Multi-Hop Attention using Diffusion MAGNA

September 8, 2021

Dimitrios K. Kelesis
Attention in GNNs

Multi-Hop Attention

Introduction

MAGNA’s equations in the background

Injecting Diffusion for Multi-Hop Neighbors

MAGNA Architecture

Benefits of MAGNA

Results

Model Analysis
Attention has proven its power in many tasks:

- NLP
- Image Classification
- Object Recognition

Recently proposed method to incorporate attention in Graph Neural Networks has proved successful and yielded interesting results.
GAT was the first attempted method to inject attention in GNNs where:

- Messages from other nodes do **not** have the same importance.
- Central node gives attention weights to other nodes.
- Attention is given only to the neighboring nodes.

\[
a_{i,j} = \text{softmax}(e_{i,j}),
\]

where

\[
e_{i,j} = \alpha(W\vec{h}_i, W\vec{h}_j) = \text{LeakyReLU}(\tilde{\alpha}^T[W\vec{h}_i || W\vec{h}_j])
\]
Receptive field

- GNNs like CNNs have a receptive field.
- Receptive field is considered as the area where the system attends in order to learn.
- The larger the receptive field the more information could be aggregated.
Single VS Multi Hop

a) Single-hop star topology

b) Multi-hop linear topology
Why Multi-Hop?

- Self-attention led to state-of-the-art results.
- All methods find attention only between neighboring nodes.
- Attention depends only on the representation of the nodes.
- Multi-Hop attention allows to account for nodes not directly connected.
- Multi-hop context information aggregation and long-range interactions.
Pros of using MAGNA

- Diffuses attention scores increasing the receptive field.
- Efficiently account for all paths between nodes.
- Acts as a low-pass filter on the spectral domain of the graph.
- Directly connected to Personalized Page Rank.
Why not only use GAT?

- Using GAT nodes only attend their direct neighbors.
- That means the receptive field of each layer is restricted.
- Multi-layer architecture results in the over-smoothing problem.
- Attention depends solely on nodes and limits the ability to explore further graph structure (e.g. more distant nodes)
MAGNA Explanation

- Graph attention diffusion layer to capture distant information.
- Computes attention weights over edges.
- Self-attention weights between disconnected pairs using diffusion.

Results in:
- Capturing long-range interactions between nodes.
- Computing content-dependent attention.
- Attention scheme depends on the paths of given length.
**Figure**: Left: GAT computes attention score between neighboring pairs (i.e. $a_{D,C} = 0$).

Right: MAGNA computes attention of D to C using the path between them.
In layer l a vector message is computed for each triple \((u_i, r_k, u_j)\).

The representation of \(u_j\) at layer \(l + 1\) results from all messages of triples incident to \(u_j\).

Attention score \(s\) of each edge \((u_i, r_k, u_j)\) is given as:

\[
s^l_{i,k,j} = \text{LeakyReLU}(u^l_{\alpha}\tanh(W^l_h h^l_i||W^l_t h^l_j||W^l_r r_k))
\]
Repeating for every edge results in an attention score matrix:

\[ S^l = \begin{cases} 
  s^l_{i,k,j}, & (i,j) \in E \\
  -\infty, & \text{otherwise} 
\end{cases} \]

The attention matrix \( A^l \) results from row-wised softmax as:
\[ A^l = \text{softmax}(S^l). \]

\( A^l_{i,j} \) denotes the attention value at layer \( l \) by aggregating messages between \( i \) and \( j \).
Attention diffusion procedure computes attention scores of distant neighbors based on powers of the attention matrix $A$.

$$A' = \sum_{i=0}^{\infty} \theta_i A^i \quad \text{where} \quad \sum_{i=0}^{\infty} \theta_i = 1 \quad \text{and} \quad \theta_i > 0 \quad (1)$$

Where $\theta_i$ is a decaying over $i$.

Powers of $A$ encode paths (of the superscript length) increasing the receptive field.

Attention depends on the previous layer representations and uses the paths between nodes resulting in attention shortcuts.
MAGNA Architecture

- MAGNA uses multi-head attention in a similar manner as original GAT does.

- It uses deep aggregation so that it can generalize the original GAT.

- Deep Aggregation (i.e. two-layer MLP) approximates many functions than the shallow elu activation function in GAT.
Spectral properties

- Authors claim that MAGNA acts as a low pass filter in the spectral domain.

- It enhances the smaller eigenvalues while suppresses the larger ones (considered to correspond to noise).

- That means MAGNA is able to better capture large-scale structures.
PageRank

- PageRank is an algorithm to rank nodes in a graph.
- PageRank counts the number and quality of links to a node to calculate its importance.
- More important nodes are likely to receive more links from other nodes.
Personalized PageRank

- Each node wants to score other nodes differently.
- Personalized PageRank is the probability of a random walk of length \( l \) starting from \( s \) to terminate on \( t \).
- It indicates the bidirectional importance between \( s \) and \( t \).
- The length could be drawn from a distribution (e.g., Geomatrical).
Attention matrix $A$ could be considered as a random walk matrix on the graph.

If we perform PPR with parameter $\alpha$ on $G$ with transition matrix $A$ then the fully PPR is defined as:

$$A_{PPR} = \alpha(I - (1 - \alpha)A)^{-1}$$

Using power series expansion this results in (1) with $\theta_i = \alpha(1 - \alpha)^i$

Attention diffusion defines a PPR with parameter $\alpha$ on $G$ where $\alpha$ stands for the teleport probability of PPR.
### Node Classification Results

<table>
<thead>
<tr>
<th>Models</th>
<th>Cora</th>
<th>Citeseer</th>
<th>Pubmed</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN (Kipf &amp; Welling, 2016)</td>
<td>81.5</td>
<td>70.3</td>
<td>79.0</td>
</tr>
<tr>
<td>Chebyshev (Defferrard et al., 2016)</td>
<td>81.2</td>
<td>69.8</td>
<td>74.4</td>
</tr>
<tr>
<td>DualGCN (Zhuang &amp; Ma, 2018)</td>
<td>83.5</td>
<td>72.6</td>
<td>80.0</td>
</tr>
<tr>
<td>JKNNet (Xu et al., 2018)*</td>
<td>81.1</td>
<td>69.8</td>
<td>78.1</td>
</tr>
<tr>
<td>LGCN (Gao et al., 2018)</td>
<td>83.3 ± 0.5</td>
<td>73.0 ± 0.6</td>
<td>79.5 ± 0.2</td>
</tr>
<tr>
<td>Diffusion-GCN (Klicpera et al., 2019b)</td>
<td>83.6 ± 0.2</td>
<td>73.4 ± 0.3</td>
<td>79.6 ± 0.4</td>
</tr>
<tr>
<td>APPNP (Klicpera et al., 2019a)</td>
<td>84.3 ± 0.2</td>
<td>71.1 ± 0.4</td>
<td>79.7 ± 0.3</td>
</tr>
<tr>
<td>g-U-Nets (Gao &amp; Ji, 2019)</td>
<td>84.4 ± 0.6</td>
<td>73.2 ± 0.5</td>
<td>79.6 ± 0.2</td>
</tr>
<tr>
<td>GAT (Veličković et al., 2018)</td>
<td>83.0 ± 0.7</td>
<td>72.5 ± 0.7</td>
<td>79.0 ± 0.3</td>
</tr>
<tr>
<td>No LayerNorm</td>
<td>83.8 ± 0.6</td>
<td>71.1 ± 0.5</td>
<td>79.8 ± 0.2</td>
</tr>
<tr>
<td>No Diffusion</td>
<td>83.0 ± 0.4</td>
<td>71.6 ± 0.4</td>
<td>79.3 ± 0.3</td>
</tr>
<tr>
<td>No Feed-Forward°</td>
<td>84.9 ± 0.4</td>
<td>72.2 ± 0.3</td>
<td>80.9 ± 0.3</td>
</tr>
<tr>
<td>No (LayerNorm + Feed-Forward)</td>
<td>84.3 ± 0.6</td>
<td>72.6 ± 0.4</td>
<td>79.6 ± 0.4</td>
</tr>
<tr>
<td><strong>MAGNA</strong></td>
<td><strong>85.4 ± 0.6</strong></td>
<td><strong>73.7 ± 0.5</strong></td>
<td><strong>81.4 ± 0.2</strong></td>
</tr>
</tbody>
</table>

* : based on the implementation in https://github.com/DropEdge/DropEdge;
° : replace the feed forward layer with *elu* used in GAT.
Model Analysis

- Extends the receptive field while avoiding over-smoothing.
- Increasing the number of layers do not decrease accuracy.
- Outperforms GAT with similar theoretical receptive field.
- Increasing parameter of hops improves accuracy.
- Parameter $\alpha$ is directly connected to the low pass effect thus affecting accuracy.
- More meaningful attention as calculated using attention distribution.
Discussion

- Is this setup equal to stacking?
- Can this setup solve the problem of over-smoothing in a few layers?
- Can we even use single layer (consider the 6-step-separation theorem)?
- Is it possible through this attention diffusion mechanism to effectively attend all import nodes despite how far they are?
Thank you