東南大學

毕业设计(论文)报告

题 目： 基于 Transformer 的图分类研究

学　　号：	71118415
姓　　名：	叶宏庭
学　　院：	软件学院
专　　业：	软件工程
指导教师：	孔佑勇
起止日期：	2022/01-2022/06

东南大学毕业（设计）论文独创性声明

本人声明所呈交的毕业（设计）论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

论文作者签名：_____ 日期：_____年_____月_____日

东南大学毕业（设计）论文使用授权声明

东南大学有权保留本人所送交毕业（设计）论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学教务处办理。

论文作者签名：_____ 导师签名：_____

日期：_____年_____月_____日 日期：_____年_____月_____日

摘　要

图结构数据 (Graph) 广泛地存在于我们的生活中，用于表示复合对象元素之间的复杂关系，例如社交网络、引文网络、生物化学网络和交通网络等。不同于结构规则的欧式数据，图数据的结构复杂，蕴含着丰富的信息。图分类是机器学习和数据挖掘中被广泛研究的问题，并在计算机视觉、生物科学、神经科学等领域具有广泛的应用。然而，现有的图分类研究都忽略了图拓扑结构信息的有效利用，并且现有的主流方法都无法很好学习长距离成对节点之间的关系。得益于 Transformer 模型的快速发展与广泛运用，我们发现 Transformer 模型可以很好地学习和表示长距离依赖信息。因此本文提出了一种基于 Transformer 的图分类方法，旨在充分利用图的节点属性和拓扑结构信息，在图分类任务上获得良好的性能。本文在多个真实世界的图分类基准数据集中，将本文提出的 GraphTransformer 方法与主流的基线方法进行了对比实验。最终实验结果表明，本文提出的 GraphTransformer 方法可以在图分类任务上具有良好的表现以及稳定性。同时，证明了 Transformer 模型在图分类任务中的可行性以及有效性。

关键词：图分类，图拓扑结构，Transformer

ABSTRACT

Graph-structured data (Graph) exists widely in our lives to represent complex relationships between elements of composite objects, such as social networks, citation networks, biochemical networks, and transportation networks. Different from European-style data with structural rules, graph data has a complex structure and contains rich information. Graph classification is a widely studied problem in machine learning and data mining, and has a wide range of applications in computer vision, biological science, neuroscience, and other fields. However, the existing graph classification studies have neglected the effective utilization of graph topology information, and none of the existing mainstream methods can learn the relationship between long-distance paired nodes well. Benefiting from the rapid development and wide application of the Transformer model, we found that the Transformer model can learn and represent long-distance dependency information well. Therefore, this paper proposes a Transformer-based graph classification method, which aims to make full use of the node attributes and topology information of graphs to obtain good performance on graph classification tasks. This paper compares the proposed GraphTransformer method with mainstream baseline methods in several real-world graph classification benchmark datasets. The final experimental results show that the GraphTransformer method proposed in this paper can have good performance and stability on graph classification tasks. At the same time, it proves the feasibility and effectiveness of the Transformer model in the task of graph classification.

KEY WORDS: Graph classification, Graph topology, Transformer

目　　录

第一章 绪论

1.1 课题背景和意义

图数据 (graph data)在我们的日常生活中广泛存在，主要用于表示复合对象实例元素之间的复杂关系，例如引文网络、生物化学网络、社交网络和交通网络等。与结构规则的欧式数据相比，不同的是，图数据的结构更为复杂，蕴含着更为丰富的信息。近年来，对图数据的研究成为了学术界的一个热点。图上的研究问题主要包括图分类[25][26]，节点分类[23][24]，链路预测[27]等。

图分类是机器学习和数据挖掘中被广泛研究的问题，并在计算机视觉、生物科学、神经科学等领域被广泛地应用。例如化学信息学之中，通过对分子图进行分类来判断化合物分子的抗癌活性、诱变性、毒性等；生物信息学中，通过对蛋白质网络进行分类判断蛋白质是不是酶，是不是具有对某种疾病的治疗能力。从这个角度上看，图分类研究具有十分重要的意义。

近年来，深度学习已被证明在许多领域都取得了巨大的成功，从声学、计算机视觉到自然语言处理。因此，深度学习能否在图分类任务中也取得较好的表现成为了一个谜题，也激发了广大研究者的研究热情。大量的研究致力于将深度学习方法应用于图的研究，使得图分析技术取得了有益的进展。然而，由于图的独特特性，将深度学习应用到复杂的图数据存在一定的挑战。图神经网络(Graph Neural Networks，GNNs)被广泛用于图数据处理，应用于图分类研究[10][11]。图神经网络的加入，让图分类研究达到了一个新的热潮，图神经网络在图分类问题上表现出了强大的学习能力，在大多数的数据集上都战胜了传统的图分类方法，可以说是图分类问题进入了一个新的研究领域。但是，图神经网络的计算方法中，主要依靠于迭代邻域聚合的方法来更新节点表示，从而在这个过程中忽略了从图的拓扑结构域中产生的特征。不同于图像等欧式数据，图具有不规则性，且现实世界中的很多图数据都具有两个属性，即图的拓扑结构和图中每一个节点的特征。对于图分类任务，现有的方法[28]通常对图的节点表示应用一些置换不变读出函数来生成整个图的表示。常见的读出函数将每个图视为一组顶点表示，因此会忽略顶点之间的交互，即图的结构信息。由于图结构与特征信息结合带来的透明性不足，图的拓扑结构信息是否被编码到生成的图表示中还不够明确。

为了解决现有的图分类模型没有充分利用图的拓扑结构信息的问题，本课题研究提出

一种基于 Transformer 的图分类方法，旨在充分利用图的节点属性和拓扑结构信息，在图分类任务上获得良好的性能。

1.2 研究现状

1.2.1 基于图核的图分类方法

核方法的基本原理是把低维空间中非线性可分问题转换为高维空间中线性可分问题，该方法的核心在于成对数据之间相似性的度量。图核（Graph kernel）是一种比较特征的核方法，其核心在于如何计算图之间的相似度，如何进行度量。

基于图核的图分类方法[20][21]实质上是先利用图核方法显式或者隐式地计算成对图之间的相似度，然后再利用核机器通过图之间的相似度进行图分类。图核方法在图分类问题中应用比较广泛，其核心在于定义合适的核函数。目前常见的核方法有，基于路径的图核、基于子图的图核以及基于子树的图核。这三类方法都是基于 R-卷积核的框架来定义的。三类方法的主要区别就是对图的分解方式不同。基于路径、子图、子树的方法分别将图分解为路径、子图以及子树，然后利用显式或者隐式的方法来计算图的相似度。通常来说，当显式方法可行时，并且得到的特征向量维数并不高，那么它比隐式方法的计算更快，内存效率也更高。

图核方法充分结合了图的表示能力以及核方法的区分能力，可以解决很多基于图相似度计算的图分类任务。但是在这些方法中，特征表示和分类是两个独立的过程，需要分开进行，无法做到统一优化。除此之外，基于图核的方法计算复杂度一般都比较高，无法很好的应用在大型图之中。

1.2.2 基于图匹配的图分类方法

图匹配原理上也是针对图的相似度进行度量，从而达到分类目的。图匹配方法目前主要分为两类，精确图匹配和非精确图匹配。在精确图匹配方法中，需要输入图中的节点映射关系作为已知信息，该方法的主要应用领域有最大公共子图判别、图同构以及子图同构。非精确图匹配基本可以形式化成图编辑距离的度量。

基于图匹配的图分类方法[3]通常用于计算机领域中，二分函数相似度的搜索，检测代码中是否存在已知的不稳定结构，同时也经常被用在生物化学领域之中，利用图的相似度分数来判断化合物的属性。

1.2.2 基于图神经网络的图分类方法

近年来，研究人员的研究兴趣集中于如何将深度学习方法应用在图数据中。得益于研究人员的努力与计算资源的提升，研究人员在卷积网络、循环网络和深度自动编码器思想的基础之上定义和设计了用于处理图数据的神经网络结构，由此一个新的研究热点——“图神经网络”应运而生。

目前主流的图神经网络主要包括：图卷积网络（Graph Convolution Networks）[16]、 图注意力网络（Graph Attention Networks）[9]、图自编码器（Graph Autoencoders）、图生成网络（Graph Generative Networks）和图时空网络（Graph Spatial-temporal Networks）[15]。

图神经网络（GNNs）使得深度学习网络能够处理结构化输入，如分子网络或社会网络。GNNs 学习映射，从其邻居的结构和特征计算图节点或边的表示。这种邻域局部聚合利用了由图的连通性编码的关系诱导偏差。与卷积神经网络（Convolutional Neural Network，CNN）类似，GNNs 可以通过叠加层来聚合来自局部邻域以外的信息，有效拓宽 GNNs 的接受域。

但是，据观察，当 GNNs 的深度超过几个层时，GNNs 性能显著下降。这种限制损害了 GNNs 在全图分类和回归任务中的性能，在这些任务中，我们希望预测一个描述整个图的目标值，该目标值可能依赖于长距离依赖关系，而具有有限接受域的 GNNs 可能无法捕获这些依赖关系。例如，考虑一个大型图，节点 A 必须关注距离 k 跳远的节点 B。如果我们的 GNNs 层只在一个节点的单跳邻居上聚合，那么需要一个 k 层 GNNs。然而，这个 GNNs 的接收域的宽度将以指数方式增长，稀释来自节点 B 的信号。也就是说，简单地将接收域扩展到 K-hop 邻域可能无法捕捉到这些远程依赖关系。通常，“太深”的 GNNs 导致节点表示在整个图上折叠成等价的，这种现象有时被称为过度平滑或过度抑制。总而言之，通用 GNNs 架构的最大上下文大小是有限制的。

许多作者提出通过类似于今天 CNN 中发现的中间池操作来解决过度平滑问题。图池化操作在渐进 GNNs 层中逐渐使图变粗，通常是通过将邻域分解为单个节点。从理论上讲，通过减少信息传播的距离和过滤掉不重要的节点，分层粗化应该能够实现更好的远程学习。然而，到目前为止，还没有发现有像 CNN 池化一样普遍适用的图池化操作。最先进的结果通常是使用没有中间图粗化的模型，一些结果表明邻域局部粗化可能是不必要的或适得其反的。

1.2.3 基于 Transformer 的图分类方法

Transformer 体系结构[8]最早是由谷歌公司开发的，用于自然语言的处理。Transformer 结构是完全基于注意力机制的，它可以在每一次序列的运算过程中，充分发挥其对节点信息的作用。现在，Transformer 体系结构在很多方面都占据着重要的地位，比如自然语言处理和计算机视觉。但是与主流的 GNNs 相比，它在流行的图级预测任务排行榜上还没有取得具有竞争力的表现。因此，Transformer 能否在图表示学习中取得良好表现仍然是一个谜。但是从这两年的研究中，我们发现越来越多的研究者将 Transformer 架构应用于图分类问题之中来，在 Transformer 架构的基础上，通过对某些模块的替换，或者是将图的其他信息也加入到编码之中来，形成了各种的 X-ormer 体系架构，并且有的 X-ormer 架构也被实验证明可以在图分类问题中取得良好的表现，例如 GraphiT[18]，Graphormer[17]。这些研究都证明了 Transformer 架构在图分类领域的强大学习能力。

1.3 本文研究内容

基于以上综述内容，本文将研究基于 Transformer 的图分类模型，期望能够在图分类任务中融入图的拓扑结构信息，并且通过 Transformer 架构来学习图中所谓的长距离依赖信息。首先，我们将调研图拓扑结构相关的研究，统计并且整理常用的图拓扑结构指标，并且查询相关库函数等来进行各个指标的计算。其次，我们需要研究如何将拓扑结构编码与节点属性进行结合才能一起输入到深度学习网络模型之中。融入图的拓扑结构信息旨在充分利用图的节点属性和拓扑结构信息建立图分类模型，更好地完成图分类任务。在网络模型选取方面，前期调研发现，GNNs 具有很强的学习能力，能够很好的聚合网络中的邻域特征信息，但是 GNNs 也存在前文所述的不足之处，为了弥补 GNNs 的不足，通过继续调研，我们发现作为 seq2seq 模型的 Transformer 架构能够在长距离依赖信息学习中取得良好表现，并且目前也存在很多相关工作可供参考，因此本文将研究基于 Transformer 的图分类模型。经过调研分析，本文将采用 GNNs 与 Transformer 组合的方式来进行网络模型设计，期望利用 GNNs 来学习图中的邻域特征信息，通过 Transformer 模型来学习图中长距离成对节点之间的相互依赖，并且本文将采用独特的 GNN“读出”模块，加入了虚拟节点来表示全图特征信息，最终进行线性变换达到图分类的目的。

我们的研究方法称之为 GraphTransformer，在编码层面，通过计算图的拓扑结构信息，并将其融入图的节点属性，输入到深度学习网络之中。在网络结构层面，将采用 GNNs 与

Transformer 架构融合的思想，使用 GNNs 学习节点近邻域的结构本地表示，同时利用 Transformer 架构作为强大的全局推理模块，学习长距离依赖信息，最后采用独特的“读出”模块来进行图分类。

1.4 本文组织结构

本文将分为五个章节对我们的研究进行阐述，其中：

第一章，绪论部分，介绍本课题的研究背景、研究现状以及本课题研究内容。

第二章，相关技术与原理，介绍本文中利用到的相关技术与原理，包括图数据、图拓扑结构信息、图神经网络以及 Transformer 网络模型等。

第三章，GraphTransformer 模型设计与实现，介绍 GraphTransformer 模型的研究动机、模型框架、实现细节等。

第四章，实验设计与结构，介绍本文的实验环境、实验设计、实验结果以及参数实验。

第五章，总结与展望，介绍本文工作的大致内容，并且对目前研究的相关展望。

第二章 相关技术与原理

本章将围绕图数据、图拓扑结构、图神经网络以及 Transformer 架构展开，对本课题涉及的相关理论与技术进行介绍。

2.1 图数据与图拓扑结构

2.1.1 图数据介绍

在计算机科学中，图是一种抽象数据类型，用于实现数学领域图论中的无向图和有向图的概念。图的数据结构包含两个集合，其中一个有限的集合作为节点集合，另一个无序对或有序对构成的集合作为边的集合。节点可以作为图结构的一部分，也可以是用整数下标或引用表示的外部实体。图的数据结构中常常还包含和每条边相关联的某些数值，例如一个数值或一个标号。

在计算机中，图具有三种常见的数据结构，邻接表、邻接矩阵以及关联矩阵。其中，在邻接表中，节点存储为对象或记录，并且为每个节点创建一个单向列表。这些单向列表可以按节点顺序存储其余的信息，将每个节点的所有邻接节点存储在该节点的单向列表之中；邻接矩阵是一个二维矩阵，其中行与列分别表示边的起点和终点（即起始节点与终点节点），每个顶点上的值需要存储在外部，矩阵中可以存储边的权值。关联矩阵也是一个二维矩阵，行表示顶点，列表示边。矩阵中的数值用于标识顶点和边的关系（是起点、是终点、不在这条边上等）。邻接表在稀疏图上比较有效率，邻接矩阵则常在图比较稠密的时候使用，判断标准一般为边的数量$|E|$接近于节点的数量的平方$|V|^2$；邻接矩阵也在查找两节点邻接情况较为频繁时使用。除此之外，其他表示和存储图数据结构的方式还包括十字链表、链式前向星、邻接多重表等。

在处理图问题时，对图数据处理的并行计算要求很高，但是图并行处理通常存在以下几种问题：处理大量的数据、数据不分散、求解非常规问题、数据存取在计算中比例很高等等。面对这些困难问题，并行计算中的图表示和存储方式就显得尤其重要，如果选取不合适的图表示方式，将带来很多不必要的计算开销，进而影响整体算法的速度与扩展性。

表 2.1、表 2.2 给出在图上进行各种简单操作时的时间复杂度及空间复杂度。其中，$|V|$表示节点数量，$|E|$表示边的数量。

总的来看，邻接表在稀疏图上具有较好的效率，邻接矩阵则是在图较为稠密的时候使用，判断图的稠密、稀疏标准一般为边的数量$|E|$接近于节点数量的平方$|V|^2$。邻接矩阵同

时也在查找两节点的邻接情况时使用频繁。

表 2.1 图数据结构的空间复杂度[①]

操作	邻接表	邻接矩阵	关联矩阵
存储一张图	$O(\|V\|+\|E\|)$	$O(\|V\|^2)$	$O(\|V\|\cdot\|E\|)$

表 2.2 图数据结构的时间复杂度[①]

操作	邻接表	邻接矩阵	关联矩阵
添加节点	$O(1)$	$O(\|V\|^2)$	$O(\|V\|\cdot\|E\|)$
添加边	$O(1)$	$O(1)$	$O(\|V\|\cdot\|E\|)$
移除节点	$O(\|E\|)$	$O(\|V\|^2)$	$O(\|V\|\cdot\|E\|)$
移除边	$O(\|V\|)$	$O(1)$	$O(\|E\|)$

2.1.2 图拓扑结构

在数学中，拓扑图论是图论的一个分支。它研究了图在曲面中的嵌入、图的空间嵌入以及图作为拓扑空间。本课题所研究的图拓扑结构，主要是图作为一个网络空间时，网络空间存在的各种拓扑指标来作为图的拓扑结构。在网络空间中，每一个节点、每一条边和整个图上都存在各种拓扑属性，包括：度、节点效率、中心性指标、小世界等。这些属性都能够表示图中节点、边以及全图的各种结构信息，这些信息有别于节点、边本身的特征，这些信息能够很好的反映节点、边在图中的重要性以及影响力。因此如何有效利用这一重要信息，成为本课题研究的重点。

2.2 图神经网络

在过往的几年里，传统的深度学习方法在欧式数据的特征提取任务中取得了非常良好的表现，但是在现实世界中，存在大量的场景数据都是基于非欧式空间生成的，在解决基于非欧式空间数据的任务之中，传统的深度学习方法的表现仍难以令人满意。例如，在如今较为广泛的电子商务场景下，一个基于图的学习系统能够很好地利用用户和产品之间的交互信息来做出较为准确的推荐，但由于图一般情况下都具有一定的复杂性，这使得现有

[①] 选自维基百科，图数据结构，https://zh.wikipedia.org/wiki/图_(数据结构)

的深度学习算法在处理复杂图时遇到了很多难以克服的问题。这是因为图是不规则的，每一张图都有一个无序的、大小不一的节点，而图中的每一个节点都有一定数目的邻接节点，这就使得某些重要的运算（如卷积）可以在图像上方便地进行运算，但是无法直接应用到图结构数据中。另外，当前的深度学习算法中，最关键的一点是，数据样本彼此间是独立的。然而，此假定并不适用于此，因为图中的每个节点都会通过边与该图中其它节点相连接，从而可以抓住各节点之间的关联性。

近年来，研究者们致力于研究如何将深度学习方法有效应用在图数据上。得益于研究人员的努力与计算资源的提升，研究人员在卷积网络、循环网络和深度自动编码器思想的基础之上定义和设计了用于处理图数据的神经网络结构，由此一个新的研究热点——“图神经网络”应运而生。

在本章节中，将介绍五中常用的图神经网络，分别是：图卷积网络、图注意力网络、图自编码器、图生成网络和图时空网络。

图卷积网络，图卷积网络是利用传统的卷积算法对图进行统计分析。学习一个映射函数 $f(\cdot)$ 是该方法的核心，在这个映射函数的运算下，图中的节点 v_i 可以通过卷积算子来聚合它自身的特征 $\boldsymbol{x_i}$ 与它的邻居节点的特征 $\boldsymbol{x_j}$ $(j \in N(v_i))$来生成节点 v_i 的新节点表示。现有的大部分图神经网络模型都是基于图卷积网络来构建的，包括基于时空网络、生成模型和自动编码器的模型等。

图卷积网络的基本思想可以分为基于频谱的和基于空间的。频谱分析法是从图信号处理的观点出发，通过引入滤波器来定义图卷积，将图卷积运算理解为消除图信号中的噪声。空间分析法是在图卷积网络算法的基础上，采用基于空间的方式来表达图卷积，使其与图池化模块相交织，使其粗化成更高的子结构。

图注意力网络，在基于序列的任务中，注意力机制已经被广泛地使用，它的优点是能够放大数据中最重要部分的影响。这个特性已经被证明在许多任务中有重要作用，例如自然语言理解和机器翻译。如今越来越多的模型尝试融入注意力机制，图神经网络也受益于此，它在聚合过程中使用注意力，整合多个模型的输出，并生成面向重要目标的随机行走。

图自编码器，图自动编码是一种图嵌入算法，它通过神经网络构造把图中的节点用低维向量表示出来。传统的方法是将多层感知机用作编码器，并通过解码器重构出节点的邻近统计数据，如 positive pointwise mutual information（PPMI）或一阶和二阶近似值。最近，

研究人员已经探索了将 GCN 作为编码器的用途，将 GCN 与 GAN 结合起来，或将 LSTM 与 GAN 结合起来设计图自动编码器。

图生成网络，图生成网络的目的是根据所观测到的图数据集合来产生新的图数据。图生成网络的很多方法都是以具体的领域为基础的。例如，在分子图生成中，一些工作模拟了称为 SMILES 的分子图的字符串表示。在自然语言处理中，通常以给定的句子为条件生成知识图或语义图。最近，人们提出了几种通用的方法。一些工作将生成过程作为节点和边的交替形成因素，而另一些则采用生成对抗训练。这类方法要么使用 GCN 作为构建基块，要么使用不同的架构。

图时空网络，图时空网络的基本原理是同时捕获时间-空间关系。时空图一般是一个整体的图形，它的每一个节点的输入都是随着时间而改变的。例如，在交通网络中，每个传感器作为一个节点连续记录某条道路的交通速度，生成一个节点的时序数据，其次交通网络中的边由传感器对之间的距离决定。图时空网络的目标是可以预测未来的节点值或标签，或者预测时空图标签。最近的相关研究仅仅探讨了 GCNs 的使用，GCNs 与 RNN 或 CNN 的结合，以及根据图结构定制的循环体系结构。

2.3 Transformer 网络架构

Transformer 最初是由谷歌公司提出的，目的是作为一种序列到序列（seq2seq）模型以更好地解决机器翻译任务。而后来的研究表明，基于 Transformer 的预训练模型（PTM）在各项任务中都取得了最优的表现。因此，Transformer 已经成为了自然语言处理（NLP）领域的首选架构，尤其是 PTM。除了语言相关的应用，Transformer 还被广泛运用于计算机视觉（CV）、音频处理、化学和生命科学等相关领域。由于取得了巨大的成功，过去几年研究者又提出了各种 Transformer 变体（又名 X-former）。这些 X-former 主要从以下三个不同的角度改进了最初的 Transformer。

模型效率。应用 Transformer 的一个关键挑战是其处理长序列时的效率低下，这主要是由于自注意力（self-attention）模块的计算和内存复杂度。改进的方法包括轻量级 attention（例如稀疏 attention 变体）和分而治之的方法（例如循环和分层机制）。

模型泛化。由于 Transformer 是一种灵活的架构，并且对输入数据的结构偏差几乎没有假设，因此很难在小规模数据上进行训练。改进方法包括引入结构偏差或正则化，对大规模未标记数据进行预训练等。

模型适配。这一系列工作旨在使 Transformer 适应特定的下游任务和应用程序。

虽然可以根据上述角度来组织 X-former，但许多现有的 X-former 可能会解决一个或几个问题。例如，稀疏 attention 变体不但降低了计算复杂度，而且在输入数据上引入了结构先验以缓解小数据集上的过度拟合问题。因此，主要根据 X-former 改进 Vanilla Transformer 的方式进行分类更加有条理：架构修改、预训练和应用。

Transformer 是一个序列到序列（seq2seq）的模型， 编码器-解码器结构是大部分竞争性神经序列转导模型的基本结构，Transformer 也不例外。在这里，编码器将符号表示序列映射到序列的连续表示，即输入序列 $(x_1, x_2, \ldots, x_n)$ 映射到连续表示 $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)$。给定 $\boldsymbol{z}$，解码器再逐步地生成一个输出序号序列 $(\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_\mathrm{n})$。在每个步骤中，该模型都会进行自回归，在下一次产生的符号单元中，以之前产生的符号单元为输入。

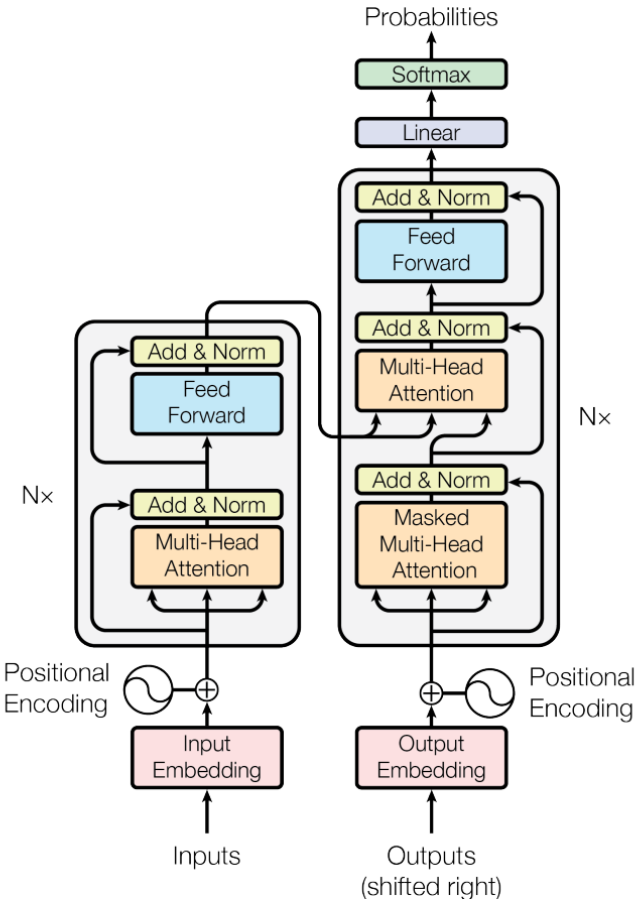

图 2-1 Transformer 模型整体框架图[8]

Transformer 也遵循上述的整体框架，使用堆叠的自注意力（self-attention）和逐点（point-wise），在编码器和解码器中都使用了全连接层，Transformer 的整体框架如图 2-1 所示。

2.3.1 编码器和解码器堆栈

编码器，由$N = 6$个相同的层组成了编码器，其中的每层内部都存在两个子层，多头自注意力机制是第一个子层，位置（position-wise）全连接前馈网络是第二个子层。并且使用残差连接包围了每个子层，层归一化操作紧跟在每个子层之后，即每个子层的输出都是$LayerNorm(x + Sublayer(x))$，其中$Sublayer(x)$是子层实现的函数。

解码器，同样由一堆$N = 6$个相同的层组成解码器，编码器中存在的两个子层在这里都有，此外，解码器中还插入了第三个子层，主要的作用是计算多头注意力编码器堆栈的输出。与编码器一样，使用残差连接与层归一化包围了每一个子层。同时，在解码器中，自注意力子层也做了一些调整，以保护后续位置的影响。这种掩藏机制，目的是确保位置 i 的预测只能依赖于位置小于 i 的已知输出。

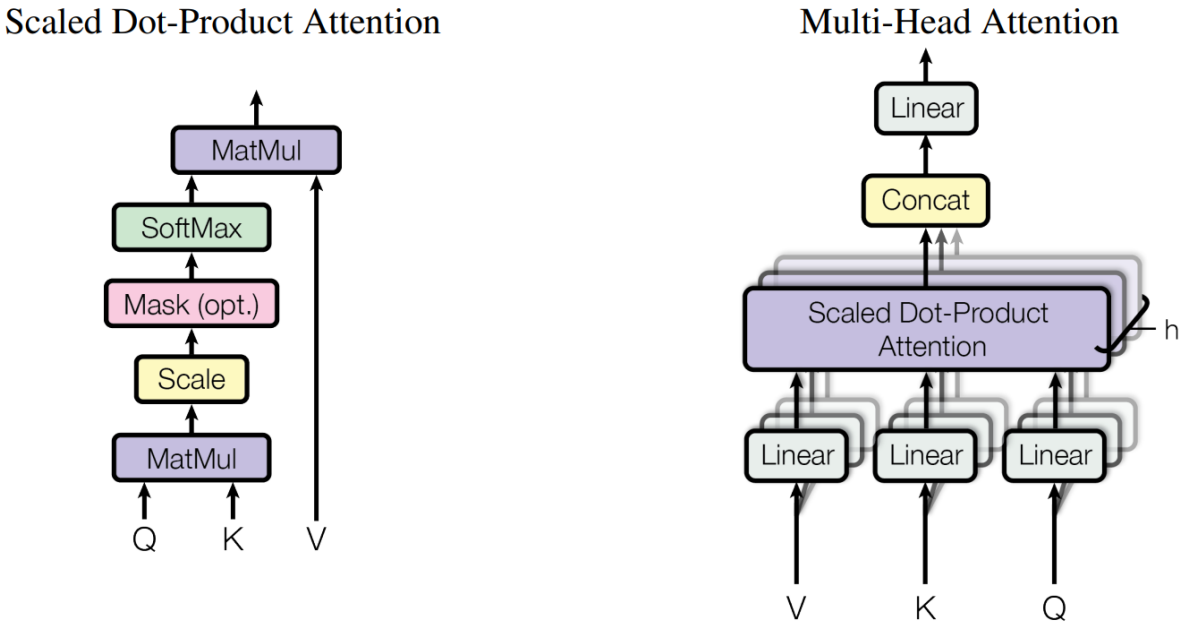

图 2-2 缩放点积注意力和多头注意力机制[8]

2.3.2 注意力

注意力功能可以用来表示查询和一系列关键值对的映射，其中查询、关键字、值和输出都是向量。该输出用一个加权和的数值来计算，在该数值中，通过查询和对应的关键字的相容性功能来计算指定的权重。

2.3.3 缩放点积注意力

Transformer 模型中使用了较为独特的“缩放点积注意力”。维度为 d_k 的查询和键以

及维度为 d_v 的值是该模块的输入。注意力的具体计算过程是，计算查询与所有键的点积，并除以 $\sqrt{d_k}$，每个值的权重是通过 softmax 函数来获取的。

在实践之中，为了并行化计算一组查询的注意力函数，我们将输入打包在一起到矩阵 Q 之中，键和值也同时打包到矩阵 K 和矩阵 V 之中，因此整体的计算公式为：

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

现在使用最多的是点积注意力和加性注意力。除了缺少标度系数 $\sqrt{d_k}$ 外，点积注意力的性质与上面所说的基本相同。加性注意力理论是利用前馈网络在单一的隐藏层中进行相容度的计算。尽管二者在理论上的复杂程度类似，但由于点积关注能够利用最优方法来实现矩阵乘法码，因而在实际操作中更快更节省空间。

2.3.4 多头注意力

Transformer 模型并不是简单使用单个注意力函数来计算d_{model}维的查询、键和值，而是采用了 h 次不同的、学习的方式对d_k，d_k，d_v进行线性投影来计算注意力，并且发现这样的作法是有益的。最后再把每次投影计算的输出拼接起来，再次进行线性计算，产生最终的输出值。多头注意力允许模型共同关注来自不同表示空间和位置的信息。对于单个的注意力头，平均化会抑制这种情况。多头注意力计算公式如下：

$$MultiHead(Q,K,V) = Concat(head_1, head_2, \dots, head_h)W^O \tag{2.2}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2.3}$$

其中，$W_i^Q \in \mathbb{R}^{d_{model}\times d_k}$，$W_i^K \in \mathbb{R}^{d_{model}\times d_k}$，$W_i^V \in \mathbb{R}^{d_{model}\times d_v}$以及$W^O \in \mathbb{R}^{hd_v\times d_{model}}$。

2.3.5 位置前馈网络

另外，Transformer 模型的编码器和解码器子层也包括一个完全连接的前向神经网络，该神经网络用于各个子层次的同一位置。该前馈网络包括了两种线性转换，并在中间加入了 ReLU 的激活。

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{2.4}$$

尽管在不同的位置，这些线性变化都是相同的，但在不同层中所采用的参数却不尽相同。也就是，两个核心都是 1 的卷积。

2.3.6 嵌入与 Softmax

Transformer 模型与其他序列转导模型类似，都使用了学习嵌入来将输入 tokens 和输出

tokens 转换为 d_{model} 维的向量，同时使用了最常见的可学习线性变换和 softmax 函数来讲解码器的输出转换为预测的下一个 token 的概率。在 Transformer 模型中，两个嵌入层和 pre-softmax 之间共享同一个权重矩阵。

2.3.7 位置编码

在 Transformer 模型中，并不包含递归计算或者卷积计算，因此，为了能够充分利用序列的顺序信息，Transformer 模型必须加入 tokens 序列的一些相对或者绝对位置信息。为了达到这个目的，Transformer 模型在编码器和解码器底部的输入嵌入过程中加入了位置编码（positional encodings）。Transformer 模型中采用了 sin 和 cos 三角函数来进行位置编码，具体公式如下：

$$PE_{(pos,2i)} = sin\left(pos/10000^{2i/d_{model}}\right) \tag{2.5}$$

$$PE_{(pos,2i+1)} = cos\left(pos/10000^{2i/d_{model}}\right) \tag{2.6}$$

其中，pos 是位置，i 是维度。

2.4 本章小结

本章首先介绍了图数据相关基础，包含图数据结构定义、图数据结构表示方法等。其次介绍了图拓扑结构。之后介绍了图神经网络中的五个主流图神经网络方法。最后介绍了 Transformer 架构的整体细节。本章通过介绍以上的相关技术为后续的研究提供理论基础。

第三章 GraphTransformer 模型设计与实现

本章主要介绍本文提出的 GraphTransformer（简称 GraphTrans）模型，先讲述研究动机，表明设计 GraphTrans 的背景与意义，进而给出 GraphTrans 的模型框架图以及讲述 GraphTrans 的具体实现细节。

3.1 研究动机

正如前文背景所述，大多数的图分类方法中，都没有充分利用图的拓扑结构信息，因此本文旨在充分利用图的拓扑结构信息来进行图分类任务。本文将在图嵌入编码时融入前文提到的几种拓扑结构属性（度序列、节点效率、聚类系数、平均聚类系数、局部效率、全局效率、模块化系数、平均路径长度）。

其次，尝试通过堆叠 GNNs 层或层次池化的方式来对图进行长距离依赖学习并没有带来性能上的提高，虽然一些工作已经成功将单个 GNNs 层的接受域扩展到单跳领域以外，但是这种方式如何扩展到具有数千万个节点的大图之上，还有待观察。

通过调研，在计算机视觉领域的文献中发现了一种可行的替代方法。近年来，注意力机制可以代替传统的 CNN 卷积。注意力层可以学习复制由局部卷积引起的强关系诱导偏差。最近，一些最先进的计算机视觉任务方法中使用了在传统 CNN 主干上的类似注意力的子模块。这些结果都表明，虽然强关系归纳偏差有助于学习局部的、短距离的依赖，但是对于长距离依赖信息来说，结构松散的模块可能更受欢迎。

在本文的 GraphTrans 模型中，我们利用了图学习领域的这一见解，在模型中，使用传统的 GNNs 子网作为骨干，将学习长距离依赖信息的任务交给没有图空间先验知识的 Transformer 子网。我们的 Transformer 子网允许每个节点关注每个其他节点（大多数 Transformer 在图上运用时，只允许关注邻居节点），这样的作法激励 Transformer 学习最重要的节点-节点关系，而不是偏向于附近的节点。

本文期望基于以上两点动机，设计实现一种基于 Transformer 的图分类方法，旨在充分利用图的节点属性和拓扑结构信息，在图分类任务上获得良好的性能。

3.2 模型框架

GraphTrans 模型框架如图 3-1 所示，模型包含两个主要的模块：一个 GNNs 子网络，紧随一个 Transformer 子网络。整体流程为图输入—特征融合—GNNs 子网络—节点嵌入—Transformer 子网络—线性变换—softmax 函数—图分类结果。

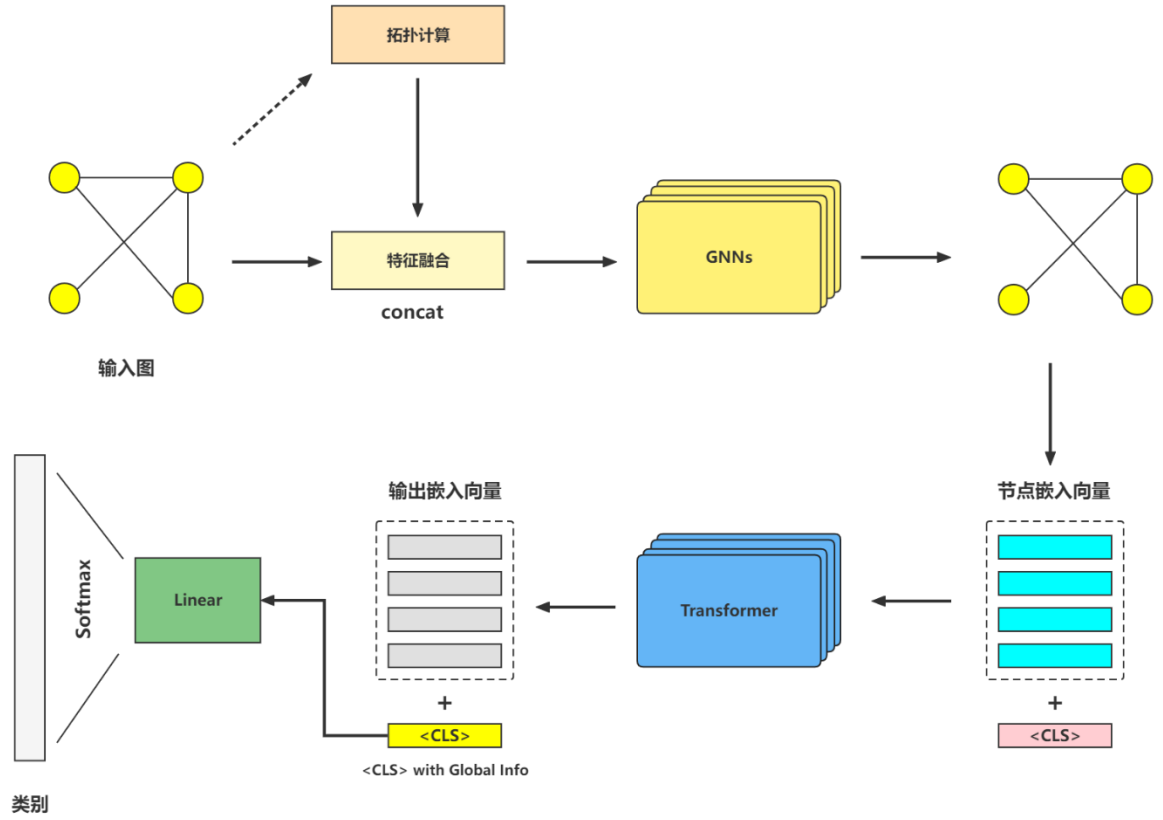

图 3-1 GraphTransformer 模型框架图

3.3 实现细节

3.3.1 拓扑特征计算

本文研究的一大重点就是在图分类任务中充分利用图拓扑结构信息，因此，在图数据输入到网络之前，我们需要进行拓扑特征的计算。这一步独立于整个 GraphTrans 网络存在，因为拓扑计算需要花费一定时间，所以需要在数据预处理阶段完成特征的计算，并且保存这些数据，在特征融合阶段进行使用。

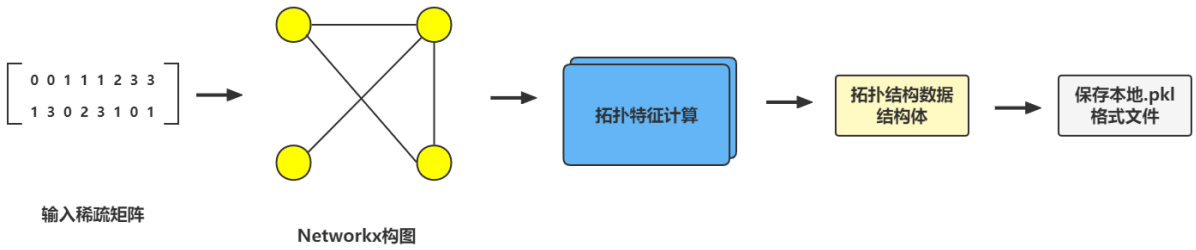

图 3-2 拓扑特征计算流程图

大致的计算流程为：图数据输入—读取稀疏邻接矩阵—使用 Networkx 进行构图—计算各个拓扑特征值（或向量）--赋值到预先定义的结构体中—保存结构体数据对象到.pkl 格式文件。

表 3.1 网络计算接口调用表

拓扑特征	Networkx 库接口	Numpy 库接口	说明
度序列	degree(graph)	/	/
节点效率	efficiency(graph, node_i, node_j)	sum()	sum(efficiency(graph, node_i, node_j)), $i \neq j \in V$
聚类系数	clustering(graph)	/	/
平均聚类系数	average_clustering(graph)	/	/
局部效率	local_efficiency(graph)	/	/
全局效率	global_efficiency(graph)	/	/
模块化系数	greedy_modularity_communities(graph) modularity(graph, communities)	/	贪心获取最优化分再计算模块化系数
平均路径长度（调和平均）	harmonic_centrality(graph)	/	/

下面主要介绍本课题研究使用到的图拓扑结构属性[5]，包括：度序列、节点效率、聚类系数、平均聚类系数、局部效率、全局效率、模块化系数、平均路径长度。本文采用 Python 语言进行编码，利用 Networkx 开源库以及 Numpy 开源库进行网络计算，计算过程调用的 Networkx 库以及 Numpy 库接口如表 3.1 所示。

度序列，在图结构数据中，图中与某个节点连接的边的数量就是该节点的度。无向图中，节点的度组成的非递增序列就是该图的度序列。度数序列是图不变量，因此同构图具有相同的度数序列。但是，度数序列通常不能唯一地标识一个图。在某些情况下，非同构图具有相同的度数序列。在本文研究中，我们将度序列在顺序上进行了一些调整，以此来适配我们的节点序列。本文中，把无向图 $V(G) = \{v_1, v_2, v_3, \dots, v_n\}$ 相应的各个节点的度组成一个序列，$\{d(v)_1, d(v)_2, d(v)_3, \dots, d(v)_n\}$ 作为图的度序列。

节点效率，节点效率度量的是给定节点 i 和网络中所有其他节点之间的平均最小路径

权重，具体计算公式见(3.1)。节点效率可以通过计算与所有其他节点之间最小路径长度的乘性倒数的平均值，能够有效反映节点在网络中的相对位置，体现一个节点在网络中的重要性和影响力。

$$E_{nodal}(i) = \frac{1}{N-1} \cdot \sum_{i \neq j \in G} \frac{1}{w_{ij}} \tag{3.1}$$

聚类系数，也称群聚系数，反映的是图中节点存在聚集性倾向的程度。现有的部分研究表明，在大多数现实世界的关系网络之中，尤其是社交网络中，通常都会存在很多紧密联系的节点群体，它们由若干个节点组成，其特点是关系密度相对较高。图中每个节点附近的集聚程度就是聚类系数所测量的值。在无权重图中，通过某个节点可能存在闭三点组[②]（邻近三点组）的分数就是该节点的聚类系数。计算公式如下：

$$c_u = \frac{2 \cdot T(u)}{d(u) \cdot (d(u)-1)} \tag{3.2}$$

其中，$T(u)$表示通过节点 u 的闭三点组数量，$d(u)$代表节点 u 的度数。

平均聚类系数，聚类系数的最小粒度是节点，因此聚类系数是有所有节点聚类系数值组成的一个向量。聚类系数向量中所有元素的平均值就是平均聚类系数，平均聚类系数反映的是一个图中整体的集聚程度的评估。

全局效率，全局效率是指网络中各节点间的平均特征路径长度的倒数，它是一种度量网络中长距离信息传输效率的度量方法。从表示某一结点的各体元与任意两个体元的时间序列的相似度出发，为有源函数的存在奠定了基础，并通过计算由 i 到其它各节点所需要的最小步数。该算法对网络中的各个结点进行了单独的处理，并对各个结点进行了单独的计算。然后对各节点的平均最小步数的倒数进行计算，并对该总的总连接进行归一化。形式上看，全局效率计算公式如下：

$$E_{glob}(G) = \frac{1}{N(N-1)} \cdot \sum_{i \neq j \in G} \frac{1}{w_{ij}} \tag{3.3}$$

局部效率，局部效率是一个范围从 0 到 1 的标度度量，值为 1 表示网络中的最大局部效率。在功能性大脑网络中，高局部效率表明拓扑组织反映分离的神经处理。网络的局部效率揭示了当节点 i 从网络中移除时，信息在节点 i 的第一个邻居之间传递的效率如何。具有高局部效率的网络中的节点倾向于在其直接的局部社区内有效地共享信息，这为网络

[②] 闭三点组：假设图中有一部分点是两两相连的，那么可以找出很多个“三角形”，其对应的三点两两相连，称为闭三点组。

中有效的隔离信息处理提供了基础。局部效率计算公式如下：

$$E_{loc}(G) = \frac{1}{N}\sum_{i\in G} E_{glob}(G_i) \tag{3.4}$$

模块化系数，模块化是衡量一个网络或图的结构性指标，它测量了一个单元的强度，也称为组，集群或社区。高模块化的网络在各模组内部的结点间紧密相连，而各模组间的结点间则是稀疏的连接。模块化系数反映了模块中节点的集中程度，而不是所有模块之间的随机分布。模块化系数计算公式如下：

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \gamma\left(\frac{d(i)d(j)}{2m}\right)\right)\delta(c_i, c_j) \tag{3.5}$$

其中，m 代表图中边的数量，$\boldsymbol{A}$ 代表图的邻接矩阵，$d(i)$代表节点 i 的度数，γ代表分辨率参数，当节点 i，j 在一个社区之中时$\delta(c_i, c_j)$=1，否则为 0。

平均路径长度，网络的平均路径长度 L 是指任意两个节点之间的平均距离，网络的平均路径长度也称为网络的特性路径长度。网络的加权特征路径长度 L 是连接网络中任意两个节点的平均最小连接权重。在本文中，为了避免节点断开的问题，通过使用任何节点对之间的“调和平均”距离（例如倒数的平均值的倒数）来测量 L。

$$L = \frac{N\cdot(N-1)}{\sum_{i\neq j}\frac{1}{w_{ij}}} \tag{3.6}$$

3.3.2 拓扑特征融合

本文研究的一大重点就是在图分类任务中充分利用图拓扑结构信息，因此，在图数据输入到网络之前，我们需要进行处理，将图拓扑结构信息融合到图节点属性之中。本文采用的作法是简单的拼接操作，即将拓扑结构信息拼接到图节点属性向量之后，作为图节点属性的一部分，最后输入到网络之中。

$$\widehat{\boldsymbol{h}}_{\boldsymbol{v}}^{\boldsymbol{0}} = \boldsymbol{h}_{\boldsymbol{v}}^{\boldsymbol{0}} + Topology(v) \tag{3.7}$$

其中，$\boldsymbol{h}_{\boldsymbol{v}}^{\boldsymbol{0}}$ 代表节点 v 的初始节点属性，$Topology(v)$ 代表节点 v 的图拓扑结构特征，$\widehat{\boldsymbol{h}}_{\boldsymbol{v}}^{\boldsymbol{0}}$ 作为最后输入到网络中的节点属性。

3.3.3 GNNs 子网络

在 GNNs 子网络中，我们考虑图的预测性质，即对于每个图 $G = (V, E)$，我们都有一个特定的预测目标 y_G。我们假设每一个节点 $v \in V$ 都有一个初始特征向量 $\boldsymbol{h}_{\boldsymbol{v}}^{\boldsymbol{0}} \in \mathbb{R}^{d_0}$。由于 GraphTrans 是一个普遍适用的模型框架，可以与各种 GNNs 协同使用，因此我们很少对

提供给 Transformer 子网络的 GNNs 层进行假设。一个通用的 GNNs 层堆栈可以表示为：

$$\boldsymbol{h}_v^l = f_l\left(\boldsymbol{h}_v^{l-1}, \left\{\boldsymbol{h}_u^{l-1} \mid u \in N(v)\right\}\right), \quad l = 1,2,\ldots,L_{GNN} \tag{3.8}$$

其中，L_{GNN}代表 GNNs 层堆栈的数量，$N(v) \subseteq V$代表节点 v 的邻接节点集，同时 $f_l(\cdot)$是由神经网络参数化的函数，许多的 GNNs 层都承认边缘特性，但是为了避免符号混乱我们这里省略了对它们的讨论。

上述公式说明，在 GNNs 计算中，每一个节点 v，在第 l 层 GNNs 网络中的表示都由第 l-1 层的节点 v 表示和其邻居节点在第 l-1 层的表示共同决定。

在本文的 GraphTrans 方法中，选取了 GCN 网络作为 GNNs 层的模型框架。GCN 与 CNN 的作用基本一致，都是一个特征提取器，但是 CNN 更多用于文本、序列等数据，而 GCN 的数据对象是图。

GCN 的核心部分就是特征矩阵和邻接矩阵。假设现有一批图数据，其中共计 N 个节点，并且每个节点都具有自己的特征向量，于是这些向量就可以组成一个$N \times D$的矩阵 X（特征矩阵）。同时，各个节点之间的连接关系也会形成一个$N \times N$ 的矩阵 A（邻接矩阵）。X 和 A就是 GCN 模型的输入。

GCN 作为一个神经网络层，它的层与层之间的传播方式如下：

$$H^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l W^l) \tag{3.9}$$

其中，$\tilde{A} = A + I$，I 是单位矩阵。$\tilde{D}$ 是 $\tilde{A}$ 的度矩阵。H 是每一层的特征，$H^0 = X$。σ 是非线性激活函数。

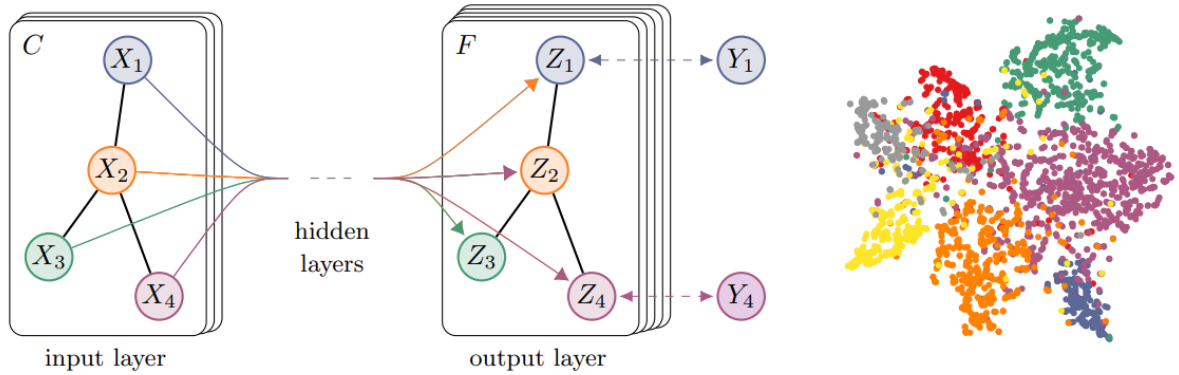

图 3-3 GCN 模型框架图[6]

如图 3-3，是 GCN 论文[6]中所展示的 GCN 网络模型与隐藏层激活图。这幅图可以直观

理解网络在训练过程中的变化，图中的 GCN 中，输入一个图，通过若干层 GCN 每个节点的特征从 X 变成了 Z，但是，无论中间有多少层，节点之间的连接关系，即 A，都是共享的。实际上，图卷积是利用其他结点的信息来推导该结点的信息。在半监督学习中，图卷积本质不是传播标签，而是在传播特征，图卷积将不知道标签的特征，传染到已知标签的特征节点上，利用已知标签节点的分类器推测其属性。另外，图中的每个结点无时无刻不因为邻居和更远的点的影响而在改变着自己的状态直到最终的平衡，关系越亲近的邻居影响越大。基于上述的 GCN 特性，所以在本文 GraphTrans 的 GNNs 层模型选取时，我们选择了 GCN 网络来作为基础模型。

3.3.4 Transformer 子网络

通过 GNNs 子网络，我们获得了每个节点在 GNNs 层中的最终表示 $\boldsymbol{h_v^{L_{GNN}}}$，再将这个表示传递给 Transformer 子网络。Transformer 子网络的操作如下。

首先，通过对 $\boldsymbol{h_v^{L_{GNN}}}$ 进行一个线性变换来将向量维度转换为 Transformer 子网络的维度，再对新向量进行一个层归一化。计算公式如下：

$$\bar{\boldsymbol{h}}_{\boldsymbol{v}}^{\boldsymbol{0}} = LayerNorm\left(W^{Proj}\boldsymbol{h_v^{L_{GNN}}}\right) \tag{3.10}$$

其中，$W^{Proj} \in \mathbb{R}^{d_{TF} \times d_{\boldsymbol{L_{GNN}}}}$ 是一个可学习的权重矩阵，d_{TF} 代表 Transformer 子网络的维度，$\boldsymbol{L_{GNN}}$ 是最后一个 GNNs 层的维度。

第二步，将 $\bar{\boldsymbol{h}}_{\boldsymbol{v}}^{\boldsymbol{0}}$ 输入到标准的 Transformer 堆栈之中，通过以下计算可以得到最终的结果。

$$\mathfrak{a}_{v,u}^{l} = \left(W_l^Q \bar{h}_v^{l-1}\right)^T \left(W_l^K \bar{h}_u^{l-1}\right) / \sqrt{d_{TF}} \tag{3.11}$$

$$\alpha_{v,u}^{l} = softmax\left(\mathfrak{a}_{v,w}^{l}\right),\ w \in V \tag{3.12}$$

$$\bar{\boldsymbol{h}}_{\boldsymbol{v}}^{\boldsymbol{\prime l}} = \sum_{w \in V} \alpha_{v,w}^{l} W_l^V \bar{\boldsymbol{h}}_{\boldsymbol{w}}^{\boldsymbol{l-1}} \tag{3.13}$$

其中，W_l^Q，W_l^K，$W_l^V \in \mathbb{R}^{d_{TF}/n_{head} \times d_{TF}/n_{head}}$ 是单个注意力头在第 l 层时的可学习的查询、键和值的矩阵。

与标准 Transformer 模型一样，我们执行 n_{head} 个并行的注意力头，并且把每一个注意力头的 $\bar{\boldsymbol{h}}_{\boldsymbol{v}}^{\boldsymbol{\prime l}}$ 输出拼接在一起，然后再将拼接的编码向量输入到 Transformer 子网络中的全连接子网络中。

3.3.5 <CLS> 嵌入

本文中使用 <CLS> 嵌入作为 GNNs 的“读出”函数。对于全图分类，我们需要一个描述整个图的嵌入向量。 在 GNN s 文献中，这个将每个节点和/或边缘的嵌入折叠为单个嵌入的模块称为“读出”模块，最常见的读出模块是简单的均值或最大池化，或单个“虚拟节点”连接到网络中的每个其他节点。

在本文研究中，提出了一个类似于 Transformers 其他应用中使用的特殊 token 读出模块。在使用 Transformers 的文本分类任务中，一种常见的做法是在输入序列中附加一个特殊的 <CLS> 标记，然后再将其传递到网络中，然后将与该标记位置对应的输出嵌入作为整个句子的表示。 通过这种方式，Transformer 将被训练以将句子的信息聚合到该嵌入，通过使用注意模块计算 <CLS> 标记与句子中的每个其他标记之间的一对一关系。

本文中使用的特征 token“读出”模块与这类似。具体来说，将每个节点的节点表示 $\overline{\boldsymbol{h}}_{\boldsymbol{v}}^{\boldsymbol{0}}$ 输入到 Transformer 子网络之前，在序列中加入一个可学习的嵌入向量 $h_{<CLS>}$，并且取 Transformer 子网络的第一个嵌入输出 $\overline{\boldsymbol{h}}_{\boldsymbol{<CLS>}}$ 作为全图的表示。最终对该全图表示进行一个线性变换以及 softmax 操作，得到预测结果。计算公式如下：

$$y = softmax(\boldsymbol{W}^{\boldsymbol{out}}\overline{\boldsymbol{h}}_{\boldsymbol{<CLS>}}^{\boldsymbol{L_{TF}}}) \tag{3.14}$$

其中，$\boldsymbol{W}^{\boldsymbol{out}}$表示输出的权重矩阵，$\boldsymbol{L_{TF}}$表示 Transformer 子网络的层数。

3.4 本章小结

本章主要内容是设计实现一种基于 Transformer 的图分类方法，旨在充分利用图的节点属性和拓扑结构信息，在图分类任务上获得良好的性能。本章首先介绍研究动机，表明设计 GraphTrans 的背景、意义以及创新点，进而给出 GraphTrans 的模型框架图以及讲述 GraphTrans 的具体实现细节（拓扑计算、特征融合、GNNs 子网以及 Transformer 子网等）。

第四章 实验设计与结果

本章主要讲述本文研究的实验设计与结果，具体包括数据集介绍、数据预处理、评价指标、对比方法选取、实验设计、实验结果以及参数实验等七个部分。

4.1 数据集介绍

本文研究的实验数据集采用了目前公开的六个数据集，分别是：NCI1、IMDB-BINARY、PTC_FR、MUTAG、IMDB-MULTI 以及 Mutagenicity。下表给出六个数据集的基本信息。

表 4.1 实验数据集基本信息表

数据集	所属领域	图的数量	类别数量	平均节点数量	平均边数量
NCI1	生物化学	4110	2	29.87	32.30
IMDB-BINARY	社交网络	1000	2	19.77	96.53
PTC_FR	生物化学	351	2	14.56	15.00
MUTAG	生物化学	188	2	17.93	19.79
IMDB-MULTI	社交网络	1500	3	13.00	65.94
Mutagenicity	生物化学	4337	2	30.32	30.77

4.2 数据预处理

本文研究目标是利用图的拓扑结构特征信息进行分类，具体的模型设计是将图的拓扑结构特征信息融合到图的节点属性之中。因此，在进行实验之前必须对数据集中的数据进行预处理，计算出图的拓扑结构信息并记录保存，以便实验时能够融入图的节点属性特征中。

4.3 评价指标

评价指标是针对将相同的数据，输入不同的算法模型，或者输入不同参数的同一种算法模型，而给出这个算法或者参数好坏的定量指标。机器学习常用的评价指标有精度、精确率、召回率、P-R 曲线、F1 Score、TPR、FPR、ROC、AUC（Area Under Curve）等指标，还有在生物领域常用的灵敏度、特异度等指标。本小节主要介绍混淆矩阵的概念以及本文中使用的三个评价指标准确率、F1 Score 以及 AUC（Area Under Curve）。

4.3.1 混淆矩阵

首先给出混淆矩阵的概念，它是一种可视化的方法，它可以将分类结果与实际情况进行对比。在该矩阵中，每个行表示一个实例的预测分类，每个列表示一个实际的分类。如下表所示：

表 4.2 混淆矩阵

实际标签	预测标签	
	正例	反例
正例	TP	FN
反例	FP	TN

注：

真正(True Positive , TP)：被模型预测为正的正样本。

假正(False Positive , FP)：被模型预测为正的负样本。

假负(False Negative , FN)：被模型预测为负的正样本。

真负(True Negative , TN)：被模型预测为负的负样本。

4.3.2 准确率

准确率是分类问题中最基础、最常用的评价度量指标，准确率指的是预测正确的结果占总样本的百分比，其计算公式如下：

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.1}$$

但是，这种方法也有一个缺点，那就是当数据类型不平衡时，尤其是有很大的偏差时，这种方法并不能很好地反映出算法的好坏。举个例子：在一个数据集中，有 100 个样本，99 个反例，1 个正例。假设模型总是预测样本为反例，则该模型的准确率为 0.99，从数值上来看很好，但实际上，这种方法并没有什么预见性，所以我们要考虑的是不是评估的标准出现了问题，因此需要再结合其它的评估方法来进行综合判断。

4.3.3 F1 Score

F1 Score，是统计学中用来测量二分类（或多任务二分类）模型精确度的一种指标。分类模型的准确率和召回率都被它考虑在内。模型准确率和召回率的一种特殊加权平均就是 F1 Score，它的最小值是 0，最大值是 1，通常来说值越小模型越差、值越大模型越好。

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.2}$$

$$precision = \frac{TP}{TP+FP} \tag{4.3}$$

$$recall = \frac{TP}{TP+FN} \tag{4.4}$$

4.3.4 AUC（Area Under Curve）

AUC（Area Under Curve），ROC 曲线下与坐标轴围成的面积即是 AUC 的值，由此可见这个面积的数值一定是小于 1.0。再因为 ROC 曲线通常来说都位于 y=x 这条直线的上方，所以 AUC 更加准确的取值范围应该是 0.5 到 1.0。AUC 越接近 1.0，检测方法真实性越高; 等于 0.5 时，则真实性最低，基本无应用价值。

目前大多数研究都采用 AUC 值作为模型的评价标准，原因是 ROC 曲线不是数值，不能直观地、清晰地反映分类器性能的优劣。但是 AUC 是一个数值类型的指标，能够直观、清晰地反映分类器的优劣，AUC 值越高则分类器的性能越好。

表 4.3 GraphTransformer 实验参数表

参数名称	参数值
batch_size	128
beta	1.0
dim_hidden	64
dropout	0.3
epochs	100
lap_dim	8
lr	0.001
nb_heads	8
nb_layer	7
warmup	2000
weight_decay	1e-05

4.4 对比方法选取

为了验证本文提出的 GraphTrans 模型的有效性，本文选取了四个对比方法进行对比实

验，分别是 GCN（Graph Convolutional Network）[6]、GAT（Graph Attention Network）[9]、GIN（Graph Isomorphism Network）[10]以及 GraphSAGE[29]。

4.5 实验设计

本文的实验将在 4.1 节中所述的五个数据集上展开，选取的基线对比方法为 GCN、GAT、GIN 以及 GraphSAGE 方法，采用一次十折交叉验证的方式进行实验（划分测试集、验证集、测试集为 8：1：1），实验的评估指标为（Accuracy，F1 Score，AUC），分别代表预测准确率，F1 得分和 AUC 数值。对于本文提出的模型 GraphTransformer（GraphTrans）的具体参数如表 4.3 所示。

4.6 实验结果

通过在六个实验数据集上进行了基线实验与 GraphTrans 实验，我们得到了以下的结果，如表 4.5 至表 4.7 所示。其中 IMDB-M 数据集是多分类任务，不存在 F1 Score 与 AUC 指标，所以 F1 Score 与 AUC 表中不记录该数据集。

从结果来看，在 NCI1 数据集上，本文的 GraphTrans 方法准确率仅比最好的 GIN 低了 1.83%。在 IMDB-B 数据集上，GraphTrans 方法取得了最好的结果，高出 GAT 方法 0.50%。在 PTC_FR 数据集上，GraphTrans 也取得了最好的结果，高出 GAT 方法 2.32%。在 MUTAG 数据集上，GraphTrans 再次取得最好结果，高出 GIN 方法 2.50%。在 IMDB-M 数据集上，GraphTrans 方法比最好的 GraphSAGE 低了 2.33%。在 Mutagenicity 数据集上，GraphTrans 也取得了最好的结果，高出 GIN 方法 0.26%。

再对比几个取得过最好结果的方法在不同数据集上的稳定性。GIN 方法在 NCI1 数据集上取得了最好的结果，但是在 IMDB-B 和 IMDB-M 两个数据集上的表现极差，具有太强的不稳定性。GraphSAGE 在 IMDB-M 数据集上取得了最好的结果，但是在 PTC_FR 和 MUTAG 两个数据集上的结果也很差，也具有不稳定性。而 GraphTrans 在 IMDB-B、PTC_FR、MUTAG 和 Mutagenicity 四个数据集上都取得了最好的结果，同时在 NCI1 和 IMDB-M 两个数据集上的结果也仅是小幅低于最优方法。

综合上述的分析，得出以下结论：GraphTrans 能够适应图分类任务，并且取得良好的表现；同时，GraphTrans 能够在不同数据集上具有良好的稳定性。

表 4.4 对比实验各方法 Accuracy 指标实验结果

实验方法	数据集					
	NCI1	**IMDB-B**	**PTC_FR**	**MUTAG**	**IMDB-M**	**Mutagenicity**
GCN	76.27±2.21	72.40±5.61	56.67±8.34	72.54±11.33	50.93±4.09	80.66±2.09
GAT	75.35±1.55	72.90±4.72	58.39±8.84	72.68±5.76	50.00±4.26	80.35±1.74
GIN	**78.54±1.55**	62.00±5.04	57.28±10.70	78.29±11.85	36.40±2.44	81.30±2.14
GraphSAGE	77.76±2.20	72.20±4.66	55.55±8.33	70.41±7.23	**51.73±3.52**	80.45±1.73
GraphTrans	76.71±1.98	**73.40±4.13**	**60.71±4.92**	**80.79±4.34**	49.40±4.24	**81.56±2.17**

表 4.5 对比实验各方法 F1 Score 指标实验结果

实验方法	数据集				
	NCI1	**IMDB-B**	**PTC_FR**	**MUTAG**	**Mutagenicity**
GCN	75.51±1.53	73.06±5.54	21.50±17.67	79.62±11.71	77.99±2.64
GAT	75.82±1.79	73.66±4.55	22.79±21.55	82.70±2.39	77.49±2.13
GIN	**77.95±1.97**	68.08±6.44	**30.74±19.04**	83.57±9.23	79.09±2.39
GraphSAGE	76.90±2.45	71.89±6.01	30.39±18.47	81.51±3.77	78.25±1.69
GraphTrans	76.61±1.45	**73.75±3.96**	21.87±11.16	**85.42±3.23**	**79.09±2.16**

表 4.6 对比实验各方法 AUC 指标实验结果

实验方法	数据集				
	NCI1	**IMDB-B**	**PTC_FR**	**MUTAG**	**Mutagenicity**
GCN	76.36±2.21	72.40±5.61	49.04±4.96	65.90±15.72	80.37±2.23
GAT	75.39±1.58	72.90±4.72	51.14±6.19	60.25±11.42	79.95±1.75
GIN	**78.48±1.57**	62.00±5.04	**51.71±8.16**	75.49±13.32	81.17±2.12
GraphSAGE	77.73±2.25	72.20±4.66	50.76±6.72	57.50±12.47	80.34±1.63
GraphTrans	76.74±1.96	**73.40±4.13**	50.39±3.96	**79.01±6.22**	**81.30±2.02**

4.7 参数实验

在深度学习模型中，过少的模型层数会导致模型过于简单，无法具有良好表现；过多的模型层数又会导致模型过于复杂，出现过拟合现象。为了探究 Transformer 模型层数对模型预测结果的影响，本小节将针对 Transformer 模型层数进行参数实验。

参数实验在 NCI1 数据集和 IMDB-B 数据集上进行，Transformer 模型层数选择从 1 到 8。其余实验环境与对比试验保持一致。

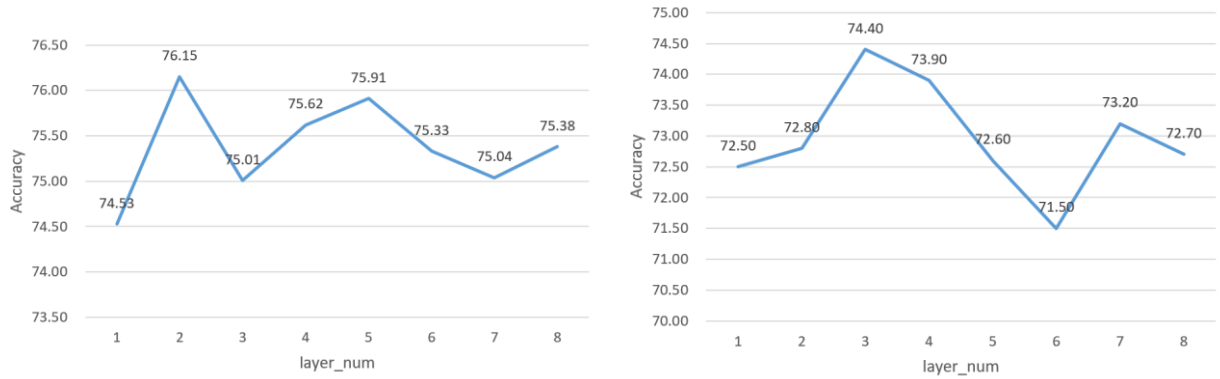

图 4-1 模型准确率随模型层数变化曲线(NCI1 左、IMDB-B 右)

实验结果展示模型预测准确率随 Transformer 模型层数变化的曲线。如图 4-1 所示。

从图中可以看出，当 Transformer 模型层数由 1-2 层时，模型性能随层数增加而增强；在 2 或 3 层时，模型在三个数据集上都具有局部最优的表现；往后随层数增加，模型性能有所下降；但在 5、6 或 7 层时，模型性能出现一个突然增强的表现，也存在一个局部较优的表现。当模型层次继续增加时，性能没有呈现规律性。综合来看，模型层次 2 或 3 是一个最优值，模型能够具有较为良好的表现。

4.8 本章小结

本章主要讲述本文研究的实验设计与结果。首先介绍了实验选取的数据集的基本情况（所属领域、图数量以及类别数量等），再而介绍了数据预处理过程（采用 Networkx 库包进行图拓扑特征计算），其次介绍了本文实验采取的评价指标（准确率、F1 Scorer 以及 AUC），同时介绍了对比方法的选取来源，之后给出了本文的实验设计部分，最后给出本文的实验结果以及参数实验结果，并且进行了结果分析。

通过本章的实验结果来看，本文提出的 GraphTrans 模型能够适应图分类任务，并且取得良好的表现；同时，GraphTrans 能够在不同数据集上具有良好的稳定性。

第五章 总结与展望

5.1 工作总结

本文通过调研图分类任务的目前研究现状，发现关于图拓扑结构特征信息使用的相关研究较少，但是图拓扑特征信息又是图结构中很重要的一个信息，因此本文认为研究如何充分利用图拓扑结构信息来提升图分类任务的性能具有很大的实际意义。同时，在目前主流的 GNNs 网络模型中，大部分模型都难以很好学习到图中长距离成对节点的依赖信息，堆叠 GNNs 层的做法或领域粗化方法并不可取，加之目前 Transformer 模型在图分类任务上有了越来越多的尝试，部分研究也确实取得了较为良好的表现。为此，本文的主要研究内容是设计一个基于 Transformer 的图分类模型，期望能够在图分类任务中融入图的拓扑结构信息，并且通过 Transformer 架构来学习图中所谓的长距离依赖信息。

我们将首先计算图的拓扑结构信息，然后将拓扑结构编码与节点属性结合，一起输入到深度学习网络模型之中，旨在充分利用图的节点属性和拓扑结构信息建立图分类模型，更好地完成图分类任务。我们采用一种不同的方法来处理图池化和学习 GNNs 中的长距离依赖信息。与层次池化一样，我们的方法也受到了计算机视觉的启发，我们采用纯粹的学习操作（注意力）来代替显示编码相关关系归纳偏差的一些原子操作（CNN 中的卷积或者空间池化，GNNs 中的领域粗化）。本文的研究方法称之为 GraphTransformer，在编码层面，通过计算图的拓扑结构信息，并将其融入图的节点属性，输入到深度学习网络之中。在网络结构层面，将采用 GNNs 与 Transformer 架构融合的思想，使用 GNNs 学习节点近邻域的结构本地表示，同时利用 Transformer 架构作为强大的全局推理模块，学习长距离依赖信息。

本文调研了很多网络分析领域的研究，同时考虑了计算的可行性、时效性等，最终确定了度序列、节点效率、聚类系数、平均聚类系数、局部效率、全局效率、模块化系数和平均路径长度等八个拓扑结构指标作为研究对象。

本文实验设计选取了多个图分类公开数据集，包括二分类和多分类的多个数据集作为实验数据集，同时选取了主流的 GCN、GAT、GIN 和 GraphSAGE 方法作为实验的基线方法，用准确率、F1 Score 和 AUC 作为评价指标，在相同环境下进行对比实验。

本文的实验结果表明 GraphTransformer 模型能够在大部分数据集中取得良好的表现，同时该模型在不同数据集上也具有很好的稳定性。

通过本文研究，基本可以确定 Transformer 模型在图分类任务上的可行性、有效性。同时也证明了拓扑结构特征在图分类任务中的重要性，能够有效提升图分类的性能。

5.2 工作展望

Transformer 模型目前以及在 CV、NLP 等领域成为了主导方法，大部分的任务中，Transformer 模型都做到了最优的表现，通过本文研究，以及一些 Graph Transformer 的相关研究，我们可以发现，Transformer 模型似乎也可以在图分类任务上取得很好的表现。不过目前主流的图分类任务排行榜中，Transformer 都还没有取得较好的成绩，所以在这方面的研究还有待进一步探索。

最后也是期望本文的研究成果能够对这一研究方向有一定的启发与指向性，当然本文也存在一些工作的不足，期待该领域的研究者们能指出不足，并且加以完善。

参考文献

[1] 刘勇, 李建中, 朱敬华. 一种新的基于频繁闭显露模式的图分类方法[J]. 计算机研究与发展, 2007, 44(7): 1169.

[2] 詹卫许, 王桂娟. 基于频繁子图的图分类方法研究[J]. 现代计算机: 下半月版, 2010 (3): 48-52.

[3] 项英倬, 谭菊仙, 韩杰思,等. 图匹配技术研究[J]. 计算机科学, 2018, 45(6): 27-31, 45.

[4] 王兆慧, 沈华伟, 曹婧,等. 图分类研究综述[J]. Journal of Software, 2021, 33(1): 171-192.

[5] Chen Z, Liu M, Gross D W, et al. Graph theoretical analysis of developmental patterns of the white matter network[J]. Frontiers in human neuroscience, 2013, 7: 716.

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

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

[8] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[9] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[10] Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks?[J]. arXiv preprint arXiv:1810.00826, 2018.

[11] Cai H, Zheng V W, Chang K C C. A comprehensive survey of graph embedding: Problems, techniques, and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9): 1616-1637.

[12] Yun S, Jeong M, Kim R, et al. Graph transformer networks[J]. Advances in neural information processing systems, 2019, 32.

[13] Dwivedi V P, Bresson X. A generalization of transformer networks to graphs[J]. arXiv preprint arXiv:2012.09699, 2020.

[14] Kriege N M, Johansson F D, Morris C. A survey on graph kernels[J]. Applied Network Science, 2020, 5(1): 1-42.

[15] Song C, Lin Y, Guo S, et al. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(01): 914-921.

[16] 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.

[17] Ying C, Cai T, Luo S, et al. Do Transformers Really Perform Badly for Graph Representation? [J]. Advances in Neural Information Processing Systems, 2021, 34.

[18] Mialon G, Chen D, Selosse M, et al. Graphit: Encoding graph structure in transformers[J]. arXiv preprint arXiv:2106.05667, 2021.

[19] 吴博, 梁循, 张树森,等. 图神经网络前沿进展与应用[J]. 计算机学报, 2022, 45(1).

[20] 翟璨. 基于图核的化合物分类研究[D]. 北京工业大学, 2015.

[21] 陈林君. 基于核方法的节点分类研究[D]. 广西师范大学, 2021.

[22] 梁演锋. 基于图神经网络的图分类理论与方法研究[D]. 西安电子科技大学, 2021.

[23] Xu B, Shen H, Cao Q, et al. Graph wavelet neural network[J]. arXiv preprint arXiv:1904.07785, 2019.

[24] Xu B, Shen H, Cao Q, et al. Graph convolutional networks using heat kernel for semi-supervise d learning[J]. arXiv preprint arXiv:2007.16002, 2020.

[25] Wu J, He J, Xu J. Net: Degree-specific graph neural networks for node and graph classification [C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 406-415.

[26] Zhang M, Cui Z, Neumann M, et al. An end-to-end deep learning architecture for graph classific ation[C]//Thirty-second AAAI conference on artificial intelligence. 2018.

[27] Cen K, Shen H, Gao J, et al. ANAE: Learning Node Context Representation for Attributed Netw ork Embedding[J]. arXiv preprint arXiv:1906.08745, 2019.

[28] Navarin N, Van Tran D, Sperduti A. Universal readout for graph convolutional neural networks[C] //2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019: 1-7.

[29] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems, 2017, 30.

致　　谢

日月如梭，光阴似箭。在东南大学的四年本科生活转瞬即逝。我已经从一个懵懂无知的少年变成了略显成熟的青年。过往四年大学生活带给了我数不尽的收获，在此我想向所有帮助过我的人表达谢意。

首先，我要感谢我的指导老师孔佑勇老师，从选题、开题、中期答辩到如今的论文撰写，孔老师都给予了我很多的帮助。在学习上，孔老师凭借自己认真负责的态度，为我提供了很多建议和思考，为我指明了学习研究的方向。在生活上，孔老师也十分关心我，疫情期间给予了很多关怀，包括为我提供了一些的食物保证等。再次感谢孔老师。

同时，我要感谢东南大学计算机、软件学院教授过我课堂知识的多位老师，感谢你们在学习生活上的指导。感谢你们带我进入了计算机科学领域，教会了我很多知识，是这些知识能够支撑我完成毕业设计以及继续未来的科研工作。

其次，我要感谢李嘉兴、张可两位研究生学长，感谢你们俩在我课题推进、基础学习以及实验研究上的指导，为我解答了很多疑惑，让我的课题能够顺利推进。

我要感谢 711184 班的每一位同学，感谢你们在生活、学习上对我的照顾，能够与你们做同学，我感到十分的荣幸。

我还要感谢我的女朋友夏苑馨同学，是你给予了我很大的精神动力。我们在不同的环境中独自完成着学业，但又彼此相伴，感谢你对我暴脾气的容忍，感谢你伴我走过四年零七个月。初见乍惊欢，久处亦怦然。

最后，我要感谢我的家人，感谢父母多年的辛劳付出，感谢两位姐姐的关心照顾。是你们的支持，我才得以完成如今的学业。望父母长乐久安，望姐姐生活美满。