# 基于 Cora 数据集的 Graph Transformer 对比学习实验

## 一.项目目标

本项目旨在基于 Cora 引文网络数据集，利用图神经网络（GNN）和 Graph Transformer 结构，结合节点的 BERT 语义特征和边的多关系概率特征，实现节点分类任务，并引入对比学习机制提升模型泛化能力。项目目标包括：

（1）利用 BERT 模型提取节点文本特征，增强节点表示能力

（2）利用多关系边特征（边概率/边关系类型）丰富图结构信息

（3）设计带边特征的 Graph Transformer 层，实现节点和边的联合特征学习

（4）通过对比学习进行预训练，提升模型鲁棒性和泛化能力

（5）对节点进行准确分类，并评估模型在不同数据划分下的表现

## 二.数据集说明

### 1.Cora 数据集

Cora 数据集是一个经典的引文网络数据集，常用于图神经网络和节点分类任务的基准测试。它包含机器学习领域的学术论文及引用关系。

| Cora 数据集 | 说明 |
| --- | --- |
| 领域 | 机器学习学术论文 |
| 论文数量（节点数） | 2708 |
| 类别数量（分类数） | 7 |
| 引用链接（边数量） | 5429 |
| 特征维度 | 1433 |

（1）节点特征

每篇论文表示为一个节点，每个节点都有 Title 和 Abstract 的文本信息。

（2）引用关系

有向边表示论文之间的多种关系，本实验发现论文之间不只是有引用关系，也有理论关联、直接引用、矛盾/挑战、实证支持、方法重叠等关系类型。因此，我们采用这五种关系类型作为多边关系类型，进行分类任务。

（3）类别标签

| 类别 | 说明 |
| --- | --- |
| Case_Based | 基于案例的推理 |
| Genetic_Algorithms | 遗传算法 |
| Neural_Networks | 神经网络 |
| Probabilistic_Methods | 概率方法 |
| Reinforcement_Learning | 强化学习 |
| Rule_Learning | 规则学习 |
| Theory | 理论 |

2.cora_feat768.csv

每一行对应一个节点（论文），包含其唯一 ID 和 BERT 提取的 768 维特征向量。

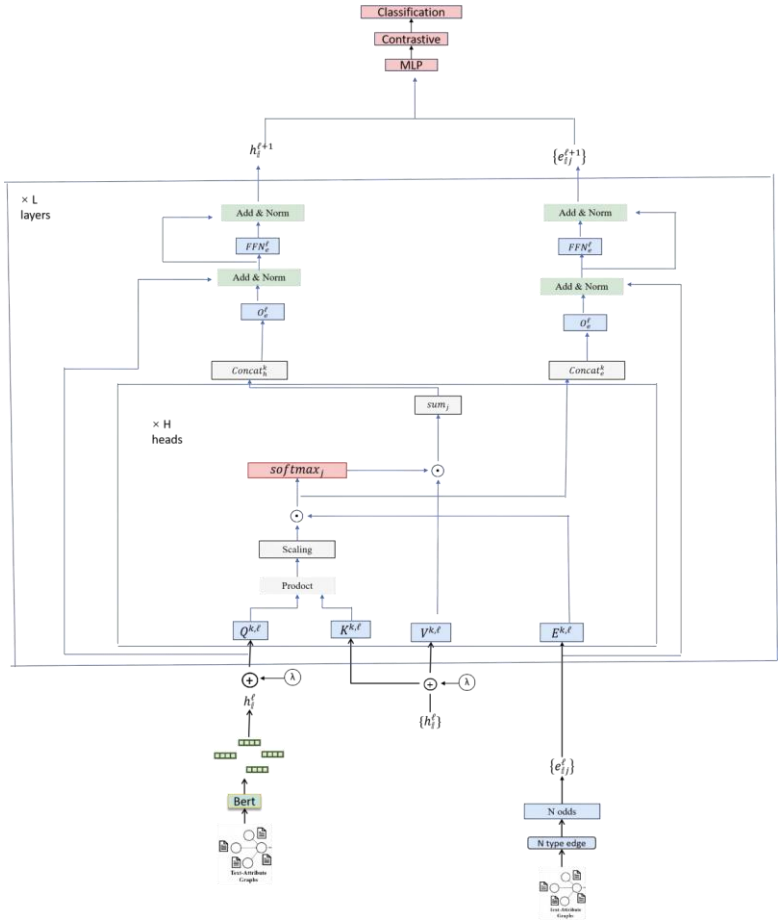
3.egge_odds_10.json

每个元素为一条边。

| 符号 | 说明 |
| --- | --- |
| source/target | 节点 ID |
| relationslist | 多关系类型（用 0，1，2，3，4 代表五种关系类型，-1 代表不存在该关系类型） |
| edge_odds | 每种关系的概率分布 |

## 三.实现思路

1.项目结构模型流程图




2.节点特征提取模块

使用 BERT 模型对每篇论文的标题和摘要进行编码，生成 768 维的节点特征向量。实现文件：cora_bert.py

输入数据是每个节点（即每篇论文）的论文标题和摘要文本。

处理流程：

（1）使用预训练的 BERT-base 模型（768 维隐藏层）

（2）对每篇论文的"[CLS]标题[SEP]摘要[SEP]"结构进行编码

（3）取[CLS]位置的输出作为整个文档的表示

（4）将特征保存为 cora_feat768.csv（形状：2708×768）

3.边特征与多关系建模

通过数据集，加载每条边的多关系类型和概率分布，将概率最大的类型作为边的显示特征输入模型

数据文件：egge_odds_10.json

处理方式：

（1）将离散关系类型转换为 one-hot 编码

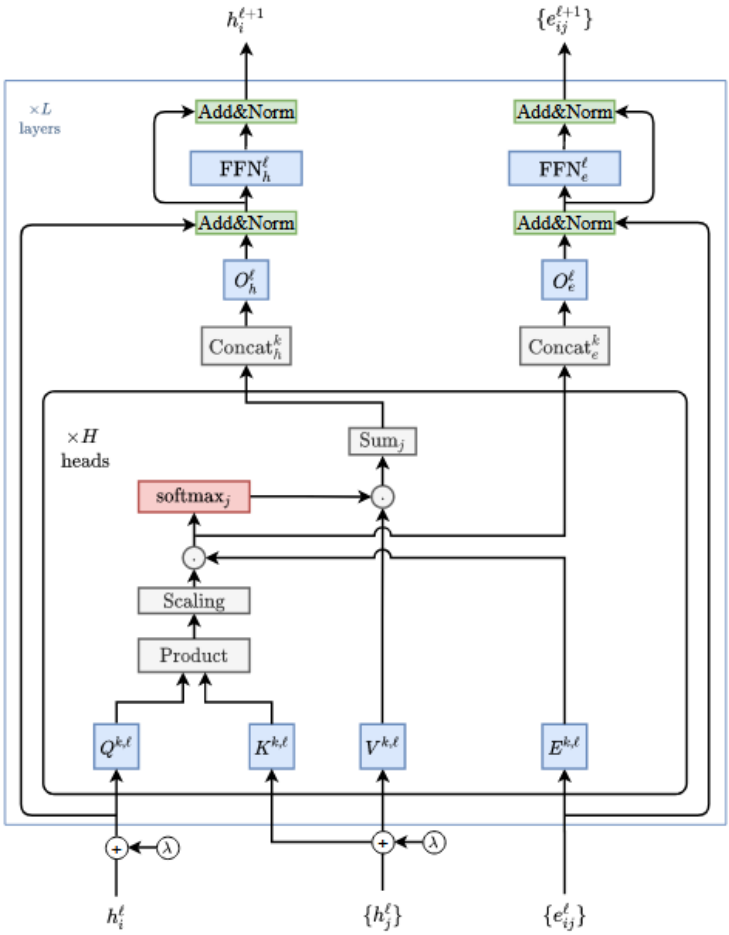
（2）将 edge_odds 作为连续特征

（3）拼接得到最终边特征向量（维度取决于关系类型数量）

4.Graph Transformer 结构设计

设计支持边特征的多头注意力机制，将节点特征、边特征和注意力机制结合，实现节点和边的联合特征学习。其中，节点特征是用 BERT 模型处理文本信息后得到的 768 维的特征向量，边特征是用 json 文件中边概率值取最大作为边特征向量。将特征向量传入到模型中，通过图结构进行特征更新。Graph Transformer 能够捕获复杂的节点和边之间的关系，为后续的分类任务提供更丰富的特征表示。更新后的节点特征被输入到一个多层感知机（MLP）中，以进一步进行特征变换和提炼。

实现文件：cora_GCL_odds_L.py



5.对比学习预训练

在分类层之前，加入一个投影层，这一步是为了进行图对比学习。通过随机丢弃部分边（随机删除 20%的边）生成增强视图，采用对比损失函数，使模型学习到更鲁棒的节点表示。在训练过程中，首先进行预训练任务，然后进行节点分类任务。

6.节点分类与评估

采用 MLP 对节点嵌入进行分类，使用评估器对训练集、验证集、测试集进行准确率评估。MLP 层通过多个线性层对输入数据进行处理，并在每层后应用 ReLU 激活函数。

注：可以通过修改模型层数，训练轮数 run，训练迭代次数 epoch，学习率 lr，来实现更准确的节点分类任务，不断提高分类准确率。

## 四.核心代码结构

1.cora_bert.py

生成节点 BERT 特征（cora_feat768.csv）

2.logger.py

训练日志与结果统计。

3.cora_GCL_odds_L.py

| | |
|---|---|
| （1）加载节点、边特征 | parse_cora_with_edge_features() |
| （2）Graph Transformer 结构 | GraphTransformerNet/Layer () |
| （3）对比学习 | pretrain/contrastive_loss() |
| （4）训练与评估 | train/test() |
| （5）主流程 | main() |