A Theoretical and Empirical Investigation into the Equivalence of Graph Neural Networks and the Weisfeiler-Leman Algorithm

From the faculty of Mathematics, Physics, and Computer Science for the purpose of obtaining the academic degree of Bachelor of Sciences.

Eric Tillmann Bill

Supervision:
Prof. Dr. rer. nat. Christopher Morris

Informatik 6
RWTH Aachen University

Contents

1 Definition

Definition 1 (1-WL Relation)**.** For any graphs G, H we will denote $G \simeq_{1\text{WL}} H$ if the 1-WL isomorphism test can not distinguish both graphs. Note that due to the soundness of this algorithm, if $G \not\simeq_{1\text{WL}} H$, we always can conclude that $G \not\simeq H$.

Definition 2. Let $f : \mathcal{X}^{n \times n} \to A^K$ be a well-defined encoding function compatible with the 1-WL+NNframework and R an arbitrary domain. Then we call $\mathcal{C}$ a collection of permutation invariant functions from $\mathcal{X}^{n \times n}$ to R that are computed by 1-WL+NN, where for every MLP working over A^K to R there exists $\beta \in \mathcal{C}$ with: $\beta : \mathcal{X}^{n \times n} \to R, G \mapsto \text{MLP} \circ f(G)$.

Definition 3. Let $\mathcal{C}$ be a collection of permutation invariant functions from $\mathcal{X}^{n \times n}$ to $\mathbb{R}$. We say $\mathcal{C}$ is **1-WL-Discriminating** if for all graphs $G_1, G_2 \in \mathcal{X}$ for which the 1-WL isomorphism test concludes non-isomorphic, there exists a function $h \in \mathcal{C}$ such that $f(G_1) \neq f(G_2)$.

Definition 4. Let $\mathcal{C}$ be a collection of permutation invariant functions from $\mathcal{X}^{n \times n}$ to $\mathbb{R}$. We say $\mathcal{C}$ is **GNN-Approximating** if for all permutation-invariant functions $\mathcal{A}$ computed by a GNN, and for all $\epsilon \in \mathbb{R}$ with $\epsilon > 0$, there exists $h_{\mathcal{A},\epsilon} \in \mathcal{C}$ such that $\|\mathcal{A} - h_{\mathcal{A},\epsilon}\|_\infty := \sup_{G \in \mathcal{X}} |f(G) - h_{\mathcal{A},\epsilon}(G)| < \epsilon$

2 Theorems

In this thesis we concentrate on finite graphs. We therefore, let $\mathcal{X}$ be $\mathcal{X} := \{1, \dots, k\}$ for an arbitrary $k \in \mathbb{N}$.

Theorem 5. Let $\mathcal{C}$ be a collection of functions computed by 1-WL+NNmodels, then C is **1-WL-Discriminating**.

Theorem 6. Let $\mathcal{C}$ be a collection of permutation-invariant functions that is **1-WL-Discriminating**, than $\mathcal{C}$ is also **GNN-Approximating**.

3 Proofs

Since we concentrate in this thesis on finite graphs, let $\mathcal{X}$ be $\mathcal{X} := \{1, \dots, k\}$ for an arbitrary $k \in \mathbb{N}$.

3.1 1-WL+NNis 1-WL-Discriminating

Proof. We consider the collection $\mathfrak{B}_k$ of permutation invariant functions, where every $\mathcal{B} \in \mathfrak{B}_k$ is of the form $\mathcal{B} = \text{MLP} \circ f$. Here MLP is an arbitrary multilayer perceptron mapping vectors from $\mathbb{N}^K$ to $\mathbb{R}$ and f the *counting-encoding* function. Further, let $G_1, G_2 \in \mathcal{X}^{n \times n}$ such that the 1-WL isomorphism test concludes non-isomorphic ($G_1 \not\simeq G_2$). We denote with $(C_\infty)_G$ the final coloring computed by the 1-WL algorithm when applied on G.

Due to the 1-WL isomorphism test concluding $G_1 \not\simeq G_2$, there exists a color $c \in \mathbb{N}$ such that $(C_\infty)_{G_1}(c) \neq (C_\infty)_{G_2}(c)$. If we now consider as MLP the following function $\text{MLP} : \mathbb{N}^K \to \mathbb{R}, v \mapsto W \cdot v$ with $W \in \mathbb{N}^{1 \times K}$ such that $W_{1,c} := 1$ and $W_{1,i} := 0$ for all $i \in [K] \setminus \{c\}$. Then we can conclude that $\mathcal{B}(G_1) \neq \mathcal{B}(G_2)$. Since G_1, G_2 are arbitrary, we can conclude the proof. $\square$

3.2 1-WL-Discrimnating is also GNN-Approximating

Lemma 7 (1-WL+NNPermuation Invariance)**.** Let $\mathcal{C}$ be a collection of functions computed by 1-WL+NN, then every function $\mathcal{B} \in \mathcal{C}$ is permutation-invariant.

Proof. Let $\mathcal{C}$ be a collection of functions computed by 1-WL+NN. Let $\mathcal{B}$ be an arbitrary function in $\mathcal{C}$, then $\mathcal{B}$ is comprised as follows: $\mathcal{B}(\cdot) = \text{MLP} \circ f_{\text{enc}} \circ \text{1-WL}(\cdot)$. Since, the 1-WL coloring algorithm is permutation-invariant, the overall function is permutation invariant. □

Lemma 8 (Composition Lemma)**.** Let $\mathcal{C}$ be a collection of functions computed by 1-WL+NN. Further, $h_1, \dots h_n \in \mathcal{C}$ and MLP a multilayer perceptron, than the function $\mathcal{A}$ composed of $\mathcal{A}(G^*) := \text{MLP}(h_1(G^*), \dots, h_n(G^*))$ is also computable by 1-WL+NN.

Proof. Assume the above and let f_{enc} be the encoding functions used by $h_1, \dots h_n$. Further, let $\text{MLP}_1, \dots, \text{MLP}_n$ be the multilayer perceptrons the functions $h_1, \dots h_n$ are composed of respectively. The idea of this proof is, that we construct a new MLP that first duplicates the encoding vector n times, simulates $\text{MLP}_1, \dots, \text{MLP}_n$ each on a copy of the encoding vector in parrallel, and then use the concatetated outut of all of them as input for the new final MLP. See Figure **??** for a sketch of the proof idea. A complete proof can be found in the Appendix, as this proof is very technical and not that interesting. □

$$
\begin{tikzcd}
 & & & v \arrow[r, mapsto, "\mathrm{MLP}_1"] & o_1 \arrow[dr] & \\
G \arrow[r, mapsto, "\text{1-WL}"] & (C_\infty)_G \arrow[r, mapsto, "f"] & v \arrow[ur] \arrow[dr] & \cdot & & o \\
 & & & v \arrow[r, mapsto, "\mathrm{MLP}_n"] & o_n \arrow[ur] &
\end{tikzcd}
$$

Figure 1: Sketch of the proof we use to prove lemma XYZ.

Lemma 9 (My Lemma)**.** Let $\mathcal{C}$ be a collection of permuation-invariant functions from $\mathcal{X}^{n \times n}$ to $\mathbb{R}$ computed by 1-WL+NNthat is 1-WL-Discriminating. Then for all $G \in \mathcal{X}^{n \times n}$, there exists a function h_G computable by 1-WL+NNsuch that for all $G^* \in \mathcal{X}^{n \times n} : h_G(G^*) = 0$ if and only if $G \simeq_{\text{1WL}} G^*$.

Proof. For any $G_1, G_2 \in \mathcal{X}^{n \times n}$ with $G_1 \not\simeq_{\text{1WL}} G_2$ let $h_{G_1,G_2} \in \mathcal{C}$ be the function distinguishing them, with $h_{G_1,G_2}(G_1) \neq h_{G_1,G_2}(G_2)$. We define the function $\overline{h}_{G_1}$ working over $\mathcal{X}^{n \times n}$ as follows:

$$
\begin{aligned}
\overline{h}_{G_1}(\cdot) &= |h_{G_1,G_2}(\cdot) - h_{G_1,G_2}(G_1)| \\
&= \max(h_{G_1,G_2}(\cdot) - h_{G_1,G_2}(G_1)) + \max(h_{G_1,G_2}(G_1) - h_{G_1,G_2}(\cdot))
\end{aligned}
$$

Note, that $h_{G_1,G_2}(G_1)$ in the formula is a fixed constant. With this function we can ensure, that if $G^* \simeq_{\text{1WL}} G : \overline{h}_{G_1,G_2}(G^*) = 0$, otherwise if $G^* \not\simeq_{\text{1WL}} G : \overline{h}_{G_1,G_2}(G^*) > 0$. This is a result of the permutation-invariance of all functions in $\mathcal{C}$. FUCK DAS GEHT NICHT! PERMUTATION INVARIANCE KANN HIER NICHT DIREKT ANGEWENDET WERDEN :3

□

Lemma 10. Let $\mathcal{C}$ be a class of permutation-invariant functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$ computed by 1-WL+NNso that for all $G \in \mathcal{X}^{n\times n}$, there exists $h_G \in \mathcal{C}$ satisfying $h_G(G^*) = 0$ if and only if $G \simeq_{1\text{WL}} G^*$ for all $G^* \in \mathcal{X}^{n\times n}$. Then for every $G \in \mathcal{X}^{n\times n}$, there exists a function φ_G computable by 1-WL+NNsuch that for all $G^* \in \mathcal{X}^{n\times n}$: $\varphi_G(G^*) = \mathbb{1}_{G\simeq_{1\text{WL}}G^*}$.

Proof. Following Lemma 9, for every $G \in \mathcal{X}^{n\times n}$ there exists $h_G \in \mathcal{C}$ such that for all $G^* \in \mathcal{X}^{n\times n} : h_G(G^*) = 0$ if and only if $G \simeq_{1\text{WL}} G^*$. Due to $\mathcal{X}$ being finite, we can define for every graph G the constant:

$$\delta_G := \frac{1}{2} \min_{G^* \in \mathcal{X}^{n\times n}, G \not\simeq_{1\text{WL}} G^*} |h_G(G^*)| > 0.$$

With this constant, we can use a so-called "bump" function working from $\mathbb{R}$ to $\mathbb{R}$ that will be similar to the indicator function. For parameter $a \in \mathbb{R}$ with $a > 0$ let:

$$\psi_a(x) := \max(\frac{x}{a} - 1,\ 0) + \max(\frac{x}{a} + 1,\ 0) - 2 \cdot \max(\frac{x}{a},\ 0).$$

The interesting property of ψ_a is that it maps every value x to 0, except when x is being drawn from the interval $(-a, a)$. In particular, it maps x to 1 if and only if x is equal to 0. See Figure **??** in the Appendix for a plot of the relevant part of this function with exemplary values for a.

We use this property to define for every graph $G \in \mathcal{X}^{n\times n}$ the function $\varphi_G(G^*) := \psi_{\delta_G}(h_G(G^*))$. We will quickly demonstrate that this function is equal to the indicator function, let G be fixed and G^* an arbitrary graph from $\mathcal{X}^{n\times n}$:

1. If $G \simeq_{1\text{WL}} G^*$, then $h_G(G^*) = 0$ resulting in $\varphi_G(G^*) = \psi_{\delta_G}(0) = 1$.
2. If $G \not\simeq_{1\text{WL}} G^*$ then $h_G(G^*) > 0$, such that $|h_G(G^*)| > \delta_G$ resulting in $\varphi_G(G^*) = 0$.

Note we can encode each φ_G via a single MLP layer, where δ_G is a constant and the max operator is replaced by the non-linear activation function ReLU of the layer. With the Composition Lemma 8 we can therefore conclude that φ_G is computable by 1-WL+NNfor every graph $G \in \mathcal{X}^{n\times n}$. □

Lemma 11. Let $\mathcal{C}$ be a collection of permutation-invariant functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$ computed by 1-WL+NNso that for all $G \in \mathcal{X}^{n\times n}$, there exists $\varphi_G \in \mathcal{C}$ satisfying $\varphi_G(G^*) = \mathbb{1}_{G\simeq_{1\text{WL}}G^*}$ for all $G^* \in \mathcal{X}^{n\times n}$, then C is also GNN-Approximating.

Proof. Assume the above. For any function $\mathcal{A}$ computed by an GNN, we want to show that we can decompose it as follows for any $G^* \in \mathcal{X}^{n\times n}$:

$$\begin{aligned}
\mathcal{A}(G^*) &= \Big(\frac{1}{|\mathcal{X}^{n\times n}/\!\simeq_{1\text{WL}}(G^*)|} \sum_{G\in\mathcal{X}^{n\times n}} \mathbb{1}_{G\simeq_{1\text{WL}}G^*} \Big) \cdot \mathcal{A}(G^*) \\
&= \frac{1}{|\mathcal{X}^{n\times n}/\!\simeq_{1\text{WL}}(G^*)|} \sum_{G\in\mathcal{X}^{n\times n}} \mathcal{A}(G) \cdot \mathbb{1}_{G\simeq G^*} \\
&= \sum_{G\in\mathcal{X}^{n\times n}} \frac{\mathcal{A}(G)}{|\mathcal{X}^{n\times n}/\!\simeq_{1\text{WL}}(G)|} \cdot \varphi_G(G^*) \qquad (0.1)
\end{aligned}$$

with $\mathcal{X}^{n\times n}/\!\simeq_{1\text{WL}}(G^*)$ denoting the set of all graphs G over $\mathcal{X}^{n\times n}$ that are equivalent to G^* according to the $\simeq_{1\text{WL}}$ relation.

Since $\mathcal{A}$ is permutation-invariant, and GNNs are at most as good as the 1-WL algorithm in distinguishing non-isomorphic graphs, we can use the fact that for every graph $G, H \in \mathcal{X}^{n\times n}$

with $G \simeq_{1\text{WL}} H$: $\mathcal{A}(G) = \mathcal{A}(H)$. Therefore, we can decompose $\mathcal{A}$ as outlined above. We can encode this decomposition in a single MLP layer with $\frac{\mathcal{A}(G)}{|\mathcal{X}^{n\times n}/\simeq_{1\text{WL}}(G)|}$ being a constant and $\varphi_G \in \mathcal{C}$ encoding the indicator function. Combined with the Lemma 8, we can conclude that $\mathcal{A}$ is computable by 1-WL+NN. Important to note, we can only do this since $\mathcal{X}$ is finite, making the overall sum finite and the size of $\mathcal{X}^{n\times n}/\simeq_{1\text{WL}}(G)$ well-defined for all graphs.

□

Appendix

Figures and graphs

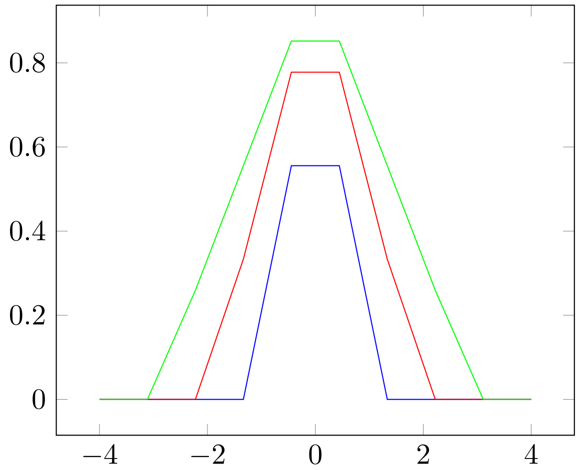

Figure 2: Illustration of the so-called "bump" function $\psi_a(x)$ with $a = 1$ in blue, $a = 2$ in red and $a = 3$ in green.

Proofs