Graph Neural Networks – basics architectures

Giorgio Piatti
SiDNN – 08.03.2022
Motivation
Motivation

• Atoms as nodes

• Bonds as edges

• Features:
  • Atom type
  • Charge
  • Bond type
Motivation

A Deep Learning Approach to Antibiotic Discovery

Authors
Jonathan M. Stokes, Kevin Yang,
Kyle Swanson, ..., Tommi S. Jaakkola,
Regina Barzilay, James J. Collins

Correspondence
regina@csail.mit.edu (R.B.),
jimc@mit.edu (J.J.C.)

In Brief
A trained deep neural network predicts
antibiotic activity in molecules that are
structurally different from known
antibiotics, among which Halcin exhibits
efficacy against broad-spectrum
bacterial infections in mice.

https://doi.org/10.1016/j.cell.2020.01.021
Learning problems

Node classification

Relation prediction
Learning problems

Graph classification
A GNN is an optimizable transformation on all attributes of the graph (nodes, edges)

Loss: cross-entropy
A GNN is an optimizable transformation on all attributes of the graph (nodes, edges)

Loss: cross-entropy
GNN architecture – link prediction

A GNN is an optimizable transformation on all attributes of the graph (nodes, edges)

Loss form: $\mathcal{L}(h_u, h_v, e_{uv})$
A GNN is an optimizable transformation on all attributes of the graph (nodes, edges)

Loss form: $\mathcal{L}(h_u, h_v, e_{uv})$
GNN layers

Input -> Graph network layer -> Node embeddings

layer 1 -> layer 2 -> ... -> layer K
**GNN layers**

**Input**

Graph $G=(V,E)$
Feature vector $x_u \in \mathbb{R}^d$ for all $u \in V$

**Node embeddings**

$h_u = f(x_u)$

Vector embedding $h_u \in \mathbb{R}^d$
Key ideas

Special class of graphs: image grids
Key ideas

Images

Convolution on grid

Graphs

Convolution on adjacent matrix?

Sensitive to node reordering & graph size
Key ideas

\[ h'_u = g(h_u, h_a, h_b, h_c, h_d ; \theta) \]
Key ideas

Desirable properties of $g$:  
  • Fixed number of parameters  
  • Leverage local information  
  • Permutation invariant
Message Passing Neural Network (MPNN)
Message Passing Neural Network (MPNN)

\[ h_u^{(k)} = \text{embedding of } u \text{ at } k^{th} - \text{iteration} \]

\[ m_{N(u)}^{(k)} = AGGREGATE^{(k)} \left( \{ h_v^{(k)} \}, \forall v \in N(u) \right) \]
Message Passing Neural Network (MPNN)

\[ h_u^{(k)} = \text{embedding of } u \text{ at } k^{\text{th}} \text{ iteration} \]

\[ m_{N(u)}^{(k)} = \text{AGGREGATE}^{(k)} \left( \{ h_v^{(k)}, \forall v \in N(u) \} \right) \]

\[ h_u^{(k+1)} \leftarrow \text{UPDATE}^{(k)} \left( h_u^{(k)}, m_{N(u)}^{(k)} \right) \]
Basic GNN

\[ h_u \leftarrow \sigma(W_{self} h_u + W_{neigh}(h_a + h_b + h_c + h_d)) \]

\[ m_{N(u)} = AGGREGATE(\{h_v \mid \forall \ v \in N(u)\}) = \sum_{v \in N(u)} h_v \]

\[ h_u \leftarrow UPDATE(h_u, m_{N(u)}) = \sigma(W_{self} h_u + W_{neigh} m_{N(u)}) \]

\[ \theta = \{W_{self}, W_{neigh}\} \]
Neighborhood normalization

\[ h_b \in N(b) \gg h_v \in N(a) \]

Assume

\[ \left| \sum_{v \in N(b)} h_v \right| \gg \left| \sum_{v \in N(a)} h_v \right| \]

Numerical instabilities & difficulties for optimization
Neighborhood normalization

\[ m_{N(u)} = \frac{1}{|N(u)|} \sum_{v \in N(u)} h_v \]

Downsides:
loss of information about structural info
Graph Convolution Network (GCN)

$h_u \leftarrow \text{UPDATE}(h_u, m_{N(u)}) = \sigma(Wm_{N(u)})$

$m_{N(u)} = \text{AGGREGATE}\left(\{h_v, \forall v \in N(u) \cup \{u\}\}\right)$

$= \sum_{v \in N(u) \cup \{u\}} \frac{h_v}{\sqrt{|N(u)| \cdot |N(v)|}}$
MPNN – limitations

Sum aggregate can distinguish, mean cannot
How to learn any set function?

$$g(X) = \varphi \left( \sum_{x \in X} f(x) \right), \quad s.t. \sum_{x \in X} f(x) \text{ unique}$$

$$h_u \leftarrow \text{UPDATE}(h_u, \text{AGGREGATE}([h_v, \forall v \in N(u)]))$$
Graph Isomorphism Network (GIN)

\[ h_u^{(k)} = MLP^{(k)} \left( (1 + \epsilon)h_u^{(k-1)} + \sum_{v \in N(u)} h_v^{(k-1)} \right) \]

MLP=multi-layer perceptron NN

[K. Xu et al, 2019]
Graph Attention Network (GAT)

How relevant a neighboring node is in relation to the center node?

\[ m_{N(u)} = AGGREGATE(\{h_v, \forall v \in N(u)\}) = \sum_{v \in N(u)} \alpha_{u,v} h_v \]

Bilinear attention model

\[ \alpha_{u,v} = \frac{\exp(a^T [W h_u \oplus W h_v])}{\sum_{v \in N(u)} \exp(a^T [W h_u \oplus W h_v])} \]

Attention via MLP

\[ \alpha_{u,v} = \frac{\exp(MLP(h_u, h_v; \theta))}{\sum_{v \in N(u)} \exp(MLP(h_u, h_v; \theta))} \]

[P. Veličković, 2018]
Graph Attention Network (GAT)

How relevant a neighboring node is in relation to the center node?

Multiple attention “heads”

\[ m_{N(u)} = AGGREGATE(\{h_v, \forall v \in N(u)\}) = [a_1 \oplus a_2 \oplus \cdots \oplus a_K] \]

\[ a_k = W_i \sum_{v \in N(u)} \alpha_{u,v,k} h_v \]
GAT - results

Transductive setting
1 graph, some labels are masked

Inductive setting
Multiple graphs, some for training, some for test
GAT - results

<table>
<thead>
<tr>
<th>Task</th>
<th>Transductive</th>
<th>Transductive</th>
<th>Transductive</th>
<th>Inductive</th>
</tr>
</thead>
<tbody>
<tr>
<td># Nodes</td>
<td>2708 (1 graph)</td>
<td>3327 (1 graph)</td>
<td>19717 (1 graph)</td>
<td>56944 (24 graphs)</td>
</tr>
<tr>
<td># Edges</td>
<td>5429</td>
<td>4732</td>
<td>44338</td>
<td>818716</td>
</tr>
<tr>
<td># Features/Node</td>
<td>1433</td>
<td>3703</td>
<td>500</td>
<td>50</td>
</tr>
<tr>
<td># Classes</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>121 (multilabel)</td>
</tr>
<tr>
<td># Training Nodes</td>
<td>140</td>
<td>120</td>
<td>60</td>
<td>44906 (20 graphs)</td>
</tr>
<tr>
<td># Validation Nodes</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>6514 (2 graphs)</td>
</tr>
<tr>
<td># Test Nodes</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>5524 (2 graphs)</td>
</tr>
</tbody>
</table>

### Transductive

<table>
<thead>
<tr>
<th>Method</th>
<th>Cora</th>
<th>Citeseer</th>
<th>Pubmed</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>55.1%</td>
<td>46.5%</td>
<td>71.4%</td>
</tr>
<tr>
<td>ManiReg (Belkin et al., 2006)</td>
<td>59.5%</td>
<td>60.1%</td>
<td>70.7%</td>
</tr>
<tr>
<td>SemiEmb (Weston et al., 2012)</td>
<td>59.0%</td>
<td>59.6%</td>
<td>71.7%</td>
</tr>
<tr>
<td>LP (Zhu et al., 2003)</td>
<td>68.0%</td>
<td>45.3%</td>
<td>63.0%</td>
</tr>
<tr>
<td>DeepWalk (Perozzi et al., 2014)</td>
<td>67.2%</td>
<td>43.2%</td>
<td>65.3%</td>
</tr>
<tr>
<td>ICA (Lu &amp; Getoor, 2003)</td>
<td>75.1%</td>
<td>69.1%</td>
<td>73.9%</td>
</tr>
<tr>
<td>Planctoid (Yang et al., 2016)</td>
<td>75.7%</td>
<td>64.7%</td>
<td>77.2%</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>GCN (Kipf &amp; Welling, 2017)</td>
<td>81.5%</td>
<td>70.3%</td>
<td><strong>79.0%</strong></td>
</tr>
<tr>
<td>MoNet (Monti et al., 2016)</td>
<td>81.7 ± 0.5%</td>
<td>—</td>
<td>78.8 ± 0.3%</td>
</tr>
<tr>
<td>GCN-64*</td>
<td>81.4 ± 0.5%</td>
<td>70.9 ± 0.5%</td>
<td><strong>79.0 ± 0.3%</strong></td>
</tr>
<tr>
<td>GAT (ours)</td>
<td><strong>83.0 ± 0.7%</strong></td>
<td>72.8 ± 0.7%</td>
<td><strong>79.0 ± 0.3%</strong></td>
</tr>
</tbody>
</table>

### Inductive

<table>
<thead>
<tr>
<th>Method</th>
<th>PPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.396</td>
</tr>
<tr>
<td>MLP</td>
<td>0.422</td>
</tr>
<tr>
<td>GraphSAGE-GCN (Hamilton et al., 2017)</td>
<td>0.500</td>
</tr>
<tr>
<td>GraphSAGE-mean (Hamilton et al., 2017)</td>
<td>0.598</td>
</tr>
<tr>
<td>GraphSAGE-LSTM (Hamilton et al., 2017)</td>
<td>0.612</td>
</tr>
<tr>
<td>GraphSAGE-pool (Hamilton et al., 2017)</td>
<td>0.600</td>
</tr>
<tr>
<td>GraphSAGE*</td>
<td>0.768</td>
</tr>
<tr>
<td>Const-GAT (ours)</td>
<td>0.934 ± 0.006</td>
</tr>
<tr>
<td>GAT (ours)</td>
<td><strong>0.973 ± 0.002</strong></td>
</tr>
</tbody>
</table>
Graph pooling

Input

Node embeddings

Graph classification

Graph Network

Pooling

NN
Graph pooling

Set pooling approaches

• \( z_G = \frac{\sum_{v \in V} z_v}{|V|} \)

• \( z_G = \sum_{v \in V} z_v \)

Graph coarsening approaches
Graph coarsening

\[ S_{ui} = \text{prob. of node } u \text{ to belong to cluster } i \]

\[ G: A, X \]

\[ A' = S^T A S \in \mathbb{R}^{c \times c} \]

\[ X' = S^T X \in \mathbb{R}^{c \times d} \]
Graph coarsening

\[ X' = S^T \cdot T \cdot X = \text{cluster } i \text{ embedding} \]
Graph pooling

Input

Graph Network

Node embeddings

Pooling

Graph classification

NN
Outlook: problems
Problems – Graph isomorphism (WL-test)
Problems – Underreaching
Problems – Underreaching

Issue

Possible solution
• Aggregate further away

• Add edges
  • Changes topology
  • May lead to oversmoothing
Problems – Oversmoothing

**Issue**

Averaging same embedding for all nodes

**Possible solution**

- Skip connections

\[
\begin{align*}
m_{N(u)}^{(k)} &= AGGREGATE^{(k)} \left( \{ h_v^{(k)} , \forall v \in N(u) \} \right) \\
h_u^{(k+1)} &\leftarrow \left[ h_u^{(k)} \oplus UPDATE^{(k)} \left( h_u^{(k)} , m_{N(u)}^{(k)} \right) \right]
\end{align*}
\]
Problems – Oversquashing

**Issue**

**Possible solution**
- Add edges
  - Changes topology
  - May lead to oversmoothing
A GNN is an optimizable transformation on all attributes of the graph (nodes, edges)
Summary

Input → Graph network layer → Node embeddings

- Input
- Graph network layer (layer 1, layer 2, ..., layer K)
- Node embeddings
A GNN is an optimizable transformation on all attributes of the graph (nodes, edges)