图卷积神经网络（GCN）

作者：Calvin
QQ: 179209347
Mail: 179209347@qq.com

介绍

笔记简介：

- 面向对象：深度学习初学者
- 依赖课程：**线性代数，统计概率**，优化理论，图论，离散数学，微积分，信息论

知乎专栏：

https://zhuanlan.zhihu.com/p/693738275

Github & Gitee 地址：

https://github.com/mymagicpower/AIAS/tree/main/deep_learning

https://gitee.com/mymagicpower/AIAS/tree/main/deep_learning

*** 版权声明：**

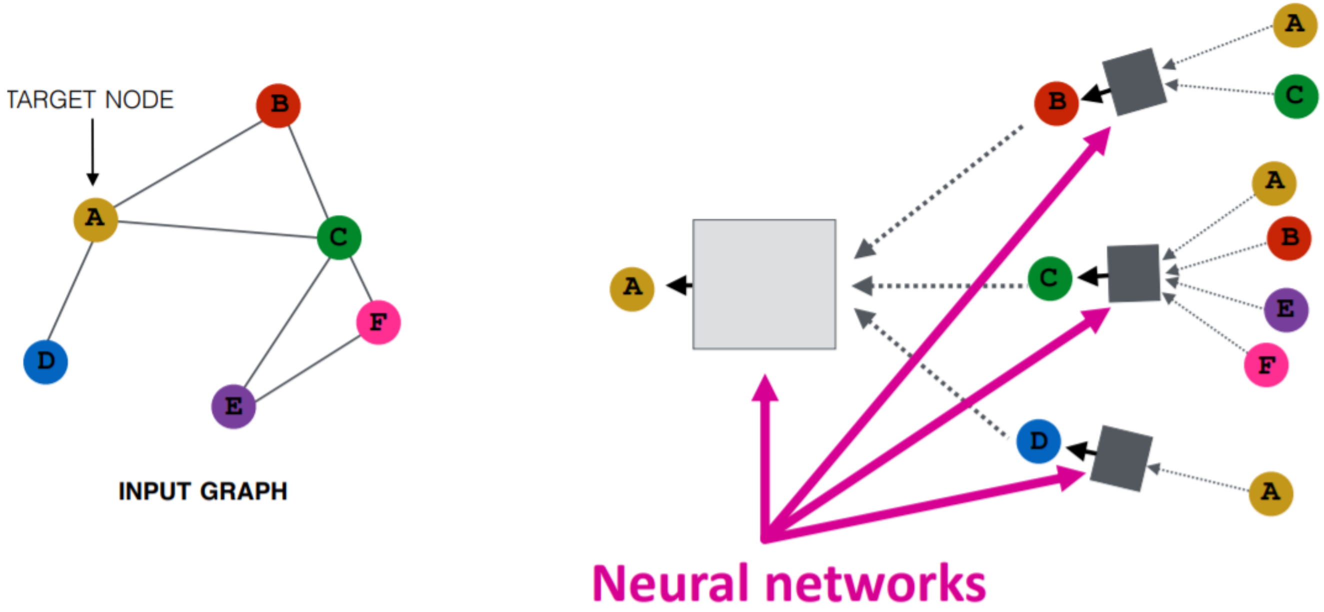
- 仅限用于个人学习
- 禁止用于任何商业用途

图卷积神经网络（GCN）

图卷积神经网络（Graph Convolutional Networks, GCN）是一类专门用于处理图结构数据的神经网络。
GCN的核心思想是将卷积操作扩展到图结构上，通过对每个节点及其邻居节点的信息进行聚合和转换，从而学习节点的表示。这类似于在图像处理中，卷积神经网络（CNN）对每个像素及其邻域进行操作。

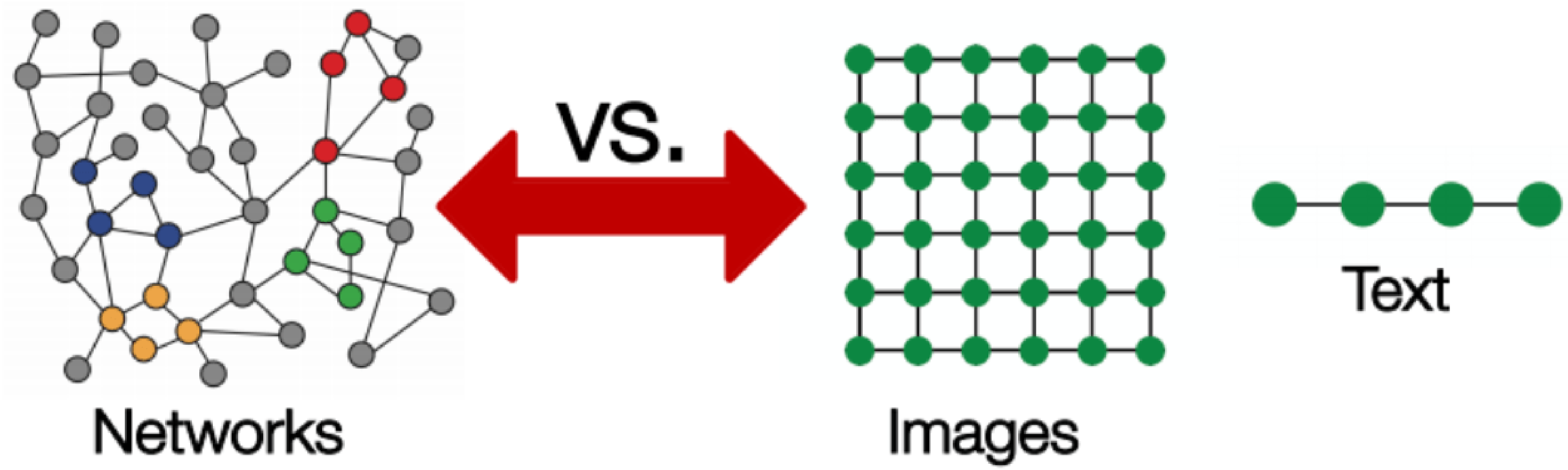
GCN 有两种类型：

- **空间图卷积网络（Spatial Graph Convolutional Networks）**使用空间特征从位于空间空间中的图进行学习。
- **谱图卷积网络（Spectral Graph Convolutional Networks）**使用图拉普拉斯矩阵的特征分解来沿节点传播信息。这些网络的灵感来自信号和系统中的波传播。



图卷积神经网络（GCN）与卷积神经网络（CNN）

传统的机器学习和深度学习工具专门针对简单的数据类型。就像具有相同结构和大小的图像，我们可以将其视为固定大小的网格图。文本和语音都是序列，因此我们可以将它们视为折线图。
但还有更复杂的图，没有固定的形式，具有可变大小的无序节点，其中节点可以具有不同数量的邻居。
现有的机器学习算法有一个核心假设，即实例彼此独立。对于图数据来说这是错误的，因为每个节点都通过各种类型的链接与其他节点相关。



图卷积神经网络（GCN） - 常见任务

常见任务：

图分类（Graph Classification）

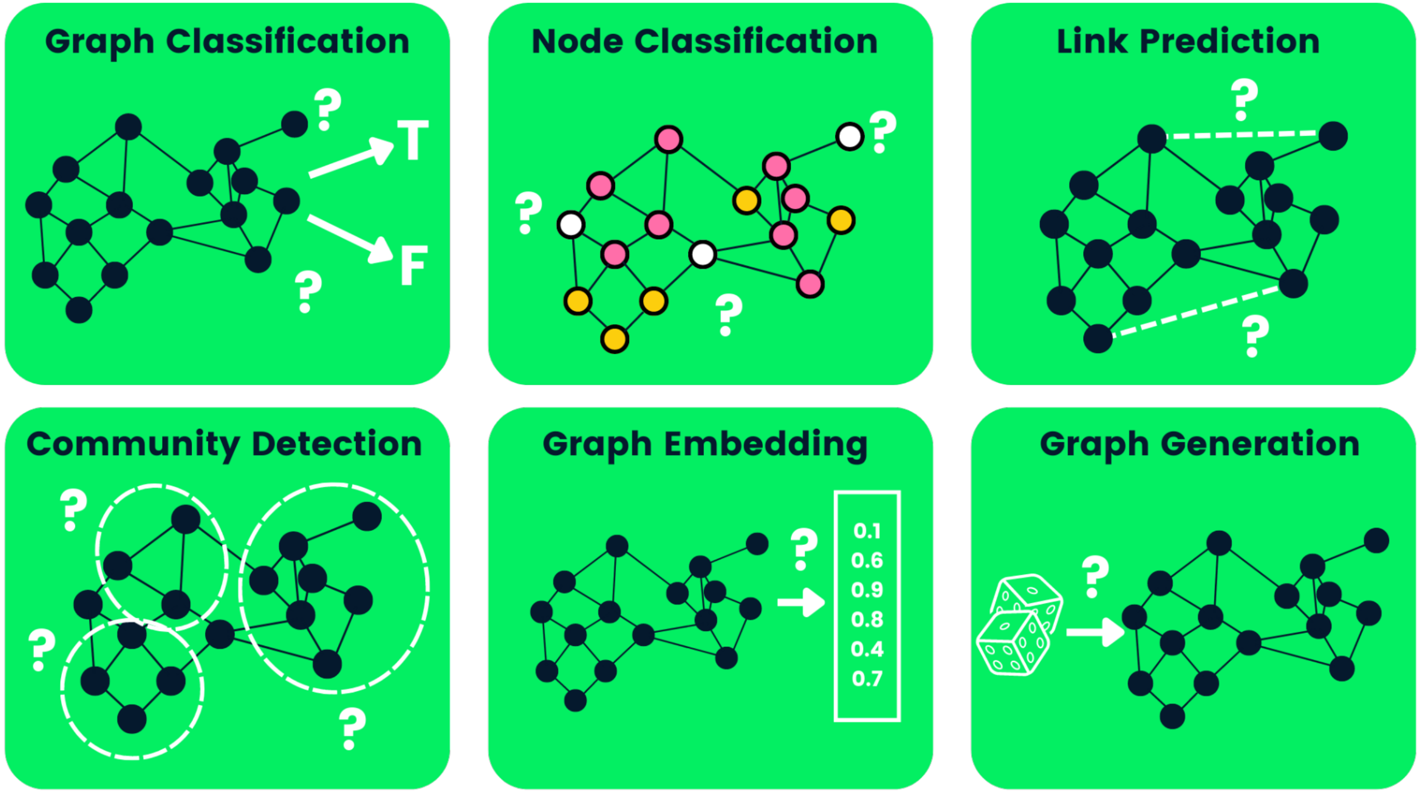
节点分类（Node Classification）

链接预测（Link Prediction）

社区检测（Community Detection）

图嵌入（Graph Embedding）

图生成（ Graph Generation）



图卷积神经网络（GCN）- 工作原理

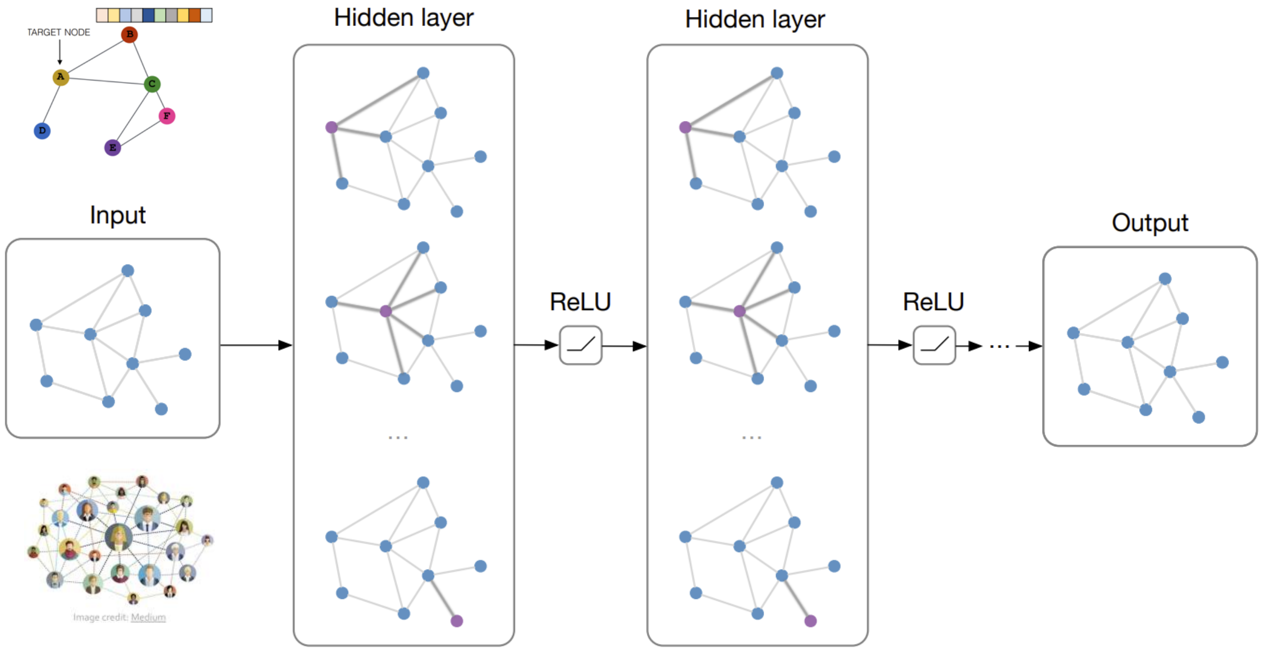
GCN的工作原理

GCN通过层叠多个图卷积层来学习节点的特征表示。每一层的操作可以概括为以下两个步骤：

- 特征聚合（Aggregation）：每个节点收集其邻居节点的特征。
- 特征更新（Update）：每个节点基于收集到的邻居特征和自身特征进行更新。

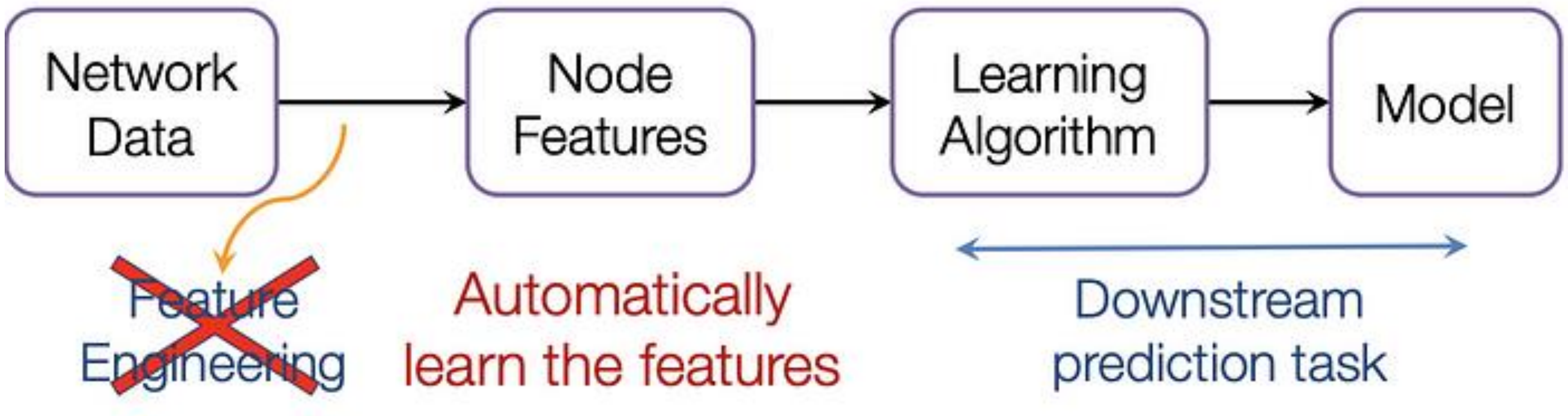
$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

- $H^{(l)}$ 是第 l 层的节点特征矩阵。
- $\tilde{A}$ =$A+I$ 是加上自连接后的邻接矩阵。
- $\tilde{D}$ 是 $\tilde{A}$ 的度矩阵。
- $W^{(l)}$ 是第 l 层的权重矩阵。
- σ 是激活函数（如ReLU）。



GCN - 工作原理 - 获取特征

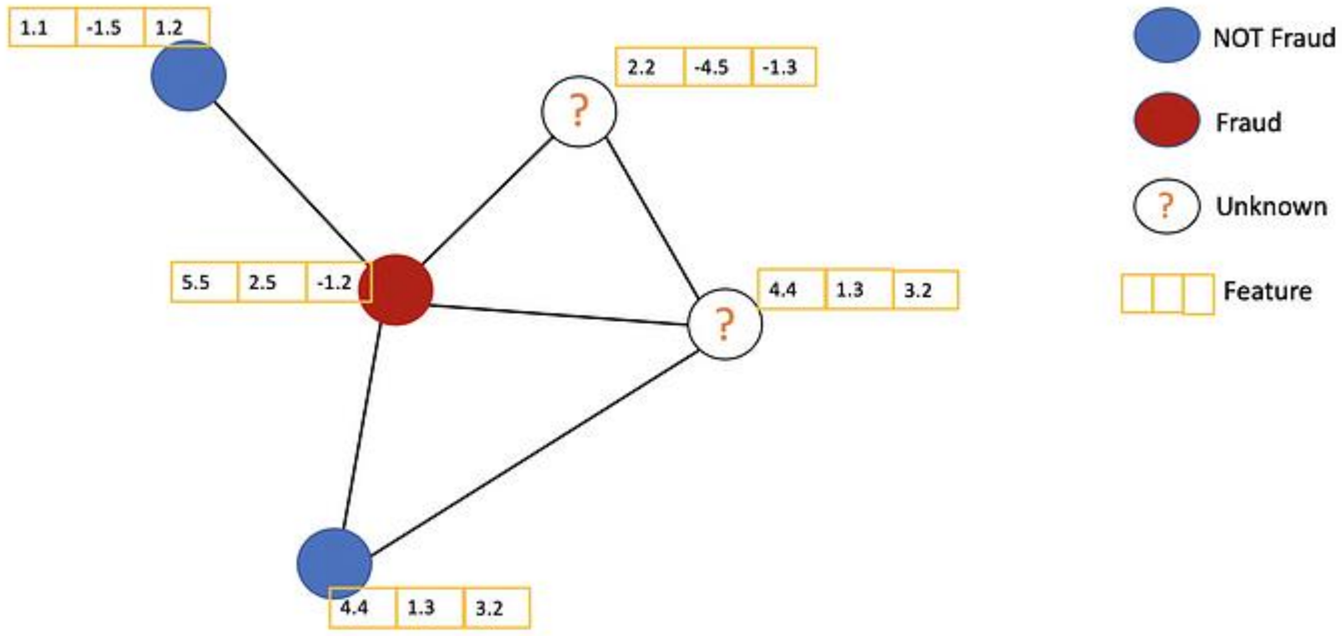
图表示学习：图能够自己学习“特征工程”。



GCN - 工作原理 – 半监督学习

GCN 解决了对图中（例如引文网络）中的节点（例如文档）进行分类的问题，其中标签仅适用于一小部分节点（半监督学习）

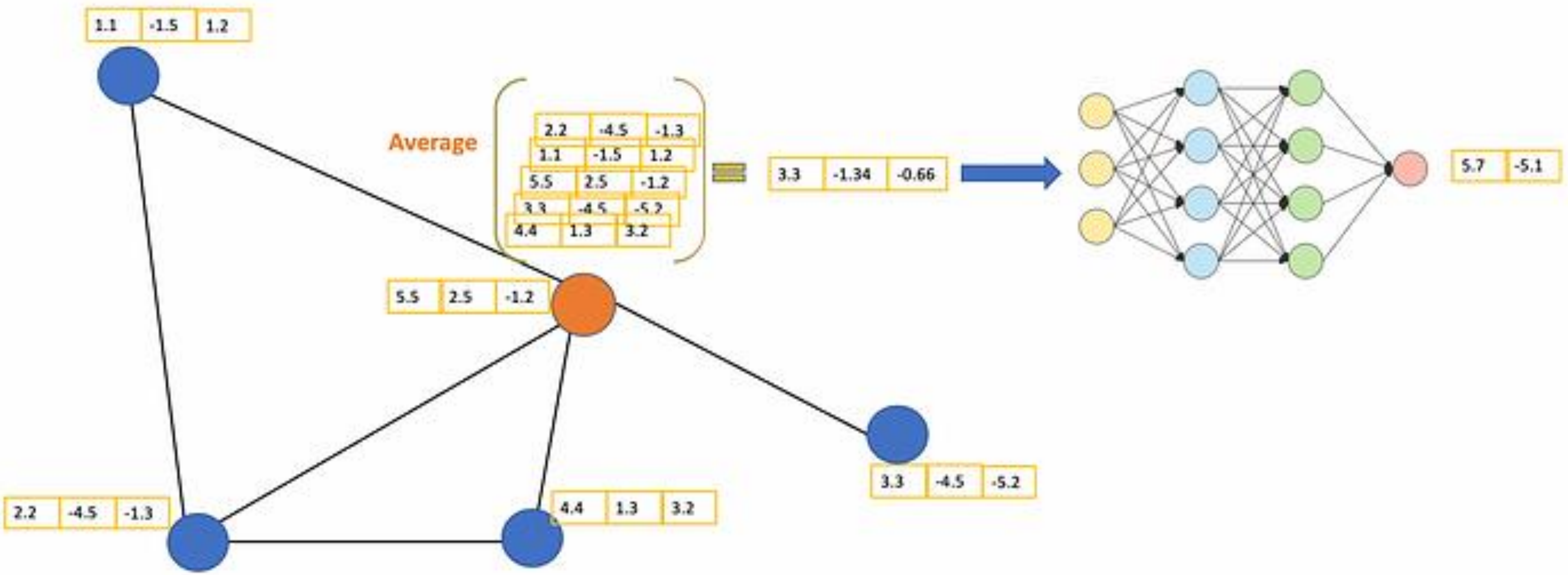
- 不需要全部标签
- 用少量标签也能训练
- 计算损失时只用有标签的



GCN 的总体思想

对于每个节点，我们从其所有邻居获取特征信息，当然还有其自身的特征。假设我们使用average()函数。我们将对所有节点执行相同的操作。最后，我们将这些平均值输入神经网络。

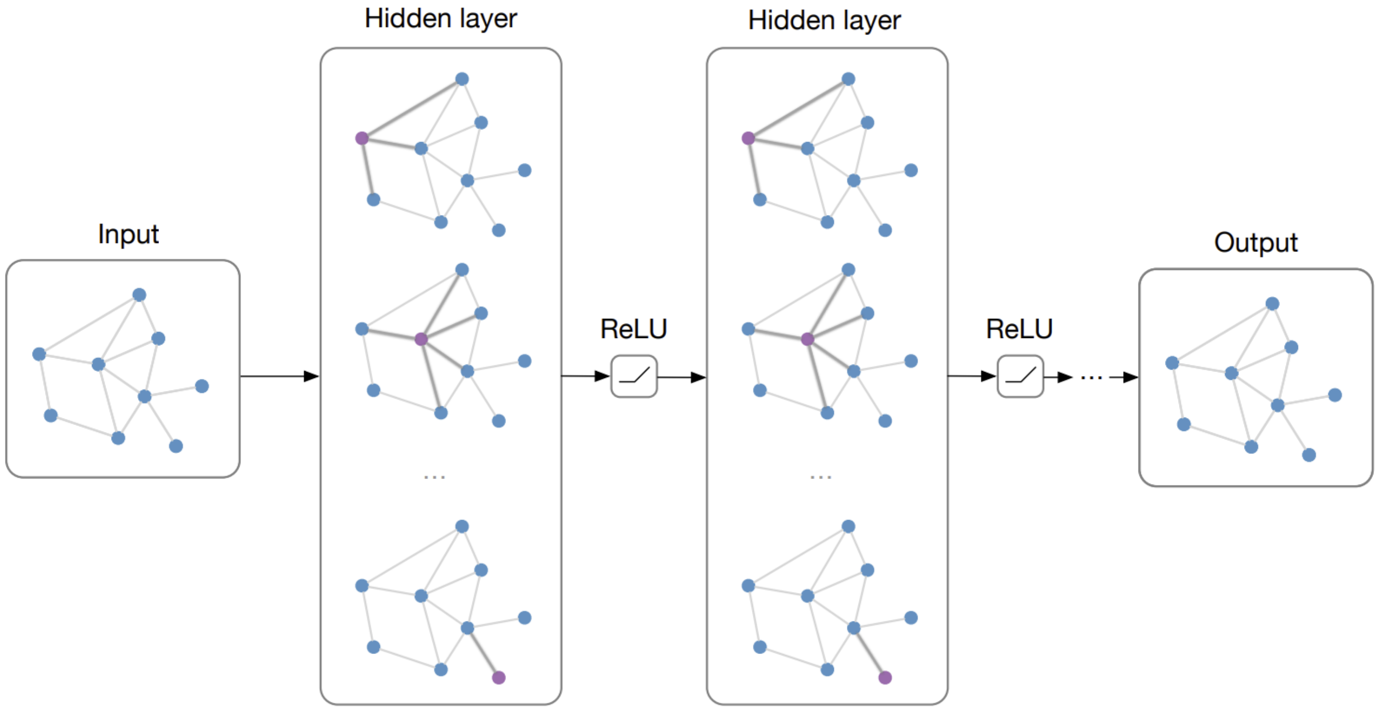
在下图中，我们有一个带有引文网络的简单示例。每个节点代表一篇研究论文，而边缘则代表引文。我们这里有一个预处理步骤。我们不使用原始论文作为特征，而是**将论文转换为向量**（通过使用 NLP 嵌入，例如*tf–idf*、*Doc2Vec*）。



考虑中间的橙色节点：首先，我们取其所有邻居（包括其自身）的平均值。之后，平均值通过神经网络。请注意，在 GCN 中，我们仅使用全连接层。在此示例中，我们得到二维向量作为输出（全连接层的 2 个节点）。

GCN - 两层简单模型

在实践中，我们可以使用更复杂的聚合函数而不是平均函数。我们还可以堆叠更多层以获得更深的 GCN。一层的输出将被视为下一层的输入。



2层GCN示例：第一层的输出是第二层的输入。

GCN - 两层简单模型

通过叠加多层GCN得到最终的模型(2层):

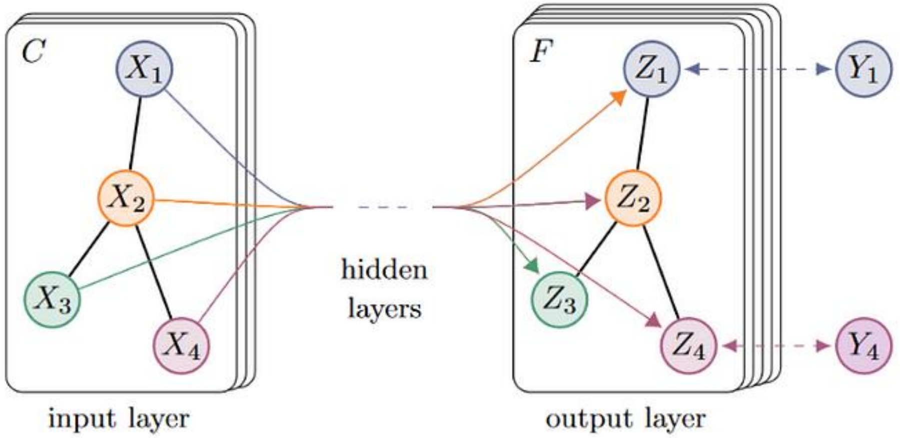
$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-1/2}\tilde{A}\widetilde{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

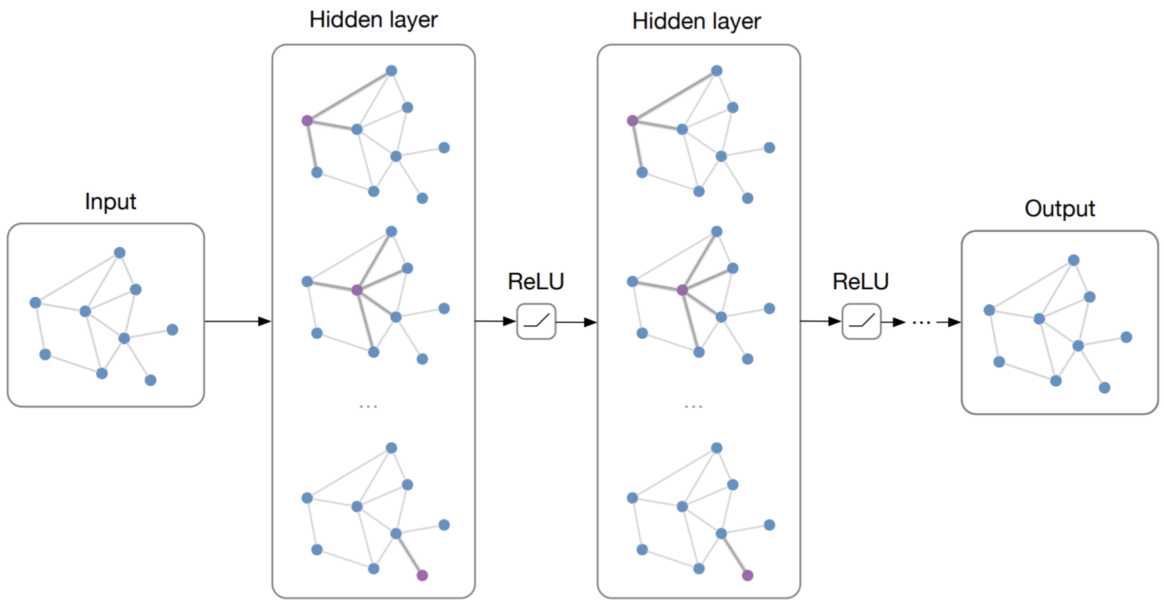
$$\widehat{\boldsymbol{Y}} = f(\boldsymbol{X}.\boldsymbol{A}) = \mathrm{Softmax}\left(\widehat{\boldsymbol{A}}\mathrm{ReLU}\left(\widehat{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}^{0}\right)\boldsymbol{W}^{1}\right)$$

$$\hat{A} = \widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}$$

$$\tilde{A} = A + I_n$$

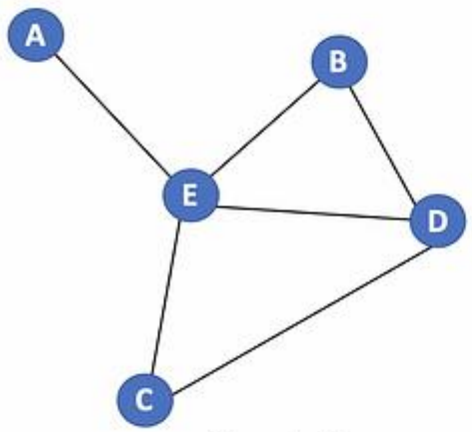
$$\widetilde{D}_{ii} = \sum \tilde{A}_{ii}$$





GCN - 两层简单模型 – 例子分析

从图 G 中，我们有一个邻接矩阵 A 和一个度矩阵 D。我们还有特征矩阵 X。



Graph G

	A	B	C	D	E
A	0	0	0	0	1
B	0	0	0	1	1
C	0	0	0	1	1
D	0	1	1	0	1
E	1	1	1	1	0

Adjacency matrix A

	A	B	C	D	E
A	1	0	0	0	0
B	0	2	0	0	0
C	0	0	2	0	0
D	0	0	0	3	0
E	0	0	0	0	4

Degree matrix D

A	-1.1	3.2	4.2
B	0.4	5.1	-1.2
C	1.2	1.3	2.1
D	1.4	-1.2	2.5
E	1.4	2.5	4.5

Feature vector X

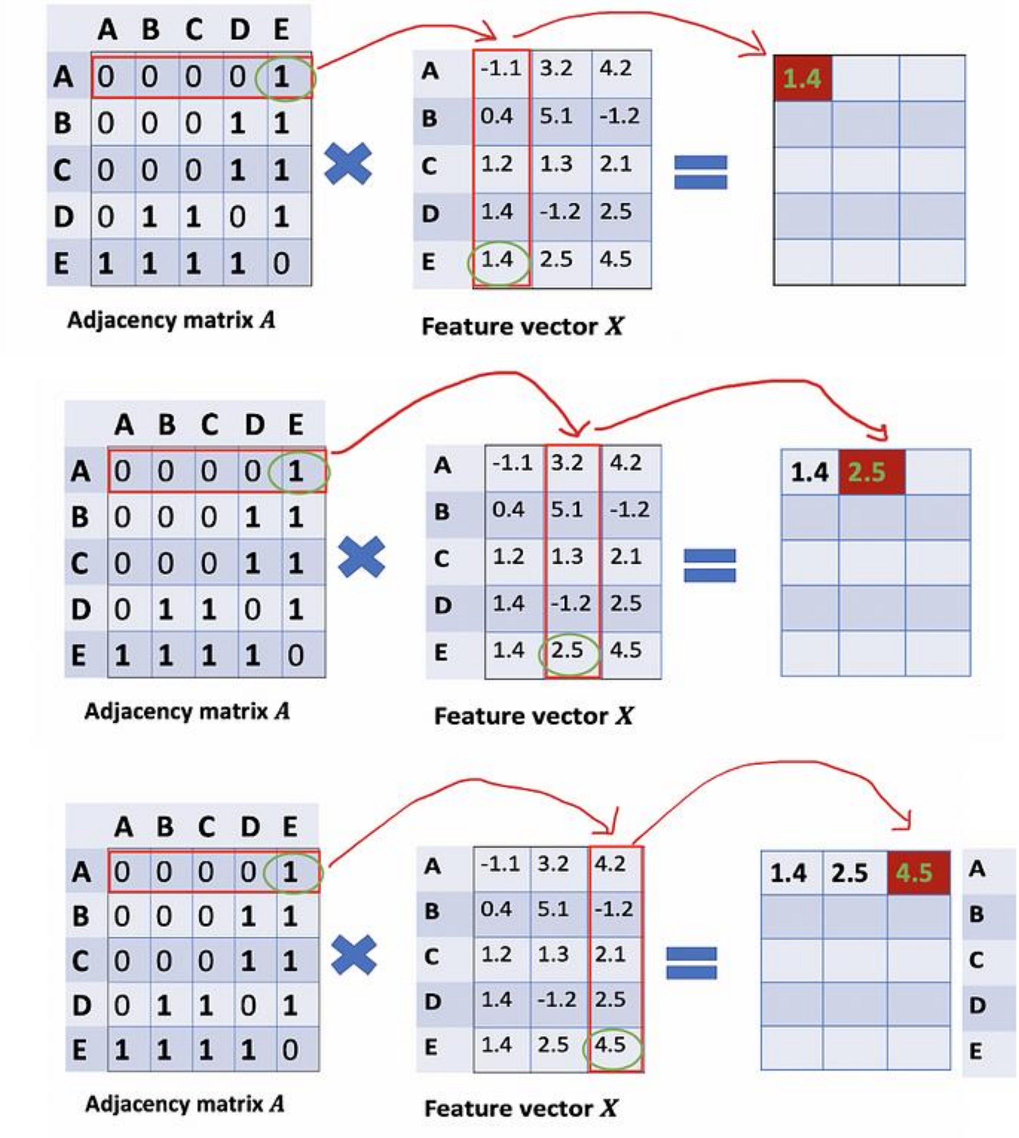
Given an undirected graph $G = (V, E)$ with N nodes $v_i \in V$, edges $(v_i, v_j) \in E$, an adjacency matrix $A \in R^{N\times N}$ (binary or weighted), degree matrix $D_{ii} = \sum_j A_{ij}$ and feature vector matrix $X \in R^{N\times C}$ (N is #nodes, C is the #dimensions of a feature vector).

GCN - 两层简单模型 – 例子分析 - 特征计算方法

我们如何从每个节点的邻居处获取所有特征值？解决方案在于 A 和 X 的乘法。

邻接矩阵与特征矩阵进行乘法操作，表示聚合邻居信息：

- 看一下邻接矩阵的第一行，我们看到节点 A 与 E 有连接。
- 结果矩阵的第一行是 A 连接到的 E 的特征向量。
- 类似地，结果矩阵的第二行是D和E的特征向量之和。
- 通过这样做，我们可以得到所有邻居的向量之和。



GCN - 两层简单模型 – 例子分析 - 特征计算方法

需要改进的地方:

1. 我们错过了节点本身的特性。

在问题 1 中，我们可以通过将单位矩阵 I 添加到 A 来得到新的邻接矩阵来解决这个问题。

$\tilde{A} = A + I$ 是加上自连接后的邻接矩阵。

$$
\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 0 & 0 & 0 & 0 & 1 \\ B & 0 & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 1 & 1 \\ D & 0 & 1 & 1 & 0 & 1 \\ E & 1 & 1 & 1 & 1 & 0 \end{array}
\quad + \quad
\begin{array}{|ccccc|} \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline \end{array}
\quad = \quad
\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & 1 & 0 & 0 & 0 & 1 \\ B & 0 & 1 & 0 & 1 & 1 \\ C & 0 & 0 & 1 & 1 & 1 \\ D & 0 & 1 & 1 & 1 & 1 \\ E & 1 & 1 & 1 & 1 & 1 \end{array}
$$

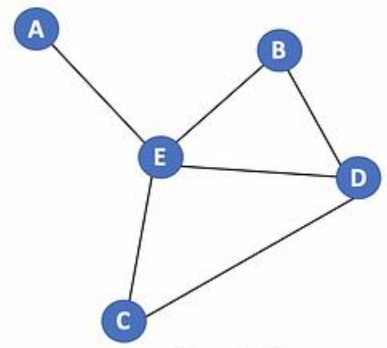
Adjacency matrix A **Identity matrix I** **New Adjacency matrix $\tilde{A}$**

GCN - 两层简单模型 – 例子分析 - 矩阵缩放

需要改进的地方:

2. 我们需要取邻居特征向量的平均值，或者更好的是加权平均值，而不是 sum() 函数。

在问题 2 中，**对于矩阵缩放，我们通常将矩阵乘以对角矩阵。**在这种情况下，我们想要取总和特征的平均值，或者从数学上讲，根据节点度数来缩放和向量矩阵 X 。这里用于缩放的对角矩阵与度矩阵 $\tilde{D}$ 有关（为什么是 $\tilde{D}$ ， 而不是D？因为我们正在考虑新邻接矩阵 $\tilde{A}$ 的度矩阵 $\tilde{D}$ ， 而不是A）。



Graph G

	A	B	C	D	E
A	0	0	0	0	1
B	0	0	0	1	1
C	0	0	0	1	1
D	0	1	1	0	1
E	1	1	1	1	0

Adjacency matrix A

	A	B	C	D	E
A	1	0	0	0	0
B	0	2	0	0	0
C	0	0	2	0	0
D	0	0	0	3	0
E	0	0	0	0	4

Degree matrix D

A	-1.1	3.2	4.2
B	0.4	5.1	-1.2
C	1.2	1.3	2.1
D	1.4	-1.2	2.5
E	1.4	2.5	4.5

Feature vector X

	A	B	C	D	E
A	1	0	0	0	1
B	0	1	0	1	1
C	0	0	1	1	1
D	0	1	1	1	1
E	1	1	1	1	1

New adjacency matrix $\tilde{A}$

2	0	0	0	0
0	3	0	0	0
0	0	3	0	0
0	0	0	4	0
0	0	0	0	5

New degree matrix $\tilde{D}$

1/2	0	0	0	0
0	1/3	0	0	0
0	0	1/3	0	0
0	0	0	1/4	0
0	0	0	0	1/5

$\tilde{D}^{-1}$

$\tilde{D}$ 逆矩阵 $\tilde{D}^{-1}$中的每个元素都是对角矩阵$\tilde{D}$的对应项的倒数。例如，节点 A 的度为 2，因此我们将节点 A 的和向量乘以1/2，而节点 E 的度为 5，我们应该将 E 的和向量乘以1/5，依此类推。

GCN - 两层简单模型 – 例子分析 - 矩阵缩放

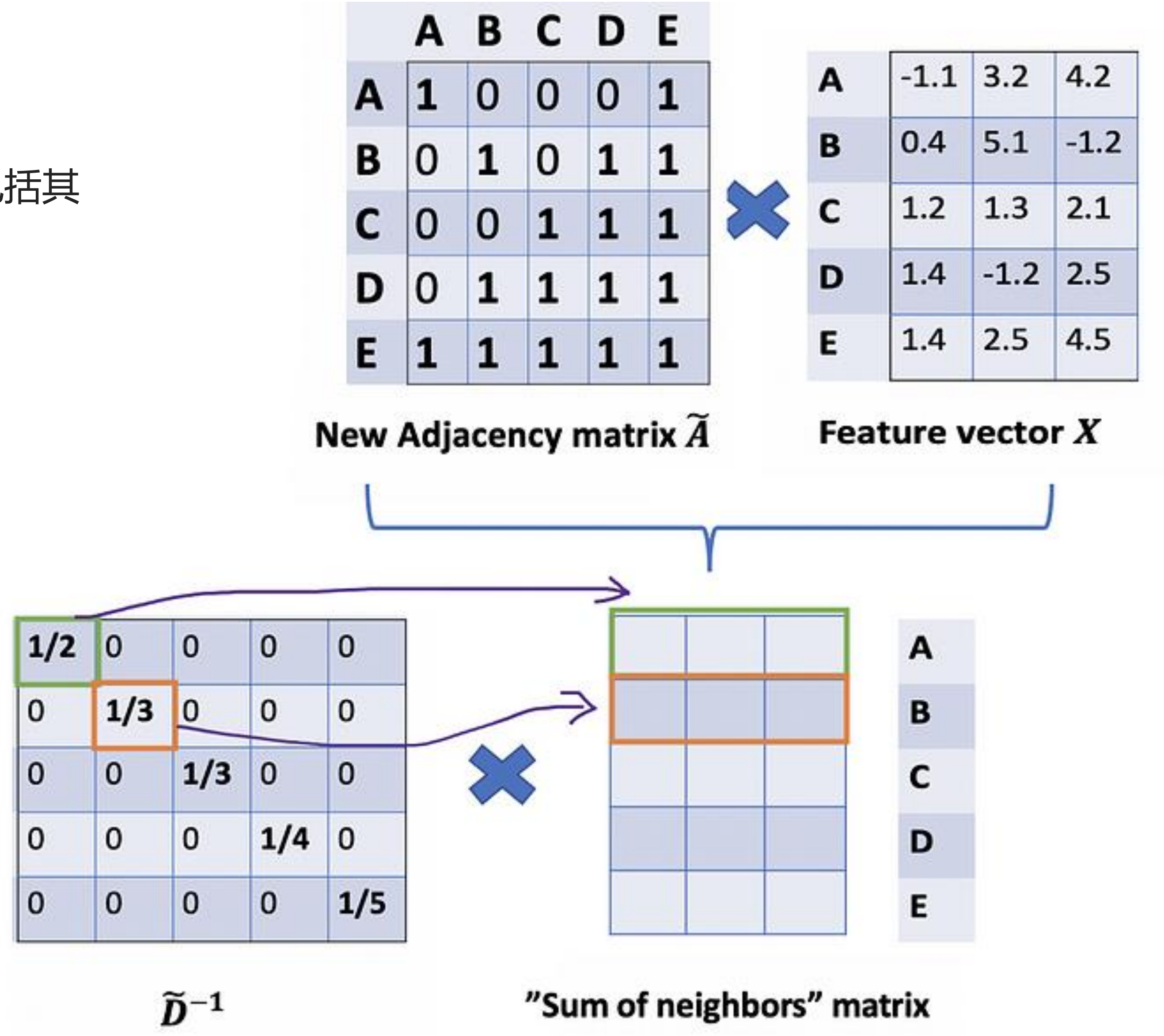
通过 $\tilde{D}^{-1}$ 与 X 的乘积，我们可以取所有邻居特征向量（包括其自身）的平均值。目前公式变成：

$$\tilde{D}^{-1}(\tilde{A}\boldsymbol{X})$$

矩阵满足结合率，所以等价于：

$$(\tilde{D}^{-1}\tilde{A})\boldsymbol{X}$$

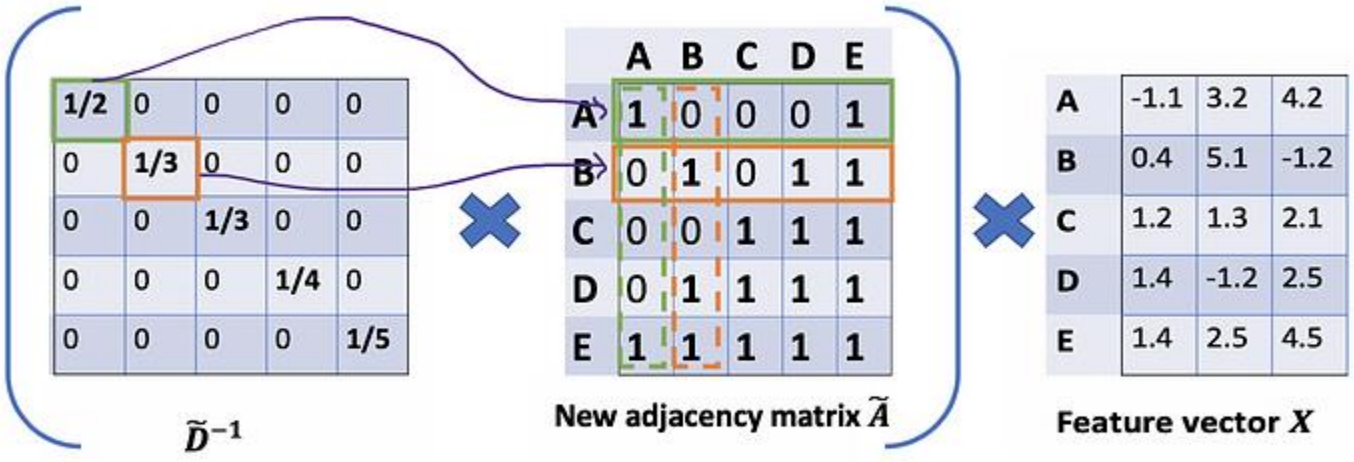
所以$\tilde{D}^{-1}$ 相当于矩阵缩放方法。

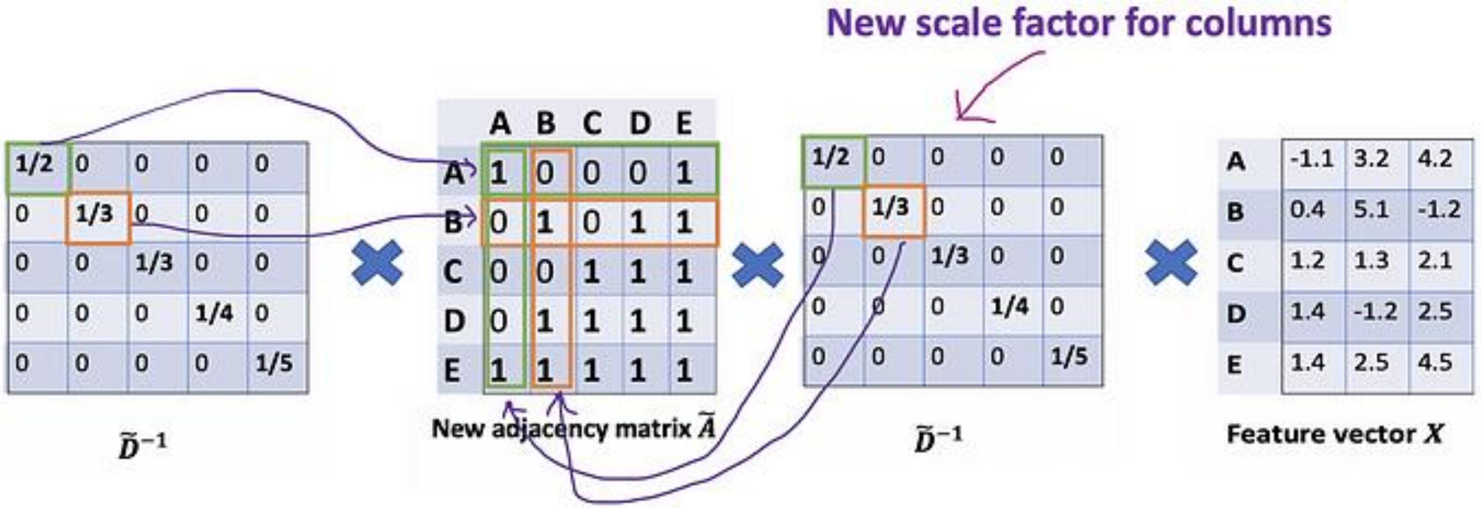


GCN - 两层简单模型 – 例子分析 - 矩阵缩放

左乘相当于对行做归一化，同理，右乘相当于对列做归一化。于是，得到公式：

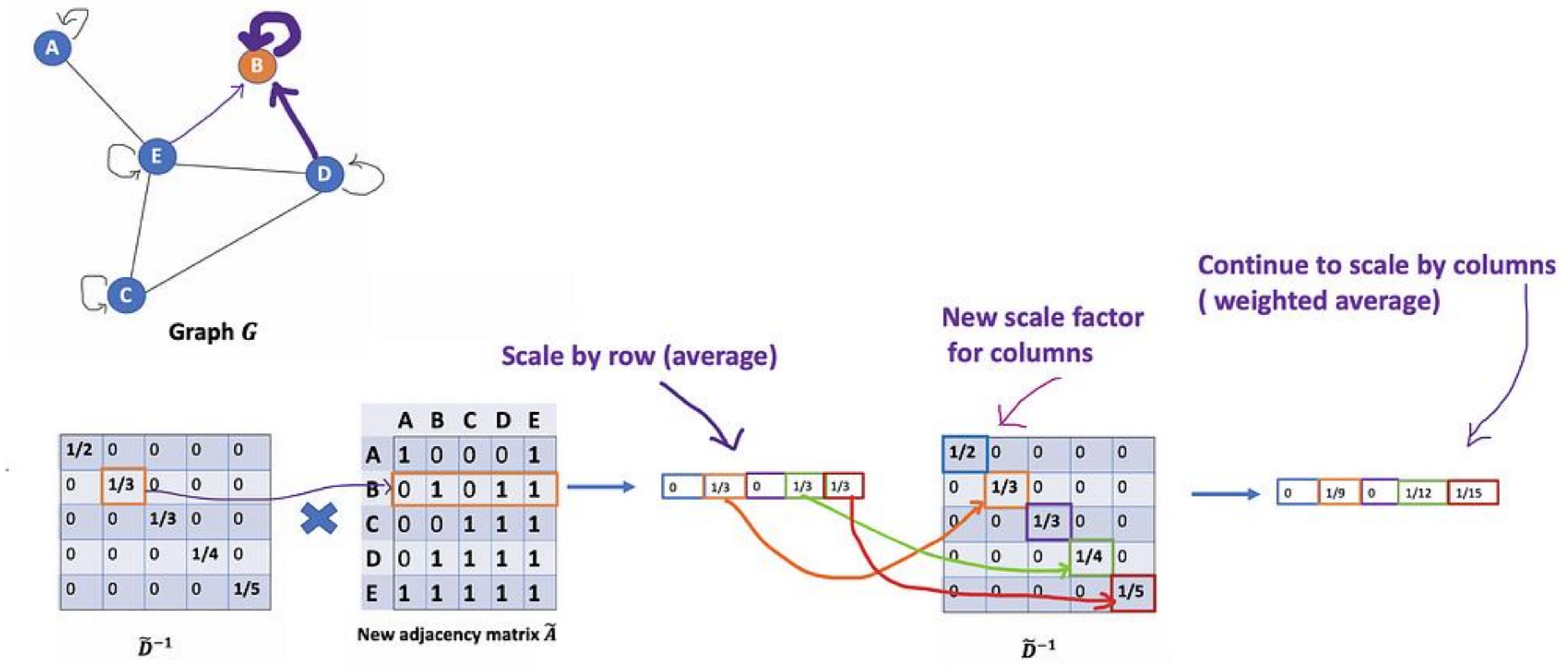
$$\tilde{D}^{-1}\tilde{A}\tilde{D}^{-1}\boldsymbol{X}$$





GCN - 两层简单模型 – 例子分析 - 矩阵缩放

新的缩放器为我们提供了“加权”平均值。我们在这里所做的就是给度数低的节点赋予更多的权重，减少度数高的节点的影响。这种加权平均的想法是，我们假设低度节点会对它们的邻居产生更大的影响，而高度节点会产生较小的影响，因为它们将影响力分散在太多的邻居上。



当聚合节点 B 的特征时，我们为节点 B 本身分配最大权重（度数为 3），为节点 E 分配最小权重（度数为 5）

GCN - 两层简单模型 – 例子分析 - 矩阵缩放

由于 $\widetilde{D}_{ii}\widetilde{D}_{jj}$ 归一化做了2次， 需要更新为 $\sqrt{\widetilde{D}_{ii}\widetilde{D}_{jj}}$，于是得到新公式:

$$\widetilde{D}^{-1/2}\tilde{A}\widetilde{D}^{-1/2}$$

2	0	0	0	0
0	3	0	0	0
0	0	3	0	0
0	0	0	4	0
0	0	0	0	5

$\widetilde{D}$

→

1/2	0	0	0	0
0	1/3	0	0	0
0	0	1/3	0	0
0	0	0	1/4	0
0	0	0	0	1/5

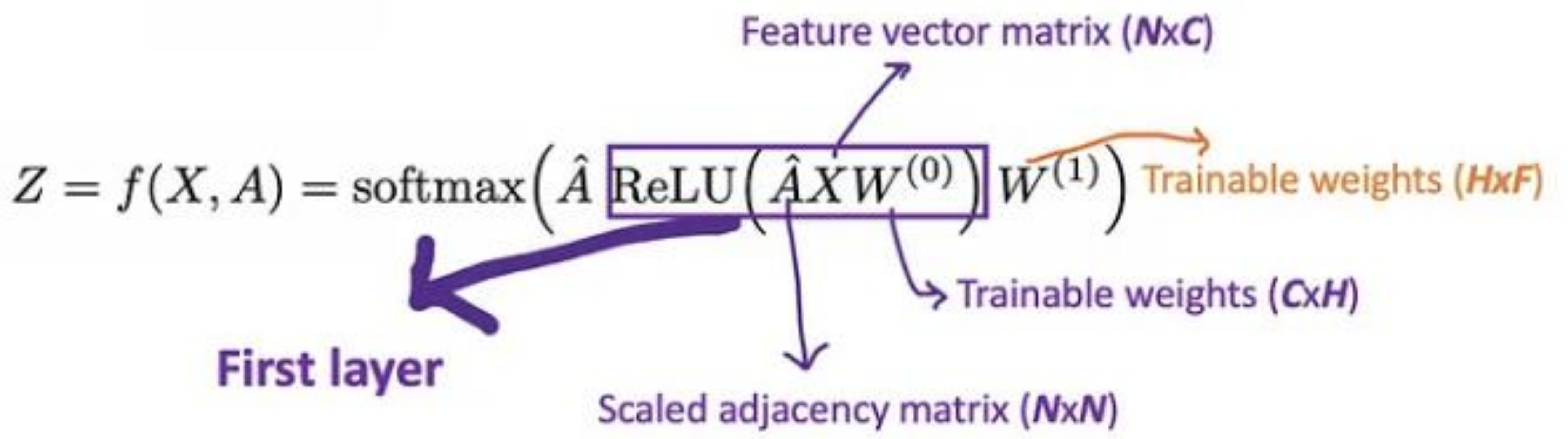
$\widetilde{D}^{-1}$

$1/\sqrt{2}$	0	0	0	0
0	$1/\sqrt{3}$	0	0	0
0	0	$1/\sqrt{3}$	0	0
0	0	0	1/2	0
0	0	0	0	$1/\sqrt{5}$

$\widetilde{D}^{-1/2}$

GCN - 两层简单模型 – 最终公式

例如，我们有一个具有 10 个类别的多分类问题，F将设置为 10。在第 2 层获得 10 维向量后，我们将这些向量通过 softmax 函数进行预测。



Thank You