L^2-GCN: Layer-Wise and Learned Efficient Training of Graph Convolutional Networks

Yuning You*, Tianlong Chen*, Zhangyang Wang, Yang Shen

Texas A&M University

* Equal Contribution

This work was presented at CVPR

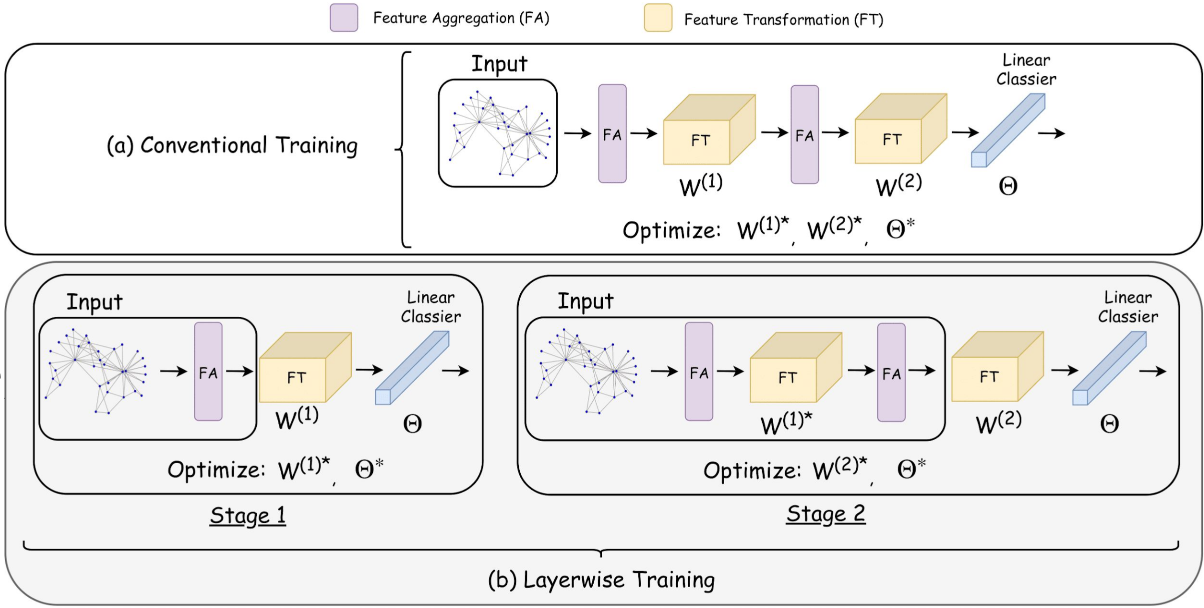
Motivation

- The forward propagation of GCN layer:
  - FA: aggregation of the neighborhood information;
  - FT: non-linear transformation.
- Concatenation of FA & FT → inefficient GCN training for large graphs.
- **Decoupling** FA & FT in GCN training can greatly reduce computational burden.

GCN: graph convolutional network,
FA: feature aggregation,
FT: feature transformation.

L-GCN: Layer-wise GCN

- Propose layer-wise training to decouple FA & FT.
- For each GCN layer, FA is performed **once** then fed for FT.
- Optimization is for each layer individually.



Theoretical Justification of L-GCN

TEXAS A&M UNIVERSITY Engineering

- We provide further analysis following the graph isomorphism framework[1]:
  - The power of aggregation-based GNN := the ability it maps different graphs (rooted subtrees of vertices) into different embeddings;
  - GNN is at most as powerful as the WL test.
- We prove that if GCN is as powerful as the WL test through conventional training, there exists the **same** powerful model through layer-wise training (see Theorem 5).

[1] K. Xu et al. How powerful are graph neural networks? ICLR 2019.
GNN: graph neural network, WL test: Weisfeiler-Lehman test.

Theoretical Justification of L-GCN

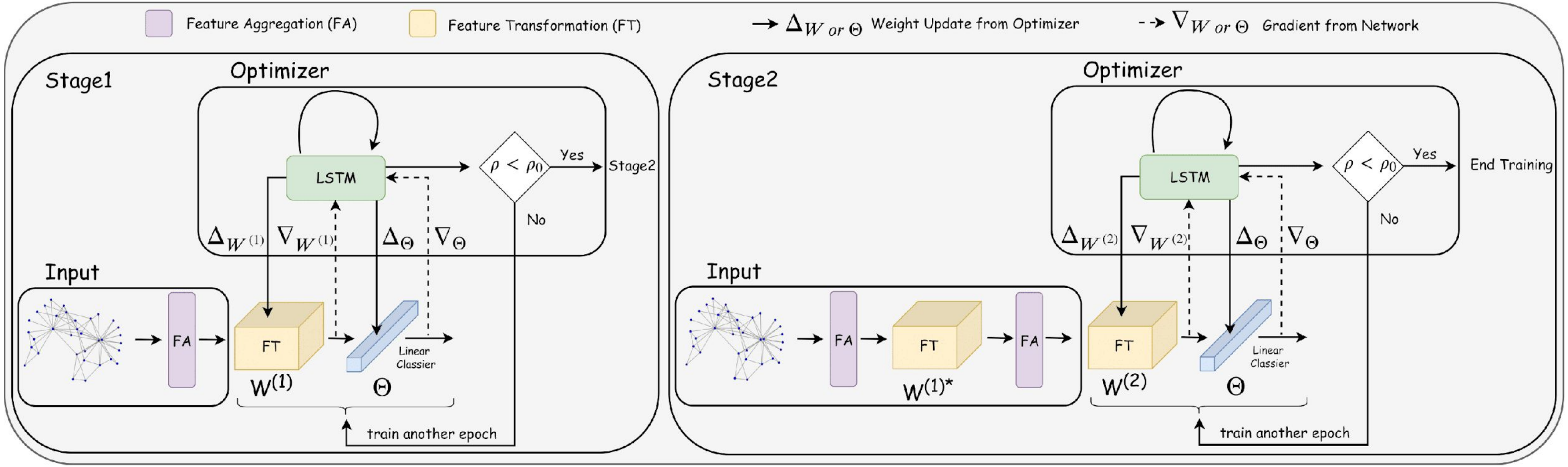
TEXAS A&M UNIVERSITY
Engineering

- Insight in Theorem 5: for the **powerful enough** GCN through conventional training, we might obtain the same powerful model through layer-wise training.

- Furthermore, we prove that if GCN is **not** as powerful as the WL test through conventional training, through layer-wise training its power is non-decreasing with layer number increasing (see Theorem 6).
  - Insight in Theorem 6: for the not powerful enough GCN through conventional training, through layer-wise training we might obtain a more powerful model if we make it deeper.

L^2-GCN: Layer-wise and Learned GCN

- Lastly, to avoid manually adjusting the training epochs for each layer, a learned controller is proposed to automatically deal with this process.



Experiments

TEXAS A&M UNIVERSITY
Engineering

- Experiments show that L-GCN is faster than state-of-the-arts by at least an order of magnitude, with a consistent of memory usage not dependent on dataset size, while maintaining comparable prediction performance. With the learned controller, L^2-GCN can further cut the training time in half.

	GraphSAGE [10]			FastGCN [4]			VRGCN [5]			L-GCN			L^2-GCN		
	F1 (%)	Time	Memory	F1 (%)	Time	Memory	F1 (%)	Time	Memory	F1 (%)	Time	Memory	F1 (%)	Time	Memory
Cora	85.0	18s	655M	85.5	6.02s	659M	85.4	5.47s	253M	84.7	0.45s	619M	84.1	0.38s	619M
PubMed	86.5	483s	675M	87.4	32s	851M	86.4	118s	375M	86.8	2.93s	619M	85.8	1.50s	631M
PPI	68.8	402s	849M	-	-	-	98.6	63s	759M	97.2	49s	629M	96.8	26s	631M
Reddit	93.4	998s	4343M	92.6	761s	4429M	96.0	201s	1271M	94.2	44s	621M	94.0	34s	635M
Amazon-670K	83.1	2153s	849M	76.1	548s	1621M	92.7	534s	625M	91.6	54s	601M	91.2	30s	613M
Amazon-3M	-	-	-	-	-	-	88.3	2165s	625M	88.4	203s	601M	88.4	125s	613M

TAMU HPRC cluster: Terra (GPU); Software: Anaconda/3-5.0.0.1

TEXAS A&M UNIVERSITY

Engineering

Thank you for listening.

Paper: https://arxiv.org/abs/2003.13606

Code: https://github.com/Shen-Lab/L2-GCN