Towards Deepening Graph Neural Networks: A GNTK-based Optimization Perspective

Wei Huang[1], Yayong Li[1], Weitao Du[3], Jie Yin[2], Richard Xu[1], Ling Chen[1], Miao Zhang[4]

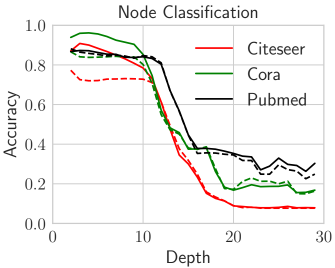
[1]University of Technology Sydney, Australia
[2]The University of Sydney, Australia
[3]Northwestern University, USA
[4]Aalborg University, Denmark

03/2022

UTS

Motivation

- Graph Neural Networks (GNNs) have shown incredible abilities to learn node or graph representations and achieved superior performance on node classification, graph classification and link prediction, etc.
- Most GNNs achieve their best only with a shallow depth, e.g., 2 or 3 layers, and their performance on those tasks would degrade as the number of layers grows.



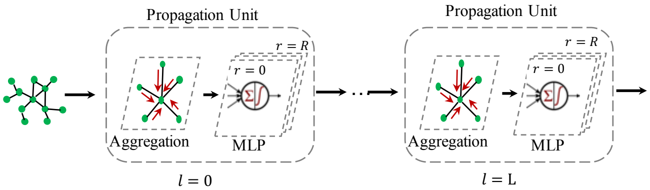
Motivation

- It remains elusive how to theoretically understand why deep GCNs fail to optimize.
- Existing theoretical investigation on GNNs focus mainly on expressivity, which measures the complexity of functions that can be represented by a neural network.
- Research Questions:
  - Can we theoretically characterize the trainability of graph neural networks with respect to depth, thus understanding why deep GCNs fail to generalize?
  - Can we further design an algorithm to facilitate deeper GCNs, benefiting from our theoretical investigation?

UTS

Graph Neural Tangent Kernel

- It has been shown that the training process of an infinitely-wide neural network with gradient descent training can be captured by its neural tangent kernel (NTK).
- We leverage the GNTK techniques of infinitely-wide networks to investigate whether ultra-wide GNNs are trainable in the large depth
- In particular, we aim to characterize the behavior of GNTK matrix $\Theta^{(l)}_{(r)}(G)$, as the depth goes to infinity.

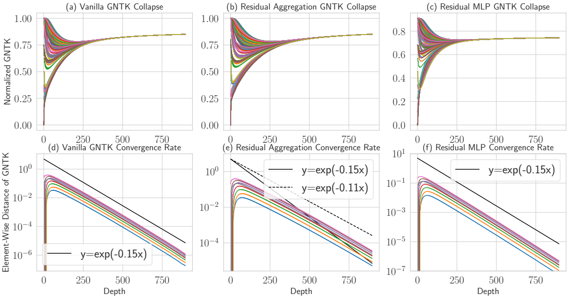


Main Theoretical Results

- The aggregation operation corresponds to a probability transition matrix $\mathcal{A}(G)$ in the GNTK formulation. The corresponding stationary distribution vector is denoted as $\vec{\pi}(G)$.

- There exist constants $0 < \alpha < 1$ and $C > 0$, and constant vectors $\vec{v}, \vec{v}'$ depending on the number of MLP iterations R, such that

$$\left|\Theta_{(r)}^{(l)}(u, u') - \vec{\pi}(G)^T \left(Rl\vec{v} + \vec{v}'\right)\right| \leq C\alpha^l$$



Critical DropEdge

- To better resolve the problem of exponential convergence rate of trainability, we need to look deeper into the root cause of the problem – the transition matrix corresponding to the aggregation operation.

- A necessary condition for matrix $\mathcal{A}(G)$ to be a probability transition matrix is that graph G is connected. Thus, breaking the connectivity condition is a promising way of better solving the exponential decay problem.

- One method is to perform edge sampling guided by the critical percolation theory. Suppose a random graph $\hat{G}$ has n nodes with a constant edge probability p.
  - If $p < p_c$, then almost every random graph is such that its largest component is of size $O(\log n)$;
  - If $p > p_c$, the random graph has a giant component of size $(1 - \alpha_p + o(1))n$, where $\alpha_p < 1$;
  - $p = p_c$, then the maximal size of a component of almost every graph has order $n^{2/3}$.

Experimental Results

Datasets	Methods	4-layer	8-layer	16-layer	32-layer
Cora	GCN	79.8 ± 1.1	73.2 ± 2.7	36.3 ± 13.8	20.1 ± 2.4
	DropEdge	82.2 ± 0.7	82.0 ± 0.9	82.2 ± 0.7	82.1 ± 0.5
	DGN	82.0 ± 0.9	80.2 ± 0.8	77.7 ± 1.0	73.0 ± 0.8
	C-DropEdge	**82.5 ± 0.7**	**82.3 ± 0.6**	**82.4 ± 0.8**	**82.6 ± 0.9**
Citeseer	GCN	61.2 ± 3.0	50.2 ± 5.7	30.8 ± 2.2	21.7 ± 3.0
	DropEdge	70.2 ± 1.0	70.8 ± 1.1	70.7 ± 1.0	70.2 ± 0.8
	DGN	69.0 ± 0.9	66.5 ± 1.1	62.9 ± 1.2	63.2 ± 0.9
	C-DropEdge	**70.8 ± 0.6**	**70.9 ± 0.9**	**71.0 ± 1.0**	**70.7 ± 0.9**
Pubmed	GCN	77.4 ± 0.7	57.2 ± 8.4	39.5 ± 10.3	36.3 ± 8.4
	Dropedge	77.6 ± 1.4	77.3 ± 1.3	76.7 ± 1.3	77.2 ± 1.3
	DGN	**78.2 ± 1.0**	77.8 ± 1.2	77.2 ± 1.3	77.0 ± 1.1
	C-DropEdge	78.0 ± 0.4	**77.9 ± 1.0**	**77.2 ± 1.0**	**77.8 ± 1.0**
Physics	GCN	90.2 ± 0.9	83.5 ± 2.2	41.6 ± 6.2	28.8 ± 9.4
	Dropedge	91.6 ± 0.8	91.5 ± 0.7	91.2 ± 0.5	91.3 ± 0.8
	DGN	**92.2 ± 1.0**	86.4 ± 0.7	83.4 ± 0.6	83.2 ± 0.8
	C-DropEdge	91.9 ± 0.7	**91.7 ± 0.6**	**92.0 ± 0.4**	**91.6 ± 0.6**