第 5 章

图注意力网络

注意力机制(Attention Mechanism)最初在机器翻译模型中被引入并使用,现在已经成为自然语言处理(Natural Language Processing,NLP)、计算机视觉(Computer Vision,CV)、语音识别(Speech Recognition,SR)领域中神经网络模型的重要组成部分。近年来,有些研究人员将注意力机制应用到图神经网络模型中,取得了很好的效果。本章聚焦于图注意力网络模型,依次介绍注意力机制的概念、图注意力网络的分类,以及四个典型的注意力模型:图注意力网络模型(Graph Attention Networks,GAT)、异质图注意力网络(Heterogeneous Graph Attention Networks,HAN)、门控注意力网络(Gated Attention Networks,GaAN)和层次图注意力网络(Hierarchical Graph Attention Networks,HGAT)。

5.1 注意力机制

在学习图注意力模型之前,需要先理解什么是注意力模型。考虑到有些读者之前没有接触过注意力模型,本节将从注意力模型的原理、变体、优势、应用场景四方面展开介绍,以帮助大家建立对注意力模型的全面认识。

注意力机制的原理可以通过类比人类生物系统的注意力功能来理解。注意力是人类一种极其重要和复杂的认知功能,具体表现为人类大脑处理信息时会重点关注有价值的信息。例如,关于听觉注意力的例子:在人山人海的火车站,尽管四周充满了噪声,我们依然可以清晰地听到家人的聊天内容;关于视觉注意力的例子:在高速路上驾驶小汽车时,我们会重点观察前方车辆的行驶情况,以及一些重要的路标信息,对于视野中的其他信息通常会忽略掉。

神经网络中的注意力机制是一种与人类生物系统注意力功能类似的信息处理机制。在处理大量的输入信息时,注意力机制会选择一些关键的输入信息进行处理,忽略与目标无关的噪声数据,从而提高神经网络模型的效果。

下面从数学原理角度进一步阐述什么是注意力机制。我们所说的注意力机制通常是指软性注意力机制(Soft Attention Mechanism),注意力机制涉及 3 个要素,即请求(Query)、键(Key)和值(Value),图 5-1 所示是软性注意力机制的计算过程,$(\boldsymbol{K},\boldsymbol{X})$是输入的键值对向量数据,包含 n 项信息,每一项信息的键用 K_i 表示,值用 X_i 表示,Q 表示与任务目标相关的查询向量,Value 是在给定 Q 的条件下,通过注意力机制从输入数据中提取的有用信

息，即输出信息。注意力机制的计算过程可以归纳为三步：①计算 K_i 与 Q 的相关性得分；②将计算的相关性得分使用 Softmax 函数进行归一化处理，归一化后的值 $\alpha_i \in [0,1]$ 称为注意力分布，又称为注意力系数，其值越大，表明第 i 个输入信息与任务目标的相关性越高；③根据注意力系数 α 对输入数据 $\boldsymbol{X}$ 进行加权求和计算。

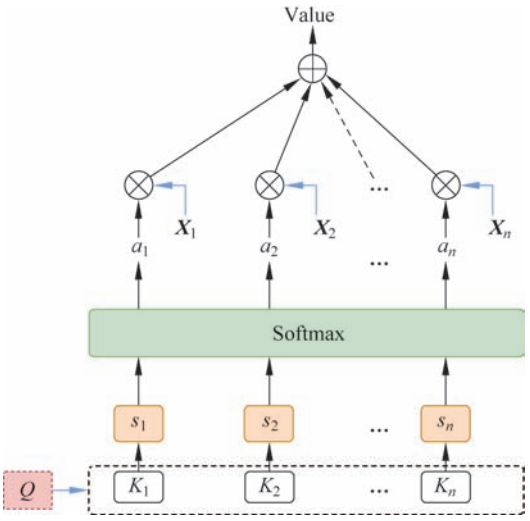

图 5-1 注意力机制

软性注意力机制的计算如式(5.1)所示。

$$\text{Value} = \sum_{i}^{n} \text{Softmax}(S(\boldsymbol{K}_i, \boldsymbol{Q})) \cdot \boldsymbol{K}_i \tag{5.1}$$

其中，$S(\boldsymbol{K}_i, \boldsymbol{Q})$ 为相似性度量函数，常见的计算方式有如下几种。

(1) 点积模型。

$$S(\boldsymbol{K}_i, \boldsymbol{Q}) = \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{K}_i \tag{5.2}$$

(2) 缩放点积模型。

$$S(\boldsymbol{K}_i, \boldsymbol{Q}) = \frac{\boldsymbol{Q}^{\mathrm{T}} \boldsymbol{K}_i}{\sqrt{d}} \tag{5.3}$$

(3) 矩阵相乘模型。

$$S(\boldsymbol{K}_i, \boldsymbol{Q}) = \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{K}_i \tag{5.4}$$

(4) 余弦相似度模型。

$$S(\boldsymbol{K}_i, \boldsymbol{Q}) = \frac{\boldsymbol{Q}^{\mathrm{T}} \boldsymbol{K}_i}{\| \boldsymbol{Q} \| \cdot \| \boldsymbol{K}_i \|} \tag{5.5}$$

(5) 加线性模型。

$$S(\boldsymbol{K}_i, \boldsymbol{Q}) = \boldsymbol{V}^{\mathrm{T}} \tanh(\boldsymbol{W}\boldsymbol{Q} + \boldsymbol{U}\boldsymbol{K}_i) \tag{5.6}$$

其中，$\boldsymbol{V}$、$\boldsymbol{W}$、$\boldsymbol{U}$ 都是可学习的网络参数，d 是输入数据的维度。最常用的两种计算方式是点积模型和加线性模型，理论上两者的复杂度相当，但在实践中，点积模型更快且更节省空间，因为它可以使用高效的矩阵乘法进行优化。当输入信息的维度 d 较大时，点积模型的值通常会出现比较大的方差，从而会使得 Softmax 函数的梯度比较小，在这种情况下，缩放点积

模型是一种更好的选择。

5.1.1　注意力机制的变体

前面介绍了基本的注意力模型，即软性注意力模型，接下来将介绍注意力的一些变体模型。

1. 硬性注意力

软性注意力会考虑所有的输入信息，根据输入信息的重要性生成相应的关注权重。此外，还存在一种注意力，它只关注输入信息中某一个位置的信息，这种注意力机制称为硬性注意力机制(Hard Attention Mechanism)。

硬性注意力会选取最高概率的输入信息，或者在注意力分布上进行随机采样选取信息。这个计算过程可以理解为在输入信息中选择一个信息，将其注意力权重设置为 1，其他的信息权重全部设置为 0。硬性注意力选择信息的方式决定了其效果具有不稳定性；另外硬性注意力最终的损失函数与注意力分布之间的函数关系不可导，导致其无法使用反向传播算法进行训练，一般而言，硬性注意力模型需要采用强化学习的方法来训练。

2. 局部注意力

软性注意力需要计算所有输入信息，效果稳定，但计算量大；硬性注意力计算量小，但效果不稳定。局部注意力则是软性注意力和硬性注意力的一种折中方案，其思路是先使用硬性注意力定位到一个位置，然后以这个位置为中心点，设置一个窗口区域，在窗口区域内使用软性注意力进行计算。其优势是窗口内的计算效率和效果稳定性可以通过参数进行调节。

3. 多头注意力

多头注意力(Multi-Head Attention)使用多个任务目标 $\boldsymbol{Q}=[q_1,q_2,\cdots,q_k]$ 独立地进行 k 次注意力计算，由于每次计算的 q 不同，所以每个注意力所关注的信息也不同，这样可以从输入信息中抽取 k 个不同的信息，最后将 k 个信息进行拼接操作。

4. 层次结构注意力

如果输入信息本身具有一定的层次结构，例如，文本可以划分为词、句子、段落、篇章等不同粒度的内容，我们可以使用层次结构注意力在每个层进行更好的信息选择，首先可以在词层面使用注意力机制生成一个句子的向量表达，然后在句子层面使用注意力机制生成一个段落的向量表达，最后在段落层面使用注意力机制生成整个文本的向量表达。

5. 自注意力

当使用神经网络对一个变长的序列数据建模时，通常可以使用卷积神经网络(Convolution Neural Networks，CNN) 或者循环神经网络(Recurrent Nerual Networks，RNN)。基于卷积神经网络的序列建模可以看成一种局部建模方式，只能建模输入数据的

局部依赖关系；循环神经网络虽然理论上可以建立长距离依赖关系，但是由于梯度消失和传递信息的容量问题，实际上对长距离依赖的建模能力较弱。

自注意力(Self-Attention)模型动态计算序列内部信息之间的权重，能够建模变长序列内部的依赖关系。相比卷积神经网络，自注意力模型能够将卷积核的固定长度感受野扩大到输入序列长度的范围；相比循环神经网络，自注意力模型对长距离依赖有更强的捕获能力，并且能够并行计算。基于以上优势，自注意力模型被广泛应用于序列数据建模领域，自然语言处理领域中著名的 Transformer 模型是自注意力模型的典型代表。

假设注意力模型的输入序列为 $\boldsymbol{X}=[x_1,x_2,\cdots,x_N]\in\mathbb{R}^{d_1\times N}$，输出序列为 $\boldsymbol{H}=[h_1,h_2,\cdots,h_N]\in\mathbb{R}^{d_2\times N}$，对输入数据 $\boldsymbol{X}$ 进行线性变换，得到以下三个向量。

$$\boldsymbol{Q}=\boldsymbol{W}_{\boldsymbol{Q}}\boldsymbol{X}\in\mathbb{R}^{d_3\times N}$$

$$\boldsymbol{K}=\boldsymbol{W}_{\boldsymbol{K}}\boldsymbol{X}\in\mathbb{R}^{d_3\times N}$$

$$\widetilde{\boldsymbol{X}}=\boldsymbol{W}_{\widetilde{\boldsymbol{X}}}\boldsymbol{X}\in\mathbb{R}^{d_2\times N}$$

其中，$\boldsymbol{Q}$、$\boldsymbol{K}$、$\widetilde{\boldsymbol{X}}$分别为查询向量、键向量和值向量。利用式(5.1)，计算得到输出向量 $\boldsymbol{h}_i$：

$$\boldsymbol{h}_i=\sum_{j=1}^{N}\mathrm{Softmax}(s(q_j,k_j))\tilde{x}_j$$

自注意力模型在计算序列数据项之间的相关性时只考虑了数据本身的信息，而忽略了输入数据的位置信息，因此在单独使用自注意力模型时，需要加入位置编码信息来进行修正，具体做法可以参考 Transformer 模型。

5.1.2 注意力机制的优势

注意力机制的优势可以归纳为以下三点。

(1) 注意力机制能够有效地使模型忽略输入数据中的噪声部分，从而提升信噪比。

(2) 注意力机制可以为输入数据中不同元素分配不同的权重系数，以突出与任务最相关的信息元素。

(3) 注意力机制为模型结果带来了更好的解释性。例如，在翻译任务中，分析句子中不同单词的权重系数，可以找出句子中的关键词。

5.1.3 应用场景

注意力机制由于其效果优、可解释性好，已成为活跃的研究领域。鉴于注意力机制在一些领域取得了非常好的效果，并成为某些情况下最流行的技术选择。下面就注意力机制的应用场景做简单说明。

1. 自然语言处理

在 NLP 领域，注意力机制有助于聚焦输入序列的相关部分，对齐输入和输出序列，以及捕获长序列的长距离依赖性问题。

2. 计算机视觉

视觉注意力已经在很多主流 CV 任务中流行起来，用于关注图像内的相关区域，并捕获图像各部分之间的、结构性的、长期依赖关系。视觉注意力还为图像中的目标检测提供很大的好处，可以帮助定位和识别图像中的对象。

3. 图计算

现实世界的很多重要的数据集以图形或网络的形式存在，包括社交网络、知识图谱、蛋白质相互作用网络和万维网等。注意力机制在图数据上的应用主要在于突出显示与目标任务更相关的图元素，包括顶点、边和子图。图的注意力计算是高效的，因为它可以跨顶点邻居对进行并行计算，可以应用于具有不同度的顶点。目前注意力机制已经应用于顶点分类、链路预测、图分类、图序列生成等任务。

5.2　同质图注意力网络

本节主要介绍图注意力网络模型，该模型首次将注意力机制引入图的消息传播步骤中。图中每个顶点的隐藏层状态由自身特征以及其邻域中的邻居特征计算得到，假设目标顶点邻域中的每个顶点对于目标顶点的重要性是有差异的（这也符合大多数真实数据的分布），引入注意力机制后，模型能够捕获邻居顶点对于目标顶点的重要性差异，并将这种差异信息转化为权重参数用于邻居信息聚合，最终的效果为越重要的顶点，其信息在聚合过程中保留越多。该模型对于信息的聚合操作更符合数据分布，因而效果相对于 GCN 模型有所提升。

5.2.1　图注意力层

图注意力网络中定义了一个图注意力层，通过叠加不同的图注意力层，可以组成复杂结构的图注意力网络。一个图注意力层的结构如图 5-2 所示。图注意力层输入的是顶点特征向量的集合。

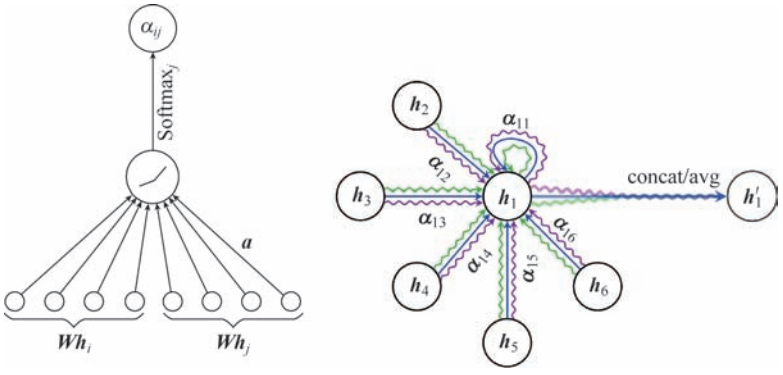

图 5-2　图注意力层

$$\boldsymbol{h}=\{\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_N\} \tag{5.7}$$

其中，N 是顶点的数量，$\boldsymbol{h}_i\in\mathbb{R}^F$，$F$ 是顶点特征的维数。

一般来说，将输入特征转换为高阶特征可以获得更好的表达能力，为了达到这个目的，需要将原始特征经过一次可学习的线性变换，变换后的特征表示为

$$\boldsymbol{h}'=\{\boldsymbol{h}'_1,\boldsymbol{h}'_2,\cdots,\boldsymbol{h}'_N\} \tag{5.8}$$

其中，$\boldsymbol{h}'_i\in\mathbb{R}^{F'}$，$F'$是变换后特征的维数。得到变换后的特征后，应用自注意力机制(Self-Attention)计算权重系数：

$$e_{ij}=\mathrm{att}(\boldsymbol{W}\boldsymbol{h}_i,\boldsymbol{W}\boldsymbol{h}_j) \tag{5.9}$$

其中，e_{ij} 表示顶点 j 对顶点 i 的重要性，att(·)是一个共享的相关性函数：$\mathbb{R}^{F'}\times\mathbb{R}^{F'}\to\mathbb{R}$，由于两个顶点的重要性由一个数值来表示，所以由式(5.9)计算得到的权重系数是一个标量，即 $e_{ij}\in\mathbb{R}$。

为了尽量保留图数据中的结构信息，这里只在目标顶点 i 的邻域顶点中计算注意力，即 $j\in N_i$，N_i 是顶点 i 的一阶邻居，通过这种计算方式能够将图中结构信息编码到目标顶点的向量表示中。为了方便比较邻域中不同顶点之间的权重系数，使用 Softmax 函数对 e_{ij} 进行归一化处理：

$$\alpha_{ij}=\mathrm{Softmax}_j(e_{ij})=\frac{\exp(e_{ij})}{\sum\limits_{k\in N_i}\exp(e_{ik})} \tag{5.10}$$

在实践中，相关性度量函数 a 可以使用一个单层的前馈神经网络，其参数矩阵表示为 $\boldsymbol{a}\in\mathbb{R}^{2F'}$，具体做法是将顶点 i 和顶点 j 变换后的特征进行拼接操作，得到一个特征为 $2F$ 维的特征向量，然后输入到前馈神经网络中，并用 LeakyReLU 函数进行激活。因此权重系数的最终计算公式如下：

$$\alpha_{ij}=\frac{\exp(\mathrm{LeakyReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i\,||\,\boldsymbol{W}\boldsymbol{h}_j]))}{\sum\limits_{k\in N_i}\exp(\mathrm{LeakyReLU}(\boldsymbol{a}^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i\,||\,\boldsymbol{W}\boldsymbol{h}_k]))} \tag{5.11}$$

其中，T 表示为矩阵的转秩操作，|| 表示特征向量的拼接操作，LeakyReLU 激活函数的公式如下：

$$\mathrm{LeakyReLU}(x)=\max(0.01x,x) \tag{5.12}$$

得到目标顶点 i 的各个邻居的权重系数后，就可以通过线性组合得到顶点 i 新的向量表示：

$$\boldsymbol{h}'_i=\sigma\Big(\sum_{j\in N_i}\alpha_{ij}\boldsymbol{W}\boldsymbol{h}_j\Big) \tag{5.13}$$

给定节点 i，设其邻居集合为$\{j_1,j_2,j_3,j_4\}$，则该节点的图注意力计算过程如图 5-3 所示。

对比图注意力层与第 4 章介绍的图卷积层的顶点更新公式，可以发现图注意力层多了一个自适应的边权重系数项。在图卷积神经网络中，这个权重值是图的拉普拉斯矩阵，而在图注意力模型中，这个权重系数是可以自适应学习的，从而使得图注意力模型具有更强的表达能力。另外，图注意力模型与 GraphSAGE 相同，保留了图的完整局部性，能够进行归纳式学习。

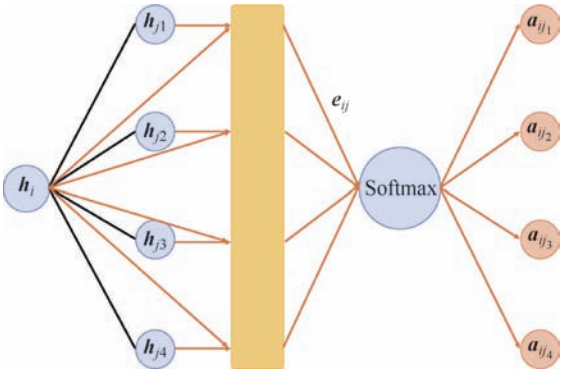

图 5-3　单层图注意力神经网络计算流程示意图

5.2.2　多头注意力

为了增强模型的表达能力和稳定性，可以使用多头注意力（Multi-Head Attention），具体的做法是对每个目标顶点执行 K 次独立的注意力计算，这样将得到目标顶点 i 的 K 个向量表示，然后可以对这 K 个向量进行拼接操作或者求平均计算，得到输出向量。

一般来说，如果多头注意力应用于模型的中间隐藏层，可以采用拼接向量的方式得到目标顶点的中间隐藏层向量表示，公式为

$$\boldsymbol{h}'_i = \Big\|_{k=1}^{K} \sigma\Big(\sum_{j\in N_i} \alpha_{ij}^k \boldsymbol{W}^k \boldsymbol{h}_j\Big) \tag{5.14}$$

其中，α_{ij}^k 是第 k 个注意力机制的归一化系数，$\boldsymbol{W}^k$ 对应第 k 个线性变换的权重矩阵，拼接后的向量 $\boldsymbol{h}'_i \in \mathbb{R}^{KF'}$。如果多头注意力应用于神经网络模型的最后预测层，拼接操作已经不再敏感有效，取而代之的是使用平均计算，并且将非线性变换延迟到平均计算之后，公式为

$$\boldsymbol{h}'_i = \sigma\Big(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i} \alpha_{ij}^k \boldsymbol{W}^k \boldsymbol{h}_j\Big) \tag{5.15}$$

其中，σ 在分类任务中通常为 Softmax 函数或者 Sigmoid 计算。

5.3　异质图注意力网络

目前大多数图神经网络，如 GraphSAGE、图注意力网络等模型主要针对同质图设计，但真实世界中的图大部分可以被自然地建模为异质图，如互联网电影数据库（Internet Movie Database，IMDB）的数据中包含三种类型的节点：Actor、Movie 和 Director，两种类型的边：Actor-Movie 和 Movie-Director。异质图神经网络具有更强的现实意义，可以更好地满足工业界需求，但是由于不同类型顶点的语义不同，顶点的特征维数也不尽相同，异质图的这些异质性给设计图神经网络带来了巨大的挑战。

图注意力网络取得了非常大的成功，但是其只能应用于同构图中。为了能够在异质图中应用注意力机制，Wang 等人提出了异质图注意力网络，利用语义级别注意力和顶点级别注意力来同时学习元路径与顶点邻居的重要性，并通过相应的聚合操作得到目标顶点的最

终向量表示。

异质图注意力网络模型整体架构如图 5-4 所示。模型可以拆分为 3 个模块：顶点级别注意力模块、语义级别注意力模块和预测模块。

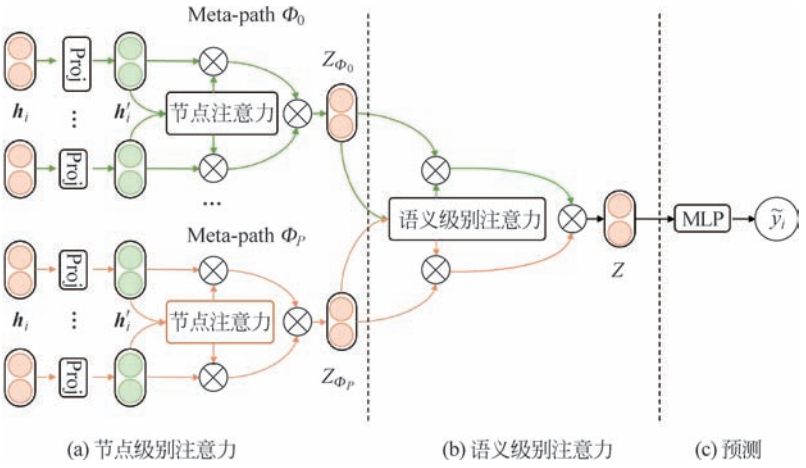

图 5-4　异质图注意力网络架构

5.3.1　顶点级别注意力

在异质图中，不同类型顶点的特征维度一般不同，为了能够统一处理不同类型顶点的特征，首先需要将不同类型的顶点特征通过转换矩阵变换到统一的特征空间。

$$\boldsymbol{h}'_i = \boldsymbol{M}_{\Phi_i} \cdot \boldsymbol{h}_i \tag{5.16}$$

其中，$\boldsymbol{M}_{\Phi_i}$ 是投影矩阵，$\boldsymbol{h}_i$ 和 $\boldsymbol{h}'_i$ 分别是投影变换前后的顶点特征。通过特定类型的投影转换操作，顶点级别注意力能够处理任意类型的顶点。完成了特征的转换后，可以利用自注意力机制(Self-Attention)学习多种类型顶点的权重。对于一个给定的元路径 Φ 的顶点对 (i,j)，顶点级别的注意力能够学习到顶点 j 对于顶点 i 的重要性。基于元路径顶点对 (i,j) 的重要性可以表示为

$$e_{ij}^{\Phi} = \mathrm{att}_{\mathrm{node}}(\boldsymbol{h}'_i, \boldsymbol{h}'_j; \Phi) \tag{5.17}$$

其中，$\mathrm{att}_{\mathrm{node}}$ 表示执行顶点级别注意力的深度神经网络，给定一个元路径 Φ，$\mathrm{att}_{\mathrm{node}}$ 层中的网络参数被该元路径下的所有顶点对共享，因为同一元路径下的顶点连接情况是相似的。分析式(5.17)可以发现，顶点 i 对顶点 j 的重要性和顶点 j 对顶点 i 的重要性可能会有所不同，表明顶点级别的注意力能够保留不对称性。具体计算注意力时，采用的是 Masked Attention，也就是计算 e_{ij}^{Φ} 时，仅考虑顶点 i 的邻居，即 $j \in N_i^{\Phi}$，这里的 N_i^{Φ} 表示的是顶点 i 在元路径 Φ 的邻居。这样就成功将目标顶点的结构信息编码到其向量表征中了。经过 Softmax 函数进行归一化后的权重系数计算公式如为

$$\alpha_{ij}^{\Phi} = \mathrm{Softmax}_j(e_{ij}^{\Phi}) = \frac{\exp(\sigma(\boldsymbol{a}_{\Phi}^{\mathrm{T}}[\boldsymbol{h}'_i \,||\, \boldsymbol{h}'_j]))}{\sum_{k \in N_i^{\Phi}} \exp(\sigma(\boldsymbol{a}_{\Phi}^{\mathrm{T}}[\boldsymbol{h}'_i \,||\, \boldsymbol{h}'_k]))} \tag{5.18}$$

其中，$[\boldsymbol{h}'_i \,||\, \boldsymbol{h}'_j]$ 表示将顶点 i 和顶点 j 变换后的特征进行拼接操作。每个顶点的表示由自

身特征和其邻居特征加权融合得到：

$$z_i^{\Phi}=\sigma\Big(\sum_{j\in N_i^{\Phi}}\alpha_{ij}^{\Phi}\cdot\boldsymbol{h}'_j\Big) \tag{5.19}$$

其中，z_i^{Φ} 是顶点在某个元路径下的表示。为了充分学习，可以将单头注意力扩展为多头注意力机制，如果有 K 头注意力机制，做 K 次学习，然后拼接得到多头的节点表达：

$$z_i^{\Phi}=\Big\|_{k=1}^{K}\sigma\Big(\sum_{j\in N_i^{\Phi}}\alpha_{ij}^{\Phi}\cdot\boldsymbol{h}'_j\Big)$$

5.3.2　语义级别注意力

给定某条元路径，顶点级别注意力可以学习到顶点在某个语义下的表示，然而在实际异质图中，往往存在多条不同语义的元路径，单条元路径只能反映顶点某一方面的信息。为了得到语义级别注意力，可以通过顶点级别的聚合操作来学习特定语义下(Semantic-specific)的顶点表示。

为了全面地表征顶点，需要利用语义级别注意力来学习语义的重要性，并融合多个语义下的顶点表示。若给定 P 组元路径$\{\Phi_1,\Phi_2,\cdots,\Phi_p\}$，则可以得到相应的 p 组节点表示$\{Z_{\Phi_1},Z_{\Phi_2},\cdots,Z_{\Phi_p}\}$。语义级别注意力的形式化描述如下：

$$(\beta_{\Phi_0},\beta_{\Phi_1},\cdots,\beta_{\Phi_p})=\mathrm{att}_{\mathrm{sem}}(Z_0^{\Phi},Z_1^{\Phi},\cdots,Z_p^{\Phi}) \tag{5.20}$$

其中，$\mathrm{att}_{\mathrm{sem}}$ 是语义级别注意力的深度神经网络，$(\beta_{\Phi_0},\beta_{\Phi_1},\cdots,\beta_{\Phi_p})$是各元路径的注意力权重。具体来说，利用单层神经网络和语义级别注意力向量来学习各个语义(元路径)的重要性，并通过 Softmax 函数进行归一化。通过对多个语义进行加权融合，可以得到最终的顶点向量表示。

为了得到语义级别注意力中不同元路径的重要性，首先采用非线性变换，将语义特定的节点表示转换到同一个特征空间。然后平均元路径 Φ_p 下每个节点表征与语义级别向量 $\boldsymbol{q}$ 的内积，得到每一条元路径的重要性：

$$w_{\Phi_p}=\frac{1}{|V|}\sum_{i\in V}\boldsymbol{q}^{\mathrm{T}}\cdot(\boldsymbol{W}Z_i^{\Phi_p}+\boldsymbol{b}) \tag{5.21}$$

其中，$\boldsymbol{W}$ 是共享权重，$\boldsymbol{b}$ 为偏置，$\boldsymbol{q}$ 是语义级注意力向量。需要注意的是，$\boldsymbol{W}$、$\boldsymbol{b}$ 和 $\boldsymbol{q}$ 对于不同的元路径是共享的。为了获得每条元路径的相对重要性，可以通过 Softmax 函数进行归一化，得到路径 Φ_p 的权重 β_{Φ_p}，公式如下：

$$\beta_{\Phi_p}=\frac{\exp(w_{\Phi_p})}{\sum_{p=1}^{P}\exp(w_{\Phi_p})} \tag{5.22}$$

通过加权求和可以得到节点的最终向量表征：

$$Z=\sum_{p=1}^{P}\beta_{\Phi_p}Z_{\Phi_p} \tag{5.23}$$

得到节点最终的表征后，可以做多种下游任务。需要指出的是，权重 β_{Φ_p} 表示路径 Φ_p 对于最终任务的重要性，β_{Φ_p} 越大越重要。

5.4 门控注意力网络

门控注意力网络不同于图注意力网络模型中同等地看待多头注意力的重要性，门控注意力网络使用一个卷积子网络来控制每个注意力头的重要性。门控注意力网络可以通过引入的门来调节参与内容的数量。得益于门的构造中只引入了一个简单的、轻量级的子网，使得计算开销可以忽略不计，而且模型易于训练。

一般形式的图聚合器(Graph Aggregators)函数可以表示为

$$\boldsymbol{y}_i = \gamma\Theta(\boldsymbol{x}_i, \{z_{N_i}\}) \tag{5.24}$$

其中，$\boldsymbol{x}_i$ 和 $\boldsymbol{y}_i$ 分别为目标顶点 $i \in V$ 的输入和输出向量，其邻域顶点集合为 N_i，$z_{N_i} = \{z_j \mid j \in N_i\}$ 是目标顶点的邻居顶点的向量集合，γ 是聚合器可学习的参数，Θ 也是可学习参数。

那么注意力形式的卷积聚合器如何构造呢？需要将给定节点 i 的特征和其邻居节点的特征按照 5.1 节介绍的注意力机制的要素(Query、Key、Value)进行改造并聚合。此处将节点 i 的特征 x_i 做线性变换，得到 Query，即 $FC_{\theta_{xa}}(x_i)$；将邻居节点的特征经过线性变换得到 Key，即 $\{FC_{\theta_{za}}(z_j), j \in N_i\}$；将邻居节点的特征经过非线性变换得到 Value，即 $\{FC^h_{\theta_v}(z_j), j \in N_i\}$，其中，$FC^h_\theta(\cdot)$ 表示非线性变换，对应全连接神经网络 $FC^h_\theta = h(Wx + b)$，其激活函数 $h(\cdot)$ 为 LeakyReLU，可学习参数为 $\theta = \{W, b\}$，而 $FC_\theta(\cdot)$ 则表示不增加激活函数的线性变换。Query 与 Key 的相似度则采用内积的形式给出：

$$\phi_w(\boldsymbol{x}_i, z_j) = \langle FC_{\theta_{xa}}(\boldsymbol{x}_i), FC_{\theta_{za}}(z_j) \rangle \tag{5.25}$$

单头注意力的输出为 Value 值的权重求和，即 $\sum_{j \in N_i} \boldsymbol{w}_{i,j} FC^h_{\theta_v}(z_j)$，其中权重 $\boldsymbol{w}_{i,j}$ 为

$$\boldsymbol{w}_{i,j} = \frac{\exp(\phi_w(\boldsymbol{x}_i, z_j))}{\sum_{l \in N_i} \exp(\phi_w(\boldsymbol{x}_i, z_l))} \tag{5.26}$$

单头注意力卷积计算可以表示为

$$\boldsymbol{y}_i = FC_{\theta_o}(\boldsymbol{x}_i \oplus \sum_{j \in N_i} \boldsymbol{w}_{i,j} FC^h_{\theta_v}(z_j)) \tag{5.27}$$

类似地，多头注意力聚合器可以表示为

$$\boldsymbol{y}_i = FC_{\theta_0}\left(\boldsymbol{x}_i \oplus \| _{k=1}^{K} \sum_{j \in N_i} \boldsymbol{w}_{i,j}^{(k)} FC^h_{\theta_v^{(k)}}(z_j)\right) \tag{5.28}$$

K 是注意力头的数量。与图注意力网络模型相比，门控注意力网络模型在计算相似性时，将输入向量经过一个全连接层做了一次线性变换，并将 Key 向量经过一个全连接层得到一个新的向量。

每个头注意力聚合器具有探索目标顶点与其邻域顶点构建的表征子空间的能力，但是多头注意力聚合器生成的多个子空间的重要性是不同的，某些顶点甚至可能不存在于某些子空间中，使用某个注意力头捕获的无用顶点表征会误导模型最终的预测效果。为了解决这个问题，计算一个额外的门控给每个注意力头分配重要性，其值分布为 0～1(0 代表低重要性，1 代表高重要性)，门控注意力聚合器的计算示意图如图 5-5 所示。权重系数向量表示为

$$\boldsymbol{g}_i = [g_i^{(1)}, \cdots, g_i^{(K)}] = \phi_g(x_i, z_{N_i}) \tag{5.29}$$

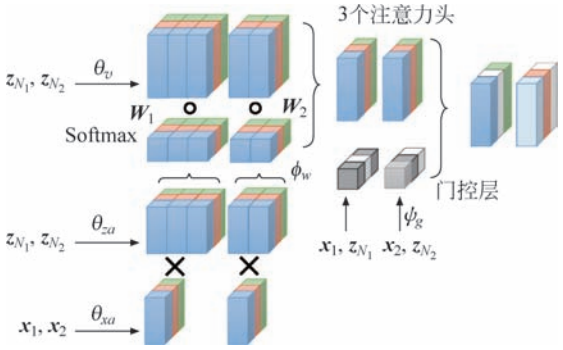

图 5-5　门控注意力网络模型示意图

其中，$\boldsymbol{g}_i \in \mathbb{R}^K$，为了避免因为加入门结构而引入过多的参数，这里使用一个卷积网络结构 ψ_g，该卷积使用目标顶点 i 和其邻居顶点的特征来生成门控值。加入门控的多头注意力聚合器的计算公式如下：

$$\boldsymbol{y}_i = FC_{\theta_0}\left(\boldsymbol{x}_i \oplus \|_{k=1}^{K}\left(g_i^{(k)} \cdot \sum_{j \in N_i} \boldsymbol{w}_{i,j}^{(k)} FC_{\theta_v^{(k)}}^{h}(z_j)\right)\right) \tag{5.30}$$

卷积网络 ψ_g 可以有多种实现方式，其中使用平均池化和最大池化结合的实现方式如下：

$$\boldsymbol{g}_i = FC_{\theta_g}^{\sigma}\left(\boldsymbol{x}_i \oplus \max_{j \in N_i}(\{FC_{\theta_m}(z_j)\}) \oplus \frac{\sum_{j \in N_i} z_j}{|N_i|}\right) \tag{5.31}$$

其中，θ_m 用于将邻居特征向量映射为 d_m 维向量，然后用于求最大值。θ_g 用于将拼接的特征向量映射为 K 个门控值。通过设置较小的 d_m，可以将计算门控值的开销降到忽略不计。

5.5　层次图注意力网络

图注意力网络也可应用于视觉关系检测任务上，因此提出了层次图注意力网络，将图像中的物体的空间相对关系抽象成图结构来处理。在图中引入先验知识和注意力机制，来减轻初始化图时的不准确引入带来不利影响。

5.5.1　视觉关系检测

对于给定的图像 I，视觉关系检测的目标是生成一些用来描述物体关系的三元组，其表示形式为<主语、谓语、宾语>。使用 O 和 P 分别表示物体集合和谓词集合，则关系集合可以表示为 $R=\{r(s,p,o) \mid s,o \in O, p \in P\}$，其中 s、p、o 分别表示关系三元组中的主语、谓语、宾语，则视觉关系检测的概率模型可以表示为

$$P(r) = P(p \mid s,o)P(s \mid b_s)P(o \mid b_o) \tag{5.32}$$

其中，b_s 和 b_o 分别表示主语和宾语对应的物体的边界框，$P(s \mid b_s)$ 和 $P(o \mid b_o)$ 分别表示主语和宾语对应物体边界框的置信度。

从视觉关系检测的定义可以看出，视觉关系检测不仅需要识别出图像中的物体及其位置，还需要识别物体之间关系。物体之间的相互关系使用三元组来表示，其中主语和宾语表

示物体，谓语用来描述两个物体的关系，这个关系可以是描述空间关系的介词、表达动作或状态的动词等，常见的介词有 on、under、above、near 等，动词有 wear、eat、take 等。

早期的工作中，将每一种关系三元组分配一个唯一的类别，但是这样带来了搜索空间的爆炸问题。假设有 N 个物体类别，K 个谓词类别，则对象检测的搜索空间为 N，将关系表示为三元组时的类别总数为 N^2K。

后来通过分离预测过程的方式解决了搜索空间爆炸的问题，与直接将关系三元组作为一个整体学习任务不同，该方法单独预测对象和谓词。在这种方式下，共享相同谓词的关系三元组被归为同一种类别，例如< truck，on，street >和< car，on，street >共享谓词 on，因而被划分为同一个类别，于是搜索空间大小变为 $N+K$。分离预测的代价是同一个谓词分类中样本差异很大。

为了更好地区分谓词，有些研究者引入视觉、空间和语义信息来表示物体，这极大地提高了模型的性能，这些方法可以捕获一对物体之间的交互关系，但是无法明确建模上下文信息。为了解决这个问题，有些研究引入图结构来探索物体之间的联系和约束。

层次图注意力网络模型通过显式地建模三元组之间的依赖关系，可以将更多上下文信息和全局约束纳入关系推理中。此外，先前工作中的图是基于对象的空间相关性构建的，可以通过考虑语义相关性来进行改进。

5.5.2 层次图注意力网络模型框架

层次图注意力网络模型的架构如图 5-6 所示，分为三个子模块：特征表示模块、层次图注意力网络和谓词预测模块。特征表示模块的输入是图像，生成带有边界框和标签的对象，并且会为每个对象提供视觉、空间、语义特征以及成对对象的关系特征。层次图注意力网络通过层次图结构进行对象级和三元组级的推理。谓词预测模块的输入是对象级别推理和三元组级别推理的特征，输出是多个关系三元组。由于本章聚焦于图注意力机制这个主题，所以接下来会重点介绍层次图注意力网络这个子模块。

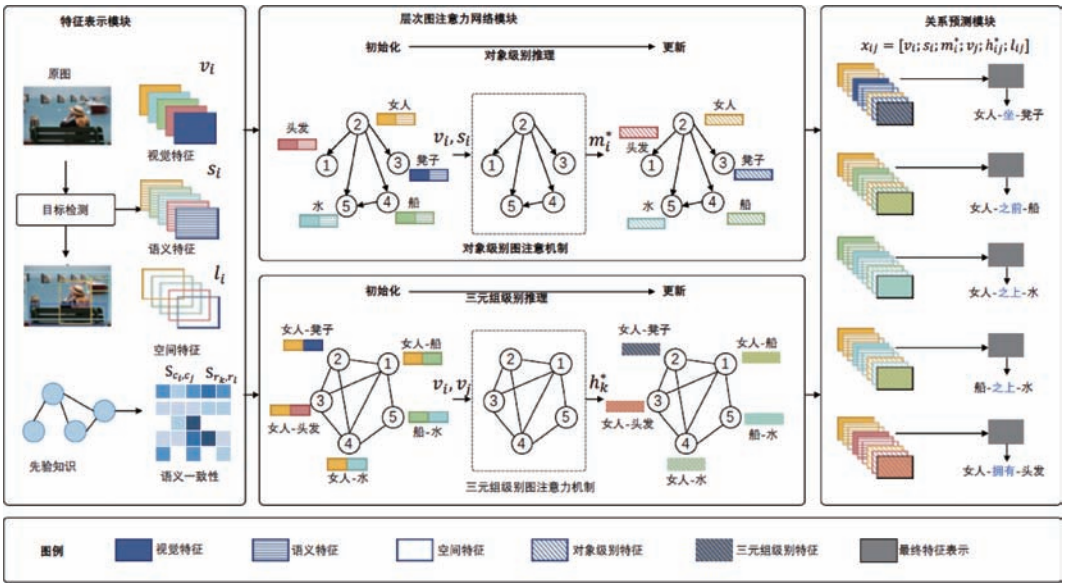

图 5-6 层次图注意力网络模型架构图

层次图注意力网络模块使用两种类型的图来建模。一种是对象级别的图，用来建模对象之间的交互并进行对象级别的推理；一种是三元组级别的图，它由三元组之间的交互关系构建，用于三元组级别的推理。这两种图都使用了注意力机制。

1. 对象级推理

一个对象级别的图表示为 $G_0=\{V_0,E_0\}$，包含顶点集合 V_0 和边集合 E_0，每个顶点 $n_i\in V_0$ 表示一个由边界框和对应的属性嵌入组成的对象，每条边 $e_{ij}^o\in E_0$ 表示为对象 n_i 和对象 n_j 之间的谓词关系。图 G_0 是一个有向图，即三元关系组 (n_i,e_{ij}^o,n_j) 和 (n_j,e_{ij}^o,n_i) 表示两个不同的实例。

在构建对象级别的图 G_0 时，需要考虑两个因素，一个是空间相关性，一个是语义相关性。使用归一化相对距离 $\mathrm{dis}(b_i,b_j)$ 和归一化交并比 $\mathrm{iou}(b_i,b_j)$ 评估两个对象物体的空间相关性，则空间图可以定义为

$$e_{ij}^{\mathrm{sp}}=\begin{cases}1, & \mathrm{dis}(b_i,b_j)<t_1 \text{ 或者 } \mathrm{iou}(b_i,b_j)>t_2\\ 0, & \text{其他}\end{cases} \tag{5.33}$$

其中，t_1 和 t_2 是两个阈值，默认设置的值为 0.5。为了评估一对对象的语义相关性，基于语义一致性建立语义图公式如下：

$$e_{ij}^{\mathrm{se}}=\begin{cases}1, & S_{c_i,c_j}>t_3\\ 0, & \text{其他}\end{cases} \tag{5.34}$$

其中，t_3 是阈值，默认为 0。S_{c_i,c_j} 指语义一致性函数，度量 c_i 和 c_j 在语义上的一致性公式如下：

$$S_{c_i,c_j}=\max\left(\ln\frac{n(c_i,c_j)N}{n(c_i)n(c_j)},0\right)$$

其中，$n(c_i,c_j)$ 是词 c_i 与 c_j 的共现概率，$n(c_i)$ 和 $n(c_j)$ 分别表示词 c_i 和 c_j 出现的频率。有了以上空间图和语义图的定义，则对象级别的图可以表示为

$$e_{ij}^o=e_{ij}^{\mathrm{sp}}\oplus e_{ij}^{\mathrm{se}} \tag{5.35}$$

其中，$\oplus$ 表示或操作。对象级别注意力机制可以表示为

$$m_i^*=\sigma\Big(\sum_{j\in N_i}\alpha_{ij}\cdot(W_{\mathrm{dir}(i,j)}^o\boldsymbol{m}_j+\boldsymbol{b})\Big) \tag{5.36}$$

其中，m_i^* 表示使用注意力后生成的隐特征，m_j 表示顶点的属性向量，由顶点的视觉和语义特征联合表达，即 $\boldsymbol{m}_j=\mathrm{concat}(v_j,s_j)$，注意力系数 α_{ij} 的计算公式为

$$\alpha_{ij}=\frac{\exp((\boldsymbol{U}^o\boldsymbol{m}_i)^{\mathrm{T}}\cdot\boldsymbol{V}_{\mathrm{dir}(i,j)}^o\boldsymbol{m}_j+\boldsymbol{c})}{\sum_{j=1}^{K}\exp((\boldsymbol{U}^o\boldsymbol{m}_i)^{\mathrm{T}}\cdot\boldsymbol{V}_{\mathrm{dir}(i,j)}^o\boldsymbol{m}_j+\boldsymbol{c})} \tag{5.37}$$

其中，$\boldsymbol{U}^o$、$\boldsymbol{V}^o\in\mathbb{R}^{d_m\times(d_v+d_s)}$ 是投影矩阵，$\boldsymbol{b}$、$\boldsymbol{c}$ 是偏置项。$\mathrm{dir}(i,j)$ 根据每个边的方向性选择变换矩阵。

2. 三元组级推理

某种关系更可能同时出现，基于这种假设，构建一个三元组级别的图，用于捕获关系实

例之间的这种依赖关系。将所有可能的关系三元组当成顶点集合 V_t，三元组之间的交互当成边集合 E_o，则三元组级别的图 $G_t=\{V_t,E_t\}$ 可以表示为

$$e_{kl}^{t}=\begin{cases}1, & S_{r_k,r_l}>t_4\\ 0, & \text{其他}\end{cases} \tag{5.38}$$

在实验中，t_4 设置为 0，三元组级别的图是一个无向图。三元组图上的注意力机制可以表示为

$$\boldsymbol{h}_k^{*}=\sigma\Big(\sum_{l\in N_k}\alpha_{kl}\cdot\boldsymbol{W}^{t}\boldsymbol{h}_l\Big) \tag{5.39}$$

其中，$\boldsymbol{h}_k^{*}$ 表示三元组实例 k 使用注意力机制之后得到的新特征表示，$\boldsymbol{h}_l$ 表示顶点 l 的属性特征，也就是三元组实例 l 的视觉特征。注意力权重系数 α_{kl} 的计算公式如下：

$$\alpha_{kl}=\frac{\exp((\boldsymbol{U}^{t}\boldsymbol{h}_k)^{\mathrm{T}}\cdot\boldsymbol{V}^{t}\boldsymbol{h}_l)}{\sum_{l\in N_k}\exp((\boldsymbol{U}^{t}\boldsymbol{h}_k)^{\mathrm{T}}\cdot\boldsymbol{V}^{t}\boldsymbol{h}_l)} \tag{5.40}$$

其中，$\boldsymbol{U}^{t}$、$\boldsymbol{V}^{t}\in\mathbb{R}^{d_h\times(d_v+d_v)}$ 是投影矩阵。

5.6 本章小结

本章聚焦于注意力机制在图神经网络模型中的应用，首先介绍了注意力模型的相关知识，然后引出注意力机制在图计算领域的应用，并对注意力机制的应用场景做了一个简单的分类。最后详细介绍了四个典型的图注意力网络模型：图注意力网络、异质图注意力网络、门控注意力网络和层次图注意力网络，其中，图注意力网络模型是将注意力应用于同构图中邻居顶点信息的聚合过程，并且使用多头注意力来提升模型效果的稳定性；异质图注意力网络模型针对异质图的特点，设计了顶点级别注意力和语义级别的注意力；门控注意力网络则在图注意力网络的基础上，引入了一个注意力门控，用于多个头注意力的聚合；层次图注意力网络模型是将图神经网络应用到视觉检测任务上的一个很好尝试，模型引入了层次注意力建模实体之间的关系。