图神经网络 GNN 之图卷积网络 (GCN)

深度学习由于其强大的特征抽取能力和拟合能力，在不少领域取得很好的效果，替代了传统的机器学习和人工抽取特征的方法。但是传统的深度学习方法通常比较适用于欧式空间中表示的数据，而不能很好地解决非欧式空间的问题。非欧式空间的数据通常用图（graph）表示，用图表示对象之间的关系。为了更好地在图上应用深度学习，很多基于图的神经网络（GNN）算法被提出，本文重点 GNN 的开山之作图卷积神经网络 GCN。

1.前言

现在机器学习的研究领域存在很多神经网络模型，例如卷积神经网络（CNN)，循环神经网络（RNN)，自编码器（AutoEncoder）等，这些模型被用到很多人工智能任务中(目标检测、人脸识别，文本情感分析等)，取得不错的效果。这些任务通常是在欧式空间中表示的，传统的深度学习算法也比较擅长提取欧式空间中的特征。

但是现在越来越多任务的数据是从非欧式空间中生成的，需要用的图（graph）结构来表示对象之间的关系。例如在一些社交媒体上有用户社交网络，用户有不同的好友（两个为好友的用户是图中相邻的节点)，可以根据用户的好友关系对用户进行分类；在电子商务领域，用户和商品也可以构成网络，用户与其好友，用户与其买过的商品之间会存在连接，可以根据网络结构更好地推荐商品。

图结构相对比较复杂，一般不是整齐的，一个网络包含不同数量的节点，不同的节点也包含不同的邻居。这使得传统的神经网络操作（如卷积操作等）不能很好地用在图结构上，另外一点传统的机器学习方法通常假设样本之间是独立的，但是在图结构上，样本之间通常具有联系（好友关系，购买关系）等。图结构通常用 G=(V, E) 表示，V 表示节点集合，E 表示边的集合。

为了将深度学习的方法用到图结构上，图神经网络（Graph Neural Network, GNN）被提出。一些重要的神经网络操作在图结构上也被重新定义，如传统的卷积操作中，每一个像素可以看成一个点，然后和周围的点进行加权求和。而对于图（graph）结构，也可以采取类似的方法进行卷积，对节点在图上的邻居进行加权求和。

常见的图神经网络有很多种：图卷积网络 Graph Convolutional Network (GCN)；图循环网络 Graph Recurrent Network (GRN)；图注意力网络 Graph Attention Network (GAT)；图自编码器 Graph Autoencoders (GAE)。本文重点介绍最早被提出的图卷积网络 GCN，论文《Semi-Supervised Classification with Graph Convolutional Networks》。

2.图（graph）定义

下文用 G=(V, E) 表示图结构（有向图或无向图），V 表示节点的集合，E 表示边的集合，n 表示节点的个数，m 表示边的个数。

$$G = (V, E)$$

$$v_i \in V \text{ 表示图中的节点}$$

$$e_{ij} = (v_i, v_j) \in E \text{ 表示} v_i \text{和} v_j \text{在图中是邻居}$$

图 (graph) 定义

用 N(v) 表示节点 v 所有邻居的集合：

$$N(v) = \{u \in V | (v, u) \in E\}$$

节点 v 的邻居集合

图结构也可以用邻接矩阵（adjacency matrix）进行表示，邻接矩阵 **A** 为 n×n 矩阵，其中 n 为图中节点数，**A**ij=1 表示节点 i 和节点 j 之间有边相连。

$$\begin{cases} A_{ij} = 1 & if \ \ e_{ij} \in E \\ A_{ij} = 0 & if \ \ e_{ij} \notin E \end{cases}$$

邻接矩阵 A

图的度矩阵 **D**，degree matrix，**D** 是对角矩阵，即除了对角线其他元素都为 0，对角值如下计算，**D**ii 表示与节点 i 相连的节点数：

$$D_{ii} = \sum_j A_{ij}$$

度矩阵 D

图中的节点 v 带有特征向量，用矩阵 **X** 保存图节点的特征向量，特征向量维度是 d：

$$X \in R^{n \times d}$$

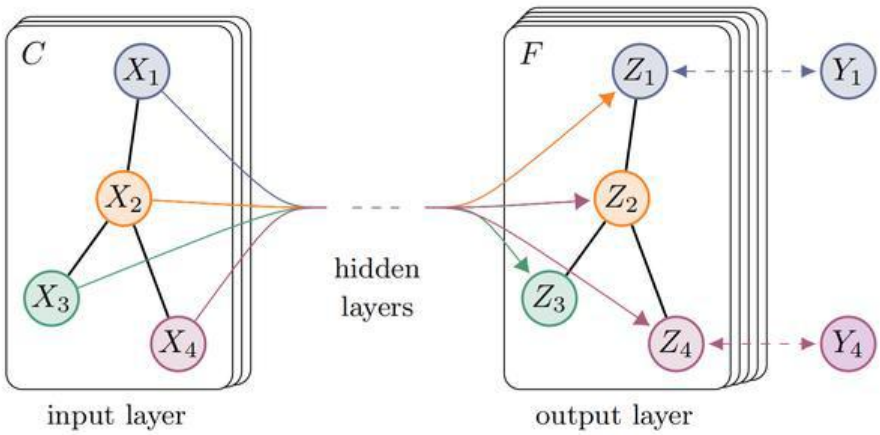
$$x_i \in X \text{ 表示节点} i \text{的特征向量}$$

$$x_i \in R^d$$

节点特征向量矩阵

3.图卷积神经网络网络 GCN

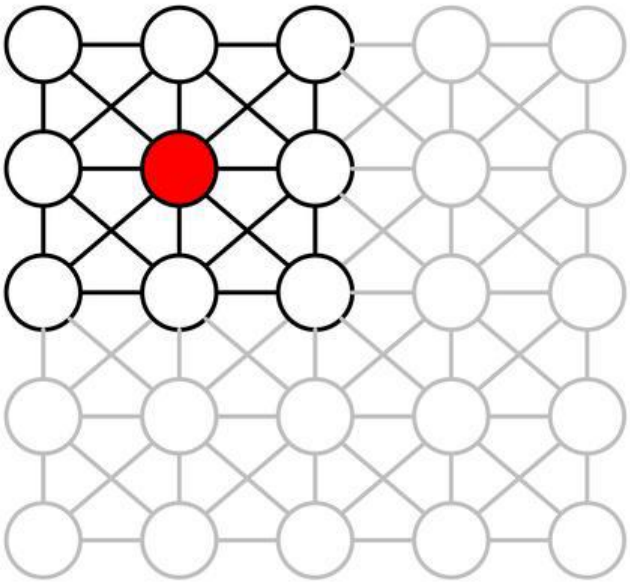
图卷积神经网络网络 Graph Convolutional Network (GCN) 最早是在 2016 年提出，2017 年发表在 ICLR 上。GCN 主要是将卷积操作应用到图结构上，如下图所示，GCN 输入的 chanel 为 C (即节点 Xi 特征向量的维度)，GCN 输出的 chanel 为 F，即每个节点 (Zi) 的特征向量维度为 F，最后用节点的特征对节点进行分类预测等：



(a) Graph Convolutional Network

GCN 卷积示意图

GCN 和 CNN 类似，具有强大的特征学习能力，它们的实质也是类似的，某个点的卷积可以看成对该点邻居的加权求和。下图为传统的图像（image）卷积操作，每个点表示一个像素，图像的像素也可以看成是一种图（graph）结构，相邻的像素之间有边连接，而卷积是对像素邻居进行加权平均得到的。



图像（image）的二维卷积

而对于图（graph）结构，也可以采取类似的方法进行卷积。例如对下图中红色的节点卷积就等于取其邻居节点进行加权平均。

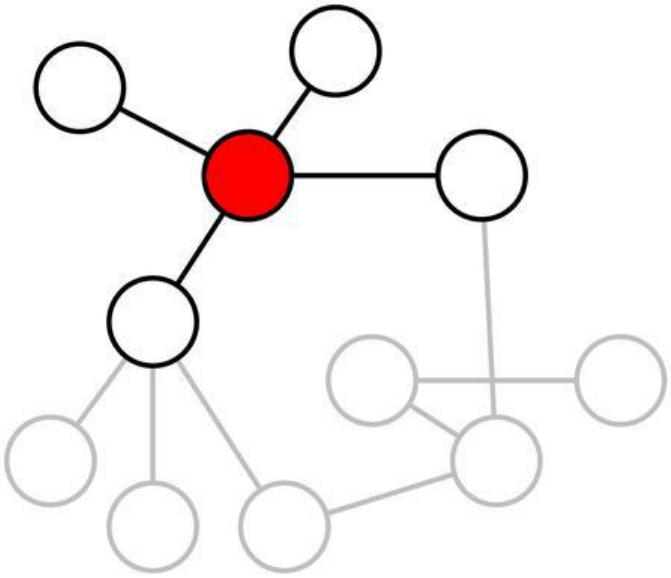


图 (graph) 结构卷积

GCN 中给出了图卷积的计算公式，如下所示，其中 H(l) 表示节点在第 l 层的特征向量，H(l+1) 表示经过卷积后节点在第 l+1 层的特征向量，W(l) 表示第 l 层卷积的参数，σ 表示激活函数。而由矩阵 **A**、**D** 组成的部分是一种拉普拉斯矩阵 (Laplacian matrix)，**A**+**I** 中的 **I** 为单位矩阵，即对角线为 1，其他为 0 的矩阵：

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

$$\tilde{A} = A + I$$

$$\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$$

GCN 卷积公式

上面的公式就是就是 GCN 卷积操作的公式了，通过多层 GCN 卷积，就可以提取出每个节点需要的信息，用于各种分类或分析。

上面的公式需要从拉普拉斯矩阵和傅里叶变换推导得出，过程比较复杂，本人目前还有些地方没有理解，以后会再写一篇文章介绍如何从拉普拉斯矩阵和傅里叶变换得到上面的公式。本文先从一种容易理解的角度去了解公式的具体含义。

另外头条搜索真心方便，里面有很多关于图卷积网络 GCN 的干货知识，大家可以直接搜索“GCN 与拉普拉斯矩阵”，或者点击下方的搜索卡片直接查看，了解 GCN 和拉

4.GCN 卷积公式的理解

对图像（image）中某一个像素进行卷积实际上是对该像素及相邻像素进行加权求和，那么在图（graph）结构中对某一个节点卷积就应该是**对该节点和其邻居节点进行加权求和。**

我们首先考虑**最简单的图卷积操作**，如下公式：

$$H^{(l+1)} = f\left(H^{(l)}, A\right) = \sigma\left(AH^{(l)}W^{(l)}\right)$$
$$H^{(0)} = X \in R^{n\times d}$$

简单的图卷积公式

上面的公式中 **W** 为卷积变换的参数，可以训练优化。**A** 矩阵为邻接矩阵，**A**ij 不为 0 则表示节点 i，j 为邻居。**H** 为所有节点的特征向量矩阵，每一行是一个节点的特征向量，

H(0) 就是 **X** 矩阵。**A** 和 **H** 的乘积其实就是把所有的邻居节点向量进行相加，如下所示。

$$\underset{\mathbf{A}}{\begin{bmatrix}0&1&1&0\\1&0&0&0\\1&0&0&1\\0&0&1&0\end{bmatrix}}\underset{\mathbf{H}}{\begin{bmatrix}1&1&1&1&1\\2&2&2&2&2\\3&3&3&3&3\\4&4&4&4&4\end{bmatrix}}=\underset{\mathbf{AH}}{\begin{bmatrix}5&5&5&5&5\\1&1&1&1&1\\5&5&5&5&5\\3&3&3&3&3\end{bmatrix}}$$

矩阵 A 和 H 相乘

上式的邻接矩阵中，节点 1 和节点 2、3 为邻居，则 **A** 乘上 **H** 后节点 1 的特征向量就用节点 2、3 特征向量加和得到，即 [2,2,2,2,2]+[3,3,3,3,3]=[5,5,5,5,5]。得到 **AH** 之后再和 **W** 相乘，最后经过激活函数 σ 就得到下一层节点的特征向量了。

但是上面的公式存在一些问题，**AH 只获得了某个节点的邻居信息，而忽略了节点本身信息。**为了解决这个问题，我们可以将矩阵 **A** 中对角线的值设为 1，即每个节点会指向自身，新的卷积公式如下：

$$H^{(l+1)}=\sigma\left(\tilde{A}H^{(l)}W^{(l)}\right)$$
$$\tilde{A}=A+I_N$$

I 为单位矩阵，即对角线为 1，其余为 0。使用上面的卷积公式即可把节点自身的信息也考虑进去，但是这个公式仍然存在问题：**矩阵 A 没有归一化，AH 会把节点所有邻居的向量都相加，这样经过多层卷积后向量的值会很大。**

因此需要对矩阵 **A** 进行归一化，归一化要用到图的度矩阵 **D**，可以直接使用矩阵 **D** 的逆和 **A** 相乘，如下：

$$A = D^{-1}A$$

$$A_{ij} = \frac{A_{ij}}{d_i}$$

邻接矩阵归一化

但是上面的归一化公式得到的 **A** 不是对称矩阵，通常使用对称归一化的方法：

$$A = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$$

$$A_{ij} = \frac{A_{ij}}{\sqrt{d_i}\sqrt{d_j}}$$

邻接矩阵对称归一化

把上面的优化方法结合在一起，就是 GCN 的卷积公式了：

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

$$\tilde{A} = A + I$$

$$\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$$

GCN 卷积公式

5.GCN 预测

GCN 可以用最后一层卷积层得到的节点特征向量进行预测，假设 GCN 只有两层，则预测的公式如下：

$$\hat{A} = \widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}$$

$$Z = \mathrm{softmax}\left(\hat{A}\mathrm{ReLu}\left(\hat{A}XW^{(0)}\right)W^{(1)}\right)$$

GCN 预测

分类的损失函数如下：

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

Y_L表示有类标的节点集合

F表示类别个数

GCN 损失函数

https://www.toutiao.com/i6841472483535618568/?group_id=6841472483535618568