

# 关于图神经网络入门研究报告

李大双

## 1. Introduction

图形式的信息广泛存在于现实生活的各个场景。社交网络，蛋白质，分子结构，交通网络都是图的应用。因此对图的研究很有价值。但是对图的处理往往需要大量的计算和空间资源开销。图嵌入(graph embedding)提供了更加有效的方法分析和处理图问题。嵌入(embedding)就是在图的信息保留的情况下，将高维度的信息转换映射到低维度中表示。因为将图用低维度的向量表示，所计算更加高效。

## 2. Graph embedding

Graph embedding 的方法有很多种，这里不做一一介绍，只对几种介绍。

### 2.1 Deepwalk

在介绍 deepwalk[1]之前先介绍一下 Word2vec。Word2vec 是 nlp 中的概念。是将单词向量化。根据语料库中单词之间的关系确定每个单词的 embedding。常见的 word2vec 的方法有两种 cbow 和 skip-gram[2]。skip-gram 是根据中心词预测上下文，cbow 与之相反。skip-gram 可以看作是一个普通的 softmax 回归问题。首先将单词 one-hot 化，然后将 one-hot 的向量喂入 softmax 模型训练，得到的网络模型中的前面输入层和紧随其后的权重(原文中把这当作输入层，如图 2.1 中的 input 层)可以看作是单词的 embedding。这样一来就得到了单词的 embedding。但是由于单词数量过多，正负样本不均匀等问题，计算量非常庞大。所以通常采用 skip-gram+负采样的方法。

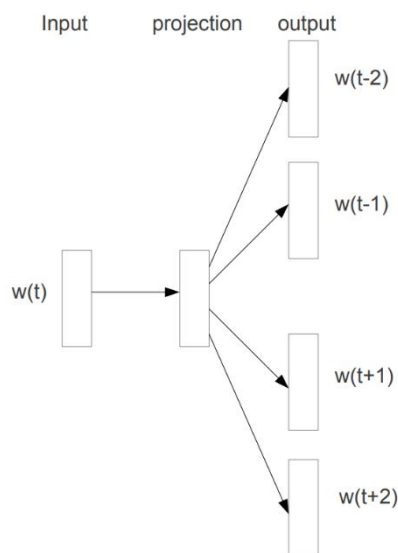


图 2.1 skip-gram 模型图

Word2vec 是根据语料库中的句子中词语之间的上下文关系进行 embedding 的。而图的结构不是序列，所以需要图进行随机游走，得到序列然后进行 embedding。Deepwalk 的方法本质就是将 word2vec 的方法用到 graph 随机游走的得到的序列上，从而得到 node 的 embedding。这里所使用的随机游走可以理解为回头看的图的深度优先遍历，可以得到一个序列，进而可以用 word2vec 进行 embedding。

### 2.2 Node2vec

Deepwalk 是一种深度遍历，注重的是网络的整体结构。GraRep 可以理解作为一种广度遍历，注重的是局部信息。为了权衡深度遍历和广度遍历的优缺点，node2vec[3]设计了一种灵活的邻域采样策略，能够在 BFS 和 DFS 之间进行平滑转换。文中通过使用一个灵活的偏置项  $\alpha$  随机游走程序来实现这一点。该程序可以以 BFS 和 DFS 的方式探索邻近区域。

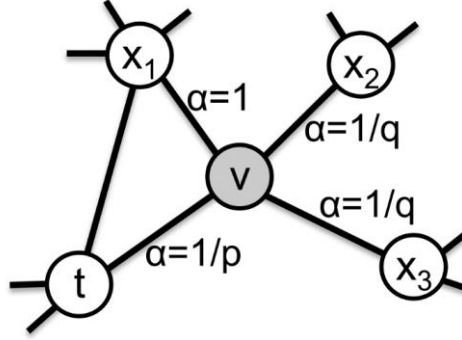


图 2.2 随机游走图

node2vec 的最具特色的部分就是随机游走。整个 node2vec 可以理解为基础 skip-gram 的特定的随机游走的方法。通过增加偏置项  $\alpha$  来控制随机游走的方向。如图 2.3 所示  $t$  表示前一个结点， $x$  表示下一个节点， $v$  表示当前节点。公式 2.1 中， $d_{tx}$  表示前一个节点与下一个节点的距离（取值范围为 0, 1, 2）。当  $d_{tx}$  等于 1 时，也就是上一个节点与下一个节点相连时， $\alpha$  等于 1。当  $d_{tx}$  等于 0 时，也就是上一个节点是  $\alpha=1/p$ 。当  $d_{tx}$  等于 2 时，也就是下一个节点与上一个节点不相连时， $\alpha$  等于  $1/q$ 。

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases} \quad (2.1)$$

在 node2vec 中通过控制  $p$  和  $q$  值来控制随机游走的趋向，当  $p$  很小的时候，可能会回退。当  $q < 1$  时，随机游走更倾向于更远的节点，也就是图的深度遍历倾向于向外探索。当  $q > 1$  时，随机游走更倾向于距离更近的节点，也就倾向于广度遍历。特别的，如果  $p=1, q=1$ ，那么此时的游走就是普通的随机游走。

### 2.3 Struc2vec

Struc2vec[4]认为 deepwalk 和 node2vec 考虑的是上下文节点，考虑的邻近性而没有考虑结构性。如图 2.3 所示， $u, v$  节点具有相似的节点结构性但是由于不相邻，在 deepwalk and node2vec 无法得到很好的效果。

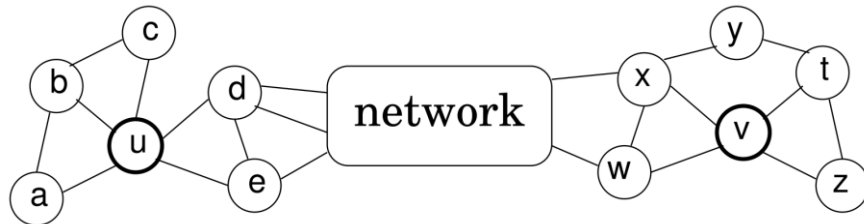


图 2.3 结构相似性图

如果两个节点的度相同，那么这两个节点结构相似；如果这两个节点的邻居度也相

同，那么这两个节点的结构相似性比前者更高。Struc2vec 基于这一核心思想，考虑的是节点的结构相似性。

$$f_k(u, v) = f_{k-1}(u, v) + g(s(R_k(u)), s(R_k(v))) \quad (2.2)$$

Struc2vec 中定义了递归求相似距离公式 2.2 来衡量节点的结构相似性。其中  $f_k(u, v)$  就是距离  $u, v$  相距为  $k$  的那些节点之间的结构距离。这是一个递归定义， $f_{k-1}(u, v)$  表示考虑  $k-1$  跳邻域时的距离，再加上只考虑  $k$  跳邻居的距离，就形成了  $k$  跳邻域的距离了，初始值  $f_1 = 0$ 。在底层，两个节点相似性的判定只依赖十分简单的信息，比如度；而在顶层，两个节点的相似性的判定则要依赖于整个网络的信息。

### 3. GCN

#### 3.1 Semi-Supervised Classification with Graph Convolutional Networks

Kipf 的这篇[5]堪称 gcn 领域的经典之作，由于本节内容在前面的报告中详细做过介绍，因此在这里不做具体介绍。文章中通过拉普拉斯进行对图 embedding。并定义了图上的傅里叶变换。通过傅里叶变换变换到谱域中的点乘来求卷积。

$$H^{(l+1)} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (3.1)$$

文章中给出了每个节点计算卷积的公式(2.3)，公式 2.3 中  $H$  是节点的属性特征， $A$  是加对角线的邻接矩阵， $D$  可以理解为度。 $W$  是可训练的权重矩阵。

#### 3.2 GraphSAGE

与 kipf 的考虑整体结构信息 GraphSAGE[6]考虑的是局部结构信息。如图 3.1 所示 GraphSAGE 学习了一种函数，通过对节点的邻域中的特征进行采样和聚合来生成嵌入，而不是为每个节点训练单个嵌入。

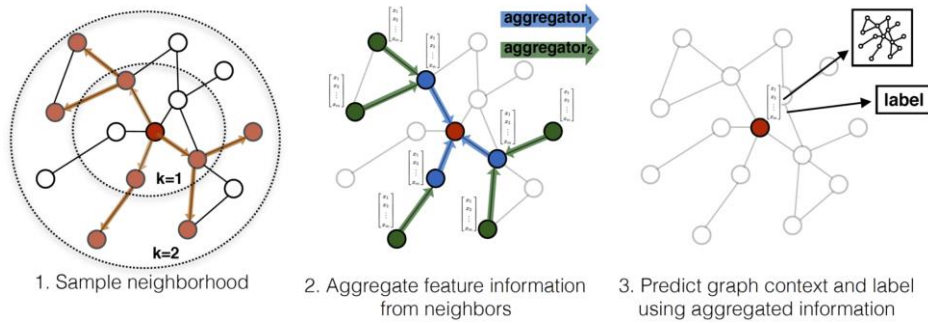


图 3.1 GraphSAGE 抽样聚合示意图

关于聚和函数，论文中提出了三种：

- ① mean aggregator 取相邻节点属性向量的同一维度的平均值

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W} \cdot \text{MEAN}(\{\mathbf{h}_u^{k-1}\} \cup \{\mathbf{h}_v^{k-1}\}, \forall u \in \mathcal{N}(v)))$$

- ② LSTM aggregator 通过随机游走得到序列。

- ③ Pooling aggregator 通过 pooling 的方法进行 aggregate，maxpooling 公式如下：

$$\text{AGGREGATE}_k^{\text{pool}} = \max(\{\sigma(\mathbf{W}_{\text{pool}} \mathbf{h}_{u_i}^k + \mathbf{b}), \forall u_i \in \mathcal{N}(v)\})$$

### 3.3 JK-NET

在对图做卷积是有个奇怪的现象，就是随着迭代层数增加 embedding 的区分效果反而下降。

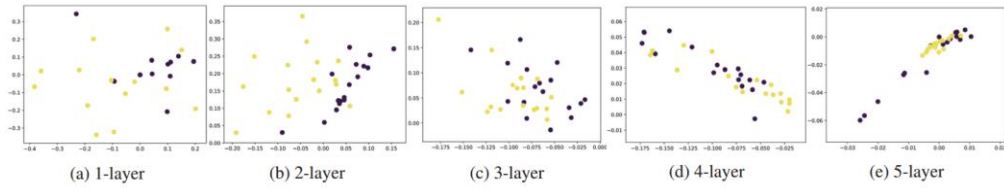


图 3.2 嵌入向量随层数增加图

图 3.2 是 18 年 AAAI 的 Deeper insights into GCN[7] 中 Zachary's karate club network 的节点 embedding 图。可以看出两层是效果最好，但是随着层数增加区分效果反而下降。GCN 模型实际上可以理解为拉普拉斯平滑。当层数增加时，GCN 就会变得过平滑。为了提高模型效果，避免过平滑的现象。大致有两种解决方法，一种是融合各阶的表达，18 年 ICML 的 JK-Net[8], 19 年 ICML 的 MixHop, 另一种是强调低阶的表达 19 年 ICLR 的 PPNP 等。本文只对 jk-net 做介绍。

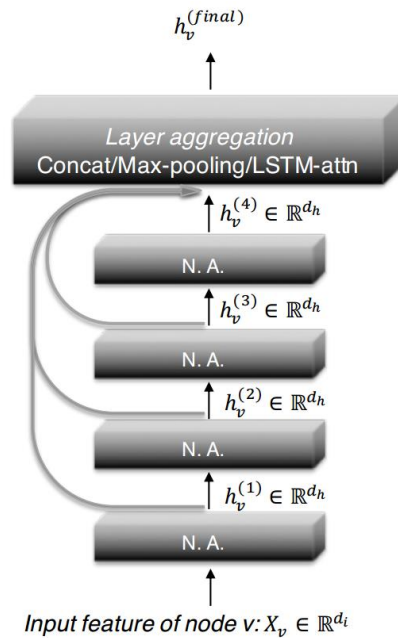


图 3.3 四层 jk-net 示意图

如图 3.3 所示 jk-net 利用各阶的表达通过融合函数的得到最终的表达。文章中选用的融合函数有 concatenate, maxpooling, lstm-attention. 最终达到了很好的效果。

## 4. Take-home-message

在 gcn 之前对图的研究大多都是利用学习 nlp 中的 word2vec 方法，将图节点转换成序列进行 embedding。Gcn 是通过普方法和空方法进行研究。对 gcn 有很多改进，比如加上 attention 机制等。

目前我们研究的图都是同质图(Homogeneous Graph)，也就是每条边都是同样的类型。但是现实情况存在很多异质图(Heterogeneous Graph)的使用场景，每条边类型不一定相同。对于异质图的研究也已经有了很多工作。今年 2019www 中纪厚业等的 HAN 就是对异质图进行研究。

## 5. Reference

- [1] Bryan Perozzi, Rami Al-Rfou and Steven Skiena. DeepWalk: Online Learning of Social Representations.
- [2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean Steven Skiena. Distributed Representations of Words and Phrases and their Compositionality.
- [3] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks
- [4] Leonardo F. R. Ribeiro, Pedro H. P. Savarese, Daniel R. Figueiredo. struc2vec: Learning Node Representations from Structural Identity.
- [5] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks.
- [6] William L. Hamilton, Rex Ying and Jure Leskovec. Inductive Representation Learning on Large Graphs.
- [7] Qimai Li, Zhichao Han, Xiao-Ming Wu. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning.
- [8] Keyulu Xu , Chengtao Li , Yonglong Tian, Tomohiro Sonobe , Ken-ichi Kawarabayashi , Stefanie Jegelka. Representation Learning on Graphs with Jumping Knowledge Networks