Graph U-Nets

Hongyang Gao[1] Shuiwang Ji[1]

[1]Texas A&M University, TX, USA

ICML 2019

Introduction

- Convolutional neural networks (CNNs) have been very succesful in image-related tasks.
- Images can be considered as special cases of graphs, in which nodes lie on regular 2D lattices.
- An important part of CNNs is the pooling (down-sampling) operation, which enables high-level feature encoding and receptive field enlargement.

Related works

- Topology based pooling: $O(|V|^3)$ for eigendecomposition; result is not very good.
- DiffPool*: $O(k|V|^2)$

*Ying, R., You, J., Morris, C., Ren, X., Hamilton, W. L., and Leskovec, J. Hierarchical graph representation learning with differentiable pooling. CoRR, abs/1806.08804, 2018.

Graph U-Nets

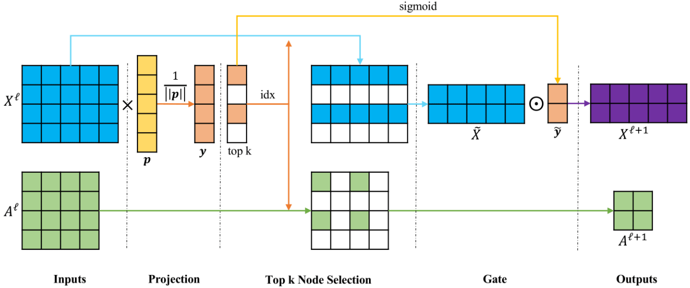
- Proposed gPool and gUnpool operation that are counterparts of the pooling and up-sampling of CNNs respectively.
- A encoder-decoder structure based on the two operations. Similar to the U-Net for images.
- Experiments show it outperforms the GNNs without gPool and gUnpool operations in both inductive and transductive tasks.

gPool operation

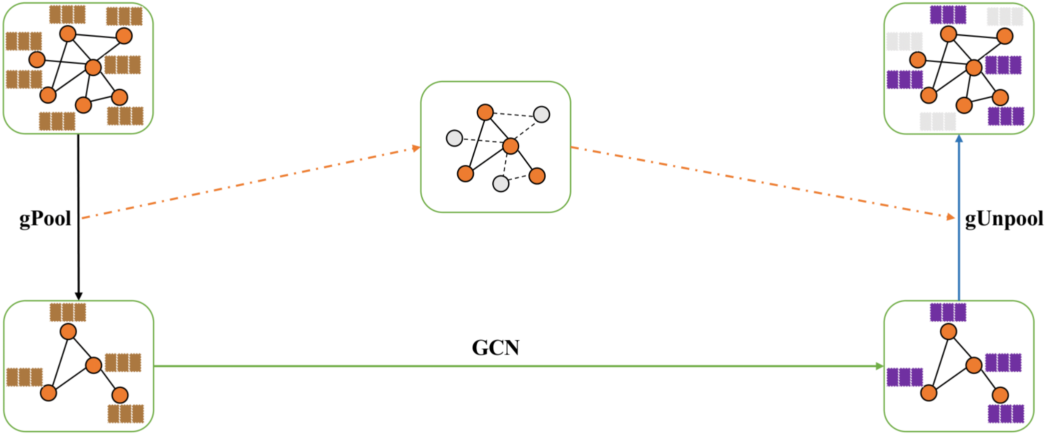
$$X_{l+1} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}X_l W_l)$$

$$\Downarrow$$

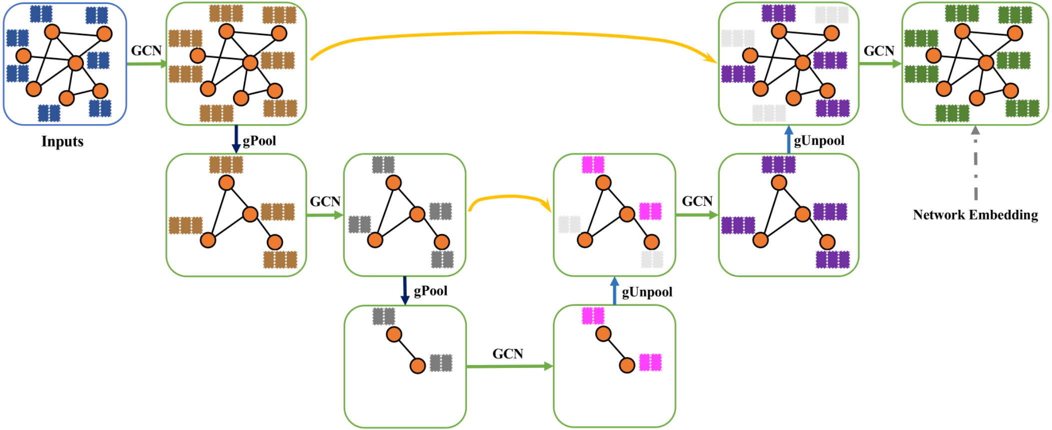
$$\begin{aligned} y &= X^l p^l / \|p^l\|, \\ \mathsf{idx} &= \mathsf{rank}(y, k), \\ \tilde{y} &= \mathsf{sigmoid}(y(idx)), \\ \tilde{X}^l &= X(idx, :), \\ A^{l+1} &= A^l(idx, idx), \\ X^{l+1} &= \tilde{X}^l \odot (\tilde{y}\mathbf{1}_C^T) \end{aligned}$$



gUnpool operation



Graph U-Net


Inputs
GCN
gPool
GCN
gPool
GCN
gUnpool
GCN
gUnpool
GCN
Network Embedding

Graph Augmentation

After the removal of some nodes, the graph may be broken into disconnected parts. To handle this, Graph U-Net augments the graph with:

$$A^2 = A^l A^l, \quad A^{l+1} = A^2(idx, idx)$$

Also they empirically found that adding higher weights to self-loops can increase the performance:

$$\hat{A} = A + 2I$$

Datasets

Table 1. Summary of datasets used in our node classification experiments (Yang et al., 2016; Zitnik & Leskovec, 2017). The Cora, Citeseer, and Pubmed datasets are used for transductive learning experiments.

Dataset	Nodes	Features	Classes	Training	Validation	Testing	Degree
Cora	2708	1433	7	140	500	1000	4
Citeseer	3327	3703	6	120	500	1000	5
Pubmed	19717	500	3	60	500	1000	6

Table 2. Summary of datasets used in our inductive learning experiments. The D&D (Dobson & Doig, 2003), PROTEINS (Borgwardt et al., 2005), and COLLAB (Yanardag & Vishwanathan, 2015) datasets are used for inductive learning experiments.

Dataset	Graphs	Nodes (max)	Nodes (avg)	Classes
D&D	1178	5748	284.32	2
PROTEINS	1113	620	39.06	2
COLLAB	5000	492	74.49	3

Transductive task (node classification)

Table 3. Results of transductive learning experiments in terms of node classification accuracies on Cora, Citeseer, and Pubmed datasets. g-U-Nets denotes our proposed graph U-Nets model.

Models	**Cora**	**Citeseer**	**Pubmed**
DeepWalk (Perozzi et al., 2014)	67.2%	43.2%	65.3%
Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	79.0%
GAT (Veličković et al., 2017)	$83.0 \pm 0.7\%$	$72.5 \pm 0.7\%$	$79.0 \pm 0.3\%$
g-U-Nets (Ours)	$\mathbf{84.4 \pm 0.6\%}$	$\mathbf{73.2 \pm 0.5\%}$	$\mathbf{79.6 \pm 0.2\%}$

Inductive task (graph classification)

Table 4. Results of inductive learning experiments in terms of graph classification accuracies on D&D, PROTEINS, and COLLAB datasets. g-U-Nets denotes our proposed graph U-Nets model.

Models	**D&D**	**PROTEINS**	**COLLAB**
PSCN (Niepert et al., 2016)	76.27%	75.00%	72.60%
DGCNN (Zhang et al., 2018)	79.37%	76.26%	73.76%
DiffPool-DET (Ying et al., 2018)	75.47%	75.62%	**82.13**%
DiffPool-NOLP (Ying et al., 2018)	79.98%	76.22%	75.58%
DiffPool (Ying et al., 2018)	80.64%	76.25%	75.48%
g-U-Nets (Ours)	**82.43%**	**77.68%**	77.56%

Ablation study

Table 5. Comparison of g-U-Nets with and without gPool or gUnpool layers in terms of node classification accuracy on Cora, Citeseer, and Pubmed datasets.

Models	**Cora**	**Citeseer**	**Pubmed**
g-U-Nets without gPool or gUnpool	$82.1 \pm 0.6\%$	$71.6 \pm 0.5\%$	$79.1 \pm 0.2\%$
g-U-Nets (Ours)	$\mathbf{84.4} \pm \mathbf{0.6\%}$	$\mathbf{73.2} \pm \mathbf{0.5\%}$	$\mathbf{79.6} \pm \mathbf{0.2\%}$

Table 6. Comparison of g-U-Nets with and without graph connectivity augmentation in terms of node classification accuracy on Cora, Citeseer, and Pubmed datasets.

Models	**Cora**	**Citeseer**	**Pubmed**
g-U-Nets without augmentation	$83.7 \pm 0.7\%$	$72.5 \pm 0.6\%$	$79.0 \pm 0.3\%$
g-U-Nets (Ours)	$\mathbf{84.4} \pm \mathbf{0.6\%}$	$\mathbf{73.2} \pm \mathbf{0.5\%}$	$\mathbf{79.6} \pm \mathbf{0.2\%}$

Parameter size

Table 8. Comparison of the g-U-Nets with and without gPool or gUnpool layers in terms of the node classification accuracy and the number of parameters on Cora dataset.

Models	Accuracy	#Params	Ratio of increase
g-U-Nets without gPool or gUnpool	$82.1 \pm 0.6\%$	75,643	0.00%
g-U-Nets (Ours)	$\mathbf{84.4 \pm 0.6\%}$	75,737	0.12%

This suggests that the improvement is not a result of more parameters, and adding gPool and gUnpool will not increase the risk of over-fitting.

What I learnt

- We can find inspirations from CNN techniques when dealing with GNN.
- The adjacency matrix can be augmented to impose different weights for links.

Thank you!