

GCN原理及应用

王康

2019.1.11

目录

CONTENTS

01

CNN

02

GCN的衍生和优势

03

GCN的研究领域和概览

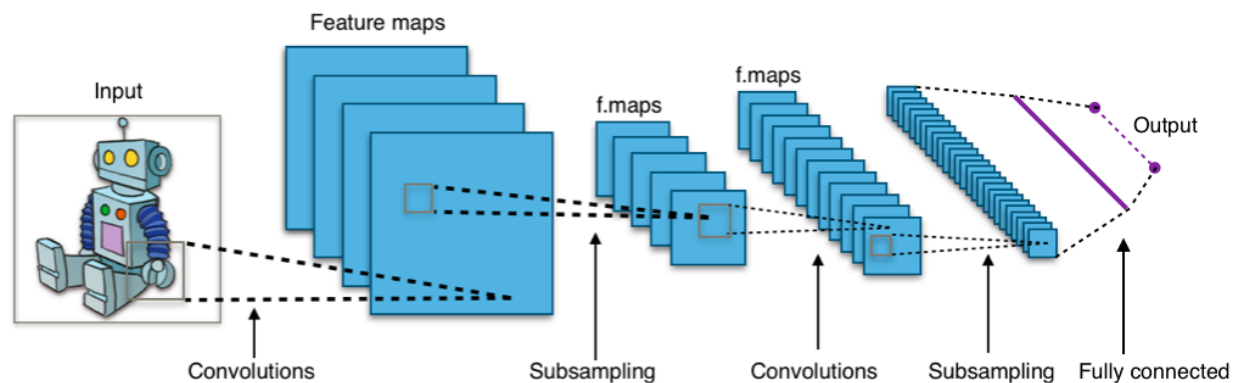
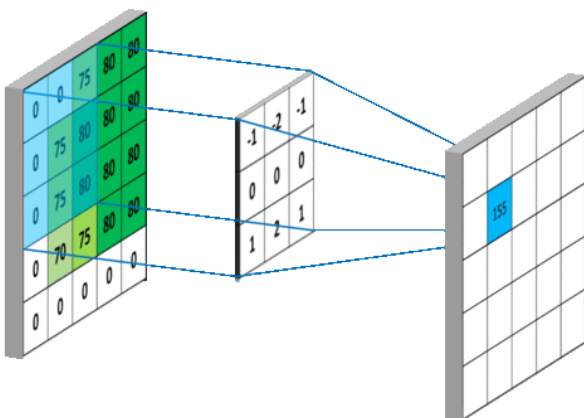
04

GCN的应用

05

Future Work and Geometric DL

CNN (Convolutional Neural Network) 卷积过程



$$(f * g)(m, n) = \sum_i \sum_j f(i, j)g(m - i, n - j)$$

CNN=Convolutional Layer+Pooling Layer+Activation layer+Fully Connected Layer

几个特点：

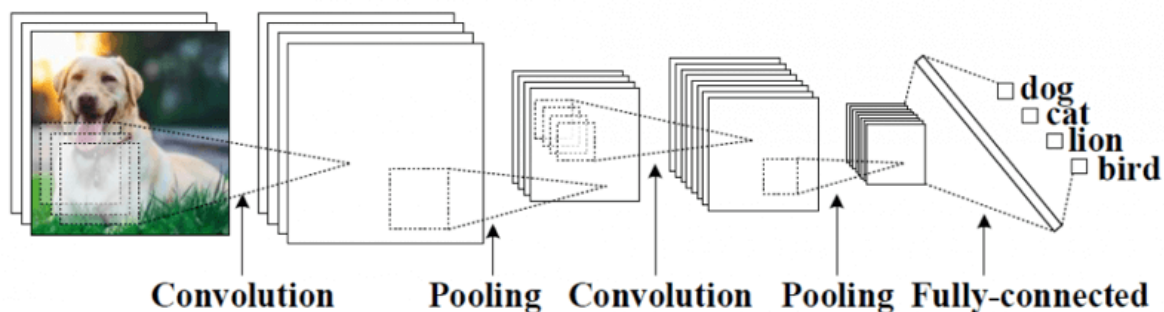
1) 局部提取(Localized)

局部提取的size大小取决于卷积核kernel的大小

2) 多尺度(Multi-scale)

分级提取不同层级的特征，例如：边缘（直线，曲线，方向）→眼睛（耳朵、鼻子等）→狗（猫、鸟等）

3) 规则格网卷积造成的平移不变性 (Stationarity)



GCN(Graph Convolutional Network)诞生的原因：

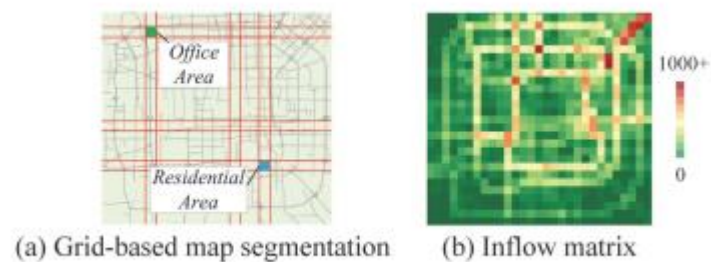
- 1) CNN无法处理Non Euclidean Structure的数据，学术上的表达是传统的离散卷积在**Non Euclidean Structure**的数据上无法保持平移不变性。通俗理解就是在拓扑图中每个顶点的相邻顶点数目都可能不同，那么当然无法用一个同样尺寸的卷积核来进行卷积运算。
- 2) 运用图论的知识，提取Graph的拓扑特征，由此催生了GCN的诞生。
- 3) GCN通过拉普拉斯矩阵保留邻近关系，并利用k-hop聚合k阶邻居对节点造成的影响。
- 4) GCN为除了CV、NLP能够解决的问题之外的其他问题提供了一种解决手段。

CNN

Euclidean Data

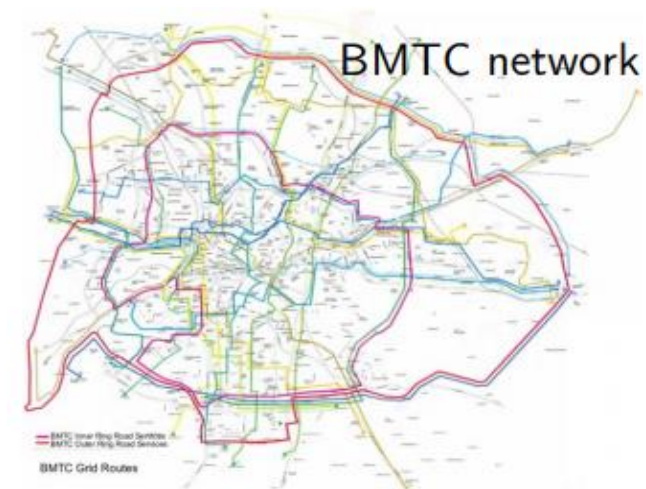


Cat vs Dog Kaggle competition

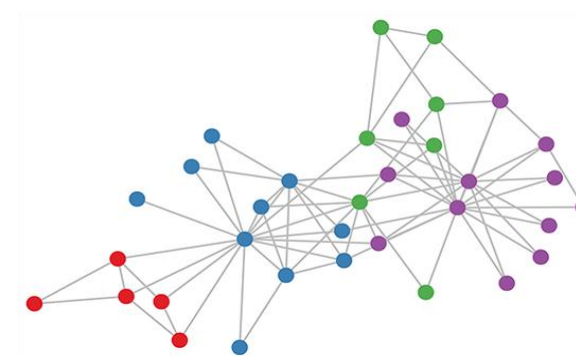


Crowd In-Out Flow Grid

Non-Euclidean Data



Road Network (Line)



Social Network (Point)

GCN

GCN是怎么做的？

- 建立节点之间的关系，使其成为一张图
- 建立全部节点的邻接矩阵
- 选择邻域节点作为当前节点的receptive field
- 依然类似采用CNN卷积操作对输入的图进行卷积操作（下一页PPT为例）

GCN的简化公式：

$$Y = \theta * W * X$$

上式中，X为输入信号值，Y为输出信号值，W为邻接矩阵， θ 为待学习参数



GCN按照研究领域划分为以下四类：

1) Spatial Domain (Vertex -Domain)

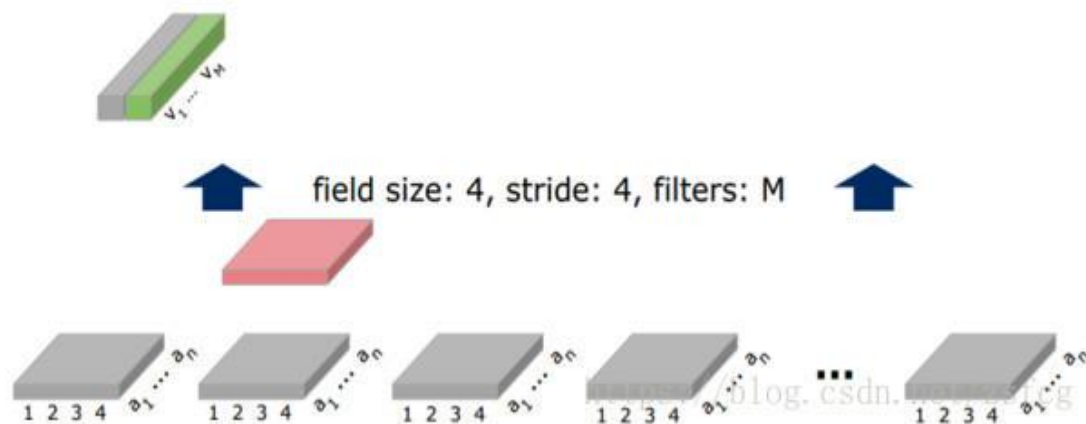
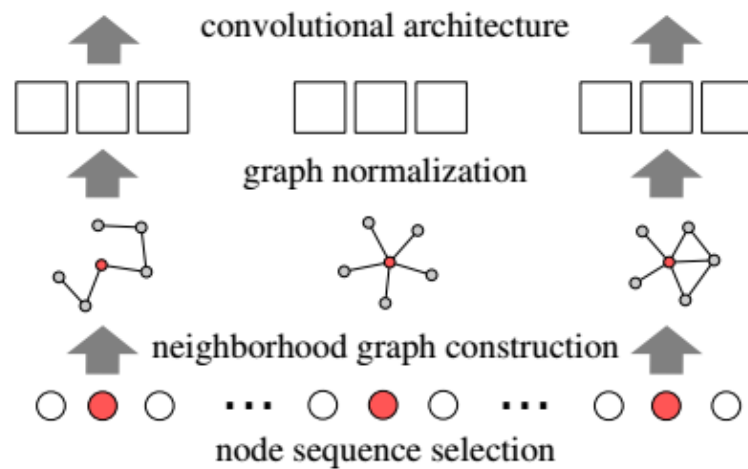
GCN在空间领域上的应用

1. 选出合适的nodes
2. 为每一个node建立一个邻域
3. 建立graph表示到 vector表示的单一映射，保证具有相似的结构特征的node可以被映射到vector当中相近的位置

缺点：

- 计算邻域节点，可能需要做dummy nodes补齐和node截断、排序
- 所有节点的邻域还是k个节点，大小一样

2016-Learning Convolutional Neural Networks for Graphs





2) Spectral Domain

频域上的应用（傅里叶变换将空间域转换为频域）

$$(f * h)_G = U((U^T h) \odot (U^T f)) \quad (1)$$

- h 为卷积核, U 为 L 的特征向量组成的矩阵, f 为signal, 也就是输入值。

$$L = I_n - D^{-1/2} W D^{-1/2} \quad (2)$$

- I_n 是单位矩阵, D 为度矩阵, W 为邻接矩阵

$$L = U \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} U^{-1} \quad (3)$$

- λ 为拉普拉斯矩阵 L 的特征值

<https://www.zhihu.com/question/54504471>



3) Spatial-Temporal Domain

图结构在时空域上的应用

2-Step :

- 时间域上操作
- 空间域上操作

典型应用：具有图结构的时空数据

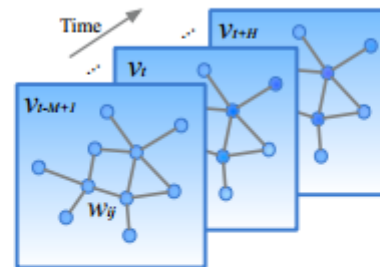


Figure 1: Graph-structured traffic data. Each v_t indicates a frame of current traffic status at time step t , which is recorded in a graph-structured data matrix.



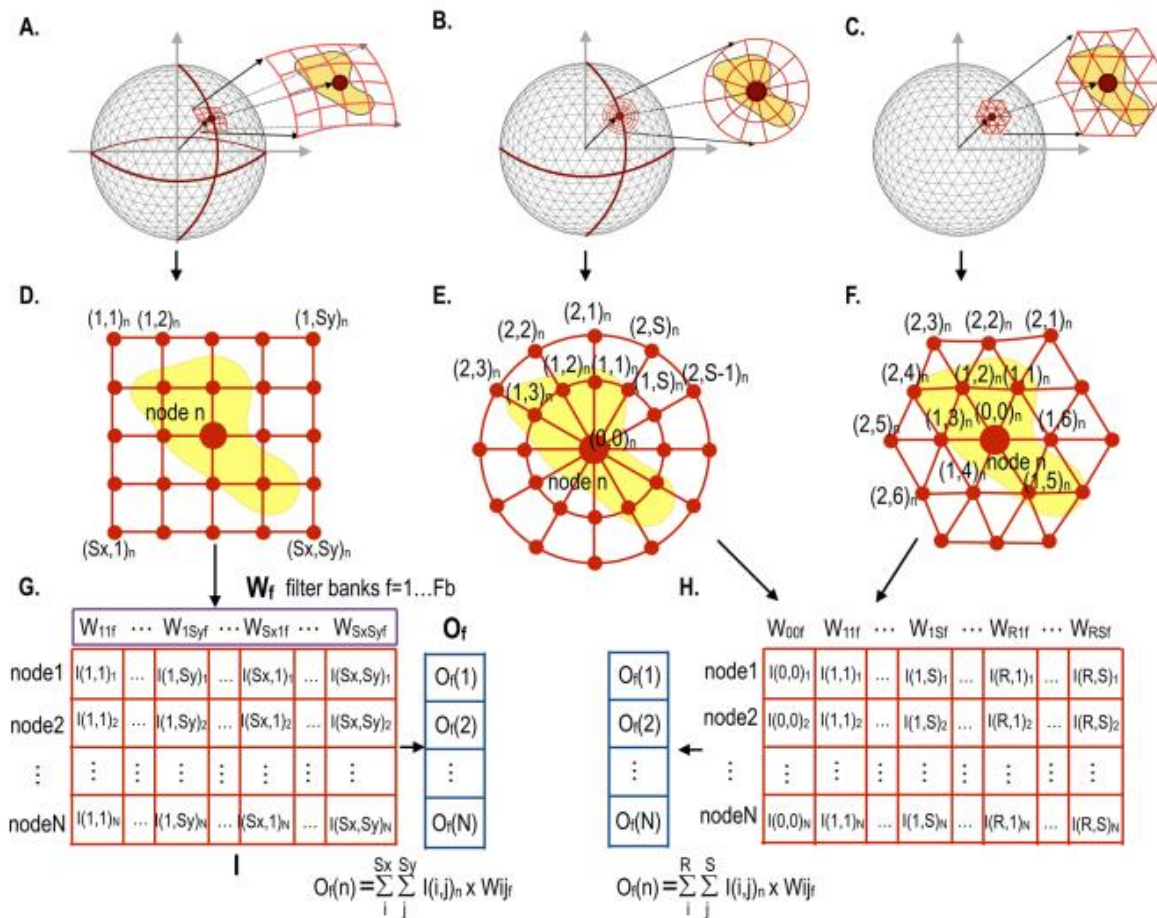
4) Manifold Domain

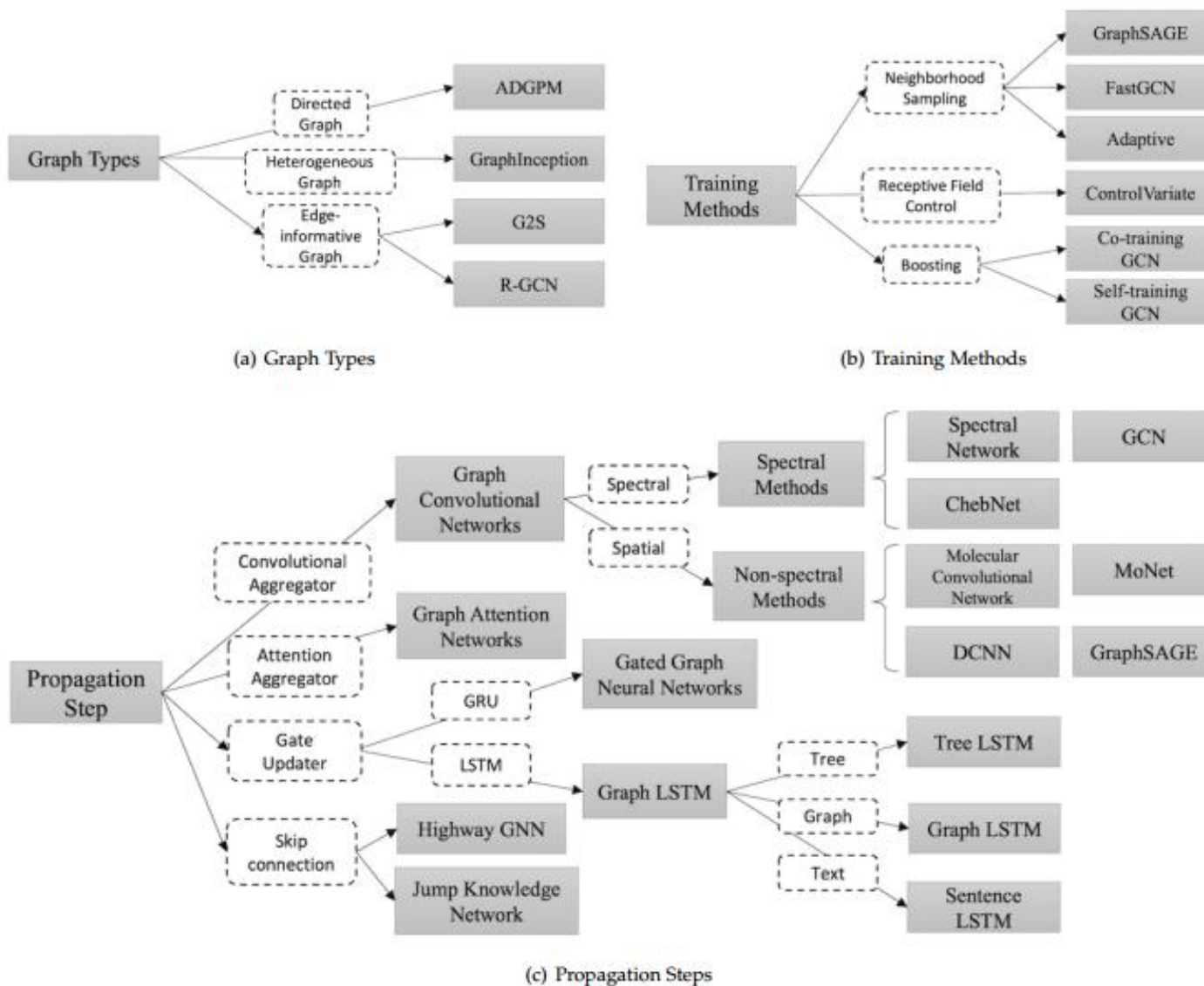
图卷积在流形领域的应用

什么是流形？

图卷积在流形上的操作流程：

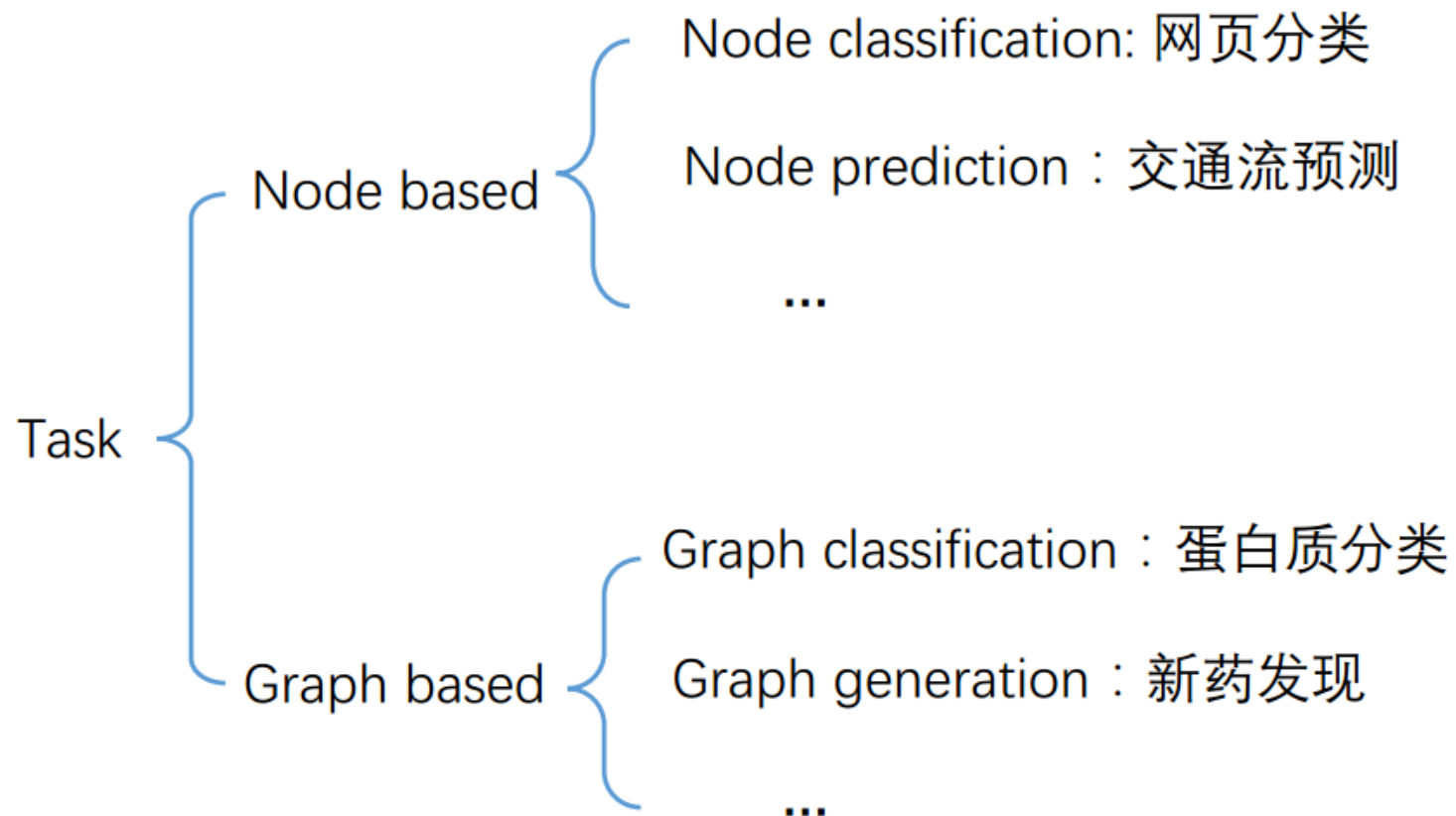
- 对流形数据mesh划分
- 每个mesh就是一个node，建立邻域矩阵
- 应用图卷积





GCN 有哪些可以改进的地方？

- 1) 邻域采样的手段（所有邻域全采样还是部分采样）
- 2) 邻域节点重要性
- 3) k-阶近似
- 4) re-normalization
- 5) aggregator的选择
- 6) shallow structure



Irregular domain



Problem :

给定道路当前时间之前的某段时间的交通流状况，预测下一个时间点的交通流状况

Methods :

ARIMA (Autoregressive Integrated Moving Average model) 系列

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) (1 - L)^d X_t = \left(1 + \sum_{i=1}^q \theta_i L^i\right) \varepsilon_t$$

ARIMA=自回归(AR)+差分整合(I)+移动平均(MA)

自回归：把信号序列看做是前k项的历史信号的线性加权求和加上随机项

移动平均：把信号序列看做平均值加上随机平均值附近的扰动误差

差分整合：差分的目的是为了将非平稳序列通过差分形式变成平稳序列

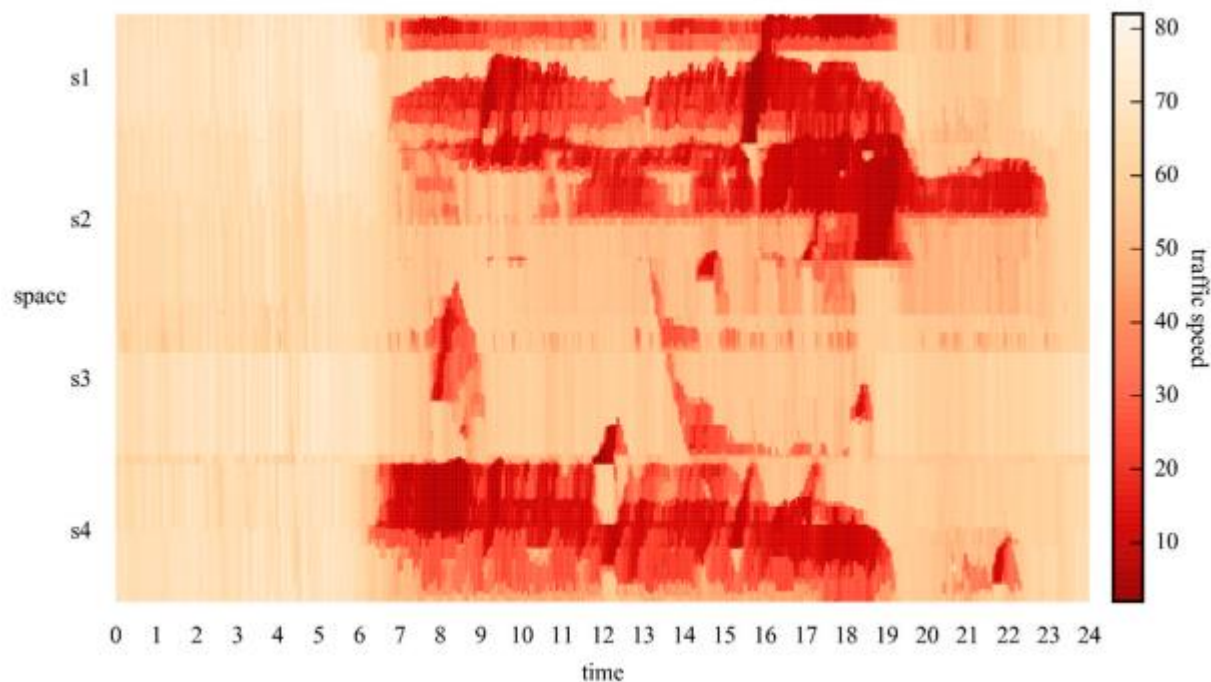
平稳序列：基本上不存在趋势的序列



GCN in Traffic Prediction (续)

CNN：将道路划分为网格，可能存在一个网格多条道路的问题

Ma X, Dai Z, He Z, et al. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction[J]. Sensors, 2017, 17(4): 818.

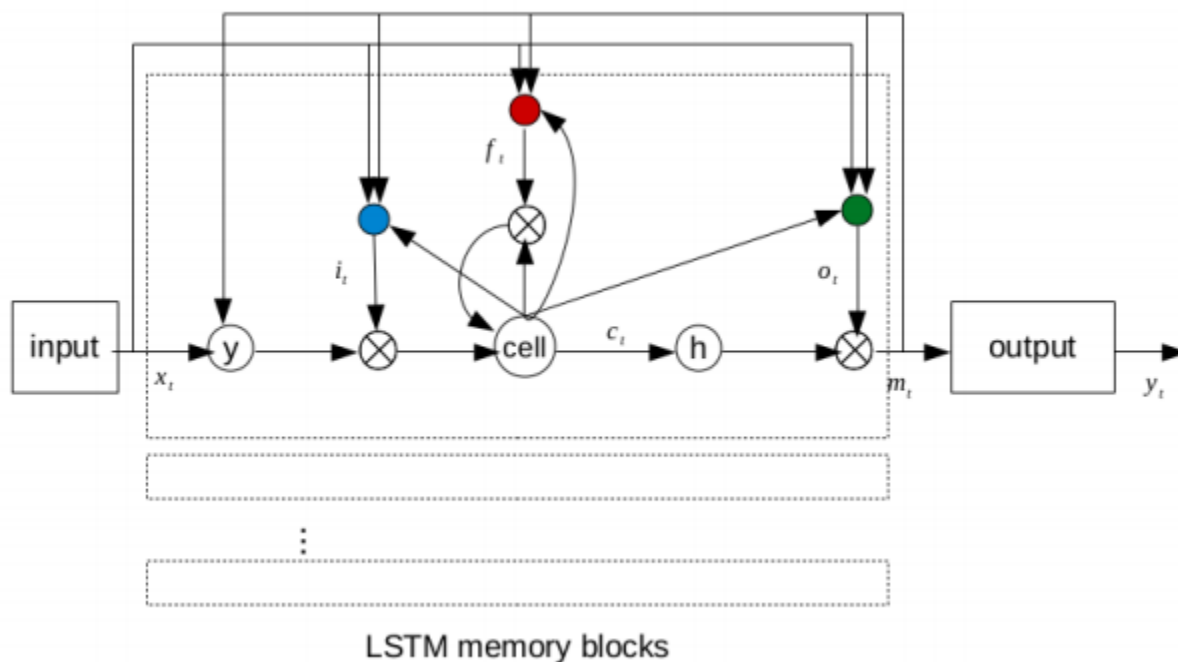


北京二环线



GCN in Traffic Prediction (续)

LSTM/GRU/RNN系列：因时间记忆机制的优势擅长时序数据的训练，单独采用该方法的时候，忽视了Spatial dependency



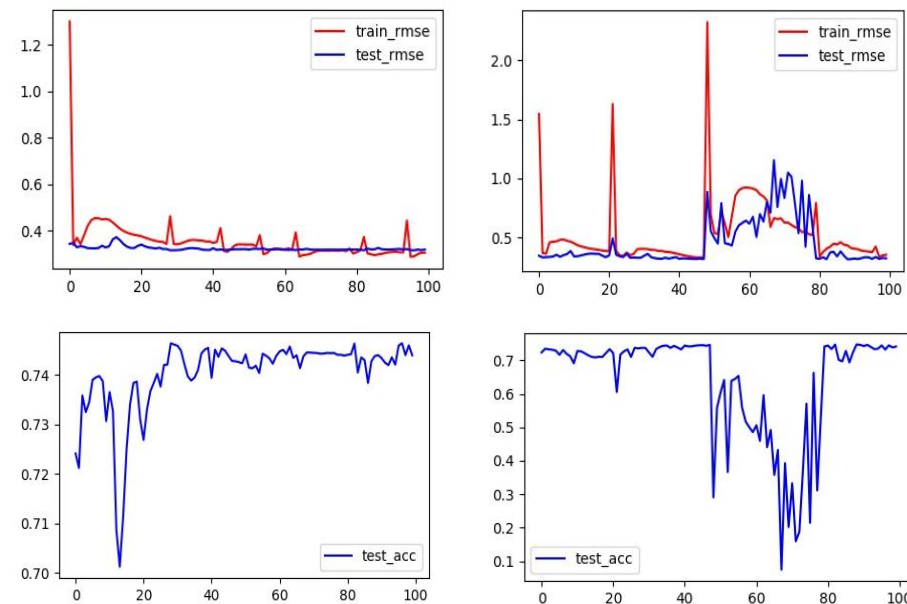
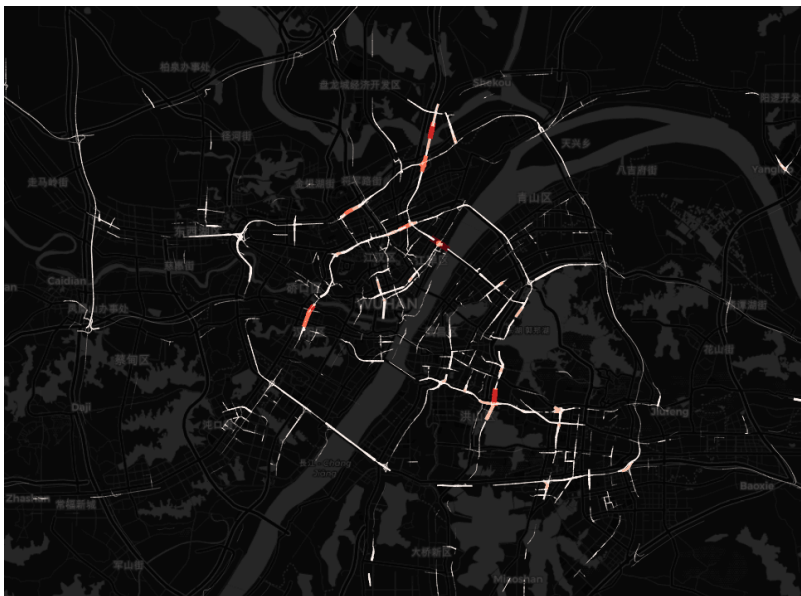
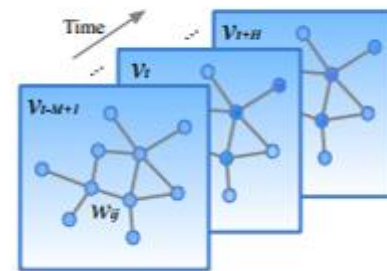
Azzouni A, Pujolle G. A long short-term memory recurrent neural network framework for network traffic matrix prediction[J]. arXiv preprint arXiv:1705.05690, 2017.



GCN in Traffic Prediction (续)

Cheby-Net/GraphSAGE/GAT/Fast-GCN/GCN等GNN系列：

引入Graph领域知识，建立空间邻接矩阵，不同的模型按照不同的手段建立感受野的范围，实现节点在邻域范围内的message passing，从而实现空间关系依赖的保留





1, Dynamic Graph

大部分的现有的网络都是按照fixed graph进行后续计算，对于节点动态更新、新增、删除的graph如何研究？

2, Complex task

当解决复杂任务时，如多标签节点分类，边上带特征的网络，异质网络，如何设计模型？

3, Scalability

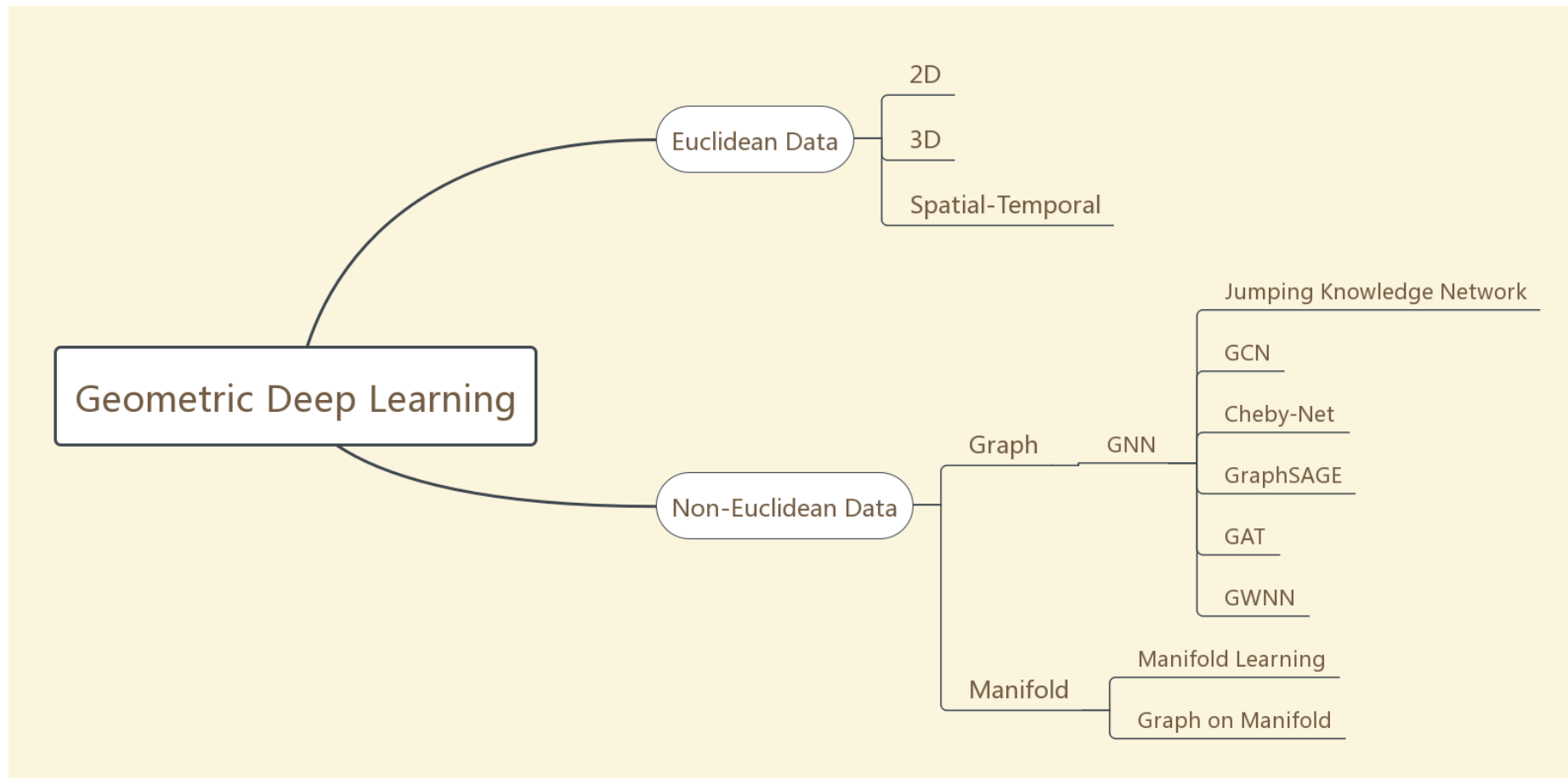
当节点规模数量巨大的时候，目前的GCN网络训练都是采用的One-Batch（针对节点而言）的方式进行训练，这样计算性能会成为瓶颈。为了提升性能，如何将节点打散，进行分批训练，这样更具有实际的应用价值。目前已经有少部分的研究已经利用将节点embedding化，计算节点数达billion级别的图卷积。

4, Non-Structural Scenarios

如何对非图结构的数据进行图结构化，建立节点之间的联系或者结合Graph Information对Non-Structural Data做补充，进而利用graph neural network进行计算？

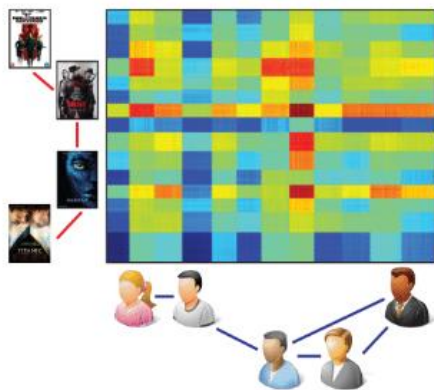
5, Shallow Structure

传统的深度神经网络通过残差模型可以实现网络深度达数百层，然而，在GNN（Graph Neural Network）方面，目前的大部分实验表明利用k-hop邻域的节点进行训练时，往往在k比较小（k=2）时能够很好地拟合数据，当k过大时，容易over-smoothing。即使融合残差网络模型，也会出现模型过拟合问题。因此，如何设计deep GNN也是一个不小的挑战。

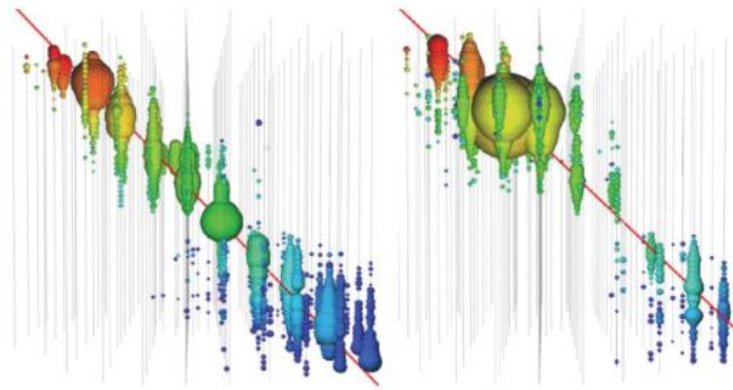




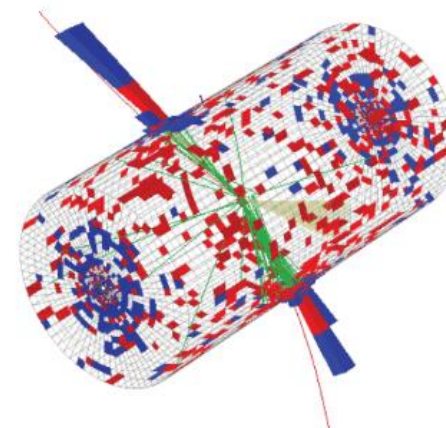
Geometric Deep Learning Application



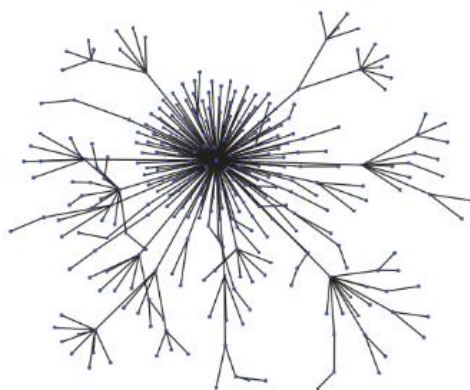
Recommender system



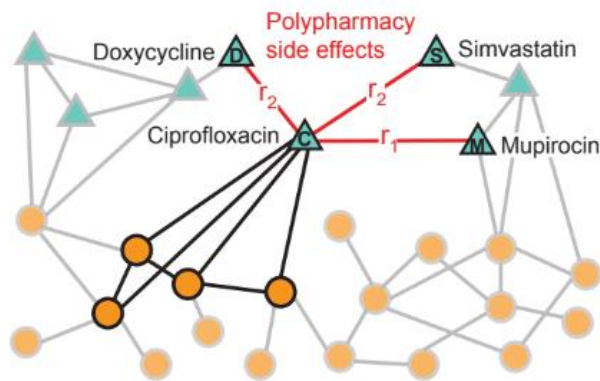
Neutrino detection



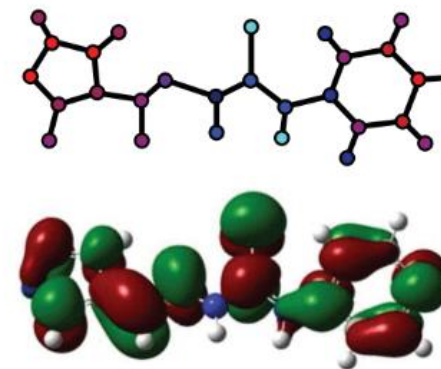
LHC



Fake news detection



Drug repurposing



Chemistry



Geometric Deep Learning Application (3D)



Computer graphics



Virtual/augmented reality



Robotics



Autonomous driving



Medicine



Drug design

- [1] Ying R, He R, Chen K, et al. Graph Convolutional Neural Networks for Web-Scale Recommender Systems[J]. arXiv preprint arXiv:1806.01973, 2018.
- [2] Veličković P, Cucurull G, Casanova A, et al. Graph Attention Networks[J]. arXiv preprint arXiv:1710.10903, 2017.
- [3] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[C]//Advances in Neural Information Processing Systems. 2017: 1024-1034.
- [4] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering[C]//Advances in Neural Information Processing Systems. 2016: 3844-3852.
- [5] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
- [6] Bronstein M M, Bruna J, LeCun Y, et al. Geometric deep learning: going beyond euclidean data[J]. IEEE Signal Processing Magazine, 2017, 34(4): 18-42.
- [7] Manessi F, Rozza A, Manzo M. Dynamic Graph Convolutional Networks[J]. arXiv preprint arXiv:1704.06199, 2017.
- [8] Xu K, Li C, Tian Y, et al. Representation Learning on Graphs with Jumping Knowledge Networks[J]. arXiv preprint arXiv:1806.03536, 2018.
- [9] Bingbing Xu, Huawei Shen, Qi Cao, et al. Graph Wavelet Neural Network [J]. ICLR2019(Under double-blind review)
- [10] Zhou J, Cui G, Zhang Z, et al. Graph Neural Networks: A Review of Methods and Applications[J]. arXiv preprint arXiv:1812.08434, 2018.

Any
Question ?