Local Augmentation for Graph Neural Networks

CS768: Learning with Graphs Course Project

Prateek Garg Bhavya Kohli Aziz Shameem

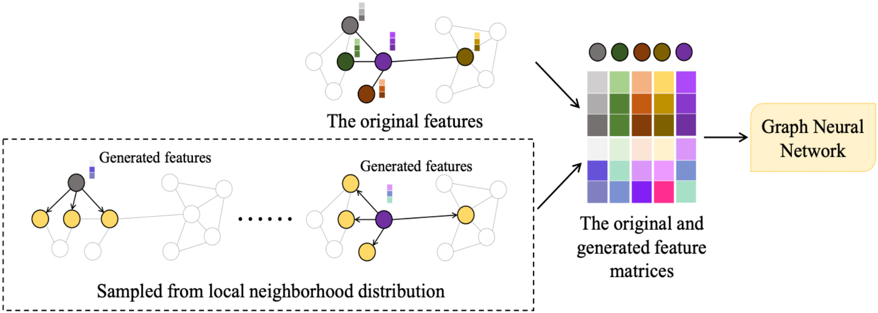
Indian Institute of Technology Bombay

{*prateekg,bhavyakohli,20d070020*}*@iitb.ac.in*

Final Presentation

February 20, 2025

Main Idea



- Uses a **CVAE**, conditioned on a central node for the generation of neighbouring node features.
  - Enriches the features for nodes which have very few neighbours
- Since the generation process is independent of the training, can be applied to a variety of existing frameworks in a **plug-and-play** manner

Reproduced Results

Table: Semi Supervised Learning

Serial No	Model	Dataset	Obtained acc	Reported acc
1	LA-GCN	Cora	84.47$\pm$0.0053	84.6$\pm$0.5
2	LA-GCN	Citeseer	74.57$\pm$0.0063	74.7$\pm$0.5
3	LA-GCN	Pubmed	81.63$\pm$0.0069	81.7$\pm$0.7
4	LA-GAT	Cora	84.56$\pm$0.0025	84.7$\pm$0.4
5	LA-GAT	Citeseer	73.75$\pm$0.0041	73.7$\pm$0.5
6	LA-GAT	Pubmed	80.87$\pm$0.0012	81.0$\pm$0.4
7	LA-GCN-II	Cora	85.63$\pm$0.0036	85.7$\pm$0.3
8	LA-GCN-II	Citeseer	73.30$\pm$0.0035	**74.1$\pm$0.5**
9	LA-GCN-II	Pubmed	80.56$\pm$0.0041	80.6$\pm$0.7
10	LA-GRAND	Cora	85.76$\pm$0.0033	85.7$\pm$0.3
11	LA-GRAND	Citesee	73.01$\pm$0.0037	**75.8$\pm$0.5**
12	LA-GRAND	Pubmed	81.10$\pm$0.0008	**83.4$\pm$0.6**

Reproduced Results

Table: Full Supervised Learning

Serial No	Model	Dataset	Obtained acc	Reported acc
1	LA-GCN	arxiv	72.15±0.13	72.08±0.14
2	LA-GCN	proteins	68.66±0.39	**73.25±0.51**
3	LA-SAGE	arxiv	72.37±0.06	72.30±0.12
4	LA-SAGE	proteins	76.56±0.18	**77.86±0.37**
5	LA-GAT	arxiv	73.84±0.0008	73.77±0.12

* metric for protein dataset is AUCROC

Graph Classification Task

In this, we extend the local augmentation technique to graph (instead of node) classification tasks.
Note that this is not straight forward, since the CVAE training in the original set-up uses the classification model, which requires node-level labels(not present in graph classification datasets).

Workaround:
Form a large graph (called *big_graph*) using the graphs in the dataset, and use the individual graph labels for the training.

Results on Graph Classification

Table: Full Supervised Learning

Serial No	Experiment	Model	Accuracy
1	MUTAG	GIN	0.6759±0.0833
2	**LA-MUTAG**	**GIN**	**0.7379±0.1187**
3	BZR	GIN	0.7672±0.0318
4	**LA-BZR**	**GIN**	**0.7902±0.0161**
5	DHFR	GIN	0.7404±0.0690
6	**LA-DHFR**	**GIN**	**0.7719±0.0633**

⋆ the model is a Graph convolutional model using GIN layers

⋆ the model uses global add pooling for the graph classification step

Link Prediction Task

The Local Augmentation framework works by enriching node features using a conditional generator, and is not exactly application specific.

For this task, training the CVAE did not require any big workarounds unlike for graph classification, but the model architecture which does the final prediction of course had to be different.

The model we chose was a simple GCN with an inner product decoder as the link predictor $\sigma(z_i^T \cdot z_j)$. Training involved using negative sampling as is standard for link prediction tasks, and the metric used is AUROC

Results on Link Prediction

Table: Full Supervised Learning

Serial No	Experiment	model	AUROC
1	CORA	GCN-IP	0.8772±0.0149
2	**LA-CORA**	**GCN-IP**	**0.9262±0.0013**
3	CITESEER	GCN-IP	0.8378±0.0282
4	**LA-CITESEER**	**GCN-IP**	**0.9093±0.0110**
5	PUBMED	GCN-IP	0.9165±0.0035
6	**LA-PUBMED**	**GCN-IP**	**0.9432±0.0037**
7	REED98	GCN-IP	0.8271±0.0004
8	**LA-REED98**	**GCN-IP**	**0.8286±0.0020**
9	AMHERST41	GCN-IP	0.8600±0.0086
10	**LA-AMHERST41**	**GCN-IP**	**0.8676±0.0007**

⋆ GCN-IP stands for GCN with Inner Product decoder

Normalizing Flows

- To test effectiveness of this technique across method, we train a Normalising flow based conditional generative model as drop-in replacement for the CVAE.
- We train a model with 4 layers, where each layer is composed of an affine transformation, followed by sigmoid transformation.

Results using the Normalizing flows model

- One particular step in normalizing is to construct the conditional distribution.
- This is memory intensive step, since it requires us to construct a covariance matrix using `torch diag_embed` function, in each forward pass.
- We were able to train this model on Cora Dataset, where we achieved the U-score of **-0.00543**, compared to the given VAE of similar number parameters which had the U-score of **-0.0354**
- Unfortunately, the time required to run main training on any dataset exceeded our colab quota, so we weren't able to test.

Future Work : Extending to 2-hop neighbours

The original code uses information from 1-hop neighbours of a node for the conditional generation of to-be-augmented features. We had originally planned to extend this to using 2-hop neighbours, to compare the performance boost obtained. This, however, will increase the features dimension considerably, thereby creating resource constraints.

Concluding Remarks

- Local-Augmentation is as good as the quality of generated node features.
- The system neither depends on the specific generation algorithm, nor the task at hand. Thus, a trade-off is established between the quality of generation and the resource/time constraints.
- the code can be here: BhavyaKohli/CS768-Project-LAGNN