问题描述

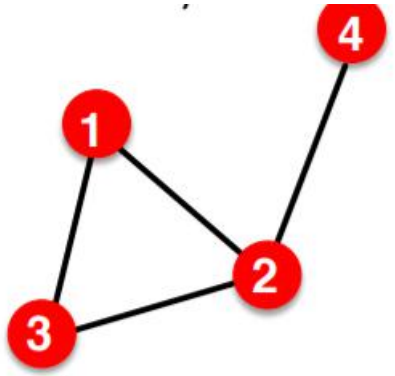
- 假设现在有一个图G：
  - V是**顶点集**　（ |V|=N，其中有N个顶点 ）
  - A是**邻接矩阵**（0代表没边，1代表有边）
  - X是一个m*|V|大小的矩阵，代表**节点特征**，每个节点有**m维的特征**
  - 每个节点的特征：
    - 对社交网络：用户资料（年龄、学历、地区），用户画像等
    - 生物信息网络：基因表达图谱，基因功能信息等
    - 节点没有特征：可以使用one-hot编码，使每个节点的特征都不同
- 我们想做的任务可以是节点分类，连接预测，整图分类等

一些例子

- 根据**节点特征**分类：

  四位同学的成绩　　A：20，B：30，C：40，D：80

- 根据**结构信息**分类：



$$A_{ij}=\begin{pmatrix}0&1&1&0\\1&0&1&1\\1&1&0&0\\0&1&0&0\end{pmatrix}$$

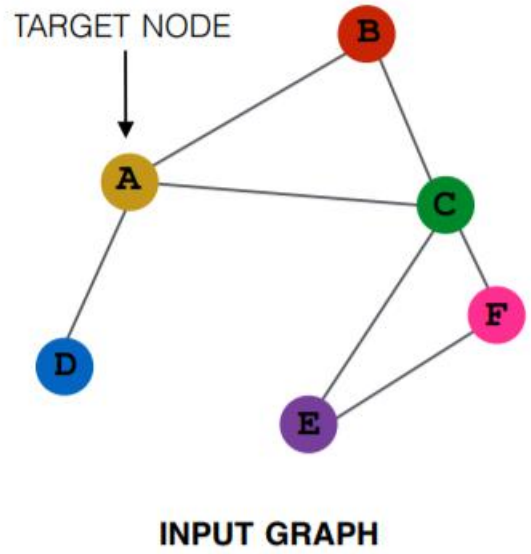
-

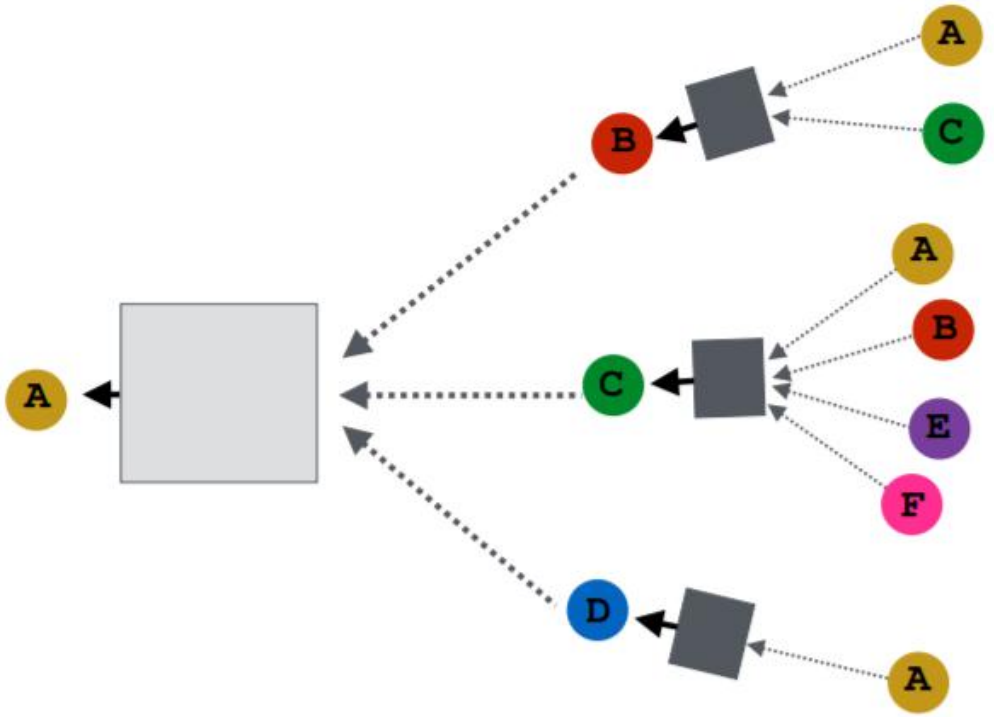
GCN同时利用结构信息和节点特征进行分类

GCN

- 关键思想：基于附近的网络邻居生成节点的嵌入表示

例 目标：生成目标节点A的嵌入表示

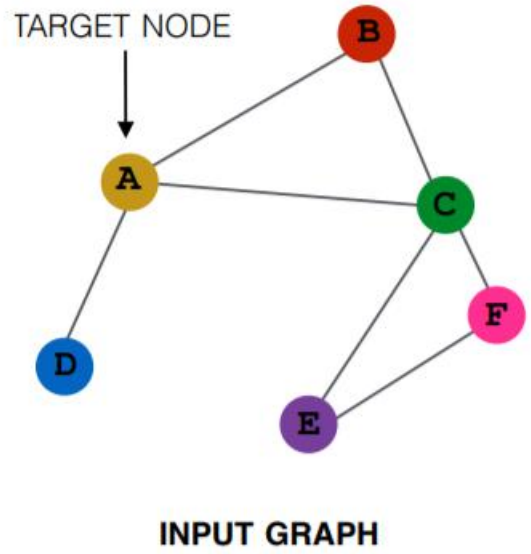


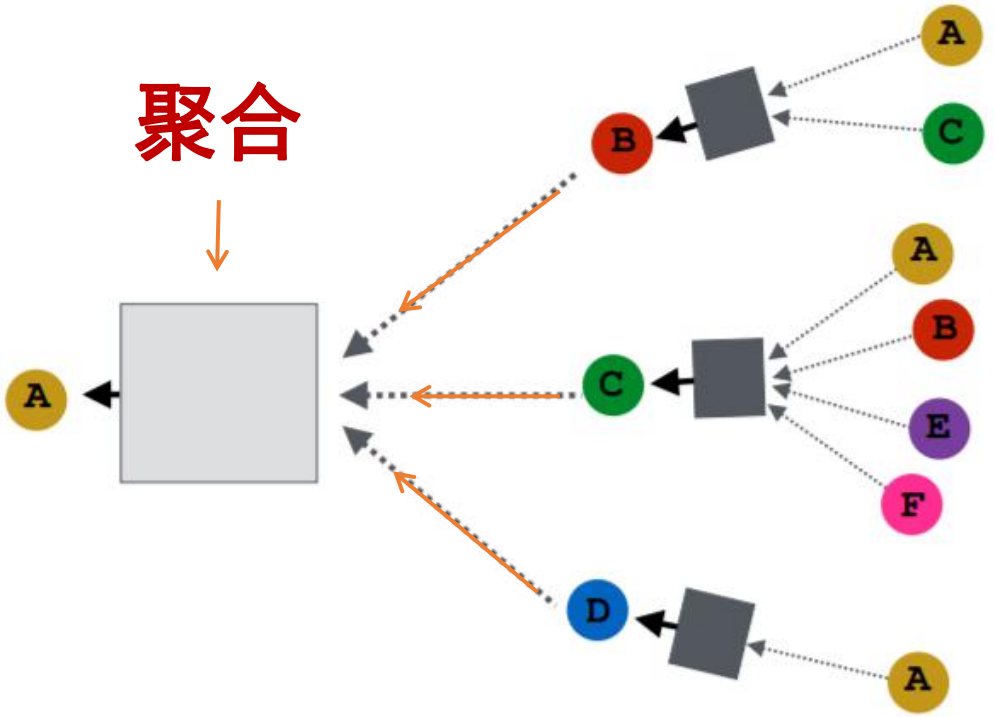


GCN

- 关键思想：基于附近的网络邻居生成节点的嵌入表示

例 目标：生成目标节点A的嵌入表示

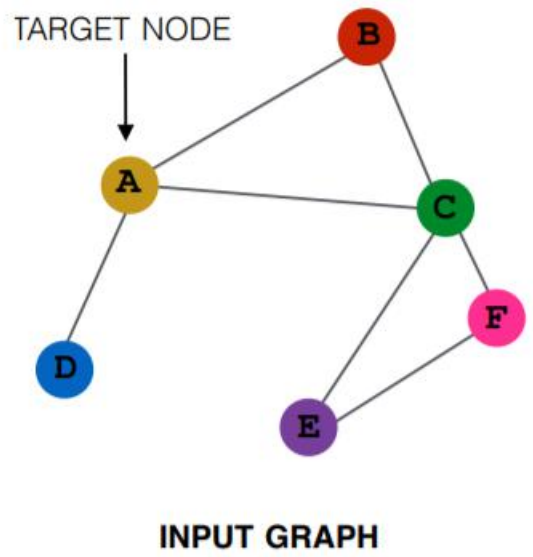




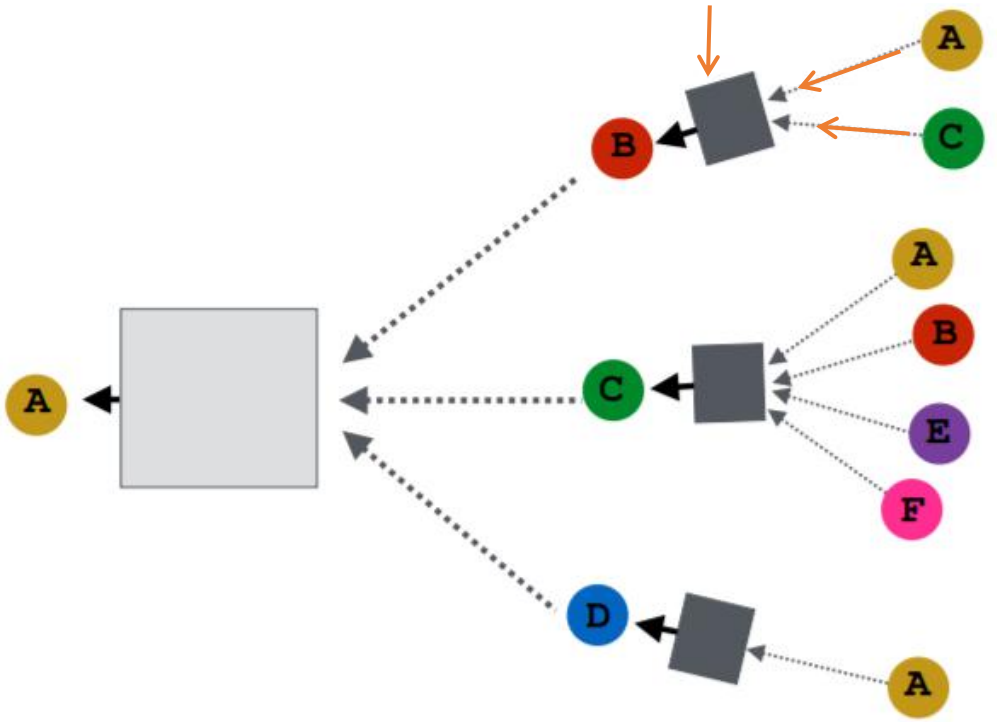
GCN

- 关键思想：基于附近的网络邻居生成节点的嵌入表示

例 目标：生成目标节点A的嵌入表示



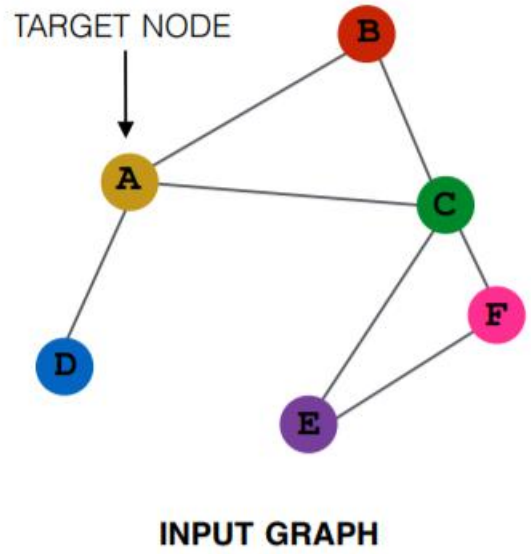
聚合



GCN

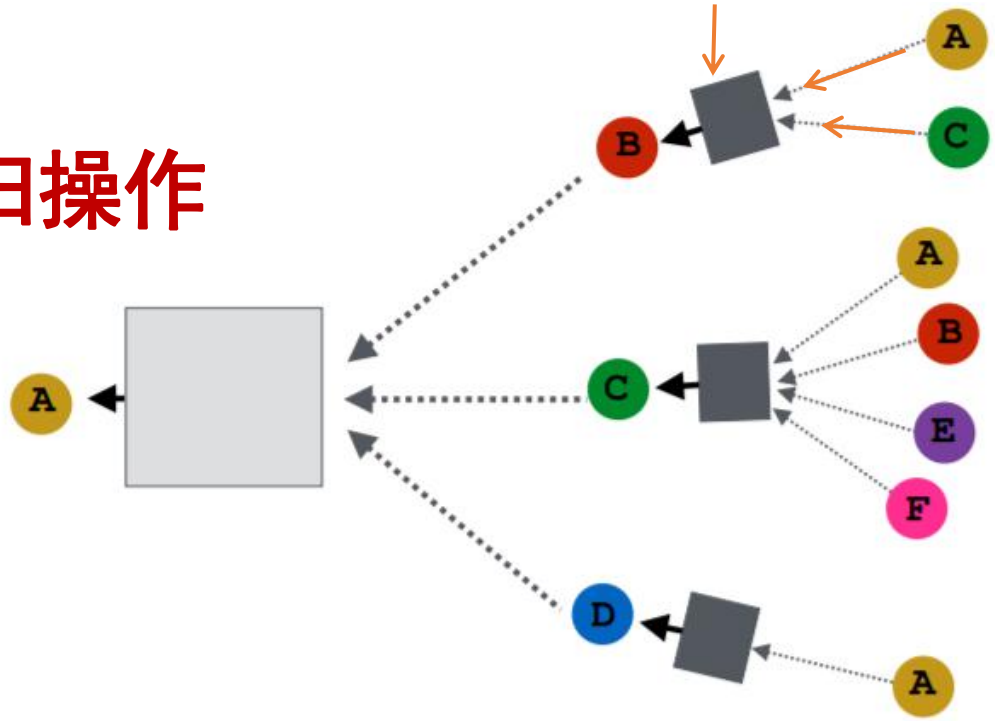
- 关键思想：基于附近的网络邻居生成节点的嵌入表示

例 目标：生成目标节点A的嵌入表示



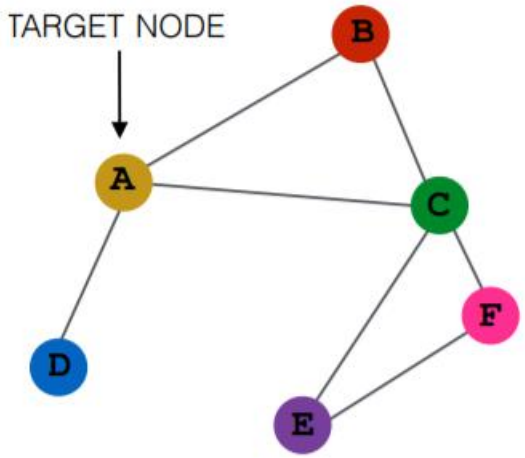
聚合

递归操作

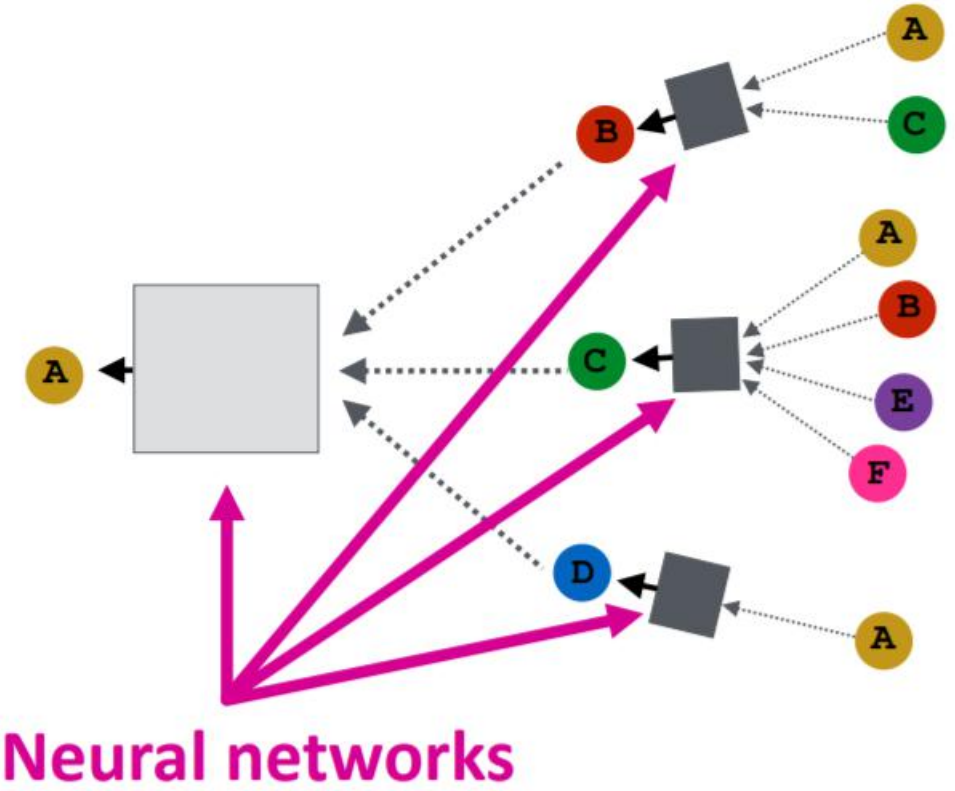


GCN

- 想法：节点使用基于**神经网络**的方法来聚合邻居的信息



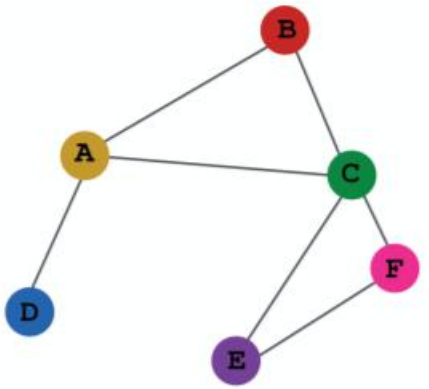
INPUT GRAPH



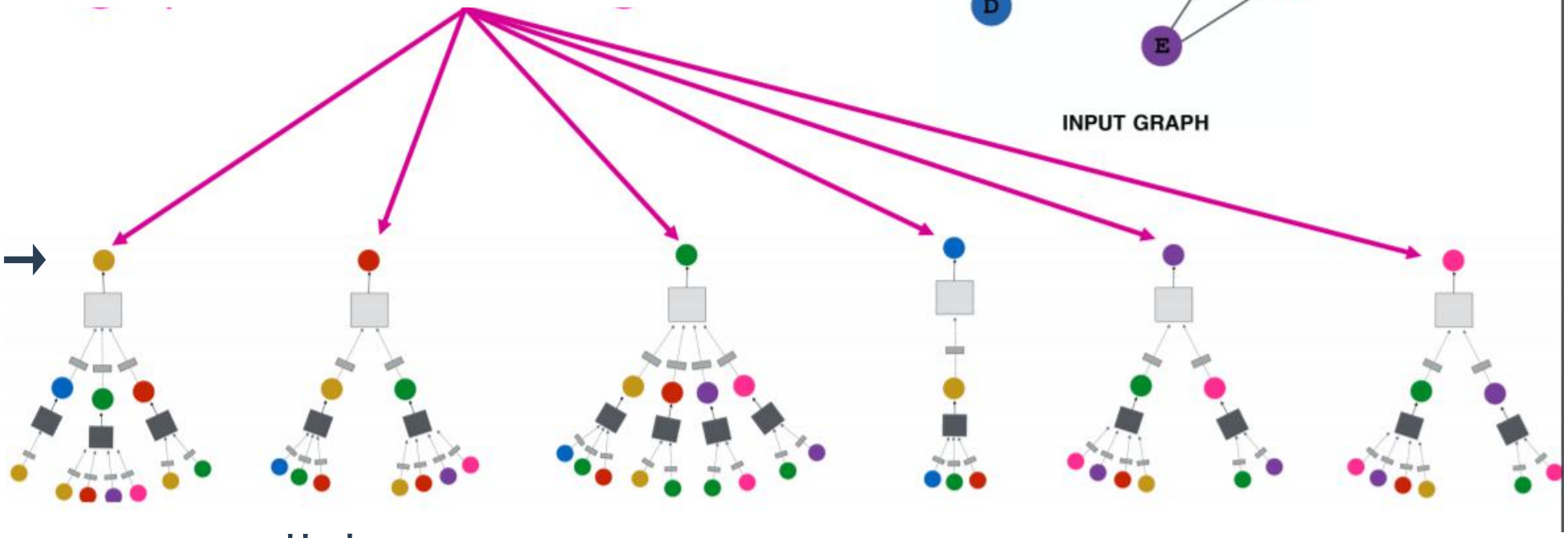
GCN

每个节点都有基于邻居的计算图

每个节点都根据邻居、
邻居的邻居来聚合信息



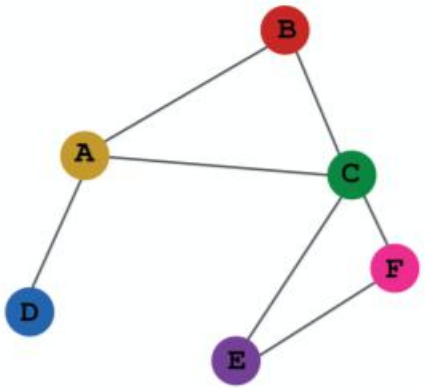
INPUT GRAPH



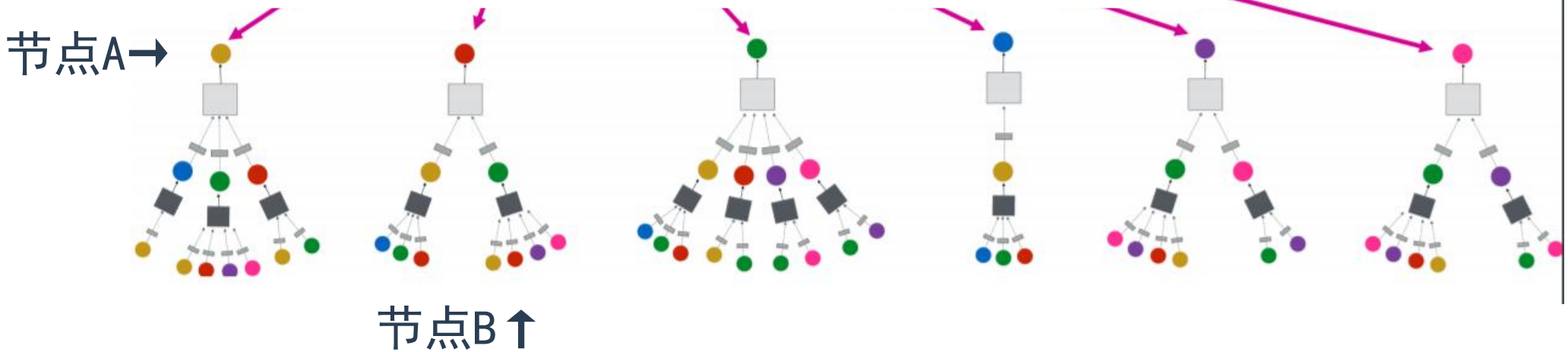
GCN

每个节点都有基于邻居的计算图

每个节点都根据邻居、
邻居的邻居来聚合信息



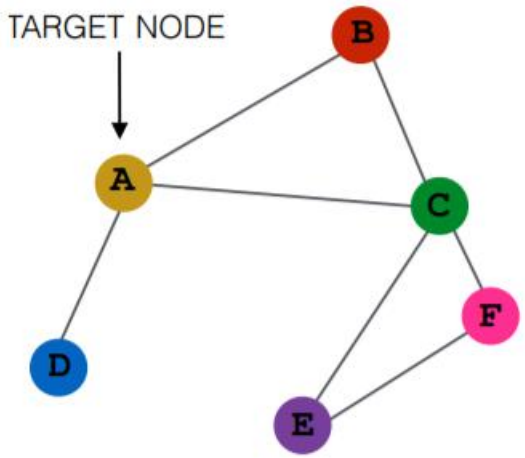
这里衍生成一个有两层的神经网络



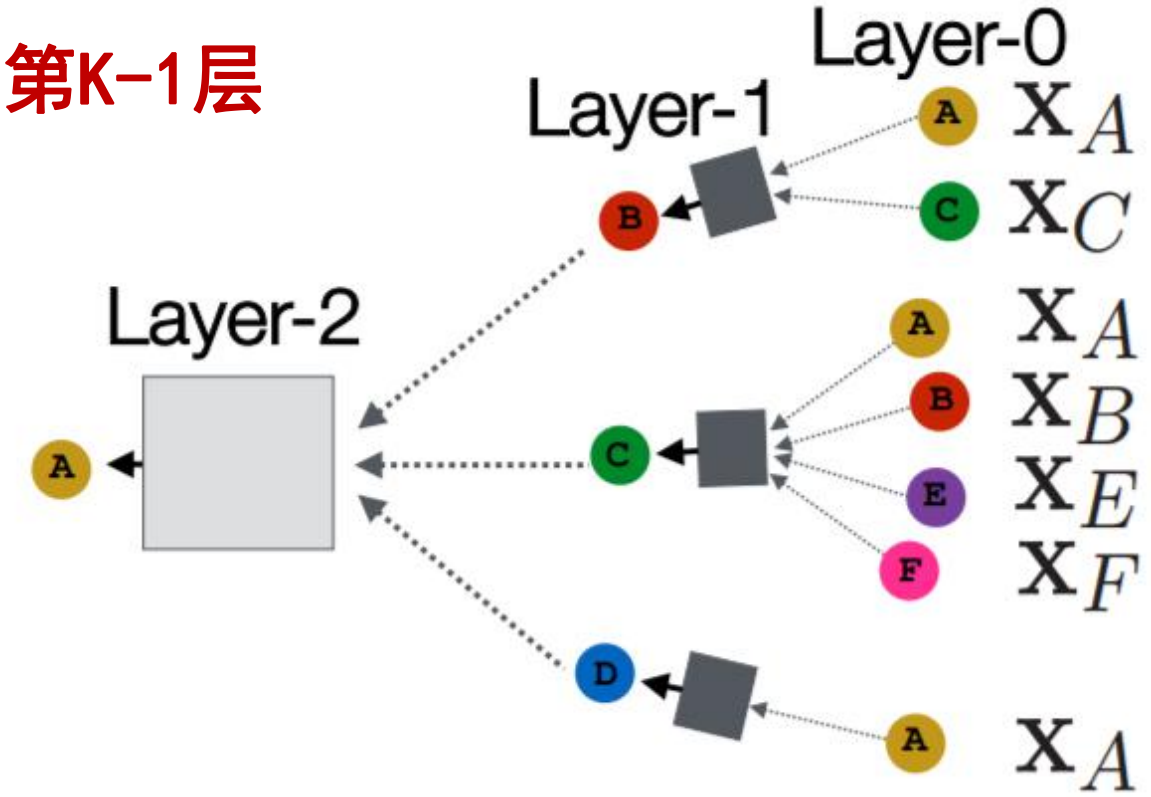
GCN可以有很多层网络

- 模型可以是**任何深度**
  - 节点在每一层都有嵌入表示
  - 节点在第**0**层的嵌入表示就是**节点特征X**
  - 第**K**层的节点的嵌入表示是**聚合第K-1层**

邻居节点的信息来计算得到的



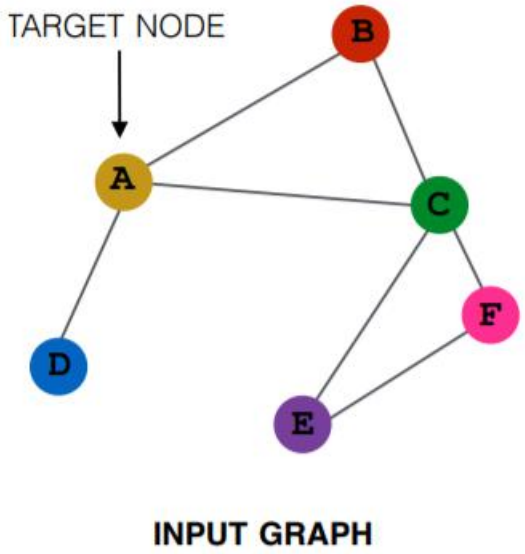
INPUT GRAPH



GCN可以有很多层网络

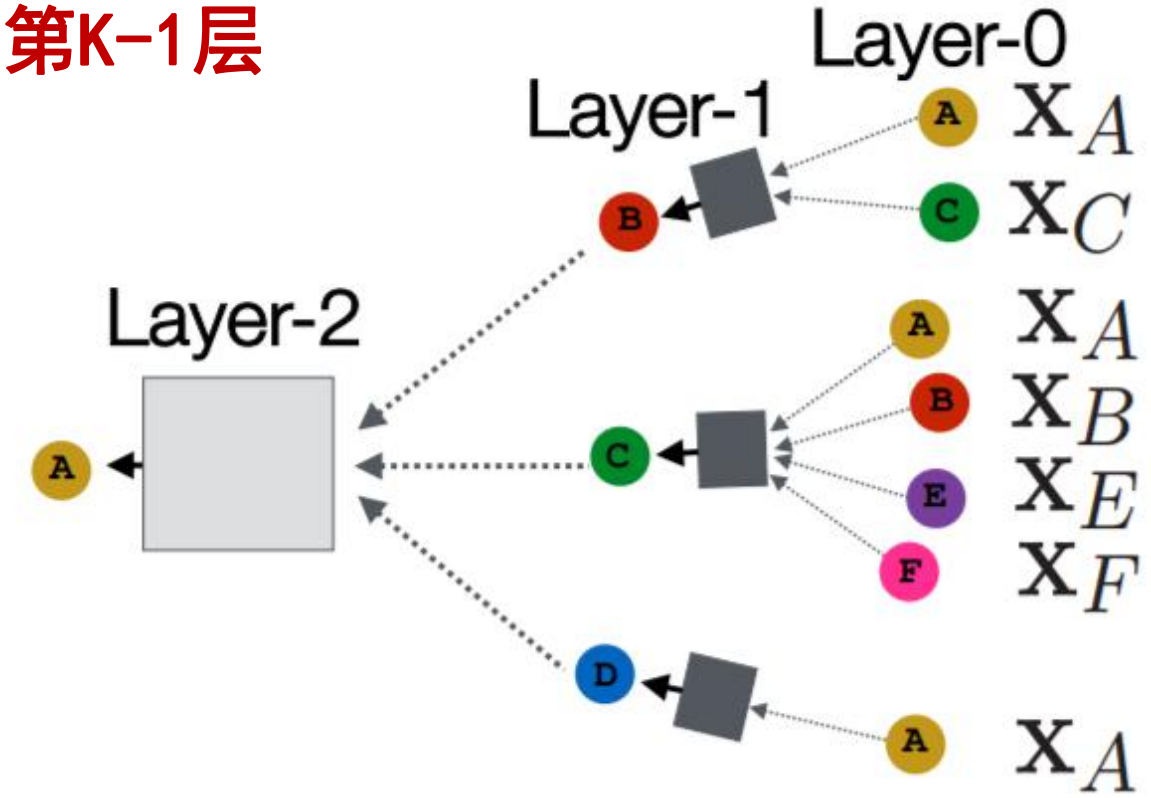
- 模型可以是任何深度
  - 节点在每一层都有嵌入表示
  - 节点在第0层的嵌入表示就是节点特征X
  - 第K层的节点的嵌入表示是聚合第K-1层

邻居节点的信息来计算得到的



第3层

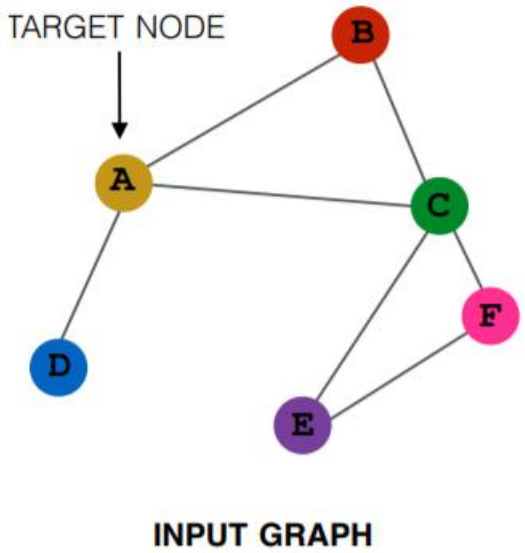
D←



GCN可以有很多层网络

- 模型可以是任何深度
  - 节点在每一层都有嵌入表示
  - 节点在第0层的嵌入表示就是节点特征X
  - 第K层的节点的嵌入表示是聚合第K-1层

邻居节点的信息来计算得到的



在此图中生成目标节点A的嵌入表示，只需要一个两层的网络就可以聚合到所有节点的特征

6度空间理论→网络深度不会无限增加

GCN聚合

- 聚合邻居信息：关键的区别是如何跨层聚合邻居的信息

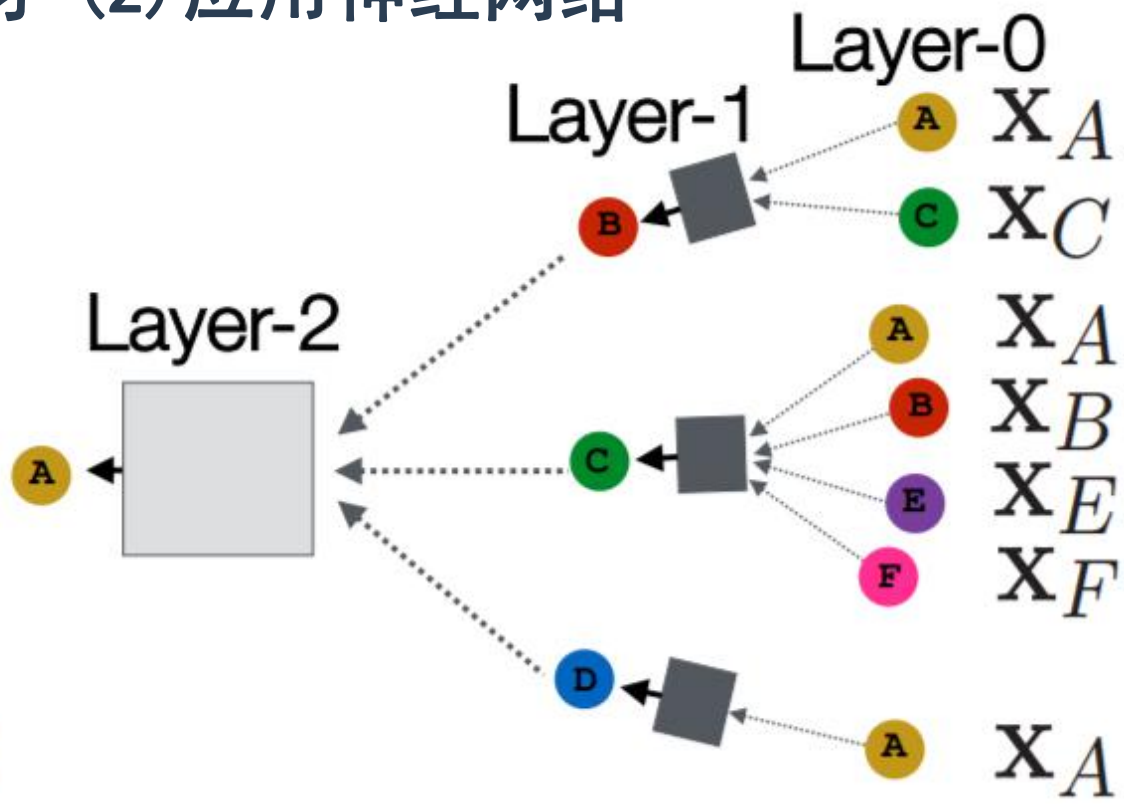
基本方法：(1)对来自邻居的信息取平均 (2)应用神经网络

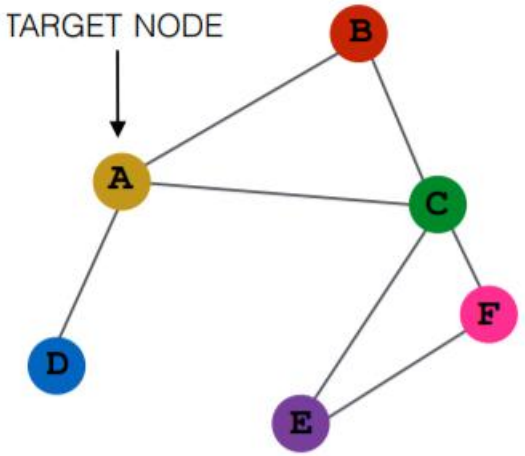
F*m m*1

对节点B：$W_1*(X_A+X_C)/2$ F*1

对节点C：$W_1*(X_A+X_B+X_E+X_F)/4$

同一层的W相同，
同一层可以共享参数





INPUT GRAPH

GCN聚合

- 聚合邻居信息：关键的区别是如何跨层聚合邻居的信息

基本方法：(1)对来自邻居的信息取平均 (2)应用神经网络

F*m m*1

对节点B：$W_1*(X_A+X_C)/2$ F*1

对节点C：$W_1*(X_A+X_B+X_E+X_F)/4$

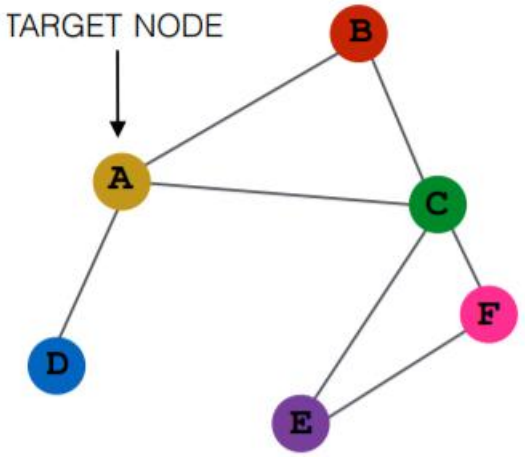
同一层的W相同，
同一层可以共享参数

A*F F*1

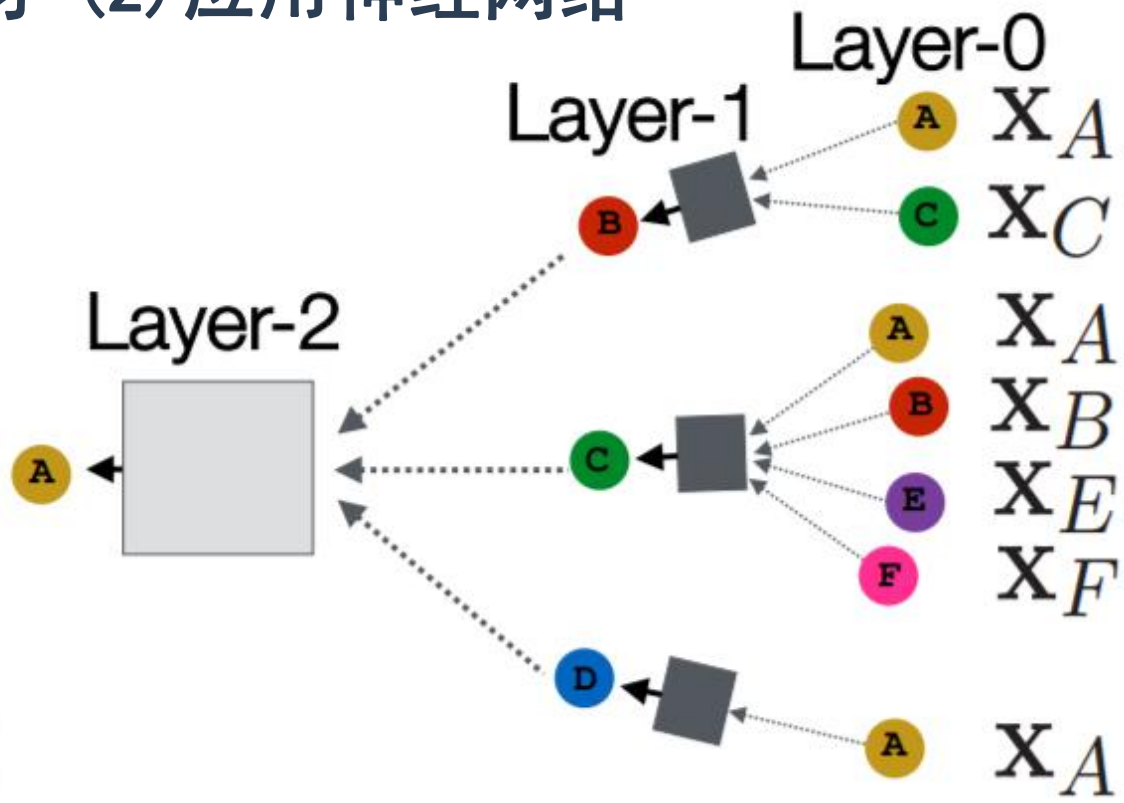
对节点A：$W_2*(X_B+X_C+X_D)/3$

A*1

不同层的W不同



INPUT GRAPH



数学表示

- 基本方法：对来自邻居的信息取平均并且应用神经网络

第0层的嵌入值=第0层节点特征

$$h_v^0 = x_v$$

神经网络可学习的权重矩阵

节点本身上一层的信息

$$h_v^{(l+1)} = \sigma\left(W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l h_v^{(l)}\right), \forall l \in \{0, \ldots, L-1\}$$

$$z_v = h_v^{(L)}$$

对来自邻居的信息做平均
邻居嵌入表示的加和/邻居个数

总层数

最后一层的节点表示
即最终的节点特征

激活函数

模型参数

等价表示，向量化地表示

$$h_v^{(0)} = x_v$$

$$h_v^{(l+1)} = \sigma\left(W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l h_v^{(l)}\right), \forall l \in \{0, \ldots, L-1\}$$

$$z_v = h_v^{(L)}$$

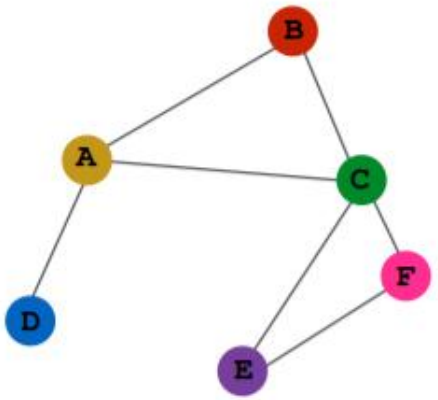
$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{H}^{(l)}\mathbf{W}_0^{(l)} + \tilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}_1^{(l)}\right)$$

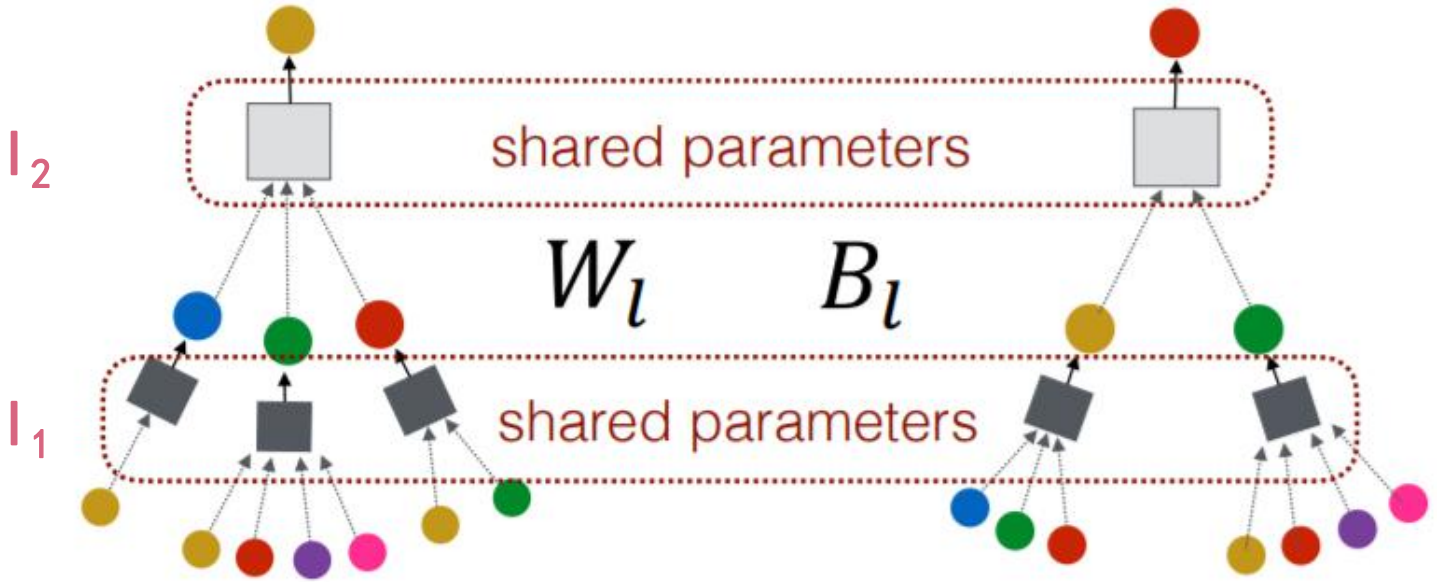
with $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$

度矩阵 对角线矩阵

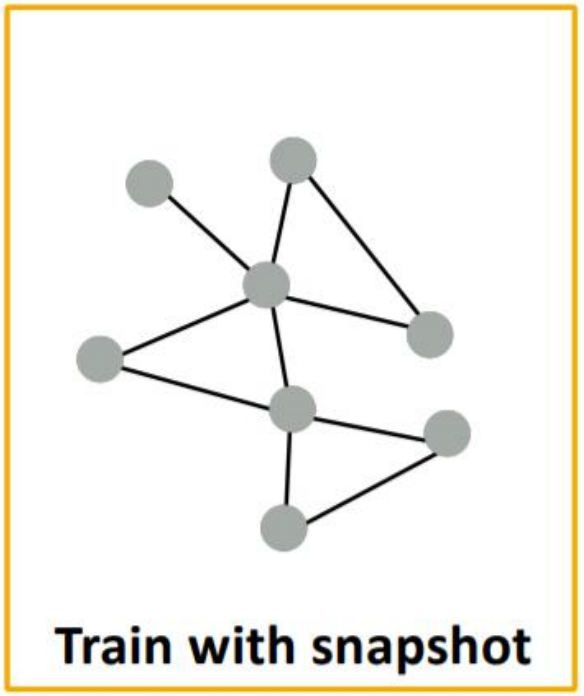
参数共享

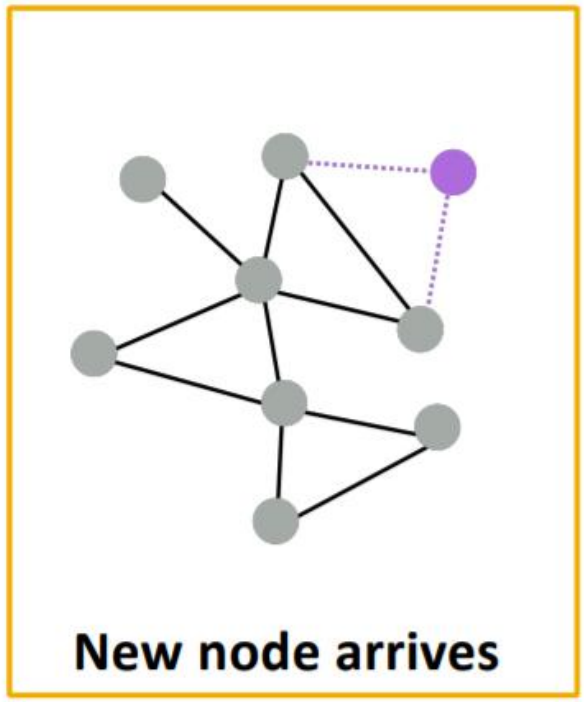
- 同一层中的节点共享共同的可学习参数W和B，并且由于参数共享，模型可以对应图中新加入的节点，而这是经典的图机器学习方法做不到的。

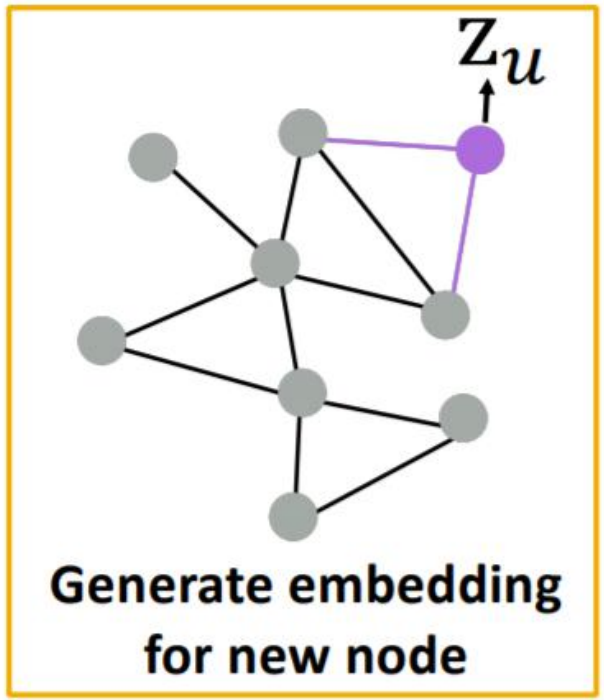




泛化能力







图中有新的节点加入，已训练出的模型可以直接泛化到新加入的节点上，用已知的参数为新节点生成嵌入