▶ Lecture 10 EN.553.744 Prof. Luana Ruiz

Last time: - Conv. GNNs cannot distinguish graphs with identical eigenvalues
↳ yet many non-isomorphic graphs share spectra
- graph isomorphism not known to be solvable in polynomial time
↳ WL test is a workable heuristic

Today: - are GNNs are as powerful as the WL test?
- computational graphs
- requirements for a powerful GNN
- graph isomorphism network

→ How powerful are GNNs? (Xu et al., 2019)

(Prop.) If a GNN can distinguish between graphs (*) G_1 & G_2, then so can the WL test.

Pf. We consider the AGGREGATE-COMBINE representation of GNNs:

$$[a_\ell]_i = \text{AGGREGATE}\left(\{[x_{\ell-1}]_u : u \in \mathcal{N}(v)\}\right) \quad (1)$$

$$[x_\ell]_i = \text{COMBINE}\left([a_\ell]_i, [x_{\ell-1}]_i\right) \quad (2)$$

with permutation-invariant READOUT layer:

$$X_G = \text{READOUT}\left(\{[X_\ell]_i \mid i \in \mathcal{V}\}\right) \quad (3)$$

- Suppose after ℓ iterations, $X_{G_1} \neq X_{G_2}$ but the WL test cannot decide if G_1 & G_2 are non-isomorphic

$\Rightarrow$ from itt $j=0$ to ℓ in WL test, G_1 & G_2 have the same color multisets $\{[c_j]_v\}$ & color neighborhoods $\{[c_j]_u : u \in \mathcal{N}(v)\}$

$$\boxed{\text{If the WL test for } v \in G_1 \text{ \& } v' \in G_2 \text{ produces } [c_j]_v = [c_j]_{v'} \ \forall\, v, v', \text{ then the GNN always produces } [x_j]_v = [x_j]_{v'}}$$

$\to$ Assuming $[x_0]_v = [x_0]_{v'} = 1 \Rightarrow$ this clearly holds for $j=0$

→ Induction hypothesis:
Suppose this holds for iteration j, and:

•) WL test outputs the same colors $[c_{j+1}]_v$ and $[c_{j+1}]_{v'}$ $\forall$ v, v', i.e.:

$$([c_j]_v, \{[c_j]_u : u \in \mathcal{N}(v)\}) = ([c_j]_{v'}, \{[c_j]_{u'} : u' \in \mathcal{N}(v')\})$$

•) By the induction hypothesis at itt j, we must have:

$$([x_j]_v, \{[x_j]_u : u \in \mathcal{N}(v)\}) = ([x_j]_{v'}, \{[x_j]_{u'} : u' \in \mathcal{N}(v')\})$$

Since the same AGGREGATE & COMBINE functions are applied to both G_1 & G_2, we have $[x_j]_v = [x_j]_{v'}$ $\forall$ v, v'

→ By induction, if $[c_j]_v = [c_j]_{v'}$ $\forall$ v, v', then $[x_j]_v = [x_j]_{v'}$ $\forall$ v, v' at any itt j

↳ This creates a valid map $[x_j]_v = g([c_j]_v)$ $\forall$ v, hence:

$$\{\{([x_j]_v, \{[x_j]_u : u \in \mathcal{N}(v)\})\}\}$$

$$= \{\{(g([c_j]_v), \{g([c_j]_u) : u \in \mathcal{N}(v)\})\}\}$$

Thus the $[x_j]_v$ & $[x_j]_{v'}$ are the same for all j, and due to permutation invariance of readout, so are $x_{G_1} = x_{G_2}$ => proof by contradiction ■

We have shown:

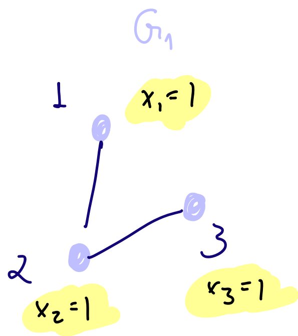
GNN tells G_1 & G_2 apart => WL test tells G_1 & G_2 apart

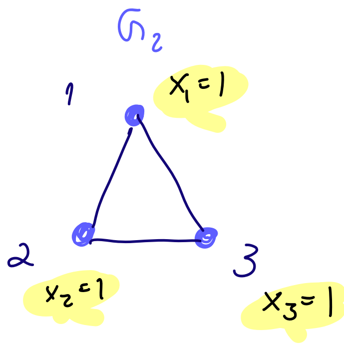
For arbitrary GNNs, is it true that:

WL test tells G_1 & G_2 apart => GNN tell G_1 & G_2 apart ?

$$\boxed{\text{I.e., are GNNs as powerful as the WL test at telling non-isomorphic graphs apart?}}$$

E.g.:





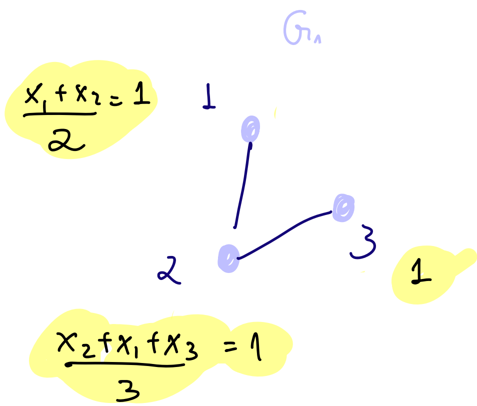
$(j=0)$

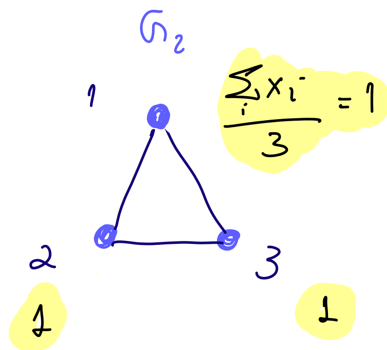
→ Can GCN tell them apart?

In aggregate-combine form, GCN is given by:

(assume $W=1$ w.l.o.g.)

$$[x_j]_v = \mathrm{ReLU}\left(W \cdot \mathrm{MEAN}\{[x_{j-1}]_u : u \in \mathcal{N}(v) \cup \{v\}\}\right)$$





$(j=1)$

$(j=2)$

⋮

$$\mathrm{READOUT}\{1,1,1\} = \mathrm{READOUT}\{1,1,1\}$$

↳ fails for any permutation invariant READOUT

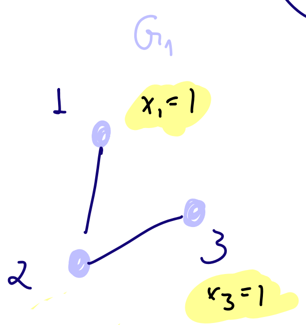
Exercise: Check if SAGE with MAX aggregation:

$$[x_j]_v = \mathrm{MAX}\left(\{\mathrm{ReLU}(W.[x_j]_u), \forall u \in \mathcal{N}(v)\}\right)$$

can tell these graphs apart

$\Rightarrow$ not every GNN is as powerful as WL

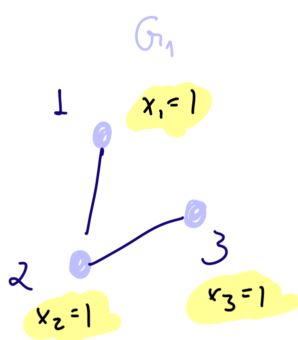
- Where do things break?



Positive: GCN (& any perm. equivariant GNN for that matter) does not distinguish btw 1 & 3

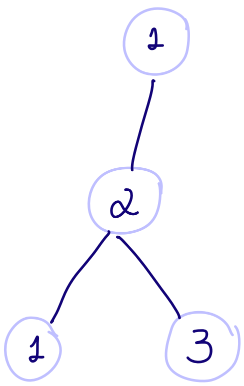
$\rightarrow$ this is great! implicit data augmentation mechanism

Negative: The GCN cannot distinguish btw nodes 1/3 & 2 in G_1:

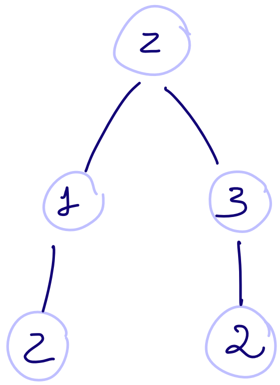


Define each node's computational graph:

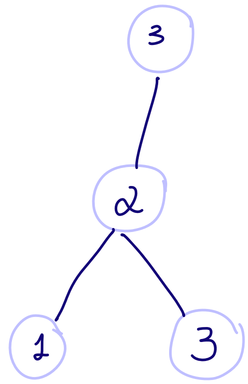
• node 1:



• node 2:



• node 3:

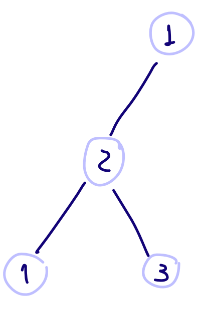


(L=2 is enough - it is the graph diameter)

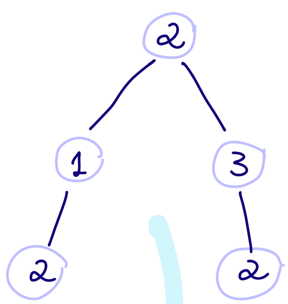
The issue with GCN is it is mapping ≠ comp. graphs to the same embedding

(computational graph space)

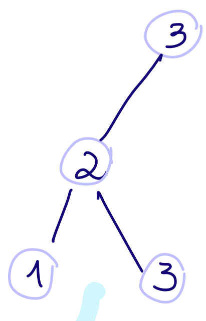
• node 1:



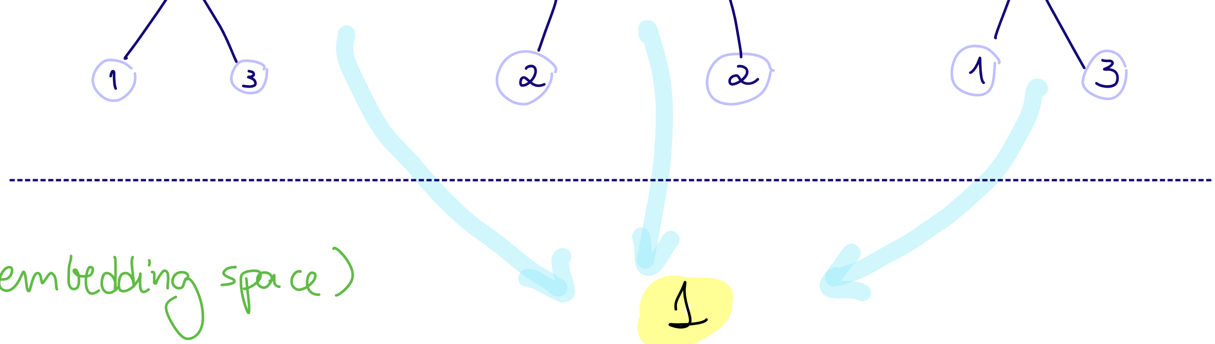
• node 2:



• node 3:



(embedding space)



(Thm) Let G_1 & G_2 be two graphs the WL test decides are non-isomorphic. A GNN Φ maps G_1 & G_2 to $\Phi(G_1) \neq \Phi(G_2)$ if the following hold:

(1) Φ aggregates & updates as:

$$[x_\ell]_v = \phi\left([x_{\ell-1}]_v, \varphi\left(\{[x_{\ell-1}]_u : u \in N(v)\}\right)\right)$$

where ϕ & φ are injective

(2) the READOUT function is injective

Pf. sketch: recall from Prop. (*):

→ By induction, if $[c_j]_v = [c_j]_{v'}$ $\forall v, v'$, then $[x_j]_v = [x_j]_{v'}$ $\forall v, v'$ at any it j

↳ This creates a valid map $[x_j]_v = g([c_j]_v)$ $\forall v$;

We need g to be injective.

Let f be the hash fcn in WL; recall it is bijective for color multisets

$$[x_j]_v = g(f([c_{j-1}]_v, \{[c_{j-1}]_u : u \in \mathcal{N}(v)\}))$$

bijective

$$[x_j]_v = g(f(g^{-1}([x_{j-1}]_v), \{g^{-1}([x_{j-1}]_u) : u \in \mathcal{N}(v)\}))$$

left inverse

$\approx \phi(\cdot, \varphi(\cdot))$, the GNN

→ to have g injective, we need the GNN to be injective

→ if ϕ, φ are injective, the GNN is injective

ϕ, φ injective $\Rightarrow$ GNN injective $\Rightarrow$ g is injective $\Rightarrow$ GNN is as powerful as WL.