评论 Zang 和 Wang 2019 年的《复杂网络上的神经动力学》

潘旻琦

2020 年 4 月 4 日

1. 图卷积网络无法进行深度学习，四年前 [Kipf and Welling, 2016] 发现使用 2 层或 3 层的图卷积网络可获得最佳结果；对于深度超过 7 层的模型，不使用残差连接的训练会变得困难，因为对于具有 K 层的模型，每增加一个层每个节点的有效上下文大小会随着其 K 阶节点邻居的大小而增加；此外，随着参数数量随模型深度的增加，过拟合可能成为问题；这也许就是 AAAI 今年举办首届图深度学习国际研讨会的原因吧！该研讨会的最佳论文 [Zang and Wang, 2019] 利用了两年前 [Chen et al., 2018] 的可高效优化的常微分方程求解器把前人的图神经网络的整数层数扩展到了无限的实数深度

2. [Zang and Wang, 2019] 的节点分类实验装置与很多其他图神经网络的实验装置都非常类似，他们都使用了 CORA、CiteSeer、PubMed 三个数据集，这三个数据集都是引文网络，其中节点是论文、边是引用；以 CORA 数据集为例，它对 2708 篇论文的每一篇都标记了是否存在 1433 个单词中的某一些，这个通过布尔值描述的 $\{0,1\}^{1433}$ 空间内的词向量就是每个节点的特征向量，[Zang and Wang, 2019] 对该特征向量做了规一化处理，因此特征向量可以看作是来自一般的 $\mathbb{R}^{1433}$ 空间；由于一张完整的 CORA 图含 2708 个节点，于是定义整张图的特征矩阵 X 定义为属于 $\mathbb{R}^{2708\times1433}$ 矩阵空间的一个矩阵；虽然原 CORA 数据集是不对称的有向图，但是作者对其邻接矩阵 A 做了下面这个变换，强制变成了无向对称图：

$$A \leftarrow A + A^T$$

CORA 数据集本来有 5429 条有向边，这样导致变成了 10556 条无向边；该数据集对每个节点都标记了 7 个类别中的某一个，这就是分类任务的训练损失的评判标准；文节点分类任务的输入是 “论文里的词” 这个特征以及 “论文互相引用” 这个图的结构，可见这里假设了互相引用的论文可能共享相同的标签；但是 [Kipf and Welling, 2016] 指出了此假设可能会限制方法的建模能力，因为图的边不一定需要编码节点的相似性，而是可能包含附加信息

3. [Zang and Wang, 2019] 在超参数 α 的调控下定义了一个 “扩散算子”Φ：

$$\Phi(\alpha) \equiv \big[\alpha I + (1-\alpha)D\big]^{-\frac{1}{2}}\big[\alpha I + (1-\alpha)A\big]\big[\alpha I + (1-\alpha)D\big]^{-\frac{1}{2}}$$

其中 D 是度数矩阵，这里的 $\big[\alpha I + (1-\alpha)D\big]^{-\frac{1}{2}}$ 肯定是个伪逆，因为如果 $\alpha = 0$，图里那些孤立的节点在 D 矩阵中是 0，这里的处理是规定它们在 $D^{-\frac{1}{2}}$ 中也是 0，避免了分母是零的问题；扩散算子 Φ 直接用于定义节点分类实验的如下的 256 维隐藏层的神经动力学模型：

$$\begin{aligned} X_h(0) &= \tanh\big[(\mathbb{R}^{2708\times 256} \leftarrow \mathbb{R}^{2708\times 1433})(X(0))\big] \\ \frac{dX_h(t)}{dt} &= \mathrm{ReLU}(\Phi X_h(t)) \\ X(T) &= \mathrm{Softmax}[(\mathbb{R}^{2708\times 7} \leftarrow \mathbb{R}^{2708\times 256})(X_h(T))] \end{aligned}$$

又用于定义如下的图卷积函数作为对照组实验的图卷积网络之用：

$$\text{图卷积}_{\text{出}\leftarrow\text{入}}(X) \equiv \Phi_{2708\times 2708}\big[(\mathbb{R}^{2708\times \text{出}} \leftarrow \mathbb{R}^{2708\times \text{入}})(X)\big]_{2708\times \text{出}}$$

其中 “入” 和 “出” 分别指定了图卷积函数之前和之后的节点特征的维度

4. 观察 [Zang and Wang, 2019] 的这个用于定义节点分类实验的神经动力公式：

$$\frac{dX_h(t)}{dt} = \mathrm{ReLU}(\Phi X_h(t))$$

可以发现与 [Kipf and Welling, 2016] 的这个逐层传播公式有异曲同工之处：

$$H^{(l+1)} = \sigma\big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\big)$$

因为后者的

$$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}} = \big[I + D\big]^{-\frac{1}{2}}\big[I + A\big]\big[I + D\big]^{-\frac{1}{2}}$$

刚好等于前者的

$$\begin{aligned} \Phi(\frac{1}{2}) &= \big[\frac{1}{2}I + \frac{1}{2}D\big]^{-\frac{1}{2}}\big[\frac{1}{2}I + \frac{1}{2}A\big]\big[\frac{1}{2}I + \frac{1}{2}D\big]^{-\frac{1}{2}} \\ &= \left(\frac{1}{2}\right)^{-\frac{1}{2}}\left(\frac{1}{2}\right)\left(\frac{1}{2}\right)^{-\frac{1}{2}}\big[I + D\big]^{-\frac{1}{2}}\big[I + A\big]\big[I + D\big]^{-\frac{1}{2}} \\ &= \big[I + D\big]^{-\frac{1}{2}}\big[I + A\big]\big[I + D\big]^{-\frac{1}{2}} \end{aligned}$$

可见 $\Phi(\frac{1}{2})$ 其实就是 [Kipf and Welling, 2016] 论文里的 $\hat{A}$，其在 §7.2 里也提到了它的 $\hat{A}$ 其实假定了自连接边与相邻节点边的同等重要性，暗示此处可以进行扩展，而 [Zang and Wang, 2019] 的 $\Phi(\alpha)$ 正是实践了这个扩展

5. 我自己举了一个例子来考察扩散算子 $\Phi(\alpha)$ 的内容；设

$$A=\begin{bmatrix}0&1&0&0&1&0\\1&0&1&0&1&0\\0&1&0&1&0&0\\0&0&1&0&1&1\\1&1&0&1&0&0\\0&0&0&1&0&0\end{bmatrix},D=\operatorname{diag}\begin{bmatrix}2\\3\\2\\3\\3\\1\end{bmatrix}$$

记 $\beta=1-\alpha$，那么

$$\begin{aligned}\Phi(\alpha)&=\operatorname{diag}\begin{bmatrix}\frac{1}{\sqrt{\alpha+2\beta}}\\\frac{1}{\sqrt{\alpha+3\beta}}\\\frac{1}{\sqrt{\alpha+2\beta}}\\\frac{1}{\sqrt{\alpha+3\beta}}\\\frac{1}{\sqrt{\alpha+3\beta}}\\\frac{1}{\sqrt{\alpha+\beta}}\end{bmatrix}\begin{bmatrix}\alpha&\beta&0&0&\beta&0\\\beta&\alpha&\beta&0&\beta&0\\0&\beta&\alpha&\beta&0&0\\0&0&\beta&\alpha&\beta&\beta\\\beta&\beta&0&\beta&\alpha&0\\0&0&0&\beta&0&\alpha\end{bmatrix}\operatorname{diag}\begin{bmatrix}\frac{1}{\sqrt{\alpha+2\beta}}\\\frac{1}{\sqrt{\alpha+3\beta}}\\\frac{1}{\sqrt{\alpha+2\beta}}\\\frac{1}{\sqrt{\alpha+3\beta}}\\\frac{1}{\sqrt{\alpha+3\beta}}\\\frac{1}{\sqrt{\alpha+\beta}}\end{bmatrix}\\&=\begin{bmatrix}\frac{\alpha}{\alpha+2\beta}&\frac{\beta}{\sqrt{\alpha+2\beta}\sqrt{\alpha+3\beta}}&0&0&\frac{\beta}{\sqrt{\alpha+2\beta}\sqrt{\alpha+3\beta}}&0\\\frac{\beta}{\sqrt{\alpha+2\beta}\sqrt{\alpha+3\beta}}&\cdots&\cdots&\cdots&\cdots&\cdots\\0&\cdots&\cdots&\cdots&\cdots&\cdots\\0&\cdots&\cdots&\cdots&\cdots&\cdots\\\frac{\beta}{\sqrt{\alpha+2\beta}\sqrt{\alpha+3\beta}}&\cdots&\cdots&\cdots&\cdots&\cdots\\0&\cdots&\cdots&\cdots&\cdots&\cdots\end{bmatrix}\end{aligned}$$

可见 Φ 右乘特征矩阵 X 上就得到了论文中写的 “平均邻里意见” 动力模型：

$$\begin{aligned}\frac{d\overrightarrow{x_i(t)}}{dt}=&\frac{\alpha}{(1-\alpha)d_i+\alpha}\overrightarrow{x_i(t)}+\\&\sum_j^n A_{i,j}\frac{1-\alpha}{\sqrt{(1-\alpha)d_i+\alpha}\sqrt{(1-\alpha)d_j+\alpha}}\overrightarrow{x_j(t)}\end{aligned}$$

我又观察到：

$$\begin{aligned}\Phi(0\%)&=D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\\\Phi(100\%)&=I^{-\frac{1}{2}}II^{-\frac{1}{2}}=I\end{aligned}$$

因此超参数 α 的意思大致是 “对邻居的无视程度”，那么上面推导时用到的符号 β 就意味着 “对邻居的尊重诚度”；$\alpha=100\%$ 意味着完全无视节点邻居的集体意见，此时神经动力学模型退化成了类似指数函数的定义 $\dot{x}(t)=\mathrm{ReLU}(x(t))$，如果

去掉 ReLU 它的解就是 $x(t) = e^t x_0$，而图卷积函数则退化成一个普通的仿射映射；$\alpha = 0\%$ 意味着最大化尊重节点邻居的集体意见，此时的算子 $\Phi(0)$ 有很强的理论联系，因为它与拉普拉斯矩阵 L 和标准化拉普拉斯矩阵 $\mathcal{L}$ 有着如下的关系：

$$\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}} = I - \Phi(0) = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

可以想象扩散算子 Φ 起到用神经网络对图结构进行编码的效果，这背后的设计思想是同时考虑图的结构和节点自身的特征，来让一个神经网络学习到图结构数据的图中节点的有用的特征

6. [Zang and Wang, 2019] 之所以能做连续时间的网络动力学预测这个实验，是得益于选用了两年前 [Chen et al., 2018] 贡献的一个可高效优化的常微分方程求解器，从而可以吸纳任意时间到达的数据；该常微分方程求解器之所以可以高效优化，是利用了伴随敏感度分析法；伴随敏感度分析法在系统控制参数很多的时候提供了一种廉价的计算系统的解的一个泛函关于控制参数的梯度的方法；我对常微分方程初值问题伴随敏感度分析法的总结如下，其中我对矩阵转置的处理比 [Chen et al., 2018] 的附录 B 更为严谨

   (a) 设 $\theta \in \mathbb{R}^M, t \in [t_0, t_1], z(t) : \mathbb{R} \to \mathbb{R}^N$，有如下关于初值问题的优化问题：

   $$\begin{aligned} &\arg\min_{\theta} \ L(z(t_1)) \\ &\text{subject to } \frac{dz(t)}{dt} = f(z(t), t, \theta) \\ &z(t_0) = z_0 \end{aligned}$$

   (b) 我们想通过计算 $\frac{\partial L}{\partial \theta}$ 来求解这个优化问题：

   $$\frac{\partial L}{\partial \theta} = \left[\frac{\partial L}{\partial \theta_1} \quad \cdots \quad \frac{\partial L}{\partial \theta_M}\right]_{1\times M} = \left[\frac{\partial L}{\partial z(t)}\right]_{1\times N} \left[\frac{\partial z(t)}{\partial \theta}\right]_{N\times M}$$

   其中 $\frac{\partial z(t)}{\partial \theta}$ 这个 $N \times M$ 的雅可比矩阵的计算非常昂贵，它的每一列都是通过拿约束条件的等式两边对控制参数 $\theta_j (j = 1, \ldots, M)$ 求偏导得到的一个新的

常微分方程的解：

$$\frac{\partial}{\partial\theta_j}\left[\frac{dz(t)}{dt}=f(z(t),t,\theta)\right]$$
$$\Downarrow$$
$$\frac{d}{dt}\frac{\partial z(t)}{\partial\theta_j}=\frac{\partial f(z(t),t,\theta)}{\partial z(t)}\frac{\partial z(t)}{\partial\theta_j}+\frac{\partial f(z(t),t,\theta)}{\partial\theta_j}$$
$$\Downarrow$$
$$\frac{d}{dt}\begin{bmatrix}\frac{\partial z_1}{\partial\theta_j}\\ \frac{\partial z_2}{\partial\theta_j}\\ \vdots\\ \frac{\partial z_N}{\partial\theta_j}\end{bmatrix}=\begin{bmatrix}\frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \cdots & \frac{\partial f_1}{\partial z_N}\\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \cdots & \frac{\partial f_2}{\partial z_N}\\ \vdots & \vdots & \vdots & \vdots\\ \frac{\partial f_N}{\partial z_1} & \frac{\partial f_N}{\partial z_2} & \cdots & \frac{\partial f_N}{\partial z_N}\end{bmatrix}\begin{bmatrix}\frac{\partial z_1}{\partial\theta_j}\\ \frac{\partial z_2}{\partial\theta_j}\\ \vdots\\ \frac{\partial z_N}{\partial\theta_j}\end{bmatrix}+\begin{bmatrix}\frac{\partial f_1}{\partial\theta_j}\\ \frac{\partial f_2}{\partial\theta_j}\\ \vdots\\ \frac{\partial f_N}{\partial\theta_j}\end{bmatrix}$$

(c) 上面展示了直接求 $\frac{\partial L}{\partial\theta}$ 的代价，即需要解多达 M 个常微分方程；当系统具有大量控制参数（M 很大）的时候，我们不如去解原初值问题的伴随问题，因为伴随问题只需要解 1 个常微分方程就能得到 $\frac{\partial L}{\partial\theta}$；为此，我们定义伴随状态 $a(t):\mathbb{R}\to\mathbb{R}^N$ 如下：

$$a(t)\equiv\left[\frac{\partial L}{\partial z(t)}\right]^T$$

那么伴随着原来关于 $z(t)$ 的初值问题存在一个关于 $a(t)$ 的终值问题：

$$\begin{aligned}[a(t)]^T&=\frac{\partial L}{\partial z(t)}\\&=\frac{\partial L}{\partial z(t+\varepsilon)}\frac{\partial z(t+\varepsilon)}{\partial z(t)}\\&=[a(t+\varepsilon)]^T\frac{\partial}{\partial z(t)}\left[z(t)+\int_t^{t+\varepsilon}f(z(t),t,\theta)dt\right]\\&=[a(t+\varepsilon)]^T\frac{\partial}{\partial z(t)}\left[z(t)+\varepsilon f(z(t),t,\theta)+O(\varepsilon^2)\right]\\&=[a(t+\varepsilon)]^T\left[I+\varepsilon\frac{\partial f(z(t),t,\theta)}{\partial z(t)}+O(\varepsilon^2)\right]\end{aligned}$$

把上式代入 $\frac{da(t)}{dt}$ 的定义式：

$$\begin{aligned}\frac{da(t)}{dt} &= \lim_{\varepsilon\to 0^+}\frac{a(t+\varepsilon)-a(t)}{\varepsilon}\\ &= \lim_{\varepsilon\to 0^+}\frac{a(t+\varepsilon)-\left[I+\varepsilon\frac{\partial f(z(t),t,\theta)}{\partial z(t)}+O(\varepsilon^2)\right]^T a(t+\varepsilon)}{\varepsilon}\\ &= \lim_{\varepsilon\to 0^+}\frac{-\varepsilon\left[\frac{\partial f(z(t),t,\theta)}{\partial z(t)}\right]^T a(t+\varepsilon)+O(\varepsilon^2)}{\varepsilon}\\ &= -\left[\frac{\partial f(z(t),t,\theta)}{\partial z(t)}\right]^T a(t)\end{aligned}$$

亦即

$$\frac{da(t)}{dt} = -\left[\frac{\partial f(z(t),t,\theta)}{\partial z(t)}\right]^T a(t)，\text{满足终值 } a(t_1)=\left[\frac{\partial L}{\partial z(t_1)}\right]^T$$

(d) 对原系统进行增广，把 θ 和 t 都加入系统状态，转变为一个自治系统：

$$\frac{d}{dt}\begin{bmatrix}z(t)\\ \theta\\ t\end{bmatrix} = \begin{bmatrix}f(z(t),\theta,t)\\ 0\\ 1\end{bmatrix}$$

伴随状态就变为了：

$$\begin{bmatrix}\frac{\partial L}{\partial z(t)} & \frac{\partial L}{\partial \theta} & \frac{\partial L}{\partial t}\end{bmatrix}^T$$

由于这个新的伴随状态中包含了 $\frac{\partial L}{\partial \theta}$，解关于这个伴随状态的终值问题就可以得到我们想要的梯度；其中，伴随状态满足的常微分方程变为了：

$$\begin{aligned}&\frac{d}{dt}\begin{bmatrix}\frac{\partial L}{\partial z(t)} & \frac{\partial L}{\partial \theta} & \frac{\partial L}{\partial t}\end{bmatrix}\\ &= -\begin{bmatrix}\frac{\partial L}{\partial z(t)} & \frac{\partial L}{\partial \theta} & \frac{\partial L}{\partial t}\end{bmatrix}\begin{bmatrix}\frac{\partial f(z(t),t,\theta)}{\partial z(t)} & \frac{\partial f(z(t),t,\theta)}{\partial \theta} & \frac{\partial f(z(t),t,\theta)}{\partial t}\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\\ &= -\begin{bmatrix}\frac{\partial L}{\partial z(t)}\frac{\partial f(z(t),t,\theta)}{\partial z(t)} & \frac{\partial L}{\partial z(t)}\frac{\partial f(z(t),t,\theta)}{\partial \theta} & \frac{\partial L}{\partial z(t)}\frac{\partial f(z(t),t,\theta)}{\partial t}\end{bmatrix}\end{aligned}$$

7. 用伴随敏感度分析法去优化 [Chen et al., 2018] 的常微分方程求解器不是必须的，也可以直接通过对求解器前向传播时每个函数求值进行反向传播来计算梯度，亦即依照计算图去执行反向自动微分，但是这会导致高昂的内存存储成本，并引入其他数值误差

8. 我尝试生成了 [Zang and Wang, 2019] 的复杂网络实验数据：

(a) 网格图：生成了一个 20×20 的网格图，即每个节点都有恰好八个邻居节点(除了边缘节点之外)；但由于编号的时候把 20×20 的网格映射到了长度为 400 的序列，邻接矩阵视觉上看起来是一个带有一个约 ± 20 个像素宽白条的对角矩阵；例如第 21 号节点的邻居的编号是：$0, 1, 2, 20, 22, 40, 41, 42$

(b) 随机图：生成了一个 Erdős–Rényi 随机图，图共含 400 个节点，并以 0.1 的概率选择了每条可能的边；然后使用 Clauset-Newman-Moore 模块最大化贪心算法在图中查找到了 4 个社区，大小为 $152, 138, 107, 3$，依据社区对节点进行重新排序

(c) 幂律图：生成了一个 Barabási–Albert 随机图，通过不断把具有 5 条边的节点附加到图上渐渐生长出 400 个节点；其间遵循马太效应，即优先生长到现有的度比较高的节点上，生成后同样进行社区发现并重新排序；这个图属于无标度网络，具有幂律度分布

(d) 小世界图：生成了一个 Newman–Watts–Strogatz 随机图，首先在 400 个节点上创建一个环，环中的每个节点都与其最近的 4 个邻居连接，对于每个边 (u, v) 以 0.5 的概率添加随机选择的现有节点 w 的新边 (u, w)，生成后同样进行社区发现并重新排序；这个图的特性是平均路径长度短，聚集度高，大部分的节点彼此并不相连，但绝大部分节点之间经过少数几步就可到达；邻接矩阵视觉上有两条断断续续的对角线，这可能是一开始那个大环，例如对角线附近 3 的邻居有 $1, 2, 4, 5$，对角线附近 4 的邻居有 $2, 3, 5, 6$

(e) 社区图：生成了一个由大小分别为 $133, 133, 100, 34$ 的四个社区组成的社区图，同一社区中的节点以概率 0.25 连接，而不同社区之间的节点以概率 0.01 连接，生成后同样进行社区发现并重新排序；邻接矩阵视觉上可以看见沿着对角线方向的四个显著的分块矩阵

参考文献

[Chen et al., 2018] Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. *CoRR*, abs/1806.07366.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

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