



图卷积神经网络



覃雄派

提纲

- 深度学习之图像的卷积及其作用
- 图数据及其分析需求
- 深度神经网络用于图数据的处理
- 图卷积神经网络、实践

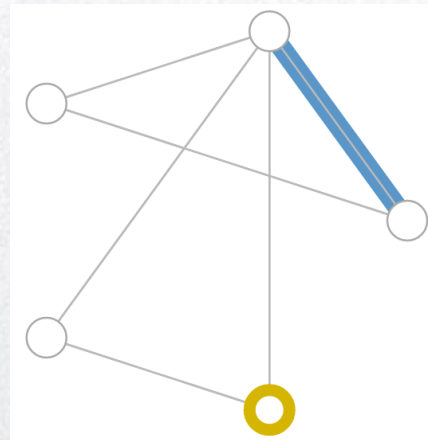
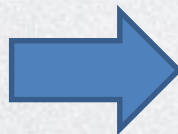
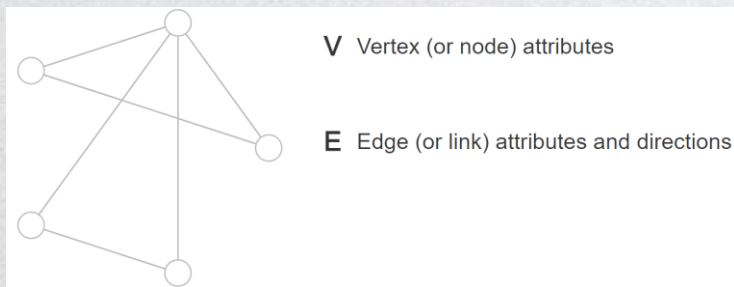


图卷积神经网络

图卷积神经网络

- 图数据回顾

没有权重

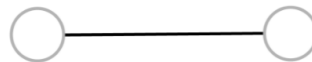


有权重



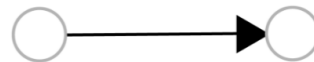
无向

Undirected edge



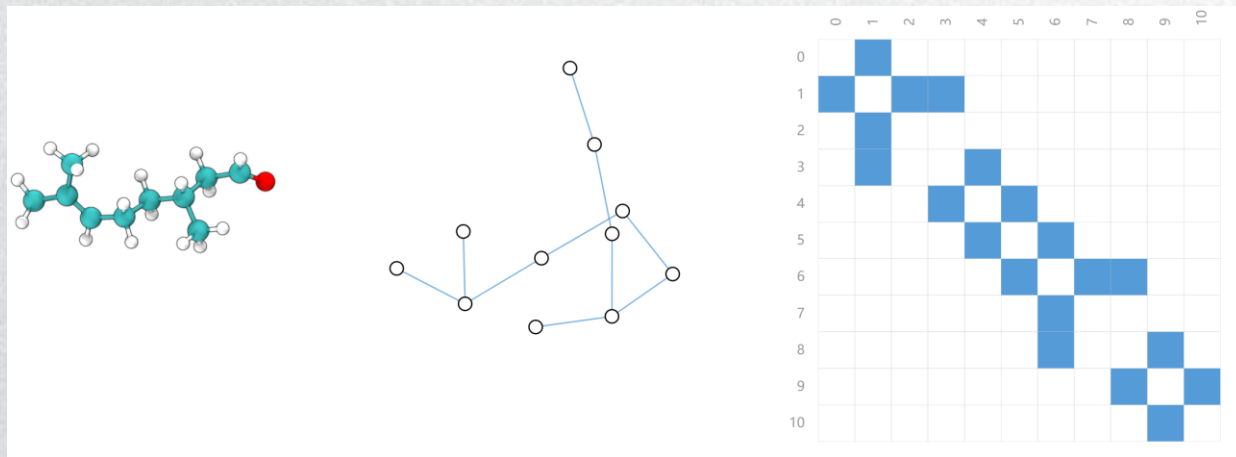
有向

Directed edge



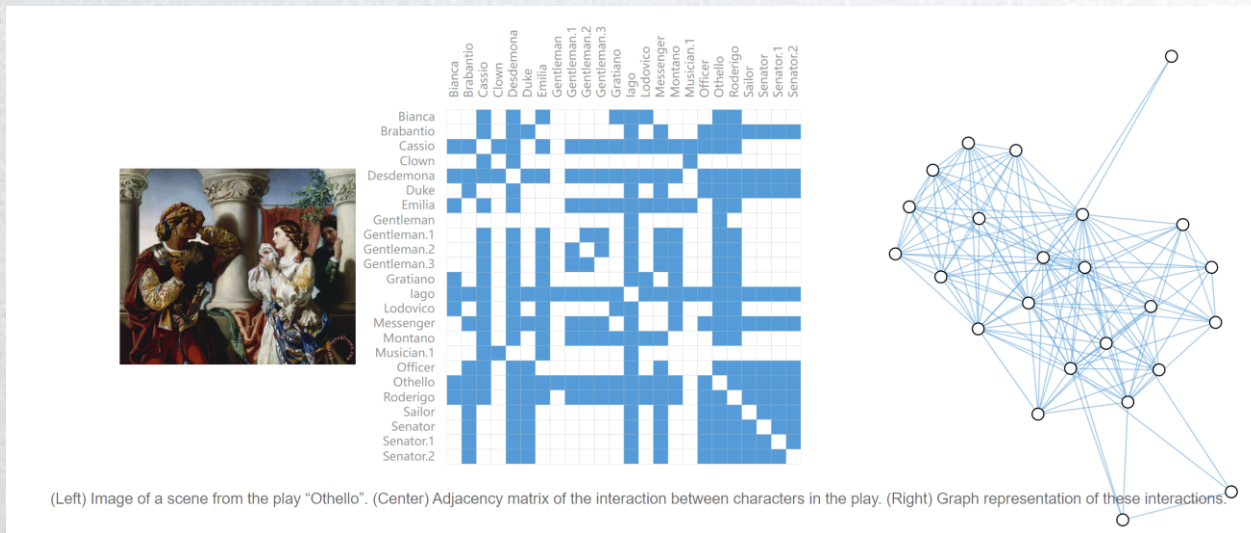
图卷积神经网络

- 图数据回顾
 - 分子结构→表示为图



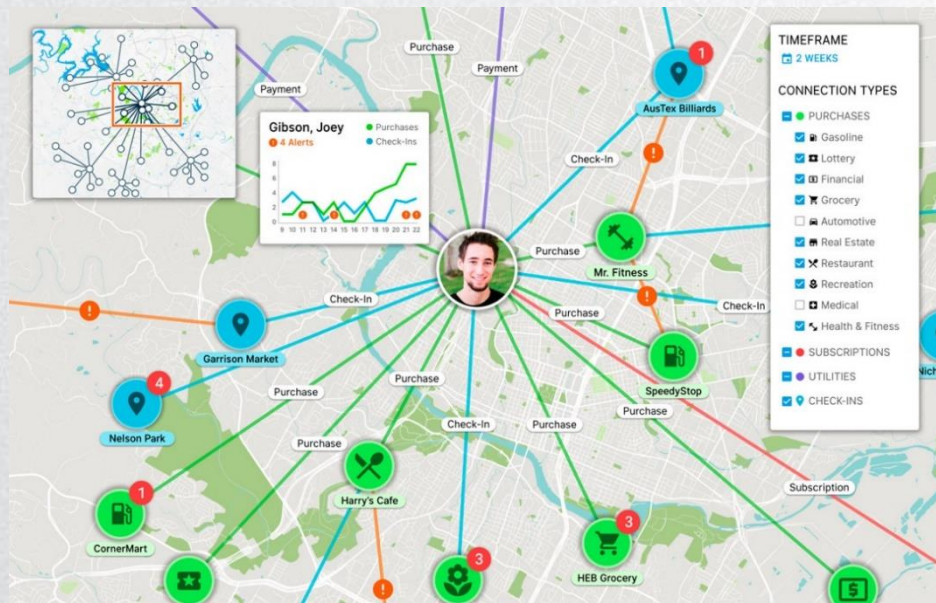
图卷积神经网络

- 图数据回顾
 - 戏剧人物关系→表示为图



图卷积神经网络

- 图数据回顾
 - 更多的应用场景里的实体及其关系→表示为图

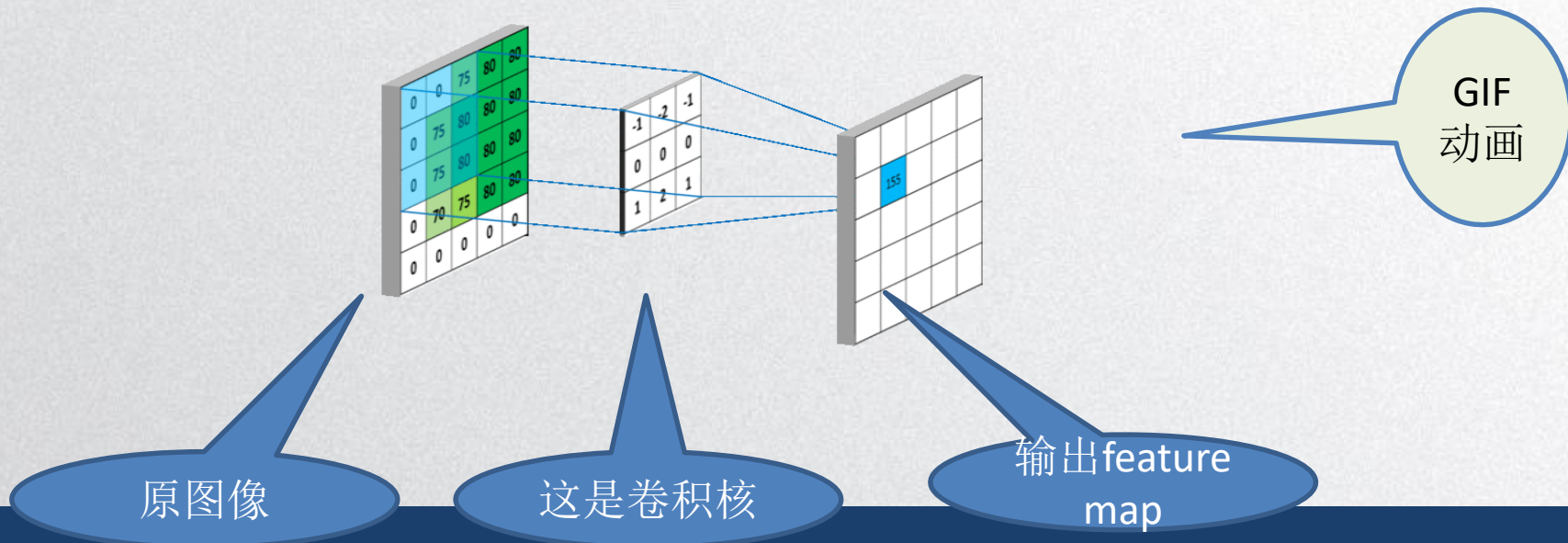


图卷积神经网络



图卷积神经网络

- 深度学习之图像的卷积及其作用
 - 目标图像的某个像素
 - 为原图像该像素，及其周围其他像素的加权和
 - 权重矩阵称为卷积核



图卷积神经网络

- 深度学习之图像的卷积及其作用
 - 通过一个实例看卷积核的作用
 - 卷积核是训练出来的
 - 不同的卷积核识别图像的不同特征

10	10	10	10	0	0	0	0
10	10	10	10	0	0	0	0
10	10	10	10	0	0	0	0
10	10	10	10	0	0	0	0
10	10	10	10	0	0	0	0
10	10	10	10	0	0	0	0
10	10	10	10	0	0	0	0
10	10	10	10	0	0	0	0

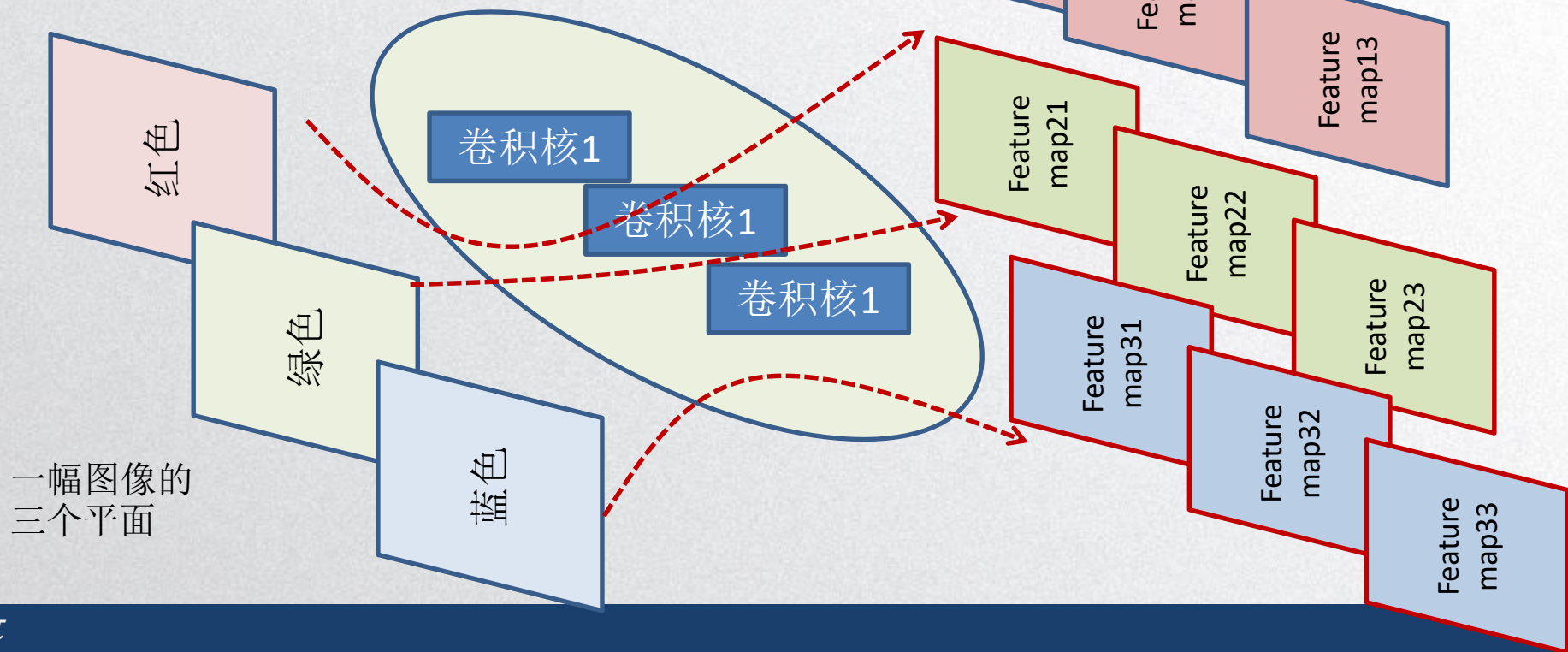
1	0	-1
1	0	-1
1	0	-1

0	0	10	10	0	0
0	0	10	10	0	0
0	0	10	10	0	0
0	0	10	10	0	0
0	0	10	10	0	0
0	0	10	10	0	0

这个卷积核识别纵向的
边缘线

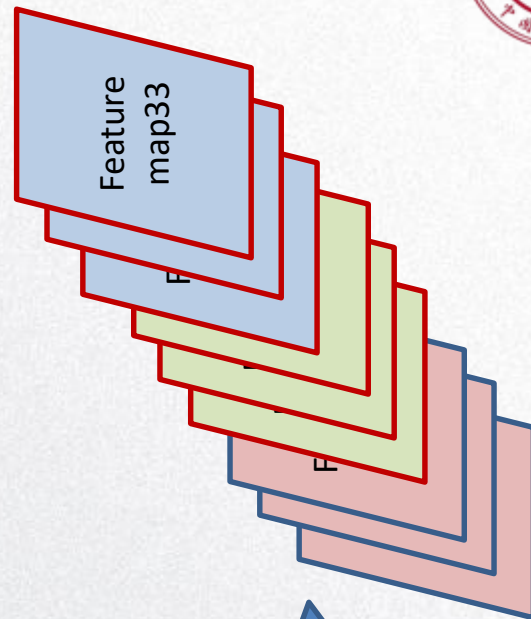
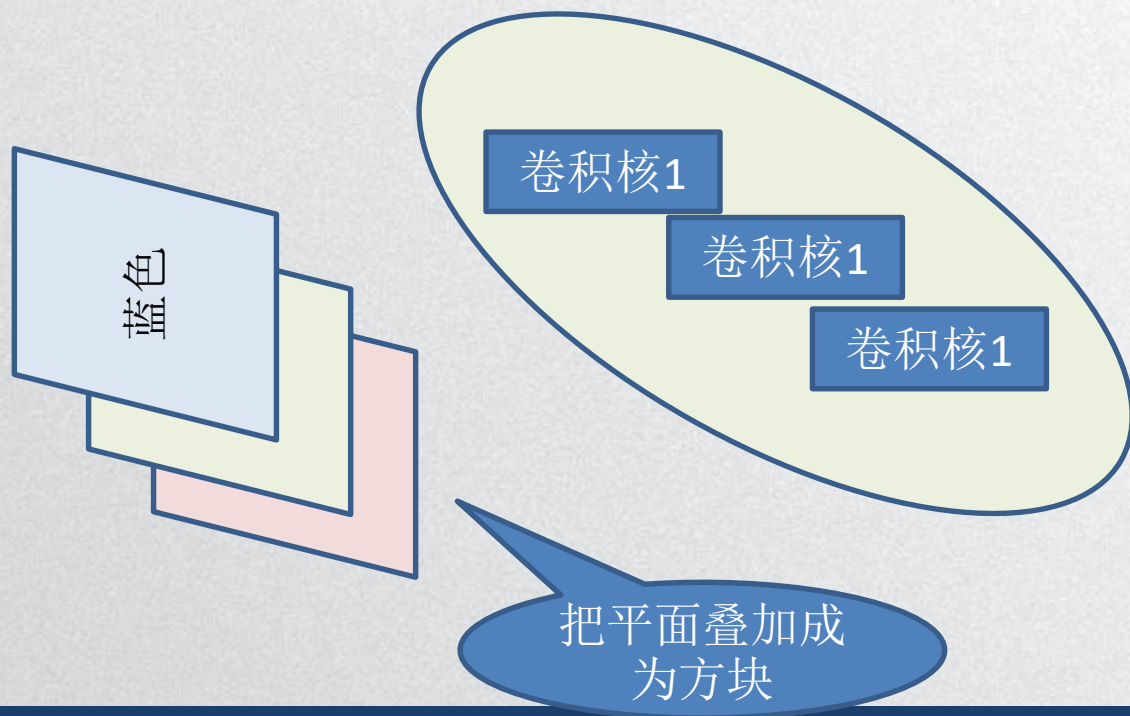
图卷积神经网络

- 深度学习之图像的卷积及其作用
 - 用多个卷积核进行卷积



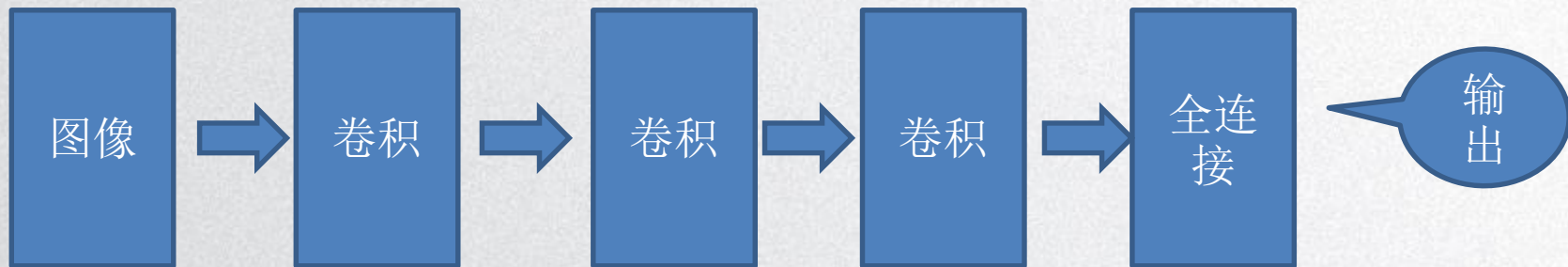
图卷积神经网络

- 深度学习之图像的卷积及其作用
 - 用多个卷积核进行卷积，平面的叠加



图卷积神经网络

- 深度学习之图像的卷积及其作用：构造全连接神经网络
 - 可以通过一系列卷积层
 - 加上最后一个全连接层
 - 构造神经网络，完成预测功能（比如分类）



- 卷积神经网络在图像处理、音频、以及文本处理中，已经取得了巨大成功
- 图像是一种二维的矩阵，CNN的核心能力在于使用小窗口的过滤器，在图片上平移并且对准，通过卷积的方式来提取特征



图卷积神经网络

- **卷积神经网络(CNN, Convolution Neural Network)**
 - 卷积神经网络是一种特殊类型的前向反馈神经网络，特别**适合于图像识别、语音分析等应用领域**
 - 卷积神经网络由于一个映射面上的神经元**共享权值**，因而减少了网络自由参数的个数，降低了网络参数选择的复杂度
 - 该优点在网络的输入是图像时表现的更为明显，可以使用**图像直接作为网络的输入**，避免了传统识别算法中复杂的特征提取过程
 - 并且，在一个映射面生共享权值，使得图像的特征被检测出来，而**不管它的位置(Location)是否发生移动**

图卷积神经网络

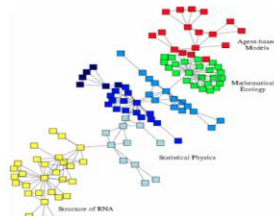


图卷积神经网络

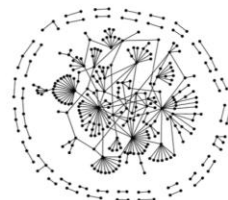
- 图数据及其分析需求
 - 图数据无处不在



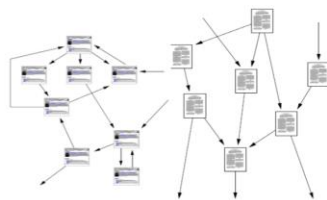
Social networks



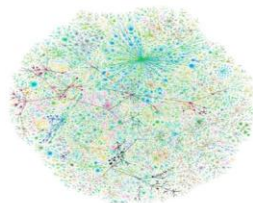
Economic networks



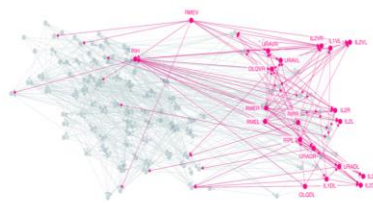
Biomedical networks



Information networks:
Web & citations



Internet



Networks of neurons

example from Jure's slides

图卷积神经网络

- 图数据及其分析需求
 - 应用场景：节点分类
 - 作者、论文、发表会议的领域分类

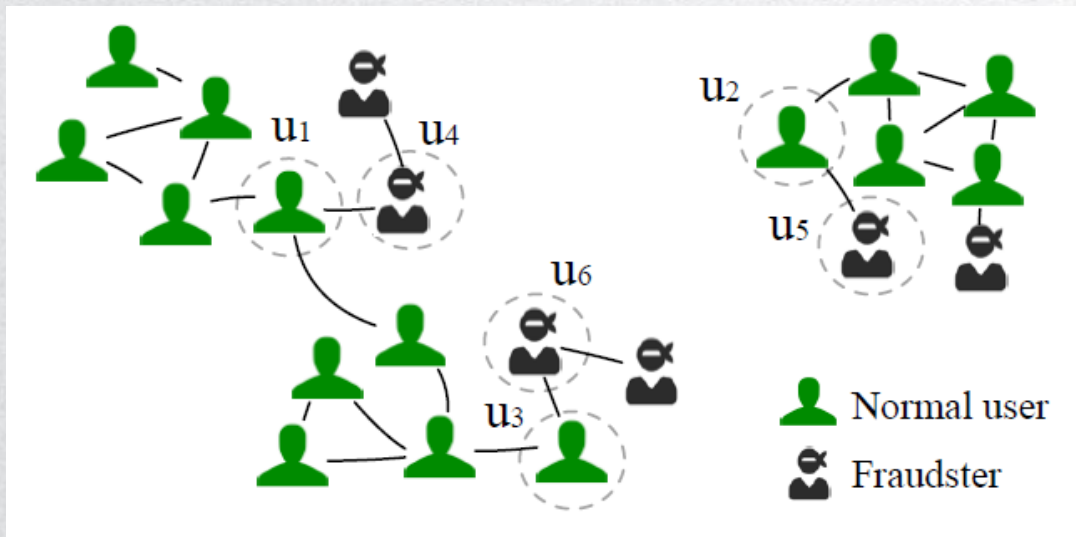


图卷积神经网络

- 图数据及其分析需求

- 应用场景：异常检测（节点分类）

- 捕捉节点与整体之间的相似关系，检测不一样的节点

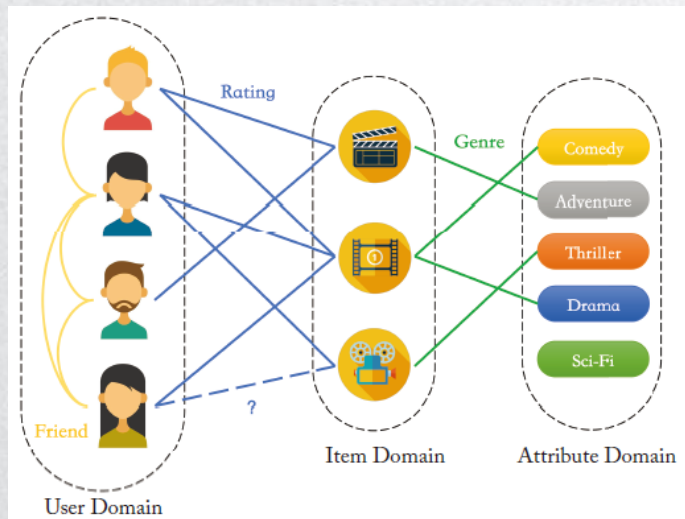


图卷积神经网络

- 图数据及其分析需求

- 应用场景：推荐系统（链接预测）

- 基于用户和商品的历史交互行为，给用户推荐新商品

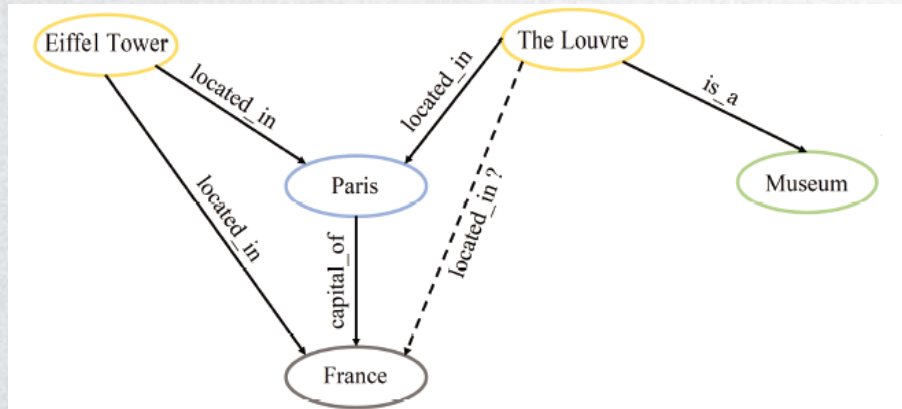


图卷积神经网络

- 图数据及其分析需求

- 应用场景：知识图谱补全（链接预测）

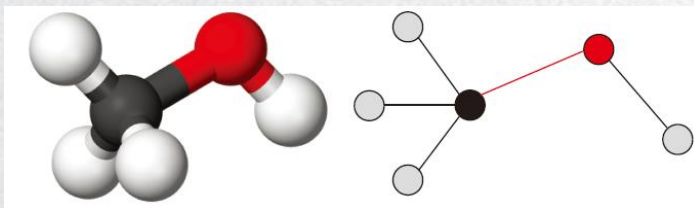
- 利用邻居节点与关系信息对节点进行表示，再对未知关系进行预测



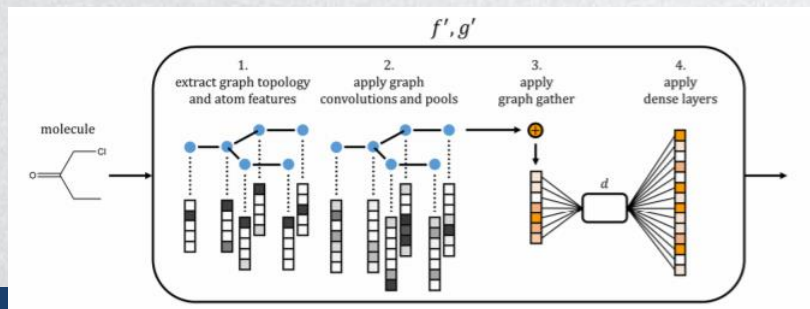
- 知识图谱的例子，其中节点代表实体，边代表关系
 - 虚线部分代表需要补全的关系

图卷积神经网络

- 图数据及其分析需求
 - 应用场景：药物设计（全图分类）
 - 提取分子的特征，用于药物性质预测等



分子可表示成图结构



图卷积神经网络





图卷积神经网络

- 深度神经网络在文本、音频、图像等数据的处理中，表现出强大的处理能力
 - 文本、音频、图像、视频等媒体完成数字化以后，都可以表达成向量或者张量的形式，有利于深度神经网络的后续处理
 - 人们就思考，能否借助深度神经网络，对图数据(社交网络)进行处理呢？
- 一个新的研究热点—图神经网络(Graph Neural Networks, GNN)应运而生
 - GNN具备对图中的节点间关系进行建模的强大功能
 - 近年来，GNN在社交网络(狭义)、知识图谱、推荐系统等领域，得到了越来越广泛的应用



图卷积神经网络

- 深度神经网络用于图数据的处理
 - 深度神经网络的能力：归纳偏置
- 归纳偏置在深度学习中无处不在
 - 卷积神经网络CNN
 - 空间相近的神经元有强相关性
 - 循环神经网络RNN
 - 时间相近的神经元有强相关性
 - 图神经网络GNN
 - 有边相连的节点有强相关性

从具体例子中寻找共性，泛化为通用的模型偏好



图卷积神经网络

- 图神经网络
- 是直接进行图数据处理的神经网络，包括
 - 图卷积神经网络(Graph Convolution Network)
 - 图注意力神经网络(Graph Attention Network)
 - 图自编码器(Graph Auto encoders)
 - 图生成式神经网络(Graph Generative Network)
 - 以及图时空神经网络(Graph Spatial-Temporal Network)等



图卷积神经网络



- GNN的应用
 - **Node classification**: for example predict the labels of nodes by considering the labels of their neighbors
 - **Link prediction**: from example prediction connections for social networks
 - **Graph clustering**: This involves dividing the nodes of a graph into clusters.
 - **Graph classification**: This entails classifying a graph into a category.
 - **Computer vision**: for example The **scene** generation model then identifies **objects** in the image and the **semantic relationship** between them

图卷积神经网络

- 图神经网络，与图嵌入(也称为网络嵌入)研究密切相关
 - 图嵌入，旨在通过保留图的网络拓扑结构和节点内容信息，将图中的顶点表示为低维稠密向量，以便使用传统的机器学习算法(比如支持向量机分类器等)、以及深度学习算法，对其进行处理
 - 图嵌入算法主要是一些无监督方法，可以分为三类
 - 包括矩阵分解
 - 随机游走
 - 和深度学习方法等

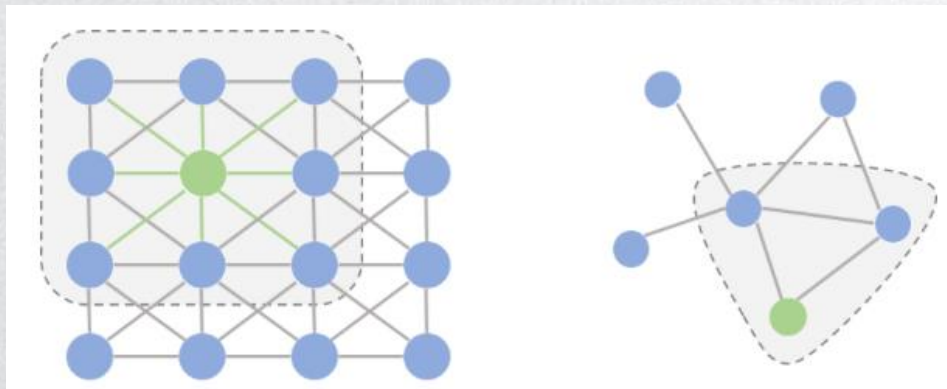
GCN兼具图数据的直接处理和图嵌入的功能

图卷积神经网络



图卷积神经网络

- 图卷积神经网络、实践：从CNN到Graph CNN
 - Graph convolution neural network 与传统 convolution neural network的区别
 - 在图像为代表的欧式空间中，结点的邻居数量都是固定的；卷积操作是用**固定大小可学习的卷积核**来抽取像素的特征
 - 图数据卷积的难点：**邻居结点数量不固定**；解决思路：找出一种**可处理变长邻居结点**的卷积核在图上抽取特征

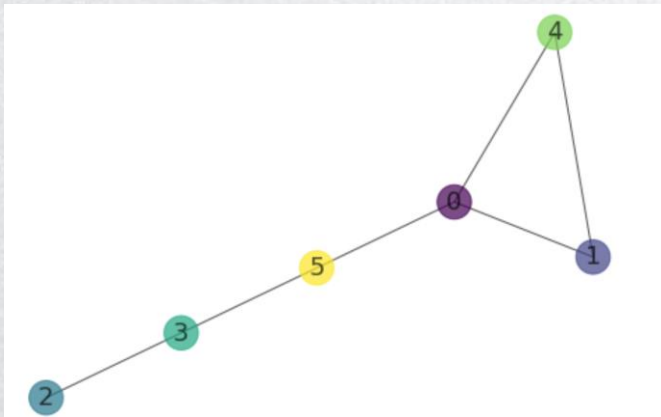


图卷积神经网络

- 图卷积神经网络、实践：CNN→GCN
 - GCN是受到CNN启发而提出的一种神经网络结构
 - 它的目的和CNN网络是类似的，即对图数据进行特征提取
 - 这些提取的特征，可以用于下游任务
 - 比如节点分类(Node Classification)
 - 图分类(Graph Classification)
 - 链路预测(Link Prediction)等
 - 也可以得到图的嵌入表示(Graph Embedding)
 - 然后用传统机器学习方法实现后续任务

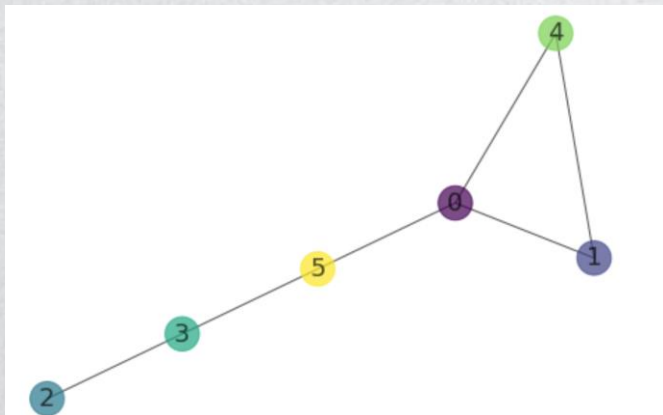
图卷积神经网络

- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - 每个节点有自己的特征(D 维)
 - 这些节点的特征构成 $N \times D$ 维的矩阵, 称为 X
 - 节点间的关系
 - 形成一个 $N \times N$ 的邻接矩阵 A (Adjacency Matrix)
 - X 和 A 是GCN的输入



图卷积神经网络

- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - 单层的图神经网络, 其计算过程表示为: $f(H^l) = \delta (H^l W^l)$, 其中 $H^0=X$, σ 是非线性激活函数
 - 而GCN的计算过程表示为: $f(H^l, A) = \delta (A H^l W^l)$, 多了一个 A 邻接矩阵
 - 下面通过剖析实例, 以获得直观的认识

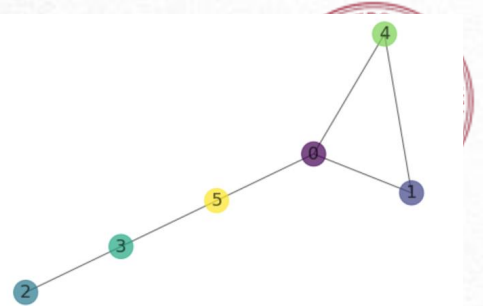


```
A=nx.adjacency_matrix(g).todense()
```

A

```
matrix([[0, 1, 0, 0, 1, 1],  
        [1, 0, 0, 0, 1, 0],  
        [0, 0, 0, 1, 0, 0],  
        [0, 0, 1, 0, 0, 1],  
        [1, 1, 0, 0, 0, 0],  
        [1, 0, 0, 1, 0, 0]], dtype=int32)
```

图卷积神经网络



- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - GCN的计算过程表示为: $f(H^l, A) = \delta(AH^lW^l)$, 多了一个 A 邻接矩阵
 - 下面通过剖析实例, 以获得直观的认识
 - 假设 X 为图中各个节点的属性

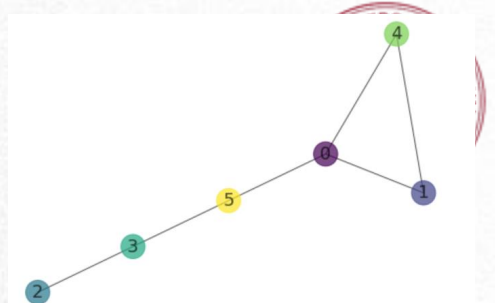
- X : $X \in R^{n \times d}$ 特征矩阵, 每一列为定义在所有节点上一个信号
- 这里每个节点用一个2维行向量表示

```
import numpy as np

X = np.matrix([
    [i, -i]
    for i in range(A.shape[0])
], dtype=float)
```

```
X
matrix([[ 0.,  0.],
        [ 1., -1.],
        [ 2., -2.],
        [ 3., -3.],
        [ 4., -4.],
        [ 5., -5.]])
```


图卷积神经网络



- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - GCN的计算过程表示为: $f(H^l, A) = \delta(AH^lW^l)$, 多了一个 A 邻接矩阵
 - 下面通过剖析实例, 以获得直观的认识
 - 假设 X 为图中各个节点的属性
 - H^l 是每一层的特征, 对于输入层的话, H^0 就是 X , 即 H^0 为 X , 有 AX

```
[[0, 1, 0, 0, 1, 1],  
 [1, 0, 0, 0, 1, 0],  
 [0, 0, 0, 1, 0, 0],  
 [0, 0, 1, 0, 0, 1],  
 [1, 1, 0, 0, 0, 0],  
 [1, 0, 0, 1, 0, 0]]
```

```
[[ 0.,  0.],  
 [ 1., -1.],  
 [ 2., -2.],  
 [ 3., -3.],  
 [ 4., -4.],  
 [ 5., -5.]]
```

$A * X$

```
matrix([[ 10., -10.],  
        [  4.,  -4.],  
        [  3.,  -3.],  
        [  7.,  -7.],  
        [  1.,  -1.],  
        [  3.,  -3.]])
```

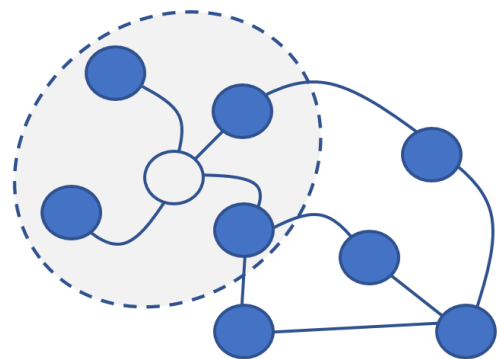
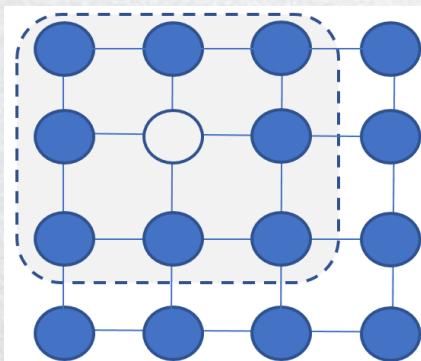
- 由此可以看出, AX 实际上是对邻接节点进行Sum操作
- 这种操作对特征的提取, 是对节点的度的倾向性进行聚合, 也就是节点的度越大, 聚合的信息就越大

$A * X$ 表示, 最简单的无参卷积方式, 将所有邻居节点的隐藏状态加和, 来更新当前节点的隐藏状态

图卷积神经网络

- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - A 为邻接矩阵, X 为节点属性
 - AX 的解释如下

- 在2维图像的CNN操作中, 我们可以把一个像素看作一个图的一个节点, 它的邻居是根据过滤器的窗口大小确定的
- 以浅灰色节点作为基准点进行2维卷积(3×3 卷积核), 是对邻居节点进行加权求和
- 所有的邻居节点, 因为是固定大小窗口内的像素, 可以认为(邻居)节点的数量是固定的, 节点是有序的



二维图像上的邻域和网络结构里的邻域

图卷积神经网络

- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - A 为邻接矩阵, X 为节点属性
 - AX 的问题**
 - A 中不包含Self-Loop信息(自己到自己的边)**
 - 上述节点的聚合表示, 不包括它自己的特征, 只是邻居节点特征的聚合
 - 但是具有自循环的节点, 应该在聚合中包含自己的特征

我们设计一个 $\hat{A}=A+I$, I 为单位矩阵, 来解决这个问题

```
I=np.matrix(np.eye(A.shape[0])) #单位矩阵
```

```
A_hat=A+I
```

```
A_hat
```

```
matrix([[1., 1., 0., 0., 1., 1.],  
        [1., 1., 0., 0., 1., 0.],  
        [0., 0., 1., 1., 0., 0.],  
        [0., 0., 1., 1., 0., 1.],  
        [1., 1., 0., 0., 1., 0.],  
        [1., 0., 0., 1., 0., 1.]])
```

图卷积神经网络

- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - A 为邻接矩阵, X 为节点属性; 现在 A 改为 \hat{A}
 - 计算过程表示为 $f(H^l, A) = \delta(\hat{A}H^lW^l)$
 - 公式还有一个问题

A_hat

```
matrix([[1., 1., 0., 0., 1., 1.],
        [1., 1., 0., 0., 1., 0.],
        [0., 0., 1., 1., 0., 0.],
        [0., 0., 1., 1., 0., 1.],
        [1., 1., 0., 0., 1., 0.],
        [1., 0., 0., 1., 0., 1.]])
```



度矩阵的计算公式为 $D_{ii} = \sum_j \hat{A}_{ij}$

- 度数大的节点在其特征表征中将具有较大的值, 度数小的节点将具有较小的值, 这可能会导致梯度消失或梯度爆炸, 会影响随机梯度下降算法的效果
- 随机梯度下降算法通常用于训练此类网络, 对每个输入特征的尺度(或值域范围)敏感
- 我们通过对特征表征进行归一化处理来解决这个问题; 通过将邻接矩阵 A 与度矩阵 D 的逆相乘, 对其进行变换, 从而通过节点的度对特征表征进行归一化

```
D=np.matrix(np.diag(np.array(A_hat.sum(axis=0))[0]))
D
```

```
matrix([[4., 0., 0., 0., 0., 0.],
        [0., 3., 0., 0., 0., 0.],
        [0., 0., 2., 0., 0., 0.],
        [0., 0., 0., 3., 0., 0.],
        [0., 0., 0., 0., 3., 0.],
        [0., 0., 0., 0., 0., 3.]])
```


图卷积神经网络

- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - A 为邻接矩阵, X 为节点属性; 现在 A 改为 \hat{A}
 - 这时候, 计算过程表示为 $f(H^l, A) = \delta (D^{-1}\hat{A}H^lW^l)$

计算 $D^{-1}\hat{A}$ 使得
矩阵的每一行是归一化的
结果, 即每一行的和是1

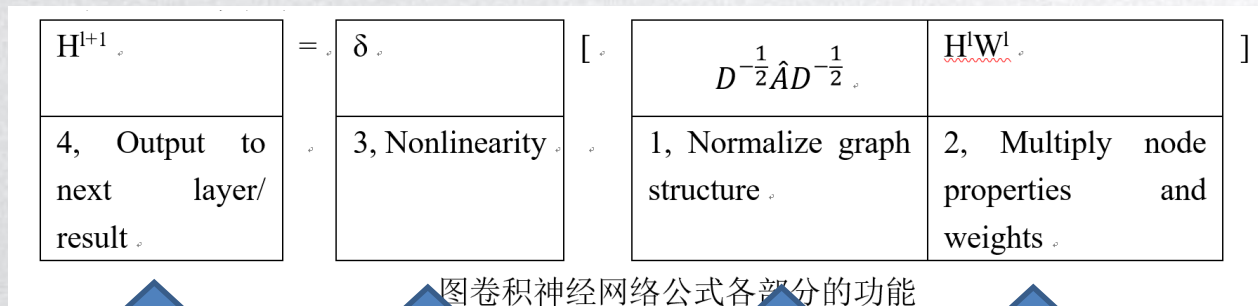
$D^{-1} \times A_hat$

```
matrix([[0.25      , 0.25      , 0.          , 0.          , 0.25      ,
          0.25      ],
         [0.33333333, 0.33333333, 0.          , 0.          , 0.33333333,
          0.          ],
         [0.          , 0.          , 0.5         , 0.5         , 0.          ,
          0.          ],
         [0.          , 0.          , 0.33333333, 0.33333333, 0.          ,
          0.33333333],
         [0.33333333, 0.33333333, 0.          , 0.          , 0.33333333,
          0.          ],
         [0.33333333, 0.          , 0.          , 0.33333333, 0.          ,
          0.33333333]])
```

规范化使得每一行的权重加起来为1

图卷积神经网络

- 图卷积神经网络、实践：一个例子
 - 假设 $G=(V, E)$, V 是节点的集合, 包含 N 个节点(Node)
 - A 为邻接矩阵, X 为节点属性
 - 也可以尝试进行**对称归一化**, 比如 $f(H^l, A) = \delta (D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}} H^l W^l)$
 - 这就是完整的GCN层与层之间传播公式的由来
 - 下图解释了各个成分的含义



图卷积神经网络





图卷积神经网络

- 图卷积神经网络实践

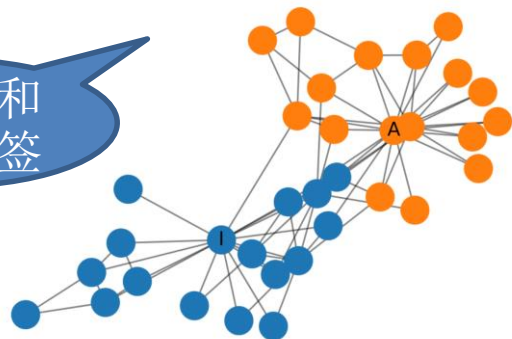
名称	类型	大小	修改日期
 01graph convolution neural network.ipynb	IPYNB 文件	157 KB	2021/11/17 20:26

图卷积神经网络

- 图卷积神经网络实践

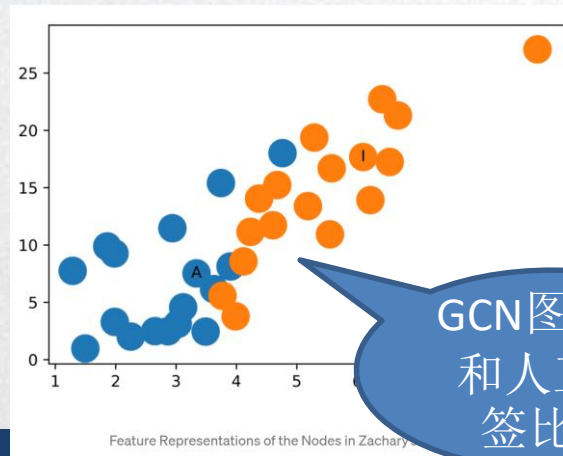
- 我们利用GCN对karate_club_graph图数据集进行处理，使用随机初始化的GCN进行特征提取，得到各个Node的Embedding表示
 - 然后根据Node的类别可视化，效果如图所示
 - 可见，在原数据中不同类别的Node，经过GCN提取出的Embedding，在空间上自动聚在一起，验证了图卷积神经网络的处理能力

图结构和
人工标签



Zachary's Karate Club

从图数据到
图嵌入



GCN图嵌入
和人工标
签比对

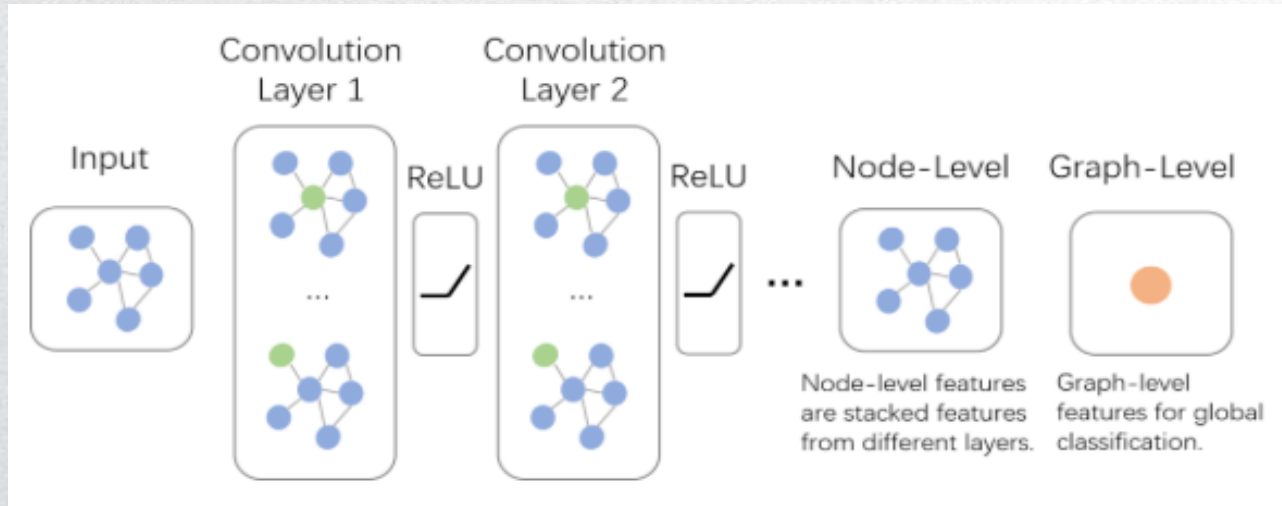
Feature Representations of the Nodes in Zachary

图卷积神经网络



图卷积神经网络

- 图卷积神经网络总结：图卷积框架



图卷积神经网络

- 图卷积神经网络总结：图卷积框架

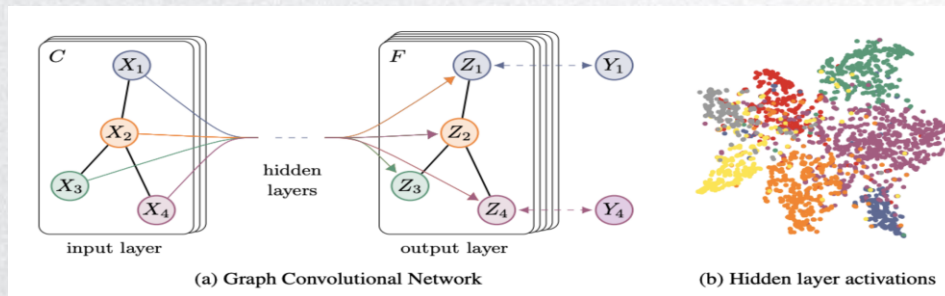
- 根据归一化的邻接矩阵聚合邻居信息
- 参数用来做特征转换

- $H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)})$
- $\tilde{A} = A + I_N$, A :邻接矩阵, I_N :单位矩阵, \tilde{A} : 考虑节点自己到自己的连接
- $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, 度矩阵
- $W^{(l)} \in \mathbb{R}^{D^{(l)} \times D^{(l+1)}}$, 第 l 层的权重矩阵
- $H^{(l)} \in \mathbb{R}^{N \times D^{(l)}}$, 第 l 层的节点的隐藏状态
- $H^{(0)} = X$, 初始特征

Thomas N. Kipf, Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. ICLR 2017

图卷积神经网络

- GCN的目标函数：一个简单的GCN



旧传号式

$$Z = f(X, A) = \text{softmax} \left(\hat{A} \text{ReLU} \left(\hat{A} X W^{(0)} \right) W^{(1)} \right)$$

$$\mathcal{L} = - \sum_{l \in \mathcal{Y}_L} \sum_{f=1}^F Y_{lf} \ln Z_{lf}$$

- 假设有3类(F)，有4个节点(yL)
- Y为实际标签，4个节点有4个标签
- Z为神经网络输出
 - X1, X2, X3, X4四个节点经过网络传播以后分别得到z1,z2,z3,z4
 - 它们都是3维向量，因为总共有3个类
- 左边的式子为y和z两者的交叉熵

节点	Y	Z
X1	100	0.8 0.1 0.1
X2	010	0.2 0.6 0.2
X3	010	0.2 0.5 0.3
X4	001	0.2 0.2 0.6

图卷积神经网络

