Lab4 GCN

和泳毅 PB19010450

一、实验目标

1. 熟悉图卷积神经网络的基本原理
2. 了解网络层数对图卷积神经网络性能的影响
3. 了解不同激活函数，self loop, DropEdge, PairNorm等技术对图卷积神经网络性能的影响。

二、数据集介绍

Dataset	Nodes	Edges	Classes	Features
Citeseer	3327	4732	6	3703
Cora	2708	5429	7	1433

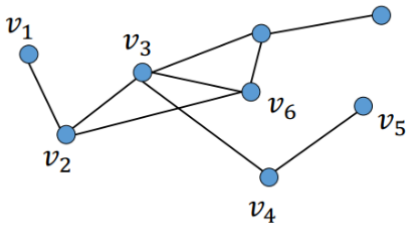
PPI数据集共24张图，平均每张图有2371个节点，共56944个节点818716条边，每个节点特征长度为50，基因本体基作为label(总共121个)，label不是one-hot编码。

三、实验原理

1.图的基本概念

图的矩阵A表示：

- 邻接矩阵$A_{ij}=1$，如果v_i和v_j相邻；
- 度矩阵$D=diag(d(v_1),\dots,d(v_N))$，$d(v_i)=\sum_{v_j\in N_{v_i}}A_{ij}$。



度矩阵

$$D=\begin{pmatrix}1&0&0&0&0&0&0&0\\0&3&0&0&0&0&0&0\\0&0&4&0&0&0&0&0\\0&0&0&2&0&0&0&0\\0&0&0&0&2&0&0&0\\0&0&0&0&0&4&0&0\\0&0&0&0&0&0&3&0\\0&0&0&0&0&0&0&1\end{pmatrix}$$

邻接矩阵

$$A=\begin{pmatrix}0&1&0&0&0&0&0&0\\1&0&1&0&0&1&0&0\\0&1&0&1&0&1&1&0\\0&0&1&0&1&0&0&0\\0&0&0&1&0&1&0&0\\0&1&1&0&1&0&1&0\\0&0&1&0&0&1&0&1\\0&0&0&0&0&0&1&0\end{pmatrix}$$

拉普拉斯矩阵的性质$L = D - A$:

- $LI = DI - AI = d - d = 0$
- $IL = 0$
- L是半正定的，最小特征值为0，其特征向量为1

归一化后的拉普拉斯矩阵:

- 对称归一化: $L^{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$
- 随机游走归一化: $L^{rw} = D^{-1} L$

度矩阵 $\boldsymbol{D}$ − 邻接矩阵 $\boldsymbol{A}$ = 拉普拉斯矩阵 $\boldsymbol{L}$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

2.图傅里叶变换

$L = U \Lambda U^T$为其特征值分解，U的列向量类比于傅里叶变换中的基。

对图上信号z的图傅里叶变换:

$$\mathcal{F}(\lambda_l) = \hat{f}(\lambda_l) = \sum_{i=1}^{n} f(i) u_l(i) = \boldsymbol{u}_l^{\top} \boldsymbol{f}$$

$$\begin{pmatrix} \hat{f}(\lambda_1) \\ \hat{f}(\lambda_2) \\ \vdots \\ \hat{f}(\lambda_N) \end{pmatrix} = \begin{pmatrix} u_1(1) & u_1(2) & \dots & u_1(N) \\ u_2(1) & u_2(2) & \dots & u_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ u_N(1) & u_N(2) & \dots & u_N(N) \end{pmatrix} \begin{pmatrix} f(1) \\ f(2) \\ \vdots \\ f(N) \end{pmatrix} \quad \Rightarrow \quad \hat{\boldsymbol{f}} = \boldsymbol{U}^{\top} \boldsymbol{f}$$

3.图傅里叶逆变换

$L = U \Lambda U^T$为其特征值分解，U的列向量类比于傅里叶变换中的基。

对图上信号z的图傅里叶逆变换:

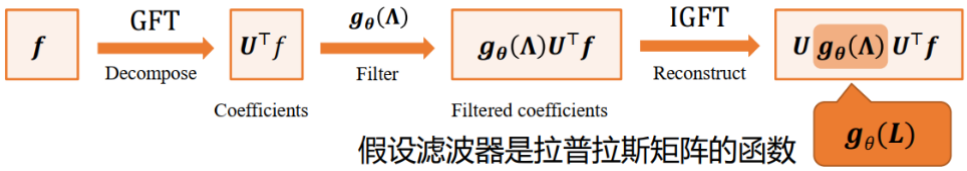
$$f(i) = \sum_{l=1}^{n} \hat{f}(\lambda_l) u_l(i) = \boldsymbol{u}(i)^{\top} \hat{\boldsymbol{f}}$$

$$\begin{pmatrix} f(1) \\ f(2) \\ \vdots \\ f(N) \end{pmatrix} = \begin{pmatrix} u_1(1) & u_2(1) & \dots & u_N(1) \\ u_1(2) & u_1(2) & \dots & u_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(N) & u_2(N) & \dots & u_N(N) \end{pmatrix} \begin{pmatrix} \hat{f}(\lambda_1) \\ \hat{f}(\lambda_2) \\ \vdots \\ \hat{f}(\lambda_N) \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{f} = \boldsymbol{U} \hat{\boldsymbol{f}}$$

任一信号$\boldsymbol{f}$是不同频率的特征向量的组合

4.多项式卷积核

局部化：L对图信号f的操作Lf相当于在图上传播一步



假设滤波器是拉普拉斯矩阵的函数

多项式卷积核：

$$g_\theta(\Lambda) = \sum_{k=0}^{K} \theta_k \Lambda^k$$

$$\hat{g}(\Lambda) = \begin{bmatrix} \sum_{k=0}^{K} \theta_k \lambda_1^k & & & \\ & \sum_{k=0}^{K} \theta_k \lambda_2^k & & \\ & & \cdots & \\ & & & \sum_{k=0}^{K} \theta_k \lambda_N^k \end{bmatrix}$$

切比雪夫多项式，通过如下递归定义：

- $T_0(x) = 1;\ T_1(x) = x$
- $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$

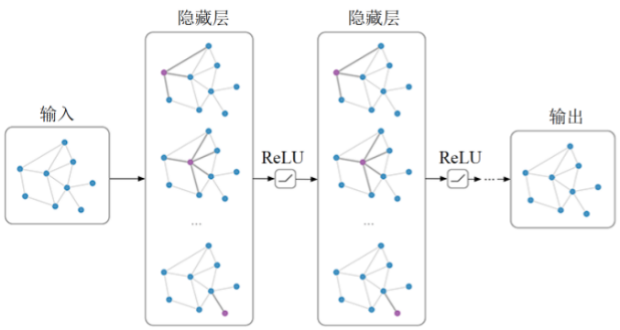
$$\hat{g}(\boldsymbol{\Lambda}) = \sum_{k=0}^{K} \theta_k T_k(\tilde{\boldsymbol{\Lambda}}) \qquad \text{其中 } \tilde{\boldsymbol{\Lambda}} = \frac{2\boldsymbol{\Lambda}}{\lambda_{max}} - \boldsymbol{I}$$

$$\boldsymbol{U}\hat{g}(\boldsymbol{\Lambda})\boldsymbol{U}^\top \boldsymbol{f} = \sum_{k=0}^{K} \theta_k T_k(\tilde{\boldsymbol{L}})\,\boldsymbol{f}, \text{其中 } \tilde{\boldsymbol{L}} = \frac{2\boldsymbol{L}}{\lambda_{max}} - \boldsymbol{I}$$

5.GCN

叠加多层GCN得到最终的模型(常取两层)：

$$\hat{\boldsymbol{Y}} = f(\boldsymbol{X}, \boldsymbol{A}) = \text{Softmax}\left(\hat{\boldsymbol{A}}\text{ReLU}(\hat{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}^0)\boldsymbol{W}^1\right)$$



其中

$$\hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}$$
$$\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_n$$
$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$$

四、核心代码

网络用多层GCN模块堆叠而成：

```
class GCNBlock(nn.Module):
    def __init__(self, in_feats, out_feats, activation=None, pair_norm=False, add_self_loops=False, drop_edge=0):
        super(GCNBlock, self).__init__()
#         self.conv = SAGEConv(in_feats, out_feats,'mean')
        self.conv = GraphConv(in_feats, out_feats)
        self.dropedge = DropEdge(p=drop_edge)
        self.self_loops = AddSelfLoop()
        self.norm = PairNorm()
        self.activation = activation
        self.pair_norm = pair_norm
        self.add_self_loops = add_self_loops
        self.drop_edge = drop_edge

    def forward(self, g, in_feat):
        if self.drop_edge:
            g = self.dropedge(g)
            self.add_self_loops = True
        if self.add_self_loops:
            g = self.self_loops(g)
        h = self.conv(g, in_feat)
        if self.pair_norm:
            h = self.norm(h)
        if self.activation:
            h = self.activation(h)
        return h

class GCN(nn.Module):
    def __init__(self, in_feats, n_hidden, n_classes, n_layers, activation, dropout=0.,
                 pair_norm=False, add_self_loops=False, drop_edge=0):
        super(GCN, self).__init__()
        if dropout:
            self.dropout = nn.Dropout(p=dropout)
        else:
            self.dropout = 0.
        self.layers = nn.ModuleList()
        # input layer
        self.layer1 = GCNBlock(in_feats, n_hidden, activation, pair_norm=pair_norm,add_self_loops=add_self_loops, drop_edge=drop_edge)
        self.layers = nn.ModuleList()
        # hidden layers
        for i in range(n_layers - 1):
```

```
            self.layers.append(GCNBlock(n_hidden, n_hidden, activation, pair_norm=pair_norm,add_self_loops=add_self_loops, drop_edge=drop_edge))
        # output layer
        self.layers.append(GCNBlock(n_hidden, n_classes, activation=None, pair_norm=pair_norm,add_self_loops=add_self_loops,drop_edge=drop_edge))

    def forward(self, g, features):
        h = self.layer1(g, features)
        for idx, layer in enumerate(self.layers):
            if idx > 0 and self.dropout:
                h = self.dropout(h)
            h = layer(g, h)
        if self.dropout:
            h = self.dropout(h)
        return h
```

五、实验结果

1.Node Classification

节点分类任务中，cora数据集和citeseer数据集是multi-class，用正确率作为评价指标，ppi数据集为multi-label，用F1指标作为评价指标。并且对于ppi数据集，采用20张图片作训练，2张图片作验证，2张图片作测试。

add_self_loop

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
cora	True	1	0.0	False	relu	**0.836**
cora	False	1	0.0	False	relu	0.811
cora	True	3	0.0	False	relu	0.817
cora	False	3	0.0	False	relu	0.798
cora	True	5	0.0	False	relu	0.807
cora	False	5	0.0	False	relu	0.791

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
citeseer	True	1	0.0	False	relu	**0.703**
citeseer	False	1	0.0	False	relu	0.660
citeseer	True	3	0.0	False	relu	0.692
citeseer	False	3	0.0	False	relu	0.674
citeseer	True	5	0.0	False	relu	0.676
citeseer	False	5	0.0	False	relu	0.624

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_f1
ppi	True	1	0.0	False	relu	0.895
ppi	False	1	0.0	False	relu	0.894
ppi	True	3	0.0	False	relu	0.923
ppi	False	3	0.0	False	relu	0.921
ppi	True	5	0.0	False	relu	**0.962**
ppi	False	5	0.0	False	relu	0.958

对于三个数据集，添加self_loop后的效果都要更好。

网络层数

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
cora	True	1	0.0	False	relu	**0.836**
cora	True	2	0.0	False	relu	0.823
cora	True	3	0.0	False	relu	0.817
cora	True	5	0.0	False	relu	0.807
cora	True	10	0.0	False	relu	0.707

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
citeseer	True	1	0.0	False	relu	**0.703**
citeseer	True	2	0.0	False	relu	0.696
citeseer	True	3	0.0	False	relu	0.688
citeseer	True	5	0.0	False	relu	0.676
citeseer	True	10	0.0	False	relu	0.551

对cora和citeseer数据集来说，浅层的网络效果要略好。

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_f1
ppi	True	1	0.0	False	relu	0.895
ppi	True	2	0.0	False	relu	0.870
ppi	True	3	0.0	False	relu	0.923
ppi	True	5	0.0	False	relu	**0.962**
ppi	True	10	0.0	False	relu	0.944

对ppi数据集来说，5层的效果要显著更好。

drop edge

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
cora	True	1	0.0	False	relu	**0.836**
cora	True	1	0.1	False	relu	0.728
cora	True	1	0.2	False	relu	0.666
cora	True	1	0.5	False	relu	0.619
cora	True	3	0.0	False	relu	0.817
cora	True	3	0.1	False	relu	0.648
cora	True	3	0.2	False	relu	0.640
cora	True	3	0.5	False	relu	0.575

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
citeseer	True	1	0.0	False	relu	**0.703**
citeseer	True	1	0.1	False	relu	0.652
citeseer	True	1	0.2	False	relu	0.605
citeseer	True	1	0.5	False	relu	0.585
citeseer	True	3	0.0	False	relu	0.688
citeseer	True	3	0.1	False	relu	0.604
citeseer	True	3	0.2	False	relu	0.581
citeseer	True	3	0.5	False	relu	0.562

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_f1
ppi	True	1	0.0	False	relu	0.895
ppi	True	1	0.1	False	relu	0.844
ppi	True	1	0.2	False	relu	0.808
ppi	True	1	0.5	False	relu	0.731
ppi	True	3	0.0	False	relu	**0.923**
ppi	True	3	0.1	False	relu	0.866
ppi	True	3	0.2	False	relu	0.827
ppi	True	3	0.5	False	relu	0.758

drop_edge在三个数据集上都没有明显的效果。

PairNorm

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
cora	True	1	0.0	False	relu	**0.836**
cora	True	1	0.0	True	relu	0.779
cora	True	3	0.0	False	relu	0.817
cora	True	3	0.0	True	relu	0.761
cora	True	5	0.0	False	relu	0.807
cora	True	5	0.0	True	relu	0.772

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
citeseer	True	1	0.0	False	relu	**0.703**
citeseer	True	1	0.0	True	relu	0.691
citeseer	True	3	0.0	False	relu	0.688
citeseer	True	3	0.0	True	relu	0.682
citeseer	True	5	0.0	False	relu	0.676
citeseer	True	5	0.0	True	relu	0.678

在cora和citeseer数据集上，pair_norm没有提升模型效果。

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_f1
ppi	True	1	0.0	False	relu	0.895
ppi	True	1	0.0	True	relu	0.732
ppi	True	3	0.0	False	relu	0.923
ppi	True	3	0.0	True	relu	0.919
ppi	True	5	0.0	False	relu	0.962
ppi	True	5	0.0	True	relu	**0.966**

在ppi数据集上，pair_norm使得模型效果有了一定的提升。

激活函数

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
cora	True	1	0.0	False	relu	**0.836**
cora	True	1	0.0	False	tanh	0.823
cora	True	1	0.0	False	sigmoid	0.751
cora	True	3	0.0	False	relu	0.817
cora	True	3	0.0	False	tanh	0.815
cora	True	3	0.0	False	sigmoid	0.319
cora	True	5	0.0	False	relu	0.807
cora	True	5	0.0	False	tanh	0.798
cora	True	5	0.0	False	sigmoid	0.319

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
citeseer	True	1	0.0	False	relu	**0.703**
citeseer	True	1	0.0	False	tanh	0.690
citeseer	True	1	0.0	False	sigmoid	0.677
citeseer	True	3	0.0	False	relu	0.688
citeseer	True	3	0.0	False	tanh	0.695
citeseer	True	3	0.0	False	sigmoid	0.275
citeseer	True	5	0.0	False	relu	0.676
citeseer	True	5	0.0	False	tanh	0.663
citeseer	True	5	0.0	False	sigmoid	0.253

在cora和citeseer数据集上，relu≈tanh＞sigmoid。

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
ppi	True	1	0.0	False	relu	0.895
ppi	True	1	0.0	False	tanh	0.900
ppi	True	1	0.0	False	sigmoid	0.890
ppi	True	3	0.0	False	relu	0.923
ppi	True	3	0.0	False	tanh	0.940
ppi	True	3	0.0	False	sigmoid	0.750
ppi	True	5	0.0	False	relu	0.958
ppi	True	5	0.0	False	tanh	**0.984**
ppi	True	5	0.0	False	sigmoid	0.424

但在ppi数据集上，tanh>relu>sigmoid。

best result

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_acc
cora	True	1	0.0	False	relu	**0.836**
citeseer	True	1	0.0	False	relu	**0.703**

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_f1
ppi	True	5	0.0	False	tanh	**0.984**

2.Link Prediction

链路预测任务中，采用AUC作为评价指标。由于ppi数据集训练时间过长，仅采用cora和citeseer对比不同技术的效果，并且对于ppi数据集取5张图片来测试效果。

add_self_loop

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
cora	True	1	0.0	False	relu	0.939
cora	False	1	0.0	False	relu	0.945
cora	True	3	0.0	False	relu	0.943
cora	False	3	0.0	False	relu	**0.960**
cora	True	5	0.0	False	relu	0.937
cora	False	5	0.0	False	relu	0.953

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
citeseer	True	1	0.0	False	relu	0.962
citeseer	False	1	0.0	False	relu	0.920
citeseer	True	3	0.0	False	relu	0.958
citeseer	False	3	0.0	False	relu	0.924
citeseer	True	5	0.0	False	relu	**0.964**
citeseer	False	5	0.0	False	relu	0.962

对于两个数据集，添加self_loop后的效果都要更好。

网络层数

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
cora	True	1	0.0	False	relu	0.945
cora	True	2	0.0	False	relu	0.945
cora	True	3	0.0	False	relu	**0.960**
cora	True	5	0.0	False	relu	0.953
cora	True	10	0.0	False	relu	0.896

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
citeseer	True	1	0.0	False	relu	0.962
citeseer	True	2	0.0	False	relu	0.957
citeseer	True	3	0.0	False	relu	0.958
citeseer	True	5	0.0	False	relu	**0.964**
citeseer	True	10	0.0	False	relu	0.963

对于两个数据集，中浅层的网络效果不错。

drop edge

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
cora	True	1	0.0	False	relu	0.945
cora	True	1	0.1	False	relu	0.892
cora	True	1	0.2	False	relu	0.900
cora	True	1	0.5	False	relu	0.896
cora	True	3	0.0	False	relu	**0.960**
cora	True	3	0.1	False	relu	0.909
cora	True	3	0.2	False	relu	0.896
cora	True	3	0.5	False	relu	0.894

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
citeseer	True	1	0.0	False	relu	**0.962**
citeseer	True	1	0.1	False	relu	0.950
citeseer	True	1	0.2	False	relu	0.932
citeseer	True	1	0.5	False	relu	0.917
citeseer	True	3	0.0	False	relu	0.958
citeseer	True	3	0.1	False	relu	0.919
citeseer	True	3	0.2	False	relu	0.899
citeseer	True	3	0.5	False	relu	0.901

对于两个数据集，drop edge都没有起到很好的效果。

PairNorm

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
cora	True	1	0.0	False	relu	0.945
cora	True	1	0.0	True	relu	0.953
cora	True	3	0.0	False	relu	**0.960**
cora	True	3	0.0	True	relu	0.954
cora	True	5	0.0	False	relu	0.953
cora	True	5	0.0	True	relu	0.946

对于cora数据集，pair_norm的效果一般。

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
citeseer	True	1	0.0	False	relu	0.962
citeseer	True	1	0.0	True	relu	0.966
citeseer	True	3	0.0	False	relu	0.958
citeseer	True	3	0.0	True	relu	**0.971**
citeseer	True	5	0.0	False	relu	0.964
citeseer	True	5	0.0	True	relu	0.958

而对于citeseer数据集，pair_norm的添加使得模型有了明显的提升。

激活函数

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
cora	True	1	0.0	False	relu	0.945
cora	True	1	0.0	False	tanh	0.952
cora	True	1	0.0	False	sigmoid	0.755
cora	True	3	0.0	False	relu	0.960
cora	True	3	0.0	False	tanh	**0.963**
cora	True	3	0.0	False	sigmoid	0.603
cora	True	5	0.0	False	relu	0.953
cora	True	5	0.0	False	tanh	0.951
cora	True	5	0.0	False	sigmoid	0.584

在cora数据集上，relu≈tanh＞sigmoid。

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
citeseer	True	1	0.0	True	relu	0.966
citeseer	True	1	0.0	True	tanh	**0.990**
citeseer	True	1	0.0	True	sigmoid	0.988
citeseer	True	3	0.0	True	relu	0.971
citeseer	True	3	0.0	True	tanh	0.981
citeseer	True	3	0.0	True	sigmoid	0.980
citeseer	True	5	0.0	False	relu	0.964
citeseer	True	5	0.0	False	tanh	0.969
citeseer	True	5	0.0	False	sigmoid	0.973

在citeseer数据集上，tanh>relu>sigmoid。

best result

data_name	add_self_loop	n_layers	drop_edge	pair_norm	activations	test_auc
cora	True	3	0.0	False	tanh	**0.963**
citeseer	True	1	0.0	True	tanh	**0.990**
ppi	True	3	0.0	False	relu	**0.871**

六、实验总结

本次实验了解的图神经网络的原理，以及学会了使用DGL，并对比了不同激活函数，self loop, DropEdge, PairNorm等技术对图卷积神经网络性能的影响。