图深度学习

介绍

图是**描述对象之间复杂交互关系**的一种方式，也被称为**网络。**

图的实际应用

图的应用可分为：

- 节点预测：预测给定节点的类型或属性的值
- 边预测：预测两个节点之间是否存在连接
- 图预测：对不同的图进行分类或预测图的属性
- 节点预测：检测节点是否形成一个社区
- 其他：图生成、图演变

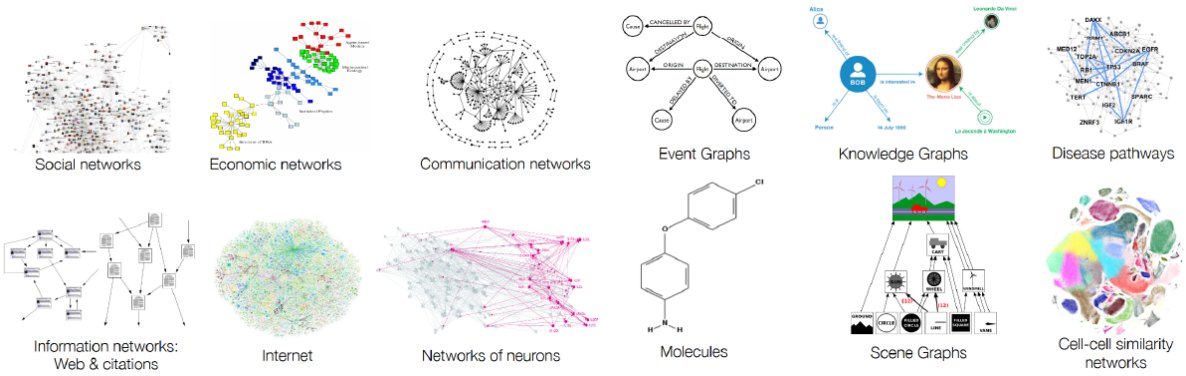


图1 图的应用

现实中的应用：

- 社交网络：社交网络是6度分离，即一个人和另一个不认识的人之间可以通过6个人相互连接起来
- 影响力传播：网络分析对于识别基础设施的弱点或病毒传播
- 知识图谱（语义网络）：现实世界试题的网络，并说明相互之间的关系。一个知识图谱只要由三部分组成：节点、边和标签。节点可以是任何对象，边定义了节点间的关系
- 推荐系统：预测用户的偏好可以抽象成预测图中边的存在的问题
- 生物化学应用：蛋白质原子结构和药物的相互作用效果建模

挑战

图数据是**非规则化、非结构化**的，具有以下特点：

- 任意的大小和复杂的拓扑结构
- 没有固定的节点排序或参考点
- 通常是动态的，具有多模态的特征
- 图的信息不仅包含节点和边，还包括图的拓扑结构

而深度学习是为规则且结构化的数据设计的，无法直接用于图数据。应用图数据的神经网络，要求：

- 适用于不同的节点
- 节点表征的计算与邻接节点的排序无关
- 不但能够根据节点信息、邻接节点和边的信息计算节点表征，还能够根据图拓扑结构计算节点表征

图理论基础

图的背景、定义、性质、连接表示和类型。

背景

柯尼斯堡七桥问题是图论中的著名问题：在所有桥都只能走一遍的前提下，如何才能把所有桥都走遍？

欧拉将实际问题抽象化成平面的点和线，每一座桥代表一条线，桥所连接的区域视为点。若从某点出发再回到该点，则这一点的线数必须是偶数。

定义

图被记为 $G = \{V, E\}$

其中，$V = \{v_1, \ldots, v_N\}$是数量为 N**节点**的集合，$E = \{e_1, \ldots, e_M\}$是数量为 M的**边**的集合。

图用**节点**表示**实体**，用**边**表示实体间的**关系。**

根据节点和边的信息的类型，可分为两类：

- 类别型。数据取值只能是哪一类别，被称为**标签**
- 数值型。数据取值范围为实数，被称为**属性**

在图的计算任务重，认为节点一定含有信息，边可能含有信息。

根据边的性质图可以分为：

1. 根据图的**边是否具有指向性**：
    1. 有向图：边具有指向性
    2. 无向图：边不具有指向性
2. 根据图的**边是否具有权重**：
    1. 无权图：边不具有权重，或有权重且权重都为1
    2. 有权图：边具有权重，记点 v_i到点 v_j的权重为 w_{ij}

注意：上述两种分类可以相互叠加。例如，边既有指向性又有权重，即有向有权图

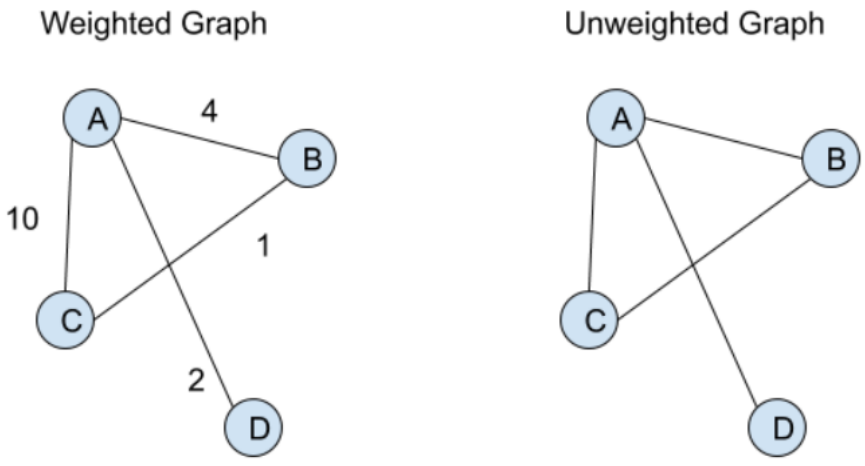


图1 有权图和无权图

性质

邻接节点

节点 v_i的邻接节点是与节点 v_i直接相连的节点，被记为 $N(v_i)$

节点 v_i的 k跳远的邻接节点是到节点 v_i要走 k步的节点（节点的2跳远的邻接节点，包含了自己本身）

度

节点 v_i的度记为 $d(v_i)$，入度记为 $d_{in}(v_i)$，出度记为 $d_{out}(v_i)$。

有向有权图：节点 v_i的出度等于从 v_i出发的边的权重之和；节点 v_i的入度等于从连向 v_i的边的权重之和

无向图：节点的出度与入度相等

无权图：边的权重都为1，节点 v_i的出度等于从 v_i出发的边的数量，节点 v_i的入度等于从连向 v_i的边的数量

平均度：是表达网络整体性质的重要参数。对于无向图，平均度为 $\bar{d}(G) = \frac{1}{N}\sum_{i=1}^{N} d_i = \frac{2M}{N}$

度分布： $P(d)$表示随机选择的节点的度为 d的概率，平均度为 $\bar{d}(G) = \sum_{i=0}^{\infty} dP(d)$

行走和路径

$walk(v_1, v_2) = (v_1, e_6, e_5, e_3, e_1, v_2)$这是一次行走，从节点 v_1出发，经过边 e_6, e_5, e_3, e_1，最终到达节点 v_2。

行走是**节点可重复**的

路径是**节点不可重复**的**行走**

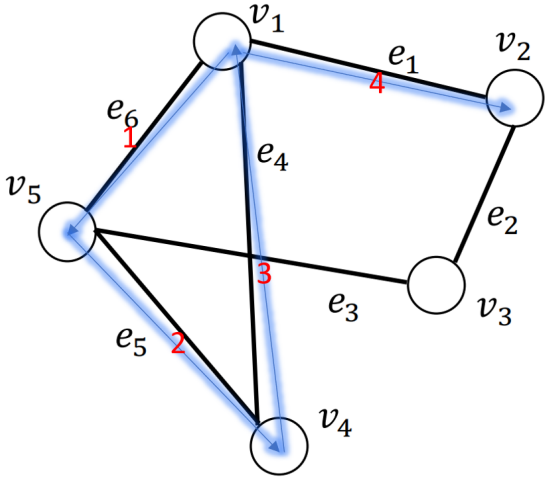


图2 行走

距离和直径

距离是两个点之间的**最短路径。**

$v_s, v_t \in V$是图上的一对节点，该对节点之间所有路径的集合记为 p_{st}。节点之间的最短路径 p_{st}^{sp}为集合中长度最短的路径最短路径定义为：

$$p_{st}^{sp} = argmin_{p \in P_{st}} |p|$$

其中， p表示集合中的一条路径， $|p|$是路径 p的长度

直径是图中所有节点对之间的**最短路径的最大值**，定义为：

$$diameter(G) = max_{v_s, v_t \in V} min_{p \in p_{st}} |p|$$

子图、连通分量和连通图

子图：有图 $G = \{V, E\}$，另有 $G' = \{V', E'\}$，其中 $V' \subset V$， $E' \subset E$，那么 $G' \subset G$，称 G' 是 G 的子图。

连通分量：子图中任意两节点都可以用路径相连，称该子图为图的连通分量

连通图：当一个图只包含一个连通分量，即本身，称该图为连通图

聚类系数

聚类系数表示给定节点的**邻居彼此连接的程度**，邻域互连越紧密，局部聚类系数越高。

C_i是节点两个邻居相互连接的概率，对于度为 d_i的节点 i，局部聚类系数定义为：

$$C_i = \frac{E_i}{T_i}$$

其中， E_i表示节点 i的邻居实际存在的边的数量， T_i表示节点 i的邻居最多存在的边的数量。

- $C_i = 0$：节点 i的邻居都没有相互连接
- $C_i = 1$：节点 i的邻居形成一个全连接图，即邻居都相互连接
- $C_i = 0.5$：节点的两个邻居有 50的概率连接

网络的聚类系数即**平均聚类系数**是所有节点的聚类系数的平均值，定义为：

$$C = \frac{1}{N} \sum_i C_i$$

接近中心度

连通图中，节点的**接近中心度**（或接近性）是网络中中心性的度量，是**该节点与图中所有其他节点之间的最短路径之和的倒数**，定义为：

$$c(v) = \frac{1}{\sum_{u \neq v} |p|}$$

节点越接近中心，它与所有其他节点越接近

连接表示

邻接矩阵

给定一个图，其对应的**邻接矩阵**记为 $A \in \{0, 1\}^{N \times N}$

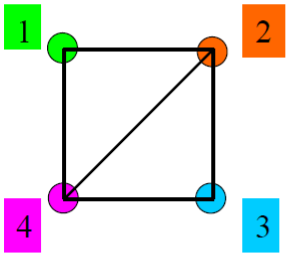
$A_{i,j} = 1$表示存在从节点 v_i到 v_j的边， $A_{i,j} = 0$则表示不存在

邻接矩阵描述的是**节点和节点**之间的关系

无向图：从节点 v_i到 v_j的边存在，则同时从节点 v_j到 v_i的边也存在，因此**邻接矩阵是对称的**

无权图：各条边的权重是等价的，即各条边的权重为1

有权图：对应邻接矩阵被记为 $W \in R^{N \times N}$，其中 $W_{i,j} = w_{ij}$表示从节点 v_i到 v_j的边的权重。若边不存在，则边的权重为0



$$A=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

(always square and symmetrical)

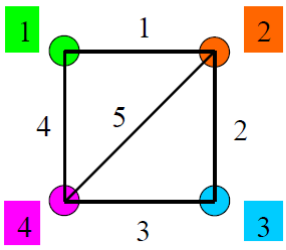
图3 无向无权图的邻接矩阵

关联矩阵

给定一个图，其对应的**关联矩阵**记为 $M \in \{0,1\}^{N\times m}$

$m_{i,j}=1$表示节点 v_i和边 e_j相连接， $m_{i,j}=0$表示节点 v_i和边 e_j不相连接

关联矩阵描述的是**节点和边**之间的关系



$$M=\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

(generally non-square)

图4 无向无权图的关联矩阵

拉普拉斯矩阵

给定一个图，其邻接矩阵为 A，其**拉普拉斯矩阵** L定义为：

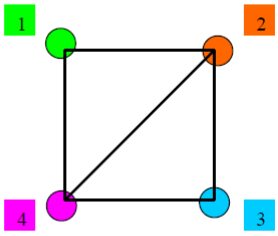
$$L=D-A$$

其中， $D=diag(d(v_1),\ldots,d(v_N))$是度矩阵

记拉普拉斯矩阵中每一个元素为 L_{ij}，那么每一个元素可以被定义为：

$$L_{ij}=\begin{cases} d_i, & if\ i=j \\ -1, & if\ i\neq j\ and\ v_i adjacent\ with\ v_j \\ 0, & otherwise \end{cases}$$

拉普拉斯矩阵的每一行和每一列的和为0



Degree matrix

$$L=\underset{D}{\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}}-\underset{A}{\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}}=\underset{L}{\begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}}$$

$$L=D-A$$

图5 无向无权图的关联矩阵

对称归一化的拉普拉斯矩阵：给定一个图，其邻接矩阵为 A，其归一化的拉普拉斯矩阵定义为：

$$L = D^{-\frac{1}{2}}(D-A)^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$$

类型

根据图的拓扑结构，规则网络可以分为：

- 全连接网络
- 环形网络
- 星形网络

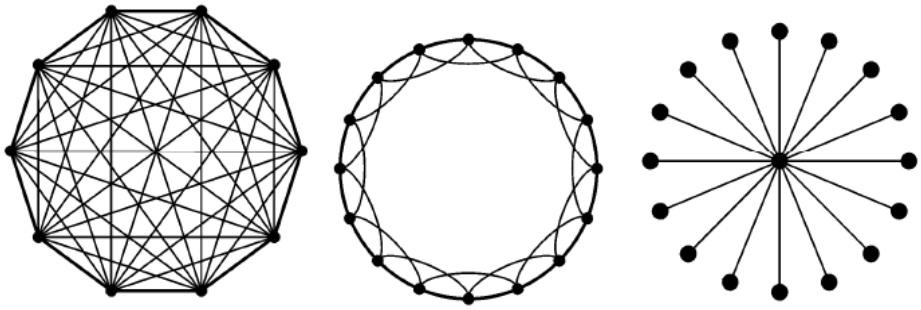


图6 图的拓扑结构

根据不同的性质，常见的图模型有：随机图、小世界图和无标度图模型

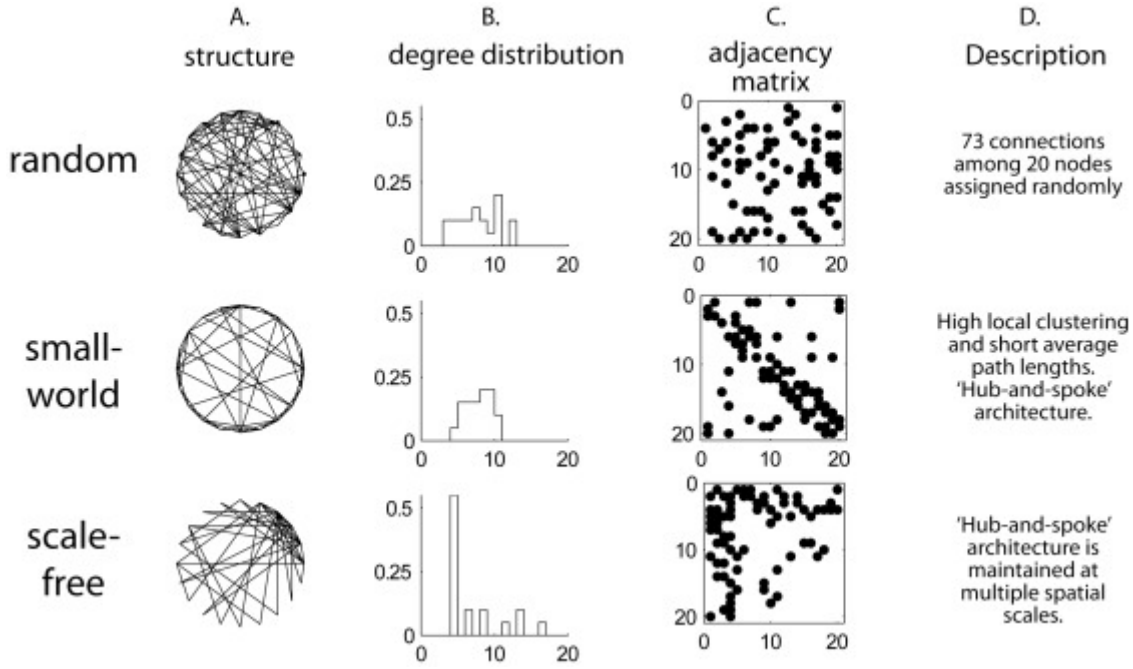


图7 常见的图模型

深度学习中同质图和异质图的定义：

- **同质图**：只有一种类型的节点和一种类型的边的图
- **异质图**：存在多种类型的节点和多种类型的边的图

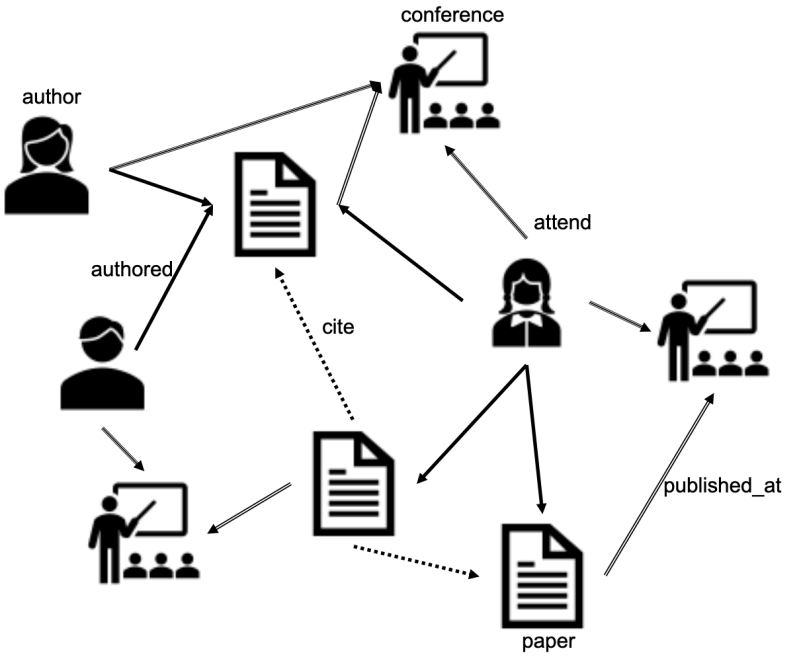


图8 异质图

二分图：节点分为两类，只有不同类的节点之间存在边。

二分图节点可以分为两个不相交的集合 U和 V，使得每个边将集合 U中的节点与集合 V中的节点连接

每个二分图可生成两个投影：

- U投影：如果两个 U节点连接到相同的 V节点，则在投影中连接该对 U节点
- V投影：如果两个 V节点连接到相同的 U节点，则在投影中连接该对 V节点

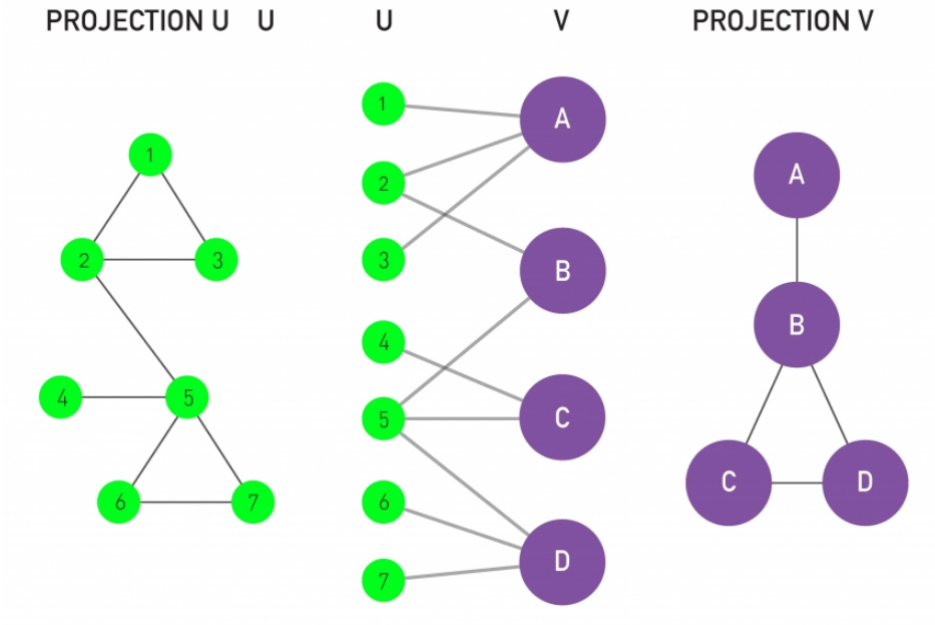


图9 二分图

深度学习基础

神经网络基本组成

结构

最常见的朴素神经网络一般只**多层感知机**

由于没有反馈回路，只有前馈路径，也被称为**前馈网络**

多层感知机包含**输入层**、**隐藏层**和**输出层**

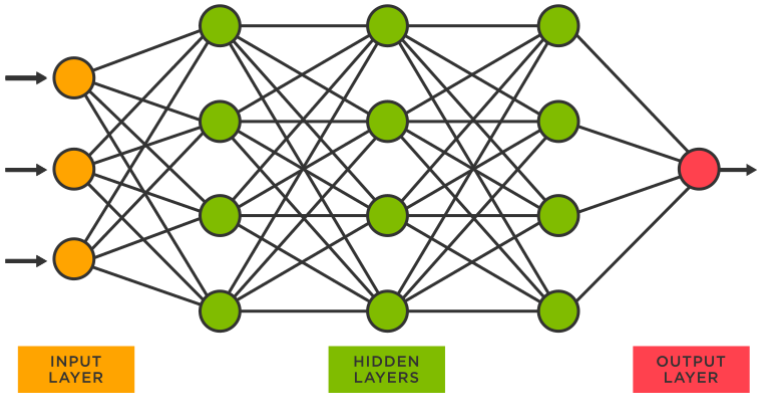


图10 多层感知机

去掉多层感知机的隐藏层，就是最初的单层感知机模型

假设矩阵 $X \in R^{n \times d}$是输入矩阵，其中 d是输入的维度， n是输入样本的数量。

单层感知机是对输入矩阵进行权重 $W \in R^{d \times h}$和偏置 $B \in R^{1 \times h}$线性加权，再通过激活函数 σ得到输出 $\hat{Y} \in R^{n \times q}$，写作

$$\hat{Y} = \sigma(WX + B)$$

将单层所得到的输出 $\hat{Y}$写作 $f^{(l)}$，增加到 L个隐藏层后的多层感知机写作

$$\hat{Y} = f^{(L)}(\dots f^{(l)}(\dots f^{(2)}(f^{(1)(X)})))$$

激活函数判断神经元是否应该被激活，常见的激活函数有：

- Sigmoid函数：仅适用于二分类
- Tanh函数：不常使用
- ReLU函数：一般情况下优先使用

输出层与损失函数

面对不同的问题，输出层和损失函数的选择也不同

假设神经网络前一层网络的输出为 h，输出层 L 的表达式为

$$\hat{y} = \sigma(W^{(L)}h + B^{(L)})$$

回归问题

一般的回归问题，可以认为激活函数是一个恒等映射

可以以 $\hat{y}$和 y的平方差作为损失函数

二分类问题

而分类问题，假定其输出是0或1，因此可以使用sigmoid函数控制输出的范围在0~1之间

使用交叉熵作为损失函数计算

最后使用阈值将输出的 $\hat{y}$转换成二元类别标签

多分类问题

对于多分类或 k分类问题，输出可以被定义为一个独热编码 $y \in \{0,1\}^k$，其中第 j个元素 $y_j = 1$表示这个样本的标签为 j。为了将输出转化为类似的分布，使用softmax函数对输出进行标准化

$$\hat{y}_j = softmax(z)_j = \frac{exp(z_j)}{\sum_k exp(z_k)}$$

使用交叉熵作为损失函数

模型优化

模型优化的算法有：动量法、AdaGrad法、RMSProp法、SGD法、Adam法等

其中最常使用的优化算法是：**SGD**和**Adam**

SGD

目标函数通常是训练数据集每个样本损失函数的均值，给定 n个样本的训练数据集

目标函数

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

梯度计算为

$$\nabla f(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x)$$

梯度下降法，时间复杂度为 $O(n)$，因此训练集庞大时，时间复杂度将很大

SGD可以降低计算代价，每次迭代随机对数据集随机均匀采一个小批量，以计算梯度

$$x \leftarrow x - \eta \nabla f_i(x)$$

Adam

Adam是一种自适应学习率的算法，结合了动量法和RMSProp法的优点

它使用指数加权移动平均值来估计梯度的动量和二次矩，即使用状态变量

$$v_t \leftarrow \beta_1 v_{t-1} + (1-\beta_1) g_t$$
$$s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2) g_t^2$$

β_1和 β_2是非负加权参数，通常设置为 $\beta_1 = 0.9$和 $\beta_2 = 0.999$

标准化状态变量由下式获得

$$\hat{v}_t = \frac{v_t}{1-\beta_1^t} \; and \; \hat{s}_t = \frac{s_t}{1-\beta_2^t}$$

重新缩放梯度为

$$g_t' = \frac{\eta \hat{v}_t}{\sqrt{\hat{s}_t} + \epsilon}$$

通常 $\epsilon = 10^{-6}$

$$x_t \leftarrow x_{t-1} - g_t'$$

过拟合和欠拟合

欠拟合：训练误差和验证误差都很大

- 模型过于简单，表征能力不够。更换复杂的模型
- 优化器设置不当（错误的学习率）

过拟合：训练误差很小但验证误差很大

- 模型表征能力过强
- 训练数据集太少

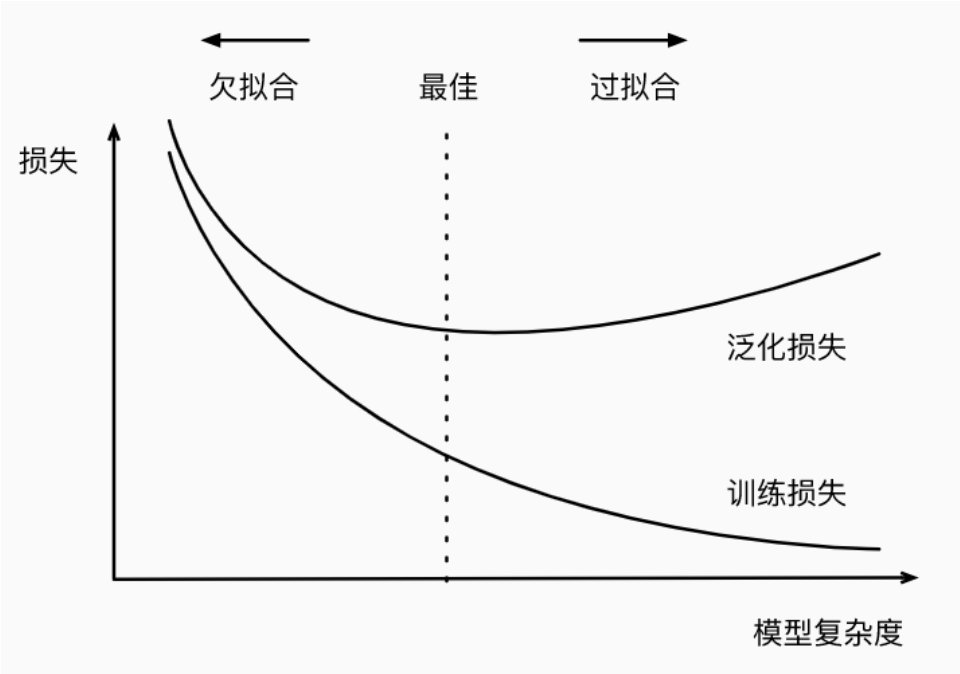


图11 欠拟合与过拟合

当小于最佳模型复杂度时：增加模型复杂度，可以降低训练损失和泛化损失，欠拟合

当大于最佳模型复杂度时：增加模型复杂度，只能降低训练损失，并且泛化损失增加，过拟合

正则化

深度学习所需的参数量庞大，因此过拟合问题的解决更为重要，通常采用正则化来缓解过拟合的问题

权重衰退

权重衰退是最常使用的正则化之一，通常也被称为 L_2正则化

通常将权重的范数作为惩罚项加到损失函数中，并使用非负超参数 λ来权衡

$$L(w,b)+\frac{\lambda}{2}||w||^2$$

暂退法Dropout

Dropout是一种实用的Bagging方法，在每个训练批次中，只使用一部分神经元，将每个批次的结果集成起来作为最后的结果。因此，可以减少模型复杂度，提高泛化能力

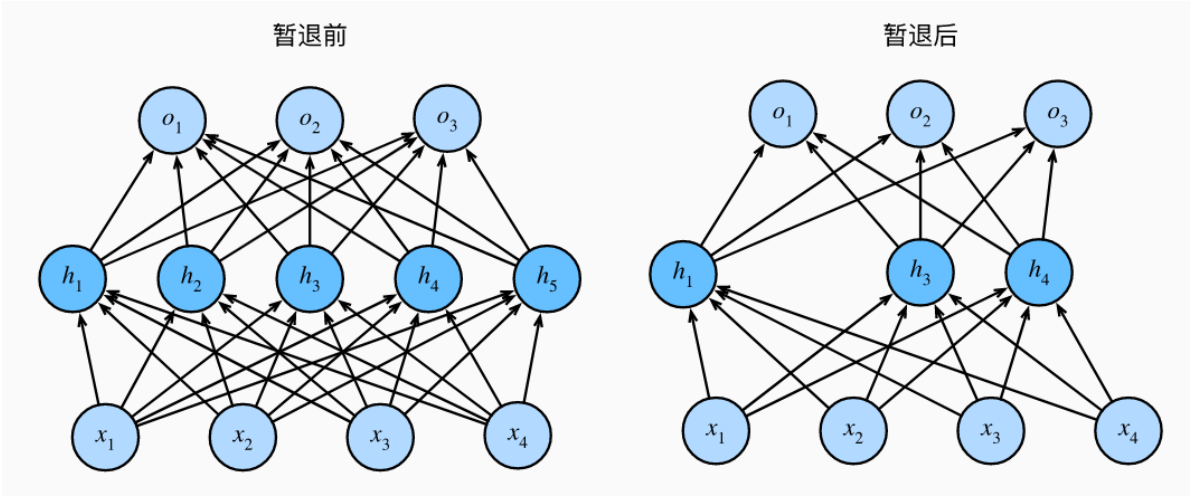


图12 Dropout

前向传播和反向传播

前向传播：按输入层到输出层的顺序计算和存储网络中每层的结果，对应的目标函数是带权重衰减的损失函数

反向传播：计算网络中所需要更新参数的梯度，采用链式法则，一次计算每个中间变量和参数的梯度

卷积神经网络

卷积神经网络是包含卷积层的特殊的神经网络

检测对象被称为卷积核或滤波器，或者是称为该卷积层的权重

卷积神经网络具有以下特性：

1. 平移不变性：神经网络前几层卷积核对相同的图像区域具有相似的反应。
2. 局部性：前几层只探索输入图像的局部区域，不在意相隔较远的区域，最终可以聚合局部特征

卷积

设两个函数 $f, g : R^d \to R$的卷积定义为：

$$(f * g)(x) = \int f(z)g(x - z)dz$$

图像卷积

实际图像卷积使用的是互相关进行计算，而不是数学定义的卷积

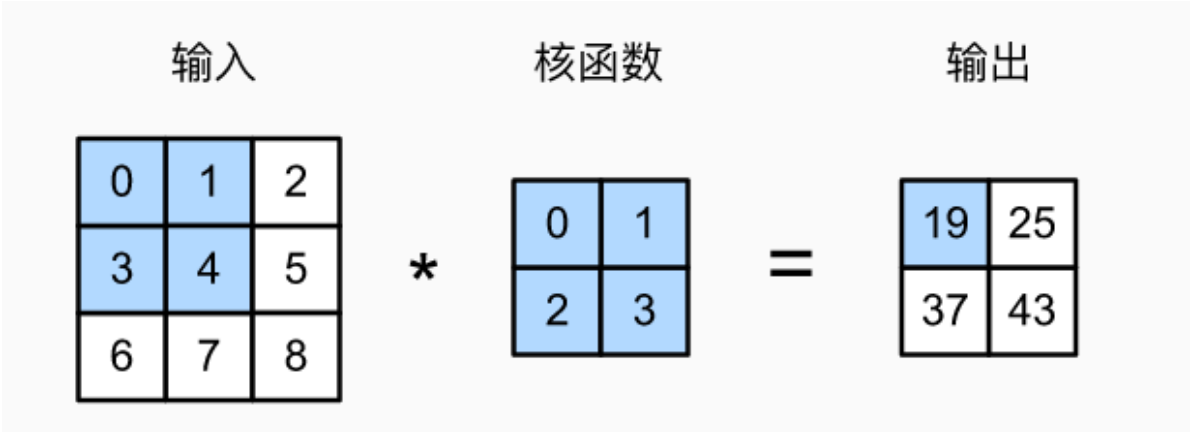


图13 卷积计算

输出的大小会小于输入的大小

填充

输出图像缩小的尺寸由卷积核的大小决定

而在较深的神经网络中最终的输出图像较原始的输入图像相比会丢失掉许多边界信息，填充是解决该问题的有效方法

填充：在输入图像的边界填充元素，通常为填充的元素是0

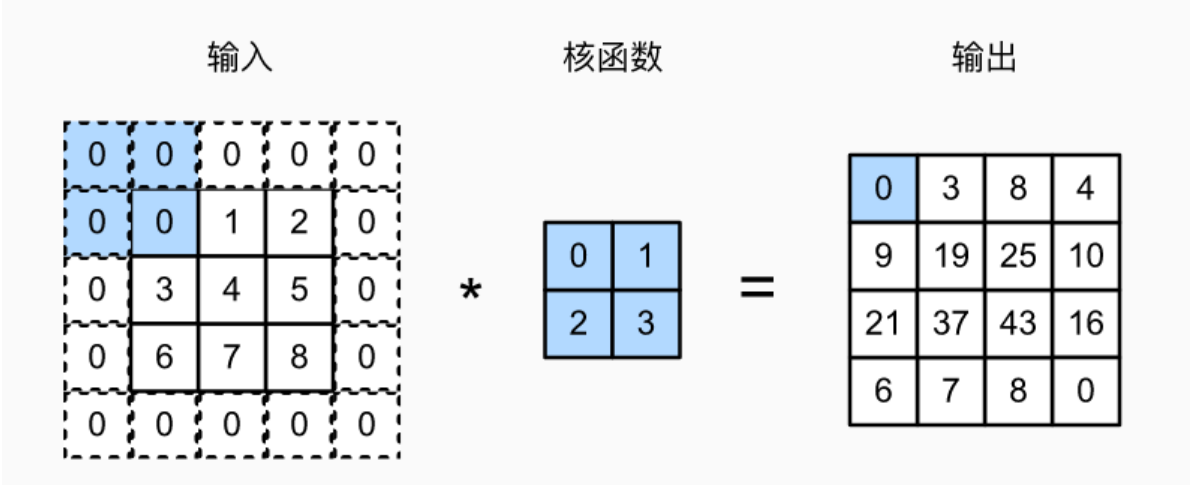


图 14 填充

步幅

为了高效计算或是缩减采样次数，卷积核每次卷积运算后滑动多个元素

每次滑动的元素数量称为步幅

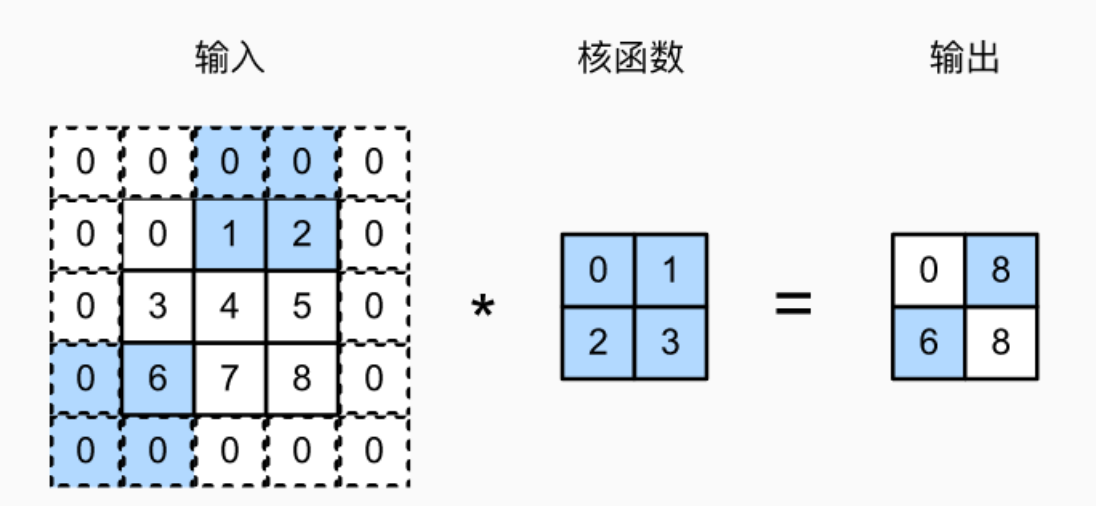


图15 步幅

特征映射与感受野

输出的卷积层有时称为特征映射，它可以被视为一个输入映射到下一卷积层的转换器

对于某一层任意元素，其感受野是指前向传播来自所有先前层可能影响计算的所有元素

越深层的神经元感受野也越大

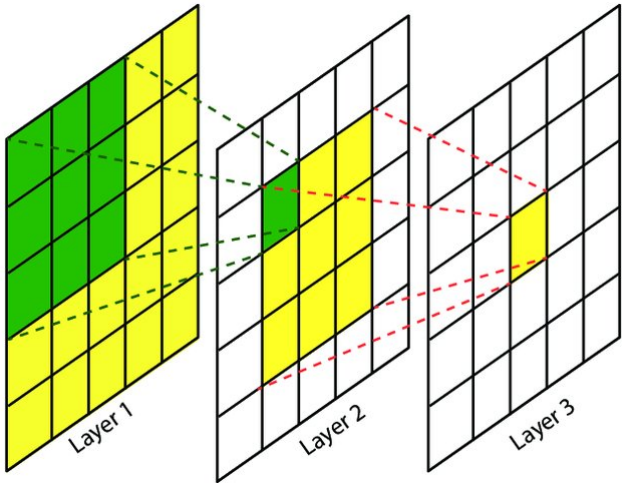


图16 感受野

池化

处理图像时，希望逐渐降低隐藏表示的空间分辨率，从而聚集信息使每个神经元的感受野增大

池化层：降低卷积层对位置的敏感性，同时降低空间采样表示的敏感性

常用的池化层有：最大池化层和平均池化层

循环神经网络

循环神经网络具有循环连接，可以将前面的信息传递到后面的步骤，捕获序列数据的时序关系

网络架构

假设在时间步 t时有小批量输入 $X_t \in R^{n\times d}$，每一行对应于来自该序列时间步 t处的一个样本

用 $H_t \in R^{n\times h}$表示时间步 t的隐状态，由当前时间步的输入与前一个时间步的隐状态一起计算得出

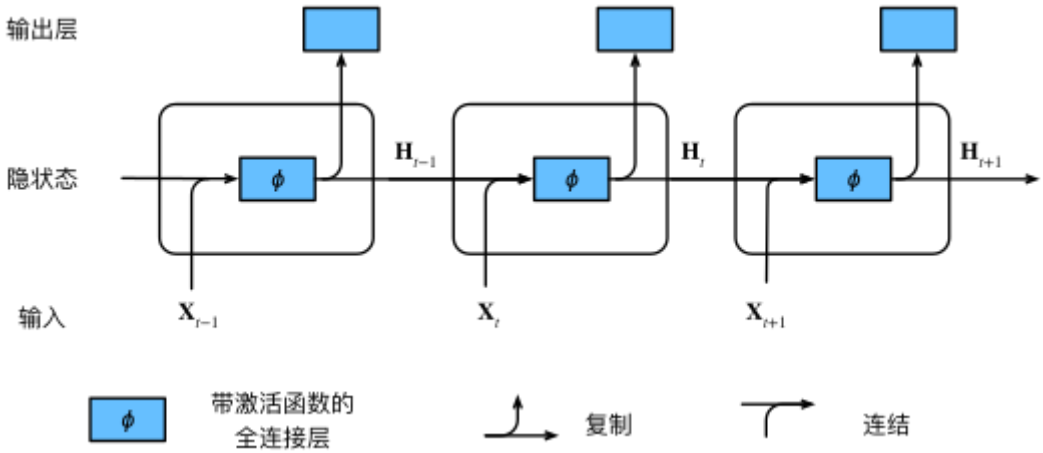


图17 循环神经网络

在不同的时间步，循环神经网络使用的参数相同。因此，循环神经网络的参数开销不会随着时间步的增加而增加。

循环神经网络的经典变体包括

- **长短期记忆网络（LSTM）**
- **门控循环单元（GRU）**

通过引入门控机制来解决传统循环神经网络中的梯度消失和梯度爆炸问题，从而改善了模型的长期依赖建模能力。

反向传播使用通过时间反向传播（ BPTT）

通过将时间展开的 RNN 视为深度前馈神经网络，并在每个时间步骤上应用标准的反向传播算法来更新模型的权重。

图表示学习

图**表示学习**或**嵌入**，指的是将图中的每个节点映射到低维空间

节点表示学习

节点嵌入是对节点进行编码，使低维空间的相似性与原始图中的相似性相近

节点嵌入算法有三个基本阶段：

1. 定义编码器。将节点映射到低维向量
2. 定义相似度函数。节点嵌入的评价指标
3. 优化编码器参数。使得低维空间的相似性

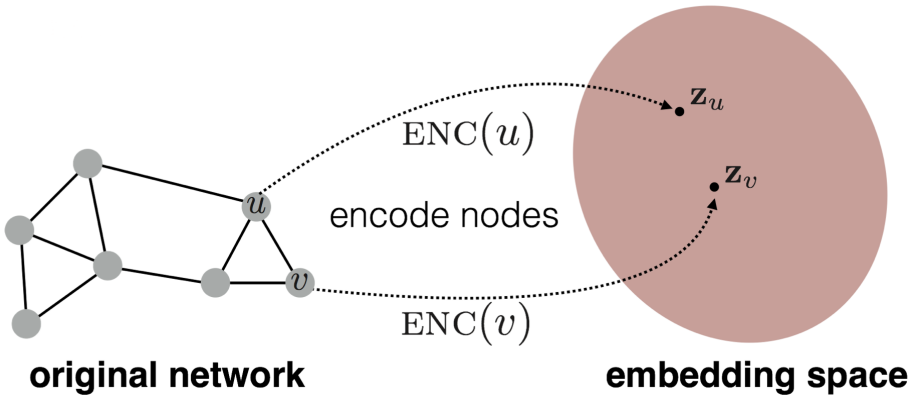


图18 节点嵌入

随机游走是节点嵌入的主要方法

- 给定一个图和一个起点，随机选择该节点的邻接节点，并移动到该邻接节点
- 随机选择该邻接节点的邻接节点，并继续移动，以此类推

最终得到一条游走的节点序列

在随机游走中，相似度定义为节点 u和 v同时出现的概率，即 $similarity(u,v) = z_u^T z_v$

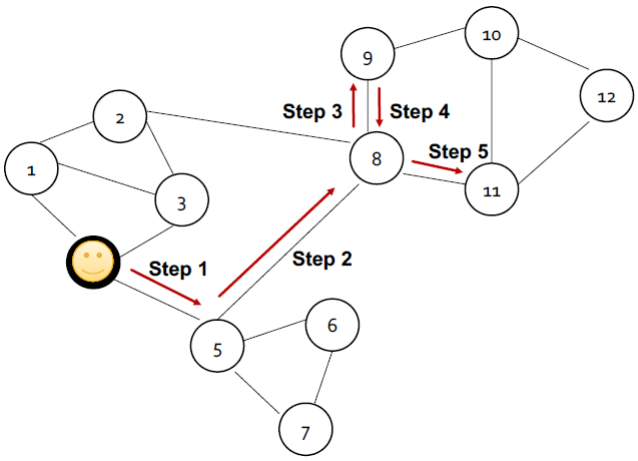


图19 随机游走 ### 深度游走

深度游走是指运行长度固定、无偏的随机游走，是最简单的随机游走

深度游走的步骤如下：

1. 从起始节点 u开始采用随机游走策略 R行进，得到邻接节点为 $N_R(u)$。
2. 要使 $N_R(u)$的相似度高，进行嵌入的优化

定义损失函数为：

$$L = \sum_{u \in V} \sum_{v \in N_R(u)} -log(P(v|z_u))$$

其中后验概率可以使用softmax表示：

$$L = \sum_{u \in V} \sum_{v \in N_R(u)} -log(\frac{exp(z_u^T z_v)}{\sum_{n \in V} exp(z_u^T z_n)})$$

上述损失函数的计算复杂度过高，引入负采样，一个表示所有点的随机概率 P_v

$$log(\frac{exp(z_u^T z_v)}{\sum_{n \in V} exp(z_u^T z_n)}) \approx log(\sigma(z_u^T z_v)) - \sum_{i=1}^{k} log(\sigma(z_u^T z_{n_i})),\ n_i \sim P_v$$

Node2Vec

Node2Vec通过图上的广度优先遍历和深度优先遍历，在图的局部和全局之间进行权衡

- 广度优先遍历，可以提供局部的邻域
- 深度优先遍历，可以提供全局的邻域

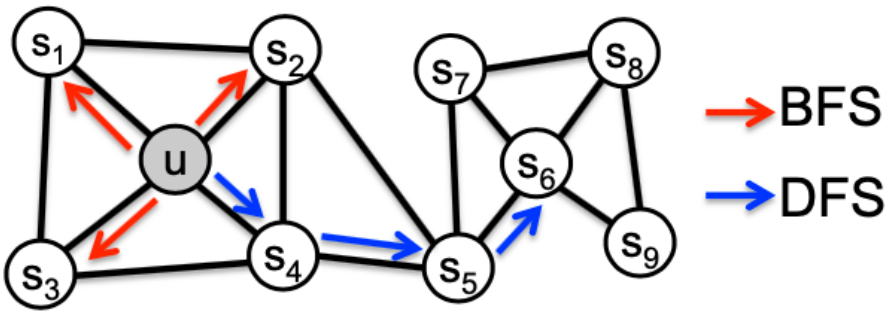


图20 广度优先遍历和深度优先遍历

定义 p代表返回前一个节点的概率， q定义广度优先搜索和深度优先搜索的比率

Node2Vec步骤如下：

1. 计算随机游走的概率
2. 模拟 r个从节点 u开始长度为 l的随机游走
3. 使用随机梯度下降优化

图表示学习

将整个图映射到低维空间：

1. 在图上使用节点嵌入，对图的节点嵌入求和或平均值
2. 引入虚拟节点来表示图，使用节点嵌入
3. 使用匿名游走嵌入，枚举所有可能的匿名游走，将图表示为游走的概率分布

图卷积网络

现有的图神经网络都基于邻居聚合的框架：每个节点通过聚合其邻居刻画结构信息

在建模图神经网络时，**研究重点**是：如何在网络上构建聚合算子

现有的构建聚合算子的方法可分成**空间域**和**谱域**

谱域图卷积神经网络

谱域图卷积神经网络主要包括谱卷积神经网络、切比雪夫网络和图卷积神经网络

谱图理论和图卷积

卷积的傅里叶变换

$$F(f*g) = F(f)\cdot F(g))$$

傅里叶逆变换

$$f*g = F^{-1}(F(f)\cdot F(g))$$

图傅里叶变换

依赖于图上的拉普拉斯矩阵 L，对 L作谱分解

$$L = U\Lambda U^T$$

其中 $\Lambda = diag(\lambda_0,\dots,\lambda_{N-1}) \in R^{N\times N}$是特征值矩阵， $U = [u_0,\dots u_{N-1}] \in R^{N\times N}$是对应的特征向量矩阵

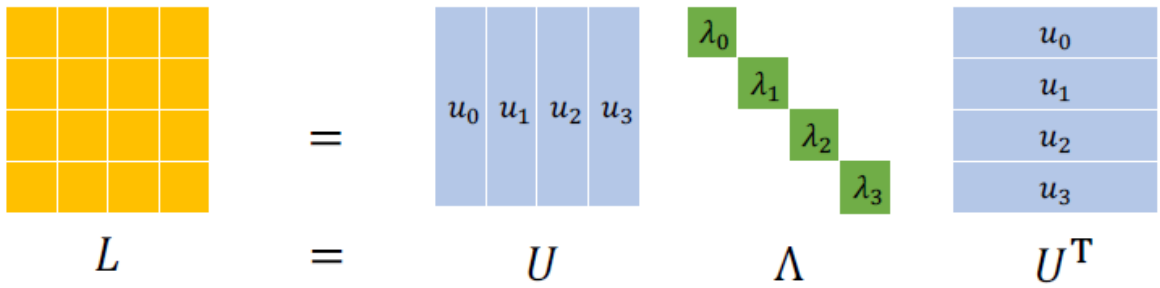


图21 谱分解

以特征向量作为谱空间下的一组基底，图上信号 x的傅里叶变换为：

$$\hat{x} = U^T x$$

其中 x是信号在节点域的原始表示， $\hat{x}$是信号 x变换到谱域后的表示， U^T表示特征向量矩阵的转置，用于傅里叶变换

信号 x的傅里叶逆变换为：

$$x = U\hat{x}$$

图卷积

图卷积可分为三步：

1. 将图傅里叶变换到谱域
2. 在谱域空间进行卷积计算
3. 对计算的结果进行傅里叶逆变换回空间域

$$y = g_\theta(U\Lambda U^T)x = g_\theta(L)x$$

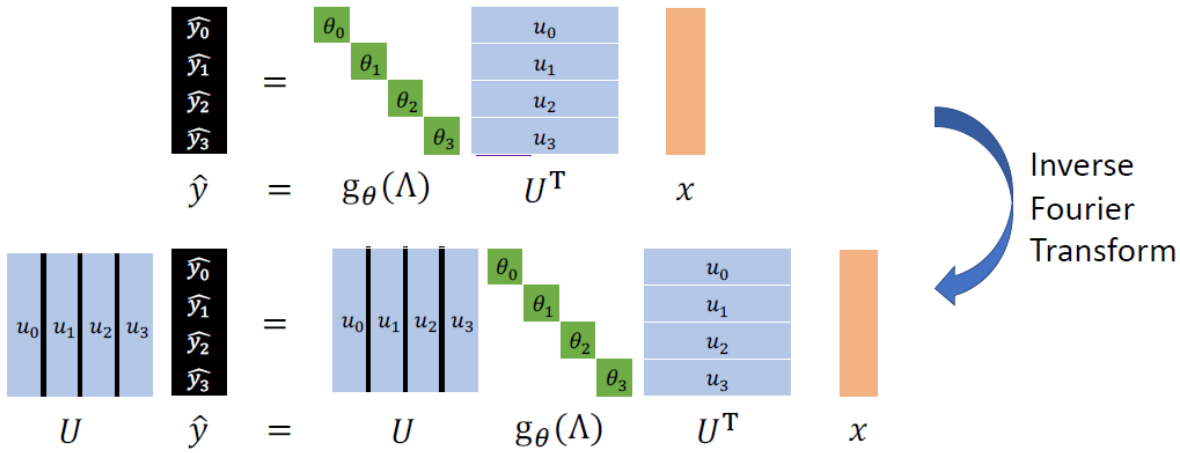


图22 图卷积 ### 谱卷积神经网络

将上式的 x和 y替换为图上节点特征 $H^{(l)}$和 $H^{(l+1)}$，第 l层的结构：

$$H^{(l+1)} = \sigma(g_\theta(U\Lambda U^T)H^{(l)})$$

切比雪夫网络

谱卷积神经网络基于全局的傅里叶来计算，难以保证局部性

计算复杂度高，难以应用于大型图网络结构

切比雪夫网络采用切比雪夫多项式替代谱卷积网络的卷积核

g_θ是需要学习的卷积核，在谱卷积神经网络中， g_θ为对角阵，有 n个参数需要学习

$$g_\theta = \sum_{i=0}^{K-1} \theta_k T_k(\hat{\Lambda})$$

其中 θ_k是需要学习的系数，定义 $\hat{\Lambda} = \frac{2\Lambda}{\lambda_{max}} - I_n$

切比雪夫的递归表达式为：

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

其中初始值 $T_0(x) = 1, T_1(x) = x$

图卷积神经网络

图卷积神经网络对切比雪夫网络进行简化，只取了0阶和1阶：

$$y = g_\theta(L)x = \sum_{i=0}^{1} \theta_k T_k(\hat{\Lambda}) = \theta_0 x + \theta_1 \hat{L}x$$

代入定义 $\hat{\Lambda} = \frac{2L}{\lambda_{max}} - I$，且 $\lambda_{max} = 2$可得

$$y = \theta_0 x + \theta_1 (L - I)x$$

此时拉普拉斯矩阵 L为标准化后的拉普拉斯矩阵，满足 $L = I - D^{-1/2}AD^{-1/2}$，令 $theta = \theta_0 = -\theta_1$可得：

$$y = \theta x - \theta(D^{-1/2}AD^{-1/2})x = \theta(I + D^{-1/2}AD^{-1/2})x$$

空间域图卷积神经网络

图卷积神经网络的空域理解

从邻居节点信息聚合的角度，图卷积神经网络作用如下：

- 对节点信息进行转换
- 对节点信息进行聚合

公式可表示为：

$$h_v^{l+1} = \sigma(W^l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + h_v^{(l)} B^{(l)})$$

第一项 $W^l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|}$表示邻居节点信息的转换和聚合，第二项 $h_v^{(l)} B^{(l)}$表示自身节点信息的变换

空域图卷积的统一范式和GraphSAGE

图卷积的统一范式

$$h_v^{l+1} = AGG_2^l(AGG_1^l(\{TRANS_u^{(l)}(h_u^{(l)}, u \in N(v))\}), TRANS_v^{(l)}(h_u^{(l)}))$$

其中 $TRANS_u^{(l)}$表示对邻居节点信息的转换， $TRANS_v^{(l)}$表示对自身节点信息的转换， AGG_1^l表示对邻居节点信息的聚合， AGG_2^l表示对自身节点信息的聚合

GraphSAGE

对传统的图卷积神经网络的两个方面进行了改进：

1. 在训练时，将图卷积神经网络的全图采样优化为部分以节点为中心的邻居抽样
2. AGG聚合函数可以使用平均、MaxPooling、LSTM

GraphSAGE运用mini-batch的步骤分为三步：

1. 对邻居进行随机采样，每跳抽样的邻居数不多于 S_k个
2. 生成目标节点的嵌入：先聚合二跳邻居的特征，生成一跳邻居的嵌入，再聚合一跳的嵌入，生成目标节点的嵌入
3. 将目标节点的嵌入输入全连接网络得到预测值

关系图卷积神经网络

异质图和知识图谱

同质图：图中的节点类型和关系类型都只有一种。所有节点具有相同类型，所有边具有相同性质

异质图：图中的节点类型或关系类型多于一种。节点可以有不同的类型，边可以表达不同的关系

数学上，异质图可以定义为 $G = (V, E, R, T)$

其中 $v_i \in V$表示节点， $r \in R$表示关系类型， $(v_i, r, v_j) \in E$表示连接节点 v_i和 v_j的边的关系 r， $T(v_i)$表示节点类型

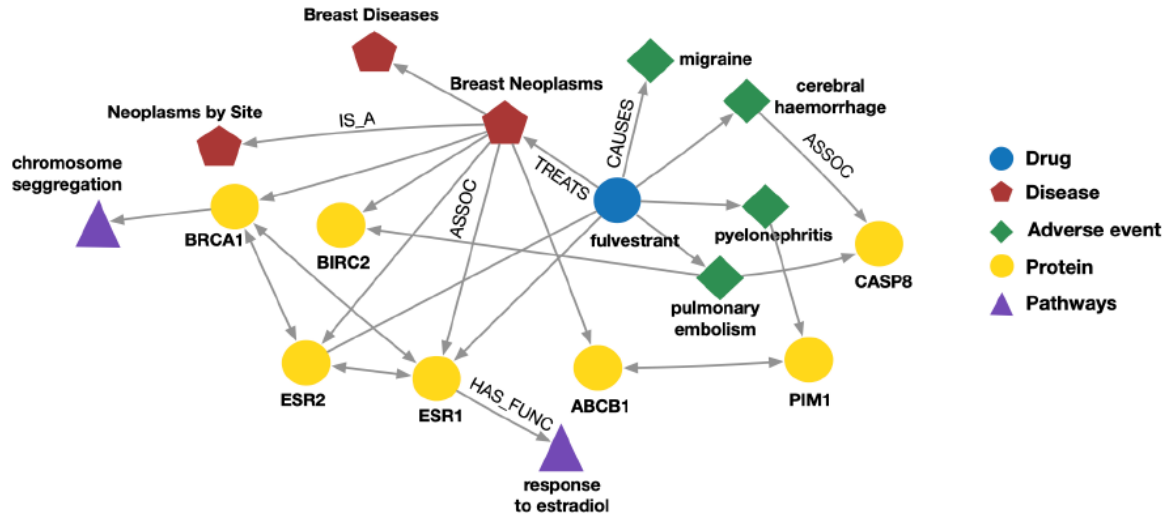


图23 生物医学异质图 **知识图谱**大多是异质图

知识图谱包含实体和实体之间的关系，并以三元组的形式存储<头实体，关系，尾实体>

知识图谱常不完整，需要对确实的信息进行补全

知识图谱补全有两种任务：链路预测和实体分类

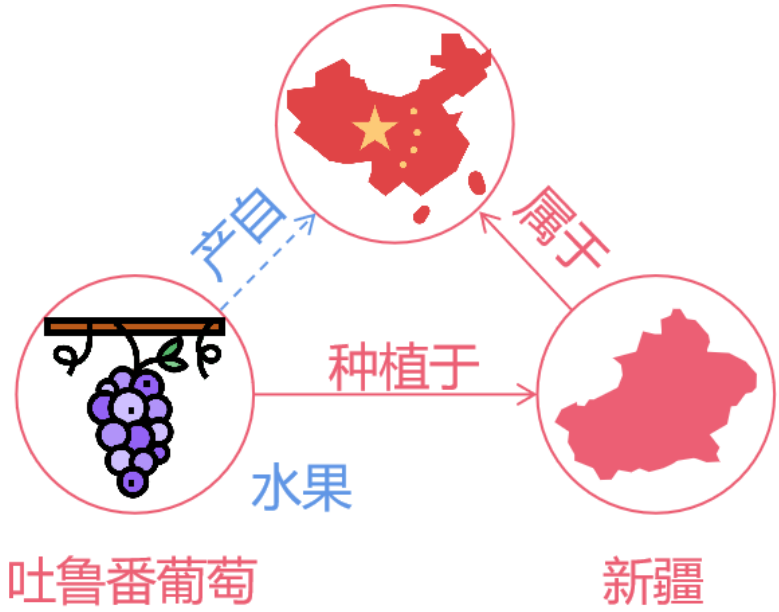


图24 知识图谱片段

处理好知识图谱实体间的不同关系，可以预测出节点的标签或实体与实体间的链路信息

关系图卷积神经网络

将一个复杂的异质图解耦成多个单一关系下的同构图，再使用同构图的方法解决

在图卷积神经网络中，第 $l+1$层中第 i个节点的隐藏向量 $h_i^{(l+1)}$计算方式如下：

$$h_i^{(l+1)} = \sigma(\sum_{j \in N_i} \frac{1}{c_i} W^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)})$$

其中 c_i是归一化常量， σ是激活函数， $W^{(l)}$是第 l层的可学习参数， N_i是中心节点 i的邻居索引集

关系图卷积神经网络

同质图中，图卷积神经网络权重参数共享

异质图中，需要为每一种关系类型 r学习一个 $W_r^{(l)}$

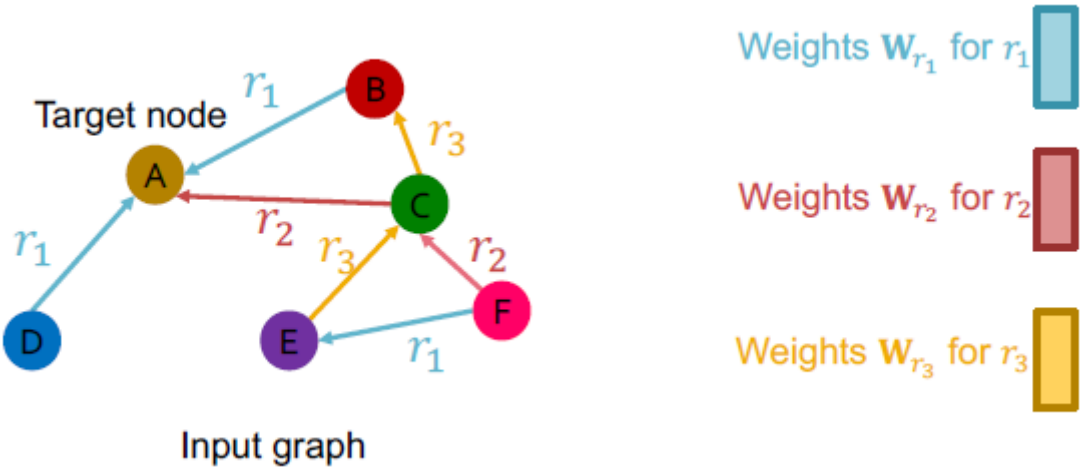


图25 异质图卷积神经网络核心思想

异质图卷积神经网络的层间递推关系为：

$$h_i^{(l)} = \sigma(\sum_{r\in R}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)})$$

其中 N_i^r表示关系 r下节点 i的邻居索引集，其中 $c_{i,r}$是针对不同问题下的一个归一化常量

如果边是有向的，方向也可以作为一种关系类型

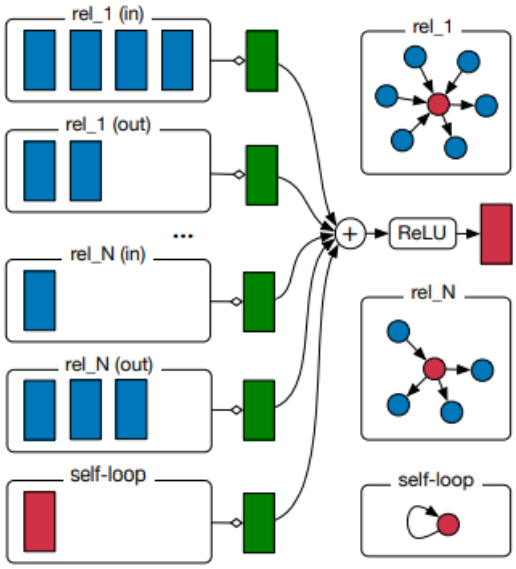


图26 单个节点更新的计算图 ## 可学习参数正则化

直接应用公式计算，模型参数量会随着关系的增多而变大，容易对罕见的边过拟合

减少模型参数量并缓解过拟合有两种方案：

1. 基底分解
2. 块对角矩阵分解

基底分解

假设第 l层关系 r下的权重 $W_r^{(l)}$可以被分解为基底 $\{V_1^{(l)}, V_2^{(l)}, \ldots, V_b^{(l)}, \ldots, V_V^{(l)}\}$的线性组合，即

$$W_r^{(l)} = \sum_{b=1}^{B} c_{rb}^{(l)} V_b^{(l)}$$

其中， $V_b^{(l)} \in R^{d^{(l+1)} \times d^{(l)}}$是基底矩阵， B表示基底的数量， $c_{rb}^{(l)}$表示基底 $V_b^{(l)}$对于关系 r的重要程度

通过基底分解，可以讲原本的 R个 $W_r^{(l)}$学习任务简化为：

1. 一组对于所有关系都通用的基底 $V_b^{(l)}$
2. R组用于组合基底和对应关系的系数 $c_{rb}^{(l)}$

基底分解可以看作不同关系类型之间的参数共享

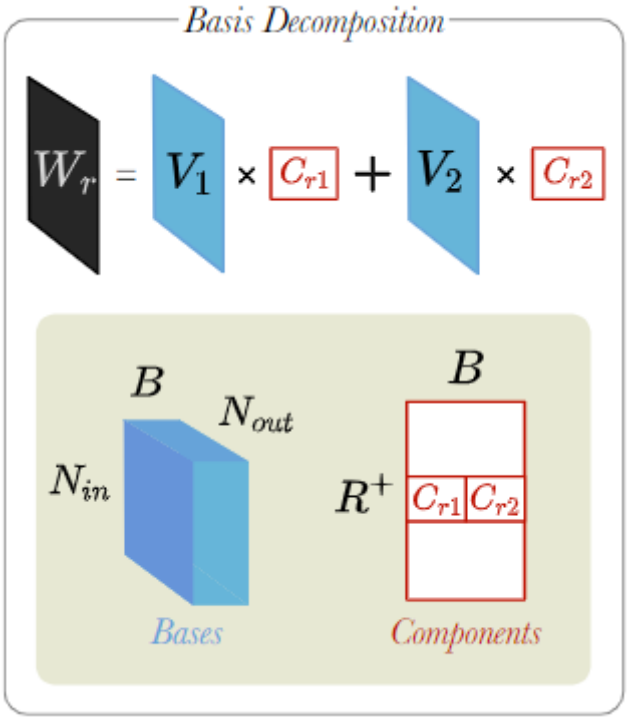


图27 基底分解示意图 ### 块对角矩阵分解

块对角矩阵分解的目的是：通过学习多个对角块矩阵，让权重稀疏，减少需要学习的参数量

令块对角矩阵的直接家和表示 W_r，即：

$$W_r^{(l)} = \bigoplus_{b=1}^{B} Q_{b,r}^{(l)}$$

其中 $\bigoplus$表示矩阵的直接加和， $W_r^{(l)}$可以表示为：

$$W_r^{(l)} = \begin{bmatrix} Q_{1,r}^{(l)} & 0 & \ldots & 0 \\ 0 & Q_{2,r}^{(l)} & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \ldots & Q_{B,r}^{(l)} \end{bmatrix}$$

块对角矩阵将 $W_r^{(l)}$划分为 B个大小的 $\frac{d^{(l+1)}}{B} \times \frac{d^{(l)}}{B}$小块，所以参数量降低到 $B \times \frac{d^{(l+1)}}{B} \times \frac{d^{(l)}}{B}$

图注意力网络

同质图中的注意力网络

注意力网络

在图卷积神经网络聚合节点的嵌入表征时，每个邻居节点的贡献程度是相同的，但实际上每个邻居节点对目标节点的重要性是有差异的。

图注意力网络的核心：对于每个顶点都计算其与邻居节点的注意力系数，通过注意力系数来聚合节点的特征

单头注意力机制

计算目标节点 i的一阶邻居节点 j的注意力权重 e_{ij}

将节点特征 h_i和 h_j通过 W映射到高维空间，并将这两个向量拼接在一起，通过全连接层 a_{fc}将拼接后的向量映射到一个标量，再使用 $LeakeyReLU$激活函数得到注意力系数 e_{ij}

$$e_{ij} = LeakeyReLU(a_{fc}(Wh_i||Wh_j))$$

其中 $LeakeyReLU(x) = max(0.01x, x)$

假设目标节点 i有 k个邻居节点 j，需要用softmax对注意力系数进行归一化得到 α_{ij}

对邻居节点特征进行线性组合，得到目标节点特征

$$h_i' = \sigma(\sum_{j \in N_i} \alpha_{ij} W h_j)$$

多头注意力机制

为了增强模型的表达能力和稳定性，使用多头注意力对每个目标节点执行 K次独立的单头注意力计算

将 K个 W^k将原特征进行映射到 K个不同的高维空间

重复上述单头注意力计算，得到 K个归一化后的注意力系数 α_{ij}^k，对 K组注意力系数和特征线性组合拼接

$$h_i' = concat_{k=1}^{K}(\sigma(\sum_{j \in N_i} \alpha_{ij}^k W^k h_j))$$

或对特征求平均值

$$h_i' = \sigma(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in N_i} \alpha_{ij}^k W^k h_j)$$

异质图中的图注意力网络

异质图注意力网络

异质图注意力网络使用了元路径来确定异质图中的每个节点的邻居节点

元路径

异质图节点之间存在复杂的联系，但这些联系并不全是有意义的

元路径是一种具有一定语义信息的构图方法，来定义一些有意义连接

节点 i在通过元路径生成的图中的邻居就是依据元路径定义的邻居

异质图注意力网络架构

将图神经网络中注意力机制从同质图拓展到节点和边类型不同的异质图，异质图注意力网络包括：

1. **节点级注意力**的目的是学习节点与基于元路径的邻居节点之间的重要性
2. **语义级注意力**的目的是学习不同元路径的重要性
3. 通过多层感知机输出节点 i的预测 $\tilde{y}_i$

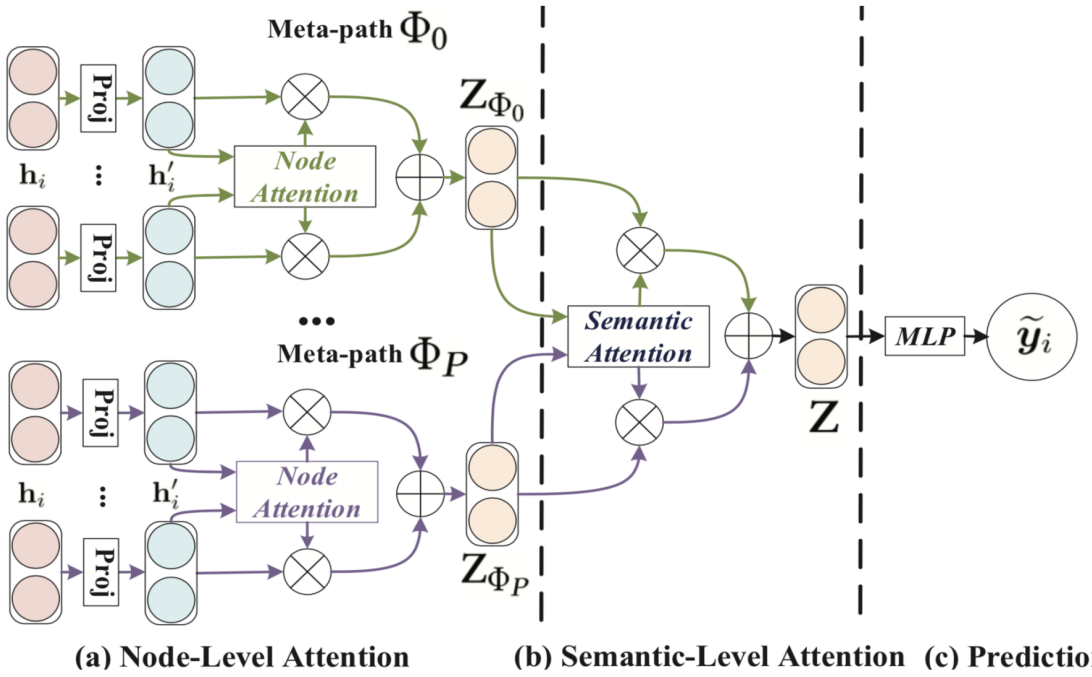


图28 异质图注意力网络

节点级注意力

节点 i的每一条元路径都会产生一个基于元路径的邻居节点的子图。假设节点i有$P+1$条元路径，对每一条元路径的邻居节点子图进行同质图卷积网络注意力计算。记Φ_j为第j条元路径，可以得到$P+1$个输出$\{Z_i^{\Phi_0},Z_i^{\Phi_1},Z_i^{\Phi_2},\cdots,Z_i^{\Phi_P}\}$

语义级注意力

每条元路径代表不同的语义信息，需要对多条元路径的特征进行语义融合

初始化一个语义级别的可学习向量q，然后用q和每个元路径的节点级特征的高维映射$WZ_i^{\Phi_P}$求内积，平均所有的节点i后，得到元路径p的语义级注意力系数

$$w_{\Phi_P}=\frac{1}{|V|}\sum_{i\in V}q^T\cdot tanh(WZ_i^{\Phi_P}+b)$$

使用softmax对P个注意力系数进行归一化得到β_{Φ_P}

对P个元路径得到的特征，通过归一化后的注意力系数进行加权求和，得到最终的节点i的输出特征Z_i

$$Z_i=\sum_{p=1}^{P}\beta_{\Phi_p}Z_i^{\Phi_p}$$