

## 自注意力超图池化网络\*

赵英伏, 金福生, 李荣华, 秦宏超, 崔鹏, 王国仁

(北京理工大学 计算机学院, 北京 100081)

通信作者: 金福生, E-mail: jfs21cn@bit.edu.cn



**摘要:**近年来,将卷积神经网络推广到图数据上的图卷积神经网络引起了广泛关注,主要包括重新定义图的卷积和池化操作.由于图数据只能表达二元关系的局限性,使其在实际应用中表现欠佳.相比之下,超图能够捕获数据的高阶相关性,利用其灵活的超边易于处理复杂的数据表示.然而,现有的超图卷积神经网络还不够成熟,目前尚无有效的超图池化操作.因此,提出了带有自注意机制的超图池化网络,使用超图结构建模,通过引入自注意力的超图卷积操作学习带有高阶数据信息的节点隐藏层特征,再经过超图池化操作选择并保留在结构和内容上的重要节点,进而得到更准确的超图表示.在文本分类、菜肴分类和蛋白质分类任务上的实验结果表明:与目前多种主流方法相比,该方法均取得了更好的效果.

**关键词:**超图;卷积神经网络;池化;图神经网络;超图神经网络

**中图法分类号:**TP311

中文引用格式:赵英伏,金福生,李荣华,秦宏超,崔鹏,王国仁.自注意力超图池化网络.软件学报,2023,34(10):4463-4476. <http://www.jos.org.cn/1000-9825/6881.htm>

英文引用格式:Zhao YF, Jin FS, Li RH, Qin HC, Cui P, Wang GR. Self-attention Hypergraph Pooling Network. Ruan Jian Xue Bao/Journal of Software, 2023, 34(10): 4463-4476 (in Chinese). <http://www.jos.org.cn/1000-9825/6881.htm>

## Self-attention Hypergraph Pooling Network

ZHAO Ying-Fu, JIN Fu-Sheng, LI Rong-Hua, QIN Hong-Chao, CUI Peng, WANG Guo-Ren

(School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

**Abstract:** Recently, graph-based convolutional neural networks (GCNs) have attracted much attention by generalizing convolutional neural networks to graph data, which includes redefining the convolution and the pooling operations on graphs. Due to the limitation that graph data can only focus on dyadic relations, it cannot perform well in real practice. In contrast, a hypergraph can capture high-order data interaction and is easy to deal with complex data representation using its flexible hyperedges. Nevertheless, the existing methods for hypergraph convolutional networks are still not mature, currently there is no effective operation for hypergraph pooling. Therefore, a hypergraph pooling network with self-attention mechanism is proposed. Using hypergraph structure for data modeling, this model can learn hidden node features with high-order data information through hyper-convolution operation which introduces self-attention mechanism, select important nodes both on structure and content through hyper-pooling operation, and then obtain more accurate hypergraph representation. Experiments on text classification, dish classification, and protein classification tasks show that the proposed method outperforms recent state-of-the-art methods.

**Key words:** hypergraph; convolutional neural network; pooling; graph neural network; hypergraph neural network

卷积神经网络(convolutional neural networks, CNNs)成功地利用了欧几里德域上的网格结构数据,在图像、语音和视频等领域中表现良好<sup>[1-3]</sup>.近年来,图卷积神经网络(graph neural networks, GNNs)<sup>[4]</sup>收获了很多关注.与传统的卷积神经网络不同的是:基于图的卷积神经网络能够处理非欧几里德域的图数据,使得它在

\* 基金项目:国家重点研发计划(2020AAA0108503);国家自然科学基金(62272045)

本文由“知识赋能的信息系统”专题特约编辑高宏教授、陈华钧教授、赵翔教授、李瑞轩教授推荐.

收稿时间:2022-06-28;修改时间:2022-08-18,2022-12-14;采用时间:2022-12-28;jos 在线出版时间:2023-01-13

推荐系统、化学研究、自然语言处理等领域的任务中显示出优于传统卷积神经网络的效果。

图卷积神经网络是将卷积操作扩展到图结构上,而池化作为卷积神经网络的另一个主要组成部分,该操作能够通过缩小特征信息的大小来减少参数的数量,从而避免过拟合并得到更好的特征表示效果.因此,在图表示任务中,图池化操作也是十分必要的.例如在生物领域中,图可以表示蛋白质结构,其中一个节点表示一个氨基酸,如果两个节点的距离小于  $6\text{\AA}$ ,则构成一条边,标签表示蛋白质的类别是酶还是非酶.对于这样的蛋白质分类任务,如果没有池化操作,只是简单地聚合所有氨基酸节点特征来表示蛋白质,难以实现准确的特征表示,分类效果也并不理想.然而,目前针对图池化的研究相比于图卷积要少很多.起初采用的是只考虑图拓扑的池化方法,之后有人利用节点特征来获得图的表示.最近提出的一种自注意图池化(self-attention graph pooling, SAGPool)<sup>[5]</sup>方法,使用图卷积来计算注意分数,据此选择保留或删除的节点.自适应图池化(graph self-adaptive pooling, GSAPool)<sup>[6]</sup>方法基于 SAGPool 做了进一步的改进,同时考虑图的结构和特征信息,构造更合理的图池化拓扑结构,并使池化节点包含更有效的图信息.

目前,这些池化方法都是建立在图数据的基础上,在实际应用中除了两两关系之外,数据关系建模的情况可能更加复杂.如多模态、社交媒体、自然语言处理、计算机视觉等任务场景中,存在大量简单图难以建模的非两两关系,使得传统图结构存在局限性. Zhou 等人提出的超图<sup>[7]</sup>是一种能够对非二元关系建模的结构,有了超图的概念,生活中很多数据就能够得到更合理的表示.例如在图 1 所示的论文作者的网中,作者 A-E 这 5 人合作发表了一篇论文,如果用普通图建模,如图 1(a)所示,需要 5 名作者顶点两两相连.这其中存在两个问题:首先,仅仅表示这一个事件就需要 10 条边,这意味着构建更多的合作关系将需要更多的边数;其次,每条边连接两个作者顶点,只能表示两个作者有合作关系,难以描述 5 人共同合作的信息.而如果建模成超图,如图 1(b)所示,只需要一条超边即可简洁、清楚地描述这种关系.因此,由于超图具有能够表示高阶信息的优势,如何将普通图的卷积和池化操作迁移到超图结构中,成为了进一步研究的热点.



图 1

然而,目前尚无有效的超图池化方法,这使得超图分类的任务不能实现很好的效果.例如:在文本分类任务中,采用超图建模能够更好地表示文本数据.即:一个节点表示一个单词,一条超边表示一个句子,连接句子中所有的单词节点.这样,一个文本就能够用一个超图来表示.使用文本中所有单词节点的特征来表示文本特征显然并不是最佳的表示方法,因为有些单词对于文本的类别判定意义不大.因此,针对超图池化操作的研究同样十分重要.

针对这一问题,本文提出了一种基于自注意的超图池化网络,进一步完善了超图卷积操作,并提出了合理、有效的超图池化操作:首先,复杂的数据关系以超图结构表达,采用自注意机制的方式计算任意节点对之间的关联度,进一步聚合得到超边权重,经过带有超边权重的超图拉普拉斯卷积层得到更新的节点特征;接下来送入超图池化层,采用 MLP 评分函数得到每个节点的重要度评分,保留评分最高的  $k$  个节点,并更新得到池化超图;最后,将池化超图送入读出层读取最大值和平均值特征,得到最终的超图表示.我们的模型能够在自注意的超图卷积层更好地捕获超图中的高阶信息,在超图池化层准确、合理地筛选重要信息,在读出层更好地汇聚有效信息,并对超图进行表示,因而在超图分类任务中拥有良好的性能.

总之,本文的贡献可以归纳为以下 3 点.

- (1) 据我们所知,我们的模型是第一个提出的超图池化网络.该网络是一个归纳式模型,在模型训练时无需加载测试数据信息,也无需引入其他外部信息,因此能够以较少的参数和内存消耗进行有效的

超图表示学习;

- (2) 我们设计了基于自注意力的超图卷积操作, 利用自注意机制计算超边权重, 进而更准确地表示超图拉普拉斯矩阵, 在卷积操作中更好地让信息从节点传播到边, 再传播回节点;
- (3) 我们在文本分类任务上进行了大量的实验, 与最新方法的比较证明了所提出模型的有效性. 实验结果表明, 该模型在文本分类任务中具有良好的性能. 说明将文本建模成超图, 并采用带注意力的超图卷积和池化网络得到超图特征表示的方法是有效的, 这也给文本分类提供了新的解决思路.

本文第1节介绍图池化和超图神经网络的相关工作和研究现状. 第2节介绍本文提出的超图池化模型. 第3节通过对比实验验证所提模型的有效性. 最后总结全文.

## 1 相关工作

### 1.1 图池化

池化通过缩小表示信息的大小来减少参数的数量, 能够避免过拟合并得到更好的表示效果. 因此在图表示学习的任务中, 图池化操作是十分必要的. 现阶段图池化方法可以分为以下几类: 基于拓扑的池化、基于图粗化的池化和基于节点选择的池化.

- 基于拓扑的池化.

早期工作使用的是基于拓扑的谱聚类算法, 利用特征分解得到粗化图. 由于一般谱聚类目标和加权核  $k$ -均值目标在数学上是等价的, Graclus 算法<sup>[8]</sup>具有不需要特征分解的优点, 大大降低了时间复杂度, 成为一种有效的谱聚类方法, 并在最近的图神经网络模型<sup>[9,10]</sup>中被用作池化模块.

- 基于图粗化的池化.

Ying 等人提出的 DiffPool<sup>[11]</sup>是这类工作的典型代表, 首先对节点进行聚类合成多个节点簇, 进而更新池化图的拓扑结构, 直到将整个图合成为一个超级节点. 具体来说, DiffPool 在深度 GNNs 的每一层学习分配矩阵, 利用分配矩阵生成节点簇. 分配矩阵的每一列对应池化前的节点, 每一行对应池化后的节点. 每层学习的分配矩阵包含了该层节点被分配给下一层簇的概率.

- 基于节点选择的池化.

最先使用节点选择机制的是 Cangea 等人提出的 gPool<sup>[12]</sup>, 该方法使用一个投影向量  $p$  来计算节点分数, 据此选择并保留  $k$  个分数排名最高的节点, 池化图的拓扑结构由保留节点决定. gPool 的池化过程如下:

$$y = Xp / \|p\|, \text{idx} = \text{top\_rank}(y, [kN]) \quad (1)$$

$$A' = A_{\text{idx}, \text{idx}} \quad (2)$$

其中,  $y$  是所有节点的得分向量,  $X$  和  $p$  是节点特征矩阵与投影向量,  $A'$  是池化图的邻接矩阵.

为了改进 gPool 在选择节点时未考虑图的拓扑结构的缺陷, SAGPool 使用一维图卷积进行节点评分, 同时考虑了节点特征和拓扑结构. GSApool 进一步改进评分机制, 使用两种评分方式来综合节点重要性评分, 并且避免了未选择节点的特征信息直接丢失, 提升了图的特征表示能力.

### 1.2 超图卷积网络

由于超图可以对非两两关系进行建模, 进而编码高阶的数据相关性, 近年来对超图的研究得到了越来越广泛的关注. 然而, 传统图的谱卷积理论不能直接迁移到超图结构中. 以往的超图卷积方法都需要拉普拉斯的逆运算, 这使得计算效率很低. 超图卷积网络(hypergraph neural network, HGNN)<sup>[13]</sup>是一种不需要拉普拉斯逆运算的方法, 该方法定义拉普拉斯矩阵为

$$\Delta = I - D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} \quad (3)$$

其中,  $D_v$  和  $D_e$  分别表示顶点度和超边度的对角矩阵, 起到归一化的作用;  $H$  是超图的关联矩阵;  $W$  是超边权重的对角矩阵(在论文作者实现的版本中,  $W$  是一个全一矩阵). 进而, 超图卷积过程可以表示为

$$Y = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} X \Theta \quad (4)$$

其中,  $\Theta$ 是可学习参数,  $X$ 和  $Y$ 分别是卷积前后的节点特征矩阵. 上式表示对超图中的节点特征  $X$ 应用滤波器  $\Theta$ 提取特征的过程.

除了 HGNN 这样的谱卷积方法外, 还有很多非谱卷积方法<sup>[14,15]</sup>. 然而, 现有的方法都只关注节点级表示, 并没有有效的方法可以对整个超图进行表示学习.

### 1.3 团嵌入与超图嵌入

团和超图的主要区别在于: 一个团内包含的节点关联度较高, 节点特征属性较为相似, 不同的团之间节点联系较弱, 差距较大; 而在超图表示中, 节点之间并非都是结构高度关联的, 节点嵌入特征之间也可能存在很大的差异, 因此超图所表示的内容范围更广、包容性更强.

然而, 以往对于团的研究主要聚焦在如何从大型图中挖掘团以及根据图的结构特征或节点嵌入特征进行聚类得到团等工作. 目前的工作存在如下问题.

- (1) 不论是由传统图算法挖掘得到团, 还是基于神经网络的方法聚类得到团, 目前这类工作的任务在于团的挖掘, 而并没有过多的研究团的准确的嵌入表示, 并且得到的团所表示的实际含义难以明确, 进而难以对团进行类别标签的判定, 无法适用于分类任务;
- (2) 对于既定团的嵌入该如何表示, 目前并没有本质上的创新方法. 由于团内节点的高度相关性和团之间的互斥性, 团嵌入可以简单地基于团内节点嵌入的平均值等操作得到. 而当节点之间特征存在较大差异时, 这样的方法并不是最有效的;
- (3) 目前, 针对超图中的团算法均采用将超图规约成图的方式进行, 而非直接针对超图本身. 基于以上分析和对比, 团和超图在特征上存在较大差别, 团嵌入的相关工作也并不能直接适用或拓展到图和超图分类任务中, 针对超图的池化工作的研究是十分必要的.

针对以上相关工作的研究现状, 本文提出的超图池化网络可以更好地表示超图, 进而应用于姿态估计、蛋白质分类、图像分类、文本分类等诸多领域的超图分类任务中.

## 2 自注意力超图池化网络

本节将介绍一种用于超图分类的自注意力超图池化网络模型. 首先, 我们介绍对模型输入(超图)的定义; 然后, 详细介绍模型的 3 个主要模块及整体框架(如图 2 所示).

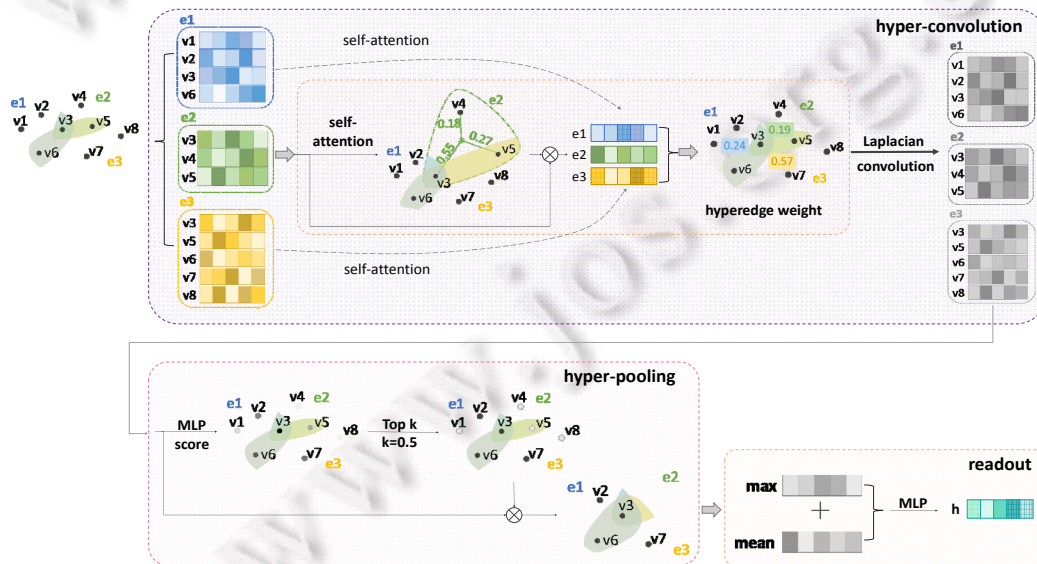


图 2 超图池化网络, 分为超图卷积层、超图池化层和读出层

图示输入超图包含 8 个节点和 3 条超边, 首先, 通过自注意机制得到超边权重, 经过超图拉普拉斯矩阵卷积后得到更新的节点特征; 在超图池化模块, 首先通过 MLP 得到评分, 保留评分最高的  $k$  个节点, 并更新得到池化后的超图; 送入读出层读取最大值和平均值特征, 进而得到最终的超图表示。

### 2.1 超图定义

超图由一个顶点集和一个超边集组成, 超图  $G$  可以定义为  $G=(V;E)$ , 其中,  $V$  是顶点集,  $E$  是超边集. 与传统图不同的是, 超图中每一个超边可以连接两个或多个顶点, 如图 3 所示。

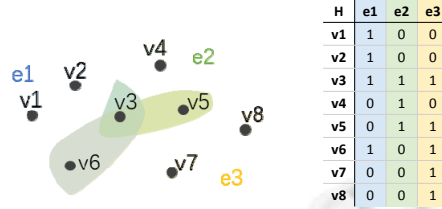


图 3 由 8 个节点 3 条超边构成的超图及其关联矩阵

超图  $G$  也可以用关联矩阵  $H \in \mathbb{R}^{|V| \times |E|}$  表示,  $H$  中的每一项定义为

$$H(v, e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e \end{cases} \quad (5)$$

此外, 顶点  $v \in V$  的度数定义为  $d(v) = \sum_{e \in E} H(v, e)$ , 超边  $e \in E$  的度数定义为  $d(e) = \sum_{v \in V} H(v, e)$ .

### 2.2 超图卷积层

我们的超图卷积层借鉴了 HGNN 中提到的超图拉普拉斯矩阵, 并在此基础上加入了带有自注意机制的超边权重, 能够更好地关注到重要的节点特征信息和结构信息, 过程如图 2 中 hyper-convolution 模块所示。

该层的输入首先是超图中所有节点的特征矩阵  $X$ ,  $X = \{x_1, x_2, \dots, x_n\}$ ,  $x_i \in \mathbb{R}^{d_i}$ , 其中,  $n$  为节点数,  $d_i$  为每个节点的输入特征维度,  $x_i$  表示第  $i$  个节点的特征. 为了获得足够的表达能力, 我们引入了自注意机制, 通过一个由共享的权值向量  $a$  参数化的单层前馈神经网络来计算注意系数:

$$att_{ij} = a(x_i, x_j) \quad (6)$$

$$att = \text{LeakyReLU} \left( \sum_{i, j \in V} att_{ij} \right) \quad (7)$$

其中,  $a \in \mathbb{R}^{2d_i \times 1}$ ,  $att_{ij}$  表示节点  $j$  对节点  $i$  的重要性,  $att \in \mathbb{R}^{n \times n}$  表示所有节点的关注矩阵,  $\text{LeakyReLU}$  是非线性激活函数。

为了将注意矩阵中节点对节点的重要性转化为节点对超边的重要性, 我们进一步将注意矩阵和超图的关联矩阵融合. 在这里, 我们考虑的超图是一种无向图, 因此我们假设节点  $i$  对节点  $j$  的重要性与节点  $j$  对节点  $i$  一致. 融合后的关联矩阵经过  $\text{softmax}$  函数对每一个超边进行归一化, 使超边中不同节点对该超边的贡献易于比较, 并有利于进一步融合超边信息. 上述过程可以表示为

$$H_{att} = \text{softmax}(H^T) \quad (8)$$

$$H^T(e, i) = \begin{cases} att_{ii} + att_{ij} + att_{ji}, & \text{if } i, j \in e \text{ and } i \neq j \\ 0, & \text{if } i \notin e \end{cases} \quad (9)$$

其中,  $e \in E$ ,  $i \in V$  分别表示超边和节点;  $H^T(e, i)$  表示超边  $e$  与节点  $i$  的关联度,  $j$  表示超边  $e$  中节点  $i$  以外的节点, 即节点  $i$  在超边  $e$  中的一阶邻居;  $H_{att}$  是归一化的自注意关联矩阵。

$H_{att}$  的每一行表示了超边连接的节点对该超边的贡献程度, 进而我们可以定义超边特征表示为

$$X_e = H_{att} * X \quad (10)$$

其中,  $X_e = \{x_{e1}, x_{e2}, \dots, x_{em}\}$ ,  $x_{ei} \in \mathbb{R}^{d_i}$ ,  $m$  为超边数,  $x_{ei}$  是第  $i$  个超边的特征。

为了得到超边权重, 我们需要对超边特征进行一个可学习的线性变换. 为此, 将一个共享的参数化权重矩阵  $b$  作用于每个超边, 再经过 *softmax* 归一化, 得到超边权重:

$$W = \text{softmax}(\text{matmul}(X_e, b)) \quad (11)$$

其中,  $b \in R^{d_1^*}$  是可学习参数,  $W \in R^{m^*}$  是归一化的超边权重.

将第 1.2 节的超图拉普拉斯矩阵应用到我们的超图卷积层:

$$X^{(l+1)} = \sigma(D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} X^{(l)} \Theta^{(l)}) \quad (12)$$

其中,  $X^{(l)} \in R^{n^* d_1}$  是第  $l$  层的节点特征矩阵;  $\Theta^{(l)} \in R^{d_1^* d_2}$  是提取隐藏层维度  $d_2$  的节点特征的滤波矩阵;  $\sigma$  是激活函数 *SeLU*<sup>[16]</sup>, 输出更新的节点特征矩阵  $X^{(l+1)} \in R^{n^* d_2}$ . 整个过程可以看作信息在节点-超边-节点上的聚合和传递, 因此能够有效地提取超图上的高阶信息.

### 2.3 超图池化层

我们的超图池化层采用第 1.1 节描述的节点选择机制, 即: 设置一个评分函数, 根据评分结果选择最高的  $k$  个节点. 池化比率  $p = \frac{k}{n} \in (0, 1)$  是一个超参数, 它决定要保留的节点比例.

经过上一节描述的超图卷积层, 每个节点特征已经能够很好地包含超图的结构信息, 只利用节点特征信息就能够进行合理的评分. 因此, 我们采用一个多层感知机(multilayer perceptron, MLP)<sup>[17]</sup>作为评分函数, 它能够自适应地聚集特征信息. 具体过程如下:

$$\text{Score} = \sigma(\text{MLP}(X)) \quad (13)$$

$$id = \text{top}_k(\text{Score}, k) \quad (14)$$

其中,  $X \in R^{n^* d_2}$  是  $d_2$  维的节点特征矩阵;  $\sigma$  是 *softmax* 激活函数, 目的是使得评分结果分布在 0-1 之间;  $\text{Score} \in R^{n^*}$  是经过归一化的一维节点特征的集合, 即每个节点的评分值. MLP 的好处是: 未选择节点的特征信息不会直接丢失, 而是隐性地包含在被选择的节点中. *top\_k* 函数根据 *Score* 返回分数最高的  $k$  个值的索引 *id*, 对应的节点将保留下来. 根据索引 *id* 更新节点特征矩阵和超图拓扑结构的过程可以描述为

$$X' = X_{id,:} \odot \text{Score}_{id,:} \quad (15)$$

$$H' = \text{process}(H_{id,:}) \quad (16)$$

其中,  $\odot$  为元素的哈达玛乘积;  $X_{id,:}$  和  $\text{Score}_{id,:}$  分别为根据索引 *id* 保留的节点特征矩阵和评分向量, 两者相乘能够对保留节点的重要程度加以区分, 进而更好地表示超图;  $H_{id,:}$  为去掉未保留节点的关联矩阵. 这里, 由于超图的关联矩阵与传统图的邻接矩阵不同, 我们定义以下情况的处理方法.

- ① 若删除未保留节点, 将使池化后的超边不包含任何节点, 则池化后的关联矩阵中删除该超边;
- ② 其他情况下, 只在关联矩阵的对应边中删除未保留节点.

公式(16)中的 *process*( $\cdot$ )即为针对上述情况的对应操作. 最终,  $X'$  和  $H'$  分别是池化后超图的特征矩阵和相应的关联矩阵. 整个池化过程可由图 2 所示的 hyper-pooling 模块表示.

### 2.4 读出层

为了将池化后的节点特征聚合起来, 生成固定大小的超图表示, 我们利用到最大值和平均值操作. 具体来说, 用池化后的超图节点在每一特征维度的最大值和平均值的和来表示超图在该维度的特征信息. 最大值能够得到每一维中最具代表性的特征, 而平均值能够得到每一维中的平均特征, 对二者求和既关注到了对超图最重要的特征信息, 又没有直接完全忽视掉其他特征信息, 并且在丰富超图表示的同时不增大信息的维度. 读出层表达式如下:

$$h = \frac{1}{N} \sum_{i=1}^N x_i + \max_{i=1}^N x_i \quad (17)$$

其中,  $N$  为池化后的超图节点数,  $x_i$  为第  $i$  个节点的特征,  $h$  是读出层的输出.

2.5 整体架构

我们采用分层池化的框架, 连续进行 3 次超图卷积-池化操作. 为了平衡较深的超图池化层可能会导致信息过平滑与较浅的超图池化层可能会导致不稳定和信息聚集不足的问题, 我们在第 1 个超图卷积层和每一个超图池化层的后面都接上读出层, 分别提取初始超图和 3 次池化超图的信息, 然后对每个读出层的输出求和, 以获得最佳超图表示. 最后, 将超图表示送入线性变换层, 得到最终的超图表示用于分类任务. 图 4 展示了模型的整体架构.

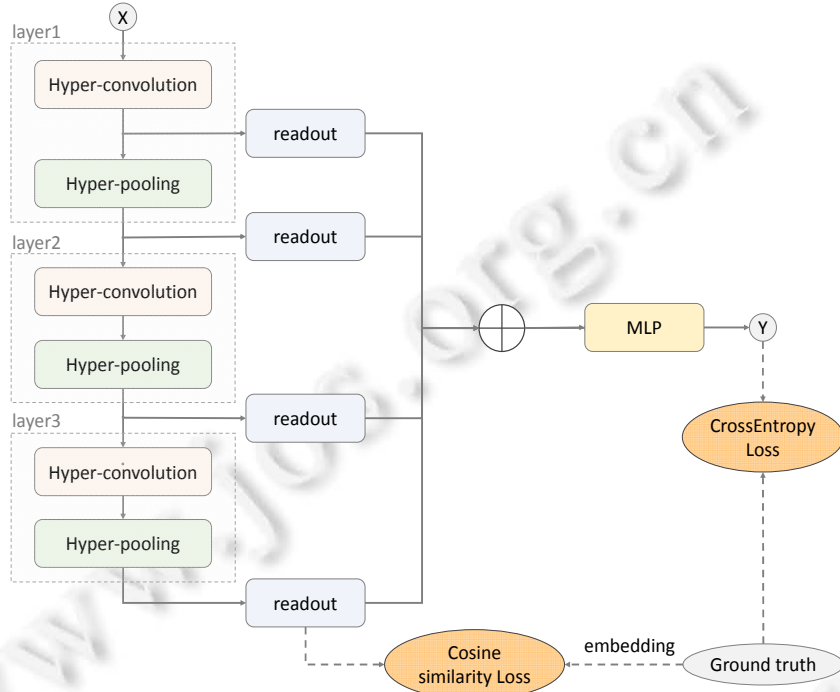


图 4 超图池化网络框架

为了更好地训练模型学习, 引入两种损失函数.

- ① 交叉熵损失. 目标是最大限度地减少真实标签与预测标签之间的差别:

$$loss1 = -\sum_{i \in D} \sum_{f=1}^F g_{if} \ln y_{if} \tag{18}$$

其中,  $D$  为训练数据集;  $F$  是输出特征的维度, 它等于总的类别数量;  $g_i$  是真实标签的独热向量;  $y_i$  是第  $i$  个训练数据的预测标签;

- ② 余弦相似度损失. 通过分析所提模型, 我们认为: 经过 3 层超图池化的模型较为复杂, 仅通过预测标签的交叉熵损失再经过长距离梯度回传并不能有效地指导模型进行参数学习. 因此, 借鉴文献 [18] 中对损失函数的设计方法, 引入余弦相似度损失:

$$loss2 = \sum_{i \in D} (1 - \cos(\text{emb}(g_i), h_3)) \tag{19}$$

$$\cos(A, B) = \frac{A \cdot B}{\|A\| * \|B\|} \tag{20}$$

其中,  $\cos(A, B)$  计算  $A$  和  $B$  的余弦相似度;  $\text{emb}(g_i) \in R^d$  是将真实标签进行编码后的向量表示;  $h_3 \in R^d$  是最后一个读出层的输出, 我们合理认为, 最后一层读出层的输出与真实情况最为相近;  $loss2$  的目标是缩小两个向量表示之间的差异程度.

最终, 训练目标是降低两种损失之和. 在实践中, 我们观察到: 引入两种损失使得训练花费了更长的时间来收敛, 但普遍能够获得更好的效果.

### 3 实验

本节在 3 个任务上评估我们的模型: 文本分类、菜肴分类和蛋白质分类. 通过与其他先进的方法进行对比实验以及设计一系列消融实验, 来分析说明我们模型的有效性.

#### 3.1 文本分类

对于一个文本, 一个节点表示单词, 一个句子视为一个超边, 连接该句子中所有单词. 这样, 每一个文本构建一个超图, 我们的任务即对文本超图进行分类.

##### 3.1.1 数据集

为了进行公平且全面的评估, 我们采用了两个不同领域的 5 个数据集, 这些数据集已经广泛用于评估文本分类的性能. R8 和 R52<sup>[19]</sup>是新闻类文本, 其中第 1 句话是标题. MR<sup>[20]</sup>和 IMDB<sup>[21]</sup>是用于二元情感分类的电影评论数据集, 不同的是, MR 属于短文本集, 大部分评论只有一句话. SST<sup>[22]</sup>是多元情感分类的电影评论数据集, 包含 5 种情感类别. 表 1 总结了文本分类数据集的概要统计.

表 1 文本分类数据集概要统计

	R8	R52	MR	IMDB	SST
Doc	7 691	9 130	10 662	50 000	11 855
Train	5 501	6 560	7 108	25 000	9 645
Test	2 190	2 570	3 554	25 000	2 210
Avg_Sentence_Length	12.86	12.97	16.52	23.85	18.06
Words	7 070	8 149	20 291	135 371	19 175
Class	8	52	2	2	5

##### 3.1.2 基线

我们选择了 4 种类型的基线模型进行比较.

- (1) 基于词嵌入的模型: 使用预先训练的单词嵌入对文本进行分类, 包括 fastText<sup>[23]</sup>和 LEAM<sup>[24]</sup>;
- (2) 基于序列的深度学习模型: 从局部连续的词序列中捕获文本特征的方法, 包括 CNN<sup>[25]</sup>、LSTM<sup>[26]</sup>和 Bi-LSTM<sup>[27]</sup>;
- (3) 基于图神经网络的模型: 通过传统图结构捕捉单词之间的交互, 包括 Graph-CNN<sup>[9]</sup>、TextGCN<sup>[19]</sup>和 InducT-GCN<sup>[28]</sup>. 在这里, 我们分别使用了 TextGCN 作者实现的直推式和归纳式版本;
- (4) 基于超图的模型: HyperGAT<sup>[29]</sup>是第一次将超图结构成功用于文本分类任务的方法. HyperGAT (w/o LDA)是不引入主题词信息的 HyperGAT.

##### 3.1.3 实验细节

我们的模型由 PyTorch 实现, 在一台 11 GB 的 2080ti GPU 上训练和测试. 在 R52 和 R8 数据集上, 随机生成节点的初始向量; 在 MR、IMDB 和 SST 数据集上, 使用 300 维的 GloVe<sup>[30]</sup>词嵌入. 隐藏层维度设置为 128, L2 正则化设置为  $10^{-5}$ , 学习率和池化比率根据不同数据集特点分别在  $10^{-4}$ 、 $5 \times 10^{-5}$ 、 $10^{-5}$  和 0.5、0.6、0.8 中选择. 我们使用早停策略训练不超过 150 个 epoch. 对于基线模型, 我们要么展示原论文中报告的结果, 要么使用原论文中描述的超参数运行作者提供的代码. 按照文献[19]中规定的训练/测试集来分割数据集进行实验, 并随机抽取 90% 的训练集来训练模型, 使用剩下的 10% 作为验证集进行验证. 所有实验都运行了 10 次并报告测试集的平均性能.

##### 3.1.4 实验结果

- 分类性能

我们评估了各模型在 5 个测试集上的准确率, 结果见表 2, OOM 表示内存消耗大于 11 GB. 我们的模型 (ours) 在所有数据集上的性能均优于所有基线. 分析其原因如下: 基于注意机制的超图卷积模块能够更准确地



捕捉并传递高阶上下文信息, 超图池化层能够准确地筛选出关键节点, 4 个读出层能够很好地融合重要节点信息, 进一步得到合理的文本超图表示.

表 2 不同模型在各文本分类数据集上的测试集准确率(%)

Model	R8	R52	MR	IMDB	SST
CNN	94.02	85.37	74.98	86.15	42.30
LSTM	93.68	85.54	75.06	85.91	41.92
Bi-LSTM	96.31	90.54	<b>77.68</b>	86.62	42.63
fastText	96.13	92.81	75.14	80.21	36.08
LEAM	93.31	91.84	76.95	83.29	<b>42.93</b>
Graph-CNN	96.99	92.75	77.22	OOM	35.23
TextGCN (transductive)	<b>97.03</b>	93.93	76.20	OOM	41.18
TextGCN (inductive)	95.93	85.86	74.71	OOM	39.05
InducT-GCN	96.66	93.20	75.15	86.22	41.28
HyperGAT	96.94	<b>94.24</b>	77.66	86.32	41.96
HyperGAT (w/o LDA)	96.39	94.08	76.51	<b>86.67</b>	39.54
Ours+LDA	96.58	92.26	75.41	86.62	44.66
Ours	<b>97.12</b>	<b>94.55</b>	<b>78.14</b>	<b>87.18</b>	<b>45.84</b>

此外, 我们还发现:

- (1) 基于图和超图的方法在前两个数据集上获得了不错的性能, 然而在情感分类任务中, 效果却不如基于序列的方法. 这表明, 现有的基于图和超图的方法能够捕获非连续的、长距离的单词交互, 但在捕获情感信息方面有待提升. 而本文提出的超图池化网络却能够准确地捕获情感信息, 在文本情感分类中优势显著. 为了更好地感受我们模型的优势并理解其是如何捕获情感信息的, 我们可视化了两个 SST 数据集的情感文本, 该文本在最好的基线(基于序列的 Bi-LSTM)方法中情感分类错误, 而我们的模型却能够准确地对其进行分类. 如图 5 所示, 我们分别可视化了一个短文本和一个长文本. 由于短文本句子长度较短, 两次池化后保留的单词节点仅为 3 个, 因此第 3 次池化将不删除任何节点. 两个文本的情感类别分别被标记为 0 和 1 (一共 0-4 共 5 个级别, 数字越大, 情感越正向). 颜色越深, 代表单词情感对文本情感贡献越大. 由于 Bi-LSTM 方法按序一次输入一个单词, 而单词本意的情感并不能代表文本的最终情感. 如图 5 中第 1 个例子, 在句式复杂的情况下, 我们认为, 该模型很难识别出“disguised as comedy”的含义, 进而高估了带有正向情感的“comedy”在文本情感中的权重, 造成分类错误. 而采用超图建模使单词的情感信息能够在整个句子中传递, 我们模型的 3 层池化操作能够逐步选择代表文本情感的关键词, 最终锁定了“dreck”而非“comedy”. 此外, 很多单词节点在超图卷积的过程中融合了其上下文信息, 被赋予了更多的含义(如“as”“a”);
- (2) 对于 TextGCN 的两个版本, 直推式 TextGCN 的输入数据构建了包括训练数据和测试数据所有文本的全局图, 只不过在模型训练阶段没有更新测试集数据对应的参数. 虽然能够实现尚可的效果, 但这种方法严重增加了参数和内存的消耗, 并且当有新的测试数据加入时, 必须重新构造文本全局图并重新训练模型, 而无法直接对新的测试数据进行类别预测; 归纳式 TextGCN 改进了上述缺陷, 在训练阶段没有引入测试数据的任何信息, 但也同时造成了分类效果的显著下降;
- (3) 与本文一样使用超图建模的 HyperGAT, 其作者实现的版本需要使用 LDA 技术通过训练文档为每个类别生成 10 个主题关键词, 并将关键词构成一个句子加入到文本超图中. 这样操作的不足是: 若训练集数据庞大, 则需要消耗很多时间以生成主题关键词; 若训练集数据很小, 则无法获取准确的主题关键词. 而没有引入主题关键词的 HyperGAT 版本, 经实验显示, 并不能实现很好的效果. 为了丰富超边, 我们也实现了引入 LDA 的模型变体, 在表 2 中对应于 Ours+LDA 一行. 然而, 该技术的引入并没有带来更好的效果, 反而在一些数据集上有明显下降. 这说明我们的模型只需要构造原始的自然文本, 无需刻意设计丰富的超边, 就能够实现很好的效果. 此外, 由于 LDA 提取主题词也存在偏差, 造成引入该超边的模型效果并不理想. 能否引入有效的其他形式的超边来丰富图的结构, 可以作为未来的研究内容.

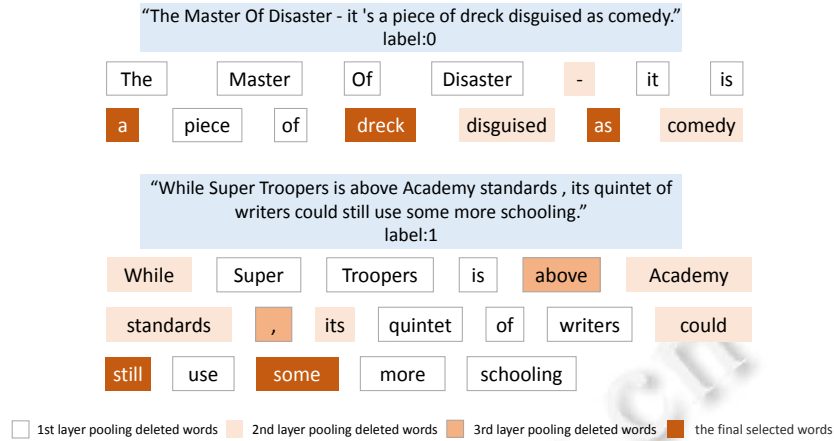


图 5 情感分类文本的池化结果可视化(上: 短文本; 下: 长文本)

• 内存消耗

表 3 给出了最具代表性的基线与我们方法的内存消耗的比较. OOM 表示内存消耗大于 11 GB. 相比之下, 我们的模型具有显著的优势. 直推式的 TextGCN 需要同时使用训练文本和测试文本构建大型文本图, 这不可避免地会消耗大量内存. 而对于 HyperGAT, 虽然构建文本超图比构建文本图占用更少的内存, 但该模型在训练时为了保证模型效果, 一次需要加载一个批次的的数据, 这也会使内存占用增大. 我们的模型是归纳式模型, 且一次加载一个文本, 因而只需要很小的内存, 并且实验结果表明, 这样并不会影响分类性能.

表 3 不同模型在各文本分类数据集上的 GPU 内存消耗比较

Model	TextGCN (transductive)	HyperGAT	Ours
R8	931.58 MB	41.75 MB	<b>33.52 MB</b>
R52	1289.48 MB	46.85 MB	<b>19.62 MB</b>
MR	3338.24 MB	80.99 MB	<b>23.14 MB</b>
IMDB	OOM	297.96 MB	<b>156.46 MB</b>
SST	665.16 MB	68.51 MB	<b>23.15 MB</b>

• 标记数据大小的影响

我们在 MR 和 SST 数据集上比较了几个模型在不同训练数据比例下的表现, 结果如图 6 所示. 一般情况下, 随着标记训练数据比例的增长, 所有方法都能获得更好的性能. 值得注意的是, 我们的模型可以在有限的标记数据下获得优于其他基线的效果, 这表明它在真实的场景中是有效的.

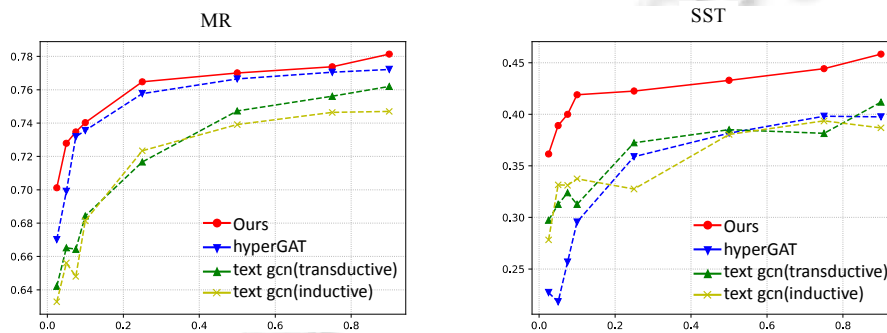


图 6 不同训练数据比例下的测试准确率

3.1.5 消融实验

为了验证各模块在本文所提模型中的作用, 我们进行了一系列的消融实验, 结果见表 4. w/o attention 去掉了超图卷积中的注意机制, 即用第 1.2 节的 HGNN 作为超图卷积操作. traditional\_att 是将本文设计的超边权

重自注意力改为传统点积注意力机制方式<sup>[31]</sup>的模型变体. w/o convolution 和 w/o pooling 分别是去掉超图卷积层和超图池化层的模型变体, w/o loss2 是去掉余弦相似度损失的模型变体. 从结果可以看出, 去掉任何一部分都会导致效果的下降. 其中, 超图卷积层起到了较为关键的作用. 若没有超图卷积的操作, 在所有数据集上, 模型准确率的降幅最为显著. 这说明单凭超图池化操作是不够的, 应在超图卷积层传递超图结构信息并更新节点特征信息的基础上引入超图池化层. 结果表明: 相比于没有超图池化操作的模型变体, 超图池化层的引入, 使得准确率在所有数据集上平均提升了1-2%. 这说明超图卷积和超图池化操作配合进行更为有效. 此外, 余弦相似度损失的引入, 也为模型效果的提升做出了贡献, 尤其是在 R52 数据集上效果最为显著. 我们分析认为: 该损失的设定是基于标签可能存在隐藏含义而考虑的, 因此在类别数较多的 R52 数据集上表现出最为明显的效果. 将传统自注意机制直接迁移到超图上并不能适应全部数据集, 在大部分数据集上甚至不如不使用自注意力机制的效果; 而本文提出的自注意力超边权重的方式更适用于超图的结构, 能够在所有数据集中表现良好, 具有更高的泛化性.

表 4 不同模块对模型准确率的影响(%)

Model	R8	R52	MR	IMDB	SST
w/o attention	96.80	94.12	77.97	86.89	45.16
traditional_att	93.84	90.51	77.60	87.06	43.21
w/o convolution	87.90	85.60	74.70	78.18	38.82
w/o pooling	96.48	93.15	76.70	87.04	43.48
w/o loss2	96.80	92.02	77.15	87.11	44.39
Ours	<b>97.12</b>	<b>94.55</b>	<b>78.14</b>	<b>87.18</b>	<b>45.84</b>

### 3.2 菜肴分类

为了体现我们的模型在超图数据上的有效性, 丰富本文提出的模型的应用任务和场景, 我们采用食谱数据集<sup>[32]</sup>进行菜肴分类的实验任务. 由于目前尚无专门针对超图分类任务的数据集, 以往的超图数据大多无法适应超图级别的分类任务. 因此, 在菜肴分类的任务中, 我们视一个节点为一个配菜, 一条超边表示配菜的组合, 一个超图表示一道菜肴, 标签是该菜肴所属国家菜系, 我们的任务是对菜肴进行分类.

#### 3.2.1 数据集与实验细节

我们采用文献[32]中针对超边分类的食谱数据集 Cooking 进行实验. 表 5 统计了菜肴分类数据集的概要情况. 这里比较特殊的是: 在模型的输入中, 每一个超图只存在一条超边. 此外, 由于文献[32]中的模型任务与我们的不同, 源数据并没有给出训练集和测试集的划分, 我们随机选择数据集的 70% 作为训练集, 10% 作为验证集, 剩余的作为测试集. 在模型输入方面, 我们随机生成节点的初始向量, 隐藏层维度设置为 128, 学习率和池化比率分别设置为 0.000 05 和 0.8. 我们使用早停策略训练不超过 150 个 epoch. 所有实验都运行了 10 次并报告测试集的平均性能.

表 5 Cooking 数据集概要统计

名字	数量
Dish	39 774
Train	27 841
Valid	3 977
Test	7 956
Avg_recipe_Length	10.76
Ingredients	6 714
Class	20

#### 3.2.2 基线和实验结果

由于实验任务的不同, 目前尚无可以直接使用的基线模型在菜肴分类这一任务上做对比实验. 我们构造了两种类型的基线模型进行比较.

- (1) HGNN\_max: 首先, 基于超图卷积网络 HGNN<sup>[13]</sup>更新节点特征; 然后, 采用简单的最大值池化进行信息聚合;

(2) HGNN\_mean: 首先, 基于超图卷积网络 HGNN<sup>[13]</sup>更新节点特征; 然后, 采用简单的平均池化进行信息聚合.

对于基线模型, 我们了改编作者提供的代码, 并使用原文献<sup>[13]</sup>中描述的超参数. 我们评估了各模型在 Cooking 数据的测试集上的准确率, 结果见表 6. 我们的模型在这一任务上同样拥有更好的性能, 不过与基线相差不大. 这说明: 首先, 我们的模型可以很好地适用仅有一条超边的数据情况; 但也由于超图结构过于简单, 只有一条超边, 并且包含的节点数不多, 因此池化操作未能起到明显的作用.

表 6 不同模型在 Cooking 数据集上的测试集准确率(%)

Model	Cooking
HGNN_max	77.62
HGNN_mean	77.62
Ours	<b>77.85</b>

### 3.3 蛋白质分类

D&D 数据集是蛋白质数据, 广泛应用于评价图分类任务的模型性能. 该数据集抽取了 1 178 个高分辨率蛋白质, 其中, 一个节点表示一种氨基酸, 如果两个节点之间的距离少于 6 埃(Angstroms), 则用一条边连接; 一个图表示一个蛋白质, 标签表示该蛋白质是酶还是非酶, 任务即对蛋白质进行分类.

#### 3.3.1 数据集与实验细节

由于我们的模型是针对超图的, 对于传统图表示的蛋白质数据, 我们采用两种方式构建蛋白质超图: 首先, 将原蛋白质图中的二元关系作为一种特殊的超边关系加入到超边中; 然后, 借鉴 DHGNN 模型<sup>[14]</sup>中由普通图数据构造超图数据的方法构造超边. 具体来说, 我们将  $k$ -NN 方法和  $K$ -means 聚类方法结合起来, 利用局部和全局结构来构造超边: 在局部结构方面, 为每个节点计算其  $k$  个近邻; 在全局结构方面, 对所有节点进行  $K$ -means 聚类, 根据欧几里德距离为每个节点选择与其最近的  $S$  簇, 进而将其  $k$  近邻节点、 $S$  簇中心节点与该节点本身共同构成一条超边. 表 7 统计了 D&D 数据集的概要情况.

表 7 D&D 数据集概要统计

名字	数量
protein	1 178
Train	942
Valid	117
Test	119
Avg_Hyperedge_Length	85
Amino acids	89
Class	2

在模型设置方面, 我们随机生成节点的初始向量, 隐藏层维度设置为 128, 学习率设置为 0.000 1, 3 层池化比率分别设置为 0.8、0.6 和 0.5. 我们使用早停策略训练不超过 150 个 epoch. 特别地, 结合我们模型的特点和蛋白质数据的特点, 根据经验将  $k$ -NN 的参数  $k$  设置成 10,  $K$ -means 的参数聚类类别  $C$  设置成 8, 选择最近的  $S$  簇设置成 2. 我们随机选择数据集的 80% 作为训练集, 10% 作为验证集, 剩余的作为测试集. 所有实验都运行了 10 次并报告测试集的平均性能.

#### 3.3.2 基线和实验结果

我们选择 gPool<sup>[12]</sup>、SAGPool<sup>[5]</sup>和 GSAPool<sup>[6]</sup>模型作为基线, 并使用原论文中描述的超参数运行作者提供的代码. 我们评估了本文所提模型和基线模型在 D&D 数据的测试集上的准确率, 结果见表 8.

表 8 不同模型在 D&D 数据集上的测试集准确率(%)

Model	D&D
gPool	75.01
SAGPool	74.78
GSAPool	77.31
Ours	<b>82.35</b>

可以看出, 我们的模型性能明显优于基线. 经过分析, 我们认为: 建模成超图结构进行卷积和池化的确能够更好地关注到图上的高阶信息, 从而提升分类的准确率. 此外, 我们还考虑了是否存在更好的将普通图转化为超图的方法, 这也可以作为未来的研究工作之一.

## 4 总结

在本研究中, 我们提出一种自注意力超图池化网络. 通过带有自注意机制的超图卷积层更新节点信息, 再经过超图池化层的筛选和读出层的汇聚, 我们的方法能够获得更强的超图表达能力, 同时具有更少的参数和空间消耗. 通过大量的实验结果表明, 我们的模型在文本分类、菜肴分类和蛋白质分类任务中均优于其他方法. 未来的研究工作包括但不限于: (1) 验证在其他领域(如视觉识别、姿态估计、图像分类等)中, 我们的模型能否同样取得良好的效果; (2) 寻找有效且可实行的方法, 加快模型训练的收敛速度, 提高模型的鲁棒性; (3) 考虑将模型拓展到社交网络领域, 基于复杂的人际关系构建多元的社交关系超图, 针对不同社交群体进行分类、聚类或相似度匹配等工作.

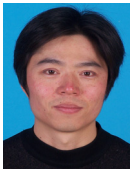
## References:

- [1] Hinton G, Deng L, Yu D, Dahl GE, Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 2012, 29(6): 82–97.
- [2] He K, Zhang X, Ren S, Sun, J. Deep residual learning for image recognition. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*. 2016. 770–778.
- [3] Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Li FF. Large-scale video classification with convolutional neural networks. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*. 2014. 1725–1732.
- [4] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*, 2016.
- [5] Lee J, Lee I, Kang J. Self-attention graph pooling. In: *Proc. of the Int'l Conf. on Machine Learning*. PMLR, 2019. 3734–3743.
- [6] Zhang L, Wang X, Li H, Zhu G, Shen P, Li P, Lu XY, Shah SAA, Bennamoun M. Structure-feature based graph self-adaptive pooling. In: *Proc. of the Web Conf.* 2020. 3098–3104.
- [7] Zhou D, Huang J, Schölkopf B. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in Neural Information Processing Systems*, 2006, 19: 1601–1608.
- [8] Dhillon IS, Guan Y, Kulis B. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2007, 29(11): 1944–1957.
- [9] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, 2016, 29.
- [10] Rhee S, Seo S, Kim S. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. *arXiv:1711.05859*, 2017.
- [11] Ying Z, You J, Morris C, Ren X, Hamilton W, Leskovec J. Hierarchical graph representation learning with differentiable pooling. *Advances in Neural Information Processing Systems*, 2018, 31.
- [12] Gao H, Ji S. Graph u-nets. In: *Proc. of the Int'l Conf. on Machine Learning*. PMLR, 2019. 2083–2092.
- [13] Feng Y, You H, Zhang Z, Ji R, Gao Y. Hypergraph neural networks. In: *Proc. of the AAAI Conf. on Artificial Intelligence*. 2019, 33(1): 3558–3565.
- [14] Jiang J, Wei Y, Feng Y, Cao JX, Gao Y. Dynamic hypergraph neural networks. In: *Proc. of the IJCAI*. 2019. 2635–2641.
- [15] Bai S, Zhang F, Torr PHS. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 2021, 110: 107637.
- [16] Klambauer G, Unterthiner T, Mayr A, Hochreiter S. Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, 2017, 30.
- [17] Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. *Proc. of the National Academy of Sciences*, 1982, 79(8): 2554–2558.
- [18] Wu WQ, Guo Z, Zhou XY, Wu H, Zhang XY, Lian RZ, Wang HF. Proactive human-machine conversation with explicit conversation goals. In: *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. 3794–3804.

- [19] Yao L, Mao C, Luo Y. Graph convolutional networks for text classification. In: Proc. of the AAAI Conf. on Artificial Intelligence. 2019, 33(1):7370–7377.
- [20] Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics. 2005. 115–124.
- [21] Maas A, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. In: Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011. 142–150.
- [22] Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proc. of the 2013 Conf. on Empirical Methods in Natural Language Processing. 2013. 1631–1642.
- [23] Joulin A, Grave E, Bojanowski P, Mikolov T. Bag of tricks for efficient text classification. In: Proc. of the 15th Conf. of the European Chapter of the Association for Computational Linguistics. 2017. 427–431.
- [24] Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R, Carin L. Joint embedding of words and labels for text classification. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics. 2018. 2321–2331.
- [25] Kim Y. Convolutional neural networks for sentence classification. In: Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing. 2014. 1746–1751.
- [26] Liu P, Qiu X, Huang X. Recurrent neural network for text classification with multi-task learning. In: Proc. of the 25th Int'l Joint Conf. on Artificial Intelligence. 2016. 2873–2882.
- [27] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv:1508.01991, 2015.
- [28] Wang K, Han SC, Poon J. InducT-GCN: Inductive graph convolutional networks for text classification. arXiv:2206.00265, 2022.
- [29] Ding K, Wang J, Li J, Li D, Liu H. Be more with less: Hypergraph attention networks for inductive text classification. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing. 2020. 4927–4936.
- [30] Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP). 2014. 1532–1543.
- [31] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. arXiv:1710.10903, 2017.
- [32] Amburg I, Veldt N, Benson A. Clustering in graphs and hypergraphs with categorical edge labels. In: Proc. of the Web Conf. 2020. 2020. 706–717.



赵英伏(1997—), 女, 硕士, 主要研究领域为图神经网络, 知识图谱。



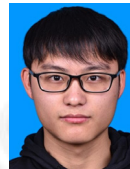
金福生(1977—), 男, 博士, 副教授, CCF 高级会员, 主要研究领域为大数据, 区块链, 人工智能。



李荣华(1985—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为大规模图数据管理与挖掘, 社交网络分析与挖掘。



秦宏超(1992—), 男, 博士, 主要研究领域为图数据库, 时序图挖掘, 知识图谱挖掘。



崔鹏(2000—), 男, 学士, 主要研究领域为自然语言处理, 图神经网络。



王国仁(1966—), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为数据管理, 大数据计算, 知识图谱, 区块链, 生物信息学。