评论 Poli 等人 2019 年的《图神经常微分方程》

潘旻琦

2020 年 4 月 10 日

1. [Poli et al., 2019] 是我读到的又一个结合图神经网络和神经微分方程的工作；因为图和微分方程是应用数学的两大建模工具，建模能力都很强，看来用深度学习的方法将它们组合起来是一个很酷很强大的想法，我想将来可以把这种思路扩展到更多种类的微分方程，而不仅仅是常微分方程，例如：偏微分方程、随机微分方程、时滞微分方程、积分微分方程、微分代数方程等

2. [Poli et al., 2019] 把经典的动力网络理论作为他的一个研究动机，经典的动力网络理论交叉了图论和动力系统两个领域，研究的是动力单元由图结构耦合在一起组成的动力系统，即在各个动力单元上发生的时间过程和将它们联系起来的空间结构之间的相互作用；[Poli et al., 2019] 的 “时空连续图架构” 这个混合系统是该文不同于我之前阅读的 [Zang and Wang, 2019] 的最大亮点，亦即实现了一个同时包括 “连续流” 和 “离散跳” 两个动态特性的动力系统，可以处理动态变化的图拓扑结构；[Zang and Wang, 2019] 的工作和 [Poli et al., 2019] 的工作都报告了一个同样的结论，那就是即使数据不是由连续动力系统生成的，连续的微分方程图卷积网络也是有用的，所以整体来讲用常微分方程改造原来离散层的图卷积网络是绝对的进步

3. 在 [Battaglia et al., 2018] 的理论框架下，[Poli et al., 2019] 的模型可以看作是同时施加了以下两个归纳偏见：

   (a) 一是关于系统状态结构的归纳偏见，当目标问题结构可以被编码为图时，或可以将关于输入实体之间的关系的先验知识描述为图的情况下，就可以假定系统状态结构属于图结构，例如模拟物理过程的时候通过图中的节点和边对物理粒子及其相互作用进行建模；作为对比，传统欧式图像数据可以施加平移不变性这个归纳偏见，导致卷积运算矩阵出现 Toeplitz 循环结构，可以在输入图像中跨地区重复使用相同的卷积参数，在时序数据上也是可以类似施加时间不变性；但是在图上就难以施加类似的不变性，因为其局部拓扑结构可能处处不同

(b) 另一个是关于产生数据的系统的模型的归纳偏见，例如选择将一个系统的动力学模型建模为一个常微分方程；这个思路甚至可以推广到将先验的科学知识合并到神经网络之中，通过利用科学文献的浓缩知识来避免依赖大型的昂贵的数据集

一般地说，为神经网络赋予更好的归纳偏见能提高样本效率和泛化性能，例如可以在贝叶斯模型的先验条件下编码一些归纳偏见，或在神经网络的架构设计中实践一些归纳偏见

4. [Poli et al., 2019] 的实验发现在 Cora，Citeseer 和 Pubmed 上用 Dormand–Prince 龙格库塔解出来的图的节点在嵌入空间内的点的轨迹是发散的，这与两年前 [Li et al., 2018] 研究的是类似的问题，亦即为什么图卷积网络无法进行深度学习，但是他们得出的结论刚好相反：[Li et al., 2018] 当时观察到图的节点在嵌入空间内的点会随着 GCN 层数的增加而收敛，并证明了：

$$\lim_{m\to\infty}(I-\alpha\mathcal{L})^m w = D^{-\frac{1}{2}}[1,1,\ldots,1]c$$

即对于对称拉普拉斯平滑算子 $\mathcal{L}$，图的每个连通子图内的顶点的特征将收敛为与关于顶点度的平方根的一个线性组合常量；[Poli et al., 2019] 的结论则恰恰相反，避免了收敛，因此使用较长积分间隔训练的图卷积微分方程模型的分类性能不会降低，为什么 [Poli et al., 2019] 的模型会发散呢？原因可能隐藏在下面这些设计细节中：

(a) 首先，[Poli et al., 2019] 一个显著不同是保留了 Cora 图的有向性，其他的工作往往对邻接矩阵 A 做变换 $A \leftarrow A + A^T$，强制把有方向的 Cora 图变成了无向对称图，但是 [Poli et al., 2019] 没有这样做；作者的邻接矩阵 A 是非对称矩阵，A 的列代表边的出发节点、行代表边的目的地节点，拉普拉斯矩阵 L 和标准化拉普拉斯矩阵 $\mathcal{L}$ 的定义也针对有向图做了调整，从源代码可以看出来作者使用的是入度矩阵 D_{in}：

$$\mathcal{L} \equiv D_{\text{in}}^{-\frac{1}{2}} L D_{\text{in}}^{-\frac{1}{2}} = D_{\text{in}}^{-\frac{1}{2}}(D_{\text{in}} - A)D_{\text{in}}^{-\frac{1}{2}} = I - D_{\text{in}}^{-\frac{1}{2}} A D_{\text{in}}^{-\frac{1}{2}}$$

注意此时的拉普拉斯矩阵 L 也不对称了，不一定可以进行正交特征值分解，之前针对无向图的理论可能要做出调整；但是 [Poli et al., 2019] 似乎并未对此做出解释，而是直接套用了前人针对无向图发明的图卷积方法

(b) 设图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$，输入的每个节点的特征向量是 1433 维的，输出的每个节点的类别是 7 个中的某一个，该实验所用的神经网络的整体定义如下：

$$\text{图卷积}_{7\leftarrow 64} \circ \text{动力系统}_f \circ \text{ReLU} \circ \text{图卷积}_{64\leftarrow 1433} \circ \text{辍学}_{40\%}$$

(c) 图卷积$_{\text{出}\leftarrow\text{入}}$ 函数的定义如下

$$\begin{aligned}\text{图卷积}_{\text{出}\leftarrow\text{入}}(X) &\equiv \big[\tilde{D}_{\text{in}}^{-\frac{1}{2}}\tilde{A}\tilde{D}_{\text{in}}^{-\frac{1}{2}}\big]_{|\mathcal{V}|\times|\mathcal{V}|}X_{|\mathcal{V}|\times\text{入}}W_{\text{入}\times\text{出}} + b_{|\mathcal{V}|\times\text{出}}\\ \tilde{D}_{\text{in}} &\equiv I + D_{\text{in}}\\ \tilde{A} &\equiv I + A\end{aligned}$$

(d) “动力系统$_f$” 是一个从相空间到相空间的映射，即用 Dormand–Prince 龙格库塔求解下面这个自治的动力系统，然后通过得到的演化函数 $z(t)$ 把系统的初始状态 $z(0)$ 映射到状态 $z(1)$：

$$\begin{aligned}\text{动力系统}_f \equiv &\big[z(0)\mapsto z(1)\big]\\ &\text{subject to } \frac{dz(t)}{dt} = f(z(t)),\quad t\in[0,1]\end{aligned}$$

为什么规定 $t\in[0,1]$，而不是可变上下界呢？作者认为任何其他上下界都可以视为 $[0,1]$ 的缩放版本，显然是寄希望于神经网络参数在训练期间自动适应这种缩放

(e) 动力系统的演化规则 f 的定义如下：

$$f \equiv \text{图卷积}_{64\leftarrow 64}\circ\text{辍学}_{90\%}\circ\text{Softmax}\circ\text{图卷积}_{64\leftarrow 64}\circ\text{辍学}_{90\%}$$

为什么辍学率高达 90% 呢？这是因为 Cora 数据集本来就不大，训练集样本少，很容易产生过拟合，需要很强的正则化，而高辍学率提高了转导节点分类任务的性能；由于动力系统有着 “很深” 的稠密的层数，因此可以利用此性质

5. 要小心常微分方程变得僵硬，僵硬的常微分方程本质上就是说它在一个区域中表现得太不稳定了；例如在使用 Dormand–Prince 龙格库塔自适应步进求解器时，步长会变得非常接近零，以至于求解器无法继续取得任何进展，导致数值下溢；[Poli et al., 2019] 采取了如下数个手段来降低微分方程的僵硬度：

   (a) 首先是拿双曲正切或 Softmax 而不是 ReLU 来作为演化规则 f 内的激活函数，因为据说平滑的激活函数会降低微分方程的僵硬度

   (b) 其次是增大隐藏动力系统相空间的维度，例如增大到 64 维

   (c) 最后是避免过高的辍学率，过高的辍学率也会增加微分方程的僵硬度

   降低僵硬度有助于避免数值不稳定；同时，在使用 Dormand–Prince 龙格库塔自适应步进求解器时，可以降低为了确保正确性达到预先指定的容忍水平而对演化规则函数 f 的调用次数

参考文献

[Battaglia et al., 2018] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

[Li et al., 2018] Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[Poli et al., 2019] Poli, M., Massaroli, S., Park, J., Yamashita, A., Asama, H., and Park, J. (2019). Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*.

[Zang and Wang, 2019] Zang, C. and Wang, F. (2019). Neural dynamics on complex networks. *arXiv preprint arXiv:1908.06491*.