

第5章

图神经网络

内容提要

- ❑ 图神经网络
- ❑ 图表示学习
- ❑ 图深度学习
- ❑ 图卷积神经网络
- ❑ 图注意力网络
- ❑ 消息传递神经网络框架
- ❑ 非局部神经网络框架
- ❑ 图网络框架
- ❑ 直推式学习
- ❑ 归纳式学习
- ❑ 图傅里叶变换
- ❑ 图傅里叶逆变换
- ❑ 拉普拉斯矩阵
- ❑ 信息汇聚函数
- ❑ 信息更新函数
- ❑ 信息池化函数

5.1 图神经网络介绍

深度学习飞速发展过程中深度神经网络（Deep Neural Networks, DNN）模型功不可没，如深度前馈神经网络（Deep Feedforward Neural Networks, DFNN）、卷积神经网络（Convolutional Neural Networks, CNN）、循环神经网络（Recurrent Neural Networks, RNN）、图神经网络（Graph Neural Networks, GNN）等。深度前馈神经网络、卷积神经网络、循环神经网络在自然语言处理、图形识别领域都取得了举世瞩目的成就，也是人工智能浪潮的前沿。图神经网络模型能够挖掘图数据信息，具有广泛的应用场景，同时其强大的图表示学习能力也得到大量工业级应用的验证，具有巨大的应用前景和研究价值^[114]。

深度前馈神经网络、卷积神经网络、循环神经网络、图神经网络模型示意图如图5-1所示。深度前馈神经网络模型可以表示为 f_{DNN} ，卷积神经网络模型可以表示为 f_{CNN} ，循环神经网络模型可以表示为 f_{RNN} ，图神经网络模型可以表示为 f_{GNN} 。图5-1中输入数据也只是简单示例，并非不能输入其他数据，如卷积神经网络模型 f_{CNN} 同样可以用时间序列数据作为输入数据。

深度前馈神经网络模型的输入向量无特定结构，输入层神经元之间无时序关系和空间关系。深度卷积神经网络模型输入数据具有一定空间关系，卷积神经网络模型具有局部性和平移不变性以及参数共享等特征。循环神经网络模型能够挖掘时序数据信息，模型内部

能存储长期历史记忆信息，并结合当前时刻输入数据进行信息提取和信息转换，能够整体考虑时间序列信息。图神经网络模型是图强化学习方法的核心组成部分。

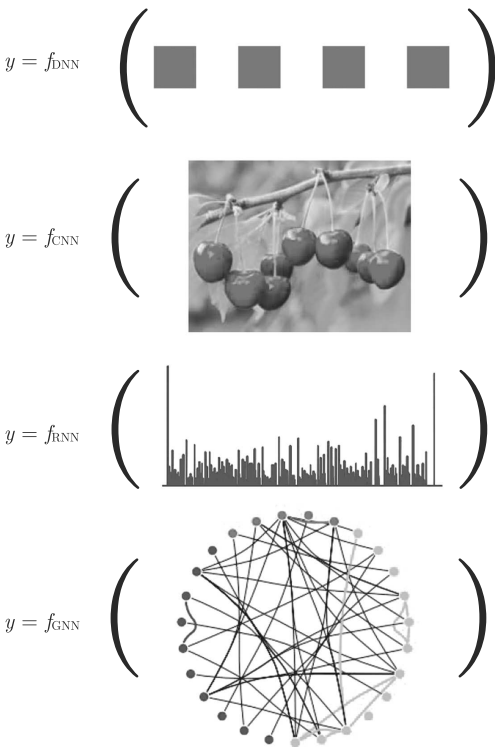


图 5-1 深度前馈神经网络、卷积神经网络、循环神经网络、图神经网络模型示意图

5.2 图神经网络特征

深度前馈神经网络、深度卷积神经网络和循环神经网络都有各自更适用的数据类型，如卷积神经网络模型更加适合图片类型数据，而循环神经网络模型更适合具有时序性质的序列数据类型。在实际应用中，卷积神经网络模型并非就一定只能处理图片类型的数据，同时也能处理时序类型数据，如时间序列和文本数据。图数据和网络数据区别于矩阵型数据（如图片），具有较大的特征差异，深刻理解图数据和网络数据的结构特征，可为理解深度图神经网络模型和深度图强化学习模型打下基础。

5.2.1 图数据特征

图和网络的经典表现形式是邻接矩阵。邻接矩阵是矩阵型数据，但是与图片等矩阵型数据有着巨大差别。虽然图像类型数据也用矩阵表示，每个矩阵元素表示一个像素，但是不能够交换行和列。如果一幅照片随机交换矩阵的行或者列，交换前后的矩阵就不能代表同一幅图。复杂网络和图的邻接矩阵却可以任意地交换行和列，同样可以表示同一个图或



者网络，也就是图论中图同构的概念。邻接矩阵的行和列的交换只是重新对节点进行编号，并不会改变网络结构和特征。图数据特征可以分成结构特征、属性特征、拓扑特征等，属性特征包括节点属性、连边属性和全局图属性。

5.2.2 端到端学习特征

图嵌入和图表示学习的诸多算法属于浅层嵌入，模型学习的参数直接是节点嵌入空间的属性向量。借鉴深度学习模型中端到端学习的优势，我们可以将深度学习模型应用到图数据之中，将网络结构属性和特定问题属性保存在深度神经网络模型的参数之中，通过深度神经网络模型来计算和生成嵌入空间的属性向量，而不是直接通过优化算法来优化网络嵌入空间的属性向量。深度图神经网络模型对图嵌入进行泛化，具有更广的应用场景和更优的实验效果。

端到端学习基于特定问题设定深度学习模型结构，众多深度学习的优化算法以基于特定问题设计的目标函数（损失函数）为优化对象，优化深度神经网络模型参数，使得深度神经网络模型自动调整网络模型参数来适应特定的问题。端到端学习过程较少引入专家经验和偏置，从而使模型更灵活，更重要的是，可以提高模型训练效率和迭代速度，其泛化性也更好。在机器学习模型构建过程中，专家经验有助于特征提取和模型训练，但过多的专家经验融入模型也会影响模型的表示能力和学习能力，以及模型迭代更新的效率。

5.2.3 归纳学习特征

深度图神经网络模型中有大量的归纳学习（Inductive Learning）方法。与归纳学习对应的一种学习方法叫作转导推理学习或者直推式学习（Transductive Learning）。通过对比分析直推式学习与归纳学习，可以发现图神经网络模型归纳学习的诸多优势。

1. 模型训练模式

在机器学习模型训练过程中，直推式学习将所有数据都放入训练集之中，包括有标签的数据和无标签的数据。例如，矩阵分解方法需要计算整个网络的结构信息（邻接矩阵或高阶邻接矩阵）。反之，归纳学习方法的训练过程无须包含无标签的数据。

2. 模型预测模式

在机器学习模型预测过程中，直推式学习无法预测没有出现在训练过程中的无标签数据。对于没有出现在训练过程中的数据，直推式学习方法需要重新训练模型，训练数据中需要包括有标签的数据和没有标签的数据。归纳学习模型能够预测一些没有出现在训练集中的样本，达到真正意义上的预测效果。

3. 模型计算复杂度

一般而言，归纳学习复杂度较低，直推式学习模型训练复杂度较高。一些矩阵分解类型的图嵌入或网络嵌入算法，属于直推式学习模型，矩阵分解的数值计算复杂度较高。归纳学习一般采用随机采样等方法来提高训练效率，并完成模型训练。

5.3 图神经网络框架

消息传递神经网络 (Message Passing Neural Networks, MPNN)、非局部神经网络 (Non-Local Neural Networks, NLNN)、图网络 (Graph Networks, GN) 都是典型的图神经网络模型框架。但是, 消息传递神经网络 (MPNN) 模型框架使用范围最广, 认可度最高, 且更适合入门深度图神经网络模型。同时, 消息传递神经网络框架也是各大主流深度图神经网络模型开源工具包的编程范式^[122, 123]。

5.3.1 图神经网络框架简介

图神经网络模型的目标是挖掘图特征数据或信息, 并进行智能决策。图理论分析、复杂网络分析和图数据挖掘 (Graph Data Mining) 等方法都是为了解构蕴含于复杂图结构和网络结构以及相关特征属性的信息。在机器学习领域, 此过程抽象成为表示学习过程, 将高维、复杂信息抽象成为低维的特征向量, 获得的特征向量更加适应下游决策任务, 有利于进行智能决策。

图神经网络是机器学习模型在图数据上的拓展, 是图表示学习 (Graph Representation Learning, GRL) 的有效方法。图神经网络模型将图数据信息转化为络节点、连边或全局特征向量, 将非欧空间信息转化成欧氏空间信息, 将信息从超高维空间映射到低维稠密空间。图神经网络模型的学习过程就是信息传递过程、信息转换过程和空间映射过程。本质上, 图神经网络模型也是一个映射函数。

5.3.2 消息传递神经网络框架

图神经网络模型和图理论分析、复杂网络分析以及图嵌入和网络嵌入存在一定区别, 有着各自的特色和效用, 在图数据挖掘方面表现出各自的特色。诸多算法之间也有着一些共同的核心思想, 如随机游走。在复杂网络分析中, 很多节点中心性指标都是基于随机游走的思想, 如特征值中心性、PageRank 中心性等。在图嵌入和网络嵌入中, 基于随机游走的图嵌入算法极大地推动了图机器学习算法的落地和实际应用。

在图神经网络模型中, 消息传递神经网络模型的核心思想是消息传递, 和网络中随机游走方法有着异曲同工之妙。基于随机游走的网络节点中心性指标融合了网络邻居节点的中心性; 而在基于随机游走的图嵌入和网络嵌入算法中, 相邻网络节点的嵌入向量具有更高的相似性。在消息传递神经网络模型中, 网络节点同样汇聚邻居节点的信息, 以更新网络节点特征向量。深度理解诸多方法之间的差异和共同点, 才能为图神经网络模型的应用和落地部署夯实基础。

消息传递神经网络模型主要包含邻域信息汇聚函数、信息更新函数和信息池化函数等。图神经网络模型是一个更加宏大的信息处理框架^[122, 123]。图数据属于非欧空间, 很多欧氏空间数据也可以转化成图数据, 借助图神经网络模型强大的表示能力和学习能力完成信息提取和智能决策。



5.3.3 邻域信息汇聚函数

在图神经网络模型中，网络数据用 $G = (V, E)$ 表示，其中 V 表示网络节点集合， E 表示节点之间的连边集合。若连边 $e_{ij} \in E$ ，则表示节点 $i, j \in V$ 之间存在一条连边，如国际贸易网络中的商品贸易关系、股票投资者交易网络中的股票买卖关系、朋友网络中好友关系等。邻域 $\mathcal{N}_i$ 定义为与节点 i 相连的所有节点的集合，即节点 i 的邻居集合。特征向量 $\mathbf{x}_i$ 表示节点 i 的特征属性向量。

网络 $G = (V, E)$ 作为图神经网络模型的输入数据，用于计算节点的隐含特征表示向量 $\mathbf{x}_i^{(k)}$ 。节点 i 的特征属性表示向量是基于节点 i 的邻居节点集合 $\mathcal{N}_i$ 计算而来，具体计算公式为

$$\mathbf{x}_{-i}^{(k)} = \text{Aggregate}_{j \in \mathcal{N}_i} f^{(k)}(\mathbf{x}_j^{(k-1)}) \quad (5.1)$$

其中： $\mathbf{x}_{-i}^{(k)}$ 表示聚合了节点 i 的邻居节点集合 $\mathcal{N}_i$ 的属性表示，但是不包括网络节点 i 自身的属性信息。

聚合函数 Aggregate 可以有多种形式，最常见的聚合函数 Aggregate 包括求和函数 (SUM)、均值函数 (MEAN)、最大值函数 (MAX) 等，也可以是深度学习模型，如 DNN、RNN 或 LSTM 等。同样，信息处理函数 $f^{(k)}$ 也可以嵌套一些深度神经网络模型，如 DNN、RNN 或 LSTM。因此，图神经网络模型也被称为深度图神经网络模型。深度神经网络模型作为子模块或计算模块处理图信息，即 $f^{(k)} = f_{\text{DNN}}^{(k)}$ 或者 $f^{(k)} = f_{\text{RNN}}^{(k)}$ 等。

在聚合函数 Aggregate 的公式中，需要特别注意的是上标 (k) ，我们可以把上标 (k) 理解成深度图神经网络模型中聚合邻居节点的网络层数。 $k = 1$ 表示聚合一阶邻居节点信息； $k = 2$ 表示聚合二阶邻居节点信息，即聚合与网络节点 i 距离为 2 的所有网络节点信息。在图神经网络模型中，网络节点的邻居层数和深度神经网络中神经网络隐藏层数不一样。在实际应用中，我们一般选择 $k = 2, 3, 4$ 基本就足够了， k 取太大容易出现过平滑等问题。

在现实世界中，一般的复杂网络结构具有小世界性质。当图神经网络模型中聚合函数的邻居层数太大时，基本上所有节点的邻居都一样，即包括网络的所有节点。比如图神经网络模型的输入数据的网络直径为 5，当 $k = 6$ 时，任何一个网络节点的 6 阶邻居都是网络中的所有节点。因此，任何一个网络节点聚合的邻居信息具有同质性，图神经网络模型容易出现过平滑问题。

5.3.4 信息更新函数

在图神经网络模型中，节点 i 聚合了邻居信息后，可以更新节点 i 自身的特征属性为

$$\mathbf{x}_i^{(k)} = \text{Update}(\mathbf{x}_i^{(k-1)}, \mathbf{x}_{-i}^{(k)}) \quad (5.2)$$

其中： $\mathbf{x}_i^{(k-1)}$ 表示第 $k-1$ 次迭代计算中节点 i 的特征表示。Update 函数可以有多种形式，最常见的更新函数 Update 是深度神经网络模型，也可以是简单的求和函数 (SUM) 或者均值函数 (MEAN) 等，或者进行简单的向量拼接。我们通过聚合函数 Aggregate 和更新

函数 Update, 进行 k 步迭代, 每迭代一次就增加了一层邻居信息, 最后得到节点 i 的属性表示向量 $\mathbf{x}_i^{(k)}$ 。在一些特殊问题中, 节点 i 的属性表示向量可以表示为

$$\mathbf{x}_i = [\mathbf{x}_i^{(1)} \parallel \mathbf{x}_i^{(2)} \parallel \cdots \parallel \mathbf{x}_i^{(k)}] \quad (5.3)$$

属性表示向量 $\mathbf{x}_i$ 拼接了每一层的属性表示向量信息, 作为节点 i 的属性向量。

5.3.5 图信息池化函数

在图神经网络模型中, 如果涉及整体网络层面的任务和决策问题, 需要对整个网络进行表征, 比如对网络进行分类、判断分子网络的物理或化学性质等。为此, 我们可以对节点属性进行池化操作, 即

$$\mathbf{x}^{(k)} = \text{Pool}_{\forall i}(\mathbf{x}_i^{(k)}) \quad (5.4)$$

图信息池化函数具有一些重要的性质。例如, 图信息池化函数必须对网络节点的顺序不敏感, 满足网络节点顺序可交换性。因此, 最简单的池化操作就是将所有网络节点属性求和为

$$\mathbf{x}^{(k)} = \text{SUM}_{\forall i}(\mathbf{x}_i^{(k)}) \quad (5.5)$$

求和函数中的加法满足交换律, 因此节点的顺序差异不影响整体网络的属性。类似于复杂网络分析中全局网络聚类系数计算公式, 我们可以计算所有节点聚类系数的平均值。同样, 为了获得全局网络属性特征, 可以将网络节点属性求平均, 即

$$\mathbf{x}^{(k)} = \text{MEAN}_{\forall i}(\mathbf{x}_i^{(k)}) \quad (5.6)$$

$\mathbf{x}^{(K)}$ 作为整个网络的特征表示向量, 下游任务可以在 $\mathbf{x}^{(K)}$ 的基础上进行智能决策, 如分类预测和回归预测。在图神经网络模型中, 池化操作的可选函数众多, 很难找到适用于不同任务和问题中的最优操作算子。在图神经网络模型的实际训练中, 池化操作可以作为超参数进行调优, 选择最优的池化函数以获得最优的决策性能。

5.4 图卷积神经网络

图卷积神经网络是图神经网络领域的经典模型, 是入门图神经网络的基础模型, 其中包含大量相关的图论基础概念和方法, 如谱图理论、拉普拉斯矩阵、图傅里叶变换、图傅里叶逆变换等。

5.4.1 谱图理论介绍

谱图理论 (Spectral Graph Theory) 是分析图的拉普拉斯矩阵特征值和特征向量及其对应图性质的理论。图用邻接矩阵表示, 通过对矩阵的分析能够洞察图的结构特征和属性信息。矩阵分析是一个应用非常广泛的工具, 矩阵的特征值和特征向量是矩阵分析中十分重要的概念。矩阵特征值和特征向量有普适的性质和规律。基于图相关矩阵的分析能够获得图或网络的特定性质和规律, 从一个崭新的角度来挖掘图结构信息。



5.4.2 拉普拉斯矩阵定义

除了邻接矩阵，图相关的矩阵中拉普拉斯矩阵（Laplacian Matrix）是图结构的一个重要的矩阵表示形式。图拉普拉斯矩阵具体定义为

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (5.7)$$

其中：矩阵 $\mathbf{A}$ 表示图 G 的邻接矩阵， $\mathbf{D}$ 为对角矩阵，对角线元素 $D_{ii} = d(v_i) = k_i$ ， k_i 为节点 i 的度。网络度对角矩阵具体形式为

$$\mathbf{D} = \begin{bmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(v_N) \end{bmatrix} = \begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_N \end{bmatrix} \quad (5.8)$$

一般情况下，我们考虑无环、无向、无权简单图，因此邻接矩阵是对称矩阵，满足 $\mathbf{A} = \mathbf{A}^T$ 。同时， $\mathbf{D}$ 为对角矩阵，也是对称矩阵，满足 $\mathbf{D} = \mathbf{D}^T$ 。因此，原始的拉普拉斯矩阵是对称矩阵，满足

$$\mathbf{L} = \mathbf{L}^T \quad (5.9)$$

拉普拉斯矩阵具有很多特性，应用广泛，图神经网络模型也将用到诸多拉普拉斯矩阵的性质。

5.4.3 随机游走归一化拉普拉斯矩阵

在实际使用过程中，拉普拉斯矩阵需要进行转换。例如，考虑随机扩散情况时，我们可以采用随机游走归一化拉普拉斯矩阵，具体定义为

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{A} \quad (5.10)$$

从定义可以看出，随机游走归一化拉普拉斯矩阵并不一定是对称矩阵，这也限制了随机游走归一化拉普拉斯矩阵在一些特定问题中的应用。

5.4.4 对称归一化拉普拉斯矩阵

在拉普拉斯矩阵的实际使用过程中，为了满足矩阵的对称性，可采用对称归一化拉普拉斯矩阵，具体定义为

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \quad (5.11)$$

原始拉普拉斯矩阵、随机游走归一化拉普拉斯矩阵和对称归一化拉普拉斯矩阵有着不同的特征，适用于不同的应用场景和实际问题。

5.4.5 拉普拉斯矩阵简单应用

在谱图理论中，拉普拉斯矩阵有着广泛应用。熟悉拉普拉斯矩阵的简单应用，有助于理解拉普拉斯矩阵的一些性质和特征。网络节点的状态或表示向量可定义为向量 $\mathbf{f}$ ，向量大小为 $N \times 1$ ，分别对应网络中 N 个节点的状态信息。此处考虑最简单情况，即节点状态信息变量的维度为 1，后面将考虑多维变量情况。节点 i 的状态信息可用 f_i 表示， $\mathbf{f}$ 为向量， f_i 为标量。如果用拉普拉斯矩阵与节点状态变量 $\mathbf{f}$ 相乘，我们可以得到新的节点状态变量，即

$$\mathbf{f}' = \mathbf{L}\mathbf{f} = (\mathbf{D} - \mathbf{A})\mathbf{f} = \mathbf{D}\mathbf{f} - \mathbf{A}\mathbf{f} \quad (5.12)$$

网络节点 i 的新状态对应向量 $\mathbf{f}'$ 的分量 f'_i ，可以表示为

$$\begin{aligned} f'_i &= d(v_i)f_i - \sum_{j=1}^N a_{ij}f_j = d(v_i)f_i - \sum_{v_j \in \mathcal{N}_i} a_{ij}f_j \\ &= \sum_{v_j \in \mathcal{N}_i} f_i - \sum_{v_j \in \mathcal{N}_i} f_j = \sum_{v_j \in \mathcal{N}_i} (f_i - f_j) \end{aligned} \quad (5.13)$$

上式的推导过程用到公式为

$$d(v_i) = \sum_{j=1}^N a_{ij} = \sum_{v_j \in \mathcal{N}_i} a_{ij} = |\mathcal{N}_i| = k_i \quad (5.14)$$

网络节点新状态是所有邻居节点与自身节点状态值的差值之和。拉普拉斯矩阵大小为 $N \times N$ 。网络节点属性或状态变量 $\mathbf{f}$ 大小为 $N \times 1$ ，因此新状态变量 $\mathbf{f}'$ 大小也为 $N \times 1$ 。我们将原来状态值 $\mathbf{f}$ 的转置 $\mathbf{f}^T$ 和新状态 $\mathbf{f}' = \mathbf{L}\mathbf{f}$ 作内积，可得

$$\begin{aligned} \mathbf{f}^T \mathbf{f}' &= \mathbf{f}^T \mathbf{L}\mathbf{f} = \sum_{v_i \in V} f_i \sum_{v_j \in \mathcal{N}_i} (f_i - f_j) \\ &= \sum_{v_i \in V} \sum_{v_j \in \mathcal{N}_i} f_i (f_i - f_j) \\ &= \sum_{v_i \in V} \sum_{v_j \in \mathcal{N}_i} (f_i f_i - f_i f_j) \\ &= \sum_{v_i \in V} \sum_{v_j \in \mathcal{N}_i} \left(\frac{1}{2} f_i f_i - f_i f_j + \frac{1}{2} f_j f_j \right) \\ &= \frac{1}{2} \sum_{v_i \in V} \sum_{v_j \in \mathcal{N}_i} (f_i - f_j)^2 \end{aligned} \quad (5.15)$$

节点属性或状态变量向量的转置 $\mathbf{f}^T$ 大小为 $1 \times N$ ，新状态变量 $\mathbf{f}'$ 大小也为 $1 \times N$ ，因此 $\mathbf{f}^T \mathbf{f}' = \mathbf{f}^T \mathbf{L}\mathbf{f}$ 为标量。由于状态向量 $\mathbf{f}$ 的任意性，因此可以说 $\mathbf{f}^T \mathbf{L}\mathbf{f}$ 总是非负的，表明拉普拉斯矩阵是半正定矩阵。 $\mathbf{f}^T \mathbf{L}\mathbf{f}$ 衡量网络中所有节点与其邻居节点状态值差异的平方和。从另一个角度而言， $\mathbf{f}^T \mathbf{L}\mathbf{f}$ 度量网络节点状态之间的平滑性或者同质性。



5.4.6 图信号处理

在图理论和方法以及复杂网络分析中，除了关系图和网络结构特征，实际网络分析中网络节点都具有关联的特征和属性，可以用特征变量表示。在实际问题分析中，我们同时结合图或网络结构信息和属性信息能够获得更多有价值的决策信息。图信号处理包括图 $G = (V, E)$ 以及将网络节点映射为数值的函数，即

$$F: V \rightarrow \mathbb{R}^{N \times d} \quad (5.16)$$

在上述映射函数中，参数 d 表示节点属性向量的维度，与图嵌入和网络嵌入中节点嵌入空间的维度类似。简单起见，此示例中我们使用 $d = 1$ 的情况，即映射函数可以表示为

$$F: V \rightarrow \mathbb{R}^{N \times 1} \quad (5.17)$$

在复杂网络分析中，从节点中心性和特征向量表示，到图嵌入和网络嵌入，再到图神经网络模型等，都是为了能够从复杂网络结构信息和语义信息中解构、挖掘、学习到最优的和有效的节点、连边或全局网络的表示向量。图表示学习和网络表示学习的核心就是构建一个映射关系，将图或网络映射为特征向量表示。如何得到最优表示向量，必须结合特定的目标任务。图节点、图连边或全局图的表示学习特征向量只是中间变量，或叫隐变量和潜变量，学习到的表示变量最终是为了完成目标任务，或揭示复杂网络和复杂图的内在结构特征信息。

5.4.7 图傅里叶变换

在经典的信号处理中，信号可以在时域和频域进行表示。通过傅里叶变换和傅里叶逆变换能够将信号在时域和频域之间进行转换。同样，图信号也可以在空域和谱域（频域）进行表示。类似于傅里叶变换，同样也可以定义图傅里叶变换（Graph Fourier Transform, GFT）为

$$\hat{f} = U^T f \quad (5.18)$$

向量 f 为图信号（节点属性变量），矩阵 U 的每一列对应拉普拉斯矩阵的一个特征向量，其矩阵转置为

$$U^T = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ u_{21} & u_{22} & \cdots & u_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{NN} \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_N^T \end{bmatrix} \quad (5.19)$$

其中： u_k 表示拉普拉斯矩阵的第 k 个特征值所对应的特征向量，为列向量，大小为 $N \times 1$ 。

图傅里叶变换也可以表示为

$$\hat{\mathbf{f}} = \mathbf{U}^T \mathbf{f} = \begin{bmatrix} \mathbf{u}_1^T \mathbf{f} \\ \mathbf{u}_2^T \mathbf{f} \\ \vdots \\ \mathbf{u}_N^T \mathbf{f} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N u_{1i} f_i \\ \sum_{i=1}^N u_{2i} f_i \\ \vdots \\ \sum_{i=1}^N u_{Ni} f_i \end{bmatrix} \quad (5.20)$$

在图傅里叶分析中，拉普拉斯矩阵的特征值所对应的特征向量表示图上的傅里叶基， $\hat{\mathbf{f}}$ 对应由傅里叶基（特征向量）张成的空间中网络节点的位置坐标。在图神经网络模型分析中，我们需要用到特征向量矩阵 $\mathbf{U}$ 的几个非常重要的性质或规律。例如，矩阵特征向量都进行了归一化，满足

$$\mathbf{u}_k^T \mathbf{u}_k = 1 \quad (5.21)$$

且

$$\mathbf{u}_k^T \mathbf{L} \mathbf{u}_k = \mathbf{u}_k^T \lambda_k \mathbf{u}_k = \lambda_k \mathbf{u}_k^T \mathbf{u}_k = \lambda_k \quad (5.22)$$

5.4.8 图傅里叶逆变换

图傅里叶逆变换（Inverse Graph Fourier Transform, IGFT）与傅里叶逆变换类似。分析和学习图傅里叶逆变换之前，需要对相关的概念进行理解。在图傅里叶分析中，拉普拉斯矩阵的特征向量矩阵 $\mathbf{U}$ 满足

$$\mathbf{I} = \mathbf{U}^T \mathbf{U} = \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_N^T \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_N \end{bmatrix} \quad (5.23)$$

由此可知，拉普拉斯矩阵的特征向量矩阵 $\mathbf{U}$ 的逆为其转置 $\mathbf{U}^T$ 。根据逆矩阵性质可知

$$\mathbf{U} \mathbf{U}^T = \mathbf{U} \mathbf{I} \mathbf{U}^T = \mathbf{U} \mathbf{U}^T \mathbf{U} \mathbf{U}^T = (\mathbf{U} \mathbf{U}^T)^2 \quad (5.24)$$

因此

$$\mathbf{U} \mathbf{U}^T (\mathbf{U} \mathbf{U}^T - \mathbf{I}) = 0 \quad (5.25)$$

同样，我们可得到

$$\mathbf{U} \mathbf{U}^T = \mathbf{I} \quad (5.26)$$

图傅里叶逆变换可以表示为

$$\mathbf{U} \hat{\mathbf{f}} = \mathbf{U} \mathbf{U}^T \mathbf{f} = \mathbf{f} \quad (5.27)$$

因此，图傅里叶逆变换也可以表示为

$$\mathbf{f} = \mathbf{U} \hat{\mathbf{f}} \quad (5.28)$$