Local Augmentation for Graph Neural Networks

ICML 2022

Songtao Liu[1†], Rex Ying[2], Hanze Dong[3], Lanqing Li[4], Tingyang Xu[4], Yu Rong[4], Peilin Zhao[4], Junzhou Huang[4], Dinghao Wu[1]

[1]The Pennsylvania State University
[2]Stanford University
[3]Hong Kong University of Science and Technology
[4]Tencent AI Lab
[†]Part of the work was done during Songtao's internship at Tencent AI Lab

Backgrounds: Aggregating local information in Graph Neural Networks

1. The key idea for GNNs is to aggregate information from local neighborhoods.

2. Deep GNNs, such as JKnet and GCNII also preserve the locality of node representations.

3. Subgraph GNNs such as GraphSNN and ShadowGNN utilize the structure information of local neighborhoods to enhance the expressive power.

Open question: Whether the local information is adequately aggregated for learning representations of nodes with few neighbors?

1. The *limited* number of neighbors in the local neighborhood restricts the expressive power of GNNs and hinders their performance.

2. Stacking graph layers to incorporate more neighbors is not a solution due to over-smoothing.

Backgrounds: Graph data augmentation

1. Topology-level augmentation methods (DropEdge, Gaug, etc.) perturb the adjacency matrix.

2. Feature-level augmentation methods (FLAG, etc.) exploit perturbation of node attributes guided by adversarial training.

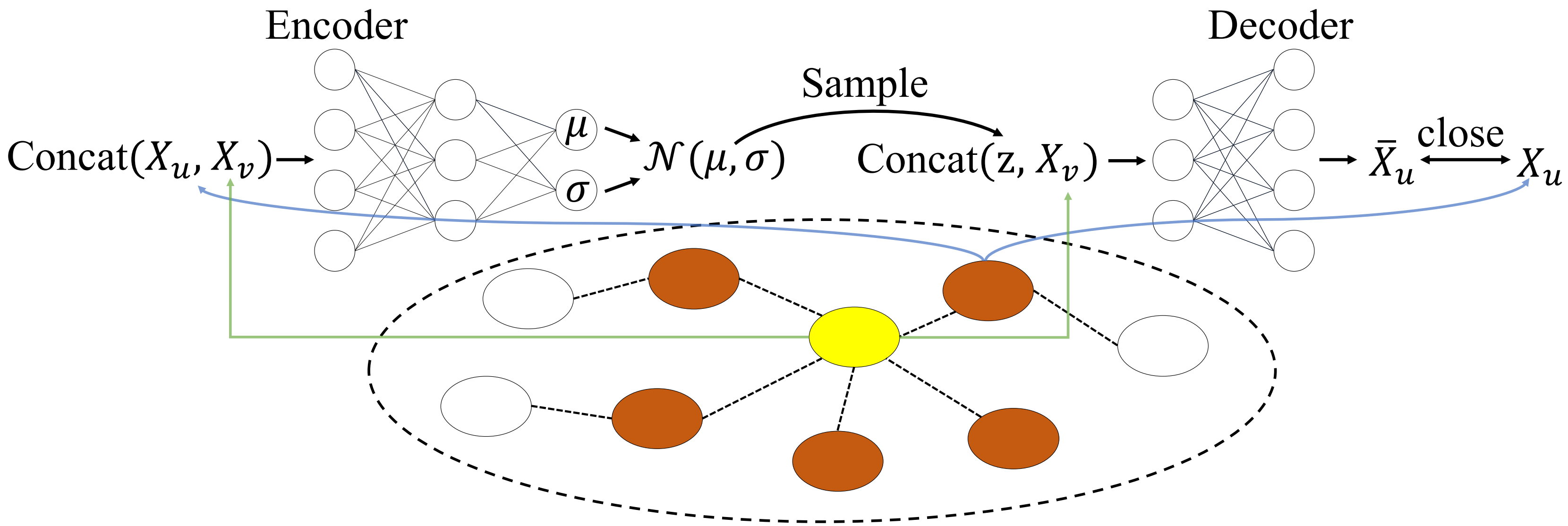
Motivation

1. Generative model can capture the distribution of the local neighborhood information

2. Enrich information in the local neighborhood via our local augmentation to generate more features to enhance the expressive power of GNNs.

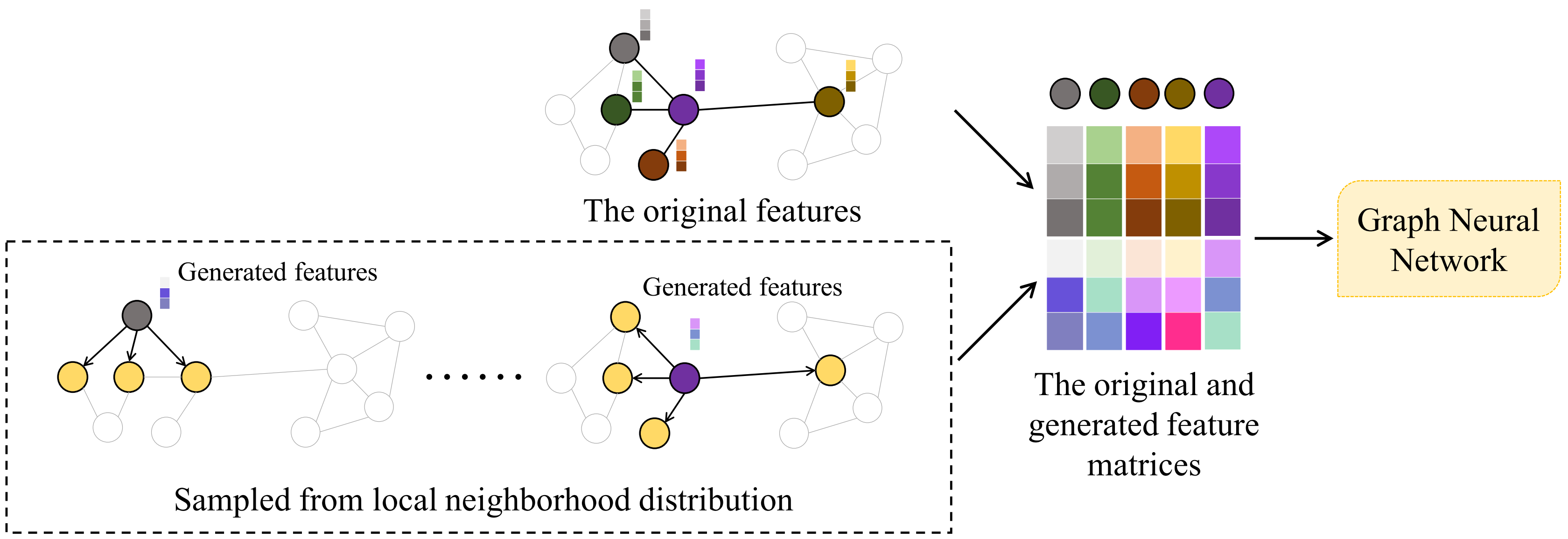
Approach

1. Learn the conditional distribution of connected neighbors' node features given on center node's features via a generative model via conditional variational auto-encoder (CVAE)



Approach

2. Exploit the well-learned distribution to generated feature vectors to enhance the expressive power of GNNs at the training stage.



Evaluation Results

Method	Cora	Citeseer	Pubmed
Chebyshev	81.2	69.8	74.4
APPNP	83.8±0.3	71.6±0.5	79.7±0.3
MixHop	81.9±0.4	71.4±0.8	80.8±0.6
Graph U-net	84.4±0.6	73.2±0.5	79.6±0.2
GSNN-M	83.9±0.5	72.2±0.5	79.1±0.3
S^2GC	83.5±0.02	73.6±0.09	80.2±0.02
GCN	81.5±0.5	70.3±0.7	79.0±0.5
G-GCN	83.7	71.3	80.9
DropEdge-GCN	82.8	72.3	79.6
GAUG-O-GCN	83.6±0.5	73.3±1.1	79.3±0.4
GraphSNN$_{GCN}$	83.1±1.8	72.3±1.5	79.8±1.2
GRAND-GCN	84.5±0.3	74.2±0.3	80.0±0.3
LA-GCN	**84.6±0.5**	**74.7±0.5**	**81.7±0.7**
GAT	83.0±0.7	72.5±0.7	79.0±0.3
GAUG-O-GAT	82.2±0.2	71.6±1.1	OOM
GraphSNN$_{GAT}$	83.8±1.2	73.5±1.6	79.6±1.4
GRAND-GAT	84.3±0.4	73.2±0.4	79.2±0.6
LA-GAT	**84.7±0.4**	**73.7±0.5**	**81.0±0.4**
GCNII	85.5±0.5	73.4±0.6	80.2±0.4
LA-GCNII	**85.7±0.3**	**74.1±0.5**	**80.6±0.7**
GRAND	85.4±0.4	75.4±0.4	82.7±0.6
LA-GRAND	**85.7±0.3**	**75.8±0.5**	**83.4±0.6**

Model	products Acc	proteins ROC-AUC	arxiv Acc
MLP	61.06±0.08	72.04±0.48	55.50±0.23
CoLinkDistMLP	62.59±0.10	-	56.38±0.16
Node2vec	72.49±0.10	68.81±0.65	70.07±0.13
GraphZoom	74.06±0.26	-	71.18±0.18
GCN	75.64±0.21	72.51±0.35	71.74±0.29
+FLAG	-	71.71±0.50	72.04±0.20
+GraphSNN	-	-	**72.20±0.90**
+LA	**76.11±0.09**	**73.25±0.51**	72.08±0.14
GraphSAGE	78.70±0.36	77.68±0.20	71.49±0.27
+FLAG	79.36±0.57	76.57±0.75	72.19±0.21
+GraphSNN	-	-	71.80±0.70
+LA	**79.44±0.25**	**77.86±0.37**	**72.30±0.12**
GAT	79.45±0.59	-	73.65±0.11
+FLAG	**81.76±0.45**	-	73.71±0.13
+LA	80.46±0.54	-	**73.77±0.12**

Ablation study

Technique	Accuracy (%)	Δ	Cumu Δ
GCN	79.0	0	0
+ Concatenation	79.3±0.4	0.3	0.3
+ Local Augmentation	81.1±0.5	1.8	2.1
+ Consistency Training	81.4±0.5	0.3	2.4
+ Sharpening Trick	81.7±0.7	0.3	2.7

Case Study

Degree	[2, 5]	[6, 20]
#Nodes	761	189
GCN	78.2	82.0
LAGCN	79.9	82.2
Δ	1.7	0.2

PennState

Any Questions?

Code: https://github.com/SongtaoLiu0823/LAGNN

arXiv: https://arxiv.org/pdf/2109.03856.pdf

Contact: skl5761@psu.edu

Thank you!

PennState