GAT reading notes (ICLR2018)

Abstract

整体来说就是在GCN这类非谱方法的基础上引入attention机制，基本思想是根据每个节点在其部分邻节点的attention，来对节点表示进行更新

对象：graph-structured data（arbitrarily structured）

方法：masked self-attentional layers

成就：同时解决了多个基于谱方法的GNN的关键挑战，并将模型应用于归纳（inductive）问题以及转导（transductive）问题

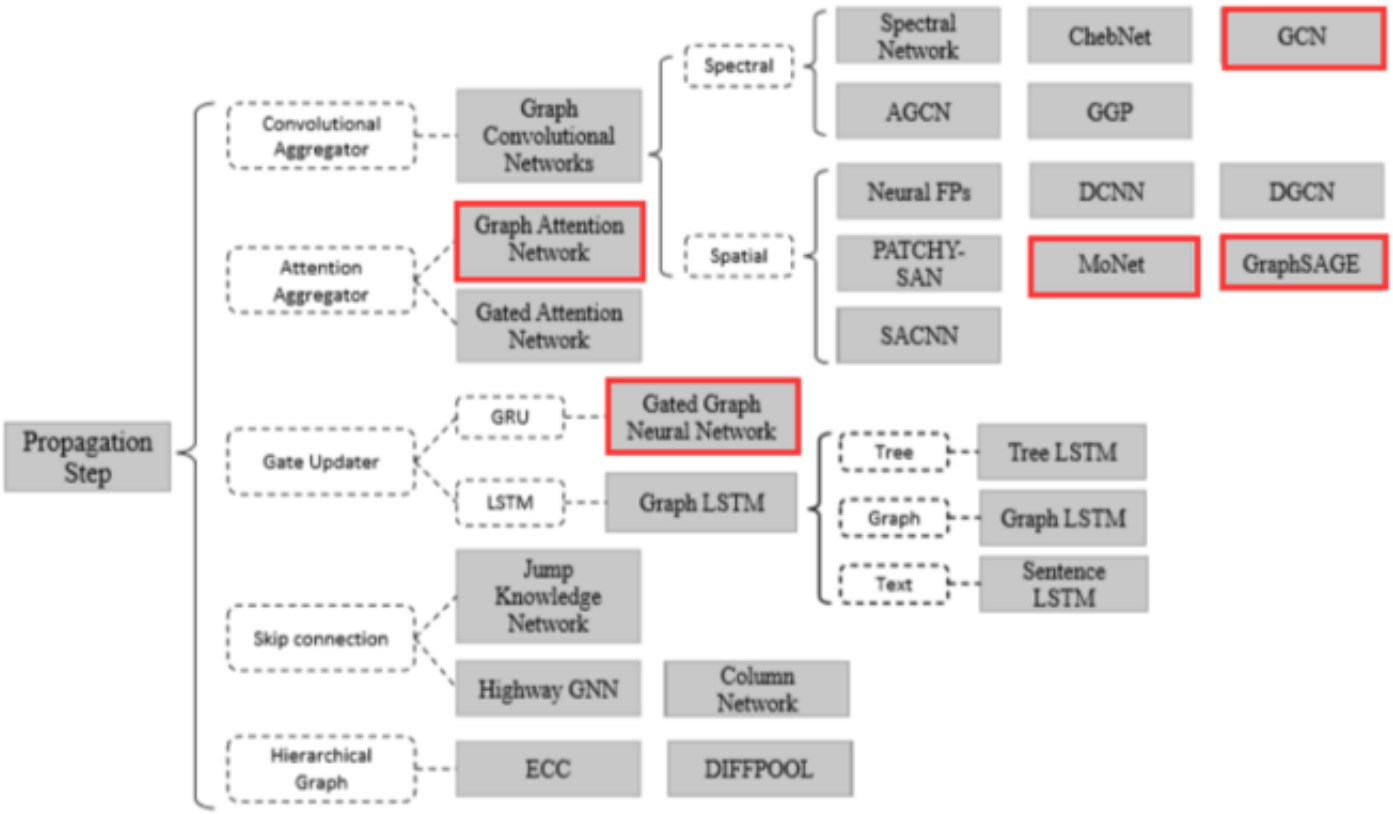
Introduction

大概就是处理graph-structured data的方法大全

文章从RNN拓展（能解决有向非循环图）说起，指出GNN（能解决普遍的图）的诞世，然后由于门控循环单元（GRU）的提出，产生了GGNN，接着说道把卷积泛化到图域，就产生了两个派别，谱方法（频域）和非谱方法（空间域）。一般的谱方法由于要进行拉普拉斯矩阵的特征值分解，计算量大且非局部计算，就诞生了GCN，但是谱方法有个致命点，就是模型依赖图结构，在特定图结构上训练的模型往往不能用于其他图结构。所以就产生了非谱方法，例如MoNet和GraphSAGE，但是它们不能较好处理可变大小的邻居和共享参数这类问题。所有就有了GAT，GAT具有以下三个特点：

- 计算速度快，可以在不同的节点上进行并行计算
- 可以同时对拥有不同度的节点进行处理
- 可以被直接用于解决归纳学习问题，即可以对从未见过的图结构进行处理

以下这张图就可以说明想表达的GNN的辉煌历史，更多内容具体参考*Graph Neural Networks: A Review of Methods and Applications*



(c) Propagation Steps

GAT Architecture

将节点i及其邻接节点j的特征向量作线性变换后，作用一个attention机制（由单层前馈网络实现），然后softmax正则化获得attention系数，该系数作为权值，计算获得节点i的新特征向量表示。更多细节的理解可以参考《Graph Attention Networks》阅读笔记

输入输出

Input: $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, ..., \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$

Output: $\mathbf{h} = \{\vec{h}'_1, \vec{h}'_2, ..., \vec{h}'_N\}, \vec{h}'_i \in \mathbb{R}^{F'}$

具体实现

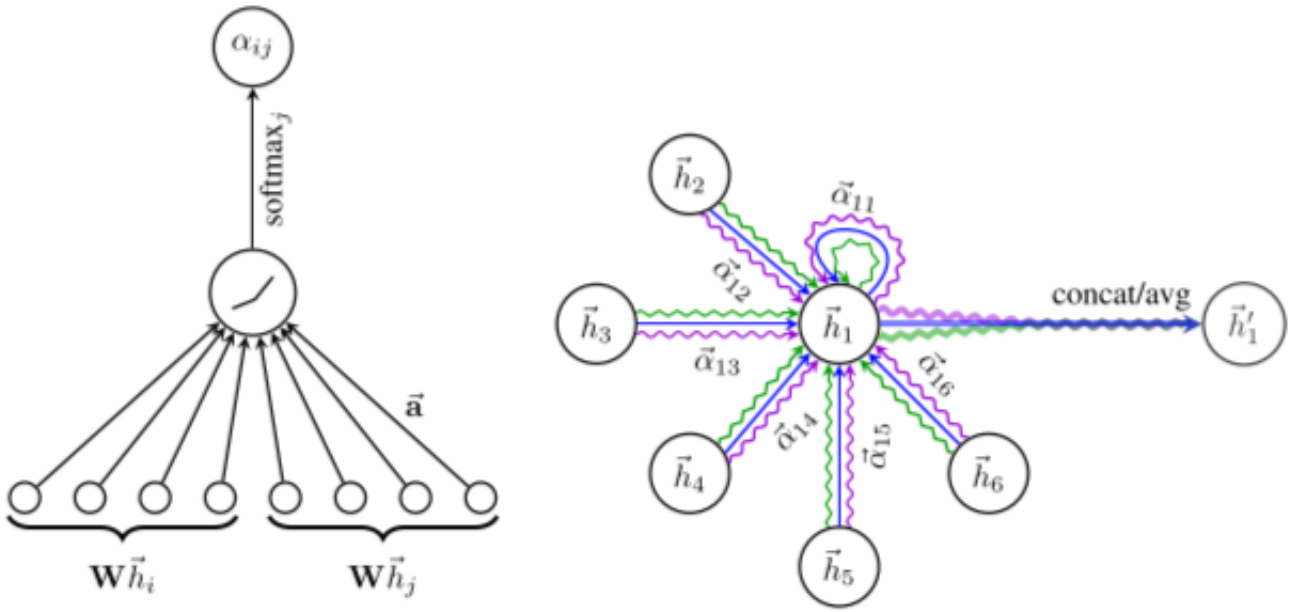


Figure 1: **Left:** The attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ employed by our model, parametrized by a weight vector $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$, applying a LeakyReLU activation. **Right:** An illustration of multi-head attention (with $K = 3$ heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain $\vec{h}'_1$.

假定图G有N个节点，输入和输出的节点特征数分别为F和F'

1. 分别对节点i和节点j的特征向量作线性变换，乘权值矩阵$\mathbf{W} \in \mathbb{R}^{F' \times F}$
2. 为节点分配attention（权重），其实是一个$\mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$的映射，具体来说实现方式是单层前馈神经网络$LeakyReLU(\vec{a}^T[\mathbf{W}\vec{h}_i||\mathbf{W}\vec{h}_j])$。最终获得attention系数

$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j) = LeakyReLU(\vec{a}^T[\mathbf{W}\vec{h}_i||\mathbf{W}\vec{h}_j])$$

3. 对e_{ij}进行softmax正则化，采用masked机制，仅考虑部分邻接节点$j \in N_i$

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})}$$

4. 节点i最终的输出特征$\vec{h}_i'$就是对N_i中所有节点的加权求和

$$\vec{h}_i' = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \mathbf{W} \vec{h}_j\right)$$

5. 也可以使用multi-head attention来稳定self-attention的学习过程，即同时使用多个W^k计算self-attention，然后将多个W^k计算得到的结果合并（连接或者求和），公式为

$$\vec{h}_i' = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right) \quad \vec{h}_i' = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j\right)$$

模型比较

- GAT是高效的，无特征值分解等复杂矩阵运算，单层GAT时间复杂度为$O(|V|FF' + |E|F')$，可以认为是顶点数和边数的线性数量级
- 相比GCN，节点的重要性可以是不同的，因此GAT具有更强的表示能力
- attention机制以共享的方式应用于图中的所有边（共享W），因此它不依赖于对全局图结构或者所有点的预先访问

Evaluation

转导学习：对于Cora, iteseer, Pubmed数据集，分类准确度相比其它模型有提升

归纳学习：对于PPI数据集，micro F1值相比其它模型有巨大提升

论文中有提出几个可以优化的点：

- 根据图结构的规则性，在这些稀疏场景中，GPU没有体现计算优势
- 感受野受模型深度限制，可用 skip connection 这样的技术来拓展深度
- 因为邻接节点高度重叠，并行计算有冗余
- GAT模型的 attention 系数的解释不明，需更多相关数据集的领域知识

在我看来，如果要在attention机制这个方向上更深入地研究GNN，可以在弄清楚attention机制的原理后，给出一种新的方法去构建论文中的映射a（attention机制），可以参考如下表格（包含各种attention机制），更多内容具体见Attention? Attention!

Name	Alignment score function	Citation
Additive(*)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$	Bahdanau2015
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$ Note: This simplifies the softmax alignment max to only depend on the target position.	Luong2015
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \mathbf{W}_a \boldsymbol{h}_i$ where $\mathbf{W}_a$ is a trainable weight matrix in the attention layer.	Luong2015
Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \boldsymbol{h}_i$	Luong2015
Scaled Dot-Product(^)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{n}}$ Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani2017
Self-Attention(&)	Relating different positions of the same input sequence. Theoretically the self-attention can adopt any score functions above, but just replace the target sequence with the same input sequence.	Cheng2016
Global/Soft	Attending to the entire input state space.	Xu2015
Local/Hard	Attending to the part of input state space; i.e. a patch of the input image.	Xu2015; Luong2015