

实验四

实验要求

使用 pytorch 或者 tensorflow 的相关神经网络库，编写图卷积神经网络模型 (GCN)，并在相应的图结构数据集上完成节点分类和链路预测任务，最后分析自环、层数、DropEdge、PairNorm、激活函数等因素对模型的分类和预测性能的影响。

实验步骤

- 网络框架**：要求选择 pytorch 或 tensorflow 其中之一，依据官方网站的指引安装包。（如果前面实验已经安装过，则这个可以跳过）
- 数据集**：本次实验使用的数据包含三个常用的图结构数据集：Cora、Citeseer、PPI。下面分别进行介绍。
 - (1) Cora**：该数据集是由2708篇机器学习论文作为节点、论文间引用关系作为有向边构成的图数据。具体的数据描述见 https://blog.csdn.net/qg_33254870/article/details/103553661。数据集下载链接 <https://lings-data.soe.ucsc.edu/public/lbc/cora.tgz>。另外，提供一个数据处理范例链接 <https://graphsandnetworks.com/the-cora-dataset/>。请同学们仔细阅读相关材料，了解文件的具体结构和数据格式。
 - (2) Citeseer**：该数据集是由3312篇论文及相互引用构成的图数据集。数据集下载链接 <https://lings-data.soe.ucsc.edu/public/lbc/citeseer.tgz>。文件的结构和数据格式与Cora类似。
 - (3) PPI**：PPI网络是蛋白质相互作用（ProteinProteinInteraction,PPI）网络的简称数描述可参考链接 <https://blog.csdn.net/ziqingnian/article/details/112979175>。数据集下载链接 <http://snap.stanford.edu/graphsage/ppi.zip>。
- 数据预处理**：你需要通过pytorch或tensorflow所提供的标准数据接口，将原始数据处理为方便模型训练脚本所使用的数据结构，如torch.utils.data.Dataset等。由于这三个数据集是非常常见的公开数据集，你可以参考一些公开代码片段，尤其是github上典型的GCN教程级实现或相关论文的源码。
- 模型搭建**：搭建GCN模型，这一步可以参考网络上公开的源码，但不能直接使用封装过的库。
- 模型训练**：将生成的训练集输入搭建好的模型进行前向的 loss 计算和反向的梯度传播，从而训练模型，同时也建议使用网络框架封装的 optimizer 完成参数更新过程。训练过程中记录模型在训练集和验证集上的损失，并绘图可视化。
- 节点分类**：在三个数据集上按照节点分类任务的需求自行划分训练集、验证集、测试集，并用搭建好的GCN模型进行节点分类。
- 链路预测**：在三个数据集上按照链路预测任务的需求自行划分训练集、验证集、测试集，并用搭建好的GCN模型进行节点分类。
- 调参分析**：将训练好的模型在验证集上进行测试，以 **Top 1 Accuracy(ACC)** 作为网络性能指标。然后，对自环、层数、DropEdge、PairNorm、激活函数进行调整，再重新训练、测试，并分析对模型性能的影响。
- 测试性能**：选择你认为最合适的（例如，在验证集上表现最好的）一组超参数，重新训练模型，并在测试集上测试（注意，这理应是你的实验中**唯一**一次在测试集上的测试），并记录测试的结果（ACC）。

实验提交

本次实验截止日期为 **1月7日 23:59:59**，需提交代码源文件及实验报告到邮箱：ustcdl2023@163.com，具体要求如下：

1. 本次实验没有示例代码，需要自行完成数据处理，模型搭建整个pipeline
2. 全部文件打包在一个压缩包内，压缩包命名为 学号-姓名-exp4.zip
3. 实验报告要求 pdf 格式，要求包含姓名、学号。内容包括简要的**实验过程**和**关键代码**展示，对超参数的**实验分析**，最优超参数下的训练集、验证集**损失曲线**以及测试集上的**实验结果**。