

# 图神经网络实验

学号：3200102555

专业班级：计科2006

姓名：李云帆

性别：男

## Project Introduction

### 选题

- 了解图神经网络相关知识
- 了解如何使用MindSpore搭建并训练图神经网络

### 工作简介

图卷积网络(Graph Convolutional Network, GCN)是近年来逐渐流行的一种神经网络结构。不同于只能用于网格结构(grid-based)数据的传统网络模型 LSTM 和 CNN，图卷积网络能够处理 具有广义拓扑图结构的数据，并深入发掘其特征和规律。

本实验主要介绍在下载 Cora 和 Citeseer 数据集上使用 MindSpore 进行图卷积网络的训练。

### 开发环境

ModelArts Ascend Notebook 环境，MindSpore1.1.1

## Technical Details

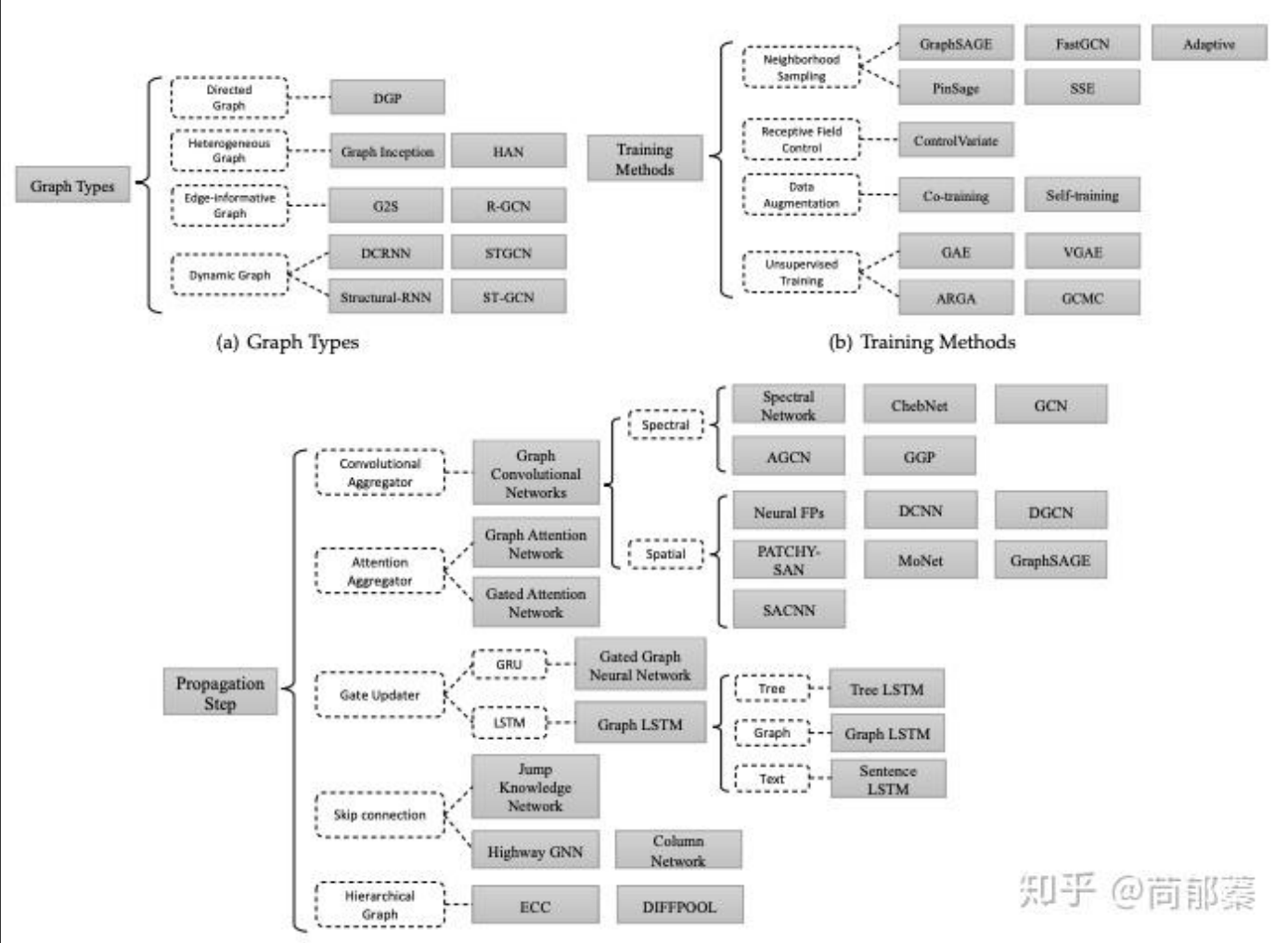
### 理论知识

GCN 的本质目的就是用来提取拓扑图的空间特征。图卷积神经网络主要有两类，一类是基于 空间域(spatial domain)或顶点域(vertex domain)的，另一类则是基于频域或谱域(spectral domain)的。GCN 属于频域图卷积神经网络。

# 算法

空间域方法直接将卷积操作定义在每个结点的连接关系上，它跟传统的卷积神经网络中的卷积 更相似一些。在这个类别中比较有代表性的方法有 Message Passing Neural Networks(MPNN), GraphSage, Diffusion Convolution Neural Networks(DCNN), PATCHY-SAN 等。

频域方法希望借助图谱的理论来实现拓扑图上的卷积操作。从整个研究的时间进程来看:首先 研究 GSP(graph signal processing)的学者定义了 graph 上的傅里叶变化(Fourier Transformation)，进而定义了 graph 上的卷积，最后与深度学习结合提出了 Graph Convolutional Network(GCN)。



知乎 @尚郁秦

## 技术细节

- `train_eval(args_opt):`
  - 接受包含各种超参数的对象 `args_opt`
  - 首先设置随机数生成器的种子
  - 加载数据集的邻接矩阵, 特征矩阵和标签矩阵

- 使用 `get_mask()` 函数将数据集分成训练, 验证和测试集
- 使用给定的超参数初始化GCN模型
- 将模型包装在 `LossAccuracyWrapper` (计算模型在验证集和测试集上的损失和准确性)和 `TrainNetWrapper` (在训练集训练模型)中
- 然后, 训练模型一定数量的 `epochs` 并打印出各种信息

## Experiment Result

```
2.2 min #训练
print("===== Starting Training =====")
train_eval(cfg)

===== Starting Training =====
[WARNING] KERNEL(3938,7f7bf0a88740,python):2023-05-18-19:04:14.473.259 [mindspore/ccsrc/plugin/device/gpu/kernel/gpu_kernel_factory.cc:117] CheckSM It is recommended to use devices with a computing capacity >= 7, but the current device's computing capacity is 6
Epoch: 0000 train_loss= 1.95315 train_acc= 0.68571 val_loss= 1.94889 val_acc= 0.33000 time= 127.91136
Epoch: 0010 train_loss= 1.86385 train_acc= 0.84286 val_loss= 1.90580 val_acc= 0.49800 time= 0.00400
Epoch: 0020 train_loss= 1.75179 train_acc= 0.87143 val_loss= 1.86120 val_acc= 0.52000 time= 0.00349
Epoch: 0030 train_loss= 1.60819 train_acc= 0.88571 val_loss= 1.80630 val_acc= 0.55400 time= 0.00356
Epoch: 0040 train_loss= 1.45687 train_acc= 0.90000 val_loss= 1.74394 val_acc= 0.59200 time= 0.00345
Epoch: 0050 train_loss= 1.28385 train_acc= 0.94286 val_loss= 1.67321 val_acc= 0.64800 time= 0.00317
Epoch: 0060 train_loss= 1.17733 train_acc= 0.94286 val_loss= 1.59864 val_acc= 0.72600 time= 0.00304
Epoch: 0070 train_loss= 1.06520 train_acc= 0.94286 val_loss= 1.52355 val_acc= 0.76600 time= 0.00301
Epoch: 0080 train_loss= 0.93439 train_acc= 0.98571 val_loss= 1.45302 val_acc= 0.77600 time= 0.00302
Epoch: 0090 train_loss= 0.88814 train_acc= 0.97857 val_loss= 1.39816 val_acc= 0.77600 time= 0.00301
Epoch: 0100 train_loss= 0.80983 train_acc= 0.97857 val_loss= 1.33890 val_acc= 0.77800 time= 0.00301
Epoch: 0110 train_loss= 0.73957 train_acc= 0.98571 val_loss= 1.29790 val_acc= 0.78400 time= 0.00309
Epoch: 0120 train_loss= 0.73278 train_acc= 0.97857 val_loss= 1.26075 val_acc= 0.78600 time= 0.00300
Epoch: 0130 train_loss= 0.68259 train_acc= 0.98571 val_loss= 1.21996 val_acc= 0.78600 time= 0.00302
Epoch: 0140 train_loss= 0.64416 train_acc= 0.98571 val_loss= 1.18979 val_acc= 0.78400 time= 0.00314
Epoch: 0150 train_loss= 0.61749 train_acc= 0.99286 val_loss= 1.16584 val_acc= 0.78400 time= 0.00307
Epoch: 0160 train_loss= 0.58801 train_acc= 0.99286 val_loss= 1.14290 val_acc= 0.78000 time= 0.00308
Epoch: 0170 train_loss= 0.54961 train_acc= 0.99286 val_loss= 1.11940 val_acc= 0.78200 time= 0.00320
Epoch: 0180 train_loss= 0.52463 train_acc= 0.97857 val_loss= 1.10643 val_acc= 0.78400 time= 0.00306
Epoch: 0190 train_loss= 0.50878 train_acc= 1.00000 val_loss= 1.08094 val_acc= 0.78600 time= 0.00310
Test set results: loss= 1.01657 accuracy= 0.81800 time= 0.18325
```

## References

1. 知乎-一文读懂图卷积GCN
2. Slides on class
3. 图神经网络实验手册