On the Global Self-attention Mechanism for Graph Convolutional Networks

Chengyuan Deng
Rutgers University

Chen Wang
Rutgers University

ICPR 2020
Global Self-attention Mechanism

GSA in Convolutional Neural Networks

- [ZGMO'18] Self-attention GAN
- Image details generated from global feature locations

Image source: https://arxiv.org/abs/1805.08318
Global Self-attention Mechanism

GSA in Graph Convolutional Networks

• Vanilla self-attention: information between neighbors
• [WD'20] Global self-attention: regardless of edge connection
Graph Convolutional Networks

Graph Convolutions [KW '17]

- Message-passing GNN
- First order approximation of spectral convolutions

Layer Update

- \[ f(H^{(l+1)}, A) = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \]

  where \( \hat{A} \) is a normalized adjacency matrix, and \( \hat{D}_{ii} = \sum_j \hat{A}_{ij} \)
Motivation

Attention methods in GNNs

• Attention methods in GNNs: [VCCRLB '18] [ZXT '20][ZSXMKY '18]

• [PWCLY '18] Weakness: "lack of the ability to capture long-range dependencies"

• [ZGMO'18] Self-attention GAN remedies the weakness

• Main motivation:
  • Similarity between convolutions on images and graphs
  • Can Global self-attention improves the performance of GCNs?
Motivation

Theoretical Analysis on GCNs

- [RHXH '19] and [LMTD '19] Over-smoothing problem of GCNs:
  - Deep GCNs: hard to optimize training loss
- [LHW '19] Why over-smoothing?
  - A special form of Laplacian smoothing
  - Converge to a feature-invariant space as
- [OS '20] Convergence rate:
  - Exponential to the maximum singular value of the convolutional filter
Contributions: Two-Fold

Theoretical Analysis

• Prove: Global Self-attention can alleviate over-fitting and over-smoothing problems

GSA-GCN: A Novel Framework

• Experiments on two classical tasks: node classification and graph classification

• Empirical results corroborate our theoretical analysis
Theoretical Analysis

Main Results on Over-fitting

Assumptions: (1) d-regular graph            (2) -1/1 feature influence between vertices

• Key Lemma: There exists a way to arrange the GSA feature relations, such that for at least $\Omega\left(\frac{n}{4^r}\right)$ ($r < n$) vertices, the GSA mechanism will eliminate the influence of one vertex in the neighbors

• Proof sketch of the Key Lemma:
  - A natural corollary of Ramsey theorem
Theoretical Analysis

Main Results on Over-fitting

- Proof sketch of the Key Lemma:
  - A natural corollary of Ramsey theorem
  - \( R(r + 1, r + 1) \) vertices, there must exist \( (r + 1) \) vertices with either +1 / -1 influence
  - Normalized GSA influence: +1 / -1, pick \( \gamma = \frac{1}{d} \)
  - GSA influence: \( +\frac{1}{d} \) or \( -\frac{1}{d} \)
  - Eliminate the influence from one \( v_i \in N(v') \)

Key Lemma: There exists a way to arrange the GSA feature relations, such that for at least \( \Omega(\frac{n}{4^r}) \) \( (r < n) \) vertices, the GSA mechanism will eliminate the influence of one vertex in the neighbors.
Theoretical Analysis

Main Results on Over-smoothing

The convergence rate to the feature-invariant subspace of GSA-GCN is slower than that of plain GCNs

- Key lemma: Applying GSA is equivalent to substituting convolution weight matrices with a larger maximum singular value

- High-level proof of the key lemma:
  - GSA matrix: Hermitian and Positive Definite
  - [KT '01] Minimum eigenvalue larger than 1
  - 2-norm connects singular value and above results

\[ PW = (I + \gamma \cdot Q)W \]

\[ \lambda_{\min}(P) \geq \lambda_{\min}(I) + \lambda_{\min}(Q) > 1 \]

\[ \tilde{s}^l = ||\tilde{W}^l|| = ||(I + \gamma \cdot Q)W^l|| \geq \lambda_{\min}(P)||W^l|| > ||W^l|| = s^l \]
GSA-GCN: Framework

Layer Update of GSA-GCN:

\[ H^{(l+1)} = \sigma((\tilde{A}H^{(l)} + \gamma O^{(l)})W^{(l)}) \]

where \( O^{(l)} \) is the output of the GSA layer, \( \gamma \) is a non-negative trainable parameter and \( W^{(l)} \) is the convolution matrix of layer \( l \).
GSA-GCN: Experiments

Node Classification: Inductive

- We strictly follow the experimental setup in the literature [KW'17]
- Two-layer GCN backbone is considered as baseline

<table>
<thead>
<tr>
<th>Model</th>
<th>Cora</th>
<th>Citeseer</th>
<th>Pubmed</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>81.5</td>
<td>70.3</td>
<td>79.0</td>
</tr>
<tr>
<td>GAT</td>
<td>83.0</td>
<td>72.5</td>
<td>79.0</td>
</tr>
<tr>
<td>JK-Net (4)</td>
<td>80.2</td>
<td>68.7</td>
<td>78.0</td>
</tr>
<tr>
<td>DropEdge-GCN</td>
<td>82.8</td>
<td>72.3</td>
<td>79.6</td>
</tr>
<tr>
<td>GSA-GCN</td>
<td>83.3</td>
<td>72.9</td>
<td>80.1</td>
</tr>
</tbody>
</table>
GSA-GCN: Experiments

Node Classification: **Transductive**

<table>
<thead>
<tr>
<th>Model</th>
<th>Cora</th>
<th>Citeseer</th>
<th>Pubmed</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>86.1</td>
<td>75.9</td>
<td>90.2</td>
</tr>
<tr>
<td>GAT</td>
<td>86.4</td>
<td>76.6</td>
<td>OOM</td>
</tr>
<tr>
<td>JK-Net</td>
<td>86.9</td>
<td>78.3</td>
<td>90.5</td>
</tr>
<tr>
<td>DropEdge-GCN</td>
<td>86.5</td>
<td>78.7</td>
<td>91.2</td>
</tr>
<tr>
<td><strong>GSA-GCN</strong></td>
<td><strong>88.2</strong></td>
<td><strong>79.1</strong></td>
<td><strong>89.4</strong></td>
</tr>
</tbody>
</table>

- GSA-GCN performs competitively to the state-of-the-art
GSA-GCN: Experiments

Alleviate Over-smoothing

- As the layers go deeper, the gap between GSA-GCN and plain GCN becomes more significant
Conclusions

• Global Self-attention:
  ‣ Propagate global node features with soft-max
  ‣ Remedy the over-fitting and over-smoothing problem of GCNs
  ‣ Competitive experimental results corroborate

Open Questions

• Remove the assumptions

• Geometry-independent self-attention
  ‣ Explicitly find the fraction of vertices that have impacts
  ‣ (Potentially) Substitute GSA with an efficient algorithm
Main References

[WD’20] On the Global Self-attention Mechanism for GraphConvolutional Networks
[KW’17] Semi-supervised classification with graph convolutional networks
[VCCRLB ’18] Graph Attention Networks
[ZXT ’20] Context aware graph convolution for skeleton-based action recognition
[ZSXMKY ’18] Gaan: Gated attention networks for learning on large and spatiotemporal graphs
[ZGMO’18] Self-attention Generative Adversarial Networks
[PWCLY ’18] Geom-gcn: Geometric graph convolutional networks
[RHXH ’19] Dropedge: Towards deep graph convolutional networks on node classification
[LMTD ’19] Deepgcns: Can GCNs go as deep as CNNs?
[LHW ’19] Deeper insights into graph convolutional networks for semisupervised learning
[OS ’20] Graph neural networks exponentially lose expressive power for node classification
[KT ’01] Honeycombs and Sums of Hermitian Matrices