# 图神经网络

## 简介

图（Graph）描述了实体之间的两两关系，在日常生活中无处不在，许多重要的现实世界数据集以图的形式出现，例如: 社交网络、知识图谱、蛋白质结构、互联网等等。除此之外，如图15-1所示，许多其他类型的数据也可以用图的形式进行展现。在现实生活中，社交媒体、电子商务、生物信息学等领域产生了大量的图数据，许多任务也需要通过图上的计算任务解决，例如，在社交网络分析中，如何识别社交网络中的社区结构、关键节点和信息传播路径等；在化学和生物信息学中，如何进行分子结构预测和蛋白质功能预测，从而加速新药物的设计和开发过程，他们在本质上都是图上的计算问题。

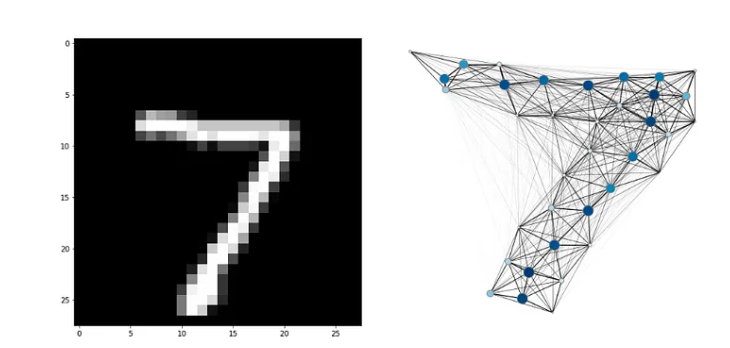


图 ‑1 使用图结构展现图片

然而，过往传统的机器学习算法通常将数据表示为向量或矩阵的形式，而忽略了不同数据点之间的关系和连接，因此无法有效地处理图这种非欧几里德的结构数据。十多年来，研究人员开发出了能够处理图结构数据的图神经网络，它能够捕捉数据之间的拓扑结构关系，将图数据和非欧几里德结构数据纳入到深度学习范畴中，使得深度学习具有更广泛的适用性和更强的表达能力。

在早期，人们主要使用针对图像和文本数据的传统机器学习算法来处理图形数据，如支持向量机、随机森林等。在2005 年，Marco Gori 首次提出图神经网络（Graph Neural Network， GNN）[1]，将图结构的相关学习过程直接架构在图数据之上，从而避免图结构信息的损失。之后在 2013 年Bruna 等人[2]利用谱图理论中已知的图卷积从而将卷积方法引入到了图神经网络中，但是基于谱域图卷积的方法需要很高的时间复杂度。2016 年，Defferrard 等人[3]利用类神经网络模型中学习到的具有自由参数的切比雪夫多项式，在谱域中逼近平滑滤波器。他们在一些常规领域中（例如MNIST数据集）取得了令人信服的结果，与简单的2D-CNN 模型非常接近。而Kipf 等人[4]则将谱域图卷积进行简化，使得图卷积操作可以在空域进行，极大的提高了计算效率，在许多基准图数据集上达到当时最先进的分类结果。2018年，Velickovic等人[5]提出了图注意力网络（Graph Attention Network，GAT），利用注意力机制来控制图中节点之间信息的传递，能够更好地处理图形数据中节点数量和连接数量不确定的情况，并取得了很好的效果。近年来，图生成模型（Graph Generative Network，GAN）也逐渐成为研究热点之一，通过对真实图性质的观察、分析和建模，提出一些图生成机制、模型和算法，在新药设计等应用中有着广泛的发展前景。

本章将对图神经网络中的重要概念和基础进行介绍。

## 图论基础

本章将介绍一些图论的基础知识，包括图的矩阵表示以及图上的重要度量和性质。此外，本章还将讨论一些基础的谱图理论和图信号处理中的概念，这是理解谱域图神经网络的重要基础。

### 图的概述

图是由顶点和边（或弧）组成的一种数据结构，其中节点表示对象，边表示对象之间的关系。图广泛应用于计算机科学、网络分析、社会科学等领域。根据边是否有向可以将图分为有向图和无向图，根据边是否带权可以将图分为带权图和无权图。

1. 图（Graph）

如图15-2所示，一个图可以被表示为顶点和边的集合，记作为：，其中是元素数量为的节点集合，是元素数量为的边集合。边描述了节点之间的联通关系，一条连接了节点的边也可以被记作，在有向图中，表达的前后顺序代表边的指向（即由指向），在无向图中，顺序则没有任何含义。

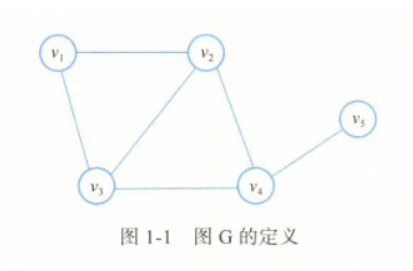


图 ‑2 图G的定义

1. 邻接矩阵与关联矩阵

在计算机中，图有很多种表示方法，每种表示都有不同的用处，需要根据实际需求选择。比较标准和常规的表示方法有：邻接矩阵和关联矩阵。这两种表示法都即可以表示无向图，也可以表示有向图。

图的邻接矩阵表示方法：首先将图中的节点按照一定顺序编号，然后使用一个矩阵来表示节点之间的连接关系。设图有个顶点，使用邻接矩阵表示该图时，首先定义一个的矩阵，如果存在一条节点到节点之间的边，则，否则。

对于无向图而言，邻接矩阵是一个对称矩阵，因为当存在从节点到节点的边时，同时也存在从节点 到节点的边。而对于有向图而言，邻接矩阵不一定是对称的。当仅在节点到节点之间存在一条有向边时，，但可以是或者，这取决于是否存在从节点到节点的有向边。对于图15-2所展示的图G，我们可以使用图15-3的邻接矩阵表示。

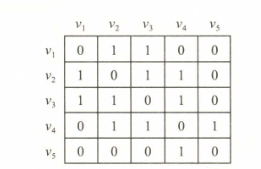


图 ‑3 图G的邻接矩阵表示

除了邻接矩阵外，我们有时也会使用关联矩阵来描述节点与边之间的关系，其中行和列分别对应于图中的节点和边。让我们设图有个顶点和条边，使用关联矩阵表示该图时，首先需要定义一个的矩阵，其中表示第个节点与第条边相连，则表示不相连。对于图15-2所展示的图G，我们可以使用图15-3的关联矩阵表示。



图 ‑4 图G的关联矩阵表示

1. 邻居和度

图的邻居是指与某个给定节点直接相连的所有其他节点。如果有一条边连接节点和，则节点和互为对方的邻居。我们记节点的所有邻居为集合，即：

度是一个给定节点与其相邻节点的数量，即该节点的度（Degree），记作：

对于无向图中的一个节点，它的度等于它与其他节点之间连接的总数。而在有向图中，则分为入度和出度，其中入度是指指向该节点的边的数量，出度是指从该节点出发的边的数量。一个节点的入度和出度可以帮助我们判断这个节点在图中起什么作用，比如入度很大的节点可能是其他节点向它汇聚流量的重要节点，而出度很大的节点则可能是流量源头。另外，在有向图中，所有节点的出度之和等于所有节点的入度之和。这是因为每一条带有方向的边都会同时造成起点的出度增加1，终点的入度增加1。

1. 子图和路径

一个图的子图（Subgraph）是指从该图中选择一些节点和它们之间的边所组成的图。设是一个图，其中是一组节点，是一组边。如果集合和满足以下条件：

1. 如果，即且，那么称是G的子图。其中称为的节点集，称为的边集。

如果存在一条由若干个节点和边按照顺序组成的序列，满足对于，有，那么称是从到的一条路径（Path）。其中，和分别称为路径的起点和终点；称为路径的长度或距离。

顶点距离：若从节点到节点存在一条路径，那么路径的长度称为节点 到节点 的距离，记为。特别地，定义节点到自身的距离为0，即，如果不存在从节点到节点的路径，则定义。

阶邻居：若，则称为的阶邻居，也称为k hops。

阶子图（k-subgraph）：的阶子图是指由中所有包含个或者更少节点的连通子图和其所包含的边组成的子图，可表示为：

如图15-5所示，阴影部分是顶点的2阶子图。

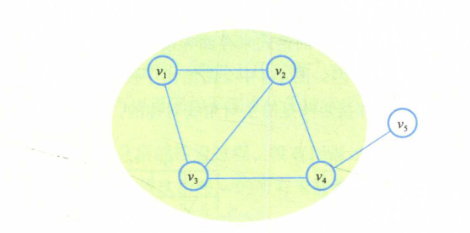


图 ‑5 图G的关联矩阵表示

### 图数据的应用场景

图是一种非常强大的和通用的数据表示方法，在现实生活中，图数据有着非常广泛的应用场景。下面让我们举几个例子进行说明，在这些示例中，每个节点的邻居数是可变的(与图像和文本的固定邻居大小相反)，这些数据很难用图以外的任何其他方式表达。

化学分子：分子是物质的组成部分，是由3D 空间中的原子和电子构成的。所有的粒子都是相互作用的，但是当一对原子相互之间保持一个稳定的距离时，我们说它们共享一个共价键。不同的原子对和键有不同的距离(例如单键、双键)。将这个3D 对象描述为一个图是非常方便和常见的抽象，其中原子表示为节点，共价键表示为边。

社交网络：社交网络是研究人们、机构和组织的集体行为模式的工具，是一类十分常见的图数据。我们可以通过将个体建模为节点，将他们之间的关系建模为边来构建代表人群的图形。通过对社交网络建模，可以帮助人们理解社交网络中的个体行为和整体特征，揭示社交网络中隐含的模式和规律。

引用网络：研究人员在发表论文时经常引用其他研究人员的研究成果。我们可以将这些引文网络可视化为一个图形，其中每篇论文是一个节点，每条有向边是一篇论文与另一篇论文之间的引文。此外，我们还可以向每个节点中添加关于每篇论文的信息，例如摘要的词嵌入。

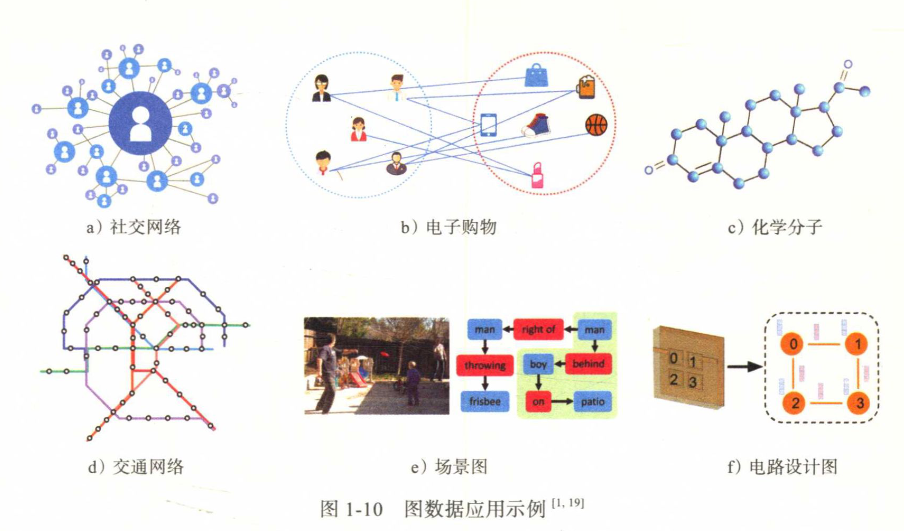


图 ‑6 生活中的图数据

虽然图计算有着很强的应用价值，但如何进行图计算充满了挑战，下面让我们看一下对图结构数据进行计算的复杂性：

1. 缺乏一致性结构：图是非常灵活的数学模型，但这也意味着它们缺乏跨实例的一致结构。现实世界图的结构在不同类型的数据之间可能有很大的不同——有些图有许多节点，它们之间的连接很少，反之亦然。根据节点的数量、边和节点的连通性，图数据集可以有很大的不同。就预测化学分子是否有毒来说：不同的分子可能有着不同数量的原子，每个原子可能有不同数量的连接，每个连接可能有不同的权重。由此可见，以一种可以计算的格式表示图结构是非常重要的，而且最终选择的表示通常在很大程度上取决于实际问题。

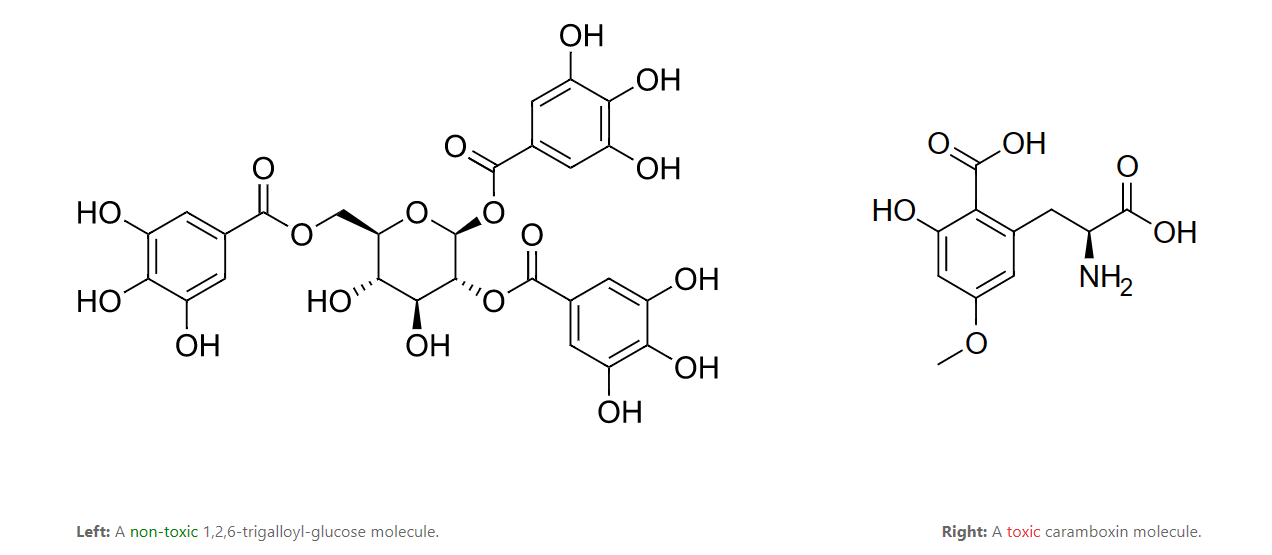


图 ‑7 化学分子结构

1. 节点顺序可变：图通常在节点之间没有固有的顺序。不同于图像其中像素在图像中的绝对位置唯一确定，当我们使用邻接矩阵表示图结构时，同一个图有多种表示方式。因此，我们希望我们的算法是节点置换不变的：它们不应该依赖于图中节点的顺序。如果我们以某种方式排列节点，那么由我们的算法计算的节点的结果表示也应该以同样的方式排列。

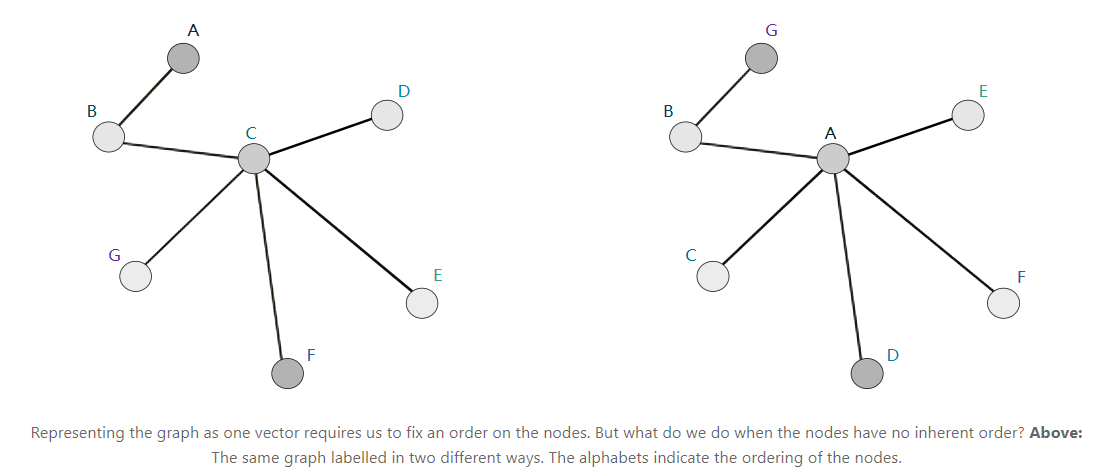


图 ‑8 同一图结构不同节点顺序

1. 图结构可以非常大，例如 Facebook 和 Twitter 这样的社交网络，它们拥有超过10亿的用户，在这样的图上做计算并不容易。不过幸运的是，大多数自然出现的图都是“稀疏”的: 它们的边数往往与顶点数呈线性关系，这意味着我们可以使用一些技巧来处理这些图，例如稀疏矩阵等。

## 图卷积神经网络

### 图信号与多项式滤波器

拉普拉斯矩阵（Laplacian Matrix）常用来研究图的结构性质，对于给定图G，其邻接矩阵为，我们可以构造关于图的对角矩阵，其对角线元素为对应节点的度，定义为：

图上的拉普拉斯矩阵为一个的方阵，定义为：

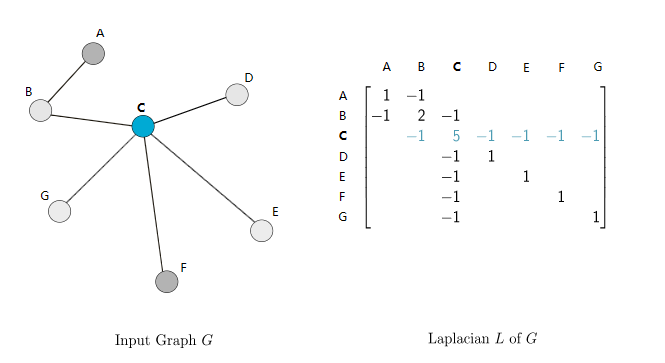


图 ‑9 拉普拉斯矩阵

另一种常用的拉普拉斯矩阵是正则化后的形式：。图的拉普拉斯算子的来源于微积分中拉普拉斯算子的离散模拟，可以用来描述中心节点和邻居节点之间的信号差异。

尽管拉普拉斯矩阵和邻接矩阵都是图的一种表示形式，在某种意义上，无论是给定邻接矩阵还是拉普拉斯矩阵，我们都可以构建出另外一个。但是拉普拉斯算子在许多涉及图的数学问题中都有所体现，如随机游动、 SVD 和扩散等等，相较邻接矩阵，拉普拉斯矩阵更适合用来研究图的结构性质。

在拉普拉斯矩阵的基础上我们可以建立拉普拉斯多项式：

其中，对于每一个w，是一个类似于的的矩阵。

这些多项式起到的作用可以被认为类似于 CNN 中的过滤器，其系数类似于过滤器的权重。为了便于说明，我们接下来将讨论图中每个节点的特征维数（图信号）为1维的情况，可以使用同样的思想将扩展到高维向量。按照邻接矩阵中的节点顺序，可以得到一个由节点信号或特征构成的矩阵，如图所示。

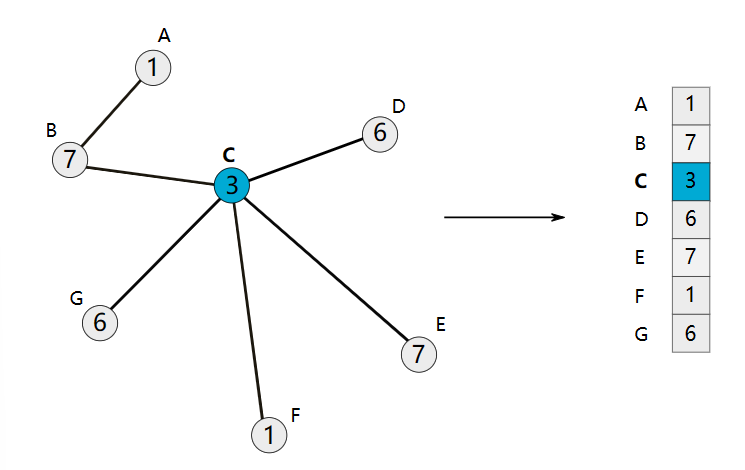


图 ‑10 图信号

之后我们可以使用一个拉普拉斯多项式定义在图信号上的卷积：

为了了解这个过程发生了什么，我们可以从一个最简单的多项式开始，令而其他所有系数都为0，在这种情况下：

接下来，我们讨论更复杂一些的情况，令，其他所有的系数都为0，我们可以得到变换后的特征为：

我们可以发现每个节点上的特征都与它的邻居节点特征进行了融合。对于熟悉拉普拉斯滤波图像的读者来说，这是完全相同的想法，如图15-11所示。

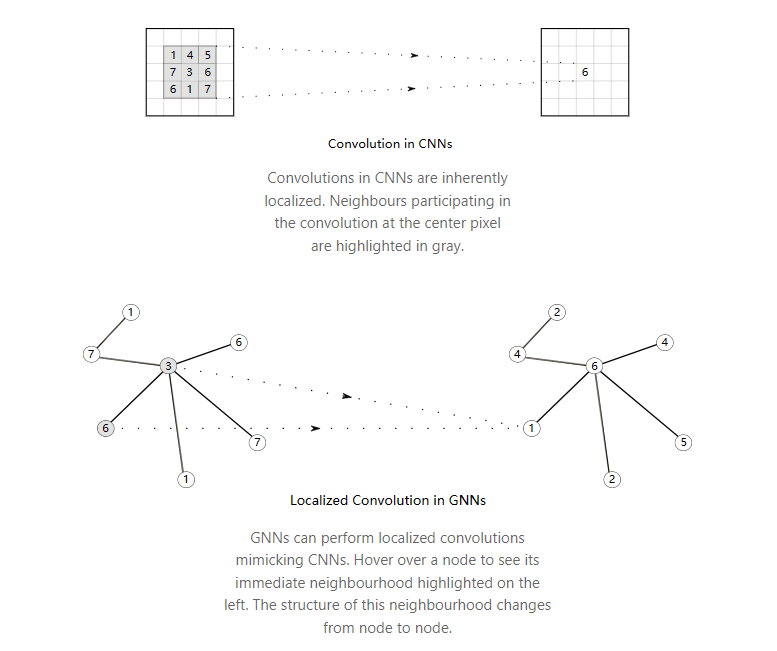


图 ‑11 图卷积

接下来我们讨论不同的次的多项式会对卷积造成什么样的影响。当我们使用次的多项式作用于节点特征时：

我们可以发现，作用与节点上的卷积操作只会涉及它的跳邻域，因此这些多项式其实都是局部的滤波器，滤波的范围由多项式的次数决定。

现在我们大致了解了滤波器是怎样工作的，下面让我们回到最初的例子，使用多项式滤波器对进行卷积：

这是一个1-hop卷积操作，我们可以把这种卷积看作是由两个步骤产生的：1） 聚合节点邻居的特征；2）结合节点自身的特征。我们发现，在每一步之后，每个节点从它的邻居那里接收一些“信息”，因此这些卷积可以被认为是相邻节点之间的“消息传递”。通过反复迭代卷积次（反复传递消息），那么卷积的接收域就可以涵盖的所有节点。那如果我们考虑不同类型的“聚合”和“组合”步骤，而不仅仅使用多项式过滤器，那么可能会怎么样呢？

下面我们将介绍图神经网络模型中的卷积神经网络通用结构。

### 图卷积神经网络

图卷积网络(Graph Convolutional Networks，GCNs)之所以称之为卷积是因为过滤器参数通常在图中的所有位置（或其子集）中共享。

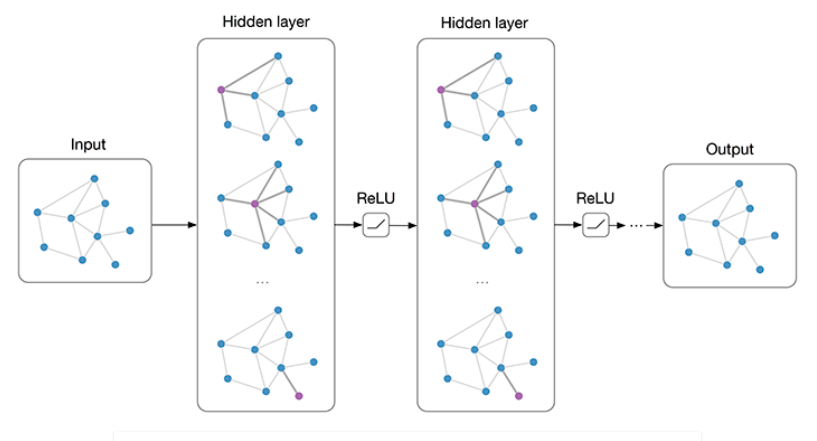


图 ‑12 基于一阶滤波器的多层图卷积网络(GCN)

对于图卷积神经网络来说，其目标是学习图上的信号或者特征的函数，它的输入为：

1. 一个的矩阵，其中是图的节点数，是特征维度，这个矩阵表示图中每个节点上的信号或者特征。
2. 矩阵形式的图结构描述表达，例如常用的邻接矩阵。

图卷积网络的输出为节点级的特征表达（一个的特征矩阵，其中是每个节点的输出特性维度），图级输出可以通过引入某种形式的池操作来获得，我们在之后图级计算任务中会具体谈到。

每层图卷积层都可以写成一个非线性函数：

其中且，是图卷积层的数目，不同模型之间的区别仅仅在于的不同选择（消息传递方式）和参数的设置。

下面让我们来看一个简单的例子，我们来思考下面这种消息传递的规则形式：

其中是第层网络的权重矩阵，是非线性激活函数，例如ReLU函数等。首先让我们看一下这个模型的局限性：使用邻接矩阵意味着对于每个节点，我们计算它的特征向量时，其所有相邻节点的特征向量都会做出贡献，但节点本身的特征向量并不参与计算(除非图中存在自环)。我们只需将恒等矩阵添加到邻接矩阵中，通过强制对图中每个节点添加自环来修正这个问题。其次，矩阵很明显是非归一化的，因此在使用做矩阵乘法时，将会改变原本特征向量的尺度。我们可以对矩阵进行归一化，使之行和为1来加强网络学习时的数值稳定性。在实践中，通常使用对称归一化方式：。结合这两个技巧，我们基本上得出了 Kipf & Welling提出的消息传递方式：

其中，是单位矩阵，是对角矩阵，对角线元素为节点对应的度。如果没有其他说明，我们称式子。。。为图卷积层（GCN layer），以此为主体堆叠多层的神经网络模型称之为图卷积模型。

### 谱域卷积

到目前为止，我们看到的方法都是进行“局部”卷积：每个节点的特征是使用其邻居的特征进行更新。

虽然执行足够多的消息传递步骤最终可以确保来自图中所有节点的信息得到传递，但是我们想知道是否有更直接的方法来直接进行“全局”卷积。我们现在将描述一种方法，这种方法实际上是在神经网络的背景下首次提出的，远早于我们上面讨论的 GNN 模型。

和前面一样，我们将重点讨论节点具有一维特性的情况。在选择了节点顺序之后，我们可以将所有的节点特征进行叠加，得到一个特征向量。在上文中我们定义了拉普拉斯矩阵，如果我们正则化使得，并引入：

我们可以得到图中关于特征向量的平滑度，拉普拉斯矩阵是一个反应图信号局部平滑度的算子。拉普拉斯矩阵是一个实对称矩阵，根据实对称矩阵可以被正交对角化可得：

其中是对角矩阵，对角线元素为排序后的特征值（）：

则是正交矩阵，其中每个向量为对应特征值的特征向量：

对于正交矩阵来说，有，在此基础上我们定义图的傅里叶变换：对于任意一个在图上的信号，其傅里叶变换为：

我们称特征向量为傅里叶基，是在第k个傅里叶基上的傅里叶系数。从定义上我们可以看出，傅里叶系数本质上是图信号在傅里叶基上的投影，衡量了图信号与傅里叶基的相似度。用矩阵形式可以计算出所有的傅里叶系数：

由于U是一个正交矩阵，对上式左乘U，则：

于是我们可以得到逆傅里叶变换：

## 图的计算任务

图上的任务一般有三种类型: 图级任务、节点级任务和边级任务。在图级任务中，每个数据样本是一个完整的图，我们希望预测整个图的性质，对图进行分类预测等；对于节点级任务，其中图中的节点作为数据样本，我们可以预测图中每个节点的一些性质，对其中的节点进行分类、聚类或者预测其中影响最大的节点；对于边界层次的任务，我们希望探寻图中边的性质或者预测节点之间的边是否存在。下面让我们详细地了解一下三类图预测问题。

### 节点级预测任务

在图计算中，节点级任务是指针对图中每个节点设计的任务，在节点级别上进行计算和处理。常见的节点级任务主要有节点分类、节点重要性评估、链接预测等，接下来我们主要讨论一个代表性任务——节点分类。

节点分类涉及到预测图中每个节点的身份或角色，其中一个经典的例子是空手道俱乐部（如图15-13）。该数据集是一个单一的社交网络图，由政治分裂后宣誓效忠于两个空手道俱乐部之一的个人组成。随着故事的发展，Hi 先生(导师)和 John H (管理员)之间的不和导致了空手道俱乐部的分裂。节点代表个别的空手道练习者，边缘代表这些成员之间的关系。该问题的目标是预测分类一个给定的成员在比赛之后是忠于Hi先生还是John H。



图 ‑13 空手道俱乐部图，颜色表示不同的社群

下面是一个节点分类实战代码，使用Cora数据集作为示例：

1. import torch
2. import torch.nn as nn
3. import torch.nn.functional as F
4. from torch\_geometric.nn import GCNConv
5. from torch\_geometric.datasets import Planetoid
6. from torch\_geometric.data import DataLoader
7. # 定义一个简单的图卷积网络（GCN）
8. class GCN(nn.Module):
9. def \_\_init\_\_(self, num\_features, hidden\_size, num\_classes):
10. super(GCN, self).\_\_init\_\_()
11. self.conv1 = GCNConv(num\_features, hidden\_size)
12. self.conv2 = GCNConv(hidden\_size, num\_classes)
13. def forward(self, data):
14. x, edge\_index = data.x, data.edge\_index
15. x = self.conv1(x, edge\_index)
16. x = F.relu(x)
17. x = F.dropout(x, training=self.training)
18. x = self.conv2(x, edge\_index)
19. return F.log\_softmax(x, dim=1)
20. # 加载数据集（这里使用Cora数据集作为示例）
21. dataset = Planetoid(root='data/Cora', name='Cora')
22. # 创建一个图卷积神经网络模型
23. model = GCN(dataset.num\_features, hidden\_size=16, num\_classes=dataset.num\_classes)
24. # 定义损失函数和优化器
25. criterion = nn.NLLLoss()
26. optimizer = torch.optim.Adam(model.parameters(), lr=0.01, weight\_decay=5e-4)
27. # 划分训练集、验证集和测试集
28. data = dataset[0]
29. data.train\_mask = data.val\_mask = data.test\_mask = data.y = None
30. data = DataLoader(data, batch\_size=1, shuffle=True)
31. # 训练模型
32. def train():
33. model.train()
34. optimizer.zero\_grad()
35. for batch in data:
36. out = model(batch)
37. loss = criterion(out[batch.train\_mask], batch.y[batch.train\_mask])
38. loss.backward()
39. optimizer.step()
40. # 测试模型
41. def test():
42. model.eval()
43. correct = 0
44. total = 0
45. for batch in data:
46. out = model(batch)
47. \_, predicted = torch.max(out[batch.test\_mask], 1)
48. total += batch.y[batch.test\_mask].size(0)
49. correct += (predicted == batch.y[batch.test\_mask]).sum().item()
50. accuracy = 100 \* correct / total
51. return accuracy
52. # 训练和测试模型
53. for epoch in range(200):
54. train()
55. if (epoch + 1) % 10 == 0:
56. accuracy = test()
57. print(f'Epoch {epoch + 1}, Accuracy: {accuracy:.2f}%')

### 图级预测任务

在图计算中，图级任务（Graph-level tasks）是指针对整个图进行处理和分析的任务。常见的图级任务主要有图分类、图聚类、图匹配（在两个或多个给定的图之间寻找相似性，以做出比较和分析）、图生成等。本节我们主要介绍其中具有代表性的一类任务——图分类。

在图分类中，我们的目标是预测整个图的性质。例如，对于一个图结构表示的分子（图15-3），我们可能想要预测这个分子是否有毒，或者它是否会与疾病有关的受体结合。这类似于图像领域中的 MNIST 和 CIFAR 的图像分类问题，我们希望将一个标签与整个图像关联起来，或者是自然语言处理中的情绪识别问题，通过整段文本判断其对应情绪。

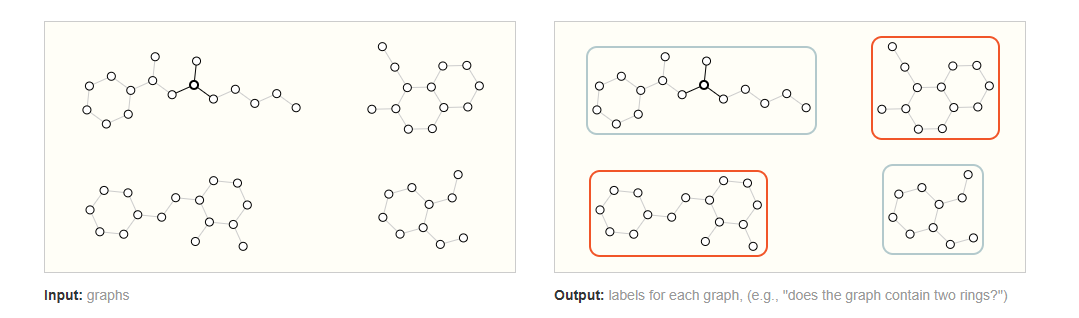


图 ‑14 分子结构预测

不同于节点级任务，图分类任务组要更多的关注图数据的全局信息，既要融合各个节点的特征，又要包含整张图的结构信息。对于图级任务来说，模型更多关注的是如何获得一个优秀的图向量表示。在CNN中，通常使用层级池化来一步步提取全局信息，最终得到一个全局表达，在图神经网络中也有着类似的操作，在这里我们主要介绍基于Top K的的自注意力图池化模型（SAGPool）。

Top K池化机制的总体来说是一个不断抛弃节点过程，通过学习节点的重要程度，对节点进行排序，每次排序丢弃掉那些权重较低的节点保留权重更高的节点。如果仅仅丢弃节点会使得缺乏对所有节点的有效信息的融合，因此常常会选择在池化层后添加一个读出层，实现对该尺度下的图的全局信息的一次性聚合，常用的读出方式有全局平局池化和全局最大池化，或者将二者进行拼接。

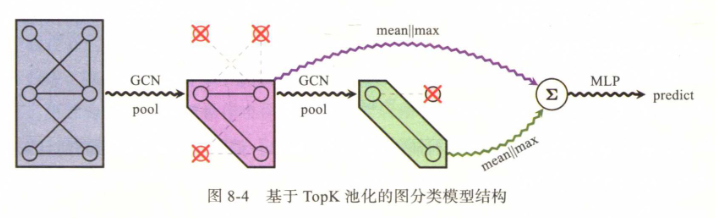


图 ‑15 基于Top K池化的图分类模型结构

如何学习节点的重要程度，不同的研究人员有不同的方法，SAGPool是使用图卷积的方式，对每一个节点赋予一个注意力分数，之后根据节点的注意力分数进行池化操作，根据池化比率（可调整超参），舍弃掉部分不重要的节点，之后对邻接矩阵和节点特征进行更新，最终得到池化结果。

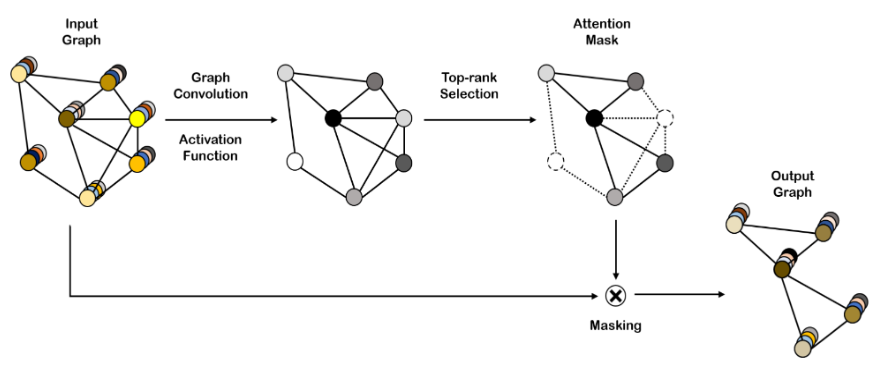
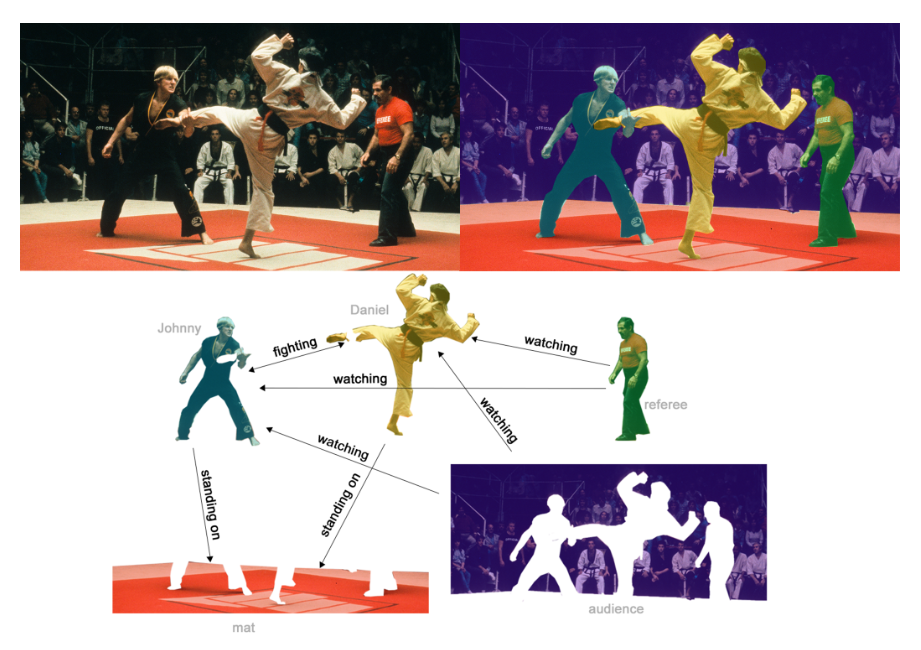


图 ‑16 SAGPool池化架构

### 边级预测任务

边级任务是指针对边进行处理和分析的任务，边级任务中一个典型的例子是图像场景理解。在图像场景理解任务中，除了识别图像中的物体，我们还可以预测节点之间的关系。对于给定图像，节点代表其中的对象，我们希望预测这些节点中的哪些节点之间有边，或者这个边的权重是多少，这类任务我们可以把它称为边级分类。如果我们希望判断节点之间是否有联系，可以首先将图设置为全连通图（每个节点之间都有边连接），之后根据预测值对边进行删除，最终得到一个稀疏图，即我们的预测结果。



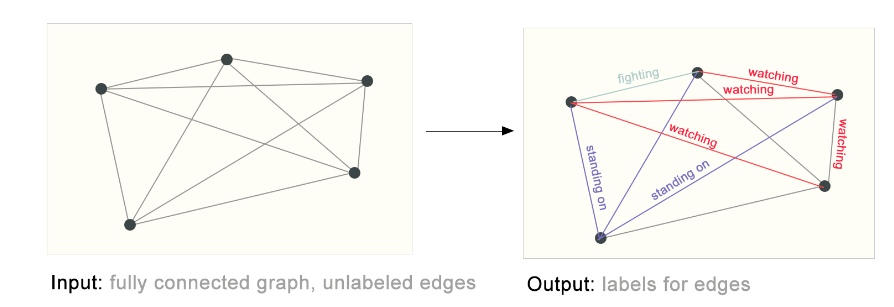


图 ‑17 边级分类任务

图是一种强大而丰富的结构化数据类型，它所有的优势和面临的挑战与图像和文本有很大不同。在这一章中，我们概述了研究人员在建立基于神经网络的模型来处理图结构数据方面所取得的一些阶段性成果。作为一种新兴的神经网络技术，GNN的快速发展离不开深度学习各方面的重要积淀。近年来，GNN的成功为深度学习解决更广泛的新问题提供了帮助，我们确信在未来几年，GNN将在越来越多的场景中发挥作用，我们很期待看到这个领域未来将会发生什么。

## 参考文献

[1] M. Gori, G. Monfardini和F. Scarselli, 《A new model for learning in graph domains》, 收入 *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Montreal, Que., Canada: IEEE, 2005, 页 729–734. doi: 10.1109/IJCNN.2005.1555942.

[2] J. Bruna, W. Zaremba, A. Szlam和Y. LeCun, 《Spectral Networks and Locally Connected Networks on Graphs》, *arXiv.org*, 2013年12月21日. https://arxiv.org/abs/1312.6203v3 (见于 2023年5月6日).

[3] M. Defferrard, X. Bresson和P. Vandergheynst, 《Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering》, *arXiv.org*, 2016年6月30日. https://arxiv.org/abs/1606.09375v3 (见于 2023年5月6日).

[4] T. N. Kipf和M. Welling, 《SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS》, 2017.

[5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò和Y. Bengio, 《Graph Attention Networks》, *arXiv.org*, 2017年10月30日. https://arxiv.org/abs/1710.10903v3 (见于 2023年5月6日).

[6] A. Vaswani等, 《Attention is all you need》, *Adv. Neural Inf. Process. Syst.*, 卷 30, 2017.