

模式识别大作业报告

甘云冲

2101213081

信息科学技术学院

摘 要

在计算机视觉任务当中，卷积神经网络被应用到图片上并取得了巨大的成功，但是在应用到图数据上却遇到了困难。而类似 U-Net 的编码器-解码器架构在图像像素级预测，如图像分割任务上已经取得了很好的效果。在图数据上没有类似的方法，其原因在于图数据上没有好的池化和反池化方法。在本次作业当中，针对 Graph U-Nets 进行复现，主要针对于节点分类任务，在 Cora, CiteSeer, PubMed 三个引文数据集上测试节点分类的效果。

关键词：深度学习，图神经网络

1 引言

卷积神经网络（CNNs）在图像数据集当中已经取得了巨大的成功，而事实上，在卷积神经网络当中嵌入了先验性息，图像数据能够应用卷积神经网络的一大特性在于其具有局部特性。

而在真实世界的的数据当中，很多数据，如社交网络和蛋白质结构，天然就容易被表示为图结构。考虑到卷积神经网络在图像数据上取得的巨大成功，已经有非常多把卷积神经网络拓展到图结构数据上的尝试。

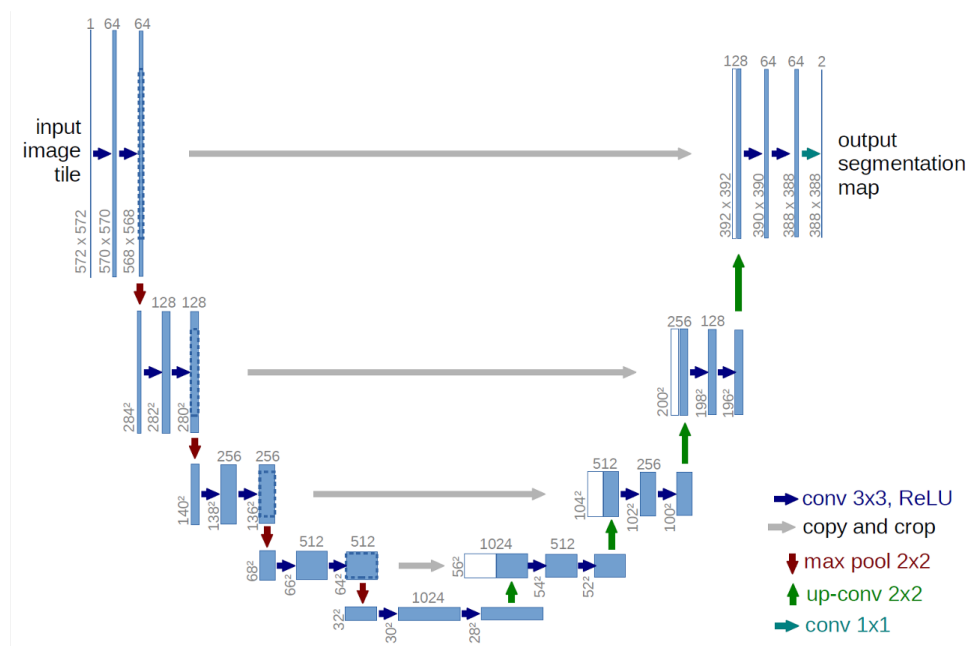


图 1: U-Net

事实上图片可以被看成一种特殊的图结构，即所有的节点都落在一个二维的网格状图上。所以对于图结构的节点分类任务实际上可以对应于图片类型数据当中像素级别的分类任务。即图像分割任务，而图像分割任务通过 U-Nets 等结构实际上已经取得了巨大的进展 [1]。U-Nets 是一种编码器-解码器（Encoder-Decoder）的结构。在这种结构当中，除去卷积神经网络之外，同时还有池化和反池化操作，来使得聚合不同层级的信息成为可能。但是在图结构的数据上去实现相关操作事实上并没那么自然，图数据并不像图片数据一样具有空间上的局部性和顺序的信息。

为了解决这个问题，Graph U-Nets 提出了图池化操作（graph pooling, gPool）和图反池化操作（graph unpooling, gUnpool）基于这两个操作，实现了一种图结构数据上的 U-Net 架构，被称为 Graph U-Net。其中图池化操作通过从整个图当中采样出一些点来构造一个小的子图，从而使得抽取高层级的特征信息成为可能。具体的内容会在之后的方法部分进行介绍。

2 方法

2.1 图池化层

池化层在卷积神经网络当中有着非常重要的地位。他们可以减小特征图的大小，同时扩大感受野的范围，因此能够得到更好的表现以及泛化效果。在如图像这种网格状的数据当中，特征图被划分成一些不重叠的矩形，之后通过下采样函数来减小每一块特征图的大小。不只是局部池化，同时还有如全局池化这种方法，会将整个特征图下采样为一个向量。

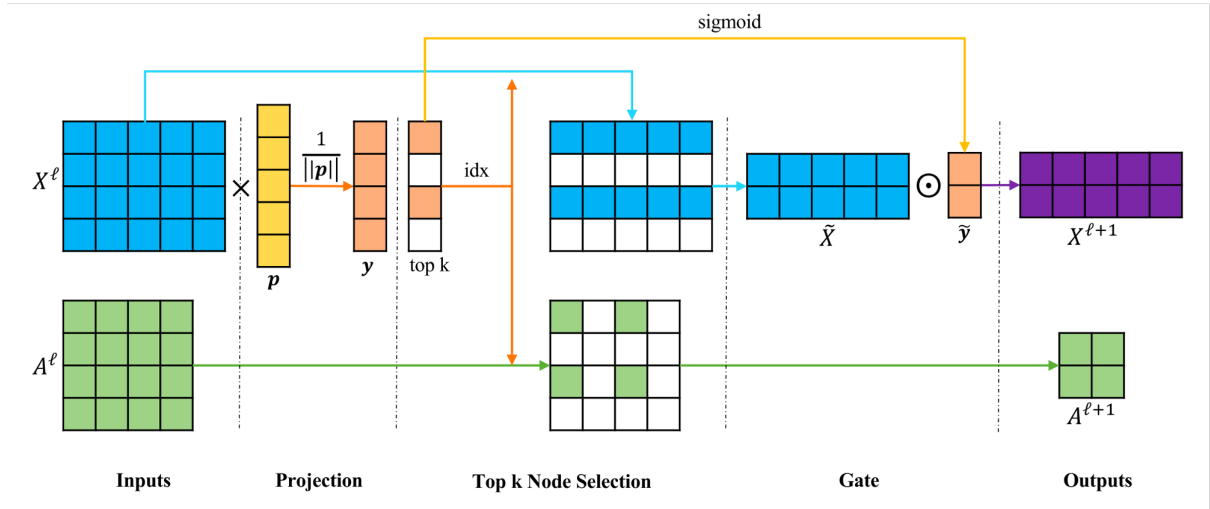


图 2: gPool

但是由于图数据并不像图像数据那么规整，没有充分的局部信息，所以不能够直接在图结构数据上应用池化操作。但是直接应用全局池化操作，将整个图聚合成一个节点会很大程度上限制网络的表示性能。 k -max 池化将图中最大的 k 个单元选择出来，从而能够在两者之中达成一个平衡。

在文中提出的图池化层就是一种在图数据上进行下采样的方法，通过自适应的从图中采样节点从而构造一个更小的图，起到如卷积神经网络当中池化层的作用。通过将每个节点的特征表示为一维的向量，可以在这撒谎给你卖弄应用 k -max 池化。对于一个节点 x_i ，通过 p 将其映

射到 $y_i = \mathbf{x}_i \mathbf{p} / \|\mathbf{p}\|$ 这样一个标量，来衡量节点 i 当中所包含的信息量。希望在池化之后所构造的子图当中，能够尽可能多的保存原始图当中的信息。所以为了实现这个要求，我们针对于映射 \mathbf{p} 保存最大的 k 个标量来构造子图。

假设在图 \mathbb{G} 当中有 N 个节点，每个节点有 C 个特征，那么图可以被表示为两个矩阵，一个是邻接矩阵 $A^l \in \mathbb{R}^{N \times N}$ ，一个是特征矩阵 $X^l \in \mathbb{R}^{N \times C}$ 。特征矩阵当中的每个非零值都代表图上的连边。所以整个采样过程可以如下方公式所示。

$$\mathbf{y} = X^l \mathbf{p}^l / \|\mathbf{p}^l\| \quad (1)$$

$$\text{idx} = \text{rank}(\mathbf{y}, k) \quad (2)$$

$$\tilde{\mathbf{y}} = \text{sigmoid}(\mathbf{y}(\text{idx})) \quad (3)$$

$$\tilde{X}^l = X^l(\text{idx}, :) \quad (4)$$

$$A^{l+1} = A^{\text{idx}, \text{idx}} \quad (5)$$

$$X^{l+1} = \tilde{X}^l \odot (\tilde{\mathbf{y}} \mathbf{1}_C^T) \quad (6)$$

其中 k 代表新图的点数。 $\text{rank}(\mathbf{y}, k)$ 是对于节点的排序操作，它返回 \mathbf{y} 中最大的 k 个值所对应的索引。

2.2 图反池化层

上采样操作对编码器-解码器网络（如 U-Net）很重要。网络的编码器通常采用池化操作来减少特征图的大小并增加感受野。而在解码器中，特征图需要被上采样以恢复其原始分辨率。在图像这样的网格状数据上，已经有人提出了一些上采样操作。然而，大部分的操作目前在图形数据上还无法使用。

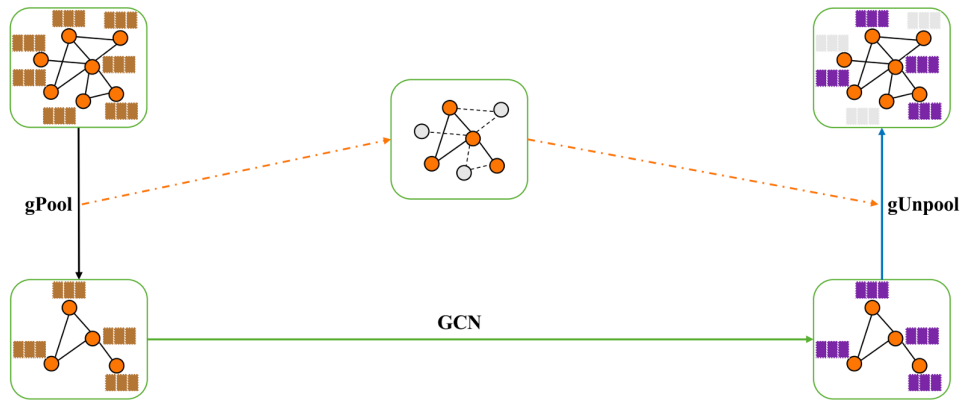


图 3: gUnpool

为了实现对图数据的上采样操作，这里提出了图反池化（gUnpool）层，它执行图池化层的逆向操作，将图恢复到原始结构。为了实现这一目标，我们记录了在相应的图池化层中选择的节点的位置，并使用这些信息将节点放回它们在图中的原始位置。可以将图反池化层的操作形式化定义如下：

$$X^{l+1} = \text{distribute}(0_{N \times C}, X^l, \text{idx}) \quad (7)$$

其中 $\text{idx} \in \mathbb{Z}^k$ 包含相应的图池化层中所选节点的索引，这里假设图反池化层对应的图池化层将图的大小从 N 个节点减少到 k 个节点。 $X^l \in \mathbb{R}^{k \times C}$ 是当前图的特征矩阵， $0_{N \times C}$ 是新图的初始空特征矩阵。 $\text{distribute}(0_{N \times C}, X, \text{idx})$ 是将 X 中的行向量根据其存储在 idx 中的相应索引分配到， $0_{N \times C}$ 特征矩阵的操作。在 X^{l+1} 中，索引在 idx 中的行向量被 X^l 中的行向量更新，而其他行向量保持为零。

2.3 Graph U-Nets

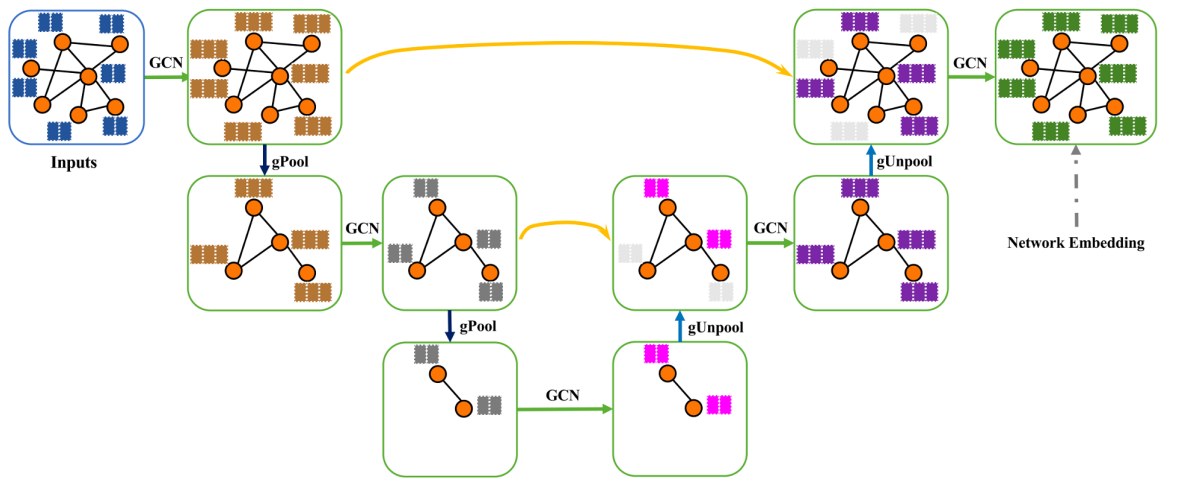


图 4: Graph U-Net

由于像 U-Net 这样的编码器-解码器网络可以在保持低层级局部位置信息的同时进行高层级特征的抽取，所以可以在如图像分割这样的像素级别预测任务上取得非常不错的效果。图像中的像素级预测任务实际上就是对于每一个输入单元都提供一个预测结果，在图结构数据上，和节点分类的任务是相似的。由于在前文已经在图结构数据上完成了图池化层和图反池化层的构造，所以这里可以直接仿照 U-Net 结构来构造 Graph U-Net 框架来处理节点分类任务。

在 Graph U-Net 当中，首先通过一个图嵌入层将节点特征给转换成一个低维表示。之后通过堆叠多个图池化层与图卷积层组合的编码模块，来抽取图中节点的高层特征。图池化层减少了图形的大小来编码高阶特征，而图卷积层则负责聚合每个节点的一阶信息。在解码器部分，我们堆叠与编码器部分相同数量的解码模块。每个解码块由一个图反池化层和一个图卷积层组成。图反池化层将图恢复到更高的分辨率结构中，而图卷积层则将邻域的信息聚合起来。在编码器和解码器层的相应块之间有残差连接，它将空间信息传输给解码器以获得更好的性能。残差连接可以通过加法或拼接完成。最后，我们采用一个图卷积层，在 Softmax 函数提供最后的预测结果。图4提供了一个在编码器和解码器中有两个块的 Graph U- Nets 的说明。

2.4 图连接增强

在图池化层当中，采样了一些重要的顶点用来构造一个更小的子图，进而做高层级特征的提取。由于在这样的一个池化过程当中，一些点被删去了，那么可能会导致一些本来存在相邻节点的节点在新的子图当中是孤立的。这种孤立节点会导致在新的子图当中，应用图卷积操作来聚合相邻节点的信息传递操作受到限制。为了解决这个问题，通过对图邻接矩阵进行幂运算，来提升子图的连接性，例如第 k 次幂即为 \mathbb{G}^k ，此时在原始图中通过 k 跳可达的节点都会构建直接连边。在这里令 $k = 2$ ，于是修改邻接矩阵公式如下所示：

$$A^2 = A^l A^l, \quad A^{l+1} = A^2(\text{idx}, \text{idx}) \quad (8)$$

这样采样之后的子图，实际上对应的是增强之后的邻接矩阵。一定程度上解决了经过采用进行图池化之后图中孤立节点的问题。

2.5 增强图卷积层

在图池化的运算过程当中，邻接矩阵在经过归一化之前，是通过 $\hat{A} = A + I$ 计算的，即对于图中的每一个节点都添加了一个自环连边。这样在图上进行信息传递的时候，实际上会给节点自身的特征向量和其邻居节点的特征向量提供相同的权重。而在这里，由于关注的是节点分类任务，从直觉上思考，节点自身的特征向量对于节点级别的预测会更关键，所以我们在消息传递的过程当中，给予节点自身的特征向量更高的权重以凸显其重要性。于是之前的运算公式被改为 $\hat{A} = A + 2I$ ，来加强自环连边。这是在图结构数据处理当中常用的技巧，在 `pytorch_geometric` 库所实现的 GCN 层当中也提供了直接的接口支持。在实验结果当中证明，这种对于图卷积层的加强能够提升最终的分类效果。

3 结果

Graph U-Nets 在原论文中测试了节点分类和图分类两种任务。在官方开源的代码仓库当中，提供了图分类任务相关的代码以及配置文件，这里接用 PyTorch Geometric 框架，主要针对于节点分类任务进行了复现，采用的数据集为 Cora、CiteSeer 和 PubMed 三个数据集。在表1中有数据集的基本信息。

表 1: 节点分类数据集

Dataset	Nodes	Features	Classes	Training	Validation	Testing	Degree
Cora	2708	1433	7	140	500	1000	4
Citeseer	3327	3703	6	120	500	1000	5
Pubmed	19717	500	3	60	500	1000	6

在复现实验当中，所有在原始论文当中所提到的超参数均保持一致，超参数具体值如表2所示。这里每一层采样的节点数目依次为 2000，1000，500，200。为了防止过拟合，采用强度为 0.001 的 L2 正则化。关于 Dropout，对于邻接矩阵的保有比例为 0.8，对于特征矩阵的为 0.08。层

数默认为 4 层。关于在论文中没有具体描述的参数，这里隐层单元数量对于 Cora 和 CiteSeer 数据集设置为 128，对于 PubMed 数据集设置为 256，残差连接的连接方式为加法。

表 2: 超参数设置

Hyperparameters	Values
L2 - regularization	0.001
Dropout - adjacency matrices	keep rates 0.8
Dropout - feature matrices	keep rates 0.08
Num of Sample Nodes	2000, 1000, 500, 200
depth	4

在表3当中为节点分类任务的三个数据集上准确率的效果，这里同原论文当中所发布的结果进行对比。可以发现由于超参数的设置与一些实现细节上的问题。节点分类准确率同原始论文的结果有一些差距。

表 3: 节点分类准确率

Models		Cora	Citeseer	Pubmed
g-U-Nets	(Official)	84.4±0.6%	73.2±0.5%	79.6±0.2%
	(Unofficial)	82.1±1.1%	66.7±1.6%	78.0±1.0%

由于网络深度在深度神经网络，尤其是图神经当中都是一个非常重要的超参数，所以针对于此，进行了实验来研究网络深度和节点分类精度方面的性能之间的关系。在节点分类任务中使用了不同的网络深度，并统计了分类准确率。在表4当中有具体的统计结果。从结果中我们可以看到，随着网络深度的增加，性能也在提高，直到深度为 4，在之后由于过拟合原因，和图神经网络存在的过度平滑的问题，网络深度的提升反而会导致分类准确性的下降。

表 4: 不同网络深度对于准确率的影响

Depth	Cora	Citeseer	Pubmed
1	81.1±0.9%	66.1±1.5%	77.6±1.0%
2	81.7±1.3%	66.7±1.1%	76.8±1.1%
3	81.9±1.2%	66.5±1.3%	77.9±1.0%
4	82.1±1.1%	66.7±1.6%	78.0±1.0%
5	80.5±1.1%	66.6±1.6%	77.4±0.8%

在上述实验中，我所采用的图池化层通过对于邻接矩阵做了二次幂运算来提升图上的连接性能。在这里，对节点分类任务进行了消融实验，探究基于 Graph U-Nets 的图连接性增强的好处。除去在图池化层当中移除图连接性增强，保持其他设置不变，在同样的三个数据集上进行对比。表5提供了有和没有图连接性增强的 Graph U-Nets 的比较结果。结果显示，在所有三个数据集上，没有图连接性增强将导致一致的性能下降。说明对于邻接矩阵做幂运算来增强图的连通性，可以帮助提高图的连通性和在采样之后的子图中节点之间的信息传递。

表 5: 关于增强邻接矩阵的消融实验

Model	Cora	Citeseer	Pubmed
g-U-Nets without augmentation	80.3±0.7%	65.9±1.2%	77.2±0.3%
g-U-Nets with augmentation	82.1±1.1%	66.7±1.6%	78.0±1.0%

表 6: 关于池化层的消融实验

Model	Cora	Citeseer	Pubmed
g-U-Nets without gPool or gUnpool	81.2±1.5%	67.6±1.3%	79.0±0.7%
g-U-Nets with gPool or gUnpool	82.1±1.1%	66.7±1.6%	78.0±1.0%

为了考察所提出的图池化操作和图反池化操作到底起到了什么作用，这里针对于图池化层和图反池化层做了消融实验。在没有图池化层的网络，实际上就是一个仅仅包含堆叠图卷积层与残差连接的网络。在原始论文的结果中，在 Cora、CiteSeer 和 PubMed 数据集上，Graph U-Nets 比没有图池化层和图反池化层的 g-U-Nets 有更好的性能，幅度分别为 2.3%、1.6% 和 0.5%。但是在复现结果的实验当中，仅仅在 Cora 数据集上有更好的效果，在 CiteSeer 和 PubMed 数据集上去除池化操作的效果更好。可以考虑到的是，去除池化操作，在相同的层数中，没有池化操作的感受野会比有池化操作的更受限。于是会更关注临近节点的信息，引文信息网络实际上对于局部信息更为看重，所以感受野缩小可能会有更好的效果也是可解释的。加之原文当中三个数据集所带来的提升依次减少，所以在 CiteSeer 和 PubMed 数据集上相对提升更少也是合理的。

4 分析

复现结果在节点分类任务上，并没有达到论文当中所描述的效果。这里采用的是标准的引文数据集，应该不会存在因为数据划分的方式不同所导致的数据上的偏差。由于作者在对应的 GitHub 仓库中只公开了图分类相关的配置文件和代码，所以这里没有办法针对于实现细节进行对比。在包括对于图连接性的增强和对于图池化与图反池化层的消融实验表明，通过对于邻接矩阵做幂次操作可以有效地提升采样子图的连接性，图池化层和图反池化层在部分数据集上能够得到更好的效果。其中 Graph U-Net 整体带有残差连接的 U-Net 结构，能够有效的融合高层级和底层级的特征，在节点分类任务上能够提供不错的增益。

表 7: 原始论文关于池化层的消融实验

Model	Cora	Citeseer	Pubmed
g-U-Nets without gPool or gUnpool	82.1±0.6%	71.6±0.5%	79.1±0.2%
g-U-Nets with gPool or gUnpool	84.4±0.6%	73.2±0.5%	79.6±0.2%

5 结论

在 Graph U-Nets 原论文 [2] 当中，提出了图池化操作和图反池化操作，将 U-Nets 结构拓展到了图结构数据上。在结果中可以发现，U-Nets 的结构对于这种节点级别预测的任务上能够取得不错的帮助。同时在消融实验中可以发现，关于邻接矩阵的增强操作能够很大地帮助解决形成的新图上孤立节点的问题。在复现论文的过程当中，对于图神经网络的实现和具体细节有了更为深刻的理解。

参考文献

- [1] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C]// International Conference on Medical image computing and computer-assisted intervention. [S.l.]: Springer, 2015: 234-241.
- [2] GAO H, JI S. Graph u-nets[C]//international conference on machine learning. [S.l.]: PMLR, 2019: 2083-2092.