Distributed, Egocentric Representations of Graphs for Detecting Critical Structures

Ruo-Chun Tzeng[1] Shan-Hung Wu[2]

[1]Microsoft Inc, Taiwan

[2]National Tsing Hua University, Taiwan

International Conference on Machine Learning, 2019

Goal

- To learn representations of graphs ***by using convolutions***

Goal

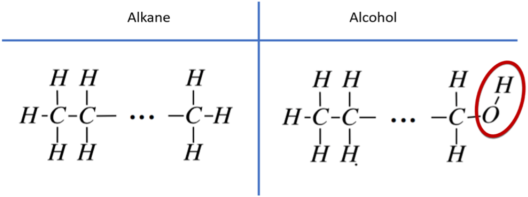
- To learn representations of graphs ***by using convolutions***
- ...while keeping nice properties of CNNs on images:
  - Filters detect location independent patterns
  - Filters at a deep layer have enlarged receptive fields

Goal

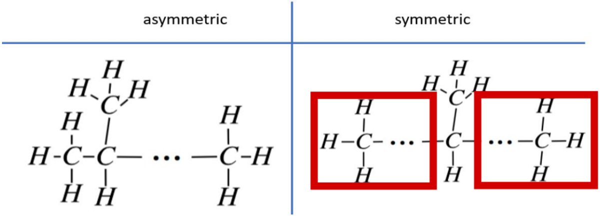
- To learn representations of graphs ***by using convolutions***
- ...while keeping nice properties of CNNs on images:
  - Filters detect location independent patterns
  - Filters at a deep layer have enlarged receptive fields
- ...and being able to detect ***critical structures***

What are the critical structures?

- ***Local-scale*** critical structures, e.g., Alkane vs Alcohol



- ***Global-scale*** critical structures, e.g, Hydrocarbon



STOA: Graph Attention Networks (GAT)[1]

- The (1-head) GAT learns an ***attention score*** α_{ij} for each edge (i, j)

$$\boldsymbol{h}_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \boldsymbol{W} \boldsymbol{h}_j^{(l-1)} + b\right)$$

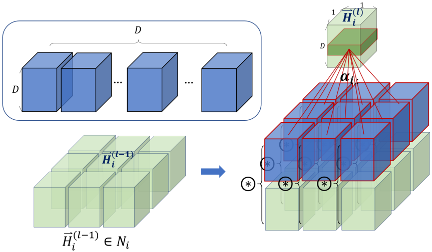
- α's explicitly point out the critical structures
    - When jointly learned with a task, α_{ij} denotes the contribution of edge (i, j) to the model prediction

[1] P Veličković, G Cucurull, A Casanova, A Romero, P Lio, Y Bengio, Graph attention networks, ICLR'18

Drawback: limited learning ability

- However, the (1-head) GAT suffers from limited learning ability

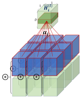
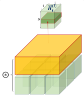
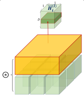
$$\boldsymbol{h}_i^{(l)} = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij} \boldsymbol{W} \boldsymbol{h}_j^{(l-1)} + b\right)$$



  - A filter $\boldsymbol{W}$ scans one node at a time
  - ***does not capture the interactions between nodes***
- Not a serious problem for node-level (e.g., classification) tasks
- But may severely degrade the performance of graph-level tasks

A new way: Ego-CNN

- Idea: to learn critical structures just like image-based CNNs

1-head GAT (ICLR'18)		$\boldsymbol{h}_i^{(l)} = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij} \boldsymbol{W} \boldsymbol{h}_j^{(l-1)} + b\right)$
Traditional Convolution		$\boldsymbol{h}_i^{(l)} = \sigma\left(\boldsymbol{W} \circledast \Vert_{j\in\mathcal{N}_i} \boldsymbol{h}_j^{(l-1)} + b\right)$
Ego-Convolution (ours)		$\boldsymbol{h}_i^{(l)} = \sigma\left(\boldsymbol{W} \circledast \Vert_{j\in\mathcal{N}_i} \boldsymbol{h}_j^{(l-1)} + b\right)$

1. For each node i, a filter $\boldsymbol{W}$ is applied to ***all nodes in the neighborhood*** $\mathcal{N}_i$ of i
2. Use common visualization techniques (e.g., deconv) to backtrack critical structures

Challenge: variable-sized $\mathcal{N}_i$ makes W ill-defined

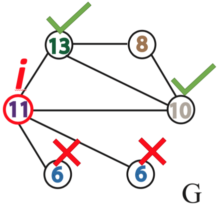
$$\boldsymbol{h}_i^{(l)} = \sigma\left(\boldsymbol{W} \circledast \|_{j\in\mathcal{N}_i}\boldsymbol{h}_j^{(l-1)} + b\right)$$

- For images, $\mathcal{N}_i$ can be easily defined
  - E.g., $K \times K$ pixel block centered at i
- But how to define $\mathcal{N}_i$ ***for graphs***?

Challenge: variable-sized $\mathcal{N}_i$ makes W ill-defined

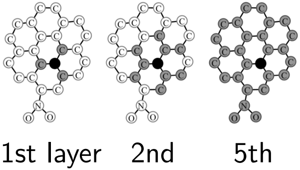
$$\boldsymbol{h}_i^{(l)} = \sigma\left(\boldsymbol{W} \circledast \|_{j\in\mathcal{N}_i}\boldsymbol{h}_j^{(l-1)} + b\right)$$

- For images, $\mathcal{N}_i$ can be easily defined
  - E.g., $K \times K$ pixel block centered at i
- But how to define $\mathcal{N}_i$ ***for graphs***?
- Solution: nodes that are most salient to the given task in a ego-network centered at i
  1. First layer: set $\mathcal{N}_i^{(1)}$ as the ***top K unique nodes in Weisfeiler-Lehnman labeling***
  2. Deep layers: $\mathcal{N}_i^{(l)} = \mathcal{N}_i^{(l-1)}$ (just like image-based CNNs)

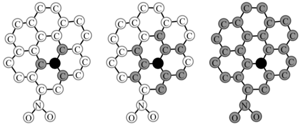


Improved learning ability on graph classification

- In Ego-CNN, a $\boldsymbol{W}$ at layer l can detect **node interaction patterns within l-hop ego-networks**



Improved learning ability on graph classification



1st layer 2nd 5th

- In Ego-CNN, a $\boldsymbol{W}$ at layer l can detect **node interaction patterns within l-hop ego-networks**

- Graph classification benchmark datasets

Table: Bioinformatic Datasets

Dataset	MUTAG	PTC	PROTEINS	NCI1
WL kernel	82.1	57.0	73.0	82.2
DGK	82.7	57.3	71.7	62.5
Subgraph2vec	87.2	60.1	73.4	80.3
MLG	84.2	63.6	**76.1**	80.8
Structure2vec	88.3			**83.7**
DCNN	67.0	56.6		62.6
Patchy-San	92.6	60.0	75.9	78.6
1-head GAT	81.0	57.0	72.5	74.3
Ego-CNN	**93.1**	**63.8**	73.8	80.7

Table: Social Network Datasets

Dataset	IMDB (B)	IMDB (M)	REDDIT (B)	COLLAB
DGK	67.0	44.6	78.0	73.0
Patchy-San	71.0	45.2	86.3	72.6
1-head GAT	70.0	-	78.8	-
Ego-CNN	**72.3**	**48.1**	**87.8**	**74.2**

- With $K = 16$, Ego-CNN is comparable to the state-of-the-arts

Ego-CNN can learn critical structures WITHOUT α

- Backtracking $\boldsymbol{W}$ with common CNN visualization techniques (e.g., deconv) reveals critical structures

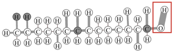
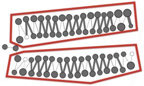
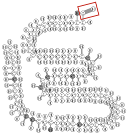
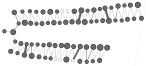
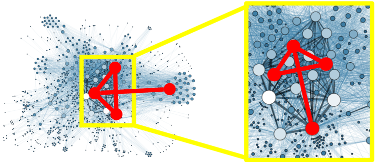
Local-Scale: Alkane vs Alcohol	Global-Scale: Symmetric vs Asymmetric
	
(a) $C_{14}H_{29}OH$	(c) Symmetric Isomer
	
(b) $C_{82}H_{165}OH$	(d) Asymmetric Isomer

Table: Visualization of the critical structures detected by Ego-CNN

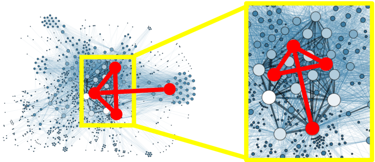
More benefits... and let's chat at Poster #22

- Ego-CNN can detect ***self-similar patterns***
  - I.e., same patterns that exist at different zoom levels
  - Commonly exist in social networks
- How?

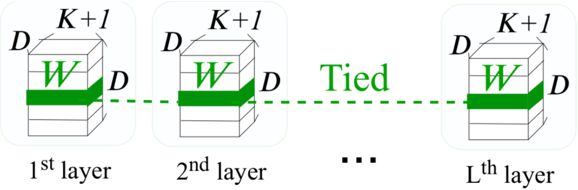


More benefits... and let's chat at Poster #22

- Ego-CNN can detect ***self-similar patterns***
  - I.e., same patterns that exist at different zoom levels
  - Commonly exist in social networks



- How? By simply tying the weights ($\boldsymbol{W}$'s) across different layers



- For more details, please go to ***Poster #22***