

图卷积存在问题及解决方案

袁铭潮

2020.01.16

1. 背景

自 2014 年开始, 研究者们开始尝试将神经网络一般化并应用到任意图结构中, 其中一些工作在某些领域已经达到了非常好的效果, 而这些领域之前主要都是由核方法、图正则化等方法来解决的。

2017 年, Kipf 等在论文中提出了最初版本的图神经网络 GCN, 这里所说的 GCN 是指个人称为基于频域的图卷积, 是从谱图卷积的框架出发的, 既能明显加快训练时间又能得到更高的预测精度, 在一些基准的图数据集上, 达到了当时最好的效果, 并在随后几年的时间里得到了更大的发展。

但是 GCN 目前还是存在很多问题尚未得到完善的解决, 本文就这些问题以及阅读过的相关论文进行简单的论述。

GCN 目前的问题一共分为三个方面, 首先是可解释性问题, 这是深度学习的通病, 暂且掠过不谈。其次是有向图问题, 由于谱图卷积的理论限制, 这是亟待解决的问题。最后是 GCN 的训练问题, 过渡平滑, 网络深度等等, 也是目前使用 GCN 的研究者面临最多的一个问题。

2. GCN 和有向图

基于频域的图卷积所使用的方法是先定义了图上的傅里叶变换, 使用拉普拉斯矩阵代替拉普拉斯算子, 进而定义出图卷积。这也就天然的导致了 GCN 不支持有向图的问题。

回顾一下拉普拉斯矩阵的定义: 有 n 个顶点的无向图 $G = (V, E)$, 根据 G , 得到度矩阵 D 和邻接矩阵 A , $L = D - A$ 。

这里使用的是最经典的拉普拉斯矩阵, 与广泛应用的对称归一化拉普拉斯矩阵一样, 都是基于度矩阵

D 和邻接矩阵 A 计算得到的。

而 GCN 需要对拉普拉斯矩阵进行谱分解, 也就意味着, 如果对有向图按照上述方式计算拉普拉斯矩阵, 得到的矩阵将是不对称的, 也就意味着不能进行谱分解, GCN 也就不能进行计算了。

面对这个问题, 目前的思路有两种:

1、要想保持理论上的通用性, 就需要重新定义图的邻接关系, 保持有向图的拉普拉斯矩阵的对称性。例如 MotifNet[1]利用 Graph Motifs 定义图的邻接关系, 可以看作是 ChebNet 的广义化。

2、逐顶点的运算方式计算有向图。这严格意义上讲已经是另一种 GCN 了。目前使用这种方式的是图注意力网络 [2]。

3. 图注意力网络 GAT

GAT 本质上可以分为两种运算方式: Global graph attention 和 Mask graph attention。

其中 Global garph attention 是每一个顶点都对图上任意顶点进行 attention 运算, 完全不依赖于图的结构, 但是丢掉了图结构的特征, 效果先不说, 运算代价是十分高昂的。

Mask graph attention 中, 注意力机制的运算只在邻居节点上进行, 也就是 GAT 论文中所使用的方法。

和所有的注意力机制一样, GAT 的计算也分为两步。

计算注意力系数并进行归一化, 如图 1 所示。对于顶点 i , 逐个计算它的邻居节点 ($j \in \mathcal{N}_i$) 和它本身的相似系数: $e_{i,j} = a([Wh_i || Wh_j]), j \in \mathcal{N}_i$ 。

共享参数 W 的线性映射对于顶点的特征进行了增维, 这是一种常见的特征增强方法。 $a(\cdot)$ 把拼接后的高维特征映射到一个实数上。

有了相关系数后需要做的就是归一化。这里使用的是 softmax: $\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(e_{ij}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(e_{ik}))}$

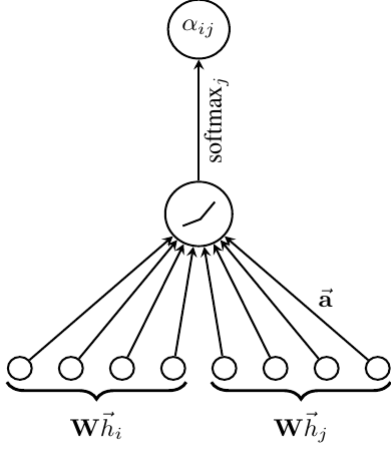


图 1. 注意力系数计算并归一化

根据计算好的注意力系数，把特征加权求和，如图 2 所示: $h'_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W h_j)$, h'_i 就是 GAT 输出的对于每个顶点 i 的新特征。

论文中增加了一步，使用 multi-head attention 对注意力机制进行强化：

$$h'_i(K) = \parallel_{k=1}^K \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W h_j)。$$

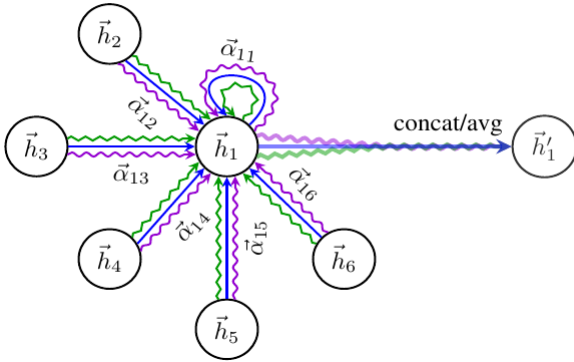


图 2. 特征加权求和

本质上而言：GCN 与 GAT 都是将邻居顶点的特征聚合到中心顶点上，利用图上的局部静止学习新的顶点特征表达。不同的是 GCN 利用了拉普拉斯矩阵，GAT 利用注意力系数。一定程度上而言，GAT 会更强，因为顶点特征之间的相关性被更好地融入到模型中。

GAT 能够支持有向图的根本原因是其运算方式是

逐顶点的运算，每一次运算都需要循环遍历图上的所有顶点来完成。逐顶点意味着摆脱了拉普拉斯矩阵的束缚，使得有向图问题迎刃而解。

4. GCN 训练中的问题

GCN 模型过深存在的问题：GCN 模型的图卷积只是拉普拉斯平滑的一种特殊形式，它混合了节点及其邻近点的特征。平滑操作使同一簇中的结点特征相似，从而大大简化了分类任务，这是 GCN 工作如此出色的关键原因。然而，它也带来了过度平滑的潜在担忧。

在 GCN 的卷积核中，含有拉普拉斯矩阵的 j 次幂，当 j 的取值较大时，GCN 学习到的特征可能会存在过渡平滑现象。

如果一个 GCN 有很多卷积层，输出特征可能会过度平滑，不同簇的结点可能变得不可区分。在只有几个卷积层的小数据集上，特征混合发生得很快。此外，向 GCN 添加更多的层将使其更难训练。

GCN 模型过浅存在的问题：浅层 GCN 模型有其自身的局限性。除了需要许多额外的标签来进行验证外，它还受到卷积核的局部特性的影响。当只给出很少的标签时，一个较浅的 GCN 就不能有效地将标签传播到整个数据图中。

对于上述局限性，目前所提出的解决方案是用随机游走模型协调训练 GCN[3]，因为随机游走可以探索全局图结构，这是对 GCN 模型的补充。

另一种方法是自训练 GCN。具体来说，首先用给定的标签训练一个 GCN，然后通过比较 SoftMax 分数，选出每个类中最有信心的预测，并将它们添加到标签集。然后，继续使用扩展标签集对 GCN 进行训练，使用预训练的 GCN 作为初始化。

为了提高标签的多样性并训练一个更强大的分类器，建议将协同训练和自训练相结合。具体来说，使用随机游走和 GCN 本身发现的最有信心的预测来扩展标签集，然后使用扩展的标签集继续训练 GCN。为了找到更准确的标签添加到标签集，建议添加随机游走和 GCN 都发现的最有信心的预测。

5. 总结

个人的理解，MotifNet 应该算是理论上较为完备的 GCN 对与有向图的解决方案，但是应用中却并不多。直观上看，可能是因为目前已经有算法复杂度比 ChebNet 更低的方法，例如基于小波滤波的图卷积的实现，把复杂度降低了近乎一个数量级，而 MotifNet 的复杂度明显是高于 ChebNet 的。

那么是不是可以结合 Graph Mitifs 与小波滤波的形式，使基于小波滤波的图卷积适用于有向图，从而解决有向图学习的问题。还是

对于 GCN 的过渡平滑问题，是否可以尝试使用残差网络的架构来加深深度？

GCN 是目前极具潜力的模型，个人认为将会是下一个推动深度学习大步发展的网络模型基础之一，但目前尚显青涩，期待 GCN 的进化蜕变。

参考文献

- [1] Monti F , Otness K , Bronstein M M . MotifNet: a motif-based Graph Convolutional Network for directed graphs[J]. 2018. 1
- [2] Veličković, Petar, Cucurull G , Casanova A , et al. Graph Attention Networks[J]. 2017. 1
- [3] Li, Qimai, Han, Zhichao, Wu, Xiao-Ming. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning[J]. 2018. 2