DOI:10.16136/j.joel.2022.12.0117

二值化身份感知图卷积神经网络

苏树智[1,2*]，卢彦丰[1]

(1.安徽理工大学 计算机科学与工程学院,安徽 淮南 232001;2.合肥综合性国家科学中心 人工智能研究院,安徽 合肥 230088)

摘要:针对有限的内存资源导致图神经网络(graph neural network,GNN)无法完全加载属性图的问题,文中提出了二值化身份感知图卷积神经网络(binary identify-aware graph convolutional network,BID-GCN)。该网络通过在消息传递过程中递归地考虑节点的信息,为了获得一个给定的节点的嵌入,BID-GCN 将提取以该节点为中心的自我网络,并进行多轮的异构消息传递,在自我网络的中心节点上应用与其他节点不同的参数。在消息传递过程中,对网络参数和输入节点特征进行二值化,并将原始的矩阵乘法修改为二值化以加速运算。通过理论分析和实验评估,BID-GCN 可以减少网络参数和输入数据的平均约 36 倍的内存消耗,并加快引文网络上平均约 49 倍的推理速度,可以提供与全精度基线相当的性能,较好地解决内存资源有限的问题。

关键词:深度学习;图卷积神经网络(GCN);消息传递;二值化方法

中图分类号:TP391 **文献标识码**:A **文章编号**:1005-0086(2022)12-1280-07

Binary identify-aware graph convolutional network

SU Shuzhi[1,2*], LU Yanfeng[1]

(1. School of Computer Science and Engineering, Anhui University of Science & Technology, Huainan, Anhui 232001, China; 2. Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui 230088, China)

Abstract: To solve the problem that graph neural network (GNN) cannot fully load the entire attributed graph due to limited memory resources, the binary identify-aware graph convolutional network (BID-GCN) is proposed. In this network, the nodes information is considered recursively during message passing, and then in order to obtain an embedding of a given node, the BID-GCN will extract the ego network centered at that node and perform multiple rounds of heterogeneous message passing, applying different parameters to the central node of the ego network to the rest of the nodes. In this process, the network parameters and input node features are binary by the network. In addition, the original matrix multiplication is modified to be binary to speed up the operation. Through theoretical analysis and experimental evaluation, BID-GCN can reduce the memory consumption by the average approximate 36 times of both the network parameters and input data, and accelerate the inference speed by the average approximate 49 times on the citation networks. It can provide comparable performance to full precision baselines, and can better tackle the problem of limited memory resources.

Key words: deep learning; graph convolutional network (GCN); message passing; binary method

1 引 言

近年来,伴随着互联网技术的飞速发展,人工智能已经应用到生活的不同方面,其中自然语言处理、图像处理和目标检测等是目前人工智能领域的研究热点。卷积神经网络(convolution neural

* **E-mail**:sushuzhi@foxmail.com

收稿日期:2022-03-01 **修订日期**:2022-04-04

基金项目:国家自然科学基金(61806006)、中国博士后科学基金(2019M660149)、安徽省重点研发计划国际科技合作专项(202004b11020029)、安徽高校协同创新项目(GXXT-2021-006)和合肥综合性国家科学中心能源研究院项目(19KZS203)资助项目

network,CNN)作为一种强有力的方法可以很好地应用于这些领域且取得的结果明显优于传统方法。CNN 研究的对象是欧式数据,然而现实生活中很多数据不具备规则的空间结构,称为非欧式数据,如推荐系统、社交网路、分子结构等抽象出来的图谱。对于非欧式数据,CNN 难以选取固定的卷积核来适应整个图的不规则性,如邻居节点数量和节点顺序的不确定性,使得 CNN 处理的结果差强人意。图神经网络[1,2](graph neural network,GNN)得益于可以从不规则数据中学习到有效的表示,它在各种基于图的任务中展现了良好的性能。基于 GNN 优越的表示能力,研究者还将其应用于许多任务,包括自然语言处理、计算机视觉、推荐系统等。图卷积神经网络[3](graph convolutional network,GCN)作为 GNN 的一种,基于图谱理论[3]实现非欧数据上的卷积操作[4],能有效学习节点的空间特征信息,且相关的研究成果应用到了文本分类、交通预测、医学诊断等任务中。GCN 通过局部感知与参数共享[5,6]有效降低网络复杂度。该网络在针对图像识别、变形、倾斜、倒置等变形形式具有较高的适应性和优越性,通过将图像特征作为直接输入,在图卷积层进行特征提取[7]和聚合[8]过程。因而 GCN 在语音识别、文本分类和图像分类等诸多问题中表现出色,而且在很多具体的领域得到了普及应用。

目前,GCN 的成功在于一个隐含的假设,即 GCN 的输入包含整个属性图。问题在于,如果整个图太大,由于内存资源有限而无法完整地输入 GNN,在训练和推理过程中,很可能当图的规模增大时,GCN 的性能反而会大幅度下降。为了解决这个问题,一个可行的解决方案是压缩输入图数据[9]和 GNN 模型的大小[10],以更好地利用有限的内存资源和计算资源。现已提出了几种压缩 CNN 的方法,如剪枝、量化参数和设计浅层网络[11]在基于量化的方法中,二值化[12]在许多基于 CNN 网络的实际视觉任务中取得了很大的成功。另一个可行的解决方案是采样,如采样一个具有合适大小的子图,以便完全加载到 GCN 中。基于采样的方法可分为邻域采样和图采样两类。邻域采样为下一层中的每个节点选择固定数量的邻居,以确保每个节点都可以被采样。因此,它可以同时用于训练和推理过程。问题在于当层数增加时,邻域爆炸问题出现,训练和推理时间都会呈指数增长。与邻域采样不同,图采样[13]在训练过程中对一组子图进行采样,可以避免邻域爆炸的问题。但是,它不能保证每个节点在整个训练/推理过程中至少可以采样一次。因此,它只对训练过程可行,原因在于测试过程通常需要 GCN 来处理图中的每个节点。

GCN 的压缩[14]具有独特的挑战。首先,由于输入图数据通常比 GNN 模型要大得多,因此对加载数据的压缩需要更多的关注。其次,GCN 通常很浅,例如,标准的 GCN 仅仅只有两层,其中包含的冗余更少,因此压缩将更难以实现。最后,节点倾向于在高层语义空间中与它的邻居相似,而在低层次的特征空间[15]中,它们往往是不同的。这与图像、视频等网格化数据不同。这一特性要求压缩的 GCN 具有足够的参数来表示。一般来说,在压缩 GCN 中的压缩比[16]和精度[17]之间的权衡需要仔细的设计和研究。为了解决内存和复杂性问题,简化图卷积神经网络(simplifying graph convolutional networks,SGC)是 1 层 GNN,通过去除连续层之间的非线性和折叠权重矩阵来压缩 GCN。这种浅层的 GNN 可以以相当的性能加速训练和推理过程。虽然 SGC 压缩网络参数,但它不会压缩加载的数据,这是使用 GNN 处理图形时的主要内存消耗。本文为了缓解内存和复杂性的问题,提出了一种二值化身份感知图卷积神经网络(binary identify-aware graph convolutional network,BID-GCN),通过在消息传递过程中递归地考虑节点的信息,然后为了获得一个给定的节点的嵌入,BID-GCN 首先提取以该节点为中心的自我网络,然后进行多轮的异构消息传递,在自我网络的中心节点上应用与其他节点不同的参数。具体来说,权值的二值化是通过将它们划分为多个特征选择器,并保持每个选择器的标量以进一步减少量化误差来实现的。类似地,节点特征的二值化可以通过分割节点特征并为每个节点分配注意权重来实现。通过使用这些额外的标量,可以有效地学习和保留更有效的信息。在对权值和节点特征进行二值化后,可以大大降低由网络参数和输入数据引起的计算复杂度和内存消耗。

2　相关工作

给定一个无向图 G,图的卷积运算可以表示为:

$$\boldsymbol{H}^{(l+1)}=\boldsymbol{\sigma}(\tilde{\boldsymbol{A}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}),\tag{1}$$

式中,$\tilde{\boldsymbol{A}}=\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}$ 是一个稀疏矩阵,$\boldsymbol{W}^{(l)}\in\mathbb{R}^{r_{\text{in}}^{(l)}\times r_{\text{out}}^{(l)}}$ 包含可学习参数,$\boldsymbol{H}^{(l+1)}$ 为第 l 层的输出和第 $l+1$ 层的输入,且 $\boldsymbol{H}^{(0)}=\boldsymbol{X}$,$\boldsymbol{\sigma}$ 是非线性激活函数,例如 Sigmoid、ReLU、tanh 等。

从空间方法的角度来看,GCN 中的图卷积层可以分解为两个步骤,其中 $\tilde{\boldsymbol{A}}\boldsymbol{H}^{(l)}$ 为聚合步骤,$\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}$ 为特征提取步骤。聚合步骤倾向于将局部邻域中的节点属性约束为相似的。然后,特征提取步骤可以

很容易地提取相邻节点之间的共性。GCN 通常使用依赖于任务的损失函数，例如，节点分类任务的交叉熵损失，其定义为：

$$\mathscr{L}=-\sum_{v_i\in\nu^{\text{label}}}\sum_{c=1}^{C}\boldsymbol{Y}_{i,c}\log(\widetilde{\boldsymbol{Y}}_{i,c}),\tag{2}$$

式中，ν^{label}表示标记节点集，C 为类数，$\boldsymbol{Y}$ 为真实标签，$\widetilde{\boldsymbol{Y}}_{i,c}=\text{soft max}(\boldsymbol{H}^{(L)})$为第 L 层 GCN 的预测。

最近提出了许多表示能力优异的 GNN。但是现有方法均引入了标准消息传递 GNN 额外的或者是特定于任务的组件。例如，GCN 的嵌入无法适用于确定最短路径距离。本文着重强调了消息传递 GNN 的优势，并证明了 GNN 在加入归纳身份信息后，在保持高效、简单和广泛适用性的同时，仍然具有良好的表示能力。CNN 存在计算成本高等问题。二值化作为一种很有前途的网络压缩技术，已被广泛用于降低 CNN 的内存和计算成本。二进制连接将网络参数二进制化，并用浮点加法替换大部分浮点乘法。二元网进一步将激活函数二值化，并使用 XNOR 操作来加速推理过程。XNOR-Net 提出了一种基于标量的二值化方法，并成功地将其应用于流行的 CNNs，如 ResNet 和 GoogleNet。

本文结合上述两种方法的优势，通过在消息传递过程中递归地考虑节点的信息，然后为了获得一个给定的节点的嵌入，BID-GCN 首先提取以该节点为中心的自我网络，然后进行多轮的异构消息传递，在自我网络的中心节点上应用与其他节点不同的参数。在此过程中，再对网络参数和输入节点特征进行二值化，以达到减少参数运算和加速推理速度的目的。

3 BID-GCN

在本节中，提出了 BID-GCN。BID-GCN 通过在消息传递过程中归纳地考虑每个节点的身份信息来获取每个节点的嵌入，然后利用身份信息进行多轮的异构消息传递。图卷积层可以分解为聚合和特征提取两个步骤。在 BID-GCN 中，本文只关注特征提取步骤的二值化，因为聚合步骤没有可学习的参数(可以忽略内存消耗)，它只需要少量的计算(与特征提取步骤相比，这可以忽略)。因此，保持了原始 GCN 的聚合步骤。对于特征提取步骤，将网络参数和节点特征都进行了二值化，以减少内存消耗。为了降低计算的复杂性和加速推理过程，使用了 XNOR 和位计数操作，而不是传统的浮点乘法。

3.1 身份感知

本文首先使用 *K-hop* 自我网络 $G_v^{(K)}$ 用于获取给定节点 $v\in V$ 的 K 层嵌入。在整个嵌入过程中，$G_v^{(K)}$ 中的节点可以分为着色节点和无着色节点。这种着色技术是归纳的，因为即使重新排列节点，自我网络的中心节点仍然可以与其他相邻节点进行区分。然后将 K 轮消息传递应用于所有提取的自我网络中。为了得到节点 $u\in G_v^{(K)}$ 的嵌入，使用式(3)以实现异构消息传递：

$$\begin{aligned}m_s^{(k)}&=MSG_{\mathbb{I}[s=v]}^{(k)}(h_s^{(k-1)})\\h_u^{(k)}&=AGG^{(k)}(\{m_s^{(k)},s\in N(u)\},h_u^{(k-1)}),\end{aligned}\tag{3}$$

式中，在 K 轮迭代等式(3)之后，仅使用 $h_v^{(k)}$ 作为节点 v 的嵌入表示。$\mathbb{I}[s=v]$ 是指示器函数，当 $s=v$ 时，$\mathbb{I}[s=v]=1$，反之亦然。通过这种方式，归纳身份着色被编码到 BID-GCN 的计算图中。

3.2 特征提取的二值化

基于向量二值化算法，可以对式(1)所示的图卷积中的特征提取步骤 $\boldsymbol{Z}^{(l)}=\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}$ 进行二值化处理。需要注意的是，对于该特征提取，采用单元方法将二值化内积运算推广到二值化矩阵乘法运算。具体来说，是将矩阵分割成多个具有固定大小的连续值的向量，并分别执行缩放操作。

由于第 l 层的参数矩阵的每列在 $\boldsymbol{Z}^{(l)}$ 的计算中是作为特征提取器，因此在 $\boldsymbol{W}^{(l)}$ 的每列被分割成一个向量。设 $\varphi^{(l)}=(\varphi_1^{(l)},\varphi_2^{(l)},\cdots,\varphi_{\tau_{\text{out}}^{(l)}}^{(l)}$，是每个向量的数值部分，$\boldsymbol{D}^{(l)}=(\boldsymbol{D}_1^{(l)},\boldsymbol{D}_2^{(l)},\cdots,\boldsymbol{D}_{\tau_{\text{out}}^{(l)}}^{(l)})\in\{-1,1\}^{\tau_{\text{in}}^{(l)}\times\tau_{\text{out}}^{(l)}}$ 为 $\boldsymbol{W}^{(l)}$ 的二值化向量。然后，基于向量二值化算法可以轻松地计算出最优的 $\boldsymbol{D}^{(l)}$ 和 $\varphi^{(l)}$：

$$\boldsymbol{D}_j^{(l)}=\text{sign}(\boldsymbol{W}_{:,j}^{(l)}),\tag{4}$$

$$\varphi_j^{(l)}=\frac{1}{\tau_{\text{out}}^{(l)}}\|\boldsymbol{W}_{:,j}^{(l)}\|_1,\tag{5}$$

式中，$\boldsymbol{W}_{:,j}^{(l)}$表示 $\boldsymbol{W}^{(l)}$的第 j 列。它可以近似为：

$$\boldsymbol{W}_{:,j}^{(l)}\approx\widetilde{\boldsymbol{W}}_{:,j}^{(l)}=\varphi_j^{(l)}\boldsymbol{D}_j^{(l)},\tag{6}$$

基于式(6)，具有二值化权值的图卷积操作可以描述为：

$$\boldsymbol{H}^{(l+1)}\approx\boldsymbol{H}_b^{(l+1)}=\boldsymbol{\sigma}(\widetilde{\boldsymbol{A}}\boldsymbol{H}^{(l)}\widetilde{\boldsymbol{W}}^{(l)}),\tag{7}$$

式中，$\boldsymbol{H}_b^{(l+1)}$ 是 $\boldsymbol{H}^{(l+1)}$的二值化参数 $\boldsymbol{W}^{(l)}$ 的二值近似。与全精度的参数相比，参数的二值化可以减少平均内存消耗约 36 倍。

3.3 节点特征的二值化

为了将节点特征二值化，根据矩阵乘法的约束将 $\boldsymbol{H}^{(l)}$ 分成向量来计算 $\boldsymbol{Z}^{(l)}$，即 $\boldsymbol{H}^{(l)}$ 的每行将对 $\boldsymbol{W}^{(l)}$ 的每列进行内部积。设 $\gamma^{(l)}=(\gamma_1^{(l)},\gamma_2^{(l)},\cdots,\gamma_N^{(l)})$ 表示 $\boldsymbol{H}^{(l)}$ 中每个向量的数值部分，$\boldsymbol{F}^{(l)}=(\boldsymbol{F}_1^{(l)},\boldsymbol{F}_2^{(l)},\cdots,\boldsymbol{F}_N^{(l)})\in\{-1,1\}^{N\times r_{\text{in}}^{(l)}}$ 为批二值化。然后，利用向量二值化算法计算最优 γ 和 $\boldsymbol{F}$：

$$\gamma_i^{(l)}=\frac{1}{N}\|\boldsymbol{H}_{i,:}^{(l)}\|_1,\tag{8}$$

$$\boldsymbol{F}_i^{(l)}=\text{sign}(\boldsymbol{H}_{i,:}^{(l)}),\tag{9}$$

式中，$\boldsymbol{H}_{i,:}^{(l)}$表示 $\boldsymbol{H}^{(l)}$的第 i 行。然后，可以得到 $\boldsymbol{H}^{(l)}$的二值化近似：

$$\boldsymbol{H}_{i,:}^{(l)} \approx \widetilde{\boldsymbol{H}}_{i,:}^{(l)} = \gamma_i^{(l)} \boldsymbol{F}_i^{(l)}, \tag{10}$$

γ 可以看作是节点权值的特征表示。最后,可以将具有二值化权值和节点特征的图卷积运算表示为:

$$\boldsymbol{H}^{(l+1)} \approx \boldsymbol{H}_b^{(l+1)} = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{H}}^{(l)}\widetilde{\boldsymbol{W}}^{(l)}。 \tag{11}$$

3.4　二值化操作

利用二值化的图卷积层,可以使用 XNOR 和位计数操作而不是浮点加法和乘法来急速运算。设 $\boldsymbol{\eta}^{(l)}$ 表示 $\boldsymbol{Z}^{(l)}$ 的近似值。然后:

$$\boldsymbol{Z}_{ij}^{(l)} \approx \boldsymbol{\eta}_{ij}^{(l)} = \gamma_i^{(l)} \varphi_j^{(l)} \boldsymbol{F}_{i,:}^{(l)} \cdot \boldsymbol{D}_{:,j}^{(l)}, \tag{12}$$

由于 $\boldsymbol{F}^{(l)}$ 和 $\boldsymbol{D}^{(l)}$ 的每个元素都是经过二值化处理过的值,这两个二值化向量之间的内积可以被二进制操作替换,式(12)可以被重写为:

$$\boldsymbol{\eta}_{ij}^{(l)} = \gamma_i^{(i)} \varphi_j^{(l)} \boldsymbol{F}_{i,:}^{(l)} \otimes \boldsymbol{D}_{:,j}^{(l)}, \tag{13}$$

式中,⊗表示使用 XNOR 和位计数运算的二进制乘法操作详细的过程,如图 1 所示。因此,普通的 GCN 中图卷积操作可以用:

$$\boldsymbol{H}^{(l+1)} \approx \boldsymbol{H}_b^{(l+1)} = \widetilde{\boldsymbol{A}}\boldsymbol{\eta}^{(l)}, \tag{14}$$

式中,$\boldsymbol{\eta}^{(l)}$ 是通过式(13)计算出来的,$\boldsymbol{H}_b^{(l+1)}$ 是第 l 层的输出,其具有二值化的参数和输入。通过使用这种二进制乘法运算,原始的浮点运算可以用相同数量的二进制浮点运算和一些其他的浮点运算来取代。这将大大加快图卷积层的处理速度。

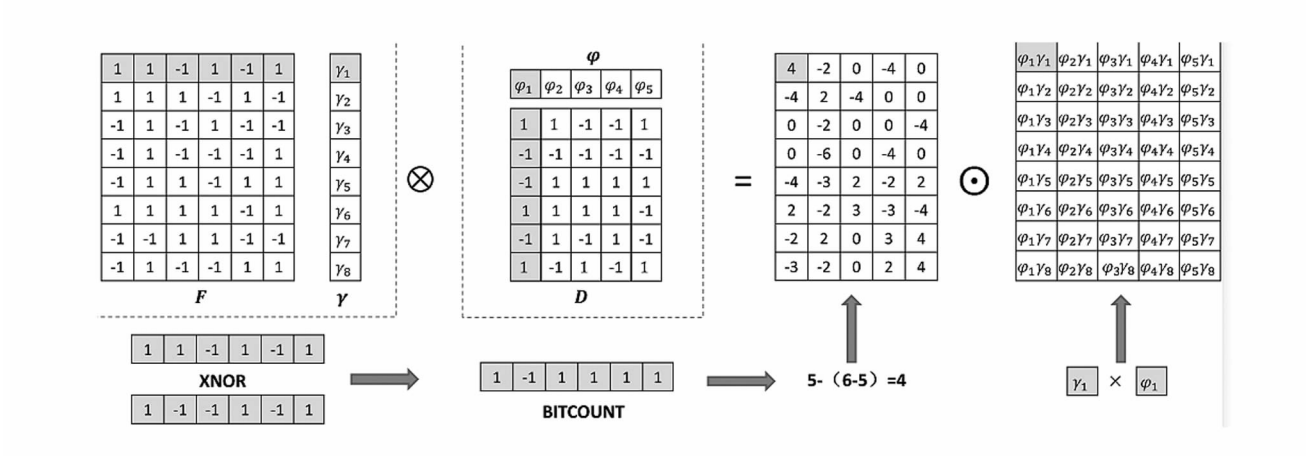

图 1　二值化特征提取过程样例

Fig. 1　An example of binary feature extraction step

3.5　数据大小压缩

目前,加载的数据往往导致大部分内存消耗。在常用的数据集中,节点特征倾向于贡献大部分的加载数据。因此,当 GNNs 处理数据集时,对加载的节点特征的二值化可以大大减少内存消耗。需要注意的是,由于通常处理过的数据图中的边是系数的,并且除法掩码的大小也很小,所以本文使用节点特征的数据大小作为整个加载数据大小的近似值。

设将加载的节点特征表示为 $\boldsymbol{X} \in \mathcal{R}^{N \times r}$,式中,$N$ 为节点数,τ 为每个节点的特征数,全精度 $\boldsymbol{X}$ 包含 $N \times r$ 个浮点值。在 BID-GCN 中,将加载的数据 $\boldsymbol{X}$ 特征化,可以得到 $N \times \tau$ 个二值化值和 N 个浮点值。所加载的数据 $\boldsymbol{X}$ 的大小可以减少一倍:

$$DC = \frac{32N\tau}{N\tau + 32N} = \frac{32\tau}{\tau + 32}, \tag{15}$$

根据式(15),加载数据大小的压缩比取决于节点特征的尺寸。在实际应用中,BID-GCN 可以平均减少约 36 倍的内存消耗,这表明一个更大的属性图完全可以加载相同的内存消耗。对于一些归纳数据集,本文可以成功地加载整个图,或者使用比全精度 GCN 中更大的子图。

3.6　加　速

在分析了内存消耗后,对 BID-GCN 与 GCN 相比的加速度进行了分析。设输入矩阵和第 l 层的参分别具有 $N \times \tau_{\text{in}}^{(l)}$、$\tau_{\text{in}}^{(l)}$ 和 $\tau_{\text{out}}^{(l)}$ 的维度。GCN 中的原始特征提取步骤需要 $N\tau_{\text{in}}^{(l)}\tau_{\text{out}}^{(l)}$ 加法和 $N\tau_{\text{in}}^{(l)}\tau_{\text{out}}^{(l)}$ 乘法运算。相反,BID-GCN 中的二值化特征提取步骤只需要 $N\tau_{\text{in}}^{(l)}\tau_{\text{out}}^{(l)}$ 个输出二进制运算和 $2N\tau_{\text{out}}^{(l)}$ 个浮点乘法运算。第 l 层特征提取步骤的加速度比可以计算为:

$$S_{fe}^{(l)} = \frac{N\tau_{\text{in}}^{(l)}\tau_{\text{out}}^{(l)}}{\frac{1}{64}N\tau_{\text{in}}^{(l)}\tau_{\text{out}}^{(l)} + 2N\tau_{\text{out}}^{(l)}} = \frac{64\tau_{\text{in}}^{(l)}}{\tau_{\text{in}}^{(l)} + 128}, \tag{16}$$

从上式可以看出,节点特征 $\tau_{\text{in}}^{(l)}$ 的尺寸决定了特征提取步骤的加速效率。

4　实　验

4.1　数据集

在 4 个常用的数据集上进行实验,分别是 Cora、

CiteSeer、PubMed 和 Flickr。在转换学习任务中，使用了 3 种常用的 Cora、CiteSeer 和 PubMed[18]。本文采用了与文献[19]相同的数据划分策略。归纳学习任务采用 Flickr，对 Flickr 采用与 GraphSAGE[14] 相同的数据划分策略。这些数据集汇总如表 1 所示。

表 1　数据集

Tab. 1　Dataset

Dataset	Nodes	Edges	Classes	Features
Cora	2 708	5 429	7	1 433
CiteSeer	3 327	4 732	6	3 703
PubMed	19 711	44 338	3	500
Flickr	89 520	899 756	7	500

4.2　实验设置

为了验证提出算法的性能，实验对 BID-GCN 和另外 4 种具有代表性的 GNN 算法 GCN[10]、Bi-GCN[20]、GAT[21] 和 FastGCN[22] 进行比较。对于转换学习任务，本文选择了一个包含 64 个隐藏层单元的 2 层 GCN 作为基线。BID-GCN 是通过将改进 GCN 进行二值化得到的。在训练过程中，GCN 和 BID-GCN 都通过学习速率为 0.001 的 Adam 优化器来训练最多 1 000 个 epochs，早期停止条件为 200 个 epochs。在将中间层的输入进行二值化后，在训练过程中使用退出层，退出率为 0.3。对 BID-GCN 中的输入特征向量应用了标准的批归一化处理。对于归纳学习任务，选择归纳式 GCN[10] 和 GraphSAGE[14] 作为本文实验的基线，本文采用了他们自己文献中的设置。通过实验对所有的特征提取步骤进行二值化，以推广它们相应的二值化版本。BID-GCN 二值化模型中的超参数被设置为与它们的全精度版本相同。

4.3　实验结果

转换学习任务的结果如表 2 所示。可以观察到，BID-GCN 提供了与全精度 GCN 和其他基线相当的性能。与此同时，与普通的 GCN、FastGCN 和 GAT 相比，BID-GCN 可以显著节省加载数据和类型的内存消耗，并减少具有相当性能的计算量，如表 3 所示。Flickr 的原始数据大小为 170.21 MB，而 BID-GCN 只需要 5.56 MB 来加载数据，这证明了二值化身份感知方法的有效性。实验选用常见的准确率(accuracy)、模型大小(model size，M. S.)、数据大小(data size，D. S.)、循环操作(cycle operations，C. O.)和 F1-micro 作为评估指标。

表 2　转换学习结果

Tab. 2　Transductive learning results

Networks	Cora				CiteSeer				PubMed			
	Accuracy /%	M. S. /kB	D. S. /MB	C. O.	Accuracy /%	M. S. /kB	D. S. /MB	C. O.	Accuracy /%	M. S. /kB	D. S. /MB	C. O.
GCN	82.7	360.0	14.8	2.50×10^8	70.3	386.1	15.3	2.61×10^8	78.0	125.7	37.6	6.3×10^8
Bi-GCN	81.1	11.5K	0.47	4.67×10^6	71.8	16.3	0.53	4.81×10^6	78.6	4.18	1.25	1.55×10^7
BID-GCN	82.1	**10.6**	**0.36**	$\mathbf{4.36\times10^6}$	**73.2**	**11.2**	**0.41**	$\mathbf{4.52\times10^6}$	**79.2**	**4.10**	**1.10**	$\mathbf{1.46\times10^7}$
GAT	**82.8**	361.2	14.9	2.53×10^8	72.4	388.3	15.3	2.62×10^8	79.1	126.3	37.6	6.44×10^8
FastGCN	79.6	360.0	14.9	2.51×10^8	72.1	386.1	15.3	2.59×10^8	79.2	125.7	37.6	6.36×10^8

表 3　归纳学习结果

Tab. 3　Inductive learning results

Networks	Flickr			
	F1-micro	M. S. /kB	D. S. /MB	C. O.
InductiveGCN	51.2	508.0	170.2	1.18×10^{10}
Bi-inductiveGCN	50.6	6.8	5.66	4.65×10^8
BID-inductiveGCN	**51.4**	**16.2**	**5.63**	$\mathbf{3.96\times10^8}$
GraphSAGE	50.9	1014.0	170.2	2.34×10^{10}
Bi-GraphSAGE	50.4	33.76	5.66	6.93×10^8
BID-GraphSAGE	50.6	32.89	5.63	6.26×10^8

图 2 显示了 GCN 和 BID-GCN 在不同模型深度的 Cora 上的转导效果。可以观察到，BID-GCN 比原始的 GCN 更适合构建更深的 GNN。当 GCN 由 3 个或3个更多的图卷积层组成时，它的精度急剧下降。相反，BID-GCN 的性能缓慢下降。根据图 3，随着层数的增加，GCN 很快就会被过拟合问题所困扰。然而 BID-GCN 可以有效地缓解这种过拟合问题。图 4 说明了内存消耗问题，当层数增加时，BID-GCN 可以节省更多的内存。对于加速度的结果，随着层数的增加，GCN 与 BID-GCN 之间的比值有略微下降的趋势，而实际降低的计算成本则有所增加。

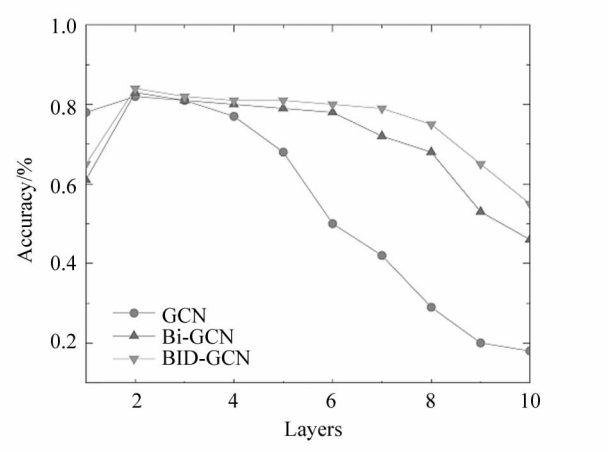

图 2　Cora 数据集上的准确率对比

Fig. 2　Comparisons of accuracy on Cora data set

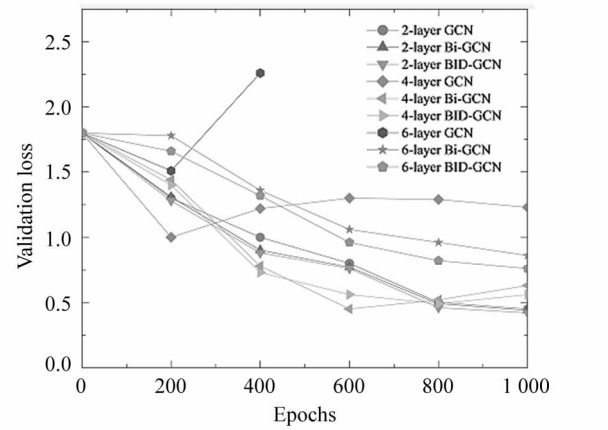

图 3　验证损失对比

Fig. 3　Comparisons of validation loss

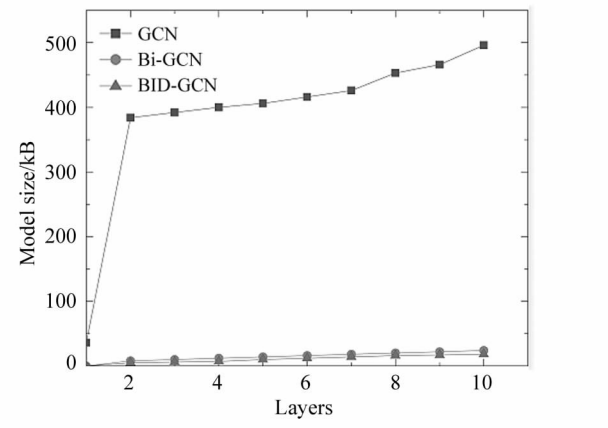

图 4　模型大小对比

Fig. 4　Comparisons of model size

5　结　论

本文提出一个基于图卷积半监督学习的二值化改进版本 GCN,利用 BID-GCN 在图半监督学习方面的优势解决 GNN 依赖于将整个属性图加载到网络中进行处理的问题。BID-GCN 通过在消息传递过程中归纳地考虑节点的身份信息,并在此过程中对其网络参数和节点属性二值化来加快推理速度减少内存消耗,扩展了现有的 GNN 体系结构,浮点运算已被推理加速的二值化运算所取代。基于理论分析,BID-GCN 可以大幅减少网络参数和节点属性的内存消耗,并显著加快引文网络的推理速度。在多个数据集上的实验表明,BID-GCN 在转换和归纳任务中都可以提供与 GCN 相当的性能,证明了 BID-GCN 方法的有效性。

参考文献:

[1] XU B B, CEN K T, HUANG J J, et al. A survey on graph convolutional neural network[J]. Chinese Journal of Computers, 2020, 43(5): 755-780.
徐冰冰,岑科廷,黄俊杰,等.图卷积神经网络综述[J].计算机学报,2020,43(5):755-780.

[2] GE Y, CHEN S C. Graph convolutional network for recommender systems[J]. Journal of Software, 2020, 31(4): 1101-1112.
葛尧,陈松灿.面向推荐系统的图卷积网络[J].软件学报,2020,31(4):1101-1112.

[3] CLAUDIO G, ALESSIO M. Fast and deep graph neural networks[C]//The Thirty-Fourth AAAI Conference on Artificial Intelligence, February 7-12, 2020, New York, USA. Menlo Park, CA: AAAI, 2020: 3898-3905.

[4] GARCIA V, BRUNA J. Few-shot Learning with graph neural networks[EB/OL]. (2018-02-12)[2022-03-01]. https://arxiv.org/abs/1711.04043.

[5] CHEN J, MA T F, XIAO C. FastGCN: Fast learning with graph convolutional networks via importance sampling [EB/OL]. (2018-01-30)[2022-03-01]. https://arxiv.org/abs/1801.10247v1.

[6] LI G H, MULLER M, THABET A, et al. DeepGCNs: Can GCNs go as deep as CNNs? [C]// International Conerence on Computer Vision 2019, October 27-November 3, 2019, Seoul, South Korea. Piscataway, NJ: IEEE, 2019: 9266-9275.

[7] ZHOU L, WANG T Y, QU H, et al. A weighted GCN with logical adjacency matrix for relation extraction[C]//The 24th European Conference on Artificial Intelligence, August 29-September 8, 2020, Santiago de Compostela, Spain. San Francisco, CA: Morgan Kaufmann, 2020: 2314-2321.

[8] RONG Y, HUANG W B, XU T Y, et al. DropEdge: Towards deep graph convolutional networks on node classification [EB/OL]. (2020-03-12)[2022-03-01]. https://arxiv.org/abs/1907.10903v3.

[9] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition, June 8-10, 2015, Boston, Massachusetts. New York: IEEE, 2015: 15524435.

[10] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[EB/OL]. (2017-02-22)[2022-03-01]. https://arxiv.org/.abs/1609.02907

[11] LI Q M, HAN Z C, WU X M. Deeper insights into graph convolutional networks for semi-supervised learning [C]//The Thirty-Second AAAI Conference on Artificial Intelligence, February 2-7, 2018, New Orleans, USA. Menlo Park, CA: AAAI, 2018: 3538-3545.

[12] MOHAMMAD E, VICENTE O, JOSEPH R M, et al. XNOR-net: Image-net classification using binary convolutional neural networks[C]//The 14th European Conference on Computer Vision, October 8-10, 2016, Amsterdam, The Netherlands. Paris: Springer, 2016: 525-542.

[13] LIU Z C, WU B Y, LUO W H, et al. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm[C]//The 15th European Conference on Computer Vision, September 8-14, 2018, Munich, Germany. Berlin: Springer, 2018: 747-763.

[14] WILLIAM L H, YING Z T, JURE L. Inductive representation learning on large graphs[C]//Neural Information Processing Systems, December 4-9, 2017, Los Angeles, USA. Cambridge: MIT Press, 2017: 1024-1034.

[15] ZENG HQ , ZHOU H K, AJITESH S, et al. GraphSAINT: Graph sampling based inductive learning method[EB/OL]. (2020-02-16) [2022-03-01]. https://arxiv.org/abs/1907.04931v2.

[16] YANG L, KANG Z S, CAO X C, et al. Topology optimization based graph convolutional network[C]//The 28th International Joint Conference on Artificial Intelligence, August 10-16, 2019, Macao, China. San Francisco: Morgan Kaufmann, 2019: 4054-4061.

[17] YAO L, MAO C S, LUO Y. Graph convolutional networks for text classification[C]//The Thirty-Third AAAI Conference on Artificial Intelligence, January 27-February 1, 2019, Hawaii, USA. Menlo Park, CA: AAAI, 2019: 7370-7377.

[18] SEN P, NAMATA G, BILGIC M, et al. Collective classification in network data[J]. AI Magazine, 2008, 29(3): 93-106.

[19] YANG Z L, WILLIAM W C, RUSLAN S, et al. Revisiting semi-supervised learning with graph embeddings[C]//The 33rd International Conference on Machine Learning, June 19-24, 2016, New York, USA. New York: ACM, 2016: 40-48.

[20] WANG J F, WANG Y H, YANG Z, et al. Bi-GCN: binary graph convolutional network[C]//The Thirty-Four IEEE Conference on Computer Vision and Pattern Recognition, June 19-25, 2021, ONLINE. Piscataway, NJ: IEEE, 2021: 73-77.

[21] CHEN J, MA T F, XIAO C. FastGCN: Fast learning with graph convolutional networks via importance sampling [EB/OL]. (2018-01-30)[2022-03-01]. https://arxiv.org/abs/1801.10247v1.

[22] PETAR V, GUILLEM C, ARANTXA C, et al. Graph attention network[EB/OL]. (2018-02-04)[2022-03-01]. https://arxiv.org/abs/1710.10903.

作者简介：

苏树智 (1987—)，男，博士，副教授，硕士生导师，主要从事模式识别、图像处理、图神经网络、深度学习方面的研究.