图论project报告

经22-计28 李昊伦

2023-06-02

1 半监督的图节点分类问题的建模过程

对于半监督的图节点分类的问题, 仅有一小部分的节点存在有效的标签. 因此我们可以将整个图结构直接用神经网络模型的方式编码, 然后在这些存在标签的节点上进行训练.

对于图的数学表达, 我们可以选用邻接矩阵的方式. 对于每个节点的标签, 实际上是每个节点都有一个与之对应的特征向量. 我们就可以用图卷积网络(GCN)对图进行卷积操作, 使得节点的特征向量同时包含了自身和邻居节点的特征信息. 由于是半监督的问题, 我们还需要对节点的标签进行传播, 而多层的GCN就可以实现这一点. 最后我们可以定义好激活函数与损失函数, 用以度量预测标签与真实标签的差距, 就可以训练模型了.

2 图卷积网络算法的具体过程

在论文中, 作者给出了图卷积网络的数学推导, 最终选择了两层线性GCN层组成的结构. 以下是最终推出的式子:

$$Z = f(X, A) = softmax(\hat{A}\, ReLU(\hat{A} X W^{(0)}) W^{(1)})$$

其中$X \in \mathbb{R}^{N \times C}$表示每个节点的$C$维特征向量组成的矩阵, A是邻接矩阵, $W^{(0)} \in \mathbb{R}^{C \times H}$是输入层到隐藏层的参数矩阵, $W^{(1)} \in \mathbb{R}^{H \times F}$是隐藏层到输出层的参数矩阵, H为隐藏层的维数. 定义$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, 其中$\tilde{A} = A + I_N$, I_N是N维单位矩阵, $\tilde{D}$是$\tilde{A}$的度矩阵.

在GCN中, 每个节点的特征向量都包含了自身的特征, 每个节点都与自身相连, 因此在邻接矩阵中加上单位矩阵, 可以使得每个节点的特征向

量都包含了自身的特征. 而与度矩阵计算, 可以使得每个节点的特征向量都包含了邻居节点的特征, 从而实现了卷积操作. 这里的第一层GCN选用了ReLU作为激活函数, 第二层GCN选用了softmax作为激活函数, 从而实现了标签的传播.

在实际训练中, 将使用梯度下降法作为对参数矩阵W的训练方法. 而损失函数则选用了交叉熵损失函数.

我复现论文中的实验, 使用了python3.7以及tensorflow 1.15.4版本. 上述的激活函数与损失函数, tensorflow都有现成的实现, 直接调用即可.

代码思路主要为通过继承tf.keras.Model和tf.keras.layers.Layer类, 重写其中的call方法, 从而实现自定义的GCN. 而论文中对邻接矩阵的归一化处理, 我将其放在了数据读取的部分, 随其他相关代码一起封装在了DataSets类中. 具体代码见附件.

3 实验结果

我使用了Cora和Citeseer两个数据集进行实验, 超参数均与论文保持一致(即隐藏层维数为16, 学习率为0.01, dropout为0.5, L2正则化系数为5×10^{-4}). 而标签率也与论文基本保持一致, 即Cora数据集的标签率为4%, Citeseer数据集的标签率为5%(清洗后). 以下实验数据是随机测试20次后的平均结果:

表 1: 实验结果

损失函数	标签率	准确率	偏差范围
Cora	4%	0.76	± 0.02
Citeseer	5%	0.56	± 0.01

特别感谢Github Copilot以及ChatGPT的帮助, 让我能快速上手tensorflow.