HOW POWERFUL ARE GRAPH NEURAL NETWORKS?

ICLR 2019

Keyulu Xu
MIT

Weihua Hu
Stanford University

Jure Leskovec
Stanford University

Stefanie Jegelka
MIT

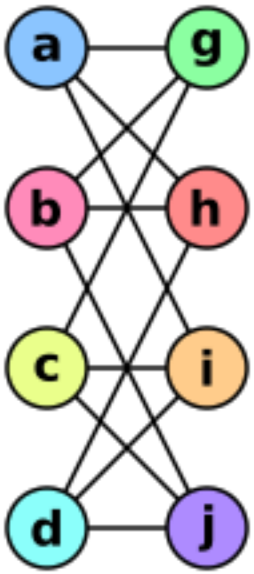
2025.07.29
DMAIS Master Course
박세준

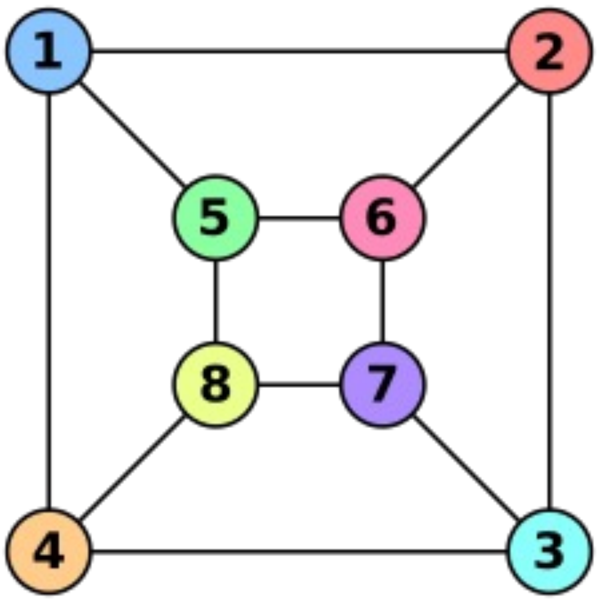
INDEX

- Background
- Motivation
- GIN
- Experiments
- Conclusions

BACKGROUND

- **Graph Isomorphism**
  - Graphs have the same structure, but they may be represented differently.
  - GNN that cannot distinguish graph isomorphism means that it lacks expressiveness





$f(a) = 1$
$f(b) = 6$
$f(c) = 8$
$f(d) = 3$
$f(g) = 5$
$f(h) = 2$
$f(i) = 4$
$f(j) = 7$

BACKGROUND

- **Weisfeiler-Lehman Graph Isomorphism Test**
  - To test two graphs are isomorphism
  - Iteratively aggregates the labels of nodes and their neighborhoods
  - Hashes the aggregated labels into unique new labels

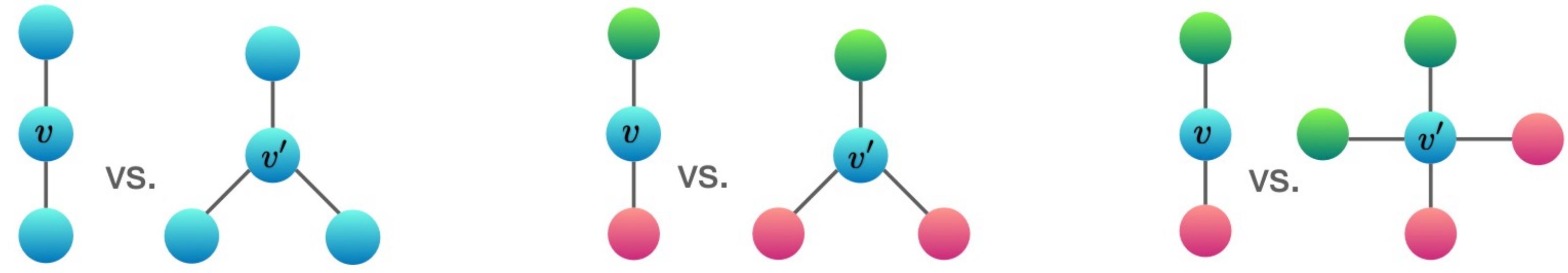
- **WL Subtree Kernel**
  - Based on WL test
  - Measures the similarity between graph

BACKGROUND

- **Graph Neural Network**
  - Effective framework for representation learning of graphs
  - GNN follow a neighborhood aggregation (or message passing) scheme
  - The design of new GNNs is mostly based on empirical intuition, heuristics, and experimental
  - Furthermore popular GNN variants cannot learn to distinguish certain simple graph structure
    - GCN, GraphSAGE…

BACKGROUND

- **Structures That Confuse Mean And Max-Pooling**
  - GraphSAGE readout function
    - $a_v^{(k)} = MAX(\{RELU(W * h_u^{(k-1)}), \forall u \in N(v)\})$
  - GCN readout function
    - $h_v^{(k)} = RELU(W * MEAN\{h_u^{(k-1)}, \forall u \in N(v) \cup \{v\}\})$
  - Graph representation
    - $h_G = READOUT(\{h_v^{(k)} | v \in G\})$

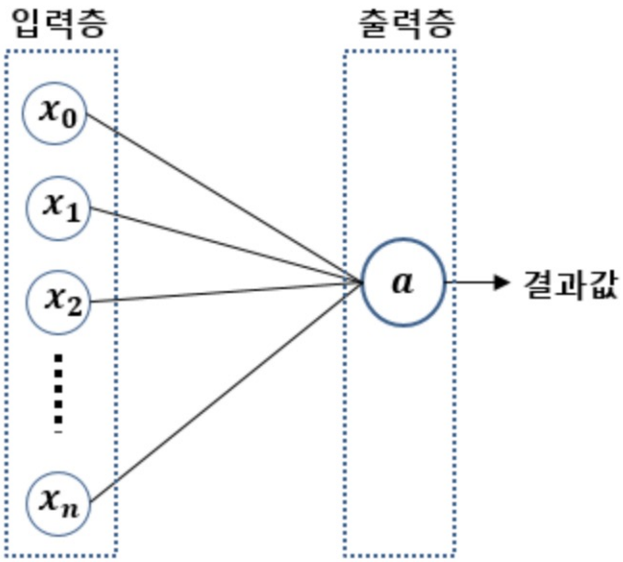


(a) Mean and Max both fail (b) Max fails (c) Mean and Max both fail

BACKGROUND

- **1-Layer Perceptrons Are Not Sufficient**

Lemma 7. *There exist finite multisets* $X_1 \neq X_2$ *so that for any linear mapping* W, $\sum_{x \in X_1} \mathrm{ReLU}(Wx) = \sum_{x \in X_2} \mathrm{ReLU}(Wx)$.



MOTIVATION

- **Make Powerful GNN**
  - Effective framework for representation learning of graphs
  - Have reprenstation power like WL test

TERM

- **Multiset**
  - Generalized concept of a set that allows multiple instances for its elements
  - Multiset is a 2-tuple $X = (S, m)$where S is the underlying set of X
  - $m: S \rightarrow \mathrm{N} \geq 1$ gives the multiplicity of the elements

- **Injective**
  - Map two different neighborhoods to different representation.
    - One to one function

GIN

- **Building Powerful Graph Neural Networks**

Lemma 2. *Let G_1 and G_2 be any two non-isomorphic graphs. If a graph neural network $\mathcal{A} : \mathcal{G} \to \mathbb{R}^d$ maps G_1 and G_2 to different embeddings, the Weisfeiler-Lehman graph isomorphism test also decides G_1 and G_2 are not isomorphic.*

Theorem 3. *Let $\mathcal{A} : \mathcal{G} \to \mathbb{R}^d$ be a GNN. With a sufficient number of GNN layers, $\mathcal{A}$ maps any graphs G_1 and G_2 that the Weisfeiler-Lehman test of isomorphism decides as non-isomorphic, to different embeddings if the following conditions hold:*

a) $\mathcal{A}$ aggregates and updates node features iteratively with

$$h_v^{(k)} = \phi\left(h_v^{(k-1)}, f\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right)\right),$$

where the functions f, which operates on multisets, and ϕ are injective.

b) $\mathcal{A}$'s graph-level readout, which operates on the multiset of node features $\left\{h_v^{(k)}\right\}$, is injective.

GIN

- **Building Powerful Graph Neural Networks**

Lemma 4. *Assume the input feature space $\mathcal{X}$ is countable. Let $g^{(k)}$ be the function parameterized by a GNN's k-th layer for $k = 1, \ldots, L$, where $g^{(1)}$ is defined on multisets $X \subset \mathcal{X}$ of bounded size. The range of $g^{(k)}$, i.e., the space of node hidden features $h_v^{(k)}$, is also countable for all $k = 1, \ldots, L$.*

- **GNN Difference with WL Test**
  - WL test based on one hot encoding
  - WL test is good at distinguishing similar graphs but can't reflect the similarity of structures
  - GNN can map similar structure graphs to similar embedding space through learning

GIN

- **Graph Isomorphism Network**

Lemma 5. *Assume $\mathcal{X}$ is countable. There exists a function $f : \mathcal{X} \to \mathbb{R}^n$ so that $h(X) = \sum_{x \in X} f(x)$ is unique for each multiset $X \subset \mathcal{X}$ of bounded size. Moreover, any multiset function g can be decomposed as $g(X) = \phi\left(\sum_{x \in X} f(x)\right)$ for some function ϕ.*

Corollary 6. *Assume $\mathcal{X}$ is countable. There exists a function $f : \mathcal{X} \to \mathbb{R}^n$ so that for infinitely many choices of ϵ, including all irrational numbers, $h(c, X) = (1+\epsilon) \cdot f(c) + \sum_{x \in X} f(x)$ is unique for each pair (c, X), where $c \in \mathcal{X}$ and $X \subset \mathcal{X}$ is a multiset of bounded size. Moreover, any function g over such pairs can be decomposed as $g(c, X) = \varphi\left((1+\epsilon) \cdot f(c) + \sum_{x \in X} f(x)\right)$ for some function φ.*

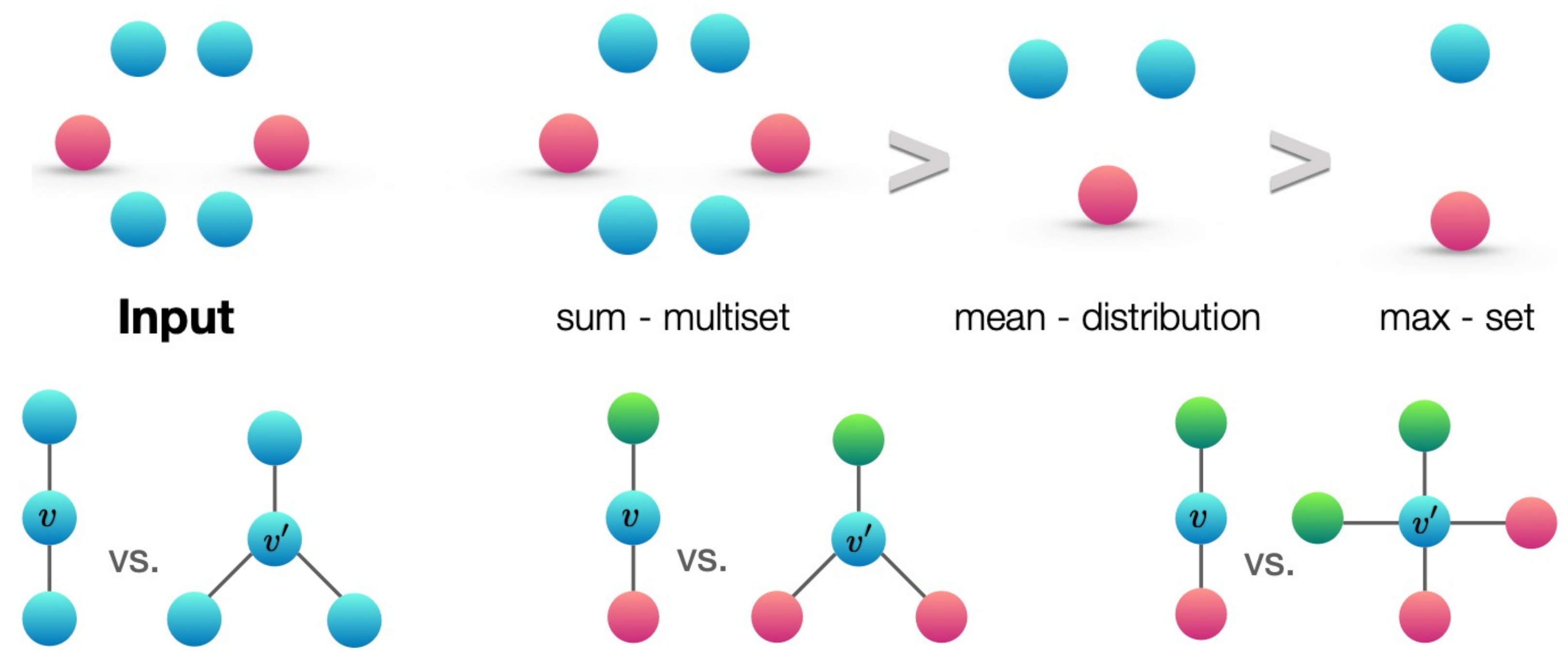
  - $h_v^{(k)} = \mathrm{MLP}^{(\mathrm{k})}\left(\left(1 + \epsilon^{(k)}\right) * h_v^{(k-1)} + \sum_{u \in N(v)} h_u(k-1)\right)$

GIN

- **Graph Level Readout Of GIN**
  - Just as node classification is important, graph classification is also very important
  - Early iterations may sometimes have better generalization performance
  - GIN combines embeddings from all layers for use
    - $h_G = \mathrm{CONCAT}(\mathrm{READOUT}(\{h_v^{(k)} | v \in G\}) | k = 0,1,\ldots,K)$
  - If GIN replaces READOUT with summing all node features from the same iterations
    - Do not need an extra MLP before summation

COMPARE WITH OTHERS

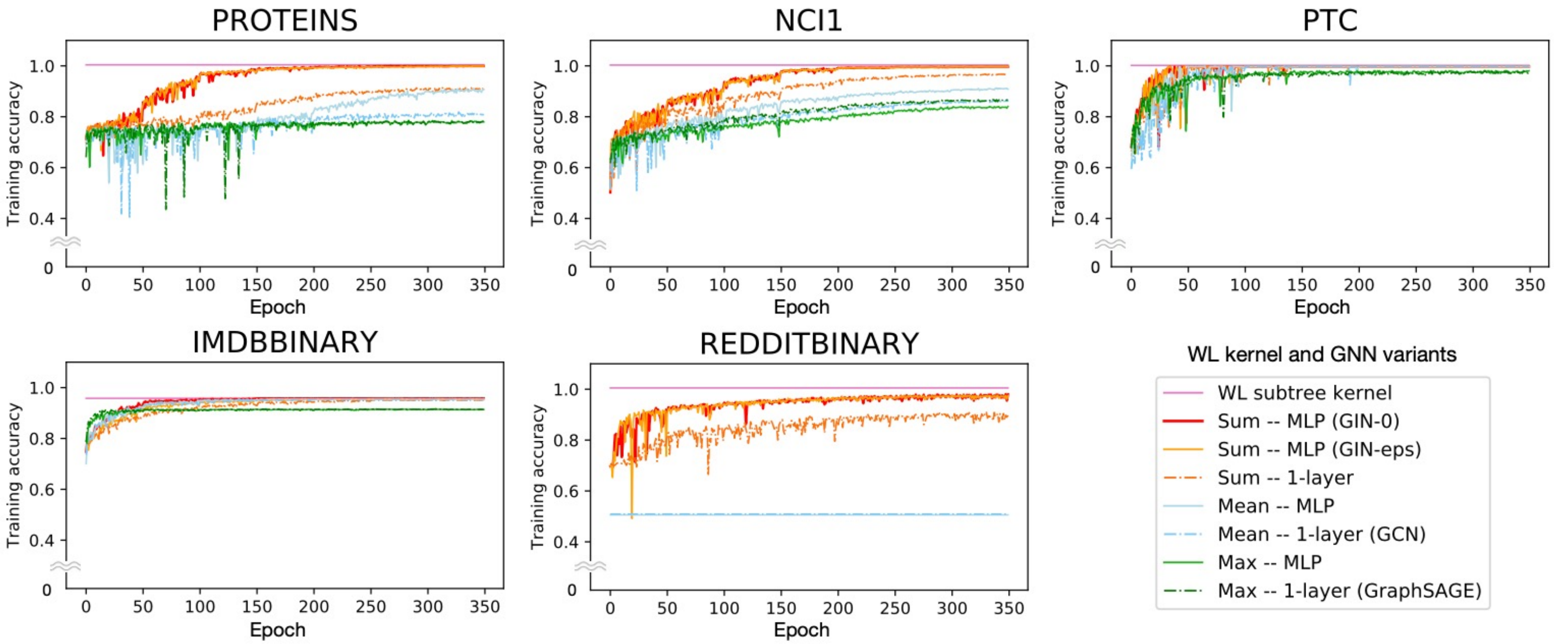
- **Structures That Confuse Mean And Max-Pooling**



(a) Mean and Max both fail (b) Max fails (c) Mean and Max both fail

EXPERIMENTS

- **Training Set Performance.**



EXPERIMENTS

- **Test Set Performance.**

		IMDB-B	IMDB-M	RDT-B	RDT-M5K	COLLAB	MUTAG	PROTEINS	PTC	NCI1
Datasets	Datasets	IMDB-B	IMDB-M	RDT-B	RDT-M5K	COLLAB	MUTAG	PROTEINS	PTC	NCI1
	# graphs	1000	1500	2000	5000	5000	188	1113	344	4110
	# classes	2	3	2	5	3	2	2	2	2
	Avg # nodes	19.8	13.0	429.6	508.5	74.5	17.9	39.1	25.5	29.8
Baselines	WL subtree	73.8 ± 3.9	50.9 ± 3.8	81.0 ± 3.1	52.5 ± 2.1	78.9 ± 1.9	90.4 ± 5.7	75.0 ± 3.1	59.9 ± 4.3	$\mathbf{86.0 \pm 1.8}$ *
	DCNN	49.1	33.5	–	–	52.1	67.0	61.3	56.6	62.6
	PatchySan	71.0 ± 2.2	45.2 ± 2.8	86.3 ± 1.6	49.1 ± 0.7	72.6 ± 2.2	$\mathbf{92.6 \pm 4.2}$ *	75.9 ± 2.8	60.0 ± 4.8	78.6 ± 1.9
	DGCNN	70.0	47.8	–	–	73.7	85.8	75.5	58.6	74.4
	AWL	74.5 ± 5.9	51.5 ± 3.6	87.9 ± 2.5	54.7 ± 2.9	73.9 ± 1.9	87.9 ± 9.8	–	–	–
GNN variants	Sum–MLP (**GIN-0**)	$\mathbf{75.1 \pm 5.1}$	$\mathbf{52.3 \pm 2.8}$	$\mathbf{92.4 \pm 2.5}$	$\mathbf{57.5 \pm 1.5}$	$\mathbf{80.2 \pm 1.9}$	$\mathbf{89.4 \pm 5.6}$	$\mathbf{76.2 \pm 2.8}$	$\mathbf{64.6 \pm 7.0}$	$\mathbf{82.7 \pm 1.7}$
	Sum–MLP (**GIN-ϵ**)	$\mathbf{74.3 \pm 5.1}$	$\mathbf{52.1 \pm 3.6}$	$\mathbf{92.2 \pm 2.3}$	$\mathbf{57.0 \pm 1.7}$	$\mathbf{80.1 \pm 1.9}$	$\mathbf{89.0 \pm 6.0}$	$\mathbf{75.9 \pm 3.8}$	63.7 ± 8.2	$\mathbf{82.7 \pm 1.6}$
	Sum–1-Layer	74.1 ± 5.0	$\mathbf{52.2 \pm 2.4}$	90.0 ± 2.7	55.1 ± 1.6	$\mathbf{80.6 \pm 1.9}$	$\mathbf{90.0 \pm 8.8}$	$\mathbf{76.2 \pm 2.6}$	63.1 ± 5.7	82.0 ± 1.5
	Mean–MLP	73.7 ± 3.7	$\mathbf{52.3 \pm 3.1}$	50.0 ± 0.0	20.0 ± 0.0	79.2 ± 2.3	83.5 ± 6.3	75.5 ± 3.4	$\mathbf{66.6 \pm 6.9}$	80.9 ± 1.8
	Mean–1-Layer (GCN)	74.0 ± 3.4	51.9 ± 3.8	50.0 ± 0.0	20.0 ± 0.0	79.0 ± 1.8	85.6 ± 5.8	76.0 ± 3.2	64.2 ± 4.3	80.2 ± 2.0
	Max–MLP	73.2 ± 5.8	51.1 ± 3.6	–	–	–	84.0 ± 6.1	76.0 ± 3.2	64.6 ± 10.2	77.8 ± 1.3
	Max–1-Layer (GraphSAGE)	72.3 ± 5.3	50.9 ± 2.2	–	–	–	85.1 ± 7.6	75.9 ± 3.2	63.9 ± 7.7	77.7 ± 1.5

CONCLUSIONS

- **Contribution**
    - Make GNN model good at distinguish Graph isomorphim
- **Theory**
    - Mathematically proved the theory and argued against heuristics and intuition
- **Performance**
    - Showed better performance than existing state-of-the-art models

FINISH