

# 图表示学习实验

## Part 1 图节点表示实验

在第一部分实验，我们将编写\*节点表示学习\*的整个流程，我们将进行3步

第一步 我们先加载经典社交网络图 [Karate Club Network](#). 并探索这个图上的多种统计信息。

接着我们尝试将图结构转化成tensor形式，为我们进一步在图上进行机器学习算法做准备

最后，我们将实现首个图算法：一个图节点表示模型。我们简化了课上学习的deepwalk/node2vec算法，但也同样具有挑战性，我们将从使用pytorch学习开始，从头实现这个算法，你可以自己尝试实现更复杂的deepwalk/node2vec获取加分。

这部分实验分为六个题目，你需要依次补全缺失的代码块，完成图节点表示学习任务

- Question 1: karate club network的 average degree(平均度)是多少(5分)
- Question 2: karate club网络的平均聚类系数是多少k? (5分)
- Question 3: 在一次PageRank迭代之后，节点0 (id为0的节点)的PageRank值是多少? (5 Points)
- Question 4: 获取karate club network的边列表并将其转换为 `torch.LongTensor` `**` `pos_edge_index` tensor的 `torch.sum` 值是多少? (10 分)
- Question 5: 请实现以下对负边进行采样的函数。然后回答哪些边(edge\_1到edge\_5)是karate club network中的负边?(10分)
- Question 6: 训练表示：你能得到的最好表示是什么?请在实验报告上记录最佳损失和准确性。(20分)

注意: 确保 顺序运行每个section的代码, 这样所有临时变量或包才能被保留到下个cell

## Setup

这个实验中，我们需要大量使用network这个包

```
import networkx as nx
```

我们使用 Karate club network(空手道俱乐部)这一经典社交网络图，[Karate Club Network](#)是一个社交网络图，它描述了一个由34个空手道俱乐部成员组成的社会网络，并记录了在俱乐部外互动的成员之间的联系。

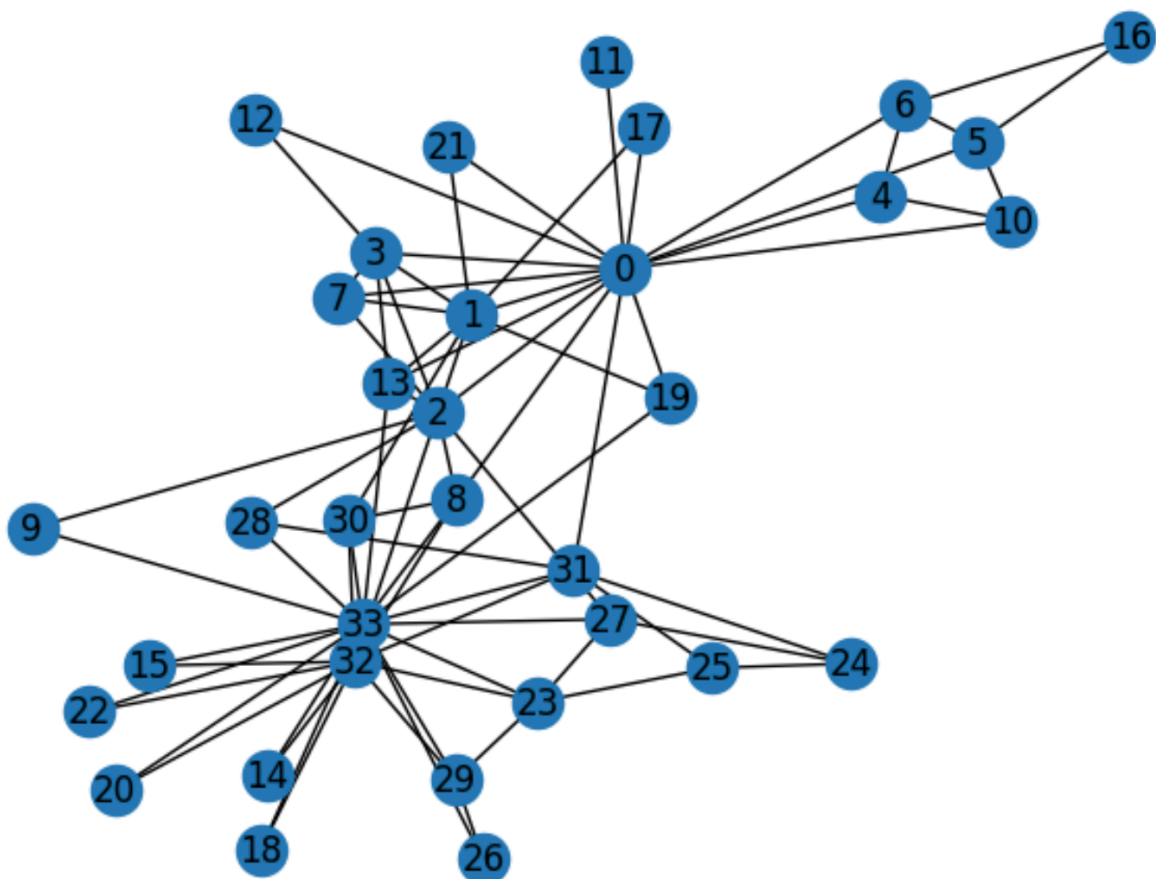
```
G = nx.karate_club_graph()

# G is an undirected graph
type(G)
```

图节点可视化

```
G = nx.karate_club_graph()

# G is an undirected graph
type(G)
# Visualize the graph
nx.draw(G, with_labels = True)
```



## Part 2 GNN

在Part 2中，我们将使用PyTorch Geometric (PyG)构建我们自己的图神经网络，然后将该模型应用于两个 Open graph Benchmark (OGB)数据集。这两个数据集将用于在两个不同的基于图的任务上对模型的性能进行基准测试：

- 1)节点属性预测，预测单个节点的属性；
- 2)图属性预测，预测整个图或子图的属性。

首先，我们将学习PyTorch Geometric如何将图存储为PyTorch张量。

然后，我们将使用 `ogb` 包加载和检查开放图形基准(OGB)数据集之一。OGB是一个现实的、大规模的、不同的基准数据集的集合，用于图上的机器学习。`ogb` 包不仅为每个数据集提供数据加载器，还提供模型评估器。

最后，我们将使用PyTorch Geometric构建我们自己的图神经网络。然后，我们将在OGB节点属性预测和图属性预测任务上训练和评估我们的模型。

这部分实验分为六个题目，你需要依次补全缺失的代码块，完成图表示学习任务

- Question 1: ENZYMES数据集中的类和特征的数量是多少?(5分)
- Question 2: 在ENZYMES数据集中索引为100的图的标签是什么?(5分)
- Question 3: 索引为200的图有多少条边?(5分)
- Question 4: 在ogbn-arxiv图中有多少特征?(5分)
- Question 5: 你的 `best_model` 验证和测试精度是多少?(20分)
- Question 6: 你的 `best_model` 验证和测试ROC-AUC分数是多少?(20分)

- **Question 7 (可选):**在Pytorch Geometric中使用另外两个全局池化层进行实验。

**\*注意\*:** 确保 **\*顺序运行每个section的代码\***, 这样所有临时变量或包才能被保留到下个cell

## Device

可能需要gpu来加速图神经网络训练, 可以在[google colab](https://colab.research.google.com/)上运行我们的脚本, 里面有免费gpu

点击 `Runtime` 然后点击 `Change runtime type`. 将 `hardware accelerator` 设置为 **\*GPU\***.

## Setup

如果你使用colab上的gpu, 则需要在Colab上安装PyG。首先让我们检查你正在运行的PyTorch的版本

```
import torch
import os
print("PyTorch has version {}".format(torch.__version__))
```

下载PyG所需的软件包。确保您的torch版本与上面单元的输出相匹配。如果有任何问题, 可以在[PyG's installation page](https://pytorch-geometric.com/installation/)上找到

```
# Install torch geometric
if 'IS_GRADESCOPE_ENV' not in os.environ:
    torch_version = str(torch.__version__)
    scatter_src = f"https://pytorch-geometric.com/whl/torch-{torch_version}.html"
    sparse_src = f"https://pytorch-geometric.com/whl/torch-{torch_version}.html"
    !pip install torch-scatter -f $scatter_src
    !pip install torch-sparse -f $sparse_src
    !pip install torch-geometric
    !pip install ogb
```