A modified attention mechanism for node classification

严汀沄
张驰

Node classification in citation networks

- To do
  - Find good node embeddings, then doing classification based on the learned embedding
- Before GNN
  - DeepWalk (Perozzi et al., 2014)
  - LINE (Tang et al., 2015)
  - Node2vec (Grover & Leskovec, 2016)
- GNN based
  - Spectral domain methods
  - Spatial domain methods

Related work

Design different filters to approximate $h(\Lambda) = diag([h(\lambda_0), \ldots, h(\lambda_{n-1})])$
For improving
- localization
- computation efficiency

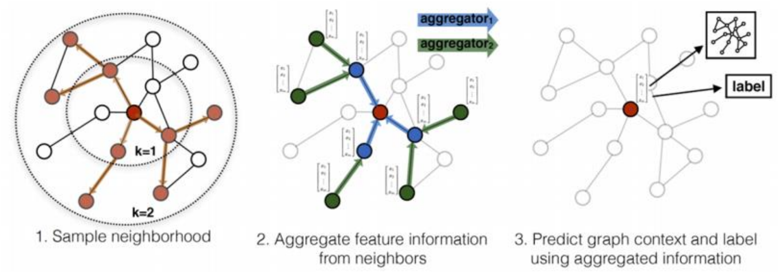
- Spectral domain methods

$$h(L)x = h(U\Lambda U^H)x = Uh(\Lambda)U^H x = Uh(\Lambda)\hat{x}$$

+ pros
  - Theoreticaly supported
+ cons
  - depend on graph structure to calculate Laplacian eigenbasis
+ representative works
  - Chebyshev filters [Hammond et al 2011]
  - Lanczaos filters [Liao et al 2019]
  - Cayley filters[Levie et al 2017]
  - ARMA filters[Bianchi et al 2019]
  - Feedback-Looped Filters[Asiri et al 2019]

Related work

- Spatial domain methods
  - Message passing framework(iterativly one-step gcn)



  - pros
    - effeciency
    - generality
    - flexibility
  - cons
    - deep layers may resulting converged node embedding, make it hard to be distinguished

How to better utilize information from (higher-order) neighbor node ?

Related work

- Spatial domain methods
  - representive works
    - Monet [Monti et al 2017]
    - MPNN [Gilmer 2017]
    - Graph SAGE [Hamilton et al 2017]
    - Graph attention networks [Velickovic et al 2018]
    - Jump knowledge networks [Xu et al 2018]
    - AdaGCN[Sun et al 2019]

Basic neiborhood aggregation

Neiborhood aggregation with directional biases

Feature Aggregation from different order neighbors

Graph Attention Network

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O$$

attention coefficients $e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$

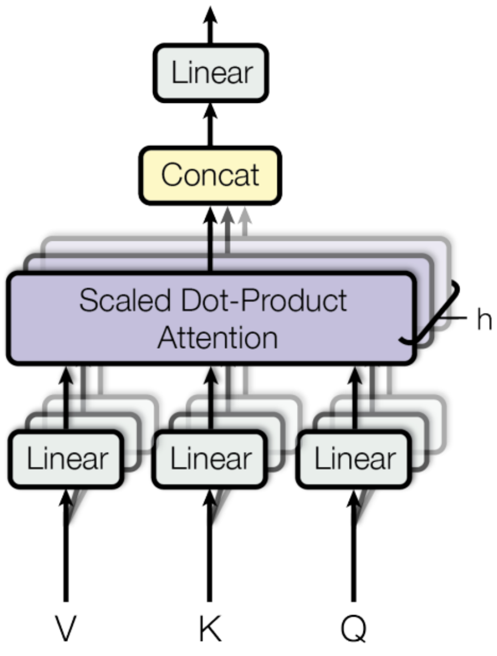
$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

softmax $\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

aggregate $\vec{h}'_i = \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}\vec{h}_j)$

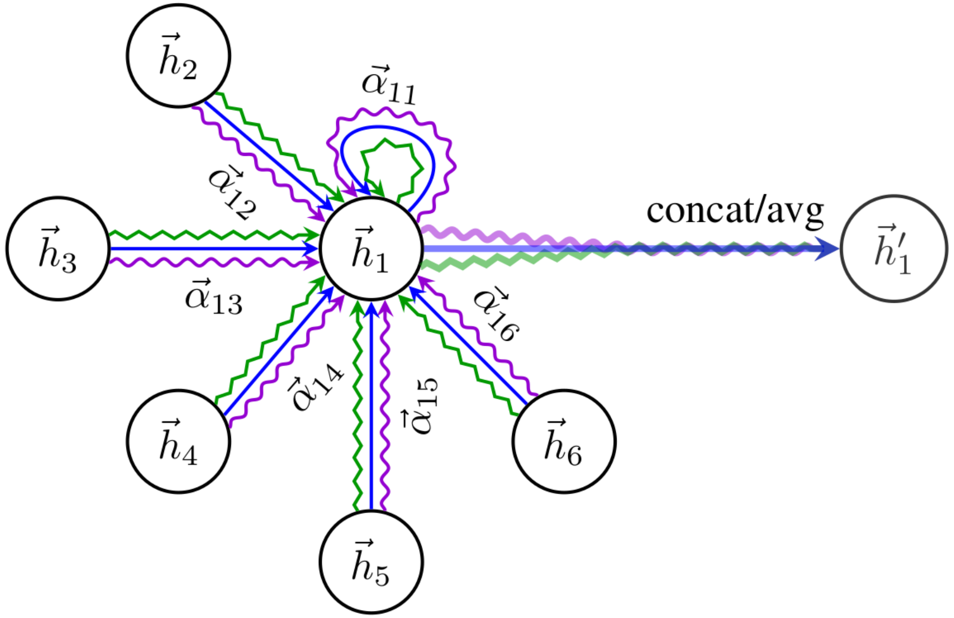
multi-head attention $\vec{h}'_i = \sigma(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j)$



Difference:

- self-attention—a shared attentional mechanism
- masked attention—only compute e_{ij} for nodes $j \in N_i$, where N_i is some neighborhood of node i in the graph.

Graph Attention Network



Method	**Cora**	**Citeseer**	**Pubmed**
MLP	55.1%	46.5%	71.4%
ManiReg (Belkin et al., 2006)	59.5%	60.1%	70.7%
SemiEmb (Weston et al., 2012)	59.0%	59.6%	71.7%
LP (Zhu et al., 2003)	68.0%	45.3%	63.0%
DeepWalk (Perozzi et al., 2014)	67.2%	43.2%	65.3%
ICA (Lu & Getoor, 2003)	75.1%	69.1%	73.9%
Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	**79.0%**
MoNet (Monti et al., 2016)	$81.7 \pm 0.5\%$	—	$78.8 \pm 0.3\%$
GCN-64*	$81.4 \pm 0.5\%$	$70.9 \pm 0.5\%$	$\mathbf{79.0} \pm 0.3\%$
GAT (ours)	$\mathbf{83.0} \pm 0.7\%$	$\mathbf{72.5} \pm 0.7\%$	$\mathbf{79.0} \pm 0.3\%$

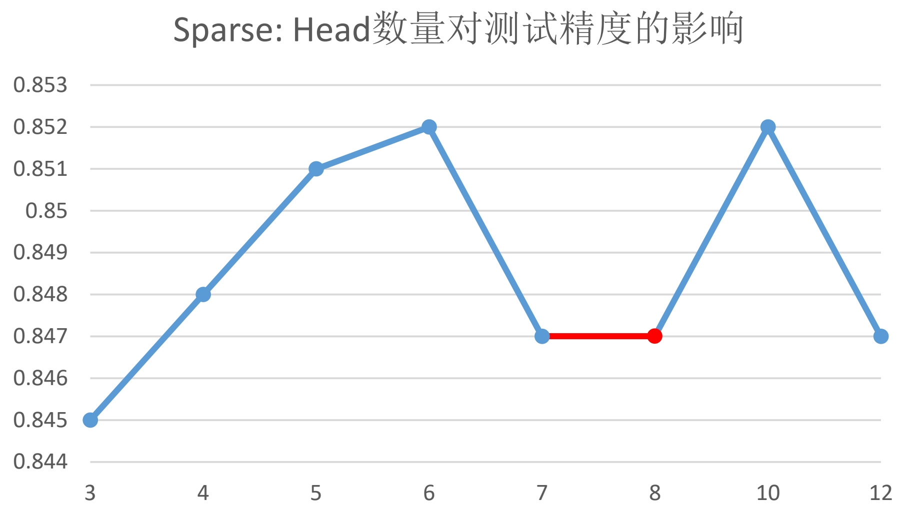
a two-layer GAT model

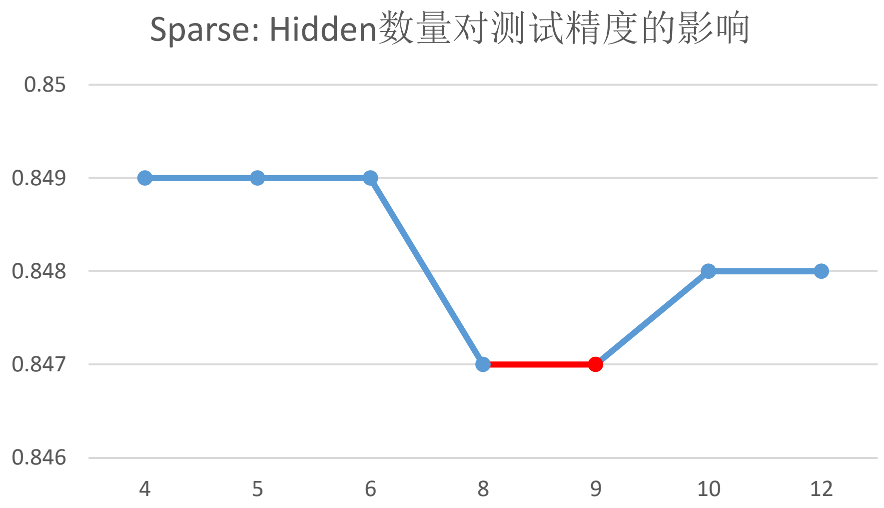
- The first layer consists of K = 8 attention heads computing F = 8 features each + ELU
- The second layer is used for classification: a single attention head computing F = 7 classes

Training = 140 Val =200-500 Test = 500-1500

问题：

1. 对GAT特征参数的选择进行对比实验：研究Head，Feature，激活函数等特征参数的在cora数据集上的影响





sparse 5head+6hidden	0.848
sparse 6head+6hidden	0.849
sparse 3head+3hidden	0.845
sparse 4head+12hidden	0.838
sparse relu	0.847

结论：

1. head的数量对测试精度的影响相对其它特征更高
2. head和hidden参数数量太大会造成过拟合，性能下降

问题：

2. 训练集为140，验证集200——500，测试集500——1500， baseline的数据集设置是否合理？

训练集

sparse Baseline	0.847(3次相同结果)
sparse 训练集(140-200)	0.850
sparse 训练集(140-500)	0.855

Baseline	0.850
Baseline 训练集(140——200)	0.861
Baseline 训练集(140——500)	0.866

结论：训练集的大小对GAT结果影响较大

测试集

sparse Baseline 训练集(140,200-500,500——1500)	0.847
sparse 训练集(140,200-500,1708——2707)	0.7648
sparse 训练集(140,200-500,500——2707)	0.8006

Baseline 训练集(140,200-500,500—1500)	0.850
Baseline 训练集(140,200-500,1708—2707)	0.7638

结论：Tranductive训练模式下，测试集的选择对GAT影响巨大，不同模型在baseline的测试集下效果好，不能说明在其它测试集上效果也好

问题：

2. 训练集为140，验证集200——500，测试集500——1500， baseline的数据集比例是否合理？

验证集

sparse 训练集(140,200-500,1708——2707)	0.7648
sparse 训练集(140,140-640,1708——2707)	0.7618
sparse 训练集(140,1000-1500,1708——2707)	0.7618

Baseline 训练集(140,200-500,1708—2707)	0.7638
Baseline 训练集(140,140-640,1708——2707)	0.7578
Baseline 训练集(140,1000-1500,1708——2707)	0.7578

结论：验证集的选择对GAT结果影响不大

全数据集

sparse Baseline 训练集(140,200-500,500——1500)	0.847
sparse 训练集(140,200-500,500——2707)	0.8006
sparse 训练集(1895,1895-2400,2400——2707)	0.8111

- baseline的数据集应该包括所有数据
- baseline的训练集，验证集和测试集的比例存在问题

问题：

3. 共享权重的线性变换是适合中心节点与邻居节点是相同类型的，对于中心点与邻居节点异构的情况，中心点和邻居节点需单独学习对应的线性变换。

$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

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

$$e_{ij} = a(W_Q\overrightarrow{h_i}, W_k\overrightarrow{h_j})$$

$$\overrightarrow{h'_i} = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W_V\overrightarrow{h_j}\right)$$

中心点的学习规律与邻居点的学习规律不同

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

模型\数据集	Cora	Citeseer
Sparse Baseline	0.85	0.708
Sparse QKV	0.85	0.708

Baseline的节点分类数据集中心点和邻居点规律相同

如何提高QKV的对实验的效果需进一步探索