

解决的问题

当图卷积网络GCN的层数超过两层时模型的表现会变差，这使得GCN只能作为浅层模型使用，且在对邻域节点的信息进行聚合时，即使同样是采用 k 层网络来聚合 k 跳邻居的信息，有着不同局部结构的顶点获得的信息也可能完全不同，以下图为例：

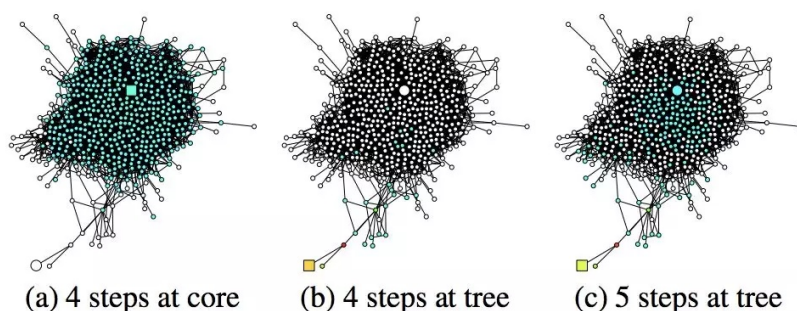


Figure 1. Expansion of a random walk (and hence influence distribution) starting at (square) nodes in subgraphs with different structures. Different subgraph structures result in very different neighborhood sizes.

图(a)中的顶点位于核心区域，因此采用4层网络把几乎整个图的信息都进行聚合了，而不是它的邻域，这会导致过度平滑，而图(b)中顶点位于图边缘的一个树状结构中，采取同样的4层网络只囊括了一小部分顶点的信息，只有在第5层囊括了核心顶点之后才有效地囊括了更多顶点的信息。

所以，对于处于核心区域的顶点，GCN中每多一层即每多一次卷积操作，节点的表达会更倾向全局，这导致核心区域的很多顶点的表示到最后没有区分性。对于这样的顶点应该减少GCN的层数来让顶点更倾向局部从而在表示上可以区分；而处于边缘的顶点，即使更新多次，聚合的信息也寥寥无几，对于这样的顶点应该增加GCN的层数，来学习到更充分的信息。因此，对于不同的顶点应该选取不同的层数，传统做法对于所有顶点都用一个值会带来偏差。

做法及创新

理论部分，论文主要讨论的问题是，在一个 k 层的GCN中，顶点 x 对顶点 y 的影响程度，即顶点 x 输入特征的改变，会对顶点 y 在最后一层得到的表示产生多大的变化，也可以说是顶点 y 对于顶点 x 有多敏感。假设输入的特征为 $X \in \mathbb{R}^{n \times f}$ ，输出的预测标签为 $Z \in \mathbb{R}^{n \times c}$ ，其中 n 为图中顶点数目， c 为类别数目， f 为特征数目，则这种影响程度可以表示为 $I(x, y) = \sum_i \sum_j \frac{\partial Z_{yi}}{\partial X_{xj}}$ 。

更特别地，论文证明了这个影响程度与从顶点 x 开始的 k 步随机漫步的分布有关，如果对 k 取极限 $k \rightarrow \infty$ ，则随机漫步的分布会收敛到 $P_{lim}(\rightarrow y)$ 。详细论证过程可见原文。这说明，结果与随机漫步的的起始顶点 x 没有关系，通过这种方法来得到 x 的邻域信息是不适用的。

另一种说法是，一个 k 层的图卷积网络等同于一个 k 阶的多项式过滤器，其中的系数是预先确定的SDC。这么一个过滤器与随机漫步类似，最终会收敛到一个静态向量，从而导致过度平滑。

实践部分，论文提出JK-Net，通过Layer aggregation来让顶点最后的表示自适应地聚合不同层的信息，局部还是全部，让模型自己来学习：

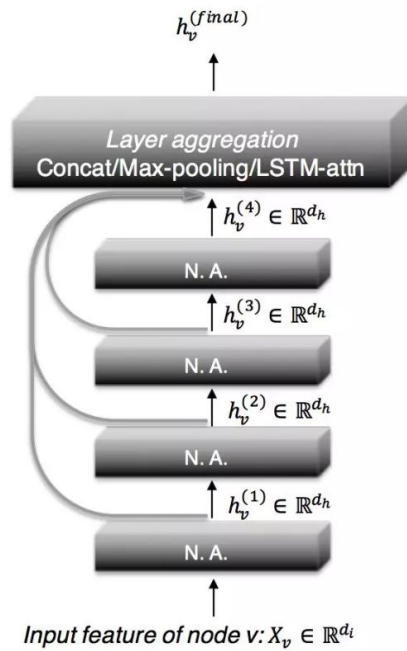


Figure 4. A 4-layer Jumping Knowledge Network (JK-Net). N.A. stands for neighborhood aggregation.

论文的重点在于最后的Layer aggregation层，可选的三种操作为：Concat、Max-pooling以及LSTM-attn。

1. Concat

将各层的表示直接拼接在一起，送入Linear层。对于小数据集及结构单一的图这种聚合方式会更好，因为它们不需要顶点在聚合邻域的顶点信息时具有什么自适应性。

2. Max-pooling

选取各层的表示中包含信息量最多的作为顶点的最终表示，在多层结构中，低层聚合更多局部信息，而高层会聚合更多全局信息，因此对于核心区域内的顶点可能会选取高层表示而边缘顶点选取低层表示。

3. LSTM-attention

对于各层的表示，attention机制通过计算一个系数 $s_v^{(l)}$ 来表示各层表示的重要性，其中 $\sum_l s_v^{(l)} = 1$ ，顶点最终的表示就是各层表示的一个加权和： $\sum_l s_v^{(l)} \cdot h_v^{(l)}$ 。

$s_v^{(l)}$ 的计算：将 k 层网络各层的表示 $h_v^{(1)}, \dots, h_v^{(k)}$ 输入一个双向LSTM中，同时生成各层 l 的前向LSTM与反向LSTM的隐式特征，分别表示为 $f_v^{(l)}$ 、 $b_v^{(l)}$ ，拼接后将 $|f_v^{(l)}||b_v^{(l)}$ 送入一个Linear层，将Linear层的结果进行Softmax归一化操作就得到了系数 s_v^l 。

数据集

Citeseer、Cora、Reddit、PPI