

从图(Graph)到图卷积(Graph Convolution): 漫谈图神经网络模型 (三)

本文属于图神经网络的系列文章, 文章目录如下:

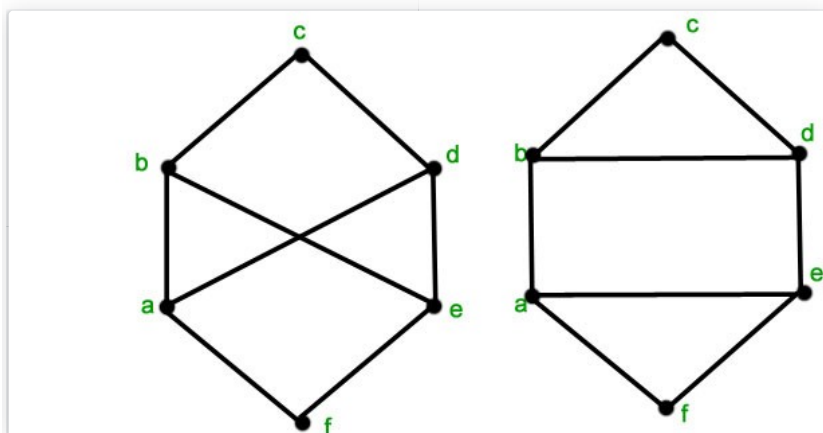
- [从图\(Graph\)到图卷积\(Graph Convolution\): 漫谈图神经网络模型 \(一\)](#)
- [从图\(Graph\)到图卷积\(Graph Convolution\): 漫谈图神经网络模型 \(二\)](#)
- [从图\(Graph\)到图卷积\(Graph Convolution\): 漫谈图神经网络模型 \(三\)](#)

恭喜你看到了本系列的第三篇! 前面两篇博客分别介绍了基于循环的图神经网络和基于卷积的图神经网络, 那么在本篇中, 我们则主要关注在得到了各个结点的表示后, 如何生成整个图的表示。其实之前我们也举了一些例子, 比如最朴素的方法, 例如图上所有结点的表示取个均值, 即可得到图的表示。那有没有更好的方法呢, 它们各自的优点和缺点又是什么呢, 本篇主要对上面这两个问题做一点探讨。篇幅不多, 理论也不艰深, 请读者放心地看。

图读出操作(ReadOut)

图读出操作, 顾名思义, 就是用来生成图表示的。它的别名有图粗化(翻译捉急, Graph Coarsening)/图池化(Graph Pooling)。对于这种操作而言, 它的核心要义在于: 操作本身要对结点顺序不敏感。

这是为什么呢? 这就不得不提到图本身的一些性质了。我们都知道, 在欧氏空间中, 如果一张图片旋转了, 那么形成的新图片就不再是原来那张图片了; 但在非欧式空间的图上, 如果一个图旋转一下, 例如对它的结点重新编号, 这样形成的图与原先的图其实是一个。这就是典型的图重构(Graph Isomorphism)问题。比如下面左右两个图, 其实是等价的:



为了使得同构图的表示能够保持一致, 图读出的操作就需要对结点顺序不敏感。在数学上, 能表达这种操作的函数也被称为对称函数。

那么我们一般如何实现图读出操作呢? 笔者接下来主要介绍两种方法: 基于统计的方法 与 基于学习的方法。

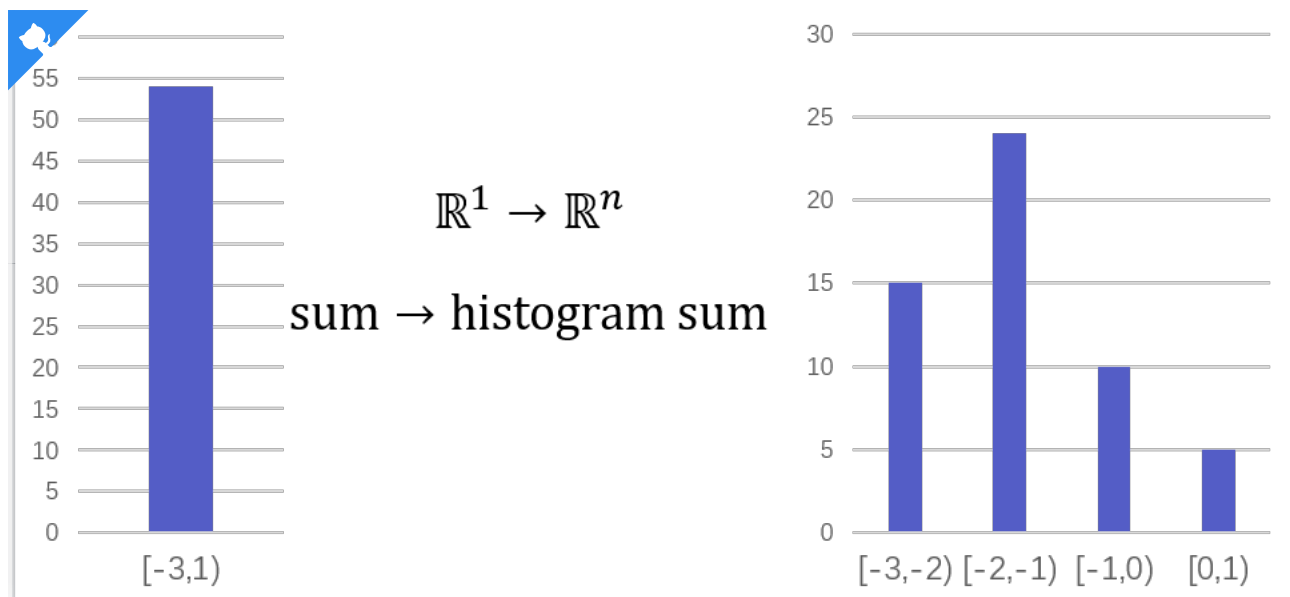
基于统计的方法(Statistics Category)

基于统计的方法应该是最常见的, 比如说我们在求各种抽象表示所使用的 平均(mean), 求和(sum), 取最大(max) 等操作。这些方法简单有效, 又不会带来额外的模型参数。但同时我们必须承认, 这些方法的信息损失太大。假设一个图里有 1000 个结点, 每个结点的表示是 100 维; 整张图本可表达 $1000 * 100$ 的特征, 这些简单的统计函数却直接将信息量直接压缩到了 100 维。尤其是, 在这个过程中, 每一维上数据的分布特性被完全抹除。

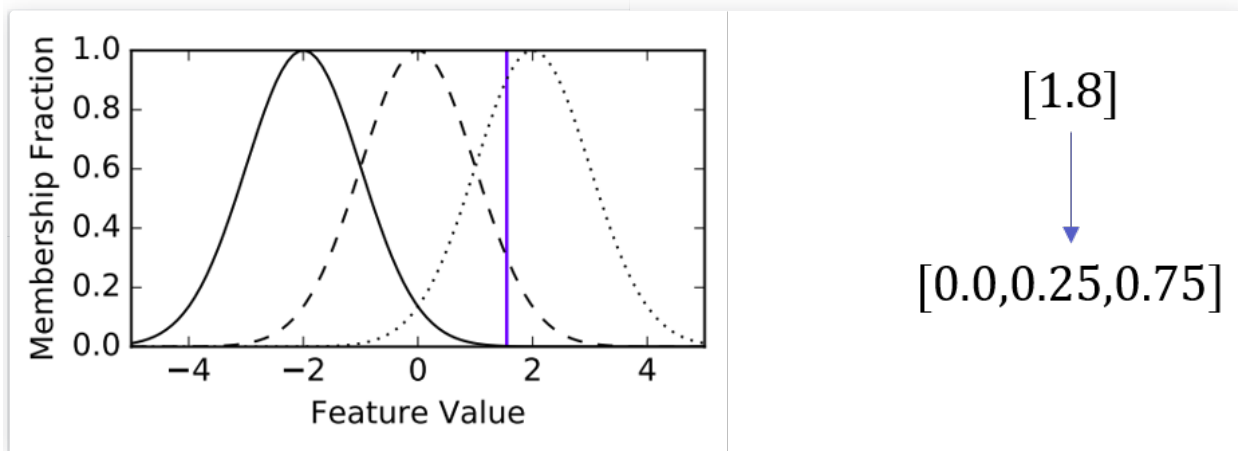
考虑到这一点, 文献[1]的作者就提出要用类似直方图的方法来对每维数据分布进行建模。具体而言, 请读者先通过下面的对比图来直观感受一下直方图是如何巧妙平衡数据信息的压缩与增强的。假设我们有 100 个介于 $[-3, 1]$ 的数字, 如果我们直接将它们求和, 如左图所示, 我们完全看不出这 100 个数据的分布; 而如果我们将 $[-3, 1]$ 等分成 4 个区域, 比如说就是 $[-3, -2)$, $[-2, -1)$, $[-1, 0)$ 和 $[0, 1]$ 。我们分开统计各个区域的和, 可以发现一点原数据的分布特征, 就如下右侧子图所示:

sum

sum



如果来实现上面这个直方图的做法, 该如何做呢? 其实也很简单, 我们举个例子。给定3个数据点, 它们的特征向量(2维)分别是 $[-2, 1]$, $[-1, 2]$ 和 $[-1, 1]$ 。如果直接求和, 全局的特征向量是 $[-2+(-1)+(-1), 1+2+1]$ 即 $[-4, 4]$ 。如果采取上述直方图的方式, 则可能会得到一个这样的全局特征向量 $[-2, -1+(-1), 1+1, 2]$ (第1,2维代表从原先的第1维统计的直方图, 对应的区域为 $[-2, 1)$, $[1, 2)$, 第3,4维的含义类似)。但在实践中, 文献[1]没有直接利用这种方法, 而是采用 **高斯函数** 来实现名为 **模糊直方图(Fuzzy Histogram)** 的操作。



模糊直方图的原理也很简单: 预先定义几个特征值区域的边界点为各个高斯分布的均值, 并预设好方差。对任一特征值, 根据其其与各个高斯分布交点的纵坐标作为其落入该区域的数值, 然后将所有数值归一化, 就得到了该特征值分布在各个区间的比例。举个例子, 图上的 $[1.8]$ 与三个高斯分布的交点分别在0.0, 0.3, 0.9处, 归一化一下, 即可知该特征值最终应该用一个3维向量 $[0.0, 0.25, 0.75]$ 来表示。

基于学习的方法(Learning Category)

基于统计的方法的一个坏处大概是它没办法参数化, 间接地难以表示结点到图向量的这个“复杂”过程。基于学习的方法就是希望用神经网络来拟合这个过程。

采样加全连接(Sample And FC)

最简单又最直接的做法, 大概就是取固定数量的结点, 通过一个 **全连接层(Fully Connected Layer)** 得到图的表示。这里不论是随机采样也好, 还是根据某些规则采样, 都需要得到确定数量的结点, 如果不够就做填充。公式也很简单直接 (\mathbf{H}^L 指的是将采样到的结点表示拼接在一起):

$$\mathbf{h}_G = FC(\mathbf{H}^L)$$

这种方法的一个问题在于很难适用于规模差距很大的图。比如说训练时见过的图只有几百个结点, 但测试的图可能上千万个结点。

泛化。

全局结点(Global Node)

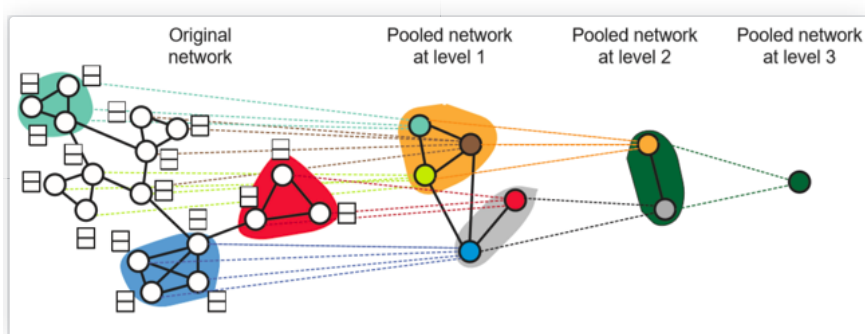
这一种做法的动机也很简单, 考虑到图同构问题和基于统计的方法, 从结点的表示生成最终的图表示主要有两个难点:

1. 很多情况下我们很难找到一个合适的根结点 (笔者觉得图的根结点一般都是根据各个领域的领域知识来确定的, 比如我们在第一篇博客中讲到的化合物分类问题)
2. 如果直接用基于统计的方法对各个结点一视同仁, 无法区别对待 (比如某些重要的结点信息更多, 就应该表达得更多)。

那我直接引入一个全局结点代表这张图的根结点, 把它跟图中的每个结点通过一种特殊的边连接, 最终拿这个结点的表示作为图的表示, 岂不妙哉。

可微池化(Differentiable Pooling)

上面两种方法都比较简单, 不会层次化地去获得图的表示。所以在论文[2]中, 作者提出了一种层次化的图表示, 而这则依赖于他们所提出的可微池化(Differentiable Pooling, DiffPool)技术。简单来讲, 它不希望各个结点一次性得到图的表示, 而是希望通过一个逐渐压缩信息的过程, 来得到最终图的表示, 如下图所示:



相比于一般先通过GCN得到所有结点表示(\mathbf{H}^l), 再通过方法汇总得到图的最终表示的方法, DiffPool则同时完成了两个任务: 结点聚类(Soft Clustering)与 结点表示(Node Representation)。这两个任务是由两个不共享参数的GCN模块分别完成的, 下文用 SC 和 NR 分别表示这两个模块。NR 模块与传统的GCN一样, 输入是各结点的隐藏状态, 通过图上的传播, 输出是传播后各个结点的表示。SC 模块则不同, 它的输入虽然也是各结点的隐藏表示, 但其输出的是各结点属于不同聚类簇的概率(注意这里每一层聚类簇的数目是预先定义的)。上图中最左侧每个结点右上方的表格即代表这个。举个例子, 假设本层子图有6个结点, 将各个结点输出的簇分类概率堆叠在一起, 即可得到矩阵 \mathbf{S}^l , 如下图所示(蓝色, 橙色和绿色分别代表三个聚类簇。在实际中, 聚类矩阵不是离散变量, 而是连续变量。):

	0	1	0
	1	0	0
	0	0	1
	1	0	0
	0	1	0
	0	1	0

$$\mathbf{S}^l \in \mathbb{R}^{6 \times 3}$$

以 \mathbf{A}^l 表示第 l 层子图结点的邻接关系, \mathbf{A}^0 即是图的邻接矩阵, N_l 表示第 l 层子图结点的个数, \mathbf{H}^l 表示第 l 层子图各个结点的隐状态矩阵, DiffPool通过如下公式得到新子图中各个结点的表示:

$$\mathbf{S}^l = \text{SC}(\mathbf{A}^l, \mathbf{H}^l), \text{ 其中 } \mathbf{S}^l \in \mathbb{R}^{(N_l \times N_{l+1})}$$



$$\tilde{\mathbf{H}}^l = NR(\mathbf{A}^l, \mathbf{H}^l)$$

$$\mathbf{H}^{l+1} = (\mathbf{S}^l)^T \tilde{\mathbf{H}}^l$$

除了各个结点的表示外, 还有一个很重要的事情是生成新子图 \mathbf{A}^{l+1} 的邻接关系:

$$\mathbf{A}^{l+1} = (\mathbf{S}^l)^T \mathbf{A}^l \mathbf{S}^l$$

其他方法

还有一些其他方法, 就如前文所提到的 PATCHY-SAN, 通过基于规则的方法对图进行排序, 将图排成序列, 然后使用类似于CNN做文本分类的方法, 使用1-D Pooling都得到图表示。

参考文献

[1]. Molecular graph convolutions moving beyond fingerprints, <https://arxiv.org/abs/1603.00856>

[2]. Hierarchical Graph Representation Learning with Differentiable Pooling, <https://arxiv.org/abs/1806.08804>

作者: SivilTaram

出处: https://www.cnblogs.com/SivilTaram/p/graph_neural_network_3.html

版权: 本作品采用「署名-非商业性使用-相同方式共享 4.0 国际」许可协议进行许可。

<http://cnblogs.com/SivilTaram>

👍 3

💬 0

« 上一篇: [从图\(Graph\)到图卷积\(Graph Convolution\): 漫谈图神经网络模型 \(二\)](#)

» 下一篇: [\[FY20 创新人才班 ASE\] 第 1 次作业成绩](#)

posted @ 2019-09-07 17:43 SivilTaram 阅读(21182) 评论(21) 编辑 收藏

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

园子动态:

- 致园友们的一封信: 都是我们的错
- 数据库实例 CPU 100% 引发全站故障
- 发起一个开源项目: 博客引擎 fluss

最新新闻:

- 京喜吉祥物牛方亮相: 省钱高手
 - 特斯拉盈利 全靠比特币?
 - 台积电技术路线图: 2纳米3纳米工艺将按时推出
 - 谷歌母公司Q1营收553亿美元同比增34% 净利179亿
 - 美股周二: 特斯拉利润创新高, 但股价仍跌4.5%
- » 更多新闻...

