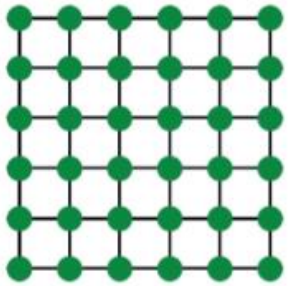
图神经网络 GNN

Graph Neural Network

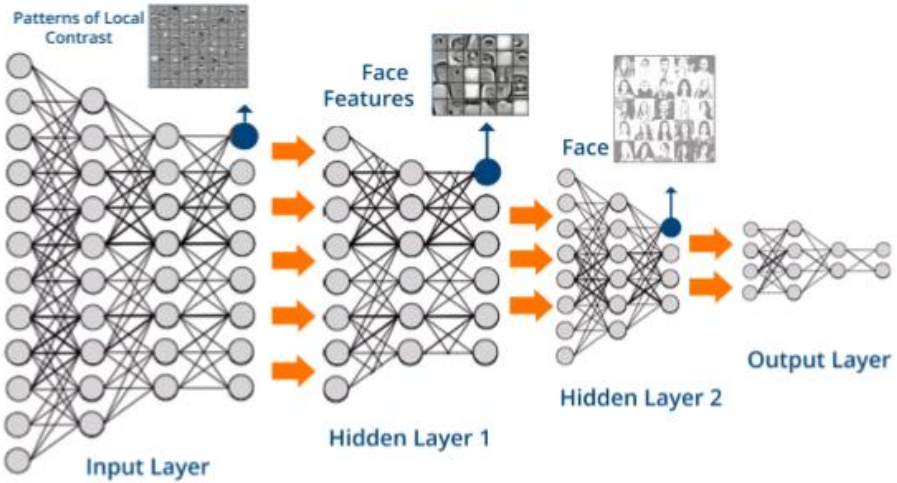
王博 天津大学智能与计算学部 2019.12

当前的两类主要深度学习模型

卷积模型

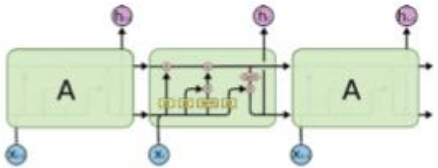


Images



序列模型

Text/Speech

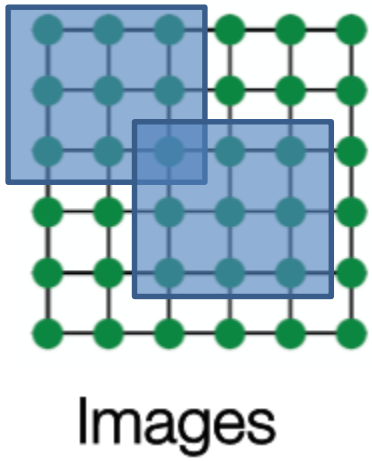


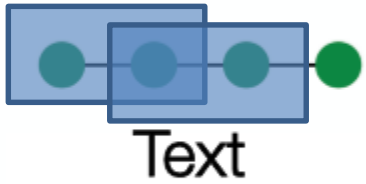
当前的两类主要深度学习模型

无论卷积还是序列模型，实际上都假定输入对象的结构是一个均匀的网络。

换言之，就是基本元素（像素、词汇）之间的关系结构是处处相同的。

均匀的结构使得同一个卷积核可以处理任何一个局部区域，并且参数共享



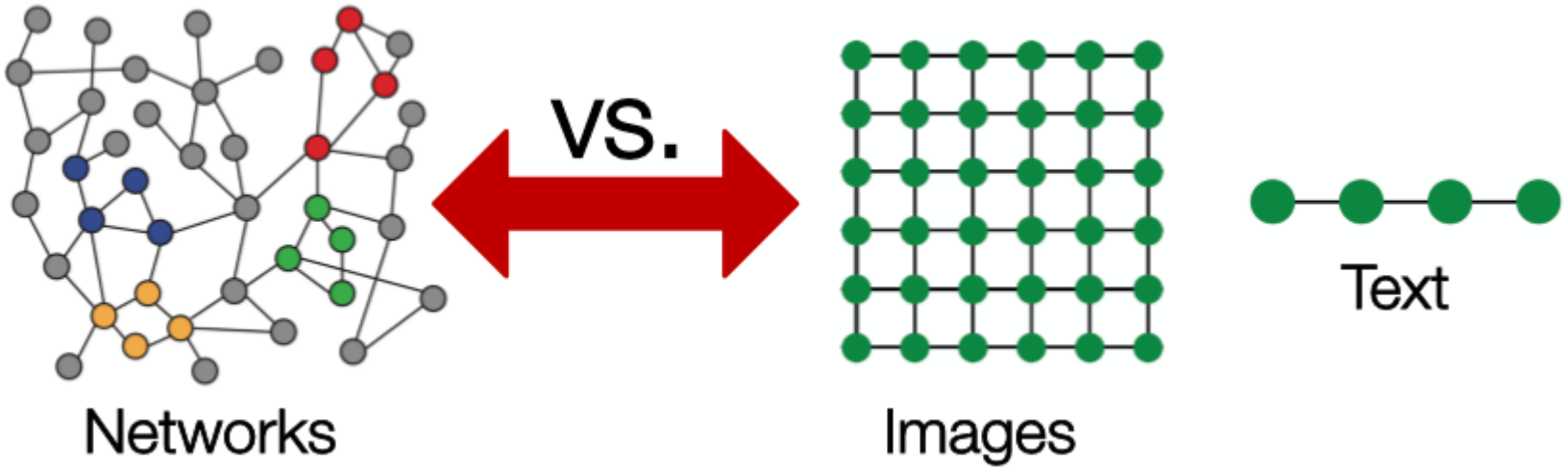


均匀的结构使得序列模型可以通过一遍简单的扫描，有序的处理全部输入信息

CNN与RNN无法处理一般网络结构

但是，现实中元素之间的结构并不总是均匀的。

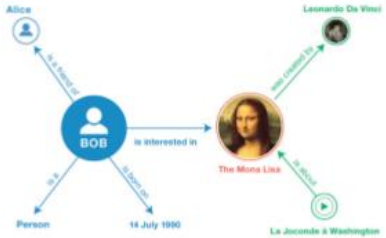
而任意图才是元素结构的一般化表示，网格与序列都只是一般图的特例



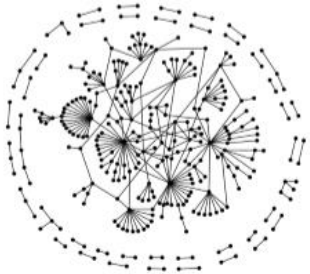
CNN与RNN无法处理一般网络结构

具有一般图结构的对象十分广泛，他们都无法用普通的CNN和RNN有效处理

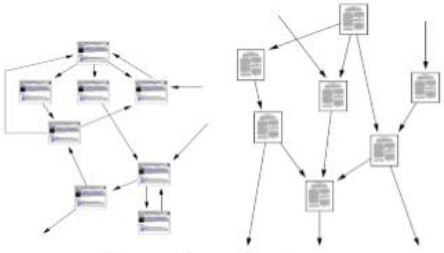
Social networks



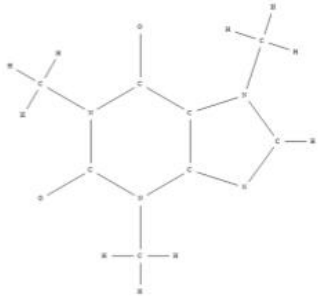
Knowledge graphs



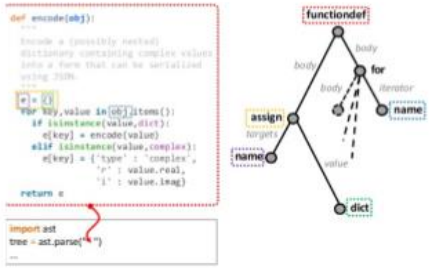
Biological networks



Complex Systems



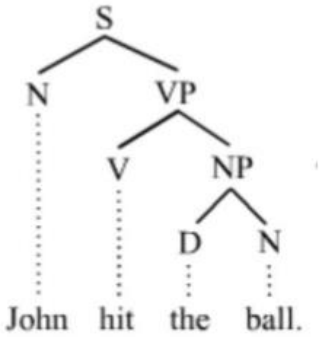
Molecules

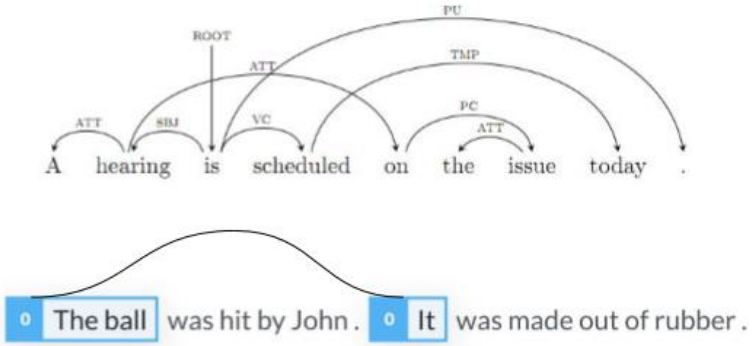


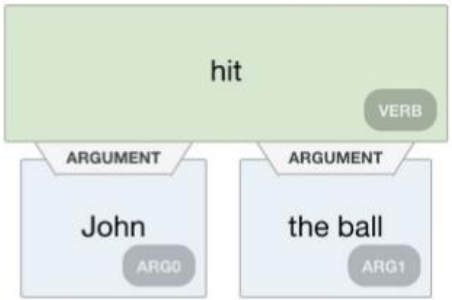
Code

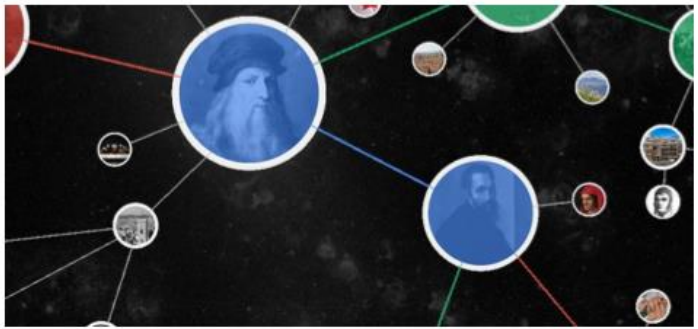
CNN与RNN无法处理一般网络结构

具有一般图结构的对象十分广泛，他们都无法用普通的CNN和RNN有效处理



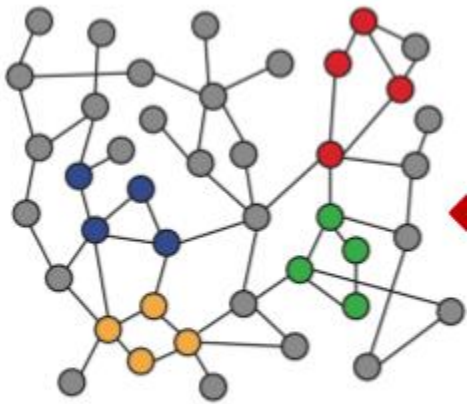






CNN与RNN无法处理一般网络结构

具有一般图结构的对象十分广泛，他们都无法用普通的CNN和RNN有效处理



试想：

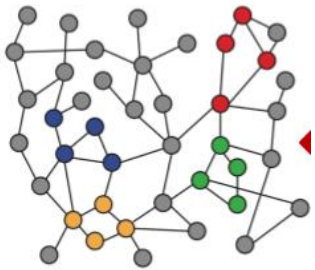
（1）若使用基于局部特征的方法来处理一般图，如何定义卷积核的尺寸和方法?

（2）若使用用序列的方法来处理一般图，如何给出序列的行走路线?

面向一般图的NN的基本目标

那么，当我们要用NN来处理一般图的时候，我们到底想做什么？

输入



Networks

传统机器学习方法：

选取相关特征，做分类（点、边、图分类）或相似度分析（边预测）

深度学习方法：

通过特定**神经网络结构**（如经典的CNN和RNN）形成蕴含了**图信息**的**表示**（NN的表示就相当于特征信息）。

图信息包括顶点的属性和拓扑结构

表示对于当前的NN来说是向量或张量表示

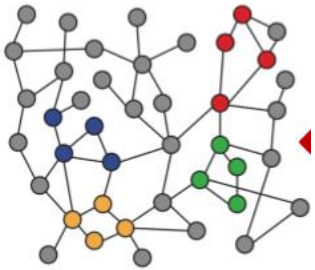
输出

- 顶点类别
- 图/子图的类别
- 边的存在与类别
- 图生成

一般图中的顶点表示

那么，图中的结点所蕴含的网络信息应该什么样呢?

输入



Networks

CNN：每个卷积的表示蕴含局部的关键特征信息

RNN：每个节点的表示蕴含前序序列的信息

Attention：对局部/前序中的信息有所侧重

GNN：每个顶点的表示要包含顶点属性和顶点拓扑结构

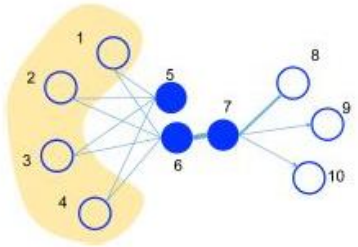
具体而言，GNN的顶点表示要满足：

一阶相似性：有边相连的节点，表示相似

二阶（高阶）相似性：领域相似的节点，表示相似

输出

- 顶点类别
- 图/子图的类别
- 边的存在与类别
- 图生成



SDNE (Structural deep network embedding)

同时优化一阶和二阶相似度：

每个结点用一个自编码器来重建领域信息，从而建模二阶相似度

节点之间使用拉普拉斯特征映射（反映节点之间的距离）来惩罚使得相邻节点距离较远的编码结果

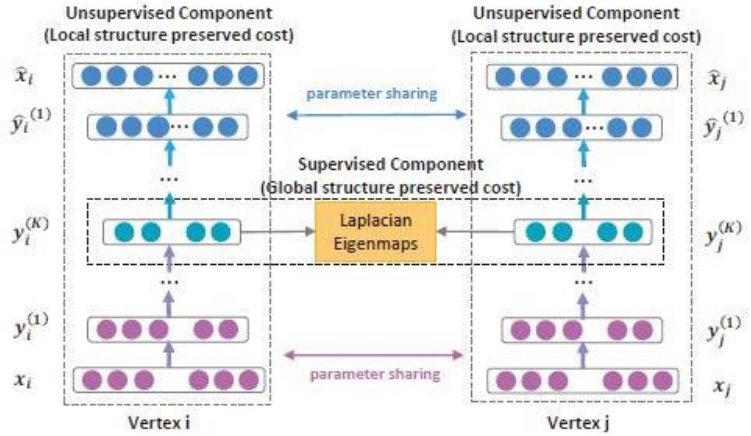
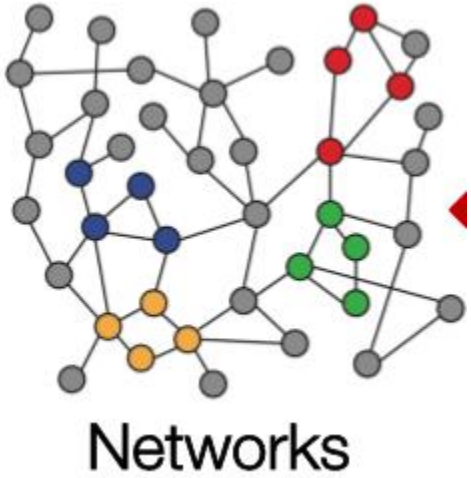


Figure 2: The framework of the semi-supervised deep model of *SDNE*

基于随机游走的方法



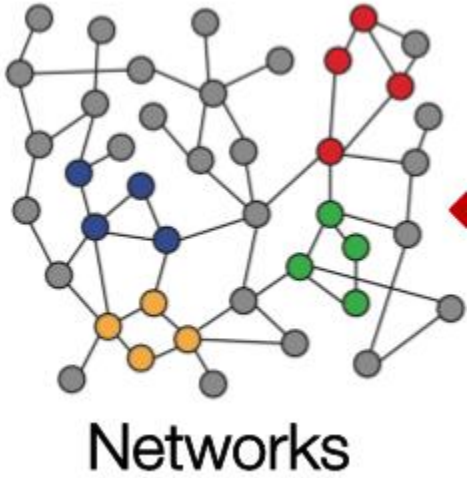
让我们进一步引入一些使GNN更有效的结构（如果用CNN或RNN来改进原始的全连接网络）

试想：

（1）若使用基于局部特征的方法来处理一般图，如何定义卷积核的尺寸和方法?

（2）若使用序列的方法来处理一般图，如何给出序列的行走路线?

基于随机游走的方法



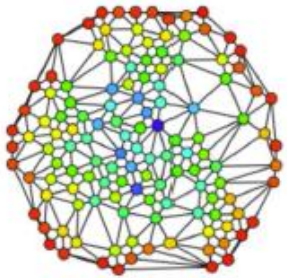
试想：

（2）若使用序列的方法来处理一般图，如何给出序列的行走路线?

通过随机游走在图中形成序列。

DeepWalk

- Generate γ random walks for each vertex
- Each random walk has length t
  - in each random walk step, jump to the next vertex uniformly.
- Example: $v_{46} \rightarrow v_{45} \rightarrow v_{71} \rightarrow v_{24} \rightarrow v_5 \rightarrow v_1 \rightarrow v_{17}$
- Finally, for vertex v_1 in a network, we have



$$
\begin{array}{cccccc}
v_{71} \rightarrow & v_{24} \rightarrow & v_5 \rightarrow & v_1 \rightarrow & v_{17} \rightarrow & v_{80} \rightarrow \\
v_{92} \rightarrow & v_2 \rightarrow & v_3 \rightarrow & v_1 \rightarrow & v_{12} \rightarrow & v_{73} \rightarrow \\
v_{37} \rightarrow & v_{34} \rightarrow & v_9 \rightarrow & v_1 \rightarrow & v_{10} \rightarrow & v_{94} \rightarrow \\
v_{73} \rightarrow & v_{64} \rightarrow & v_5 \rightarrow & v_1 \rightarrow & v_{12} \rightarrow & v_1 \rightarrow \\
v_{75} \rightarrow & v_{14} \rightarrow & v_6 \rightarrow & v_1 \rightarrow & v_{13} \rightarrow & v_{61} \rightarrow
\end{array}
$$

- 每个序列视为一个句子
- 每个顶点视为一个词汇
- 应用词向量技术构建顶点表示

Node2Vec

与与DeepWalk的最大区别在于，node2vec采用有偏随机游走，在广度优先（bfs）和深度优先（dfs）图搜索之间进行权衡，从而产生比DeepWalk更高质量和更多信息量的嵌入

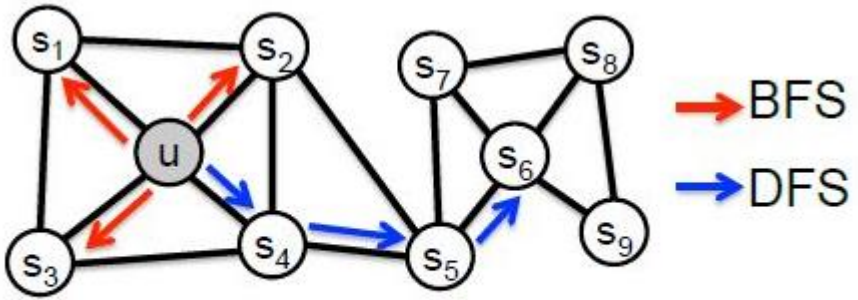


Figure 1: BFS and DFS search strategies from node u ($k = 3$).

Node2Vec

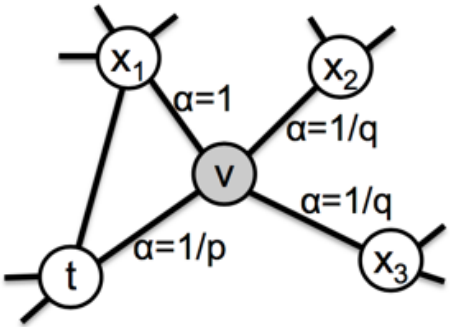
通过调整参数可以使得顶点的上下文在远距离邻居（DFS）和近距离邻居（BFS）之间调整

Return parameter p:

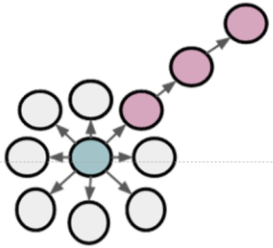
Return back to the previous node

In-out parameter q:

Moving outwards (DFS) vs. inwards (BFS)



BFS:
Micro-view of neighbourhood



DFS:
Macro-view of neighbourhood

Node2Vec

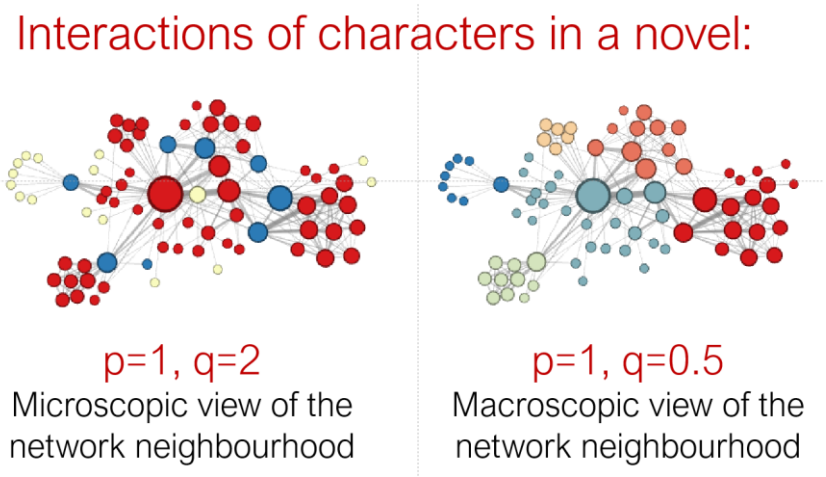
通过调整参数可以使得顶点的上下文在远距离邻居（DFS）和近距离邻居（BFS）之间调整

Return parameter p:

Return back to the previous node

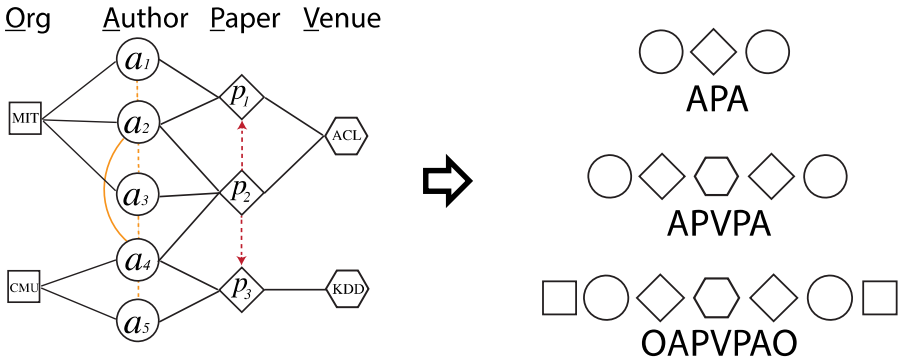
In-out parameter q:

Moving outwards (DFS) vs. inwards (BFS)



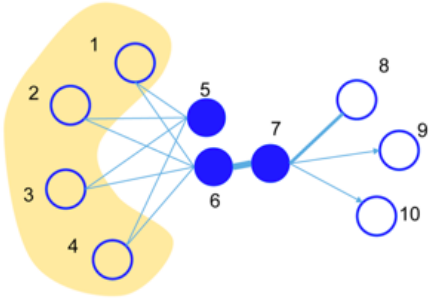
Metapath2vec：异质性网络中的顶点表示

随机路径必须符合预设的若干元路径（Metapath）

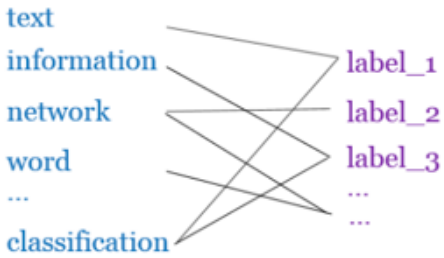


基于随机游走的方法

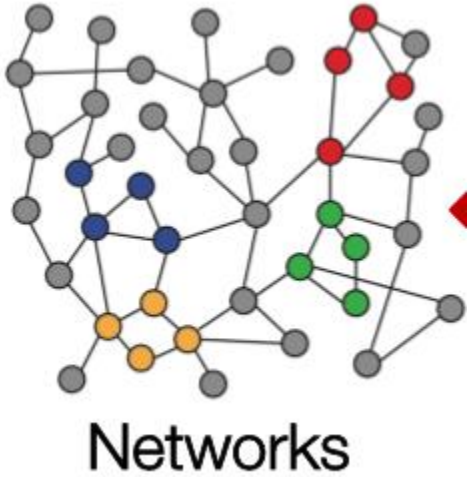
- **LINE:** explicitly preserves both first-order and second-order proximities.



- **PTE:** learn heterogeneous text network embedding via a semi-supervised manner.



图卷积神经网络GCN

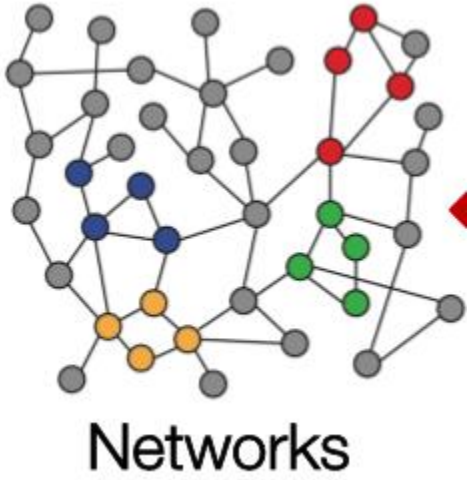


试想：

（1）若使用基于局部特征的方法来处理一般图，如何定义卷积核的尺寸和方法?

（2）若使用序列的方法来处理一般图，如何给出序列的行走路线?

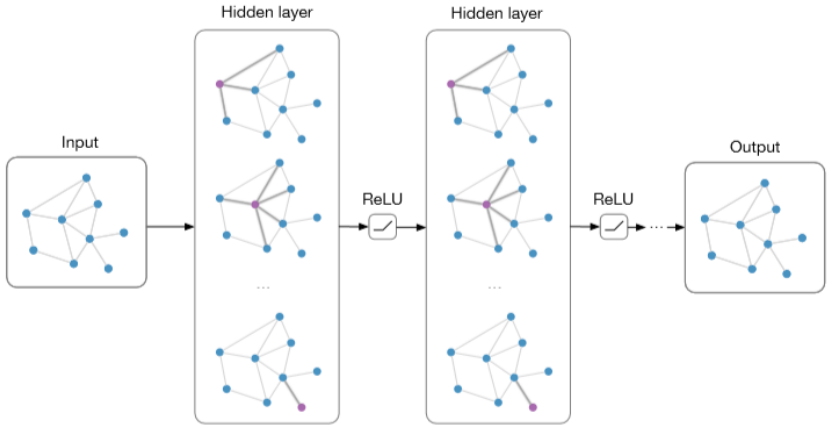
图卷积神经网络GCN



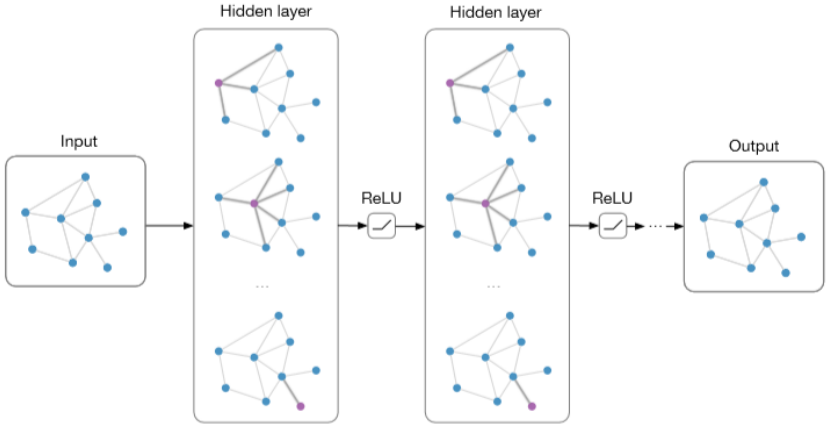
试想：

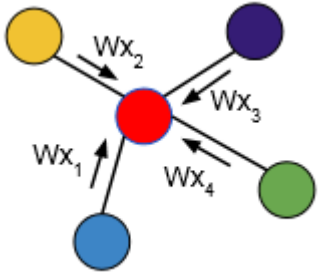
（1）若使用基于局部特征的方法来处理一般图，如何定义卷积核的尺寸和方法?

以每个节点为核心，将其邻域设为卷积范围，卷积方法是汇聚邻居节点的信息做为核心节点的表示。



图卷积神经网络GCN





$$h_v = f\left(\frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} W x_u + b\right), \quad \forall v \in \mathcal{V}.$$

Normalization

Neighborhood Aggregation

Bias (Model-parameter)

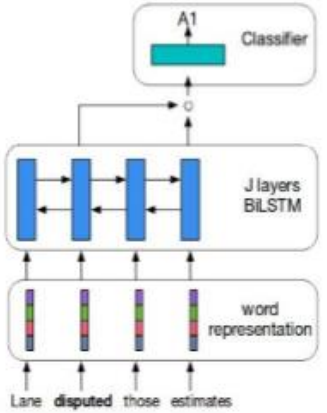
Non-Linearity

ReLU

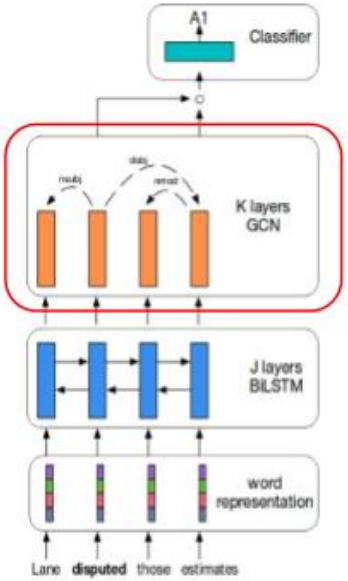
Filter matrix (Model-parameter)

Initial Node Features

基于GCN的语义角色标注



Standard Deep Learning Architecture
for NLP problems
(above is for Semantic-Role Labeling (SRL))



Model with GCN as part
of the network

GCN weights are
trained based on the
final objective

注意力图卷积神经网络GAT

基本的GCN中邻居节点的权重是平均的。

GAT中邻居节点的权重是可以训练的参数。

(a) Graph Convolution Networks [14] explicitly assign a non-parametric weight $a_{ij} = \frac{1}{\sqrt{deg(v_i)deg(v_j)}}$ to the neighbor v_j of v_i during the aggregation process.

(b) Graph Attention Networks [15] implicitly capture the weight a_{ij} via an end-to-end neural network architecture, so that more important nodes receive larger weights.

学习如何聚合邻居 GraphSage

GCN和GAT 是Transductive learning：训练语料包含待标注语料，标注在训练过程中完成。优点，质量高；缺点，扩展性差（标注新样本需要全局重新训练）

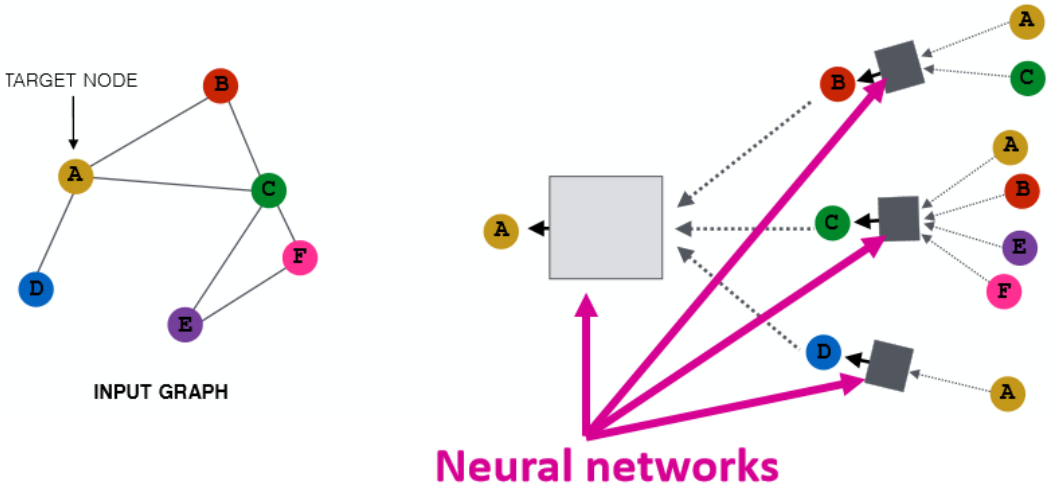
GCN和GAT的缺点：网络的任何变化都要重新进行全局训练 （类似word embedding）

GraphSage是Inductive learning：训练语料不包含待标注语料，先训练获得模型，然后泛化到测试语料上。

GraphSage学习一个由邻居节点形成中心节点表示的神经网络模型。

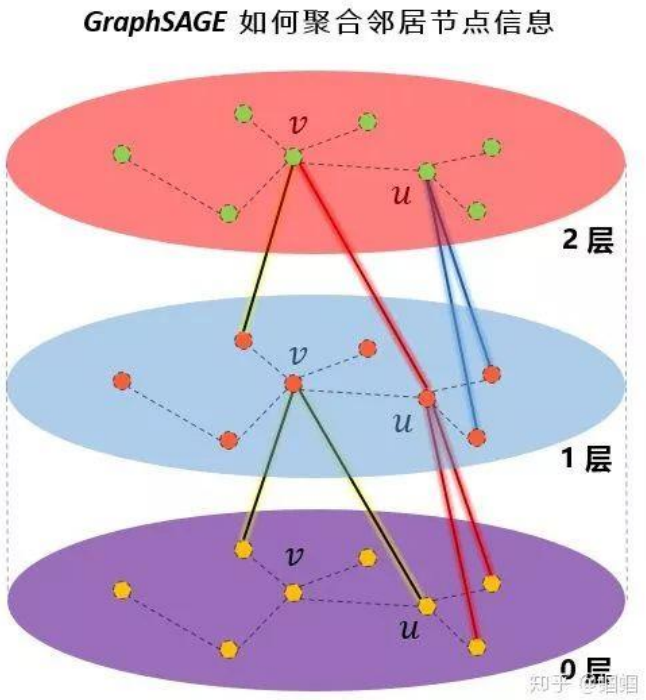
学习如何聚合邻居 GraphSage

GraphSage学习一个由邻居节点形成中心节点表示的神经网络模型（聚合函数）。



学习如何聚合邻居 GraphSage

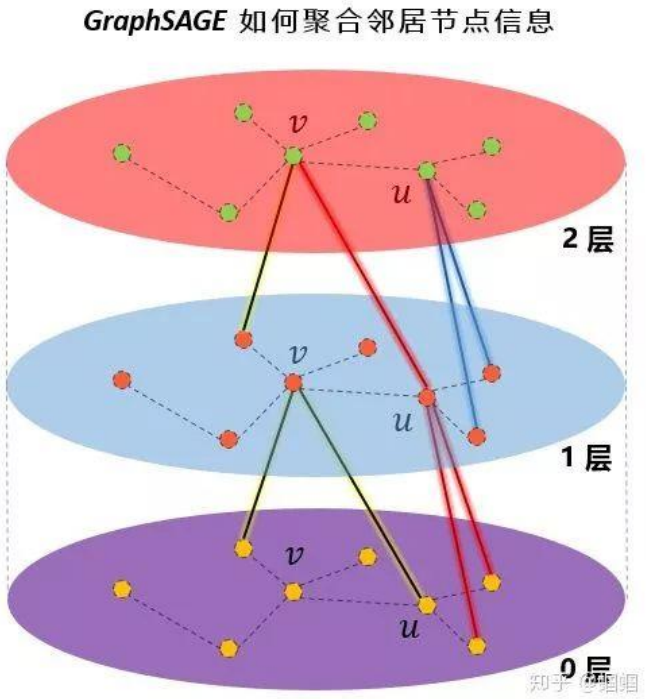
GraphSage学习一个由邻居节点形成中心节点表示的神经网络模型。



- GraphSage是分层的，类似神经网络的层次。每一层的节点表示由前一层的邻居节点通过聚合函数获得。
- 随着层次的推进，每个结点实际上不仅可以获得邻居结点的信息，还可以获得更远距离的结点的信息。

学习如何聚合邻居 GraphSage

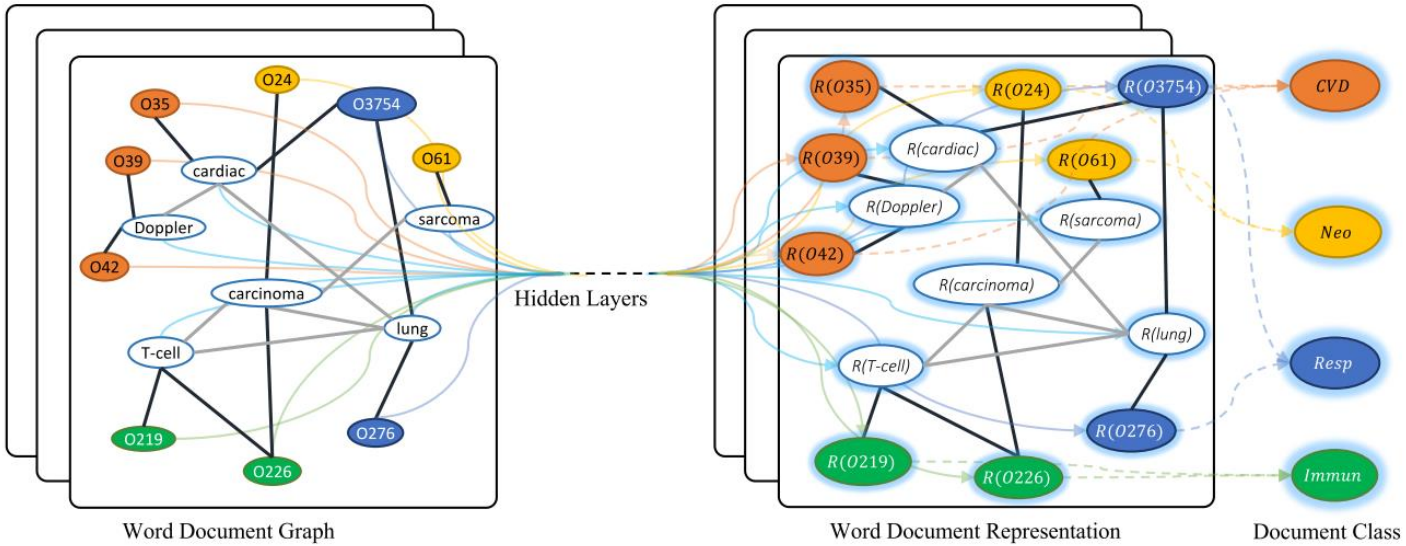
GraphSage学习一个由邻居节点形成中心节点表示的神经网络模型。



- GraphSage的参数学习需要设计一个损失函数。
- 损失函数可以是无监督的，也可以是有监督的。
- 对于无监督学习，损失函数应该让临近的节点的拥有相似的表示。

文本分类： Text-GCN 2019

以文档和词汇为结点构造异质性网络，训练获得文档的向量表示并分类到类别。



关系抽取： 2018

以依存句法树做为GCN的输入图，得到词汇的表示，进而分类词汇是否为关系标记词

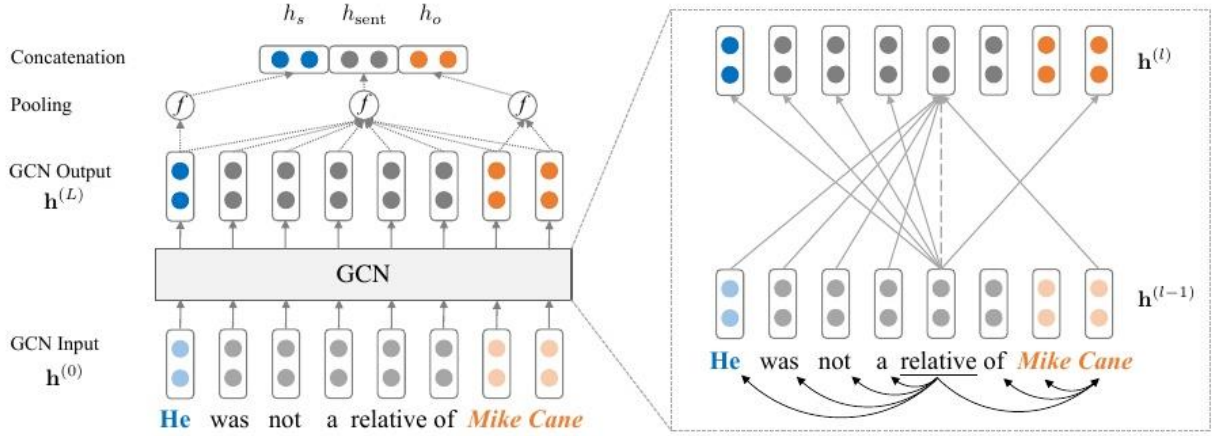


Figure 2: Relation extraction with a graph convolutional network. The left side shows the overall architecture, while on the right side, we only show the detailed graph convolution computation for the word "relative" for clarity. A full unlabeled dependency parse of the sentence is also provided for reference.

GNN的应用

个性化推荐： 2018

- 建立用户-用户-物品关系图。
- 在关系图上分别得到用户和物品的表示。
- 基于用户和物品的表示建立Rating预测模型

是否需要建立物品之间的关系？

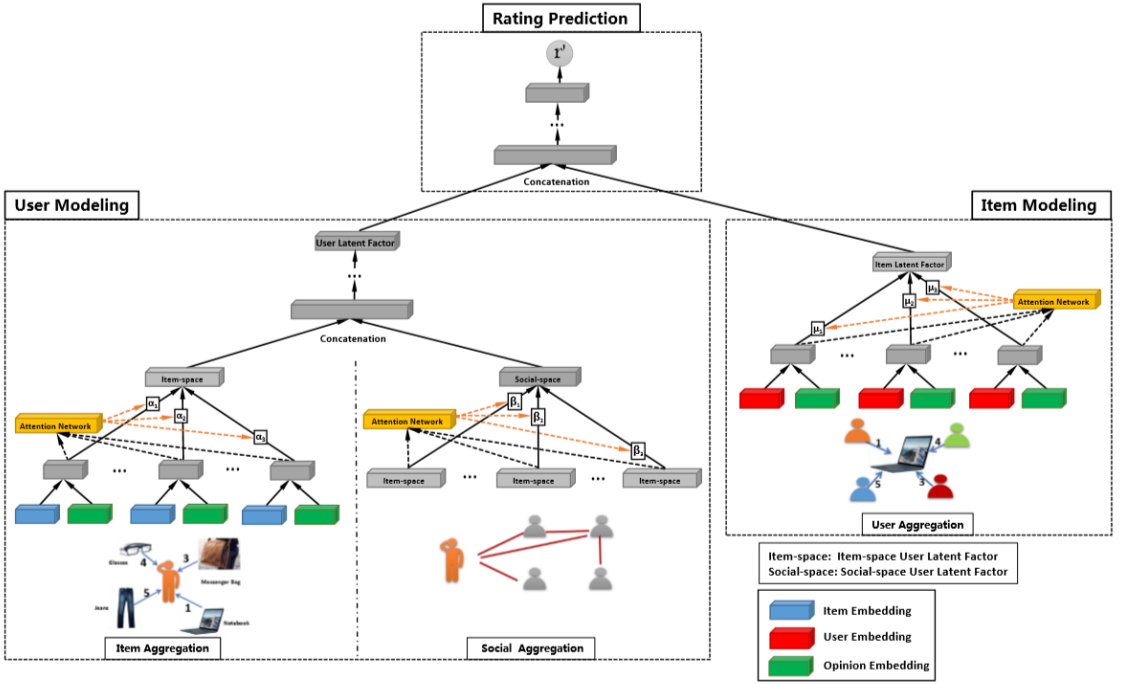


Figure 2: The overall architecture of the proposed model. It contains three major components: user modeling, item modeling, and rating prediction.

Thanks !