文件目录

- GCN-ForceDirected
  - dymgcn-worked-date1.07（capsule-gcn节点分类代码以及利用capsule-gcn逼近力导引算法）
    - data（cora、citeseer、pumbed数据）
    - Primary school（数据集）
    - calpos.py（利用力导引算法计算图的可视化结果）
    - capnets.py（hinton-capsnet部分核心源码）
    - draw.py（用于可视化计算结果）
    - force.py（利用capsule-gcn逼近力导引算法）
    - graphlet_count.py（计算不同graphlet的出现频率）
    - layers.py（不同神经网络层的代码，包括dense、gcn、capsule-gcn）
    - metrics.py（loss函数、评价指标）
    - paper_exp.py（capsule-gcn接点分类代码）
    - train.py（原gcn的train文件）
    - utils.py（数据处理代码）
  - force-directed-graph（力导引布局算法源码）
    - cal_pos.py（生成图网络数据，并计算其可视化结果）
    - force_directed_graph.py和force_utils.py（力导引算法的代码）
  - graphlet_counting（计算图基元源码）
  - graphnet_force（利用graphnet拟合力导引布局算法的代码）
    - train.py（训练文件）
    - utils.py（工具文件）
    - visual.py（画图函数）

参数

paper_exp.py和force.py

- model1：gcn、gcn_cheby、dense、dr_gcn_cheby、dr_gcn 模型
  - 若使用gcn gcn_cheby dense，需要将paper_exp.py中下面部分更改

```
#model1 = model_func1(placeholders1, adj=adj,
input_shape=features[2], logging=logging, act=tf.nn.relu)
model1 = model_func1(placeholders1, features[2][1],
logging=logging)
```

  - 若使用dr_gcn等，需要将paper_exp.py中下面部分更改

```
model1 = model_func1(placeholders1, adj=adj,
input_shape=features[2], logging=logging, act=tf.nn.relu)
#model1 = model_func1(placeholders1, features[2][1],
logging=logging)
```

- dataset：cora、citeseer、pubmed、primaryschool 数据集

- dist_type：cos、eud casule的相似性计算方式
- learning_rate：学习率
- epochs：训练轮数
- hidden1：隐藏层特征为度
- dropout：dropout rate
- weight_decay：loss函数中的L2项的权重系数
- max_degree：chebyshev的K跳值

数据划分方式

按GCN原论文的打标签方式：

```
去掉utils.py的88-93行的注释，将96行注释
```

随机抽样下不同标签率的打标签方式：

```
注释utils.py的88-93行，去掉96行的注释
其中标签率通过paper_exp.py的load_data参数0.01来更改标签率
adj, features, y_train, y_val, y_test, train_mask, val_mask, test_mask =
load_data(FLAGS.dataset, rate, 0.01)
```

dymgcn-worked-date1.07/gcn/layer.py

- class Layer 所有层的父类
- class Dense 全连接层
- class GraphConvolution GCN
- class DRLayer + class AfterDR_GraphConvolution 二者要连着用，DRLayer在前，AfterDR_GraphConvolution在DRLayer的下一层，作用等同于GCN+Capsule-GCN

dymgcn-worked-date1.07/gcn/models.py

- class Model 所有模型的父类
- class MLP 两层全连接层组成的模型
- class GCN 两层GCN组成的模型
- class DRGCN GCN+Capsule-GCN
- class FORCE_DRGCN 用于逼近力导引算法的Capsule-GCN

节点分类实验

直接运行paper_exp.py

Capsule-GCN拟合力导引算法

直接运行force.py