深层图注意力对抗变分自动编码器

翁自强 张维玉* 孙 旭
(齐鲁工业大学(山东省科学院)计算机科学与技术学院 山东 济南 250353)

摘 要 现有的图自动编码器忽视了图邻居节点的差异和图潜在的数据分布。为了提高图自动编码器嵌入能力,提出图注意力对抗变分自动编码器(AAVGA-d),该方法将注意力引入编码器,并在嵌入训练中使用对抗机制。图注意力编码器实现了对邻居节点权重的自适应分配,对抗正则化使编码器生成的嵌入向量分布接近数据的真实分布。为了加深图注意力层数,设计一种针对注意力网络的随机边删除技术(RDEdge),减少了层数过深引起的过平滑信息丢失。实验结果表明,AAVGA-d 的图嵌入能力与目前流行的图自动编码器相比具有竞争优势。

关键词 图注意力 过平滑 自动编码器 对抗

中图分类号 TP391 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.09.023

DEEP GRAPH ATTENTION ADVERSARIAL VARIATIONAL AUTOENCODER

Weng Ziqiang Zhang Weiyu* Sun Xu
(*School of Computer Science and Technology*, *Qilu University of Technology* (*Shandong Academy of Sciences*), *Jinan 250353*, *Shandong*, *China*)

Abstract The existing graph autoencoder ignores the difference between the neighbor nodes of the graph and the potential data distribution of the graph. In order to improve the embedding ability of the graph autoencoder, the graph attention adversarial variational autoencoder (AAVGA-d) is proposed. This method introduced attention to the encoder and used an adversarial mechanism in the embedding training. The graph attention encoder realized the adaptive allocation of the weights of neighbor nodes, and the adversarial regularization made the distribution of the embedding vector generated by the encoder close to the true distribution of the data. In order to deepen the number of graph attention layers, a random edge deletion technology (RDEdge) for attention networks was designed to reduce the loss of over-smooth information caused by excessively deep layers. The experimental results prove that the graph embedding capability of AAVAG-d has a competitive advantage compared with the current popular graph autoencoders.

Keywords Graph attention Over-smoothing Autoencoder Adversarial

0 引 言

与诸如图像、文本和语音数据之类的欧几里得空间数据相比,图数据这种非欧几里得数据很难处理。因此,图嵌入算法已经成为研究的热点。图研究集中于节点分类[1]、链接预测[2]、图分类[3]和图生成[4]等任务,图嵌入算法可分为图分解、随机游走和图神经网络三类。

最近,图嵌入算法已经进入神经网络时代。Kipf 等[1]简化了频域卷积的定义,并提出了在空域进行卷积运算的 GCN,这极大地提高了图卷积模型的嵌入能力。从那时起,研究人员已经提出了 GCN 的许多变体。GraphSAGE[5]并不将采样限制在节点的拓扑结构信息中,相反,它利用了节点的内在特征并放弃了涉及大量参数的扩散机制,从而实现了大规模图数据的分布式训练和归纳学习。图注意力网络[6](GAT)使用注意力机制在邻居节点上进行聚合操作以自适应地分

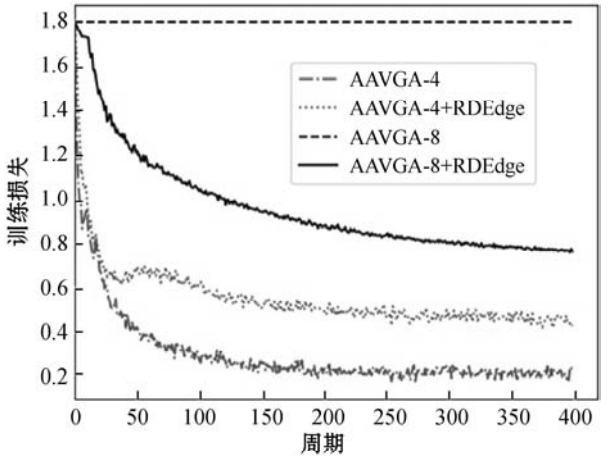
收稿日期:2021 - 04 - 24。国家重点研发计划项目(2018YFC0831704);国家自然科学基金项目(61806105);山东省自然科学基金项目(ZR2017MF056)。**翁自强**,硕士生,主研领域:数据挖掘,图神经网络。**张维玉**,副教授。**孙旭**,硕士生。

配邻居权重。

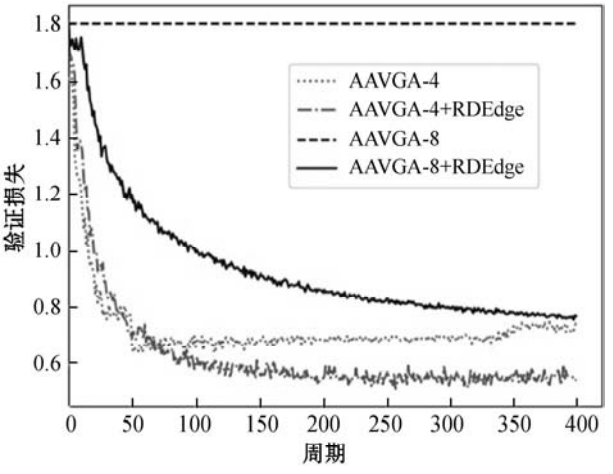
上述方法是有监督图嵌入方法。近年来,图数据的应用变得越来越广泛,图的结构更加复杂。在实际应用场景中,许多数据标签通常具有较高的采集阈值。因此,研究在图数据上进行有效的无监督图表示学习具有重要的价值。基于重构损失的图自动编码器是一种典型的无监督学习方法。GAE 和 VGAE[7] 使用编码器获得潜在向量,而解码器使用潜在变量来重构图结构。由于图数据的高维和复杂分布的特性,利用编码器编码得到的潜在向量的分布与实际分布存在偏差。为了解决有关编码数据分布的问题,DVNE[8] 直接根据高斯分布嵌入节点,并使用 Wasserstein 距离作为分布之间的相似性度量,从而有效地建模了网络中节点的不确定性。ARGA 和 ARVGA[9] 进一步引入了对抗机制[10],该机制通过对抗训练迫使编码器生成更接近数据真实分布的潜在向量。尽管这些自动编码器取得了一定的成果,但它们并未考虑节点重要性的差异。由于各个邻居节点重要性的不同,因此要学习鲁棒和稳定的节点嵌入,应在聚合过程中自适应分配相邻节点的权重。

为了解决这一问题,本文专注于邻居节点的差分表示,提出图注意力对抗变分自动编码器(AAVGA)。目的是区分图结构信息,并应用对抗正则化机制以提高模型的图嵌入能力。编码器通过图注意层来生成潜在的特征向量,利用权重的自适应分配在嵌入过程中差异化节点表示,本文添加了多组独立的注意力机制,从而使注意力聚合更为稳定。为了规范化编码数据的分布,我们将对抗机制引入到基于注意力的图变分自动编码器中。该组件可以确定输入是来自图网络的低维表示还是来自样本的真实分布。鉴别器鼓励编码器生成具有更接近数据真实分布的低维变量,并学习图的有效表示形式。

然而,当我们重新审视各类图嵌入方法时,无论是有监督图嵌入还是无监督图嵌入都采用了浅层的图神经网络(通常为 2 层)。在面对大型图嵌入学习时,加深图神经网络层数十分必要。受到深层 CNN 在图像分类上的成功的启发,研究者已经提出了一些尝试来探索如何构建深层 GCN 的想法,包括 GCN[1]、GraphSAGE[5]、ResGCN[11] 和 JKNet[12]。然而,它们都没有提供详细表达的架构。过拟合和过平滑是阻碍图模型加深的两大问题。过度拟合来自以下情况:当使用超参数模型来拟合训练数据有限的分布时,所学习的模型非常适合训练数据,但对测试数据的推广却很差,在小图上应用深层 GCN 时,过拟合问题尤为突出。过度平滑,朝另一个极端发展,使得训练深层 GCN 非常困难。正如 Li 等[13] 首先对其进行介绍,Xu 等[14] 进一步解释,图卷积的本质是聚合,如果使用的层数不限,则所有节点的表示都将收敛到一个固定点,这将使结果与输入要素隔离,并导致梯度消失,这种现象被称为过平滑。为了更直观地表明其影响,本文在图 1 中使用 4 层和 8 层图注意力层的 AAVGA 进行示例实验,可以观察到图注意力层叠加到一定程度时会出现明显的过拟合现象,而当层数进一步加深时,过平滑现象出现,模型无法收敛。为了缓解上述两个问题,我们引入了随机边删除技术(RDEdge),这能帮助模型在每个训练周期随机丢弃输入图的某些边。



(a) 训练损失



(b) 验证损失

图 1　不同注意力层数的 AAVGA 在 Cora 数据集上的损失

本文将 RDEdge 视为一种数据增强技术。通过 RDEdge 生成原始图的不同随机变形副本。这增强了输入数据的随机性和多样性,因此能够更好地防止过度拟合。RDEdge 还可以被视为消息传递缩减器,在图注意力层中,相邻节点之间消息的传递是沿着边进行的。删除某些边使节点连接更加稀疏,因此在图

注意力层变深时可以在某种程度上避免过度平滑。RDEdge 会对图模型训练带来很多帮助,如图 1 所示,在结合了 RDEdge 之后,AAVGA 可以很好地应对过拟合和过平滑问题。这允许本文进一步加深模型的编码层,提高模型的图嵌入能力。本文将图注意力对抗自动编码器(AAVGA)与随机边删除技术(RDEdge)结合,加深编码器的图注意力层数,并有效应对过拟合和过平滑问题,进一步提升 AAVGA 的图嵌入能力,通过链路预测实验证明了模型的有效性。

1 相关工作

1.1 浅层图神经网络

图信号处理(GSP)[15]将信号处理的基本概念(例如傅里叶变换和滤波)移植到了图上,以实现图信号结构的压缩、变换和重构,这为图嵌入学习奠定了基础。受到 GSP 中图信号卷积滤波定义的启发,研究人员已经开发了一系列基于图卷积运算的神经网络模型。Bruna 等[16]将卷积引入图神经网络,并开发了基于频域卷积运算概念的图卷积网络模型。从那时起,研究人员就不断提出基于频域图卷积的神经网络的改进和扩展模型[17-20]。

图神经网络(GNN)具有强大的端到端学习能力,可以与相应的无监督损失函数结合使用,以实现无监督图表示学习。根据设计,损失函数可以分为两类:基于对比的损失函数和基于重建的损失函数。基于对比损失的方法,例如:GraphSAGE[5]使用邻居作为上下文;Hu 等[21]使用子图作为对比学习的上下文;DGI[22]将整个图视为上下文。基于重建损失的方法,例如:VGAE[7],它基于 VAE[23]并使用 GNN 编码和学习图数据;ARVGA[9]引入了一种对抗机制,用于训练鉴别器并强制编码器生成的图数据表示向量服从先验分布。但是,这些框架没有考虑图邻居节点之间的差异,这可能会导致原始图结构在编码过程中被破坏。本文引入注意机制来解决这个问题。受到图注意力网络(GAT)的启发[6],引入注意机制来解决这个问题。利用注意力机制对邻居节点进行聚合操作,实现邻居权重的自适应分配。与 GraphSAGE[5]相似,图注意力模型保留了完整的结构信息,也可以进行归纳学习。此外,本文结合了 GAT[6]、VGAE[7]和 GAN[10]的策略,在确保编码向量符合数据先验分布的同时,提高了图编码器的表达能力。模型中的对抗策略源自 GAN[10]:在 minimax 游戏中,生成器和鉴别器联合训练并优化,通过对抗训练可以增强图嵌入模型的泛化性能;GraphGAN[24]是第一个在图嵌入学习中利用对抗策略的网络;ANE[25]将嵌入向量视为生成的结果,并通过将实际数据的分布假设为先验分布,在可用的网络嵌入方法(如 DeepWalk[26])中将 GAN[10]用作附加的正则化项;ARVGA[9]在变分自动编码器中使用了上述策略,但没有考虑节点之间重要性的差异。本文方法改善了这一点。

1.2 深层图神经网络

尽管研究者们在图神经网络领域取得了丰硕的成果,但先前大多数的工作只针对浅层 GNN,而很少讨论较深的扩展。建立深层 GNN 的尝试可追溯到 GCN[1],其中应用了残差机制。然而,如他们的实验所示,当深度为 3 或更大时,残差的 GCN 仍然表现较差。Xu 等[14]首先指出了构建过度平滑的深层 GNN 的主要困难,但是,他们没有提出任何解决方法。在后续研究中 Klicpera 等[27]通过使用个性化的 PageRank 来应对过平滑问题,该过程还将根节点纳入了消息传递循环中,但是,当深度从 2 开始增加时,仍会观察到精度降低。JKNet[13]采用密集连接进行多跳消息传递,是另一种可以潜在防止过度平滑的工具。在其表述中,JKNet 将每个隐藏层紧密地连接到顶层,因此仍保留了几乎不受过度平滑影响的较低层中的特征映射。Oono 等[28]从理论上证明了深层 GCN 的节点特征将收敛到子空间并引起信息丢失,它通过进一步考虑 ReLU 函数和卷积滤波器概括了 Li 等[13]的结论。最近的一种方法[11]将残差层、密集连接和膨胀卷积整合到 GCN 中,以促进深度架构的开发。尽管如此,该模型是针对图级分类的,其中数据点是图并且自然地彼此断开。在本文的链路预测任务中,样本是节点,并且它们彼此耦合,因此更需要解决过度平滑的问题。通过利用 RDEdge,可以加深 AAVGA 的注意力编码层,提高模型的图嵌入能力。

2 深层图注意力对抗变分自动编码器

本节将首先介绍所提出的深层图注意力对抗变分自动编码器(AAVGA-d)。该框架的主要结构如图 2 所示。主要包括三个组件:编码器、解码器和鉴别器。其次,将对随机边删除技术(RDEdge)的步骤及原理进行说明。

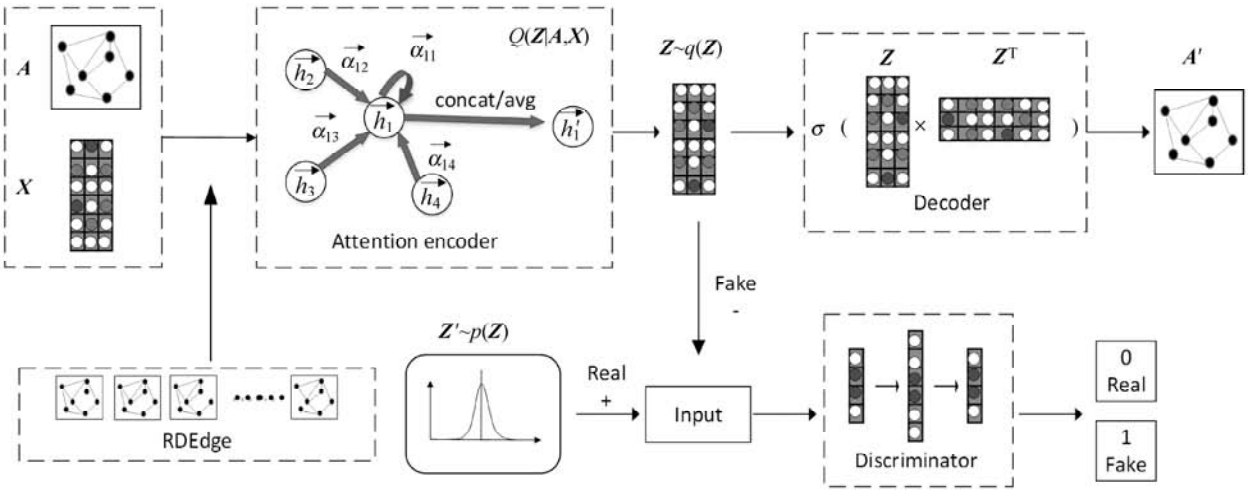


图 2　AAVGA-d 模型框架

2.1　图注意力变分自动编码器

2.1.1　图注意力编码器

VGAE[7] 和 ARVGA[9] 使用原始的两层 GCN[1] 作为编码器。本文提出的 AAVGA-d 结合了 GAT[6] 的策略,并用多层图注意力网络代替了通用编码器中的两层图卷积网络,以生成图数据的潜在表示。形式定义如下:令与 L 层中节点 v_i 对应的特征向量为 $\boldsymbol{h}_i$,$\boldsymbol{h}_i \in \mathbf{R}^{d^{(l)}}$,其中 $d^{(l)}$ 表示节点的特征长度。经过以注意力机制为核心的聚合操作后,输出每个节点的新特征向量 $\boldsymbol{h}'_i$,其中 $\boldsymbol{h}'_i \in \mathbf{R}^{d^{(l+1)}}$,$d^{(l+1)}$ 表示输出特征向量的长度,这里将此聚合操作称为图注意力层。假设中心节点为 v_i,将相邻节点 v_j 到 v_i 的权重系数设置为:

$$\boldsymbol{e}_{ij} = a(\boldsymbol{W}\boldsymbol{h}_i, \boldsymbol{W}\boldsymbol{h}_j) \tag{1}$$

式中:$\boldsymbol{W} = \mathbf{R}^{d^{(l+1)} \times d^{(l)}}$ 是该层节点特征变换的权重参数;而 $a(\cdot)$ 是计算两个节点之间相关性的函数。原则上,这里可以计算图中任何节点到节点 v_i 的权重参数;但是,为了简化计算,只将其限制为一阶邻居。a 可以使用向量的内积来定义无参数的相关性计算 $\langle \boldsymbol{w}\boldsymbol{h}_i, \boldsymbol{w}\boldsymbol{h}_j \rangle$。或者,可以将其定义为有参数的神经网络层。如果满足 $a: \mathbf{R}^{d^{(l+1)}} \times \mathbf{R}^{d^{(l+1)}} \rightarrow \boldsymbol{R}$,则输出代表两个矢量之间相关性的标量值。这里本文选择单层全连接层作为相关性函数:

$$\boldsymbol{e}_{ij} = \text{Leaky ReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i \parallel \boldsymbol{W}\boldsymbol{h}_j]) \tag{2}$$

式中:权重参数是 $\boldsymbol{a} \in \mathbf{R}^{2d^{(l+1)}}$,激活函数是 LeakyReLU。为了更好地分配权重,必须通过 softmax 归一化对所有邻居的相关性计算进行归一化:

$$\boldsymbol{\alpha}_{ij} = \text{softmax}_j(\boldsymbol{e}_{ij}) = \frac{\exp(\boldsymbol{e}_{ij})}{\sum\limits_{v_k \in \tilde{N}(v_i)} \exp(\boldsymbol{e}_{ik})} \tag{3}$$

式中:α_{ij} 是权重系数。通过式(3)的处理,可以保证所有邻居的权重系数之和为 1。结合式(2)和式(3)可以得到完整的权重系数计算公式:

$$\boldsymbol{\alpha}_{ij} = \frac{\exp(\text{Leaky ReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i \parallel \boldsymbol{W}\boldsymbol{h}_j]))}{\sum\limits_{v_k \in \tilde{N}(v_i)} \exp(\text{Leaky ReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i \parallel \boldsymbol{W}\boldsymbol{h}_j]))} \tag{4}$$

计算得到上述权重系数后,根据注意力机制的加权求和策略,得到节点 v_i 的新特征向量:

$$\boldsymbol{h}'_i = \sigma\Big(\sum_{v_j \in \tilde{N}(v_i)} \boldsymbol{\alpha}_{ij} \boldsymbol{W}\boldsymbol{h}_j\Big) \tag{5}$$

为了进一步提高注意力层的表达能力,本文还在 AAVGA-d 中引入了多头注意力机制,其中式(5)用于形成 K 组独立的注意力机制。与 GAT[6] 相比,为了减小输出潜在特征向量的维数,使用平均运算替换了拼接运算:

$$\boldsymbol{h}'_i = \sigma\Big(\frac{1}{k}\sum_{k=1}^{k}\sum_{v_j \in \tilde{N}(v_i)} \boldsymbol{\alpha}_{ij}^{(k)} \boldsymbol{w}^{(k)} \boldsymbol{h}_j\Big) \tag{6}$$

通过汇总多组独立的注意力机制,多头注意力机制可以将注意力分布施加于中心节点和邻居节点之间的多个相关特征上,从而增强了编码器的表示能力。本文使用基于注意力机制的编码器来拟合 $\boldsymbol{\mu}$ 和 $\boldsymbol{\sigma}$:

$$\boldsymbol{\mu} = GNN_{\mu}(\boldsymbol{X}, \boldsymbol{A}) \tag{7}$$

$$\log\boldsymbol{\sigma} = GNN_{\sigma}(\boldsymbol{X}, \boldsymbol{A}) \tag{8}$$

式中:$\boldsymbol{\mu}$ 是均值向量 $\boldsymbol{\mu}_i$ 的矩阵;$\log\boldsymbol{\sigma}$ 在注意力层中与 $\boldsymbol{\mu}$ 共享权重 $\boldsymbol{w}$。变分图自动编码器的定义如下:

$$P(\boldsymbol{Z} \mid \boldsymbol{X}, \boldsymbol{A}) = \prod_{i=1}^{N} q(\boldsymbol{z}_i \mid \boldsymbol{X}, \boldsymbol{A}) \tag{9}$$

$$q(\boldsymbol{z}_i \mid \boldsymbol{X}, \boldsymbol{A}) = \mathcal{N}(\boldsymbol{z}_i \mid \boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2)) \tag{10}$$

2.1.2　解码器

对于解码器,本文遵循 VGAE[7] 和 ARVGA[9] 的策略,目的是通过解码器重建图 $\boldsymbol{A}$。

$$\boldsymbol{Z} = GNN(\boldsymbol{X}, \boldsymbol{A}) \tag{11}$$

$$\hat{\boldsymbol{A}} = \sigma(\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}) \tag{12}$$

$\hat{\boldsymbol{A}}$ 使用图低维表示向量的内积来重构邻接关系,

从而预测图中任何两个点间是否存在连接边。

$$P(\boldsymbol{A}\mid\boldsymbol{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{N}p(\boldsymbol{A}_{ij}\mid\boldsymbol{z}_i,\boldsymbol{z}_j) \tag{13}$$

$$P(\boldsymbol{A}_{ij}=1\mid z_i,z_j)=\sigma(\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j) \tag{14}$$

2.1.3 损失函数

根据经验,选择标准正态分布作为图的潜在变量 z 的先验分布:

$$P(\boldsymbol{Z}) = \prod_i p(\boldsymbol{z}_i) = \prod_i \mathcal{N}(\boldsymbol{z}_i,0,\boldsymbol{I}) \tag{15}$$

完整的损失函数定义如下:

$$\mathcal{L}=\mathcal{L}_{\mathrm{recon}}+\mathcal{L}_{kl} = -\mathbb{E}_{q(\boldsymbol{Z}\mid\boldsymbol{X},\boldsymbol{A})}[\log p(\boldsymbol{A}\mid\boldsymbol{Z})]+ KL[q(\boldsymbol{Z}\mid\boldsymbol{X},\boldsymbol{A})\parallel p(\boldsymbol{Z})] \tag{16}$$

如果仅将 $\mathcal{L}_{\mathrm{recon}}$ 用作损失函数来优化编码器的性能,则模型获得的方差值为零,因为从固定的正态分布中采样,将会减少生成的样本数以及样本之间的实际差异。但是,主要目标是优化变分自动编码器。为了实现这一目标,本文在损失函数中加入潜在向量的分布和标准正态分布的 KL 散度,迫使每个潜在向量的分布逼近标准正态分布。

2.2 联合编码器的对抗机制

本文的对抗模型由两个部分组成:图注意自动编码器中的编码器充当对抗网络的生成器。生成器尝试通过生成伪造数据来欺骗鉴别器,其中伪造数据是指编码器由图数据编码获得的潜在变量。生成器的损失如式(16)所示。鉴别器的任务是区分样本来自真实数据还是生成器。鉴别器将来自先验分布 p_z 输出的数据判断为正,将来自潜在变量 $\boldsymbol{z}$ 输出的数据判断为负,其损失如下:

$$-\frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_z}\log D(\boldsymbol{Z})\frac{-1}{2}\mathbb{E}_{\boldsymbol{X}}\log(1-D(G(\boldsymbol{X},\boldsymbol{A}))) \tag{17}$$

本文使用高斯分布作为图数据的先验分布,并假设编码器生成的潜在向量 $\boldsymbol{z}$ 不满足数据在欧几里得空间中的先验分布;因此,使用标准的多层感知器作为鉴别器。在嵌入和学习的过程中,对抗性正则化约束被施加以减少训练过程中数据分布的偏差。对抗模型的主要目标是通过 minimax 游戏共同训练编码器和鉴别器,以使它们彼此优化:

$$\min_G\max_D \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log D(\boldsymbol{Z})]+ \mathbb{E}_{\boldsymbol{x}\sim p(x)}[\log(1-D(G(\boldsymbol{X},\boldsymbol{A})))] \tag{18}$$

训练 D 以最大程度地从训练数据和 G 中正确地区分样本。同时,训练 G 以最小化对数 $(1-D(G(\boldsymbol{Z})))$。

2.3 随机边删除技术

为了应对加深图注意力层所带来的过拟合和过平滑问题,本文设计一种针对图注意力模型的随机边删除技术(RDEdge)。DropEdge[29]已经为图模型的抗过平滑做出了贡献,但是它适用的对象是深层 GCN[1],由于图注意力在聚合过程中并不依赖图的拉普拉斯矩阵,所以 DropEdge[29]并不适用于深层图注意力模型的抗过平滑。为此本文的工作专注于深层图注意模型,提出 RDEdge。在每次训练中,RDEdge 技术都会随机删除输入图的一定比例的边。具体而言,它随机地将邻接矩阵 $\boldsymbol{A}$ 的 V_p 个非零元素重置为零,其中:V 是边的总数;p 是删除率。如果将随机删除边后的邻接矩阵表示为 $\boldsymbol{A}_{\mathrm{rd}}$,则其与原图邻接矩阵 $\boldsymbol{A}$ 的关系为:

$$\boldsymbol{A}_{\mathrm{rd}}=\boldsymbol{A}-\boldsymbol{A}' \tag{19}$$

式中:$\boldsymbol{A}'$是原图邻接矩阵 $\boldsymbol{A}$ 的子集,大小为 V_p 的一个稀疏矩阵。在图注意力的聚合过程中,邻接矩阵 $\boldsymbol{A}$ 并不直接参与运算,但是需要借助邻接矩阵 $\boldsymbol{A}$ 来识别邻居节点。在图注意力层通过式(2)得到未归一化的注意系数 $\boldsymbol{e}_{ij}$ 后,需要进行掩膜操作(mask·):

$$\boldsymbol{e}_{ij\mathrm{rd}}=\boldsymbol{A}_{\mathrm{rd}}\odot\boldsymbol{e}_{ij} \tag{20}$$

本文将经过 mask 操作得到后的 $\boldsymbol{e}_{ij\mathrm{rd}}$ 送入式(3)中进行归一化操作就可以得到各邻居节点的权重系数 $\boldsymbol{\alpha}_{ij\mathrm{rd}}$。最后通过得到的权重系数进行图注意力聚合就可以实现 RDEdge 对深层图注意力模型的优化。

对于 GNN 的过平滑问题,Oono 等[28]认为随着层数的加深,节点的表示最终会收敛到一个子空间。本文将借助这一概念对 RDEdge 的抗过平滑能力进行理论分析。首先给出以下定义:

定义 1 子空间。令 $\mathcal{M}:=\{(\boldsymbol{EC}\mid\boldsymbol{C}\in\mathbf{R}^{M\times C}\}$ 是空间 $\boldsymbol{R}^{N\times C}$ 中的 M 维子空间(N 是节点数,C 是节点特征的维度。其中 $\boldsymbol{E}\in\mathbf{R}^{N\times M}$ 是正交矩阵,并有 $\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}=\boldsymbol{I}_M$,$M\leqslant N$。

式中:投影矩阵 $\boldsymbol{P}\in\mathbf{R}^{N\times d_w}$,$d_w$ 指上下文用户的特征维数,投影矩阵中的每一行 P_i,表示用户上下文的唯一转换。

定义 2 过平滑。给定独立于输入特征的子空间 $\mathcal{M}$,若第 l 层的隐藏矩阵 $\boldsymbol{H}^{(l)}$ 中的所有向量的距离不超过 $\epsilon(\epsilon>0)$,则称 GNN 中的节点特征发生了过拟合现象。

$$l^*(\mathcal{M},\epsilon):=\min_l\{\boldsymbol{d}_{\mathcal{M}}(\boldsymbol{H}^{(l)})<\epsilon\} \tag{21}$$

定义 3 令原始图为 $\mathcal{G}$,经过 RDEdge 随机删除边后的图为 $\mathcal{G}'$ 给定最小值 ϵ,假定 $\mathcal{G}$ 和 $\mathcal{G}'$ 于子空间 $\mathcal{M}$ 和 $\mathcal{M}'$ 遇到过平滑问题,则有式(22)、式(23)删除足够多边后成立。

$$\hat{l}(\mathcal{M},\epsilon)\leqslant\hat{l}(\mathcal{M}',\epsilon) \tag{22}$$

$$N-\dim(\mathcal{M})>N-\dim(\mathcal{M}') \tag{23}$$

根据 Oono 等[28]的结论,深层 GNN 在一定条件下,对于任意小的ϵ值,都会有过平滑问题,但是他们没有提出对应的解决方案。本文从两个角度说明 RDEdge 有助于缓解过平滑问题:

(1) 通过减少节点间的连接,RDEdge 可以降低过平滑的收敛速度,提高了ϵ-平滑层的上界。

(2) 原始空间和收敛子空间的维度之差($N-M$)衡量了信息的损失量,差值越大说明信息损失越严重。RDEdge 可以增加子空间的维度,具有减少信息损失的能力。

2.4　AAVGA-d 框架

在每次训练前,运用 RDEdge 技术对图的邻接矩阵 $\boldsymbol{A}$ 进行稀疏处理得到 $\boldsymbol{A}_{\mathrm{rd}}$。通过多层图注意力层拟合图的节点特征 $\boldsymbol{X}$ 得到未归一化的注意力系数 $\boldsymbol{e}_{ij}$。紧接着进行 RDEdge 最为关键的一步:将 $\boldsymbol{A}_{\mathrm{rd}}$ 和 $\boldsymbol{e}_{ij}$ 进行掩膜操作得到 $\boldsymbol{e}_{ij\mathrm{rd}}$。之后,对 $\boldsymbol{e}_{ij\mathrm{rd}}$ 进行归一化处理得到最后的注意力权重系数 $\alpha_{ij}^{(k)}$。对图进行注意力聚合得到图的低维表示矩阵 $\boldsymbol{Z}$,进一步对表示矩阵 $\boldsymbol{Z}$ 和数据的先验分布 p_z 进行采样,样本用于训练鉴别器。在训练中尝试使用编码器欺骗鉴别器,这也可以理解为用鉴别器训练编码器,以使编码器生成的数据分布更接近真实分布。最后,使用 AAVGA-d 的总体损失函数来训练整个模型。

3　实验与结果分析

3.1　数据集

本文在图嵌入学习中使用了三个最受欢迎的引文数据集(Cora、Citeseer 和 Pubmed)来评估提出的模型。数据集的结构在表 1 中进行了描述。节点对应于数据集中的论文,特征是每篇论文的特点,边缘代表论文之间的链接关系。以 Cora 数据集为例:Cora 数据集由机器学习论文组成,并且近年来在图深度学习中非常受欢迎。在数据集中,论文分为以下七个类别:基于案例、遗传算法、神经网络、概率方法、强化学习、规则学习和理论。语料库中有 2 708 篇论文,论文选择标准是最终语料库中至少添加了对另一篇论文的引用。语料库中有 2 708 篇论文。在数据集中有一个包含多个单词的词汇表,在删除开头和结尾后,仅保留了 1 433 个唯一词,文档频率小于 10 的所有单词都将被删除。Cora 数据集包含 1 433 个唯一词,因此,特征是 1 433 维,论文中单词的不存在和存在分别用 0 和 1 表示。Citeseer 和 Pubmed 数据集的结构与 Cora 数据集相似。

表 1　三个引文数据集的统计信息

信息	Cora	Citeseer	Pubmed
节点	2 708	3 327	19 717
边缘	5 429	4 732	44 338
特征	1 433	3 703	500
标签	7	6	3

3.2　评价指标及基准

链路预测算法在经过训练后可以得到网络中每一对节点的相似值(即边的相似值)。本文将 ROC 曲线和 x 轴所包围的图形区域视为综合测量指标,称为 AUC。AUC 可以理解为在测试集中边的相似值比实际不存在边的相似值高的概率。具体而言,每次随机从测试集中选取一条边与随机选择的不存在的边进行比较,如果测试集中的边的相似值大于不存在的边的相似值,就加 1 分;如果两个分数值相等,就加 0.5 分。独立地比较 n 次,如果有 n_1 次测试集中的边的相似值大于不存在的边的相似值,有 n_2 次两相似值相等,则 AUC 定义为:

$$A_{\mathrm{UC}}=\frac{n_1+0.5n_2}{n} \tag{24}$$

另一个评估指标是 AP,代表 Precision-recall(PR)曲线和 x 轴所包围的图形区域。Precision 和 Recall 定义如下:

$$P_{\mathrm{recision}}=\frac{T_{\mathrm{P}}}{a_{\mathrm{ll\ detections}}} \tag{25}$$

$$R_{\mathrm{ecall}}=\frac{T_{\mathrm{P}}}{a_{\mathrm{ll\ groundtruths}}} \tag{26}$$

式中:T_{P} 表示预测存在连接且预测正确的节点对数;$a_{\mathrm{ll\ detections}}$ 表示预测存在连接的节点对数;$a_{\mathrm{ll\ ground\ truths}}$ 表示实际存在连接的节点对数。

上述两个指标是链路预测任务的主要评估指标。本文将数据集分为训练集、验证集和测试集。验证集中包含 5% 的边用于超参数优化,测试集中包含有 10% 的边用于评估性能。为了确保准确性,对每个数据集进行了 10 次实验,计算得到实验结果的平均数值。

为了验证本文提出的 AAVGA-d 框架具有竞争性的图嵌入能力,将其与六种流行的图嵌入算法进行了比较:

(1) Spectral Clustering[30]。这是基于图论的聚类方法,加权无向图被分为两个或多个最佳子图,以使子图的内部尽可能相似,并且子图之间的距离尽可能大,以实现共同的聚类目标。

(2) Deep Walk[26]。如果网络的顶点很少,则通

过截断的随机游走来学习社交表示会产生更好的结果,并且该方法还具有可扩展性并且可以适应网络的变化。

(3) GAE[7]。无监督图嵌入学习的代表,基于重构损失通过编码和解码来学习输入图数据的有效表示形式。

(4) VGAE[7]。编码器不再学习样本的低维向量表示,它学习的是样本表示的分布。假设此表示遵循正态分布,然后,从学习得到的分布中采样以获得低维向量表示。

(5) ARGA[9]。在图自动编码器的基础上增加了对抗机制,以确保训练过程中数据分布的一致性。

(6) ARVGA[9]。在图变分自动编码器的基础上引入了对抗机制,直接从真实数据分布中采样,并通过鉴别器与编码器得到的潜在向量进行分布差异辨别。

对于上述基准测试方法,遵循相应论文中的参数设置。

3.3 实验设置

与原始的 AAVGA 不同的是,AAVGA-d 加深了注意力层数,对于 Cora 和 Citeseer 两个较小的引文数据集,AAVGA-d 由原来的单注意力层变成 4 层,第一层的神经元个数设为 64,往后各层依次除 2 递减。在嵌入过程中将注意力的 head 数设为 3,即有着 3 个独立的注意力系数矩阵,将学习得到的 3 个表示向量作加和平均处理。在优化时选用 Adam 算法,编码器和鉴别器的学习率设为 0.001,RDEdge 的删除率设为 10%,共进行 400 迭代训练。由于 Pubmed 数据集的数据规模远大于前两个数据集,为提高编码器的表达能力,本文将图注意力层加深到 8 层,第一层神经元个数设为 256,往后各层依次除 2 递减,注意力的 head 数设为 6。RDEdge 的删除率设为 20%,共进行 1 000 次迭代训练。其余参数与 Cora、Citeseer 保持一致。在训练完成后,输出图的嵌入向量 $\boldsymbol{Z}$,可以将 $\boldsymbol{Z}$ 理解为某种意义上图节点的相似度,通过向量内积得到邻接矩阵 $\boldsymbol{A}$,$\boldsymbol{A}$ 表示节点对间存在边的概率,将预测概率与测试集中的正负样本(正样本代表两点间存在连接边为 1,负样本代表两点间不存在连接边为 0)进行计算得到 AUC 和 AP 指标。

3.4 实验结果

在本文模型中,通过使用 Cora、Citeseer 和 Pubmed 三个数据集进行分析实验。以 Cora 数据集为例,数据集由 2 078 篇机器学习论文组成,文章间的引用数达到 5 429 次,并划分出了 1 433 个词汇。从图的视角观察,数据集拥有 2 078 个定点、1 433 维特征和 5 429 条边。本文使用深层图注意力对抗变分自动编码器对数据进行嵌入学习,首先,对数据进行 one-hot 编码得到邻接矩阵 $\boldsymbol{A}$ 和特征 $\boldsymbol{X}$,通过深层注意力编码器对特征进行嵌入得到表示向量,利用注意力权重分配机制充分考虑了邻居节点间的重要性差异,对于相似节点(具有多个相同词汇)在聚合过程中赋予较大权重,对于不相似节点(相同词汇较少)赋予较小权重。其次,在嵌入学习中使用鉴别器对编码器进行对抗监督,迫使得到的表示向量服从 Cora 数据集的真实分布,从而得到更为准确的嵌入结果。最后,利用得到的图表示向量进行重建得到预测矩阵,矩阵中的数据代表两篇文章间是否存在引用关系。通过编码器、解码器、监督器三者的联合训练最终实现了图嵌入性能的提升。

链接预测实验的结果显示在表 2 - 表 4 中。本文方法 AAVGA-d 在三个数据集上均得到了出色结果。与 AAVGA 相比,运用了 RDEdge 技术的 AAVGA-d 图嵌入性能更佳,三个数据集的 AP 和 AUC 均实现了超越。这表明加深图注意力层并运用 RDEdge 技术抗过拟合和抗过平滑的策略是可行的。除 Cora 的 AUC 外,数据集上的其他指标均超过 94%。与其他基准相比,模型在 Citeseer 数据集上表现最为出色,与 VGAE 相比,AUC 和 AP 分别提高了 3.7 百分点和 3 百分点,而与 ARVGA 相比则提高了 2.1 百分点和 2 百分点。与非编码器图嵌入方法相比,该方法的性能有了显著提高。在 Citeseer 数据集上,AAVGA-d 的 AUC 比 Spectral Clustering 和 DeepWalk 的 AUC 高 14 百分点;AP 分别增长了 10 百分点和 11.4 百分点。实验结果表明,通过结合图编码器中的注意力机制和对抗机制,可以实现图嵌入能力的提升。

表 2 Cora 链路预测实验结果(%)

方法	Cora	
	AUC	AP
SC	84.6 ±0.01	85.5 ±0.00
DW	83.1 ±0.01	85.0 ±0.00
GAE *	84.3 ±0.02	88.1 ±0.01
VGAE *	84.0 ±0.02	87.7 ±0.01
GAE	91.0 ±0.02	92.0 ±0.03
VGAE	91.4 ±0.01	92.6 ±0.01
ARGE	92.4 ±0.003	93.2 ±0.003
ARVGE	92.4 ±0.004	92.6 ±0.004
AAVGA	93.3 ±0.005	94.2 ±0.005
AAVGA-d	**93.8 ±0.005**	**94.6 ±0.005**

表 3　Citeseer 链路预测实验结果(%)

方法	Citeseer	
	AUC	AP
SC	80.5 ±0.01	85.0 ±0.01
DW	80.5 ±0.02	83.6 ±0.01
GAE *	78.7 ±0.02	84.1 ±0.02
VGAE *	78.9 ±0.03	84.1 ±0.02
GAE	89.5 ±0.04	89.9 ±0.05
VGAE	90.8 ±0.02	92.0 ±0.02
ARGE	91.9 ±0.003	93.0 ±0.003
ARVGE	92.4 ±0.003	93.0 ±0.003
AAVGA	94.0 ±0.004	94.6 ±0.004
AAVGA-d	**94.5 ±0.004**	**95.0 ±0.004**

表 4　Pubmed 链路预测实验结果(%)

方法	Pubmed	
	AUC	AP
SC	84.2 ±0.02	87.8 ±0.01
DW	84.4 ±0.00	84.1 ±0.00
GAE *	82.2 ±0.01	87.4 ±0.00
VGAE *	82.7 ±0.01	87.5 ±0.01
GAE	96.4 ±0.00	96.5 ±0.00
VGAE	94.4 ±0.02	94.7 ±0.02
ARGE	96.8 ±0.001	97.1 ±0.001
ARVGE	96.5 ±0.001	96.8 ±0.001
AAVGA	97.2 ±0.002	97.4 ±0.002
AAVGA-d	**97.8 ±0.002**	**97.9 ±0.002**

3.5　随机边删除技术的深入分析

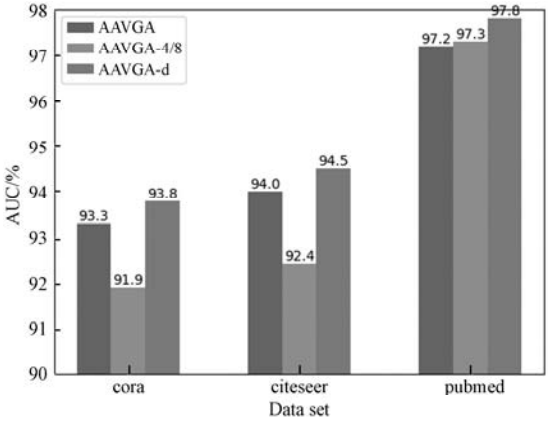
在 3.4 节中,本文已经验证了 AAVGA-d 具有良好的图嵌入性能。本节将进一步讨论 RDEdge 对模型的作用。在图 1 中已经表明了 RDEdge 对深层图注意力模型具有抗过拟合与抗过平滑的作用。具体地,为了说明 RDEdge 技术对模型精度的提升,本文做了进一步的工作:比较 AAVGA、AAVGA-4/8 和 AAVGA-d 三个模型在链路预测实验下的精度。

(1) AAVGA:单层图注意力对抗变分自动编码器。

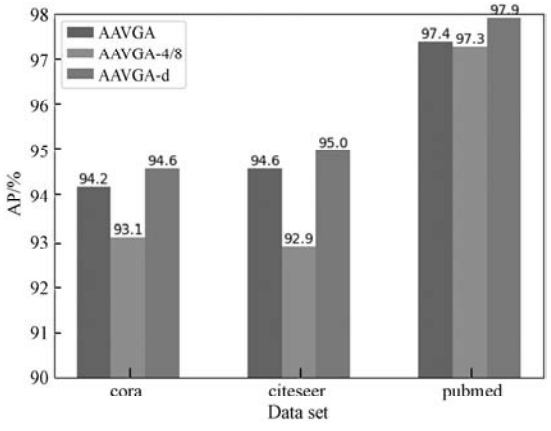
(2) AAVGA-4/8:AAVGA 的多注意力层版本。

(3) AAVGA-d:在加深图注意力层的同时,运用了随机边删除技术。

实验各超参数设置与 3.3 节相同,其中,AAVGA 在三个数据集上都只使用单层图注意力层。AAVGA-4/8 和 AAVGA-d 在 Cora、Citeseer 数据集上将图注意力层加深到 4 层,Pubmed 为 8 层。实验结果如图 3 所示。注意到在 Cora 和 Citeseer 数据集上单纯加深编码器的图注意层会导致 AUC 和 AP 的精度下降,这与本文之前的设想一致,因为深层图注意力会导致过拟合与过平滑问题。其次,在 Pubmed 数据集上并未看到明显的实验精度下降,本文认为原因在于 Pubmed 数据集本身数据量足够大,而且本文的图注意力模型在聚合的时候限制为节点的一阶邻居,所以没有出现明显的过拟合与过平滑问题。值得注意的是,结合了 RDEdge 技术的 AAVGA-d 在三个数据上都有着出色的表现。这表明 RDEdge 技术对图注意力模型确实存在抗过拟合与抗过平滑的能力;另一方面,这也表明了适当地加深图注意力层数可以提升模型的图嵌入能力。



(a) AUC



(b) AP

图 3　3 个模型的链路预测实验结果

上述实验中,本文将随机边删除技术应用于 AAVGA 模型时,所有注意力层共享同一个 $\boldsymbol{A}_{\mathrm{rd}}$。值得注意的是,模型可以为每个注意力层单独执行 RDEdge,具体来说,通过式(19)来独立计算每层的 $\boldsymbol{A}_{\mathrm{rd}}^{(l)}$,这样可以使注意力层获得独有 $\boldsymbol{A}_{\mathrm{rd}}^{(l)}$,进而形成注意力的多样化表达,获得更多的随机性。本文从损失函数的角度对逐层的随机边删除技术(RDEdge-L)在 Cora 数据集上进行实验评估,如图 4 所示,结合了逐层随机边删除技术的 AAVGA-dl 比 AAVGA-d 拥有更小的训练损失,但是两者验证集上损失曲线接近于重合,表现相差不大,这表明逐层的随机边删除技术更有利于训练的进行。

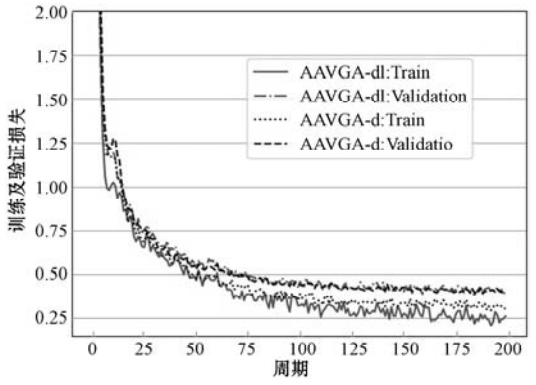
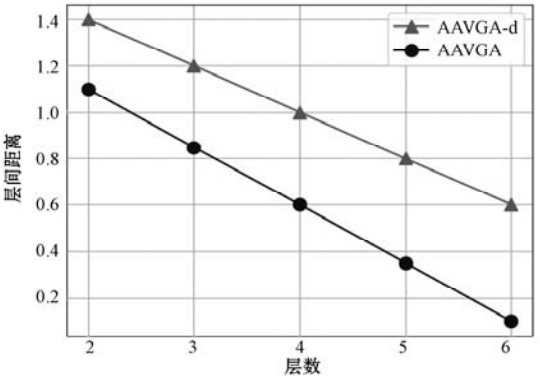
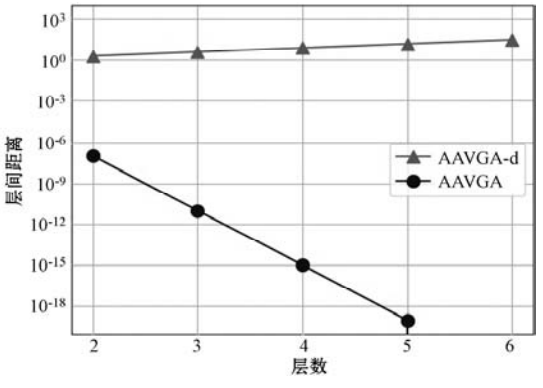


图 4 逐层 RDEdge 与 RDEdge 的损失对比

为了进一步验证 AAVGA-d 具有缓解过平滑的能力,本文通过计算当前注意力层的输出和下一层输出之间的差异来量化过平滑的程度,用欧氏距离来进行差异计算,距离越小意味着过平滑程度越严重。实验在 Cora 数据集上进行,AAVGA 和 AAVGA-d 都使用了 6 层图注意力。如图 5(a)所示,在训练前,随着层数的增长,过平滑现象变得严重。但是,AAVGA-d 层与层之间的距离相对较大,且下降速率较慢。如图 5(b)所示,在经过 400 次训练后,没有使用 RDEdge 技术的 AAVGA,第 5 层和第 6 层间的差异几乎为零,这表明隐藏特征已经聚合收敛到某一固定点。相反,AAVGA-d 层间的距离没有缩小,呈现缓慢上升的趋势,更为直观地证明了 AAVGA-d 能够很好地应对由层数加深带来的过平滑问题。



(a) 训练前



(b) 训练后

图 5 注意力层输出间的距离

4 结 语

本文将图注意力和对抗机制引入图自动编码器中,通过在嵌入过程中实现对邻居节点权重的自适应分配和对表示向量分布的正则化约束,实现图注意力对抗变分自动编码器对图数据的良好嵌入。同时设计一种针对图注意力模型的随机边删除技术(RDEdge),它有助于加深图注意力对抗变分自动编码器的注意力层数,并且很好地应对了由此带来的过拟合与过平滑问题。RDEdge 通过随机丢弃图中一定比例的边,使输入数据产生随机变形,增加数据的多样性来应对过拟合。在注意力聚合过程中减少消息传递以减轻过平滑。结合 RDEdge 技术后,本文开发深层图注意力对抗变分自动编码器(AAVGA-d),这是一种考虑节点间重要性差异的深层图注意力编码器,并联合对抗机制,保证编码器嵌入得到的图表示向量分布与先验分布的一致性。注意力层数的加深使编码器的图嵌入能力得到进一步的提升。

本文的图注意力编码器在计算邻居节点的注意力权重时,只限定节点的一阶邻居参与计算,这大大降低计算的复杂度,也在一定程度上避免更为严重的过平滑问题。但是,当把参与计算的邻居扩大到二阶、三阶乃至更多阶时,RDEdge 技术是否还能高效地应对过平滑问题,值得进一步探索与研究。

参 考 文 献

[1] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[EB]. arXiv: 1609.02907, 2016.

[2] Grover A, Leskovec J. Node2vec: Scalable feature learning for networks[C]//22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 855-864.

[3] Ying R, You J X, Morris C, et al. Hierarchical graph representation learning with differentiable pooling[EB]. arXiv: 1806.08804, 2018.

[4] You J X, Ying R, Ren X, et al. GraphRNN: Generating realistic graphs with deep auto-regressive models[C]//International Conference on Machine Learning, 2018: 5708-5717.

[5] Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs[EB]. arXiv: 1706.02216, 2017.

[6] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[EB]. arXiv:1710.10903,2017.

[7] Kipf T N, Welling M. Variational graph auto-encoders [EB]. arXiv:1611.07308,2016.

[8] Zhu D Y, Cui P, Wang D X, et al. Deep variational network embedding in Wasserstein space[C]//24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining,2018:2827 - 2836.

[9] Pan S R, Hu R Q, Long G D, et al. Adversarially regularized graph autoencoder for graph embedding[EB]. arXiv: 1802.04407,2018.

[10] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[EB]. arXiv:1406.2661,2014.

[11] Li G, Muller M, Thabet A, et al. DeepGCNs: Can GCNs go as deep as CNNs? [C]//IEEE International Conference on Computer Vision,2019:9267 - 9276.

[12] Xu K, Li C T, Tian Y L, et al. Representation learning on graphs with jumping knowledge networks[C]//35th International Conference on Machine Learning,2018:5453 - 5462.

[13] Li Q M, Han Z C, Wu X M. Deeper insights into graph convolutional networks for semi-supervised learning[C]// 32nd AAAI Conference on Artificial Intelligence,2018:3538 - 3545.

[14] Xu K L, Li C T, Tian Y L, et al. Representation learning on graphs with jumping knowledge networks[C]//International Conference on Machine Learning,2018:5453 - 5462.

[15] Shuman D I, Narang S K, Frossard P, et al. The emerging filed of signal processing on graphs[J]. IEEE Signal Processing Magazine,2013,30(3):83 - 98.

[16] Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs[EB]. arXiv: 1312.6203,2013.

[17] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering [EB]. arXiv:1606.09375,2016.

[18] Xu B, Shen H W, Cao Q, et al. Graph wavelet neural network[EB]. arXiv:1904.07785,2019.

[19] 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.

[20] 管珊珊,张益农. 基于残差时空图卷积网络的 3D 人体行为识别[J]. 计算机应用与软件,2020,37(3):198 - 201, 250.

[21] Hu W H, Liu B W, Gomes J, et al. Strategies for pre-training graph neural networks[EB]. arXiv:1905.12265,2019.

[22] Velickovic P, Fedus W, Hamilton W L, et al. Deep graph infomax[EB]. arXiv:1809.10341,2018.

[23] Kingma D P, Welling M. Auto-encoding variational bayes [EB]. arXiv:1312.6114,2013.

[24] Wang H W, Wang J, Wang J L, et al. GraphGAN: Graph representation learning with generative adversarial nets [C]//32nd AAAI Conference on Artificial Intelligence, 2018:531 - 542.

[25] Dai Q Y, Li Q, Tang J, et al. Adversarial network embedding[C]//Proceedings of the AAAI Conference on Artificial Intelligence,2018:865 - 871.

[26] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[C]//20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014:701 - 710.

[27] Klicpera J, Bojchevski A, Günnemann S. Predict then propagate: Graph neural networks meet personalized pagerank [EB]. arXiv:1810.05997,2018.

[28] Oono K, Suzuki T. On asymptotic behaviors of graph CNNs from dynamical systems perspective[EB]. arXiv: 1905.10947,2019.

[29] Rong Y, Huang W B, Xu T, et al. Dropedge: Towards deep graph convolutional networks on node classification[EB]. arXiv:1907.10903,2019.

[30] Tang L, Liu H. Leveraging social media networks for classification[J]. Data Mining and Knowledge Discovery,2011, 23(3):447 - 478.

(上接第 120 页)

[9] 郭建军,鲍雨亭,荆芒. 基于医联体的多路径远程会诊平台建设[J]. 医学信息学杂志,2018,39(1):22 - 25.

[10] Dorigo M. Optimization, learning and natural algorithms[D]. Milan: Politecnico di Milano,1992.

[11] 陈祖林,金忠林,刘建华. 基于“信息交换平台”建立的远程会诊支持系统的研发应用[J]. 中国数字医学,2014 (3):88 - 89.

[12] Bai M, Wang X T, Xin J C, et al. An efficient algorithm for distributed density-based outlier detection on big data[J]. Neurocomputing,2016,181:19 - 28.

[13] Atzei N, Bartoletti M, Cimoli T. A survey of attacks on Ethereum smart contracts(SoK)[M]//Principles of Security and Trust. Springer,2017:164 - 186.

[14] 王雷,李明,刘志虎,等. 基于吸引场的蚁群算法在 TSP 中的应用[J]. 江苏大学学报(自然科学版),2015,36(5): 573 - 577.

[15] 周自廉,杨进,马良. 基于混合的细菌觅食算法求解 TSP 问题[J]. 数学的实践与认识,2015,45(16):159 - 165.