Benchmarking Graph Neural Networks

Vijay Prakash Dwivedi Chaitanya K. Joshi Thomas Laurent Yoshua Bengio Xavier Bresson

20 March 2020

Presenter: Sanchit Sinha

https://qdata.github.io/deep2Read/

Motivation

- Need for Benchmarking:
  - GNNs are thought of as much more generalisable
  - Graph data holds a lot more information
  - Not well explored
- Problems with Benchmarking:
  - **Datasets**: The datasets which exist are not representative/challenging
  - **Performance**: Some GNN models don't perform as well as non-GNNs
  - **Settings**: Experimental settings are not yet standardized

Background

- Graph Neural Networks (only a few used in this paper):
  - **Basic**: $$\hat{h}_i^{\ell+1} = \frac{1}{\deg_i} \sum_{j \in \mathcal{N}_i} h_j^{\ell}, \qquad h_i^{\ell+1} = \sigma(U^{\ell}\, \hat{h}_i^{\ell+1})$$
  - **Mean GraphStage**: $$\hat{h}_i^{\ell+1} = \text{Concat}\Big(h_i^{\ell}\ ,\ \frac{1}{\deg_i} \sum_{j \in \mathcal{N}_i} h_j^{\ell}\Big)$$
  - **GIN**: $$\hat{h}_i^{\ell+1} = (1+\epsilon)\, h_i^{\ell} + \sum_{j \in \mathcal{N}_i} h_j^{\ell},$$ $$h_i^{\ell+1} = \sigma\Big(U^{\ell}\, \sigma\big(\,\text{BN}(\,V^{\ell}\hat{h}_i^{\ell+1}\,)\,\big)\Big),$$
  - **Anisotropic:** $$\hat{h}_i^{\ell+1} = w_i^{\ell} h_i^{\ell} + \sum_{j \in \mathcal{N}_i} w_{ij}^{\ell} h_j^{\ell},$$

Claim / Target Task

- Propose a benchmark, with plug and play methods - model and datasets
- Create new datasets by converting well known datasets into graphs
- Proposed/tested building blocks of GNN
- Compare performance

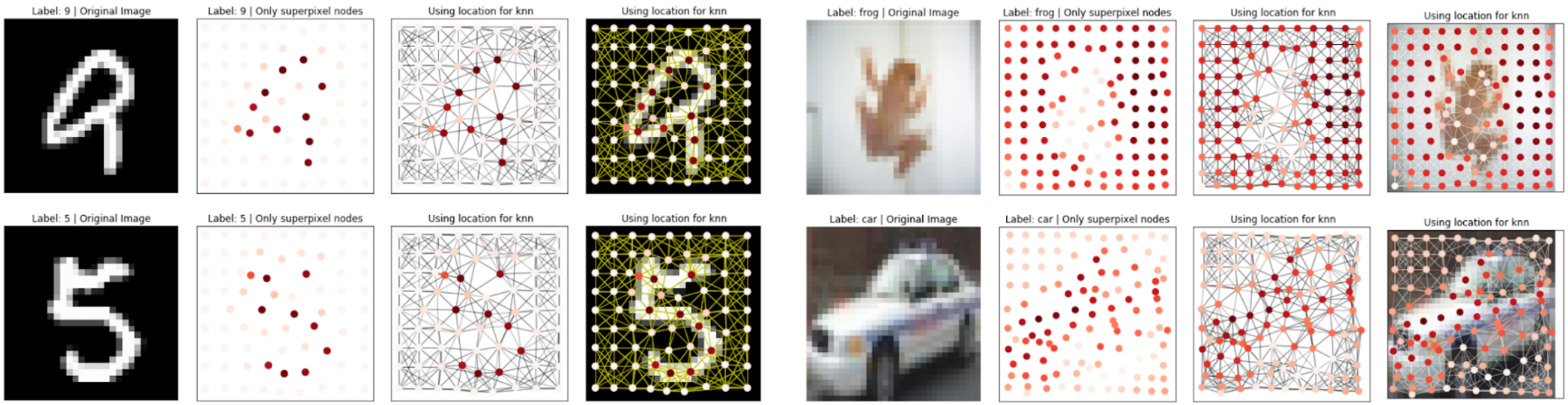
Data Summary

- Old Graph Datasets:
  - CORA
  - TU
- Converted Datasets:

Table 1. Summary statistics of proposed benchmark datasets.

Domain/Construction	Dataset	# graphs	# nodes
Computer Vision/ Graphs constructed with super-pixels	MNIST CIFAR10	70K 60K	40-75 85-150
Chemistry/ Real-world molecular graphs	ZINC	12K	9-37
Artificial/ Graphs generated from Stochastic Block Model	PATTERN CLUSTER	14K 12K	50-180 40-190
Artificial/ Graphs generated from uniform distribution	TSP	12K	50-500

Proposed Solution - Superpixel



(a) MNIST (b) CIFAR10

Domain/Construction	Dataset	# graphs	# nodes
Computer Vision/ Graphs constructed with super-pixels	MNIST CIFAR10	70K 60K	40-75 85-150
Chemistry/ Real-world molecular graphs	ZINC	12K	9-37
Artificial/ Graphs generated from Stochastic Block Model	PATTERN CLUSTER	14K 12K	50-180 40-190
Artificial/ Graphs generated from uniform distribution	TSP	12K	50-500

Experimental Results

Table 3. Performance on the standard test sets of MNIST and CIFAR10 (higher is better). Results are averaged over 4 runs with 4 different seeds. **Red**: the best model, Violet: good models. **Bold** indicates the best model between residual and non-residual connections (both models are bold if they perform equally).

Dataset	Model	#Param	Residual		No Residual	
			Acc	Epoch/Total	Acc	Epoch/Total
MNIST	MLP	104044	*not used*		94.46±0.28	21.82s/1.02hr
	MLP (Gated)	105717	*not used*		95.18±0.18	22.43s/0.73hr
	GCN	101365	**89.99±0.15**	78.25s/1.81hr	**89.05±0.21**	79.18s/1.76hr
	GraphSage	102691	**97.09±0.02**	75.57s/1.36hr	**97.20±0.17**	76.80s/1.42hr
	GIN	105434	**93.91±0.63**	34.30s/0.73hr	**93.96±1.30**	34.61s/0.74hr
	DiffPool	106538	**95.02±0.42**	170.55s/4.26hr	**94.66±0.48**	171.38s/4.45hr
	GAT	110400	**95.62±0.13**	375.71s/6.35hr	**95.56±0.16**	377.06s/6.35hr
	MoNet	104049	**90.36±0.47**	581.86s/15.31hr	**89.73±0.48**	567.12s/12.05hr
	GatedGCN	104217	**97.37±0.06**	128.39s/2.01hr	**97.36±0.12**	127.15s/2.13hr
	GatedGCN-E*	104217	**97.24±0.10**	135.10s/2.25hr	**97.47±0.13**	127.86s/2.15hr
CIFAR10	MLP	104044	*not used*		56.01±0.90	21.82s/1.02hr
	MLP (Gated)	106017	*not used*		56.78±0.12	27.85s/0.68hr
	GCN	101657	**54.46±0.10**	100.91s/2.73hr	51.64±0.45	100.30s/2.44hr
	GraphSage	102907	**65.93±0.30**	96.67s/1.88hr	**66.08±0.24**	96.00s/1.79hr
	GIN	105654	**53.28±3.70**	45.29s/1.24hr	47.66±0.47	44.30s/0.93hr
	DiffPool	108042	**57.99±0.45**	298.06s/10.17hr	56.84±0.37	299.64s/10.42hr
	GAT	110704	**65.40±0.38**	389.40s/7.32hr	**65.48±0.33**	386.14s/7.75hr
	MoNet	104229	**53.42±0.43**	836.32s/22.45hr	50.99±0.17	869.90s/21.79hr
	GatedGCN	104357	**69.19±0.28**	146.80s/2.48hr	**68.92±0.38**	145.14s/2.49hr
	GatedGCN-E*	104357	68.64±0.60	158.80s/2.74hr	**69.37±0.48**	145.66s/2.43hr

*GatedGCN-E uses the graph adjacency weight as edge feature.

Experimental Results

Table 5. Performance on the standard test sets of PATTERN and CLUSTER SBM graphs (higher is better). Results are averaged over 4 runs with 4 different seeds. **Red**: the best model and Violet: good models. **Bold** indicates the best model between residual and non-residual connections.

Dataset	Model	#Param	Residual		No Residual	
			Acc	Epoch/Total	Acc	Epoch/Total
PATTERN	MLP	105263	*not used*		50.13±0.00	8.68s/0.10hr
	MLP (Gated)	103629	*not used*		50.13±0.00	9.78s/0.12hr
	GCN	100923	**74.36±1.59**	97.37s/2.06hr	55.22±0.17	97.46s/2.30hr
	GraphSage	98607	78.20±3.06	79.19s/2.57hr	**81.25±3.84**	79.43s/2.14hr
	GIN	100884	96.98±2.18	14.12s/0.32hr	**98.25±0.38**	14.11s/0.37hr
	GAT	109936	**90.72±2.04**	229.76s/5.73hr	88.91±4.48	229.65s/8.78hr
	MoNet	103775	95.52±3.74	879.87s/21.80hr	**97.89±0.89**	870.05s/24.86hr
	GatedGCN	104003	95.05±2.80	115.55s/2.46hr	**97.24±1.19**	115.03s/2.59hr
CLUSTER	MLP	106015	*not used*		20.97±0.01	6.54s/0.08hr
	MLP (Gated)	104305	*not used*		20.97±0.01	7.37s/0.09hr
	GCN	101655	**47.82±4.91**	66.58s/1.26hr	34.85±0.65	66.81s/1.21hr
	GraphSage	99139	44.89±3.70	54.53s/1.05hr	**53.90±4.12**	54.40s/1.19hr
	GIN	103544	49.64±2.09	11.60s/0.27hr	**52.54±1.03**	11.57s/0.27hr
	GAT	110700	49.08±6.47	158.23s/4.08hr	**54.12±1.21**	158.46s/4.53hr
	MoNet	104227	**45.95±3.39**	635.77s/15.32hr	39.48±2.21	600.04s/11.18hr
	GatedGCN	104355	**54.20±3.58**	81.39s/2.26hr	50.18±3.03	80.66s/2.07hr

Table 6. Performance on TSP test set graphs with and without residual connections (higher is better). Results are averaged over 2 runs with 2 different seeds. **Red**: the best model and Violet: good models. **Bold** indicates the best model between residual and non-residual connections (both models are bold if they perform equally).

Model	#Param	Residual		No Residual	
		F1	Epoch/Total	F1	Epoch/Total
k-NN Heuristic	k=2	F1: 0.693			
MLP	94394	*not used*		0.548±0.003	53.92s/2.85hr
MLP (Gated)	115274	*not used*		0.548±0.001	54.39s/2.44hr
GCN	108738	**0.627±0.003**	163.36s/11.26hr	0.547±0.003	164.41s/10.28hr
GraphSage	98450	**0.663±0.003**	145.75s/16.05hr	**0.657±0.002**	147.22s/14.33hr
GIN	118574	**0.655±0.001**	73.09s/5.44hr	**0.657±0.001**	74.71s/5.60h
GAT	109250	**0.669±0.001**	360.92s/30.38hr	0.567±0.003	360.74s/20.55hr
MoNet	94274	**0.637±0.010**	1433.97s/41.69hr	0.569±0.002	1472.65s/42.44hr
GatedGCN	94946	**0.794±0.004**	203.28s/15.47hr	**0.791±0.003**	202.12s/15.20hr
GatedGCN-E*	94946	**0.802±0.001**	201.40s/15.19hr	0.794±0.003	201.32s/15.05hr

*GatedGCN-E uses the pairwise distance as edge feature.

Generating correlated features (for images) - ?

1 Simulation Data Generation

Assuming we have p features. $\Delta, R_I \in \mathbb{R}^{p\times p}$, Δ and R_I are both Erdos Renyi graphs, with probability p_d and p_i respectively.

In a multivariate normal distribution, the key property of the precision matrix(inverse of covariance) is that its zeros indicate conditional independence. The values indicate partial correlation of two variables. Specifically: Ω_{ij}=0 if and only if X_i and X_j are conditionally independent given all other coordinates of X. We generate data from two classes A and B using the following equations:

$$\Omega_A = \Delta + R_I \tag{1}$$

$$\Omega_B = R_I \tag{2}$$

$$X_A \sim N(0, \Omega_A^{-1}) \tag{3}$$

$$X_B \sim N(0, \Omega_B^{-1}) \tag{4}$$

Main takeaway: Generating the covariance matrix

In the code a random E-R graph is made and its adjacency matrix' inverse is taken as covariance matrix to the normal distributions