Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 11, n. 1, 2025.

Trabalho apresentado no XLIII CNMAC, Centro de Convenções do Armação Resort - Porto de Galinhas - PE, 2024

**Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**

# SRGNN: Simple Recurrent Graph Neural Network

Juan Belieni[1], Diego Mesquita[2]
School of Applied Mathematics, Getulio Vargas Foundation, Rio de Janeiro, RJ

Graph Neural Networks (GNNs) are capable of learning graph representations via message passing, which consists of performing aggregation and combination operations on the vertices of a graph. The aggregation operation computes a message based on the representation of the neighborhood of a vertex, and the combination operation updates its representation based on the message and the current representation of the vertex. The expressive power of standard GNNs is tied to the color refinement algorithm for detecting isomorphism [2]. Noticeably, this property only requires that we have injective aggregation and combination operations.

Typically, GNN variants like GCN [3] and GIN [4] are implemented as a multi-layer neural network, where each layer can learn different weights. However, it is not so common to define these networks as recurrent GNNs, where all the layers have the same weights. This work has the goal of evaluating the learning capabilities of this simpler variant of GNNs, that will be called Simple Recurrent GNNs (SRGNNs).

The standard variant of GNNs are the Message Passing Neural Networks (MPNNs) [1], being the base for the definition of SRGNNs. This type of network is defined as follows:

**Definition 1.** *Let $G = (V, E)$ be a graph and $\xi$ be a $L$-layer MPNN. For all layers $\zeta^{(l)}$ of $\xi$, $1 \leq l \leq L$, the network updates the representation $h_v^{(l)}$ for all vertices $v \in V$ by performing*

$$m_v^{(l)} = agg_{\theta^{(l)}}(\{\!\{h_u^{(l-1)} \mid u \in \mathcal{N}(v)\}\!\}), \tag{1}$$

$$h_v^{(l)} = comb_{\phi^{(l)}}(h_v^{(l-1)}, m_v^{(l)}), \tag{2}$$

*where $\{\!\{\cdot\}\!\}$ denotes a multiset, $\mathcal{N}(v)$ denotes the neighborhood of $v$ in $G$ and $h_v^{(0)}$ represents the initial state of the vertex $v$. Also, $agg_{\theta^{(l)}}$ and $comb_{\phi^{(l)}}$ are functions indexed by the weights $\theta^{(l)}$ and $\phi^{(l)}$, respectively, where $agg_{\theta^{(l)}}$ must be a function that operates on multisets and, as a consequence, be permutation invariant [5].*

MPNNs are known to have the same expressiveness as the color refinement algorithm to distinguish graphs. This fact is described by the theorem 1:

**Theorem 1** ([2])**.** *Let $G = (V(G), E(G))$ and $H = (V(H), E(H))$ be graphs and $\xi$ be a $L$-layer MPNN. Then, for all vertices $v_G \in V(G)$ and $v_H \in V(H)$,*

$$cr^{(L)}(G, v_G) = cr^{(L)}(H, v_H) \implies \xi(G, v_G) = \xi(H, v_H),$$

*where $cr^{(L)}$ denotes the color refinement algorithm with $L$ interactions and $\xi(G, v_G)$ denotes the representation of the node $v_G$ calculated by the last layer of $\xi$ on $G$, i.e., $h_{v_G}^{(L)}$.*

The SRGNN, as mentioned before, is defined as a special case of a MPNN, and inherits its expressive power:

[1] juanbelieni@gmail.com
[2] diego.mesquita@fgv.br

**Definition 2.** *Let $G = (V, E)$ be a graph and $\xi$ be a $L$-layer MPNN. $\xi$ is called a SRGNN if all the layers $\zeta^{(l)}$, $1 \leq l \leq L$, have the same weights, i.e., if $\theta^{(l)} = \theta$ and $\phi^{(l)} = \phi$, $1 \leq l \leq L$. This layer will simply be called $\zeta$.*

The model implementation for the SRGNN is very concise. First, each vertex $v \in G$ passes through a MLP, $\phi_{\text{input}} : \mathbb{R}^D \mapsto \mathbb{R}^H$. After this, the layer $\zeta$ of the SRGNN is recurrently applied to the output of the last layer $L$ times. Finally, all the representations computed by the last layer are pooled, and the result is passed through another MLP, $\phi_{\text{output}} : \mathbb{R}^H \mapsto \mathbb{R}^{|\mathcal{C}|}$, where $|\mathcal{C}|$ is the number of classes, calculating the probabilities for each class.

At certain datasets, this model obtained similar performance when compared to the GIN-0 model from Xu, Hu, Leskovec, and Jegelka [4]'s paper. The results from the experiments and the equivalent results for the GIN-0 are presented in table 1.

Table 1: experiment accuracy (%) mean and standard deviation results.

| Model | MUTAG | PROTEINS | PTC | NCI1 |
|---|---|---|---|---|
| GIN-0 | 89.4 ± 5.6 | 76.2 ± 2.8 | 64.6 ± 7.0 | 82.7 ± 1.7 |
| SRGNN | 90.0 ± 2.9 | 73.7 ± 1.3 | 62.9 ± 3.0 | 67.5 ± 2.9 |

# References

[1] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. "Neural Message Passing for Quantum Chemistry". In: **Proceedings of the 34th International Conference on Machine Learning**. International Conference on Machine Learning. PMLR, July 17, 2017, pp. 1263–1272. URL: https://proceedings.mlr.press/v70/gilmer17a.html (visited on 03/06/2024).

[2] M. Grohe. **The Logic of Graph Neural Networks**. Jan. 9, 2022. DOI: 10.48550/arXiv.2104.14624. arXiv: 2104.14624 [cs]. URL: http://arxiv.org/abs/2104.14624 (visited on 02/07/2024). preprint.

[3] T. N. Kipf and M. Welling. **Semi-Supervised Classification with Graph Convolutional Networks**. Feb. 22, 2017. DOI: 10.48550/arXiv.1609.02907. arXiv: 1609.02907 [cs, stat]. URL: http://arxiv.org/abs/1609.02907 (visited on 03/05/2024). preprint.

[4] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. **How Powerful Are Graph Neural Networks?** Feb. 22, 2019. DOI: 10.48550/arXiv.1810.00826. arXiv: 1810.00826 [cs, stat]. URL: http://arxiv.org/abs/1810.00826 (visited on 02/28/2024). preprint.

[5] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. "Deep Sets". In: **Advances in Neural Information Processing Systems**. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html (visited on 03/06/2024).