基于 PYTORCH 对图神经网络 GCN、SGC 的实现、比较与改进

石云天 3020205015

摘要

近年来，由于图这一结构的强大表现力，利用机器学习方法分析图的研究越来越受到重视。图神经网络（GNN）是一类基于深度学习的用来处理图域信息的方法。因其较好的性能和可解释性，GNN 逐渐成为一种被广泛应用的图分析方法。本次研究主要围绕各类 GNN 算法，从最经典的图卷积神经网络 GCN 入手，同时包含其变种改进（如 SGC 等），探索图卷积神经网络的核心内容以及影响其性能的主要因素。

在本次实践中，基于 Pytorch 实现了 GCN、SGC 等图神经网络，并讨论和对比各类算法框架的核心细节。在图领域的经典数据集上完成 transductive 的分类任务，通过调整各种架构组成和参数进行计算，对结果进行比较和分析，从而深入理解图卷积神经网络的核心内容，探索影响性能的关键因素。

关键词 — Pytorch, GNN, GCN, SGC

1. 引言

图结构数据是一种在现实生活中广泛存在的数据形式，也可以说最基本的数据结构。常见的一维序列和二维矩阵都可以看作是特殊的规整而固定的图结构。而其他实体——互联网、论文引用网络、社交网络、蛋白质分子结构等则是较为普遍的复杂图结构数据。如何有效地处理和分析这类数据，并将其应用于广阔的场景中，具有重大研究意义。目前各类 GNN 研究发展迅速，能处理的图结构问题也越来越丰富，具体见下图 1-1。

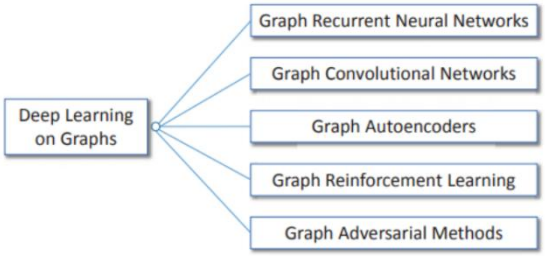

图 1-1 GNN 研究拓展

本次研究以 GCN 为出发点来深入研究 GNN。虽然 GCN 网络具有强大的处理图数据的能力，但其缺点也较为突出：由于使用拉普拉斯矩阵进行正则化，要求邻接矩阵为对称矩阵，即需要输入的图为无向图，从现实意义角度出发，将有向图视作无向图会造成准确率的下降。

本次研究基于 Pytorch 实践来理解 GCN 的特点，同时实现改进的 SGC 版本，通过网络结构对比以及在实际数据集上的测试表现对比分析来进一步探索 GCN。

2. 研究方法

在阅读各类相关文献后，我对图神经网络的特点逐渐有了一些自己的理解。图数据简单而言就是点和边的组合，而图的所有边就包含了其结构信息。如何根据图的结构传播与聚合信息，就是解决图问题的核心。相邻节点可以认为具有相似的特征，存在某种联系。所以问题也可以表达为如何根据某个节点的邻居节点来定义一种计算图，通过某种方式来聚合信息。

本次研究的起点是实现图卷积神经网络——GCN，通过阅读文献，我明白 GCN 是通过邻接矩阵实现聚合所有单跳邻居节点的信息，并通过谱域的对称拉普拉斯矩阵进行正则化，通过谱分解构造卷积核。谱域的概念较为复杂，但简单理解 GCN，即为每次卷积运算就是选取节点的所有一阶邻居节点构造计算图，所有节点权重相同，使用正则化技术来处理某些度过大的节点，再完成卷积后，可以接着进行非线性激活等操作。每层过

后，节点的特征，即 embedding 发生变化。在最后一层通过 softmax 函数输出长度为类别数的 embedding，便可进行节点分类，利用交叉熵损失函数即可反向传播训练参数。

根据 GCN 的理论，网络的深度不能过深，否则会出现“过平滑”的问题，即节点信息传播太多次而导致所有节点的信息趋于一致。官方给出的图卷积层数量为 2，在充分理解了 GCN 的原理和过程后，通过 Pytorch 实现 GCN 并在数据集上进行训练和测试。网络架构大致与官方版本相同，在一些细节结构上进行微调，例如，修改卷积层之间的非线性激活函数，增加 dropout 层进行正则化等，具体见下图 2-1。

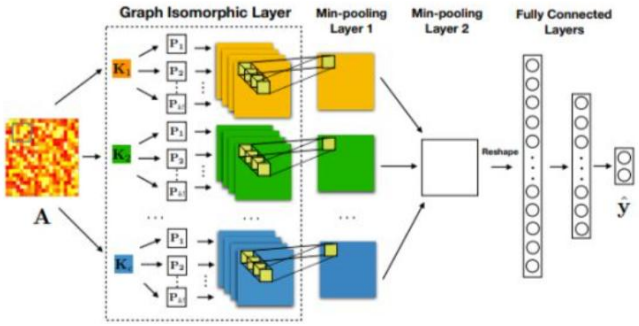

图 2-1 GNN 网络架构示意图

在实现 GCN 后，根据分析的 GCN 特点进行一系列探索和研究。主要运用的方法有可视化呈现、追踪训练过程并绘制曲线、调整网络结构与超参数后开展一系列对比实验。

通过可视化的方式探索 GCN 的处理过程，同时根据分析得到的 GCN 缺点进行一些网络架构和参数层面的改进探索研究。在本次研究中，我尝试对原始 GCN 进行修改以适应有向图的处理和计算。这一改进的核心思路就是以非对称矩阵适用的正则化方式来替代原模型中的拉普拉斯矩阵，详细实验过程见下文第 3 节。

在神经网络超参数优化方面，由于 GCN 在结构和深度层面较为简单，供调节的超参数主要集中于反向传播的优化器上。GCN 最终得到的 embeddings 可以直接用于分类和损失函数。基于半监督的训练方式是指 GCN 始终在全图上进行计算处理，我们使用已知的部分节点的分类标签来训练。使用 SGD 即可实现反向传播调参。在官方实现中使用了 Adam 优化器，我在经验范围内手动调参并使用 Hyperopt 库进行自动化选参，主要的超参数为学习率与衰减权重。对比不同超参数下性能的差异，同时尝试其他种类的优化器并在测试集上对比结果性能。

SGC 是一种简化的 GCN，在大大提高效率的同时保持了与 GCN 相近的性能。SGC 舍弃了原模型中的线性激活，只在最后使用一个全连接层处理节点进行分类。而节点之间的信息传递则是根据 GCN 从单跳邻居节点获取信息的基本思想，以无参数自环相乘来实现特征传播，这样从全局来看，SGC 只包括了自环特征提取和线性逻辑回归分类器，从而大大减少了 GCN 的计算量和复杂度。同时也对我们理解 GCN 的作用过程有了很大的帮助，本次实验中同样利用 Pytorch 实现 SGC 网络。

在实现 SGC 后，在与 GCN 测试时所用的相同的数据集上进行测试，对比分析二者的性能差异和效率差异，最终推导出 GCN 的关键核心和实际作用。

3. 实验与结果分析

3.1 实验数据集

本次实验选取的数据集为经典的图领域论文广泛引用的数据集，在这些数据集上实现节点分类任务。数据集的详细信息见下表 3-1。

表 3-1 本次实验所用数据集详细信息

数据集	来源	图	节点	边	特征	标签
Cora	“Collective classification in network data,” AI magazine,2008	1	2078	5429	1433	7
Citeseer	“Collective classification in network data,” AI magazine,2008	1	3327	4732	3730	6
Pubmed	“Collective classification in network data,” AI magazine,2008	1	19717	44338	500	3

所选取的三个数据集的结构相似，基本可以看作同种数据。以 Cora 为例，该数据集由机器学习论文组成，共有 2708 篇论文（节点），每篇论文均与其他论文存在引用关系。其初始特征是 1433 维，表示 1433 个独特单词，共有 5429 条边，表示其中存在 5429 个引用关系。其中值得注意的是论文引用实际上是一种单向的联系，引用和被引用具有不同的意义，因此 Cora 等数据集实

际上均为有向图。在数据集.cite 文件中的引用关系均为$p_1 \rightarrow p_2$的关系，即p_2引用p_1。但是由于 GCN 的正则化要求邻接矩阵为对称矩阵，在计算时需要将该图网络视作无向图，对数据进行预处理时要生成对称的邻接矩阵。当然，后续在对 GCN 的改进调整中考虑还原为有向图进行处理。

3.2 实验环境配置

本次实验主要在自己的电脑上进行，使用 CPU 运算。代码支持 cuda，在实验过程中在 Google colab 上使用 GPU 资源进行后测试，后续的实验结果为本机测试结果。

对于实验中所使用的数据集，调用 PyTorch Geometric 中的接口来加载数据，PyG 实现了本次实验所使用的三种数据集的统一封装，使用时只需进行统一预处理即可。

3.3 实验内容

3.3.1 图算法运行测试与可视化

首先测试 GCN 的分类过程，同时追踪整个训练过程。使用官方推荐的参数进行实验，借助 pca 降维将每个阶段算法输出的 embedding 可视化，以便我们观察到 GCN 的无监督学习过程，可视化结果见下图 3-1、3-2、3-3、3-4。

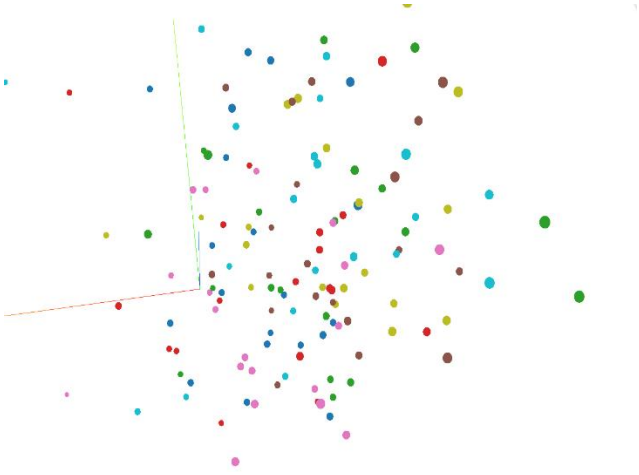

图 3-1 Train_set: range(140), Epoch = 50

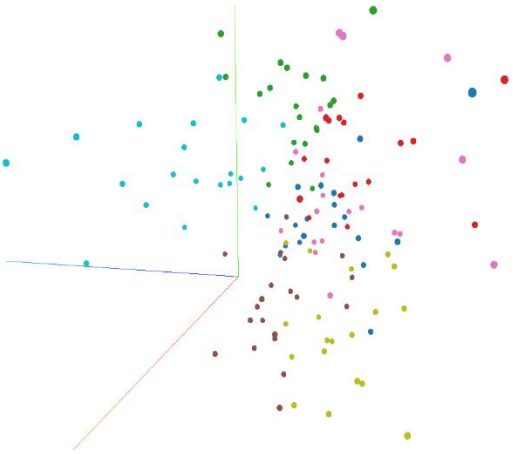

图 3-2 Train_set: range(140), Epoch = 100

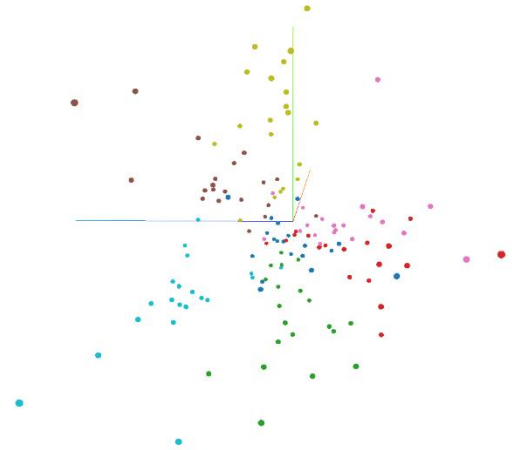

图 3-3 Train_set: range(140), Epoch = 150

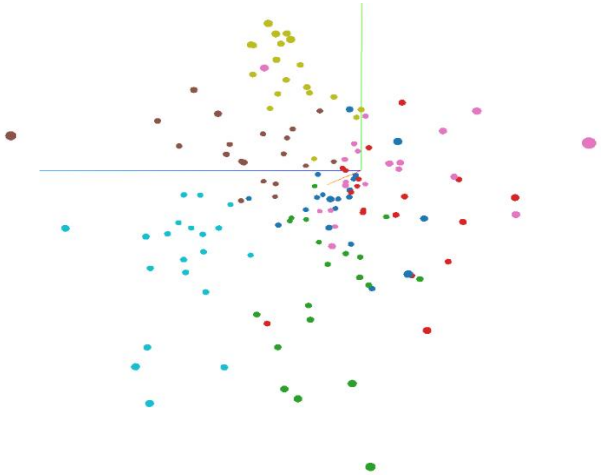

图 3-4 Train_set: range(140), Epoch = 200

从训练集上 embedding 分布的变化可以明显观察到在无监督学习过程中相同类别的节点在相互靠近，GCN 的处理能力可见一斑。在最后一个 Epoch 中结果已经相当出色，由于正则化技术的存在，训练集测试结果没有达到 100%的准确率。

从结果上可以观察到，即使是没有任何参数学习的自环传播，两次传播过后节点之间的聚类效果也十分出色，某些类别单独占据一片区域，也有一些类别虽然与其他类别一定程度上混杂在一起，但是聚集的趋势非常明显。该实验结果可以证明 GCN，SGC 通过邻接矩阵聚合邻居节点特征的方式可以起到很好的特征提取效果。因此才会使 SGC 彻底舍弃了卷积操作之间的非线性变换，将模型转为一种线性逻辑回归模型，其中最关键的组件是 k-step 传播过程。**该过程类似于一种低通滤波，节点之间信息共享，达到一种平滑的效果。**

3.3.2 图算法性能测试

在利用可视化手段表现出节点传播效果的强大后，对两种算法的实际分类性能进行实验测试。以 Cora 数据集训练可视化为例，使用两种图神经网络计算后对比结果测试集的 Accuracy 和 loss，其中蓝色线表示 SGC，红色线表示 GCN。

从中可以明显观察到 SGC 训练的收敛速度要比 GCN 快很多。在训练时统一设置 Epochs 为 200，但是 SGC 可以较快结束训练。无论是准确率还是 loss，SGC 的更新速度都要快好几倍。接下来观察模型最终的性能效果，两者不分上下。图 3-5、3-6 非常直观地展示出 SGC 在舍弃多余运算，仅保留核心的节点信息传播和最终的逻辑回归后，性能没有发生明显的下降，甚至与 GCN 基本持平。

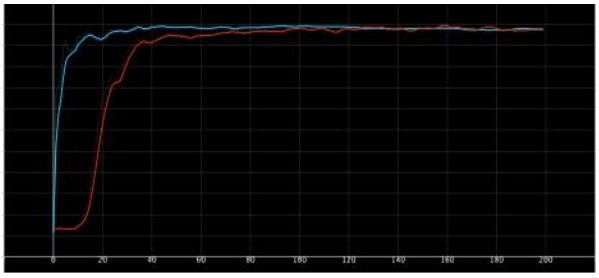

图 3-5 SGC、GCN Accuracy 对比

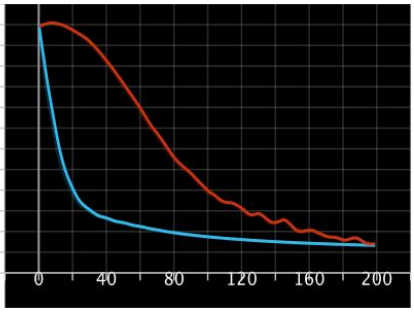

图 3-6 SGC、GCN loss 对比

接下来使用全部三个数据集对两种网络进行实验测试。统一在相同的训练集条件下训练 200Epochs, 所使用的超参数均为官方所给参数，最终在测试集上的评估结果如下表 3-2、3-3 所示：

表 3-2 SGC 评估结果

数据集	loss	Accuracy	Time(s)
Cora	0.7997	0.8130	0.7400
Citeseer	1.0555	0.7120	0.8151
Pubmed	0.5718	0.7950	0.4131

表 3-3 GCN 评估结果

数据集	loss	Accuracy	Time(s)
Cora	0.7871	0.8190	6.3408
Citeseer	1.0324	0.7100	18.3622
Pubmed	0.5956	0.7810	24.6462

综合三个数据集的实验评估结果，再次印证了 SGC 在保证了分类性能与 GCN 相近的前提下，大大提高训练效率和收敛速度，这也证明了原 GCN 的非线性变换对于结果基本没有影响，使用该方法进行图数据的特征提取和节点分类只需进行 SGC 中保留的两个关键部分即可。

4. 结论

本次研究实验围绕图神经网络的深入理解，从图卷积神经网络 GCN 出发，利用 Pytorch 实现了 GCN、SGC 图神经网络。主要运用了结构分析，对比实验，实验结果比较分析等方法进行研究。深入探索 GCN 等图神经网络的特性，对理解图卷积神经网络和后续的改进方向与改进思路有很大帮助。

在整个课题研究中，通过可视化的方式观察了图卷积传递和聚合信息的效果。通过 GCN 和 SCG 的一系列对比试验理解到了图卷积起作用的关键核心。通过对 GCN 算法进行改进以适应有向图探索了该类图卷积的局限性，更进一步掌握其结构特点，从而理解后续 GAT 诞生的整体思路和与关键思想。

5. 参考文献

[1]Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, Maosong Sun, Graph Neural Networks: A Review of Methods and Applications.

[2] Davide Bacciua, Federico Erricaa, Alessio Michelia, Marco Podd , A Gentle Introduction to Deep Learning for Graphs.
[3]Wu, Felix and Souza, Amauri and Zhang, Tianyi and Fifty, Christopher and Yu, Tao and Weinberger, Kilian:Simplifying Graph Convolutional Networks.
[4]Wu, Zonghan , et al. "A Comprehensive Survey on Graph Neural Networks." (2019).