35,226	38,908	42,452	13%
Mobile data	7,201	11,183	16,646	24,220	34,382	48,270	46%

## 未来网络架构

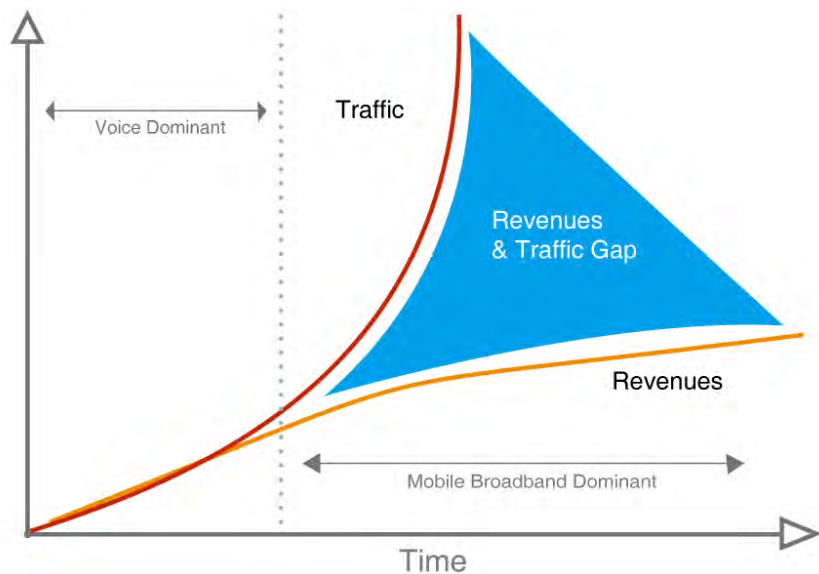
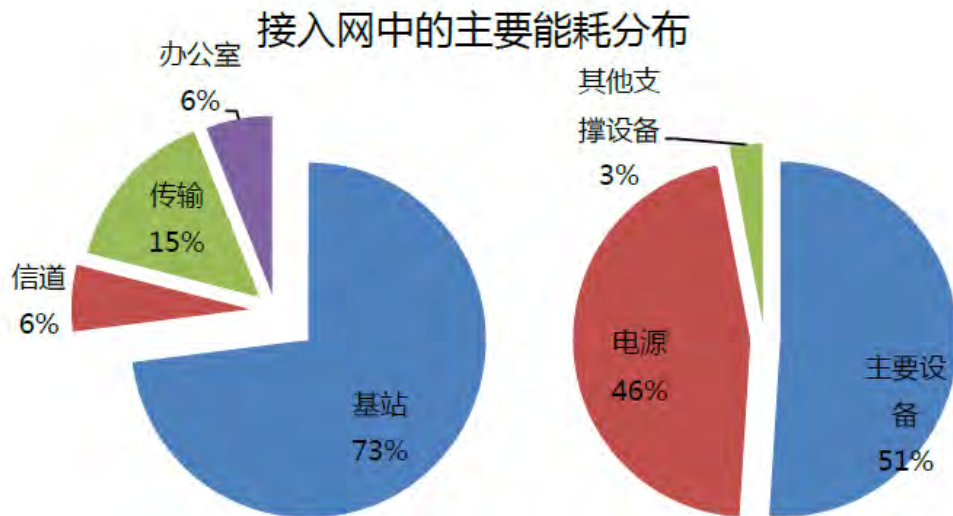
为满足用户的多样化需求，未来通信网络朝着超密集组网的方向发展





## □ 通信网络的能耗巨大

- 近500亿度电/年
- 蜂窝网无线接入部分（基站侧）占比60~80%
- 基站数量还在不断攀升

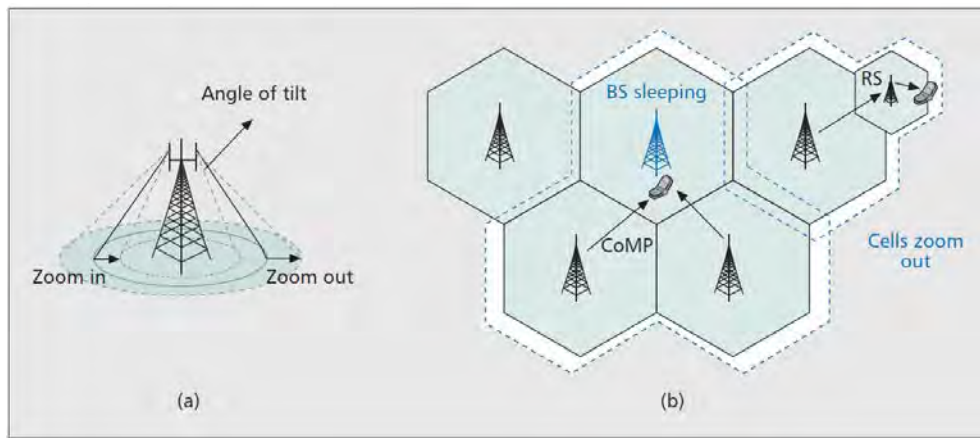


1. 通信业“十二五”节能减排情况  
2. 中国移动，《TD-LTE未来标准化需求》

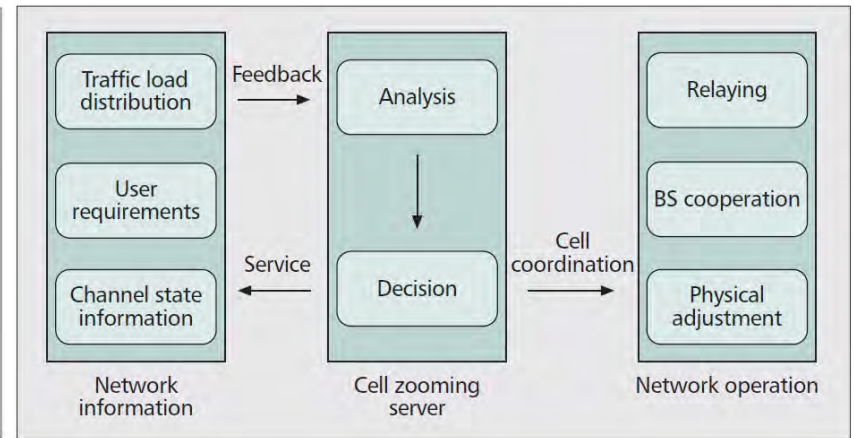


## □ 实现绿色、节能的途径：

- 使用可再生能源（风能、太阳能等）或者收集能量（Energy Harvesting）；
- 根据香农公式，蜂窝网可以从时间、空间、频率等多个角度提高能效，或者先进的信号处理技术（大规模天线等）；
- 网络柔性覆盖，基站为中心→用户为中心。将信令和业务承载分离、上行和下行分离。在流量低时关闭/休眠基站，对蜂窝网容量和覆盖进行基站缩放（Cell Zooming）实现能效优化。



Cell zooming图示



Cell zooming工作框架



□除了能效，未来通信网络还需要：

- 资源**弹性**分配

- ✓ 根据用户在时/空上的动态特性弹性的配置网络资源，通过网络协作，多个同构或异构小区形成簇，用户根据实际需求和网络状况选择合适的小区接入；

- 业务**适度**服务

- ✓ 特定业务优化→多业务/多服务质量需求。主动利用多业务的需求差异性和用户群体行为的趋同性动态的调整服务模式（实时/软实时/非实时，单播/多播/广播）。

**未来网络要实现这些特性，需要建立在对流量（ traffic ）的精准建模、分析以及预测之上！**

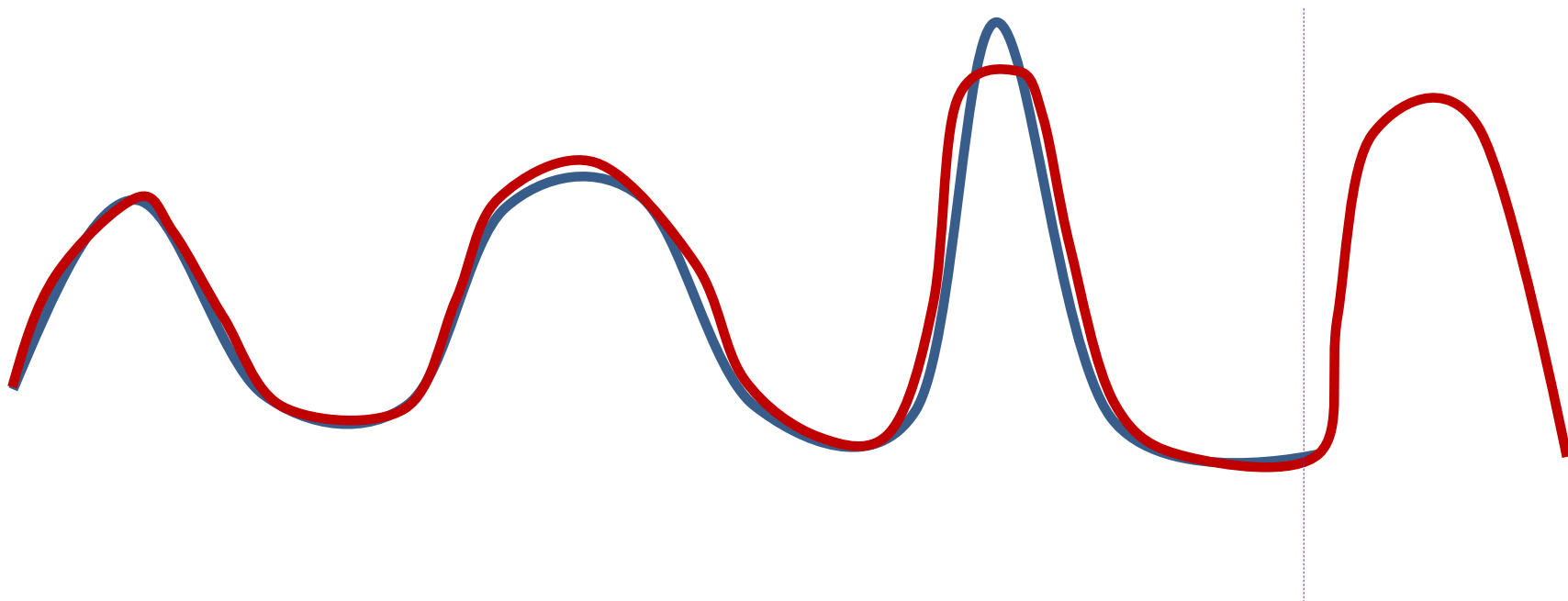


# 背景介绍



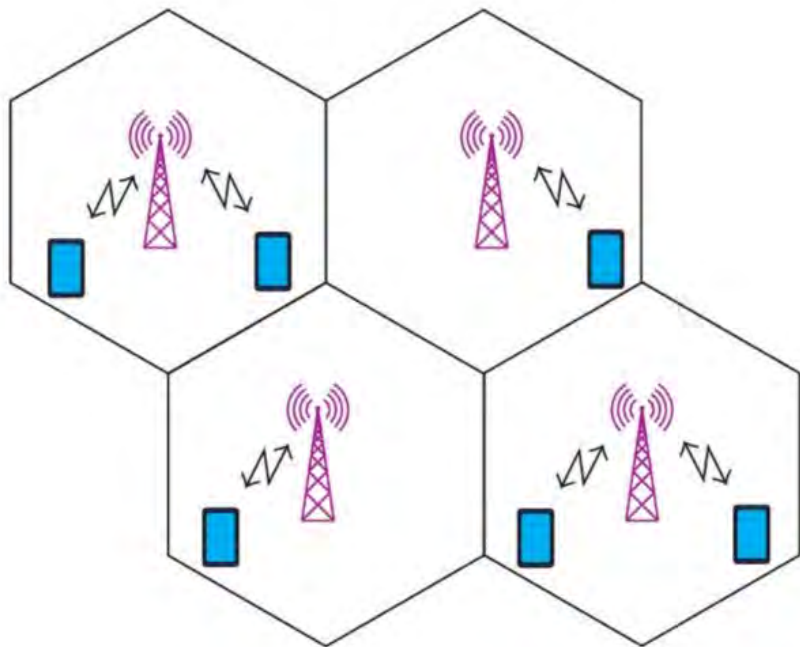
## □ 精准的流量预测/分析，可以：

- 改善网络管理：动态网络拥塞控制
- 降低运营成本：精准射频资源购置
- 增强网络能效：智能基站缩放/开关





## □ 流量预测分类



- 单输入-单输出：单个基站层面预测，仅利用该基站的历史信息（时间）；
- 多输入-单输出：基站以及周边基站的信息作为输入，预测当前基站的流量（时间+空间）；
- 多输入-单输出：用当前基站的信息，预测该基站以及周边基站的流量信息；
- 多输入-多输出：输入多个基站的流量信息，同时预测多个基站的流量信息。





# 目录

## 1 背景介绍

## 2 基于深度学习的流量预测

### 2.1 深度学习基本概念

### 2.2 数据情况

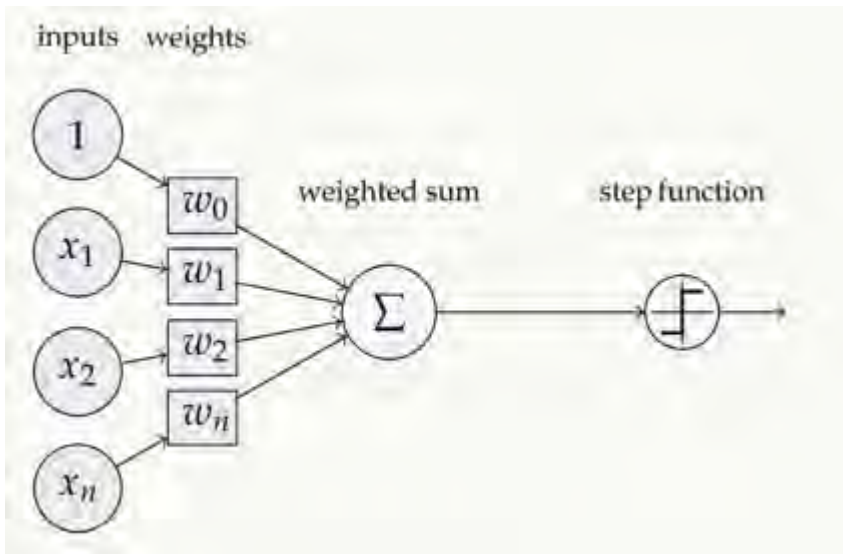
### 2.3 流量预测模型STDenseNet

### 2.4 结果及分析

## 3 下一步工作



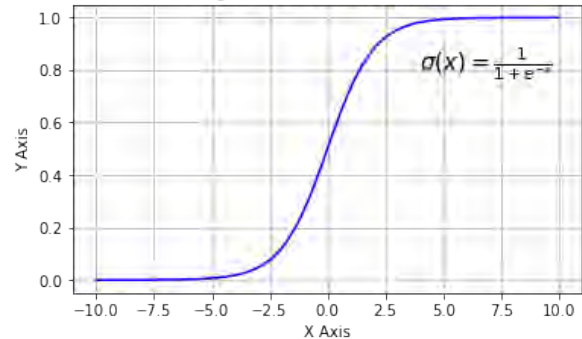
## □ 神经元示意图



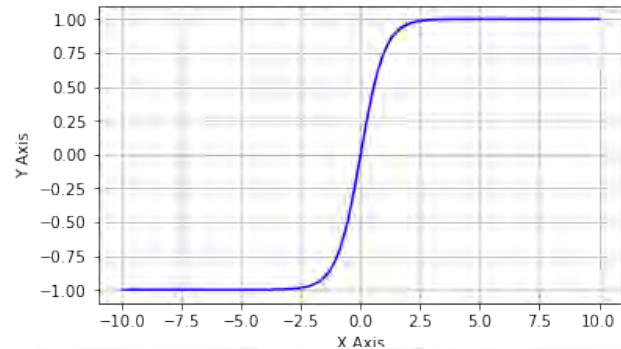
为什么用Rectified Linear Unit, ReLU

- 前两个在接近饱和区域，梯度消失，收敛慢
- 计算高昂

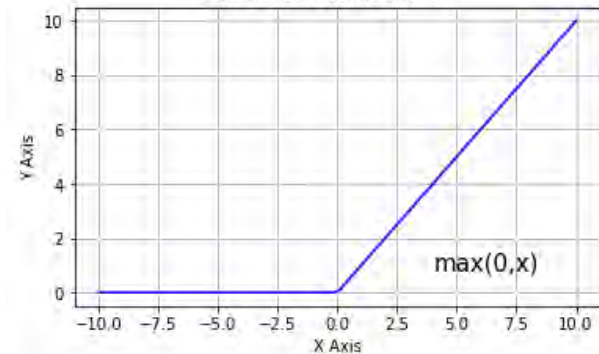
Sigmoid Activation Function



Tanh Activation Function

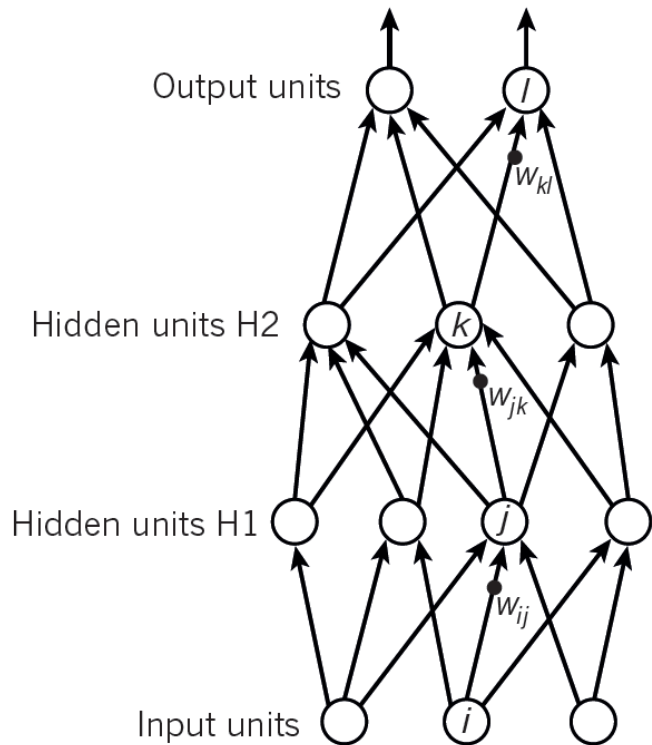


ReLU Activation Function





## □ Deep Neural Networks (Multilayer Perceptron) 深度神经网络，多层感知机



$$y_l = f(z_l)$$

$$z_l = \sum_{k \in H2} w_{kl} y_k$$

$$y_k = f(z_k)$$

$$z_k = \sum_{j \in H1} w_{jk} y_j$$

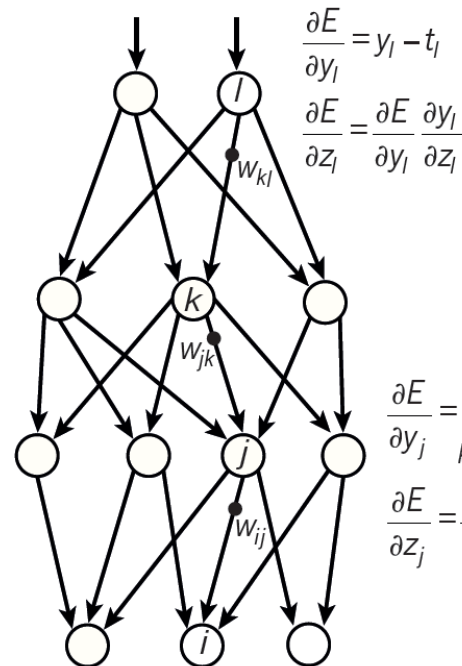
$$y_j = f(z_j)$$

$$z_j = \sum_{i \in \text{Input}} w_{ij} x_i$$

$$\frac{\partial E}{\partial y_k} = \sum_{l \in \text{out}} w_{kl} \frac{\partial E}{\partial z_l}$$

$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_k}$$

Compare outputs with correct answer to get error derivatives



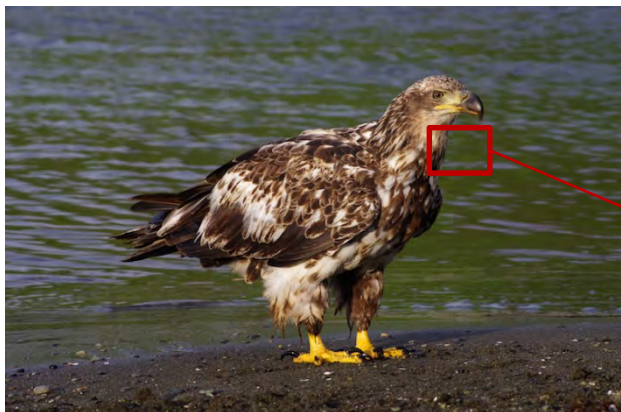
$$\frac{\partial E}{\partial y_j} = \sum_{k \in H2} w_{jk} \frac{\partial E}{\partial z_k}$$

$$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial z_j}$$



## Convolutional Neural Networks (卷积神经网络)

卷积操作 (convolution)  
 滤波器 (kernel)  
 步长 (stride)



1	0	-1
2	0	-2
1	0	-1

$$\begin{aligned}
 &93*1+92*0+91*-1 \\
 &+98*2+98*0+98*-2 \\
 &+102*1+104*0+105*-1 \\
 &=-1
 \end{aligned}$$

93	92	91	90	89	90	90	90	91	93
98	98	98	97	97	95	94	94	95	96
102	104	105	105	105	103	103	103	102	102
104	105	107	108	109	108	108	108	108	107
102	104	106	108	108	107	107	107	107	106
99	99	101	103	105	104	104	103	103	102
99	99	97	101	103	105	105	104	103	103
96	95	95	97	98	102	102	103	103	104
98	96	96	97	99	101	103	105	107	109
101	100	101	103	105	107	109	112	115	117

1	0	-1
2	0	-2
1	0	-1

Sobel算子，主要用于边缘检测

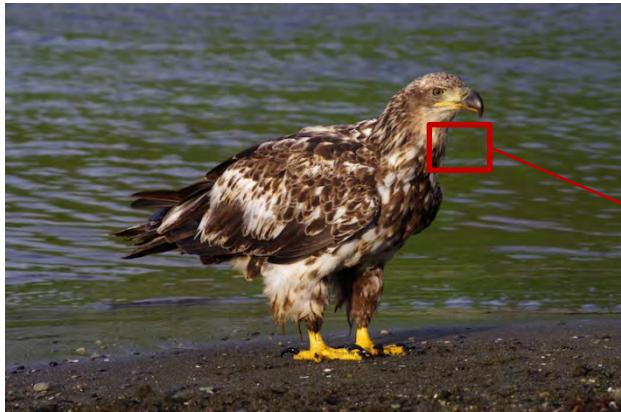


# 深度学习基本概念



## Convolutional Neural Networks (卷积神经网络)

卷积操作 (convolution)  
 滤波器大小 (kernel)  
 步长 (stride)



1	0	-1
2	0	-2
1	0	-1

$$\begin{aligned}
 &92*1+91*0+90*-1 \\
 &+98*2+98*0+97*-2 \\
 &+104*1+105*0+105*-1 \\
 &=3
 \end{aligned}$$

Stride=1

93	92	91	90	89	90	90	90	91	93
98	98	98	97	97	95	94	94	95	96
102	104	105	105	105	103	103	103	102	102
104	105	107	108	109	108	108	108	108	107
102	104	106	108	108	107	107	107	107	106
99	99	101	103	105	104	104	103	103	102
99	99	97	101	103	105	105	104	103	103
96	95	95	97	98	102	102	103	103	104
98	96	96	97	99	101	103	105	107	109
101	100	101	103	105	107	109	112	115	117

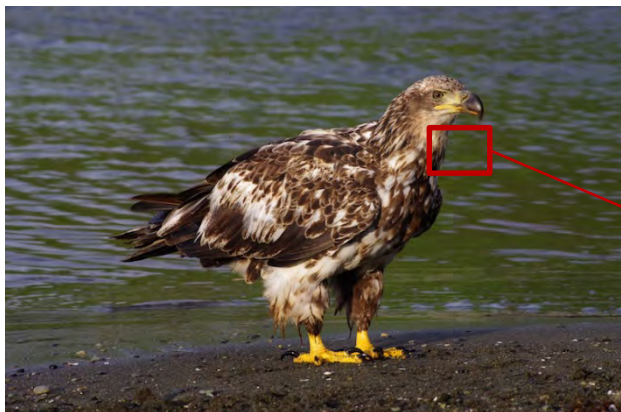
1	0	-1
2	0	-2
1	0	-1

Sobel算子，主要用于边缘检测



## Convolutional Neural Networks (卷积神经网络)

卷积操作 (convolution)  
 滤波器大小 (kernel)  
 步长 (stride)



1	0	-1
2	0	-2
1	0	-1

$$\begin{aligned}
 &93*1+92*0+91*-1 \\
 &+98*2+98*0+98*-2 \\
 &+102*1+104*0+105*-1 \\
 &=-1
 \end{aligned}$$

93	92	91	90	89	90	90	90	91	93
98	98	98	97	97	95	94	94	95	96
102	104	105	105	105	103	103	103	102	102
104	105	107	108	109	108	108	108	108	107
102	104	106	108	108	107	107	107	107	106
99	99	101	103	105	104	104	103	103	102
99	99	97	101	103	105	105	104	103	103
96	95	95	97	98	102	102	103	103	104
98	96	96	97	99	101	103	105	107	109
101	100	101	103	105	107	109	112	115	117

通过这个操作，我们得到了一种特征！



# 深度学习基本概念



更加形象化的展示卷积操作

Input Volume (+pad 1) (7x7x3)

$x[:, :, 0]$						
0	0	0	0	0	0	0
0	0	0	1	0	2	0
0	1	0	2	0	1	0
0	1	0	2	2	0	0
0	2	0	0	2	0	0
0	2	1	2	2	0	0
$x[:, :, 1]$						
0	0	0	0	0	0	0
0	2	1	2	1	1	0
0	2	1	2	0	1	0
0	0	2	1	0	1	0
0	1	2	2	2	2	0
0	0	1	2	0	1	0
$x[:, :, 2]$						
0	0	0	0	0	0	0
0	2	1	1	2	0	0
0	1	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	2	1	0	0
0	2	2	1	1	1	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

$w0[:, :, 0]$		
-1	0	1
0	0	1
1	-1	1
$w0[:, :, 1]$		
-1	0	1
1	-1	1
0	1	0
$w0[:, :, 2]$		
-1	1	1
1	1	0
0	-1	0
Bias $b0$ (1x1x1)		
$b0[:, :, 0]$		
1		

Filter W1 (3x3x3)

$w1[:, :, 0]$		
0	1	-1
0	-1	0
0	-1	1
$w1[:, :, 1]$		
-1	0	0
1	-1	0
1	-1	0
$w1[:, :, 2]$		
-1	1	-1
0	-1	-1
1	0	0
Bias $b1$ (1x1x1)		
$b1[:, :, 0]$		
0		

Output Volume (3x3x2)

$o[:, :, 0]$		
2	3	3
3	7	3
8	10	-3
$o[:, :, 1]$		
-8	-8	-3
-3	1	0
-3	-8	-5

Pooling操作

102	104	106	108	108	107
99	99	101	103	105	104
99	99	97	101	103	105
96	95	95	97	98	102
98	96	96	97	99	101
101	100	101	103	105	107

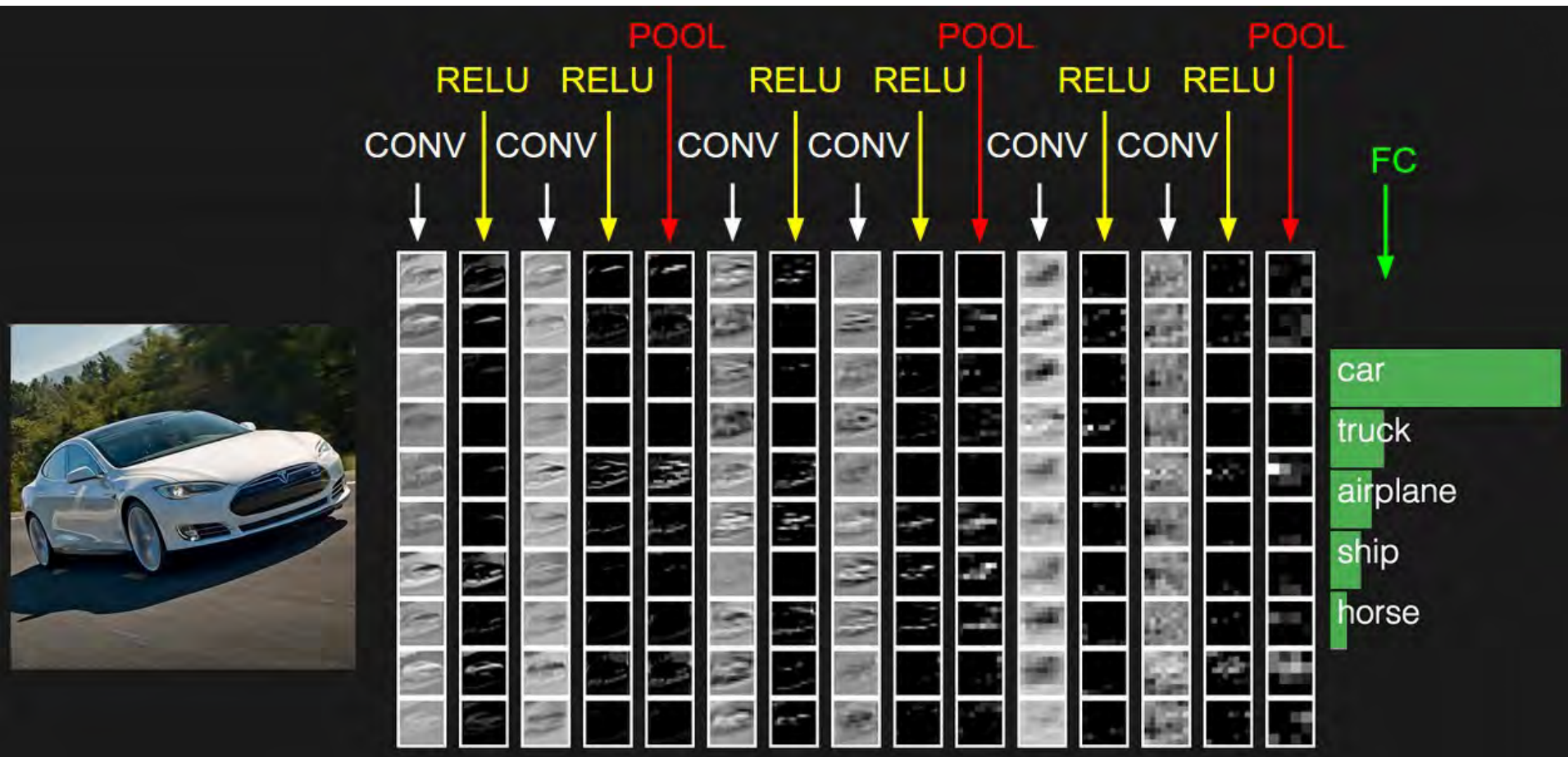
106	108
101	107



# 深度学习基本概念



## 深度学习网络架构示意



逐层学习，提取更高层次的抽象特征





# 深度学习基本概念



## □ 深度学习，从几行代码开始...

```
from keras.models import Sequential
from keras.layers import Conv2D, Dense, Activation, MaxPool2D, Flatten

model = Sequential()
first_layer = Conv2D(filters=32, kernel_size=3, strides=1, input_shape=(3, 32, 32))
model.add(first_layer)
for i in range(3):
    model.add(Conv2D(filters=32, kernel_size=3, strides=1))
    model.add(Activation('relu'))
    model.add(Conv2D(filters=32, kernel_size=3, strides=1))
    model.add(Activation('relu'))
    model.add(MaxPool2D(pool_size=2, strides=1))
model.add(Flatten())
model.add(Dense(10))
model.add(Activation('sigmoid'))
```



## □ 网络架构及输出大小

- 7个卷积层，1个全连接层
- 3个pooling层

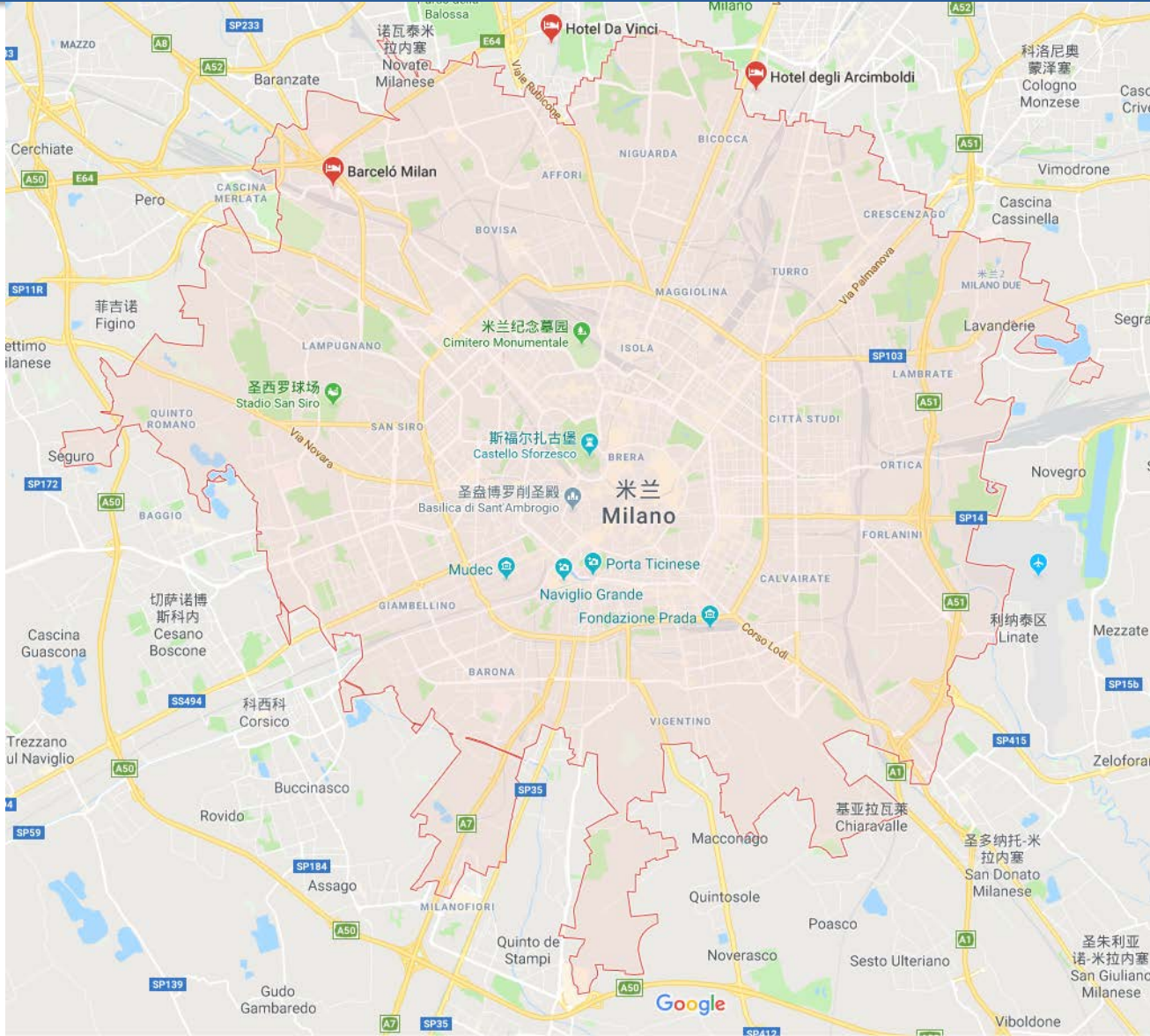
## □ 参数计算

- 输入channels\*32\*32
- Kernel size (3\*3)
- 输出32个特征
- 参数大小:  $3*3*channels*features$
- Channels=3, 参数=864
- 32个bias
- 共: 896

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 30, 30)	896
conv2d_2 (Conv2D)	(None, 32, 28, 28)	9248
activation_1 (Activation)	(None, 32, 28, 28)	0
conv2d_3 (Conv2D)	(None, 32, 26, 26)	9248
activation_2 (Activation)	(None, 32, 26, 26)	0
max_pooling2d_1 (MaxPooling2D)	(None, 32, 25, 25)	0
conv2d_4 (Conv2D)	(None, 32, 23, 23)	9248
activation_3 (Activation)	(None, 32, 23, 23)	0
conv2d_5 (Conv2D)	(None, 32, 21, 21)	9248
activation_4 (Activation)	(None, 32, 21, 21)	0
max_pooling2d_2 (MaxPooling2D)	(None, 32, 20, 20)	0
conv2d_6 (Conv2D)	(None, 32, 18, 18)	9248
activation_5 (Activation)	(None, 32, 18, 18)	0
conv2d_7 (Conv2D)	(None, 32, 16, 16)	9248
activation_6 (Activation)	(None, 32, 16, 16)	0
max_pooling2d_3 (MaxPooling2D)	(None, 32, 15, 15)	0
flatten_1 (Flatten)	(None, 7200)	0
dense_1 (Dense)	(None, 10)	72010
activation_7 (Activation)	(None, 10)	0



# 数据介绍—Open Big Data Challenges

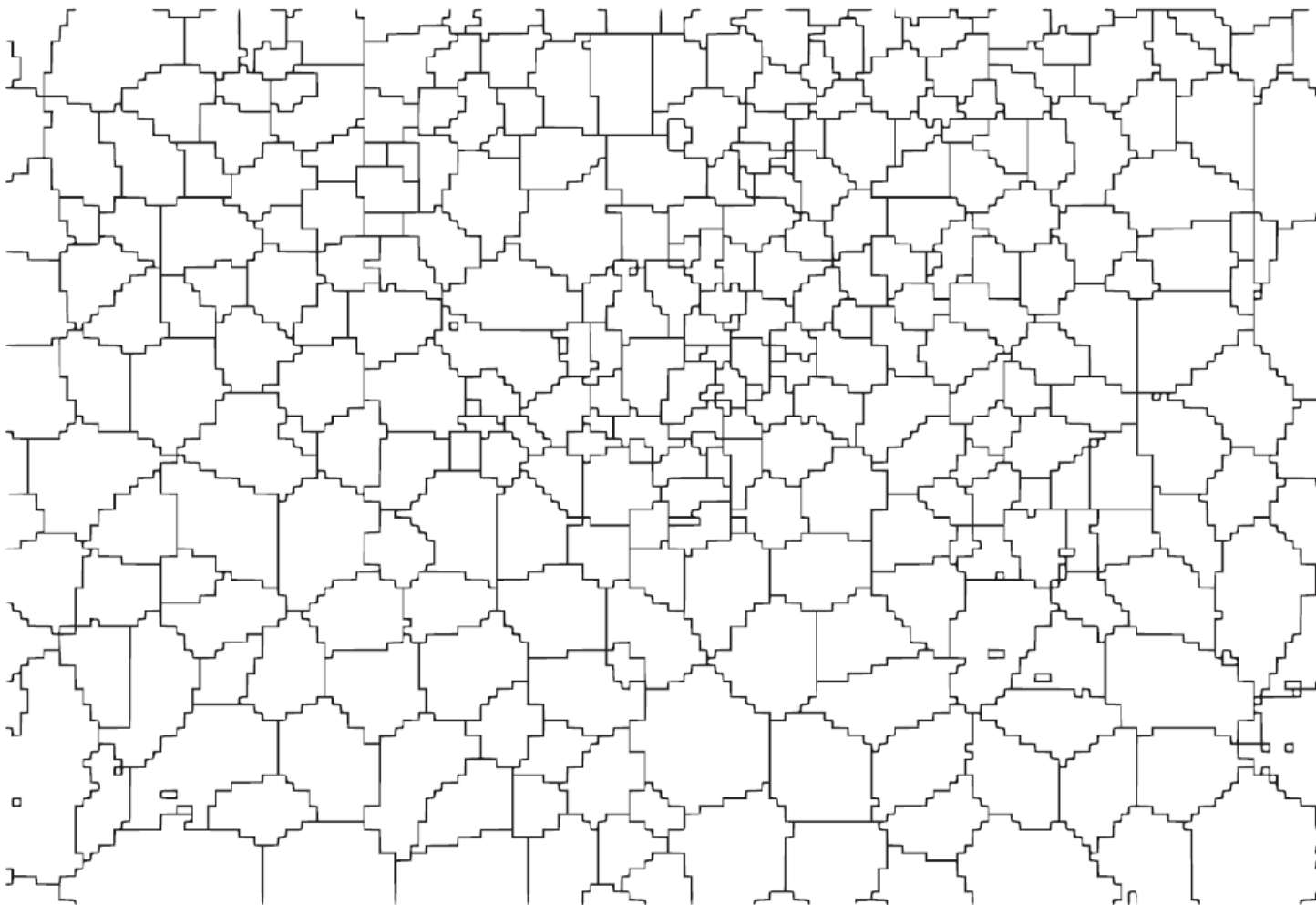


意大利米兰  
CDR数据

100多万人口



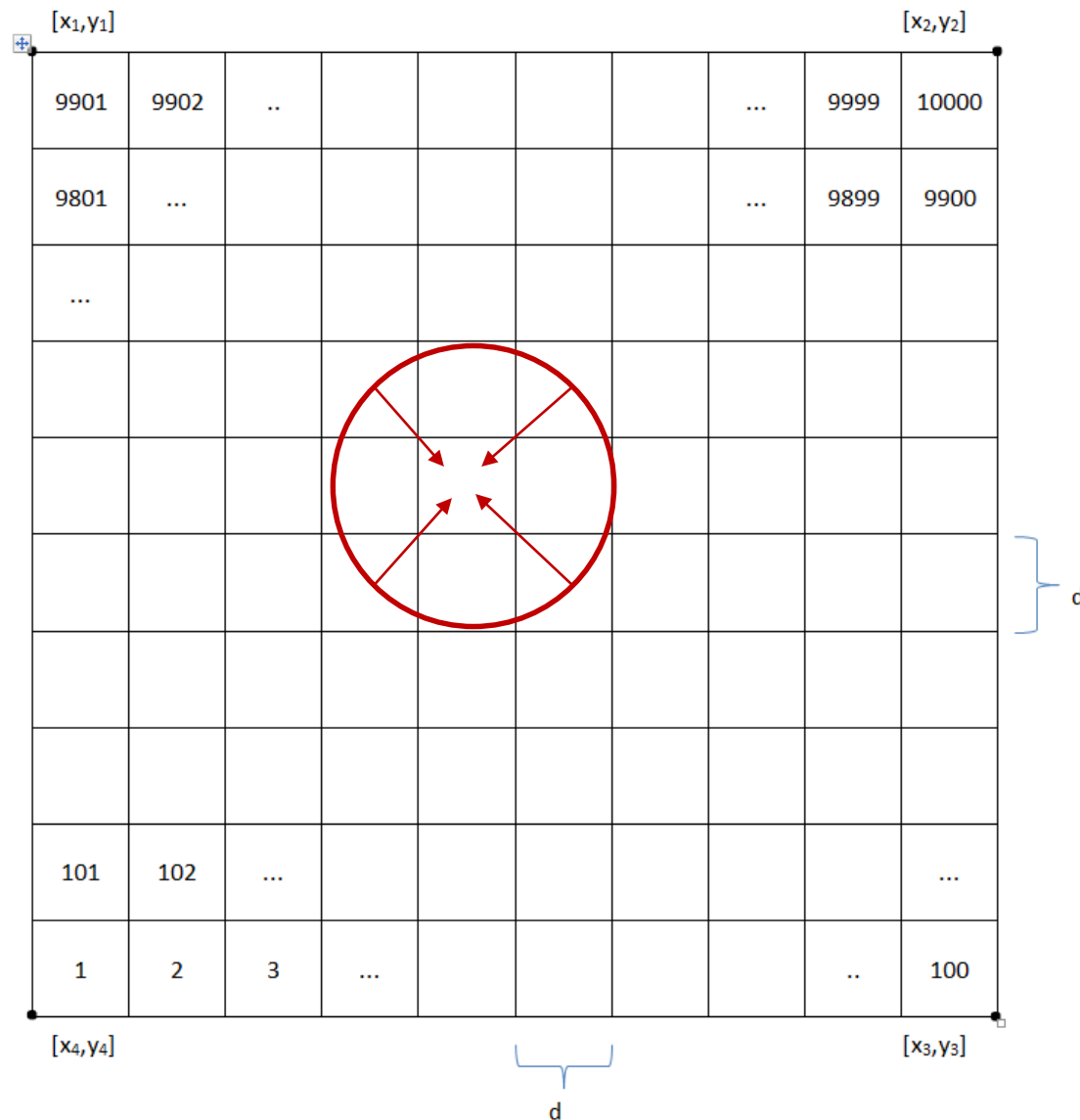
# 数据介绍



基站覆盖区域图



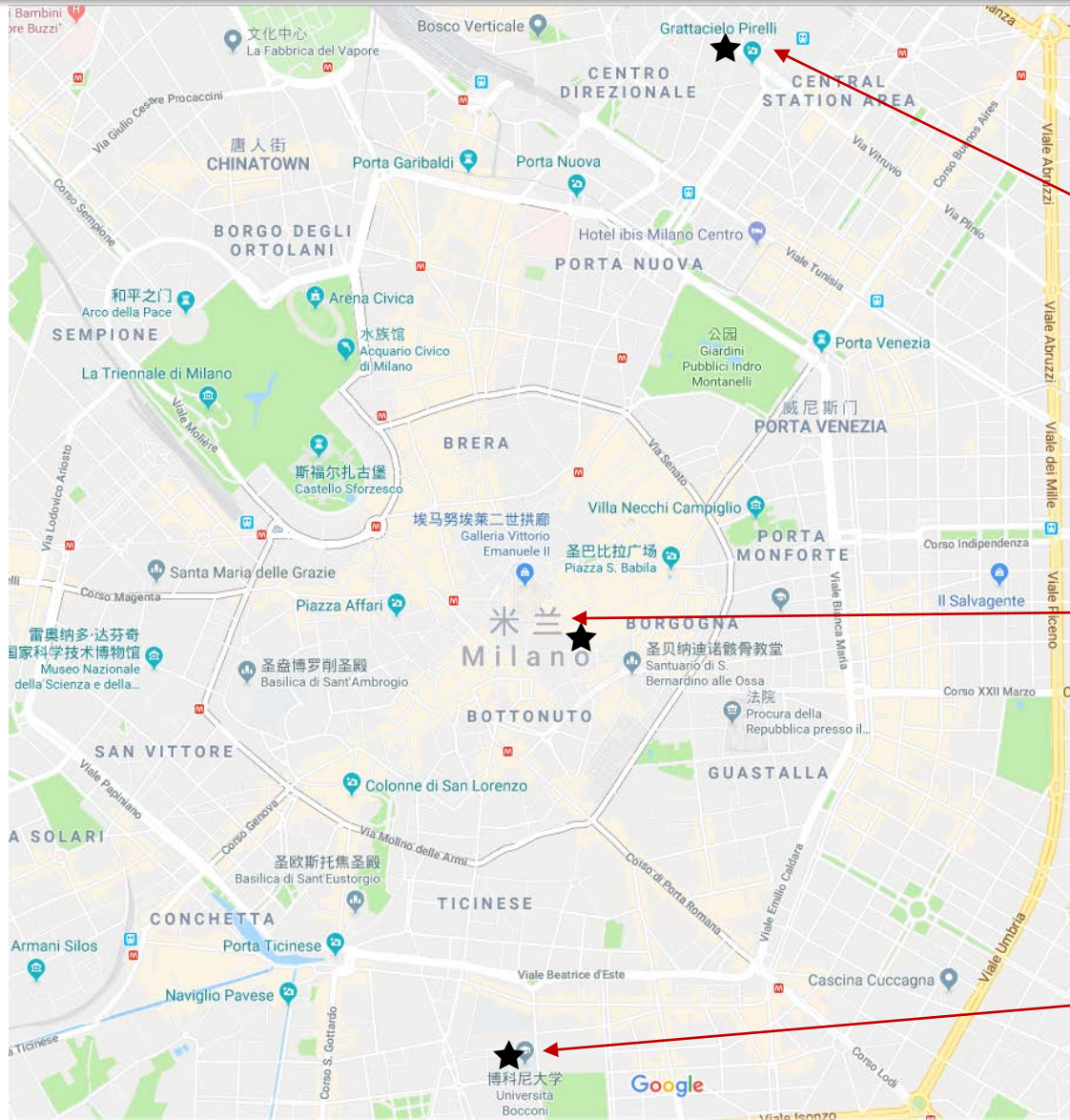
# 数据介绍—网格划分/预处理



- 整个城市被分成了100\*100的网格
- 每个网格大小235m\*235m
- 时间范围：2013-11-01~2014-01-01
- 每个网格所记录的数据如下：
  - 呼入/呼出电话数量
  - 接收/发出的短信数量
  - 上网的流量
- SMS和CALL都是按10分钟的尺度记录，即：10分钟内的和，记录一次；然后再过10分钟，记录一次。Internet流量是按照15分钟记录一次或者产生的流量达到了5MB。



# 三个Cell ID展示



纳维利：米兰著名夜生活区域（酒吧、娱乐场所）



米兰大教堂：城市中心



博科尼大学：米兰著名的一所大学

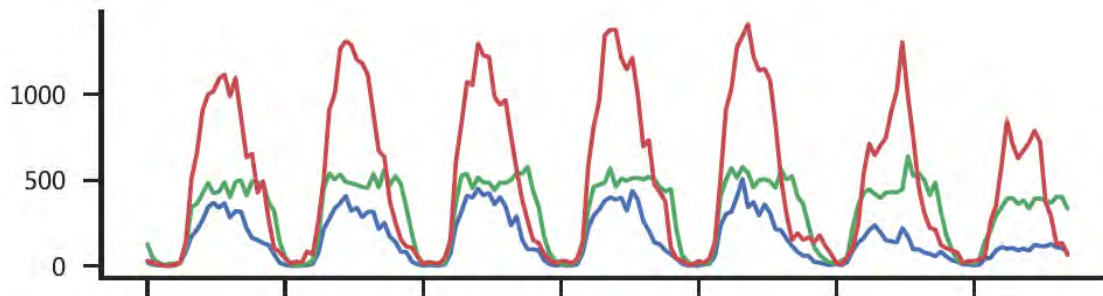




# 数据介绍—时间空间分布

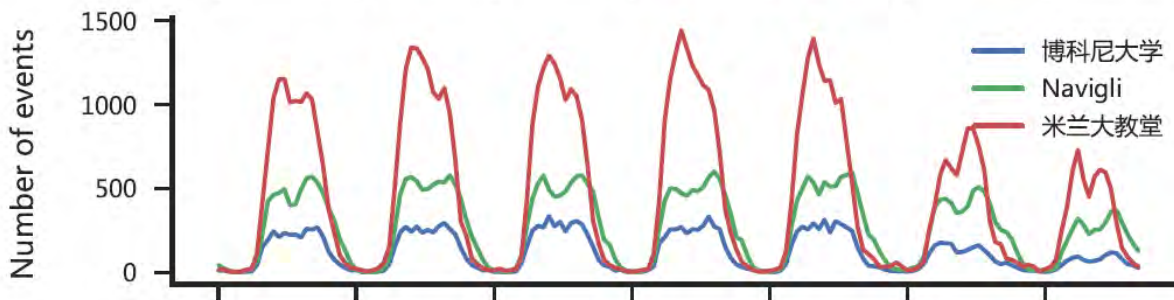


短信



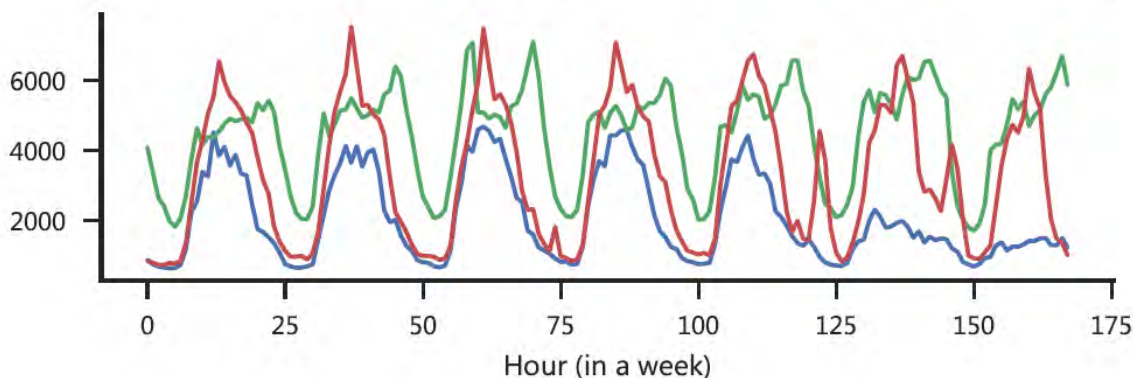
城市中心，三种业务量都较高；

通话



繁华地区，周末SMS和CALL业务量下降，但上网业务量不变；

上网

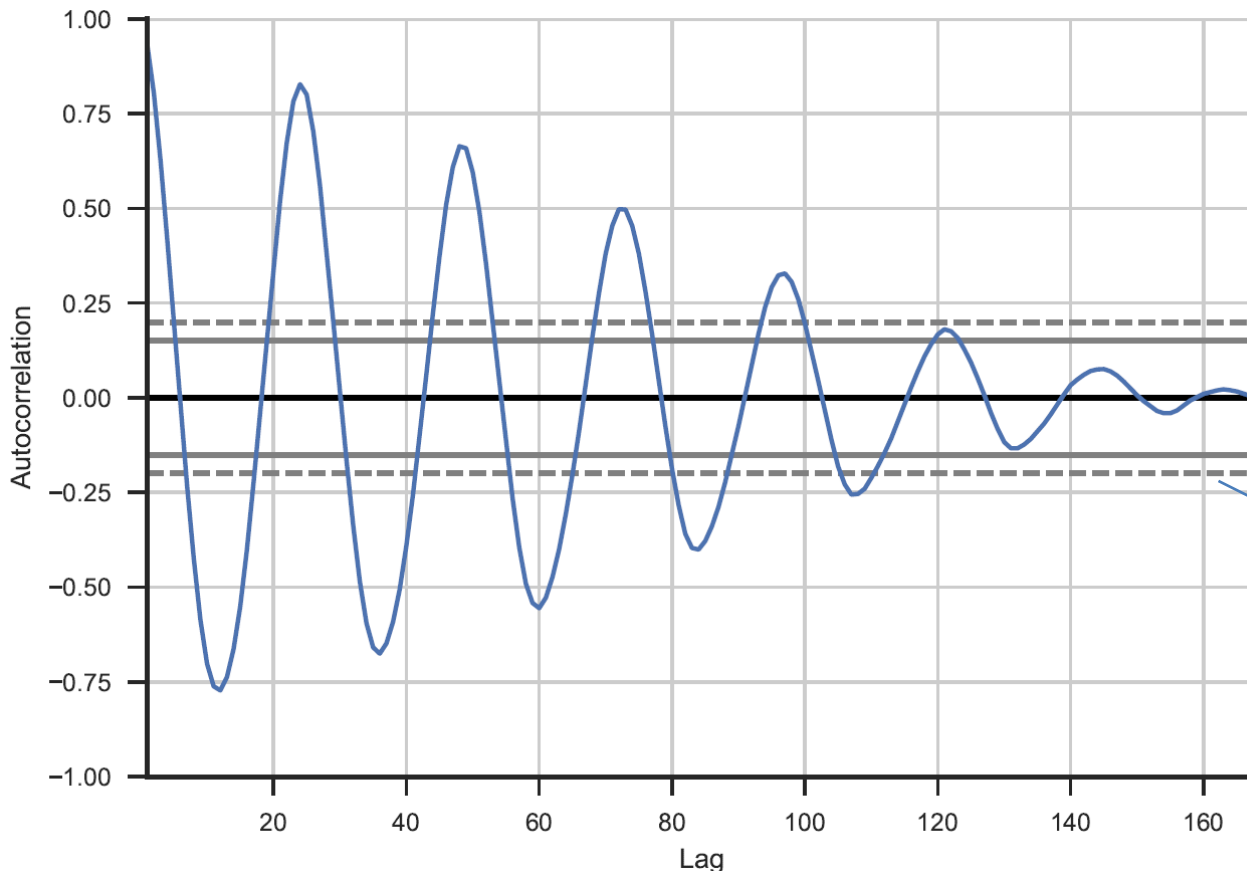


纳维利地区的业务量随着夜晚的到来而增加，高峰段有明显的“时延”；

上网业务量最难预测！



## 时间尺度上相关性



自相关(Autocorrelation), 也叫序列相关, 是一个信号于其自身在不同时间点的互相关

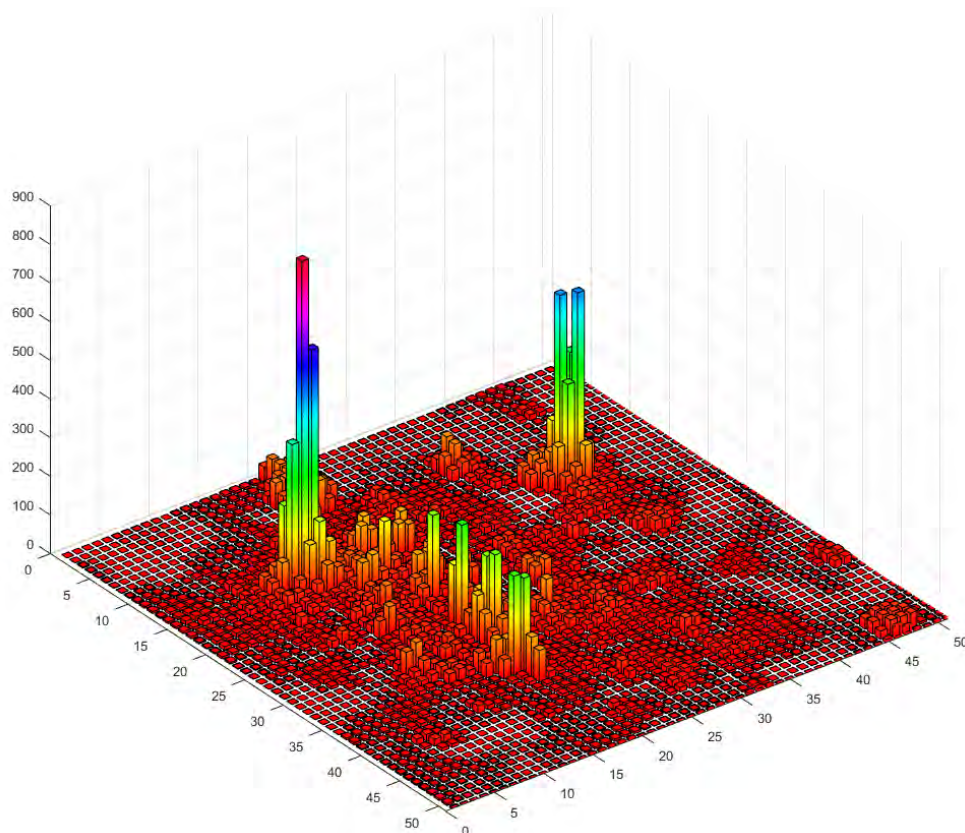
99%置信区间。如果信号是随机产生、没有规律的, 那么在任意lag上的自相关值都趋于0。

$$R(k) = \frac{E[(X_i - \mu_i)(X_{i+k} - \mu_{i+k})]}{\sigma^2}$$





## 空间尺度上的分布特点及相关性



1	0.62	0.75	0.7	0.56	0.46	0.46	0.69	0.75	0.83
2	0.53	0.65	0.75	0.71	0.77	0.67	0.63	0.8	0.78
3	0.61	0.62	0.72	0.71	0.82	0.73	0.63	0.7	0.72
4	0.69	0.67	0.71	0.71	0.86	0.87	0.81	0.8	0.75
5	0.67	0.7	0.74	0.94	1	0.85	0.84	0.79	0.79
6	0.64	0.78	0.75	0.95	0.96	0.82	0.84	0.75	0.82
7	0.66	0.8	0.76	0.82	0.89	0.87	0.83	0.79	0.79
8	0.58	0.71	0.73	0.74	0.84	0.84	0.81	0.6	0.81
9	0.61	0.65	0.73	0.77	0.85	0.83	0.81	0.81	0.8

空间分布不均匀，但小区间具有相关性，有可能离得远，但仍具有高相关性。

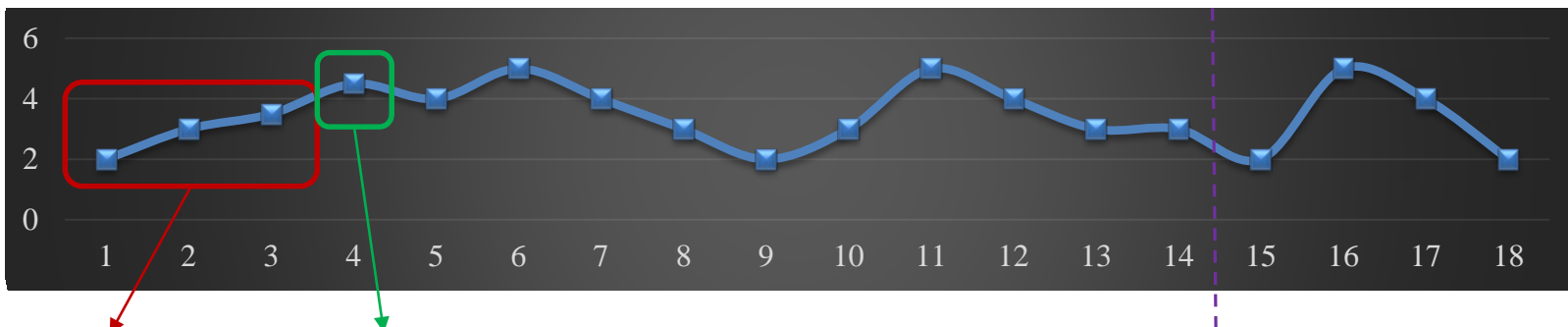
$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X\sigma_Y} = \frac{E((X - \mu_X)(Y - \mu_Y))}{\sigma_X\sigma_Y}$$



# 样本构建



- 根据前面的分析可以得出，蜂窝网的流量预测必须要同时考虑时间相关和空间相关。那么，如何去做？
- 先考虑时间相关性：用前N个时间点，预测下一个时间点



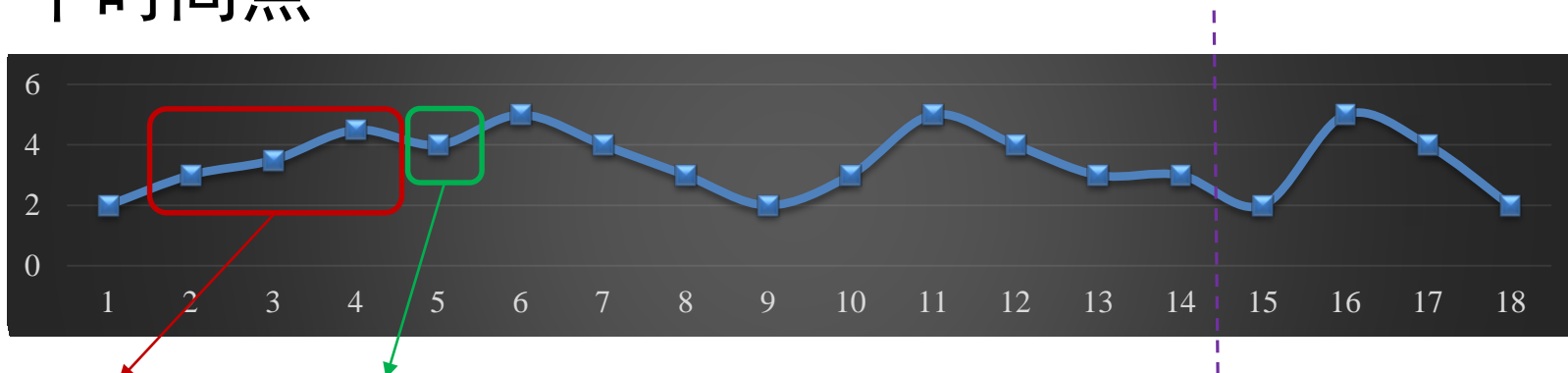
特征 :X

标记 :y

$$x_{11}, x_{21}, x_{31} \rightarrow y_1$$



- 根据前面的分析可以得出，蜂窝网的流量预测必须要同时考虑时间相关和空间相关。那么，如何去做？
- 先考虑时间相关性：用前N个时间点，预测下一个时间点



特征 :X

标记 :y

$$x_{11}, x_{21}, x_{31} \rightarrow y_1$$

$$x_{21}, x_{22}, x_{23} \rightarrow y_2$$



# 样本构建



- 根据前面的分析可以得出，蜂窝网的流量预测必须要同时考虑时间相关和空间相关。那么，如何去做？
- 先考虑时间相关性：用前N个时间点，预测下一个时间点



特征 :X      标记 :y

$X_{11}, X_{21}, X_{31} \rightarrow y_1$   
 $X_{21}, X_{22}, X_{23} \rightarrow y_2$   
 $X_{n1}, X_{n2}, X_{n3} \rightarrow y_n$

训练样本，用来训练一个模型

$X_{11}, X_{21}, X_{31} \rightarrow y_1$   
 $X_{m1}, X_{m2}, X_{m3} \rightarrow y_1$

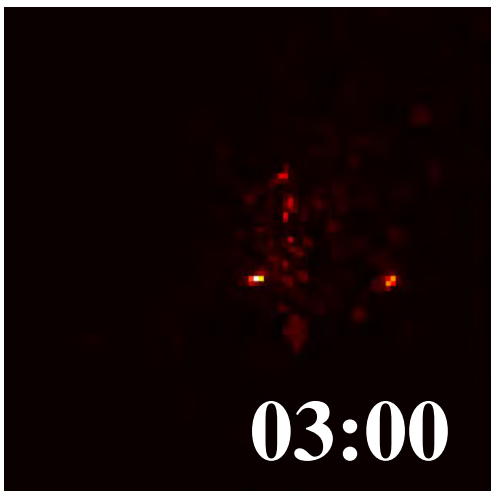
测试样本，用来测试模型的好坏



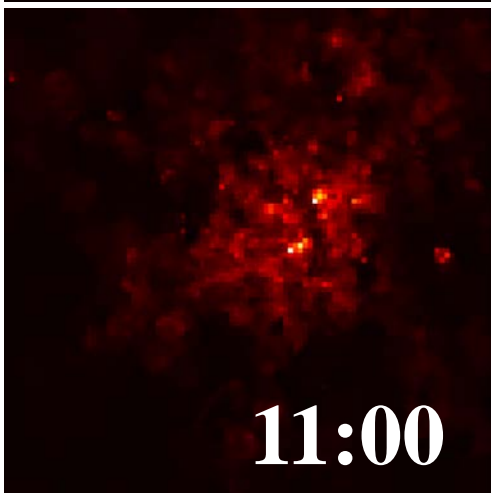
# 空间相关性



如何加入空间相关性？把城市尺度的流量当成图像，再看卷积操作...



03:00



11:00

1	0	-1
2	0	-2
1	0	-1

$$\begin{aligned}
 &92*1+91*0+90*-1 \\
 &+98*2+98*0+97*-2 \\
 &+104*1+105*0+105*-1 \\
 &=3
 \end{aligned}$$

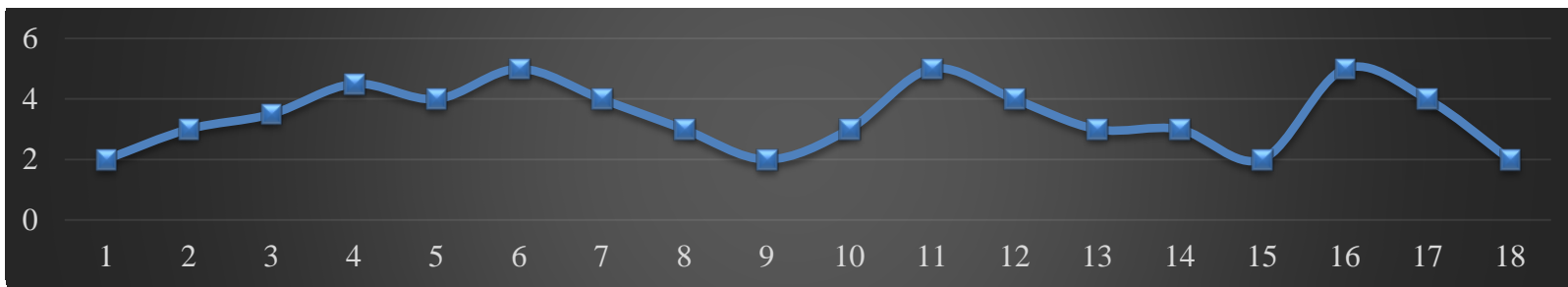
93	92	91	90	89	90	90	90	91	93
98	98	98	97	97	95	94	94	95	96
102	104	105	105	105	103	103	103	102	102
104	105	107	108	109	108	108	108	108	107
102	104	106	108	108	107	107	107	107	106
99	99	101	103	105	104	104	103	103	102
99	99	97	101	103	105	105	104	103	103
96	95	95	97	98	102	102	103	103	104
98	96	96	97	99	101	103	105	107	109
101	100	101	103	105	107	109	112	115	117



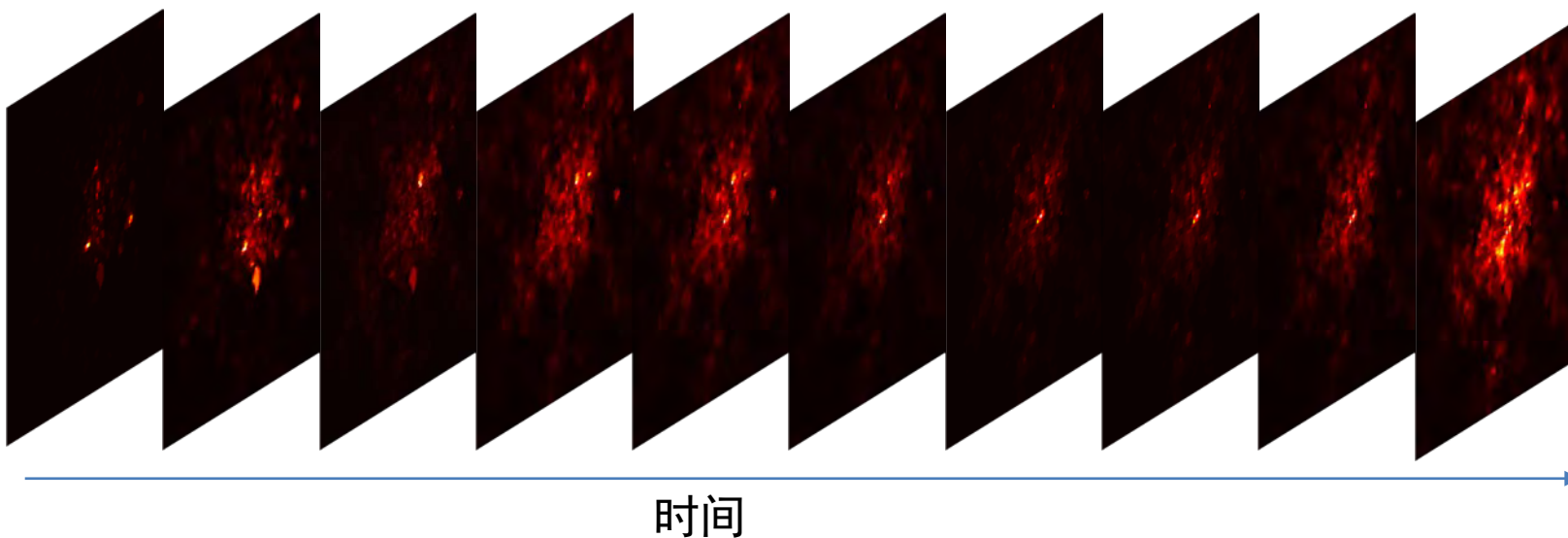
# 单个基站与所有基站类比



## □ 单个基站业务量变化



## □ 所有基站（10000个）业务量

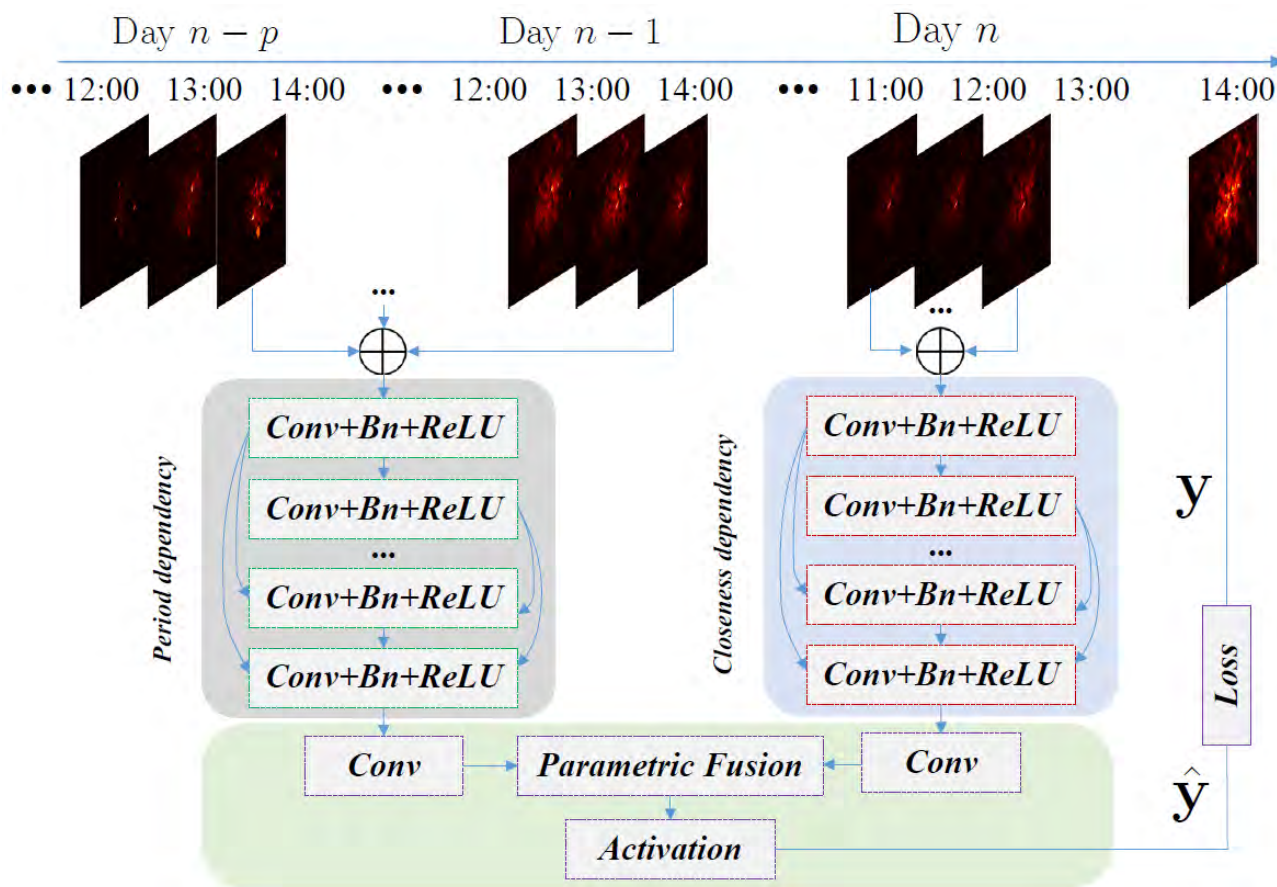




# 预测模型



除了短期时间相关性和空间相关性，还考虑周期性 →  
STDenseNet (Spatial-Temporal Densely Connected Convolutional Neural networks)



层数的大小  
可以控制空  
间相关性的  
尺度



## □ 参数选择

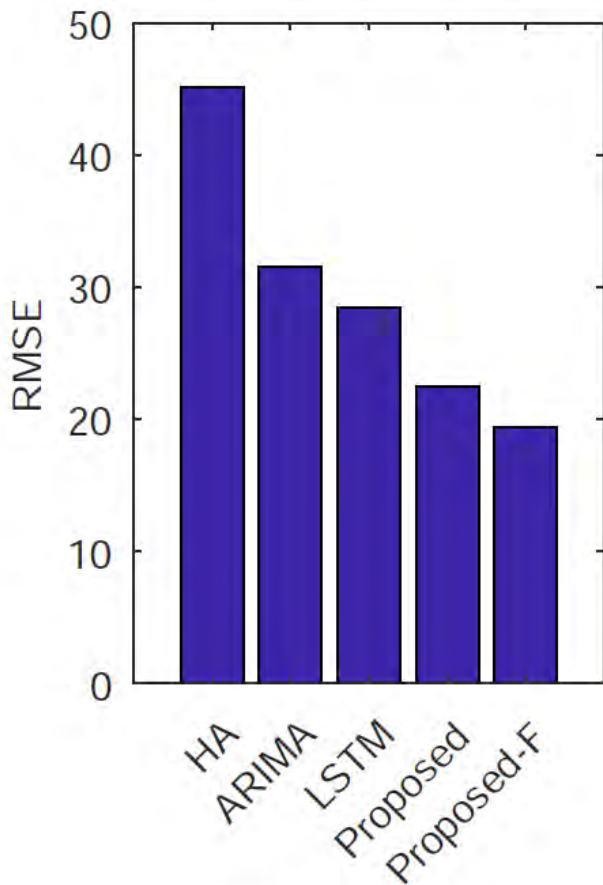
评价标准: 
$$\text{RMSE} = \sqrt{\frac{1}{z} \sum_i (x_i - \hat{x}_i)^2}$$

参数	取值	备注
Epochs	100	How many times we trained on the data
Learning rate	0.005	Learning rate of optimizer
Dropout	0.2	Drop a specific percent of connections between two layers
Filters	32	Number of feature maps
Closeness size	3	Short term dependency
Period size	3	Middle term dependency

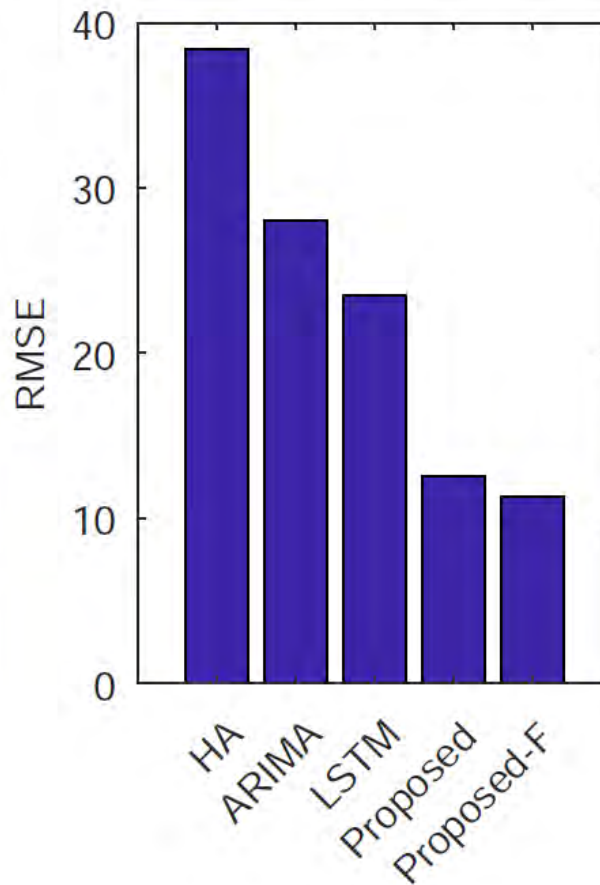




## □ 实验结果



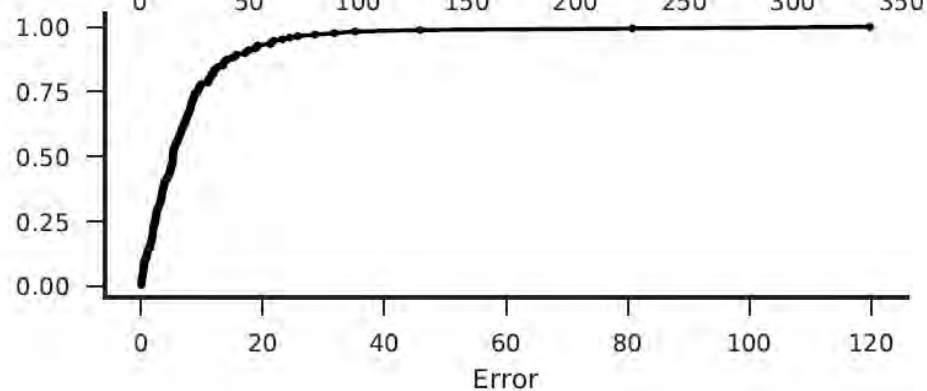
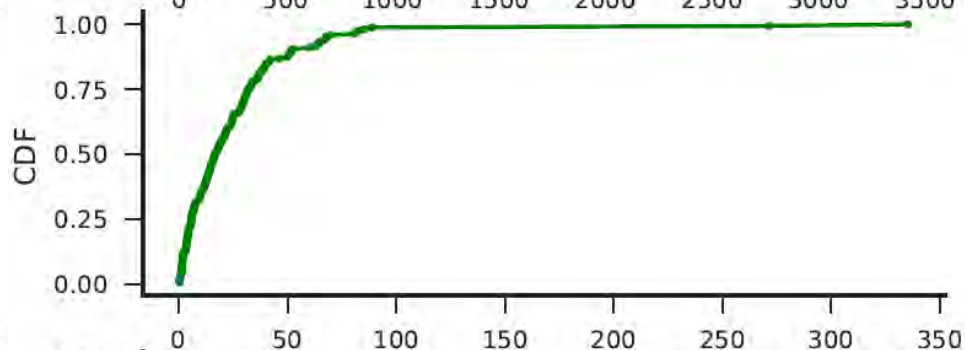
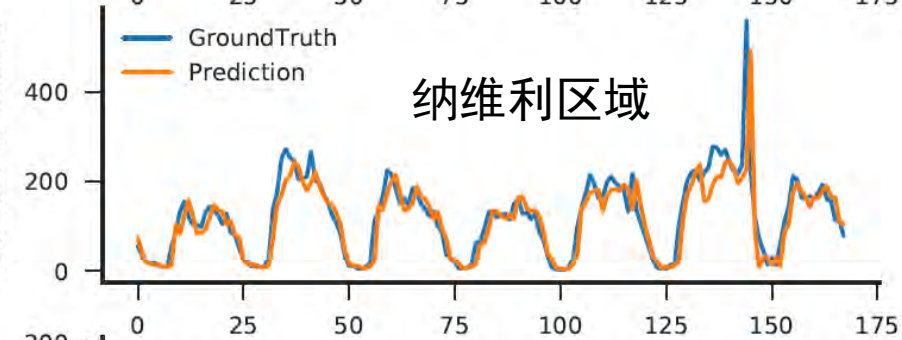
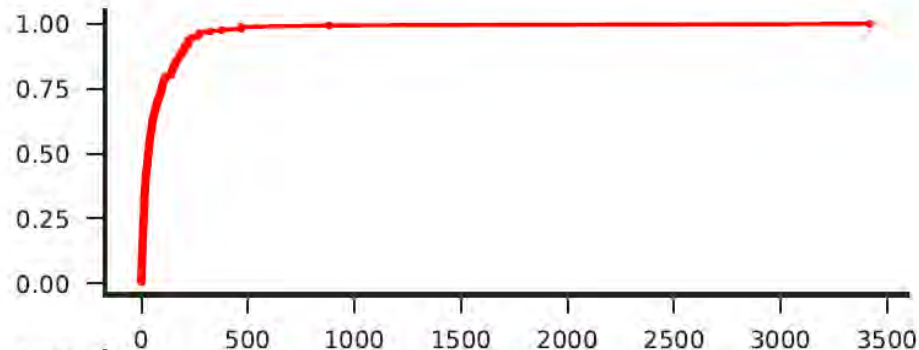
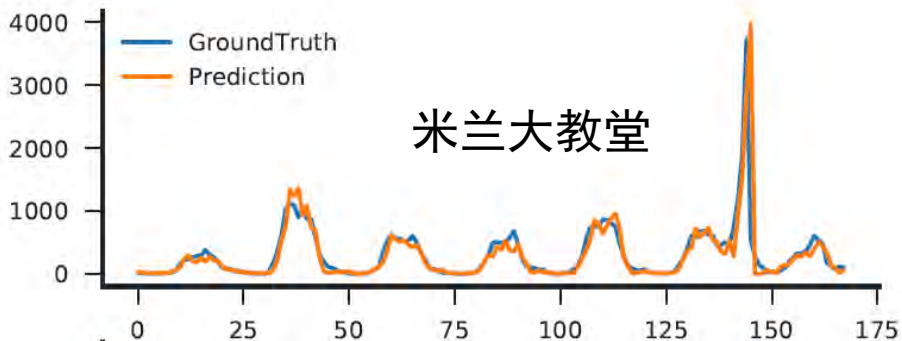
(a) SMS dataset



(b) CALL dataset

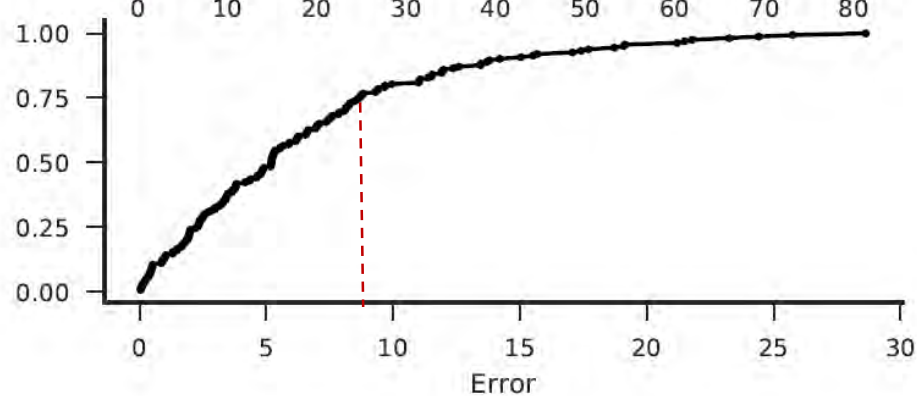
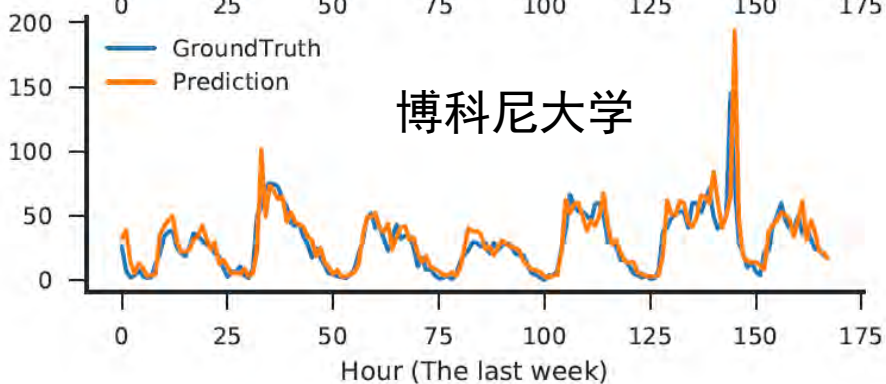
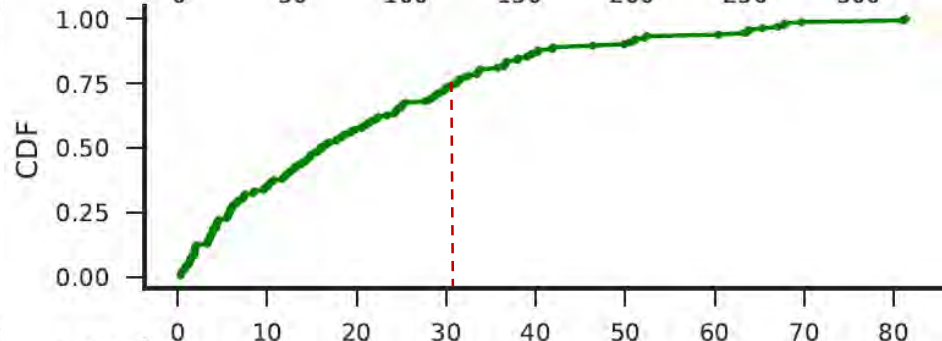
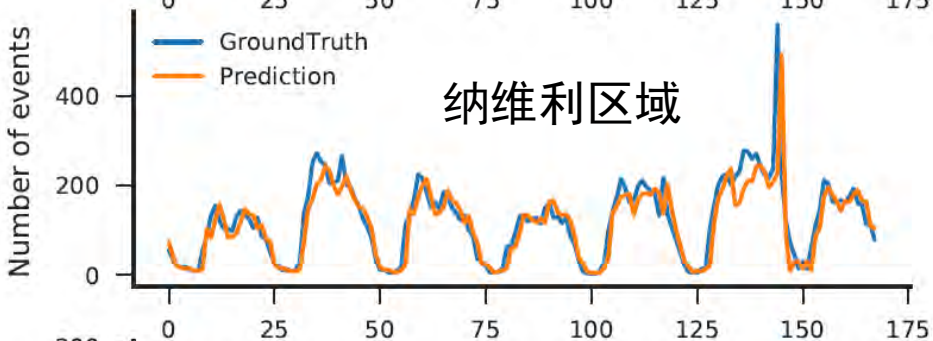
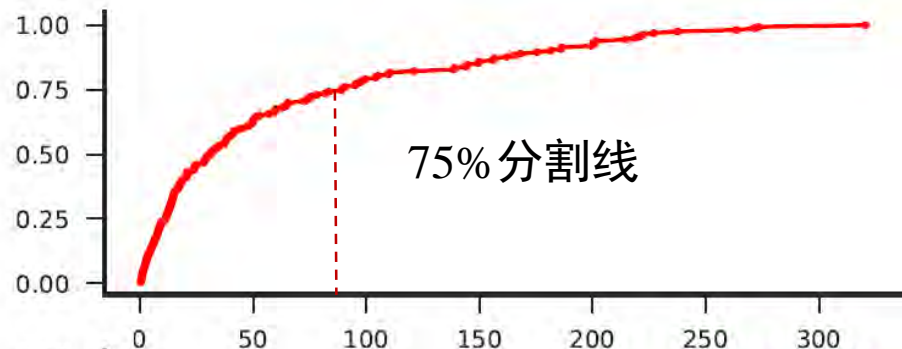
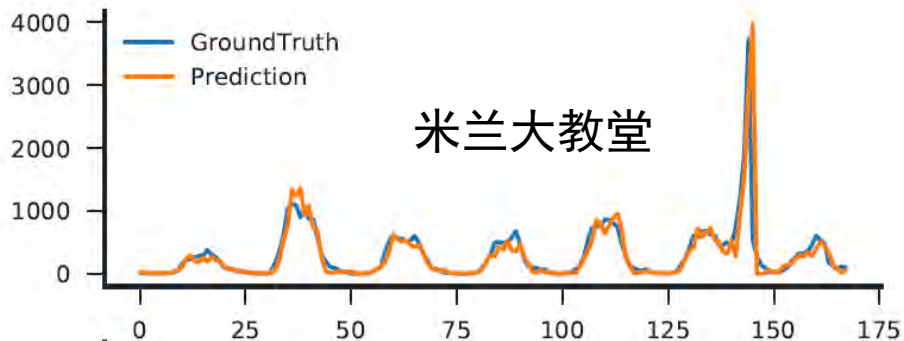


# 不同区域的预测结果





# 不同区域的预测结果





# 目录

## 1 背景介绍

## 2 基于深度学习的流量预测

### 2.1 深度学习基本概念

### 2.2 数据情况

### 2.3 流量预测模型STDenseNet

### 2.4 结果及分析

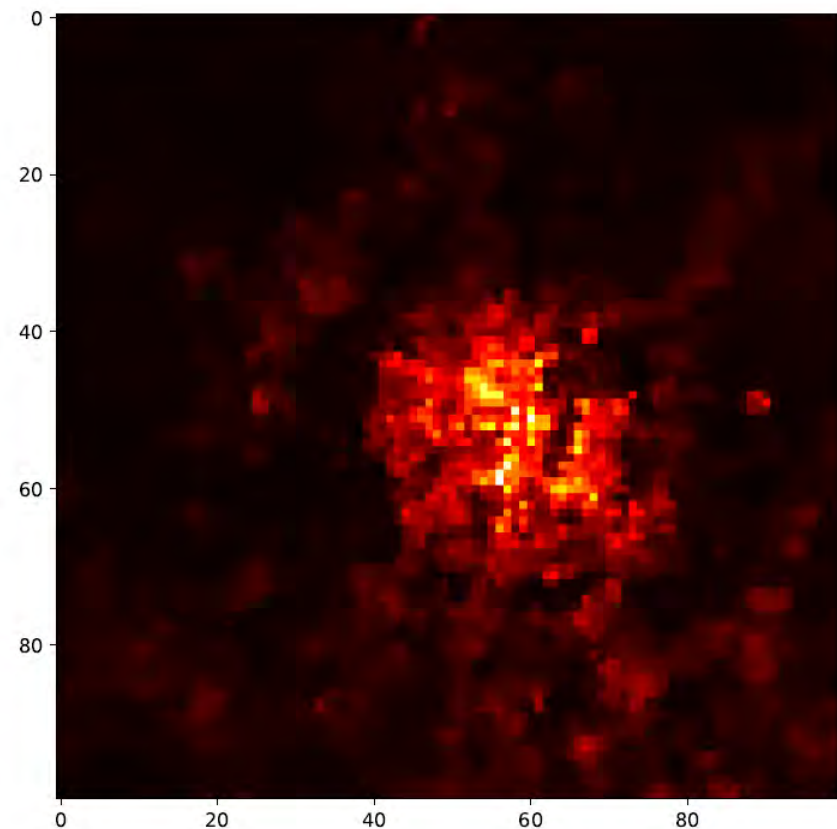
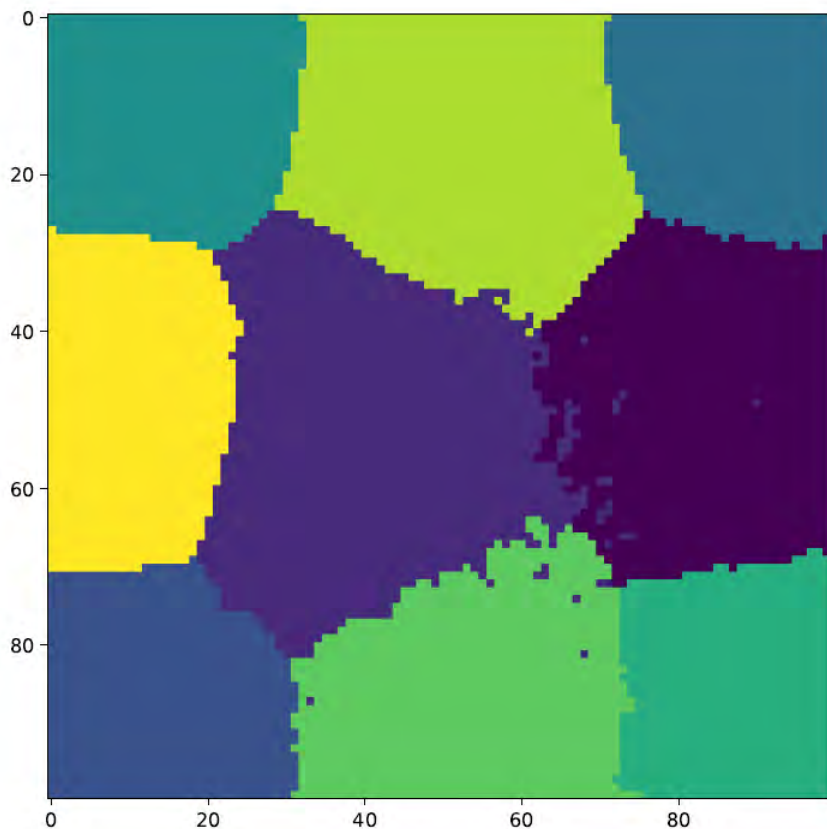
## 3 下一步工作



# 第一个思路

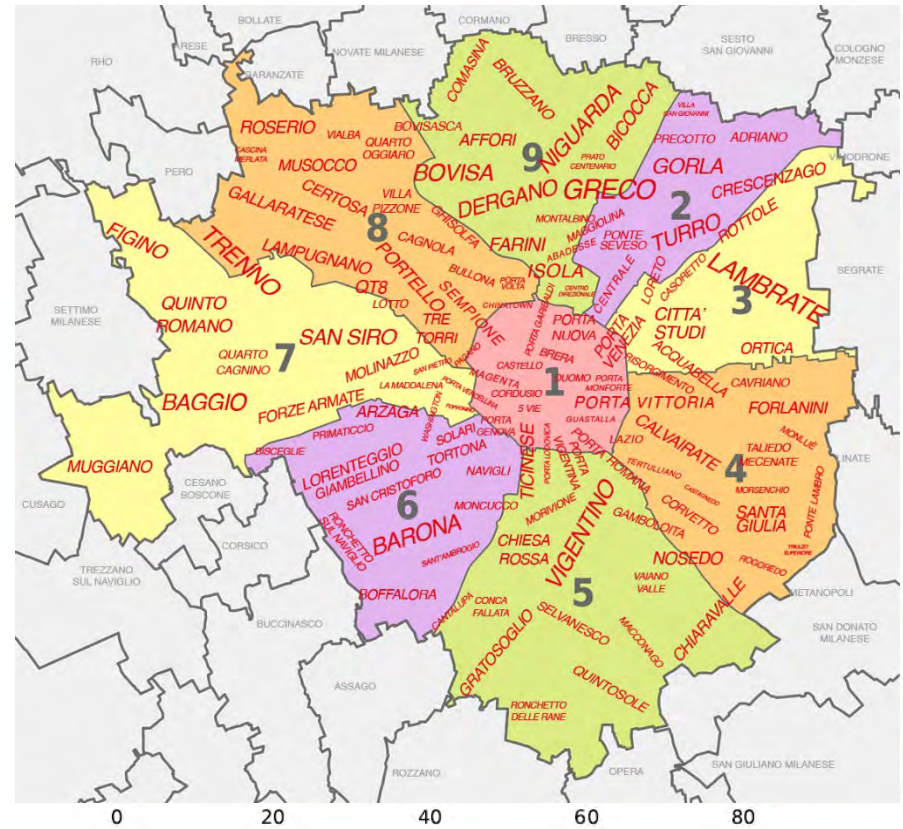
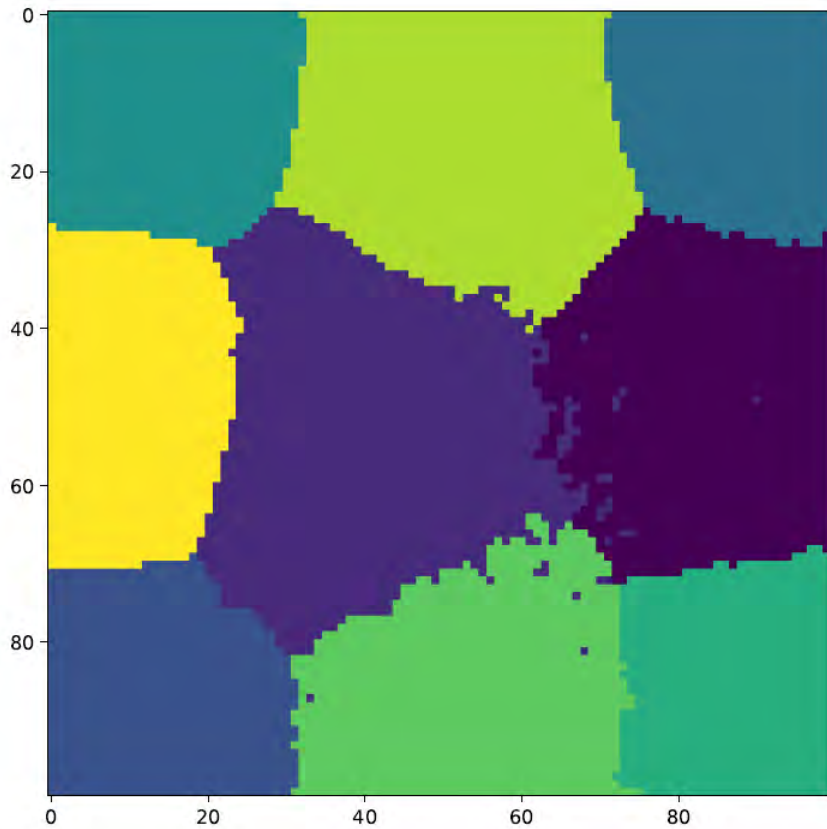


□ 城市功能区识别，然后精细化预测





## 城市功能区识别，然后精细化预测

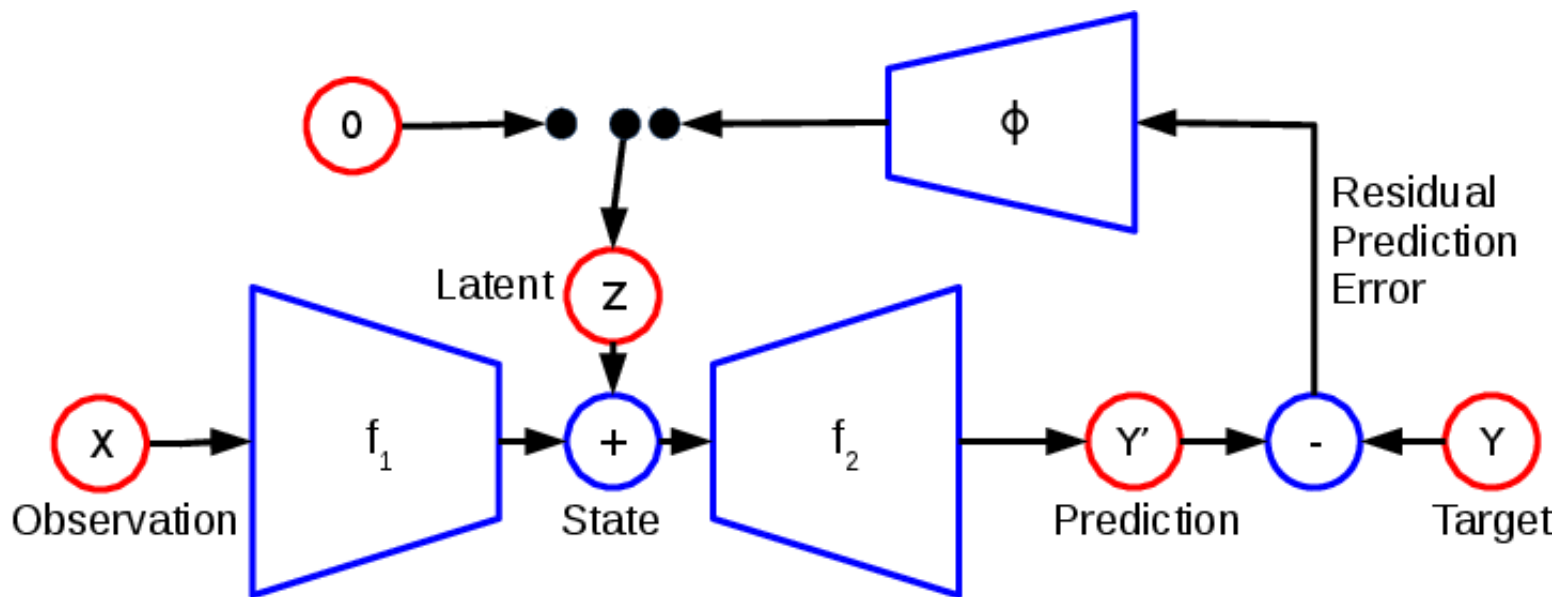




# 第二个思路



- 未来状态=确定部分+随机部分（不确定性部分）；
- 把不确定性（误差）进行编码，利用不确定性改善预测结果。



损失函数: 
$$\mathcal{L}_c(\theta, \phi) = \sum_i \|y_i - f_{\theta}(x_i, \phi(y_i - f_{\theta_-}(x_i, 0)))\|$$



## □ 基于流量预测的蜂窝网络节能

# Traffic Prediction Based Power Saving in Cellular Networks: A Machine Learning Method

<sup>1</sup>Sheng Zhang, <sup>2,3,4</sup>Shenglin Zhao, <sup>4</sup>Mingxuan Yuan, <sup>4</sup>Jia Zeng,

<sup>1</sup>Jianguo Yao, <sup>2,3</sup>Michael R. Lyu, <sup>2,3</sup>Irwin King \*

<sup>1</sup>Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiaotong University, Shanghai, China

<sup>2</sup>Shenzhen Key Laboratory of Rich Media Big Data Analytics and Application, Shenzhen Research Institute,  
The Chinese University of Hong Kong, Shenzhen, China

<sup>3</sup>Department of Computer Science & Engineering, The Chinese University of Hong Kong, Hong Kong

<sup>4</sup>Huawei Noah's Ark Lab, Hong Kong

zhangsheng1730@sjtu.edu.cn, slzhao@cse.cuhk.edu.hk, {yuan.mingxuan, zeng.jia}@huawei.com

jianguo.yao@sjtu.edu.cn, {lyu, king}@cse.cuhk.edu.hk





谢谢！

Chuanting Zhang

Code: <https://github.com/zctzzy>