A Single Layer of a GNN

一个GNN层包含以下组件:

1) message function: $m_u^{(l)} = MSG^{(l)}(h_u^{(l-1)})$,

每个节点基于嵌入产生信息并发出

2) Aggregation: $h_v^{(l)} = AGG^{(l)}(\{m_u^{(l)}, u \in N(v)\})$

聚合接收到的来自邻居的信息,并更新特征

这种模式的缺点是每次进行特征更新时会抛弃原特征。

通过在 message function 或 aggregation 中引入邻居节点/原节点特征解决:

1) message function: $m_u^{(l)} = MSG^{(l)}(h_u^{(l-1)}), u \in \{N(v) \cup v\}$

2) aggregation: $h_v^{(l)} = concat(AGG(\{m_u^{(l)}, u \in N(v)\}), m_v^{(l)})$

同时在 message function/aggregation 中加入非线性变换以提升表现力。

因此,正式来说,相较于 $h_v^{(l)} = \sigma(W^{(l)} \sum_{u \in N(v)} \frac{h_u^{(l-1)}}{|N(v)|})$

只对嵌入进行聚合,图卷积(GCN)层增加了生成消息的过程 $h_v^{(l)} = \sigma(\underbrace{\sum_{u \in N(v)} W^{(l)} \frac{h_u^{(l-1)}}{|N(v)|}}_{\text{对消息求和并归一化}})$

日期:　　/

另一个结构GraphSAGE的改进之处在于引入了任意的聚合函数而非只是求均值，同时在更新时考虑原特征：

$$h_v^{(l)} = \sigma(W^{(l)} \cdot \text{concat}(h_v^{(l-1)}, AGG(\{h_u^{(l-1)}, \forall u \in N(v)\})))$$

可用的AGG(·)包括：

Mean : $AGG = \sum_{u \in N(v)} \frac{h_u^{(l-1)}}{|N(v)|}$

Pool : $AGG = \text{Mean}(\{MLP(h_u^{(l-1)}), \forall u \in N(v)\})$

...

此外，GraphSAGE引入了L2归一化的概念：

$$h_v^{(l)} \leftarrow \frac{h_v^{(l)}}{\|h_v^{(l)}\|_2}, \forall v \in V \text{ where } \|u\|_2 = \sqrt{\sum_i u_i^2} \ (L_2 \text{ norm})$$

h_v 将保持单位长度

以上方式的缺点在于，在进行消息聚合时，各邻居节点有相同的权重。

图注意力网络(GAT)在aggregation中引入注意力机制：

$$h_v^{(l)} = \sigma(\sum_{u \in N(v)} \alpha_{vu} W^{(l)} h_u^{(l-1)})$$

注意力权重

α_{vu} 是定义的注意力机制的产物：

定义的注意力机制

1) 首先计算注意力分数：$e_{vu} = a(W_q^{(l)} h_v^{(l-1)}, W_k^{(l)} h_k^{(l-1)})$

e_{vu}表示 u点信息对v的重要性

日期:　　　/

2) 归一化，计算注意力权重:

$$\alpha_{vu} = \frac{\exp(e_{vu})}{\sum_{k\in N(v)} \exp(e_{vk})}$$

3) 基于注意力权重对消息进行加权

$$h_v^{(l)} = \sigma\left(\sum_{u\in N(v)} \alpha_{vu} W^{(l)} h_u^{(l-1)}\right)$$

而注意力机制a有许多可行选择:

如 $a(W_q^{(l)} h_v^{(l-1)}, W_k^{(l)} h_v^{(l-1)}) = \mathrm{Linear}(\mathrm{concat}(W_q^{(l)} h_v^{(l-1)}, W_k^{(l)} h_v^{(l-1)}))$

此外可使用多头注意力机制多次捕获注意力权重:

可以稳定训练过程，易于收敛

$$h_v^{(l)}[1] = \sigma\left(\sum_{u\in N(v)} \alpha_{vu}^{1} W^{(l)} h_u^{(l-1)}\right)$$

$$h_v^{(l)}[2] = \sigma\left(\sum_{u\in N(v)} \alpha_{vu}^{2} W^{(l)} h_u^{(l-1)}\right)$$

$$h_v^{(l)}[3] = \sigma\left(\sum_{u\in N(v)} \alpha_{vu}^{3} W^{(l)} h_u^{(l-1)}\right)$$

$$h_v^{(l)} = \mathrm{AGG}(h_v^{(l)}[1], h_v^{(l)}[2], h_v^{(l)}[3])$$

日期:　　/

(Batch Normalization)

在GNN的训练过程中引入批量归一化可以稳定训练过程。

1) $\mu_j = \frac{1}{N}\sum_{i=1}^{N} X_{i,j}$ 节点嵌入　　2) $\hat{X}_{i,j} = (X_{i,j} - \mu_j)/\sqrt{\sigma_j^2+\varepsilon}$

$\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(X_{i,j}-\mu_j)^2$　　$Y_{i,j} = \gamma_j \hat{X}_{i,j} + \beta_j$

可学习参数

Dropout 可以避免模型过拟合（用于消息传递中的线性层）