2020-2021学年秋季学期

自然语言处理
Natural Language Processing

授课教师： 胡玥

助 教： 于静

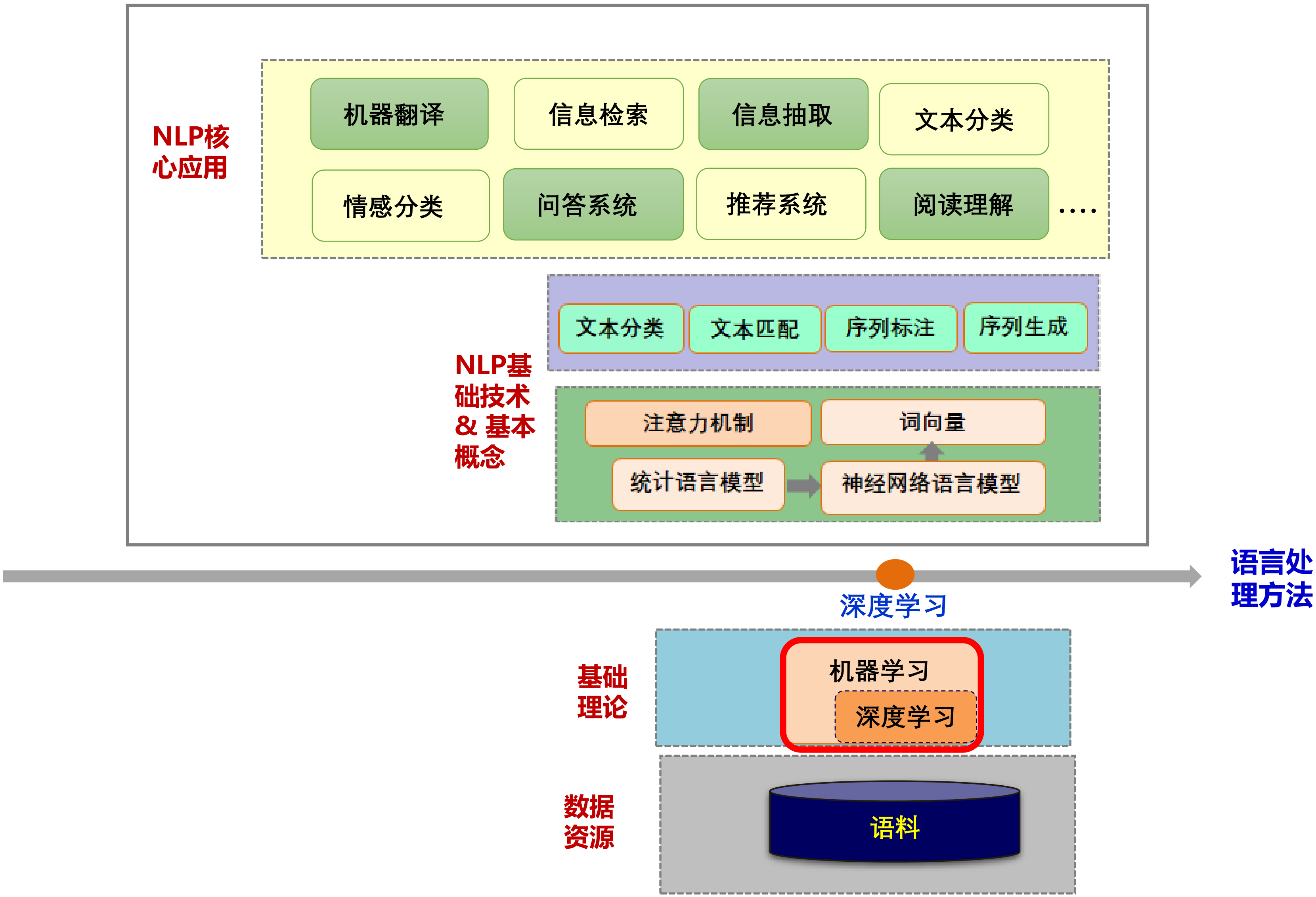
自然语言处理
Natural Language Processing

第 6 章　图卷积神经网络

授课教师：　胡玥

授课时间：2020.9

基于深度学习的自然语言处理课程内容



概 要

本章主要内容：

介绍图卷积神经网络（GNN）的基本概念，模型结构

本章教学目的:

了解并掌握图卷积神经网络（GNN）的相关知识

内 容 提 要

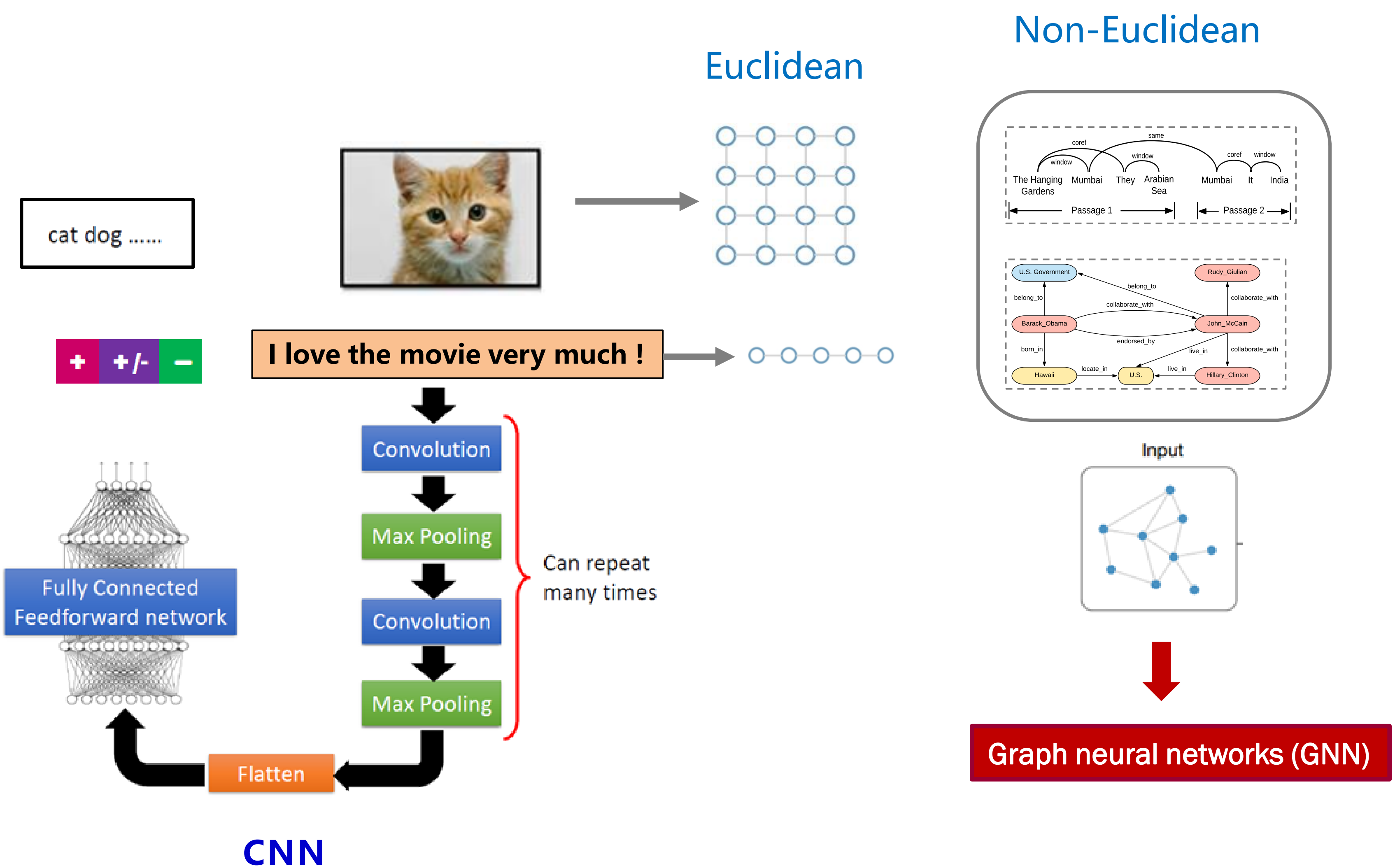
6.1　概述

6.2　Spatial-based GNN

6.3　GNN 变形

6.1 概述

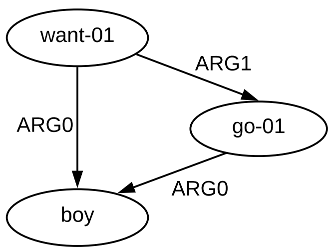
问题引入：

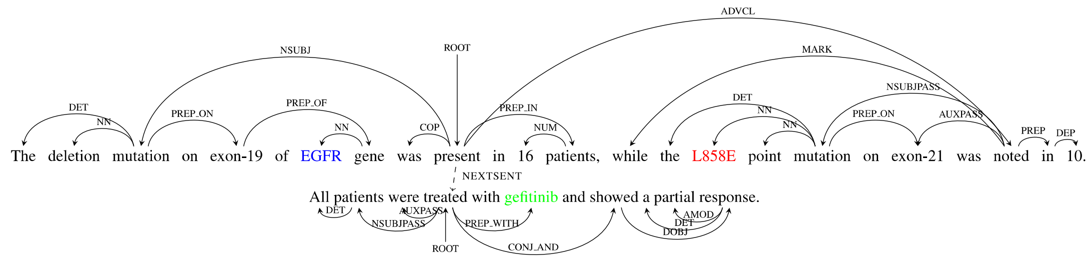


Euclidean
Non-Euclidean
cat dog ……
+
+/-
−
I love the movie very much !
Convolution
Max Pooling
Convolution
Max Pooling
Can repeat many times
Fully Connected Feedforward network
Flatten
CNN
coref
same
window
The Hanging Gardens
Mumbai
They
Arabian Sea
Mumbai
It
India
Passage 1
Passage 2
U.S. Government
Rudy_Giulian
belong_to
collaborate_with
Barack_Obama
John_McCain
endorsed_by
born_in
live_in
Hawaii
locate_in
U.S.
Hillary_Clinton
Input
Graph neural networks (GNN)

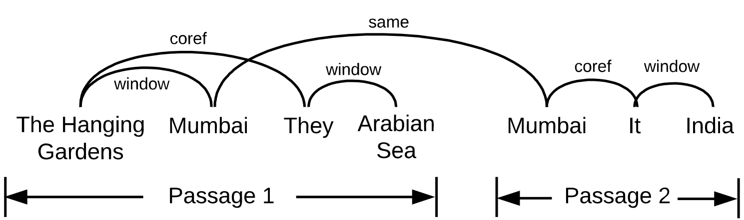
6.1 概述

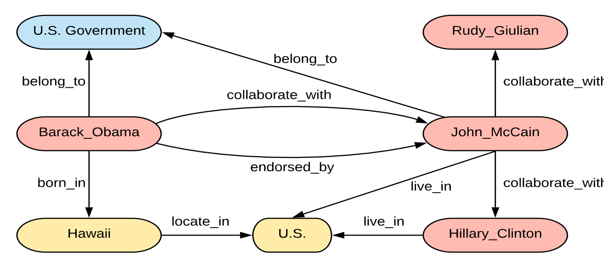
GNN in NLP





Graph neural networks (GNN)



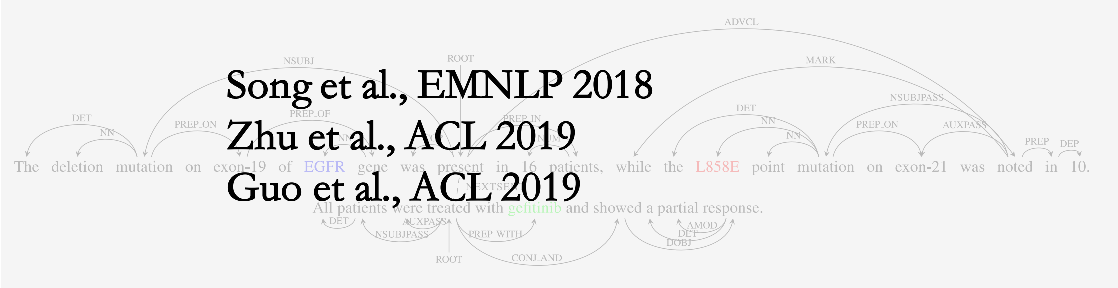


…

6.1 概述

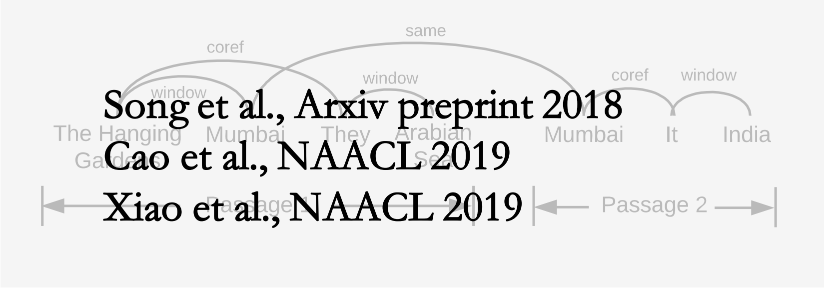
GNN in NLP

Song et al., ACL 2018
Song et al., TACL 2019
Ding et al., ACL 2019
Tu et al., ACL 2019

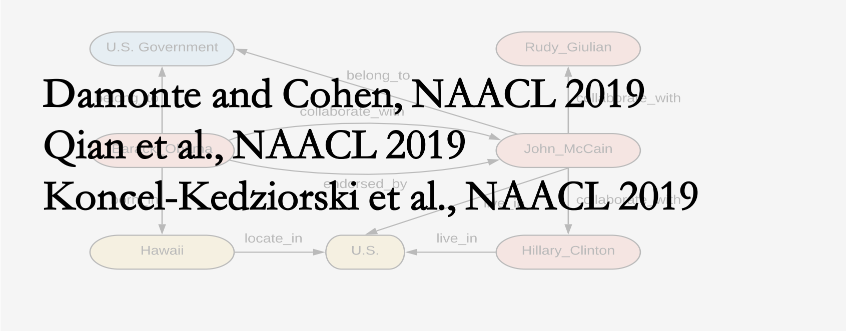


Song et al., EMNLP 2018
Zhu et al., ACL 2019
Guo et al., ACL 2019

Graph neural networks (GNN)



Song et al., Arxiv preprint 2018
Cao et al., NAACL 2019
Xiao et al., NAACL 2019



Damonte and Cohen, NAACL 2019
Qian et al., NAACL 2019
Koncel-Kedziorski et al., NAACL 2019

...

6.1 概述

图卷积神经网络

- **Spectral-based Graph Convolutional neural Networks**

Define Convolution in Spectral domain , Convolutional is defined via graph Fourier transform and Convolutional theorem.

Challenge : Convolution filter define in spectral domain (not in vertex domain)

- **Spatial-based Graph Convolutional neural Networks**

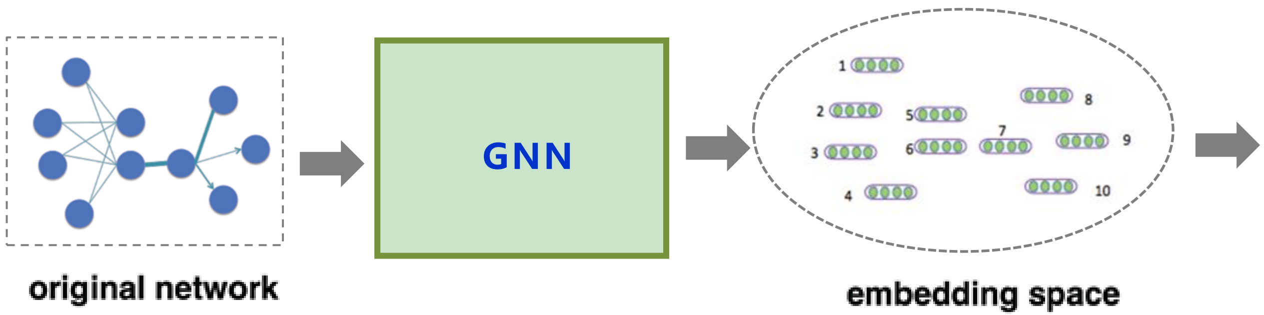
Define Convolution in the vertex domain , Convolutional is defined as a weighted average function over all vertices located in the neighborhood of target vertex.

Challenge : The size of neighborhood varies remarkably across nodes

内 容 提 要

6.2 Spatial-based GNN

GNN Goal



- Node Classification
- Node Clustering
- Link Prediction
- Recommendation
- Document Classification

.....

输入： original network

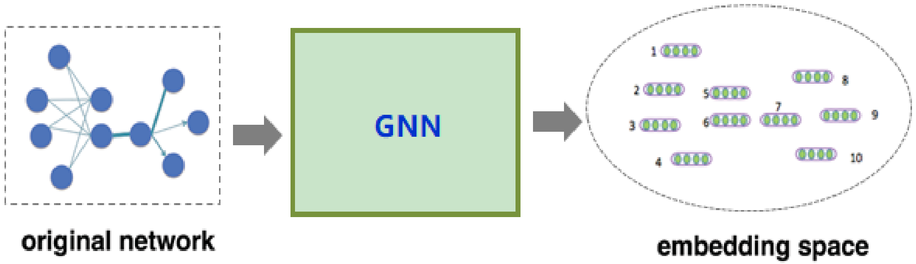
输出： Node representation set

每个表示带有原图的特征信息

6.2 Spatial-based GNN

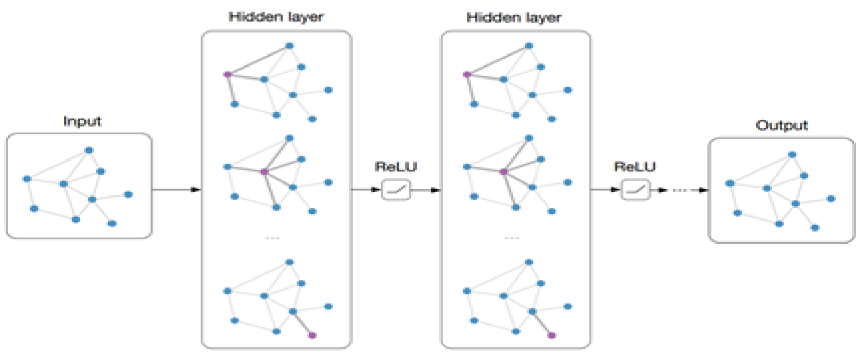
GNN模型结构：

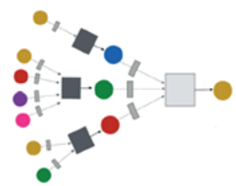
输入



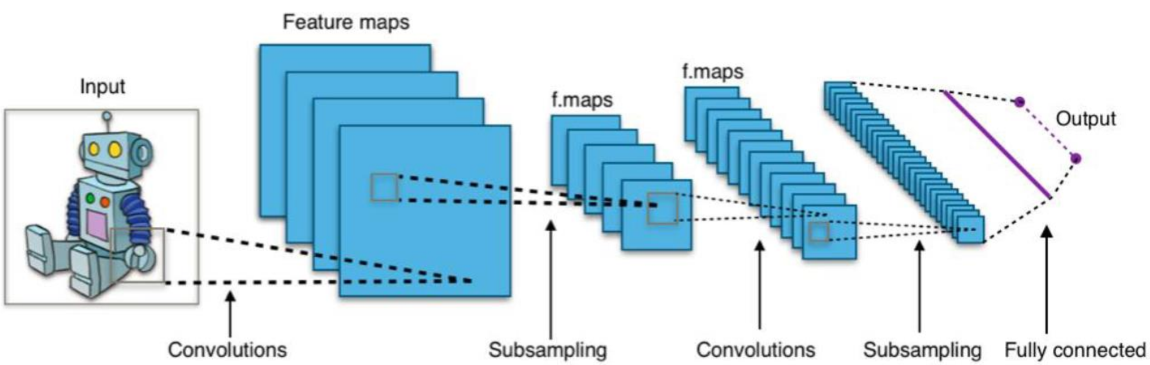
输出

GNN





CNN



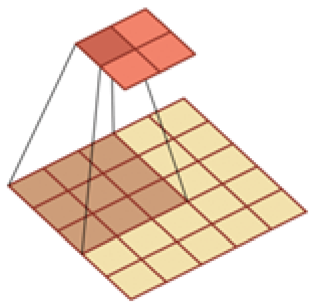
- **卷积层**
  Weight sharing
- **池化层**
  Reduce the number of parameters

6.2 Spatial-based GNN

GNN参数如何确定

- **参数个数**

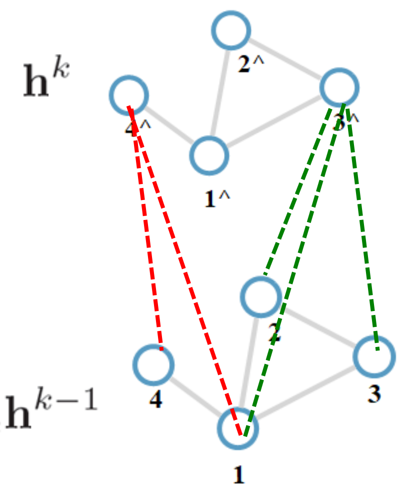
CNN



L 层到 L+1 对于同一个卷积核共享一组参数

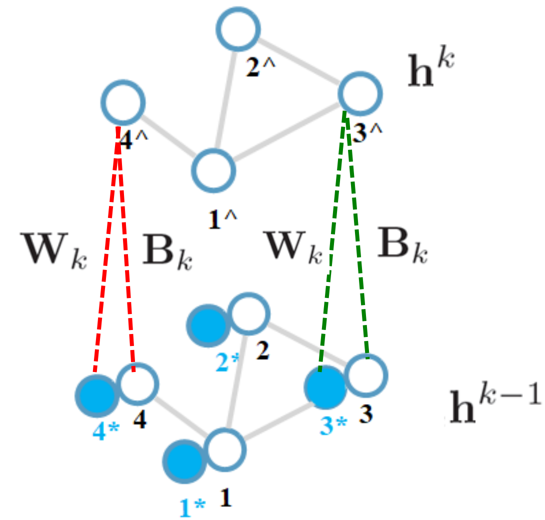
（参数个数固定）

GNN



问题：邻接节点个数不确定

如何解决?



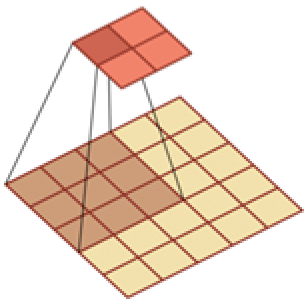
实心结点是对应的空心结点的邻接结点聚集

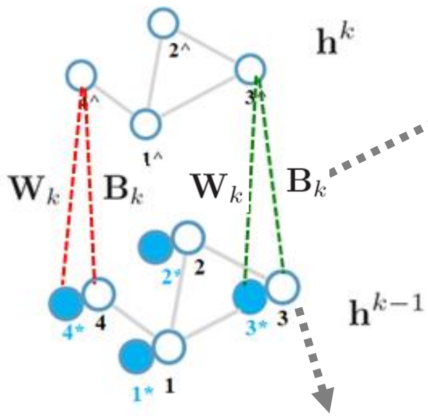
6.2 Spatial-based GNN

■ **参数维度**

$$\mathbf{X}^{(\mathbf{h}^{k-1})} \times \mathbf{B} = \mathbf{X}^{(\mathbf{h}^{k})}$$

h^k: feature matrix $\mathbf{X}$^∈ $\mathbb{R}^{N \times K}$





参数B

	k1	k2	k3	k4
w1	0.1	0.7	1.2	1.1
w2	0.3	0.4	0.3	0.1
w3	0.5	0.3	1.5	0.4
w4	1.0	0.1	0.2	0.5
w5	2.0	0.4	0.1	2.4

h^{k-1}:weight matrix $\boldsymbol{B} \in \mathbb{R}^{M \times K}$

以X为例，X* 同理

X* 为 X 的邻接聚集结点

图上结点X

	f1	f2	f3	f4	f5
Node 1	0.1	0.7	1.2	1.1	0.9
Node 2	0.3	0.4	0.3	0.1	1.2
Node 3	0.5	0.3	1.5	0.4	0.6
Node 4	1.0	0.1	0.2	0.5	0.1

h^{k-1}:feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times M}$

参数：W，B $\in \mathbb{R}^{M \times K}$

M：$\mathbf{X}^{(\mathbf{h}^{k-1})}$ 特征维度

K：$\mathbf{X}^{(\mathbf{h}^{k})}$ 特征维度

- Weight sharing
- Reduce the number of parameters

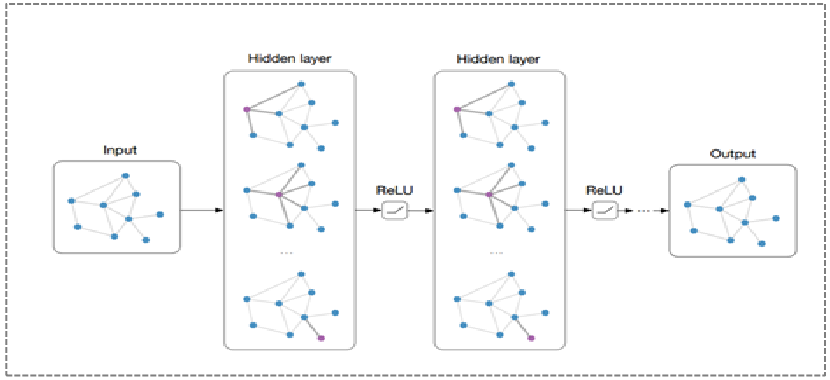
6.2 Spatial-based GNN

GNN 卷积步骤：

Step1: Aggregation

Step2: Transformation

GNN 池化？



Basic GNN

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1}\right)$$

Initial "layer 0" embeddings are equal to node features

$$\boxed{\mathbf{h}_v^0 = \mathbf{x}_v}$$

previous layer embedding of v

$$\boxed{\mathbf{h}_v^k} = \boxed{\sigma}\left(\mathbf{W}_k \boxed{\sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|}} + \mathbf{B}_k \boxed{\mathbf{h}_v^{k-1}}\right), \quad \forall k > 0$$

kth layer embedding of v

non-linearity (e.g., ReLU or tanh)

average of neighbor's previous layer embeddings

GCN

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v) \cup v} \frac{\mathbf{h}_u^{k-1}}{\sqrt{|N(u)||N(v)|}}\right)$$

use the same transformation matrix for self and neighbor embeddings

instead of simple average, normalization varies across neighbors

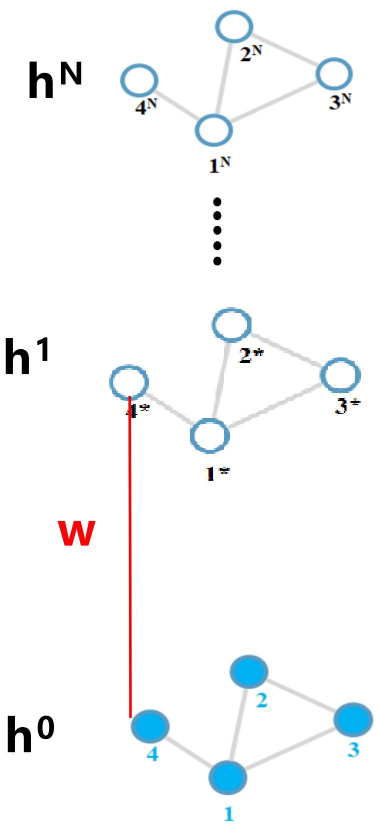
- More parameter sharing.
- Down-weights high degree neighbors.

Kipf et al.'s Graph Convolutional Networks (GCNs) are a slight variation on the neighborhood

例：

Output

$$\mathbf{h}_v^k = \sigma\left(\boxed{\mathbf{W}_k \boxed{\sum_{u\in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|}}} + \mathbf{B}_k \mathbf{h}_v^{k-1}\right)$$



Step2: Transformation

	f1	f2	f3	f4	f5
Node 1	1.8	0.8	...		
Node 2	0.6	1.0	...		
Node 3	...				
Node 4	...				

feature matrix $X' \in \mathbb{R}^{N \times M}$ ×

	k 1	k2	k3	k 4
w 1	0.1	0.7	1.2	1.1
w 2	0.3	0.4	0.3	0.1
w 3	0.5	0.3	1.5	0.4
w 4	1.0	0.1	0.2	0.5
w 5	2.0	0.4	0.1	2.4

weight matrix $W \in \mathbb{R}^{M \times K}$ W =

	k1	k 2	k3	k4
Node 1	2.8	4.8	...	
Node 2	0.6	4.0	...	
Node 3	...			
Node 4	...			

feature matrix $H \in \mathbb{R}^{N \times K}$

Step1: Aggregation

	Node 1	Node 2	Node 3	Node 4
Node 1	0	1	1	1
Node 2	1	0	1	0
Node 3	1	1	0	0
Node 4	1	0	0	0

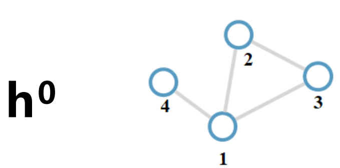
adjacency matrix $A \in \mathbb{R}^{N \times N}$ ×

	f1	f2	f3	f4	f5
Node 1	0.1	0.7	1.2	1.1	0.9
Node 2	0.3	0.4	0.3	0.1	1.2
Node 3	0.5	0.3	1.5	0.4	0.6
Node 4	1.0	0.1	0.2	0.5	0.1

feature matrix $X \in \mathbb{R}^{N \times M}$ =

	f1	f2	f3	f4	f5
Node 1	1.8	0.8	...		
Node 2	0.6	1.0	...		
Node 3	...				
Node 4	...				

feature matrix $X' \in \mathbb{R}^{N \times M}$



Input：

	Node 1	Node 2	Node 3	Node 4
Node 1	0	1	1	1
Node 2	1	0	1	0
Node 3	1	1	0	0
Node 4	1	0	0	0

adjacency matrix A

	f1	f2	f3	f4	f5
Node 1	0.1	0.7	1.2	1.1	0.9
Node 2	0.3	0.4	0.3	0.1	1.2
Node 3	0.5	0.3	1.5	0.4	0.6
Node 4	1.0	0.1	0.2	0.5	0.1

feature matrix $X \in \mathbb{R}^{N \times M}$

6.2 Spatial-based GNN

GNN模型训练：

trainable matrices
(i.e., what we learn)

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

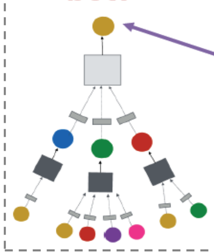
$$\mathbf{h}_v^k = \sigma\left(\boxed{\mathbf{W}_k} \sum_{u\in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \boxed{\mathbf{B}_k}\mathbf{h}_v^{k-1}\right), \ \forall k \in \{1,\dots,K\}$$

$$\boxed{\mathbf{z}_v = \mathbf{h}_v^K}$$

在最后一层（K层）得到每个结点的表示后，可以根据任务将其代入任何损失函数，然后用梯度下降法训练参数

例1：　结点分类（有监督）

Human or bot?



classification weights

$$\mathcal{L} = \sum_{v\in V} \boxed{y_v}\log(\sigma(\boxed{\mathbf{z}_v}^\top\boxed{\theta})) + (1-\boxed{y_v})\log(1-\sigma(\boxed{\mathbf{z}_v}^\top\boxed{\theta}))$$

output node embedding

node class label

例2：　相似结点表示相似（无监督）

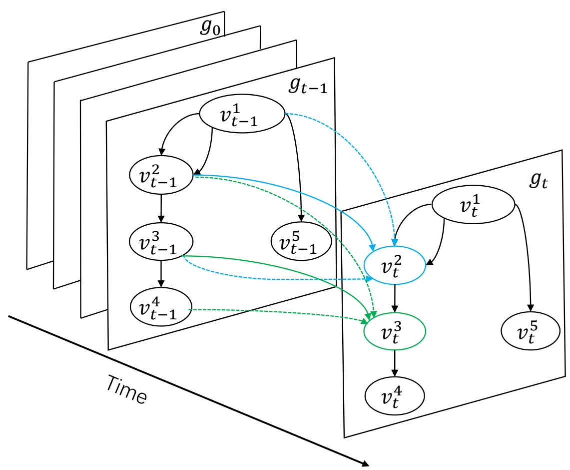
6.2 Spatial-based GNN

有向图的GNN：

GNN 卷积步骤：

Step1: Aggregation

Step2: Transformation



$$\mathbf{h}_{t-1}^k \longrightarrow \mathbf{h}_t^i$$

Step1: Aggregation

$$\mathbf{m}_t^{\uparrow i} = \sum_{k \in \Omega_{\uparrow}(i)} \mathbf{h}_{t-1}^k$$

$$\mathbf{m}_t^{\downarrow i} = \sum_{k \in \Omega_{\downarrow}(i)} \mathbf{h}_{t-1}^k$$

$$\mathbf{m}_t^i = [\mathbf{m}_t^{\uparrow i}; \mathbf{m}_t^{\downarrow i}]$$

Step2: Transformation

$$\mathbf{h}_t^i = \sigma(\mathbf{W}_g^m \mathbf{m}_t^i + \mathbf{W}_g^x \mathbf{x}_t^i + \mathbf{b}_g)$$

内 容 提 要

6.1　概述

6.2　Spatial-based GNN

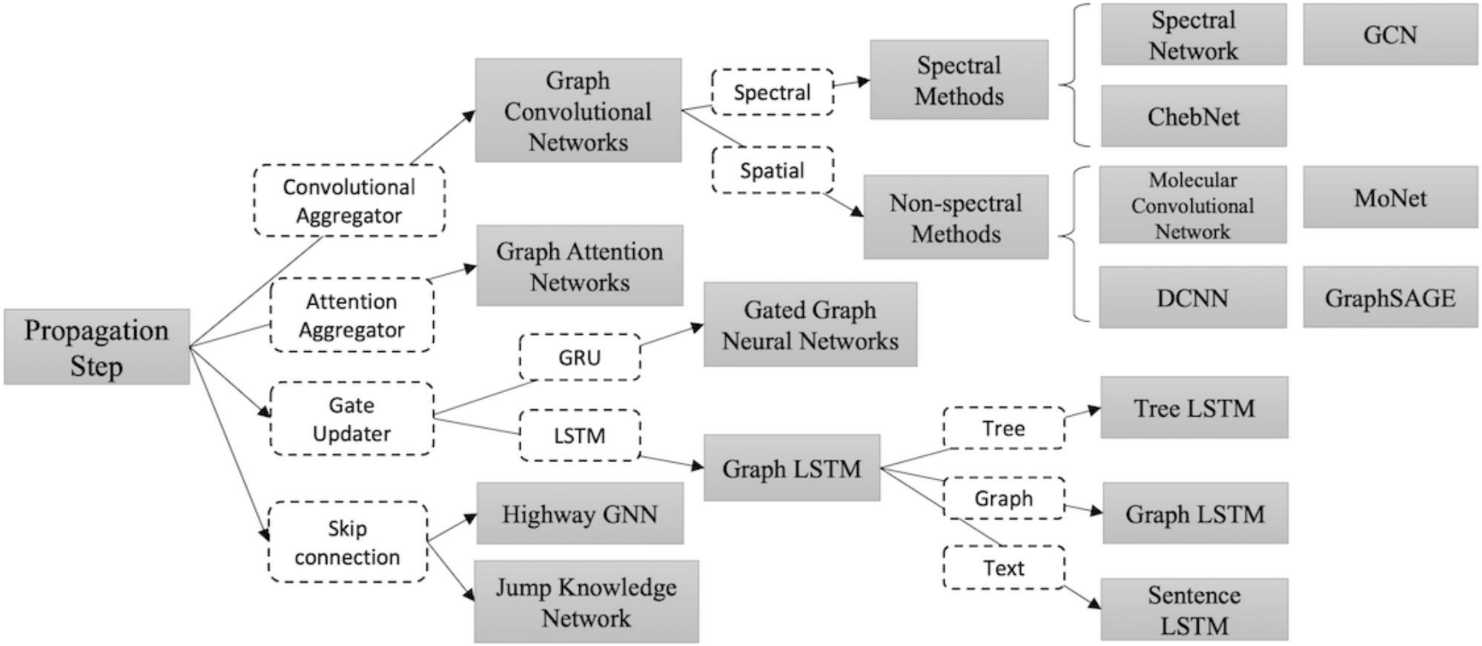
6.3　GNN 变形

6.3 GNN变形

GNN变形：

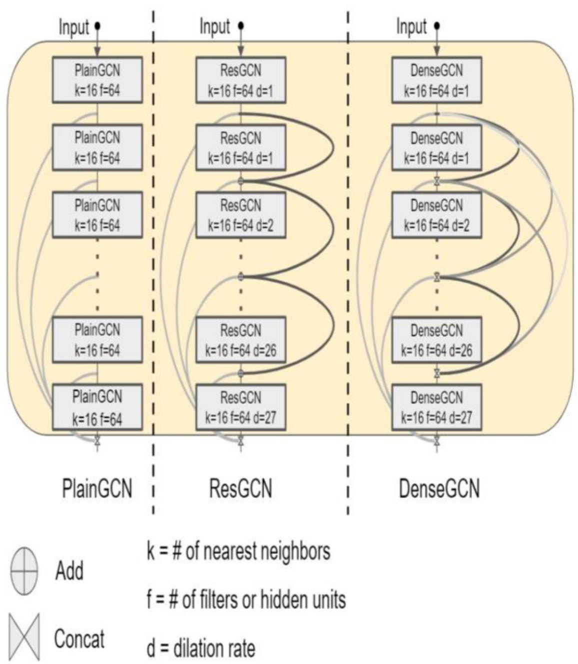
根据结点聚集和层级连接方法的不同有大量不同形式的GNN

- GNN Models based on Propagation Step

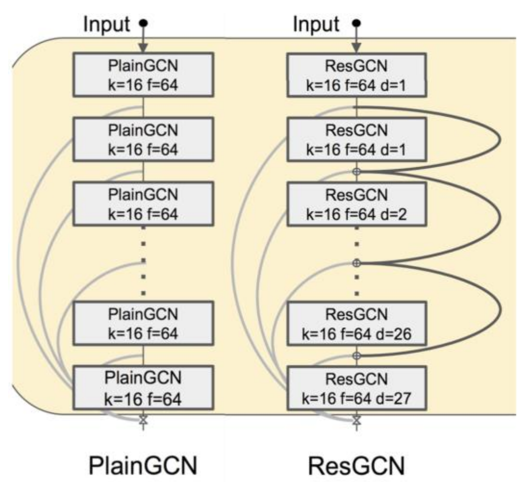


GNN Models based on Connection

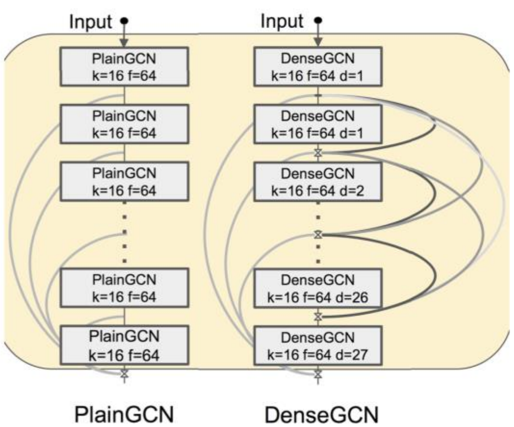
Deep Graph Convolutional Networks



DGCN: Residual Connections



DGCN: Dense Connections



参考文献：

Zhuan Zhou，Graph Convolutional Neural Networks：An introduction Tutorial ,2018

Jing Yu，Deep Learning on Graphs with Graph Convolutional Networks

Yue Zhang， Graph Neural Networks in NLP

在 此 表 示 感 谢!

谢谢各位！

Q&A