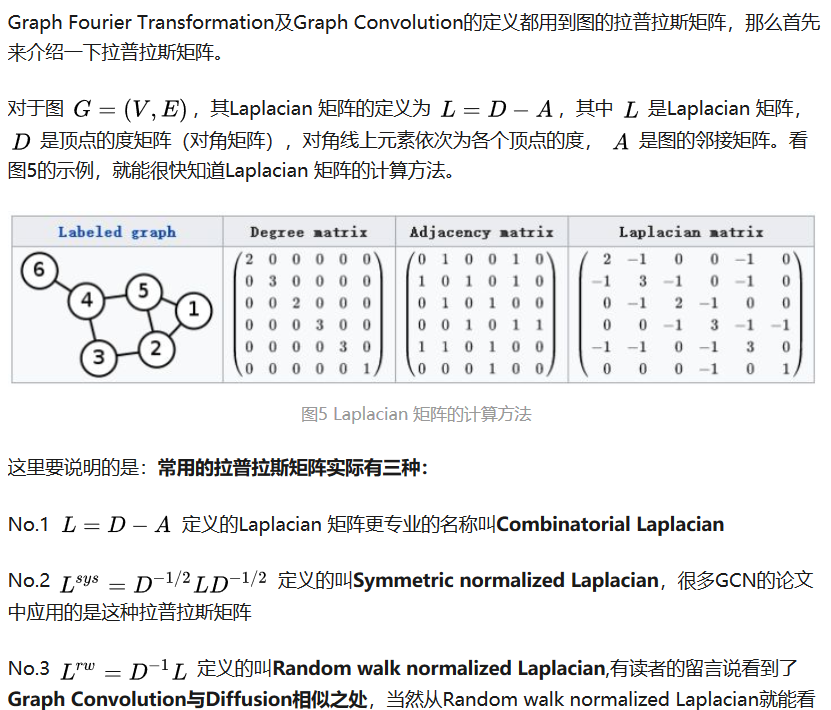
GCN <https://mp.weixin.qq.com/s/sg9O761F0KHAmCPOfMW_kQ>

GCN的本质就是提取图拓扑空间特征，，目前有两种主流方式来提取图空间特征

1. **vertex domain**(spatial domain)是非常直观的一种方式

顾名思义：提取拓扑图上的空间特征，那么就把每个顶点相邻的neighbors找出来

1. 按照什么条件去找中心vertex的neighbors，也就是如何确定receptive field
2. 按照什么方式处理包含不同数目neighbors的特征
3. **spectral domain**就是GCN的理论基础了这种思路就是希望借助图谱的理论来实现拓扑图上的卷积操作。
4. [Spectral graph theory](https://link.zhihu.com/?target=https%3A//en.wikipedia.org/wiki/Spectral_graph_theory" \t "_blank)请参考这个，简单的概括就是**借助于图的拉普拉斯矩阵的特征值和特征向量来研究图的性质**



来自知乎的解释：

<https://zhuanlan.zhihu.com/p/58178060?utm_source=wechat_session&utm_medium=social&utm_oi=685930719541465088>

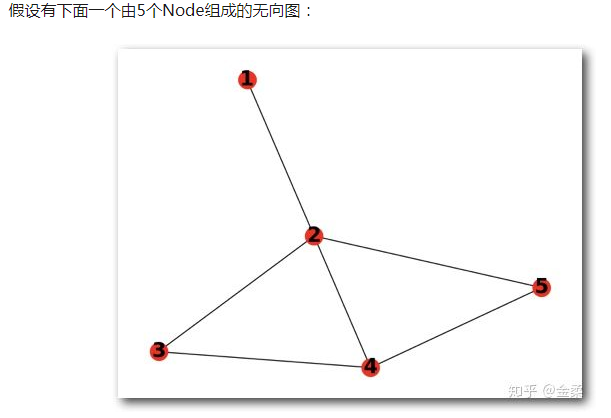
GCN下的embedding的思想是，每一个节点聚合其领域节点以及自己的特征数据作为当前节点的特征，怎么学习的：

邻接矩阵：A， 特征数据: X 单位矩阵：I 度矩阵：D

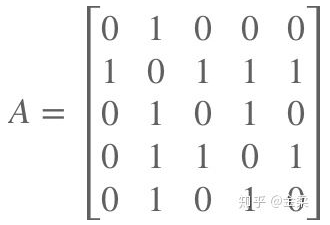
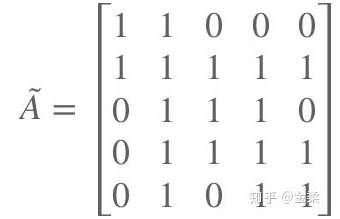
从数学上看 A\*X就可以聚合当前节点的一阶领域节点的特征和但是没有自身特征参与，

(A+I)\*X可以聚合自身节点以及邻居节点的特征，

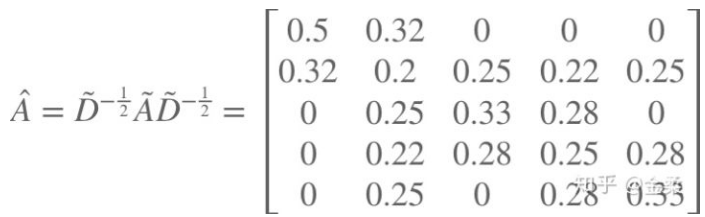
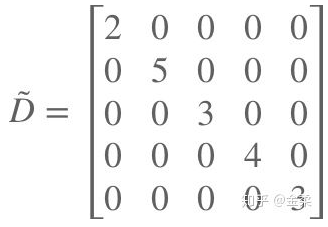
D\*\*-1\*A\*X 可以将特征归一化，这一步参考拉普拉斯矩阵分解



邻接矩阵A 邻接矩阵A的基础上加上自环（单位对角阵I）

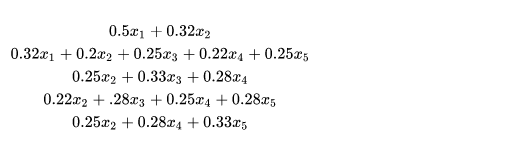
度矩阵 卷积操作



节点特征（一维）



在神经网络第一层的传播中，输入是节点的特征。为了方便，假设每个Node的特征向量是1维的，于是X=(x1,x2,x3,x4,x5)'，将特征矩阵左乘 [公式] ，即可以得到变换后的特征：



可以看到，对比一个普通的NN，GCN只是在特征上做了一个变换，而这个变换的实质就是特征通过拓扑结构进行了传播。每个节点的特征不再是自身的特征，而是自身和其邻居节点的特征加权求和。讲到这里，大家就很清楚为什么加自环的吧，其目的是需要在特征变换时保留了自身特征的贡献。

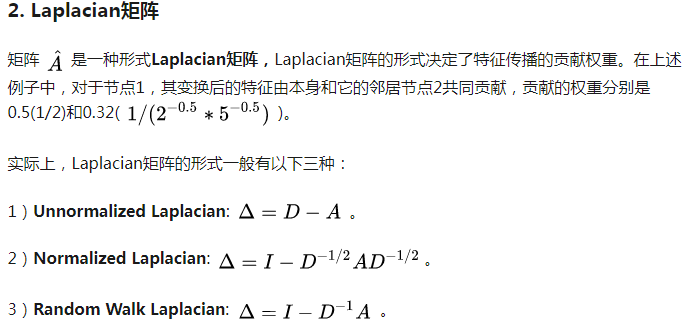
D\_hat\*\*-1 \* A\_hat \* X 已经完成了一阶领域的特征聚合，这就是图神经网络中的卷积操作

对于卷积中的感受域来说这里的感受域就是一阶邻居，全局就是整个图，经过二次卷积，感受域扩大一倍。

一个隐藏层的完整表达就是：relu(D\_hat\*\*-1 \* A\_hat \* X \* W)

D\_hat 具有自环的度矩阵， A\_hat 自环的邻接矩阵 W：权重系数， relu(): 激活函数。

在神经网络第二层的传播中，节点依然会按照上述机制进行特征传播，但这里的特征不再是节点的原始特征，而是经过变换后（传播->聚合->Emdedding->非线性变换）的特征。这个时候，每个节点可以接收到2-hop的信息，感受域进一步增加。



邻接矩阵A的基础上加上自环（单位对角阵I）现在的问题是为什么在GCN中要使用Normalized Laplacian，这里我们只从直觉上说明这个问题。

假设节点A的邻居只有B，为了对A进行特征传播，一个最直接的聚合方法是就是平均贡献，即：X=0.5A+0.5B（X=1/degree(A)\*A+1/degree(A)\*B）。这样的做法看起来很有道理，但是如果B的邻居非常非常多（极端情况是与图上其他所有点都连接），那么经过特征变换后图上很多点的特征都会非常像相似，因此显然在传播的时候需要考虑邻居的度。

因此，我们可以将传播公式改写成X=1/degree(A)\*A+1/degree(B)\*B，然而这样的问题在于若degree(B)很大，B的贡献将几乎忽略不计。综合上述两种形式，既不会让邻居B的贡献过大、也不会让邻居B的贡献过小，于是得到传播公式：

X=1/degree(A)\*A+1/(sqrt(degree(A)\*degree(B))\*B

这种传播公式对应的变换矩阵就是Normalized Laplacian。

**消息式聚合：**随着图卷积越来越火，工业界逐渐加入了基础设施建设的队伍。借鉴 GraphX 等思路，出现一些不依赖邻接矩阵（或是屏蔽了邻接矩阵细节的）的消息聚合库，比较有名的有 PyG（比较早，实现多）和 DGL（比较新，易上手）。**在这些库中，节点可以发出信息，并接受周围节点的信息，显式地完成消息聚合**。在这种情况下，越来越多复杂的聚合方法出现了

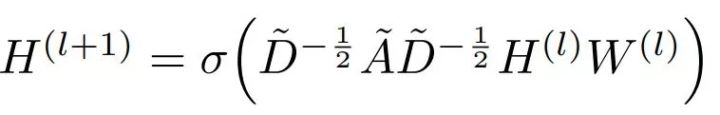
**如何优雅地理解图卷积**

图卷积是非常神奇的一个模型，它能处理图这种结构化的数据。但是如果要理解图卷积，我们又会发现它要求的数学基础似乎非常高，傅立叶变换、拉普拉斯算子等一大波数学在等着我们，这就是最初的频域思路。

阿里研究者说：「自 ICLR 2017 Kipf 的文章发表以来，图卷积才逐渐受到更多的关注，该论文从频域的角度将 CNN 转移到了 Graph，并推导出了非常简单优雅的形式。后来研究者又从空域的角度提出了 GraphSAGE，它利用直观的节点采样与特征聚合高效地生成节点向量，后面还有 Bengio 组的 GAT 与 MIT 的 jumping knowledge net。想了解图卷积的同学可以从这几篇文章入手。」

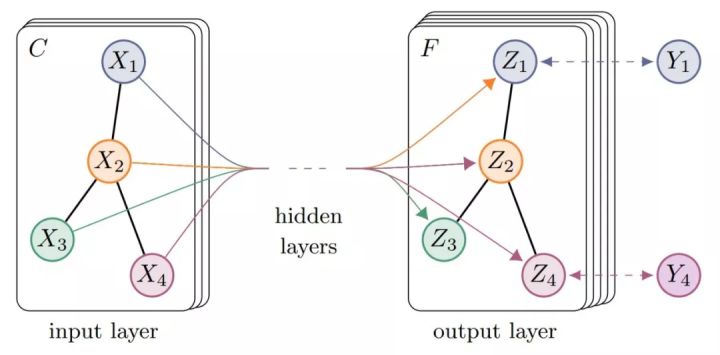
* 图卷积开山之作：Semi-Supervised Classification with Graph Convolutional Networks
* 论文地址：[https://arxiv.org/abs/1609.02907](https://link.zhihu.com/?target=https://arxiv.org/abs/1609.02907" \t "https://zhuanlan.zhihu.com/p/_blank)

图卷积的核心思想是希望**利用近邻节点的信息进行聚合而生成当前节点的新表征**，这样的节点表示可以进一步用于下游任务。如果我们直接从核心表达式出发，跳过推导过程，其实能更容易地理解。如下所示为两层图卷积网络之间的传播方法，它看起来只不过比常规的神经网络多了 D tilde 与 A tilde 这几项。



如果我们的图有 n 个节点，那么节点与节点之间的关系可以用 n\*n 的邻接矩阵表示，它再加上由节点特征向量组成的矩阵 H 就是图卷积的输入。在上式中，A tilde 以及 D tilde 就是由邻接矩阵算出来的东西，它对于同一张图是不变的，因此可以预先计算好。

现在，剩下的 H×W 就是输入Embedding H经过一层全连接层了，以这样方式进行层级传播的卷积网络就是图卷积，我们可以**将传播理解为每个节点拿到邻居节点信息，并聚合到自身嵌入向量上**。



如上图所示，图卷积网络的输入是表示节点及边的特征向量，经过一系列隐藏层的变换，可以计算出每个节点的深度表征。这样的 Z 再来做预测或生成就会非常有效。直观而言，图卷积将图片的 RGB 像素值替换成节点特征，并且通过边的关系引入了邻居的概念，完成卷积运算。