Compression of Graph Neural Networks

Abhinay, Kalpit, Naman

Problem Statement

- Our objective is to compress Graph neural networks with Knowledge distillation, a method widely used to compress Neural Networks but not explored much in graphs.
- Graph networks are widely used in a variety of datasets that have a graphical structure like social media datasets. These datasets have relational nature and to learn them, networks get dense
- The large model size of GNN becomes the bottleneck in distributing this model to edge computing devices like Mobiles and embedded devices.

Related work

- For model compression, knowledge distillation was developed, in which a tiny light-weight student model is taught to imitate the soft predictions of a big teacher model that has been pre-trained. The knowledge from the instructor model will be transmitted to the student model after distillation. The student model can therefore decrease time and spatial complexity while maintaining prediction quality.
- Knowledge distillation has been implemented and tested only for neural networks where the networks are trained on grid based datasets.

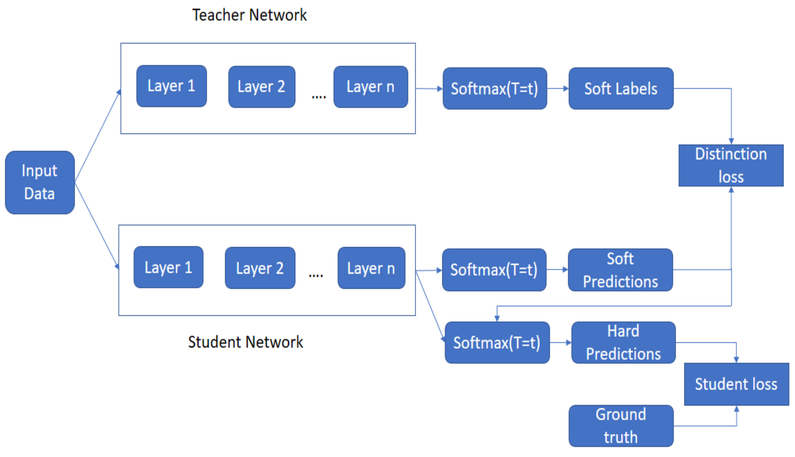

Figure 1: Knowledge Distillation

Approach

- The aggregation approach is critical for incorporating the node characteristics during the training process. However, distilling information that accurately describes the aggregation function and transferring it to the student is difficult.
- Instead of distilling the aggregation function directly, we distill the outcomes of such function: the embedded topological structure. The student can then be guided by matching the str ucture embedded by itself and that embedded by the teacher.
- By matching similarity score between student and teacher distributions, topology-aware knowledge transfer from the teacher is enabled.

Knowledge Distillation with LSP

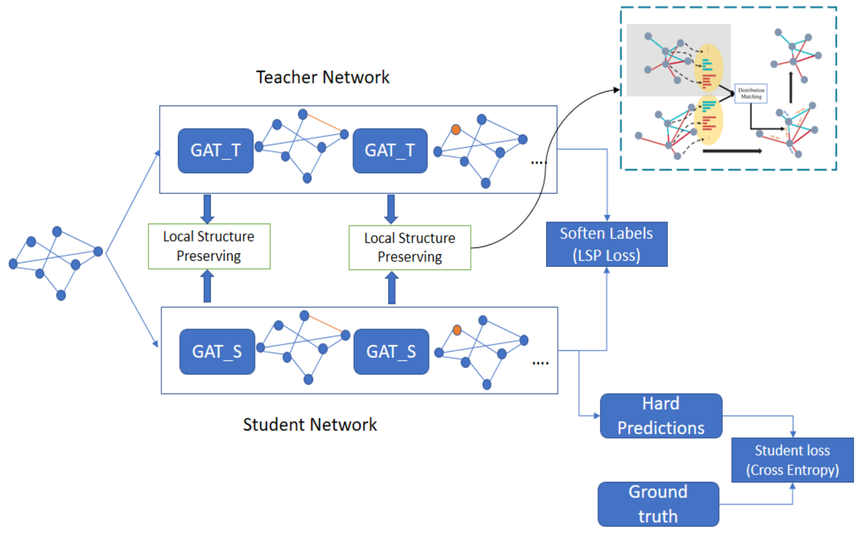

Figure 2: Knowledge Distillation with LSP for GAT

- We first compute the distribution of the local structure for each node and then match the distributions of the teacher with that of the student. The student model will be optimized by minimizing the difference between the distributions among all local structures.
- The total loss is formulated as:

$$L = H(p_s, y) + \lambda L_{lsp}$$

where y is the label and p_s is the prediction of the student model, λ is the hyperparameter to balance these two losses and H represents cross entropy loss function

Early results

Model	Attention heads	Layers	Hidden features	# of Parameters	Accuracy
Teacher	4,4,6	3	256,256,121	3643522	98.07
Student	2,2,2,2,2	5	64,64,64,64,121 (Base)	168918	95.83
Student (w/o KD)	2,2,2,2,2	5	64,64,64,64,121	168918	94.2

# of Parameters	Accuracy	Compression Factor	KD Transfer Epoch	Hidden Features	Layers
60246	83.04	60	100	32	5
68630	83.92	53	100	32	6
168918	95.17	21.5	100	64	5
202070	96.29	18	100	64	6
60246	83.56	60	50	32	5
68630	84.1	53	50	32	6
168918	95.83	21.5	50	64	5
202070	96.43	18	50	64	6

- Our experiments are performed on Protein-Protein Interaction (PPI) dataset where the graphs come from human tissues.
- We adopt Graph Attention Network for student and teacher for node classification.
- Our model was able to achieve 96% accuracy with >20x compression. There is a tradeoff between compression factor and accuracy. There is direct relation between #layers/#hidden nodes in student and accuracy.
- Number of epochs for knowledge transfer is also a hyper-parameter.

Work in Progress

- We can use set of student networks to build a more powerful student model via ensemble learning.
- We are analyzing the trade off between attention heads and number of hidden layers in GAT.
- Our Experiments are on Node Classification using Knowledge distillation and local structure preserving. How does the result change when graph contain different relations and student model should perform link prediction or graph classification.

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON