

Local Augmentation for Graph Neural Networks

Songtao Liu Rex Ying Hanze Dong Lanqing Li Tingyang Xu Yu Rong Peilin Zhao Junzhou Huang Dinghao Wu

摘要

图形神经网络 (Graph Neural Networks, GNN) 在基于图形的任务上取得了显著的性能。GNN 的关键思想是通过聚集来自局部邻域的信息来获得信息表示。然而, 邻域信息是否被充分地聚合以用于具有很少邻居的节点的学习表示仍然是一个悬而未决的问题。为了解决这一问题, 我们提出了一种简单而有效的数据增强策略, 即局部扩增, 以学习基于中心节点特征的相邻节点特征的分布, 并利用生成的特征增强 GNN 的表达能力。局部扩增是一种通用框架, 可以以即插即用的方式应用于任何 GNN 模型。它从学习到的条件分布中采样与每个节点相关的特征向量, 作为每个训练迭代中主干模型的附加输入。广泛的实验和分析表明, 当应用于各种 GNN 体系结构时, 局部扩增在不同的基准集上一致地提高了性能。

关键词: 图神经网络; 局部扩增

1 引言

近年来, 由于图结构的强大表现力, 用机器学习方法分析图的研究越来越受到重视。图神经网络 (Graph Neural Networks, GNN) 是一类基于深度学习的处理图域信息的方法。由于其较好的性能和可解释性, 图神经网络最近已成为一种广泛应用的图分析方法。GNN 在基于图的任务上取得了显著的性能, GNN 的关键思想是通过聚集来自当地社区的信息来获得信息表示。尽管 GNN 在学习节点表示方面取得了进展, 但是从局部信息来看, 这仍然是一个悬而未决的问题。对于那些邻居数量有限的节点, 局部邻域信息是否足以获得有效的节点表示, 局部邻居的数量限制了 GNN 的表现力并阻碍其性能。堆叠图形层^[1]以放大感受野可以包含多跳相邻信息, 但会导致过度平滑, 不是解决此问题的直接解决方案。GNN 模型的现有工作架构无法解决结点邻居非常有限的问题因此, 本文重点关注丰富局部信息, 以获得有效的表示。为了解决这个问题, 在本文中提出了一种简单有效的数据增强策略, 即局部扩增, 以根据中心节点的特征来学习邻居节点特征的分布, 并通过生成的特征来增强 GNN 的表达能力。

2 相关工作

2.1 图神经网络

GNN 及其变体在各种基于图形的任务, 包括推荐系统^[2]、药物发现^[3]和交通预测^[4]。GNN 的核心是消息传递机制, 以生成信息表示。深层 GNN 的最新发展, 如 JKnet^[5]、GCNII^[6]和 RevGNN-Deep^[7]将浅层结构增加到具有残余风格设计的深层。此外, 最近的研究^[8]利用图结构, 以设计高效的局部邻域的信息的消息传递聚合方案, 以增强 GNN 的释放能力。这些工作表明, 局部邻域信息在训练 GNN 中发挥着重要作用。

2.2 数据扩增

数据扩增在计算机视觉和自然语言处理中得到了很好的研究, 然而关于图结构数据的研究仍然不足。现有图数据增长方法仅在从全局角度看拓扑级别和要素级别, 可分为两类: 拓扑级增加^[9]和特征水平增加^[10]。拓扑级别增加了伴随矩阵的扰动, 产生了不同图结构。另一方面, 现有特征水平上升在节点属性上生成扰动, 通过对手的训练来提高整体性。这些增加技术有显著的回报, 然而它们关注全局增长, 图的整个分布, 而不是单个节点, 忽略了邻居的局部信息。

3 本文方法

3.1 本文方法概述

为了促进聚合方案, 在局部邻域中具有更多生成的特征, 本文提出了一种新颖有效的数据增强框架: 图神经网络的局部扩增 (Local Augmentation for Graph Neural Networks, LAGNN)。“局部扩增”是指通过基于局部结构和节点特征的生成模型生成邻域特征。本文提出的框架包括预训练步骤, 它学习连接的条件分布生成模型。给定一个中心节点的特征的邻居节点特征, 然后利用该分布生成该中心节点的附加输入。此外, 我们将生成的预训练模型与下游 GNN 训练解耦, 以允许我们的数据以即插即用的方式运用该模型。首先利用条件变分自动编码器 (Conditional Variational Auto Encoder, CVAE)^[11]以学习连接邻居的节点特征的条件分布, 只为所有节点训练一个 CVAE。在生成阶段, 将所有节点作为特征, 每个节点的相应的邻居结点特征为条件, 输入到 CVAE 的编码器对潜在变量进行采样。然后将潜在变量输入到 CVAE 的解码器中, 生成与所有结点的扩增特征向量。接着将原始数据与扩增数据输入到 GNN 中共同参与训练。图 1 展示了局部扩增的基本过程:

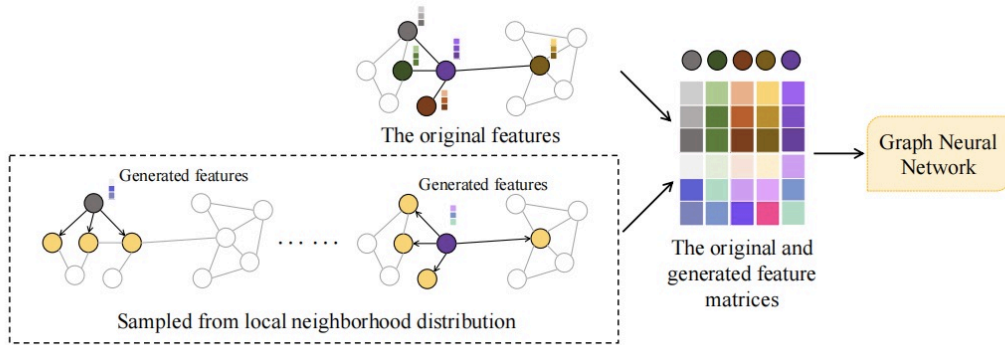


图 1: 局部扩增示意图。图上的黄色圆圈对应于相邻节点。假设我们了解了局部邻域的分布情况, 从局部邻域分布生成特征, 然后采取原始特征和生成的特征作为下游 GNN 的输入。

3.2 CVAE 模块

基于 MLP 构建 CVAE。编码和解码器是两层 MLP, 其中每个层都有 256 个隐藏单元, 图形数据集。对于节点 v 及其邻居 N_v , 在训练阶段提取邻居对 (X_v, X_u) 作为 CVAE 的输入, 其中 u 为节点 v 的邻居结点。在推理阶段, 从高斯分布 $N(0, 1)$ 和中心节点 v 中提取潜在变量 z , 特征向量 X_v 作为 CVAE 解码器的输入。因此, 可以获得生成的扩增特征向量 \overline{X}_v 。CVAE 的详细信息如图 2 所示:

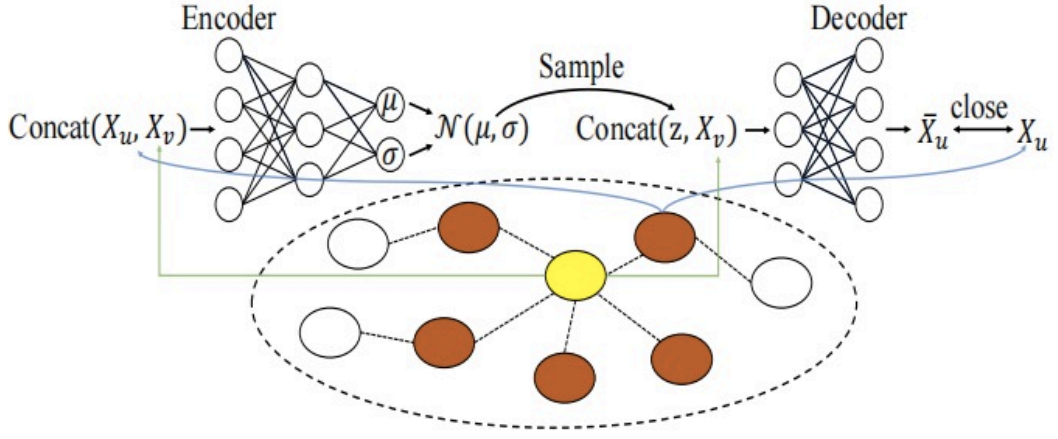


图 2: CVAE 的示意图。图上的黄色和棕色圆圈对应于中心节点及其邻居, 分别提取它们的特征向量作为 CVAE 的输入。

3.3 LAGNN 的传播公式

这一小节将介绍 LAGNN 的传播公式, 使用我们的本地增强模型生成的特征作为附加输入。为了说明局部扩增模型, 提供了两种不同的方法进行架构的串联设计。对于 LAGCN, 仅在 GCN 的图卷积的第一层做一点微小改变, 具体公式如下所示, 其中 $P = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, \parallel 表示在第二个维度上进行级联。权重矩阵 W 中的下标和上标表示层和参数的序号。为了不更改 GCN 模型的参数大小, $W_0^{(1)}$ 和 $W_1^{(1)}$ 的第二维度的和等于 GCN 的 $W^{(1)}$ 的第二维度。

$$H^{(1)} = \sigma(PXW_0^{(1)}) \parallel \sigma(PXW_1^{(1)}) \quad (1)$$

相似地, LAGAT 的第一层定义如下所示, 其中 α_{vu}^d 计算在 X ($1 \leq d \leq D/2$) 或者 \bar{X} ($D/2 + 1 \leq d \leq D$)。 $W_d^{(1)}$ 的第二维度与 GAT 相同。 \bar{X} 仅代替输入一般的注意力头。

$$H^{(1)} = (\parallel_{d=1}^{D/2} \sigma(\sum_{u \in N_v} \alpha_{vu}^d W_d^{(1)} X_u)) \parallel (\parallel_{d=D/2+1}^D \sigma(\sum_{u \in N_v} \alpha_{vu}^d W_d^{(1)} \bar{X}_u)) \quad (2)$$

$$(3)$$

3.4 损失函数定义

使用原始特征矩阵 X 和生成的特征矩阵 \bar{X} 作为训练的输入去训练 LAGNN。给定增强的训练标签 L_T, K 和特征矩阵 $\bar{X}^{(k)}, \bar{Z}^{(k)} = f_{LAGNN}(A, X, \bar{X}^{(k)}, \Theta)$, 节点分类任务的监督损失如下:

$$L_s = -\frac{1}{K} \sum_{k=1}^K \sum_{i \in L_T} Y_i \log Z_i^{(k)} \quad (4)$$

给定 $\bar{Z}_i = \frac{1}{K} \sum_{k=1}^K Z_i^{(k)}, \bar{Z}_i' = \bar{Z}_i T / \sum_{c=1}^C Z_{ic} T$, X 为调整类分布的温度参数, 一致性正则化损失函数为以下形式:

$$L_c = \frac{1}{K} \sum_{k=1}^K \sum_i^N \|\bar{Z}_i' - Z_i^{(k)}\|_2^2 \quad (5)$$

总体损失函数如下:

$$L = L_c + \beta L_s \quad (6)$$

4 复现细节

4.1 与已有开源代码对比

复现过程中参考了 <https://github.com/SongtaoLiu0823/LAGNN> 中获得 CAVE 生成器的代码，具体如下伪代码所示。

Procedure 1 The framework to obtain the Generator $Q\Phi$ with active learning trick on Cora, Citeseer, and Pubmed

Input: Adjacency matrix A , feature matrix X

Output: Generator $Q\Phi$

Initialize $U=-\text{inf}$, $Q\Phi$, and $Q0\Phi$

```
for  $l$  in the number of pre-training epochs do
    Update the parameters of generator  $Q\Phi$ 
    Generate feature matrix  $X$  using  $Q\Phi$ 
    Compute  $U(X)$ 
    if  $U(X) > U$  then
         $U = U(X)$   $Q0\Phi = Q\Phi$ 
    end
```

end

$Q\Phi = Q0\Phi$

return: Generator $Q\Phi$

4.2 实验环境搭建

本文实验运行在 python3.9.0, 运行的的 python 依赖包为 pytorch==1.12.1, torch-geometric==1.12.1, networkx=2.4.1。由于实验运行在 A100 显卡上，下载的 pytorch 及 torch-geometric 为 GPU 版本。

4.3 使用说明

data 文件夹存放的是运行的数据三个数据集，运行 gat_la.py 将得到 LAGAT 结果，运行 gcn_la.py 将得到 LAGCN 结果，运行 gcnii_la.py 将得到 LAGCNII 结果，运行 grand_la.py 将得到 LAGRAND 结果，运行 sage_la.py 将得到 LASAGE 结果。训练得到的结果将会保存在 log 文件夹下。

4.4 创新点

论文原文将局部扩增模型与 GNN 流行的 GCN、GAT、GRAND、GCNII 模型进行了结合，并在 Cora、Citeseer、Pubmed 数据集上进行了实验分析。而 GraphSAGE 是 GNN 模型中一个经典的模型，故局部扩增与 GraphSAGE 模型结合能否带来 GraphSAGE 模型的性能提升是一个值得研究的问题。于是在复现的过程中，不仅实现了论文原文中将 GCN、GAT、GRAND、GCNII 模型与局部扩增模型进行结合进行训练，而且也实现了论文原文未实现的将 GraphSAGE 模型与局部扩增模型进行结合，并对实现的模型在数据集上进行了实验分析。

5 实验结果分析

5.1 实验设置

所有的实验在图数据集上进行,利用三个公共用于半监督结点分类的引文网络数据集 Cora、Citseer 和 Pubmed^[12]进行测试评估。采用标准固定,对 Cora、Citseer 和 Pubmed 的分裂,每个类有 20 个节点用于训练、500 个节点用于验证以及 1000 个节点用于测试。训练时，在各个数据集下 lr 均设置为 0.01, weight-decay 设置为 5e-4，实验其他参数的详细设置如表 1 所示。

表 1: 在三个引文数据集训练的超参数设置

Dataset	Backbone	Hyper-parameters
Cora	LAGCN	epochs: 2000, dropout: 0.5, β : 1.0 T: 0.5
	LAGAT	epochs: 1000, heads: [4, 1], dropout: 0.6, alpha: 0.2, β : 1.0 T: 0.5
	LAGCNII	α : 0.1, λ : 0.5, dropout: 0.6
	LAGRAND	dropout: 0.5, β : 1.0, T: 0.5
	LASAGE	epochs: 2000, dropout: 0.5, β : 1.0 T: 0.5
Citeseer	LAGCN	epochs: 2000, dropout: 0.5, β : 1.0 T: 0.5
	LAGAT	epochs: 1000, heads: [4, 1], dropout: 0.6, alpha: 0.2, β : 1.0 T: 0.5
	LAGCNII	α : 0.1, λ : 0.6, dropout: 0.7, β : 1.0, T: 0.5
	LAGRAND	dropout: [0.2, 0.1], propagation step: 2, β : 0.7, T: 0.2
	LASAGE	epochs: 2000, dropout: 0.5, β : 1.0 T: 0.5
Pubmed	LAGCN	epochs: 2000, dropout: 0.5, β : 1.0 T: 0.5
	LAGAT	epochs: 1000, heads: [2, 1], dropout: 0.6, alpha: 0.2, β : 1.0 T: 0.5
	LAGCNII	α : 0.1, λ : 0.4, dropout: 0.5, β : 1.0, T: 0.5
	LAGRAND	dropout: [0.7, 0.8], propagation step: 5, β : 1.2, T: 0.2
	LASAGE	epochs: 2000, dropout: 0.5, β : 1.0 T: 0.5

5.2 实验结果

考虑了四种流行的图神经网络:GCN^[13]、GAT^[14]、GraphSAGE^[15]和 GCNII^[6]作为实施骨干。GCN、GAT 和 GraphSAGE 和 GCNII 是最先进的 GNN 架构, 而 GCNII 是具有跳跃连接设计的深度 GNN 模型。同时还将数据增强模型 GRAND 与提出的局部扩增方法结合, 进行实验对比。详细的实验结果见表 2所示. 所示实验结果是在 10 次运行后的平均节点分类精度, 其余基线为相应论文中已报告的基线指标。结果表明, 配备了局部扩增模型在所有三个数据集上实现了更好的性能。复现结果与原文结果之间的比对见表 3所示, 原文结果用”表示, 对比原文与复现结果, 复现结果与原文结果仅有细微的差别。

6 总结与展望

本文复现的局部扩增模型, 利用生成模型来学习中心节点邻居特征的条件分布及中心节点的特征。将生成的特征矩阵从经过预训练的生成模型馈送到主干 GNN 模型, 以提高其性能, 实验结果也表明局部扩增的方法可以提高性能。目前的实现过程中, 仅在小型数据集 Cora、Cite seer、Pubmed 上进行实验, 并未在大型数据集上进行实验, 大型数据集更能体现出方法的性能优劣, 故在未来进行在大型数据集上的研究。并且对于中心结点度较小的图及大型图的研究还未涉及, 未来将进一步研究这方面的问题。

参考文献

- [1] LI Q, HAN Z, WU X M. Deeper insights into graph convolutional networks for semi-supervised learning [C]//Thirty-Second AAAI conference on artificial intelligence. 2018.
- [2] 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. 2018: 974-983.

- [3] DAI H, LI C, COLEY C, et al. Retrosynthesis prediction with conditional graph logic network[J]. Advances in Neural Information Processing Systems, 2019, 32.
- [4] GUO S, LIN Y, FENG N, et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 33: 01. 2019: 922-929.
- [5] XU K, LI C, TIAN Y, et al. Representation learning on graphs with jumping knowledge networks[C]//International conference on machine learning. 2018: 5453-5462.
- [6] CHEN M, WEI Z, HUANG Z, et al. Simple and deep graph convolutional networks[C]//International Conference on Machine Learning. 2020: 1725-1735.
- [7] LI G, MÜLLER M, GHANEM B, et al. Training graph neural networks with 1000 layers[C]//International conference on machine learning. 2021: 6437-6449.
- [8] WIJESINGHE A, WANG Q. A New Perspective on” How Graph Neural Networks Go Beyond Weisfeiler-Lehman?”[C]//International Conference on Learning Representations. 2021.
- [9] FANG C, DONG H, ZHANG T. Mathematical models of overparameterized neural networks[J]. Proceedings of the IEEE, 2021, 109(5): 683-703.
- [10] KONG K, LI G, DING M, et al. Flag: Adversarial data augmentation for graph neural networks[J]. arXiv preprint arXiv:2010.09891, 2020.
- [11] SOHN K, LEE H, YAN X. Learning structured output representation using deep conditional generative models[J]. Advances in neural information processing systems, 2015, 28.
- [12] SEN P, NAMATA G, BILGIC M, et al. Collective classification in network data[J]. AI magazine, 2008, 29(3): 93-93.
- [13] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
- [14] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.
- [15] HAMILTON W, YING Z, LESKOVEC J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems, 2017, 30.

表 2: 在三个引文数据集下的分类复现结果 (%)

Method	Cora	Citeseer	Pumbed
Chebyshev	81.2	69.8	74.4
APNP	83.8±0.3	71.6±0.5	79.7±0.3
MixHop	81.9±0.4	71.4±0.8	80.8±0.6
Graph U-net	84.4±0.6	73.2±0.5	79.6±0.2
GSNN-M	83.9±0.5	72.2±0.5	79.1±0.3
S2GC	83.5±0.02	73.6±0.09	80.2±0.02
GCN	81.5±0.5	70.3±0.7	79.0±0.5
G-GCN	83.7	71.3	80.9
DropEdge-GCN	82.8	72.3	79.6
GAUG-O-GCN	83.6±0.5	73.3±1.1	79.3±0.4
GRAND-GCN	83.1±1.8	72.3±1.5	79.8±1.2
LA-GCN	84.4±0.4	74.5±0.3	81.7±0.7
GAT	83.0±0.7	72.5±0.7	79.0±0.3
GAUG-O-GAT	82.2±0.2	71.6±1.1	OOM
GraphSNN-GAT	83.8±1.2	73.5±1.6	79.6±1.4
GRAND-GAT	84.3±0.4	73.2±0.4	79.2±0.6
LA-GAT	84.0±0.3	74.4±0.4	81.5±0.8
GCNII	85.5±0.5	73.4±0.6	80.2±0.4
LA-GCNII	85.6±0.4	73.7±0.4	81.8±0.7
GRAND	85.4±0.4	75.4±0.4	82.7±0.6
LA-GRAND	85.6±0.2	75.5±0.6	83.9±0.2
GraphSAGE	80.3±0.2	69.8±0.9	78.9±0.5
LA-GraphSAGE	82.6±0.5	74.4±0.7	81.4±0.8

表 3: 在三个引文数据集下原文与复现结果对比 (%)

Method	Cora	Citeseer	Pumbed
LA-GCN	84.4±0.4	74.5±0.3	81.7±0.7
'LA-GCN'	84.6±0.5	74.7±0.5	81.7±0.7
LA-GAT	84.0±0.3	74.4±0.4	81.5±0.8
'LA-GAT'	84.7±0.4	73.7±0.5	81.0±0.4
LA-GCNII	85.6±0.4	73.7±0.4	81.8±0.7
'LA-GCNII'	85.7±0.3	74.1±0.5	80.6±0.7
LA-GRAND	85.6±0.2	75.5±0.6	83.9±0.2
'LA-GRAND'	85.7±0.3	75.8±0.5	83.4±0.6
LA-GraphSAGE	82.6±0.5	74.4±0.7	81.4±0.8