

DON'T STACK LAYERS IN GRAPH NEURAL NETWORKS, WIRE THEM RANDOMLY

SUPPLEMENTARY MATERIAL

Anonymous authors

Paper under double-blind review

1 PROOF LEMMA 3.1

Let us first consider the number of paths of length l from node k to the sink. We define the path length as the number of nodes in the path. In a randomly wired network with n architecture nodes, we have that the first node of all the paths is node k and the last one is node n (i.e., the sink node). Therefore, the minimum path length is 2. If $l \geq 2$, the number of all possible paths of length l between node k and the sink is $\binom{n-k-1}{l-2}$. Since in a path of length l there are $l-1$ edges and each edge has probability p of being generated by the Erdős-Renyi model, each one of the paths of length l has probability p^{l-1} of being present in the network. Thus, the expected number of paths with length l between node k and the sink is $\mathbb{E}[N_l^{(k)}] = \binom{n-k-1}{l-2} p^{l-1}$. If we set $k = 1$, we obtain the average number of paths of length l from source to sink $\mathbb{E}[N_l] = \binom{n-2}{l-2} p^{l-1}$. We can now compute the average total number of paths $\mathbb{E}[N^{(k)}]$ as follows

$$\mathbb{E}[N^{(k)}] = \sum_{l=2}^{n-k+1} \binom{n-k-1}{l-2} p^{l-1} = \sum_{\tilde{l}=0}^{\tilde{n}} \binom{\tilde{n}}{\tilde{l}} p^{\tilde{l}+1} = p \sum_{\tilde{l}=0}^{\tilde{n}} \binom{\tilde{n}}{\tilde{l}} p^{\tilde{l}} = p(1+p)^{\tilde{n}} = p(1+p)^{n-k-1},$$

where $\tilde{n} = n - k - 1$, $\tilde{l} = l - 2$ and the fourth equality follows from the binomial theorem. If we set $k = 1$, we obtain the average total number of paths from source to sink $\mathbb{E}[N_p] = p(1+p)^{n-2}$.

2 PROOF LEMMA 3.3

From Lemma 3.1, we can compute the average length of the paths from node k to the sink as follows

$$\mathbb{E}[l^{(k)}] = \sum_{l=2}^{n-k+1} l \mathbb{E} \left[\frac{N_l^{(k)}}{N^{(k)}} \right] \approx \frac{\sum_{l=2}^{n-k+1} l \mathbb{E}[N_l^{(k)}]}{\mathbb{E}[N^{(k)}]} = \frac{\sum_{l=2}^{n-k+1} \binom{n-k-1}{l-2} p^{l-1} l}{p(1+p)^{n-k-1}}, \quad (1)$$

where we have neglected the higher order terms (Elandt-Johnson & Johnson, 1980). The numerator in (1) can be computed as follows

$$\begin{aligned} \sum_{l=2}^{n-k+1} \binom{n-k-1}{l-2} p^{l-1} l &= \sum_{\tilde{l}=0}^{\tilde{n}} \binom{\tilde{n}}{\tilde{l}} p^{\tilde{l}+1} (\tilde{l}+2) = \sum_{\tilde{l}=0}^{\tilde{n}} \binom{\tilde{n}}{\tilde{l}} p^{\tilde{l}+1} \tilde{l} + 2 \sum_{\tilde{l}=0}^{\tilde{n}} \binom{\tilde{n}}{\tilde{l}} p^{\tilde{l}+1} \\ &= p^2 \sum_{\tilde{l}=0}^{\tilde{n}} \binom{\tilde{n}}{\tilde{l}} p^{\tilde{l}-1} \tilde{l} + 2p \sum_{\tilde{l}=0}^{\tilde{n}} \binom{\tilde{n}}{\tilde{l}} p^{\tilde{l}} = p^2 \tilde{n}(1+p)^{\tilde{n}-1} + 2p(1+p)^{\tilde{n}} \\ &= p^2(n-k-1)(1+p)^{n-k-2} + 2p(1+p)^{n-k-1}, \end{aligned}$$

where $\tilde{n} = n - k - 1$, $\tilde{l} = l - 2$ and the fourth equality is obtained differentiating with respect to p the binomial theorem. Then, we obtain

$$\mathbb{E}[l^{(k)}] = \frac{p}{1+p} (n-k-1) + 2.$$

If we consider $k = 1$, i.e. the sink, we obtain $\mathbb{E}[l] = \frac{p}{1+p} (n-2) + 2$.

3 PROOF LEMMA 3.4

From Lemma 3.3, we can compute the average length of the paths going through the edge $\{i, j\}$ as follows

$$\mathbb{E}[l_{ij}] = \mathbb{E}[l^{n-i+1} + l^j] \approx \frac{p}{1+p}(n - (j - i) - 3) + 4.$$

4 MONTECARLO DROPPATH REGULARIZATION

Let us consider two paths p_1 and p_2 with at least one common edge, we can compute the covariance between these two paths as follows:

$$\begin{aligned} \text{Cov}(\lambda_{p_1}, \lambda_{p_2}) &= \mathbb{E}[\lambda_{p_1} \lambda_{p_2}] - \mathbb{E}[\lambda_{p_1}] \mathbb{E}[\lambda_{p_2}] \\ &= \prod_{\{i,j\} \in \mathcal{E}^{p_1}} \omega_{ij} \prod_{\{i,j\} \in \mathcal{E}^{p_2}} \omega_{ij} \mathbb{E} \left[\prod_{\{i,j\} \in \mathcal{I}(p_1, p_2)} z_{ij}^2 \prod_{\{i,j\} \in \mathcal{D}(p_1, p_2)} z_{ij} \right] \\ &\quad - \prod_{\{i,j\} \in \mathcal{E}^{p_1}} \omega_{ij} \prod_{\{i,j\} \in \mathcal{E}^{p_2}} \omega_{ij} \mathbb{E} \left[\prod_{\{i,j\} \in \mathcal{E}^{p_1}} z_{ij} \right] \mathbb{E} \left[\prod_{\{i,j\} \in \mathcal{E}^{p_2}} z_{ij} \right] = 0, \end{aligned}$$

where $\mathcal{I}(p_1, p_2) = \mathcal{E}^{p_1} \cap \mathcal{E}^{p_2}$, $\mathcal{D}(p_1, p_2) = (\mathcal{E}^{p_1} \cup \mathcal{E}^{p_2}) - (\mathcal{E}^{p_1} \cap \mathcal{E}^{p_2})$, $\lambda_{p_1} = \prod_{\{i,j\} \in p_1} z_{ij} \omega_{ij}$, $\lambda_{p_2} = \prod_{\{i,j\} \in p_2} z_{ij} \omega_{ij}$, $z_{ij} \sim \text{Bernoulli}(1 - p_{\text{drop}})$, and we have assumed all ω_{ij} deterministic.

5 EXPERIMENTAL DETAILS

Table 1 shows the chosen hyperparameters for the various experiments. The number of parameters refers to one realization of the random architecture graph. The values of the hyperparameters that do not depend on the random architecture are the same for the baseline architectures.

Training has been performed on a workstation with an AMD Ryzen 3700X CPU, 64 GB of RAM and a Titan RTX GPU. Training times can vary run-to-run due to the convergence criterion of no improvement over the validation loss after reaching the minimum learning rate. Training the random architecture is roughly 5% slower than training the baseline.

6 EXPERIMENTAL RECEPTIVE FIELD RADIUS

Table 2 shows the average radius of the receptive field obtained in the experiments. In particular, we compare its value at initialization and the value after training. We remark that the average receptive field radius is computed as:

$$\bar{\rho} = \frac{\sum_{l=2}^n l \rho_l}{\sum_{l=2}^n \rho_l}$$

We can notice how the CLUSTER experiment clearly favours larger receptive fields, already optimizing a larger radius even without the sequential path. We can also see how the sequential path promotes a further increases the optimal radius. We remark that the baseline ResNets have a fixed radius equal to 17.

7 ADDITIONAL RESULTS

Table 3 reports the results on the ENZYMES dataset, a historically relevant dataset. As it can be noticed and reported in recent literature, this dataset is inadequate to assess relative performance of graph neural network architectures. The large standard deviations due to small dataset size and train/test splits do not allow to infer any meaningful ranking of the proposed method with respect to the ResNet baseline.

Finally, Table 4 reports the results for $L = 4$ layers, used in the paper to compute the average gains when capacity is increased.

Table 1: Experimental setting

Dataset	Model	Hyperparameters										others		
		batch size (L=8,16)	batch size (L=32)	start lr	end lr	hidden feat	out feat	p	p_drop	sequential	#params (L=8)			#params (L=16)
ZINC	R-GCN	128	128	1.00E-03	1.00E-05	145	145	0.6	0	true	188931	360661	704235	readout:mean
	R-GIN	128	128	1.00E-03	1.00E-05	110	110	0.6	0	true	204745	406330	809632	nnlpGIN:2; learnpsGIN:True; neighborGIN:sum; readout:sum
	R-GatedGCN	128	128	1.00E-03	1.00E-05	70	70	0.6	0	true	206427	407711	810071	edgefeat:False; readout:mean
	R-GraphSage	128	128	1.00E-03	1.00E-05	108	108	0.6	0	true	199655	388924	767565	sageaggregator:meanpool; readout:mean
CLUSTER	R-GAT	128	128	1.00E-03	1.00E-05	144	144	0.6	0	true	183444	358193	699399	nheads:8; readout:mean
	R-GCN	128	48	1.00E-03	1.00E-05	146	146	0.6	0.01	true	188683	362787	711065	readout:mean
	R-GIN	128	48	1.00E-03	1.00E-05	110	110	0.6	0.01	true	207430	413455	825635	nnlpGIN:2; learnpsGIN:True; neighborGIN:sum; readout:sum
	R-GatedGCN	128	48	1.00E-03	1.00E-05	70	70	0.6	0.01	true	204899	406043	808573	edgefeat:False; readout:mean
CIFAR10	R-GraphSage	128	48	1.00E-03	1.00E-05	106	106	0.6	0.01	true	190315	372687	737557	sageaggregator:meanpool; readout:mean
	R-GAT	128	48	1.00E-03	1.00E-05	19	152	0.6	0.01	true	205564	395317	774934	nheads:8; readout:mean
	R-GCN	128	128	1.00E-03	1.00E-05	146	146	0.6	0.01	false	188686	362769	711058	readout:mean
	R-GIN	128	128	1.00E-03	1.00E-05	110	110	0.6	0.01	false	211316	420893	840177	nnlpGIN:2; learnpsGIN:True; neighborGIN:sum; readout:sum
ENZYMES	R-GatedGCN	128	128	1.00E-03	1.00E-05	70	70	0.6	0.01	false	204927	406029	808553	edgefeat:False; readout:mean
	R-GraphSage	128	128	1.00E-03	1.00E-05	108	108	0.6	0.01	false	197540	387651	765442	sageaggregator:meanpool; readout:mean
	R-GAT	128	128	1.00E-03	1.00E-05	19	152	0.6	0.01	false	205579	395304	774929	nheads:8; readout:mean
	R-GCN	128	128	1.00E-03	1.00E-05	128	128	0.6	0	false	147122	281318	537923	readout:mean
ENZYMES	R-GIN	128	128	1.00E-03	1.00E-05	96	96	0.6	0	false	160688	319010	633316	nnlpGIN:2; learnpsGIN:True; neighborGIN:sum; readout:sum
	R-GatedGCN	128	128	1.00E-03	1.00E-05	64	64	0.6	0	false	173614	342166	679512	edgefeat:False; readout:mean
	R-GraphSage	128	128	1.00E-03	1.00E-05	96	96	0.6	0	false	157578	307390	607124	sageaggregator:meanpool; readout:mean
	R-GAT	128	128	1.00E-03	1.00E-05	16	128	0.6	0	false	148146	283366	553916	nheads:8; readout:mean

Table 2: Experimental average receptive field radius

	R-GatedGCN		+ Sequential		+ DropPath		+ Sequential + DropPath	
	Init	Optimized	Init	Optimized	Init	Optimized	Init	Optimized
ZINC	9.26	9.80	12.57	13.28	-	-	-	-
CLUSTER	9.26	10.44	12.57	14.87	-	-	12.58	14.89
CIFAR10	9.26	9.93	-	-	9.21	9.94	-	-

Table 3: ENZYMES Accuracy.

	$L = 8$	$L = 16$	$L = 32$
GCN	63.46 ± 4.81	64.75 ± 6.40	65.50 ± 6.10
R-GCN	66.67 ± 7.32	66.71 ± 7.53	68.17 ± 6.34
GIN	65.04 ± 6.35	63.77 ± 5.96	61.17 ± 6.06
R-GIN	67.88 ± 6.27	63.75 ± 5.73	63.00 ± 8.46
GatedGCN	68.04 ± 6.58	68.67 ± 6.70	69.33 ± 7.64
R-GatedGCN	68.03 ± 5.59	67.33 ± 5.64	68.17 ± 4.96
GraphSage	68.04 ± 5.47	69.08 ± 5.17	68.50 ± 5.84
R-GraphSage	68.33 ± 4.91	69.13 ± 5.32	66.17 ± 3.42
GAT	69.63 ± 5.21	68.83 ± 4.66	68.67 ± 6.66
R-GAT	66.67 ± 4.33	65.33 ± 3.06	64.00 ± 2.91

Table 4: $L = 4$ performance.

	$L = 4$		
	ZINC	CLUSTER	CIFAR
GCN	0.469 ± 0.002	48.68 ± 3.47	54.28 ± 0.35
R-GCN	0.509 ± 0.015	50.82 ± 5.37	55.31 ± 0.25
GIN	0.418 ± 0.002	42.35 ± 6.10	48.64 ± 2.29
R-GIN	0.411 ± 0.009	45.04 ± 3.95	46.68 ± 6.14
GatedGCN	0.368 ± 0.007	48.96 ± 6.00	69.26 ± 0.36
R-GatedGCN	0.364 ± 0.007	51.18 ± 8.27	68.55 ± 0.03
GraphSage	0.428 ± 0.007	47.17 ± 6.78	66.14 ± 0.21
R-GraphSage	0.429 ± 0.010	47.86 ± 5.63	65.02 ± 0.47
GAT	0.464 ± 0.005	52.88 ± 1.74	65.72 ± 0.54
R-GAT	0.502 ± 0.019	53.57 ± 1.67	65.32 ± 0.79

REFERENCES

Regina C Elandt-Johnson and Norman L Johnson. *Survival models and data analysis*, volume 110. John Wiley & Sons, 1980.