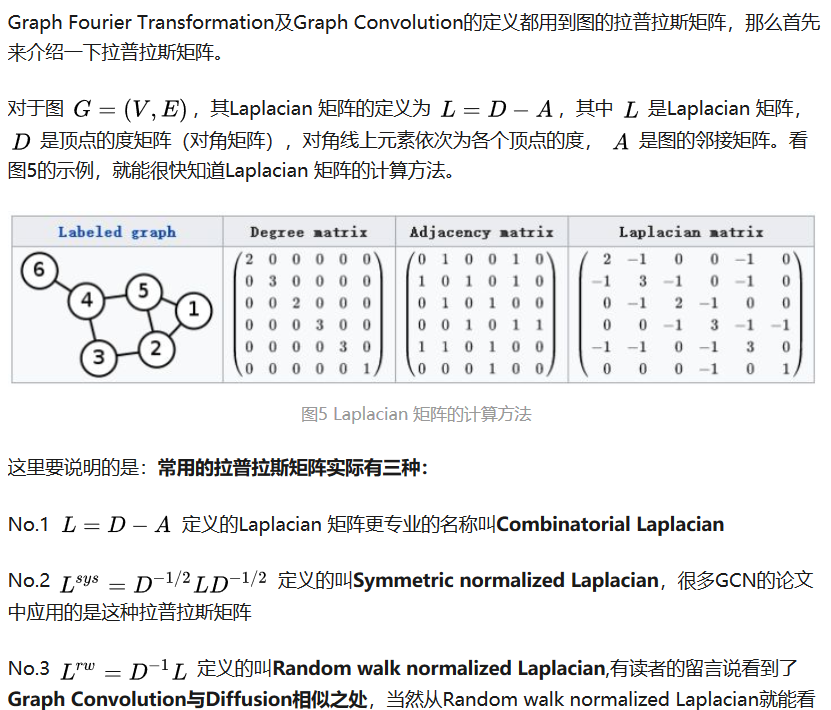
GCN <https://mp.weixin.qq.com/s/sg9O761F0KHAmCPOfMW_kQ>

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，

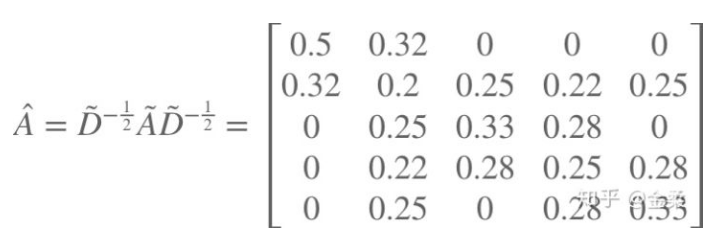
(A+I)\*X可以聚合自身节点以及邻居节点的特征，

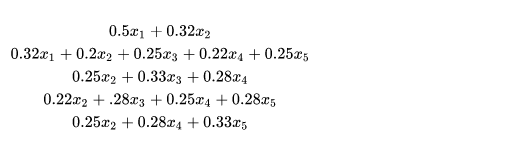
D\*\*-1\*A\*X 可以将特征归一化，这一步参考拉普拉斯矩阵分解

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。

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。

**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