Graph Attention Networks

- published in ICLR’18 (1 (https://arxiv.org/abs/1710.10903))

作者:

> Veličković, Petar
> Cucurull, Guillem
> Casanova, Arantxa
> Romero, Adriana
> Liò, Pietro
> Bengio, Yoshua

研究目的

- 解決傳統方式 (spectral-based method) 在做 graph convolution 的問題
  - 利用 self-attention layers
  - 利用 stacking layers 來取得 node 的 neighborhoods features，此外，也不需要 costly matrix operation 以及 graph structure
- Node classification of graph-structured data via attention mechanism

研究方式

- Previous literatures:
  - Recursive NN: (2 (https://pdfs.semanticscholar.org/3edf/d97cf8657e02d2c796db9aa412ceb077b0eb.pdf))
  - Graph NN (GNN): (3 (https://www.researchgate.net/profile/Franco_Scarselli/publication/4202380_A_new_model_for_earning_in_raph_domains/links/0c9605188cd580504f000000/A-new-model-for-earning-in-raph-domains.pdf)) (4 (https://persagen.com/files/misc/scarselli2009graph.pdf))
  - Spectral based: requires graph structure data
  - Graph Convolution Network (GCN): (5 (https://arxiv.org/abs/1509.09292)) (MoNET (https://arxiv.org/abs/1611.08402))(GraphSAGE (https://arxiv.org/abs/1706.02216)) works with different sized neighborhoods and maintains the weight sharing property of CNNs., sampling a fixed-size neighborhood of each node, then aggregate through them
  - Attention based

- 資料結構: 一連串的 node，且每個 node (h_i) 有 F 個 features，要輸出有另一組 F features 的 vector $\mathbf{h} = \{h_1, h_2, \ldots, h_N\}$
- **Traditional attention mechnism**Construct weight matrix, then perform self-attention (將注意力分配到圖中其他節點)

  $$e_{ij} = a(\mathbf{W}\vec{h_i}, \mathbf{W}\vec{h_j})$$

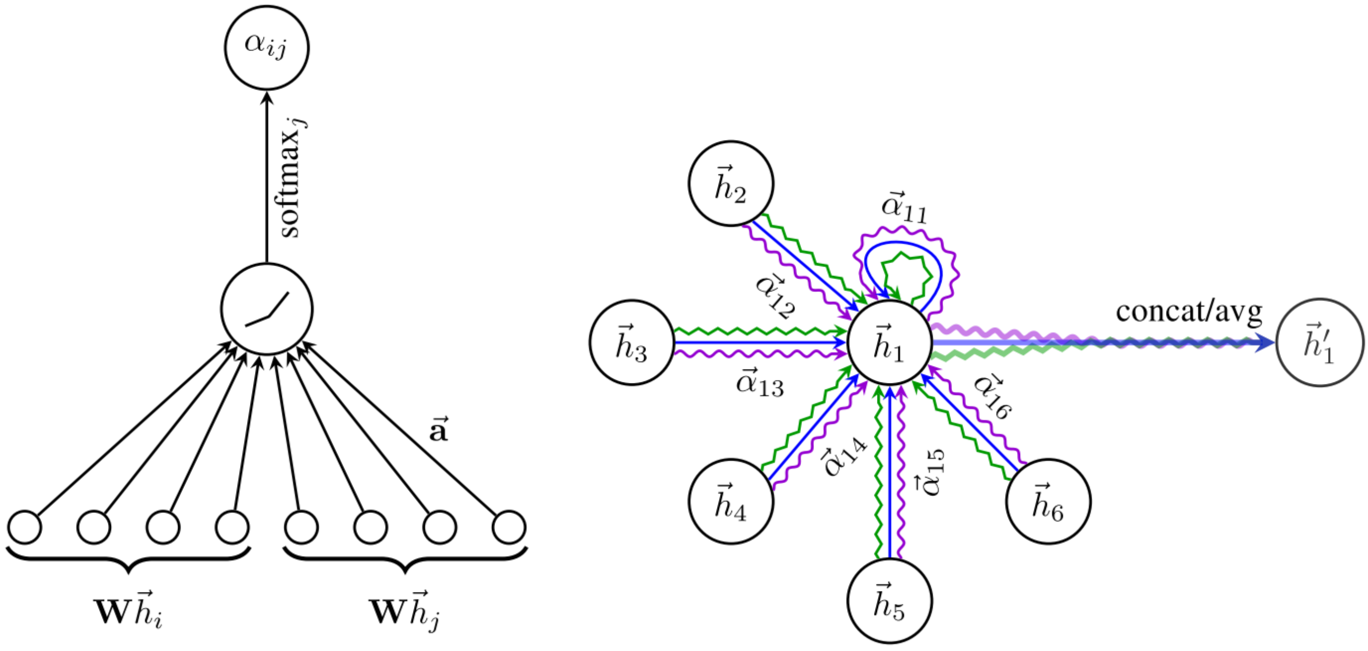
  where $\mathbf{W} \in \mathbb{R}^{F'xF}$ and a is a shared attentional mechanism $a = \mathbb{R}^{F'} * \mathbb{R}^{F'} \rightarrow \mathbb{R}$

  $$a_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}$$

  - e_{ij} 代表 node j 對 node i 的重要程度
  - N_i 是 node i 的 neighbors 而 $j \in N_i$
  - 本文中只用了 first-order neighbors of i (including i)
- **Attention mechanism in this paper, GAT**

  $$a_{ij} = \frac{\exp(\mathrm{LeakyReLU}(\vec{a}^T[\mathbf{W}\vec{h_i} || \mathbf{W}\vec{h_j}]))}{\sum_{k \in N_i} \exp(\mathrm{LeakyReLU}(\vec{a}^T[\mathbf{W}\vec{h_i} || \mathbf{W}\vec{h_k}]))}$$

  where LeakyReLU use a negative input slope of $\alpha = 0.2$



左圖是 attention mechnism 的示意圖，右圖為 multi-head attention 的示意圖

- Single-head attention: 最後根據 attention 的結果計算 linear combination of the featrues corresponding to them -> 每個 node 的 output features

  $$\vec{h_i'} = \sigma(\sum_{j \in N_i} \alpha_{ij}\mathbf{W}\vec{h_j})$$

- (Extended) Multi-head attention - concatenation

$$\overrightarrow{h'_i} = ||_{k=1}^{K}\sigma(\sum_{j\in N_i}\alpha_{ij}\mathbf{W}\overrightarrow{h_j})$$

- (Extended) Multi-head attention - averaging

$$\overrightarrow{h'_i} = \sigma(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}^{k}\mathbf{W}^{k}\overrightarrow{h_j})$$

- Time complexity of a single GAT: $O(|V|FF' + |E|F')$, where F is the number of input features, and $|V|$ and $|E|$ are the numbers of nodes and edges in the graph; multiply by K for K-head model
- 有另外實作針對 sparse matrix 的版本

評估方法

- Transductive learning: 訓練中只知 testing data (unlabelled data), 調整 training node 的數量，衡量不同情境下的準確度
- Inductive learning: 訓練中不知道 testing data, 訓練好模型後去解決未知的 testing data

研究貢獻

- Attention 機制是共享的，是一種局部模型