Efficient Training of Graph Classification Models

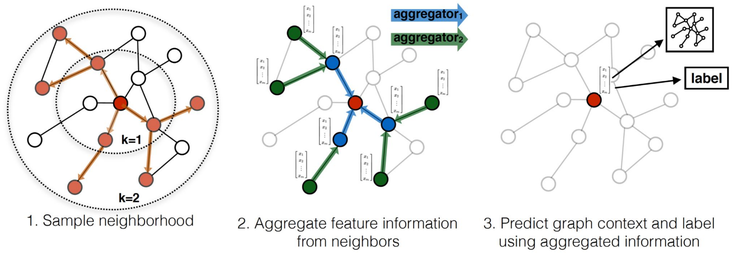
Hao-Yu Shih, Winfred Li, Noah Cohen Kalafut

Problem Statement

- *Graph classification* is a subset of *graph analytics* that focuses on producing a classification (e.g. +/-, healthy/unhealthy) from an input graph.
- Graph classification models are utilized in medical technologies, document tagging, and many other vital services.
- Existing solutions are not specifically focused on graph classification, and rather focus on the broader field of *graph analytics*.

Related work

- *Two stage training* focuses on efficient training through transfer learning; specifically referring to embedding generation models. However, it doesn't attempt to parallelize the two training steps.
- *Distributed Computing for Large-Scale Graphs* creates several standalone applications for use in graph computation on distributed systems, However, these approaches are not specifically tailored to graph classification.
- *GraphSAGE* is a graph neural network we study in this project. Each layer aggregates neighboring nodes, concatenates with the current node, applies a linear transformation and then an activation function.



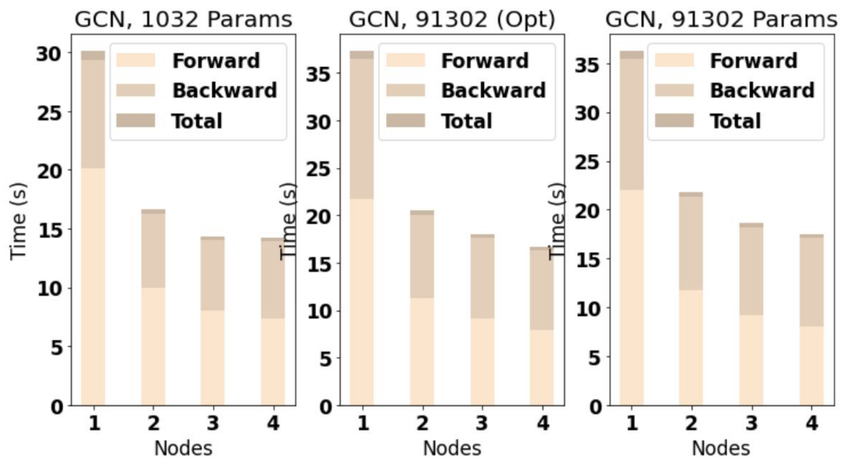
1. Sample neighborhood
2. Aggregate feature information from neighbors
3. Predict graph context and label using aggregated information

http://snap.stanford.edu/graphsage

What is your Approach

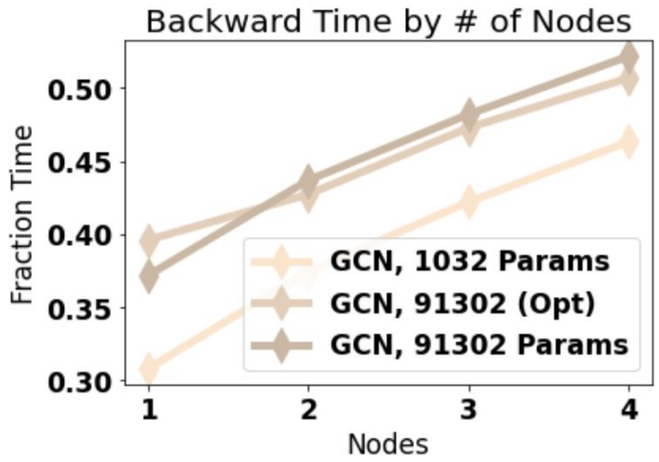
- Find out what the best algorithms need for training
- Minimize time spent not training with a minimally invasive approach
- Verify results on standard models
- Provide similar convergence properties and classification accuracy results to standard optimization
- Test application on state-of-the-art graph classification models

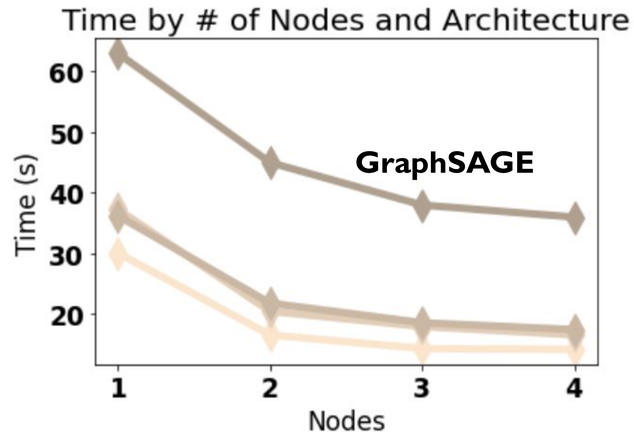
DGLight



- We compare communication overhead and computation overlap on standard models using *DGL*-standard optimization.
- We analyze the time distribution while controlling for model complexity.
- Implement partial synchronizations to cut down on communication cost for larger models.
- We test our solutions on state-of-the-art algorithms

Early results





- Synchronization skips combined with AllReduce provide better scaling properties
- Staggered ring-reduce (partial node-by-node synchronization) shows promise

Other Contributions / WIP

- Experiment with other data partitioning techniques
- Various weight update strategies (modified ring-reduce, etc.)
- Implement more models
- Test using more nodes

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON