

# 结合残差网络和噪声处理的节点分类模型

曹磊亮, 谢瑾奎, 郭天晟

(华东师范大学 计算机科学与技术学院, 上海 200333)

E-mail: jlxie@cs.ecnu.edu.cn

**摘要:** 图卷积网络(GCNs)通过迭代堆叠多层卷积聚合操作来学习节点表示,在许多图学习的任务中显现出强大的潜力。但是,由于图卷积运算的过度平滑效应,大多数基于GCN的模型无法对深层进行建模。此外,实际应用中往往会有许多噪声数据,在深层建模时会极大地干扰GCN的节点分类性能。为了解决上述问题,本文提出了一种结合残差网络和噪声处理的节点分类模型(NHGCN)。针对噪声节点,模型使用自适应的残差处理方法并修改了范数约束,提高了对噪声节点的修正能力。针对过度平滑问题,模型利用残差映射优化图卷积的信息传播,缓解了图卷积聚合操作中的梯度消失问题,保留了节点独特性,增强了节点分类效果。在Cora, Citeseer, Pubmed这3个公开数据集上进行了训练和测试,实验结果表明:相较于现有的节点分类模型,本文模型实现了更优的处理过度平滑效果和抗噪性能。

**关键词:** 节点分类; 过度平滑; 图卷积网络; 噪声处理

中图分类号: TP391

文献标识码: A

文章编号: 1000-1220(2024)06-1331-08

## Node Classification Model Combining Residual Network and Noise Processing

CAO Leiliang, XIE Jinkui, GUO Tiansheng

(School of Computer Science and Technology, East China Normal University, Shanghai 200333, China)

**Abstract:** Graph Convolutional Networks (GCNs) learn the representation of nodes by iteratively stacking multiple layers of convolutional aggregation operations and show powerful potential in many tasks of graph learning. However, most GCN-based models are unable to model deep layers due to the excessive smoothing effect of graph convolutional operations. In addition, there are often many noisy data in practical applications, which can greatly interfere with the node classification performance of GCNs when modeling deep layers. To solve the above problems, this paper proposes a node classification model (NHGCN) that combines residual networks and noise processing. For noisy nodes, the model uses an adaptive residual processing method and modifies the parametric constraints to improve the correction ability of noisy nodes. For the over-smoothing problem, the model uses the residual mapping to optimize the information propagation of the graph convolution, which alleviates the gradient disappearance problem in the graph convolution aggregation operation, preserves the node uniqueness, and enhances the node classification effect. The model in this paper was trained and tested on three publicly available datasets, Cora, Citeseer, and Pubmed. The experimental results show that the model in this paper achieves better processing over-smoothing effect and noise immunity compared with the existing node classification models.

**Keywords:** node classification; over-smoothing; graph convolutional networks; noise handling

## 0 引言

近年来,图卷积网络 GCN<sup>[1]</sup>引起越来越多的关注,GCN是卷积神经网络 CNN<sup>[2]</sup>的有效变体,用于处理基于图形的数据,其基本思想是通过迭代聚合来自其局部图邻居的信息来更新目标节点的嵌入。此外,GCN及其变体已广泛应用于社交网络分析<sup>[3]</sup>、交通预测<sup>[4]</sup>和推荐系统<sup>[5]</sup>等应用,其中,GCN在提升节点分类任务的性能上展现出巨大的潜力。

尽管基于GCN的节点分类相对成功,但本文认为基于GCN的节点分类任务中存在两个重要问题仍未解决。一方面,目前的大多数GCN模型都是浅层模型,如:GCN、GAT<sup>[6]</sup>在两层模型中实现了最佳性能,这种浅层结构限制了它们从

高阶邻居中提取信息的能力。然而,堆叠更多层和添加非线性往往会降低这些模型的性能。GCN可以看作是一个低通滤波器,使用多层GCN之后,节点的区分性越来越差,节点的表示向量趋于一致,这就使得以GCN为基础的后续学习任务难以进行,这种现象称为过度平滑问题<sup>[7]</sup>。另一方面,现有的GCN模型在大量基准数据集中取得了优异的性能,其中节点特征被假设为完整且信息丰富。然而,在实际应用中,节点特征可能会从各个方面出现异常。例如,在社交网络中,新用户在与他人建立联系之前可能没有完整的个人资料,从而导致用户功能缺失。在交通网络中,由于存在某些噪声,节点特征可能会有噪声交通信息观测中的不确定性和动态性。在电商销售中,某些商家会采用购买虚假用户进行虚拟评分,不仅影响到

总体评分结果还会影响用户相似度的计算。

为了解决上述问题,本文提出了一种结合残差网络和噪声处理的节点分类模型 NHGCN(Noise Handling Graph Convolutional Network)。本文的主要贡献如下:

1) 针对噪声节点,模型使用自适应残差处理方式,并将 Frobenius 范数修改为 21 范数,加强了对噪声特征的修正能力,提高对噪声节点的恢复能力和对正常节点的性能。

2) 提出了一种基于残差网络的节点分类模型 NHGCN,在每一层学习剩余项偏好,使得节点的独特性在较低层得以保留,有效缓解 GCN 迭代过程中的过度平滑问题。

3) 在 Cora, Citeseer, Pubmed 3 个公开数据集上进行了训练和测试,相较于现有的节点分类模型,本文模型实现了更优的处理过平滑效果和抗噪性能,证明模型 NHGCN 的有效性。

## 1 相关工作

节点分类是图分析中的重要课题,早已受到了众多学者的关注,近年来,图神经网络的热潮更是给该课题注入了新活力,但是,过平滑问题严重干扰了深层网络下节点分类的准确率。图的噪声处理是一种能够提高节点分类准确率的技术,它可以减少误差,让原始数据产出更多价值,是近年来的一个热点研究方向。接下来的部分将详细介绍图结构中的过平滑问题与噪声处理的研究现状。

### 1.1 过平滑问题

随着科学进步,节点分类的任务在不断的优化与创新,GCN 能够充分的整合节点的信息,发掘文章、视频、图片之间的潜在语义关系,在处理图数据方面应用广泛。尽管 GCN 为图数据的学习带来了一种全新的方式,但是,随着网络层数的增加和迭代次数的增加,节点的表征会趋向于收敛到同一个值,这种过度平滑问题会影响下游任务的进行。现有的针对过平滑问题工作可以分为两类,使用剩余连接或跳过连接以及将深度传播与浅层神经网络相结合。

一方面,JKNet<sup>[8]</sup> 为每个节点灵活地利用不同的邻域范围,以实现更好的结构感知表示,它使用密集跳过连接来组合每层的输出,以保持节点表示的局部性。DropEdge<sup>[9]</sup> 提出,通过从输入图中随机删除一些边,可以减轻过度平滑的影响。Chiang 等人<sup>[10]</sup> 认为不能完全忽略了相邻节点的权重,他们在卷积公式的基础上做出优化。ResNet<sup>[11]</sup> 提出了一个残差学习框架,以简化网络的训练,通过数据的预处理以及在网络中使用 BN 层来解决梯度消失问题。N-GCN<sup>[12]</sup> 通过不同尺度感受野的组合提高模型对节点的表征能力。Huang 等人<sup>[13]</sup> 使用 RNN 对层之间的长期依赖建模,以此来提高模型性能。GCNII<sup>[14]</sup> 通过初始残余连接和身份映射来防止过度平滑,在每一层,初始残差从输入层构造一个跳跃连接,与此同时,单位映射将单位矩阵添加到权重矩阵。

另一方面,有几种方法尝试将深度传播与浅层神经网络相结合。SGC<sup>[15]</sup> 试图通过在一个神经网络层中应用图卷积矩阵的高阶次方来捕获图中的信息,它通过去除层之间的所有可训练权重来简化模型,以避免过度拟合的可能性。APPNP<sup>[16]</sup> 提出个性化信息传播,利用神经预测的个性化传播及其快速近似,使用一个大的、可调整的邻域的信息对来处理过

平滑问题。GDC<sup>[17]</sup> 提高了在无监督任务上的广泛模型的准确性,并且模型只需要很少的超参数,泛化图扩散过程,进一步扩展了 APPNP。

### 1.2 噪声处理

图神经网络的目标是学习每个图节点的低维潜在表示,以用于下游的分析任务,如:分类、聚类、推荐。噪声节点会大幅提高图形结构噪声的脆弱性,国内外很多学者在这方面进行了研究。Pro-GNN<sup>[18]</sup> 从扰动图中联合学习结构图和鲁棒图神经网络模型,用于处理噪声节点。IDGL<sup>[19]</sup> 将迭代学习引入图结构学习,提出了一种新的端到端图学习框架,用于学习图结构和图嵌入。当学习的图结构接近用于预测的优化图时,IDGL 动态停止。ElasticGNN<sup>[20]</sup> 将一种新的、高效的通用消息传递方案加入到模型中,结合深度神经网络,形成了对噪声攻击具有显著的鲁棒性弹性 GNN 模型。AirGNN<sup>[21]</sup> 从图拉普拉斯平滑的角度分析了噪声节点和正常节点之间的关系,并且提出一种新的节点自适应过渡方式,有效处理了噪声节点和对抗性节点对模型的影响。

## 2 预备知识

### 2.1 符号定义

本文使用  $X$  来表示矩阵,给定一个矩阵  $X \in R^{n \times d}$ ,本文用  $X_i$  表示矩阵第  $i$  行,用  $X_{ij}$  表示第  $i$  行第  $j$  列的元素,矩阵的 Frobenius 范数和 21 范数定义为  $\|X\|_F = \sqrt{\sum_{ij} X_{ij}^2}$  和  $\|X\|_{21} = \sum_i \|X_i\|_2 = \sum_i \sqrt{\sum_j X_{ij}^2}$ 。此外,本文定义  $\|X\|_2 = \sigma_{\max}(X)$ ,其中  $\sigma_{\max}(X)$  是矩阵  $X$  的最大奇异值。最后,本文定义近似点算子为  $\text{prox}_f(x) = \underset{u}{\text{argmin}}(f(u) + \frac{1}{2} \|u - x\|_2^2)$ 。

用  $G = (V, E)$  表示带有节点集  $V = \{v_1, \dots, v_n\}$  和无向边集  $E = \{e_1, \dots, e_m\}$  的图。本文用  $N(v_i)$  来表示包含  $v_i$  节点的邻居集。假设每个节点都与  $d$  维特征向量相关联,那么可以得出特征矩阵  $X_{fea} \in R^{n \times d}$ 。邻接矩阵  $A \in A^{n \times n}$  中的每个元素  $A_{ij}$  定义如下,当  $v_i$  和  $v_j$  有边相连时,  $A_{ij} = 1$ , 否则,  $A_{ij} = 0$ 。图的拉普拉斯矩阵定义为  $L = D - A$ ,其中  $D$  是对角度矩阵。GCN<sup>[1]</sup> 中常用的特征聚合矩阵表示  $\bar{A} = \hat{D}^{-0.5} \hat{A} \hat{D}^{-0.5}$ ,其中  $\hat{A} = A + I$  是具有自环的邻接矩阵,其度矩阵为  $\hat{D}$ ,相应的拉普拉斯矩阵定义为  $\hat{L} = I - \hat{A}$ 。

### 2.2 图卷积网络

对于每个节点  $v_i \in V$ ,本文用  $h_i^0$  表示节点的初始嵌入,用  $x_i$  表示节点  $v_i$  的特征向量(即  $h_i^0 = x_i$ )。在图  $G$  中,GCN 的关键思想是递归消息传递或特征传播方式堆叠  $K$  个步骤,以学习节点嵌入。对于步骤中的每个节点  $v_i$ ,它通过以下两个步骤递归计算:特征传播和非线性特征变换。

特征传播的过程如下,对于每个节点  $v_i$ ,聚合来自图邻居  $S_i$  的嵌入及其在前一层的自身嵌入  $h_i^k$ ,以此来形成新的嵌入。其特征聚合函数为:

$$\bar{H}^{(k+1)} = \hat{D}^{-0.5} \hat{A} \hat{D}^{-0.5} H^{(k)} \quad (1)$$

给定第  $k$  层的特征  $H^{(k)}$ ,特征传播输出  $H^{(k+1)}$  可被视为前一层特征的拉普拉斯平滑。

非线性变换层是标准的多层感知器(MLP),它是通过输入特征传播步骤的输出,非线性变换产生每个节点的第  $(k +$

1)层嵌入,如公式(2)所示:

$$H^{(k+1)} = \sigma(\bar{H}^{(k+1)} W^k) \quad (2)$$

其中,  $\sigma(x)$  是非线性激活函数.

在每个层中以定义的深度  $K$  迭代执行两个步骤后,每个节点在深度  $K$  处的最终嵌入是  $h_i^K$ . 对于大多数基于 GCN 的应用程序,存在一个预测函数  $f(\cdot)$ :

$$\hat{y} = f(h_i^K | v_i \in v) \quad (3)$$

### 3 模型设计

#### 3.1 模型概述

本文设计了基于残差网络和噪声处理的节点分类模型

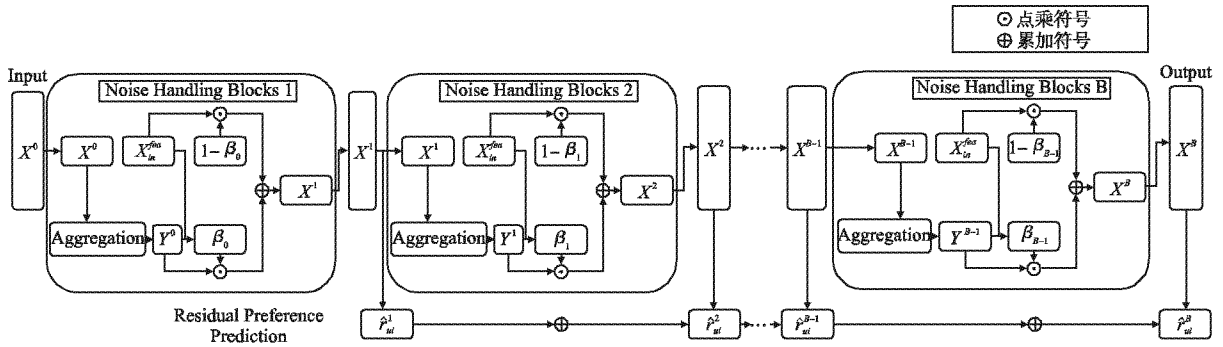


图1 模型框架图

Fig. 1 Model framework

处理噪声节点,修正噪声特征,强化正常特征.如果某节点与其邻域节点显著不一致,即噪声节点,此时,给予特征聚合较大的权重,利用邻居节点的特征来修正噪声节点.反之,如果某节点与其邻域节点基本一致,给予残差连接较大的权重,这样节点的独特性得以保留,加强了模型对正常节点的性能.

#### 3.2.1 初步研究

为了研究 GCNII 在噪声和正常节点特征上的行为,本文在数据集 Cora 上设计了半监督节点分类实验研究,实验将分配 10% 的节点来模拟噪声节点,具有或不具有噪声节点,表示为 GCNII-AB 和 GCNII.如图 2 所示,随着层数增加,噪声节点不仅会导致模型的性能下降,还会使模型性能浮动范围扩大,从而验证了处理噪声节点的必要性.

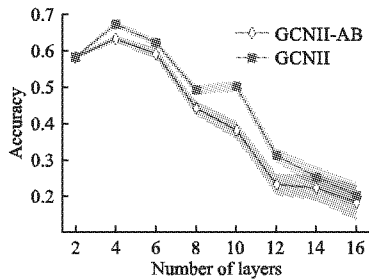


图2 GCNII 在不同深度的节点分类精度表现 (Cora)

Fig. 2 Node classification accuracy performance of GCNII at different depths (Cora)

#### 3.2.2 设计启发

国内外许多学者在处理噪声节点方面进行了自己的研

NHGCN,与现有的基于 GCN 的节点分类模型不同,NHGCN 不仅能有效缓解过平滑问题,还能处理噪声节点,模型的目标是利用自适应的噪声处理方式有效处理噪声节点,利用残差学习提高节点的区分度,二者相互增强,从而提高模型的整体性能.模型的整体架构如图 1 所示,噪声处理模块通过使用自适应残差处理方式并调整范数约束,修正噪声节点,从而提高对噪声节点的恢复能力和对正常节点的性能.残差学习模块则是在每一层根据嵌入提取节点偏好,然后进行累加学习,从而进行节点分类.

#### 3.2 噪声处理模块

噪声处理模块 NHB (Noise Handling Block) 的目的是有效

究,为了处理噪声节点并调整特征平滑度,APPNP<sup>[16]</sup> 在消息传递中使用剩余连接,如公式(4)所示:

$$X^{k+1} = (1 - \alpha) \bar{A} X^k + \alpha X_{in}^{fea} \quad (4)$$

其中  $X_0 = X_{in}^{fea}$ ,可以视为初始化  $X = X_{in}^{fea}$ 、步长  $\gamma = 1 - \alpha$  拉普拉斯平滑问题的迭代解<sup>[22]</sup>:

$$\begin{aligned} \arg \min_{X \in R^{n \times d}} \sigma_1(X) &= \frac{\alpha}{2(1 - \alpha)} \|X - X_{in}^{fea}\|_F^2 + \\ &\frac{1}{2} \text{tr}(X^T (I - \bar{A}) X) \end{aligned} \quad (5)$$

GCNII 在这之上做了更深的拓展,它采用了类似的消息传递,但在每个消息传递步骤中进一步结合了特征转换层,这导致了耦合架构,与 APPNP 的解耦架构形成对比.如上的消息传递方式,避免了特征仅取决于节点度的问题,从而缓解噪声节点带来的信息干扰.更直观地说,通过残差学习提供了层间的直接信息流,可以保留一些必要的高频信号,以便更好地区分节点类别.

特征聚合有助于平滑噪声特征,而它可能导致正常特征的不适当平滑,而残余连接可以有效提高节点的区分度,对于调整特征平滑度至关重要,但它也可能放大噪声特征的影响.由此,保持等式(5)中输入和输出特征之间的接近性很重要.

#### 3.2.3 调整范数约束

对于等式(5)中输入和输出特征,尽管可以通过调整参数(如 APPNP<sup>[16]</sup>中的残差权重  $\alpha$ )来一定程度上缓解这种冲突,但是这种全局调整往往不能适应节点的子集,例如具有噪声特征的节点,而在实践中,经常会遇到只有一部分节点包含噪声特征的情况.此外,Frobenius 范数对于噪声特征的处理效果仍需要改进,于是本文决定修改范数约束.在公式(5)直

接用 Frobenius 范数的平方来处理偏差可能过于激进,也就是  $\|X - X_{in}^{fea}\|_F^2 = \sum_{i=1}^n \|X_i - (X_{in}^{fea})_i\|_2^2$ . 这种惩罚不能容忍大偏差的事实削弱了通过拉普拉斯平滑去除噪声特征的能力,这促使本文考虑使用 21 范数来替代 Frobenius 范数,即:

$$\|X - X_{in}^{fea}\|_{21} = \sum_{i=1}^n \|X_i - (X_{in}^{fea})_i\|_2 \quad (6)$$

使用 21 范数不仅可以促进进行稀疏性  $X - X_{in}^{fea}$ , 另外, 其对较大值的惩罚不太激进, 从而增加了删除噪声特征的可能性. 因而, 本文采用该偏差. 该偏差控制正则化的拉普拉斯平滑问题为:

$$\arg \min_{X \in R^{n \times d}} \theta \|X - X_{in}^{fea}\|_{21} + \frac{1}{2} \text{tr}(X^T(I - \tilde{A})X) \quad (7)$$

式中  $\theta \in [0, \infty)$  是一个参数, 用于调整接近度和拉普拉斯平滑之间的平衡. 为了便于调节参数  $\theta$ , 本文对公式(7)进行了修改:

$$\arg \min_{X \in R^{n \times d}} \sigma(X) := \theta \|X - X_{in}^{fea}\|_{21} + (1 - \theta) \text{tr}(X^T(I - \tilde{A})X) \quad (8)$$

### 3.2.4 NHB 模块

$\sigma(X)$  是包含非平滑和平滑组件的复合目标. 本文通过近端梯度下降对其进行优化, 并获得以下迭代作为噪声处理模块 NHB:

$$Y^k = X^k - 2\gamma(1 - \theta)(I - \tilde{A})X^k \\ = (1 - 2\gamma(1 - \theta))X^k + 2\gamma(1 - \theta)\tilde{A}X^k \quad (9)$$

$$X^{k+1} = \arg \min_X \left\{ \theta \|X - X_{in}^{fea}\|_{21} + \frac{1}{2\gamma} \|X - Y^k\|_F^2 \right\} \quad (10)$$

其中  $X^0 = X_{in}^{fea}$ ,  $\gamma$  是稍后指定的步长. 令  $Z = X - X_{in}^{fea}$ , 那么公式(10)可被重写为:

$$Z^{k+1} = \arg \min_Z \left\{ \theta \|Z\|_{21} + \frac{1}{2\gamma} \|Z - (Y^k - X_{in}^{fea})\|_F^2 \right\} \quad (11)$$

根据近似点算子的定义<sup>[23]</sup>:

$$\text{prox}_f(x) = \arg \min_u (f(u) + \frac{1}{2} \|u - x\|_2^2). \quad (12)$$

公式(11)可被重写为:

$$Z^{k+1} = \text{prox}_{\gamma\theta} \cdot \|_{21} (Y^k - X_{in}^{fea}) \quad (13)$$

$$X^{k+1} = X_{in}^{fea} + Z^{k+1} \quad (14)$$

公式(13)中近端算子的第  $i$  行可以通过解析计算得出:

$$(\text{prox}_{\gamma\theta} \cdot \|_{21} (X))_i = \frac{X_i}{\|X_i\|_2} \max(\|X_i\|_2 - \gamma\theta, 0) \\ = \max\left(1 - \frac{\gamma\theta}{\|X_i\|_2}, 0\right) \cdot X_i \quad (15)$$

在公式(15)中用  $Y^k - X_{in}^{fea}$  代入  $X$  然后结合公式(13)和公式(14), 公式(14)可被改写为:

$$X_i^{k+1} = (X_{in}^{fea})_i + \beta_i (Y_i^k - (X_{in}^{fea})_i) \\ = (1 - \beta_i)(X_{in}^{fea})_i + \beta_i Y_i^k, \forall i \in [n] \quad (16)$$

其中  $\beta_i = \max\left(1 - \frac{\gamma\theta}{\|Y_i^k - (X_{in}^{fea})_i\|_2}, 0\right)$ . 具体而言, NHB 模块的工作原理如下:

第 1 步在本地邻居中进行特征聚合, 自循环的权重为  $1 - 2\gamma(1 - \theta)$ , 第 2 步利用局部偏差  $\|Y_i^k - (X_{in}^{fea})_i\|_2$  计算每个节点  $v_i$  的权重  $\beta_i$ , 最后一步是将输入特征  $X_{in}^{fea}$  和聚合特征  $Y_i^k$  进行线性组合, 权重分别为  $(1 - \beta_i)$  和  $\beta_i$ .

$$Y^k = (1 - 2\gamma(1 - \theta))X^k + 2\gamma(1 - \theta)\tilde{A}X^k \\ \left\{ \begin{aligned} \beta_i &= \max\left(1 - \frac{\gamma\theta}{\|Y_i^k - (X_{in}^{fea})_i\|_2}, 0\right), \forall i \in [n] \\ X_i^{k+1} &= (1 - \beta_i)(X_{in}^{fea})_i + \beta_i Y_i^k, \forall i \in [n] \end{aligned} \right. \quad (17)$$

如果设置步长  $\gamma = \frac{1}{4(1 - \theta)}$  或  $\gamma = \frac{1}{2(1 - \theta)}$ , NHB 会被简化为

$Y^k = \frac{1}{2}X^k + \frac{1}{2}\tilde{A}X^k$  和  $Y^k = \tilde{A}X^k$ , 步长的选择只会影响 NHB 的收敛速度, 并不会影响 NHB 收敛到不动点解的最终结果.

### 3.2.5 NHB 的解释

首先, 如果节点  $v_i$  也就是  $(X_{in}^{fea})_i$  与其邻居节点的特征也就是  $Y_i^k$  显著不一致, 那么局部偏差  $\|Y_i^k - (X_{in}^{fea})_i\|_2$  就会很大, 导致  $\beta_i$  趋向于 1, 在最后一步中, 残差  $(1 - \beta_i)(X_{in}^{fea})_i$  只有很小的权重值, 聚合特征  $Y^k$  会占主要地位.

相反, 如果  $(X_{in}^{fea})_i$  与其邻居节点的特征也就是  $Y_i^k$  基本完全一致, 那么局部偏差  $\|Y_i^k - (X_{in}^{fea})_i\|_2$  就会很小, 导致  $\beta_i$  趋向于 0, 在最后一步中, 残差  $(1 - \beta_i)(X_{in}^{fea})_i$  会有较大权重, 聚合特征  $Y^k$  占较小比重.

总而言之, 局部偏差  $\|Y_i^k - (X_{in}^{fea})_i\|_2$  提供了从  $\beta_i \rightarrow 1$  到  $\beta_i \rightarrow 0$  的自然过渡, 这种转变可以由参数  $\theta$  实现, 而参数  $\theta$  可以通过交叉验证作为超参数学习或调整.

根据图结构数据的同质假设<sup>[24]</sup>, 正常节点的特征表示相比噪声节点更贴近局部邻居. 因此, NHB 将为正常特征分配更多的残差 (即较小的  $\beta$ ), 为噪声特征分配较少的残差 (即较大的  $\beta$ ), 从而在特征聚合和残差连接之间提供定制的权衡. 因此, 它可以提高对噪声特征的恢复能力和对正常特征的性能.

### 3.3 残差学习模块

残差学习模块的目标是学习剩余项偏好, 使得节点的独特性在较低层得以保留, 从而有效减少梯度消失, 缓解 GCN 迭代过程中的过度平滑问题.

GCN 在 Cora 数据集不同深度的性能表现如图 3 所示, 在特征聚合中有或没有残余连接, 表示为 GCN-RE 和 GCN. 所有方法都遵循原始论文中的超参数设置.

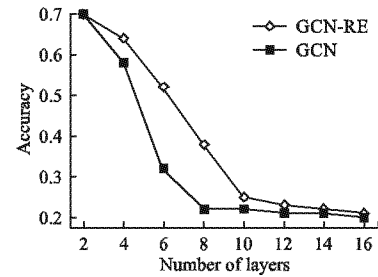


图 3 GCN 在 Cora 数据集不同深度的性能表现  
Fig. 3 Node classification accuracy performance of GCN at different depths (Cora)

如图 3 所示, 基于 GCN 的节点分类模型在层数为二时达到最佳性能, 残余连接可以有效减少 GCN 在迭代过程中的信息损失, 反映在图 3 中就是分类准确率性能衰减的相对较慢. 本文推测一个可能的原因是, 在第  $k$  层, 每个节点的嵌入都是

由二部图中的  $k$  阶邻居平滑的. 因此, 当层数递增时, 更深层的节点嵌入往往会过度平滑, 即它们更相似, 从而难以区分. 残差网络在每一层学习剩余项的偏好, 减少了信息损失, 保留了节点的独特性, 也就缓解过度平滑问题.

带有相似特征的节点更有可能有相连的边, 相连的节点之间通常也有类似的特征, 因此, 许多基于嵌入的节点分类模型将偏好  $\hat{r}_{wi}^k$  预测为节点与节点潜在向量之间的内积, 如公式 (18) 所示:

$$\hat{r}_{wi}^k = \langle e_u^k, e_v^k \rangle \quad (18)$$

其中  $\langle, \rangle$  表示向量内积运算, 本文考虑当两个节点的嵌入重合度越高, 两个节点就越相似, 对于预定深度  $K$  的节点  $a$  和节点  $b$ , 它们的相似度计算为:

$$\text{sim}(a, b) = \frac{\langle e_a^K, e_b^K \rangle}{\|e_a^K\|^2 \|e_b^K\|^2} \quad (19)$$

对于不同的层数  $K$  值, 计算平均成对节点-节点的嵌入相似度.

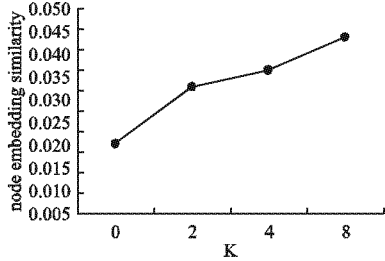


图4 节点嵌入相似度

Fig.4 Node embedding similarity

在图4中绘制了节点对的相似性. 如图4所示: 当  $K$  增加时, 节点嵌入之间的相似度逐渐增加, 这是因为 GCN 的卷积操作在层数过多时会使得每个节点几乎都能间接融合其它所有节点的信息, 导致所有节点的表征趋于相同, 这也会导致节点的区分度变低, 从而干扰节点分类的性能.

基于以上两个观察结果, 本文认为残差学习可以保留节点独特性, 有效缓解过平滑问题带来的信息损失, 所以本文选择不直接近似每个层上每个节点对的偏好, 而是执行剩余偏好学习:

$$\hat{r}_{wi}^{B+1} = \hat{r}_{wi}^B + \langle e_u^{B+1}, e_i^{B+1} \rangle \quad (20)$$

如图1的结构所示, 本文选择将前一个残差块的输出作为下一个块的输入, 将提取的信息从一层传递到下一层. 层之间的信息传递可以帮助模型不断细化嵌入, 以更准确地建模节点偏好. 基于上述等式(20)中的残差偏好预测, 可以得到:

$$\begin{aligned} \hat{r}_{wi}^{B+1} &= \hat{r}_{wi}^B + \langle e_u^{B+1}, e_i^{B+1} \rangle \\ &= \hat{r}_{wi}^{B-1} + \langle e_u^B, e_i^B \rangle + \langle e_u^{B+1}, e_i^{B+1} \rangle \\ &= \hat{r}_{wi}^{B-1} + \langle e_u^1, e_i^1 \rangle + \dots + \langle e_u^{B+1}, e_i^{B+1} \rangle \end{aligned} \quad (21)$$

等式(21)相当于把每个层的嵌入串联起来, 形成每个节点的最终嵌入. 由于每个节点的子图都不同, 本文记录每个层的表示, 这样形成每个节点的最终嵌入信息更丰富, 从而能为模型提供更多的信息量.

### 3.4 模型训练

本文在模型中采用了基于成对排序的损失函数:

$$L = \sum_{a=1}^n \sum_{(i,j) \in D_a} -\ln(s(\hat{r}_{ai} - \hat{r}_{aj})) \quad (22)$$

其中,  $n$  是矩阵的行数,  $V$  是顶点集,  $s(x)$  是一个 sigmoid 函

数,  $R_a$  表示  $a$  正向显示反馈的项目集,  $D_a = \{(i, j) | i \in R_a \wedge j \in V - R_a\}$  表示  $a$  的成对训练数据.

## 4 实验设置和数据分析

本文提出的 NHGCN 模型使用 Pytorch 框架实现, 所有实验都在一台 CPU 为 Intel i7-9700, 显卡为 NVIDIA GTX1650, 内存为 32G 的机器上运行.

### 4.1 数据集

本文采用 3 个广泛使用的数据集进行了实验, 分别是 Cora, Citeseer, Pubmed, 这 3 个数据集都是一张无向图, 其节点代表文件(论文文献), 边代表引用关系, 标签在节点上. 节点特征是 word vector, 每一元素都是 0 或者 1, 描述每个单词在 paper 中是否存在. 3 个数据集的详细情况如表 1 所示.

表1 3 个数据集统计数据

Table 1 Statistics for three data sets

Dataset	Classes	Nodes	Edges	Features
Cora	7	2708	5429	1433
Citeseer	6	3327	4732	3703
Pubmed	3	19717	44338	500

### 4.2 评估设置

为了评估节点分类性能, 本文对数据集进行随机划分, 60% 的数据为训练集, 20% 的数据为验证集, 20% 的数据为测试集. 本文评估了噪声特征场景下的性能, 在模型训练后, 将噪声特征注入随机选择的测试节点. 默认情况下, 当数据集干净时, 所有超参数都会根据验证集的性能进行调整.

### 4.3 参数设置

对于所有基线, 模型遵循其原始论文中的最佳超参数设置, 此外, 本文将 GCNII 的最佳剩余权重  $\alpha$  调整在  $[0, 1]$  范围内, 对于 AirGNN, 本文使用两层 MLP 作为基础模型.

正则化参数  $\lambda$  为 0.01, 模型的批处理大小为 64, 模型的学习率固定为 0.01, 退出率为 0.8, 权重衰减为 0.0005, 设置步长  $\gamma = \frac{1}{2(1-\theta)}$ , 并在范围  $[0, 1]$  调整参数  $\theta$ , 为了保证稳定性, 本文使用随机划分的训练\验证\测试集进行 10 次实验, 统计平均结果.

### 4.4 比较方法

本文使用了 4 种基线方法, 包括最先进的浅层模型 GCN, 还包括 3 个最新的深度模型 JKNNet, GCNII 和 AirGNN.

GCN<sup>[1]</sup>: 提出通过谱图卷积的局部一阶近似来激励卷积结构的神经网络, 有效对图结构和节点特征进行编码.

JKNNet<sup>[8]</sup>: 提出了一种新的节点表示学习聚合方案, 可以将邻域范围分别适应于节点.

GCNII<sup>[14]</sup>: 采用初始剩余和身份映射, 有效地缓解了过度平滑问题, 提高模型的节点分类性能.

AirGNN<sup>[21]</sup>: 提出一种简单高效的传递方案, 实现特征聚合和剩余连接之间的节点自适应转换, 降低图节点特征中的自然噪声或恶意操作对模型的性能影响.

### 4.5 实验结果与分析

#### 4.5.1 NHGCN 与其他模型基线的比较

3 个数据集的分类准确率的结果如表 2 所示, 括号中的

数字对应于模型达到最优性能的层数,根据实验结果可知,本文提出的模型 NHGNC 在 3 个数据集上始终优于所有基线,表明了残差学习的意义,验证了模型 NHGNC 的有效性.

表 2 分类准确率(%)

Table 2 Classification accuracy (%)

Method	Cora	Citeseer	Pubmed
GCN	81.5	71.1	79.0
JKNet	83.2(4)	72.6(16)	79.2(8)
GCNII	83.3(4)	72.4(16)	80.2(16)
AirGNN	84.1(16)	72.8(16)	81.3(16)
NHGNC	84.9(16)	73.3(16)	81.8(16)

在 4 种基线中, JKNet, GCNII 以及 AirGNN 明显优于 GCN,这是因为 GCN 的模型层次较浅,无法对深层进行建模,节点之间的交互信息得不到有效利用,当层数超过 4 层,性能会因过度平滑问题急速下降. 在所有数据集中,模型 NHGNC 的表现始终最优, JKNet 和 GCNII 使用跳跃连接或初始剩余来处理平滑问题,忽略了噪声节点对深层建模的影响, AirGNN 则相反,考虑了噪声节点的处理,有效利用节点的邻域信息,但忽略了过平滑问题. 与之前的方法相比, NHGNC 不仅将 Frobenius 范数修改为 21 范数,增强了模型修正噪声特征的能力,而且利用残差学习保留了节点独特性,提高了节点分类性能.

4.5.2 噪声节点对 NHGNC 模型的影响

这一小节主要研究噪声节点的加入对模型性能的影响,模型会将噪声特征注入随机选择的测试节点.

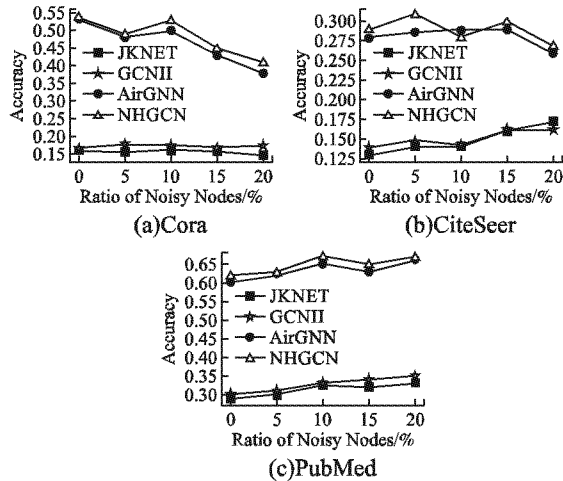


图 5 噪声节点的节点分类精度

Fig. 5 Node classification accuracy on noisy nodes

从不同噪声节点比例实验结果中可以观察到:本文提出的 NHGNC 相较于其他模型,在准确率方面有明显的提高. 这表明相比于直接利用从邻接矩阵收集到的邻居信息,除噪声后收集到的信息对节点分类更有帮助. 这验证了基于自适应残差处理来处理噪声节点的方式是科学有效的,它成功减少大量噪声节点的影响,提高了节点分类准确率.

模型在噪声节点的节点分类精度如图 5 所示, NHGNC 在噪声节点上的性能显著优于所有数据集上的所有基线,这验证了 NHGNC 能够提高对噪声特征的恢复能力.

模型在正常节点的节点分类精度如图 6 所示,当存在噪声节点时, NHGNC 提高了模型对正常节点的性能. 这是因为 NHGNC 可以去掉一些对正常节点有害的噪声特征,并对正常节点的特征进行了强化.

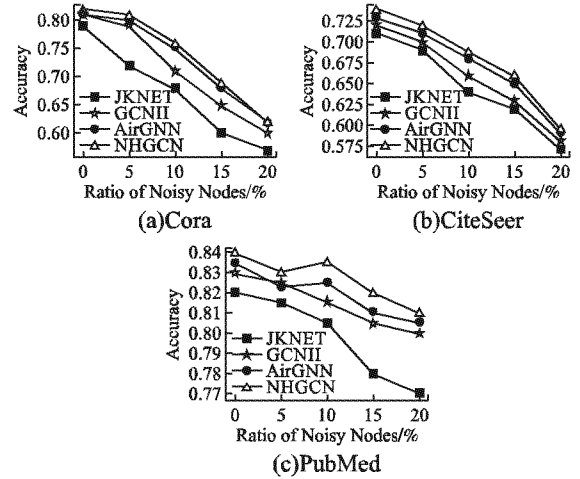


图 6 正常节点的节点分类精度

Fig. 6 Node classification accuracy on normal nodes

4.5.3 GCN 层数的影响

表 3 总结了具有不同层数的深度模型的结果. 如表 3 所示,在 3 个数据集上,随着层数的增加, NHGNC 的性能不断提高,在 16 层上实现了最佳效果. 当层数继续增加,模型的性能

表 3 不同深度的分类精度(%)结果总结

Table 3 Summary of classification accuracy (%) results for different depths

Dataset	Method	Layers			
		2	4	8	16
Cora	GCN	81.1	80.4	69.5	64.9
	JKNet	-	80.2	80.7	80.2
	GCNII	80.2	<b>83.3</b>	82.6	83.0
	AirGNN	82.1	82.7	83.4	84.1
	NHGNC	<b>82.2</b>	82.6	<b>84.2</b>	<b>84.9</b>
Citeseer	GCN	70.8	67.6	30.2	18.3
	JKNet	-	68.7	67.7	66.8
	GCNII	68.2	68.9	70.6	71.4
	AirGNN	<b>70.9</b>	71.1	72.3	72.2
	NHGNC	69.4	<b>71.6</b>	<b>73.0</b>	<b>73.3</b>
Pubmed	GCN	79.0	76.5	61.2	40.9
	JKNet	-	78.0	78.1	72.6
	GCNII	78.2	78.8	79.3	80.2
	AirGNN	78.7	79.1	80.2	81.3
	NHGNC	<b>79.2</b>	<b>80.5</b>	<b>81.4</b>	<b>81.7</b>

能提升有限,一个合理的猜想是,此时模型收集的信息已经足够丰富,而从高阶邻居收集的信息没有太大作用,给节点分类带来了噪声,干扰了节点分类的性能.

通过残差学习和噪声节点处理,本文可以真正做到深层建模,有效缓解过度平滑问题. 另一方面,当层数超过 8 层时, GCN 和 JKNet 的性能下降,这意味着它们仍然存在过度平滑问题. 根据实验结果可知,本文提出的模型 NHGNC 在 3 个数

据集上始终优于所有基线,表明了处理噪声节点和残差学习的意义,验证了模型 NHGCN 的有效性。

#### 4.5.4 过度平滑分析

这一小节主要研究过度平滑问题对模型性能的影响。本文猜想具有更高的度的节点更有可能遭受过度平滑的影响。为了验证这一猜想,本文研究了模型在 Cora 数据集上的节点分类任务中,分类精度如何随节点度而变化。更具体地说,根据每个图的节点的度数对其进行分组,用  $GCN_2$ 、 $GCN_4$ 、 $GCN_8$ 、 $GCN_{16}$ 、 $NHGCN_{16}$  表示模型在该层数的节点分类性能。在表 4 中展示了具有不同网络深度的模型针对不同节点度的平均分类精度。

表 4 分类准确率(Cora)

Degree	5	10	50	100
$GCN_2$	79.0	81.3	<b>100</b>	<b>100</b>
$GCN_4$	73.1	74.5	72.6	64.5
$GCN_8$	60.5	62.3	61.2	41.3
$GCN_{16}$	16.5	15.7	7.8	0
$NHGCN_{16}$	<b>83.3</b>	<b>84.5</b>	<b>100</b>	<b>100</b>

如表 4 所示,首先,2 层 GCN 模型的精度随着节点度的增加而增加。这是因为度数较高的节点通常从其邻居处获得更多信息,提高节点区分度,从而在节点分类任务中有更好的效果。

然而,随着网络深度的扩展,对 GCN 而言,高阶节点的精度反而会开始下降,而且下降的幅度很大,这是因为此时节点的邻居巨大,邻居节点的信息过多,导致节点自身的特征模糊不清,降低了节点的区分度,从而使得节点分类性能下降。值得注意的是,具有 16 层的 GCN 对于度数大于 100 节点的节点分类效果是 0,意味着 16 层的 GCN 无法对度数大于 100 的节点进行分类,这表明过度平滑确实会对度数更高的节点产生更大的影响,同时也验证了处理过度平滑问题的必要性。本文的模型 NHGCN 利用残差学习,保留了节点的信息,与之前的方法相比,16 层的 NHGCN 可以处理度数较大的节点,模型的性能也有较大的提升。

#### 4.5.5 消融实验

为了更好地验证噪声处理和残差学习部分的有效性,本文设计了 NHGCN 的两个变体: $NHGCN_A$  和  $NHGCN_R$ ,其中  $NHGCN_A$  去除了噪声处理模块, $NHGCN_R$  去除了残差学习模块,消融实验的结果如表 5 所示。

表 5 消融实验

Table 5 Ablation experiments

Model	Cora	Citeseer	Pubmed
	Accuracy		
$NHGCN_A$	82.1	71.5	79.6
$NHGCN_R$	81.4	70.2	79.3
$NHGCN$	<b>84.9</b>	<b>73.3</b>	<b>81.7</b>

$NHGCN$  和  $NHGCN_A$  都能取得比  $NHGCN_R$  更好的性能,证明了有效处理过度平滑问题的意义,缓解梯度消失可以保留节点的独特性,提高节点的区分度。 $NHGCN_A$  删去了噪

声处理模块,性能有所降低,说明当有一定的噪声节点时,模型的性能会被干扰,无法准确的进行节点分类,证明了除噪对节点分类的重要性。此外,当层数较少时, $NHGCN$  和  $NHGCN_A$  并没有明显的性能差异,本文猜测可能的原因是在邻域聚合时,对于低阶邻域,每个邻域都只有有限的邻域,过平滑效应不明显,而对于深层,过度平滑效果变得更加严重。最后模型  $NHGCN$  始终表现最优,说明了融合两种组件能达到最好的节点分类效果,忽略其中一个模块可能都无法达到最佳性能。

## 5 结 语

本文提出了一个结合残差网络和噪声处理的节点分类模型 NHGCN。首先,针对噪声节点,模型使用自适应残差处理方式并将 Frobenius 范数修改为 21 范数,增强了模型修正噪声特征的能力,使模型提高对噪声节点的恢复能力和对正常节点的性能。然后,模型基于残差网络,在每一层学习剩余项偏好,节点的独特性在较低层得以保留,减轻了梯度消失,有效缓解 GCN 迭代过程中的过度平滑问题。最后,将两者结合,来增强节点分类的效果。通过在 3 个数据集上的实验表明,本文提出的模型在节点分类任务上具有优势。下一步的任务将考虑在噪声处理的过程中添加对抗性节点,并研究模型在对抗性节点的节点分类精度。

## References:

- [1] Thomas N Kipf, Max Welling. Semi-supervised classification with graph convolutional networks [J]. arXiv preprint arXiv: 1609.02907, 2016.
- [2] Lecun Y, Bengio Y. Convolutional networks for images, speech, and time-series [J]. Handbook of Brain Theory & Neural Networks, 1995.
- [3] Xu F, Lian J, Han Z, et al. Relation-aware graph convolutional networks for agent-initiated social e-commerce recommendation [C]//28th ACM International Conference, 2019: 529-538.
- [4] Zhao L, Song Y, Zhang C, et al. T-GCN: a temporal graph convolutional network for traffic prediction [J]. arXiv preprint arXiv: 1811.05320, 2018.
- [5] Liu F, Cheng Z, Zhu L, et al. A2-GCN: an attribute-aware attentive GCN model for recommendation [J]. IEEE Transactions on Knowledge and Data Engineering, 2022, (9): 34-35, doi: 10.1109/TKDE.2020.3040772.
- [6] Petar Velickovic, Guillem Cucurull, Arantxa casanova, et al. Graph attention networks [J]. arXiv preprint arXiv: 1710.10903, 2017.
- [7] Li Q, Han Z, Wu X M. Deeper insights into graph convolutional networks for semi-supervised learning [J]. arXiv preprint arXiv: 1801.07606, 2018.
- [8] Xu K, Li C, Tian Y, et al. Representation learning on graphs with jumping knowledge networks [J]. arXiv preprint arXiv: 1806.03536, 2018.
- [9] Rong Y, Huang W, Xu T, et al. DropEdge: towards deep graph convolutional networks on node classification [C]//International Conference on Learning Representations, 2020.
- [10] Chiang W L, Liu X, Si S, et al. Cluster-GCN: an efficient algorithm for training deep and large graph convolutional networks [C]//

