GCN, GAT, Graph SAGE

笔记本: 深度学习

创建时间: 2024/5/17 20:48 **更新时间:** 2024/5/18 14:21

作者: ztn

GCN: 图卷积网络

与CNN(卷积神经网络)的作用类似,都是特征提取器,但是GCN的处理对象是图数据;

首次将图像处理中的卷积操作简单的用到图结构数据处理中来,并且给出了具体的推导

(穿插: embedding: 将离散或高维的数据(文本,图像,图等)转换到一个连续的低维向量空间中的过程,这种转换保留了原始数据中的某些重要属性或者关系,使得数据可以在新的空间中更容易地进行处理和分析;、

常见的embedding:



设计了一种从图数据中提取特征的方法,这些特征可以用于图数据的节点分类,图分类,边预测等任务,顺便可以得到图的嵌入表示。

GCN的一个重要结点是<mark>其节点的聚合表征不包含它自己的特征</mark>,而是相邻节点的特征聚合,只有具有**自环**(self_loop)的节点才会在该聚合中包含自己的特征。为了避免大的节点在其特征表征中具有较大的值,度小的节点具有较小的值,可能会导致梯度消失或者梯度爆炸的问题,可以通过增加自环来解决。

什么是节点的聚合表征?

结点的聚合表征是通过聚合其邻域节点的信息来生成的,这个过程可以视为一种**特征提取**,用于捕捉节点的局部结构和属性信息;

具体来说:在图神经网络中,每个节点都与去其相邻节点通过边相连,为了生成节点的表征,图神经网络会聚合这些节点及其邻域节点的信息,这通常通过一系列的图卷积操作来实现,每个节点都会将其自身特征和邻域节点进行聚合,从而生成新的节点表征,(由于聚合了邻域节点的信息,节点的表征也具有一定的全局性,能够反映节点在整个图结构中的位置和角色)

聚合:聚合过程的目标是通过图中的节点和边特征来<mark>更新</mark>每个节点的表示,这通常包括两个 步骤,信息传递和信息聚合,信息传递是指通过节点之间的边将信息从邻居节点传递到目标 节点。而信息

聚合则是将传递过来的信息与目标节点的特征进行融合。

(eq:节点聚合可以学习用户和物品的嵌入,以捕捉用户的兴趣和物品的特征。)

在图像领域,CNN被拿来自动提取图像的结构——CNN处理的图像或者视频数据中像素点是排列成很整齐的矩阵,虽然图结构不整齐(不同的点具有不同数目的neighbors),但我们是不是可以用同样的方法去抽取图的特征呢?

于是就出现了两种方式来提取图的特征:一个是空间域卷积,一个是频域卷积;第一种方式由于每个顶点提取出来的neighbors不同,处理比较麻烦,同时效果没有频域卷积好。

GCN的计算原理:

$$H^{l+1} = \sigma(D^{-1/2}AD^{-1/2}H^lW^l)$$

(GCN的卷积核心计算公式)

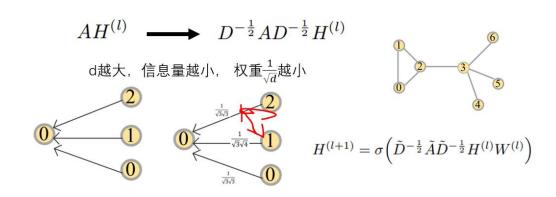
HI和HI+1分别是第I和第I+1的<mark>节点特征矩阵</mark>,其中每一行代表<mark>一个节点的特征向量</mark>,D为A

$$\tilde{D}_{ii} = \sum_{j} \tilde{A}_{ij \circ}$$

的过渡矩阵, A为邻接矩阵 (可能包含自环)并进行了<mark>归一化处理</mark>, Wl表示第l层的权重矩阵; 前面的σ是激活函数, A=A+In,是添加了自环的邻接矩阵 (In是单位矩阵), D就是A的过度矩阵;

描述了从一个层到下一个层的节点的特征矩阵的更新方式:即首先对邻接矩阵添加自环并进行归一化处理:DAD(上述),然后将归一化后的邻接矩阵与当前层的节点特征矩阵HI相乘,在与权重矩阵相乘,最后通过激活函数得到下一层的节点矩阵特征HI+1;

层? ——上次看的神经网络的基本结构 "三个层"

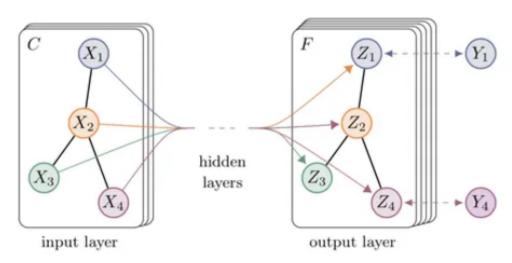


$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right).$$



 $\frac{1}{\sqrt{3}}*\frac{1}{\sqrt{3}}$ 那其实根据度矩阵以及上述计算方式:0-0 "DAD" $= \frac{1}{\sqrt{3}}*\frac{1}{\sqrt{3}}$,对应的边的权值是所连接 1 1 1 1

的两个顶点的对应的D的值;0-1: $\frac{1}{\sqrt{3}}*\frac{1}{\sqrt{3}}$, $\frac{1}{\sqrt{3}}*\frac{1}{\sqrt{4}}$



(a) Graph Convolutional Network

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

计算所有带有标签的样本的交叉熵:

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

Algorithm 1: WL-1 algorithm (Weisfeiler & Lehmann, 1968)

(也就是上面的GCN的核心计算公式的实现;