图卷积神经网络温和入门

李孙博闻

武汉科技大学
理学院
冶金工业过程系统科学湖北省重点实验室

2024 年 1 月 27 日

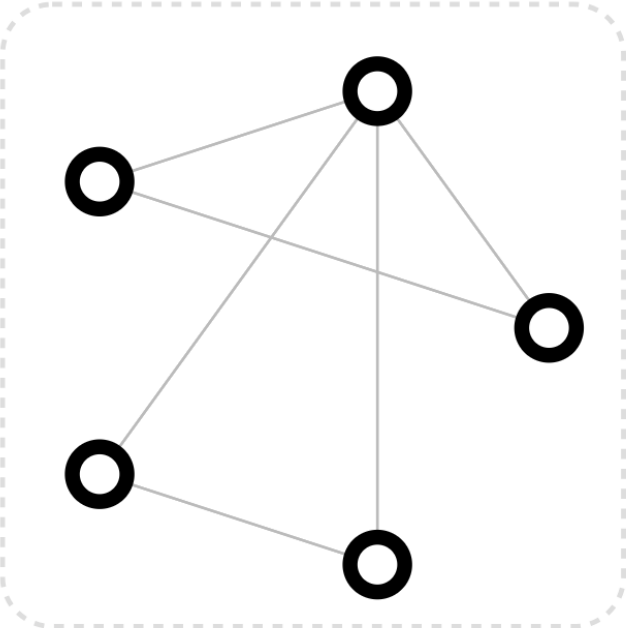
前言

尽管图网络具体实现可通过简单调包实现, 但在此之前仍然需要了解图的基础知识, 特别是 “图嵌入” 和 “信息传递” 的概念.

本幻灯片材料主要参考 Sanchez-Lengeling[?] 等人的论文。

目录

Vertex, Edge and Global



V Vertex (or node) attributes

e.g., node identity, number of neighbors

E Edge (or link) attributes and directions

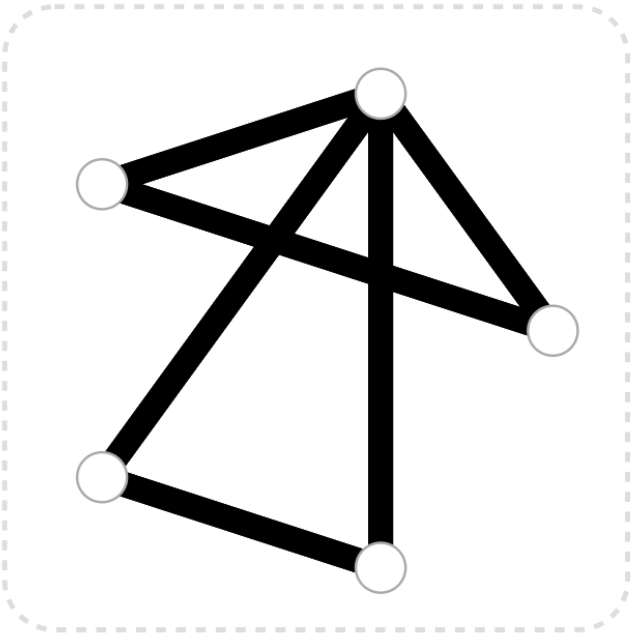
e.g., edge identity, edge weight

U Global (or master node) attributes

e.g., number of nodes, longest path

图 1: 节点，节点一般有自己的特征。

Vertex, Edge and Global



V Vertex (or node) attributes

e.g., node identity, number of neighbors

E Edge (or link) attributes and directions

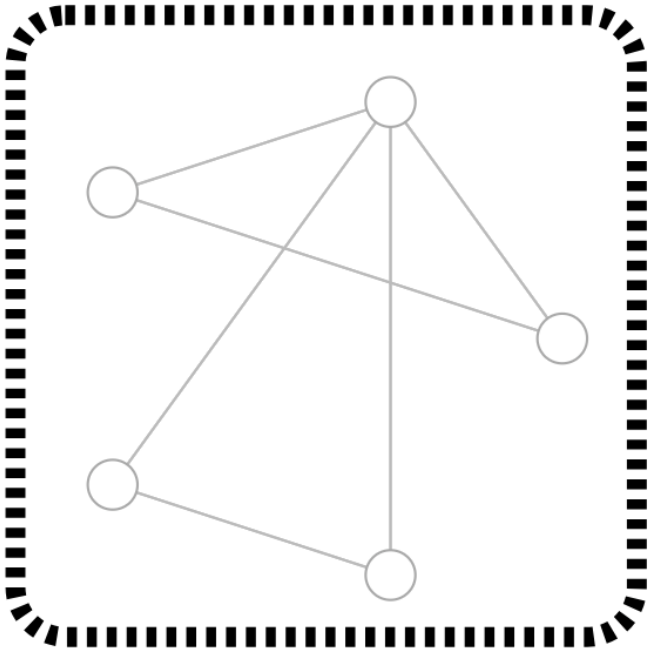
e.g., edge identity, edge weight

U Global (or master node) attributes

e.g., number of nodes, longest path

图 2: 边，节点一般也有自己的特征。

Vertex, Edge and Global



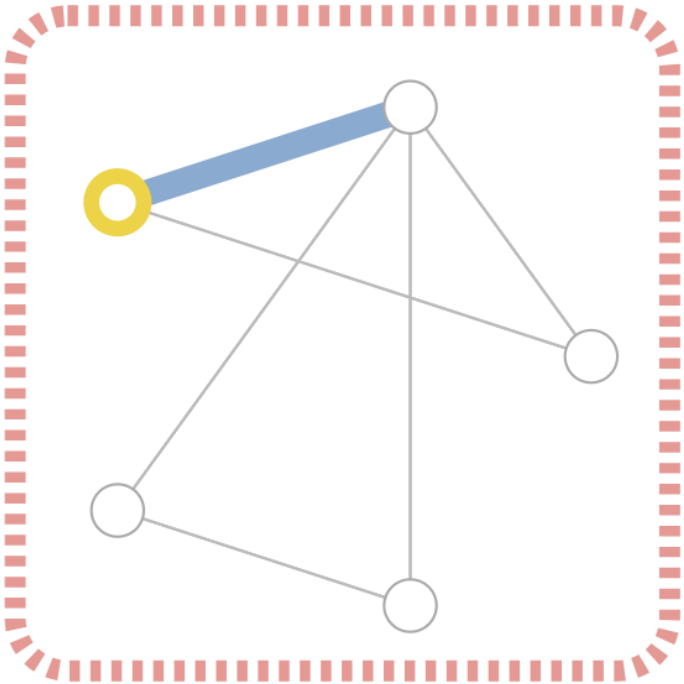
V Vertex (or node) attributes
e.g., node identity, number of neighbors

E Edge (or link) attributes and directions
e.g., edge identity, edge weight

U Global (or master node) attributes
e.g., number of nodes, longest path

图 3: 全局特征，做图分类任务的时候存在。

Vertex, Edge and Global



Information in the form of scalars or embeddings can be stored at each graph node (left) or edge (right).

图的三种元素都可以包含嵌入向量, 这里嵌入是一维的, 可视化呈现.

图像

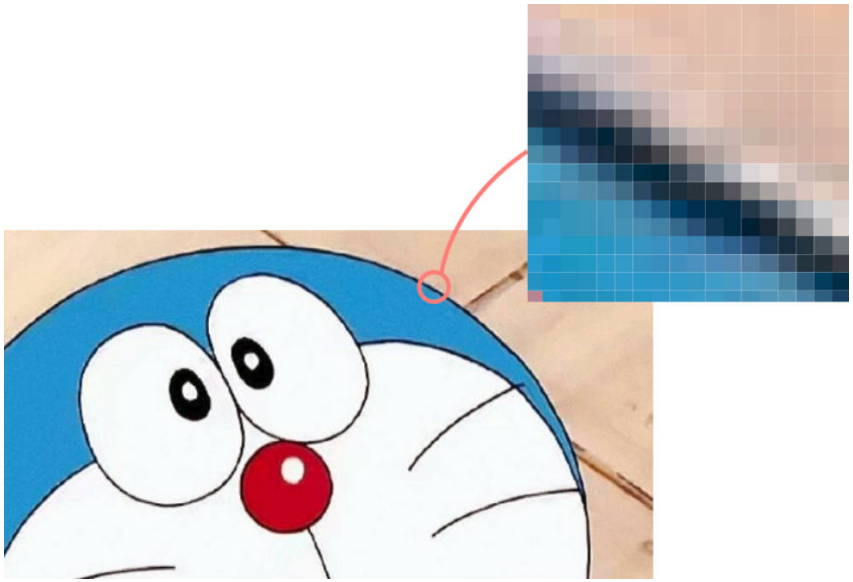


图 4: 图片是由许多像素点构成的, 每个像素点有一个或多个数值, 如灰度值或者 RGB 值.

从图像到图

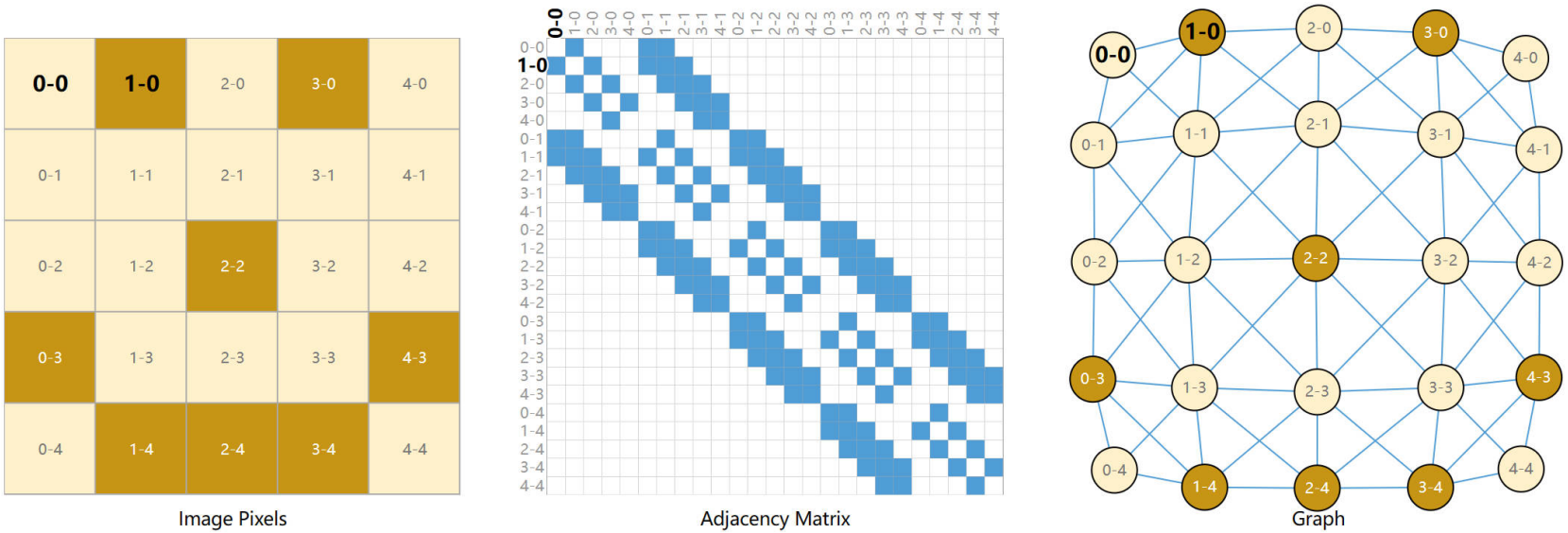


图 5: 左边是图像, 中间是邻接矩阵, 右边是图.

图像也可以表示为图, 一个 5*5 的图像可以表示为邻接矩阵和图的形式. 这里三种不同的表现方法是同一个信息的不同表示方法.

句子表示为图

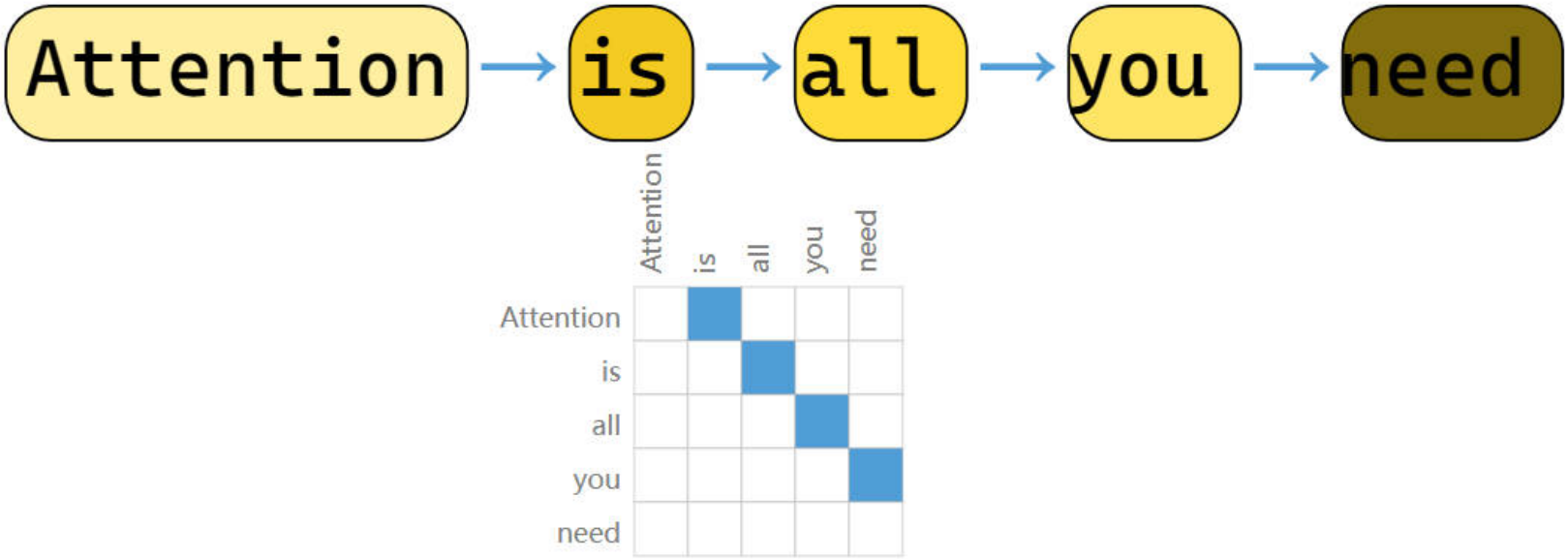


图 6: 语句表现为图的形式.

这里邻接矩阵去掉了对角连接和下三角表示. 另外, 语句在机器学习中通常不会用图表示, 而是通过编码并映射到嵌入向量.

分子结构图

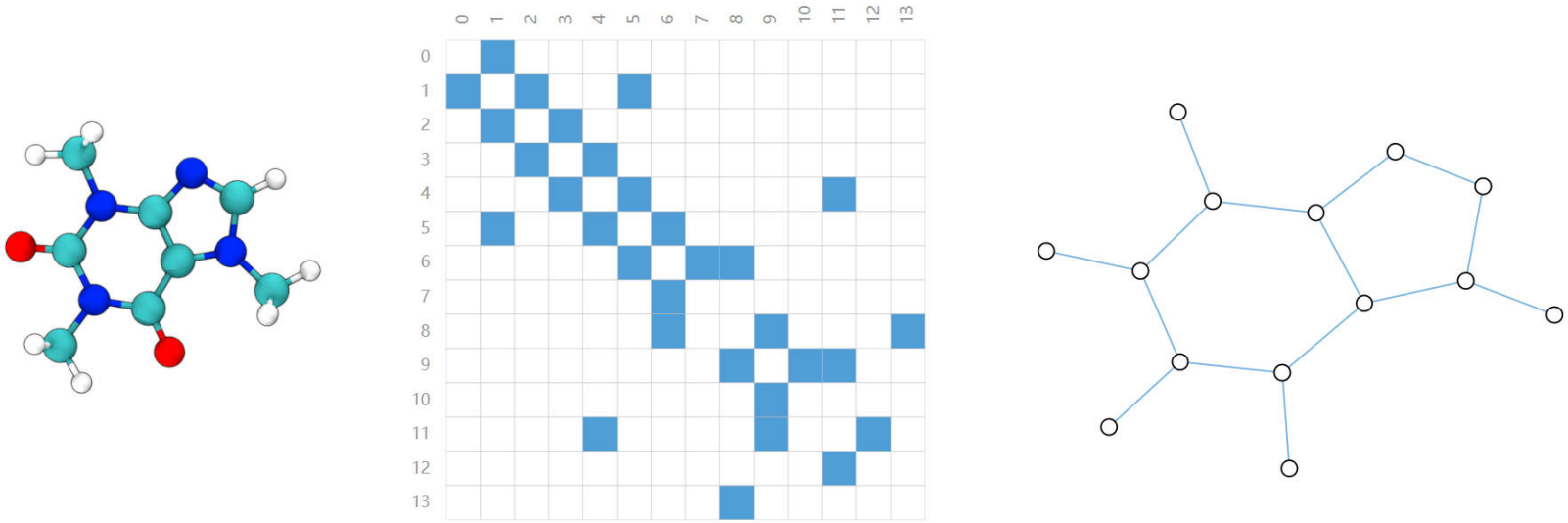
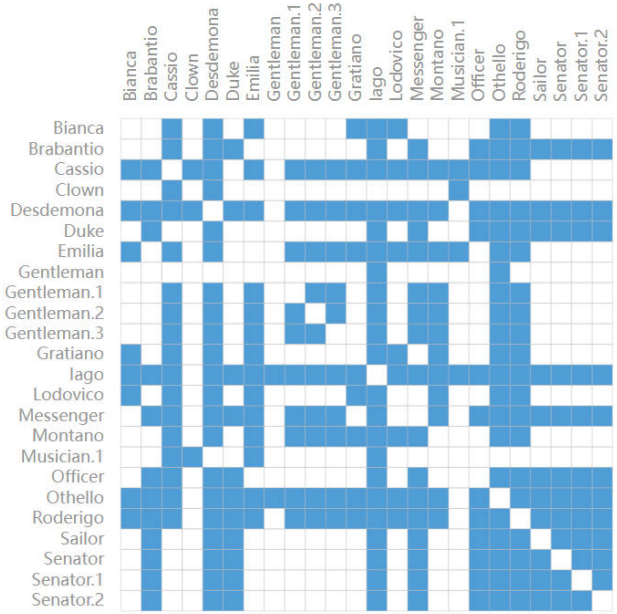


图 7: 分子结构表现为图.

相比于前面的图, 分子图更具有异质性.

人物关系图



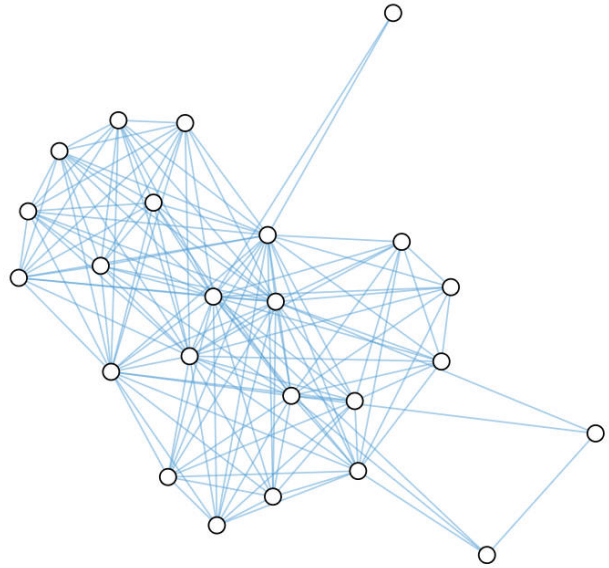


图 8: 人物关系建模为图.

话剧奥赛罗中的人物关系可以建模为图, 节点表现为角色, 边建模为人物间联系.

论文引用关系图

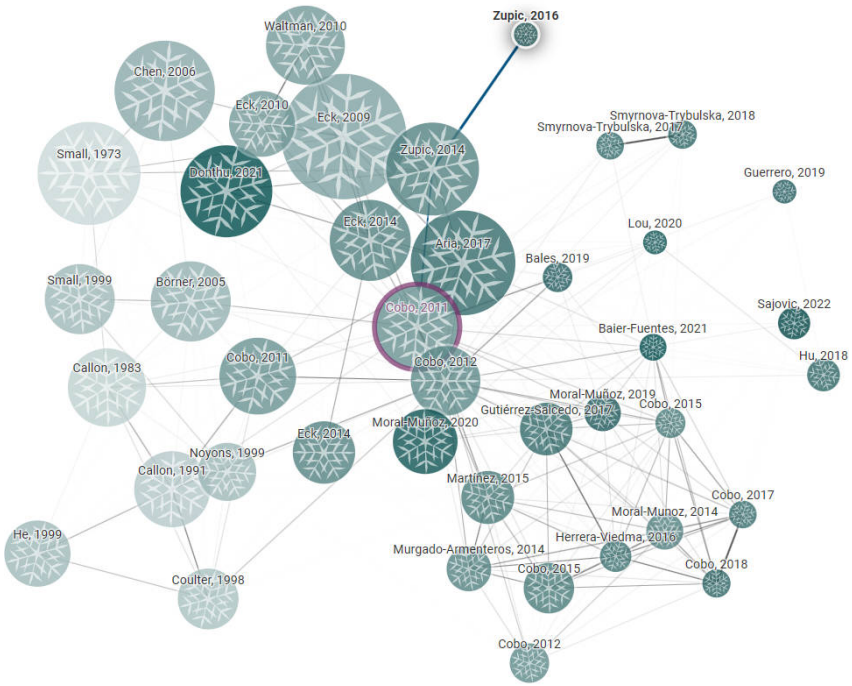


图 9: 文章引用表示为图

图分类任务

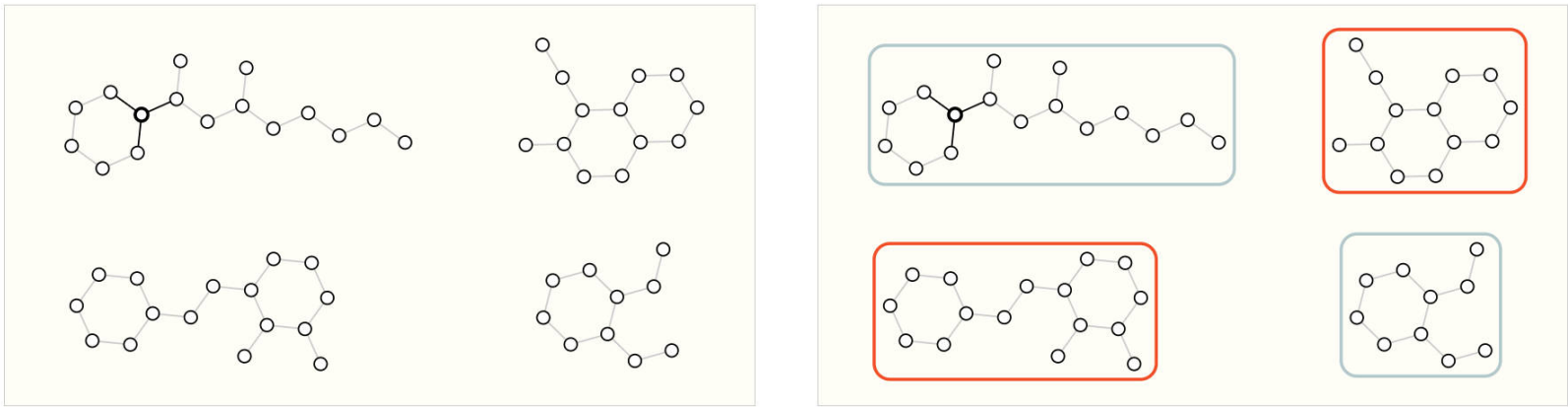


图 10: 识别哪些分子有两个苯环

输入多个分子的图信息, 输出各个图的类别.

点分类任务

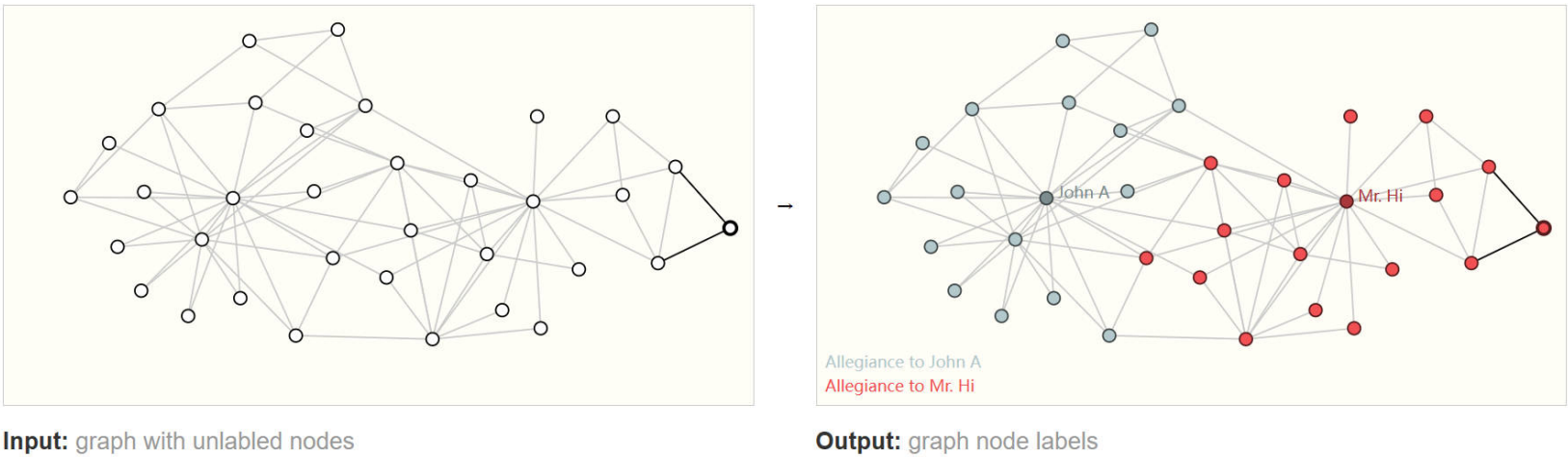


图 11: 识别那些人支持 John H, 哪些人支持 Hi.

输入一个图, 输出各个点的类别, 这里是二分类.

边分类任务

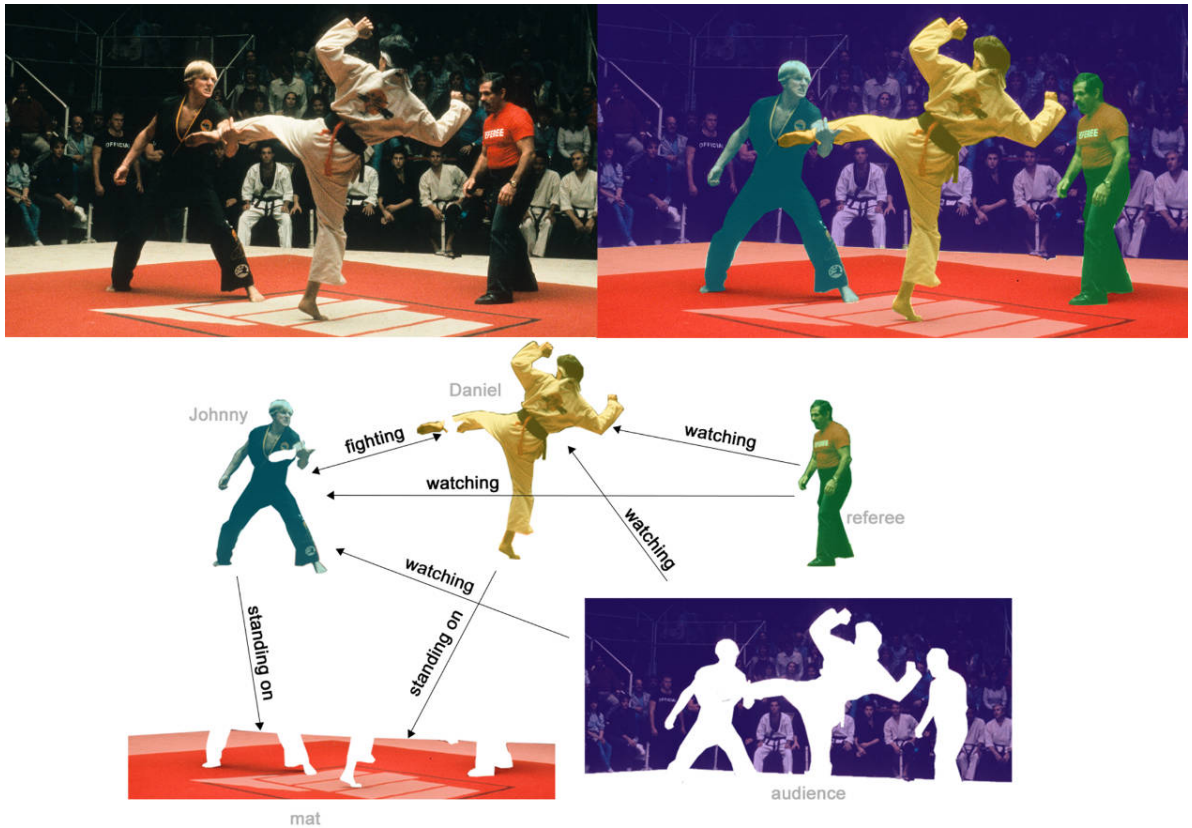


图 12: 先对图片做分割, 然后把被分割的部分作为节点, 识别各个节点之间的关系. 对局者站在垫子上, 对局者相互对抗, 裁判和观众观看对抗.

边分类任务

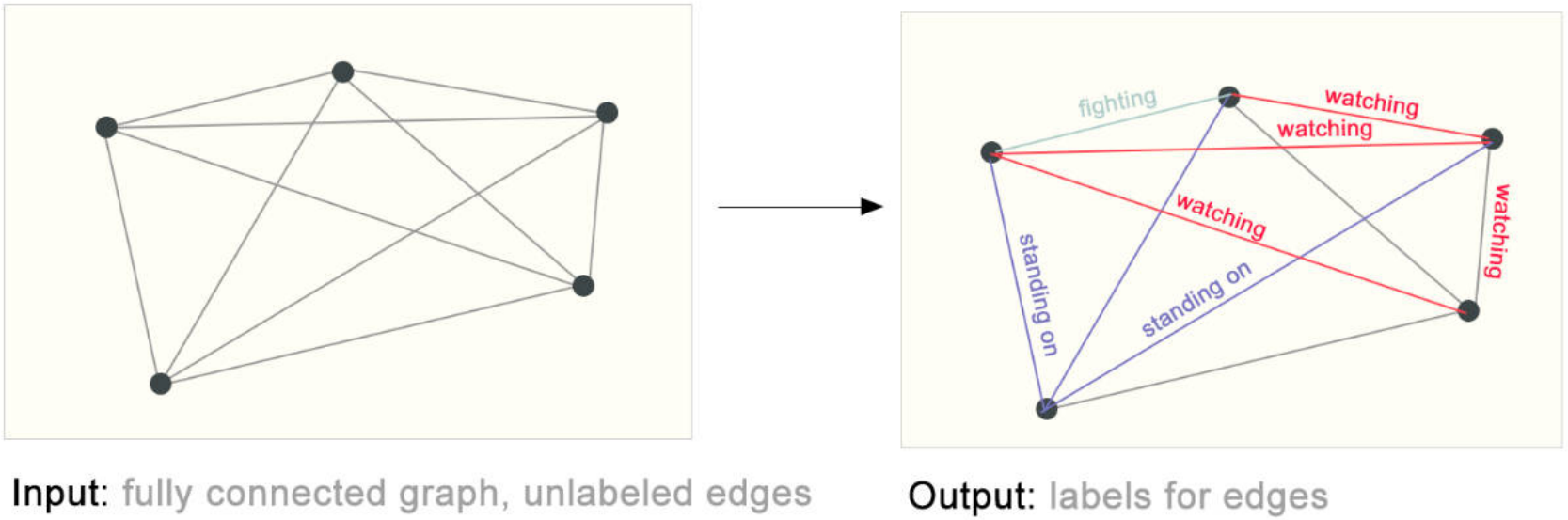


图 13: 图 9 简化图.

邻接矩阵不足以表示图信息

Two adjacency matrices representing the same graph.

图 14: 邻接矩阵经过初等变换, 即行变换或者列变换, 其含义并没有改变, 但是矩阵特征发生很大改变.

邻接矩阵不足以表示图信息

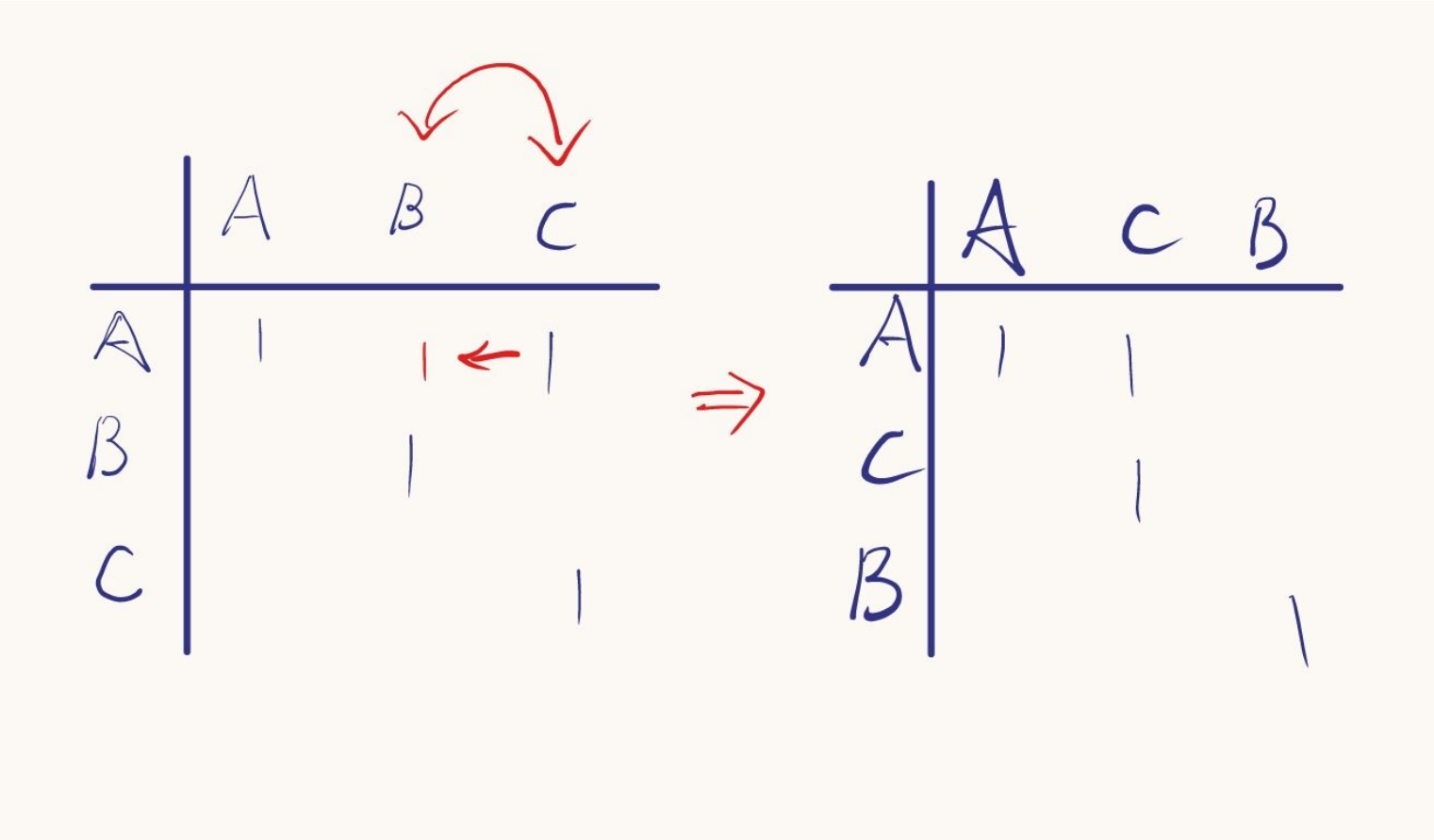


图 15: 简单例子

图嵌入

下面的图片展示了图如何构建为嵌入向量, 需要注意的是, 在邻接表（edge_index）中, 哪两个节点有连接是按列表示的, 也同时按照顺序定义了方向.

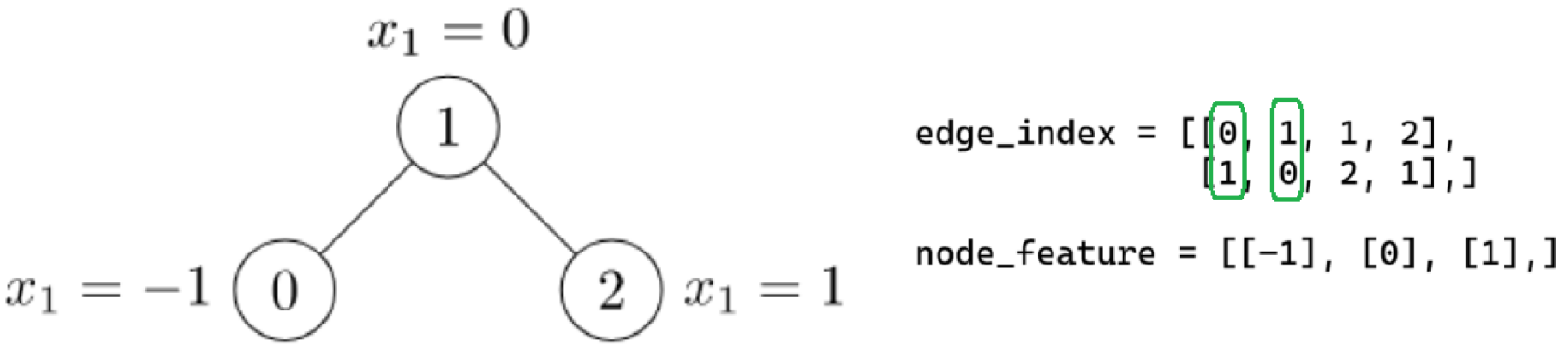


图 16: 邻接矩阵无法用于学习, 但是图可以转化为嵌入向量形式以学习. 通过 edge_index、node_feature 等若干嵌入向量可以准确表示图信息.

图嵌入

- 以节点举例, 图 ?? 中一个节点的属性可以表示为-1,0,-1 三个数字, 也可以表示为任意维度的张量, 其他元素同理.

图嵌入

- 以节点举例, 图 ?? 中一个节点的属性可以表示为-1,0,-1 三个数字, 也可以表示为任意维度的张量, 其他元素同理.
- 如果一个节点表示一个学生, 它的节点属性可以是包含多个信息的向量, 比如 [性别、班级、学校].

图嵌入

- 以节点举例, 图 ?? 中一个节点的属性可以表示为-1,0,-1 三个数字, 也可以表示为任意维度的张量, 其他元素同理.
- 如果一个节点表示一个学生, 它的节点属性可以是包含多个信息的向量, 比如 [性别、班级、学校].
- 如果要对节点分类, 可以给出节点标签；如果是对图分类, 可以给出图标签, 此时节点标签就不是必要的, 比较重要的是节点特征和邻接表：节点特征是图的基本信息, 邻接表引定义图结构.

图嵌入

- 以节点举例, 图 ?? 中一个节点的属性可以表示为-1,0,-1 三个数字, 也可以表示为任意维度的张量, 其他元素同理.
- 如果一个节点表示一个学生, 它的节点属性可以是包含多个信息的向量, 比如 [性别、班级、学校].
- 如果要对节点分类, 可以给出节点标签；如果是对图分类, 可以给出图标签, 此时节点标签就不是必要的, 比较重要的是节点特征和邻接表：节点特征是图的基本信息, 邻接表引定义图结构.
- 这里的邻接表可以表示为其他不同的形式, 以使用的图网络的包的要求为准.

处理图结构的基础方法

分开处理每个嵌入向量

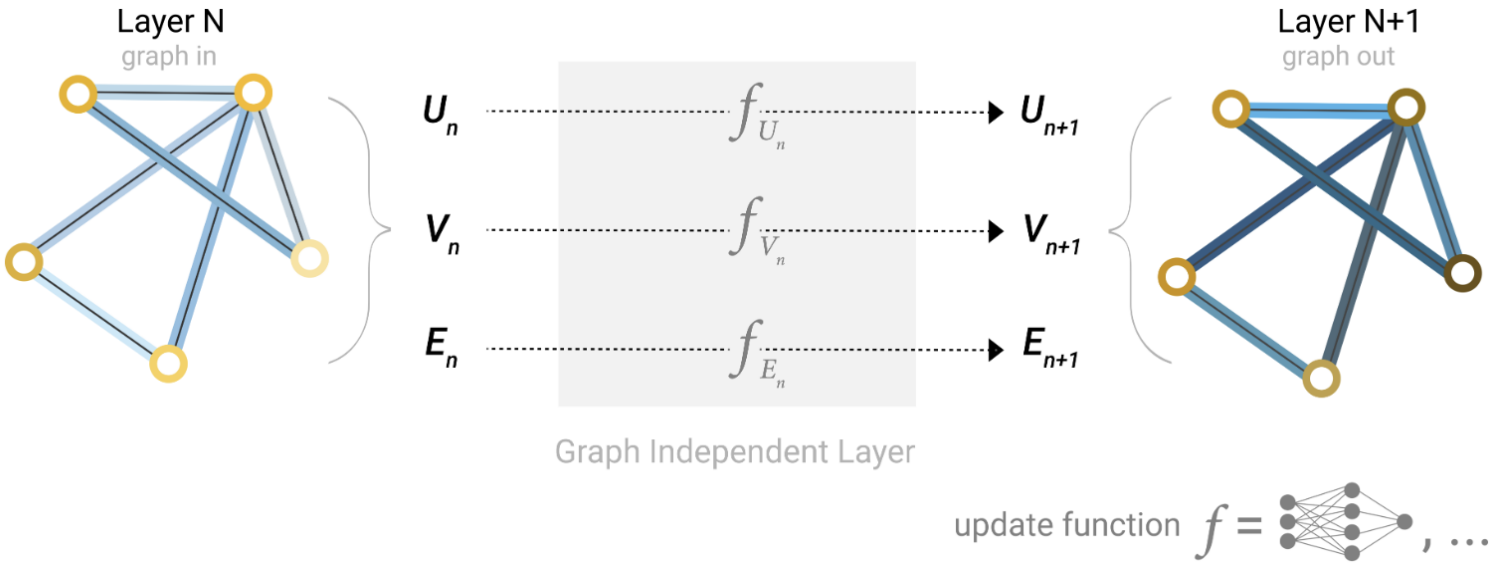


图 17: 简易 GNN. 前面我们已经把图转化成包含点信息、边信息、全局信息的嵌入形式, 因此我们可以分别构建三个神经网络去学习它们各自的特征. 另外, 这里图神经网络没有改变节点直接的连接性, 因此经过处理前后的邻接矩阵也不会变化.

基础点分类

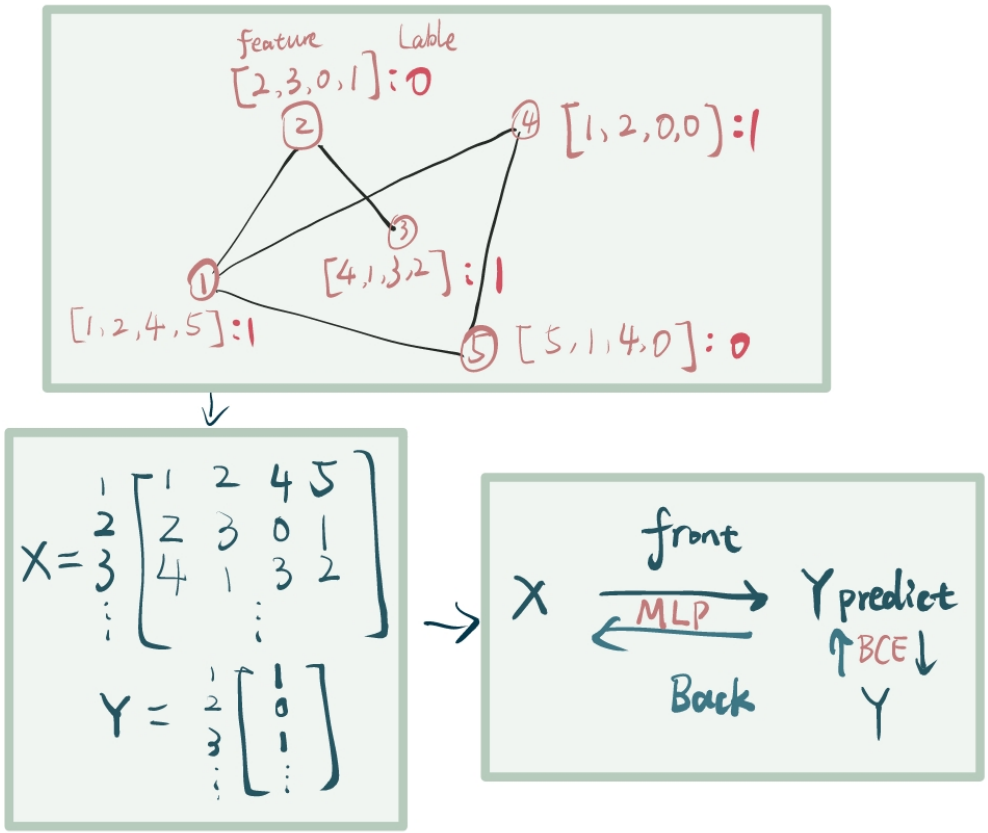


图 18: 基础点分类图解. 把点特征整理为一个矩阵, 点标签整理为列向量, 发现与普通机器学习并无区别. 边分类和图分类同理.

信息传递

信息传递是图神经网络的关键要素.

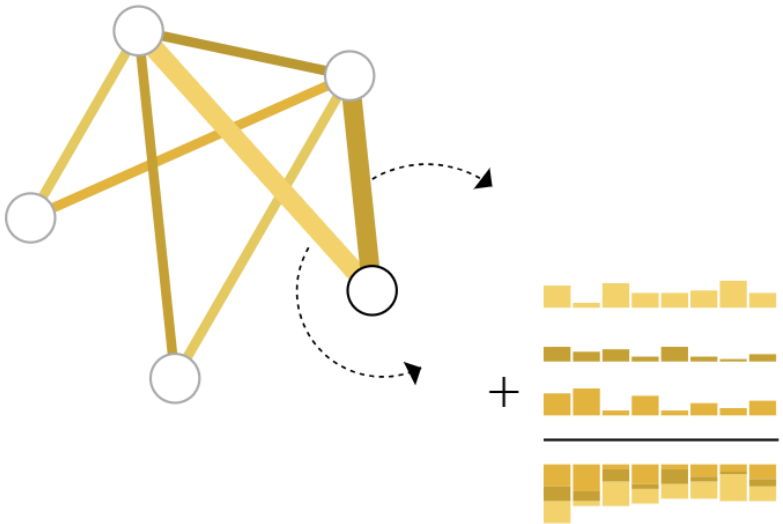


图 19: 信息传递机制. 信息传递也叫聚合. 一个节点可以从与其连接的边或者节点聚合信息. 边也可以聚合节点的信息. 边嵌入和点嵌入至少要有一个. 聚合操作一般是指池化. 聚合之后同样可以通过 MLP 学习.

问题:1. 池化会不会损失太多信息?

节点信息聚合到全局

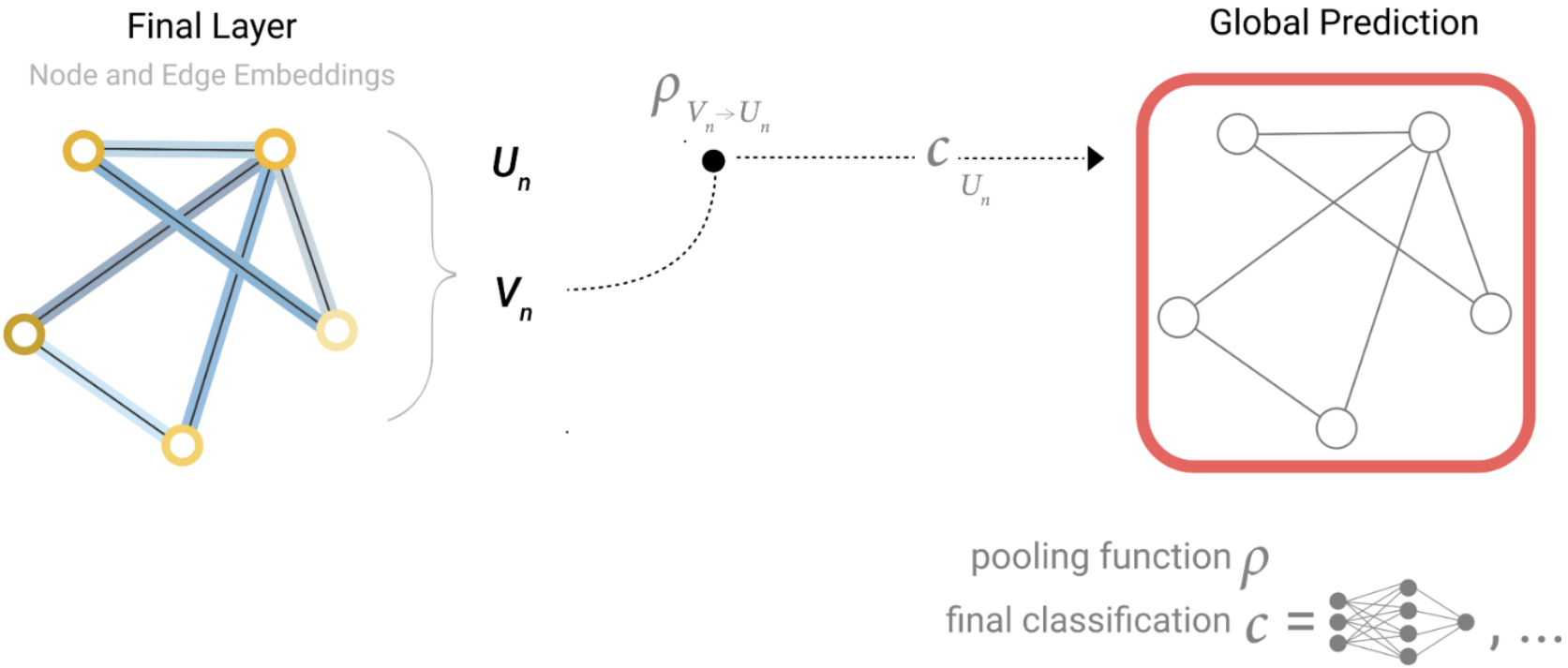


图 20: 节点预测全局. 对全局节点做池化聚合到全局嵌入, 可以实现全局信息聚合. 同理还可以聚合边的信息到全局.

比较复杂的聚合

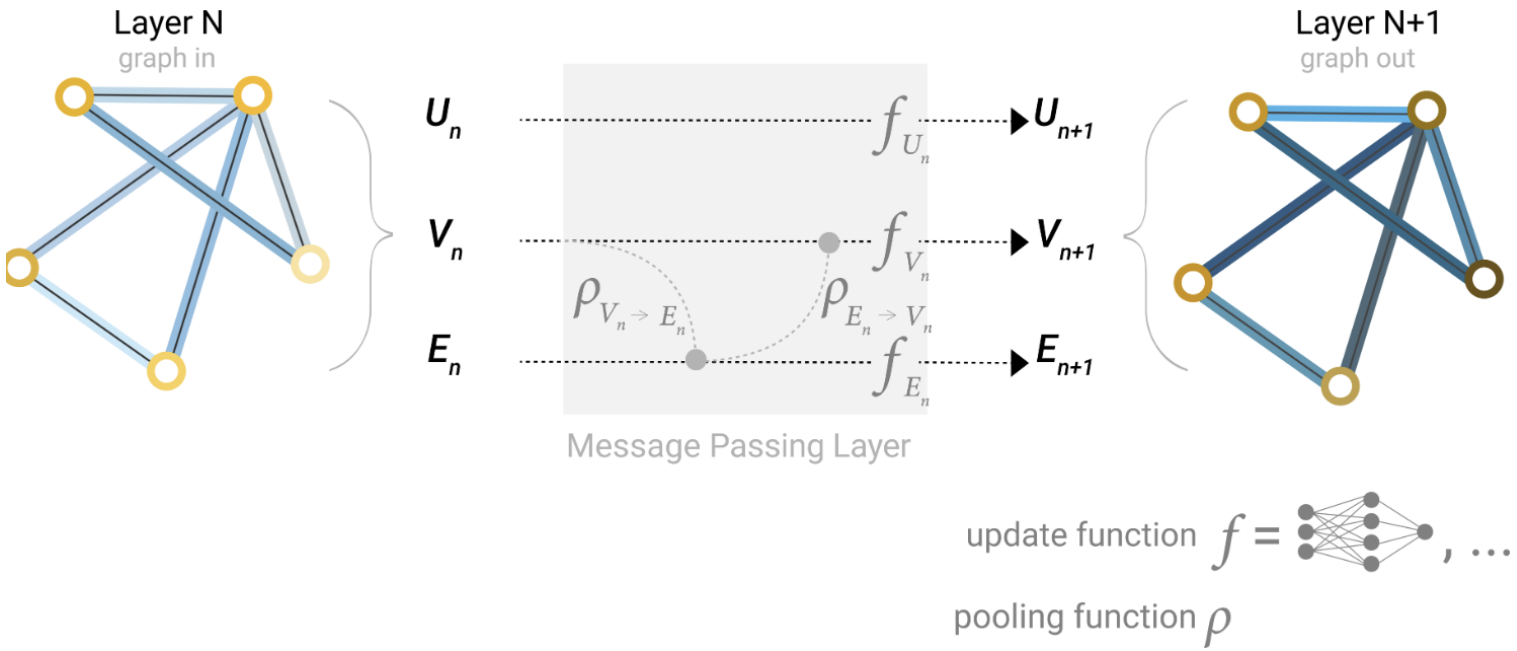


图 21: 节点-边-节点聚合.

比较复杂的聚合

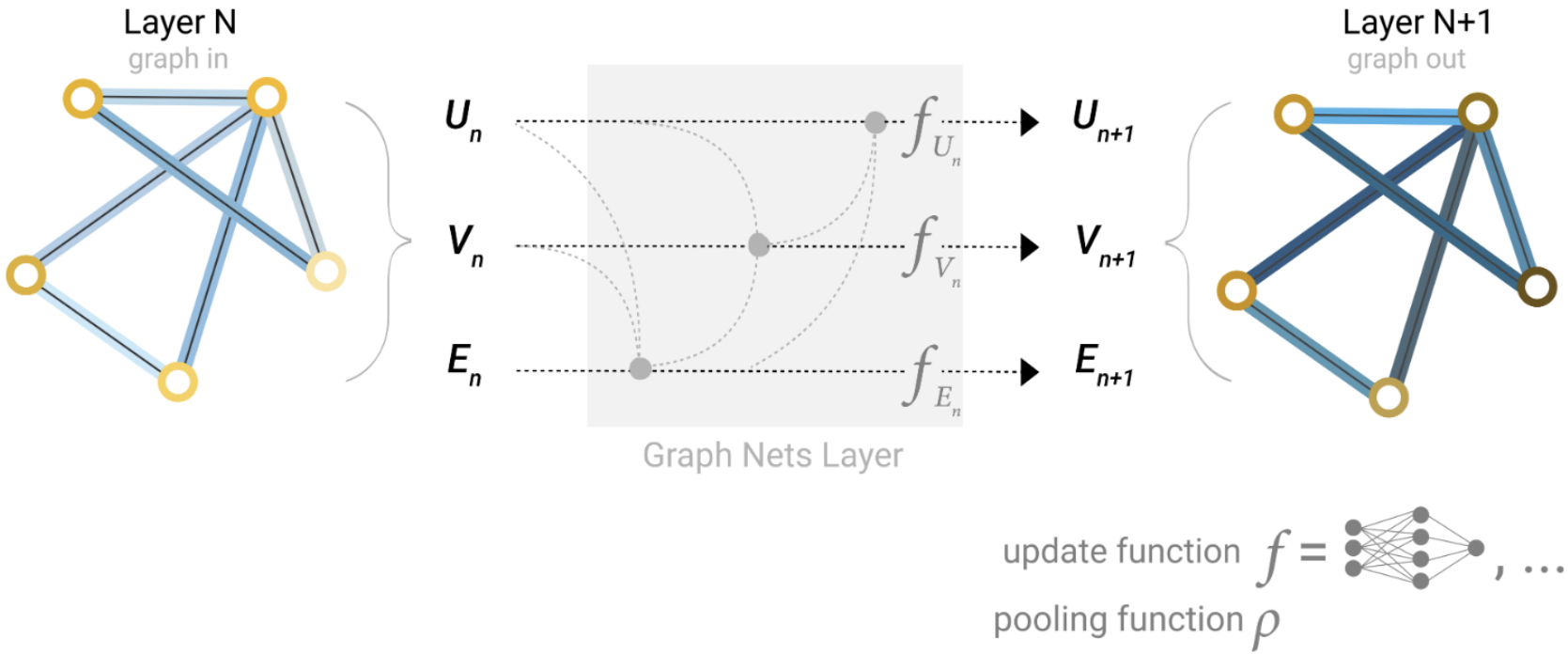


图 22: 节点-边-全局混合聚合.

- 池化聚合是最简单的方法之一, 图卷积方法是现在的主流聚合方案, 在此基础上很容易加入注意力机制.
- 值得注意的是图卷积和图像卷积在操作上几乎没有相同点.
- 创新聚合方法是一种主要的改进方向.

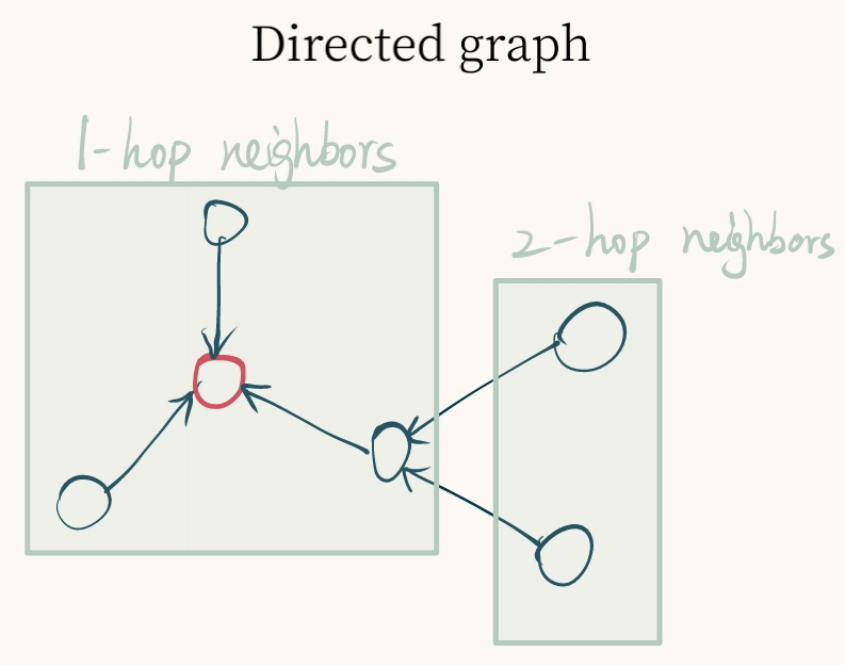


图 23: 一种有向图. 这表示边是单向的, 消息传递只能按照边方向进行. 其中红色节点包括 1-hop 邻居和 2-hop 邻居.

Note

无向图是指边双向连接而不是无方向连接.

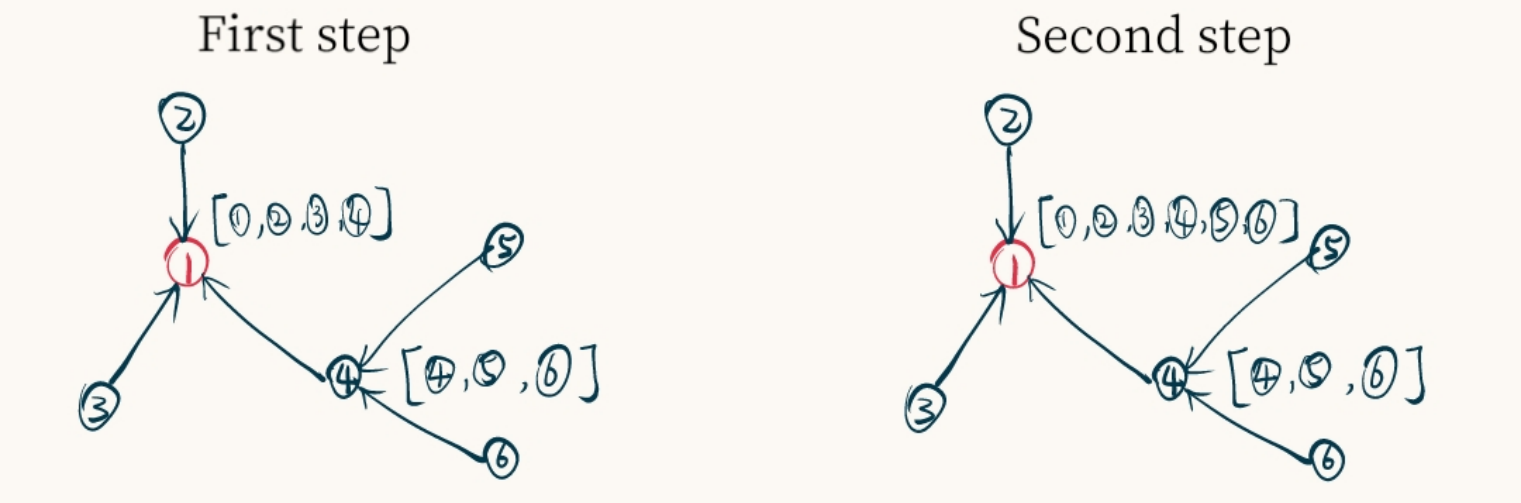


图 24: 对于红色节点, 通过两次信息传递就可以获得全局信息.

每一次信息传递都伴随一次神经网络处理, 因此这里的 step 可以看作神经网络中的 epoch, 两步消息传递也就是两个 epoch. 这里仅构建了有向图, 对于无向图来说, 每个节点每一步都在聚合邻居信息.

方法一

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \qquad H = \begin{bmatrix} h_{11} & \dots & \dots & \dots \\ h_{21} & \dots & \dots & \dots \\ h_{31} & \dots & \dots & \dots \\ h_{41} & \dots & \dots & \dots \end{bmatrix}$$

这里随机给出一个邻接矩阵 A 和其度矩阵 D，最简单的聚合公式是：

$$H^{(t+1)} = \sigma\left(AH^{(t)}W^{(t)}\right)$$

H 是特征矩阵，包含所有节点的特征，所有上标都表示更新步，σ 是非线性激活函数，A 是邻接矩阵，之前说过图神经网络不改变连接性，因此 A 是不变的，W 是可训练的权重矩阵，等于神经网络的线性映射.

方法一

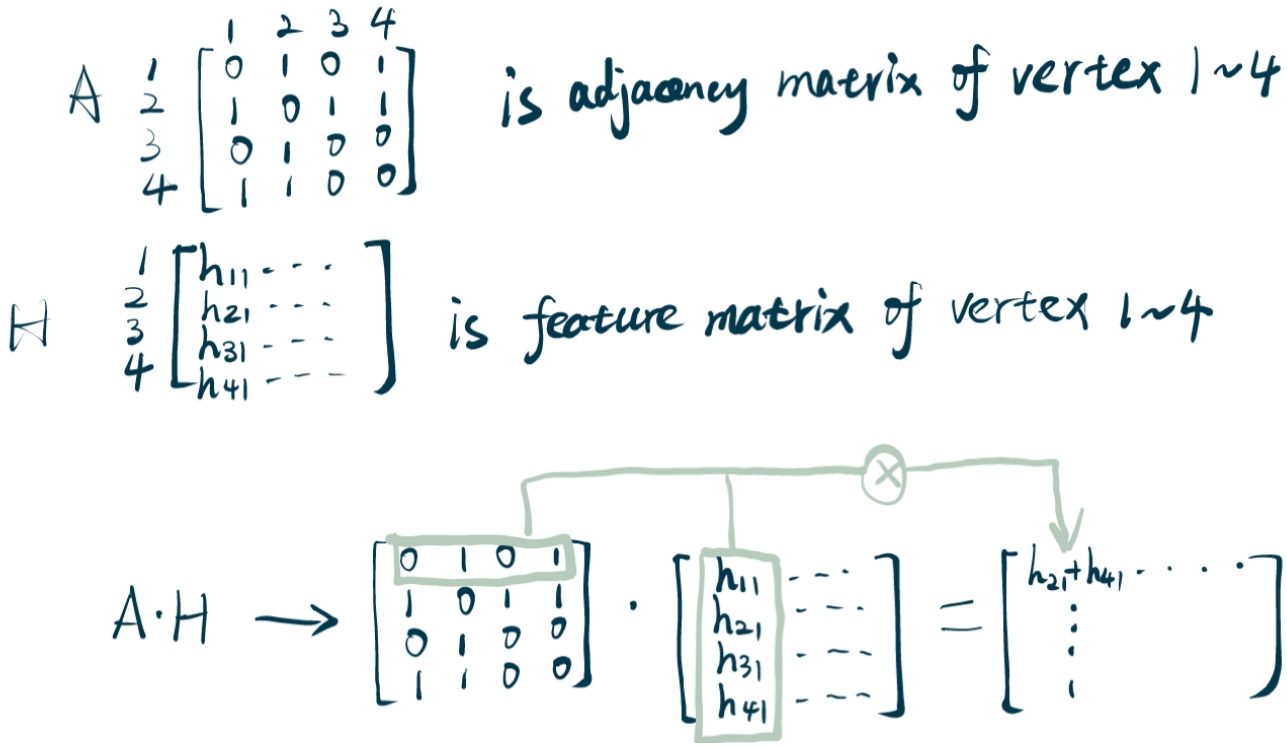


图 25: AH 运算使得 h_{11} 变为 $h_{21}+h_{41}$, 因为 V_{11} 与节点 V_{21}, V_{41} 相邻.

但是这样运算显然丢失了 V_{11} 自己的信息. 其他节点同理.

方法二

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

引入度矩阵 D, 表现为一个节点有多少个入度, 构建拉普拉斯矩阵 (Combinatorial Laplacian) L:

$$L = D - A$$

图卷积更新公式:

$$H^{(t+1)} = \sigma(LH^{(t)}W^{(t)})$$

这样在运算时就考虑了节点自身的信息. 但有些节点的度可能非常大, 从而显著影响分类效果, 所以方法三对 L 矩阵做了标准化.

方法三: 最常用的图卷积方法

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$D^{-\frac{1}{2}} = \begin{bmatrix} \sqrt{2}^{-1} & 0 & 0 & 0 \\ 0 & \sqrt{3}^{-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{2}^{-1} \end{bmatrix}$$

$$\tilde{L} = \begin{bmatrix} 1 & -\frac{1}{\sqrt{6}} & 0 & -\frac{1}{2} \\ -\frac{1}{\sqrt{6}} & 1 & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ 0 & -\frac{1}{\sqrt{3}} & 1 & 0 \\ -\frac{1}{2} & -\frac{1}{\sqrt{6}} & 0 & 1 \end{bmatrix}$$

前面已经介绍了 L 的公式: $L = D - A$, 可以通过度矩阵标准化 A, 然后给出 $\tilde{L}$, 这里给出标准化公式:

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

通过标准化, 将度和邻接表示进行放缩, 其中度被放缩到 1, 从而避免了了数值太大影响分类效果.

方法三: 最常用的图卷积方法

最后, 给出最常见的图卷积表达式:

$$H^{(t+1)} = \sigma(\tilde{L}H^{(t)}W^{(t)})$$

在图卷积中，由于每次卷积都是在聚合邻居的信息，因此每一对邻居节点都可以依据自身的特征给对方分配一个注意力分数，这就是图卷积注意力方法。

现在你可以调包了:)

PyTorch Geometric 是图网络常用的包之一 `https://pytorch-geometric.readthedocs.io/en/latest/index.html`

- 图需要转化为嵌入向量才能被处理;
- 图非常擅长描述具有拓扑结构的数据;
- 图神经网络的特点在于信息传递。

[1] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B. Wiltschko.
A gentle introduction to graph neural networks.
Distill, 2021.
https://distill.pub/2021/gnn-intro.

Thanks!