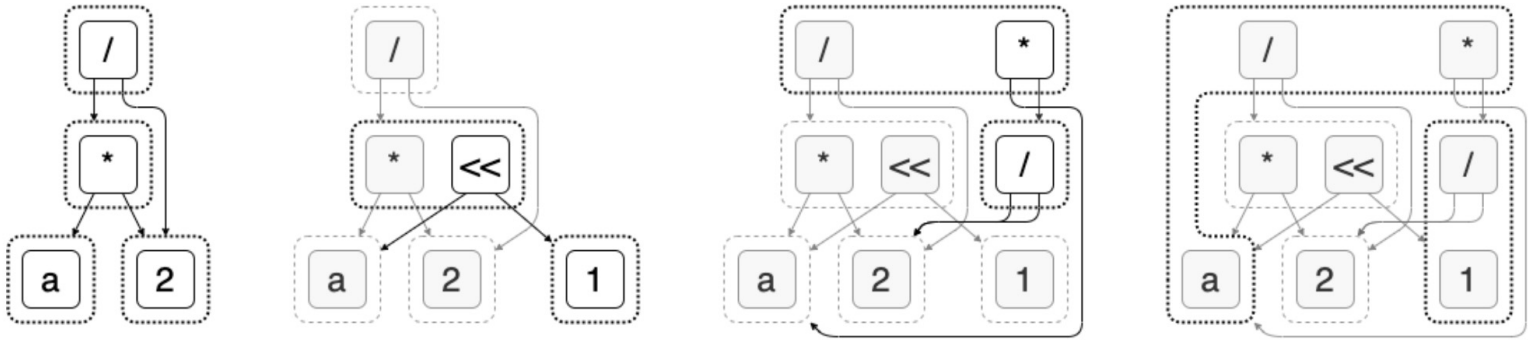
Tensor graph superoptimization for graph neural networks

Project Proposal by Antonia Boca

Motivation

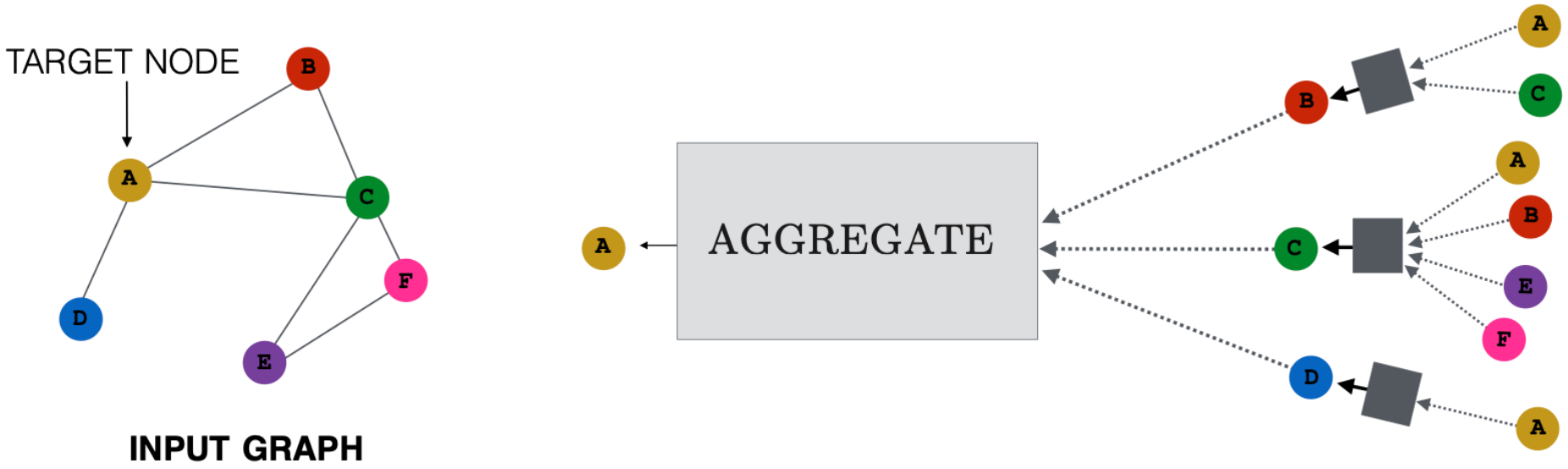
- Tensat [1] uses *equality saturation* to optimize computation graphs



- It is built on top of egg [2] and TASO [3]
- It performs equality saturation for a bunch of operators…
- …but not for GNN operators!

Graph Neural Networks

- Most popular GNNs use the 'message-passing' framework
- Each architecture has an associated aggregation **operator**



The basic GNN operator

- A “layer” in a GNN is usually an iteration of the aggregation & update operators

$$\mathbf{h}_u^{(k+1)} = \mathrm{UPDATE}^{(k)}(\mathbf{h}_u^{(k+1)}, \mathrm{AGGREGATE}^{(k)}(\{\mathbf{h}_v^{(k+1)}, \forall v \in \mathcal{N}(u)\})) \quad (1)$$

$$= \mathrm{UPDATE}^{(k)}(\mathbf{h}_u^{(k+1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}) \quad (2)$$

- GCN Convolution [4]:

$$\mathbf{h}_u^{(k)} = \sigma(\mathbf{W}^{(k)} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)}})$$

Operator	Description	Inputs	Type signature
ewadd	Element-wise addition	$input_1$, $input_2$	(T, T) → T
ewmul	Element-wise multiplication	$input_1$, $input_2$	(T, T) → T
matmul	Matrix multiplication	activation, $input_1$, $input_2$	(N, T, T) → T
conv [a]	Grouped convolution	$stride_h$, $stride_w$, pad., act., input, weight	(N, N, N, N, T, T) → T
relu	Relu activation	input	T → T
tanh	Tanh activation	input	T → T
sigmoid	Sigmoid activation	input	T → T
poolmax	Max pooling	input, $kernel_{\{h,w\}}$, $stride_{\{h,w\}}$, pad., act.	(T, N, N, N, N, N, N) → T
poolavg	Average pooling	input, $kernel_{\{h,w\}}$, $stride_{\{h,w\}}$, pad., act.	(T, N, N, N, N, N, N) → T
transpose [b]	Transpose	input, permutation	(T, S) → T
enlarge [c]	Pad a convolution kernel with zeros	input, ref-input	
$concat_n$ [d]	Concatenate	axis, $input_1$, …, $input_n$	
split [e]	Split a tensor into two	axis, input	
$split_0$	Get the first output from split	input	
$split_1$	Get the second output from split	input	
merge [f]	Update weight to merge grouped conv	weight, count	
reshape [g]	Reshape tensor	input, shape	
input	Input tensor	identifier [h]	
weight	Weight tensor	identifier [h]	
noop [i]	Combine the outputs of the graph	$input_1$, $input_2$	

Decide on a subset of GNN operators to optimize…

Graph Neural Network Operators

Name	SparseTensor	edge_weight	edge_attr	bipartite	static	lazy
GCNConv (Paper)	✓	✓			✓	✓
ChebConv (Paper)		✓			✓	✓
SAGEConv (Paper)	✓			✓	✓	✓
GraphConv (Paper)	✓	✓		✓	✓	✓
GatedGraphConv (Paper)	✓	✓			✓	
ResGatedGraphConv (Paper)	✓			✓	✓	✓
GATConv (Paper)	✓		✓	✓		✓
GATv2Conv (Paper)	✓		✓	✓		✓
TransformerConv (Paper)	✓		✓	✓		✓

Project idea

- Implement equality saturation for a subset of GNN operators using the egg library
  - Extend Tensat with these new operators
  - Evaluate extended Tensat
- Experiment settings:
  - Run extended Tensat on GNN architectures (e.g. GCN, GAT, GraphSAGE) and benchmark datasets (e.g. Reddit and Citation networks)
- Evaluate architectures optimized by extended Tensat:
  - Against architectures optimized by normal Tensat
  - Against architectures optimized by FlexFlow [5] (built on top of Unity)

Workplan

- Preparation
  - Understand how GCNConv, GATConv, SAGEConv are implemented in pytorch-geometric
  - Learn (basic) Rust
  - Understand the implementation of Tensat operator optimisations
- Come up with optimisations
  - Implement optimisations using egg
  - Extend the Tensat framework
- Run experiments on GNN architectures and datasets
  - NVIDIA core; plan to run this on AWS

Questions and suggestions

References

- [1] Max Willsey, Chandrakana Nandi, Yisu Remy Wang, Oliver Flatt, Zachary Tatlock, and Pavel Panchekha. 2021. Egg: Fast and extensible equality saturation. Proc. ACM Program. *Lang. 5, POPL, Article 23 (January 2021), 29 pages.*
- [2] Yang, Yichen, et al. Equality saturation for tensor graph superoptimization. *Proceedings of Machine Learning and Systems 3 (2021): 255-268.*
- [3] Z. Jia, O. Padon, J. Thomas, T. Warszawski, M. Zaharia, A. Aiken. TASO: Optimizing Deep Learning Computation with Automated Generation of Graph Substitutions. *SOSP, 2019.*
- [4] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907 (2016).*
- [5] Colin Unger, Zhihao Jia, Wei Wu, Sina Lin, Mandeep Baines, Carlos Efrain Quintero Narvaez, Vinay Ramakrishnaiah, Nirmal Prajapati, Pat McCormick, Jamaludin Mohd-Yusof, Xi Luo, Dheevatsa Mudigere, Jongsoo Park, Misha Smelyanskiy, and Alex Aiken. Unity: Accelerating DNN Training Through Joint Optimization of Algebraic Transformations and Parallelization. *In Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI), July 2022.*