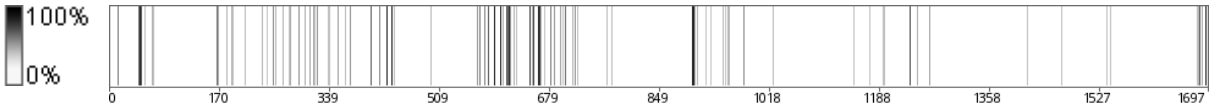
PaperPass旗舰版检测报告
简明打印版

比对结果(相似度):

总　体： 9% (总体相似度是指本地库、互联网的综合对比结果)
本地库： 7% (本地库相似度是指论文与学术期刊、学位论文、会议论文、图书数据库的对比结果)
期刊库： 4% (期刊库相似度是指论文与学术期刊库的比对结果)
学位库： 5% (学位库相似度是指论文与学位论文库的比对结果)
会议库： 0% (会议库相似度是指论文与会议论文库的比对结果)
图书库： 2% (图书库相似度是指论文与图书库的比对结果)
互联网： 3% (互联网相似度是指论文与互联网资源的比对结果)

报告编号：5EC4C61AA236C0TF3
检测版本：旗舰版
论文题目：基于深度图卷积网络的结点分类算法的研究与实现
论文作者：刘唐
论文字数：39600字符(不计空格)
段落个数：1042
句子个数：1697 句
提交时间：2020-5-20 13:54:34
比对范围：学术期刊、学位论文、会议论文、书籍数据 、互联网资源
查询真伪：http://www.paperpass.com/check

句子相似度分布图:



本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据):
暂无本地库相似资源

互联网相似资源列表：
1.相似度： 1% 标题：《神经网络中的网络优化和正则化（四）之正则化_网络...》
https://blog.csdn.net/gamer_gyt/article/details/101033364
2.相似度： 1% 标题：《深度学习——正则化 - 知乎》
https://zhuanlan.zhihu.com/p/69025058?from_voters_page=true

全文简明报告:

学号_____20165004_____　密级_____

东北大学本科毕业论文

{63%：基于深度图卷积网络的结点分类算法的研究与实现}

学　院　名　称：　软件学院

专　业　名　称：　软件工程

学　生　姓　名：　刘唐

指　导　教　师：　张伟　副教授

黄增峰　副教授

20XX　年　X　月

{63%：基于深度图卷积网络的结点分类算法的研究与实现}

作者姓名：

刘唐

校内指导教师：

张伟

副教授

校外指导教师：

黄增峰

副教授

单位名称：

软件学院

专业名称：

软件工程

东　北　大　学

20XX年X月

Research and Implementation of Deep Graph Convolutional Networks on Node Classification

by Liu Tang

Supervisor:

Associate Professor

Zhang Wei

Associate Supervisor:

Associate Professor

Huang Zengfeng

Northeastern University

June 20xx

郑 重 声 明

{68%：本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。} {93%：尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。} {97%：对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。} {74%：本学位论文的知识产权归属于培养单位。}

本人签名： 日期：

摘 要

图卷积神经网络是近年来深度学习领域新兴起的方向，在图的半监督结点分类等任务上表现突出。 深度学习的成功在于深层网络架构，然而实验研究表明，随着模型层数增加，图卷积神经网络的性能会急剧下降。

{43%：本文首先分析了深度图卷积神经网络的研究现状，对这些模型所存在的主要问题进行了阐述。} 接着重点从三个方面展开了研究，分别是过拟合、梯度消失和过光滑。过拟合和梯度消失是传统深度神经网络面临的问题，而过光滑则是深度图卷积神经网络特有的问题。 针对过拟合问题，在图卷积神经网络上引入了三种正则化方法，分别是权重衰减、提前终止和丢弃法。 针对梯度消失问题，也在图卷积神经网络上引入了三种传统方法，分别是Xavier初始化、梯度修剪和批量归一化。 实验研究表明，过拟合和梯度消失都不是限制图卷积神经网络加深的主要原因，已有的几种传统方法足以缓解这些问题， 但是无法阻止随着层数增加模型性能的骤降。 针对过光滑问题，本文不仅从理论角度进行了分析，也精心设计了实验对理论进行验证。 不同于以往用GCN研究过光滑问题，本文采用SGC进行实验研究，避免了过拟合和梯度消失的干扰。 本文基于理论分析和已有模型，从四个角度提出了缓解方法。 {41%：从图数据预处理角度，本文基于结点相似度对DropEdge进行了改进。} {46%：从控制邻居权重角度，本文基于余弦相似度对GAT进行了改进。} 从平衡局部全局角度，本文引入了残差连接和密集连接，并提出了带权重的残差连接。 {42%：从增强自身特征角度，本文提出了一种新的网络结构，在每一层引入输入层的带权重跳接。} {41%：实验研究表明，过光滑是限制图卷积神经网络加深的主要原因，本文提出的几种方法大大缓解了该问题，} 在多个数据集上取得了显著的效果。

本文系统地对图卷积神经网络无法加深这一问题进行了研究，并针对性地引入或提出了多种行之有效的方法。 但是由于图神经网络的基准数据集规模比较小，因此实验结果对模型表现的区分性还不够。

关键词： 深度图卷积神经网络； 半监督结点分类任务； 过拟合； 梯度消失； 过光滑

ABSTRACT

Graph convolutional networks are emerging directions in the field of deep learning in recent years.They are outstanding in tasks such as semi-supervised node classification of graphs. The success of deep learning lies in the deep network architecture. However, experimental research shows that as the number of model layers increases, the performance of graph convolutional networks will decrease sharply.

This paper first analyzes the research status of deep graph convolutional neural networks, and explains the main problems of these models. Then focus on research from three aspects, namely overfitting, vanishing gradient and oversmoothing. Overfitting and vanishing gradient are problems faced by traditional deep neural networks, while oversmoothing is a unique problem of deep graph convolutional networks. For the problem of overfitting, three regularization methods are introduced on the graph convolutional network, which are weight decay, early stopping and dropout. For the problem of vanishing gradient, three traditional methods are also introduced on graph convolutional neural networks, which are Xavier initialization, gradient clipping and batch normalization. Experimental studies have shown that neither overfitting nor vanishing gradient are the main reasons for limiting the depth of graph convolutional networks. There are several traditional methods that are sufficient to alleviate these problems, but they cannot prevent the model performance from dropping as the number of layer increases. Aiming at the problem of over-smoothness, we not only analyze from a theoretical perspective, but also carefully designs experiments to verify the theory. Different from previous research, we use SGC for experimental research to avoid the interference of overfitting and vanishing gradient. Based on theoretical analysis and existing models, we propose methods from four perspectives. From the perspective of graph data preprocessing, we improve DropEdge based on node similarity. From the perspective of controlling neighbor weights, we improve GAT based on cosine similarity. From the perspective of balancing local and global information, we introduce residual connections and dense connections, and propose weighted residual connections. From the perspective of enhancing its own characteristics, we propose a new network structure that introduces weighted jumpers from input layer to each layer. Experimental research shows that oversmoothing is the main reason for limiting the depth of graph convolutional networks. Several methods proposed in this paper greatly alleviate this problem

and have achieved significant results on multiple data sets.

This paper systematically studies the problem that graph convolutional networks cannot deepen, and introduces or proposes a variety of effective methods. However, because the scale of the benchmark data set of the graph neural network is relatively small, the experimental results are not sufficient to distinguish the model performance.

Key words: Deep GCN; Semi-supervised Node Classification; overfitting; vanishing gradient; oversmoothing

目 录

摘 要I

ABSTRACTIII

第1章 绪 论1

1.1 研究背景1

1.2 研究现状1

1.3 研究内容3

1.4 组织结构3

第2章 相关工作5

2.1 图卷积神经网络5

2.1.1 问题定义5

2.1.2 谱图卷积5

2.1.3 传播公式6

2.1.4 结点分类6

2.2 深度图卷积神经网络7

2.2.1 PPNP7

2.2.2 JK-Net8

2.2.3 Cluster-GCN9

2.2.4 N-GCN10

2.2.5 RGCN10

2.2.6 DeepGCN11

2.2.7 DropEdge12

2.2.8 PairNorm12

2.3 本章小结14

第3章 实验规范15

3.1 实验数据15

3.2 实验设置16

第4章 面向过拟合的方法19

4.1 问题定义19

4.2 权重衰减20

4.3 提前终止20

4.4 丢弃法21

4.5 实验分析22

4.6 本章小结25

第5章 面向梯度消失的方法27

5.1 问题定义27

5.2 Xavier初始化27

5.3 梯度修剪28

5.4 批量归一化29

5.5 实验分析30

5.6 本章小结33

第6章 面向过光滑的方法35

6.1 问题定义35

6.2 实验验证36

6.2.1 批量归一化验证36

6.2.2 过光滑理论验证36

6.3 基于图数据预处理的方法38

6.4 基于控制邻居权重的方法38

6.5 基于平衡局部全局的方法39

6.6 基于增强自身特征的方法40

6.7 实验分析41

6.8 本章小结45

第7章 总结与展望47

7.1 本文总结47

7.2 下一步工作47

参考文献49

致 谢50

第1章 绪 论

{44%：本章首先介绍深度图卷积神经网络和图上的半监督结点分类的研究背景，接着分析近年来国内外研究现状，} {68%：然后介绍本文的研究内容和主要贡献，最后给出该论文的组织结构。}

1.1 研究背景

{48%：随着训练数据的大量增长和计算资源的快速发展，深度学习在语音识别、目标检测、自然语言处理等方面取得了巨大成功。} {40%：这归功于深度学习能从欧式数据如语音、文本、图像等中提取有效的特征表示。} 但是越来越多的任务要求对非欧式数据，如引用网络、社交网络、蛋白质结构等图数据进行处理。 然而由于图的不规则、异质性、大规模等特点，传统的神经网络CNN、RNN等无法胜任。 近年来，研究人员相继提出了图递归神经网络GRNNs、图卷积神经网络GCNs、图自编码器GAEs、图强化学习GRL等模型， 在图数据处理上取得了优越的效果。

图分类、结点分类、链路预测是常见的图上的学习任务。 其中结点分类一般指半监督结点分类任务： 给定包含结点信息和结构信息的图数据集，带有标签的部分结点作为训练集，预测剩余结点的标签类别。 {41%：有研究者运用近似技巧从谱图卷积推导出图卷积神经网络的逐层传播公式，} 使得图像处理中的卷积操作能够被简单应用到图结构数据处理中，在图的半监督结点分类任务上取得了不错的表现[1]。

深度学习的成功在于深层网络架构，该架构具有更高的模型复杂度，因此也具有更强的学习能力。 {45%：此外，加深网络相比加宽网络具有逐层处理、特征变换等优点。} 在图像分类任务中，杰出的ResNet具有152层[2]。 然而研究表明，随着层数增加，GCN的性能会急剧下降。 目前对该问题的研究还较少，为什么性能会下降，如何才能加深GCN，是GCN发展面临的两个挑战[1]。

通过将关系数据自然地建模为图结构数据，GCN等基于图的深度学习模型被广泛应用于其他学科， {43%：如计算机视觉、推荐系统、自然语言处理、疾病或药物预测、基于图的NP问题等。} 对如何加深GCN的研究，能够提升模型的性能，从而促进更深入地挖掘现有图数

据的丰富价值。

1.2 研究现状

{47%：图神经网络是近年来新兴起的研究热点，对深度图卷积神经网络的研究也刚起步不久。}

理论方面，研究者们揭示了K层GCN与K步随机游走的关系[3]； {41%：证明了图卷积是一种特殊形式的拉普拉斯平滑[4]；} 分析了在图同构测试任务上GNN性能的上限[5]。

模型方面，PPNP将神经网络与传播算法分离，融入Personalized PageRank算法，在聚合时可以获取更大范围的邻居信息[6]； JK-Net以层级聚合的方式自适应地融合不同层的信息，从而平衡不同邻域的结点的局部与全局信息[3]； Cluster-GCN通过变换邻接矩阵并添加正则化，在考虑权重的同时强化邻近邻居结点的信息[7]； N-GCN在不同尺度下进行图卷积操作，最后融合所有卷积结果得到结点的特征表示， {41%：通过对不同尺寸感受野的组合提高模型的表征能力[8]；} RGCN基于第K层捕获了K-Hop邻居结点信息，这些相邻层之间存在依赖关系，使用RNN（GRU，LSTM）对层间的长期依赖建模[9]； DeepGCN借鉴CNN的成功经验，基于梯度消失/爆炸的问题，引入残差连接、密集连接和空洞卷积，在点云语义分割任务上进行了实验[10]； Dropedge在每轮训练中从图中随机删除一定比例的边，从数据增强角度缓解了过拟合，从减缓传播角度缓解了过光滑[11]； Snowball和Truncated Krylov均利用了多尺度信息，在一定条件下两种网络结构是等价的， 是谱图卷积和深度GCN在块Krylov空间下的推广形式[12]； PairNorm通过引入正则化项改进目标函数，既保证了同一类簇的结点信息趋于一致，又促进了不同类簇的结点信息差异扩大[13]。

这些模型都增强了GCN的学习能力，但是各自也存在着一些不足之处，在几个引用数据集上性能提升有限， 多数模型在超过两层后性能仍然会下降。 其中，PPNP的传播部分借鉴的是Personalized PageRank，它是基于经验假设设计的，直接进行量化计算而不需要参数学习过程，但是也失去了神经网络的优势； JK-Net只在最后一层对所有层进行融合，层之间的传播方式没有改变，较深层产生的输出仍然存在不同类簇间的结点混合问题， RGCN虽然有一些改进，但是仍然存在相似的问题； Cluster-GCN是基于矩阵的改进，当数据集庞大时，就会面临内存限制问题，而大数据集恰恰最需要更深的GCN； {43%：N-GCN、Snowball和Truncated Krylov的思路都类似于Inception，从“宽度”上对网络进行拓展，在同一层级上运行多个不同尺寸的卷积核，} {43%：但是大尺寸的卷积核不可避免地会引起过光滑问题；} DeepGCN的动机在于解决梯度消失/爆炸，但是它不是阻碍GCN加深的主要原因，同时，DeepGCN只在点云数据集上进行了实验， 该任务属于图层次的分类，每张图之间不连通，不存在过光滑问题； DropEdge采用随机割边的方法，在稀疏连接图数据上有一定效果，在密集连接图数据上却有反作用； PairNorm的正则化项扩大的是所有不相连结点对间的差异，总体来说对于缓解过光滑问题效果有限。

1.3 研究内容

文献方面，本文从理论和模型两个角度，对当前深度GCN研究的最新进展进行了总结归纳，确定了过光滑问题是限制GCN层数加深的主要原因。

模型方面，本文从两个角度展开了研究：

针对过拟合问题，在GCN上研究并实验了几种传统方法： 权重衰减weight decay、提前终止Early Stopping和丢弃法Dropout。

针对梯度消失/爆炸问题，在GCN上研究并实验了几种传统方法： Xavier初始化、梯度

修剪和批量归一化Batch Norm。

针对过光滑的问题，从图数据预处理的角度，在DropEdge的基础上做了两种改进，分别是基于结点度数的DegreeDrop和基于特征相似DistanceDrop； 从控制邻居权重的角度，利用特征的余弦相似度，经过Softmax归一化处理，作为结点与邻居间的权重； 从平衡局部全局的角度，将残差连接引入GCN网络，并提出了带可学习权重的残差连接，同时进一步引入并尝试了密集连接； 从增强结点自身的角度，在每层引入输入层的链接，并在初始特征和当前特征间赋予平衡权重。

实验方面，本文额外引入了几个密集连接的图数据集，使得实验结果更能区分模型的学习能力； 设计了对过光滑理论分析的验证实验，有力地佐证了过光滑是阻碍GCN加深的主要问题； 规范了对过光滑问题的模型的实验设置，排除了过拟合和梯度消失/爆炸对实验的混合影响。

本文的创新之处主要体现在模型和实验方面，一方面基于理论分析从不同角度计了多个有效缓解过光滑问题的有效模型， 另一方面通过改进和规范实验设置与流程，使得对深度GCN的实验研究更具有说明力。

1.4 组织结构

本文的主要研究内容为探索限制图卷积神经网络在结点分类任务上的深度的因素，并且提出相应的缓解方法。 因此以深度图卷积神经网络面临的问题以及相应的缓解方法为导向进行论文的组织结构。

第1章 绪论。 {44%：介绍深度图卷积神经网络和图上的半监督结点分类的研究背景，分析近年来国内外研究现状，} {56%：阐述本文的研究内容和主要贡献，最后介绍该论文的组织结构。}

第2章 相关工作。 {46%：介绍图卷积神经网络的理论基础，以及图上的半监督结点分类任务的定义，介绍了几种主要的深度图卷积神经网络模型，} 分析了各自的优缺点以及本文所做的改进。

第3章 实验规范。 介绍实验中用到的9个开源数据集，并说明后几章通用的一些实验设置，如数据集划分、损失函数选择，评价指标选择等。

第4章 面向过拟合的方法。 {40%：从理论角度分析了过拟合，接着引入了3种正则化方法：} {41%：权重衰减、提前终止和丢弃法，最后在引用数据集上进行了实验。}

第5章 面向梯度消失的方法。 从理论角度分析了梯度消失，接着引入了3种传统方法：{42%：Xavier初始化、梯度修剪和批量归一化，最后在引用数据集上进行了实验。}

第6章 面向过光滑的方法。 从理论角度分析了过光滑，接着设计了实验对理论进行验证，然后从四个角度提出了缓解方法： 基于图数据预处理的方法、基于控制邻居权重的方法、基于平衡局部全局的方法和基于增强自身特征的方法，最后在9个数据集上进行了实验。

第7章 总结与展望。 {57%：对本文进行了总结，并说明了下一步工作。}

第2章 相关工作

{64%：图卷积神经网络是近年来深度学习领域新兴起的方向。} {45%：本章首先介绍图

卷积神经网络的理论基础，以及图上的半监督结点分类任务的定义，接着介绍了深度图卷积神经网络的主要模型，} 并分析了各自的优缺点以及本文所做的改进。

2.1 图卷积神经网络

{43%：图卷积神经网络GCN由ChebNet近似推导而来，是一种简单有效的层式传播模型，是深度图卷积神经网络的基础[1]。}

2.1.1 问题定义

图上的半监督结点分类任务是指： 给定包含结点信息和结构信息的图数据集，将带标签的部分结点作为训练集，预测剩余结点的标签类别。 对该任务可采取的学习策略见公式（2.1）-（2.2）表述。

L=L0+λLreg#2.1

Lreg=i，jAijfXi-fXj2=fXTΔfX#2.2

其中L0表示带标签的结点的监督损失，Lreg表示图结构信息引入的损失，f(X)表示神经网络的可微分函数， {46%□λ是权重系数，X是结点特征向量矩阵，Δ=D-A表示无向图G=V，E的拉普拉斯矩阵，} A是邻接矩阵，D是度矩阵，Dii=jAij。

正则化项Lreg基于相邻结点更加相似的假设，然而该假设可能会限制模型的能力。 GCN使用神经网络模型f(X，A)直接编码图结构信息，回避了损失函数中的正则化项Lreg的使用。

2.1.2 谱图卷积

经过对称归一化后拉普拉斯矩阵为L=IN-D-12AD-12=UΛUT， 其中U是L的特征向量矩阵，Λ是L的特征值矩阵。 给定输入信号x∈RN，傅里叶域的滤波器gθ=diagθ□θ∈RN，谱图卷积见公式（2.3）表述。

gθ*x=UgθUTx#2.3

{54%：其中UTx是对x做图上的傅里叶变换，gθ可视作L的特征值得函数gθ(Λ)□} {41%：然而，大图上L的特征分解很低效，特征向量矩阵乘法的时间复杂度也较高。} {48%：这里可以用切比雪夫多项式Tk(x)近似gθ(Λ)□见公式（2.4）表述。}

gθ'Λ≈k=0Kθ'kTkΛ#2.4

这里Λ=2λmaxΛ-IN·λmax是L的最大特征值，θ'∈Rk是切比雪夫因子。 切比雪夫多项式由递推公式Tkx=2xTk-1x-Tk-2x定义，其中T0x=1，T1x=x。 将切比雪夫近似公式代入谱图卷积公式，见公式（2.5）表述。 其中L=2λmaxL-IN。

gθ'*x≈k=0Kθ'kTkLx#2.5

2.1.3 传播公式

在公式（2.5）中，令K=1，谱图卷积近似为关于L的线性函数，我们可以堆叠多层获得卷积能力； {42%：令λmax≈2，我们期望神经网络的参数在训练过程中自适应该变化。} 在这些条件下近似结果见公式（2.6）表述。

gθ'*x≈θ'0x+θ'1L-INx=θ'0x-θ'1D-12AD-12x#2.6

其中参数θ'0和θ'1可以共享于所有结点的计算。 我们引入θ=θ'0=-θ'1进一步近似，见公式（2.7）表述。 {56%：这在一定程度上可以缓解过拟合，并减少计算量。}

gθ'*x≈θIN+D-12AD-12x#2.7

这里IN+D-12AD-12的特征值范围是[0, 2]，重复该操作会导致数值不稳定等问题。为此我们引入再正则化技巧IN+D-12AD-12 D-12AD-12，其中A=A+IN，Dii=jAij。

我们将公式（2.7）进一步泛化，给定输入信号X∈RN×C，X有C个通道（即C维特征），卷积包含F个滤波器，见公式（2.8）表述。

Z=D-12AD-12Xθ#2.8

其中θ∈RC×F是滤波器的参数矩阵，Z∈RN×F是卷积操作后的信号矩阵。 在多层神经网络中，习惯上把变换后的X记做H，表示结点在隐藏层的嵌入。 {47%：此外，习惯上把神经网络的参数记做W而非θ□当前层的输出Z会作为下一层输入。} 经过这些符号替换后，得到GCN逐层传播公式，见公式（2.9）表述。

Hl+1=σD-12AD-12HlWl#2.9

2.1.4 结点分类

经过GCN处理后的输出嵌入可以用于下游任务，两层的用于半监督结点分类任务的GCN见公式（2.10）表述。 其中A=D-12AD-12。

Z=fX，A=softmax(A ReLUAXW0W1#2.10

{51%：这里W0∈RH×C是权重矩阵，在输入层和隐藏层间做线性变换，}{51%：W(1)∈RH×F也是权重矩阵，在隐藏层和输出层间做线性变换。} 变换结果经过softmax激活函数输出作为分类结果。 对于半监督结点分类任务，在带标签的样本上评估交叉熵，见公式（2.11）表述。

L= -l∈yLf=1FYlf ln Zlf#2.11

其中yL表示所有带标签的结点的集合。 {51%：通过梯度下降法我们可以训练神经网络的权重。}

2.2 深度图卷积神经网络

{40%：研究发现，超过2-3层后，随着层数增加，GCN的性能会急剧下降。} 研究者在加深GCN上做了一些尝试，提出了一些深度GCN模型。

2.2.1 PPNP

传统的GCN模型在每一层变换时包括特征变换和1阶邻居的聚合，通常只能使用有限的邻居结点信息并且难以扩展， {54%：但是边缘结点，稀疏结点等需要更多的邻居结点信息。}{41%：然而，简单地堆叠层以获取更多邻居结点信息会带来两个问题：} 一是聚合次数过多会导致过平滑，丧失了结点的局部特性； 二是堆叠层数过多会导致参数量过大，有可能造成过拟合。

受Personalized PageRank启发，研究者提出一种新的传播算法，该方法可以平衡局部性和对更大范围邻居信息的需求，从而缓解过光滑的问题。 此外将神经网络与传播过程分离，神经网络的深度完全独立于传播过程，从而回避过拟合的问题[6]。 普通的PageRank见公式（2.12）表述。

πpr=Arwπpr， Arw=AD-1#2.12

Personalized PageRank考虑了根节点ix，见公式（2.13），其中A=D-12AD-12是添加了自循环的对称归一化邻接矩阵。

πpprix=1-αAπpprix+αix#2.13

求解公式（2.13）并矩阵化后见公式（2.14）表述。 它的每个元素(yx)表示结点x对y的影响分数大小。

∏ppr=αIn-1-αA-1#2.14

为了利用上述Personalized PageRank影响分数，我们将该分数与高层特征一起用于生成每个结点的类别概率分布，见公式（2.15）表述。

ZPPNP=softmaxαIn-1-αA-1H， Hi，： =fθXi，： #2.15

{41%：其中X是结点的输入特征矩阵，H∈Rn×c是结点的隐层特征矩阵，fθ表示任意特征映射函数，} 如神经网络，fθ独立地对每个结点进行变换，该过程中不涉及聚合操作。

{42%：直接计算矩阵∏ppr的时间复杂度和空间复杂度都很高，我们可以用迭代的方法近似求解，} 见公式（2.16）-（2.18）表述。

Z0=H=fθX#2.16

Zk+1=1-αAZk+αH#2.17

ZK=softmax1-αAZk-1+αH#2.18

其中K是超参数，表示迭代轮数。 我们可以通过设置转移概率α控制邻居结点的范围，对于不同类型的图和任务选择不同的转移概率α

PPNP的传播部分借鉴的是Personalized PageRank，在该过程中聚合图的结构信息，它是基于经验假设设计的， 直接进行量化计算而不需要参数学习过程，但是也失去了神经网络的优势。 本文在增强结点自身的角度，将Personalized PageRank引入了GCN，可以同时自动学习结构信息和特征信息。

2.2.2 JK-Net

虽然GCN能适应不同结构的图数据，但是GCN固定的层级结构无法满足不同邻域结构的结点对平滑范围的要求。 K步随机游走在不同邻域结构的结点上的效果见图2.1，从左往右分别是中心结点的4步随机游走， {52%：边缘结点的4步随机游走和5步随机游走。} 可以看到，位于连接紧密的中心结点，平滑范围扩散过快； 位于连接稀疏的边缘结点，平滑范围扩散过慢，但是一旦触及中心，平滑范围就会陡增。

{46%：图2.1 K步随机游走在不同邻域结构的结点上的效果}

底层信息更具局部性，高层信息更具全局性，JK-Net以层级聚合的方式自适应的融合不同层的信息，从而平衡不同邻域结构结点的局部与全局信息[3]，见图2.2。具体的融合方式有Concatenation、Max-pooling和LSTM-attention。

Concatenation将各层输出拼接，之后作线性变换用于分类；Max-pooling将各层输出聚在一起做元素级别的最大池化操作。

LSTM-attention是最复杂的融合方式，为每层学习注意力系数，该系数表示各层的重要程度。{40%：将各层输入依次送入一个双向LSTM，将每层的前向表达和后向表达拼接后作线性变换得到一个系数，}{41%：对该系数作softmax归一化得到最终的注意力系数，最后对各层输出依据注意力系数加权求和。}

加入融合机制后，GCN就存在两种聚合方式，横向的邻居聚合学习结构信息，纵向的层级聚合促使模型有选择地学习结构信息，从而使得GCN模型能够堆叠更多层。

图2.2 JK-Net网络结构

JK-Net能自适应地根据结点的邻域结构组合不同层的信息，但是由于JK-Net只在最后一层对所有层进行融合，层之间的传播方式没有改变，较深层产生的输出仍然存在不同类簇间的结点混合问题。本文引入DenseNet对此作了改进。

2.2.3 Cluster-GCN

{55%：近距离的邻居结点比远距离的邻居结点贡献更大，}通过放大GCN中邻接矩阵的对角部分，可以在每层的聚合中对上一层的表达施加更多权重[7]，见公式（2.19）表述。

Xl+1=σ(A+IXlW(l)#2.19

然而该改进存在两个弊端：一是对所有结点使用相同的权重可能是不合理的；二是随着层数增加可能会导致数值不稳定。我们可以为原始矩阵添加自循环后再标准化，从而规避数值不稳定问题，见公式（2.20）表述。

A=D+I-1(A+I)#2.20

接着考虑到结点的权重添加正则化，见公式（2.21）表述。

Xl+1=σ(A+λdiagAXlWl#2.21

基于矩阵的实现，当数据集庞大时，就会面临内存限制问题，而大数据集恰恰最需要更深的GCN。本文基于DGL框架实现模型，采用消息发送与接收的方式进行邻居聚合和结点更新。

2.2.4 N-GCN

受Inception的启发，我们可以在不同尺度下进行卷积，最后融合所有卷积结果得到结点的特征表示，{44%：通过组合不同尺寸感受野来提高模型的表征能力[8]。}N-GCN的原理见公式（2.22）表述。

N-GCNfcA，A; Wfc，θ=softmax([GCN(A0，X; θ0，GCNA1，X; θ1□... Wfc)#2.22

N-GCN相当于采用Concatenation融合方式的JK-Net，较深层产生的输出仍然存在不同类簇间的结点混合问题。

2.2.5 RGCN

对于一个n层的GCN，第i层捕获了i-hop邻居结点的信息，相邻层之间存在依赖关系，我们可以用RNN对各层之间的长期依赖建模。 RGCN的原理见公式（2.23）-（2.24）表述。

Hl+1=RNNGNNHl，A； θl，Hl， l≥0#2.23

H0=RNNWiX+bi，0#2.24

基于门控的循环神经网络引入门控机制来控制信息的累积速度，包括有选择地加入新的信息，{48%：并有选择的遗忘之前累积的信息，有效地改善了循环神经网络的长程依赖问题，} {71%：常用的有长短期记忆网络和门控循环单元网络。}

{63%：长短期记忆网络LSTM是循环神经网络的一个变体，可以有效地解决简单循环神经网络的梯度消失/爆炸问题。} 将LSTM引入GCN，RGCN-LSTM的原理见公式（2.25）-（2.31）表述。

Xl+1=D-12AD-12HlΘl#2.25

Il+1=σ(Xl+1Wi+HlUi+biN)#2.26

Fl+1=σ(Xl+1Wf+HlUf+bfN)#2.27

Ol+1=σ(Xl+1Wo+HlUo+boN)#2.28

Cl+1=tanh(Xl+1Wc+HlUc+bcN)#2.29

Cl+1=Fl+1° Cl+Il+1° Cl+1#2.30

Hl+1=Ol+1° tanh⑩(Cl+1)#2.31

{58%：门控循环单元网络GRU是一种比LSTM网络更加简单的循环神经网络。} GRU网络引入门控机制来控制信息更新的方式。 和LSTM不同，GRU不引入额外的记忆单元。{59%：GRU引入一个更新门来控制当前状态需要从历史状态中保留多少信息，以及需要从候选状态中接受多少信息。} 将GRU引入GCN，RGCN-GRU的原理见公式（2.32）-（2.36）表述。

Xl+1=D-12AD-12HlΘl#2.32

Zl+1=σ(Xl+1Wz+HlUz+bzN)#2.33

Rl+1=σ(Xl+1Wr+HlUr+brN)#2.34

Hl+1=tanh(Xl+1Wh+Uh(Rl+1° Hl)+bhN)#2.35

Hl+1=1-Zl+1° Hl+Zl+1° Hl+1#2.36

与JK-Net在最后一层使用LSTM融合各层信息不同，RGCN使用RNN对各层之间的长期依赖建模，缓解了更深的层不同类簇结点混合的问题。

2.2.6 DeepGCN

{44%：借鉴深度CNN的经验，我们可以将残差连接、密集连接引入GCN解决由于网络加深

导致的梯度消失/爆炸问题，} 将空洞卷积引入GCN解决由于池化操作导致的空间信息丢失问题[10]。

{90%：残差网络ResNet通过给非线性的卷积层增加直连边的方式来提高信息的传播效率。}将残差连接引入GCN，ResGCN的原理见公式（2.37）描述。

Gl+1=HGl，Wl=FGl，Wl+Gl#2.37

{42%：密集网络DenseNet通过密集连接来改进信息流并重用层之间的特征。} 将密集连接引入GCN，以利用不同层的信息流，DenseGCN的原理见公式（2.38）描述。

Gl+1=HGl，Wl =TFGl，Wl，Gl =TFGl，Wl，… ，FGO，WO，GO#2.38

{68%：空洞卷积是一种不增加参数数量，同时增加输出单元感受野的方法，也称为膨胀卷积。} {100%：空洞卷积通过给卷积核插入“空洞”来变相地增加其大小。} 在特征空间上使用L2距离，根据与目标结点的距离将邻居结点排序，见公式（2.39）表述。

u1，u2，… ，uk×d#2.39

{42%：给定空洞系数d，目标结点的邻居结点见公式（2.40）表述。}

Ndν=u1，u1+d，u1+2d… ，u1+k-1d#2.40

DeepGCN的动机在于解决梯度消失/爆炸，但是它不是阻碍GCN加深的主要原因，同时，DeepGCN只在点云数据集上进行了实验， 该任务属于图层次的分类，每张图之间不连通，不存在过光滑问题。 本文在引用数据集等多个图数据集上实验了ResGCN、DenseGCN，并提出了带可学习权重的ResGCN。

2.2.7 DropEdge

DropEdge在每轮训练中随机删除图数据集中一定数量的边，在验证集和测试集上不使用DropEdge机制[11]。 {41%：具体而言，在随机矩阵A中随机选取VP个非零元素置零，其中V是原始图的总边数，} P是删除概率，最后得到邻接矩阵Adrop，见公式（2.41）描述。

Adrop=A-A’#2.41

接着对Adrop添加自循环并做对称归一化，将得到的结果Adrop代替GCN中的A。 我们可以让所有层共享同一个Adrop，也可以在每一层进行DropEdge，在第l层得到邻接矩阵A(l)drop，这样可以赋予原始数据更多随机性。

从数据增强角度，DropEdge在训练中不断随机删除原始图的边，增强了输入数据的随机性和多样性，从而缓解了过拟合问题。 从消息传递角度，GCN中结点间通过连边进行消息传递，随机删除一些边可以使得结点连接变稀疏，从而缓解了过光滑问题。

在基于结点采样的方法DropNode中，删除某个结点相当于删除了与该结点相连的所有边，可以视为DropEdge的特殊形式。 与DropNode相比，DropEdge是面向边的，保留了所有结点的特征，更具灵活性。 此外DropNode对所有边的采样是并行的，更具高效性。

Dropout是一种正则化方法，在训练中随机丢弃一部分神经元，即随机将特征向量的部分维度置零， 与DropEdge相比，它可以缓解过拟合但是不能缓解过平滑。 图稀疏性通过复杂的优化算法删掉部分边来压缩图，与DropEdge相比时间复杂度往往很高。

DropEdge是一种简单而高效的方法，但是实验发现，在引用数据集等稀疏图上表现良好，在其他几个密集图数据集上却起到了反作用，　　这可能是因为相当一部分有效边被随机删除了。　　本文基于DropEdge做了一些改进，分别是基于结点度数的DegreeDrop和基于特征相似DistanceDrop。

2.2.8　PairNorm

{43%：给定X∈Rn×d表示结点的新特征矩阵，其中xi∈Rd表示特征矩阵X的第i行，}　图上的正则化最小平方GRLS优化问题见公式（2.42）描述[14]。

minXi∈Vxi-xiD2+i，j∈Exi-xj22#2.42

这里运算ziD2=ziTDzi，第一项可以看做带度数权重的最小平方，第二项表示图结构上新特征之间的差异。　　该优化问题的目标在于保证新特征与原特征相似，同时促使新特征在图上更光滑。

GRLS问题有解析解X=(2I-Arw)-1X，其中Arw=D-1A，ArwX是一阶泰勒近似，即ArwX≈X。用Asym=D-12AD-12代替Arw，也就是X=AsymX≈X。　　因此，图卷积是GRLS问题的近似解。

理想情况下，我们希望同一类簇类更光滑，同时不同类簇间不光滑，但是公式（2.42）只能确保前者。　　为了同时实现两个目标，我们可以在公式（2.42）中增加一项，该项表示不相连结点对之间的距离总和，见公式（2.43）描述。

minXi∈Vxi-xiD2+i，j∈Exi-xj22-λi，j∉Exi-xj22#2.43

其中系数λ用于平衡两个目标的重要程度。　　我们可以求出公式（2.43）的解析解，并用图卷积近似，从而得到带系数λ的新的图卷积操作，但是这样不具备通用性。

{53%：给定图卷积的输出X作为PairNorm层的输入，X是PairNorm层的输出。}　图卷积X=AsymX只实现了第一个目标，PairNorm作为正则化层，通过增加不相连结点对间的总距离来实现第二个目标。　　结点对间的总距离记做TPSD，PairNorm确保TPSD(X)=TPSD(X)，见公式（2.44）描述。

i，j∈Exi-xj22+i，j∉Exi-xj22=i，j∈Exi-xj22+i，j∉Exi-xj22#2.44

随着图卷积操作的不断平滑，i，j∈Exi-xj22和i，j∈Exi-xj22越来越接近，　　因此i，j∉Exi-xj22和i，j∉Exi-xj22也越来越接近。　　TPSD(X)是与数据集特性相关的常数，记做超参数C。

为了对X进行正则化，我们需要计算TPSD(X)，然而对于大数据集直接计算时间复杂度很高。TPSD(X)的等价形式见公式（2.45）描述。

TPSDX=i，j∈nxi-xj22　=2n21ni=1nxi22-1ni=1nxi22#2.45

为进一步简化，将xi中心化，第二项为0，TPSD的值不变。　　具体地，PairNorm分为两步，中心化见公式（2.46）描述[13]，缩放化见公式（2.47）描述。

xic=xi-1ni=1nxi#2.46

xi=s·xic1ni=1nxic22=sn·xicXcF2#2.47

PairNorm具有坚实的理论基础，但是由于扩大的是所有不相连结点对间的差异，总体来说对于缓解过光滑问题效果有限， 在受过光滑较严重的密集连接图数据集上表现不佳。 本文提出的基于DropEdge的改进方法直接针对过光滑问题，取得了较好的效果。

2.3 本章小结

{44%：本章首先介绍了图卷积神经网络的理论基础和模型详情，以及GCN在图的半监督结点分类任务上的应用，} 接着介绍了8种主要的深度图卷积神经网络模型，讨论分析了它们各自的优缺点以及本文的主要工作以及在已有工作上的改进。

第3章 实验规范

本章介绍了实验中用到的9个开源数据集，并说明了后几章通用的一些实验设置，如数据集划分、损失函数选择，评价指标选择等。

3.1 实验数据

本文使用9个开源图数据集验证提出的方法和模型[15]。 这些数据集的详细信息如表3.1所示。

表3.1 图数据集

Dataset

Nodes

Edges

Features

Classes

Cora

2708

5429

1433

7

Cite.

3327

4732

3703

6

Pubm.

19717

44338

500

3

Cham.

2277

36101

2325

4

Squi.

5201

217073

2089

4

Actor

7600

33544

931

4

Corn.

183

295

1703

5

Texa.

183

309

1703

5

Wisc.

251

499

1703

5

Citiation networks: Cora、Citeseer和Pubmed是3个标准的引用网络基准数据集。{50%：在引用网络中，结点表示论文，边表示论文之间的引用关系。} 结点特征是论文的词袋模型表示，结点标签是论文的学术主题。

WebKB: WebKB是从各大学计算机系收集的网页数据集，我们使用了它的3个子集：Cornell、Texas和Wisconsin。 {56%：在WebKB数据集中，结点表示网页，边表示网页之间的超链接关系。} 结点特征是网页的词袋模型表示。 网页被人为分成5类：student、project、course、staff和faculty。

Actor co-occurrence network: 该数据集是电影-导演-演员-编剧网络的诱导子图，只包含了演员。 结点表示演员，边表示演员在同一维基百科页面的共现关系。{52%：结点特征表示维基百科页面的某些关键词。} 我们人为将其分为4类。

Wikipedia network: Chameleon和Squirrel是维基百科中特定主题下的page-page网络。 {63%：结点表示网页，边表示网页之间的相互链接关系。} {45%：结点特征是维基百科页面中的一些信息量丰富的名词。} 我们人为将其分为4类。

可以看到，Cora、Citeseer和Pubmed边比较少，连接比较稀疏。
而Chameleon、Squirrel和Actor边比较多，连接比较密集，因此过光滑问题也更严重。

3.2 实验设置

数据集划分

{60%：对于所有图数据集，我们按照60%、20%、20%的比例将其划分为训练集、验证集和测试集。} {58%：其中训练集用于训练模型，验证集用于超参数寻优、测试集用于评估模型。}

对于过拟合和梯度消失/爆炸问题，我们只使用引用数据集验证模型。 对于过光滑问题，我们使用所有数据集验证模型。

损失函数

我们用交叉熵损失函数来训练模型。 {69%：交叉熵损失函数一般用于分类问题。}
假设样本的标签y∈{1，… ，C}为离散的类别，模型fx； {69%θ∈[0，1]C的输出类别

为类别标签的条件概率分布，即py=cx;} θ=fc(x; θ)□并满足fcx; θ∈0，1，c=1Cfcx; θ=1□ 我们可以用一个C维的one-hot向量y来表示样本标签。 {90%：假设样本的标签为k，那么标签向量y只有第k维的值为1，} {71%：其余元素的值都为0.标签向量y可以看做样本标签的真实条件概率分布pr(y x)，} 即第c维是类别为c的真实条件概率。对于两个概率分布，一般可以用交叉熵来衡量它们的差异，标签的真实分布y和模型预测分布f(x; θ)之间的交叉熵见公式（3.2）描述。

Ly，fx; θ=-yTlogfx; θ =-c=1Cyclogfc(x; θ)#3.2

评价指标

{100%：对于分类问题，常见的评价标准有准确率、精确率、召回率和F值等。} 给定测试集T={x1，y1，… ，xN，yN}，假设标签yn∈{1，… ，C}，用学习好的模型f(x; {47%□θ*)对测试集中的每一个样本进行预测，结果为{y1，…} ，yN}。 我们用准确率来评价模型，见公式（3.3）描述，其中I(·)为指示函数。

A=1Nn=1NI(yn=yn

超参数寻优

{77%：我们用网格搜索进行超参数寻优。} {86%：网格搜索是一种通过尝试所有超参数的组合来寻址合适一组超参数配置的方法。} {63%：假设总共有K个超参数，第k个超参数可以取 mk个值，那么总共的配置组合数量为m1×m2×…} ×mK。 {91%：网格搜索根据这些超参数的不同组合分别训练一个模型，然后测试这些模型在验证集上的性能，选取一组性能最好的配置。}

激活函数

{55%：常用的非线性激活函数有Sigmoid、ReLU等，原始的GCN采用ReLU作为激活函数。}然而研究表明，由于过光滑问题，随着层数加深，在GCN中Tanh函数更有利于保持特征列之间的线性无关性[12]， 效果比ReLU要好，因此本文也采用Tanh作为激活函数。

优化算法

常用的优化算法有动量法、Nesterov加速梯度、RMSprop算法、Adam算法等[16]。 Adam算法是动量法和RMSprop算法的结合，不仅使用动量作为参数更新方向，而且可以自适应调整学习率。 {46%：本文统一采用Adam优化算法，初始学习率设置为1e-2。}

第4章 面向过拟合的方法

过拟合是限制传统神经网络加深的问题，本章首先对该问题进行了理论分析，接着引入了3种正则化方法： {41%：权重衰减、提前终止和丢弃法，最后在引用数据集上进行了实验。}

4.1 问题定义

过拟合可以形式化地定义为： 给定一个假设空间F，一个假设f属于F，如果存在其他的假设f’也属于F， 使得在训练集上f的损失比f’的损失小，但在整个样本空间上f’的损失比f的损失小， {80%：那么就说假设f过度拟合训练数据[17]。}

我们可以用期望风险R(θ)衡量模型fx; θ的好坏，见公式（4.1）表述。 其

中L(y，fx; θ)是损失函数，描述了两个变量的差异。 pr(x，y)是数据的真实分布。

Rθ=Ex，y~prx，yLy，fx; θ#4.1

一般来说，期望风险越小，表示模型fx; θ越优秀。 但是我们无法获悉数据的真实分布和映射函数，因此也无法计算模型的期望风险R(θ)。 给定一个训练集D={xn，yn}n=1N，我们可以计算训练集上的平均损失，也就是经验风险，见公式（4.2）表述。

RDempθ=1Nn=1NLyn，fxn; θ#4.2

因此，我们可以采用经验风险最小化原则作为学习准则，也就是找到一组使得经验风险最小的参数θ*。见公式（4.3）表述。

θ*=argminθRDempθ#4.3

{81%：根据大数定理理论，当训练集的规模趋向于无穷大时，经验风险也会趋向于期望风险。} {52%：但是实际情况是，我们一般无法获取足够数量的样本。} 训练样本往往是从真实数据采样的一个非常小的子集，并且该子集中通常会包含一些噪声数据。 {64%：因此训练样本不能很好的反映数据的真实分布。} {46%：经验风险最小化常常会导致过拟合，即在训练集上的准确率很高，但是在测试集也就是未知数据上的准确率很低。}

导致过拟合问题的原因有训练数据较少，数据包含噪声，模型能力过强等。 {80%：正则化是一类通过限制模型复杂度，从而缓解过拟合提高模型泛化能力的方法。} {56%：常用的正则化方法有权重衰减、提前终止、丢弃法等。}

4.2 权重衰减

权重衰减通过在每次更新参数时引入一个衰减系数限制模型复杂度，从而缓解过拟合问题，是一种有效的正则化方法，见公式（4.4）表述。 {83%：其中β为权重衰减系数，一般取值比较小。} α为学习率，gt为第t步更新时的梯度。

θt 1-βθt-1-αgt#4.4

{91%：在标准的随机梯度下降中，权重衰减正则化和L2正则化效果相同。} {60%：因此，在一些深度学习框架中权重衰减通过L2正则化来实现。} {82%：L1和L2正则化是机器学习中最常用的正则化方法，通过约束参数的} {81%：L1和L2范数来减小模型在训练数据集上的过拟合现象。} {45%：通过加入L1和L2正则化，优化问题见公式（4.5）描述。}

θ*=argminθ1Nn=1NLyn，fxn; θ+λLpθ#4.5

{62%：这里N是训练样本的数量，L(·)是损失函数，f(·)为待学习的模型，} θ是它的参数，Lp是范数函数，p表示范数的类型，取值为{1, 2}时表示L1和L2范数，λ是正则化系数。{41%：带正则化的优化问题可以转化为带约束条件的优化问题，见公式（4.6）-（4.7）表述。}

θ*=argminθ1Nn=1NLyn，fxn; θ#4.6

Lpθ≤1#4.7

对于给定的特征向量x=x1，x2，… ，xnT，其p范数见公式（4.8）表述。

xp=x1p+x2p+…+xnp1p#4.8

4.3 提前终止

{83%：由于深度神经网络的拟合能力很强，因此特别容易在训练集上过拟合。} {62%：在梯度下降优化的过程中，我们可以用验证集上的错误代替期望错误，当验证集的错误率不再下降，就停止模型的迭代。} {45%：验证集也叫开发集，常用于超参数寻优。} 验证集的错误率变化不一定是平缓曲线，可能会在某处先升高再降低，因此，我们需要根据实际任务进行优化，选取恰当的早停窗口。

具体而言，提前终止主要有三种停止标准。

第一类停止标准是指，当泛化损失超过指定阈值时停止训练，泛化损失见公式（4.9）表述， {47%：它表示的是当前迭代周期中，泛化误差相对目前最小误差的增长率。} {52%：的其中Eva(t)表示第t次迭代时验证集的误差，描述的是泛化误差，}
Eoptt=minEvat'，t'≤t表示迭代t次后取得的最小的验证集误差。

GLt=100·EvatEoptt-1#4.9

第二类停止标准基于一个假设： {44%：过拟合出现在训练集误差降低很慢的时候。}
也就是训练集误差依然下降很快时，泛化误差可能会在未来被修正。 给定一个周期k，度量进展见公式（4.10）描述。

Pkt=1000·t'=t-k+1tEtrt'k·mint'=t-k+1tEtrt'-1#4.10

度量进展描述了在某段时间内训练集误差的平均下降情况。 {50%：当训练过程不稳定时，该变量的值可能会很大。} 训练了较长时间后，该变量会趋向于0。 因此，引入第二类停止标准，泛化损失和度量进展的比值超过指定阈值时停止训练，该比值见公式（4.11）描述。

PQα=GLtPkt#4.11

{40%：第三类停止标准完全基于泛化误差的变化，在连续k个周期内泛化误差持续增长时停止训练。} 该停止标准可以用作剪枝算法。

4.4 丢弃法

通过随机丢弃一定比例的神经元，从而缓解深度神经网络在训练集上过拟合，这就是丢弃法[18]。 {64%：给定神经网络层y=f(Wx+b)，我们可以引入掩蔽函数mask(·)，得到y=f(Wmaskx+b)。} 掩蔽函数mask(·)的定义见公式（4.12）描述。

maskx=m x 训练阶段px 测试阶段#4.12

这里m∈0，1D是通过概率为p的的零一分布生成的丢弃掩码。 {72%：在训练阶段，激活的神经元的平均数量只有原来的比例p。} 在测试阶段，所有神经元都被激活。 {62%：因此训练和测试阶段神经网络的输出不一致。} 我们可以在测试阶段将神经层输入x乘上p，从而缓解该问题。 我们可以用验证集来选取一个最优的保留率p。 一般来说，将隐藏层的保留率设置为p=0.5最好，这适用于大多数网络和任务。 {72%：当p=0.5时，在训练阶段丢弃了一半的神经元，只剩下一半的神经元可以激活，} {46%：相当于不同的神经网络的平均，更具多样性。} {65%：对于输入层的神经元，通常将保留率设置为近似1的值，这样输入的变化不会太大。} 丢弃输入层的神经元，相当于在数据中增加噪声，训练出的模型更具

鲁棒性。

{66%：从集成学习的角度，每丢弃一次神经元，相当于从原始网络采样一个子网络。}{56%：对于一个有n个神经元的网络，一共可以采样出2n个子网络，这些子网络共享原始网络的参数，}　{41%：最终训练得到的网络相当于指数级别数量的网络的组合模型。}

4.5　实验分析

权重衰减

{42%：实验中采用了L2正则化，层数为[1，8]时正则化系数设置为10-3，}　层数为[9，16]时正则化系数设置为10-2。　[2，4，8，16]层时GCN的准确率见表4.1。　其中GCN(WD)表示使用了权重衰减的GCN。

表4.1　权重衰减方法的实验结果

数据集

模型

2层

4层

8层

16层

Cora

GCN

87.15

85.94

86.35

26.51

GCN(WD)

85.94

87.55

85.94

42.17

Citeseer

GCN

76.42

75.94

72.64

63.21

GCN(WD)

76.42

75

75

72.64

Pubmed

GCN

86.87

85.6

84.43

59.99

GCN(WD)

85.55

86.11

84.69

70.59

可以看到，当层数为2时，权重衰减方法会起到反作用，此时会造成模型轻微的欠拟合。{43%：当层数为[4, 8, 16]时，模型的参数增多，学习能力变强，产生了过拟合问题。}{46%：此时权重衰减方法降低了模型的复杂度，缓解了过拟合问题。} 特别地，当层数达到16层时，权重衰减方法的作用非常明显。

图4.1 Pubmed上权重衰减方法的实验结果

Pubmed的数据规模最大，更能区分模型的有效性。 我们在该数据集上实验了[1, 16]层的GCN采用权重衰减方法的效果，见图4.1。 可以看到，当不使用权重衰减方法时，随着层数加深性能下降，使用权重衰减方法后得到缓和。 {43%：值得注意的是，当层数超过14层

后，训练集和测试集的准确率都会骤降，} 这说明还有其他因素阻碍着GCN的加深，这就是后面会讲到的过光滑问题。

表4.2 提前终止方法的实验结果

数据集

模型

2层

4层

8层

16层

Cora

GCN

85.14

83.94

85.14

26.51

GCN(ES)

87.15

85.94

86.35

26.51

Citeseer

GCN

69.81

67.92

68.4

62.81

GCN(ES)

76.42

75.94

72.64

63.21

Pubmed

GCN

86.46

84.94

84.08

40.37

GCN(ES)

86.87

85.6

84.43

59.99

提前终止

实验中采用了第三类停止标准，以准确率作为早停指标。 在其他方法的实验中，我们用该标准作为剪枝算法，以节省不必要的训练开销。 不使用提前终止方法时，训练轮数设置为400轮。 使用提前终止方法时，训练轮数设置为400轮，变化窗口设置为50轮。 [2，4，8，16]层时GCN的准确率见表4.2。 其中GCN(ES)表示采用了提前终止方法的GCN。

图4.2 Pubmed上提前终止方法的实验结果

可以看到，在所有数据集上，在任意层数GCN上，采用了提前终止方法都会有一定的性能提升。 即使在两层的GCN上，如果不采用正则化方法，随着训练轮数的不断增加，最终也会产生过拟合问题。 由于Pubmed的结点的特征向量的维数比较小，当层数过多时过拟合会更明显。

同样地，我们在规模最大的Pubmed数据集上实验了[1，16]层的GCN，见图4.2。 准确率随层数增加的整体走势与图4.1相似。 由于在进行其他方法的实验时，都采用了提前终止作为辅助手段，所以图4.2的整体准确率不高。

表4.3 丢弃法的实验结果

数据集

模型

2层

4层

8层

16层

Cora

GCN

87.15

85.94

86.35

26.51

GCN(DO)

88.76

87.15

87.95

33.33

Citeseer

GCN

76.42

75.94

72.64

63.21

GCN(DO)

75.94

75.94

74.53

76.42

Pubmed

GCN

86.87

85.6

84.43

59.99

GCN(DO)

88.44

85.85

84.94

82.25

提前终止

实验中的保留率统一设置为0.5。 层数为[2，3，8，16]的GCN的实验结果见表4.3。 其中GCN（DO）表示采用了丢弃法的GCN。 可以看到，相比较其他两种正则化方法，丢弃法的效果最好，GCN的性能有很大提升。

图4.3 Pubmed上丢弃法的实验结果

我们也在Pubmed数据集上实验了[1，16]层的GCN，实验结果见图4.3。 与表4.3显示的情况有所出入，采用了丢弃法的GCN在层数为[10，15]时性能出现了波动， 在12层时甚至大幅下降，这可能是因为丢弃法涉及了随机过程。 通过设置较大的早停窗口，增加实验次数取均值等方法，我们可以获得更加准确的实验结果。

4.6 本章小结

本章首先介绍了过拟合问题的具体含义，接着介绍了三种常用的正则化方法： 权重衰减、提前终止和丢弃法，并在GCN上做了一些实验。 实验表明，过拟合问题也是限制GCN加深的一个因素，传统的正则化方法可以用于缓解该问题。 但是实验发现，即使缓解了过拟合，当GCN层数超过14层后，训练集和测试集的准确率都会骤降， 这与过拟合现象不符，限制GCN加深的关键因素不是过拟合。

第5章 面向梯度消失的方法

梯度消失是限制传统神经网络加深的问题，本章首先对该问题进行了理论分析，接着引入了3种传统方法： {42%：Xavier初始化、梯度修剪和批量归一化，最后在引用数据集上进行了实验。}

5.1 问题定义

{48%：在神经网络中误差反向传播的迭代公式见公式（5.1）表述。}

δl=fl'zl Wl+1Tδl+1#5.1

{100%：误差从输出层反向传播时，在每一层都要乘以该层的激活函数的导数。} 当我们使用Sigmoid型函数，Logistic函数σ(x)或Tanh函数时，其导数见公式（5.2）-（5.3）表述。

σ'x=σx1-σx∈0， 0.25#5.2

tanh'x=1-tanhx2∈0，1#5.3

{74%：Sigmoid型函数的导数的值域都小于或等于1。} {100%：由于Sigmoid型函数的饱和性，饱和区的导数更是接近于0。} 这样，误差经过每一层传播都会不断衰减。 {94%：当网络的层数很深时，梯度就会不停衰减，甚至消失，使得整个网络很难训练。} 这就是梯度消失问题。 {43%：除了激活函数的导数，神经网络的参数的初始值也会导致梯度消失问题。} 类似地，还有梯度爆炸问题，统称梯度消失/爆炸问题。

5.2 Xavier初始化

{43%：给定神经网络第l层的神经元a(l)，其输出值见公式（5.4）表述。} 其中ail-1，1≤i≤Ml-1为前一层Ml-1个神经元的输出 。 f(·)是激活函数，wil是学习参数。

al=f(i=1Ml-1wilail-1#5.4

假设f(·)为恒等激活函数，wil和ail-1相互独立且均值为0，那么al的均值见公式（5.5）描述。

Eal=E[i=1Ml-1wilail-1] =i=1Ml-1EwilEail-1 =0#5.5

同样地，我们可以推导出al的方差，见公式（5.6）描述。

varal=vari=1Ml-1wilail-1 =i-1Ml-1varwilvarail-1 =Ml-1varwilvarail-1#5.6

可以看到，输入信号的方差被神经元缩放了Ml-1varwil倍。 通过使每个神经元的输入与输出的方差尽可能保持一致，确保输入信号在经过许多层网络后不被过分缩放[19]。 我们可以将Ml-1varwil设置为1，见公式（5.7）表述。

varwil=1Ml-1#5.7

在反向传播过程中，误差信号也会被缩放，为此我们可以采用同样的方法，见公式（5.8）表述。

varwil=1Ml#5.8

{40%：同时考虑前向传播和反向传播过程中信号的缩放，见公式（5.9）表述。}

varwil=2Ml-1+Ml#5.9

计算出参数的约束方差后，我们可以通过均匀分布或正态分布对其进行随机初始化。如果采用正态分布，可以按N(0，2/(Ml-1+Ml))进行初始化。 如果采用均匀分布，可以按[-r，r]进行初始化，其中r的取值见公式（5.10）表述。 上述方法就是Xavier初始化。

r=6Ml-1+Ml#5.10

{40%：神经元的参数和输入的绝对值一般比较小，处于Logistic函数和Tanh函数的线性区间，此时他们可以近似为线性函数，也可以使用Xavier初始化。} 在实际使用中，根据使用的激活函数，通常将方差var(wil)乘以一个缩放因子ρ□

5.3 梯度修剪

梯度修剪主要用于缓解梯度爆炸问题。 在梯度下降中，如果梯度骤增，用大梯度更新参数会使得其远离最优点。 梯度修剪通过将梯度的模限制在一个区间内来缓解该问题。主要有两类修剪方式[17]。

一类是按值修剪。 给定区间[a，b]，如果参数的梯度超过b，将其设置为b； 如果参数的梯度小于a，将其设置为a，见公式（5.11）表述，其中gt是第t次迭代时参数的梯度。

gt=maxmingt，b，a#5.11

一类是按模修剪。 按模修剪通过将梯度的模限制为一个给定的阈值b来缓解该问题，见公式（5.12）表述。

gt=gt， gt2≤bbgtgt， gt2>b#5.12

阈值b是超参数，一般设置为一个较小的值就可以取得不错的结果。

5.4 批量归一化

批量归一化是逐层归一化方法的一种。 逐层归一化是传统机器学习中的一种数据归一化方法，通过对隐藏层的输入进行归一化，从而使网络的训练更加容易[20]。

{43%：给定激活函数f·，可学习参数W和b，第l层的净输入zl，第l层神经元的输出见公式（5.13）表述。}

al=fzl=fWal-1+b#5.13

通过保持净输入zl的分布一致，我们可以提高优化的效率，例如将zl归一化为标准正态分布。 {40%：在实践中，一般在仿射变换后，激活函数前进行归一化操作。} {40%：我们可以使用标准化将zl的每个维度归一化为标准正态分布，见公式（5.14）表述。}

zl=zl-Ezlvarzl+ε#5.14

这里E[zl]和var(zl)是指在当前参数下，在整个训练集上，zl的每个维度的期望和方差。{45%：但是在小批量随机梯度下降法中，无法准确地计算zl的期望和方差。} 因此，我们只能用小批量样本集近似估计，见公式（5.15）-（5.16）表述。 {45%：其中K为小批量样本集合的容量，zk，l为第l层神经元的净输入，μB和σB2为均值和方差。}

μB=1Kk=1Kzk，l#5.15

σB2=1Kk=1Kzk，l-μB zk，l-μB#5.16

经过标准归一化后，zl的取值会集中在0附近，该取值区间是一些激活函数的近似线性变换区间，削弱了神经网络的非线性能力。 因此，我们附加一个缩放和平移变化操作来修正

取值区间，见公式（5.17）表述。

zl=zl-μBσB2+ε γ+β#5.17

这里γ和β分别是缩放和平移参数。 当γ=σB2，β=μB时，zl=zl。 批量归一化可以作为一个神经层，作用在激活函数之前，见公式（5.18）表述。 {54%：批量归一化包含了平移变换，因此仿射变换不再需要偏置参数。}

αl=fBNγ□βzl#5.18

需要注意的是，小批量样本的均值和方差是变化的，在计算梯度时需要考虑该影响。一般我们可以用移动平均代替计算。

批量归一化不仅可以提高优化效率，也能起到正则化方法的作用，使得模型不会在某个特定样本上过拟合。

5.5 实验分析

Xavier初始化

实验中GCN采用tanh激活函数，所以需要将方差乘以一个缩放因子。 [2，4，6，8]层GCN的准确率见表5.1，其中GCN（Xa）表示使用了Xavier初始化的GCN。 可以看到，使用了Xavier初始化后，GCN的性能有一定提升。 即使是浅层的GCN，也得益于恰当的初始值，分类性能有所增强。 {41%：但是当层数达到16层时，GCN的性能大幅下降，这Xavier初始化起到的作用有限。}

图5.1 Pubmed上Xavier初始化方法的实验结果

同样地，我们在规模最大的Pubmed数据集上实验了[1，16]层的GCN，见图5.1。 {58%：总体而言，Xavier初始化起到了一定的效果。} 但是注意到在[13，16]层，即使使用了Xavier初始化，GCN的性能仍会骤降。 并且在该区间内，GCN的性能发生了抖动，这可能是由于Xavier初始化本身的随机性经过了GCN多层放大。

Xavier初始化是一种比较实用且常用的工程技巧，在其他实验中，我们同样采用它作为一种辅助手段。

表5.1 Xavier初始化方法的实验结果

数据集

模型

2层

4层

8层

16层

Cora

GCN

86.35

83.13

86.35

30.92

GCN(Xa)

87.15

89.16

85.14

30.92

Citeseer

GCN

76.89

74.06

71.7

65.28

GCN(Xa)

79.25

76.89

75.94

64.32

Pubmed

GCN

86.56

85.5

84.03

44.02

GCN(Xa)

86.41

85.8

83.37

54.87

梯度修剪

{43%：实验中采用按模修剪，使用L2范数，阈值b设置为2。} [2，4，6，8]层GCN的准确率见表5.2，其中GCN（GC）表示使用了梯度修剪的GCN。 可以看到，对于浅层GCN，此时不存在梯度消失/爆炸问题，梯度修剪会导致信息损失，因此GCN的性能有所下降。 当层数达到16层时，梯度消失/爆炸问题较明显，梯度修剪的增益性有所体现。

图5.2 Pubmed上梯度修剪方法的实验结果

同样地，我们在规模最大的Pubmed数据集上实验了[1，16]层的GCN，见图5.2。 不使用梯度修剪时，层数达到12层时训练集和测试集的准确率都发生了骤降，此时可能发生了严重得梯度消失/爆炸问题。 使用了梯度修剪后，GCN的性能相对有了大幅提升，进一步验证了梯度消失/爆炸问题的存在。 GCN层数较少时，该问题不明显，梯度修剪的作用有限。 当层数为16层，即使使用了梯度修剪，GCN的性能还是骤降，可能还存在其他因素限制着GCN加深。

表5.2 梯度修剪方法的实验结果

数据集

模型

2层

4层

8层

16层

Cora

GCN

87.15

89.16

85.14

30.92

GCN(GC)

87.15

85.94

85.54

69.08

Citeseer

GCN

79.25

76.89

75.94

64.32

GCN(GC)

76.42

75.94

72.64

71.23

Pubmed

GCN

86.41

85.8

83.37

54.87

GCN(GC)

86.87

85.6

84.43

60.9

批量归一化

实验中，ε的值设置为10-5，移动平均的动量值设置为0.1，缩放和平移变量为可学习参数。由于GCN的几个基准数据集规模都比较小，所以实际上计算的是整个训练集上的均值和方差。[2，4，6，8]层GCN的准确率见表5.2，其中GCN（BN）表示使用了批量归一化的GCN。 可以看到，在Cora和Citeseer数据集上，GCN层数较少时，批量归一化产生了负面影响。 在Pubmed数据集上，批量归一化的表现最好。 当层数达到16层时，批量归一化在所有数据集上都对GCN有增益。

图5.3 Pubmed上批量归一化方法的实验结果

同样地，我们在规模最大的Pubmed数据集上实验了[1，16]层的GCN，见图5.3。 采用了批量归一化后，准确率的走势表现得非常好。 即使当层数达到16层时，GCN的性能也没有骤降。 在第6章中，我们会在多个数据集上，更大的层数区间上，进一步探究批量归一化。实验表明，当层数进一步加深时，使用了批量归一化的GCN在多个数据集上性能仍会骤降。此处在Pubmed上表现较好的原因是，引用网络结点间连接比较稀疏，过光滑不是特别严重。而批量归一化隐含数据增强的效果，在一定程度上起到了图数据预处理的作用，对过光滑有一些缓解

表5.2 梯度修剪方法的实验结果

数据集

模型

2层

4层

8层

16层

Cora

GCN

87.15

89.16

85.14

30.92

GCN(BN)

77.51

77.51

81.93

85.54

Citeseer

GCN

79.25

76.89

75.94

64.32

GCN(BN)

67.92

61.79

70.28

73.58

Pubmed

GCN

86.41

85.8

83.37

54.87

GCN(BN)

87.73

85.45

83.92

83.57

5.6 本章小结

本章首先介绍了梯度消失/爆炸问题的具体含义，接着介绍了三种常用的缓解该问题的方法： Xavier初始化，梯度修剪和批量归一化，并在GCN上做了一些实验。 实验表明，梯度消失/爆炸也是限制GCN加深的一个因素，传统的几种方法可以缓解该问题。 但是当层数增加到一定程度时，GCN性能仍然会骤降，限制GCN加深的主要因素不是梯度消失/爆炸问题。

第6章 面向过光滑的方法

过光滑是限制图卷积神经网络加深的特有问题，本章首先对该问题进行了理论分析，接着设计了实验对理论进行验证，然后从四个角度提出了缓解方法： 基于图数据预处理的方法、基于控制邻居权重的方法、基于平衡局部全局的方法和基于增强自身特征的方法，最后在9个数据集上进行了实验。

6.1 问题定义

GCN可以分为两个步骤。 首先对结点特征进行图卷积操作，接着再进行一次线性变换操作，其中图卷积是性能提升的关键。 {40%：我们定义结点特征的每个通道的拉普拉斯平滑见公式（6.1）。}

yi=1-γxi+γjaijdixj#6.1

其中aij是添加了自循环的邻接矩阵A=A+I的分量，0<γ<1是平衡结点自身特征和邻居特征的权重参数。 我们可以将公式（6.1）写成矩阵形式，见公式（6.2）。

Y=X-γD-1LX=I-γD-1LX#6.2

这里L=D-A，D-1L是归一化拉普拉斯矩阵。 假设不使用自身特征，令γ=1则Y=D-1AX，我们得到拉普拉斯平滑的标准形式。 {42%：如果用对称归一化拉普拉斯矩阵代替归一化拉普拉斯矩阵，我们进一步得到GCN，所以GCN是一种特殊的拉普拉斯平滑，即对称拉普拉斯平滑。} {45%：由于邻接矩阵添加了自循环，拉普拉斯平滑仍然包含结点自身特征。} 通过计算自身特征和邻居特征的基于结点度数的加权平均，我们得到结点特征的新的表示。

连通分量的指示向量的定义见公式（6.3）表述，该指示向量描述结点j是否在分量Ci中。

1ji=1，vj∈Ci0，vj∉Ci#6.3

对于任意的α∈0，1，w∈Rn，我们有关于图卷积的结论[4]，见公式（6.4）-（6.5）表述。其中θ1∈Rk，θ2∈Rk。

limm +∞I-αLrwmw=11，12，… ，1kθ1#6.4

limm +∞I-αLsymmw=D-1211，12，… ，1kθ2#6.5

可以看到，随着归一化拉普拉斯平滑的不断使用，图的每个连通分量内结点的特征会收敛到同一个值。 对于对称归一化拉普拉斯平滑，该值与结点度数的二分之一次幂成正比。如果每个类簇恰好是一个连通分量，那么这将有利于分类任务。 但是事实上实验用到的图数据集，不同类簇之间是连通的，甚至整张图都是连通的， 而重复使用拉普拉斯平滑可能会混合不同类簇中的结点特征使得它们难以被区分， 随着层数增加最终所有结点都收敛到相似的值完全无法区分。

6.2 实验验证

6.2.1 批量归一化验证

在第5章中，我们在Pubmed数据集上用批量归一化方法实验了[1，16]层的GCN，实验结果表明GCN的准确率没有发生骤降。 考虑到批量归一化隐含的数据增强效果，以及引用网络结点间的连接比较稀疏，过光滑问题不是很严重， 我们又在Chameleon数据集上用批量归一化实验了[1，16]层的GCN，见图6.1。 该数据集结点间的连接很密集，过光滑问题比较严重。

图6.1 Chameleon上批量归一化方法的实验结果

{41%：可以看到，即使使用了批量归一化方法，训练集和测试集的准确率也都会在波动中大幅下降。} 此外，该方法还起到了负面作用， 损害了GCN的性能。

6.2.2 过光滑理论验证

过光滑问题是由于重复使用拉普拉斯平滑，不同类簇中的结点特征发生混合而难以区分。同一类簇的内部结点倾向于连接比较密集，越往中心连接越密集，而边缘结点连接比较稀疏，不同类簇间连接也比较稀疏。 当层数较少时混杂的结点也较少，此时性能缓慢下降，当层数到达某一阈值时，平滑范围触及了连接密集区域，混杂的结点骤增，因而性能急剧下降。当层数在[1-3]区间时，浅层学习到的结构信息匮乏为主要问题，增加层数可以学习到更多结构信息，因此增加层数可以提高性能。 我们可以人为地将图数据集按照标签类别分割为几个连通分量，根据对过光滑的分析， 此时随着层数增加，连通分量内的结点收敛到各自的值，准确率会持续上升直到趋于稳定。

图6.2 过光滑理论验证

为排除过拟合和梯度消失/爆炸问题的混合影响，我们采用SGC模型开展对过光滑问题的研究。 SGC模型是对GCN模型的简化，它假设GCN层间的非线性不是关键，局部邻居的聚合操作才是关键[21]。 通过删除层间的非线性激活函数，只保留分类任务中最终的softmax函数，得到的SGC模型见公式（6.6）表述。

Y=softmaxA⋯ AAXΘ1Θ2… ΘK =softmaxAKXΘ#6.6

这里A是添加了自循环的对称归一化邻接矩阵。 可以看到，SGC模型由两部分组成，一个不含参数的特征提取器X=AKX，一个线性逻辑回归分类器Y=softmax(XΘ)。 由于SGC只包含一层可学习参数，K表示的是拉普拉斯平滑的次数，K的增加并不会导致过拟合和梯度消失/爆炸问题， 因此SGC适合用来研究过光滑问题。

对过光滑理论的验证实验结果见图6.2，其中Cut表示进行了图割处理，即去除了不同类簇间的噪声边。 {42%：可以看到，未做图割处理时，在[1，2]层训练集和测试集的准确率上升，} 2层后准确率缓慢下降，8层后准确率开始骤降，随着层数增加最终趋于稳定。{42%：进行图割处理后，训练集和测试集的准确率随着层数的增加持续上升直至趋于稳定。}{44%：该实验结果与理论分析完全吻合，充分证实了过光滑的成因分析。} 我们可以根据该分析来提出一些缓解方法。

6.3 基于图数据预处理的方法

根据对过光滑问题的理论分析，只要我们能够去除不同类簇间的噪声边，就能够从根本上解决该问题。 理想情况下，不同类簇间完全不连通，也就不存在过光滑问题。

DropEdge通过随机丢弃一定比例的边，在引用数据集上取得了一定的效果。 被丢弃的边集合中包含了一部分噪声边，因此DropEdge起到了缓解过光滑的作用。 但是实验表明，在密集连接的Chameleon等数据集上，DropEdge反而会降低GCN的性能。 这是因为Chameleon等数据集包含大量的噪声边，基于随机性丢弃边会导致同一类簇间的有效边的损失。 如果我们能够根据某种指标，针对性地丢弃一些边，使得被丢弃的边中包含更多的噪声边，那么就能提升DropEdge的性能。 我们基于两种假设，分别改进了DropEdge。

一类是基于结点相似度的改进。 假设不同类簇间的结点相似度较小，我们可以据此对

图数据进行割边，使得更多的噪声边被丢弃。 我们采用余弦相似度进行计算，并将计算结果进行softmax归一化处理，见公式（6.7）表述。

a0，j=expcosx0，xjk∈Nv0expcosx0，xk#6.7

{45%：这里N(v0)表示结点v0的邻居集合，a0，j表示注意力权重，描述了结点和邻居的相对重要性。} 我们将边按照注意力权重排序，按照比例α删除权重值较小的边。 在代码实现中，通过将边的权重置为零进行删边操作。

另一类是基于结点度数的改进。 假设不同类簇间的结点连接比较稀疏，我们可以据此对图数据进行割边，使得更多的噪声边被丢弃。 我们将结点按照度数排序，按照比例α筛选出度数较小的结点，接着在这些结点上按照比例β随机删除边。

6.4 基于控制邻居权重的方法

我们也可以通过控制结点对邻居结点的聚合权重来缓解过光滑问题。 理想情况下，A类簇的结点对位于B类簇的邻居结点的聚合权重为0，不同类簇的结点不会产生混合，过光滑问题得到解决。 我们在基于注意力机制的GAT模型上做了一点改进，提高了模型的运行速度，同时保证性能不受影响。 GAT利用参数向量学习结点和邻居间的相对重要性[22]，见公式（6.8）表述。

a0，j=expLeakyReLU(a WxOWxj)k∈Nv0expLeakyReLU(a WxOWxk)#6.8

这里a是可学习的参数向量，用于学习相对重要性。 W是可学习的参数矩阵，用于对输入特征做线性变换， 是向量拼接操作。 由于GAT包含许多可学习参数，训练速度相对来说比较慢。 我们用余弦相似度直接计算相对重要性，代替参数向量a和激活函数LeakyReLU(·)，见公式（6.9）表述。

a0，j=expcosWx0，Wxjk∈Nv0expcosWx0，Wxk#6.9

6.5 基于平衡局部全局的方法

近距离的邻居比远距离的邻居更重要，并且远距离的邻居容易导致过光滑。 如果GCN能够平衡好局部与全局的信息，就能在一定程度上缓解过光滑问题。 我们可以借鉴CNN中的残差网络ResNet和密集网络DenseNet[23]的结构来改进GCN，见图6.3。

图6.3 ResGCN、DenseGCN网络结构

提出残差连接的初衷，是让模型的内部结构至少有恒等映射的能力， {77%：以保证在堆叠网络的过程中，网络不会因为继续堆叠而产生退化[2]。} 此外，残差连接还有其他一些作用。 即使批量归一化处理后梯度的模稳定在正常范围，但是梯度的相关性会随着层数增加持续衰减， 而残差连接可以有效较少这种相关性的衰减[24]。 另外，浅层特征具有高分辨率低级语义，深层特征具有高级语义低分辨率，而残差连接可以实现不同分辨率特征的组合[25]。

我们可以将残差连接应用到GCN，从而组合高低层不同范围的邻居信息。 进一步地，我们可以使用密集连接加强该作用。 此外，相比较JK-Net，残差连接和密集连接都缓解了较深层产生的输出仍然存在不同类簇间的结点混合的问题

我们在残差连接上引入了权重参数，该参数平衡了局部和全局的相对重要性，

{45%□α的值越大，说明局部性越重要，见公式（6.10）表述。}

Hl+1=σD-12AD-12HlWl+αHl#6.10

6.6 基于增强自身特征的方法

{45%：一个结点的信息主要由两部分组成，自身信息和结构信息。} 其中自身信息由结点特征体现，结构信息由邻居结点体现。 然而，随着层数加深，越来越多的邻居结点被聚合，结点自身的信息越来越匮乏。 为了缓解该问题，我们对GCN做了一些改进，见公式（6.11）表述，该方法也可用于SGC等模型。

Hl+1=σ1-αD-12AD-12Hl++αH0Wl#6.12

该公式相当于在每一层引入了输入层的带权重的跳接，见图6.4。 从随机游走的角度来看，α表示游走过程中回退到出发结点的概率，也起到了平衡局部和全局的作用。

图6.4 SelfNet网络结构

6.7 实验分析

基于图数据预处理的方法

实验中采用基于结点相似度的改进方法，将原方法DropEdge作为对照组之一，分别在SGC和GCN模型上进行了实验。 我们用网格搜索对丢弃比例超参数α进行了优化，搜索空间为[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.05, 0.01]，最终确定了9个数据集上各自的最优值，其中原方法DropEdge为[0.2, 0.1, 0.05, 0.05, 0.01, 0.01, 0.6, 0.1, 0.01]，基于结点相似度的改进方法为[0.05, 0.05, 0.01, 0.6, 0.6, 0.7, 0.6, 0.5, 0.6]。 详细的实验结果见表6.1。

这里SGC（DR0）和SGC（DR1）分别表示采用了原方法DropEdge和基于结点相似度的改进方法的SGC， {51%：括号里的数字表示取得最佳准确率的层数。} 可以看到，DropEdge对性能的提升有限，而基于结点相似度改进的DropEdge却在多个数据集上表现突出， 特别是在过光滑比较严重的密集连接数据集上有较大提升，并且取得最佳准确率的层数也有所提高。我们也在GCN上进行了实验，实验结果表明，基于结点相似度改进的DropEdge对GCN也有所增益。{41%：此外，GCN模型比SGC模型的整体结果更好，说明非线性变换可以增强学习能力。}

图6.5 Pubmed上基于图数据预处理的方法的实验结果

为了排除过拟合和梯度消失/爆炸的混合影响，我们用SGC模型在层数区间[1, 16]上进行了实验，见图6.5。 可以看到，采用了基于结点相似度改进的DropEdge方法后，过光滑问题得到了很大程度的缓解。

表6.1 基于图数据预处理的方法的实验结果

模型

SGC

SGC(DR0)

SGC(DR1)

GCN

GCN(DR0)

GCN(DR1)

Cora

84.34(3)

85.54(2)

84.34(3)

87.95(3)

86.75(6)

87.35(4)

Cite.

76.89(2)

75.24(1)

77.36(2)

76.18(2)

76.65(2)

76.65(2)

Pubm.

82.25(1)

82.18(2)

82.3(2)

86.92(2)

87.04(2)

87.2(2)

Cham.

42.98(2)

41.01(2)

45.83(2)

43.2(2)

44.3(2)

46.27(1)

Squi.

28.28(5)

28.31(2)

29.17(2)

27.83(6)

27.64(2)

29.08(2)

Actor

28.51(1)

28.09(1)

34.93(1)

27.63(2)

27.76(2)

33.55(3)

Corn.

26.32(1)

34.21(2)

34.21(4)

26.32(2)

26.32(1)

63.16(3)

Texa.

64.91(2)

65.79(1)

73.68(2)

68.42(2)

63.16(2)

71.05(3)

Wisc.

60.26(2)

59.62(4)

80.77(4)

57.69(2)

57.69(2)

82.69(5)

基于控制邻居权重的方法

我们在SGC和GCN上分别对基于控制邻居权重的方法进行了实验，详细的实验结果见表6.2，其中SGC（WE）表示使用了该方法的SGC模型。 可以看到，基于控制邻居权重的方法有一定效果，但是不如基于图数据预处理的方法。

图6.6 Pubmed上基于控制邻居权重的方法的实验结果

同样地，我们也用SGC模型在Pubmed上进行了实验，见图6.6。 可以看到，基于控制邻居权重的方法效果有限，并且也无法缓解过光滑问题，层数达到9层后训练集和测试集的准确率都大幅下降。 这是因为采用了该方法后，噪声边的权重确实降低了，但是仍然是一个正值，经过多层聚合叠加后，不同类簇的结点特征还是发生了混合。

表6.2 基于图数据预处理的方法的实验结果

模型

SGC

SGC(WE)

GCN

GCN(WE)

Cora

84.34(3)

85.34(3)

87.95(3)

87.55(2)

Cite.

76.89(2)

74.76(1)

76.18(2)

76.42(2)

Pubm.

82.25(1)

82.96(2)

86.92(2)

87.22(3)

Cham.

42.98(2)

45.83(4)

43.2(2)

46.49(3)

Squi.

28.28(5)

28.98(4)

27.83(6)

29.17(1)

Actor

28.51(1)

33.22(1)

27.63(2)

33.03(1)

Corn.

26.32(1)

26.32(1)

26.32(2)

26.32(1)

Texa.

64.91(2)

68.42(1)

68.42(2)

68.42(2)

Wisc.

60.26(2)

63.46(1)

57.69(2)

59.85(3)

基于平衡局部全局的方法

我们用GCN模型实验了残差连接，带权重的改进残差连接和密集连接，分别用GCN（RES0）、GCN（RES1）和GCN（DEN）表示。通过网格搜索方法对权重超参数进行了优化，搜索空间为[0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]，最后在9个数据集上的优化值为[0.5, 1.5, 1.5, 2.5, 1, 2, 3, 2, 4]。详细的实验结果见表6.3。可以看到，带权重的改进残差连接效果最好，其次是原始残差连接。密集连接几乎没有效果，这可能是因为对多层输出拼接做线性变换引入过多参数，不足以学习到局部和全局的平衡信息。

图6.7 Pubmed上基于图数据预处理的方法的实验结果

我们用GCN在Pubmed上实验了残差连接，见图6.7。可以看到，残差连接的表现非常好，在每层上都有提升，同时随着层数增加，性能还会缓慢上升。{40%：残差连接能够很好地学习局部与全局信息。}

表6.3 基于图数据预处理的方法的实验结果

模型

GCN

GCN(RES0)

GCN(RES1)

GCN(DEN)

Cora

87.95(3)

87.15(6)

87.55(3)

85.74(2)

Cite.

76.18(2)

76.89(2)

78.07(2)

76.65(2)

Pubm.

86.92(2)

88.18(5)

88.59(5)

86.82(6)

Cham.

43.2(2)

46.49(1)

49.12(1)

43.64(4)

Squi.

27.83(6)

30.81(8)

30.71(8)

27.83(7)

Actor

27.63(2)

33.88(1)

34.54(1)

26.71(2)

Corn.

26.32(2)

34.21(2)

60.53(2)

26.32(2)

Texa.

68.42(2)

71.05(1)

78.95(4)

68.42(8)

Wisc.

57.69(2)

65.38(1)

75.0(2)

59.62(2)

基于增强自身特征的方法

{42%：实验中采用网格搜索进行超参数寻优，搜索空间为[0.1, 0.2, 0.3, } 0.4，0.5， 0.6， 0.7， 0.8， 0.9， 0.05， 0.01]，在9个数据集上的搜索结果为[0.2, 0.4, 0.2, 0.4， 0.9， 0.9， 0.9， 0.8， 0.7]。 详细的实验结果见表6.4。 其中SGC（SE）表示使用了基于增强自身特征的方法的SGC。 可以看到，

该方法对SGC和GCN模型都有较好的增益效果。

图6.8 Pubmed上基于图数据预处理的方法的实验结果

我们也在Pubmed上用SGC实验了该方法，见图6.8。 {43%：可以看到，该方法和残差连接一样表现出色，训练集和测试集的准确率都随着层数持续波动上升。} 不仅缓解了过光滑问题，同时整体性能也有提高。

表6.3 基于增强自身特征的方法的实验结果

模型

SGC

SGC(SE)

GCN

GCN(SE)

Cora

84.34(3)

84.34(6)

87.95(3)

87.75(3)

Cite.

76.89(2)

78.3(1)

76.18(2)

77.12(2)

Pubm.

82.25(1)

83.22(2)

86.92(2)

86.71(2)

Cham.

42.98(2)

48.25(5)

43.2(2)

43.86(3)

Squi.

28.28(5)

30.9(4)

27.83(6)

30.04(2)

Actor

28.51(1)

37.11(1)

27.63(2)

32.43(2)

Corn.

26.32(1)

26.32(1)

26.32(2)

50.0(1)

Texa.

64.91(2)

68.42(2)

68.42(2)

71.05(3)

Wisc.

60.26(2)

65.38(1)

57.69(2)

65.38(1)

6.8 本章小结

本章首先对过光滑问题进行了理论分析，接着精心设计实验验证了该理论，然后基于理论分析从不同角度提出了缓解方法： 基于图数据预处理的方法、基于控制邻居权重的方法、基于平衡局部全局的方法和基于增强自身特征的方法，最后对这些方法进行了充分的实验。实验结果表明，以上方法都有一定效果，其中基于结点相似度的改进DropEdge，带权重的残差连接表现最突出。

第7章 总结与展望

7.1 本文总结

本文通过理论分析和实验验证相结合的方式，系统地对图卷积神经网络无法加深这一问题开展了研究。

{41%：文章首先介绍了几种主要的深度图卷积神经网络模型，阐述了它们的优缺点以及本文所做的改进。}

接着针对过光滑问题，从理论角度进行了分析，在图卷积神经网络上引入了三种正则化方法： 权重衰减、提前终止和丢弃法，并在引用数据集上进行了实验。 实验结果表明，过光滑是限制图卷积神经网络加深的一个因素，但不是主要因素，传统的正则化方法在该问题上对GCN也有效。 本文将提前终止作为实验的一种辅助手段，以节省不必要的计算开销。

然后针对梯度消失问题，从理论角度进行了分析，在图卷积神经网络上引入了三种传统方法： {41%：Xavier初始化、梯度修剪和批量归一化，同样在引用数据集上进行了实验。}实验结果表明，梯度消失也不是限制图卷积神经网络加深的主要因素，传统的几种方法在该问题上对GCN也有效。 由于Xavier初始化的有效性，本文将其作为GCN的固定配置。 值得注意的是，批量归一化在引用数据集上表现突出，在[1，16]层数区间内保持了稳定的性能。考虑到引用数据集的连接稀疏性，本文引入了几个密集连接的数据集，在共计9个数据集上开展后续实验研究。

最后针对过光滑问题，也从理论角度进行了分析，并精心设计了验证实验。 本文采用SGC模型进行过光滑的实验研究，避免了过拟合和梯度消失问题的干扰。 从图数据预处理的角度，基于结点相似度对DropEdge进行了改进。 {47%：从控制邻居权重的角度，基于余弦相似度对GAT进行了改进。} 从平衡局部全局的角度，在GCN上引入了CNN中的残差连接和密集连接，并提出了带权重的残差连接以进一步强调局部全局。 从增强自身特征的角度，在网络的每一层引入了输入层的跳接，形成了新的网络结构。 实验结果表明，过光滑是限制图卷积神经网络加深的主要因素，除了基于余弦相似度的改进GCN，本文提出的几种方法都能较好地缓解该问题。

7.2 下一步工作

尽管本文引入或提出的方法在多个数据集上取得了较好的表现，但是由于图神经网络的基准数据集规模较小， 因此实验结果对方法表现的区分度还不够。 最近有研究者新提出了几个中等规模的图基准数据集，后续可以在这些数据集上进一步开展实验。 此外，个别方法也存在着不足之处。 基于结点相似度改进的DropEdge在密集连接数据集上表现突出，然而在稀疏连接的引用数据集上提升有限， 可以寻找其他标准来进行割边，以进一步提高噪

声边被丢弃的概率。

参考文献

[1] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv: 1609.02907, 2016.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[3] Xu K, Li C, Tian Y, et al. Representation learning on graphs with jumping knowledge networks[J]. arXiv preprint arXiv: 1806.03536, 2018.

[4] Li Q, Han Z, Wu X M. Deeper insights into graph convolutional networks for semi-supervised learning[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[5] Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks? [J]. arXiv preprint arXiv: 1810.00826, 2018.

[6] Klicpera J, Bojchevski A, Günnemann S. Predict then propagate: Graph neural networks meet personalized pagerank[J]. arXiv preprint arXiv: 1810.05997, 2018.

[7] Chiang W L, Liu X, Si S, et al. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 257-266.

[8] Abu-El-Haija S, Kapoor A, Perozzi B, et al. N-gcn: Multi-scale graph convolution for semi-supervised node classification[J]. arXiv preprint arXiv: 1802.08888, 2018.

[9] Huang B, Carley K M. Residual or gate? towards deeper graph neural networks for inductive graph representation learning[J]. arXiv preprint arXiv: 1904.08035, 2019.

[10] Li G, Müller M, Thabet A, et al. Can GCNs Go as Deep as CNNs? [J]. arXiv preprint arXiv: 1904.03751, 2019.

[11] Rong Y , Huang W , Xu T , et al. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification[J]. 2019.

[12] Luan S, Zhao M, Chang X W, et al. Break the Ceiling: Stronger Multi-scale Deep Graph Convolutional Networks[C]//Advances in Neural Information Processing Systems. 2019: 10943-10953.

[13] Zhao L, Akoglu L. PairNorm: Tackling Oversmoothing in GNNs[J]. arXiv preprint arXiv: 1909.12223, 2019.

[14] Hoang N T, Maehara T. Revisiting graph neural networks: All we have is low-pass filters[J]. arXiv preprint arXiv: 1905.09550, 2019.

[15] Pei H, Wei B, Chang K C C, et al. Geom-gcn: Geometric graph convolutional networks[J]. arXiv preprint arXiv: 2002.05287, 2020.

[16] Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems[M]. O'Reilly Media, 2019.

[17] 邱锡鹏. 神经网络与深度学习[M]. 第1版. 北京： 机械工业出版社, 2020.

[18] Srivastava N, Hinton G, Krizhevsky A, et al., 2014. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 15(1): 1929-1958.

[19] Glorot X, Bengio Y, 2010. Understanding the difficulty of training deep feedforward neural networks[C]//Proceedings of International conference on artificial intelligence and statistics. 249-256.

[20] Ioffe S, Szegedy C, 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//Proceedings of the 32nd International Conference on Machine Learning. 448-456.

[21] Wu F, Zhang T, Souza Jr A H, et al. Simplifying graph convolutional networks[J]. arXiv preprint arXiv: 1902.07153, 2019.

[22] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv: 1710.10903, 2017.

[23] Huang G, Liu S, Van der Maaten L, et al. Condensenet: An efficient densenet using learned group convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2752-2761.

[24] Balduzzi D, Frean M, Leary L, et al. The

shattered gradients problem: If resnets are the answer, then what is the question? [C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 342-350.

[25] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

致 谢

{54%：光阴似箭，日月如梭，四年的本科生活也将在尚未结束的疫情中画上句号。} 回首这四年来，经历过曲折和坎坷，也收获过喜悦与快乐。 许许多多的人曾帮助或指导过我，在我学习和生活的道路上不断地支持我， {41%：鼓励我，让我拥有了一个美好而难忘的大学生活，我将永远心存感激。}

{45%：首先我要感谢东北大学，感谢软件学院，感谢学校给我们提供的良好的教育资源和学习环境，} {45%：感谢软件学院的老师们辛勤的付出，是你们给我四年的精心教育和指导，} {74%：谢谢你们的教育为我以后的学习和生活打下了坚实的基础。} 感谢毕业设计的校内指导老师张伟老师，感谢您在我毕设期间辛勤的付出，对我们的论文和相关材料认真的审阅和校对，并给予我细心的指导。

特别地，感谢复旦大学大数据学院的黄增峰老师，很幸运能在黄老师的指导和帮助下进行此次毕业设计。 在毕设期间，每当我遇到研究工作中的难题时，黄老师总是耐心指导，给予细致、具体的说明，更给与我极大的鼓励和支持。 {48%：黄老师扎实的学术功底、认真严谨且一丝不苟的学术作风，不辞劳累的工作态度让我深深地折服与敬佩。}

{49%：还要感谢我亲爱的朋友们，是你们在生活中给我鼓励，陪我前行。} 生活中我们有争吵也有欢喜，是你们陪我度过了本科四年中最长的岁月，与你们的一起的时光是我永远珍贵的回忆， 与你们的友谊也将是我一生珍惜的情谊。

最后深深的感谢呵护我成长的父母。 {100%：每当我遇到困难的时候，父母总是第一个给我鼓励的人。} {95%：回顾二十多年来走过的路，每一个脚印都浸满着他们无私的关爱和谆谆教诲，四年年的在外求学之路，寄托着父母对我的殷切期望。} {100%：他们在精神上和物质上的无私支持，坚定了我追求人生理想的信念。}

衷心地感谢所有帮助和支持过我的人。 在人生的新阶段，我也将朝着下一个目标继续努力，