GCN 深度问题总结

1. 神经网络的深度

(1) 为什么深度神经网络有效？

深度神经网络同时具有逐层处理、特征变换和高模型复杂度的特点。决策树也是逐层处理，但是它的深度不会超过特征个数，所以模型复杂度有限，并且决策树的学习过程没有特征变换，始终在同一个原始特征空间学习。加宽神经网络（增加神经元）也可以增加模型复杂度提高学习能力，但是缺少对特征的逐层抽象加工，所以效果不如深度神经网络。

(2) 加深网络要解决哪些问题？

① 过拟合：加深神经网络增加了模型的复杂度从而提高了模型的学习能力，但是如果模型的复杂度超过了训练集的规模就会产生过拟合问题，此时可以采用一些手段降低模型复杂度，例如各种正则化方法：Early Stopping、L_1L_2 范数、Max-Norm、Dropout、数据增强等。或者使用更大的数据集训练。

② 梯度消失/爆炸：随着层数加深，反向传播过程中梯度更容易不稳定，最终导致网络不收敛，可以通过 Xavier 和 He 初始化、更换非饱和激活函数、批处理规范化 BN、梯度修剪等方法在很大程度上解决该问题。

③ 训练太慢：训练一个大规模深度神经网络往往需要很多时间，可以通过更换优化器如 AdaGrad、RMSProp、Adam 等加速收敛，也可以通过 GPU 加速训练。

2. GCN 深度问题

(1) 问题来源

Kipf 在**[1]**中运用近似技巧推导出了 GCN 的逐层传播公式，并应用于图的半监督结点分类任务，实验发现 2-3 层的 GCN 效果最好，继续增加层数性能会下降，超过一定层数后性能会骤降。作者初步尝试了残差连接，旨在增强前一层的信息流动，但是超过 3 层后性能仍会有所下降。后续的 GCN 深度相关的工作都建立在 Kipf 的 GCN 模型的基础之上。

(2) 可能原因

过拟合和梯度消失/爆炸是深度神经网络存在已久并且很大程度上已被解决的问题，而过光滑则是深度 GCN 特有的问题，我认为也是 GCN 无法叠深的主要原因。已有的研究都是围绕这几点展开。

① 过拟合

② 梯度消失/爆炸

③ 过光滑

图卷积是一种特殊形式的拉普拉斯平滑-对称拉普拉斯平滑。拉普拉斯平滑计算了结点的新特征，即结点自身和邻居的加权平均。同一类簇的结点倾向于连接得更紧密，平滑后结点特征变得更相似，因此使分类任务变得简单**[8]**。

重复使用拉普拉斯平滑，结点的特征以及图的每个连通分量会收敛到相同的值。对于对称的拉普拉斯平滑，收敛到的值与顶点度数的二分之一次幂成正比**[8]**。如果每个类簇恰好是一个连通分量，那么这将有利于分类任务。但是事实上实验用到的图数据集，不同类簇之间是连通的，甚至整张图都是连通的，而重复使用拉普拉斯平滑可能会混合不同类簇中的顶点的特征使得它们难以被区分，随着层数增加最终所有结点都收敛到相似的值（该值与整张图的性质相关**[5]**）完全无法区分，数据集越小越严重。

以上理解可以很好地解释 GCN 超过 2-3 层后性能先缓慢下降，再急剧下降。参照**图 4**，同一类簇的内部结点倾向于连接比较密集，越往中心连接越密集，而边缘结点连接比较稀疏，不同类簇间连接也比较稀疏。当层数较少时混杂的结点也较少（主要是边缘结点），此时性能缓慢下降，当层数到达某一阈值时，平滑范围触及了连接密集区域，混杂的结点骤增，因而性能急剧下降。1-3 层时性能上升可以解释为，此时浅层学习到的结构信息匮乏为主要问题，增加层数可以学习到更多结构信息。

以上理解也可以较好地解释 ResGCN、JK-Net、RGNN 等“深度”GCN 模型（并不够深）的作用和不足。

(3) 已有研究

已有研究主要可以分为两种思路，一种借鉴 ResNet 等将网络加深，一种借鉴 Inception 将网络“加宽”。形式上或是改进网络结构，或是变换邻接矩阵。大多数模型我理解为本质上是不同深度的 GCN 的集成，通过不同的方式缓解了过光滑，但是都没有从根本上解决问题。与一百多层的 ResNet 相比，这些深度 GCN 模型都远不够深。**[13]**虽然叠到了 56 层，但是只在点云语义分割任务上进行了实验，该任务用到的图数据集不连通不受过光滑影响**[14]**，并没有解决 GCN 无法叠深的问题。

“深度”或“宽度”GCN

论文	模型	针对问题	相似技术	具体方法	实验深度
Semi-Supervised Classification with Graph Convolutional Networks	ResGCN	过光滑	ResNet	添加 residual connections	范围：1-10 层 最佳：2-3 层
Representation Learning on Graphs with Jumping Knowledge Networks	JK-Net	过光滑	/	layer-aggregation	范围：1-6 层 最佳：6 层
Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks	Cluster-GCN with *	梯度消失，过光滑	/	改进对称归一化矩阵同时进行正则化	范围：1-8 层 最佳：5 层
N-GCN: Multi-scale Graph Convolution for Semi-supervised Node Classification	N-GCN	过光滑	Inception	组合不同尺度感受野的 GCN	“加宽”网络
Residual or Gate? Towards Deeper Graph Neural Networks for Inductive Graph Representation Learning	RGNN	过光滑	RNN	使用 RNN 对各层之间的长期依赖建模	范围：1-8 层 最佳：2-4 层
DeepGCNs: Can GCNs Go as Deep as CNNs?	ResGCN	梯度消失/爆炸	ResNet	添加 residual connections	最佳：56 层
	DenseGCN		DenseNet	添加 dense connections	最佳：28 层
DropEdge: Towards Deep Graph Convolutional Networks on Node Classification	DropEdge	过拟合，过光滑	Dropout	从输入图随机删除一定数量的边	范围：1-32 层 最佳：8 层
Break the Ceiling: Stronger Multi-scale Deep Graph Convolutional Networks	Snowball	过光滑	DenseNet	添加 dense connections	范围：1-15 最佳：/
	Truncated Krylov		Inception	组合不同尺度感受野的 GCN	

3. 对 Predict then Propagate Graph Neural Networks meet Personalized PageRank[4]的一些疑问

PPNP（APPNP）架构分为 Prediction 和 Personalized PageRank，Prediction 部分可以用 NN 提取结点自身的高级特征，Personalized PageRank 部分通过迭代计算聚合邻居信息（平衡局部和全局）。回头看这篇论文时产生了一些疑问：

(1) PageRank 和 Personalized PageRank 都是基于一定经验假设不含可学习参数的图传播算法，类似的算法还有 HITS 等，这类算法都是收敛的，可以通过

迭代求解。**[19]**

① 适用于 PageRank（或 Personalized PageRank）的经验假设不一定在所有类型的图数据集的特定任务上都能发挥最佳效果。例如，PageRank 最初提出是为了解决网页排序问题，基于两条经验假设：数量假设——被更多网页链接到的网页更重要；权重假设——有更少外链的网页将会传递更高的权重。而在解决其他类型图数据集的特定任务时，可能就需要不同的或者更全面的经验假设。

② Personalized PageRank 是基于经验假设设计的，是直接进行量化计算的，不需要参数学习过程，而神经网络却具有可学习参数，可以自动学习数据的模式。PPNP（APPNP）并没有解决也没有缓解 GCN 的深度问题，而是回避了 GCN 的使用，但是也失去了神经网络的优势。

③ 论文实验表明了 PPNP（APPNP）随着 K（迭代次数）增加性能会逐步上升并趋于稳定并与 GCN 作了比较，但是 Personalized PageRank 本来就是收敛的，不能说明对解决 GCN 深度问题有帮助。

4. 我的想法和后续实验

(1) 如果能从根本上解决随着层数加深不同类簇结点加速混合的问题，过光滑问题应该就能得到有效解决了。我想到的思路有：

① 能否对图进行分割等预处理，将图分为不同的连通分量，理想情况下，不同类簇间完全不连通，也就不存在过光滑问题了。

② 能否控制结点对邻居结点的聚合权重，理想情况下，A 类簇的边缘结点对位于 B 类簇的邻居结点的聚合权重为 0（或者该值非常小，也能极大缓解），不同类簇的结点不会产生混合，也就不存在过光滑问题了。

③ 已有研究中平衡局部信息与全局信息，组合不同尺度感受野等方法本质上是相似的，在一定程度上缓解了过光滑问题。可以尝试将 PPNP 平衡局部和全局的方法引入 GCN 中，如

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

或者在残差连接上进行改动：

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

④ 部分论文提到了“增强结点自身特征”，我想到一个结点的信息主要由两部分组成，自身信息和结构信息（邻居结点）。随着层数加深，越来越多的邻居结点被聚合，结点自身的信息越来越匮乏，最终得到的 embedding 可能几乎不包含结点自身信息。是否可以在最后将得到的 embedding 与结点的原始特征（或者经过提取的高层特征，提取过程不进行聚合操作，形如

PPNP 中的 Prediction）结合起来用于结点分类从而辅助提高性能。

5. 相关论文要点理解和疑问

(1) Semi-Supervised Classification with Graph Convolutional Networks[1]

① 要点

A. 从谱图卷积近似推导出了 GCN 逐层传播公式，并应用在图的半监督结点分类任务上，后续的各种神经网络或多或少受其启发。

B. 从 WEISFEILER-LEHMAN 算法的角度研究了 GCN，论文**[9]**基于 WEISFEILER-LEHMAN 算法研究了 GCN 及其变种的表达能力，设计了 GIN 并在图分类任务上取得了不错的效果，也许也是受**[1]**启发。

C. 随机初始化权重矩阵的未经过训练的 GCN 也能取得不错的效果，说明了添加了自循环的对称归一化后的邻接矩阵是 GCN 的关键。

D. 实验发现 2-3 层的 GCN 效果最好，再增加层数性能将会下降，初步尝试了添加残差连接缓解该问题。后续的关于 GCN 深度问题的研究便来源于此。

图 1

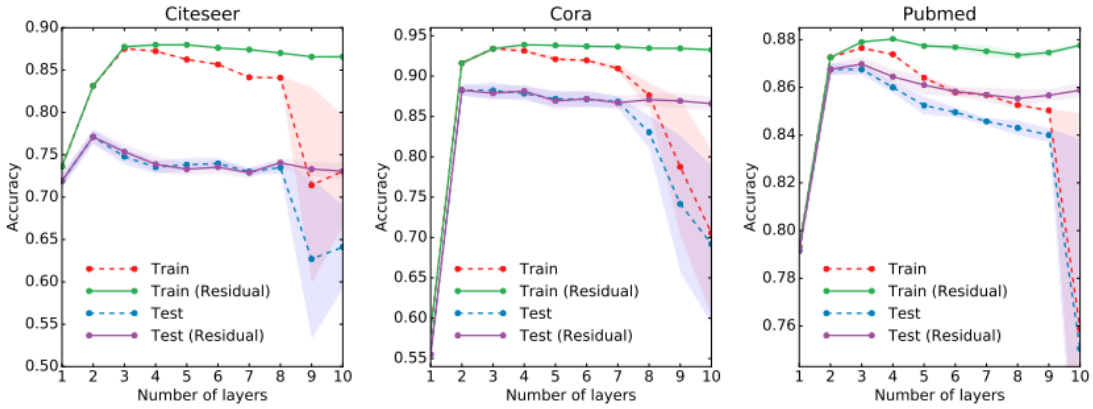


Figure 5: Influence of model depth (number of layers) on classification performance. Markers denote mean classification accuracy (training vs. testing) for 5-fold cross-validation. Shaded areas denote standard error. We show results both for a standard GCN model (dashed lines) and a model with added residual connections (He et al., 2016) between hidden layers (solid lines).

② 理解和疑问

A. 从**图 1** 可以看出，GCN 超过 3 层后，随着层数增加，训练准确率和测试准确率都在下降，可以排除过拟合。过光滑的可能性更大。

表 1

Table 1: Dataset statistics, as reported in Yang et al. (2016).

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

B. 结合**表 1** 可以发现，随着数据集变大，取得最佳性能的深度也在增加。增大数据集既可以缓解过拟合，也可以缓解过光滑。

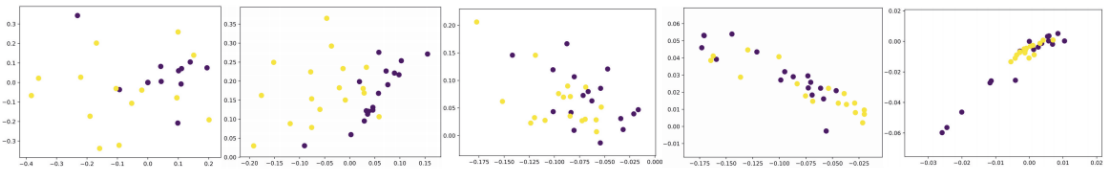
(2) Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning[8]

① 要点

A. 证明了图卷积是一种特殊形式的拉普拉斯平滑-对称拉普拉斯平滑。拉普拉斯平滑计算了结点的新特征，也就是结点自身和邻居的加权平均。同一类簇的结点倾向于连接的更紧密，平滑后结点特征变得更相似，因此使分类任务变得简单。

B. 证明了重复使用拉普拉斯平滑，结点的特征以及图的每个连通分量会收敛到相同的值。对于对称的拉普拉斯平滑，收敛到的值与顶点度数的二分之一次幂成正比。如果每个类簇恰好是一个连通分量，那么这将有利于分类任务。但是事实上实验用到的图数据集，不同类簇之间是连通的，甚至整张图都是连通的，而重复使用拉普拉斯平滑可能会混合不同类簇中的顶点的特征使得它们难以被区分，随着层数增加最终所有结点都收敛到相似的值完全无法区分，数据集越小越严重。**图 2** 展示了层数为 1-2 层时，Zachary's karate club 数据集（规模很小）不同类簇的结点越来越可分，但是随着层数继续增加，不同类簇的结点混合在一起难以区分。

图 2



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

C. 提出协同训练和自训练的方法解决浅层 GCN 存在的问题：在半监督学习中标签很少的情况下无法将标签传播到整个图中。（因为与深度问题关系不大，这部分没有细看）

② 理解和疑问

A. 对**图 2** 的分析也可以用来解释**图 1**。更贴切的解释需要结合**[5]**的理解：一个 K 层 GCN 聚合了结点的 K-hop 范围的邻居信息。**图 1** 的数据集比**图 2** 大很多，但是随着层数加深，首先一个类簇的边缘结点会混入其他类簇的边缘结点信息导致难以区分，GCN 的性能开始下降。到达一定层数后，性

能会骤降，我猜想原因是，同一类簇的内部结点连接比较密集，越往中心连接越密集，而边缘结点连接比较稀疏，不同类簇间连接也比较稀疏，这样的图结构特点（形如**图** ）导致层数较少时混杂的结点也较少，当层数到达某一阈值时，平滑范围触及了连接密集区域，混杂的结点骤增，因而性能急剧下降。

B. **图 1** 中的 ResGCN 相比 GCN 性能有了很大的提升。ResNet 源自**[16]**，作者通过增加短路连接，构建了深层 CNN 并应用在图像分类上取得了巨大的性能提升。ResNet 解决了什么问题？根据**[16]**的分析，normalized initialization 和 intermediate normalization layers 已经基本解决了梯度消失/梯度爆炸问题。从**图 3** 可以看到，层数增加后训练误差和测试误差都很大，可以排除过拟合问题。作者认为假设一个比较浅的网络已经可以达到不错的效果，那么即使之后堆上去的网络什么也不做，模型的效果也不会变差。Residual Learning 的初衷，就是让模型的内部结构至少有恒等映射的能力，以保证在堆叠网络的过程中，网络至少不会因为继续堆叠而产生退化。

图 3

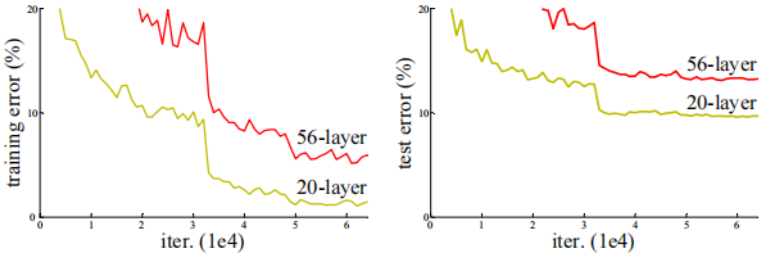


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

关于 ResNet 还有其他一些理解。

- **[17]**认为，即使 BN 处理后梯度的模稳定在正常范围，但是梯度的相关性会随着层数增加持续衰减。而经过证明，ResNet 可以有效减少这种相关性的衰减。
- **[18]**认为，浅层特征具有高分辨率低级语义，深层特征具有高级语义低分辨率，而 ResNet 可以实现不同分辨率特征的组合。进一步可以用 DenseNet 充分利用这个优点。
- ResNet 具有更加灵活的结构，在训练过程中，模型可以在每一部分选择更倾向于进行卷积与非线性变换还是恒等变换（什么也不做），或是将两者相结合，模型可以自适应本身的结构。

C. 以上 ResNet 的优点同样可以在深层 GCN 中起到作用，所以相比 GCN，ResGCN 的性能有较大提升。此外，我理解为 ResGCN 对过光滑也有较大缓解作用：

- ResGCN 可以组合高低层不同范围的邻居信息。
- 每增加一层，GCN 就会聚合更大一跳范围的邻居信息，随着层数增加会导致过光滑。而 ResGCN 保证了恒等变换的能力，即在某一层可以选择不去聚合邻居信息，也许可以自适应不同邻域结构结点的平滑范围需求。DenseNet 可以更充分发挥这两个优点，后期会通过实验进行验证。

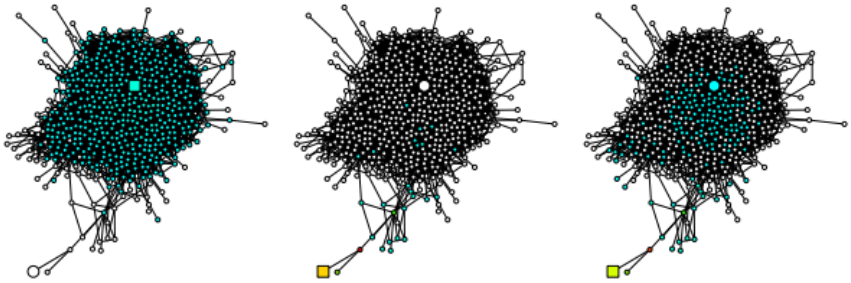
(3) Representation Learning on Graphs with Jumping Knowledge Networks[5]

① 要点

A. 从 K 步随机游走的角度分析了 K 层 GCN，随着层数增加 GCN 收敛到随机游走的极限分布，该分布是全图的性质，与随机游走起点无关，不再适合描述邻居结点的性质。

B. 固定层数的 GCN 无法满足不同邻域结构的结点对平滑范围的要求。如**图** 所示，位于连接紧密的中心结点，平滑范围扩散过快；位于连接稀疏的边缘结点，平滑范围扩散过快，但是一旦触及中心，平滑范围就会陡增。

图 4



(a) 4 steps at core (b) 4 steps at tree (c) 5 steps at tree

Figure 1. Expansion of a random walk (and hence influence distribution) starting at (square) nodes in subgraphs with different structures. Different subgraph structures result in very different neighborhood sizes.

C. ResGCN 大致相当于 lazy random walk：每一步都有很大几率停留在当前结点。并且这对所有结点都适用，所以无法满足不同邻域结构的结点的对平滑范围的要求。

D. 提出 JK-Net，以 layer-aggregation（Concatenation、Max-pooling、LSTM-attention）的方式自适应地融合不同层的信息（如**图 5** 所示），旨在平衡不

同邻域结构结点的 local 与 global 信息。

图 5

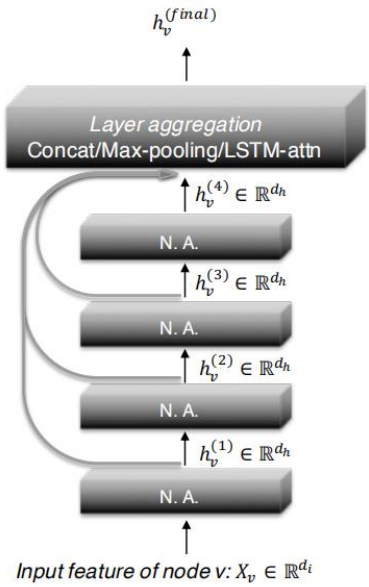


Figure 4. A 4-layer Jumping Knowledge Network (JK-Net). N.A. stands for neighborhood aggregation.

② **理解和疑问**

A. **[5]**和**[8]**的结论类似，最终所有结点的特征都会收敛到一个相似的值，但是**[8]**强调了同一连通分量。对于图的半监督结点分类问题，如果不同类簇为不同的连通分量，那么同一连通分量所有结点的特征收敛到相似值将会有利于分类。关键在于实验的数据集不同类簇间是连通的，从而带来结点特征混合的问题。我觉得**[8]**的解释更充分，结合**[5]**对结点的邻域结构的分析可以很好地解释深度 GCN 性能下降及其下降趋势。

B. 我的理解是，即使 JK-Net 能自适应地根据结点的邻域结构组合不同层的信息，但是由于 JK-Net 只在最后一层对所有层进行融合，层之间的传播方式没有改变，较深层产生的输出仍然存在不同类簇间的结点混合问题。事实上，JK-Net 的性能提升确实有限。

(4) How Powerful are Graph Neural Networks？[9]

① **要点**

A. 证明了在图同构测试问题上 Weisfeiler-Lehman 算法是 GNN 性能的上限，分析了 GCN、GraphSAGE 等在捕获图结构能力上的不足。

B. 提供了如何构建与 WL 算法一样有效的理论支撑，并据此构建了图同构网络 GIN。

② **理解**

A. 本文的分析主要针对图层次的分类任务，相关结论不一定适用于结点层次

的分类任务。

B. 结点的分类结果是由结点自身的信息和结点的结构信息共同决定的吗？而设计 GCN 等图神经网络框架的目的在于自动学习这两个维度的信息并融入 embedding 中。由此我有一个疑问，随着层数加深，聚合的邻居结点范围扩大，最后的 embedding 中的结点自身的信息会不会非常匮乏，即 embedding 中几乎只包含结点的结构信息，因此层数加深性能会下降。根据这个猜测，在做图的半监督结点分类时，是否可以在最后将得到的 embedding 与结点自身的特征共同用于分类任务（通过级联等方式）。

(5) Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks[11]

① 要点

A. **[11]**认为，近距离的邻居结点比远距离的邻居结点更重要，**[1]**通过添加残差连接促进了低层信息（邻近结点）向前流动，作者通过改进 GCN 传播公式中的对称归一化矩阵得到 $X^{(l+1)} = \sigma((A' + I)X^{(l)}W^{(l)})$来达到该目的。考虑到数值稳定，令 $\tilde{A} = (D + I)^{-1}(A + I)$；考虑到邻近结点的权重，采用以下正则化：$X^{(l+1)} = \sigma((\tilde{A} + \lambda \mathrm{diag}(\tilde{A}))X^{(l)}W^{(l)})$。

② 理解和疑问

A. **[1]**中的对称归一化的邻接矩阵添加的自循环也起到了相似的作用，这里再次强化了邻近结点的信息，同时提出的正则化方法相比添加自循环，添加残差连接考虑到了邻近结点的权重。

B. 基于矩阵的改进，当数据集庞大时，就会面临内存限制问题，而大数据集恰恰最需要更深的 GCN。但是在 DGL 的 GCN 实现中，采用的是消息发送与接收的方式进行邻居聚合和结点更新，没有涉及到邻接矩阵的计算，所以在代码实现时还是可行的。

(6) N-GCN: Multi-scale Graph Convolution for Semi-supervised Node Classification[6]

① 要点

A. 提出 N-GCN，在不同尺度下进行卷积，最后融合所有卷积结果得到结点的特征表示，通过对不同尺寸感受野的组合提高模型的表征能力。

$$\text{N-GCN}_{\text{fc}}(\hat{A}, A; W_{\text{fc}}, \theta) = \text{softmax}\Big(\big[\text{GCN}(\hat{A}^0, X; \theta^{(0)}) \,\vdots\, \text{GCN}(\hat{A}^1, X; \theta^{(1)}) \,\vdots\, \dots \big] W_{\text{fc}} \Big).$$

② **理解和疑问**

A. 思路类似于 Inception，从“宽度”上对网络进行拓展，在同一层级上运行多个不同尺寸的卷积核。

(7) Residual or Gate? Towards Deeper Graph Neural Networks for Inductive Graph Representation Learning[12]

① **要点**

A. **[12]**认为，对于一个 n 层 GCN，第 i 层捕获了 i-hop 邻居结点的信息，相邻层之间有依赖关系，使用 RNN（GRU，LSTM）对各层之间的长期依赖建模（如**图 6** 所示）。

图 6

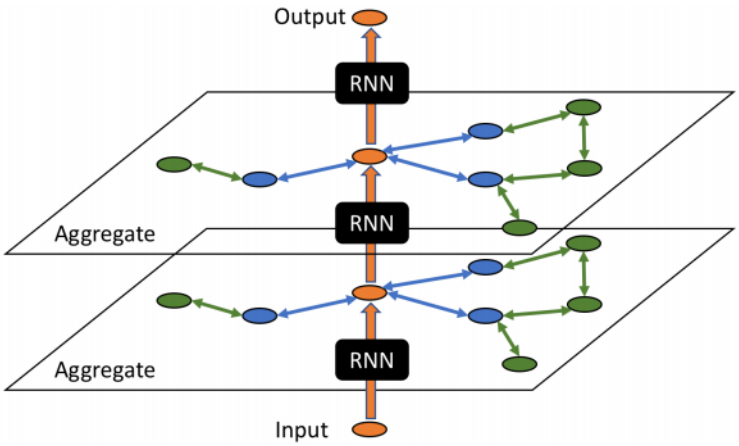


Fig. 1: A visual illustration of the feedforward process by a central node in a two-layer RGNN model.

② **理解和疑问**

A. 如**图 6** 所示，与 JK-Net（**图 5**）在最后一层使用 LSTM 融合各层信息不同，**[12]** 使用 RNN 对各层之间的长期依赖建模，缓解了更深的层不同类簇结点混合的问题。

(8) DeepGCNs: Can GCNs Go as Deep as CNNs?[13]

① **要点**

A. 借鉴深度 CNN，引入 residual connections, dense connections, 和 dilated convolutions 解决深度 GCN 梯度消失的问题，构建了一个 56 层的 GCN，

并在点云语义分割任务中取得了显著的性能提升。

② **理解和疑问**

A. normalized initialization 和 intermediate normalization layers 已经基本解决了梯度消失/梯度爆炸问题，提出 residual connections 主要是为了增强恒等变换的能力**[16]**，但是**[13]**引入 residual connections 的动机却是解决梯度消失问题，这让我感到疑惑。

B. 本文使用的点云数据集不同类簇之间是不连通的**[14]**，过光滑反而利于分割任务。

(9) DropEdge: Towards Deep Graph Convolutional Networks on Node Classification[14]

① **要点**

A. 提出 DropEdge，在每轮训练中从输入图随机删除一定数量的边，从而缓解过拟合和过光滑的问题，也可以用来增强其他 GCN 模型的性能。

② **理解和疑问**

A. 思路像是 Dropout 在图数据集上的扩展。作者在文中分析了 DropEdge 相比较 Dropout、DropNode、Graph-Sparsification 的优势。

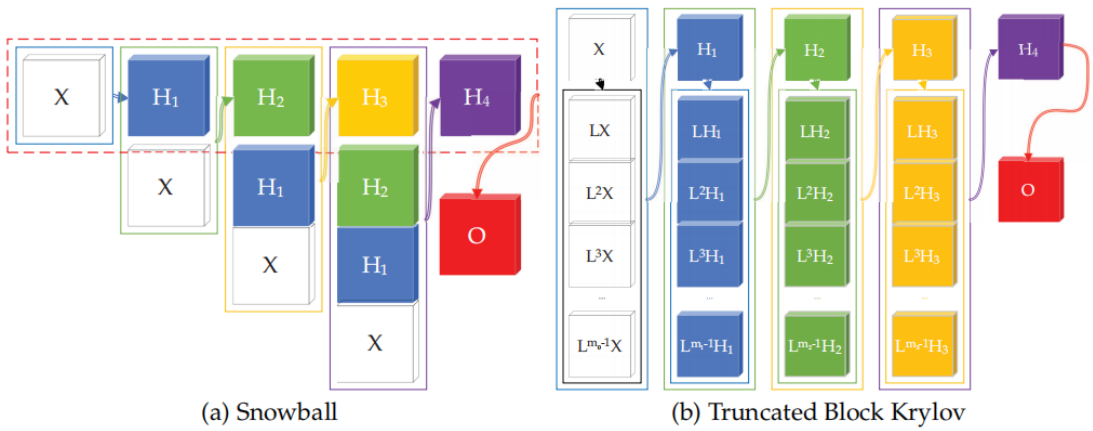
B. 最佳性能都是在 8 层取得，并且 GCN 叠加到一定层数时性能仍然会骤降，没有从根本上解决问题，只能作为一种辅助手段。

(10) Break the Ceiling: Stronger Multi-scale Deep Graph Convolutional Networks[15]

① **要点**

A. 分析了激活函数对 GCN 表达能力的影响，Tanh 保持列特征线性无关的效果最好，即使恒等函数也胜过 ReLU。

图 7



(a) Snowball (b) Truncated Block Krylov

Figure 2: Deep GCN Architectures

B. 将谱图卷积和深度 GCN 推广到块 Krylov 空间形式，在该空间下设计了两种结构（如**图 7**）：Snowball 和 Truncated Krylov。两者都利用了多尺度信息，在一定条件下，两种网络结构是等价的。

② 理解和疑问

A. Snowball 类似于 DenseNet，Truncated Krylov 类似于 Inception。

B. 激活函数赋予了神经网络拟合非线性的能力，但是不同的激活函数效果也不同，本文实验验证了最适合 GCN 的激活函数为 Tanh，即使恒等函数也比其他激活函数的效果要好。当激活函数为恒等函数时，相当于去除了 GCN 的非线性，GLN**[7]**和 SGC**[10]**也有涉及。

参考资料

[1] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[2] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[C]//Advances in neural information processing systems. 2017: 1024-1034.

[3] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[4] Klicpera J, Bojchevski A, Günnemann S. Predict then propagate: Graph neural networks meet personalized pagerank[J]. arXiv preprint arXiv:1810.05997, 2018.

[5] Xu K, Li C, Tian Y, et al. Representation learning on graphs with jumping knowledge networks[J]. arXiv preprint arXiv:1806.03536, 2018.

[6] Abu-El-Haija S, Kapoor A, Perozzi B, et al. N-gcn: Multi-scale graph convolution for semi-supervised node classification[J]. arXiv preprint arXiv:1802.08888, 2018.

[7] Thekumparampil K K, Wang C, Oh S, et al. Attention-based graph neural network for semi-supervised learning[J]. arXiv preprint arXiv:1803.03735, 2018.

[8] Li Q, Han Z, Wu X M. Deeper insights into graph convolutional networks for semi-supervised learning[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[9] Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks?[J]. arXiv preprint arXiv:1810.00826, 2018.

[10] Wu F, Zhang T, Souza Jr A H, et al. Simplifying graph convolutional networks[J]. arXiv preprint arXiv:1902.07153, 2019.

[11] Chiang W L, Liu X, Si S, et al. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 257-266.

[12] Huang B, Carley K M. Residual or gate? towards deeper graph neural networks for inductive graph representation learning[J]. arXiv preprint arXiv:1904.08035, 2019.

[13] Li G, Müller M, Thabet A, et al. Can GCNs Go as Deep as CNNs?[J]. arXiv preprint arXiv:1904.03751, 2019.

[14] Rong Y , Huang W , Xu T , et al. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification[J]. 2019.

[15] Luan S, Zhao M, Chang X W, et al. Break the Ceiling: Stronger Multi-scale Deep Graph Convolutional Networks[C]//Advances in Neural Information Processing Systems. 2019: 10943-10953.

[16] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[17] Balduzzi D, Frean M, Leary L, et al. The shattered gradients problem: If resnets are the answer, then what is the question?[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 342-350.

[18] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

[19] https://mp.weixin.qq.com/s?__biz=MzI2MDE5MTQxNg==&mid=2649687693&idx=1&sn=1dc186d11b7c802ef518b32785c78e4a&chksm=f276c35ac5014a4cfb7d8fa6eb636ba011c99f519f6079512370fbfcc2cd5a37a739de72a7f2&scene=21#wechat_redirect