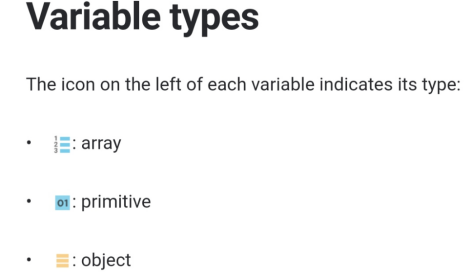
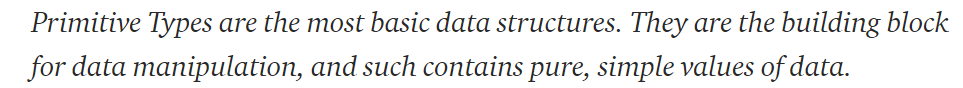
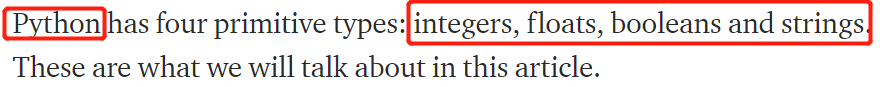
**Pycharm Debug符号解释:**



**Primitive 原始数据结构类型**





改造思路:

数据结构对齐训练模块所需数据格式

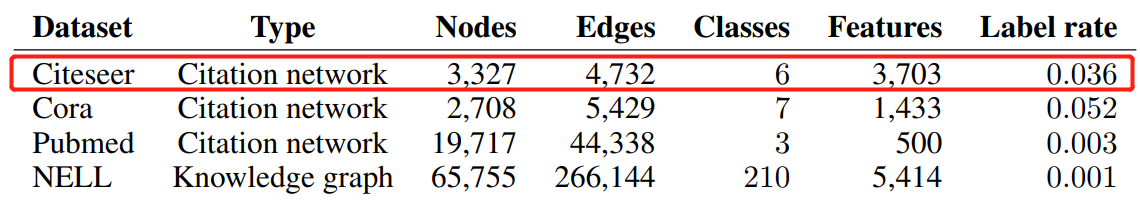
1.注意数据集划分

Reinforcement\_Learning\_NAS

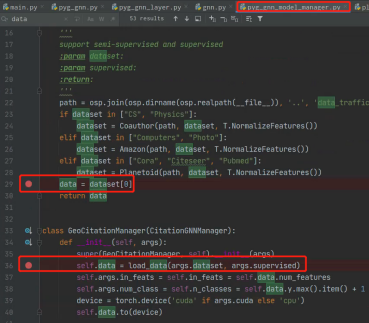
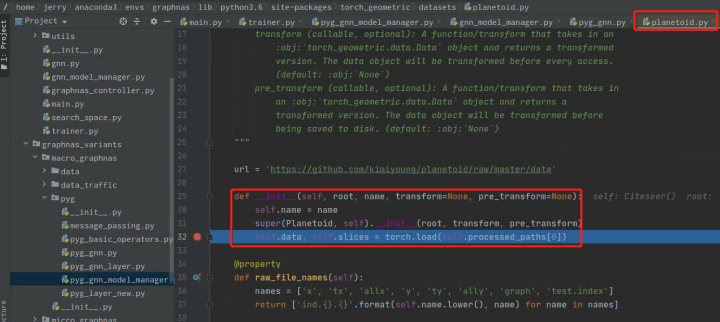
数据配置的重点

**1.图数据的节点编号要从0开始**

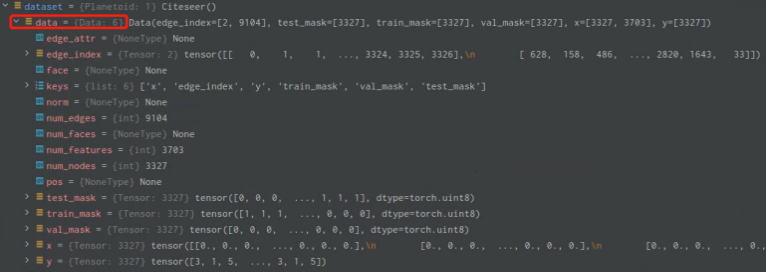
原始数据集信息：Citeseer



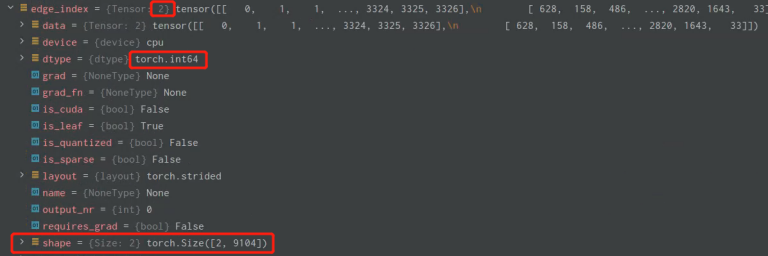
1.数据获模块

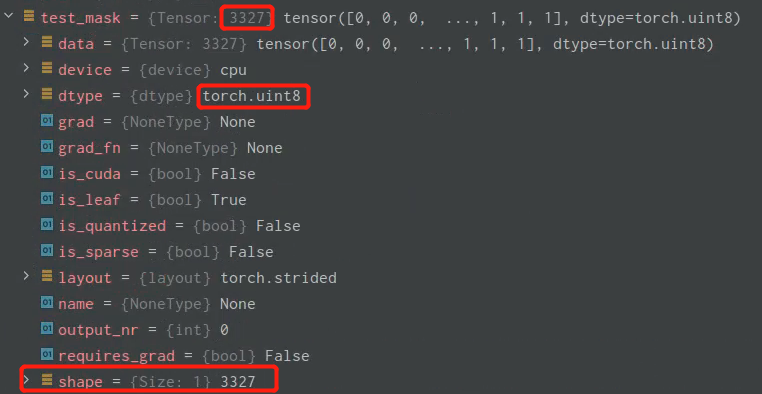
**1.1.data数据结构：**



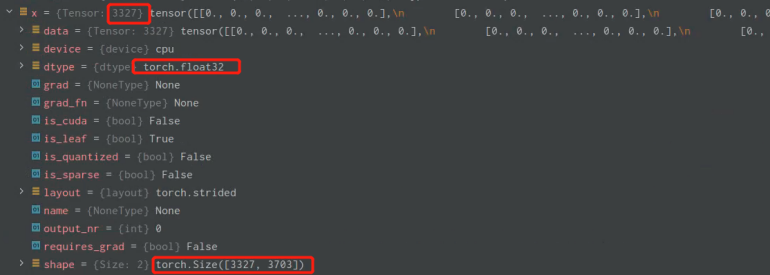
1.1.1.Edge\_index结构：



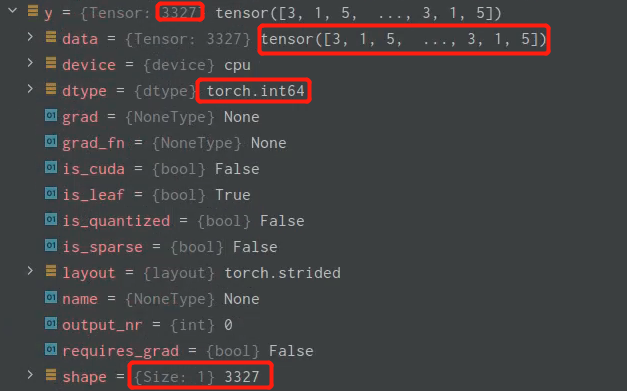
1.1.2.test\_mask数据结构



1.1.3.x数据结构

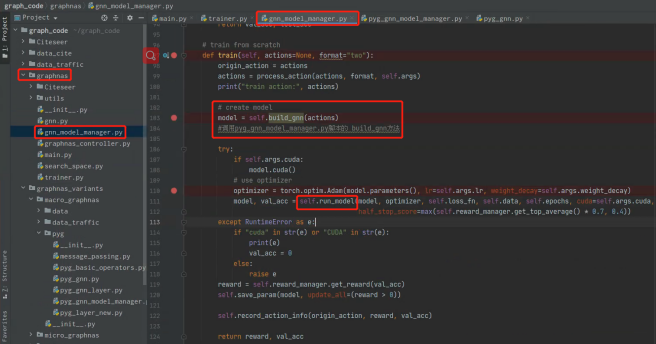


1.1.4.y数据结构



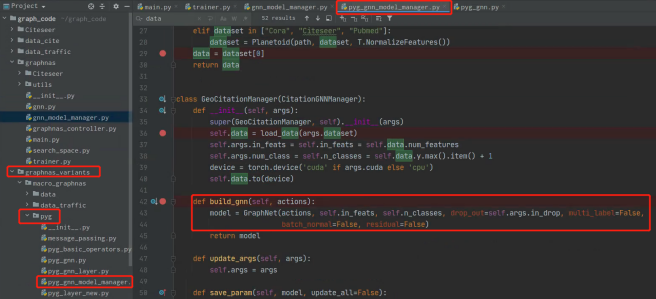
2.训练模块

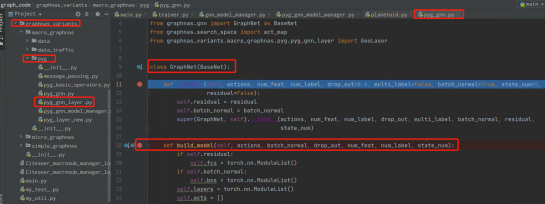
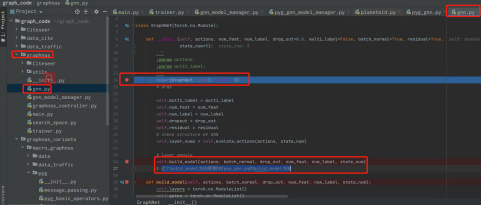
2.1.模型训练主控逻辑：



模型构建：

入口:

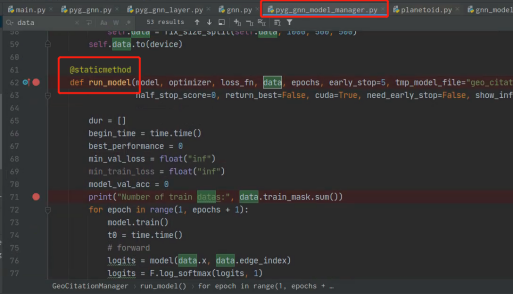


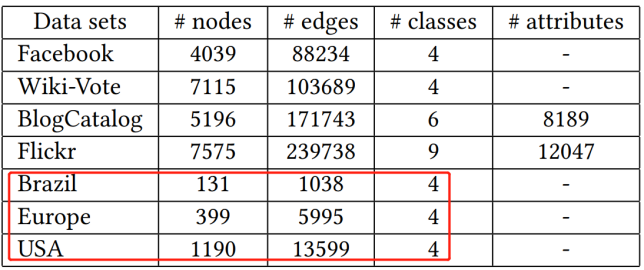
 

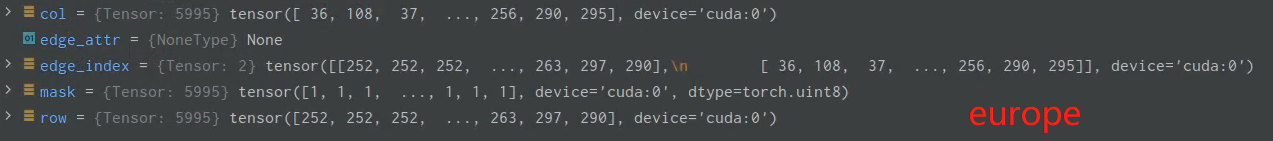
1.super方法调用父类GraphNet的\_\_init\_\_方法

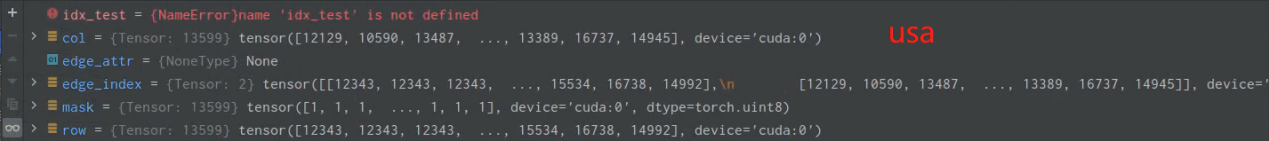
2.GraphNet中的self.build\_model调用子类(GraphNe(BaseNet))自己的build\_model方法

模型训练模块：

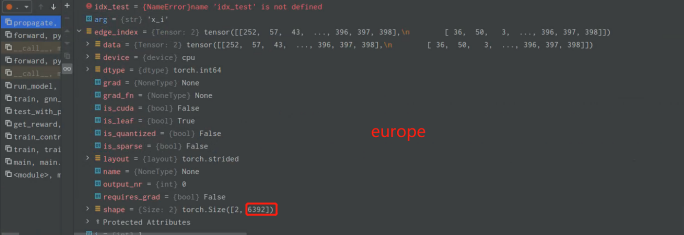


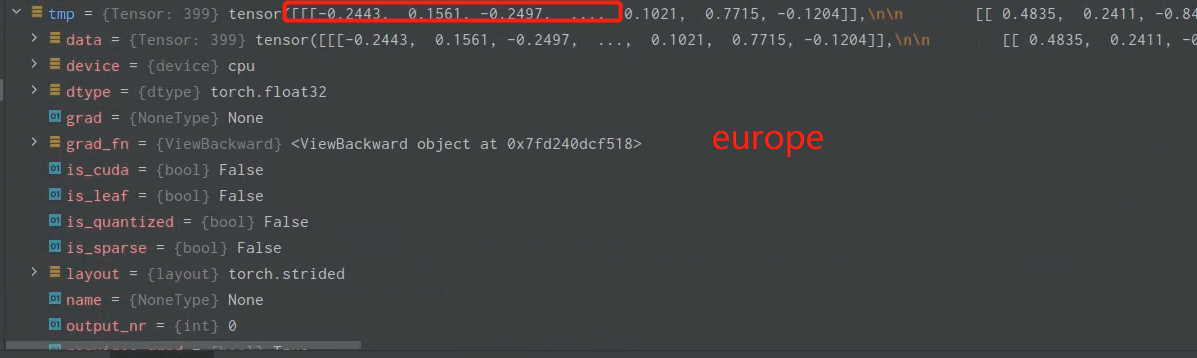




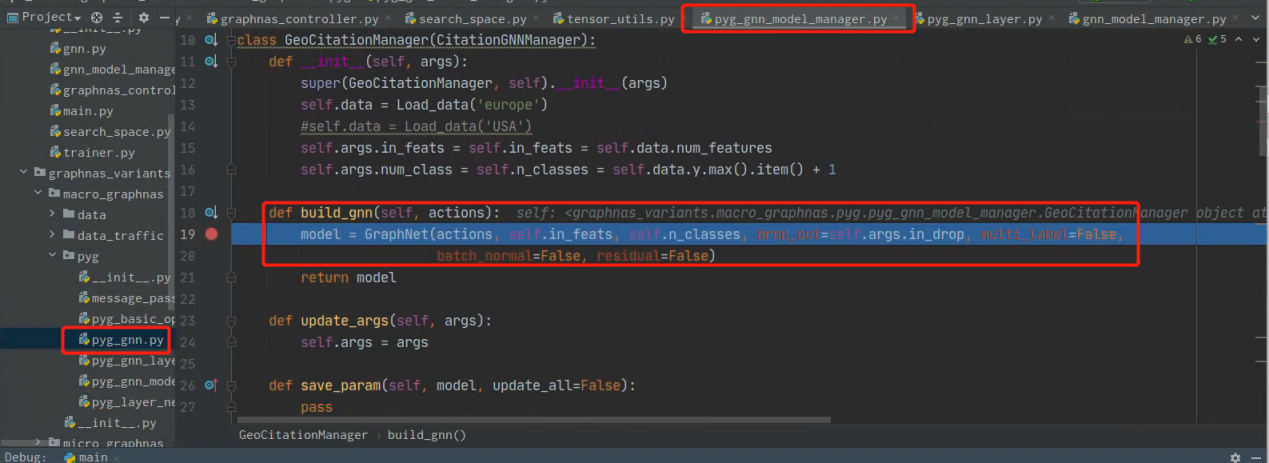


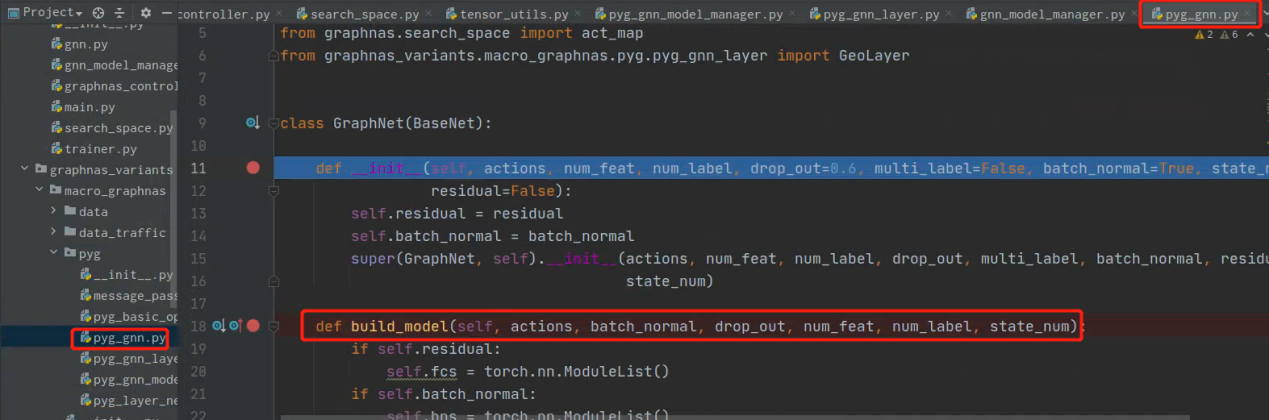




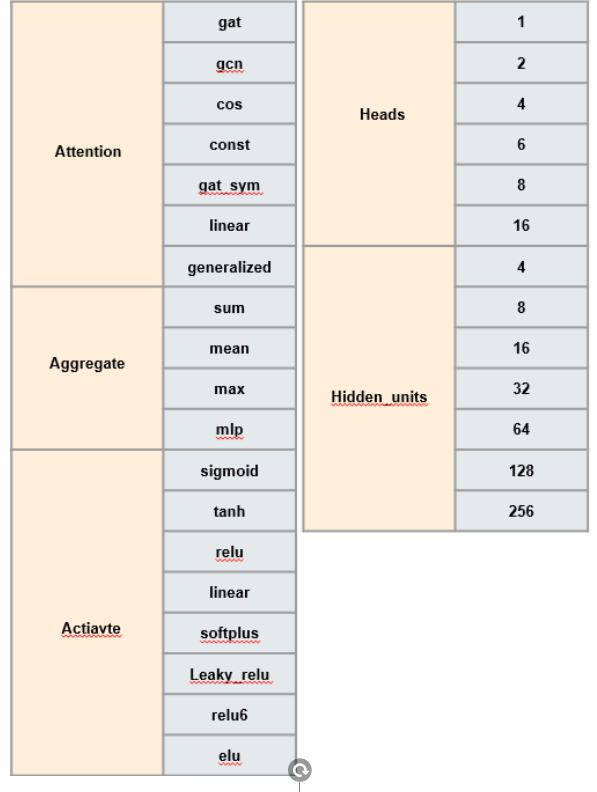


建模模块：



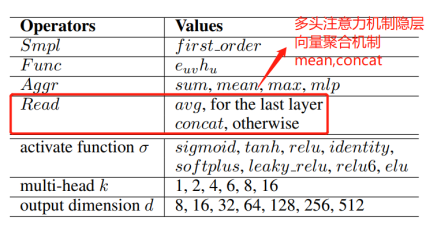


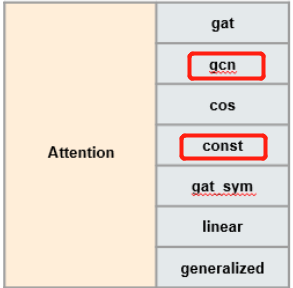
代码解析：



1.attention

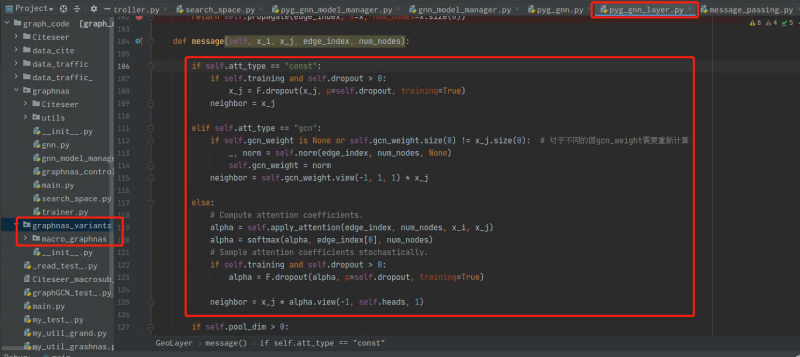
ijcai发表论文中search space定义：



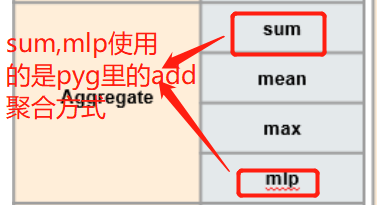


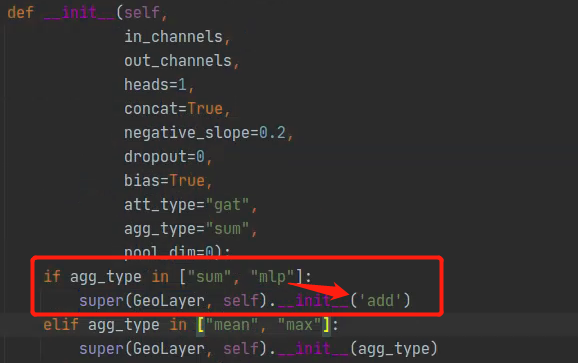


代码路径：

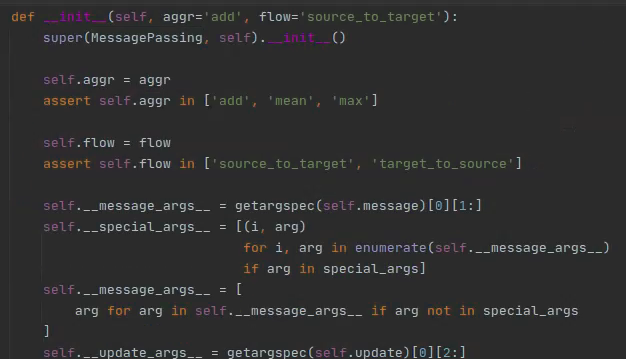
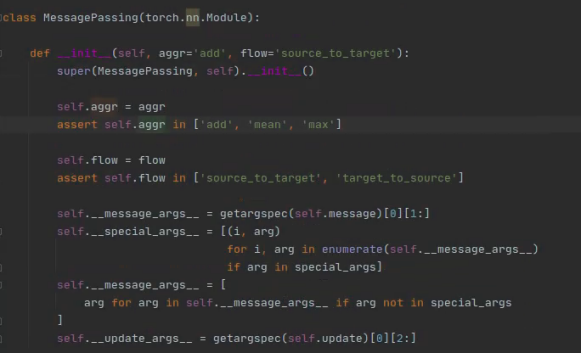


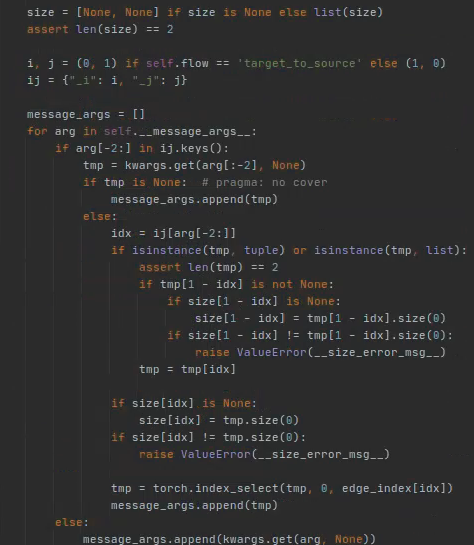
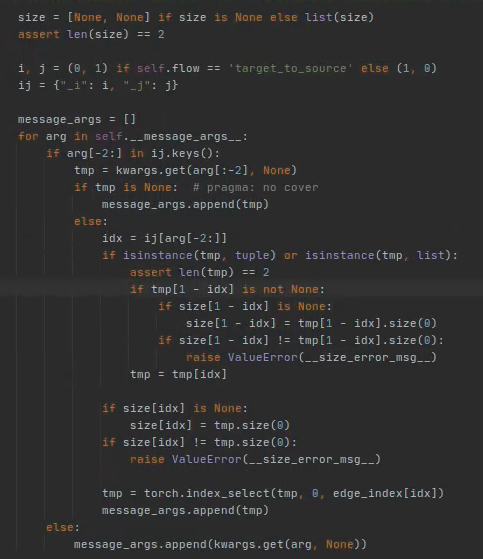
2.aggregate

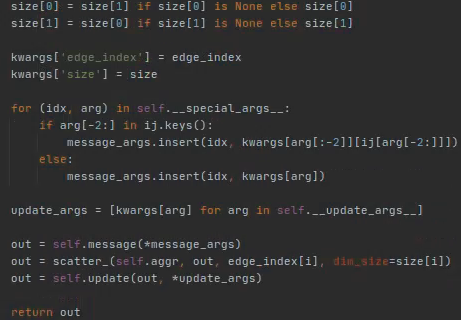
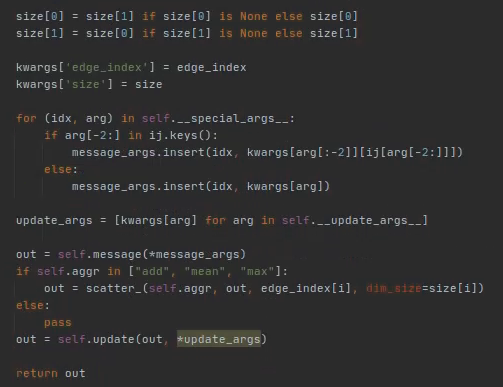


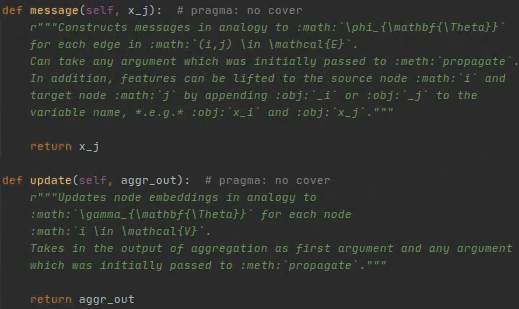
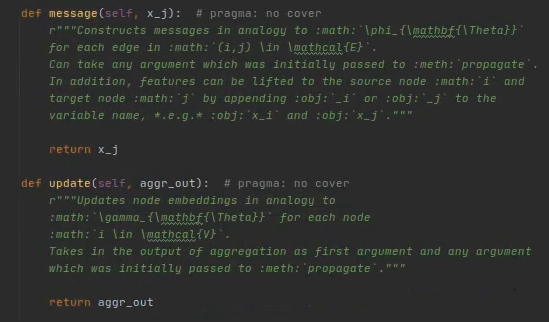


Graphnas message : Pyg message:



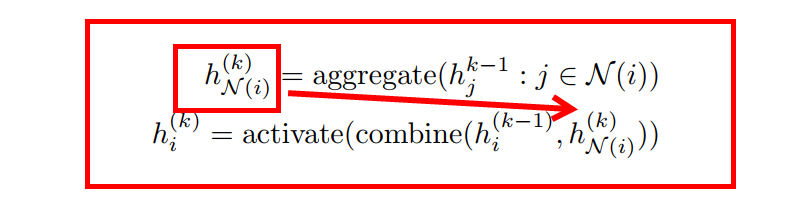


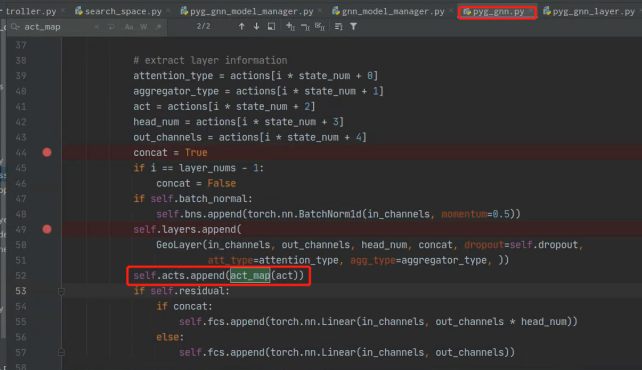


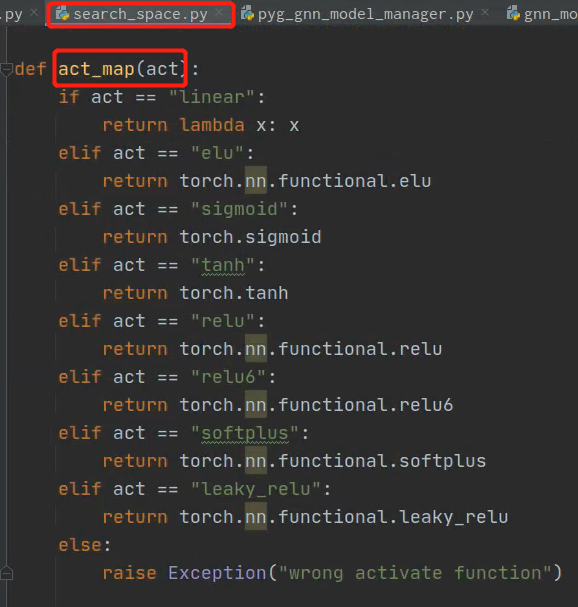


**结论：graphnas message 类与pyg message没有区别**

**3.activation：对节点i聚合后的表示向量使用激活函数，表示向量的每一个值使用激活函数后仍然得到一个向量**

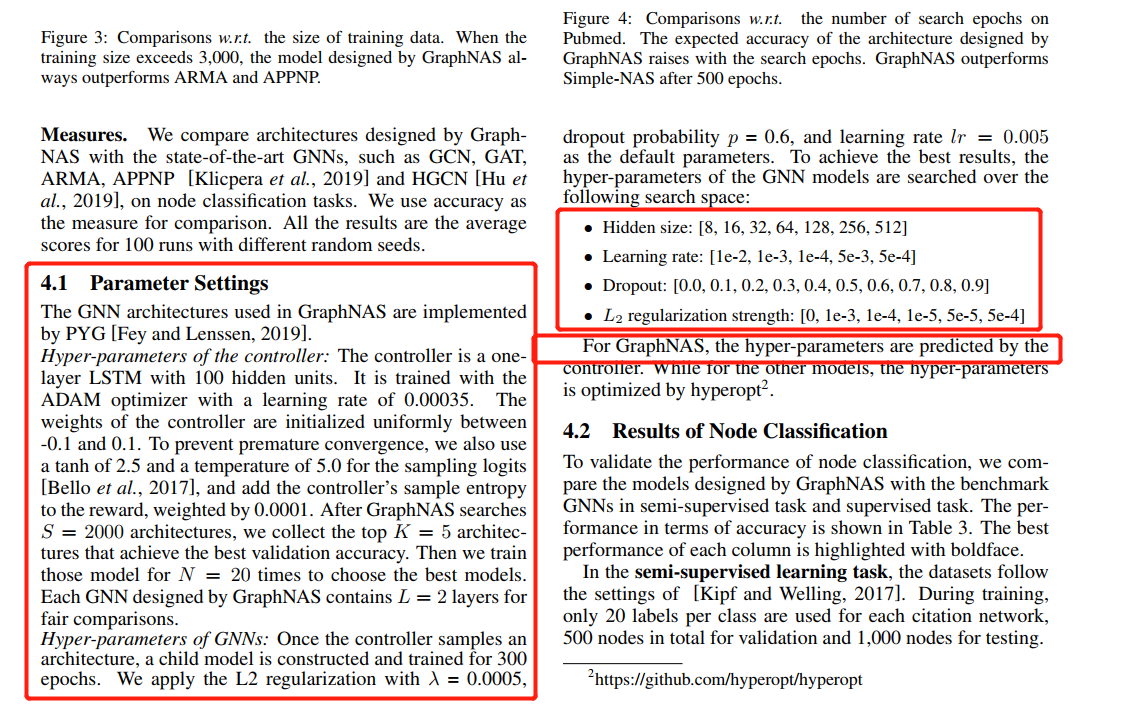






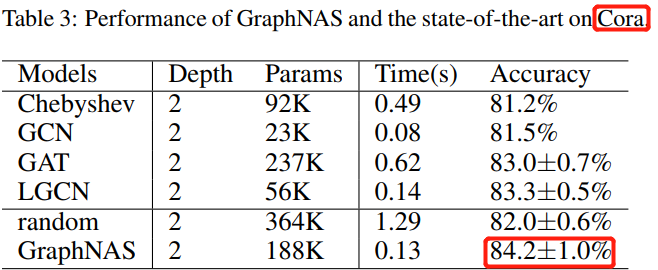
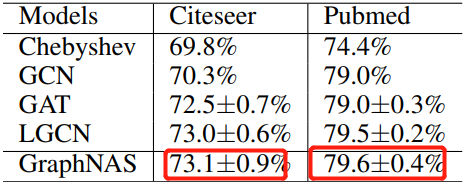
**发表论文中超参也在搜索空间中：**

Graphnas考虑了超参搜索（github代码没有体现）：

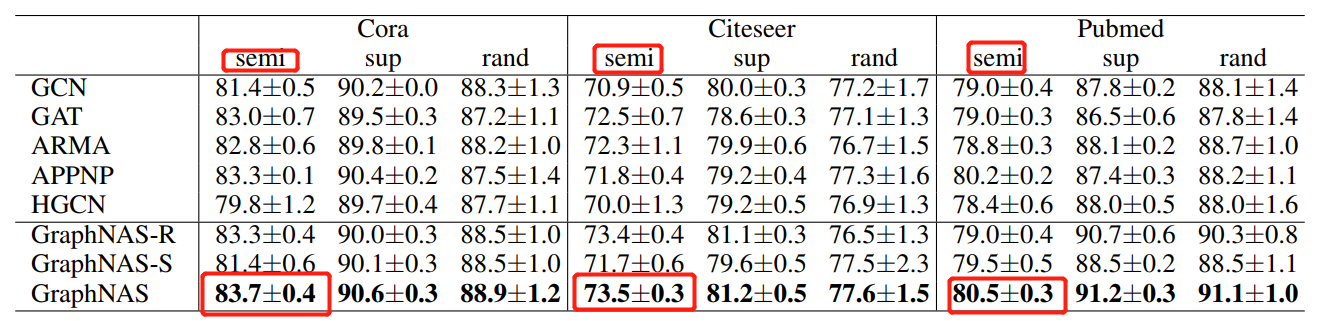


1.将gnn\_build\_module模块剥离

arvix结果：



ijcai结果：



github结果：

