How Attentive are Graph Attention Networks?

not that much...

Shaked Brody
Technion

Uri Alon
Language Technologies Institute
Carnegie Mellon University

Eran Yahav
Technion
But what kind of attention?
Attention

The ability of different queries to learn to “focus” differently on a set of keys

[Bahdanau et al., ICLR 2015]
Graph Attention Networks (GAT) [Veličković et al., 2018]
GAT uses an Addition of Two Dot Products

\[
e(h_i, h_j) = \text{LeakyReLU}(a_1^T [W h_i] \| W h_j)
\]
GAT Attends to the Same Key Regardless of Query

\[ e(h_i, h_j) = \text{LeakyReLU}(a_1^\top \cdot W h_i + [a_2^\top \cdot W h_j] \cdot S_j) \]

\[ S_8 \geq S_6 \geq S_2 \geq \ldots \]
GATv2: Fixing Graph Attention Mechanism

GAT, Veličković et al., 2018:

GATv2, this work:

\[ e(h_i, h_j) = \text{LeakyReLU}(a^T [W h_i \parallel W h_j]) \]

\[ e(h_i, h_j) = a^T \text{LeakyReLU}(W \cdot [h_i \parallel h_j]) \]
Static Attention

For any sets of:

Queries

Keys

There is always a key that gets the most attention, regardless of the query
Dynamic Attention

For any set of queries, keys, and any desired mapping between them:

There exist learned parameters that “implement” this mapping
Experimental Results

- **GATv2** always outperformed **GAT** in 12 benchmarks of node- link- and graph-prediction
- **GATv2** is more robust to noisy edges (which did not exist in the original graph)
Summary

- Define **static attention** vs. **dynamic attention**

- **GAT** computes **static attention**

- **GATv2**: a simple modification that is strictly more expressive than **GAT**
  - More accurate across 12 benchmarks and more robust to noise

- Use **GATv2** instead of **GAT** whenever possible

- **GATv2** is available on:
  - PyTorch Geometric: from torch_geometric.nn import GATv2Conv
  - DGL: from dgl.nn.pytorch import GATv2Conv
  - TensorFlow GNN: from tensorflow_gnn.keras.layers import GATv2

[shakedbr@cs.technion.ac.il](mailto:shakedbr@cs.technion.ac.il)