题目

摘要

大多数图神经网络（GNN）遵循消息传递机制。然而，当多次消息传递应用于图时，它将面临过度平滑问题，导致不可区分的节点表示，并阻止模型有效地学习更远节点之间的依赖关系。另一方面，具有不同标签的相邻节点的特征可能被错误地混合，从而导致异质性问题。在这项工作中，我们建议将消息传递到节点的表示中，特定的神经元块针对特定跳内的消息传递。这是通过将中心节点的根树的层次结构与其节点表示中的神经元有序对齐来实现的。我在大量数据集上进行实验，结果表明，模型可以同时在同质和异质数据集上实现最先进的效果，验证了模型的性能。此外，我还针对模型的过度平滑性能进行测试，在不同深度的模型上进行实验，证实了模型可以较好的防止过度平滑问题。

关键词：图神经网络；异质性；消息传递；过度平滑

1 引言

图神经网络（GNN）已经成为学习图形表示的主要方法，例如在社交网络、生物医学信息网络和通信网络中实现相邻节点之间的交互。尽管在许多任务上取得了巨大成功，但传统的消息传递型 GNN 仍然面临两个根本但致命的问题：一是难以推广到相邻节点共享不同特征或标签的异质性，二是在处理具有异质性的真实世界网络时，简单的多层感知机的表现往往超过了许多 GNN。此外，我们还观察到，在堆叠多层时，节点表示变得不可区分，导致性能急剧下降，形成所谓的“过度平滑”问题，限制了 GNN 在利用高阶邻域信息方面的有效性。为了解决这两个问题，我们引入了消息传递的联合收割机阶段，并强调了这一设计的重要性。关键思想在于通过从根树层次中整合归纳偏差，使 GNN 能够以一定的顺序准确地编码邻域信息，避免了跨跃内部的特征混合问题。这一创新设计旨在提高 GNN 对异质性网络的适应性，并有效地克服“过度平滑”问题，从而更好地利用高阶邻域信息。

2 相关工作

为了解决这两个缺点，已经提出了许多方法。它们中的大多数集中在消息传递的聚合阶段。一些设计签名消息以区分属于不同类的邻居，允许 GNN 捕获高频信号; Min 等人 [1] 设计特定的滤波器来捕获带通信号; 一些应用个性化聚合与强化学习 [2] 或神经架构搜索 [3]; 其他人试图不仅从直接邻居，而且从嵌入空间聚合消息 [4] 或高阶邻居 [5]。这些聚合器的设计取得了良好的性能，但是，他们主要集中在单轮消息传递过程中，忽略了多跳消息的集成。另一方面，研究了多跳信息的有效利用，主要通过设计各种跳连接来实现。一些人（Klicpera 等

人 [6]; Chen 等人 [7]）提出了通过堆叠多个 GNN 层来防止自我或本地信息被“洗掉”的初始连接; 受到 ResNet 的启发 [8]，一些作品（Li 等人 [9]；Chen 等人 [10]）探索了在 GNN 上应用残差连接来改善梯度; 其他人将中间 GNN 层的输出与精心设计的组件相结合，例如 concat [11]、可学习权重，符号权重或 RNN 类架构。这些工作是简单而有效的，然而，它们只能在几跳内对信息进行建模，但不能在某些阶上精确地对信息进行建模，这导致不同阶的特征的混合; 此外，这些方法中的许多方法无法为每个节点做出个性化决策。这些缺陷导致次优性能。除了关注模型方面，其他方法关注如何修改图结构。这些方法被称为“图重新布线”，包括随机移除边缘或节点，或者用启发式算法计算新的图。

3 本文方法

3.1 *OrderedGNN*

文章提出了一个有序的形式的消息传递机制。也就是说，某个节点的节点嵌入中的神经元与该节点的根树中的层次结构对齐。在这里，节点的根树指的是以节点本身为根，其邻居为子节点的树。递归地，对于每个子节点，其子节点再次是子节点的相邻节点。（比照图 1）。我们通过提出一种新的有序门控机制来实现对齐，该机制控制神经元的分配以编码具有不同深度的子树。在大量数据集上的实验结果表明，该模型可以同时缓解异质性和过度平滑问题。我们的模式具有以下优点：

1. 在我们设计的联合收割机阶段的引导下，根树层次结构，一个非常普遍的拓扑归纳偏见与最少的假设邻居的分布，这允许一个灵活的整合信息在不同的顺序中，并导致上级性能的异质性和同质性。

2. 有序选通机制防止了跳内节点特征的混合，使我们能够以不同的顺序对信息进行建模。这为在异质性下为每个节点提取相似的邻域模式打开了一扇大门; 这也使得很容易保留自我和局部信息，从而有效地缓解过度平滑。

3. 我们的模型通过显式的门控机制将相邻结构与节点嵌入中的块对齐，因此门控机制可以提供可视化来揭示数据的连接类型并提供可解释性。

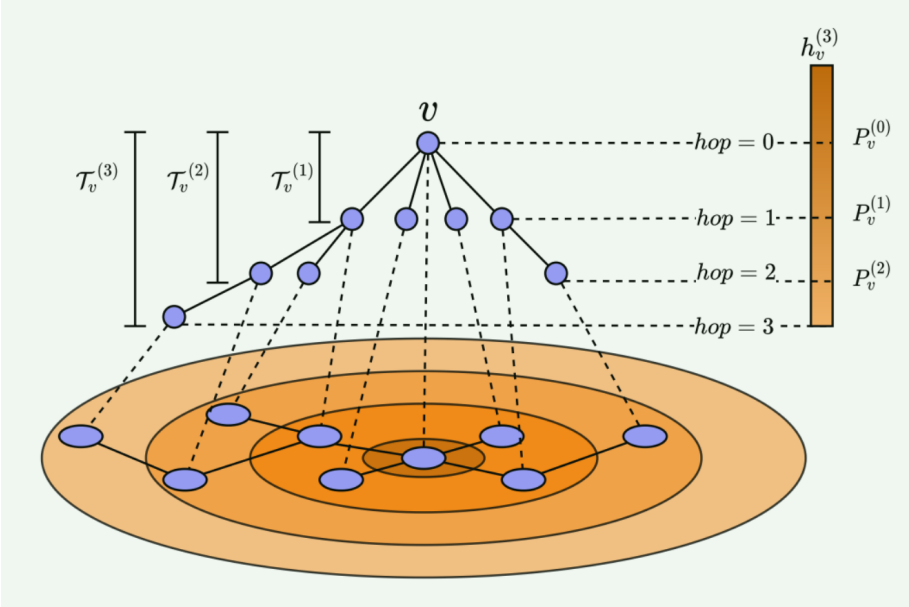

图 1. 对齐有根树的层次结构

3.2 对齐节点表示与根数层次

从节点的角度来看，围绕它的多跳内的相邻节点可以自然地呈现为有根树。对于节点 v，其 k 阶根树 $T_V^{(k)}$ 是通过将其第一阶邻居作为其子节点，然后递归地为每个子节点，将子节点的相邻节点作为其子节点（除了也是节点祖先的子节点），直到其 k 阶邻居（图 1）。

由于每个 GNN 层中只有一轮消息传递，因此堆叠多层 GNN 对应于进行多轮消息传递。并且由于对于每一轮消息传递，只有直接邻居交换它们的信息，因此第 k 层 $h_V^{(k)}$ 中 v 的节点表示仅取决于其第 k 阶根树 $T_V^{(k)}$ 内的节点。我们的目标是将 k 阶根树中的信息正确编码为固定大小的向量 $h_V^{(k)}$。

注意 v 的根树之间有一个嵌套结构，我们将利用这个结构作为基础，将根树与 v 的节点嵌入对齐。很明显，第 $k-1$ 根树 $T_V^{(k-1)}$ 是第 k 根树 $T_V^{(k)}$ 的子树，具有相同的根 v 和相同的树结构，直到深度 k–1。因此，我们有（假设模型有 K 层）

$$\mathcal{T}_v^{(0)} \subseteq \mathcal{T}_v^{(1)} \subseteq \cdots \subseteq \mathcal{T}_v^{(k)} \subseteq \cdots \subseteq \mathcal{T}_v^{(K)}. \tag{1}$$

其中 $T_v^{(0)}$ 对应于用于表示节点的最初信息的神经元。随着 k 的增长，$T_v^{(k)}$ 变得更大、更复杂，需要更多的神经元来编码它的信息。因此，很自然地假设用于表示 $T_V^{(k-1)}$ 内的消息传递的神经元应该是 $T_V^{(k)}$ 的神经元的子集。相应地，我们将在 v 的节点嵌入中实现相同的嵌套结构。我们首先将 h_v 中的所有神经元排列成一个序列，因此我们可以用数字索引它们。对于 $T_V^{(k-1)}$ 内的信息，我们允许它在前 $P_V^{(k-1)}$ 个神经元内编码。对于下一级根树 $T_V^{(k)}$，涉及更多的神经元，对应于前 $P_V^{(k)}$ 个神经元。显然这里我们有 $P_v^{(k-1)} \leq P_v^{(k)}$.。

这种排列神经元的方式有一个关键的特性。考虑两个分裂点 $P_V^{(k-1)}$ 和 $P_V^{(k)}$ 之间的神经元。它们反映了 $k-1$ 和 k 轮消息传递之间的增量，以 k 阶精确地编码邻居信息。直观地说，它以有序的形式组织消息传递，避免了在一跳内不同顺序的邻居特征的混合。我们用门控机制实现有序神经元，因此将门可视化为探测模型消息传递的一种方式是很自然的。

现在我们有了 v 的节点嵌入，它被一组分裂点分裂

$$\mathcal{P}_v^{(0)} \subseteq \mathcal{P}_v^{(1)} \subseteq \cdots \subseteq \mathcal{P}_v^{(k)} \subseteq \cdots \subseteq \mathcal{P}_v^{(K)}. \tag{2}$$

3.3 *THESPLITPOINTS*

考虑分裂点 $P_v^{(k)}$，它将有序节点嵌入分为两个块，左块包含索引在 $[0,\ P_v^{(k-1)}]$ 和右块 $[P_v^{(k)},\ D]$ 之间的神经元。我们可以通过 D 维门控向量 $g_v^{(k)}$ 来表示分裂，其中其左侧 $P_v^{(k)}$ 条目为 1，其余条目为 0。该选通向量然后可以在联合收割机阶段中使用，以控制作为第 k 层 $h_v^{(k-1)}$ 的输入的自我表示与聚合上下文 $m_v^{(k)}$ 之间的整合。

$$h_v^{(k)} = g_v^{(k)} \circ h_v^{(k-1)} + \left(1 - g_v^{(k)}\right) \circ m_v^{(k)} \tag{3}$$

其中，x 表示逐元素乘法。直觉上，对于 $h_v^{(k)}$ 中的左 $P_v^{(k)}$ 神经元，由于它们在 $g_v^{(k)}$ 中对应的门控向量是 1，它们抑制了新聚合的上下文 $m_v^{(k)}$，并且只继承最后一层的输出 $g_v^{(k-1)}$。对于门控向量为 0 的右侧神经元块，反之亦然。理想情况下，我们更喜欢一个清晰的分割，在 $P_v^{(k)}$ 处有一个清晰的边界，在 $g_v^{(k)}$ 中有二进制门控向量。然而，这将导致离散化操作，使模型不可微。在这项工作中，我们通过预测它们的期望来“软化”门，从而保持整个模型的可微性。期望向量 $\hat{g}_v^{(k)}$ 被表示为节点嵌入中的每个位置是分裂点 $P_v^{(k)}$ 的概率的累积和。具体地，$\hat{g}_v^{(k)}$ 被参数化为

$$\hat{g}_v^{(k)} = \text{cumax}_{\leftarrow}\left(f_\xi^{(k)}\left(h_v^{(k-1)}, m_v^{(k)}\right)\right) = \text{cumax}_{\leftarrow}\left(W^{(k)}\left[h_v^{(k-1)}; m_v^{(k)}\right] + b^{(k)}\right) \quad (6) \tag{4}$$

3.4 可微 OR Operator

在上面的部分中，我们提供了一种可微分的方法来预测分裂点的位置，以及实现分裂操作的相应门。但是，不能保证预测的分割点符合公式 4中的限制。不确定公式 4会破坏有根树和节点嵌入之间的对齐。为了确保这样的相对幅度，我们在新计算的门控向量 $\hat{g}_v^{(k)}$ 和它的前任之间进行逐位 OR 运算符。我们将符号改为 $\tilde{g}_v^{(k)}$，以将其与原始计算的门控向量 $\hat{g}_v^{(k)}$ 区分开来。同样，OR 算子本质上是不可微的。相反，我们通过以下方式实现它的软化版本：

$$\tilde{g}_v^{(k)} = \text{SOFTOR}(\tilde{g}_v^{(k-1)}, \hat{g}_v^{(k)}) = \tilde{g}_v^{(k-1)} + (1 - \tilde{g}_v^{(k-1)}) \circ \hat{g}_v^{(k)} \tag{5}$$

3.5 总结

通过上面的讨论，我们可以通过将所有内容放在一起来获得第 k 个 GNN 层的更新规则，从而得到 Ordered GNN 模型：

$$\begin{aligned}
m_v^{(k)} &= \text{MEAN}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right) \\
\hat{g}_v^{(k)} &= \text{cumax}_{\leftarrow}\left(f_\xi^{(k)}\left(h_v^{(k-1)}, m_v^{(k)}\right)\right) \\
\tilde{g}_v^{(k)} &= \text{SOFTOR}\left(\tilde{g}_v^{(k-1)}, \hat{g}_v^{(k)}\right) \\
h_v^{(k)} &= \tilde{g}_v^{(k)} \circ h_v^{(k-1)} + \left(1 - \tilde{g}_v^{(k)}\right) \circ m_v^{(k)}
\end{aligned} \tag{6}$$

第一个方程对应于聚集阶段，而后三个方程对应于组合阶段。在实践中，我们可以通过应用分块技巧来减少门控网络中的参数（第二个等式中的参数），我们设置 $D_m = D/C$ 作为选通向量的大小，其中 D 是 GNN 的隐藏状态的维度，并且 C 是分块大小因子。在这种设置中，每个门控制 C 个神经元，从而大大减少了要预测的门的数量。

4 复现细节

4.1 与已有开源代码对比

文章接收时间比较短，只开源了部分代码。文章在 3 个同质数据集及 6 个异质数据集上进行了实验，但是文章只公开了其中两个数据集的实验代码。同时文章也在不同的网络层次上测试了模型的性能，我对其进行了扩展，加深了网络层次，验证模型缓解过度平滑的能力。同时我也在两个更大的数据集上进行了实验，进一步验证模型的性能。此部分为必填内容。如果没有参考任何相关源代码，请在此明确申明。如果复现过程中引用参考了任何其他人发布的代码，请列出所有引用代码并详细描述使用情况。同时应在此部分突出你自己的工作，包括创新增量、显著改进或者新功能等，应该有足够差异和优势来证明你的工作量与技术贡献。

4.2 实验

在本节中，我在 11 个数据集上测试了文章中的模型。我主要从三个方面来评估模型：在同质和异质数据集上的性能，以及对过度平滑问题的鲁棒性。还有数据集的大小，这两个数据集分别来自大规模同质性和异质性基准。

数据集和实验设置：对于同质性数据，我使用三个引文网络数据集，即，Cora、CiteSeer 和 PubMed。同时六个异质网络数据集，即 Actor，Texas，Cornell，Wisconsin，Squirrel 和 Chameleon，被用来评估我们的模型在异质性数据集中的性能。在引文网络中，节点和边分别对应于文档和引文。节点特征是文档的特征表示。在 Web 网络中，节点和边表示网页和超链接，节点特征是网页的特征表示。对于过度平滑问题，我们使用 Cora，CiteSeer 和 PubMed 作为测试平台。对于两个较大的数据集，我们选择了 ogbn-arxiv 和 arXiv-year。这两个数据集都是引文网络，代表论文和引用。在 ogbn-arxiv 中，节点标签代表论文的学术主题，这往往是一个同质性设置，而在 arXivyear 中，节点标签代表出版年份，因此表现出更多的异质性标签模式。我们包括边缘同质性得分 $H(G)$，其代表同质性水平，以及表1中总结的数据集的统计数据。我们使用 Adam 优化器，并对 f_θ 和 $f_\xi^{(k)}$ 应用 *dropout* 和 $L2$ 正则化。我们还插入 $LayerNorm$（每隔一层。我将隐藏层维度设置为 256，块大小因子设置为 4。对于每个数据集，除非另有说明，否则使用 8 层模型（已经比大多数流行的 GNN 模型更深）。我们执行网格搜索来调整所有模型的超参数。

4.2.1 同质和异质数据集

我评估了文章的模型在上述 9 个数据集上的性能，包括三个同质性数据集和六个异质性数据集。结果示于表2中。针对每个数据集都进行了五次实验，最后记录的结果取平均值。

4.2.2 过度平滑

为了验证模型的过度平滑性能，在三个同质数据集上针对不同的网络层次进行了实验，每个实验进行了五次取平均值得到的结果如表3

表 1. 数据集统计

Dataset	Classes	Nodes	Edges	Features	H(G)
Cora	7	2708	5429	1433	0.81
PubMed	3	19717	44338	500	0.80
CiteSeer	6	3327	4732	3703	0.74
Cornell	5	183	280	1703	0.30
Chameleon	5	2277	31421	2325	0.23
Squirrel	5	5201	198493	2089	0.22
Actor	5	7600	26752	931	0.22
Texas	5	183	295	1703	0.21
Wisconsin	5	251	466	1703	0.21
ogbn-arxiv	40	169343	1166243	128	0.66
arXiv-year	5	169343	1166243	128	0.22

表 2. 在同质和异质数据集上的实验结果

Dataset	Texas	Wisconsin	Actor	Squirrel	Chameleon	Cornell	CiteSeer	PubMed	Cora
OrderedGNN	0.94	0.90	0.36	0.62	0.71	0.83	0.76	0.89	0.86
原文	0.86	0.88	0.37	0.62	0.72	0.87	0.77	0.90	0.88

表 3. 针对过度平滑问题在不同网络层次下节点分类的结果

Dataset	2	4	8	16	32	64
Cora	0.875	0.847	0.841	0.835	0.832	0.881
CiteSeer	0.758	0.772	0.755	0.765	0.758	0.815
PubMed	0.879	0.898	0.901	0.899	0.897	0.906

5 实验结果分析

5.1 同质和异质数据集实验结果分析

文章的 OrderedGNN 模型在同质和异质数据集上都能始终如一地实现 SOTA。在同质数据集上，OrderedGNN 的性能与最优的 GCNII 相近。然而，GCNII 在异质数据集上的表现较差，在 Squirrel 数据集上留下了超过 20% 的差距。在这个数据集上的惊人表现可能与“过度挤压”问题有关。我们还在异质数据集上实现了最优的性能，并在所有数据集上始终优于基线。在同质和异质数据上的上级性能显示了我们模型的通用性，说明模型可以很好的处理同质和异质数据集，我们提出的门控机制起到了很好的效果。

5.2 过度平滑实验结果分析

性能如表3所示。与基准模型相比，我们的模型在所有数据集和所有层上都能保持最佳性能。此外，对于每个数据集，我们的模型的性能不会像一些基线那样显著下降。请注意，这是在不使用诸如 DropEdge 之类的训练技巧的情况下实现的。甚至，在我将模型加深到 64 层时，模型依然能够保持同样好的性能。这表明，我们的模型可以很好地缓解过度平滑的问题。

6 总结与展望

文章提出了 Ordered GNN 模型，一种新的 GNN 模型，该模型将中心节点的根树的层次结构与其节点表示中的有序神经元对齐。通过将传递的信息排序到嵌入节点中，这种对齐实现了一种更有组织的消息传递机制。它们与节点的距离该模型的结果是更好的解释和更强的性能。通过本次复现工作证实它在各种数据集上达到了最先进的水平，包括同质性和异质性数据集设置。此外，通过扩展实验，证明当模型变得非常深时，该模型可以有效地防止过度平滑。对于该方面未来的研究，我认为可以从以下三个方面进行探索。

6.1 图数据的异质性

异质性是 GNN 的一个棘手问题，一种广泛接受的处理方法是允许消息系数为负，该消息签名让模型区分有害的邻居信息（“消极的异质性”）并拒绝它们，我们也可以用我们提出的门控机制来做到这一点; 一些最近的工作表明，也有对 GNN 友好的“积极的异质性”，但目前还没有出现主动捕获它而不是被动接受它的方法。值得注意的是，我们的模型对组合阶段进行了设计，因此我们可以将签名的消息合并到聚合阶段，以进一步提高性能。

6.2 过度平滑

过度平滑是图学习领域的一个活跃研究方向。尽管我们的模型主要关注消息传递机制中的组合阶段，但我们能够将其与先前取得成功的方法联系起来，为我们模型的工作原理提供更深入的理解。与 JK-Net 类似，该模型在联合收割阶段允许不同神经元块表示不同跳的信息。我们的模型设计使得跳数为 0 的节点自我信息被锁定在指定的神经元块中，从而有效地保留该信息。这一设计与 APPNP 和 GCNII 等方法中的初始连接类似。尽管这些有用的跳过连接似乎是特定于任务的，但我们的有序门控机制为将它们整合提供了一种统一的方式。通

过引入有序门控机制，我们的模型超越了以往简单地在不同跳之间进行消息聚合的模型结构。我们进一步建模了上下文之间的内在关系，形成了一种树层次结构。这一创新使得模型能够更好地理解图中节点之间的关联，提高了对图结构的表示学习能力。

6.3 有序的神经元

有序神经元最初提出用于从基于 LSTM 的语言模型中归纳句法树结构。该方法采用两个主门控网络来控制神经元在不同位置的更新频率，并使用句法距离将门控结果转换为语法树。虽然我们的方法也依赖于对神经元进行排序，但归纳的方向却相反：我们的模型先验地知道了根树结构，而门控结果是基于树结构进行学习的。这意味着我们的方法在学习过程中从树结构中获得信息，而不是将信息转化为树结构。这一差异使得我们的方法能够更灵活地适应各种树结构，并更好地捕捉节点之间的关系。

参考文献

[1] Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering gcn: Overcoming oversmoothness in graph convolutional networks. *Advances in neural information processing systems*, 33:14498–14508, 2020.

[2] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

[3] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

[4] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

[5] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[6] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[7] Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. Straight to the tree: Constituency parsing with neural syntactic distance. *arXiv preprint arXiv:1806.04168*, 2018.

[8] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*, 2018.

[9] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

[10] Ke Sun, Zhanxing Zhu, and Zhouchen Lin. Adagcn: Adaboosting graph convolutional networks into deep models. *arXiv preprint arXiv:1908.05081*, 2019.

[11] Susheel Suresh, Vinith Budde, Jennifer Neville, Pan Li, and Jianzhu Ma. Breaking the limit of graph neural networks by improving the assortativity of graphs with local mixing patterns. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1541–1551, 2021.