

# 图神经网络研究综述

侯磊<sup>1</sup> 刘金环<sup>1</sup> 于旭<sup>2</sup> 杜军威<sup>1</sup>

1 青岛科技大学数据科学学院 山东 青岛 266061

2 中国石油大学(华东)计算机科学与技术学院 山东 青岛 266580

(houlei\_fly@163.com)

**摘要** 随着人工智能的快速发展,深度学习已经在图像、文本和语音等可在欧氏空间表示的数据中取得了巨大成功,但却一直无法很好地应用于非欧氏空间。近年来,图神经网络在非欧几里得空间中展现出了强大的表示学习能力,并广泛应用于推荐系统、自然语言处理以及机器视觉等众多领域。图神经网络模型基于信息的传播机制,具体地,图中的目标节点通过聚合邻居节点的信息来更新自身的嵌入表示。利用图神经网络,可将众多现实问题(如社交网络、知识图谱和药物化学成分等)抽象成图网络,借助图中的连接边,对不同节点之间的依赖关系进行合理建模。鉴于此,对图神经网络进行了系统综述,首先介绍了图结构数据方面的基础知识,然后对图游走算法和不同类型的图神经网络模型进行了系统梳理。进一步地,详细阐述了当前图神经网络的通用框架和应用领域,最后对图神经网络的未来进行了总结与展望。

**关键词:** 图结构数据;图游走算法;图卷积神经网络;图注意力网络;图残差网络;图递归网络

**中图分类号** TP183;TP389.1

## Review of Graph Neural Networks

HOU Lei<sup>1</sup>, LIU Jinhuan<sup>1</sup>, YU Xu<sup>2</sup> and DU Junwei<sup>1</sup>

1 School of Data Science, Qingdao University of Science and Technology, Qingdao, Shandong 266061, China

2 School of Computer Science and Technology, China University of Petroleum (East China), Qingdao, Shandong 266580, China

**Abstract** With the rapid development of artificial intelligence, deep learning has achieved great success in data that can be represented in Euclidean spaces, such as images, text, and speech. However, it has been difficult to apply deep learning to non-Euclidean spaces. In recent years, with the emergence of graph neural networks, it has demonstrated powerful representation learning abilities in non-Euclidean spaces and has been widely applied in various fields such as recommendation systems, natural language processing, and computer vision. The graph neural network model is based on the mechanism of information propagation. Specifically, the target node in the graph updates its embedding representation by aggregating the information of neighboring nodes. With graph neural networks, many real-world problems (such as social networks, knowledge graphs, and drug chemical compositions) can be abstracted into graph networks and the dependence relationships between different nodes can be modeled reasonably using the connecting edges in the graph. Therefore, this paper systematically reviews graph neural networks, introduces the basic knowledge of graph-structured data, and systematically reviews graph walk algorithms and different types of graph neural network models. Furthermore, it also details the current general framework and application areas of graph neural networks, and concludes with a summary and outlook on future research in graph neural networks.

**Keywords** Graph-structure data, Graph walk algorithm, Graph convolutional networks, Graph attention networks, Graph residual networks, Graph recurrent networks

## 1 引言

近几年,深度学习得到快速发展,现已成为人工智能研究中的主流方法之一。深度学习的基本思想是通过堆叠多层的神经网络,且经过线性变换和非线性激活两大步骤,来获得相对准确的特征表示。典型的神经网络有卷积神经网络

(CNN)<sup>[1]</sup>、循环神经网络(RNN)<sup>[2]</sup>和生成对抗神经网络(GAN)<sup>[3]</sup>。它们已经在图像、文本和语音等欧氏数据上取得了巨大成功。之所以能取得如此巨大的成功,其一在于计算资源(如GPU)的快速发展使得计算机在处理大规模数据集时有更强的计算能力和更大的存储空间;其二在于深度学习能在欧氏空间表示的数据中提取有效的特征表示。

到稿日期:2023-04-03 返修日期:2023-09-30

基金项目:国家自然科学基金(62202253,62172249);山东省自然科学基金(ZR2021QF074,ZR2021MF092)

This work was supported by the National Natural Science Foundation of China(62202253,62172249) and National Natural Science Foundation of Shandong Province, China(ZR2021QF074,ZR2021MF092).

通信作者:刘金环(liujinhuan\_sdu@gmail.com)

例如 CNN 的局部性和平移不变性,天然适合处理图像等规则的结构数据。然而,在纷繁复杂的现实世界中,许多问题(如社交网络和知识图谱等)可使用图结构数据进行建模表示。图数据结构的优势在于可利用图中固有的拓扑结构,编码节点与节点之间复杂的成对关系,从而获得更多的信息。

图神经网络(Graph Neural Networks,GNN)是在图域上进行操作的深度学习方法,由于该网络具有强大的表征能力,现已成为一种应用广泛的图分析方法。GNN 最早是由 Gori 等<sup>[4]</sup>于 2005 年提出的,并经过 Scarelli 等<sup>[5]</sup>的改进得到了进一步的发展和完善。现如今,图神经网络已经成为神经网络中的一个重要分支。最初对图神经网络的研究,借鉴了在欧几里得空间中处理数据的思想,试图固定图中节点的数量和节点的排列顺序,从而将其转换为规则的数据结构,然后采用卷积等操作完成。Niepert 等<sup>[6]</sup>提出的 PATCHY-SAN、Gao 等<sup>[7]</sup>提出的 LGCN,以及 Atwood 等<sup>[8]</sup>提出的 DCNN 均是按照这一思想展开的。随着 GNN 的快速发展,产生了基于游走算法的图神经网络,如 Deep Walk<sup>[9]</sup>,Node2vec<sup>[10]</sup>等。这类方法的主要思想是通过随机游走生成预定长度的节点序列,然后采用自然语言处理中的表示方法(如 Skip-gram<sup>[11]</sup>)对节点进行嵌入表示。在图神经网络发展的初期,此类方法发挥了一定的作用,但游走算法的性能严重依赖于节点序列长度。当序列长度越大,训练模型所需的迭代次数就越多,算法的运行时间也越长。针对游走类算法的不足,不同种类的图神经网络模型应运而生。如今,它们已经成为学术界和工业界主流的图表示学习方法。图神经网络的出现,弥补了传统神经网络无法有效处理非欧氏数据的空白,对推动人工智能向前发展具有里程碑意义。研究发现,GNN 现已成为处理大量应用领域任务的优秀学习框架之一。例如 GCMC<sup>[12]</sup>,GAT<sup>[13]</sup>和 GraphSAGE<sup>[14]</sup>等基于图结构数据的神经网络模型已经在一些众所周知的领域(如社交网络、生物信息学)中取得了优异成绩。当然,GNN 的成功应用远远不限于此,其还渗透到了推荐系统<sup>[15-19]</sup>、自然语言处理<sup>[20-23]</sup>、计算机视觉<sup>[24-28]</sup>、药物发现<sup>[29]</sup>和异常检测<sup>[30-31]</sup>等多个领域。

随着图神经网络的不断发展,也衍生出了很多前沿研究,这些研究可分为节点级别、边级别和图级别三大类。节点级别方面主要包括节点分类任务,其目标在于将网络中的不同节点分为几个预定义的类别。例如在社交网络中,可预测每一个用户的政治偏见。在互联网中,可将网页分成不同的语义类别。边级别方面主要包括链路预测任务,它可预测网络中两个节点之间的链路问题。例如社交网络中的朋友推荐以及生物网络中的蛋白质相互作用预测都有很好的应用。在图级别方面,主要包括图分类和图生成等任务。图分类的目标在于预测一组图的类别标签,如化学分子的性质;而图生成的目标在于捕获图形数据中的复杂依赖关系,以生成真实的图形。

目前已经有很多学者对 GNN 的相关技术进行了综述,但涵盖的知识面相对狭窄。为了对图神经网络进行更加系统且全面的综述,本文第 3 章对图神经网络发展过程中出现的不同方法进行了系统梳理与总结,第 4 章详细介绍了图神经

网络的通用框架,第 5 章介绍了图神经网络的一些主流应用领域,第 6 章对图神经网络未来的发展进行了总结和展望。

## 2 背景知识

### 2.1 图的定义

图是由顶点的有穷非空集合和顶点之间的边集所构成的一种数据结构,可表示为:

$$G=(V,E) \quad (1)$$

其中, $V$  表示图中所有顶点构成的集合, $E$  表示图中所有边构成的集合。图 1 展示了简单图的示例。

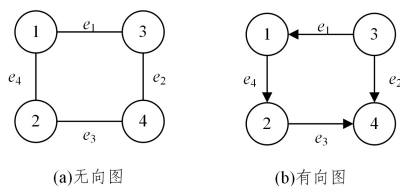


图 1 简单图示例

Fig. 1 Simple graph example

### 2.2 图的定义

GNN 在对现实问题进行建模时,根据问题的不同,可采用不同类型的图。除了常见的无向图以外,还有以下 5 种常用的无向图变体形式。

1)有向图:作为无向图最简单的一种变体,有向图指连接图中节点之间的边具有方向性的图。在实际问题的处理中,有向图可能比无向图包含更多额外的信息。例如知识图谱中从头实体指向尾实体的边代表了信息流的传播方向。具体地,在基于有向图的 GNN 研究中,Cui 等<sup>[32]</sup>提出的 NGNN 模型在图神经网络中通过构建时尚单品之间的有向图来对服装进行兼容性建模。Tong 等<sup>[33]</sup>提出的有向图卷积网络 DGCN 通过一阶和二阶近似性将基于谱域的图卷积网络扩展到有向图,不仅保留了有向图的连接特性,而且扩大了卷积操作的接收域。Kampffmeyer 等<sup>[34]</sup>提出的 DGP 模型通过额外的连接来利用知识图谱的分层图结构,从而改善了信息在有向图中的传播。

2)带权图:图中的边具有权重信息的图。边上权重的大小可定量描述两个节点之间关系的强弱。例如在推荐系统中常用带权图来刻画用户对项目的评分情况,评分的高低反映了用户对项目的喜爱程度。除此之外,Duan 等<sup>[35]</sup>提出的 Steam-Group 方法使用动态权重有向图实现了对图数据的有效信息挖掘。

3)异构图:图中的节点和边具有多种不同类型的图。处理异构图最简单的方式是使用独热编码将每个节点的类型转换为与原始特征相关的表示向量。在异构图的研究中,Zhang 等<sup>[36]</sup>将元路径的概念引入异构图的信息传播中,从而有效地实现了节点分类等任务。

4)动态图:无向图的另外一种变体是动态图。与具有稳定结构的静态图不同,动态图的节点和边可随着时间的推移不断变化。在这方面的研究中,为了将图结构数据和时间序列数据一起建模,Li 等<sup>[37]</sup>提出了 DCRNN 模型。该模型先通过 GNN 收集空间位置信息,然后将其输入序列模型中进行

处理。而 Yan 等<sup>[38]</sup>提出的 ST-GCN 模型同时收集空间和时间信息,并且用时序信息扩展了静态图结构,因此可在拓展图上直接应用 GNN。最近, Huang 等<sup>[39]</sup>根据链接持续时间将动态网络分为几个类别,同时还为动态模型建立了一个可融入不同 GNN 模型的通用框架。

5) 边信息图: 一种在连接图中节点之间的边上可包含多

种不同附加信息的图,例如边的权重和类型。边信息图可在边上存储额外的信息,从而使得图结构数据更加易于管理。Berk 等<sup>[40]</sup>提出的 G2S 模型通过与门控图神经网络(Gated Graph Neural Networks, GGNN)<sup>[41]</sup>相结合,在处理自然语言任务时,大大提升了语义理解和机器翻译的效率。

图 2 展示了上述 5 种不同图变体的示例。

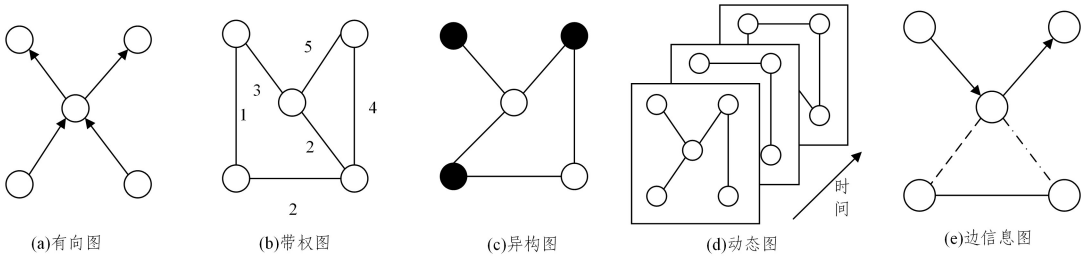


图 2 5 种不同的图变体示例

Fig. 2 Five different examples of graph variants

2.3 图的表示方法

GNN 在对现实问题进行建模时,从存储结构和代数两方面考虑。常用的图表示方法有字典表示法、邻接矩阵表示法、度矩阵表示法、拉普拉斯矩阵表示法、对称归一化拉普拉斯矩阵表示法、随机游走归一化拉普拉斯矩阵表示法以及关联矩阵表示法。

1) 字典表示法: 使用“键-值对”来表示图中节点与节点之间的连接关系。在一个完整的图中,若存在指向  $a$  节点的  $b, c, d$  节点。则将节点  $a$  作为字典的键,将  $b, c, d$  节点构成的集合作为字典的值,用来表示节点  $a$  与其余 3 个节点的关系,即  $\{a: \{b, c, d\}\}$ 。

2) 邻接矩阵表示法: 即采用数据结构中的二维矩阵来表示图中节点与节点之间的连接关系。以无权图为例,在无向图中,如果节点  $a$  与节点  $b$  有边相连,则  $e_{ab} = 1$ , 否则  $e_{ab} = 0$ 。而在有向图中,如果存在由  $b$  节点指向  $a$  节点的边,但不存在由  $a$  节点指向  $b$  节点的边,则  $e_{ba} = 1, e_{ab} = 0$ 。

3) 拉普拉斯矩阵表示法: 拉普拉斯矩阵也被称为导纳矩阵、基尔霍夫矩阵或离散拉普拉斯算子。详细数学定义如下:

$$L = D - A \tag{2}$$

其中,  $D$  表示度矩阵,  $A$  表示邻接矩阵。

4) 对称归一化拉普拉斯矩阵: 以拉普拉斯矩阵表示为基础,增加归一化操作之后,便可得到对称归一化之后的拉普拉斯矩阵。其数学定义如下:

$$L^{sym} = I_n - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{3}$$

其中,  $I_n \in R^{n \times n}$  表示单位矩阵,  $D$  表示度矩阵,  $A$  表示邻接矩阵。对称归一化拉普拉斯矩阵的具体元素定义如下:

$$L_{ij}^{sym} = \begin{cases} 1, & \text{if } i = j \text{ and } d_i \neq 0 \\ \frac{1}{-\sqrt{d_i d_j}}, & \text{if } i \text{ and } j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

5) 随机游走归一化拉普拉斯矩阵: 数学定义如下:

$$L^{rw} = D^{-1} L = I_n - D^{-1} A \tag{5}$$

其中,  $D$  表示度矩阵,  $L$  表示拉普拉斯矩阵,  $A$  表示邻接矩阵。随机游走归一化拉普拉斯矩阵中的具体元素定义如下:

$$L_{ij}^{rw} = \begin{cases} 1, & \text{if } i = j \text{ and } d(v_i) \neq 0 \\ \frac{1}{-d(v_i)}, & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

6) 关联矩阵: 在对图进行表示时,还存在一种常用且相对简单的表示方法,即关联矩阵。对一个具有  $n$  个节点、 $m$  条边的有向图来说,对应的关联矩阵  $M$  可定义为:

$$M_{ij} = \begin{cases} 1, & \text{if } \exists k, \text{ s. t. } e_j = \{v_i, v_k\} \\ -1, & \text{if } \exists k, \text{ s. t. } e_j = \{v_j, v_k\} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

而对无向图来说,对应的关联矩阵可表示为:

$$M_{ij} = \begin{cases} 1, & \text{if } \exists k, \text{ s. t. } e_j = \{v_i, v_k\} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

以图 1 中的无向图为例,表 1 列出了上述几种不同的图表示方法。

表 1 不同的图表示方法

Table 1 Different graph representation methods

| 图      | 字典表示法                                            | 邻接矩阵                                                                                             | 拉普拉斯矩阵                                                                                                   | 归一化拉普拉斯矩阵                                                                                                                                                                                | 随机游走归一化拉普拉斯矩阵                                                                                                                                                                            | 关联矩阵                                                                                             |
|--------|--------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| 图 1(a) | $\{1: [2, 3], 2: [1, 4], 3: [1, 4], 4: [2, 3]\}$ | $\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ | $\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$ | $\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & 1 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$ |

## 2.4 图表示学习

事实上,图神经网络可归纳为图表示学习问题,而图表示学习又属于在图域上进行操作的表示学习方法。因此,本节首先介绍表示学习,然后过渡到图表示学习。

### 2.4.1 表示学习

近年来,机器学习领域取得了一些不错的成绩,这不仅得益于算法本身的设计,更要归功于准确的数据特征表示。包含不正确或者大量冗余信息的数据表示可能无法发挥算法的最大性能。为了获得有效且准确的数据表示,机器学习领域中出现了一种非常重要的研究方法,即表示学习。它能够在没有专家等人工手段的干预下自动地发现隐藏在海量数据中的特征模式,进而从数据中提取足够但最少的信息用于后续的下游任务。

表示学习发展到今天,经历了从基于机器学习的传统表示学习模型到基于深度学习的现代表示学习模型的蜕变。传统的表示学习模型属于浅层次的表示方法,主要目标在于学习数据的变换,从而在构建分类器或预测器时更加容易地提取有用信息。在机器学习领域中,传统的表示学习方法主要有主成分分析(Principal Component Analysis, PCA)<sup>[42]</sup>、线性判别降维(Linear Discriminant Analysis, LDA)<sup>[43]</sup>和独立成分分析(Independent Component Analysis, ICA)<sup>[44]</sup>等方法。而基于深度学习的现代表示学习模型主要是借助神经网络以产生更加抽象且准确的向量表示。当前主流的深度学习表示模型可分为监督学习、无监督学习和迁移学习三大类。

### 2.4.2 图表示学习

图表示学习是表示学习在图上的具体应用,旨在在有效保持图结构的条件下,利用图中节点之间的连接关系,把每个节点表示映射到低维且稠密的向量空间中。

图表示学习主要有两大目标:1)在学习到的表示空间中重构原始图,它要求原始图中的两个节点如果有直接的边

相连,那么这两个节点在新的表示空间中的距离应该相对较小;2)学习到的表示空间应当支持如链接预测、推断节点标签等图推理任务。因此,要想在图表示学习的下游任务中取得良好的结果,需依赖于一个精确的节点表示。但目前的图表示学习研究仍面临着诸多挑战,具体如下:1)图数据具有高度的非线性结构。与具有清晰数据结构的图像、文本和声音不同,图网络存在多种多样的拓扑结构,且底层结构具有高度非线性。因此,如何设计出一个合适的图网络模型以便能在时间和空间上捕捉网络的非线性结构是目前面临的首要挑战。2)图结构和图属性的保留性。图表示学习的其中一个目标是进行图上的推理,而图结构和图属性的完整性是保证图推理准确性的必要前提。因此,保留图的结构和属性给研究人员提出了巨大挑战。3)图数据的稀疏性。现实世界网络图中的每个节点与其度数的关系是遵循幂律的,这就会导致整个网络十分稀疏,因而无法提供足够的链路信息以进行利用,这对图表示学习是十分不利的。

## 3 图神经网络算法

### 3.1 图游走算法

图随机游走是一种方便而且有效的网络采样方法。该方法可在保持节点之间原始关系的前提下生成节点序列,然后采用自然语言处理中处理单词序列的方法,来生成节点的嵌入表示。

#### 3.1.1 DeepWalk

DeepWalk 是由 Perozzi 等<sup>[9]</sup>于 2014 年提出的一种经典的图游走算法。该算法首次将深度学习应用于图网络分析,奠定了网络表示学习的基石。DeepWalk 的基本思想是把随机游走算法与语言建模方法结合起来,首先将游走产生的节点序列视为句子的等价物,然后借助 Skip-gram<sup>[11]</sup>算法在图网络中学习节点的嵌入表示。该模型的整体结构图如图 3 所示。

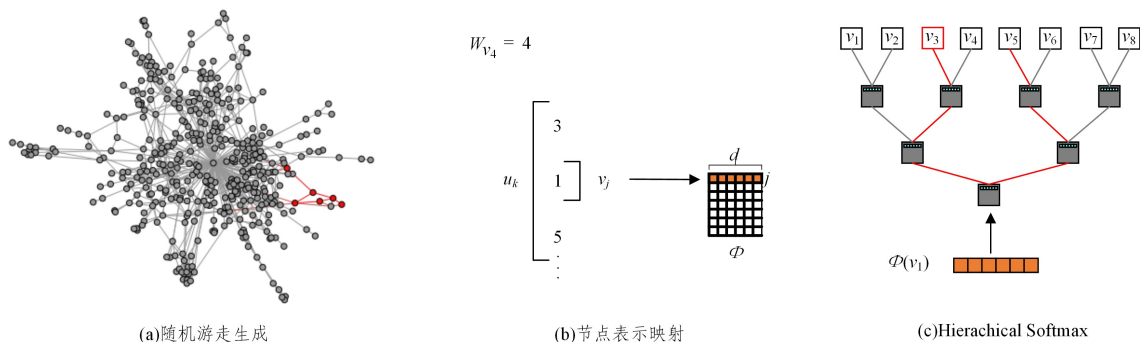


图3 DeepWalk 整体结构图  
Fig. 3 Overall structure of DeepWalk

DeepWalk 算法主要由两部分组成,其一是随机游走生成器,该部分主要是通过 RandomWalk 算法实现;其二是更新过程,该部分由 Skip-gram 算法实现。

1) RandomWalk。该算法基本思想十分简单,可总结为:首先选择一个随机顶点  $v_i$  作为一次随机游走的根节点,并将本次随机游走序列记为  $S_{v_i}$ ,之后在  $v_i$  的邻域内均匀采样邻居节点,然后再以采样到的邻居节点为依据,使用同样的方法

均匀采样其邻居节点,不断重复该过程,直到游走序列  $S_{v_i}$  的长度达到给定长度  $l$ 。考虑到图网络的局部结构具有连续性,可使用较短的随机游走序列作为在网络结构中提取信息的工具。

2) Skip-gram。作为一种经典的语言模型, Skip-gram 算法最大化了一个句子的不同单词在同一个窗口中共同出现的概率。Skip-gram 算法的基本过程如算法 1 所示。



**算法 1** Skip-gram( $\Phi, S_{v_i}, w$ )

```

1. for each  $v_j \in S_{v_i}$  do
2.   for each  $u_k \in S_{v_i} [j-w : j+w]$  do
3.      $J(\Phi) = -\log \Pr(u_k | \Phi(v_j))$ 
4.      $\Phi = \Phi - \alpha * \frac{\partial J}{\partial \Phi}$ 
5.   end for
6. end for

```

算法 1 中的第 1–2 行迭代了在给定窗口  $w$  下所有可能的随机游走组合,第 3 行使用 Hierarchical Softmax<sup>[9]</sup> 来计算每一个节点  $u_k$  的损失函数,然后使用梯度下降来更新模型中的参数值。需要特别指出的是,对每一个顶点  $v_j$ ,先将其映射为当前的表示向量  $\Phi(v_j) \in R^d$ ,  $d$  为向量空间的维度。借助 Random Walk 和 Skip-gram 算法,DeepWalk 算法的伪代码如算法 2 所示。

**算法 2** DeepWalk( $G, w, d, \gamma, t$ )

输入:图  $G(V, E)$ ,窗口大小  $w$ ,嵌入向量大小  $d$ ,随机游走初始节点  $\gamma$ ,游走长度  $t$

输出:节点表示矩阵  $\Phi \in R^{|V| \times d}$

```

1. 对取样节点进行初始化
2. 从节点集合  $V$  中构建二叉树  $T$ 
3. for  $i=0$  to  $\gamma$  do
4.    $O = \text{Shuffle}(V)$ 
5.   for each  $v_i \in O$  do
6.      $W_{v_i} = \text{RandomWalk}(G, v_i, t)$ 
7.     Skip-gram( $\Phi, W_{v_i}, w$ )
8.   end for
9. end for

```

Deepwalk 虽然可获得网络中节点的嵌入表示,但是该算法在游走过程中完全是随机的,缺乏特定的目标来指导游走序列的产生;并且 DeepWalk 无法突出网络中的局部结构和全局结构。以上两点不足在一定程度上限制了其性能。

**3.1.2 Node2Vec**

与 DeepWalk 的基本原理类似, Grover 等<sup>[10]</sup> 提出的 Node2Vec 也是一种基于网络的拓扑结构来生成节点嵌入表示的方法。鉴于 DeepWalk 的采样模式不足以捕获网络中连接模式的多样性, Grover 等设计了一种新的随机游走采样策略。通过调整预先定义的参数  $p$  和  $q$ ,可对网络进行广度优先和深度优先两种采样方式,从而弥补了 Deepwalk 无法同时捕获网络中局部结构和全局结构的不足。

具体地, Node2Vec 等在二阶随机游走过程中通过定义返回概率参数  $p$  和离开概率参数  $q$  来指导游走序列的生成。如图 4 所示,假设节点  $v$  表示游走序列现在的位置,节点  $t$  表示游走序列的前一个位置。Grover 等通过定义如下公式来决定随机游走下一步所需访问节点  $x$  的概率:

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p}, & \text{if } d_{tx} = 0 \\ 1, & \text{if } d_{tx} = 1 \\ \frac{1}{q}, & \text{if } d_{tx} = 2 \end{cases} \quad (9)$$

其中,  $d_{tx}$  表示节点  $t$  和节点  $x$  之间的最短路径。参数  $p$  表示随机游走访问前一个节点的可能性,将其设置为一个较大的值时,可在一定程度上避免对已访问过的节点进行采样,而将其设置为一个较小的值时,会使随机游走“局部”靠近初始节点。参数  $q$  允许搜索区分“向内”和“向外”的节点,当  $q > 1$  时,随机游走会偏向于靠近节点  $t$  的节点,这与广度优先搜索策略类似,此时的游走序列由一个较小的邻域节点组成;而当  $q < 1$  时,随机游走会访问距离节点  $t$  更远的节点,这与深度优先搜索策略一致。

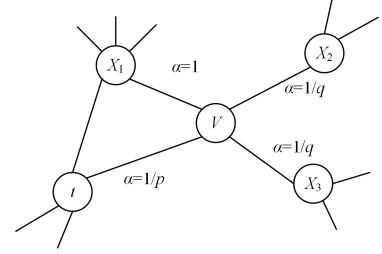


图 4 Node2Vec 随机游走过程

Fig. 4 Random walk process of Node2Vec

**3.1.3 LINE**

为了能在大规模信息网络中学习节点的嵌入表示, Tang 等<sup>[45]</sup> 提出了一种新颖的网络节点嵌入表示方法 LINE。不同于 DeepWalk 和 Node2Vec 只能采用一阶近似性来捕捉网络结构, LINE 方法同时使用一阶近似性和二阶近似性来捕获图中的局部结构和全局结构。下面结合图 5 给出一阶近似性和二阶近似性的定义。

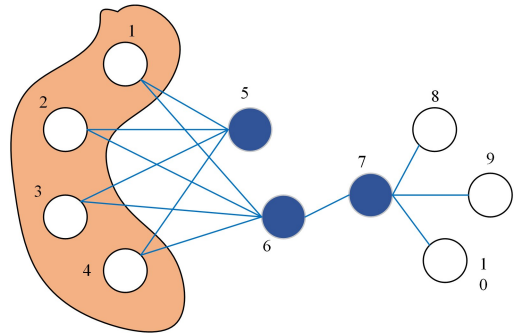


图 5 信息网络示例

Fig. 5 Information network example

**一阶近似性:**一阶近似性是网络中两个节点之间的局部成对近似。对于由边  $(u, v)$  连接的成对节点,边的权重表示节点  $u$  和节点  $v$  之间的一阶近似程度。如果节点  $u$  和节点  $v$  没有边相连,则一阶近似性为 0。例如图 5 中节点 6 和节点 7 之间的边的权重较大,因此具有较高的一阶近似性。当这两个节点的特征被映射嵌入空间时,二者应该彼此接近。需要特别注意的是,一阶近似性只能应用于无向图。

**二阶近似性:**二阶近似性表示网络中成对节点之间邻域网络结构的相似性。当图中节点具有共享的邻域结构时,二阶近似性就越大。例如在图 5 中,尽管节点 5 和 6 之间并没有直接的边相连,但这两个节点具有很多相同的邻居节点(如节点 1, 2, 3, 4),因此节点 5 和 6 具有较高的二阶近似性,这

两个节点的特征被映射到嵌入空间时,也应当彼此接近。二阶近似性的提出有效地弥补了一阶近似性的稀疏性,可更好地保持网络的全局结构。

LINE 方法的主要优势有以下两点:1)可处理任何类型的信息网络,如有向图、无向图和带权图等;2)解决了经典随机梯度下降带来的局限性,提高了推理的有效性和效率。

表 2 列出了 3 种不同游走算法的对比结果。虽然图游走算法对生成节点的嵌入表示发挥了一定的作用,但是游走算法的性能严重依赖于节点序列长度。序列长度越大,训练模型所需的迭代次数就越多,算法的运行时间也会越长。基于此,不同种类的 GNN 算法应用而生。

表 2 3 种不同游走算法的对比

Table 2 Comparison of three different walk algorithms

| 算法       | 搜索邻域策略    | 是否应用一阶近似性 | 是否应用二阶近似性 | 适用图的类型   |
|----------|-----------|-----------|-----------|----------|
| DeepWalk | 深度优先      | 否         | 是         | 无权图      |
| Node2vec | 广度优先、深度优先 | 是         | 否         | 带权图、无权图等 |
| LINE     | 广度优先      | 是         | 否         | 带权图、无权图等 |

3.2 图神经网络算法

3.2.1 图卷积神经网络

近几年,CNN 已经在图像、文本和语音等欧氏数据中取得了巨大成功。但由于传统的卷积滤波器对非欧氏结构数据的限制,CNN 并不能很好地应用于图结构数据中。基于此问题,为了将卷积的概念迁移到非欧几里得结构空间中,众多研究者经过不懈的努力,成功提出了图卷积神经网络(GCN)。具体地,GCN 可分为基于谱域的图卷积神经网络和基于空域的图卷积神经网络。接下来对 GCN 进行详细介绍与总结。

1)基于谱域的图卷积神经网络

谱域 GCN 的主要理论基础在于傅里叶变换和图信号处理,并通过对图的拉普拉斯矩阵进行特征分解,从而在图上进行卷积操作。下文将从谱图理论入手,逐渐引出基于谱域的图卷积神经网络。

(1)谱论背景知识:信号论中傅里叶变换的实质是将信号函数从时域空间转换到谱域空间。在函数连续的情况下,傅里叶变换可表示为:

$$F(\omega)=h(\omega)=\int_{-\infty}^{+\infty}f(t)e^{-j\omega t}dt \tag{10}$$

其中,函数  $h$  表示傅里叶变换, $f(t)$ 表示信号函数。由傅里叶变换,容易得出傅里叶逆变换:

$$f(t)=h^{-1}(t)=\int_{-\infty}^{+\infty}F(\omega)e^{j\omega t}d\omega \tag{11}$$

卷积定理作为傅里叶变换的一个重要性质,指出了函数卷积的傅里叶变换是函数傅里叶变换的乘积。

$$h(f * g)=h(f) \times h(g) \tag{12}$$

其中, $f$ 和 $g$ 表示两个原始信号函数, $*$ 表示卷积算子。对式(12)两端同时进行傅里叶逆变换便可得到:

$$f * g=h^{-1}(h(f) \times h(g)) \tag{13}$$

(2)图域傅里叶变换:在图域上进行傅里叶变换,与信号处理中经典的傅里叶变换类似。其中,图的拉普拉斯矩阵

特征值可表示图信号的频率,其特征向量可用作在图域上进行傅里叶变换的基向量。具体地,将图的拉普拉斯矩阵进行特征分解能够得到:

$$L=U\Lambda U^{-1} \tag{14}$$

其中, $\Lambda$ 表示拉普拉斯矩阵的特征值构成的对角矩阵。 $U=(u_1,u_2,\cdots,u_n)$ 是由特征值对应的特征向量所构成的向量矩阵,其中, $u_i \in \mathbf{R}$ 。由于 $L$ 是实对称矩阵,故 $U$ 是正交矩阵,进一步地,式(14)可改写为:

$$L=U\Lambda U^{-1}=U\Lambda U^T \tag{15}$$

在图上进行傅里叶变换依赖于拉普拉斯矩阵的特征向量。把传统傅里叶变换中的 $e^{-j\omega t}$ 用 $U^T$ 进行替换,便可得到图域上信号  $y$  的傅里叶变换及其逆变换,分别如式(16)和式(17)所示:

$$\hat{y}=U^T y \tag{16}$$

$$y=U \hat{y} \tag{17}$$

(3)谱域图卷积:利用卷积定理以及图域上的傅里叶变换和逆变换,便可得到图域上的卷积公式:

$$f * _G g=U((U^T f) \odot (U^T g)) \tag{18}$$

其中, $* _G$ 表示在图上定义的卷积算子, $\odot$ 表示哈达玛积。使用对角矩阵(即卷积核) $g_\theta$ 代替向量 $U^T g$ ,此时哈达玛乘积便可转化为矩阵乘法。因此。作用在图上的卷积公式可表示为:

$$f * _G g=U g_\theta (U^T f) \tag{19}$$

(4)模型:通过图卷积算子的定义,陆续出现了很多相关工作。例如 Eruna 等<sup>[46]</sup>提出的谱域卷积神经网络(Spectral CNN)是最早的基于谱域的 GCN,也被称为谱域下的第一代图卷积神经网络。该模型的提出使得传统 CNN 可成功迁移到图这种不规则的数据结构下,并且不再要求待处理的数据具有平移不变性,从而完美地适应了图结构数据的节点无序、邻居个数不确定的特点<sup>[47]</sup>。具体地,Spectral CNN 在第  $k+1$  层的网络结构可表示为:

$$\mathbf{X}_j^{(k+1)}=\sigma(U \sum_{i=1}^p g_{\theta_{i,j}}^{(k)} U^T \mathbf{X}_i^{(k)}),j=1,2,\cdots,q \tag{20}$$

其中, $p$ 和 $q$ 分别表示输入特征和输出特征的维度; $\mathbf{X}_i^{(k)}$ 表示图上某一具体节点在第  $k$  层的第  $i$  个特征表示; $g_{\theta_{i,j}}^{(k)}$ 表示谱域空间下的卷积核; $\sigma$ 表示非线性激活函数。

Spectral CNN 虽然为图卷积神经网络的研究奠定了基石,但其在卷积过程中依赖于拉普拉斯矩阵的特征分解。在大规模图上进行卷积操作时,计算复杂度较高。针对第一代 GCN 的缺点,Defferrard 等<sup>[48]</sup>提出了一种新的图卷积神经网络模型,即切比雪夫网络(Chebyshev network,ChebNet)。该网络证明了谱域中的卷积核  $g_\theta$  可通过切比雪夫多项式的截断展开来近似,从而省略了拉普拉斯矩阵中复杂的特征分解操作,降低了计算复杂度,提高了计算效率。其中,切比雪夫多项式的递推关系如式(21)所示:

$$T_{k+1}(x)=2xT_k(x)-T_{k-1}(x),k \in N^+ \tag{21}$$

其中, $T_0=1,T_1=x$ 。由此,谱域中的卷积核 $g_\theta$ 可表示为:

$$g_\theta(\Lambda)=\sum_{k=0}^{K-1}\theta_k T_k(\tilde{\Lambda}) \tag{22}$$

其中,  $\tilde{\mathbf{A}} = \frac{2\mathbf{A}}{\lambda_{\max}} - \mathbf{I}_n$ ,  $\mathbf{A}$  表示由拉普拉斯矩阵  $\mathbf{L}$  的特征值所构成的对角矩阵,  $\lambda_{\max}$  表示矩阵  $\mathbf{L}$  中的最大特征值,  $\mathbf{I}_n$  表示维度为  $n$  的单位矩阵;  $\boldsymbol{\theta} \in \mathbf{R}^K$  表示切比雪夫系数向量。除了式(22)以外, ChebNet 的卷积操作还可写成:

$$g_{\boldsymbol{\theta}}(\mathbf{L})f = \sum_{k=0}^{K-1} \boldsymbol{\theta}_k T_k(\tilde{\mathbf{L}})f \quad (23)$$

其中,  $\tilde{\mathbf{L}} = \frac{2\mathbf{L}}{\lambda_{\max}} - \mathbf{I}_n$ 。

利用切比雪夫多项式近似的卷积核  $g_{\boldsymbol{\theta}}$  来替换式(20)中的  $g_{\boldsymbol{\theta}_{i,j}^{(k)}}$ , 便能够得到 ChebNet 下的卷积公式:

$$\mathbf{X}_j^{(k+1)} = \sigma(\mathbf{U} \sum_{i=1}^p \sum_{k=0}^{K-1} \boldsymbol{\theta}_k T_k(\tilde{\mathbf{A}}) \mathbf{U}^T \mathbf{X}_i^{(k)}), j=1, 2, \dots, q \quad (24)$$

进一步地, 将式(24)化简之后便可得到:

$$\mathbf{X}_j^{(k+1)} = \sigma(\sum_{i=1}^p \sum_{k=0}^{K-1} \boldsymbol{\theta}_k T_k(\tilde{\mathbf{L}}) \mathbf{X}_i^{(k)}), j=1, 2, \dots, q \quad (25)$$

与 Spectral CNN 相比, ChebNet 具有学习复杂度低和计算效率高等优势。为了形象地表达图上的卷积操作, 文献[48]给出了图卷积的基本结构, 如图 6 所示。

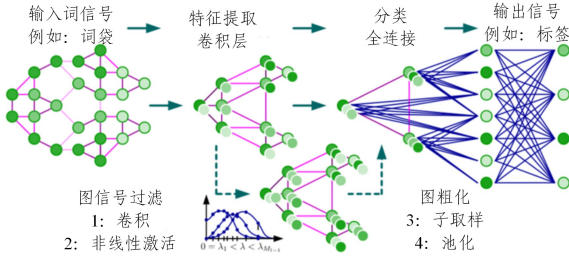


图 6 图卷积网络的基本结构

Fig. 6 Basic structure of graph convolutional networks

为了进一步降低计算复杂度, Kipf 和 Welling<sup>[49]</sup> 提出的 GCN 模型只考虑了一阶切比雪夫多项式, 这就使得每个卷积核只有一个可训练的参数。进一步地, 将拉普拉斯矩阵的最大特征值  $\lambda_{\max}$  近似为 2, 便可将式(23)简化为:

$$g_{\boldsymbol{\theta}}f \approx \theta_0' f + \theta_1' (\mathbf{L} - \mathbf{I}_N) f = \theta_0' f - \theta_1' \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} f \quad (26)$$

其中,  $\theta_0'$  和  $\theta_1'$  表示两个自由参数。

在实践中, 为了提高计算效率并解决过拟合问题, Kipf 和 Welling 指出最小化卷积层的运算操作(如矩阵乘法)和减少模型的参数是有好处的, 于是令  $\theta = \theta_0' = -\theta_1'$ , 便得到了式(27):

$$g_{\boldsymbol{\theta}}f \approx \theta(\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}) f \quad (27)$$

针对半监督节点分类问题, 为了在图上有效地传播信息, 文献[49]引入了一个简单且灵活的模型  $f(\mathbf{X}, \mathbf{A})$ 。之所以借助数据  $\mathbf{X}$  和邻接矩阵  $\mathbf{A}$  对模型  $f(\mathbf{X}, \mathbf{A})$  进行调节, 是因为在某些场景下邻接矩阵包含数据  $\mathbf{X}$  中不存在的信息, 例如引文网络中文档之间的引用关系或者知识谱图中的关系。图 7 给出了用于半监督学习的多层 GCN 结构。

在实际的半监督节点分类问题中, Kipf 和 Welling 使用了一个两层 GCN 结构, 先在预处理阶段计算得到  $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$ , 然后模型在神经网络的前向传播中采用如下形式:

$$\begin{cases} \mathbf{Z} = f(\mathbf{X}, \mathbf{A}) \\ f(\mathbf{X}, \mathbf{A}) = \text{softmax}(\hat{\mathbf{A}} \text{ReLU}(\hat{\mathbf{A}} \mathbf{X} \mathbf{W}^{(0)})) \mathbf{W}^{(1)} \end{cases} \quad (28)$$

其中,  $\mathbf{W}^{(0)} \in \mathbf{R}^{C \times H}$  表示由输入层到隐藏层的权重矩阵,  $\mathbf{W}^{(1)} \in \mathbf{R}^{H \times F}$  表示由隐藏层到输出层的权重矩阵, softmax 表示激活函数。对于半监督多标签分类问题, 最后的损失可使用交叉熵进行计算:

$$\text{Loss} = - \sum_{i \in y_L} \sum_{f=1}^F Y_{if} \ln Z_{if} \quad (29)$$

其中,  $y_L$  表示所有节点标签的集合。

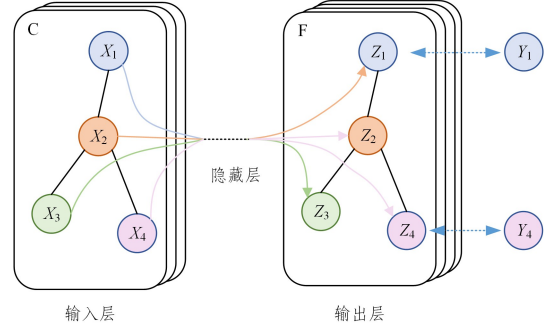


图 7 图卷积网络

Fig. 7 Graph convolutional networks

除了上述基于谱域的 GCN 外, 还有很多学者对这方面的工作进行了详细研究。例如 Li 等<sup>[50]</sup> 针对大部分 GCN 模型只能在固定大小的图结构上进行卷积操作的问题, 提出了自适应图卷积网络 (Adaptive Graph Convolution Network, AGCN), 该网络可在训练过程中处理任意类型的图输入, 并得到一个基于任务驱动的自适应图表示; Xu 等<sup>[51]</sup> 提出的图小波神经网络 (Graph Wavelet Neural Network, GWNN) 模型借助快速算法得到图小波变换, 避免了计算量庞大的矩阵分解操作, 从而提高了算法的效率。但不难看出, 基于谱域的图卷积神经网络严重依赖图拉普拉斯矩阵的特征分解, 并且计算复杂度较高。这限制了其应用于大规模图, 同时学习滤波器依赖于与图相关的拉普拉斯特征基, 这意味着在某一特定结构上训练的模型无法直接应用于不同结构的图。针对基于谱域图卷积神经网络的缺点, 已经有很多工作将研究重点转移到基于空域的图神经网络上。鉴于基于空域的图卷积神经网络强大的优势以及应用范围, 下面对其进行详细介绍。

## 2) 基于空域的图卷积神经网络

基于空域的 GCN 直接在图上定义卷积, 并且基于消息传播机制, 利用中心节点的邻居特征来更新自身的特征表示。这种方法面临的主要挑战是需要图的拓扑结构上定义具有不同大小邻域的卷积操作并保持 CNN 的局部不变性。最早基于空域的 GCN 可追溯到 Micheli<sup>[52]</sup> 提出的 NN4G 模型, 但该模型在当时并没有引起太大的反响, 直到 2016 年 Niepert 等<sup>[6]</sup> 提出了 PATCHY-SAN 模型, 才正式开启了对空域 GCN 的研究。PATCHY-SAN 模型的基本思想是, 首先为每个中心节点选择  $K$  个邻居节点, 并对其进行规范化处理, 然后将归一化的邻域用作图卷积操作的感受野。具体地, PATCHY-SAN 模型共包含 4 个步骤:

(1) 节点序列选择: 这是为输入图确定节点序列并为其构建卷积操作感受野的前置过程。首先根据给定的图标签对输入图进行排序, 然后使用给定的步长  $s$  为每一个



中心节点选择  $w$  个节点。

(2)为中心节点选择固定大小的邻域:在这一步中,构建从上一步中选择出的  $w$  个中心节点的感受野。针对每一个中心节点,其邻居节点均属于候选者。具体地,PATCHY-SAN 模型使用广度优先搜索策略为每个中心节点选择  $k$  个邻居节点,如果一阶邻域的节点个数少于  $k$  个,则从高阶邻域中进行选择,直到选出  $k$  个邻居节点。

(3)图规范化:其目的在于为每个中心节点邻域中的节点标注顺序,以便将节点从无序图空间映射到具有线性顺序的向量空间。其基本思想是,当且仅当两个不同的节点在图中的结构角色相似时,将其分配到各自邻接矩阵中的相对位置。

(4)卷积架构:中心节点的邻域经过第 3 步的图规范化后可视为卷积操作的感受野,节点和边的属性值可作为卷积的通道。

图 8 展示了 PATCHY-SAN 模型进行图卷积的基本过程。虽然 PATCHY-SAN 模型成功地在图上应用了 CNN 的思想,但仅限于在小规模图上进行操作,难以在大规模图上生成有效的节点嵌入表示。

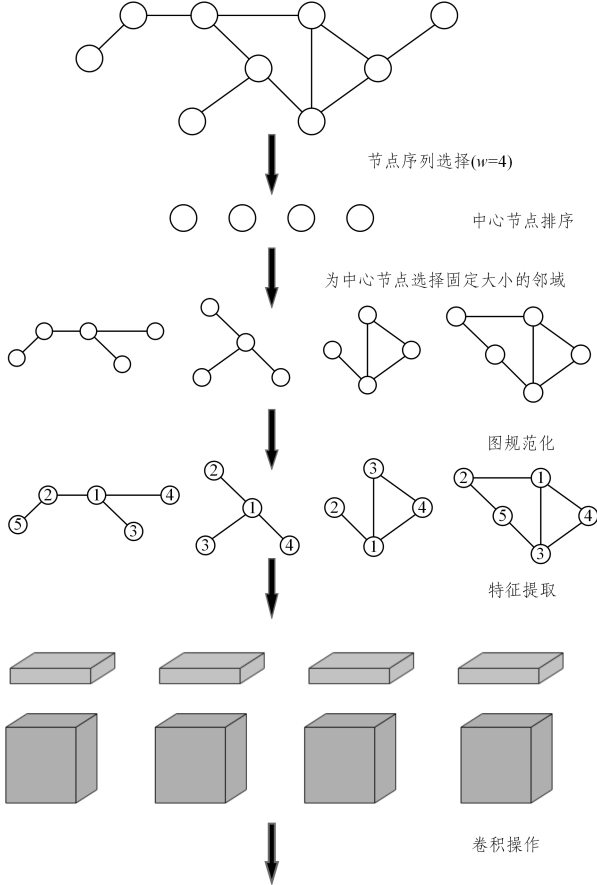


图 8 PATCHY-SAN 模型的卷积操作过程

Fig. 8 Convolution operation of PATCHY-SAN model

针对该问题,Hamilton 等<sup>[14]</sup>提出了一种归纳式学习方法 GraphSAGE。与传统 GNN 的学习方法不同,GraphSAGE 并不需要为每一个节点训练单独的嵌入表示,而是通过学习一个可以先从目标节点的邻域进行取样,然后再聚合特征的函数。借助该函数,便可得到目标节点的嵌入表示。详细地,GraphSAGE 可通过以下 3 个步骤生成目标节点的嵌入表示:

(1)从目标节点的一阶或二阶邻域内取样邻居节点;(2)使用不同的聚合函数聚合邻居节点的特征;(3)生成目标节点的嵌入表示。算法 3 描述了 GraphSAGE 节点嵌入表示的生成过程。

### 算法 3 GraphSAGE 节点嵌入生成算法

输入:图  $G(V, E)$ , 输入特征  $\{x_v, \forall v \in V\}$ , 深度  $K$ , 权重矩阵  $W^k$ ,  $\forall k \in \{1, \dots, K\}$ , 非线性激活函数  $\sigma$ , 可区分的聚合函数  $AGGREGATE_k, \forall k \in \{1, \dots, K\}$ , 邻居函数  $N: v \rightarrow 2^V$

输出:对每一个节点  $v \in V$ , 其向量表示为  $z_v$

```

1.  $h_v^0 \leftarrow x_v, \forall v \in V$ 
2. for  $k=1, \dots, K$  do
3.   for  $v \in V$  do
4.      $h_{N(v)}^k \leftarrow AGGREGATE_k(\{h_u^{k-1}, \forall u \in N(v)\})$ ;
5.      $h_v^k \leftarrow \sigma(W^k \cdot \text{CONCAT}(h_v^{k-1}, h_{N(v)}^k))$ 
6.   end for
7.    $h_v^k \leftarrow h_v^k / \|h_v^k\|_2, \forall v \in V$ 
8. end for
9.  $z_v \leftarrow h_v^K, \forall v \in V$ 

```

整个算法的核心步骤在于第 4 行和第 5 行的两个公式,对这两个公式进行详细分析:

$$\begin{cases} h_{N(v)}^k = AGGREGATE_k(\{h_u^{k-1}, \forall u \in N(v)\}) \\ h_v^k = \sigma(W^k \cdot \text{CONCAT}(h_v^{k-1}, h_{N(v)}^k)) \end{cases} \quad (30)$$

上式表示中心节点  $v$  先通过聚合函数聚合邻居节点特征得到  $h_{N(v)}^k$ , 然后将其与目标节点在图神经网络第  $k-1$  层的表示  $h_v^{k-1}$  进行特征融合,得到目标节点的最终嵌入表示。文献[53]研究了式(30)中 3 个不同的聚合函数:

(1)均值聚合器:与直推式 GCN 中使用的卷积传递规则类似,该聚合器只需要对邻居节点的特征  $\{h_u^{k-1}, \forall u \in N(v)\}$  按元素取均值。

$$h_v^k = \sigma(W \cdot \text{MEAN}(\{h_v^{k-1}\} \cup \{h_u^{k-1}, \forall u \in N(v)\})) \quad (31)$$

均值聚合器与其他类型聚合器的不同之处在于它并不执行目标节点在前一层的特征表示  $h_v^{k-1}$  与邻居节点特征  $h_{N(v)}^k$  的拼接操作,而是使用了一种叫作“跳跃连接”的方式来实现更好的性能。

(2)长短期记忆聚合器:该聚合器基于 LSTM 架构,与均值聚合器相比,该函数更复杂且表示能力更强大。由于 LSTM 处理的数据需要按照顺序方式进行输入,因此该聚合函数并不是对称的,若要在图这种无序集中进行操作,只需要将节点的邻居进行随机排列即可。

(3)池化聚合器:池化聚合器中的每个邻居节点的向量会独立地通过一个全连接神经网络,经过该线性变换之后,在邻居节点的向量集合中使用最大池化操作来聚合信息。

$$AGGREGATE_k^{\text{pool}} = \max(\{\sigma(W_{\text{pool}} h_{u_i}^k + b), \forall u_i \in N(v)\}) \quad (32)$$

其中,  $\max$  表示取各个元素的最大值,  $\sigma$  表示非线性激活函数。

根据 GraphSAGE 算法的基本思想,图 9 展示了该方法在预测节点标签时的详细过程。GraphSAGE 算法之所以能够应用于大规模网络图,是因为该算法只需要统一地对固定大小的邻居集合进行采样,而不需要对所有邻居进行处理,从而保证了整个算法的计算效率。



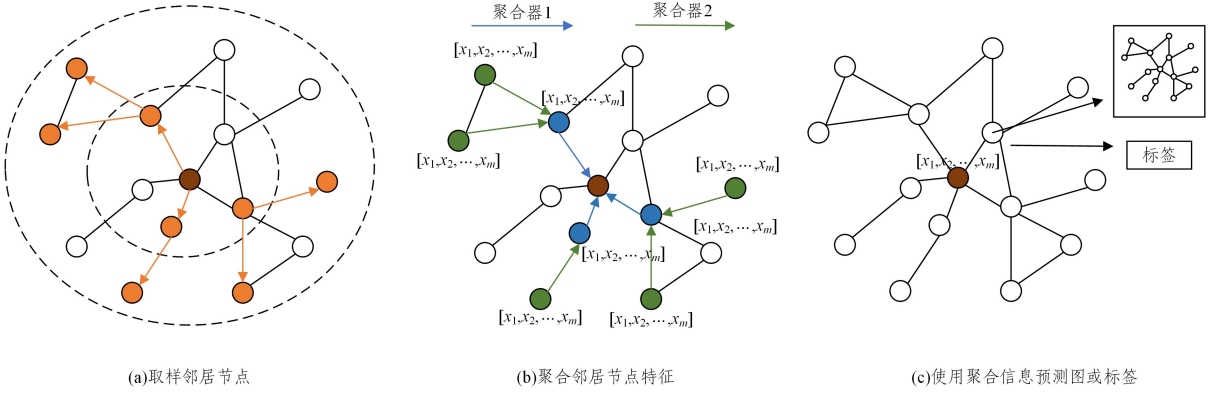


图9 GraphSAGE 取样和聚集信息示意图

Fig. 9 Sample and aggregate information diagram of GraphSAGE

### 3.2.2 图注意力神经网络

注意力机制已经在机器翻译<sup>[53-55]</sup>、机器阅读<sup>[56]</sup>等许多序列建模任务中取得了巨大成功,其优势在于可关注一些更感兴趣的信息,从而更好地完成既定任务。与同等对待所有节点重要性的GCN模型不同,在GNN上应用注意力机制便可形成图注意力网络模型,其可为目标节点的邻居节点分配不同的注意力分数,从而确定更重要的邻居节点。Velickovic等<sup>[13]</sup>提出的GAT(Graph Attention Networks)模型首次将注意力机制引入GNN中,核心思想是通过神经网络学习不同邻居节点的权重,进而在对目标节点进行信息聚合时,只关注对目标节点作用大的节点,而忽视一些作用较小的节点。GAT模型的基本结构如图10所示。

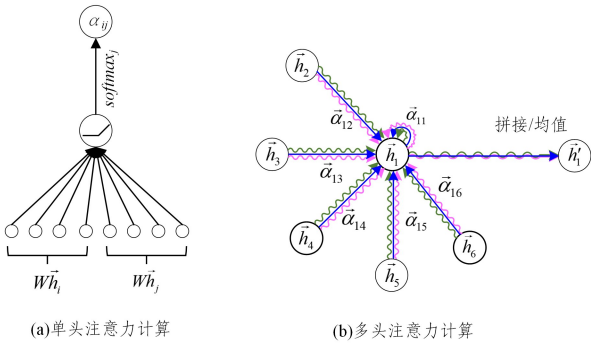


图10 GAT模型基本结构

Fig. 10 Basic structure of GAT model

Velickovic等通过堆叠单个图注意力层来构建任意的图注意力网络,该注意力层可通过如下公式计算节点*i*和*j*之间的注意力分数:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j]))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_k]))} \quad (33)$$

其中, $\alpha_{ij}$ 表示节点*j*对节点*i*的注意力系数, $N_i$ 表示节点*i*的全部邻居。GAT中输入节点的特征可表示为 $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N\}$ ,  $\mathbf{h}_i \in \mathbf{R}^F$ 。其中*N*表示图上所有节点的数量,*F*表示向量维度。输出节点特征可表示为 $\mathbf{h}' = \{\mathbf{h}'_1, \mathbf{h}'_2, \dots, \mathbf{h}'_N\}$ ,  $\mathbf{h}'_i \in \mathbf{R}^{F'}$ 。 $\mathbf{W} \in \mathbf{R}^{F' \times F}$ 表示应用于每个节点的共享权重矩阵。 $\mathbf{a} \in \mathbf{R}^{2F'}$ 表示权重向量,符号 $\parallel$ 表示向量拼接操作。LeakyReLU表示泄露性修正线性单元函数。之后应用非线性激活函数 $\sigma$ ,便可得到每个节点的最后输出特征:

$$\mathbf{h}'_i = \sigma(\sum_{j \in N_i} \alpha_{ij} \mathbf{W}\mathbf{h}_j) \quad (34)$$

为了稳定自注意力机制的学习过程, Velickovic等应用了多头注意力机制,把原来只进行一次注意力分数的计算过程改为由*K*个不同的注意力机制进行计算。这*K*个注意力机制在工作过程中彼此相互独立,互不影响。最后通过向量拼接或求均值的方法得到节点的最终特征表示。

$$\mathbf{h}'_i = \parallel_{k=1}^K \sigma(\sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j) \quad (35)$$

$$\mathbf{h}'_i = \sigma\left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in N_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j\right) \quad (36)$$

其中,*K*表示注意力头的总个数, $\alpha_{ij}^k$ 表示从第*k*个注意力头中计算得到的注意力系数, $\mathbf{W}^k$ 表示第*k*个注意力头的线性变换矩阵。

与之前只在图结构数据上进行操作的神经网络相比,GAT模型具有计算高效、可解释性强和适用性广3点优势。除GAT模型以外,Zhang等<sup>[57]</sup>提出的门控注意力网络(Gated Attention Network, GaAN)同样采用了多头注意力机制,但与GAT使用全连接计算注意力系数不同的是,GaAN模型采用的是“键-值”注意力机制,并使用点积计算两个节点之间的注意力系数。具体地,GaAN模型首先通过线性变换将中心节点的特征 $\mathbf{x}_i$ 映射为“查询向量”,然后用类似的方式得到邻居节点的“键向量”和“值向量”。详细的信息聚合过程如下:

$$\begin{cases} \mathbf{y}_i = FC_{\theta_{\alpha}}(\mathbf{x}_i \oplus \parallel_{j \in N_i}^K \omega_{i,j}^{(k)} FC_{\theta_{\alpha}^{(k)}}(\mathbf{z}_j)) \\ \omega_{i,j}^{(k)} = \frac{\exp(Q_w^{(k)}(\mathbf{x}_i, \mathbf{z}_j))}{\sum_{l=1}^{N_i} \exp(Q_w^{(k)}(\mathbf{x}_i, \mathbf{z}_l))} \\ Q_w^{(k)}(\mathbf{x}, \mathbf{z}) = \langle FC_{\theta_{\alpha}^{(k)}}(\mathbf{x}), FC_{\theta_{\alpha}^{(k)}}(\mathbf{z}) \rangle \end{cases} \quad (37)$$

其中, $FC_{\theta}^h(\mathbf{x}) = h(\mathbf{W}\mathbf{x} + \mathbf{b})$ 表示使用LeakyReLU作为激活函数的全连接层, $\mathbf{W}$ 和 $\mathbf{b}$ 为可学习的参数,而 $FC_{\theta_{\alpha}}(\mathbf{x})$ 表示不使用激活函数的全连接层;*K*表示注意力头的数量; $\omega_{i,j}^{(k)}$ 表示中心节点*i*和邻居节点*j*在第*k*个注意力机制下的权重; $N_i$ 和 $\mathbf{z}_i$ 分别表示中心节点*i*的全部邻居和邻居节点向量;符号 $\langle \cdot, \cdot \rangle$ 代表点积操作; $\theta_{\alpha}^{(k)}$ ,  $\theta_{\alpha}^{(k)}$ 和 $\theta_{\alpha}^{(k)}$ 分别表示计算查询、键和值向量的第*k*个注意力头的参数; $\oplus$ 表示向量拼接操作。多头注意力机制计算得到的向量与中心节点的初始向量 $\mathbf{x}_i$ 先进行拼接,然后通过一个参数为 $\theta_{\alpha}$ 的全连接层,便可得到中心节点的最终表示向量 $\mathbf{y}_i$ 。图11展示了3头

门控注意力机制的计算过程。

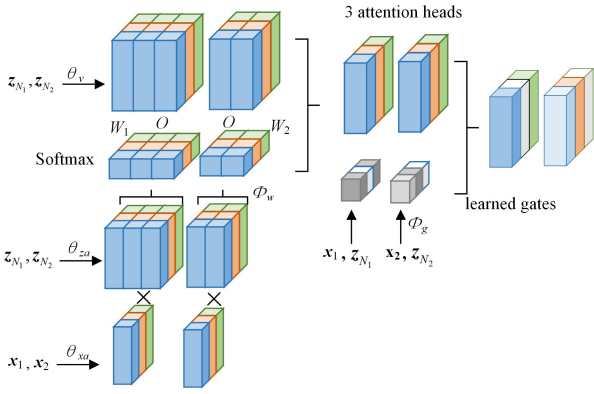


图 11 3 头门控注意力机制的计算

Fig. 11 Calculation of 3-head gated attention mechanism

GaAN 模型指出,多头注意力机制能够学习和探索中心节点及其邻域之间的多个表示子空间,但不同注意力头得到的子空间表示并非同等重要,这会导致某些注意力头可能无法捕获到有用的节点表示信息。针对此问题,文献[57]首先计算了一个介于 0(低重要性)和 1(高重要性)之间的附加软门,以便为不同的注意力头分配不同的权重。然后结合多头注意力聚合器,设计出了门控注意力聚合器。其详细的数学公式如下:

$$\begin{cases} y_i = FC_{\theta_a}(x_i \oplus \bigvee_{k=1}^K (g_i^{(k)} \sum_{j \in N_i} w_{i,j}^{(k)} FC_{\theta_c}^h(z_j))) \\ g_i = [g_i^{(1)}, \dots, g_i^{(K)}] = \Psi_g(x_i, z_{N_i}) \end{cases} \quad (38)$$

其中,  $z_{N_i}$  表示中心节点的所有邻居节点特征;  $g_i^{(k)}$  是一个标量,表示中心节点  $i$  在第  $k$  个注意力机制下的门控值。为了保证附加软门不会引入太多的额外参数,使用卷积网络  $\Psi_g$  对中心节点和邻居节点进行处理而生成门控值。其余参数与式(37)具有相同的含义。

特别地,卷积网络  $\Psi_g$  的设计有多种方式,在文献[57]中,作者采用最大池化和平均池化相结合的方式进行构造:

$$g_i = FC_{\theta_g}^a \left( x_i \oplus \max_{j \in N_i} (FC_{\theta_m}(z_j)) \oplus \frac{\sum_{j \in N_i} z_j}{|N_i|} \right) \quad (39)$$

其中,参数  $\theta_m$  表示将邻居节点的特征向量  $z_j$  映射到  $d_m$  维空间中,  $\theta_g$  表示将拼接好的特征向量映射到最终的  $K$  个门控值中。Zhang 等[57]在 PPI 和 Reddit 数据集上证明了 GaAN 模型在节点分类上的优越性能,并且该模型也被成功应用于交通速度预测问题上。

### 3.2.3 图残差网络

为了在 GNN 中获得更准确的节点特征表示,许多应用程序试图通过堆叠更多层的网络来进行处理,因为更多的层意味着可从中心节点的高阶邻域内进行聚集信息。但随着层数的增加,模型性能往往会由于过平滑性问题而下降。受到计算机视觉领域中残差网络[58]的启发,许多研究者企图通过残差连接来解决过平滑性问题进而提升模型的性能。本文将在 GNN 中应用残差连接的网络称为图残差网络。

在图残差网络的研究中,比较知名的有 Li 等[59]提出的 DeepGCN 模型。该模型使用残差连接和密集连接来解决多层图神经网络中的梯度消失问题。基于此, Li 等在标准 GCN

(PlainGCN)的基础上分别提出了 ResGCN 和 DenseGCN 方法。图 12 展示了 PlainGCN, ResGCN 和 DenseGCN 的原理图。

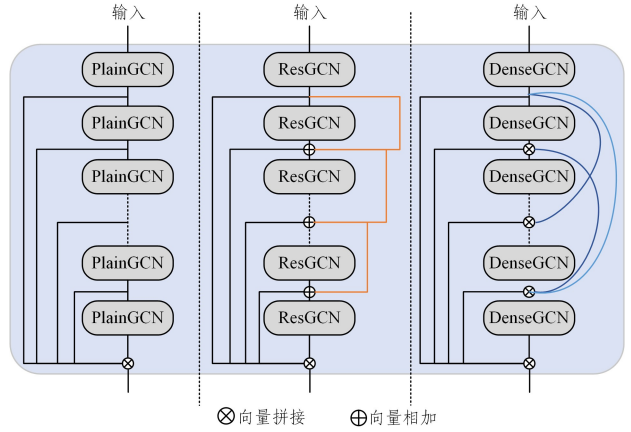


图 12 PlainGCN, ResGCN 和 DenseGCN 的原理图

Fig. 12 Schematic diagram of PlainGCN, ResGCN and DenseGCN

1) ResGCN: 在原始的图学习框架中,第  $l$  层的图卷积运算可表示为以下的聚合和更新操作。

$$\begin{cases} \mathbf{G}_{l+1} = F(\mathbf{G}_l, \mathbf{W}_l) \\ F(\mathbf{G}_l, \mathbf{W}_l) = \text{Update}(\text{Aggregate}(\mathbf{G}_l, \mathbf{W}_l^{\text{agg}}), \mathbf{W}_l^{\text{update}}) \end{cases} \quad (40)$$

其中,  $\mathbf{G}_l$  和  $\mathbf{G}_{l+1}$  分别表示图神经网络在第  $l$  层的输入图和输出图;  $\mathbf{W}_l^{\text{agg}}$  和  $\mathbf{W}_l^{\text{update}}$  分别表示聚合函数和更新函数可学习的权重。对图残差网络来说,第  $l$  层的聚合更新操作可表示为:

$$\mathbf{G}_{l+1}^{\text{res}} = \mathbf{G}_{l+1} + \mathbf{G}_l = F(\mathbf{G}_l, \mathbf{W}_l) + \mathbf{G}_l \quad (41)$$

其中,隐藏状态矩阵  $\mathbf{G}_l$  可直接添加到经过图卷积操作的矩阵之后,从而得到最终的图表示矩阵  $\mathbf{G}_{l+1}^{\text{res}}$ 。

2) DenseGCN: Huang 等[60]提出的 DenseNet 旨在利用层间的密集连接,从而改善网络中的信息流,并实现层间特征的高效重用。受到 DenseNet 的启发,他们将类似的思想应用到 GCN 中,以便利用来自不同 GCN 层的信息流。具体地,在 DenseGCN 中,有如下状态更新公式:

$$\begin{cases} \mathbf{G}_{l+1}^{\text{dense}} = T(\mathbf{G}_{l+1}, \mathbf{G}_l, \dots, \mathbf{G}_0) \\ T(\mathbf{G}_{l+1}, \mathbf{G}_l, \dots, \mathbf{G}_0) = T(F(\mathbf{G}_l, \mathbf{W}_l), \\ F(\mathbf{G}_{l-1}, \mathbf{W}_{l-1}), \dots, \mathbf{G}_0) \end{cases} \quad (42)$$

其中,运算符  $T$  是一个顶点连接函数,它可将输入图  $\mathbf{G}_0$  与 GCN 每一层的输出做特征融合。

3) 空洞卷积:在 CNN 中,池化操作通常会造成空间信息的丢失。针对该问题, Yu 等[61]提出使用空洞卷积来替代密集预测任务(如语义图像分割等)中的连续池化层。经实验验证,空洞卷积可在不丢失分辨率的情况下扩大感受野的范围,由此可通过聚集多尺度上下文信息来提高语义分割任务的准确性。基于空洞卷积的思想,他们在 GCN 中引入了空洞聚合。具体地,在每一层的 GCN 之后,使用空洞率为  $d$  的空洞  $k$ -NN 来查找空洞邻居,从而构建空洞图。针对每一个节点,该方法首先使用一个预先定义的度量标准来计算  $k * d$  个最近邻居,然后通过每  $d$  跳来选择邻居,直到获得  $k$  个邻居。例如节点  $v$  的最近  $k * d$  邻居集合为  $(u_1, u_2, \dots, u_{k*d})$ , 则其空洞邻居为  $(u_1, u_{1+d}, u_{1+2d}, \dots, u_{1+(k-1)d})$ 。空洞卷积利用不同的上下文信息来扩大目标节点的感受野,并在过平滑性问题

上被证明是有效的。图 13 展示了空洞率分别为 1, 2, 3 的空洞卷积示例。

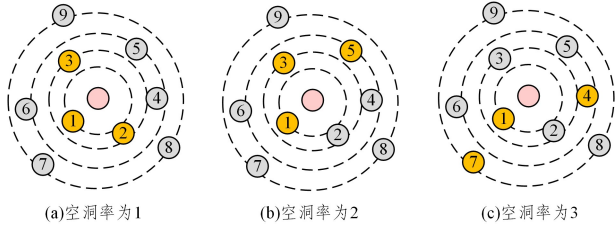


图 13 空洞卷积示例

Fig. 13 Example of hollow convolution

### 3.2.4 图递归网络

受到门控机制的启发,在图神经网络的信息传播过程中应用 GRU 或 LSTM 等门控循环单元来提高模型处理序列数据的有效性并解决长距离依赖性问题已经成为一个新的发展趋势。在 GNN 中应用门控单元的模型被称为图递归网络。Li 等<sup>[41]</sup>提出的门控图神经网络(GGNN)是一种典型的图递归神经网络,主要思想是在 GNN 中使用 GRU 门控单元对固定的传播步数  $T$  进行迭代循环,从而实现对每个节点隐藏状态的更新。GGNN 在信息传播过程中主要通过以下 6 个公式实现:

$$\mathbf{h}_v^{(1)} = [\mathbf{x}_v^T, 0]^T \quad (43)$$

$$\mathbf{a}_v^{(t)} = \mathbf{A}_{v,:}^T [\mathbf{h}_1^{(t-1)^T} \cdots \mathbf{h}_{|V|}^{(t-1)^T}]^T + \mathbf{b} \quad (44)$$

$$\mathbf{z}_v^{(t)} = \sigma(\mathbf{W}^z \mathbf{a}_v^{(t)} + \mathbf{U}^z \mathbf{h}_v^{(t-1)}) \quad (45)$$

$$\mathbf{r}_v^{(t)} = \sigma(\mathbf{W}^r \mathbf{a}_v^{(t)} + \mathbf{U}^r \mathbf{h}_v^{(t-1)}) \quad (46)$$

$$\widetilde{\mathbf{h}}_v^{(t)} = \tanh(\mathbf{W} \mathbf{a}_v^{(t)} + \mathbf{U}(\mathbf{r}_v^{(t)} \odot \mathbf{h}_v^{(t-1)})) \quad (47)$$

$$\mathbf{h}_v^{(t)} = (1 - \mathbf{z}_v^{(t)}) \odot \mathbf{h}_v^{(t-1)} + \mathbf{z}_v^{(t)} \odot \widetilde{\mathbf{h}}_v^{(t)} \quad (48)$$

其中,式(43)中的  $\mathbf{h}_v^{(1)} \in R^D$  表示节点  $v$  的初始状态;  $\mathbf{x}_v$  表示节点注释向量,如果  $\mathbf{h}_v^{(1)}$  的维度大于  $\mathbf{x}_v$  的维度,则将剩余维度用 0 补齐。式(44)中的  $\mathbf{A}_{v,:} \in R^{D|V| \times 2D}$  表示矩阵  $\mathbf{A} \in R^{D|V| \times 2D|V|}$  的子矩阵,与矩阵  $\mathbf{A}$  中目标节点相关的列对应,  $\mathbf{a}_v^{(t)}$  聚集了节点  $v$  的邻居节点信息。式(45)一式(48)类似于 GRU 门控单元的状态更新,  $\mathbf{z}_v^{(t)}$  表示在第  $t$  步的更新门,  $\mathbf{r}_v^{(t)}$  表示在第  $t$  步的重置门。符号  $\sigma$  表示 sigmoid 激活函数,  $\odot$  表示哈达玛积。

GGNN 模型专门为输出是节点序列的图问题而定义,可支持多种不同类型的下游任务,如节点级别或图级别。在节点级别任务中,可在每个节点  $v$  上通过式(49)来获得节点的分值:

$$o_v = g(\mathbf{h}_v^{(T)}, \mathbf{x}_v) \quad (49)$$

其中,  $o_v$  表示节点的输出分数,  $\mathbf{h}_v^{(T)}$  表示节点  $v$  的最终表示。而对于图级别的输出,可采用如下公式:

$$\mathbf{h}_G = \tanh(\sum_{v \in V} \sigma(i(\mathbf{h}_v^{(T)}, \mathbf{x}_v)) \odot \tanh(j(\mathbf{h}_v^{(T)}, \mathbf{x}_v))) \quad (50)$$

其中,  $\sigma(i(\mathbf{h}_v^{(T)}, \mathbf{x}_v))$  和  $\tanh(j(\mathbf{h}_v^{(T)}, \mathbf{x}_v))$  充当软注意力机制,决定哪些节点与当前图级别任务相关。  $i$  和  $j$  表示神经网络,将  $\mathbf{h}_v^{(T)}$  和  $\mathbf{x}_v$  的拼接向量作为输入。进一步地,在 GGNN 的基础上, Li 等还提出了门控图序列神经网络(Gated Graph Sequence Neural Networks, GGS-NN),该模型使用若干个 GGNN 来产生输出序列  $\mathbf{o}^{(1)}, \mathbf{o}^{(2)}, \dots, \mathbf{o}^{(K)}$ 。

综上所述,对图神经网络的各个模型的详细总结如表 3 所列。

表 3 不同 GNN 模型总结

Table 3 Summary of different GNN models

| 划分标准         | 模型               | 年份   | 文献   | 应用         |
|--------------|------------------|------|------|------------|
| 基于谱域的图卷积神经网络 | Spectral Network | 2014 | [46] | 子图匹配       |
|              | ChebNet          | 2016 | [48] | 节点分类       |
|              | GCN              | 2017 | [49] | 节点分类       |
|              | AGCN             | 2018 | [50] | 多任务预测、节点分类 |
| 基于空域的图卷积神经网络 | GWNN             | 2019 | [51] | 节点分类       |
|              | PATCHY-SAN       | 2016 | [6]  | 图分类        |
|              | DCNN             | 2016 | [8]  | 节点分类、图分类   |
|              | DGCN             | 2018 | [62] | 节点分类       |
| 图注意力网络       | LGCN             | 2018 | [7]  | 节点分类       |
|              | GraphSAGE        | 2017 | [14] | 节点分类       |
| 图递归网络        | GAT              | 2018 | [13] | 节点分类       |
|              | GaAN             | 2018 | [57] | 节点分类和预测    |
| 图残差网络        | GGNN             | 2016 | [41] | 分类和预测      |
|              | Graph LSTM       | 2016 | [63] | 语义对象分析     |
|              | DeepGCN          | 2019 | [59] | 点云语义分割     |

## 4 图神经网络通用框架

所谓的图神经网络通用框架,即图神经网络编程的通用范式,是对多种不同变体的图神经网络结构进行的一般化总结<sup>[64]</sup>,旨在将不同的模型整合到一个框架中。研究图神经网络通用框架有助于我们对现有的图神经网络模型做出灵活的扩展与改进,从而使其适应不同类型的任务。本章详细介绍消息传递神经网络(Message Passing Neural Network, MPNN)<sup>[65]</sup>、非局部神经网络(Non-local Neural Network, NLNN)<sup>[66]</sup>和图网络(Graph Network, GN)<sup>[67]</sup>这 3 种通用的图神经网络框架。

### 4.1 MPNN

文献[65]提出的 MPNN 框架是在图上进行监督学习的通用框架,它基于消息传播机制对不同类型的 GNN 模型进行了一般意义上的总结。MPNN 指出在 GNN 的前向传播过程中共包括消息传播和读出两个阶段。进一步地,消息传播阶段由消息函数  $M_t$  和节点更新函数  $U_t$  经过  $T$  步迭代定义。在网络中进行消息传播时,图中节点的隐藏状态  $\mathbf{h}_v^t$  可借助消息  $\mathbf{m}_v^t$  得到。

$$\begin{cases} \mathbf{m}_v^{t+1} = \sum_{w \in N(v)} M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}) \\ \mathbf{h}_v^{t+1} = U_t(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}) \end{cases} \quad (51)$$

其中,  $N(v)$  表示图  $G$  中节点  $v$  的邻居节点,  $\mathbf{e}_{vw}$  表示从节点  $v$  到节点  $w$  的边上的表示向量。在读出阶段使用读出函数  $R$  计算整个图的特征向量。

$$\hat{\mathbf{y}} = R(\{\mathbf{h}_v^T | v \in G\}) \quad (52)$$

其中,  $T$  表示在图神经网络中总的迭代次数。

MPNN 框架可通过设置不同的消息函数、节点更新函数以及读出函数来泛化不同的模型。在 Li 等<sup>[41]</sup>提出的 GGNN 中,函数设置如下:

$$\begin{cases} M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}) = \mathbf{A}_{\mathbf{e}_{vw}} \mathbf{h}_w^t \\ U_t = \text{GRU}(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}) \\ R = \sum_{v \in V} \sigma(i(\mathbf{h}_v^T, \mathbf{h}_v^0)) \odot (j(\mathbf{h}_v^T)) \end{cases} \quad (53)$$



其中,  $\mathbf{A}_{e_{vw}}$  表示邻接矩阵, 每一个元素表示边标签, GRU 表示门控循环单元。读出函数中的  $i$  和  $j$  表示神经网络, 符号  $\odot$  为哈达玛积。

Duvenaud 等<sup>[68]</sup>在用于指纹分子学习的卷积网络中对函数的设置如下:

$$\begin{cases} M(\mathbf{h}_v, \mathbf{h}_w, \mathbf{e}_{vw}) = (\mathbf{h}_w, \mathbf{e}_{vw}) \\ U_t(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}) = \sigma(\mathbf{H}_t^{\deg(v)} \mathbf{m}_v^{t+1}) \\ R = f(\sum_{v \neq t} \text{softmax}(\mathbf{W}_t \mathbf{h}_v^t)) \end{cases} \quad (54)$$

其中, 消息函数中的符号  $(\cdot, \cdot)$  表示向量拼接操作。节点更新函数中的  $\sigma$  表示 sigmoid 激活函数,  $\deg(v)$  表示节点  $v$  的度,  $\mathbf{H}^N$  表示在第  $t$  步迭代次数和节点度数为  $N$  的情况下的学习矩阵。读出函数中的  $f$  表示神经网络,  $\mathbf{W}_t$  表示在第  $t$  步迭代下的学习矩阵。

在 Schütt 等<sup>[69]</sup>提出的深度张量神经网络中, 函数设置如下:

$$\begin{cases} M_t = \tanh(\mathbf{W}^{fc}((\mathbf{W}^{cf} \mathbf{h}_w^t + \mathbf{b}_1) \odot (\mathbf{W}^{df} \mathbf{e}_{vw} + \mathbf{b}_2))) \\ U_t(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}) = \mathbf{h}_v^t + \mathbf{m}_v^{t+1} \\ R = \sum_v NN(\mathbf{h}_v^T) \end{cases} \quad (55)$$

其中,  $\mathbf{W}^{fc}$ ,  $\mathbf{W}^{cf}$  和  $\mathbf{W}^{df}$  表示神经网络中的权重矩阵,  $\mathbf{b}_1$  和  $\mathbf{b}_2$  表示偏置向量。表 4 简单列出了上述函数的设计。

表 4 MPNN 框架下不同 GNN 模型的消息函数和节点更新函数

Table 4 Message function and node update function of different GNN models under MPNN framework

| 模型       | 文献   | 消息函数 $M_t$                                                                                                                       | 更新函数 $U_t$                                          |
|----------|------|----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
| GGNN     | [41] | $\mathbf{A}_{e_{vw}} \mathbf{h}_w^t$                                                                                             | $GRU(\mathbf{h}_v^t, \mathbf{m}_v^{t+1})$           |
| 指纹分子卷积网络 | [68] | $(\mathbf{h}_w, \mathbf{e}_{vw})$                                                                                                | $\sigma(\mathbf{H}_t^{\deg(v)} \mathbf{m}_v^{t+1})$ |
| 深度张量神经网络 | [69] | $\tanh(\mathbf{W}^{fc}((\mathbf{W}^{cf} \mathbf{h}_w^t + \mathbf{b}_1) \odot (\mathbf{W}^{df} \mathbf{e}_{vw} + \mathbf{b}_2)))$ | $\mathbf{h}_v^t + \mathbf{m}_v^{t+1}$               |

4.2 NLNN

为了在深度神经网络中捕获数据间的长距离依赖关系, Wang 等<sup>[66]</sup>根据计算机视觉中经典的非局部均值滤波<sup>[70]</sup>思想, 提出了非局部神经网络 (Non-local Neural Networks, NLNN)。该模型的基本思想是通过非局部操作计算所有位置特征的加权和来求解任意位置的响应。其中, 位置可以来自自序列数据中的时间坐标, 也可以来自图像数据中的空间坐标。NLNN 可看作是不同自注意力方法的统一, 其非局部操作的定义如下:

$$\mathbf{y}_i = \frac{1}{C(\mathbf{x})} \sum_j f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j) \quad (56)$$

其中,  $i$  表示输出位置的索引,  $j$  表示枚举所有可能位置的索引。  $\mathbf{x}$  表示图像、序列或视频数据的输入信号,  $\mathbf{y}$  表示与  $\mathbf{x}$  同维度大小的输出信号。函数  $f$  用于计算位置  $i$  和  $j$  上元素之间的相关度, 函数  $g$  用于对输入信号  $\mathbf{x}$  进行特征变换。因子  $\frac{1}{C(\mathbf{x})}$  用于对结果进行归一化。

从式(56)中不难看出, NLNN 框架的核心在于函数  $f$  和函数  $g$  的设计。对于特征变换函数  $g$ , 为了简便, 可直接使用线性变换函数:

$$g(\mathbf{x}_j) = \mathbf{W}_g \mathbf{x}_j \quad (57)$$

其中,  $\mathbf{W}_g$  表示可学习的权重矩阵。针对函数  $f$ , 主要存在以下 4 种形式:

1) 点积: 函数  $f$  最简单的一种形式可定义为点积操作, 用于直接计算信号  $\mathbf{x}_i$  和  $\mathbf{x}_j$  之间的相似度。文献[71]中首先将信号  $\mathbf{x}_i$  和  $\mathbf{x}_j$  映射到嵌入空间, 然后再计算两者的内积。

$$f(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \mathcal{O}(\mathbf{x}_j) \quad (58)$$

其中,  $\theta(\mathbf{x}_i) = \mathbf{W}_\theta \mathbf{x}_i$ ,  $\mathcal{O}(\mathbf{x}_j) = \mathbf{W}_\mathcal{O} \mathbf{x}_j$  分别表示两个不同的嵌入向量。式(56)中的归一化因子  $C(\mathbf{x}) = N$ , 其中  $N$  表示输入  $\mathbf{x}$  的数量。

2) 高斯函数: 基于非局部均值滤波器和双边滤波器<sup>[72]</sup>, 函数  $f$  可使用高斯函数。具体地, 表达式如下:

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\mathbf{x}_i^T \mathbf{x}_j} \quad (59)$$

其中,  $\mathbf{x}_i^T \mathbf{x}_j$  表示使用点积计算输入信号  $\mathbf{x}_i$  和  $\mathbf{x}_j$  之间的相似度。式(56)中的归一化因子  $C(\mathbf{x}) = \sum_{j \neq i} f(\mathbf{x}_i, \mathbf{x}_j)$ 。

3) 嵌入高斯函数: 作为高斯函数的简单扩展, 嵌入高斯函数是在嵌入空间中计算输入信号  $\mathbf{x}_i$  和  $\mathbf{x}_j$  间的相似度。此时, 函数  $f$  有如下表达式:

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\theta(\mathbf{x}_i)^T \mathcal{O}(\mathbf{x}_j)} \quad (60)$$

其中,  $\theta(\mathbf{x}_i) = \mathbf{W}_\theta \mathbf{x}_i$ ,  $\mathcal{O}(\mathbf{x}_j) = \mathbf{W}_\mathcal{O} \mathbf{x}_j$  分别表示两个不同的嵌入向量。式(56)中的归一化因子  $C(\mathbf{x}) = \sum_j f(\mathbf{x}_i, \mathbf{x}_j)$ 。嵌入高斯函数和点积之间的主要区别在于嵌入高斯存在 softmax 函数, 这可起到激活函数非线性变换的作用。

4) 向量拼接: 除上述方法外, 函数  $f$  还可定义为向量拼接操作:

$$f(\mathbf{x}_i, \mathbf{x}_j) = \text{ReLU}(\mathbf{w}_f^T [\theta(\mathbf{x}_i) \parallel \mathcal{O}(\mathbf{x}_j)]) \quad (61)$$

其中, 符号  $\parallel$  表示向量拼接操作,  $\mathbf{w}_f^T$  是将向量映射到标量的权重参数。

最后, 表 5 详细汇总了 NLNN 框架下不同的函数  $f$  和  $g$  的情况。NLNN 框架的非局部操作属于一种非常灵活的构建块, 可与 CNN 中的卷积层和 RNN 中的循环层结合使用。与通常在神经网络最后使用的全连接层不同, 非局部操作可在神经网络的开始位置使用, 这可结合局部信息和非局部信息, 从而建立一个更丰富的层次结构。

表 5 函数  $f$  和  $g$  的设计

Table 5 Design of functions  $f$  and  $g$

| 函数 $f$ | 函数 $f(\mathbf{x}_i, \mathbf{x}_j)$ 表达式                                                   | 函数 $g(\mathbf{x}_j)$        | 归一化因子 $C(\mathbf{x})$                  |
|--------|------------------------------------------------------------------------------------------|-----------------------------|----------------------------------------|
| 高斯函数   | $e^{\mathbf{x}_i^T \mathbf{x}_j}$                                                        | $\mathbf{W}_g \mathbf{x}_j$ | $\sum_j f(\mathbf{x}_i, \mathbf{x}_j)$ |
| 嵌入高斯函数 | $e^{\theta(\mathbf{x}_i)^T \mathcal{O}(\mathbf{x}_j)}$                                   | $\mathbf{W}_g \mathbf{x}_j$ | $\sum_j f(\mathbf{x}_i, \mathbf{x}_j)$ |
| 点积     | $\theta(\mathbf{x}_i)^T \mathcal{O}(\mathbf{x}_j)$                                       | $\mathbf{W}_g \mathbf{x}_j$ | $N$                                    |
| 向量拼接   | $\text{ReLU}(\mathbf{w}_f^T [\theta(\mathbf{x}_i) \parallel \mathcal{O}(\mathbf{x}_j)])$ | $\mathbf{W}_g \mathbf{x}_j$ | —                                      |

4.3 GN

相比 MPNN 和 NLNN 框架, Battaglia 等<sup>[67]</sup>提出的图网络 (Graph Network, GN) 对 GNN 做出了更一般化的总结, 支持通过简单的构建块来构建复杂的模型架构。本节首先介绍文献[67]给出的图定义, 然后描述 GN 块和计算步骤, 最后对 GN 的基本设计原则进行讲解。

1) 图定义: 根据 GN 框架, 一个图可通过三元组  $G = (u, V, E)$  加以定义。其中,  $u$  表示全局属性。  $V = \{v_i\}_{i=1, N^v}$  表示节点的集合, 其中,  $v_i$  表示节点的属性。  $E = \{(e_k, r_k, s_k)\}_{k=1, N^e}$  表示边的集合, 其中,  $e_k$  表示边的特征,  $s_k$  和  $r_k$  分别表示发送节点和接收节点。



2)GN 块:一个 GN 块包含 3 个更新函数  $\mathcal{O}$  和 3 个聚集函数  $\rho$ :

$$e'_k = \mathcal{O}^e(e_k, v_{r_k}, v_{s_k}, u) \quad (62)$$

$$\bar{e}'_i = \rho^{e \rightarrow v}(E'_i) \quad (63)$$

$$v'_i = \mathcal{O}^v(\bar{e}'_i, v_i, u) \quad (64)$$

$$\bar{e}' = \rho^{e \rightarrow u}(E') \quad (65)$$

$$u' = \mathcal{O}^u(\bar{e}', \bar{v}', u) \quad (66)$$

$$\bar{v}' = \rho^{v \rightarrow u}(V') \quad (67)$$

其中,  $E'_i = \{(e'_k, r_k, s_k)\}_{r_k=i, k=1, N'}$ ,  $V' = \{v'_i\}_{i=1, N'}$ ,  $E' = \bigcup_i E'_i = \{(e'_k, r_k, s_k)\}_{k=1, N'}$ .

3)GN 块的计算步骤:当一个图作为 GN 块的输入时,计算顺序从边级别开始,然后再到节点级别,最后到图级别。具体地,整个计算过程可分为以下 6 步:

(1)更新函数  $\mathcal{O}^e$  作用于每一条边,其参数为  $(e_k, v_{r_k}, v_{s_k}, u)$ ,且该函数的输出为  $e'_k$ 。每个节点  $i$  的每条边的输出集合和每条边的输出的集合分别为:

$$E'_i = \{(e'_k, r_k, s_k)\}_{r_k=i, k=1, N'} \quad (68)$$

$$E' = \bigcup_i E'_i = \{(e'_k, r_k, s_k)\}_{r_k=i, k=1, N'} \quad (69)$$

(2)聚合函数  $\rho^{e \rightarrow v}$  作用在  $E'_i$ ,把投影到节点  $i$  的边的所有边更新聚集到  $\bar{e}'_i$ ,以在下一步节点更新中使用。

(3)更新函数  $\mathcal{O}^v$  作用于每一个节点  $i$ ,以计算更新节点的属性  $v'_i$ 。每个节点输出的结果集合为  $V' = \{v'_i\}_{i=1, N'}$ 。

(4)聚合函数  $\rho^{v \rightarrow u}$  作用在  $V'$ ,把所有边的更新聚合到  $\bar{v}'$ ,以在下一步的全局更新中使用。

(5)聚合函数  $\rho^{e \rightarrow u}$  作用在  $V'$ ,把所有节点的更新聚合到  $\bar{e}'$ ,以在下一步的全局更新中使用。

(6)更新函数  $\mathcal{O}^u$  作用于每一个图上,以更新全局属性  $u'$ 。

图 14 展示了 GN 块的更新示意图。需要注意的是,上面的计算步骤并不是固定不变的,可根据实际的需要进行必要的调整。例如可以将更新函数从全局更新到每个节点的更新再到每条边的更新。

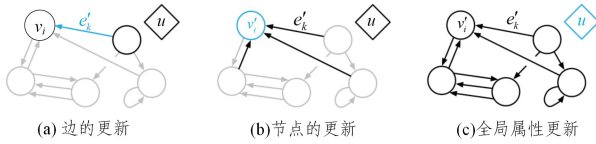


图 14 GN 块更新示意图

Fig. 14 Schematic of GN block update

4)GN 的基本设计原则:GN 的设计基于灵活的表示、可配置的块内结构和可组合的多块架构 3 个基本原则。

(1)灵活的表示。GN 在属性表示和图自身的结构表示方面高度灵活。在属性表示方面,GN 块的全局属性、节点属性和边属性可使用任意的表示形式。在深度学习的实现中,实值向量和张量是最常见的,当然也可使用其他数据结构,例如序列、集合甚至图。GN 块的输出可根据任务的特定需求进行简单的定制,例如 Battaglia 等<sup>[71]</sup>聚焦于节点的输出,Hamrick 等<sup>[73]</sup>聚焦于边的输出,Gilmer 等<sup>[65]</sup>聚焦于整个图的输出。在图结构的自身表示方面,当输入数据为图形结构时,通常有两种情况:一是在具有显式的图数据结构时,就输入明确的指定关系结构;二是在图数据结构的关系不明确时,

必须先推断或假设关系结构,然后再进行输入。

(2)可配置的块内结构。GN 块内的结构和功能可通过不同的方式进行配置,这为哪些信息可作为函数的输入以及如何输出边、节点和全局更新方面提供了灵活性。基于不同的结构和功能设置,GN 框架可实现不同种类的变体模型(如 MPNN、NLNN 以及其他的变体)。

(3)可组合的多块架构。可通过组合 GN 块来构建复杂的架构,并且这些 GN 块可通过共享或非共享的参数进行组织。Battaglia 等<sup>[67]</sup>使用 GN 块构建出了 encode-process-decode 架构和基于 GN 块的循环架构。

## 5 图神经网络的应用领域

### 5.1 推荐系统

推荐系统作为当今人工智能领域最成功的商业应用,其中项目和用户的交互可天然地借助图结构数据进行建模表示,因此十分适合使用图神经网络进行处理。矩阵补全作为推荐系统中的一项重要任务,旨在通过一定的方法先将矩阵的缺失值补充完整,然后基于此为用户进行推荐。Berg 等<sup>[12]</sup>提出的 GCMC 模型作为一种经典的矩阵补全方法,在空域中对节点信息进行聚合和更新。具体地,GCMC 模型采用的是编码器-解码器架构。其中,编码器部分用于获得用户和项目的嵌入表示,解码器部分用于预测用户对项目的评分。为了使矩阵补全方法具有归纳性,之前的许多方法需要依赖辅助信息(如用户的年龄或电影的类型)。然而,在现实生活中,很多高质量的辅助内容总是不可用的,或者说是无法获得的。为了克服这一缺陷,Zhang 等<sup>[74]</sup>提出了 IGMC 模型。该模型可在不使用辅助信息的情况下完成对矩阵补全模型的训练,同时取得与当前先进转换方法类似或更好的性能。

随着全球服务的不断增长,为了能在包含数十亿对象的大规模网络图中学习节点的嵌入表示,Ying 等<sup>[75]</sup>成功将 GCN 应用到推荐系统中,并提出了一种高效模型 PinSage。其整体架构与 GraphSAGE 类似,不同点在于 GraphSAGE 采用的是随机抽样策略,而 PinSage 是通过随机游走生成样本数据。社交推荐作为推荐系统的一个重要分支,主要使用用户自身的社交网络信息来为用户产生推荐。Fan 等<sup>[16]</sup>提出的 GraphRec 模型,通过构建用户和项目之间的异构交互图来分别学习用户和项目的嵌入表示。同时,为了突出不同邻居节点的重要性,GraphRec 模型还使用了注意力机制,从而在聚合节点特征信息时为不同节点分配不同的权重。在图神经网络的自身设计方面,受到简化图卷积网络研究的启发,He 等<sup>[15]</sup>指出移除图卷积神经网络中常用的特征变换和非线性激活对协同过滤并没有太大的影响,由此设计了轻量级图卷积网络模型 LightGCN。除社交推荐外,GNN 还被成功应用于序列推荐<sup>[7-9]</sup>、跨域推荐<sup>[10-12]</sup>、多行为推荐<sup>[13-14]</sup>、捆绑推荐<sup>[15-16]</sup>和会话推荐<sup>[17-18]</sup>领域中,均取得了良好的结果。

### 5.2 自然语言处理

被誉为人工智能皇冠上的明珠的自然语言处理已经随着深度学习的快速发展取得了巨大成功。传统的自然语言处理方法大都基于词袋模型对文本进行表示学习。近年来,随着词嵌入方法的研究,卷积神经网络和循环神经网络已被广泛

应用于自然语言处理中,并取得了十分优异的成绩。但自然语言处理中的很多问题除了可采用序列建模以外,还能够借助图结构数据进行表达。如文本序列中的句子结构信息可通过句法分析树进行表示,然后通过合并特定的任务知识来增强原始数据。该领域在使用图结构数据对实际问题进行建模时,比较常用的有依赖关系图、知识图谱和词共现图等。

为了解决注意力机制和卷积神经网络无法解释情感分类问题中相关句法约束和长距离单词依赖性的问题,Zhang等<sup>[76]</sup>提出在句子的依存关系树上构建 GCN,利用句法信息和单词依存关系来提高基于方面的情感分析任务的准确度。知识图谱作为一种旨在积累和传达真实世界中知识的多关系数据图,在自然语言处理领域中得到了越来越广泛的应用。如 Zhang 等<sup>[77]</sup>基于知识图谱提出了一种距离监督关系提取方法。具体地,他们首先通过知识图谱的嵌入表示来学习类标签之间的隐式关系知识,然后使用图卷积网络学习显式关系知识,最后通过知识感知注意力模块将隐式关系知识和显式关系知识整合到关系提取模型中进行学习训练。Saxena等<sup>[78]</sup>提出了嵌入式知识图谱问答模型,用于解决知识图谱多跳问答任务中常见的知识图谱不完整性问题。

除了依赖关系图和知识图谱以外,词共现图也是自然语言处理中的一种常用图,主要用于文本分类任务。如 Peng等<sup>[79]</sup>提出了一个基于深度学习的 graph-CNN 模型。该模型首先基于词共现网络将文本转为词图,然后使用图卷积操作对词图进行卷积,从而实现了大规模分层文本分类任务。Yao 等<sup>[80]</sup>首先基于词共现和文档词关系构建语料库的单个文本图,然后使用图卷积神经网络对文本进行分类,在不使用外部知识的情况下,取得了较好的实验结果。与之前的序列建模方法不同,图神经网络能挖掘文本语言中复杂的语义关系,从而在自然语言处理领域展现出了优越的性能。

### 5.3 计算机视觉

CNN 凭借局部性和平移不变性已经在计算机视觉领域取得了十分出色的成绩。然而,基于 CNN 的方法大多缺乏视觉数据间的细粒度分析。例如图像可表示为空间地图,图中的不同区域往往在空间和语义上是相互关联的。因此可将区域抽象为图中的节点,区域之间的依赖关系借助图中的边进行表示。进一步地,使用 GNN 来捕获图像中细粒度关系,从而提高机器视觉中不同任务的性能。

鉴于 GNN 强大的特征表示能力,它已经被成功应用于计算机视觉的很多实际问题(如图像分类、目标检测和视觉推理等)中。在图像分类任务中,零样本学习和少样本学习已经越来越流行。其中,知识图谱可作为辅助信息来指导图片的零样本识别分类任务。如 Wang 等<sup>[81]</sup>通过构建知识图谱,将每个节点视为一个目标类,并利用节点的词嵌入作为模型的输入,从而预测不同的类别。该方法在图像分类中取得了较好的实验结果,但由于使用了多层图神经网络,引入了常见的过平滑问题。针对该问题,Kampffmeyer 等<sup>[34]</sup>提出了 DGP 模型,成功应用了较大邻域的单层 GCN,在该邻域中包含图中的单跳和多跳节点,从而有效解决了由拉普拉斯平滑性所造成的远程节点知识被稀释的问题,并且该方法验证了其于现有的零样本分类器的优越性。针对少样本学习问题,Gar-

cia 等<sup>[82]</sup>基于图像相似性构建了一个加权全连接图像网络,在图上进行消息传递,从而应用于少样本学习任务。

目标检测旨在从给定图像中定位某一具体的视觉对象实例。随着 GNN 的逐渐流行,现在该网络也逐渐应用于目标检测任务中。如 Shi 等<sup>[83]</sup>提出了一个在雷达点云中目标检测的图神经网络模型 Point-GNN,旨在预测图中每个节点所属对象的类别和形状。视觉推理是计算机视觉的一个重要研究方向,由于在推理过程中需要空间和语义信息,因此天然适合使用图进行处理。视觉问答作为视觉推理的一项典型任务,已经吸引了很多研究者的目光。例如,Teney 等<sup>[84]</sup>为了在视觉和文本领域进行联合推理,分别构建了图像场景图和问题语义图,之后借助门控图神经网络训练最终答案的预测表示。在计算机视觉中,GNN 与 CNN 的不同之处在于可深层次地挖掘数据之间更细粒度的语义关系,从而对计算机视觉中的不同任务进行建模和学习。

## 6 研究与展望

近年来,虽然 GNN 凭借着强大的特征表示能力已经在众多领域取得了不错的成绩,但总体来说,GNN 的发展仍然不够完善,还存在许多问题亟待解决。因此,本章从 5 个方面对 GNN 未来的研究趋势进行了总结。

1)深层次网络模型:传统的深度学习能取得良好的性能,是通过堆叠多层甚至百层的网络结构实现的。因为网络结构越深,参数数量就越多,表示学习的能力也就越强。但在图神经网络中,层次往往是比较浅显的,大多数不超过 3 层。因为过深的网络层次往往会引入过平滑性问题,从而导致模型性能下降。因此,如何突破图神经网络的层次限制,设计出具有更深网络层的 GNN 是未来研究中的一项重要课题。

2)动态图:目前 GNN 模型在对图数据进行处理时,基本都停留在静态图的研究上,很少涉及动态图。与具有稳定结构的静态图不同,动态图中节点和边的状态是可跟随时间变化的,当图中的节点和边出现或消失时,GNN 模型若不能自适应地变化,就会给后续图数据的建模工作带来很大的挑战。因此,如果 GNN 模型可应用到动态图上,则能解决现有模型无法解决的很多现实问题。这是一个非常有前景的研究方向。

3)对抗攻击:深度神经网络虽然在诸多领域中取得了杰出的成绩,但是仍然存在训练不稳定的问题,这在生成对抗网络中尤为明显。例如在原始图片上增加轻微的扰动,虽然在人眼看来并没有发生变化,但是神经网络模型可能会将其判别为其他类型。此现象引发了人们对神经网络对抗攻击的思考与研究。同样,在图神经网络的研究中,设计出可抵抗外界攻击的鲁棒性模型为研究者指明了下一步的研究方向。

4)可扩展性:目前对图神经网络的研究还仅限于小规模图,对于需要经常处理大规模图的工业应用来说,模型的可扩展性已经成为了重大限制。如 Twitter 等社交网络平台的节点数目可达百万个,边的数目也可达数十亿,要想在这种大规模数据环境下进行信息的聚合与更新是十分耗时的,而且计算代价非常高。因此,如何将现有的模型扩展到工业级别的大规模图上已经成为了一个炙手可热的研究问题,

这关乎当前的算法能否真正实现工业落地,促进经济快速向前发展。

5)可解释性:深度学习一直以来就像黑盒一样存在可解释性低的问题,GNN 作为深度学习的一种,也不例外。将 GNN 模型应用于具有可解释性的实际应用程序非常重要,但目前仅有少数工作对该问题进行了研究。因此,图的可解释性是一个十分重要的研究方向。

**结束语** 近年来,图神经网络凭借在非欧几里得空间中强大的特征表示能力,已经成为深度学习领域最流行的技术之一。通过阅读图神经网络方面的大量文献,本文对图神经网络的相关技术进行了系统总结并对其未来发展方向进行了展望。

具体地,本文第 1 章介绍了与图相关的基本知识,第 2 章先从图嵌入表示学习的角度着手,对图游走算法进行了有关记述,之后从谱域和空域两个角度详细介绍了 GNN 中的卷积神经网络。进一步地,还叙述了其他类型的 GNN 模型,如图注意力网络、图递归网络和图残差网络。第 3 章和第 4 章分别对 GNN 中的通用范式和应用领域进行了细致描述。最后指出了 GNN 未来的发展方向。

图神经网络由于能够处理图形等不规则的数据,已经成为神经网络家族中强有力的工具。目前对图神经网络的研究可谓是百花齐放,出现了可应用于不同任务的模型,并且取得了不错的成绩。尽管有些模型还存在一些缺陷亟待解决,但相信在不久的将来,这些问题终会得到解决,届时人工智能的发展又会达到一个新的高峰。

参 考 文 献

[1] GU J, WANG Z, KUEN J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018, 77: 354-377.

[2] MEDSKER L R, JAIN L C. Recurrent neural networks[J]. Design and Applications, 2001, 5: 64-67.

[3] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

[4] GORI M, MONFARDINI G, SCARSELLI F. A new model for learning in graph domains [C]// 2005 IEEE International Joint Conference on Neural Networks. New York: IEEE, 2005: 729-734.

[5] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model[J]. IEEE Transactions on Neural Networks, 2008, 20(1): 61-80.

[6] NIEPERT M, AHMED M, KUTZKOV K. Learning convolutional neural networks for graphs[C]// International Conference on Machine Learning. New York: ACM, 2016: 2014-2023.

[7] GAO H, WANG Z, JI S. Large-scale learnable graph convolutional networks[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2018: 1416-1424.

[8] ATWOOD J, TOWSIEV D. Diffusion-convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2016, 9: 2001-2009.

[9] PEROZZI B, AL-RFOU R, SKIENA S. Deepwalk: Online learning of social representations [C]// Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2014: 701-710.

[10] GROVER A, LESKOVEC J. node2vec: Scalable feature learning for networks[C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016: 855-864.

[11] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [J]. arXiv: 1310. 4546, 2013.

[12] BERG R, KIPF T N, WELLING M. Graph convolutional matrix completion[J]. arXiv: 1706. 02263, 2017.

[13] VELICKOVIC P, CUCURULL G, CASANOVA A, et al. Graph attention networks[J]. arXiv: 1710. 10903, 2018.

[14] HAMILTON W, YING Z, LESKOVEC J. Inductive representation learning on large graphs[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017: 1025-1035.

[15] HE X, DENG K, WANG X, et al. Lightgcn: Simplifying and powering graph convolution network for recommendation[C]// Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2020: 639-648.

[16] FAN W, MA Y, LI Q, et al. Graph neural networks for social recommendation[C]// The world Wide Web Conference. New York: ACM, 2019: 417-426.

[17] LIU F, CHENG Z, ZHU L, et al. An attribute-aware attentive GCN model for attribute missing in recommendation[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(9): 4077-4088.

[18] WANG X, HE X, CAO Y, et al. Kgat: Knowledge graph attention network for recommendation[C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2019: 950-958.

[19] FAN S, ZHU J, HAN X, et al. Metapath-guided heterogeneous graph neural network for intent recommendation [C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2019: 2478-2486.

[20] VASHISHTH S, BHANDARI M, YADAV P, et al. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks [J]. arXiv: 1809. 04283, 2018.

[21] WANG K, SHEN W, YANG Y, et al. Relational graph attention network for aspect-based sentiment analysis [J]. arXiv: 2004. 12362, 2020.

[22] LIU Z, XIONG C, SUN M, et al. Fine-grained fact verification with kernel graph attention network [J]. arXiv: 1910. 09796, 2019.

[23] WANG D, LIU P, ZHENG Y, et al. Heterogeneous graph neural networks for extractive document summarization [J]. arXiv: 2004. 12393, 2020.

[24] CHEN Z M, WEI X S, WANG P, et al. Multi-label image recognition with graph convolutional networks[C]// Proceedings of



- the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York:IEEE,2019:5177-5186.
- [25] GAO J,ZHANG T,XU C. I know the relationships:Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI, 2019, 33(1):8303-8311.
- [26] HAN K,WANG Y,GUO J,et al. Vision GNN: An Image is Worth Graph of Nodes[J]. arXiv:2206. 00272,2022.
- [27] ZHANG Z,SHI Y,YUAN C,et al. Object relational graph with teacher-recommended learning for video captioning[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York:IEEE,2020:13278-13288.
- [28] MOHAMED A,QIAN K,ELHOSEINY M,et al. Social-stgcn: A social spatio-temporal graph convolutional neural network for human trajectory prediction[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York:IEEE,2020:14424-14432.
- [29] HAN K,LAKSHMINARAYANNA B,LIU J. Reliable graph neural networks for drug discovery under distributional shift [J]. arXiv:2111. 12951,2021.
- [30] ZHAO H,WANG Y,DUAN J,et al. Multivariate time-series anomaly detection via graph attention network[C]// 2020 IEEE International Conference on Data Mining (ICDM). New York: IEEE,2020:841-850.
- [31] DENG A,HOOI B. Graph neural network-based anomaly detection in multivariate time series[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI,2021, 35(5):4027-4035.
- [32] CUI Z,LI Z,WU S,et al. Dressing as a whole:Outfit compatibility learning based on node-wisegraph neural networks[C]// The World Wide Web Conference. New York:ACM,2019:307-317.
- [33] TONG Z,LIANG Y,SUN C,et al. Directed graph convolutional network[J]. arXiv:2004. 13970,2020.
- [34] KAMPPFMEYER M,CHEN Y,LIANG X,et al. Rethinking knowledge graph propagation for zero-shot learning[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York:IEEE,2019:11487-11496.
- [35] DUAN D,LI Y,JIN Y,et al. Community mining on dynamic weighted directed graphs[C]// Proceedings of the 1st ACM International Workshop on Complex Networks Meet Information & Knowledge Management. New York:ACM,2009:11-18.
- [36] ZHANG Y,XIONG Y,KONG X,et al. Deep collective classification in heterogeneous information networks[C]// Proceedings of the 2018 World Wide Web Conference. New York: ACM, 2018:399-408.
- [37] LI Y,YU R,SHAHABI C,et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting[J]. arXiv: 1707. 01926,2017.
- [38] YAN S,XIONG Y,LIN D. Spatial temporal graph convolutional networks for skeleton-based action recognition [C] // Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI,2018.
- [39] HUANG Y,XU H,DUAN Z,et al. Modeling complex spatial patterns with temporal features via heterogenous graph embedding networks[J]. arXiv:2008. 08617,2020.
- [40] BECK D,HAFFARI G,COHN T. Graph-to-sequence learning using gated graph neural networks[J]. arXiv: 1806. 09835, 2018.
- [41] LI Y,TARLOW D,BROCKSCHMIDT M,et al. Gated graph sequence neural networks[J]. arXiv:1511. 05493,2015.
- [42] ABDI H,WILLIAMS L J. Principal component analysis[J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(4):433-459.
- [43] BLEI D M,NG A Y,JORDAN M I. Latent dirichlet allocation [J]. Journal of machine Learning Research, 2003, 3 (Jan): 993-1022.
- [44] CARDOSO J F,COMON P. Independent component analysis,a survey of some algebraic methods[C]// 1996 IEEE International Symposium on Circuits and Systems,Circuits and Systems Connecting the World(ISCAS 96). New York:IEEE,1996:93-96.
- [45] TANG J,QU M,WANG M,et al. Line:Large-scale information network embedding[C]// Proceedings of the 24th International Conference on World Wide Web. New York:ACM,2015:1067-1077.
- [46] ERUNA J,ZAREMBA W,SZLAM A,et al. Spectral networks and deep locally connected networks on graphs[C]// 2nd International Conference on Learning Representations. 2014.
- [47] MA S,LIU J W,ZUO X. Survey on Graph Neural Network[J]. Journal of Computer Research and Development, 2022, 59(1): 47-80.
- [48] DEFFERRARD M,BRESSON X,VANDERGHEYNST P. Convolutional neural networks on graphs with fast localized spectral filtering[C]// Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016:3844-3852.
- [49] KIPF T N,WELLING M. Semi-supervised classification with graph convolutional networks[J]. arXiv:1609. 02907,2016.
- [50] LI R,WANG S,ZHU F,et al. Adaptive graph convolutional neural networks[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Menlo Park: AAAI,2018.
- [51] XU B,SHEN H,CAO Q,et al. Graph wavelet neural network [J]. arXiv:1904. 07785,2019.
- [52] MICHELI A. Neural network for graphs:A contextual constructive approach[J]. IEEE Transactions on Neural Networks, 2009, 20(3):498-511.
- [53] BAHDANAU D,CHO K,BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv: 1409. 0473,2014.
- [54] GEHRING J,AULI M,GRANGIER D,et al. A convolutional encoder model for neural machine translation[J]. arXiv:1611. 02344,2016.
- [55] VASWANI A,SHAZEER N,PARMAR N,et al. Attention is all you need[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.
- [56] CHENG J,DONG L,LAPATA M. Long short-term memory-networks for machine reading[J]. arXiv:1601. 06733,2016.
- [57] ZHANG J,SHI X,XIE J,et al. Gaan:Gated attention networks for learning on large and spatiotemporal graphs[J]. arXiv:1803.



07294,2018.

[58] 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. New York: IEEE, 2016:770-778.

[59] LI G,MULLER M,THABET A,et al. Deepgcns:Can gcns go as deep as cnns? [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. New York:IEEE,2019: 9267-9276.

[60] HUANG G,LIU Z,VAN DER MAATEN L,et al. Densely connected convolutional networks [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York:IEEE,2017:4700-4708.

[61] YU F,KOLTUN V. Multi-scale context aggregation by dilated convolutions[J]. arXiv:1511. 07122,2015.

[62] ZHENG Y,GAO C,CHEN L,et al. Dgcnn:Diversified recommendation with graph convolutional networks[C]//Proceedings of the Web Conference 2021. New York:ACM,2021:401-412.

[63] LIANG X,SHEN X,FENG J,et al. Semantic object parsing with graph lstm[C]//Computer Vision—ECCV 2016:14th European Conference, Amsterdam, The Netherlands, Part I 14. Amsterdam:Springer,2016:125-143.

[64] WU Z,PAN S,CHEN F,et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems,2020,32(1):4-24.

[65] GILMER J,SCHOENHOLZ S S,RILEY P F,et al. Neural message passing for quantum chemistry[C]//International Conference on Machine Learning. PMLR,2017:1263-1272.

[66] WANG X,GIRSHICK R,GUPTA A,et al. Non-local neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:7794-7803.

[67] BATTAGLIA P W,HAMRICK J B,BAPST V,et al. Relational inductive biases,deep learning, and graph networks[J]. arXiv: 1806. 01261,2018.

[68] DUVENAUD D K,MACLAURIN D,IPARRAGUIRRE J, et al. Convolutional networks on graphs for learning molecular fingerprints[C]// Proceedings of the 28st International Conference on Neural Information Processing Systems December. 2015:2224-2232.

[69] SCHUTT K T,ARBABZADAH F,CHMIELA S,et al. Quantum-chemical insights from deep tensor neural networks[J]. Nature Communications,2017,8(1):1-8.

[70] BUADES A,COLL B,MOREL J M. A non-local algorithm for image denoising[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'05). New York:IEEE,2005:60-65.

[71] BATTAGLIA P,PASCANU R,LAI M,et al. Interaction networks for learning about objects,relations and physics[C]// Proceedings of the 28st International Conference on Neural Information Processing Systems December. 2016:4509-4517.

[72] TOMASI C,MANDUCHI R. Bilateral filtering for gray and color images[C]//Sixth international Conference on Computer Vision. New York:IEEE,1998:839-846.

[73] HAMRICK J B,ALLEN K R,BAPST V,et al. Relational inductive bias for physical construction in humans and machines [J]. arXiv:1806. 01203,2018.

[74] ZHANG M,CHEN Y. Inductive matrix completion based on graph neural networks[J]. arXiv:1904. 12058,2019.

[75] YING R,HE R,CHEN K,et al. Graph convolutional neural networks for web-scale recommender systems[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York:ACM,2018:974-983.

[76] ZHANG C,LI Q,SONG D. Aspect—based sentiment classification with aspect-specific graph convolutional networks[J]. arXiv:1909. 03477,2019.

[77] ZHANG N,DENG S,SUN Z,et al. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks[J]. arXiv:1903. 01306,2019.

[78] SAXENA A,TRIPATHI A,TALUKDAR P. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 4498-4507.

[79] PENG H,LI J,HE Y,et al. Large-scale hierarchical text classification with recursively regularized deep graph-cnn[C]// Proceedings of the 2018 World Wide Web Conference. 2018:1063-1072.

[80] YAO L,MAO C,LUO Y. Graph convolutional networks for text classification[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019,33(1):7370-7377.

[81] WANG X,YE Y,GUPTA A. Zero-shot recognition via semantic embeddings and knowledge graphs [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York:IEEE,2018:6857-6866.

[82] GARCIA V,BRUNA J. Few-shot learning with graph neural networks[J]. arXiv:1711. 04043,2017.

[83] SHI W,RAJKUMAR R. Point-gnn:Graph neural network for 3d object detection in a point cloud[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York:IEEE,2020:1711-1719.

[84] TENEY D,AMDERSON P,HE X,et al. Tips and tricks for visual question answering: Learnings from the 2017 challenge [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:4223-4232.



**HOU Lei**, born in 1998, postgraduate, is a member of CCF (No. Q7342G). His main research interests include recommendation system, graph neural networks and information retrieval.



**LIU Jinhuan**, born in 1989, Ph.D, is a member of CCF(No. I1628M). Her main research interests include machine learning, recommendation system, and information retrieval.